Simple Quesries in SQL & Table Creation and Data Manipulation

Based on CBSE Curriculum

Class -~~11~~

Chapter- 17 & 18

By-
Neha Tyagi
PGT CS
KV 5 Jaipur II Shift
Jaipur Region

Introduction

- SQL was developed in 1970 in IBM lab. It is also known as SEQUEL which was a non-procedural language.
- It always specifies that WHAT is required?
- We can perform following tasks in SQL-

I. Creation and modification of Database structure.

II. Change in security settings for system.

III. Giving permission to User to work with database or table

IV. Quering a Database .

V. Insertion/modification/deletion of data..... etc

Elements of MySQL

- Main elements of MySQL are-
  - *Literals*
  - *Datatypes*
  - *Nulls*
  - *Comments*
- **Literals** : generally known as fixed data value. It can be character, numeric or text literal.
- **Data Types** : these are of following types-
  - (i) **Numeric** :INT, TINYINT, SMALLINT, MEDIUMINT, FLOAT , DOUBLE etc
  - (ii) **Date and Time** : DATE, DATETIME, TIME, YEAR etc
  - (iii) **String type** : CHAR, VARCHAR, BLOB or TEXT TINYTEXT, ENUM etc

Elements of MySQL

- **NULL Values** : if a column in a row has no value, then column is said to be null. NULL can appear in a column of any data type provided they are not restricted by NOT Null or Primary Key.

- **Comments** : it is a text which does not execute. Comments are used only for documentation purpose. Three types of comments are-
- /* comment */
- -- comment
- # comment

SQL COMMAND SYNTAX

- SQL provides a set of predefined commands.
- SQL syntax has following elements-
- Keyword:
  - Statements:
  - Clauses:
  - Arguments:
- SQL commands are not case sensitive.

Creation of Database

- Following command is used to create a Database

```
mysql> CREATE DATABASE <database name >;
```

For ex-

```
mysql> create database school;
```

Using Database

- Following command is used to use a Database

mysql> USE <database name >;

For ex -

mysql> USE school;

A message will come saying- “database changed”

C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin\mysql.exe

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.33-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE DATABASE SCHOOL;
Query OK, 1 row affected (0.19 sec)

mysql> use school
Database changed
mysql>
```

See the Commands carefully

Table Creation

- To create a table in Database, following command is used-

```
mysql> CREATE TABLE <Table Name> (<Col1>  <DataType(Size)>,
                                  <Col2><DataType(size)>, . . . );
```

For ex-

mysql>create table **student** (Roll INT(4) Primary Key, Name CHAR(20),
(Age INT(2), City CHAR(10)) ;

A message will come saying- “Query OK”

```
mysql> Create table Student(Roll INT(4) Primary key, Name CHAR(20),
    -> Age INT(2), City CHAR(10));
Query OK, 0 rows affected (0.35 sec)
```

Primary key restrict a column to have unique values only.

Viewing Table structure

- To see structure of a table in Database, following command is used-

```
mysql> DESC <TableName>;
```

For ex -

mysql>DESC Student;

It displays whole structure of the table-

```
mysql> DESC Student;
+--------+-----------+------+-----+---------+-------+
| Field  | Type      | Null | Key | Default | Extra |
+--------+-----------+------+-----+---------+-------+
| Roll   | int(4)    | NO   | PRI | NULL    |       |
| Name   | char(20)  | YES  |     | NULL    |       |
| Age    | int(2)    | YES  |     | NULL    |       |
| City   | char(10)  | YES  |     | NULL    |       |
+--------+-----------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Modification in Table structure

- To modify structure of a table in Database, following command is used-

mysql>ALTER TABLE <Table name> ADD/MODIFY (<Col> <type(size)>, ...)

For ex-

mysql> Alter Table Student Add (class INT(2));

A message comes saying "Query OK" .

```
mysql> alter table student add (class INT(2));
Query OK, 0 rows affected (0.42 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Again run the DESC command-

```
mysql> DESC Student;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| Roll  | int(4)   | NO   | PRI | NULL    |       |
| Name  | char(20) | YES  |     | NULL    |       |
| Age   | int(2)   | YES  |     | NULL    |       |
| City  | char(10) | YES  |     | NULL    |       |
| class | int(2)   | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```

- A new column has been add.
- Columns can be added.
- Column size can be changed.

Dropping a Table

- To drop a table in Database, following command is used-

mysql> DROP Table <Table Name>;

For ex -

mysql>drop table <Student>

A message will come saying- “Query OK” now if you want to see the structure of the table you cant see because it has already been deleted.

```
mysql> drop table student;
Query OK, 0 rows affected (0.03 sec)

mysql> desc student;
ERROR 1146 (42S02): Table 'school.student' doesn't exist
mysql>
```

Data Integrity by Constraints

- The checks or conditions applied on one or more columns of a table are known as ***CONSTRAINTS .***
- These are set to maintain integrity in a table hence also known as integrity constraints.
- When a constraint is applied on a table, all the data should follow this constraint.
- Constraints are to be set at the time of table creation so that it should be followed at the time of data insertion. Syntax is -

```
mysql> CREATE TABLE <TableName>
                (<Col1> <type(size)> <Column Constraint>,
                 <Col2> <type(Size)> <Column Constraint>, . . . . );
```

Integrity Constraints

- Constraints maintains the integrity of a database. Some of the constraints are-
- *Unique Constraint* : This constraint ensure that all the data of the column should be unique. It allows null values.
- *Primary key Constraint* : This constraints is used to uniquely identify data. It does not accept null values.
- *Default Constraint* : This constraint is used to set a default values in case no value is provided by user for a column.
- *Check Constraint* : This constraint is used to set a limit for a column. for ex- no data should be inserted less than 20 in age column.
- *Foreign key Constraint* : it is a non –key attribute of one table derived from primary key from other table.

Creation of a Table

```
mysql> create table student
        (Roll INT(4) PRIMARY KEY,
        NAME CHAR(20) NOT NULL,
        Age INT(2) CHECK (age>5),
        Class CHAR(3) Default 'I',
        City CHAR(10));
```

This column will be primary key.

This column will not accept null.

This column will not accept age less than 5.

It will store 'I' in case of no value inserted.

Such constraints are known as column level constraints.

```
mysql> CREATE TABLE Student
    -> (ROLL INT(4) PRIMARY KEY,
    -> NAME CHAR(20) NOT NULL,
    -> AGE INT(2) CHECK(AGE>5),
    -> CLASS CHAR(3) DEFAULT 'I',
    -> CITY CHAR(10));
Query OK, 0 rows affected (0.09 sec)
```

Create table command

```
mysql> DESC Student;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| ROLL  | int(4)   | NO   | PRI | NULL    |       |
| NAME  | char(20) | NO   |     | NULL    |       |
| AGE   | int(2)   | YES  |     | NULL    |       |
| CLASS | char(3)  | YES  |     | I       |       |
| CITY  | char(10) | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

DESC command

Table level constraints Setting

```
mysql> create table student
        (Enroll INT(4),
         Roll INT(4) ,
        NAME CHAR(20) NOT NULL,
        Age INT(2),
        Class CHAR(3) Default 'I',
        City CHAR(10),
        PRIMARY KEY (Enroll, Roll));
```

These constraints are known as Table level constraints.

Insertion of a record in Table

Syntax to insert a record in a Table is-

mysql> INSERT INTO <TableName> (<Col1> <Col2> <Col3> <Col4>
VALUES (<val1>,<val2>,<val3>,<val4>,. . .);

```
mysql> INSERT INTO STUDENT (ROLL, NAME, AGE, CLASS, CITY) VALUES
    -> (1001,'Pankaj',6,'I','Barabanki');
Query OK, 1 row affected (0.03 sec)
```

We can change the order of columns as-

```
mysql> INSERT INTO STUDENT (ROLL, NAME, CLASS,AGE, CITY) VALUES
    -> (1002,'Naresh','II',8,'Kanpur');
Query OK, 1 row affected (0.01 sec)
```

Here, we can insert values without specifying column names provided the order of values for columns should be same as in table.

```
mysql> INSERT INTO STUDENT VALUES
    -> (1003,'Sunita',7,'II','Barabanki');
Query OK, 1 row affected (0.03 sec)
```

Insertion of a record in Table

The columns in which you are inserting values will have the values. The columns previously set with default vales will have Default value. Other columns will have null .

```
mysql> INSERT INTO STUDENT (ROLL, NAME, AGE) VALUES
    -> (1004,'Meera',9);
Query OK, 1 row affected (0.05 sec)
```

Displaying records of the Table

```
mysql> Select * from Student;
+------+---------+------+--------+------------+
| ROLL | NAME    | AGE  | CLASS  | CITY       |
+------+---------+------+--------+------------+
| 1001 | Pankaj  |    6 | I      | Barabanki  |
| 1002 | Naresh  |    8 | II     | Kanpur     |
| 1003 | Sunita  |    7 | II     | Barabanki  |
| 1004 | Meera   |    9 | I      | NULL       |
| 1005 | Nita    |    7 | I      | NULL       |
+------+---------+------+--------+------------+
5 rows in set (0.00 sec)
```

No constraint was set for city hence it got null.

These values were set by default constraints

Updating a record in Table

Syntax to update a record in a Table is-

```
mysql> UPDATE <TableName>  SET <ColName>=<NewValue>
                                WHERE  <Condition>
```

```
mysql> Select * from Student;
+------+---------+------+-------+------------+
| ROLL | NAME    | AGE  | CLASS | CITY       |
+------+---------+------+-------+------------+
| 1001 | Pankaj  |    6 | I     | Barabanki  |
| 1002 | Naresh  |    8 | II    | Kanpur     |
| 1003 | Sunita  |    7 | II    | Barabanki  |
| 1004 | Meera   |    9 | I     | NULL       |
| 1005 | Nita    |    7 | I     | NULL       |
+------+---------+------+-------+------------+
5 rows in set (0.00 sec)
```

In this table, age of meera is to be set 6. and city of roll 1004 and 1005 is to be set as Lucknow.

```
mysql> update student set age=6 where roll=1004;
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update student set city ='Lucknow'
    -> where roll in (1004,1005);
Query OK, 2 rows affected (0.03 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

age is changed by the command.

City is changed by the command. |

Deletion of a record from a Table

- Syntax to delete a record from a Table is-

mysql> DELETE FROM <TableName> WHERE <Condition>

```
mysql> Delete from Student where roll in (1001,1004);
Query OK, 2 rows affected (0.03 sec)

mysql> select * from Student;
+------+---------+------+-------+-----------+
| ROLL | NAME    | AGE  | CLASS | CITY      |
+------+---------+------+-------+-----------+
| 1002 | Naresh  |    8 | II    | Kanpur    |
| 1003 | Sunita  |    7 | II    | Barabanki |
| 1005 | Nita    |    7 | I     | Lucknow   |
+------+---------+------+-------+-----------+
3 rows in set (0.00 sec)
```

Viewing records after deletion.

To delete all records from a table, following command will be used-

```
mysql> delete from student;
Query OK, 3 rows affected (0.03 sec)

mysql> select * from student;
Empty set (0.00 sec)
```

Accessing a Table

Syntax to access Data from a table is-

```
mysql> SELECT <Col Names> FROM <Table Name>
                                WHERE <Condition>
```

```
mysql> Select * from Student;
+------+--------+------+-------+-----------+
| ROLL | NAME   | AGE  | CLASS | CITY      |
+------+--------+------+-------+-----------+
| 1001 | Pankaj |    6 | I     | Barabanki |
| 1002 | Sunita |    7 | II    | Barabanki |
| 1003 | Mukesh |    5 | I     | Barabanki |
| 1004 | Ramesh |    8 | II    | Lucknow   |
| 1005 | Yogesh |    9 | III   | Kanpur    |
+------+--------+------+-------+-----------+
5 rows in set (0.00 sec)
```

Here * means all columns and without condition it will displays all records.

```
mysql> Select * from Student Where city='Barabanki';
+------+--------+------+-------+-----------+
| ROLL | NAME   | AGE  | CLASS | CITY      |
+------+--------+------+-------+-----------+
| 1001 | Pankaj |    6 | I     | Barabanki |
| 1002 | Sunita |    7 | II    | Barabanki |
| 1003 | Mukesh |    5 | I     | Barabanki |
+------+--------+------+-------+-----------+
3 rows in set (0.00 sec)
```

Here only those records will display where city is Barabanki.

Accessing a Table

Syntax to access Data from a table is-

```
mysql> SELECT <Col Names> FROM <Table Name>
                                WHERE <Condition>
```

```
mysql> Select Name, Class from Student where City <> 'Barabanki';
+--------+-------+
| Name   | Class |
+--------+-------+
| Ramesh | II    |
| Yogesh | III   |
+--------+-------+
2 rows in set (0.00 sec)
```

Here Name and class of only those records are displayed which are not from Barabanki.

```
mysql> Select Name, Roll, City, Age, Class
    -> from Student;
+---------+------+------------+------+-------+
| Name    | Roll | City       | Age  | Class |
+---------+------+------------+------+-------+
| Pankaj  | 1001 | Barabanki  |    6 | I     |
| Sunita  | 1002 | Barabanki  |    7 | II    |
| Mukesh  | 1003 | Barabanki  |    5 | I     |
| Ramesh  | 1004 | Lucknow    |    8 | II    |
| Yogesh  | 1005 | Kanpur     |    9 | III   |
+---------+------+------------+------+-------+
5 rows in set (0.00 sec)
```

Here columns have been rearranged.

Distinct keyword

```
mysql> select DISTINCT City from Student;
+------------+
| City       |
+------------+
| Barabanki  |
| Lucknow    |
| Kanpur     |
+------------+
3 rows in set (0.00 sec)
```

Here DISTINCT keyword is used to remove duolicacy from city column. With this we can get an dea of total cities in the table.

Viewing Tables in a Database

```
mysql> show tables;
+-------------------+
| Tables_in_school  |
+-------------------+
| student           |
+-------------------+
1 row in set (0.00 sec)
```

Displays all tables in a Databse.

Pattern Matching

```
mysql>
mysql> Select * from Student Where
    -> name like 'S%';
+------+--------+------+-------+-----------+
| ROLL | NAME   | AGE  | CLASS | CITY      |
+------+--------+------+-------+-----------+
| 1002 | Sunita |    7 | II    | Barabanki |
+------+--------+------+-------+-----------+
1 row in set (0.00 sec)
```

With 'like' two symbols are to be used '%' and '_'. '%'represent multiple characters whereas '_' represents one charachetr .

In above example all the names starting with 'S' are shown.
In example given below all the names having 'u' as second character are shown.

```
mysql> Select * from Student Where
    -> name like '_u%';
+------+--------+------+-------+-----------+
| ROLL | NAME   | AGE  | CLASS | CITY      |
+------+--------+------+-------+-----------+
| 1002 | Sunita |    7 | II    | Barabanki |
| 1003 | Mukesh |    5 | I     | Barabanki |
+------+--------+------+-------+-----------+
2 rows in set (0.00 sec)
```

Creation of Table from another Table

Syntax for creation of a table from another table is -

```
mysql>CREATE TABLE <TableName>
                AS (SELECT <Cols> FROM <ExistingTable>
                WHERE <Condition>);
```

```
mysql>
mysql> create table bbkscl as (select name, age from student
    -> where city='Barabanki');
Query OK, 3 rows affected (0.09 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> select * from bbkscl;
+---------+-------+
| name    | age   |
+---------+-------+
| Pankaj  |     6 |
| Sunita  |     7 |
| Mukesh  |     5 |
+---------+-------+
3 rows in set (0.00 sec)
```

See the example carefully

Other SQL Commands

- Select * from Student where city *in* (‘Jaipur’,’Ajmer’);
- Select * from Student where city *Not in* (‘Jaipur’,’Ajmer’);
- Select * from Student where age *between* 5 *and* 7;
- Select * from Student *Order by* name *DESC* ;
- Select *5 * 6* from *DUAL* ;
- Select *avg(age)* from student; //Similarly count etc functions

Thank you

Please follow us on our blog

www.pythontrends.wordpress.com