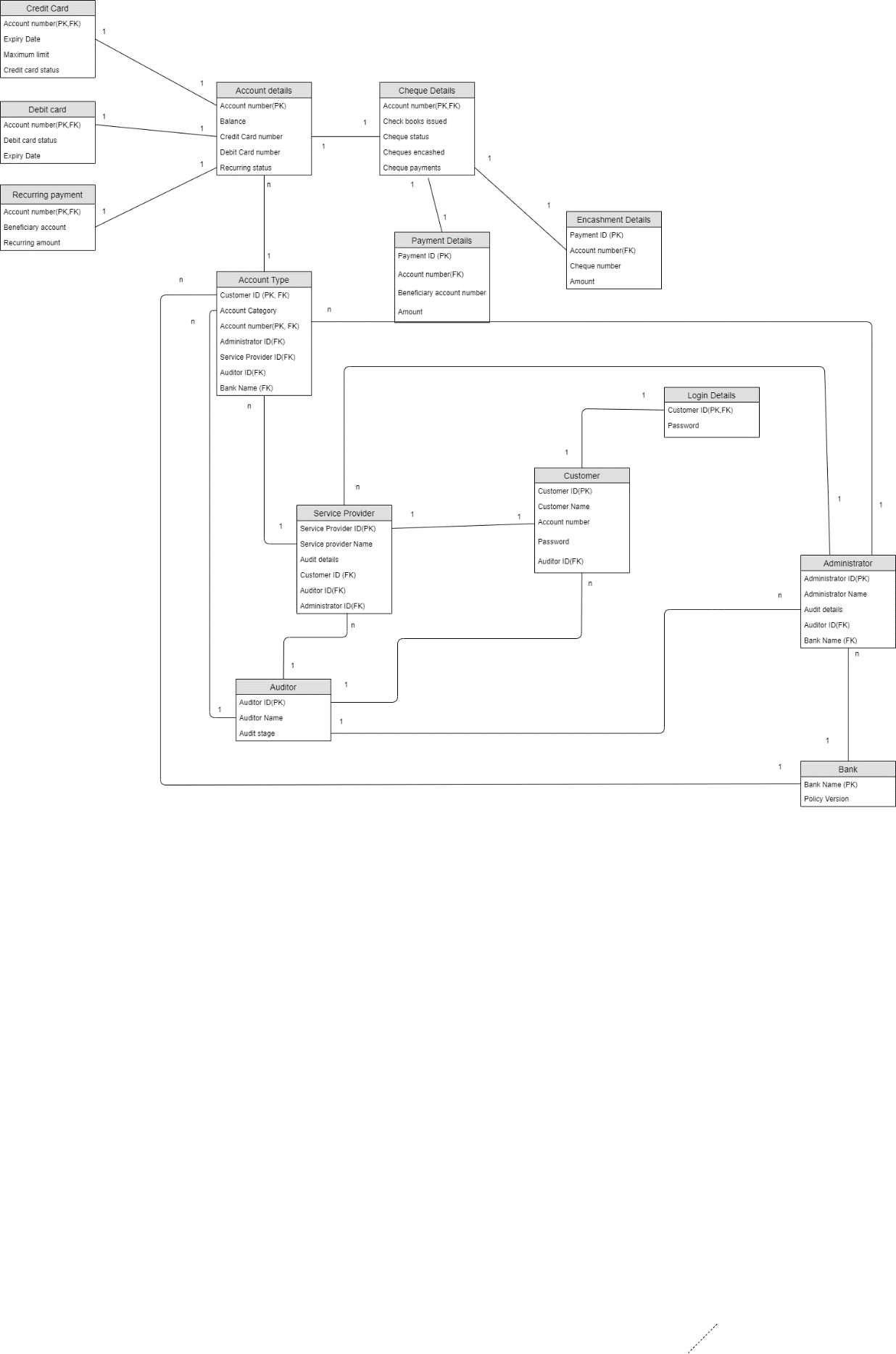
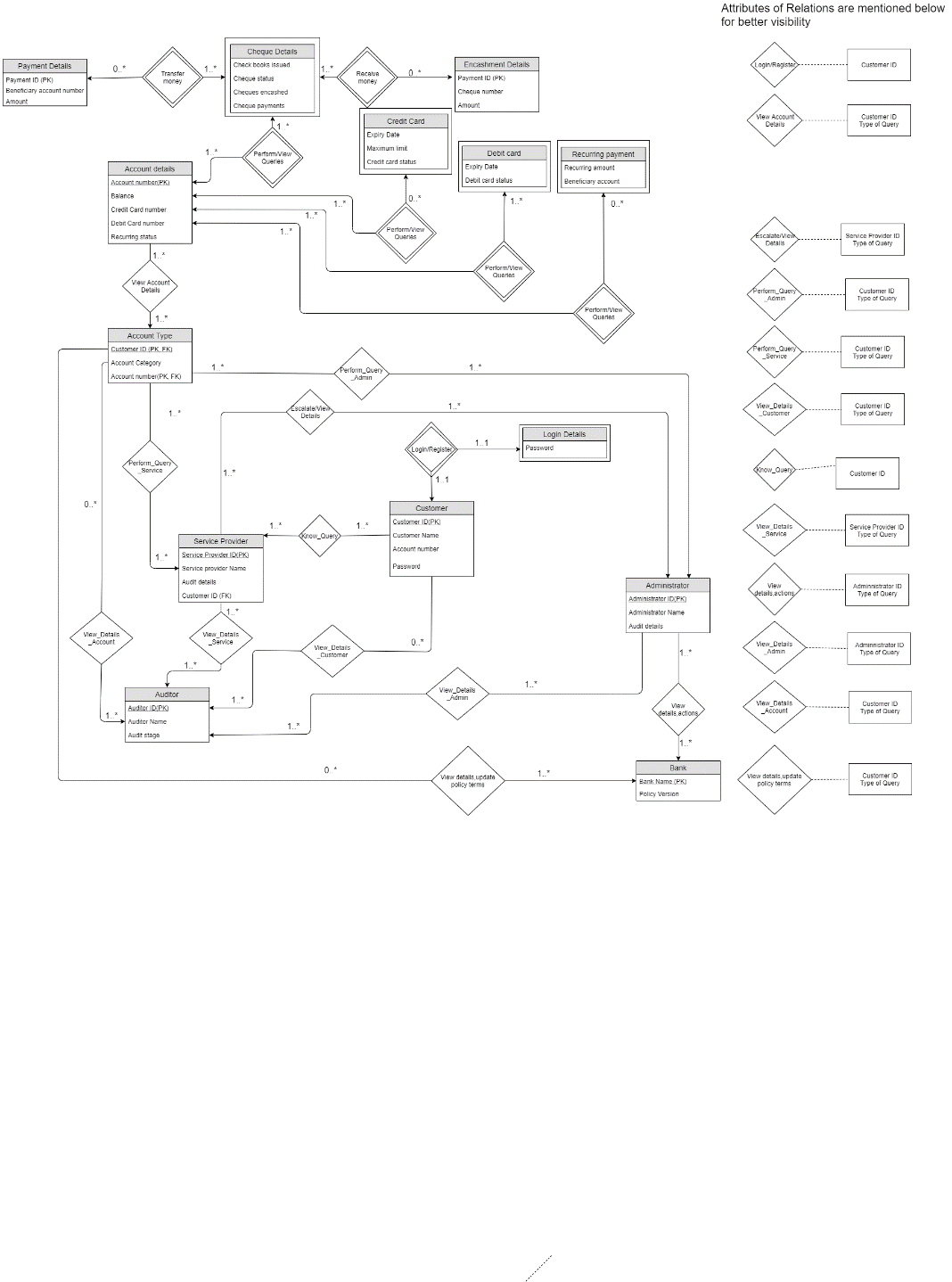
**Relational SCHEMA**

**ER MODEL V3**

****

**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)

1. 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, Customer\_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, Account\_number),

FOREIGN KEY (Account\_number) REFERENCES Account\_Details\_1 ON DELETE CASCADE,

FOREIGN KEY (Administrator\_ID) REFERENCES Administrator ON DELETE CASCADE,

FOREIGN KEY (Service\_Provider\_ID, Customer\_ID) REFERENCES Service\_Provider ON DELETE CASCADE,

FOREIGN KEY (Auditor\_ID) REFERENCES Auditor ON DELETE CASCADE,

FOREIGN KEY (Bank\_Name) REFERENCES Bank ON DELETE CASCADE)

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 CHAR(20),

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\_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