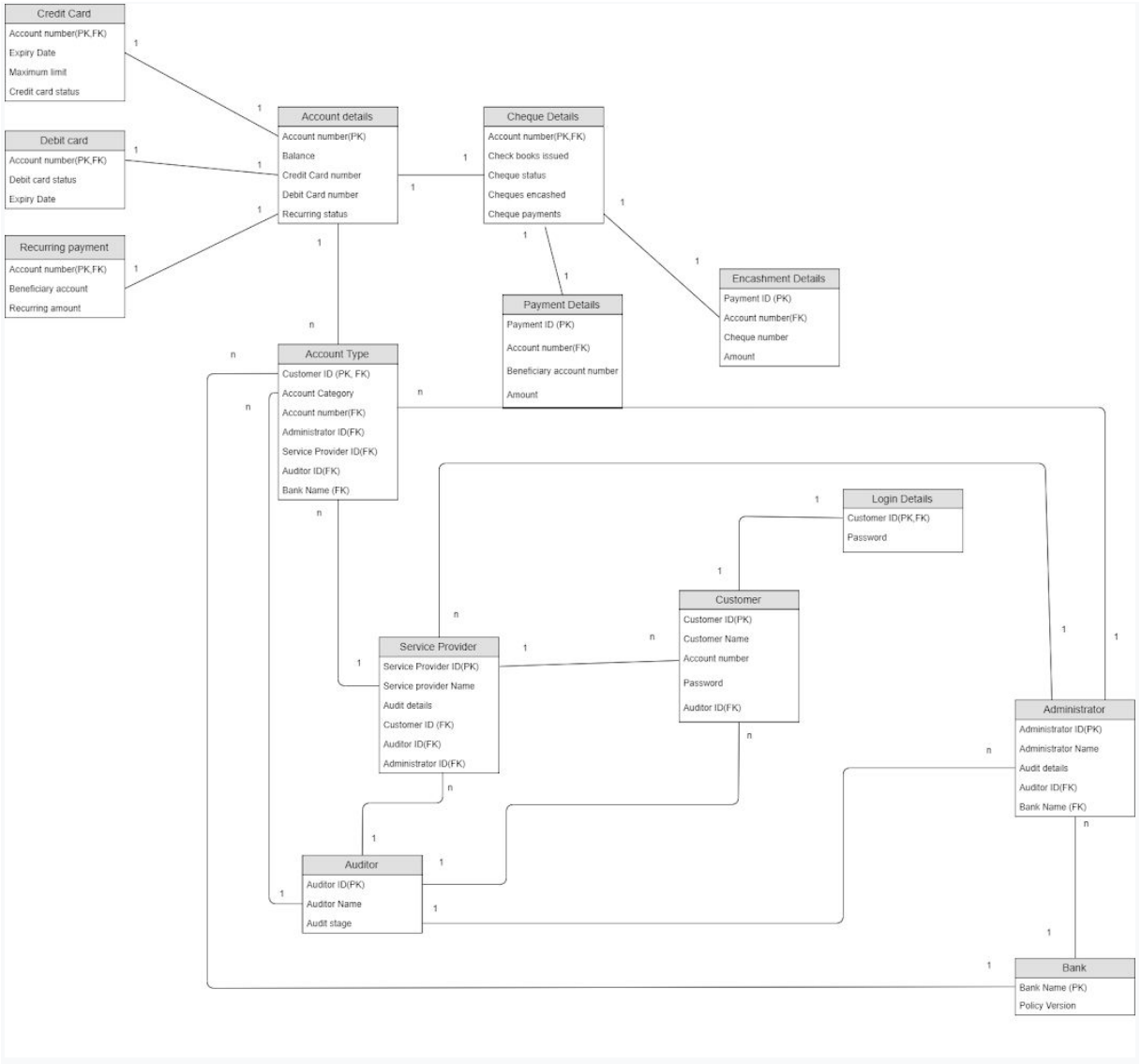
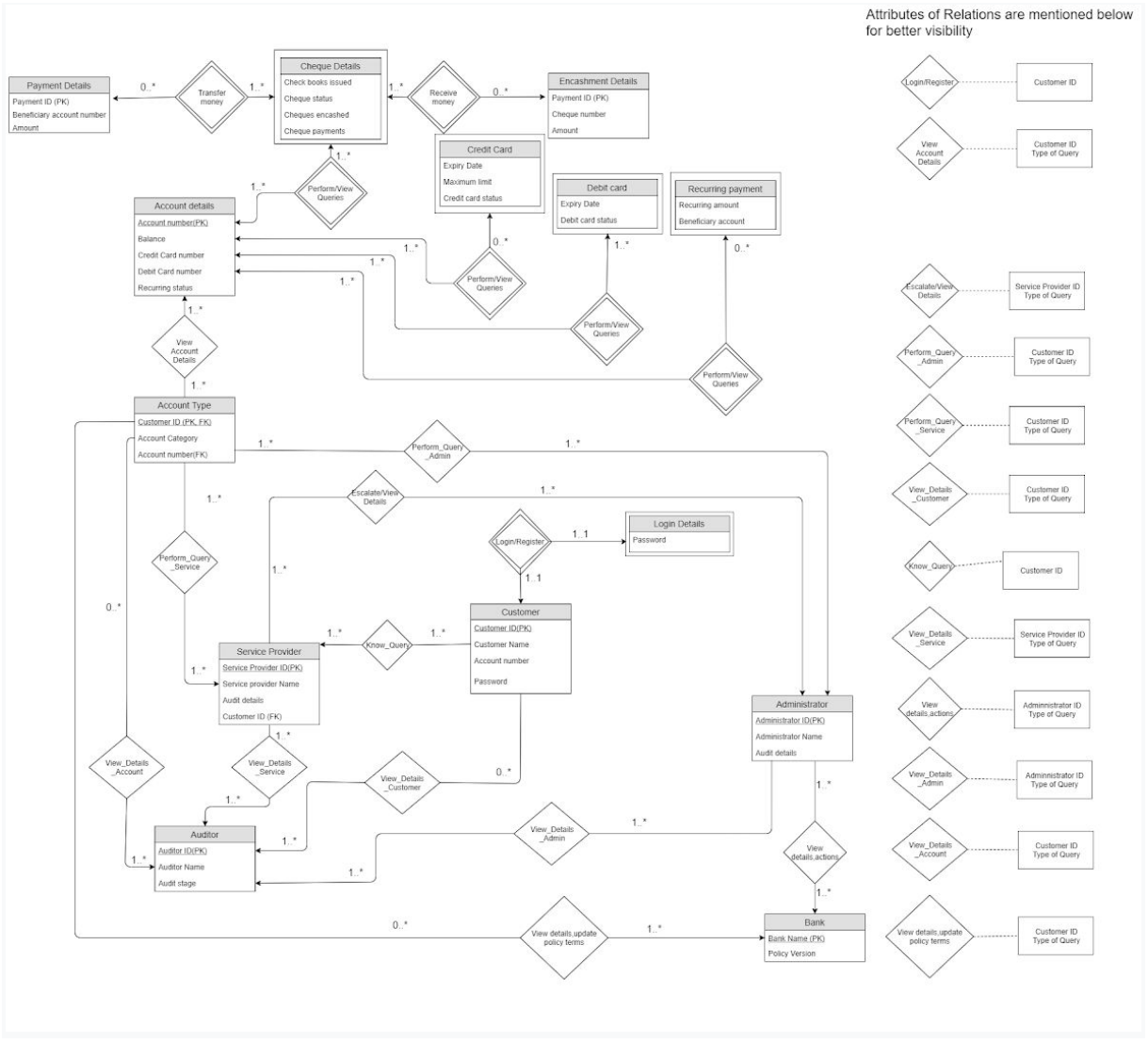


Relational SCHEMA

ER MODEL V3


Attributes of Relations are mentioned below for better visibility
Payment Details
Payment ID (PK)
Beneficiary account number
Amount
Transfer money
Cheque Details
Check books issued
Cheque status
Cheques encashed
Cheque payments
Receive money
Encashment Details
Payment ID (PK)
Cheque number
Amount
Credit Card
Expiry Date
Maximum limit
Credit card status
Debit card
Expiry Date
Debit card status
Recurring payment
Recurring amount
Beneficiary account
Perform/View Queries
Account details
Account number(PK)
Balance
Credit Card number
Debit Card number
Recurring status
View Account Details
Account Type
Customer ID (PK, FK)
Account Category
Account number(FK)
Perform_Query _Admin
Escalate/View Details
Perform_Query _Service
Login/Register
Login Details
Password
Customer
Customer ID(PK)
Customer Name
Account number
Password
Know_Query
Service Provider
Service Provider ID(PK)
Service provider Name
Audit details
Customer ID (FK)
Administrator
Administrator ID(PK)
Administrator Name
Audit details
View_Details _Account
View_Details _Service
View_Details _Customer
View_Details _Admin
View details,actions
Auditor
Auditor ID(PK)
Auditor Name
Audit stage
View details,update policy terms
Bank
Bank Name (PK)
Policy Version
Login/Register
Customer ID
View Account Details
Customer ID Type of Query
Escalate/View Details
Service Provider ID Type of Query
Perform_Query _Admin
Customer ID Type of Query
Perform_Query _Service
Customer ID Type of Query
View_Details _Customer
Customer ID Type of Query
Know_Query
Customer ID
View_Details _Service
Service Provider ID Type of Query
View details,actions
Administrator ID Type of Query
View_Details _Admin
Administrator ID Type of Query
View_Details _Account
Customer ID Type of Query
View details,update policy terms
Customer ID Type of Query

OLD DDL SCRIPT

1) Credit Card
CREATE TABLE Credit_card(
Account_Number CHAR(20) NOT NULL,
Expiry_Date DATE,
Maximum_Limit INTEGER
Credit_Card Status BOOLEAN
PRIMARY KEY (Account_Number),
FOREIGN KEY (Account_Number) REFERENCES Account_Details,
ON DELETE CASCADE)

2) Debit Card
CREATE TABLE Debit_card(
Account_Number CHAR(20) NOT NULL,
Expiry_Date DATE,
Debit_Card Status BOOLEAN
PRIMARY KEY (Account Number),
FOREIGN KEY (Account Number) REFERENCES Account Details,
ON DELETE CASCADE)

3)Recurring Payment
CREATE TABLE Recurring payment(
Account Number CHAR(20) NOT NULL,
Beneficiary account INTEGER,
Recurring amount INTEGER,
PRIMARY KEY (Account Number),
FOREIGN KEY (Account Number) REFERENCES Account Details,
ON DELETE CASCADE)

4)Login Details
CREATE TABLE Login Details(
Customer ID CHAR(20) NOT NULL,
Password CHAR(20),
PRIMARY KEY(Customer ID),
FOREIGN KEY (Customer ID) REFERENCES Customer,
ON DELETE CASCADE)

5) Payment Details
CREATE TABLE Payment Details(
Account Number CHAR(20) ,
Payment ID CHAR (20) NOT NULL
Beneficiary Account Number CHAR(20)
Amount INTEGER
PRIMARY KEY (Payment ID),
FOREIGN KEY (Account Number) REFERENCES Cheque Details,
ON DELETE CASCADE)

6) Account details
CREATE TABLE Account details
(Account Number CHAR(20) NOT NULL,
Balance FLOAT(8,2),
Credit Card number INTEGER,
Debit Card number INTEGER,
Recurring Status BOOLEAN,
PRIMARY KEY (Account Number),

)
6.1) Account_Details_1
CREATE TABLE Account_Number_1(
Account Number CHAR(20) NOT NULL,
Credit Card number INTEGER,
PRIMARY KEY (Account Number),
)

6.2)Account_Details_2
CREATE TABLE Account_Number_2(
Account Number CHAR(20) NOT NULL,
Debit Card number INTEGER,
PRIMARY KEY (Account Number),
)

6.3)Account_Details_3
CREATE TABLE Account_Number_3(
Account Number CHAR(20) NOT NULL,
Balance FLOAT(8,2),
Recurring Status BOOLEAN,
PRIMARY KEY (Account Number),
)

7) Bank
CREATE TABLE Bank(
Bank Name CHAR(20) NOT NULL,
Policy Version FLOAT(4,2),
PRIMARY KEY(Bank Name),
)

8) Service Provider
CREATE TABLE Service Provider(
Service Provider ID CHAR(20) NOT NULL,
Service Provider Name CHAR(20)
Audit Details FLOAT(4,2)
Auditor ID CHAR(20)
Administrator ID CHAR(20)
Customer ID CHAR(20)
PRIMARY KEY(Service Provider ID),
FOREIGN KEY (Customer ID) REFERENCES Customer,
FOREIGN KEY (Administrator ID) REFERENCES Administrator,
FOREIGN KEY (Auditor ID) REFERENCES Auditor,

)

9) Administrator
CREATE TABLE Administrator(
Audit Details FLOAT(4,2)
Auditor ID CHAR(20)
Administrator ID CHAR(20) NOT NULL
Administrator Name CHAR(20)
BANK NAME CHAR (20)
PRIMARY KEY(Administrator ID),
FOREIGN KEY (Bank Name) REFERENCES Bank,
FOREIGN KEY (Auditor ID) REFERENCES Auditor,

```
)

10) Cheque details
CREATE TABLE Cheque details(
Account Number CHAR(20) NOT NULL,
Cheques encashed INTEGER
Cheque payments FLOAT(8,2)
Check books issues INTEGER
Cheque Status BOOLEAN
PRIMARY KEY (Account Number),
FOREIGN KEY (Account Number) REFERENCES Account Details,
ON DELETE CASCADE)

11) Auditor
CREATE TABLE Auditor(
Auditor ID CHAR(20) NOT NULL,
Auditor Name CHAR(20),
Audit stage INT,
PRIMARY KEY (Auditor ID),
)

12) Encashment Details - OLD UNDECOMPOSED TABLE
CREATE TABLE Payment Details(
Payment ID CHAR(20) NOT NULL,
Account Number CHAR(20) NOT NULL,
Cheque Number CHAR(20),
Amount INTEGER,
PRIMARY KEY (Payment ID CHAR(20))
FOREIGN KEY (Account Number) REFERENCES Cheque Details,
ON DELETE CASCADE)

12.1)Payment_ID_1 - NEW DECOMPOSED TABLE
CREATE TABLE Payment_Details_1(
Payment _ID CHAR(20) NOT NULL,
Account Number CHAR(20) NOT NULL,
Cheque_Number CHAR(20)
PRIMARY KEY (Payment_ID CHAR(20))
FOREIGN KEY (Account Number) REFERENCES Cheque Details,
ON DELETE CASCADE
)

12.2)Payment_ID_2 -  NEW DECOMPOSED TABLE
CREATE TABLE Payment_Details_2(
Payment _ID CHAR(20) NOT NULL,
Account Number CHAR(20) NOT NULL,
Amount INTEGER,
PRIMARY KEY (Payment_ID CHAR(20))
FOREIGN KEY (Account Number) REFERENCES Cheque Details,
ON DELETE CASCADE
)

13) Customer
CREATE TABLE Customer(
Customer ID CHAR(20) NOT NULL,
Customer Name CHAR(20),
```

Account number CHAR(20),
Password CHAR(20),
Auditor ID,
PRIMARY KEY (Customer ID),
FOREIGN KEY (Auditor ID) REFERENCES Auditor)

14) Account Type

```
CREATE TABLE Account_type(
        Customer ID CHAR(20) NOT NULL,
        Account Category CHAR(20),
        Account number CHAR(20),
        Administrator ID CHAR(20),
        Service Provider ID,CHAR(20),
        Auditor ID CHAR(20),
        Bank Name CHAR(20)
        PRIMARY KEY (Customer ID),
        FOREIGN KEY (Customer ID) REFERENCES Service Provider,
        FOREIGN KEY (Account number) REFERENCES Account_Details_1,
        FOREIGN KEY (Administrator ID) REFERENCES Administrator,
        FOREIGN KEY (Service Provider ID) REFERENCES Service Provider,
        FOREIGN KEY (Auditor ID) REFERENCES Auditor,
        FOREIGN KEY (Bank Name) REFERENCES Bank)
        )
```

For the removal of redundancies, for most of the cases, i.e. many of the tables had a single primary key and so there were no redundancies in that. For two tables that are Encashment details, and Account details, we removed the redundancies while removing and modifying the DDL and schema accordingly. We analysed each and every table this way.

For the anomalies,i.e delete,insert,and update,

For any table if we change a tuple then the corresponding foreign keys associated with it also needs to be updated. Similarly we can extend it to the delete and insert anomalies as well. This consistency needs to be maintained and we have analyzed it for all the tables. It has been done in a generic sense to remove redundancies.

Credit Card: (BCNF)

- Primary Key : Account Number
- Foreign Key : Account Number
- Functional Dependency

  Account Number →Maximum Limit ,Credit Card Status, Expiry Date
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Debit Card (BCNF)

- Primary Key : Account Number
- Foreign Key : Account Number
- Functional Dependency

  Account Number →Credit Card Status, Expiry Date
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well

Recurring Payment (BCNF)

- Primary Key : Account Number
- Foreign Key : Account Number
- Functional Dependency

  Account Number →Beneficiary account, Recurring amount
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Login Details (BCNF)

- Primary Key : Customer ID
- Foreign Key : Customer ID
- Functional Dependency

  Customer ID → Password
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well

Payment Details (BCNF)

- Primary Key : Payment ID
- Foreign Key : Account Number
- Functional Dependency

  Payment ID → Account Number, Beneficiary Account Number, Amount

Account details (BCNF)

- Primary Key : Account Number
- Functional Dependency

  Account Number → Balance, Credit Card number, Debit Card number, Recurring Status
  Credit Card number→ Balance, Debit Card number, Recurring Status
  Debit Card number→ Balance, Recurring Status

  (Account Number,Credit Card number) - 1
  (Account Number, Debit Card number,) -2
  (Account Number,Balance Recurring Status) - 3

- We have applied Heath's theorem for transforming a non-BCNF table to a BCNF table.
  Let us take an example for explaining the heath's theorem where we have Initialized S = {R}
  While S has a relation R' that is not in BCNF do:
  Pick a FD: X->Y that holds in R' and violates BCNF

Add the relation XY to S
Update R' = R'-Y
Return S
So, now if s={ABCDE}
S = {ACDE, AB} // Pick FD: A->B which violates BCNF
S = {ACE, AB, CD} // Pick FD: C->D which violates BCNF
// Return S as all relations are in BCNF

Bank (BCNF)

- PRIMARY KEY:- Bank Name
- Functional Dependency

  Bank Name → Policy Version
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Auditor (BCNF)

- Primary Key : Auditor ID
- Foreign Key : Auditor ID
- Functional Dependency

  Auditor ID →Auditor Name ,Audit Stage
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Encashment Details (BCNF)

- Primary Key : Payment ID
- Foreign Key : Account Number
- Functional Dependency

  Payment ID→Cheque Number , Amount

  Cheque Number → Amount

  (Payment ID,Cheque Number) -1

  (Payment ID,Amount)-2
- We have applied Heath's theorem for transforming a non-BCNF table to a BCNF table.

  Let us take an example for explaining the heath's theorem where we have Initialized S = {R}

  While S has a relation R' that is not in BCNF do:

  Pick a FD: X->Y that holds in R' and violates BCNF

  Add the relation XY to S

  Update R' = R'-Y

  Return S

  So, now if s={ABCDE}

  S = {ACDE, AB} // Pick FD: A->B which violates BCNF

  S = {ACE, AB, CD} // Pick FD: C->D which violates BCNF

  // Return S as all relations are in BCNF

Cheque details (BCNF)

- Primary Key : Account Number
- Foreign Key : Account Number
- Functional Dependency

Account Number → Cheques encashed, Cheque payments, Check books issued, Cheque Status

- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Service Provider (BCNF)

- PRIMARY KEY:- Service Provider ID
- FOREIGN KEY:- Customer ID, Administrator ID, Auditor ID
- Functional Dependency

  Service Provider ID →Service Provider Name, Audit Details
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Administrator (BCNF)

- PRIMARY KEY:- Administrator ID
- FOREIGN KEY:- Bank Name, Auditor ID
- Functional Dependency

  Administrator ID →Administrator Name
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

Account Type (BCNF)

- Primary Key : Auditor ID
- Foreign Key : Account Number, Administrator ID, Service Provider ID, Auditor ID, Bank Name
- Functional Dependency

  Auditor ID → Account Number, Administrator ID, Service Provider ID, Bank Name

Customer (BCNF)

- Primary Key : Customer ID
- Foreign Key : Auditor ID
- Functional Dependency

  Customer ID →Customer Name,Account number,Password,Auditor ID
- Since we have atomic attributes, hence it is in first normal form. Furthermore we have a single attribute primary key, hence it is in second normal form. Since there is one candidate key, hence it is in BCNF as well.

PostgreSQL CODE to create tables: (UPDATED)

```
CREATE TABLE Auditor(
Auditor_ID CHAR(20) NOT NULL,
Auditor_Name CHAR(20),
Audit_stage INT,
PRIMARY KEY (Auditor_ID)
)

CREATE TABLE Customer(
Customer_ID CHAR(20) NOT NULL,
Customer_Name CHAR(20),
Account_number CHAR(20),
Pass_word CHAR(20),
Auditor_ID CHAR(20),
PRIMARY KEY (Customer_ID),
FOREIGN KEY (Auditor_ID) REFERENCES Auditor)

CREATE TABLE Login_Details(
Customer_ID CHAR(20) NOT NULL,
Pass_word CHAR(20),
PRIMARY KEY(Customer_ID),
FOREIGN KEY (Customer_ID) REFERENCES Customer
ON DELETE CASCADE)

CREATE TABLE Bank(
Bank_Name CHAR(20) NOT NULL,
Policy_Version numeric(4,2),
PRIMARY KEY(Bank_Name)
)

CREATE TABLE Administrator(
Audit_Details NUMERIC(4,2),
Auditor_ID CHAR(20),
Administrator_ID CHAR(20) NOT NULL,
Administrator_Name CHAR(20),
BANK_NAME CHAR (20),
PRIMARY KEY(Administrator_ID),
FOREIGN KEY (Bank_Name) REFERENCES Bank,
FOREIGN KEY (Auditor_ID) REFERENCES Auditor
)

CREATE TABLE  Service_Provider(
Service_Provider_ID CHAR(20) NOT NULL,
Service_Provider_Name CHAR(20),
Audit_Details NUMERIC(4,2),
Auditor_ID CHAR(20),
Administrator_ID CHAR(20),
Customer_ID CHAR(20),
PRIMARY KEY(Service_Provider_ID),
FOREIGN KEY (Customer_ID ) REFERENCES customer,
FOREIGN KEY (Administrator_ID) REFERENCES Administrator,
```

```
FOREIGN KEY (Auditor_ID) REFERENCES Auditor
)

CREATE TABLE Account_Details_1(
Account_Number CHAR(20) NOT NULL,
Credit_Card_number INTEGER,
PRIMARY KEY (Account_Number)
)
CREATE TABLE Account_Details_2(
Account_Number CHAR(20) NOT NULL,
Debit_Card_number INTEGER,
PRIMARY KEY (Account_Number)
)
CREATE TABLE Account_Details_3(
Account_Number CHAR(20) NOT NULL,
Balance NUMERIC(8,2),
Recurring_Status BOOLEAN,
PRIMARY KEY (Account_Number)
)

CREATE TABLE Account_type(
Customer_ID CHAR(20) NOT NULL,
Account_Category CHAR(20),
Account_number CHAR(20),
Administrator_ID CHAR(20),
Service_Provider_ID CHAR(20),
Auditor_ID CHAR(20),
Bank_Name CHAR(20),
PRIMARY KEY (Customer_ID),
FOREIGN KEY (Customer_ID) REFERENCES Service_Provider,
FOREIGN KEY (Account_number) REFERENCES Account_Details_1,
FOREIGN KEY (Administrator_ID) REFERENCES Administrator,
FOREIGN KEY (Service_Provider_ID) REFERENCES Service_Provider,
FOREIGN KEY (Auditor_ID) REFERENCES Auditor,
FOREIGN KEY (Bank_Name) REFERENCES Bank)

CREATE TABLE Credit_card(
Account_Number CHAR(20) NOT NULL,
Expiry_Date DATE,
Maximum_Limit INTEGER,
Credit_Card_Status BOOLEAN,
PRIMARY KEY (Account_Number),
FOREIGN KEY (Account_Number) REFERENCES Account_Details_1
ON DELETE CASCADE)

CREATE TABLE Debit_card(
Account_Number CHAR(20) NOT NULL,
Expiry_Date DATE,
Debit_Card_Status BOOLEAN,
PRIMARY KEY (Account_Number),
FOREIGN KEY (Account_Number) REFERENCES Account_Details_2
ON DELETE CASCADE)

CREATE TABLE Recurring_payment(
Account_Number CHAR(20) NOT NULL,
Beneficiary_account INTEGER,
Recurring_amount INTEGER,
PRIMARY KEY (Account_Number),
```

```
FOREIGN KEY (Account_Number) REFERENCES Account_Details_3
ON DELETE CASCADE)

CREATE TABLE Cheque_details(
Account_Number CHAR(20) NOT NULL,
Cheques_encashed INTEGER,
Cheque_payments NUMERIC(8,2),
Check_books_issues INTEGER,
Cheque_Status BOOLEAN,
PRIMARY KEY (Account_Number),
FOREIGN KEY (Account_Number) REFERENCES Account_Details_1
ON DELETE CASCADE)

CREATE TABLE Payment_Details(
Account_Number CHAR(20) ,
Payment_ID CHAR (20) NOT NULL,
Beneficiary_Account_Number CHAR(20),
Amount INTEGER,
PRIMARY KEY (Payment_ID),
FOREIGN KEY (Account_Number) REFERENCES Cheque_Details
ON DELETE CASCADE)

CREATE TABLE Payment_Details_1(
        Payment_ID CHAR(20) NOT NULL,
        Account_Number CHAR(20) NOT NULL,
        Cheque_Number CHAR(20),
        PRIMARY KEY (Payment_ID),
        FOREIGN KEY (Account_Number) REFERENCES Cheque_Details
        ON DELETE CASCADE
)

CREATE TABLE Payment_Details_2(
        Payment_ID CHAR(20) NOT NULL,
        Account_Number CHAR(20) NOT NULL,
        Amount INTEGER,
        PRIMARY KEY (Payment_ID),
        FOREIGN KEY (Account_Number) REFERENCES Cheque_Details
        ON DELETE CASCADE
)
```

SCREENSHOT OF THE RELATIONS