```
-- Create the departments table
CREATE TABLE departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(100) NOT NULL
);

-- Create the employees table with a foreign key referencing departments
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(100) NOT NULL,
    job_title VARCHAR(100) NOT NULL,
    department_id INT,
    salary DECIMAL(10, 2),
    FOREIGN KEY (department_id) REFERENCES departments(dept_id)
);

-- Create the salary_ranges table
CREATE TABLE salary_ranges (
    job_title VARCHAR(100) PRIMARY KEY,
    min_salary DECIMAL(10, 2) NOT NULL,
    max_salary DECIMAL(10, 2) NOT NULL
```

```
);
```

```
-- Insert records into the departments table
INSERT INTO departments (dept_id, dept_name) VALUES
(1, 'IT'),
(2, 'Project Management'),
(3, 'Human Resources'),
(4, 'Sales'),
```

```
(5, 'Marketing');


-- Insert records into the employees table
INSERT INTO employees (emp_id, emp_name, job_title, department_id, salary) VALUES
(1, 'Rahul Sharma', 'Software Engineer', 1, 60000),
(2, 'Priya Verma', 'Software Engineer', 1, 65000),
(3, 'Amit Patel', 'Project Manager', 2, 75000),
(4, 'Sunita Singh', 'HR Manager', 3, 55000),
(5, 'Neha Gupta', 'Software Engineer', 1, 62000),
(6, 'Ravi Kumar', 'Project Manager', 2, 77000),
(7, 'Meera Nair', 'HR Manager', 3, 57000),
(8, 'Arjun Reddy', 'Sales Executive', 4, 50000),
(9, 'Sneha Iyer', 'Sales Executive', 4, 51000),
(10, 'Anjali Desai', 'HR Specialist', 3, 52000);


-- Insert records into the salary_ranges table
INSERT INTO salary_ranges (job_title, min_salary, max_salary) VALUES
('Software Engineer', 60000, 70000),
('Project Manager', 75000, 80000),
('HR Manager', 55000, 60000),
('Sales Executive', 50000, 52000),
```

('HR Specialist', 52000, 55000);

select * from departments;

SELECT e.emp_name, e.job_title, d.dept_name

FROM employees AS e

INNER JOIN departments AS d ON e.department_id = d.dept_id;

SELECT e.emp_name, e.job_title, d.dept_name

FROM employees AS e

LEFT JOIN departments AS d ON e.department_id = d.dept_id;

-- Insert an employee with a non-existing department ID (6)

-- Insert an employee with a non-existing department ID (7)

INSERT INTO employees (emp_id, emp_name, job_title, department_id, salary) VALUES

(12, 'Kiran Kumar', 'Business Analyst', 7, 58000);

-- Insert a department with no employees (dept_id 5 already exists)

INSERT INTO departments (dept_id, dept_name) VALUES

```
(7, 'Customer Support');


-- Left Join --
SELECT e.emp_name, e.job_title, d.dept_name
FROM employees AS e
LEFT JOIN departments AS d ON e.department_id = d.dept_id;

SELECT e.emp_name, e.job_title, d.dept_name
FROM employees AS e
LEFT JOIN departments AS d ON e.department_id = d.dept_id;

SELECT e.emp_name, e.job_title, d.dept_name
FROM employees AS e
RIGHT JOIN departments AS d ON e.department_id = d.dept_id;

SELECT e.emp_name, e.job_title, d.dept_name
FROM employees AS e
FULL OUTER JOIN departments AS d ON e.department_id = d.dept_id;



SELECT e.emp_name, e.job_title, d.dept_name
```

```
FROM employees AS e
LEFT JOIN departments AS d ON e.department_id = d.dept_id
UNION
SELECT e.emp_name, e.job_title, d.dept_name
FROM employees AS e
RIGHT JOIN departments AS d ON e.department_id = d.dept_id;

-- self join

SELECT employees.name, salaries.salary
FROM employees
INNER JOIN salaries ON employees.employee_id > salaries.employee_id;


SELECT emp_name AS name FROM employees
UNION
SELECT dept_name AS name FROM departments;

SELECT emp_name AS name FROM employees
UNION ALL
SELECT dept_name AS name FROM departments;
```