1. **Database Design for Human Resource Management System**

1. Employee

* employee\_id (Primary Key): Unique identifier for each employee.
* first\_name: First name of the employee.
* last\_name: Last name of the employee.
* date\_of\_birth: Date of birth of the employee.
* gender: Gender of the employee.
* hire\_date: Date when the employee was hired.
* department\_id: Identifier of the department to which the employee belongs.
* position\_id: Identifier of the position held by the employee.
* salary: Salary of the employee.

2. Department

* department\_id (Primary Key): Unique identifier for each department.
* name: Name of the department.

3. Position

* position\_id (Primary Key): Unique identifier for each position.
* title: Title of the position.
* description: Description of the position.

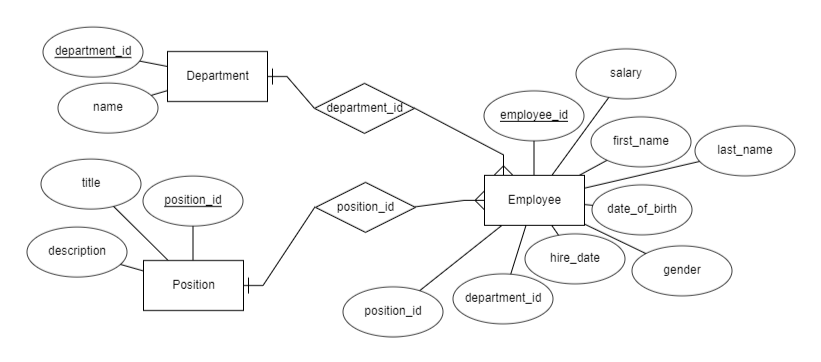
**Employee to Department Relationship**

* **Many-to-one relationship:**Many employees can belong to one department.
* **Foreign key:** department\_id in Employee table referencing department\_id in Department table.

**Employee to Position Relationship**

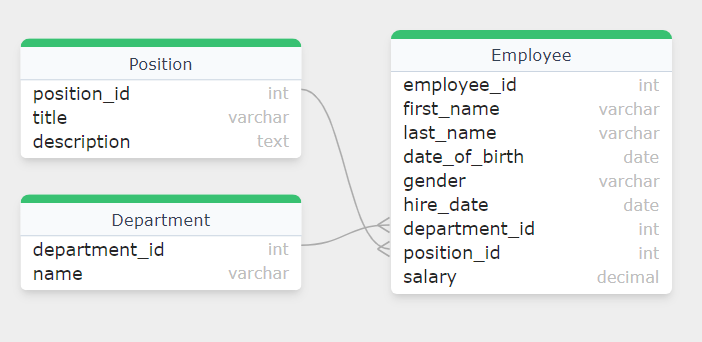
* **Many-to-one relationship:** Many employees can hold one position.
* **Foreign key:** position\_id in Employee table referencing position\_id in Position table.

**ER Diagram for Human Resource Management System**



**Entities Structures in SQL Format**

Create Employee table  
CREATE TABLE Employee (  
 employee\_id INT PRIMARY KEY,  
 first\_name VARCHAR(255) NOT NULL,  
 last\_name VARCHAR(255) NOT NULL,  
 date\_of\_birth DATE NOT NULL,  
 gender VARCHAR(10) NOT NULL,  
 hire\_date DATE NOT NULL,  
 department\_id INT NOT NULL,  
 position\_id INT NOT NULL,  
 salary DECIMAL(10, 2) NOT NULL,  
 FOREIGN KEY (department\_id) REFERENCES Department(department\_id),  
 FOREIGN KEY (position\_id) REFERENCES Position(position\_id)  
);  
  
Create Department table  
CREATE TABLE Department (  
 department\_id INT PRIMARY KEY,  
 name VARCHAR(255) NOT NULL  
);  
  
Create Position table  
CREATE TABLE Position (  
 position\_id INT PRIMARY KEY,  
 title VARCHAR(255) NOT NULL,  
 description TEXT  
);

**Database Model for Human Resource Management System**

1. **Human Resource Management System Features**

Designing an ER diagram for an HRM system involves identifying and defining the key entities and their attributes. These entities usually include **employees, departments, attendance records, leave requests, payroll records and training sessions.** The relationships between these entities such as an**employee belonging to a department or participating in training are also important.**

1. **Employee Information Management**
   * This feature allows HR departments to store and manage employee data such as **personal information**, **contact** **details**, **job history**, and **performance** **records**.
   * It helps in maintaining accurate employee records, **tracking**changes in employment **status** and **ensuring**compliance with **legal**requirements.
   * Employee information management systems include features for onboarding **new employees, managing promotions**and **transfers,** and recording **disciplinary actions**.
2. **Recruitment and Applicant Tracking**
   * This feature fasts the recruitment process by allowing HR teams to post job vacancies, review applications and track candidates progress through the hiring process.
   * It helps in creating **job postings, managing candidate databases, scheduling interviews** and making **job offers**.
   * Applicant tracking systems also provide analytics to assess the effectiveness of recruitment strategies and improve the hiring process over time.
3. **Payroll and Benefits Administration**
   * This feature automates the calculation and disbursement of employee **salaries, taxes and benefits**.
   * It helps in managing **payroll schedules, tracking employee attendance** and **leave** and ensuring compliance with **tax regulations**.
   * Payroll and benefits administration systems also provide employees with access to their **pay stubs, tax forms** and benefit information.
4. **Performance Management**
   * This feature helps organizations evaluate and improve employee performance through **goal setting, performance reviews and feedback** mechanisms.
   * It facilitates the creation and tracking of performance **goals**, conducting **performance** **appraisals** and identifying areas for employee development.
   * Performance management systems often include tools for **360**–**degree**feedback, **self**–**assessment**, and performance improvement planning.
5. **Training and Development**
   * This feature supports the **planning**, **delivery**, and **tracking**of employee **training** and **development**programs.
   * It helps in identifying training needs, scheduling training sessions and tracking employee participation and progress.
   * Training and development systems also provide employees with access to training materials, resources and online courses.

**Entities and Attributes of Human Resource Management**

**1. Employee: Represents individuals working within the organization.**

* **EmployeeID (Primary Key):** It is a Unique identifier for each employee.
* **Name:**Full name of the employee.
* **DepartmentID (Foreign Key):** References the department to which the employee belongs.
* **Position:** Job position/title held by the employee.
* **Hire Date:** This is a date when the employee was hired.

**2. Department: Represents organizational departments.**

* **DepartmentID (Primary Key):** It is a Unique identifier for each department.
* **Name:**Name of the department.
* **ManagerID (Foreign Key):** References the employee who serves as the manager of the department.

**3. Attendance: Tracks employee attendance records.**

* **AttendanceID (Primary Key):** Unique identifier for each attendance record.
* **EmployeeID (Foreign Key):** References the employee associated with the attendance record.
* **Date:**Date of the attendance record.
* **Time In:** Time when the employee clocked in.
* **Time Out:** Time when the employee clocked out.

**4. Leave: Manages employee leave requests.**

* **LeaveID (Primary Key):** Unique identifier for each leave request.
* **EmployeeID (Foreign Key):** References the employee requesting the leave.
* **StartDate:** Start date of the leave request.
* **EndDate:** End date of the leave request.

**5. Payroll: Handles employee salary and payment information.**

* **PayrollID (Primary Key):** Unique identifier for each payroll record.
* **Salary:** Amount of salary earned by the employee.
* **NetPay:** Net amount received by the employee after deductions and bonuses.

**6. Training: Manages employee training sessions.**

* **TrainingID (Primary Key):** Unique identifier for each training session.
* **Title:**Title or name of the training session.
* **Date:**Date of the training session.

**Relationship Between These Entities**

**1. Employee – Department Relationship**

* One-to-many relationship: An employee belongs to only one department but a department can have multiple employees.
* Foreign key: DepartmentID in the Employee table references DepartmentID in the Department table.

**2. Department – Employee (Manager) Relationship**

* One-to-one relationship: Each department has only one manager and each manager manages only one department.
* Foreign key: ManagerID in the Department table references EmployeeID in the Employee table.

**3. Employee – Attendance Relationship**

* One-to-many relationship: An employee can have multiple attendance records but each attendance record belongs to only one employee.
* Foreign key: EmployeeID in the Attendance table references EmployeeID in the Employee table.

**4. Employee – Leave Relationship**

* One-to-many relationship: An employee can have multiple leave requests but each leave request is made by only one employee.
* Foreign key: EmployeeID in the Leave table references EmployeeID in the Employee table.

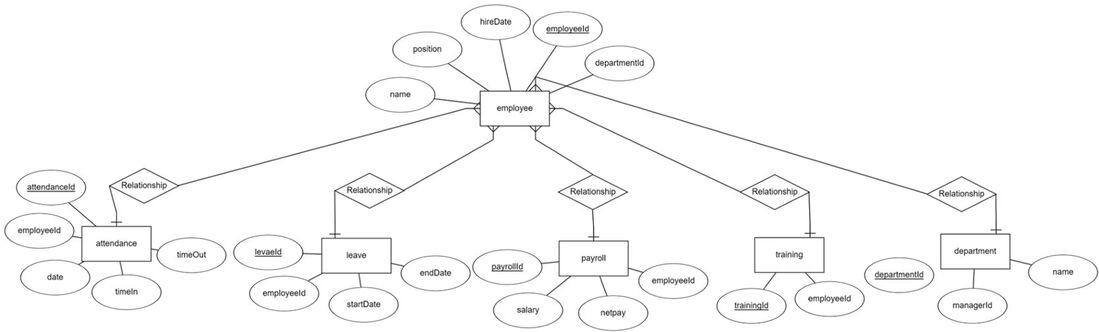
**5. Employee – Payroll Relationship**

* One-to-many relationship: An employee can have multiple payroll records but each payroll record belongs to only one employee.
* Foreign key: EmployeeID in the Payroll table references EmployeeID in the Employee table.

**6. Employee – Training Relationship**

* Many-to-many relationship: An employee can participate in multiple training sessions and each training session can have multiple employees.
* Junction table: To implement the many-to-many relationship a junction table with EmployeeID and TrainingID is used to track employee participation in training sessions.

**ER diagram**



**Creation of Tables**

**1.Employee Table**

CREATE TABLE Employee (

EmployeeID INT PRIMARY KEY,

Name VARCHAR(100),

DepartmentID INT,

Position VARCHAR(100),

HireDate DATE,

FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)

);

**2. Department Table**

CREATE TABLE Department (

DepartmentID INT PRIMARY KEY,

Name VARCHAR(100),

ManagerID INT,

FOREIGN KEY (ManagerID) REFERENCES Employee(EmployeeID)

);

**3. Attendance Table**

CREATE TABLE Attendance (

AttendanceID INT PRIMARY KEY,

EmployeeID INT,

Date DATE,

TimeIn TIME,

TimeOut TIME,

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

);

**4. Leave Table**

CREATE TABLE Leave (

LeaveID INT PRIMARY KEY,

EmployeeID INT,

StartDate DATE,

EndDate DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

);

* 1. **Payroll Table**

CREATE TABLE Payroll (

PayrollID INT PRIMARY KEY,

EmployeeID INT,

Salary DECIMAL(10, 2),

NetPay DECIMAL(10, 2),

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

);

**6. Training Table**

CREATE TABLE Training (

TrainingID INT PRIMARY KEY,

EmployeeID INT,

Title VARCHAR(100),

Date DATE,

FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)

);

**Job Posting**

* 1. **JobPostings Table**

CREATE TABLE JobPostings (

JobID INT PRIMARY KEY,

Title VARCHAR(100),

DepartmentID INT,

Description TEXT,

PostDate DATE,

ClosingDate DATE,

FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)

);

* 1. **Candidate Table**

CREATE TABLE Candidate (

CandidateID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

PhoneNumber VARCHAR(20),

Resume TEXT, -- Stores resume details or file location

ApplicationDate DATE

);

* 1. **ApplicationTracking Table**

CREATE TABLE ApplicationTracking (

TrackingID INT PRIMARY KEY,

CandidateID INT,

JobID INT,

ApplicationStatus VARCHAR(50), -- e.g., "Applied", "Interviewed", "Offered", "Hired"

InterviewDate DATE,

OfferDate DATE,

HireDate DATE,

FOREIGN KEY (CandidateID) REFERENCES Candidate(CandidateID),

FOREIGN KEY (JobID) REFERENCES JobPostings(JobID)

);

* 1. **InterviewSchedule Table**

CREATE TABLE InterviewSchedule (

ScheduleID INT PRIMARY KEY,

CandidateID INT,

JobID INT,

Interviewer VARCHAR(100),

InterviewDate DATE,

InterviewTime TIME,

FOREIGN KEY (CandidateID) REFERENCES Candidate(CandidateID),

FOREIGN KEY (JobID) REFERENCES JobPostings(JobID)

);

* 1. **Analytics Table** (Optional)

CREATE TABLE RecruitmentAnalytics (

AnalyticsID INT PRIMARY KEY,

JobID INT,

TotalApplications INT,

InterviewsScheduled INT,

OffersMade INT,

CandidatesHired INT,

FOREIGN KEY (JobID) REFERENCES JobPostings(JobID) );