

My SQL

In our day to day life we use some kind of data. Banks, Universities , Libraries are some of the organisations which require to deal with huge data. The voluminous data when systematically arranged , is known as database.

Database in My SQL contains one or more tables.

A table consists of rows and columns. Columns are also known as fields.

Each column represents a particular type of information eg. Roll Number , Name , Address etc.

Rows contain different type of information.

Rows are also known as records.

Columns / Fields

Roll No	Name	Address	Year	Course
1001	Mr. Aniket	Mumbai	2015-16	BMS
1002	Mr. Jayant	Pune	2016-17	BBI
1003	Mr. Deep	Mumbai	2016-17	BMS
1004	Mr. Jitesh	Pune	2015-16	BFM

Rows / Records

- Database is a collection of related tables
- Table consists of records
- Record consists of fields

Rules of database name

- Maximum 64 characters long
- Can not be reserve word like SELECT , FROM , DISCRIBE etc.
- Can not contain special characters such as / , \ , ; , . Etc
- May contain blank space in between but not at the beginning or at end.
- It should be unique
- It is case sensitive ie STUDENT , Student , student all 3 are different

Data Types

- My SQL permits 4 types of data : Character , Numeric , Boolean , Temporal

Character / String Data

- Data that can not be used for mathematical calculations are called character type of data.
- e.g. Name , Address , Tel Number , Pin code

Storing Character type data

- CHAR(width) – Data is sored as fixed length string . Maximum width is 255 bytes
- eg. Name CHAR(10) will reserve 10 bytes to store Name. Even if Name is Raju, it will reserve 10 bytes.
- VARCHAR(width) – Data is sored as variable length string. It uses as many bytes as actual size of data. Maximum width is 64 KB
- Here if Name is Raju, it will take space of 4 bytes.
- TEXT – It is commonly used to store string , maximum up to 64 KB
- MEDIUMTEXT – It is commonly used to store medium sized string , maximum up to 16 MB
- LONGTEXT – It is commonly used to store long string , maximum up to 4 GB

Storing Numeric type data

- Numeric data is the data which can be used for mathematical calculations. Eg Marks , Salary , Hours , Rate etc.
- They can be
- integer (smallint , Bigint) where decimal point is not allowed
- Fractional (float , double , decimal) which support use of decimal point

Column Type	Storage size
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
BIGINT	8 bytes
FLOAT(w,d)	4 bytes
DOUBLE(w,d)	8 bytes
DECIMAL(w,d)	4 bytes for every 9 digits

W is the total width , d is number of digits after decimal point

Unsigned & Zerofill

- Unsigned – This argument can be used only with numeric data columns. It means no sign. In other words value can not be negative
- Zerofill - This argument can be used only with numeric data columns. It attaches zeros on the left hand side of the number upto specified length.
- Zerofill automatically adds unsigned declaration.
- MARKS SMALLINT(3) ZEROFILL
- Here if marks are 57 , it will be stored as 057

Boolean Type of data are represented by TRUE or FALSE

Temporal Type of data are used to represent Date and Time

Type	Format	Storage
DATE	YYYY-MM-DD	3 bytes
YEAR	YYYY	1 byte
TIME	HH:MM:SS	3 bytes
DATETIME	YYYY-MM-DD HH:MM:SS	8 bytes

Column Specifications

- NOTNULL means the column doesn't contain any empty (blank)value.
- PRIMARY KEY- Every table has one column whose value uniquely identifies each row in the table.

This column is called primary key of the table.

It automatically adds NOT NULL to the primary key column.

- AUTO INCREMENT – This can be used with numeric column and primary key column only.
- This will add 1 to the maximum positive value stored. By default it starts with 1 .
- DEFAULT – If value is not specified , it uses default value
- UNIQUE – It makes sure that value in the column are unique. This means no value can be repeated.
- ENUM – It is used to set a value from specified list. A list can contain maximum 65,535 values
- CITY ENUM(“Mumbai”, “Pune” , “Nagpur”)
- GRADE ENUM(1 , 2 ,3 , 4)

Starting My SQL

- Start → All Programs → My SQL

My SQL Command Line Client

- Each command should be typed at mysql > prompt and end with ;
- Statements are not case sensitive
- A command need not be typed in a single line. Several lines can be used to enter a command by pressing enter key. It shows prompt -- > on the next line, where we can continue typing
- All keywords will be UPPERCASE and all user defined words will be in lowercase.

CREATE

- CREATE – It is used to create a database with specific name.

```
mysql> CREATE DATABASE University;
```

- USE – It makes the database active

```
mysql> USE University;
```

CREATE TABLE– It is used to create a table with required specifications.

CREATING TABLES

Table employee

Information	Column Name	Data Type
-------------	-------------	-----------

Employee number	emp_no	smallint (unsigned)
-----------------	--------	---------------------

First name	f_name	varchar(15)
------------	--------	-------------

Last name	l_name	varchar(15)
-----------	--------	-------------

Job code	j_code	smallint (unsigned)
----------	--------	---------------------

Hire date	h_date	date
-----------	--------	------

Salary sal smallint (unsigned)

Mysql> CREATE TABLE employee

```
-> ( emp_no SMALLINT UNSIGNED ,  
- > f_name VARCHAR (15) ,  
- > l_name VARCHAR (15) ,  
- > j_code SMALLINT  UNSIGNED ,  
- > h_date DATE ,  
- > Sal SMALLINT UNSIGNED);
```

Important points for Create command

Command should be typed at Mysql> prompt only

Create table employee – word table is compulsory

Do not give , or ; after Create table employee just strike the enter key

It gives you -> Start with (give the list of columns after description of every column
give , after description of last column end with);

Table department

Information	Column Name	Data Type
Department number	dept_no	smallint (unsigned)
Department name	dept_name	varchar(15)
Job type	j_type	varchar(15)
Job code	j_code	smallint (unsigned)

Mysql> CREATE TABLE department

```
-> ( dept_no SMALLINT UNSIGNED ,  
- > dept_name VARCHAR (15) ,  
- > j_type VARCHAR (15) ,  
- > j_code SMALLINT UNSIGNED );
```

Table student

Information	Column Name	Data Type
Student name	s_name	varchar(15)
Address	s_adrs	varchar(25)
Birthdate	s_bod	date
Gender	s_gender	Character 1
Class	s_class	varchar (15)
Marks	s_marks	Smallint

mysql> CREATE TABLE student

```
-> ( s_name VARCHAR (15) ,  
    -> s_adrs  VARCHAR (25) ,  
    -> s_bod   DATE ,  
    -> s_gender CHAR(1) ,  
    -> s_class VARCHAR(15) ,  
    -> s_marks SMALLINT UNSIGNED) ;
```

SHOW

- SHOW command is used to get information about databases , tables , columns etc.

My sql> SHOW DATABASES;

Will display names of existing databases.

My sql> USE University ;

My sql> SHOW TABLES;

Will display names of existing tables under database University.

It shows table names employee, department , student

My sql> SHOW COLUMNS FROM employee;

It shows information about all the columns in the specified table.

DESCRIBE

It displays the structure of the specified table.

My sql> DESCRIBE employee;

Its output is same as that of

```
My sql> SHOW COLUMNS FROM employee;
```

INSERT

It is used to enter the data in a table.

It consists of 3 parts such as

- Name of the table
- Field names / column names in to which data is to be added
- Actual values to be added in respective columns
- My Sql > INSERT INTO department (dept_no, dept_name, j_type , j_code) VALUES (225, 'SALES' , 'CLERK' , 5);
- Note that String values are enclosed in quotation marks,
- Numbers are written as they are ,
- Dates are written in yyyy-mm-dd form

Writing column names in INSERT statement is optional

```
My Sql > INSERT INTO department VALUES (225, 'SALES' , 'CLERK' , 5);
```

```
My Sql > INSERT INTO employee VALUES
```

```
( 1500 , 'Nita' , 'Thakur' ,5, 2011-03-15 , 20000);
```

It means emp_no = 1500 , f_name = Nita , l_name = Thakur , j-code=5 , h_date = 2011-03-15 & Sal = 20000

We can add data for more than 1 record

```
My Sql > INSERT INTO employee VALUES
```

```
( 1500 , 'Nita' , 'Thakur' ,5, 2011-03-15 , 20000),
```

```
( 1200 , 'Ram' , 'Kale' ,3, 2012-02-11 , 15000),
```

```
( 1307 , 'Jitesh' , 'Shah' ,6, 2005-08-10 , 28000);
```

My Sql > INSERT INTO student VALUES

('Jaya K.' , 'Mumbai' , 2000-03-15 , 'F' , 'FYBCom' , 85);

My Sql > INSERT INTO student VALUES

('Preeti S.' , 'Mumbai' , 2000-06-15 , 'F' , 'FYBCom' , 78),

('Jeevan P.' , 'Mumbai' , 1999-06-17 , 'M' , 'SYBCom' , 67),

('S.'B. Kale , 'Mumbai' , 1998-02-15 , 'M' , 'TYBCom' , 80);

SELECT

- It displays contents of table in desired way- entire records, only specified columns , records meeting certain condition etc.
- My Sql > SELECT * FROM student;

Will display all the records in table student

- My Sql > SELECT s_name , s_class , s_marks FROM student;

Will display s_name , s_class , s_marks from table student

UPDATE

- It is used to make changes in data values. It can be applied to data in a particular column for all records or to change contents of columns of records which satisfy given condition

My Sql > UPDATE department

-> SET dept_name = lower(dept_name);

This will change contents of dept_name in lower case for all records

WHERE Clause

- It allows to choose rows which match the given condition.
- It can be used with UPDATE , DELETE , SELECT etc.
- Change the address as 'DADAR' & date of birth as 5th March 1990 for a student with name 'Karan' in table student

- My Sql > UPDATE student

-> SET s_bod = '1990_03_05', s_adrs = 'DADAR' WHERE s_name = 'Karan' ;

- Change Class and marks of KINJAL BHOME to FYBMS and 73 respectively

My Sql > UPDATE student

-> SET s_class = 'FYBMS', s_marks = 73 WHERE s_name = 'KINJAL BHOME' ;

- Change Class of SUCHETA K. to FYBMS

My Sql > UPDATE student

-> SET s_class = 'FYBMS' WHERE s_name = 'SUCHETA K.' ;

- Increase Salary of employee hired before 1st June 2007 by 10%

My Sql > UPDATE employee

-> SET sal = 1.1 * sal WHERE h_date <= '2007-06-01' ;

- Set 1 as job code for all managers

My Sql > UPDATE department

-> SET j_code = 1 WHERE j_type = 'MANAGER' ;

DELETE

- It is used to delete a few records which satisfy the condition

My Sql > DELETE FROM employee where j_code =3;

All records with j_code 3 will be entirely deleted.

- Write SQL command to delete records of employee hired after 1st January 2010 in the table employee.

My Sql > DELETE FROM employee where h_date > '2010-01-01' ;

DROP

- It is used to delete a columns from a table , tables from a database or even entire database.

My Sql > DROP TABLE employee ;

My Sql > DROP DATABASE university ;

RENAME

- The databases , table names can be changed by using RENAME statement.

My Sql > RENAME TABLE employee to new_employee ;

My Sql > RENAME TABLE student to ty_student;

My Sql > RENAME TABLE department to department _1 ;

ALTER

- It is used to modify structure of a table such as adding new columns , deleting existing columns , changing column names , their data type , width etc.
- Keywords such as CHANGE , MODIFY , ADD , DROP , RENAME etc are used in ALTER command to get the required results.

CHANGE

To change name , width of existing columns

My Sql > ALTER TABLE student

-> CHANGE s_adrs s_address

->VARCHAR(30);

This will rename column s_adrs as s_address as well as change the width to variable length 30. Here we don't have to specify the old width.

MODIFY

- If data type or other details of column name are to be changed retaining the column name, MODIFY is used.

My Sql > ALTER TABLE student

-> MODIFY s_class VARCHAR(12);

My Sql > ALTER TABLE employee

-> MODIFY sal INT;

ADD

- It is used to add new columns to the table.

My Sql > ALTER TABLE student

-> ADD bd_group VARCHAR(10);

My Sql > ALTER TABLE department

-> ADD week_prod VARCHAR(10);

My Sql > ALTER TABLE employee

-> ADD emp_edu VARCHAR(10);

DROP

- Used to delete existing columns , table.

My Sql > ALTER TABLE student

-> DROP bd_group;

My Sql > ALTER TABLE department

-> DROP week_prod;

My Sql > ALTER TABLE employee

-> DROP emp_edu ;

@@@@@@@@@@@@