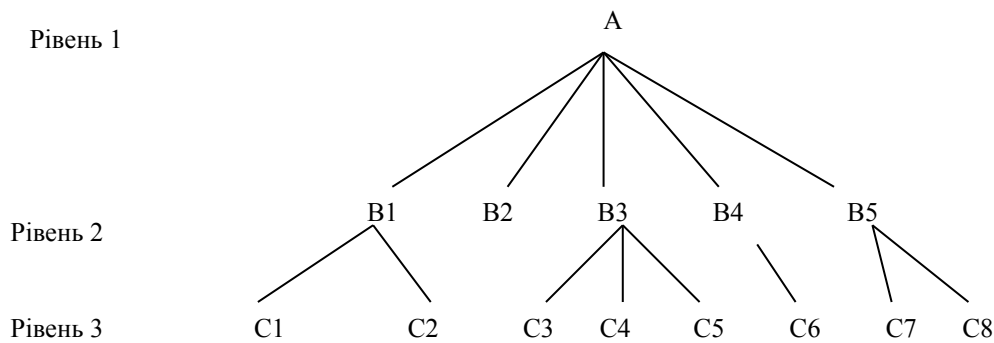


Ієрархічна модель даних

Ієрархічна модель організує дані у вигляді деревовидної структури (дерева).

До основних понять ієрархічної структури відносяться: рівень, елемент (вузол), зв'язок. Дерево представляє собою ієрархію елементів, які називаються вузлами. Вузол - це сукупність атрибутів даних, що описують деякий об'єкт. На самому верхньому рівні ієрархії є один і тільки один вузол - корінь. Кожний вузол, крім кореня, пов'язаний з вузлом на більш високому рівні, який має назву вихідним для даного вузла. Жоден елемент не має більш одного вихідного. Кожний елемент може бути зв'язаний з іншим або декількома елементами на більш низькому рівні. Вони мають назву породженими (мал.6).



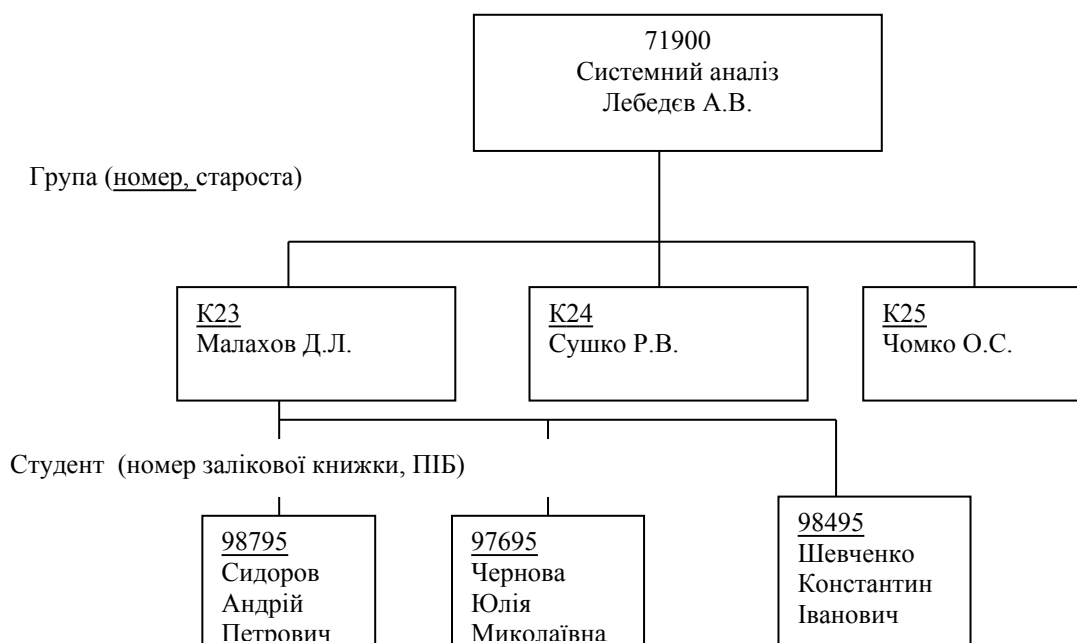
Мал. 6. Графічне зображення ієрархічної структури БД

До кожного запису бази даних існує лише один (ієрархічний) шлях від кореневого запису. Наприклад, як видно з мал.6, для запису C4 шлях проходить через записи A і B3,

Приклад, поданий на мал.7 ілюструє застосування ієрархічної моделі бази даних. Для запропонованого прикладу ієрархічна структура є виправданою, бо кожен студент вчиться у конкретній (лише одній) групі, яка відноситься до конкретного (лише одного) факультету.

Мал.7 Приклад ієрархічної структури БД

Факультет (спеціальність, назва, гарант)



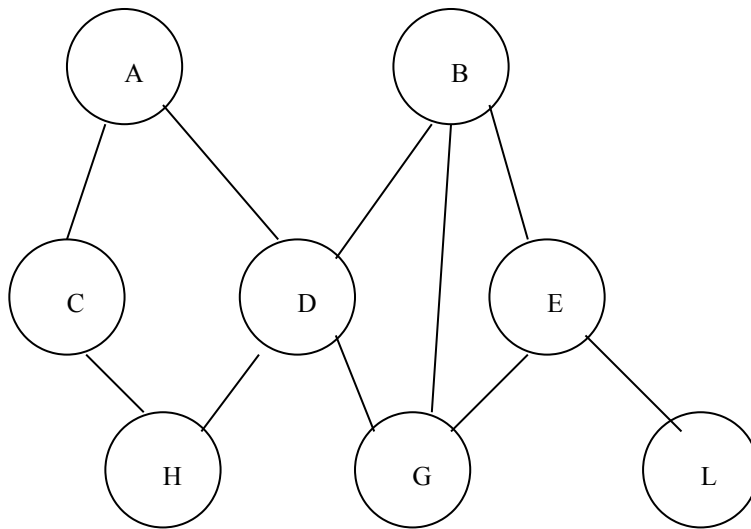
Мережева (сітьова) модель даних

Мережева модель організує дані у вигляді сітьової мережевої структури.

Структура називається мережевою, якщо у відношеннях між даними кожен елемент може мати більш за один вихідний зв'язок (на відміну від ієрхічної моделі, коли кожен запис-нащадок має лише один зв'язок з вузлом (записом) більш високого рівня).

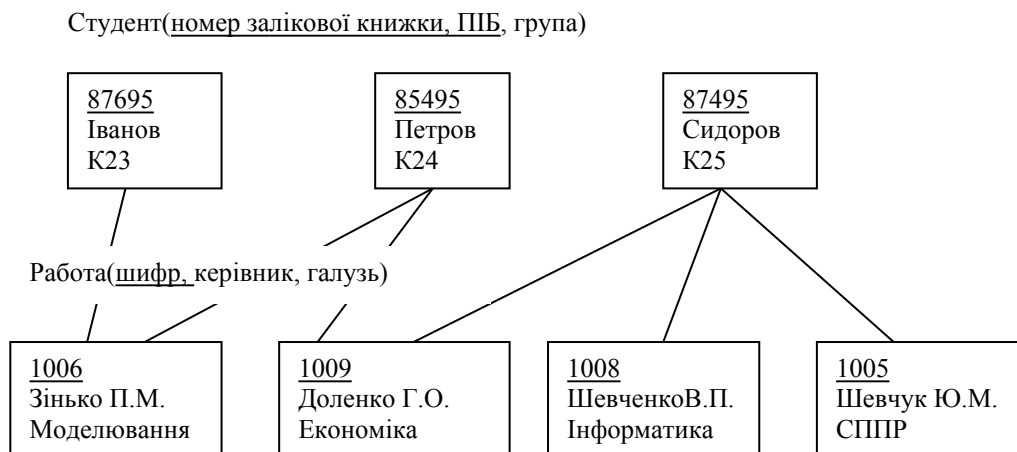
Іншими словами, у мережевій структурі при тих самих основних поняттях (рівень, вузол, зв'язок) кожен елемент може бути пов'язаний з довільним іншим елементом.

На мал.8 зображено мережеву структуру бази даних у вигляді графа.



Мал.8. Графічне зображення мережевої (сітьової) структури у вигляді неорієнтованого графу

Прикладом складної мереженої структури може служити структура бази даних, що містить відомості про студентів, які приймають участь у науково-дослідних роботах (НДРС). Можлива участь одного студента в декількох НДРС, а також участь декількох студентів в розробці однієї НДРС. Графічне зображення описаної мережевої структури, яка складається лише з двох типів записів, показано на мал.9. Єдине відношення представляє собою складний зв'язок між записами в обох напрямленнях.



Мал.9. Приклад мережевої структури БД

Реляційна модель даних

Поняття "реляційний" (англ. relation - відношення) пов'язане з роботами відомого американського спеціаліста в області систем баз даних Е. Кодда.

Ці моделі характеризуються простотою структури даних, зручним для користувача табличним поданням і можливістю використання формального апарату алгебри відношень і реляційного числення для обробки даних.

Реляційна модель зорієнтована на організацію даних у вигляді двовимірних таблиць. За допомогою таблиці зручно описувати найпростіший вид зв'язку між даними: власне, ділення одного об'єкта на множину підоб'єктів (кожен з яких є записом або рядком у таблиці). При цьому, кожен рядок має однакову структуру, яка описується відповідним набором полів з чітко визначеними типами. Виходячи з простоти подання даних у таблиці, немає іншого способу встановлення зв'язків, і тому використовується зв'язування таблиць. Наприклад, реляційною таблицею можна подати інформацію про студентів, які навчаються у виші (мал. 10).

№ особ. справи	Прізвище	Ім'я	По-батькові	Дата народж.	Група
16493	Сергєєв	Петро	Михайлович	01.01.76	K23
16593	Петрова	Анна	Володимирівна	15.03.75	K24
16693	Науменко	Андрій	Борисович	14.04.76	K25

Мал.10. Приклад реляційної таблиці

Реляційна модель даних є сукупністю взаємопов'язаних двовимірних таблиць об'єктів моделі.

Зв'язки між двома логічно пов'язаними таблицями в реляційній моделі встановлюються на основі рівності значень однакових атрибутів цих таблиць.

Кожна реляційна таблиця представляє собою двовимірний масив і описується такими властивостями:

- кожний елемент таблиці - один елемент даних (неділимість даних, прості типи);
- усі стовбці у таблиці однорідні, тобто усі елементи в стовбці мають один тип (числовий, символічний і т.д.) та довжину;
- кожний стовбець має унікальне ім'я;
- однакові рядки у таблиці відсутні;
- порядок слідування рядків і стовбців може бути довільним.

При описуванні реляційної моделі часто використовують такі терміни: відношення, кортеж, домен.

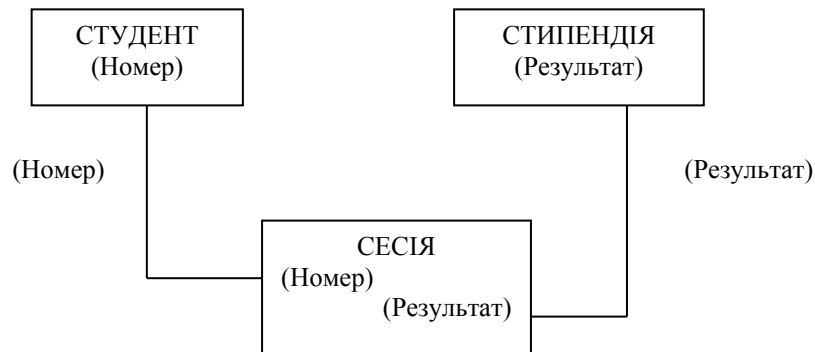
Відношення подаються у вигляді таблиць, рядки яких відповідають записам (кортежам), а стовбці полям, атрибутам відношень (доменам).

Поле, кожне значення якого однозначно визначає відповідний запис, називається простим ключом (ключовим полем). Якщо записи однозначно визначаються значеннями декількох полів, то така таблиця бази даних має складений ключ

У прикладі, наведеному на мал.10, ключовим полем таблиці є "№ особ. справи".

Як все було сказано, між двома реляційними таблицями можуть бути сформовані зв'язки. Різні таблиці, можуть бути пов'язані між собою через спільне поле даних.

На мал.11 наведено приклад реляційної моделі, побудованої на основі відношень: СТУДЕНТ, СЕСІЯ, СТИПЕНДІЯ.



Мал.11. Приклад реляційної моделі

Таблиця СТУДЕНТ має поля: Номер п/п, ПІБ, Дата народження, Група;
СЕСІЯ – Номер ст., Оцінка 1, Оцінка 2, Оцінка 3, Оцінка 4, Результат;
СТИПЕНДІЯ - Результат, Відсоток

Таблиці СТУДЕНТ та СЕСІЯ мають ключі, що співпадають (Номер), що дає можливість легко організувати зв'язок між ними. Таблиця СЕСІЯ має первинний ключ «Номер ст.» і містить зовнішній ключ Результат, який забезпечує її зв'язок з таблицею СТИПЕНДІЯ.

Завдяки такій організації досягаються такі переваги:

1. Вдається уникнути дублювання інформації. Усі необхідні дані можна зберігати лише в одній таблиці. Так, наприклад, немає необхідності в таблиці СЕСІЯ зберігати номер групи кожного студента, який здає іспити, достатньо лише задати зв'язок з таблицею СТУДЕНТ.

2. В реляційних базах даних легко проводити зміни. Якщо у таблиці СЕСІЯ змінити будь-які значення, то коректна інформація автоматично буде пов'язана з іншими таблицями, що посилаються на першу (наприклад, таблиця СТИПЕНДІЯ).

3. В нереляційних базах даних складно передати усе наявні залежності, тобто зв'язати один з одним дані з різних таблиць. Реляційна база даних виконує всі ці дії автоматично.

4. В реляційних базах даних вдається легко уникнути встановлення помилкових зв'язків між різними таблицями даних, а необхідний обсяг пам'яті скорочується до мінімуму.

Таким чином, виділяємо основні поняття реляційних БД:
таблиця, запис (кортеж), поле (атрибут), домен, відношення, ключ

Типи даних

Значення даних, що зберігаються у таблицях, є строго типізованими. Поняття типу даних відповідає поняттю типу даних в мовах програмування. Традиційне визначення типу даних складається з трьох компонентів: множини значень типу, набору операцій, способу подання.

Типи даних в різних СУБД можуть подаватися різними ключовими словами, перелік типів, що підтримується, також може бути різним. Серед найбільш відомих типів назовемо символічний, числовий, спеціальні числові типи (наприклад, фінансові дані), темпоральні типи (час, дата, інтервал).

Серед усіх понять необхідно зупинитися більш детально на понятті «домен». У загальному вигляді поняття домен визначається 2 властивостями: базовим типом даних та довільним логічним виразом, який визначає належність даних до даного домену (це називається обмеженням домену). Наприклад, за допомогою цілого типу можна описувати вік, кількість деталей або покупок, оцінки на іспитах і т.і. Однак, кожного разу логічне призначення атрибута, формалізоване у вигляді обмеження, дозволяє звужити множину цілих значень до реалістичної.

Зрозуміло, що домен має своє ім'я, яке є унікальним у сукупності доменів бази даних. І останній важливий факт. Для проведення операцій з даними (наприклад, порівняння), потрібно пам'ятати про семантику домену: дані обробляються та порівнюються лише у тому випадку, якщо вони належать одному домену.