

Бази даних та SQL

Введення в SQL

Основні запити маніпуляції даними

Бази даних та SQL

План уроку

- Види баз даних
- Реляційна БД
- Система управління базами даних (СУБД)
- Створення базових запитів
 - SELECT
 - DISTINCT
 - LIMIT
 - WHERE
 - LIKE
 - ORDER BY

Бази даних та SQL

Бази даних та SQL

Бази даних та SQL

Що таке бази даних?

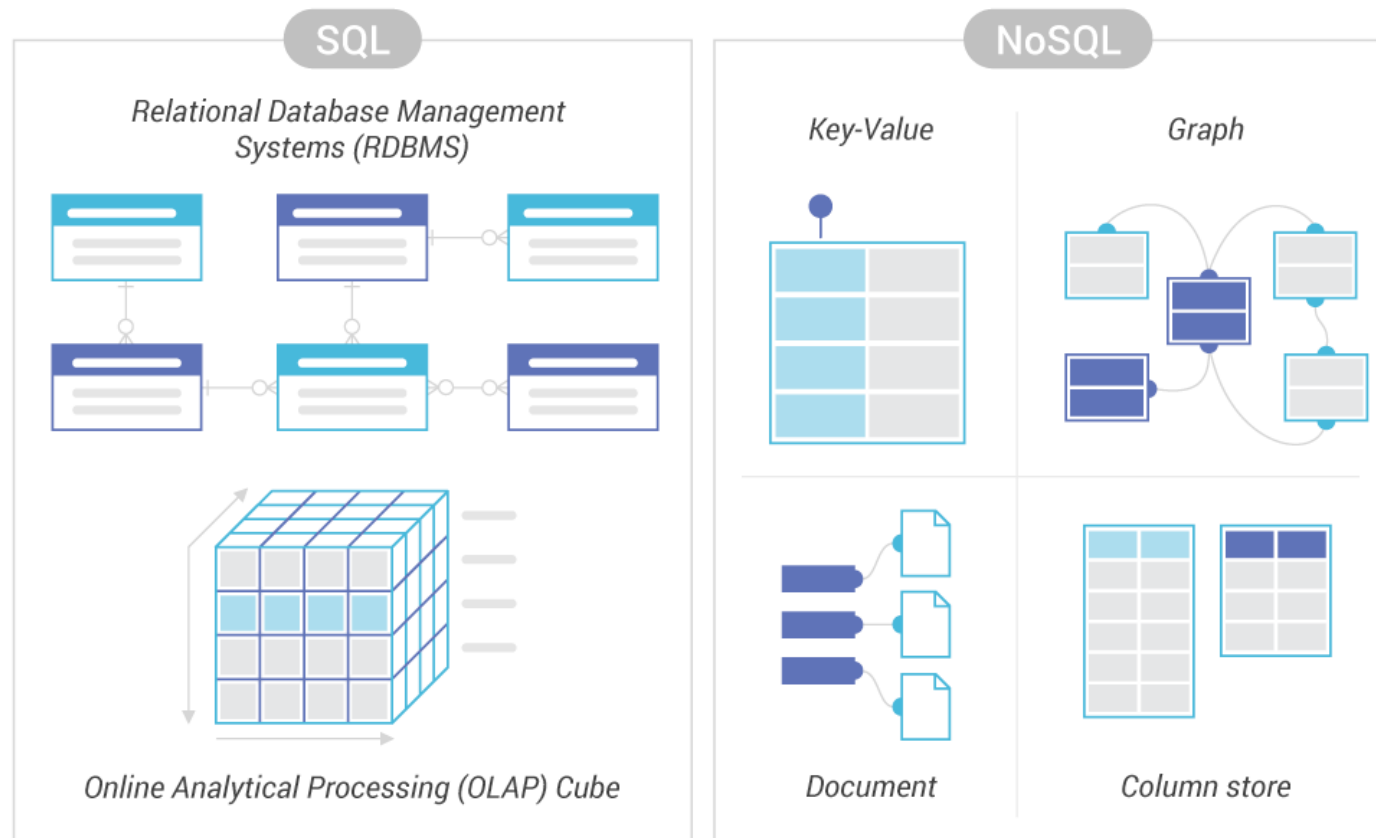


База даних – це сукупність пов'язаних даних, організованих за певними правилами.

Ці правила передбачають загальні принципи опису, маніпулювання та зберігання даними.

Бази даних та SQL

Види баз даних



Бази даних та SQL

Реляційні бази даних

Переваги

1. **Категоризація даних.**
Реляційні бази даних також легко розширювати і вони не залежать від фізичної організації. Після створення бази даних можна додати нову категорію даних без необхідності вносити зміни в існуючі додатки.
2. **Точність.**
Дані зберігаються лише один раз, що виключає дуплікацію даних у процедурах зберігання.
3. **Простота використання.**
Користувачі можуть легко виконувати складні запити за допомогою SQL, основної мови запитів, що використовується в реляційних базах даних.
4. **Безпека.**
Прямий доступ до даних у таблицях в RDBMS може бути обмежений для певних користувачів.

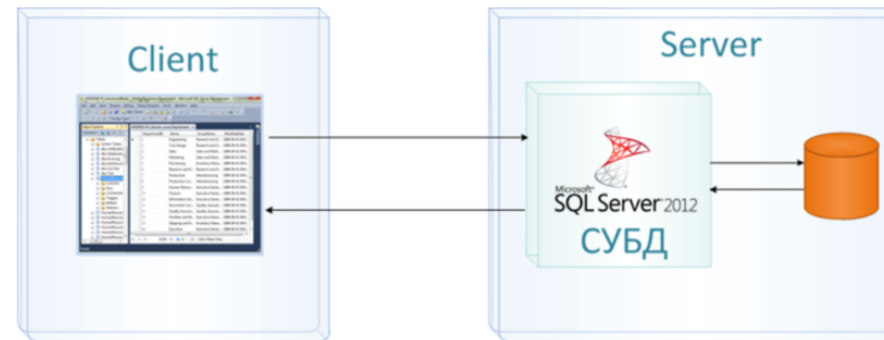
Недоліки

1. **Структура.** Реляційні бази даних вимагають багато структури і певного рівня планування, оскільки стовпці повинні бути визначені, а дані повинні правильно вписуватися в дещо жорсткі категорії.
2. **Складна підтримка.** Розробники та інший персонал, відповідальний за базу даних, повинні витрачати час на управління та оптимізацію бази даних по мірі додавання до неї даних.
3. **Відсутність масштабованості.** Реляційні бази даних погано масштабуються по горизонталі у фізичних структурах зберігання з декількома серверами. Працювати з реляційними базами даних на декількох серверах складно, оскільки зі збільшенням обсягу та розподіленості даних структура порушується, а використання декількох серверів впливає на продуктивність – наприклад, на час відгуку додатків – і доступність.

Бази даних та SQL

Система управління базами даних

СУБД – це комплекс мовних та програмних засобів, призначений для створення, ведення та спільного використання баз даних багатьма користувачами.

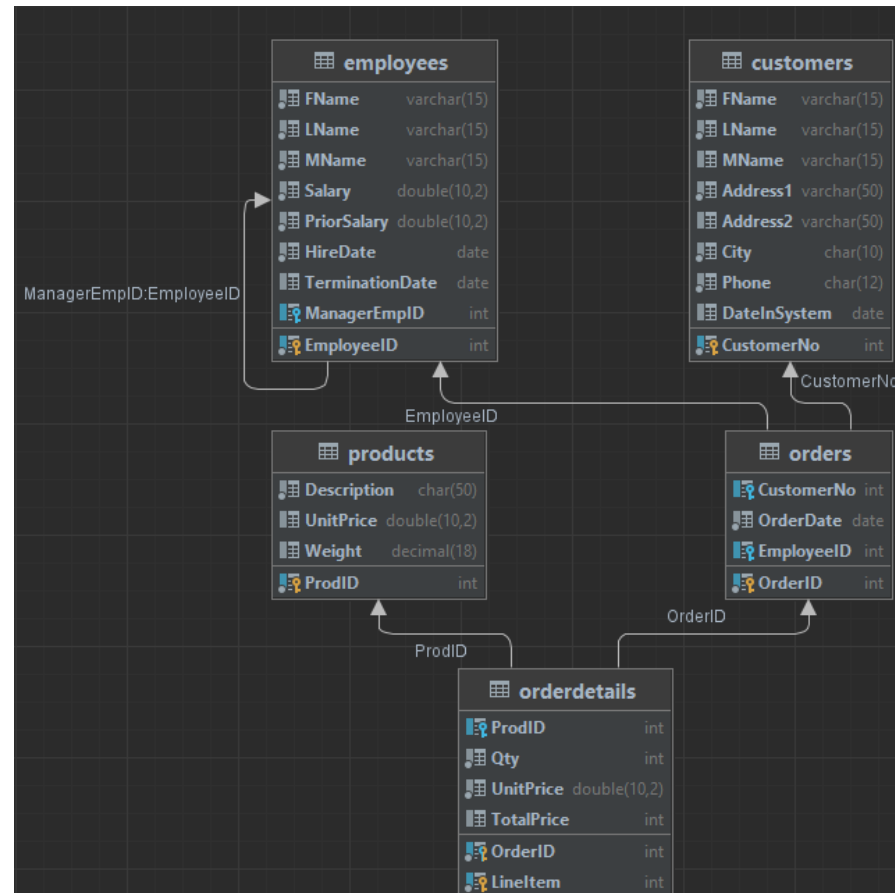


Зазвичай СУБД як і БД розрізняють по моделі даних. Так, СУБД, що ґрунтуються на використанні реляційної моделі даних, називають реляційними СУБД.

Бази даних та SQL

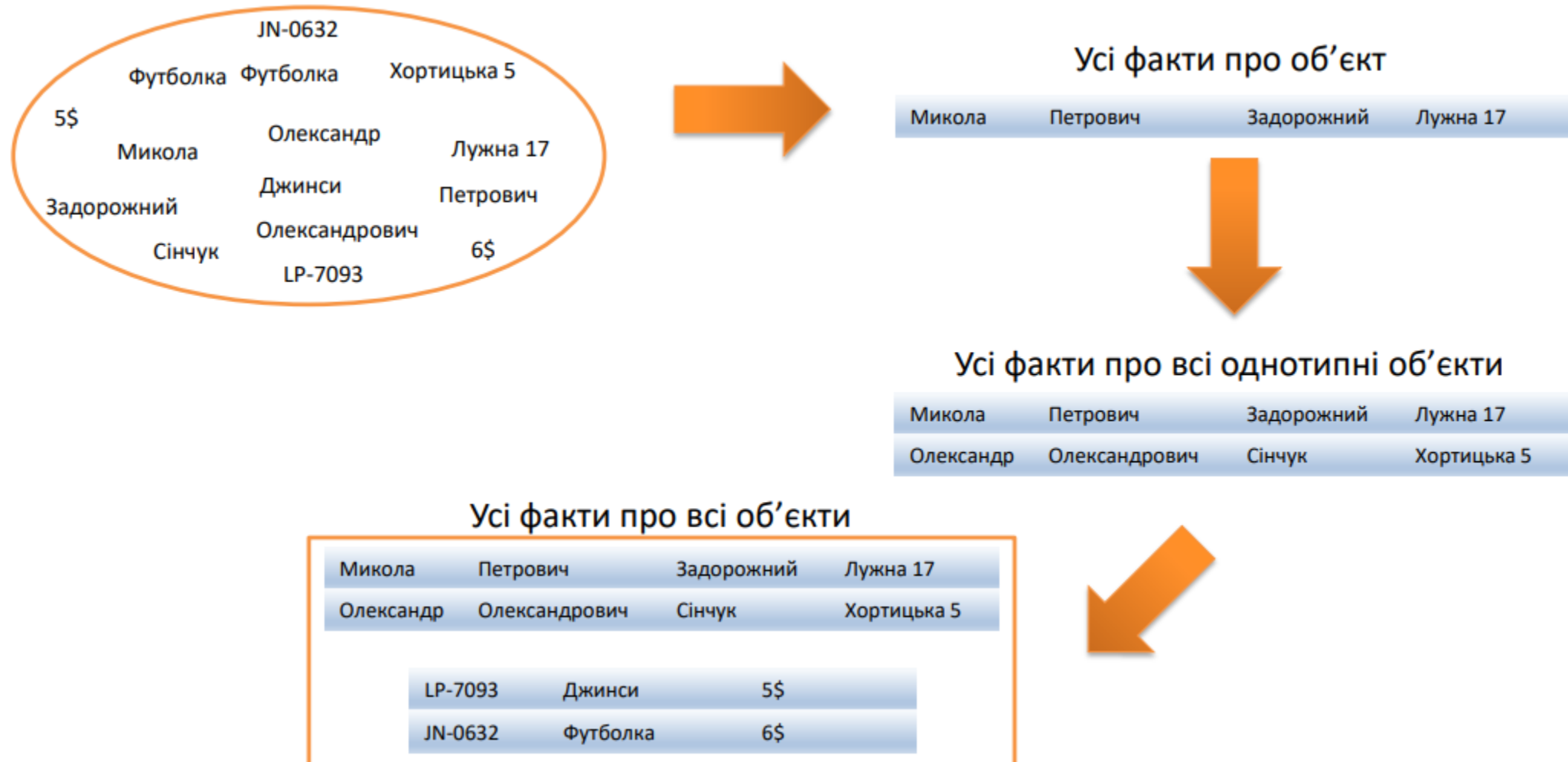
RDB (Relational DataBase)

Реляційна база даних – це сукупність пов'язаних даних, які у двомірних таблицях.



Бази даних та SQL

Структурування даних по таблицях



Бази даних та SQL

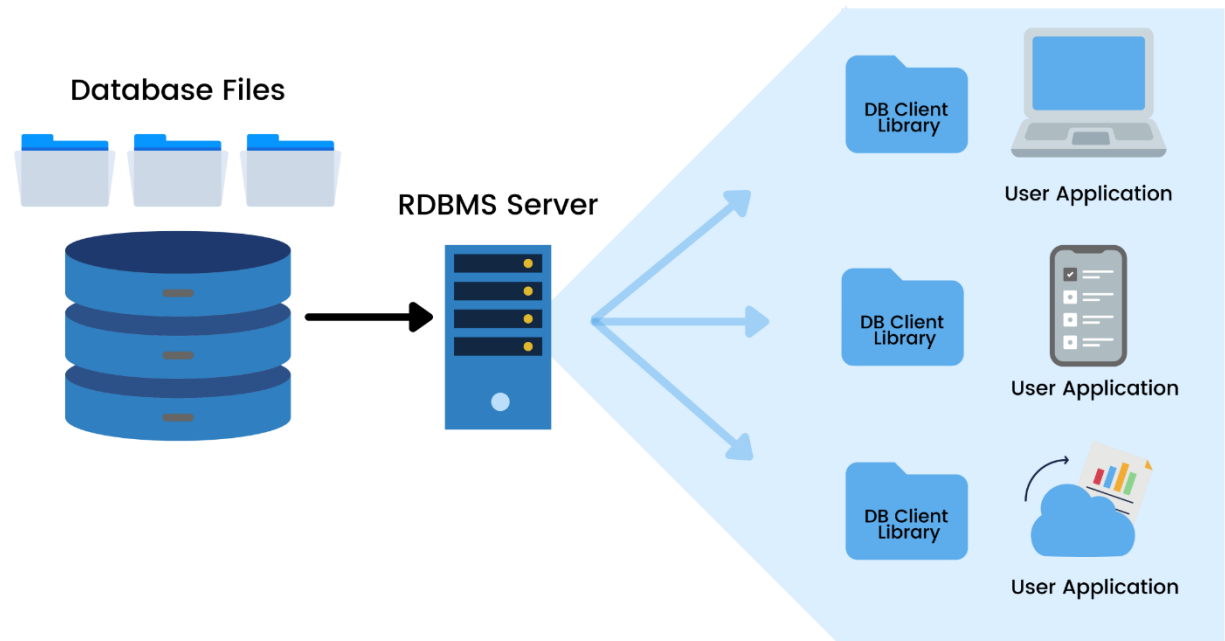
RDB (relational database)

RDBMS - Relational Database Management System.

RDBMS — це програма, яка використовується для підтримки реляційної бази даних.

RDBMS є основою для всіх сучасних систем баз даних, таких як MySQL, Microsoft SQL Server, Oracle і Microsoft Access.

RDBMS використовує запити SQL для доступу до даних у базі даних.



Бази даних та SQL

Таблиці

Таблиця — це сукупність пов'язаних записів даних, яка складається зі стовпців і рядків.

Стовпець містить конкретну інформацію про кожен запис у таблиці.

Запис (або рядок) — це кожен окремий запис, який існує в таблиці.

customer_id	first_name	last_name	phone	email	street	city	state	zip_code
1	Debra	Burks	NULL	debra.burks@yahoo.com	9273 Thome Ave.	Orchard Park	NY	14127
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street	Campbell	CA	95008
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.	Redondo Beach	CA	90278
4	Daryl	Spence	NULL	daryl.spence@aol.com	988 Pearl Lane	Uniondale	NY	11553
5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.	Sacramento	CA	95820
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road	Fairport	NY	14450
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.	Buffalo	NY	14215
8	Jacqueline	Duncan	NULL	jacqueline.duncan@yahoo.com	15 Brown St.	Jackson Heights	NY	11372
9	Genoveva	Baldwin	NULL	genoveva.baldwin@msn.com	8550 Spruce Drive	Port Washington	NY	11050
10	Pamelia	Newman	NULL	pamelia.newman@gmail.com	476 Chestnut Ave.	Monroe	NY	10950

Бази даних та SQL

Таблиці

Таблиці зберігають безпосередньо дані та стани полів та записів

Основні властивості поля таблиці:

1. Ім'я
2. Тип даних
3. Допустимість порожнього значення

customer_id	first_name	last_name	phone	email	street	city	state	zip_code
1	Debra	Burks	NULL	debra.burks@yahoo.com	9273 Thome Ave.	Orchard Park	NY	14127
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street	Campbell	CA	95008
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.	Redondo Beach	CA	90278
4	Daryl	Spence	NULL	daryl.spence@aol.com	988 Pearl Lane	Uniondale	NY	11553
5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.	Sacramento	CA	95820
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road	Fairport	NY	14450
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.	Buffalo	NY	14215
8	Jacquiline	Duncan	NULL	jacquiline.duncan@yahoo.com	15 Brown St.	Jackson Heights	NY	11372
9	Genoveva	Baldwin	NULL	genoveva.baldwin@msn.com	8550 Spruce Drive	Port Washington	NY	11050
10	Pamelia	Newman	NULL	pamelia.newman@gmail.com	476 Chestnut Ave.	Monroe	NY	10950

Бази даних та SQL

Structured Query Language

SQL – інформаційно-логічна мова, призначена для опису зміни та вилучення даних, що зберігаються в реляційних базах даних.

Основний набір операцій SQL

- Створення у базі даних нової таблиці;
- Додавання до таблиці нових записів;
- Зміна записів;
- Видалення записів;
- Вибір записів з однієї або декількох таблиць (відповідно до заданої умови);
- Зміна структури таблиць.

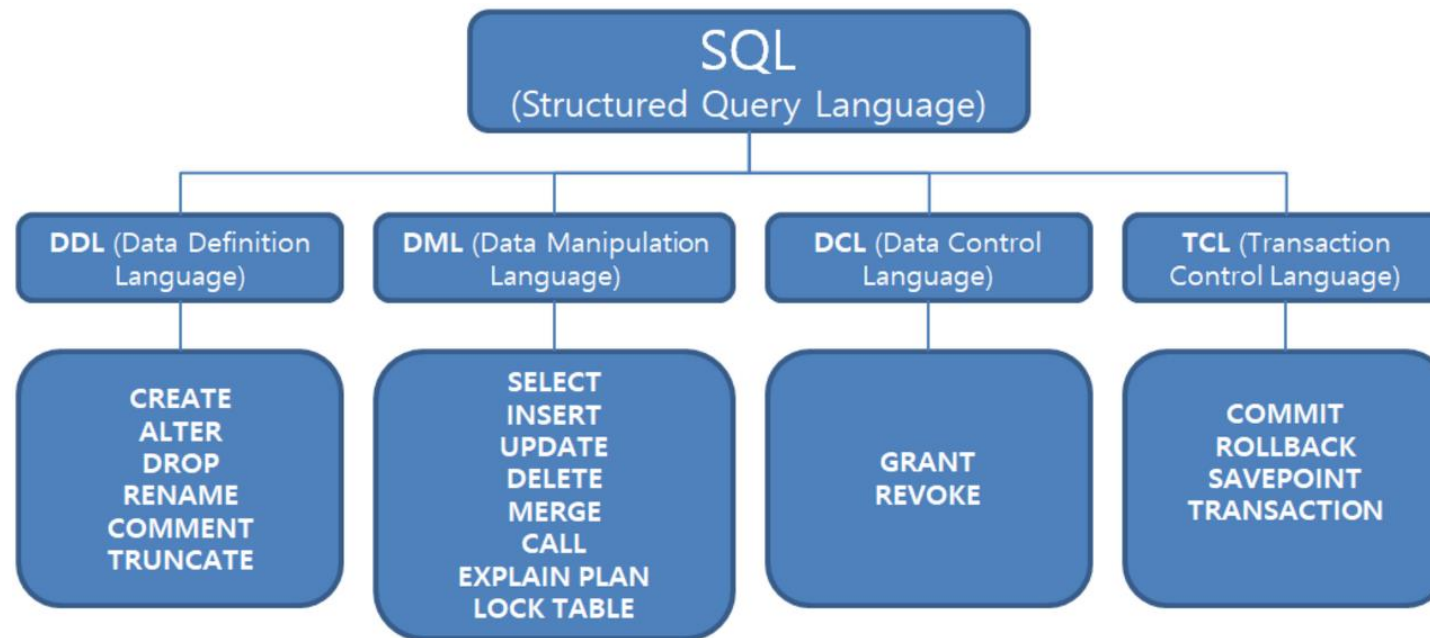
Бази даних та SQL

Типи даних MySQL

- Цілі числа **INT** – (15)
- Дійсні числа **FLOAT, DOUBLE** – (15.5)
- Рядки **CHAR, VARCHAR, TEXT** – ('a', 'Alex'),
- Дата та час **DATETIME** – (2023.08.16),
- Логічні дані **BOOL, TINYINT** – (true, false)

Бази даних та SQL

Мови у SQL



Data Manipulation Language (DML)

Бази даних та SQL

Data Manipulation Language

Data Manipulation Language (DML) – це сімейство комп'ютерних мов, що використовуються користувачами баз даних для отримання, вставки, видалення або зміни даних у базах даних.

Функції DML:

SELECT – вибірка даних

INSERT – додавання даних

UPDATE – оновлення даних

DELETE – видалення даних

Бази даних та SQL

SELECT

Оператор **SELECT** використовується для вибору даних із бази даних. Повернуті дані зберігаються в таблиці результатів, яка називається набором результатів.

SELECT *column1, column2, ...* -- це назви полів таблиці, з якої потрібно вибрати дані
FROM *table_name;* -- назва таблиці

7 • **SELECT** mark, model **FROM** cars;

mark	model
BMW	320d
Audi	A5
Porsche	911
BMW	X5
BMW	X7
BMW	M7
BMW	M7

Вибір полів mark model з таблиці

14 • **select** * **from** cars;

	id	mark_id	model	price
*	NULL	NULL	NULL	NULL

Вибір усіх полів таблиці



Бази даних та SQL


SELECT DISTINCT

Оператор **SELECT DISTINCT** використовується для повернення **лише** різних значень стовпців

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```


7 • **SELECT** mark, model **FROM** cars;

mark	model
BMW	320d
Audi	A5
Porsche	911
BMW	X5
BMW	X7
BMW	M7
BMW	M7



8 • **SELECT distinct** mark, model **FROM** cars;

mark	model
BMW	320d
Audi	A5
Porsche	911
BMW	X5
BMW	X7
BMW	M7



Бази даних та SQL

SELECT LIMIT

Оператор **LIMIT** використовується для визначення кількості записів, які потрібно повернути

Оператор LIMIT корисно для великих таблиць із тисячами записів. Повернення великої кількості записів може вплинути на продуктивність.

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```

```
36 • SELECT * FROM cars
37 WHERE mark LIKE 'BMW
38 limit 2;
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	1	BMW	320d	225	36000
	4	BMW	X5	225	36000
*	NULL	NULL	NULL	NULL	NULL



Фільтрація та організація набору результатів

Бази даних та SQL

WHERE

Твердження **WHERE** використовується для фільтрації записів.
Воно використовується для вилучення лише тих записів, які відповідають певній умові.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Приклад:

Наступний оператор SQL вибирає всіх клієнтів із "Мексики":

```
SELECT * FROM Customers  
WHERE Country = 'Mexico';
```



Зауважте, твердження **WHERE** також використовується в операторах **UPDATE**, **DELETE**, тощо.

Бази даних та SQL

Текстові та числові поля

Синтаксис SQL вимагає:

- Для текстових значень - **одинарних лапок** (більшість систем баз даних також допускають подвійні лапки).
- Для числових полів – **ніяких додаткових символів**

```
36 • SELECT * FROM cars
37 WHERE mark = 'Audi';
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	2	Audi	A5	250	50000
*	NULL	NULL	NULL	NULL	NULL

```
36 • SELECT * FROM cars
37 WHERE car_id = 3;
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	3	Porsche	911	285	100000
*	NULL	NULL	NULL	NULL	NULL



Бази даних та SQL

Оператори порівнянь

У твердженні WHERE можна використовувати такі оператори порівнянь:

- = Рівно
- > Більше ніж
- < Менше ніж
- >= Більше або дорівнює
- <= Менше або дорівнює
- <> Не дорівнює. У деяких версіях SQL використовується інший оператор !=
- BETWEEN Між певним діапазоном
- LIKE Пошук шаблону
- IN Для визначення кількох можливих значень для стовпця

```
36 • SELECT * FROM cars
37 WHERE car_id > 3;
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	4	BMW	X5	225	36000
	5	BMW	X7	225	36000
	6	BMW	M7	225	36000
	7	BMW	M7	225	36000
*	NULL	NULL	NULL	NULL	NULL

```
36 • SELECT * FROM cars
37 WHERE car_id BETWEEN 1 AND 3;
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	1	BMW	320d	225	36000
	2	Audi	A5	250	50000
	3	Porsche	911	285	100000
*	NULL	NULL	NULL	NULL	NULL



Бази даних та SQL

LIKE

Оператор **LIKE** використовується в твердженні WHERE для пошуку заданого шаблону в стовпці.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE pattern;
```

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'Alfreds Futterkiste';
```

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT *
FROM Customers
WHERE country **LIKE** 'UK';

customer_id	first_name	last_name	age	country
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

Бази даних та SQL

LIKE

У шаблонах оператора LIKE є два основних символи підстановки:

- Відсоток (%) – позначає будь-яку кількість символів
- Підкреслення (_) – позначає один символ

```
36 • SELECT * FROM cars
37 WHERE mark LIKE '%i';
```

	car_id	mark	model	speed	price
▶	2	Audi	A5	250	50000
*	NULL	NULL	NULL	NULL	NULL

```
36 • SELECT * FROM cars
37 WHERE mark LIKE 'B%';
```

	car_id	mark	model	speed	price
▶	1	BMW	320d	225	36000
	4	BMW	X5	225	36000
	5	BMW	X7	225	36000
	6	BMW	M7	225	36000
	7	BMW	M7	225	36000
*	NULL	NULL	NULL	NULL	NULL

```
36 • SELECT * FROM cars
37 WHERE mark LIKE '%o%';
```

	car_id	mark	model	speed	price
▶	3	Porsche	911	285	100000
*	NULL	NULL	NULL	NULL	NULL



Порада. Комбінуйте будь-яку кількість умов за через оператори AND та OR.



Бази даних та SQL

Вправи по LIKE

Що повернуть наступні запити? Поясніть їх ідею

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'a%';
```

Запит вибирає всіх клієнтів із CustomerName, що починається на "a"

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'a%o';
```

Запит вибирає всіх клієнтів із CustomerName, що починається з "a" та закінчується на "o"

```
SELECT * FROM Customers  
WHERE CustomerName LIKE '%or%';
```

Запит вибирає всіх клієнтів із CustomerName, які мають «or» в будь-якій позиції

```
SELECT * FROM Customers  
WHERE CustomerName LIKE 'a__%';
```

Запит вибирає всіх клієнтів із CustomerName, які починаються з "a" і містять принаймні 3 символи

Бази даних та SQL

ORDER BY

Ключове слово **ORDER BY** використовується для сортування набору результатів

За замовчуванням сортування йде в порядку зростання (**ASC**)

Щоб відсортувати в порядку спадання, використовуйте ключове слово **DESC**.

ORDER BY Синтаксис

SELECT *column1, column2, ...*

FROM *table_name*

ORDER BY *column1, column2, ... ASC|DESC;*

```
36 • SELECT * FROM cars
37   order by car_id asc;
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	1	BMW	320d	225	36000
	2	Audi	A5	250	50000
	3	Porsche	911	285	100000
	4	BMW	X5	225	36000
	5	BMW	X7	225	36000
	6	BMW	M7	225	36000
	7	BMW	M7	225	36000
*	NULL	NULL	NULL	NULL	NULL

```
36 • SELECT * FROM cars
37   order by car_id desc;
```

Result Grid					
Filter Rows:					
	car_id	mark	model	speed	price
▶	7	BMW	M7	225	36000
	6	BMW	M7	225	36000
	5	BMW	X7	225	36000
	4	BMW	X5	225	36000
	3	Porsche	911	285	100000
	2	Audi	A5	250	50000
	1	BMW	320d	225	36000
*	NULL	NULL	NULL	NULL	NULL



Бази даних та SQL

ORDER BY за кількома стовпцями

Наступний оператор SQL вибирає всіх клієнтів із таблиці «Клієнти», відсортованих за стовпцями «Country» та «CustomerName».

```
SELECT * FROM Customers  
ORDER BY Country, CustomerName;
```

Це означає, що він упорядковує їх за країною, але якщо деякі рядки мають **однакову країну**, вони впорядковуються за назвою клієнта:



Бази даних та SQL

Вправи по ORDER BY

Що повернуть наступні запити? Поясніть їх ідею

```
SELECT * FROM Customers  
ORDER BY Country DESC
```

Запит вибирає всіх клієнтів із таблиці «Клієнти», відсортованих за спаданням за стовпцем «Country»

```
SELECT * FROM Customers  
ORDER BY Country
```

Запит вибирає всіх клієнтів із таблиці «Клієнти», відсортованих за зростанням за стовпцем «Country»

```
SELECT * FROM Customers  
ORDER BY Country ASC,  
CustomerName DESC
```

Запит вибирає всіх клієнтів із таблиці «Клієнти», відсортованих за зростанням за стовпцем «Country» та за спаданням за стовпцем «CustomerName»:

Manual QA

Що ми сьогодні вивчили

Бази даних та SQL

План уроку

- Види баз даних
- Реляційна БД
- Система управління базами даних (СУБД)
- Створення базових запитів
 - SELECT
 - DISTINCT
 - LIMIT
 - WHERE
 - LIKE
 - ORDER BY

Manual QA

Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

Manual QA

Дякую за увагу! До нових зустрічей!

Інформаційний відеосервіс для розробників програмного забезпечення

