

Mini Banking System – Solutions

(PM Jat @ DAIICT)

Update Operations

1. Add two customers

```
insert into customer
values ('C906', 'Rajesh', '05,swagat parisar', 'Rajesh@gmail.com', 9
898989899, 'student'),
      ('C907', 'Ramesh', '06,satellite', 'Ramesh@gmail.com', 91919
19191, 'student');
```

2. Add two branches

```
insert into branch
values ('B906', '05jantarmantar', 'gandhinagar', 'gujarat', 'IN', 'IC
00165', 123412341, 'ICICINBBXXX'),
      ('B907', '05shubhnagar', 'gandhinagar', 'gujarat', 'IN', 'IC00
701', 127819841, 'ICICINCCYXX')
```

3. Add 5 accounts

```
insert into account
values ('ACC910', 'Current', 'B906', 10000),
      ('ACC911', 'Savings', 'B907', 15000),
      ('ACC912', 'Current', 'B906', 20000),
      ('ACC913', 'Fixed Deposit', 'B906', 25000),
      ('ACC914', 'Current', 'B907', 30000);
```

4. Add 10 transactions

```
insert into transaction
values ('T1501', 'ACC910', '2024-06-04', 'Groceries', 'Cr', 1000,
100000, 'UPI', 1),
      ('T1502', 'ACC912', '2024-06-04', 'Rent', 'Dr', 500, 100000,
, 'RETGS', 2),
      ('T1503', 'ACC912', '2024-06-04', 'Groceries', 'Cr', 1500,
300000, 'CHQ-ISSUE-RET', 3),
      ('T1504', 'ACC910', '2024-06-04', 'Rent', 'Dr', 700, 450000,
, 'RETGS', 4),
      ('T1505', 'ACC913', '2024-06-04', 'Groceries', 'Cr', 2000,
50000, 'UPI', 5),
      ('T1506', 'ACC911', '2024-06-04', 'Rent', 'Dr', 600, 59000
0, 'RETGS', 6),
      ('T1507', 'ACC912', '2024-06-04', 'Groceries', 'Cr', 1200
, 650000, 'RETGS', 7),
      ('T1508', 'ACC911', '2024-06-04', 'Rent', 'Dr', 800, 86906
1, 'RETGS', 8),
      ('T1509', 'ACC914', '2024-06-04', 'Groceries', 'Cr', 1700,
5600, 'UPI', 9),
      ('T1510', 'ACC914', '2024-06-04', 'Rent', 'Dr', 900, 98888,
'CHQ-ISSUE-RET', 10);
```

5. Add 5 rows in accountcustomer

```
insert into accountcustomer
values ('ACC910', 'C906'),
      ('ACC911', 'C906'),
      ('ACC912', 'C907'),
      ('ACC913', 'C907'),
      ('ACC914', 'C906'),
      ('ACC914', 'C907');
```

6. Change Phone number of a customer

```
update customer set phone = 9697969796 where cno = 'C906';
```

7. Change class of a Customer

```
update customer set cclass = 'corporate' where cno = 'C906';
```

8. Delete a transaction for transaction id (TID)

```
delete from transaction where tid = 'T906';
```

9. Delete all transaction for a given account number (ACCNO)

```
delete from transaction where accountno = 'ACC910';
```

10. REMOVE a customer from an joint account (CUSTID, ACCNO)

```
delete from accountcustomer  
where cno = 'C907' and accountno = 'ACC914';
```

11. Add an existing Customer to a given bank account (hence making the account join)
(CUSTID, ACCNO)

```
insert into accountcustomer (accountno, cno)  
values ('ACC910', 'C907');
```

12. Move an account to another branch (ACCNO, ToBID)

```
update account set bid= 'B907' where accountno = 'ACC910';
```

13. Change Description of a transaction (TID, NewDescription)

```
update transaction set description = 'Rent' where tid= 'T1510';
```

Querying Operations

Selection and Projection based queries

1. List all accounts at branch 'B001'

```
select * from account where bid = 'B001';
```

2. List "Premium" customers (Having CClass='Premium')

```
select * from customer where cclass = 'Premium';
```

3. Show transactions where account no ACC001 has deposited/withdrawn 10000 or more in cash.

```
select * from transaction  
where amount>=10000 and accountno='ACC001';
```

4. List all salary accounts at a specific branch (BID)

```
select * from account where type='Salary' and bid = 'B001';
```

5. List all accounts where the balance is more than 40,000 /-

```
select * from account where balance > 40000;
```

6. Status of an issued check (checkNo)

```
select tr_type from transaction where cheqno = 3;
```

7. List all dates, balance on which a given account number has the negative balance

```
select date, accountno from dailybalance where cb < 0;
```

8. List all neft transactions on an account number (ACCNO)

```
select * from transaction  
where tr_type = 'NEFT' and accountno = 'ACC001';
```

9. List customer name, address and contact number

```
select name, address, phone from customer;
```

Join based queries

1. List Details of all accounts of a given Customer (CNO)

```
select * from account  
natural join accountcustomer where cno = 'C001';
```

2. List Idle accounts that had no transactions in the last three months

```
select * from account  
where accountno NOT IN  
    (select accountno  
     from transaction  
     where ts between current_date - INTERVAL '3 MONTH'  
           and current_date);
```

3. Show transactions where deposited/withdrawn 10000 or more in any of the accounts of the customer 101.

```
select t.* from transaction as t
natural join accountcustomer as ac
where cno = 'C001' and amount >= 10000;
```

4. List all JOINT-ACCOUNTS in branch 'B001'.

```
select accountno from account natural join
(select accountno, count(cno) as cnt from accountcustomer
group by accountno
having count(cno)>1)
where bid='B001';
```

5. List all account numbers that are having check bounces more 3 in last 6 months

```
select accountno, count(cheqno) from transaction
where tr_type = 'CHQ-ISSUE-RET'
and ts between current_date - INTERVAL '6 MONTH'
and current_date
group by accountno
having count(cheqno) > 3;
```

6. List customer name and address who do not have any account

```
select c.name, c.address
from customer c
left join accountcustomer ac on c.cno = ac.cno
where ac.accountno is null;
```

Or

```
select name, address from customer
where cno NOT IN
    (select c.cno from customer as c
     natural join accountcustomer
     group by cno
     having count(accountno) >= 1);
```

7. List customers who have never used UPI

```
select cno from customer
where cno NOT IN
    (select ac.cno from transaction as t
     natural join accountcustomer as ac
     where tr_type = 'UPI');
```

8. List customer who have done atleast one RETGS

```
select distinct(ac.cno) from transaction
natural join accountcustomer as ac
where tr_type = 'RETGS';
```

9. List all (accountNo, dates, balance) on which a given customer number has the negative balance.

Note : Following customers are having negative balance.

[C074, C079, C015, C011, C076, C030, C042, C047]

```
select db.accountno, db.date, db.cb from accountcustomer ac
join dailybalance db on ac.accountno = db.accountno
where ac.cno = 'C074' and db.cb < 0;
```

10. List customer name,email that are having account at branch 'B001'

```
select c.name, c.email from customer as c
natural join accountcustomer
natural join account
where bid='B001';
```

Aggregation based queries

1. Give branch wise count of Accounts

```
select bid, count(accountno) from account
group by bid;
```

2. List Cities in which we have more than one branches

```
select city, count(bid) from branch
group by city
having count(bid) > 1;
```

3. List branches where we have more than 5 account

```
select bid, count(accountno) as account_count from account
group by bid
having count(accountno) > 5;
```

4. Give top 10 accounts in terms of their current balance

```
select accountno, balance from account
order by balance desc
limit 10;
```


5. Find out the daily minimum balance.

```
select Date, min(cb) as minimum_balance
from dailybalance
group by Date;
```

6. Find out the average monthly balance of accounts for account 'ACC002' for a given year (2024)

```
select extract(month from Date) as month,
       trunc(avg(cb)) as avg_monthly_balance
from dailybalance
where accountno = 'ACC002' and extract(year from Date) = 2024
group by extract(month from Date);
```

7. Give the top 10 accounts in terms of their monthly average balance for a given year

```
select accountno, avg(cb) as curb
from dailybalance
where Date between '2023-01-01' and '2023-12-31'
group by accountno
order by avg(cb) desc
limit 10;
```

8. Give branch wise count of Customers

```
select bid, count(cno) from account
natural join accountcustomer
group by bid;
```

9. List cities where we have more than 50 accounts

```
select city, count(accountno)
from branch
natural join account
group by city
having count(accountno) > 50;
```

10. List customers whose cheque bounced are more than once in the last six months

```
select ac.cno, count(t.tid) as bounced_count
from transaction t
inner join accountcustomer ac on t.accountno = ac.accountno
where t.tr_type = 'CHQ-ISSUE-RET'
    and t.ts between current_date - INTERVAL '6 MONTH'
    and current_date
group by ac.cno
having count(t.tid) > 1;
```

11. Compute the current balance of account no 'ACC001'. Assume that we have Opening balance only as on 01-01-2021 and say that is 25000.

```
select 25000 + sum(case
    when crdr = 'Cr' then 1 else -1
end * amount) as current_balance
from transaction
where accountno = 'ACC001';
```

12. Compute the closing balance of account no 'ACC001' as on 31-03-2024. Assume that we have an opening balance as on 01-01-2021 is 25000.

```
select 25000 + sum(case
    when crdr = 'Cr' then 1 else -1
end * amount) as closing_balance
from transaction
where accountno = 'ACC001' and ts <= date '2024-03-31';
```

13. Enlist city wise count of customers.(city, customerCount)

```
select city, count(distinct(cno)) from account
natural join accountcustomer
natural join branch
group by city;
```

14. List customers having more than two accounts (Including joint accounts)

```
select cno, count(*) as account_count from accountcustomer
group by cno
having count(*) > 2;
```

15. Compute Interest for all “Saving Account” for the quarter of 01-01-2024 to 31-03-2024

```
select accountno,
    trunc(sum(cb)*4.0*(1.0/(365*100.00)), 2) as interest
from dailybalance natural join account
where "type" = 'Savings' and "date" >= '2024-01-01'
    and "date" <= '2024-03-31'
group by accountno;
```