**[ATTENDANCE TRACKING SYSTEM]**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

Ramya reddy|AP22110010209

Jalapathi Benila|AP22110010241

Radha Kalyani|AP22110010244

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[May, 2024]**





**Abstract**:

The Attendance Record System is being developed to streamline and automate the process of tracking employee attendance within our organization. As our company continues to grow, manually monitoring and recording attendance data has become increasingly inefficient and error-prone. Implementing an automated system will not only save time and resources but also ensure accuracy and reliability in maintaining attendance records. The primary objective of the Attendance Record System is to provide a centralized platform for employees to clock in and out, record leave requests, and track holidays. By digitizing these processes, we aim to eliminate manual paperwork, reduce administrative overhead, and improve overall efficiency. Additionally, the system will generate comprehensive reports and analytics to help management make informed decisions regarding workforce management and resource allocation.

**Aim**:

To create a centralized database for storing and retrieving attendance records of students.

**Entity:**

**Employee Attendance Tracking :**  Employees will have access to a user-friendly interface to record their attendance by clocking in and out electronically. The system will capture timestamps for each attendance event, providing real-time visibility into employee work hours.

**Leave Management :** Employees can submit leave requests through the system, specifying the start and end dates of their leave and the type of leave (e.g., vacation, sick leave). Managers will have the ability to approve or reject leave requests, ensuring proper staffing levels are maintained at all times.

**Holiday Calendar :** The system will maintain a centralized holiday calendar, listing all observed holidays for the organization. This calendar will be accessible to all employees, allowing them to plan their work schedules accordingly.

**Reporting and Analytics :** The system will generate various reports and analytics, including attendance summaries, leave balances, and trends analysis. These insights will enable management to identify patterns, address potential issues, and optimize workforce management strategies.

**Attributes**

Employees:

* employee\_id (Primary Key)
* employee\_name
* department\_id (Foreign Key referencing Departments)

Departments:

* department\_id (Primary Key)
* department\_name

Attendance:

* attendance\_id (Primary Key)
* employee\_id (Foreign Key referencing Employees)
* attendance\_date
* clock\_in\_time
* clock\_out\_time

Holidays:

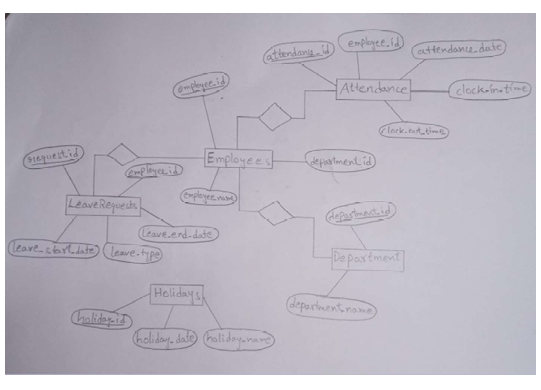
* holiday\_id (Primary Key)
* holiday\_date
* holiday\_name

LeaveRequests:

* request\_id (Primary Key)
* employee\_id (Foreign Key referencing Employees)
* leave\_start\_date
* leave\_end\_date
* leave\_type

**E-R Model:**

ER diagram here



In an Entity-Relationship (ER) diagram, each table represents an entity, and relationships between entities are depicted through connecting lines. Here's a description of the entities represented by the tables in the provided schema:

1. Employees: -

This entity represents individual employees within an organization. Each employee has a unique `employee\_id`, a name (`employee\_name`), and belongs to a specific department (`department\_id`). The relationship between the `Employees` and `Departments` tables is established through a foreign key constraint on the `department\_id` column, indicating that each employee is associated with exactly one department.

2. Departments:

- This entity represents departments or functional units within the organization. Each department has a unique `department\_id` and a name (`department\_name`). The `Departments` table serves as a lookup table for department information and is referenced by the `Employees` table through the `department\_id` foreign key.

3. Attendance:

- This entity tracks the attendance records of employees. Each attendance record is identified by a unique `attendance\_id` and includes details such as the `employee\_id` of the employee who attended, the `attendance\_date`, `clock\_in\_time`, and `clock\_out\_time`. The `Attendance` table is linked to the `Employees` table via the `employee\_id` foreign key, indicating that each attendance record corresponds to a specific employee.

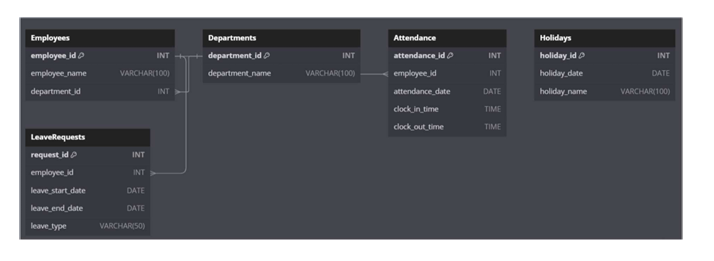
4. Holidays:

- This entity represents holidays observed by the organization. Each holiday is identified by a unique `holiday\_id` and includes the `holiday\_date` and `holiday\_name`. The `Holidays` table stores information about holidays independently of any specific employee or department.

5. LeaveRequests:

- This entity records leave requests submitted by employees. Each leave request has a unique `request\_id` and includes details such as the `employee\_id` of the requesting employee, `leave\_start\_date`, `leave\_end\_date`, and `leave\_type` (e.g., vacation, sick leave). The `LeaveRequests` table is linked to the `Employees` table via the `employee\_id` foreign key, indicating the association between leave requests and employees.

**Relational Diagram**:



**1. Employees:**

- This table stores information about employees in an organization.

- employee\_id: A unique identifier for each employee (Primary Key).

- employee\_name: The name of the employee.

- department\_id: The identifier of the department to which the employee belongs (Foreign Key referencing Departments).

- This table allows tracking of basic employee details such as name and department affiliation.

**2. Departments: -**

This table contains information about different departments within the organization.

- department\_id: A unique identifier for each department (Primary Key).

- department\_name: The name of the department.

- This table facilitates organizing employees into different functional units or teams.

**3. Attendance:**

- This table tracks the attendance records of employees.

- attendance\_id: A unique identifier for each attendance record (Primary Key).

- employee\_id: The identifier of the employee whose attendance is being recorded (Foreign Key referencing Employees).

- attendance\_date: The date of the attendance record.

- clock\_in\_time: The time when the employee clocked in.

- clock\_out\_time: The time when the employee clocked out.

- This table helps monitor employee attendance and working hours.

**4. Holidays:**

- This table stores information about holidays.

- holiday\_id: A unique identifier for each holiday (Primary Key).

- holiday\_date: The date of the holiday.

- holiday\_name: The name or description of the holiday.

- This table assists in identifying days when employees are not expected to work due to holidays.

**5. LeaveRequests:**

- This table records leave requests submitted by employees.

- request\_id: A unique identifier for each leave request (Primary Key).

- employee\_id: The identifier of the employee who requested leave (Foreign Key referencing Employees).

- leave\_start\_date: The start date of the requested leave.

- leave\_end\_date: The end date of the requested leave.

- leave\_type: The type or reason for the leave request.

- This table helps manage and track employee absences due to various types of leave, such as vacation or sick leave.

**Normalization:**

Normalization is a process in database design aimed at minimizing redundancy and dependency issues within the database schema, leading to more efficient storage and retrieval of data. In achieving third normal form (3NF), each table in the database undergoes a series of transformations. Beginning with the first normal form (1NF), which ensures atomicity by eliminating repeating groups and multivalued attributes, the schema progresses to the second normal form (2NF), which addresses partial dependencies by ensuring each non-key attribute is fully functionally dependent on the primary key. Finally, in 3NF, the schema eliminates transitive dependencies, ensuring that each non-key attribute is dependent only on the primary key and not on other non-key attributes. Through this process, the database schema is optimized, promoting data integrity, reducing redundancy, and enhancing query performance. In the context of the provided tables, normalization to 3NF ensures that each table efficiently represents a distinct entity, such as employees, departments, attendance records, holidays, and leave requests, while maintaining data integrity and minimizing data redundancy.

**ENTITY RECORDS**



-- Employees table

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

employee\_name VARCHAR(100),

department\_id INT,

-- Add other columns as needed

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id)

);

-- Departments table

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100)

-- Add other columns as needed

);

-- Attendance table

CREATE TABLE Attendance (

attendance\_id INT PRIMARY KEY,

employee\_id INT,

attendance\_date DATE,

clock\_in\_time TIME,

clock\_out\_time TIME,

-- Add other columns as needed

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

-- Holidays table

CREATE TABLE Holidays (

holiday\_id INT PRIMARY KEY,

holiday\_date DATE,

holiday\_name VARCHAR(100)

-- Add other columns as needed

);

-- LeaveRequests table

CREATE TABLE LeaveRequests (

request\_id INT PRIMARY KEY,

employee\_id INT,

leave\_start\_date DATE,

leave\_end\_date DATE,

leave\_type VARCHAR(50)

-- Add other columns as needed

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

**Creation of Data in the tables**

-- Insert data into Employees table

INSERT INTO Employees (employee\_id, employee\_name, department\_id)

VALUES (1, 'John Doe', 1),

(2, 'Jane Smith', 2),

(3, 'Michael Johnson', 1),

(4, 'Emily Brown', 2);

-- Insert data into Departments table

INSERT INTO Departments (department\_id, department\_name)

VALUES (1, 'Engineering'),

(2, 'Marketing');

-- Insert data into Attendance table

INSERT INTO Attendance (attendance\_id, employee\_id, attendance\_date, clock\_in\_time, clock\_out\_time)

VALUES (1, 1, '2024-04-24', '09:00:00', '17:00:00'),

(2, 2, '2024-04-24', '08:30:00', '16:30:00'),

(3, 3, '2024-04-24', '09:15:00', '17:15:00');

-- Insert data into Holidays table

INSERT INTO Holidays (holiday\_id, holiday\_date, holiday\_name)

VALUES (1, '2024-01-01', 'New Year'),

(2, '2024-12-25', 'Christmas');

-- Insert data into LeaveRequests table

INSERT INTO LeaveRequests (request\_id, employee\_id, leave\_start\_date, leave\_end\_date, leave\_type)

VALUES (1, 2, '2024-05-01', '2024-05-05', 'Vacation'),

(2, 4, '2024-04-20', '2024-04-21', 'Sick Leave');

**Few sql queries on the created tables**

-- Stored Procedure: SP\_MarkAttendance

CREATE PROCEDURE SP\_MarkAttendance

@employee\_id INT,

@attendance\_date DATE,

@clock\_in\_time TIME,

@clock\_out\_time TIME

AS

BEGIN

INSERT INTO Attendance (employee\_id, attendance\_date, clock\_in\_time, clock\_out\_time)

VALUES (@employee\_id, @attendance\_date, @clock\_in\_time, @clock\_out\_time)

END;

-- Stored Procedure: SP\_RequestLeave

CREATE PROCEDURE SP\_RequestLeave

@employee\_id INT,

@leave\_start\_date DATE,

@leave\_end\_date DATE,

@leave\_type VARCHAR(50)

AS

BEGIN

INSERT INTO LeaveRequests (employee\_id, leave\_start\_date, leave\_end\_date, leave\_type)

VALUES (@employee\_id, @leave\_start\_date, @leave\_end\_date, @leave\_type)

END;

-- Trigger: TR\_CheckLeaveOnAttendanceInsert

CREATE TRIGGER TR\_CheckLeaveOnAttendanceInsert

ON Attendance

AFTER INSERT

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted i

INNER JOIN LeaveRequests l ON i.employee\_id = l.employee\_id

WHERE i.attendance\_date BETWEEN l.leave\_start\_date AND l.leave\_end\_date)

BEGIN

RAISERROR ('Cannot mark attendance on leave day.', 16, 1)

ROLLBACK TRANSACTION

END

END;

-- Trigger: TR\_CheckHolidayOnAttendanceInsert

CREATE TRIGGER TR\_CheckHolidayOnAttendanceInsert

ON Attendance

AFTER INSERT

AS

BEGIN

IF EXISTS (SELECT 1 FROM inserted i

INNER JOIN Holidays h ON i.attendance\_date = h.holiday\_date)

BEGIN

RAISERROR ('Cannot mark attendance on a holiday.', 16, 1)

ROLLBACK TRANSACTION

END

END;

**🡪VIEWS:**

CREATE VIEW Employees\_View AS

SELECT e.employee\_id, e.employee\_name, d.department\_name

FROM Employees e

JOIN Departments d ON e.department\_id = d.department\_id;

CREATE VIEW Attendance\_View AS

SELECT a.attendance\_id, e.employee\_name, a.attendance\_date, a.clock\_in\_time, a.clock\_out\_time

FROM Attendance a

JOIN Employees e ON a.employee\_id = e.employee\_id;

CREATE VIEW Holidays\_View AS

SELECT holiday\_id, holiday\_date, holiday\_name

FROM Holidays;

CREATE VIEW LeaveRequests\_View AS

SELECT lr.request\_id, e.employee\_name, lr.leave\_start\_date, lr.leave\_end\_date, lr.leave\_type

FROM LeaveRequests lr

JOIN Employees e ON lr.employee\_id = e.employee\_id;

CREATE VIEW Departments\_View AS

SELECT department\_id, department\_name

FROM Departments;