`employee` Table

emp_id	emp_name	emp_age	emp_salary	dept_id	hire_date
1	Alice	30	70000.00	1	2015-03-15
2	Bob	25	50000.00	2	2018-07-10
3	Charlie	35	80000.00	1	2012-11-03
4	David	28	60000.00	3	2019-01-22
5	Eve	40	90000.00	2	2010-05-13

`department` Table

dept_id	dept_name
1	HR
2	Engineering
3	Marketing

Find the average salary of employees in each department:

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

Retrieve the details of employees whose salary is above the average salary:

```
SELECT e.employee_id, e.name, e.age, e.department_id,
e.salary FROM employee e WHERE e.salary > (SELECT
AVG(salary) FROM employee);
```

Retrieve the employee details along with their department name

```
SELECT e.employee_id, e.name, e.age, e.salary,
d.department_name
FROM employee e
INNER JOIN department d ON e.department_id =
d.department_id;
```

Find the highest paid employee in each department:

```
SELECT
    e.employee_id,
    e.name,
    e.age,
    e.salary,
    e.department_id,
    d.department_name
FROM
    employee e
JOIN
    department d ON e.department_id = d.department_id
WHERE
    e.salary = (
        SELECT MAX(e2.salary)
        FROM employee e2
        WHERE e2.department_id = e.department_id
    );
```

Retrieve the total salary expense for each department:

```
SELECT
    d.department_id,
    d.department_name,
    SUM(e.salary) AS total_salary_expense
FROM
    employee e
JOIN
    department d ON e.department_id =
d.department_id
GROUP BY
    d.department_id, d.department_name
ORDER BY
    d.department_id;
```

Find the departments that have more than 2 employees:

```
SELECT
   d.department_id,
   d.department_name,
   COUNT(e.employee_id) AS num_employees
FROM
   department d
LEFT JOIN
   employee e ON d.department_id = e.department_id
GROUP BY
   d.department_id, d.department_name
HAVING
   COUNT(e.employee_id) > 2;
```

Find the names of employees who work in the 'HR' or 'Marketing' departments:

```
SELECT
   e.name
FROM
   employee e
JOIN
   department d ON e.department_id = d.department_id
WHERE
   d.department_name IN ('HR', 'Marketing');
```

Retrieve the details of employees who are least paid in their department:

```
SELECT e.employee_id, e.name, e.salary,
e.department_id
FROM employee e
JOIN (
    SELECT department_id, MIN(salary) AS
min_salary
    FROM employee
    GROUP BY department_id
) AS min_salaries ON e.department_id =
min_salaries.department_id AND e.salary =
min_salaries.min_salary;
```

Find the second highest salary in the company:

```
SELECT DISTINCT salary
FROM employee
ORDER BY salary DESC
LIMIT 1 OFFSET 1;
```

Find the total number of employees who have a salary between $50,000 and $80,000

```
SELECT COUNT(*) AS num_employees

FROM employee

WHERE salary BETWEEN 50000 AND 80000;
```

Retrieve the names of employees who have 'e' as the second character in their name:

```
SELECT name FROM employee WHERE name LIKE '_e%';
```

Finding Duplicate Names of employees

```
SELECT name, COUNT(*) AS name_count FROM employee
GROUP BY name HAVING COUNT(*) > 1;
```

Find the names of employees whose names contain exactly five characters:

```
SELECT name FROM employee WHERE name LIKE '_____';
```

OR

```
SELECT name FROM employee WHERE LENGTH(name) = 5;
```

Retrieve the details of employees who dont belong to departments with 'Finance' or 'IT' in their names:

```
SELECT e.employee_id, e.name, e.age, e.salary,
e.department_id FROM employee e WHERE e.department_id
NOT IN ( SELECT department_id FROM department WHERE
department_name LIKE '%Finance%' OR department_name LIKE
'%IT%' );
```

Find the names of employees whose salaries are between $45,000 and $75,000 and who are in the 'IT' department

```
SELECT e.name FROM employee e JOIN department d ON
e.department_id = d.department_id WHERE e.salary BETWEEN
45000 AND 75000 AND d.department_name = 'IT';
```

Retrieve the top 5 highest-paid employees:

```
SELECT employee_id, name, salary FROM employee ORDER
BY salary DESC LIMIT 5;
```

Add a new column `manager_id` to the `employee` table

```
ALTER TABLE employee ADD COLUMN manager_id INT;
```

Drop the `manager_id` column from the `employee` table:

```
ALTER TABLE employee DROP COLUMN manager_id;
```

Modify the `emp_salary` column in the `employee` table to set a default value of 50000:

```
ALTER TABLE employee ALTER COLUMN emp_salary SET
DEFAULT 50000;
```

Create an index on the `hire_date` column in the `employee` table:

```
CREATE INDEX idx_hire_date ON employee(hire_date);
```

Add a unique constraint on the `emp_name` column in the `employee` table:

```
ALTER TABLE employee ADD CONSTRAINT uc_emp_name
UNIQUE (emp_name);
```

Create a foreign key on the `employee` table referencing `dept_id` in the `department` table with cascading deletes

```
ALTER TABLE employee ADD CONSTRAINT
fk_employee_dept_id FOREIGN KEY (dept_id) REFERENCES
department(dept_id) ON DELETE CASCADE;
```