1. **Department**
   * DepartmentID (Primary Key)
   * DepartmentName
2. **Role**
   * RoleID (Primary Key)
   * RoleName
3. **Employee**
   * EmployeeID (Primary Key)
   * Name
   * DepartmentID (Foreign Key)
   * RoleID (Foreign Key)
4. **Contract**
   * ContractID (Primary Key)
   * EmployeeID (Foreign Key, Unique)
   * ContractDetails
5. **LeaveRequest**
   * LeaveRequestID (Primary Key)
   * EmployeeID (Foreign Key)
   * LeaveDate
   * Reason
6. **Salary**
   * SalaryID (Primary Key)
   * EmployeeID (Foreign Key, Unique)
   * Amount
7. **Attendance**
   * AttendanceID (Primary Key)
   * EmployeeID (Foreign Key)
   * AttendanceDate
   * Status (e.g., Present, Absent, Leave)
8. **PerformanceReview**
   * ReviewID (Primary Key)
   * EmployeeID (Foreign Key)
   * ReviewDate
   * Comments
9. **JobPosting**
   * JobPostingID (Primary Key)
   * JobTitle
   * JobDescription
10. **Recruitment**
    * RecruitmentID (Primary Key)
    * JobPostingID (Foreign Key)
    * RecruitmentStage
11. **Applicant**
    * ApplicantID (Primary Key)
    * Name
    * JobPostingID (Foreign Key)
12. **Interview**
    * InterviewID (Primary Key)
    * ApplicantID (Foreign Key)
    * InterviewDate
    * Comments
13. **Task**
    * TaskID (Primary Key)
    * TaskName
14. **Employee\_Task** (Junction Table)
    * EmployeeID (Foreign Key)
    * TaskID (Foreign Key)
15. **Notification**
    * NotificationID (Primary Key)
    * EmployeeID (Foreign Key)
    * Message
    * SentDate
16. **Payroll**
    * PayrollID (Primary Key)
    * SalaryID (Foreign Key, Unique)
    * PaymentDate

**Relationships**

* **Employee** to **Department**: Many-to-One
* **Employee** to **Role**: Many-to-One
* **Employee** to **Contract**: One-to-One
* **Employee** to **LeaveRequest**: One-to-Many
* **Employee** to **Salary**: One-to-One
* **Employee** to **Attendance**: One-to-Many
* **Employee** to **PerformanceReview**: One-to-Many
* **JobPosting** to **Recruitment**: One-to-Many
* **JobPosting** to **Applicant**: One-to-Many
* **Applicant** to **Interview**: One-to-Many
* **Employee** to **Task**: Many-to-Many (via Employee\_Task)
* **Employee** to **Notification**: One-to-Many
* **Salary** to **Payroll**: One-to-One

I'll create a visual ER diagram based on this information. Give me a moment.

exaplain relations in deatailed manner

Absolutely, let's dive into the detailed relationships between the entities in the HR management system:

**Detailed Relationships**

1. **Employee** to **Department**:
   * **Many-to-One Relationship**: Each department can have multiple employees, but each employee is assigned to only one department.
   * **Foreign Key**: The DepartmentID in the Employee entity references DepartmentID in the Department entity, indicating which department the employee belongs to.
2. **Employee** to **Role**:
   * **Many-to-One Relationship**: Each role can be assigned to multiple employees, but each employee has only one role.
   * **Foreign Key**: The RoleID in the Employee entity references RoleID in the Role entity, indicating the role of the employee.
3. **Employee** to **Contract**:
   * **One-to-One Relationship**: Each employee has one unique contract, and each contract is associated with one employee.
   * **Foreign Key**: The EmployeeID in the Contract entity is both a foreign key and unique, ensuring each contract is linked to one specific employee.
4. **Employee** to **LeaveRequest**:
   * **One-to-Many Relationship**: Each employee can submit multiple leave requests, but each leave request is associated with only one employee.
   * **Foreign Key**: The EmployeeID in the LeaveRequest entity references EmployeeID in the Employee entity, indicating which employee made the leave request.
5. **Employee** to **Salary**:
   * **One-to-One Relationship**: Each employee has one salary record, and each salary record is associated with one employee.
   * **Foreign Key**: The EmployeeID in the Salary entity is both a foreign key and unique, ensuring each salary record is linked to one specific employee.
6. **Employee** to **Attendance**:
   * **One-to-Many Relationship**: Each employee can have multiple attendance records, but each attendance record is associated with only one employee.
   * **Foreign Key**: The EmployeeID in the Attendance entity references EmployeeID in the Employee entity, indicating which employee the attendance record belongs to.
7. **Employee** to **PerformanceReview**:
   * **One-to-Many Relationship**: Each employee can have multiple performance reviews, but each performance review is associated with only one employee.
   * **Foreign Key**: The EmployeeID in the PerformanceReview entity references EmployeeID in the Employee entity, indicating which employee the review is for.
8. **JobPosting** to **Recruitment**:
   * **One-to-Many Relationship**: Each job posting can be part of multiple recruitment stages, but each recruitment stage is associated with only one job posting.
   * **Foreign Key**: The JobPostingID in the Recruitment entity references JobPostingID in the JobPosting entity, indicating the job posting related to the recruitment stage.
9. **JobPosting** to **Applicant**:
   * **One-to-Many Relationship**: Each job posting can have multiple applicants, but each applicant applies to only one job posting.
   * **Foreign Key**: The JobPostingID in the Applicant entity references JobPostingID in the JobPosting entity, indicating the job posting the applicant is applying to.
10. **Applicant** to **Interview**:
    * **One-to-Many Relationship**: Each applicant can have multiple interviews, but each interview is associated with only one applicant.
    * **Foreign Key**: The ApplicantID in the Interview entity references ApplicantID in the Applicant entity, indicating which applicant the interview is for.
11. **Employee** to **Task**:
    * **Many-to-Many Relationship**: Each employee can be assigned multiple tasks, and each task can be assigned to multiple employees.
    * **Junction Table**: The Employee\_Task entity has EmployeeID and TaskID as foreign keys, representing the many-to-many relationship between employees and tasks.
12. **Employee** to **Notification**:
    * **One-to-Many Relationship**: Each employee can receive multiple notifications, but each notification is associated with only one employee.
    * **Foreign Key**: The EmployeeID in the Notification entity references EmployeeID in the Employee entity, indicating which employee the notification is for.
13. **Salary** to **Payroll**:
    * **One-to-One Relationship**: Each salary record has one corresponding payroll entry, and each payroll entry is associated with one salary record.
    * **Foreign Key**: The SalaryID in the Payroll entity is both a foreign key and unique, ensuring each payroll entry is linked to one specific salary record.

**Visual Representation**

I'll generate a visual ER diagram based on this detailed explanation of the relationships. Please give me a moment.

Message Copilot