

*Database Management System*  
*1<sup>st</sup> semester - 2022*

# PROJECT EVALUATION

Project name: Online Banking System

Student's information:

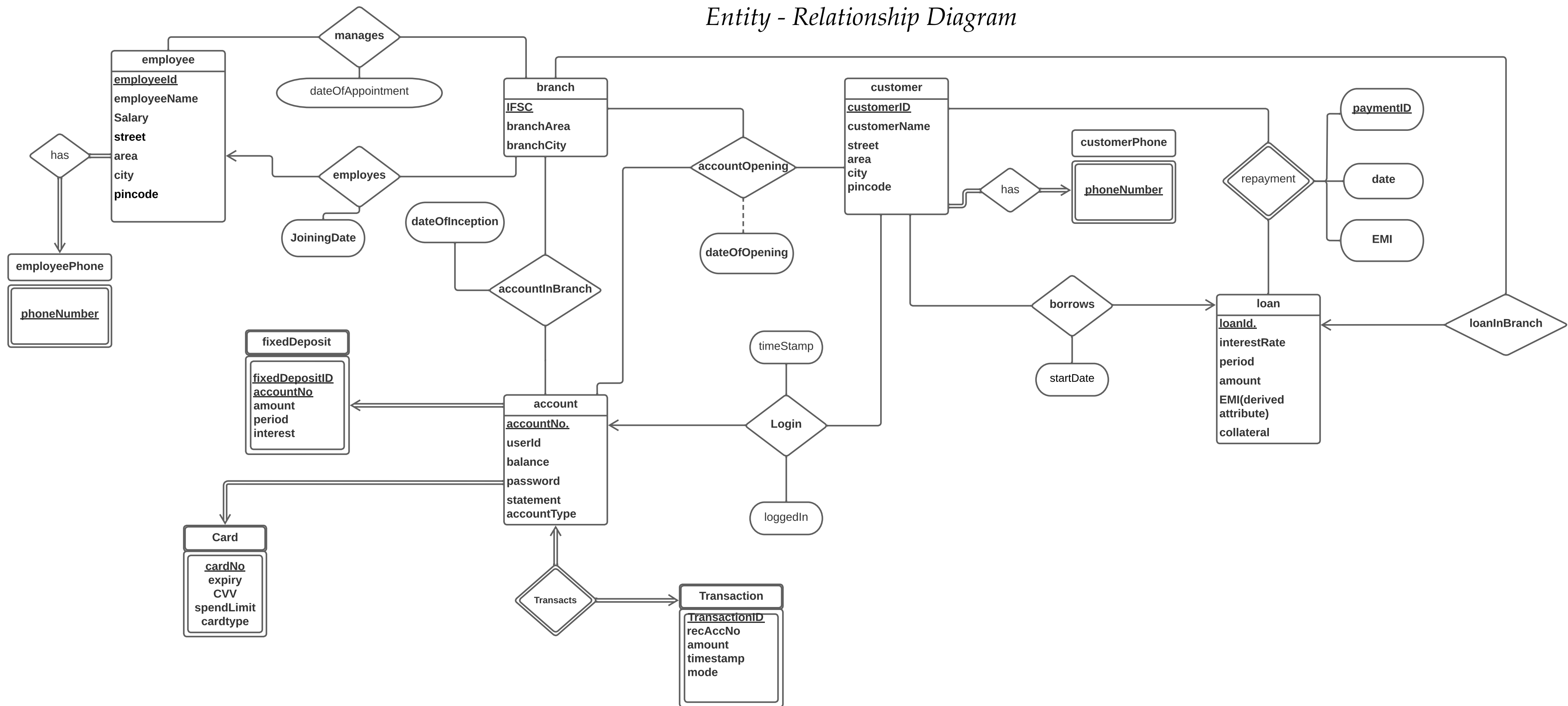
- Name: Phạm Nhật Minh
- ID: 21020651
- Class: K66T-CLC
- Phone: 0901728992
- Email: 21020651@vnu.edu.vn

# Project overview:

Traditional banking is time consuming since going to the bank for simple tasks such as money transfers, loans,... has long been considered tedious.

Therefore, in today's evergrowing importance of the Internet, I have implemented the database of an Online Banking System. The application will save time, redundancy of data and provide hassle-free access to various operations such as funds transfer, loans, fixed deposit, balance check, statement history,...

# Entity - Relationship Diagram



## Entities:

1. Account
2. Customer
3. Employee
4. Branch
5. Loan
6. Fixed\_Deposit
7. Cards
8. Transaction

## Relations:

1. Account\_Opening :
  - customer, Branch and Account
2. AccountInBranch:
  - account and branch
3. Loan\_Branch
  - Loan and Branch
4. Login:
  - Customer and Account
5. Transacts:
  - Account and Transaction
6. Borrow:
  - Loan and Customer
7. Repayment:
  - Customer and Loan
8. Employed:
  - Branch, Employee
9. Manages:
  - Branch, Employee

### **Weak Entity :**

- Transaction : As it cannot uniquely identified with a account
- Repayment : It cannot exist without any loan
- FixedDeposit :It cannot exist without any account
- Card: It cannot exist without account
- EmployeePhone : It cannot exist without employee
- CustomerPhone: It cannot exist without customer

### **Ternary Relationship :**

A ternary relationship is when three entities participate in the relationship

- AccountOpening : Since it have 3 entities :Branch,customer and account

### **Entities participation type,Relationship roles and Constraints:**

Account balance constraint : Cannot perform transaction if balance is null.

Account opening constraint: **Cardinality Constraints on Ternary Relationship**  
(One customer at one branch can only have one account)

### Relational Model:

account(accountNo, userID, balance, password, accountType)

customer(customerID, customerName, creditScore, street, area, city, pincode)

customerPhone(customerID, phoneNumber)

employee(employeeID, employeeName, salary, street, area, city, pincode)

employeePhone(employeeID, phoneNumber)

branch(IFSC, branchArea, branchCity)

loan(loanID, customerID, interestRate, period, amount, EMI (derived), collateral)

fixedDeposit(fixedDepositID, accountNo, amount, period, interest)

card(cardNo, accountNo, Expiry, CVV, cardType, spendLimit)

Receiver(recAccNo, IFSC)

accountOpening(accountNo, customerID, IFSC, dateOfOpening)

accountInBranch(accountNo, dateOfInception, IFSC)

loanInBranch(loanID, IFSC)

login(accountNo, time\_Stamp, CustomerID, loggedIn)

transaction(transactionID, accountNo., recAccNo., mode, amount,  
time\_Stamp)

lockIn(accountNo, fixedDepositID, startDate)

borrow(loanID, customerID, startDate)

repayment(paymentID, customerID, loanID, dateOfPayment, EMI)

employs (employeeID, joiningDate, IFSC)

manages(employeeID, dateOfAppointment, IFSC)



## ALWAYS FAITHFUL

EMPLOYEE

CUSTOMER



## EMPLOYEE OR MANAGER

LOGIN AS MANAGER?

LOGIN

LOGIN AS EMPLOYEE?

LOGIN





## EMPLOYEE VIEW OF CUSTOMERS

Customer Name	Credit Score	Account Number	Balance	Account Type	Time
bytearray(b'Meet Popat')	999	bytearray(b'1111111111111111')	9.992992292929293e+16	bytearray(b'savings')	
bytearray(b'Meet Popat')	999	bytearray(b'2111111111111111')	9.992992292929293e+16	bytearray(b'CURRENT')	
bytearray(b'Shouiya Jindal')	9	bytearray(b'1111111111111112')	-100.0	bytearray(b'SAVINGS')	
bytearray(b'Shouiya Jindal')	9	bytearray(b'2111111111111112')	-100.0	bytearray(b'current')	
bytearray(b'Aditya Nangia')	899	bytearray(b'1111111111111113')	10.0	bytearray(b'savings')	
bytearray(b'Aditya Nangia')	899	bytearray(b'2111111111111113')	10.0	bytearray(b'CURRENT')	
bytearray(b'Sufyan Azam')	900	bytearray(b'1111111111111114')	999.0	bytearray(b'SAVING')	
bytearray(b'Sufyan Azam')	900	bytearray(b'2111111111111114')	999.0	bytearray(b'current')	
bytearray(b'Pavitra Gupta')	80	bytearray(b'1111111111111115')	50.0	bytearray(b'savings')	
bytearray(b'Pavitra Gupta')	80	bytearray(b'2111111111111115')	50.0	bytearray(b'CURRENT')	



phpMyAdmin

Server: 127.0.0.1 / Database: onlinebank / View: employee\_view\_cust

Recent Favorites

- mysql
- net\_banking
- onlinebank

Tables

- New
- account
- accountinbranch
- accountopening
- branch
- branch
- card
- customer
- customerphone
- employee
- employeephone
- employees
- fixeddeposit
- loan
- loaninbranch
- login
- manages
- repayment
- transaction

Views

- new
- employee\_view\_cust
- manager\_view\_cust

Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesired behavior.

Showing rows 0 - 9 (10 total). Query took 0.0003 seconds.

SELECT \* FROM `employee\_view\_cust`

Profiling

[ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Showall Number of rows: Filter rows: Search this table

Extra options:

	customerName	creditScore	accountNo	balance	accountType
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Meet Popat	999	1111111111111111	9.992992292929293e16	savings
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Meet Popat	999	2111111111111111	9.992992292929293e16	CURRENT
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Shouiya Jindal	9	1111111111111112	-100	SAVINGS
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Shouiya Jindal	9	2111111111111112	-100	current
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Aditya Nangia	899	1111111111111113	10	savings
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Aditya Nangia	899	2111111111111113	10	CURRENT
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sufyan Azam	900	1111111111111114	999	SAVING
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Sufyan Azam	900	2111111111111114	999	current
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Pavitra Gupta	80	1111111111111115	50	savings
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Pavitra Gupta	80	2111111111111115	50	CURRENT

Check all With selected: Edit Copy Delete Export

Console: all Number of rows: 25 Filter rows: Search this table

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)



## MANAGER VIEW OF EMPLOYEES

Employee ID	Employee Name	Salary
1	bytearray(b'sibin')	30000
2	bytearray(b'rahul')	40000
3	bytearray(b'saksham')	50000
4	bytearray(b'subham')	60000
5	bytearray(b'BANSAL')	70000



phpMyAdmin

Server: 127.0.0.1 / Database: onlgebank / View: manager\_view\_emp

Browse Structure SQL Search Insert Export Privileges Operations

Recent Favorites

Current selection does not contain a unique column. Grid edit, Copy and Delete features may result in undesired behavior.

Showing rows 0 - 4 (5 total). Query took 0.0003 seconds.

SELECT \* FROM 'manager\_view\_emp'

Profiling

[ Edit \_ \_ \_ ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: Filter rows: Search this table

Extra options

	employeeID	employeeName	salary
<input type="checkbox"/> Edit Copy Delete	1	sibin	30000
<input type="checkbox"/> Edit Copy Delete	2	rahul	40000
<input type="checkbox"/> Edit Copy Delete	3	saksham	50000
<input type="checkbox"/> Edit Copy Delete	4	subham	60000
<input type="checkbox"/> Edit Copy Delete	5	BANSAL	70000

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

Views

- new
- employee\_view.s
- manager\_view.s
- manager\_view.e

performance.schema

phpmyadmin

test



## TRANSACTIONS

Transaction ID	Account Number	Receiver Account Number	Transaction Mode	Amount	Time
2345	bytearray(b'1111111111111111')	bytearray(b'1111111111111112')	bytearray(b'Card')	3000	2022-01-01 00:00:01



phpMyAdmin

Server: 127.0.0.1 / Database: onlinebank / Table: transaction

Showing rows 0 - 4 (5 total, Query took 0.0002 seconds)

SELECT \* FROM 'transaction'

Query results operations

	transactionID	accountNo	recAccto	mode	amount	time Stamp
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1234	1111111111111114	1111111111111115	UPI	2000	2020-01-01 00:00:01
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2345	1111111111111111	1111111111111112	Card	3000	2022-01-01 00:00:01
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	3456	1111111111111113	2111111111111115	Online	5000	2021-01-01 00:00:01
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	4567	2111111111111115	1111111111111114	UPI	6000	2022-01-07 00:00:01
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	5567	2111111111111112	1111111111111111	Card	8000	2022-06-07 00:00:01

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

FIXED DEPOSITS

VIEW CURRENT FIXED DEPOSIT

APPLY FOR NEW FIXED DEPOSIT

FD AMOUNT

Amount

FD Period

Period

INTEREST = 5%

SUBMIT

FD

Fixed ID	Interest Rate (in %)	Period (in years)	Amount	
bytearray(b'1111111111111111')	1	7	10	100000.0
bytearray(b'1111111111111111')	2	7	5	200000.0

## QUERIES AVAILABLE IN THE DATABASE-

### User Mode:

- View Account Info of a customer (Show everything in account through given customer ID in a branch)
- View Personal Information of self (Show customerName, creditScore, street, area, city, pincode through customerID)
- View All Transactions (Show transactionID, accountNo., mode, amount, time\_Stamp and recAccNo., IFSC from receiver)
- View Balance/Statement
- Update Password
- Funds Transfer (Withdraw, Deposit through card or netbanking)
- Repay Loan
- Create a fixed deposit

### Manager/Admin Mode:

- Create an account (Fill customerID, customerName, creditScore, street, area, city, pincode)
- Delete an account
- View All Accounts of every customer
- View personal data of employees
- View Transactions History of every account
- Update personal data of employees

## EMBEDDED QUERIES-

User-                   **// Login by password and customerId**

View Details (input- customerId)

```
SELECT customerName, street, area, city, pincode FROM customer C
WHERE C.customerID = 'customerId' and exists (SELECT L.customerID FROM Login L
WHERE
L.customerId = 'customerId' and L.loggedIn = True);
```

View All Transactions (Input- accountNo)

```
SELECT * FROM transaction WHERE accountNo = 'accountNo';
```

View Balance (Input- accountNo)

```
SELECT balance FROM account WHERE accountNo = 'accountNo';
```

Funds Withdrawl (Input- accountNo, amount)

```
START transaction;
SET transaction READ WRITE;
UPDATE account set balance = balance - 'given amount'
WHERE accountNo = 'accountNo';
commit;
```

Funds Deposit (Input- accountNo, amount)

```
START transaction;
SET transaction READ WRITE;
UPDATE account set balance = balance + 'given amount'
WHERE accountNo = 'accountNo';
commit;
```

Funds Transfer (Input- accountNo, recAccountNo, amount)

```
START transaction;
SET transaction READ WRITE;
UPDATE account set balance = balance - 'given amount'
WHERE accountNo = 'accountNo';
SAVEPOINT s1;
UPDATE account set balance = balance + 'given amount'
WHERE accountNo = 'RecAccountNo';
commit;
```

Create a Fixed Deposit (Input- accountNo, amount, period, interest)

```
START transaction;                   // TODO
SET transaction READ WRITE;
```

Change Name (Input- accountNo, updated customer name)

```
UPDATE customer SET customerName = 'Updated Name' WHERE customerID in (SELECT L.customerID from login L WHERE L.accountNo = 'accountNo' and L.loggedIn = True);
```

Change Password (Input- accountNo, updated password)

```
UPDATE account SET password = 'Updated Password' WHERE accountNo in (SELECT L.accountNo from login L WHERE L.accountNo = 'accountNo' and L.loggedIn = True);
```

Change PhoneNo (Input- accountNo, updated phoneNo)

```
UPDATE customerPhone SET phoneNo = 'Updated PhoneNo' WHERE customerID in (SELECT L.customerID from login L WHERE L.accountNo = 'accountNo' and L.loggedIn = True);
```

Employee Mode-

- Create an account (Fill customerID, customerName, creditScore, street, area, city, pincode)
- Delete an account
- View All Accounts of every customer
- View Transactions History of every account
- Update personal data of employees

View Details (Input- employeeID)

```
SELECT * FROM employee  
WHERE employeeID = 'employeeID' UNION  
SELECT * FROM employeePhone where employeeID = 'employeeID';
```

;

SQL STATEMENTS-

- *View Account info*

```
select AccountNo,UserID,IFSC,branchArea,branchCity  
from account A, branch B, accountInBranch I  
where A.account = I.Account and I.IFSC = B.IFSC
```

- *View accounts from particular IFSC*

```
select AccountNo,UserID,IFSC,branchArea,branchCity  
from account,branch  
where IFSC ="PAYTM0123456"
```

- *View accounts by balance*

```
select AccountNo,UserID,IFSC,branchArea,branchCity
from account,branch
order by balance
```

- *View accounts with balance >1000000 and from Jaipur branch*

```
select AccountNo,UserID,IFSC,branchArea,branchCity
from account,branch
where balance>1000000
      and branchCity = "Jaipur"
```

- *View loan customers who took loan in 2015 and 2017*

```
Select C.customerID, C.customerName
from customer as C, borrows as B, loan as L
where L.loanID in ((select LoanID from Loan where where StartDate>1.1.2015 and
StartDate<31.12.2015)
union (select loanID from loan where StartDate>1.1.2016 and StartDate<31.12.2017)) and
B.loanID = L.loanID and C.customerID in (select B.customerID from borrows)
```

- *Take a loan of 100000 on the customerID 203333112 for a period of 3 years giving your bike as collateral at an interest rate of 12%*

```
INSERT INTO loan(loanID, interestRate, period, amount, EMI (derived), collateral)
VALUES(1001, 12, 3, 100000, 12000, 'bike_papers')
```

- *Transfer 2000 from accountNo 20312234 to accountNo 21122221 through UPI mode*

```
((UPDATE account set balance=(balance-2000) where accountNo = 20312234)
union
(UPDATE account set balance=(balance+2000) where accountNo = 21122221)
union
(INSERT INTO transaction(transactionID, accountNo., recAccNo., mode, amount, time_Stamp)
VALUES(1001, 20312234, 21122221, 'UPI', 2000, '11.02.2022 23:02')))
```

- *Repay the loan of 100000 where loanID was 1001*

```
DELETE FROM loan where loanID = 1001
```

- *Withdraw 2000 from accountNo 20312234*

```
UPDATE account set balance=(balance-2000) where accountNo = 20312234
```



- *Update from accountNo 20312234*

UPDATE account set password="#OONGA\_BOONGA#" where accountNo = 20312234 and password = "old\_password"

- *View Total balance of an accountNo*

SELECT balance from account where accountNo = 20312234

- *Avg balance of customerID 2*

SELECT avg(balance) from account where accountNo = 20312234

- *Find number of employees who have been a manager more than once*

Select employeeID, employeeName  
from employee as e, manages as m  
where count(e.employeeID = m.employeeID) > 1

- *average salary is greater than 42,000*

SELECT employeeName, avg\_salary  
from ( select employee, avg (salary) as avg\_salary  
from employee  
where avg\_salary > 42000 )

- *View Transaction history of accountNo 2033122*

SELECT \* FROM transaction WHERE accountNo = 2033122

- Select accountNo, balance,  
From account  
Group by balance

- Select customer.customer.Name, account.accountNo  
From customer  
CROSS Join account

## EMBEDDED QUERIES-

1)

select A.accountNo, O.IFSC, A.balance from account as A, accountOpening as O where A.accountType = %s and A.accountNo = O.accountNo and O.customerID = %s

2)

SELECT T.transactionID, T.accountNo, T.recAccNo, T.mode, T.amount, T.time\_Stamp from transaction T, accountOpening O WHERE O.customerID = %s and T.accountNo = O.accountNo

3)

SELECT \* from Loan L where L.loanID in (SELECT B.loanID from borrow B where B.customerID = %s)

4)

SELECT \* from fixeddeposit D where D.accountNo in (SELECT A.accountNo from accountopening A where A.customerID = %s)

5)

SELECT C.cardNo, C.Expiry, C.CVV, C.spendLimit, A.balance FROM Card C, Account A WHERE C.accountNo in (SELECT O.accountNo FROM accountopening O WHERE O.customerID = %s) and C.cardtype = %s and A.accountNo = C.accountNo

6)

SELECT \* from employee\_view\_cust

7)

SELECT \* from manager\_view\_cust

8)

SELECT \* from manager\_view\_emp

9)

INSERT into employee (employeeName, salary, street, area, city, pincode) values(%s, %s, %s, %s, %s, %s)

10)

INSERT into transaction(accountNo, recAccNo, amount) values (%s, %s, %s)

## VIEWS-

-- VIEW 1

create view employee\_view\_cust as

select c.customerName, c.creditScore, a.accountNo, a.balance, a.accountType  
from customer c, account a, accountOpening ao  
where ao.customerID = c.customerId and ao.accountNo = a.accountNo ;

-- VIEW 2

```
create view manager_view_cust as
select c.customerName, c.creditScore, c.customerID, a.accountNo, a.balance, a.accountType ,
b.IFSC
from customer c, account a, accountOpening ao, accountInBranch aib, branch b
where ao.customerID = c.customerID and ao.accountNo = a.accountNo and aib.accountNo =
a.accountNo and b.IFSC = aib.IFSC;
```

-- VIEW 3

```
create view manager_view_emp as
select e.employeeID, e.employeeName, e.salary
from employee e;
```

## INDEXES-

```
CREATE INDEX idx_transact ON transaction(accountNo);
CREATE INDEX idx_acc ON account(accountNo);
CREATE UNIQUE INDEX idx_card ON card(accountNo);
CREATE INDEX idx_accOpen ON accountopening(customerID);
CREATE INDEX idx_customer ON customer(customerID);
CREATE INDEX idx_loan ON loan(loanID);
CREATE INDEX idx_FD ON fixeddeposit(accountNo);
```

## GRANTS-

1)

```
create User 'manager'@'localhost' IDENTIFIED BY 'password';
Grant all on onlinebank.* to 'manager'@'localhost' with grant option;
flush privileges;
```

2)

```
create User 'customer'@'localhost' IDENTIFIED BY 'password';
Grant SELECT ON onlinebank.* to 'customer'@'localhost';
```

3)

```
create User 'admin'@'localhost' IDENTIFIED BY 'password';
Grant ALL ON onlinebank.* to 'customer'@'localhost' with grant option;
```

4)

```
create User 'employee'@'localhost' IDENTIFIED BY 'password';
GRANT ALL ON onlinebank.transaction,onlinebank.account,onlinebank.employee,
onlinebank.customer, onlinebank.accountopening to 'employee'@'localhost';
```