



## **Project on Employee Database Management System**

**FINC535 Data Management Final Project**

**Instructor: Xiaodi Zhu**

### **Team Members:**

Manoj Nair

Preeta Pritam Jha

Thi Diem My Nguyen

**Date: December 17, 2021.**



## **1. Introduction**

### **1.1. Business problem**

Manual handling of employee information poses a number of challenges. This is evident in procedures such as leave management where an employee is required to fill in a form which may take several weeks or months to be approved. The use of paperwork in handling some of these processes could lead to human error, papers may end up in the wrong hands and not forgetting the fact that this is time consuming.

A number of current systems lack employee self-service meaning employees are not able to access and manage their personal information directly without having to go through their HR departments or their managers.

The aforementioned problems can be tackled by designing and implementing a web-based HR management system. This system will maintain employee information in a database by fully privacy and authority access. The project is aimed at setting up an employee information system about the status of the employee, the educational background, and the work experience in order to help monitor the performance and achievements of the employee through a password-protected system.

## **1.2. Purpose of the system**

In this world of growing technologies, everything has been computerized. With a large number of work opportunities, the Human workforce has increased. Thus, there is a need for a system which can handle the data of a large number of Employees. This project simplifies the task of maintaining records because of its user-friendly nature.

The objective of this project is to provide a comprehensive approach towards the management of employee information. This will be done by designing and implementing an HR management system that will bring up a major paradigm shift in the way that employee information is handled.

The objectives of this system include:

- Design of a web-based HR management system to fulfill requirements such as project management, leave management, report generation to assist in performance appraisal, ESS and employee training.
- Well-designed database to store employee information.
- A user friendly front-end for the user to interact with the system.

## **1.3. Benefits of the system**

This system is expected to be user friendly and will offer easy access to data as well as services such as online leave management, compensation and timely report generation. Some benefits of the system:

- This system will reduce the complexity of employee management.
- By using this system, we can easily maintain all the records about "ON EMPLOYEES" or "OFF EMPLOYEES" (inactive employee).
- It will reduce searching time.
- It can be easily handled by the person who has elementary knowledge of computers because it provides a user-friendly environment.

## 2. Collect database

We are going to design a web application for Managing Employees. The expected audiences of this document are the developers and the admin of the web application. Now with the help of this system the admin has the information on his fingertips and can easily prepare a good record based on their requirements. This software has met the scope. As there is no data involved in the system, processing on the file, and the behavior of this project is already identified and bundled in a quantitative manner.

## 3. Collected entities, tables and data type:

**Employee** - We designed the Employee Table to store the employee data.

Below mentioned is the description of all the columns of the Employee emp\_id is a primary key for Employee.

EMPLOYEE					
Column Name	data Type	Primary Key	Foreign Key	Not Null	Description
emp_dep_id	Int		Yes	Yes	Department Id of employee
emp_id	int	Yes		Yes	Primary Key Id of the employee
user_id	Int		Yes		user id of employee
emp_name	varchar(100)				The name of employee
emp_mobile	int				The mobile number of the employee
emp_email	varchar(100)				The mail of the employee
emp_address	varchar(100)				The address of the employee

**LEAVE-** We designed the Leave table to store the information of the employee leaves

LEAVE					
Column Name	data Type	Primary Key	Foreign Key	Not Null	Description
leave_id	Int	Yes		Yes	Id of leaving
emp_id	int		Yes	Yes	Employee Id of leaving
leave_type	varchar(10)				Type of leaving
leave_from	date				Date leaving
Leave_to	date				Leaving to date
leave_status	varchar(10)				Leave status

**DEPARTMENT-** We designed the department table to store the Department information of the employee

DEPARTMENT					
Column Name	data Type	Primary Key	Foreign Key	Not Null	Description
emp_dep_id	int	Yes	Yes	Yes	Department Id of employee
dept_title	varchar(100)				Title of the Department
dept_desc	varchar(100)				Description of the Department

**LOGIN-** We designed the Login table to store the login details of the employees

LOGIN					
Column Name	Data Type	Primary Key	Foreign Key	Not Null	Description
login_id	int	Yes		Yes	Login ID
login_role_id	int		Yes	Yes	ID of the login role
login_username	varchar (45)				Username of the login
login_userpwd	varchar (45)				Password of the login

**USERS-** We designed the User Table to store user information. Users can manage their own profiles. Also, the users can use the application according to the system roles assigned to them.

USERS					
Column Name	Data Type	Primary Key	Foreign Key	Not Null	Description
user_id	int	Yes		Yes	User ID
login_login_id	int		Yes	Yes	login user id
user_name	varchar (45)				Name of the user
user_email	varchar (45)				Email of the user
user_mobile	int				Mobile of the user
user_address	varchar (45)				Address of the user

**ROLES-**we designed the Role Table to store the system and organization roles. The column type can be used to identify whether the role is for the application users or organization employees.

ROLES					
Column Name	Data Type	Primary Key	Foreign Key	Not Null	Description
role_id	int	Yes		Yes	Role ID
User_user_id	int		Yes	Yes	User id
role_name	varchar (45)				Name of role
role_desc	varchar (45)				Description of the role

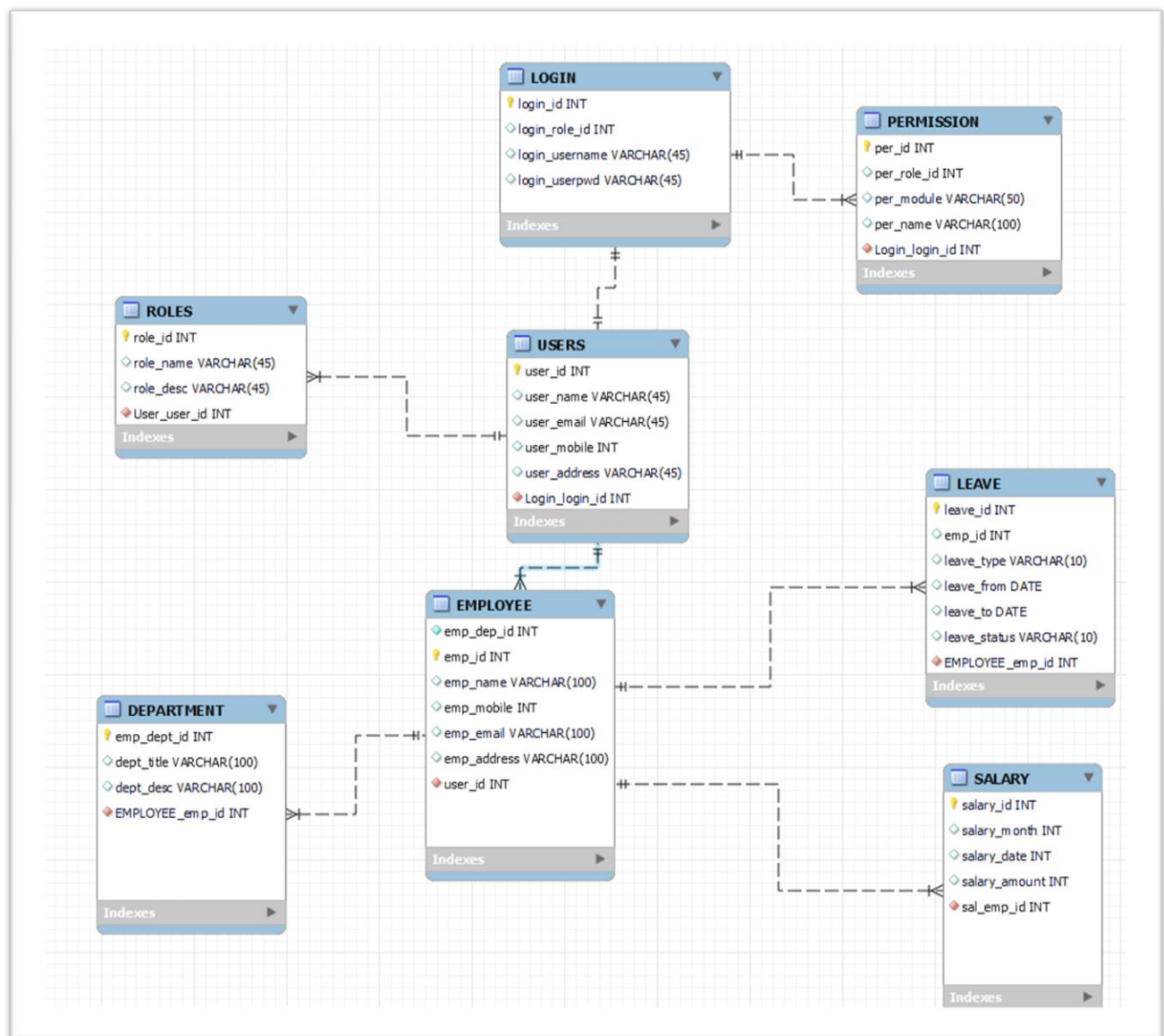
**PERMISSION-** we designed the Permission Table to store the system and organization permissions. The column type can be used to identify whether the permission is for the application users or organization employees.

PERMISSION					
Column Name	Data Type	Primary Key	Foreign Key	Not Null	Description
per_id	int	Yes		Yes	Permission ID
per_role_id	int		Yes	Yes	Permission role ID
per_module	varchar (50)				Permission module
per_name	varchar(100)				Permission name

**SALARY-** We designed the SALARY table to store the salary information of the employee

SALARY					
Column Name	Data Type	Primary Key	Foreign Key	Not Null	Description
salary_id	int	Yes		Yes	Salary ID
sal_emp_id	int		Yes	Yes	Employee id
salary amount	int				Amount of salary
salary_month	int				Salary month
salary_date	Date				Salary date

#### 4. Design a relational database to store the data



#### 5. Database selection

Our data is collected and stored in MySQL, to select data from MySQL, SELECT QUERY is used to fetch the data from the MySQL database. The purpose of MySQL Select is to return from the database tables, one or more rows that match a given criterion.

##### Some demos of data selections:

- To see the complete information about the employees:



```
SELECT * FROM employee;
```

- To find employees who first names start with the letter 'Da':

```
SELECT emp_name FROM employee WHERE emp_name LIKE 'Da%';
```

- To list the name and salary of FRANK:

```
SELECT emp_name, salary_amount  
FROM employee, salary  
WHERE  
employee.emp_id = salary.sal_empl_id AND  
employee.emp_name = 'FRANK';
```

- List of employees who has taken leave

```
SELECT emp_name FROM employee, leave  
WHERE  
Employee.emp_id = leave.employee_emp_id
```

- Department wise list of employees

```
SELECT dept_desc, emp_name FROM employee, department  
Where Employee.emp_id = department.employee_emp_id
```

## 6. Store the data using RDBMS and NoSQL database

With over four decades of presence, Structured Query Language (SQL) works with relational database management systems (RDBMS). It is the traditional way of creating databases and managing large volumes of data. MySQL, an open-source DBMS is still the most popular database management system and drives large organizations like Google, Twitter, and even Facebook. Some other SQL based DBMS include Oracle and SQL Server.

But owing to the increasing demand for data and better performance, organizations are testing new technologies for database management. This is the reason NoSQL has become a popular choice in the last few years. As the name suggests, NoSQL is for data that is 'Not only for SQL'. Thus, NoSQL is table less which enables it to handle multiple data types.

Despite NoSQL existing since the early 1960s, it is only because of options like MongoDB and HBase that it has gained traction. So, with this sudden influx of user base, it has become important to analyze the SQL vs NoSQL argument based on two major factors: Performance and Security.

**\*\*MySQL** is an open-source database that is platform independent and can easily interface with a number of scripting languages. The number of advantages of

using MySQL which include, the ability to handle stored procedures, triggers, SQL and User defined functions. It also offers a high-speed data load utility and support for various drivers (ODBC, JDBC, .NET, PHP). Deploying a MySQL database has proved to be cheap and easy as it doesn't require special hardware or software requirements, it can work well on any web server, but most professionals recommend the apache web server. MySQL is an excellent database to use when developing web-based applications because its platform independent and can easily interface with a number of scripting languages.

### **Pros and cons of using RDBMS and NoSQL database for our case:**

#### **Features Of RDBMS**

- RDBMS are table-based databases
- Data store in rows and columns
- Structured way of data storage
- Each row contains a unique instance of data for the categories defined by the columns.
- Provide facility primary key, to uniquely identify the rows
- Integrity constraints maintain data consistency across multiple tables

#### **Limitations for RDBMS**

**Scalability:** Users have to scale relational database on powerful servers that are expensive and hard to handle. To scale relational database, it must be distributed on to multiple servers. Handling tables across different servers is difficult.

**Complexity:** In case your database doesn't fit into tables anyhow, then you will need to design your database structure that will be complex and again difficult to handle.

#### **Features of NoSQL**

- It's a collection of key-values pair, documents and wide-column stores without any standard schema definition
- No Schema or Fixed Data model
- Highly and easily scalable
- No complex relationships, such as the ones between tables and RDBMS

- Maintaining NoSQL Servers is Less Expensive
- Supports integrated caching

## **LIMITATIONS OF NOSQL**

- NoSQL database is Open Source and Open Source at its greatest strength but at the same time its greatest weakness because there are not many defined standards for NoSQL databases, so no two NoSQL databases are equal
- No Stored Procedures in mongo dB (NoSQL database).
- Lack of reporting tools for performance testing and analysis
- too difficult for finding NoSQL experts because it is latest technology and NoSQL developer are in learning mode

## **Conclusion**

Looking at the two most important factors for any database management system we have come to the conclusion that achieving the best of both worlds is not possible in all aspects. The users need to analyze their requirements and act accordingly.

While SQL has been around for longer and is a favorite of the developers it will not be able to match the performance features that a NoSQL system will offer. At the same time, to be able to expand with NoSQL to get better performance, security will have to be compromised.

Hence, RDBMS and NoSQL both database are great in data management and both are used to keep data storage and retrieval optimized and smooth. It's hard to say which technology is better so developer take decision according to requirement and situations.

With our current requirement SQL database is the best options.

## **7. Potential issues**

To implement the project, we may face a number of challenges in collecting consistent and quality data. To develop methods to improve data collection practices, it is necessary to first identify barriers to consistent data collection. This section we identify common data collection challenges:

### **Inconsistent data collection standards**

Data standards outline how common data items and demographic information should be collected.

When collecting data, the HR may not always broadly apply these standards or may themselves be inconsistent, and this can impact the comparability of data collections.

As a result, the scope and detail of information collection may not be consistent across the company, making it complex to compare data, or to population level data sets.

### **Economic and IT restrictions**

Some organizations may not have the capacity or infrastructure to prioritize improvements to data collection systems and processes. This may be due to a backlog of paper-based records to be digitized, a small workforce to input and maintain data, and lack of budget to upgrade records management systems. It is also important to note that many IT systems are provided by government departments, who also carry the responsibility of resourcing and conducting system updates. These updates can be expensive and take time.

In some cases, these IT systems may have limited capacity to include multiple response values or dynamic questioning, that supports sophisticated data collection. The introduction of multiple response options may also present problems for exporting and analyzing data.