**Creating Databases and Tables in phpMyAdmin Using SQL Queries**

**A.DATABASE ANALYSE**

**Lesson Objective**

By the end of this lesson, students will be able to:

1. Create databases using SQL queries in phpMyAdmin

2. Create tables with proper structure using SQL queries

3. Understand basic data types and constraints

4. Execute SQL queries successfully in phpMyAdmin

**1. Introduction to phpMyAdmin Interface**

Accessing phpMyAdmin

1. Open your web browser

2. Navigate to your local server address (usually http://localhost/phpmyadmin)

3. Log in with your credentials (username/password)

**Main Interface Components**

- Left sidebar: Database/tables list

- Top menu: Home, SQL, Status, etc.

- Main workspace: Where you execute queries

**B.DATABASE DESIGN**

**2. Creating a Database with SQL Query**

**Basic Syntax**

CREATE DATABASE database\_name;

**Example**

CREATE DATABASE student\_management;

**Best Practices**

- Use meaningful names (avoid spaces, use underscores)

- Stick to lowercase for consistency

- Avoid special characters

**Exercise 1**

Write a SQL query to create a database named "library\_system"

**3. Selecting a Database**

Before creating tables, you need to select which database to use:

USE database\_name;

Example:

USE student\_management;

**4. Creating Tables with SQL Query**

Basic Syntax

CREATE TABLE table\_name (

column1 datatype constraints,

column2 datatype constraints,

...

);

**Example: Students Table**

CREATE TABLE students (

student\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE,

date\_of\_birth DATE,

enrollment\_date DATE DEFAULT CURRENT\_DATE,

department\_id INT,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

); **Exercise 2**

Create a "courses" table with these fields:

- course\_id (primary key, auto increment)

- course\_name (max 100 chars, not null)

- course\_code (unique, max 20 chars)

- credit\_hours (integer)

- department (max 50 chars)

**5. Viewing Table Structure**

After creation, verify your table:

DESCRIBE table\_name;

Example:

DESCRIBE students;

**6. Common Errors and Troubleshooting**

1. Forgetting to select a database first

2. Missing commas between columns

3. Forgetting to close parentheses

4. Using reserved words as names (wrap in backticks if needed)

5. Syntax errors in data types or constraints

**7. Additional Table Options**

Adding Primary Key After Creation

ALTER TABLE table\_name ADD PRIMARY KEY (column\_name);

Adding Foreign Key Relationship

ALTER TABLE table1

ADD CONSTRAINT fk\_name

FOREIGN KEY (column) REFERENCES table2(column);

**C.IMPLEMENT DATABASE**

**8.Basic INSERT Syntax**

Single Record Insertion

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

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

**Examples**

INSERT INTO students (first\_name, last\_name, email, date\_of\_birth)

VALUES ('John', 'Doe', 'john.doe@example.com', '2000-05-15');

**9.Practice Exercises**

**Exercise 3**

1. Create a database named "**company\_db**"

2. Create a "**departments**" table with:

- dept\_id (PK, auto increment)

- dept\_name (not null)

- location

3. Create an "**employees**" table with:

- emp\_id (PK, auto increment)

- first\_name

- last\_name

- email (unique)

- hire\_date

- salary

- dept\_id (foreign key to departments)

**10. Summary**

- Use **`CREATE DATABASE`** to create new databases

- Use **`USE`** to select a database

- Use **`CREATE TABLE`** with proper columns, data types and constraints

- Verify with **`DESCRIBE`**

- Always test your queries with simple examples first