**DBMS Project**

**Employee Payroll Management System**

**Project Background:**

In today's dynamic business environment, efficient management of human resources is crucial for organizational success. As businesses grow and expand, so does the complexity of managing employee-related tasks such as payroll processing, attendance tracking, and leave management. Recognizing the need for a streamlined and automated solution, we have embarked on the development of an Employee Payroll Management System.

**Description of the Project:**

The Employee Payroll Management System is a comprehensive software solution aimed at simplifying and optimizing the management of employee-related processes within our organization. It is designed to centralize all employee data and streamline various HR and payroll tasks, ultimately enhancing productivity, accuracy, and compliance.

**Key Features and Functionality:**

**Employee Information Management:**

The system will maintain a centralized database of employee information, including personal details, contact information, employment history, and educational qualifications. It will allow HR administrators to easily update and retrieve employee records, ensuring data accuracy and completeness.

**Payroll Processing:**

Automated calculation of employee salaries based on predefined parameters such as gross salary, hourly pay rates, and tax deductions. Generation of pay slips and statements for each pay period, providing detailed breakdowns of earnings, deductions, and net pay.

**Attendance Tracking:**

Recording and monitoring of employee attendance, including work hours, overtime, absences, and leaves. Integration with biometric devices or time clocks to accurately capture attendance data.

**Leave Management:**

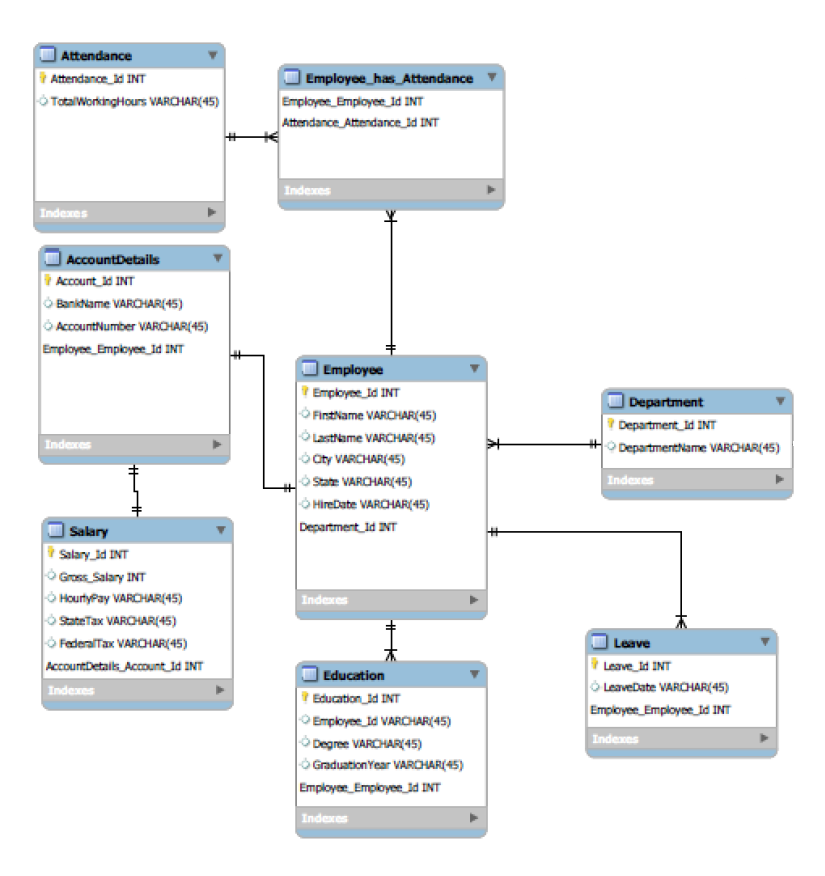
Employees can submit leave requests through the system, specifying the type of leave and duration. Managers can review, approve, or reject leave requests based on organizational policies and staffing requirements.

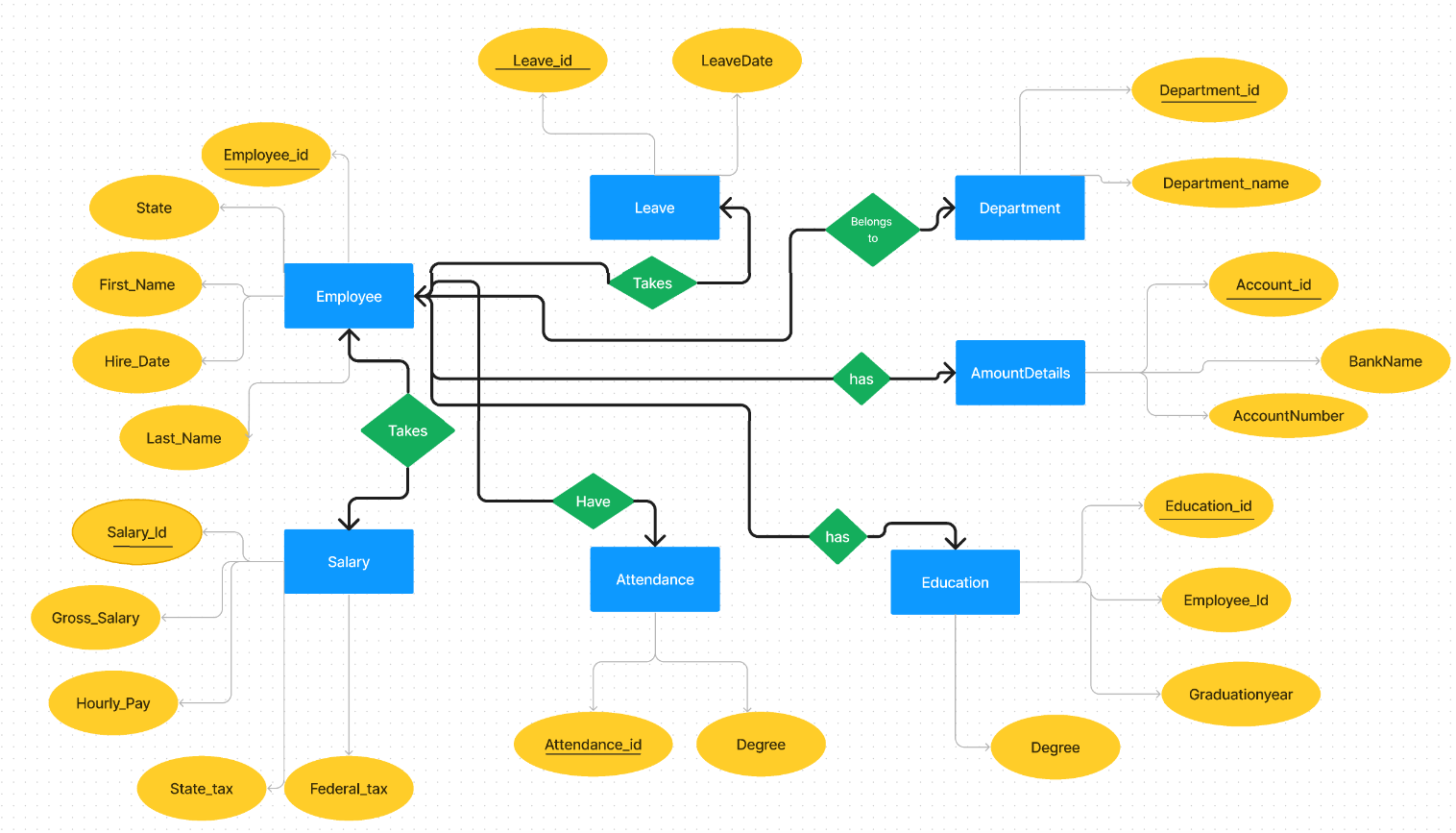
**Tax Compliance:**

Automatic calculation and deduction of taxes from employee salaries in accordance with tax laws and regulations. Generation of tax reports and filings to ensure compliance with government requirements.

**Reporting and Analytics:**

Comprehensive reporting capabilities to generate HR and payroll reports, including payroll summaries, attendance trends, leave balances, and tax deductions. Utilization of analytics tools to gain insights into employee performance and payroll expenses for informed decision-making

**Schema Diagram:-**

**E-R Diagram:-**

**Details Of E-R Diagram : -**

**Entities:**

**Employee:** Stores personal details of employees, including Employee\_Id, First\_Name, Last\_Name, Hire\_Date, City, and State.

Department: Manages information about different departments within the organization, including Department\_Id and Department\_Name.

**Salary:** Contains salary-related information such as Gross\_Salary, Hourly\_Pay, State\_Tax, Federal\_Tax, and Account\_Id.

**AccountDetails:** Maintains details of bank accounts linked to employees for salary payments, including Account\_Id, Bank\_Name, Account\_Number, and Employee\_Id.

**Education:** Tracks educational qualifications and degrees obtained by employees, including Education\_Id, Employee\_Id, Degree, and Graduation\_Year.

**Leave:** Records leave taken by employees, including Leave\_Id, Employee\_Id, and Leave\_date.

**Attendance:** Stores data related to employee attendance, including Attendance\_Id and Hours\_Worked.

**Relationships:**

**Employee-Salary:** Represents the relationship between employees and their salary information. It's a one-to-one relationship, meaning each employee has one salary record, and each salary record is associated with only one employee.

**Employee-AccountDetails:** Represents the relationship between employees and their bank account details. Again, it's a one-to-one relationship, indicating that each employee has one account for salary payment, and each account is associated with only one employee.

**Employee-Education:** Indicates the relationship between employees and their educational qualifications. It's a one-to-many relationship, meaning each employee can have multiple educational qualifications, but each qualification is associated with only one employee.

**Employee-Leave:** Represents the relationship between employees and their leave records. It's a one-to-many relationship, indicating that each employee can take multiple leaves, but each leave record is associated with only one employee.

**Employee-Attendance:** Indicates the relationship between employees and their attendance records. It's a one-to-many relationship, meaning each employee can have multiple attendance records, but each attendance record is associated with only one employee.

**Employee-Department Relationship:**

Each employee belongs to one department, while each department can have multiple employees. This represents a one-to-many relationship between the "Employee" and "Department" entities.

The ER diagram provides a visual representation of how these entities are related within the organization's database. Each entity is represented by a rectangle, with its attributes listed inside. Relationships between entities are depicted by lines connecting them, with cardinality (one-to-one, one-to-many) indicated on each end of the line.

**Conversion of ER diagram into Tables :-**

**Employee**

CREATE TABLE Employee (

Employee\_Id INT(6),

First\_Name VARCHAR(25),

Last\_Name VARCHAR(25),

Hire\_Date DATE,

City VARCHAR(25),

State VARCHAR(25),

Department\_id int,

FOREIGN KEY (Department\_id) REFERENCES Department(Department\_Id)

CONSTRAINT EMPLOYEE\_PK PRIMARY KEY (Employee\_Id)

);

**Department**

CREATE TABLE Department (

Department\_Id INT,

Department\_Name VARCHAR(30),

CONSTRAINT DEPARTMENT\_PK PRIMARY KEY (Department\_Id)

);

**Salary**

CREATE TABLE Salary (

Salary\_Id int,

Gross\_Salary int,

Hourly\_Pay int,

State\_Tax int,

Federal\_Tax int,

Account\_Id int,

CONSTRAINT SALARY\_PK PRIMARY KEY (Salary\_Id),

FOREIGN KEY (Account\_Id) REFERENCES AccountDetails(Account\_Id)

);

**AccountDetails**

CREATE TABLE AccountDetails (

Account\_Id int,

Bank\_Name VARCHAR(50),

Account\_Number VARCHAR(50),

Employee\_Id int,

CONSTRAINT Account\_PK PRIMARY KEY (Account\_Id),

FOREIGN KEY (Employee\_Id) REFERENCES Employee(Employee\_Id)

);

**Education**

CREATE TABLE Education (

Education\_Id int,

Employee\_Id int,

Degree VARCHAR(30),

Graduation\_Year int(4),

CONSTRAINT Education\_PK PRIMARY KEY (Education\_Id),

FOREIGN KEY (Employee\_Id) REFERENCES Employee(Employee\_Id)

);

**Leave**

CREATE TABLE Leave\_ (

Leave\_Id int,

Employee\_Id int,

Leave\_date DATE,

CONSTRAINT Leave\_PK PRIMARY KEY (Leave\_Id),

FOREIGN KEY (Employee\_Id) REFERENCES Employee(Employee\_Id)

);

**Employee\_Attendance**

CREATE TABLE Employee\_Attendance (

Employee\_Id int,

Attendance\_Id int,

CONSTRAINT EMPLOYEEATTENDANCE\_PK PRIMARY KEY (Employee\_Id, Attendance\_Id),

FOREIGN KEY (Employee\_Id) REFERENCES Employee(Employee\_Id),

FOREIGN KEY (Attendance\_Id) REFERENCES Attendance(Attendance\_Id)

);

**Attendance**

CREATE TABLE Attendance (

Attendance\_Id int,

Hours\_Worked int,

CONSTRAINT Attendance\_PK PRIMARY KEY (Attendance\_Id)

);

**Description of Tables**

**Employee:**

1. Contains information about employees such as their ID, first name, last name, hire date, city, state, and the department they belong to.
2. The Employee\_Id is the primary key, and there's a foreign key constraint referencing the Department table.

**Department:**

1. Stores details about different departments within the organization.
2. Includes a Department\_Id as the primary key.

**Salary:**

1. Holds salary-related information like gross salary, hourly pay, state tax, federal tax, and the associated account details.
2. The Salary\_Id is the primary key, and there's a foreign key referencing AccountDetails.

**AccountDetails:**

1. Contains bank account details of employees including the account ID, bank name, account number, and the associated employee.
2. The Account\_Id is the primary key, and there's a foreign key referencing Employee.

**Education:**

1. Stores educational information of employees, such as their degree and graduation year.
2. Each entry is linked to an employee through the Employee\_Id.
3. The Education\_Id is the primary key.

**Leave:**

1. Records leave-related data including the leave ID, employee ID, and leave date.
2. The Leave\_Id is the primary key, and there's a foreign key referencing Employee.

**Employee\_Attendance:**

1. Keeps track of employee attendance.
2. Utilizes a composite primary key consisting of Employee\_Id and Attendance\_Id.
3. The Employee\_Id is a foreign key referencing Employee, and Attendance\_Id is a foreign key referencing Attendance.

**Attendance:**

1. Contains attendance-related information such as hours worked.
2. The Attendance\_Id is the primary key.

These tables are interconnected through foreign key relationships, allowing for the storage of comprehensive employee data, including personal details, salary, education, attendance, and leave records.

**Normalization of tables up to 3-NF:-**

**Third Normal Form (3NF):**

Third Normal Form (3NF) is a level of database normalization designed to eliminate redundancy and ensure that data is logically organized. To achieve 3NF, a table must first meet the requirements of First Normal Form (1NF) and Second Normal Form (2NF).

**In 3NF:**

1. All data in a table should depend only on the table's primary key, and there should be no transitive dependencies.
2. Every non-key attribute must be functionally dependent on the primary key directly, not transitively through another non-key attribute.

**Analysis of Provided Tables in 3NF:**

**Employee Table:**

1. Employee\_Id is the primary key.
2. First\_Name, Last\_Name, Hire\_Date, City, State are functionally dependent on Employee\_Id.
3. Department\_Id is a foreign key referencing Department table, and Department\_Id is the primary key in Department table.

The table satisfies 3NF requirements.

**Department Table:**

1. Department\_Id is the primary key.
2. Department\_Name is fully functionally dependent on the primary key.

The table satisfies 3NF requirements.

**Salary Table:**

1. Salary\_Id is the primary key.
2. Gross\_Salary, Hourly\_Pay, State\_Tax, Federal\_Tax are functionally dependent on Salary\_Id.
3. Account\_Id is a foreign key referencing AccountDetails table.

The table satisfies 3NF requirements.

**AccountDetails Table:**

1. Account\_Id is the primary key.
2. Bank\_Name, Account\_Number are functionally dependent on Account\_Id.
3. Employee\_Id is a foreign key referencing Employee table.

The table satisfies 3NF requirements.

**Education Table:**

1. Education\_Id is the primary key.
2. Degree, Graduation\_Year are functionally dependent on Education\_Id.
3. Employee\_Id is a foreign key referencing Employee table.

The table satisfies 3NF requirements.

**Leave Table:**

1. Leave\_Id is the primary key.
2. Leave\_date is functionally dependent on Leave\_Id.
3. Employee\_Id is a foreign key referencing Employee table.

The table satisfies 3NF requirements.

**Employee\_Attendance Table:**

1. The primary key consists of two columns: Employee\_Id and Attendance\_Id.
2. Both Employee\_Id and Attendance\_Id are functionally dependent on the primary key.

The table satisfies 3NF requirements.

**Attendance Table:**

1. Attendance\_Id is the primary key.
2. Hours\_Worked is functionally dependent on Attendance\_Id.

The table satisfies 3NF requirements.

In conclusion, all tables adhere to the requirements of Third Normal Form (3NF) as they exhibit no transitive dependencies and every non-key attribute is directly dependent on the primary key.

**Creation of Data In Tables:-**

-- Inserting employee data

INSERT INTO Employee VALUES (101,'Ojas','Phansekar','2016-04-14','New York City','New York',1);

INSERT INTO Employee VALUES (102,'Vrushali','Patil','2018-06-21','Boston','Massachusetts',2);

INSERT INTO Employee VALUES (103,'Pratik','Parija','2019-09-13','Chicago','Illinois',3);

INSERT INTO Employee VALUES (104,'Chetan','Mistry','2011-04-12','Miami','Florida',4);

INSERT INTO Employee VALUES (105,'Anugraha','Varkey','2017-08-16','Atlanta','Georgia',5);

INSERT INTO Employee VALUES (106,'Rasagnya','Reddy','2018-07-25','San Mateo','California',6);

INSERT INTO Employee VALUES (107,'Aishwarya','Boralkar','2010-12-18','San Francisco','California',7);

INSERT INTO Employee VALUES (108,'Shantanu','Savant','2015-11-27','Seattle','Washington',8);

INSERT INTO Employee VALUES (109,'Kalpita','Malvankar','2016-04-24','Boston','Massachusetts',8);

INSERT INTO Employee VALUES (110,'Saylee','Bhagat','2014-05-21','San Francisco','California',7);



-- Inserting department data

INSERT INTO Department VALUES (1,'Human Resources');

INSERT INTO Department VALUES (2,'Software Development');

INSERT INTO Department VALUES (3,'Data Analysis');

INSERT INTO Department VALUES (4,'Data Science');

INSERT INTO Department VALUES (5,'Business Intelligence');

INSERT INTO Department VALUES (6,'Data Engineering');

INSERT INTO Department VALUES (7,'Manufacturing');

INSERT INTO Department VALUES (8,'Quality Control');



-- Inserting account details

INSERT INTO AccountDetails VALUES (40,'Santander','S12344',101);

INSERT INTO AccountDetails VALUES (41,'Santander','S12345',102);

INSERT INTO AccountDetails VALUES (42,'Santander','S12346',103);

INSERT INTO AccountDetails VALUES (43,'Santander','S12347',104);

INSERT INTO AccountDetails VALUES (44,'Chase','C12344',105);

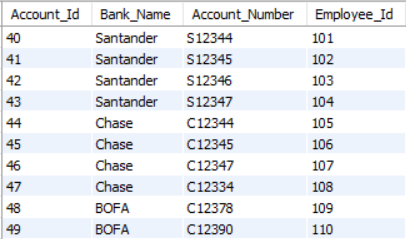
INSERT INTO AccountDetails VALUES (45,'Chase','C12345',106);

INSERT INTO AccountDetails VALUES (46,'Chase','C12347',107);

INSERT INTO AccountDetails VALUES (47,'Chase','C12334',108);

INSERT INTO AccountDetails VALUES (48,'BOFA','C12378',109);

INSERT INTO AccountDetails VALUES (49,'BOFA','C12390',110);



-- Inserting education data

INSERT INTO Education VALUES (10,101,'MS',2017);

INSERT INTO Education VALUES (11,102,'MS',2019);

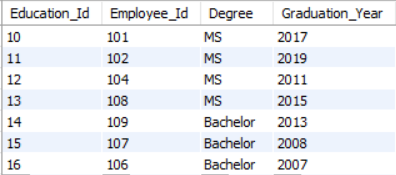
INSERT INTO Education VALUES (12,104,'MS',2011);

INSERT INTO Education VALUES (13,108,'MS',2015);

INSERT INTO Education VALUES (14,109,'Bachelor',2013);

INSERT INTO Education VALUES (15,107,'Bachelor',2008);

INSERT INTO Education VALUES (16,106,'Bachelor',2007);



-- Inserting leave data

INSERT INTO Leave\_ VALUES (51,104,'2019-12-01');

INSERT INTO Leave\_ VALUES (52,108,'2019-12-02');

INSERT INTO Leave\_ VALUES (53,109,'2019-12-03');

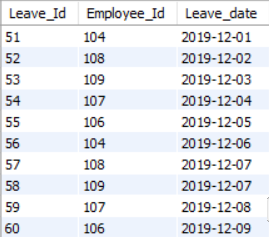
INSERT INTO Leave\_ VALUES (54,107,'2019-12-04');

INSERT INTO Leave\_ VALUES (55,106,'2019-12-05');

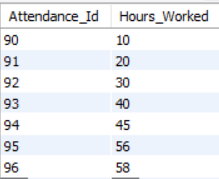
INSERT INTO Leave\_ VALUES (56,104,'2019-12-06');

INSERT INTO Leave\_ VALUES (57,108,'2019-12-07');

INSERT INTO Leave\_ VALUES (58,109,'2019-12-07');

INSERT INTO Leave\_ VALUES (59,107,'2019-12-08');

INSERT INTO Leave\_ VALUES (60,106,'2019-12-09');

-- Inserting attendance data

INSERT INTO Attendance VALUES (90,10);

INSERT INTO Attendance VALUES (91,20);

INSERT INTO Attendance VALUES (92,30);

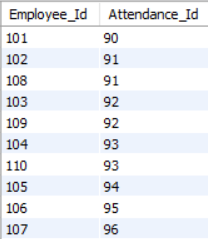
INSERT INTO Attendance VALUES (93,40);

INSERT INTO Attendance VALUES (94,45);

INSERT INTO Attendance VALUES (95,56);

INSERT INTO Attendance VALUES (96,58);

-- Inserting employee attendance data

INSERT INTO Employee\_Attendance VALUES (101,90);

INSERT INTO Employee\_Attendance VALUES (102,91);

INSERT INTO Employee\_Attendance VALUES (103,92);

INSERT INTO Employee\_Attendance VALUES (104,93);

INSERT INTO Employee\_Attendance VALUES (105,94);

INSERT INTO Employee\_Attendance VALUES (106,95);

INSERT INTO Employee\_Attendance VALUES (107,96);

INSERT INTO Employee\_Attendance VALUES (108,91);

INSERT INTO Employee\_Attendance VALUES (109,92);

INSERT INTO Employee\_Attendance VALUES (110,93);

-- Inserting salary data

INSERT INTO Salary VALUES (1,57600,30,200,1000,40);

INSERT INTO Salary VALUES (2,76800,40,300,1300,41);

INSERT INTO Salary VALUES (3,96000,50,400,1500,42);

INSERT INTO Salary VALUES (4,115200,60,500,1700,43);

INSERT INTO Salary VALUES (5,57600,30,200,1000,44);

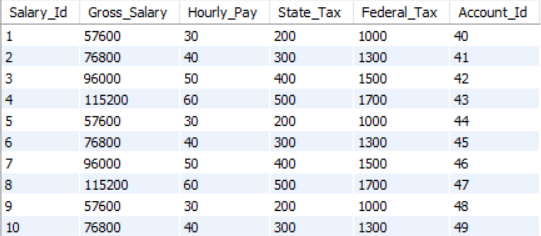
INSERT INTO Salary VALUES (6,76800,40,300,1300,45);

INSERT INTO Salary VALUES (7,96000,50,400,1500,46);

INSERT INTO Salary VALUES (8,115200,60,500,1700,47);

INSERT INTO Salary VALUES (9,57600,30,200,1000,48);

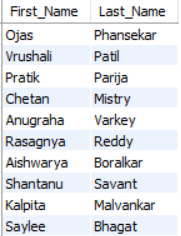
INSERT INTO Salary VALUES (10,76800,40,300,1300,49);



**Some SQL Queries :-**

Q1) Select all employees' first names and last names:

**SELECT First\_Name, Last\_Name FROM Employee;**



Q2)SQL Query To Calculate the average gross salary

**SELECT AVG(Gross\_Salary) AS Average\_Gross\_Salary FROM Salary;**

****

Q3) Select employee names along with their department names:

**SELECT e.First\_Name, e.Last\_Name, d.Department\_Name**

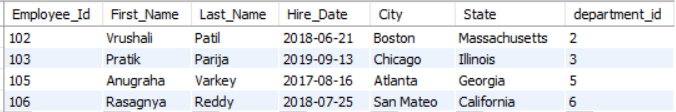
**FROM Employee e**

**INNER JOIN Department d ON e.Department\_id = d.Department\_Id;**



Q4)Select employees hired after a certain date:

SELECT \* FROM Employee WHERE Hire\_Date > '2016-12-12';



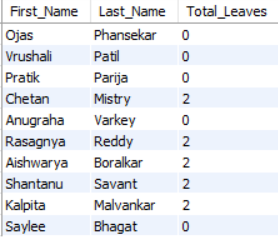
Q5) SQL Query To Find employee’s Leave Count

**SELECT e.First\_Name, e.Last\_Name, COUNT(l.Leave\_Id) AS Total\_Leaves**

**FROM Employee e**

**LEFT JOIN Leave\_ l ON e.Employee\_Id = l.Employee\_Id**

**GROUP BY e.First\_Name, e.Last\_Name;**



Q6) Find the bank with the highest number of accounts.

SELECT ad.Bank\_Name, COUNT(ad.Account\_Id) AS Total\_Accounts

FROM AccountDetails ad

GROUP BY ad.Bank\_Name

ORDER BY Total\_Accounts DESC

LIMIT 1;

Q7) Find the average number of hours worked by employees in each department.

**SELECT d.Department\_Name, AVG(a.Hours\_Worked) AS Avg\_Hours\_Worked**

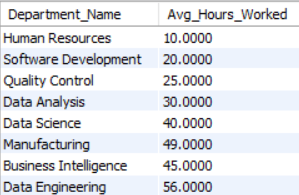
**FROM Department d**

**INNER JOIN Employee e ON d.Department\_Id = e.Department\_id**

**INNER JOIN Employee\_Attendance ea ON e.Employee\_Id = ea.Employee\_Id**

**INNER JOIN Attendance a ON ea.Attendance\_Id = a.Attendance\_Id**

**GROUP BY d.Department\_Name;**



**VIEWS:-**

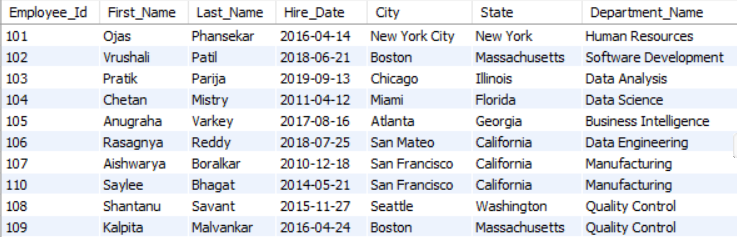
**1)Employee Details View:** This view combines information from the Employee table with the Department table to provide details about employees and their departments.

CREATE VIEW EmployeeDetails AS

SELECT e.Employee\_Id, e.First\_Name, e.Last\_Name, e.Hire\_Date, e.City, e.State, d.Department\_Name

FROM Employee e

INNER JOIN Department d ON e.Department\_id = d.Department\_Id;



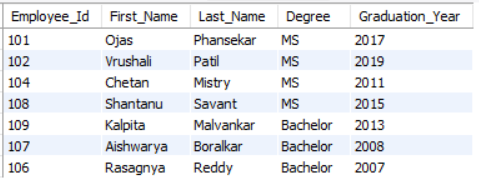
2)**Employee Education View :** This view combines information from the Employee table with the Education table to provide details about employee and their education.

**CREATE VIEW EmployeeEducation AS**

**SELECT e.Employee\_Id, e.First\_Name, e.Last\_Name, edu.Degree, edu.Graduation\_Year**

**FROM Employee e**

**INNER JOIN Education edu ON e.Employee\_Id = edu.Employee\_Id;**



**3)Employee Net Salary View :** This view combines information from the Employee table with the Salary and AccountDetails Tables to provide details about employee and their Net Salary.

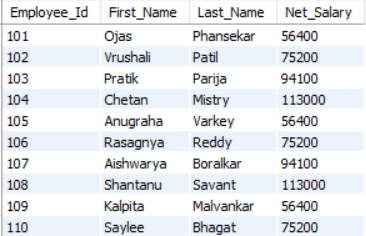
CREATE VIEW Employee\_Net\_Salary AS

SELECT E.Employee\_Id, E.First\_Name, E.Last\_Name, (S.Gross\_Salary - S.State\_Tax - S.Federal\_Tax) AS Net\_Salary

FROM Employee E

JOIN AccountDetails A ON E.Employee\_Id = A.Employee\_Id

JOIN Salary S ON A.Account\_Id = S.Account\_Id;



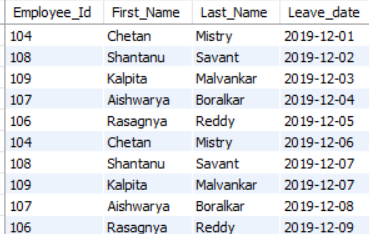
**4)Employee Leave View :** This view combines information from the Employee table with the Leave Tables to provide details about employee and their Leave History.

**CREATE VIEW Employee\_Leave\_History AS**

**SELECT E.Employee\_Id, E.First\_Name, E.Last\_Name, L.Leave\_date**

**FROM Employee E**

**JOIN Leave L ON E.Employee\_Id = L.Employee\_Id;**



**5)Employee Attendence View :** This view combines information from the Employee table with Attendance Tables to provide details about employee and their Attendance History.

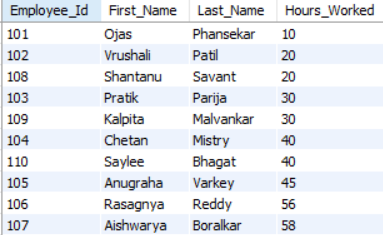
CREATE VIEW Employee\_Attendance\_Summary AS

SELECT E.Employee\_Id, E.First\_Name, E.Last\_Name, A.Hours\_Worked

FROM Employee E

JOIN Employee\_Attendance EA ON E.Employee\_Id = EA.Employee\_Id

JOIN Attendance A ON EA.Attendance\_Id = A.Attendance\_Id;



**6)Employee Hourly pay and Worked Hours View :** This view combines information from the Employee table with Attendance,AccountDetails,Salary,Employee\_Attendance Tables to provide details about employee and their Employee Hourly Pay and Worked Hour View.

**CREATE VIEW Employee\_Hourly\_Pay\_Hours\_Worked AS**

**SELECT E.Employee\_Id, E.First\_Name, E.Last\_Name, S.Hourly\_Pay, A.Hours\_Worked**

**FROM Employee E**

**JOIN AccountDetails AD ON E.Employee\_Id = AD.Employee\_Id**

**JOIN Salary S ON AD.Account\_Id = S.Account\_Id**

**JOIN Employee\_Attendance EA ON E.Employee\_Id = EA.Employee\_Id**

**JOIN Attendance A ON EA.Attendance\_Id = A.Attendance\_Id**;



|  |  |  |
| --- | --- | --- |
| 01 | K Sree Charan | AP22110010111 |
| 02 | Nukala Sai Charan | AP22110010091 |
| 03 | Ambatipalli Sree Charan Kumar | AP22110010072 |
| 04 | Revala Chandradithya | AP22110010127 |