Lesson 1: Introduction to Databases and SQL

Introduction:
In today's data-driven world, understanding how to store, access, and manipulate data is a key skill for anyone in IT or software development. This module introduces you to the fundamentals of SQL (Structured Query Language) and how relational databases work.

Objectives:
By the end of this module, students should be able to:

- Define SQL and understand its role in data management.
- Differentiate between DBMS and RDBMS.
- Identify the key components of a database: tables, rows, columns, primary and foreign keys.
- Write and execute basic SQL queries.

Lesson:

What is SQL?
Definition: SQL (Structured Query Language) is a standard language for accessing and manipulating databases.
Purpose: It allows users to create, retrieve, update, and delete data in a database.
Common SQL Commands:

- SELECT – retrieve data
- INSERT – add new data
- UPDATE – modify existing data
- DELETE – remove data
- CREATE, DROP, ALTER – manage database structure

Tables, Rows, Columns, Keys
Table: A collection of related data in rows and columns.
Row (Record): A single entry in a table.
Column (Field): A specific attribute or field within a table.
Primary Key: A unique identifier for each record (e.g., student_id).
Foreign Key: A column that creates a link between two tables (e.g., course_id in students that refers to courses).

Access XAMPP

- **For Windows -** XAMPP on Windows is typically installed in C:\xampp.
- **Steps:**
    - Open Command Prompt:

        - Press Win + R, type cmd, and hit Enter.

    - Navigate to XAMPP directory:
        - cd C:\xampp
    - Access MySQL from terminal:
        - cd C:\xampp\mysql\bin
        - mysql -u root -p
- **For MAC -** XAMPP is installed in /Applications/XAMPP
- **Steps:**
    - Open Terminal:
        - You can search "Terminal" via Spotlight (Cmd + Space).
    - Navigate to XAMPP directory:
        - cd /Applications/XAMPP
    - Access MySQL CLI:
        - sudo /Applications/XAMPP/xamppfiles/bin/mysql -u root -p

Database

- **Show all database**
    - SHOW DATABASES;

    You'll see output like:

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
```

- **Create database**
    - CREATE DATABASE <database_name>;
- **Delete database**
    - DROP DATABASE <database_name>;
- **Use database**
    - USE <database_name>**;**

Table

- **Show all tables**
    - SHOW TABLES;
- **Show table structure**
    - DESCRIBE <table_name>

- **Delete a table**
  - DROP TABLE <table_name>;
- **Rename a table**
  - RENAME TABLE <table old_name> TO <table new_name>;
- **Create a table**
  - *CREATE TABLE students (*
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    email VARCHAR(100)
    );

Data Manipulation

- **Insert data**
  - INSERT INTO <table_name> (col1, col2) VALUES (val1, val2);
- **View table data**
  - SELECT *
    FROM <table_name>;
- **Update data**
  - UPDATE <table_name>
    SET col1 = val
    WHERE condition;
- **Delete data**
  - DELETE FROM <table_name>
    WHERE condition;

Activity – Lesson 1

1. Create a database named **school_db**.
2. Show all databases.
3. Use the database you have created.
4. Create a table named **students** with the following columns:

Column name	Data Type	Notes
id	INT	Auto increment, primary key
name	VARCHAR(100)	
age	INT	
email	VARCHAR(100)	
course	VARCHAR(100)	

5. Show table structure
6. Insert 3 students into the table.
    a. Alice Johnson / 20 / alice@example.com / BSCS
    b. Bob Smith / 22 / bob@example.com / BSIT
    c. Clara Davis / 21 / clara@example.com / BSEMC
7. Display all records in the **students** table.
8. Display only names and emails.
9. Display students older than 20.
10. Change Clara's course to Data Science.
11. Delete the student named Bob Smith.

Activity – Lesson 1 Olagueber, Jhon M.

1. Create a database named **school_db**.

```
MariaDB [school_db]> CREATE DATABASE school_db;
```

2. Show all databases.

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| school_db          |
| test               |
+--------------------+
6 rows in set (0.001 sec)
```

3. Use the database you have created.

```
MariaDB [(none)]> USE school_db;
Database changed
```

4. Create a table named **students** with the following columns:

Column name	Data Type	Notes
id	INT	Auto increment, primary key
name	VARCHAR(100)	
age	INT	
email	VARCHAR(100)	
course	VARCHAR(100)	

```
MariaDB [school_db]> CREATE TABLE students (
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(100),
    -> age INT,
    -> email VARCHAR(100),
    -> course VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.214 sec)
```

5. Show table structure

```
MariaDB [school_db]> DESCRIBE students;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| name   | varchar(100) | YES  |     | NULL    |                |
| age    | int(11)      | YES  |     | NULL    |                |
| email  | varchar(100) | YES  |     | NULL    |                |
| course | varchar(100) | YES  |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
5 rows in set (0.018 sec)
```

6. Insert 3 students into the table.
    a. Alice Johnson / 20 / alice@example.com / BSCS
    b. Bob Smith / 22 / bob@example.com / BSIT
    c. Clara Davis / 21 / clara@example.com / BSEMC

```
MariaDB [school_db]> INSERT INTO students (name, age, email, course) VALUES
    -> ('Alice Johnson', 20, 'alice@example.com', 'BSCS'),
    -> ('Bob Smith', 22, 'bob@example.com', 'BSIT'),
    -> ('Clara Davis', 21, 'clara@example.com', 'BSEMC');
Query OK, 3 rows affected (0.263 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

7. Display all records in the **students** table.

```
MariaDB [school_db]> SELECT * FROM students;
+----+---------------+------+-------------------+--------+
| id | name          | age  | email             | course |
+----+---------------+------+-------------------+--------+
|  1 | Alice Johnson |   20 | alice@example.com | BSCS   |
|  2 | Bob Smith     |   22 | bob@example.com   | BSIT   |
|  3 | Clara Davis   |   21 | clara@example.com | BSEMC  |
+----+---------------+------+-------------------+--------+
3 rows in set (0.000 sec)
```

8. Display only names and emails.

```
MariaDB [school_db]> SELECT name, email FROM students;
+---------------+-------------------+
| name          | email             |
+---------------+-------------------+
| Alice Johnson | alice@example.com |
| Bob Smith     | bob@example.com   |
| Clara Davis   | clara@example.com |
+---------------+-------------------+
3 rows in set (0.000 sec)
```

9. Display students older than 20.

```
MariaDB [school_db]> SELECT * FROM students WHERE age > 20;
+----+-------------+------+-------------------+--------+
| id | name        | age  | email             | course |
+----+-------------+------+-------------------+--------+
|  2 | Bob Smith   |   22 | bob@example.com   | BSIT   |
|  3 | Clara Davis |   21 | clara@example.com | BSEMC  |
+----+-------------+------+-------------------+--------+
2 rows in set (0.000 sec)
```

10. Change Clara's course to Data Science.

```
MariaDB [school_db]> UPDATE students
    -> SET course = 'Data Science'
    -> WHERE name = 'Clara Davis';
Query OK, 1 row affected (0.093 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

11. Delete the student named Bob Smith.

```
MariaDB [school_db]> DELETE FROM students
    -> WHERE name = 'Bob Smith';
Query OK, 1 row affected (0.120 sec)
```