SQL Analysis Report – Sample Queries & Insights

1. Total Tasks Completed by Each Employee

Purpose: To evaluate the total number of tasks completed by each employee.

```
mysql> SELECT
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     SUM(t.tasks_completed) AS total_tasks_completed
    -> FROM employees e
    -> JOIN tasks t ON e.employee_id = t.employee_id
    -> GROUP BY e.employee_id, e.name
    -> ORDER BY total_tasks_completed DESC;
+-------------+----------------+-----------------------+
| employee_id | employee_name  | total_tasks_completed |
+-------------+----------------+-----------------------+
| E007        | David Lee      |                    17 |
| E003        | Robert Brown   |                    15 |
| E005        | Michael Wilson |                    13 |
| E001        | John Doe       |                    11 |
| E004        | Emily Davis    |                    11 |
| E010        | Olivia Lewis   |                    10 |
| E006        | Sarah Johnson  |                     9 |
| E002        | Jane Smith     |                     7 |
| E008        | Laura Clark    |                     7 |
| E009        | James Hall     |                     5 |
+-------------+----------------+-----------------------+
10 rows in set (0.01 sec)
```

2. Total Hours Worked by Each Employee

Purpose: To calculate the total hours worked by each employee across all recorded days.

```
mysql> SELECT
```

```
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     SUM(a.hours_worked) AS total_hours_worked
    -> FROM employees e
    -> JOIN attendance a ON e.employee_id = a.employee_id
    -> GROUP BY e.employee_id, e.name
    -> ORDER BY total_hours_worked DESC;
+-------------+----------------+--------------------+
| employee_id | employee_name  | total_hours_worked |
+-------------+----------------+--------------------+
| E007        | David Lee      |              17.50 |
| E003        | Robert Brown   |              17.00 |
| E005        | Michael Wilson |              16.00 |
| E010        | Olivia Lewis   |              16.00 |
| E001        | John Doe       |              15.50 |
| E002        | Jane Smith     |              15.50 |
| E006        | Sarah Johnson  |              15.50 |
| E008        | Laura Clark    |              14.00 |
| E004        | Emily Davis    |              13.50 |
| E009        | James Hall     |              12.50 |
+-------------+----------------+--------------------+
10 rows in set (0.00 sec)
```

3. Average Tasks Completed Per Day for Each Employee

Purpose: To understand the average number of tasks each employee completes per day.

```
mysql> SELECT
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     AVG(t.tasks_completed) AS avg_tasks_per_day
    -> FROM employees e
    -> JOIN tasks t ON e.employee_id = t.employee_id
    -> GROUP BY e.employee_id, e.name
    -> ORDER BY avg_tasks_per_day DESC;
+-------------+----------------+-------------------+
| employee_id | employee_name  | avg_tasks_per_day |
+-------------+----------------+-------------------+
| E007        | David Lee      |            8.5000 |
| E003        | Robert Brown   |            7.5000 |
| E005        | Michael Wilson |            6.5000 |
| E001        | John Doe       |            5.5000 |
| E004        | Emily Davis    |            5.5000 |
| E010        | Olivia Lewis   |            5.0000 |
| E006        | Sarah Johnson  |            4.5000 |
| E002        | Jane Smith     |            3.5000 |
| E008        | Laura Clark    |            3.5000 |
| E009        | James Hall     |            2.5000 |
+-------------+----------------+-------------------+
10 rows in set (0.00 sec)
```

4. Employee Attendance Summary

Purpose: To get a summary of attendance, showing the total number of days each employee attended.

```
mysql> SELECT
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     COUNT(a.attendance_id) AS days_present
    -> FROM employees e
    -> JOIN attendance a ON e.employee_id = a.employee_id
    -> WHERE a.hours_worked > 0
    -> GROUP BY e.employee_id, e.name
    -> ORDER BY days_present DESC;
+-------------+----------------+--------------+
| employee_id | employee_name  | days_present |
+-------------+----------------+--------------+
| E001        | John Doe       |            2 |
| E002        | Jane Smith     |            2 |
| E003        | Robert Brown   |            2 |
| E004        | Emily Davis    |            2 |
| E005        | Michael Wilson |            2 |
| E006        | Sarah Johnson  |            2 |
| E007        | David Lee      |            2 |
| E008        | Laura Clark    |            2 |
| E009        | James Hall     |            2 |
| E010        | Olivia Lewis   |            2 |
+-------------+----------------+--------------+
10 rows in set (0.00 sec)
```

5. Employee Productivity (Tasks Completed vs Hours Worked)

Purpose: To identify the productivity of each employee by comparing the number of tasks completed with hours worked.

```
mysql> SELECT
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     SUM(t.tasks_completed) AS total_tasks_completed,
    ->     SUM(a.hours_worked) AS total_hours_worked,
    ->     ROUND(SUM(t.tasks_completed) / SUM(a.hours_worked), 2) AS tasks_per_hour
    -> FROM employees e
    -> JOIN tasks t ON e.employee_id = t.employee_id
    -> JOIN attendance a ON e.employee_id = a.employee_id
    -> GROUP BY e.employee_id, e.name
    -> ORDER BY tasks_per_hour DESC;
+-------------+----------------+-----------------------+--------------------+----------------+
| employee_id | employee_name  | total_tasks_completed | total_hours_worked | tasks_per_hour |
+-------------+----------------+-----------------------+--------------------+----------------+
| E007        | David Lee      |                    34 |              35.00 |           0.97 |
| E003        | Robert Brown   |                    30 |              34.00 |           0.88 |
| E004        | Emily Davis    |                    22 |              27.00 |           0.81 |
| E005        | Michael Wilson |                    26 |              32.00 |           0.81 |
| E001        | John Doe       |                    22 |              31.00 |           0.71 |
| E010        | Olivia Lewis   |                    20 |              32.00 |           0.63 |
| E006        | Sarah Johnson  |                    18 |              31.00 |           0.58 |
| E008        | Laura Clark    |                    14 |              28.00 |           0.50 |
| E002        | Jane Smith     |                    14 |              31.00 |           0.45 |
| E009        | James Hall     |                    10 |              25.00 |           0.40 |
+-------------+----------------+-----------------------+--------------------+----------------+
10 rows in set (0.01 sec)
```

6. Employee Workload Analysis (Tasks Assigned vs Completed)

Purpose: To understand how many tasks were assigned versus completed for each employee.

```
mysql> SELECT
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     COUNT(t.task_id) AS total_assigned_tasks,
    ->     SUM(CASE WHEN t.tasks_completed > 0 THEN 1 ELSE 0 END) AS completed_tasks
    -> FROM employees e
    -> LEFT JOIN tasks t ON e.employee_id = t.employee_id
    -> GROUP BY e.employee_id, e.name
    -> ORDER BY completed_tasks DESC;
+-------------+----------------+----------------------+-----------------+
| employee_id | employee_name  | total_assigned_tasks | completed_tasks |
+-------------+----------------+----------------------+-----------------+
| E001        | John Doe       |                    2 |               2 |
| E002        | Jane Smith     |                    2 |               2 |
| E003        | Robert Brown   |                    2 |               2 |
| E004        | Emily Davis    |                    2 |               2 |
| E005        | Michael Wilson |                    2 |               2 |
| E006        | Sarah Johnson  |                    2 |               2 |
| E007        | David Lee      |                    2 |               2 |
| E008        | Laura Clark    |                    2 |               2 |
| E009        | James Hall     |                    2 |               2 |
| E010        | Olivia Lewis   |                    2 |               2 |
+-------------+----------------+----------------------+-----------------+
10 rows in set (0.00 sec)
```

7. Employee Task Completion on a Specific Date

Purpose: To see how many tasks were completed by each employee on a specific date.

```
mysql> SELECT
    ->     e.employee_id,
    ->     e.name AS employee_name,
    ->     t.date,
    ->     SUM(t.tasks_completed) AS total_tasks_completed
    -> FROM employees e
    -> JOIN tasks t ON e.employee_id = t.employee_id
    -> WHERE t.date = '2025-05-01'  -- Change to any specific date
    -> GROUP BY e.employee_id, e.name, t.date
    -> ORDER BY total_tasks_completed DESC;
+-------------+----------------+------------+-----------------------+
| employee_id | employee_name  | date       | total_tasks_completed |
+-------------+----------------+------------+-----------------------+
| E007        | David Lee      | 2025-05-01 |                     9 |
| E003        | Robert Brown   | 2025-05-01 |                     8 |
| E005        | Michael Wilson | 2025-05-01 |                     7 |
| E004        | Emily Davis    | 2025-05-01 |                     6 |
| E001        | John Doe       | 2025-05-01 |                     5 |
| E010        | Olivia Lewis   | 2025-05-01 |                     5 |
| E002        | Jane Smith     | 2025-05-01 |                     4 |
| E006        | Sarah Johnson  | 2025-05-01 |                     4 |
| E008        | Laura Clark    | 2025-05-01 |                     3 |
| E009        | James Hall     | 2025-05-01 |                     2 |
+-------------+----------------+------------+-----------------------+
10 rows in set (0.01 sec)
```

8. Department-wise Task Completion Summary

Purpose: To get an overview of how many tasks each department has completed.

```
mysql> SELECT
    ->     e.department,
    ->     SUM(t.tasks_completed) AS total_tasks_completed
    -> FROM employees e
    -> JOIN tasks t ON e.employee_id = t.employee_id
    -> GROUP BY e.department
    -> ORDER BY total_tasks_completed DESC;
+------------+-----------------------+
| department | total_tasks_completed |
+------------+-----------------------+
| IT         |                    32 |
| Marketing  |                    21 |
| Sales      |                    20 |
| Finance    |                    18 |
| HR         |                    14 |
+------------+-----------------------+
5 rows in set (0.00 sec)
```

Summary of Employee Performance & Productivity Analysis Using SQL

This SQL analysis report provides a comprehensive overview of employee performance and departmental productivity based on data from the employees, attendance, and tasks tables. The queries calculate total and average metrics such as tasks completed, hours worked, and tasks per hour to evaluate individual and team efficiency. Attendance patterns are analyzed to determine presence across workdays, while department-level aggregation highlights the performance of different business units. The report also includes a daily breakdown of task completion and a comparison of workload and output per employee. These insights are valuable for performance reviews, resource planning, and strategic decision-making in workforce management.