DBMS PROJECT on

<u>LINKING OF AADHAR WITH ACCOUNT AND MOBILE NUMBER</u>

Submitted by-

Himanshu Amethia (101611023) **COE9**

Submitted to-

Ms. Avleen Kaur Malhi

INTRODUCTION

Database is collection of data, or information, that is specially organized for rapid search and retrieval by a computer. Databases are structured to facilitate the storage, retrieval, modification, and deletion of data in conjunction with various data-processing operations.
A database management system (DBMS) extracts information from the database in response to queries. It is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data.

Recent guidelines provided by the government of India make it **compulsory for the all citizens to link their aadhar number with account and mobile number.** As Aadhar number uniquely indentifies a citizen of India clearly linking it will be very beneficial.

It will help in identifying many fake bank accounts that are created as well as fake sim cards issued for wrong purposes.

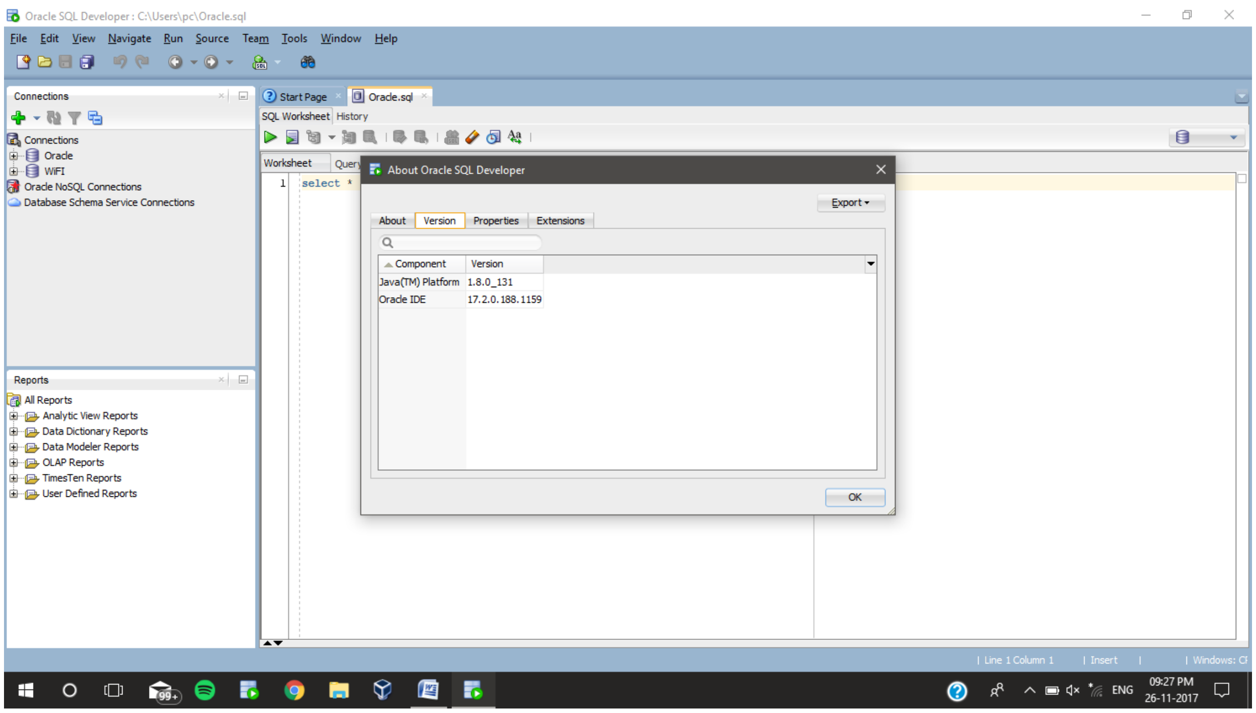
As India is a populated country, clearly a very efficient database system is required. Then only linking can be performed in a better and efficient way.

In this project er-diagram and pl/sql codes are made for linking of aadhar with mobile and account number.

It can be very useful for organizing and managing such an enmormous data in an efficient way.

System Requirements

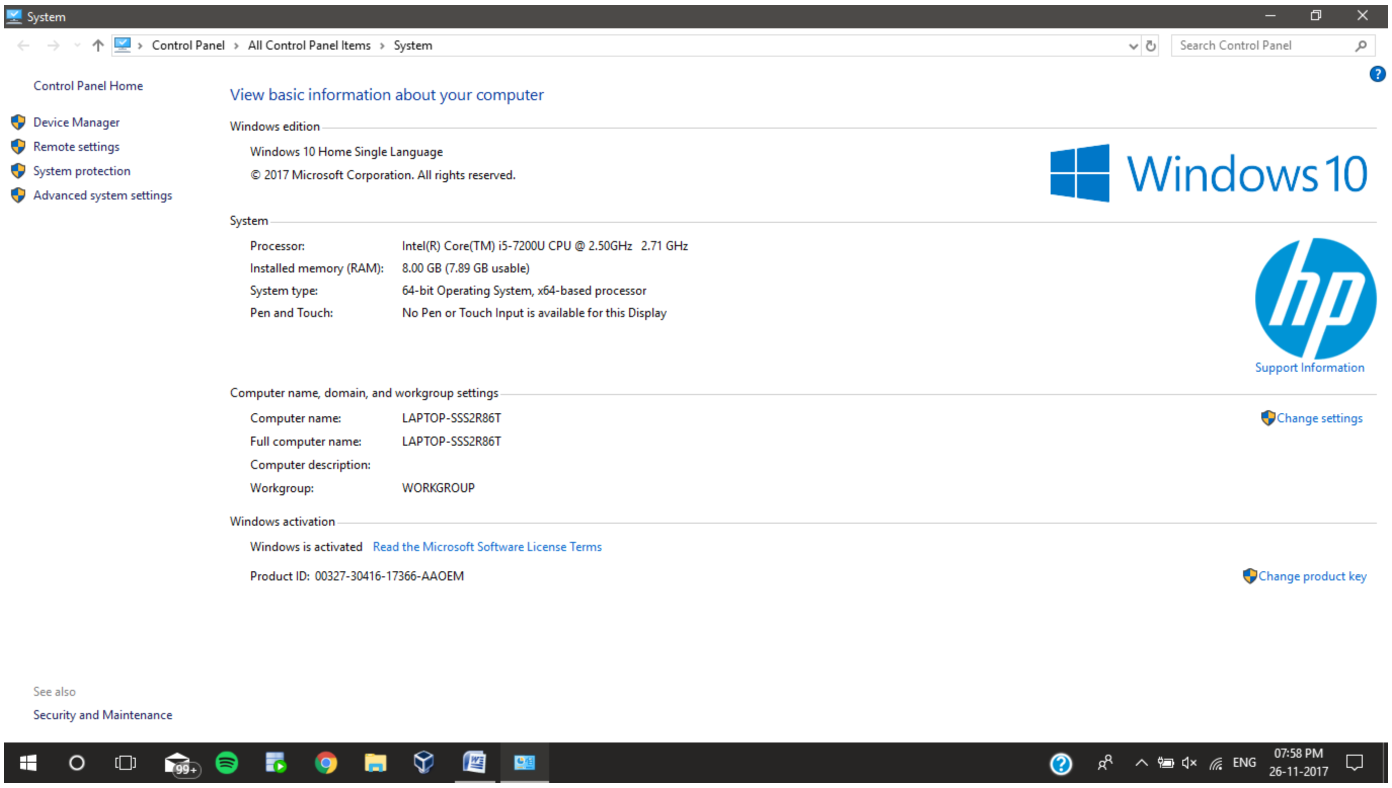
Oracle SQL Developer 11g Version 17.2.0.188 Build 188.1159



```
1 select *
```

Component	Version
Java(TM) Platform	1.8.0_131
Oracle IDE	17.2.0.188.1159

Windows 10, i5 processer



View basic information about your computer

Windows edition

Windows 10 Home Single Language

© 2017 Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
Installed memory (RAM):	8.00 GB (7.89 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name:	LAPTOP-SSS2R86T
Full computer name:	LAPTOP-SSS2R86T
Computer description:	
Workgroup:	WORKGROUP

Windows activation

Windows is activated Read the Microsoft Software License Terms

Product ID: 00327-30416-17366-AAOEM

INPUT / OUTPUT

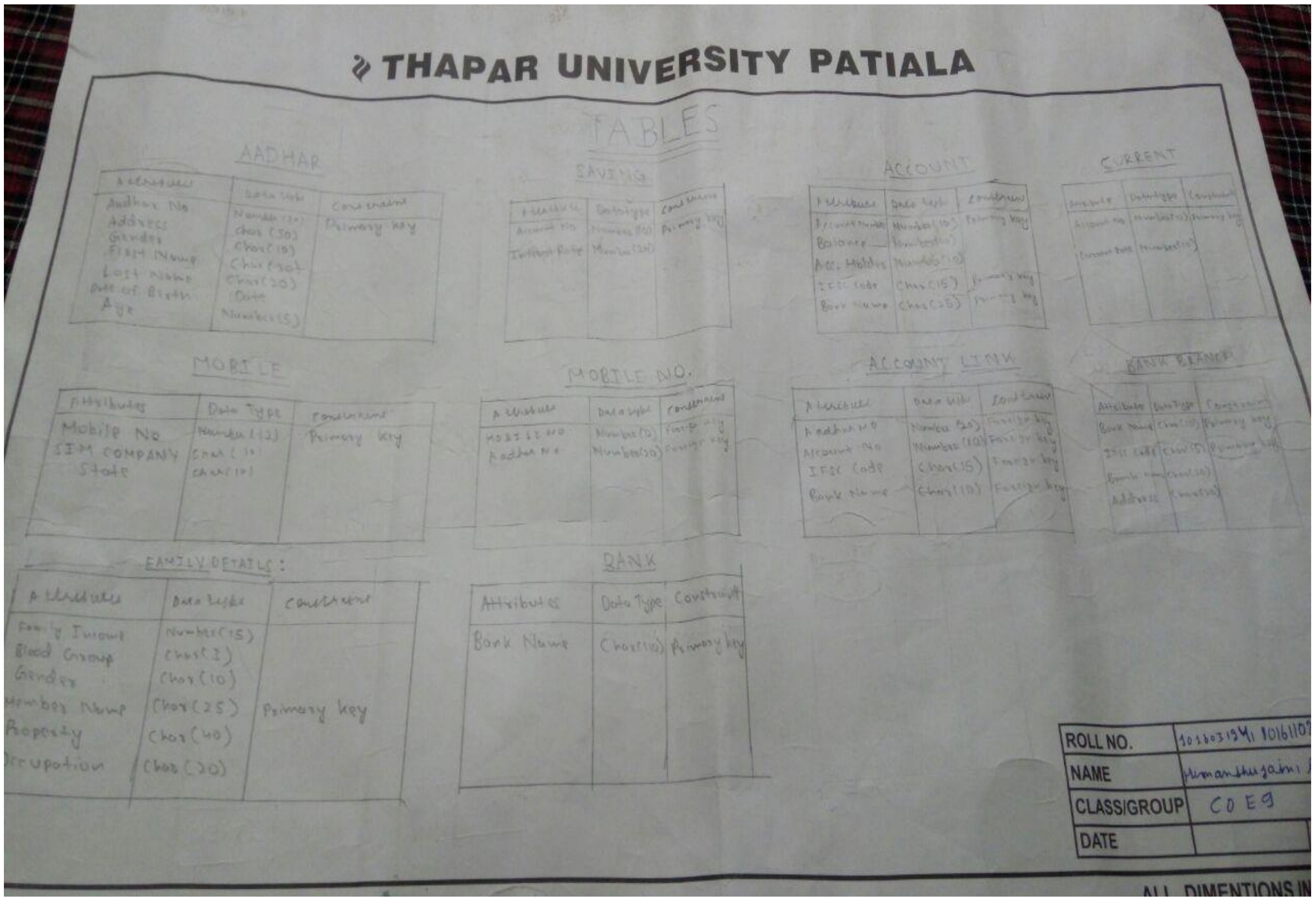
Basically this database system requires the user to **input** his **aadhar details** such as name, aadhar number, address etc. He is also required to enter his **account details** such as account number, ifsc code, bank name etc. Similarly he enters his **mobile details** like mobile number, sim company and state.

Clearly the main attributes in the table are **mobile, account and aadhar number.**

So, in the **output** this project **links** aadhar number with mobile number and account number. There are also pl/sql codes which take input from the user and display other important details like displaying family details for a particular aadhar number, checking number of accounts linked with a particular aadhar number, etc.

These codes make the database system more effective and is more beneficial for analysis.

ER DIAGRAM



THAPAR UNIVERSITY PATIALA

TABLES

AADHAR

Attribute	Data type	Constraint
Aadhar No	Number(20)	Primary key
Address	Char(50)	
Gender	Char(10)	
First Name	Char(20)	
Last Name	Char(20)	
Date of Birth	Date	
Age	Number(5)	

SAVING

Attribute	Datatype	Constraint
Account No	Number(10)	Primary key
Interest Rate	Number(20)	

ACCOUNT

Attribute	Data type	Constraint
Account Number	Number(10)	Primary key
Balance	Number(10)	
Acc. Holder	Number(10)	
IFSC Code	Char(15)	Primary key
Bank Name	Char(25)	Primary key

CURRENT

Attribute	Datatype	Constraint
Account No	Number(10)	Primary key
[illegible] Rate	Number(10)	

MOBILE

Attributes	Data Type	Constraint
Mobile No	Number(12)	Primary key
SIM COMPANY	Char(10)	
State	Char(10)	

MOBILE NO.

Attribute	Data type	Constraint
MOBILE NO	Number(12)	Foreign key
Aadhar No	Number(20)	Foreign key

ACCOUNT LINK

Attribute	Data type	Constraint
Aadhar No	Number(20)	Foreign key
Account No	Number(10)	Foreign key
IFSC Code	Char(15)	Foreign key
Bank Name	Char(10)	Foreign key

BANK BRANCH

Attribute	Data Type	Constraint
Bank Name	Char(10)	Primary key
IFSC Code	Char(15)	Primary key
Branch Name	Char(10)	
Address	Char(30)	

FAMILY DETAILS :

Attributes	Data type	Constraint
Family Income	Number(15)	
Blood Group	Char(3)	
Gender	Char(10)	
Member Name	Char(25)	Primary key
Property	Char(40)	
Occupation	Char(20)	

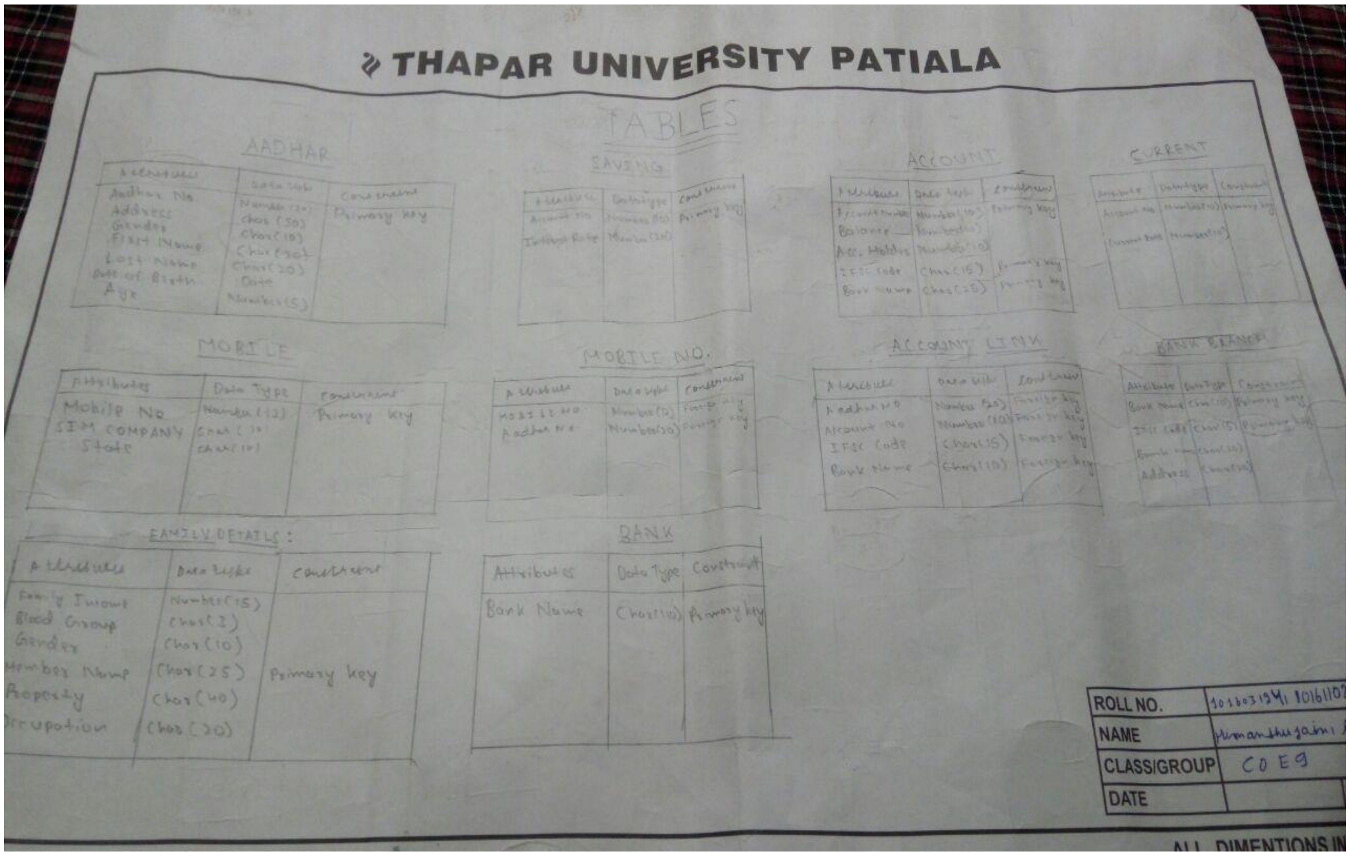
BANK

Attributes	Data Type	Constraint
Bank Name	Char(10)	Primary key

ROLL NO.	101603194, 1016110
NAME	Himanshu Jain,
CLASS/GROUP	COE9
DATE	

ALL DIMENTIONS IN

ER TO TABLE CONVERSION



THAPAR UNIVERSITY PATIALA

TABLES

AADHAR

Attributes	Data type	Constraint
Aadhar No	Number (20)	Primary key
Address	Char (50)	
Gender	Char (10)	
First Name	Char (20)	
Last Name	Char (20)	
Date of Birth	Date	
Age	Number (5)	

SAVING

Attributes	Datatype	Constraint
Account No	Number (10)	Primary key
Interest Rate	Number (20)	

ACCOUNT

Attributes	Data type	Constraint
Account Number	Number (10)	Primary key
Balance	Number (10)	
Acc. Holder	Number (10)	
IFSC Code	Char (15)	Primary key
Bank Name	Char (25)	Primary key

CURRENT

Attribute	Datatype	Constraint
Account No	Number (10)	Primary key
Current Rate	Number (10)	

MOBILE

Attributes	Data Type	Constraint
Mobile No	Number (12)	Primary key
SIM COMPANY	Char (10)	
State	Char (10)	

MOBILE NO.

Attribute	Data type	Constraint
MOBILE NO	Number (12)	Foreign key
Aadhar No	Number (20)	Foreign key

ACCOUNT LINK

Attribute	Data type	Constraint
Aadhar No	Number (20)	Foreign key
Account No	Number (10)	Foreign key
IFSC Code	Char (15)	Foreign key
Bank Name	Char (10)	Foreign key

BANK BRANCH

Attribute	Data Type	Constraint
Bank Name	Char (10)	Primary key
IFSC Code	Char (15)	Primary key
Branch Name	Char (10)	
Address	Char (30)	

FAMILY DETAILS :

Attributes	Data type	Constraint
Family Income	Number (15)	
Blood Group	Char (3)	
Gender	Char (10)	
Member Name	Char (25)	Primary key
Property	Char (40)	
Occupation	Char (20)	

BANK

Attributes	Data Type	Constraint
Bank Name	Char (10)	Primary key

ROLL NO.	101603194, 1016110
NAME	Himanshujaini
CLASS/GROUP	COE 9
DATE	

ALL DIMENTIONS IN

NORMALIZATION

We consider the family details table as follows :

Aadhar Number	Family Income	Property	No. of members	Occupation
12332321	**20000**	**Residental, Apartment**	**4**	**Business**
12332215	32000	Farm	3	Farming
12432153	21000	Townhouse	6	Teacher

Ist Normal Form:

- Each cell to be single valued
- Entries in a column of the same type
- Rows uniquely identified

So, clearly each cell is not single valued. There are more than one property for the same aadhar number. So, to make it in 1st normal form we make the property column single valued.

Below is the table normalized in 1st normal form:

Aadhar Number	Family Income	Property	No. of members	Occupation
12332321	20000	Residental	4	Business
12332321	20000	Apartment	4	Business
12332215	32000	Farm	3	Farming
12432153	21000	Townhouse	6	Teacher

2nd Normal Form:

All non key attributes should be dependent on the key attribute (i.e. primary key)

So, functional dependency for the above table are:

- Family income depends on aadhar number
- Property depends on aadhar number
- No. of members depends on aadhar number
- Occupation depends on aadhar number

Here, aadhar number is the prime key attribute and others are non key attributes. So, all non key attributes depend on the primary key.

Hence, table is already in second normal form.

3rd Normal Form:

- All columns in the table can be determined only by the prime key attribute and no other column

In the given table family income, property, no. of members and occupation depend only on aadhar number and cannot be determined by any other non key attribute.

So, the table is already in third normal form.

4th Normal Form:

- There should be no multi value dependency in the table.

Aadhar Number	Family Income	Property	No. of members	Occupation
12332321	**20000**	**Residental**	**4**	**Business**
12332321	**20000**	**Apartment**	**4**	**Business**
12332215	32000	Farm	3	Farming
12432153	21000	Townhouse	6	Teacher

Clearly there is **multi value dependency** in the given table. **One aadhar number has two property.**

So, decomposing the table into two tables as shown:

Aadhar Number	Family Income	No. of members	Occupation
12332321	20000	4	Business
12332215	32000	3	Farming
12432153	21000	6	Teacher

Aadhar Number	Property
12332321	Residental
12332321	Apartment
12332215	Farm
12432153	Townhouse

Here now Aadhar number in 1st table is the primary key and aadhar number in 2nd table is the foreign key.

Now, clearly the above tables are in fourth normal form as there is no multi value dependency.

5th Normal Form:

A table T is in fifth normal form if every join dependency in T is a consequence only of the candidate keys of T.

Clearly, the given table is already in fifth normal form.

Next consider the table Bank Branch:

IFSC Code	Branch Name	Address
SBIN1002	Court Jagadhri	Sector 17, Jagadhri, Haryana
VITY1223	Sectt Shamli	Matka Road, Shamli, UP

Ist Normal Form:

- Each cell to be single valued
- Entries in a column of the same type
- Rows uniquely identified

Every row is single is valued and uniquely identified by IFSC Code.

So, the table is already in 1st normal form.

2nd Normal Form:

All non key attributes should be dependent on the key attribute (i.e. primary key)

So, functional dependency for the above table are:

- Branch Name depends on IFSC Code
- Address depends on IFSC Code

Here, IFSC Code is the prime key attribute and others are non key attributes. So, all non key attributes depend on the primary key i.e. IFSC Code.

Hence, table is already in second normal form.

3rd Normal Form:

- All columns in the table can be determined only by the prime key attribute and no other column

IFSC Code	**Branch Name**	**Address**
SBIN1002	**Court Jagadhri**	**Sector 17, Jagadhri, Haryana**
VITY1223	**Sectt Shamli**	**Matka Road, Shamli, UP**

Here, clearly branch name and branch address are dependent on primary key **but branch address can be determined by branch name.**

So, the table is not in third normal form. So decomposing it into two table as follows:

IFSC Code	**Branch Name**
SBIN1002	Court Jagadhri
VITY1223	Sectt Shamli

Branch Name	Address
Court Jagadhri	Sector 17, Jagadhri, Haryana
Sectt Shamli	Matka Road, Shamli, UP

Now, clearly all non key attributes are dependent only on key attribute i.e. IFSC Code. **Branch name is second table is primary key and branch name is 1st table is foreign key.**

Hence, the table is in third normal form.

4th Normal Form:

- There should be no multi value dependency in the table.

  In given table clearly there is no multi value dependency. As **one IFSC Code corresponds to single branch name and branch address.**

So, the table is already in fourth normal form.

5th Normal Form:

A table T is in fifth normal form if every join dependency in T is a consequence only of the candidate keys of T.

Clearly, the given table is already in fifth normal form.

Rest of the tables already satisfy all the five normal forms mentioned above. Hence, rest of the tables are already normalized upto fifth normal form.

IMPLEMENTATION OF CODE

```
create table mobile(mobileno number(15) primary key, simcompany
char(15),state char(15));

create table aadhar(aadharno number(15) primary key,firstname
char(15),lastname char(15), gender char(15),address char(15),dob
date,age number(15))

create table mobilelink(mobileno number(15) references
mobile(mobileno),aadharno number(15) references
aadhar(aadharno),primary key(mobileno,aadharno))

create table bank(bankname char(15) primary key);

create table branchadd(branchname char(15) primary key,address
char(15));

create table bankbranch(ifsccode char(15) primary key,branchname
char(15) references branchadd(branchname));

create table branches(ifsccode char(15) references
bankbranch(ifsccode),bankname char(15) references
bank(bankname),primary key(ifsccode,bankname))

create table account(accountno number(15) primary key,accholder
char(15),balance number(15))

create table accountlink(accountno number(15) references
account(accountno),aadharno number(15) references
aadhar(aadharno),primary key(accountno,aadharno))
```

```
create table accounts(ifsccode char(15) references
bankbranch(ifsccode),accountno number(15) references
account(accountno), primary key(ifsccode,accountno));

create table saving(aacountno number(15) primary key,interestrate
number(15));

create table current1(aacountno number(15) primary key,currentrate
number(15));

create table familydetails(aadharno number(15) primary
key,familyincome number(15),totalmembers
number(15),mainoccupation char(15));

create table propertydetails(aadharno number(15) references
familydetails(aadharno),property char(15));
```

mobile

```
insert into mobile values(9799670662,'airtel','up')

insert into mobile values(9799650661,'airtel','punjab')

insert into mobile values(9899650661,'vodafone','rajsthan')

insert into mobile values(9499650676,'bsnl','ap')

insert into mobile values(9599650676,'idea','mp')
```

aadhar

```
insert into aadhar
values(322112345450,'himanshu','amethia','male','gorakhpur',(TO_DAT
E('22/April/2011','DD/MON/YY')),18)

insert into aadhar
values(312112345450,'himanshu','jain','male','hissar',(TO_DATE('21/Ap
ril/1998','DD/MON/YY')),19)

insert into aadhar
values(242112345450,'himanshu','sharma','male','kota',(TO_DATE('25/
April/1998','DD/MON/YY')),19)

insert into aadhar
values(142112345450,'anurag','kamboj','male','jagadhri',(TO_DATE('25/
April/1996','DD/MON/YY')),20)

insert into aadhar
values(562112345450,'hk','amethia','male','jaipur',(TO_DATE('25/march
/1996','DD/MON/YY')),20)
```

mobile link

```
insert into mobilelink values(9799670662,322112345450)

insert into mobilelink values(9799650661,322112345450)

insert into mobilelink values(9899650661,312112345450)

insert into mobilelink values(9499650676,242112345450)

insert into mobilelink values(9599650676,142112345450)

insert into mobilelink values(9599650676,562112345450)
```

bank

```
insert into bank values('hdfc')

insert into bank values('icici')

insert into bank values('sbi')

insert into bank values('bandhan')

insert into bank values('pnb')
```

branchadd

```
insert into branchadd values('fatehabad','modeltown')

insert into branchadd values('faridabad','anaj mandi')

insert into branchadd values('patiala','22 no.')

insert into branchadd values('ambala','new market')

insert into branchadd values('jagadhri','old market')
```

bankbranch

```
insert into bankbranch values('punb0019100','fatehabad')

insert into bankbranch values('hdfc0000116','faridabad')

insert into bankbranch values('icici00001','patiala')

insert into bankbranch values('bdbl0001014','ambala')

insert into bankbranch values('sbin0003897','jagadhri')
```

branches

```
insert into branches values('punb0019100','pnb')

insert into branches values('hdfc0000116','hdfc')

insert into branches values('icici00001','icici')

insert into branches values('bdbl0001014','bandhan')

insert into branches values('sbin0003897','sbi')
```

account

```
insert into account values('00987654321','himanshujain',1500)

insert into account values('00987654320','himanshujain',15000)

insert into account values('00987654323','anuragkamboj',10)

insert into account values('00987654324','himanshuamethia',20)

insert into account values('00987654325','himanshusharma',200000)
```

accountlink

```
insert into accountlink values('00987654321',322112345450)

insert into accountlink values('00987654320',322112345450)

insert into accountlink values('00987654323',312112345450)

insert into accountlink values('00987654324',242112345450)
insert into accountlink values('00987654325',142112345450)


accounts


insert into accounts values('punb0019100',00987654321)

insert into accounts values('hdfc0000116',00987654320)

insert into accounts values('icici00001',00987654323)

insert into accounts values('bdbl0001014',00987654324)

insert into accounts values('sbin0003897',00987654325)


saving

insert into saving values(00987654321,8)

insert into saving values(00987654320,10)

insert into saving values(00987654323,10)

insert into saving values(00987654324,9)

insert into saving values(00987654325,9)


current

insert into current1 values(00987654321,7)

insert into current1 values(00987654320,8)

insert into current1 values(00987654323,8)
insert into current1 values(00987654324,9)

insert into current1 values(00987654325,9)
```

family detail

```
insert into familydetails values(322112345450,500000,7,'agriculture')

insert into familydetails values(312112345450,1000000,7,'buisness')

insert into familydetails values(242112345450,100000,8,'govtemployee')

insert into familydetails values(142112345450,100000,8,'teaching')

insert into familydetails
values(562112345450,700000,8,'propertydealer')
```

property details

```
insert into propertydetails values(322112345450,'residential')

insert into propertydetails values(322112345450,'apartment')

insert into propertydetails values(242112345450,'apartment')

insert into propertydetails values(242112345450,'farm')

insert into propertydetails values(562112345450,'townhouse')
```

....end...

PL/SQL CODES

PROCEDURES AND FUNCTIONS

PROCEDURE
IT is a subprogram that can accept parameters perform on action and return parameters.

IT is of types:-
1. Local procedure
2.stored procedure

STORED PROCEDURE-
1-

```
CREATE PROCEDURE show_balance (account_no NUMBER,balan
out NUMBER)
AS
BEGIN
SELECT balance INTO balan FROM account  WHERE accountno =
account_no;
END;
```

CALLING PROCEDURE SHOW_BALANCE/////////
/* IT WILL SHOW BALANCE OF A PARTICULAR ACCOUNT */

```
DECLARE
bal number;
BEGIN
show_balance(:accno,bal);
dbms_output.put_line(bal);
END;
```

2-

```
CREATE PROCEDURE close_account (account_no NUMBER)
AS
BEGIN
DELETE FROM account WHERE accountno = account_no;
END;
```

CALLING PROCEDURE CLOSE_ACCOUNT////////////////////

/* IT WILL CLOSE THE PARTICULAR ACCOUNT*/

```
DECLARE
e number;
BEGIN
e:= :ACCOUNTNO;
close_account(e);
END;
```

/////////////////////////

LOCAL PROCEDURE-

3-

```
DECLARE
PROCEDURE insertd(accountno number,accholder char,balance
number)
AS
BEGIN
INSERT INTO account values(accountno,accholder,balance);
END;
BEGIN
insertd(:accn,:accholder,:balance);
END;
```

/* USE TO INSERT VALUE*/

................................./////////////..

END OF PROCEDURES...

TRIGGERS WITH EXCEPTION HANDLING:

A database trigger is a stored procedure that is fired when an INSERT, UPDATE, or DELETE statements is issued against the associate table.

1-

```
CREATE OR REPLACE TRIGGER PRIMARY_KEY
BEFORE INSERT ON account
FOR EACH ROW
DECLARE
a account.accountno%TYPE;
BEGIN
IF (:NEW.accountno IS NULL) THEN
RAISE_APPLICATION_ERROR(-20002, 'PRIMARY KEY
VIOLATION
BECAUSE IT CANNOT BE NULL');
END IF;
SELECT accountno INTO a FROM account WHERE
accountno=:NEW.accountno;
RAISE_APPLICATION_ERROR(-20003, 'PRIMARY KEY
VIOLATION
BECAUSE IT SHOULD BE UNIQUE');
EXCEPTION
WHEN NO_DATA_FOUND THEN
NULL;
END;

/* PRIMARY KEY VIOLATION */
```

2-

```
CREATE OR REPLACE TRIGGER FOREIGN_KEY
BEFORE INSERT ON accountlink
FOR EACH ROW
DECLARE
aNum accountlink.accountno%TYPE;
BEGIN
SELECT accountno INTO aNum FROM accountlink WHERE
accountno=:NEW.accountno;
NULL;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20004, 'FOREIGN KEY
VIOLATED
BECAUSE VALUE IS NOT FOUND IN THE PARENT TABLE');
END;

/* FOREIGN KEY VIOLATION*/
```

3-

```
CREATE OR REPLACE TRIGGER bank_UPDATE
AFTER UPDATE OF bankname ON bank FOR EACH ROW
BEGIN
DBMS_OUTPUT.PUT_LINE('OLD NAME: ' ||:OLD.BANKNAME);
DBMS_OUTPUT.PUT_LINE('NEW NAME:' ||:NEW.BANKNAME);
END
```

---------------------------------END-------------------------------------

CURSORS

Cursor is a memory (work) area that a oracle engine uses for its internal processing for executing and
storing the results of SQL statement, and this work area is reserved for SQL's operations also called
Oracle's Private area or CURSOR .

1-TO SHOW HOW MANY MOBILENO ARE LINKED WITH AN AADHAR

```
DECLARE
CURSOR C1 IS SELECT mobileno FROM mobilelink WHERE
aadharno=322112345450;
REC C1%ROWTYPE;
BEGIN
OPEN C1;
LOOP
FETCH C1 INTO REC;
EXIT WHEN C1%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('mobileno '||REC.mobileno);
END LOOP;
CLOSE C1;
END;
```

/* IT WILL TELL THAT WHICH MOBILW NO. IS LINKED WITH A GIVEN MOBILE.NO */

OUTPUT OF ABOVE CODE

```
mobileno 9799650661
mobileno 9799670662
```

2-TO COUNT NO. OF ACCOUNT LINKED WITH GIVEN AADHAR CARD

```
DECLARE
N NUMBER;
BEGIN
DELETE from accountlink WHERE aadharno=:aadharno;
N:=SQL%ROWCOUNT;
DBMS_OUTPUT.PUT_LINE('TOTAL NUMBER OF account
FOUND' || N);
END;
```

/* IT WILL TELL THAT HOW MANY ACCOUNT NO.LINKED WITH GIVEN AADHAR NO.*/

3-TO SHOW THAT ACCOUNT IS EXIST IN TABLE OR NOT;

```
BEGIN
DELETE from accountlink WHERE accountno=:accountno;
IF SQL%NOTFOUND THEN
DBMS_OUTPUT.PUT_LINE('RECORD NOT found');
ELSE
DBMS_OUTPUT.PUT_LINE('RECORD found');
END IF;
END;
```

/* IT WILL TELL THAT WHETHER A PARICULAR ACCOUNT EXIST IN TABLE */