IT-214 DBMS PROJECT

Aadhar Card System

(Group ID - T615)

Group Members: -

1. Utsav Pansuriya –202201468
2. Priyank Asodariya- 202201470
3. Shrey Bavishi –202201478
4. Jish Chanchapra –202201501
5. Zenil Rupareliya -202201527

Prof. P. M. Jat

TA. Princy Chauhan

SUMMARY

We implement the data base of uidai.com website, in that our main motivation is to provide various services to Aadhar-card holders. The following website aims to help user in various ways such as online booking of appointment, updating Aadhar-information, checking status related to their Aadhar-card and to get information about previous transactions.

This web-site provides data to the government, so that they can use it for finding various statistical study on basis these data such as finding total number of Aadhar holder, checking if bank is linked to implement various financial beneficiary scheme. The government can also make various monetary decision regarding enrolment centre which may be burden on government due to less generation of revenue on long run.

The following website also used for user's authentication for government, private and NGO organisation.

EXPERIENCE

The project provided us an opportunity to do a real-life application of various theoretical studies which we learned during our current semester. It also helped us in identifying real life problem and thus solving it.

While implementing our solution we faced many problems such as changing ER and Relational diagram due to various real-life problem which would arise during discussion.

So, we were stuck on whether to allow users without Aadhar cards onto our website. We offer different services for non-Aadhar users, like scheduling appointments for registration. After some discussion, we decided to accommodate both. But then came the challenge of how to manage Aadhar and non-Aadhar users in our system. We thought of putting them in the same database table but needed a way to distinguish between them when scheduling appointments. Finally, we settled on representing them as subclasses, which sorted out our main issue.

Then, when it came to handling service availability, we hit another roadblock. Figuring out the status of services was a bit tricky, but we managed to find a solution in the end. The project provided us a practical learning while solving problem which arise during our project.

DDL: -

```
CREATE SCHEMA  AADHAARDATABASE;
SET SEARCH_PATH TO AADHAARDATABASE;

CREATE TABLE PIN(
PINCODE INT PRIMARY KEY,
STATE VARCHAR(255)
);

CREATE TABLE LOCAL(
LOCALITY VARCHAR(255) PRIMARY KEY,
PINCODE INT,
STATE VARCHAR(255)
);

CREATE TABLE ENROLLMENT_CENTER (
  EC_ID INT PRIMARY KEY,
  EC_TYPE VARCHAR(255),
  EC_NAME VARCHAR(255),
  PIN INT REFERENCES PIN(PINCODE) ON UPDATE CASCADE ON
DELETE CASCADE,
  LOCALITY VARCHAR(255) REFERENCES LOCAL(LOCALITY)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE CITIZEN (
  CITIZEN_NO BIGINT PRIMARY KEY,
  GENDER CHAR(1),
  NAME VARCHAR(255) NOT NULL,
  DOB DATE,
  LOCALITY VARCHAR(255) REFERENCES LOCAL(LOCALITY)ON UPDATE CASCADE ON DELETE CASCADE,
  PINCODE INT REFERENCES PIN(PINCODE)ON UPDATE CASCADE ON DELETE CASCADE,
  EC_ID INT REFERENCES ENROLLMENT_CENTER(EC_ID) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE REGISTERED_USER (
  AADHAR_NO BIGINT PRIMARY KEY,
  CITIZEN_NO BIGINT,
  REGISTERED_MOBILE_NO BIGINT,
  FOREIGN KEY (CITIZEN_NO) REFERENCES CITIZEN(CITIZEN_NO) ON UPDATE CASCADE ON DELETE CASCADE
);
```

```
CREATE TABLE APPOINTMENT (
  APPOINTMENT_ID BIGINT PRIMARY KEY,
  DATE DATE,
  TIME TIME,
  AADHAR_NO BIGINT REFERENCES
REGISTERED_USER(AADHAR_NO) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE FINDS(
  CITIZEN_NO BIGINT,
  EC_ID INT,
  PINCODE INT,
  PRIMARY KEY (CITIZEN_NO, EC_ID),
  FOREIGN KEY (CITIZEN_NO) REFERENCES
CITIZEN(CITIZEN_NO) ON UPDATE CASCADE ON DELETE
CASCADE,
  FOREIGN KEY (EC_ID) REFERENCES
ENROLLMENT_CENTER(EC_ID) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE UPDATE_REQUIREMENT (
  UPDATE_TYPE VARCHAR(255) PRIMARY KEY,
  UPDATE_FEE INT
);

CREATE TABLE APPOINTMENT_FOR (
```

```
  APPOINTMENT_ID BIGINT,
  UPDATE_TYPE VARCHAR(255),
  EC_ID INT,
  STATUS VARCHAR(255),
  PRIMARY KEY (APPOINTMENT_ID, UPDATE_TYPE),
  FOREIGN KEY (UPDATE_TYPE) REFERENCES
UPDATE_REQUIREMENT(UPDATE_TYPE) ON UPDATE CASCADE
ON DELETE CASCADE,
  FOREIGN KEY (APPOINTMENT_ID) REFERENCES
APPOINTMENT(APPOINTMENT_ID) ON UPDATE CASCADE ON
DELETE CASCADE,
  FOREIGN KEY (EC_ID) REFERENCES
ENROLLMENT_CENTER(EC_ID) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE UPDATE_DOCUMENT (
   UPDATE_TYPE VARCHAR(255),
   DOCUMENT VARCHAR(255),
   PRIMARY KEY(DOCUMENT,UPDATE_TYPE),
   FOREIGN KEY (UPDATE_TYPE) REFERENCES
UPDATE_REQUIREMENT(UPDATE_TYPE) ON UPDATE CASCADE
ON DELETE CASCADE
  );

CREATE TABLE DOCUMENTS (
```

```
  DOCUMENT_TYPES VARCHAR(255),
  DOCUMENT_SUBMITTED VARCHAR(255),
  PRIMARY KEY(DOCUMENT_TYPES,AADHAR_NO),
 AADHAR_NO BIGINT REFERENCES
REGISTERED_USER(AADHAR_NO) ON UPDATE CASCADE ON
DELETE CASCADE
);

CREATE TABLE ALL_STATUS(
  STATUS_TYPE VARCHAR(255) PRIMARY KEY
);

CREATE TABLE ONLINE_CHECK_STATUS (
AADHAR_NO BIGINT REFERENCES
REGISTERED_USER(AADHAR_NO) ON UPDATE CASCADE ON
DELETE CASCADE,
STATUS_TYPE VARCHAR(255) REFERENCES
ALL_STATUS(STATUS_TYPE) ON UPDATE CASCADE ON DELETE
CASCADE,
 STATUS_VALUE VARCHAR(255),
PRIMARY KEY(AADHAR_NO, STATUS_TYPE)
);

CREATE TABLE EMPLOYEE (
  E_ID BIGINT PRIMARY KEY,
  E_NAME VARCHAR(255),
  E_EMAIL VARCHAR(255),
  E_PHONE_NO BIGINT,
```

```
  DEPARTMENT_NAME VARCHAR(255),
  EC_ID INT REFERENCES ENROLLMENT_CENTER(EC_ID) ON
UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE BANK_ACCOUNT (
  ACCOUNT_NO BIGINT,
  IFSC_CODE BIGINT,
  BANK_NAME VARCHAR(255),
  BRANCH VARCHAR(255),
        PRIMARY KEY(ACCOUNT_NO,IFSC_CODE)
);

CREATE TABLE AADHAR_SERVICES (
  VIRTUAL_ID INT PRIMARY KEY,
  LOCK_STATUS BOOLEAN,
  E_KYC_STATUS BOOLEAN,
  BIOMETRIC_LOCK_STATUS BOOLEAN,
  ACCOUNT_NO BIGINT,
        IFSC_CODE BIGINT,
        FOREIGN KEY (ACCOUNT_NO,IFSC_CODE)REFERENCES
BANK_ACCOUNT(ACCOUNT_NO,IFSC_CODE) ON UPDATE
CASCADE ON DELETE CASCADE
);


CREATE TABLE PAYMENT (
```

```
  VIRTUAL_ID INT ,
  PAYMENT_HISTORY VARCHAR(255),
  FOREIGN KEY (VIRTUAL_ID) REFERENCES
AADHAR_SERVICES(VIRTUAL_ID) ON UPDATE CASCADE ON
DELETE CASCADE,
  PRIMARY KEY(PAYMENT_HISTORY,VIRTUAL_ID)
);

CREATE TABLE AVAIL_SERVICES (
  VIRTUAL_ID INT ,
  AADHAR_NO BIGINT,
  FOREIGN KEY (VIRTUAL_ID) REFERENCES
AADHAR_SERVICES(VIRTUAL_ID) ON UPDATE CASCADE ON
DELETE CASCADE,
  FOREIGN KEY (AADHAR_NO) REFERENCES
REGISTERED_USER(AADHAR_NO) ON UPDATE CASCADE ON
DELETE CASCADE,
  PRIMARY KEY(AADHAR_NO,VIRTUAL_ID)
);

CREATE TABLE FETTCH (
  VIRTUAL_ID INT,
  FETCH_HISTORY VARCHAR(255),
  FOREIGN KEY (VIRTUAL_ID) REFERENCES
AADHAR_SERVICES(VIRTUAL_ID) ON UPDATE CASCADE ON
DELETE CASCADE,
  PRIMARY KEY (FETCH_HISTORY, VIRTUAL_ID));
```

FUNCTIONAL DEPENDENCIES AND NORMALIZATION

- **Citizen**
  Attributes – Citizen (CitizenNo, Gender, Fname, Mname, Lname, Locality,

  State, Pincode, Dob,EC_ID)

- **Minimal Functional Dependencies Set**
  CitizenNo ->Fname

  CitizenNo -> Gender

  CitizenNo ->Mname

  CitizenNo ->Lname

  CitizenNo -> Locality

  CitizenNo -> State

  CitizenNo -> Pincode

  CitizenNo -> DOB

  CitizenNo -> EC_ID

  Pincode -> State

  Locality -> Pincode

  Locality -> State

  Let find closure of CitizenNo

  CitizenNo+ = {CitizenNo,Gender,Fname

  ,Mname,Lname,Locality,State,Pincode,DOB,EC_ID}

  Since its closure include all attribute of Citizen relation , therefor CitizenNo is Primary Key.

Last 3 functional dependencies does not have super-key on left side of FD hence the given relation is not in BCNF, it is 2NF. We can decompose

it into BCNF by having 3 relation as following

R1(CitizenNo,Gender,Fname ,Mname,Lname, DOB,EC_ID)

R2(Pincode,State)

R3(Locality,Pincode,State)

- **Enrollment Centre**
  Attributes – Enrollment Centre

  (EC_ID,EC_Type,EC_Name,Pincode,State,Locality)

- **Minimal Functional Dependencies Set**
  EC_ID -> EC_Name

  EC_ID -> EC_Type

  EC_ID -> Pincode

  EC_ID -> State

  EC_ID -> Locality

  Pincode -> State

  Locality -> Pincode

  Locality -> State

  Let find closure set of EC_ID

  EC_ID+ = {EC_ID , EC_Name ,EC_Type,Pincode,State,Locality}

  Since it involves all attributes of Enrollment Centre relation it is Primary Key.

Last 3 functional dependencies does not have super-key on left side of FD hence the given relation is not in BCNF, it is 2NF. We can decompose

it into BCNF by having 3 relation as following

R1(EC_ID , EC_Name ,EC_Type)

R2(Pincode,State)

R3(Locality,Pincode,State)

- **Finds**
  Attribute – Finds(CitizenNo ,EC_ID, Pincode)

- **Minimal Functional Dependencies Set**
  {CitizenNo, EC_ID} - > Pincode

  Primary Key is {CitizenNo , EC_ID}.

  Relation is in BCNF as super-key is present on left side of FD.

- **Appointment For**
  Attributes – Appointment For (AppointmentID , UpdateType, Status , EC_ID)

- **Minimal Functional Dependencies Set**
  {AppointmentID , UpdateType} - > Status

  {AppointmentID , UpdateType} - > EC_ID

  Primary Key is {AppointmentID , UpdateType} as it determines all attribute of given relation

  Since primary key is present on left side of all FDs of relation we can say “Appointment For” relation to be in BCNF.

- **Appointment**
  Attribute – Appointment (AppointmentID , Date , Time, AadharNo)

- **Minimal Functional Dependencies Set**
  AppointmentID -> AadharNo

  AppointmentID -> Date

  AppointmentID -> Time

  Primary Key is AppointmentID as it determines all other attribute of relation. Relation is in BCNF as all FD has AppointmentID on left side of its FD's.

- **Registered User**
  Attribute – Registered User (AadharNo , CitizenNo , RegisteredMobileNo)

- **Minimal Functional Dependencies Set**
  AadharNo - > CitizenNo
  AadharNo -> RegisteredMobileNo

  Primary Key is AadharNo as it determines all other attribute of relation. Relation is in BCNF as all FD has AadharNo on left side of its FD's.

- **Documents**
  Attributes – Documents (DocumentType , AadharNo , DocumentSubmitted)

- **Minimal Functional Dependencies Set**
  {DocumentType , AadharNo} -> DocumentSubmitted

Primary Key is {DocumentType , AadharNo} as it determines all other attribute of relation. Relation is in BCNF as all FD has {DocumentType , AadharNo} on left side of its FD's.

- **Employee**
  Attribute – Employee

  (E_ID,Name,Email,PhoneNo,DepartmentName,EC_ID)
- **Minimal Functional Dependencies Set**
  E_ID->Name

  E_ID -> Email

  E_ID->PhoneNo

  E_ID ->DepartmentName

  E_ID - >EC_ID

  Primary Key is E_ID as it determines all other attribute of relation. Relation is in BCNF as all FD has E_ID on left side of its FD's.

- **Aadhar Services**
  Attribute – Aadhar Services(VirtualID, LockStatus,

  EKYCStatus,BiometricLockStatus,AccountNo,FetchHistory(multivalue),Pay mentHistory(multivalue))

- **Minimal Functional Dependencies Set**
  VirtualID -> LockStatus

  VirtualID -> EKYCStatus

  VirtualID -> BiometricLockStatus

  VirtualID -> AccountNo

  VirtualID- >> FetchHistory

  VirtualID ->> PaymentHistory

Primary Key is VirtualID as its closure include all attribute of relation.

However the relation is not in BCNF due to presence of multivalue dependencies such as FetchHistory and PaymentHistory.Since this multivalue dependencies are independent of each other we can so we decompose the given relation into 3 new relation as following

R1 (VirtualID, LockStatus, EKYCStatus,BiometricLockStatus,AccountNo)

R2 (VirtualID , PaymentHistory) R3 (VirtualID , FetchHistory)
Now relation is in 4NF.

- **BankAccount**
  Attribute – BankAccount (AccountNo , IFSCCode , Name,Branch)

- **Minimal Functional Dependencies Set**
  IFSCCode - >Name

  IFSCCode -> Branch

  Primary Key is {AccountNo ,IFSCCode} as its closure include all attributes of relation. Relation is in BCNF.

- **Online Check Status**
  Attribute – Online Check Status (AadharNo , StatusType,Status)

- **Minimal Functional Dependencies Set**
  {AadharNo,StatusType} -> Status

{AadharNo , StatusType} together form composite primary key as it closure include all attribute of relation. The relation above is in BCNF as primary key is present on left of above FD.

- **Update Requirement**
  Attribute – Update Requirement(Update Type , Update Fee)

- **Minimal Functional Dependencies Set**
  Update Type -> Update Fee

  Update Type is primary key as it closure include all attribute. The relation is in BCNF as primary key is present on left of above FD

- **Update Document**
  Attribute – Update Document(Update Type , Document)

  Update Type and Document together form composite primary key hence there is no FD and table is in BCNF

- **All Status**
  Attribute – All Status(StatusType)

  All status have Status Type as it primary key hence no FD is present and thus table is BCNF.

- **Avail Services**
  Attribute – Avail Services(VirtualID , AadharNo)

  Both (VirtualID , AadharNo) form composite primary key and hence relation is in BCNF.

Queries:

Queries For Government :-

1) Find the how much revenue generated month wise for given year(Here year = 2024)?

```
WITH R AS (
   SELECT EC_ID, DATE, update_FEE
   FROM APPOINTMENT
   NATURAL JOIN APPOINTMENT_FOR
   NATURAL JOIN UPDATE_REQUIReMENT
   WHERE EXTRACT('YEAR' FROM DATE) = 2024
)
SELECT EXTRACT('MONTH' FROM DATE) AS MONTH,
      SUM(update_FEE) AS REVENUE
FROM R
GROUP BY EXTRACT('MONTH' FROM DATE)
ORDER BY SUM(update_FEE) DESC;
```

2) Find the users who got error in updating.(besides this government can send the notification for re_appointment.)

```
SELECT AADHAR_NO,MOBILE_NO ONLINE_CHECK_STATUS
NATURAL JOIN REGISTERED_USER NATURAL JOIN CITIZEN
WHERE STATUS = 'ERROR'.
```

3) Find the which EC work efficiently? (Which EC approved appointment of less number of people so that government can increase there a number of employes or number of ECs in that area.)

```
SELECT PIN,EC_ID,C1/(C1+C2) FROM

((SELECT EC_ID, COUNT(APPOINTMENT_ID) AS C1

FROM APPOINTMENT_FOR

WHERE STATUS != 'APPROVED' GROUP BY (EC_ID)) AS T1

NATURAL JOIN

(SELECT EC_ID, COUNT(APPOINTMENT_ID) AS C2

FROM APPOINTMENT_FOR WHERE STATUS != 'PENDING'

GROUP BY (EC_ID)) AS T2) NATURAL JOIN
ENROLLMENT_CENTER

ORDER BY(C1/(C1 + C2)) DESC;
```

---

4) Find the time-table of appointments of Aadhar-holders in any enrollment_center?

```
SELECT APPOINTMENT_ID, DATE, TIME, EC_id, Aadhar_no,
UPDATE_TYPE

    FROM APPOINTMENT

    NATURAL JOIN APPOINTMENT_FOR

    WHERE EXTRACT(MONTH FROM DATE) = 4;
```

---

5) Find the Aadhar-numbers who have no benefits of Aadhar-service ?

```
SELECT AADHAR_NO FROM Registered_user

WHERE AADHAR_NO NOT IN (SELECT AADHAR_NO FROM
AVAIL_SERVICES);
```

---

6) Find how many number of people takes appointments in each enrollment center?

SELECT EC_ID,COUNT(APPOINTMENT_ID) FROM APPOINTMENT_FOR group by(EC_id);

---

7) Find the appointment timetable for particular EC on particular date.

SELECT APPOINTMENT_ID,DATE,TIME,UPDATE_TYPE,DATE,EC_ID

FROM APPOINTMENT NATURAL JOIN APPOINTMENT_FOR WHERE DATE = '2024-04-26',EC_id = 101;

---

8) Find the Aadhar-holder who don't have Aadhar linked bank account? (For any government subsidy or policy the Aadhar linked bank account is required.)

SELECT AADHAR_NO FROM APPOINTMENT WHERE AADHAR_NO NOT IN

(SELECT AADHAR_NO FROM AVAIL_SERVICES NATURAL JOIN AADHAR_SERVICES);

---

9) Find the Appointment ids where they wants to do updation of Particular update type.

SELECT EC_ID,APPOINTMENT_ID FROM APPOINTMENT_FOR

WHERE APPOINTMENT_ID IN

(SELECT APPOINTMENT_ID FROM APPOINTMENT_fOR

WHERE UPDATE_TYPE = 'Name Update');

---

10) Find the how many employes work in all ECs?

SELECT COUNT(E_ID) , EC_ID FROM EMPLOYEE NATURAL JOIN ENROLLMENT_CENTER GROUP BY EC_ID;

11) Find the details of all the employes.

```
SELECT E_NAME,E_PHONE_NO,DEPARTMENT_NAME FROM EMPLOYEE;
```

```
SELECT E_NAME,E_PHONE_NO,DEPARTMENT_NAME FROM EMPLOYEE WHERE E_ID = '1';(particular employe)
```

12) Find what percentage of the people have the Aadhar-card state-wise?

```
SELECT (C2*100)/C1 AS Percentage, S1 as STATE
FROM (
   SELECT DISTINCT STATE AS S1, COUNT(CITIZEN_NO) AS C1
   FROM (citizen
   NATURAL JOIN pin)
   GROUP BY (STATE)
) AS T1
JOIN (
   SELECT DISTINCT STATE AS S2, COUNT(CITIZEN_NO) AS C2
   FROM (registered_user
   NATURAL JOIN citizen
       natural join pin)
   GROUP BY STATE
) AS T2
ON T1.S1 = T2.S2;
```

QUERY FOR CITIZENS : -

1) Find the status of updating whether this is completed or not?

SELECT AADHAR_NO,STATUS_TYPE

FROM ONLINE_CHECK_STATUS;

2) Find the avail services status for particular Aadhar card.

SELECT * FROM PAYMENT NATURAL JOIN AVAIL_SERVICES WHERE AADHAR_NO = 111000000011;

3) Find the bank linked details for given Aadhar card. (for user to check whether his bank account is linked with bank or not?)

SELECT BANK_NAME ,BRANCH,ACCOUNT_NO,IFSC_CODE FROM BANK_ACCOUNT

NATURAL JOIN AADHAR_SERVICES NATURAL JOIN AVAIL_SERVICES

WHERE AADHAR_NO = 121000000021;

4) Find the enrollment center and it's type for given pincode.(for user find EC for appointment)

Select EC_ID,EC_TYPE from findS Natural join ENROLLMENT_CENTER where pincode ='given';

5) Find the documents required for do the particular updatation.

SELECT DOCUMENT FROM UPDATE_DOCUMENT WHERE UPDATE_TYPE = 'Biometric Update';

6) Find How many times the user has done the Aadhar updation and other history related to Aadhar card.

SELECT * FROM AVAIL_SERVICES NATURAL JOIN AADHAR_SERVICES NATURAL JOIN FETCH WHERE AADHAR_NO = ‘121000000021’

7) Find How many times the user has paid for the aadhar services or updating?

SELECT * FROM AVAIL_SERVICES NATURAL JOIN AADHAR_SERVICES NATURAL JOIN PAYMENT WHERE AADHAR_NO = ‘121000000021’

TOP-3 QUERIES

Queries For Government :-

1) Find the how much revenue generated month wise for given year(Here year = 2024)?

```
WITH R AS (
   SELECT EC_ID, DATE, update_FEE
   FROM APPOINTMENT
   NATURAL JOIN APPOINTMENT_FOR
   NATURAL JOIN UPDATE_REQUIReMENT
   WHERE EXTRACT('YEAR' FROM DATE) = 2024
)
SELECT EXTRACT('MONTH' FROM DATE) AS MONTH,
      SUM(update_FEE) AS REVENUE
FROM R
GROUP BY EXTRACT('MONTH' FROM DATE)
ORDER BY SUM(update_FEE) DESC;
```

OUTPUT :-

	month double precision	revenue bigint
1	5	3900
2	6	3750
3	7	2500
4	4	1650

2) Find what percentage of the people have the Aadhar-card state-wise?

```
SELECT (C2*100)/C1 AS Percentage, S1 as STATE
FROM (
    SELECT DISTINCT STATE AS S1, COUNT(CITIZEN_NO) AS C1
    FROM (citizen
    NATURAL JOIN pin)
    GROUP BY (STATE)
) AS T1
JOIN (
    SELECT DISTINCT STATE AS S2, COUNT(CITIZEN_NO) AS C2
    FROM (registered_user
    NATURAL JOIN citizen
        natural join pin)
    GROUP BY STATE
) AS T2
ON T1.S1 = T2.S2;
```

OUTPUT :-

	percentage bigint	state character varying (255)
1	50	Kerala
2	50	Dadra and Nagar Haveli and Daman and Diu
3	50	Goa
4	50	Gujarat
5	50	Haryana
6	50	Himachal Pradesh
7	50	Jammu and Kashmir
8	50	Andhra Pradesh

3) Find the bank linked details for given Aadhar card. (for user to check whether his bank account is linked with bank or not?)

SELECT BANK_NAME ,BRANCH,ACCOUNT_NO,IFSC_CODE FROM BANK_ACCOUNT

NATURAL JOIN AADHAR_SERVICES NATURAL JOIN AVAIL_SERVICES

WHERE AADHAR_NO = 121000000021;

OUTPUT :-

	bank_name character varying (255)	branch character varying (255)	account_no [PK] bigint	ifsc_code [PK] bigint
1	Bank of India	Delhi Main	345678678901	345678