University of Toronto • Faculty of Information • Master of Information

INF 1343 • Data Modelling and Database Design

Course Project – Phase 2

Instructor Name: Nada Almasri

Student Details:

Omer Imran	1008712190	Bhanvi Gupta	1009060472
Fatima Zohra	1004167876	Michaela Drouillard	1002949358

Table of Contents

Word Count: 9,547

Executive Summary

Scotiabank is one of the "Big 5" retail banks in Canada. Its core business objectives are to provide a wide variety of banking and financial services like personal and commercial banking, wealth management, corporate lending, investment banking, and more. Scotiabank has a long history dating back to 1832 when it was founded in Halifax, Nova Scotia as the Bank of Nova Scotia. It has grown over the years to become an international bank with operations in over 55 countries around the world.

Since one of our team members works for a bank, we've chosen to develop a database for Scotiabank's retail banking department, where the key objectives are to manage customer accounts and relationships, process transactions efficiently, provide competitive lending rates, issue credit cards, and offer customer service. Typically this database design is proprietary, so our model is our interpretation of the demands of a retail banking department.

The core business processes that this database needs to support are:

- Opening and closing personal and commercial bank accounts
- Processing daily transactions like deposits, withdrawals, transfers
- Providing customer loans, mortgages, lines of credit
- Issuing credit cards and managing credit card accounts
- Managing branches and employee records
- Setting competitive interest rates for deposits and lending

The database we created is a simplified version of Scotiabank where we support the business processes pointed out above. Customers can get loans once they have an account set up. We record their debit and credit card details as well as any transactions from the account. Moreover, customers can open their own investment accounts with three products GIC, Mutual Funds, and Cryptocurrencies. We also have the administrative side of Scotiabank in this database: we have details of branches, account managers, branch managers, and other employees. With all of this data, we created reports with Power BI for various business KPIs such as branch performance and investment portfolio of the whole bank. To support this, we created four categories of users who have specific access to the databases such as HR is limited to employee data. We rigorously tested the database and constraints as well.

The Relational Model

Management and Oversight Relationships

Rather than create entities for the relationships between employees, their branches, their supervision, and the accounts managed, we operationalized foreign keys to represent these relationships. For instance, rather than a "Managed_by" entity to connect an Account to its Branch and Account Manager, we insert the account manager and the branch identification numbers as foreign keys. This relationship is continuous, and there is no additional data to the relationship itself in our scope. So, we represent it through foreign keys.

Investment Types

For investments, we chose to have GIC, Mutual Funds, and Crypto inherit the nested data from Investments and Account, rather than have them as flags. This is because each different kind of investment has additional data unique to the investment type. Also, investment specialists working on a specific kind of investments can then select for those tables easier and faster.

Employments and Age

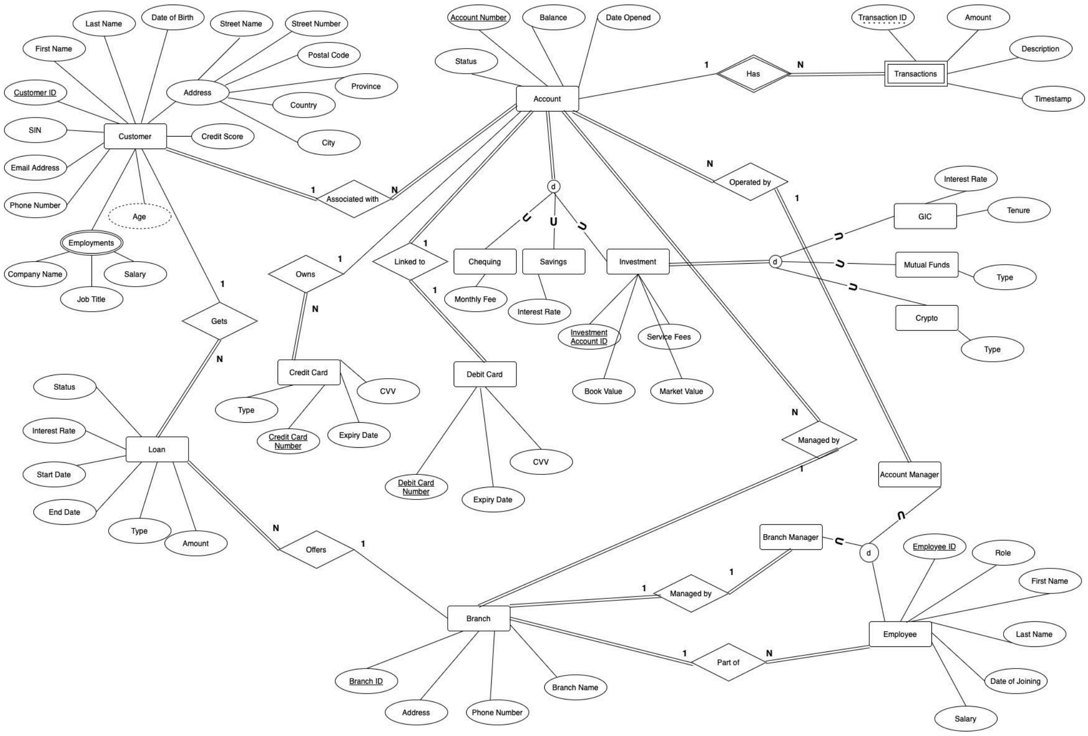
We received feedback on Phase 1 that this attribute isn't explained properly as it is a multi-valued and composite attribute which we would like to clarify. This is because each customer can have multiple employments e.g. full-time and part-time work. Each of these employments would have details like company name, salary, job title. We additionally substituted YearsWorked from Employee and added Age as a calculated attribute in Customer.

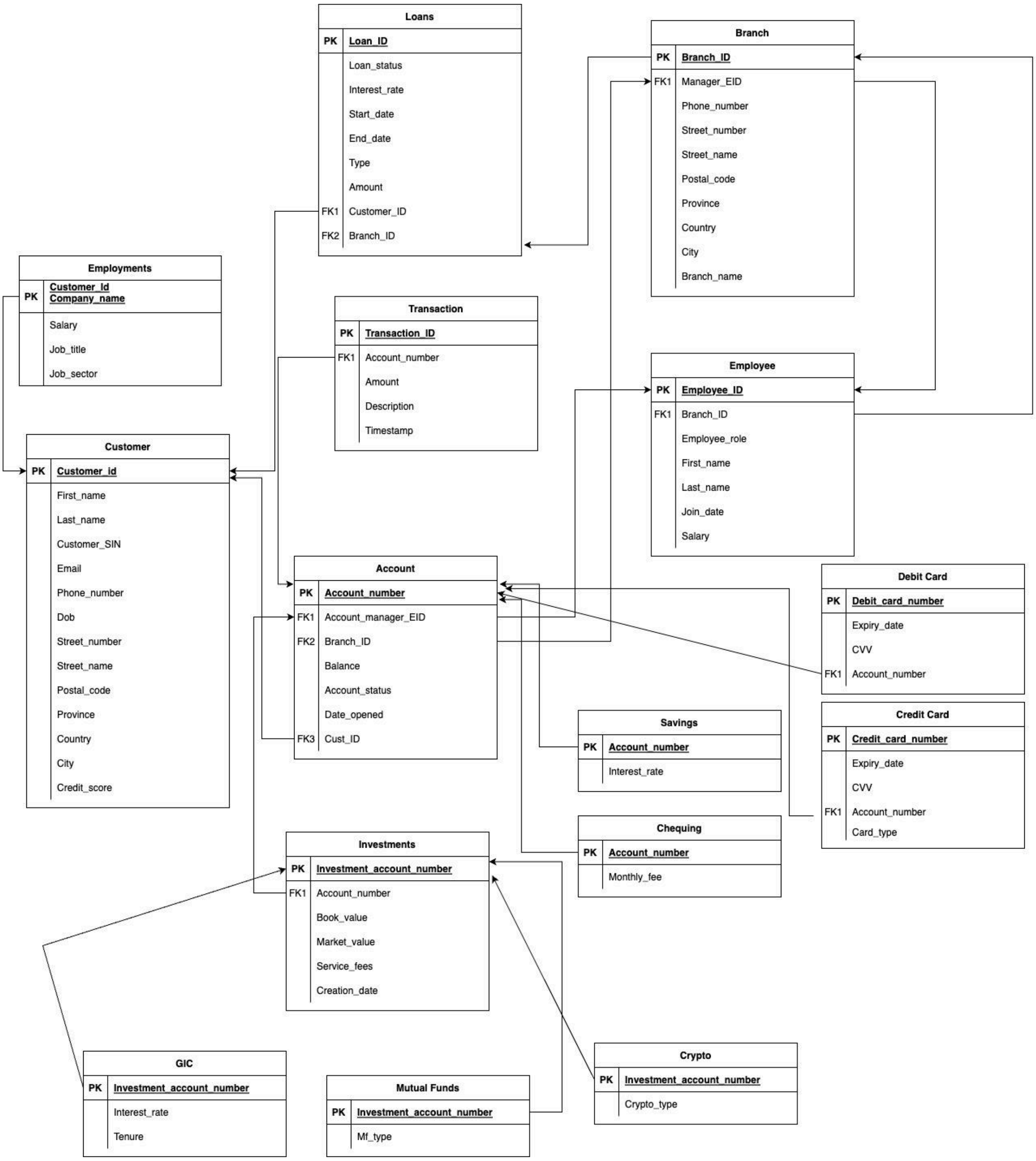
Loan (Gets), Branch (Offers), Transaction (Has), Employee (Part of), Credit Card (Linked To), Account (Associated with) Relationships

For all of these 1-N binary relationships, we used a consistent mapping method for mapping into the relational model: the foreign key approach we discussed in class. This approach made most sense because it is intuitive and it minimizes the number of tables we need to create. Less tables means the database is easier to understand and makes reporting easier.

Account and Debit Card Relationship

This is the other 1:1 relationship that we have. For this we also applied the same foreign key approach as mentioned before with the foreign key for Customer in Account being Not Null and Unique. The goal here again was to minimize the number of relations that we needed to create.





SQL Script for Creating Schema and Tables

```
#CREATE DATABASE AND TABLES QUERIES

CREATE SCHEMA `SCOTIABANK`;

use SCOTIABANK;

#Setting the foreign key checks to 0 so that we can create tables without the
foreign key constraint error.
SET foreign_key_checks = 0;
```

```
CREATE TABLE Customer
(       Customer_ID         CHAR(8)             NOT NULL UNIQUE,
        First_name  VARCHAR(15)         NOT NULL,
        Last_name   VARCHAR(15)         NOT NULL,
        Customer_SIN                CHAR(8)             NOT NULL UNIQUE,
        Email  VARCHAR(30)              NOT NULL UNIQUE,
        DoB             DATE            NOT NULL,
  Street_number     int         NOT NULL,
  Street_name       VARCHAR (30)        NOT NULL,
  Postal_code       CHAR (6) NOT NULL,
  Province   CHAR (2)       NOT NULL,
  Country           CHAR (2)        NOT NULL,
  City VARCHAR (30)         NOT NULL,
  Credit_score              CHAR(3)         NOT NULL CHECK (300 < Credit_score <
850),
        PRIMARY KEY (Customer_ID)
);

CREATE TABLE Employments
(       Company_name     VARCHAR(30) NOT NULL,
        Customer_ID                 CHAR(8) NOT NULL UNIQUE,
  Salary        FLOAT NOT NULL CHECK (Salary > 0),
  Job_title     VARCHAR(20),
  Job_sector VARCHAR(20),
  PRIMARY KEY (Customer_ID, Company_name),
  FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);

CREATE TABLE Account
(       Account_number   CHAR(8) NOT NULL UNIQUE,
        Balance                 INT NOT NULL,
  Acc_status          BOOL,
  Date_opened         DATE,
  Account_manager_EID CHAR(8) NOT NULL,
  Branch_IDCHAR(8)                  NOT NULL,
  Cust_ID CHAR (8) NOT NULL UNIQUE,
  PRIMARY KEY (Account_number),
  FOREIGN KEY (Account_manager_EID) REFERENCES
Employee(Employee_ID),
  FOREIGN KEY (Branch_ID) REFERENCES Branch(Branch_ID),
  FOREIGN KEY (Cust_ID) REFERENCES Customer(Customer_ID)
);

CREATE TABLE Branch
(       Branch_ID   CHAR(8) NOT NULL UNIQUE,
        Address                 VARCHAR(70),
  Street_number     INT     NOT NULL,
  Street_name       VARCHAR (30)        NOT NULL,
  Postal_code       CHAR (6) NOT NULL,
  Province    CHAR (2)      NOT NULL,
  Country              CHAR (2)      NOT NULL,
  City VARCHAR (30)         NOT NULL,
  Phone_number CHAR(10)        NOT NULL,
  Manager_EID              CHAR(8)       UNIQUE,
  PRIMARY KEY (Branch_ID),
  FOREIGN KEY (Manager_EID) REFERENCES Employee(Employee_ID)
);

CREATE TABLE Employee
(        Employee_ID          CHAR(8) NOT NULL UNIQUE,
         Employee_role       VARCHAR(20)          NOT NULL,
  First_name          VARCHAR(20) NOT NULL,
  Last_name           VARCHAR(20)          NOT NULL,
  Join_date   DATE  NOT NULL,
  Salary         FLOAT NOT NULL CHECK (Salary > 0),
  Branch_IDCHAR(8)       NOT NULL,
  PRIMARY KEY (Employee_ID),
  FOREIGN KEY (Branch_ID) REFERENCES Branch(Branch_ID)
);


CREATE TABLE Transactions
(        Transaction_ID      CHAR(8) NOT NULL UNIQUE,
         Amount         INT NOT NULL,
  Transaction_description                    VARCHAR(50),
  Time_stamp         DATETIME NOT NULL,
  Account_number           CHAR(8)       NOT NULL,
  PRIMARY KEY (Transaction_ID),
  FOREIGN KEY (Account_number) REFERENCES
Account(Account_number)
);


CREATE TABLE Loans
(        Loan_ID        CHAR(8) NOT NULL UNIQUE,
         Loan_status BOOL  NOT NULL,
  Interest_rate        FLOAT NOT NULL CHECK(Interest_rate>=0),
  Start_date DATE  NOT NULL,
  End_date   DATE  NOT NULL,
  Loan_type VARCHAR(15)          NOT NULL,
  Amount INT         NOT NULL CHECK(Amount>0),
  Customer_ID              CHAR(8)              NOT NULL UNIQUE,
  Branch_ID          CHAR(8)              NOT NULL,
  PRIMARY KEY (Loan_ID),
  FOREIGN KEY (Branch_ID) REFERENCES Branch(Branch_ID),
  FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);
```

```
CREATE TABLE Chequing
(       Account_number CHAR(8)      NOT NULL UNIQUE,
        Monthly_fee FLOAT NOT NULL CHECK(Monthly_fee >=0),
  PRIMARY KEY (Account_number),
  FOREIGN KEY (Account_number) REFERENCES    Account(Account_number)
);

CREATE TABLE Savings
(       Account_number CHAR(8)      NOT NULL UNIQUE,
        Interest_rate FLOAT NOT NULL CHECK(Interest_rate>=0),
  PRIMARY KEY (Account_number),
  FOREIGN KEY (Account_number) REFERENCES    Account(Account_number)
);

CREATE TABLE Credit_Card
(
        Credit_card_number CHAR(16) NOT NULL UNIQUE,
  CVV CHAR(3) NOT NULL,
  Expiry_date DATE NOT NULL,
  Card_type VARCHAR(20) NOT NULL,
  Account_number CHAR(8)    NOT NULL,
  PRIMARY KEY (Credit_card_number),
  FOREIGN KEY (Account_number) REFERENCES    Account(Account_number)

);

CREATE TABLE Debit_Card
(
        Debit_card_number CHAR(16) NOT NULL UNIQUE,
  CVV CHAR(3) NOT NULL,
  Expiry_date DATE NOT NULL,
  Account_number CHAR(8)    NOT NULL,
  PRIMARY KEY (Debit_card_number),
  FOREIGN KEY (Account_number) REFERENCES    Account(Account_number)

);

CREATE TABLE Investments
(
        Investment_account_number CHAR(8) NOT NULL UNIQUE,
  Creation_date date NOT NULL,
  Book_value FLOAT NOT NULL CHECK(Book_value>0),
  Market_value FLOAT NOT NULL,
  Service_fees FLOAT NOT NULL DEFAULT (10) CHECK(Service_fees>=0),
  Account_number CHAR(8) NOT NULL,
  FOREIGN KEY (Account_number) REFERENCES    Account(Account_number)
);

CREATE TABLE GIC
```

```
(
	Investment_account_number CHAR(8) NOT NULL UNIQUE,
   Interest_rate FLOAT NOT NULL CHECK (Interest_rate>=0),
   Tenure INT NOT NULL,
   FOREIGN KEY (Investment_account_number) REFERENCES
Investments(Investment_account_number)
);

CREATE TABLE Mutual_Funds
(
	Investment_account_number CHAR(8) NOT NULL UNIQUE,
   Mf_type VARCHAR(20) NOT NULL,
   FOREIGN KEY (Investment_account_number) REFERENCES
Investments(Investment_account_number)
);

CREATE TABLE Crypto
(
	Investment_account_number CHAR(8) NOT NULL UNIQUE,
   Crypto_type VARCHAR(20) NOT NULL,
   FOREIGN KEY (Investment_account_number) REFERENCES
Investments(Investment_account_number)
);

SET foreign_key_checks = 1;
```

Data dictionary

Table Name	Attribute Name	Data Type	Constraints	Meaning/Purpose	Sample Example
Customer	Customer_ID	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each customer	'C0000001'
	First_name	VARCHAR(15)	NOT NULL	Customer's first name	'John'
	Last_name	VARCHAR(15)	NOT NULL	Customer's last name	'Doe'

	Customer_SIN	CHAR(8)	NOT NULL, UNIQUE	Customer's Social Insurance Number	'12345678'
	Email	VARCHAR(30)	NOT NULL, UNIQUE	Customer's email address	'john@example.com'
	DoB	DATE	NOT NULL	Customer's date of birth	'1990-01-01'
	Street_number	INT	NOT NULL	Customer's street number	123
	Street_name	VARCHAR(30)	NOT NULL	Customer's street name	'Main St'
	Postal_code	CHAR(6)	NOT NULL	Customer's postal code	'A1A1A1'
	Province	CHAR(2)	NOT NULL	Customer's province	'ON'
	Country	CHAR(2)	NOT NULL	Customer's country	'CA'
	City	VARCHAR(30)	NOT NULL	Customer's city	'Toronto'
	Age	Derived	NOT NULL	Derived attribute, calculated via query	‘34’
	Credit_score	CHAR(3)	NOT NULL, NOT NULL CHECK(300 < Credit_score < 850)	Customer's credit score	'750'
Employments	Company_name	VARCHAR(30)	NOT NULL	Name of the company where the customer is employed	'ABC Inc.'
	Customer_ID	CHAR(8)	NOT NULL, FOREIGN KEY, PRIMARY KEY	Unique identifier for each customer	'C0000001'
	Salary	FLOAT	NOT NULL, CHECK (Salary > 0)	Customer's salary	50000
	Job_title	VARCHAR(20)	NOT NULL	Customer's job title	'Manager'
	Job_sector	VARCHAR(20)	NOT NULL, ON UPDATE CASCADE	Customer’s job sector	‘Finance’
Account	Account_number	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each account	'A0000001'

	Balance	INT	NOT NULL	Account balance	10000
	Acc_status	BOOL		Account status (active or inactive)	TRUE
	Date_opened	DATE		Date when the account was opened	'2022-01-01'
	Account_manager_EID	CHAR(8)	NOT NULL, FOREIGN KEY	Employee ID of the account manager	'E0000001'
	Branch_ID	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for each branch	'B0000001'
Branch	Branch_ID	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each branch	'B0000001'
	Address	VARCHAR(70)		Branch address	'123 Main St'
	Street_number	INT	NOT NULL	Branch street number	123
	Street_name	VARCHAR(30)	NOT NULL	Branch street name	'Main St'
	Postal_code	CHAR(6)	NOT NULL	Branch postal code	'A1A1A1'
	Province	CHAR(2)	NOT NULL	Branch province	'ON'
	Country	CHAR(2)	NOT NULL	Branch country	'CA'
	City	VARCHAR(30)	NOT NULL	Branch city	'Toronto'
	Phone_number	CHAR(10)	NOT NULL	Branch phone number	'4161234567'
	Manager_EID	CHAR(8)	NOT NULL, FOREIGN KEY, UNIQUE	Employee ID of the branch manager	'E0000001'
Employee	Employee_ID	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each employee	'E0000001'
	Employee_role	VARCHAR(20)	NOT NULL	Employee's role	'Manager'
	First_name	VARCHAR(20)	NOT NULL	Employee's first name	'Jane'
	Last_name	VARCHAR(20)	NOT NULL	Employee's last name	'Smith'

	Join_date	DATE	NOT NULL	Date when the employee joined the company	'2020-01-01'
	Salary	FLOAT	NOT NULL, CHECK (Salary > 0)	Employee's salary	75000.00
	Branch_ID	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for the branch where the employee works	'B0000001'
Transactions	Transaction_ID	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each transaction	'T0000001'
	Amount	INT	NOT NULL	Transaction amount	500
	Transaction_description	VARCHAR(50)		Description of the transaction	'Grocery'
	Time_stamp	DATETIME	NOT NULL	Date and time when the transaction occurred	'2023-04-01 10:30:00'
	Account_number	CHAR(8)	NOT NULL, FOREIGN KEY	Account number associated with the transaction	'A0000001'
Loans	Loan_ID	CHAR(8)	NOT NULL, PRIMARY KEY	Unique identifier for each loan	'L0000001'
	Loan_status	BOOL	NOT NULL	Loan status (active or inactive)	TRUE
	Interest_rate	VARCHAR(10)	NOT NULL, CHECK(Interest_rate >= 0)	Interest rate of the loan	'5.5%'
	Start_date	DATE	NOT NULL	Date when the loan started	'2023-01-01'
	End_date	DATE	NOT NULL	Date when the loan ends	'2028-01-01'
	Loan_type	VARCHAR(15)	NOT NULL	Type of loan (e.g., personal, business)	'Personal'
	Amount	INT	NOT NULL, CHECK(Amount>0)	Loan amount	50000
	Customer_ID	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for the customer associated with the loan	'C0000001'

	Branch_ID	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for the branch associated with the loan	'B0000001'
Chequing	Account_number	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY	Account number associated with the chequing account	'A0000001'
	Monthly_fee	FLOAT	NOT NULL, CHECK(Monthly_fee >=0)	Monthly fee for the chequing account	10.00
Savings	Account_number	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY	Account number associated with the savings account	'A0000002'
	Interest_rate	FLOAT	NOT NULL, CHECK(Interest_rate>=0)	Interest rate for the savings account	0.05
Credit_Card	Credit_card_number	CHAR(16)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each credit card	'1234567890123456'
	CVV	CHAR(3)	NOT NULL	Card Verification Value (CVV) for the credit card	'123'
	Expiry_date	DATE	NOT NULL	Expiry date of the credit card	'2025-12-31'
	Card_type	VARCHAR(20)	NOT NULL	Type of credit card (e.g., Visa, Mastercard)	'Visa'
	Account_number	CHAR(8)	NOT NULL, FOREIGN KEY	Account number associated with the credit card	'A0000001'
Investments	Investment_account_number	CHAR(8)	NOT NULL, UNIQUE, PRIMARY KEY	Unique identifier for each investment account	'I0000001'
	Book_value	FLOAT	NOT NULL, CHECK(Book_value>0)	Book value of the investment	10000.00
	Market_value	FLOAT	NOT NULL	Market value of the investment	12000.00
	Service_fees	FLOAT	NOT NULL, DEFAULT(10), CHECK(Service_fees>=0)	Service fees associated with the investment	100.00

	Account_number	CHAR(8)	NOT NULL, FOREIGN KEY	Account number associated with the investment	'A0000001'
	Creation_date	DATE	NOT NULL	The date the investment account opened	'2025-12-31'
GIC	Investment_account_number	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for each GIC investment account	'I0000002'
	Interest_rate	FLOAT	NOT NULL, CHECK(Interest_rate >= 0)	Interest rate for the GIC investment	0.03
	Tenure	INT	NOT NULL	Tenure of the GIC investment (in months)	12
Mutual_Funds	Investment_account_number	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for each mutual fund investment account	'I0000003'
	Mf_type	VARCHAR(20)	NOT NULL	Type of mutual fund (e.g., equity, bond)	'Equity'
Crypto	Investment_account_number	CHAR(8)	NOT NULL, FOREIGN KEY	Unique identifier for each crypto investment account	'I0000004'
	Crypto_type	VARCHAR(20)	NOT NULL	Type of cryptocurrency (e.g., Bitcoin, Ethereum)	'Bitcoin'

Test Cases

Test Cases	
Account Table	
This test case helps check the referential integrity where a non-existent Employee ID from Employee table is being referenced in the insert statement for Account table where Account_Manager_EID=210 .	INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status, Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES ('10009985', '5000', '1', '2022-08-14', '210', '2', 'Z9CV3B1K');
Customer Table	
Test Case Purpose	**Statements**
Checks whether the system properly handles and rejects a customer record that	INSERT INTO SCOTIABANK.Customer (Last_name, Customer_SIN, Email, DoB,

includes invalid Province and Country codes, which should not exist in the database's geographical data constraints.	Street_number, Street_name, Postal_code, Province, Country, City, Credit_score) VALUES ('Johnson', '234567890', 'emily.johnson@example.com', '1985-09-09', '789', 'Elm Street', 'C7D8E9', 'O', 'C', 'Somecity', '650');

This test case checks that the system enforces the validation rule for Credit_score being within the range of 300 to 850. It tries to insert a new customer record with a Credit_score outside this valid range, which should be rejected. This INSERT will be rejected because it does not meet the minimum limit	INSERT INTO SCOTIABANK.Customer (Last_name, Customer_SIN, Email, DoB, Street_number, Street_name, Postal_code, Province, Country, City, Credit_score)
	VALUES ('Doe', '987654321', 'jane.doe@example.com', '1995-05-05', '123', 'Main Street', 'A1A1A1', 'ON', 'CA', 'Anytown', '250');

For this test case, the Credit_score value '900' exceeds the maximum allowable score of 850. This should trigger a rejection by the database system.	INSERT INTO SCOTIABANK.Customer (Last_name, Customer_SIN, Email, DoB, Street_number, Street_name, Postal_code, Province, Country, City, Credit_score)
	VALUES ('Smith', '123456789', 'john.smith@example.com', '1980-01-01', '456', 'Oak Avenue', 'B2B2B2', 'BC', 'CA', 'Othertown', '900');

Checks whether the correct data is added to each of the columns with the correct query	SELECT * FROM SCOTIABANK.Customer;
	INSERT INTO SCOTIABANK.Customer (Customer_ID, First_name, Last_name, Customer_SIN, Email, DoB, Street_number, Street_name, Postal_code, Province, Country, City, Credit_score) VALUES ('L3SD9F6G', 'Olivia', 'Hernandez', '56789012', 'olivia.hernandez@hotmail.com', '1979-12-30', '174', 'Birch Avenue', 'S7K2J3', 'SK', 'CA', 'Saskatoon', '615');

Verifies that the DELETE operation correctly removes a specific customer record from the SCOTIABANK.Customer table. The test confirms that the record with Customer_ID equal to 'A9Z5X1R4' is permanently deleted from the database.	DELETE FROM SCOTIABANK.Customer
	WHERE Customer_ID = 'A9Z5X1R4';

Employments Table	

This test case attempts to insert a record into the Employments table with a Customer_ID that does not exist in the Customer table, expecting the operation to fail due to a foreign key constraint violation.	INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID, Salary, Job_title)
	VALUES ('Tech Innovations', 'NONEXIST1', 50000, 'Software Developer');

This test case checks whether the system enforces the constraint that the Salary must be greater than zero. An attempt will be made to insert a record with a salary of zero, which should fail if such a constraint is correctly implemented.	INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID, Salary, Job_title)
	VALUES ('Global Tech', 'EXISTCUST2', 0, 'Network Engineer');

Investments Table	

This test case is designed to verify that the database rejects an INSERT operation into the Investments table when the Creation_date is provided in an incorrect format. The database is expected to enforce the correct date format (typically YYYY-MM-DD), and any deviation from this should result in an error and prevent the insertion of the record.	INSERT INTO SCOTIABANK.Investments (Investment_account_number, Book_value, Market_value, Service_fees, Account_number, Creation_date)
	VALUES ('INVTEST01', 12000, 12500, 100, '10008622', 'Hello');

This test case checks the system's response when an attempt is made to insert a record into the Investments table with a Book_value that is not greater than zero. The test is expected to fail because a book value should be a positive number representing the value of the investment.	INSERT INTO SCOTIABANK.Investments (Investment_account_number, Book_value, Market_value, Service_fees, Account_number, Creation_date)
	VALUES ('INVTEST02', '0', '12500', '100', '10008622', '2023-04-01');

This test case will attempt to insert a record into the Investments table with Service_fees set to a value that is not greater than zero. The expectation is for the database to reject this record, as service fees typically cannot be zero or negative.	INSERT INTO SCOTIABANK.Investments (Investment_account_number, Book_value, Market_value, Service_fees, Account_number, Creation_date)
	VALUES ('INVTEST03', '12000', '12500', '-100', '10008622', '2023-04-01');

GIC Table	

This test case ensures that interest rates for investments must be greater than zero. A positive interest rate is standard for investments, and any non-positive value should be considered invalid. The test will try to insert a record with an Interest_rate of zero or a negative number, which should be rejected based on the database constraints.	INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate, Tenure)
	VALUES ('INVNEG', -1, 12);

Mutual Funds Table	
This test case attempts to insert a record into the Mutual_Funds table with an	INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number, Mf_type)

Mf_type value that exceeds the maximum length of 20 characters defined by the VARCHAR(20) data type. This operation should be rejected by the database because it violates the column's length constraint.	VALUES ('MFNEW1234', 'VeryLongMutualFundTypeExceedsLimit');
Crypto Table	

This test case attempts to insert a record into the Crypto table with a Crypto_type value that exceeds the maximum length of 20 characters specified by the VARCHAR(20) data type. This test is expected to fail, as the inserted Crypto_type will not conform to the defined schema constraints.	INSERT INTO SCOTIABANK.Crypto (Investment_account_number, Crypto_type)
	VALUES ('CRYP12345', 'ExcessivelyLongCryptoName');

Debit Card Table	

This test case is designed to verify that the database enforces the character length limit for the CVV field in the Debit_Card table. The CVV is expected to be a 3-digit number, so the test will attempt to insert a record with a CVV number that exceeds this length, which should result in a failed insertion.	INSERT INTO SCOTIABANK.Debit_Card (Debit_card_number, CVV, Expiry_date, Account_number)
	VALUES ('1234567890123456', '12345', '2023-12-01', '10008622');

Credit Card Table	

This test case will attempt to insert a new credit card record into the Credit_Card table with a Credit_card_number that exceeds the maximum length constraint of 16 characters. The SQL database is expected to enforce this constraint and reject the insertion.	INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV, Expiry_date, Card_type, Account_number)
	VALUES ('12345678901234567', '123', '2025-12-31', 'Visa', '10008592');

Savings	

This test case attempts to insert a record into the Savings table with an Account_number that not only exceeds the maximum length of 8 characters but also does not match any existing Account_number in the Account table. This operation should be rejected by the database on two counts: the length of the Account_number field and the violation of a foreign key constraint, assuming there is one referencing the Account table.	INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate)
	VALUES ('123456789', 1.5);

This test case ensures that negative interest rates are not allowed in the Savings table. An attempt will be made to	INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate)
	VALUES ('10008612', '-0.5');

insert a record with an Interest_rate less than 0, expecting the operation to be rejected because interest rates should logically be non-negative values.	
Transactions	

This test case aims to validate that the Amount column in the Transactions table only accepts integer values. The test will attempt to insert a record with a non-integer value for Amount, which should be rejected by the SQL database engine.	INSERT INTO SCOTIABANK.Transactions (Transaction_ID, Amount, Transaction_description, Time_stamp, Account_number)
	VALUES ('TXN99999', 'abc', 'Test Invalid Amount', '2023-03-19 10:00:00', '10008592');

Loan Table	

This test case ensures that the Loans table does not accept negative values for Interest_rate. It will attempt to insert a loan record with a negative interest rate, which should be rejected by the database's validation rules or constraints.	INSERT INTO SCOTIABANK.Loans (Loan_ID, Loan_status, Interest_rate, Start_date, End_date, Loan_type, Amount, Customer_ID, Branch_ID)
	VALUES ('6001', '1', '-5', '2023-04-01', '2028-04-01', 'Personal loan', '25000', 'B5VN1H3G', '1');

This test case checks whether the Loans table rejects records where Amount is less than or equal to zero. A loan with a zero or negative amount does not make sense in a practical scenario, and such entries should be prevented by the database.	INSERT INTO SCOTIABANK.Loans (Loan_ID, Loan_status, Interest_rate, Start_date, End_date, Loan_type, Amount, Customer_ID, Branch_ID)
	VALUES ('6002', '1', '4.5', '2023-04-01', '2028-04-01', 'Personal loan', '0', 'D4TM5H2S', '2');

Chequing Table	

This test case aims to ensure that the Monthly_fee in the Chequing table cannot be zero or negative, reflecting that an account must have a positive monthly fee associated with it. The test will try to insert a record with a Monthly_fee of zero or less, expecting the database to reject the entry due to a constraint on this column.	INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee)
	VALUES ('10008612', '-10');

Key Reports

In order to stay on top of business processes, we have designed 3 dashboards which will help us provide a comprehensive evaluation of the business. Each report utilizes both numerical and categorical data to provide insights into different aspects of the bank's operations, with visualizations making the data accessible for quick interpretation and decision-making. They can be used by the management team to gauge performance, inform strategic decisions, optimize operations, and tailor services to better meet customer needs.

We took a bottom-up approach towards designing the Key Reports segment; we identified the insights which would prove to be the most beneficial to the bank and worked on generating queries which would help us arrive at those results. Further, we used Power BI to visualize our findings.

1. Investment Portfolio Report:

We developed the Investment portfolio to give an overview of the investments made by the bank's customers and analyze the portfolio's composition, growth, and distribution across different demographics and job sectors. This report provides a detailed analysis of the investment preferences and behaviors of clients based on their salary, age, and job sector, which can inform targeted marketing strategies, future business plans and actions and customer preferences. The growth trend of investments over the year could be useful to evaluate the performance of investment products and strategize for the future.

Tables and attributes used to identify the key parameters:

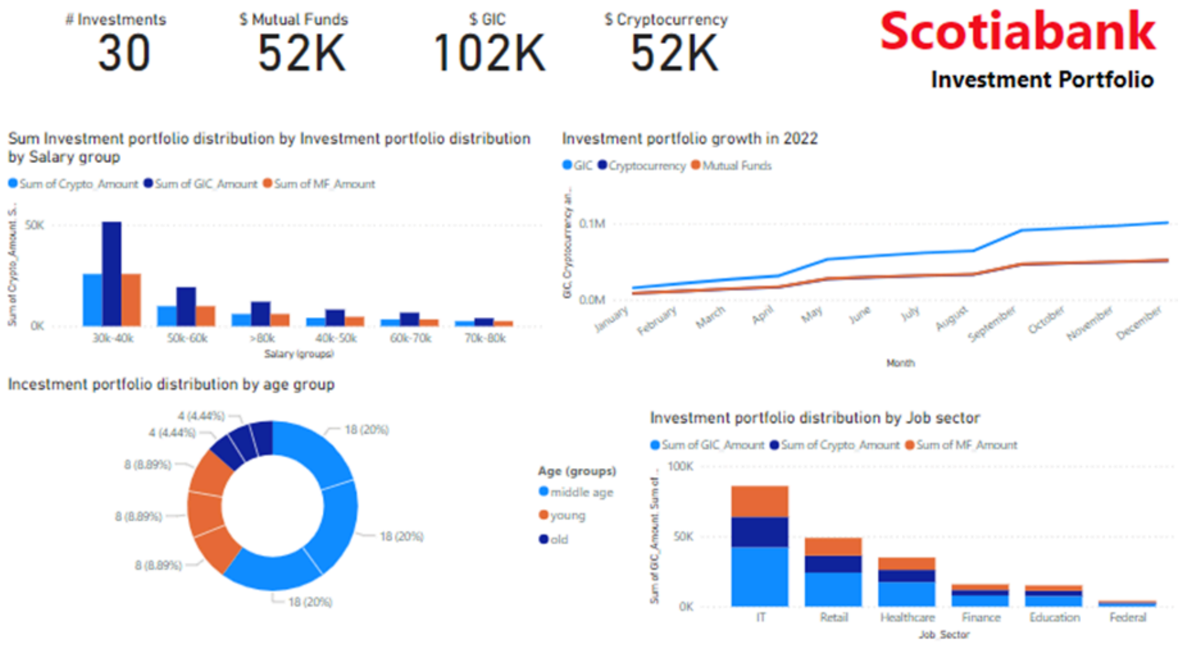
- Investments table: Investment_account_number, Book_value, Account_number, Creation_date
- Account table: Account_number, Cust_ID
- Employments table: Customer_ID, Job_Sector
- Customer table: Customer_ID, DoB
- Crypto table: Investment_account_number, Crypto_amount
- GIC table: Investment_account_number, GIC_Amount
- Mutual_Funds table: Investment_account_number, MF_Amount

SQL code for Investment Portfolio

A consolidated table to compare each investment type for different types of users categorized by age, job_sector ,salary

```
SELECT i.Book_value, i.Creation_date, i.Account_number, e.Salary,
e.Job_Sector, FLOOR(DATEDIFF(CURRENT_DATE, cust.DoB) / 365.25) AS Age,
    CASE WHEN c.Investment_account_number IS NOT NULL THEN 'Crypto'
       WHEN g.Investment_account_number IS NOT NULL THEN 'GIC'
```

```
        WHEN m.Investment_account_number IS NOT NULL THEN 'Mutual
Funds'
        ELSE 'Unknown' END AS Investment_Type
FROM Investments i
LEFT JOIN Crypto c ON i.Investment_account_number =
c.Investment_account_number
LEFT JOIN GIC g ON i.Investment_account_number =
g.Investment_account_number
LEFT JOIN Mutual_Funds m ON i.Investment_account_number =
m.Investment_account_number
INNER JOIN Account ON Account.Account_number = i.Account_number
INNER JOIN Employments e ON Account.Cust_ID = e.Customer_ID
INNER JOIN Customer cust ON Account.Cust_ID= cust.Customer_ID;
```



# Investments
30
$ Mutual Funds
52K
$ GIC
102K
$ Cryptocurrency
52K
Scotiabank
Investment Portfolio
Sum Investment portfolio distribution by Investment portfolio distribution by Salary group
● Sum of Crypto_Amount ● Sum of GIC_Amount ● Sum of MF_Amount
Sum of Crypto_Amount, S..
50K
0K
30k-40k
50k-60k
>80k
40k-50k
60k-70k
70k-80k
Salary (groups)
Investment portfolio growth in 2022
● GIC ● Cryptocurrency ● Mutual Funds
GIC, Cryptocurrency an...
0.1M
0.0M
January
February
March
April
May
June
July
August
September
October
November
December
Month
Incestment portfolio distribution by age group
4 (4.44%)
4 (4.44%)
8 (8.89%)
8 (8.89%)
8 (8.89%)
18 (20%)
18 (20%)
18 (20%)
Age (groups)
● middle age
● young
● old
Investment portfolio distribution by Job sector
● Sum of GIC_Amount ● Sum of Crypto_Amount ● Sum of MF_Amount
Sum of GIC_Amount, Sum of ...
100K
50K
0K
IT
Retail
Healthcare
Finance
Education
Federal
Job_Sector

2. Branch KPIs Report:

The branch key performance indicator report would help us evaluate the performance of bank branches in terms of transaction volume, new accounts, and human resource growth as well as to provide a geographical overview of customer density per branch. For the purpose of this report, we have assumed Scotiabank to have 2 branches and the report has been designed to conduct a comparative study with respect to the 2 branches. This report can be used to inform management about how each branch is performing in terms of business growth and human resources, guiding decisions on where to allocate resources or focus efforts for expansion. The map visualization provides a quick glance at market penetration per branch and could be instrumental for strategic decisions on where to establish new branches or enhance marketing campaigns.

Tables and attributes used to identify the key parameters:
- Branch table: Branch_ID, Postal_code, Country, City
- Account table: Branch_ID, Account_number, Date_opened
- Employee table: Employee_ID, Join_date
- Investment table: Account_number, Book_value
- Credit_card table: Credit_card_number, Account_number

SQL code for Branch KPIs

Retrieve the accounts opened over the last year in both branches

```
SELECT Account_number, Branch_ID, Date_opened from Account;
```

Number of Employees per branch

```
SELECT COUNT(Employee_ID), Branch_ID
from Employee
group by Branch_ID;
```

Employee hiring trajectory- growth per branch

```
Select Join_date, employee_ID, Branch_ID
from Employee;
```

Customer density map per branch

```
SELECT Customer.Customer_ID, Customer.City,
Customer.Postal_code,Account.Branch_ID
from  Account, Customer
where Customer.Customer_ID=Account.Cust_ID;
```

Which branch has the most investments?

```
SELECT SUM(Book_value), Branch_ID
from Investments, Account
where Investments.Account_number=Account.Account_number
group by Branch_ID;
```

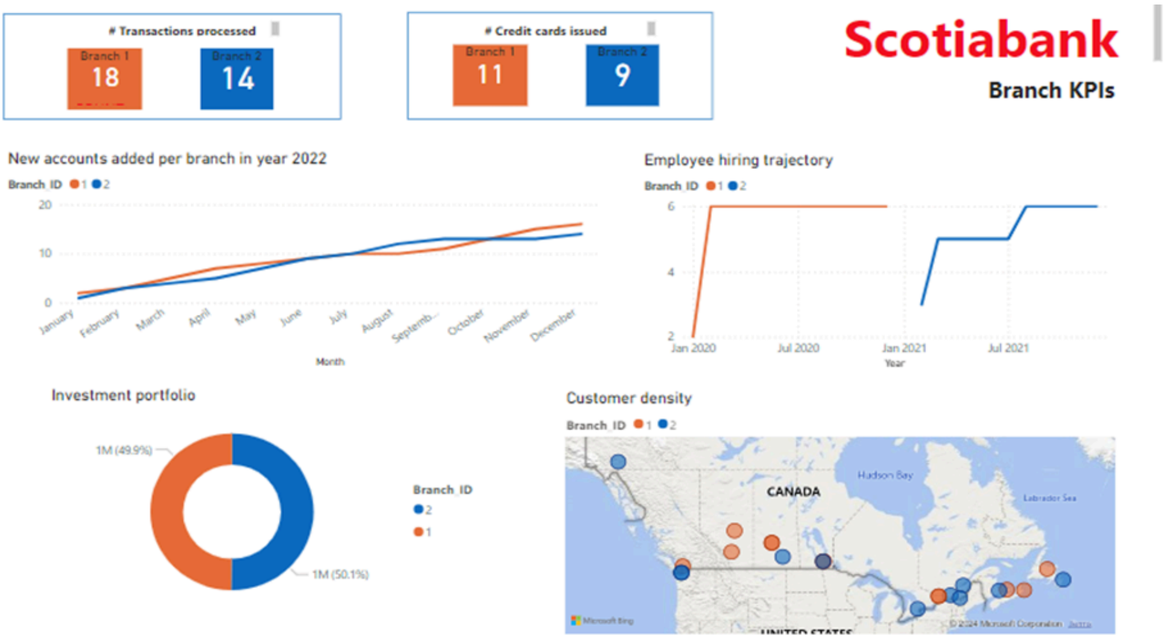
Which branch is selling more credit cards?

```
SELECT COUNT(Credit_card_number), branch_id
from Account, Credit_Card
where Account.Account_number=Credit_Card.Account_number
group by branch_id;
```

Number of Transactions per branch

```
Select COUNT(Transaction_id), Branch_ID
```

```
from Transactions, Account
where Account.Account_number=Transactions.Account_number
group by Branch_ID;
```



3. Customer Portfolio Report:

The purpose of the Customer portfolio is to analyze the customer base in terms of loan issuance, credit card issuance, and client demographics. The distribution of loans, credit cards, and client demographics could help identify which sectors and age groups are more engaged with the bank's products, assisting in customer service improvement and product development. Credit score distribution offers a risk assessment of the clientele that could help in managing credit risk and setting lending policies.

Tables and attributes used to identify the key parameters:

- Account table: Account_number, Balance, Cust_ID

- Loans table: Customer_ID, Loan_ID

- Customer table: Customer_ID, Credit_score, DoB, Credit_score

- Credit_Card table: Account_number, Credit_card_number

SQL code for Customer Portfolio

Customers who have taken loans and their job sectors

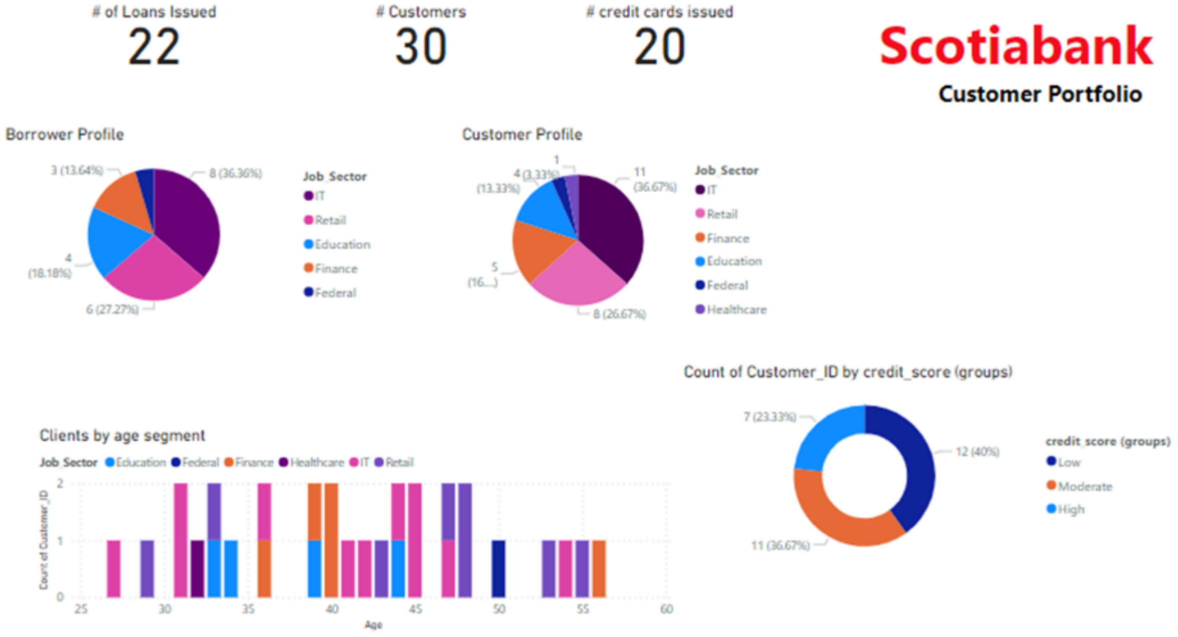
```sql
select Job_Sector, Customer.Customer_ID, Loan_ID
from Customer,Employments,Loans
where Customer.Customer_ID=Employments.Customer_ID and
Customer.Customer_ID=Loans.Customer_ID;
```

Customer credit score, balance, job sector

```sql
select Job_Sector, Customer.Customer_ID, credit_score,  balance
from Customer,Employments, Account
where Customer.Customer_ID=Employments.Customer_ID and
Customer.Customer_ID=Account.Cust_ID;
```

Retrieve Customers by age group

```sql
SELECT Customer.Customer_ID, FLOOR(DATEDIFF(CURRENT_DATE, DoB) /
365.25) AS Age, Job_Sector
FROM  Customer, Employments where
Customer.Customer_ID=Employments.Customer_ID;
```



Access Control

By restricting access based on user categories and their specific roles, the banking system ensures that users have access to only the necessary data and functionalities required for their job responsibilities. This approach enhances data security, maintains confidentiality, and prevents unauthorized access or modifications to sensitive information. It allows each user category to focus on their specific tasks while ensuring the overall integrity and security of the banking system.

The Branch Manager plays a crucial role in overseeing the operations of a specific branch. They require access to all customer information, including personal details, account information, and employment history, to make informed decisions and provide comprehensive support to customers. Additionally, they need access to branch-related information, such as branch ID, address, phone number, and branch name, to manage the branch effectively. The Branch Manager is also responsible for managing employees within their branch, so they have access to employee records, including employee ID, name, date of joining, salary, and role. To fulfill their responsibilities, the Branch Manager is granted permissions to perform SELECT, INSERT, UPDATE, and DELETE operations on the Customer, Account, Debit_card, Credit_card, Loans, Transactions, Employee, and branch tables. However, their access to the investment table is limited to SELECT operations only, as they may not need to modify investment-related data directly.

The Account Manager is responsible for managing specific customer accounts. This includes access to debit card and credit card information associated with the managed accounts. The Account Manager also needs the ability to view and update customer employment history to maintain accurate records. They have access to loan information for the customers they manage to assist with loan-related queries and processes. To perform their duties effectively, the Account Manager is granted permissions to perform SELECT, INSERT, and UPDATE operations on the Customer, Account, Debit_card, Credit_card, and Loans tables, but only for the accounts they manage. They have SELECT access to the Transactions and investment tables, limited to the accounts they handle, to view relevant information without the ability to modify it.

The Investment Specialist focuses on managing and advising on customer investments. They require access to customer investment information, including investment account ID, amount, and service fees, to provide accurate and timely investment advice. The Investment Specialist also needs access to specific investment types, such as GIC, Mutual funds, and Crypto, along with their associated details, to offer specialized guidance and support. To carry out their responsibilities effectively, the Investment Specialist is granted permissions to perform SELECT, INSERT, UPDATE, and DELETE operations on the investment table. They have SELECT access to the Customer and Account tables, limited to the necessary information for investment purposes, to ensure they can view relevant customer and account details without the ability to modify them, as this is the role of the Account Manager.

The HR (Human Resources) department is responsible for managing employee-related matters across all branches. They require access to employee records, including employee ID, first name, last name, date of joining, salary, and role, to maintain accurate and up-to-date employee information. The HR department needs the ability to view and manage employee information across all branches to facilitate HR processes, such as hiring, performance management, and payroll. They also require access to branch information for HR-related purposes, such as tracking employee assignments and managing branch-specific HR tasks. To fulfill their duties, the HR department is granted permissions to perform SELECT, INSERT, UPDATE, and DELETE operations on the Employee table. They have SELECT access to the branch table to view branch-related information without the ability to modify it.

Please note the SQL queries for Access Control are part of the Script1 file at the end.

Role	**Permissions**
Branch Manager	● SELECT, INSERT, UPDATE, DELETE on Customer, Account, Debit_card, Credit_card, Loans, Transactions, Employee, and branch tables. ● SELECT on Investments table.
Account Manager	● SELECT, INSERT, UPDATE on Customer, Account, Debit_card, Credit_card, and Loans tables (limited to the Accounts they manage). ● SELECT on Transactions and investment tables (limited to the Accounts they manage).
Investment Specialist	● SELECT, INSERT, UPDATE, DELETE on Investments table. ● SELECT on Customer and Account tables (limited to the necessary information for investment purposes).
Human Resources	● SELECT, INSERT, UPDATE, DELETE on Employee table. ● SELECT on Branch table.

Statement of Individual Contributions

Omer Imran and Michaela Drouillard worked on the relational model mapping and the access control together. Fatima Zohra and Bhanvi Gupta worked on the key reports and test cases. Every team member wrote their respective parts in the report and slides. The editing and formatting of both report and slides was done together. Please note that we used ChatGPT for writing the queries for inserting the data.

Appendix

Insert Statements for all Tables

CUSTOMER TABLE

```
INSERT INTO SCOTIABANK.Customer (Customer_ID, First_name, Last_name,
Customer_SIN, Email, DoB, Street_number, Street_name, Postal_code,
Province, Country, City, Credit_score)
VALUES
('A7BM2K4J', 'Charlotte', 'Lee', '09123456', 'charlotte.lee@gmail.com',
'1979-10-13', 866, 'Elm Circle', 'T2P5M5', 'AB', 'CA', 'Calgary', 618),
('B5VN1H3G', 'Benjamin', 'Walker', '19234567',
'benjamin.walker@hotmail.com', '1967-04-28', 349, 'Oak Road', 'R2C0A1',
'MB', 'CA', 'Winnipeg', 634),
('D4TM5H2S', 'Alexander', 'Young', '31456789', 'alexander.young@gmail.com',
'1994-12-18', 450, 'Spruce Street', 'B3J2N5', 'NS', 'CA', 'Halifax', 705),
('E2UN6J4K', 'Grace', 'Hill', '42567890', 'grace.hill@hotmail.com',
'1978-07-26', 1211, 'Pine Ridge Road', 'R3L0L5', 'MB', 'CA', 'Winnipeg', 650),
('F1VM8L6Q', 'Natalie', 'Ramirez', '53678901', 'natalie.ramirez@gmail.com',
'1988-02-14', 789, 'Maple Drive', 'G1R4P3', 'QC', 'CA', 'Quebec', 680),
('F4JL2D8P', 'Mia', 'Roberts', '12345678', 'mia.roberts@hotmail.com',
'1983-05-24', 129, 'Pine Road', 'H0H0H0', 'QC', 'CA', 'Montreal', 620),
('G6JK8L0P', 'Noah', 'Brown', '23456789', 'noah.brown@gmail.com',
'1989-07-30', 768, 'Maple Avenue', 'B1A1A1', 'NS', 'CA', 'Halifax', 660),
('G9XL3B7D', 'Zachary', 'Baker', '64789012', 'zachary.baker@hotmail.com',
'1990-10-31', 132, 'Beechwood Drive', 'T5J3M7', 'AB', 'CA', 'Edmonton', 665),
('H3FJ5M1R', 'Emma', 'Garcia', '34567890', 'emma.garcia@hotmail.com',
'1978-11-12', 550, 'Elm Road', 'V6B4N5', 'BC', 'CA', 'Vancouver', 590),
('H7YM2K9F', 'Lucas', 'Wright', '75890123', 'lucas.wright@gmail.com',
'1983-03-22', 304, 'Cherry Lane', 'S7N2A8', 'SK', 'CA', 'Saskatoon', 692),
('I4KP9N2U', 'Liam', 'Martinez', '45678901', 'liam.martinez@gmail.com',
'1990-09-05', 324, 'Cedar Lane', 'T5K2L3', 'AB', 'CA', 'Edmonton', 710),
('I5ZN1H8G', 'Hannah', 'Martin', '86901234', 'hannah.martin@hotmail.com',
'1976-11-11', 556, 'Elm Court', 'V8X1H5', 'BC', 'CA', 'Victoria', 730),
('J3WM9L5R', 'Evelyn', 'Clark', '97012345', 'evelyn.clark@gmail.com',
'1975-08-15', 890, 'Willow Avenue', 'A1C5K4', 'NL', 'CA', 'St. Johns', 615),
('K2UN3J7T', 'Gabriel', 'Rodriguez', '08123456',
'gabriel.rodriguez@hotmail.com', '1982-05-09', 1234, 'Cedar Street', 'L4W1A2',
'ON','CA', 'Mississauga', 670),
('L3SD9F6G', 'Olivia', 'Hernandez', '56789012',
'olivia.hernandez@hotmail.com', '1979-12-30', 174, 'Birch Avenue', 'S7K2J3',
'SK', 'CA', 'Saskatoon', 615),
('M9QN4T5R', 'William', 'Lopez', '67890123', 'william.lopez@gmail.com',
'1968-08-15', 800, 'Pine Street', 'R3B2T9', 'MB', 'CA', 'Winnipeg', 640),
('N6YH8V3C', 'Sophia', 'Gonzalez', '78901234',
'sophia.gonzalez@hotmail.com', '1992-03-22', 920, 'Oak Lane', 'C1A5M4', 'PE',
'CA', 'Charlottetown', 698),
('O4LF5G2H', 'Ethan', 'Perez', '89012345', 'ethan.perez@gmail.com',
'1981-06-18', 407, 'Chestnut Road', 'A2A2A2', 'NL', 'CA', 'St. Johns', 587),
('P2TG4S1J', 'Isabella', 'Sanchez', '90123456', 'isabella.sanchez@hotmail.com',
'1974-01-29', 115, 'Spruce Drive', 'X1A1A1', 'NT', 'CA', 'Yellowknife', 672),
('Q1RM3F6B', 'Michael', 'Ramirez', '01234567', 'michael.ramirez@gmail.com',
'1984-10-09', 658, 'Elm Avenue', 'Y1A2B3', 'YT', 'CA', 'Whitehorse', 603),
('R5DV2J0L', 'Ava', 'Anderson', '10234567', 'ava.anderson@hotmail.com',
'1996-04-12', 289, 'Beech Street', 'J1A2B3', 'QC', 'CA', 'Quebec', 720),
('S3HG6K8N', 'Matthew', 'Thomas', '21345678', 'matthew.thomas@gmail.com',
'1975-07-19', 963, 'Cedar Boulevard', 'K2B5M3', 'ON', 'CA', 'Ottawa', 568),
('T7CK1F4M', 'Madison', 'Jackson', '32456789',
'madison.jackson@hotmail.com', '1987-11-23', 812, 'Walnut Street', 'V3H4K5',
'BC', 'CA', 'Vancouver', 610),
('U8EB2L5Q', 'David', 'White', '43567890', 'david.white@gmail.com',
'1980-12-15', 437, 'Pine Ridge', 'M3J8H5', 'ON', 'CA', 'Toronto', 699),
('U8ZD3H5J', 'James', 'Wilson', '87654321', 'james.wilson@gmail.com',
'1971-03-11', 402, 'Oak Street', 'K1A0B1', 'ON', 'CA', 'Ottawa', 580),
('V6JW3H7D', 'Elizabeth', 'Harris', '54678901', 'elizabeth.harris@hotmail.com',
'1993-02-28', 529, 'Maple Lane', 'S4P3Y2', 'SK', 'CA', 'Regina', 582),
('W4KG2F8S', 'Joseph', 'Martinez', '65789012', 'joseph.martinez@gmail.com',
'1969-05-31', 611, 'Birch Street', 'H2Z1A4', 'QC', 'CA', 'Montreal', 638),
('X2HN5L9T', 'Emily', 'Clark', '76890123', 'emily.clark@hotmail.com',
'1976-09-17', 284, 'Chestnut Lane', 'L5N4G2', 'ON', 'CA', 'Mississauga', 605),
('Y1DF4G6U', 'Daniel', 'Rodriguez', '87901234', 'daniel.rodriguez@gmail.com',
'1991-08-20', 730, 'Willow Avenue', 'V8N1A3', 'BC', 'CA', 'Victoria', 689),
('Z9CV3B1K', 'Oliver', 'Lewis', '98012345', 'oliver.lewis@hotmail.com',
'1985-01-05', 159, 'Cedar Path', 'E4P3R6', 'NB', 'CA', 'Fredericton', 672);
```

BRANCH TABLE

```
INSERT INTO SCOTIABANK.Branch (Branch_ID, Address, Street_number,
Street_name, Postal_code, Province, Country, City, Phone_number) VALUES
('1', '3446 YONGE STREET', '3446', 'YONGE STREET', 'M4N2N2', 'ON', 'CA',
'Toronto', '6478569922'),
('2', '100 City Center road', '100', 'City Center road', 'L5B2C9', 'ON', 'CA',
'Toronto', '9851214455');
```

EMPLOYEE TABLE

```
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (101, 'Cashier',
'Ana', 'Reznik', '2020-01-20', 50000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (102, 'Branch
manager', 'Planak', 'Li', '2020-02-16', 80000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (103, 'Personal
banker', 'Paul', 'Chiu', '2020-01-20', 70000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (104, 'Personal
banker', 'Nina', 'Tong', '2020-02-02', 71000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (105, 'Personal
banker', 'Kitty', 'Forman', '2020-02-03', 72000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (106, 'Loan
officer', 'Liam', 'Hunting', '2020-02-05', 75000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (201, 'Cashier',
'Moana', 'Jung', '2021-02-15', 45000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (202, 'Branch
manager', 'Ulong', 'Rank', '2021-02-20', 82000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (203, 'Personal
banker', 'Hank', 'Muller', '2021-02-22', 73000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (204, 'Personal
banker', 'Flora', 'Maine', '2021-03-18', 85000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (205, 'Personal
banker', 'Mark', 'Jacob', '2021-08-30', 80000, 1);
INSERT INTO SCOTIABANK.Employee (Employee_ID, Employee_role,
First_name, Last_name, Join_date, Salary, Branch_ID) VALUES (206, 'Loan
officer', 'Justin', 'Tim', '2021-03-15', 77000, 1);
```

ADD TO BRANCH TABLE

```
UPDATE SCOTIABANK.Branch
SET Manager_EID = '102'
WHERE Branch_ID = '1';
UPDATE SCOTIABANK.Branch
SET Manager_EID = '202'
WHERE Branch_ID = '2';
```

ACCOUNT TABLE

```
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008592', 3000, 1, '2022-01-15', '102', '1', 'A7BM2K4J');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008593', 4000, 1, '2022-02-22', '103', '1', 'B5VN1H3G');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008594', 2500, 1, '2022-03-18', '104', '1', 'D4TM5H2S');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008595', 3500, 1, '2022-04-03', '202', '2', 'E2UN6J4K');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008596', 2800, 1, '2022-05-07', '203', '2', 'F1VM8L6Q');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008597', 3200, 1, '2022-06-14', '204', '2', 'F4JL2D8P');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008598', 3700, 1, '2022-07-20', '105', '1', 'G6JK8L0P');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008599', 2900, 1, '2022-08-25', '205', '2', 'G9XL3B7D');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008600', 4100, 1, '2022-09-09', '102', '1', 'H3FJ5M1R');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008601', 2600, 1, '2022-10-13', '103', '1', 'H7YM2K9F');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008602', 3200, 1, '2022-11-21', '104', '1', 'I4KP9N2U');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008603', 3700, 1, '2022-12-05', '202', '2', 'I5ZN1H8G');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008604', 2900, 1, '2022-01-28', '203', '2', 'J3WM9L5R');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008605', 4100, 1, '2022-02-03', '204', '2', 'K2UN3J7T');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008606', 5600, 1, '2022-03-17', '105', '1', 'L3SD9F6G');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008607', 3200, 1, '2022-04-29', '102', '1', 'M9QN4T5R');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008608', 5700, 1, '2022-05-12', '103', '1', 'N6YH8V3C');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008609', 2900, 1, '2022-06-25', '104', '1', 'O4LF5G2H');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008610', 4100, 1, '2022-07-30', '202', '2', 'P2TG4S1J');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008611', 2600, 1, '2022-08-04', '203', '2', 'Q1RM3F6B');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008612', 50000, 1, '2022-09-19', '204', '2', 'R5DV2J0L');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008613', 3700, 1, '2022-10-22', '105', '1', 'S3HG6K8N');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008614', 2900, 1, '2022-11-30', '102', '1', 'T7CK1F4M');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008615', 4100, 1, '2022-12-10', '103', '1', 'U8EB2L5Q');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008616', 26000, 1, '2022-01-05', '104', '1', 'U8ZD3H5J');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008617', 3200, 1, '2022-02-11', '202', '2', 'V6JW3H7D');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008618', 3700, 1, '2022-03-24', '203', '2', 'W4KG2F8S');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008619', 3000, 1, '2022-04-15', '105', '1', 'X2HN5L9T');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008620', 35000, 1, '2022-05-22', '202', '2', 'Y1DF4G6U');
INSERT INTO SCOTIABANK.Account (Account_number, Balance, Acc_status,
Date_opened, Account_manager_EID, Branch_ID, Cust_ID) VALUES
('10008621', 3200, 1, '2022-06-18', '203', '2', 'Z9CV3B1K');
```

CHEQUEING TABLE

```
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008593', '110');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008594', '120');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
```

```
('10008595', '130');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008596', '140');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008597', '150');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008598', '160');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008599', '170');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008600', '180');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008601', '190');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008602', '200');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008603', '210');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008604', '220');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008605', '230');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008606', '240');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008607', '250');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008608', '260');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008609', '270');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008610', '280');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008611', '290');
INSERT INTO SCOTIABANK.Chequing (Account_number, Monthly_fee) VALUES
('10008612', '300');
```

INVESTMENTS TABLE

```
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00001', 10000.00, 11000.00, 100.00, '10008592', '2022-02-02');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00002', 15000.00, 16500.00, 150.00, '10008593', '2022-02-10');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00003', 20000.00, 22000.00, 200.00, '10008594', '2022-03-05');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00004', 25000.00, 27500.00, 250.00, '10008595', '2022-04-12');
```

```
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00005', 30000.00, 33000.00, 300.00, '10008596', '2022-05-20');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00006', 35000.00, 38500.00, 350.00, '10008597', '2022-06-15');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00007', 40000.00, 44000.00, 400.00, '10008598', '2022-07-08');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00008', 45000.00, 49500.00, 450.00, '10008599', '2022-08-11');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00009', 50000.00, 55000.00, 500.00, '10008600', '2022-09-25');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00010', 55000.00, 60500.00, 550.00, '10008601', '2022-10-17');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00011', 60000.00, 66000.00, 600.00, '10008602', '2022-11-30');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00012', 65000.00, 71500.00, 650.00, '10008603', '2022-01-14');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00013', 70000.00, 77000.00, 700.00, '10008604', '2022-02-03');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00014', 75000.00, 82500.00, 750.00, '10008605', '2022-03-27');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00015', 80000.00, 88000.00, 800.00, '10008606', '2022-04-19');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00016', 85000.00, 93500.00, 850.00, '10008607', '2022-05-08');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00017', 90000.00, 99000.00, 900.00, '10008608', '2022-02-02');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00018', 95000.00, 104500.00, 950.00, '10008609', '2022-02-10');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00019', 100000.00, 110000.00, 1000.00, '10008610',
'2022-03-05');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00020', 105000.00, 115500.00, 1050.00, '10008611', '2022-04-12');
```

```
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00021', 110000.00, 121000.00, 1100.00, '10008612', '2022-05-20');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00022', 115000.00, 126500.00, 1150.00, '10008613', '2022-06-15');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00023', 120000.00, 132000.00, 1200.00, '10008614',
'2022-07-08');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00024', 125000.00, 137500.00, 1250.00, '10008615', '2022-08-11');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00025', 130000.00, 143000.00, 1300.00, '10008616',
'2022-09-25');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00026', 135000.00, 148500.00, 1350.00, '10008617', '2022-10-17');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00027', 140000.00, 154000.00, 1400.00, '10008618',
'2022-11-30');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00028', 145000.00, 159500.00, 1450.00, '10008619',
'2022-01-14');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00029', 150000.00, 165000.00, 1500.00, '10008620',
'2022-02-03');
INSERT INTO SCOTIABANK.Investments (Investment_account_number,
Book_value, Market_value, Service_fees, Account_number, Creation_date)
VALUES ('INV00030', 155000.00, 170500.00, 1550.00, '10008621',
'2022-03-27');
```

CREDIT_CARD TABLE

```
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1000857900000001',
'789', '2025-07-14', 'Visa', '10008597');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1000858900000001',
'654', '2026-02-18', 'American Express', '10008598');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1000859900000002',
'234', '2023-09-23', 'Mastercard', '10008599');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1000860000000001',
'789', '2025-04-05', 'Visa', '10008600');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1000860200000001',
'234', '2023-08-28', 'Mastercard', '10008602');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1000860300000001',
'789', '2025-06-13', 'Visa', '10008603');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('1245365345224287',
'121', '2025-10-28', 'Visa', '10008592');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('2568345123874563',
'456', '2024-08-15', 'Mastercard', '10008593');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('3658741239658741',
'789', '2026-03-20', 'American Express', '10008593');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('3658741239658742',
'234', '2023-03-22', 'Visa', '10008594');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('3658741239658743',
'789', '2025-06-13', 'Mastercard', '10008595');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('3658741239658744',
'456', '2026-03-20', 'Visa', '10008596');
INSERT INTO SCOTIABANK.Credit_Card (Credit_card_number, CVV,
Expiry_date, Card_type, Account_number) VALUES ('3658741239658745',
'234', '2023-11-17', 'American Express', '10008596');
```

CRYPTO TABLE

```
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00001', 'Bitcoin (BTC)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00002', 'Ethereum (ETH)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00003', 'Ripple (XRP)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00004', 'Litecoin (LTC)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00005', 'Cardano (ADA)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00006', 'Polkadot (DOT)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00007', 'Bitcoin (BTC)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00008', 'Ethereum (ETH)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00009', 'Ripple (XRP)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00010', 'Litecoin (LTC)');
INSERT INTO SCOTIABANK.Crypto (Investment_account_number,
Crypto_type) VALUES ('INV00011', 'Cardano (ADA)');
```

DEBIT_CARD TABLE

```
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536566', '213',
'2026-11-07', '10008593');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536567', '214',
'2026-12-08', '10008594');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536568', '215',
'2027-01-09', '10008595');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536569', '216',
'2027-02-10', '10008596');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536570', '217',
'2027-03-11', '10008597');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536571', '218',
'2027-04-12', '10008598');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536572', '219',
'2027-05-13', '10008599');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536573', '220',
'2027-06-14', '10008600');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536574', '221',
'2027-07-15', '10008601');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536575', '222',
'2027-08-16', '10008602');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536576', '223',
'2027-09-17', '10008603');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536577', '224',
'2027-10-18', '10008604');
INSERT INTO `SCOTIABANK`.`Debit_Card` (`Debit_card_number`, `CVV`,
`Expiry_date`, `Account_number`) VALUES ('1212552264536578', '225',
'2027-11-19', '10008605');
```

EMPLOYMENTS TABLE

```
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Acme Corp', 'A7BM2K4J', 75000,
'Analyst', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Globex Inc', 'B5VN1H3G', 32000,
'Consultant', 'Finance');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Initech LLC', 'D4TM5H2S', 48000,
'Sales Assoc.', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Umbrella Corp', 'E2UN6J4K', 58000,
'Proj Manager', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Soylent Corp', 'F1VM8L6Q', 83000,
'Developer', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Vehement Capital', 'F4JL2D8P', 95000,
'Fin Analyst', 'Finance');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Massive Dynamic', 'G6JK8L0P', 40000,
'Scientist', 'Education');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Hooli', 'G9XL3B7D', 69500, 'Mkt
Director', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Pied Piper', 'H3FJ5M1R', 37000, 'Prod
Manager', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Bluth Company', 'H7YM2K9F', 157000,
'CFO', 'Finance');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Sterling Cooper', 'I4KP9N2U', 67500,
'Copywriter', 'Education');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Wayne Enterprises', 'I5ZN1H8G', 85000,
'HR Manager', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Dunder Mifflin', 'J3WM9L5R', 43000,
'Sales Rep.', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Cyberdyne Systems', 'K2UN3J7T',
62000, 'Sys Analyst', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Oscorp Industries', 'L3SD9F6G', 39000,
'Biologist', 'Education');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Wonka Industries', 'M9QN4T5R',
54000, 'Creat Director', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Stark Industries', 'N6YH8V3C', 89000,
'Engineer', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Skynet Systems', 'O4LF5G2H', 103000,
'AI Spec', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Waystar Royco', 'P2TG4S1J', 46000,
'Attorney', 'Federal');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Spacely Sprockets', 'Q1RM3F6B', 58000,
'Ops Manager', 'Education');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Cogswell Cogs', 'R5DV2J0L', 30500,
'Mech Engineer', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('BiffCo Enterprises', 'S3HG6K8N',
50000, 'Comm Spec', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Gringotts Bank', 'T7CK1F4M', 72000,
'Bank Manager', 'Finance');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Kwik-E-Mart', 'U8EB2L5Q', 31000,
'Store Mgr', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Acme Corp', 'U8ZD3H5J', 55000, 'Logis
Coord', 'Retail');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Globex Inc', 'V6JW3H7D', 63000, 'Prod
Dev', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Initech LLC', 'W4KG2F8S', 48000, 'IT
Support', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Umbrella Corp', 'X2HN5L9T', 77000,
'Lab Tech', 'IT');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Soylent Corp', 'Y1DF4G6U', 33000,
'Dietitian', 'Healthcare');
INSERT INTO SCOTIABANK.Employments (Company_name, Customer_ID,
Salary, Job_title, Job_sector) VALUES ('Vehement Capital', 'Z9CV3B1K', 96000,
'Inv Banker', 'Finance');
```

GIC TABLE

```
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00012', 2.6, 12);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00013', 2.7, 16);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00014', 2.8, 7);
```

```
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00015', 2.9, 9);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00016', 3.0, 14);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00017', 3.1, 18);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00018', 3.2, 22);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00019', 3.3, 3);
INSERT INTO SCOTIABANK.GIC (Investment_account_number, Interest_rate,
Tenure) VALUES ('INV00020', 3.4, 6);
```

LOANS TABLE

```
INSERT INTO SCOTIABANK.Loans (Loan_ID, Loan_status, Interest_rate,
Start_date, End_date, Loan_type, Amount, Customer_ID, Branch_ID)
VALUES
('5002', '1', '5', '2022-08-15', '2025-09-15', 'Personal loan', '30000',
'B5VN1H3G', '2'),
('5003', '1', '6', '2022-07-20', '2026-07-20', 'Car loan', '25000', 'D4TM5H2S',
'1'),
('5004', '1', '4.5', '2022-06-25', '2024-06-25', 'Education loan', '40000',
'E2UN6J4K', '2'),
('5005', '1', '5.5', '2022-05-30', '2025-05-30', 'Home loan', '60000',
'F1VM8L6Q', '1'),
('5006', '1', '4.2', '2022-04-12', '2027-04-12', 'Personal loan', '35000',
'F4JL2D8P', '1'),
('5007', '1', '6.2', '2022-03-18', '2025-03-18', 'Car loan', '20000', 'G6JK8L0P',
'2'),
('5008', '1', '3.8', '2022-02-22', '2024-02-22', 'Education loan', '45000',
'G9XL3B7D', '2'),
('5009', '1', '5.7', '2022-01-25', '2026-01-25', 'Home loan', '55000',
'H3FJ5M1R', '1'),
('5010', '1', '4.3', '2021-12-30', '2024-12-30', 'Personal loan', '32000',
'H7YM2K9F', '1'),
('5011', '1', '6.5', '2021-11-05', '2025-11-05', 'Car loan', '23000', 'I4KP9N2U',
'2'),
('5012', '1', '4.7', '2021-10-10', '2023-10-10', 'Education loan', '42000',
'I5ZN1H8G', '2'),
('5013', '1', '5.8', '2021-09-15', '2024-09-15', 'Home loan', '58000',
'J3WM9L5R', '1'),
('5014', '1', '4.1', '2021-08-20', '2023-08-20', 'Personal loan', '33000',
'K2UN3J7T', '1'),
('5015', '1', '6.3', '2021-07-25', '2025-07-25', 'Car loan', '21000', 'L3SD9F6G',
'2'),
('5016', '1', '3.9', '2021-06-30', '2024-06-30', 'Education loan', '47000',
'M9QN4T5R', '2'),
('5017', '1', '5.9', '2021-06-05', '2026-06-05', 'Home loan', '62000',
'N6YH8V3C', '1'),
('5018', '1', '4.4', '2021-05-10', '2023-05-10', 'Personal loan', '34000',
'O4LF5G2H', '1'),
('5019', '1', '6.6', '2021-04-15', '2025-04-15', 'Car loan', '24000', 'P2TG4S1J',
'2'),
('5020', '1', '4.8', '2021-03-20', '2023-03-20', 'Education loan', '43000',
'Q1RM3F6B', '2'),
('5021', '1', '6.7', '2021-02-25', '2026-02-25', 'Home loan', '57000',
'R5DV2J0L', '1'),
('5022', '1', '4.2', '2021-01-30', '2024-01-30', 'Personal loan', '31000',
'S3HG6K8N','1');

####MUTUAL_FUNDS TABLE####
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00021', 'Index Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00022', 'Balanced Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00023', 'Money Market');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00024', 'Sector Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00025', 'Equity Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00026', 'Bond Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00027', 'Index Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00028', 'Balanced Funds');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00029', 'Money Market');
INSERT INTO SCOTIABANK.Mutual_Funds (Investment_account_number,
Mf_type) VALUES ('INV00030', 'Sector Funds');
```

SAVINGS TABLE

```
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008592', 2.5);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008593', 3.2);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008594', 2.8);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008595', 3.6);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008596', 3.1);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008597', 2.9);
```

```
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008598', 3.8);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008599', 3.3);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008600', 3.5);
INSERT INTO SCOTIABANK.Savings (Account_number, Interest_rate) VALUES
('10008601', 2.7);
```

TRANSACTIONS TABLE

```
INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000002', '35', 'BC Ferries-Vancouver', '2022-03-15 13:45:00',
'10008593');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000003', '50', 'Old Port-Montreal', '2022-03-16 09:30:00',
'10008594');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000004', '15', 'Niagara Falls-Ontario', '2022-03-17 14:10:00',
'10008595');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000005', '28', 'Calgary Stampede-Alberta', '2022-03-18 16:20:00',
'10008596');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000006', '42', 'CN Tower-Toronto', '2022-03-19 10:45:00',
'10008597');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000007', '18', 'Stanley Park-Vancouver', '2022-03-20 12:30:00',
'10008598');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000008', '37', 'St. Lawrence Market-Toronto', '2022-03-21
15:15:00', '10008599');
```

```
INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000009', '25', 'Whistler Ski Resort-BC', '2022-03-22 08:50:00',
'10008600');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000010', '30', 'Banff National Park-Alberta', '2022-03-23 11:55:00',
'10008601');

-- Add more INSERT statements for other transactions with different account
numbers
INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000011', '19', 'Old Quebec City-Quebec', '2022-03-24 14:40:00',
'10008602');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000012', '33', 'Rideau Canal-Ottawa', '2022-03-25 16:05:00',
'10008603');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000013', '22', 'Lake Louise-Alberta', '2022-03-26 09:20:00',
'10008604');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000014', '40', 'Whale Watching-British Columbia', '2022-03-27
10:55:00', '10008605');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000015', '27', 'PEI National Park-Prince Edward Island',
'2022-03-28 13:30:00', '10008606');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000016', '45', 'Royal Ontario Museum-Toronto', '2022-03-29
15:45:00', '10008607');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000017', '32', 'Capilano Suspension Bridge-Vancouver',
'2022-03-30 08:10:00', '10008608');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
```

```
VALUES ('10000018', '29', 'Mont Tremblant-Quebec', '2022-03-31 11:25:00',
'10008609');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000019', '38', 'Peggys Cove-Nova Scotia', '2022-04-01 14:50:00',
'10008610');

INSERT INTO `SCOTIABANK`.`Transactions` (`Transaction_ID`, `Amount`,
`Transaction_description`, `Time_stamp`, `Account_number`)
VALUES ('10000020', '21', 'Whales Bone-Ottawa', '2022-04-02 17:15:00',
'10008611');
```