

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ВНУТРІШНІХ СПРАВ**

Балтовський О.А., Форос Г.В., Пядишев В.Г., Сіфоров О.І.

СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Навчальний посібник

**Одеса
2022**

*Рекомендовано до друку науково-методичною радою
Одеського державного університету внутрішніх справ*

Авторський колектив:

Балтовський О.А. – доктор технічних наук, доцент, професор кафедри кібербезпеки та інформаційного забезпечення ОДУВС;

Форос Г.В. – кандидат юридичних наук, доцент, доцент кафедри кібербезпеки та інформаційного забезпечення ОДУВС;

Пядишев В.Г. – доктор юридичних наук, доцент, професор кафедри кібербезпеки та інформаційного забезпечення ОДУВС

Сіфоров О.М. – кандидат технічних наук, доцент, доцент кафедри кібербезпеки та інформаційного забезпечення ОДУВС

Рецензенти:

М.В. Корнієнко - доктор юридичних наук, професор, проректор Одеського державного університету внутрішніх справ;

Н.І. Логінова – кандидат педагогічних наук, доцент, завідувачка кафедри інформаційних технологій Національного університету «Юридична академія»

Балтовський О.А., Форос Г.В., Пядишев В.Г., Сіфоров О.І.

Системи підтримки прийняття рішень/ за заг. ред. д.т.н., доц.

О.А. Балтовського. Одеський держ. унів-т внутр. справ, 2022. 148 с.

В навчальному посібнику розглянуто найважливіші теоретичні та практичні питання загальні принципи підтримки ухвалення управлінських рішень, широко поширені в сучасних умовах, і повсюдно застосовні методи адаптації до підтримки прийняття управлінських рішень популярного на практиці сучасного програмного забезпечення, призначеного для автоматизації обліку та управління.

Посібник розрахований на здобувачів вищої освіти, вищих навчальних закладів спеціальності «Системний аналіз» та може бути корисний викладачам, аспірантам та всім, хто цікавиться системним, кримінальним аналізом та інформаційно-аналітичним забезпеченням.

УДК 004.5197(076)

© О.А. Балтовський, Г.В. Форос,
В.Г. Пядишев, О.І. Сіфоров
© ОДУВС, 2022

ЗМІСТ

ПЕРЕДМОВА	6
Розділ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ЗВ'ЯЗКУ ІНФОРМАЦІЇ, МОДЕЛЮВАННЯ, УПРАВЛІННЯ	13
1.1. Логічні аспекти поняття «інформаційна система»	13
<i>Контрольні питання до підрозділу 1.1.</i>	<i>18</i>
1.2. Інформаційні системи автоматизації обліку та управління.....	18
<i>Контрольні питання до підрозділу 1.2</i>	<i>21</i>
1.3. Інформатика, моделювання та підтримка управлінських рішень	22
<i>Контрольні питання до підрозділу 1.3</i>	<i>25</i>
Розділ 2. ІНФОРМАЦІЙНА ПІДТРИМКА ПРИНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ	26
2.1. Змістова сутність підтримки прийняття рішень.....	26
<i>Контрольні питання до підрозділу 2.1</i>	<i>32</i>
2.2. Коротка історія розвитку «систем підтримки прийняття рішень»	32
<i>Контрольні питання до підрозділу 2.2</i>	<i>36</i>
2.3. Класифікація «систем підтримки прийняття рішень	36
<i>Контрольні питання до підрозділу 2.3</i>	<i>38</i>
Розділ 3. ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ	39
3.1. Основні поняття та архітектура систем баз даних	39
3.1.1. Основні поняття і визначення	39
3.1.2. Архітектура СКБД	40
3.1.3. Робота з СКБД	41
3.1.4. Класифікація СКБД	43
3.1.5. Функції СКБД	45
<i>Контрольні питання до підрозділу 3.1</i>	<i>47</i>
3.2. Реляційна модель даних та нормалізація	48
3.2.1. Реляційна структура даних	48
3.2.2. Типи таблиць і ключів в реляційних базах даних	50
3.2.3. Типи відношень	52
3.2.4. Фізична організація файлів баз даних	52
3.2.5. Тринадцять правил Кодда для реляційних СКБД	54
3.2.6. Нормалізація даних в реляційній моделі	57
3.2.7. Деякі поняття алгебри відношень	58
<i>Контрольні питання до підрозділу 3.2</i>	<i>61</i>

3.3. Загальна характеристика баз знань	62
3.3.1. Базові поняття щодо баз знань	62
3.3.2. Стратегії отримання знань	63
3.3.3. Висновки на базі знань	64
3.3.4. Елементи експертних систем	66
<i>Контрольні питання до підрозділу 3.3</i>	68
Розділ 4. АНАЛІЗ ДАНИХ І ЗНАНЬ	69
4.1. Основні поняття аналізу даних і знань	69
4.1.1. Чим відрізняються «дані» від «знань»?	69
4.1.2. Що є аналізом даних?	70
4.1.3. Що є аналізом знань?	72
<i>Контрольні питання до підрозділу 4.1</i>	73
4.2. Розділи аналізу даних	73
4.2.1. Попередня обробка	73
4.2.2. Кореляційний аналіз	74
4.2.3. Дисперсний аналіз	74
4.2.4. Регресійний аналіз	76
4.2.5. Коваріаційний аналіз	79
4.2.6. Дискримінантний аналіз	79
4.2.7. Кластерний аналіз	79
4.2.8. Аналіз часових рядів	81
<i>Контрольні питання до підрозділу 4.2</i>	85
4.3. Методи інтелектуального аналізу даних	86
4.3.1. Проблеми аналізу даних із застосуванням класичних статистичних підходів	86
4.3.2. Технологія KDD (Knowledge Discovery in Databases)	87
4.3.3. Відмінності між засобами KDD і OLAP	87
<i>Контрольні питання до підрозділу 4.3</i>	88
4.4. Data mining (виявлення знань в базах даних)	89
4.4.1. Загальні відомості	89
4.4.2. Постановка завдання	90
4.4.3. Data mining і бази даних	91
4.4.4. Коло завдань	92
<i>Контрольні питання до підрозділу 4.4</i>	93
4.5. Інтерактивна аналітична обробка даних (OLAP)	93
4.5.1. Правила аналітичної обробки даних в реальному часі	93
4.5.2. Дія OLAP	95
4.5.3. Реалізація OLAP	96
<i>Контрольні питання до підрозділу 4.5</i>	98

Розділ 5. СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

НА ОСНОВІ СХОВИЩ ДАНИХ ТА OLAP-СИСТЕМ	99
5.1. Розвиток та застосування СППР на основі сховищ даних та OLAP-систем	99
5.1.1. Передумови та сутність СППР на основі сховищ даних та OLAP-систем	99
5.1.2. Базові концепції та визначення	107
5.1.3. Взаємопов'язана архітектура орієнтованих на дані СППР	108
5.1.4. Загальне проектування і процес розроблення орієнтованих на дані СППР	109
<i>Контрольні питання до підрозділу 5.1</i>	110
5.2. Концепція сховищ даних, її реалізація в інформаційних системах	111
5.2.1. Побудова сховищ даних	111
5.2.2. Архітектура сховищ даних	115
5.2.3. Моделі побудови сховищ даних	121
5.2.4. Проектування сховищ даних	126
<i>Контрольні питання до підрозділу 5.2</i>	129
5.3. Система аналітичного інтерактивного оброблення (OLAP)	130
5.3.1. Зародження і розвиток OLAP-систем	130
5.3.2. Інструментальні засоби кінцевих користувачів в OLAP	133
5.3.3. Система оперативного аналітичного оброблення Oracle Express OLAP	135
<i>Контрольні питання до підрозділу 5.3</i>	141
ЛІТЕРАТУРА	142

ПЕРЕДМОВА

Стандартизація більшою мірою визначає діяльність людини у професійній сфері. Це стосується і інформаційних технологій, зокрема, у зв'язку з розвитком систем підтримки прийняття рішень (Decision Support Systems – DSS). Чому доводиться звертати увагу на стандартизацію? Насамперед тому, що всіх цікавить комерційне застосування прикладних рішень. Продукт (товар, послуга) стає масовим лише після уніфікації вимог щодо нього, навіть з урахуванням побажань «вузьких» груп споживачів. Через гостру актуальність застосування систем підтримки прийняття рішень в різних галузях діяльності, за кордоном багато робіт присвячено різним аспектам їх розвитку. Серед вітчизняних на особливу увагу заслуговують роботи В. Ф. Ситника, зокрема навчальний посібник «Системи підтримки прийняття рішень», виданий у 2009 р.[1].

Сучасний спеціаліст, і не лише в галузі інформаційних технологій, як правило, адекватно сприймає аббревіатуру програми ERP (enterprise resource planning), що дослівно перекладається, як «планування ресурсів» і призначено для планування ресурсів, прискорення процесів, що ведуться у бізнесі. З його допомогою можна знизити негативний вплив людського фактору та оптимізувати функціонування компанії, всередині якої багато відділів, підрозділів та співробітників (при тому, що стандарти ERP – це лише етап у розвитку інформаційних стандартів ряду MRP, MRP II,...). Тут MRP (Manufacturing resource planning) – це планування ресурсів виробництва – метод ефективного планування всіх ресурсів виробничого підприємства. Отже система управління об'єднує: планування потреб у ресурсах, замкнутий цикл і плани потужностей виробництва... Хоча зовсім недавно це було не так. Досі багато в чому це – «не так» стосується поняття «системами підтримки прийняття рішень». З одного боку, ERP-системи та багато інших прикладних програмних засобів можна віднести за функціональним призначенням до систем підтримки прийняття рішень (вони і для цього створювалися). З іншого боку – сучасні інформаційні додатки повинні слідувати традиціям, що формуються. Деякі відхилення допустимі, але, як правило, стосовно певної базової специфікації.

Відповідно, виникає питання – що рахувати DSS (Decision Support System)? Це – СПР (система підтримки рішень) – комп'ютеризована система, яка через збирання та аналіз великого обсягу інформації може впливати на процес прийняття керівничих рішень в бізнесі та підприємстві. Це логічна чи інша характеристика? У 80-х роках XX ст. виникло визначення DSS як заснованої на використанні моделей сукупність процедур обробки даних і суджень, що допомагають у прийнятті рішень [2-4]. Або, DSS є класом комп'ютеризованих інформаційних систем, які підтримують діяльність із прийняття рішень [4,5]. Зазначене визначення мало що пояснює та не позиціонує DSS у широкому переліку видів інформаційних систем. Тому пізніше стали

додавати: «система, що полегшує прийняття рішень», «...що представляє дані у зручній для прийняття рішень формі» тощо.

Ось ще одне визначення (Bonczek, Holsapple&Whinston, 1981, [6]): DSS допомагає особі, яка приймає рішення, у вирішенні непрограмованих, неструктурованих (або напівструктурованих) проблем; DSS-система повинна пропонувати можливості формування інтерактивних запитів у природній мові, близькій до предметної та легко досліджуваної.

Пізніше, у 2002 році, Даніель Пауер [7] охарактеризував DSS-системи як системи, що оперують даними, зв'язками, документами, знаннями та моделями. За Пауером DSS-система – це інтерактивна комп'ютерна система, призначена для допомоги особі, яка приймає рішення, у використанні даних, зв'язків, документів, знань та моделей з метою ідентифікації проблем, формування рішень. І це вже принаймні більш конструктивно, хоча під дане визначення попадають багато видів сучасних інформаційних систем: ERP, GIS, DocFlow, BusinessModeller, SCADA/DCE, Project Management та ін.

У деяких визначеннях DSS згадується можливість увімкнення функціональних можливостей штучного інтелекту. Згадуються також як необхідні можливості графічного представлення даних. При цьому виявляється задіяним зв'язкове поняття – Business Intelligence Tools (інструментальні засоби бізнес-інтелекту) – програмне забезпечення, яке дає можливість користувачам спостерігати та використовувати великі обсяги даних. Зазвичай виділяють три типи таких інструментальних засобів:

- засоби багатовимірного аналізу - також відомі як OLAP (On-Line Analytical Processing) - програмне забезпечення, яке дає користувачеві можливість спостерігати дані в різних вимірах, напрямках або перерізах;
- інструментальні засоби запитів (QueryTools) – програмне забезпечення, що дозволяє формувати запити до даних за змістом чи зразком;
- інструментальні засоби пошуку даних (DataMiningTools) - програмне забезпечення (ПЗ), що здійснює автопошук зразків, залежностей у даних.

У сферу застосування систем DSS зараз, по суті, потрапляє більшість завдань, що покладаються на IT-служби. Це підтверджує аналіз ринку інформаційних систем. Ринок DSS систем, що оформився як ринок ПЗ до середини 90-х років XX ст., Нині зростає великими темпами. При цьому якщо ринок систем DSS в даний час в основному пов'язаний з фінансовим сектором, великоформатною торгівлею та телекомунікаціями, то можна очікувати на поступову асиміляцію можливостей DSS-систем у існуючі системи ERP-класу. Це, мабуть, призведе до позбавлення процесів оновлень версій ERP-систем.

По суті, вказаний процес уже йде. Так, практично всі поширені ERP системи вже мають функціональні можливості прогнозування з використанням різноманітних статистичних методів. Видається перспективним розвиток DSS-систем в управлінні активами, зокрема, в організації експлуатації та ремонтів обладнання. Компанії великоформатної торгівлі та електронної комерції стали першими замовниками DSS-систем. Завдання, що вирішувалися у цьому секторі свого часу вперше, і нині є актуальними. До таких су-

часних завдань доречно відносити: аналіз асортименту (селективний дохід, оборотність запасів, управління запасами, фондовіддача); розподіл використовуваних площ; аналіз ефективності менеджменту та мотивації; планування та аналіз ефективності реклами.

Ринок DSS-систем у фінансових інститутах зараз найемніший. Сфера застосування DSS-систем у банках стосується насамперед платіжних пластикових карток, аналізу ризиків, запобігання шахрайству; аналізу споживчої поведінки та проектування нових фінансових послуг. Останнє ґрунтується на аналізі та формуванні споживчих груп, які характеризуються подібною поведінкою. Запобігання шахрайству – це перспективна зона використання методів штучного інтелекту, яка ніколи не буде вичерпана. У страхових компаніях DSS-системи ще немає такого широкого поширення, але це передбачає перспективність даного ринку.

У промисловості до сфер застосування DSS-систем можна зарахувати: управління взаємовідносинами з клієнтами; статистичне управління запасами; фінансове та бюджетне планування та управління; аналіз та управління ризиками. У зв'язку з цим характерні зміни при управлінні промисловістю, що сталися останні півстоліття? До 60-х років виробництво розвивалося головним чином за рахунок розвитку технологій, що виражалося тезою: "виробляти і продавати". У той час, безумовно, пропозиція явно формувало попит. При цьому основні виробничі фонди були здебільшого матеріальними: будівлі, споруди, обладнання, за яким стояли патентовані технології. До кінця XX століття визнаною тезою, що виражає раціональну ринкову поведінку, стала парадигма «сприймати і реагувати». Темп появи нових революційних матеріальних технологій сповільнився. А фронт конкурентної боротьби перемістився до галузі проектування нових продуктів і послуг. При цьому превалюючим стали наміри та побажання клієнтів. Як приклади можна навести практично повний перехід на замовлення конфігурування промисловості, постійно зростаючий спектр пропозицій послуг у сфері телекомунікацій при тому самому обладнанні і т.п.

Останнім часом дедалі більшого значення набуває інформація та методи роботи з нею. Це тим більше актуально в розвинених країнах світу на тлі тенденції перенесення безпосередньо матеріального виробництва в країни, що розвиваються, з низькою вартістю робочої сили, енергетичних і сировинних ресурсів. Концепція DSS-систем прямо відповідає задачі інформаційного забезпечення цієї парадигми. Якими є сьогодні основні промислові тенденції? Це і глобалізація, і укрупнення, і спеціалізація, і інтеграція в мережі, і фокусування на розробці нових продуктів і послуг, і необхідність одночасно конкурувати як за якість, так і за ціною.

Інформаційна підтримка реалізації вищезгаданих рекомендацій з боку DSS-систем має такий вигляд: «приділяти ... увагу стратегічному плануванню...» – аналізувати історичні дані щодо структури собівартості, динаміки цін; "вивчати стратегію конкурентів" - аналізувати динаміку ринків; «приділяти увагу виробничим функціям» – аналізувати витрати на управління акти-

вами, динаміку тарифів, ефективність використання устаткування й фондовіддачу; "усунути комунікативні бар'єри" - аналізувати бізнес-процеси; "Визнати цінність розвитку інформаційних зв'язків" - аналізувати взаємини з клієнтами та постачальниками.

Ефективне вирішення даних завдань потребує поглибленого аналізу, як ринкового оточення, і динаміки використання всіх внутрішніх ресурсів. Особливого значення в конкурентній боротьбі за практично рівної ситуації по можливості доступу до технологій набуває персонал та підходи до управління. У розвинених країнах світу персонал, принаймні провідний у стратегічному плануванні, перемістився з категорії «витрати» (cost) до категорії «фонди» – перші треба неухильно скорочувати, а другі треба розвивати та інвестувати. В наш час проявляється глобальна тенденція переважного розвитку ринку послуг у порівнянні зі сферою безпосередньо виробництва, і саме із застосуванням DSS.

У сфері державного будівництва роль DSS-систем поки що невелика. Потенційно їх сфера використання пов'язана з оцінкою ефективності державних програм. Це пов'язано насамперед з тим, що державні програми не зводяться до економічного ефекту як такого. Розвиток DSS-систем у цій сфері великою мірою залежить від участі держави, оскільки основну роль цьому процесі має впровадження критеріїв і підходів до оцінки.

Узагальнений портрет DSS-систем можна скласти з урахуванням короткого аналізу пропозицій компаній Cognos, SAS, Hyperion, Oracle. Перш за все слід звернути увагу на те, що перелік ключових гравців на ринку DSS-систем не збігається з лідируючим списком виробників систем ERP. Присутність компанії Oracle у наведеному списку відображає явно виражений намір компанії Oracle розвивати цей напрямок, наявність дійсно розвиненого інструментального набору для виконання подібних проектів, останні придбання компанії в цій галузі. З цієї точки зору до аналізованого списку можна було б додати і IBM з Microsoft.

До основного функціонального набору DSS-систем сьогодні входять: фінансове планування та бюджетування; формування консолідованої звітності; створення інформаційної системи стратегічного управління на основі ключових показників діяльності (Balance Score cards) із налаштованими бібліотеками показників; аналіз взаємовідносин з клієнтами та постачальниками; аналіз тенденцій; функціонально-вартісний аналіз (ABC-Costing); функціонально-вартісне керування (Activity Based Management, ABM); система постійних покращень; багатовимірний аналіз даних (OLAP); виявлення прихованих закономірностей (Data Mining); виявлення моделей (структур) даних; статистичний аналіз та прогнозування часових рядів; подієве управління бізнесом (Event-driven BI); аналіз ризиків; формування запропонованих запитів; інтелектуальний пошук (за неповними даними та неформальними запитами); бізнес-моделювання та аналіз ефективності виконання бізнес-процесів; референтні галузеві моделі.

Інформаційною платформою DSS є сховища даних (Data Warehouse). Інструментальне середовище – інтеграційні системи, що базуються на відкритих стандартах. Ці системи відповідають вимогам: - інформаційної безпеки; масштабованості; відкритості; багатовимірною та багатоваріантною подання даних; інтелектуального інтерфейсу; інтегрованості з основними платформами та бізнес-додатками, інтеграція даних з різноманітних джерел, мережна інтеграція (насамперед WEB) тощо.

Технічне забезпечення пов'язане з обробкою даних, надійним зберіганням та забезпеченням цілісності, архівованістю (і відновлюваністю), мережевим та телекомунікаційним забезпеченням, криптографічним забезпеченням, керуванням доступом користувачів, експортом-імпортом даних, у тому числі з використанням засобів інтелектуального інтерфейсу (розпізнавання образів: мови, зображень). Особливістю DSS-продуктів є значно більша, ніж у ERP-системах, науковість обробки даних.

Методи застосовуваного DSS-системах статистичного аналізу можуть бути розділені на класи: описової статистики; перевірки статистичних гіпотез; регресійного аналізу; дисперсійного аналізу; аналізу категоріальних даних; багатовимірний аналіз; дискримінантний (кластерний) аналіз; аналізу часових рядів; статистичного планування експериментів, статистичного контролю (зокрема – якості).

Застосовувані в DSS засоби Data Mining (глибинний аналіз даних – процес напівавтоматичного аналізу великих баз даних з метою пошуку корисних фактів) надають можливість ставити і вирішувати як традиційні, так і нетрадиційні завдання аналізу. Наприклад, традиційною є постановка завдання: «визначити, чи є певного виду статистичний зв'язок між такими показниками, як обсяг виробництва товару та обсяг його реалізації». Нетрадиційною була наступна постановка задачі: «є кілька десятків (або навіть сотень) показників діяльності організації, і необхідно визначити, між якими з них слід шукати статистичні зв'язки взагалі, якого роду зв'язку слід шукати (чи вважати показники рівноправними, чи вважати одні показники незалежними), інші залежними змінними), яких об'єктах ці зв'язки проявляються».

Протягом роботи застосунку на етапі вибірки відбувається формування підмножини спостережень з вихідних даних (відбір за критеріями чи випадковий добір). На етапах дослідження та модифікації можуть бути здійснені: фільтрація даних, відкидання даних з великими викидами, перетворення вихідних змінних. На етапі моделювання здійснюється побудова регресій та оптимізація підмножини змінних, прийняття рішень на основі методик нейронних мереж, що реалізують різні алгоритми навчання класифікації об'єктів, побудова класифікаційних дерев для відбору оптимального набору змінних та оптимального розбиття множини об'єктів, кластеризація та оптимальне групування об'єктів. Нарешті, на етапі огляду та оцінки результатів користувач має можливість зіставити різні результати моделювання,

На етапі підготовки даних DSS повинні забезпечувати доступ до будь-яких реляційних баз даних, файлів як текстових, так і інших типів. Додаткові

засоби перетворення та очищення даних повинні дозволяти змінювати вид подання, проводити нормалізацію значень, виявляти невизначені чи відсутні значення. На основі підготовлених даних спеціальні процедури мають дозволяти автоматично будувати різні моделі для подальшого прогнозування, класифікації нових ситуацій, виявлення аналогій. Повинні підтримуватися побудова різних типів моделей (нейронних мереж, класифікаційних та регресійних дерев, байєсівського навчання, кластеризації та ін.).

На практиці у межах систем підтримки прийняття рішень повинні широко застосовуватися засоби інтелектуального аналізу даних (ІАД, Data Mining). Як базовий напрямок розвитку засобів ІАД доречно виділити методи статистичної обробки даних, які можна розділити на чотири типи: попередній аналіз природи статистичних даних (перевірка гіпотез стаціонарності, нормальності, незалежності, однорідності, оцінка виду функції розподілу та її параметрів); виявлення зв'язків та закономірностей (лінійний та нелінійний регресійний аналіз, кореляційний аналіз); багатовимірний статистичний аналіз (лінійний та нелінійний дискримінантний аналіз, кластер-аналіз, компонентний аналіз, факторний аналіз); динамічні моделі та прогнози на основі часових рядів.

У DSS (СППР) прийнятні інтегральні рішення із застосуванням найпопулярніших стандартизованих програмних засобів статистичного аналізу Statistica, SPSS, Systat, Statgraphics, SAS, BMDP, TimeLab, Data-Desk, S-Plus, Scenario (BI), "Мезозавр". Особливий напрямок у спектрі аналітичних засобів ІАД становлять методи, що ґрунтуються на нечітких множинах. Їх застосування дозволяє ранжувати дані щодо ступеня близькості до бажаних результатів.

Другий напрямок розвитку DSS доречно пов'язувати з методами, заснованими на принципах систем, що саморозвиваються (нейронних мереж, еволюційного та генетичного програмування). Однак рішення, отримані такими методами, часто не допускають наочних інтерпретацій, що певною мірою ускладнює змістовний аналіз. До програмних продуктів, що використовують такі методи ІАД, відносяться системи PolyAnalyst, NeuroShell, GeneHunter, BrainMaker, OWL, 4Thought (BI), IBM I2 Analyst's Notebook.

До третього напрямку розвитку ІАД можна віднести традиційні методи вирішення оптимізаційних завдань – варіаційні методи, методи дослідження операцій, що включають різні види математичного програмування (лінійне, нелінійне, дискретне, цілісне, динамічне), імітаційні методи (методи теорії систем масового обслуговування). Програмні реалізації такого виду методів входять до широко поширених стандартних програмних засобів, наприклад - MathCAD, MatLab.

Четвертий напрям розвитку засобів ІАД пов'язаний з експертними методами, тобто пов'язаними з безпосереднім використанням досвіду експертів. До них доцільно відносити метод «найближчого сусіда», який ліг в основу таких програмних продуктів, як Pattern Recognition Workbench або KATE tools. Інший аналогічний підхід до вибору рішення пов'язаний із послідовним логічним висновком дерева рішень, у кожному вузлі якого експерт здійснює

вибір («так» – «ні»). Тут процес статистичного навчання виведено за межі програми та сконцентрований у вигляді деякого апріорного досвіду, укладеного у набір гілок-рішень. Одним із різновидів методу дерев рішень є алгоритм дерев класифікації та регресії, що пропонує набір правил для дихотомічної класифікації сукупності вихідних даних. Цей підхід зазвичай застосовується для передбачення того, які послідовності подій матимуть заданий результат. На основі дерев рішень розроблені такі програмні продукти, як IDIS, C5.0 та SIPINA.

До експертних методів слід віднести і предметно-орієнтовані системи аналізу ситуацій, що ґрунтуються на фіксованих моделях, що відповідають тій чи іншій теоретичній концепції. При цьому роль експерта полягає у виборі найбільш адекватної системи та інтерпретації отриманого алгоритму. Такими прикладами є інформаційні системи у сфері фінансів: Wall Street Money, Meta Stock, Super Charts, Candlestick Forecaster. До експертних методів ІАД, що використовуються в DSS-системах, допустимо відносити і методи візуалізації даних, результатів їх аналізу, що дозволяють наочно відображати отримані висновки для експертів та/або керівників проектів. До програмних продуктів, що дозволяють формувати попередні звіти та візуалізувати результати, можуть бути віднесені системи Mineset та Impromptu.

Таким чином, можна стверджувати, що, з одного боку, у підтримці прийняття управлінських рішень «усі засоби добрі», що забезпечують адекватні реальності прогнози (засновані на них рішення), що узгоджуються надалі з тим, що відбувається на практиці, що допомагає отримувати додаткові відомості про природу досліджуваних явищ, як наслідок – необхідні у прийнятті управлінських рішень через ринкову конкуренцію. З іншого боку, програми формату DSS повинні бути достатньо стандартизовані (як логічно, так і функціонально). Тому що актуально розвивати такі проекти надалі, застосовувати їх з урахуванням відображеного у стандартах досвіду попередніх аналогічних додатків.

Загалом підтримка прийняття рішень – актуальна сфера інформаційних додатків. Цей напрямок інтенсивно розвивається як у формі нескладно інтегрованих відокремлених програмних рішень, так і у формі внутрішньосистемних інформаційних засобів. Забезпечується потреба практично переходити від завдань автоматизації обліку до завдань підтримки прийняття управлінських рішень виходячи з накопичених при автоматизованому обліку даних. Вибір у конкретній організації виду та форми DSS-системи багато в чому визначається реальними зовнішніми та внутрішніми умовами. У рамках цього навчального посібника особлива увага приділяється найбільш прийнятним для широких програм стандартним DSS.

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ЗВ'ЯЗКУ ІНФОРМАЦІЇ, МОДЕЛЮВАННЯ, УПРАВЛІННЯ

1.1. Логічні аспекти поняття «інформаційна система»

Спочатку термін «інформація» (походження від латинського «informatio» – роз'яснення, оповіщення) застосовувався для позначення відомостей, що передаються для людей усно, письмово чи іншими способами. З ХХ ст. інформація – це вже загальнонаукове поняття, що характеризує обмін відомостей: як для людей, і між людьми і «автоматами» (автоматичними перетворювачами інформації), більше, і, власне, між автоматами. Крім того, це відносно нове, але вже широко застосовуване поняття стали застосовувати і до процесів, що давно існують у природі. Так, «інформація» тепер використовується і для характеристики продовжує розвиватися, тепер уже часто під керуванням людини, обміну сигналами у тваринному, рослинному світі, генетичного спадкування тощо.

У цілому нині, поняття «інформація» пов'язані з особливими матеріальними процесами, спеціально які виділяються у природі людиною. Наприклад, усна мова для людей, які говорять і слухають, супроводжується відповідними фізичними явищами, хімічними реакціями, що протікають з часом у певному місці матеріального світу. Зауважимо, що подібно виділяються природні процеси характеризуються людьми як інформаційні лише у сфері людського сприйняття, для людей, відповідно до прийнятих людьми традицій, зокрема – мовних. Тобто поняття «інформація» акумулює і об'єктивні, і суб'єктивні аспекти процесу пізнання, застосування знань у суспільстві.

Нині поняття «інформація» прийнято пов'язувати з інформатикою – наукою про інформаційні (комп'ютерні) технології, пов'язані з автоматизацією людської управлінської діяльності. Щодо інформатики вивчають структуру та загальні властивості інформації, а також питання, пов'язані з її збиранням, зберіганням, пошуком, переробкою, перетворенням, поширенням та використанням у різних сферах. Інтерес до інформаційних додатків у суспільстві зростає. Так, з допомогою відповідних матеріальних процесів вдається передавати корисний досвід. Зокрема – досвід управління в ситуаціях певного виду, що повторюються (при відновлюваних схожих умовах). Як наслідок, проявляється потреба класифікувати загальні типові фрагменти інформаційних процесів (як збирання, зберігання, пошук...).

Таким чином, інформація надається людьми з часом за допомогою матеріальних носіїв. У зв'язку з цим практично широко застосовується і поняття «дані» – відомості, необхідні конкретної мети, якогось висновку, рішення. До передачі даних про об'єкт дані, зазвичай, форматується, кодується (представляється як даних). Причому так, що потім їх вдалося успішно декодувати, сприйняти, використовувати. По-суті, інформацію люди (через свої кінцеві

психофізіологічні можливості, через обмеженість часу сприйняття) зазвичай передають як дані кінцевого доступу до огляду. І дані передбачають формування, кодування відповідно до певної мети та умов збору, зберігання, планованого використання. В підсумку, передача інформації (у випадку конкретного реального об'єкта – потенційно нескінченної через необмеженості пізнання у часі будь-якого матеріального об'єкта людьми загалом) у зоні сприйняття людей передбачає оперування кінцевими даними (внаслідок наявності обмежень по сприйняттю конкретною людиною). У цілому нині, на практиці потрібні обидва поняття: «інформація» більшою мірою характеризує розгорнуту у часі необмежену за змістом методику пізнання, тоді як «дані» представляють цю методику «замороженою» – у час, у кінцевому обсязі – під конкретну мету.

Управління – функція, що забезпечує збереження структури, підтримка умов функціонування, досягнення цілей.

Досліднику (що займається вдосконаленням управління, практикуючому спеціалісту) реальний об'єкт доступний у певний час за фіксованих умов. При цьому прийнятно отримання потенційно сприймається нескінченної інформації, але завжди у вигляді відповідних кінцевих даних, згідно з еталонами вимірювань (порівнянь з типовими аналогами). Пізнаність забезпечується тим, що дані доречно поповнювати з часом, зменшуючи зону інформаційної невизначеності, невідомості. Оперування поняттями «інформація», «дані» пов'язані з особливого виду матеріальними процесами, є принципово важливими цілеспрямованого вдосконалення будь-яких керованих матеріальних процесів, поновлюваних з часом у схожих умовах. Такими подіями насичене життя людей у природі. Загалом за допомогою інформації (даних) забезпечується систематизація людського досвіду. А саме – забезпечується накопичення, передача досвіду можливого керування. Інформація у всьому її різноманітті характеризує суб'єктивне сприйняття властивостей матеріальних об'єктів, що відображаються. Так, при відображенні інформації (даних) на матеріальних носіях багато в подальшому сприйнятті визначає спосіб кодування. Причому, відповідно до цілей, повинні бути узгоджені способи кодування до передачі та декодування після прийому даних. Узгоджено – з існуючими (використовуваними) еталонами за кількісними показниками, з мовним синтаксисом за текстами, з відповідними стандартами, традиціями за іншими форматами даних, що використовуються та зберігаються. Передача досвіду можливого керування. Інформація у всьому її різноманітті характеризує суб'єктивне сприйняття властивостей матеріальних об'єктів, що відображаються. Так, при відображенні інформації (даних) на матеріальних носіях багато в подальшому сприйнятті визначає спосіб кодування. Причому, відповідно до цілей, повинні бути узгоджені способи кодування до передачі та декодування після прийому даних. Узгоджено – з існуючими (використовуваними) еталонами за кількісними показниками, з мовним синтаксисом за текстами, з від-

повідними стандартами, традиціями за іншими форматами даних, що використовуються та зберігаються. передача досвіду можливого керування. Інформація у всьому її різноманітті характеризує суб'єктивне сприйняття властивостей матеріальних об'єктів, що відображаються. Так, при відображенні інформації (даних) на матеріальних носіях багато в подальшому сприйнятті визначає спосіб кодування. Причому, відповідно до цілей, повинні бути узгоджені способи кодування до передачі та декодування після прийому даних. Узгоджено – з існуючими (використовуваними) еталонами за кількісними показниками, з мовним синтаксисом за текстами, з відповідними стандартами, традиціями за іншими форматами даних, що використовуються та зберігаються. при відображенні інформації (даних) на матеріальних носіях багато в подальшому сприйнятті визначає спосіб кодування. Причому, відповідно до цілей, повинні бути узгоджені способи кодування до передачі та декодування після прийому даних. Узгоджено – з існуючими (використовуваними) еталонами за кількісними показниками, з мовним синтаксисом за текстами, з відповідними стандартами, традиціями за іншими форматами даних, що використовуються та зберігаються. при відображенні інформації (даних) на матеріальних носіях багато в подальшому сприйнятті визначає спосіб кодування. Причому, відповідно до цілей, повинні бути узгоджені способи кодування до передачі та декодування після прийому даних. Узгоджено – з існуючими (використовуваними) еталонами за кількісними показниками, з мовним синтаксисом за текстами, з відповідними стандартами, традиціями за іншими форматами даних, що використовуються та зберігаються.

При використанні інформації (даних) в організації, вдосконалення управління природним чином виявився повсюдно застосовуваним комплексний підхід, що передбачає оперування поняттям «система» (від грецької – ціле, складене з елементів, з'єднання). Система – безліч елементів, що у відносинах і зв'язках друг з одним, утворюють певну цілісність, єдність. Термін «система», системний підхід в інформатиці виявився широко затребуваним внаслідок мінливості реального світу, через необмежену пізнаваність, через потребу безперервного поглиблення пізнання з метою вдосконалення управління на практиці (за наявності «важелів управління» та потенційно поновлюваних умов).

Система відрізняється від безлічі елементів, що об'єднуються тим, що їй притаманні «системоутворюючі» властивості, що проявляються лише для взятих у сукупності елементів (відсутні у елементів, обраних окремо). Наприклад, загальні більшість систем логічні властивості ієрархічність, керованість виявилися задіяними під час конструювання, вдосконаленні багатьох конкретних програмно-реалізованих практично інформаційних систем, призначених для автоматизації обліку та управління.

Особливий інтерес у зв'язку з комплексним підходом до використання інформації (даних) проявляється стосовно технологій як способів виробництва, розподілу товарів, послуг. Проектування поняття технологія на інформаційні

процеси призвело до виникнення так званих «інформаційних технологій», зокрема, як сукупності знань про способи використання інформації (даних) для вдосконалення управління виробництвом за допомогою автоматизації. А надалі – стосовно автоматизації прийняття управлінських рішень взагалі. У сучасних умовах «інформаційна технологія» – це і методи, і, можливо, виробничі процеси, та програмно-технічні засоби, що визначають процес використання даних для зниження трудомісткості, підвищення надійності, оперативності під час продукування нових управлінських рішень.

З одного боку, інформаційна технологія – сукупність методів, виробничих, програмно-технологічних засобів, об'єднаних у ланцюжок для збирання, зберігання, обробки, розповсюдження даних для певних цілей. З іншого боку – це комплекс заходів, засобів, які забезпечують збирання, зберігання, обробку, передачу, захист, відображення інформації (даних) взагалі.

Вважають, що до XIX ст. продуктивність інформаційно-технологічної обробки була низькою (атрибутами «паперових» технологій були перо, чорнило, папір...). До XIX ст., вже застосовувалися «механіко-технологічні» засоби: друкарські машинки, телефони... Початок XX ст. прийнято пов'язувати із виникненням «електричних» інформаційних технологій. У середині XX ст. виникли «ЕОМ», як наслідок, повсюдно впроваджувалися нові технології – «комп'ютерні». З 80-х інтенсивно розвиваються і поширюються персональні комп'ютери, мережі. І ефективно знання спеціалістів доповнюються програмним забезпеченням у формі профільованих інформаційних систем (ІС), які представляють, по суті, мережні комп'ютерні технології.

В інформатиці прийнято виділяти аналогову та цифрову інформацію. Людина, використовуючи свої органи чуття, має справу з приблизними «аналогами», а обчислювальна техніка оперує точними цифрами. Аналогову інформацію вдається зіставляти з еталонами і «оцифровувати». Немає двох однакових матеріальних об'єктів – їх зіставлення з зразками виявляє інформацію аналогову. Якщо кольорам, сприйнятим по-різному більшістю людей, зіставити різні номери, а різним звукам – ноти, аналогову інформацію можна подати цифровий. Так, "жива" музика неповторна, їй відповідає нескінченна інформація, але вона співвідносна з "грубими" аналогами у формі еталонів-нот. Ноти забезпечують якісне відтворення звуків (невідмінне для більшості людей від оригінального звучання). Історія інформатики – це історія кодування, подання даних: передачі та зберігання за допомогою різних технічних пристроїв.

У зв'язку з використанням інформаційних технологій виникло поняття "інформаційна система" (ІС). Причому це сталося природним чином до появи комп'ютерів. Змістовна інтерпретація ІС – це комплексний підхід до застосування інформації на практиці, що забезпечує підвищення продуктивності праці та зниження ймовірності помилок у прийнятті управлінських рішень. Конкретна ІС, по суті, забезпечує реалізацію інформаційних технологій у

певній предметній (професійній) сфері. Хоча логічне поняття ІС з'явилося ще до комп'ютерів, відповідні методики були настільки поширеними, як, наприклад, нині. Справа в тому, що швидкість обробки даних, привнесена технічними обчислювальними пристроями, сприяла поширенню логічного поняття «ІС».

Отже, до появи комп'ютерної техніки, мережевих технологій досягнення суттєвих переваг за допомогою ІС було ускладнено через недостатню швидкість збору, обробки даних. У наш час інформатика є, по суті, наукою про мережеві комплексні (системні) комп'ютерні технології нового рівня. Тому доцільно за сучасних умов термін ІС доповнювати характеристикою: апаратно-програмне забезпечення зберігання, обробки даних, що забезпечує якісно нового рівня результати управлінських додатків. За змістом ІС - це і "дані", і "алгоритми їх обробки" у формі систем управління базами даних ("СУБД"), причому зі зростанням наукомістких додатків математичних моделей та методів. Тобто, на практиці наростає інтерес до підтримки прийняття управлінських рішень як такої, все це внаслідок природного перебігу історичних подій. Причому, за принципом: «від простого – до складнішого (до продуктивнішого)».

Отже, в даний час ІС - це система, елементами якої є і дані, що перетворюються в процесі її функціонування, і методики обробки, систематизації даних, додатки їх на практиці для вдосконалення управління в тій чи іншій предметній сфері. Наприклад, планування та управління організацією – це ІС. Загалом управління економікою країни – теж ІС, але більших розмірів. ІС взагалі – це система збору, зберігання, накопичення, пошуку та передачі даних, що застосовуються у процесі управління, планування. Дедалі частіше у наші дні, характеризуючи сучасне програмне забезпечення, переходять логічно і змістовно від поняття «СУБД» до поняття «ІС». Це виправдано і історично, і змістовно. ІС виступає засобом інтегрованої обробки даних сучасної корпорації, фірми.

Популярність на практиці ІС, призначених для автоматизації обліку та управління, в наш час багато в чому визначається наявністю високопродуктивних типових автоматизованих засобів реєстрації даних та отримання звітів, насамперед – за підсумками та за оборотами. Причому - за підсумками, оборотами в розрізі різноманітних облікових статей (часто в сукупності - багатовимірно, ієрархічно, пов'язано), за багатьма ресурсними показниками (кількісним, сумовим, ваговим і т.п.). Усе це характеризує сучасний автоматизований облік у створенні їх як кількісний, сумовий, багатовимірний (ієрархічний) аналітичний. І на підставі всього перерахованого вище – як управлінський облік. На практиці потенційно широко застосовуються призначені для автоматизації обліку та управління ІС, що забезпечують добре апробовані прийоми обліку та відповідного управління, що реалізується в результаті детально обізнаними про стан справ в організації експертами. Причини користувачьких переваг на користь зазначеного виду ІС - очевидні. Справа в тому, що навіть проста систематизація накопичуваних даних забезпечує отримання

переваг у порівнянні з конкурентами, дозволяє приймати більш обґрунтовані управлінські рішення в конкретній організації за інших постійних умов.

Контрольні питання до підрозділу 1.1

- 1. Якою є загальна характеристика поняття «інформація»?*
- 2. Які питання безпосередньо пов'язані з інформацією внаслідок розвитку комп'ютерних технологій?*
- 3. Як пов'язані поняття «інформація», «дані»?*
- 4. Яким чином загалом поняття «інформація» узгоджено з процесами накопичення та систематизації досвіду управління?*
- 5. Чим характеризується поняття «система», чому системний підхід є актуальним в інформатиці?*
- 6. У чому є сутність поняття «інформаційна технологія»?*
- 7. Як прийнято характеризувати інформаційні технології різних історичних етапів межі XIX-XX в.?*
- 8. Якою є змістовна інтерпретація поняття «інформаційна система» (ІС)?*
- 9. Як трансформувалася характеристика ІС у зв'язку з розвитком комп'ютерних технологій останнім часом?*
- 10. Чим, передусім, нині характеризуються ІС, призначені для автоматизації обліку та управління?*

1.2. Інформаційні системи автоматизації обліку та управління

Пов'язані з прогнозуванням, паралельним пошуком оптимальних рішень громіздкі обчислювальні розрахунки все ширше реалізуються внаслідок вдосконалення обчислювальної техніки, через розвиток мережових комп'ютерних технологій. Тим самим, загалом, повсюдно з'являється можливість оперативно відповідно до даних попередніх періодів виконувати багатоваріантну обробку, застосовувати відповідні результати управління. І вільна ринкова конкуренція диктує необхідність додатків зазначеного виду адекватного реальним умовам економіко-математичного моделювання.

ІС, призначена для автоматизації обліку та управління, забезпечує можливість дискретно у часі зберігати дані про реальні явища. Чисельними методами оброблювані дані, відповідно до адекватної реальності інформаційним моделям, описують очікувану поведінку необхідних показників. Це дозволяє обчислювати оптимальні значення керованих факторів відповідно до прогнозів, що виявляються (з використанням чисельно продукованих багатоваріантних розрахунків) і відстежувати з часом відхилення прогнозів від фактичних практичних результатів. Пізніше допустимо коригувати інформаційні моделі відповідно до зазначених відмінностей.

Сучасні ІС прийнято класифікувати. І, насамперед, за предметними областями додатків внаслідок професійної стандартизації. Наприклад, окремо виділяють ІС для автоматизації обліку, управління. Їх застосування забезпечують суттєві переваги перед конкурентами при прийнятті управлінських рішень стосовно соціально-економічних об'єктів. Зауважимо, що зазначена класифікація є ієрархічною. Так, при автоматизації обліку, управління виділяють і вузкі області. Зокрема – бухгалтерію, оперативний облік, кадри.

Класифікація ІС корисна – вона звужує зону пошуку, зону вибору професійного програмного забезпечення. Характерний рельєфний «класичний» приклад ІС у формі так званих «САПР», які забезпечують при здійсненні архітектурного проектування суттєве підвищення продуктивності праці в порівнянні з аналогічними «паперовими технологіями», що передували. Взагалі, спеціалізоване (зокрема, згідно з предметно характеризуються «класами») програмне забезпечення, що інтерпретується як ІС, широко застосовується на практиці. Таке програмне забезпечення зарекомендувало себе з позитивного боку, як наслідок – набуває подальшого поширення.

Використання ІС пов'язане з іншого виду класифікацією – згідно з цілями додатків. Так, прийнято виділяти потенційно паралельно вживані на конкретному підприємстві (у певній предметній сфері) «класи»:

- автоматизовані системи навчання (АСН);
- автоматизовані інформаційно-довідкові системи (АІДС);
- Системи підтримки прийняття рішень (СППР).

АСН передусім передбачають професійне стандартизоване навчання. АІДС прийнято застосовувати при формуванні архівів для забезпечення великомасштабного пошуку (відборів). Вузько профільовані додатки у вигляді СППР призначені для управлінців певного профілю та характеризуються наявністю можливості реалізовувати відносно високий рівень формалізації вироблення рекомендацій щодо прийняття управлінських рішень (і, відповідно, знижується роль «людського фактору»).

Популярність на практиці ІС, призначених для автоматизації обліку та управління, багато в чому визначається наявністю високопродуктивних типів автоматизованих засобів реєстрації даних та отримання звітів за підсумками, оборотами. Причому - за підсумками, оборотами в розрізі різноманітних облікових статей, за багатьма ресурсними показниками (кількісними, сумовими, ваговими і т.п.). Усе це характеризує сучасний автоматизований облік у створенні як кількісний, сумовий, багатовимірний (ієрархічний) аналітичний. І, виходячи з усього перерахованого вище, як – управлінський облік.

Таким чином, на практиці потенційно застосовні призначені для автоматизації обліку та управління ІС, що забезпечують апробовані прийоми обліку та відповідного управління, що реалізується в результаті добре обізнаними про стан справ в організації експертами. Причини користувацьких переваг на користь зазначеного виду підходу очевидні. Справа в тому, що навіть проста

систематизація накопичуваних даних забезпечує отримання переваг у порівнянні з конкурентами, дозволяє приймати більш обґрунтовані управлінські рішення в організації за інших незмінних умов.

За даними попередніх періодів допустимо як стандартними (щодо простими) методами групувати підсумки, обороти у багатьох розрізах аналітичних статей, за багатьма ресурсним показниками, а й додатково отримувати адекватно даним попередніх періодів прогнози, засновані на економіко-математичному моделюванні. Ті самі підсумки та обороти – за своєю сутністю засновані на найпростіших математичних моделях. З використанням такого роду і більш складного моделювання допустимо математичними методами виявляти і оптимальні управлінські рекомендації згідно з прогнозами, що продукуються (і все це практично реалізується засобами внутрішньосистемних алгоритмічних мов програмування).

Удосконалення інформаційного забезпечення підвищення ефективності бізнесу стає дедалі актуальним практично. Економічне моделювання, стандартизоване за носіями та методами обробки даних, що включає як інформаційні, так і математичні аспекти, стає типовим засобом підвищення конкурентоспроможності. На більшості підприємств формуються бази даних, що забезпечують детальний формальний аналіз. Отже, вже заснована на застосуванні аналізу залишків, оборотів, оперативна обробка даних дозволяє поглиблювати експертні знання про природу економічних явищ. І сучасну ІС дедалі частіше асоціюють із поняттям «соціально-економічна модель», яка передбачає спільне використання понять «предметна модель», «інформаційна модель» (формати, періодичність даних, що зберігаються, і способи обробки, систематизації цих даних), «економіко-математична модель».

Як типові, так і індивідуальні (можливо, ексклюзивні) завдання автоматизації підтримки управлінських рішень відносно швидко (з низькими трудовитратами), що реалізуються в ІС обліку та управління за допомогою стандартизованих підсистем зберігання та обробки даних, типових засобів налаштування-конфігурування. При цьому допустимо використовувати внутрішньосистемну алгоритмічну мову, що забезпечує гнучкість, застосування якого доступне в модулях типових підсистем (зокрема, для забезпечення використання економіко-математичних моделей і методів). Крім того, технічні нововведення багато що визначають на практиці - так останнім часом повсюдно впроваджуються "Internet" ("WEB")-технології доступу до даних та методів інформаційних баз).

Доречно виділяти притаманні всім ІС інформаційні стандарти (які виявляються при заповненні полів, при роботі з вікнами тощо) та предметні, що відображають традиції професійної галузі додатків. Стандартизація багато в чому пов'язана з процесом впровадження: загальні складові ІС визначають так звану програмну частину («платформу»), вона проявляє себе однаково на всіх організаціях. Специфіка предметної області даних відокремлена у формі

так званої інформаційної бази (ІБ). ІБ є, по суті, сукупністю конфігурації та даних, доступних в режимі користувача відповідно до типової (або специфічної) конфігурації.

Конфігурованість ІБ забезпечує реалізацію великого спектра предметних специфічних завдань автоматизації обліку та управління. Однакова конфігурація різних ІБ забезпечує схожі умови користувальницької експлуатації. Дані, що реєструються в ІБ один раз, можуть бути потрібні багаторазово. Різнопрофільні ІБ забезпечують створення єдиного інформаційного простору для підприємства, оскільки конфігурацію конкретної ІБ доречно розглядати як інформаційну модель певної предметної області, сфери обліку, управління. І при цьому реалізовані зв'язки між ІБ. В цілому, ІС автоматизації обліку та управління доречно розглядати як сучасне інструментальне середовище для ефективної реалізації різних соціально-економічних проєктів, що вдосконалюються з часом. Причому,

Конфігурованістю ІБ забезпечується і тиражування типових підсистем відповідно до вже накопиченим практичним досвідом автоматизації обліку, управління. Так, забезпечується можливість адаптуватися до особливостей обліку на будь-якому конкретному підприємстві, безперервно вдосконалювати облік, управління в процесі експлуатації відповідно до існуючої (з часом змінюється) специфіки. І все частіше на практиці програмне забезпечення, що є СУБД за стилем зберігання даних, характеризують як ІС – згідно з сучасними способами оновлення сутнісних, форматних аспектів цілям, що адекватно змінюються з часом, і способам обробки даних, використання єдиного інформаційного простору.

Розширення додатків програмного забезпечення у вигляді ІС, призначених для автоматизації обліку та управління, обумовлено наявністю стандартів структури, форматів зберігання даних, використання шаблонних методів обробки даних. Основна мета використання стандартної обробки в сучасній економіці - це оперативний аналіз поточного стану справ, що швидко налаштовується, що й забезпечують типові реєстратори облікових подій і звіти про підсумки, обороти (щодо нескладно додатково настроюються конфігуруються за наявності відповідних знань).

Контрольні питання до підрозділу 1.2

- 1. Яким чином ІС обліку та управління дозволяє отримувати адекватні реальності рекомендації щодо прийняття управлінських рішень?*
- 2. Чому і як програмне забезпечення формату ІС обліку та управління забезпечує конструктивне економічне моделювання?*
- 3. Як прийнято класифікувати ІС в областях додатків?*
- 4. Яким чином групують ІС відповідно до цілей досліджень?*
- 5. Чим, насамперед, нині визначається популярність ІС, призначених для автоматизації обліку та управління?*

6. У чому причини переваг стандартизованого ПЗ у формі ІС, призначених для автоматизації обліку та управління?

7. Які загальні умови додатків економіко-математичного моделювання в ІС автоматизації обліку та управління?

8. Які типові засоби забезпечують підвищення ефективності бізнесу в рамках сучасних ІС обліку та управління?

9. Які форми найбільш загальних впроваджувальних традицій-стандартів ІС обліку та управління?

10. Як ІС, призначена для автоматизації обліку та управління, забезпечує ефективну реалізацію різних проектів?

1.3. Інформатика, моделювання та підтримка управлінських рішень

Вдосконалення інформаційного забезпечення для підвищення ефективності бізнесу – це все більш актуальне завдання для будь-якої конкретної організації. Економічне моделювання, стандартизація за носіями та методами обробки даних, повсюдно виступають типовим засобом підвищення конкурентоспроможності. На більшості підприємств формуються бази даних, що забезпечують детальний формальний аналіз. Засноване на відстеженні залишків, оборотів оперативна обробка даних дозволяє поглиблювати експертні знання про природу соціально-економічних явищ. І сучасне програмне забезпечення корпорацій, фірм формату ІС обліку та управління дедалі частіше асоціюють із поняттям «економічна модель». У цьому логічно передбачається спільне використання понять «предметна модель» (економічна), «інформаційна модель» (формати,

Економічне моделювання передбачає використання інформаційних технологій автоматизації обліку та підтримки прийняття управлінських рішень. Автоматизований облік, перш за все, застосовують там, де його відсутність є неприпустимим через складні умови. Наприклад, відповідно до чинного законодавства. Або якщо відсутність обліку свідомо призводить до низьких фінансових результатів (зокрема через присутність конкуруючих підприємств, які займаються тими ж або подібними видами бізнесу із застосуванням нових інформаційних комп'ютерних технологій). У другому випадку має пройти час, доки стануть відчутними переваги за рахунок застосування досконаліших методів. І другий випадок здійснюється поступове витіснення з ринку конкурентів, управляючих ресурсами менш ефективно.

Основна сутність комп'ютеризованого обліку: організація електронного відстеження однотипних облікових подій, що циклічно повторюються (обов'язково за кількісними показниками, хоча б по одному). Наприклад, за кількістю об'єктів, що враховуються, і сумі (добутку ціни та кількості – окремо або додатково). Можливо, у розрізі статей обліку. Зокрема, щодо об'єктів обліку (номенклатури чи контрагентів тощо). Допустимо, додатково – за іншими показниками, наприклад, місцями обліку тощо. Електронний облік

повинен забезпечувати можливість у місцях відстеження матеріальних, фінансових потоків для кожної елементарної події фіксувати зміни кількісних показників (із зазначенням відповідних облікових статей, якщо такі є). Комп'ютеризована система обліку повинна забезпечувати відповідні дії, що враховується, підсумовування, віднімання (згідно з приходом або витратою). Іноді потрібно відстежувати оборот (додатково чи окремо). При цьому повинна забезпечуватися можливість отримувати звіти: за підсумками залишків кількісних показників (у розрізі аналітики) на поточний або довільний момент; по оборотах задаються інтервалів відстеження динаміки процесів, що враховуються.

Наявність хоча б одного циклічно відстежуваного кількісного показника – обов'язковий атрибут електронного обліку. Такий облік є знеособленим. Інші кількісні показники статті обліку є, по суті, додатковими атрибутами. Сутність аналітичного обліку характеризується обов'язковою наявністю хоча б однієї облікової статті на додаток до хоча б одного кількісного показника. Вибір практично тих чи інших облікових статей, відповідних кількісних показників визначається специфікою конкретної предметної області (наприклад, певного організації, обраних видів діяльності тощо.) і здійснюється, зазвичай, «апріорі» – до початку обліку. Додавання (зміна) атрибутів електронного обліку у процесі експлуатації допустиме, але потребує регламентованих облікових операцій, зокрема – інвентаризації.

Поняття "модель" є найважливішою категорією теорії пізнання. На ідеї моделювання засновані як теоретичні методи, пов'язані з використанням абстрактних моделей, так і експериментальні, що передбачають практичні застосування. Дослідити процеси, явища допомагає адекватний цілі символічний (інформаційний) аналог-модель, що дозволяє, зокрема, обґрунтовано підбирати значення керованих характеристик для прообразу. По суті, «моделювання» - це дослідження якого-небудь класу явищ шляхом побудови, вивчення їх аналогів-моделей. Актуально застосовувати математичне моделювання для вдосконалення управління із застосуванням комп'ютерної техніки (у зв'язку з автоматизацією обліку, управління, що повсюдно поширюється).

Предметна модель відтворює певний фрагмент природної чи соціальної реальності і використовується, передусім, розширення знань про оригіналі, застосованих у конкретній практичній сфері. Характеризована потрібними властивостями предметна модель часто застосовуються заміни оригіналу при обчислювальних експериментах. У будь-якому випадку модель передбачає абстрагування та ідеалізацію. Зокрема, відповідно до мети досліджень відтворюють лише ключові властивості прообразу моделі. Модельний аналог об'єкта, який відтворюється із застосуванням комп'ютерних технологій, дозволяє будувати прогнози, підбирати оптимальні кількісні значення керованих характеристик. При відстеженні якості моделювання з часом допустимо переходити від простих моделей, що менш розкривають сутність, до більш адекватних реальної модельним комплексам, що є, як правило, більш складними.

Побудові моделі, підбору необхідних здійснення моделювання даних передує виявлення мети досліджень, у межах яких модель передбачається застосовувати. І модель відтворює фрагменти реальності, що об'єднуються та аналізуються вглиб відповідно до мети та об'єктивних умов. При організації управління моделювання часто залучається тому, що вдається чисельно відтворити функціональні властивості прообразу, що з часом змінюється. І це дозволяє здійснювати вибір найбільш прийнятних значень керованих характеристик, оперативно реагуючи на наслідки, що відстежуються (очікувані).

В даний час моделюванням допустимо займатися, не вникаючи в сутність процесу пошуку рішень, наприклад, при застосуванні спеціалізованого програмного забезпечення (предметно орієнтованих ІС). У такій ситуації цілі математичного моделювання досягаються опосередковано: заносяться вхідні дані, значення параметрів, що визначають характеристики моделі, ініціюються обчислення, аналізуються, використовуються на практиці результати моделювання.

Математична модель – наближений опис будь-якого класу явищ зовнішнього світу, виражений за допомогою математичної символіки... Останнім часом, з урахуванням розвитку обчислювальної техніки, комп'ютерних технологій, доречно додавати: або із застосуванням спеціалізованого ПЗ. Внаслідок розвитку комп'ютерних технологій з'являються додаткові можливості реалізації моделювання. У результаті – шириться сфера додатків та економічного моделювання.

Прикладне математичне моделювання є циклічно поновлюваним процесом. Його традиційно виділені етапи такі:

- 1) змістовний аналіз об'єкта моделювання. Розробка моделі із застосуванням математичної символіки (спеціалізованого програмного забезпечення);
- 2) вирішення прямої задачі – пошук рішення за побудованою моделлю;
- 3) вирішення зворотного завдання – перевірка адекватності прогнозів результатам застосування модельного рішення на практиці;
- 4) коригування моделі (програмного забезпечення) на підставі п.3 та перехід до п.2.

Термін «моделювання» використовують у економіці по-різному. Так, при налаштуванні ІС обліку та управління оперують поняттям, інформаційна модель, що визначає, перш за все, формати зберігання даних. Інформаційне моделювання, як правило, доповнюється складними розрахунками, що характеризується математичним моделюванням (хоча б у віддаленій перспективі). «Економічна модель», з одного боку, об'єднує зазначені два поняття, з іншого – відбиває і предметну сутність управлінських проблем, що вирішуються.

ІС, призначена для автоматизації обліку та управління, забезпечує можливість дискретно у часі зберігати дані про реальні явища, відтворювати відповідні реальним образам, аналогам предметні моделі. Чисельними методами оброблювані дані, відповідно до адекватних реальності моделям, описують

очікувану поведінку різних показників. Це дозволяє обчислювати оптимальні значення керованих факторів відповідно до прогнозів, що виявляються (на підставі аналітичних або чисельнопродовжених багатоваріантних розрахунків), і відстежувати з часом відповідність прогнозів фактичним практичним результатам. Потім з'являється можливість коригувати моделі згідно з зазначеними відмінностями.

Періодично допустимо вдосконалювати предметні інформаційні моделі відповідно до відповідності, що адекватно відстежується з часом. Відповідно до аналізу відповідності між реальними даними та попередніми модельними прогнозами у погоджених із предметним середовищем формах. Необхідність вдосконалення обліку, управління економіки багато в чому пояснюється стохастичної природою ринкового середовища: особам, які приймають рішення, потрібно періодично оперативно приймати рішення щодо вибору конкретних керуючих впливів ринку (з управління ресурсами, цінами тощо.). Часто виникає потреба вирішувати аналітичні завдання великої розмірності. Конкуренція змушує займатися вдосконаленням обліку та управління. При цьому доцільно застосовувати моделювання, що забезпечує поглиблений кількісний аналіз даних, що накопичуються.

Синтез понять «економічне моделювання» та «інформатика» – наказ часу. Доцільно починати їх спільне освоєння на підприємствах з розгляду нескладних моделей, із застосування програмного забезпечення, що неважко освоюється, для виявлення модельних рішень. В цілому, далі, актуальна доступна для реалізації (потенційно розвивається) методика економічного моделювання та управління із застосуванням і складних детермінованих модельних блоків, і економетричних (відображають випадкову природу взаємозв'язків між показниками) підсистем, і імітаційних підсистем, що детально описують стохастичну природу соціально-економічних у динаміці чинника «час».

Контрольні питання до підрозділу 1.3

- 1. У чому основна сутність електронного обліку?*
- 2. Якими є обов'язкові, додаткові атрибути обліку?*
- 3. Які завдання загалом допомагає вирішувати «моделювання»?*
- 4. Чим характеризуються предметні моделі?*
- 5. Яке значення мають на меті та умови моделювання?*
- 6. У чому переваги комп'ютерного інформаційного (чисельного) моделювання?*
- 7. Який сенс вкладається у поняття «математична модель»?*
- 8. Які основні етапи математичного моделювання?*
- 9. Як інтерпретується поняття "економічна модель"?*
- 10. У чому причини зростання інтересу до моделювання? Якою є ефективна схема освоєння економіко-математичного моделювання?*

РОЗДІЛ 2. ІНФОРМАЦІЙНА ПІДТРИМКА ПРИНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ

2.1. Змістова сутність підтримки прийняття рішень

Діяльність ділових людей пов'язана з необхідністю щодня ухвалювати рішення різної складності. Прикладами можуть служити: вибір напряму розвитку фірми, вибір варіанта автоматизації діяльності компанії, визначення варіанту розміщення філії, типу товару, що випускається або закуповується, вибір приміщення під офіс, типу обладнання, вибір кредитора, співвиконавця роботи, призначення на вакантну посаду одного з багатьох кандидатів і т.п. (див., наприклад, [8]).

Обґрунтованість та професійний рівень прийнятих рішень визначає, зрештою, ефективність діяльності фірми. Необхідність обліку при ухваленні управлінських рішень великої кількості політичних, економічних, соціальних, юридичних та моральних факторів значно ускладнює завдання вибору правильного варіанта вирішення. Насамперед це пов'язано з необхідністю збору необхідної для прийняття рішення інформації. У цьому плані істотну допомогу керівнику надають сучасні інформаційні системи. Однак володіння необхідною інформацією – необхідна, але недостатня умова для прийняття правильного рішення. Справді, проведемо таку аналогію. Уявіть собі, що ви маєте всі аналізи, рентгенограми, кардіограми, томограми і так далі, за якими можна поставити правильний діагноз і прийняти правильне рішення про методи та засоби лікування. Однак для цього потрібно мати знання лікаря, причому, як правило, у складних випадках залучати лікарів кількох спеціальностей.

При прийнятті справді складних рішень необхідно залучати експертів – фахівців у різних галузях знань. Не може одна, навіть найталановитіша, людина бути справді великим професіоналом у багатьох галузях знань. Однак, щоб ефективно використовувати знання експертів, потрібно, по-перше, знати, які експерти вам необхідні, по-друге, які питання ставити перед ними, і, нарешті, як використовувати їх знання для ухвалення рішення. Помилка доручатиме експерту приймати рішення. Маючи знання експертів, приймає рішення керівник, який поставив завдання.

Припустимо далі, що якось вам вдалося зібрати потрібних експертів та витягти їх потрібні для обґрунтування рішення знання. Проте експерти показали, що існує безліч взаємозалежних факторів, що впливають на ефективність вирішення. Кожен із експертів знає, як пов'язані чинники, що стосуються його компетенції, але й тільки. А для ухвалення рішення потрібно врахувати взаємозв'язок усіх факторів, що належать до компетенції різних експертів. Таких зв'язків може бути десятки, а складних проблем (наприклад, визначення стратегічного напрямку розвитку фірми) і більше. У той самий час керівник, який приймає рішення, - звичайна людина з притаманними всім людям

психофізіологічними обмеженнями, які, зокрема, у цьому, що людина може оперувати одночасно трохи більше, ніж поруч об'єктів. Де ж шляхи вирішення наявного протиріччя? Воно може бути дозволено лише з урахуванням застосування математичних методів, втілених у сучасних інформаційних системах підтримки прийняття рішень (СППР). Досвід людини, багатьох може бути акумульований у формі накопиченого тисячоліттями досвіду логіки, накопиченого досвіду найближчих попередніх років у формі додатків конкретного виду моделей до однотипних ситуацій. Нарешті, досвід відображається у вигляді порівнянних з еталонами даних найближчих попередніх періодів для конкретного підприємства, у конкретних реальних умовах, і такий досвід може бути потрібний поряд із зазначеним методичним досвідом у вигляді моделей та методів обробки наявних даних. втілених у сучасних інформаційних системах підтримки прийняття рішень (СППР). Досвід людини, багатьох може бути акумульований у формі накопиченого тисячоліттями досвіду логіки, накопиченого досвіду найближчих попередніх років у формі додатків конкретного виду моделей до однотипних ситуацій. Нарешті, досвід відображається у вигляді порівнянних з еталонами даних найближчих попередніх періодів для конкретного підприємства, у конкретних реальних умовах, і такий досвід може бути потрібний поряд із зазначеним методичним досвідом у вигляді моделей та методів обробки наявних даних.

Ці системи починають все ширше застосовуватися державними організаціями та великими корпораціями (US Navy, NASA, IBM, General Motors, Xerox, 3M, Rockwell International, Reiter Consulting Group International та ін.) утворення США на період 1985-2000 років; вибір методів завоювання ринку побутової техніки; оцінка привабливості у найближчі 10 років регіонів США для працевлаштування людей, які закінчили коледж; розподіл коштів між заходами, спрямованими зменшення бандитизму; оцінка перспективності видів альтернативного пального для автомобілів; розподіл коштів між проектами соціальної програми гуманітарного спрямування; відбір науково-технічних проектів у рамках конкурсу; вибір перспективних напрямів інформатизації країни;

Головним завданням, яке доводиться вирішувати при ухваленні рішення, є вибір альтернативи, найкращої для досягнення певної мети, або ранжування безлічі можливих альтернатив за ступенем їхнього впливу на досягнення цієї

мети. В даний час розроблено велику кількість методів надання допомоги особі, яка приймає рішення (ОПР) при здійсненні цього завдання. Розглянемо основні їх.

Насамперед, необхідно вирішити нетривіальне завдання вибору безлічі критеріїв оцінки альтернатив. Для цього може бути використаний, зокрема, підхід, що передбачає декомпозицію головної мети до рівня деталізації, коли для нижнього рівня ієрархії цілей можна сформулювати критерії, що дозволяють адекватно описати ступінь досягнення цілей при прийнятті тієї чи іншої альтернативи.

Стосовно завдання вибору варіанта інформатизації в якості головної мети фірми може бути, наприклад, підвищення рентабельності фірми, а критеріями оцінки варіантів можуть виступати, наприклад, витрати на інформатизацію, здатність підтримувати рішення, можливість адаптації до інших видів діяльності фірми, можливість захисту інформації, час реакції на запит, надійність обладнання та ін. Набори значень цих критеріїв використовуються для опису результатів альтернативних варіантів рішень (надалі, «альтернатив») з подальшим формуванням функції корисності, що відображає переваги ОПР та дозволяє кожному набору значень оцінок альтернатив за обраними критеріями поставити у відповідність кілька, що характеризує ефективність альтернативи. Побудова функції корисності на підставі формального опису переваг передбачає, зокрема, здатність ОПР виконати заміщення, тобто у найпростішому випадку вибрати в якості кращої одну з двох альтернатив, причому одна з них перевершує другу за деяким критерієм А, але поступається за критерієм при рівних оцінках за іншими. Для цього ОПР має повністю володіти проблемою. Стосовно завдання вибору варіанта інформатизації фірми ОПР повинен, зокрема, вказати, який із двох варіантів кращий для підвищення рентабельності фірми: той, який забезпечує можливість підтримки прийняття рішень, але поступається за часом реакції, або варіант, що забезпечує менший час реакції на запит, але не містить засобів підтримки прийняття рішень за рівних оцінок за іншими критеріями. Як бачимо, ОПР зовсім не просто зробити вибір! Для справді складних комплексних проблем припущення про здатність ОПР здійснювати подібні заміщення дуже сильно, ОПР неспроможна мати всіма знаннями, необхідні кваліфікованого розв'язання цього завдання. Тому для вирішення таких складних проблем слід залучати багатьох фахівців у різних галузях знань, що при використанні такого підходу дуже важко.

Спрощення побудови функції корисності досягається під час використання методу аналітичних ієрархічних процесів (АІП). Сутність підходу ось у чому.

ОПР здійснює спочатку попарне порівняння важливості обраних критеріїв. Потім цей метод використовується для попарного порівняння альтернатив щодо кожного обраного критерію. За підсумками цього СППР розраховує коефіцієнти значимості критеріїв, коефіцієнти значимості альтернатив щодо

кожного критерію, що дозволяє розрахувати кожної альтернативи значення лінійної функції корисності. Розвитком цього підходу є метод аналітичних мережевих процесів (ANP), який дає змогу врахувати взаємозв'язки між критеріями. Однак використання кожного з трьох названих підходів пов'язане з необхідністю ОПП вирішувати складне завдання вибору набору критеріїв, які адекватно описують вплив альтернатив на досягнення головної мети. Особливо складним стає це завдання при ухваленні рішень щодо формування комплексних цільових програм.

Досвід розробки складних СППР показує, що для ОПП поняття «головна мета» програми ближче, ніж поняття «критерій». У результаті розробки складних СППР зручніше користуватися поняттями «мета», «підціль», «над ціль» і розглядати процес виконання програми як досягнення взаємозалежних підцелей. Для оцінки ефективності проектів у разі найбільш прийнятним представляється підхід, який можна назвати багатоцільовим оцінюванням альтернатив. Сутність його полягає у наступному. Кожен із проектів (альтернатива) оцінюється єдиним показником ефективності – ступенем впливу його виконання на досягнення головної мети програми. Стосовно завдання вибору варіанта інформатизації фірми головною метою може бути «підвищення рентабельності». Безпосередня оцінка впливу виконання проектів для досягнення головної мети програми є важким завданням. Тому використовується прийом, заснований на побудові ієрархії цілей, тобто декомпозиції головної мети програми на підціль, де на нижньому рівні ієрархії знаходяться цілі проектів.

Стосовно вибору варіанта інформатизації фірми як безпосередні підціль головної мети можуть бути конкретні цілі такі, як: прискорення оборотності коштів, збільшення контрольованої частини ринку, зниження ціни продукції і т. д. Зауважимо, що між підцільми можуть існувати взаємозв'язки, наприклад, підціль «зниження ціни продукції» позитивно впливає на досягнення підціль «збільшення контрольованої частини ринку». Стосовно аналізованого прикладу як проекти виступають різні варіанти інформатизації. У загальному випадку ієрархія цілей містить як стимулюючі, так і гнітючі прямі та зворотні зв'язки між цілями різних типів, що дозволяє відобразити різноманіття взаємовпливів цілей повніше, ніж при використанні критеріїв.

Після побудови ієрархії експерти кількісно оцінюють вплив підцелей, зокрема і проектів, для досягнення безпосередніх над-цілей, використовуючи способи парних порівнянь чи безпосереднього оцінювання. З цієї інформації, і навіть структури ієрархії підцелей розраховуються показники відносної ефективності проектів. Така технологія побудови ієрархії цілей дозволяє дійсно кваліфіковано структурувати проблему, намітити проміжні цілі та перетворити її на базу знань про проблему, що акумулює знання експертів саме в галузях їхньої дійсної («перевіреної часом») компетенції.

Очевидно, напрошується питання: «А де тут ОПП? Чи не довірився він повністю експертам і чи не передав їм право приймати рішення?» Ні це не

так. ОПР вирішує стратегічні питання: формулює головну мету програми, висловлюючи цим свій задум. З іншого боку, він, зазвичай, визначає підцілі головної мети, вирішуючи оперативні завдання. Якщо до того ж він – високий професіонал у якійсь вузькій галузі (наприклад, фінансист), то може виступати в ролі експерта і при декомпозиції деякої мети нижчого порядку, що належить до своєї компетенції. Чи може виникнути ще одне питання: «Чи не можна, здійснивши декомпозицію головної мети на її безпосередні підцілі, розділити складне завдання на простіші, для яких отримані підцілі є головними?» Як правило, не можна, оскільки при цьому не враховуватимуться багато реально існуючих зв'язків між цілями. Як приклад того, до чого може це привести, згадаємо, як у недалекому минулому організовувалася робота деяких галузей, наприклад, залізничного транспорту. Показником ефективності цієї галузі були тонно-кілометри. При цьому часто ці тонно-кілометри набиралися за рахунок абсолютно непотрібних і навіть шкідливих з точки зору ефективності всього господарства зустрічних перевезень. Чи виробництво товарів, що не знаходять попиту? Але галузі плани щодо своїх показників ефективності перевищували! Стосовно вирішення завдання інформатизації фірми незалежний від головної мети розгляд такої проблеми, як «підвищення якості діловодства», може привести до висновку про необхідність заміни, наприклад, РС/АТ 486,

Яка складність ієрархії цілей? Жодній людині не під силу врахувати велику кількість зв'язків, що розростається, при прийнятті рішень. Однак сучасні ПК справляються з такими завданнями. Побудова у повному обсязі ієрархії досить трудомістка та дорога. Однак – така праця виправдовується відповідним виникненням бази знань для вирішення не одного управлінського завдання. Вона може і повинна використовуватись для вирішення багатьох завдань. І тому передбачаються кошти «підв'язки» варіантів рішень (альтернатив) різних завдань до ієрархії цілей. Можна, наприклад, ієрархію цілей, побудовану вибору варіанта інформатизації, використовуватиме визначення найкращого щодо тієї ж головної мети варіанта організації рекламної кампанії, вибору типу продукції тощо. буд. Тут зміниться лише зміст проєктів та його зв'язку з підцілями.

Розглянуті методи підтримки рішень ґрунтуються на визначенні коефіцієнтів впливу експертами чи ОПР. Проте розроблено методи підтримки прийняття рішень, що адаптуються до попереднього досвіду ОПР. Необхідність такого підходу виникає за періодичного вирішення однотипних завдань. Пояснимо це простому прикладі. Нехай вам, як керівнику, потрібно вибрати одного з кількох кандидатів та призначити його на вакантну посаду, причому на подібних посадах у вас працює кілька співробітників. Система запропонує вам проранжувати цих співробітників за рівнем успішності роботи. Як правило, вам це не складе труднощів, хоча в більшості випадків вам буде важко чітко сформулювати критерій, за яким ви провели ранжування. Далі система

запропонує вам сформулювати приватні критерії (професійно важливі якості), які, на вашу думку, визначають успішність роботи на даній посаді (наприклад, сумлінність, обов'язковість, ерудиція, працьовитість, комунікабельність і т. д.) і проранжувати співробітників, що вже працюють на даній посаді, по кожному з цих приватних критеріїв. На підставі цього система розраховує коефіцієнти важливості цих приватних критеріїв, які несуть інформацію про ваш досвід у прийнятті рішення щодо цього кадрового питання. Далі вам слід проранжувати кандидатів на посаду за кожним із приватних критеріїв, і система, використовуючи ці оцінки та раніше розраховані коефіцієнти значущості приватних критеріїв, визначить рейтинг кожного з кандидатів та запропонує вам найгіднішого кандидата. Неважко бачити, що ця система допоможе вам вибрати тип мережевого обладнання, виконавця робіт з інформатизації вашої фірми тощо.

Виходячи з викладених принципів підтримки рішень, доцільно визначити три класи СППР залежно від складності розв'язуваних завдань та областей застосування.

СППР першого класу, що мають найбільші функціональні можливості, призначені для застосування в органах державного управління вищого рівня (адміністрація президента, міністерства) та органах управління великих компаній (рада директорів корпорації) при плануванні великих комплексних цільових програм для обґрунтування рішень щодо включення до програми різних політичних, соціальних або економічних заходів та розподілу між ними ресурсів на основі оцінки їх впливу на досягнення основної мети програми. СППР цього класу є системами колективного користування,

СППР другого класу є системами індивідуального користування, бази знань яких формуються безпосереднім користувачем. Вони призначені використання державними службовцями середнього рангу, і навіть керівниками малих і середніх фірм на вирішення оперативних завдань управління.

СППР третього класу є системами індивідуального користування, що пристосовуються до досвіду користувача. Вони призначені на вирішення часто зустрічаються прикладних завдань системного аналізу та управління (наприклад, вибір суб'єкта кредитування, вибір виконавця роботи, призначення посаду тощо.). Такі системи забезпечують одержання розв'язання поточної задачі на основі інформації про результати практичного використання рішень цієї задачі, прийнятих у минулому. Крім того, системи цього класу призначені для використання в торгових підприємствах, які торгують дорогими товарами тривалого користування, як засіб «інтелектуальної реклами», що дозволяє покупцеві вибрати товар на основі свого досвіду застосування товарів аналогічного призначення.

У комп'ютерному світі відбуваються дивовижні події: люди одного покоління стали свідками стрімкого прогресу в галузі обчислювальної техніки та програмного забезпечення – від «мастодонтів» типу «Урал» та програмуван-

ня обчислювальних завдань у машинних кодах до сучасних ПЕОМ та конструювання складних інформаційних систем в об'єктно-орієнтованому середовищі. Такий самий прогрес спостерігається і в прикладному програмному забезпеченні. Очевидно, у недалекому майбутньому звичайними користувачами повсякденно широко застосовуватимуться системи, які є інтелектуальними помічниками, зокрема системи підтримки прийняття рішень. Такі системи реально існують. Треба повірити у їхні можливості та подолати психологічний бар'єр (як, віддати «ящику» святе право приймати рішення?!). Це право завжди залишається за людиною, просто система врахує ваші бажання, узагальнить думки знаючих людей та запропонує вам оцінки варіантів. (Адже навіть диктатори завжди мали радників!) Той, хто раніше почне застосовувати ці системи у повсякденній діяльності (так само широко, як, наприклад, Word), отримає перевагу перед своїми конкурентами.

Контрольні питання до підрозділу 2.1

- 1. Чому володіння інформацією (даними) – необхідна, але не достатня умова для прийняття правильного управлінського рішення?*
- 2. Як і чому пов'язані підтримка прийняття управлінських рішень та багаторазово відновлюваний багатьма експертний аналіз?*
- 3. Що є підтримка прийняття управлінських рішень як логічне явище?*
- 4. Чим характеризуються "системи підтримки прийняття рішень" як сучасне спеціалізоване ПЗ?*
- 5. У чому загальна класична сутність СППР?*
- 6. Який взаємозв'язок особи, яка приймає рішення (ОПР) і СППР?*
- 7. У чому суть методу аналітичних ієрархічних (мережевих) процесів?*
- 8. Як у СППР прийнято виділяти (пов'язувати) цілі, підцілі?*
- 9. Які СППР(ПЗ) виділяють залежно від способу формування та додатків?*
- 10. Що таке «психологічний бар'єр» під час освоєння СППР? Які шляхи його подолання?*

2.2. Коротка історія розвитку «систем підтримки ухвалення рішень»

Система підтримки прийняття рішень (СППР) (англ. DecisionSupportSystem, DSS) – комп'ютерна автоматизована система, метою якої є допомога людям, які приймають рішення у складних умовах для повного та об'єктивного аналізу предметної діяльності. СППР виникли внаслідок злиття управлінських інформаційних систем та систем управління базами даних.

Для аналізу та вироблення пропозицій у СППР використовуються різні методи. Це можуть бути: інформаційний пошук, інтелектуальний аналіз даних, пошук знань у базах даних, міркування на основі прецедентів, імітаційне моделювання, генетичні алгоритми, нейронні мережі та ін. Деякі з цих мето-

дів розроблено в рамках штучного інтелекту. Якщо в основі роботи СППР лежать методи штучного інтелекту, то говорять про інтелектуальну СППР, або ІСППР. Близькі до СППР класи систем - це експертні системи та автоматизовані системи управління.

Сучасні СППР є системи, максимально пристосовані до вирішення завдань повсякденної управлінської діяльності, є інструментом, покликаним надати допомогу особам, що приймають рішення (ОПР). З допомогою СППР може проводитися вибір рішень деяких неструктурованих і слабоструктурованих завдань, зокрема і багатокритеріальних. СППР зазвичай є результатом мультидисциплінарного дослідження, що включає теорії баз даних, штучного інтелекту, інтерактивних комп'ютерних систем, методів імітаційного моделювання.

Як слушно зазначено у [9], «... з появи перших розробок зі створення СППР, було надано чітке визначення СППР...».

Ранні визначення СППР (на початку 70-х років минулого століття) відображали такі три моменти:

(1) можливість оперувати з неструктурованими чи слабоструктурованими завданнями, на відміну завдань, із якими має справу дослідження операцій;

(2) інтерактивні автоматизовані (тобто реалізовані на базі комп'ютера) системи;

(3) поділ даних та моделей. Наведемо визначення СППР: СППР – сукупність процедур із обробки даних і суджень, які допомагають керівнику прийняття рішень, заснована на використанні моделей [2].

Далі уточнювалося, що СППР – це інтерактивні автоматизовані системи, які допомагають особі, яка приймає рішення, використовувати дані та моделі для вирішення слабоструктурованих проблем [10], або ... СППР – це система, яка забезпечує користувачам доступ до даних та/або моделей, тому вони можуть приймати найкращі рішення [11].

Слід визнати, що й нині немає загальноприйнятого визначення СППР, оскільки конструкція СППР істотно залежить від виду завдань, на вирішення яких розробляється, від доступних даних, інформації та знань, і навіть від користувачів системи. Проте деякі елементи і характеристики, загальновізані, як частини СППР: СППР – у більшості випадків – це інтерактивна автоматизована система, яка допомагає користувачеві (ОПР) використовувати дані та моделі для ідентифікації та вирішення завдань і прийняття рішень. Система повинна мати можливість працювати з інтерактивними запитамі з досить простою для вивчення мовою запитів [7].

СППР володіє такими чотирма основними характеристиками [12]:

- 1) СППР використовує і дані та моделі;
- 2) СППР призначені для допомоги менеджерам у прийнятті рішень для слабоструктурованих та неструктурованих завдань;
- 3) Вони підтримують, а чи не замінюють, вироблення рішень менеджерами;

4) Мета СППР – покращення ефективності рішень.

Характеристики ідеальної СППР:

- оперує із слабоструктурованими рішеннями;
- призначена для ОПР різного рівня;
- може бути адаптована для групового та індивідуального використання;
- підтримує як взаємозалежні, і послідовні рішення;
- підтримує три фази процесу вирішення: інтелектуальну частину, проектування та вибір;
- підтримує різноманітні стилі та методи вирішення, що може бути корисно при вирішенні задачі групою ОПР;
- є гнучкою та адаптується до змін як організації, так і її оточення;
- проста у використанні та модифікації;
- покращує ефективність процесу прийняття рішень;
- дозволяє людині керувати процесом прийняття рішень з допомогою комп'ютера, а чи не навпаки;
- підтримує еволюційне використання і легко адаптується до вимог, що змінюються;
- може бути легко побудовано, якщо може бути сформульована логіка конструкції СППР;
- підтримує моделювання;
- дозволяє використовувати знання.

До середини 60-х років минулого століття створення великих інформаційних систем (ІС) було надзвичайно дорогим, тому перші ІС менеджменту (так звані Management Information Systems - MIS) були створені в ці роки лише у досить великих компаніях. MIS призначалися на підготовку періодичних структурованих звітів для менеджерів. Наприкінці 60-х років з'являється новий тип ІС - модель-орієнтовані СППР (Model-oriented Decision Support Systems - DSS) або системи управлінських рішень (Management Decision Systems - MDS). На думку одних із першовідкривачів СППР Keen PGW, Scott Morton MS [13], концепція підтримки рішень була розвинена на основі «теоретичних досліджень у галузі прийняття рішень... та технічних робіт із створення інтерактивних комп'ютерних систем».

У 1971 р. – опубліковано книгу Scott Morton'a, у якій, мабуть, вперше було описано результати впровадження СППР, заснованої на використанні математичних моделей..

1974 р. – у роботі [14] дано визначення ІС менеджменту – MIS (Management Information System): «MIS – це інтегрована людино-машинна система забезпечення інформацією, що підтримує функції операцій, менеджменту та прийняття рішень в організації. Системи використовують комп'ютерну техніку та програмне забезпечення, моделі управління та прийняття рішень, а також базу даних».

1975 р. – J.D.C. Little у роботі [2] запропонував критерії проектування СППР у менеджменті.

1978 – опубліковано підручник з СППР [13], в якому вичерпно описані аспекти створення СППР: аналіз, проектування, впровадження, оцінка та розробка.

1980 р. - в [15] дано основи класифікації СППР.

1981 р. – Bonczek, Holsapple та Whinston у книзі [7] створили теоретичні засади проектування СППР. Вони виділили чотири необхідні компоненти, властиві всім СППР:

1) Мовна система (Language System - LS) - СППР може приймати всі повідомлення;

2) Система презентацій (Presentation System (PS)) (СППР може видавати свої повідомлення);

3) Система знань (Knowledge System – KS) - всі знання СППР зберігає;

4) Система обробки завдань (Problem-ProcessingSystem (PPS)) – програмний «механізм», який намагається розпізнати та вирішувати завдання під час роботи СППР.

1981 р. – у книзі [16] R.Sprague та E.Carlson описали, яким чином на практиці можна побудувати СППР. Тоді була розроблена інформаційна система керівника (Executive Information System (EIS)) – комп'ютерна система, призначена для забезпечення поточної адекватної інформації для підтримки прийняття управлінських рішень менеджером.

Починаючи з 1990-х, розробляють звані DataWarehouses – сховища даних.

У 1993 р. Е. Коддін (E.F. Codd) для СППР спеціального виду було запропоновано термін OLAP (OnlineAnalyticalProcessing) - оперативний аналіз даних, онлайн аналітична обробка даних для підтримки прийняття важливих рішень. Вихідні дані для аналізу представлені у вигляді багатовимірного куба, яким можна отримувати потрібні розрізи - звіти. Виконання операцій над даними здійснюється OLAP-машиною. За способом зберігання даних розрізняють MOLAP, ROLAP та HOLAP. За місцем розміщення OLAP-машини розрізняються OLAP-клієнти та OLAP-сервери. OLAP-клієнт робить побудову багатовимірного куба та обчислення на клієнтському ПК, а OLAP-сервер отримує запит, обчислює та зберігає агрегатні дані на сервері, видаючи лише результати.

На початку нового тисячоліття було створено СППР з урахуванням Web.

27 жовтня 2005 року в Москві на Міжнародній конференції "Інформаційні та телемедичні технології в охороні здоров'я" (ITTHC 2005), А. Пастухов (Росія) представив СППР нового класу - PSTM (Personal Information Systems of Top Managers). Основною відмінністю PSTM від існуючих СППР є побудова системи для конкретної особи, яка приймає рішення з попередньою логіко-аналітичною обробкою інформації в автоматичному режимі та виведенням інформації на один екран.

Контрольні питання до підрозділу 2.2

1. Що забезпечували перші СППР 60-х ХХ в.?
2. Чим характеризувалися СППР 70-х ХХ в.?
3. Що додалося нового у СППР 75-80 роках ХХ ст.?
4. Чим додатково характеризувалися СППР 80-х ХХ в.?
5. Коли було розроблено інформаційну систему керівника (*Executive Information System (EIS)*), підтримки управлінських рішень менеджером?
6. Які найповніші показники СППР, додано після 90-х ХХ в.?
7. Коли СППР почали застосовувати звані *DataWarehouses*-сховища даних?
8. Коли у СППР почали застосовувати так звані *OLAP*-технології? Що означає *OLAP*?
9. Коли СППР почали застосовувати *Web* – технології?
10. Коли з'явилися СППР *PSTM (Personal Information System sof Top Managers)* класу?

2.3. Класифікація «СППР»

Для СППР відсутня як єдине загальноприйняте визначення, а й вичерпна класифікація. Різні автори пропонують різні класифікації.

На рівні користувача *Naettenschwiler (1999) [17]* поділяє СППР на пасивні, активні та кооперативні СППР. Пасивною СППР називається система, яка допомагає процесу ухвалення рішення, але не може винести пропозицію, яке рішення ухвалити. Активна СППР може зробити пропозицію, яке рішення слід вибрати. Кооперативна дозволяє ОПР змінювати, поповнювати або покращувати рішення, запропоновані системою, посилаючи потім ці зміни до системи для перевірки. Система змінює, поповнює чи покращує ці рішення та посилає їх знову користувачеві. Процес продовжується до отримання узгодженого рішення.

На концептуальному рівні *Power (2003) [18]* відрізняє СППР, керовані повідомленнями (*Communication-Driven DSS*), СППР, керовані даними (*Data-Driven DSS*), СППР, керовані документами (*Document-Driven DSS*), СППР, керовані знаннями (*Knowledge -Driven DSS*) та СППР, керовані моделями (*Model-Driven DSS*). СППР, керовані моделями, характеризуються переважно доступ і маніпуляції з математичними моделями (статистичними, фінансовими, оптимізаційними, імітаційними). Зазначимо, що деякі *OLAP* системи, що дозволяють здійснювати складний аналіз даних, можуть бути віднесені до гібридних СППР, які забезпечують моделювання, пошук та обробку даних.

Керована повідомленнями (*Communication-Driven DSS*) (раніше групова СППР - *GDSS*) СППР підтримує групу користувачів, які працюють над виконанням спільного завдання.

СППР, керовані даними (Data-Driven DSS) чи СППР, орієнтовані працювати з даними (Data-oriented DSS) переважно, орієнтуються доступ і маніпуляції з даними. СППР, керовані документами (Document-Driven DSS), керують, здійснюють пошук та маніпулюють неструктурованою інформацією, заданою у різних форматах. Нарешті, СППР, керовані знаннями (Knowledge-Driven DSS), забезпечують вирішення завдань як фактів, правил, процедур.

На технічному рівні Power (1997) [19] розрізняє СППР всього підприємства та настільну СППР. СППР всього підприємства підключена до великих сховищ інформації та обслуговує багатьох менеджерів підприємства. Настільна СППР - це мала система, яка обслуговує лише один комп'ютер користувача. Існують і інші класифікації (Alter [15], Holsapple та Whinston [20], Golden, Hevner та Power [21]). Зазначимо лише, що чудова для свого часу класифікація Alter'a, яка розбивала все СППР на сім класів, нині дещо застаріла.

Залежно від даних, з якими ці системи працюють, СППР умовно можна поділити на оперативні та стратегічні. Оперативні СППР призначені для негайного реагування на зміни поточної ситуації в управлінні фінансово-господарськими процесами компанії. Стратегічні СППР спрямовані на аналіз значних обсягів різномірної інформації, які з різних джерел. Найважливішою метою цих СППР є пошук найбільш раціональних варіантів розвитку бізнесу компанії з урахуванням впливу різних факторів, таких як кон'юнктура цільових для компанії ринків, зміни фінансових та ринків капіталів, зміни в законодавстві та ін. СППР першого типу отримали назву Інформаційних Систем Керівництва Systems, ICP). По суті, вони є кінцевими наборами звітів, побудовані на підставі даних із транзакційної інформаційної системи підприємства, що в ідеалі адекватно відображає в режимі реального часу основні аспекти виробничої та фінансової діяльності. Для ICP характерні такі основні риси:

- звіти, як правило, базуються на стандартних для організації запитах; кількість останніх відносно невелика;
- ICP представляє звіти у максимально зручному вигляді, що включає, поряд з таблицями, ділову графіку, мультимедійні можливості тощо;
- як правило, ICP орієнтовані на конкретний вертикальний ринок, наприклад, фінанси, маркетинг, управління ресурсами.

СППР іншого типу припускають досить глибоке опрацювання даних, спеціально перетворених те щоб їх було зручно використовувати під час процесу прийняття рішень. Невід'ємним компонентом СППР цього рівня є правила ухвалення рішень, які на основі агрегованих даних дають можливість менеджерам компанії обґрунтовувати свої рішення, використовувати фактори сталого зростання бізнесу компанії та знижувати ризики. СППР другого типу останнім часом активно розвиваються. Технології цього будуються на принципах багатовимірної подання та аналізу даних (OLAP). Під час створення СППР можна використовувати Web-технології. Нині СППР з урахуванням Web-технологій низки компаній є синонімами СППР підприємства.

Архітектура СППР є різними авторами по-різному. Наприклад, Marakas в [22] запропонував узагальнену архітектуру, що складається з п'яти різних частин:

- (a) система управління даними (the data management system – DBMS),
- (b) система управління моделями (the model management system – MBMS),
- (c) машина знань (the knowledge engine (KE)), (d) інтерфейс користувача (the user interface) та
- (e) користувачі (user(s)).

Контрольні питання до підрозділу 2.3

1. Як прийнято класифікувати СППР користувачам згідно Haettenschwiler (1999)?
2. Як прийнято класифікувати СППР на концептуальному рівні за Power (2003)?
3. Як прийнято класифікувати СППР технічно рівні по Power (1997)?
4. Як прийнято класифікувати СППР залежно від даних, із якими ці системи працюють?
5. Яку архітектуру СППР запропонував Marakas 1999 р.?
6. Як доречно класифікувати сучасні СППР згідно з застосовуваними математичними моделями?
7. Як класифікують СППР за способами формування та відповідними додатками?

РОЗДІЛ 3. ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ

3.1. Основні поняття та архітектура систем баз даних

3.1.1. Основні поняття і визначення

Сучасні автори часто вживають терміни «банк даних» і «база даних» як синоніми, однак ці поняття розрізняються. Мають місце наступні визначення банку даних і бази даних [23]:

База даних (БД) – іменована сукупність даних, що відбиває стан об'єктів і їхнє відношення в розглянутій предметній області.

Банк даних (БнД) – це система спеціальним образом організованих даних – баз даних, програмних, технічних, мовних, організаційно-методичних засобів, призначених для забезпечення централізованого нагромадження й колективного багатоцільового використання даних.

Найпростіші визначення:

Дані – всі можливі відомості (потрібні та непотрібні).

Інформація – дані, але тільки ті, що необхідні для прийняття рішення (певному суб'єктові, у певний час, з певного питання).

База даних (БД) – дані, упорядковані таким чином, що з ними можна проводити автоматичну роботу (додавання нових даних, ефективний пошук, тощо).

Система керування базами даних (СКБД або СУБД) – система (програмні засоби, тощо) забезпечення автоматизації роботи з базою даних.

Банк даних (БД або БнД) – поєднання бази даних та системи керування базами даних.

Тобто **БнД = БД+СКБД**

Термінологія в СКБД, та й самі терміни «база даних» і «банк даних», частково запозичені з фінансової діяльності. Це запозичення – не випадкове й пояснюється тим, що робота з інформацією і робота з грошовими масами багато в чому схожі, оскільки і там і тут відсутня персоніфікація об'єкта роботи: дві банкноти номіналом у сто доларів настільки ж нерозрізювані й взаємозамінні, як два однакові байти. Ви можете покласти гроші на деякий рахунок і надати можливість вашим родичам чи колегам використовувати їх для інших цілей; ви можете доручити банку оплачувати ваші витрати з вашого рахунка чи одержати гроші готівкою в іншому банку, і це будуть вже інші грошові купюри, але їхня цінність буде еквівалентна тій, котру ви мали, коли клали їх на рахунок.

Система керування базами даних (СКБД) (СУБД) – сукупність мовних і програмних засобів, призначених для створення, ведення й спільного використання БД багатьма користувачами.

Застосунки (застосування, додатки, програми) – програми, за допомогою яких користувачі працюють з базою даних, називаються.

Користувач БД – програма або людина, що звертається до БД мовою мета-даних (ММД).

У загальному випадку з однією базою даних можуть працювати багато різних додатків. Наприклад, якщо база даних моделює деяке підприємство, то для роботи з нею може бути створений додаток, що обслуговує підсистему обліку кадрів, інший додаток може бути призначений для роботи підсистеми розрахунку заробітної плати співробітників, третій додаток працює як підсистема складського обліку, четвертий додаток займається плануванням виробничого процесу і т.п. При розгляді додатків, що працюють з однією базою даних, передбачається, що вони можуть працювати паралельно і незалежно друг від друга, і саме СКБД призначена забезпечити роботу багатьох додатків з єдиною базою даних таким чином, щоб кожен із них виконувалося коректно, але враховуючи всі зміни в базі даних, внесені іншими додатками.

Запит – це процес звертання користувача до БД із метою введення, одержання або зміни інформації в БД.

Транзакція – послідовність операцій модифікації даних у БД, що переводить БД із одного несуперечливого стану в інший несуперечливий стан.

3.1.2. Архітектура СКБД

Трирівнева модель організації СКБД ANSI

У процесі наукових досліджень, присвячених тому, як саме повинна бути скомпонована СКБД, пропонувалися різні моделі. Найбільш життєздатною з них виявилася запропонована американським комітетом зі стандартизації ANSI (American National Standards Institute) трирівнева система організації БД [24].

Рівні організації СКБД за моделлю ANSI:

1. Рівень зовнішніх моделей – самий верхній рівень, де кожна модель має своє «бачення» даних. Цей рівень визначає точку зору на БД окремих додатків. Кожен додаток бачить і обробляє тільки ті дані, що необхідні саме цьому додатку. Наприклад, система розподілу робіт використовує зведення про кваліфікацію співробітника, але її не цікавлять зведення про оклад, домашню адресу і телефон співробітника; навпаки, саме ці зведення використовуються в підсистемі відділу кадрів.

2. Концептуальний рівень – центральна керуюча ланка, тут база даних представлена в найбільш загальному вигляді, що поєднує дані, використовувани всіма додатками, які працюють з даною БД. Фактично концептуальний рівень відбиває узагальнену модель предметної області (об'єктів реального світу), для якої створювалася база даних. Як будь-яка модель, концептуальна модель відбиває тільки істотні, з погляду обробки, особливості об'єктів реального світу (наприклад, вартість банкноти, а не колір).

3. Фізичний рівень – власне дані, розташовані в файлах чи в сторінкових структурах, розташованих на зовнішніх носіях інформації.

Фізична і логічна незалежність при роботі з даними [25]

Архітектура ANSI дозволяє забезпечити логічну (між рівнями 1 і 2) і фізичну (між рівнями 2 і 3) незалежність при роботі з даними.

Логічна незалежність припускає можливість зміни одного додатка без координування інших додатків, що працюють з цією ж базою даних.

Фізична незалежність припускає можливість переносу збереженої інформації з одних носіїв на інші при збереженні працездатності всіх додатків, що працюють з даною базою даних. Це саме те, чого не вистачало при використанні файлових систем.

А виділення концептуального рівня дозволило розробити апарат централізованого керування базою даних.

3.1.3 Робота з СКБД

Процес проходження користувачького запиту

Запит – це процес звертання користувача до БД із метою введення, одержання або зміни інформації в БД [26].

Розглянемо послідовність взаємодій:

1. Користувач посилає СКБД запит на одержання даних із БД {1}.
2. Аналіз прав користувача і зовнішньої моделі даних, що відповідає даному користувачу, й підтверджує чи забороняє доступ даного користувача до запитаних даних {2, 3}.
3. У випадку заборони на доступ до даних СКБД повідомляє користувача про це {12} і припиняє подальший процес обробки даних, у протилежному випадку СКБД визначає частину концептуальної моделі, що зачіпається запитом користувача.
4. СКБД одержує інформацію про запитану частину концептуальної моделі {4, 5}.
5. СКБД запитує інформацію про місце розташування даних на фізичному рівні (файли чи фізичні адреси) {6}.
6. У СКБД повертається інформація про місце розташування даних у термінах операційної системи {7}.
7. СКБД просить операційну систему надати необхідні дані, використовуючи засоби операційної системи {8}.
8. Операційна система здійснює перекачування інформації з пристроїв зберігання і пересилає її в системний буфер {9}.
9. Операційна система оповіщає СКБД про закінчення пересилання {10}.
10. СКБД вибирає з доставленої інформації, що знаходиться в системному буфері, тільки те, що потрібно користувачу, і пересилає ці дані в робочу область користувача {11}.

БМД – це База Метаданих, саме тут і зберігається вся інформація про використовувані структури даних, логічної організації даних, права доступу користувачів і, нарешті, фізичне розташування даних [27]. Для керування

БМД існує спеціальне програмне забезпечення адміністрування баз даних, що призначено для коректного використання єдиного інформаційного простору багатьма користувачами. Чи завжди запит проходить повний цикл? Звичайно, ні. СКБД має досить розвинутий інтелект, що дозволяє їй не повторювати безглуздих дій. І тому, наприклад, якщо цей же користувач повторно звернеться до СКБД із новим запитом, то для нього вже не будуть перевірятися зовнішня модель і права доступу, а якщо подальший аналіз запиту покаже, що дані можуть знаходитися в системному буфері, то СКБД здійснить тільки {11} і {12} кроки в обробці запиту.

Зрозуміло, механізм проходження запиту в реальних СКБД набагато складніший, але і ця спрощена схема показує, наскільки серйозними і складними повинні бути механізми обробки запитів, підтримувані реальними СКБД.

Користувачі банків даних

Як будь-який програмно-організаційно-технічний комплекс, банк даних існує в часі й у просторі. Він має визначені стадії свого розвитку:

1. Проектування.
2. Реалізація.
3. Експлуатація.
4. Модернізація і розвиток.
5. Повна реорганізація.

На кожному етапі свого існування з банком даних зв'язані різні категорії користувачів. Визначимо **основні категорії користувачів** і їхню роль у функціонуванні банку даних [28].

Кінцеві користувачі. Це основна категорія користувачів, в інтересах яких і створюється банк даних. У залежності від особливостей створюваного банку даних коло його кінцевих користувачів може істотно розрізнитися. Це можуть бути випадкові користувачі, що звертаються до БД час від часу за одержанням деякої інформації, а можуть бути регулярні користувачі. Як випадкові користувачі можуть розглядатися, наприклад, можливі клієнти вашої фірми, що переглядають каталог вашої продукції чи послуг з узагальненим чи докладним описом того й іншого. Регулярними користувачами можуть бути ваші співробітники, що працюють зі спеціально розробленими для них програмами, що забезпечують автоматизацію їхньої діяльності при виконанні своїх посадових обов'язків. Наприклад, менеджер, що планує роботу сервісного відділу комп'ютерної фірми, має у своєму розпорядженні програму, що допомагає йому планувати і розподіляти поточні замовлення, контролювати хід їхнього виконання, замовляти на складі необхідні комплектуючі для нових замовлень.

Головний принцип полягає в тому, що від кінцевих користувачів не повинно вимагатися яких-небудь спеціальних знань в області обчислювальної техніки і мовних засобів.

Адміністратори банку даних. Це група користувачів, що:

- на початковій стадії розробки банку даних відповідають за його оптимальну організацію з погляду одночасної роботи багатьох кінцевих користувачів;

- на стадії експлуатації відповідають за коректність роботи даного банку інформації в багатокористувацькому режимі;
- на стадії розвитку і реорганізації відповідають за можливість коректної реорганізації банку без припинення його поточної експлуатації.

Розробники й адміністратори додатків. Це група користувачів, що функціонує під час проектування, створення й реорганізації банку даних. Адміністратори додатків координують роботу розробників при розробці конкретного додатка чи групи додатків, об'єднаних у функціональну підсистему. Розробники конкретних додатків працюють з тією частиною інформації з бази даних, яка потрібна для конкретного додатка.

Не в кожному банку даних можуть бути виділені всі типи користувачів. Наприклад, при розробці інформаційних систем з використанням настільних СКБД (2 етап) адміністратор банку даних, адміністратор додатків і розробник часто існували в одному обличчі. Однак при побудові сучасних складних корпоративних баз даних, що використовуються для автоматизації всіх чи більшої частини бізнес-процесів у великій фірмі чи корпорації, можуть існувати і групи адміністраторів додатків, і відділи розробників. Найбільш складні обов'язки покладені на групу адміністратора БД.

Розглянемо її більш докладно. У складі групи адміністратора БД повинні бути:

- системні аналітики;
- проектувальники структур даних і зовнішнього стосовно банку даних інформаційного забезпечення;
- проектувальники технологічних процесів обробки даних;
- системні і прикладні програмісти;
- оператори і фахівці з технічного обслуговування.

Якщо мова йде про комерційний банк даних, то важливу роль тут грають фахівці з маркетингу.

У цьому підрозділі приводиться класифікація СКБД і розглядаються основні їхні функції. У якості основних класифікаційних ознак можна використовувати наступні:

- вид програми;
- характер використання;
- модель даних.

Названі ознаки істотно впливають на цільовий вибір СКБД і ефективність використання розроблюваної інформаційної системи.

3.1.4. Класифікація СКБД

Класифікація СКБД за видом програм [29]

Повнофункціональні СКБД (ПФСКБД) являють собою традиційні СКБД, що спочатку з'явилися для великих машин, потім для міні-машин і для ПЕОМ. З числа всіх СКБД сучасні ПФСКБД є найбільш численними і потужними по своїх можливостях. До ПФСКБД відносяться, наприклад, такі па-

кети, як ClarionDatabaseDeveloper, DataBase, Dataplex, dBase IV, Microsoft Access, Microsoft FoxPro і Paradox R:BASE.

Звичайно ПФСКБД мають розвинутий інтерфейс, що дозволяє за допомогою команд меню виконувати основні дії з БД: створювати і модифікувати структури таблиць, вводити дані, формувати запити, розробляти звіти, виводити їх на друк і т.п. Для створення запитів і звітів не обов'язкове програмування, а зручно користатися мовою QBE (QueryByExample – формулювання запитів за зразком). Більшість ПФСКБД включають також засоби програмування для професійних розроблювачів.

Деякі системи мають у якості допоміжних і додаткові засоби проектування схем БД чи CASE-підсистеми. Для забезпечення доступу до інших БД чи до даних SQL-серверів повнофункціональні СКБД мають факультативні модулі.

Сервери БД призначені для організації центрів обробки даних у мережах ЕОМ. Ця група БД у даний час менш численна, але їхня кількість поступова росте. Сервери БД реалізують функції керування базами даних, запитувані іншими (клієнтськими) програмами, звичайно за допомогою операторів SQL.

Прикладами серверів БД є наступні програми: NetWare SQL (Novell), MS SQL Server (Microsoft), InterBase (Borland), SQLBase Server (Gupta), IntelligentDatabase (Ingress).

У ролі **клієнтських програм** для серверів БД у загальному випадку можуть використовуватися різні програми: ПФСКБД, електронні таблиці, текстові процесори, програми електронної пошти і т.п. При цьому елементи пари «клієнт-сервер» можуть належати одному чи різним виробникам програмного забезпечення.

У випадку, коли клієнтська і серверна частини виконані однією фірмою, природно очікувати, що розподіл функцій між ними виконаний раціонально. В інших випадках звичайно переслідується мета забезпечення доступу до даних «за будь-яку ціну». Прикладом такого з'єднання є випадок, коли одна з повнофункціональних СКБД відіграє роль сервера, а друга СКБД (іншого виробника) – роль клієнта. Так, для сервера БД SQL Server (Microsoft) у ролі клієнтських (фронтальних) програм можуть виступати більшість СКБД, такі як dBASE IV, BiythSoftware, Paradox, DataEase, Focus, 1-2-3, MDBS III, Revelation і інші.

Засоби розробки програм роботи з БД можуть використовуватися для створення наступних програм:

- клієнтських програм;
- серверів БД і їхніх окремих компонентів;
- користувацьких додатків.

Програми першого і другого виду досить нечисленні, тому що призначені, головним чином, для системних програмістів. Пакетів третього виду набагато більше, але все ж менше, ніж повнофункціональних СКБД.

До засобів розробки користувацьких додатків відносяться системи програмування, наприклад Clipper, різноманітні бібліотеки програм для різних мов програмування, а також пакети автоматизації розробок (у тому числі систем типу клієнт-сервер). У числі найбільш розповсюджених можна назвати наступні інструментальні системи: Delphi і PowerBuilder (Borland), VisualBasic (Microsoft), SILVERRUN (ComputerAdvisersInc.), S-Designer (SDP і Powersoft) і ERwin (LogicWorks).

Крім перерахованих засобів, для керування даними й організації обслуговування БД використовуються різні додаткові засоби, наприклад, монітори транзакцій

Класифікація СКБД за моделлю даних.

По використовуваній моделі даних СКБД (як і БД), розділяють на ієрархічні, мережеві, реляційні, об'єктно-орієнтовані й інші типи. Деякі СКБД можуть одночасно підтримувати кілька моделей даних.

3.1.5. Функції СКБД

З погляду користувача, СКБД реалізує функції збереження, зміни (поповнення, редагування і видалення) і обробки інформації, а також розробки й одержання різних вихідних документів. Для роботи з інформацією, що зберігається в базі даних, СКБД надає програмам і користувачам наступні два типи мов:

- мова опису даних – високорівнева непроцедурна мова декларативного типу, призначена для опису логічної структури даних;
- мова маніпулювання даними – сукупність конструкцій, що забезпечують виконання основних операцій по роботі з даними:

- введення,
- модифікація,
- вибірка даних по запитах.

Названі мови в різних СКБД можуть мати відмінності. Найбільше поширення одержали дві стандартизовані мови: QBE (QueryByExample) – мова запитів за зразком і SQL (StructuredQueryLanguage) – структурована мова запитів.

QBE в основному має властивості мови маніпулювання даними, SQL поєднує у собі властивості мов обох типів – опису і маніпулювання даними.

Перераховані вище функції СКБД, у свою чергу, використовують наступні основні функції більш низького рівня, що назвемо низькорівневими:

- керування даними в зовнішній пам'яті;
- керування буферами оперативної пам'яті;
- керування транзакціями;
- ведення журналу змін у БД;
- забезпечення цілісності і безпеки БД.

Дамо коротку характеристику необхідності й особливостям реалізації перерахованих функцій у сучасних СКБД.

Керування даними в зовнішній пам'яті. Реалізація функції керування даними в зовнішній пам'яті в різних системах може розрізнятися:

- на рівні керування ресурсами: використовуючи файлові системи ОС чи безпосереднє керування пристроями ПЕОМ;

- за логікою самих алгоритмів керування даними: в основному методи й алгоритми керування даними є «внутрішньою справою» СКБД і прямого відношення до користувача не мають. Якість реалізації цієї функції найбільш сильно впливає на ефективність роботи специфічних ІС, наприклад, з величезними БД, зі складними запитамі, великим обсягом обробки даних.

Керування буферами оперативної пам'яті. Необхідність буферизації даних і як наслідок реалізації функції керування буферами оперативної пам'яті обумовлене тим, що обсяг оперативної пам'яті менше обсягу зовнішньої пам'яті.

Буфери являють собою області оперативної пам'яті, призначені для прискорення обміну між зовнішньою й оперативною пам'яттю. У буферах тимчасово зберігаються фрагменти БД, дані з яких передбачається використовувати при звертанні до СКБД чи планується записати в базу після обробки.

Керування транзакціями. Механізм транзакцій використовується в СКБД для підтримки цілісності даних у базі.

Транзакція – послідовність операцій модифікації даних у БД, що переводить БД із одного несуперечливого стану в інший несуперечливий стан. Тобто транзакцією називається деяка неподільна послідовність операцій над даними БД, що відслідковується СКБД від початку і до завершення. Якщо за певними причинами (збої і відмови устаткування, помилки в програмному забезпеченні, включаючи додатки) транзакція залишається незавершеною, то вона скасовується.

Транзакція має три основних властивості:

- атомарність (виконуються всі включені в транзакцію операції або жодна);
- серіалізованість (відсутній взаємний вплив виконуваних у той самий час транзакцій);
- довговічність (навіть крах системи не приводить до втрати результатів зафіксованої транзакції).

Прикладом транзакції є операція переведення грошей з одного рахунку на інший у банківській системі. Тут необхідний, принаймні, двокроковий процес. Спочатку знімають гроші з одного рахунку, потім додають їх до іншого рахунку. Якщо хоча б одна з дій не виконається успішно, результат операції виявиться невірним і буде порушений баланс між рахунками.

Контроль транзакцій важливий в однокористувацьких і в багатокористувацьких СКБД, де транзакції можуть бути запущені паралельно. В останньому випадку говорять про серіалізованість транзакцій. Під серіалізацією паралельно виконуваних транзакцій розуміється складання такого плану їхнього виконання (серіального плану), при якому сумарний ефект реалізації транзакцій еквівалентний ефекту їхнього послідовного виконання.

При паралельному виконанні набору транзакцій можливе виникнення конфліктів (блокувань), розв'язання яких є функцією СКБД. При виявленні таких випадків звичайно виконується «відкат» шляхом скасування змін, зроблених однією чи декількома транзакціями.

Ведення журналу змін у БД (журналізація змін) виконується СКБД для забезпечення надійності збереження даних у базі при наявності апаратних збоїв і відмовлень, а також помилок у програмному забезпеченні.

Журнал СКБД – це особлива БД чи частина основної БД, безпосередньо недоступна користувачу і використовувана для запису інформації про всі зміни бази даних. У різних СКБД у журнал можуть заноситися записи, що відповідають змінам у СКБД на різних рівнях: від мінімальної внутрішньої операції модифікації сторінки зовнішньої пам'яті до логічної операції модифікації БД (наприклад, вставки запису, видалення стовпця, зміни значення в полі) – і навіть транзакції.

Для ефективною реалізації функції ведення журналу змін у БД необхідно забезпечити підвищену надійність збереження і підтримки в робочому стані самого журналу. Іноді для цього в системі зберігають кілька копій журналу.

Забезпечення цілісності і безпеки БД. Забезпечення цілісності БД складає необхідну умову успішного функціонування БД, особливо для випадку використання БД у мережі.

Цілісність БД є властивістю бази даних, яка означає, що в ній міститься повна, несуперечлива й адекватно відбиваюча предметну область інформація. Підтримка цілісності БД включає перевірку цілісності БД і її відновлення у випадку виявлення протиріч у базі даних. Цілісний стан БД описується за допомогою обмежень цілісності у виді умов, яким повинні задовольняти збережені в базі дані. Прикладом таких умов може служити обмеження діапазонів можливих значень атрибутів об'єктів, зведення про які зберігаються в БД, або відсутність повторюваних записів у таблицях реляційних БД.

Забезпечення безпеки досягається в СКБД шифруванням прикладних програм, даних, захисту паролем, підтримкою рівнів доступу до бази/даних і до окремих її елементів (таблиць, форм, звітів і т.п.).

Контрольні питання до підрозділу 3.1

- 1. Зробіть порівняльний аналіз понять «дані» та «інформація».*
- 2. Зробіть порівняльний аналіз понять «банк даних» та «база даних».*
- 3. Надайте визначення поняттю «система керування базами даних»*
- 4. Що визначає поняття «транзакція» стосовно баз даних?*
- 5. У чому полягає призначення рівню зовнішніх моделей при організації СКБД за моделлю ANSI?*
- 6. У чому полягає логічна незалежність при роботі з даними в архітектурі ANSI ?*
- 7. У чому полягає фізична незалежність при роботі з даними в архітектурі ANSI ?*

8. Перелічить етапи здійснення запиту користувача при роботі з СКБД?
9. Які дані зберігає база метаданих ?
10. Перелічить стадії розвитку банка даних. Яких фахівців має містити група адміністратора баз даних ?
11. Яких фахівців має містити група адміністратора баз даних ?
12. У чому полягає призначення серверів баз даних?
13. Наведіть класифікацію СКБД за моделлю даних.
14. У чому полягає призначення буферизації при роботі з базами даних?
15. З якою метою в СКБД використовується механізм транзакції ?

3.2. Реляційна модель даних та нормалізація

В деяких випадках при ієрархічному і мережному представленні зростання бази даних може привести до порушення логічної організації даних. Такі ситуації виникають при появі нових користувачів, нових застосувань та видів запитів, при обліку інших логічних зв'язків між елементами даних. Недоліки ієрархічної і мережної моделей привели до появи нової, реляційної моделі даних, створеної Коддом у 1970 році. Реляційна модель була спробою спростити структуру бази даних і забезпечити незалежність представлення та опису даних від прикладних програм. У ній були відсутні явні покажчики на предків і нащадків, а всі дані були представлені у виді простих таблиць, розбитих на рядки і стовпці [30].

3.2.1. Реляційна структура даних

Реляційною називається база даних, у якій всі дані, доступні користувачу, організовані у вигляді таблиць, а всі операції над даними зводяться до операцій над цими таблицями. Для представлення реляційних баз даних розроблена формальна теорія баз даних, теоретичну основу якої складає алгебра та математична логіка. Основні взаємопов'язані поняття фізичного, спеціального прикладного та математичного рівнів побудови реляційної бази даних та їх взаємовідношення показані в таблиці 3.1, в рядках якої знаходяться еквівалентні поняття.

Таблиця 3.1

Основні поняття реляційної бази даних та їх взаємозв'язок

	Фізичний рівень	Спеціальний прикладний рівень	Математичний рівень
	Файл	Таблиця	Відношення
	Запис	Рядок	Кортеж
	Поле	Стовпець	Атрибут

У реляційній базі даних інформація організована у вигляді таблиць, розділених на рядки і стовпці, на перетині яких містяться значення даних. У кожній таблиці є унікальне ім'я, що описує її вміст. Масив значень, що можуть міститися в стовпці, називається доменом цього стовпця.

Така двохвимірنا таблиця в математиці отримала назву відношення (relations).

Таблиця зрозуміла оглядово і звична для людей. Оригінальність підходу Е.Ф.Кодда полягає в застосованні до відношень впорядкованої системи операцій, що дозволяє отримувати складені відношення з бази (виводити, обчислювати, проводити арифметичні операції). Використання відношень дозволяє ділити інформацію на таку, що зберігається постійно і таку, яка отримується в результаті визначених перетворень над постійною. Е.Ф.Кодд показав, що набір відношень (таблиць) може бути використаний для збереження даних про об'єкти реального світу і моделювання зв'язків між ними. Під таблицею будемо розуміти структуру заголовку даної таблиці плюс сукупність записів даних у відповідності із заголовком. Наприклад, для збереження інформації про об'єкти з назвою «студент» можна використати відношення «СТУДЕНТИ» (Таблиця 3.2), в якому властивості об'єктів розміщуються в стовпцях таблиці.

Таблиця 3.2
Приклад відношення «СТУДЕНТИ»

Прізвище І.Б.	Дата народження	Курс	Напрямок
Сидорчук Р.Д.	12.02.1990	5	Комп'ютерні науки
Петрук Д.В.	15.04.1992	3	Інформатика
Іванюк І.І.	18.11.1993	2	Прикладна математика
Федорук М.Ж.	31.11.1995	1	Комп'ютерні науки

У кожного стовпця в таблиці є своє ім'я (атрибут), що і служить заголовком стовпця. Список імен стовпців відношення називається схемою відношення. Усі стовпці в одній таблиці повинні мати унікальні імена, однак дозволяється привласнювати однакові імена стовпцям, розташованим в різних таблицях.

Таблиці не дозволяють проводити узгодження наступних трьох способів маніпулювання даними: впорядкування, групування по певних ознаках, доступ по дереву параметрів. Це пов'язано з тим, що в таблиці всі три способи маніпулювання жорстко закріплені: дані впорядковані по одному параметру і не впорядковані по іншому.

Стовпці таблиці упорядковані зліва направо, й їхній порядок визначається при формуванні таблиці. У будь-якій таблиці завжди є як мінімум один стовпець.

Як правило, не вказується максимально допустиме число стовпців у таблиці, однак майже у всіх комерційних СКБД ця межа існує і, як правило, складає приблизно 255 стовпців.

На відміну від стовпців, рядки таблиці (кортежі) не мають визначеного порядку. Це значить, що якщо послідовно виконати два однакових запити для відображення вмісту таблиці, то немає гарантії, що обидва рази рядки будуть перераховані в тому самому порядку.

Рядки таблиці утворюють данні різного формату і різного типу, тобто можна стверджувати, що рядки таблиці є кортежами.

У таблиці може міститися будь-яка кількість рядків. Цілком припустиме існування таблиці з нульовою кількістю рядків. Така таблиця називається порожньою. Порожня таблиця зберігає структуру, визначену її стовпцями, просто в ній не містяться дані. Стандарти реляційних баз даних не накладають обмежень на кількість рядків у таблиці, і в багатьох СКБД розмір таблиць обмежений лише вільним дисковим простором комп'ютера.

Як правило, в сучасних реляційних БД допускається збереження символьних, числових даних, бігових рядків, спеціалізованих числових даних (таких як «грошовий»), а також спеціальних «темпоральних» даних (дата, час, часовий інтервал).

Найменша одиниця даних реляційної моделі – це окреме атомарне (неподільне) для даної моделі значення даних. Так, в одній предметній області прізвище, ім'я і по-батькові можуть розглядатися як єдине значення, а в інший – як три різноманітних значення.

Відношення називаються еквівалентними, якщо вони відрізняються тільки порядком чергування атрибутів.

3.2.2. Типи таблиць і ключів в реляційних базах даних

Таблиці реляційної бази даних діляться на два класи: базові та розвідні.

Базова таблиця (каталоги предметної області). В реляційній базі даних базовою таблицею називається таблиця, яка включає один або декілька стовпців властивостей об'єкту і містить первинний ключ, що однозначно визначає цей об'єкт. Базова таблиця повинна містити первинний ключ. Базові таблиці часто називають первинними, оскільки вони мають первинний ключ.

Розвідні (проміжні) таблиці. Таблиця, що не є базовою, яка використовується для забезпечення зв'язків між іншими таблицями, називається проміжною таблицею. Ключові поля в проміжній таблиці повинні бути зовнішніми ключами, що зв'язані з первинними ключами базових таблиць. Проміжна таблиця може мати і інші атрибути, які функціонально залежать від цього зв'язку.

В реляційних базах даних використовуються спеціальні типи атрибутів – первинні (унікальні) та вторинні (зовнішні) ключі [30].

Первинний ключ. Первинний ключ складається з набору значень, які однозначно визначають рядок (запис) базової таблиці. Будь-якому значенню первинного ключа повинен відповідати один і тільки один рядок (запис) таблиці. Первинний ключ, як правило, знаходиться у першому стовпці (№п/п). Інші атрибути функціонально залежать від даного ключа. Ключ може включати кілька атрибутів (складний ключ). Значення первинного ключа не по-

винні дублюватися. Це основне обмеження в реляційній базі даних для збереження цілісності даних.

Ключі-кандидати. Будь-який стовпець або група стовпців, які задовольняють вимогам, що накладаються на значення первинного ключа, є кандидатами на те, щоб стати первинним ключем.

Складені ключі. Якщо для виконання умов, що накладаються на значення первинного ключа, заданий ключ включає декілька полів таблиці, то тоді він називається складеним.

Вторинні (зовнішні) ключі. Зовнішній ключ – це стовпець, значення якого відповідають значенням первинного ключа з іншої зв’язаної таблиці. Він визначає посилання на конкретні записи інших таблиць, формуючи зв’язок між таблицями.

Умови і обмеження, які накладаються на відношення реляційних баз даних на табличному рівні представлення, можна сформулювати наступним чином:

- не може бути однакових первинних ключів, тобто всі рядки (записи) повинні бути унікальними;
- всі рядки повинні мати однакову типову структуру;
- імена стовпців в таблиці повинні бути різними, а значення стовпців повинні бути однотиповими;
- значення стовпців повинні бути атомарними, тобто не можуть бути компонентами інших відношень;
- повинна зберігатися цілісність для зовнішніх ключів;
- порядок розміщення рядків у таблиці неістотний – він впливає лише на швидкість доступу до потрібного рядка.

Інфологічні моделі даних. Діаграма, на якій показано логічне представлення даних, називається схемою. Існує багато методів створення схем даних. Одним з найбільш розповсюджених є структурний метод схем даних (діаграми Бахмана).

Графічний опис структури таблиць у формі полосок, які містять імена полів і показують спрощені відношення між даними, використовується для того, щоб користувачам було легше зрозуміти розроблену модель даних.

Іншим розповсюдженим підходом є метод, в якому використовується схема “Елемент – Відношення” (E-R), яка була розроблена Пітером Ченом в 1976р.. E-R схеми призначені для наглядного представлення відношень між об’єктами і поведінки елементів.

Елементи даних вказані у прямокутниках, атрибути даних – в овалах, а відношення між елементами – в ромбах. Відношення між об’єктами бази даних на концептуальному етапі можуть визначатися їх поведінкою.

Таким чином E-R схеми включають принаймні одне дієслово, об’єкт якого знаходиться справа від символу відношення. Символи наносяться на схему по мірі конкретизації моделі. Однією з переваг E-R схем є те, що їх можна використовувати для представлення на порівняно невеликому просторі концептуальної моделі великих схем з багатьма базами даних.

3.2.3. Типи відношень [30].

Відношення “один-до-одного”

Найпростішим відношенням між таблицями є відношення “один-до-одного”. В такому відношенні одному запису однієї таблиці відповідає тільки один запис у іншій. Таблиці, що зв’язані відношенням “один-до-одного” можна об’єднати в одну таблицю, яка складається з полів обох таблиць. Відношення “один-до-одного” часто використовують для розділення таблиць, що містять велику кількість полів. Наприклад, це може бути потрібним для того, щоб скоротити час перегляду полів, що містять певний набір даних. В деяких випадках необхідно керувати доступом до частин таблиць, які містять важливі або конфіденційні дані.

Відношення “один-до-багатьох” Відношення “один-до-багатьох” зв’язує один запис першої таблиці з декількома записами другої за допомогою первинного ключа базової таблиці і відповідного йому зовнішнього ключа зв’язаної таблиці. Зовнішній ключ таблиці, що містить велику кількість відношень, може входити до складеного первинного ключа, але він є зовнішнім по відношенню до базової таблиці. Відношення “один-до-багатьох” використовується найбільш часто.

Відношення “багато-до-одного” протилежно відношенню “один-до-багатьох”.

Якщо вибір відношення “багато-до-одного” або “один-до-багатьох” не має великої ролі, то відношення між таблицями називається рефлексивним. Відношення “багато-до-одного” є відображенням відношення “один-до-багатьох”. Всі відношення “багато-до-одного” в Access є рефлексивними. На E-R схемі рефлексивні відношення позначаються дієсловом у відповідній формі, який розміщується зовні ромба, що визначає відношення.

Відношення “багато-до-багатьох” У випадку, коли відсутнє забезпечення цілісності зв’язків між таблицями, можна говорити про відношення “багато-до-багатьох”. Оскільки забезпечення цілісності зв’язків між таблицями є базовою вимогою реляційної моделі даних, то відношення “багато-до-багатьох” не використовуються, а при зустрічі уникається введенням розвідної таблиці.0

3.2.4. Фізична організація файлів баз даних

Сучасні бази даних можуть представлятися сукупністю взаємопов’язаних таблиць. Тіло типового файлу бази даних для однієї таблиці містить заголовки таблиці та власне послідовно організований набір записів. В заголовку файлу міститься наступна інформація:

- 1) перелік полів з характеристиками (назва, тип, довжина поля);
- 2) час створення чи оновлення файлу;
- 3) кількість записів, що знаходяться у файлі,

4) інша допоміжна інформація (наприклад, інформація про зв'язок з індексним файлом, відомості про мову заповнення та інше).

Кожен запис таблиці складається із окремих полів (атрибутів), які діляться на три групи: ключові, вказівні та допоміжні. Ключовими є ті атрибути, якими однозначно ідентифікується даний запис. Двох записів з однаковими ключовими атрибутами бути не може. Вказівні атрибути виконують роль ключових атрибутів в інших базах даних, на які посилається даний запис. З їх допомогою можна отримати додаткову інформацію для заданого запису. Допоміжні атрибути – це характеристики для заданого запису і вони, як правило, можуть повторюватись.

Файл бази даних може бути індексованим або ні. Наявність індексації означає, що всі записи в файлі баз даних перевпорядковані у відповідності із наперед заданим принципом (алфавітний порядок, порядок зростання чи спадання і так далі) по одному з полів. Інформація про індексацію міститься в індексному файлі у вигляді набору взаємопов'язаних пар чисел. Система індексації застосовується для зручності використання бази даних: для заданих полів створюється впорядкований перелік пар чисел, перше з яких вказує фізичний реальний порядковий номер запису в файлі, друге – порядковий номер даного запису у перевпорядкованому списку. Розрізняють так звані одиничні і множинні індексні файли. Для кожного фізичного запису одиничний індексний файл містить тільки одну пару чисел, в той час як множинний – множинну з декількох впорядкованих пар чисел. Кожне друге з наступної пари чисел в наборі означає порядковий номер у новому списку. Характеристикою типу інформації даного атрибуту є тип поля. Найчастіше вживані наступні позначення типів полів, які наводяться в таблиці 3.3.

Таблиця 3.3
Типи полів файлів баз даних

№	Позначення	Тип поля
1	N	Числовий
2	F	Числовий з плаваючою комою
3	C	Символьний
4	D	Дати/час
5	M	Мемо-поле
6	L	Логічний

В деяких сучасних системах керування базами даних введено нові типи полів, але вони або надають нові можливості (наприклад тип графічного об'єкту) або просто розширюють наведені типи полів. Крім цього, для кожного числового чи символічного поля задається його довжина в символах. Так, для поля дати відведено 8 позицій, а для логічного – одне (логічний нуль –

“false” або одиниця – “true”). Поле мемо містить стандартної довжини вказівник на текстову інформацію. Ці вказівники містять адреси послідовно організованих записів в допоміжному файлі.

3.2.5. Тринадцять правил Кодда для реляційних СКБД

Кодд у 1985 році сформулював 13 правил (0..12), яким повинна задовольняти будь-яка СКБД, що претендує на тип реляційної. З того часу дванадцять правил Кодда вважаються визначенням реляційної СКБД [31].

Тринадцять правил Кодда, яким повинна відповідати реляційна СКБД, є наступними.

1. Фундаментальне правило (FoundationRule) Реляційна СКБД повинна бути здатна повністю управляти базою даних через її реляційні можливості.

1. Правило інформації (InformationRule) Вся інформація в базі даних повинна бути надана винятково на логічному рівні і тільки одним способом – у вигляді значень, що містяться в таблицях.

2. Правило гарантованого доступу (Guaranteed Access Rule) Логічний доступ до всіх і кожного елементу даних (атомарному значенню) у реляційній базі даних повинний забезпечуватися шляхом використання комбінації імені таблиці, первинного ключа та імені стовпця. Правило 2 вказує на роль первинних ключів при пошуку інформації в базі даних. Ім'я таблиці дозволяє знайти необхідну таблицю, ім'я стовпця дозволяє знайти необхідний стовпець, а первинний ключ дозволяє знайти рядок, який містить шуканий елемент даних.

3. Правило підтримки недійсних значень (SystematicTreatmentofNull Values)

У реляційній базі даних повинна бути реалізована підтримка недійсних значень, що відрізняються від рядка символів нульової довжини, рядка символів пробілів чи нуля будь-якого іншого числа і використовуватися для представлення відсутніх даних незалежно від типу цих даних. Правило 3 вимагає, щоб відсутні дані можна було представити за допомогою недійсних значень (NULL).

4. Правило динамічного каталогу, заснованого на реляційній моделі (DynamicOnlineCatalogBasedontheRelationalModel) Опис бази даних на логічному рівні необхідно представити в тому ж вигляді, що й основні дані, щоб користувачі, які володіють відповідними правами, могли працювати з ним за допомогою тієї ж реляційної мови, яку вони застосовують для роботи з основними даними. Правило 4 говорить, що реляційна база даних повинна сама себе описувати. Іншими словами, база даних повинна містити набір системних таблиць, що описують структуру самої бази даних.

5. Правило вичерпної підмови даних (ComprehensiveDataSublanguageRule) Реляційна система може підтримувати різні мови і режими взаємодії з користувачем (наприклад, режим питань і відповідей). Однак повинна існувати

принаймні одна мова, оператори якої підтримують наступні елементи: визначення даних; визначення представлень; обробку даних (інтерактивну і програмну); умови цілісності; ідентифікацію прав доступу; границі транзакцій (початок, завершення і скасування).

Правило 5 вимагає, щоб СКБД використовувала мову реляційної бази даних, наприклад SQL, хоча явно SQL у правилі не згадають. Така мова повинна підтримувати всі основні функції СКБД – створення бази даних, читання і введення даних, реалізацію захисту бази даних і т.д..

6. Правило відновлення представлень (ViewUpdatingRule) Усі представлення, які теоретично можна оновити, повинні бути доступні для відновлення. Правило 6 дозволяють показувати різним користувачам різні фрагменти структури бази даних.

7. Правило додавання, відновлення і вилучення (High-levelInsert./Update/Delete) Можливість працювати з відношенням як з одним операндом повинна існувати не тільки при читанні даних, але і при додаванні, відновленні і вилученні даних. Правило 7 акцентує увагу на тому, що бази даних по своїй природі орієнтовані на множини. Воно вимагає, щоб операції додавання, видалення і відновлення можна було виконувати над множинами рядків. Це правило призначене для того, щоб заборонити реалізації, у яких підтримуються тільки операції над одним рядком.

8. Правило незалежності фізичних даних (PhysicalDataIndependence) Прикладні програми й утиліти для роботи з даними повинні на логічному рівні залишатися недоторканими при будь-яких змінах методів збереження даних чи методів доступу до них.

9. Правило незалежності логічних даних (LogicalDataIndependence) Прикладні програми й утиліти для роботи з даними повинні на логічному рівні залишатися недоторканими при внесенні в базові таблиці будь-яких змін, які теоретично дозволяють зберегти недоторканими дані, що містяться в цих таблицях. Правила 8 і 9 стверджують, що конкретні способи реалізації чи збереження доступу, які використовуються в СКБД, і навіть зміни структури таблиць бази даних не повинні впливати на можливість користувача працювати з даними.

10. Правило незалежності умов цілісності (IntegrityIndependence)

Повинна існувати можливість визначати умови цілісності, специфічні для конкретної реляційної бази даних, на підмові реляційної бази даних і зберігати їх у каталозі, а не в прикладній програмі. Правило 10 вимагає, щоб мова бази даних підтримувала обмежувальні умови, які накладаються на дані, що вводяться, і дії, що можуть бути виконані над даними.

11. Правило незалежності поширення (DistributionIndependence) Реляційна СКБД не повинна залежати від потреб конкретного клієнта. Правило 11 говорить, що мова бази даних повинна забезпечувати можливість роботи з розподіленими даними, розташованими на інших комп'ютерних системах.

12. Правило одиницності (TheNonsubversionRule) Якщо в реляційній системі є низькорівнева мова (що обробляє один запис за один раз), то повинна

бути відсутня можливість використання її для того, щоб обійти правила й умови цілісності, виражені на реляційній мові високого рівня (що обробляє кілька записів за один раз).

Правило 12 запобігає використанню інших можливостей для роботи з базою даних, крім мови бази даних, оскільки це може порушити її цілісність.

Жодна з сучасних СУБД не дотримується цих правил від і до. Причина цього – складність правил.

Згідно Дейту реляційна модель складається з трьох частин, що описують різні аспекти реляційного підходу: структурної частини, маніпуляційної частини та і цілісної частини. У структурній частині моделі фіксується, що єдиною структурою даних, яка використовується в реляційних БД, є нормалізоване n -арне відношення. У маніпуляційній частині моделі стверджується два фундаментальних механізми маніпулювання реляційними БД – реляційна алгебра і реляційне числення. Перший механізм базується в основному на класичній теорії множин (з деякими уточненнями), а другий – на класичному логічному апараті числення предикатів першого порядку.

У цілісній частині реляційної моделі даних фіксуються дві базових вимоги цілісності, що повинні підтримуватися в будь-якій реляційній СКБД [28].

Перша вимога називається вимогою цілісності сутностей. Об'єкту або сутності реального світу в реляційній БД відповідають кортежі відношень. Конкретно вимога полягає в тому, що будь-який кортеж будь-якого відношення відрізняється від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення повинно мати первинний ключ.

Друга вимога називається вимогою цілісності по посиланнях.

Очевидно, що при дотриманні нормалізованості відношень складні сутності реального світу представляються в реляційній БД у виді декількох кортежів декількох відношень

Вимога цілісності по посиланнях, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, який з'являється у відношенні, до якого посилаються, повинний існувати кортеж із таким же значенням первинного ключа. Так, наприклад, якщо для співробітника зазначений номер відділу, то цей відділ повинен існувати.

Степінь відношення – це число його атрибутів. Відношення степеня один називають унарним, степеня два – бінарним, степеня три – тринарним, ... , а степеня n – n -арним.

Кардинальне число або потужність відношення – це число його кортежів. Кардинальне число відношення змінюється в часі на відміну від його степені.

До переваг реляційної бази можна віднести:

- незалежність від фізичного рівня представлення;
- зручність і розуміння організації даних користувачами;
- максимальна гнучкість при обробці непередбачених запитів;
- можливість розширення бази приєднанням нових елементів, записів без зміни при цьому існуючих підсхем та прикладних програм.

3.2.6. Нормалізація даних в реляційній моделі [32].

Експлуатаційні вимоги до реляційних баз даних

Практика розробки та експлуатації баз даних викристалізувала базові вимоги до їх побудови. Це зокрема вимоги:

- цілісності даних (обмеження; правила використання обмежень; правила обробки при порушенні обмежень цілісності; ефективність використання обмежень);
- узгодженості даних;
- відновлюваності даних (при збоях обладнання);
- безпеки (від несанкціонованих дій);
- ефективності (яка визначається такими параметрами, як швидкодія, мінімальний час доступу та мінімальна пам'ять).

Комерційні вимоги до реляційних баз даних

З іншої сторони, в середовищі прикладних користувачів-розробників були сформовані так звані бізнес-правила для роботи з базами даних. Їх суть полягає в ефективній реалізації роботи з даними, яка гарантує, що дані, які містяться в базах даних, повинні відповідати політиці й стратегії організації, зокрема і її правилам. Наприклад, для реалізації цього принципу потрібно встановити вимоги до кожного стовпця в таблиці баз даних так, щоб він не допускав значень, які суперечили б політиці організації. Фактично бізнес-правила вимагають виконання трьох основних функцій:

- зберігати небажані дані поза базами даних;
- однозначно описувати і строго визначати зв'язки між стовпцями і таблицями;
- при необхідності надавати інформацію звідки, коли і ким дані внесені в базу даних.

П'ять нормальних форм реляційних баз даних

В першу чергу, реляційні бази даних повинні задовольняти так званим умовам нормалізації. Процес трансформації даних в реляційну форму називається нормалізацією баз даних. Нормалізація – це видалення надлишкових даних з кожної таблиці бази даних. Процес нормалізації переслідує подвійну мету: видалити надлишкові копії даних і забезпечити максимальну гнучкість, як в структурах таблиць, так і в її інтерфейсних програмах на випадок можливих майбутніх змін баз даних. Найчастіше зустрічаються п'ять форм нормалізації, що означає п'ять різних установок реляційного критерію нормалізації баз даних.

1. Перша нормальна форма. Для першої нормальної форми потрібно, щоб таблиця була двовимірною і не містила груп, що повторюються. Вона не повинна містити комірок, що включають кілька значень (атомарність, неподільність.) Поле вважається неподільним, якщо воно містить тільки один елемент даних.

2. Друга нормальна форма. Перед перевіркою на відповідність другій нормальній формі таблиця повинна бути приведена до першої нормальної форми. Для другої нормальної форми потрібно, щоб дані у всіх не ключових стовпцях повністю залежали від первинного ключа. Під повною залежністю розуміється те, що значення в кожному не ключовому стовпці однозначно визначається значенням первинного ключа. Якщо одне з полів не залежить від величини первинного ключа, то необхідно включити в ключ доповнювальні таблиці. Друга нормальна форма дозволяє видалити більшу частину даних, що повторюються, які часто залишаються після першого етапу нормалізації.

3. Третя нормальна форма. Потрібно, щоб таблиці були попередньо приведені до першої та другої нормальної форми. Для третьої нормальної форми потрібно, щоб всі неключові стовпці таблиці не тільки залежали від первинного ключа таблиці, але й були незалежними один від одного, тобто, щоб були відсутні транзитивні функціональні залежності між стовпцями таблиці.

4. Четверта нормальна форма. Таблиця має задовольняти вимогам третьої нормальної форми і, крім того, забороняється зберігати незалежні елементи в одній і тій же таблиці, коли між цими елементами існує зв'язок (зв'язки) багато до багатьох.

5. П'ята нормальна форма має місце таблиця задовольняє вимогам четвертої нормальної форми та існує можливість модернізації даних таблиці. Передбачається, що при розробці баз даних і особливо при проведенні нормалізації таблиць потрібно звертати велику увагу на те, щоб випадково не проігнорувати суттєву характеристику або параметри об'єкта. Нормалізація збільшує число відношень в базах даних і тим самим час обробки. Але за рахунок коректності і усунення дублювання відбувається прискорення виконання доступу до даних.

3.2.7. Деякі поняття алгебри відношень

Ефективність реляційної моделі бази даних визначається здатністю виконувати над відношеннями наступні операції алгебри відношень: об'єднання, перетин, різниця, декартовий добуток, ділення, проекція, вибір, з'єднання [33].

Ступінь відношення – це кількість атрибутів, які в нього входять (або кількість стовпців у таблиці).

Потужність відношення – кількість записів у таблиці відношень (або кількість рядків без заголовку в таблиці).

Операція об'єднання

Операція об'єднання проводиться над двома відношеннями. Результуюче відношення включає всі записи першого відношення і ті записи другого відношення, яких немає в першому. Наприклад:

Відношення 1:

Прізвище І.П.	Вік
Анататійчук Р.І.	30
Бас І.М.	25
Білань І.М.	24
Вільховська С.О.	32

Відношення 2:

Прізвище І.П.	Вік
Анататійчук Р.І.	30
Вільховська С.О.	32
Гамар О.М.	25

Відношення-результат:

Прізвище І.П.	Вік
Анататійчук Р.І.	30
Бас І.М.	25
Білань І.М.	24
Вільховська С.О.	32
Гамар О.М.	25

Операція перетину

Перетин виконується над двома відношеннями. Результуюче відношення містить тільки ті записи, які є одночасно в першому і другому відношеннях. Наприклад:

Відношення 1:

Прізвище
Гасюк У.І.
Дулін О.В.
Дробенко Ю.Г.

Відношення 2:

Предмет	Дата іспиту
ОБДЗ	9.01.95
МСШ	14.01.94

Відношення-результат:

Прізвище	Предмет	Дата іспиту
Гасюк У.І.	ОБДЗ	9.01.95
Гасюк У.І.	МСШ	14.01.94
Дулін О.В.	ОБДЗ	9.01.95
Дулін О.В.	МСШ	14.01.94
Дробенко Ю.Г.	ОБДЗ	9.01.95
Дробенко Ю.Г.	МСШ	14.01.94

Операція різниці

Операція різниці проводиться над двома відношеннями. Результуюче відношення містить ті записи першого відношення, яких немає в другому відношенні. Наприклад:

Відношення 1:

Прізвище І.П.	Вік
Анататійчук Р.І.	30
Бас І.М.	25
Білань І.М.	24
Вільховська С.О.	32

Відношення 2:

Прізвище І.П.	Вік
Анататійчук Р.І.	30
Вільховська С.О.	32
Гамар О.М.	25

Відношення-результат:

Прізвище І.П.	Вік
Бас І.М.	25
Білань І.М.	24
Гамар О.М.	25

Операція декартового добутку

Декартовий добуток виконується над двома відношеннями, степінь результуючого відношення дорівнює сумі степенів первинних відношень, а потужність рівна добутку їх потужностей. Результуюче відношення містять всі можливі комбінації в записі первинних відношень. Наприклад:

Відношення
1:

Прізвище
Гасюк У.І.
Дулін О.В.
Дробенко Ю.Г.

Відношення 2:

Предмет	Дата іспиту
ОБДЗ	9.01.95
МСШ	14.01.94

Відношення-результат:

Прізвище	Предмет	Дата іспиту
Гасюк У.І.	ОБДЗ	9.01.95
Гасюк У.І.	МСШ	14.01.94
Дулін О.В.	ОБДЗ	9.01.95
Дулін О.В.	МСШ	14.01.94
Дробенко Ю.Г.	ОБДЗ	9.01.95
Дробенко Ю.Г.	МСШ	14.01.94

Операція ділення

Операція ділення – відношення дільника повинно містити підмножину атрибутів відношення діленого. Результуюче відношення включає тільки ті записи декартового добутку результуючого відношення з дільником, які містяться в діленому. Крім того, результуюче відношення містить тільки ті відношення діленого, яких немає в дільнику. Наприклад:

Відношення 1:

Прізвище	Предмет	Дата іспиту
Гасюк У.І.	ОБДЗ	9.01.95
Гасюк У.І.	МСШ	14.01.94
Дулін О.В.	ОБДЗ	9.01.95
Дулін О.В.	МСШ	14.01.94
Дробенко Ю.Г.	ОБДЗ	9.01.95
Дробенко Ю.Г.	МСШ	14.01.94

Відношення 2:

Предмет	Дата іспиту
ОБДЗ	9.01.95
МСШ	14.01.94

Відношення-результат:

Прізвище
Гасюк У.І.
Дулін О.В.
Дробенко Ю.Г.

Операція проєкції

Операція проєкції виконується над одним відношенням. Результуюче відношення включає частину атрибутів вихідного, на які виконується проєкція. Наприклад, для відношення 1 знайдемо перелік посад для кожного відділу.

Відношення 1:

Прізвище	Номер відділу	Посада
Ткаченко О.В.	1	Інженер
Холод Н.Б.	1	Інженер
Рапів І.М.	2	Інженер
Ковальчук С.Т.	2	Технік

Відношення-результат:

Номер відділу	Посада
1	Інженер
2	Інженер
2	Технік

Контрольні питання до підрозділу 3.2

1. Які особливості має реляційна база даних ?
2. Які обмеження покладає реляційність на групу пов'язаних таблиць?
3. Визначте поняття «первинний ключ» у реляційних базах,
4. Визначте поняття «ключ-кандидат» у реляційних базах,
5. У чому полягає сутність інфологічної моделі даних?
6. Поясніть принцип відношень “один-до-одного” у реляційних базах.
7. Поясніть принцип відношень “один-до-багатьох” у реляційних базах.
8. Поясніть зміст поняття «індексований файл» бази даних.
9. Поясніть правило «гарантованого доступу» з системи правил Кодда.
10. Поясніть правило «динамічного каталогу» з системи правил Кодда.
11. Поясніть відмінність вимоги цілісності сутностей та вимоги цілісності по посиланням при роботі з базами даних.
12. Що відноситься до переваг реляційної бази даних?
13. У чому полягають експлуатаційні вимоги до реляційних баз даних ?
14. У чому полягають комерційні вимоги до реляційних баз даних ?
15. Охарактеризуйте п'ять нормальних форм реляційних баз даних.

3.3. Загальна характеристика баз знань

3.3.1. Базові поняття щодо баз знань

Знання пов'язані з даними, базуються на них, але представляють результат розумової діяльності людини, узагальнюють його досвід, отриманий в ході виконання якої-небудь практичної діяльності. Вони є результатом емпіричного досвіду. Знання - це виявлені закономірності предметної області (принципи, зв'язки, закони), що дозволяють вирішувати завдання в цій галузі [34]. При обробці на ЕОМ знання трансформуються аналогічно даним:

- знання в пам'яті людини як результат мислення
- матеріальні знання (підручники, методичні посібники);
- поле знань умовне опис основних об'єктів наочної області, їх атрибутів і закономірностей, які їх пов'язують;
- знання описані на мовах подання знань (продукційні мови, семантичні мережі, фрейми);
- бази знань

Часто використовуються такі визначення знань: знання - це добре структуровані дані, або дані про дані, або метадані.

Існує безліч образів визначати поняття. Один з широко вживаних образів, заснований на ідеї інтенціонал. Інтенціонал поняття - це визначення через поняття більш високого рівня абстракції із зазначенням специфічних властивостей. Цей образ визначає знання. Інший спосіб визначає поняття через перерахування понять нижчого рівня ієрархії або фактів, які відносяться до визначеного. Це є визначення через дані, або екстенціонал поняття.

Знання можуть бути класифіковані за такими категоріями:

- поверхневі - знання про видимі взаємозв'язки між окремими подіями і фактами у наочній області;
- глибинні - абстракції, аналогії, схеми, які відображають структуру і процеси у наочній області.

Знання можна розділити на процедурні та декларативні.

- Декларативні знання – це знання, які записані в пам'яті інтелектуальної системи так, що вони безпосередньо доступні для використання після звернення до відповідного поля пам'яті. Зазвичай декларативні знання використовуються для подання інформації про властивості та факти предметної області. За формою подання декларативні знання протиставляються процедурним знанням.

- Процедурні знання – це знання, що зберігаються в пам'яті інтелектуальної системи у вигляді описів процедур, за допомогою яких їх можна отримати. Зазвичай процедурні знання використовуються для представлення інформації про способи вирішення завдань в проблемній області. Це також різні інструкції, методики тощо.

Історично первинними були процедурні знання, тобто знання, "розкриті" в алгоритмах. Вони управляли даними. Для їх зміни необхідно було міняти

програми. Однак з розвитком штучного інтелекту пріоритет даних поступово змінювався, і все більша частина знань зосереджувалася в структурах даних (таблиці, списки, абстрактні типи даних), тобто збільшувалася роль декларативних знань.

Сьогодні знання придбали чисто декларативну форму, тобто знаннями вважають пропозиції, записані на мовах подання знань, наближених до природних і зрозумілим неспеціалістам. Існують десятки моделей (або мов) подання знань для різних предметних областей. Більшість з них може бути зведені до наступних класів: продукційні; семантичні мережі; фрейми; формальні логічні моделі та багато інших.

Ці та інші моделі ми розглянемо в наступній лекції розділу

3.3.2. Стратегії отримання знань

Існує кілька стратегій отримання знань [35]. Найбільш поширені – такі:

- придбання;
- витягування;
- формування.

Під **придбанням знань** розуміється спосіб автоматизованого побудови бази знань за допомогою діалогу експерта і спеціальної програми (при цьому структура знань заздалегідь закладається в програму). Ця стратегія потребує суттєвого попередньої обробки предметної області. Системи набуття знань дійсно купують готові фрагменти знань згідно структурам, закладеним розробникам систем. Більшість цих інструментальних засобів орієнтована на конкретні експертні системи з жорстко визначеною предметною областю і моделлю представлення знань, тобто не є універсальними.

Термін **витягування знань** стосується безпосереднього живого контакту інженера по знаннях і джерела знань. Автори схильні використовувати цей термін як більш ємний і такий, що більш точно відображає суть процедури перенесення компетентності експерта через інженера по знаннях в базу знань експертної системи.

Термін **формування знань** традиційно закріпився за надзвичайно перспективною галуззю інженерії знань, яка займається розробкою моделей, методів і алгоритмів аналізу даних для отримання знань і навчання, які активно розвиваються. Ця область включає індуктивні моделі формування гіпотез на основі повчальних вибірок, вчення аналогічно і інші методи.

3.3.3. Висновки на базі знань

База знань – сукупність знань, які відносяться до деякої предметної області, формально представленим так, щоб на них основі можна було здійснювати роздуми. Бази знань найчастіше використовуються в контексті експертних систем, де з їх допомогою подаються навички та досвід експертів, зайнятих практичною діяльністю у відповідній галузі (наприклад, в медицині або в математиці). Зазвичай база знань є сукупністю правил виведення [36].

Не дивлячись на всі недоліки, найбільше поширення набула продукційна модель. При використанні продукційної моделі база знань складається з

набору правил. Програма, яка управляє перебором правил, називається машиною виведення. Машина виведення (інтерпретатор правил) виконує дві функції: по-перше, перегляд існуючих фактів з робочої пам'яті (бази даних) і правил з бази знань і додавання (в міру можливості) в робочу пам'ять нових фактів, а по-друге, визначення порядку перегляду і вживання правил. Цей механізм управляє процесом консультації, зберігаючи для користувача інформацію про отримані висновки, і запрошує у нього інформацію, коли для спрацювання чергового правила в робочій пам'яті виявляється недостатньо даних.

У переважній більшості систем, заснованих на знаннях, механізм виведення є невеликою за обсягом програмою і включає два компонента: один реалізує власно висновок, інший керує цим процесом. Дія компонента виведення ґрунтовано на вживанні правила, яке називається *modusponens*.

Правило *modusponens*. Якщо відомо, що достеменно твердження А, то існує правило виду - «Якщо А, то В», тоді твердження У той же істинно. Правила спрацьовують, коли знаходяться факти, які задовольняють їх лівій частині: якщо посилання істинна, то повинен бути справжній і висновок. Компонент виведення повинен функціонувати навіть при нестачі інформації. Отримане рішення може і не бути точним, проте, система не повинна зупинятися через те, що відсутня будь-яка частина вхідної інформації. Компонент, який управляє, визначає порядок вживання правил і виконує чотири функції:

1. Зіставлення зразок правила зіставляється з наявними фактами.
2. Вибір якщо в конкретній ситуації може бути застосовано відразу кілька правил, то з них вибирається одне, найбільш відповідне по заданому критерію (рішення конфлікту).
3. Спрацювання якщо зразок правила при зіставленні збігся з якимись фактами з робочої пам'яті, то правило спрацьовує.
4. Дія робоча пам'ять піддається зміні шляхом додавання в неї виведення спрацьованого правила. Якщо в правій частині правила міститься вказівка на будь-яку дію, то воно виконується (як, наприклад, в системах забезпечення безпеки інформації).

Інтерпретатор продукції працює циклічно. У кожному циклі він переглядає всі правила, щоб вказати ті, посилання яких збігаються з відомими на даний момент фактами з робочої пам'яті. Після вибору правило спрацьовує, його висновок заноситься в робочу пам'ять, і потім цикл повторюється спочатку. В одному циклі може спрацювати лише одне правило. Якщо кілька правил успішно зіставлені з фактами, то інтерпретатор виконує вибір за певним критерієм єдиного правила, яке спрацьовує в даному циклі. Цикл роботи інтерпретатора схематично наведений на рис. 3.1.

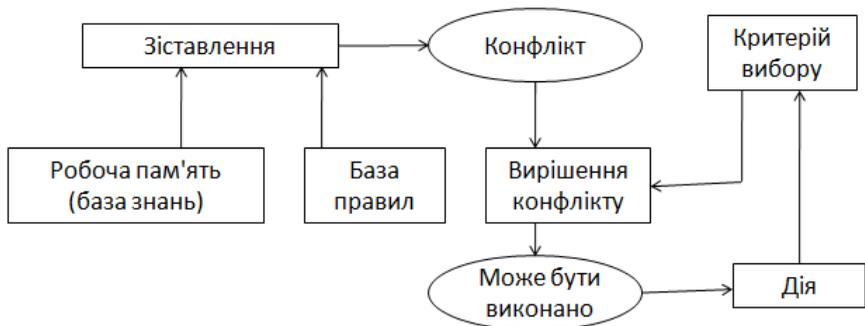


Рис.3.1. Цикл роботи інтерпретатора

Інформація з робочої пам'яті послідовно зіставляється з посиланнями правил для виявлення успішного зіставлення. Сукупність відібраних правил складає так зване конфліктне безліч. Для вирішення конфлікту інтерпретатор має критерій, за допомогою якого він обирає єдине правило, після чого правило спрацьовує. Це виражається в занесенні фактів, які створюють висновок правила, в робочу пам'ять або в зміні критерію вибору конфліктуючих правил. Якщо ж по закінченню правила згадується назва якого-небудь дії, то воно виконується. Робота машини виведення залежить лише від стану робочої пам'яті і від стану бази знань. На практиці зазвичай враховується історія обробки, тобто поведінку механізму випадку в попередніх циклах. Інформація про поведінку механізму виведення запам'ятовується в пам'яті станів. Зазвичай пам'ять станів містить протокол системи.

Від обраного методу пошуку, тобто стратегії виведення, буде залежати порядок вживання і спрацьовування правил. Процедура вибору зводиться до визначення напрямку пошуку і способу його здійснення. Процедури, які реалізують пошук, зазвичай закладені в механізм виведення, тому в більшості систем інженери знань не мають до них доступу і, отже, не можуть у них нічого міняти за власним бажанням. При розробці стратегії управління виводом важливо визначити два питання:

1. Яку точку в просторі станів прийняти як початкову? Від вибору цієї точки залежить і метод здійснення пошуку - в прямому або зворотному напрямку.
2. Якими методами можна підвищити ефективність пошуку рішення?

Ці методи визначаються обраною стратегією перебору – глибиною, шириною, по підзадачам або ін. При зворотному порядку виведення спочатку висувається деяка гіпотеза, а потім механізм виведення як би повертається назад, переходячи до фактів, намагаючись знайти ті, які підтверджують гіпотезу (рис. 11.2, зліва). Якщо вона виявилася правильною, то вибирається наступна гіпотеза, яка деталізує першу і є по відношенню до неї підцилі. Далі

відшуковують факти, що підтверджують істинність підпорядкованої гіпотези. Висновок такого типу називається керованим цілями, або керованим консеквента. Зворотний пошук застосовується в тих випадках, коли цілі відомі і їх порівняно небагато.

3.3.4. Елементи експертних систем

Експертна система - це комплекс комп'ютерного програмного забезпечення, яке допомагає людині приймати обґрунтовані рішення [37; 38]. Експертні системи використовують інформацію, отриману заздалегідь від експертів-людей, які в якій-небудь області є кращими фахівцями. Всі експертні системи містять, принаймні, три основні елементи: базу знань, машину висновку й інтерфейс користувача (рис. 3.2).



Рис. 3.2. Елементи експертної системи

База знань. База знань містить відомі факти, виражені у вигляді об'єктів, атрибутів і умов. Крім описових уявлень про дійсність, вона включає вираження невизначеності - обмеження на достовірність факту. В цьому відношенні вона відрізняється від традиційної бази даних внаслідок свого символічного, а не числового або літерного вмісту. При обробці інформації бази даних користуються заздалегідь певних логічних правил. Відповідно, база знань, яка представляє вищий рівень абстракції, має справу з класами об'єктів, а не з самими об'єктами. База знань створюється людьми - консультантами.

Машина виведення. Головним в експертній системі є механізм, який здійснює пошук в базі знань з правил раціональної логіки для отримання рішень [39; 40]. Ця машина виведення наводиться в дію при отриманні запиту користувача і виконує такі завдання:

- порівнює інформацію, яка міститься в запиті користувача, з інформацією бази знань;
- шукає певну мету або причинні зв'язку;

- оцінює відносну визначеність фактів, спираючись на відповідні коефіцієнти довіри, пов'язані з фактом.

Як випливає з її назви, машина виведення призначена для побудови висновків. Її дія аналогічно міркуванням експерта-людини, який оцінює проблему і пропонує гіпотетичні рішення. У пошуку цілей на основі запропонованих правил, машина виведення звертається до бази знань до тих пір, поки не знайде достовірний шлях до отримання прийнятного результату.

Інтерфейс користувача. Завдання інтерфейсу користувача полягає в організації обміну інформацією між оператором і машиною виведення. Інтерфейс з використанням природної мови створює видимість довільної бесіди, застосовуючи повсякденні вирази в правильно побудованих пропозиціях. Очевидно, що чим природніше такий інтерфейс, тим вище вимоги до зовнішньої і оперативної пам'яті.

Отже, системи, які надають користувачеві максимум зручностей, витрачають більше ресурсів основної машини, доступних з віддалених робочих станцій. Інструментальні засоби, призначені для роботи на персональних комп'ютерах, які мають обмежені можливості, неминуче приносять "дружність" в жертву ефективності.

Інтерфейс прямого введення повинен вміти розпізнавати мову, або, по крайній мірі, достатню кількість ключових слів і фраз, щоб вловлювати їх зв'язок з даною проблемою і її пропонованими рішеннями.

Аспект людський. У роботі з експертними системами беруть участь як мінімум три групи людей. По-перше, адміністрація встановлює призначення експертної системи, обмежує предметну область, яку повинна охоплювати система, і точно визначає, які вигоди організація зможе отримувати з її використання. По-друге, фахівець з збору знань (інженер - когнітологією) збирає інформацію, необхідну для бази знань, порівнює відповідні дані і евристичний організовує інформацію. По-третє, потенційний користувач вказує, як буде використовуватися система, якого роду проблеми доведеться вирішувати, і яким чином буде здійснюватися взаємодія програми з оператором. І, врешті-решт, системі потрібен експерт (частіше група експертів) в певній предметній встановленій наочній області, для отримання від нього знань, як в формі фактичної інформації, так і щодо аналітичних методів, які застосовуються для вирішення проблем в цій галузі.

Машинний аспект. Комп'ютерну частину системи представляють компоненти програмного забезпечення, які обробляють отриману інформацію про дійсність, закладеної в символічному вигляді в базу знань. У базу знань надходять факти. Зв'язок між фактами представлена евристичними правилами - виразами декларативного знання про відносини між об'єктами. Кожне таке правило має складову "якщо і компонент" то "(висновок-дія), які визначають пряма або зворотна причинно-наслідковий зв'язок. Дійсні затвердження лише достовірні, іншими словами, міра їх визначеності не завжди абсолютна.

Кількісні коефіцієнти визначеності збільшують точність міркування експертних систем. Загальноприйнята схема полягає в тому, щоб варіювати рівень довіри від 0, що представляє мінімальну міру визначеності, до 100 - вища міра.

Контрольні питання до підрозділу 3.3

- 1. Дайте визначення поняття «знання».*
- 2. Поясніть відмінність баз даних і баз знань.*
- 3. Назвіть основні перетворення знань при машинній обробці?*
- 4. Поясніть поняття інтенціонал і екстенціонал поняття.*
- 5. Визначте призначення інтерпретатора стосовно баз знань.*
- 6. Наведіть класифікацію знань.*
- 7. Які способу отримання знань ви знаєте?*
- 8. Дайте визначення бази знань.*
- 9. Як ви розумієте правило *modus ponens*?*
- 10. Які методи виведення ви знаєте?*
- 11. Які стратегії управління виводу ви знаєте?*
- 12. Дайте визначення поняття «Експертна система».*
- 13. Укажіть склад експертної системи*
- 14. Визначте поняття «інтерфейс»*
- 15. Назовіть основні функції експертної системи?*

РОЗДІЛ 4. АНАЛІЗ ДАНИХ І ЗНАНЬ

4.1. Основні поняття аналізу даних і знань

Головна мета машинної обробки експериментальних або статистичних даних полягає в автоматичному виявленні прихованих в них закономірностей. Ці закономірності або знання дозволяють зрозуміти сутність досліджуваного процесу і, спираючись на наявні дані, передбачати нові факти.

У цьому розділі описуються методи обробки інформації, представленої в різній формі – у вигляді «даних», «знань», «структур» тощо. В основі аналізу всіх цих видів інформації лежать дві процедури: процедура виявлення закономірностей, що містяться у наданій інформації, і процедура використання виявлених закономірностей для передбачення значення однієї частини інформації за відомими значеннями іншої її частини. Але перш ніж переходити до опису цих процедур, потрібно пояснити сенс вживаних термінів, зокрема таких поширених, як дані та знання.

4.1.1. Чим відрізняються «дані» від «знань»?

Існують Специфічні особливості, які дають змогу відрізнити знання від даних [41].

Вхідна інформація, яку треба обробляти, частіше за все надається у вигляді числових таблиць (матриць), що складаються з *т*рядків та *n* стовпців. Рядки $a_1, a_2, \dots, a_i, \dots, a_m$ відображають інформацію про досліджувані об'єкти або явища, а стовпці $x_1, x_2, \dots, x_j, \dots, x_n$ відображають властивості (ознаки, характеристики) цих об'єктів або явищ. Природа об'єктів може бути будь-якою – це можуть бути фізичні тіла, живі організми, сигнали, окремі соціальні процеси, заводи, види спорту, родовища тощо. Зрозуміло, що набір ознак, що описують ці об'єкти, буде в кожному випадку своїм і повинен відображати їх найбільш важливі властивості.

На перетині *i*-го рядка та *j*-го стовпця вказується значення (b_{ij}) *j*-ї ознаки в *i*-го об'єкта. Такий факт (наприклад, що *i*-й будинок має висоту 12 м) вважаємо атомарною частиною даних про конкретний *i*-й об'єкт. Повні дані про *i*-й об'єкт містяться в сукупності всіх елементів *i*-го рядка. Інформація ж про всі задані властивості всіх досліджуваних об'єктів, записана в таблиці «об'єкт-властивість», і називається таблицею даних. Таким чином, дані представляють собою сукупність окремих конкретних фактів.

Нехай в таблиці даних представлені описи великої кількості житлових будинків, а нас цікавлять тільки три властивості цих будинків: з якого матеріалу вони побудовані, в який колір пофарбовані їх стіни і якої вони висоти. Після вивчення таблиці даних ми можемо виявити деякі закономірності. Наприклад, з'ясується, що все панельні будинки, пофарбовані в сірий колір,

мають висоту від 15 до 25 м, панельні зелені будинки – від 8 до 16 м, а цегляні, незалежно від кольору стін, мають висоту менше 10 м. Позначимо ознаку «вид будівельного матеріалу» через x_1 . Ця ознака приймає два зрозумілих значення: $x_1 = n$ (панель) або $x_1 = k$ (цегла). Ознака «колір стін» приймає значення: $x_2 = \text{сірий}$, $x_2 = \text{зелений}$ або $x_2 = \text{будь-який}$. Ознака x_3 «висота» може приймати будь-яке числове значення від нуля до 30 м. Тоді виявлені закономірності можна стисло записати у вигляді таких логічних висловлювань:

если ($x_1 = n$) и ($x_2 = \text{серый}$), то ($x_3 = 15 - 25$);

если ($x_1 = n$) и ($x_2 = \text{серый}$), то ($x_3 = 15 - 25$);

если ($x_1 = k$) и ($x_2 = \text{любой}$), то ($x_3 < 10$).

Ці висловлювання не містять інформації у вигляді конкретних характеристик кожного окремого будинку, але проте відображають наші знання про деякі узагальнені характеристики всіх будинків, описаних в таблиці даних.

Так виглядає перехід від даних до знань. Знання є коротким узагальненим описом основного змісту інформації, наданої у даних. Знання можуть бути представлені в різній формі. Надалі ми будемо користуватися наведеною вище формою у вигляді логічних правил типу «якщо ... то ...».

4.1.2. Що є аналізом даних?

Аналіз даних – розділ математики, що займається розробкою методів обробки даних незалежно від їх природи [42].

Аналіз даних включає виконання послідовних, логічних дій з інтерпретації зібраних даних (наприклад, відповідей респондентів) та їх перетворення на статистичні форми, потрібні для ухвалення маркетингових та керівницьких рішень.

Можна виділити такі етапи аналізу даних: отримання даних, обробка, аналіз та інтерпретація результатів обробки.

Аналіз даних можна вважати прикладним розділом математичної статистики, проте потрібно наголосити, що аналіз даних охоплює обробку як кількісних, так і якісних даних. Причому, не обов'язково використання імовірнісних моделей в описі досліджуваних об'єктів, явищ та процесів.

Якщо даних занадто багато, то ми маємо такі варіанти подальшої роботи:

1. Обробляючи тільки частину, а все інше залишити наступникам.
2. Аналізувати всі дані, і вибрати найінформативніші частини.
3. Групувати дані

Групування даних має на меті їх стиснення без суттєвої втрати вибіркою інформативності, та полегшити подальшу обробку.

Хай наприклад маємо набір результатів вимірювань величини

$$\xi: x_1, x_2, \dots, x_n.$$

Тепер знаходимо межі проміжку на якому вони знаходяться:

$$[x_{\min}, x_{\max}].$$

Розбиваємо цей проміжок на m рівних частин. Для кожного проміжку записуємо його середину, та кількість величин в ньому.

Також можна задати умову того, що в кожному проміжку має бути не менше k значень. В такому разі, той проміжок в якому їх менше, об'єднують з одним із сусідніх.

Серед завдань прикладної математики пояснимо місце того напрямку, який з подачі французьких математиків отримав назву аналізу даних.

Класичний напрямок прикладної математики пов'язаний з методами обчислень одних характеристик досліджуваного об'єкта або явища за відомими значеннями інших його характеристик. При цьому модель об'єкта вважається відомою, а залежності між характеристиками описуються аналітичним виразом у вигляді рівняння або системи рівнянь або нерівностей. Проблеми, що виникають при вирішенні таких завдань, пов'язані, наприклад, з великими обсягами обчислень, із захистом від похибок, що накопичуються в комп'ютері через округлення чисел.

Пізніше з'явилися завдання аналізу об'єктів, математична модель яких відома з точністю до параметрів. Відомий набір характеристик, що впливають на цільову характеристику, відомий також загальний вид залежності між характеристиками, але коефіцієнти, показники ступеня та інші параметри моделі невідомі, і, щоб їх визначити, використовуються протоколи спостережень, що відображають значення одних характеристик при різних значеннях інших. Робиться серія припущень про значення невідомих параметрів моделі і ці припущення перевіряються на протоколах. В результаті вибираються такі значення параметрів, при яких модель із заданою точністю дозволяє по одним (вхідним) характеристикам визначати інші (вихідні або цільові) характеристики. Такого роду завдання називаються завданнями ідентифікації моделей.

Нарешті, з появою кібернетики стали формулюватися завдання аналізу «чорної скриньки»: досліднику відомий набір характеристик, серед яких є характеристики, що впливають на цільову властивість об'єкта, але які з них є визначальними (інформативними) і якою математичною моделлю описуються закономірності їх впливу на цільову характеристику, не відомо. Потрібно вибрати інформативні характеристики і побудувати модель, що дозволяє обчислювати значення цільової характеристики за значеннями інших характеристик.

Єдиним джерелом інформації для вирішення такого завдання є таблиця експериментальних даних типу «стимул-реакція» з описом вхідних і вихідних характеристик об'єкта, що спостерігається, або множини об'єктів. Як ми бачили раніше, такі таблиці даних називають таблицями «об'єкт-властивість». Тепер вибір моделі і її параметрів робиться шляхом перевірки різних емпіричних гіпотез на матеріалі таблиці даних. Коло завдань, що виникає при цьому, і становить напрямок, іменованій завданнями аналізу даних.

Повертаючись до початку, можна відзначити, що обчислювальна математика зазвичай не має справи з етапом висунення гіпотез про те, які характеристики повинні включатися в модель об'єкта і якою має бути ця модель. Ризик зробити помилку у виборі моделі і її параметрів залишається поза по-

лем уваги, але акуратні обчислення за наявною моделлю створюють враження високої якості вирішення проблеми в цілому.

Завдання ідентифікації моделей вимагають від математика відповідальності за правильний вибір параметрів моделі. Наявність цього ризикованого кроку в процесі виконання завдання позбавляє результат ореолу суворой математичної чистоти.

На результатах вирішення завдань аналізу даних лежить явний слід великого числа евристичних або експертних припущень – і про вибір характеристик об'єкта, і про клас моделей, і про параметри обраної моделі. Ці припущення надаються на мові математичних формул, але витoki їх появи лежать поза математики, так що основна частина процесу вирішення завдань аналізу даних пов'язана з проникненням у природу досліджуваного явища і характерна скоріше для природничо-наукових областей.

Ситуація погіршується ще й тим, що реальні дані мають такі особливості, які ускладнюють застосування суворих математичних методів. Досить зазначити, що таблиці даних часто бувають представлені малими вибірками у просторах великої розмірності при відсутності інформації про характер і ступінь залежності одних характеристик від інших, різнотипності вимірювальних шкал, наявності шумів і прогалин. У цих умовах методи вирішення завдань аналізу даних вимушено ґрунтуються як на коректних математичних процедурах, так і на чисто евристичних прийомах. Тож не дивно, що рішення, що отримуються, сприймаються насторожено, а багато методів вирішення виглядають недостатньо строго обґрунтованими.

Ця обставина об'єктивно відображає той факт, що на будь-якому етапі розвитку прикладної математики виникають реальні завдання, для вирішення яких добре обґрунтовані математичні методи ще не готові. Разом з тим важливість завдань не дозволяє відкласти їх рішення і змушує приймати ризиковані емпіричні гіпотези і використовувати несурові евристичні прийоми. Якщо одержувані при цьому результати (передбачення, прогнози) підтверджуються фактами, то настороженість в сприйнятті використаної моделі змінюється впевненістю в її адекватності досліджуваному явищу, а увагу математика переноситься на аналітичне дослідження моделі та обчислювальні труднощі, пов'язані з її використанням.

4.1.3. Що є аналізом знань?

Існують певні визначення філософського аспекту [43]. В задачах аналізу даних ми маємо справу з різними методами обробки таблиць даних. Останнім часом з'явилися методи, за допомогою яких можна обробляти і набори логічних висловлювань (знань), представлених у формі кон'юнкції «якщо ... то ...». При цьому на основі знань ставляться і вирішуються завдання, аналогічні тим, що ставляться і вирішуються на основі даних: виявлення закономірностей в масиві знань (тобто знань про знання) і використання цих закономірностей (мета знань) для передбачення одних частин знань за відомими значеннями їх інших частин.

Методи аналізу знань мають багато спільного з методами аналізу даних.

Зокрема, як і при аналізі даних, тут можна спиратися як на евристичні припущення про моделі закономірностей і параметри цих моделей, так і на окремі знання («прецеденти») з числа тих, що є в базі знань. З цієї причини значна частина успіху залежить від того, наскільки вдало вибрані припущення про моделі або прецеденти, наскільки представницьким є матеріал, що аналізується тощо.

Контрольні питання до підрозділу 4.1

1. Що є даними?
2. Перелічить основні етапи аналізу даних.
3. Виконання яких дій включає аналіз даних для перетворення їх на статистичні форми, потрібні для ухвалення маркетингових та керівницьких рішень?
4. Поясніть застосування концепції «чорної скриньки» для вивчення впливу входних параметрів об'єкта на вихідні?
5. Який зв'язок виникає між даними і знаннями при створенні останніх?
6. Що спільного з методами аналізу даних є у методів аналізу знань?

4.2. Розділи аналізу даних

Аналіз даних має свої методи, процеси, типи [44].

4.2.1. Попередня обробка

Попередня обробка – це розділ аналізу даних що займається отриманням характеристик для подальшого використання у наступних розділах аналізу даних.

1. Обчислення базових характеристик (центральні моменти)

У теорії ймовірностей та математичній статистиці, **центральный момент** k -го порядку випадкової величини з дійсними значеннями це величина:

$$M(X - MX)^k,$$

де M – математичне сподівання.

Для дискретного одновимірного розподілу з функцією центрального моменту порядку k відносно середнього v дорівнює:

$$\mu_k = \sum_i (x_i - v)^k p(x_i)$$

Дисперсія випадкової величини – це центральный момент другого порядку.

2. Перевірка основних гіпотез (симетричності, однорідності)
3. Перевірка стохастичності вибірки
4. Видалення аномальних спостережень
5. Розвідувальний аналіз (попередній експрес-аналіз даних шляхом їх перетворення та/або представлення у зручному вигляді: графічному, табличному, схем, діаграм і т.д.).

4.2.2. Кореляційний аналіз

Кореляційний аналіз – це статистичне дослідження (стохастичної) залежності між випадковими величинами (англ. *correlation* – взаємозв'язок). У найпростішому випадку досліджують дві вибірки (набори даних), у загальному – їх багатовимірні комплекси (групи) [45].

Мета кореляційного аналізу – виявити чи існує істотна залежність однієї змінної від інших.

Головні завдання кореляційного аналізу:

1. оцінка за вибірковими даними коефіцієнтів кореляції
2. перевірка значущості вибіркових коефіцієнтів кореляції або кореляційного відношення
3. оцінка близькості виявленого зв'язку до лінійного
4. побудова довірчого інтервалу для коефіцієнтів кореляції.

Обмеження кореляційного аналізу

Кореляція відображає лише лінійну залежність величин, але не відображає їх функціональної зв'язаності. Наприклад, якщо обчислити коефіцієнт кореляції між величинами

$$A = \sin(x) \text{ та } B = \cos(x),$$

він буде наближений до нуля, тобто залежність між величинами відсутня.

Між тим, величини A та B очевидно зв'язані між собою за законом

$$\sin^2(x) + \cos^2(x) = 1.$$

Використання можливе у випадку наявності достатньої кількості випадків для вивчення: для конкретного типу коефіцієнту кореляції становить від 25 до 100 пар спостережень.

Кореляція не означає причинність (!).

4.2.3. Дисперсійний аналіз

Дисперсійний аналіз (англ. *Analysis of variance (ANOVA)*) являє собою статистичний метод аналізу результатів, які залежать від якісних ознак [46]. Кожен фактор може бути дискретною чи неперервною випадковою змінною, яку розділяють на декілька сталих рівнів (градацій, інтервалів). Якщо кількість вимірювань (проб, даних) на всіх рівнях кожного з факторів однакова, то дисперсійний аналіз називають рівномірним, інакше – нерівномірним.

В основі дисперсійного аналізу є такий принцип (факт з математичної статистики): якщо на випадкову величину діють взаємно незалежні фактори A , B , ..., то загальна дисперсія дорівнює сумі дисперсій, зумовлених дією окремо кожного з факторів:

$$\sigma^2 = \sigma^2_A + \sigma^2_B + \dots$$

Задачі дисперсійного аналізу [47].

В будь-якому експерименті середні значення досліджуваних величин змінюються у зв'язку зі зміною основних факторів (кількісних та якісних), що визначають умови досліду, а також і випадкових факторів. Дослідження впливу тих чи інших факторів на мінливість середніх є задачею дисперсійного аналізу. Дисперсійний аналіз використовує властивість адитивності дисперсії випадкової величини, що обумовлено дією незалежних факторів. В залежності від числа джерел дисперсії розрізняють однофакторний та багатфакторний дисперсійний аналіз.

Дисперсійний аналіз особливо ефективний при вивченні кількох факторів. При класичному методі вивчення змінюють тільки один фактор, а решту залишають постійними. При цьому для кожного фактору проводиться своя серія спостережень, що не використовується при вивченні інших факторів. Крім того, при такому методі досліджень не вдається визначити взаємодію факторів при одночасній їх зміні. При дисперсійному аналізі кожне спостереження служить для одночасної оцінки всіх факторів та їх взаємодії. Дисперсійний аналіз полягає у виділенні й оцінюванні окремих факторів, що викликають зміну досліджуваної випадкової величини. При цьому проводиться розклад сумарної вибіркової дисперсії на складові, обумовлені незалежними факторами. Кожна з цих складових є оцінкою дисперсії генеральної сукупності. Щоб дати оцінку дієвості впливу даного фактору, необхідно оцінити значимість відповідної вибіркової дисперсії у порівнянні з дисперсією відтворення, обумовленою випадковими факторами. Перевірка значимості оцінок дисперсії проводять з допомогою критерію Фішера. Коли розрахункове значення критерію Фішера виявиться меншим табличного, то вплив досліджуваного фактору немає підстав вважати значимим. Коли ж розрахункове значення критерію Фішера виявиться більшим табличного, то цей фактор впливає на зміни середніх. В подальшому ми вважаємо, що виконуються наступні припущення:

1. Випадкові помилки спостережень мають нормальний розподіл.
2. Фактори впливають тільки на зміну середніх значень, а дисперсія спостережень залишається постійною.

Фактори, що розглядаються в дисперсійному аналізі, бувають трьох родів:

- з випадковими рівнями, коли вибір рівнів проходить з безмежної сукупності можливих рівнів та супроводжується рандомізацією і рівні вибираються випадковим чином;
- з фіксованими рівнями;
- змішаного типу – частина факторів розглядається на фіксованих рівнях, але рівні решти вибираються випадковим чином.

Дисперсійний аналіз застосовується в різних формах в залежності від структури об'єкту, що досліджується; вибір відповідної форми є однією з головних труднощів в практичному застосуванні аналізу. Дисперсійний аналіз використо-

вує властивість адитивності дисперсії випадкової величини, що обумовлено дією незалежних факторів. В залежності від числа джерел дисперсії розрізняють одnofакторний та багатofакторний дисперсійний аналіз.

4.2.4. Регресійний аналіз

Регресійний аналіз – це розділ математичної статистики, присвячений методам аналізу залежності однієї величини від іншої [48]. На відміну від кореляційного аналізу не з'ясовує чи істотний зв'язок, а займається пошуком моделі цього зв'язку, вираженої у функції регресії.

Регресійний аналіз використовується в тому випадку, якщо відношення між змінними можуть бути виражені кількісно у виді деякої комбінації цих змінних. Отримана комбінація використовується для передбачення значення, що може приймати цільова (залежна) змінна, яка обчислюється на заданому наборі значень вхідних (незалежних) змінних. У найпростішому випадку для цього використовуються стандартні статистичні методи, такі як лінійна регресія. На жаль, більшість реальних моделей не вкладаються в рамки лінійної регресії. Наприклад, розміри продажів чи фондові ціни дуже складні для передбачення, оскільки можуть залежати від комплексу взаємозв'язків множин змінних. Таким чином, необхідні комплексні методи для передбачення майбутніх значень.

Регресійні моделі

До регресійної моделі відносяться наступні параметри і змінні:

- Невідомі параметри, що позначаються як β , і які задають скаляри або вектор.
- Незалежні змінні, X .
- Залежна змінна, Y .

У різних сферах застосування, може використовуватися різна термінологія замість понять залежних та незалежних змінних.

Регресійна модель співвідносить Y до функції по X і β .

$$Y \approx f(X, \beta).$$

Наближення зазвичай формалізують як $E(Y|X) = f(X, \beta)$. Для проведення регресійного аналізу необхідно визначити форму функції f . Іноді форма функції основана на знаннях про взаємовідношення між X і Y і не покладається на дані. Якщо таких знань немає, обирають гнучку або зручну форму для визначення f .

Припустимо, що вектор невідомих параметрів β має довжину k . Для здійснення регресійного аналізу необхідно надати інформацію про залежну змінну Y :

- Якщо спостерігали N точок даних у формі (Y, X) , де $N < k$, більшість класичних методів регресійного аналізу не можна виконати: оскільки система рівнянь, що визначає регресійну модель є недовизначеною, в ній не достатньо даних для відновлення β .

- Якщо спостерігалось точно $N = k$ точок даних, а функція f є лінійною, рівняння $Y \approx f(X, \beta)$ можна розв'язати точно, а не наближено. Задача спрощується до вирішення множини з N рівнянь із N невідомими (елементами β , які мають унікальне рішення за умови що X лінійно незалежні. Якщо f не є лінійною, рішення може не існувати, або може існувати багато рішень.

- Найбільш типовою ситуацією є випадок, де спостерігають $N > k$ точок даних. В такому випадку, існує достатньо інформації з даних, для того щоб оцінити унікальні значення β , які в деякому розумінні підходять найкраще, а регресійна модель у застосування до даних може розглядатися як перевизначена система для β .

В останньому випадку, регресійний аналіз надає засоби для:

1. Знаходження рішення щодо невідомих параметрів β , які будуть, наприклад, мінімізувати відстань між вимірними і прогнозованими значеннями залежної змінної Y (що відомо як метод найменших квадратів).

2. При певних статистичних припущеннях, регресійний аналіз використовує надлишок інформації для надання статистичної інформації про невідомі параметри β і передбачені значення залежних змінних Y .

Необхідна кількість незалежних вимірювань.

Розглянемо регресійну модель, яка має три невідомі параметри, $\beta_0, \beta_1, \text{ і } \beta_2$. Припустимо, що експериментатор провів 10 вимірювань всі при точно однаковому значенні вектору незалежних змінних X (який містить незалежні змінні $X_0, X_1, \text{ і } X_2$). У такому випадку, регресійний аналіз не дозволяє визначити унікальний набір оцінених значень для трьох невідомих параметрів; експериментатор не надав достатньо інформації. Найкраще що можна зробити це оцінити середнє значення стандартного відхилення залежної змінної. Аналогічним чином, вимірювання при двох різних значеннях X дозволило б мати достатньо даних для регресії із двома невідомими, але не з трьома і більше невідомими.

Якщо експериментатор виконав вимірювання при трьох різних значеннях вектору незалежних змінних X , тоді регресійний аналіз поверне унікальний набір оцінок для трьох невідомих параметрів у β .

У випадку із загальною лінійною регресією, вищенаведене твердження є еквівалентним вимозі, що матриця X повинна бути невиродженою.

Коли кількість вимірювань, N , є більшим ніж кількість невідомих параметрів, k , а виміряні похибки σ_i мають нормальний розподіл тоді *надлишок інформації*, що міститься у $(N - k)$ вимірюваннях використовують, щоб зробити статистичне передбачення щодо невідомих параметрів. Цей надлишок інформації називають ступенями свободи регресії.

Задачі регресійного аналізу [49].

1. Визначення ступеня детермінованості варіації критеріальної (залежної) змінної предикторами (незалежними змінними)

2. Прогнозування значення залежної змінної за допомогою незалежної.

3. Визначення внеску окремих незалежних змінних у варіацію залежної.

Регресійний аналіз не можна використовувати для визначення наявності зв'язку між змінними, оскільки наявність такого зв'язку і є передумова для застосування аналізу.

Припущення

До класичних припущень, що лежать в основі регресійного аналізу, відносяться:

- Для здійснення прогнозування вибірка повинна бути репрезентативна для даної сукупності.

- Похибка є випадковою величиною із нульовим умовним математичним сподіванням для описових змінних.

- Незалежні змінні виміряні без похибки. (Примітка: Якщо це не так, замість цього можна використати техніки моделювання похибки величин).

- Незалежні змінні є лінійно незалежними, тобто. не можливо вирази будь-яку з цих змінних як лінійну комбінацію інших змінних.

- Похибки некорельовані, тобто, коваріаційна матриця похибок є діагональною, а кожен не нульовий елемент цієї матриці є дисперсією похибки.

- Дисперсія похибки є сталою поміж спостережень (гомоскедастичність). Якщо це не так, слід використовувати метод найменших квадратів або інші методи.

Виконання цих умов достатньо щоб оцінка найменших квадратів мала бажані властивості; зокрема, ці припущення означатимуть, що оцінки параметрів будуть незміщеними, конзистентними, ефективними у класі лінійних незміщених оцінок. Варто відмітити, що фактичні дані вимірювань дуже рідко задовольняють припущенням. Таким чином, метод використовують навіть коли припущення не правдиві. Відхилення від даних припущень, іноді використовують як міру того, наскільки корисною є дана модель. Багато з цих припущень можуть бути пом'якшені при більш ширшому трактуванні. У результатах статистичного аналізу як правило наводять результати тестів вибірових даних та методологію щодо відповідності і вживаності моделі.

4.2.5. Коваріаційний аналіз [50].

Коваріаційний аналіз – це розділ аналізу даних, що намагається визначити модель зв'язку між залежною величиною, та набором кількісних та якісних величин. Таким чином він є ніби синтезом регресійного та дисперсійного аналізів.

4.2.6. Дискримінантний аналіз [51].

Дискримінантний аналіз – це різновид багатовимірної аналізу, призначеного для вирішення задач розпізнавання образів. Використовується для прийняття рішення про те, які змінні розділюють (тобто «дискримінують») певні масиви даних (так звані «групи»).

Дискримінантний аналіз є близьким до дисперсійного і регресійного аналізів, які також намагаються виразити одну із залежних змінних у вигляді лінійної комбінації інших показників або вимірювань. Однак, у двох інших методів залежна змінна є числовий величиною, в той час як у дискримінантному аналізі це категорійна змінна. Більш подібними до дискримінантного аналізу є логістична і пробіт-регресія, оскільки вони також пояснюють категорійну змінну. Ці та інші методи використовуються переважно в тих випадках, коли не припускається нормальний розподіл незалежних змінних, що є основним припущенням методу дискримінантного аналізу.

Дискримінантний аналіз широко застосовується в економіці маркетингових досліджень при вирішенні питань сегментації ринку, при об'єктивній оцінці ступеня новизни товарів тощо.

У хемометриці – статистичний метод встановлення границь, що розділяють дані на певні класи (категорії), і знаходження набору відповідних дескрипторів, що відображають кожний клас.

4.2.7. Кластерний аналіз

Кластерний аналіз (англ. *Data clustering*) – це задача розбиття заданої вибірки об'єктів (ситуацій) на підмножини, які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися [52]. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя.

Кластерний аналіз – це не якийсь один алгоритм, а загальна задача, для розв'язання якої використовуються різні підходи. Зокрема, алгоритми побудови кластерів можуть суттєво відрізнятися у розумінні того, що відносити в один кластер і як їх ефективно шукати. Серед популярних концепцій кластерів є групи з елементами, які утворюються ґрунтуючись на відстані між ними, на щільності ділянок у просторі даних, інтервалах або на конкретних статистичних розподілах. Тому кластеризація може бути сформульована як задача багатокритеріальної оптимізації. Відповідний алгоритм кластеризації та вибору параметрів (включаючи такі параметри, як функція відстані, порогове значення щільності або кількість очікуваних кластерів) залежать від конкретного набору даних та мети використання результатів. Кластерний аналіз як такий є не автоматизованим завданням, а ітераційним процесом виявлення знань або інтерактивної багатокритеріальної оптимізації, який містить спроби та невдачі. Часто доводиться змінювати процес опрацювання даних та параметри моделі поки не буде отримано з результат з заданими властивостями.

Окрім терміну *кластеризація* існує багато термінів з аналогічним значенням, серед яких автоматична класифікація, числова таксономія та типологіч-

ний аналіз. Тонкі розбіжності часто полягають у використанні результатів: для добування даних, отримані групи є предметом інтересу, при автоматичній класифікації, навпаки, більш важливий степінь розбіжності.

Кластерний аналіз походить з антропології, де він був започаткований Драйвером (англ. *Driver*) і Крьобером (англ. *Kroeber*) у 1932 році. В психологію він був введений Зубіним у 1938 році і Робертом Тріоном у 1939. Став відомий завдяки використанню Кеттелем для класифікації теорії ознак в психології особистості, починаючи з 1943 року

Кластерний аналіз – це багатовимірна статистична процедура, яка виконує збір даних, що містять інформацію про вибірку об'єктів і потім упорядковує об'єкти в порівняно однорідні групи – кластери (Q-кластеризація, або Q-техніка, власне кластерний аналіз).

Основна мета кластерного аналізу – знаходження груп схожих об'єктів у вибірці. Спектр застосувань кластерного аналізу дуже широкий: його використовують в археології, антропології, медицині, психології, хімії, біології, державному управлінні, філології, маркетингу, соціології та інших дисциплінах. Однак універсальність застосування привела до появи великої кількості несумісних термінів, методів і підходів, що ускладнюють однозначне використання і несуперечливу інтерпретацію кластерного аналізу.

Формальне визначення кластеризації

Нехай X – множина об'єктів, Y – множина номерів (імен, міток) кластерів. Задано функцію відстані між об'єктами $p(x, x')$. Є кінцева вибірка об'єктів $X_m = \{x_1, \dots, x_m\}$ з X . Потрібно розбити вибірку на непересічні підмножини, що називаються кластерами, так, щоб кожен кластер складався з об'єктів, близьких по метриці p , а об'єкти різних кластерів істотно відрізнялися. При цьому кожному об'єкту $x_i \in X^m$ приписується номер кластеру u_i .

Алгоритм кластеризації – це функція $\alpha: X \rightarrow Y$, яка будь-якому $x \in X$ ставить у відповідність номер кластера $u \in Y$. Множина Y в деяких випадках відома заздалегідь, проте частіше ставиться завдання визначити оптимальне число кластерів, з погляду деякого критерію якості кластеризації.

Завдання кластеризації

Кластерний аналіз виконує наступні основні завдання:

- Розробка типології або класифікації.
- Дослідження корисних концептуальних схем групування об'єктів.
- Породження гіпотез на основі дослідження даних.
- Перевірка гіпотез або дослідження для визначення, чи дійсно групи, виділені тим чи іншим способом, присутні в наявних даних.

4.2.8. Аналіз часових рядів [53].

Аналіз часових рядів – сукупність математико-статистичних методів аналізу, призначених для виявлення структури часових рядів і для їх прогнозування. Сюди належать, зокрема, методи регресійного аналізу. Виявлення структу-

ри часового ряду необхідно для того, щоб побудувати математичну модель того явища, яке є джерелом аналізованого часового ряду. Прогноз майбутніх значень часового ряду використовується для ефективного прийняття рішень.

Дві основні цілі аналізу часових рядів. Існують дві основні мети аналізу часових рядів: (1) визначення природи ряду і (2) прогнозування (проорокування майбутніх значень часового ряду по теперішнім і минулим значенням). Обидві ці цілі вимагають, щоб модель ряду була ідентифікована і, більшменш, формально описана. Як тільки модель визначена, ви можете з її допомогою інтерпретувати представлені дані (наприклад, використовувати у вашій теорії для розуміння сезонної зміни цін на товари, якщо займаєтеся економікою). Не звертаючи уваги на глибину розуміння і справедливості теорії, ви можете екстраполювати потім ряд на основі знайденої моделі, тобто передбачити його майбутні значення.

Методи аналізу часових рядів. Часові ряди досліджуються з різними цілями. В одному ряді випадках буває достатньо отримати опис характерних особливостей ряду, а в іншому ряді випадків потрібне не тільки передбачати майбутні значення часового ряду, а й управляти його поведінкою. Метод аналізу часового ряду визначається, з одного боку, цілями аналізу, а з іншого боку, ймовірнісної природою формування його значень.

Найпоширеніші методи аналізу часових рядів:

Спектральний аналіз – дозволяє знаходити періодичні складові часового ряду

Кореляційний аналіз – дозволяє знаходити суттєві періодичні залежності і відповідні їм затримки (лаги) як всередині одного ряду (автокореляція), так і між кількома рядами (кроскореляція).

Моделі авторегресії і ковзного середнього – моделі орієнтовані на опис процесів, що виявляють однорідні коливання, порушувані випадковими впливами. Дозволяють передбачати майбутні значення ряду.

Багатоканальні моделі авторегресії і ковзного середнього – моделі застосовуються в тих випадках, коли є кілька корельованих між собою часових рядів. У них є коливання, порушувані однією причиною. Дозволяють передбачати майбутні значення ряду.

Сезонна модель Бокса-Дженкінса – застосовується, коли часовий ряд містить явно виражений лінійний тренд і сезонні складові. Дозволяє передбачати майбутні значення ряду. Модель була запропонована у зв'язку з аналізом авіаперевезень.

Прогноз експоненціально зваженим ковзаючим середнім – найпростіша модель прогнозування часового ряду. Застосовна в багатьох випадках. У тому числі, охоплює модель ціноутворення на основі випадкових блукань

Стадії аналізу часових рядів. Зазвичай при практичному аналізі часових рядів послідовно проходять такі етапи:

1. Графічне подання і опис поведінки часового ряду.
2. Виділення та видалення закономірних складових часового ряду, що залежать від часу: тренда, сезонних і циклічних складових.

3. Виділення та видалення низько- або високочастотних складових процесу (фільтрація).

4. Дослідження випадкової складової часового ряду, що залишилася після видалення перерахованих вище складових.

5. Побудова (підбір) математичної моделі для опису випадкової складової і перевірка її адекватності.

6. Прогнозування майбутнього розвитку процесу, представленого часовим рядом.

Систематична складова і випадковий шум. Як і більшість інших видів аналізу, аналіз часових рядів передбачає, що дані містять систематичну складову (зазвичай включає кілька компонент) і випадковий шум (помилку), який ускладнює виявлення регулярних компонент. Більшість методів дослідження часових рядів включає різні способи фільтрації шуму, що дозволяють побачити регулярну складову більш чітко.

Часовий ряд – це послідовність впорядкованих у часі числових показників, що характеризують рівень стану і зміни досліджуваного явища.

Класифікація часових рядів. Всякий часовий ряд включає два обов'язкових елемента: по-перше, час і, по-друге, конкретне значення показника, або рівень ряду.

Отже, часові ряди розрізняються за такими ознаками часу – моментні та інтервальні. Інтервальний ряд – послідовність, в якій рівень явища відносять до результату, накопиченому або знову виробленому за певний інтервал часу. Такі, наприклад, такі ряди показників обсягу продукції підприємства по місяцях року, кількості відпрацьованих людиною днів по окремих періодах (місяцях, кварталах, півріччях, роках, тощо) і т. д. Якщо ж рівень ряду характеризує досліджуване явище в конкретний момент часу, то сукупність рівнів формує моментний ряд.

Забезпечення порівнянності часових рядів. Найважливішою умовою правильного формування часових рядів є порівнянність рівнів, що утворюють ряд. Рівні ряду, що підлягають вивченню, повинні бути однорідні за економічним змістом і враховувати сутність досліджуваного явища і його мету.

Статистичні дані, представлені у вигляді часових рядів, повинні бути порівняні по території, колу охоплених об'єктів, одиницях вимірювання, моменту реєстрації, методикою розрахунку, цінами, достовірності. Неспівмірність за територією виникає в результаті змін кордонів країн, регіонів, господарств тощо Для приведення даних до порівнянного вигляду проводиться перерахунок колишніх даних з урахуванням нових кордонів.

Цілісність охоплення різних частин явища – найважливіша умова порівнянності рівнів ряду. Вимога однакової цілісності охоплення різних частин досліджуваного об'єкта означає, що рівні ряду за окремі періоди повинні характеризувати різноманітність заходів того чи іншого явища по одному і тому ж колу, які входять до складу його частин. Наприклад, при характеристиці динаміки врожайності овочевих культур у регіоні по роках не можна в

одні роки враховувати тільки сільськогосподарські підприємства, а в інші – всі категорії господарств, у тому числі присадибні ділянки сільських жителів і сади, городи городян.

При визначенні порівнюваних рівнів ряду необхідно використовувати єдину методику їх розрахунку. Особливо часто ця проблема виникає при міжнародних порівняннях.

Неспівмірність показників, що виникає в силу застосування різних одиниць вимірювання, сама по собі очевидна. З різницею застосовуваних одиниць вимірювання доводиться зустрічатися при вивченні динаміки: виробничих ресурсів, коли вони представляються то у вартісному, то в трудовому обчисленні; енергетичних потужностей (кВт-год, л.с.); атмосферного тиску і т. д.

Труднощі при порівнянні даних по моменту реєстрації виникають через сезонні явища. Чисельність худоби в домашніх господарствах через економічну доцільність відрізняється взимку і влітку, тому рівні при порівнянні повинні ставитися до певної дати щорічно.

При аналізі показників у вартісному вираженні слід враховувати, що з плином часу відбувається безперервна зміна цін. Причин цього процесу безліч – інфляція, зростання витрат, ринкові умови (попит і пропозиція) і т. д. У цьому зв'язку при характеристиці вартісних показників обсягів продукції в часі має бути усунуто вплив трансформаційних змін цін. Для вирішення цього завдання кількість продукції, виробленої в різні періоди, оцінюють в цінах одного періоду, які називають фіксованими або в визначеними в статистичних органах – порівнянними цінами.

Широке використання в статистичних дослідженнях вибіркового методу вимагає враховувати достовірність кількісних і якісних характеристик досліджуваних явищ в динаміці. Різна репрезентативність вибірки за періодами внесе суттєві похибки в величини рівнів ряду.

Однією з умов порівнянності рівнів інтервального ряду, крім рівності періодів, за які наводять дані, є однорідність етапів, у межах яких показник підпорядковується одному закону розвитку. У цих випадках проводять періодизацію часових рядів, типологічну угруповання в часі. Всі вищезазвані обставини слід враховувати при підготовці інформації для аналізу змін явищ у часі (динаміці).

Зазвичай, метою прикладного статистичного аналізу часових рядів є побудова моделі ряду, за допомогою якої можна пояснити поведінку ряду і здійснити прогноз на майбутні періоди.

Побудова і вивчення графіка. Аналіз часового ряду починається з побудови і вивчення його графіка. Якщо нестационарність часового ряду очевидна, то спочатку необхідно виокремити його нестационарну складову. Процес виокремлення тренду та інших компонент ряду, що призводять до порушення стаціонарності, може проходити в декілька етапів. На кожному з них розглядається ряд залишків, отриманий у результаті вирахування з вихідного ряду підбраної моделі тренду, або результат різницевих і інших перетворень ряду.

Крім графіків, ознаками нестационарності часового ряду можуть служити автокореляційна функція, що прямує не до нуля (за винятком дуже великих значень *лагів*) і наявність яскраво виражених піків на низьких частотах у періодограмі. За допомогою автокореляційної функції досліджують також внутрішні зв'язки між елементами часових рядів.

У вибіркових дослідженнях найпростіші числові характеристики описової статистики (середнє, медіана, дисперсія, стандартне відхилення, коефіцієнти асиметрії ексцесу) звичайно дають достатньо інформативне уявлення про вибірку. Графічні методи зображення й аналізу вибірок при цьому грають лише допоміжну роль, дозволяючи краще зрозуміти локалізацію і концентрацію даних, їхній закон розподілу.

Роль графічних методів при аналізі часових рядів цілком інша. Табличне представлення часового ряду й описових статистик частіше за все не дозволяє зрозуміти характер процесу, у той час як за графіком часового ряду можна зробити досить багато висновків. Надалі вони можуть бути перевірені й уточнені за допомогою розрахунків.

Людське око досить упевнено визначає за графіком часового ряду:

- 1) наявність тренду і його характер;
- 2) наявність сезонних і циклічних компонент;
- 3) ступінь повільності або переривчастості змін послідовних значень ряду після усунення тренду (за цим показником можна судити про характер і розмір кореляції між сусідніми елементами ряду).

Так графічний аналіз ряду звичайно задає напрямок його подальшого аналізу.

Вибір моделі часового ряду. Після того, як вихідний процес максимального наближення до стаціонарного, можна приступити до вибору різноманітних моделей отриманого процесу. Мета цього етапу – опис і урахування надалі аналізу кореляційної структури аналізованого процесу. Модель може вважатися підбраною, якщо залишкова компонента ряду є процесом типу, як правило, «білого шуму». Після підбору залишки аналізуються для перевірки адекватності моделі та побудови надійних інтервалів.

Прогнозування і інтерполяція. Останнім етапом аналізу часового ряду може бути прогнозування його майбутніх (екстраполяція) або відновлення пропущених (інтерполяція) значень і визначення точності цього прогнозу на базі підбраної моделі. Добре підібрати математичну модель вдається не для всякого часового ряду. Нерідко буває і так, що для опису підходять відразу декілька моделей. Неоднозначність вибору моделі може спостерігатися як на етапі виділення детермінованого компонента ряду, так і при виборі структури ряду залишків. Тому досить часто розробляють декілька прогнозів, зроблених за допомогою різних моделей.

Детальне вивчення цих методів здійснено в наступних роботах: Anderson (1976), Бокс і Дженкінс (1976), Kendall (1984), Kendall and Ord (1990), Montgomery, Johnson, and Gardiner (1990), Pankratz (1983), Shumway (1988), Vandaele (1983), Walker (1991), Wei (1989).

Контрольні питання до підрозділу 4.2

- 1. Перелічіть розділи аналізу даних.*
- 2. Перелічіть основні етапи попередньої обробки при здійсненні аналізу даних.*
- 3. Перелічіть головні завдання кореляційного аналізу аналізу даних.*
- 4. Який принцип полягає в основі дисперсійного аналізу даних?*
- 5. Які припущення (стосовно помилок спостережень і факторів впливу) вважаються виконаними при проведенні дисперсійного аналізу даних?*
- 6. Регресійний аналіз, як розділ математичної статистики, присвячений методам аналізу чого?*
- 7. Перелічіть задачі регресійного аналізу.*
- 8. У чому полягає зв'язок коваріаційного аналізу з регресійним та дисперсійним аналізами?*
- 9. Дискримінантний аналіз здебільшого присвячується вирішенню яких задач?*
- 10. У чому полягає задача кластерного аналізу?*
- 11. Яким двом цілям присвячений аналіз часових рядів?*
- 12. Перелічіть найпоширеніші методи аналізу часових рядів.*
- 13. Перелічіть стадії аналізу часових рядів.*

4.3. Методи інтелектуального аналізу даних

4.3.1 Проблеми аналізу даних із застосуванням класичних статистичних підходів

Технології аналізу даних, що базуються на застосуванні класичних статистичних підходів, мають низку недоліків [54]. Відповідні методи ґрунтуються на використанні усереднених показників, на підставі яких важко з'ясувати справжній стан справ у досліджуваній сфері (наприклад, середня зарплата по країні не відбиває її розміру у великих містах та в селах). Методи математичної статистики виявилися корисними насамперед для перевірки заздалегідь сформульованих гіпотез та «грубого» розвідницького аналізу, що становить основу оперативної (в режимі он-лайн) аналітичної обробки даних (OLAP).

Наприклад, дослідження спеціалістів Гарвардського інституту показують, що на основі наявної інформації за допомогою стандартних статистичних методів не можна було передбачити великої депресії кінця 1920-х років.

Окрім того, стандартні статистичні методи відкидають (нехтують) нетипові спостереження – так звані піки та сплески. Проте окремі нетипові значення можуть становити самостійний інтерес для дослідження, характеризуючи деякі виняткові, але важливі явища. Навіть сама ідентифікація цих спостережень, не говорячи про їх подальший аналіз і докладний розгляд, може бути корисною для розуміння сутності досліджуваних об'єктів чи явищ. Як показують сучасні дослідження, саме такі події можуть стати вирішальними щодо майбутнього поведіння та розвитку складних систем.

Ці недоліки статистичних методів спонукали до розвитку нових методів дослідження складних систем, що базуються на нелінійній динаміці, теорії катастроф, фрактальній геометрії тощо (див. розд. 5).

Водночас постала нагальна потреба в такій технології, яка автоматично видобувала б із даних нові нетривіальні знання у формі моделей, залежностей, законів тощо, гарантуючи при цьому їхню статистичну значущість. Новітні підходи, спрямовані на розв'язання цих проблем, дістали назву технологій інтелектуального аналізу даних.

В основу цих технологій покладено **концепцію шаблонів (патернів)**, що відбивають певні фрагменти багатоаспектних зв'язків у множині даних, характеризуючи закономірності, притаманні підвибіркам даних, які можна компактно подати у зрозумілій людині формі. Шаблони відшуковують методами, що виходять за межі апріорних припущень стосовно структури вибірки та вигляду розподілів значень аналізованих показників. Важлива особливість цієї технології полягає в нетривіальності відшукуваних шаблонів. Це означає, що вони мають відбивати неочевидні, несподівані регулярності у множині даних, складові так званого прихованого знання. Адже сукупність первинних («сирих») даних може містити й глибинні шари знань.

4.3.2. Технологія KDD (Knowledge Discovery in Databases)

Knowledge Discovery in Databases (дослівно: «виявлення знань у базах даних») – KDD) – аналітичний процес дослідження значних обсягів інформації із залученням засобів автоматизації, що має на меті виявити приховані у множині даних структури, залежності й взаємозв'язки [55]. При цьому передбачається повна чи часткова відсутність апріорних уявлень про характер прихованих структур та залежностей. KDD передбачає, що людина попередньо осмислює задачу й подає неповне (у термінах цільових змінних) її формулювання, перетворює дані до формату придатного для їх автоматизованого аналізу й попередньої обробки, виявляє засобами автоматичного дослідження даних приховані структури й залежності, апробує виявлені моделі на нових даних, не використовуваних для побудови моделей, та інтерпретує виявлені моделі й результати.

Отже, KDD – це синтетична технологія, що поєднує в собі останні досягнення штучного інтелекту, чисельних математичних методів, статистики й евристичних підходів. Методи KDD особливо стрімко розвиваються протягом останніх 20 років, а раніше задачі комп'ютерного аналізу баз даних виконувалися переважно за допомогою різного роду стандартних статистичних методів.

DataMining (дослівно: «Розробка, добування даних») – DM) – дослідження «сирих» даних і виявлення в них за допомогою «машини» (алгоритмів, засобів штучного інтелекту) прихованих нетривіальних структур і залежностей, які раніше не були відомі й мають практичну цінність та придатні для того, щоб їх інтерпретувала людина (докладно буде розглянуто у підрозділі 4.4).

4.3.3. Відмінності між засобами KDD і OLAP

Технологія OLAP (OnLineAnalyticalProcessing) спрямована на підтримання процесу прийняття управлінських рішень і використовується з метою пошуку відповіді на запитання: чому деякі речі є такими, якими вони є насправді [56]? При цьому користувач сам формує модель-гіпотезу про дані чи відношення між даними, а далі, застосовуючи серію запитів до бази даних, підтверджує чи відхиляє висунуті гіпотези. Засоби Data Mining відрізняються від засобів OLAP тим, що замість перевірки передбачуваних користувачем взаємозалежностей вони на основі наявних даних самі можуть будувати моделі, які дають змогу кількісно та якісно оцінювати ступінь впливу різних досліджуваних факторів на задану властивість об'єкта. Крім того, засоби DM дають змогу формулювати нові гіпотези про характер досі невідомих, але таких, що реально існують, залежностей між даними.

Засоби OLAP застосовуються на ранніх стадіях процесу KDD, оскільки вони дають змогу краще зрозуміти дані, що, у свою чергу, забезпечує ефективніший результат процесу KDD.

Головна мета технології KDD – побудова моделей і відношень, прихованих у базі даних, тобто таких, які не можна знайти звичайними методами. Варто зазначити, що на комп'ютери перекладаються не лише рутинні опера-

ції (скажімо, перевірка статистичної значущості гіпотез), а й операції, що донедавна були аж ніяк не рутинними (вироблення нових гіпотез). KDD дає змогу побачити такі відношення між даними, що залишалися поза увагою дослідників.

Будуючи моделі, ми встановлюємо кількісні зв'язки між характеристиками досліджуваного явища. Щодо призначення можна виокремити моделі двох типів: прогностні та описові (deskриптивні). Моделі першого типу використовують набори даних із відомими результатами для побудови моделей, що явно прогнозують результати для інших наборів даних, а моделі другого типу описують залежності в наявних даних. Обидва типи моделей використовуються для прийняття управлінських рішень.

Технологія KDD дає змогу не лише підтверджувати (відкидати) емпіричні висновки, а й будувати нові, невідомі раніше моделі. Знайдена модель не зможе здебільшого претендувати на абсолютне знання, але вона надає аналітикові деякі переваги вже завдяки самому факту виявлення альтернативної статистично значущої моделі, а також, можливо, стає приводом для пошуку відповіді на запитання: чи справді існує виявлений взаємозв'язок і чи є він причинним? А це, у свою чергу, стимулює поглиблені дослідження, сприяючи глибшому розумінню досліджуваного явища.

Отже, найважливіша мета застосування технології KDD до дослідження реальних систем – це поліпшення розуміння суті їх функціонування.

Відзначимо, що процес виявлення знань не є цілком автоматизованим – він вимагає участі користувача (експерта, особи що приймає рішення). Користувач має чітко усвідомлювати, що він шукає, ґрунтуючись на власних гіпотезах. Зрештою замість того, щоб підтверджувати наявну гіпотезу, процес пошуку часто сприяє появі ряду нових гіпотез. Усе це позначається терміном «discovery-driven data mining» (DDDM), і терміни DataMining, Knowledge Discovery у загальному випадку стосуються до технології DDDM.

Контрольні питання до підрозділу 4.3

1. Перелічіть недоліки статистичних методів, що спонукали до розвитку нових методів дослідження складних систем.

2. У чому полягає концепція шаблонів (патернів) у процесі отримання нових знань?

3. У чому полягає сутність технології KDD (Knowledge Discovery in Databases)?

4. У чому полягає сутність технології DM (Data Mining)?

5. У чому полягає сутність технології OLAP (OnLine Analytical Processing)?

6. У чому полягають відмінності між засобами KDD і OLAP?

7. Які методи складають основу методів data mining?

8. При здійсненні процесу data mining, якими характеристиками наділяються так звані «приховані знання»?
9. Які задачі обслуговує мова SQL?
10. Який сенс (зміст) має термін «патерн» у процесі видобування знань методами data mining?
11. На які дві групи прийнято поділяти завдання, що вирішуються методами data mining?
12. У чому полягає необхідна підготовка набору аналізованих даних перед використанням алгоритмів data mining?
13. Яку назву у сучасній літературі має технологія обробки даних, яка полягає в підготовці сумарної (агрегированной) інформації на основі великих масивів даних, структурованих по багатомірному принципу?
14. Хто сформулював «12 правил аналітичної обробки даних в реальному часі»?
15. У чому полягає основна причина використання OLAP для обробки запитів?
16. Які три типи OLAP Вам відомі ?
17. У чому полягає складність у застосуванні OLAP ?
18. У чому полягає різниця між «фізичною OLAP» та «віртуальною OLAP»?

4.4. Datamining (виявлення знань в базах даних)

4.4.1. Загальні відомості

Datamining (укр. видобуток даних, інтелектуальний аналіз даних, глибинний аналіз даних) – збірна назва, що використовується для позначення сукупності методів виявлення в даних раніше невідомих, нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності [57]. Термін введений Григорієм П'ятецьким-Шапіро в 1989 році.

Англійське словосполучення «data mining» поки не має усталеного перекладу на українську мову. При передачі українською мовою використовуються наступні словосполучення:

- просіювання інформації,
- видобуток даних,
- вилучення даних,
- інтелектуальний аналіз даних.

Більш повним і точним є словосполучення «виявлення знань в базах даних» (англ. Knowledge discovery in databases, KDD) [58].

Основу методів data mining складають всілякі методи класифікації, моделювання і прогнозування, засновані на застосуванні:

- дерев рішень,
- штучних нейронних мереж,

- генетичних алгоритмів,
- еволюційного програмування,
- асоціативної пам'яті,
- нечіткої логіки.

До методів data mining нерідко відносять статистичні методи (частку з яких докладно розглянуто у підрозділі 4.2.2):

- описувальний аналіз,
- кореляційний та регресійний аналіз,
- факторний аналіз,
- дисперсійний аналіз,
- компонентний аналіз,
- дискримінантний аналіз,
- аналіз часових рядів,
- аналіз виживаності, аналіз зв'язків).

Такі методи, проте, припускають деякі апріорні уявлення про аналізовані дані, що виникає певна розбіжність з цілями data mining (виявлення раніше невідомих нетривіальних і практично корисних знань).

Одне з найважливіших призначень методів data mining полягає в наочному поданні результатів обчислень (візуалізація), що дозволяє використовувати інструментарій data mining людьми, які не мають спеціальної математичної підготовки.

Застосування статистичних методів аналізу даних вимагає доброго володіння теорією ймовірностей і математичної статистики.

Методи data mining (або, що те ж саме, knowledge discovery in data, скорочено KDD) лежать на стику баз даних, статистики та штучного інтелекту.

Область data mining почалася з семінару, проведеного Григорієм Пятецьким-Шапіро в 1989 році.

Раніше, працюючи в компанії GTE Labs, Григорій Пятецький-Шапіро зацікавився питанням: чи можна автоматично знаходити певні правила, щоб прискорити деякі запити до великих баз даних. Тоді ж було запропоновано два терміни – data mining («видобуток даних» [9]) і knowledge discovery in data (який слід перекладати як «відкриття знань в базах даних»).

У 1993 році вийшла перша розсилка «Knowledge Discovery Nuggets», а в 1994 році був створений один з перших сайтів по data mining.

4.4.2. Постановка завдання

Спочатку завдання ставиться таким чином:

- є досить велика база даних;
- передбачається, що в базі даних знаходяться якісь «приховані знання».

Необхідно розробити методи виявлення знань, прихованих в великих обсягах вихідних «сирих» даних. У поточних умовах глобальної конкуренції саме знайдені закономірності (знання) можуть бути джерелом додаткової конкурентної переваги.

Що означає «приховані знання»? Це повинні бути обов'язково знання:

- раніше невідомі - тобто такі знання, які повинні бути новими (а не підтверджують якісь раніше отримані відомості);
- нетривіальні - тобто такі, які не можна просто так побачити (при безпосередньому візуальному аналізі даних або при обчисленні простих статистичних характеристик);
- практично корисні - тобто такі знання, які представляють цінність для дослідника або споживача;
- доступні для інтерпретації - тобто такі знання, які легко уявити в наочній формі та легко пояснити в термінах предметної області.

Ці вимоги багато в чому визначають суть методів data mining і то, в якому вигляді і в якому співвідношенні в технології data mining використовуються системи управління базами даних, статистичні методи аналізу і методи штучного інтелекту.

4.4.3. Data mining і бази даних

Методи data mining можуть бути застосовані як для роботи з великими даними, так і для обробки порівняно малих обсягів даних (отриманих, наприклад, за результатами окремих експериментів, або при аналізі даних про діяльність компанії). В якості критерію достатньої кількості даних розглядається як область дослідження, так і застосовуваний алгоритм аналізу.

Розвиток технологій баз даних спочатку привів до створення спеціалізованої мови – мови запитів до баз даних. Для реляційних баз даних – це мова SQL, яка надає широкі можливості для створення, зміни та вилучення даних, що зберігаються. Потім виникла необхідність в отриманні аналітичної інформації (наприклад, інформації про діяльність підприємства за певний період), і тут виявилось, що традиційні реляційні бази даних, добре пристосовані, наприклад, для ведення оперативного обліку на підприємстві, погано пристосовані для проведення аналізу. Це призвело, в свою чергу, до створення так званих «Сховищ даних», сама структура яких найкращим способом відповідає проведенню всебічного математичного аналізу.

Знання, що видобуваються методами data mining, прийнято представляти у вигляді закономірностей (патернів). Серед таких виступають:

- асоціативні правила;
- дерева рішень;
- кластери;
- математичні функції.

Алгоритми пошуку таких закономірностей знаходяться на перетині областей: штучний інтелект, математична статистика, математичне програмування, візуалізація, OLAP.

4.4.4. Коло завдань

Завдання, що вирішуються методами data mining, прийнято розділяти на описові (англ. descriptive) і передбачувальна (англ. predictive).

В описових завданнях найголовніше - це дати наочний опис наявних прихованих закономірностей, в той час як в передбачувальних завданнях на першому плані стоїть питання про передбачення для тих випадків, для яких даних ще немає.

До описових завдань відносяться:

- пошук асоціативних правил або патернів (зразків);
- групування об'єктів, кластерний аналіз;
- побудова регресійної моделі.

До завдань передбачення відносяться:

- класифікація об'єктів (для заздалегідь заданих класів);
- регресійний аналіз, аналіз часових рядів.

Алгоритми навчання. Для задач класифікації характерно «навчання з учителем», при якому побудова (навчання) моделі проводиться за вибіркою, що містить вхідні та вихідні вектори.

Для задач кластеризації та асоціації застосовується «навчання без учителя», при якому побудова моделі проводиться за вибіркою, в якій немає вихідного параметра. Значення вихідного параметра («відноситься до кластеру ...», «схожий на вектор ...») підбирається автоматично в процесі навчання.

Для завдань скорочення опису характерна відсутність поділу на вхідні і вихідні вектори. Починаючи з класичних робіт К. Пірсона за методом головних компонент, основна увага приділяється апроксимації даних.

Етапи вирішення завдань методами data mining:

1. Постановка завдання аналізу;
2. Збір даних;
3. Підготовка даних (фільтрація, доповнення, кодування);
4. Вибір моделі (алгоритму аналізу даних);
5. Підбір параметрів моделі і алгоритму навчання;
6. Навчання моделі (автоматичний пошук інших параметрів моделі);
7. Аналіз якості навчання. Якщо він незадовільний, то перехід на п. 5 або п. 4;
8. Аналіз виявлених закономірностей. Якщо він незадовільний, то перехід на п. 1, 4 або 5.

Підготовка даних. Перед використанням алгоритмів data mining необхідно провести підготовку набору аналізованих даних. Оскільки інтелектуальний аналіз даних (ІАД) може виявити лише присутні в даних закономірності, вихідні дані з одного боку повинні мати достатній обсяг, щоб ці закономірності були в них присутні, а з іншого – бути достатньо компактними, щоб аналіз зайняв прийнятний час. Найчастіше в якості вихідних даних виступають сховища або вітрини даних. Підготовка необхідна для аналізу багатовимірних даних до кластеризації або інтелектуального аналізу даних.

Далі дані фільтруються. Фільтрація видаляє вибірки з шумами і пропущеними даними.

Відфільтровані дані зводяться до наборів ознак (або векторах, якщо алгоритм може працювати тільки з векторами фіксованою розмірності), один набір ознак на спостереження. Набір ознак формується відповідно до гіпотезами про те, які ознаки сирих даних мають високу прогностичну силу в розрахунку на необхідну обчислювальну потужність для обробки. Наприклад, чорнобіле зображення особи розміром 100×100 пікселів містить 10 тис. Біт сирих даних. Вони можуть бути перетворені в вектор ознак шляхом виявлення в зображенні очей і рота. В результаті відбувається зменшення обсягу даних з 10 тис. Біт до списку кодів положення, значно зменшуючи обсяг аналізованих даних, а значить і час аналізу.

Ряд алгоритмів вміють обробляти пропущені дані, що мають прогностичну силу (наприклад, відсутність у клієнта покупок певного виду). Скажімо, при використанні методу асоціативних правил обробляти не вектори ознак, а набори змінної розмірності.

Вибір цільової функції буде залежати від того, що є метою аналізу; вибір «правильної» функції має основоположне значення для успішного інтелектуального аналізу даних.

Спостереження діляться на дві категорії - навчальний набір і тестовий набір. Навчальний набір використовується для «навчання» алгоритму data mining, а тестовий набір – для перевірки знайдених закономірностей.

Контрольні питання до підрозділу 4.4

- 1. Якими особливостями характеризуються так звані «прихованні знання»?*
- 2. У чому полягає сутність постановки завдання щодо здійснення datamining?*
- 3. Які переваги мають «сховища даних» перед традиційними базами даних?*
- 4. У чому полягає різниця у описових та перебачувальних завданнях щодо здійснення datamining?*
- 5. Які основні етапи містить вирішення завдань методами data mining?*
- 6. Які суперечливі вимоги висуваються до даних при їхній підготовці до процесу datamining?*

4.5. Інтерактивна аналітична обробка даних (OLAP)

4.5.1. Правила аналітичної обробки даних в реальному часі

OLAP (англ. Online analytical processing – інтерактивна аналітична обробка) – технологія обробки даних, яка полягає в підготовці сумарної (агрегованої) інформації на основі великих масивів даних, структурованих по багатовимірному принципу. Реалізації технології OLAP є компонентами програмних рішень класу BusinessIntelligence [1].

Основоположник терміна OLAP - Едгар Кодд, запропонував в 1993 році «12 правил аналітичної обробки в реальному часі» (за аналогією з раніше сформульованими «12 правил для реляційних баз даних»).

Правило 0: Основне правило (Foundation Rule):

Система, яка рекламується або позиціонується як реляційна система управління базами даних, повинна бути здатна управляти базами даних, використовуючи виключно свої реляційні можливості.

Правило 1: Інформаційне правило (The Information Rule):

Вся інформація в реляційній базі даних на логічному рівні повинна бути явно представлена єдиним способом: значеннями в таблицях.

Правило 2: Гарантований доступ до даних (Guaranteed Access Rule):

У реляційній базі даних кожне окреме (атомарний) значення даних має бути логічно доступно за допомогою комбінації імені таблиці, значення первинного ключа і імені стовпця.

Правило 3: Систематична підтримка відсутніх значень (Systematic Treatment of Null Values):

Невідомі, або відсутні значення NULL, відмінні від будь-якого відомого значення, повинні підтримуватися для всіх типів даних при виконанні будь-яких операцій. Наприклад, для числових даних невідомі значення не повинні розглядатися як нулі, а для символічних даних - як порожні рядки.

Правило 4: Доступ до словника даних в термінах реляційної моделі (Active On-Line Catalog Based on the Relational Model):

Словник даних повинен зберігатися в формі реляційних таблиць, і СУБД повинна підтримувати доступ до нього за допомогою стандартних мовних засобів, тих же самих, які використовуються для роботи з реляційними таблицями, що містять призначені для користувача дані.

Правило 5: Повнота підмножини мови (Comprehensive Data Sublanguage Rule):

Система керування базами даних повинна підтримувати хоча б один реляційний мову, який

(А) має лінійний синтаксис,

(Б) може використовуватися як інтерактивно, так і в прикладних програмах,

(В) підтримує операції визначення даних, визначення уявлень, маніпулювання даними (інтерактивні та програмні), обмежувачі цілісності, управління доступом і операції управління транзакціями (begin, commit і rollback).

Правило 6: Можливість зміни уявлень (View Updating Rule):

Кожна вистава має підтримувати всі операції маніпулювання даними, які підтримують реляційні таблиці: операції вибірки, вставки, зміни та видалення даних.

Правило 7: Наявність високорівневих операцій управління даними (High-Level Insert, Update, and Delete):

Операції вставки, зміни та видалення даних повинні підтримуватися не тільки по відношенню до одного рядка реляційної таблиці, але і по відношенню до будь-якого безлічі рядків.

Правило 8: Фізична незалежність даних (Physical Data Independence):

Додатки не повинні залежати від використовуваних способів зберігання даних на носіях, від апаратного забезпечення комп'ютерів, на яких знаходиться реляційна база даних.

Правило 9: Логічна незалежність даних (Logical Data Independence):

Представлення даних в додатку не повинно залежати від структури реляційних таблиць. Якщо в процесі нормалізації одна реляційна таблиця розділяється на дві, подання має забезпечити об'єднання цих даних, щоб зміна структури реляційних таблиць не позначалося на роботі додатків.

Правило 10: Незалежність контролю цілісності (Integrity Independence):

Вся інформація, необхідна для підтримки цілісності, повинна знаходитися в словнику даних. Мова для роботи з даними повинен виконувати перевірку вхідних даних і автоматично підтримувати цілісність даних.

Правило 11: Незалежність від розташування (Distribution Independence):

База даних може бути розподіленою, може перебувати на декількох комп'ютерах, і це не повинно впливати на додатки. Перенесення бази даних на інший комп'ютер не повинен впливати на додатки.

Правило 12: Узгодження мовних рівнів (The Nonsubversion Rule):

Якщо використовується низькорівнева мова доступу до даних, вона не повинна ігнорувати правила безпеки і правила цілісності, які підтримуються мовою більш високого рівня.

4.5.2. Дія OLAP

Причина використання OLAP для обробки запитів - швидкість. Реляційні бази даних зберігають сутності в окремих таблицях, які зазвичай добре нормалізовані. Ця структура зручна для операційних баз даних (системи Online Transaction Processing – OLTP), але складні багато табличні запити в ній виконуються відносно повільно.

OLAP-структура, створена з робочих даних, називається OLAP-куб. Куб створюється з з'єднання таблиць із застосуванням схеми зірки або схеми сніжинки. У центрі схеми зірки знаходиться таблиця фактів, яка містить ключові факти, за якими робляться запити. Множинні таблиці з вимірами приєднані до таблиці фактів. Ці таблиці показують, як можуть аналізуватися агреговані [en] реляційні дані. Кількість можливих агрегування визначається кількістю способів, якими початкові дані можуть бути ієрархічно відображені.

Наприклад, всі клієнти можуть бути згруповані по містах або регіонах країни (Захід, Схід, Північ і так далі), таким чином, 50 міст, вісім регіонів і дві країни складуть три рівня ієрархії з 60-ма членами. Також клієнти можуть бути об'єднані по відношенню до продукції; якщо існують 250 продуктів по 20 категоріям, три групи продукції і три виробничих підрозділи, то кількість

агрегатів складе 16 560. При додаванні вимірювань в схему кількість можливих варіантів швидко досягає десятків мільйонів і більше.

OLAP-куб містить базові дані та інформацію про вимірювання (агрегати). Куб потенційно містить всю інформацію, яка може знадобитися для відповідей на будь-які запити. При величезній кількості агрегатів часто повний розрахунок відбувається тільки для деяких вимірювань, для інших же проводиться «на вимогу».

Існують три типи OLAP: [2]

- багатовимірна OLAP (Multidimensional OLAP - MOLAP);
- реляційна OLAP (Relational OLAP - ROLAP);
- гібридна OLAP (Hybrid OLAP - HOLAP).

MOLAP - класична форма OLAP, так що її часто називають просто OLAP. Вона використовує підсумовує базу даних і створює необхідну багатовимірну схему даних зі збереженням як базових даних, так і агрегатів.

ROLAP працює безпосередньо з реляційною базою даних, факти і таблиці з вимірами зберігаються в реляційних таблицях, і для зберігання агрегатів створюються додаткові реляційні таблиці.

HOLAP використовує реляційні таблиці для зберігання базових даних і багатовимірні таблиці для агрегатів.

Особливим випадком ROLAP є «ROLAP реального часу» (Real-time ROLAP - R-ROLAP). На відміну від ROLAP в R-ROLAP для зберігання агрегатів не створюються додаткові реляційні таблиці, а агрегати розраховуються в момент запиту. При цьому багатовимірний запит до OLAP-системи автоматично перетворюється в SQL-запит до реляційних даних.

Кожен тип зберігання має певні переваги, хоча є розбіжності в їх оцінці у різних виробників. MOLAP найкраще підходить для невеликих наборів даних, він швидко розраховує агрегати і повертає відповіді, але при цьому генеруються величезні обсяги даних. ROLAP вважається більш масштабованим рішенням, до того ж більш економічним до простору зберігання, але з обмеженнями по можливостям аналітичної обробки. HOLAP знаходиться посеред цих двох підходів, він досить добре масштабується, і дозволяє подолати ряд обмежень. Архітектура R-ROLAP дозволяє виробляти багатовимірний аналіз OLTP-даних в режимі реального часу.

Складність в застосуванні OLAP полягає в створенні запитів, виборі базових даних і розробці схеми, в результаті чого більшість продуктів OLAP поставляються разом з величезною кількістю попередньо налаштованих запитів. Інша проблема - в базових даних, вони повинні бути повними та не суперечили одна.

4.5.3. Реалізація OLAP

Історично першою багатовимірною системою управління базами даних, по суті є OLAP-реалізацією, вважається система Express, розроблена в 1970-му році компанією IRI (пізніше права на продукт було придбано корпорацією Oracle і перетворений в OLAP-опцію для Oracle Database) [3]. Термін OLAP ввів Едгар Кодд в публікації в журналі Computer world в 1993 році [4], в якій

він запропонував 12 принципів аналітичної обробки, по аналогії з 12 правилами для реляційних баз даних, сформульованими ним же десятиліттям раніше, в якості референтного продукту, який задовольняє запропонованими принципами, Кодд вказав систему Essbase компанії Arbor (поглиненої в 1997 році компанією Huregion, яку, в свою чергу, в 2007-му році купила Oracle). Примітно, що згодом публікація була вилучена з архівів Computer world через можливого конфлікту інтересів, так як Кодд пізніше надавав консультаційні послуги для Arbor [5].

З точки зору реалізації, діляться на «фізичну OLAP» і «віртуальну» (реляційну, англ. Relational OLAP, ROLAP). «Фізична», в свою чергу, в залежності від реалізації підрозділяється на багатовимірну (англ. Multidimensional OLAP, MOLAP) і гібридну - (англ. Hybrid OLAP, HOLAP).

У першому випадку в наявності програма, яка виконує на етапі попереднього завантаження даних в OLAP попередній розрахунок агрегатів (обчислень по декількох початкових значень, наприклад «підсумок за місяць»), які потім зберігаються в спеціальну багатовимірну базу даних, що забезпечує швидке вилучення та економічне зберігання.

Гібридна реалізація є комбінацією: самі дані зберігаються в реляційній базі даних, а агрегати - в багатовимірній.

У ROLAP-реалізаціях всі дані зберігаються і обробляються в реляційних системах управління базами даних, а агрегати можуть не існувати взагалі або створюватися за першим запитом до бази даних або у кеші аналітичного програмного забезпечення.

З точки зору користувача, всі варіанти виглядають схожими за можливостями. Найбільше застосування OLAP знаходить в продуктах для фінансового планування, сховищах даних, рішеннях класу Business Intelligence.

Серед комерційних продуктів виділяють:

Microsoft SQL Server Analysis Services,

Essbase,

PowerPlay,

Business Objects [en],

Micro Strategy,

SAP BW,

Cartesis Magnitude,

OracleDatabase OLAP Option, TM1.

Існує кілька вільних рішень, серед них відзначаються Mondrian і Palo.

Контрольні питання до підрозділу 4.5

1. Які умови визначають необхідність застосування технології обробки даних "Online analytical processing" (OLAP)?
2. Скільки правил запропонував Едгар Кодд стосовно аналітичної обробки даних в реальному часі?
3. Основна перевага, яку надає використання OLAP для обробки запитів?
4. Яка структура даних має назву OLAP-куб?
5. Які типи OLAP Вам відомі?
6. Які особливості має багатовимірна інтерактивна аналітична обробка даних (MOLAP)?
7. Які особливості має реляційна інтерактивна аналітична обробка даних (ROLAP)?
8. У чому полягає сутність «гібридності» інтерактивної аналітичної обробки даних (HOLAP)?
9. Які особливості має реляційна інтерактивна аналітична обробка даних Real-time ROLAP – R-ROLAP?

РОЗДІЛ 5. СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ СХОВИЩ ДАНИХ ТА OLAP-СИСТЕМ

5.1. Розвиток та застосування СППР на основі сховищ даних та OLAP-систем

5.1.1. Передумови та сутність СППР на основі сховищ даних та OLAP-систем

Системи підтримки прийняття рішень на основі сховищ даних та OLAP-систем, як і самі сховища даних (Data Warehouses) та системи аналітичного онлайн-оброблення даних, належать до типу орієнтованих на дані СППР [59]. У загальному вигляді орієнтовану на дані систему підтримки прийняття рішень (ОДСППР) можна визначити як *інтерактивну комп'ютеризовану систему, що допомагає ОПР використовувати дуже велику базу даних із внутрішніх даних компанії і деякі зовнішні дані з навколишнього середовища системи з метою прийняття обґрунтованих рішень*. Наприклад, система може надавати дані щодо збуту продукції як самої компанії, так і її конкурентів. Деякі дані можуть бути деталізованими даними транзакцій, а деякі – агрегованими. У більшості реалізованих нині ОДСППР користувачі можуть виконувати незаплановані або в режимі на даний випадок (ad hoc) аналізи даних і формулювати запити. За допомогою таких систем менеджери обробляють дані для ідентифікації фактів і отримання висновків у вигляді графічних зображень (діаграм, графіків, трендів).

Орієнтовані на дані СППР, зокрема системи аналітичного онлайн-оброблення, інколи називають бізнес-інформаційними (*Businessintelligence*). Цей популярний, широко відомий термін запропонував 1989 року аналітик Говард Дрезнер (Howard Dresner) з групи «Gartner», що консулює фірми «Fortune1000» з питань інформаційних технологій, для описання низки понять і методів, призначених для вдосконалення бізнесових рішень за допомогою використання основаних на фактах систем підтримки прийняття рішень (власне замість терміна «програмне забезпечення» («decision support») був запропонований термін «Business intelligence»). Зокрема, 1994 року Говард Дрезнер писав: «До 1996 року використання розв'язків Business intelligence різко переміститься від спеціалізованих аналітиків до всіх менеджерів і професіоналів як найкращий шлях розуміння бізнесу... Замість вузького кола аналітиків, які витрачають 100 відсотків свого часу на аналізування даних, усі менеджери і професіонали витратять 10 відсотків свого часу для цього, використовуючи програмне забезпечення Business intelligence» [60]. Хоча ні 1996, ні навіть 2001 року повною мірою цей прогноз не справдився, проте багато менеджерів освоїли і продовжують вивчати цей тип програмних продуктів. Така обставина пояснюється тим, що програмне забезпечення орієнтованих на дані СППР надає змогу користувачам досягати як загальної

для всієї корпорації мети, тобто підвищення рівня конкурентоспроможності за рахунок своєчасного отримання важливої для успіху інформації, недосяжної раніше іншими шляхами, так і поліпшувати особисту управлінську продуктивність, тобто різко скорочувати завдяки таким СППР час на отримання таких самих результатів.

Термін «Business intelligence» використовується як синонім «OLAP» в інструкціях щодо використання і в описаннях виконавчих інформаційних систем. «Business intelligence» є терміном з маркетингу. Він уживається також для позначення орієнтованих на дані СППР. Цей тип СППР допомагає користувачам реалізувати практичне оброблення вниз (drilldown) для детальнішого огляду інформації, оброблення вгору (drillup), щоб оглянути ширшу, інтегровану інформацію, «розрізати базу даних тонкими скибочками й нарізувати кубиками» («slice and dice»), щоб змінити вимірність, яку вони переглядають [61]. Результати «drilling» і «slicing та dicing» подаються в таблицях і схемах. Чому саме такі та подібні їм можливості стали конче потрібними ОПР та менеджерам?

Традиційна технологія підготовки інтегрованої інформації на основі запитів і звітів у звичайних системах підтримки прийняття рішень стала неефективною через різке зростання кількості і збільшення різноманітності вихідних даних, що знаходяться в численних оперативних і виробничих системах організації. Ця обставина стала гальмувати виконання функцій управління у разі необхідності швидко створювати й приймати рішення. Крім того, постійне накопичення даних у корпоративній базі даних для прийняття рішень і подальший їх аналіз гальмують оперативну роботу з даними.

У типовій організації доступні оперативні бази даних розроблялися для здійснення регулярних процедур (транзакцій). Такі потреби можуть включати, наприклад, підготовку нового або поновлення попереднього замовлення. Регулярні процедури рідко пов'язані з запитами, тому, зазвичай, оперативні бази даних містять тільки поточну інформацію. Вони не можуть задовольнити інформаційні потреби користувачів СППР, тому що їм не вистачає архівних даних та їх стабільності, що стає вирішальним у разі необхідного аналізування. Наприклад, ОПР може звернутися до системи із запитанням: «Як проведені компанією останні заходи щодо розповсюдження товару впливали на його продаж за останній квартал у порівнянні з таким самим просуванням товару за останній рік?» Швидкою та вичерпною відповіді на подібні запитання традиційні СППР забезпечити не могли.

Крім того, часто дані знаходяться в різних операційних системах, які мають суттєві відмінності в організації даних. Фактично поєднання даних із систем DB2, Oracle та COBOL з даними бази даних Sybase чи Informix може бути нелегким та ускладнюючим аналіз цих даних[62]. Оперативні бази даних за структурою та розміром є недостатніми для загального аналізу. Як результат із цього випливає, що коли потреби оперативних баз даних повністю відрізняються від потреб СППР, то вони не є оптимальними для використання в СППР, що призводить до неефективного виконання запитів до СППР.

В історичному контексті це означає, що СППР не були такими корисними, як могли б бути, оскільки необхідні дані не були вчасно доступними. Для компенсації цього недоліку часто адаптовували окремі «заморожені» фрагменти виходів інформаційної системи для наступного аналізу. Ці фрагменти забезпечують лише інформацією про *відібрані* об'єкти в *окремі моменти часу*. Хоча такі фрагменти ефективніші, ніж безпосереднє використання оперативної бази даних, але їм недостає широкого спектру інформації для повного її аналізу або гнучких специфічних запитів. Інакше кажучи, урахуовуючи всі переваги такого розв'язання проблеми, воно загалом не створює сприятливого середовища для використання СППР.

Альтернативним та популярнішим підходом є створення *сховища даних* для підтримки потреб СППР, а також застосування інструментальних засобів *OLAP-систем*, які забезпечують доступ і оброблення накопичених за достатній період внутрішніх даних організації, а також у деяких випадках і зовнішніх.

У загальному випадку дані ОДСППР є операційними даними щодо фіксації щоденних бізнесових операцій компанії, тобто дані про транзакції (ведення справ, угод) і бізнесові випадки (явища) [63]. Вони формуються, щоб забезпечити тактичні й стратегічні бізнесові дії на основі операційних і доречних зовнішніх даних. Відмінність у цілях застосування операційних баз даних і баз даних СППР полягає в тому, що формати і структури цих видів даних відрізняються принаймні за шістьма основними показниками: *структурою даних, діапазоном часу, підсумовуванням даних, мінливістю даних, вимірністю даних і метаданими*. Короткий виклад цих відмінностей наведено в табл. 5.1. Зупинимося на них докладніше.

Таблиця 5.1
Зіставлення операційних даних і даних ОДСППР

Показники	Операційні дані	Дані СППР
Структура даних	стандартизована	інтегрована
Діапазон часу	поточні	історичні (накопичувані)
Підсумовування	не здійснюється	розширюване в системах
Мінливість даних	змінюються	довготермінові
Вимірність даних	одновимірні	багатовимірні
Метадані	бажані	обов'язкові й імпортовані

Структури даних. Розглянемо діапазон і суть відмінностей операційних даних і даних ОДСППР за форматом і структурою. Почнемо з операційних даних, які часто упорядковуються системою керування реляційної бази даних. Реляційні системи оброблення запитів (транзакційні системи) мають структури даних, які називаються таблицями, що повинні бути надзвичайно

нормалізованими. Таблиці нормалізують, щоб уникнути аномалій у даних, коли виконуються, наприклад, такі операції як оновлення, додавання чи вилучення записів. Нормалізація є процесом переведення складної структури даних у найпростішу, найстійкішу структуру. За нормалізації вилучаються зайві атрибути, ключі й відношення у концептуальній моделі даних.

Задля зберігання операційних даних або даних транзакційних систем програмне забезпечення і апаратні засоби оптимізовані так, щоб підтримувати транзакції стосовно щоденних операцій компанії. Наприклад, кожного разу, коли продається виріб, то все, що з цим актом пов'язане, мусить бути записане і враховане, тобто обчислене у відповідній таблиці транзакцій. Також пов'язані з цією операцією дані – дані щодо замовників і запасів матеріалів, змінюються в системах оброблення транзакцій. Для того, щоб забезпечити ефективну й ефектну актуалізацію БД, системи транзакцій зберігають дані в багатьох малих таблицях, кожна з яких має мінімальну кількість полів. Так, наприклад, простій транзакції з оброблення запиту щодо збуту продукції потрібно мати дані, елементи яких записуються в п'яти або більше різних таблицях, оскільки потрібно добавляти або змінювати запис у таблиці накладної, в таблиці рядків накладної, таблиці дисконтів, таблиці запасів і таблиці департаменту.

Хоч такий структурований підхід до створення багатьох малих таблиць розглядається як ефективний для бази даних транзакцій, однак він не призначений для організації даних в ОДСППР. За такого підходу запити будуть виконуватися повільно, оскільки потрібно з'єднати багато таблиць, щоб завершити опрацювання запиту, на що витрачається багато часу і використовуються обширні ресурси системи.

Операційні дані, зазвичай, зберігаються в багатьох різних таблицях і містять інформацію про специфічні транзакції, а дані ОДСППР – у значно меншій кількості таблиць, причому в них не завжди можна відшукати детальні відомості щодо кожної транзакції, бо вони є переважно підсумковими даними транзакцій. Дані з численних операційних баз даних інтегровані, агреговані й підсумовані в базі даних ОДСППР, щоб задовольняти наперед невизначені потреби щодо підтримки прийняття рішень.

Орієнтовані на дані СППР можуть містити надлишкові дані, якщо це сприяє прискоренню оброблення запитів. Компонентами даних ОДСППР на основі сховища даних є: метадані, поточні деталізовані дані, давніші докладні дані, підсумовані дані тощо. Загальна нормалізація не ефективна для даних ОДСППР і навіть деяка часткова нормалізація може реально зменшити ефективність оброблення запитів у орієнтованих на дані СППР.

Діапазон часу. Операційні дані є поточними даними, оскільки вони відображають теперішній стан бізнесових транзакцій. Дані ОДСППР – це миттєві знімки станів у задані моменти часу, тобто вони являють собою упорядкова-

ну за часом сукупність або серію операційних даних. Фактично в ОДСППР зберігаються численні «часові зрізи» операційних даних.

Підсумовування. Дані ОДСППР можуть підсумовуватися за допомогою програмного забезпечення аналітичного оброблення. Можна відправити деякі дані з бази даних ОДСППР у мультимірний куб даних для прискореного аналізу [61]. Деякі бази даних ОДСППР утворюються виключно з підсумків або (як їх часто називають) похідних чи вторинних даних. Наприклад, скоріше, ніж зберігати кожну із 10 000 транзакцій зі збуту в окремих елементах пам'яті на даний період, база даних ОДСППР може містити загальну кількість проданих одиниць і обсяг збуту. Дані ОДСППР можна було б подати у такий спосіб, щоб спостерігати обсяги збуту в грошовому еквіваленті для кожного магазину або збут у різних одиницях виміру для кожного типу продукту. Метою підсумовування є визначення і оцінювання трендів продажу або порівняння збуту різних типів продукції. Користувач може захотіти, наприклад, дізнатися: який тренд продукту *A*? чи доцільно припинити продаж деякого продукту? чи були ефективними витрати на рекламу для створення сприятливих змін у збуті? На всі ці запитання можна відповідати, використовуючи інтегровані дані. Операційні дані не підсумовуються в базах даних.

Мінливість даних. Тільки два види дій відбуваються в сховищі даних або бази даних ОДСППР: завантаження даних і організація доступу до них. Можна додавати дані в пакетах, але це вже не буде інтерактивною актуалізацією даних. Отже, дані СППР не змінюються з часом. Операційні дані мають непостійний характер, оскільки вони змінюються, як тільки відбуваються нові транзакції.

Вимірність даних. Багатовимірність даних є, можливо, найхарактернішою особливістю даних ОДСППР. З погляду менеджера чи аналітика дані ОДСППР завжди пов'язані між собою багатьма різними способами. Наприклад, коли аналізується збут продукту окремому споживачеві протягом певного проміжку часу, то можна зробити такий запит: «Скільки виробів типу *X* було продано споживачеві *Y* протягом останніх шести місяців?» Дані ОДСППР можуть досліджуватися в різних аспектах, наприклад, за видами продуктів, за регіонами і за часом. Здатність аналізувати, виділяти і подавати дані як інформацію в зручному вигляді є однією із головних позитивних характеристик ОДСППР. На противагу ним, операційні дані мають тільки одну вимірність.

Метадані. В орієнтованій на дані СППР важливо розробити і підтримувати *метадані* про дані СППР. Словники баз даних можуть бути і для систем оброблення транзакцій, але через те, що дані ОДСППР можуть надходити від багатьох джерел, створення словників і метаданих є особливо важливим для СППР. *Метадані* – це «інформація про дані» в базі даних СППР. До ресурсів метаданих належать каталоги і словники бази даних, а також імена змінних, довжини полів, допустимі значення змінних і описи елементів даних. Семантичні дані часто зберігаються в словнику бази даних. Метадані зберігають інформацію про зміни у схемі початкових джерел сховища даних або бази даних.

Орієнтовані на дані СППР часто відносять до типу *аналітичних систем (АС)*, тобто інформаційних систем, метою яких є лише аналіз даних. Інколи терміни «АС» і «ОДСППР» уживають як синоніми. Зауважимо, що стосовно інформаційних процесів аналітичні системи є вторинними по відношенню до операційних транзакційних систем OLTP (On-line transaction processing), оскільки всі дані, що використовуються для аналізу, необхідно спочатку нагромадити і, за можливості, частково обробити, чим і займаються різні транзакційні системи, а лише потім їх проаналізувати. Основні відмінності систем оброблення транзакцій OLTP (онлайн-ових систем оброблення даних) і аналітичних систем (орієнтованих на дані СППР) наведені в табл. 5.2.[64].

Таблиця 5.2
Основні відмінності систем оброблення транзакцій (OLTP) і аналітичних систем

Характеристика	Онлайнова система оброблення транзакцій	Аналітична система
Метасистеми	Облік, зберігання і оперативне оброблення первинних, деталізованих даних, що характеризують поточний стан об'єктів предметної галузі (ПГ)	Отримання і зберігання узагальнених даних про ПГ і подання їх у вигляді, зручному для бізнес-аналізу та підтримки прийняття рішень
Джерела та номенклатура даних	Поточні оперативні дані, що деталізовано характеризують стан об'єктів ПГ, як правило, за останній та кілька попередніх місяців	Крім детальних, потрібні узагальнені дані за певні періоди, а також фактичні дані, нагромаджені за тривалий час. Крім внутрішніх потрібні ще й зовнішні дані
Вигляд даних	Оперативні БД можуть містити семантично еквівалентну інформацію, подану в різних форматах, яка не завжди може бути узгодженою (з причин використання різних технологій та різних СКБД)	Сховище даних має містити узгоджену інформацію, що подається в однакових форматах і максимально відповідає оперативній БД. Тобто сховища даних містять компоненти для зведення до єдиного вигляду інформації з різних джерел.
Частота оновлення	Дані є динамічними, поточними, тобто безперервно оновлюються і дуже часто змінюються	Дані є статичними, тобто вони практично не змінюються, а лише доповнюються новими записами
Характер запитів до системи	Перелік запитів до транзакційних систем відомий ще	За розв'язання аналітичних завдань переважають нерег-

	за їх проектування. Переважають регламентні запити, які детерміновані в часі, тобто створюються з певною періодичністю і мають фіксований перелік вихідних повідомлень. Зав'язання розв'язання таких задач переважають дуже часті вибірки з БД даних невеликими порціями. Транзакційні системи, головно, містять задачі прямого розрахунку	регламентні запити, які потребують оброблення великих обсягів агрегованих даних (сум, мінімальних, максимальних, середніх та інших значень показників). АС має надавати аналітику різноманітні інструменти для оброблення даних та методики аналізу (наприклад, весь спектр статистичних методів, генетичних алгоритмів, нечіткої логіки і т. п.)
Подання результатів роботи	Складання фіксованого ряду звітних форм наперед відомої структури. Переважна більшість цих звітів потребує первинної деталізованої інформації	Велика кількість різноманітних звітів на основі агрегованих даних. Надання аналітики можливості самому визначати характер і форму використовуваних звітів. Подання результатів аналізу в зручному для розуміння вигляді (графічному, табличному тощо)
Захист	Для оперативної БД, як правило, достатньо захисту на рівні таблиць	Аналітичні дані потребують більшого захисту, зокрема, на рівні окремих значень аналітичних показників
Наявність метаданих	Метаданими в OLTP-системах користуються переважно лише адміністратори систем	Репозитарій метаданих – це необхідна компонента, яка є довідником про дані сховища для користувачів системи
Необхідність перепроєктування	Бази даних транзакційної системи мають бути спроектовані так, щоб вони не потребували подальшого перепроєктування	Створення сховищ даних є ітеративним за своєю суттю і потребує регулярного перепроєктування протягом усього їхнього життєвого циклу

Аналітичні завдання залежно від концепції аналізу можна поділити на дві групи: завдання статичного та завдання оперативного аналізу. Ці дві групи аналітичних завдань суттєво відрізняються між собою. Перша група завдань характеризується тим, що вони реалізуються на основі традиційної технології автоматизації розв'язання. За цієї технології спочатку формулюється техніч-

не завдання, яке передається програмісту для програмування. Програміст складає програму та тестує її і лише після цього отримує результат у вигляді регламентованих, тобто чітко визначених форм. За такого підходу виникає велика затримка в часі між моментом виникнення потреби в аналізі та отриманням відповідного результату. Дуже часто результат аналізу, який був потрібний аналітику, отримують пізно і рішення приймається без його врахування. Тому для прийняття оперативних рішень такий вид аналізу не підходить.

Саме потреба в оперативному багатоаспектному бізнес-аналізі привела до виникнення нової OLAP-технології розв'язання аналітичних завдань. Ця технологія призначена забезпечувати аналітиків динамічним багатовимірним аналізом консолідованих даних. Як уже зазначалося, розв'язання аналітичних завдань не може обмежуватись лише даними транзакційних систем. Для порівняльного аналізу та виявлення тенденцій потрібно мати великі обсяги зовнішніх даних з різних статистичних збірників, з електронних та інших джерел. Зручним способом зберігання даних для розв'язання оперативних аналітичних завдань є сховища даних, що утворюють основу аналітичних інформаційних систем. Узагальнена схема інформаційної аналітичної системи, котра ураховує описані засади, показана на рис. 5.1.

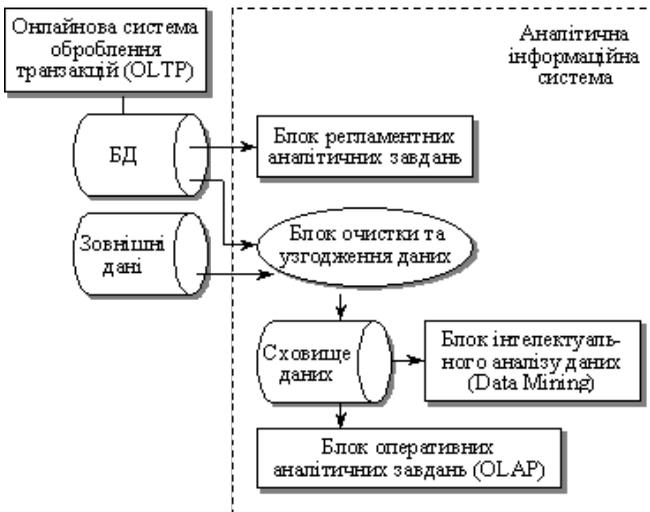


Рис. 5.1. Узагальнена схема інформаційної аналітичної системи

Орієнтовані на дані СППР мусять мати дані найвищої якості, інакше дані можуть призвести до невдач у розв'язанні проблем. Дані найвищої якості – це *точні, своєчасні, значимі* (важливі) і *повні* (комплектні) дані. Оцінювання або вимірювання якості джерел даних є попереднім завданням, яке пов'язане з оцінюванням технічної здійсності проекту орієнтованої на дані СППР.

5.1.2. Базові концепції та визначення

Зважаючи на те, що сховища даних і системи оперативного аналітичного оброблення (OLAP-системи) в принциповому плані виконують фактично такі самі функції, а саме: на основі нагромаджених за багато періодів даних про поточні ділові операції появляється можливість отримувати інформацію для створення кращих, основаних на фактах, управлінських рішень, деякі коментатори стали їх ототожнювати. Насправді, сховища даних і OLAP – різні види орієнтованих на дані СППР.

Сховище даних (Data Warehouses) є специфічною базою даних, яка проектується і наповнюється, щоб підтримувати створення рішень в організації [65]. Це є пакет, своєрідна система керування базою даних, що існує окремо від оперативних систем, обновлюється і структурується для термінових оперативних запитів і управлінських підсумків. За змістом та часовим горизонтом вона відрізняється від оперативних систем. При цьому сховище даних є незмінним у часі, а, отже, здатним підтримувати різноманітні види аналізу. Переважно такі бази даних є архівами операційних даних, відібраних для забезпечення підтримки прийняття рішень та оптимізованих для взаємодії з СППР організації. На рис. 5.2 зображена спрощена схема формування та використання сховища даних у СППР.

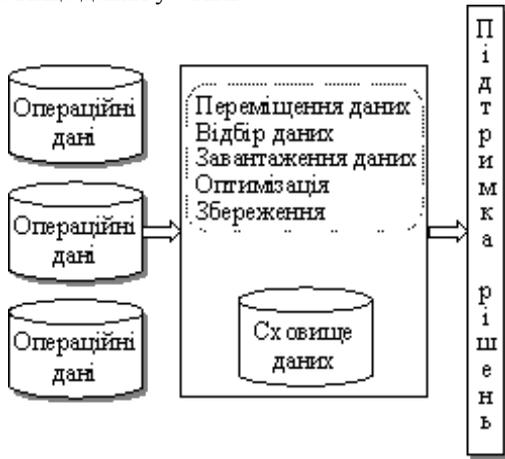


Рис. 5.2. Схема формування і використання сховища даних у СППР

Дані беруться з різноманітних джерел оперативних даних. Після переміщення проводиться їх відбір для гарантування того, що вони мають достатню значимість, є безперервними і точними. Потім дані завантажуються в реляційні таблиці, які в змозі підтримувати різноманітні види аналізу та запитів, і оптимізуються для тих таблиць, які, як очікується, будуть найчастіше застосовуватися. І, нарешті, дані зберігаються для подальшого використання в СППР.

5.1.3. Взаємопов'язана архітектура орієнтованих на дані СППР

Архітектура орієнтованих на дані СППР має низку взаємопов'язаних елементів [66]. Команди проектувальників мають починати розроблення проекту СППР з дослідження принципів побудови сховищ даних і орієнтованих на дані СППР з метою отримання архітектурного шаблону, зокрема для виявлення типових компонентів системи (наприклад, інтерфейсів), і того, як вони пристосовуються до типової організації, які є характерні чинники успіху або невдачі проекту. Після цього розробники мають накреслити карту типових даних і інтерфейсів, що можуть використовуватися за специфічних умов досліджуваної компанії, вияснити, які суб'єкти підключаються до ОДСППР та які типові запити вони можуть робити. Як мінімум, потрібно забезпечити розроблення відповідних структур для збереження даних, отримати засадні принципи добування і менеджерські інструменти фільтрування даних, мати інтерфейси інструментальних засобів створення запитів, а також деякі наперед визначені схеми і таблиці з метою використання їх у разі аналізування даних та засоби подання інформації.

Компоненти **сховища (масиву) даних** складаються з однієї або більше баз даних, збудованих з використанням реляційної СКБД, багатовимірної системи керування базою даних або обох типів цих систем. Як уже зазначалося, бізнес-дані вибираються з операційних баз даних та із зовнішніх джерел даних. Зовнішні джерела забезпечують тими даними, які не можуть бути знайдені в системах оброблення транзакцій компанії, але які доречні для аналізу бізнес-процесів, як наприклад, цінами акцій та іншими ринковими показниками. Сховище даних є компіляцією багатьох «моментальних знімків» фінансових, операційних і бізнесових ситуацій компанії. Коли формуються бізнесові дані для сховища даних, то операційні дані підсумовуються і впорядковуються в структурах, які оптимізовані з метою проведення швидкого аналізу, пошуку або вибирання даних. Процес старіння елементів у масивах даних запобігається шляхом переміщення поточних детальних даних на місце застарілих.

Компоненти, що забезпечують виділення і фільтрування даних використовуються для вибирання і перевіряння достовірності даних, отримуваних від операційної бази даних і зовнішніх джерел. Наприклад, для визначення відносної частки ринку для вибраних асортиментів продукції СППР потребує даних стосовно продуктів конкурентів. Такі дані можуть бути розміщені в зовнішніх базах даних, які створюються й підтримуються за допомогою індустріальної групи або компаній, що торгують на ринку такими даними. Як це впливає з назви, ці компоненти вибирають дані з різних джерел, фільтрують їх, щоб підібрати доречні дані, і форматують так, щоб можна було їх додавати до сховищ даних СППР.

Аналітик даних або менеджер можуть створювати запити для доступу до бази даних СППР за допомогою спеціального **«інструментального засобу запитів кінцевого користувача»**, інтерфейс якого відповідно настроюється для полегшення використання.

Інструментальні засоби кінцевого користувача для аналізу та подання інформації допомагають менеджеру виконувати обчислення і вибирати найвідповідніший формат подання. Наприклад, менеджер може бажати отримувати зведені звіти у вигляді таблиць, карт, секторних або стовпчикових діаграм. Інструментальний засіб запити та інструментальний засіб подання належать до зовнішнього інтерфейсу СППР. Клієнт/серверна технологія забезпечує можливість цим компонентам взаємодіяти з іншими, щоб сформувати завершену архітектуру СППР.

Як тільки архітектура програмного забезпечення розроблена для СППР, що призначається для досягнення наперед визначених цілей конкретної компанії, то потім виникає багато запитань, пов'язаних з проектуванням, розробленням та реалізацією нової орієнтованої на дані системи підтримки прийняття рішень.

5.1.4. Загальне проектування і процес розроблення орієнтованих на дані СППР

Різні автори по-своєму налагоджували процес розроблення орієнтованих на дані СППР [67]. Раніше були розглянуті два загальні підходи до побудови СППР: «розроблення життєвого циклу системи» і «швидке макетування». Для відносно малих проектів ОДСППР, подібних до вітрин даних, доцільно застосовувати швидке макетування, а для створення проектів корпоративних сховищ даних або ОДСППР можна здійснювати п'ять загальних, *орієнтованих на рішення*, кроків.

Перший крок – початкове збирання даних або діагностика. Цей крок передбачає ідентифікування й інтерв'ювання головних майбутніх користувачів СППР, визначення провідних тем СППР, ідентифікацію моделі даних транзакцій, визначення рівня монопольного використання даних, частоти оцінювання і оновлення системи, вимог до інтерфейсу кінцевого користувача, а також форм подання даних. За розроблення проекту головну увагу слід приділяти творцям рішень і самим рішенням протягом усіх кроків.

Другий крок – проектування і створення карти (Mapping) розміщення масиву даних (Data Store). У середовищі реляційної СКБД спочатку необхідно вибрати зіркоподібну схему структури даних і виявити факти, виміри, атрибути, а потім створити діаграми зіркоподібних схем, ієрархію атрибутів і рівні агрегування. Ці концептуальні моделі слід подати у вигляді реляційних таблиць. У середовищі багатовимірної бази даних мають бути визначені ключові змінні і виміри. В базу даних вмонтовують доречні для СППР дані.

Третій крок – завантаження і тестування даних. Створення бази даних СППР включає підготовку до введення даних, визначення переліку початкових даних для завантаження і процесу актуалізації або оновлення. В той же час

аналітики визначають необхідні перетворення транзакцій і будь-яких зовнішніх даних, таблиць операційних даних транзакцій, інтегрують і трансформують дані. Дані завантажуються, індексуються й перевіряються на достовірність і, наразті, перевіряють метадані та куби даних або зіркоподібні схеми.

Четвертий крок – побудова і випробування орієнтованих на дані СППР. Аналітикам потрібно створити меню, розробити вихідні формати даних, визначити передбачувані запити, підготувати тести для інтерфейсів і результатів, знайти оптимальні рішення щодо швидкості й точності роботи системи. При цьому вони мусять спілкуватися з кінцевими користувачами протягом макетування і тестування елементів проекту, підготувавши їх до використання середовища розроблення СППР. У такий спосіб творці рішень мають бути серйозно залученими до процесу побудови і випробування нової, орієнтованої на дані СППР.

Останній крок – розгортання (Rollout) і зворотний зв'язок (Feedback). Цей крок передбачає реальне розгортання СППР, забезпечення додаткової підготовки, установа зворотного зв'язку з користувачами, супроводження системи і в багатьох випадках також її розширення та вдосконалення. За такого підходу є підстави сподіватися, що СППР дасть змогу удосконалити створення рішень і тим самим принесе користь як компанії загалом, так і творцям рішень зокрема.

Контрольні питання до підрозділу 5.1

1. *Надайте визначення орієнтованій на дані системі підтримки прийняття рішень (ОДСППР).*
2. *У чому полягала проблема, що призвела до створення сховищ даних?*
3. *За якими шістьма основними показниками відрізняються формати і структури операційних баз даних бізнесу і баз даних СППР?*
4. *Що таке агреговані дані?*
5. *Які особливості визначають таку базу даних, як «Сховище даних» (Data Warehouse) ?*
6. *Звідкіля збираються бізнес-дані для сховища даних?*
7. *Що таке «метадані » в СППР та які їх ресурси Вам відомі?*
8. *Якими чотирма характеристиками володіють дані найвищої якості?*
9. *У чому полягає взаємопов'язаність архітектура орієнтованих на дані СППР?*
10. *Перелічить п'ять, орієнтованих на рішення, кроків розроблення орієнтованих на дані СППР?*

5.2. Концепція сховищ даних, її реалізація в інформаційних системах

5.2.1. Побудова сховищ даних

Сховище даних (Data Warehouse) являє собою предметно-орієнтоване, прив'язане до часу і незмінне зібрання даних для підтримання процесу прийняття керуючих рішень [68]. Дані у сховище надходять з оперативних систем, призначених для автоматизації бізнес-процесів, а також із зовнішніх джерел, (наприклад, зі статистичних звітів). Головним призначенням сховища є надання інформації для аналізу в одному місці та в інтуїтивно зрозумілій структурі.

Практика свідчить, що різні компанії та організації будують сховища даних з метою створення своїх специфічних додатків, причому в багатьох із них перше сховище даних, зазвичай, призначається для виконавчих інформаційних систем. Сховище даних потенційно орієнтоване на різні джерела (рис. 5.3), включаючи операційні бази даних або дані транзакцій. Інші внутрішні дані, які надходять від електронних таблиць і внутрішньокорпоративних документів, та зовнішні дані, як наприклад бази даних новин чи цін акцій, можуть також бути збережені в сховищі даних. У типовій організації операційні дані розпорошені через використання кількох СКБД у дуже різних форматах і на різних апаратних платформах. Наприклад, операційні дані можуть міститися у великих ЕОМ (мейнфреймах) типу IBM або DEC і знаходитися у файлах інформаційних систем менеджменту (на рис. 5.3 – IMS), забезпечуватися віртуальним методом доступу до файлів даних (VSAM), або знаходитися в СКБД DB2 і базах даних. Успадковані інформаційні системи, як наприклад ті, що зображені на рис. 5.3 з ілюстративною метою, є загальним джерелом даних для сховищ даних.

Дані, які вибираються зі щойно названих джерел, необхідно «очистити» для того, щоб перетворити їх формат на прийнятний з метою використання в сховищі. Інакше кажучи, дані мають подаватися у форматах, відповідних тим додаткам, для яких розробляється сховище даних. Наприклад, якщо деякі потрібні операційні дані рідко використовуються на індивідуальному транзакційному рівні, то їх потрібно підсумувати або агрегувати перед збереженням у сховищі.

Важливим чинником стимулювання організації сховищ даних у корпораціях стала поява спеціального програмного забезпечення, призначеного для побудови та зберігання сховища даних, а також для забезпечення доступу до них [69]. Як видно із рис. 5.3, до групи продавців цих програмних продуктів належать як добре відомі розробники СППР та ВІС (наприклад Pilot, Comshare, IRI) і баз даних (наприклад, Oracle, Focus), так і менш знайомі продавці (RedBrick, Platinum, Carleton), які мають значний вплив у сфері організації сховищ даних. У табл. 5.3 подані основні розробники та створені ними інструментальні засоби, призначені для побудови окремих компонентів та виконання функцій сховищ даних. Проте слід зауважити, що кількість як самих розробників, так і інструментальних засобів створення сховищ даних постійно зростає, так само як зростають обсяги продажу програмних продуктів сховищ даних, які 2000 року перевищили 5 млрд доларів США.

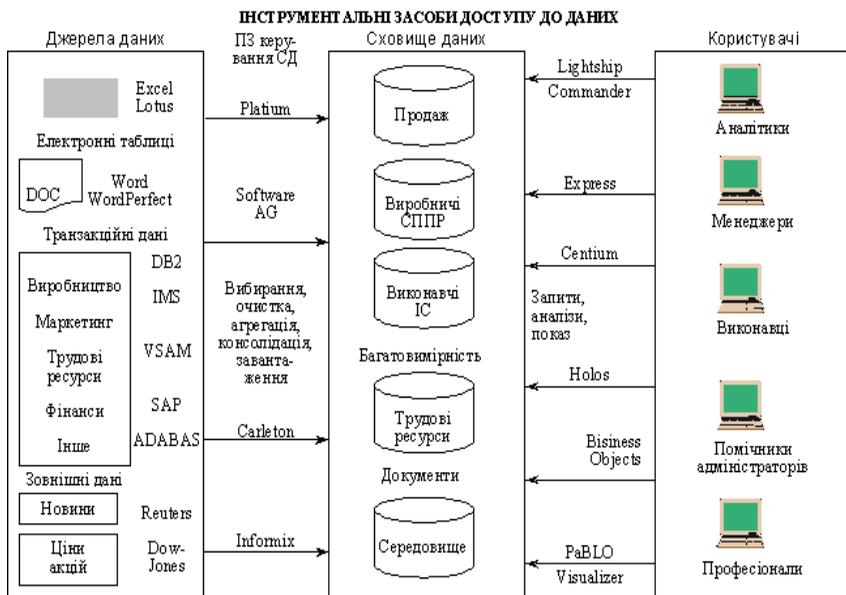


Рис. 5.3. Будова і використання сховищ даних

Таблиця 5.3

Інструментальні засоби та їх виробники для створення сховищ даних

Функція (елемент) сховищ даних	Виробники та їхні інструментальні засоби
Проектування	Oracle: <i>Designer/2000</i> Sybase: <i>Power Designer</i>
Доставка даних	Oracle: <i>Web Server Spy Glass</i> IBM: <i>Lotus Notes, WWW</i> Sybase: <i>Sybase Enterprise Connect, Replication Server, Omni Connect</i>
Підготовка даних	Oracle: <i>Open Gateways, Symmetric Replication, Parallel Loader</i> IBM: <i>Data Propagator Relational, Data Refresher, Data Propagator Non-Relational, Vality's Integrity</i> Sybase: <i>Carleton Passport, Informatica Power Mart</i>
Перетворення даних	Prism: <i>Warehouse Manager</i> Carleton: <i>Passport</i> Apertus: <i>Enterprise Integrator</i> ETI: <i>Extract</i>

	Harte Hanks: <i>Trillium</i> Platinum: <i>Transport, Info Refiner</i>
Тиражування даних	Platinum: <i>Into Pump</i>
Керування даними	Microsoft: <i>SQL Server</i> Oracle: <i>Oracle</i> Sybase: <i>Sybase IQ, Sybase SQL Server 11, Sybase MPP</i> Informix: <i>Informix</i> IBM: <i>DB2 PE, DB2/400 SMP, DB2/MVS</i> (включаючи <i>ParallerSysplex</i>)
Керування метаданими	IBM: <i>Data Guide</i> Sybase: <i>Prism Warehouse Manager, Intellidex Meta Center</i>
Добування знань	Angoss: <i>Knowledge Seeker</i> SABRE: <i>Decision Technologies</i> ISL: <i>KDW/Clementine</i>
Доступ до даних	Andyne: <i>GQL/Pablo</i> Business Objects: <i>Business Objects</i> Cognos: <i>Power Play, Impromptu</i> Information Advantage: Decision Suite Micro Strategy: <i>DSS Agent/Server</i> Oracle: <i>Express</i> Platinum: <i>Info Beacon, Forest & Trees</i> Sybase: <i>PowerBuilder, Info Maker</i>
Доступ до даних	Software AG: <i>Esperant</i> Oracle: <i>Developer/2000, Discoverer, Oracle Express</i> IBM: <i>Lotus Approach, Intelligent Decision Server, Arbor</i> <i>Essbase, Information Advantage Decision Suite, Cognos</i> <i>Impromptu і Powerplay, Business Objects і Pilot Discovery Server</i>
Керування	CA: <i>UnicenterTNG</i> NCR: <i>Teradata Tools, Top End</i> Oracle: <i>Enterprise Manager</i> Sybase: <i>Warehouse Architect</i>
Проміжний шар	Sybase: <i>ODBC, JDBC, Netimpact Dynamo,</i> <i>Sybase Open Client/ Open Server</i> IBM: <i>Data Joiner</i>

Сховище даних містить тільки моментальні знімки фактичних даних на конкретний день, як, наприклад, на останню п'ятницю або на останній день місяця. Обновлення даних залежить від потреб користувачів і бізнесового циклу, який асоціюється з даними. Наприклад, торговому менеджеру, який

установлює ціни на кожний день, необхідне оновлення даних щодня або навіть частіше. Фінансовий аналіз на кінець місяця потребує тільки помісячної актуалізації даних. Технологічний цикл оновлення даних у сховищі, як правило, триває менше однієї години, тому оновлення суттєво не впливає на поточні бізнесові дії та їх результати.

Як уже зазначалося, сховище даних також містить *метадані* – інформацію про дані, тобто відомості про те, звідки й від кого дані надходять, хто має доступ до них і як часто, які бізнесові процеси вони забезпечують і які критичні фактори успіху вони підтримують. Ці метадані життєво важливі для підтримки сховищ даних і для кінцевих користувачів, яким потрібно знати, як розмішувати дані.

Дані, які містяться в сховищах даних, можуть бути доступними через програмне забезпечення клієнта, що використовується для підтримки прийняття рішень. Воно може призначатися для ВІС (наприклад, Lightship, Commander), СППР (наприклад, Visual IFPS, Express), або незапланованих (ad hoc) запитів (наприклад, Business Objects, Visualizer). У всіх випадках мається на увазі, що програмне забезпечення легке для користування, так що організаційний персонал з обмеженою комп'ютерною майстерністю і досвідом може ефективно його застосовувати.

Незважаючи на те, що з формального погляду сховище даних являє собою різновид звичайної БД, котра призначена для зберігання в погодженому вигляді фактичної інформації, що поступає з різних оперативних систем та зовнішніх джерел, процес їх проектування суттєво відрізняється.

Для звичайних БД процес їх створення відбувається за схемою: вивчення предметної галузі; побудова інформаційної моделі; розроблення на основі інформаційної моделі проекту бази даних; створення бази даних. Обов'язкові етапи створення сховищ даних відрізняються від такої схеми. Такими етапами є: визначення інформаційних потреб користувачів щодо даних, котрі нагромаджуються в базах даних операційних систем (OLTP-систем), що служать джерелами оперативних даних; вивчення локальних баз даних OLTP-систем; виокремлення від кожної бази даних підмножини даних, необхідних для завантаження в сховище даних; інтегрування локальних підмножин даних і розроблення загальної погодженої схеми сховища. Для виконання робіт зі створення сховищ даних за даною схемою існують різні інструментальні засоби, зокрема, програмний продукт «Oracle Designer» та його спрощена версія «Oracle Data Mart Designer» [70]. Коли сховища даних уже створені та оптимізовані, необхідно ефективно завантажувати нові дані в систему без переривання процесу підтримки прийняття рішень. Однак у разі збільшення обсягу даних розробникам необхідно визначити нові синтаксичні формати та формати запитів, які є швидшими й легшими, а також визначити нові підходи до поєднання реляційних таблиць та добування даних з цих дуже великих баз даних з використанням різновиду програмних агентів – інтелектуальних («розумних») агентів (Intelligent agents), що розглянуті раніше.

5.2.2. Архітектура сховищ даних

Узагальнена схема архітектури сховища даних

Оскільки, як буде описано нижче, кожна велика компанія-розробник сховищ даних пропонує своє бачення їх архітектури і ряду відповідних інструментальних засобів, то є певний сенс розглянути деяку узагальнену архітектуру сховищ даних, не прив'язуючись до реально наявної системи [71]. Така узагальнена архітектура показана на рис. 5.4, де виділені окремі компоненти, інструментальні засоби та джерела сховища даних. Опишемо деякі з них.

Менеджер завантаження, якого часто називають зовнішньою компонентою сховища даних, виконує всі операції, пов'язані з вибиранням необхідних даних та їх завантаженням до сховища. Функції менеджера завантаження полягають в очищенні, конвертації та зведенні даних до стандартного вигляду для їх подання в сховищі даних (СД).

Менеджер сховища виконує операції аналізу та керування даними. До таких основних операцій належать: аналіз узгодженості та відсутності суперечливостей у даних; перетворення та переміщення даних з тимчасового сховища в основні таблиці СД; створення індексів; денормалізація даних за необхідності; часткове чи глибоке узагальнення даних; резервне копіювання й архівування даних.

Менеджер запитів – це внутрішній елемент сховища даних, який виконує всі операції, що пов'язані з керуванням запитів користувачів. Він є складовою частиною СКБД, яка підтримує сховище даних.

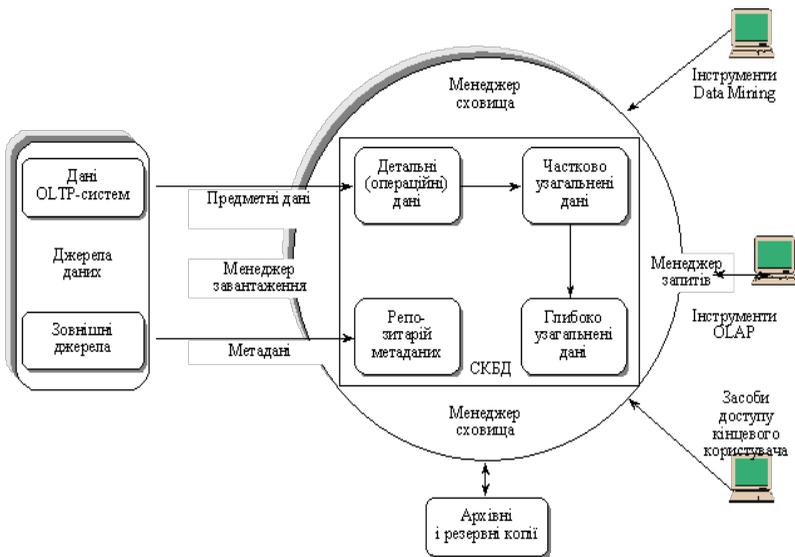


Рис. 5.4. Узагальнена архітектура сховища даних

Детальні (операційні) дані – ця складова компонента архітектури містить усі детальні дані, які визначені схемою сховища даних. Це можуть бути як первинні дані найнижчого рівня деталізації, так і узагальнені до певного рівня агрегування.

Частково і глибоко узагальнені дані – ці елементи містять дані, які попередньо оброблені менеджером сховища з метою їх часткового чи глибокого узагальнення. У цій частині зберігаються у певний спосіб відсортовані та згруповані дані, необхідні для виконання запитів. Дана частина сховища є тимчасовою і змінною, так як вона постійно модифікується відповідно до змін у запитах. Необхідність цієї компоненти пов'язана з підвищенням продуктивності виконання запитів. Узагальнені дані обновляються у міру надходження нових даних до системи.

Репозитарій метаданих – це інформація про дані, що зберігаються в сховищі даних. Структура метаданих може відрізнятися залежно від їх призначення. Метадані використовуються для таких основних цілей:

- **вибирання і завантаження даних.** Метадані містять інформацію про джерела даних, про способи та періодичність їх вибирання і завантаження в СД;
- **обслуговування сховища.** Метадані використовуються для автоматизації процедур узагальнення даних;
- **обслуговування запитів.** Метадані використовуються для визначення переліку таблиць для виконання запитів.

Визначаючи програмно-технологічну архітектуру сховища даних, потрібно мати на увазі, що система підтримки прийняття рішень, на яких би візуальних засобах вона не ґрунтувалася, має надавати користувачеві можливість деталізування інформації, тобто операцію Drilldown. Керівник підприємства чи фірми, отримавши дані в інтегрованому вигляді й висновки, зроблені на їх основі, може зажадати детальніших даних, що уточнюють джерело даних або причини висновків. З погляду проектувальника сховищ даних, це означає, що в деяких випадках необхідно забезпечувати взаємодію бази даних із системою оброблення транзакцій.

Альтернативні (фірмові) архітектури сховищ даних

Ідея сховищ даних (СД), запропонована Б. Інмоном [72], і концепція оперативного аналітичного оброблення даних (OLAP), розроблена Е. Коддом, вдало доповнили одна одну. За минуле десятиріччя аналітики розробили біля десятка аналітичних інформаційних систем різної архітектури на основі сховищ і вітрин даних, призначених для підтримки прийняття рішень і аналітичних досліджень. До них належать:

- віртуальні сховища даних; незалежні вітрини даних;
- централізовані сховища даних;
- інмонова модель з кулями детальних і консолідованих даних;
- розширена інмонова модель з персональними вітринами даних;
- інверсна інмонова модель;
- централізоване сховище з накопиченням даних у незалежних вітринах;
- централізоване сховище з тематичними вітринами даних;

- централізоване очищення даних з паралельними сховищами і вітринами даних.

Відповідно до цього провідні фірми пропонують свої рішення, що ґрунтуються на продуктах, які розробляються та випускаються ними (див. табл. 5.3). У створенні великих сховищ даних лідирують корпорації IBM, Informix, NCR, Oracle, RedBrick, SAS, Sybase, Microsoft. Крім того, на ринку таких продуктів для побудови і використання сховищ даних значне місце займають Brann Software, Business Objects, Cayenne Software, Computer Associates, Micro Strategy, Prism Solutions, Brio Technology, Cognos, Platinum Technology.

IBM: Visual Warehouse

Корпорація IBM (International Business Machines) належить до тих компаній, що надають повний перелік послуг, програмне і апаратне забезпечення, необхідне для побудови сховищ даних [73]. Під назвою *Visual Warehouse* фірма IBM пропонує архітектурні варіанти вітрин даних і програмного компонента для їх створення. Це технологія збору даних з різних транзакційних систем, локальних і віддалених плоских файлів, великих блоків двійкових об'єктів (Binary Large Object Block – BLOB) та інших джерел. Пакет Visual Warehouse містить інтегровані програмні продукти, відповідні різним рівням архітектури. Дані з цих джерел трансформуються за правилами метаданих, що визначаються за допомогою графічного інтерфейсу користувача на платформі Win – OS/2 або Microsoft Windows. На основі одного або кількох джерел для користувача або групи користувачів можна підготувати бізнес-огляди.

Informix: MetaCube

Компанія «Informix» пропонує сервер Informix Dynamic Server і п'ять опцій для розширення його функцій: Advanced Decision Support, що підтримує оптимізацію оброблення запитів за допомогою спеціалізованих індексів підтримки рішень (Decision Support Indexes); Extended Parallel для використання InformixDynamic Server у багатопроцесорних комплексах різної архітектури; Meta Cube, що забезпечує багатовимірний аналіз інформації в реляційних інтерактивних системах аналітичного оброблення (OLAP); Universal Data Option для підтримки нових типів даних; WebIntegration Option для інтеграції баз даних з Web-серверами.

NCR: Scalable Data Warehouse

Компанія «NCR» володіє досить відпрацьованою методикою, в яку вкладено весь п'ятнадцятирічний досвід створення і впровадження більше 600 сховищ даних. Фірмі належить рекорд з розроблення найбільшого у світі сховища (від 7 до 24 Тб різних даних). Основою технології її продукту *Scalable Data Warehouse* є реляційна СКБД NCR Teradata, розроблена спеціально для архітектури з масовим паралелізмом і яка функціонує під керуванням ОС UNIX SVR4. Наявна також можливість перенесення СКБД Teradata на ОС Windows NT корпорації «Microsoft» і Solaris фірми, «Sun Micro systems». Ця технологія дає змогу будувати сховища даних на основі СКБД Informix, SQL Server і Oracle. Як апаратне забезпечення NCR пропонує свої сервери World Mark 5100, масштабована архітектура яких полегшує розширення

сховища. Компанія також надає консультаційні послуги щодо підготовки архітектурного проекту сховища даних, його реалізації і керування ним.

Oracle: архітектура мережевих обчислень

За побудови корпоративних інформаційних сховищ Oracle використовує традиційну архітектуру, що реалізує довільний доступ до будь-яких даних з різних джерел. Розроблена корпорацією «Oracle» архітектура мережевих обчислень дає основу для переходу від принципу клієнт – сервер до концепції Web. Ця архітектура містить п'ять логічних куль. Кулью джерел утворюють транзакційні бази даних, успадковані додатки на мейнфреймах, додатки клієнт-сервера, плоскі файли й інші зовнішні джерела даних. Інформація вибирається із джерел, перетворюється, денормалізується і транспортується в сховище або вітрину даних.

Продукти інституту SAS

На відміну від основних постачальників програмного забезпечення для сховищ даних SAS Institute пропонує організувати сховища не на основі реляційних СКБД, а в SAS-наборах, підтримуючих пакетне завантаження і читання великих обсягів даних [74]. SAS-набори – це аналоги таблиць у реляційних СКБД, що являють собою файли обсягом до 2 Гб для деяких ОС UNIX, які можуть розташовуватися в різних каталогах і на різних дисках. Сховище утворюється з безлічі таких наборів і досягає 3 Тб. Архітектура сховищ SAS зображена на рис. 5.5 (термін *cronfunction* означає функцію планування з економією часу і затримкою завдань), а інструментальні засоби для їх створення перераховані в табл. 5.4. Побудова сховищ даних, згідно з методикою SAS Institute, включає процеси завантаження, керування даними і експлуатації сховища. За допомогою продуктів SAS можна створити централізоване, розподілене або віртуальне сховище. Модульний принцип цього програмного забезпечення дає змогу використовувати його в різній архітектурі.

Таблиця 5.4

SAS-засоби створення сховищ даних

Компонент сховища даних	Інструментальні засоби для його створення
Проектування	DataWarehouseAdministrator
Підготовка даних	BASE/SAS, SAS/FSP, SAS/Access
Керування даними	SAS System
Доставка даних	SAS/Connect, SAS/SPDS, SAS/Share
Управління	DataWarehouseAdministrator
Доступ до даних	SAS/MDDB, SAS/EIS, EnterpriseMiner, EnterpriseReporter, SAS/Assist, SAS/Insight, SAS/Graph, SAS/AF. SAS/Stat, SAS/ETS

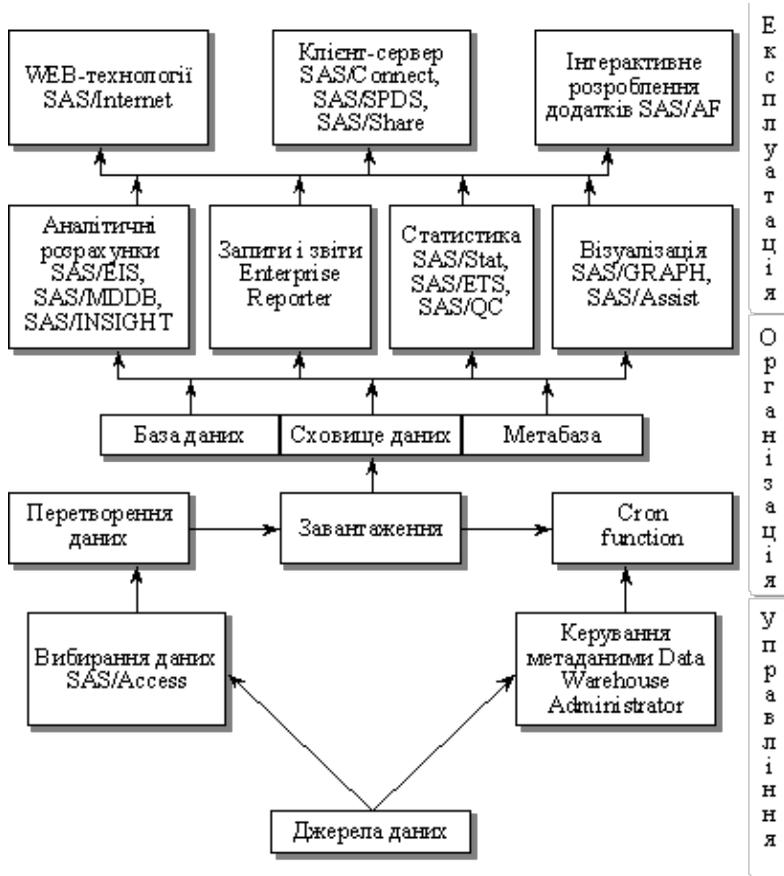


Рис. 5.5. Архітектура сховища даних інституту SAS

Sybase: Warehouse NOW

У компанії «Sybase» централізований підхід вважають досить ризикованим, тому основний акцент тут переноситься на створення розподілених вітрин даних, які надалі можуть бути об'єднані в централізоване сховище.

Для проектування корпоративних сховищ даних є сімейство продуктів **Power Designer**, що складається з шести інтегрованих модулів: *Process Analyst* – для дослідження потоків даних; *Data Architect* – для послідовного розроблення концептуальної і фізичної моделей; *App Modeler* – для створення фізичної моделі даних і об'єктів додатків; *Meta Works* – для групової роботи, спільного доступу до інформації і керування моделлю; *Warehouse Architect* –

для проектування сховищ даних; *Viewer* – для графічного перегляду інформації про модель.

Microsoft Data Warehousing Framework

Однією з останніх розробок у галузі створення сховищ даних є пропозиція корпорації «Microsoft» під назвою «Data Warehousing Framework». Кілька років тому «Microsoft» дійшла висновку про існування насущної потреби в ряді інтегруючих технологій, які б забезпечували легкість спільної роботи продуктів різних виробників. Таке розуміння привело до створення концепції Microsoft Data Warehousing Framework (інфраструктура побудови сховищ даних Microsoft), що являє собою план розвитку не тільки майбутніх продуктів Microsoft, таких як SQL Server 7.0, але й технологій, необхідних для інтеграції продуктів багатьох сторонніх виробників, включаючи як ділових партнерів Microsoft, так і її конкурентів.

Висновок

Розглядаючи фірмові архітектури сховищ даних, можна дійти несподіваного висновку: ці рішення не конкурують одне з одним, а швидше адресовані різним сегментам ринку. Більше того, за створення корпоративної інформаційної системи на основі сховищ даних можлива її модульна побудова з використанням програмного забезпечення різних фірм. Таке рішення може найповніше врахувати специфіку конкретної організації, її потреби, фінансові можливості, наявність кваліфікованих фахівців з програмного забезпечення. Тому важко розробити конкретні рекомендації щодо вибору тієї чи іншої архітектури та інструментальних засобів побудови сховища даних. Можна лише дати деякі орієнтири для цього, виходячи з обмеженої низки міркувань.

Якщо важливе значення мають конфіденційність і безпека даних, що розміщуються в сховищі, то найкраще використати програмно-апаратні комплекси IBM, що зарекомендували себе як найзахищеніші. Програмні рішення на основі продуктів фірми «Informix» відомі своєю невисокою ціною і їх можна рекомендувати для організацій з обмеженим бюджетом. Фірма «NCR» володіє не тільки рекордним сервером World Mark і СКБД Teradata, але і величезним досвідом створення сховищ: її послугами можна скористатися як для реалізації терабайтних проектів, так і для отримання консультацій.

Сервер баз даних Oracle реалізований практично на всіх апаратних платформах, що в сукупності зі стійким станом фірми дає змогу рекомендувати його для використання в довготривалих проектах, розрахованих на масштабованість. Фірма «SAS» крім засобів створення сховищ пропонує один з кращих пакетів аналітичної і статистичної обробки даних, який може бути використаний на робочому місці в поєднанні з будь-яким сховищем або вітриною даних. Рішення фірми «Sybase», відомі своєю швидкодією, являють особливий інтерес для тих, хто зупинив свій вибір на розподілених незалежних вітринах даних. Середовище розроблення сховищ даних корпорації «Microsoft» забезпечує по-справжньому потужну платформу для бізнес-аналізу.

5.2.3. Моделі побудови сховищ даних

Багатовимірна модель

Найвдалішою формою подання даних, що надає можливість багатовимірної їх параметризації, є не плоска реляційна модель, а багатовимірна. Базовані на сервері системи інтерактивного аналітичного оброблення на основі багатовимірних моделей мають назву «MOLAP-системи» (Multidimensional OLAP) [75]. Мультивимірне зображення даних може бути подане у вигляді кубів і/або гіперкубів, де дані звично розміщені в клітинках пам'яті на перетині причино зумовлених або описово доречних вимірів [61]. Куби і/або гіперкуби фактичних поточних значень формуються з двовимірних файлів або реляційних баз даних. Вимірність масиву характеризує значення в елементах пам'яті. Значення в елементах пам'яті ніколи не бувають значеннями вимірних категорій. Замість цього вони є значеннями атрибутів, які змінюються через розмірність. На рис. 5.6 зображено приклад трьохвимірного масиву даних – куба, який заповнюється значеннями проданих одиниць продукції в різний час з відповідними значеннями прибутковості. Наприклад, виділений на рисунку елемент 181,0 означає, що продукція А в період з 18 до 24 доби була реалізована в обсязі 181 одиниці, при цьому прибуток від реалізації одиниці вимірювання цієї продукції був у межах від 40 до 60 копійок.

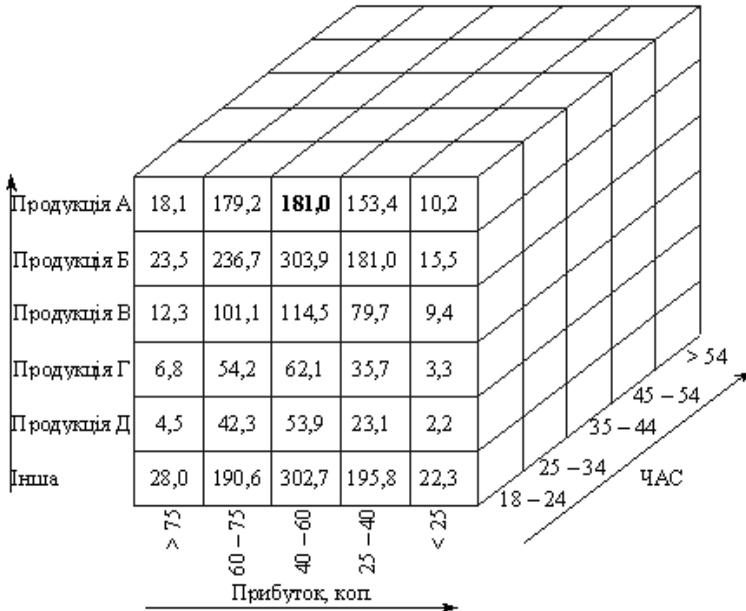


Рис. 5.6. Приклад трьохвимірного куба, комірки якого заповнені даними від продажу продукції в різні періоди з відповідною прибутковістю

Змінні, які часто поміщають у масиви даних, є економічними показниками, як, наприклад, ціни, витрати, обсяги збуту, прибутки, ймовірності сприятливих відповідей і середні значення життєвих циклів споживачів (клієнтів).

Багатовимірні бази даних можуть мати повний перелік логічних операцій, які застосовуються в реляційних базах даних. Крім того, однак, є специфічні логічні відносини й дії, які набагато легше виконувати в багатовимірних базах даних, оскільки вони постійно «зашиті» в проектах комерційних продуктів і тому не потрібно проводити дуже вправних і точних маніпуляцій з фундаментальнішими логічними операціями. До таких дій належать:

- визначення відношень спадкоємності типу батько-дитина у межах вимірності і конструювання вимірної ієрархії в розрізі територій, підприємств, часу та інших важливих аспектів;
- легке виконання матричних обчислень, які дають змогу оперувати зразу всіма векторами масивів;
- ранжування (subsetting) – організація підмножин багатовимірних масивів для забезпечення сфокусованих описів, звітів і аналізів;
- ротація (обертання) для дослідження різних зображень багатовимірного масиву без проведення повторного компонування з початкових даних;
- агрегування чи деагрегування (disaggregating) багатовимірних масивів для показу відповідно вищого або нижчого рівнів їх ієрархії як, наприклад, періоду територіально-адміністративного поділу або підприємства (операції також відомі як «rolling-up» (згортання) чи «drilling-down» (розкриття)).

Багатовимірні бази даних використовуються разом з реляційними базами даних за допомогою або дружніх мов запиту, або візуальних інструментальних засобів, що дають змогу формувати практичні запити на даний випадок (ad hoc) до бази даних аналітикам з маркетингу, аналітикам зі збуту, фінансовим аналітикам та іншим фахівцям.

Багатовимірна модель сховища даних допускає подальше ускладнення моделі даних, яке може виконуватись у кількох напрямках:

- збільшення кількості вимірів, наприклад, дані про продаж можна аналізувати не тільки за часом, видами товарів і прибутковістю, а й за регіонами. У такому разі вони утворюють чотирьохвимірну модель;
- ускладнення вмісту комірки – наприклад, у комірках розташовуються не лише обсяги продажу, а й прибутки та залишки на складі. Тоді в комірці буде кілька значень.

Показники, що знаходяться в комірках багатовимірної моделі, можна відшукувати за будь-яким виміром чи їх комбінацією. Тобто, аналізуючи обсяги продажу, можна підрахувати кількість виробів, проданих протягом останнього року, підсумувавши дані про обсяги продажу за кожен місяць. Якщо необхідно з'ясувати, який з регіональних відділів продажу мав кращі результати, то можна здійснити пошук за регіонами і знайти максимальний обсяг продажу по певному регіону.

Моделі сховища даних реляційного типу

Альтернативним варіантом побудови сховища даних є моделі реляційного типу [76], що утворюють основу реляційних систем інтерактивного аналітичного оброблення (OLAP) – ROLAP. Продавці ROLAP приєднуються до доктрини OLAP, згідно з якою кінцеві користувачі повинні мати можливість презентувати багатовимірне зображення бізнесових даних. Але вони сильно розходяться в думках щодо того, чи необхідно фізично зберігати дані багатовимірні, чи достатньо використовувати логічну модель, яка трансформує дані в куб або гіперкуб для забезпечення багатовимірного подання даних. Фахівці ROLAP розробили принципи вимірного моделювання на основі застосування таких видів структур моделей даних, як зіркоподібна (StarSchema), «сніжинка» (Snow flake Schema), сузір'я (Constellation) і якісна (Qualities). Найчастіше застосовуються перші два види структур, тобто «зірка» чи «сніжинка».

Згідно з концепцією засновника сховищ даних Б. Інмона (Bill Inmon) «сховище даних є тематично орієнтованою, інтегрованою, динамічною (time-variant), довготривалою сукупністю даних для підтримання процесів менеджменту, котрі розробляють рішення». Р. Кімбал (Ralph Kimball, 1996), інший піонер створення сховищ даних, вважає, що «сховище даних містить копії даних транзакцій, специфічно структурованих для запитів і аналізу».

Головними властивостями сховищ даних, які виділені Інмоном, є:

Тематична (суб'єктна) орієнтованість – означає фокусування на темах, пов'язаних з бізнесом або організаційною активністю суб'єктів подібно діям споживачів (клієнтів), службовців і постачальників, тобто сховища даних мають орієнтуватися на предметну галузь, а не на специфіку програмних засобів, які будуть взаємодіяти з цими додатками.

Інтегрованість – означає, що дані зберігаються в узгодженому форматі завдяки використанню іменованих правил (умовних позначень, домовленостей), обмежень домена (області допустимих значень, сфери дії), фізичних атрибутів і вимірів.

Динамічність (time-variant) – це відповідність даних певним моментам часу, тобто це є часові ряди (або серії), що не мають поточного статусу.

Довготривалість – означає, що дані тільки читаються і не змінюються з часом, як тільки вони переміщуються в сховище,

і зберігаються для підтримки рішень.

Крім цих характеристик існують і інші важливі властивості сховищ даних, зокрема:

Підсумовуваність даних – операційні дані набувають придатності для підтримки рішень форми, тобто бази даних утримують агреговані значення для управлінських рішень як окремий відбиток від баз даних, які використовуються для онлайнового оброблення транзакцій – OLTP (On Line Transaction Processing).

Масштабність – мається на увазі, що в сховищах даних зберігається набагато більше даних, ніж у звичайних базах даних.

Ненормалізованість – дані в сховищах можуть бути надмірними, ненормалізованими.

Наявність метаданих – метадані в СППР на основі сховищ даних обов'язкові. Ці метадані життєво важливі для підтримки сховищ даних і для кінцевих користувачів, яким потрібно знати, як розмістити дані [77]. Сховища даних недешеві. До цих пір потрібні багатомільйонні загальні витрати і тривалий час для їх проектування та реалізації. Велике корпоративне сховище даних може потребувати 2–3 мільйонів доларів, необхідних для створення програмного та апаратного забезпечення, оплати праці розробників, витрат на навчання та 2–3 років для його створення. Оскільки сховища даних розробляються з метою забезпечення кожного менеджера підприємства загальним рядом даних, то вони, як правило, великі за обсягом даних і збільшуються з часом. Типові розміри потрібної пам'яті – від 50 гігабайт (гігабайт = 1 073 741 844 байтам) до понад один Тб (терабайт = 1000 гігабайтам). Проте, незважаючи на значні фінансові та витрати часу, організація сховищ даних стає все популярнішою. Нині побудова сховищ даних є головним напрямом розвитку інформатики. Оцінки дещо різняться, але можна впевнено стверджувати, що більше половини корпорацій, які входять до 500 найуспішніших, мають або планують мати сховища даних.

Зі сховищем даних пов'язаний термін «**вітрина даних**» (**Data Mart**) [78]. Інколи застосовують такі його синоніми: кіоск даних, підмножина сховища даних, сховище даних рівня підприємства, ярмарка даних. *Вітрина даних* – це певна частина сфокусованих на окрему тему даних або виокремлені елементи сховища даних. Наприклад, деякі компанії скоріше будують вітрину даних щодо споживачів, ніж багатопредметне сховище даних. Така вітрина даних містить всю необхідну інформацію про споживачів. Багато організацій і бізнесових структур починають будувати свої повномасштабні (корпоративні) сховища даних шляхом побудови серії сфокусованих вітрин даних.

Коли сховище даних уже створене та оптимізоване, то необхідно ефективно завантажувати нові дані в систему без переривання процесу підтримки прийняття рішень. Однак у разі збільшення кількості даних розробникам необхідно визначати нові синтаксичні формати та формати запитів, які були б швидшими та легшими. Найпоширенішим засобом організації сховищ даних для задоволення різних аналітичних запитів є використання багатовимірної моделі даних, що пов'язується з поняттям OLAP, зокрема, з його реляційним різновидом.

Група продавців технології OLAP, яка є асоціацією захисників програмних продуктів OLAP, що має призначення сприяти більшому розумінню системи і її головних принципів, сформувала Раду OLAP. Рада запропонувала таке визначення OLAP:

Оперативне (онлайнове) аналітичне оброблення (On-line analytical processing – OLAP) є категорією технології програмного забезпечення, яке дало змогу аналітикам, менеджерам і виконавцям підсилити подання даних завдяки швидкому, узгодженому, інтерактивному доступу до широкого діапазону можливих зображень інформації, яка була одержана шляхом перетворення неопрацьованих (первинних) даних для відображення в реальній вимірності, зрозумілій користувачам, стану підприємства.

З практичного погляду OLAP є якраз тим, чого очікували від СППР протягом багатьох років, тобто перспективною системою, яка проста для використання, містить спеціалізовані (спеціально виділені) дані і пристосована до потреб користувачів. Ця система використовує сховища даних, а також містить велику кількість інструментальних засобів кінцевого користувача для організації доступу до даних і проведення їх аналізу.

OLAP здійснюється в багатокористувацькому клієнт/серверному режимі і уможливило узгоджену швидку відповідь на запити, незалежно від обсягу і складності бази даних. OLAP допомагає користувачеві синтезувати інформацію підприємства завдяки порівняльному, конкретизованому перегляду даних, а також завдяки аналізу фактичних і розрахункових показників у варіантах аналізу типу «що ..., якщо...?». Все це досягається за допомогою використання сервера OLAP.

Сервер OLAP є високорозрядним, багатокористувацьким механізмом (або процесором бази даних) маніпулювання даними, специфічно розробленим для того, щоб підтримувати і здійснювати операції з багатовимірними структурами даних. Багатовимірна структура упорядкована так, щоб кожний елемент даних був розміщений і забезпечений доступом на основі перетину компонентів вимірностей, які визначають той елемент. Сервер і структура даних оптимізовані для швидкого пошуку інформації, для проведення аналізу типу «на даний випадок» (ad-hoc) у будь-якому аспекті, а також для швидкого та гнучкого обчислення і перетворення первинних даних, що ґрунтується на формульних взаємозалежностях.

Досягнутий на даний час стан технологічних засобів і вимоги кінцевих користувачів щодо узгоджених і швидких відповідей визначають, що найкращим підходом до організації оброблення даних є установа багатовимірної бази даних у сервері OLAP. Прикладами багатовимірних серверів можуть бути Light ship Server від Pilot Software і Ess base від Arbor Software.

Функціональні можливості OLAP характеризують підтримку кінцевих користувачів-аналітиків динамічним багатовимірним аналізом консолідованих підприємницьких даних і навігаційними діями, включаючи: обчислення і моделювання, що застосовується за допомогою вимірності, ієрархії і/або компонентів; аналіз трендів за послідовні періоди; квантування підмножин (підмножини даних аналізують за допомогою тонких зрізів) для візуалізації на екрані; практичне оброблення з підвищенням рівня деталізації до найгли-

бших рівнів консолідації основних даних; прямий доступ до основних деталізованих даних; повернення до порівнянь у нових вимірах із застосуванням візуалізації.

5.2.4. Проектування сховищ даних

Підходи до проектування сховищ даних

Як уже зазначалося, хоча сховище даних і є окремим типом бази даних, проте підходи до його проектування мають свої особливості [79]. Можна виділити такі три підходи до проектування сховищ даних: метод реконструкції; проектування за шаблоном та проектування за замовленням.

Метод реконструкції – це проектування сховища даних (СД) на основі використання наявної моделі даних OLTP-системи. Цей підхід полягає в реконструкції наявної моделі бази даних у модель сховища даних. Проектування через реконструкцію рекомендується в тому разі, коли модель OLTP-системи відносно нова і охоплює всі предметні галузі, для яких планується розроблення СД, і коли багато таблиць з фактичними даними в різних вимірах містяться в моделі БД.

Проектування за шаблоном – це такий підхід до розроблення сховища даних, коли за основу для проектування береться функціонуюча модель сховища даних в аналогічній предметній галузі. В цю модель вносяться зміни, що відображають специфіку конкретного об'єкта управління. Кращим варіантом цього підходу є придбання стандартної галузевої моделі побудови СД, якщо така є на підприємствах галузі.

Проектування за замовленням – це проектування з «чистого листка». Цей підхід ігнорує врахування наявних галузевих моделей СД та моделей OLTP-систем. У цьому варіанті проектування розробники цілком зосереджують свою увагу на потребах користувачів у результатах бізнес-аналізу та особливостях предметної галузі, що досліджується.

Основні завдання та способи проектування СД

Як відомо, всі дані, які зберігаються в сховищах даних, можна поділити на такі три основні групи: метадані, детальні дані та агреговані дані. Тому основними завданнями проектування СД є вибір та описування структури детальних даних; вибір параметрів і ступеня узагальнення даних та опис структури агрегованих даних; описування регламенту і процедур завантаження, трансформування, контролю й очистки даних.

Застосовуються два способи проектування сховищ даних: *низхідний* і *висхідний*. Низхідний – це такий підхід, коли спочатку проектується корпоративне сховище, а потім воно стає джерелом інформації для вітрин даних, тобто вітрини є залежними. Висхідний підхід полягає в тому, що спочатку проектуються вітрини даних, які охоплюють окремі напрями діяльності корпорації, чи певні його підрозділи.

За умови узгодженого форматування даних можна на основі семантичного об'єднання вітрин одна з одною створювати розподілені корпоративні сховища даних. Такий поетапний підхід забезпечує можливість у дуже стислі терміни отримати відчутні вигоди, оскільки для формування корпоративного сховища потрібно кілька років, а окремі вітрини можна створити за кілька місяців.

Проектування вітрин даних має свою особливість і специфіку, що робить цей процес відмінним від проектування баз даних транзакційних систем. Вітрини даних не схожі на бази даних оперативного оброблення транзакцій – OLTP-систем, які обслуговують поточну діяльність підприємств та організацій. Вітрини даних використовуються для підтримки прийняття рішень, що дає змогу аналітикам виявляти тенденції, проводити порівняння і прогнозувати майбутні результати.

Джерелом наповнення сховищ та вітрин даних може бути як первинна оперативна інформація, що зберігається в БД, так і результати розв'язання задач OLTP. Для даних, які не можуть бути отримані з OLTP-систем, визначають зовнішні джерела інформації. При цьому по кожному джерелу інформації дається повна характеристика, тобто визначаються формати даних, що надходять до СД, періодичність та форми подання, системи кодування, алгоритми оброблення, інакше кажучи, аналізуються метадані, тобто інформація про ті дані, що підлягають збереженню в СД.

Визначення фактів і вимірів

Для проектування сховищ і вітрин даних потрібно визначити таблицю фактів і перелік вимірів, установити, факти якого типу найефективніші і чим їх відрізнити від вимірів. *Факти* – це елементи, які можуть бути виміряні й проаналізовані. Кращими фактами вважаються числові, адитивні дані, що можуть бути подані послідовною низкою значень.

Числові факти – це реквізити основи, які відображають кількісні величини. Над числовими фактами допускається виконання різних математичних операцій; їх легко вимірювати. Характерною ознакою таких фактів є те, що вони часто змінюються. Змінність конкретних значень фактів є дуже важливою їх ознакою з погляду аналізу, оскільки досліджувати величини, що не змінюються в часі, немає сенсу.

Адитивний факт може підсумовуватися за всіма вимірами. Наприклад, кількість проданих виробів є адитивним фактом, оскільки шляхом підсумовування можна визначити, скільки виробів продано на минулому тижні (за часом); скільки виробів продано у регіонах; скільки продано кожному споживачу. В сховищах даних надається перевага адитивним фактам, тому що, підсумовуючи їх, можна отримати компактні ряди результатів.

Деякі факти є *напів-адитивними* (semiadditive). Їх можна коректно підсумовувати за певними вимірами, але не за всіма. Наприклад, доцільно підсумовувати складські запаси за видами продуктів або за регіонами, але не за

часом. Напів-адитивні факти не має сенсу підсумовувати за окремими вимірами, але їх можна оцінювати інакше. Наприклад, має значення середня величина складських запасів у часі, а також максимальний і мінімальний рівні запасів та деякі інші статистичні показники.

І нарешті, мають місце *неадитивні* (nonadditive) факти. Ці факти неможливо підсумовувати за жодним виміром. Прикладом числових неадитивних фактів є табельні номери співробітників. Неадитивні факти не можна підсумовувати, але їх можна полічити, тому вони вимірні. Неадитивні факти частіше групуються з вимірами. *Виміри* – це описові, якісні атрибути, що служать умовами вибирання даних у запитах, або є заголовками рядів у звітах. До таблиць вимірів належать таблиці, що містять умовно-постійну інформацію. Тобто дані, що характеризують виміри, як правило, текстові, відносно статичні і неадитивні.

Не завжди легко можна відрізнити факти від вимірів. Іноді в таблиці вимірів можуть міститися числові, адитивні атрибути, такі як ціна одиниці товару. А іноді статичний і неадитивний атрибут доцільно розмістити в таблиці фактів. Все залежить від призначення конкретної вітрини. Наприклад, стать співробітника – це текстовий, статичний і неадитивний елемент. Але якщо проектується вітрина, присвячена аналізу трудових ресурсів, і аналітики часто задають запитання типу: «яке співвідношення між кількістю співробітників чоловічої і жіночої статі, що займають керівні посади?», то ознаку статі краще розташувати в таблиці фактів.

Визначення ступеня деталізації даних є наступним кроком проектування. Найдрібніші неподільні елементи – нижній рівень деталізації даних, що зберігаються у вітрині. Дані можуть підсумовуватися на різних рівнях відповідно до вибраної ієрархії вимірів. Щоденні обсяги продажу можна підсумовувати за тиждень, місяць або рік; продані товари можна підсумовувати за кожним товаром, зокрема, за їх видами чи товарними групами. Іноді застосовують кілька ієрархічних систем. Магазины можна групувати за поштовими регіонами з однаковими тарифами на пересилання або за територіями з аналогічною структурою продажу для проведення маркетингових досліджень. Користувачі вітрин можуть працювати з різними рівнями даних, що підсумовуються. Питання полягає в тому, яка ступінь деталізації потрібна користувачу? Завжди є вірогідність того, що якийсь користувач у своєму пошуку побажає дійти до базової транзакції, і на основі цього можна дійти висновку, що за проектування вітрини необхідно прагнути до забезпечення найнижчого рівня деталізування. Але, зазвичай, у цьому немає необхідності, враховуючи те, що вітрини призначені для проведення бізнес-аналізу, метою якого є виявлення тенденції, а не вивчення конкретних детальних фактів. Проте ступінь деталізації фактів, що зберігаються у вітрині, передусім, визначається метою її створення. Так, наприклад, керівництву роздрібного магазину, бажаному відстежити зміни складських запасів, немає потреби зберігати у вітрині дані

про кожну торговельну операцію; обсяги продажу можна підсумовувати на рівні товарів. Але якщо необхідно проаналізувати поведінку споживачів, то буде потрібна інформація про окремі торговельні операції для їх аналізу і вивчення уподобань споживачів.

Визначення ступеня деталізації та агрегації даних – це дуже важливий момент проектування, тому що він впливає на розмір фізичної моделі. Іноді вибір ступеня деталізації визначається інтенсивністю змін: для відстеження заробітної плати службовців протягом тривалого періоду не потрібно мати її величини за день або тиждень, якщо, наприклад, перегляд оплат праці співробітників виконується раз на рік.

Установивши ступінь деталізації, можна визначити атрибути таблиці вимірювань. У вітрині фактичних даних одним із вимірів є час, і в таблицю вимірів необхідно помістити інформацію про одиниці вимірювання часу, відповідні для таблиць фактів даної вітрини. Якщо для аналізу потрібні підсумкові значення за місяць або рік, то тоді для організації таблиці вимірювань будуть потрібні лише такі одиниці часу, як рік і місяць. Фінансові вітрини бажано доповнити таким виміром як квартал.

Контрольні питання до підрозділу 5.2

- 1. У чому полягає різниця створення сховищ даних від створення звичайних БД?*
- 2. Які Вам відомі обов'язкові етапи створення сховищ даних?*
- 3. Які основні елементи містить узагальнена архітектура сховища даних?*
- 4. Для яких основних цілей використовуються метадані?*
- 5. З чого формуються куби і/або гіперкуби фактичних поточних значень?*
- 6. Які вісім головних властивостей сховищ даних вказані Б. Інмоном?*
- 7. Який сенс має поняття «вітрина даних» стосовно поняття «сховище даних»?*
- 8. Якими позитивними рисами відрізняється OLAP (онлайнове аналітичне оброблення) від попередніх СППР?*
- 9. Яка особливість відрізняє сервер OLAP ?*
- 10. Які існують три підходи до проектування сховищ даних?*
- 11. У чому полягають особливості низхідного та висхідного способів проектування сховищ даних?*
- 12. У чому полягає різниця між адитивними, напів-адитивними та неадитивними фактами., які необхідно визначати при проектуванні сховищ та вітрин даних?*

5.3. Система аналітичного інтерактивного оброблення (OLAP)

5.3.1. Зародження і розвиток OLAP-систем

OLAP (аббревіатура від **On-line Analytical Processing** – інтерактивне аналітичне оброблення) фактично означає не окремі конкретні програмні продукти, а технологію багатовимірного аналізу даних [72], основу якої започаткувала опублікована 1993 року праця Е. Ф. Кодда (E. F. Codd) [80], «OLAP для користувачів-аналітиків: яким воно має бути», у якій він запропонував 12 правил, які виражали концепцію оперативного аналітичного оброблення даних і фактично послужили стандартом інструментальних засобів оперативного аналітичного оброблення (табл. 5.5.)

Таблиця 5.5

Правила Кодда для OLAP

<p>1. Багатовимірне концептуальне зображення (Multidimensional conceptual view). Уможливорює користувачу перегляд даних, які можна аналізувати за регіонами, часом тощо</p> <p>2. Прозорість (Transparency). Робить базову (що є основою) аналітичну здатність цілком прозорою для користувача.</p> <p>3. Доступність (Accessibility). Дає змогу створювати власну логічну схему для запам'ятовування неоднорідних фізичних даних.</p> <p>4. Узгоджена продуктивність щодо створення повідомлень (Consistent reporting performance). Забезпечує надійну продуктивність підготовки звітів для будь-якої кількості вимірів, яку обирає користувач.</p> <p>5. Архітектура клієнт/сервер (Client/server architecture). Забезпечує мінімум зусиль для використання цієї архітектури.</p> <p>6. Генерування вимірів (Generic dimensionality). Має тільки одну логічну структуру для подання всіх вимірів.</p> <p>7. Динамічне оброблення розріджених матриць (Dynamic sparse matrix handling). Ефективно поводитьься з пустими або порожніми пропусками в матриці, що роблять матрицю розрідженою.</p> <p>8. Багатокористувацька підтримка (Multiuser support). Уможливорює одночасний доступ, захист і цілісність для багатьох користувачів.</p> <p>9. Необмежені перехресно-вимірні операції (Unrestricted cross-dimensional operations). Виконує обчислення і інші операції над вимірами без втручання користувача.</p> <p>10. Інтуїтивне маніпулювання даними (Intuitive data manipulation). Уможливорює оброблення з деталізацією (drilling-down), наближення/віддалення об'єкта, переорієнтацію і консолідоване подання даних і аналізів.</p> <p>11. Гнучка організація створення звітів (Flexible reporting). Дає змогу користувачам легко і ефективно маніпулювати звітами даних.</p> <p>12. Необмеженість вимірів і рівнів агрегації (Unlimited dimensions and aggregation levels). Містить щонайменше 15, а то і 20 вимірів даних.</p>

1995 року до них було додано ще кілька правил, що у своїй сукупності визначили основні сучасні вимоги до OLAP-систем. Ці правила опісля були поділені на чотири групи.

1. **Базові характеристики:** багатовимірність моделі даних; інтуїтивні механізми маніпулювання даними; доступність; пакетне отримання даних; клієнт-серверна архітектура; прозорість (для користувача); багатокористувачька робота.

2. **Спеціальні характеристики:** оброблення неформалізованих даних; зберігання результатів окремо від вхідних даних; виокремлення даних, яких бракує (тобто вони мусять відрізнятися від нульового значення); оброблення значень, яких бракує (всі значення, яких бракує, мають бути проігноровані в разі аналізу).

3. **Характеристики побудови звітів:** гнучкі можливості одержання звітів; стабільна продуктивність за підготовки звітів; автоматичне регулювання фізичного рівня.

4. **Керування розмірністю:** загальна функціональність; необмежена кількість вимірів і рівнів агрегування; необмежена кількість операцій над даними різних вимірів.

Ці концепції покладені в основу технології OLAP, хоча реально наявні OLAP-системи мають далеко не повний перелік описаних характеристик. OLAP-технологія, яку можна назвати також *інтерактивним (діалоговим) аналітичним обробленням*, дає змогу на основі багатовимірної (гіперкубічної) моделі даних (на відміну від плоскої реляційної моделі даних) моделювати реальні структури і зв'язки, що є виключно важливими для аналітичних систем. Вона призначена для створення багатопараметричних моделей з метою адекватнішого відображення реальних бізнес-процесів. Технологія OLAP дає змогу швидко змінювати погляди на дані залежно від вибраних параметрів і забезпечувати особу, що приймає рішення, повною картиною щодо ситуацій, які аналізуються.

Усі OLAP-системи побудовані на двох базових принципах:

1) дані, необхідні для прийняття рішень, слід попередньо агрегувати на всіх відповідних рівнях і організувати так, щоб забезпечити максимально швидкий доступ до них;

2) мова маніпулювання даними основана на бізнес-поняттях.

Дані параметруються кількома рівноправними вимірами, наприклад, дані стосовно продажу у великій торговельній компанії можна аналізувати в таких вимірах:

- «час» (день, тиждень, місяць, квартал, рік),
- «географія» (місто, область, країна),
- «товар» (фірма-виробник, вид товару),
- «покупець» (стать, вік).

Засобами користувачького інтерфейсу OLAP-системи можна здійснювати такі базові операції над гіперкубом моделі даних [81]:

1. *поворот*, щоб змінювати вимірну орієнтацію звіту або сторінку показу на екрані;
2. *проекція* (при проекції значення в комірках, що лежать на осі проекції, підсумовуються відповідно до визначеної ознаки);
3. *розкриття* (drill-down), тобто коли одне із значень виміру замінюється сукупністю значень із наступного рівня ієрархії виміру, в такому разі замінюються значення в комірках гіперкуба;
4. *згортка* (roll-up/drill-up) є операцією, протилежною до операції розкриття;
5. *перетин* (slice-and-dice) – ініційований користувачем процес навігації за допомогою інтерактивного показу сторінки у вигляді зрізу, використовуючи операції повороту і розкриття.

Незважаючи на те, що екран комп'ютера плоский, користувач має змогу спостерігати гіперкуб моделі даних за допомогою відповідних плоских зрізів, використовуючи вищезазначені операції.

Оскільки в основі OLAP-технології лежить концепція гіперкуба моделі даних, то залежно від відповіді на запитання, чи існує гіперкуб як окрема фізична структура? чи це є лише віртуальна модель даних? розрізняють два основних типи аналітичного оброблення даних: MOLAP і ROLAP.

MOLAP (Multidimensional OLAP) – це багатовимірна OLAP-система, в котрій гіперкуб реалізується як окрема база даних нереляційної структури, яка забезпечує багатовимірне зберігання, оброблення і подання даних. Програмні продукти, що належать до цього типу OLAP-технології, як правило, мають сервер багатовимірних баз даних. Ця структура забезпечує максимально ефективний щодо швидкості доступ до даних, проте потребує додаткового ресурсу пам'яті. Крім того, велика розмірність моделі даних і розрідженість гіперкубів приводить до необхідності витрат великих апаратних ресурсів, що не завжди може бути доцільним. Є чотири провідні продавці серверів MOLAP: Arbor Software (Essbase), Kenan Technologies (Acumate Enterprise), Oracle/IRI (Express), D& B/Pilot Software (Light ship). Крім того, недавно були розроблені нова версія сервера Cognos Power Play і новий мультिवимірний сервер інституту SAS.

У ROLAP (Relational OLAP) багатовимірна структура реалізується реляційними таблицями, тобто гіперкуб – це лише користувацький інтерфейс, котрий емулюється на звичайній реляційній СКБД. Така структура забезпечує зберігання великих обсягів інформації, проте є менш продуктивною з погляду ефективності OLAP-операцій. Незважаючи на те, що продавці ROLAP увійшли в ринок програмних продуктів тільки протягом останніх кількох років, чотири з них стали досить відомими. Два з них знаходяться серед найшвидше зростаючих корпорацій у Сполучених Штатах. Цими чотирма компаніями є: Microstrategy, Information Advantage, Stanford Technology Group (недавно придбаний Informix), IQ Software. Нині відкриті сайти

ROLAP, що містять аналітичні механізми сервера, створення звітів і інструментальні засоби аналізу, інструментальні засоби проектування системи та програмне забезпечення.

Певні недоліки, характерні основним типам OLAP-технологій, привели до появи нового типу аналітичних інструментів – HOLAP-систем, що являють собою гібридне (hybrid) оперативне аналітичне оброблення даних, де реалізуються обидва підходи, тобто доступ може надаватися як до багатовимірних баз даних, так і до даних реляційного типу [81].

На даний час розроблено досить багато систем підтримки прийняття рішень, сконструйованих з використанням OLAP-технологій (Hyperion OLAP, Elite OLAP, Oracle Express та багато інших). Ринок програмних OLAP-продуктів постійно розширюється. Сучасні системи оперативного аналітичного оброблення надають користувачам можливості розв'язувати ключові завдання управління бізнес-процесом, зокрема додатки Hyperion OLAP дають змогу користувачам виконувати такі завдання: аналіз прибутковості; аналіз напрямів розвитку продукції; аналіз продажу; аналіз стану на ринку; аналіз асортименту продуктів; аналіз ризику; аналіз конкурентоспроможності; складання звітів з продуктивності; моделювання сценарію; аналіз бюджету і прогнозів тощо. Слід ще раз підкреслити, що згідно із сучасними поглядами на створення інформаційних систем OLAP-системи мають базуватися на спеціальній базі даних – сховищі даних, але можуть використовуватися і автономно.

5.3.2. Інструментальні засоби кінцевих користувачів в OLAP

Є низка інструментальних засобів для кінцевих користувачів, доступних для підтримки OLAP. До них належать: Business Object Inc.'s Business Objects, програмне забезпечення AG'sEsperant, Andyne'sPaBLO, Visualizer IBM's і Platinum's Forest&Trees. У той час, як ці продукти суттєво відрізняються, вони використовуються з однією метою: полегшити для технічно не підготовленого персоналу доступ до даних, їх аналізування і зображення без допомоги фахівців з інформаційних систем (ІС).

Протягом багатьох років фактично вся інформація, яка була згенерована комп'ютерами в організаціях, отримувалась завдяки зусиллям співробітників ІС. При цьому потрібні були роки для реалізації нового запиту щодо отримання інформації, для розроблення очікуваних додатків. Умови різко змінилися у зв'язку з появою персональних комп'ютерів (ПК) і їх програмного забезпечення (наприклад, електронних таблиць, програмного забезпечення машинної графіки тощо). Користувачі отримали засоби, які дали змогу їм самостійно отримувати потрібну інформацію. Однак нерозв'язані проблеми ще залишилися, оскільки деякі особливі дані, найпотрібніші кінцевому користувачу, були розміщені в операційних базах даних і були важкодоступними. Важливим кроком у напрямі до розв'язання цієї проблеми було об'єднання в мережу персональних комп'ютерів і надання можливості доступу користува-

чам до окремих файлів, а віднедавна – до сховищ даних. У той час, як деякі інструментальні засоби кінцевого користувача були доступними для маніпулювання даними, в багатьох випадках користувачі мусили вивчити і використовувати мови запитів SQL.

Для добре підготовлених користувачів це було прийнятним варіантом, проте для більшості інших це було не так, оскільки підхід на основі мови запитів потребував, щоб користувачі розуміли деталі структури баз даних. Кінцеві користувачі часто не мали відповідних знань для самостійного формулювання складних запитів, і, що найважливіше, вони не завжди отримували очікувані результати через те, що некоректно описували запити. Ця ситуація привела до появи генерації інструментальних засобів OLAP.

Найбільшу популярність із них отримав інструментальний засіб підтримки прийняття рішень *Business Objects* [82]. *Business Objects* є ефективним прикладом сучасного програмного забезпечення, яке розроблене для надання можливості кінцевим користувачам забезпечувати власну підтримку прийняття рішень, тобто виконувати свої запити без розуміння деталей структур бази даних. Це робиться за допомогою подання інформації не технічним, а бізнес-орієнтованим способом, який добре знайомий користувачеві. Ці зображення даних називаються «бізнес-об'єктами» («*Business Objects*»). Кожний бізнес-об'єкт є основним визначенням структурованої мови запитів, який відповідає його відображенню в базі даних. Наприклад, бізнес-об'єкт може бути простим, типу «останнє ім'я замовника», що є окремим полем у таблиці і визначається як `customers.last_name`, або комплексним – «дохід від збуту», який фактично обчислюється за рівнянням, а необхідні для цього дані розміщені на багатьох полях у різних таблицях.

Кожний кінцевий користувач має доступ до множини бізнес-об'єктів, які містяться всередині середовища, що називається «всесвітом» (*universe*). *Universe* є логічним відображенням бази даних додатка для специфічної групи користувачів. Наприклад, маркетинговий аналітик міг би працювати зі «всесвітом» з маркетингу, який містить такі бізнес-об'єкти: «клієнт», «дохід від продажу», «продукт» тощо. Працівник відділу кадрів міг би мати доступ до *universe*, який включає бізнес-об'єкти: «ім'я службовця», «заробітна плата» і «дата прийняття на роботу».

Business Objects дає змогу користувачам легко формувати запити за допомогою виділення і клацання на бізнес-об'єктах, які були наперед визначені. Результати запиту *Business Objects* можуть бути відображені або використовуватися кількома способами. Один спосіб – подати інформацію у визначеному користувачем виді звіту даних; інший – показати інформацію у вигляді вибраного користувачем виду діаграми. Результати можна також поміщати в будь-яке програмне забезпечення, яке підтримує протокол Microsoft DDE (динамічного обміну даними). Наприклад, користувач міг би створити вели-

коформатні таблиці Microsoft Excel або Lotus 1–2–3, що базуються на результатах запитів Business Objects.

Сьогодні набір інструментальних засобів для значно поширено [83].

5.3.3. Система оперативного аналітичного оброблення Oracle Express OLAP

Однією з найвідоміших реалізацій ідеї оперативного аналітичного оброблення, що інтенсивно впроваджується в Україні, є сімейство програмних продуктів Oracle Express OLAP, яке являє собою інструментально-технологічне програмне забезпечення, призначене для створення прикладних аналітичних систем підтримки прийняття рішень на основі багатовимірного аналізу даних [84]. Воно забезпечує багатofункціональним інструментарієм як для створення і підтримки багатовимірних баз даних, так і для одержання прикладних програм, що реалізують функції аналізу даних та користувацького інтерфейсу. Oracle Express OLAP побудований за архітектурою клієнт/сервер і містить такі серверні і клієнтські програмні продукти (рис. 5.7):

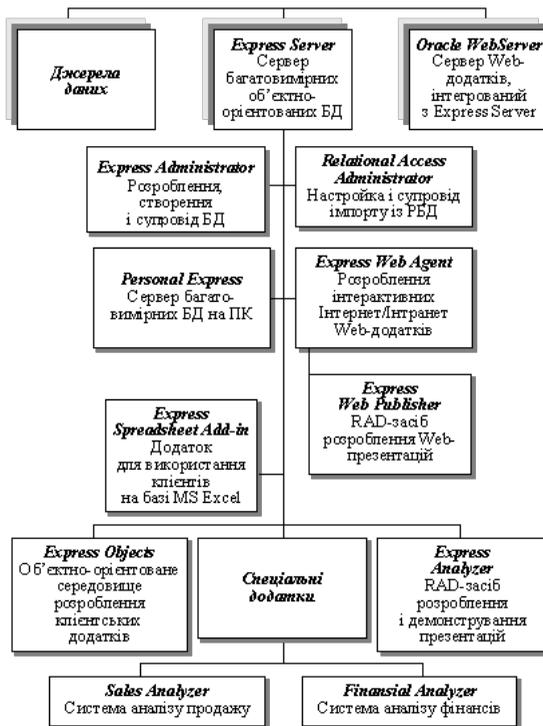


Рис. 5.7. Схема зв'язку програмних продуктів Oracle Express OLAP

Oracle Express Server – сервер керування багатовимірною базою даних (ББД), що забезпечує аналіз, прогнозування і подання звітів про підприємницьку діяльність;

Personal Express – система керування ББД, що базується на персональних комп'ютерах (ПК), яка забезпечує функції Oracle Express Server для ПК;

Oracle Express Objects – професійне інструментальне середовище для візуального об'єктно-орієнтованого розроблення клієнт/ серверних додатків OLAP для MS Windows;

Oracle Express Analyzer – інструментальне середовище кінцевого користувача, що призначене для перегляду й аналізу багатовимірних даних, створення представницьких OLAP-додатків для MS Windows і демонстрування готових додатків, розроблених в Oracle Express Objects та в Oracle Express Analyzer;

Excel Spreadsheet Add-In – додаток, що забезпечує використання клієнтів на базі MS Excel;

Oracle Financial Analyzer – розподілена обчислювальна система OLAP-додатків, яка забезпечує планування, контроль і складання звітів про фінансову діяльність на рівні підприємства і призначена для локальної обчислювальної мережі ПК;

Oracle Sales Analyzer – інструментальний OLAP-засіб аналізу корпоративних даних в інтересах відділів продажу і маркетингу, що дає змогу на основі корпоративних даних оцінювати тенденції розвитку в різних сферах підприємницької діяльності;

Oracle Express Web Publisher і *Oracle Express WebAgent* – доповнення до Express Server, які уможливають створення додатків Express OLAP, що виконуються з використанням будь-якого Web-броузера;

Oracle Express Administrator – інструмент проектування, створення та супроводження баз даних.

Oracle Express Server. Аналітична потужність і гнучкість засобів розроблення OLAP-додатків базується на Oracle Express Server – багатофункціональному інструменті підтримки прийняття рішень та системі керування ББД. Він використовує багатовимірну модель даних, що найефективніше відображає уявлення користувачів про предметну галузь. У рамках цієї моделі визначаються виміри аналізу, від яких залежать показники. Виміри мають ієрархічну структуру і тому користувачі можуть використовувати як детальні, так і агреговані дані. Завдяки можливості моделювати взаємозв'язки між даними, що зберігаються, Oracle Express Server дає змогу в широких межах описувати підприємницьку діяльність будь-якої компанії.

Personal Express. Є версія Express Server, функціонуюча на персональному комп'ютері під керуванням Windows 95/98/NT, – Personal Express. Щодо архітектури, моделі даних і аналітичних можливостей ці продукти абсолютно ідентичні. Personal Express використовується передусім мобільними користу-

вачами, що проводять аналіз даних, будучи від'єднаними від корпоративної мережі. Фактично Personal Express – це функціональний аналог Express Server для IBM-сумісного ПК.

Система забезпечує гнучку взаємодію на рівні процесів з Express Server, що функціонує на інших платформах, має доступ через Express Server до всіх реляційних БД, з якими сервер взаємодіє. Вона підтримує ту ж систему команд мови Express, яку підтримує і Express Server, що без ускладнень забезпечує використання розробленого користувачем програмного забезпечення з Personal Express в Express Server і навпаки. Personal Express надає можливість розробляти власне меню Windows, що дає змогу користувачеві ефективніше виконувати команди і використовувати додатки. Ця система підтримує клієнт/серверну взаємодію й динамічний обмін даними (DDE) з іншими додатками MS Windows, написаними мовами C і Visual Basic.

Oracle Express Objects

Oracle Express Objects – це інструментальне графічне середовище для об'єктно-орієнтованого розроблення клієнт/серверних додатків OLAP під MS Windows професіоналами, яке належить до клієнтської частини програмного забезпечення Oracle Express. Використовуючи Express Objects, можна створювати і налаштовувати клієнт/серверний додаток, що дає змогу кінцевому користувачеві ефективно застосовувати всю потужність Express OLAP-продуктів (рис. 5.8). Через Express Objects можливий доступ до будь-яких об'єктів, які зберігаються в БД під керуванням Personal Express або Express Server, а також до реляційних БД, доступних за їхньою допомогою.

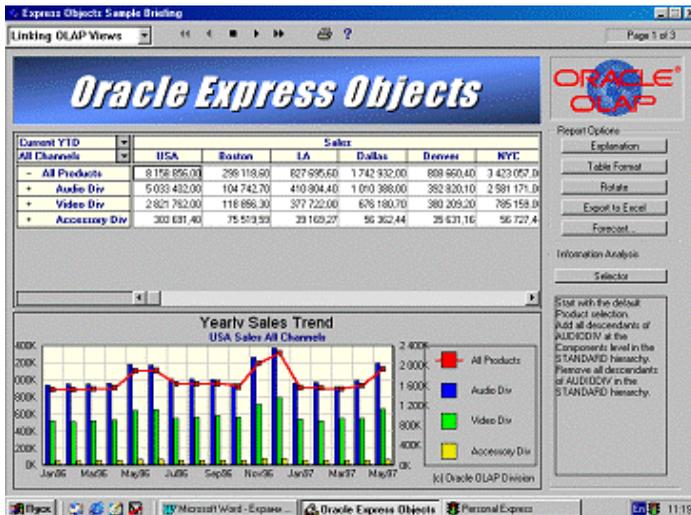


Рис. 5.8. Сторінка прикладної програми, створеної в Express Objects

Для розроблення програм використовують *Express Basic*, який має загальні з *Visual Basic* основне ядро, функції і вирази. Можливим є динамічний обмін даними (*DDE*) і вбудовування *OLE*, а також *Visual Basic*-об'єктів. Ефективність візуального розроблення прикладних застосувань забезпечує набір інструментів: *Object Browser*, *Database Browser* та *Object Inspector*. Це середовище підтримує всі можливості MS Windows і велику кількість удосконалень, що містить Windows 95.

Oracle Express Analyzer

Oracle Express Analyzer – засіб кінцевого користувача, призначений для самостійного формування звітів, аналізу багатовимірних даних Express і передавання результатів на Web. Інтерфейс Express Analyzer настільки простий і інтуїтивно зрозумілий, що навіть невідготовлений користувач негайно отримує доступ до можливостей OLAP. Розроблення презентацій полягає у конструюванні їх з готових компонентів (сторінок, таблиць, графіків, OLE-об'єктів, баннерів, кнопок та об'єктів Express Output, призначених для виконання на сервері команд Express).

Oracle Express Analyzer дає змогу: аналізувати дані – створювати таблиці й графіки, що відображають дані з БД, і маніпулювати їх поданням для вивчення даних у різних аспектах; створювати й редагувати представницькі додатки, які можуть містити кілька сторінок з таблицями і графіками, що відображають інформацію, яка зберігається в БД, у різних вимірах; переглядати додатки, підготовлені в Express Analyzer і Express Objects іншими користувачами.

Express Spreadsheet Add-In

Крім Express Analyzer або Express Objects, користувачі можуть отримувати динамічний доступ до багатовимірних даних, використовуючи як інтерфейс Microsoft Excel 95 чи 97. Спеціальна компонента Express Spreadsheet Add-In доповнює стандартні можливості цього процесора електронних таблиць функціями OLAP, даючи змогу за допомогою простого «Майстра» інтерактивно будувати в середовищі Excel багатовимірні звіти. Користувачам доступні ті ж основні маніпуляції даними, що і в інших інструментах OLAP, – отримання різних зрізів, деталізація і «Селектор». Для отримання графічного подання даних застосовуються відповідні можливості Excel.

Spreadsheet Add-In пропонує власний інтерфейс прикладного програмування (API) для розроблення складніших додатків на базі стандартних електронних таблиць. Користуючись ним, розробники можуть створювати макроси для автоматизованого виконання багатьох операцій, закладених у SpreadsheetAdd-In.

Oracle Financial Analyzer

Oracle Financial Analyzer – розподілена обчислювальна система OLAP-додатків, інтегрований засіб для формування фінансової звітності, аналізу, прогнозування, складання і контролю за виконанням бюджетів у масштабах

підприємства, яка призначена для використання у локальній обчислювальній мережі ПК. Через неї можливий доступ до будь-яких даних, які зберігаються в БД під керуванням Personal Express або Express Server, а також до реляційних БД, доступних за їх допомогою. Об'єднуючи розподілену архітектуру з централізованим керуванням потоками даних і аналітичною потужністю Express, додаток уможливує здійснення контролю витрат, аналіз ефективності й оцінювання потенціалу на всіх рівнях організації в рамках єдиної системи.

Financial Analyzer дає змогу розв'язувати такі завдання: складати й аналізувати корпоративні фінансові звіти і формувати бюджет; здійснювати фінансове моделювання; конфігурувати фінансову систему, виходячи з конкретних потреб підприємницької діяльності; формувати ієрархічну систему аналізу, супроводу і поширення фінансових даних у масштабах корпорації.

Oracle Sales Analyzer

Oracle Sales Analyzer – інструментальний OLAP-засіб аналізу корпоративних даних, додаток, призначений для застосування на рівні підприємства з метою аналізу продажу і маркетингових досліджень. Він уможливує доступ до будь-яких даних, які зберігаються в БД під керуванням Personal Express або Express Server, а також до реляційних БД, доступних через них. За допомогою цього пакета можуть бути оцінені тенденції і прогноз продажу, ефективність рекламних кампаній, прибутковість продукту або замовника, життєвий цикл продукту, однак цим його функціональні можливості не обмежуються. Sales Analyzer може використовуватися для розв'язування загальних аналітичних завдань, не пов'язаних з комерційною діяльністю, для яких потрібні можливості OLAP.

Oracle Express Web Agent і Oracle Express Web Publisher

Технологія Express дає змогу користувачам проводити інтерактивний аналіз не тільки в середовищі клієнт-сервера, але і в архітектурі Web. Це уможливує просте поширення аналітичної інформації в рамках організації і за її межами, дешевший супровід OLAP-додатків, зниження апаратних вимог до робочих станцій і незалежність додатка від клієнтської платформи, оскільки для доступу до даних використовується звичайний браузер.

Oracle Express Web Agent – компонента Express Server, яка дає змогу створювати додатки Express OLAP, що використовуються в мережах Інтра-нет/Інтернет з будь-яким Web-браузером. Oracle Express WebAgent, що складається з інтерфейсного модуля до Express Server і ряду Java-апплетів, які подають дані в графічному вигляді, для візуалізації і опрацювання даних, надає будь-якому авторизованому співробітнику організації всі можливості OLAP, не потребуючи жодного програмного забезпечення Express на його персональному комп'ютері. Динамічний доступ до даних гарантує користувачам отримання документів, що завжди містять найактуальнішу інформацію.

Web-додатки в середовищі Express можна створювати кількома способами. Найпростіший з них, доступний навіть непідготовленим користувачам,

полягає в стандартній опції експорту на Web будь-якої таблиці або діаграми в Express Analyzer/Objects. Досвідчені користувачі і розробники можуть створювати Web-брифінги за допомогою графічного середовища Express Web Publisher. Oracle Express Web Publisher – RAD (Rapid Application development) є засобом швидкого розроблення OLAP презентацій для Web.

Використання Web-технологій в Oracle Express – найпривабливіший засіб для розроблення СППР. Використовуючи технології Інтранет/Інтернет, можна забезпечити доступ до СППР звідусіль, де є з'єднання з Інтернетом, у тому числі з офісу клієнта чи із залу проведення торгів або переговорів. Застосування клієнтом простого Web-браузера зробить систему підтримки прийняття рішень комфортною навіть для самого невідготовленого в інформаційних технологіях менеджера. Застосування Web-технологій також забезпечує просте розв'язання проблеми інтегрування різних платформ в інформаційних системах. У цей час практично для всіх платформ Web-браузери розповсюджуються умовно безкоштовно.

Oracle Express Administrator

Основним інструментом побудови і адміністрування багатовимірних вітрин даних є Oracle Express Administrator. Це візуальне середовище розробника, що істотно полегшує створення, модифікацію і керування об'єктами БД Express, а також завантаження даних з різних джерел.

За допомогою графічного інтерфейсу, до складу якого входить низка «Майстрів», можна визначати й редагувати будь-які об'єкти багатовимірної БД, генерувати оптимізований програмний код для завантаження даних і їх агрегування, редагувати ієрархії у вимірах з використанням техніки «drag-and-drop» (перетягнути і залишити). Вбудований редактор Express SPL дає змогу створювати власні процедури, що зберігаються, і модифікувати код, згенерований «Майстрами». Після компіляції ці процедури виконуються безпосередньо у вікні команд Express або вносяться в розклад за допомогою планувальника завдань.

Адміністратор може визначити конкретну дату і час запуску, періодичність виконання і залежності між завданнями (послідовність виклику процедур). Усе це дає змогу повністю автоматизувати процеси періодичного завантаження даних у багатовимірну вітрину, а також агрегувати їх, у той же час надаючи адміністратору можливість відстежувати важливість і протокол виконання завдань за допомогою спеціального монітора.

На закінчення зазначимо, що OLAP, зокрема Oracle Express, є відносно новою технологією, що постійно нарощується і модифікується. Проте ця технологія має стабільні особливості: доступ до дуже великих обсягів інформації, накопичених протягом кількох років; аналіз взаємозв'язків між безліччю аспектів підприємницької діяльності, наприклад між продажем, виробництвом, підрозділами, регіонами, різними продуктами; деталізація, узагальнення

і порівняння ієрархічних даних, наприклад прибутки по містах, регіонах, країнах, континентах або за день, місяць, квартал чи рік; подання даних з різних поглядів, наприклад прибутків по регіонах або по різних продуктах у межах одного регіону; складні обчислювальні операції над даними, наприклад розрахунок очікуваних прибутків у вигляді функції щорічного продажу по кожному виду продукції в окремо взятому регіоні; можливість оброблення запитів у «реальному часі» – в темпі процесу аналітичного осмислення даних користувачем.

До традиційних додатків OLAP-систем прийнято відносити: проведення аналізу ринків збуту, управління продажем, аналіз фінансів. Багатовимірні бази даних Oracle Express можуть знайти застосування скрізь, де потрібна консолідація з розрізнених джерел і аналіз значних обсягів емпіричних даних, визначених у просторі з великою кількістю вимірів.

Контрольні питання до підрозділу 5.3

1. Скільки всього правил для інтерактивного аналітичного оброблення даних було запропоновано Е. Ф. Коддом?

2. На яких двох базових принципах побудовані усі OLAP-системи?

3. В основі OLAP-технології лежить концепція гіперкуба моделі даних, у зв'язку з цим які два основних типи аналітичного оброблення даних розрізняють?

4. Яку назву має багатовимірна OLAP-система, в якій гіперкуб реалізується як окрема база даних нереляційної структури, яка забезпечує багатовимірне зберігання, оброблення і подання даних?

5. Яку назву має багатовимірна OLAP-система, в якій багатовимірна структура реалізується реляційними таблицями, тобто гіперкуб – це лише користувацький інтерфейс, який емулюється на звичайній реляційній СКБД?

6. Яку назву має оперативне аналітичне оброблення даних, де реалізуються обидва підходи: Multidimensional OLAP та Relational OLAP?

7. З якою важливою метою створювалися сучасні інструментальні засоби для підтримки OLAP, такі як Business Objects?

8. Сімейство яких програмних продуктів щодо оперативного аналітичного оброблення найбільш інтенсивно впроваджується в Україні?

9. Які особливості відрізняють версію Personal Express від основної версії Oracle Express Server?

10. Яку назву має інструментальне графічне середовище для об'єктно-орієнтованого розроблення клієнт/серверних додатків OLAP під MS Windows професіоналами, яке належить до клієнтської частини програмного забезпечення Oracle Express?

ЛІТЕРАТУРА

1. Ситник В. Ф. Системи підтримки прийняття рішень: Навч. посіб. К.: КНЕУ, 2009. 614 с. URL : https://dut.edu.ua/uploads/l_2208_74641698.pdf (дата звернення: 10.05.2022).
2. Little, J.D.C. Models and Managers. The Concept of a Decision Calculus//Management Science. 2007. v.16. №8.
3. Decision Support Systems: Issues and Challenges/ Ed. By G. Fickand RH Sprague. Oxford: Pergamon Press, 2018. 189 p.
4. Системи підтримки прийняття рішень: навчальний посібник для самостійного вивчення дисципліни / С. М. Братушка, С. М. Новак, С. О. Хайлук; Державний вищий навчальний заклад «Українська академія банківської справи Національного банку України». Суми : ДВНЗ “УАБС НБУ”, 2010. 265 с.
5. С. В. Корнеєв. Системи підтримки прийняття рішень у бізнесі//Мережі & Бизнес.№6, 2005.
6. Bonczek RH, Holsapple C. Whinston AB Foundation of Decision Support Systems. New York: Academic Press, 2011.
7. Power DJ Decision Support Systems: Concepts and Resources for Managers: Quorum Books, unit Greenwood Publishing, 2002, 272 p.
8. В. Тоценко. Системи підтримки прийняття рішень ваш інструмент для правильного вибору// Комп'ютера. №34. 1998.
9. Keen PGW Decision Support Systems: Next Decades // Decision Support Systems, 2016. V. 3. pp. 253- 265.
10. Power DJ. Web Based Model Driven Decision Support Systems: concepts and issues. Americas Conference on Information Systems, Long Beach, California, 2000.
11. Edwards JS Expert Systems in Management and Administration – Are they really different from Decision support systems? // European Journal of Operational Research, 1992. V. 61. pp. 114-121.
12. Turban, E. Decision support and expert systems: management support systems. Englewood Cliffs, NJ: Prentice Hall, 1995. 887 p.
13. Keen PGW, Scott Morton MS Decision support systems: an organizational perspective. Reading, Mass.: Addison Wesley Pub. Co., 2016.
14. Ginzberg MI, Stohr. by HG Sol. Amsterdam: North Holland Pub. Co, 2013.
15. Alter, S.L. Decision support systems: current practice and permanent functions. Reading, Mass.: Addison Wesley Pub., 2018.
16. Sprague RH, Carlson ED Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice Hall, 2012.
17. Haettenschwiler P. Neuesanwender freundliches Konzeptder Entscheidungsunterstützung. Gutes Entscheidenin Wirtschaft & Politik. Zurich: Hochschulverlag.1999.S. 189-208.

18. Power DJ Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, <http://DSSResources.COM/history/dsshistory.html>, version 2.8, May 31, 2003.
19. Power DJ "What is a DSS?" // The On Line Executive Journal for Data Intensive Decision Support, 1997. v. 1. N3.
20. Holsapple CW, Whinston AB Decision Support Systems: A Knowledge based Approach. Minneapolis: West Publishing Co., 1996.
21. Golden B., Hevner A., Power DJ Decision Insight Systems: A Critical Evaluation // Computers and Operations Research, 2016. v. 13. N2/3. p. 287-300.
22. Marakas GM Decision support systems in 21st century. Upper Saddle River, NJ: Prentice Hall, 1999.
23. Бази даних визначення: навчально-практичний посібник для самостійної роботи студентів [Електронний ресурс] / М. Ю. Лосєв, В. В. Федько. Харків : ХНЕУ ім. С. Кузнеця, 2018. 233 с.
24. Трьохрівнева архітектура ANSI-SPARK. StudFiles. Site. URL : <https://studfile.net/preview/5454386/page:3/> (дата звернення: 10.05.2022).
25. Лосєв М. Ю. Бази даних: навчально-практичний посібник для самостійної роботи студентів [Електронний ресурс] / М. Ю. Лосєв, В. В. Федько. Харків : ХНЕУ ім. С. Кузнеця, 2018. 233 с.
26. Бази даних. Системи управління базами даних. Databaseinform11. Site. URL : <https://sites.google.com/site/databaseinform11/vikoristovuemo-zapiti> (дата звернення: 10.05.2022).
27. Kononow P. Metadata in Relation Databases. *Dataedo*. 30th September 2018. URL : <https://dataedo.com/kb/databases/all/metadata> (дата звернення: 10.05.2022).
28. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч. посібник. Електронне видання, 2018. 118 с.
29. Гайдаржи В.І., Дацюк О.А. Основи проектування та використання баз даних [Текст]: навч. посіб.-2-ге вид., випр. і доповн. К.: Політехніка. 2004. 256 с.
30. Основи баз даних. СУБД Access 2010 (2013). Посібник до вивчення та практичної роботи / автори: Д.А. Покришень, Ю.О.Крепкий, І.Т.Атрошенко, О.П.Дрозд, І.Й.Сподаренко. ТОВ НВП «Інтерсервіс», Чернігів. 2013. 225 с.
31. 12 Codd's Rules. JavaTPoint. Site. URL : <https://www.javatpoint.com/12-codds-rules> (дата звернення: 10.05.2022).
32. Нормалізація реляційних баз даних. *Stud.Com.Ua*. Site. URL : https://stud.com.ua/93792/informatika/normalizatsiya_relyatsiynih_danih (дата звернення: 10.05.2022).
33. Булатецька Л. В., Булатецький В. В. Реляційна алгебра. реляційне числення Методичні вказівки для підготовки до контрольної роботи з норма-

тивних навчальних дисциплін «Бази даних та розподілені інформаційно-аналітичні системи» та «Організація баз даних та знань» Луцьк 2020. 36 с.

34. Мазурець О.В. Організація баз даних та знань: лабораторний практикум для студентів напрямку «Комп'ютерні науки». Хмельницький: ХНУ, 2012. 100 с.

35. Неня А.В. Організація баз даних та знань: конспект лекцій для студентів заочної форми навчання. Суми: СумДУ, 2010. 109 с.

36. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань: навч. посібник. Львів: «Магнолія 2006», 2012. 584с.

37. Експертні системи: Лекції 1 2. Початкові дані. Інженерія знань. URL: http://baklaniv.at.ua/ES/ekspertni_sistemi_lekcija_1_2.pdf (дата звернення: 10.05.2022).

38. Експертні системи як прикладна галузь штучного інтелекту. URL: http://ua.referat.com/Експертні_системи_як_прикладна_галузь_штучного_інтелекту (дата звернення: 10.05.2022).

39. Експертні системи та їх використання. URL: http://ua.referat.com/Експертні_системи_та_їх_використання (дата звернення: 10.05.2022).

40. Експертні системи: інформаційні технології та моделювання. URL: https://pidruchniki.com/10811007/informatika/ekspertni_sistemi (дата звернення: 10.05.2022).

41. Поняття знань та відмінності їх від даних. Studopedia.Su https://studopedia.su/5_36737_ponyattya-znan-ta-vidminnosti-ih-vid-danih.html (дата звернення: 10.05.2022).

42. Аналіз даних: приклади, методи, особливості, реалізація, програмне забезпечення. Site. URL : <https://www.questionpro.de/uk/аналіз-даних/> (дата звернення: 10.05.2022).

43. The Analysis of knowledge. URL : http://www.reading.ac.uk/AcaDepts/ld/Philos/jmp/Theory%20of%20Knowledge/Analysis_of_Knowledge.htm (дата звернення: 10.05.2022).

44. What is Data Analysis: Methods, Process and Types Explained. Data Science & Business Analytics. Site. URL : <https://www.simplilearn.com/data-analysis-methods-process-types-article> (дата звернення: 10.05.2022).

45. Кореляційний аналіз. Матеріал з Вікіпедії – вільної енциклопедії. Site. URL : https://uk.wikipedia.org/wiki/Кореляційний_аналіз (дата звернення: 10.05.2022).

46. Теоретичні основи і принципова схема дисперсійного аналізу. Навчальні матеріали онлайн. Site. URL : https://pidru4niki.com/1929100153035/statistika/dispersiynyi_analiz (дата звернення: 10.05.2022).

47. Задача дисперсійного аналізу. Wiki.Tntu Site. URL : https://wiki.tntu.edu.ua/Задача_дисперсійного_аналізу (дата звернення: 10.05.2022).

48. Регресійний аналіз. *Навчальні матеріали онлайн*. Site. URL : https://pidru4niki.com/17280924/ekonomika/regresiyuniy_analiz (дата звернення: 10.05.2022).
49. Задачі регресійного аналізу. *Студпедія*. Site. URL : https://studopedia.ru/2_90230_zadachi-regresiyynogo-analizu.html (дата звернення: 10.05.2022).
50. Коваріаційний аналіз. *Wiki.Tntu*. Site. URL : https://wiki.tntu.edu.ua/Коваріаційний_аналіз (дата звернення: 10.05.2022).
51. Дискримінантний аналіз. *Uk.Economy-Pedia*. Site. URL : <https://uk.economy-pedia.com/11040389-discriminant-analysis> (дата звернення: 10.05.2022).
52. Кластерний аналіз. *Навчальні матеріали онлайн*. Site. URL : *Навчальні матеріали онлайн*. Site. URL : https://pidru4niki.com/11800912/ekonomika/lkasterniy_analiz (дата звернення: 10.05.2022).
53. Аналіз часових рядів. *Stud.Com.Ua*. Site. URL : https://stud.com.ua/154766/informatika/analiz_chasovih_ryadiv (дата звернення: 10.05.2022).
54. Колодчак О.М. Інтелектуальний аналіз даних. *Lviv Polytechnic National University Institutional Repository* . URL : <https://science.lpnu.ua/sites/default/files/journal-paper/2017/nov/6688/1049-58.pdf> (дата звернення: 10.05.2022).
55. Knowledge Discovery in Databases (KDD). *Techopedia*. Last updated: August 18, 2017. Site. URL : <https://www.techopedia.com/definition/25827/knowledge-discovery-in-databases-kdd> (дата звернення: 10.05.2022).
56. Засоби інтелектуального аналізу даних. *Stud.Com.Ua*. Site. URL : https://stud.com.ua/121125/informatika/zasobi_intelektualnogo_analizu_danih (дата звернення: 10.05.2022).
57. Добування даних. *Матеріал з Вікіпедії – вільної енциклопедії*. Site. URL : [https://uk.wikipedia.org/wiki/Добування_даних#:~:text=Добування%20даних%20\(англ.,задачі%20класифікації%2C%20моделювання%20та%20прогнозування](https://uk.wikipedia.org/wiki/Добування_даних#:~:text=Добування%20даних%20(англ.,задачі%20класифікації%2C%20моделювання%20та%20прогнозування) (дата звернення: 10.05.2022).
58. Knowledge discovery in databases. *Techopedia*. Last updated: August 18, 2017. Site. URL : <https://www.techopedia.com/definition/25827/knowledge-discovery-in-databases-kdd> (дата звернення: 10.05.2022).
59. Плєскач В.Лі., Затонацька Т.Г. Інформаційні системи і технології на підприємствах: підручник. Знання. 2018. 718 с.
60. Rockar Н.М. Н. Business intelligence – at the speed of light. ResearchGate.Net. January 2009. Site. URL: https://www.researchgate.net/publication/255639895_B_USINESS_INTELLIGENCE_-_AT_THE_SPEED_OF_LIGHT (дата звернення: 10.05.2022).

61. Wan der Aaist, W. Process Cubes: Slicing, Dicing, Rolling Up and Drilling Event Data for Process Mining. *ResearchGate.Net*. August 2013. Site. URL: https://www.researchgate.net/publication/291246021_Process_Cubes_Slicing_Dicing_Rolling_Up_and_Drilling_Down_Event_Data_for_Process_Mining (дата звернення: 10.05.2022).
62. Oracle, Sybase, DB2, and Informix Parameters. *IBM.com*. Last Updated: 2021-08-18. Site. URL: <https://www.ibm.com/docs/kk/app-connect-pro/7.5.4?topic=activities-oracle-sybase-db2-informix-parameters> (дата звернення: 10.05.2022).
63. Орієнтовані на дані СППР. *StudAll.org*. Site. URL: <https://studall.org/all3-98088.html> (дата звернення: 10.05.2022).
64. Основні відмінності систем оброблення транзакцій (oltp) і аналітичних систем. *StudAll.org*. Site. URL: <https://studall.org/all3-98131.html> (дата звернення: 10.05.2022).
65. Taylor, D. What is Data Warehouse? Types, Definition & Example. *Guru99*. Updated June 18, 2022. Site. URL: <https://www.guru99.com/data-warehousing.html> (дата звернення: 10.05.2022).
66. Класифікація СППР. *Pidru4niki.Com*. Site. URL: https://pidru4niki.com/1058032047767/informatika/klasifikatsiya_sppr (дата звернення: 10.05.2022).
67. Бідюк П.І., Коршевніук Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник. Київ: РПК «ІПСА» НТУУ «КПІ», 2010. 340 с.
68. Сховища даних. *StudFile.Net*. Site. URL: <https://studfile.net/preview/9708436/page:12/> (дата звернення: 10.05.2022).
69. Anamika Sharma. 8 Insanely Useful Ways How Data Warehouse Can Improve Your Business Reporting. *DataToBi*. Site. URL: <https://www.datatobiz.com/blog/data-warehouse-improves-business-reporting/> (дата звернення: 10.05.2022).
70. 10 Importing Design Definitions from Oracle Designer. Oracle Database Online Documentation 11g, Release 2 (11.2). *ORACLE Help Centre*. Site. URL: https://docs.oracle.com/cd/E11882_01/owb.112/e10582/imp_thirdparty_design.htm#WBDDOD10504 (дата звернення: 10.05.2022).
71. Узагальнена схема архітектури сховища даних. *Stud All*. Site. URL: <https://studall.org/all3-98132.html> (дата звернення: 10.05.2022).
72. Inmon, W.H. OLAP and Data Warehouse. *DocPlayer.Net*. 2000. Site. URL: <http://docplayer.net/15369163-Olap-and-data-warehouse-by-w-h-inmon.html> (дата звернення: 10.05.2022).
73. Holt, J. How to Implement Visual control in Your Warehouse. *Kenco Group.Com*. Site. URL: <https://blog.kencogroup.com/how-to-implement-visual-control-in-your-warehouse> (дата звернення: 10.05.2022).

74. SAS Institute. From Wikipedia, the free encyclopedia. Site. URL: https://en.wikipedia.org/wiki/SAS_Institute (дата звернення: 10.05.2022).
75. Taylor, D. What is MOLAP (Multidimensional OLAP) in Data Warehouse? *Guru99*. Site. URL: <https://www.guru99.com/multidimensional-online-analytical-processing.html> (дата звернення: 10.05.2022).
76. Relational Databases For Data Warehousing. White Paper. Logistics Online. Site. URL: <https://www.logisticsonline.com/doc/relational-databases-for-data-warehousing-0001> (дата звернення: 10.05.2022).
77. Kranz, G. Metadata. *TechTarget*. Site. URL: <https://www.techtarget.com/whatis/definition/metadata> (дата звернення: 10.05.2022).
78. What is Data Mart? *Oracle.Com*. Site. URL: <https://www.oracle.com/cis/autonomous-database/what-is-data-mart/> (дата звернення: 10.05.2022).
79. Проектування сховищ даних. *Helpiks.Org*. Site. URL: <https://helpiks.org/7-49599.html> (дата звернення: 10.05.2022).
80. Intro to OLAP: Codd defined 12 Rules. URL: <https://homel.vsb.cz/~dan11/dzdb/Codd%2012%20rules%20of%20OLAP.pdf> (дата звернення: 10.05.2022).
81. Taylor, D. What is OLAP? Cube, Analytical Operations in Data Warehouse. *Guru99*. Site. URL: <https://www.guru99.com/online-analytical-processing.html> (дата звернення: 10.05.2022).
82. SAP BusinessObjects Business Intelligence Suite. Sap.Com. Site. URL: <https://www.sap.com/italy/products/technology-platform/bi-platform.html> (дата звернення: 10.05.2022).
83. OLAP Tools. *Software Advice*. Site. URL: <https://www.softwareadvice.com/bi/olap-comparison/> (дата звернення: 10.05.2022).
84. Oracle Express. *Racle FAQ's*. Site. URL: <https://www.orafaq.com/wiki/Express> (дата звернення: 10.05.2022).

Навчальний посібник

Балтовський О.А., Форос Г.В.,
Пядишев В.Г., Сіфоров О.І.

СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Навчальний посібник

Підписано до друку 29.11.2022. Формат 60x84/16. Папір офсетний.
Гарнітура «Times New Roman»/ Друк цифровий. Ум. друк .арк. 8,6
Наклад 50 прим.
Видавництво ОДУВС
м. Одеса, вул. Успенська, 1
Свідоцтво суб'єкта видавничої справи ДК № 3507 від 25.06.2009 р.