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**
C2	C113	Vidhie Jhunjhunwala
C2	C118	Mithil Parekh
Date of Submission: 01/05/2024		

Contribution of each project Members:

Roll No.	Name:	Contribution
C113	Vidhie Jhunjhunwala	Creation of Schema, Report Formation, Select queries
C118	Mithil Parekh	ER Diagram,Relational Model, Debugging

GitHub link of your project:

https://github.com/vidhie1/Dbms-Project-Employee-Management

Project Report

Employee Management System

By

Vidhie Jhunjhunwala,C113

Mithil Parekh, C118

Course: Database Management System

AY: 2023-24

Table of Contents

I. 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..

II. 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).

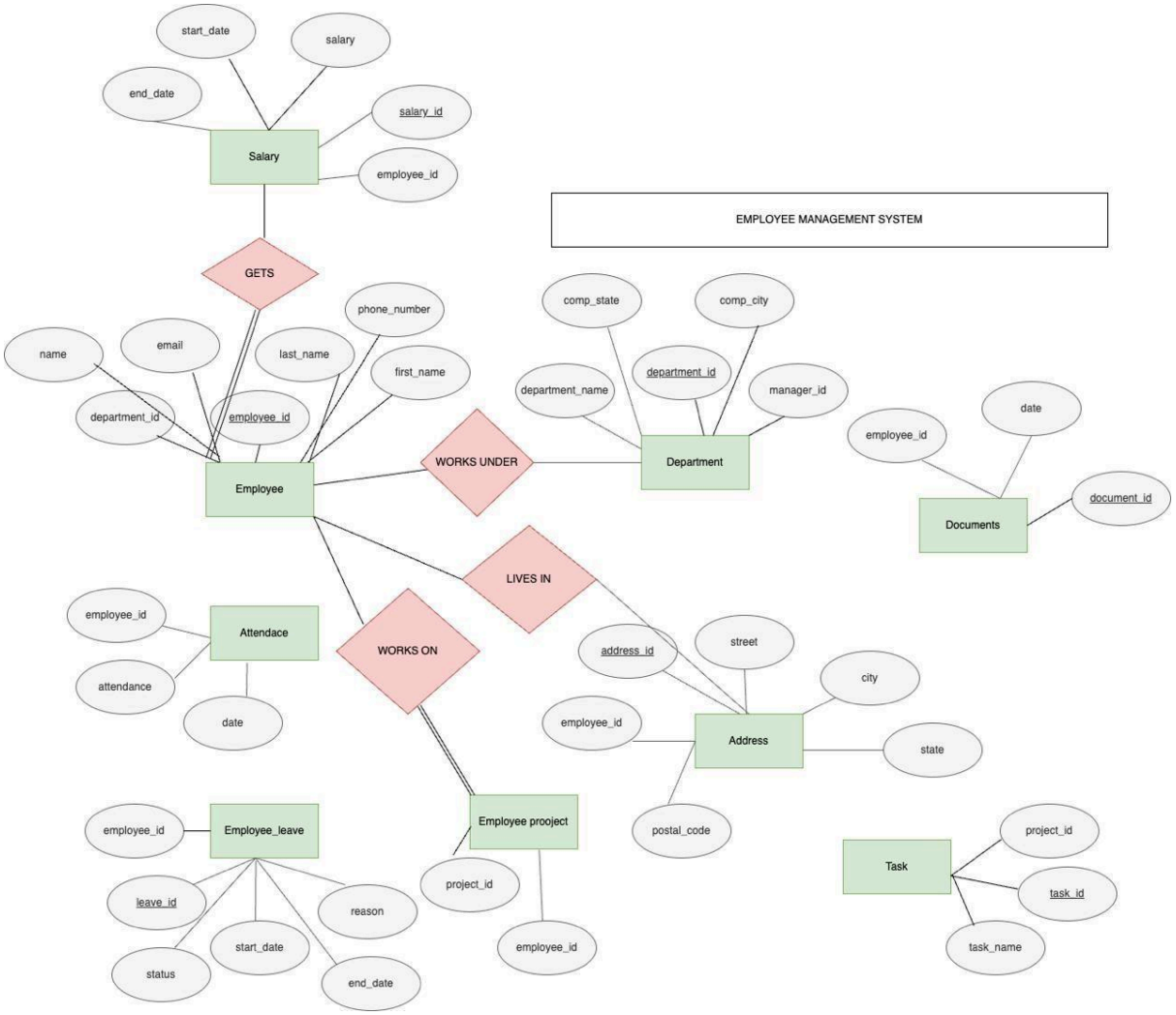
7. Task:
   - Attributes:
     - task_id (Primary Key)
     - task_name
     - project_id (Foreign Key)
   - Relationships:
     - Each task is associated with one project (1:M with `Project`).
   - Participation:
     - All tasks must be part of a project, so participation is total on `project_id` (M:1).

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).

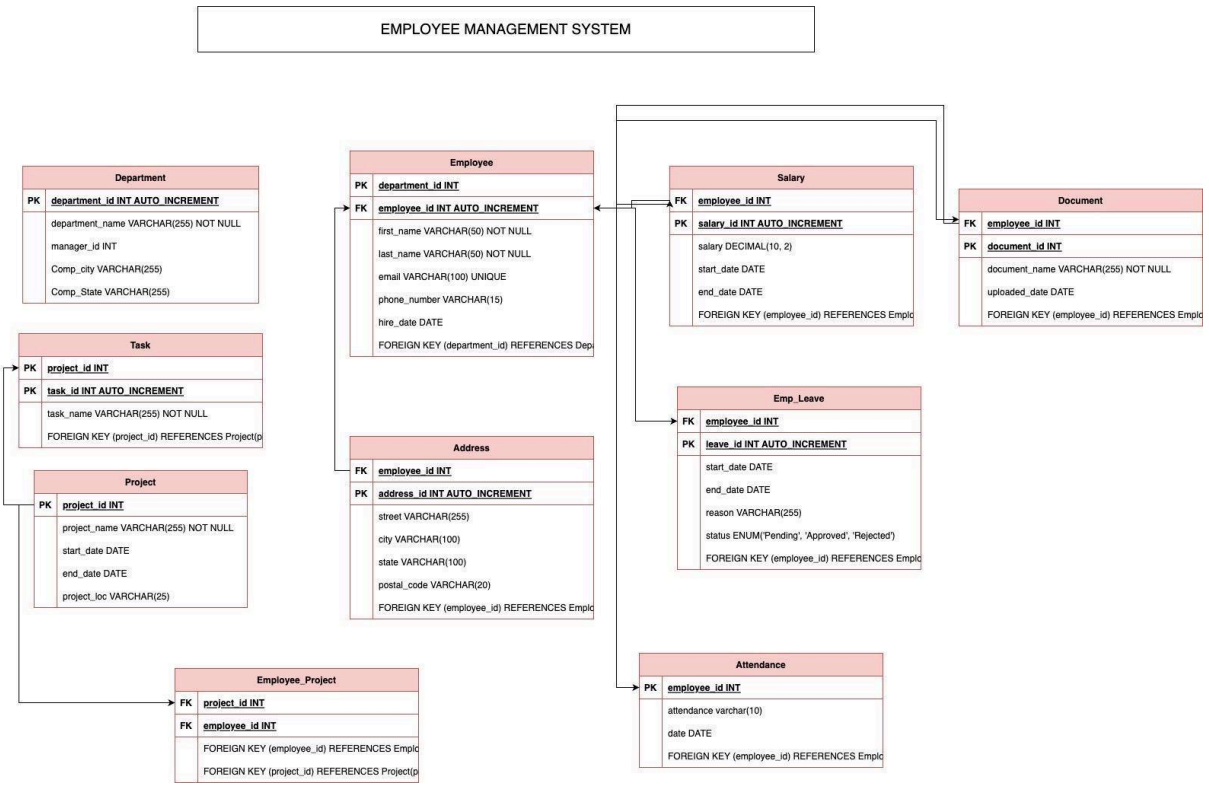
9. Emp_Leave:
   - Attributes:
     - leave_id (Primary Key)
     - employee_id (Foreign Key)
     - start_date
     - end_date
     - reason
     - status
   - Relationships:
     - Each leave request is associated with one employee (1:M with `Employee`).
   - Participation:
     - All employees must 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).

III. Entity Relationship Diagram



IV. Relational Diagram



Normalization:

First Normal Form (1NF):

- Ensure that each column contains atomic values. No multivalued attributes.
- Ensure each column has a unique name.
- Ensure each column holds only one value for each row.

Let's check if the tables are in 1NF:

- All tables seem to satisfy 1NF criteria. Each table has unique column names, and each cell contains atomic values.

Let's analyze each table for 2NF compliance:

Employee table:

- The Employee table has a primary key employee_id, and all non-key attributes (first_name, last_name, email, phone_number, hire_date, department_id) are functionally dependent on the primary key.

Department table:

- The Department table has a primary key department_id, and all non-key attributes (department_name, manager_id, Comp_city, Comp_State) are functionally dependent on the primary key.

Salary table:

- The Salary table has a composite primary key (employee_id, start_date), and all non-key attributes (salary, end_date) are functionally dependent on the composite primary key.

Address table:

- The Address table has a primary key address_id, and all non-key attributes (employee_id, street, city, state, postal_code) are functionally dependent on the primary key.

Project table:

- The Project table has a primary key project_id, and all non-key attributes (project_name, start_date, end_date, project_loc) are functionally dependent on the primary key.

Employee_Project table:

- The Employee_Project table has a composite primary key (employee_id, project_id), and there are no non-key attributes.

Task table:

- The Task table has a primary key task_id, and all non-key attributes (task_name, project_id) are functionally dependent on the primary key.

Attendance table:

- The Attendance table has a composite primary key (attendance, employee_id, date), and there are no non-key attributes.

Emp_Leave table:

- The Emp_Leave table has a primary key leave_id, and all non-key attributes (employee_id, start_date, end_date, reason) are functionally dependent on the primary key.

Document table:

- The Document table has a primary key document_id, and all non-key attributes (document_name, employee_id, uploaded_date) are functionally dependent on the primary key.

V. SQL Queries (ADDED AFTER CONCLUSION FOR CLARITY AND NEATNESS)

VI. Project Demonstration(ADDED AFTER CONCLUSION FOR CLARITY AND NEATNESS

VII. 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.

VIII. 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.

IX. 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.

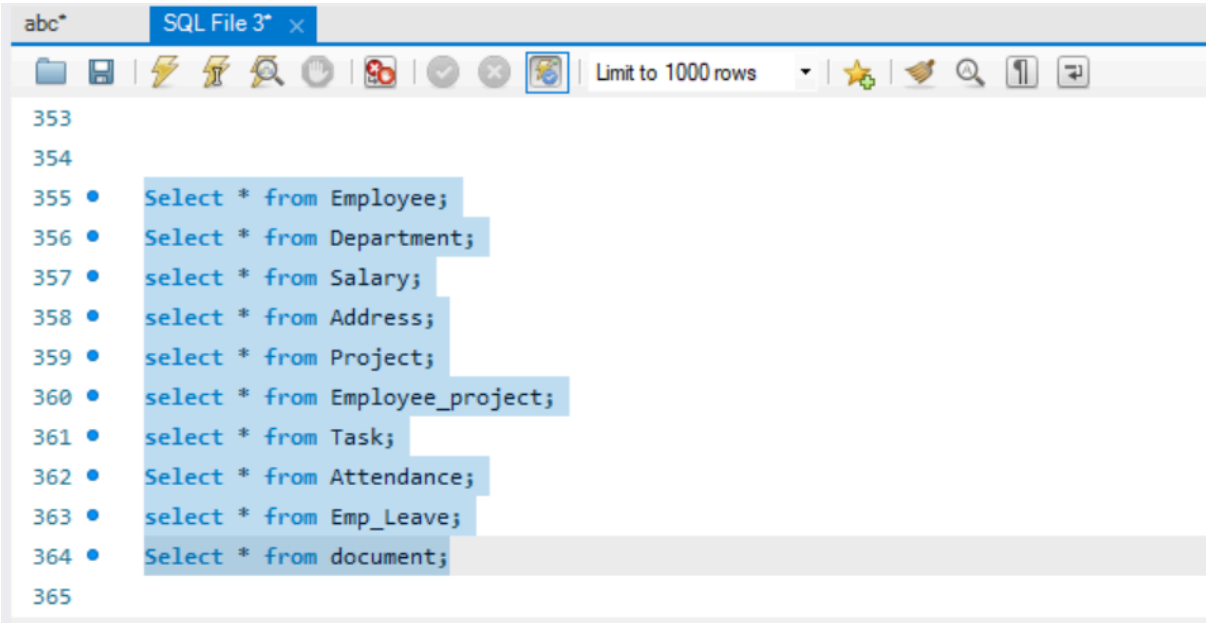
Time Effeciency: Managing project deadlines and allocating adequate time for planning, design, implementation, testing, and documentation are critical for project success.

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.

X. 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



Department table

department_id	department_name	manager_id	Comp_city	Comp_State
1	HR Department	101	Mumbai	Maharashtra
2	IT Department	102	Bangalore	Karnataka
3	Finance Department	103	Delhi	Delhi
4	Marketing Department	104	Kolkata	West Bengal
5	Operations Department	105	Chennai	Tamil Nadu
6	Marketing Department	104	Kolkata	West Bengal
7	Operations Department	105	Chennai	Tamil Nadu
8	Sales Department	106	Hyderabad	Telangana
9	Production Department	107	Pune	Maharashtra
10	Customer Service Department	108	Jaipur	Rajasthan
11	Research & Development De...	109	Ahmedabad	Gujarat
12	Quality Assurance Department	110	Gurgaon	Haryana
13	Supply Chain Department	111	Noida	Uttar Pradesh
14	Legal Department	112	Lucknow	Uttar Pradesh
15	Human Resources Department	113	Indore	Madhya Pra...

Salary table

salary_id	employee_id	salary	start_date	end_date
1	1	5000.00	2022-01-01	2022-12-31
2	2	6000.00	2022-01-01	2022-12-31
3	3	7000.00	2022-01-01	2022-12-31
4	4	8000.00	2022-01-01	2022-12-31
5	5	9000.00	2022-01-01	2022-12-31
6	6	10000.00	2022-01-01	2022-12-31
7	7	11000.00	2022-01-01	2022-12-31
8	8	12000.00	2022-01-01	2022-12-31
9	9	13000.00	2022-01-01	2022-12-31
10	10	14000.00	2022-01-01	2022-12-31
11	11	15000.00	2022-01-01	2022-12-31
12	12	16000.00	2022-01-01	2022-12-31
13	13	17000.00	2022-01-01	2022-12-31
14	14	18000.00	2022-01-01	2022-12-31
15	15	19000.00	2022-01-01	2022-12-31

Address table

address_id	employee_id	street	city	state	postal_code
1	1	Chhatrapati Shivaji Marg	Mumbai	Maharashtra	12345
2	2	Sane Guruji Marg	Bangalore	Karnataka	67890
3	3	Bombay Samachar Marg	Delhi	Delhi	54321
4	4	Jawaharlal Nehru Road	Kolkata	West Bengal	98765
5	5	MG Road	Chennai	Tamil Nadu	23456
6	6	Sector 17	Chandigarh	Punjab	87654
7	7	Devaraja Urs Road	Mysore	Karnataka	34567
8	8	Cross Cut Road	Coimbatore	Tamil Nadu	76543
9	9	JLN Road	Jaipur	Rajasthan	45678
10	10	Banjara Hills	Hyderabad	Telangana	65432
11	11	Ashram Road	Ahmedabad	Gujarat	98765
12	12	FC Road	Pune	Maharashtra	23456
13	13	Hazratganj	Lucknow	Uttar Pradesh	87654
14	14	Wardha Road	Nagpur	Maharashtra	34567
15	15	RS Puram	Coimbatore	Tamil Nadu	76543

Project table

project_id	project_name	start_date	end_date	project_loc
101	Project A	2024-01-01	2024-06-30	Location A
102	Project B	2024-02-15	2024-12-31	Location B
103	Project C	2024-03-10	2024-09-30	Location C
104	Project D	2024-04-05	2024-11-30	Location D
105	Project E	2024-05-20	2024-10-31	Location E
106	Project F	2024-06-15	2024-12-15	Location F
107	Project G	2024-07-01	2024-12-31	Location G
108	Project H	2024-08-10	2024-11-30	Location H
109	Project I	2024-09-05	2024-10-31	Location I
110	Project J	2024-10-01	2024-12-31	Location J
111	Project K	2024-11-15	2024-12-15	Location K
112	Project L	2024-12-01	2024-12-31	Location L
113	Project M	2024-01-15	2024-06-30	Location M
114	Project N	2024-02-20	2024-12-31	Location N
115	Project O	2024-03-05	2024-09-30	Location O

Emp_project table

employee_id	project_id
1	101
21	101
2	102
22	102
3	103
23	103
4	104
24	104
5	105
25	105
6	106
7	107
8	108
9	109
10	110

Task table

task_id	task_name	project_id
1	Research market trends	101
2	Design user interface	103
3	Write test cases	102
4	Optimize database performance	104
5	Conduct user testing	107
6	Coordinate project meetings	109
7	Implement security measures	111
8	Perform code review	113
9	Prepare project presentation	115
10	Integrate third-party APIs	117
11	Resolve bug reports	119
12	Create data visualizations	120
13	Manage project timeline	110
14	Develop backend functionality	103
15	Write test cases	105

Attendance table

attendance	employee_id	date
present	1	2024-03-22
absent	2	2024-03-22
present	3	2024-03-22
absent	4	2024-03-22
present	5	2024-03-22
absent	6	2024-03-22
present	7	2024-03-22
absent	8	2024-03-22
present	9	2024-03-22
absent	10	2024-03-22
present	11	2024-03-22
absent	12	2024-03-22
present	13	2024-03-22
absent	14	2024-03-22

Leave table

leave_id	employee_id	start_date	end_date	reason
7	7	2024-03-28	2024-03-30	Paternity Leave
8	8	2024-03-29	2024-03-31	Personal Leave
9	9	2024-03-30	2024-04-01	Mental Health Day
10	10	2024-03-31	2024-04-02	Remote Work
11	11	2024-04-01	2024-04-03	Comp Time
12	12	2024-04-02	2024-04-04	Unpaid Leave
13	13	2024-04-03	2024-04-05	Jury Duty
14	14	2024-04-04	2024-04-06	Volunteer Time Off
15	15	2024-04-05	2024-04-07	Study Leave
16	16	2024-04-06	2024-04-08	Sabbatical
17	17	2024-04-07	2024-04-09	Travel Leave
18	18	2024-04-08	2024-04-10	Administrative Leave
19	19	2024-04-09	2024-04-11	Training Leave
20	20	2024-04-10	2024-04-12	Sabbatical
21	21	2024-04-11	2024-04-13	Personal Time Off

Document table

document_id	document_name	employee_id	uploaded_date
1	Aadhar	1	2024-03-22
2	Pancard	2	2024-03-23
3	Drivers License	3	2024-03-24
4	Aadhar	4	2024-03-25
5	Pancard	5	2024-03-26
6	Drivers License	6	2024-03-27
7	Aadhar	7	2024-03-28
8	Pancard	8	2024-03-29
9	Drivers License	9	2024-03-30
10	Aadhar	10	2024-03-31
11	Pancard	11	2024-04-01
12	Drivers License	12	2024-04-02
13	Aadhar	13	2024-04-03
14	Pancard	14	2024-04-04

20 select queries

```
SELECT * FROM Employee WHERE department_id = 2 AND hire_date > '2023-01-01';
```

employee_id	first_name	last_name	email	phone_number	hire_date	departmer
2	Aryan	Desai	aryan.desai@abc.com	9876543211	2023-01-15	2
7	Saisha	Kumar	saisha.kumar@abc.com	9876543216	2023-04-01	2
12	Atharva	Rajput	atharva.rajput@abc.com	9876543221	2023-06-15	2
17	Kiara	Dube	kiara.dube@abc.com	9876543226	2023-09-01	2
22	Rohan	Sharma	rohan.sharma@abc.com	9876543231	2023-11-15	2

```
381 •    SELECT * FROM Attendance WHERE NOT attendance = 'present';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

project_id	project_name	start_date	end_date	project_loc
101	Project A	2024-01-01	2024-06-30	Location A
102	Project B	2024-02-15	2024-12-31	Location B

```
375
376 •  SELECT * FROM Salary WHERE salary BETWEEN 5000 AND 10000;
377
```

Result Grid | Filter Rows: | Edit: | Export/Import:

salary_id	employee_id	salary	start_date	end_date
1	1	5000.00	2022-01-01	2022-12-31
2	2	6000.00	2022-01-01	2022-12-31
3	3	7000.00	2022-01-01	2022-12-31
4	4	8000.00	2022-01-01	2022-12-31
5	5	9000.00	2022-01-01	2022-12-31
6	6	10000.00	2022-01-01	2022-12-31

```
SELECT * FROM Employee WHERE first_name LIKE 'A%';
```

Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

ployee_id	first_name	last_name	email	phone_number	hire_date	department_id
	Aryan	Desai	aryan.desai@abc.com	9876543211	2023-01-15	2
	Anaya	Verma	anaya.verma@abc.com	9876543213	2023-02-15	4
	Anika	Reddy	anika.reddy@abc.com	9876543220	2023-06-01	1
	Atharva	Rajput	atharva.rajput@abc.com	9876543221	2023-06-15	2
	Aayan	Bajaj	aayan.bajaj@abc.com	9876543225	2023-08-15	1
	Arnav	Nair	arnav.nair@abc.com	9876543227	2023-09-15	3

ee 21 ×

```
379
380 •  SELECT * FROM Department ORDER BY department_id;
381
```

Result Grid | Filter Rows: | Edit: | Export/Import:

department_id	department_name	manager_id	Comp_city	Comp_State
11	Research & Development De...	109	Ahmedabad	Gujarat
12	Quality Assurance Department	110	Gurgaon	Haryana
13	Supply Chain Department	111	Noida	Uttar Pradesh
14	Legal Department	112	Lucknow	Uttar Pradesh
15	Human Resources Department	113	Indore	Madhya Pra...
16	Training Department	114	Bhopal	Madhya Pra...

Department 22 ×

```
379
380 •  SELECT * FROM Attendance WHERE NOT attendance = 'present';
381
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

attendance	employee_id	date
absent	2	2024-03-22
absent	4	2024-03-22
absent	6	2024-03-22
absent	8	2024-03-22
absent	10	2024-03-22
absent	12	2024-03-22

```
386 •  SELECT *
387    FROM Employee
388    LEFT OUTER JOIN Address ON Employee.employee_id = Address.employee_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	hire_date	department_id	address_id	employ
1	Keya	Shah	keya.shah@abc.com	9876543210	2023-01-01	1	1	1
2	Aryan	Desai	aryan.desai@abc.com	9876543211	2023-01-15	2	2	2
3	Demira	Ramnani	demira.ramnani@abc.com	9876543212	2023-02-01	3	3	3
4	Anaya	Verma	anaya.verma@abc.com	9876543213	2023-02-15	4	4	4
5	Ishaan	Gupta	ishaan.gupta@abc.com	9876543214	2023-03-01	5	5	5
6	Vihaan	Joshi	vihaan.joshi@abc.com	9876543215	2023-03-15	1	6	6
7	Saisha	Kumar	saisha.kumar@abc.com	9876543216	2023-04-01	2	7	7

```
390
391 •  SELECT *
392    FROM Employee
393    NATURAL JOIN Department;
394
395
396
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_id	employee_id	first_name	last_name	email	phone_number	hire_date	department_name
1	1	Keya	Shah	keya.shah@abc.com	9876543210	2023-01-01	HR Department
2	2	Aryan	Desai	aryan.desai@abc.com	9876543211	2023-01-15	IT Department
3	3	Demira	Ramnani	demira.ramnani@abc.com	9876543212	2023-02-01	Finance Department
4	4	Anaya	Verma	anaya.verma@abc.com	9876543213	2023-02-15	Marketing Department
5	5	Ishaan	Gupta	ishaan.gupta@abc.com	9876543214	2023-03-01	Operations Departmer
1	6	Vihaan	Joshi	vihaan.joshi@abc.com	9876543215	2023-03-15	HR Department
2	7	Saisha	Kumar	saisha.kumar@abc.com	9876543216	2023-04-01	IT Department

Result Grid | Form Editor

Result 26 ×

Read Only

```
394
395 • SELECT Employee.employee_id, Employee.first_name, Employee.last_name, Attendance.attendance, Attendance.
396   FROM Employee
397   RIGHT OUTER JOIN Attendance ON Employee.employee_id = Attendance.employee_id;
398
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	first_name	last_name	attendance	date
1	Keya	Shah	present	2024-03-22
2	Aryan	Desai	absent	2024-03-22
3	Demira	Ramnani	present	2024-03-22
4	Anaya	Verma	absent	2024-03-22
5	Ishaan	Gupta	present	2024-03-22
6	Vihaan	Joshi	absent	2024-03-22
7	Saisha	Kumar	present	2024-03-22
8	Reyansh	Shah	absent	2024-03-22

Result 27

```
413
414 • SELECT MAX(project_id) AS max_project_id
415   FROM Project;
416
417
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

max_project_id
120

```
398
399 • SELECT *
400   FROM Department
401   ORDER BY manager_id ASC;
402
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

department_id	department_name	manager_id	Comp_city	Comp_State
1	HR Department	101	Mumbai	Maharashtra
2	IT Department	102	Bangalore	Karnataka
3	Finance Department	103	Delhi	Delhi
4	Marketing Department	104	Kolkata	West Bengal
6	Marketing Department	104	Kolkata	West Bengal
5	Operations Department	105	Chennai	Tamil Nadu
7	Operations Department	105	Chennai	Tamil Nadu
8	Sales Department	106	Hyderabad	Telangana

Department 28

Output

```
402
403 •  SELECT *
404    FROM Salary
405    ORDER BY salary DESC;
406
```

Result Grid | Filter Rows: | Edit: | Export/Imp

salary_id	employee_id	salary	start_date	end_date
25	25	29000.00	2022-01-01	2022-12-31
24	24	28000.00	2022-01-01	2022-12-31
23	23	27000.00	2022-01-01	2022-12-31
22	22	26000.00	2022-01-01	2022-12-31
21	21	25000.00	2022-01-01	2022-12-31
20	20	24000.00	2022-01-01	2022-12-31
19	19	23000.00	2022-01-01	2022-12-31
18	18	22000.00	2022-01-01	2022-12-31

Salary 29 ×

```
406
407 •  SELECT COUNT(*) AS total_employees
408    FROM Employee;
409
410
```

Result Grid | Filter Rows: | Export:

total_employees
25

```
409
410 ●  SELECT AVG(manager_id) AS avg_manager_id
411    FROM Department;
412
413
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_manager_id
109.0000

```
415    FROM Project;
416
417 ●  SELECT employee_id, COUNT(*) AS total_present
418    FROM Attendance
419    WHERE attendance = 'present'
420    GROUP BY employee_id;
421
422
423
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	total_present
1	1
3	1
5	1
7	1
9	1
11	1
13	1
15	1

```
421
422
423 ●  SELECT department_id, AVG(manager_id) AS avg_manager_id
424     FROM Department
425     GROUP BY department_id
426     HAVING avg_manager_id > 105;
427
428
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

department_id	avg_manager_id
8	106.0000
9	107.0000
10	108.0000
11	109.0000
12	110.0000
13	111.0000
14	112.0000
15	113.0000

Result 34

```
429
430
431 ●  SELECT e.employee_id, e.first_name, e.last_name, e.email, e.phone_number, d.department_name, s.salary
432     FROM Employee e
433     INNER JOIN Department d ON e.department_id = d.department_id
434     INNER JOIN Salary s ON e.employee_id = s.employee_id
435     WHERE s.salary > 10000;
436
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	department_name	salary
7	Saisha	Kumar	saisha.kumar@abc.com	9876543216	IT Department	11000.00
8	Reyansh	Shah	reyansh.shah@abc.com	9876543217	Finance Department	12000.00
9	Prisha	Singhania	prisha.singhania@abc.com	9876543218	Marketing Department	13000.00
10	Kabir	Mishra	kabir.mishra@abc.com	9876543219	Operations Department	14000.00
11	Anika	Reddy	anika.reddy@abc.com	9876543220	HR Department	15000.00
12	Atharva	Rajput	atharva.rajput@abc.com	9876543221	IT Department	16000.00
13	Myra	Malhotra	myra.malhotra@abc.com	9876543222	Finance Department	17000.00
14	Kian	Gupta	kian.gupta@abc.com	9876543223	Marketing Department	18000.00

Result 36 | Read Only

Output

```
436
437 •   SELECT DISTINCT department_name
438     FROM Department;
439
440
```

Result Grid | Filter Rows: | Export: | Wrap Cell

department_name
HR Department
IT Department
Finance Department
Marketing Department
Operations Department
Sales Department
Production Department
Customer Service Department
Research & Development Department

```
439
440 •  SELECT SUM(salary) AS total_salary_expenditure
441    FROM Salary;
442
443
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_salary_expenditure
425000.00

Result 39

```
439
440 •  SELECT e.employee_id, e.first_name, e.last_name, s.salary
441    FROM Employee e
442    INNER JOIN Salary s ON e.employee_id = s.employee_id
443    ORDER BY s.salary DESC
444    LIMIT 5;
445
446
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	first_name	last_name	salary
25	Aarav	Gandhi	29000.00
24	Yash	Thakur	28000.00
23	Ishita	Choudhary	27000.00
22	Rohan	Sharma	26000.00
21	Anika	Patel	25000.00

Limit to 1000 rows

```
460
461    -- Creating a view to combine Employee, Department, and Salary information
462 •  CREATE VIEW EmployeeDetail AS
463    SELECT e.employee_id, e.first_name, e.last_name, e.email, e.phone_number,
464           d.department_name, d.Comp_city, d.Comp_State,
465           s.salary
466    FROM Employee e
467    INNER JOIN Department d ON e.department_id = d.department_id
468    INNER JOIN Salary s ON e.employee_id = s.employee_id;
469
```

```
469
470     -- Example query using ORDER BY to sort employees by salary in descending order
471     SELECT *
472     FROM EmployeeDetails
473     ORDER BY salary DESC;
474
475
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	first_name	last_name	email	phone_number	department_name	Comp_city	C
25	Aarav	Gandhi	aarav.gandhi@abc.com	9876543234	Operations Department	Chennai	Ta
24	Yash	Thakur	yash.thakur@abc.com	9876543233	Marketing Department	Kolkata	We
23	Ishita	Choudhary	ishita.choudhary@abc.com	9876543232	Finance Department	Delhi	De
22	Rohan	Sharma	rohan.sharma@abc.com	9876543231	IT Department	Bangalore	Ka
21	Anika	Patel	anika.patel@abc.com	9876543230	HR Department	Mumbai	Ma
20	Rudra	Sharma	rudra.sharma@abc.com	9876543229	Operations Department	Chennai	Ta
19	Neha	Sachdeva	neha.sachdeva@abc.com	9876543228	Marketing Department	Kolkata	We
18	Arnav	Nair	arnav.nair@abc.com	9876543227	Finance Department	Delhi	De

Result Grid

Form Editor