FinSight Database Optimization

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!apt-get install sqlite3
!pip install pandas
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  sqlite3-doc
The following NEW packages will be installed:
  sqlite3
0 upgraded, 1 newly installed, 0 to remove and 35 not upgraded.
Need to get 769 kB of archives.
After this operation, 1,874 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 sqlite3 amd64 3.37.2-2ubuntu0.5 [769 kB]
Fetched 769 kB in 1s (1,122 kB/s)
Selecting previously unselected package sqlite3.
(Reading database ... 126374 files and directories currently installed.)
Preparing to unpack .../sqlite3_3.37.2-2ubuntu0.5_amd64.deb ...
Unpacking sqlite3 (3.37.2-2ubuntu0.5) ...
Setting up sqlite3 (3.37.2-2ubuntu0.5) ...
Processing triggers for man-db (2.10.2-1) ...
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.
```

```
import sqlite3
import pandas as pd
import os
```

```
conn=sqlite3.connect('my_database.db')
```

```
#Define the connect path for the database
db_path='/content/drive/MyDrive/my_database.db'
```

```
db_folder='/content/drive/MyDrive/sqlite3 folder'
os.makedirs(db_folder,exist_ok=True) #ensure the folder exist
db_path=os.path.join(db_folder,'my_database.db')

conn=sqlite3.connect(db_path)
cursor=conn.cursor()

print('Database connected at:',db_path)
```

```
Database connected at: /content/drive/MyDrive/sqlite3 folder/my_database.db
```

CUSTOMER TABLE

```
cursor.execute('''
CREATE TABLE CUSTOMER ( customer_id INT ,
                customer_name VARCHAR(20),
                Address     VARCHAR(20),
                state_code  VARCHAR(3) ,
                Telephone   VARCHAR(10));

''')
conn.commit()
```

```
def insert_CUSTOMER (customer_id,customer_name,Address,state_code,Telephone):
  cursor.execute('INSERT INTO CUSTOMER(customer_id,customer_name,Address,state_code,Telephone) VALUES(?,?,?,?,?)',(customer
  conn.commit()

#INSERT
insert_CUSTOMER(123001,"Oliver", "225-5, Emeryville", "CA" , "1897614500");
insert_CUSTOMER(123002,"George", "194-6,New brighton","MN" , "1897617000");
insert_CUSTOMER(123003,"Harry", "2909-5,walnut creek","CA" , "1897617866");
insert_CUSTOMER(123004,"Jack", "229-5, Concord",    "CA" , "1897627999");
```

```
insert_CUSTOMER(123005,"Jacob", "325-7, Mission Dist","SFO", "1897637000");
insert_CUSTOMER(123006,"Noah", "275-9, saint-paul" ,  "MN" , "1897613200");
insert_CUSTOMER(123007,"Charlie","125-1,Richfield",   "MN" , "1897617666");
insert_CUSTOMER(123008,"Robin","3005-1,Heathrow",   "NY" , "1897614000");
```

```
!pip install tabulate
```

```
Requirement already satisfied: tabulate in /usr/local/lib/python3.12/dist-packages (0.9.0)
```

```
from tabulate import tabulate

#Fetch data
cursor.execute('SELECT * FROM CUSTOMER')
rows=cursor.fetchall()

#get the column names
column_names=[desc[0] for desc in cursor.description]

#print in tabular format
print(tabulate(rows,headers=column_names,tablefmt='grid'))
```

```
+---------------+-----------------+---------------------+--------------+-------------+
|   customer_id | customer_name   | Address             | state_code   |   Telephone |
+===============+=================+=====================+==============+=============+
|        123001 | Oliver          | 225-5, Emeryville   | CA           |  1897614500 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123002 | George          | 194-6,New brighton  | MN           |  1897617000 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123003 | Harry           | 2909-5,walnut creek | CA           |  1897617866 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123004 | Jack            | 229-5, Concord      | CA           |  1897627999 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123005 | Jacob           | 325-7, Mission Dist | SFO          |  1897637000 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123006 | Noah            | 275-9, saint-paul   | MN           |  1897613200 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123007 | Charlie         | 125-1,Richfield     | MN           |  1897617666 |
+---------------+-----------------+---------------------+--------------+-------------+
|        123008 | Robin           | 3005-1,Heathrow     | NY           |  1897614000 |
+---------------+-----------------+---------------------+--------------+-------------+
```

BANK_ACCOUNT_DETAIL TABLE

```
cursor.execute('''
CREATE TABLE Bank_Account_Detail(Customer_id INT,
                                 Account_Number VARCHAR(19),
                                 Account_type VARCHAR(25),
                                 Balance_amount INT,
                                 Account_status VARCHAR(10),
                                 Relationship_type varchar(1) ) ;


''')
conn.commit()
```

```
def insert_Account(Customer_id,Account_Number,Account_type,Balance_amount,Account_status,Relationship_type):
  cursor.execute('INSERT INTO Bank_Account_Detail(Customer_id,Account_Number,Account_type,Balance_amount,Account_status,Re
  conn.commit()

#insert values
insert_Account(123001, "4000-1956-3456",  "SAVINGS" , 200000 ,"ACTIVE","P");
insert_Account(123001, "5000-1700-3456", "RECURRING DEPOSITS" ,9400000 ,"ACTIVE","S");
insert_Account(123002, "4000-1956-2001",  "SAVINGS", 400000 ,"ACTIVE","P");
insert_Account(123002, "5000-1700-5001",  "RECURRING DEPOSITS" ,7500000 ,"ACTIVE","S");
insert_Account(123003, "4000-1956-2900",  "SAVINGS" ,750000,"INACTIVE","P");
insert_Account(123004, "5000-1700-6091", "RECURRING DEPOSITS" ,7500000 ,"ACTIVE","S");
insert_Account(123004, "4000-1956-3401",  "SAVINGS" , 655000 ,"ACTIVE","P");
insert_Account(123005, "4000-1956-5102",  "SAVINGS" , 300000 ,"ACTIVE","P");
insert_Account(123006, "4000-1956-5698",  "SAVINGS" , 455000 ,"ACTIVE" ,"P");
insert_Account(123007, "5000-1700-9800",  "SAVINGS" , 355000 ,"ACTIVE" ,"P");
insert_Account(123007, "4000-1956-9977",  "RECURRING DEPOSITS" , 7025000,"ACTIVE" ,"S");
insert_Account(123007, "9000-1700-7777-4321",  "Credit Card"    ,0  ,"INACTIVE", "P");
insert_Account(123007, '5900-1900-9877-5543', "Add-on Credit Card" ,   0   ,"ACTIVE", "S");
insert_Account(123008, "5000-1700-7755",  "SAVINGS"     ,0      ,"INACTIVE","P");
insert_Account(123006, '5800-1700-9800-7755', "Credit Card"   ,0    ,"ACTIVE", "P");
insert_Account(123006, '5890-1970-7706-8912', "Add-on Credit Card"   ,0     ,"ACTIVE", "S");
```

```
from tabulate import tabulate

#Fetch data
cursor.execute('SELECT DISTINCT * FROM Bank_Account_Detail')
rows=cursor.fetchall()

#get the column names
column_names=[desc[0] for desc in cursor.description]

#print in tabular format
print(tabulate(rows,headers=column_names,tablefmt='grid'))
```

```
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|   Customer_id | Account_Number      | Account_type       |   Balance_amount | Account_status   | Relationship_type    |
+===============+=====================+====================+==================+==================+======================+
|        123001 | 4000-1956-3456      | SAVINGS            |           200000 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123001 | 5000-1700-3456      | RECURRING DEPOSITS |          9400000 | ACTIVE           | S                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123002 | 4000-1956-2001      | SAVINGS            |           400000 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123002 | 5000-1700-5001      | RECURRING DEPOSITS |          7500000 | ACTIVE           | S                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123003 | 4000-1956-2900      | SAVINGS            |           750000 | INACTIVE         | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123004 | 5000-1700-6091      | RECURRING DEPOSITS |          7500000 | ACTIVE           | S                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123004 | 4000-1956-3401      | SAVINGS            |           655000 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123005 | 4000-1956-5102      | SAVINGS            |           300000 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123006 | 4000-1956-5698      | SAVINGS            |           455000 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123007 | 5000-1700-9800      | SAVINGS            |           355000 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123007 | 4000-1956-9977      | RECURRING DEPOSITS |          7025000 | ACTIVE           | S                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123007 | 9000-1700-7777-4321 | Credit Card        |                0 | INACTIVE         | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123007 | 5900-1900-9877-5543 | Add-on Credit Card |                0 | ACTIVE           | S                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123008 | 5000-1700-7755      | SAVINGS            |                0 | INACTIVE         | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123006 | 5800-1700-9800-7755 | Credit Card        |                0 | ACTIVE           | P                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
|        123006 | 5890-1970-7706-8912 | Add-on Credit Card |                0 | ACTIVE           | S                    |
+---------------+---------------------+--------------------+------------------+------------------+----------------------+
```

BANK_ACCOUNT_RELATIONSHIP_DETAILS TABLE

```
#create table
cursor.execute('''
  CREATE TABLE BANK_ACCOUNT_RELATIONSHIP_DETAILS(
        Customer_id INT,
        Account_Number VARCHAR(19),
        Account_type VARCHAR(25),
        Linking_Account_Number VARCHAR(19));
''')
conn.commit()
```

```
def insert_RELATION_DETAILS(Customer_id,Account_Number,Account_type,Linking_Account_Number):
  cursor.execute('INSERT INTO BANK_ACCOUNT_RELATIONSHIP_DETAILS(Customer_id,Account_Number,Account_type,Linking_Account_Nur
  conn.commit()

#insert
insert_RELATION_DETAILS(123001, "4000-1956-3456",  "SAVINGS" , "");
insert_RELATION_DETAILS(123001, "5000-1700-3456",  "RECURRING DEPOSITS" , "4000-1956-3456");
insert_RELATION_DETAILS(123002, "4000-1956-2001",  "SAVINGS" , "" );
insert_RELATION_DETAILS(123002, "5000-1700-5001",  "RECURRING DEPOSITS" , "4000-1956-2001" );
insert_RELATION_DETAILS(123003, "4000-1956-2900",  "SAVINGS" , "" );
insert_RELATION_DETAILS(123004, "5000-1700-6091",  "RECURRING DEPOSITS" , "4000-1956-2900" );
insert_RELATION_DETAILS(123004, "5000-1700-7791",  "RECURRING DEPOSITS" , "4000-1956-2900" );
insert_RELATION_DETAILS(123007, "5000-1700-9800",  "SAVINGS" , "" );
insert_RELATION_DETAILS(123007, "4000-1956-9977",  "RECURRING DEPOSITS" , "5000-1700-9800" );
insert_RELATION_DETAILS(None, "9000-1700-7777-4321",  "Credit Card" , "5000-1700-9800" );
insert_RELATION_DETAILS(None, '5900-1900-9877-5543', 'Add-on Credit Card', '9000-1700-7777-4321' );
insert_RELATION_DETAILS(None, '5800-1700-9800-7755', 'Credit Card', '4000-1956-5698' );
insert_RELATION_DETAILS(None, '5890-1970-7706-8912', 'Add-on Credit Card', '5800-1700-9800-7755' );
```

```
from tabulate import tabulate

#Fetch data
```

```
cursor.execute('SELECT * FROM BANK_ACCOUNT_RELATIONSHIP_DETAILS')
rows=cursor.fetchall()

#coloum values
coloum_values=[desc[0] for desc in cursor.description]

#print table
print(tabulate(rows,headers=coloum_values,tablefmt='grid'))
```

```
+---------------+---------------------+--------------------+--------------------------+
|   Customer_id | Account_Number      | Account_type       | Linking_Account_Number   |
+===============+=====================+====================+==========================+
|        123001 | 4000-1956-3456      | SAVINGS            |                          |
+---------------+---------------------+--------------------+--------------------------+
|        123001 | 5000-1700-3456      | RECURRING DEPOSITS | 4000-1956-3456           |
+---------------+---------------------+--------------------+--------------------------+
|        123002 | 4000-1956-2001      | SAVINGS            |                          |
+---------------+---------------------+--------------------+--------------------------+
|        123002 | 5000-1700-5001      | RECURRING DEPOSITS | 4000-1956-2001           |
+---------------+---------------------+--------------------+--------------------------+
|        123003 | 4000-1956-2900      | SAVINGS            |                          |
+---------------+---------------------+--------------------+--------------------------+
|        123004 | 5000-1700-6091      | RECURRING DEPOSITS | 4000-1956-2900           |
+---------------+---------------------+--------------------+--------------------------+
|        123004 | 5000-1700-7791      | RECURRING DEPOSITS | 4000-1956-2900           |
+---------------+---------------------+--------------------+--------------------------+
|        123007 | 5000-1700-9800      | SAVINGS            |                          |
+---------------+---------------------+--------------------+--------------------------+
|        123007 | 4000-1956-9977      | RECURRING DEPOSITS | 5000-1700-9800           |
+---------------+---------------------+--------------------+--------------------------+
|               | 9000-1700-7777-4321 | Credit Card        | 5000-1700-9800           |
+---------------+---------------------+--------------------+--------------------------+
|               | 5900-1900-9877-5543 | Add-on Credit Card | 9000-1700-7777-4321      |
+---------------+---------------------+--------------------+--------------------------+
|               | 5800-1700-9800-7755 | Credit Card        | 4000-1956-5698           |
+---------------+---------------------+--------------------+--------------------------+
|               | 5890-1970-7706-8912 | Add-on Credit Card | 5800-1700-9800-7755      |
+---------------+---------------------+--------------------+--------------------------+
```

BANK_ACCOUNT_TRANSACTION TABLE

```
cursor.execute('''
  CREATE TABLE BANK_ACCOUNT_TRANSACTION (
                Account_Number VARCHAR(19),
                Transaction_amount Decimal(18,2) ,
                Transcation_channel VARCHAR(18) ,
                Province varchar(3) ,
                Transaction_Date Date) ;
''')
conn.commit()
```

```
def insert_TRANSACTION(Account_Number,Transaction_amount,Transcation_channel,Province,Transaction_Date):
  cursor.execute('INSERT INTO BANK_ACCOUNT_TRANSACTION(Account_Number,Transaction_amount,Transcation_channel,Province,Trans
  conn.commit()

#insert
insert_TRANSACTION( "4000-1956-3456",  -2000, "ATM withdrawl" , "CA", "2020-01-13");
insert_TRANSACTION( "4000-1956-2001",  -4000, "POS-Walmart"   , "MN", "2020-02-14");
insert_TRANSACTION( "4000-1956-2001",  -1600, "UPI transfer"  , "MN", "2020-01-19");
insert_TRANSACTION( "4000-1956-2001",  -6000, "Bankers cheque", "CA", "2020-03-23");
insert_TRANSACTION( "4000-1956-2001",  -3000, "Net banking"   , "CA", "2020-04-24");
insert_TRANSACTION( "4000-1956-2001",  23000, "cheque deposit", "MN", "2020-03-15");
insert_TRANSACTION( "5000-1700-6091",  40000, "ECS transfer"  , "NY", "2020-02-19");
insert_TRANSACTION( "5000-1700-7791",  40000, "ECS transfer"  , "NY", "2020-02-19");
insert_TRANSACTION( "4000-1956-3401",   8000, "Cash Deposit"  , "NY", "2020-01-19");
insert_TRANSACTION( "4000-1956-5102",  -6500, "ATM withdrawal" , "NY", "2020-03-14");
insert_TRANSACTION( "4000-1956-5698",  -9000, "Cash Deposit"  , "NY", "2020-03-27");
insert_TRANSACTION( "4000-1956-9977",  50000, "ECS transfer"  , "NY", "2020-01-16");
insert_TRANSACTION( "9000-1700-7777-4321",  -5000, "POS-Walmart", "NY", "2020-02-17");
insert_TRANSACTION( "9000-1700-7777-4321",  -8000, "Shopping Cart", "MN", "2020-03-13");
insert_TRANSACTION( "9000-1700-7777-4321",  -2500, "Shopping Cart", "MN", "2020-04-21");
insert_TRANSACTION( "5800-1700-9800-7755", -9000, "POS-Walmart","MN", "2020-04-13");
insert_TRANSACTION( '5890-1970-7706-8912', -11000, "Shopping Cart" , "NY" , "2020-03-12") ;
```

```
insert_TRANSACTION('4000-1956-9977' ,    40000.00     ,'ECS transfer',     'MN' ,    '2020-03-18' ) ;
insert_TRANSACTION('4000-1956-9977' ,    60000.00     ,'ECS transfer',     'MN' ,    '2020-04-18' ) ;
insert_TRANSACTION('4000-1956-9977' ,    20000.00     ,'ECS transfer',     'MN' ,    '2020-03-20' ) ;
insert_TRANSACTION('4000-1956-9977' ,    49000.00     ,'ECS transfer',     'MN' ,    '2020-06-18' ) ;
```

```
from tabulate import tabulate

#Fetch data
cursor.execute('SELECT * FROM BANK_ACCOUNT_TRANSACTION')
rows=cursor.fetchall()

#coloum names
coloum_names=[desc[0] for desc in cursor.description]

#print table
print(tabulate(rows,headers=coloum_names,tablefmt='grid'))
```

```
+--------------------+---------------------+-----------------------+------------+--------------------+
| Account_Number     |   Transaction_amount | Transcation_channel   | Province   | Transaction_Date   |
+====================+=====================+=======================+============+====================+
| 4000-1956-3456     |               -2000 | ATM withdrawl         | CA         | 2020-01-13         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-2001     |               -4000 | POS-Walmart           | MN         | 2020-02-14         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-2001     |               -1600 | UPI transfer          | MN         | 2020-01-19         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-2001     |               -6000 | Bankers cheque        | CA         | 2020-03-23         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-2001     |               -3000 | Net banking           | CA         | 2020-04-24         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-2001     |               23000 | cheque deposit        | MN         | 2020-03-15         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 5000-1700-6091     |               40000 | ECS transfer          | NY         | 2020-02-19         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 5000-1700-7791     |               40000 | ECS transfer          | NY         | 2020-02-19         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-3401     |                8000 | Cash Deposit          | NY         | 2020-01-19         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-5102     |               -6500 | ATM withdrawal        | NY         | 2020-03-14         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-5698     |               -9000 | Cash Deposit          | NY         | 2020-03-27         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-9977     |               50000 | ECS transfer          | NY         | 2020-01-16         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 9000-1700-7777-4321 |               -5000 | POS-Walmart           | NY         | 2020-02-17         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 9000-1700-7777-4321 |               -8000 | Shopping Cart         | MN         | 2020-03-13         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 9000-1700-7777-4321 |               -2500 | Shopping Cart         | MN         | 2020-04-21         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 5800-1700-9800-7755 |               -9000 | POS-Walmart           | MN         | 2020-04-13         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 5890-1970-7706-8912 |              -11000 | Shopping Cart         | NY         | 2020-03-12         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-9977     |               40000 | ECS transfer          | MN         | 2020-03-18         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-9977     |               60000 | ECS transfer          | MN         | 2020-04-18         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-9977     |               20000 | ECS transfer          | MN         | 2020-03-20         |
+--------------------+---------------------+-----------------------+------------+--------------------+
| 4000-1956-9977     |               49000 | ECS transfer          | MN         | 2020-06-18         |
+--------------------+---------------------+-----------------------+------------+--------------------+
```

BANK_CUSTOMER_MESSAGES TABLE

```
cursor.execute('''
  CREATE TABLE BANK_CUSTOMER_MESSAGES(
        Event VARCHAR(24),
        Customer_message VARCHAR(75),
        Notice_delivary_mode VARCHAR(15)
  );
''')
conn.commit()
```

```
def insert_MESSAGE(Event,Customer_message,Notice_delivary_mode):
  cursor.execute('INSERT INTO BANK_CUSTOMER_MESSAGES(Event,Customer_message,Notice_delivary_mode) VALUES(?,?,?)',(Event,Cus
  conn.commit()

#insert values
insert_MESSAGE( "Adhoc", "All Banks are closed due to announcement of National strike", "mobile" ) ;
insert_MESSAGE( "Transaction Limit", "Only limited withdrawals per card are allowed from ATM machines", "mobile" );
```

```
from tabulate import tabulate

#fetch data
cursor.execute('SELECT * FROM BANK_CUSTOMER_MESSAGES')
rows=cursor.fetchall()
```

```
# col values
coloum_names=[desc[0] for desc in cursor.description]

#print table
print(tabulate(rows,headers=coloum_names,tablefmt='grid'))
```

```
+------------------+-----------------------------------------------------------------+------------------------+
| Event            | Customer_message                                                | Notice_delivary_mode   |
+==================+=================================================================+========================+
| Adhoc            | All Banks are closed due to announcement of National strike     | mobile                 |
+------------------+-----------------------------------------------------------------+------------------------+
| Transaction Limit | Only limited withdrawals per card are allowed from ATM machines | mobile                 |
+------------------+-----------------------------------------------------------------+------------------------+
```

BANK_INTEREST_RATE TABLE

```
cursor.execute('''
  CREATE TABLE BANK_INTEREST_RATE(
               Account_type varchar(24),
              Interest_rate decimal(4,2),
               month varchar(2),
               year  varchar(4));

''')
conn.commit()
```

```
def insert_RATE(Account_type,Interest_rate,month,year):
  cursor.execute('INSERT INTO BANK_INTEREST_RATE(Account_type,Interest_rate,month,year) VALUES(?,?,?,?)',(Account_type,Inte
  conn.commit()

#insert values
insert_RATE( "SAVINGS" , 0.04 , '02' , '2020' );
insert_RATE( "RECURRING DEPOSITS" , 0.07, '02' , '2020' );
insert_RATE( "PRIVILEGED_INTEREST_RATE" , 0.08 , '02' , '2020' );
```

```
from tabulate import tabulate

#fetch data
cursor.execute('SELECT * FROM BANK_INTEREST_RATE')
rows=cursor.fetchall()

# col names
coloum_names=[desc[0] for desc in cursor.description]

#print table
print(tabulate(rows,headers=coloum_names,tablefmt='grid'))
```

```
+--------------------------+-----------------+---------+--------+
| account_type             |   interest_rate |   month |   year |
+==========================+=================+=========+========+
| SAVINGS                  |            0.04 |      02 |   2020 |
+--------------------------+-----------------+---------+--------+
| RECURRING DEPOSITS       |            0.07 |      02 |   2020 |
+--------------------------+-----------------+---------+--------+
| PRIVILEGED_INTEREST_RATE |            0.08 |      02 |   2020 |
+--------------------------+-----------------+---------+--------+
```

Question 1:

Print customer Id, customer name and average account_balance maintained by each customer for all of his/her accounts in the bank.(8 Rows)

```
# SQL query to get customer_id, customer_name, and average balance amount
query = """
SELECT
    c.customer_id,
    c.customer_name,
    AVG(a.balance_amount) AS avg_balance
FROM CUSTOMER c
LEFT JOIN BANK_Account_Detail a ON c.customer_id = a.customer_id
GROUP BY c.customer_id, c.customer_name
LIMIT 8;  -- Show only 8 rows
"""

# Execute the query
cursor.execute(query)
```

```
# Fetch column names
columns = [desc[0] for desc in cursor.description]

# Fetch all results
results = cursor.fetchall()

# Print table headers
print("{:<12} {:<20} {:<15}".format(*columns))
print("-" * 50)
#print(tabulate(results,headers=columns,tablefmt='grid'))

# Print each row in a properly formatted way
for row in results:
    print("{:<12} {:<20} {:<15.2f}".format(*row))  # Format avg_balance to 2 decimal places
```

```
customer_id  customer_name        avg_balance
--------------------------------------------------
123001       Oliver               4800000.00
123002       George               3950000.00
123003       Harry                750000.00
123004       Jack                 4077500.00
123005       Jacob                300000.00
123006       Noah                 151666.67
123007       Charlie              1845000.00
123008       Robin                0.00
```

Question 2:

Print customer_id , account_number and balance_amount , condition that if balance_amount is nil then assign transaction_amount for account_type = "Credit Card"(4 Rows)

```
query = """
  SELECT
    a.Customer_id,a.Account_Number,
    COALESCE(NULLIF(a.Balance_amount, 0), t.Transaction_amount) AS Final_Balance
FROM BANK_Account_Detail a
LEFT JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number
WHERE a.Account_type = 'Credit Card'
LIMIT 4;

"""
result = pd.read_sql_query(query, conn)
print(tabulate(result, headers='keys', tablefmt='psql'))
```

```
+----+---------------+---------------------+-----------------+
|    |   Customer_id | Account_Number      |   Final_Balance |
|----+---------------+---------------------+-----------------|
|  0 |        123007 | 9000-1700-7777-4321 |           -8000 |
|  1 |        123007 | 9000-1700-7777-4321 |           -5000 |
|  2 |        123007 | 9000-1700-7777-4321 |           -2500 |
|  3 |        123006 | 5800-1700-9800-7755 |           -9000 |
+----+---------------+---------------------+-----------------+
```

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(12 Rows)

```
query=('''
SELECT DISTINCT a.Account_Number,a.Balance_amount,
       t.Transaction_amount,t.Transaction_date
FROM Bank_Account_Detail a
LEFT JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number
WHERE T.Transaction_date BETWEEN '2020-03-01' AND '2020-04-31'
LIMIT 12;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------------+-----------------+---------------------+--------------------+
|    | Account_Number      |   Balance_amount |   Transaction_amount | Transaction_Date   |
+====+=====================+=================+=====================+====================+
|  0 | 4000-1956-2001      |          400000 |               -6000 | 2020-03-23         |
+----+---------------------+-----------------+---------------------+--------------------+
|  1 | 4000-1956-2001      |          400000 |               -3000 | 2020-04-24         |
+----+---------------------+-----------------+---------------------+--------------------+
|  2 | 4000-1956-2001      |          400000 |               23000 | 2020-03-15         |
+----+---------------------+-----------------+---------------------+--------------------+
|  3 | 4000-1956-5102      |          300000 |               -6500 | 2020-03-14         |
+----+---------------------+-----------------+---------------------+--------------------+
|  4 | 4000-1956-5698      |          455000 |               -9000 | 2020-03-27         |
```

```
+----+---------------------+-----------------+---------------------+--------------------+
|  5 | 9000-1700-7777-4321 |               0 |               -8000 | 2020-03-13         |
+----+---------------------+-----------------+---------------------+--------------------+
|  6 | 9000-1700-7777-4321 |               0 |               -2500 | 2020-04-21         |
+----+---------------------+-----------------+---------------------+--------------------+
|  7 | 5800-1700-9800-7755 |               0 |               -9000 | 2020-04-13         |
+----+---------------------+-----------------+---------------------+--------------------+
|  8 | 5890-1970-7706-8912 |               0 |              -11000 | 2020-03-12         |
+----+---------------------+-----------------+---------------------+--------------------+
|  9 | 4000-1956-9977      |         7025000 |               40000 | 2020-03-18         |
+----+---------------------+-----------------+---------------------+--------------------+
| 10 | 4000-1956-9977      |         7025000 |               60000 | 2020-04-18         |
+----+---------------------+-----------------+---------------------+--------------------+
| 11 | 4000-1956-9977      |         7025000 |               20000 | 2020-03-20         |
+----+---------------------+-----------------+---------------------+--------------------+
```

Question 4:

Print all of the customer id, 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 (22 Rows)

```
#from calendar import month

query=('''
SELECT
    c.Customer_id,
    a.Account_Number,a.Balance_amount,
    t.Transaction_amount,t.Transaction_date
FROM CUSTOMER c
INNER JOIN Bank_Account_Detail a ON c.Customer_id = a.Customer_id
INNER JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number
WHERE t.Transaction_date LIKE '2020-03-%'
-- WHERE MONTH[t.Transaction_date] = '03'
LIMIT 22;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------+---------------------+------------------+----------------------+--------------------+
|    |   customer_id | Account_Number      |   Balance_amount |   Transaction_amount | Transaction_Date   |
+====+===============+=====================+==================+======================+====================+
|  0 |        123002 | 4000-1956-2001      |           400000 |                -6000 | 2020-03-23         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  1 |        123002 | 4000-1956-2001      |           400000 |                -6000 | 2020-03-23         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  2 |        123002 | 4000-1956-2001      |           400000 |                23000 | 2020-03-15         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  3 |        123002 | 4000-1956-2001      |           400000 |                23000 | 2020-03-15         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  4 |        123005 | 4000-1956-5102      |           300000 |                -6500 | 2020-03-14         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  5 |        123005 | 4000-1956-5102      |           300000 |                -6500 | 2020-03-14         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  6 |        123006 | 4000-1956-5698      |           455000 |                -9000 | 2020-03-27         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  7 |        123006 | 4000-1956-5698      |           455000 |                -9000 | 2020-03-27         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  8 |        123007 | 9000-1700-7777-4321 |                0 |                -8000 | 2020-03-13         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  9 |        123007 | 9000-1700-7777-4321 |                0 |                -8000 | 2020-03-13         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 10 |        123006 | 5890-1970-7706-8912 |                0 |               -11000 | 2020-03-12         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 11 |        123006 | 5890-1970-7706-8912 |                0 |               -11000 | 2020-03-12         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 12 |        123007 | 4000-1956-9977      |          7025000 |                40000 | 2020-03-18         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 13 |        123007 | 4000-1956-9977      |          7025000 |                40000 | 2020-03-18         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 14 |        123007 | 4000-1956-9977      |          7025000 |                20000 | 2020-03-20         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 15 |        123007 | 4000-1956-9977      |          7025000 |                20000 | 2020-03-20         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
```

Question 5:

Print only the customer id, 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.(16 Rows)

```
query=('''
SELECT
    c.Customer_id,
    a.Account_Number,a.Balance_amount,
    t.Transaction_amount,t.Transaction_date
```

```
FROM CUSTOMER AS c
INNER JOIN Bank_Account_Detail AS a ON c.Customer_id = a.Customer_id
INNER JOIN BANK_ACCOUNT_TRANSACTION AS t ON a.Account_Number = t.Account_Number
WHERE t.Transaction_date BETWEEN '2020-01-01' AND '2020-03-31'
LIMIT 16;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------+---------------------+------------------+----------------------+--------------------+
|    |   customer_id | Account_Number      |   Balance_amount |   Transaction_amount | Transaction_Date   |
+====+===============+=====================+==================+======================+====================+
|  0 |        123001 | 4000-1956-3456      |           200000 |                -2000 | 2020-01-13         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  1 |        123002 | 4000-1956-2001      |           400000 |                -4000 | 2020-02-14         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  2 |        123002 | 4000-1956-2001      |           400000 |                -1600 | 2020-01-19         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  3 |        123002 | 4000-1956-2001      |           400000 |                -6000 | 2020-03-23         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  4 |        123002 | 4000-1956-2001      |           400000 |                23000 | 2020-03-15         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  5 |        123004 | 5000-1700-6091      |          7500000 |                40000 | 2020-02-19         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  6 |        123004 | 4000-1956-3401      |           655000 |                 8000 | 2020-01-19         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  7 |        123005 | 4000-1956-5102      |           300000 |                -6500 | 2020-03-14         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  8 |        123006 | 4000-1956-5698      |           455000 |                -9000 | 2020-03-27         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
|  9 |        123007 | 4000-1956-9977      |          7025000 |                50000 | 2020-01-16         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 10 |        123007 | 9000-1700-7777-4321 |                0 |                -5000 | 2020-02-17         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 11 |        123007 | 9000-1700-7777-4321 |                0 |                -8000 | 2020-03-13         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 12 |        123006 | 5890-1970-7706-8912 |                0 |               -11000 | 2020-03-12         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 13 |        123007 | 4000-1956-9977      |          7025000 |                40000 | 2020-03-18         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
| 14 |        123007 | 4000-1956-9977      |          7025000 |                20000 | 2020-03-20         |
+----+---------------+---------------------+------------------+----------------------+--------------------+
```

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.(8 Rows)

```
query=('''
  SELECT
    a.Account_Number,
    m.Event,m.Customer_message
  FROM Bank_Account_Detail a
  CROSS JOIN BANK_CUSTOMER_MESSAGES m ON m.Event = 'Adhoc'
  WHERE a.Account_type = 'SAVINGS'
  LIMIT 8;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+------------------+---------+-------------------------------------------------------------+
|    | Account_Number   | Event   | Customer_message                                            |
+====+==================+=========+=============================================================+
|  0 | 4000-1956-3456   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  1 | 4000-1956-2001   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  2 | 4000-1956-2900   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  3 | 4000-1956-3401   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  4 | 4000-1956-5102   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  5 | 4000-1956-5698   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  6 | 5000-1700-9800   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
|  7 | 5000-1700-7755   | Adhoc   | All Banks are closed due to announcement of National strike |
+----+------------------+---------+-------------------------------------------------------------+
```

Question 7:

Print all Customer_id, 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) .(27 Rows)

```
query=('''
  SELECT
    a.Customer_id,a.Account_Number,a.Account_type,-- a.Balance_amount,t.Transaction_amount,a.Relationship_type,
    ((a.Balance_amount) + (t.Transaction_amount)) AS Deducted_Balance_Amount
  FROM Bank_Account_Detail a
  LEFT JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number
  WHERE a.Relationship_type ='P' AND t.Transaction_amount < 0
  LIMIT 27;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------+---------------------+----------------+---------------------------+
|    |   Customer_id | Account_Number      | Account_type   |   Deducted_Balance_Amount |
+====+===============+=====================+================+===========================+
|  0 |        123001 | 4000-1956-3456      | SAVINGS        |                    198000 |
+----+---------------+---------------------+----------------+---------------------------+
|  1 |        123002 | 4000-1956-2001      | SAVINGS        |                    394000 |
+----+---------------+---------------------+----------------+---------------------------+
|  2 |        123002 | 4000-1956-2001      | SAVINGS        |                    396000 |
+----+---------------+---------------------+----------------+---------------------------+
|  3 |        123002 | 4000-1956-2001      | SAVINGS        |                    397000 |
+----+---------------+---------------------+----------------+---------------------------+
|  4 |        123002 | 4000-1956-2001      | SAVINGS        |                    398400 |
+----+---------------+---------------------+----------------+---------------------------+
|  5 |        123005 | 4000-1956-5102      | SAVINGS        |                    293500 |
+----+---------------+---------------------+----------------+---------------------------+
|  6 |        123006 | 4000-1956-5698      | SAVINGS        |                    446000 |
+----+---------------+---------------------+----------------+---------------------------+
|  7 |        123007 | 9000-1700-7777-4321 | Credit Card    |                     -8000 |
+----+---------------+---------------------+----------------+---------------------------+
|  8 |        123007 | 9000-1700-7777-4321 | Credit Card    |                     -5000 |
+----+---------------+---------------------+----------------+---------------------------+
|  9 |        123007 | 9000-1700-7777-4321 | Credit Card    |                     -2500 |
+----+---------------+---------------------+----------------+---------------------------+
| 10 |        123006 | 5800-1700-9800-7755 | Credit Card    |                     -9000 |
+----+---------------+---------------------+----------------+---------------------------+
| 11 |        123001 | 4000-1956-3456      | SAVINGS        |                    198000 |
+----+---------------+---------------------+----------------+---------------------------+
| 12 |        123002 | 4000-1956-2001      | SAVINGS        |                    394000 |
+----+---------------+---------------------+----------------+---------------------------+
| 13 |        123002 | 4000-1956-2001      | SAVINGS        |                    396000 |
+----+---------------+---------------------+----------------+---------------------------+
| 14 |        123002 | 4000-1956-2001      | SAVINGS        |                    397000 |
+----+---------------+---------------------+----------------+---------------------------+
| 15 |        123002 | 4000-1956-2001      | SAVINGS        |                    398400 |
+----+---------------+---------------------+----------------+---------------------------+
| 16 |        123005 | 4000-1956-5102      | SAVINGS        |                    293500 |
+----+---------------+---------------------+----------------+---------------------------+
| 17 |        123006 | 4000-1956-5698      | SAVINGS        |                    446000 |
+----+---------------+---------------------+----------------+---------------------------+
| 18 |        123007 | 9000-1700-7777-4321 | Credit Card    |                     -8000 |
+----+---------------+---------------------+----------------+---------------------------+
| 19 |        123007 | 9000-1700-7777-4321 | Credit Card    |                     -5000 |
+----+---------------+---------------------+----------------+---------------------------+
| 20 |        123007 | 9000-1700-7777-4321 | Credit Card    |                     -2500 |
+----+---------------+---------------------+----------------+---------------------------+
| 21 |        123006 | 5800-1700-9800-7755 | Credit Card    |                     -9000 |
+----+---------------+---------------------+----------------+---------------------------+
```

Question 8:

a) Display records of All Accounts , their Account_types, the transaction amount.

```
query=('''
  SELECT DISTINCT a.Account_Number,a.Account_type,
         t.Transaction_amount
  FROM Bank_Account_Detail a
  LEFT JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number;
''')


result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------------+--------------------+----------------------+
|    | Account_Number      | Account_type       |   Transaction_amount |
+====+=====================+====================+======================+
|  0 | 4000-1956-3456      | SAVINGS            |                -2000 |
+----+---------------------+--------------------+----------------------+
|  1 | 5000-1700-3456      | RECURRING DEPOSITS |                  nan |
+----+---------------------+--------------------+----------------------+
|  2 | 4000-1956-2001      | SAVINGS            |                -6000 |
+----+---------------------+--------------------+----------------------+
```

```
|  3 | 4000-1956-2001      | SAVINGS            |                -4000 |
+----+---------------------+--------------------+----------------------+
|  4 | 4000-1956-2001      | SAVINGS            |                -3000 |
+----+---------------------+--------------------+----------------------+
|  5 | 4000-1956-2001      | SAVINGS            |                -1600 |
+----+---------------------+--------------------+----------------------+
|  6 | 4000-1956-2001      | SAVINGS            |                23000 |
+----+---------------------+--------------------+----------------------+
|  7 | 5000-1700-5001      | RECURRING DEPOSITS |                  nan |
+----+---------------------+--------------------+----------------------+
|  8 | 4000-1956-2900      | SAVINGS            |                  nan |
+----+---------------------+--------------------+----------------------+
|  9 | 5000-1700-6091      | RECURRING DEPOSITS |                40000 |
+----+---------------------+--------------------+----------------------+
| 10 | 4000-1956-3401      | SAVINGS            |                 8000 |
+----+---------------------+--------------------+----------------------+
| 11 | 4000-1956-5102      | SAVINGS            |                -6500 |
+----+---------------------+--------------------+----------------------+
| 12 | 4000-1956-5698      | SAVINGS            |                -9000 |
+----+---------------------+--------------------+----------------------+
| 13 | 5000-1700-9800      | SAVINGS            |                  nan |
+----+---------------------+--------------------+----------------------+
| 14 | 4000-1956-9977      | RECURRING DEPOSITS |                20000 |
+----+---------------------+--------------------+----------------------+
| 15 | 4000-1956-9977      | RECURRING DEPOSITS |                40000 |
+----+---------------------+--------------------+----------------------+
| 16 | 4000-1956-9977      | RECURRING DEPOSITS |                49000 |
+----+---------------------+--------------------+----------------------+
| 17 | 4000-1956-9977      | RECURRING DEPOSITS |                50000 |
+----+---------------------+--------------------+----------------------+
| 18 | 4000-1956-9977      | RECURRING DEPOSITS |                60000 |
+----+---------------------+--------------------+----------------------+
| 19 | 9000-1700-7777-4321 | Credit Card        |                -8000 |
+----+---------------------+--------------------+----------------------+
| 20 | 9000-1700-7777-4321 | Credit Card        |                -5000 |
+----+---------------------+--------------------+----------------------+
| 21 | 9000-1700-7777-4321 | Credit Card        |                -2500 |
+----+---------------------+--------------------+----------------------+
| 22 | 5900-1900-9877-5543 | Add-on Credit Card |                  nan |
+----+---------------------+--------------------+----------------------+
| 23 | 5000-1700-7755      | SAVINGS            |                  nan |
+----+---------------------+--------------------+----------------------+
| 24 | 5800-1700-9800-7755 | Credit Card        |                -9000 |
+----+---------------------+--------------------+----------------------+
| 25 | 5890-1970-7706-8912 | Add-on Credit Card |               -11000 |
+----+---------------------+--------------------+----------------------+
```

b) Along with first step, Display other columns with corresponding linking account number, account types (15 Rows)

```
query=('''
  SELECT a.Account_Number,a.Account_type,
         t.Transaction_amount,
         r.Customer_id,r.Linking_Account_Number,r.Account_type AS Acc_Type
  FROM Bank_Account_Detail a
  LEFT JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number
  LEFT JOIN BANK_ACCOUNT_RELATIONSHIP_DETAILS r ON r.Linking_Account_Number = a.Account_Number
  LIMIT 15;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|    | Account_Number   | Account_type       |   Transaction_amount |   Customer_id | Linking_Account_Number   | Acc_Type
+====+==================+====================+======================+===============+==========================+============
|  0 | 4000-1956-3456   | SAVINGS            |                -2000 |        123001 | 4000-1956-3456           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  1 | 5000-1700-3456   | RECURRING DEPOSITS |                  nan |           nan |                          |
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  2 | 4000-1956-2001   | SAVINGS            |                -6000 |        123002 | 4000-1956-2001           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  3 | 4000-1956-2001   | SAVINGS            |                -4000 |        123002 | 4000-1956-2001           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  4 | 4000-1956-2001   | SAVINGS            |                -3000 |        123002 | 4000-1956-2001           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  5 | 4000-1956-2001   | SAVINGS            |                -1600 |        123002 | 4000-1956-2001           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  6 | 4000-1956-2001   | SAVINGS            |                23000 |        123002 | 4000-1956-2001           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  7 | 5000-1700-5001   | RECURRING DEPOSITS |                  nan |           nan |                          |
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  8 | 4000-1956-2900   | SAVINGS            |                  nan |        123004 | 4000-1956-2900           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
|  9 | 4000-1956-2900   | SAVINGS            |                  nan |        123004 | 4000-1956-2900           | RECURRING D
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
| 10 | 5000-1700-6091   | RECURRING DEPOSITS |                40000 |           nan |                          |
+----+------------------+--------------------+----------------------+--------------+--------------------------+------------
| 11 | 4000-1956-3401   | SAVINGS            |                 8000 |           nan |                          |
```

```
+----+------------------+--------------------+----------------------+--------------+--------------------------+-------------
| 12 | 4000-1956-5102   | SAVINGS            |                -6500 |          nan |                          |
+----+------------------+--------------------+----------------------+--------------+--------------------------+-------------
| 13 | 4000-1956-5698   | SAVINGS            |                -9000 |          nan | 4000-1956-5698           | Credit Card
+----+------------------+--------------------+----------------------+--------------+--------------------------+-------------
| 14 | 5000-1700-9800   | SAVINGS            |                  nan |          nan | 5000-1700-9800           | Credit Card
+----+------------------+--------------------+----------------------+--------------+--------------------------+-------------
```

Question 9:

a) Display records of All Accounts , their Account_types, the transaction amount.

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

c) After retrieving all records of accounts and their linked accounts, display the transaction amount of accounts appeared in another column.(26 Rows)

```
query = ('''
  SELECT DISTINCT
         a.Account_Number,a.Account_type,
         t.Transaction_amount,
         r.Linking_Account_Number,r.Account_type AS Acc_Type,
         tr.Transaction_amount AS Amount
  FROM Bank_Account_Detail a
  LEFT JOIN BANK_ACCOUNT_TRANSACTION t ON a.Account_Number = t.Account_Number
  LEFT JOIN BANK_ACCOUNT_RELATIONSHIP_DETAILS r ON a.Account_Number = r.Account_Number
  LEFT JOIN BANK_ACCOUNT_TRANSACTION tr ON r.Linking_Account_Number = tr.Account_Number
  LIMIT 26;
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|    | Account_Number      | Account_type       |   Transaction_amount | Linking_Account_Number   | Acc_Type           |   A
+====+=====================+====================+======================+==========================+====================+====
|  0 | 4000-1956-3456      | SAVINGS            |                -2000 |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  1 | 5000-1700-3456      | RECURRING DEPOSITS |                  nan | 4000-1956-3456           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  2 | 4000-1956-2001      | SAVINGS            |                -6000 |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  3 | 4000-1956-2001      | SAVINGS            |                -4000 |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  4 | 4000-1956-2001      | SAVINGS            |                -3000 |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  5 | 4000-1956-2001      | SAVINGS            |                -1600 |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  6 | 4000-1956-2001      | SAVINGS            |                23000 |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  7 | 5000-1700-5001      | RECURRING DEPOSITS |                  nan | 4000-1956-2001           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  8 | 5000-1700-5001      | RECURRING DEPOSITS |                  nan | 4000-1956-2001           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
|  9 | 5000-1700-5001      | RECURRING DEPOSITS |                  nan | 4000-1956-2001           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 10 | 5000-1700-5001      | RECURRING DEPOSITS |                  nan | 4000-1956-2001           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 11 | 5000-1700-5001      | RECURRING DEPOSITS |                  nan | 4000-1956-2001           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 12 | 4000-1956-2900      | SAVINGS            |                  nan |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 13 | 5000-1700-6091      | RECURRING DEPOSITS |                40000 | 4000-1956-2900           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 14 | 4000-1956-3401      | SAVINGS            |                 8000 |                          |                    |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 15 | 4000-1956-5102      | SAVINGS            |                -6500 |                          |                    |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 16 | 4000-1956-5698      | SAVINGS            |                -9000 |                          |                    |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 17 | 5000-1700-9800      | SAVINGS            |                  nan |                          | SAVINGS            |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 18 | 4000-1956-9977      | RECURRING DEPOSITS |                20000 | 5000-1700-9800           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 19 | 4000-1956-9977      | RECURRING DEPOSITS |                40000 | 5000-1700-9800           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 20 | 4000-1956-9977      | RECURRING DEPOSITS |                49000 | 5000-1700-9800           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 21 | 4000-1956-9977      | RECURRING DEPOSITS |                50000 | 5000-1700-9800           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 22 | 4000-1956-9977      | RECURRING DEPOSITS |                60000 | 5000-1700-9800           | RECURRING DEPOSITS |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 23 | 9000-1700-7777-4321 | Credit Card        |                -8000 | 5000-1700-9800           | Credit Card        |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 24 | 9000-1700-7777-4321 | Credit Card        |                -5000 | 5000-1700-9800           | Credit Card        |
+----+---------------------+--------------------+----------------------+--------------------------+--------------------+----
| 25 | 9000-1700-7777-4321 | Credit Card        |                -2500 | 5000-1700-9800           | Credit Card        |
```

```
+----+---------------------+-------------------+---------------------+-------------------------+-------------------+----
```

Question 10:

Display all saving account holders have “Add-on Credit Cards" and “Credit cards" (3 Rows)

```
query=('''
  SELECT
    c.Customer_id, c.Customer_name
  FROM CUSTOMER c
  JOIN Bank_Account_Detail a ON c.Customer_id = a.Customer_id -- AND a.Account_type = 'SAVINGS'
  JOIN Bank_account_Detail ac ON c.Customer_id = ac.Customer_id -- AND ac.Account_type = 'Add-on Credit Card'
  JOIN Bank_account_Detail acc ON c.Customer_id = acc.Customer_id -- AND acc.Account_type = 'Credit Card'
  WHERE a.Account_type = 'SAVINGS' AND ac.Account_type = 'Add-on Credit Card' AND acc.Account_type = 'Credit Card'
  GROUP BY c.Customer_id,c.Customer_name ;
  ''')
result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+---------------+-----------------+
|    |   customer_id | customer_name   |
+====+===============+=================+
|  0 |        123006 | Noah            |
+----+---------------+-----------------+
|  1 |        123007 | Charlie         |
+----+---------------+-----------------+
```

Question 11:

Display records of “SAVINGS” account linked with “Credit cards" account_type and its credit aggregate sum of transaction amount. (1 Row)

Ref: Check linking relationship in bank_transaction_relationship_details.

Check transaction_amount in bank_account_transaction.

```
query=('''
  SELECT r.Account_Number,r.Account_type AS Account,
         re.Account_type, re.Linking_Account_Number,
         SUM(t.Transaction_amount) AS Transaction_Amount
  FROM BANK_ACCOUNT_RELATIONSHIP_DETAILS r
  JOIN BANK_ACCOUNT_RELATIONSHIP_DETAILS re ON r.Account_number = re.Linking_Account_Number AND r.Account_type = 'SAVINGS'
  JOIN BANK_ACCOUNT_TRANSACTION t ON t.Account_Number = re.Account_Number
  WHERE re.Account_type = 'Credit Card';
''')

result=pd.read_sql_query(query,conn)
print(tabulate(result,headers='keys',tablefmt='grid'))
```

```
+----+------------------+-----------+----------------+--------------------------+----------------------+
|    | Account_Number   | Account   | Account_type   | Linking_Account_Number   |   Transaction_Amount |
+====+==================+===========+================+==========================+======================+
|  0 | 5000-1700-9800   | SAVINGS   | Credit Card    | 5000-1700-9800           |               -15500 |
+----+------------------+-----------+----------------+--------------------------+----------------------+
```

```
Start coding or generate with AI.
```