

DBMS PROECT

TITLE - BANKING SYSTEM

Team members-

1)Sujay Gad PES1201700177

2)Mayur R Baggan PES1201700714

3)Jnanesh D PES12017001822

Introduction-

Our database is based on banking system. The database stores records of various employees ,customers ,accounts ,loans in various branches of a particular bank. The database stores and provides information in a systematic about all the attributes associated with a bank.

Purpose and Application of Database-

Banking system database can be used to compare performances of various branches in different aspects such as in which branch maximum business was done, Number of loans given to customers in a particular branch every year from 2011-2015.Database can also be used to compare various aspects of Employees working in different branches. From the information available in the database we can find out in which branch many loans,accounts are being created so that we can open up another branch in the same locality/city.

Miniworld-

The miniworld /universe of discourse for banking database is set of branches of a particular type of a bank located in various locations. Database stores information about various branches along their locations. Information is also stored about the Employees working in all these branches. Customer information along their respective accounts/loans details are also maintained in the database. The database also stores info about about various departments of a branch along with their respective locations inside the branch. Various type of analysis can be done with the help of data stored in the system.

Actions which can performed

On the database-

Insertion-

1)Each time new branch is opened info about that branch is stored in the branch table.

2)Whenever Customer takes a loan a new tuple is inserted into loan table.

Retrieval-

1)Names, age of all Employees having salary greater than 50k can be retrieved from the Employee table.

2)Names of all customers who have fixed deposit can be retrieved from account table.

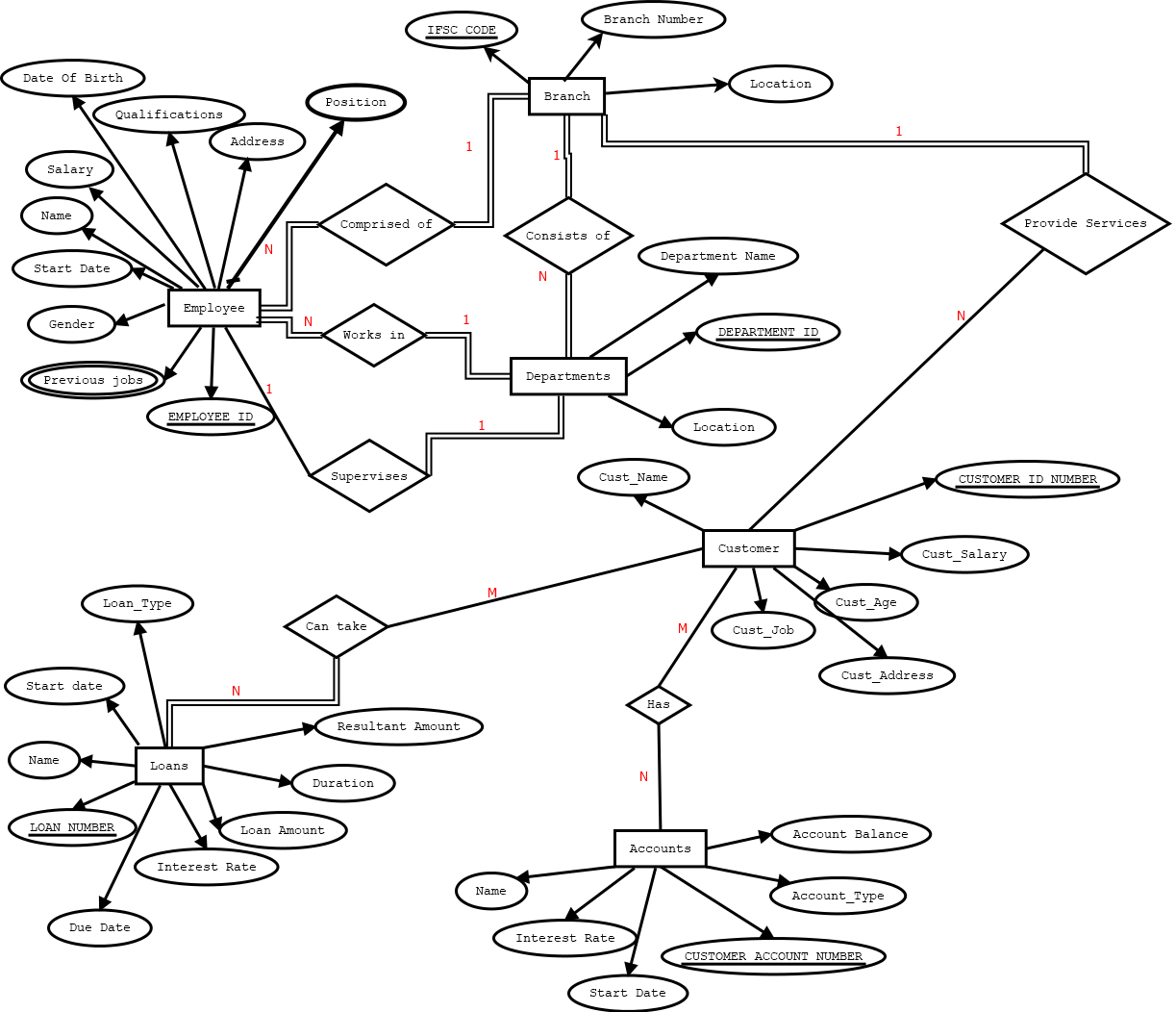
Update-

1)Suppose a particular branch has shifted to a different locality than its location has to be updated in the branch table.

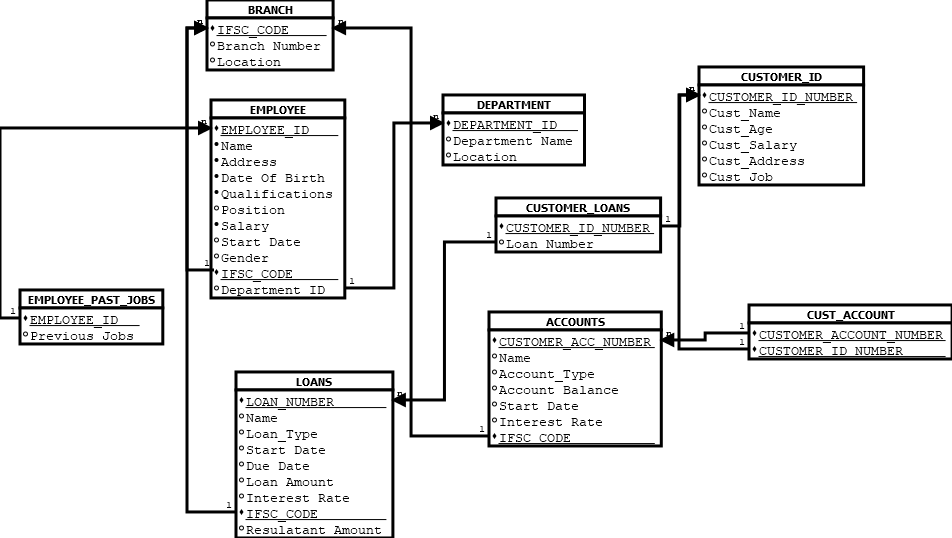
Delete-

1)If an Employee quits his job than his details are deleted from the Employee details since the information is no longer required.

**ER DIAGRAM**



Schema Diagram



Create table statements

CREATE TABLE BRANCH

(

IFSC\_CODE CHAR(11) NOT NULL,

Branch Number INT NOT NULL,

Location VARCHAR(30) NOT NULL,

PRIMARY KEY (IFSC\_CODE)

UNIQUE(Branch Number));

CREATE TABLE EMPLOYEE

(

EMPLOYEE\_ID CHAR(10) NOT NULL,

Name VARCHAR(30) NOT NULL,

Address VARCHAR(40) NOT NULL,

Date of Birth DATE,

Qualifications VARCHAR(20),

Position VARCHAR(20),

Salary DECIMAL(10,0),

Start Date DATE,

Gender VARCHAR(10),

IFSC\_CODE CHAR(11) NOT NULL,

DEPARTMENT\_ID INT NOT NULL,

PRIMARY KEY(EMPLOYEE\_ID),

FOREIGN KEY(IFSC\_CODE) REFERENCES BRANCH(IFSC\_CODE))

FOREIGN KEY(DEPARTMENT\_ID) REFERENCES DEPARTMENT(DEPARTMENT\_ID));

CREATE TABLE EMPLOYEE\_PAST\_JOBS

(

EMPLOYEE\_ID CHAR(10) NOT NULL,

Previous Jobs VARCHAR(20),

PRIMARY KEY(EMPLOYEE\_ID,Previous Jobs),

FOREIGN KEY(EMPLOYEE\_ID) REFERENCES EMPLOYEE(EMPLOYEE\_ID));

CREATE TABLE DEPARTMENT

(

DEPARTMENT\_ID INT NOT NULL,

Department Name VARCHAR(20) NOT NULL,

Location VARCHAR(20) NOT NULL,

PRIMARY KEY(DEPARTMENT\_ID),

UNIQUE(Department Name));

CREATE TABLE CUSTOMER

(

CUSTOMER\_ID\_NUMBER INT NOT NULL,

Customer\_Name VARCHAR(20) NOT NULL,

Customer\_Age INT,

Customer\_Salary INT,

Customer\_Address VARCHAR(30),

Customer\_Job VARCHAR(20),

PRIMARY KEY(CUSTOMER\_ID\_NUMBER));

CREATE TABLE ACCOUNT

(

CUSTOMER\_ACCOUNT\_NUMBER INT NOT NULL,

Name VARCHAR(20) NOT NULL,

Account\_Type VARCHAR(20) NOT NULL,

Account Balance DECIMAL(10,2) NOT NULL,

Start Date DATE NOT NULL,

Interest Rate VARCHAR(8) NOT NULL,

IFSC\_CODE CHAR(11) NOT NULL,

PRIMARY KEY(CUSTOMER\_ID\_NUMBER),

FOREIGN KEY(IFSC\_CODE) REFERENCES BRANCH(IFSC\_CODE));

CREATE TABLE CUSTOMER\_ACCOUNT

(

CUSTOMER\_ID\_NUMBER INT NOT NULL,

CUSTOMER\_ACCOUNT\_NUMBER INT NOT NULL,

PRIMARY KEY(CUSTOMER\_ID\_NUMBER,CUSTOMER\_ACCOUNT\_NUMBER),

FOREIGN KEY(CUSTOMER\_ID\_NUMBER) REFERENCES CUSTOMER(CUSTOMER\_ID\_NUMBER);

FOREIGN KEY(CUSTOMER\_ACCOUNT\_NUMBER) REFERENCES ACCOUNT(CUSTOMER\_ACCOUNT\_NUMBER));

CREATE TABLE LOAN

(

LOAN\_NUMBER INT NOT NULL,

Name VARCHAR(30) NOT NULL,

Loan\_Type VARCHAR(30) NOT NULL,

Start Date DATE NOT NULL,

Due Date DATE NOT NULL,

Loan Amount DECIMAL(10,2) NOT NULL,

Interest Rate VARCHAR(10) NOT NULL,

Resultant Amount DECIMAL(10,2) NOT NULL,

IFSC\_CODE CHAR(11) NOT NULL,

PRIMARY KEY(LOAN\_NUMBER),

FOREIGN KEY(IFSC\_CODE) REFERENCES BRANCH(IFSC\_CODE));

CREATE TABLE CUSTOMER\_LOAN

(

CUSTOMER\_ID\_NUMBER INT NOT NULL,

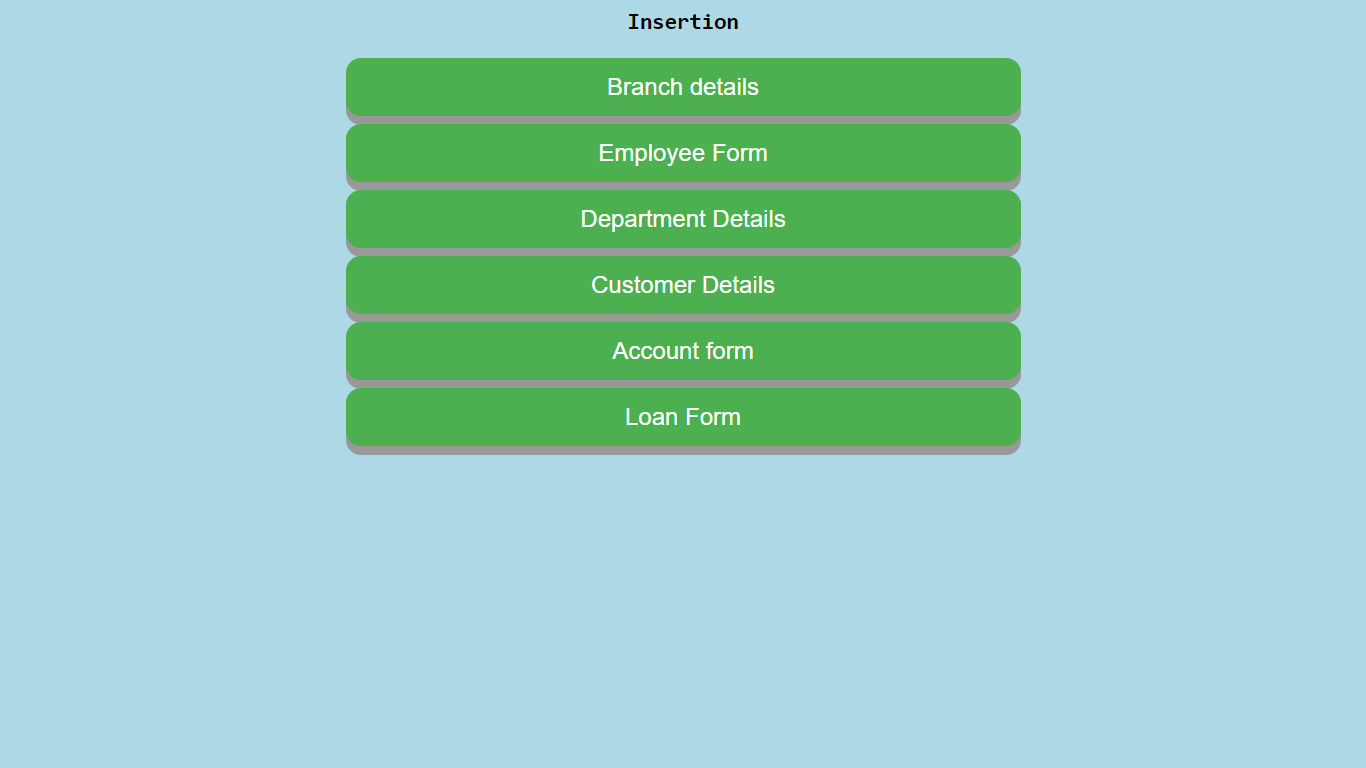
LOAN\_NUMBER INT NOT NULL,

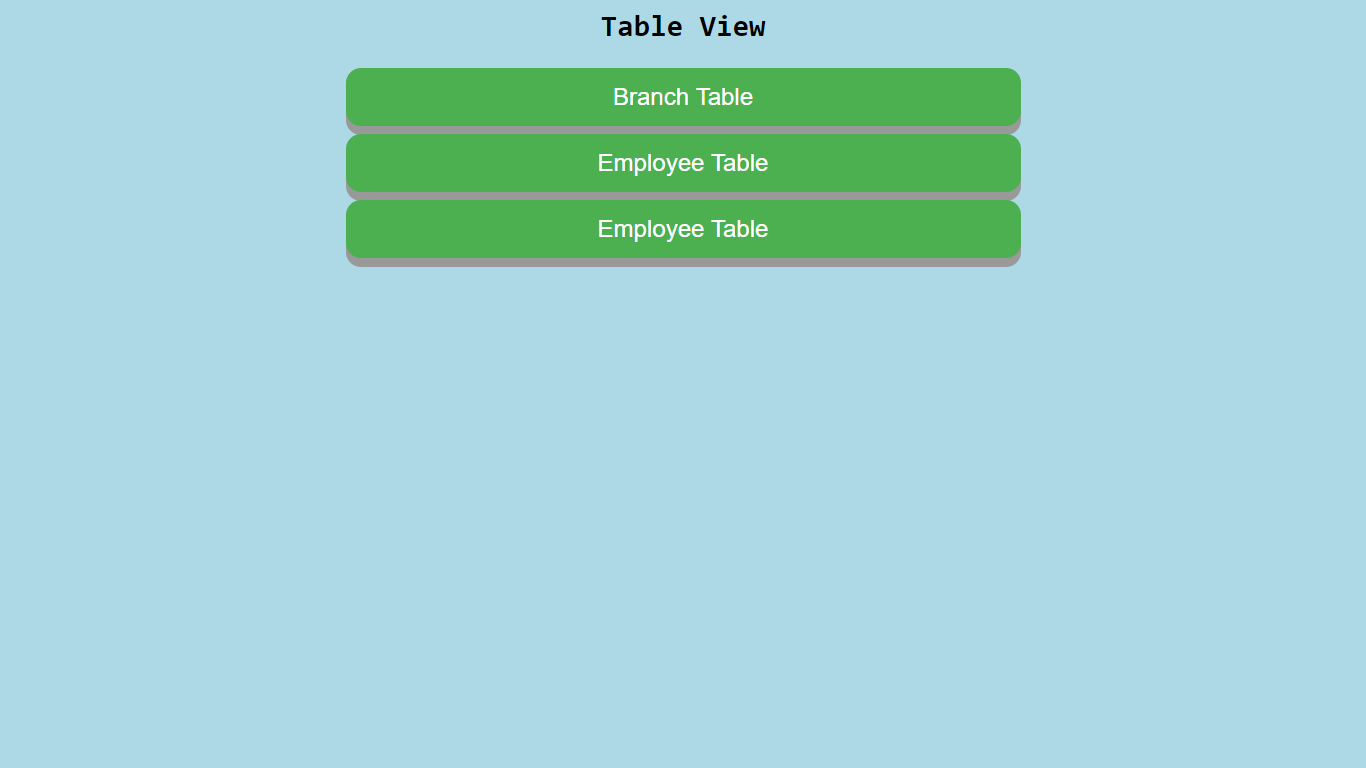
PRIMARY KEY(CUSTOMER\_ID\_NUMBER,LOAN\_NUMBER),

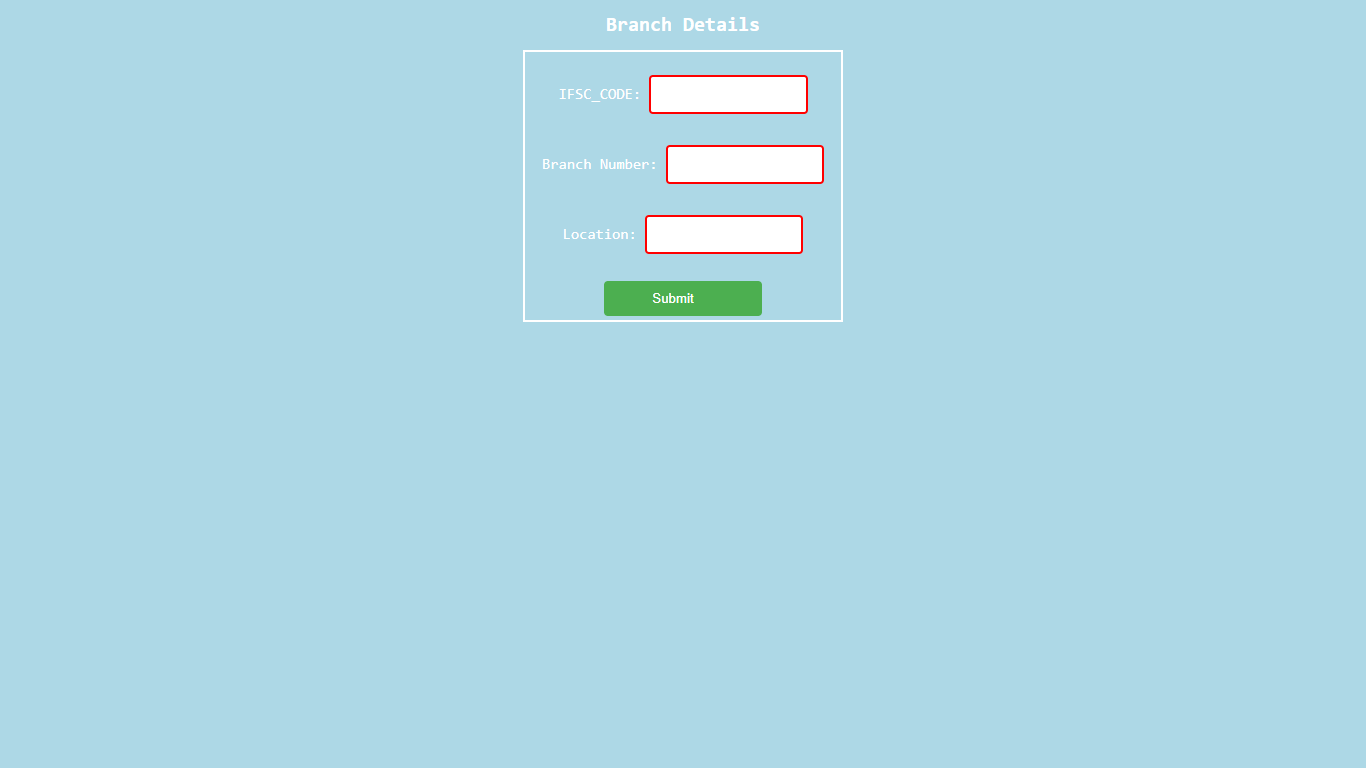
FOREIGN KEY(CUSTOMER\_ID\_NUMBER) REFERENCES CUSTOMER(CUSTOMER\_ID\_NUMBER);

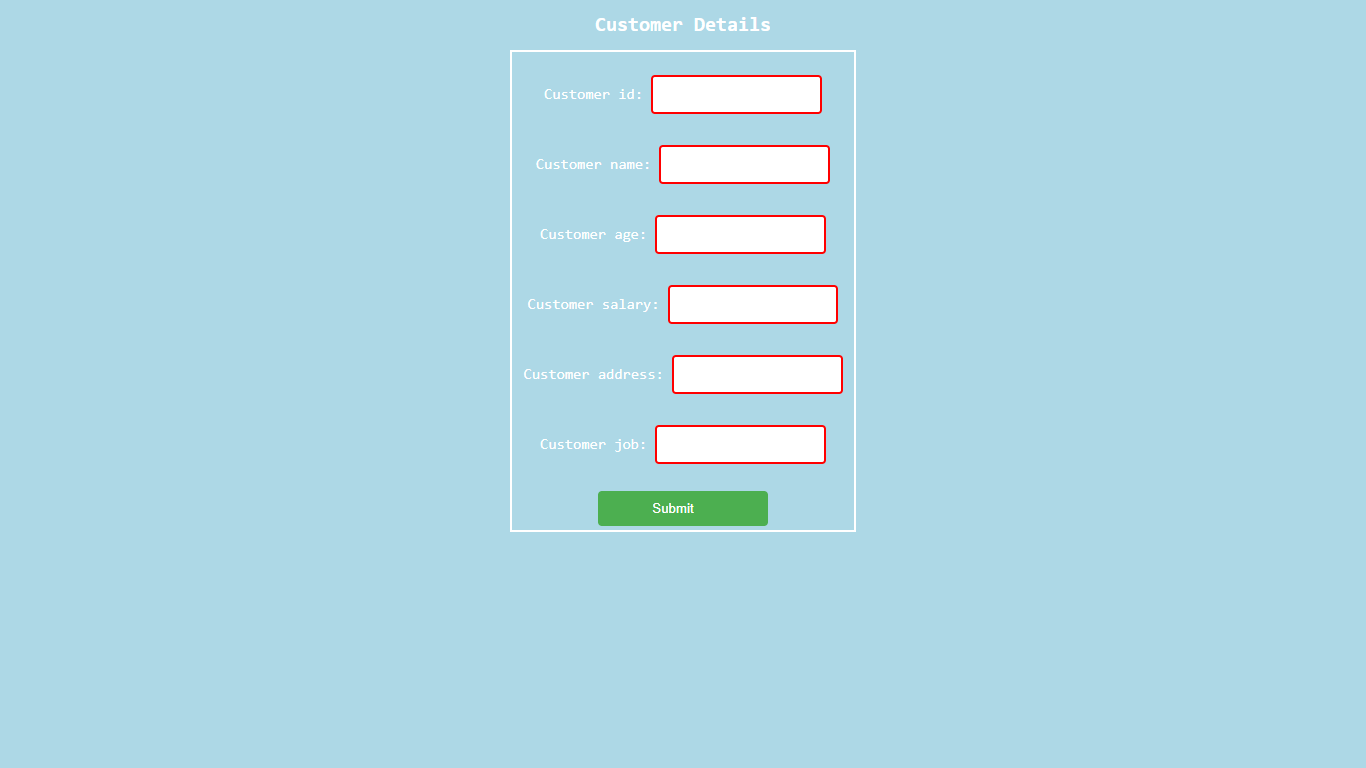
FOREIGN KEY(LOAN\_NUMBER) REFERENCES LOAN(LOAN\_NUMBER));

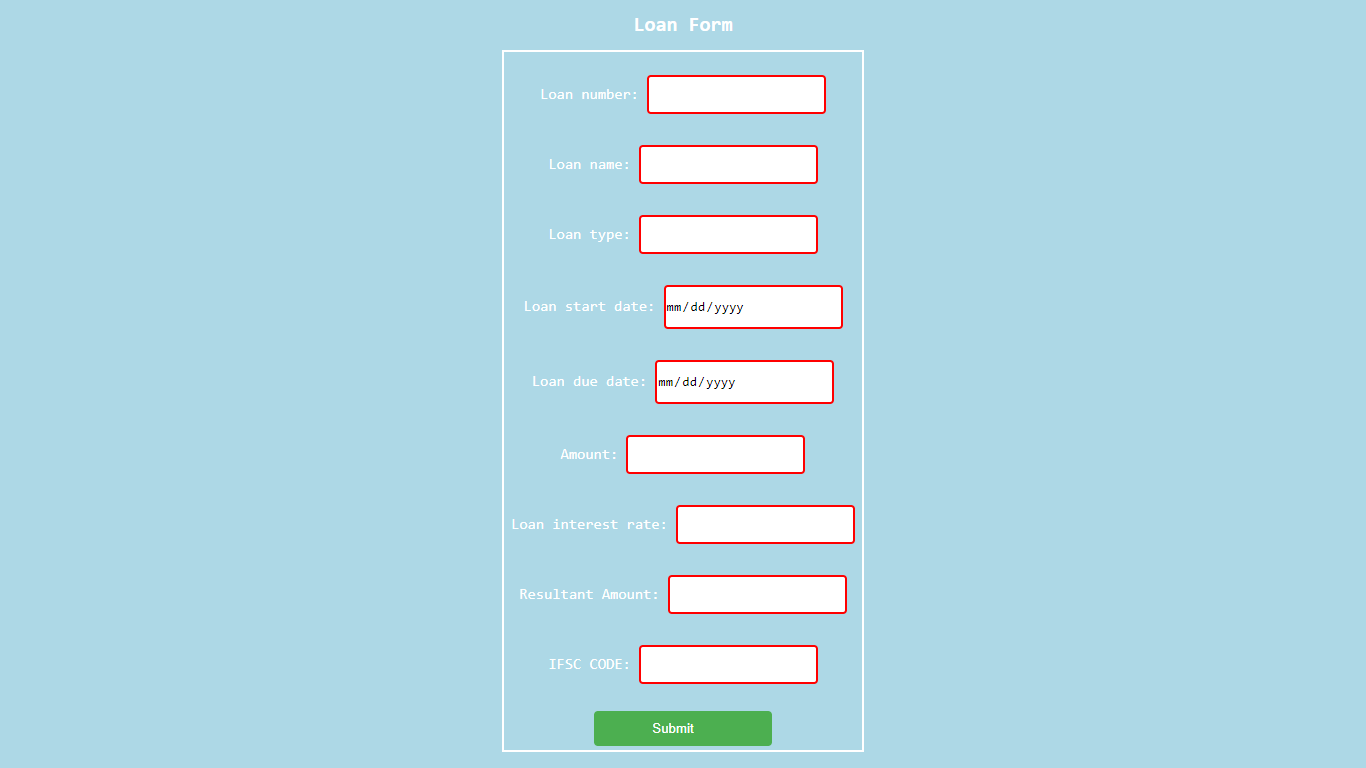
Php Forms/UI Screens



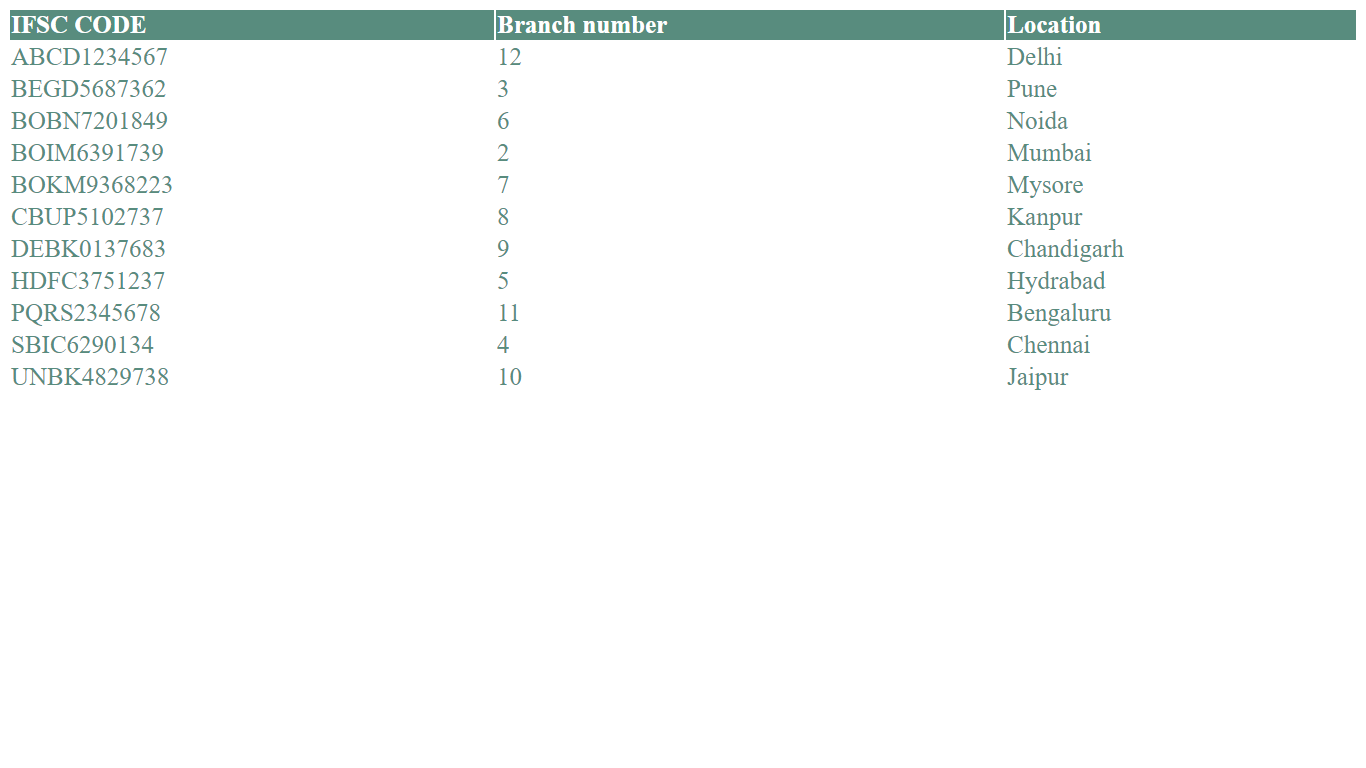


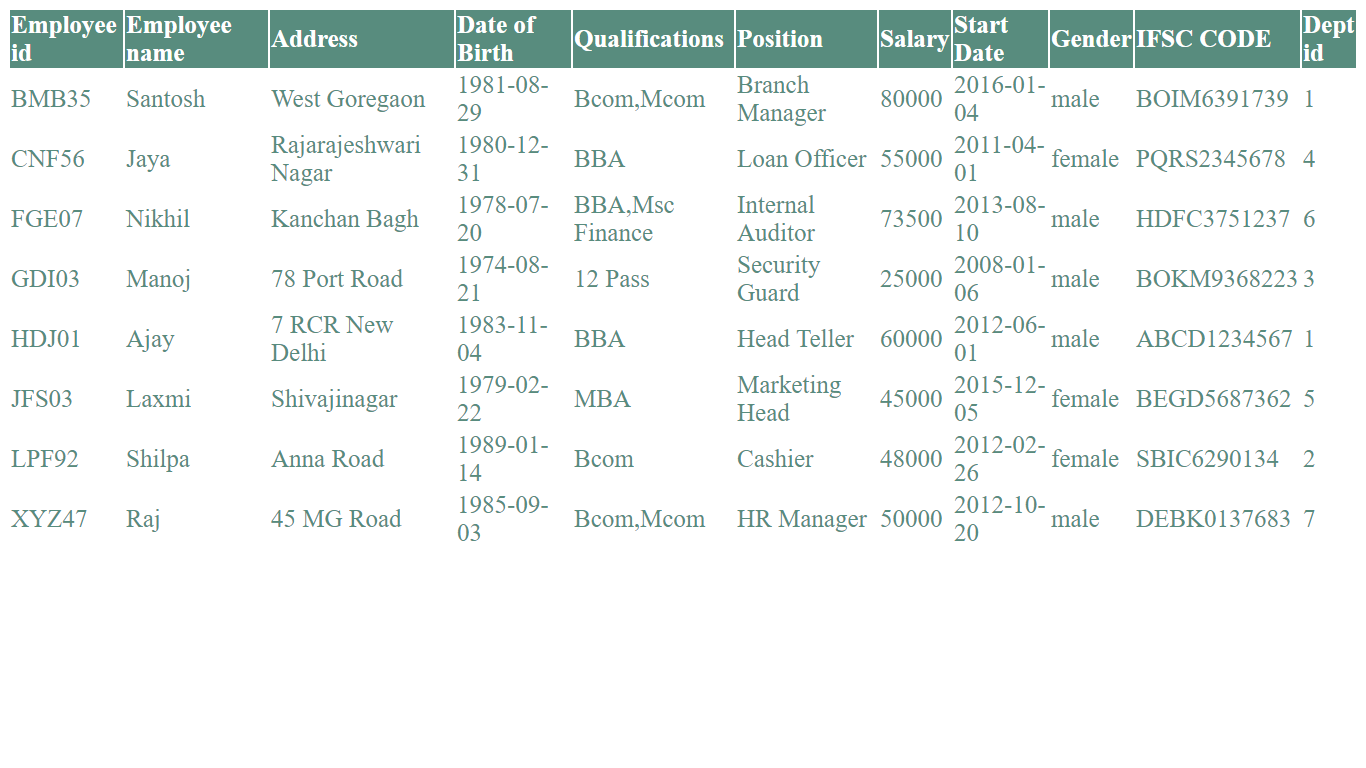


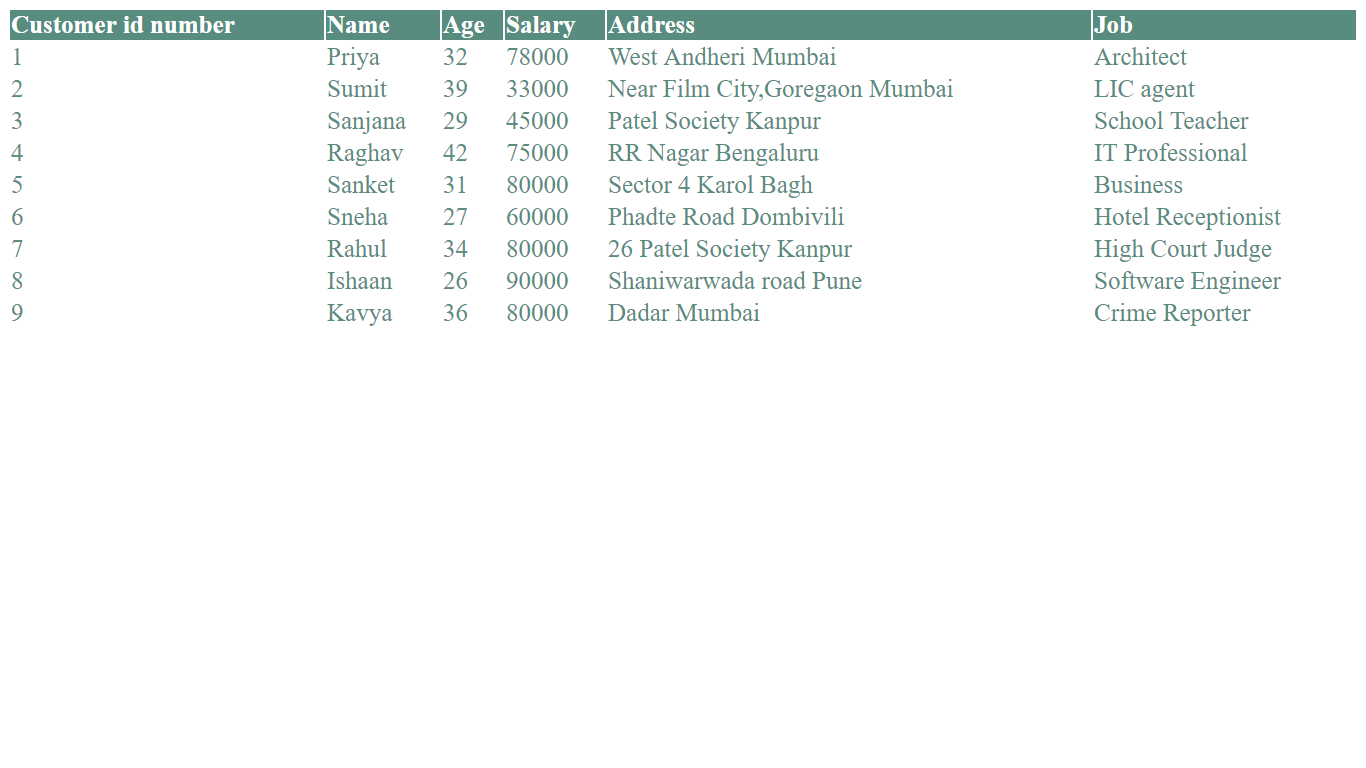




Tables







Simple SQL Queries

INSERT Statements-

INSERT INTO BRANCH VALUES('GIRS9879873',6,'Pune');

INSERT INTO EMPLOYEE VALUES('SQA50','Neha','78 Port Road','1984-03-09','BBA,Mcom','Head Teller',55000,'2013-06-16','Female','BOIA9921461',7);

INSERT INTO EMPLOYEE\_PAST\_JOBS('HMS79','Regional Head');

INSERT INTO DEPARTMENT VALUES('002','Account','Ground Floor');

INSERT INTO CUSTOMER VALUES(20,'Tanmay',28,56000,'Bagmari Road Ghosh Bagan Kolkata','Architect');

INSERT INTO ACCOUNT VALUES('RD0000000001','Raj','Recurring Deposit',28400,'2009-12-31','6.00%','BOBM9023947');

INSERT INTO CUSTOMER\_ACCOUNT VALUES(2,'FD0000000001');

INSERT INTO LOAN VALUES('CRED000001','Vinay','Credit Card loan','2018-04-01','2018-10-01',50000,'15%',53750,'DLTM738275O');

INSERT INTO CUSTOMER\_LOAN VALUES(11,'CRED000001');

ALTER statements-

ALTER TABLE EMPLOYEE ALTER COLUMN POSITION TYPE VARCHAR(15);

ALTER TABLE CUSTOMER DROP COLUMN CUSTOMER\_AGE;

ALTER TABLE EMPLOYEE MODIFY DATE\_OF\_BIRTH DATE NOT NULL;

ALTER TABLE CUSTOMER\_LOAN DROP CONSTRAINT customer\_loan\_customer\_customer\_id\_number\_fkey;

DELETE statements-

DELETE FROM CUSTOMER\_ACCOUNT WHERE CUST\_ID\_NUMBER=2;

DELETE FROM EMPLOYEE WHERE em\_id='HAB93';

DELETE FROM CUSTOMER\_LOAN CASCADE;

UPDATE statements-

UPDATE LOAN SET loan\_type='Home Loan' WHERE name='Vinay';

UPDATE EMPLOYEE SET address='7th Block Indiranagar' WHERE em\_id='ABC01';

UPDATE ACCOUNT SET interest\_rate='4.50%' WHERE account\_type='Savings Account';

UPDATE DEPARTMENT SET location='First Floor' WHERE department name='Loan';

Complex SQL Queries

1)

SELECT customer\_name,customer\_age

FROM customer

WHERE NOT(EXISTS(SELECT cust\_id\_number

FROM customer\_account

WHERE customer\_id\_number=cust\_id\_number)

OR

EXISTS(SELECT c\_id\_number

FROM customer\_loan

WHERE customer\_id\_number=c\_id\_number));

2)

SELECT AVG(customer\_salary) AS Salary,AVG(loan\_amount) AS Loan\_Amount

FROM customer as C,Loan AS L

WHERE C.customer\_name IN( SELECT L.name

FROM LOAN

WHERE C.customer\_name=L.name AND C.customer\_salary>=50000);

3)

SELECT account\_name,AVG(account\_balance),count(\*) AS no\_of\_accounts

FROM ACCOUNT

WHERE account\_balance>=50000

GROUP BY account\_name

HAVING count(\*)>1;

4)

SELECT em\_name,address,date\_of\_birth,salary

FROM EMPLOYEE

Where (SELECT COUNT(\*)

FROM EMPLOYEE\_PAST\_JOBS

WHERE employee\_id=em\_id)>=2;

5)

SELECT em\_id,em\_name,salary,gender,branch\_location

FROM (EMPLOYEE JOIN BRANCH ON ifsc=ifsc\_code)

WHERE branch\_location LIKE 'Delhi' OR branch\_location LIKE 'Mumbai';

6)

SELECT customer\_name,customer\_age,account\_type,account\_balance

FROM ((CUSTOMER LEFT OUTER JOIN ACCOUNT ON customer\_name=account\_name) LEFT OUTER JOIN CUSTOMER\_ACCOUNT ON customer\_account\_number=cust\_account\_number)

WHERE start\_date<='2011-10-10';

7)

SELECT customer\_name,customer\_address,loan\_type,loan\_amount,branch\_location

FROM (((LOAN LEFT OUTER JOIN CUSTOMER ON name=customer\_name) JOIN BRANCH ON ifsc\_=ifsc\_code) JOIN CUSTOMER\_LOAN ON loan\_number=l\_number)

WHERE branch\_location IN( SELECT branch\_location

FROM BRANCH

WHERE branch\_location <>'Mumbai');

8)

SELECT MAX(customer\_age) AS max\_age,MIN(customer\_age) AS min\_age,MAX(customer\_salary) AS max\_salary,MIN(customer\_salary) AS min\_salary

FROM CUSTOMER

WHERE customer\_name IN(SELECT account\_name

FROM ACCOUNT

EXCEPT(SELECT account\_name

FROM ACCOUNT

WHERE account\_type='Recurring Deposit'));

9)

CREATE ASSERTION DATE\_CONSTRAINT

CHECK(NOT EXISTS(SELECT \*

FROM LOAN

WHERE loan\_start\_date>=loan\_due\_date));

Conclusion-

Banking system database has wide implications, applications in real life since millions of transactions occur everyday in banks around the world.Large amount of records should be stored in a systematic way in a banking system.