Unit 4 : Introduction of SQL

LECTURER: NANDAN PANDYA

COURSE: BACHELOR OF COMPUTER APPLICATIONS (BCA)

SEMESTER: 2ND

NAME OF THE SUBJECT: WEB PROGRAMMING

SUBJECT CODE: CS-08

COLLEGE: KAMANI SCIENCE COLLEGE, AMRELI

Prepared by: Prof. Nandan Pandya
Kamani Science College, Amreli

Topics

- Working with MySQL using PhpMyAdmin (practical)
- SQL DML Statement (Insert, Update, Select, Delete) Command
- PHP-MySQLi Connectivity
- PHP-MySQLi Functions: mysqli_connect, mysqli_close, mysqli_error, mysqli_select_db, mysqli_query, mysqli_fetch_array, mysqli_num_rows, mysqli_affected_rows, mysqli_fetch_assoc, mysqli_fetch_field, mysqli_fetch_object,mysqli_fetch_row, mysqli_insert_id, mysqli_num_fields, mysqli_data_seek

What is phpMyAdmin

- phpMyAdmin is one of the most popular applications for MySQL database management. It is a free tool written in PHP. Through this software, you can create, alter, drop, delete, import and export MySQL database tables. You can run MySQL queries, optimize, repair and check tables, change collation and execute other database management commands.

SQL

- Structured Query Language(SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.
- These SQL commands are mainly categorized into four categories as:
  - **DDL** – Data Definition Language
  - **DQl** – Data Query Language
  - **DML** – Data Manipulation Language
  - **DCL** – Data Control Language

DDL

- DDL(Data Definition Language) : DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
- Examples of DDL commands:
- **CREATE** – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
- **DROP** – is used to delete objects from the database.
- **ALTER-is** used to alter the structure of the database.
- **TRUNCATE**–is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT** –is used to add comments to the data dictionary.
- **RENAME** –is used to rename an object existing in the database.

DML

- DML(Data Manipulation Language): The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.
- Examples of DML:
- **INSERT** – is used to insert data into a table.
- **UPDATE** – is used to update existing data within a table.
- **DELETE** – is used to delete records from a database table.

DQL

- DQL (Data Query Language) :DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it.
- Example of DQL:
- **SELECT** – is used to retrieve data from the database.

DCL

- DCL(Data Control Language): DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions and other controls of the database system.
- Examples of DCL commands:
- **GRANT** -gives user's access privileges to the database.
- **REVOKE-** withdraw user's access privileges given by using the GRANT command.

TCL

- TCL(transaction Control Language): TCL commands deal with the transaction within the database.
- Examples of TCL commands:
- **COMMIT**– commits a Transaction.
- **ROLLBACK**– rollbacks a transaction in case of any error occurs.
- **SAVEPOINT**–sets a savepoint within a transaction.
- **SET TRANSACTION**–specify characteristics for the transaction.

PHP Connect to MySQL

- Establish Connection with database
- Select database
- Execute queries
- Close connection

mysqli_connect

- Before we can access data in the MySQL database, we need to be able to connect to the server
- **Syntax**: mysqli($servername, $username, $password);

mysqli_select_db

- The mysqli_select_db() function is used to change the default database for the connection.
- **Syntax**: mysqli_select_db(*connection, name*)

mysqli_query

- The mysqli_query() function performs a query against a database.
- **Syntax**: mysqli_query(connection, query)

mysqli_close

- The mysqli_close() function closes a previously opened database connection.
- **Syntax**: mysqli_close(*connection*)

mysqli_error

- The mysqli_error() function returns the last error description for the most recent function call, if any.
- **Syntax**: mysqli_error(*connection*)

mysqli_fetch_array

- The mysqli_fetch_array() function fetches a result row as an associative array, a numeric array, or both.
- **Note**: Fieldnames returned from this function are case-sensitive.
- **Syntax**: mysqli_fetch_array(*result, resulttype*)
- ***resulttype***
  - Optional. Specifies what type of array that should be produced. Can be one of the following values:
    - MYSQLI_ASSOC
    - MYSQLI_NUM
    - MYSQLI_BOTH (this is default)

mysqli_num_rows

- The mysqli_num_rows() function returns the number of rows in a result set.
- **Syntax**: mysqli_num_rows(*result*);

mysqli_affected_Rows

- The affected_rows / mysqli_affected_rows() function returns the number of affected rows in the previous SELECT, INSERT, UPDATE, REPLACE, or DELETE query.
- **Syntax**: mysqli_affected_rows(*connection*)

mysqli_fetch_assoc

- The mysqli_fetch_assoc() function fetches a result row as an associative array.
- **Note**: Fieldnames returned from this function are case-sensitive.
- **Syntax**: mysqli_fetch_assoc(result)

mysqli_fetch_field

- The mysqli_fetch_field() function returns the next field (column) in the result-set, as an object.
- **Syntax**: mysqli_fetch_field(*result*)

mysqli_fetch_object

- The mysqli_fetch_object() function returns the current row of a result-set, as an object.
- Note: Fieldnames returned from this function are case-sensitive.
- mysqli_fetch_object(result, classname, params)
- result: Required. Specifies a result set identifier returned by mysqli_query(), mysqli_store_result() or mysqli_use_result()
- Classname: Optional. Specifies the name of the class to instantiate, set the properties of, and return
- Params: Optional. Specifies an array of parameters to pass to the constructor for classname objects

mysqli_fetch_row

- The mysqli_fetch_row() function fetches one row from a result-set and returns it as an enumerated array.
- Syntax: mysqli_fetch_row(result)

mysqli_insert_id

- The mysqli_insert_id() function returns the id (generated with AUTO_INCREMENT) from the last query.
- Syntax: mysqli_insert_id(connection)

mysqli_num_fields

- The mysqli_num_fields() function returns the number of fields (columns) in a result set.
- Syntax: mysqli_num_fields(result);

mysqli_data_seek

- The mysqli_data_seek() function adjusts the result pointer to an arbitrary row in the result-set.
- Syntax: mysqli_data_seek(*result,offset);*
- *Result: Required. Specifies a result set identifier returned by mysqli_query(), mysqli_store_result() or mysqli_use_result()*
- *Offset: Required. Specifies the field offset. Must be between 0 and the total number of rows - 1*