CATHOLIC UNIVERSITY OF EASTERN AFRICA

PROJECT REPORT GROUP 29

EMPLOYEE PAYROLL MANAGEMENT SYSTEM

1061093- MITHAMO JOYRITA NJOKI

1048837- SYMINGTON MJOMBA

1061964- WARUI MATHENGE

1049351- ESCRIVA USHINDI

1049054- ISAYA DAVID MOMANYI

November 20, 2024

INTRODUCTION

Overview

The Employee Payroll Management System is designed to automate and streamline the payroll process. This includes tax calculations and automation of salary processes through handling various tasks, from calculating employee salaries and deductions to generating pay slips and tax reports.

Key Features of the system:

1. Employee Information

Employee data is essential to maintain a proper record of the employees and their personal information for various purposes like contacting them for inviting to certain summits, feedback from the company from the employee data

2. Maintaining Salary

It is very important to keep this data which will help not only the managers and the HR to keep track of the employee salaries but also help the company or its board to analyze what amount they are spending on a particular employee of a particular company

3. Work Location

An organization, small or big, needs to have a record of all the work locations they operate from to see how they can develop in that region and increase the hiring in that region so that the organization can increase their Market Outreach in that area.

4. Projects

To be successful the company should be involved in various projects, so they also need to maintain a record of the salaries each employee is being paid for a particular type of project he/she is working on.

Rationale and Objectives

- Efficiency and accuracy

Automated calculations reduce human error in data entry and manual calculations together with saving time and resources since they are streamlined.

- Compliance

The system adheres to labor laws and regulations and ensures compliance with the ever-changing tax laws. The records are kept up-to-date and accurate.

- Employee Satisfaction

The system is to ensure accurate and timely payments together with providing transparency in payroll calculations.

- Cost reduction

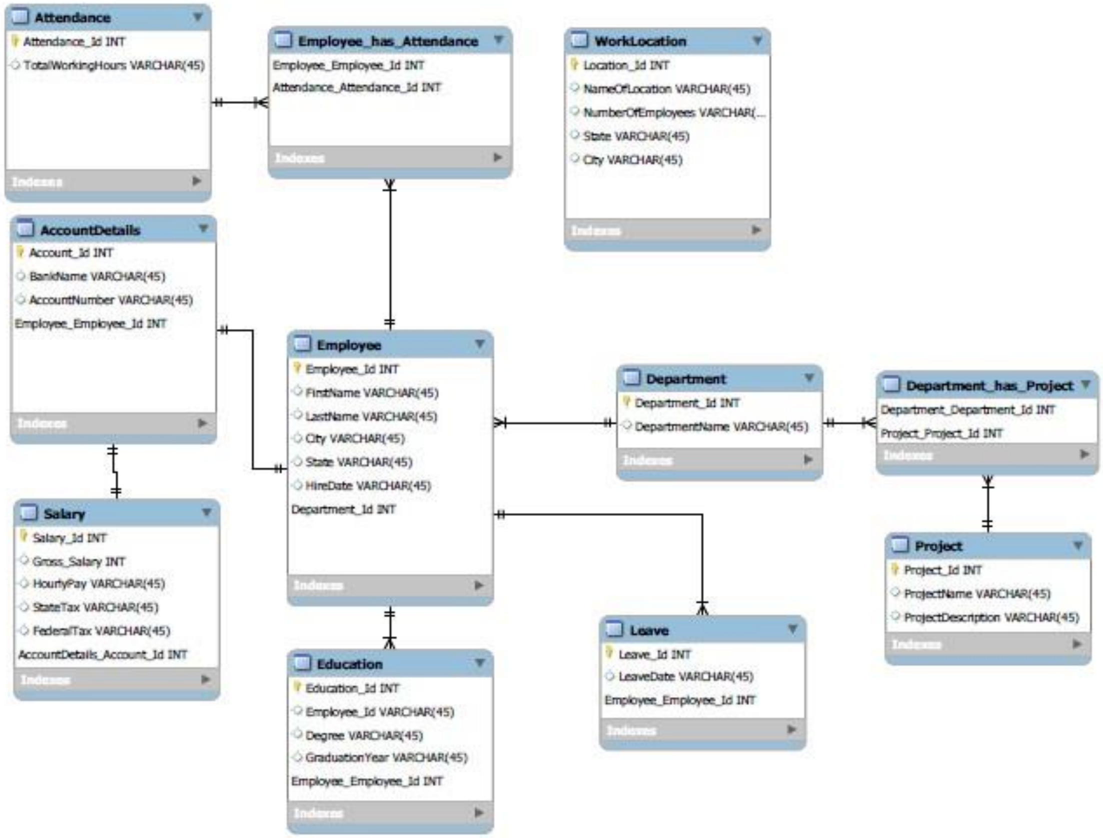
The system is to automate manual tasks and reduce the need for paperwork. This increases overall operational efficiency.

- Enhanced decision-

The system needs to provide real-time insights into making payroll costs and support informed decision-making based on accurate data.

SYSTEM DESIGN

1. EER DIAGRAM



2. TABLE STRUCTURE AND SQL SCHEMA

Employee

- Employee_Id (INT, PRIMARY KEY)
- First_Name (VARCHAR)
- Last_Name (VARCHAR)
- Hire_Date (DATE)
- City (VARCHAR)
- State (VARCHAR)

Department

- Department_Id (INT, PRIMARY KEY)
- Department_Name (VARCHAR)

Salary

- Salary_Id (INT, PRIMARY KEY)
- Gross_Salary (INT)
- Hourly_Pay (INT)
- State_Tax (INT)
- Federal_Tax (INT)
- Account_Id (INT, FOREIGN KEY)

DepartmentProject

- Department_Id (INT, PRIMARY KEY, FOREIGN KEY)
- Project_Id (INT, PRIMARY KEY, FOREIGN KEY)

Project

- Project_Id (INT, PRIMARY KEY)
- Project_Name (VARCHAR)
- Project_Description (VARCHAR)

AccountDetails

- Account_Id (INT, PRIMARY KEY)
- Bank_Name (VARCHAR)
- Account_INT (VARCHAR)
- Employee_Id (INT, FOREIGN KEY)

Education

- Education_Id (INT, PRIMARY KEY)
- Employee_Id (INT, FOREIGN KEY)
- Degree (VARCHAR)
- Graduation_Year (INT)

Restdays

- Leave_Id (INT, PRIMARY KEY)
- Employee_Id (INT, FOREIGN KEY)
- Leave_date (DATE)

EmployeeAttendance

- Employee_Id (INT, PRIMARY KEY, FOREIGN KEYS)
- Attendance_Id (INT, PRIMARY KEY)

Attendance

- Attendance_Id (INT, PRIMARY KEY)
- Hours_Worked (INT),

WorkLocation

- Location_Id (INT, PRIMARY KEY)
- Location (VARCHAR)
- INT_Of_Employees (INT)
- City (VARCHAR)
- State (VARCHAR)

IMPLEMENTATION

CRUD OPERATIONS AND ADVANCED SQL QUERIES

1. **Create Operations**

- **Adding a New Employee:** Insert a new record into the employee's table, specifying the employee ID, name, hire date, and salary.

- **Adding a New Payroll Record in the Salary Table:** Insert a new record into the salary table, specifying the employee ID, gross pay, and taxes applied.

2. **Read Operations**

- **Retrieving All Employees:** Select all records from the employee's table.

- **Retrieving a Specific Employee:** Select records from the employee's table based on a specific employee ID.

- **Retrieving Salary Records for a Specific Employee:** Select records from the Salary table based on a specific employee ID.

3. **Update Operations**

- **Updating Employee Information:** Update specific fields in the employees' table for a given employee ID.

- **Updating Salary records:** Update specific fields in the payroll table for a given employee ID and pay period.

4. **Delete Operations**

- **Deleting an Employee:** Delete a record from the employee's table based on a specific employee ID.
- **Deleting a Payroll Record:** Delete a record from the payroll table based on a specific employee ID and pay period.

3. Advanced SQL Queries

- **Calculating Total Payroll Expenses for a Specific Period:** Calculate the sum of gross pay for a given date range.
- **Identifying Top Earners:** Sort employees by salary in descending order and limit the result to a specific number.
- **Analyzing Employee Attendance:** Calculate the total number of employees hired and those that have been active.

CONCLUSION AND RECOMMENDATIONS

Summary

The Employee Payroll Management System has been successfully implemented, providing a solution for managing employee payroll processes. The system has streamlined operations, reduced manual errors, and improved overall payroll accuracy. Key features such as tracking attendance, generating reports, and centralized employee data, have significantly enhanced payroll management efficiency.

Future Improvements

1. **Integration with Human Resource Information System:** This will streamline data synchronization and reduce manual data entry.
2. **Enhanced Reporting Capabilities:** Customizable reports and data visualizations will enable deeper insights into payroll trends and costs.
3. **Mobile Accessibility:** Implementing a mobile application will allow authorized users to access and view payroll information on the go.
4. **Time and Attendance Integration:** This will automate time tracking and calculation of employee hours, improving payroll accuracy.
5. **Tax Compliance Updates:** Regularly updating the system to comply with changing tax laws and regulations will ensure accurate tax calculations and reporting.
6. **Security and Access Controls:** Implementing security measures, such as role-based access control and data encryption, to protect sensitive payroll data will be a huge step in the security of the system.

7. **Cloud-Based Deployment:** Consider migrating the system to a cloud-based platform to improve scalability, accessibility, availability, and disaster recovery capabilities

REFERENCES

Textbooks

Silberschatz, A., Korth, H.F., & Sudarshan, S. (2019) Database System Concepts. Mc.Graw-Hill Education

W3Schools SQL Tutorial: https://www.w3schools.com/sql/

MySQL Documentation: https://dev.mysql.com/doc/

Oracle Documentation: https://docs.oracle.com/en/database/oracle/oracle-database/

APPENDICES

Create statements

```
----------------------------------------------------------------------
/* Create a database and use the database for the rest of the SQL script */
----------------------------------------------------------------------
Create database Employee_DBMS;
use Employee_DBMS;
----------------------------------------------------------------------
/* Create tables that you will later input values into
Each entity had its code just like the one below*/
----------------------------------------------------------------------
CREATE TABLE Employee(
  Employee_Id int(6) primary key,
  First_Name VARCHAR(25),
  Last_Name VARCHAR(25),
  Hire_Date DATE,
  City VARCHAR(25),
  State VARCHAR(25) );
-------------------------------------------------
Department
-------------------------------------------------
  CREATE TABLE Department(
  Department_Id INT PRIMARY KEY,
  Department_Name VARCHAR(30)
  );
```

```
 ------------------------------------------------
 Salary
 ------------------------------------------------

 CREATE TABLE Salary(
 Salary_Id INT Primary Key,
 Gross_Salary INT,
 Hourly_Pay INT,
 State_Tax INT,
 Federal_Tax INT,
 Account_Id INT,
 FOREIGN KEY (Account_Id)  REFERENCES AccountDetails(Account_Id)
 );

 ------------------------------------------------
 DepartmentProject Bridge
 ------------------------------------------------
 CREATE TABLE DepartmentProject(
 Department_Id INT,
 Project_Id INT,
PRIMARY KEY (Department_Id,Project_Id),
 FOREIGN KEY (Department_Id) REFERENCES Department(Department_Id),
 FOREIGN KEY (Project_Id) REFERENCES Project(Project_Id)
 );
 -------------------------------------------------
 Project
```

```
------------------------------------------------
CREATE TABLE Project(
Project_Id INT Primary key,
Project_Name VARCHAR2(50),
Project_Description VARCHAR2(50) );

-------------------------------------------------
AccountDetails
-------------------------------------------------

CREATE TABLE AccountDetails(
Account_Id INT primary key,
Bank_Name VARCHAR(50),
Account_Id VARCHAR(50),
Employee_Id INT,
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),
PRIMARY KEY (Education_Id),
FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_Id)
);
```

```
 -------------------------------------------------
 Restdays
 -------------------------------------------------

 CREATE TABLE Restdays(
 Leave_Id INT primary key,
 Employee_Id INT,
 Leave_date DATE,
 FOREIGN KEY (Employee_Id) REFERENCES Employee(Employee_Id)
 );

--------------------------------------------------
EmployeeAttendance
--------------------------------------------------

 CREATE TABLE Employee_Attendance(
 Employee_Id INT,
 Attendance_Id INT,
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 primary key,

Hours_Worked INT,

);


---------------------------------------------------

WorkLocation

---------------------------------------------------

CREATE TABLE Work_Location(

Location_Id INT Primary key,

Location VARCHAR(25),

No_Of_Employees INT,

City VARCHAR(25),

State VARCHAR(25)  );
```

Insert Statements

```
INSERT INTO Employee VALUES (101,'Ojas','Phansekar','2024-04-07','New York City','New York'),

(102,'Vrushali','Patil','2018-06-21','Boston','Massachusetts'),

(103,'Pratik','Parija','2019-09-13','Chicago','Illinois'),

(104,'Chetan','Mistry','2011-04-12','Miami','Florida'),

(105,'Anugraha','Varkey','2017-08-16','Atlanta','Georgia'),

(106,'Rasagnya','Reddy','2018-06-25','San Mateo','California'),

(107,'Aishwarya','Boralkar','2010-12-15','San Francisco','California'),

(108,'Shantanu','Savant','2015-11-27','Seattle','Washington'),

(109,'Kalpita','Malvankar','2016-04-24','Boston','Massachusetts'),

(110,'Saylee','Bhagat','2014-05-19','San Francisco','California');
```

```
INSERT INTO Department VALUES (1,'Human Resources'),
(2,'Software Development'),
(3,'Data Analysis'),
(4,'Data Science'),
(5,'Business Intelligence'),
(6,'Data Engineering'),
(7,'Manufacturing'),
(8,'Quality Control');

INSERT INTO Project VALUES (21,'Dev','Whatever'),
(22,'Prod','do something'),
(23,'Test','focus'),
(24,'Nothing','do nothing'),
(25,'Research','focus on everything'),
(26,'Next Steps','find some way out');

INSERT INTO AccountDetails VALUES (40,'Santander','S12344',101),
(41,'Santander','S12345',102),
(42,'Santander','S12346',103),
(43,'Santander','S12347',104),
(44,'Chase','C12344',105),
(45,'Chase','C12345',106),
(46,'Chase','C12347',107),
(47,'Chase','C12334',108),
(48,'BOFA','C12378',109),
(49,'BOFA','C12390',110);
```

```
INSERT INTO Education VALUES (10,101,'MS',2017),
(11,102,'MS',2019),
 (12,104,'MS',2011),
(13,108,'MS',2015),
(14,109,'Bachelor',2013),
(15,107,'Bachelor',2008),
(16,106,'Bachelor',2007);

INSERT INTO Restdays VALUES (51,104,'2019-12-01'),
(52,108,'2019-08-19'),
(53,109,'2019-07-23'),
(54,107,'2019-03-19'),
(55,106,'2019-06-04'),
(56,104,'2019-08-05'),
(57,108,'2019-04-06'),
(58,109,'2019-12-07'),
(59,107,'2019-05-08'),
(60,106,'2019-01-09');

INSERT INTO Attendance VALUES (90,10),
(91,20),
(92,30),
(93,40),
(94,45),
(95,56),
(96,58);
```

```
INSERT INTO Work_Location VALUES (71,'North',4,'New York City','New York'),
(72,'North',4,'Boston','Massachusetts'),
(73,'North',4,'Chicago','Illinois'),
(74,'North',89,'Miami','Florida'),
(75,'South',90,'Atlanta','Georgia'),
(76,'South',100,'San Mateo','California'),
(77,'South',4,'San Francisco','California'),
(78,'South',2,'Seattle','Washington'),
(79,'South',25,'Alpharetta','Georgia'),
(80,'South',20,'Keene','New Hampshire'),
(81,'South',22,'Hampton','New Hampshire');


INSERT INTO Employee_Attendance VALUES (101,90),
(102,91),
(103,92),
(104,93),
(105,94),
(106,95),
(107,96),
(108,91),
(109,92),
(110,93);


INSERT INTO DepartmentProject VALUES (1,21),
(2,22),
(3,23),

(4,24),

(5,25),

(6,26),

(7,21),

(8,24);


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

(2,76800,40,300,1300,41),

(3,96000,50,400,1500,42),

(4,115200,60,500,1700,43),

(5,57600,30,200,1000,44),

(6,76800,40,300,1300,45),

(7,96000,50,400,1500,46),

(8,115200,60,500,1700,47),

(9,57600,30,200,1000,48),

(10,76800,40,300,1300,49);
```

Update statements

```
Update Employee

set State= 'Massachusetts'

where Employee_Id = 105;
```

Procedures

```
/*This procedure is used when we want to get information of a certain employee*/

------------------------------------------------------------------------------------------
```

```
CREATE PROCEDURE `GetEmployeeInfo`(IN p_employee_id INT)

BEGIN

  SELECT

    Employee_Id,

    First_Name,

    Last_Name,

    Hire_Date

  FROM

    Employee

  WHERE

    Employee_Id = p_employee_id;

END

-----------------------------------------------------------------------------------------

~~ Call the procedure

-----------------------------------------------------------------------------------------

CALL GetEmployeeInfo(106);




-----------------------------------------------------------------------------------------

~~ This procedure is used to update the salary of a certain employee

-----------------------------------------------------------------------------------------

CREATE PROCEDURE `UpdateEmployeeSalary`(IN p_employee_id INT, IN p_new_salary DECIMAL(10,2))

BEGIN

  UPDATE Salary

  SET Gross_Salary = p_new_salary
```

```
   WHERE Salary_Id = p_employee_id;

END

-------------------------------------------------------------------------------------

~~ Call the procedure

-------------------------------------------------------------------------------------

CALL UpdateEmployeeSalary(3,102000);
```

Views

```
~~ This view is used to display the inactive employees in the organization

CREATE VIEW inactive_employees AS

SELECT Employee_Id, First_Name,Last_Name, Hire_Date

FROM Employee

WHERE Employee_Id = 105;

 SELECT * FROM inactive_employees;


~~ This view is used to display the active employees

CREATE VIEW active_employees AS

SELECT Employee_Id, First_Name,Last_Name, Hire_Date

FROM Employee

WHERE Employee_Id = 107;

SELECT * FROM active_employees;


~~ This view is used to display an overview of Federal Tax and the total amount collected from
salary


CREATE VIEW tax_summary AS
```

```
SELECT Federal Tax, SUM(Gross_Salary) AS total_tax

FROM Salary

INNER JOIN AccountDetails ON AccountDetails.Account_Id = Salary.Account_Id

GROUP BY Federal_Tax;

SELECT * FROM tax_summary;
```