



# Experiment 1

**Student Name:** RAVI RANJAN KUMAR

**UID:** 25MCA20098

**Branch:** MCA general

**Section/Group:** 25MCA\_KAR-1

**Semester:** II

**Date of Performance:** 05-01-2026

**Subject Name:** Technical Training

**Subject Code:** 25CAP-652

## 1. Aim:

To design and implement a sample database system using DDL, DML, and DCL commands, including database creation, data manipulation, schema modification, and role-based access control to ensure data integrity and secure, read-only access for authorized users.

## 2. Objective:

To gain practical experience in implementing Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL) operations in a real database environment. This will also include implementing role-based privileges to secure data.

## 3. Implementation/Code:

```
-- DDL
-- DEPARTMENT TABLE
CREATE TABLE department(
    department_id INT PRIMARY KEY,
    department_name VARCHAR(20) NOT NULL UNIQUE,
    salary FLOAT CHECK(salary>=0)
);
```

```
-- EMPLOYEE TABLE
CREATE TABLE employee(
    employee_id INT PRIMARY KEY,
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
employee_name VARCHAR(20) NOT NULL,  
department_id INT NOT NULL REFERENCES department(department_id),  
employee_contact VARCHAR(20),  
join_date DATE NOT NULL,  
end_date DATE CHECK(end_date>=join_date)  
);
```

```
ALTER TABLE employee ADD work_location VARCHAR(20);  
ALTER TABLE employee DROP work_location;  
ALTER TABLE employee ADD status VARCHAR(20) DEFAULT 'active';
```

```
-- PROJECT TABLE  
CREATE TABLE project(  
project_id INT PRIMARY KEY,  
project_name VARCHAR(20) NOT NULL UNIQUE,  
department_id INT NOT NULL REFERENCES department(department_id),  
start_date DATE NOT NULL,  
end_date DATE CHECK(end_date>=start_date)  
);
```

```
INSERT INTO department  
VALUES  
(101,'Manager',90000),  
(102,'HR',70000),  
(103,'EMPLOYEE',50000);  
UPDATE department set salary=80000 WHERE department_id=101;  
UPDATE department set salary=60000 WHERE department_id=102;  
UPDATE department set salary=100000 WHERE department_id=103;
```

```
UPDATE department SET department_name='Employee' WHERE  
department_id=103;
```

```
INSERT INTO department
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

VALUES

```
(104,'DEVELOPER',-30000);
```

INSERT INTO department

VALUES

```
(104,'DEVELOPER',30000);
```

```
DELETE FROM department WHERE department_id=104;
```

INSERT INTO employee

VALUES

```
(1,'Rahul',101,8888888888,'2001-04-12','2010-07-13'),  
(2,'Anuj',102,7777777777,'2003-06-10','2004-05-11'),  
(3,'Aman',103,6666666666,'2006-05-20','2009-09-11'),  
(4,'Naman',103,5555555555,'2006-06-25','2009-08-11'),  
(5,'Karan',103,4444444444,'2006-03-12','2009-05-11');
```

```
DELETE FROM employee WHERE employee_id=3;
```

INSERT INTO project

VALUES

```
(11,'P1',103,'2025-08-14','2025-09-14'),  
(12,'P2',103,'2025-08-14','2025-08-30');
```

-- DQL

```
SELECT * FROM department;
```

```
SELECT * FROM employee;
```

```
SELECT * FROM project;
```

-- DCL

```
CREATE ROLE reporting_user
```

```
LOGIN
```

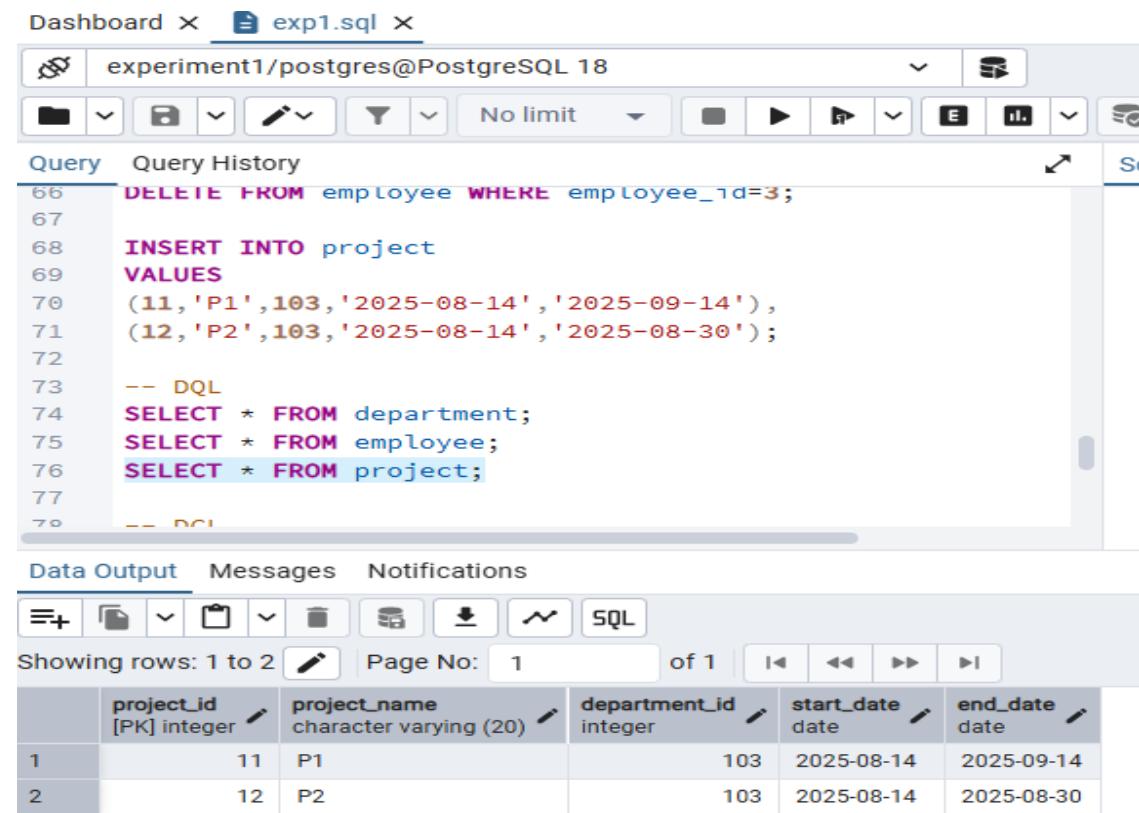
```
PASSWORD
```

```
'user123';
```

```
GRANT SELECT ON department TO reporting_user;  
REVOKE SELECT ON department FROM reporting_user;  
GRANT SELECT ON project TO reporting_user;  
REVOKE CREATE ON SCHEMA PUBLIC FROM reporting_user;
```

```
-- QUERY FROM reporting_user  
SELECT * FROM project;
```

#### 4. Output: Project Table



The screenshot shows the pgAdmin interface with the query results for the project table. The top part displays the SQL code used to delete from employee, insert into project, and select from department, employee, and project tables. The bottom part shows the resulting data output for the project table, which contains two rows of data: P1 and P2.

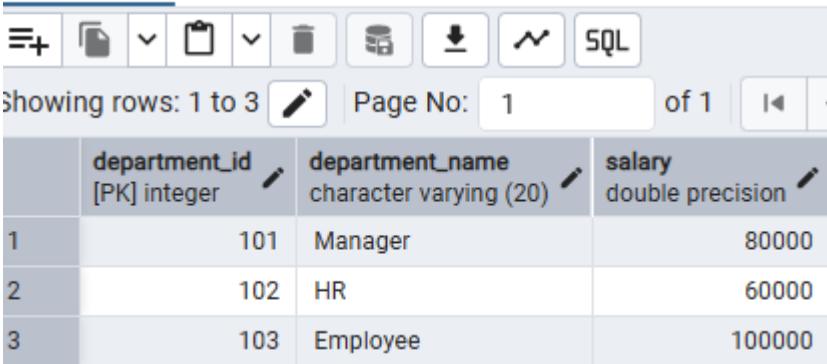
project_id [PK] integer	project_name character varying (20)	department_id integer	start_date date	end_date date
11	P1	103	2025-08-14	2025-09-14
12	P2	103	2025-08-14	2025-08-30

```

75  SELECT * FROM department;
76  SELECT * FROM employee;
77  SELECT * FROM project;
78
79  -- DCL

```

Data Output Messages Notifications



The screenshot shows a PostgreSQL query tool interface. At the top, there are buttons for file operations (New, Open, Save, Print, etc.) and a SQL button. Below that is a toolbar with icons for search, refresh, and other functions. The main area displays the results of a query:

```

Showing rows: 1 to 3
Page No: 1 of 1

```

The table has three columns: department\_id, department\_name, and salary. The data is as follows:

	department_id [PK] integer	department_name character varying (20)	salary double precision
1	101	Manager	80000
2	102	HR	60000
3	103	Employee	100000

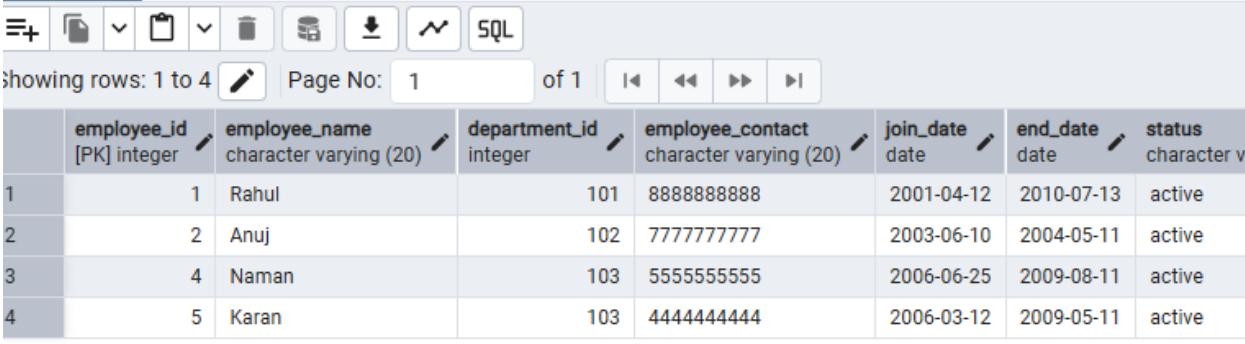
## Employee Table

```

76  SELECT * FROM employee;
77  SELECT * FROM project;
78
79  -- DCL

```

Data Output Messages Notifications



The screenshot shows a PostgreSQL query tool interface. At the top, there are buttons for file operations (New, Open, Save, Print, etc.) and a SQL button. Below that is a toolbar with icons for search, refresh, and other functions. The main area displays the results of a query:

```

Showing rows: 1 to 4
Page No: 1 of 1

```

The table has eight columns: employee\_id, employee\_name, department\_id, employee\_contact, join\_date, end\_date, and status. The data is as follows:

	employee_id [PK] integer	employee_name character varying (20)	department_id integer	employee_contact character varying (20)	join_date date	end_date date	status character v
1	1	Rahul	101	8888888888	2001-04-12	2010-07-13	active
2	2	Anuj	102	7777777777	2003-06-10	2004-05-11	active
3	4	Naman	103	5555555555	2006-06-25	2009-08-11	active
4	5	Karan	103	4444444444	2006-03-12	2009-05-11	active

## 5. Learning Outcomes:

1. About query writing in PostgreSQL.
2. About various DDL, DML and DCL commands.
3. About the application of CHECK constraint.
4. About role-based privileges to secure data.