**Structured Query Language**

**SQL**

***SQL*** - це мова запитів до баз даних, яка використовується для створення, зміни

та запитів даних з різних реляційних баз даних.

***SQL*** є стандартом ANSI (American National Standards Institute), тобто мова запитів до баз даних, яка використовується у багатьох різних реляційних базах даних.

***SQL*** дозволяє виконувати з а пити для створення, зміни та видалення таблиць та їх структури, встановлення і зміну обмежень доступу, додавання, видалення та редагування даних у таблицях, зміну і запити даних з однієї або декількох таблиць одночасно, створення та використання індексів для покращення швидкості

запитів, а також виконання інших складних операцій.

***SQL*** є основною мовою запитів для багатьох реляційних баз даних, таких як ***MySQL***, ***Oracle*** , ***Microsoft SQL Server***, ***PostgreSQL***, та інших.

***Структура SQL***

***DML (Data Manipulation Language)*** та ***DDL (Data Definition Language)*** - це дві основні

групи команд ***SQL***, які використовуються для маніпулювання та визначення структури даних в реляційній базі даних.

***DML*** використовується для маніпулювання даними в таблицях, таких як вставка, оновлення та видалення даних. Деякі з команд ***DML*** включа ють ***SELECT***, ***INSERT***, ***UPDATE***, ***DELETE*** та інші. ***DDL*** використовується для визначення структури даних, таких як створення та видалення таблиць, визначення обмежень та індексів. Деякі з команд ***DDL*** включають ***CREATE***, ***ALTER***, ***DROP*** та інші.

***TCL (Transaction Control Language)*** та **DCL (Data Control Language )** - це дві додаткові групи команд SQL, які використовуються для керування транзакціями та правами доступу до даних в реляційній базі даних.

***TCL*** використовується для ***керування транзакціями***, які є групою операцій з базою даних, які повинні бути виконані як одне ціле . Команди ***TCL*** дозволяють контролювати

транзакції, такі як ***COMMIT*** для збереження змін у базі даних та ***ROLLBACK*** для відкату змін.

***DCL*** використовується для ***керування правами доступу до даних***. Команди ***DCL*** дозволяють встановлювати та змінювати права доступу користувачів до таблиць та інших об'єктів бази даних. Деякі з команд ***DCL*** включають ***GRANT*** для надання прав доступу та ***REVOKE*** для відкликання прав доступу.

***Деякі з найважливіших команд SQL***

* SELECT - витягує дані з бази даних
* UPDATE - оновлює дані в базі даних
* DELETE - видаляє дані з бази даних
* INSERT INTO - вставляє нові дані в базу даних
* CREATE DATABASE - створює нову базу даних
* ALTER DATABASE - модифікує базу даних
* CREATE TABLE - створює нову таблицю
* ALTER TABLE - модифікує таблицю
* DROP TABLE - видаляє таблицю
* CREATE INDEX - створює індекс (ключ пошуку)
* DROP INDEX - видаляє індекс

**DML Запити**

***- SELECT;***

***- INSERT;***

***- UPDATE;***

***- DELETE.***

### **SELECT Syntax**

Оператор ***SELECT*** використовується для вибору даних з бази даних.

Повернуті дані зберігаються в таблиці результатів, яка називається ***result-set***.

SELECT *column1*, *column2, ...*

FROM *table\_name*;

Тут ***column1, column2,*** ... - це назви полів таблиці, з якої ви хочете вибрати дані. Якщо ви хочете вибрати всі поля, доступні в таблиці, використовуйте наступний синтаксис:

SELECT \* FROM *table\_name*;

### **UPDATE Syntax**

Оператор ***UPDATE*** використовується для зміни існуючих записів у таблиці.

UPDATE *table\_name*

SET *column1* = *value1*, *column2* = *value2*, ...

WHERE *condition*;

Будьте обережні при оновленні записів у таблиці! Зверніть увагу на речення ***WHERE*** в інструкції ***UPDATE***. Речення ***WHERE*** вказує, який запис (записи) слід оновити. Якщо ви опустите речення ***WHERE***, будуть оновлені всі записи в таблиці!

### **DELETE Syntax**

Оператор ***DELETE*** використовується для видалення існуючих записів у таблиці.

DELETE FROM *table\_name* WHERE *condition*;

Будьте обережні при видаленні записів у таблиці! Зверніть увагу на речення ***WHERE*** в інструкції ***DELETE***. Речення ***WHERE*** вказує, який запис (записи) слід видалити. Якщо ви опустите оператор ***WHERE***, то буде видалено всі записи в таблиці!

### **INSERT INTO Syntax**

Оператор ***INSERT INTO*** використовується для вставки нових записів у таблицю.

Оператор ***INSERT INTO*** можна записати двома способами:

1. Вказати як імена стовпців, так і значення, що вставляються:

***INSERT INTO table\_name (column1, column2, column3, ...)***

***VALUES (value1, value2, value3, ...);***

2. Якщо ви додаєте значення для всіх стовпців таблиці, вам не потрібно вказувати імена стовпців у SQL-запиті. Однак переконайтеся, що порядок значень відповідає порядку стовпців у таблиці. У цьому випадку синтаксис INSERT INTO буде наступним:

***INSERT INTO table\_name***

***VALUES (value1, value2, value3, ...);***

**More SQL Syntaxes**

### 

### **WHERE Syntax**

Речення ***WHERE*** використовується для фільтрації записів.

З його допомогою витягуються тільки ті записи, які задовольняють заданій умові.

*SELECT column1, column2, ...*

*FROM table\_name*

*WHERE condition;*

Речення ***WHERE*** використовується не тільки в операторах ***SELECT***, але й в операторах ***UPDATE***, ***DELETE*** і т.д.!

В операторі WHERE можна використовувати наступні операції:

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| = | Equal |  |
| > | Greater than |  |
| < | Less than |  |
| >= | Greater than or equal |  |
| <= | Less than or equal |  |
| <> | Not equal. **Note:** In some versions of SQL this operator may be written as != |  |
| BETWEEN | Between a certain range |  |
| LIKE | Search for a pattern |  |
| IN | To specify multiple possible values for a column |  |

## 

## **AND, OR and NOT Operators**

Речення ***WHERE*** можна комбінувати з операторами ***AND, OR*** та ***NOT***.

Оператори ***AND*** та ***OR*** використовуються для фільтрації записів на основі більш ніж однієї умови:

Оператор ***AND*** відображає запис, якщо всі умови, розділені символом ***AND***, мають значення ***TRUE***.

Оператор ***OR*** відображає запис, якщо будь-яка з умов, розділених символом ***OR***, є ***TRUE***.

Оператор ***NOT*** відображає запис, якщо умова(и) не є ***TRUE***.

### **AND Syntax**

SELECT *column1*, *column2, ...*

FROM *table\_name*

WHERE *condition1* AND *condition2* AND *condition3 ...*;

### 

### **OR Syntax**

SELECT *column1*, *column2, ...*

FROM *table\_name*

WHERE *condition1* OR *condition2* OR *condition3 ...*;

### 

### **NOT Syntax**

SELECT *column1*, *column2, ...*

FROM *table\_name*

WHERE NOT *condition*;

### **ORDER BY Syntax**

Ключове слово ***ORDER BY*** використовується для сортування набору результатів за зростанням або спаданням.

За замовчуванням ключове слово ***ORDER BY*** сортує записи за зростанням. Щоб відсортувати записи за спаданням, використовуйте ключове слово ***DESC***.

SELECT *column1*, *column2, ...*

FROM *table\_name*

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

### **INSERT INTO SELECT Syntax**

Оператор ***INSERT INTO SELECT*** копіює дані з однієї таблиці і вставляє їх в іншу таблицю.

Оператор ***INSERT INTO SELECT*** вимагає, щоб типи даних у вихідній і цільовій таблицях збігалися. Існуючі записи в цільовій таблиці не змінюються.

***Скопіюйте всі стовпці з однієї таблиці в іншу:***

INSERT INTO *table2*

SELECT \* FROM *table1*

WHERE *condition*;

***Скопіювати лише деякі стовпці з однієї таблиці в іншу:***

INSERT INTO *table2* (*column1*, *column2*, *column3*, ...)

SELECT *column1*, *column2*, *column3*, ...

FROM *table1*

WHERE *condition*;

### **GROUP BY Syntax**

Оператор ***GROUP BY*** групує рядки з однаковими значеннями в підсумкові рядки, наприклад, "знайти кількість клієнтів у кожній країні".

Оператор ***GROUP BY*** часто використовується з агрегатними функціями ***(COUNT(), MAX(), MIN(), SUM(), AVG())*** для групування набору результатів за одним або кількома стовпчиками.

SELECT *column\_name(s)*

FROM *table\_name*

WHERE *condition*

GROUP BY *column\_name(s)*

ORDER BY *column\_name(s);*