

# **EMPLOYEES DATABASE MANAGEMENT SYSTEM**

# INTRODUCTION

The Employee Database Management System helps to store and manage employees, department, project, and work-hour details in an organized way.

# PROJECT GOALS

- The goal of this project is to design a structured database that stores employee, department, project, and work-hour information.
- The project helps understand real-world database operations like creating tables, inserting data, and analyzing employee performance.

# DATABASE SCHEME

## TABLES INCLUDE:

- Departments
- Employees
- Project
- Timesheet

# DATA INSERT:

14	14:21:40	desc Employees	8 row(s) returned
15	14:24:44	desc departments	3 row(s) returned
16	14:25:19	desc Projects	6 row(s) returned
17	14:25:35	desc Timesheets	5 row(s) returned

22	14:39:22	INSERT INTO Departments (D_name, location) VALUES ('HR', 'New York'), ('Finance', 'Chicago'), ('Engineering', 'San Fr...
23	14:39:57	INSERT INTO Employees (first_name, last_name, email, phone_number, hire_date, job_title, department_id) VALUES ('John', 'Doe', 'j...
24	14:40:18	INSERT INTO Projects (P_name, P_Description, start_date, end_date, department_id) VALUES ('Project A', 'HR system overhaul', '2...
25	14:40:32	INSERT INTO Timesheets (employee_id, project_id, date_ts, hours) VALUES (1, 1, '2023-02-15', 8.0), (2, 3, '2023-05-10', 6.5), (2, 3, '...

# 1. Find out the working location of each department.

```
105      #1 fine out the working location of each employess
106 •  SELECT
107      e.first_name,
108      e.last_name,
109      d.location AS working_location
110  FROM Employees e
111  JOIN Departments d
112    ON e.department_id = d.id;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	first_name	last_name	working_location
▶	John	Doe	New York
	Jane	Smith	San Francisco
	Tom	Brown	Chicago
	Lucy	Davis	Los Angeles
	Mark	Taylor	Boston
	Emma	Watson	San Diego
	David	Miller	San Diego
	Sophia	Wilson	Seattle
	Liam	Johnson	Seattle
	Olivia	Martinez	New York
	Lucas	Garcia	Los Angeles
	Mia	Davis	Boston
	James	Brown	Houston
	Isabella	Rodriguez	Miami

Result 10 × Read Only

2.create salary column and update the salary of each department.

```
SQL File 3: X |           Limit to 1000 rows |      
113      # 2.create salary column and update the salaries for each department  
114 • ALTER TABLE Employees  
115      ADD salary INT;  
116 • UPDATE Employees SET salary = 40000 WHERE department_id = 1; -- D1  
117 • UPDATE Employees SET salary = 30000 WHERE department_id = 2; -- D2  
118 • UPDATE Employees SET salary = 55000 WHERE department_id = 3; -- D3  
119 • UPDATE Employees SET salary = 60000 WHERE department_id = 4; -- D4  
120 • UPDATE Employees SET salary = 75000 WHERE department_id = 5; -- D5  
121 • UPDATE Employees SET salary = 100000 WHERE department_id = 6; -- D6  
122 • UPDATE Employees SET salary = 125000 WHERE department_id = 7; -- D7  
123 • UPDATE Employees SET salary = 130000 WHERE department_id = 8; -- D8  
124 • UPDATE Employees SET salary = 150000 WHERE department_id = 9; -- D9  
125 • select* from Employees;
```

### 3. Find which project the employee is working on.

```
126      #3 find which project the employee is working on
127 •  SELECT CONCAT(e.first_name, ' ', e.last_name) AS employee_name,
128          p.P_name AS project_name
129  FROM Employees e
130  JOIN Timesheets t
131      ON e.id = t.employee_id
132  JOIN Projects p
133      ON t.project_id = p.id;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	employee_name	project_name
▶	John Doe	Project A
	John Doe	Project C
	John Doe	Project C
	Jane Smith	Project C
	Jane Smith	Project C
	Jane Smith	Project C
	Tom Brown	Project B
	Lucy Davis	Project D
	Lucy Davis	Project D
	Mark Taylor	Project E
	Emma Watson	Project F
	David Miller	Project F

## 4. Find out the total hour worked on each project by employee.

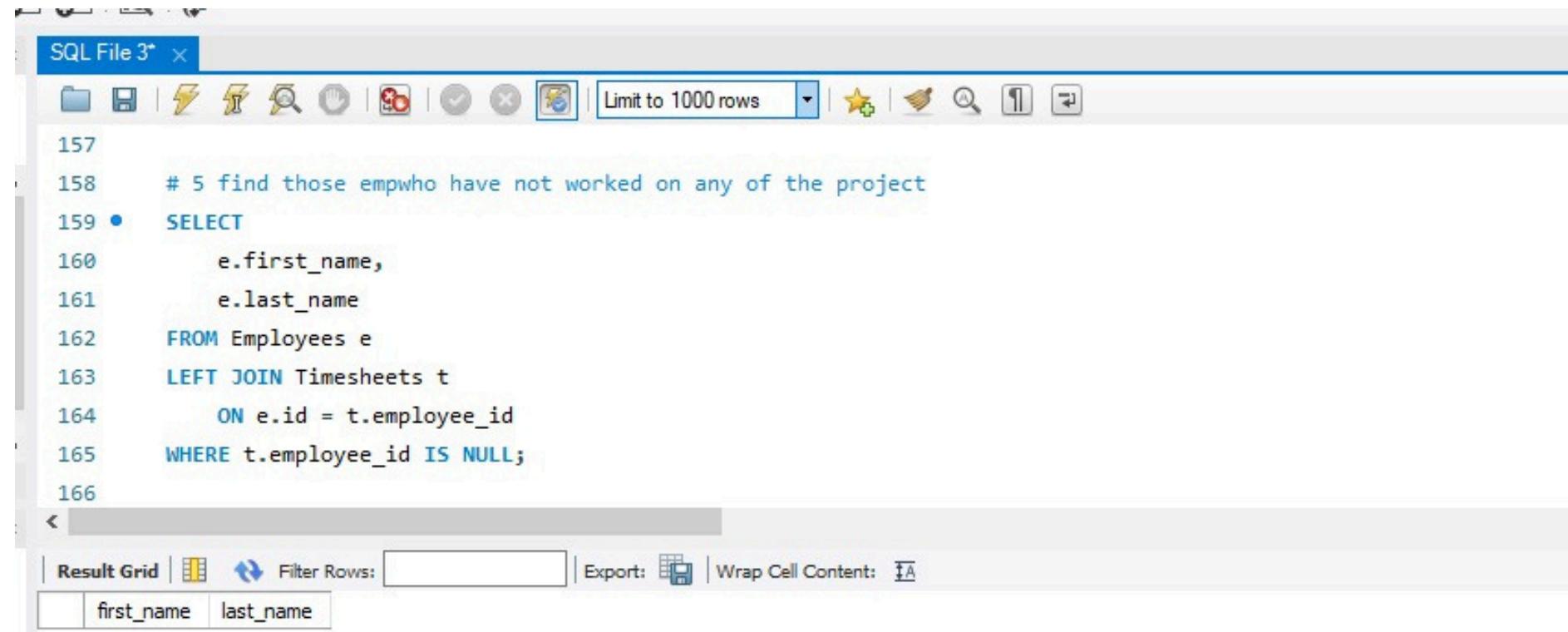
```
34      # 4 find out the total hr worked on each project by each emp
35 •  SELECT e.first_name,
36          e.last_name,
37          p.P_name AS project_name,
38          SUM(t.hours) AS total_hours
39      FROM Employees e
40      JOIN Timesheets t
41          ON e.id = t.employee_id
42      JOIN Projects p
43          ON t.project_id = p.id
```

result Grid | Filter Rows:  | Export: | Wrap Cell Content:

first_name	last_name	project_name	total_hours
John	Doe	Project A	8.00
John	Doe	Project C	15.50
Jane	Smith	Project C	20.50
Tom	Brown	Project B	8.00
Lucy	Davis	Project D	13.00
Mark	Taylor	Project E	6.00
Emma	Watson	Project F	7.50
David	Miller	Project F	14.50
Sophia	Wilson	Project G	8.00

Result 1 x Read Input

## 5. Find those employee who have not worked on any of the project.

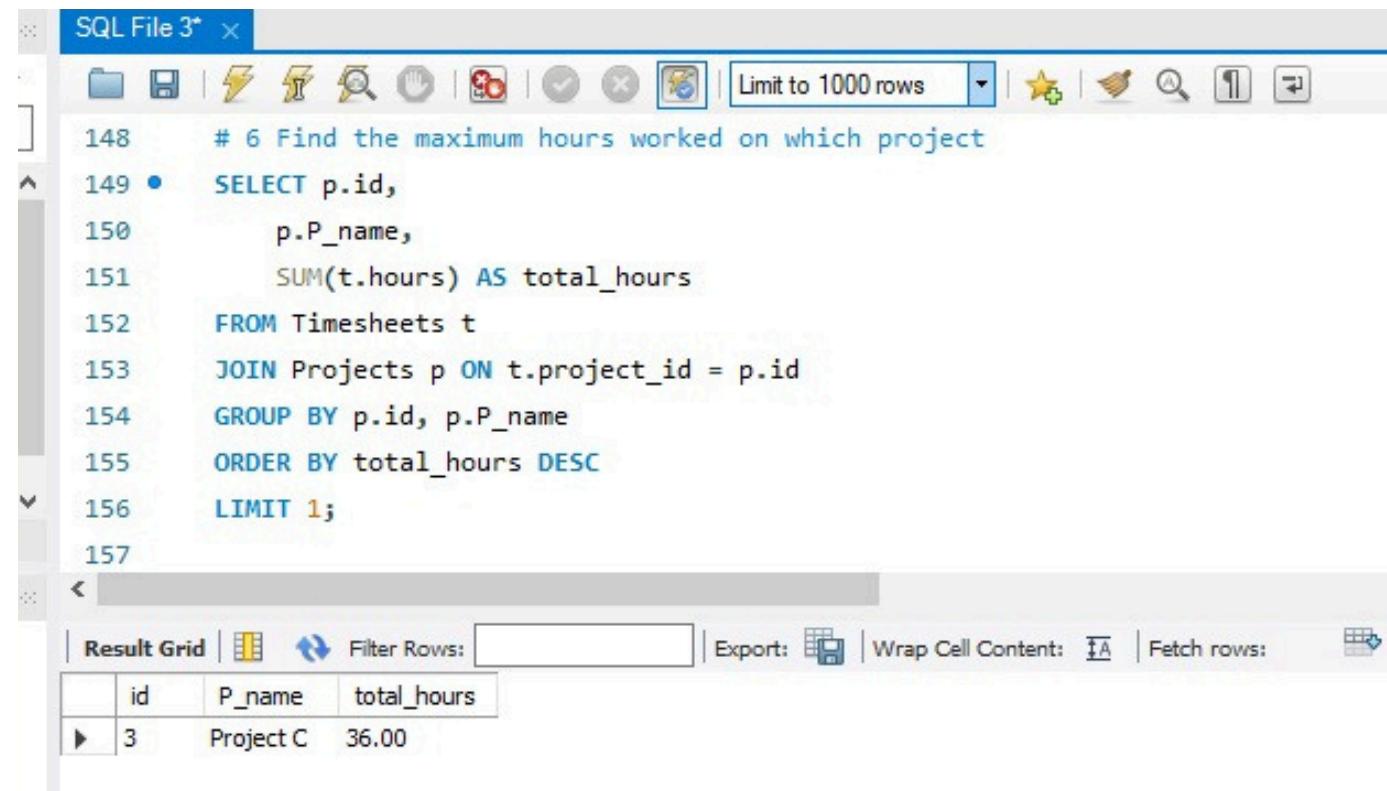


The screenshot shows a SQL editor window titled "SQL File 3\*". The code is as follows:

```
157
158  # 5 find those empwho have not worked on any of the project
159 • SELECT
160     e.first_name,
161     e.last_name
162 FROM Employees e
163 LEFT JOIN Timesheets t
164     ON e.id = t.employee_id
165 WHERE t.employee_id IS NULL;
166
```

The code uses a LEFT JOIN between the Employees table and the Timesheets table, filtering for employees who have no entries in the Timesheets table.

## 6. Find the maximum hours worked on which project.



The screenshot shows a SQL editor window titled "SQL File 3\*". The code is as follows:

```
148  # 6 Find the maximum hours worked on which project
149 • SELECT p.id,
150     p.P_name,
151     SUM(t.hours) AS total_hours
152 FROM Timesheets t
153 JOIN Projects p ON t.project_id = p.id
154 GROUP BY p.id, p.P_name
155 ORDER BY total_hours DESC
156 LIMIT 1;
157
```

The code joins the Timesheets table with the Projects table, groups by project ID and name, and orders by total hours in descending order, limiting the result to one row.

	id	P_name	total_hours
▶	3	Project C	36.00

# 7.create a view which store employees department name and their salary.

The screenshot shows a database management interface with two main sections. The top section is a code editor with the following SQL script:

```
10 •    INSERT INTO Timesheets (employee_id, project_id, date_ts, hours) VALUES (1, 1, '2023-02-15', 8.0), (2, 3, 11 • #Q7
12 •    SELECT * FROM Employee_Department_Salary;
13 •    ALTER TABLE Employees ADD salary DECIMAL(10,2);
14 •    CREATE VIEW Employee_Department_Salary AS
15     SELECT
16         e.id AS employee_id, e.first_name, e.last_name, d.D_name AS department_name,
17         e.salary
18     FROM Employees e
19     JOIN Departments d
20     ON e.department_id = d.id;
21
```

The bottom section is a grid view displaying employee data. The columns are: employee\_id, first\_name, last\_name, department\_name, and salary. The salary column contains NULL values for all employees.

employee_id	first_name	last_name	department_name	salary
	John	Doe	HR	NULL
	Olivia	Martinez	HR	NULL
	Tom	Brown	Finance	NULL
	Jane	Smith	Engineering	NULL
	Lucy	Davis	Marketing	NULL
	Lucas	Garcia	Marketing	NULL
	Mark	Taylor	Sales	NULL
	Mia	Davis	Sales	NULL
	Emma	Watson	IT Support	NULL
	David	Miller	IT Support	NULL
	Sophia	Wilson	Legal	NULL
	Liam	Johnson	Legal	NULL
	James	Brown	Operations	NULL
	Isabella	Rodriguez	Customer Service	NULL

## 8.create a view which stores the project allocated to each employee.

Employee Work Log Data						
employee_id	first_name	last_name	project_id	project_name	hours	date_ts
3	Tom	Brown	2	Project B	8.00	2023-03-20
3	Tom	Brown	2	Project B	8.00	2023-03-20
3	Tom	Brown	2	Project B	8.00	2023-03-20
4	Lucy	Davis	4	Project D	5.50	2023-08-10
4	Lucy	Davis	4	Project D	7.50	2023-08-11
4	Lucy	Davis	4	Project D	5.50	2023-08-10
4	Lucy	Davis	4	Project D	7.50	2023-08-11
4	Lucy	Davis	4	Project D	5.50	2023-08-10
4	Lucy	Davis	4	Project D	7.50	2023-08-11
5	Mark	Taylor	5	Project E	6.00	2023-10-15
5	Mark	Taylor	5	Project E	6.00	2023-10-15
5	Mark	Taylor	5	Project E	6.00	2023-10-15
6	Emma	Watson	6	Project F	7.50	2023-06-15
6	Emma	Watson	6	Project F	7.50	2023-06-15
6	Emma	Watson	6	Project F	7.50	2023-06-15
7	David	Miller	6	Project F	8.00	2023-06-16
7	David	Miller	6	Project F	6.50	2023-06-17
7	David	Miller	6	Project F	8.00	2023-06-16
7	David	Miller	6	Project F	6.50	2023-06-17
7	David	Miller	6	Project F	8.00	2023-06-16
7	David	Miller	6	Project F	6.50	2023-06-17
Employee Work Log Data						
employee_id	first_name	last_name	project_id	project_name	hours	date_ts
11	Lucas	Garcia	4	Project D	6.00	2023-05-22
11	Lucas	Garcia	4	Project D	5.50	2023-05-23
11	Lucas	Garcia	4	Project D	6.00	2023-05-22
11	Lucas	Garcia	4	Project D	5.50	2023-05-23
11	Lucas	Garcia	4	Project D	6.00	2023-05-22
11	Lucas	Garcia	4	Project D	5.50	2023-05-23
12	Mia	Davis	5	Project E	7.00	2023-09-15
12	Mia	Davis	5	Project E	7.00	2023-09-15
12	Mia	Davis	5	Project E	7.00	2023-09-15
13	James	Brown	8	Project H	6.50	2023-07-05
13	James	Brown	8	Project H	8.00	2023-07-06
13	James	Brown	8	Project H	6.50	2023-07-05
13	James	Brown	8	Project H	8.00	2023-07-06
13	James	Brown	8	Project H	6.50	2023-07-05
13	James	Brown	8	Project H	8.00	2023-07-06
14	Isabella	Rodriguez	9	Project I	6.00	2023-07-10
14	Isabella	Rodriguez	9	Project I	6.50	2023-07-11
14	Isabella	Rodriguez	9	Project I	6.00	2023-07-10
14	Isabella	Rodriguez	9	Project I	6.50	2023-07-11
14	Isabella	Rodriguez	9	Project I	6.00	2023-07-10
14	Isabella	Rodriguez	9	Project I	6.50	2023-07-11

## 9.Find the employee who have worked more than 20 hour on a single project.

LECT

```
SELECT  
    e.id AS employee_id, e.first_name, e.last_name, p.P_name AS project_name, SUM(t.hours) AS total_hours  
    FROM Timesheets t JOIN Employees e ON t.employee_id = e.id JOIN Projects p ON t.project_id = p.id  
    GROUP BY e.id, p.id HAVING SUM(t.hours) > 20;
```

Result Grid					
	employee_id	first_name	last_name	project_name	total_hours
▶	2	Jane	Smith	Project C	20.50

## 10.create a query that classified employee based on their job titles as ‘Manager’, ‘Developer’ or other.

•

```
SELECT  
    first_name,  
    last_name,  
    job_title,  
    CASE  
        WHEN job_title LIKE '%Manager%' THEN 'Manager'  
        WHEN job_title LIKE '%Developer%' THEN 'Developer'  
        ELSE 'Other'  
    END AS job_category  
    FROM Employees;
```

	first_name	last_name	job_title	job_category
▶	John	Doe	Manager	Manager
	Jane	Smith	Developer	Developer
	Tom	Brown	Manager	Manager
	Lucy	Davis	Developer	Developer
	Mark	Taylor	Sales Rep	Other
	Emma	Watson	Manager	Manager
	David	Miller	Developer	Developer
	Sophia	Wilson	Manager	Manager
	Liam	Johnson	Developer	Developer
	Olivia	Martinez	HR Specialist	Other
	Lucas	Garcia	Marketing A...	Other
	Mia	Davis	Sales Rep	Other
	James	Brown	Operations ...	Manager
	Isabella	Rodriguez	Customer S...	Other

## 11. retrieve a list of employee who worked on multiple project

```
SELECT  
    e.first_name,  
    e.last_name,  
    COUNT(DISTINCT t.project_id) AS project_count  
FROM Employees e  
JOIN Timesheets t ON e.id = t.employee_id  
GROUP BY e.id, e.first_name, e.last_name  
HAVING COUNT(DISTINCT t.project_id) > 1;
```

	first_name	last_name	project_count
▶	John	Doe	2

## 12. Retrieve each emp total hr worked on project and show the rank of each emp based on total hour worked

```
SELECT  
    e.first_name,  
    e.last_name,  
    SUM(t.hours) AS total_hours,  
    RANK() OVER (ORDER BY SUM(t.hours) DESC) AS rank_hours  
FROM Employees e  
JOIN Timesheets t ON e.id = t.employee_id  
GROUP BY e.id, e.first_name, e.last_name  
ORDER BY rank_hours;
```

	first_name	last_name	total_hours	rank_hours
▶	John	Doe	23.50	1
	Jane	Smith	20.50	2
	Liam	Johnson	15.50	3
	David	Miller	14.50	4
	James	Brown	14.50	4
	Lucy	Davis	13.00	6
	Isabella	Rodriguez	12.50	7
	Lucas	Garcia	11.50	8
	Tom	Brown	8.00	9
	Sophia	Wilson	8.00	9
	Olivia	Martinez	8.00	9
	Emma	Watson	7.50	12
	Mia	Davis	7.00	13
	Mark	Taylor	6.00	14

# 13.list all the employee whose Total hour worked are above the overall avg using subquery.

The screenshot shows a MySQL Workbench interface with a query editor window. The query is:

```
134      #13 list all the emp whose total hours on multiple projects
135 •  SELECT
136      e.first_name,
137      e.last_name,
138      SUM(t.hours) AS total_hours
139  FROM Employees e
140  JOIN Timesheets t
141      ON e.id = t.employee_id
142  GROUP BY e.id
143  HAVING SUM(t.hours) > (
144      SELECT AVG(emp_hours)
145      FROM (
146          SELECT SUM(hours) AS emp_hours
147          FROM Timesheets
```

The result grid below the query shows the following data:

	first_name	last_name	total_hours
▶	John	Doe	23.50
	Jane	Smith	20.50
	Lucy	Davis	13.00
	David	Miller	14.50
	Liam	Johnson	15.50
	James	Brown	14.50
	Isabella	Rodriguez	12.50

14.update the employee table by changing the job title of all employee working in the ‘Engineering’ department to ‘Senior Developer’ except for those who are ‘Manager’ or ‘HR Specialist’.

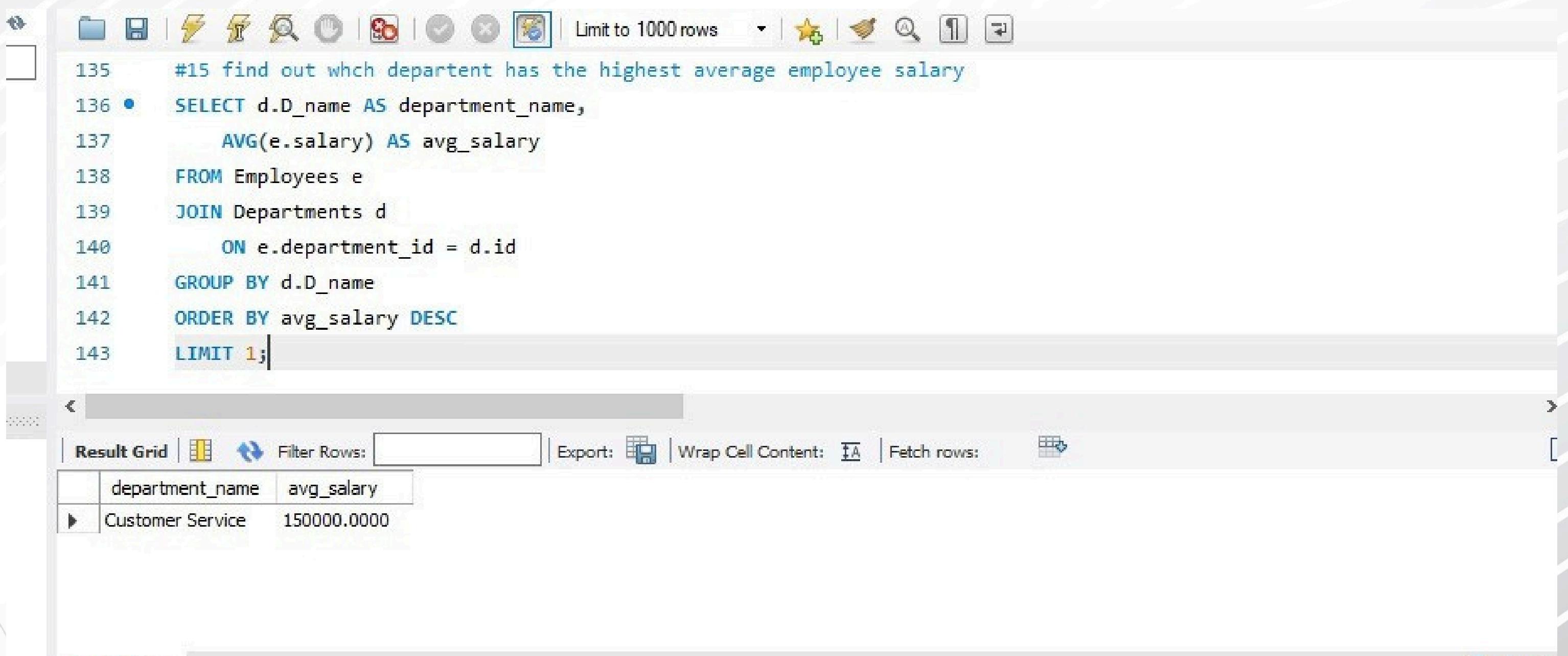
SQL File 3\*

```
160      # 14 Update the Employees table by changing the job _title of all employees working in the 'Engineering' department
161 • UPDATE Employees
162     SET job_title = 'Senior Developer'
163     WHERE department_id = 3
164     AND job_title NOT IN ('Manager', 'HR Specialist');
165 • select * from Employees;
166
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id	first_name	last_name	email	phone_number	hire_date	job_title	department_id	salary
▶	1	John	Doe	john.doe@example.com	555-1234	2022-01-15	Manager	1	40000
	2	Jane	Smith	jane.smith@example.com	555-2345	2021-06-22	Senior Developer	3	55000
	3	Tom	Brown	tom.brown@example.com	555-3456	2020-09-12	Manager	2	30000
	4	Lucy	Davis	lucy.davis@example.com	555-4567	2019-11-02	Developer	4	60000
	5	Mark	Taylor	mark.taylor@example.com	555-5678	2018-08-30	Sales Rep	5	75000
	6	Emma	Watson	emma.watson@example.com	555-8765	2020-05-14	Manager	6	100000
	7	David	Miller	david.miller@example.com	555-7654	2019-04-20	Developer	6	100000
	8	Sophia	Wilson	sophia.wilson@example.com	555-6543	2018-12-01	Manager	7	125000
	9	Liam	Johnson	liam.johnson@example.com	555-5432	2021-07-17	Developer	7	125000
	10	Olivia	Martinez	olivia.martinez@example.com	555-4321	2022-09-09	HR Specialist	1	40000
	11	Lucas	Garcia	lucas.garcia@example.com	555-3210	2023-02-22	Marketing Analyst	4	60000
	12	Mia	Davis	mia.davis@example.com	555-2109	2023-03-15	Sales Rep	5	75000
	13	James	Brown	james.brown@example.com	555-1098	2021-10-11	Operations Man...	8	130000
	14	Isabella	Rodriguez	isabella.rodriguez@example....	555-0987	2017-08-06	Customer Servic...	9	150000

## 15.find out which department has the highest avg employee salary.



The screenshot shows a MySQL Workbench interface. The top section contains a toolbar with various icons for file operations, search, and connection management. Below the toolbar, a query editor window displays the following SQL code:

```
135 #15 find out whch departent has the highest average employee salary
136 • SELECT d.D_name AS department_name,
137     AVG(e.salary) AS avg_salary
138 FROM Employees e
139 JOIN Departments d
140     ON e.department_id = d.id
141 GROUP BY d.D_name
142 ORDER BY avg_salary DESC
143 LIMIT 1;
```

The code uses aliasing for the department name and salary, performs a join between the Employees and Departments tables, groups by department name, orders by average salary in descending order, and limits the result to one row. The result grid below the query editor shows the output:

department_name	avg_salary
Customer Service	150000.0000

# CONCLUSION

The Employee Database Management System was successfully created using SQL by designing tables, inserting data, and generating meaningful reports.

The project helped understand how relational databases store and manage employee, department, project, and work-hour information efficiently.

Overall, the project strengthened practical knowledge of database design and SQL operations.

# THANK YOU

-Dikshya Panda