LIST OF EXPERIMENTS

1. Create a database table, add constraints (primary key, unique, check, Not null), insert rows, update and delete rows using SQL DDL and DML commands.
2. Create a set of tables, add foreign key constraints and incorporate referential integrity.
3. Query the database tables using different ‘where’ clause conditions and also implement aggregate functions.
4. Query the database tables and explore sub queries and simple join operations.
5. Query the database tables and explore natural, equi and outer joins.
6. Write user defined functions and stored procedures in SQL.
7. Execute complex transactions and realize DCL and TCL commands.
8. Write SQL Triggers for insert, delete, and update operations in a database table.
9. Create View and index for database tables with a large number of records.
10. Create an XML database and validate it using XML schema.
11. Create Document, column and graph based data using NOSQL database tools.
12. Develop a simple GUI based database application and incorporate all the above-mentioned features
13. Case Study using any of the real life database applications from the following list

a) Inventory Management for a EMart Grocery Shop

b) Society Financial Management

c) Cop Friendly App – Eseva

d) Property Management – eMall

e) Star Small and Medium Banking and Finance

- Build Entity Model diagram. The diagram should align with the business and functional goals stated in the application.
- Apply Normalization rules in designing the tables in scope.
- Prepared applicable views, triggers (for auditing purposes), functions for enabling enterprise grade features.
- Build PL SQL / Stored Procedures for Complex Functionalities, ex EOD Batch Processing for calculating the EMI for Gold Loan for each eligible Customer.
- Ability to showcase ACID Properties with sample queries with appropriate settings

Table of Contents

S.No	Date	Name of the Experiment	Pg.No	Sign
1		CREATION OF TABLES (DDL ,DML COMMANDS)		
2		CREATION OF TABLES WITH CONSTRAINTS		
3		WHERE CLAUSE CONDITIONS AND AGGREGATE FUNCTIONS		
4		SIMPLE JOIN OPERATIONS		
5		NATURAL, EQUI AND OUTER JOIN OPERATIONS		
6		USER DEFINED FUNCTIONS AND STORED PROCEDURES		
7		DCL AND TCL COMMANDS		
8		TRIGGERS		
9		VIEWS AND INDEX		
10		XML DATABASE AND XML SCHEMA VALIDATION		
11		DOCUMENT,COLUMN AND GRAPH BASED DATA USING NOSQL DATABASE		
12		DEVELOP A SIMPLE GUI BASED DATABASE APPLICATION		

EX: NO: 1 CREATION OF TABLES (DDL COMMANDS)

AIM:

To execute and verify the Data Definition Language commands.

SQL Command Categories

SQL commands are grouped into four major categories depending on their functionality. They are as follows:

Data Definition Language (DDL)

These SQL commands are used for creating, modifying, and dropping the structure of database objects. The commands are CREATE, ALTER, DROP, RENAME, and TRUNCATE.

Data Manipulation Language (DML)

These SQL commands are used for storing, retrieving, modifying, and deleting data. These commands are SELECT, INSERT, UPDATE, and DELETE.

Transaction Control Language (TCL)

These SQL commands are used for managing changes affecting the data. These commands are COMMIT, ROLLBACK, and SAVEPOINT.

Data Control Language (DCL)

These SQL commands are used for providing security to database objects. These commands are GRANT and REVOKE.

DDL (DATA DEFINITION LANGUAGE)

- CREATE
- ALTER
- DROP
- TRUNCATE
- RENAME

PROCEDURE

STEP 1: Start
STEP 2: Create the table with its essential attributes.
STEP 3: Execute different Commands and extract information from the table.
STEP 4: Stop

SQL COMMANDS

1. COMMAND NAME: **CREATE**
COMMAND DESCRIPTION: **CREATE** command is used to create objects in the database.

```
 CREATE <OBJ.TYPE> <OBJ.NAME> (COLUMN NAME.1<DATATYPE> (SIZE), COLUMN
NAME.1 <DATATYPE> (SIZE)................................);
```

Syntax For Create A from an Existing Table With All Fields

```
  CREATE TABLE <TARGET TABLE NAME> AS SELECT * FROM  <SOURCE TABLE
NAME>;
```

2. COMMAND NAME: **DROP**
COMMAND DESCRIPTION: **DROP** command is used to delete the object from the database.

Syntax for drop a new column:

```
ALTER TABLE <TABLE NAME> DROP COLUMN <COLUMN NAME>;
```

Syntax for drop a table:

```
Drop table <tablename>;
```

3. COMMAND NAME: **TRUNCATE**
COMMAND DESCRIPTION: **TRUNCATE** command is used to remove all the records from the table

Syntax truncating the tables.

```
Truncate table <tablename>;
```

4. COMMAND NAME: **ALTER**
COMMAND DESCRIPTION: **ALTER** command is used to alter the structure of database.

```
ALTER <TABLE NAME> MODIFY <COLUMN NAME> <DATATYPE>(SIZE);
```

Syntax for alter table with multiple column:

```
SQL > ALTER <TABLE NAME> MODIFY <COLUMN NAME1> <DATATYPE>
(SIZE), MODIFY <COLUMN NAME2> <DATATYPE> (SIZE)......;
```

Syntax for add a new column:

SQL> ALTER TABLE <TABLE NAME> ADD (<COLUMN NAME1> <DATA TYPE> <SIZE>,<COLUMN NAME2> <DATA TYPE> <SIZE>,…………………………);

5. COMMAND NAME: **RENAME**
COMMAND DESCRIPTION: **RENAME** command is used to rename the objects.

Syntax For Renaming A table

Rename table <oldname> To <newname>;

Syntax For Renaming A Column

ALTER TABLE tablename **RENAME COLUMN** old column name **TO** new column name;

<u>Data base commands:</u>

Create database: create database <databasename>;

Show database: show databases;

Use: use <databasename>;

Show table: show tables;

Description of a table: desc <tablename>;

QUERY: 01
Q1. Write a query to create a table employee with empno, ename, designation, and salary.

QUERY: 01

SQL>CREATE TABLE EMP (EMPNO INT(4),ENAME VARCHAR(10),DESIGNATION VARCHAR(10),SALARY FLOAT(8,2));

Table created.

QUERY: 02
Q2. Write a query to display the column name and datatype of the table employee.

SQL> DESC EMP;

QUERY: 03

Q3. Write a query for create a new table from an existing table with all the fields.

QUERY: 03

```
SQL> CREATE TABLE EMP1 AS SELECT * FROM EMP;
Table created.
```

```
SQL> DESC EMP1
```

QUERY: 04

Q4. Write a query to create a table from an existing table with selected fields.

Syntax

```
SQL> CREATE TABLE <TARGET TABLE NAME> SELECT EMPNO, ENAME
FROM <SOURCE TABLE NAME>;
```

QUERY: 04

```
SQL> CREATE TABLE EMP2 AS SELECT EMPNO, ENAME FROM EMP;
Table created.
```

```
SQL> DESC EMP2;
```

ALTER & MODIFICATION ON TABLE

QUERY: 06

Q6. Write a Query to Alter the column EMPNO NUMBER (4) TO EMPNO NUMBER(6).

QUERY: 06

```
SQL>ALTER TABLE EMP MODIFY EMPNO NUMBER (6);
Table altered.
```

```
SQL> DESC EMP;
```

QUERY: 07

Q7. Write a Query to Alter the table employee with multiple columns (EMPNO, ENAME.)

Syntax for alter table with multiple column:

```
SQL > ALTER <TABLE NAME> MODIFY <COLUMN NAME1> <DATATYPE>
```

(SIZE), MODIFY <COLUMN NAME2> <DATATYPE> (SIZE)…….;

QUERY: 07

```
SQL>ALTER TABLE EMP MODIFY EMPNO INT (7),MODIFY  ENAME VARCHAR(12));

Table altered.

SQL> DESC EMP;
```

QUERY: 08

Q8. Write a query to add a new column in to employee

QUERY: 08

```
SQL> ALTER TABLE EMP ADD QUALIFICATION VARCHAR2(6);
Table altered.
SQL> DESC EMP;
```

QUERY: 09

Q9. Write a query to add multiple columns in to employee

```
SQL>ALTER TABLE EMP ADD (DOB DATE, DOJ DATE);

Table altered.
SQL> DESC EMP;
```

QUERY: 10

Q10. Write the query to change the table name emp as employee

```
SQL> Rename table emp to employee;
```

QUERY: 11

Q11. Write the query to change the column name empno to eno of the table employee

```
SQL>   ALTER TABLE employee RENAME COLUMN EMPNO TO ENO;
```

```
SQL> DESC EMPLOYEE;
```

REMOVE / DROP

QUERY: 12

Q12. Write a query to drop a column from an existing table employee

```
SQL> ALTER TABLE EMPLOYEE DROP COLUMN DOJ;
```

```
SQL> DESC EMP;
```

QUERY: 13

Q13. Write a query to truncate table employee

```
SQL>  truncate table employee;
```

QUERY: 14

Q14. Write a query to drop table employee

```
SQL>  drop table employee;
```

DML COMMANDS

Data Manipulation Language (DML)

These SQL commands are used for storing, retrieving, modifying, and deleting data. These commands are SELECT, INSERT, UPDATE, and DELETE.

DML (DATA MANIPULATION LANGUAGE)

- **SELECT**- It is used to retrieve information from the table. It is generally referred to as querying the table.

- **INSERT-** This is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined.

- **DELETE-** After inserting row in a table we can also delete them if required. The delete command consists of a from clause followed by an optional where clause.

- **UPDATE**- It is used to alter the column values in a table. A single column may be updated or more than one column could be updated.

SQL COMMANDS

1. COMMAND NAME: **INSERT**
COMMAND DESCRIPTION: INSERT command is used to Insert objects in the database.

2. COMMAND NAME: **SELECT**
COMMAND DESCRIPTION: SELECT command is used to SELECT the object from the database.

3. COMMAND NAME: **UPDATE**
COMMAND DESCRIPTION: **UPDATE** command is used to UPDATE the records from the table

4. COMMAND NAME: **DELETE**
COMMAND DESCRIPTION: DELETE command is used to DELETE the Records form the table

INSERT

QUERY: 01
Q1. Write a query to insert the records in to employee.

Syntax for Insert Records in to a table:
SQL > INSERT INTO <TABLE NAME> VALUES< VAL1, ‘VAL2’,…..);
A(
QUERY: 01

INSERT A RECORD INTO AN EXISTING TABLE:

```
MYSQL>INSERT INTO EMP VALUES(101,'NAGARAJAN','LECTURER',15000);
```

```
MYSQL >INSERT INTO EMP VALUES(102 ,‘SARAVANAN’,’ LECTURER’,15000);
MYSQL >INSERT INTO EMP VALUES(103,’PANNERSELVAM’,’ ASST. PROF,20000);
MYSQL >INSERT INTO EMP VALUES(104,’CHINNI HOD’,         ‘PROF’,45000);
```

SELECT

QUERY: 02

Q3. Write a query to display the records from employee.

Syntax for select Records from the table:

SQL> SELECT * FROM <TABLE NAME>;

QUERY: 02

DISPLAY THE EMP TABLE:

```
SQL> SELECT * FROM EMP;
```

UPDATE

QUERY: 04
Q1. Write a query to update the records from employee.
Syntax for update Records from the table:

```
SQL> UPDATE <<TABLE NAME> SET <COLUMNANE>=<VALUE> WHERE
<COLUMN NAME=<VALUE>;
```

QUERY: 04

```
SQL> UPDATE EMP SET SALARY=16000 WHERE EMPNO=101;
1 row updated.

SQL> SELECT * FROM EMP;
```

UPDATE MULTIPLE COLUMNS

QUERY: 05
Q5. Write a query to update multiple records from employee.

Syntax for update multiple Records from the table:

```
SQL> UPDATE <<TABLE NAME> SET <COLUMNAME>=<VALUE> WHERE
<COLUMN NAME=<VALUE>;
```

QUERY: 05

```
SQL>UPDATE EMP SET SALARY = 16000, DESIGNATIN='ASST. PROF' WHERE
EMPNO=102;
1 row updated.
SQL> SELECT * FROM EMP;
```

DELETE

QUERY: 06

Q5. Write a query to delete records from employee.

Syntax for delete Records from the table:

```
SQL> DELETE <TABLE NAME> WHERE <COLUMN NAME>=<VALUE>;
```

QUERY: 06

```
SQL> DELETE EMP WHERE EMPNO=103;

SQL> SELECT * FROM EMP;
```

<u>Result:</u>

Thus the DDL, DML commands are executed in MySQL and verified successfully.