

СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ

Поняття «Система баз даних»

- У широкому значенні **система баз даних** розуміється фактично як синонім поняття «Інформаційна система»
- **Система баз даних** - це комп'ютеризована система, призначена для збереження інформації, яка представлена у вигляді записів і може бути корисною для даного користувача або організації, які її експлуатують. Система баз даних забезпечує зручні засоби для отримання та модифікації необхідної інформації. Інформація, яка розміщується в записах, є базою даних

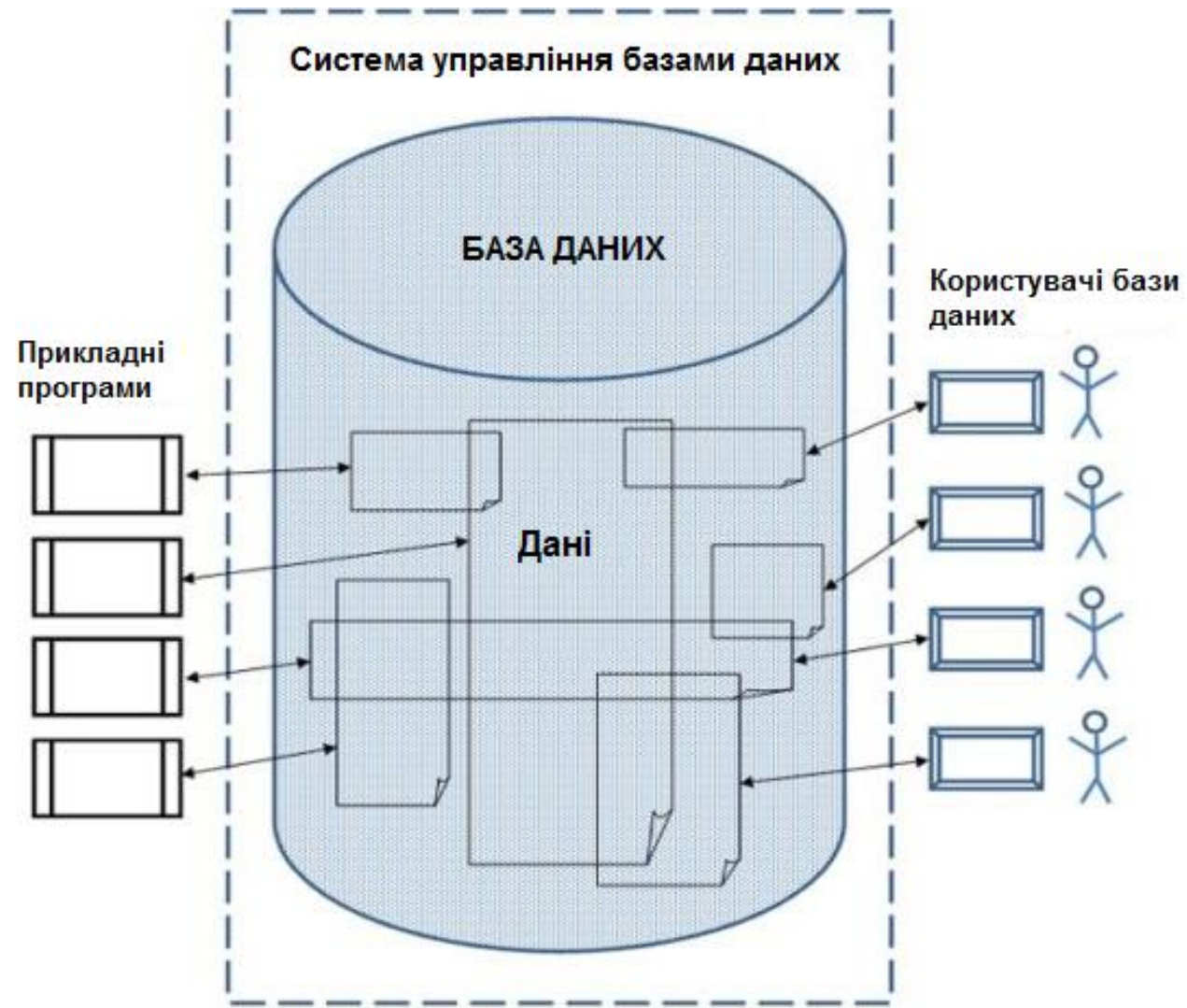
Класифікація системи баз даних за кількістю користувачів

- **однокористувацькі (single user system).** В однокористувацьких системах доступ до бази даних одночасно може мати не більше одного користувача
- **багатокористувацькі (multi user system).** У багатокористувацькій системі доступ до бази даних можуть мати одночасно кілька користувачів. Для користувача робота багатокористувацькій системі нічим не відрізняється від роботи в однокористувацькій системі

Схема багатокористувацької бази даних

ОСНОВНІ КОМПОНЕНТИ системи баз даних:

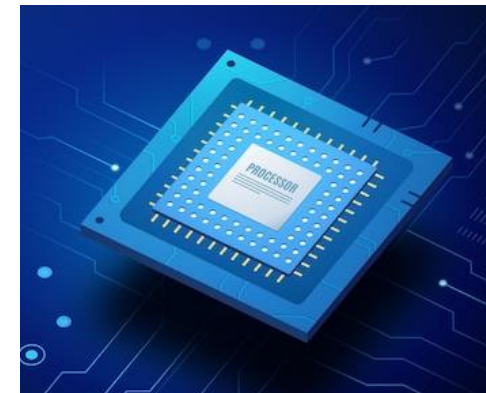
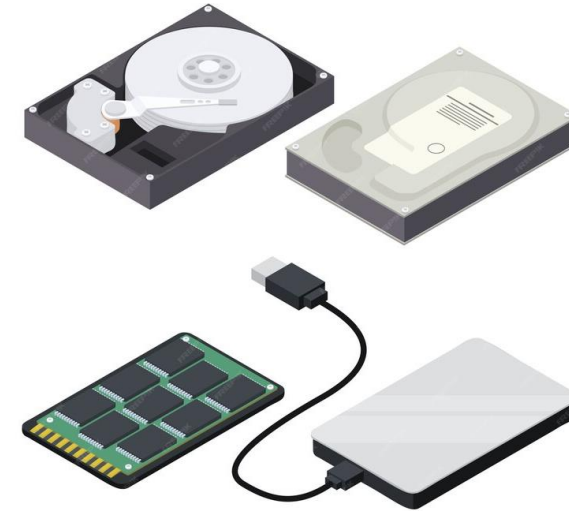
- дані;
- апаратне забезпечення;
- програмне забезпечення;
- користувачі



Апаратне забезпечення системи баз даних

Основні категорії пристроїв

- засоби, що забезпечують збереження інформації: зовнішня пам'ять (накопичувачі), відповідні пристрої введення виводу, контролери пристроїв, канали вводу-виводу тощо;
- засоби, що підтримують роботу програмного забезпечення системи баз даних: процесори та оперативна пам'ять



Програмне забезпечення

- система управління базами даних СУБД (Database Management System - DBMS). Систему управління базами даних називають менеджером баз даних (Database Manager) або сервером баз даних (Database Server). СУБД є рівнем, що пов'язує між собою фізичну базу даних (реально збережені дані) і користувача системи;
- утиліти для роботи з базою даних;
- засоби розробки додатків, які взаємодіють із відомими базами даних;
- засоби проектування баз даних;
- генератори звітів на основі даних, розміщених у базі даних;
- менеджер транзакцій (transaction manager) або диспетчер виконання транзакцій (TP monitor)

Класифікація користувачів бази даних

- **прикладні програмісти** – люди, які розробляють програмне забезпечення, що використовує базу даних;
- **кінцеві користувачі** – працюють із базою даних за допомогою спеціального програмного забезпечення. Як правило, програмне забезпечення або користувач працює з базою даних за допомогою спеціальної мови запитів. Найбільш поширена мова запитів до бази даних – це мова SQL (structured query language);
- **адміністратори баз даних** – люди, які забезпечують необхідну технічну підтримку бази даних. Вони здійснюють створення бази даних, організовують технічний контроль, проводять дії із забезпечення необхідної швидкодії та його технічне обслуговування.

Система управління базами даних

Система управління базами даних (СУБД) - це комплекс мовних і програмних засобів, призначений для створення, ведення та спільного використання БД багатьма користувачами.

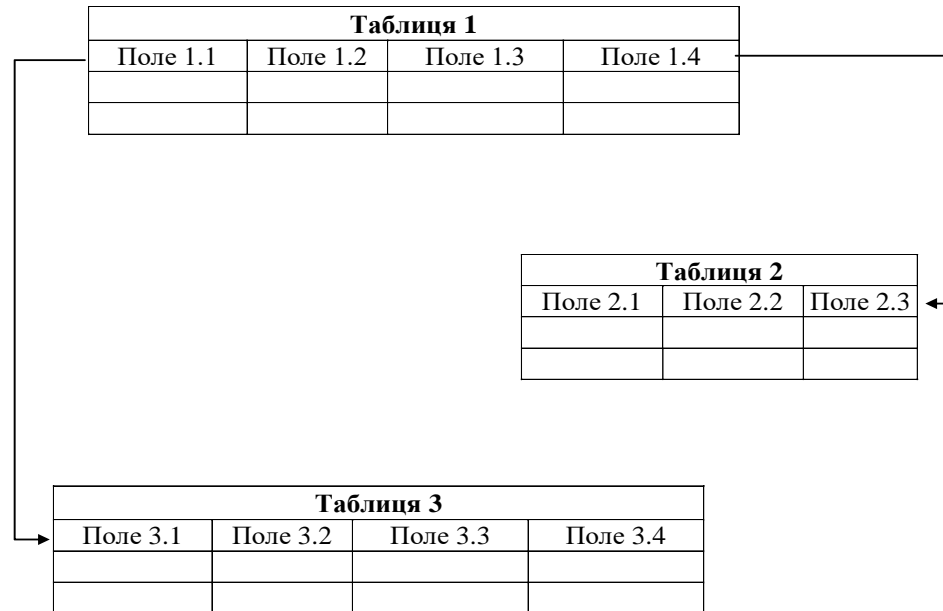
Зазвичай СУБД розрізняють за використовуваної моделі даних. СУБД, засновані на використанні реляційної моделі даних, називають **реляційними СУБД**

Функції СУБД



Реляційна база даних

- реляційна модель зберігання даних базується на зберіганні даних у вигляді взаємозв'язаних таблиць. Зв'язок між таблицями може бути реалізований за деяким полем і називається відношенням (relation)
- Найпростіша база даних містить одну таблицю, складні – десятки й сотні



Реляційна база даних

Номер	Прізвище, ініціали	Дата народження	Спеціальність	Стипендія
1	Петренко А.І.	02.11.2006	Комп науки	так
2	Копач М.П.	13.05.2007	ІСТ	ні
3	Маляр Р.В.	23.12.2006	Комп науки	ні
4	Діденко Н.К.	09.09.2005	Прогр інженер	так
5	Дзюба Н.М.	19.03.2006	ІСТ	ні

Які таблиці з даними можуть бути пов'язані з цією таблицею?

З цією таблицею пов'язані таблиці, що містять дані про успішність (оцінки), викладачів дисциплін, наявність академічних боргів....

Проектування бази даних

Етапи проектування БД:

- дослідження предметної області;
- аналіз даних (сутностей та їх атрибутів);
- визначення відносин між сутностями і визначення первинних і вторинних (зовнішніх) ключів

При проектуванні структури баз даних необхідно уникати повторення даних і створювати для них окремі таблиці. Цей процес називається **нормалізацією**

Не кожна таблиця може бути об'єктом бази даних. Щоб таблиця була об'єктом БД необхідно здійснити **нормалізацію**

Створена таблиця повинна узгоджуватись з основними вимогами до таблиць в реляційній БД



Базові поняття моделі БД «сутність - зв'язок»

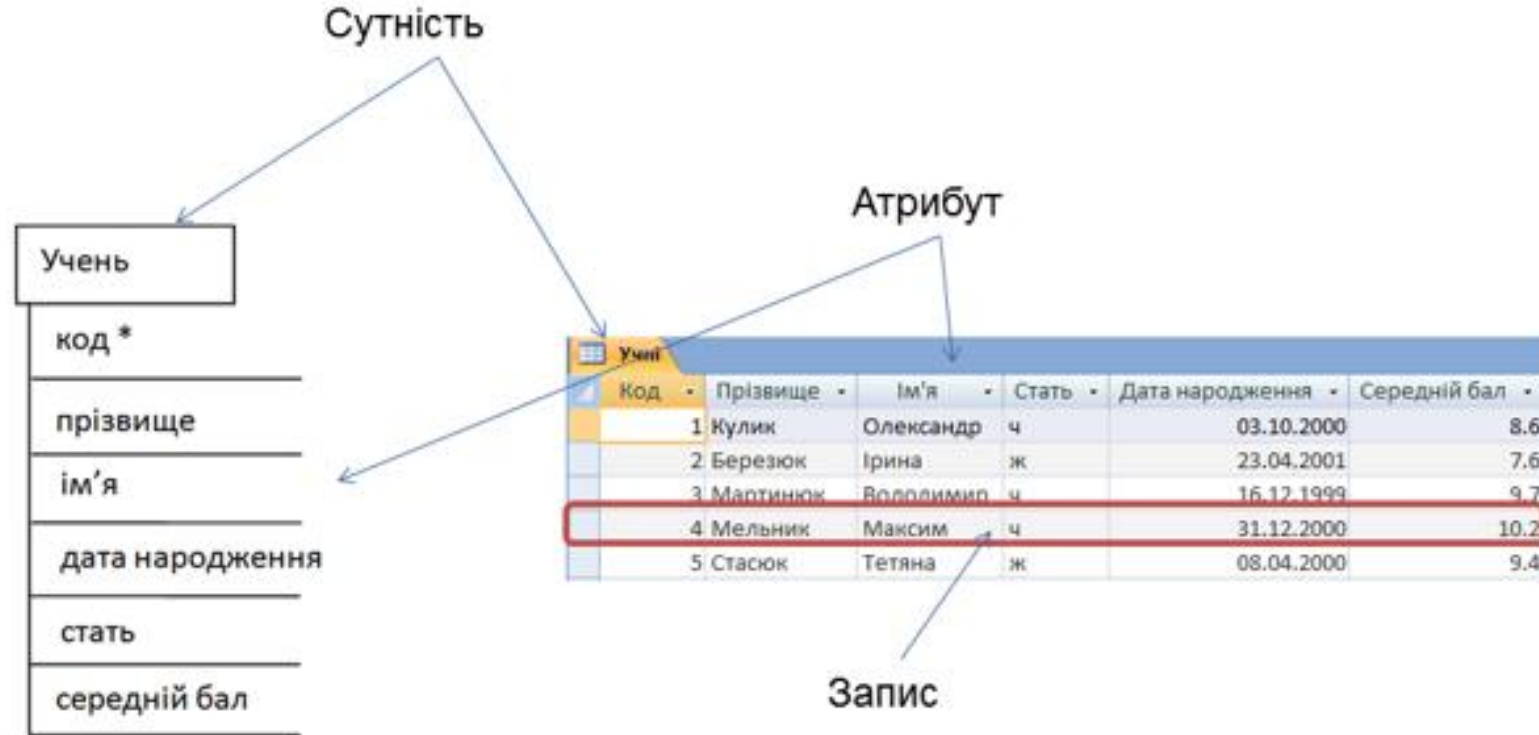
- **Сутність** - будь-який конкретний або абстрактний об'єкт у розглянутій предметній області. Сутності - це базові типи інформації, які зберігаються в БД (в реляційній БД кожній сутності призначається таблиця). Сутність - це те, про що зберігається інформація. Наприклад, студенти, клієнти, підрозділи
- Поняття **тип сутності** відноситься до набору однорідних особистостей, предметів або подій, що виступають як ціле (наприклад, студент, клієнт і т.д.). **Екземпляр сутності** відноситься, наприклад, до конкретної особистості в наборі. Типом сутності може бути студент, а екземпляром - Петров, Сидоров і т. д.

Базові поняття моделі БД «сутність - зв'язок»

Атрибут - це властивість сутності в предметної області. Його найменування повинне бути унікальним для конкретного типу сутності. Наприклад, для сутності студент можуть бути використані наступні атрибути: прізвище, ім'я, по батькові, дата і місце народження, паспортні дані і т.д.

У реляційній БД атрибути зберігаються в полях таблиць

Сутності, атрибути, записи в базі даних



Структура таблиці – це набір поіменованих полів, в яких описують властивості об'єктів

Базові поняття моделі БД «сутність - зв'язок»

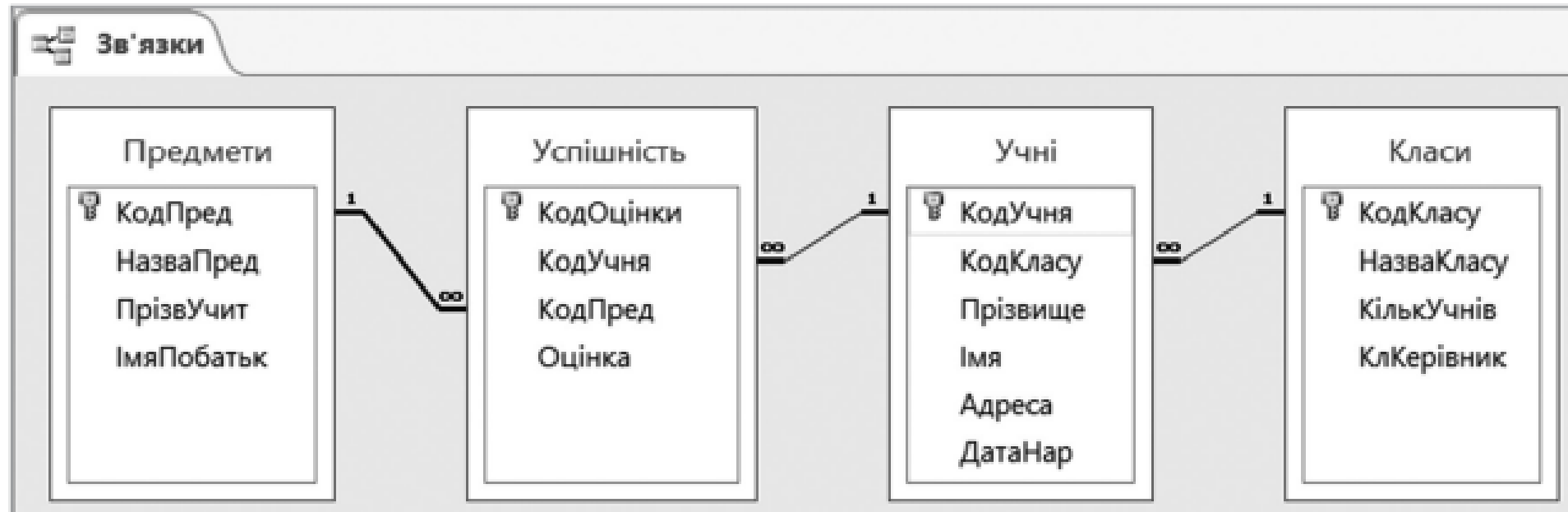
Зв'язок - взаємозв'язок між сутностями в предметній області. Зв'язки представляють собою з'єднання між частинами БД (в реляційній БД - це з'єднання між записами таблиць)

Види логічного зв'язку. Зв'язок встановлюється між двома загальними полями (стовпцями) двох таблиць:

- **«один до одного»** - кожному запису з однієї таблиці відповідає один запис у іншій таблиці;
- **«один до багатьох»** - кожному запису з однієї таблиці відповідає кілька записів іншій таблиці;
- **«багато до багатьох»** - безлічі записів з однієї таблиці відповідає кілька записів в іншій таблиці

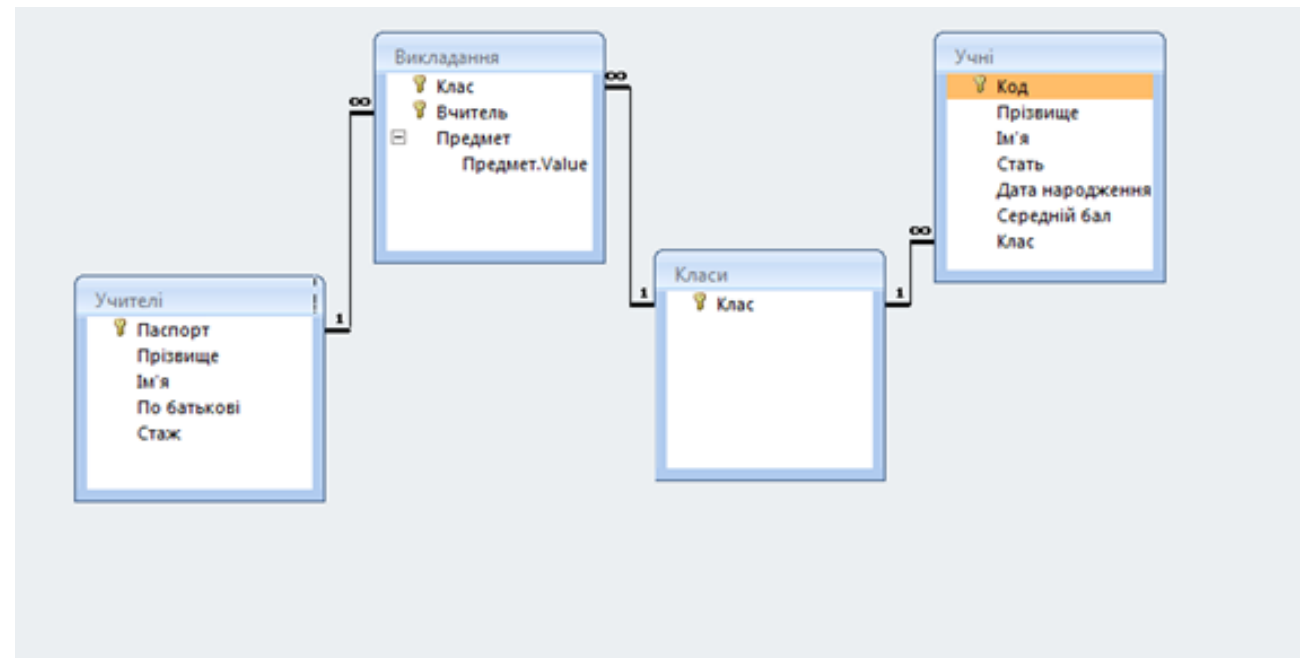
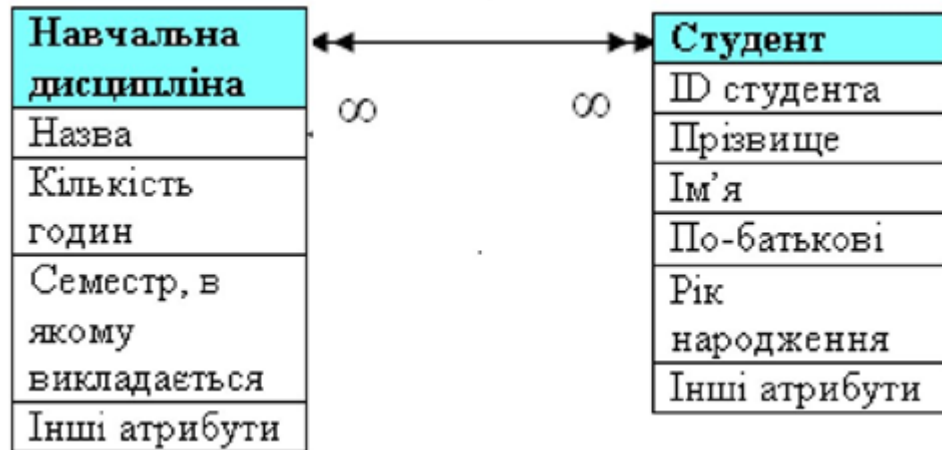
Зв'язок «один до багатьох»

Використовується для зв'язування одного запису в таблиці з декількома записами в іншій таблиці. Подібний зв'язок буде створено у тому випадку, коли одне з полів зв'язано з ключовим чи індексованим полем при умові відсутності повторювань



Зв'язок «багато до багатьох»

Такий зв'язок використовується досить рідко, оскільки для нього необхідна додаткова проміжна таблиця



Ключі

Ключ - це стовпець (може бути декілька стовпців), що додається до таблиці і дозволяє встановити зв'язок із записами в іншій таблиці

Кожна таблиця має містити поле, або набір полів, які задаватимуть унікальні значення кожному запису в таблиці, за яким можна точно визначити потрібний запис. Таке поле, або набір полів називається **ключем**

Код	Прізвище	Ім'я	Дата народ	Стать	Клас	Успішність
1	Іванов	Іван	12.12.2005	ч		
2	Петров	Василь	13.05.2004	ч		
3	Гриценко	Ліана	11.04.2004	ж		
4	Матвієнко	Марія	25.09.2006	ж		
5	Питренко	назар	16.01.2006	ч		
6	Гриб	Юлія	25.09.2005	ж		

Ключі

Існують **ключі двох типів**: первинні і вторинні (зовнішні)

Первинний ключ - це одне або кілька полів (стовпців), комбінація значень яких *однозначно визначає кожний запис у таблиці*. Первинний ключ не допускає значень Null і завжди повинен мати унікальний індекс. Первинний ключ використовується для зв'язування таблиці з зовнішніми ключами в інших таблицях.

Зовнішній (вторинний) ключ - це одне або кілька полів (стовпців) у таблиці, що містять посилання на поле або поля первинного ключа в іншій таблиці. Зовнішній ключ визначає спосіб об'єднання таблиць

Ключі

З двох логічно пов'язаних таблиць одну називають таблицею первинного ключа або **головною таблицею**, а іншу таблицею вторинного (зовнішнього) ключа або **підпорядкованої таблиці**