Practical 1

Task3

Draw EER Diagram and Relational Model for Bank Management System

Scenario Description:
A bank needs a system to manage its customers, accounts, transactions, loans, and employees efficiently. The system must support the following:

1. Customers who hold various types of accounts and apply for loans.
2. Different types of accounts, such as savings and checking accounts, with unique features.
3. Transactions involving deposits, withdrawals, and transfers.
4. Loans issued to customers and their repayment schedules.
5. Employees managing customer services and bank operations.

Entities and Attributes:

1. **Customers**
   - Attributes: CustomerID (Primary Key), Name, Address, Phone, Email.
   - Multivalued Attribute: `{AlternateContactNumbers}` (customers can have multiple contact numbers).
2. **Accounts**
   - Attributes: AccountID (Primary Key), AccountType (Savings, Checking), Balance, InterestRate.
   - Generalization/Specialization:
     - `Accounts` can be specialized into `SavingsAccounts` and `CheckingAccounts`.
     - `SavingsAccounts` include attributes like `MinBalance`.
     - `CheckingAccounts` include attributes like `OverdraftLimit`.
3. **Transactions**
   - Attributes: TransactionID (Primary Key), Date, Amount, Type (Deposit, Withdrawal, Transfer).
   - Relationships: Transactions are linked to specific `Accounts` and may involve one or more `Customers`.
4. **Loans**
   - Attributes: LoanID (Primary Key), LoanAmount, InterestRate, StartDate, EndDate, Status.
   - Weak Entity: `LoanRepayment` dependent on `Loans`, with attributes like `InstallmentID`, `InstallmentDate`, and `InstallmentAmount`.
5. **Employees**

- Attributes: EmployeeID (Primary Key), Name, Role (e.g., Manager, Teller), AssignedBranch.
- Generalization/Specialization:
  - Employees may be specialized into `Managers` and `Tellers`.
  - `Managers` have attributes like `Department`.

6. **Branches**
   - Attributes: BranchID (Primary Key), Location, PhoneNumber.
   - Relationships:
     - Each `Customer` is associated with a `Branch`.
     - Employees are assigned to specific `Branches`.

Relationships:

1. **Customers Own Accounts:**
   - A customer can have multiple accounts, but each account belongs to one customer.
2. **Transactions Performed On Accounts:**
   - An account can have multiple transactions.
3. **Customers Apply for Loans:**
   - A customer can apply for multiple loans, and a loan is linked to a single customer.
4. **Employees Manage Branches:**
   - Employees are responsible for operations at their assigned branches.

Extended ER (EER) Features to Include:

1. **Specialization/Generalization:**
   - `Accounts` → `{SavingsAccounts, CheckingAccounts}`.
   - `Employees` → `{Managers, Tellers}`.
2. **Aggregation:**
   - Create an aggregated entity `LoanManagement` that represents the interaction between `Customers`, `Loans`, and `LoanRepayment`.
3. **Weak Entities:**
   - `LoanRepayment` is a weak entity dependent on `Loans` with attributes like `InstallmentDate` and `InstallmentAmount`.
4. **Multivalued Attributes:**
   - `{AlternateContactNumbers}` for `Customers`.
5. **Ternary Relationship:**
   - A `Transaction` might involve multiple parties, such as `SourceAccount`, `DestinationAccount`, and `Customer`.

Steps to Draw the EER Diagram:

1. **Identify Core Entities:**
   - Define `Customers`, `Accounts`, `Transactions`, `Loans`, `Employees`, and `Branches`.
2. **Define Relationships:**
   - Use relationships like "Owns" between `Customers` and `Accounts`, or "Processes" between `Employees` and `Transactions`.
3. **Incorporate Specialization/Generalization:**
   - Represent hierarchical relationships like `Accounts` (specialized into `SavingsAccounts` and `CheckingAccounts`) and `Employees`.
4. **Add Aggregation:**
   - Use `LoanManagement` to group relationships between `Loans`, `Customers`, and `LoanRepayment`.
5. **Include Weak Entities:**
   - Show the dependency of `LoanRepayment` on `Loans` using a double rectangle.
6. **Highlight Extended Features:**
   - Use notation for multivalued attributes and derived attributes where applicable.