Unit-IV

MySQL

Introduction to MySql

- MySQL is a database system used for developing web-based software applications.
- MySQL is an open-source, fast reliable, and flexible relational database management system, typically used with PHP
- MySQL used for both small and large applications.
- MySQL is a relational database management system *(RDBMS)*.
- MySQL is fast, reliable, and flexible and easy to use.
- MySQL supports standard SQL *(Structured Query Language)*.
- MySQL is free to download and use.
- MySQL was developed by Michael Widenius and David Axmark in 1994.
- MySQL is presently developed, distributed, and supported by Oracle Corporation.
- MySQL Written in C, C++.

Features of MYSQL

- MySQL server design is multi-layered with independent modules.
- MySQL is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available.
- MySQL provides transactional and non-transactional storage engines.
- MySQL has a high-speed thread-based memory allocation system.
- MySQL supports in-memory heap table.
- MySQL Handles large databases.
- MySQL Server works in client/server or embedded systems.
- MySQL Works on many different platforms.

Benefits of Mysql

1. Data Security

2. On-Demand Scalability

3. High Performance

4. Comprehensive Transactional support

5. Open Source

Reserve words, Keywords, variables

The following list shows the keywords and reserved words in MySQL 5.5, along with changes to individual words from version to version.

- `ADD`
- `AFTER`
- `AGGREGATE`
- `ALL`
- `AND`
- `AS`
- `ASC`
- `AUTO_INCREMENT`
- `AVG`

B

- `BACKUP`
- `BEFORE`
- `BEGIN`
- `BETWEEN`
- `BIGINT`
- `BINARY`
- `BLOB`
- `BLOCK`
- `BOOLEAN`
- `BOTH`
- `BY`
- `BYTE`

C

- `CACHE`
- `CALL`

D

- DATA
- DATABASE
- DATE
- DATETIME
- DAY
- DAY_HOUR
- DAY_MICROSECOND
- DAY_MINUTE

Data types

MySQL numeric data types

Numeric Types	Description
TINYINT	A very small integer
SMALLINT	A small integer
MEDIUMINT	A medium-sized integer
INT	A standard integer
BIGINT	A large integer
DECIMAL	A fixed-point number
FLOAT	A single-precision floating point number
DOUBLE	A double-precision floating point number
BIT	A bit field

MySQL string data types

String Types	Description
CHAR	A fixed-length nonbinary (character) string
VARCHAR	A variable-length non-binary string
BINARY	A fixed-length binary string
VARBINARY	A variable-length binary string
TINYBLOB	A very small BLOB (binary large object)
BLOB	A small BLOB
MEDIUMBLOB	A medium-sized BLOB
LONGBLOB	A large BLOB
TINYTEXT	A very small non-binary string
TEXT	A small non-binary string
MEDIUMTEXT	A medium-sized non-binary string
LONGTEXT	A large non-binary string

String Types	Description
ENUM	An enumeration; each column value may be assigned one enumeration

MySQL date and time data types

Date and TimeTypes	Description
DATE	A date value in CCYY-MM-DD format
TIME	A time value in hh:mm:ss format
DATETIME	A date and time value inCCYY-MM-DD hh:mm:ssformat
TIMESTAMP	A timestamp value in CCYY-MM-DD hh:mm:ss format
YEAR	A year value in CCYY or YY format

MySQL Boolean data types

MySQL does not have the built-in BOOLEANor BOOL data type. To represent Boolean values, MySQL uses the smallest integer type which isTINYINT(1). In other words, BOOLEAN and BOOL are synonyms for TINYINT(1).

Types of Commands

i) Data Definition Command-

Data Definition Language (DDL) statements are used to define the database structure or schema. Data Definition Language understanding with database schemas and describes how the data should consist in the database, therefore language statements like CREATE TABLE or ALTER TABLE belongs to the DDL. DDL is about "metadata".

DDL includes commands such as CREATE, ALTER and DROP statements.DDL is used to CREATE, ALTER OR DROP the database objects (Table, Views, Users).

Data Definition Language (DDL) are used different statements :

- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database

- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- COMMENT - add comments to the data dictionary
- RENAME - rename an object

***(Note for student:** write example and output by your own on the basis of syntax)*

CREATE TABLE

Syntax: Create table table name(fieldname1 datatype(),fieldname2 datatype()...);

Ex: create table student(id int, name varchar(20));

ALTER TABLE
1. ADD
2.MODIFY

ADD
Syntax:alter table table name ADD (fieldname datatype()...);

MODIFY
syntax: alter table table name modify (fieldname datatype()...);

DESCRIBE TABLE
Syntax: DESCRIBE TABLE NAME;

DROP TABLE
Syntax: DROP Table name;

COMMENT - add comments to the data dictionary

RENAME - rename a table
Synatax: rename table table name to new table name

ii)Data Manipulation Command

Data Manipulation Language (DML) statements are used for managing data within schema objects DML deals with data manipulation, and therefore includes most common SQL statements such SELECT, INSERT, etc. DML allows to add / modify / delete data itself.

DML is used to manipulate with the existing data in the database objects (insert, select, update, delete).

DML Commands:
1.INSERT
2.SELECT
3.UPDATE
4.DELETE

***INSERT:**
Syntax: INSERT INTO Table name values();

Ex: insert into student values(20,"ABC");

***SELECT:**
Syntax: Select*from <table name>

EX: Select * from student;

***UPDATE:**
Syntax: Update<table name> set to(calculation);

Ex: update student set name="XYZ" where id=20;

***DELETE:**
Syntax: Delete from<table name> where condition;

EX: delete from student where id=20;

iii)Data Control Command

DCL is the abstract of Data Control Language. Data Control Language includes commands such as GRANT, and concerns with rights, permissions and other controls of the database system. DCL is used to grant / revoke permissions on databases and their contents. DCL is simple, but MySQL permissions are a bit complex. DCL is about security. DCL is used to control the database transaction.DCL statement allow you to control who has access to specific object in your database.

1. GRANT
2. REVOKE

GRANT : It provides the user's access privileges to the database. In the MySQL database offers both the administrator and user a great extent of the control options. By the administration side of the process includes the possibility for the administrators to control certain user privileges over the MySQL server by restricting their access to an entire the database or user limiting permissions for a specific table. It Creates an entry in the security system that allows a user in the current database to work with data in the current database or execute specific statements.

Syntax :
Statement permissions:

```
GRANT { ALL | statement [ ,...n ] }
TO security_account [ ,...n ]
```

Normally, a database administrator first uses CREATE USER to create an account, then GRANT to define its privileges and characteristics.

For example:

```
CREATE USER 'arjun'@'localhost' IDENTIFIED BY 'mypass';

GRANT ALL ON db1.* TO 'arjun'@'localhost';

GRANT SELECT ON child TO 'arjun'@'localhost';

GRANT USAGE ON *.* TO 'arjun'@'localhost' WITH MAX_QUERIES_PER_HOUR 90;
```

REVOKE : The REVOKE statement enables system administrators and to revoke the privileges from MySQL accounts.

Syntax : REVOKE

```
priv_type [(column_list)]
[, priv_type [(column_list)]] ...
ON [object_type] priv_level
FROM user [, user] ...

REVOKE ALL PRIVILEGES, GRANT OPTION
FROM user [, user] ...
```

For example:

```
mysql> REVOKE INSERT ON *.* FROM 'arjun'@'localhost';
```

Command Clauses-

1.where: The WHERE clause is used to filter records. The WHERE clause is used to extract only those records that fulfill a specified condition.

WHERE Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Example: (**Notes for student**: While giving example- take one table and extract values from those table)

```
SELECT * FROM students
WHERE name='abc';
```

2.order by: The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

Example:

```
SELECT * FROM Students
ORDER BY name;
```

3.Group by

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Example:

```
SELECT COUNT(studID), name
FROM Students
GROUP BY name;
```

4. Having

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

Example

```
SELECT COUNT(studID), name
FROM Student
GROUP BY name
HAVING COUNT(studID) > 5;
```

5.Like

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

- % - The percent sign represents zero, one, or multiple characters
- _ - The underscore represents a single character

syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

Example:

The following SQL statement selects all customers with a CustomerName starting with "a":

```
SELECT * FROM Student
WHERE name LIKE 'a%';
```

6.Between

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

Example

```
SELECT * FROM Employee
WHERE salary BETWEEN 10000 AND 20000;
```

Database Connectivity:

PHP 5 and later can work with a MySQL database using:

- **MySQLi extension** (the "i" stands for improved)
- **PDO (PHP Data Objects)**

Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.

Should I Use MySQLi or PDO?

If you need a short answer, it would be "Whatever you like".

Both MySQLi and PDO have their advantages:

PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.

So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.

Both are object-oriented, but MySQLi also offers a procedural API.

Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security

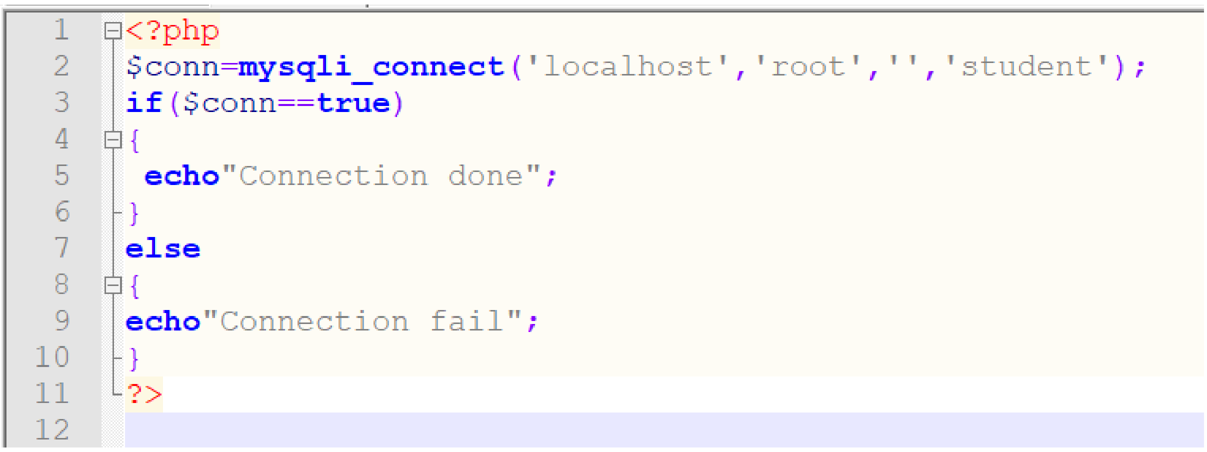
Syntax of MYSQLI:

```
$conn = mysqli_connect( “servername”, “username”, “password”, “
        Database-name”);
```

Syntax of PDO:

```
    $conn = new PDO("mysql:host=servername;dbname=myDB", username,
    password);
```

Example of Database connectivity through MYSQLI:



```
<?php
$conn=mysqli_connect('localhost','root','','student');
if($conn==true)
{
 echo"Connection done";
}
else
{
echo"Connection fail";
}
?>

```