Unit -3
Accessing Mysql with PHP

PREPARED BY: PROF.ANERI PATEL

Introduction to MySQL

- MySQL was first released in 1995
- MySQL is a relational database management system (RDBMS) that is used to store and retrieve data.
- It is open-source software and is widely used for web-based applications.
- MySQL is used to store information such as user accounts, product information, and other types of data required by a web application.
- MySQL is very fast, reliable, scalable, and easy to use.

MySQL DataType

- String Data Type:

  - CHAR(size) : A FIXED length string (can contain letters, numbers, and special characters). The *size* parameter specifies the column length in characters - can be from 0 to 255. Default is 1
  - VARCHAR(size): A VARIABLE length string (can contain letters, numbers, and special characters). The *size* parameter specifies the maximum column length in characters - can be from 0 to 65535
  - TEXT(size) :Holds a string with a maximum length of 65,535 bytes

MySQL DataType

- Numeric Data Type:

  - INT(*size*): A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The *size* parameter specifies the maximum display width (which is 255)
  - FLOAT
  - DOUBLE(*size*, *d*)
  - DECIMAL(*size*, *d*)

MySQL DataType

Date and Time Data Type:

- DATE : A date. Format: YYYY-MM-DD
- DATETIME : A date and time combination. Format: YYYY-MM-DD hh:mm:ss.
- TIMESTAMP : A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss.
- TIME : A time. Format: hh:mm:ss.
- YEAR : A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000.

MySQL function

- ❑MySQL has many built-in functions.
- ❑This reference contains string, numeric, date, and some advanced functions in MySQL.

String Function in MySQL

❑CHAR_LENGTH : Returns the length of a string (in characters)

- Syntax: CHAR_LENGTH(*string*)
- Example: SELECT CHAR_LENGTH(“String Function") AS LengthOfString;

❑ CONCAT : Adds two or more expressions together

- Syntax: CONCAT(*expression1*, *expression2*, *expression3*,...)
- SELECT CONCAT(“MySQL ", "Tutorial ", "is ", "fun!") AS ConcatenatedString;

❑INSERT : Inserts a string within a string at the specified position and for a certain number of characters

- Syntax: INSERT(*string*, *position*, *number*, *string2*)
- SELECT INSERT("google.com", 1, 6, "yahoo");

String Function in MySQL

- ❑ INSTR :Returns the position of the first occurrence of a string in another string
  - ❑Syntax: `INSTR(string1, string2)`
  - ❑SELECT INSTR("google.com", "COM") AS MatchPosition;
- ❑ LCASE / LOWER : Converts a string to lower-case
  - ❑Syntax: `LCASE(text)    / LOWER(text)`
  - ❑`SELECT LCASE(“MySQL Tutorial");    / SELECT LOWER(“MySQL Tutorial");`
- ❑LENGTH : Returns the length of a string (in bytes)
  - Syntax: `LENGTH(string)`
  - SELECT LENGTH("MySQL Tutorial") AS LengthOfString;

String Function in MySQL

❑LEFT : Extracts a number of characters from a string (starting from left)

   ❑Syntax: `LEFT(`*`string`*`, `*`number_of_chars`*`)`

   ❑Example: SELECT LEFT("Hello,How are You?", 3) AS ExtractString;

❑ RIGHT : Extracts a number of characters from a string (starting from right)

   ❑ Syntax : `RIGHT(`*`string`*`, `*`number_of_chars`*`)`

   ❑Example: SELECT RIGHT("Hello, How are you?", 4) AS ExtractString;

❑ UCASE / UPPER : Converts a string to upper-case

   ❑`Syntax: UCASE(`*`text`*`)  / UPPER(`*`text`*`)`

   ❑`Example: SELECT UCASE("MySQL Tutorial") AS UppercaseText;`

   ❑`SELECT UPPER("MySQL Tutorial") AS UppercaseText;`

String Function in MySQL

- MID :Extracts a substring from a string (starting at any position)
  - Syntax: MID(*string*, *start*, *length*)
  - Example : SELECT MID("MySQL Tutorial", 7, 3) AS ExtractString;
- POSITION : Returns the position of the first occurrence of a substring in a string
  - Syntax: POSITION(*substring* IN *string*)
  - SELECT POSITION("le" IN "google.com") AS MatchPosition;
- REPEAT : Repeats a string as many times as specified
  - Syntax: REPEAT(*string*, *number*)
  - SELECT REPEAT("MySQL Tutorial", 3);

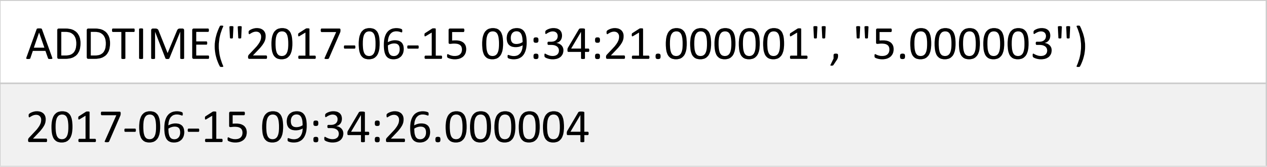
String Function in MySQL

- ❑ REPLACE : Replaces all occurrences of a substring within a string, with a new substring
  - ❑Syntax: `REPLACE(string, substring, new_string)`
  - ❑ SELECT REPLACE("MySQL Tutorial", "MySQL", "HTML");
- ❑ REVERSE : Reverses a string and returns the result
  - ❑Syntax : `REVERSE(string)`
  - ❑ SELECT REVERSE("Hello Everyone!");
- ❑ STRCMP : Compares two strings
  - ❑`STRCMP(string1, string2)`
  - ❑ SELECT STRCMP("SQL Tutorial", "SQL Tutorial");

Date and Time Function in MySQL

- ADDDATE : Adds a time/date interval to a date and then returns the date
  - Syntax: `ADDDATE(date, days)`
  - SELECT ADDDATE("2024-09-25", INTERVAL 10 DAY);

- ADDTIME : Adds a time interval to a time/datetime and then returns the time/datetime
  - Syntax: `ADDTIME(datetime, addtime)`
  - Example: SELECT ADDTIME("2017-06-15 09:34:21.000001", "5.000003");
  - Output:



ADDTIME("2017-06-15 09:34:21.000001", "5.000003")
2017-06-15 09:34:26.000004

- CURDATE / CURRENT_DATE: Returns the current date
  - Syntax: `CURDATE() / CURRENT_DATE()`
  - SELECT CURDATE(); / SELECT CURRENT_DATE();

Date and Time Function in MySQL

- ❑CURRENT_TIME : Returns the current time
  - ❑Syntax: `CURRENT_TIME()`
  - ❑ SELECT CURRENT_TIME();
- ❑ CURRENT_TIMESTAMP : Returns the current date and time
  - ❑ Syntax: `CURRENT_TIMESTAMP()`
  - ❑ Example : SELECT CURRENT_TIMESTAMP();
  - ❑Output: 2024-10-03 07:04:05
- ❑ CURTIME : Returns the current time
  - ❑Syntax: `CURTIME()`
  - ❑ Example: SELECT CURTIME();

Date and Time Function in MySQL

- DATE : Extracts the date part from a datetime expression
  - Syntax: `DATE(`*`expression`*`)`
  - SELECT DATE("2024-09-03");

- DATEDIFF : Returns the number of days between two date values(date1-date2)
  - Syntax : `DATEDIFF(`*`date1, date2`*`)`
  - SELECT DATEDIFF("2024-10-02", "2024-09-25");

- DATE_FORMAT : Formats a date
  - Syntax: `DATE_FORMAT(`*`date, format`*`)`
  - `SELECT DATE_FORMAT("2024-10-03", "%Y");      //2024`
  - `SELECT DATE_FORMAT("2017-06-15", "%d");      // 15`
  - `SELECT DATE_FORMAT("2017-06-15", "%j");      // 166(day of the year)`

Date and Time Function in MySQL

- DAY : Returns the day of the month for a given date
  - Syntax: `DAY(`*`date`*`)`
  - `SELECT DAY("2024-10-03");`
- DAYNAME : Returns the weekday name for a given date
  - Syntax: `DAYNAME(`*`date`*`)`
  - SELECT DAYNAME("2024-10-04"); // Friday
- DAYOFMONTH : Returns the day of the month for a given date
  - Syntax: `DAYOFMONTH(`*`date`*`)`
  - `SELECT DAYOFMONTH("2024-10-04");    //4`

Date and Time Function in MySQL

- ❑DAYOFWEEK : Returns the weekday index for a given date
  - ❑Syntax:`DAYOFWEEK(date)`
  - ❑ `SELECT DAYOFWEEK("2024-10-04");    //6 means friday`
- ❑ DAYOFYEAR : Returns the day of the year for a given date
  - ❑Syntax: `DAYOFYEAR(date)`
  - ❑ `SELECT DAYOFYEAR("2024-10-04");    //278`
- ❑ HOUR : Returns the hour part for a given date
  - ❑Syntax: `HOUR(datetime)`
  - ❑ `SELECT HOUR("2024-10-04 09:34:00");`

Date and Time Function in MySQL

- LAST_DAY : Extracts the last day of the month for a given date
  - Syntax: LAST_DAY(*date*)
  - SELECT LAST_DAY("2024-10-04");
- MINUTE : Returns the minute part of a time/datetime
  - Syntax: MINUTE(*datetime*)
  - SELECT MINUTE("2024-10-04 09:34:00");
- MONTH : Returns the month part for a given date
  - Syntax: MONTH(*date*)
  - SELECT MONTH("2024-10-04");

Date and Time Function in MySQL

- MONTHNAME : Returns the name of the month for a given date
  - Syntax: `MONTHNAME(`*`date`*`)`
  - `SELECT MONTHNAME("2024-10-04");   //october`
- NOW : Returns the current date and time
  - Syntax: `NOW()`
  - `SELECT NOW();  // 2024-10-04 07:43:29`
- SECOND : Returns the seconds part of a time/datetime
  - Syntax: `SECOND(`*`datetime`*`)`
  - `SELECT SECOND("2024-10-04 09:34:01.000023");    //1`

Date and Time Function in MySQL

- ❑ SYSDATE : Returns the current date and time
  - ❑Syntax: `SYSDATE()`
  - ❑ `SELECT SYSDATE() + 1;` // YYYYMMDDHHMMSS
- ❑ TIME :Extracts the time part from a given time/datetime
  - ❑Syntax: `TIME(`*`expression`*`)`
  - ❑ `SELECT TIME("2024-10-04 19:30:10");` // 19:30:10
- ❑ TIMEDIFF : Returns the difference between two time/datetime expressions
  - ❑Syntax: `TIMEDIFF(`*`time1, time2`*`)`
  - ❑ SELECT TIMEDIFF("13:11:11", "13:10:10"); // 00:01:01

Date and Time Function in MySQL

- ❑ TO_DAYS : Returns the number of days between a date and date "0000-00-00‟
  - ❑Syntax: `TO_DAYS(`*`date`*`)`
  - ❑

- ❑ WEEK : Returns the week number for a given date
  - ❑Syntax: `WEEK(`*`date, firstdayofweek`*`)`
  - ❑ `SELECT WEEK("2024-10-04");  //39`

- ❑ YEAR : Returns the year part for a given date
  - ❑Syntax: `YEAR(`*`date`*`)`
  - ❑ SELECT YEAR("2024-10-04"); //2024

MySQL Vs. MySQLi function

- □ **Object-oriented interface**. You can still use the "old procedural" way of calling the mysql extension but the OO version groups the functions by their purpose.
- □ **Prepared Statements**. Those are useful to prevent SQL injections and are executed faster.
- □ **Multiple Statements**. With this "feature", you can execute multiple SQL queries inside only one "mysqli" call. This reduces the round trips between the database server and the PHP server.
- □ **Support for Transactions**. This is really useful to write robust applications. It gives you the ability to write a group of SQL statements that will either be executed or all rolled back (usually if there is an error somewhere in the process).
- □ **Enhanced debugging capabilities**. As an example, you can use "mysqli_debug(...)" to save debugging information into a file.
- □ **Embedded server support**. Since MySQL 4.0, there is a library available that can be used to run a complete MySQL server embedded inside a program, usually a desktop application.

Opening and Closing a MySQL Connection

- Open a connection to a MySQL database server with the **`mysqli_connect()`** function.
- The **`mysqli_connect()`** function returns a positive integer if it connects to the database successfully or `FALSE` if it does not.
- Assign the return value from the **`mysqli_connect()`** function to a variable that you can use to access the database in your script.

Syntax

mysql_connect(server,user,pwd,database)

Parameter Description

server : Optional. Specifies the server to connect to (can also include a port number, e.g. "hostname:port" or a path to a local socket for the localhost). Default value is "localhost:3306"

user : Optional. Specifies the username to log in with. Default value is the name of the user that owns the server process

pwd : Optional. Specifies the password to log in with. Default is “ "

Connecting to MySQL with PHP

- Example :

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

$database = “databasename";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error)

{
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Closing to MySQL connection

- Close a database connection using the **`mysqli_close()`** function
- Syntax : mysqli_close(connection variablename);

`example : mysqli_close($conn);`