

## 4.2 Бази даних і знань як інформаційні моделі

Для *автоматизованих систем керування виробництвом, установами, адміністративними територіями тощо* характерні задачі *прийняття рішень* на основі великої кількості даних, накопичених за великий проміжок часу. Найчастіше ці дані зберігаються у вигляді *баз даних* (БД), операції з якими виконуються за допомогою *систем управління базою даних* (СУБД). Узагальнена структура системи прийняття керівних рішень з використанням СУБД зображена на рис. 4.2.

Алгоритм прийняття рішення можна розглядати як перетворення  $F$  задання  $X$  на рішення  $Y$  з параметрами, які отримуються на основі бази даних  $B$  і опису ситуації  $I$

$$Y = F(I, B)[X] \quad (4.15)$$

Таким чином, набори даних  $I$  і  $B$  є інформаційною моделлю системи прийняття рішення.

Алгоритм прийняття рішень у такій системі залежить від призначення системи і особливостей її застосування, отже, для різних систем алгоритми відрізняються. Деяку іншу структуру має інформаційна модель системи прийняття рішень на основі *бази знань*.

База знань складається з набору відомих *фактів* (бази даних) і набору відомих залежностей між ними (*правил*). Алгоритм прийняття рішень на основі фактів і правил практично не залежить від призначення системи управління. Конкретні особливості системи впливають лише на структуру бази даних і вид правил.

Головне питання, яке розглядається при побудові бази даних як інформаційної моделі, – вибір адекватного і одночасно зручного опису типу зв'язку між елементами даних.

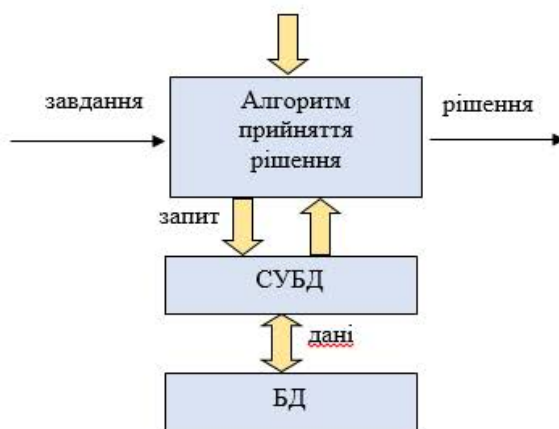


Рисунок 4.2 – Система прийняття рішень

Сучасні СУБД ґрунтуються на використанні *моделей даних* (МД), які дозволяють описувати об'єкти предметних областей і взаємозв'язки між ними. Існують три основні МД і їх комбінації, на яких ґрунтуються СУБД: реляційна (РМД), мережева (ММД), ієрархічна (ІМД).

Основна відмінність між цими моделями даних полягає у способах опису взаємодії між об'єктами й атрибутами. Взаємозв'язок виражає відношення між множинами даних. Використовують взаємозв'язки “один до одного”, “один до багатьох” і “багато до багатьох”. “Один до одного” – це взаємно однозначна відповідність, яка встановлюється між одним об'єктом і одним атрибутом. “Багато до багатьох” – це відповідність між багатьма об'єктами й багатьма атрибутами. Взаємозв'язки між об'єктами й атрибутами зручно зображувати у вигляді графів.

На персональних комп'ютерах в основному використовують СУБД, які підтримують *реляційну модель даних*. Відповідно до реляційної моделі база даних подається у вигляді сукупності таблиць, над якими можуть виконуватися операції, що формулюються у термінах реляційної алгебри і реляційного числення. В реляційній моделі операції над об'єктами бази даних мають теоретико-множинний характер.

Основними елементами реляційної БД є *атрибути, кортежі, відношення*. Відношенням називається деяка сукупність об'єктів, яка характеризується однаковим набором атрибутів. Зручно подавати відношення як таблицю, де кожний рядок є кортеж і кожний стовпець є атрибут. Стовпці таблиці – це елементи даних, а рядки – записи.

Таким чином, реляційна база даних подається четвіркою  $\langle E, A, C, R \rangle$ , де  $E$  – множина відношень,  $A$  – множина атрибутів,  $C$  – множина кортежів,  $R$  – мно-

жина зв'язків.

Відношення може бути подане у вигляді файлу. Записи у файлі відповідають кортежам. Вони складаються з полів, які відповідають атрибутам відношення. Зв'язок між відношеннями здійснюється через *ключі*. Відношення не повинно мати двох кортежів, в яких збігаються всі атрибути ключа.

Найчастіше реляційні бази даних зображуються у вигляді ER-моделі (Entity-Relationship Model). Ця модель подає відношення і зв'язки між ними через ключі у графічному вигляді.

Основна перевага реляційного підходу – його простота й доступність, незалежність даних, гнучкість, теоретичне обґрунтування на основі реляційної алгебри.

В *мережній моделі даних* елементарні дані і відношення між ними подаються у вигляді орієнтованого графу (вершини – дані, дуги – відношення). Основні елементи мережної бази даних – тип запису і тип набору.

*Запис* – сукупність логічно пов'язаних полів, яка характеризується іменем і полями, що входять до нього. *Поле* називається єдина неподільна одиниця інформації, яка характеризується ідентифікатором, типом і довжиною.

Запис може існувати в БД не тільки самостійно, але й бути одночасно детальним або головним записом деяких наборів залежно від того, чи описаний його тип як тип головного запису або детального запису набору. В записах можуть міститися довільні елементи даних, значення яких залежать від значень інших елементів даних того запису, в який входить даний запис.

Основні переваги мережної моделі даних – простота реалізації відношень “багато до багатьох”. Основний недолік – її складність. При реорганізації БД можлива втрата незалежних даних.

*Ієрархічна модель даних* основана на понятті *дерев*, які складаються з вершин і ребер. Вершина дерева ставиться у відповідність сукупності атрибутів даних, що характеризують деякий об'єкт. Вершини і ребра дерева утворюють ієрархічну структуру, яка складається з  $n$  рівнів.

Першу вершину називають *кореневою вершиною*. Вона задовольняє умови:

1. Ієрархія починається з кореневої вершини;
2. Кожна вершина відповідає одному або декільком атрибутам;
3. На рівнях з більшим номером знаходяться залежні вершини. Вершина попереднього рівня є початковою для нових залежних вершин;
4. Кожна вершина, яка знаходиться на рівні  $i$ , з'єднана з однією вершиною рівня  $i-1$ , за винятком кореневої вершини;
5. Коренева вершина може бути пов'язана з однією або кількома залежними вершинами;
6. Доступ до кожної вершини здійснюється через кореневу єдиним шляхом;

7. Існує довільна кількість вершин кожного рівня.

Основні переваги ієрархічної моделі даних: простота побудови і використання, забезпечення певного рівня незалежності даних, простота оцінювання характеристик. Основні недоліки: відношення “багато до багатьох” реалізується дуже складно, дає громіздку структуру і вимагає зберігання надлишкових даних, ієрархічна впорядкованість ускладнює операції вилучення і занесення даних, доступ до будь-якої вершини можливий лише через кореневу, що збільшує час доступу.

Існує декілька способів здійснення операцій з інформаційними моделями у вигляді баз даних. Серед них оперування з даними за допомогою інтегрованого діалогового середовища системи управління базою даних (СУБД), оперування за допомогою спеціалізованої проблемно-орієнтованої програми; оперування за допомогою запитів на спеціальній мові (SQL). Останній спосіб в зв'язку з розповсюдженням мережних інформаційних технологій поступово набуває найбільшого розповсюдження.

SQL (англ. Structured Query Language — мова структурованих запитів) — універсальна комп'ютерна мова, яка застосовується для створення, модифікації і управління даними в реляційних базах даних.

SQL ґрунтується на реляційній алгебрі. Основними операціями, за допомогою яких модифікується база даних, є вставка, вилучення і модифікація.

Мова SQL поділяється на три частини:

- оператори означення даних;
- оператори маніпулювання даними;
- оператори визначення доступу до даних.

Запити до бази даних повертають результат у вигляді таблиць, які теж можуть використовуватися як об'єкт запитів.

У загальному випадку запит до бази даних можна подати як операторне перетворення

$$\langle E', A', C', R' \rangle F(S) \{ \langle E, A, C, R \rangle \}, \quad (4.16)$$

де  $S$  — множина параметрів запиту.

Вираз (4.16) є формальним записом інформаційної моделі системи.

В системах з використанням елементів штучного інтелекту широко застосовуються інформаційні моделі у вигляді *баз знань*. База знань звичайно складається з двох частин: бази фактів і бази правил. *База фактів* – це фактично база даних про предметну область (загальні дані) і конкретну систему. *База правил* встановлює логічні зв'язки між фактами у вигляді

*якщо* (факт1, факт2, ... фактN), *то* фактX.

Наявність логічних зв'язків між фактами є головною відмінністю баз знань від баз даних, адже одною з головних вимог до баз даних є їх «нормальна фор-

ма», яка саме і передбачає відсутність будь-яких функціональних або логічних зв'язків між даними.

Отримання логічного висновку на основі бази знань може здійснюватися шляхом послідовного застосування правил. Тоді з початкових фактів отримують спочатку проміжні, а зрештою, і остаточні висновки. Можливий також спосіб отримання висновку, при якому правила об'єднуються за допомогою логічних операцій. Такий логічний вираз є оператором над базою знань, який встановлює зв'язок між початковими даними і висновком–результатом.