**SVKM’s NMIMS**

**School of Technology Management & Engineering, Mumbai**

**A.Y. 2023 - 24**

**Course: Database Management Systems**

**Project Report**

|  |  |  |
| --- | --- | --- |
| **Program** | B.Tech Integrated Computers | |
| **Semester** | VIII | |
| **Name of the Project:** | Employee Management System | |
|  | | |
| **Details of Project Members** |  |  |
| **Batch** | **Roll No.** | **Name** |
| B1 | C026 | Aryan desai |
| B1 | C022 | Keya shah |
| B1 | C030 | Demira Ramnani |
| **Date of Submission:** 23/03/2024 | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| **Roll No.** | **Name:** | **Contribution** |
| C026 | Aryan desai | Creation of Schema, Report Formation |
| C022 | Keya shah | Formation of select queries, Report Content |
| C030 | Demira Ramnani | ER Diagram,Relational Model, Debugging |

**GitHub link of your project:**

**Project Report**

**Employee Management System**

**By**

**Aryan desai, C026**

**Keya shah, C022**

**Demira Ramnani, C030**

**Course: Database Management System**

**AY: 2023-24**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline |  |
| **2** | Components of Database Design |  |
| **3** | Entity Relationship Diagram |  |
| **4** | Relational Model |  |
| **5** | Normalization |  |
| **6** | SQL Queries |  |
| **7** | Learning from the Project |  |
| **8** | Project Demonstration |  |
| **9** | Self-learning beyond classroom |  |
| **10** | Learning from the project |  |
| **8** | Challenges faced |  |
| **9** | Conclusion |  |

1. **Storyline  
   Executive Summary:**

Given the ever-changing landscape of managing employees, this project seeks to optimize the efficiency and effectiveness of an Employee Management System (EMS) through detailed data analysis and strategic recommendations. By leveraging SQL database querying and analysis, our team has delved into the system's data to uncover valuable insights and propose actionable solutions.

**Introduction:**   
The employee management system is always changing, so we need to keep improving how we manage employees. Our project uses SQL to help make better decisions about managing employees.

**Methodology:**

We conducted a thorough analysis of the employee management system's database, including employee records, performance metrics, training history, and scheduling data. Using SQL queries, we processed and examined these datasets to identify trends, relationships, and opportunities for enhancement..

1. **Components of Database Design**

Components of Database Design:

1. Entities Represented by tables, such as `Department`, `Employee`, `Salary`, etc.

2. Attributes: Fields or columns in each table, like `department\_name`, `employee\_id`, `salary`, etc.

3. Relationships: Connections between entities, like one-to-many or many-to-many relationships.

4. Constraints: Rules that enforce data integrity, such as primary keys, foreign keys, and unique constraints.

Tables, Relationships, Participation, and Attributes:

1. Department:

- Attributes:

- department\_id (Primary Key)

- department\_name

- manager\_id

- Comp\_city

- Comp\_State

- Relationships:

- None

- Participation:

- All departments must have a manager, so participation is total on `manager\_id` (1:1).

2. Employee:

- Attributes:

- employee\_id (Primary Key)

- first\_name

- last\_name

- email

- phone\_number

- hire\_date

- department\_id (Foreign Key)

- Relationships:

- One department can have many employees (1:M with `Department`).

- Participation:

- All employees must belong to a department, so participation is total on `department\_id` (M:1).

3. Salary:

- Attributes:

- salary\_id (Primary Key)

- employee\_id (Foreign Key)

- salary

- start\_date

- end\_date

- Relationships:

- Each employee can have multiple salary entries (1:M with `Employee`).

- Participation:

- Every salary entry must be linked to an employee, so participation is total on `employee\_id` (M:1).

4. Address:

- Attributes:

- address\_id (Primary Key)

- employee\_id (Foreign Key)

- street

- city

- state

- postal\_code

- Relationships:

- Each employee can have one address (1:1 with `Employee`).

- Participation:

- All employees must have an address, so participation is total on `employee\_id` (1:1).

5. Project:

- Attributes:

- project\_id (Primary Key)

- project\_name

- start\_date

- end\_date

- project\_loc

- Relationships:

- Employees can work on multiple projects, and projects can have multiple employees (M:N with `Employee` through `Employee\_Project`).

- Participation:

- Projects can exist without employees, and employees can exist without projects, so participation is partial on both sides.

6. Employee\_Project:

- Attributes:

- employee\_id (Foreign Key)

- project\_id (Foreign Key)

- Relationships:

- Represents the many-to-many relationship between `Employee` and `Project`.

- Participation:

- Every employee must be assigned to at least one project, and every project must have at least one employee, so participation is total on both sides (M:N).

7. Task:

- Attributes:

- task\_id (Primary Key)

- task\_name

- project\_id (Foreign Key)

- Relationships:

- Each task is associated with one project (1:M with `Project`).

- Participation:

- All tasks must be part of a project, so participation is total on `project\_id` (M:1).

8. Attendance:

- Attributes:

- attendance

- employee\_id (Foreign Key)

- date

- Relationships:

- Each attendance record is linked to one employee (1:M with `Employee`).

- Participation:

- All employees must have an attendance record, so participation is total on `employee\_id` (M:1).

9. Emp\_Leave:

- Attributes:

- leave\_id (Primary Key)

- employee\_id (Foreign Key)

- start\_date

- end\_date

- reason

- status

- Relationships:

- Each leave request is associated with one employee (1:M with `Employee`).

- Participation:

- All employees must have a leave record, so participation is total on `employee\_id` (M:1).

10. Document:

- Attributes:

- document\_id (Primary Key)

- document\_name

- employee\_id (Foreign Key)

- uploaded\_date

- Relationships:

- Each document is associated with one employee (1:M with `Employee`).

- Participation:

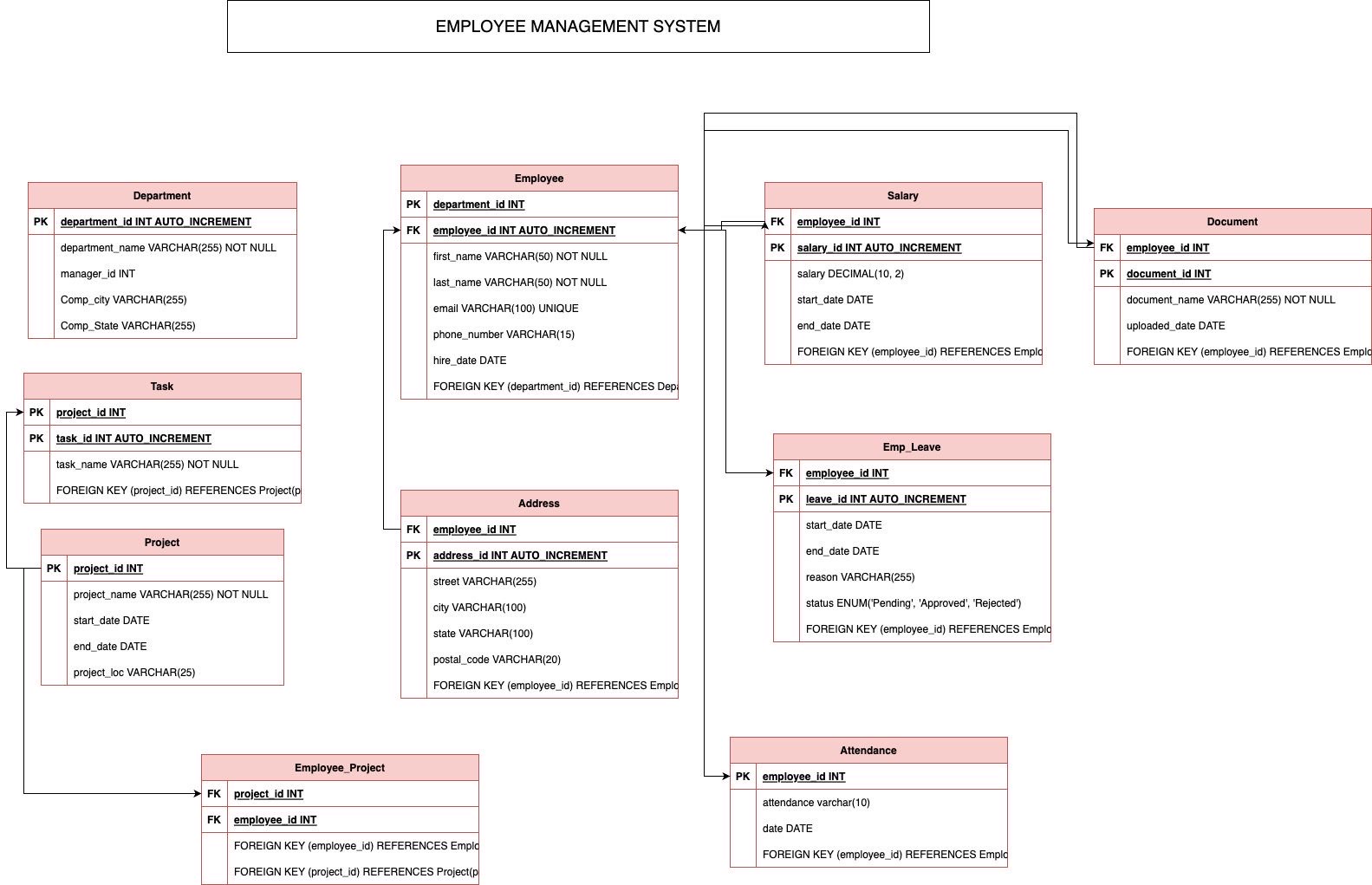
- All employees must have at least one document, so participation is total on `employee\_id` (M:1).

1. **Entity Relationship Diagram**

**A diagram of a company

Description automatically generated**

1. **Relational Diagram**

****

1. **SQL Queries (ADDED AFTER CONCLUSION FOR CLARITY AND NEATNESS)**
2. **Project Demonstration(ADDED AFTER CONCLUSION FOR CLARITY AND NEATNESS**
3. **Self-Learning Beyond Classroom**

The project greatly enhanced our team's proficiency in SQL, especially in querying and analyzing databases, which is crucial for managing employee information effectively. We learned to navigate complex datasets, create efficient queries, and extract valuable insights to inform decision-making related to employee management.

Additionally, we gained hands-on experience in data cleaning and validation, ensuring that employee data was accurate and consistent. This involved identifying and correcting errors and anomalies within the dataset, which is essential for maintaining reliable employee records.

Engaging in exploratory data analysis and trend identification honed our analytical skills, allowing us to identify patterns, correlations, and outliers within employee data. This is valuable for understanding employee performance, identifying training needs, and making informed decisions about promotions or salary adjustments.

Furthermore, the project improved our problem-solving abilities, as we tackled real-world challenges in managing employee information. We learned to approach problems systematically, breaking them down into manageable components and devising effective solutions. This is essential for resolving issues related to employee data management, such as data entry errors or database performance optimization.

Collaborating on this project also improved our communication and collaboration skills, as we learned to effectively communicate complex technical concepts to non-technical stakeholders, such as HR managers or department heads. This is crucial for ensuring that the employee management system meets the needs of all users and stakeholders.

Overall, the project highlighted the transformative potential of data-driven decision-making in employee management, driving improvements in operational efficiency and employee performance.

1. **Learning From The Project**

* Understanding SQL Basics: Working on an employee management system database project allowed us to solidify our grasp of SQL fundamentals. We learned how to query employee data, perform calculations, and extract valuable information from the database..
* Data Refinement and Handling: Working with SQL in the project allowed us to efficiently refine and handle employee data. We were able to manipulate and analyze the data by applying SQL functions and operations to gain insights into employee performance and other relevant metrics.
* Optimization for Better Performance: As we tackled SQL queries, we realized the importance of optimizing them for faster performance. We learned techniques such as indexing, query restructuring, and optimizing execution plans to reduce query execution time and enhance the overall efficiency of the employee management system.
* Resolving Errors and Debugging: Dealing with errors in SQL queries was a valuable learning experience. We became adept at identifying and fixing common errors like syntax issues, logical errors, and data inconsistencies using debugging tools and techniques provided by SQL environments.

1. **Challenges Faced**

Understanding Database Concepts: Initially, newcomers to database management may find concepts like normalization and transactions overwhelming.

Database Design: An essential aspect of creating a robust database involves carefully designing entities, attributes, relationships, and constraints.

Time Effeciency: Managing project deadlines and allocating adequate time for planning, design, implementation, testing, and documentation are critical for project success.

Documentation: Thorough documentation of the database schema, data dictionary, system architecture, and user manual is necessary for system understanding and operational efficiency. Students should excel in creating clear and easily understandable documentation.

1. **Conclusion**

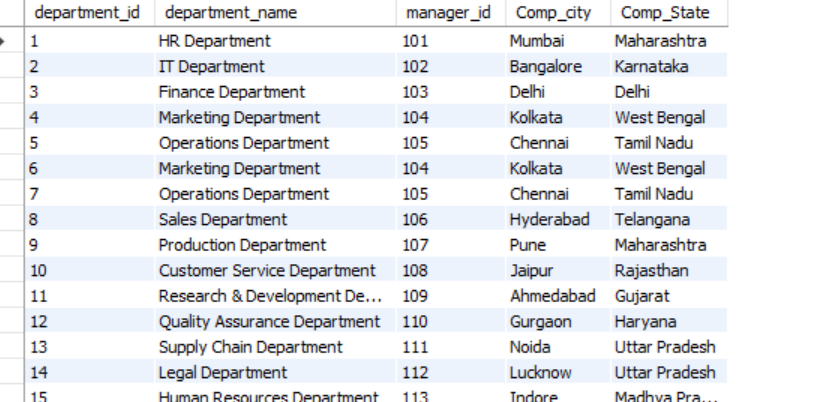
Completing this project has proven to be a complex yet exhilarating journey. While grappling with issues related to data quality and performance optimization posed significant challenges, we managed to deepen our understanding of SQL fundamentals and enhance our data analysis skills.

***PROJECT DEMO //TABLES //QUERIES***

A screenshot of a computer

Description automatically generated

**Department table**



**Salary table**

A screenshot of a data

Description automatically generated

**Address table**

A screenshot of a computer

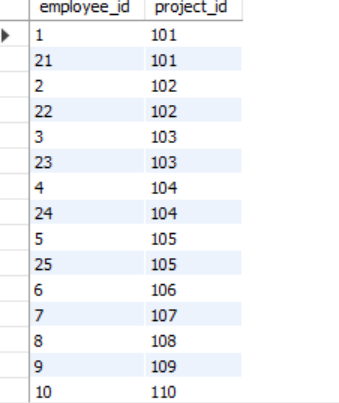
Description automatically generated

**Project table**

**A screenshot of a computer

Description automatically generated**

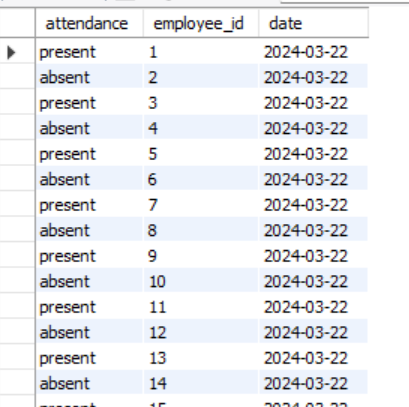
**Emp\_project table**

****

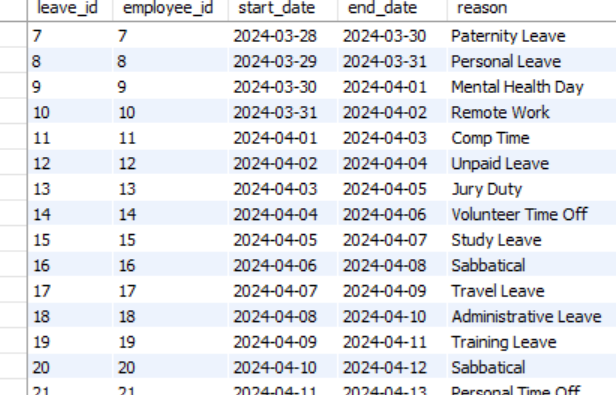
**Task table**

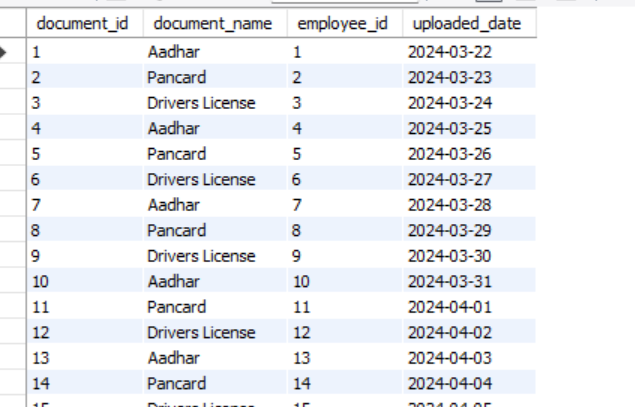
****

**Attendance table**

****

**Leave table**

****

**Document table**

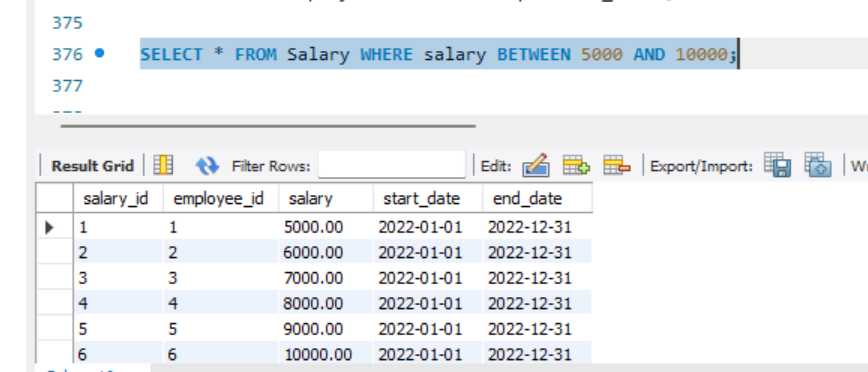
**20 select queries**

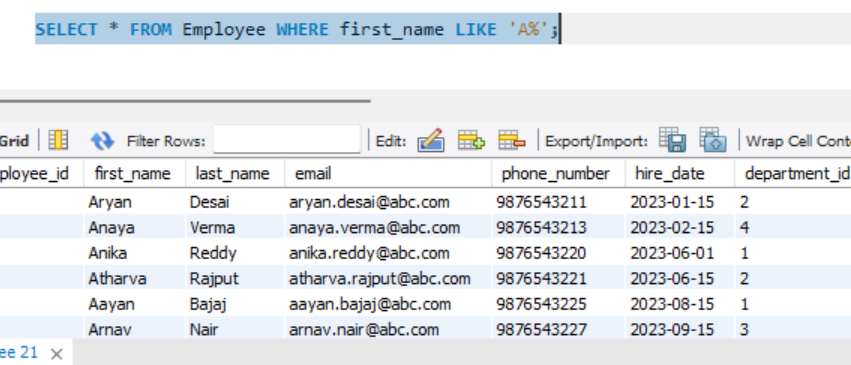
A screenshot of a computer

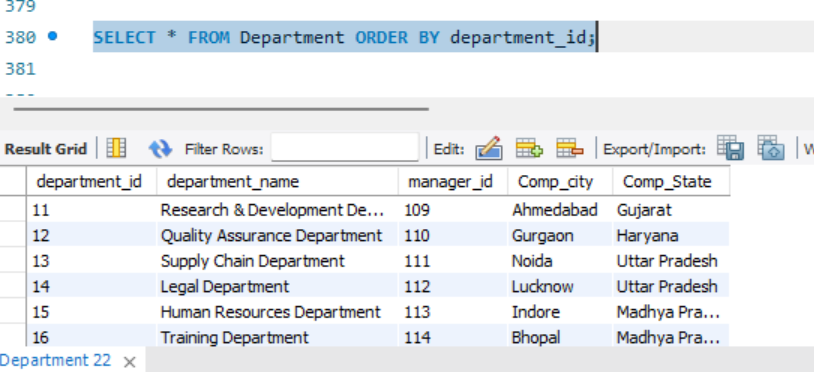
Description automatically generated

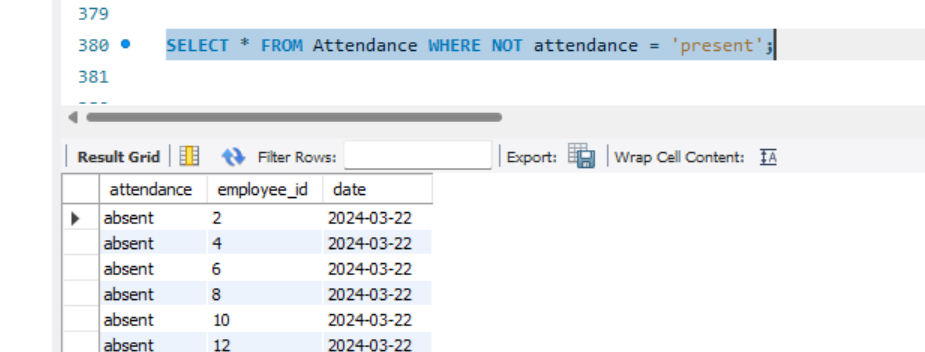
A screenshot of a computer

Description automatically generated









A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

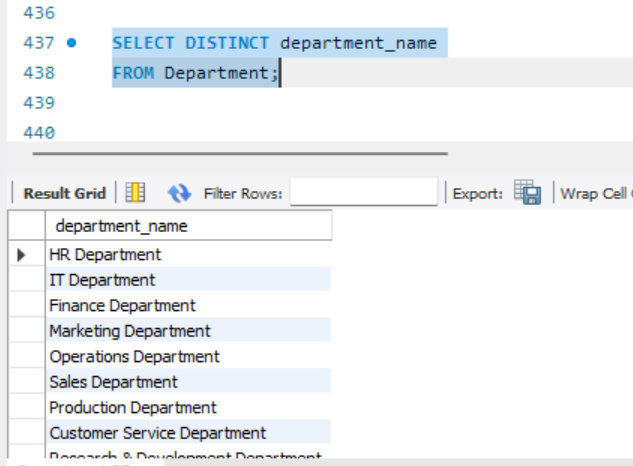
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated