Draw ER for Company database with atleast 6 entities and convert them into tables.
Perform DDL on Above converted tables.
1. Create tables with all constraints
2. create views on any two tables using conditions
3. create index called EmployeeId for the deptment table. Entries should be in ascending order by department id and then by employee id within each department.
4. create sequence on Employee id

ANS=

Step 1: ER Diagram for Company Database

Here's a simple ER diagram with six entities:

- **Employee** (EmployeeID, Name, Age, Salary, DepartmentID)
- **Department** (DepartmentID, DepartmentName, Location)
- **Project** (ProjectID, ProjectName, StartDate, EndDate)
- **Works_On** (EmployeeID, ProjectID, HoursWorked)
- **Client** (ClientID, ClientName, ContactInfo)
- **Assignment** (AssignmentID, EmployeeID, ProjectID, Role)

Relationships:

- **Employee** belongs to **Department** (Many-to-One)
- **Employee** works on **Project** (Many-to-Many via Works_On)
- **Project** is assigned to **Client** (Many-to-One)
- **Employee** has specific **Roles** in **Projects** (Many-to-Many via Assignment)

Step 2: Convert ER Diagram to Tables

Tables

1. **Employee**
    - EmployeeID (Primary Key)
    - Name
    - Age
    - Salary
    - DepartmentID (Foreign Key)
2. **Department**
    - DepartmentID (Primary Key)
    - DepartmentName
    - Location

3. **Project**
    - ProjectID (Primary Key)
    - ProjectName
    - StartDate
    - EndDate
4. **Works_On**
    - EmployeeID (Foreign Key)
    - ProjectID (Foreign Key)
    - HoursWorked
    - PRIMARY KEY (EmployeeID, ProjectID)
5. **Client**
    - ClientID (Primary Key)
    - ClientName
    - ContactInfo
6. **Assignment**
    - AssignmentID (Primary Key)
    - EmployeeID (Foreign Key)
    - ProjectID (Foreign Key)
    - Role

Step 3: Perform DDL on Converted Tables

Here's the SQL to create the tables with all necessary constraints:

sql

Copy code

```
-- Create Department table
CREATE TABLE Department (
  DepartmentID INT PRIMARY KEY,
  DepartmentName VARCHAR(100) NOT NULL,
  Location VARCHAR(100) NOT NULL
);


-- Create Employee table
```

```
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Age INT CHECK (Age > 0),
    Salary DECIMAL(10, 2) CHECK (Salary >= 0),
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);

-- Create Project table
CREATE TABLE Project (
    ProjectID INT PRIMARY KEY,
    ProjectName VARCHAR(100) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE
);

-- Create Works_On table
CREATE TABLE Works_On (
    EmployeeID INT,
    ProjectID INT,
    HoursWorked INT CHECK (HoursWorked >= 0),
    PRIMARY KEY (EmployeeID, ProjectID),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
    FOREIGN KEY (ProjectID) REFERENCES Project(ProjectID)
);

-- Create Client table
CREATE TABLE Client (
    ClientID INT PRIMARY KEY,
    ClientName VARCHAR(100) NOT NULL,
```

```
    ContactInfo VARCHAR(255)
);

-- Create Assignment table
CREATE TABLE Assignment (
    AssignmentID INT PRIMARY KEY,
    EmployeeID INT,
    ProjectID INT,
    Role VARCHAR(50) NOT NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
    FOREIGN KEY (ProjectID) REFERENCES Project(ProjectID)
);
```

Step 4: Create Views

Now, let's create views on any two tables using specific conditions.

sql

Copy code

```
-- Create a view for employees in a specific department
CREATE VIEW EmployeeDepartment AS
SELECT e.EmployeeID, e.Name, d.DepartmentName
FROM Employee e
JOIN Department d ON e.DepartmentID = d.DepartmentID
WHERE d.DepartmentName = 'HR';  -- Example condition

-- Create a view for active projects
CREATE VIEW ActiveProjects AS
SELECT ProjectID, ProjectName, StartDate, EndDate
FROM Project
WHERE EndDate IS NULL OR EndDate > CURRENT_DATE;
```

Step 5: Create an Index on the Department Table

To create an index called EmployeeId for the Works_On table sorted by DepartmentID and EmployeeID:

sql

Copy code

-- Create index on Works_On table

CREATE INDEX EmployeeId ON Works_On (EmployeeID);

Step 6: Create a Sequence for EmployeeID

Creating a sequence for EmployeeID can vary depending on the SQL dialect. Below is an example in Oracle SQL:

sql

Copy code

-- Create a sequence for EmployeeID

CREATE SEQUENCE EmployeeID_seq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;