

**ASSIGNMENT – 5**

**CSA0593 – DBMS**

**NAME – S.VARSHA**

**REG NO – 192311425**

**DATE – 16:11:2024**

1. Design a database to manage employees, departments, attendance, and performance reviews.
  - Model tables for employees, departments, attendance records, and performance reviews.
  - Write stored procedures for recording attendance, conducting performance evaluations, and assigning employees to departments.
  - Implement triggers to automatically update attendance records and notify HR of frequent absences.
  - Write SQL queries to analyze employee attendance trends, performance statistics, and department productivity.

ANSWER:

CONCEPTUAL ER DIAGRAM:

- **Entities:** Employees, Departments, Attendance Records, Performance Reviews.
- **Relationships:** Employees belong to Departments, Attendance tracks daily Employee presence, Performance Reviews evaluate Employee achievements.
- **Constraints:** Foreign keys link Employees to Departments, Attendance to Employees, and Performance Reviews to Employees and Departments.

## **CODE:**

Create Departments Table

```
CREATE TABLE Departments (  
    department_id INT AUTO_INCREMENT PRIMARY KEY,  
    department_name VARCHAR(255) NOT NULL,  
    location VARCHAR(255),  
    manager_id INT,  
    FOREIGN KEY (manager_id) REFERENCES  
Employees(employee_id) -- Reference to the manager of the  
department  
);
```

-- Create Employees Table

```
CREATE TABLE Employees (  
    employee_id INT AUTO_INCREMENT PRIMARY KEY,  
    first_name VARCHAR(100) NOT NULL,  
    last_name VARCHAR(100) NOT NULL,  
    date_of_birth DATE NOT NULL,  
    gender CHAR(1), -- M/F/O  
    phone_number VARCHAR(20),  
    email VARCHAR(100),  
    hire_date DATE,
```

```
salary DECIMAL(10, 2),
department_id INT,
position VARCHAR(100),
FOREIGN KEY (department_id) REFERENCES
Departments(department_id) -- Reference to department
);
```

-- Create Attendance Table

```
CREATE TABLE Attendance (
    attendance_id INT AUTO_INCREMENT PRIMARY KEY,
    employee_id INT,
    attendance_date DATE NOT NULL,
    status ENUM('Present', 'Absent', 'Sick', 'Vacation', 'Other')
NOT NULL,
    reason TEXT,
    FOREIGN KEY (employee_id) REFERENCES
Employees(employee_id) -- Reference to employee
);
```

-- Create PerformanceReviews Table

```
CREATE TABLE PerformanceReviews (
    review_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
employee_id INT,  
review_date DATE NOT NULL,  
reviewer_id INT, -- The employee who gave the review  
(could be a manager)  
performance_score DECIMAL(5, 2), -- Performance rating,  
e.g., from 1.0 to 5.0  
comments TEXT,  
FOREIGN KEY (employee_id) REFERENCES  
Employees(employee_id), -- Reference to the employee being  
reviewed  
FOREIGN KEY (reviewer_id) REFERENCES  
Employees(employee_id) -- Reference to the reviewer  
(employee giving the review)  
);
```

## **EMPLOYEES TABLE**

Column Name	Data Type	Description
employee_id	INT AUTO_INCREMENT	Primary Key, Unique identifier for each employee
first_name	VARCHAR(100)	Employee's first name
last_name	VARCHAR(100)	Employee's last name
date_of_birth	DATE	Employee's date of birth
gender	CHAR(1)	Gender (M/F/O)
phone_number	VARCHAR(20)	Employee's phone number
email	VARCHAR(100)	Employee's email address
hire_date	DATE	Date when the employee was hired
salary	DECIMAL(10, 2)	Employee's salary
department_id	INT	Foreign Key referencing Departments(department_id)
position	VARCHAR(100)	Job position/title of the employee

# DEPARTMENT TABLE

Column Name	Data Type	Description
department_id	INT AUTO_INCREMENT	Primary Key, Unique identifier for each department
department_name	VARCHAR(255)	Name of the department
location	VARCHAR(255)	Department location (e.g., Building, Floor)
manager_id	INT	Foreign Key referencing Employees(employee_id) — the department manager

# ATTENDANCE TABLE

Column Name	Data Type	Description
attendance_id	INT AUTO_INCREMENT	Primary Key, Unique identifier for each attendance record
employee_id	INT	Foreign Key referencing Employees(employee_id)
attendance_date	DATE	Date of attendance
status	ENUM('Present', 'Absent', 'Sick', 'Vacation', 'Other')	Employee's attendance status
reason	TEXT	Reason for absence (if applicable)

# PERFORMANCE REVIEW TABLE

Column Name	Data Type	Description
review_id	INT AUTO_INCREMENT	Primary Key, Unique identifier for each performance review
employee_id	INT	Foreign Key referencing Employees(employee_id)
review_date	DATE	Date of the performance review
reviewer_id	INT	Foreign Key referencing Employees(employee_id) — the person conducting the review
performance_score	DECIMAL(5, 2)	Performance score (e.g., from 1 to 5)
comments	TEXT	Additional comments or feedback from the reviewer

## CONCLUSION

This database design captures essential details about employees, departments, attendance records, and performance reviews in an organization. Key relationships are established through foreign keys, ensuring consistency and making it easy to generate reports and insights based on employee performance, attendance trends, and departmental organization. This structure is highly scalable and can be extended further to include more data, such as project assignments, salary history, or leave records.