```
CREATE DATABASE Company_simple_DB;

CREATE TABLE Departments (
   department_id SERIAL PRIMARY KEY,
   department_name VARCHAR(100) NOT NULL
);

INSERT INTO Departments (department_name) VALUES
('Software Development'),
('Quality Assurance'),
('Product Management'),
('Marketing'),
('Finance and Accounting'),
('Human Resources ');


-- Create employees table
CREATE TABLE employees (
   employee_id SERIAL PRIMARY KEY,
   employee_name VARCHAR(50) NOT NULL,
   email VARCHAR(100) UNIQUE NOT NULL,
   salary NUMERIC(10, 2),
   department_id INT REFERENCES Departments(department_id)
);

-- Insert employees
INSERT INTO employees (employee_name, email, salary, department_id)
VALUES
('Sunil', 'sunil123@gmail.com',  65000, 1),
('Mudit', 'Mudit34@gmail.com', 70000, 2),
('Yash', 'Yash54@gmail.com', 75000, 3),
```

```
('Diana', 'diana.p@example.com', 48000, 3),
('Ravi', 'Ravi.c@example.com', 62000, 4),
('Fiona','fiona.w@example.com',50000, 5);



-- Create projects table
CREATE TABLE projects (
   project_id SERIAL PRIMARY KEY,
   project_name VARCHAR(100) NOT NULL,
   employee_id INT REFERENCES employees(employee_id)
);



INSERT INTO projects (project_name,  employee_id)
VALUES
('HR Portal',1),
('Mobile Banking App',3),
('AI Chatbot Assistant',5),
('Smart Billing System',1);

SELECT * FROM Departments;
SELECT * FROM employees;
SELECT * FROM  projects ;

-- Practice with JOINs
--  1. Retrieve all employees with their department names

SELECT e.employee_id, d.department_name
FROM employees as e
JOIN Departments as d
ON e.department_id = d.department_id;
```

	employee_id integer	department_name character varying (100)
1	1	Software Development
2	2	Quality Assurance
3	3	Product Management
4	4	Product Management
5	5	Marketing
6	6	Finance and Accounting

-- 2. List all employees and the projects they are working on. Include employees even if not assigned to any project.

```
SELECT e.employee_id, p.project_name
FROM employees as e
LEFT JOIN projects as p
ON e.employee_id = p.employee_id;
```

	employee_id integer	project_name character varying (100)
1	1	HR Portal
2	3	Mobile Banking App
3	5	AI Chatbot Assistant
4	1	Smart Billing System
5	2	[null]
6	6	[null]
7	4	[null]

-- 3. Show all departments and the number of employees in each department.

```
SELECT
   d.department_name,
   COUNT(e.employee_id) as employee_count
FROM departments as d
LEFT JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name;
```

	department_name character varying (100)	employee_count bigint
1	Marketing	1
2	Product Management	2
3	Software Development	1
4	Quality Assurance	1
5	Human Resources	0
6	Finance and Accounting	1

-- 4. List employees who are not assigned to any project.

```
SELECT e.employee_id
FROM employees as e
LEFT JOIN projects as p ON
e.employee_id = p.employee_id
WHERE p.project_id IS NULL;
```

	employee_id [PK] integer
1	2
2	6
3	4

-- 5. Get the names of all employees along with their department names and the projects they are working on (if any).

```
SELECT e.employee_id, d.department_name, p.project_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id
LEFT JOIN projects p ON e.employee_id = p.employee_id;
```

	employee_id integer	department_name character varying (100)	project_name character varying (100)
1	1	Software Development	HR Portal
2	3	Product Management	Mobile Banking App
3	5	Marketing	AI Chatbot Assistant
4	1	Software Development	Smart Billing System
5	2	Quality Assurance	[null]
6	6	Finance and Accounting	[null]
7	4	Product Management	[null]

- Part 2: Practice with GROUP BY and Aggregates

-- 6. Show the total salary paid by each department

```
SELECT
   d.department_name,
   SUM(e.salary) AS total_salary
FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name;
```

	department_name character varying (100)	total_salary numeric
1	Marketing	62000.00
2	Product Management	123000.00
3	Software Development	65000.00
4	Quality Assurance	70000.00
5	Finance and Accounting	50000.00

-- 7. Find the average salary in each department

```
SELECT
   d.department_name,
   AVG(e.salary) AS average_salary
FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name;
```

	department_name character varying (100)	average_salary numeric
1	Marketing	62000.000000000000
2	Product Management	61500.000000000000
3	Software Development	65000.000000000000
4	Quality Assurance	70000.000000000000
5	Finance and Accounting	50000.000000000000

```
-- 8. List departments having more than 3 employees
SELECT
    d.department_name,
    COUNT(e.employee_id) AS employee_count
FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name
HAVING COUNT(e.employee_id) > 1
```

	department_name character varying (100)	employee_count bigint
1	Product Management	2

```
-- 9. Display the department with the highest average salary
SELECT
    d.department_name,
    AVG(e.salary) AS avg_salary
FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_name
ORDER BY avg_salary DESC LIMIT 1;
```

	department_name character varying (100)	avg_salary numeric
1	Quality Assurance	70000.000000000000

-- 10. Count the number of projects each employee is assigned to

```
SELECT
    e.employee_id,
    e.employee_name,
    COUNT(p.project_id) AS project_count
FROM employees e
LEFT JOIN projects p ON e.employee_id = p.employee_id
GROUP BY e.employee_id, e.employee_name;
```

	employee_id [PK] integer	employee_name character varying (50)	project_count bigint
1	3	Yash	1
2	5	Ravi	1
3	4	Diana	0
4	6	Fiona	0
5	2	Mudit	0
6	1	Sunil	2