



Web Development Using PHP

Unit-4

Semester-4

THE INSB IITMS BCA & PGDCA College, Idar

Introduction to MySQL

- **What is MySQL?**

MySQL is an open-source relational database management system (RDBMS). It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle Corporation.

- The data in a MySQL database are stored in tables which consists of columns and rows.
- MySQL is a database system that runs on a server.
- MySQL is ideal for both small and large applications.
- MySQL is very fast, reliable, and easy to use database system. It uses standard SQL.
- MySQL compiles on a number of platforms.
- **What is Database:-** A database is used to store a collection of data has one or more distinct APIs for creating, accessing, managing, searching and replicating the data.

- **Features of MySQL:-**

- ❖ **Open-Source:-** MySQL is open-source, which means this software can be downloaded, used and modified by anyone. It is free-to-use and easy-to-understand. The source code of MySQL can be studied, and changed based on the requirements.
- ❖ **Quick and Reliable:-** MySQL stores data efficiently in the memory ensuring that data is consistent, and not redundant. data access and manipulation using MySQL is quick. It is considered one of the fastest relational databases with higher productivity as well.
- ❖ **High Performance:-** MySQL provides comparatively higher performance without affecting its functionality. It also has a very little memory leakage making it memory efficient as well.
- ❖ **Scalable :-** Scalability refers to the ability of systems to work easily with small amounts of data, large amounts of data, clusters of machines, and so on. MySQL server was developed to work with large databases.

- ❖ **Data Types:-** It contains multiple data types such as unsigned integers, signed integers, float (FLOAT), double (DOUBLE), character (CHAR), variable character (VARCHAR), text, date, time, datetime, timestamp, year, and so on.
- ❖ **Character Sets :-** It supports different character sets, and this includes latin1 (cp1252 character encoding), German, Ujis, other Unicode character sets and so on.
- ❖ **Secure :-** It provides a secure interface since it has a password system which is flexible, and ensures that it is verified based on the host before accessing the database.
- ❖ **Platform Independent:-** MySQL can be run on various operating systems including Windows, Linux, macOS etc. in several programming languages like C, C++, Java, Python, Perl, PHP etc.
- ❖ **Support for large databases.**
- ❖ **Client and Utility Programs.**

Merits and Demerits

Advantages	Disadvantages
Open source and free of cost	Unsafe to use and handle since it has an open-source accessible to all
It can run on various OS	Not for running giant applications
Loading applications is straightforward since it can load over slow internet	Weak and causes input of incorrect data
Easy to learn and use	One needs to learn built-in functionalities to avoid writing excess codes
Good support and assistance	Using many features can cause poor performance
Helps to manage codes easily	Does not allow modification of online applications
Powerful library support	Not equivalent in performance to others
Build-in database connection helps to connect databases	Tough to manage and incompetent

MySQL Constraints

- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.
- The following constraints are commonly used in SQL
- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE**- Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY**- Prevents actions that would destroy links between tables → **CHECK** - Ensures that the values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column if no value is specified
- **CREATE INDEX** - Used to create and retrieve data from the database very quickly

MySQL Datatypes

- An SQL developer must decide what type of data that will be stored inside each column when creating a table. The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.
- MySQL there are three main data types: string, numeric, and date and time.

Data type	Description
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
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The size parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The size parameter specifies the maximum column length in bytes.

Numeric

Data type	Description
BIT(size)	A bit-value type. The number of bits per value is specified in <i>size</i> . The <i>size</i> parameter can hold a value from 1 to 64. The default value for <i>size</i> is 1.
TINYINT(size)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The <i>size</i> parameter specifies the maximum display width (which is 255)
BOOL	Zero is considered as false, nonzero values are considered as true.
BOOLEAN	Equal to BOOL
SMALLINT(size)	A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The <i>size</i> parameter specifies the maximum display width (which is 255)
MEDIUMINT(size)	A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The <i>size</i> parameter specifies the maximum display width (which is 255)
INT(size)	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The <i>size</i> parameter specifies the maximum display width (which is 255)
INTEGER(size)	Equal to INT(size)
BIGINT(size)	A large integer. Signed range is from -9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The <i>size</i> parameter specifies the maximum display width (which is 255)
FLOAT(size, d)	A floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions
FLOAT(p)	A floating point number. MySQL uses the <i>p</i> value to determine whether to use FLOAT or DOUBLE for the resulting data type. If <i>p</i> is from 0 to 24, the data type becomes FLOAT() . If <i>p</i> is from 25 to 53, the data type becomes DOUBLE()
DOUBLE(size, d)	A normal-size floating point number. The total number of digits is specified in <i>size</i> . The number of digits after the decimal point is specified in the <i>d</i> parameter

Date and Time data type

Data type	Description
DATE	A date. Format:YYYY-MM-DD.The supported range is from '1000-01-01' to '9999-12-31'
DATETIME(<i>fsp</i>)	A date and time combination. Format:YYYY-MM-DD hh:mm:ss.The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time
TIMESTAMP(<i>fsp</i>)	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.The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC.Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT_TIMESTAMP and ON UPDATE CURRENT_TIMESTAMP in the column definition
TIME(<i>fsp</i>)	A time. Format: hh:mm:ss.The supported range is from '-838:59:59' to '838:59:59'
YEAR	A year in four-digit format.Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format.

MySQL Integration

- The MySQL integration collects logs and metrics data, providing comprehensive insights into database operations and performance.
- Collect error and slow query logs, as well as status, replication status metrics, to provide insights into database operations, query performance and replication health.
- Create informative visualizations to track usage trends, measure key metrics, and derive actionable business insights.
- Set up alerts to minimize Mean Time to Detect (MTTD) and Mean Time to Resolve (MTTR) by quickly referencing relevant logs during troubleshooting.

Basic SQL Commands

- **SELECT** - extracts data from a database.
- **UPDATE** - updates data in a database.
- **DELETE** - deletes data from a database.
- **INSERT INTO** - inserts new data into a database.
- **CREATE DATABASE** - creates a new database.
- **ALTER DATABASE** - modifies a database.
- **CREATE TABLE** - creates a new table.

PHP Connect to MySQL

- PHP and later can work with a MySQL database using:
- **MySQLi extension** (the "i" stands for improved)
- **PDO (PHP Data Objects)**
- **Connection to MySQL using MySQLi:-**
- PHP provides `mysql_connect()` function to open a database connection.
- This function takes a single parameter, which is a connection returned by the `mysql_connect()` function.
- You can disconnect from the MySQL database anytime using another PHP function `mysql_close()`.
- There is also a procedural approach of MySQLi to establish a connection to MySQL database from a PHP script.

PHP Connect to MySQL

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="database";
// Create connection
$conn = mysqli_connect($servername, $username,
$password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_error($conn));
}
echo "Connected successfully";
?>
```

Mysql Functions

- **mysql_connect():**- The connect() / mysql_connect() function opens a new connection to the MySQL server.
- **Syntax:-** \$mysql -> new mysql(*host, username, password, dbname, port, socket*)
- **mysql_select_db():**- The select_db() / mysql_select_db() function is used to change the default database for the connection.
Ex. \$mysql -> select_db(*name*)
- **mysql_query():**- The query() / mysql_query() function performs a query against a database.
Ex. \$mysql -> query(*query, resultmode*)
- **mysql_num_rows():**- The mysql_num_rows() function returns the number of rows in a result set.
- Ex. mysql_num_rows(*result*); Specifies a result set identifier returned by mysql_query()

- **mysqli_fetch_array():**- The `fetch_array()` / `mysqli_fetch_array()` function fetches a result row as an associative array, a numeric array, or both.
 - Ex. `$mysqli_result -> fetch_array(resulttype)`
 - *resulttype*:-`MYSQLI_ASSOC`, `MYSQLI_NUM`, `MYSQLI_BOTH` (this is default)
- **mysqli_fetch_field():**- The `fetch_field()` / `mysqli_fetch_field()` function returns the next field (column) in the result-set, as an object.
 - Ex:- `$mysqli_result -> fetch_field()`
- **mysqli_fetch_row():**- The `fetch_row()` / `mysqli_fetch_row()` function fetches one row from a result-set and returns it as an enumerated array.
 - Ex:- `$mysqli_result -> fetch_row()`
- **mysqli_connect_error():**- The `connect_error` / `mysqli_connect_error()` function returns the error description from the last connection error, if any.
 - Ex:- `$mysqli -> connect_error`.

- **mysql_affected_rows():**- The `affected_rows / mysql_affected_rows()` function returns the number of affected rows in the previous `SELECT`, `INSERT`, `UPDATE`, `REPLACE`, or `DELETE` query.
Ex:- `$mysqli -> affected_rows()`
- **mysql_error():**- The `error / mysql_error()` function returns the last error description for the most recent function call, if any. Ex:- `$mysqli -> error()`
- **mysql_free_result():**- The `mysql_free_result()` function frees the memory associated with the result.
Ex:- `mysql_free_result(result);`
- **mysql_close():**- The `close() / mysql_close()` function closes a previously opened database connection.
Ex:- `$mysqli -> close()`