Lab Sheet - 5

Objective :- To understand and apply DDL Commands in SQL for creating, modifying and managing database structure such as tables columns.

1. Create the Department - ------ NULL

```
CREATE TABLE Departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(50) NOT NULL,
    location VARCHAR(50)
);
```

2. Create the Employes - -------- Departments

```
CREATE TABLE Employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50) NOT NULL,
    Salary DECIMAL(10,2) Check (Salary > 30000),
    hire_date DATE,
    dept_id INT,
    email VARCHAR(100) UNIQUE,
    FOREIGN KEY (dept_id) REFERENCE Department
        (dept_id)
);
```

3. Create the Project - ---- ---50000.

```
CREATE TABLE Project (
    Proj_id VARCHAR(10) PRIMARY KEY,
```

```
Proj_name VARCHAR(100) NOT NULL,
    Start_date DATE,
    end_date DATE,
budget DECIMAL(12,2) CHECK(budget>=50000),
    dept_id INT
FOREIGN KEY(dept_id) REFERENCE Department
    (dep_id)
);
```

4. Create the employee ———————— Projects

```
CREATE TABLE Employee Projects(
    emp_id INT,
    Proj_id VARCHAR(10),
    role VARCHAR(50) NOT NULL,
    PRIMARY KEY(emp_id, Proj_id),
FOREIGN KEY(emp_id) REFERENCES Employes(emp_id),
FOREIGN KEY(proj_id) REFERENCES project
    (project_id)
);
```

5. Alter the Employee ———— ———— ————Constraint

```
ALTER TABLE Employees
ADD phone_number VARCHAR(15) UNIQUE;
```

6. Alter the Project - - - - - - - - - NOT NULL

```
ALTER TABLE projects
MODIFY budget DECIMAL(12,2) NOT NULL;
```

7. Drop the Employee Project table

```
Drop TABLE Employees Projects;
```

8. Recreate the Employee ------ before.

```
CREATE TABLE Employee Project (
    emp-id INT,
Proj-id VARCHAR(10),
role VARCHAR(50) NOT NULL.
PRIMARY KEY (emp-id, Proj-id),
FOREIGN KEY (emp-id) REFERENCES
    Employee (emp-id)
FOREIGN KEY (proj-id) REFERENCES
    projects (proj-id)
);
```

9. Rename the Department table to Dept-Info.

```
RENAME TABLE Departments To Dept-info;
```

10. Add - - ---------- Departments

```
ALTER TABLE Dept-info ALTER location
SET DEFAULT 'unknown';
```

11. Insert ---------- 'New York'.

```
INSERT INTO Dept-info (dept-id, dept-name
                       location)
VALUES (1, 'HR', 'NEW YORK');
```

12. Insert ---------- Statement

```
INSERT INTO Departments (dept_id,
                         dept_name, location)

VALUES
  (101, 'HR', 'New York'),
  (102, 'IT', 'San Francisco'),
  (103, 'Finance', 'Chicago'),
  (104, 'Marketing', 'Boston');
```

13. Insert ---------- Valid.

```
INSERT INTO Employee (emp_id, emp_name
  Salary, hire_date, dept_id, email)

VALUES (1, 'Alice Johnson', 60000, '2020-02-10', 101,
  'alice.Johnson@gmail.com');
```

14. Insert ---------- uniqueemails

```
INSERT INTO Employees (emp_id, emp_name,
    Salary, hire_date, dept_id, email)

VALUES
(1, 'Alice Johnson', 60000, '2020-02-10', 101,
    'alice.jhonson@gmail.com'),
(2, 'Bob Smith', 75 000, '2019-06-15', 102,
    'bob.Smith@gmail.com');
(3, 'Charlie Brown', 80000, '2021-03-12', 103,
    'Charlie brown@gmail.com');
```

15. Insert a new project ---------- 75000

```
INSERT INTO Projects (Proj_id, Proj_name)
Start_date, end_date, budget, dept_id)
Values ('P5' 'New CRM System', 2023-02-01',
        '2023-12-31', 75000, 102);
```

16. Insert mutiple ---------- budget

```
INSERT INTO projects (Proj_id, Proj_name,
Start_data, end_date, budget, dept_id)
```

17. Insert ---------- role.

```
INSERT INTO Employee Projects (emp_id,
Proj_id, role)
VALUES (1, 'P2', 'Developer');
```

18. Update Salary ---------- by 10%.

```
UPDATE Employees
SET Salary = Salary * 1.10
WHERE emp_id = 2;
```

19. Update location ---------- 'Boston'.

```
UPDATE Dept_info
SET location = 'Boston'
WHERE location = 'New York';
```

20. Update the role ——————— Projects

```
UPDATE Employee Project
SET role = 'Team lead'
WHERE emp_id = 1 AND proj_id = 'P2';
```

21. Delete a project with a Specific proj-id.

```
DELETE from Projects
WHERE proj_id = 'P3';
```

22. Delete an employee only if their Department ID is 2.

```
DELETE FROM Employees WHERE dept_id = 2;
```

23. Delete all records - - —— - - - proj-id.

```
DELETE FROM Employes Projects WHERE
project_id = 'P2';
```

24. Insert a department with a duplicate dept-id and observe the result.

```
INSERT INTO Department (dept_id, dept_name,
location)

VALUES (101, 'Duplicate Dept', 'Test location');
```

25. Insert an employee ——— Violation.

```
INSERT INTO Employees (emp_id, emp_name,
Salary, hire_date, dept_id, email)
VALUES (6, 'Test Negative', -50000, '2024-01-01',
101, 'test_negative@gmail.com');
```

26. Insert an employee without ——— Null / UNIQUE behaviour

```
INSERT INTO Employees (emp_id, emp_name,
Salary, hire_data, dept_id)
Values (7, 'No Email', 60000, '2024-02-02', 101);
```

27. Update the budget ——— Constraint?

```
UPDATE Projects
SET Budget = 40000
WHERE Proj_id = 'P1';
```

28. Insert an Employee ——— Violation).

```
INSERT INTO Employee Project (emp_id, Proj_id
role)
VALUES (999, 'P1', 'Tester');
```

29. Truncate Employee table & insert new data

```
TRUNCATE TABLE Employees;

INSERT INTO Employees (emp_id, emp_name,
    Salary, hire_date, dept_id, email)
VALUES (1, 'New Employed', 55000, '2025-01-01',
    lol; new employee@gmail.com);
```

30. Drop all tables in revese depency order.

```
DROP TABLE Employees Projects;
DROP TABLE Projects;
DROP TABLE Employees;
DROP TABLE Departments;
```

8/10/25