Experiment No: 2

FAMILIARIZATION TO MYSQL

AIM : To Familiarize with MySQL

MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is named after co-founder Monty Widenius's daughter, My. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

MySQL Data Types

Each data type in MySQL can be determined by the following characteristics:

- Kind of values it can represent.
- The space that takes up and whether the values are fixed-length or variable-length.
- Does the values of the data type can be indexed.
- How MySQL compares the value of a specific data type.

DATATYPE	**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
CHAR	A fixed-length non-binary (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
ENUM	An enumeration; each column value may be assigned one enumeration member

SET	A set; each column value may be assigned zero or more set members
DATE	A date value in ‘CCYY-MM-DD’ format
TIME	A time value in ‘hh:mm:ss’ format
DATETIME	A date and time value in ‘CCYY-MM-DD hh:mm:ss’ format
TIMESTAMP	A timestamp value in ‘CCYY-MM-DD hh:mm:ss’ format
YEAR	A year value in CCYY or YY format
GEOMETRY	A spatial value of any type
POINT	A point (a pair of X Y coordinates)
LINESTRING	A curve (one or more POINT values)
POLYGON	A polygon
GEOMETRYCOLLECTION	A collection of GEOMETRY values
MULTILINESTRING	A collection of LINESTRING values
MULTIPOINT	A collection of POINT values
MULTIPOLYGON	A collection of POLYGON values

Data Definition Language (DDL) Commands

1. Creating Database

```
CREATE DATABASE [IF NOT EXISTS] database_name;
```

2. Displaying Database

```
SHOW DATABASES;
```

3. Selecting a database to work with

```
USE database_name;
```

4. Removing Database

```
DROP DATABASE [IF EXISTS] database_name;
```

5. Creating tables

```
CREATE TABLE [IF NOT EXISTS] table_name(
        column_list ) engine=table_type;
```

engine specifies the storage engine for the table such as the storage engine for the table

```
column_name data_type[size] [NOT NULL|NULL] [DEFAULT value]

[AUTO_INCREMENT]
```

To create a PRIMARY KEY constraint for the table, specify the PRIMARY KEY in the primary key column’s definition.

To set particular columns of the table as the primary key

```
PRIMARY KEY (col1,col2,...)
```

Foreign Key constraint

```
CONSTRAINT constraint_nameFOREIGN KEY foreign_key_name (columns)

REFERENCES parent_table(columns)ON DELETE actionON UPDATE action
```

6. To add a new column to an existing table or to make changes to it.

```
ALTER TABLE table_name

ADD column_name datatype;
```

7. You can also use the MODIFY statement to change column data types.

```
ALTER TABLE table_name

MODIFY COLUMN  column_name datatype;
```

8. To delete a column in a table

```
ALTER TABLE table_name

DROP COLUMN column_name;
```

9. To Rename a Table:

```
            ALTER TABLE names RENAME AS new name;
```

10. To Delete a Table:

```
                DROP [TEMPORARY] TABLE [IF EXISTS] table_name [,
table_name] ...
                [RESTRICT | CASCADE]
```

We can check the NOTE, which is generated by MySQL because of non-existent table, by using the SHOW WARNING statement as follows:

```
            SHOW WARNINGS;
```

Data Manipulation Language (DML) Commands

1. To Insert data into Tables:

```
        INSERT INTO table(column1,column2...)
        VALUES (value1,value2,...)
```

To insert multiple rows:

```
        INSERT INTO table(column1,column2...)
        VALUES (value1,value2,...),
               (value1,value2,...),
```

2. To Query Tables:

```
        SELECT column_1,column_2...
        FROM table_1
        [INNER | LEFT |RIGHT] JOIN table_2 ON conditions
        WHERE conditions
        GROUP BY group
```

```
HAVING group_conditions

ORDER BY column_1 [ASC | DESC]

LIMIT offset, row_count
```

To remove duplicate rows:

```
SELECT DISTINCT columns

FROM table_name

WHERE where_conditions
```

The IN operator allows you to determine if a specified value matches any one of a list or a subquery.

```
SELECT column_list

FROM table_name

WHERE (expr|column) IN ('value1','value2',...)
```

The BETWEEN operator allows you to specify a range to test.

```
expr (NOT) BETWEEN begin_expr AND end_expr
```

The LIKE operator is commonly used to select data based on patterns. MySQL provides two wildcard characters for using with the LIKE operator, the percentage % and underscore _.

- The percentage (%) wildcard allows to match any string of zero or more characters.
- The underscore (_) wildcard allows you to match any single character.

MySQL supports two kinds of aliases which are known as column alias and table alias.

Column Alias:

```
SELECT [col1 | expression] AS `descriptive name`

FROM table_name
```

Table alias:

```
table_name AS table_alias
```

3. To Update Tables:

```
UPDATE [LOW_ PRIORITY] [IGNORE] table_name [, table_name...]
 SET column_name1 = expr1
          [, column_name2=expr2 ...]
          [WHERE condition]
```

4. To Delete Tables:

```
DELETE FROM table
[WHERE conditions] [ORDER BY ...] [LIMIT rows]
```

Views

A database view is a virtual table or logicaltable which is defined as a SQL SELECT query with joins. Because a database view is similar to a database table, which consists of rows and columns, so you can query data against it. Most database management systems, including MySQL, allows you to update data in the underlying tables through the database view with some prerequisites.

```
CREATE
VIEW [database_name].[view_name]
 AS
 [SELECT  statement]
```