

Birla Institute of Technology & Science, Pilani, K. K. BIRLA Goa campus
Database Systems (CS F212)
Second Semester 2021-2022
Lab-1
To study DDL and DML commands.

Introduction: SQL stands for Structured Query Language. It was invented & developed by IBM in the early 1970's. SQL is a standard language common to all relational databases. It is actually a database language. A table is a primary database object of SQL.

For the Database Lab we will be using the MySQL database.

Through MYSQL we can :

- enter, edit, store, retrieve and run SQL commands and PL/SQL blocks.
- format, perform calculation, store, & print query results.
- list column definitions for any table.
- access and copy data between SQL databases.
- send/accept messages from/to the end user.

With SQL we can:

- Create tables in the database
- Store data
- Retrieve data
- Change data and change the structure of the underlying tables
- Combine and calculate data
- Provide security

The aim of this reading material is to introduce you to the syntax and functionality of different SQL commands. It assumes that you already have installed MySQL on your machine. The first thing that we need to do is to log into MySQL.

Once logged in you will be able to run SQL queries on the MySQL command prompt.

Before proceeding to different types of SQL commands we will look at four commands:

1. show databases;

This command shows different databases that are present in the mysql server. If you have not logged in as a root user, then some databases may not be shown. A database can contain several different tables.

2. create database <xyz>;

Now we are ready to learn the different types of SQL commands.

Types of SQL commands:

Data Definition Language (DDL) Commands

CREATE TABLE
CREATE INDEX
CREATE VIEW
DESC
ALTER TABLE
DROP DATABASE
DROP TABLE
TRUNCATE

Data Manipulation Language (DML) Commands

DELETE
INSERT
UPDATE

Data Query Language (DQL) Commands

SELECT

Conventions used:

Keywords in syntax are shown in bold letters.

Words in syntax between '<' and '>' signify the name of the table or column given by the user.

Words in syntax between '[' and ']' are optional.

NOTE: It is healthy to make a habit of writing your SQL command in an editor (such as gedit), and then copy and paste by right-clicking on MySQL prompt.

DDL COMMANDS

Creating Database:

Syntax to create a database:

create database <database name>;

Example:

mysql> create database mydb;

Listing Databases:

Syntax:

show databases;

Using Database:

Syntax to use a database:

use <database name>;

Example:

mysql> use mydb;

Listing tables in current database:

Syntax:

show tables;

NOTE: If you are getting a syntax error when executing these commands (after copy pasting), check the quotes. Due to formatting on Word, MySQL throws an error for quotes. Retype the quotes and it should work.

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use...

Creating tables:

Syntax to create a table

create table <table name> (column definition 1, column definition 2, ,);

Example:

```
mysql> CREATE TABLE students (id CHARACTER (12),  
-> name VARCHAR(30),  
-> hostel INTEGER NOT NULL,  
-> percentage DECIMAL(5,2) DEFAULT 0.0,  
-> phone INT,  
-> bdate DATE,  
-> gender ENUM('F','M'),  
-> CONSTRAINT uk UNIQUE(name));
```

(OR)

```
CREATE TABLE students (id CHAR(12), name VARCHAR(30) UNIQUE, hostel  
INT NOT NULL, percentage DECIMAL(5,2) DEFAULT 0.0, phone INT UNIQUE, bdate  
DATE, gender ENUM('F','M'));
```

Check using 'show tables' command.

NOTE: 1) Every SQL statement ends with a semicolon.

2) SQL keywords are not case sensitive, but the data in the table and the name of the table are case sensitive.

3) It is a good practice to write the query with line separator to make error detection more convenient.

There are various inbuilt data types like char (for fixed-length character string to store alphanumeric values), varchar (for variable-length character strings to store alphanumeric values), integer (for integer values), decimal (to store floating-point values), date (for date and time) and others like RAW, Long RAW, LOB, BLOB, CLOB, Bfile for audio and video files.

//Explore big int

Syntax to add constraints like primary key, unique, not null and check while creating a table:

create table <table name> (column definition 1, column definition 2, , **primary key**(column name));

Example:

```
mysql> CREATE TABLE gradstudents (id CHARACTER (12), name VARCHAR(30), hostel  
INTEGER NOT NULL, percentage DECIMAL(5,2) DEFAULT 0.0, phone INT, bdate DATE,  
gender ENUM('F','M'), CONSTRAINT ue UNIQUE (phone),PRIMARY KEY (id) );
```

(OR)

```
mysql> CREATE TABLE gradstudents (id CHARACTER (12) PRIMARY KEY, name  
VARCHAR(30), hostel INTEGER NOT NULL, percentage DECIMAL(5,2) DEFAULT 0.0,  
phone INT, bdate DATE, gender ENUM('F','M'),CONSTRAINT ue UNIQUE (phone));
```

TRY THIS (justify if you get errors):

```
mysql> CREATE TABLE temp (id char(10), name varchar(30) unique, hostel int, percentage  
decimal(5,2) default 0.0, phone VARCHAR(11), bdate date, gender enum('F','M'), primary  
key(id), primary key(name) );
```

Correct method to create a composite primary key:

```
mysql> CREATE TABLE temp (id char(10), name varchar(30) unique, hostel int, percentage  
decimal(5,2) default 0.0, phone varchar(11), bdate date, gender enum('F','M'), primary  
key(id,name) );
```

Describe Table:

Syntax to view the table structure

desc <table name>;

Example:

```
mysql> desc students;
```

```
mysql> desc gradstudents;
```

Why is the NULL column of the 'id' row in 'students' table YES? Why is it NO in the case of the gradstudents table?

To alter a table:

Syntax to alter the table structure:

alter table <table name> **modify** <column name> <column definition>;

alter table <table name> **rename** <new table name>;

alter table <table name> **add column** (column definition);

alter table <table name> **change** <old column name> <new name> <old column definition>;

alter table <table name> **drop column** <column name>;

alter table <table name> **add constraint** <constraint name> (condition);

alter table <table name> **drop constraint** <constraint name>;

Example:

mysql> alter table students modify name varchar (40); *//to increase the size.*

mysql> alter table students modify name varchar (10); *//to decrease the size.*

mysql > alter table students add column address varchar (25); *//to add new column.*

mysql> alter table students change address postaladdress varchar (25); *//to rename a column.*

mysql > alter table students drop column postaladdress; *//to remove a column.*

mysql > alter table students modify phone varchar(10); *//changing type, to accept realistic phone numbers.*

mysql > alter table gradstudents drop index ue; *//to remove unique constraint on phone.*

Check the result of the above queries by 'desc' command.

DML COMMANDS

Insert data:

Syntax to insert a row in a table:

insert into <table name> [field names] **values** (a list of data values);

Examples:

mysql> INSERT INTO students VALUES ('PS99305017', 'Mohan Sharma', 13,76.23, '9800000002', '2001-03-15', 'M');

(OR)

mysql> INSERT INTO students (id, name, hostel, percentage, phone, bdate, gender) VALUES ('PS99305017', 'Sai Sundar', 11, 77.23, '9800000001', '2001-01-25', 'M');
 //(Date format is 'YYYY-MM-DD')

TRY THIS (justify if you get errors):

mysql > INSERT INTO students VALUES ('Sai Sundar', 'PS99305017', 11, 77.23, '2001-01-25', '9800000001', 'M');

Correcting this:

(We cannot enter duplicate values of phone into the table, as we have defined it to be unique)

mysql > INSERT INTO students (name, id, hostel, percentage, bdate, phone, gender) VALUES ('Jay Singh', 'PS99305012', 11, 83.73, '2000-07-04', '9900000002', 'M');

```
/* to violate unique constraint by inserting same name to another student*/
mysql> INSERT INTO students VALUES ('PS99305018','Sai
Sundar',11,90.23,'9800000001','2001-01-25','M');
```

```
/* to insert blank in primary key column by giving blank id to a student*/
mysql> INSERT INTO students VALUES ('','Shyam Sundar',11,90.23,9800000004,'2001-01-
25','M');
```

```
/* to insert blank in column having unique constraint by giving blank name to a student*/
mysql> INSERT INTO students VALUES ('PS99305018', NULL, 11, 90.23, '9800000009',
'2001-01-25','M');
mysql> INSERT INTO students VALUES ('PS99305018','', 11, 90.23, '9800000009', '2001-01-
25','M');
```

Try running both commands twice. As you can see, NULL does not count as a repeated value.

```
/* to violate not null constraint by inserting NULL in hostel column*/
mysql >INSERT INTO students VALUES ('PS99305020','Sundaram',NULL,90.23,'9800000005',
'2001-01-25','M');
```

```
/* to miss single quote in char or varchar data type);*/
mysql> INSERT INTO students VALUES (PS99305020,
'Sundaram',11,90.23,9800000006,'2001-01-25','M');
```

```
/*to exceed the size of an attribute(name here)*/
mysql > INSERT INTO students VALUES ('PS99305021','Ram Prabhu Sundaran',11,90.23,
'9800000006','2001-01-25','M');
```

```
/*to exceed date limit*/
mysql> INSERT INTO students VALUES ('PS99305023','Ramnarayan
Sundaran',11,90.23,'9800000006','2001-02-30','M');
```

```
/*to violate enum datatype */
mysql> INSERT INTO students VALUES ('PS99305025', 'Narayan
Sundar',11,90.23,'9800000007','2001-02-16','K');
```

DQL COMMAND

Syntax of select command:

- *to display all rows*
select * from <tablename>;

Example:

```
mysql>SELECT * FROM students;
```

Observe the bdate of student with id PS99305023, it is the default value.

Observe percentage of student with id PS99305019, it is 0.0, the default value set by user.
Observe value in gender column of student with id PS99305025, its blank.

- *to display particular column(s)*

select column name **from** table name;

Example:

```
mysql > SELECT id FROM students;
```

```
mysql > SELECT id, name FROM students;
```

- *to display distinct rows*

select distinct column name **from** tablename;

Example:

```
mysql > SELECT DISTINCT name from students;
```

Update Query

Syntax of update command

update tablename **set** field=value, ... **where** condition;

Example:

```
mysql > UPDATE students SET percentage=90.46 WHERE id='PS99305018';
```

```
mysql > UPDATE students SET name ='Sham' WHERE name ='Sai Sundar';
```

```
mysql > UPDATE students SET hostel=hostel*10;
```

Delete Query:

Syntax of delete command

delete from <table name> **where** condition(and/or conditions);

Example: to delete record of a particular student

```
mysql > delete from students where id='PS99305018';
```

Display system date:

```
mysql > SELECT sysdate() from dual;
```

Some more DDL Commands

Enter some records in gradstudnets table and try following commands

Truncate the table: (Used to delete all records from a table)

Syntax to truncate a table:

truncate table <table name>;

Example:

```
mysql> truncate table gradstudents;
```

```
mysql> select * from gradstudents; //to test test result of truncate
```

Dropping the database:

Syntax to drop a database:

drop database <database name>;

Example:

mysql> drop database mydb;

mysql> show databases; //to check status

Dropping the tables:

Syntax to drop a table:

drop table <table name>;

Example:

mysql> drop table gradstudents;

mysql> show tables; //to check status

Q. Find the difference between truncate and drop SQL command?

Try enum ('f', 'F', 'm', 'M') and study the error.