A Report

ON

"Database Project Theory"

"Banking Database"

Representors

Group 3

Muhammad Uneeb Khan 243729

Muhammad Hamza Khan 243774

Muhammad Anas Akram 243766

Under the Guidance of

Sir Saqib Imran

Department of Computer Science Air University Multan

TABLE OF CONTENT:

Contents

Banking Database

1) Case Type Scenario:

The aim of this case study is to design and develop a **relational database** for a bank to manage the information related to customers, accounts, transactions, loans, and employees. The system should support the basic operations of banking, such as creating and managing customer accounts, handling deposits and withdrawals, managing loans, and tracking employee responsibilities.

Business Scenario

Alfalah Bank is a commercial bank that operates across multiple cities. The bank offers a wide range of services including:

- Customer account management (savings, current, fixed deposit)
- Transaction processing (deposits, withdrawals, transfers)
- Loan services (home, personal, education)
- Employee and branch operations

Each **customer** may:

- Have multiple accounts (e.g., savings, current, FD)
- Take multiple loans of different types
- Perform many transactions across accounts

Each **branch** has:

- A unique code
- A team of employees including clerks, cashiers, and loan officers

The bank tracks:

- Customer account balances
- Transaction histories
- Loan disbursements and repayments
- One employee manages one branch.
- Each employee works in only one branch but may manage multiple customer services.

2) Entities Names:

Following are the tables used in the Bank Management System database:

1. **CUSTOMER**
   Stores details about the customers of the bank. Includes customer ID, name, and phone number.

2. **BRANCH**
   Contains information about all branches of the bank like branch ID and branch name.

3. **EMPLOYEE**
   Holds details about bank employees including employee ID, name, role (e.g., manager, clerk), and the branch they work at.
   Constraint: Employee ID is unique. Branch ID must exist in the BRANCH table.

4. **ACCOUNT**
   Stores account details for customers such as account number, type (savings/current), balance, and the branch it's held at.
   Constraint: Account number is unique. Customer ID and Branch ID must exist.

5. **TRANSACTION**
   Records all transactions (deposit, withdrawal, etc.) made on accounts, along with transaction date and amount.
   Constraint: Transaction ID is unique. Account number must exist in the ACCOUNT table.

6. **LOAN**
   Contains loan information for customers such as loan ID, type (personal/home), and loan amount.
   Constraint: Loan ID is unique. Customer ID must exist in the CUSTOMER table.

7. **REPAYMENT**
   Stores loan repayment records including repayment ID, loan ID, amount paid, and date of payment.

8. **SERVICE**

Purpose: To store details about services the bank offers (e.g., ATM card, online banking, locker).

9. **ACCOUNT_CARD**

Purpose: Store debit card information associated with customer accounts.

10. **Transfers**

Purpose: To transfer amount from one account to another with dates.

11. Complain

Purpose: To Complain any issues the customer is facing.

12. LOAN_DISTRBUTION:

Purpose: To distribute the loan amount of the customer which takes the loan.

3) Normalized table (Upto 3NF):

Customer:

Cust_ID	Name	Phone
C101	Ali	0300-1
C102	Sara	0300-2
C103	Bilal	0300-7
C104	Hina	0300-9
C105	Ahmed	0300-8

Branch:

Branch_ID	Branch_Name
B01	Multan
B01	Multan
B02	Lahore
B02	Lahore
B03	Karachi

Employee:

Emp_ID	Emp_Name	Branch_ID	Designation
E01	Umar	B01	Manager
E02	Ayesha	B01	Cahsier
E03	Naveed	B02	Loan Officer
E03	Naveed	B02	Service Rep
E04	Farhan	B03	IT Support

Account:

Account_No	Cust_ID	Branch_ID	Account_Type	Account_Balance
A01	C101	B01	Saving	50000
A02	C101	B01	Current	20000
A03	C102	B01	Saving	30000
A04	C103	B02	Current	15000
A05	C104	B02	Saving	40000
A06	C105	B03	Current	60000
A07	C105	B03	Saving	25000

Transaction:

Txn_ID	Txn_Acc	Txn_Type	Txn_Amt
T001	A01	Deposit	10000
T002	A01	Withdraw	5000
T003	A03	Deposit	15000
T004	A04	Withdraw	3000
T005	A05	Deposit	20000
T006	A05	Withdraw	5000
T007	A06	Deposit	30000
T008	A07	Deposit	10000

Loan:

Loan_ID	Cust_ID	Loan_Type	Loan_Amount
L01	C101	Home	200000
L02	C102	Education	50000
l03	C104	Personal	80000

Repayment:

Repay_ID	Loan_ID	Amount
R001	L01	50000
R002	L01	50000
R003	L02	25000
R004	L03	40000
R005	L03	20000

Loan Distribution:

Loan_ID	Disbursement_Amount
L01	50000
L01	50000
L02	25000
L03	40000
L03	20000

Services:

Cust_ID	Service_Type
C101	ATM
C101	MobileBanking
C102	MobileBanking
C103	ATM
C104	Locker
C104	MobileBanking
C405	ATM

Complain:

Complaint_ID	Cust_ID	Description
CPN01	C101	Delay in loan processing
CPN02	C103	Rude behaviour of staff

Account_Card:

Card_ID	Account_ID	Card_Type	Issue_Date	Expiry_Date	Status
card_001	A01	Debit	01/01/2023	01/01/2027	Active
card_002	A03	ATM	01/06/2022	01/06/2026	Active
card_003	A04	Debit	01/03/2023	01/03/2027	Active
card_004	A05	Debit	01/12/2021	01/12/2025	Active
card_005	A06	ATM	01/04/2022	01/04/2026	Active

Transfer:

Amount	From_Accountid	To_Accountid
2000	A01	A02
1000	A02	A01
500	A05	A06
1000	A03	A05
5000	A02	A07

4) Strong and Weak Entities:

- Repaymnet is a weak entity as it depends on Loan_ID, uses Repay_ID.
- Loan Distribution is weak entity as it depends on Loan_ID.
- Account_Card is a weak entity as it depends on Account_ID, Card_Type.

5) ER Diagram:

Cardinality of Relationships:

- One customer can have **multiple accounts.**
- One Customer can have **multiple customer services.**
- One account can have **multiple account card.**
- One Customer can have **multiple complains.**
- One customer can take **multiple loans.**
- One account can have **multiple transactions.**
- One loan can have **multiple repayments.**
- One branch can have **multiple employee.**
- One branch handles **multiple accounts.**
- One account can do **multiple transfers.**
- One Loan can be distributed to **multiple repayments.**

Diagram:

6) Relational Diagram:

Schema:

- Cstomer (Cust_ID, Cust_Name, Phone)
- Branch (Branch_ID, Branch_Name)
- Account (Account_No, Cust_ID(Customer), Branch_ID(Branch), Type, Balance)
- Loan (Loan_ID, Cust_ID(Customer), Loan_Type, Amount)
- Transaction (Txn_ID, Account_No(Account), Txn_Type, Amount)
- Repayment (Repay_ID, Loan_ID(Loan), Amount)
- Employee(Emp_ID, Emp_Name, Branch_ID(Branch),Designation)
- SERVICE(ServiceID, ServiceName)
- ACCOUNT_CARD(CardID, Account_No, CardType, IssueDate, ExpiryDate, Status)
- COMPLAINT(ComplaintID, Cust_ID, Subject)
- Transfer(Transferid, From_Accountid,To_Accountid, Amount)
- LoanDistribution(Distributed_Amount,LoanID(Loan))

Diagram:

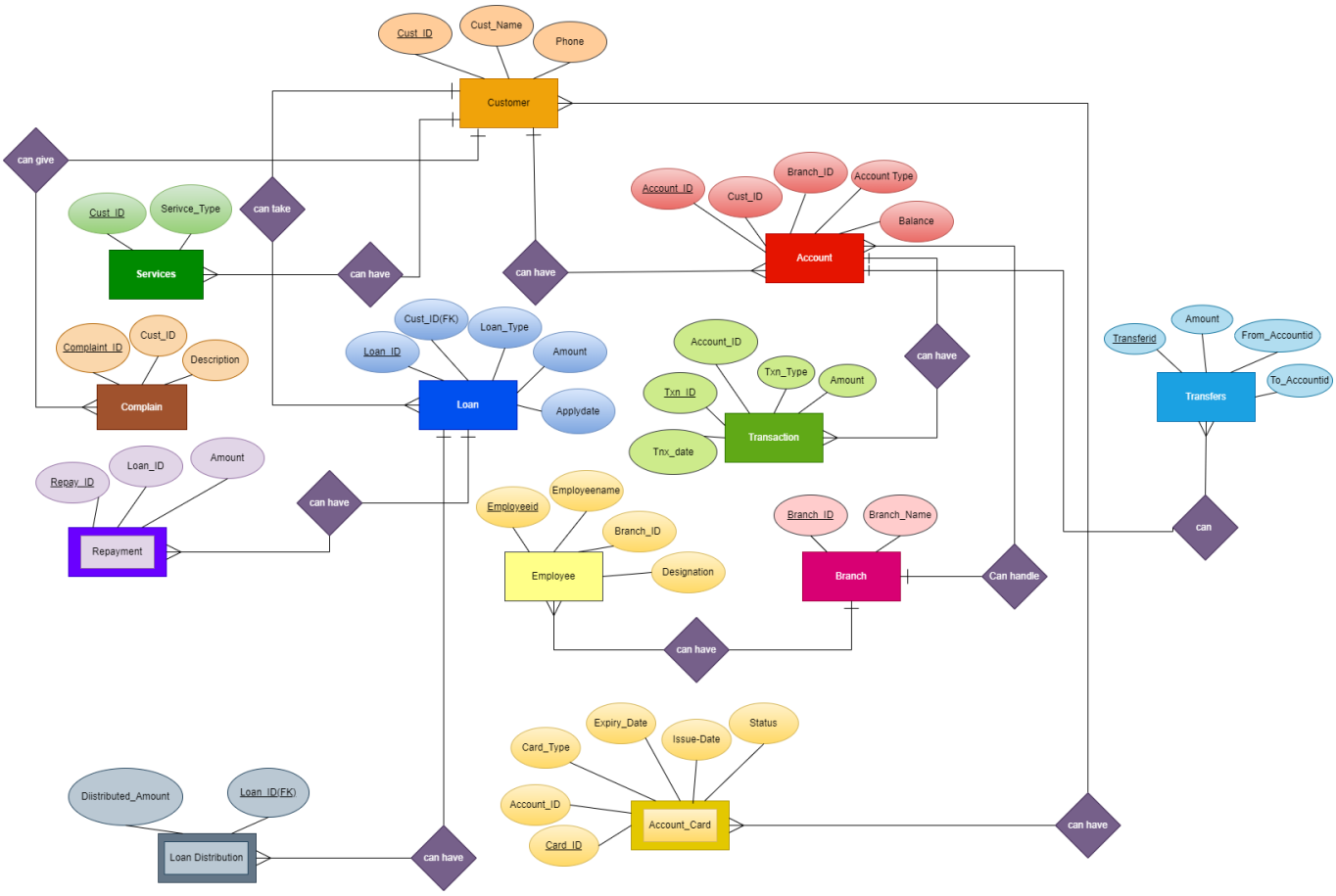
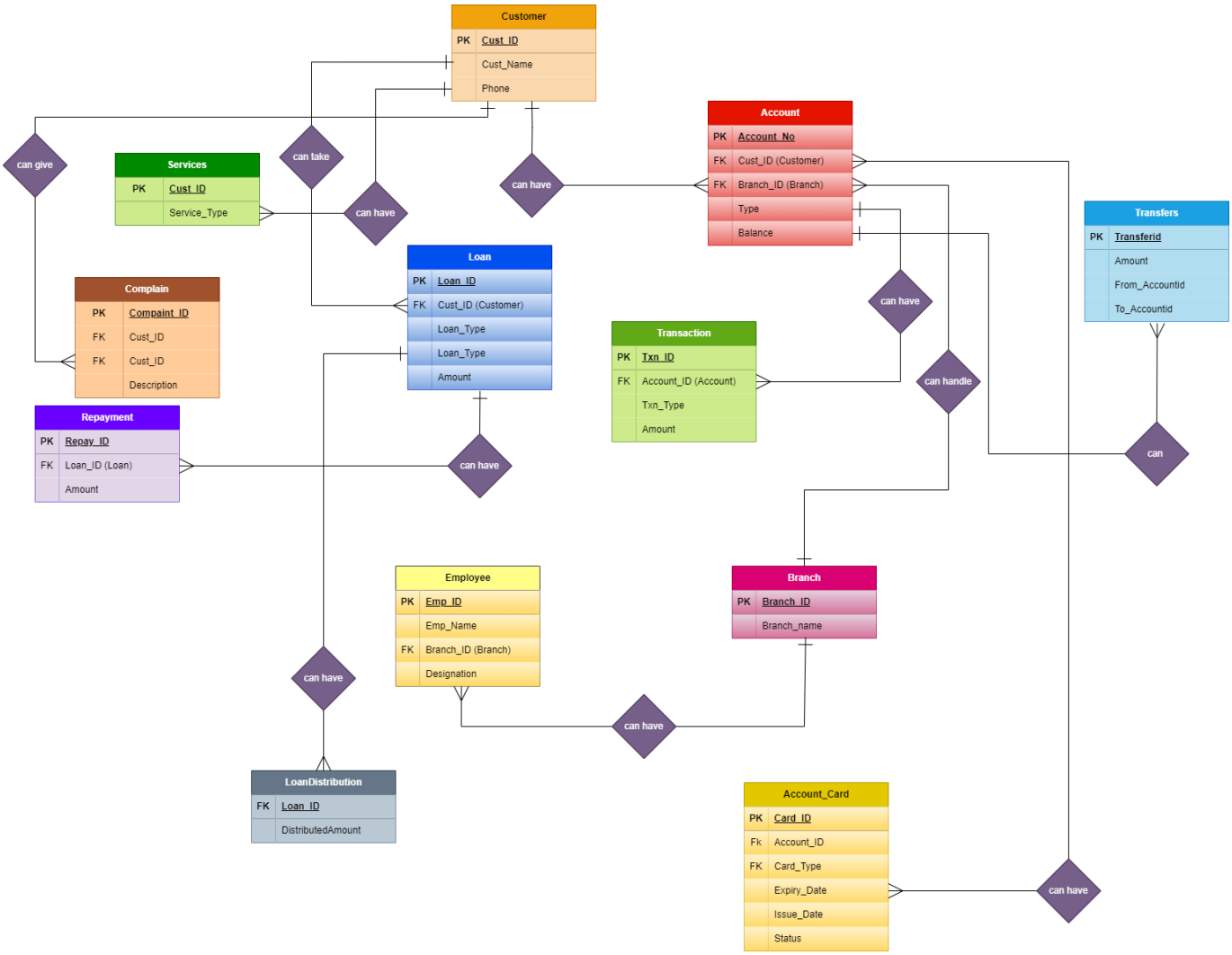

Cust_ID
Cust_Name
Phone
Customer
can give
Cust_ID
Serivce_Type
can take
Services
can have
can have
Account_ID
Cust_ID
Branch_ID
Account Type
Balance
Account
Complaint_ID
Cust_ID
Description
Complain
Cust_ID(FK)
Loan_Type
Loan_ID
Amount
Loan
Applydate
Account_ID
Txn_Type
Amount
Txn_ID
Tnx_date
Transaction
can have
Transferid
Amount
From_Accountid
To_Accountid
Transfers
Repay_ID
Loan_ID
Amount
Repayment
can have
Employeeid
Employeename
Branch_ID
Employee
Designation
Branch_ID
Branch_Name
Branch
Can handle
can
can have
Distributed_Amount
Loan_ID(FK)
Loan Distribution
can have
Card_Type
Expiry_Date
Issue-Date
Status
Account_ID
Card_ID
Account_Card
can have

Diagram:


Customer
PK Cust_ID
Cust_Name
Phone
Account
PK Account_No
FK Cust_ID (Customer)
FK Branch_ID (Branch)
Type
Balance
Transfers
PK Transferid
Amount
From_Accountid
To_Accountid
Services
PK Cust_ID
Service_Type
Loan
PK Loan_ID
FK Cust_ID (Customer)
Loan_Type
Loan_Type
Amount
Complain
PK Compaint_ID
FK Cust_ID
FK Cust_ID
Description
Transaction
PK Txn_ID
FK Account_ID (Account)
Txn_Type
Amount
Repayment
PK Repay_ID
FK Loan_ID (Loan)
Amount
Employee
PK Emp_ID
Emp_Name
FK Branch_ID (Branch)
Designation
Branch
PK Branch_ID
Branch_name
LoanDistribution
FK Loan_ID
DistributedAmount
Account_Card
PK Card_ID
Fk Account_ID
FK Card_Type
Expiry_Date
Issue_Date
Status
can give
can take
can have
can have
can have
can have
can handle
can
can have
can have
can have