

Government College of Engineering, Jalgaon
(An Autonomous Institute of Government of Maharashtra)

Name :	Semester : V	PRN :
Class : T. Y. B.Tech Computer	Academic Year : 2024-25	
Subject : CO307 DBMSL	CourseTeacher : Mr. Vinit Kakde	
Date of Performance :	Date of Completion :	

Practical no. 3

Aim: Create department table with the following structure

Name	Type
Deptno	INTERGER
Deptname	Varchar(10)
Location	Varchar(10)

- Add column designation to the department table.
- Insert values into the table.
- List the records of dept table grouped by deptno.
- Update the record where deptno is 9.
- Delete any column data from the table.
- Use of Rollback and commit.
- Add primary key constraints, not null constraints, unique key constraints

Theory:

Database Management System is a software or technology used to manage data from a database. Some popular databases are MySQL, Oracle, MongoDB, etc. DBMS provides many operations e.g. creating a database, Storing in the database, updating an existing database, delete from the database. DBMS is a system that enables you to store, modify, and retrieve data in an organized way. It also provides security to the database.

Concept and Purpose of Database

A database is a repository of data, stored as a table made up of masses of data that have some connection to each other.

The database is organized in that it uses records and fields so that the data is easier to use – whether it's to analyze, add, delete or otherwise manipulate it. Additionally, a database is structured so it is clear how the data points within it relate to one another. The structure of the data allows it to be accessible as needed by users and computer programs.

A database schema is a blueprint of how the data will look in a database, the type of database structure, if you will. It doesn't house the data but describes the shape of the data and how it relates to other tables.

What is COMMIT in SQL?

COMMIT is a transaction control language in SQL. It lets a user permanently save all the changes made in the transaction of a database or table. Once you execute the COMMIT, the database cannot go back to its previous state in any case.

What is ROLLBACK in SQL?

ROLLBACK is a transactional control language in SQL. It lets a user undo those transactions that aren't saved yet in the database. One can make use of this command if they wish to undo any changes or alterations since the execution of the last COMMIT.

Difference between COMMIT and ROLLBACK in SQL

Parameters	COMMIT in SQL	ROLLBACK in SQL
Basics/Definition	The COMMIT statement lets a user save any changes or alterations on the current transaction. These changes then remain permanent.	The ROLLBACK statement lets a user undo all the alterations and changes that occurred on the current transaction after the last COMMIT.
Condition of Transaction	Once you use the COMMIT command to (completely) execute the current transaction, then it cannot undo and get back to its previous state in any way.	On the other hand, the ROLLBACK command assists a user to get the command back to its previous state. It lets them undo the current transaction.
Syntax of the Command	The syntax of COMMIT is: Commit;	The syntax of ROLLBACK is: Rollback;
Successful Execution of the Statement	If one executes all the statements successfully with no error, then the COMMIT command will finally save the current state attained there.	In case an operation fails while completing a transaction, then it shows that the execution of all the changes is not successful. In this case, we can easily undo these changes with the ROLLBACK command.
Occurrence	One uses the COMMIT statement after successful completion of the intended transaction.	One uses the ROLLBACK statement after a transaction is unsuccessful due to abortion, incorrect execution, power failure, or system failure.
Visible Changes	Whenever a user executes the COMMIT command, then the current state of the transaction becomes permanently visible to all the users.	On the other hand, the transaction state after the ROLLBACK command also stays visible to all of its viewers- but the current transaction may contain a wrong set of information (it may also be right).

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Prevents actions that would destroy links between tables
- **CHECK** - Ensures that the values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column if no value is specified
- **CREATE INDEX** - Used to create and retrieve data from the database very quickly

Queries and outputs:

Create table Syntax:

```
CREATE TABLE table_name ( column1 datatype , column2 datatype , column3  
datatype , ... );
```

Query: create table department(deptno int, deptname varchar(10), location varchar(10));

```
mysql> use mysql  
Database changed  
mysql> create table department(deptno int, deptname varchar(10), location varchar(10));  
Query OK, 0 rows affected (2.38 sec)  
  
mysql> desc department;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| deptno     | int           | YES  |     | NULL    |       |  
| deptname   | varchar(10)   | YES  |     | NULL    |       |  
| location   | varchar(10)   | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.07 sec)
```

1. Add column designation to the department table Syntax: Alter table table_name add(column_name type); Query:
alter table department add(designation varchar(10));

```
mysql> desc department;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| deptno     | int           | YES  |     | NULL    |       |
| deptname   | varchar(10)   | YES  |     | NULL    |       |
| location   | varchar(10)   | YES  |     | NULL    |       |
| designation | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

2. Insert values into the table Syntax:
insert into table_name values
(value1,value2,.....,valueN);
Query: insert into department values(9,'accounting','hyderabad','manager');
insert into department values(10,'research','chennai','professor'); insert into
department values(11,'sales','banglore','salesman'); insert into department
values(12,'operations','mumbai','operator');

```
mysql> select * from department;
+-----+-----+-----+-----+
| deptno | deptname   | location | designation |
+-----+-----+-----+-----+
|      9 | accounting | hyderabad | manager     |
|     10 | research   | chennai  | professor   |
|     11 | sales      | banglore | salesman    |
|     12 | operations | mumbai   | operator    |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

3. List the records of dept table grouped by deptno Syntax: select
column1 from table_name group by column1; Query:select
deptno,deptname, from department group by deptno,
deptname;

```
+-----+-----+
| deptno | deptname |
+-----+-----+
|      9 | accounting |
|     10 | research   |
|     11 | sales      |
|     12 | operations |
+-----+-----+
4 rows in set (0.02 sec)
```

4. Update the record where deptno is 9 Syntax: update
table_name set column1=value1,column2=value2,.....
columnN=valueN;

Query: update department set designation ='accountant' where deptno=9;

```
mysql> select * from department;
```

deptno	deptname	location	designation
9	accounting	hyderabad	accountant
10	research	chennai	professor
11	sales	banglore	salesman
12	operations	mumbai	operator

4 rows in set (0.00 sec)

5. Delete any column data from the table Syntax:
alter table_name drop column(column_Name); Query:
alter table department drop column designation;

```
mysql> select * from department;
```

deptno	deptname	location
9	accounting	hyderabad
10	research	chennai
11	sales	banglore
12	operations	mumbai

4 rows in set (0.00 sec)

6. Insert any three records in the employee table and use rollback.
Check the result.
Syntax: insert into table_name (column1,column2,...columnN)
Values (value1,value2, ...,valueN);
Query:Start transaction;
Insert into department values where deptno=9;
Rollback;

```
mysql> rollback;
Query OK, 0 rows affected (0.19 sec)

mysql> select * from department;
+-----+-----+-----+
| deptno | deptname | location |
+-----+-----+-----+
|      9 | accounting | hyderabad |
|     10 | research  | chennai  |
|     11 | sales     | banglore |
|     12 | operations | mumbai   |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
+-----+-----+-----+
| deptno | deptname | location |
+-----+-----+-----+
|      9 | accounting | hyderabad |
|     10 | research  | chennai  |
|     11 | sales     | banglore |
|     12 | operations | mumbai   |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.10 sec)

mysql> delete from department where deptno=9;
Query OK, 1 row affected (0.11 sec)

mysql> select * from department;
+-----+-----+-----+
| deptno | deptname | location |
+-----+-----+-----+
|     10 | research | chennai  |
|     11 | sales    | banglore |
|     12 | operations | mumbai   |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Conclusion:

In this practical we performed different queries on table department. We first created dept named table then altered its name, column and at the last we deleted the table.

Questions:

1) Why ALTER TABLE command is used ?

ALTER TABLE is used to add,delete/drop or modify columns in the exsisting table.It is also used to add and drop various constraints on the exsisting table.

2) Why DROP COLUMN command is used ?

DROP COLUMN is used to drop columns in a table i.e deleting unwanted columns from the table

3) Why MODIFY COLUMN command is used ?

MODIFY COLUMN is used to modify the existing columns In a table.Multiple columns can also be modified at once.

4) Why ADD command is used ?

ADD is used to add columns into the existing table. With the use of ADD, we do not require to create a whole database again in case of adding additional information.

5) Why RENAME command is used ?

The RENAME command is used to change the name of an existing database object like table, column to a new name.

Name & Sign of Course Teacher

Mr. Vinit Kakde