UE17CS252

Database Management Systems

PROJECT REPORT

Prof. Raghu B.A. Rao

4 ‘F’

B. Tech -CSE

BANK DATABASE MANAGEMENT SYSTEM

PES1201700218	**Sharanya Venkat**
PES1201700301	**Anagha Ananth**
PES1201700872	**Aishwarya M A Ramanath**

TABLE OF CONTENTS

INTRODUCTION

Banks deal with huge volumes of varieties of data - all of which needs to be retrieved and updated in as little time as possible. Bank managers and employees require a quick and hassle free way to retrieve information from their client accounts for their customers, tax purposes, advertising schemes, financial balance, keeping track of suspicious activity etc. They need a system that allows them to access all this information, a banking management system.

PURPOSE

Our project aims to provide the bank (user) with the required resources to keep track of each customer's account, hence allowing easy accessibility to all financial transactions that links the customer and their respective bank account. This includes essential personal details of the customer the account number, branch, the main employee assigned to each customer, loans withstanding and credit cards linked to the account. Using our management system, more detailed information regarding each of these sectors will be easily available to the user. Relations between each of these important aspects are clearly defined, so as to maintain a logical system which can be accessed from different points of view depending on the information that is required to be gathered at the moment by the user.

THE MINI WORLD

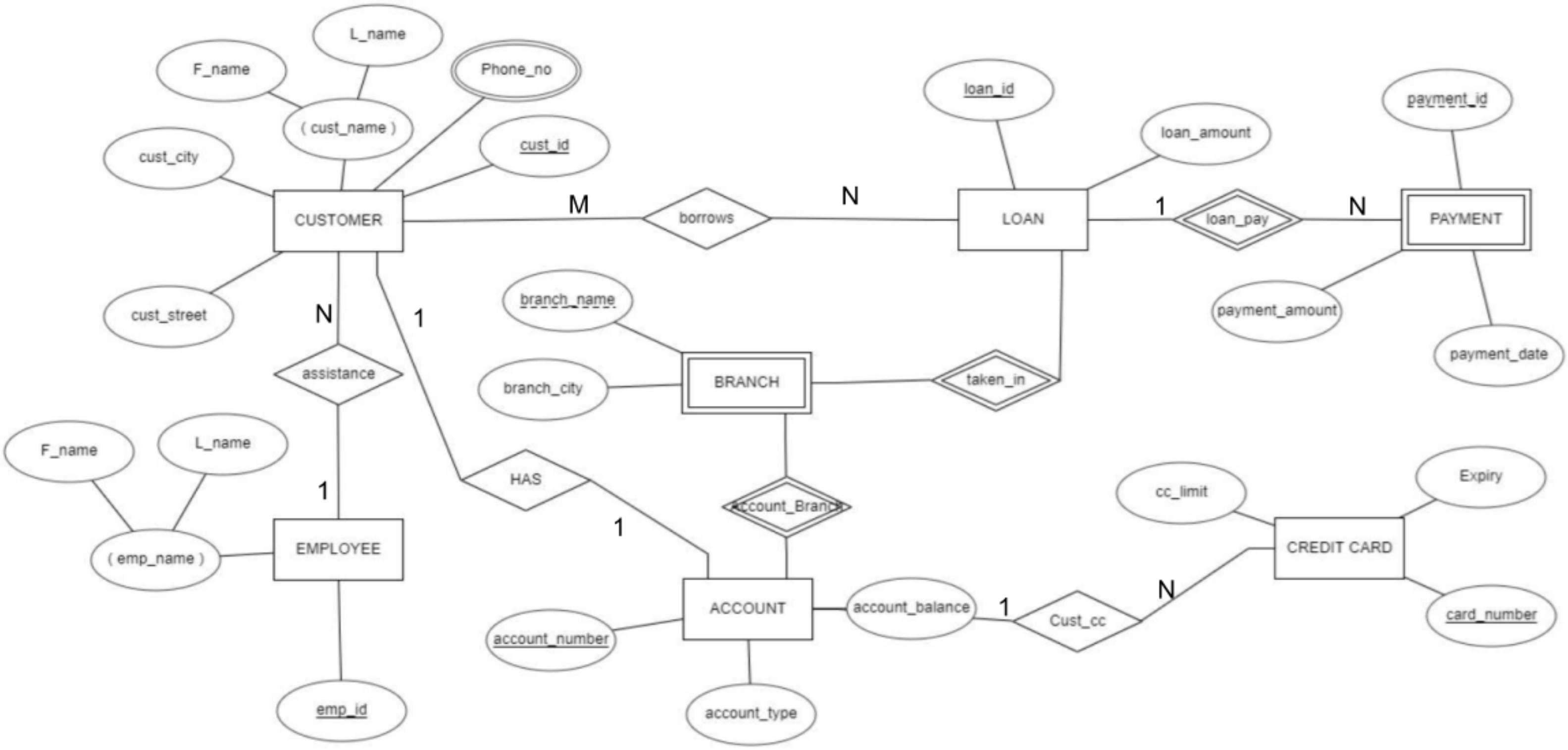
Our mini world represents a bank - managed by its employees and containing the following data:

1. Employees and their IDs
2. Customers and their Home Addresses, phone numbers and IDs
3. Branches of the bank in different cities
4. Accounts created in the bank, their type and amount withstanding
5. Credit Cards issued by the bank, their expiry date and withdrawal limit
6. Loans borrowed from the bank and their payment details

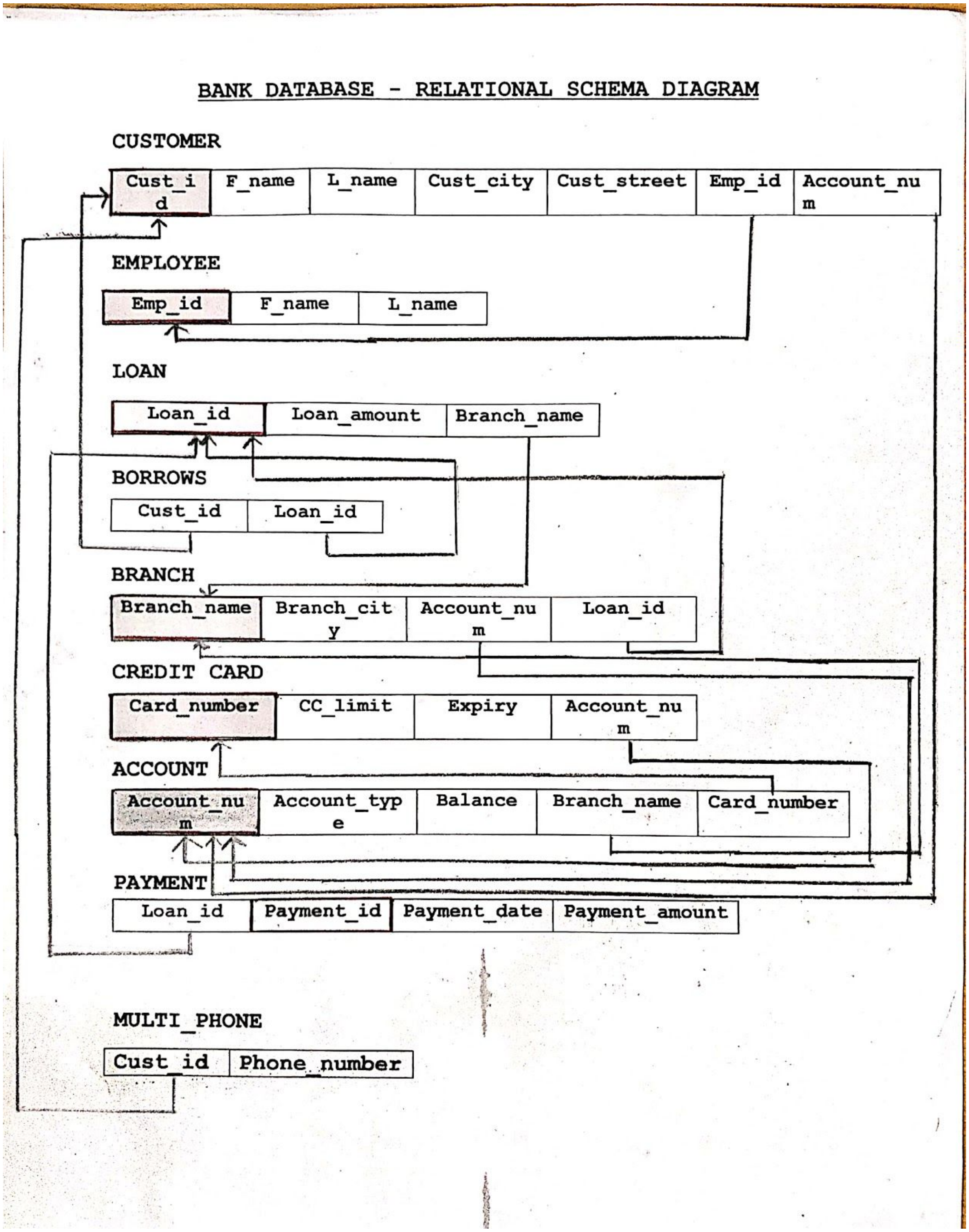
REQUIREMENTS

1. Given the customer's ID, the user can view the customer's personal, account, loan and credit card details. He can assign the customer a new credit card and change the branch in which the customer's account is situated.
2. The average and total balances of all the various bank branches, and the bank with the highest average balance can be retrieved.
3. A lucky draw feature is implemented to provide employees with bonuses
4. A list of the most recent payments made by the customers can be retrieved
5. Based on specific conditions, elite customers are listed

ER DIAGRAM



SCHEMA DIAGRAM



BANK DATABASE - RELATIONAL SCHEMA DIAGRAM

CUSTOMER

Cust_id	F_name	L_name	Cust_city	Cust_street	Emp_id	Account_num

EMPLOYEE

Emp_id	F_name	L_name

LOAN

Loan_id	Loan_amount	Branch_name

BORROWS

Cust_id	Loan_id

BRANCH

Branch_name	Branch_city	Account_num	Loan_id

CREDIT CARD

Card_number	CC_limit	Expiry	Account_num

ACCOUNT

Account_num	Account_type	Balance	Branch_name	Card_number

PAYMENT

Loan_id	Payment_id	Payment_date	Payment_amount

MULTI_PHONE

Cust_id	Phone_number

INSERT STATEMENTS:

```
INSERT INTO LOAN(loan_amount,loan_id,branch_name_loan_fkey)

VALUES(900000,'2345','MG Road');

INSERT INTO LOAN(loan_amount,loan_id,branch_name_loan_fkey)

VALUES(1500000,'8976','MG Road');

INSERT INTO LOAN(loan_amount,loan_id,branch_name_loan_fkey)

VALUES(2400000,'3269','Mysore Road');

INSERT INTO LOAN(loan_amount,loan_id,branch_name_loan_fkey)

VALUES(50000,'0214','Palace Road');



INSERT INTO
ACCOUNT(account_number,balance,account_type,credit_card_number_
account_fkey)

VALUES('2876',200000,'Savings','6789 5673 9734 2122');

INSERT INTO
ACCOUNT(account_number,balance,account_type,credit_card_number_
account_fkey)

VALUES('3452',1300000,'Savings','9823 5769 9438 5200');

INSERT INTO
ACCOUNT(account_number,balance,account_type,credit_card_number_
account_fkey)

VALUES('9846',2500000,'Savings','8459 2904 7826 7768');

INSERT INTO
ACCOUNT(account_number,balance,account_type,credit_card_number_
account_fkey)
```

```
VALUES('2132',1200000,'Savings','2045 6721 8953 2134');

INSERT INTO
ACCOUNT(account_number,balance,account_type,credit_card_number_
account_fkey)

VALUES('6754',50000,'Savings','9176 5490 3478 2086');



INSERT INTO borrows(cust_id,loan_id)

VALUES('10001','1948');

INSERT INTO borrows(cust_id,loan_id)

VALUES('20024','2345');

INSERT INTO borrows(cust_id,loan_id)

VALUES('20095','8976');

INSERT INTO borrows(cust_id,loan_id)

VALUES('30018','3269');

INSERT INTO borrows(cust_id,loan_id)

VALUES('90137','0214');


INSERT INTO
PAYMENT(payment_id,payment_amount,payment_date,loan_id_payment_
fkey)

VALUES('1234', 25000,'1-12-19','1948');

INSERT INTO
PAYMENT(payment_id,payment_amount,payment_date,loan_id_payment_
fkey)
```

```
VALUES('2344', 11000,'20-3-18','2345');

INSERT INTO
PAYMENT(payment_id,payment_amount,payment_date,loan_id_payment_fkey)

VALUES('3456', 57000,'9-12-18','8976');

INSERT INTO
PAYMENT(payment_id,payment_amount,payment_date,loan_id_payment_fkey)

VALUES('5678', 100000,'24-11-18','3269');

INSERT INTO
PAYMENT(payment_id,payment_amount,payment_date,loan_id_payment_fkey)

VALUES('6789', 3500,'3-2-19','0214');


INSERT INTO
BRANCH(branch_city,branch_name,branch_account_number,branch_loan_id)

VALUES('Bangalore','MG Road','9846','1948');

INSERT INTO
BRANCH(branch_city,branch_name,branch_account_number,branch_loan_id)

VALUES('Bangalore','MG Road','3452','2345');

INSERT INTO
BRANCH(branch_city,branch_name,branch_account_number,branch_loan_id)

VALUES('Bangalore','Mysore Road','2132','8976');

INSERT INTO
```

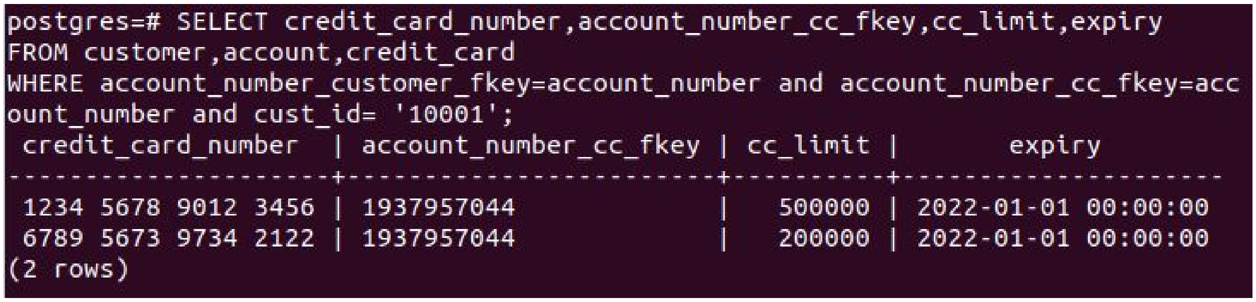
```
BRANCH(branch_city,branch_name,branch_account_number,branch_loan_id)

VALUES('Bangalore','Mysore Road','2876','3269');

INSERT INTO
BRANCH(branch_city,branch_name,branch_account_number,branch_loan_id)

VALUES('Mysore','Palace Road','6754','0214');
```

Note: Further values were added to the database by exporting CSV Files

SIMPLE SQL QUERIES AND DDL STATEMENTS

Display all credit cards linked to a particular customer's account:

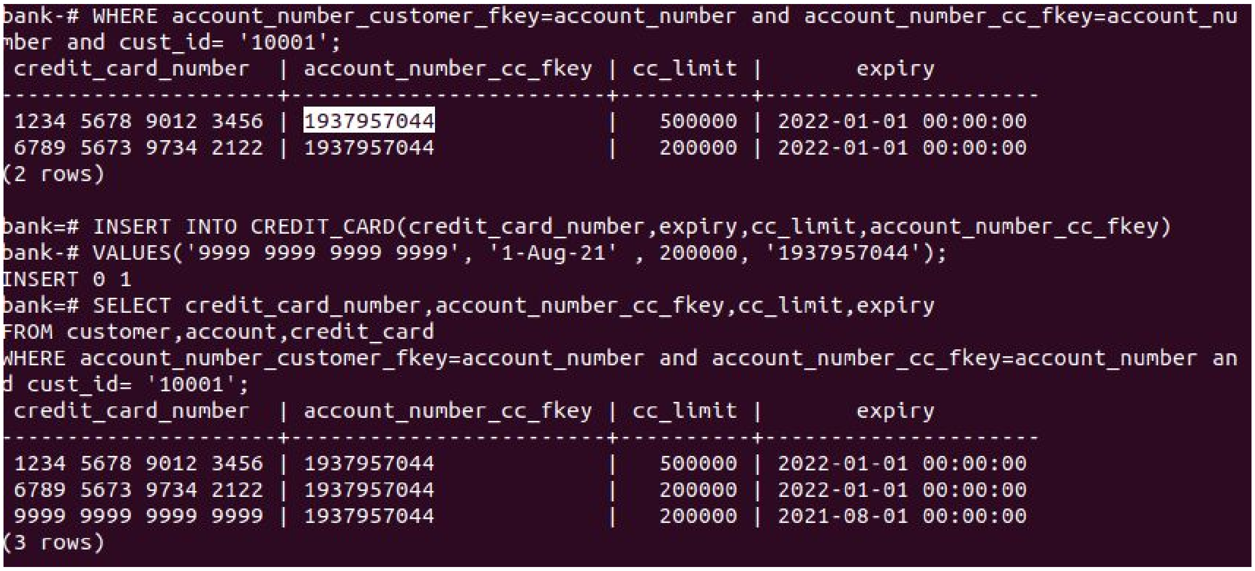
```
SELECT
credit_card_number,account_number_cc_fkey,cc_limit,expiry
FROM customer,account,credit_card
WHERE account_number_customer_fkey=account_number and
account_number_cc_fkey=account_number and cust_id= your_cust_id;
```



```
postgres=# SELECT credit_card_number,account_number_cc_fkey,cc_limit,expiry
FROM customer,account,credit_card
WHERE account_number_customer_fkey=account_number and account_number_cc_fkey=acc
ount_number and cust_id= '10001';
 credit_card_number  | account_number_cc_fkey | cc_limit |       expiry
---------------------+------------------------+----------+---------------------
 1234 5678 9012 3456 | 1937957044             |   500000 | 2022-01-01 00:00:00
 6789 5673 9734 2122 | 1937957044             |   200000 | 2022-01-01 00:00:00
(2 rows)
```

Add a new credit card to a customer's account:

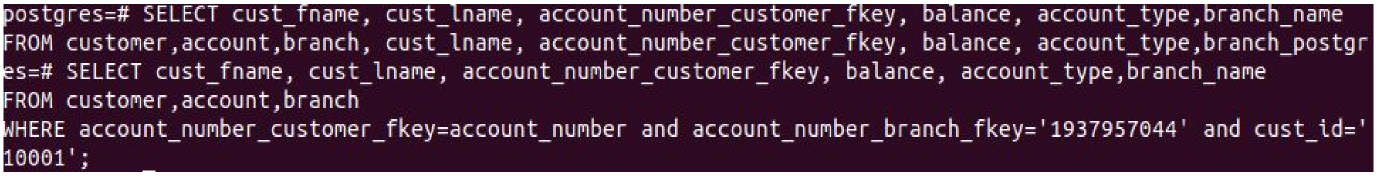
```
INSERT INTO
CREDIT_CARD(credit_card_number,expiry,cc_limit,account_number_cc_fkey)
VALUES(generated_credit_card_no, '1-Aug-21' , desired_limit,
customer’s_acc_no);
```



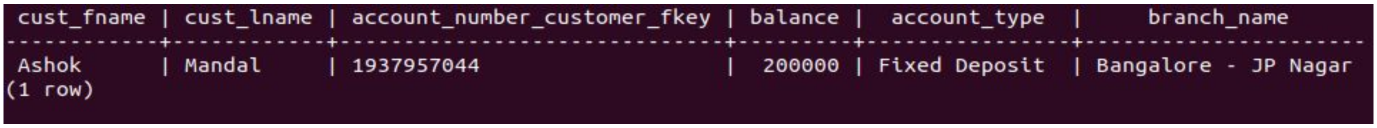
```
bank-# WHERE account_number_customer_fkey=account_number and account_number_cc_fkey=account_nu
mber and cust_id= '10001';
 credit_card_number  | account_number_cc_fkey | cc_limit |       expiry
---------------------+------------------------+----------+---------------------
 1234 5678 9012 3456 | 1937957044             |   500000 | 2022-01-01 00:00:00
 6789 5673 9734 2122 | 1937957044             |   200000 | 2022-01-01 00:00:00
(2 rows)

bank=# INSERT INTO CREDIT_CARD(credit_card_number,expiry,cc_limit,account_number_cc_fkey)
bank-# VALUES('9999 9999 9999 9999', '1-Aug-21' , 200000, '1937957044');
INSERT 0 1
bank=# SELECT credit_card_number,account_number_cc_fkey,cc_limit,expiry
FROM customer,account,credit_card
WHERE account_number_customer_fkey=account_number and account_number_cc_fkey=account_number an
d cust_id= '10001';
 credit_card_number  | account_number_cc_fkey | cc_limit |       expiry
---------------------+------------------------+----------+---------------------
 1234 5678 9012 3456 | 1937957044             |   500000 | 2022-01-01 00:00:00
 6789 5673 9734 2122 | 1937957044             |   200000 | 2022-01-01 00:00:00
 9999 9999 9999 9999 | 1937957044             |   200000 | 2021-08-01 00:00:00
(3 rows)
```

Display all customer details given his customer ID

```
SELECT cust_fname, cust_lname, account_number_customer_fkey,
balance, account_type,branch_name
FROM customer,account,branch
WHERE account_number_customer_fkey=account_number and
account_number_branch_fkey=account_number and
cust_id=your_cust_id;
```



```
postgres=# SELECT cust_fname, cust_lname, account_number_customer_fkey, balance, account_type,branch_name
FROM customer,account,branch, cust_lname, account_number_customer_fkey, balance, account_type,branch_postgr
es=# SELECT cust_fname, cust_lname, account_number_customer_fkey, balance, account_type,branch_name
FROM customer,account,branch
WHERE account_number_customer_fkey=account_number and account_number_branch_fkey='1937957044' and cust_id='
10001';
```

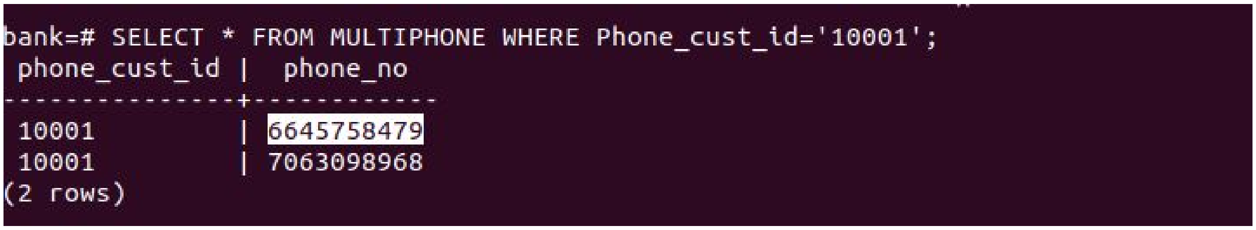


```
 cust_fname | cust_lname | account_number_customer_fkey | balance |  account_type  |     branch_name
------------+------------+------------------------------+---------+----------------+----------------------
 Ashok      | Mandal     | 1937957044                   |  200000 | Fixed Deposit  | Bangalore - JP Nagar
(1 row)
```

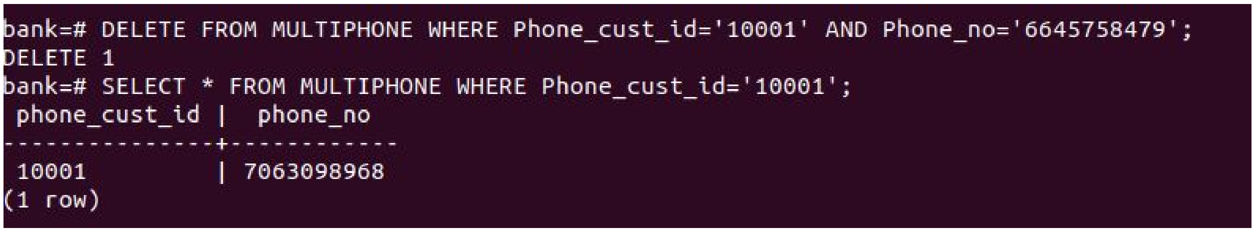
Delete phone number given customer ID:

```
DELETE FROM MULTIPHONE WHERE Phone_cust_id='" . $cid . "' AND
```

```
Phone_no='" . $phone . "';
```



```
bank=# SELECT * FROM MULTIPHONE WHERE Phone_cust_id='10001';
 phone_cust_id |  phone_no
---------------+------------
 10001         | 6645758479
 10001         | 7063098968
(2 rows)
```

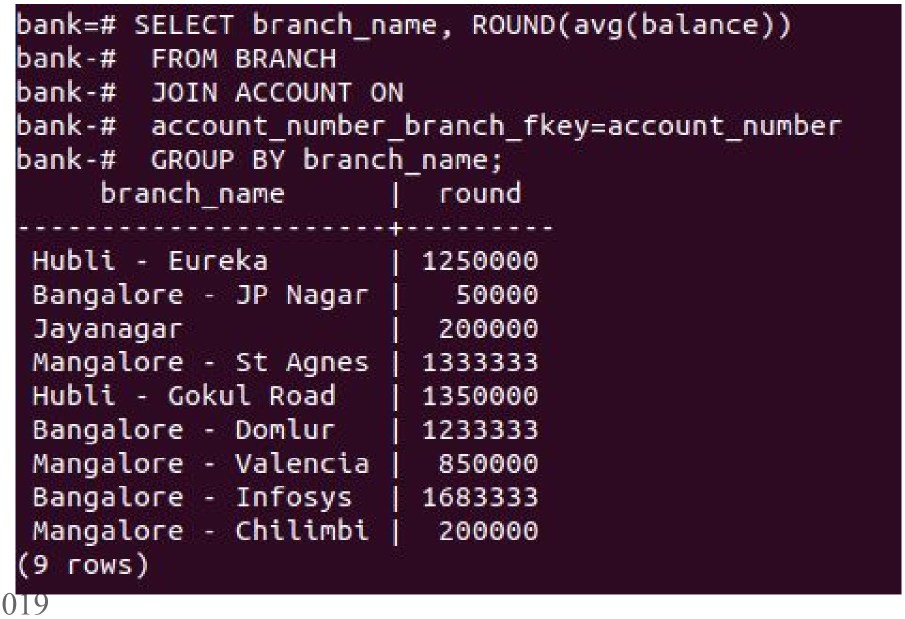


```
bank=# DELETE FROM MULTIPHONE WHERE Phone_cust_id='10001' AND Phone_no='6645758479';
DELETE 1
bank=# SELECT * FROM MULTIPHONE WHERE Phone_cust_id='10001';
 phone_cust_id |  phone_no
---------------+------------
 10001         | 7063098968
(1 row)
```

COMPLEX SQL QUERIES

Find the average balances of each branch:

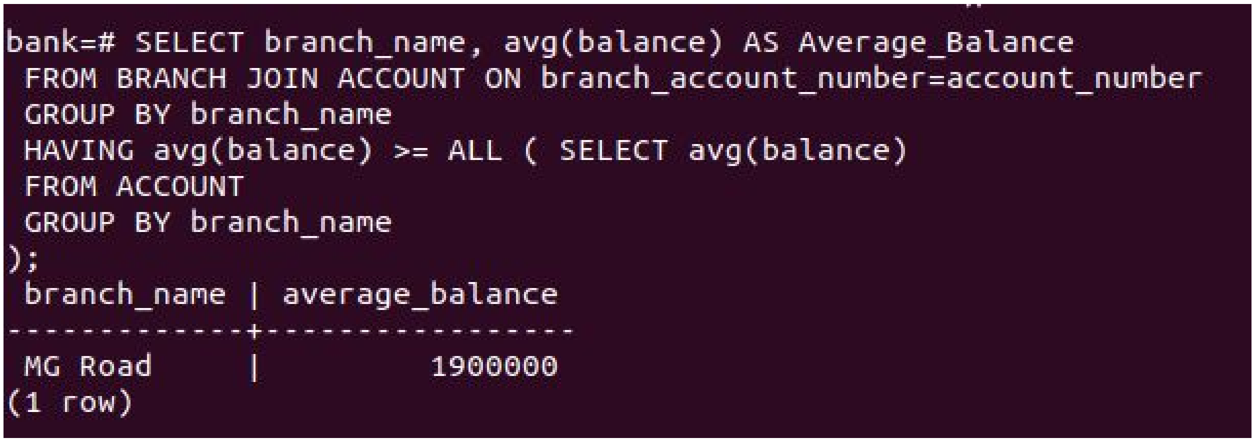
```
SELECT branch_name, ROUND(avg(balance))
FROM BRANCH
JOIN ACCOUNT ON
account_number_branch_fkey=account_number
GROUP BY branch_name;
```



```
bank=# SELECT branch_name, ROUND(avg(balance))
bank-#  FROM BRANCH
bank-#  JOIN ACCOUNT ON
bank-#  account_number_branch_fkey=account_number
bank-#  GROUP BY branch_name;
     branch_name      |  round
----------------------+---------
 Hubli - Eureka       | 1250000
 Bangalore - JP Nagar |   50000
 Jayanagar            |  200000
 Mangalore - St Agnes | 1333333
 Hubli - Gokul Road   | 1350000
 Bangalore - Domlur   | 1233333
 Mangalore - Valencia |  850000
 Bangalore - Infosys  | 1683333
 Mangalore - Chilimbi |  200000
(9 rows)
```

Find the bank with the highest average balance

```
SELECT branch_name, avg(balance) as maximum_average
FROM BRANCH JOIN ACCOUNT ON
account_number_branch_fkey=account_number
GROUP BY branch_name

HAVING avg(balance) =

(SELECT  max(x.avg) as maximum_average FROM
     (SELECT avg(balance) FROM
     BRANCH JOIN ACCOUNT ON
account_number_branch_fkey=account_number

GROUP BY branch_name

 )x

);
```

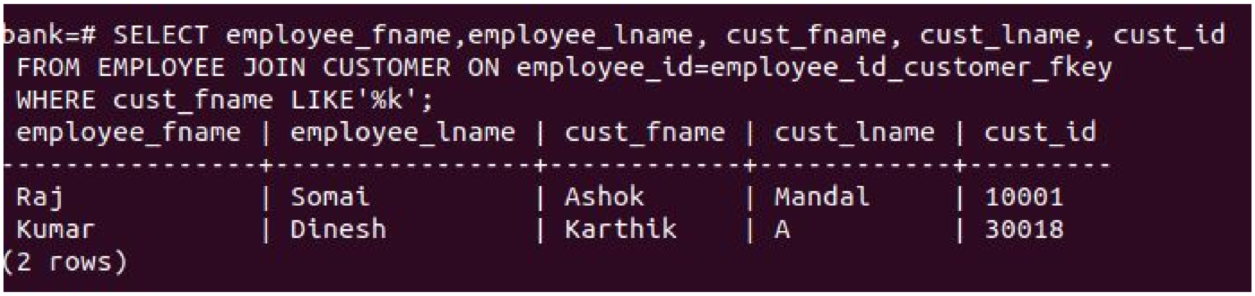


```
bank=# SELECT branch_name, avg(balance) AS Average_Balance
 FROM BRANCH JOIN ACCOUNT ON branch_account_number=account_number
 GROUP BY branch_name
 HAVING avg(balance) >= ALL ( SELECT avg(balance)
 FROM ACCOUNT
 GROUP BY branch_name
);
 branch_name | average_balance
-------------+----------------
 MG Road     |         1900000
(1 row)
```

Find the employees associated with customers whose first names end with ‘k’

```
SELECT employee_fname,employee_lname, cust_fname, cust_lname
FROM EMPLOYEE JOIN CUSTOMER ON
```

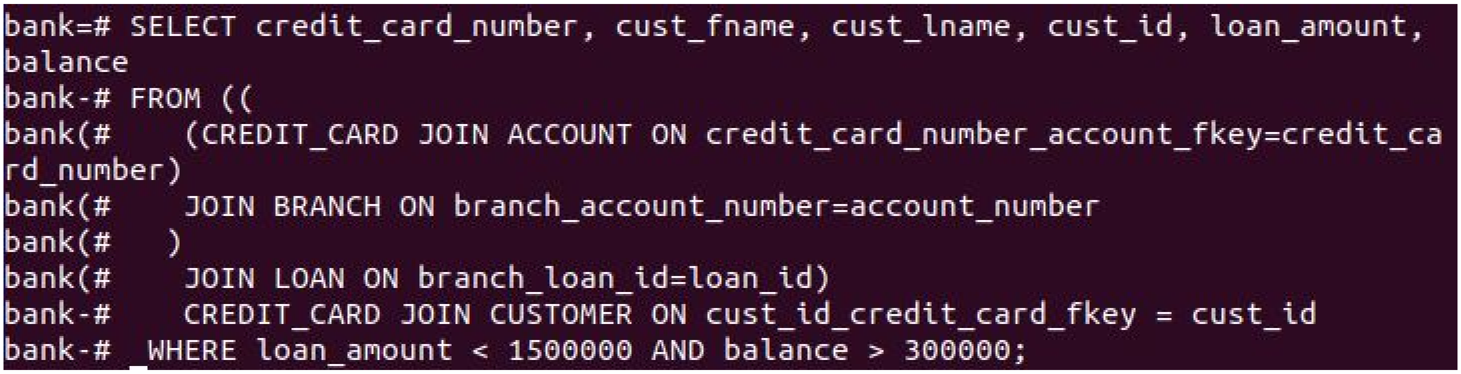
```
employee_id=employee_id_customer_fkey
     WHERE cust_fname LIKE'%k';
```



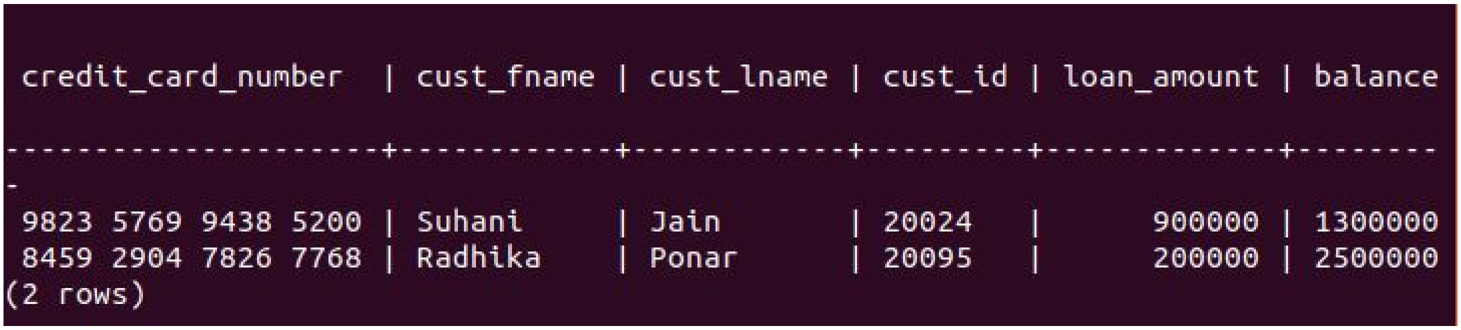
```
bank=# SELECT employee_fname,employee_lname, cust_fname, cust_lname, cust_id
 FROM EMPLOYEE JOIN CUSTOMER ON employee_id=employee_id_customer_fkey
 WHERE cust_fname LIKE'%k';
 employee_fname | employee_lname | cust_fname | cust_lname | cust_id
----------------+----------------+------------+------------+---------
 Raj            | Somai          | Ashok      | Mandal     | 10001
 Kumar          | Dinesh         | Karthik    | A          | 30018
(2 rows)
```

Find the credit card numbers of all the customers whose balance in the bank account is more than 3,00,000 and loan amounts lesser than 15,00,000

```
     SELECT credit_card_number, cust_fname, cust_lname, cust_id,
loan_amount, balance
     FROM (((CREDIT_CARD JOIN ACCOUNT ON account_number_cc_fkey =
account_number)
     JOIN BRANCH ON account_number_branch_fkey=account_number)
     JOIN LOAN ON loan_id_branch_fkey=loan_id )
     ACCOUNT JOIN CUSTOMER ON
account_number=account_number_customer_fkey
     WHERE loan_amount < 1500000 AND balance > 300000;
```



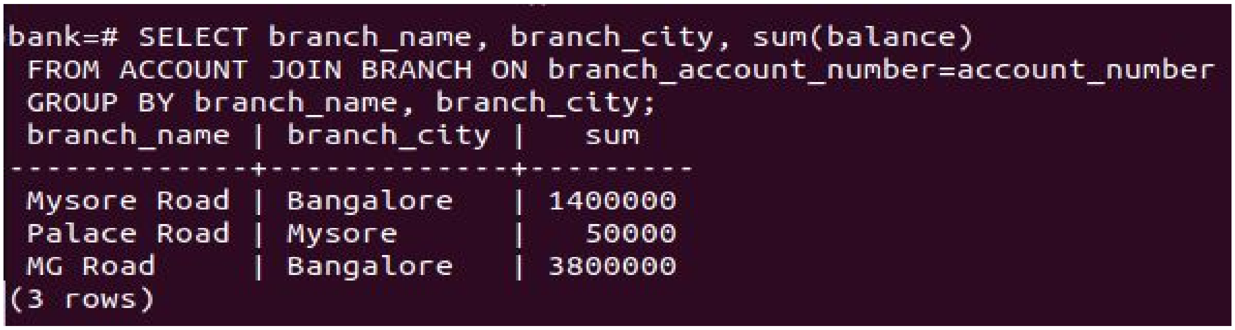
```
bank=# SELECT credit_card_number, cust_fname, cust_lname, cust_id, loan_amount,
balance
bank-# FROM ((
bank(#    (CREDIT_CARD JOIN ACCOUNT ON credit_card_number_account_fkey=credit_ca
rd_number)
bank(#    JOIN BRANCH ON branch_account_number=account_number
bank(#   )
bank(#    JOIN LOAN ON branch_loan_id=loan_id)
bank-#    CREDIT_CARD JOIN CUSTOMER ON cust_id_credit_card_fkey = cust_id
bank-#  WHERE loan_amount < 1500000 AND balance > 300000;
```



```
 credit_card_number  | cust_fname | cust_lname | cust_id | loan_amount | balance
---------------------+------------+------------+---------+-------------+--------
-
 9823 5769 9438 5200 | Suhani     | Jain       | 20024   |      900000 | 1300000
 8459 2904 7826 7768 | Radhika    | Ponar      | 20095   |      200000 | 2500000
(2 rows)
```

Find the total balance of each branch in the database

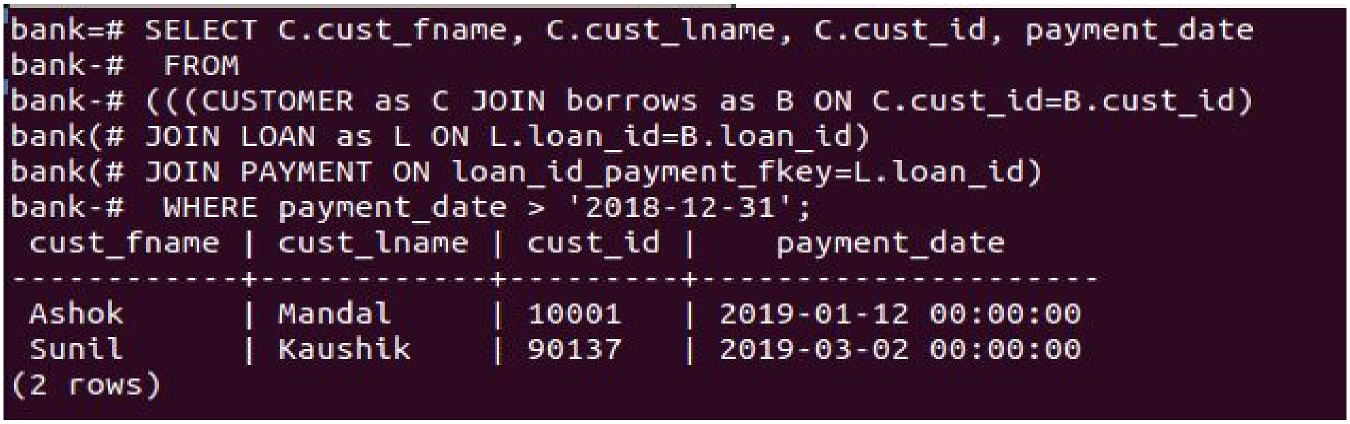
```
     SELECT branch_name, sum(balance)
     FROM ACCOUNT JOIN BRANCH ON
account_number_branch_fkey=account_number
     GROUP BY branch_name;
```



```
bank=# SELECT branch_name, branch_city, sum(balance)
 FROM ACCOUNT JOIN BRANCH ON branch_account_number=account_number
 GROUP BY branch_name, branch_city;
 branch_name | branch_city |   sum
-------------+-------------+---------
 Mysore Road | Bangalore   | 1400000
 Palace Road | Mysore      |   50000
 MG Road     | Bangalore   | 3800000
(3 rows)
```

Display all customers who have made their last payment after 2018

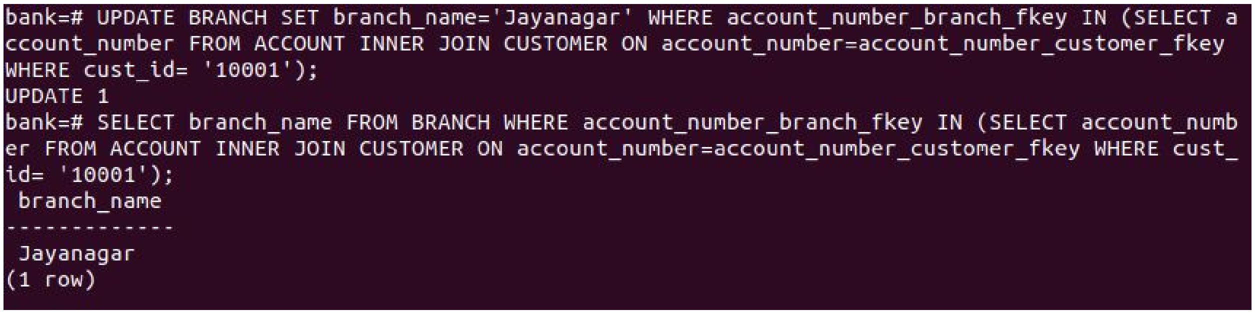
```
     SELECT C.cust_fname, C.cust_lname, C.cust_id, payment_date
     FROM
     (((CUSTOMER as C JOIN borrows as B ON C.cust_id=B.cust_id)
     JOIN LOAN as L ON L.loan_id=B.loan_id)
     JOIN PAYMENT ON loan_id_payment_fkey=L.loan_id)
     WHERE payment_date > '2018-12-31';
```



```
bank=# SELECT C.cust_fname, C.cust_lname, C.cust_id, payment_date
bank-#  FROM
bank-# (((CUSTOMER as C JOIN borrows as B ON C.cust_id=B.cust_id)
bank(# JOIN LOAN as L ON L.loan_id=B.loan_id)
bank(# JOIN PAYMENT ON loan_id_payment_fkey=L.loan_id)
bank-#  WHERE payment_date > '2018-12-31';
 cust_fname | cust_lname | cust_id |    payment_date
------------+------------+---------+---------------------
 Ashok      | Mandal     | 10001   | 2019-01-12 00:00:00
 Sunil      | Kaushik    | 90137   | 2019-03-02 00:00:00
(2 rows)
```

Update the branch of the customer, given his customer ID

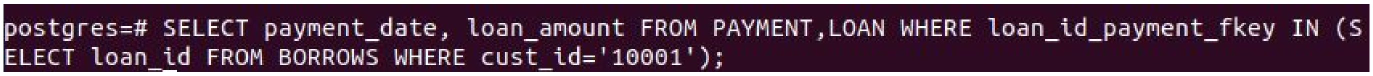
```
     UPDATE BRANCH SET branch_name='" . $getval . "' WHERE
account_number_branch_fkey IN (SELECT account_number FROM ACCOUNT
     INNER JOIN CUSTOMER ON
account_number=account_number_customer_fkey WHERE cust_id=
your_cust_id);
```



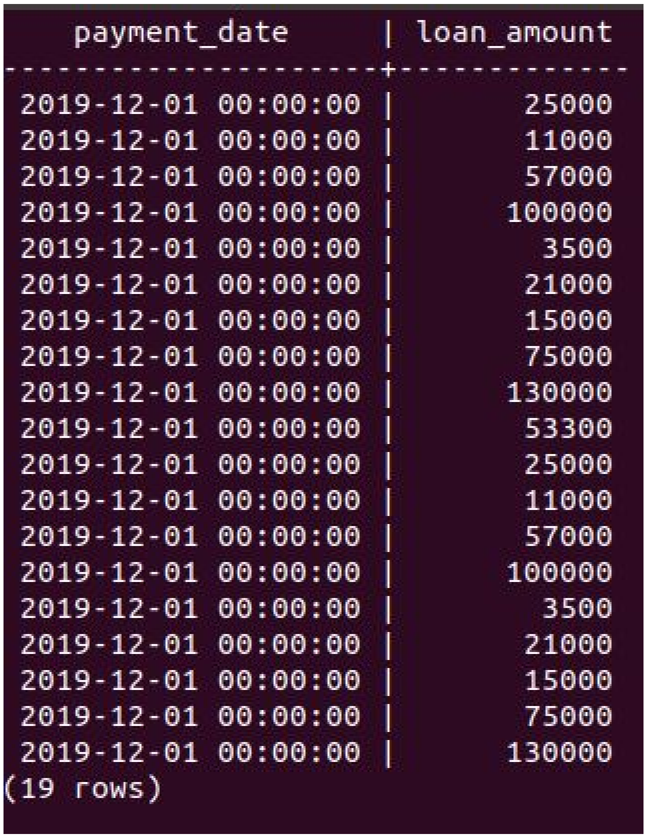
```
bank=# UPDATE BRANCH SET branch_name='Jayanagar' WHERE account_number_branch_fkey IN (SELECT a
ccount_number FROM ACCOUNT INNER JOIN CUSTOMER ON account_number=account_number_customer_fkey
WHERE cust_id= '10001');
UPDATE 1
bank=# SELECT branch_name FROM BRANCH WHERE account_number_branch_fkey IN (SELECT account_numb
er FROM ACCOUNT INNER JOIN CUSTOMER ON account_number=account_number_customer_fkey WHERE cust_
id= '10001');
 branch_name
-------------
 Jayanagar
(1 row)
```

Select payment date and loan amount given the customer ID

```
SELECT payment_date, loan_amount FROM PAYMENT WHERE
loan_id_payment_fkey IN (SELECT loan_id FROM BORROWS WHERE cust_id='"
. $custid . "');
```



```
postgres=# SELECT payment_date, loan_amount FROM PAYMENT,LOAN WHERE loan_id_payment_fkey IN (S
ELECT loan_id FROM BORROWS WHERE cust_id='10001');
```



```
    payment_date     | loan_amount
---------------------+-------------
 2019-12-01 00:00:00 |       25000
 2019-12-01 00:00:00 |       11000
 2019-12-01 00:00:00 |       57000
 2019-12-01 00:00:00 |      100000
 2019-12-01 00:00:00 |        3500
 2019-12-01 00:00:00 |       21000
 2019-12-01 00:00:00 |       15000
 2019-12-01 00:00:00 |       75000
 2019-12-01 00:00:00 |      130000
 2019-12-01 00:00:00 |       53300
 2019-12-01 00:00:00 |       25000
 2019-12-01 00:00:00 |       11000
 2019-12-01 00:00:00 |       57000
 2019-12-01 00:00:00 |      100000
 2019-12-01 00:00:00 |        3500
 2019-12-01 00:00:00 |       21000
 2019-12-01 00:00:00 |       15000
 2019-12-01 00:00:00 |       75000
 2019-12-01 00:00:00 |      130000
(19 rows)
```

CREATING THE TABLES

Employee Table:

```
CREATE TABLE EMPLOYEE
(employee_id varchar(10) PRIMARY KEY UNIQUE,
employee_fname varchar(100) NOT NULL,
employee_lname varchar(100) NOT NULL);
```

Accounts Table:

```
CREATE TABLE ACCOUNT
(account_number varchar(10) PRIMARY KEY UNIQUE,
balance float NOT NULL,
account_type varchar(50) NOT NULL);

Customer Table:
CREATE TABLE CUSTOMER
(cust_id varchar(10) PRIMARY KEY NOT NULL UNIQUE,
cust_fname varchar(100) NOT NULL,
cust_lname varchar(100) NOT NULL,
cust_city varchar(50) NOT NULL,
cust_street varchar(50),
employee_id_customer_fkey varchar(10),
account_number_customer_fkey varchar(10),
FOREIGN KEY(employee_id_customer_fkey)
REFERENCES EMPLOYEE(employee_id)
ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY(account_number_customer_fkey)
REFERENCES ACCOUNT(account_number)
ON UPDATE CASCADE ON DELETE CASCADE);
```

Credit Card Table:

```
CREATE TABLE CREDIT_CARD
(credit_card_number varchar(40) PRIMARY KEY UNIQUE NOT NULL,
expiry TIMESTAMP NOT NULL,
cc_limit float,
account_number_cc_fkey varchar(10),
FOREIGN KEY(account_number_cc_fkey)
REFERENCES ACCOUNT(account_number)
ON UPDATE CASCADE ON DELETE CASCADE);
```

Loan Table:

```
CREATE TABLE LOAN
(loan_id varchar(10) NOT NULL,
loan_amount float NOT NULL,
UNIQUE(loan_id),
PRIMARY KEY(loan_id));
```

Branch Table:

```
CREATE TABLE BRANCH(
branch_city varchar(50) NOT NULL,
branch_name varchar(20) NOT NULL,
account_number_branch_fkey varchar(10) NOT NULL,
loan_id_branch_fkey varchar(10) NOT NULL,
FOREIGN KEY(loan_id_branch_fkey) REFERENCES LOAN(loan_id)
ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY(account_number_branch_fkey) REFERENCES
ACCOUNT(account_number)
ON UPDATE CASCADE ON DELETE CASCADE);
```

Payment Table:

```
CREATE TABLE PAYMENT
(payment_id varchar(10) PRIMARY KEY UNIQUE NOT NULL,
payment_amount float,
payment_date TIMESTAMP NOT NULL,
loan_id_payment_fkey varchar(10) NOT NULL,
FOREIGN KEY(loan_id_payment_fkey) REFERENCES LOAN(loan_id)
ON UPDATE CASCADE ON DELETE CASCADE);
```

Borrows Relation:

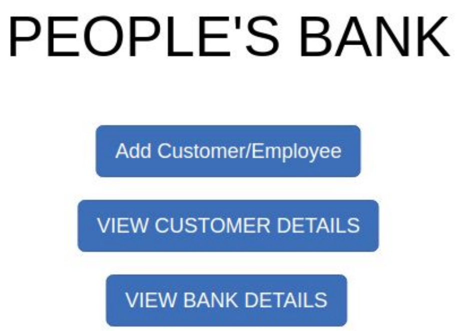
```
CREATE TABLE borrows
(cust_id varchar(10) REFERENCES CUSTOMER(cust_id)
ON UPDATE CASCADE ON DELETE CASCADE,
loan_id varchar(10) REFERENCES LOAN(loan_id)
ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT borrows_pkey PRIMARY KEY (cust_id, loan_id));
```

Multiple phone numbers of each customer:

```
CREATE TABLE multiphone
(Phone_cust_id varchar(10) REFERENCES CUSTOMER(cust_id)
ON UPDATE CASCADE ON DELETE CASCADE,
phone_no numeric(10));
```

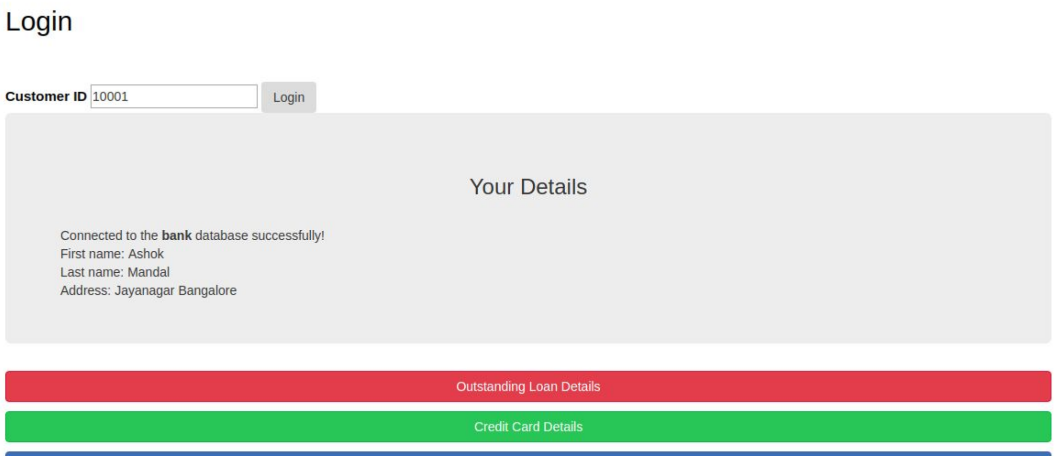
WIREFRAME DIAGRAMS

Home page:



Customer Relevant Details - View Only:

1. Login



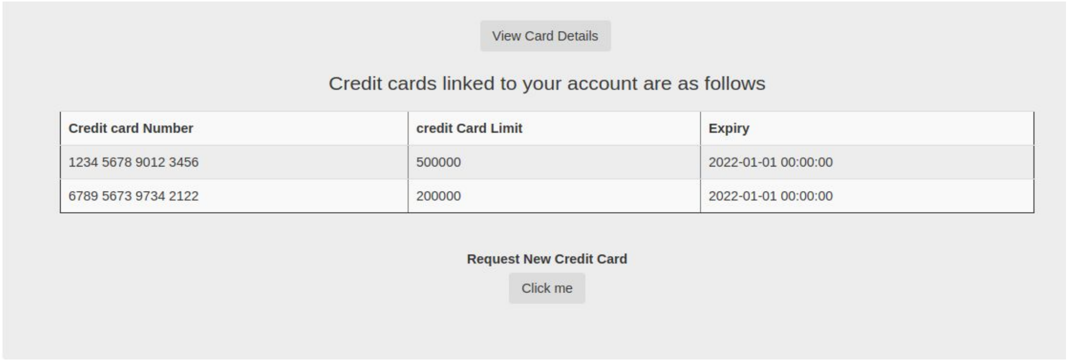
Login

Customer ID 10001

Your Details

Connected to the **bank** database successfully!
First name: Ashok
Last name: Mandal
Address: Jayanagar Bangalore

2. Credit card Details

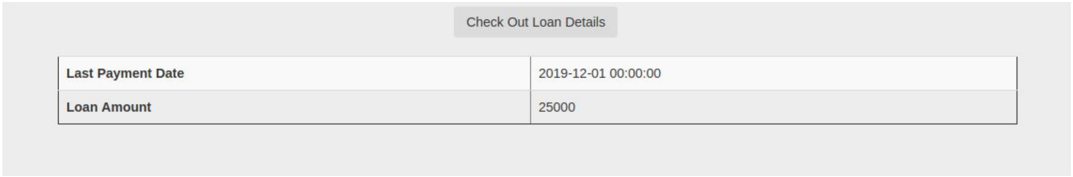


Credit cards linked to your account are as follows

Credit card Number	credit Card Limit	Expiry
1234 5678 9012 3456	500000	2022-01-01 00:00:00
6789 5673 9734 2122	200000	2022-01-01 00:00:00

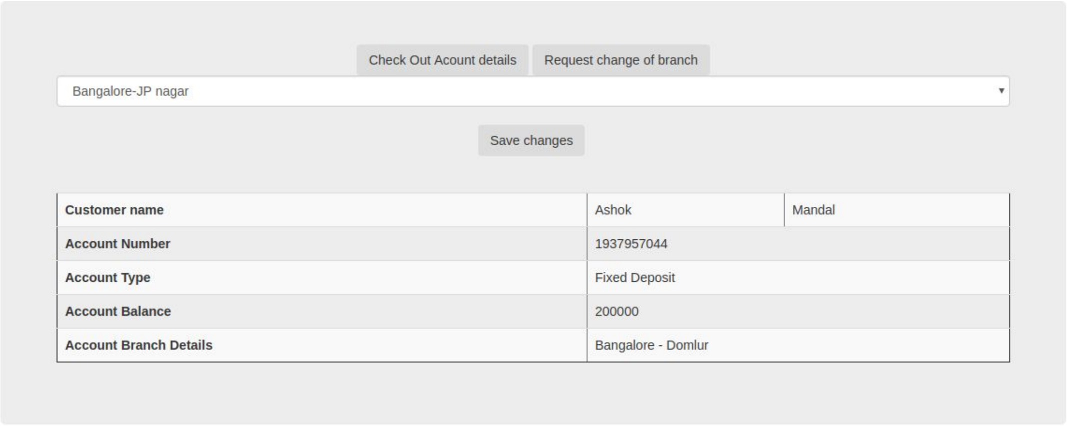
Request New Credit Card

3. Loan Details



Last Payment Date	2019-12-01 00:00:00
Loan Amount	25000

4. Account Details

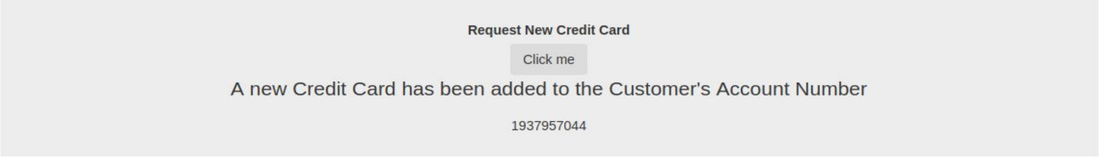


Bangalore-JP nagar

Customer name	Ashok	Mandal
Account Number	1937957044	
Account Type	Fixed Deposit	
Account Balance	200000	
Account Branch Details	Bangalore - Domlur	

Customer Relevant Details - Updation:

5. Requesting a new credit card

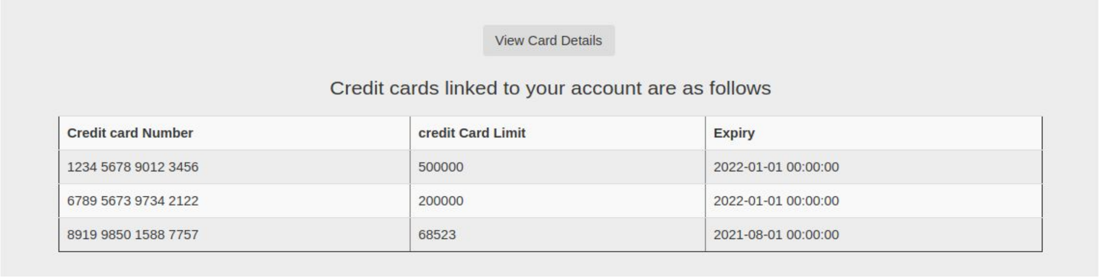


Request New Credit Card

A new Credit Card has been added to the Customer's Account Number

1937957044

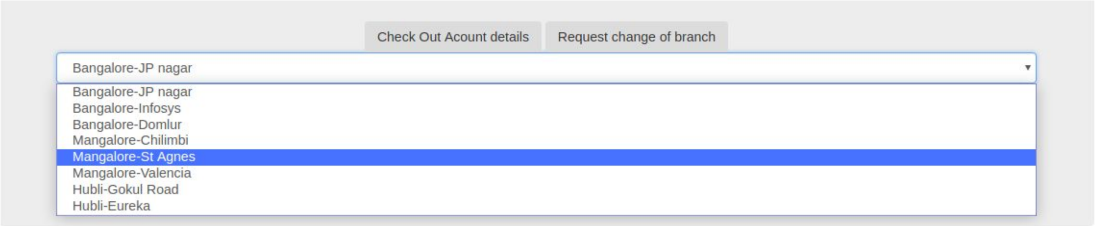
Note the insertion of a new credit card, by comparing with Fig (2)

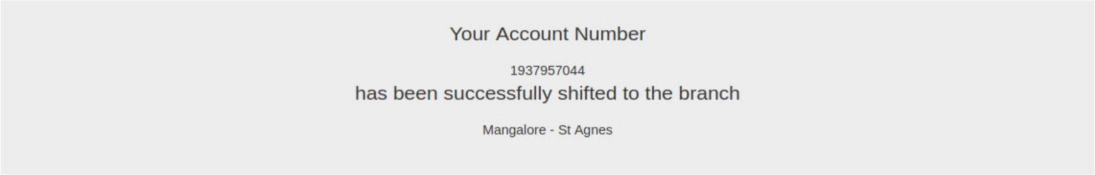


Credit cards linked to your account are as follows

Credit card Number	credit Card Limit	Expiry
1234 5678 9012 3456	500000	2022-01-01 00:00:00
6789 5673 9734 2122	200000	2022-01-01 00:00:00
8919 9850 1588 7757	68523	2021-08-01 00:00:00

6. Requesting change of branch





Your Account Number

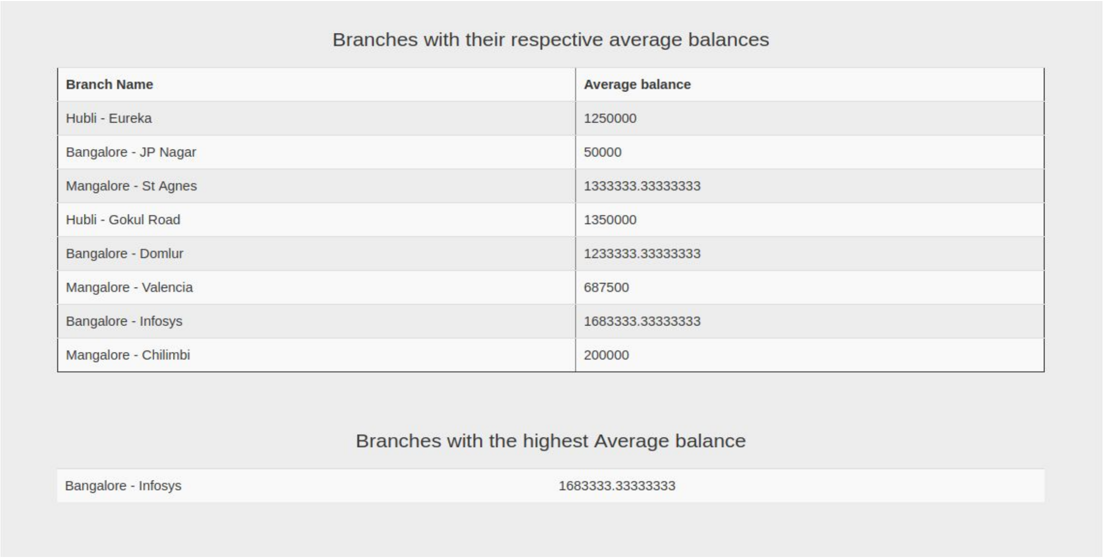
1937957044

has been successfully shifted to the branch

Mangalore - St Agnes

Bank Relevant Details

1. Branch Relevant

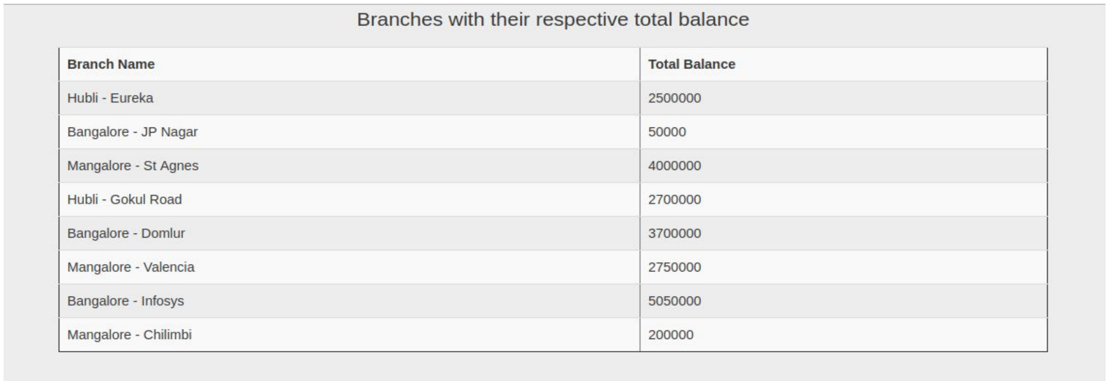


Branches with their respective average balances

Branch Name	Average balance
Hubli - Eureka	1250000
Bangalore - JP Nagar	50000
Mangalore - St Agnes	1333333.33333333
Hubli - Gokul Road	1350000
Bangalore - Domlur	1233333.33333333
Mangalore - Valencia	687500
Bangalore - Infosys	1683333.33333333
Mangalore - Chilimbi	200000

Branches with the highest Average balance

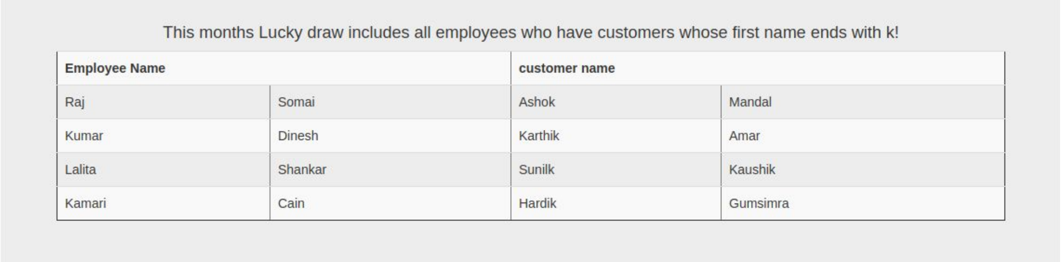
Bangalore - Infosys	1683333.33333333



Branches with their respective total balance

Branch Name	Total Balance
Hubli - Eureka	2500000
Bangalore - JP Nagar	50000
Mangalore - St Agnes	4000000
Hubli - Gokul Road	2700000
Bangalore - Domlur	3700000
Mangalore - Valencia	2750000
Bangalore - Infosys	5050000
Mangalore - Chilimbi	200000

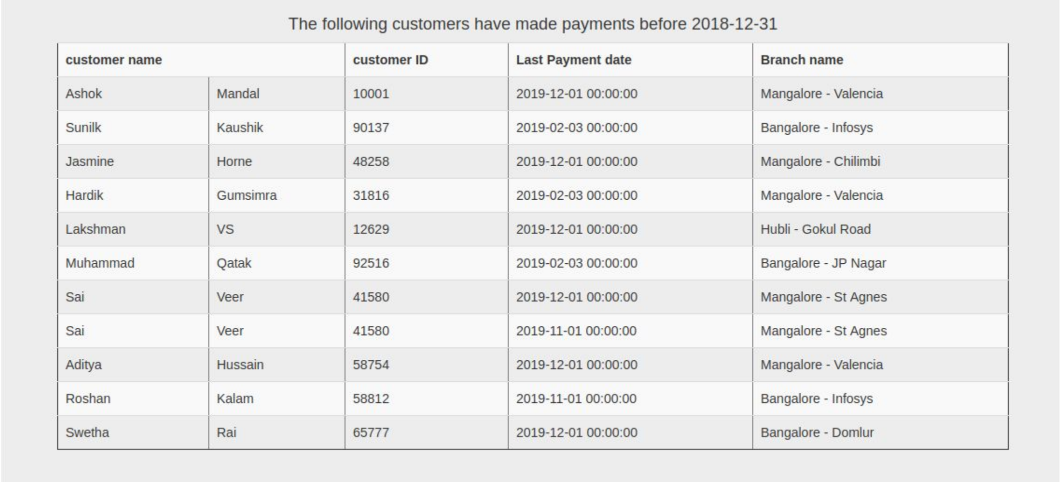
2. Lucky draw for employees



This months Lucky draw includes all employees who have customers whose first name ends with k!

Employee Name		customer name	
Raj	Somai	Ashok	Mandal
Kumar	Dinesh	Karthik	Amar
Lalita	Shankar	Sunilk	Kaushik
Kamari	Cain	Hardik	Gumsimra

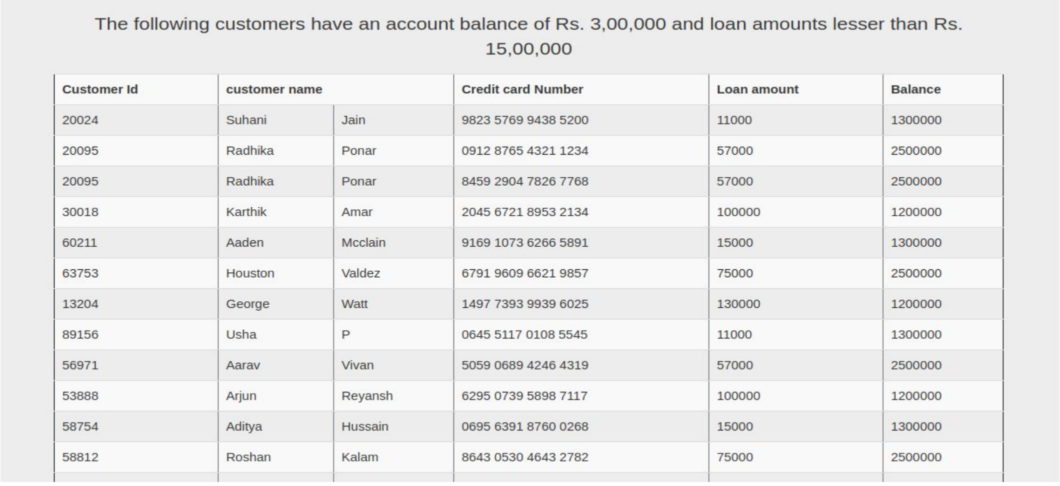
3. Checkout most recent payments made by customers



The following customers have made payments before 2018-12-31

customer name		customer ID	Last Payment date	Branch name
Ashok	Mandal	10001	2019-12-01 00:00:00	Mangalore - Valencia
Sunilk	Kaushik	90137	2019-02-03 00:00:00	Bangalore - Infosys
Jasmine	Horne	48258	2019-12-01 00:00:00	Mangalore - Chilimbi
Hardik	Gumsimra	31816	2019-02-03 00:00:00	Mangalore - Valencia
Lakshman	VS	12629	2019-12-01 00:00:00	Hubli - Gokul Road
Muhammad	Qatak	92516	2019-02-03 00:00:00	Bangalore - JP Nagar
Sai	Veer	41580	2019-12-01 00:00:00	Mangalore - St Agnes
Sai	Veer	41580	2019-11-01 00:00:00	Mangalore - St Agnes
Aditya	Hussain	58754	2019-12-01 00:00:00	Mangalore - Valencia
Roshan	Kalam	58812	2019-11-01 00:00:00	Bangalore - Infosys
Swetha	Rai	65777	2019-12-01 00:00:00	Bangalore - Domlur

4. Elite Customers

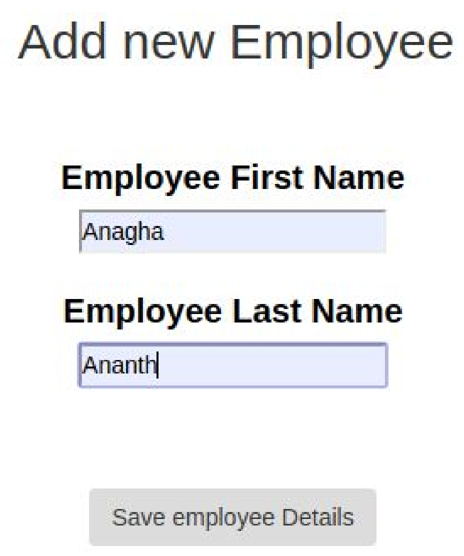


The following customers have an account balance of Rs. 3,00,000 and loan amounts lesser than Rs. 15,00,000

Customer Id	customer name		Credit card Number	Loan amount	Balance
20024	Suhani	Jain	9823 5769 9438 5200	11000	1300000
20095	Radhika	Ponar	0912 8765 4321 1234	57000	2500000
20095	Radhika	Ponar	8459 2904 7826 7768	57000	2500000
30018	Karthik	Amar	2045 6721 8953 2134	100000	1200000
60211	Aaden	Mcclain	9169 1073 6266 5891	15000	1300000
63753	Houston	Valdez	6791 9609 6621 9857	75000	2500000
13204	George	Watt	1497 7393 9939 6025	130000	1200000
89156	Usha	P	0645 5117 0108 5545	11000	1300000
56971	Aarav	Vivan	5059 0689 4246 4319	57000	2500000
53888	Arjun	Reyansh	6295 0739 5898 7117	100000	1200000
58754	Aditya	Hussain	0695 6391 8760 0268	15000	1300000
58812	Roshan	Kalam	8643 0530 4643 2782	75000	2500000

Insertion of Customer and Employee

1. Add new Employee:



Add new Employee

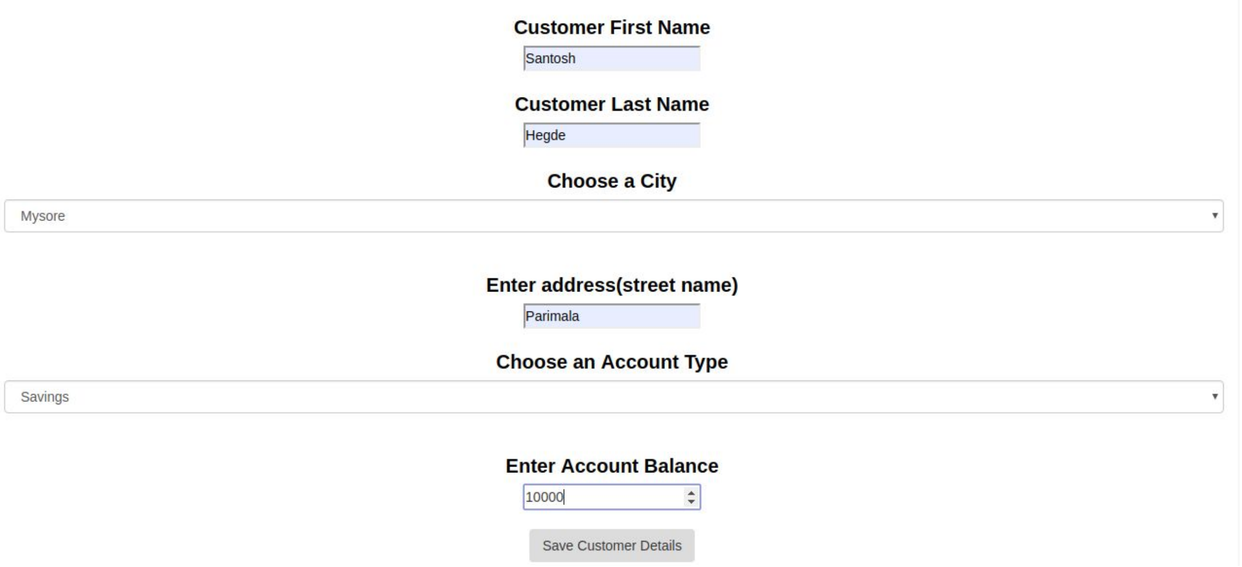
Employee First Name

Anagha

Employee Last Name

Ananth

2. Add new Customer:



Customer First Name

Santosh

Customer Last Name

Hegde

Choose a City

Mysore

Enter address(street name)

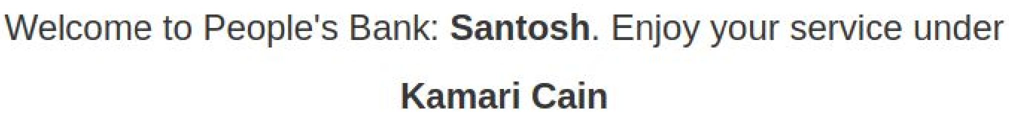
Parimala

Choose an Account Type

Savings

Enter Account Balance

10000



Welcome to People's Bank: **Santosh**. Enjoy your service under

Kamari Cain

ERROR HANDLED/TEST CASES

1. A customer can have a maximum of only 3 credit cards
2. A phone number/credit card cannot be deleted, if it is the only one linked to a customer
3. Before adding a customer to the database an account is created for him, and an employee is assigned to him
4. While inserting tuples in the database, no two customers can have the same credit card number
5. All credit card numbers are generated randomly and uniquely
6. The credit card limit of a customer is always maintained at four times his bank balance amount.

DISCUSSION & CONCLUSION

In summation, the Banking Database Management System seems to solve all problems it aimed to resolve, essentially making it easier for our user, the bank to obtain relevant information regarding its customers on demand such as account details, updating their branch and accessing the last payment the customer made towards the loan, as well as being constantly updated on the performance of different branches affiliated with the bank and customers who are worthy of being sent invites to invest in the bank's mutual fund. Additionally, it included features wherein customers who are selected as part of a lucky draw are presented to the bank to enable the bank to send special offers to them. Customers can add credit cards to their associated accounts to enable dependents to access their bank account with a separate credit card. On a whole, the Banking Database Management system has taken an all round approach to solving the bank's issue of maintaining records on customers by providing an easy way to access all the important information they require while still having the potential to expand into a larger domain by helping with the management of other important resources both within the bank and beyond it.

REFERENCES

1. Fundamentals of Database Systems, 7th Edition - Elmasri & Navathe
2. PostgreSQL Tutorial: http://www.postgresqltutorial.com/postgresql-data-types/
3. Official PHP Documentation: https://www.php.net/docs.php

DATABASES, TOOLS AND TECHNOLOGIES USED

1. Database Management Software: **Postgresql**
2. Frontend Development: **HTML, CSS, Bootstrap**
3. Backend Development: **PHP**
4. Web server deployed through **XAMPP**