Structured Query Language (SQL)

SQL Commands

To understand the commands in SQL, let us consider a one question and will execute all possible commands and their constraints.

Question : Create a database by the name **school**. In the database **school** create two tables **student and fees.** The data as shown below as reference.

student ⇒ Table 1

Roll no	Name	Gender	Marks	DOB
1	PRATIK	M	95	22-Apr-2003
2	GARIMA	F	85	05-Mar-2004
3	KHUSHI	F	92	24-Jan-2003
4	BABU	NULL	NULL	15-Aug-2000
5	AMIT	M	60	26-Jan-2002
6	SHARMILA	F	90	12-Dec-2003

fees ⇒ Table 2

Roll no	Name	Fees	Attendance
1	PRATIK	5250.00	98
2	GARIMA	1250.00	70
4	BABU	1250.00	75
6	SHARMILA	5250.00	85

Structured Query Language (SQL)

1. Dispaly Databases: (SHOW DATABASES)

To get started on your own database by the name **school,** we can first **check** whether it is exist or not exist in MySQL server.

It is used to display all the databases which is already been created and stored in MySQL server by using the command **SHOW DATABASES**. .

Syntax: SHOW DATABASES;

Example:

mysql>SHOW DATABASES;

Result: Displayed all databases.

Note: If database school is not seen then proceed to point no. 2 else point no. 3

Structured Query Language (SQL)

2. Creating Databases: (CREATE DATABASES)

It is used to create a database in RDBMS by using **CREATE DATABASE** command. (*only first time*)

Syntax: CREATE DATABASE <database_name>;
Example:
mysql>CREATE DATABASE school;

Result: Creates database with the name school.

Note: To verify use the command show databases.
mysql> SHOW DATABASES;
it can be verified.

Structured Query Language (SQL)

3. Opening Databases: (USE)

Once a database is created, next time you need not to create once again just you need to open it to work on it by using **USE** command.

Syntax: USE <database_name>;

Example:

mysql>USE school;

Result: Database changed. i.e school database is opened.

Structured Query Language (SQL)

4. Removing Databases: (DROP DATABASES)

It is used to delete a database along with its tables inside by using **DROP DATABASE** command.

Syntax: DROP DATABASE <database_name>;

Example:

mysql>DROP DATABASE school;

Result: Database deleted. i.e school database is deleted.

Note: To verify use the command show databases.

mysql> SHOW DATABASES;

it can be verified.

Structured Query Language (SQL)

5. Createing a Table: (CREATE TABLE)

Assume database '**school**' is created and ready to work on it. Now next step is to create a table '**student**' and **'fees'.**
To create a table in a database use **CREATE TABLE** command.

Syntax: CREATE TABLE <table_name>
{
<column_name1><datatype>[(size)],
<column_name2><datatype>[(size)],
<column_name3><datatype>[(size)],

.....
<column_name5><datatype>[(size)],
};

Structured Query Language (SQL)

5. Cont…. (CREATE TABLE)

Example:

```
mysql> CREATE TABLE student
       {
         Rollno  int(2) PRIMARY KEY,
         Name  varchar(20)  NOT NULL,
         Gender  char(1),
         Marks  numeric(6,2),
         DOB  date
       };
```

Result: Table student is created.

Note: Similarly table fees also to be created.

```
mysql> CREATE TABLE fees
       {Rollno int(2), Fees decimal(6,2), Attendance  smallint };
```

Result: Table fees is created.

Structured Query Language (SQL)

6. Viewing a Tables: (SHOW TABLES)

To verify that the table has been created or not, **SHOW TABLES** command to be used.

Command **SHOW TABLES** is used to view/ display the name of the tables present in a database**.**

Syntax: SHOW TABLES ;

Example:

mysql>SHOW TABLES;

Result: Displayed all the tables present in the database.
i.e Both tables '**student'** and '**fees'** will be listed.

Structured Query Language (SQL)

7. Viewing a Table Structure: (DESCRIBE)

Command **DESCRIBE** is used to view/ display the structure of the table.

Syntax: DESCRIBE<table name> ;

Example:

mysql>DESCRIBE student;

Result: Displayed the structure of the table student as shown below:

Field	Type	Null	Key	Default	Extra
RollNo	int(2)	NO	PRI	NULL	
Name	varchar(20)	NO		NULL	
Gender	char(1)	YES		NULL	
Marks	numeric(6,2)	YES		NULL	
DOB	date	YES		NULL	
5 rows in set (0.01 sec)					

Structured Query Language (SQL)

8.1 Change in Table Structure: (ALTER)
(Adding a new column to an existing table.)

It is used to modify/ change the structure of the tables by using the command **ALTER.**

8.1 Adding a new column to an existing table.

Syntax: ALTER<table name>ADD(<column name><datatype>[(size)] ;

Example:

mysql>ALTER table student ADD (Mobile integer);

mysql>ALTER table student ADD (Address varchar(25));

Result: New column **Mobile and Address** is added into the table.

Structured Query Language (SQL)

8.2 Change in Table Structure: (ALTER)
(Adding a new column into a table with default value.)

It is used to modify/ change the structure of the tables by using the command **ALTER.**

8.2 Adding a new column into a table with default value.

Syntax:

ALTER TABLE <table name> ADD (<column name> <datatype> default data);

Example:

mysql>ALTER TABLE student ADD (City char(10) DEFAULT 'KOLKATA');

Result: New column **City** is added with default value as "**KOLKATA**" into the table.

Structured Query Language (SQL)

8.3 Change in Table Structure: (ALTER) (Modifying an existing data type of a column.)

It is used to modify/ change the structure of the tables by using the command **ALTER.**

8.3 Modifying an existing data type of a column / Column definition.

Syntax:

ALTER TABLE <table name> MODIFY([column_name]<datatype>);

Example:

mysql>ALTER TABLE student MODIFY Name varchar(15);

Result: The data type **varchar(20)** of a column **Name** is modified to **varchar(15).**

8.4 Change in Table Structure: (ALTER) (Rename a column.)

It is used to modify/ change the structure of the tables by using the command **ALTER.**

8.4 Rename a column.

Syntax:

ALTER TABLE <table name>
CHANGE([old column_name] [new column_name]<datatype>);

Example:

mysql>ALTER TABLE student CHANGE Marks Percent numeric(6,2);

Result: The column **Marks** is changed to **Percent.**

Note: Some SQL version **RENAME** or **MODIFY** also can be used in place of **CHANGE**.

Structured Query Language (SQL)

8.5 Change in Table Structure: (ALTER) (Removing a Column.)

It is used to modify/ change the structure of the tables by using the command **ALTER.**

8.5 Removing a Column.

Syntax:

ALTER TABLE <table name>DROP <column_name>;

Example:

mysql>ALTER TABLE student DROP Address;

Result: The column **Address** is deleted from the table **student**.

Note: The modification and changes carried out in the table structure (from 8.1 to 8.5) can be observed by using the command **'describe student'.** (see the next slide)

Contd..

Structured Query Language (SQL)

8. To view the changes carried out in the table structure from 8.1 to 8.5

Command **DESCRIBE** is used to view/ display the structure of the table**. We will use the command to verify the changes made.**

.

mysql>DESCRIBE student;

.

Result: Displayed the modified structure of the table student as shown below:

Field	Type	Null	Key	Default	Extra
RollNo	int(2)	NO	PRIMARY	NULL	
Name	varchar(15)	NO		NULL	
Gender	char(1)	YES		NULL	
Percent	numeric(6,2)	YES		NULL	
DOB	date	YES		NULL	
Mobile	int(11)	YES		NULL	
City	char(10)	YES		KOLKATA	
7 rows in set (0.01 sec)					

Note: Now keep the structure of the table as per the question given in beginning. Remove extra column by using ALTER command, which was carried out to learn the commands. Now we will learn how to insert the data into the table.

Structured Query Language (SQL)

9A. Inserting Data into a Table: (INSERT) (First Method)

INSERT INTO command is used to insert a new record/ row/ tuple in a table.

First Method :

Syntax: INSERT INTO <table name>VALUES(value1, value2, …..value5);

Example:

mysql>INSERT INTO student VALUE(1, "PRATIK", 'M', 95, '2003-04-22');

mysql>INSERT INTO student VALUE(2, 'GARIMA', 'F', 85, '2004-03-05');

Result: All the records are inserted into the table student as shown below:

Note : Following command will display the data inserted (explained later).

mysql> SELECT * FROM student;

Student

Rollno	Name	Gender	Marks	DOB
1	PRATIK	M	95	22-Apr-2003
2	GARIMA	F	85	05-Mar-2004

Structured Query Language (SQL)

9B. Inserting Data into a Table: (INSERT) (Second Method)

Second Method :

Syntax: INSERT INTO <table name> (colum1, colum2….)
VALUES(value1, value2….);

Example:

```
mysql>INSERT INTO student (RollNo, Name, Gender, Marks, DOB)
      VALUE(3, "KHUSHI", 'F', 92, '2003-01-24');
mysql>INSERT INTO student (RollNo, Name, Gender, Marks, DOB)
      VALUE(4, 'BABU', 'NULL', NULL, '2000-08-15');
mysql>.....
```

Result: All the records are inserted into the table student as shown below:

Student

Rollno	Name	Gender	Marks	DOB
1	PRATIK	M	95	22-Apr-2003
2	GARIMA	F	85	05-Mar-2004
3	KHUSHI	F	92	24-Jan-2003
4	BABU	NULL	NULL	15-Aug-2000
5	AMIT	M	60	26-Jan-2002
6	SHARMILA	F	90	12-Dec-2003

Structured Query Language (SQL)

10. Removing data from a table: (DELETE)

Command **DELETE** is used to delete a rows/ record from the table.

Syntax: DELETE FROM <table name>[WHERE <condition>];

Example 1:

mysql>DELETE FROM student WHERE name='BABU';

Result 1: Particular record/row will be deleted where name **BABU** is found.

Example 2:

mysql>DELETE FROM student;

or

mysql>TRUNCATE TABLE student;

Result 2: Both the command will delete all the rows from the student table. The difference is **DELETE** doesn't free the space whereas **TRUNCATE** free the space containing the table.

Structured Query Language (SQL)

11. Delete a table: (DROP TABLE)

Command **DROP TABLE** is used to delete a table permanently.

Syntax: DROP TABLE <table name>;

Example:

mysql>DROP TABLE student;

Result : Particular table **student** will be deleted from **school** database. You can verify the same by using command.

mysql> SHOW TABLES;

Note: Sometimes we may need to physically remove a table which is not in use. **DROP TABLE** command is used to delete a table permanently. But it cannot be deleted if it contains records. So, first delete all the rows of the table (DELETE FROM student;) and only then can the table be deleted.

Structured Query Language (SQL)

12. Creation of table 'fees'.

Table **'student'** has been created and data has been entered. In the same fashion, now you can create a table **'fees'** and inset the data into it as per the question given.

Consider in the database 'school' two tables 'student' and 'fees' has been created. The data are shown below as reference.

student ⇒ Table 1

Roll no	Name	Gender	Marks	DOB
1	PRATIK	M	95	22-Apr-2003
2	GARIMA	F	85	05-Mar-2004
3	KHUSHI	F	92	24-Jan-2003
4	BABU	NULL	NULL	15-Aug-2000
5	AMIT	M	60	26-Jan-2002
6	SHARMILA	F	90	12-Dec-2003

fees ⇒ Table 2

Roll no	Name	Fees	Attendance
1	PRATIK	5250.00	98
2	GARIMA	1250.00	70
4	BABU	1250.00	75
6	SHARMILA	5250.00	85

THANK YOU