



Structured Query Language

SQL (Structured Query Language) is a standard language used to manage and manipulate relational databases. It allows users to create, retrieve, update, and delete data efficiently. SQL is widely used in database systems such as MySQL, Oracle, and SQL Server. There are two main ways to use SQL: Interactive SQL, where users directly execute commands, and Embedded SQL, where SQL is integrated into programming languages. Understanding SQL is essential for database design and application development.



MySQL

Source: Iamsiriwong, O. (2008). Database systems. Bangkok: SE-ED Education.



Structured Query Language

Interactive SQL

Interactive SQL allows users to directly execute SQL commands through a user interface. It is commonly used for querying and analyzing data. For example:

```
//  
SELECT *  
FROM Student  
WHERE branchNO = 'com001';  
_
```



Structured Query Language

Embedded SQL

Embedded SQL refers to SQL statements written inside a programming language. It improves performance and flexibility. There are two approaches

- Application Programming Interface (API)

- Embedded SQL Statements



Structured Query Language

Types of SQL Commands

SQL commands are divided into three main categories:

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Data Control Language (DCL)



Structured Query Language

Types of SQL Commands

Data Definition Language (DDL)

It is a set of commands used to define, specify, or create data in a database.

CREATE TABLE, CREATE VIEW



Structured Query Language

Types of SQL Commands

Data Manipulation Language (DML)

"It is a set of commands used to process or manipulate data in a database, including commands such as SELECT, INSERT, and UPDATE."

CREATE TABLE, CREATE VIEW
CREATE TABLE, CREATE VIEW

SELECT, INSERT, UPDATE

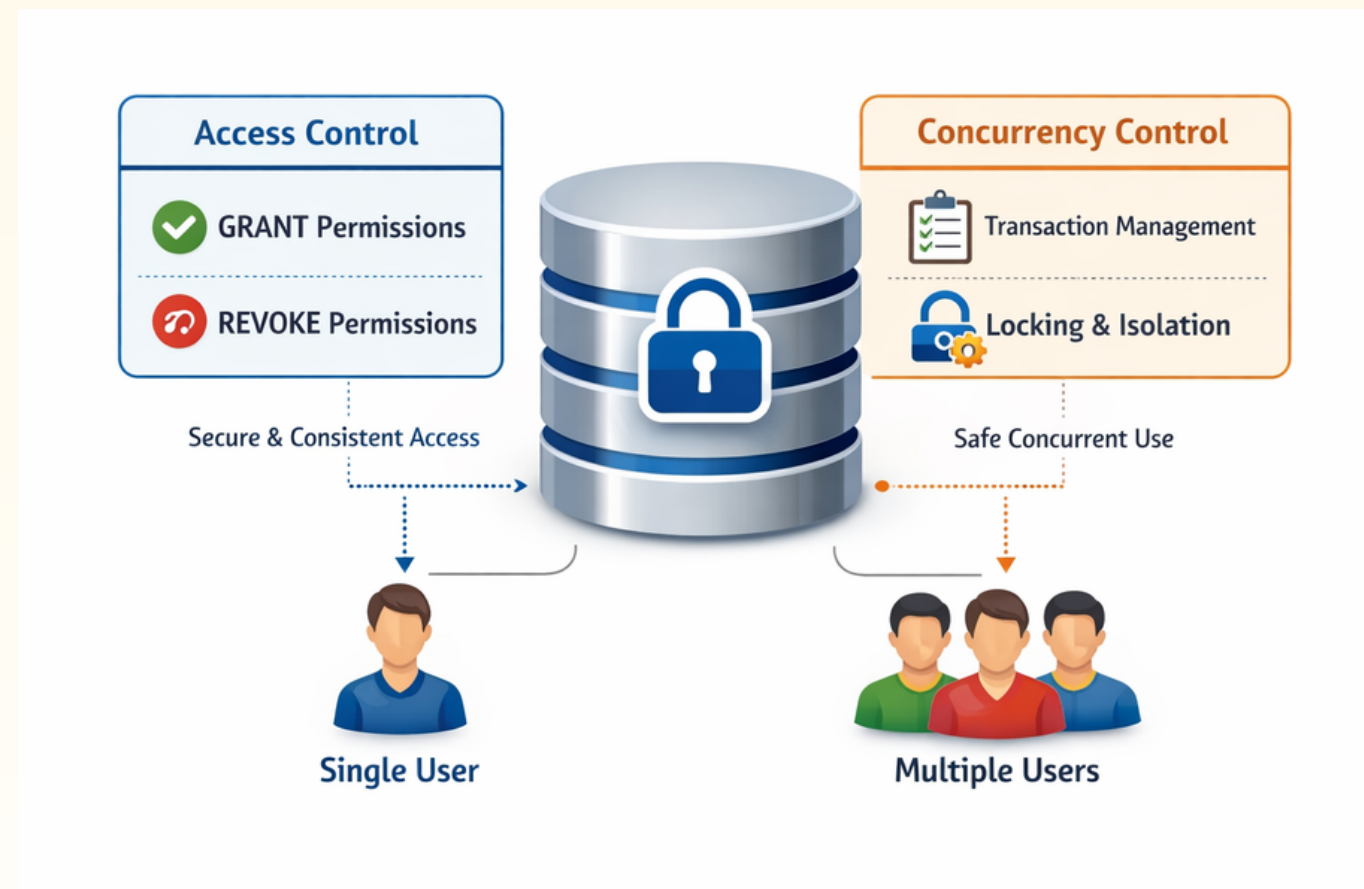


Structured Query Language

Types of SQL Commands

Data Control Language (DCL)

It is a set of commands used to control user access permissions to data, as well as to manage concurrent use of the database by multiple users.





Structured Query Language

ER Model – Bookstore System

The ER Model represents a bookstore system with three tables

CREATE TABLE:

Customers, Books, Orders

Relationships:

One-to-Many (1:M) and Many-to-Many.(M:N)



Structured Query Language

CREATE DATABASE

Database creation command

```
CREATE DATABASE bookstore_db;
```

Command terminator

Command to create a new database

Database name

Database selection command

```
USE bookstore_db;
```

Name of the created database

Command terminator

Command to tell MySQL to use this database



Structured Query Language

CREATE TABLE customers

```
CREATE TABLE customers(  
customer_in INT AUTO_INCREMENT PRIMARY KEY,  
fullname VARCHAR(100) NOT NULL,  
email VARCHAR(100) UNIQUE,  
phone VARCHAR(100));
```



Structured Query Language

CREATE TABLE books

```
CREATE TABLE books (  
  book_id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(150) NOT NULL,  
  author VARCHAR(100),  
  price DECIMAL(10,2) NOT NULL);
```



Structured Query Language

CREATE TABLE orders

```
CREATE TABLE orders (  
  order_id INT AUTO_INCREMENT PRIMARY KEY,  
  customer_id INT NOT NULL,  
  book_id INT NOT NULL,  
  order_date DATE NOT NULL,  
  quantity INT NOT NULL,
```



Structured Query Language

Table Relationship

Define a constraint name
for the foreign key.

A user-defined name to
indicate that this foreign
key is linked to the
customers table.

Declare that this
column will be a key

This is the column in this table
(orders) that will be linked to
another table.

Specifies which table
this column references.

The customer_id column in the customers table.

Specifies that if the customer_id value in the
customers table is updated, the corresponding
customer_id values in the orders table will be
automatically updated as well.

```
CONSTRAINT fk_customer FOREIGN KEY (customer_id)
REFERENCES customers(customer_id)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CONSTRAINT fk_book FOREIGN KEY (book_id)
REFERENCES books(book_id)
ON UPDATE CASCADE ON DELETE CASCADE
```



Structured Query Language

Insert

-- Add a customer.

```
INSERT INTO customers (fullname, email, phone) VALUES  
( 'aungpao', 'somchai@example.com', '0812345678'),  
( 'wipa', 'wipa@example.com', '0898765432');
```

INSERT INTO — A command that tells the system to “add a new row into a table.”

customers — The name of the table where the data will be inserted (in this case, the customers table).

(fullname, email, phone) — The list of columns where values will be inserted (the order is important — each value set must follow this order).

VALUES — Followed by sets of values (value tuples) to be inserted:

```
( 'Somchai Jaidee', 'somchai@example.com', '0812345678'),  
( 'Wipa Srisuk', 'wipa@example.com', '0898765432');
```

-- Add books

```
INSERT INTO books (title, author, price) VALUES  
( 'Learning MySQL', 'Ajarn Ekachai', 350.00),  
( 'Python Programming', 'Dr. Sunee', 450.00);
```

Each (...) represents one row that will be inserted.

First row: fullname = 'Somchai Jaidee', email = 'somchai@example.com', phone = '0812345678'.

Second row: fullname = 'Wipa Srisuk', email = 'wipa@example.com', phone = '0898765432'.

Text values must be enclosed in single quotes '... '.

Rows are separated by commas (,) — for multiple rows, they can be written in this format (multiple-row insert).

The statement ends with a semicolon (;) — indicating the end of the SQL command.



Structured Query Language

Insert

Insert orders

```
INSERT INTO orders (customer_id, book_id, order_date, quantity)
VALUES (1, 1, '2025-09-18', 2),
(1, 2, '2025-09-18', 1),
(2, 2, '2025-09-18', 3);
```

Each set of values in order:

The first value = customer_id (must exist in the customers table).

The second value = book_id (must exist in the books table).

The third value = order_date ('YYYY-MM-DD' — a date format accepted by the DATE type).

The fourth value = quantity (must be an integer).

Example of the first row: (1, 1, '2025-09-18', 2) = Customer ID 1 orders Book ID 1, quantity 2, on 2025-09-18.

Important: If the orders table defines order_id as AUTO_INCREMENT and it is not included in the column list, MySQL will automatically generate the order_id value (e.g., 1, 2, 3).



Structured Query Language

Update

1. Select the 'customers' table from the database.

2. Click the 'SQL' tab to enter the query editor.

3. Write the UPDATE query:

```
UPDATE `customers` SET `customer_id`='7777',`fullname`= 'aungpao jaidee',`email`='somchai@example.com',`phone`='1234567890' WHERE customer_id=16666;
```

4. Click the 'Go' button to execute the query.

5. The query is executed successfully.

6. The result shows '1 row affected. (Query took 0.0024 seconds.)'

7. The query is saved.



Structured Query Language

Update

The name of the table to be updated.

Columns and data to be updated.

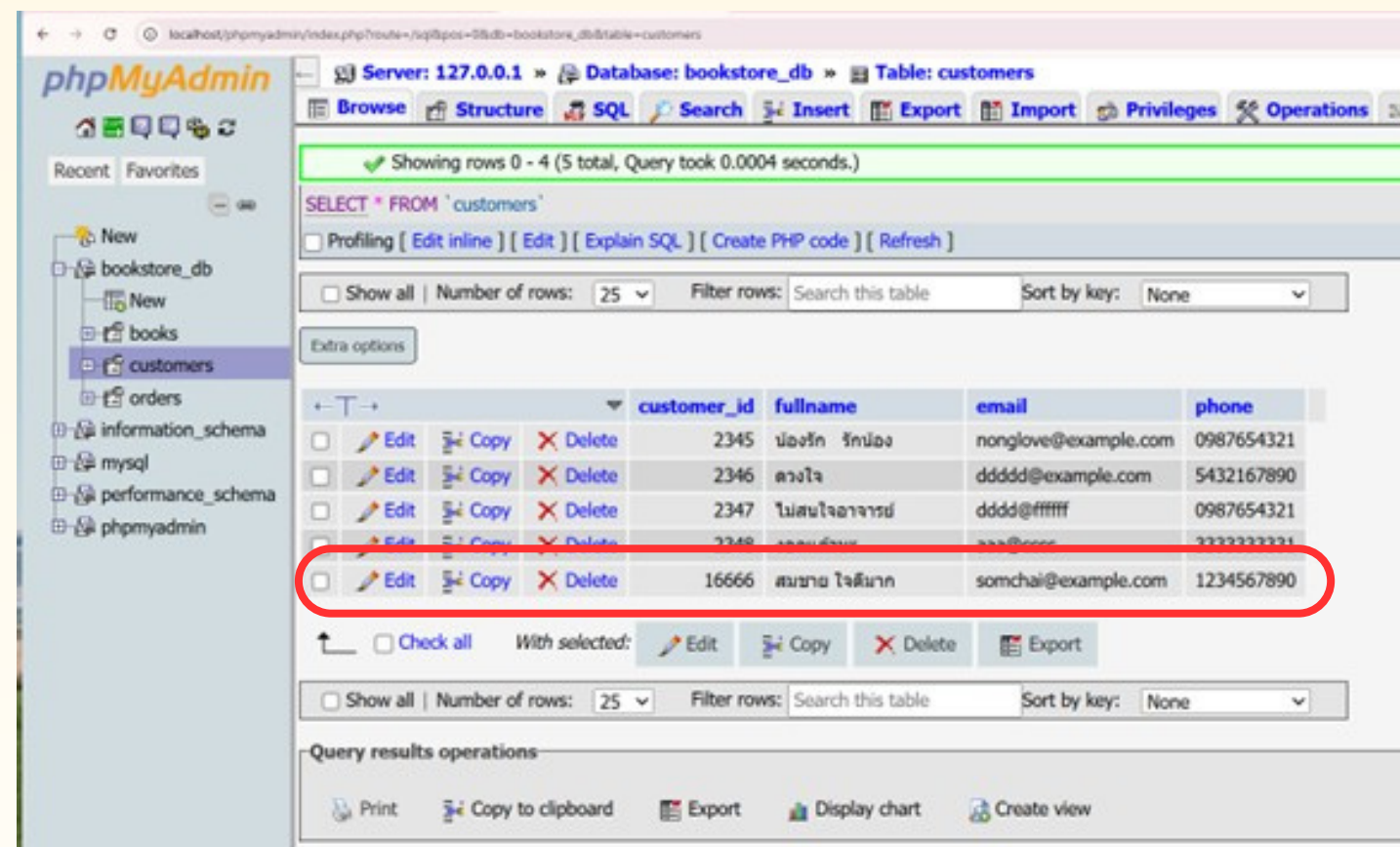
```
UPDATE `customers` SET `customer_id`='7777',`fullname`='somchai mak  
,`email`='somchai@example.com',`phone`='1234567890' WHERE customer_id=16666;
```

Columns and existing data



Structured Query Language

Update



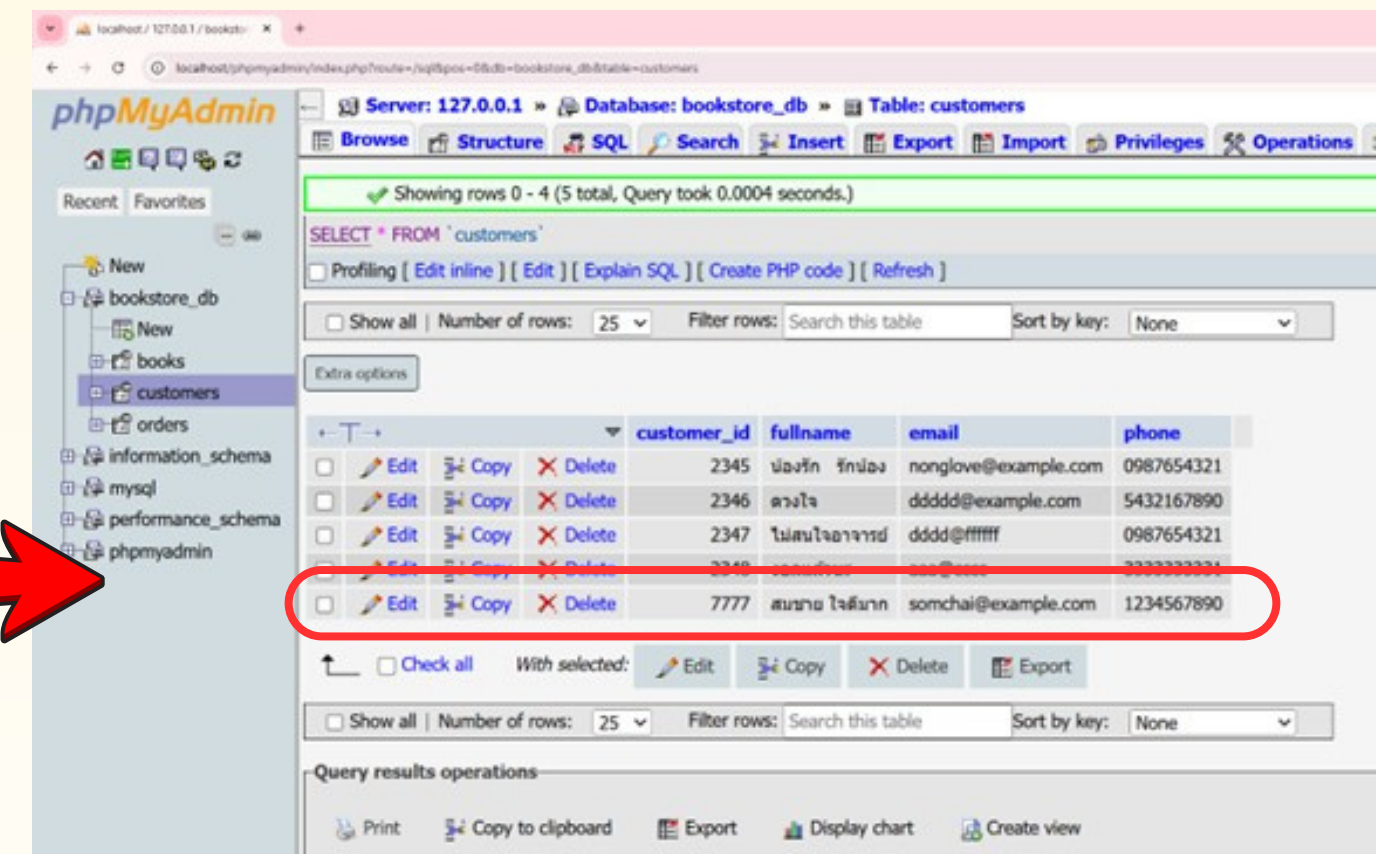
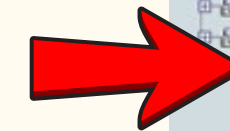
Server: 127.0.0.1 » Database: bookstore_db » Table: customers

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

SELECT * FROM `customers`

Number of rows: 25 Filter rows: Search this table Sort by key: None

	customer_id	fullname	email	phone
☐	2345	นงนุช รัตนวง	nonglove@example.com	0987654321
☐	2346	ดวงใจ	dddd@example.com	5432167890
☐	2347	ไม้มโนจาจรรณ	dddd@ffffff	0987654321
☐	2348	สมชาย ใจดี	somchai@example.com	1234567890
☐	16666	สมชาย ใจดี	somchai@example.com	1234567890



Server: 127.0.0.1 » Database: bookstore_db » Table: customers

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

SELECT * FROM `customers`

Number of rows: 25 Filter rows: Search this table Sort by key: None

	customer_id	fullname	email	phone
☐	2345	นงนุช รัตนวง	nonglove@example.com	0987654321
☐	2346	ดวงใจ	dddd@example.com	5432167890
☐	2347	ไม้มโนจาจรรณ	dddd@ffffff	0987654321
☐	2348	สมชาย ใจดี	somchai@example.com	1234567890
☐	7777	สมชาย ใจดี	somchai@example.com	1234567890



Structured Query Language

Delete

The screenshot shows the phpMyAdmin interface for the 'bookstore_db' database. The 'customers' table is selected in the left sidebar (1). The 'SQL' tab is active (2). The query 'DELETE FROM customers WHERE customer_id = 2346;' is entered in the query box (4). The 'DELETE' button is highlighted (3). The 'Go' button is highlighted (5). The query execution result shows the table structure with columns: customer_id, fullname, email, and phone.

```
DELETE FROM customers WHERE customer_id = 2346;
```