26 CASE STUDY - OFFICE

Problem Statement

- An office wants to maintain records of employees, departments, projects, employee assignments, attendance, and salary payments in a database
- The database can be loaded with employee and department information
- Employees can be assigned to projects
- Attendance is tracked daily for each employee
- Salary is calculated based on the attendance and the hourly rate

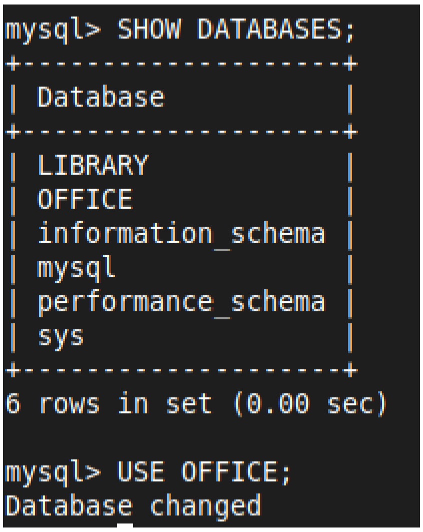
Database Schema

Entity	Attribute	Primary Key	Foreign Key
EMPLOYEE	Employee_id, Name, Department_id, Phone, Email, Date_of_joining, Salary, Status	Employee_id	Department_id
DEPARTMENT	Department_id, Name, Location	Department_id	
PROJECT	Project_id, Name, Start_date, End_date, Status	Project_id	
EMPLOYEE_PROJECT	Employee_id, Project_id, Role, Date_assigned	Employee_id, Project_id	Employee_id, Project_id
ATTENDANCE	Attendance_id, Employee_id, Date, Hours_worked	Attendance_id	Employee_id
SALARY	Salary_id, Employee_id, Month, Year, Total_hours, Total_salary	Salary_id	Employee_id

SQL - DDL

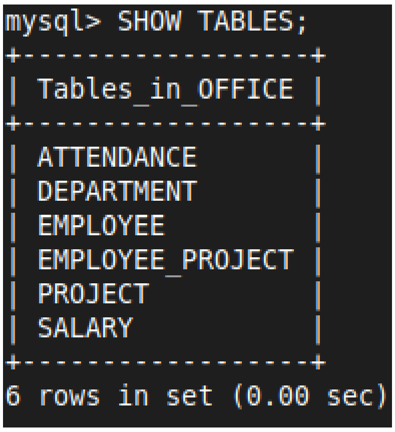
1. Create and Use Database

```
CREATE DATABASE OFFICE;
USE OFFICE;
```



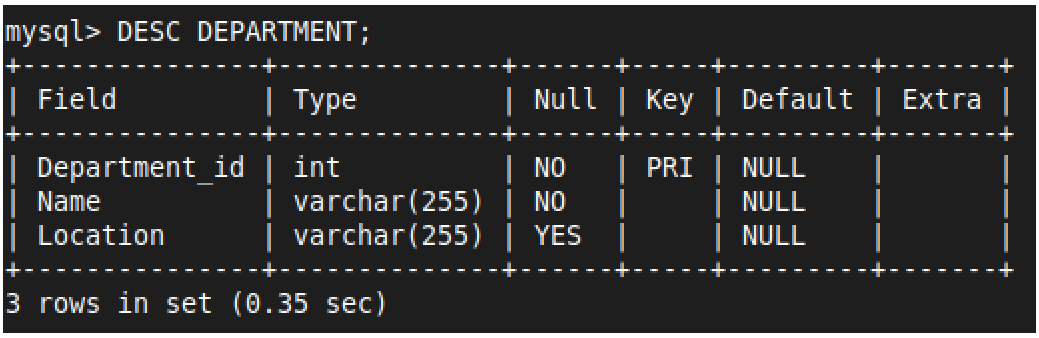
```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| LIBRARY            |
| OFFICE             |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> USE OFFICE;
Database changed
```

2. Creating Tables



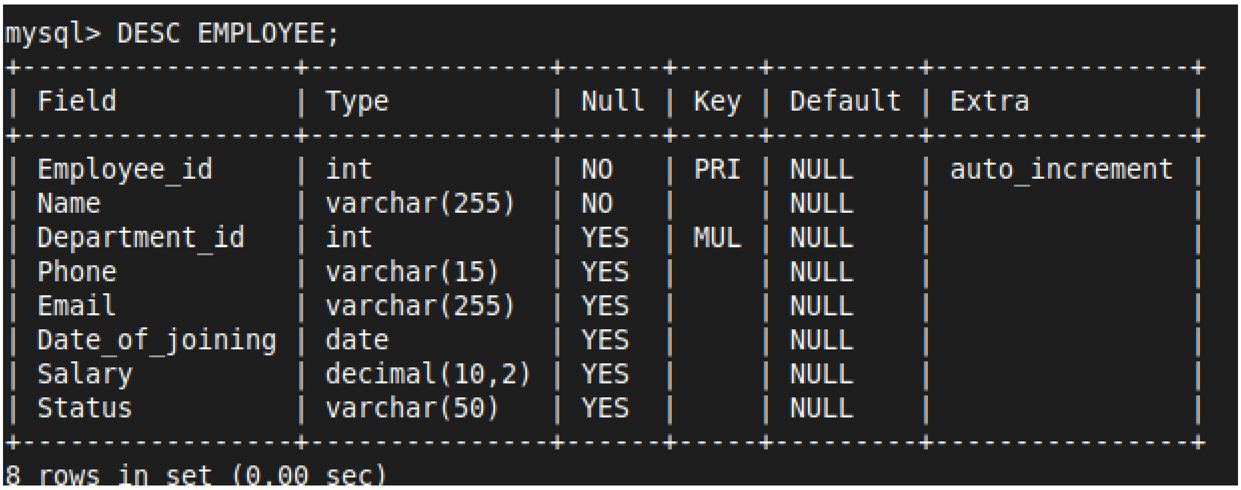
```
mysql> SHOW TABLES;
+------------------+
| Tables_in_OFFICE |
+------------------+
| ATTENDANCE       |
| DEPARTMENT       |
| EMPLOYEE         |
| EMPLOYEE_PROJECT |
| PROJECT          |
| SALARY           |
+------------------+
6 rows in set (0.00 sec)
```

```
CREATE TABLE DEPARTMENT (
    Department_id INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Location VARCHAR(255)
);
```



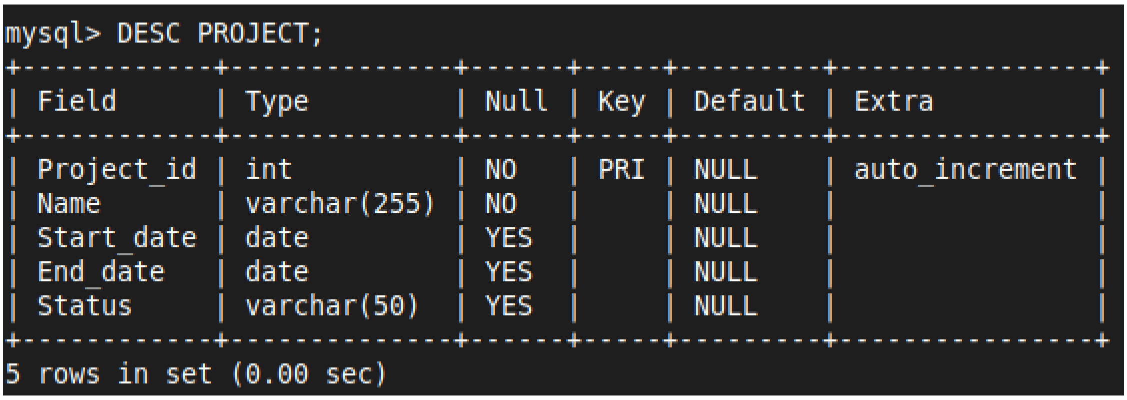
```
mysql> DESC DEPARTMENT;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Department_id | int          | NO   | PRI | NULL    |       |
| Name          | varchar(255) | NO   |     | NULL    |       |
| Location      | varchar(255) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
3 rows in set (0.35 sec)
```

```
CREATE TABLE EMPLOYEE (
    Employee_id INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Department_id INT,
    Phone VARCHAR(15),
    Email VARCHAR(255),
    Date_of_joining DATE,
    Salary DECIMAL(10, 2),
    Status VARCHAR(50),
    FOREIGN KEY (Department_id) REFERENCES DEPARTMENT(Department_id) ON
DELETE CASCADE ON UPDATE CASCADE
);
```



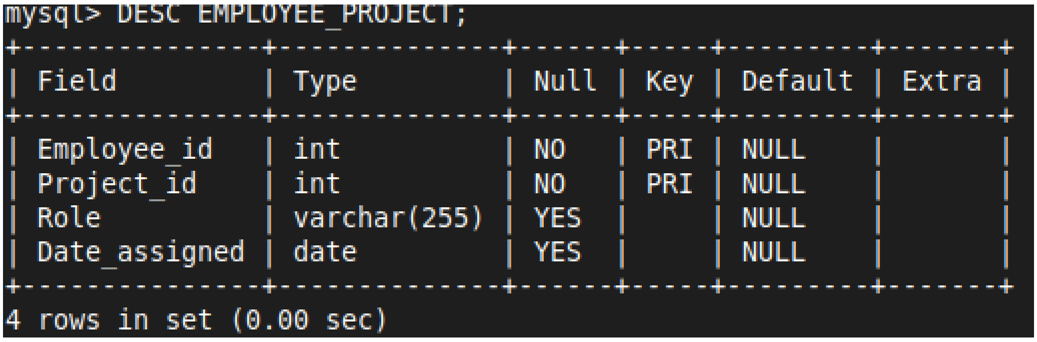
```
mysql> DESC EMPLOYEE;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| Employee_id     | int          | NO   | PRI | NULL    | auto_increment |
| Name            | varchar(255) | NO   |     | NULL    |                |
| Department_id   | int          | YES  | MUL | NULL    |                |
| Phone           | varchar(15)  | YES  |     | NULL    |                |
| Email           | varchar(255) | YES  |     | NULL    |                |
| Date_of_joining | date         | YES  |     | NULL    |                |
| Salary          | decimal(10,2)| YES  |     | NULL    |                |
| Status          | varchar(50)  | YES  |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

```
CREATE TABLE PROJECT (
    Project_id INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(255) NOT NULL,
    Start_date DATE,
    End_date DATE,
    Status VARCHAR(50)
);
```



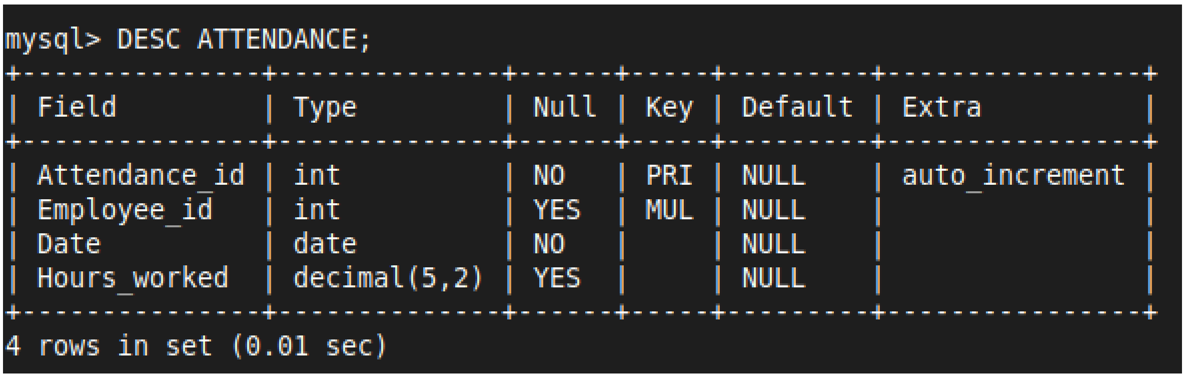
```
mysql> DESC PROJECT;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| Project_id | int          | NO   | PRI | NULL    | auto_increment |
| Name       | varchar(255) | NO   |     | NULL    |                |
| Start_date | date         | YES  |     | NULL    |                |
| End_date   | date         | YES  |     | NULL    |                |
| Status     | varchar(50)  | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
CREATE TABLE EMPLOYEE_PROJECT (
    Employee_id INT,
    Project_id INT,
    Role VARCHAR(255),
    Date_assigned DATE,
    PRIMARY KEY (Employee_id, Project_id),
    FOREIGN KEY (Employee_id) REFERENCES EMPLOYEE(Employee_id) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Project_id) REFERENCES PROJECT(Project_id) ON DELETE
CASCADE ON UPDATE CASCADE
);
```



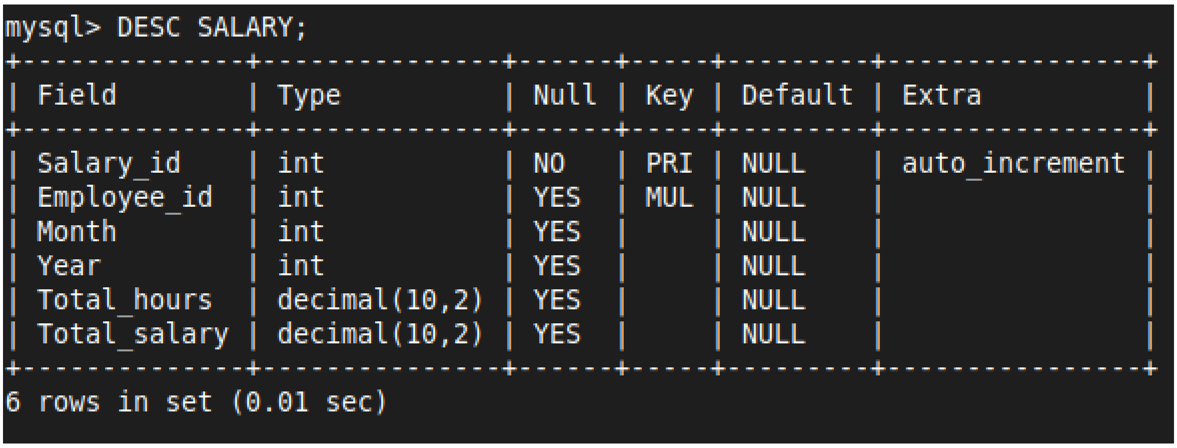
```
mysql> DESC EMPLOYEE_PROJECT;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Employee_id   | int          | NO   | PRI | NULL    |       |
| Project_id    | int          | NO   | PRI | NULL    |       |
| Role          | varchar(255) | YES  |     | NULL    |       |
| Date_assigned | date         | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
CREATE TABLE ATTENDANCE (
    Attendance_id INT PRIMARY KEY AUTO_INCREMENT,
    Employee_id INT,
    Date DATE NOT NULL,
    Hours_worked DECIMAL(5, 2),
    FOREIGN KEY (Employee_id) REFERENCES EMPLOYEE(Employee_id) ON DELETE
CASCADE ON UPDATE CASCADE
);
```



```
mysql> DESC ATTENDANCE;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| Attendance_id | int          | NO   | PRI | NULL    | auto_increment |
| Employee_id   | int          | YES  | MUL | NULL    |                |
| Date          | date         | NO   |     | NULL    |                |
| Hours_worked  | decimal(5,2) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

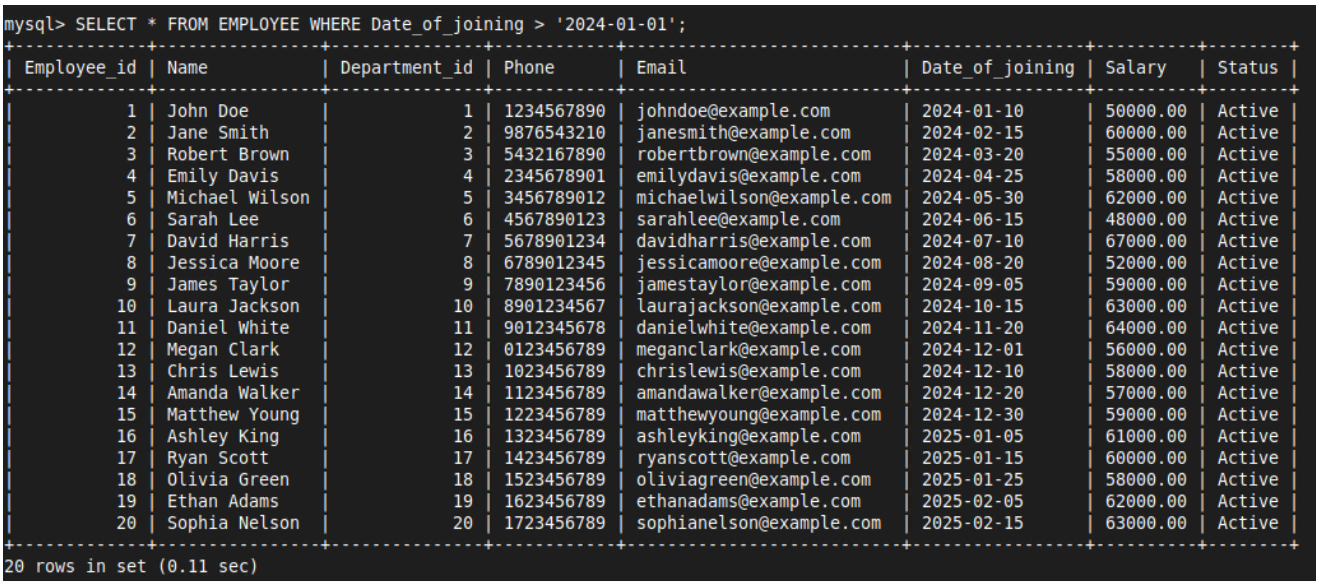
```
CREATE TABLE SALARY (
    Salary_id INT PRIMARY KEY AUTO_INCREMENT,
    Employee_id INT,
    Month INT,
    Year INT,
    Total_hours DECIMAL(10, 2),
    Total_salary DECIMAL(10, 2),
    FOREIGN KEY (Employee_id) REFERENCES EMPLOYEE(Employee_id) ON DELETE
CASCADE ON UPDATE CASCADE
);
```



```
mysql> DESC SALARY;
+--------------+---------------+------+-----+---------+----------------+
| Field        | Type          | Null | Key | Default | Extra          |
+--------------+---------------+------+-----+---------+----------------+
| Salary_id    | int           | NO   | PRI | NULL    | auto_increment |
| Employee_id  | int           | YES  | MUL | NULL    |                |
| Month        | int           | YES  |     | NULL    |                |
| Year         | int           | YES  |     | NULL    |                |
| Total_hours  | decimal(10,2) | YES  |     | NULL    |                |
| Total_salary | decimal(10,2) | YES  |     | NULL    |                |
+--------------+---------------+------+-----+---------+----------------+
6 rows in set (0.01 sec)
```

- **Select all employees who joined after 2024-01-01**

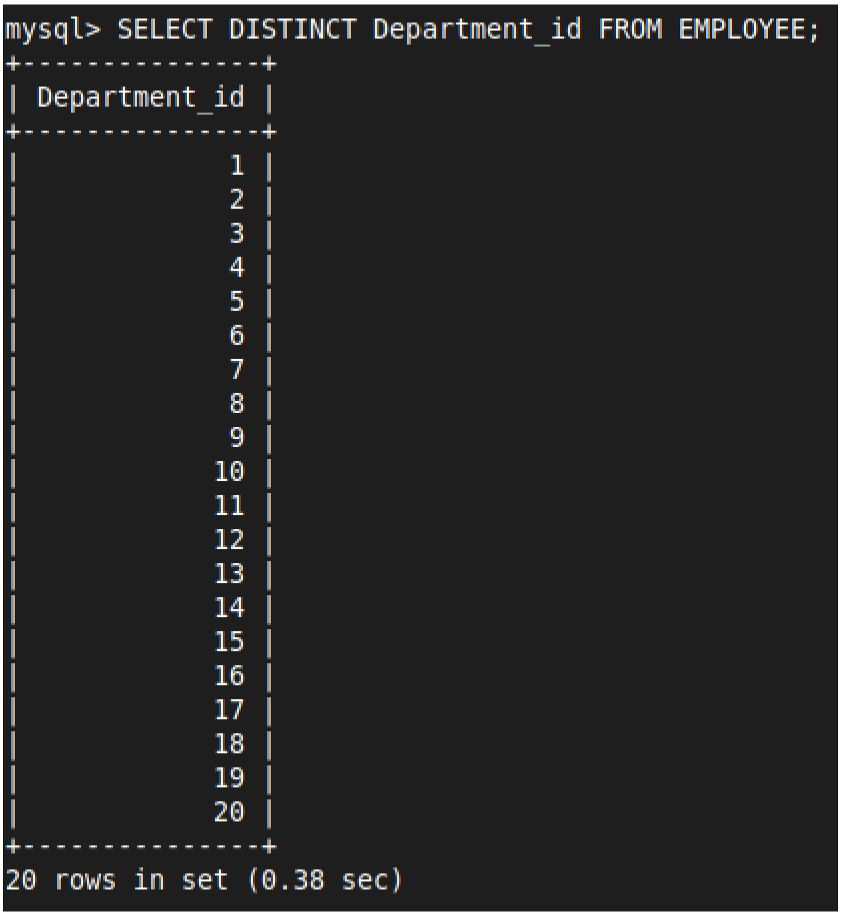
```
SELECT * FROM EMPLOYEE WHERE Date_of_joining > '2024-01-01';
```



```
mysql> SELECT * FROM EMPLOYEE WHERE Date_of_joining > '2024-01-01';
+-------------+----------------+---------------+------------+---------------------------+-----------------+----------+--------+
| Employee_id | Name           | Department_id | Phone      | Email                     | Date_of_joining | Salary   | Status |
+-------------+----------------+---------------+------------+---------------------------+-----------------+----------+--------+
|           1 | John Doe       |             1 | 1234567890 | johndoe@example.com       | 2024-01-10      | 50000.00 | Active |
|           2 | Jane Smith     |             2 | 9876543210 | janesmith@example.com     | 2024-02-15      | 60000.00 | Active |
|           3 | Robert Brown   |             3 | 5432167890 | robertbrown@example.com   | 2024-03-20      | 55000.00 | Active |
|           4 | Emily Davis    |             4 | 2345678901 | emilydavis@example.com    | 2024-04-25      | 58000.00 | Active |
|           5 | Michael Wilson |             5 | 3456789012 | michaelwilson@example.com | 2024-05-30      | 62000.00 | Active |
|           6 | Sarah Lee      |             6 | 4567890123 | sarahlee@example.com      | 2024-06-15      | 48000.00 | Active |
|           7 | David Harris   |             7 | 5678901234 | davidharris@example.com   | 2024-07-10      | 67000.00 | Active |
|           8 | Jessica Moore  |             8 | 6789012345 | jessicamoore@example.com  | 2024-08-20      | 52000.00 | Active |
|           9 | James Taylor   |             9 | 7890123456 | jamestaylor@example.com   | 2024-09-05      | 59000.00 | Active |
|          10 | Laura Jackson  |            10 | 8901234567 | laurajackson@example.com  | 2024-10-15      | 63000.00 | Active |
|          11 | Daniel White   |            11 | 9012345678 | danielwhite@example.com   | 2024-11-20      | 64000.00 | Active |
|          12 | Megan Clark    |            12 | 0123456789 | meganclark@example.com    | 2024-12-01      | 56000.00 | Active |
|          13 | Chris Lewis    |            13 | 1023456789 | chrislewis@example.com    | 2024-12-10      | 58000.00 | Active |
|          14 | Amanda Walker  |            14 | 1123456789 | amandawalker@example.com  | 2024-12-20      | 57000.00 | Active |
|          15 | Matthew Young  |            15 | 1223456789 | matthewyoung@example.com  | 2024-12-30      | 59000.00 | Active |
|          16 | Ashley King    |            16 | 1323456789 | ashleyking@example.com    | 2025-01-05      | 61000.00 | Active |
|          17 | Ryan Scott     |            17 | 1423456789 | ryanscott@example.com     | 2025-01-15      | 60000.00 | Active |
|          18 | Olivia Green   |            18 | 1523456789 | oliviagreen@example.com   | 2025-01-25      | 58000.00 | Active |
|          19 | Ethan Adams    |            19 | 1623456789 | ethanadams@example.com    | 2025-02-05      | 62000.00 | Active |
|          20 | Sophia Nelson  |            20 | 1723456789 | sophianelson@example.com  | 2025-02-15      | 63000.00 | Active |
+-------------+----------------+---------------+------------+---------------------------+-----------------+----------+--------+
20 rows in set (0.11 sec)
```

- **Find all distinct departments**

```
SELECT DISTINCT Department_id FROM EMPLOYEE;
```

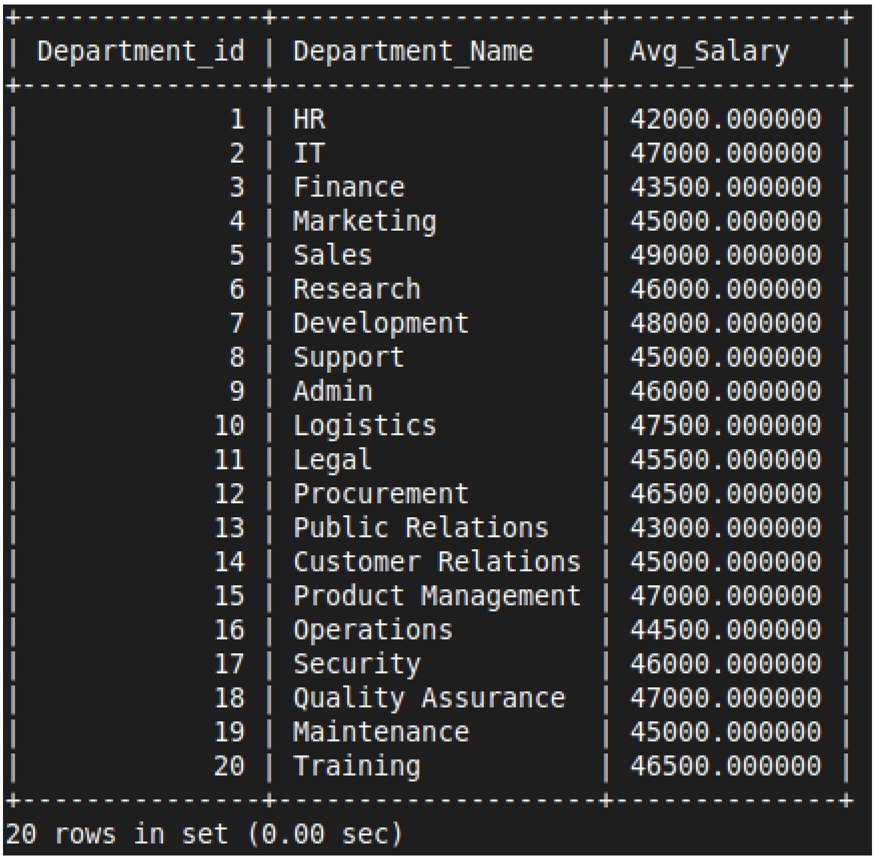


```
mysql> SELECT DISTINCT Department_id FROM EMPLOYEE;
+---------------+
| Department_id |
+---------------+
|             1 |
|             2 |
|             3 |
|             4 |
|             5 |
|             6 |
|             7 |
|             8 |
|             9 |
|            10 |
|            11 |
|            12 |
|            13 |
|            14 |
|            15 |
|            16 |
|            17 |
|            18 |
|            19 |
|            20 |
+---------------+
20 rows in set (0.38 sec)
```

- **Calculate the average salary of employees in each department**

```
SELECT Department_id, AVG(Total_salary) AS Avg_Salary
FROM SALARY
JOIN EMPLOYEE ON SALARY.Employee_id = EMPLOYEE.Employee_id
GROUP BY Department_id;
```

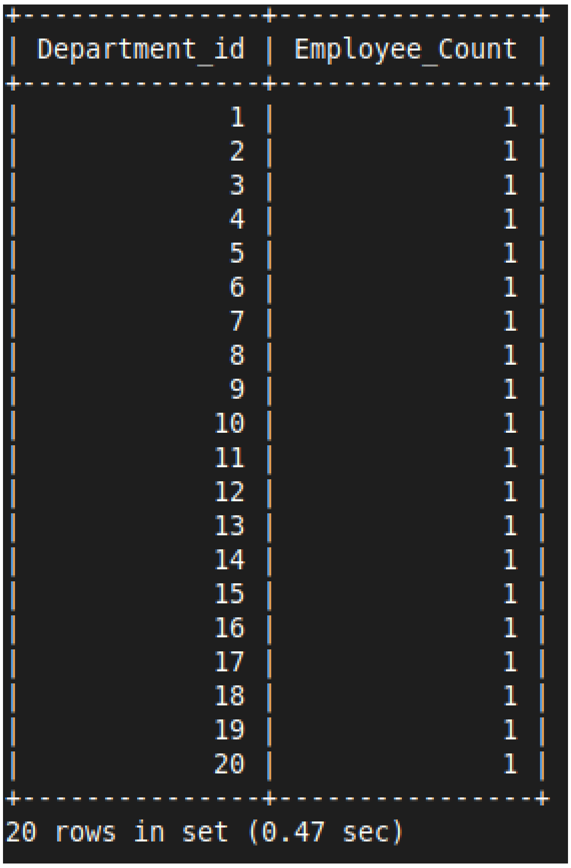
```
SELECT EMPLOYEE.Department_id, DEPARTMENT.Name AS Department_Name,
AVG(SALARY.Total_salary) AS Avg_Salary
FROM EMPLOYEE, DEPARTMENT, SALARY
WHERE EMPLOYEE.Department_id = DEPARTMENT.Department_id
AND EMPLOYEE.Employee_id = SALARY.Employee_id
GROUP BY EMPLOYEE.Department_id;
```



```
+---------------+---------------------+--------------+
| Department_id | Department_Name     | Avg_Salary   |
+---------------+---------------------+--------------+
|             1 | HR                  | 42000.000000 |
|             2 | IT                  | 47000.000000 |
|             3 | Finance             | 43500.000000 |
|             4 | Marketing           | 45000.000000 |
|             5 | Sales               | 49000.000000 |
|             6 | Research            | 46000.000000 |
|             7 | Development         | 48000.000000 |
|             8 | Support             | 45000.000000 |
|             9 | Admin               | 46000.000000 |
|            10 | Logistics           | 47500.000000 |
|            11 | Legal               | 45500.000000 |
|            12 | Procurement         | 46500.000000 |
|            13 | Public Relations    | 43000.000000 |
|            14 | Customer Relations  | 45000.000000 |
|            15 | Product Management  | 47000.000000 |
|            16 | Operations          | 44500.000000 |
|            17 | Security            | 46000.000000 |
|            18 | Quality Assurance   | 47000.000000 |
|            19 | Maintenance         | 45000.000000 |
|            20 | Training            | 46500.000000 |
+---------------+---------------------+--------------+
20 rows in set (0.00 sec)
```

- **Count the number of employees in each department**

```
SELECT Department_id, COUNT(*) AS Employee_Count
FROM EMPLOYEE
GROUP BY Department_id;
```

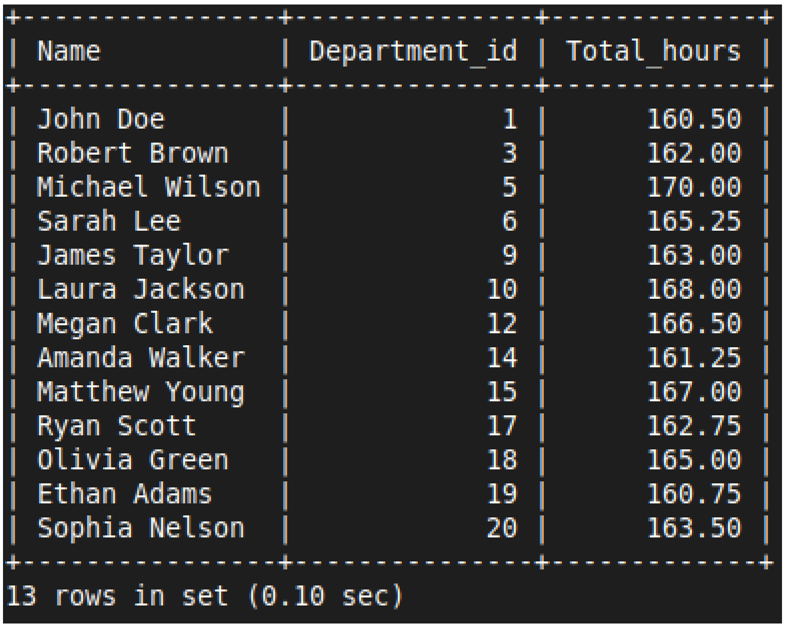


```
+---------------+----------------+
| Department_id | Employee_Count |
+---------------+----------------+
|             1 |              1 |
|             2 |              1 |
|             3 |              1 |
|             4 |              1 |
|             5 |              1 |
|             6 |              1 |
|             7 |              1 |
|             8 |              1 |
|             9 |              1 |
|            10 |              1 |
|            11 |              1 |
|            12 |              1 |
|            13 |              1 |
|            14 |              1 |
|            15 |              1 |
|            16 |              1 |
|            17 |              1 |
|            18 |              1 |
|            19 |              1 |
|            20 |              1 |
+---------------+----------------+
20 rows in set (0.47 sec)
```

- **List employees who worked more than 160 hours in a month**

```
SELECT EMPLOYEE.Name, EMPLOYEE.Department_id, SALARY.Total_hours
FROM EMPLOYEE, SALARY
WHERE EMPLOYEE.Employee_id = SALARY.Employee_id
AND SALARY.Total_hours > 160;
```

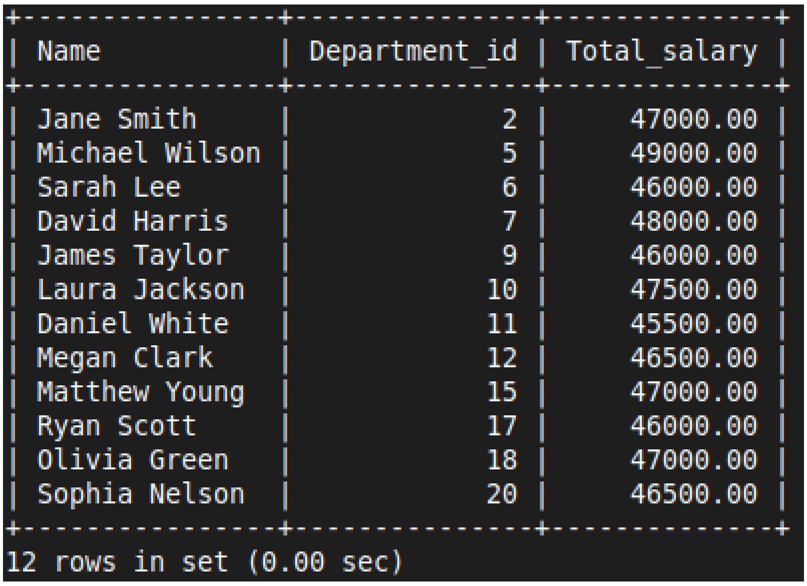
```
SELECT EMPLOYEE.Name, EMPLOYEE.Department_id, SALARY.Total_hours
FROM EMPLOYEE
JOIN SALARY ON EMPLOYEE.Employee_id = SALARY.Employee_id
WHERE SALARY.Total_hours > 160;
```



```
+----------------+---------------+-------------+
| Name           | Department_id | Total_hours |
+----------------+---------------+-------------+
| John Doe       |             1 |      160.50 |
| Robert Brown   |             3 |      162.00 |
| Michael Wilson |             5 |      170.00 |
| Sarah Lee      |             6 |      165.25 |
| James Taylor   |             9 |      163.00 |
| Laura Jackson  |            10 |      168.00 |
| Megan Clark    |            12 |      166.50 |
| Amanda Walker  |            14 |      161.25 |
| Matthew Young  |            15 |      167.00 |
| Ryan Scott     |            17 |      162.75 |
| Olivia Green   |            18 |      165.00 |
| Ethan Adams    |            19 |      160.75 |
| Sophia Nelson  |            20 |      163.50 |
+----------------+---------------+-------------+
13 rows in set (0.10 sec)
```

- **Find employees with a salary greater than 45000**

```
SELECT EMPLOYEE.Name, EMPLOYEE.Department_id, SALARY.Total_salary \FROM
EMPLOYEE, SALARY
WHERE EMPLOYEE.Employee_id = SALARY.Employee_id
AND SALARY.Total_salary > 45000;
```

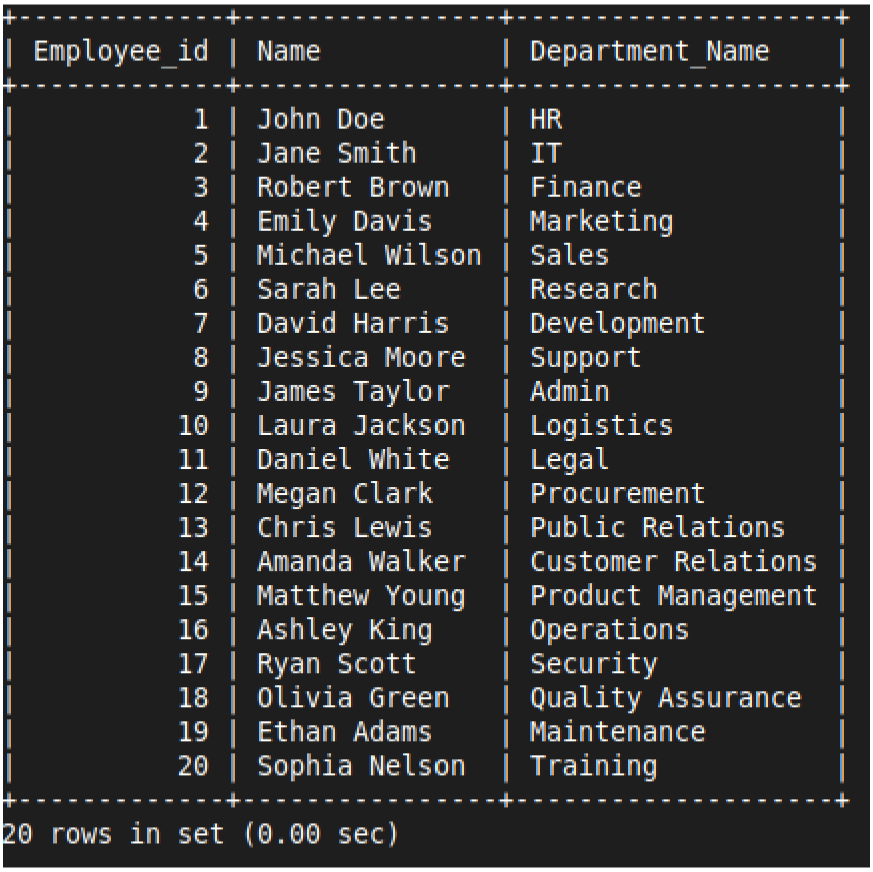


```
+----------------+---------------+--------------+
| Name           | Department_id | Total_salary |
+----------------+---------------+--------------+
| Jane Smith     |             2 |     47000.00 |
| Michael Wilson |             5 |     49000.00 |
| Sarah Lee      |             6 |     46000.00 |
| David Harris   |             7 |     48000.00 |
| James Taylor   |             9 |     46000.00 |
| Laura Jackson  |            10 |     47500.00 |
| Daniel White   |            11 |     45500.00 |
| Megan Clark    |            12 |     46500.00 |
| Matthew Young  |            15 |     47000.00 |
| Ryan Scott     |            17 |     46000.00 |
| Olivia Green   |            18 |     47000.00 |
| Sophia Nelson  |            20 |     46500.00 |
+----------------+---------------+--------------+
12 rows in set (0.00 sec)
```

- **Retrieve the employee details along with their department name**

```
SELECT E.Employee_id, E.Name, D.Name AS Department_Name
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.Department_id = D.Department_id;
```

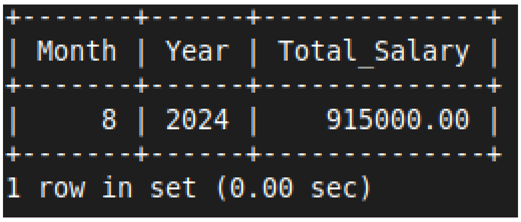
```
SELECT E.Employee_id, E.Name, D.Name AS Department_Name
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.Department_id = D.Department_id;
```



```
+-------------+----------------+--------------------+
| Employee_id | Name           | Department_Name    |
+-------------+----------------+--------------------+
|           1 | John Doe       | HR                 |
|           2 | Jane Smith     | IT                 |
|           3 | Robert Brown   | Finance            |
|           4 | Emily Davis    | Marketing          |
|           5 | Michael Wilson | Sales              |
|           6 | Sarah Lee      | Research           |
|           7 | David Harris   | Development        |
|           8 | Jessica Moore  | Support            |
|           9 | James Taylor   | Admin              |
|          10 | Laura Jackson  | Logistics          |
|          11 | Daniel White   | Legal              |
|          12 | Megan Clark    | Procurement        |
|          13 | Chris Lewis    | Public Relations   |
|          14 | Amanda Walker  | Customer Relations |
|          15 | Matthew Young  | Product Management |
|          16 | Ashley King    | Operations         |
|          17 | Ryan Scott     | Security           |
|          18 | Olivia Green   | Quality Assurance  |
|          19 | Ethan Adams    | Maintenance        |
|          20 | Sophia Nelson  | Training           |
+-------------+----------------+--------------------+
20 rows in set (0.00 sec)
```

- **Get the total salary paid to employees in each month**

```
SELECT Month, Year, SUM(Total_salary) AS Total_Salary
FROM SALARY
GROUP BY Month, Year;
```



```
+-------+------+--------------+
| Month | Year | Total_Salary |
+-------+------+--------------+
|     8 | 2024 |    915000.00 |
+-------+------+--------------+
1 row in set (0.00 sec)
```

- **Find employees who are working on more than 2 projects**

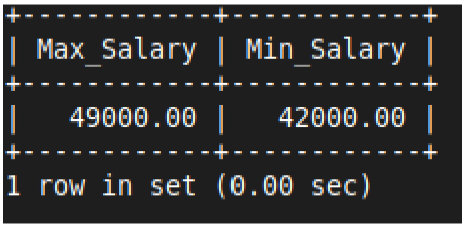
```
SELECT Employee_id, COUNT(Project_id) AS Project_Count
FROM EMPLOYEE_PROJECT
GROUP BY Employee_id
HAVING COUNT(Project_id) > 2;
```

- **List all projects and the number of employees working on each project**

```
SELECT Project_id, COUNT(Employee_id) AS Employee_Count
FROM EMPLOYEE_PROJECT
GROUP BY Project_id;
```

- **Get the maximum and minimum salary in the company**

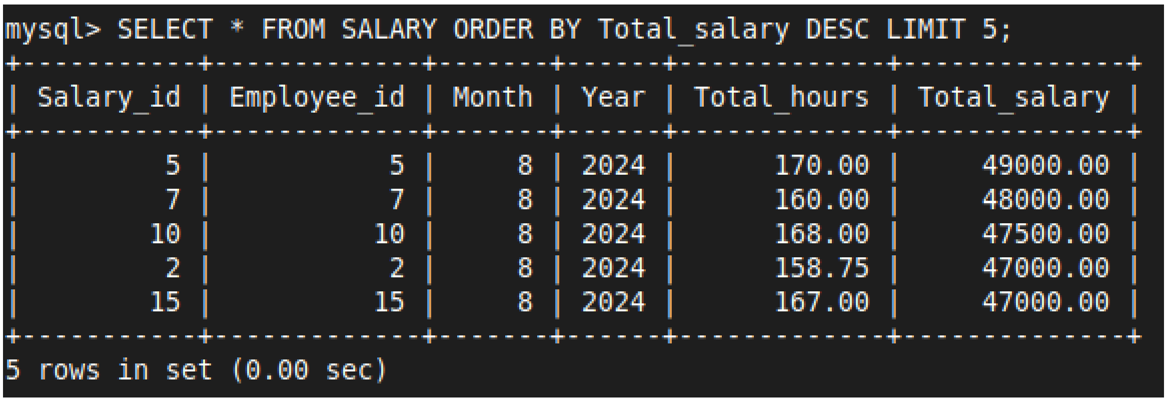
```
SELECT MAX(Total_salary) AS Max_Salary, MIN(Total_salary) AS Min_Salary
FROM SALARY;
```



```
+------------+------------+
| Max_Salary | Min_Salary |
+------------+------------+
|   49000.00 |   42000.00 |
+------------+------------+
1 row in set (0.00 sec)
```

- **Retrieve the top 5 highest-paid employees**

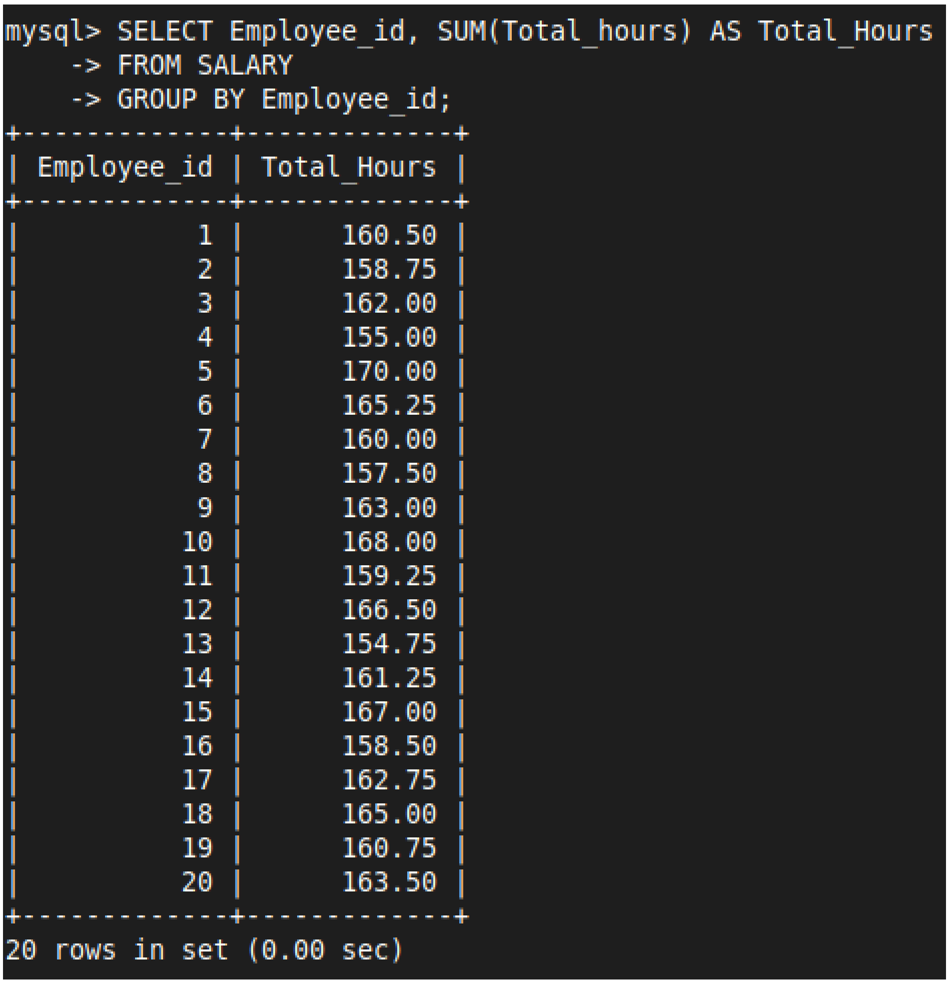
```
SELECT * FROM SALARY ORDER BY Total_salary DESC LIMIT 5;
```



```
mysql> SELECT * FROM SALARY ORDER BY Total_salary DESC LIMIT 5;
+-----------+-------------+-------+------+-------------+--------------+
| Salary_id | Employee_id | Month | Year | Total_hours | Total_salary |
+-----------+-------------+-------+------+-------------+--------------+
|         5 |           5 |     8 | 2024 |      170.00 |     49000.00 |
|         7 |           7 |     8 | 2024 |      160.00 |     48000.00 |
|        10 |          10 |     8 | 2024 |      168.00 |     47500.00 |
|         2 |           2 |     8 | 2024 |      158.75 |     47000.00 |
|        15 |          15 |     8 | 2024 |      167.00 |     47000.00 |
+-----------+-------------+-------+------+-------------+--------------+
5 rows in set (0.00 sec)
```

- **Find the total number of hours worked by each employee**

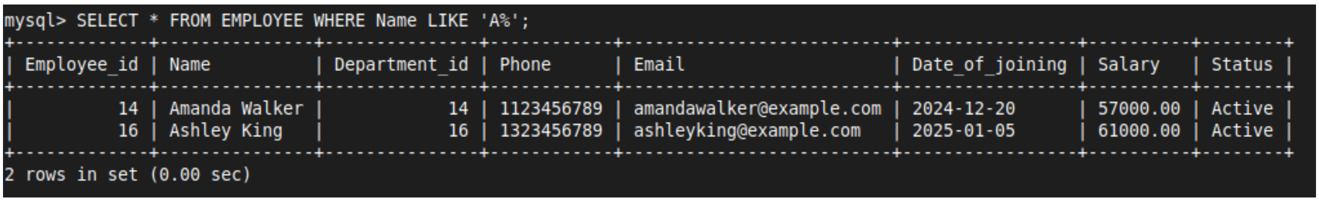
```
SELECT Employee_id, SUM(Total_hours) AS Total_Hours
FROM SALARY
GROUP BY Employee_id;
```



```
mysql> SELECT Employee_id, SUM(Total_hours) AS Total_Hours
    -> FROM SALARY
    -> GROUP BY Employee_id;
+-------------+-------------+
| Employee_id | Total_Hours |
+-------------+-------------+
|           1 |      160.50 |
|           2 |      158.75 |
|           3 |      162.00 |
|           4 |      155.00 |
|           5 |      170.00 |
|           6 |      165.25 |
|           7 |      160.00 |
|           8 |      157.50 |
|           9 |      163.00 |
|          10 |      168.00 |
|          11 |      159.25 |
|          12 |      166.50 |
|          13 |      154.75 |
|          14 |      161.25 |
|          15 |      167.00 |
|          16 |      158.50 |
|          17 |      162.75 |
|          18 |      165.00 |
|          19 |      160.75 |
|          20 |      163.50 |
+-------------+-------------+
20 rows in set (0.00 sec)
```

- **Find all employees whose name starts with 'A'**

```
SELECT * FROM EMPLOYEE WHERE Name LIKE 'A%';
```

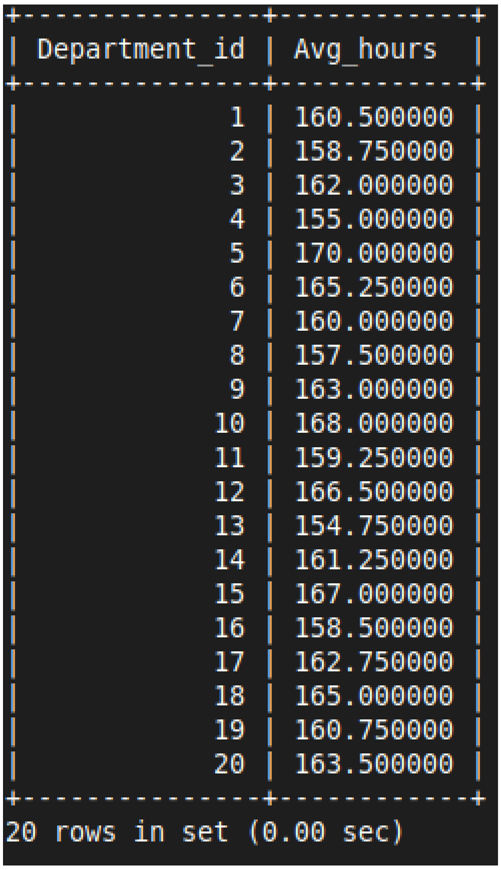


```
mysql> SELECT * FROM EMPLOYEE WHERE Name LIKE 'A%';
+-------------+---------------+---------------+------------+--------------------------+-----------------+----------+--------+
| Employee_id | Name          | Department_id | Phone      | Email                    | Date_of_joining | Salary   | Status |
+-------------+---------------+---------------+------------+--------------------------+-----------------+----------+--------+
|          14 | Amanda Walker |            14 | 1123456789 | amandawalker@example.com | 2024-12-20      | 57000.00 | Active |
|          16 | Ashley King   |            16 | 1323456789 | ashleyking@example.com   | 2025-01-05      | 61000.00 | Active |
+-------------+---------------+---------------+------------+--------------------------+-----------------+----------+--------+
2 rows in set (0.00 sec)
```

- **Calculate the average number of hours worked by employees in each department**

```
SELECT E.Department_id, AVG(S.Total_hours) AS Avg_Hours
FROM SALARY S
JOIN EMPLOYEE E ON S.Employee_id = E.Employee_id
GROUP BY E.Department_id;
```

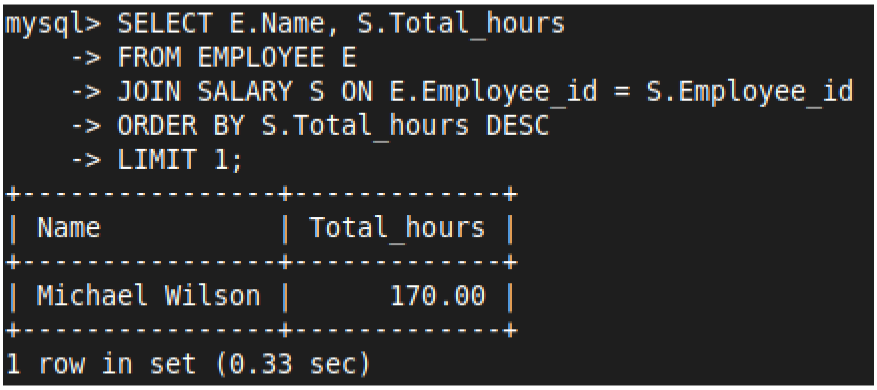
```
SELECT E.Department_id, AVG(S.Total_hours) AS Avg_hours
FROM EMPLOYEE E, SALARY S
WHERE E.Employee_id = S.Employee_id
GROUP BY E.Department_id;
```



```
+---------------+------------+
| Department_id | Avg_hours  |
+---------------+------------+
|             1 | 160.500000 |
|             2 | 158.750000 |
|             3 | 162.000000 |
|             4 | 155.000000 |
|             5 | 170.000000 |
|             6 | 165.250000 |
|             7 | 160.000000 |
|             8 | 157.500000 |
|             9 | 163.000000 |
|            10 | 168.000000 |
|            11 | 159.250000 |
|            12 | 166.500000 |
|            13 | 154.750000 |
|            14 | 161.250000 |
|            15 | 167.000000 |
|            16 | 158.500000 |
|            17 | 162.750000 |
|            18 | 165.000000 |
|            19 | 160.750000 |
|            20 | 163.500000 |
+---------------+------------+
20 rows in set (0.00 sec)
```

- **Find the employee who has worked the most hours in a single month**

```
SELECT E.Name, S.Total_hours
FROM EMPLOYEE E, SALARY S
WHERE E.Employee_id = S.Employee_id
ORDER BY S.Total_hours DESC LIMIT 1;
```



```
mysql> SELECT E.Name, S.Total_hours
    -> FROM EMPLOYEE E
    -> JOIN SALARY S ON E.Employee_id = S.Employee_id
    -> ORDER BY S.Total_hours DESC
    -> LIMIT 1;
+----------------+-------------+
| Name           | Total_hours |
+----------------+-------------+
| Michael Wilson |      170.00 |
+----------------+-------------+
1 row in set (0.33 sec)
```

- **List all departments that have more than 5 employees**

```
SELECT Department_id, COUNT(*) AS Employee_Count
FROM EMPLOYEE
GROUP BY Department_id
HAVING COUNT(*) > 5;
```

- **Find the total salary paid to each employee**

```
SELECT Employee_id, SUM(Total_salary) AS Total_Salary
FROM SALARY
GROUP BY Employee_id;
```

- **Retrieve the employee details along with the total number of projects they are working on**

```
SELECT E.Employee_id, E.Name, COUNT(EP.Project_id) AS Project_Count
FROM EMPLOYEE E
LEFT JOIN EMPLOYEE_PROJECT EP ON E.Employee_id = EP.Employee_id
GROUP BY E.Employee_id, E.Name;
```

- **Get the total number of hours worked by employees in each project**

```
SELECT EP.Project_id, SUM(S.Total_hours) AS Total_Hours
FROM EMPLOYEE_PROJECT EP
JOIN SALARY S ON EP.Employee_id = S.Employee_id
GROUP BY EP.Project_id;
```

- **Find all employees who have not worked on any project**

```
SELECT * FROM EMPLOYEE WHERE Employee_id NOT IN (SELECT DISTINCT
Employee_id FROM EMPLOYEE_PROJECT);
```

- **Calculate the total salary paid to employees in each department**

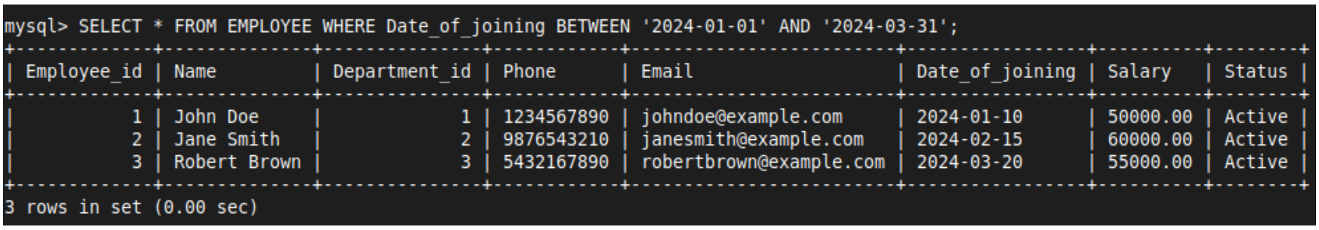
```
SELECT E.Department_id, SUM(S.Total_salary) AS Total_Salary
FROM SALARY S
JOIN EMPLOYEE E ON S.Employee_id = E.Employee_id
GROUP BY E.Department_id;
```

- **List employees along with their total hours worked in each month**

```
SELECT E.Employee_id, E.Name, S.Month, S.Year, SUM(S.Total_hours) AS
Total_Hours
FROM EMPLOYEE E
JOIN SALARY S ON E.Employee_id = S.Employee_id
GROUP BY E.Employee_id, E.Name, S.Month, S.Year;
```

- **Retrieve employees who joined between January and March 2024**

```
SELECT * FROM EMPLOYEE WHERE Date_of_joining BETWEEN '2024-01-01' AND
'2024-03-31';
```



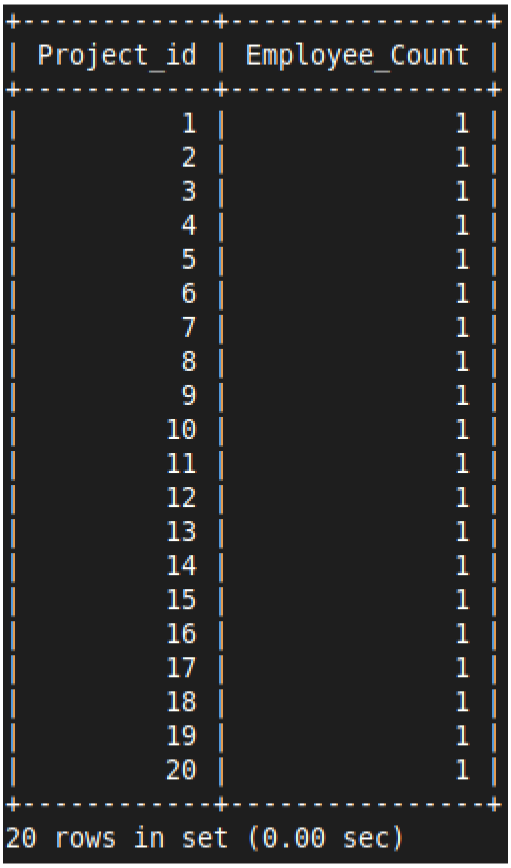
```
mysql> SELECT * FROM EMPLOYEE WHERE Date_of_joining BETWEEN '2024-01-01' AND '2024-03-31';
+-------------+--------------+---------------+------------+-------------------------+-----------------+----------+--------+
| Employee_id | Name         | Department_id | Phone      | Email                   | Date_of_joining | Salary   | Status |
+-------------+--------------+---------------+------------+-------------------------+-----------------+----------+--------+
|           1 | John Doe     |             1 | 1234567890 | johndoe@example.com     | 2024-01-10      | 50000.00 | Active |
|           2 | Jane Smith   |             2 | 9876543210 | janesmith@example.com   | 2024-02-15      | 60000.00 | Active |
|           3 | Robert Brown |             3 | 5432167890 | robertbrown@example.com | 2024-03-20      | 55000.00 | Active |
+-------------+--------------+---------------+------------+-------------------------+-----------------+----------+--------+
3 rows in set (0.00 sec)
```

- **List all employees and their respective department names**

```
SELECT E.Name, D.Name AS Department_Name
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.Department_id = D.Department_id;
```

- **Get the total number of employees in each project**

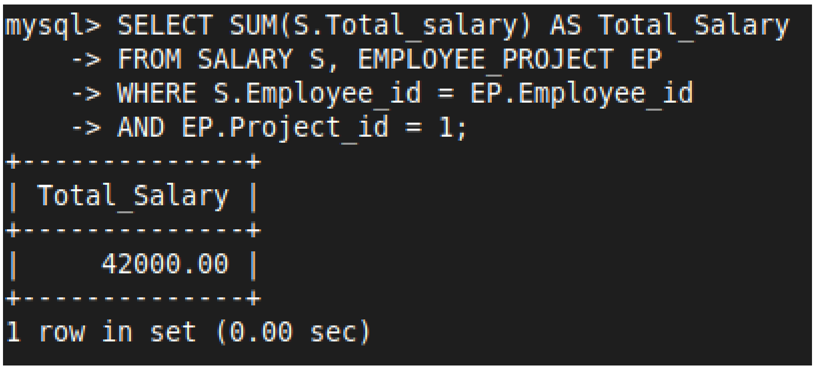
```
SELECT P.Project_id, COUNT(EP.Employee_id) AS Employee_Count
FROM PROJECT P
LEFT JOIN EMPLOYEE_PROJECT EP ON P.Project_id = EP.Project_id
GROUP BY P.Project_id;
```



```
+------------+----------------+
| Project_id | Employee_Count |
+------------+----------------+
|          1 |              1 |
|          2 |              1 |
|          3 |              1 |
|          4 |              1 |
|          5 |              1 |
|          6 |              1 |
|          7 |              1 |
|          8 |              1 |
|          9 |              1 |
|         10 |              1 |
|         11 |              1 |
|         12 |              1 |
|         13 |              1 |
|         14 |              1 |
|         15 |              1 |
|         16 |              1 |
|         17 |              1 |
|         18 |              1 |
|         19 |              1 |
|         20 |              1 |
+------------+----------------+
20 rows in set (0.00 sec)
```

- **Find the total salary of employees who worked on project 1**

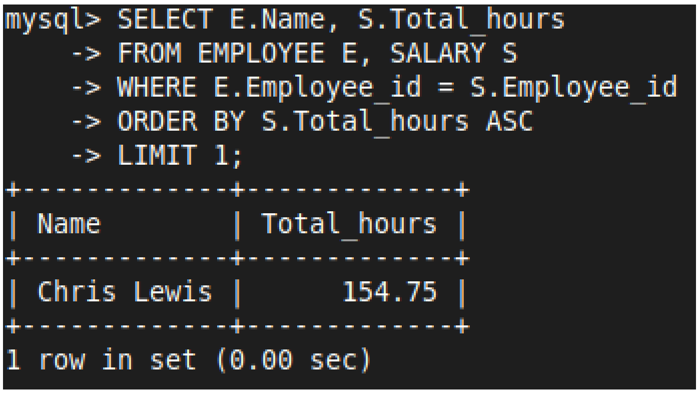
```
SELECT SUM(S.Total_salary) AS Total_Salary
FROM SALARY S
JOIN EMPLOYEE_PROJECT EP ON S.Employee_id = EP.Employee_id
WHERE EP.Project_id = 1;
```



```
mysql> SELECT SUM(S.Total_salary) AS Total_Salary
    -> FROM SALARY S, EMPLOYEE_PROJECT EP
    -> WHERE S.Employee_id = EP.Employee_id
    -> AND EP.Project_id = 1;
+--------------+
| Total_Salary |
+--------------+
|     42000.00 |
+--------------+
1 row in set (0.00 sec)
```

- **Find the employee who has worked the least hours in a single month**

```
SELECT E.Name, S.Total_hours
FROM EMPLOYEE E, SALARY S
WHERE E.Employee_id = S.Employee_id
ORDER BY S.Total_hours ASC LIMIT 1;
```



```
mysql> SELECT E.Name, S.Total_hours
    -> FROM EMPLOYEE E, SALARY S
    -> WHERE E.Employee_id = S.Employee_id
    -> ORDER BY S.Total_hours ASC
    -> LIMIT 1;
+-------------+-------------+
| Name        | Total_hours |
+-------------+-------------+
| Chris Lewis |      154.75 |
+-------------+-------------+
1 row in set (0.00 sec)
```

- **Retrieve the list of all employees, their department, and the total salary they earned**

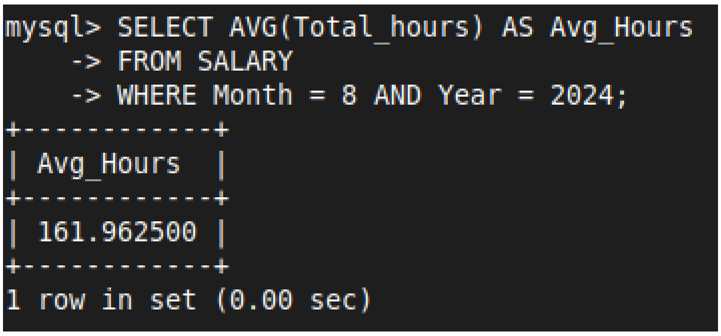
```sql
SELECT E.Employee_id, E.Name, D.Name AS Department_Name,
SUM(S.Total_salary) AS Total_Salary
FROM EMPLOYEE E
JOIN DEPARTMENT D ON E.Department_id = D.Department_id
JOIN SALARY S ON E.Employee_id = S.Employee_id
GROUP BY E.Employee_id, E.Name, D.Name;
```

- **Find employees who worked on both project 1 and project 2**

```sql
SELECT Employee_id
FROM EMPLOYEE_PROJECT
WHERE Project_id IN (1, 2)
GROUP BY Employee_id
HAVING COUNT(DISTINCT Project_id) = 2;
```

- **Calculate the average number of hours worked by employees in August 2024**

```sql
SELECT AVG(Total_hours) AS Avg_Hours
FROM SALARY
WHERE Month = 8 AND Year = 2024;
```



```
mysql> SELECT AVG(Total_hours) AS Avg_Hours
    -> FROM SALARY
    -> WHERE Month = 8 AND Year = 2024;
+------------+
| Avg_Hours  |
+------------+
| 161.962500 |
+------------+
1 row in set (0.00 sec)
```

- **Find employees who earned the same salary in two different months**

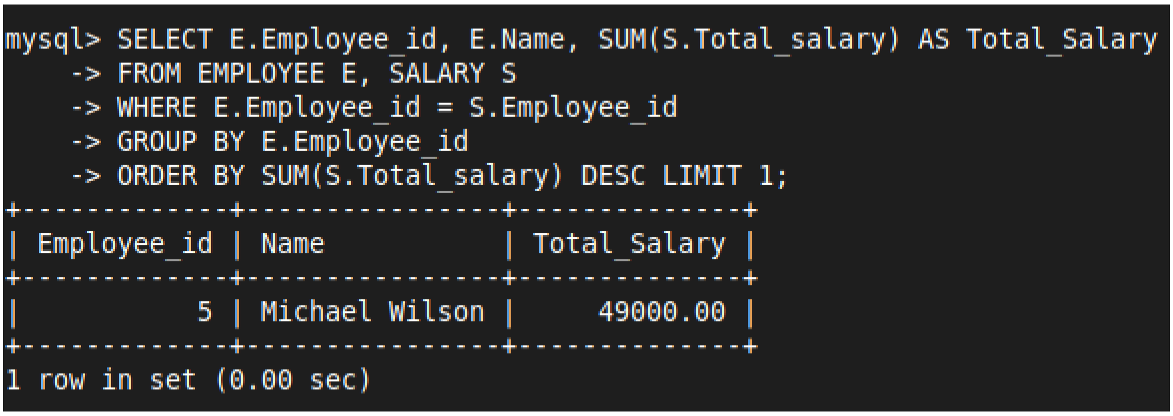
```
SELECT Employee_id, Total_salary
FROM SALARY
GROUP BY Employee_id, Total_salary
HAVING COUNT(*) > 1;
```

- **List all projects and the total salary paid to employees working on each project**

```
SELECT EP.Project_id, SUM(S.Total_salary) AS Total_Salary
FROM EMPLOYEE_PROJECT EP
JOIN SALARY S ON EP.Employee_id = S.Employee_id
GROUP BY EP.Project_id;
```

- **Get the employee details with the highest total salary over all months**

```
SELECT E.Employee_id, E.Name, SUM(S.Total_salary) AS Total_Salary
FROM EMPLOYEE E
JOIN SALARY S ON E.Employee_id = S.Employee_id
GROUP BY E.Employee_id, E.Name
ORDER BY SUM(S.Total_salary) DESC LIMIT 1;
```



```
mysql> SELECT E.Employee_id, E.Name, SUM(S.Total_salary) AS Total_Salary
    -> FROM EMPLOYEE E, SALARY S
    -> WHERE E.Employee_id = S.Employee_id
    -> GROUP BY E.Employee_id
    -> ORDER BY SUM(S.Total_salary) DESC LIMIT 1;
+-------------+----------------+--------------+
| Employee_id | Name           | Total_Salary |
+-------------+----------------+--------------+
|           5 | Michael Wilson |     49000.00 |
+-------------+----------------+--------------+
1 row in set (0.00 sec)
```