# Employee Performance Dashboard

# 1. Top 5 Employees by Total Sales

**MySQL Query**

```sql
-- 1. Top 5 Employees by Total Sales

SELECT
    ep.EmployeeID,
    ep.EmployeeName,
    SUM(ep.SalesAmount) AS total_Sales
FROM
    employee_performance_dashboard.employeeperformance ep
GROUP BY ep.EmployeeID , ep.EmployeeName
ORDER BY total_Sales DESC
LIMIT 5;

```

**Its Result**

| EmployeeID | EmployeeName | total_Sales |
|---|---|---|
| KIRKAU957 | Kirandeep Kaur | 9998.15 |
| RANSIN758 | Ranjitpal Singh | 9988.08 |
| AMRKAU160 | Amrik Kaur | 9950.84 |
| SHANAI784 | Sharad Nair | 9912.07 |
| MILMIS601 | Milad Mistry | 9905.82 |

Result 1

# 2. Employees with the Most Absence in the last 3 Months

## MySQL Query

```sql
-- 2. Employees with the Most Absence in the last 3 Months

SELECT
    ep.EmployeeID, ep.EmployeeName,
    (SUM(ar.TotalWorkingDays) - SUM(ar.PresentDays)) AS total_absent_days
FROM
    employee_performance_dashboard.employeeperformance ep
    JOIN employee_performance_dashboard.attendancerecords ar ON ep.EmployeeID = ar.EmployeeID
WHERE
    STR_TO_DATE(CONCAT(ar.Att_Year, '-',
                    CASE ar.Att_Month
                        WHEN 'January' THEN '01'
                        WHEN 'February' THEN '02'
                        WHEN 'March' THEN '03'
                        WHEN 'April' THEN '04'
                        WHEN 'May' THEN '05'
                        WHEN 'June' THEN '06'
                        WHEN 'July' THEN '07'
                        WHEN 'August' THEN '08'
                        WHEN 'September' THEN '09'
                        WHEN 'October' THEN '10'
                        WHEN 'November' THEN '11'
                        WHEN 'December' THEN '12' END, '-01'), '%Y-%m-%d') >= CURDATE() - INTERVAL 3 MONTH
GROUP BY ep.EmployeeID , ep.EmployeeName ORDER BY total_absent_days DESC
LIMIT 1;
```

## Its Result

| EmployeeID | EmployeeName | total_absent_days |
|---|---|---|
| AMASIN360 | Amarjeet Singh | 11 |

Result 1

# 3. Department Performance: Average Performance Rating for Each Department

**MySQL Query**

```sql
-- 3. Department Performance: Average Performance Rating for Each Department

SELECT
    ep.Department,
    ROUND(AVG(pr.PerformanceRating), 2) AS AveragePerformanceRating
FROM
    employee_performance_dashboard.employeeperformance ep
        JOIN
    employee_performance_dashboard.performanceratingstatus pr
    ON pr.EmployeeID = ep.EmployeeID
GROUP BY ep.Department
ORDER BY AveragePerformanceRating DESC;

```

**Its Result**

| Department | AveragePerformanceRating |
|---|---|
| IT | 5.00 |
| HR | 4.00 |
| Product Management | 3.82 |
| Public Relations | 3.80 |
| Administration | 3.80 |

Result 1

# 4. Top 3 Managers by Total Sales of Their Teams

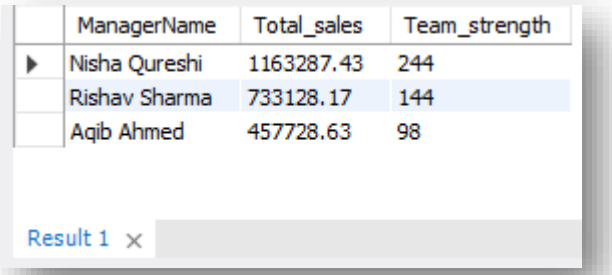
**MySQL Query**

```sql
-- 4. Top 3 Managers by Total Sales of Their Teams

SELECT
    m.ManagerName,
    SUM(ep.SalesAmount) AS Total_sales,
    COUNT(ep.EmployeeID) AS Team_strength
FROM
    employee_performance_dashboard.managers m
        JOIN
    employee_performance_dashboard.employeeperformance ep
    ON m.ManagerID = ep.ManagerID
GROUP BY m.ManagerName
ORDER BY Total_sales DESC
LIMIT 3;

```

**Its Result**



| ManagerName | Total_sales | Team_strength |
|---|---|---|
| Nisha Qureshi | 1163287.43 | 244 |
| Rishav Sharma | 733128.17 | 144 |
| Aqib Ahmed | 457728.63 | 98 |

Result 1

# 5. Employees Without a Bonus

**MySQL Query**

```sql
-- 5. Employees Without a Bonus

SELECT
    ep.EmployeeID, ep.EmployeeName, ep.Department
FROM
    employee_performance_dashboard.employeeperformance ep
WHERE
    ep.BonusEarned = 0;

```

**Its Result**

| EmployeeID | EmployeeName | Department |
|---|---|---|
| ABDRAH407 | Abdul Rahman | Public Relations |
| ANSRED667 | Anshika Reddy | Marketing |
| BAKSIN398 | Bakhshish Singh | Customer Support |
| CHATAY832 | Charles Taylor | Logistics |
| CYRDAS713 | Cyrus Dastoori | Procurement |

performance 1

# 6. Employees with Sales Above the Department Average

**MySQL Query**

```sql
-- 6. Employees with Sales Above the Department Average

SELECT
    ep.EmployeeName, ep.Department, ep.SalesAmount
FROM
    employee_performance_dashboard.employeeperformance ep
WHERE
    ep.SalesAmount > (SELECT
            AVG(ep2.SalesAmount)
        FROM
            employee_performance_dashboard.employeeperformance ep2
        WHERE
            ep2.Department = ep.Department
        GROUP BY ep2.Department)
order by ep.SalesAmount desc;

```

**Its Result**

| EmployeeName | Department | SalesAmount |
|---|---|---|
| Kirandeep Kaur | HR | 9998.15 |
| Ranjitpal Singh | Research & Development | 9988.08 |
| Amrik Kaur | Customer Support | 9950.84 |
| Sharad Nair | Public Relations | 9912.07 |
| Milad Mistry | Training & Development | 9905.82 |

performance 1

# 7. Managers with Employees Who Received a Rating of 5

**MySQL Query**

```sql
-- 7. Managers with Employees Who Received a Rating of 5

SELECT
    ManagersWithRating.ManagerName,
    COUNT(ManagersWithRating.PerformanceRating) AS Emp_Count
FROM
    (SELECT
        m.ManagerName, ps.PerformanceRating
    FROM
        employee_performance_dashboard.managers m
    JOIN employee_performance_dashboard.employeeperformance ep ON m.ManagerID = ep.ManagerID
    JOIN employee_performance_dashboard.performanceratingstatus ps ON ep.EmployeeID = ps.EmployeeID
    WHERE
        ps.PerformanceRating = 5) AS ManagersWithRating
GROUP BY ManagersWithRating.ManagerName
ORDER BY Emp_Count DESC

```

**Its Result**

| ManagerName | Emp_Count |
|---|---|
| Nisha Qureshi | 66 |
| Rajinderjeet Kaur | 32 |
| Rishav Sharma | 28 |
| Aqib Ahmed | 25 |
| Ajit Singh | 20 |

Result 1

# 8. Top 5 Departments by Total Sales

**MySQL Query**

```sql
-- 8 Top 5 Departments by Total Sales

SELECT
    ep.Department,
    COUNT(ep.EmployeeName) AS total_employee,
    SUM(ep.SalesAmount) AS total_Sales
FROM
    employee_performance_dashboard.employeeperformance ep
GROUP BY ep.Department
ORDER BY total_Sales DESC
LIMIT 5;

```

**Its Result**

| Department | total_employee | total_Sales |
|---|---|---|
| Sales | 244 | 1163287.43 |
| Customer Support | 144 | 733128.17 |
| Logistics | 98 | 457728.63 |
| Public Relations | 89 | 401588.94 |
| Strategy & Planning | 53 | 288794.09 |

Result 1

# 9. Employees with More Than 10 Absences in the last 6 Months

## MySQL Query

```sql
-- 9. Employees with More Than 10 Absences in the last 6 Months

SELECT
    ep.EmployeeID, ep.EmployeeName, SUM(ar.AbsentDays) AS total_absent
FROM
    employee_performance_dashboard.employeeperformance ep JOIN
    employee_performance_dashboard.attendancerecords ar ON ep.EmployeeID = ar.EmployeeID
WHERE
    STR_TO_DATE(CONCAT(ar.Att_Year,'-',CASE ar.Att_Month
                        WHEN 'January' THEN '01'
                        WHEN 'February' THEN '02'
                        WHEN 'March' THEN '03'
                        WHEN 'April' THEN '04'
                        WHEN 'May' THEN '05'
                        WHEN 'June' THEN '06'
                        WHEN 'July' THEN '07'
                        WHEN 'August' THEN '08'
                        WHEN 'September' THEN '09'
                        WHEN 'October' THEN '10'
                        WHEN 'November' THEN '11'
                        WHEN 'December' THEN '12' END, '-01'),
            '%Y-%m-%d') >= CURDATE() - INTERVAL 5 MONTH
GROUP BY ep.EmployeeID , ep.EmployeeName
HAVING total_absent > 10
ORDER BY total_absent DESC;
```

## Its Result

| EmployeeID | EmployeeName | total_absent |
|---|---|---|
| NAVKAU810 | Navjotpal Kaur | 16 |
| PARKAU495 | Pardeep Kaur | 15 |
| DARGHA602 | Dara Ghadiali | 15 |
| ELLMOR342 | Ella Morgan | 15 |
| ARMPES755 | Arman Pestonji | 14 |

Result 1

# 10. Employees Who Have Received a Bonus Above $2000

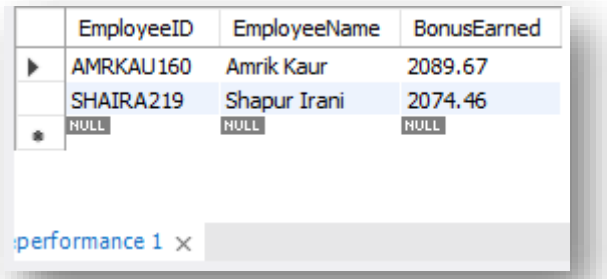
**MySQL Query**

```sql
-- 10. Employees Who Have Received a Bonus Above $2000

SELECT
    ep.EmployeeID, ep.EmployeeName, ep.BonusEarned
FROM
    employee_performance_dashboard.employeeperformance ep
WHERE
    ep.BonusEarned > 2000

```

**Its Result**



| | EmployeeID | EmployeeName | BonusEarned |
|---|---|---|---|
| ▶ | AMRKAU160 | Amrik Kaur | 2089.67 |
| | SHAIRA219 | Shapur Irani | 2074.46 |
| * | NULL | NULL | NULL |

performance 1 ×

# 11. Employees with Attendance Below 85% in the Last 6 Months

**MySQL Query**

```sql
-- 11. Employees with Attendance Below 85% in the Last 6 Months

SELECT
    ep.EmployeeID, ep.EmployeeName, round((SUM(ar.PresentDays)/sum(ar.TotalWorkingDays)*100),2) AS Attendance_Percentage
FROM
    employee_performance_dashboard.employeeperformance ep JOIN
    employee_performance_dashboard.attendancerecords ar ON ep.EmployeeID = ar.EmployeeID
WHERE STR_TO_DATE(CONCAT(ar.Att_Year, '-', CASE ar.Att_Month
                        WHEN 'January' THEN '01'
                        WHEN 'February' THEN '02'
                        WHEN 'March' THEN '03'
                        WHEN 'April' THEN '04'
                        WHEN 'May' THEN '05'
                        WHEN 'June' THEN '06'
                        WHEN 'July' THEN '07'
                        WHEN 'August' THEN '08'
                        WHEN 'September' THEN '09'
                        WHEN 'October' THEN '10'
                        WHEN 'November' THEN '11'
                        WHEN 'December' THEN '12' END,'-01'),
            '%Y-%m-%d') >= CURDATE() - INTERVAL 6 MONTH
GROUP BY ep.EmployeeID , ep.EmployeeName
having Attendance_Percentage < 85
order by Attendance_Percentage;
```

**Its Result**

| EmployeeID | EmployeeName | Attendance_Percentage |
|---|---|---|
| SHAAHM153 | Shaheer Ahmed | 82.35 |
| ARMPES755 | Arman Pestonji | 83.01 |
| DARBIL041 | Darius Billimoria | 83.01 |
| BALSIN612 | Baljeet Singh | 84.31 |
| KIAMIS677 | Kiana Mistry | 84.31 |

Result 1

# 12. Top 3 Performing Employees by Sales Amount and Performance Rating

**MySQL Query**

```sql
-- 12. Top 3 Performing Employees by Sales Amount and Performance Rating

SELECT
    ep.EmployeeID,
    ep.EmployeeName,
    pr.PerformanceRating,
    SUM(ep.SalesAmount) AS total_Sales
FROM
    employee_performance_dashboard.employeeperformance ep
        JOIN
    employee_performance_dashboard.performanceratingstatus pr ON ep.EmployeeID = pr.EmployeeID
GROUP BY ep.EmployeeID , ep.EmployeeName , pr.PerformanceRating
ORDER BY total_Sales desc, pr.PerformanceRating desc
LIMIT 3;

```

**Its Result**

| EmployeeID | EmployeeName | PerformanceRating | total_Sales |
|---|---|---|---|
| KIRKAU957 | Kirandeep Kaur | 3 | 9998.15 |
| RANSIN758 | Ranjitpal Singh | 3 | 9988.08 |
| AMRKAU160 | Amrik Kaur | 5 | 9950.84 |

Result 1

# 13. Department Attendance Percentage by Month

**MySQL Query**

```sql
-- 13. Department Attendance Percentage by Month

SELECT
    e.Department,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'January' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS January,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'February' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS February,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'March' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS March,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'April' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS April,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'May' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS May,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'June' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS June,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'July' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS July,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'August' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS August,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'September' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS September,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'October' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS October,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'November' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS November,
    COALESCE(Round(AVG(CASE WHEN a.Att_Month = 'December' THEN (a.PresentDays / a.TotalWorkingDays) * 100 END),2), 0) AS December
FROM employee_performance_dashboard.attendancerecords a
JOIN employee_performance_dashboard.employeeperformance e
ON a.EmployeeID = e.EmployeeID
GROUP BY
    e.Department;
```

**Its Result**

| Department | January | February | March | April | May | June | July | August | September | October | November | December |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sales | 90.20 | 92.69 | 89.99 | 92.04 | 90.68 | 91.41 | 90.04 | 91.20 | 91.75 | 90.79 | 91.65 | 90.35 |
| Logistics | 89.83 | 93.53 | 90.29 | 90.95 | 90.29 | 91.29 | 91.34 | 90.39 | 92.86 | 89.86 | 92.13 | 90.06 |
| Marketing | 91.07 | 92.71 | 89.33 | 91.73 | 92.25 | 92.31 | 89.39 | 91.69 | 91.41 | 90.94 | 90.77 | 89.92 |
| Research & Development | 91.65 | 91.99 | 89.75 | 91.76 | 89.85 | 92.35 | 90.99 | 91.08 | 91.86 | 89.80 | 91.42 | 89.90 |
| Customer Support | 89.87 | 93.30 | 90.52 | 91.99 | 90.32 | 91.48 | 90.21 | 89.74 | 91.76 | 90.09 | 91.92 | 90.40 |

Result 8

# 14. Low-Performing Employees: Attendance and Sales Analysis

**MySQL Query**

```sql
-- 14. Low-Performing Employees: Attendance and Sales Analysis

SELECT
    ep.EmployeeID,
    ep.EmployeeName,
    pr.PerformanceRating,
    SUM(ep.SalesAmount) AS total_sales,
    round((SUM(ar.PresentDays)/SUM(ar.TotalWorkingDays)*100),2) AS Attendance_Percentage
FROM
    employee_performance_dashboard.employeeperformance ep
        JOIN
    employee_performance_dashboard.performanceratingstatus pr ON ep.EmployeeID = pr.EmployeeID
        JOIN
    employee_performance_dashboard.attendancerecords ar ON ep.EmployeeID = ar.EmployeeID
GROUP BY ep.EmployeeID , ep.EmployeeName , pr.PerformanceRating
HAVING pr.PerformanceRating < 3
ORDER BY total_sales asc;

```

**Its Result**

| EmployeeID | EmployeeName | PerformanceRating | total_sales | Attendance_Percentage |
|---|---|---|---|---|
| ABDRAH407 | Abdul Rahman | 2 | 0.00 | 89.52 |
| CYRDAS713 | Cyrus Dastoori | 2 | 0.00 | 93.45 |
| GABBAK036 | Gabriel Baker | 2 | 0.00 | 91.27 |
| HARKAU352 | Harminder Kaur | 2 | 0.00 | 88.43 |
| INDSIN206 | Inderjeet Singh | 2 | 0.00 | 91.05 |
| MANSIN467 | Manjot Singh | 2 | 0.00 | 93.23 |

Result 10

# 15. Performance Rating Distribution Under Each Manager

**MySQL Query**

```sql
-- 15. Performance Rating Distribution Under Each Manager

select m.ManagerName,
    round(sum(case when pr.PerformanceRating = 1 then 1 else 0 end)/count(ep.EmployeeID)*100,2) as Rating1,
    round(sum(case when pr.PerformanceRating = 2 then 1 else 0 end)/count(ep.EmployeeID)*100,2) as Rating2,
    round(sum(case when pr.PerformanceRating = 3 then 1 else 0 end)/count(ep.EmployeeID)*100,2) as Rating3,
    round(sum(case when pr.PerformanceRating = 4 then 1 else 0 end)/count(ep.EmployeeID)*100,2) as Rating4,
    round(sum(case when pr.PerformanceRating = 5 then 1 else 0 end)/count(ep.EmployeeID)*100,2) as Rating5
from employee_performance_dashboard.employeeperformance ep
join employee_performance_dashboard.managers m
on ep.ManagerID = m.ManagerID
join employee_performance_dashboard.performanceratingstatus pr
on ep.EmployeeID = pr.EmployeeID
GROUP by m.ManagerName
```

**Its Result**

| | ManagerName | Rating1 | Rating2 | Rating3 | Rating4 | Rating5 |
|---|---|---|---|---|---|---|
| | Neelu Sharma | 0.00 | 22.64 | 28.30 | 26.42 | 22.64 |
| | Nisha Qureshi | 0.00 | 24.59 | 23.77 | 24.59 | 27.05 |
| | Priyanka Sharma | 0.00 | 0.00 | 0.00 | 0.00 | 100.00 |
| | Rajinderjeet K... | 0.00 | 20.22 | 15.73 | 28.09 | 35.96 |
| | Rauf Lala | 0.00 | 30.77 | 32.69 | 13.46 | 23.08 |

Result 5