

EXPERIMENT - 15

[EXPLAIN MYSQL DATABASE]

- MYSQL IS AN OPEN SOURCE RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) THAT IS WIDELY USED FOR MANAGING STRUCTURED DATA.
- IT IS ONE OF THE MOST POPULAR DATABASE SYSTEMS IN THE WORLD KNOWN FOR ITS RELIABILITY AND EASE OF USE

[KEY ASPECTS OF MYSQL]

- OPEN SOURCE
- MULTIPLE CLIENTS CAN ACCESS
- HIGHLY ~~SCARCE~~ SCALABLE
- CROSS PLATFORM COMPATIBILITY
- LARGE COMMUNITY

[HOW TO CREATE A DATABASE AND TABLE]

[STEP 1]

- OPEN XAMPP CONTROL PANEL AND START MYSQL SERVICE THEN CLICK ON "ADMIN" BUTTON.

[STEP 2]

— CLICK ON DATABASE TABS

[STEP 3]

— ENTER THE DATABASE NAME AND
CLICK "CREATE" BUTTON

[STEP 4]

— SELECT A DATABASE, THEN CLICK
"STRUCTURE". AT THE BOTTOM CLICK ON
"CREATE TABLE".

[STEP 5]

— ENTER TABLE NAME AND
NUMBER OF COLUMNS THEN ADD
THE COLUMN NAME, DATA TYPES
AND CLICK CREATE

[CONNECTING TO DATABASE]

[STEP 1]

— MAKE SURE MYSQL SERVICE IN
XAMPP IS RUNNING.

[STEP 2]

- CREATE A (PHP) FILE.

[STEP 3]

- STORE YOUR "SERVER NAME", "USERNAMe", "PASSWORD" AND "DATABASE" NAME IN 4 VARIABLES.

[STEP 4]

- USE THE FOLLOWING SYNTAX —

```
connection_variable = mysqli_connect  
(  
    server_name,  
    user_name,  
    password,  
    database_name  
)
```

[STEP 5]

- CHECK CONNECTION VARIABLE, IF ITS TRUE THEN ECHO "CONNECTION SUCCESSFUL"
ELSE "CONNECTION FAILED"

[INSERT OPERATION]

[STEP 1]

- CREATE / ESTABLISH A CONNECTION

[STEP 2]

- CREATE YOUR INSERT QUERY ACCORDING TO YOUR TABLE

[STEP 3]

- EITHER SUBMIT A FORM OR SEND STATIC VALUES

[STEP 4]

- USE THE FOLLOWING SYNTAX -

var_A = <VALUE FROM HTML FORM>

var_B = <VALUE FROM HTML FORM>

.....

var_N = <VALUE FROM HTML FORM>

query_var = <TYPE YOUR QUERY FOLLOWED BY VALUES>

`res_var = mysqli_query(conn_var, query_var)`

[SELECT OPERATION]

[STEP 1]

- CREATE / ESTABLISH A CONNECTION

[STEP 2]

- CREATE YOUR SELECT QUERY

[STEP 3]

- USE THE FOLLOWING SYNTAX -

`query_var = <TYPE QUERY HERE>`

`res_var = mysqli_query(conn_var,
query_var)`

[STEP 4]

- USE EITHER OF THE METHODS TO LOOP OVER YOUR "res_var" AND DISPLAY INFO TO USER

NXT Pg
→

[METHODS]

mysql_fetch_row() - FETCHES A SINGLE ROW

mysql_fetch_assoc() - RETURNS AN ASSOC. ARRAY

mysql_fetch_array() - RETURNS BOTH TYPES ARRAY

mysql_fetch_object() - RETURNS AN OBJECT

mysql_fetch_lengths() - RETURNS LENGTH OF FIELDS

mysql_fetch_field() - RETURNS NEXT FIELD

[SIFP S]

→ CHECK IF "res_num" LENGTH IS GREATER
THAN ZERO THEN DISPLAY DATA ELSE
DISPLAY "EMPTY TABLE"

Process

Product

Total

Sum

EXPERIMENT - 16

[UPDATE OPERATION]

[STEP 1]

- ESTABLISH A CONNECTION

[STEP 2]

- TAKE INPUT FOR WHERE CLAUSE FROM USER VIA (HTML) FORM

[STEP 3]

- CREATE YOUR UPDATE QUERY

[STEP 4]

- USE FOLLOWING SYNTAX -

where_var = < VALUE FROM HTML FORM >

query = < TYPE YOUR QUERY WITH WHERE CLAUSE VALUE >

res_var = mysqli_query (conn_var,
) query

[STEP 5]

- IF "res_var" THEN ECHO "UPDATED"
ELSE "FAILED"

[STEP 6]

- SEE IN TABLE THE CHANGES.

[DELETE OPERATION]

[STEP 1]

- ESTABLISH CONNECTION

[STEP 2]

- TAKE INPUT FROM USER FOR WHERE CLAUSE VIA (HTML) FORM.

[STEP 3]

- CREATE QUERY TO DELETE.

[STEP 4]

- USE FOLLOWING SYNTAX -

where, var = C VALUE FROM HTML FORM >

query = c' TYPE QUERY WITH WHERE
CLAUSE VALUES >

STOP ST

- ++ IF "res-var" ECHO "DELETED" ELSE
"FAILED"

STGP 6

- + VERIFY DEGGA VALUES IN TABLE