Nikit Gokhe

Class- Comp D1

Roll No. 224042

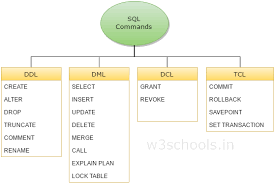
GR No. 21810522

**ASSIGNMENT NO.01**

**AIM:​** Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym.

**THEORY:**

DATA DEFINITION LANGUAGE (DDL) QUERIES



**DDL-​**

Data Definition Language (DDL) statements are used to define the database structure or schema. Data Definition Language understanding with database schemas and describes how the data should consist in the database, therefore language statements like CREATE TABLE or ALTER TABLE belong to the DDL.

DDL includes commands such as CREATE, ALTER and DROP statements.DDL is used to CREATE, ALTER OR DROP the database objects (Table, Views, Users). Data Definition Language (DDL) are used different statements :

**1] ​CREATE** -​ to create objects in the database

**2] ​ALTER​-** alters the structure of the database

**3]​DROP -​** delete objects from the database

**4] ​TRUNCATE​-** remove all records from a table, including all spaces allocated for the records are removed

**5] ​COMMENT​-** add comments to the data dictionary

**6]​RENAME​-** rename an object

**7] ​Create Database:​** From the MySQL command line, enter the command CREATE DATABASE <DATABASENAME>

**8] Select your database:** Once the database has been created, you will need to select it in order to begin editing it. Enter the command USE pune; You will see the message Database changed, letting you know that your active database is now pune.

**9] Display a list of your available databases:** Enter the command ​SHOW DATABASES;​ to list all of the databases you have stored. Besides the database you just created, you will also see a mysql database and a test database.

**10] Create table**:​We define an SQL relation by using the create table command. The following command creates a relation department in the database.

Example: create​ table department(dept name varchar (20),building varchar (15),budget numeric (12,2), primary key (dept name));

**11] Insert values in table:​**

Example:​insert into instructor values (10211, ’Smith’, ’Biology’, 66000);

**12] Drop table:​**To remove a relation from an SQL database, we use the drop table command.

**13] Alter Table :**We​ use the alter table command to add attributes to an existing relation.

**14] View:** SQL​ allows a “virtual relation” to be defined by a query, and the relation conceptually contains the result of the query.

**15] Create View:​**We define a view in SQL by using the create view command.

The form of the create view command is: create view v as<query expression>; where <query expression> is any legal query expression.

**16] Alter View:**​The CREATE VIEW statement creates a new view, or replaces an existing view if the OR REPLACE clause is given. If the view does not exist, CREATE OR REPLACE VIEW is the same as CREATE VIEW.

**17] Drop view:​**Use the DROP VIEW statement to remove a view or an object view from the database. You can change the definition of a view by dropping and re-creating it. Syntax:

Drop view viewname;

**Create Index:​**A database index is a data structure that improves the speed of operations in a table.

CREATE UNIQUE INDEX index\_name ON table\_name ( column1, column2,...);

**Alter Index:​**

There are four types of statements for adding indexes to a table −

**1] ALTER TABLE tbl\_name ADD PRIMARY KEY (column\_list)** −​ This statement adds a PRIMARY KEY, which means that the indexed values must be unique and cannot be NULL.

**2] ALTER TABLE tbl\_name ADD UNIQUE index\_name (column\_list)** −​ This statement creates an index for which the values must be unique (except for the NULL values, which may appear multiple times).

**3] ALTER TABLE tbl\_name ADD INDEX index\_name (column\_list) −​** This adds an ordinary index in which any value may appear more than once.

**4] ALTER TABLE tbl\_name ADD FULLTEXT index\_name (column\_list) −​** This creates a special FULLTEXT index that is used for text-searching purposes.

The following code block is an example to add index in an existing table.

ALTER TABLE student ADD INDEX (id);

**Drop Index:** You​ can drop any INDEX by using the DROP clause along with the ALTER command.

Try out the following example to drop the above-created index.

ALTER TABLE student DROP INDEX (id);

You can drop any INDEX by using the DROP clause along with the ALTER command.

**Sequence: ​**To create a sequence value in SQL query, we can use AUTO\_INCREMENT with optional starting value.

1] create table sonali(id int primary key auto\_increment, name varchar(20));

2] create table sonali(id int primary key auto\_increment = 10, name varchar(20));

**CODE AND OUTPUT:**

**mysql> create database student;**

**Query OK, 1 row affected (0.01 sec)**

**mysql> use student;**

**Database changed**

**mysql> create table student\_info(S\_idint(10),S\_Name**

**Varchar (30), S\_Address Varchar (30));**

**Query OK, 0 rows affected, 1 warning (0.03 sec)**

**mysql> insert into student\_info**

**values(101,'Sakshi','Aurangabad');**

**Query OK, 1 row affected (0.01 sec)**

**mysql> insert into student\_info**

**values(102,'Hem','Pune');**

**Query OK, 1 row affected (0.00 sec)**

**mysql> insert into student\_info**

**values(103,'Riya','Kolhapur');**

**Query OK, 1 row affected (0.00 sec)**

**mysql> insert into student\_info**

**values(104,'Ashish','Pune');**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from student\_info;**

**+------+--------+-----------+**

**| S\_id | S\_Name | S\_Address |**

**+------+--------+-----------+**

**| 101 | Sakshi | Aurangabad |**

**| 102 | Hem | Pune |**

**| 103 | Riya | Kolhapur |**

**| 104 | Ashish | Pune |**

**+------+--------+-----------+**

**mysql> create index group on student\_info(S\_Name);**

**Query OK, 0 rows affected (0.03 sec)**

**Records: 0 Duplicates: 0 Warnings: 0**

**mysql> show index from student\_info;**

**+--------------+------------+----------+--------------+-------------+-----------+------**

**-------+----------+--------+------+------------+---------+---------------+---------+---**

**---------+**

**| Table | Non\_unique | Key\_name | Seq\_in\_index |**

**Column\_name | Collation | Cardinality | Sub\_part | Packed | Null |**

**Index\_type | Comment | Index\_comment | Visible | Expression |**

**+--------------+------------+----------+--------------+-------------+-**

**----------+-------------+----------+--------+------+------------+-**

**--------+---------------+---------+------------+**

**| student\_info | 1 | group | 1 | S\_Name**

**| A | 4 | NULL | NULL | YES | BTREE**

**| | | YES | NULL |**

**+--------------+------------+----------+--------------+-------------+-**

**----------+-------------+----------+--------+------+------------+-**

**--------+---------------+---------+------------+**

**1 row in set (0.01 sec)**

**mysql> create view assign**

**-> as**

**-> select S\_Address**

**-> from student\_info;**

**Query OK, 0 rows affected (0.01 sec)**

**mysql> select \* from assign;**

**+-----------+**

**| S\_Address |**

**+-----------+**

**| Aurangabad |**

**| Pune |**

**| Kolhapur |**

**| Pune |**

**+-----------+**

**4 rows in set (0.00 sec)**

**CONCLUSION:**

⦁ Creating View, replacing or updating view and dropping view

⦁ Create view view1 as select \* from emp;

⦁ Create or replace view1 as select \* from emp1;

⦁ Drop view view1;

⦁ Crating sequence and inserting values

⦁ create table sonali(id int primary key auto\_increment, name varchar(20));

⦁ create table sonali(id int primary key auto\_increment = 10, name

varchar(20));

⦁ insert into sonali(name) values('a'),('b'),('c');

⦁ insert into sonali values(NULL,'sonali');

⦁ insert into sonali(name) values('sonali');

⦁ ​Creating index

⦁ create index id1 on sonali(id)

⦁ create index id1 on sonali(id desc)

In this assignment, we have studied and demonstrated various DDL statements in SQL.