Table Definitions:

1. BANK\_CUSTOMER - Details of Customers of the Bank

2. BANK\_CUSTOMER\_EXPORT - Details of Customers of the Bank - to be used only when explicitly asked.

3. Bank\_Account\_Details - Account Details of the customers along with ADD on cards. One customer can have multiple details for Savings deposits, Recurring deposits, Credit Cards and Add on credit cards.

4. Bank\_Account\_Relationship\_Details - Details of secondary accounts linked to primary accounts.

5. BANK\_ACCOUNT\_TRANSACTION - Details of the transactions.

6. BANK\_CUSTOMER\_MESSAGES - Details of Messages sent to customers after a transaction takes place.

7. BANK\_INTEREST\_RATE - Current interest rates for savings, RD and other accounts.

8. Bank\_Holidays - Details of Bank Holidays.

CREATE TABLE BANK\_CUSTOMER

( customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(20),

Address VARCHAR(100),

state\_code VARCHAR(3) ,

Telephone VARCHAR(10) );

drop table BANK\_CUSTOMER;

INSERT INTO BANK\_CUSTOMER VALUES (123001,"Oliver", "225-5,Emeryville", "CA" , "1897614500");

INSERT INTO BANK\_CUSTOMER VALUES (123002,"George", "194-6,New brighton","MN" , "1897617000");

INSERT INTO BANK\_CUSTOMER VALUES (123003,"Harry", "2909-5,walnut creek","CA" , "1897617866");

INSERT INTO BANK\_CUSTOMER VALUES (123004,"Jack", "229-5, Concord",null,null);

INSERT INTO BANK\_CUSTOMER VALUES (123005,"Jacob", "325-7,Mission Dist","SFO", "1897637000");

INSERT INTO BANK\_CUSTOMER VALUES (123006,"Noah", "275-9,saint-paul" , "MN" , "1897613200");

INSERT INTO BANK\_CUSTOMER VALUES (123007,"Charlie","125-1,Richfield", "MN" , "1897617666");

INSERT INTO BANK\_CUSTOMER VALUES (123008,"Robin","3005-1,Heathrow",null,null);

SELECT\*from BANK\_CUSTOMER;

CREATE TABLE BANK\_CUSTOMER\_EXPORT

(

customer\_id CHAR(10)PRIMARY KEY,

customer\_name CHAR(20),

Address CHAR(20),

state\_code CHAR(3) ,

Telephone CHAR(10));

INSERT INTO BANK\_CUSTOMER\_EXPORT VALUES ("123001 ","Oliver", "225-5, Emeryville", "CA" , "1897614500") ;

INSERT INTO BANK\_CUSTOMER\_EXPORT VALUES ("123002 ","George", "194-6,New brighton","MN" , "189761700");

CREATE TABLE Bank\_Account\_Details

(Customer\_id INT,

Account\_Number VARCHAR(19) PRIMARY KEY,

Account\_type VARCHAR(25) ,

Balance\_amount INT,

Account\_status VARCHAR(10),

Relationship\_type varchar(1)) ;

ALTER TABLE Bank\_Account\_Details ADD FOREIGN KEY (Customer\_id) REFERENCES bank\_customer(Customer\_id);

INSERT INTO Bank\_Account\_Details VALUES (123001, "4000-1956-3456", "SAVINGS" , 200000 ,"ACTIVE","P");

INSERT INTO Bank\_Account\_Details VALUES (123001, "5000-1700-3456", "RECURRING DEPOSITS" ,9400000 ,"ACTIVE","S");

INSERT INTO Bank\_Account\_Details VALUES (123002, "4000-1956-2001", "SAVINGS", 400000 ,"ACTIVE","P");

INSERT INTO Bank\_Account\_Details VALUES (123002, "5000-1700-5001", "RECURRING DEPOSITS" ,7500000 ,"ACTIVE","S");

INSERT INTO Bank\_Account\_Details VALUES (123003, "4000-1956-2900", "SAVINGS" ,750000,"INACTIVE","P");

INSERT INTO Bank\_Account\_Details VALUES (123004, "5000-1700-6091", "RECURRING DEPOSITS" ,7500000 ,"ACTIVE","S");

INSERT INTO Bank\_Account\_Details VALUES (123004, "4000-1956-3401", "SAVINGS" , 655000 ,"ACTIVE","P");

INSERT INTO Bank\_Account\_Details VALUES (123005, "4000-1956-5102", "SAVINGS" , 300000 ,"ACTIVE","P");

INSERT INTO Bank\_Account\_Details VALUES (123006, "4000-1956-5698", "SAVINGS" , 455000 ,"ACTIVE" ,"P");

INSERT INTO Bank\_Account\_Details VALUES (123007, "5000-1700-9800", "SAVINGS" , 355000 ,"ACTIVE" ,"P");

INSERT INTO Bank\_Account\_Details VALUES (123007, "4000-1956-9977", "RECURRING DEPOSITS" , 7025000,"ACTIVE" ,"S");

INSERT INTO Bank\_Account\_Details VALUES (123007, "9000-1700-7777-4321", "Credit Card",null,null,null);

INSERT INTO Bank\_Account\_Details VALUES (123007, '5900-1900-9877-5543', "Add-on Credit Card" , 0 ,"ACTIVE", "S");

INSERT INTO Bank\_Account\_Details VALUES (123008, "5000-1700-7755", "SAVINGS",null,null,null);

INSERT INTO Bank\_Account\_Details VALUES (123006, '5800-1700-9800-7755', "Credit Card" ,0,null,null);

INSERT INTO Bank\_Account\_Details VALUES (123006, '5890-1970-7706-8912', "Add-on Credit Card" ,0,null,null);

INSERT INTO Bank\_Account\_Details VALUES (123004,'5000-1700-7791','RECURRING DEPOSITS',40000,'ACTIVE','S');

CREATE TABLE Bank\_Account\_Relationship\_Details

( Customer\_id INT ,

Account\_Number VARCHAR(19) PRIMARY KEY ,

Account\_type VARCHAR(25),

Linking\_Account\_Number VARCHAR(19)

);

ALTER TABLE Bank\_Account\_Relationship\_Details ADD FOREIGN KEY (Customer\_id) REFERENCES bank\_customer(Customer\_id);

ALTER TABLE Bank\_Account\_Relationship\_Details ADD FOREIGN KEY (Linking\_Account\_Number) REFERENCES bank\_account\_details(Account\_Number);

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123001, "4000-1956-3456", "SAVINGS" , NULL);

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123001, "5000-1700-3456", "RECURRING DEPOSITS" , "4000-1956-3456");

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123002, "4000-1956-2001", "SAVINGS" , NULL );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123002, "5000-1700-5001", "RECURRING DEPOSITS" , "4000-1956-2001" );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123003, "4000-1956-2900", "SAVINGS" , NULL );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123004, "5000-1700-6091", "RECURRING DEPOSITS" , "4000-1956-2900" );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123004, "5000-1700-7791", "RECURRING DEPOSITS" , "4000-1956-2900" );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123007, "5000-1700-9800", "SAVINGS" , NULL);

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (123007, "4000-1956-9977", "RECURRING DEPOSITS" , "5000-1700-9800" );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (NULL, "9000-1700-7777-4321", "Credit Card" , "5000-1700-9800" );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (NULL, '5900-1900-9877-5543', 'Add-on Credit Card', '9000-1700-7777-4321' );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (NULL, '5800-1700-9800-7755', 'Credit Card', '4000-1956-5698' );

INSERT INTO Bank\_Account\_Relationship\_Details VALUES (NULL, '5890-1970-7706-8912', 'Add-on Credit Card', '5800-1700-9800-7755' );

CREATE TABLE BANK\_ACCOUNT\_TRANSACTION

(

Account\_Number VARCHAR(19),

Transaction\_amount Decimal(18,2) ,

Transcation\_channel VARCHAR(18) ,

Province varchar(3) ,

Transaction\_Date Date

) ;

ALTER TABLE Bank\_Account\_Transaction ADD FOREIGN KEY (Account\_number) REFERENCES Bank\_Account\_Details(Account\_Number);

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-3456", -2000, "ATM withdrawl" , "CA", "2020-01-13");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-2001", -4000, "POS-Walmart" , "MN", "2020-02-14");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-2001", -1600, "UPI transfer" , "MN", "2020-01-19");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-2001", -6000, "Bankers cheque", "CA", "2020-03-23");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-2001", -3000, "Net banking" , "CA", "2020-04-24");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-2001", 23000, "cheque deposit", "MN", "2020-03-15");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "5000-1700-6091", 40000, "ECS transfer" , "NY", "2020-02-19");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "5000-1700-7791", 40000, "ECS transfer" , "NY", "2020-02-19");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-3401", 8000, "Cash Deposit" , "NY", "2020-01-19");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-5102", -6500, "ATM withdrawal" , "NY", "2020-03-14");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-5698", -9000, "Cash Deposit" , "NY", "2020-03-27");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "4000-1956-9977", 50000, "ECS transfer" , "NY", "2020-01-16");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "9000-1700-7777-4321", -5000, "POS-Walmart", "NY", "2020-02-17");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "9000-1700-7777-4321", -8000, "Shopping Cart", "MN", "2020-03-13");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "9000-1700-7777-4321", -2500, "Shopping Cart", "MN", "2020-04-21");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( "5800-1700-9800-7755", -9000, "POS-Walmart","MN", "2020-04-13");

INSERT INTO BANK\_ACCOUNT\_TRANSACTION VALUES ( '5890-1970-7706-8912', -11000, "Shopping Cart" , "NY" , "2020-03-12") ;

INSERT INTO bank\_account\_transaction(Account\_Number, Transaction\_amount, Transcation\_channel, Province, Transaction\_Date) VALUES

('4000-1956-9977' , 10000.00 ,'ECS transfer', 'MN' , '2020-02-16' ) ;

INSERT INTO bank\_account\_transaction(Account\_Number, Transaction\_amount, Transcation\_channel, Province, Transaction\_Date) VALUES

('4000-1956-9977' , 40000.00 ,'ECS transfer', 'MN' , '2020-03-18' ) ;

INSERT INTO bank\_account\_transaction(Account\_Number, Transaction\_amount, Transcation\_channel, Province, Transaction\_Date) VALUES

('4000-1956-9977' , 60000.00 ,'ECS transfer', 'MN' , '2020-04-18' ) ;

INSERT INTO bank\_account\_transaction(Account\_Number, Transaction\_amount, Transcation\_channel, Province, Transaction\_Date) VALUES

('4000-1956-9977' , 20000.00 ,'ECS transfer', 'MN' , '2020-03-20' ) ;

CREATE TABLE BANK\_CUSTOMER\_MESSAGES

(

Event VARCHAR(24),

Customer\_message VARCHAR(75),

Notice\_delivery\_mode VARCHAR(15)) ;

INSERT INTO BANK\_CUSTOMER\_MESSAGES VALUES ( "Adhoc", "All Banks are closed due to announcement of National strike", "mobile" ) ;

INSERT INTO BANK\_CUSTOMER\_MESSAGES VALUES ( "Transaction Limit", "Only limited withdrawals per card are allowed from ATM machines", "mobile" );

CREATE TABLE BANK\_INTEREST\_RATE(

account\_type varchar(24)PRIMARY KEY,

interest\_rate decimal(4,2),

month varchar(2),

year varchar(4)

);

INSERT INTO BANK\_INTEREST\_RATE VALUES ( "SAVINGS" , 0.04 , '02' , '2020' );

INSERT INTO BANK\_INTEREST\_RATE VALUES ( "RECURRING DEPOSITS" , 0.07, '02' , '2020' );

INSERT INTO BANK\_INTEREST\_RATE VALUES ( "PRIVILEGED\_INTEREST\_RATE" , 0.08 , '02' , '2020' );

Create table Bank\_Holidays (

Holiday date PRIMARY KEY,

Start\_time datetime ,

End\_time timestamp);

Insert into bank\_holidays values ( '2020-05-20','2020-05-20','2020-05-20' ) ;

Insert into bank\_holidays values( '2020-03-13' ,'2020-03-13' ,'2020-03-13' ) ;

############################# Questions ######################################

Question 1:

Print customer Id, customer name and average account\_balance maintained by each customer for all

of his/her accounts in the bank.

**CODE:**

SELECT

BC.customer\_id, BC.customer\_name,

AVG(BAD.Balance\_amount) AS average\_account\_balance

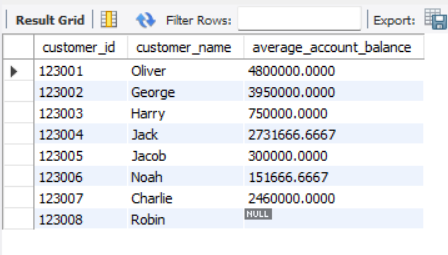
FROM BANK\_CUSTOMER BC

LEFT JOIN

Bank\_Account\_Details BAD ON BC.customer\_id = BAD.Customer\_id

GROUP BY BC.customer\_id, BC.customer\_name;

**OUTPUT:**

****

Question 2:

Print customer\_id , account\_number and balance\_amount for all the accounts.

for account\_type = "Credit Card" apply the condition that if balance\_amount is nil then assign transaction\_amount

**CODE:**

SELECT

BAD.Customer\_id, BAD.Account\_Number,

CASE

WHEN BAD.Account\_type = 'Credit Card' AND BAD.Balance\_amount IS NULL THEN BAT.Transaction\_amount

ELSE BAD.Balance\_amount

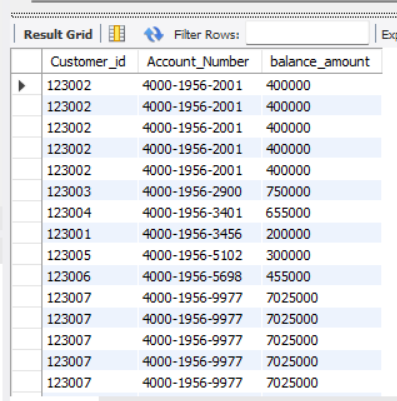
END AS balance\_amount

FROM Bank\_Account\_Details BAD

LEFT JOIN

BANK\_ACCOUNT\_TRANSACTION BAT ON BAD.Account\_Number = BAT.Account\_Number;

**OUTPUT:**



Question 3:

Print account\_number and balance\_amount , transaction\_amount,Transaction\_Date from Bank\_Account\_Details and bank\_account\_transaction for all the transactions occurred during march,2020 and april, 2020

**CODE:**

**SELECT**

**BAD.Account\_Number,**

**BAD.Balance\_amount,**

**BAT.Transaction\_amount,**

**BAT.Transaction\_Date**

**FROM**

**Bank\_Account\_Details BAD**

**JOIN**

**BANK\_ACCOUNT\_TRANSACTION BAT**

**ON**

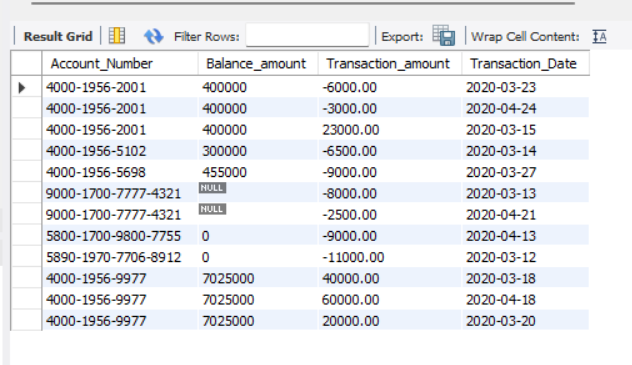
**BAD.Account\_Number = BAT.Account\_Number**

**WHERE**

**YEAR(BAT.Transaction\_Date) = 2020**

**AND MONTH(BAT.Transaction\_Date) IN (3, 4);**

**OUTPUT:**



Question 4:

Print all the customer ids, account number, balance\_amount, transaction\_amount , Transaction\_Date

from bank\_customer, Bank\_Account\_Details and bank\_account\_transaction tables where excluding

all of their transactions in march, 2020 month

**CODE:**

**SELECT**

**BC.customer\_id, BAD.Account\_Number, BAD.Balance\_amount, BAT.Transaction\_amount,**

**BAT.Transaction\_Date**

**FROM BANK\_CUSTOMER BC JOIN Bank\_Account\_Details BAD**

**ON BC.customer\_id = BAD.Customer\_id**

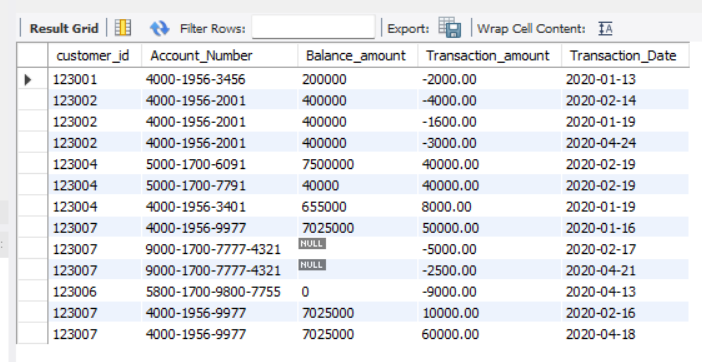
**JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

**ON BAD.Account\_Number = BAT.Account\_Number**

**WHERE YEAR(BAT.Transaction\_Date) != 2020**

**OR (YEAR(BAT.Transaction\_Date) = 2020 AND MONTH(BAT.Transaction\_Date) != 3);**

**OUTPUT:**



Question 5:

Print the customer ids, account\_number, balance\_amount,transaction\_amount ,transaction\_date who did transactions during the first quarter. Do not display the accounts if they have not done any transactions in the first quarter.

**CODE:**

**SELECT**

**BC.customer\_id, BAD.Account\_Number, BAD.Balance\_amount, BAT.Transaction\_amount,**

**BAT.Transaction\_Date**

**FROM**

**BANK\_CUSTOMER BC JOIN Bank\_Account\_Details BAD**

**ON**

**BC.customer\_id = BAD.Customer\_id**

**LEFT JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

**ON**

**BAD.Account\_Number = BAT.Account\_Number**

**AND YEAR(BAT.Transaction\_Date) = 2020**

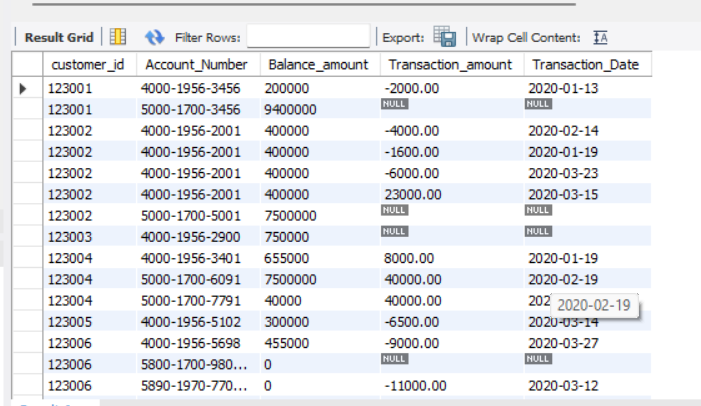
**AND MONTH(BAT.Transaction\_Date) BETWEEN 1 AND 3**

**WHERE**

**(YEAR(BAT.Transaction\_Date) = 2020 AND MONTH(BAT.Transaction\_Date) BETWEEN 1 AND 3)**

**OR BAT.Transaction\_Date IS NULL;**

**OUTPUT:**



Question 6:

Print account\_number, Event and Customer\_message from BANK\_CUSTOMER\_MESSAGES and Bank\_Account\_Details to

display an “Adhoc" Event for all customers who have “SAVINGS" account\_type account.

**\*\* We cannot display any result due to no common column in both tables\*\***

Question 7:

Print all the Customer\_ids, Account\_Number, Account\_type, and display deducted balance\_amount by

subtracting only negative transaction\_amounts for Relationship\_type =

P ( P - means Primary , S - means Secondary ) .

**CODE:**

**SELECT**

**BAD.Customer\_id, BAD.Account\_Number, BAD.Account\_type,**

**BAD.Balance\_amount - SUM(CASE WHEN BAT.Transaction\_amount < 0 THEN BAT.Transaction\_amount ELSE 0 END) AS deducted\_balance**

**FROM Bank\_Account\_Details BAD**

**LEFT JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

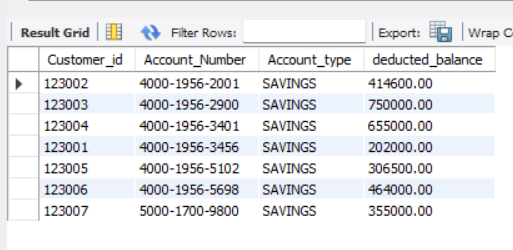
**ON BAD.Account\_Number = BAT.Account\_Number**

**WHERE BAD.Relationship\_type = 'P'**

**GROUP BY**

**BAD.Customer\_id, BAD.Account\_Number, BAD.Account\_type, BAD.Balance\_amount;**

**OUTPUT:**



Question 8:

Display records of All Accounts , their Account\_types, the balance amount. Along with first step, Display other columns with corresponding linking account number, account types

**CODE:**

**SELECT**

**BAD.Account\_Number AS primary\_account\_number,**

**BAD.Account\_type AS primary\_account\_type,**

**BAD.Balance\_amount AS primary\_balance\_amount,**

**BARD.Linking\_Account\_Number,**

**BAD\_REL.Account\_type AS linked\_account\_type**

**FROM**

**Bank\_Account\_Details BAD**

**LEFT JOIN**

**Bank\_Account\_Relationship\_Details BARD**

**ON**

**BAD.Account\_Number = BARD.Account\_Number**

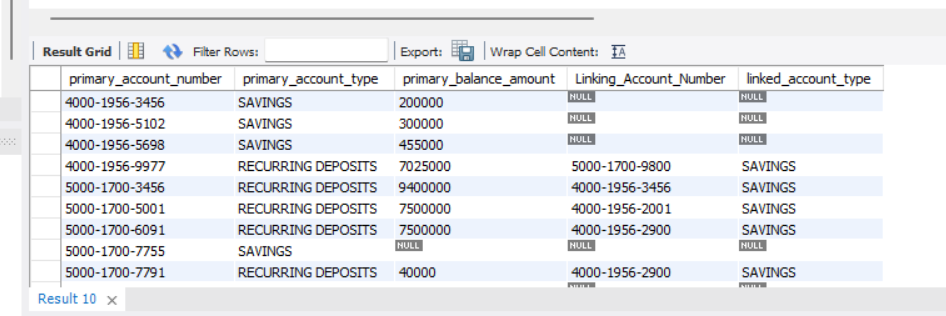
**LEFT JOIN**

**Bank\_Account\_Details BAD\_REL**

**ON**

**BARD.Linking\_Account\_Number = BAD\_REL.Account\_Number;**

**OUTPUT:**



Question 9:

Display records of All Accounts , their Account\_types, the balance amount.

Along with first step, Display other columns with corresponding linking account number, account types

After retrieving all records of accounts and their linked accounts, display the transaction amount of accounts appeared in another column.

**CODE:**

**SELECT**

**BAD.Account\_Number AS primary\_account\_number,**

**BAD.Account\_type AS primary\_account\_type,**

**BAD.Balance\_amount AS primary\_balance\_amount,**

**BARD.Linking\_Account\_Number,**

**BAD\_REL.Account\_type AS linked\_account\_type,**

**BAT.Transaction\_amount AS transaction\_amount**

**FROM**

**Bank\_Account\_Details BAD LEFT JOIN Bank\_Account\_Relationship\_Details BARD**

**ON**

**BAD.Account\_Number = BARD.Account\_Number**

**LEFT JOIN**

**Bank\_Account\_Details BAD\_REL**

**ON**

**BARD.Linking\_Account\_Number = BAD\_REL.Account\_Number**

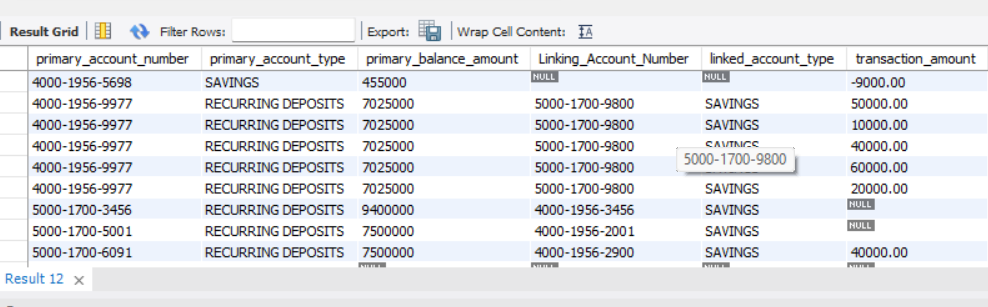
**LEFT JOIN**

**BANK\_ACCOUNT\_TRANSACTION BAT**

**ON**

**BAD.Account\_Number = BAT.Account\_Number;**

**OUTPUT:**



Question 10:

Display all account holders from Bank\_Accounts\_Details table who have “Add-on Credit Cards" and “Credit cards"

**CODE:**

**SELECT**

**Customer\_id, Account\_Number, Account\_type**

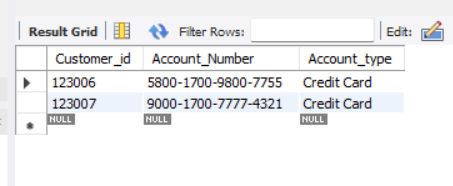
**FROM**

**Bank\_Account\_Details**

**WHERE**

**Account\_type IN ('Add-on Credit Cards', 'Credit Card');**

**OUTPUT:**



Question 11:

Display records of “SAVINGS” accounts linked with “Credit card" account\_type and its creditaggregate sum of transaction amount.

Ref: Use bank\_Account\_Details for Credit card types

# Check transaction\_amount in bank\_account\_transaction.

**CODE:**

**SELECT**

**BAD.Account\_Number AS savings\_account\_number,**

**SUM(BAT.Transaction\_amount) AS credit\_aggregate\_sum\_of\_transaction\_amount**

**FROM Bank\_Account\_Details BAD**

**JOIN Bank\_Account\_Relationship\_Details BARD**

**ON BAD.Account\_Number = BARD.Linking\_Account\_Number**

**JOIN Bank\_Account\_Details BAD\_Credit**

**ON BARD.Account\_Number = BAD\_Credit.Account\_Number**

**JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

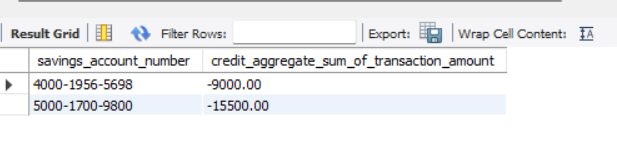
**ON BAD\_Credit.Account\_Number = BAT.Account\_Number**

**WHERE BAD.Account\_type = 'SAVINGS'**

**AND BAD\_Credit.Account\_type = 'Credit Card'**

**GROUP BY BAD.Account\_Number;**

**OUTPUT:**



Question 12:

Display all type of “Credit cards” accounts including linked “Add-on Credit Cards"

type accounts with their respective aggregate sum of transaction amount.

Ref: Check Bank\_Account\_Details\_table for all types of credit card.

# Check transaction\_amount in bank\_account\_transaction.

**CODE:**

**SELECT**

**BAD.Account\_type AS Credit\_Card\_Type,**

**SUM(BAT.Transaction\_amount) AS Credit\_Card\_Aggregate\_Sum\_of\_Transaction\_Amount**

**FROM Bank\_Account\_Details BAD**

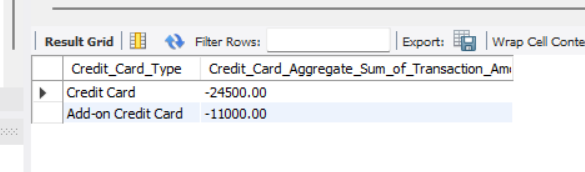
**LEFT JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

**ON BAD.Account\_Number = BAT.Account\_Number**

**WHERE BAD.Account\_type IN ('Credit Card', 'Add-on Credit Card')**

**GROUP BY BAD.Account\_type;**

**OUTPUT:**



Question 13:

Display “SAVINGS” accounts and their corresponding aggregate sum of transaction amount

of all recurring deposits

**CODE:**

**SELECT**

**BAD.Account\_Number AS Savings\_Account\_Number,**

**SUM(BAT.Transaction\_amount) AS Aggregate\_Sum\_of\_Transaction\_Amount**

**FROM Bank\_Account\_Details BAD**

**JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

**ON BAD.Account\_Number = BAT.Account\_Number**

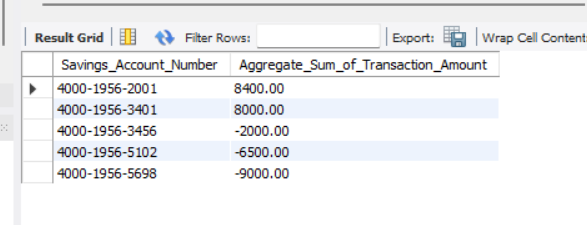
**WHERE**

**BAD.Account\_type = 'SAVINGS'**

**AND BAD.Relationship\_type = 'P'**

**GROUP BY BAD.Account\_Number;**

**OUTPUT:**



Question 14:

Display every accounts total no of transactions for every year and each month.

**CODE:**

**SELECT**

**BAD.Account\_Number,**

**YEAR(BAT.Transaction\_Date) AS Transaction\_Year,**

**MONTH(BAT.Transaction\_Date) AS Transaction\_Month,**

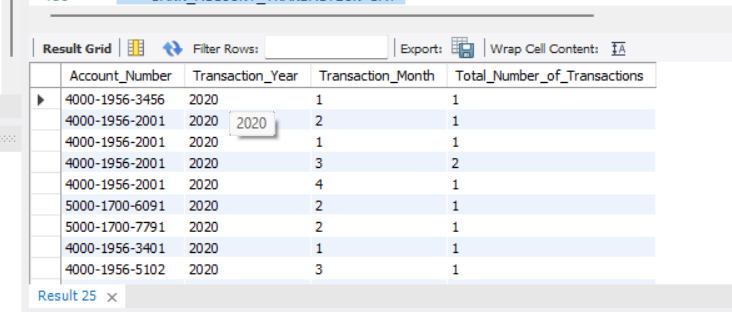
**COUNT(\*) AS Total\_Number\_of\_Transactions**

**FROM Bank\_Account\_Details BAD**

**JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

**ON BAD.Account\_Number = BAT.Account\_Number**

**GROUP BY BAD.Account\_Number, Transaction\_Year, Transaction\_Month;**

**OUTPUT:** 

Question 15:

Compare the aggregate sum transaction amount of Feb2020 month versus Jan2020 Month for each account number.

Display account\_number, transaction\_amount ,

sum of feb month transaction amount ,

sum of Jan month transaction amount ,

**CODE:**

**SELECT**

**BAT.Account\_Number,**

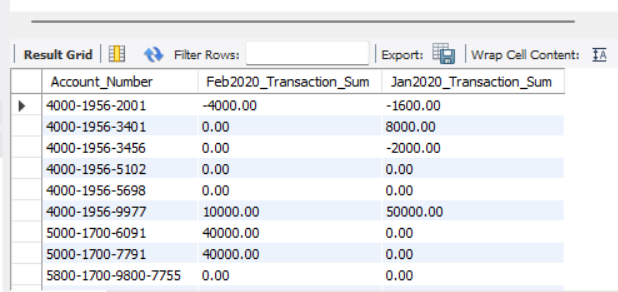
**SUM(CASE WHEN MONTH(BAT.Transaction\_Date) = 2 AND YEAR(BAT.Transaction\_Date) = 2020 THEN BAT.Transaction\_amount ELSE 0 END) AS Feb2020\_Transaction\_Sum,**

**SUM(CASE WHEN MONTH(BAT.Transaction\_Date) = 1 AND YEAR(BAT.Transaction\_Date) = 2020 THEN BAT.Transaction\_amount ELSE 0 END) AS Jan2020\_Transaction\_Sum**

**FROM BANK\_ACCOUNT\_TRANSACTION BAT**

**GROUP BY BAT.Account\_Number;**

**OUTPUT:**



Question 16:

Display the customer names who have all three account types -

savings, recurring and credit card account holders.

**CODE:**

**SELECT**

**DISTINCT c.Customer\_name**

**FROM Bank\_Account\_Details b**

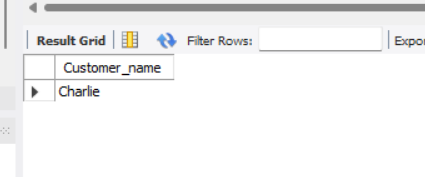
**JOIN Bank\_Customer c ON b.Customer\_id = c.Customer\_id**

**WHERE b.Account\_type IN ('SAVINGS', 'RECURRING DEPOSITS', 'Credit card')**

**GROUP BY c.Customer\_id**

**HAVING COUNT(DISTINCT b.Account\_type) = 3;**

**OUTPUT:**



Question 17:

Display savings accounts and its corresponding Recurring deposits transactions that are occuring more than 4 times.

**CODE:**

**SELECT**

**s.Account\_Number AS Savings\_Account\_Number,**

**r.Account\_Number AS Recurring\_Account\_Number,**

**COUNT(t.Transaction\_date) AS Transaction\_Count**

**FROM Bank\_Account\_Details s**

**JOIN Bank\_Account\_Relationship\_Details rel**

**ON s.Account\_Number = rel.Linking\_Account\_Number**

**JOIN Bank\_Account\_Details r**

**ON rel.Account\_Number = r.Account\_Number**

**JOIN Bank\_Account\_Transaction t**

**ON r.Account\_Number = t.Account\_Number**

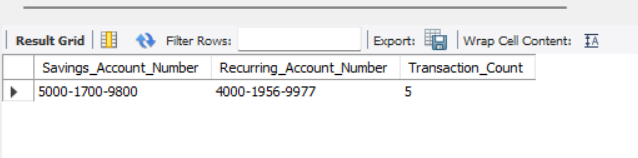
**WHERE s.Account\_type = 'SAVINGS'**

**AND r.Account\_type = 'RECURRING DEPOSITS'**

**GROUP BY s.Account\_Number, r.Account\_Number**

**HAVING COUNT(t.Transaction\_date) > 4;**

**OUTPUT:**



Question 18:

Find the no. of transactions of credit cards including add-on Credit Cards

**CODE:**

**SELECT**

**BAD.Account\_type AS Credit\_Card\_Type,**

**COUNT(DISTINCT BAT.Account\_Number) AS Total\_Number\_of\_Transactions**

**FROM Bank\_Account\_Details BAD**

**LEFT JOIN BANK\_ACCOUNT\_TRANSACTION BAT**

**ON BAD.Account\_Number = BAT.Account\_Number**

**WHERE BAD.Account\_type IN ('Credit Card', 'Add-on Credit Card')**

**GROUP BY BAD.Account\_type;**

**OUTPUT:**

