1. **Create a table called Employee & execute the following.** **DATE:**
   Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION)
   1. Create a user and grant all permissions to the user.
   2. Insert the any three records in the employee table contains attributes **EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION** and use rollback. Check the result.
   3. Add primary key constraint and not null constraint to the employee table.
   4. Insert null values to the employee table and verify the result.

DESCRIPTION

DDL COMMANDS:

i) Create
ii) Alter
iii) Drop

DML COMMANDS:

i) Insert
ii) Select
iii) Update
iv) Delete

1. DDL (Data Definaion Language) Comands.

DDL Comands are used to define and manage the Structure of a database.

a) Create.
used to Create a new database or table.

b) Alter.
used to modify an exishing table Structure (add, delete or modify Column)

e) DROP.

Used to delete an entire table, database, or column permanently.

Q. Data Manipulation language (DML) commands.

DML commands are used to manipulate and retrieve data in a database.

a) INSERT.

Used to insert records into a table.

b) SELECT

Used to retrive data from a table.

c) Update.

Used to modify existing records in a table.

d) DELETE.

Used to delete records from a table.

Create a table called Employee with its attributes
Employee (EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION)

```
CREATE TABLE EMP.EMPLOYEE (
    EMPNO NUMBER(5),
    ENAME VARCHAR(20),
    JOB VARCHAR(15),
    MANAGER_NO NUMBER(5),
    SAL NUMBER(6),
    COMMISSION NUMBER(6)
);
```

1. Create a user and grant all permissions to the user.

```
GRANT ALL ON EMPLOYEE TO RAVINDRA;
```

2. Insert the any three records in the employee table contains attributes EMPNO, ENAME, JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result.

```
INSERT INTO EMPLOYEE VALUES ('E1', 'RAM', 'MANAGER',
                2000000);
INSERT INTO EMPLOYEE VALUES ('E2', 'SANJU', 'LAYMAN',
    'E1', 1600000);
INSERT INTO EMPLOYEE VALUES ('E3', 'RAVINDA', 'DESIGNA',
    'E1', 152000, 200000);
INSERT INTO EMPLOYEE VALUES ('E4', 'TONY', 'DEV', 'E2',
        100000);
ROLLBACK;
```

TABLES WITH VALUES

EMPNO	ENAME	JOB	MANAGER_NO	SAL	COMMISSION
E1	RAM	MANAGER	NULL	2000000	NULL
E2	SANJU	LAYMAN	E1	1600000	NULL
E3	RAVINDRA	DESIGNER	E1	1520000	200000
E4	TONY	DEV.	E2	100000	NULL

3. Add primary key constraint and not null constraint to the employee table.

```
ALTER TABLE EMPLOYEE
ADD CONSTRAINT PK_EMPLOYEE PRIMARY KEY (EMPNO);
CONSTRAINT FK_EMPLOYEE FOREIGN KEY (MANAGER_NO)
REFERENCE EMPLOYEE (EMPNO);
```

Result:

TABLE UPDATED.

4. Insert null values to the employee table and verify the result.

```
INSERT INTO EMPLOYEE VALUES ("E6', 'BRUCE', 'DEV',
NULL, 180000, NULL);
```

Result:

EMPO	ENAME	JOB.	MANAGER_NO.	SAL	COMMISSION
		EXISTING Data.			
E6.	BRUCE	DEV.	NULL	180000	NULL.

DATE:

2. Create a table called Employee that contain attributes EMPNO, ENAME, JOB, MGR, SAL & execute the following.

1. Add a column commission with domain to the Employee table.
2. Insert any five records into the table.
3. Update the column details of job
4. Rename the column of Employ table using alter command.
5. Delete the employee whose Empno is 105.

Create a table called Employee that contain attributes EMPNO, ENAME, JOB, MGR, SAL.

```
CREATE TABLE EMP (
    EMPNO NUMBER(5),
EMPENAME VARCHAR(20),
  JOB     VARCHAR(15),
 MANAGER_NO CHAR(5),
 SAL.     NUMBER(5),
 CONSTRANT PK_EMP PRIMARY KEY(EMPNO),
 CONSTRANT FK_EMP FOREIGN KEY(MANAGER_NO)
  REFERENCES EMP(EMPNO);
```

1. Add a column commission with domain to the Employee table.

```
ALTER TABLE EMPLOYEE
  ADD COLUMN COMMISSION NUMBER(5);
```

Result:

EMPNO	EMP ENAME	JOB	MANAGER_NO	SAL	COMMISSION.

2. Insert any five records into the table.

```
INSERT INTO EMPLOYEE VALUES('101','RAVENDRA','MANAGER
  200000);
```

INSERT INTO EMPLOYEE VALUES (102, 'TONY', 'SALESMAN', 101, 1600000, 5000);

Result:

EMPNO	EMPNAME	JOB	MANAGER NO	SAL	COMMISSION
101	RAVINDRA	MANAGER	NULL	2000000	NULL
102	TONY	SALESMAN	101	1600000	5000

3. Update the column details of job

```
UPDATE EMPLOYEE
SET JOB = 'DIRECTOR'
WHERE EMPNO = 102;
```

Result:

EMPNO	EMPNAME	JOB	MANAGERNO	SAL	COMMISSION
101	RAVINDRA	MANAGER	NULL	2000000	NULL
102	TONY	DIRECTOR	101	1600000	5000

4. Rename the column of Employee table using alter command

```
ALTER TABLE EMP RENAME EMP as EMPS;
```

Result:

TABLE NAME UPDATED.

5. Delete the employee whose Empno is 105. (Assumption)

```
DELETE FROM EMP WHERE EMPNO = 105;
```

Before Delete Result:

EMPNO	EMPNAME	JOB	MANAGER NO	SAL	COMMISSION
101	RAVINDRA.	MANAGER	NULL	2000000	NULL
102	TONY	DIRECTOR	101	1600000	45000
103	STANK	PLAY.	101	100000	NULL
104.	BRUCE .	PANZER.	104	200200	20000
105 .	Niper .	DEVOPS.	102 .	1000250	NULL

After Delete Result:

EMPNO	ENDNAME	JOB	MANAGER NO	SAL	COMMISSION
101	RAVINDRA	MANAGGR	NULL	2000000	NULL
102	TONY	DIRECTOR	101	1600000	50000
103	STANK	PLAY	101	1000000	NULL
104 ,	BRUCE	PANZER	104	2002000	20020

DATE:

3. Queries using aggregate functions COUNT, AVG,MIN,MAX,SUM), Group by, Order by, Employee (E_id, E_name, Age, Salary)

1. Create Employee table containing all Records E_id, E_name, Age, Salary.
2. Count number of employee names from employee table
3. Find the Maximum age from employee table.
4. Find the Minimum age from employee table.
5. Find salaries of employee in Ascending Order.
6. Find grouped salaries of employees.

DESCRIPTION

i) Aggregate Functions with example
ii) Group by &Order by with example

1) Aggregate functions

This functions perform calculations on a set of values. and return a single values.

Count() : Returns the number of rows : SELECT COUNT(*) FROM ;

SUM(): Returns the Sum of a Column : SELECT SUM(SAL) FROM EMP;

AVG(): Returns the Avg value : SELECT AVG(SAL) FROM EMP;

MAX(); Returns the highest value: SELECT MAX(SAL) FROM EMP;

MIN(); Returns the lowest value: SELET MIN(SAL) FROM EMP;

Q. Group by and Order By

=> Group by

The rows that have the same values into summary row.

Ex :-

```
SELECT job, COUNT (*) AS total-employees FROM EMP.
  GROUP BY job;
```

=> ORDER by

The order by clause is used to sort the result in ascending (ASC) or descending (DESC) order.

Ex:

```
SELECT ename, salary FROM EMPLOYEE.
ORDER BY SALARY . DESC;
```

1. Create Employee table containing all Records E_id, E_name, Age, Salary.

```
CREATE TABLE EMPLOYEE (
    E_id NUMBER (6),
    E-NAME CHAR (50),
    Age NUMBER (6),
    Salary NUMBER(8));
ALTER TABLE EMPLOYEE UPDATE CONSTRANT
PK_EMP PRIMARY KEY (EMPNO), CONSTRANT FK_
EMP FOREIGN KEY (MANAGER_NO) REFERENCES EMPLOYEE;
```

2. Count number of employee names from employee table

```
SELECT COUNT (*) FROM EMPLOYEE;
```

Result:

```
COUNT (*)
---------
5
```

3. Find the Maximum age from employee table.

```
SELECT MAX (Age) FROM EMPLOYEE;
```

Result:

```
MAX (Age)
---------
50
```

4. Find the Minimum age from employee table.

```
SELECT MIN (Age) FROM EMPLOYEE;
```

Result:

```
MIN (Age)
---------
23
```

5. Find salaries of employee in Ascending Order.

```
SELECT * FROM EMPLOYEE ORDER BY SALARY;
```

Result:

```
 SALARY
--------
 1600000
 2600000
 5600000
 7600000
 9000000
17600000
```

6. Find grouped salaries of employees.

```
/*SELECT * FROM EMPLOYEE
  GROUP BY AGE; */
```

Result: `SELECT age, COUNT(*) FROM GROUP BY age;`

```
 AGE | COUNT(*)
-----|---------
 23  |    1
 25  |    1
 27  |    1
 29  |    1
 30  |    1
 50  |    1
```