

**PES UNIVERSITY, Bengaluru Department of Computer Science  
and Engineering  
B. Tech (CSE) – 5<sup>th</sup> Semester – Aug-Dec 2024**



**Subject: Database Management System  
Project: Employee Leave Management System**

**NAME-1: PUNITH KUMAR C  
SRN: PES1UG23CS845**

**NAME-2: MOHAN P B  
SRN: PES1UG23CS848**

**SECTION: L**

**1. Frontend (PHP):**

- User Interface: PHP is used to create dynamic web pages for employees and administrators. Employees can log in, view their leave balances, request leave, and track their leave status. Administrators can view and approve leave requests, manage employee details, and update department information.
- User Authentication: Secure login functionality for employees and admin users, likely using sessions or cookies to maintain secure access.

**2. Backend (XAMPP with Apache and MySQL):**

- MySQL Database: Stores essential data, including employee records, leave types, department info, and leave requests. Tables such as `tblemployees`, `tblleaves`, and `tbldepartments` help organize and retrieve this data efficiently.
- Apache Server: Hosts the application locally through XAMPP, providing a testing environment where PHP scripts can run and interact with the database.
- Transactions: Used to maintain data integrity during critical operations, like submitting leave requests or updating leave balances, ensuring that related operations either complete fully or not at all.
- Procedures and Functions: Custom functions and stored procedures simplify complex queries, like calculating an employee's remaining leave balance or retrieving department names by ID.
- Triggers: Although not explicitly implemented here, triggers could be used to automate tasks like updating leave balances or sending notifications when a leave request is created or updated.

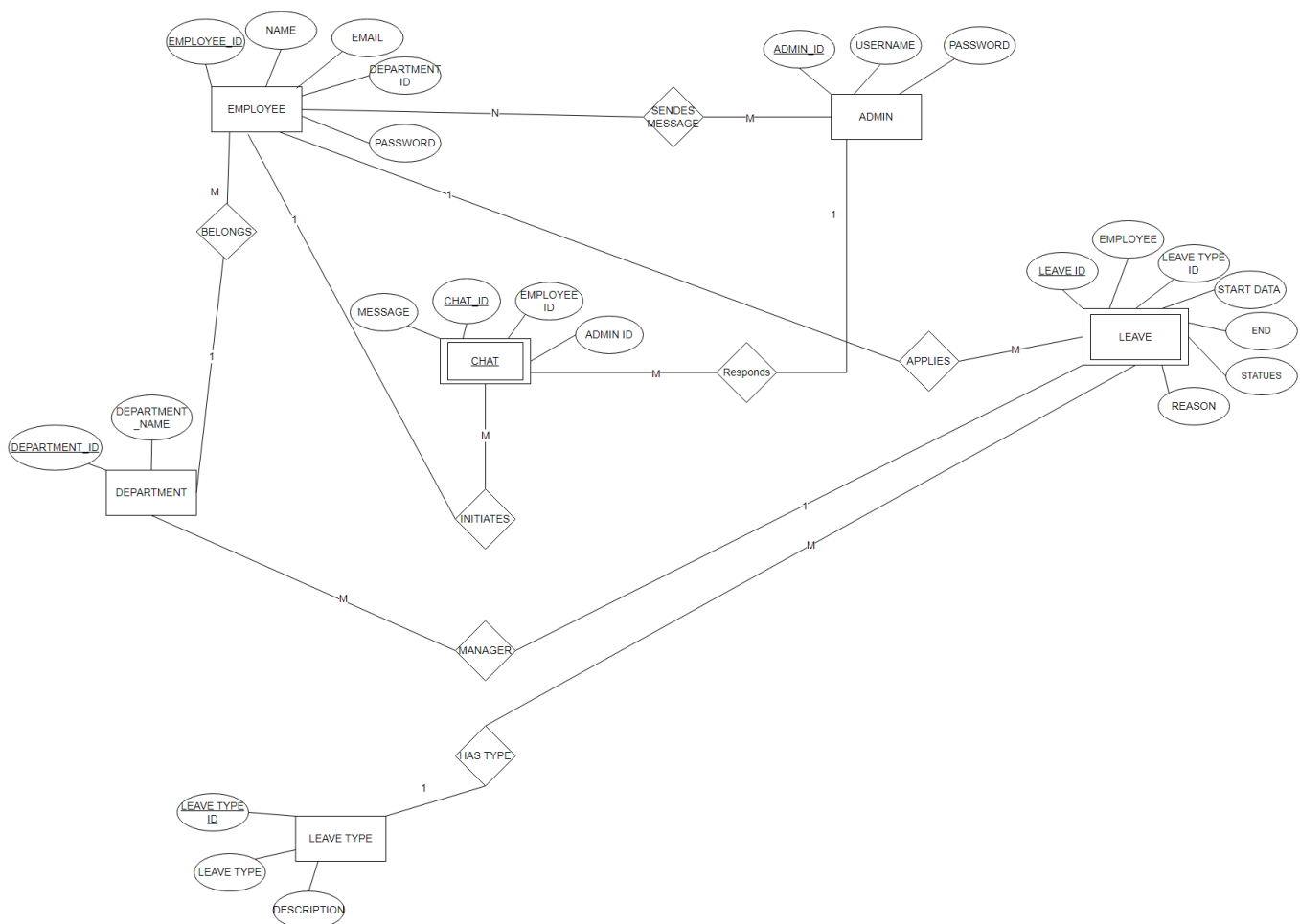
## System Workflow:

1. Employee Actions: Employees log in, view their current leave status, and request leave by specifying leave type, start and end dates, and a description.
2. Admin Actions: Administrators review and approve leave requests, add or update employee and department details, and manage overall leave policies.
3. Data Integrity: Transactional operations and database procedures maintain data consistency and accuracy, especially with sensitive data like leave balances and request statuses.

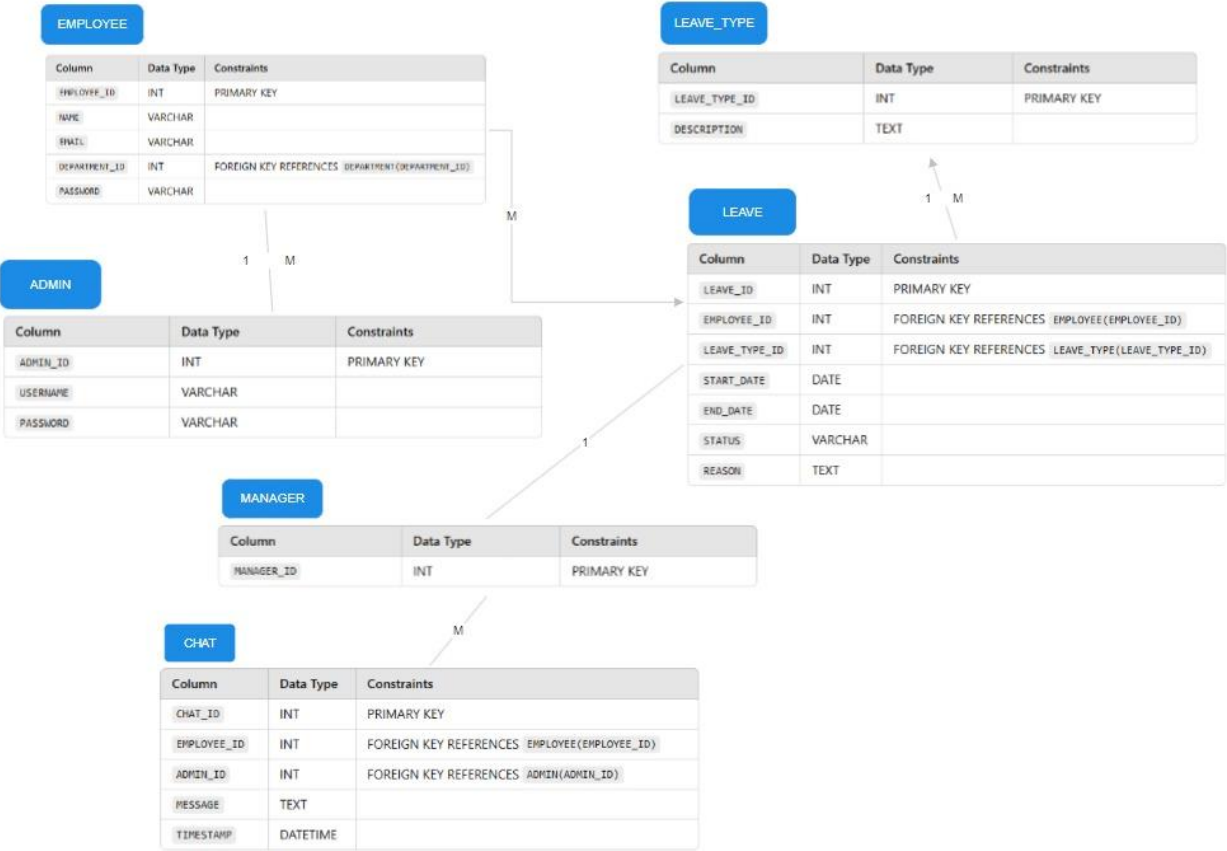
## Development Environment

- XAMPP: A local server solution that packages Apache, MySQL, and PHP, making it easier to test and deploy the application locally.
- MySQL Workbench or phpMyAdmin: Likely used to manage the database, create tables, and run queries.

## ER Diagram:



# Relational Model:



## Frontend: Employee Login:

[Admin Login](#) [Employee Login](#)

### Welcome to ELMS

#### EMPLOYEE LOGIN

Enter Registered Email Id

jackie@gmail.com

Enter Password


....

LOGIN


[Emp Password Recovery?](#)


## Employee profile:


Employee Leave Management System




**Jackie chan**  
12

 My Profiles

 Change Password

 Leaves >

 Logout

UPDATE EMPLOYEE DETAILS

Employee Code

12

Male

▼

Date of Birth

19 November, 2024

First name

Jackie

Last name

chan

Human Resource

▼

Country

India

Email

jackie@gmail.com

City/Town

Chitradurga

Address

cta

Mobile number


1235

UPDATE


ELMS


## Applying for leave:


Employee Leave Management System




Jackie chan  
12

 My Profiles

 Change Password

 Leaves >

 Logout

ELMS

APPLY FOR LEAVE

From Date  
15/02/2024

To Date  
21/02/2024

Casual Leave

Description  
Please grant leave

APPLY

18/500

## Admin Login:

Admin LoginEmployee Login

Welcome to ELMS

ADMIN LOGIN


Enter Username  
admin

Enter Password  
\*\*\*\*\*

LOGIN

Admin Dashboard:

Employee Leave Management System



Admin

Dashboard

Department

Leave Type

Employees

Leave Management

Change Password

Log Out

ELMS

TOTLE REGD EMPLOYEE

2

TOTAL DEPARTMENTS

3

TOTAL LEAVE TYPE


3

LATEST LEAVE APPLICATIONS

Sl No.	Employee Name	Leave Type	Posting Date	Status	Action
1	Jackie chan(12)	Casual Leave	2024-11-19 14:21:56	waiting for approval	<div>VIEW DETAILS</div>
2	Jackie chan(12)	Casual Leave	2024-11-19 14:17:48	waiting for approval	<div>VIEW DETAILS</div>

Admin taking action on leave:

Employee Leave Management System



Admin

Dashboard

Department

Leave Type

Employees

Leave Management

Change Password

Log Out

ELMS

LEAVE DETAILS

LEAVE DETAILS

Employee Name :	Jackie chan	Emp Id :	12	Gender :	Male
Emp Email id :	jackie@gmail.com	Emp Contact No. :	1235		
Leave Type :	Casual Leave	Leave Date . :	From 15/02/2024 to 21/02/2024	Posting Date	2024-11-19 14:21:56
Employee Leave Description :	Please grant leave				
leave Status :	waiting for approval				
Admin Remark:	waiting for Approval				
Admin Action taken date :	NA				

TAKE ACTION

Approving the leave:

LEAVE TAKE ACTION

Choose your option

Choose your option

Approved

Not Approved

SUBMIT

LEAVE DETAILS					
LEAVE DETAILS					
SUCCESS : Leave updated Successfully					
Employee Name :	Jackie chan	Emp Id :	12	Gender :	Male
Emp Email id :	jackie@gmail.com	Emp Contact No. :	1235		
Leave Type :	Casual Leave	Leave Date . :	From 15/02/2024 to 21/02/2024	Posting Date	2024-11-19 14:21:56
Employee Leave Description :	Please grant leave				
leave Status :	Approved				
Admin Remark:	Approved, Enjoy!!				
Admin Action taken date :	2024-11-19 14:24:43				

**Admin can manage all leaves here:**



**Admin**



Dashboard



Department



Leave Type



Employees



Leave Management



All Leaves

Pending Leaves

Approved Leaves

Not Approved Leaves



Chnage Password



Log Out



## Executing all the queries in php mysql model:

Server: 127.0.0.1:8111

Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins

Recent Favorites

New  
elms\_db  
elms\_db  
elms\_db\_new  
information\_schema  
mysql  
performance\_schema  
phpmyadmin  
test

Import has been successfully finished, 30 queries executed. (elms\_db\_new.sql)

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
-- SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"; -- START TRANSACTION; -- SET time_zone = "+00:00"; -- CREATE DATABASE elms_db; -- USE elms_db; -- -- -- Database: `elms_db` -- --  
-- SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Error: #1046 No database selected

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
START TRANSACTION;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Error: #1046 No database selected

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
SET time_zone = "+00:00";
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Error: #1046 No database selected

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

Console

✓ 1 row inserted. (Query took 0.0026 seconds.)

```
INSERT INTO `tblemployees` (`id`, `EmpId`, `FirstName`, `LastName`, `EmailId`, `Password`, `Gender`, `Dob`, `Department`, `Address`, `City`, `Country`, `Phonenumber`, `Status`, `RegDate`) VALUES (1, '6231415', 'Mark', 'Cooper', 'mcooper@gmail.com', 'c7162ff89c647f444fcaa5c635dac8c3', 'Male', '15 July, 1994', 'Marketing Department', 'There', 'Here', 'Philippines', '0912345678', 1, '2022-06-03 06:50:24');
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0099 seconds.)

```
-- Table structure for `tblleaves` CREATE TABLE `tblleaves` ( `id` int(11) NOT NULL, `LeaveType` varchar(110) NOT NULL, `ToDate` varchar(120) NOT NULL, `FromDate` varchar(120) NOT NULL, `Description` mediumtext NOT NULL, `PostingDate` timestamp NOT NULL DEFAULT current_timestamp(), `AdminRemark` mediumtext DEFAULT NULL, `AdminRemarkDate` varchar(120) DEFAULT NULL, `Status` int(1) NOT NULL, `IsRead` int(1) NOT NULL, `empid` int(11) DEFAULT NULL ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 1 row inserted. (Query took 0.0025 seconds.)

```
INSERT INTO `tblleaves` (`id`, `LeaveType`, `ToDate`, `FromDate`, `Description`, `PostingDate`, `AdminRemark`, `AdminRemarkDate`, `Status`, `IsRead`, `empid`) VALUES (1, 'Casual Leave', '07/02/2022', '06/02/2022', 'Sample only', '2022-06-03 07:28:07', 'Sample Approve Remarks', '2022-06-03 13:00:50', 1, 1, 3);
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0092 seconds.)

```
-- Table structure for `tblleavetype` CREATE TABLE `tblleavetype` ( `id` int(11) NOT NULL, `LeaveType` varchar(200) DEFAULT NULL, `Description` mediumtext DEFAULT NULL, `CreationDate` timestamp NOT NULL DEFAULT current_timestamp() ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0027 seconds.)

```
CREATE FUNCTION GetLeaveBalance(emp_id INT) RETURNS INT BEGIN DECLARE total_leaves INT DEFAULT 20; DECLARE used_leaves INT; SELECT COUNT(*) INTO used_leaves FROM tblleaves WHERE empid = emp_id AND Status = 1; RETURN total_leaves - used_leaves; END;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

⚠ Error: #1046 No database selected

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0034 seconds.)

```
CREATE FUNCTION GetDepartmentName(dept_id INT) RETURNS VARCHAR(150) BEGIN DECLARE dept_name VARCHAR(150); SELECT DepartmentName INTO dept_name FROM tbldepartments WHERE id = dept_id; RETURN dept_name; END;
```

[ [Edit inline](#) ] [ [Edit](#) ] [ [Create PHP code](#) ]

⚠ Error: #1046 No database selected

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

```
COMMIT;
```

## Explanation of each query:

### 1.Setup:

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
START TRANSACTION;
```

```
SET time_zone = "+00:00";
```

```
CREATE DATABASE elms_db;
```

```
USE elms_db;
```

- `SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";`: Sets SQL mode so that 0 can be inserted into an AUTO\_INCREMENT column if desired.
- `START TRANSACTION;`: Begins a transaction to ensure that changes are grouped and committed together.
- `SET time_zone = "+00:00";`: Sets the default timezone to UTC.
- `CREATE DATABASE elms_db;` and `USE elms_db;`: Creates and selects the elms\_db database for subsequent operations.

### 2.Creating admin Table:

```
CREATE TABLE `admin` (  
  `id` int(11) NOT NULL,  
  `UserName` varchar(100) NOT NULL,  
  `Password` varchar(100) NOT NULL,  
  `updateDate` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON UPDATE  
  current_timestamp())
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- id: Integer primary key, unique identifier for each admin.
- UserName: Stores the admin username.
- Password: Stores a hashed password.
- updationDate: Timestamp that updates to the current time whenever the record is modified.

```
INSERT INTO `admin` (`id`, `UserName`, `Password`, `updationDate`) VALUES (1, 'admin',  
'0192023a7bbd73250516f069df18b500', '2022-06-03 07:00:14');
```

- Inserts a sample admin user with hashed password and update timestamp.

### 3.Creating tbldepartments Table:

```
CREATE TABLE `tbldepartments` (
```

```
  `id` int(11) NOT NULL,
```

```
  `DepartmentName` varchar(150) DEFAULT NULL,
```

```
  `DepartmentShortName` varchar(100) NOT NULL,
```

```
  `DepartmentCode` varchar(50) DEFAULT NULL,
```

```
  `CreationDate` timestamp NULL DEFAULT current_timestamp()
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- id: Primary key for the department.
- DepartmentName: Name of the department.
- DepartmentShortName: Short name for department identification.
- DepartmentCode: Code for department identification.
- CreationDate: Timestamp for when the record was created.

```
INSERT INTO `tbldepartments` (`id`, `DepartmentName`, `DepartmentShortName`,  
`DepartmentCode`, `CreationDate`) VALUES
```

```
(1, 'Human Resource', 'HR', 'HRD', '2017-11-01 07:16:25'),
```

```
(2, 'Information Technology', 'IT', 'ITD', '2017-11-01 07:19:37'),
```

```
(3, 'Marketing Department', 'Marketing - updated', 'MktgD', '2022-06-03 05:29:06');
```

- Inserting values to the tbldepartments

#### 4. Creating tbemployees Table:

```
CREATE TABLE `tbemployees` (  
  `id` int(11) NOT NULL,  
  ...  
  `RegDate` timestamp NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- Columns: Various fields store employee data such as EmpId, FirstName, LastName, EmailId, Password, etc.
- RegDate: Timestamp for employee registration.

#### 5. tbleaves Table:

```
CREATE TABLE `tbleaves` (  
  `id` int(11) NOT NULL,  
  ...  
  `IsRead` int(1) NOT NULL,  
  `empid` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- Columns: Stores details of leave applications such as LeaveType, ToDate, FromDate, Description, etc.
- Status: Leave status (e.g., approved, pending).
- IsRead: Indicator if leave has been read by admin.
- empid: Foreign key linking to the tbemployees table.

#### 6. tbleavetype Table:

```
CREATE TABLE `tbleavetype` (  
  `id` int(11) NOT NULL,  
  ...  
  `CreationDate` timestamp NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- id: Primary key for each leave type.
- LeaveType and Description: Type and description of leave.

- CreationDate: Timestamp for when the leave type was created.

## 7. Indexes and Primary Keys:

```
ALTER TABLE `admin` ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `tbldepartments` ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `tblemployees` ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `tblleaves` ADD PRIMARY KEY (`id`), ADD KEY `UserEmail` (`empid`);
```

```
ALTER TABLE `tblleavetype` ADD PRIMARY KEY (`id`);
```

- Adds primary keys to each table and an additional index (UserEmail) on the empid column in tblleaves to speed up queries that filter by employee ID.

## 8. Auto-increment Settings:

```
ALTER TABLE `admin` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

```
ALTER TABLE `tbldepartments` MODIFY `id` int(11) NOT NULL AUTO_INCREMENT,  
AUTO_INCREMENT=6;
```

- Sets each table's primary key to auto-increment and specifies the starting value for the next insert.

## 9. Creating a Procedure to add employee details:

```
DELIMITER //
```

```
CREATE PROCEDURE AddEmployee(
```

```
    IN emp_id VARCHAR(100),
```

```
    IN first_name VARCHAR(150),
```

```
    IN last_name VARCHAR(150),
```

```
    IN email VARCHAR(200),
```

```
    IN password TEXT,
```

```
    IN gender VARCHAR(100),
```

```
    IN dob VARCHAR(100),
```

```
    IN department VARCHAR(255),
```

```
    IN address VARCHAR(255),
```

```
    IN city VARCHAR(200),
```

```
    IN country VARCHAR(150),
```

```

    IN phone CHAR(11),
    IN status INT
)
BEGIN
    INSERT INTO tblemployees (EmpId, FirstName, LastName, EmailId, Password, Gender, Dob,
    Department, Address, City, Country, Phonenummer, Status)
    VALUES (emp_id, first_name, last_name, email, password, gender, dob, department, address, city,
    country, phone, status);
END //
DELIMITER ;

```

- Inserts a new employee into tblemployees with given data.

### **10.Procedure to Request a leave:**

-- Procedure to request leave

```

DELIMITER //
CREATE PROCEDURE RequestLeave(
    IN leave_type VARCHAR(110),
    IN to_date VARCHAR(120),
    IN from_date VARCHAR(120),
    IN description MEDIUMTEXT,
    IN status INT,
    IN is_read INT,
    IN emp_id INT
)
BEGIN
    INSERT INTO tblleaves (LeaveType, ToDate, FromDate, Description, Status, IsRead, empid)
    VALUES (leave_type, to_date, from_date, description, status, is_read, emp_id);
END //
DELIMITER ;

```

- Inserts a new leave request for an employee into tblleaves.

### **11.Function to calculate the leave balance of an employee:**

```
DELIMITER //

CREATE FUNCTION GetLeaveBalance(emp_id INT) RETURNS INT

BEGIN

    DECLARE total_leaves INT DEFAULT 20;

    DECLARE used_leaves INT;

    SELECT COUNT(*) INTO used_leaves

    FROM tblleaves

    WHERE empid = emp_id AND Status = 1;

    RETURN total_leaves - used_leaves;

END //

DELIMITER ;
```

- Calculates an employee's remaining leave balance, assuming a total of 20 leaves and counting the number of approved (Status = 1) leave records.

### **12. Function to get department name by ID**

```
DELIMITER //

CREATE FUNCTION GetDepartmentName(dept_id INT) RETURNS VARCHAR(150)

BEGIN

    DECLARE dept_name VARCHAR(150);

    SELECT DepartmentName INTO dept_name

    FROM tbldepartments

    WHERE id = dept_id;

    RETURN dept_name;
```

END //

- Retrieves the name of a department based on the given department ID.

### 13. COMMIT;

- **COMMIT saves all changes** in the transaction permanently. Finalizes all changes made within the transaction, ensuring they are saved to the database.

**GitHub repository link:** <https://github.com/Punith777/Employee-leave-management-system>