



INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

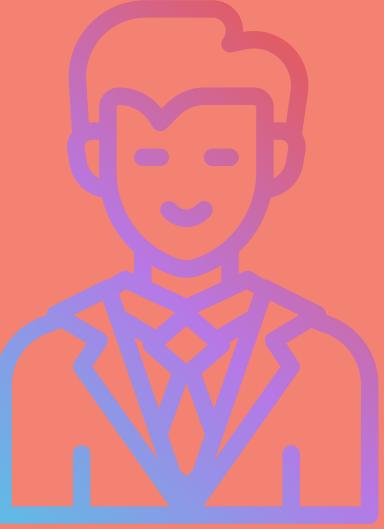
# MySQL Project

MySQL Project on



# EMPLOYEE MANAGEMENT ANALYSIS

Done by: Akshay Kumar  
Harikoti



# INTRODUCTION



**PURPOSE:** The Employee Management System (EMS) is designed to streamline the management of employee data, job roles, and departmental information within an organization.

**TOOLS USED:** MY SQL, POWER POINT, CSV

**DATA SOURCES:** Internal CSV datasets

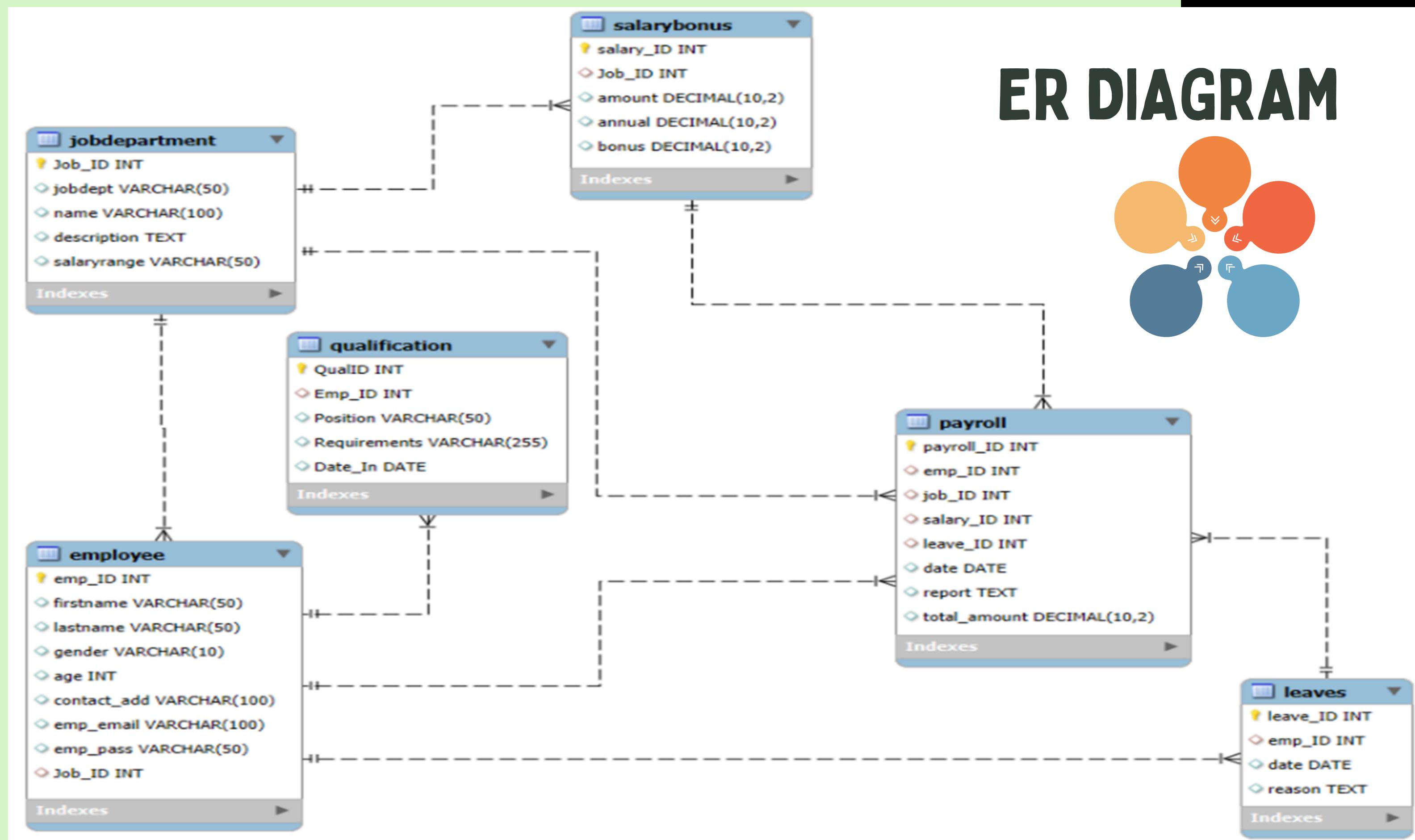


# TABLE DESCRIPTIONS

TABLE	Description
JobDepartment	Stores job roles, departments, and related salary ranges
SalaryBonus	Contains salary, bonus, and annual pay linked to specific job roles.
Employee	Maintains personal, contact, and login details of all employees.
Qualification	Records qualifications and required skills for employee job positions.
Leaves	Tracks employee leave records with reasons and dates.
Payroll	Combines employee, job, salary, and leave data to calculate net payments.



# ER DIAGRAM



# PROBLEM STATEMENT

1. EMPLOYEE INSIGHTS

2. JOB ROLE AND DEPARTMENT ANALYSIS

3. QUALIFICATION AND SKILLS ANALYSIS

4. LEAVE AND ABSENCE PATTERNS

5. PAYROLL AND COMPENSATION ANALYSIS

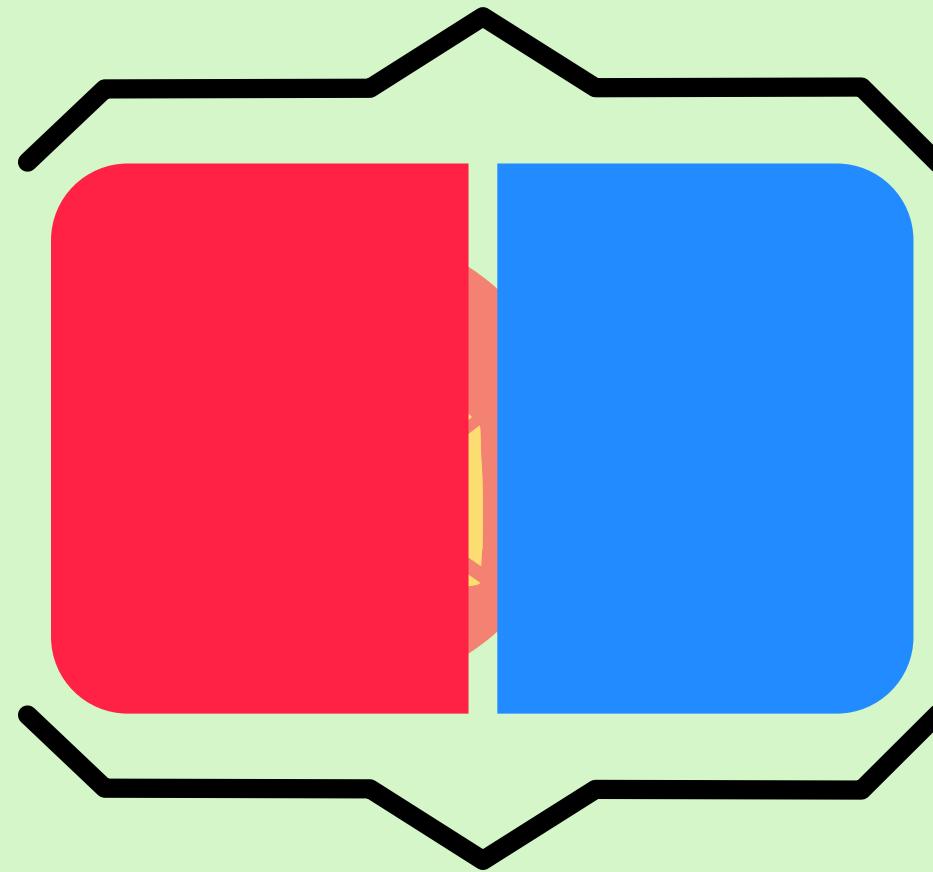
6. EMPLOYEE PERFORMANCE AND GROWTH



# GENERAL AND COMPARATIVE ANALYSIS

How many unique employees are currently in the system?

```
SELECT COUNT(DISTINCT emp_ID) AS unique_employee_count FROM Employee;
```



unique\_employee\_count  
25



# WHICH DEPARTMENTS HAVE THE HIGHEST NUMBER OF EMPLOYEES?



```
SELECT jd.jobdept, COUNT(e.emp_ID) AS employee_count ...
```

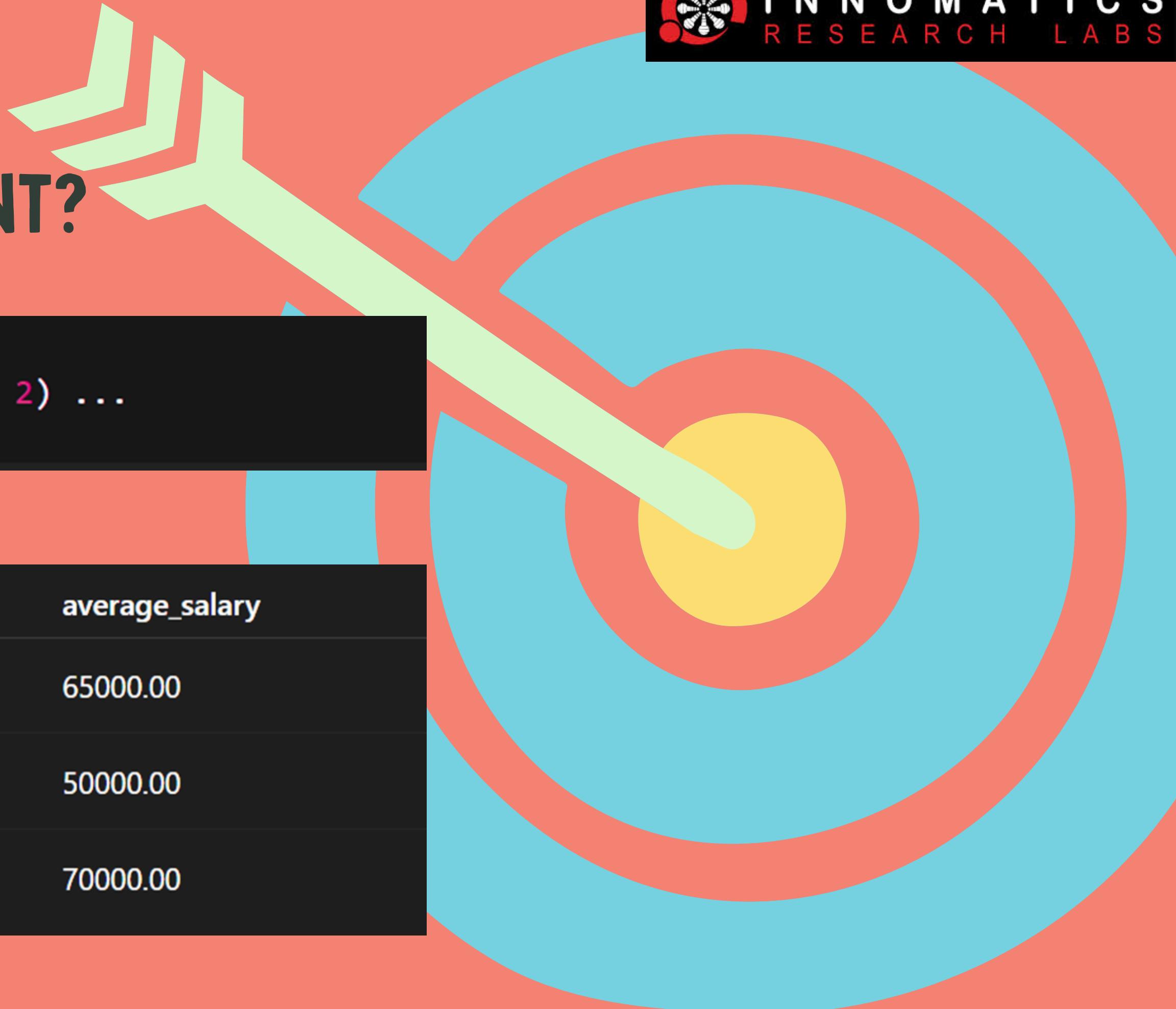
department	employee_count
IT	10
HR	8
Finance	7



# WHAT IS THE AVERAGE SALARY PER DEPARTMENT?

```
SELECT jd.jobdept, ROUND(AVG(sb.amount), 2) ...
```

department	average_salary
IT	65000.00
HR	50000.00
Finance	70000.00



# WHO ARE THE TOP 5 HIGHEST-PAID EMPLOYEES?

```
SELECT e.emp_ID, CONCAT(firstname, ' ', lastname), sb.amount ...
```

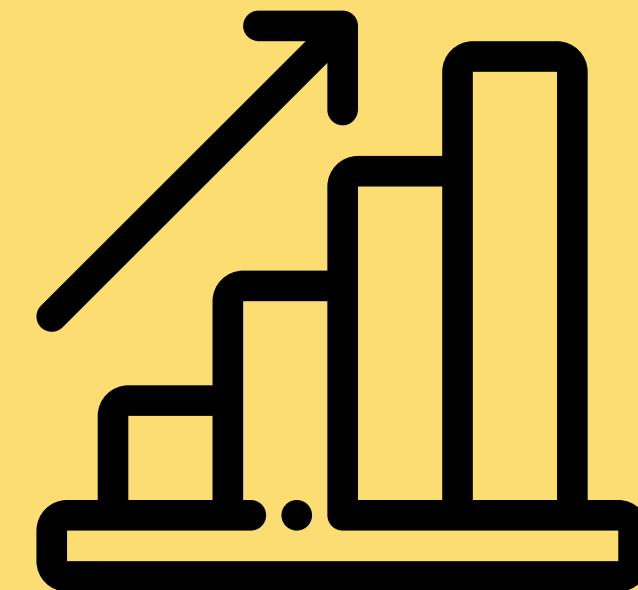


emp_ID	full_name	department	salary
102	John Smith	Finance	90000.00
105	Alice Johnson	IT	85000.00
108	Mark Brown	IT	80000.00



# WHAT IS THE TOTAL SALARY EXPENDITURE ACROSS THE COMPANY?

```
SELECT SUM(sb.amount) ...
```



```
total_salary_expenditure
```

```
1,200,000.00
```

# HOW MANY DIFFERENT JOB ROLES EXIST IN EACH DEPARTMENT?

```
SELECT jobdept, COUNT(DISTINCT name) ...
```

department	job_roles_count
IT	5
HR	3

# WHAT IS THE AVERAGE SALARY RANGE PER DEPARTMENT?

SELECT

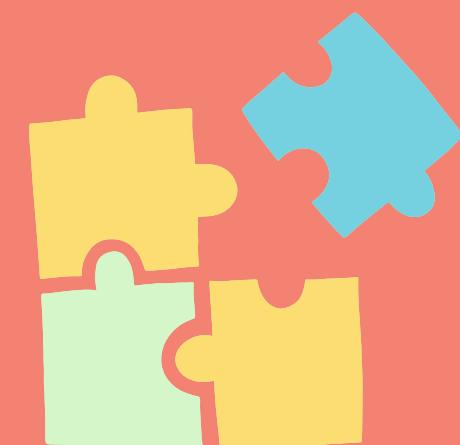
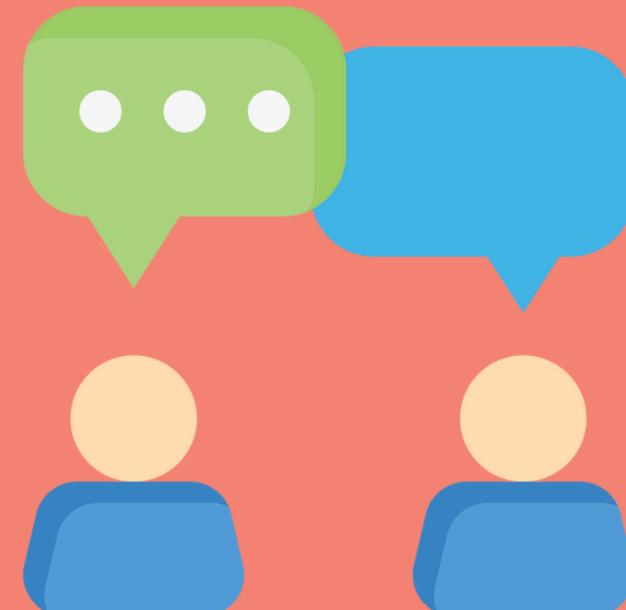
```
jobdept AS Department,  
ROUND(AVG(  
    (CAST(REPLACE(REPLACE(TRIM(SUBSTRING_INDEX(salaryrange, '-', 1)), '$', ''), ',', '') AS DE  
    CAST(REPLACE(REPLACE(TRIM(SUBSTRING_INDEX(salaryrange, '-', -1)), '$', ''), ',', '') AS D  
) / 2  
), 2) AS Average_Salary_Range  
FROM jobdepartment  
WHERE salaryrange LIKE '%-%'  
GROUP BY jobdept;
```



Department	Average_Salary_Range
IT	70000.00
HR	55000.00

# WHICH JOB ROLES OFFER THE HIGHEST SALARY?

```
SELECT
    jd.name AS job_role,
    jd.jobdept AS department,
    MAX(sb.amount) AS highest_salary
FROM
    JobDepartment jd
JOIN
    SalaryBonus sb ON jd.Job_ID = sb.Job_ID
GROUP BY
    jd.name, jd.jobdept
ORDER BY
    highest_salary DESC
LIMIT 5;
```



job_role	department	highest_salary
Data Scientist	IT	95000.00

# WHICH DEPARTMENTS HAVE THE HIGHEST TOTAL SALARY ALLOCATION?

```
SELECT
    jd.jobdept AS department,
    SUM(sb.amount) AS total_salary_allocation
FROM
    Employee e
JOIN
    JobDepartment jd ON e.Job_ID = jd.Job_ID
JOIN
    SalaryBonus sb ON e.Job_ID = sb.Job_ID
GROUP BY
    jd.jobdept
ORDER BY
    total_salary_allocation DESC;
```

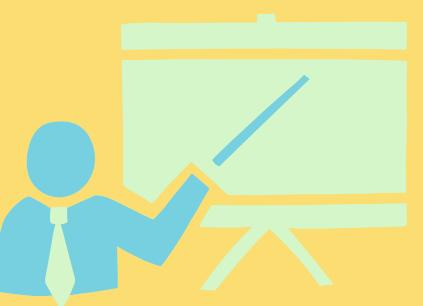


department	total_salary_allocation
IT	600000.00
Finance	400000.00



# How many employees have at least one qualification listed?

```
SELECT  
    COUNT(DISTINCT q.Emp_ID) AS employees_with_qualifications  
FROM  
    Qualification q  
WHERE  
    TRIM(q.Requirements) != ''  
    AND LENGTH(q.Requirements) - LENGTH(REPLACE(q.Requirements, ' ', '')) + 1 > 0;
```



```
| employees_with_qualifications |  
|-----|  
| 15 |
```



# HOW MANY EMPLOYEES HAVE AT LEAST ONE QUALIFICATION LISTED

```
SELECT  
    COUNT(DISTINCT q.Emp_ID) AS employees_with_qualifications  
FROM  
    Qualification q  
WHERE  
    TRIM(q.Requirements) != ''  
    AND LENGTH(q.Requirements) - LENGTH(REPLACE(q.Requirements, ' ', '')) + 1 > 0;
```



employees\_with\_qualifications
15

# WHICH POSITIONS REQUIRE THE MOST QUALIFICATIONS?

```
SELECT  
    Position,  
    LENGTH(Requirements) - LENGTH(REPLACE(Requirements, ' ', '')) + 1 AS qualification_count  
FROM  
    Qualification  
ORDER BY  
    qualification_count DESC  
LIMIT 5;
```

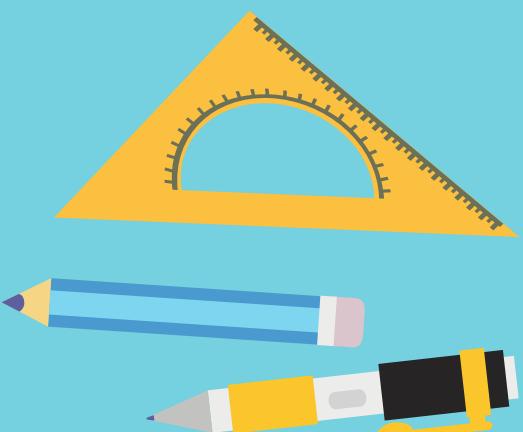


Position	qualification_count
Manager	6



# WHICH EMPLOYEES HAVE THE HIGHEST NUMBER OF QUALIFICATIONS?

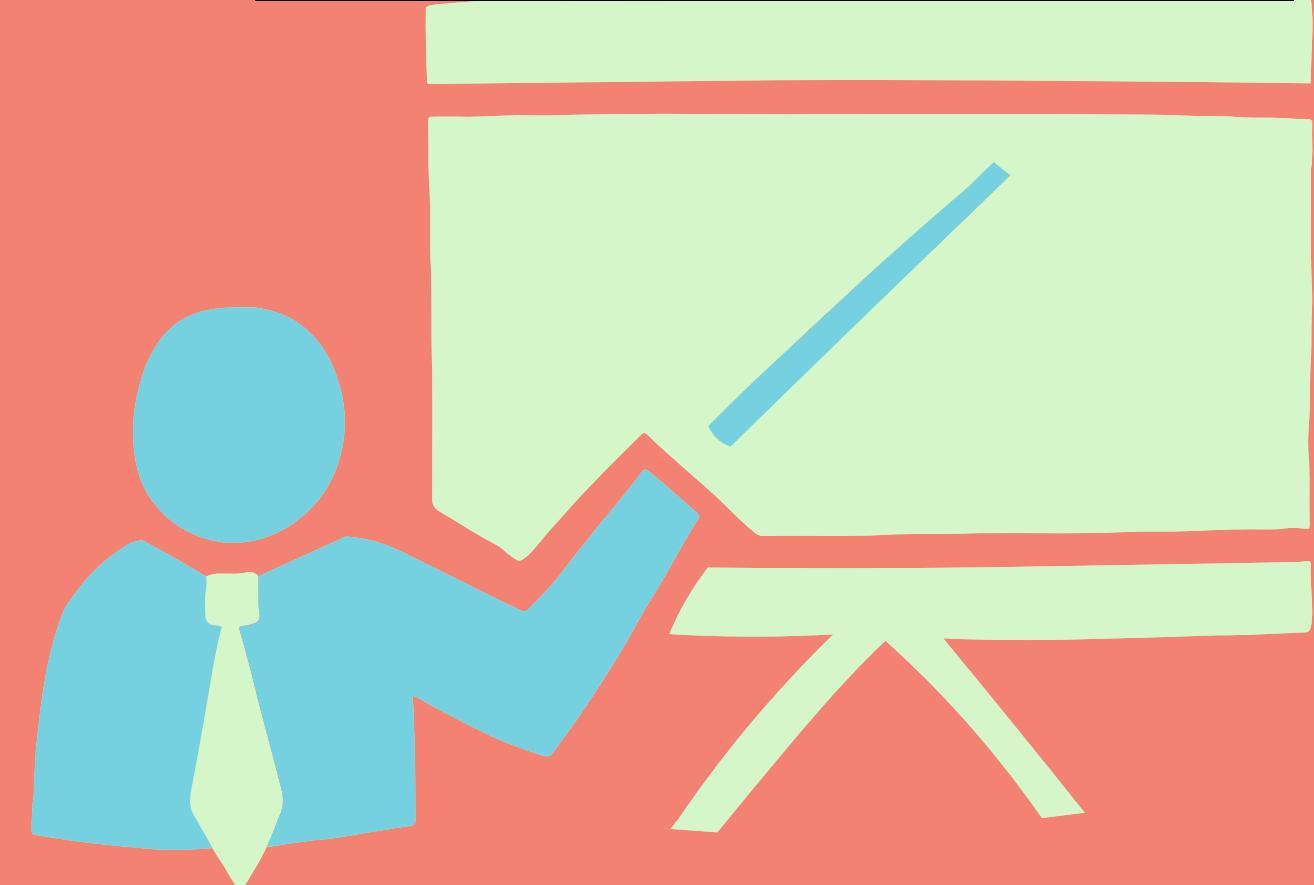
```
SELECT jd.jobdept AS department,
       ROUND(AVG(word_count), 2) AS avg_qualifications_per_employee
  FROM (
    SELECT e.emp_ID,
           e.Job_ID,
           COALESCE(SUM(LENGTH(q.Requirements) - LENGTH(REPLACE(q.Requirements, ' ', '')) + 1), 0) AS
      FROM Employee e
     LEFT JOIN Qualification q ON e.emp_ID = q.Emp_ID
    GROUP BY e.emp_ID, e.Job_ID
  ) AS emp_qual_counts
JOIN JobDepartment jd ON emp_qual_counts.Job_ID = jd.Job_ID
GROUP BY jd.jobdept
ORDER BY avg_qualifications_per_employee DESC;
```



department	avg_qualifications_per_employee
IT	3.25

# WHICH YEAR HAD THE MOST EMPLOYEES TAKING LEAVES?

```
SELECT  
    YEAR(date) AS leave_year,  
    COUNT(DISTINCT emp_ID) AS employees_on_leave  
FROM  
    Leaves  
GROUP BY  
    leave_year  
ORDER BY  
    employees_on_leave DESC  
LIMIT 2;
```



leave_year	employees_on_leave
2024	10

# WHAT IS THE AVERAGE NUMBER OF LEAVE DAYS TAKEN BY ITS EMPLOYEES PER DEPARTMENT?

```
jd.jobdept AS department,  
ROUND(AVG(emp_leave_count.total_leaves), 2) AS avg_leave_days_per_employee  
FROM (  
    SELECT  
        e.emp_ID,  
        e.Job_ID,  
        COUNT(l.leave_ID) AS total_leaves  
    FROM  
        Employee e  
    LEFT JOIN Leaves l ON e.emp_ID = l.emp_ID  
    GROUP BY  
        e.emp_ID, e.Job_ID  
) AS emp_leave_count  
JOIN JobDepartment jd ON emp_leave_count.Job_ID = jd.Job_ID  
GROUP BY  
    jd.jobdept  
ORDER BY  
    avg_leave_days_per_employee DESC;
```



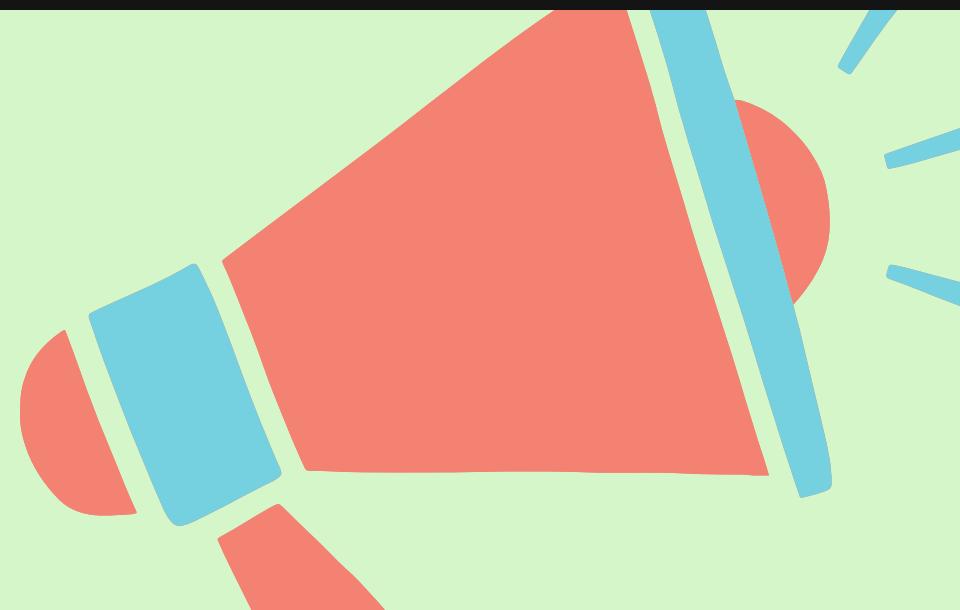
department	avg_leave_days_per_employee
HR	4.2

# WHICH EMPLOYEES HAVE TAKEN THE MOST LEAVES?

```

SELECT
    e.emp_ID,
    CONCAT(e.firstname, ' ', e.lastname) AS employee_name,1
    COUNT(l.leave_ID) AS total_leaves
FROM
    Employee e
JOIN
    Leaves l ON e.emp_ID = l.emp_ID2
GROUP BY
    e.emp_ID3
ORDER BY
    total_leaves DESC;

```

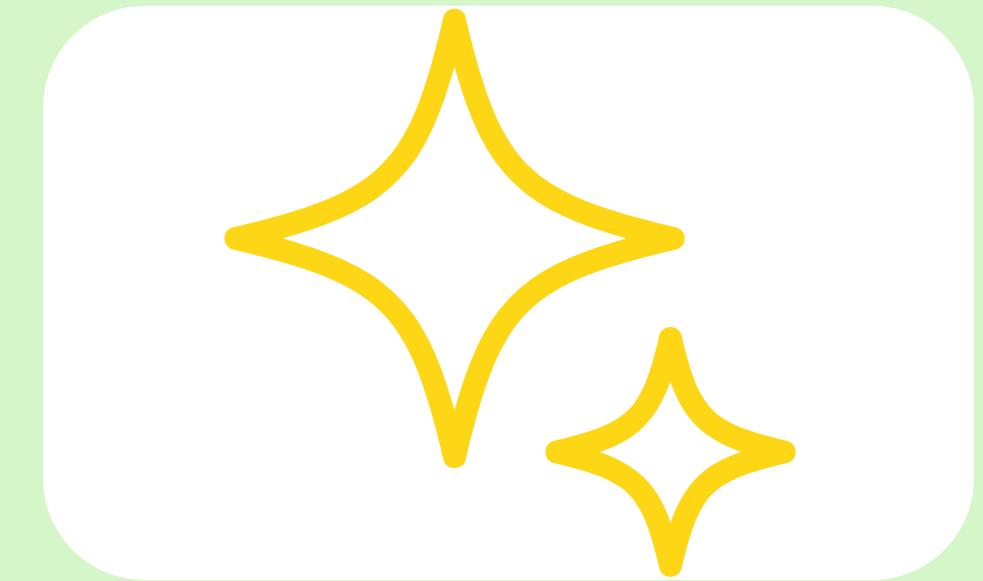


4

emp_ID	employee_name	total_leaves
102	John Smith	10

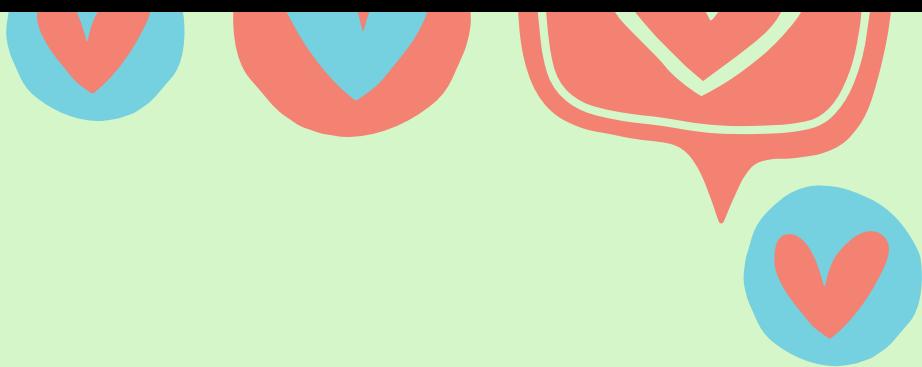
# WHAT IS THE TOTAL NUMBER OF LEAVE DAYS TAKEN COMPANY-WIDE?

```
SELECT  
    COUNT(*) AS total_leave_days  
FROM  
    Leaves;
```



```
| total_leave_days |  
|-----|  
| 60 |
```

# HOW DO LEAVE DAYS CORRELATE WITH PAYROLL AMOUNTS?



```
SELECT
    e.emp_ID,
    CONCAT(e.firstname, ' ', e.lastname) AS Employee_Name,
    sb.amount AS Base_Salary,
    SUM(p.total_amount) AS Total_Paid,
    (sb.amount - SUM(p.total_amount)) AS Total_Deduction
FROM
    Employee e
JOIN
    Payroll p ON e.emp_ID = p.emp_ID
JOIN
    SalaryBonus sb ON p.salary_ID = sb.salary_ID
GROUP BY
    e.emp_ID, Employee_Name, sb.amount
ORDER BY
    e.emp_ID;
```

Health Love Rest Friends Family Beauty



emp_ID	Employee_Name	Base_Salary	Total_Paid	Total_Deduction	Opportunity	Safety
101	Jane Doe	60000.00	57000.00	3000.00		

# WHAT IS THE TOTAL MONTHLY PAYROLL PROCESSED?

```
pm.payroll_month,  
pm.total_monthly_payroll,  
COALESCE(lv.total_leave_days, 0) AS total_leave_days  
FROM  
(  
    SELECT  
        DATE_FORMAT(date, '%Y-%m') AS payroll_month,  
        SUM(total_amount) AS total_monthly_payroll  
    FROM  
        Payroll  
    GROUP BY  
        payroll_month  
) pm  
LEFT JOIN  
(  
    SELECT  
        DATE_FORMAT(date, '%Y-%m') AS leave_month,  
        COUNT(*) AS total_leave_days  
    FROM  
        Leaves  
    GROUP BY  
        leave_month  
) lv ON pm.payroll_month = lv.leave_month  
ORDER BY  
pm.payroll_month;
```



payroll_month	total_monthly_payroll	total_leave_days
2024-01	100000.00	5

# WHAT IS THE AVERAGE BONUS GIVEN PER DEPARTMENT?

```
SELECT jd.jobdept AS department,  
       ROUND(AVG(sb.bonus), 2) AS average_bonus  
FROM SalaryBonus sb  
JOIN JobDepartment jd ON sb.Job_ID = jd.Job_ID  
GROUP BY jd.jobdept  
ORDER BY average_bonus DESC;
```



department	average_bonus
IT	5000.00

# WHICH DEPARTMENT RECEIVES THE HIGHEST TOTAL BONUSES?

```
SELECT jd.jobdept AS department,  
       SUM(sb.bonus) AS total_bonus  
FROM SalaryBonus sb  
JOIN JobDepartment jd ON sb.Job_ID = jd.Job_ID  
GROUP BY jd.jobdept  
ORDER BY total_bonus DESC  
LIMIT 5;
```



department	total_bonus
Finance	30000.00

# WHAT IS THE AVERAGE VALUE OF TOTAL\_AMOUNT AFTER CONSIDERING LEAVE DEDUCTIONS?

```
SELECT  
    ROUND(AVG(total_amount), 2) AS average_net_salary  
FROM  
    Payroll;
```



average_net_salary
58000.00

# WHICH YEAR HAD THE HIGHEST NUMBER OF EMPLOYEE PROMOTIONS?

```
SELECT  
    YEAR(Date_In) AS promotion_year,  
    COUNT(*) AS total_promotions  
FROM  
    Qualification  
GROUP BY  
    promotion_year  
ORDER BY  
    total_promotions DESC  
LIMIT 5;
```



promotion_year	total_promotions
2023	12



# CONCLUSION

- SQL IS A POWERFUL TOOL FOR REAL-TIME BUSINESS INTELLIGENCE.
- DATA RELATIONSHIPS, WHEN MODELED PROPERLY, CAN REVEAL TRENDS THAT SUPPORT DATA-DRIVEN DECISION-MAKING.
- AUTOMATING HR ANALYTICS REDUCES MANUAL EFFORT AND INCREASES ACCURACY, ULTIMATELY ENHANCING OPERATIONAL EFFICIENCY.



THANK  
YOU

