राष्ट्रीय प्रौद्योगिकी संस्थान दिल्ली
National Institute of Technology Delhi
(An autonomous Institute under the aegis of Ministry of HRD, Govt. of India)

Database Management System Project

(BANK MANAGEMENT SYSTEM)

By

Ahan Bandyopadhyay (211210008)

Gaurav Singh (211210024)

Kanha Madan (211210031)

CONTENTS

INTRODUCTION

- The bank management system is a set of essential tools and processes that allow banks and their credit institutions to carry out their functions.

- The components of the bank management system may differ depending on the bank, but generally, the system includes core banking to manage basic transactions, loans, mortgages, and payments accessible via ATM, mobile banking, and branches.

- Database Management is the core of modern data, as handling and managing data is the key to making exponential progress in today's world.

Here, the undersigned students of CSE 2nd year:

1. AHAN BANDYOPADHYAY (211210008)
2. GAURAV SINGH (211210024)
3. KANHA MADAN (211210031)

Have completed a project on the Database of the Bank Management System which cites all the information.

1.1 Project objective

. To allow only authorized user to access various function and processed available in the system.
. Locate any A/C wanted by the user.
. Reduced clerical work as most of the work done by computer.
. Provide greater speed & reduced time consumption.

1.2 Project benefits

- To save time and make better accounting system.
- To increase efficiency of employees.
- For faster access of data and information.
- For smooth and fair running of the organization.
- To manipulate the banking transaction with instant confirmation for the withdrawal, deposit, loan etc.

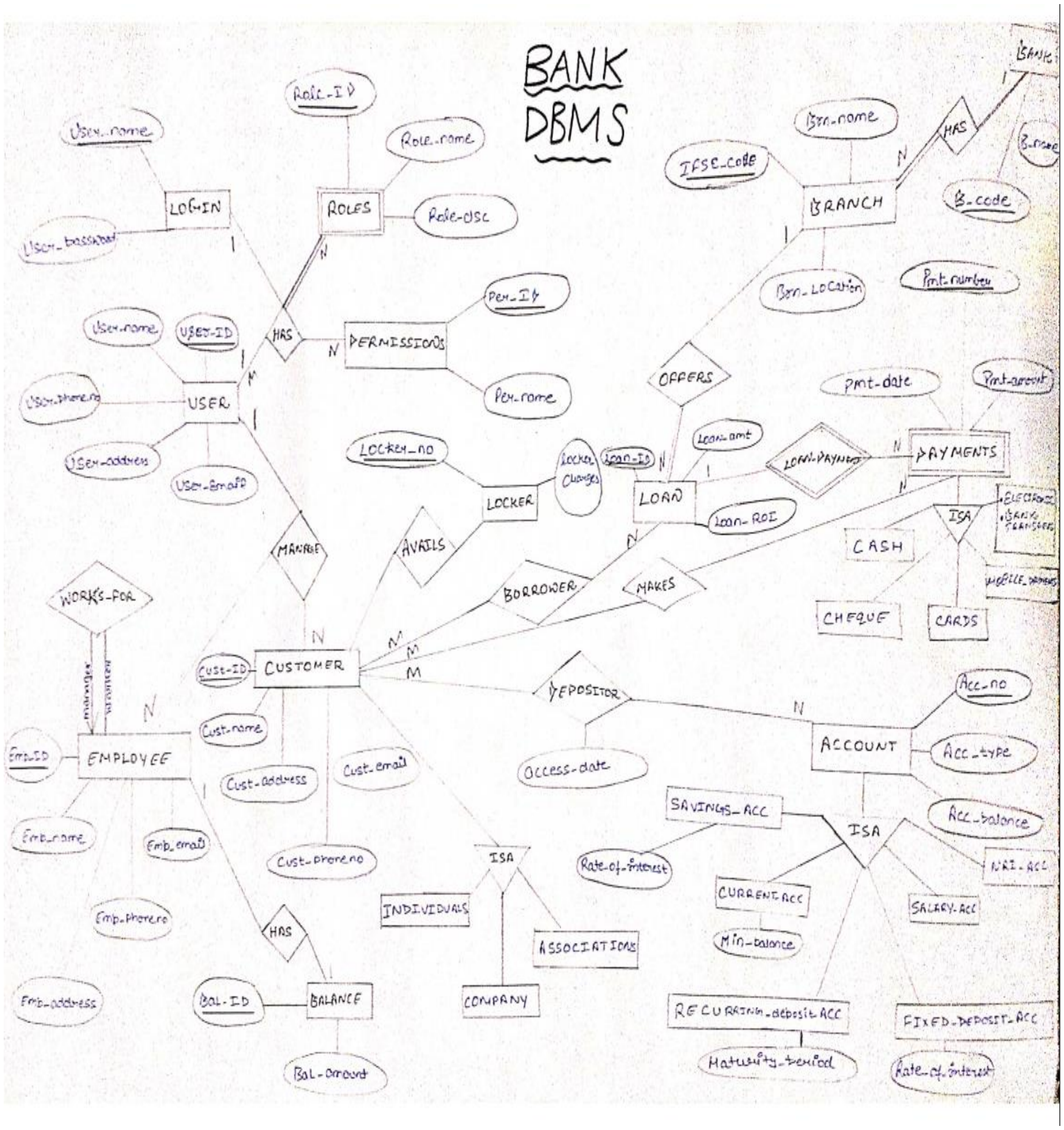
1.3 Project scope

Banking activities are considered to be the life blood of national economy. Without banking services, trading and business activities cannot be carried on smoothly. Banks are the distributors and protectors of liquid capital which is of vital significance to a developing country.
Efficient administration of the banking system helps in the economic Growth of the nation. Banking is useful to trade and commerce.

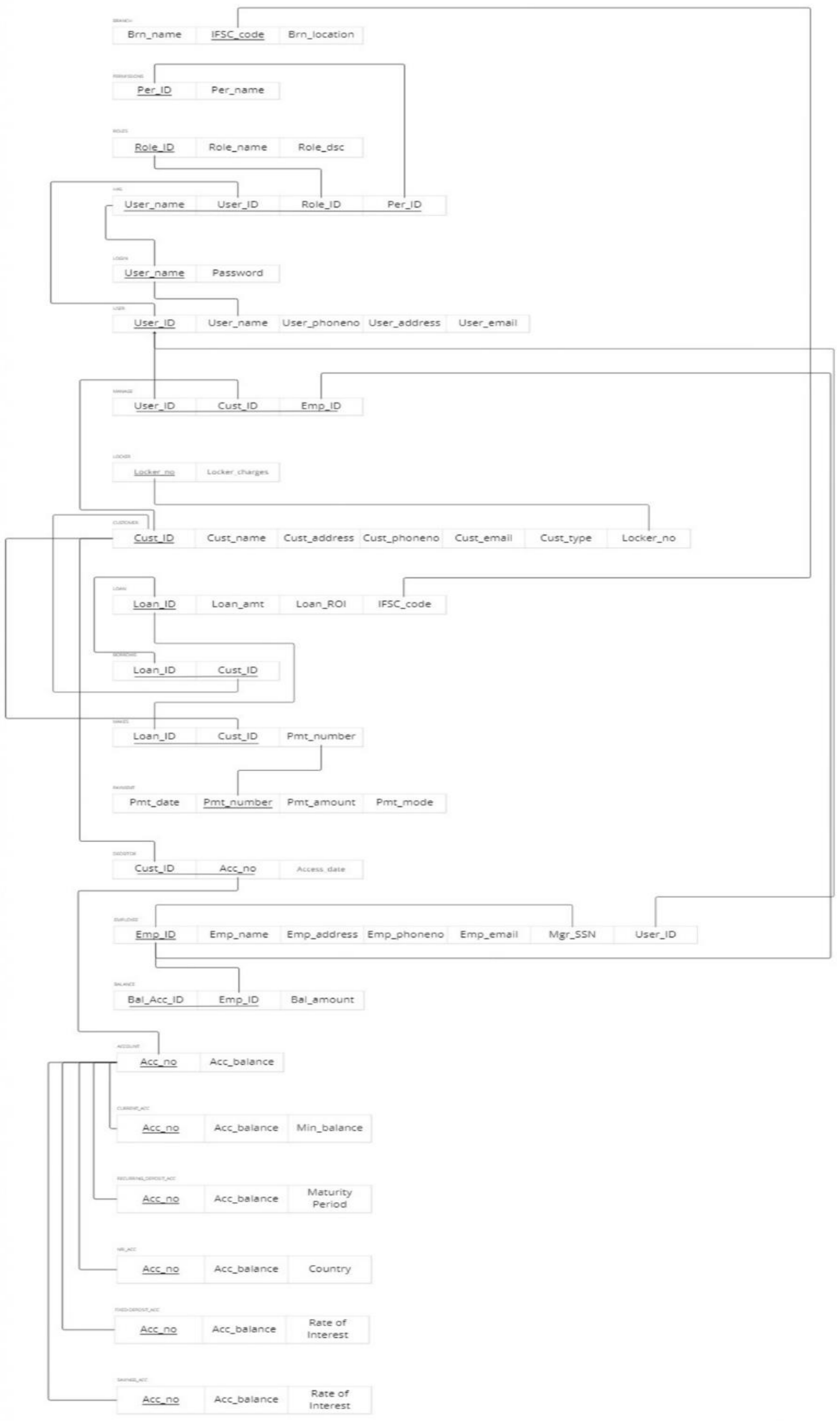
Software requirements:

In the development of a project the selection of an appropriate DBMS Software and a platform is of primary importance. With many software options available a developer has to consider the various features and functionalities and ease of handling the software, keeping an account of such things we decided to use Bootstrap Studio for designing the front-end, the front-end has been developed by the use of HTML, CSS and JS. MySQL has been used as a back-end query language. PHP has been chosen as a scripting language. The server chosen is the localhost which would be hosting the website on the machine itself.

ER DIAGRAM


BANK DBMS
Role-ID
User_name
Role-name
LOGIN
ROLES
Role-dsc
User_password
HAS
PERMISSIONS
Per_ID
Per-name
User-name
USER-ID
USER
User-phone.no
User-address
User-Email
Locker_no
LOCKER
Locker Charges
Loan-ID
LOAN
Loan-amt
Loan-ROI
LOAN-PAYMENT
PAYMENTS
pmt-date
Pmt-amount
Pmt-number
ISA
CASH
CHEQUE
CARDS
ELECTRONIC
BANK TRANSFER
MOBILE_PAYMENTS
BANK
HAS
B_name
B_code
BRANCH
Brn-name
IFSC_CODE
Brn_Location
OFFERS
MANAGE
AVAILS
BORROWER
MAKES
WORKS-FOR
manager
worker
EMPLOYEE
Emp-ID
Emp_name
Emp_email
Emp_phone.no
Emp-address
CUSTOMER
Cust-ID
Cust-name
Cust-address
Cust-email
Cust_phone.no
HAS
BALANCE
Bal-ID
Bal-amount
ISA
INDIVIDUALS
ASSOCIATIONS
COMPANY
DEPOSITOR
access-date
ACCOUNT
Acc-no
Acc_type
Acc_balance
ISA
SAVINGS_ACC
Rate-of-interest
CURRENT ACC
Min-balance
NRI-ACC
SALARY-ACC
RECURRING_deposit ACC
Maturity_period
FIXED-DEPOSIT-ACC
Rate-of-interest
1
N
M

MAPPING ER->RDBMS MODEL


Brn_name | IFSC_code | Brn_location
Per_ID | Per_name
Role_ID | Role_name | Role_dsc
User_name | User_ID | Role_ID | Per_ID
User_name | Password
User_ID | User_name | User_phoneno | User_address | User_email
User_ID | Cust_ID | Emp_ID
Locker_no | Locker_charges
Cust_ID | Cust_name | Cust_address | Cust_phoneno | Cust_email | Cust_type | Locker_no
Loan_ID | Loan_amt | Loan_ROI | IFSC_code
Loan_ID | Cust_ID
Loan_ID | Cust_ID | Pmt_number
Pmt_date | Pmt_number | Pmt_amount | Pmt_mode
Cust_ID | Acc_no | Access_date
Emp_ID | Emp_name | Emp_address | Emp_phoneno | Emp_email | Mgr_SSN | User_ID
Bal_Acc_ID | Emp_ID | Bal_amount
Acc_no | Acc_balance
Acc_no | Acc_balance | Min_balance
Acc_no | Acc_balance | Maturity Period
Acc_no | Acc_balance | Country
Acc_no | Acc_balance | Rate of Interest
Acc_no | Acc_balance | Rate of Interest

CREATION Table queries:

```
create database bank;


use bank;
create table roles
(
  Role_ID char(10) not null,
  Role_name varchar(30),
  Role_dsc varchar(30)
  );

use bank;
create table permissions
(
          Per_ID char(10) not null,
          Per_name varchar(30)
);


use bank;
create table locker
(
 Locker_no char(10) not null,
 Locker_charges int
 );

use bank;
```

```
create table locker
(
 Locker_no char(10) not null,
 Locker_charges int
 );

use bank;
create table makes
(
 Loan_Id char(10) not null,
  Cust_ID char(10) not null,
  Pmt_number char(10) not null
  );

use bank;
create table borrows
(
  Loan_Id char(10) not null,
  Cust_ID char(10) not null
  );

use bank;
create table branch
(
  Brn_name varchar(20),
  IFSC_code char(10) not null,
  Brn_location varchar(50)
);

use bank;
create table payment
(
 Pmt_date date,
 Pmt_number char(10)  not null,
 Pmt_amount int,
 Pmt_mode varchar(30)
 );

use bank;
create table depositor
(
 Cust_ID char(10) not null,
 Acc_no char(10) not null,
 Access_date date
 );

use bank;
create table savings_acc
(
 Acc_no char(10) not null,
 Acc_balance bigint,
 Rate_of_interest int
 );

use bank;
create table current_acc
(
 Acc_no char(10) not null,
 Acc_balance bigint,
 Min_balance int
 );

use bank;
create table recurring_deposit_acc
(
 Acc_no char(10) not null,
 Acc_balance bigint,
 Maturity_period int
 );

use bank;
create table fixed_deposit_acc
(
 Acc_no char(10) not null,
 Acc_balance bigint,
 Rate_of_interest int
 );
```

```
use bank;
create table NRI_acc
(
 Acc_no char(10) not null,
 Acc_balance bigint,
 country varchar(30)
 );

use bank;
create table balance
(
  Bal_Acc_ID char(10) not null,
  Emp_ID char(10) not null,
  Bal_amount bigint
);

use bank ;
create table manage
(
  User_ID char(10) not null,
  Cust_ID char(10) not null,
  Emp_ID char(10) not null
);

use bank;
create table has
```

```
(
	User_name char(30) not null unique,
	User_ID char(10) not null,
	Role_ID  char(10) not null,
	Per_ID  char(10) not null
);

use bank;
create table employee
(
  Emp_ID char(10) not null,
  Emp_name char(30),
  Emp_address varchar(50),
  Emp_phoneno bigint,
  Emp_email varchar(30),
  Mgr_SSN char(10) not null
);

use bank;

insert into login
values
('abc001', '123abc'),
('def002', '456def'),
('ghi003', '789ghi'),
('abcd_1', '123abcd'),
('efgh_2', '456efgh'),
('ijkl_3', '789ijkl');

use bank;
create table user
(
 User_ID char(10) not null,
 User_name char(30) not null unique,
 User_phoneno bigint,
 User_address varchar(50),
 User_email varchar(30)
);

use bank;
create table account
(
 Acc_no char(10) not null,
 Acc_balance bigint
 );

use bank;
create table customer
(
 Cust_ID char(10) not null,
 Cust_name varchar(30),
 Cust_address varchar(50),
 Cust_phoneno bigint,
 Cust_email varchar(30),
 Cust_type varchar(30),
 Locker_no char(10) not null
);

use bank;
create table login
(
  User_name char(30),
  Password varchar(30)
);
```

INSERTION queries:

```
use bank;

insert into branch

values

('TILAKNAGAR BRANCH','gsr0210395','west delhi'),

('WRIGHTGANJ BRANCH','gsr0210456','south delhi'),

('ALIPUR BRANCH','gsr0210678','north delhi');


use bank;

insert into loan

values

(0201,3000000,0.1,'gsr0210395'),

(1234,4000000,0.095,'gsr0210456'),

(2345,100000,0.085,'gsr0210678');


use bank;

insert into borrows

values

(0201,'abc'),

(1234,'def'),

(2345,'ghi');


use bank;

insert into current_acc

values

(5037720309,25000,1000),
(5147756908,200024,500),
(2314567890,1975308,1);

use bank;
insert into recurring_deposit_acc
values
(5037720309,25000,6),
(5147756908,200024,8),
(2314567890,1975308,10);

use bank;
insert into fixed_deposit_acc
values
(5037720309,25000,2.75),
(5147756908,200024,3.0),
(2314567890,1975308,2.14);

use bank;
insert into NRI_acc
values
(5037720309,25000,'norway'),
(5147756908,200024,'swizerland'),
(2314567890,1975308,'london');

use bank;
insert into makes
```

```
values
(0201,'abc',000256),
(1234,'def',000255),
(2345,'ghi',000254);

use bank;
insert into payment
values
('2018-09-20', 000256,4000,'check'),
('2019-03-15', 000255,20000,'cash'),
('2023-12-25',000254,100000,'netbanking');

use bank;
insert into account
values
(5037720309,100000),
(5147756908,1000123),
(2314567890,9876543);

use bank;
insert into depositor
values
('abc',5037720309,'2023-03-12'),
('def',5147756908,'2024-04-13'),
('ghi',2314567890,'2025-05-14');
```

```
use bank;

insert into balance
values
('9999', 'abcd', 200000),
('8888', 'efgh', 1000000),
('7777', 'ijkl', 5000000);

use bank;
insert into savings_acc
values
(5037720309,25000,0.001),
(5147756908,200024,0.002),
(2314567890,1975308,0.0003);

use bank;
insert into locker
values
(32,3000),
(315,5000),
(123,2000);

use bank;

insert into user
values
```

```
('abc001', 'ahan001', 1234567890, 'narela', 'abc@gmail.com'),
('def002', 'gaurav002', 7291994633, 'south delhi', 'def@gmail.com'),
('ghi003', 'kanha003', 9899140714, 'rohtak', 'ghi@gmail.com'),
('abcd_1', 'varuag01', 9918463280, 'kalkaji', 'abcd@gmail.com'),
('efgh_2', 'aanchal02', 9567438812, 'noida', 'efgh@gmail.com'),
('ijkl_3', 'gsr03', 7865432517, 'chitranjanpark', 'ijkl@gmail.com');

use bank;
insert into customer
values
('abc', 'ahan', 'narela', 1234567890, 'abc@gmail.com', 'individual', 32),
('def', 'gaurav', 'south delhi', 7291994633, 'def@gmail.com', 'individual', 315),
('ghi', 'kanha', 'rohtak', 9899140714, 'ghi@gmail.com', 'individual', 123);

use bank;
insert into employee
values
('abcd','varuag','kalkaji',9918463280,'abcd@gmail.com', 'ijkl'),
('efgh','aanchal','noida',9567438812,'efgh@gmail.com', 'ijkl'),
('ijkl','gsr','chitranjanpark',7865432517,'ijkl@gmail.com', 'ijkl');
```

Primary keys declaration:

```
use bank;
alter table account add primary key(Acc_no);
alter table balance add primary key(Bal_Acc_ID, Emp_ID);
alter table borrows add primary key(Loan_ID, Cust_ID);
alter table branch add primary key(IFSC_code);
alter table current_acc add primary key(Acc_no);
alter table customer add primary key(Cust_ID);
alter table depositor add primary key(Cust_ID, Acc_no);
alter table employee add primary key(Emp_ID);
alter table fixed_deposit_acc add primary key(Acc_no);
alter table has add primary key(User_name, User_ID, Role_ID, Per_ID);
alter table loan add primary key(Loan_ID);
alter table locker add primary key(Locker_no);
alter table login add primary key(User_name);
alter table makes add primary key(Loan_ID, Cust_ID);
alter table manage add primary key(User_ID, Cust_ID, Emp_ID);
alter table nri_acc add primary key(Acc_no);
alter table payment add primary key(Pmt_number);
alter table permissions add primary key(Per_ID);
alter table recurring_deposit_acc add primary key(Acc_no);
alter table roles add primary key(Role_ID);
alter table savings_acc add primary key(Acc_no);
alter table user add primary key(User_ID);
```

Foreign Keys Declarations:

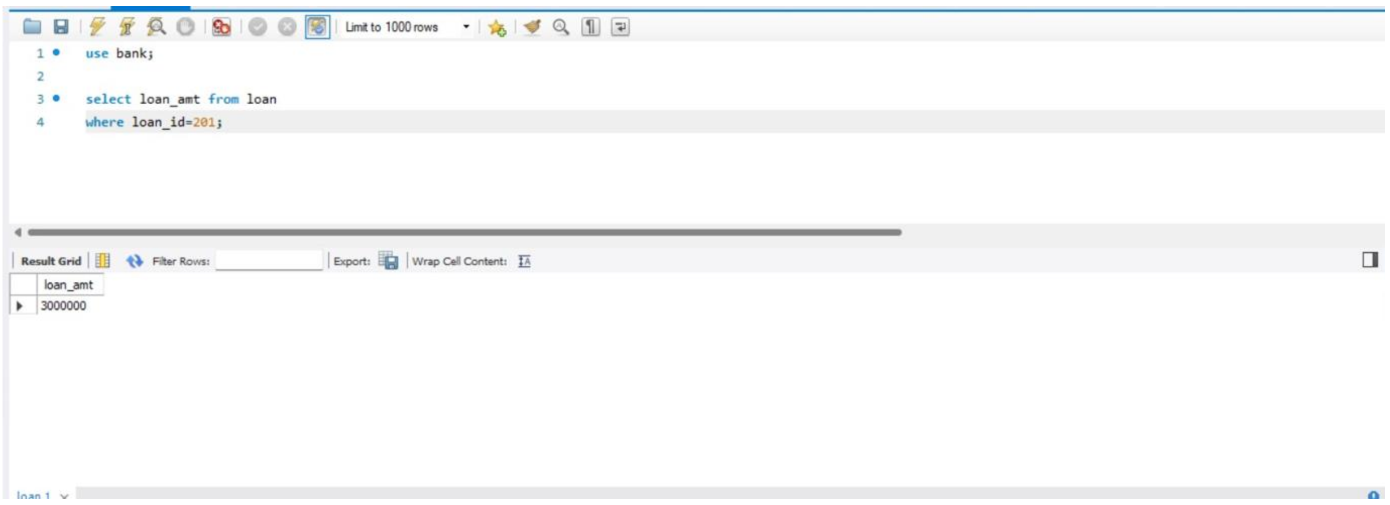
```
use bank;

-- alter table customer add foreign key(Locker_no) references
locker(Locker_no);
-- alter table manage add foreign key (User_ID) references user(User_ID);
-- alter table manage add foreign key (Cust_ID) references customer(Cust_ID);
-- alter table manage add foreign key (Emp_ID) references employee(Emp_ID);
-- alter table loan add foreign key (IFSC_code) references branch(IFSC_code);
-- alter table borrows add foreign key (Loan_ID) references loan(Loan_ID);
-- alter table borrows add foreign key (Cust_ID) references customer(Cust_ID);
-- alter table makes add foreign key (Loan_ID) references loan(Loan_ID);
-- alter table user add foreign key(User_name) references login(User_name);
-- alter table makes add foreign key (Cust_ID) references customer(Cust_ID);
-- alter table makes add foreign key (Pmt_number) references
payment(Pmt_number);
-- alter table depositor add foreign key (Cust_ID) references
customer(Cust_ID);
-- alter table depositor add foreign key (Acc_no) references account(Acc_no);
-- alter table employee add foreign key (Mgr_SSN) references
employee(Emp_ID);
-- alter table balance add foreign key (Emp_ID) references employee(Emp_ID);
-- alter table current_acc add foreign key (Acc_no) references
account(Acc_no);
-- alter table recurring_deposit_acc add foreign key (Acc_no) references
account(Acc_no);
-- alter table nri_acc add foreign key (Acc_no) references account(Acc_no);
-- alter table fixed_deposit_acc add foreign key (Acc_no) references
account(Acc_no);

-- alter table savings_acc add foreign key (Acc_no) references account(Acc_no);

-- alter table user add foreign key(User_name) references login(User_name);
```

SQL queries

QUERIES TO CREATE THE RELATIONS AND POPULATE DATABASE:

Basic queries

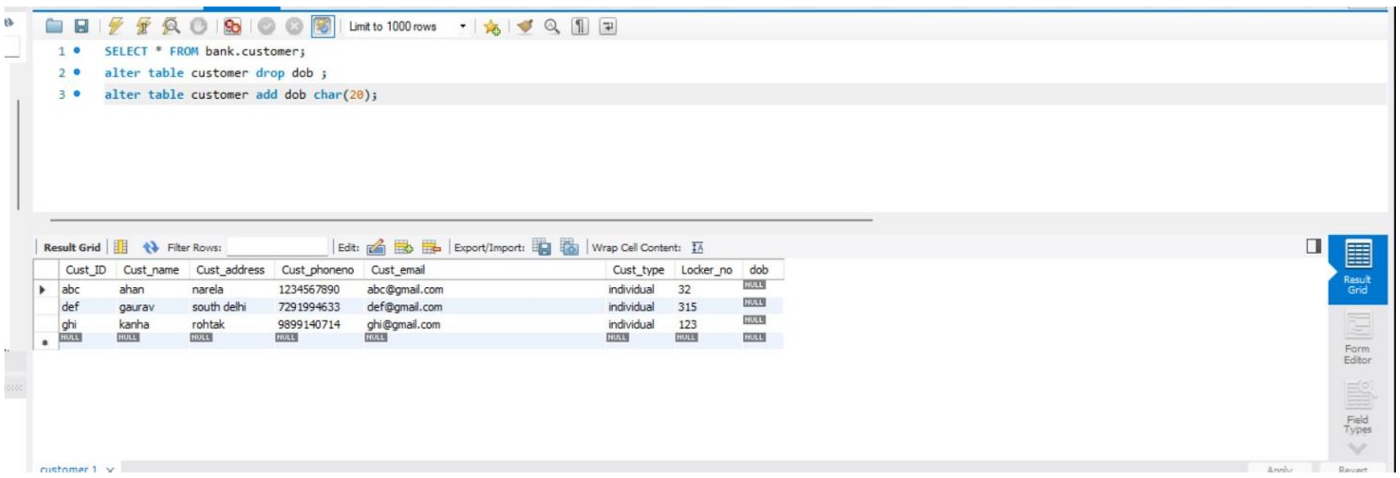
Question 1: Retrieve loan details where loan id = 201



```
use bank;

select loan_amt from loan
where loan_id=201;
```

loan_amt
3000000

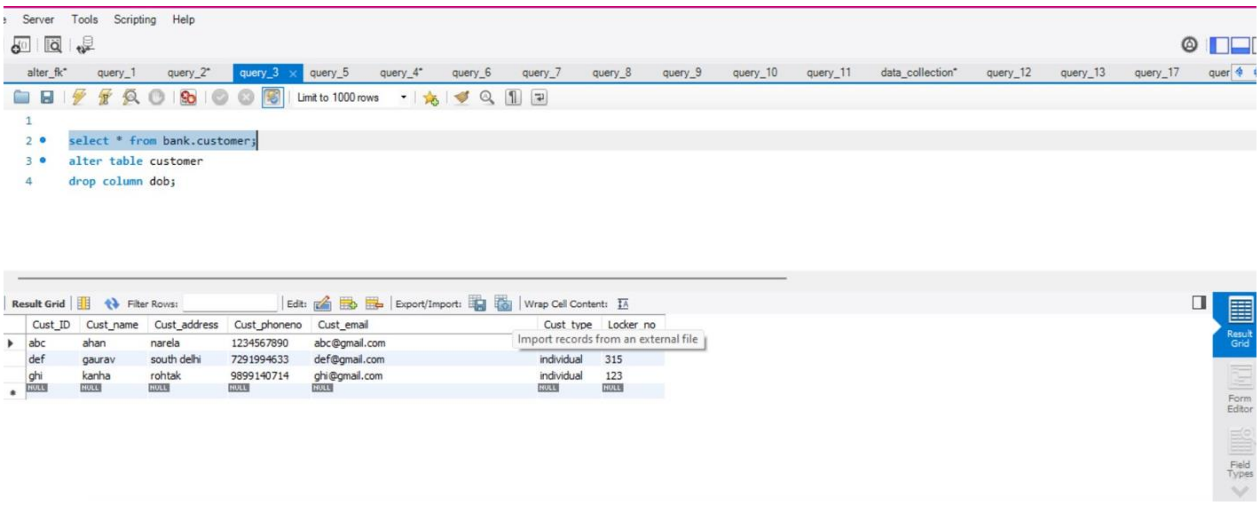
Question 2: Add attribute date of birth (dob) for customer



```
SELECT * FROM bank.customer;
alter table customer drop dob ;
alter table customer add dob char(20);
```

Cust_ID	Cust_name	Cust_address	Cust_phoneno	Cust_email	Cust_type	Locker_no	dob
abc	ahan	narela	1234567890	abc@gmail.com	individual	32	NULL
def	gaurav	south delhi	7291994633	def@gmail.com	individual	315	NULL
ghi	kanha	rohtak	9899140714	ghi@gmail.com	individual	123	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

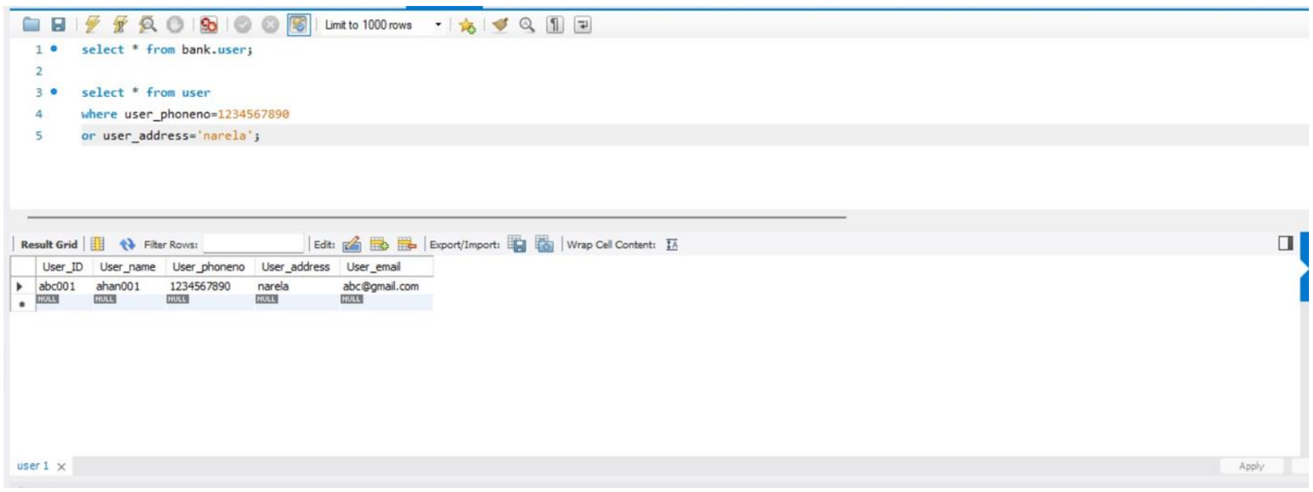
Question 3: Drop attribute dob for customer


```
select * from bank.customer;
alter table customer
drop column dob;
```

Cust_ID	Cust_name	Cust_address	Cust_phoneno	Cust_email	Cust_type	Locker_no
abc	ahan	narela	1234567890	abc@gmail.com	[illegible]	[illegible]
def	gaurav	south delhi	7291994633	def@gmail.com	individual	315
ghi	kanha	rohtak	9899140714	ghi@gmail.com	individual	123
NULL	NULL	NULL	NULL	NULL	NULL	NULL

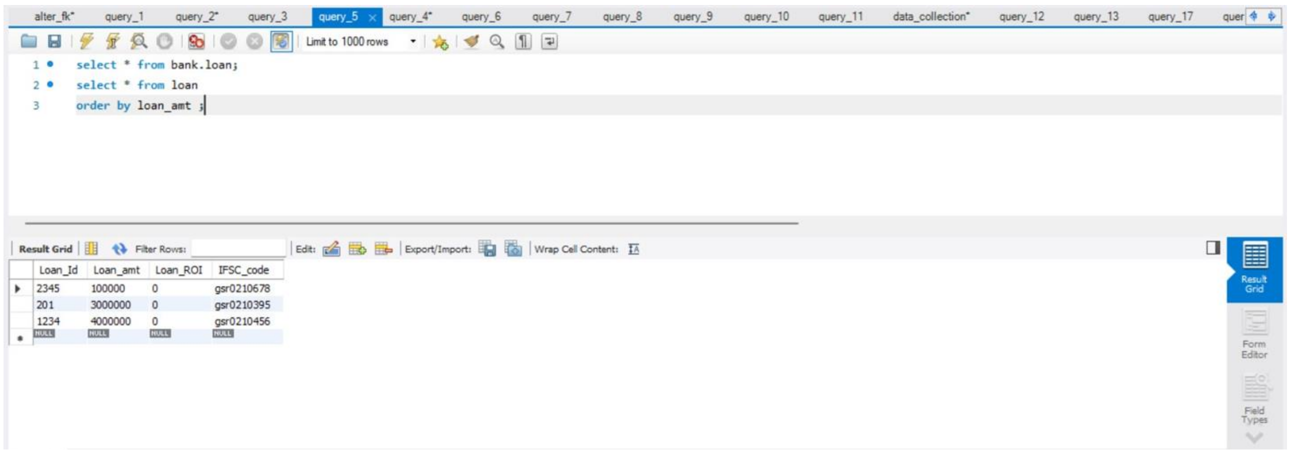
Import records from an external file

Question 4: Retrieve user details for user having mobile no. = 1234567890 and address = 'Narela'


```
select * from bank.user;

select * from user
where user_phoneno=1234567890
or user_address='narela';
```

User_ID	User_name	User_phoneno	User_address	User_email
abc001	ahan001	1234567890	narela	abc@gmail.com
NULL	NULL	NULL	NULL	NULL

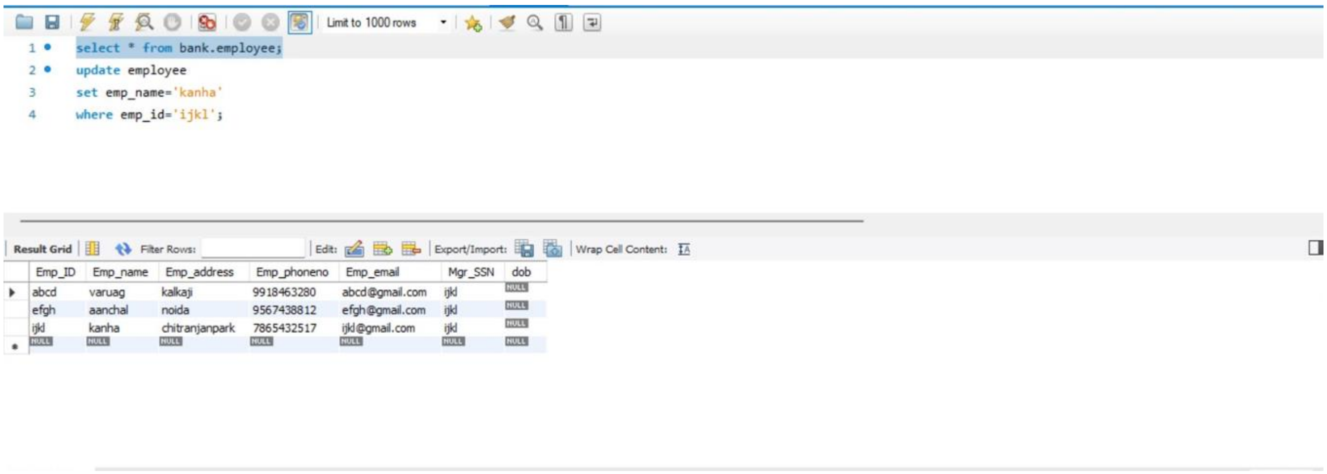
Question 5: Retrieve loan details order by ascending loan amount


alter_fk* query_1 query_2* query_3 query_5 query_4* query_6 query_7 query_8 query_9 query_10 query_11 data_collection* query_12 query_13 query_17
Limit to 1000 rows
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Form Editor
Field Types

```sql
select * from bank.loan;
select * from loan
order by loan_amt ;
```

Loan_Id	Loan_amt	Loan_ROI	IFSC_code
2345	100000	0	gsr0210678
201	3000000	0	gsr0210395
1234	4000000	0	gsr0210456
NULL	NULL	NULL	NULL

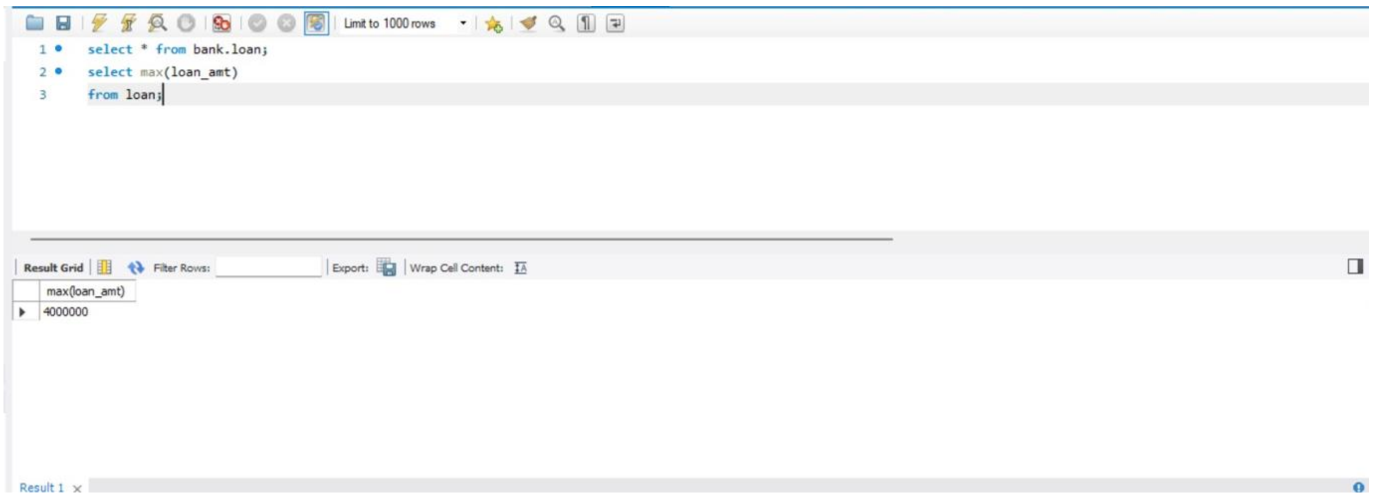
Question 6: Update name of employee having employee ID = 'ijkl'



```sql
select * from bank.employee;
update employee
set emp_name='kanha'
where emp_id='ijkl';
```

Emp_ID	Emp_name	Emp_address	Emp_phoneno	Emp_email	Mgr_SSN	dob
abcd	varuag	kalkaji	9918463280	abcd@gmail.com	ijkl	NULL
efgh	aanchal	noida	9567438812	efgh@gmail.com	ijkl	NULL
ijkl	kanha	chitranjanpark	7865432517	ijkl@gmail.com	ijkl	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

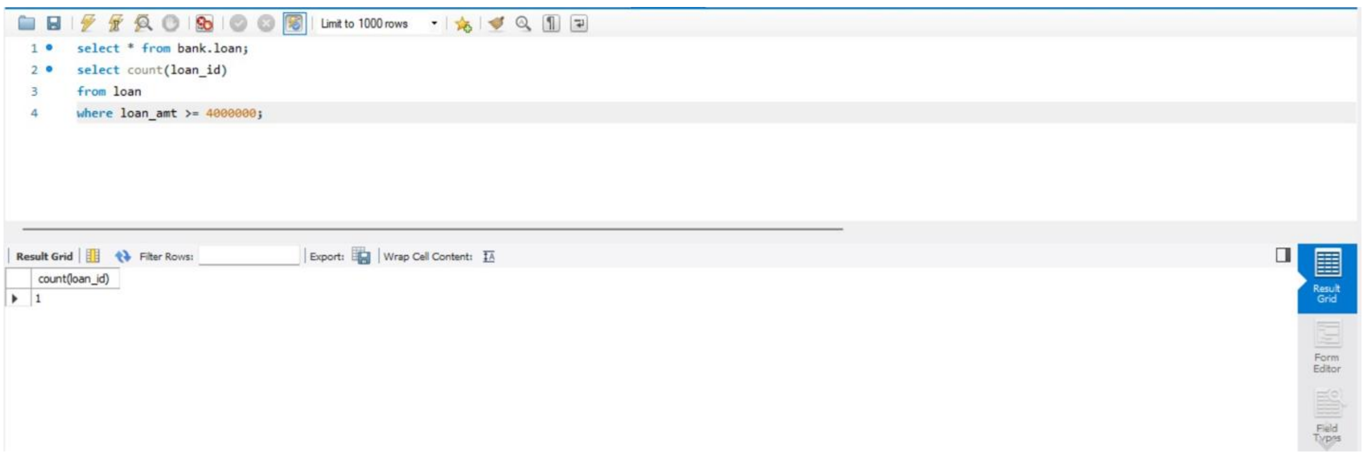
Question 7: Retrieve maximum amount of loan lend by any customer from bank



```
select * from bank.loan;
select max(loan_amt)
from loan;
```

```
max(loan_amt)
4000000
```

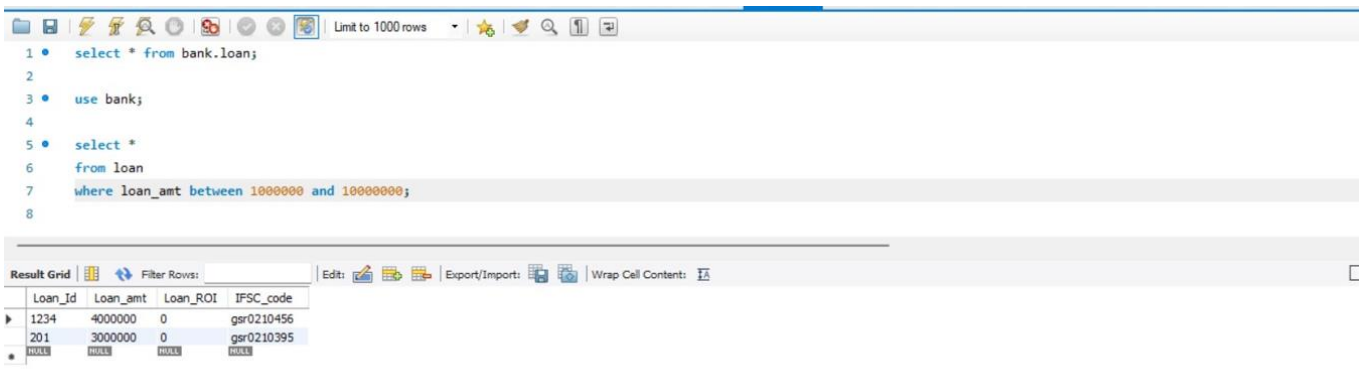
Question 8: Count the distinct users that have taken loan of and above 4000000



```
select * from bank.loan;
select count(loan_id)
from loan
where loan_amt >= 4000000;
```

```
count(loan_id)
1
```

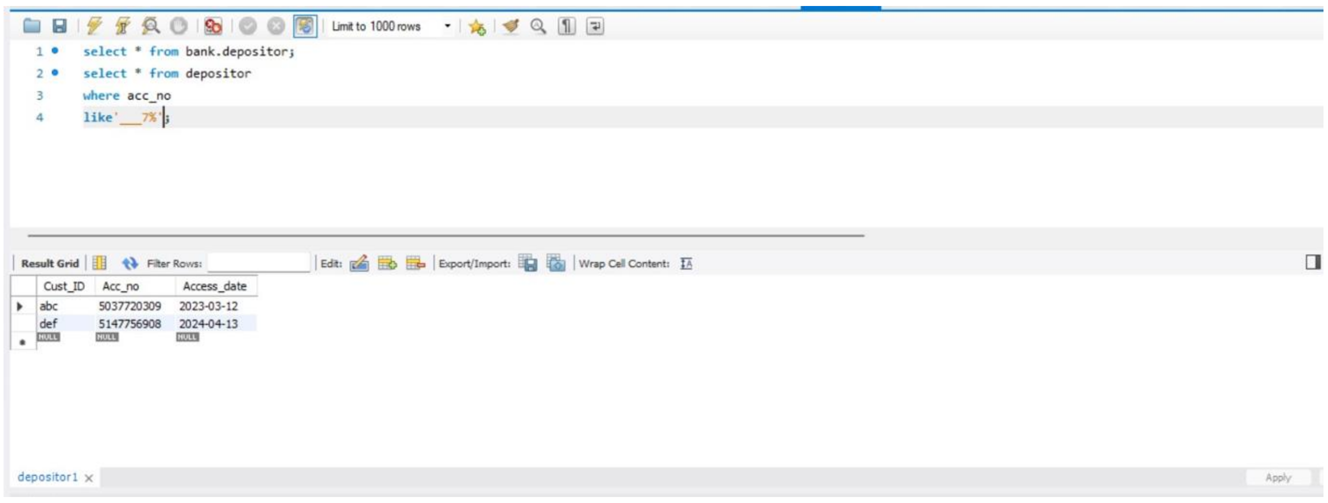
Question 9: Retrieve loan details of user who have lend a loan between 1000000 to 100000000



```sql
select * from bank.loan;

use bank;

select *
from loan
where loan_amt between 1000000 and 10000000;
```

Loan_Id	Loan_amt	Loan_ROI	IFSC_code
1234	4000000	0	gsr0210456
201	3000000	0	gsr0210395
NULL	NULL	NULL	NULL

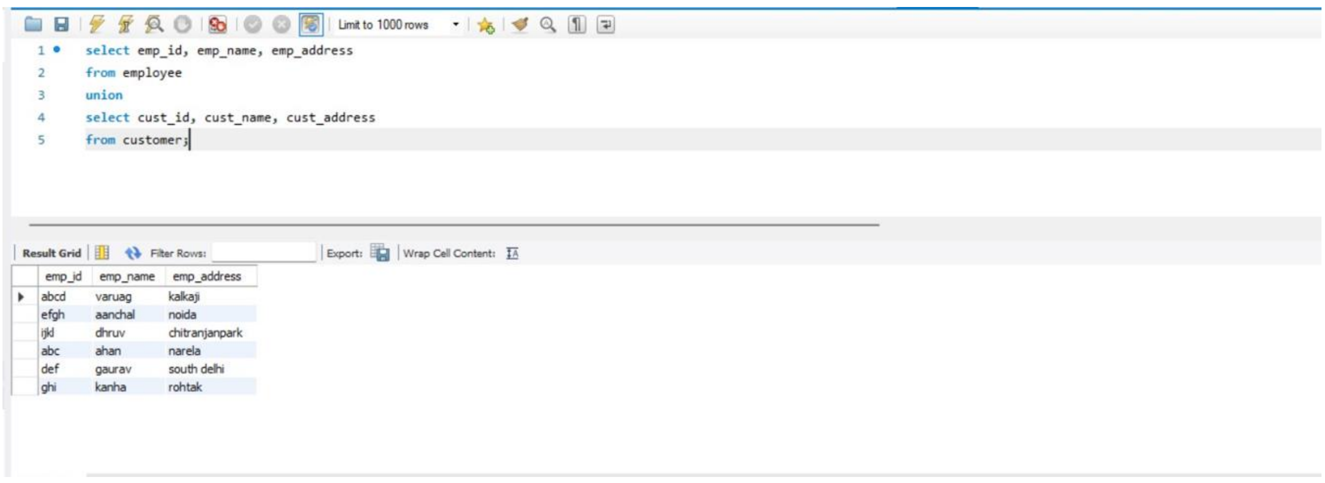
Question 10: Retrieve depositor details for user having 7 in their account no. anywhere



```sql
select * from bank.depositor;
select * from depositor
where acc_no
like'___7%';
```

Cust_ID	Acc_no	Access_date
abc	5037720309	2023-03-12
def	5147756908	2024-04-13
NULL	NULL	NULL

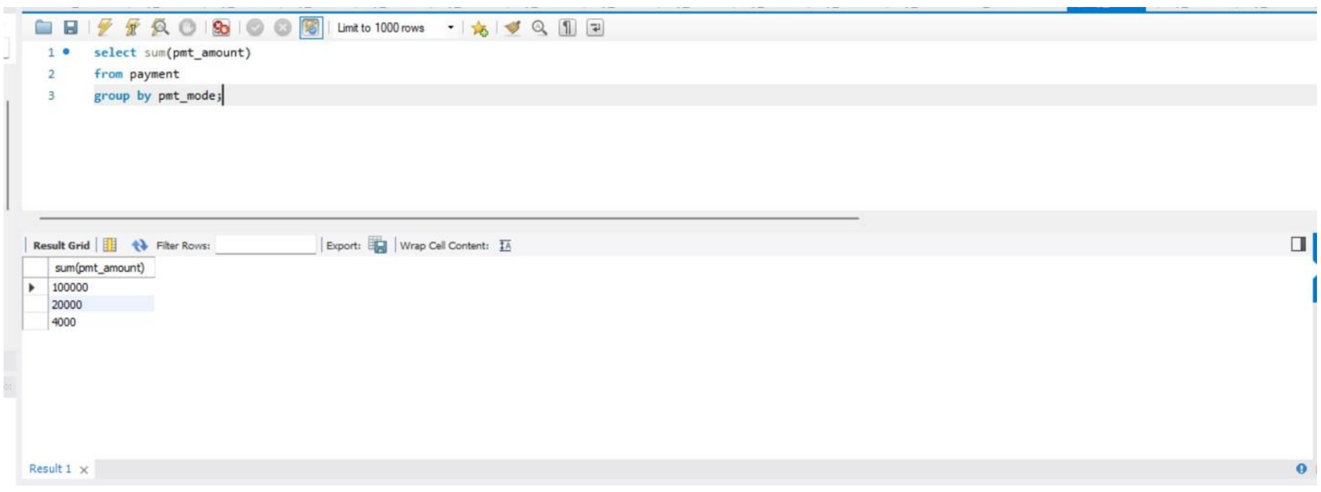
Question 11: Retrieve combine data of employee data and customer table as employee ID , employee names, employee address



```
select emp_id, emp_name, emp_address
from employee
union
select cust_id, cust_name, cust_address
from customer;
```

```
emp_id  emp_name  emp_address
abcd    varuag    kalkaji
efgh    aanchal   noida
ijkl    dhruv     chitranjanpark
abc     ahan      narela
def     gaurav    south delhi
ghi     kanha     rohtak
```

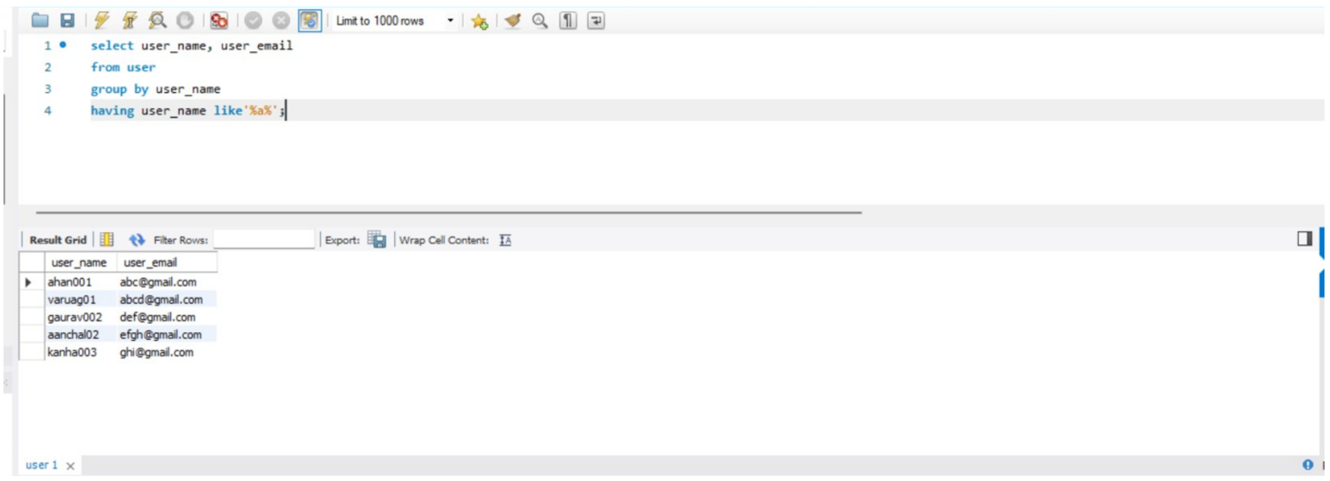
Question 12: Find total amount of payments performed in each payment mode



```
select sum(pmt_amount)
from payment
group by pmt_mode;
```

```
sum(pmt_amount)
100000
20000
4000
```

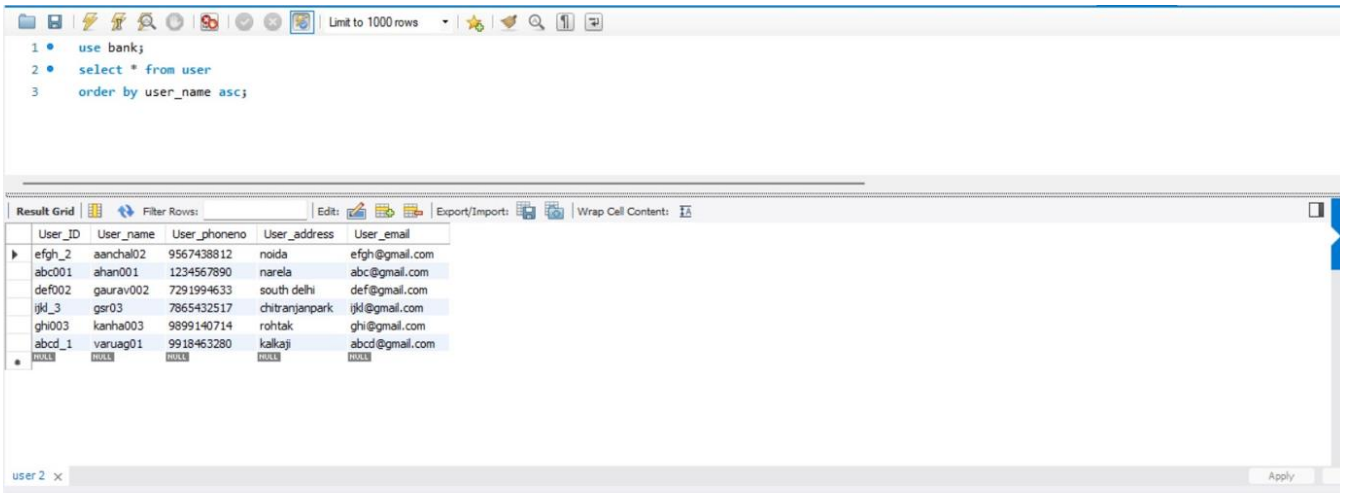
Question 13: Retrieve user details of user having 'a' in their name.



```
select user_name, user_email
from user
group by user_name
having user_name like'%a%';
```

user_name	user_email
ahan001	abc@gmail.com
varuag01	abcd@gmail.com
gaurav002	def@gmail.com
aanchal02	efgh@gmail.com
kanha003	ghi@gmail.com

Question 14: Retrieve user data in ascending order of their names

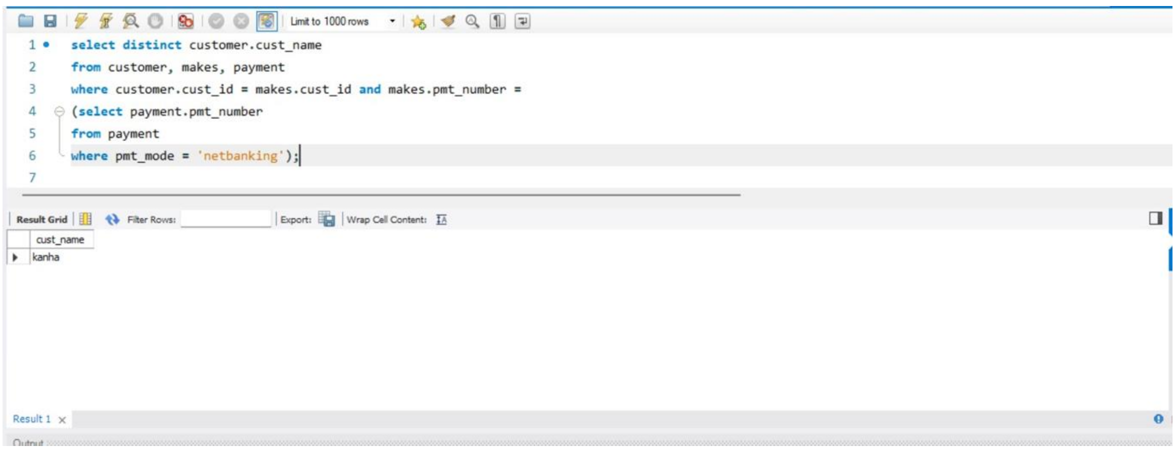


```
use bank;
select * from user
order by user_name asc;
```

User_ID	User_name	User_phoneno	User_address	User_email
efgh_2	aanchal02	9567438812	noida	efgh@gmail.com
abc001	ahan001	1234567890	narela	abc@gmail.com
def002	gaurav002	7291994633	south delhi	def@gmail.com
ijkl_3	gsr03	7865432517	chitranjanpark	ijkl@gmail.com
ghi003	kanha003	9899140714	rohtak	ghi@gmail.com
abcd_1	varuag01	9918463280	kalkaji	abcd@gmail.com
NULL	NULL	NULL	NULL	NULL

Advance queries

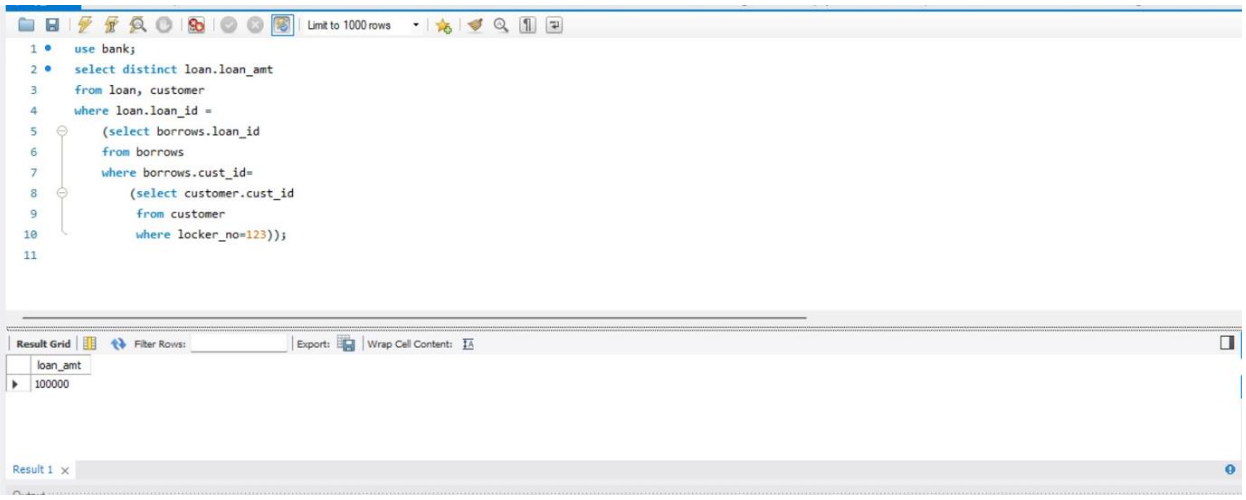
Question 15: Retrieve names of all customers who have performed payment via net banking



```
select distinct customer.cust_name
from customer, makes, payment
where customer.cust_id = makes.cust_id and makes.pmt_number =
(select payment.pmt_number
from payment
where pmt_mode = 'netbanking');
```

```
cust_name
kanha
```

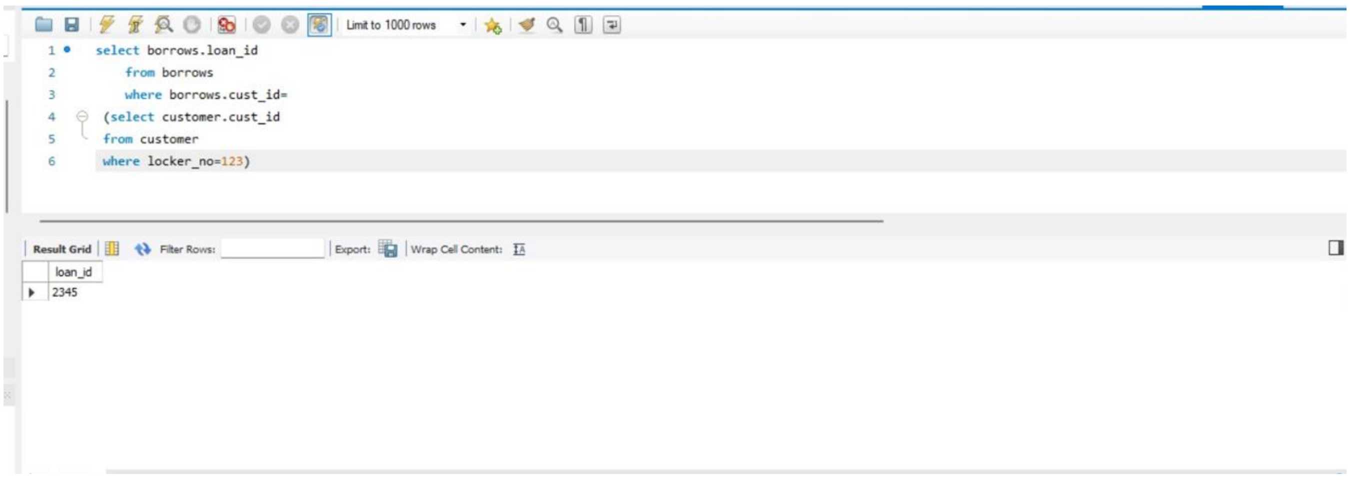
Question 16: retrieve loan amount taken by the customer having locker no. = 123



```
use bank;
select distinct loan.loan_amt
from loan, customer
where loan.loan_id =
    (select borrows.loan_id
    from borrows
    where borrows.cust_id=
        (select customer.cust_id
         from customer
         where locker_no=123));
```

```
loan_amt
100000
```

Question 17: Loan details of customer with locker no '123'

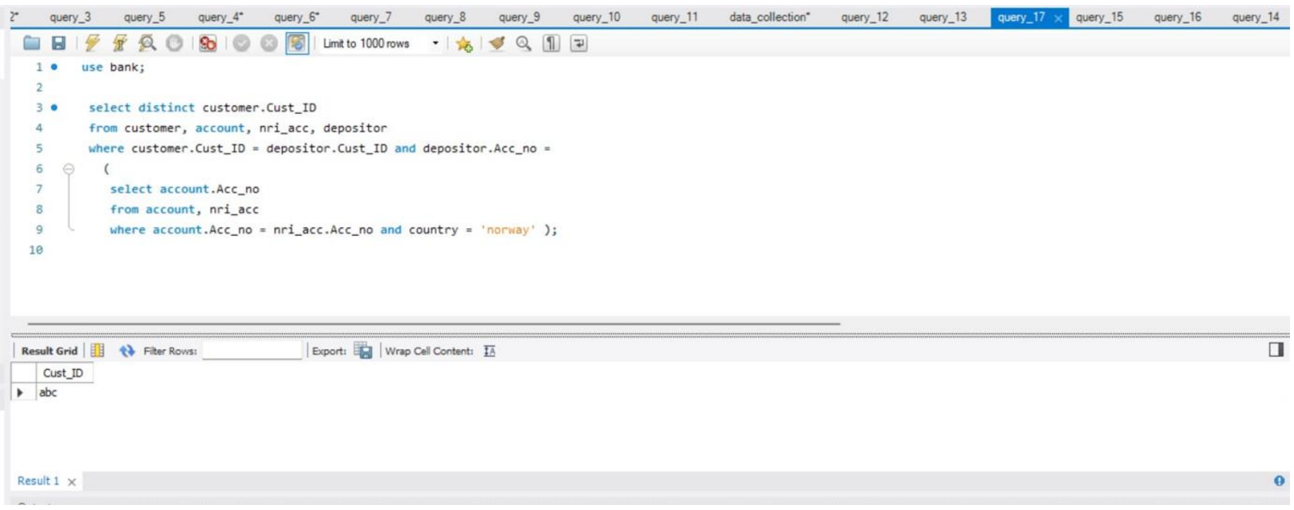


```
select borrows.loan_id
   from borrows
   where borrows.cust_id=
 (select customer.cust_id
 from customer
 where locker_no=123)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
loan_id
2345
```

Question 18: Retrieve ID of all customers having account in bank. That are from 'Norway



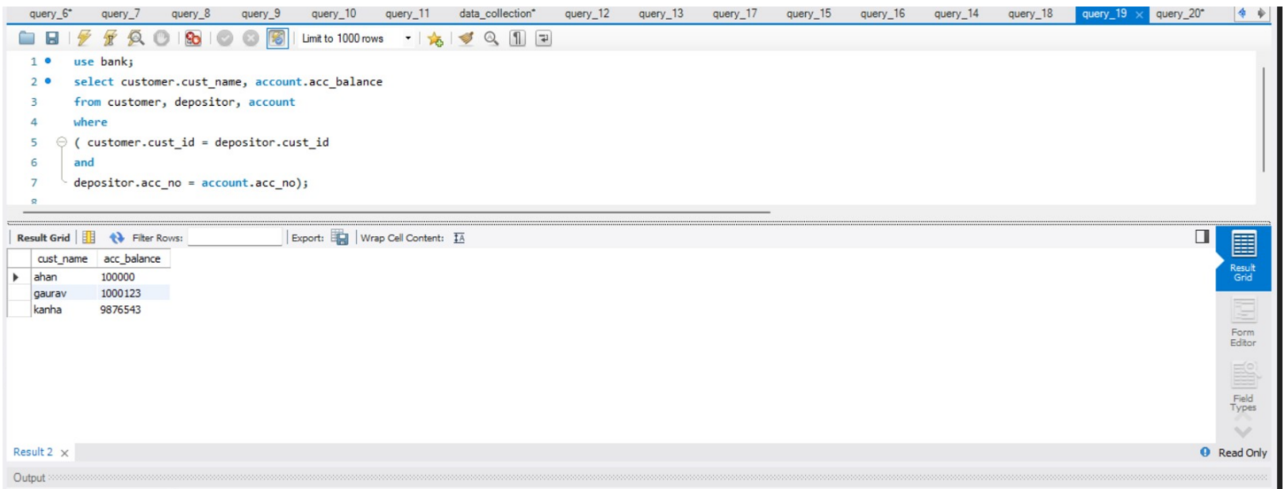
```
use bank;

 select distinct customer.Cust_ID
 from customer, account, nri_acc, depositor
 where customer.Cust_ID = depositor.Cust_ID and depositor.Acc_no =
   (
    select account.Acc_no
    from account, nri_acc
    where account.Acc_no = nri_acc.Acc_no and country = 'norway' );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
Cust_ID
abc
```

Result 1

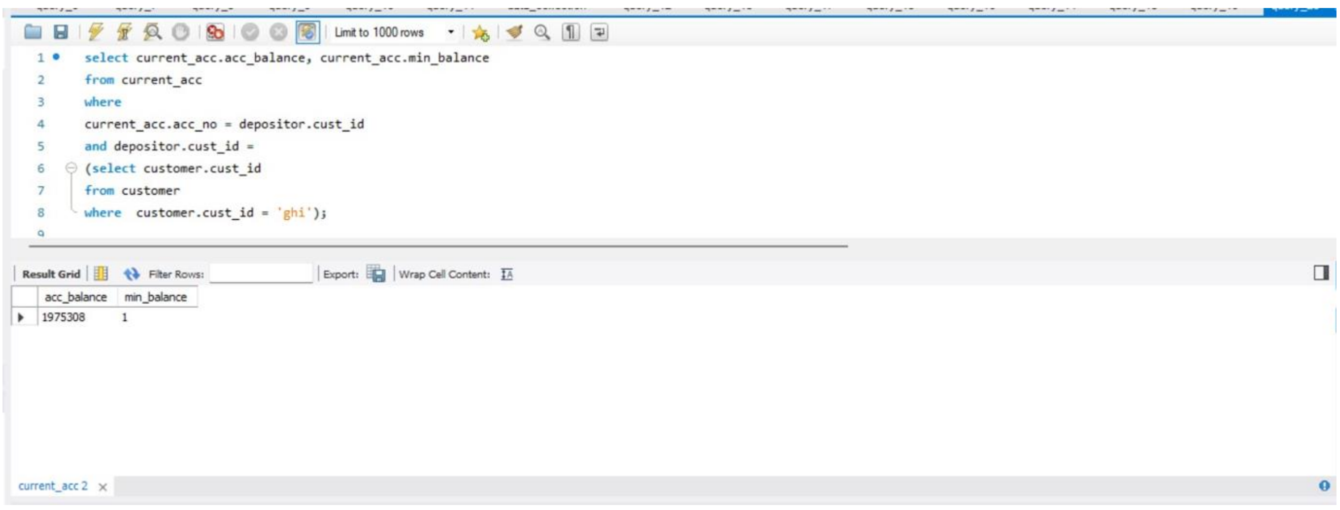
Question 19: Show customer names with their account balance.



```
use bank;
select customer.cust_name, account.acc_balance
from customer, depositor, account
where
( customer.cust_id = depositor.cust_id
and
depositor.acc_no = account.acc_no);
```

```
cust_name  acc_balance
ahan       100000
gaurav     1000123
kanha      9876543
```

Question 20: Show account balance and account minimum balance limit for customer having id='ghi'.



```
select current_acc.acc_balance, current_acc.min_balance
from current_acc
where
current_acc.acc_no = depositor.cust_id
and depositor.cust_id =
(select customer.cust_id
from customer
where  customer.cust_id = 'ghi');
```

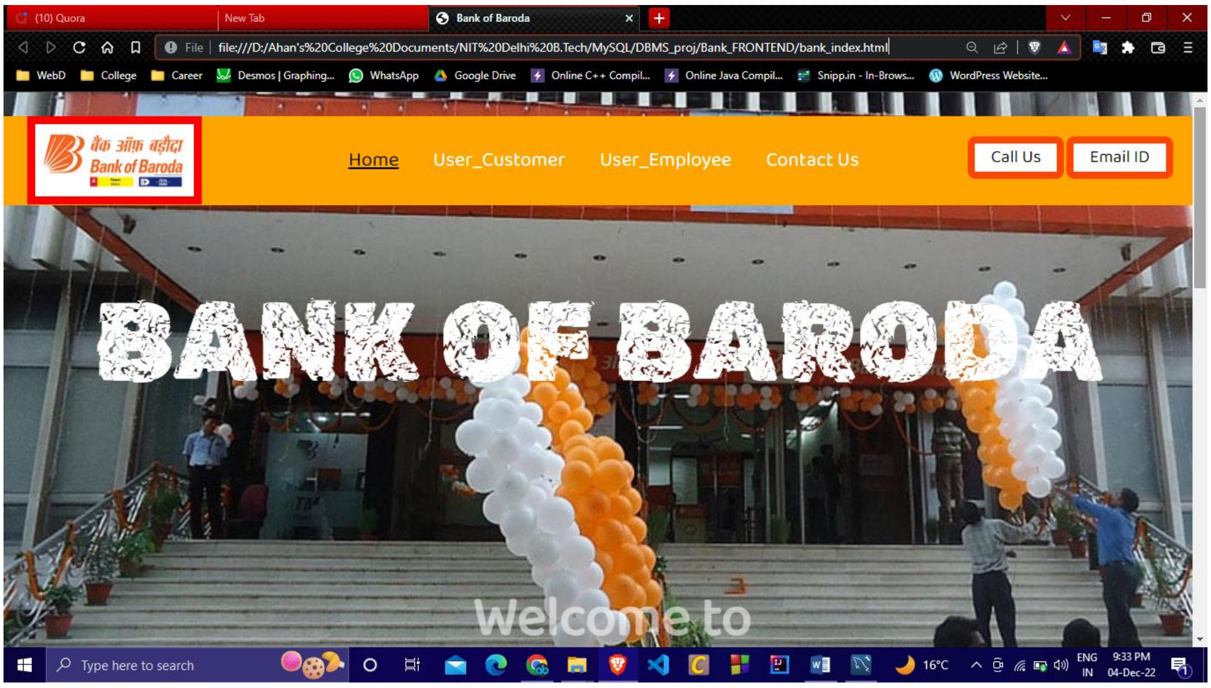
```
acc_balance  min_balance
1975308      1
```

Front End

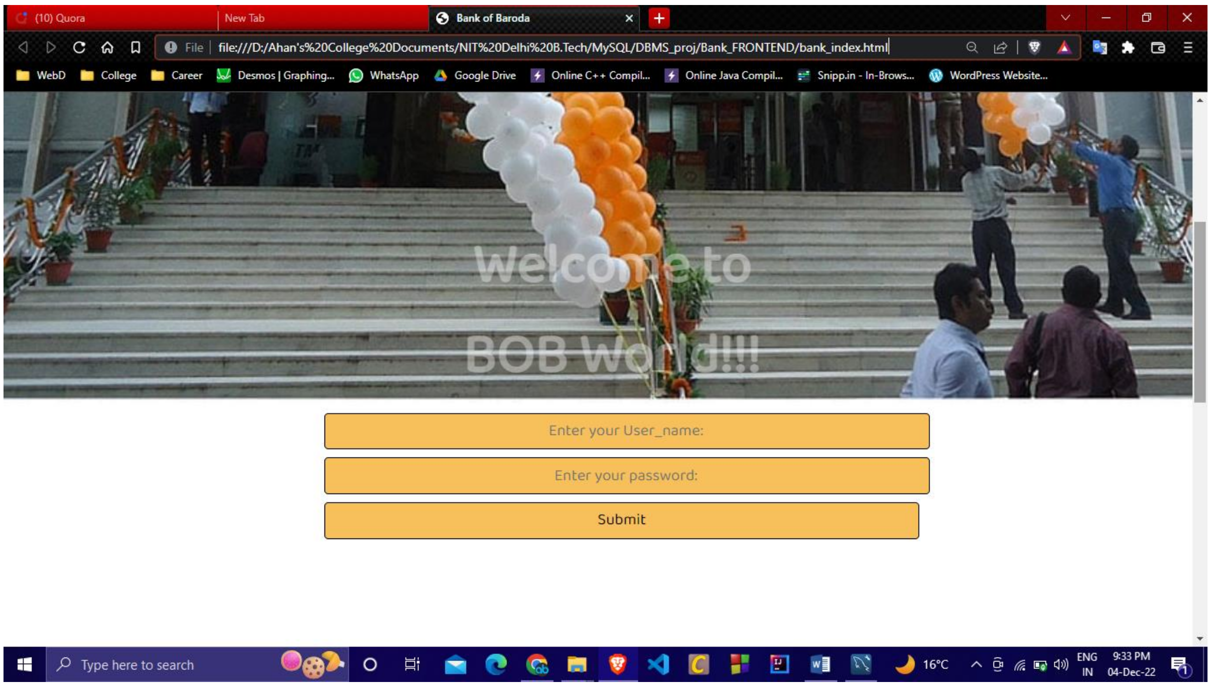
The Front End of the Bank website involves interconnected webpages to form a crude website of Bank of Baroda prototype.

Tech stack used includes:

- **HTML**
- **CSS**
- **JavaScript**
- **PHP for connectivity**



file:///D:/Ahan's%20College%20Documents/NIT%20Delhi%20B.Tech/MySQL/DBMS_proj/Bank_FRONTEND/bank_index.html
Home
User_Customer
User_Employee
Contact Us
Call Us
Email ID
BANK OF BARODA
Welcome to


(10) Quora
New Tab
Bank of Baroda
file:///D:/Ahan's%20College%20Documents/NIT%20Delhi%20B.Tech/MySQL/DBMS_proj/Bank_FRONTEND/bank_index.html
WebD
College
Career
Desmos | Graphing...
WhatsApp
Google Drive
Online C++ Compil...
Online Java Compil...
Snipp.in - In-Brows...
WordPress Website...
Welcome to
BOB World!!!
Enter your User_name:
Enter your password:
Submit
Type here to search
16°C
ENG IN
9:33 PM
04-Dec-22

BIBLIOGRAPHY

https://www.bankofbaroda.in/

https://www.youtube.com/watch?v=qlLpDBSOe3o

https://miro.com/

https://www.youtube.com/playlist?list=PLu0W_9lII9agiCUZYRsvtGTXdxkzPyItg

https://www.w3schools.com/php/php_mysql_connect.asp

https://www.cs.purdue.edu/homes/bb/cs448_Fall2017/lpdf/Chapter09.pdf