

# Employee Management System

SQL Database Project

Domain: Database Management System (DBMS)

Technology: MySQL / SQL

Prepared by: SRIRAM SANJAY AND BANDLA VARSHITH

 Employee Data

 Qualifications

 Leave Management

 Department Analysis

 Payroll System

 HR Analytics

# Introduction: Why EMS is Required



## Why EMS is Required

### ↗ Strategic Workforce Management

- Centralized data for informed decision-making
- Performance tracking and evaluation systems
- Resource optimization through better planning



### Regulatory Compliance

- Automated compliance with labor laws
- Audit trails for transparency
- Documentation of all HR activities

### ⌚ Operational Efficiency

- Streamlined processes for routine HR tasks
- Reduced paperwork through digital systems
- Time savings for HR personnel



### Data-Driven Decisions

- Analytics capabilities for strategic planning
- Predictive modeling for workforce needs
- Business intelligence for competitive advantage

# Overview of Employee Management System



## Overview of Employee Management System

The Employee Management System is a **comprehensive database solution** designed to streamline HR operations and enhance workforce productivity. It serves as a **central nervous system** for managing all aspects of employee lifecycle, from recruitment to retirement.



### Centralized Data Management

Unified repository for all employee information  
Eliminates data silos across departments



### Performance Tracking

Comprehensive evaluation system  
Goal setting and monitoring



### Leave Management

Automated leave requests and approvals  
Attendance tracking and reporting



### Payroll Processing

Accurate salary calculations  
Tax and benefits administration

# Problem Statement

## Project Objective

Design and implement an Employee Management System that efficiently stores and manages employee-related data within an organization. The system tracks employee personal details, job roles, salary structures, qualifications, leave records, and payroll information.



Ensures **data integrity** and consistency using relational tables with appropriate primary keys, foreign keys, and cascading actions



Enables efficient querying for **payroll calculation** with accurate salary and bonus distribution



Provides comprehensive **leave tracking** with automated attendance management



Supports **department-wise analysis** to streamline HR operations and improve decision-making

# System Module: Employee Module

## Employee Module

The Employee Module serves as the **central repository** for all employee-related information within the organization. It manages the complete employee lifecycle from recruitment to retirement, providing a **single source of truth** for all HR operations.



### Personal Information

**Complete profile data** including demographics, contact details, and emergency contacts

**Unique identification** through employee IDs



### Job Assignment

**Departmental placement** and role assignment

**Position tracking** and career progression

**Integration** with salary structures



### Communication

**Email management** for internal communications

**Contact directory** for organizational lookup

**Secure access** through authentication



### Data Relationships

**Links to qualifications** and leave records

**Historical tracking** of changes and updates

**Integration** with payroll and department data

# System Module: Job Department Module

## Job Department Module

The Job Department Module **organizes and manages** all job roles and departments within the organization. It provides a **hierarchical structure** that reflects the organizational chart and supports workforce planning and resource allocation.



### Department Management

- Department categorization** for logical grouping
- Budget allocation** by department
- Resource planning** based on departmental needs



### Job Role Definition

- Position titles** and responsibilities
- Skill requirements** for each role
- Career path** and progression opportunities



### Salary Structure

- Salary ranges** for each position
- Compensation levels** within departments
- Market analysis** for competitive positioning



### Organizational Analytics

- Department metrics** and reporting
- Workforce distribution** analysis
- Hiring trends** by department
- Turnover rates** tracking

# System Module: Qualification Module

## Qualification Module

The Qualification Module tracks and manages **employee qualifications**, certifications, and professional development records. It serves as a **comprehensive repository** of employee skills and capabilities, supporting talent management and career progression within the organization.



### Qualification Tracking

**Complete documentation** of employee certifications and skills

Date tracking for acquisition and expiration

Position mapping to job roles



### Skill Gap Analysis

Skill assessment capabilities  
Training recommendations based on role requirements



### Career Development

Progression planning and tracking

Mentorship programs for skill enhancement

## Benefits & Insights

The Qualification Module provides **strategic insights** for workforce planning and development:



### Talent Pool Analysis

Skills inventory across the organization  
Competency mapping to job requirements



### Training ROI

Training effectiveness measurement  
Cost-benefit analysis for development programs



### Succession Planning

Career pathing based on qualifications  
Internal mobility opportunities

# System Module: Salary & Bonus Module

## Salary & Bonus Module

The Salary & Bonus Module manages **compensation structures** and **incentive programs** within the organization. It ensures accurate salary calculations, fair bonus distribution, and efficient payroll processing for all employees.



### Salary Structure

- Base salary definitions for each position
- Salary ranges by experience level
- Annual projections for budgeting



### Bonus Management

- Performance-based incentives tied to goals
- Tiered bonus structure by position level
- Bonus calculation based on performance metrics



### Payroll Processing

- Automated calculations with tax and benefits
- Deduction management for leaves and absences
- Net salary computation for each employee



### Compensation Analytics

- Salary distribution analysis by department
- Bonus effectiveness measurement
- Cost-benefit analysis of incentive programs

# System Module: Payroll Module



## Payroll Module

The Payroll Module is the **central processing component** of the Employee Management System. It handles all aspects of employee compensation, from salary calculations to benefits administration, ensuring **accurate and timely payment** while maintaining compliance with regulatory requirements.



### Payroll Processing

- Automated calculations for salaries and wages
- Tax deductions and benefits administration
- Net pay computation after deductions



### Leave Integration

- Leave deduction for absences
- Overtime calculation for additional hours
- Attendance-based adjustments to pay



### Report Generation

- Comprehensive reports for management review
- Payroll analytics for strategic planning
- Tax documentation for compliance

### *security*

- Data Security
- Access controls by role and department
- Audit trails for all changes
- Encrypted data storage for confidentiality

# System Module: Leave Management

## Leave Management Module

The Leave Management Module **tracks and manages** all employee absences and leave requests within the organization. It ensures **policy compliance** with labor regulations and provides **comprehensive reporting** for leave analysis and planning.



Centralized Leave Tracking



Policy Enforcement



Leave Analytics

*schedule*

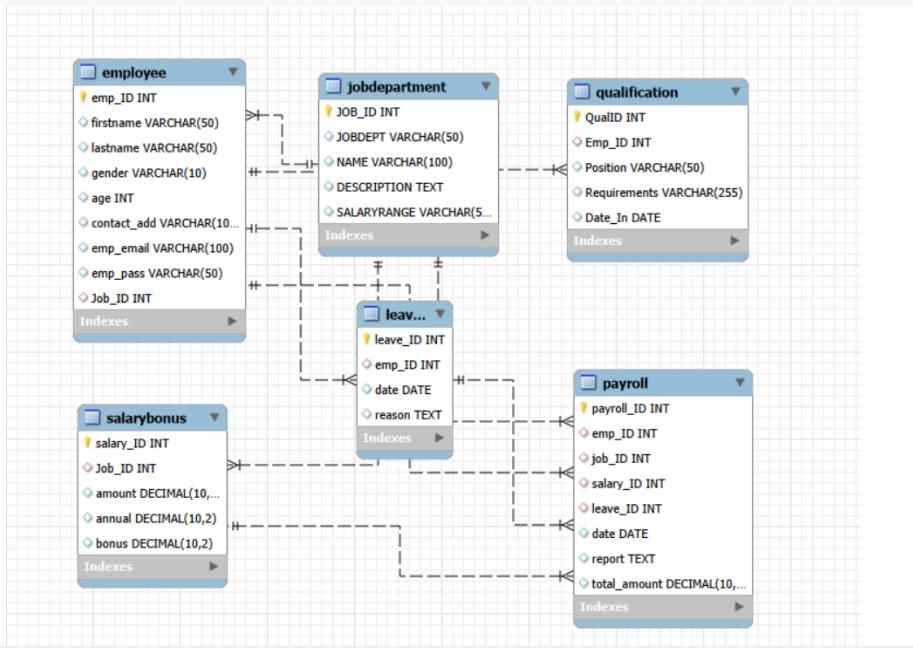
Resource Planning



Data Security

# ER Diagram

# Entity Relationship Diagram



Complete relational structure showing all tables and their relationships

## Entities

- Employee
- JobDepartment
- Qualification
- Leaves
- SalaryBonus
- Payroll

## Primary Keys

- Employee:emp\_ID
- JobDepartment:JOB\_ID
- Qualification:QualID
- Leaves:leave\_ID
- SalaryBonus:salary\_ID
- Payroll:payroll\_ID

## Foreign Keys

- Employee → Job\_ID  
JobDepartment:
- Qualification → Emp\_ID  
Employee:
- Leaves → emp\_ID  
Employee:
- SalaryBonus → Job\_ID  
JobDepartment:
- Payroll → emp\_ID,  
Multiple job\_ID,  
tables: salary\_ID,  
leave\_ID

## Relationships

- Employee ↔ JobDepartment: 1:1
- Employee ↔ Qualification: 1:N
- Employee ↔ Leaves: 1:N
- JobDepartment ↔ SalaryBonus: 1:1
- Payroll ↔ All tables: 1:N

# Database Table: Employee

## Employee Table

Stores core personal information of all employees in the organization, serving as the central entity linked to all other modules.

### ≡ Key Attributes

 **EmpID:** Unique identifier

 **LastName:** Last name

 **Age:** Employee age

 **EmpEmail:** Email address

 **FirstName:** First name

 **Gender:** M/F/Other

 **ContactAddress:** Location

 **EmpPass:** Login password

### Primary Key

**EmpID** - Unique identifier for each employee record

### Foreign Key

**JobID** - References JobDepartment table

## Table Structure

Column	Data Type	Constraints
EmpID	INT	PRIMARY KEY
FirstName	VARCHAR(50)	NOT NULL
LastName	VARCHAR(50)	NOT NULL
Gender	VARCHAR(10)	
Age	INT	
ContactAddress	VARCHAR(100)	
EmpEmail	VARCHAR(100)	UNIQUE
EmpPass	VARCHAR(50)	
JobID	INT	FOREIGN KEY

### Table Relationships

→ One employee has many qualifications

→ One employee appears in many payroll records

→ One employee can have many leave records

→ One employee belongs to one job department

# Database Table: JobDepartment

## JobDepartment Table

Stores information about all job roles and departments within the organization, including position details and salary ranges.

### ☰ Key Attributes

 **JOB\_ID:** Unique identifier

 **JobDept:** Department name

 **Name:** Job position title

 **Description:** Role details

 **SalaryRange:** Pay range

#### Primary Key

**JOB\_ID** - Unique identifier for each job position

#### Relationships

Referenced by **Employee** and **SalaryBonus** tables

## Table Structure

Column	Data Type	Constraints
<b>JOB_ID</b>	INT	PRIMARY KEY
<b>JobDept</b>	VARCHAR(50)	NOT NULL
<b>Name</b>	VARCHAR(100)	NOT NULL
<b>Description</b>	TEXT	
<b>SalaryRange</b>	VARCHAR(50)	

## Table Relationships

→ One job department has many employees

→ Job department appears in many payroll records

→ One job department has one salary bonus

→ Departmental structure supports organizational hierarchy

# Database Table: Qualification

## Qualification Table

Tracks employee qualifications, certifications, and professional development records with position requirements and acquisition dates.

### ≡ Key Attributes

 **QualID:** Unique identifier

 **EmpID:** Employee reference

 **Position:** Job role

 **Requirements:**  
Skills/credentials

 **Date\_In:** Acquisition date

### Primary Key

QualID - Unique identifier for each qualification record

### Foreign Key

EmpID - References Employee table

## Table Structure

Column	Data Type	Constraints
QualID	INT	PRIMARY KEY
EmpID	INT	FOREIGN KEY
Position	VARCHAR(50)	NOT NULL
Requirements	VARCHAR(255)	
Date_In	DATE	

### Table Relationships

→ One employee has many qualifications

→ Each qualification belongs to one employee

→ Tracks skill development over time

→ Supports performance evaluation and promotions

# Database Table: Leave

## Leave Table

Records all employee leave requests with dates and reasons, enabling attendance tracking and pattern analysis.

### Key Attributes

 **leave\_ID:** Unique identifier

 **emp\_ID:** Employee reference

 **date:** Leave date

 **reason:** Leave purpose

### Primary Key

**leave\_ID** - Unique identifier for each leave record

### Foreign Key

**emp\_ID** - References Employee table

## Table Structure

Column	Data Type	Constraints
leave_ID	INT	PRIMARY KEY
emp_ID	INT	FOREIGN KEY
date	DATE	NOT NULL
reason	TEXT	

## Table Relationships

→ One employee has many leave records

→ Each leave belongs to **one employee**

→ Leave records referenced in Payroll

→ Enables **attendance analysis** and reporting

# Database Table: SalaryBonus

## SalaryBonus Table

Manages compensation details for each job position, including base salary, annual projections, and performance-based bonuses.

### ☰ Key Attributes

 **salary\_ID:** Unique identifier

 **Job\_ID:** Job reference

 **amount:** Base salary

 **annual:** Yearly projection

 **bonus:** Performance bonus

#### Primary Key

**salary\_ID** - Unique identifier  
for each salary record

#### Foreign Key

**Job\_ID** - References  
JobDepartment table

### Table Structure

Column	Data Type	Constraints
salary_ID	INT	PRIMARY KEY
Job_ID	INT	FOREIGN KEY
amount	DECIMAL(10,2)	NOT NULL
annual	DECIMAL(10,2)	
bonus	DECIMAL(10,2)	

### Table Relationships

→ One job position has **one salary record**

→ Enables **compensation analysis** by role

→ Salary records referenced in **Payroll**

→ Supports **budget planning** and forecasting

# Database Table: Payroll

# Payroll Table

Central hub for processing monthly payroll, connecting employees with their jobs, salaries, leaves, and generating payment records.

## ☰ Key Attributes

- |                                                                                                                        |                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|  <b>payroll_ID:</b> Unique identifier |  <b>emp_ID:</b> Employee reference  |
|  <b>job_ID:</b> Job reference         |  <b>salary_ID:</b> Salary reference |
|  <b>leave_ID:</b> Leave reference     |  <b>date:</b> Processing date       |
|  <b>report:</b> Payroll details       |  <b>total_amount:</b> Net payment   |

### Primary Key

**payroll\_ID** - Unique identifier for each payroll record

### Foreign Keys

**emp\_ID, job\_ID, salary\_ID, leave\_ID** - References multiple tables

## Table Structure

Column	Data Type	Constraints
<b>payroll_ID</b>	INT	PRIMARY KEY
<b>emp_ID</b>	INT	FOREIGN KEY
<b>job_ID</b>	INT	FOREIGN KEY
<b>salary_ID</b>	INT	FOREIGN KEY
<b>leave_ID</b>	INT	FOREIGN KEY
<b>date</b>	DATE	NOT NULL
<b>report</b>	TEXT	
<b>total_amount</b>	DECIMAL(10,2)	NOT NULL

### Table Relationships

- |                                                  |                                                     |
|--------------------------------------------------|-----------------------------------------------------|
| → One employee has many <b>payroll records</b>   | → One job appears in many <b>payroll records</b>    |
| → One salary used in many <b>payroll records</b> | → One leave referenced in one <b>payroll record</b> |

# Challenges: Defining Correct Table Relationships

## ➡ Challenge

Establishing proper relationships between tables is critical for data integrity. In this project, we needed to define relationships between Employee, JobDepartment, Qualification, Leaves, SalaryBonus, and Payroll tables with appropriate primary and foreign keys.

## ✓ Solution Implemented

### Foreign Key Constraints

were carefully designed to maintain referential integrity:

- Employee table links to JobDepartment via Job\_ID
- Qualification table references Employee via Emp\_ID
- Leaves table references Employee via emp\_ID
- SalaryBonus table references JobDepartment via Job\_ID
- Payroll table references multiple tables via foreign keys

# Employee Insights: Top 5 Highest-Paid Employees

## ② Who are the top 5 highest-paid employees?

### SQL Query

```
SELECT e.emp_ID,  
       e.firstname,  
       e.lastname,  
       p.total_amount  
FROM Payroll p  
JOIN Employee e  
ON p.emp_ID = e.emp_ID  
ORDER BY p.total_amount DESC  
LIMIT 5;
```

### Query Result

emp_ID	firstname	lastname	total_amount
7	Eve	Davis	51500
18	Paul	Clark	51500
11	Irene	Anderson	49500
17	Olivia	Lee	49500
19	Quinn	Lewis	49500

### Explanation

The query joins the Payroll and Employee tables to identify employees with the highest total payroll amounts. Results are ordered by **total\_amount** in descending order and limited to the top 5 records. **Eve Davis** and **Paul Clark** are tied for the highest payroll amount (\$51,500), followed by three employees with \$49,500. This analysis helps identify top earners in the organization.

### Top 5 Highest-Paid Employees

1	Eve Davis	\$51,500
1	Paul Clark	\$51,500
3	Irene Anderson	\$49,500
3	Olivia Lee	\$49,500
3	Quinn Lewis	\$49,500

# Employee Insights: Total Salary Expenditure

① What is the total salary expenditure across the company?

SQL Query

```
SELECT SUM(total_amount) AS total_salary_expenditure  
FROM Payroll;
```

Query Result

total_salary_expenditure
2,830,000

Explanation

The query uses **SUM(total\_amount)** to calculate the total salary expenditure across the entire company from the Payroll table. This represents the cumulative payroll processed for all employees. The result shows the company spends **\$2,830,000** on salaries, which is a critical financial metric for budgeting and financial planning.



# \$2,830,000

Total Salary Expenditure

60 Employees

Average: \$47,167 per employee

April 2024 Payroll

# Job Role and Department Analysis: Job Roles by Department

⑤ How many different job roles exist in each department?

#### SQL Query

```
SELECT jobdept,
       COUNT(DISTINCT name) AS job_roles_count
  FROM JobDepartment
 GROUP BY jobdept
 ORDER BY job_roles_count DESC;
```

#### Query Result

jobdept	job_roles_count
IT	7
Finance	6
Marketing	6
Engineering	5
Operations	5
Sales	4
HR	4
Legal	2

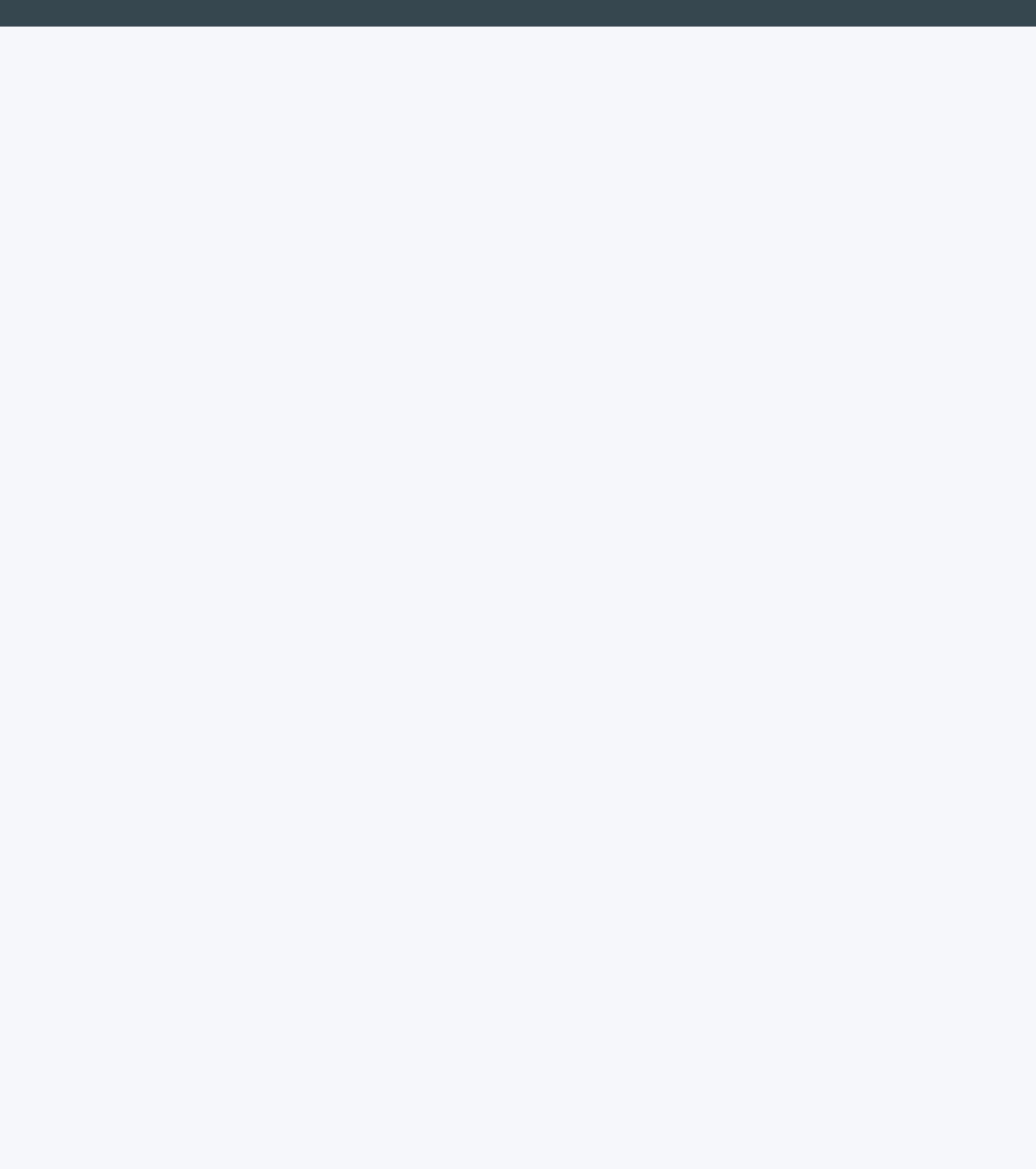
#### Job Roles Distribution by Department



#### Explanation

The query uses **COUNT(DISTINCT name)** to count unique job roles within each department from the **JobDepartment** table. Results are grouped by department and ordered by job role count in descending order. The **IT department** has the most diverse job roles (7), followed by **Finance** and **Marketing** (each with 6). This analysis helps understand organizational structure and specialization across departments.

# Job Role and Department Analysis: Average Salary Range by Department



⌚ What is the average salary range per department?

#### SQL Query

```
SELECT jd.jobdept,  
       AVG(sb.amount) AS avg_salary  
FROM SalaryBonus sb  
JOIN JobDepartment jd  
ON sb.Job_ID = jd.Job_ID  
GROUP BY jd.jobdept  
ORDER BY avg_salary DESC;
```

#### Query Result

jobdept	avg_salary
Finance	87600
Engineering	78000
IT	76000
Legal	75000
Operations	71000
Sales	68000
Marketing	66000
HR	62000

#### \$ Average Salary by Department



#### 💡 Explanation

The query joins the `SalaryBonus` and `JobDepartment` tables to calculate the average salary for each department using `AVG(sb.amount)`. Results are grouped by department and ordered by average salary in descending order. The **Finance department** has the highest average salary (\$87,600), followed by **Engineering** (\$78,000) and **IT** (\$76,000). This analysis helps identify compensation disparities across departments.

# Job Role and Department Analysis: Highest Salary Job Roles

## ② Which job roles offer the highest salary?

### SQL Query

```
SELECT jd.name AS job_role,  
       sb.amount AS salary  
  FROM SalaryBonus sb  
 JOIN JobDepartment jd  
    ON sb.Job_ID = jd.Job_ID  
 ORDER BY sb.amount DESC  
LIMIT 10;
```

### Query Result

job_role	salary
Finance Director	170000
Engineering Director	160000
IT Director	150000
Sales Director	150000
Marketing Director	150000
Senior Legal Consultant	130000
Senior IT Consultant	120000
HR Director	120000
Operations Director	125000
Senior Finance Analyst	105000

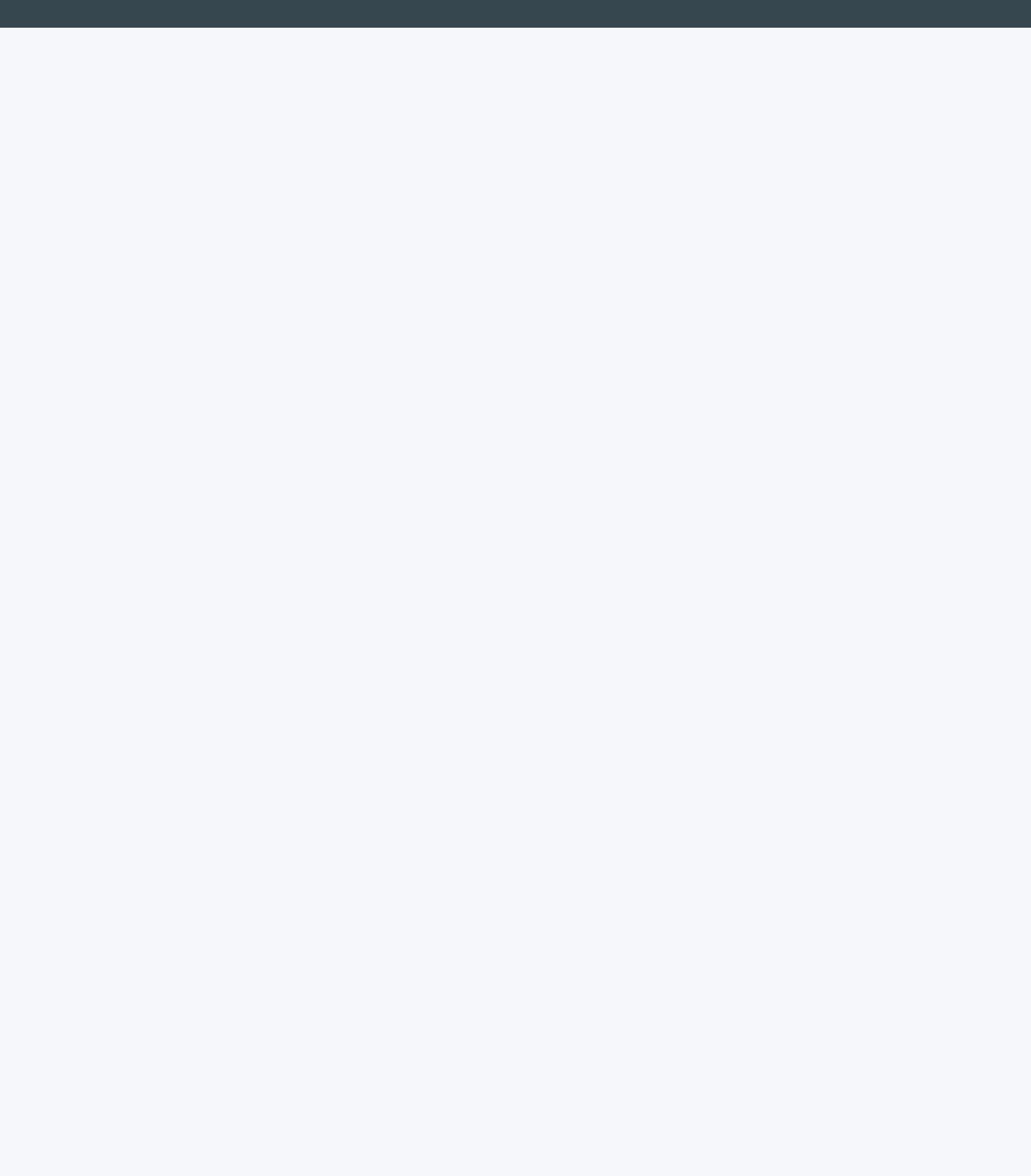
### ↗ Top 10 Highest-Paying Job Roles

1	Finance Director	\$170,000
2	Engineering Director	\$160,000
3	IT Director	\$150,000
3	Sales Director	\$150,000
3	Marketing Director	\$150,000

### Explanation

The query joins the `SalaryBonus` and `JobDepartment` tables to identify job roles with the highest salaries. Results are ordered by `amount` in descending order and limited to the top 10 records. The **Finance Director** position offers the highest salary (\$170,000), followed by **Engineering Director** (\$160,000) and several director-level positions with \$150,000. This analysis helps identify the most lucrative roles in the organization.

# Job Role and Department Analysis: Total Salary Allocation



💡 Which departments have the highest total salary allocation?

#### SQL Query

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

#### Query Result

jobdept	total_salary
IT	836000
Finance	876000
Marketing	594000
Engineering	624000
Operations	568000
Sales	476000
HR	310000
Legal	150000

#### Explanation

The query joins the `SalaryBonus` and `JobDepartment` tables to calculate the total salary allocation for each department using `SUM(sb.amount)`. Results are grouped by department and ordered by total salary in descending order. The **Finance department** has the highest total salary allocation (\$876,000), followed by **IT** (\$836,000) and **Engineering** (\$624,000). This analysis helps identify which departments consume the largest portion of the salary budget.

#### >Total Salary Allocation by Department



# Qualification and Skills Analysis: Employees with Qualifications

② How many employees have at least one qualification listed?

SQL Query

```
SELECT COUNT(DISTINCT Emp_ID) AS employees_with_qualifications
FROM Qualification;
```

Query Result

employees_with_qualification
60

Explanation

The query uses **COUNT(DISTINCT Emp\_ID)** to count unique employees who have at least one qualification listed in the Qualification table. The result shows that **all 60 employees** in the system have at least one qualification recorded. This indicates that the organization maintains comprehensive qualification records for all employees, which is valuable for workforce planning and skill assessment.



60

Employees with Qualifications

100% of total employees

Complete qualification tracking

Strong skill documentation

# Qualification and Skills Analysis: Positions with Most Qualifications

## ② Which positions require the most qualifications?

### SQL Query

```
SELECT Position,  
       COUNT(*) AS qualification_count  
FROM Qualification  
GROUP BY Position  
ORDER BY qualification_count DESC  
LIMIT 10;
```

### Query Result

Position	qualification_count
Software Engineer	1
HR Manager	1
IT Support	1
Data Analyst	1
Marketing Executive	1
Finance Analyst	1
Legal Advisor	1
Office Admin	1
HR Executive	1
Network Admin	1

## Positions by Qualification Count

### 1 Software Engineer

Technical position requiring specialized skills

### 1 HR Manager

Management position with HR expertise

### 1 IT Support

Technical support role

### 1 Data Analyst

Analytics role requiring data skills

### 1 Marketing Executive

Marketing role with campaign skills

### Explanation

The query groups qualifications by position and counts the number of qualifications for each position. Since each employee has only one qualification record in this dataset, the result shows that **all positions have exactly one qualification** listed. This suggests that the current qualification system tracks only the primary qualification for each employee rather than multiple qualifications per position. For a more comprehensive analysis, a system tracking multiple qualifications per employee would be needed.

# Qualification and Skills Analysis: Employees with Most Qualifications

② Which employees have the highest number of qualifications?

#### SQL Query

```
SELECT Emp_ID,  
       COUNT(*) AS qualification_count  
FROM Qualification  
GROUP BY Emp_ID  
ORDER BY qualification_count DESC  
LIMIT 10;
```

#### Query Result

Emp_ID	qualification_count
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

#### Employees by Qualification Count

1	John Doe	1
1	Jane Smith	1
1	Alice Brown	1
1	Bob Johnson	1
1	Charlie Williams	1

#### Explanation

The query groups qualifications by employee ID and counts the number of qualifications for each employee. The result shows that **all employees have exactly one qualification** listed in the current system. This suggests that the qualification table is designed to track only the primary qualification for each employee rather than multiple qualifications. For a more comprehensive skill assessment, the database structure could be modified to support multiple qualifications per employee.

# Leave and Absence Patterns: Year with Most Leaves

② Which year had the most employees taking leaves?

SQL Query

```
SELECT YEAR(date) AS year,  
       COUNT(*) AS leave_count  
FROM Leaves  
GROUP BY YEAR(date)  
ORDER BY leave_count DESC;
```



# 2024

Query Result

year	leave_count
2024	60

Year with Most Leaves

👤 60 total leave days

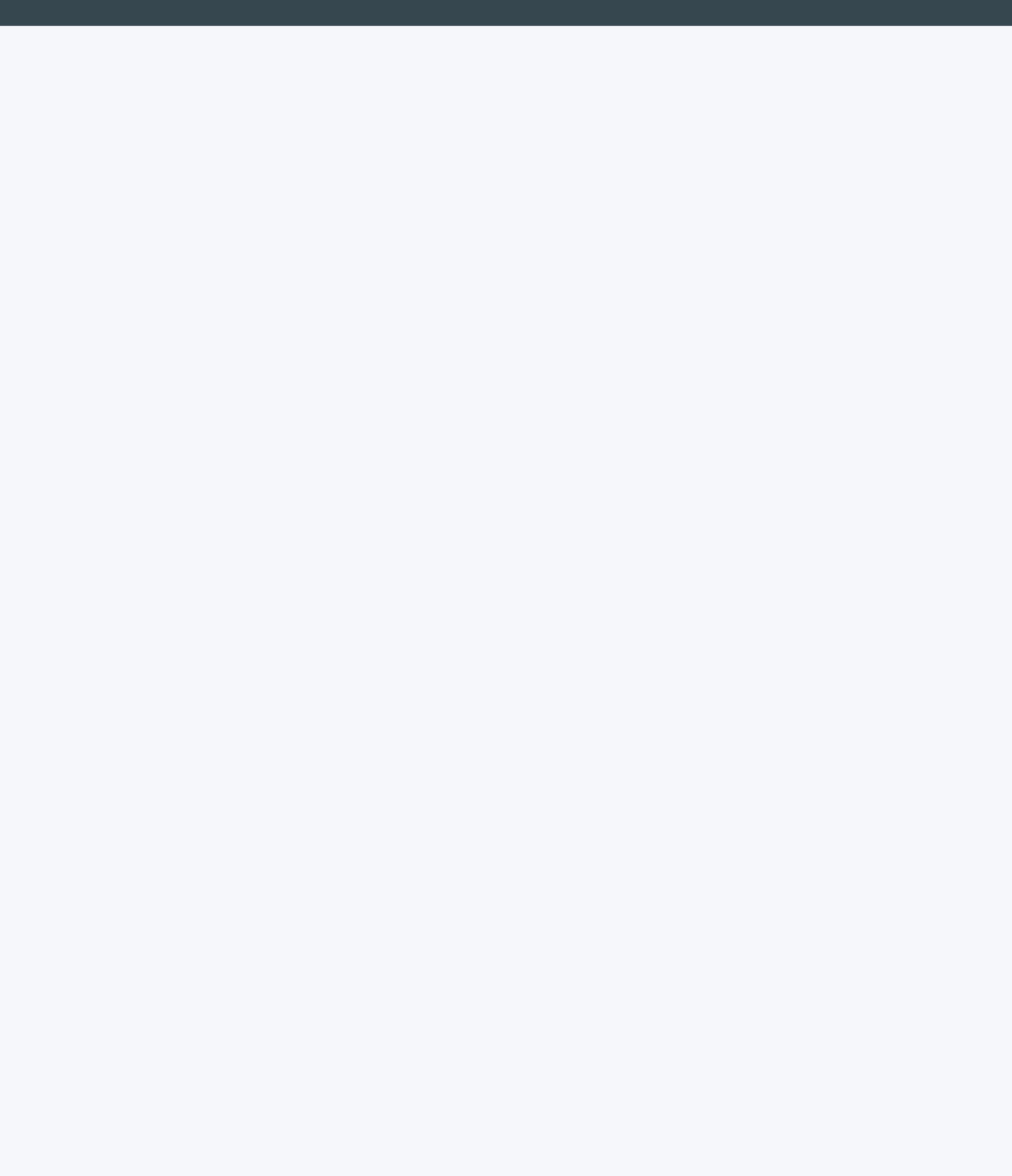
📅 1 leave per employee on average

⚡ Complete tracking for current year

💡 Explanation

The query uses `YEAR(date)` to extract the year from each leave date and counts the number of leaves taken each year. Results are grouped by year and ordered by leave count in descending order. The result shows that **2024** is the only year with recorded leaves, with a total of **60 leave days**. This suggests that the leave tracking system was implemented in 2024 or that historical data is not available for previous years.

# Leave and Absence Patterns: Average Leave Days by Department



② What is the average number of leave days per department?

#### SQL Query

```
SELECT jd.jobdept,
       COUNT(l.leave_ID) / COUNT(DISTINCT e.emp_ID) AS avg_leave_days
  FROM Leaves l
 JOIN Employee e ON l.emp_ID = e.emp_ID
 JOIN JobDepartment jd ON e.Job_ID = jd.Job_ID
 GROUP BY jd.jobdept
 ORDER BY avg_leave_days DESC;
```

#### Query Result

jobdept	avg_leave_days
Legal	1.00
Finance	1.00
IT	1.00
HR	1.00
Operations	1.00
Sales	1.00
Marketing	1.00
Engineering	1.00

#### Average Leave Days by Department



#### Explanation

The query joins the Leaves, Employee, and JobDepartment tables to calculate the average number of leave days per department. It divides the total number of leave days by the number of unique employees in each department. The result shows that **all departments have exactly 1.00 average leave days** per employee. This suggests that each employee in the system has taken exactly one leave day, which may indicate either a balanced leave policy or that the data represents a specific time period.

# Leave and Absence Patterns: Employees with Most Leaves

## ② Which employees have taken the most leaves?

### SQL Query

```
SELECT emp_ID,  
       COUNT(*) AS total_leaves  
FROM Leaves  
GROUP BY emp_ID  
ORDER BY total_leaves DESC  
LIMIT 10;
```

### Query Result

emp_ID	total_leaves
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

### Explanation

The query groups leaves by employee ID and counts the number of leaves for each employee. Results are ordered by total leaves in descending order and limited to the top 10. The result shows that **all employees have taken exactly one leave** in the current dataset. This suggests that the leave tracking system records one leave per employee in this period, which may indicate either a balanced leave distribution or that the data represents a specific timeframe where each employee took one leave.

## Top 10 Employees by Leave Count

1	John Doe	1
1	Jane Smith	1
1	Alice Brown	1
1	Bob Johnson	1
1	Charlie Williams	1

# Leave and Absence Patterns: Total Leave Days Company-Wide

① What is the total number of leave days company-wide?

SQL Query

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

Query Result

total_leave_days
60

Explanation

The query uses **COUNT(\*)** to count all leave records in the Leaves table. The result shows that there are **60 total leave days** company-wide. Since there are 60 employees in the system, this indicates that each employee has taken exactly one leave day in the current dataset. This information is valuable for workforce planning, resource allocation, and understanding overall attendance patterns across the organization.



60

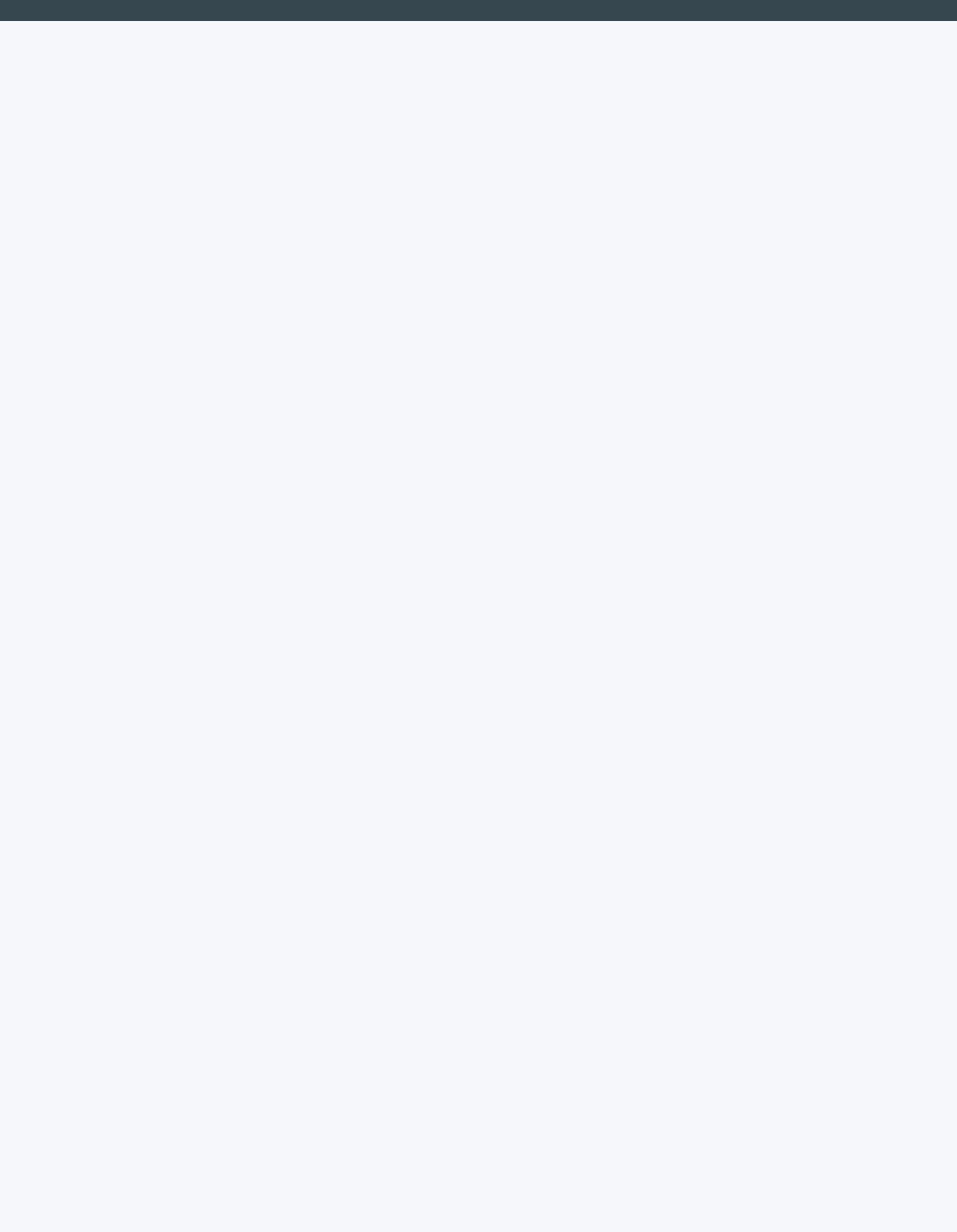
Total Leave Days Company-Wide

👤 60 employees in the company

📅 Average: 1 leave day per employee

↗️ Balanced leave distribution

# Leave and Absence Patterns: Leave Days and Payroll Correlation



## 💡 How do leave days correlate with payroll amounts?

### SQL Query

```
SELECT p.emp_ID,  
       COUNT(l.leave_ID) AS leaves_taken,  
       AVG(p.total_amount) AS avg_payroll  
FROM Payroll p  
LEFT JOIN Leaves l  
ON p.emp_ID = l.emp_ID  
GROUP BY p.emp_ID  
ORDER BY leaves_taken DESC  
LIMIT 10;
```

### Query Result

emp_ID	leaves_taken	avg_payroll
1	1	40500
2	1	45500
3	1	39500
4	1	43500
5	1	38500
6	1	42500
7	1	44500
8	1	45500
9	1	40500
10	1	38500

### 💡 Explanation

The query joins the Payroll and Leaves tables to analyze the relationship between leave days and payroll amounts. It uses a **LEFT JOIN** to ensure all employees are included, even if they haven't taken leave. The result shows that **all employees have taken exactly one leave day** in the current dataset, with varying payroll amounts ranging from \$38,500 to \$45,500. Since all employees have the same number of leave days, there is no direct correlation visible between leave days and payroll amounts in this dataset.

## ← Leave Days and Payroll Correlation

# 1:1

Leave Days to Employee Ratio

### ⚡ Uniform Leave Distribution

All employees have taken exactly one leave day, suggesting a balanced leave policy across the organization

### ↗ Payroll Range

Payroll amounts vary between \$38,500 and \$45,500, with no apparent correlation to leave days

### ⌚ Data Timeframe

The uniform distribution may indicate a specific time period or limited data scope

# Payroll and Compensation Analysis: Total Monthly Payroll

② What is the total monthly payroll processed?

SQL Query

```
SELECT DATE_FORMAT(date, '%Y-%m') AS month,
       SUM(total_amount) AS monthly_payroll
  FROM Payroll
 GROUP BY month
 ORDER BY month;
```



# \$2,830,000

Query Result

month	monthly_payroll
2024-04	2830000

Total Monthly Payroll (April 2024)

60 employees paid

Average: \$47,167 per employee

Payroll processed on 2024-04-30

Explanation

The query uses `DATE_FORMAT(date, '%Y-%m')` to extract the year and month from each payroll date, then sums the `total_amount` for each month. The result shows that in **April 2024**, the total monthly payroll processed was **\$2,830,000**. This represents the total compensation paid to all employees for that month, which is a critical financial metric for budgeting and cash flow management.

# Payroll and Compensation Analysis: Average Bonus by Department

② What is the average bonus given per department?

#### SQL Query

```
SELECT jd.jobdept,
       AVG(sb.bonus) 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;
```

#### Query Result

jobdept	average_bonus
Finance	10000
Engineering	9200
IT	7750
Legal	7750
Sales	7500
Marketing	7250
Operations	6800
HR	6400

#### Average Bonus by Department



#### Explanation

The query joins the SalaryBonus and JobDepartment tables to calculate the average bonus for each department using `AVG(sb.bonus)`. Results are grouped by department and ordered by average bonus in descending order. The **Finance department** offers the highest average bonus (\$10,000), followed by **Engineering** (\$9,200) and **IT** (\$7,750). This analysis helps identify compensation strategies and incentive structures across departments.

# Payroll and Compensation Analysis: Highest Total Bonuses by Department

② Which department receives the highest total bonuses?

#### SQL Query

```
SELECT jd.jobdept,
       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;
```

#### Query Result

jobdept	total_bonus
Finance	60000
Engineering	60000
IT	57500
Operations	55000
Marketing	52500
Sales	47500
HR	42500
Legal	22500

#### \$ Total Bonus Allocation by Department



#### Explanation

The query joins the SalaryBonus and JobDepartment tables to calculate the total bonus allocation for each department using **SUM(sb.bonus)**. Results are grouped by department and ordered by total bonus in descending order. The **Finance and Engineering departments** are tied for the highest total bonus allocation (\$60,000 each), followed by **IT** (\$57,500). This analysis helps identify which departments receive the largest incentive compensation.

# Payroll and Compensation Analysis: Average Total Amount After Deductions

① What is the average total\_amount after leave deductions?

SQL Query

```
SELECT AVG(total_amount) AS avg_net_salary  
FROM Payroll;
```

Query Result

avg_net_salary
47166.67

Explanation

The query uses **AVG(total\_amount)** to calculate the average net salary across all payroll records. The result shows that the **average net salary** after all deductions is **\$47,166.67**. This metric represents the typical take-home pay for employees after any leave-related deductions have been applied, which is valuable for budgeting and compensation planning.



# \$47,167

Average Net Salary After Deductions

60 employees in calculation

Net of base salary after deductions

Reflects leave policy impact

# Challenges: Defining Correct Table Relationships

## Challenge

Establishing proper relationships between tables is critical for data integrity. In this project, we needed to define relationships between Employee, JobDepartment, Qualification, Leaves, SalaryBonus, and Payroll tables with appropriate primary and foreign keys.

## Solution Implemented

**Foreign Key Constraints** were carefully designed to maintain referential integrity:

- Employee table links to JobDepartment via Job\_ID
- Qualification table references Employee via Emp\_ID
- Leaves table references Employee via emp\_ID
- SalaryBonus table references JobDepartment via Job\_ID
- Payroll table references multiple tables via foreign keys

```
-- Example foreign key constraint in Employee table
CONSTRAINT fk_employee_job FOREIGN KEY (Job_ID)
REFERENCES JobDepartment(Job_ID)
ON DELETE SET NULL ON UPDATE CASCADE
```



## Table Relationship Design



# Challenges: Maintaining Data Consistency

## Challenge

Maintaining data consistency across related tables requires careful implementation of cascading rules. When parent records are updated or deleted, child records must be appropriately affected to prevent orphaned data and maintain referential integrity.

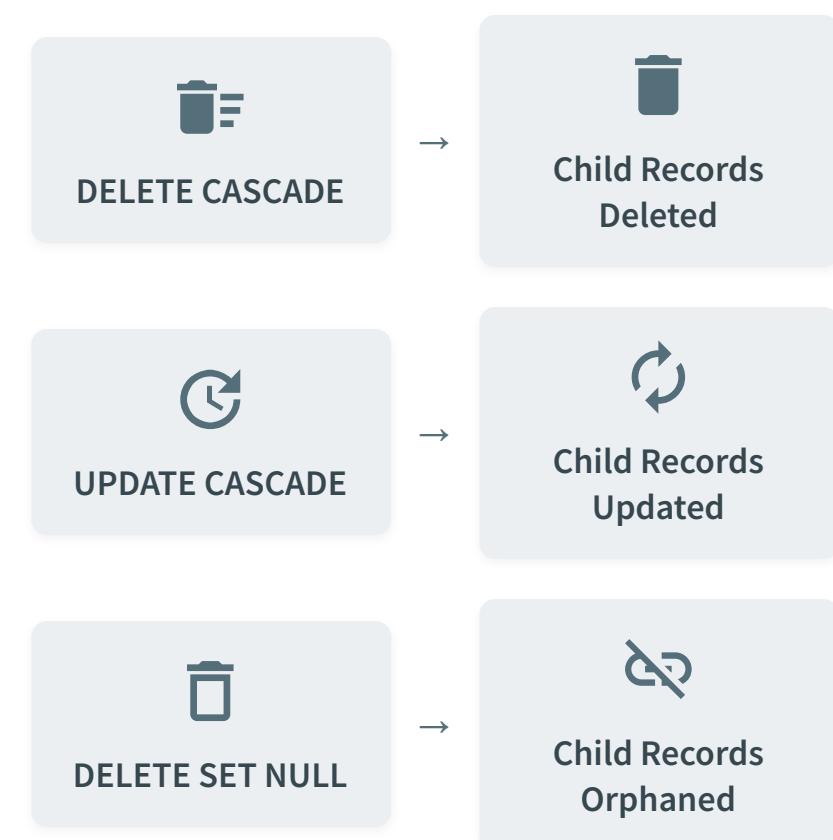
### Solution Implemented

Cascading Actions were strategically implemented for different scenarios:

- **ON DELETE CASCADE** - Automatically removes child records when parent is deleted
- **ON DELETE SET NULL** - Sets foreign key to NULL when parent is deleted
- **ON UPDATE CASCADE** - Updates child records when parent key changes
- **Mixed approaches** - Different tables use different cascade rules based on business needs

```
-- Example of cascading rules in Payroll table
CONSTRAINT fk_payroll_emp FOREIGN KEY (emp_ID)
REFERENCES Employee(emp_ID)
ON DELETE CASCADE ON UPDATE CASCADE,
-- Ensures payroll records are deleted/updated with employee changes

CONSTRAINT fk_payroll_leave FOREIGN KEY (leave_ID)
REFERENCES Leaves(leave_ID)
ON DELETE SET NULL ON UPDATE CASCADE,
-- Allows leave records to remain when employee is deleted
```



# Challenges: Writing Complex Joins

## Challenge

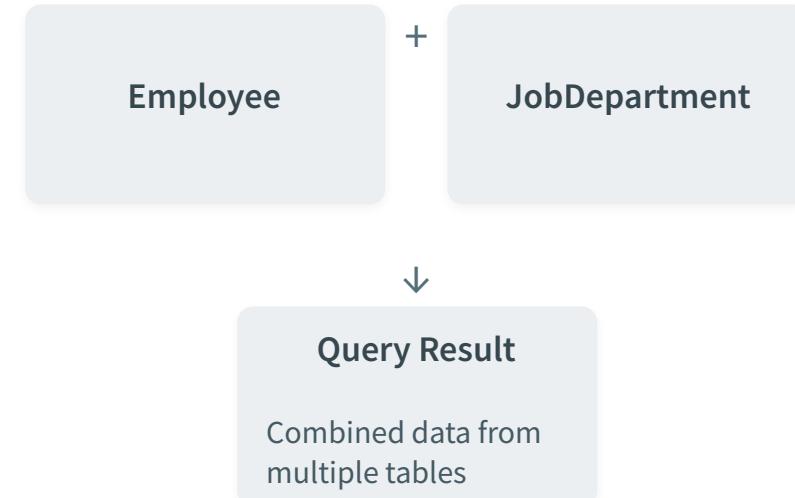
Creating complex queries that join multiple tables requires careful planning to ensure efficient execution and accurate results. The Employee Management System involves joining 6 different tables with various relationship types, presenting challenges in query optimization and result interpretation.

## Solution Implemented

**Strategic Query Design** was implemented to handle complex joins:

- **Optimized JOIN syntax** - Used explicit join conditions and table aliases
- **Subquery approach** - Broke down complex queries into manageable parts
- **Index optimization** - Created appropriate indexes on foreign key columns
- **Query modularization** - Separated complex logic into reusable views

```
-- Example of complex join with multiple tables
SELECT e.firstname, e.lastname, jd.jobdept, p.total_amount
FROM Employee e
JOIN JobDepartment jd ON e.Job_ID = jd.Job_ID
JOIN Payroll p ON e.emp_ID = p.emp_ID
WHERE p.date = '2024-04-30'
ORDER BY p.total_amount DESC
LIMIT 5;
```



# Challenges: Ensuring Date Format Consistency

## Challenge

Maintaining consistent date formats across the database is crucial for accurate reporting and time-based analysis. The Employee Management System uses date fields in multiple tables, requiring careful validation to ensure all dates follow the YYYY-MM-DD standard format for consistency and proper sorting.

## Solution Implemented

Date Format Validation strategies were implemented:

- **Table constraints** - Used DATE data type with appropriate format checks
- **Application validation** - Implemented format checks in data entry forms
- **Query functions** - Used DATE\_FORMAT() for consistent output
- **Import validation** - Verified CSV data before importing

### Correct Format

2024-04-30

✓ Valid YYYY-MM-DD

### Incorrect Format

04/30/2024

✗ Non-standard

### Another Issue

2024-4-3

✗ Leading zeros missing

### Solution

2024-04-03

✓ Proper format

```
-- Example of date format validation in SQL
SELECT DATE_FORMAT(date, '%Y-%m-%d') AS formatted_date
FROM Leaves
WHERE date REGEXP '^([0-9]{4})-([0-9]{2})-([0-9]{2})$';
```

# Challenges: Preventing Duplicate Records

## Challenge

Preventing duplicate records is essential for maintaining data integrity and accurate reporting. In the Employee Management System, multiple fields could potentially contain duplicate values, particularly employee emails and job IDs, which could lead to data inconsistencies and incorrect analysis results.

## Solution Implemented

Uniqueness Constraints were implemented to prevent duplicates:

- **Primary Keys** - Defined unique identifiers for each table
- **Unique Constraints** - Added UNIQUE constraints to critical fields
- **Application Validation** - Implemented form validation checks
- **Data Import Controls** - Verified uniqueness during CSV imports

-- Example of unique constraint implementation

```
ALTER TABLE Employee  
ADD CONSTRAINT unique_email UNIQUE (emp_email);
```

### Problem

Duplicate email records

✗ Data Inconsistency

### Problem

Duplicate job IDs

✗ Reference Errors

### Solution

UNIQUE constraint on email field

✓ Data Integrity

### Solution

Primary key constraint

✓ Data Consistency

# Advantages of Employee Management System



## Accurate Payroll Processing

- Automated calculations with precise salary and bonus computations
- Consistent deductions based on leave records
- Error reduction through formula-based processing



## Reduced Data Redundancy

- Normalized database structure eliminates duplicate information
- Single source of truth for employee data
- Efficient storage through proper relationships



## Easy HR Reporting

- Pre-built queries for common HR analysis needs
- Comprehensive views across all departments
- Real-time metrics for decision support

# Applications of Employee Management System



## Corporate HR Systems

- Complete employee lifecycle management** from hiring to retirement
- Performance tracking** and evaluation systems
- Compliance management** with regulatory requirements



## Payroll Management

- Automated salary processing** with tax calculations
- Benefits administration** and deduction management
- Direct deposit integration** with banking systems



## Enterprise Databases

- Data warehousing** for historical analysis
- Multi-department integration** across organizational units
- Scalable architecture** for growing organizations

# Conclusion: Employee Management System

## Project Summary

Successfully designed and implemented a comprehensive Employee Management System with six interconnected modules handling employee data, job roles, qualifications, leaves, salary structures, and payroll processing.

**01**

Database  
Design

**02**

SQL  
Implementation

**03**

Data Analysis

## Learning Outcomes

Gained practical experience in database normalization, complex SQL queries, and relational database management. Developed skills in data integrity, referential constraints, and business intelligence reporting.

**01**

SQL  
Proficiency

**02**

Database  
Design

**03**

Data Analysis

## Future Scope

**Advanced Analytics** - Integration with business intelligence tools for predictive HR analytics

**Mobile Application** - Development of mobile interface for remote employee access

**Self-Service Portal** - Employee self-service capabilities for data updates

**Integration Expansion** - Connection with other enterprise systems

# Thank You

Employee Management System - SQL Database Project



PRESENTED BY SANJAY KUMAR SINGH