<u>GROUP PROJECT- PHASE1</u>

For
George Wanganga Instructor

of *Database design and implementation*

Sheridan College

Brampton, Ontario

December 6,2023

Banking DatabaseDesign

Introduction:

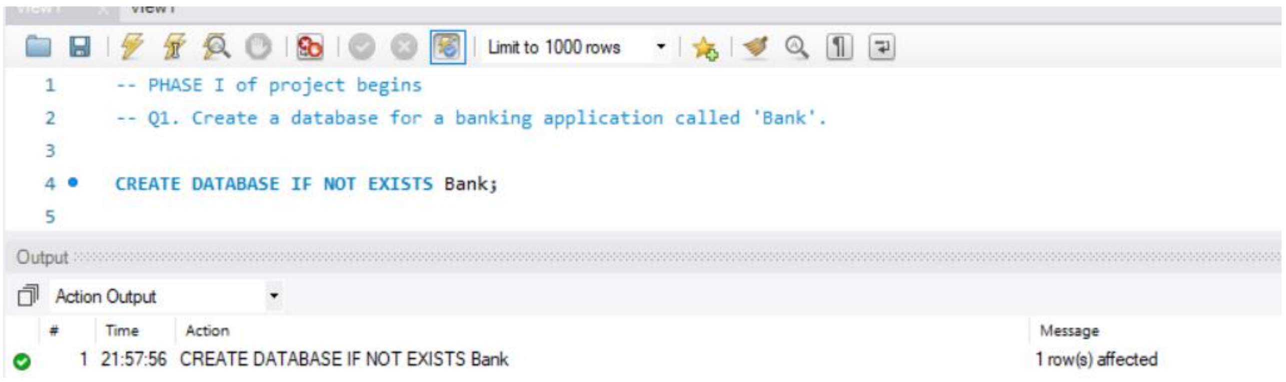
There are only two types of accounts at this time: Checking and Savings accounts. The provided column list should be separated into appropriate entities (tables) with relationships between these entities defined. The most efficient choices as far as your primary key constraints and foreign key constraints, and picked the appropriate data types for each of the columns.

Project Goals:

The goal of the project is to understand database entities in more depth and have practical experience of working with different objects of SQL.

Phase I:

1. **Create a database for a banking application called "Bank".**

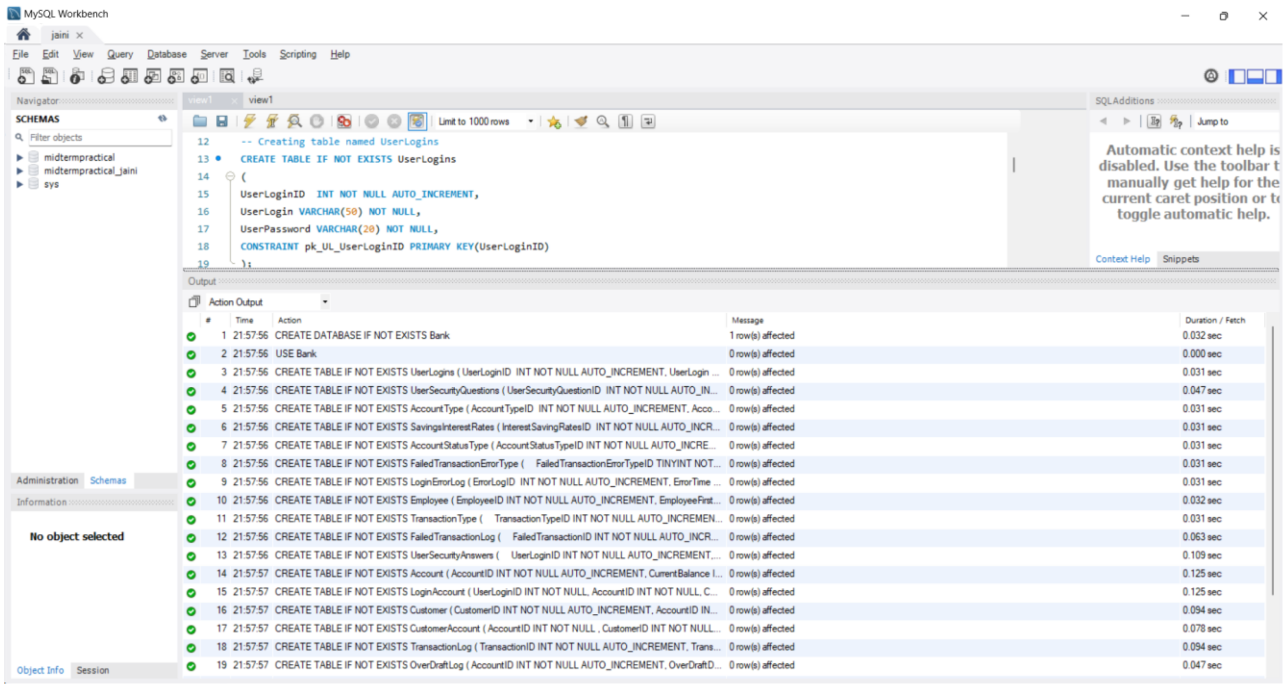


```sql
-- PHASE I of project begins
-- Q1. Create a database for a banking application called 'Bank'.

CREATE DATABASE IF NOT EXISTS Bank;
```

```
#  Time      Action                                Message
1  21:57:56  CREATE DATABASE IF NOT EXISTS Bank    1 row(s) affected
```

2. **Create all the tables mentioned in the database diagram.**



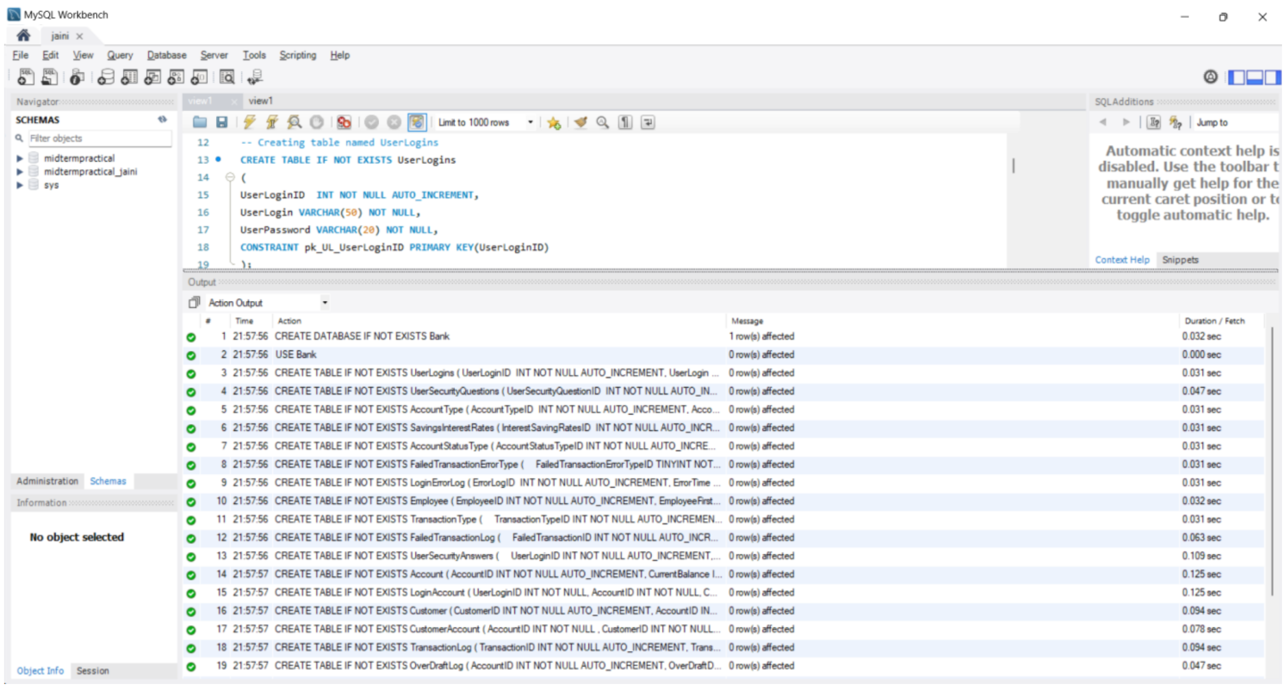
```sql
-- Creating table named UserLogins
CREATE TABLE IF NOT EXISTS UserLogins
(
UserLoginID  INT NOT NULL AUTO_INCREMENT,
UserLogin VARCHAR(50) NOT NULL,
UserPassword VARCHAR(20) NOT NULL,
CONSTRAINT pk_UL_UserLoginID PRIMARY KEY(UserLoginID)
);
```

```
#   Time      Action                                                                                          Message             Duration / Fetch
1   21:57:56  CREATE DATABASE IF NOT EXISTS Bank                                                              1 row(s) affected   0.032 sec
2   21:57:56  USE Bank                                                                                        0 row(s) affected   0.000 sec
3   21:57:56  CREATE TABLE IF NOT EXISTS UserLogins ( UserLoginID  INT NOT NULL AUTO_INCREMENT, UserLogin ...     0 row(s) affected   0.031 sec
4   21:57:56  CREATE TABLE IF NOT EXISTS UserSecurityQuestions ( UserSecurityQuestionID  INT NOT NULL AUTO_IN...  0 row(s) affected   0.047 sec
5   21:57:56  CREATE TABLE IF NOT EXISTS AccountType ( AccountTypeID  INT NOT NULL AUTO_INCREMENT, Acco...        0 row(s) affected   0.031 sec
6   21:57:56  CREATE TABLE IF NOT EXISTS SavingsInterestRates ( InterestSavingRatesID  INT NOT NULL AUTO_INCR...  0 row(s) affected   0.031 sec
7   21:57:56  CREATE TABLE IF NOT EXISTS AccountStatusType ( AccountStatusTypeID INT NOT NULL AUTO_INCRE...      0 row(s) affected   0.031 sec
8   21:57:56  CREATE TABLE IF NOT EXISTS FailedTransactionErrorType (    FailedTransactionErrorTypeID TINYINT NOT...  0 row(s) affected   0.031 sec
9   21:57:56  CREATE TABLE IF NOT EXISTS LoginErrorLog ( ErrorLogID  INT NOT NULL AUTO_INCREMENT, ErrorTime ...   0 row(s) affected   0.031 sec
10  21:57:56  CREATE TABLE IF NOT EXISTS Employee ( EmployeeID INT NOT NULL AUTO_INCREMENT, EmployeeFirst...      0 row(s) affected   0.032 sec
11  21:57:56  CREATE TABLE IF NOT EXISTS TransactionType (    TransactionTypeID INT NOT NULL AUTO_INCREMEN...     0 row(s) affected   0.031 sec
12  21:57:56  CREATE TABLE IF NOT EXISTS FailedTransactionLog (    FailedTransactionID INT NOT NULL AUTO_INCR...  0 row(s) affected   0.063 sec
13  21:57:56  CREATE TABLE IF NOT EXISTS UserSecurityAnswers (    UserLoginID INT NOT NULL AUTO_INCREMENT,...     0 row(s) affected   0.109 sec
14  21:57:57  CREATE TABLE IF NOT EXISTS Account ( AccountID INT NOT NULL AUTO_INCREMENT, CurrentBalance I...     0 row(s) affected   0.125 sec
15  21:57:57  CREATE TABLE IF NOT EXISTS LoginAccount ( UserLoginID INT NOT NULL, AccountID INT NOT NULL, C...    0 row(s) affected   0.125 sec
16  21:57:57  CREATE TABLE IF NOT EXISTS Customer ( CustomerID INT NOT NULL AUTO_INCREMENT, AccountID IN...       0 row(s) affected   0.094 sec
17  21:57:57  CREATE TABLE IF NOT EXISTS CustomerAccount ( AccountID INT NOT NULL , CustomerID INT NOT NULL...    0 row(s) affected   0.078 sec
18  21:57:57  CREATE TABLE IF NOT EXISTS TransactionLog ( TransactionID INT NOT NULL AUTO_INCREMENT, Trans...     0 row(s) affected   0.094 sec
19  21:57:57  CREATE TABLE IF NOT EXISTS OverDraftLog ( AccountID INT NOT NULL AUTO_INCREMENT, OverDraftD...      0 row(s) affected   0.047 sec
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

No object selected

3. Create all the constraints based on the database diagram.

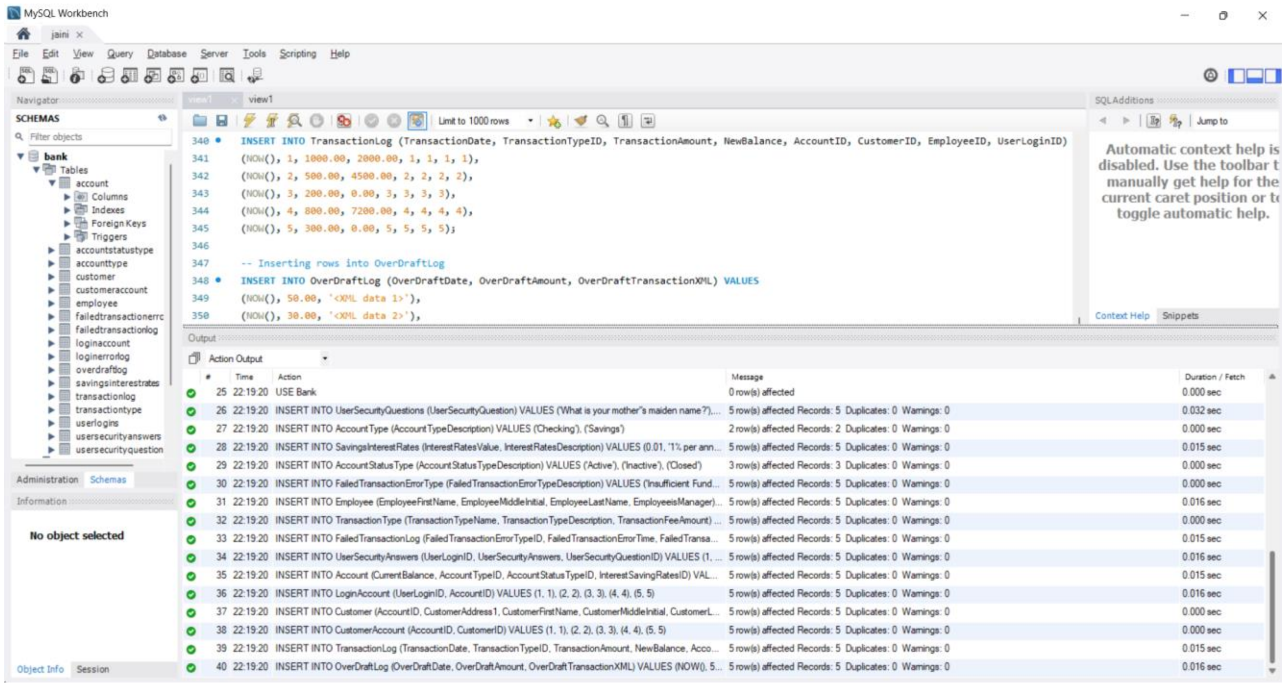


```sql
-- Creating table named UserLogins
CREATE TABLE IF NOT EXISTS UserLogins
(
UserLoginID  INT NOT NULL AUTO_INCREMENT,
UserLogin VARCHAR(50) NOT NULL,
UserPassword VARCHAR(20) NOT NULL,
CONSTRAINT pk_UL_UserLoginID PRIMARY KEY(UserLoginID)
);
```

```
 #  Time      Action                                                                                              Message            Duration / Fetch
 1  21:57:56  CREATE DATABASE IF NOT EXISTS Bank                                                                  1 row(s) affected  0.032 sec
 2  21:57:56  USE Bank                                                                                            0 row(s) affected  0.000 sec
 3  21:57:56  CREATE TABLE IF NOT EXISTS UserLogins ( UserLoginID  INT NOT NULL AUTO_INCREMENT, UserLogin ...       0 row(s) affected  0.031 sec
 4  21:57:56  CREATE TABLE IF NOT EXISTS UserSecurityQuestions ( UserSecurityQuestionID  INT NOT NULL AUTO_IN...   0 row(s) affected  0.047 sec
 5  21:57:56  CREATE TABLE IF NOT EXISTS AccountType ( AccountTypeID  INT NOT NULL AUTO_INCREMENT, Acco...         0 row(s) affected  0.031 sec
 6  21:57:56  CREATE TABLE IF NOT EXISTS SavingsInterestRates ( InterestSavingRatesID  INT NOT NULL AUTO_INCR...   0 row(s) affected  0.031 sec
 7  21:57:56  CREATE TABLE IF NOT EXISTS AccountStatusType ( AccountStatusTypeID INT NOT NULL AUTO_INCRE...        0 row(s) affected  0.031 sec
 8  21:57:56  CREATE TABLE IF NOT EXISTS FailedTransactionErrorType (    FailedTransactionErrorTypeID TINYINT NOT... 0 row(s) affected  0.031 sec
 9  21:57:56  CREATE TABLE IF NOT EXISTS LoginErrorLog ( ErrorLogID  INT NOT NULL AUTO_INCREMENT, ErrorTime ...     0 row(s) affected  0.031 sec
10  21:57:56  CREATE TABLE IF NOT EXISTS Employee ( EmployeeID INT NOT NULL AUTO_INCREMENT, EmployeeFirst...       0 row(s) affected  0.032 sec
11  21:57:56  CREATE TABLE IF NOT EXISTS TransactionType (    TransactionTypeID INT NOT NULL AUTO_INCREMEN...     0 row(s) affected  0.031 sec
12  21:57:56  CREATE TABLE IF NOT EXISTS FailedTransactionLog (    FailedTransactionID INT NOT NULL AUTO_INCR...  0 row(s) affected  0.063 sec
13  21:57:56  CREATE TABLE IF NOT EXISTS UserSecurityAnswers (    UserLoginID INT NOT NULL AUTO_INCREMENT,...     0 row(s) affected  0.109 sec
14  21:57:57  CREATE TABLE IF NOT EXISTS Account ( AccountID INT NOT NULL AUTO_INCREMENT, CurrentBalance I...      0 row(s) affected  0.125 sec
15  21:57:57  CREATE TABLE IF NOT EXISTS LoginAccount ( UserLoginID INT NOT NULL, AccountID INT NOT NULL, C...     0 row(s) affected  0.125 sec
16  21:57:57  CREATE TABLE IF NOT EXISTS Customer ( CustomerID INT NOT NULL AUTO_INCREMENT, AccountID IN...        0 row(s) affected  0.094 sec
17  21:57:57  CREATE TABLE IF NOT EXISTS CustomerAccount ( AccountID INT NOT NULL , CustomerID INT NOT NULL...     0 row(s) affected  0.078 sec
18  21:57:57  CREATE TABLE IF NOT EXISTS TransactionLog ( TransactionID INT NOT NULL AUTO_INCREMENT, Trans...      0 row(s) affected  0.094 sec
19  21:57:57  CREATE TABLE IF NOT EXISTS OverDraftLog ( AccountID INT NOT NULL AUTO_INCREMENT, OverDraftD...       0 row(s) affected  0.047 sec
```

4. Insert at least 5 rows in each

table.



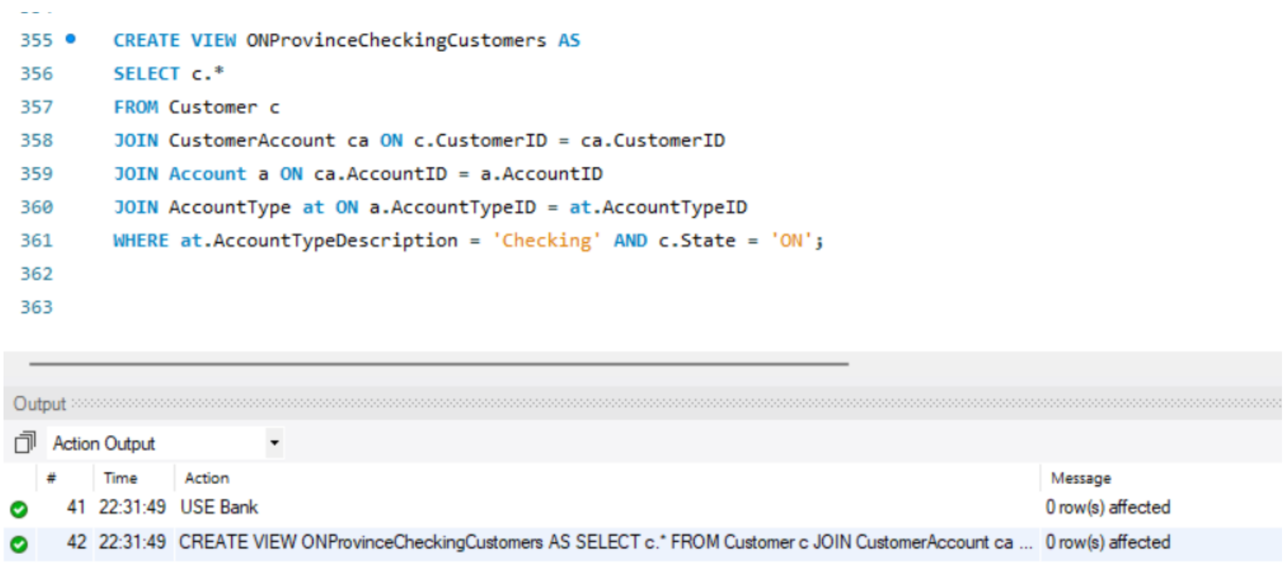
```sql
INSERT INTO TransactionLog (TransactionDate, TransactionTypeID, TransactionAmount, NewBalance, AccountID, CustomerID, EmployeeID, UserLoginID)
(NOW(), 1, 1000.00, 2000.00, 1, 1, 1, 1),
(NOW(), 2, 500.00, 4500.00, 2, 2, 2, 2),
(NOW(), 3, 200.00, 0.00, 3, 3, 3, 3),
(NOW(), 4, 800.00, 7200.00, 4, 4, 4, 4),
(NOW(), 5, 300.00, 0.00, 5, 5, 5, 5);

-- Inserting rows into OverDraftLog
INSERT INTO OverDraftLog (OverDraftDate, OverDraftAmount, OverDraftTransactionXML) VALUES
(NOW(), 50.00, '<XML data 1>'),
(NOW(), 30.00, '<XML data 2>'),
```

```
 #  Time      Action                                                                                                    Message                                                   Duration / Fetch
25  22:19:20  USE Bank                                                                                                  0 row(s) affected                                         0.000 sec
26  22:19:20  INSERT INTO UserSecurityQuestions (UserSecurityQuestion) VALUES ('What is your mother''s maiden name?'),...  5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.032 sec
27  22:19:20  INSERT INTO AccountType (AccountTypeDescription) VALUES ('Checking'), ('Savings')                          2 row(s) affected Records: 2  Duplicates: 0  Warnings: 0  0.000 sec
28  22:19:20  INSERT INTO SavingsInterestRates (InterestRatesValue, InterestRatesDescription) VALUES (0.01, '1% per ann...  5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.015 sec
29  22:19:20  INSERT INTO AccountStatusType (AccountStatusTypeDescription) VALUES ('Active'), ('Inactive'), ('Closed')     3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
30  22:19:20  INSERT INTO FailedTransactionErrorType (FailedTransactionErrorTypeDescription) VALUES ('Insufficient Fund... 5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.000 sec
31  22:19:20  INSERT INTO Employee (EmployeeFirstName, EmployeeMiddleInitial, EmployeeLastName, EmployeeisManager)...    5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.016 sec
32  22:19:20  INSERT INTO TransactionType (TransactionTypeName, TransactionTypeDescription, TransactionFeeAmount) ...    5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.000 sec
33  22:19:20  INSERT INTO FailedTransactionLog (FailedTransactionErrorTypeID, FailedTransactionErrorTime, FailedTransa... 5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.015 sec
34  22:19:20  INSERT INTO UserSecurityAnswers (UserLoginID, UserSecurityAnswers, UserSecurityQuestionID) VALUES (1, ...   5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.016 sec
35  22:19:20  INSERT INTO Account (CurrentBalance, AccountTypeID, AccountStatusTypeID, InterestSavingRatesID) VAL...     5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.015 sec
36  22:19:20  INSERT INTO LoginAccount (UserLoginID, AccountID) VALUES (1, 1), (2, 2), (3, 3), (4, 4), (5, 5)           5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.016 sec
37  22:19:20  INSERT INTO Customer (AccountID, CustomerAddress1, CustomerFirstName, CustomerMiddleInitial, CustomerL...  5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.000 sec
38  22:19:20  INSERT INTO CustomerAccount (AccountID, CustomerID) VALUES (1, 1), (2, 2), (3, 3), (4, 4), (5, 5)         5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.000 sec
39  22:19:20  INSERT INTO TransactionLog (TransactionDate, TransactionTypeID, TransactionAmount, NewBalance, Acco...      5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.015 sec
40  22:19:20  INSERT INTO OverDraftLog (OverDraftDate, OverDraftAmount, OverDraftTransactionXML) VALUES (NOW(), 5...     5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0  0.016 sec
```

Phase II:

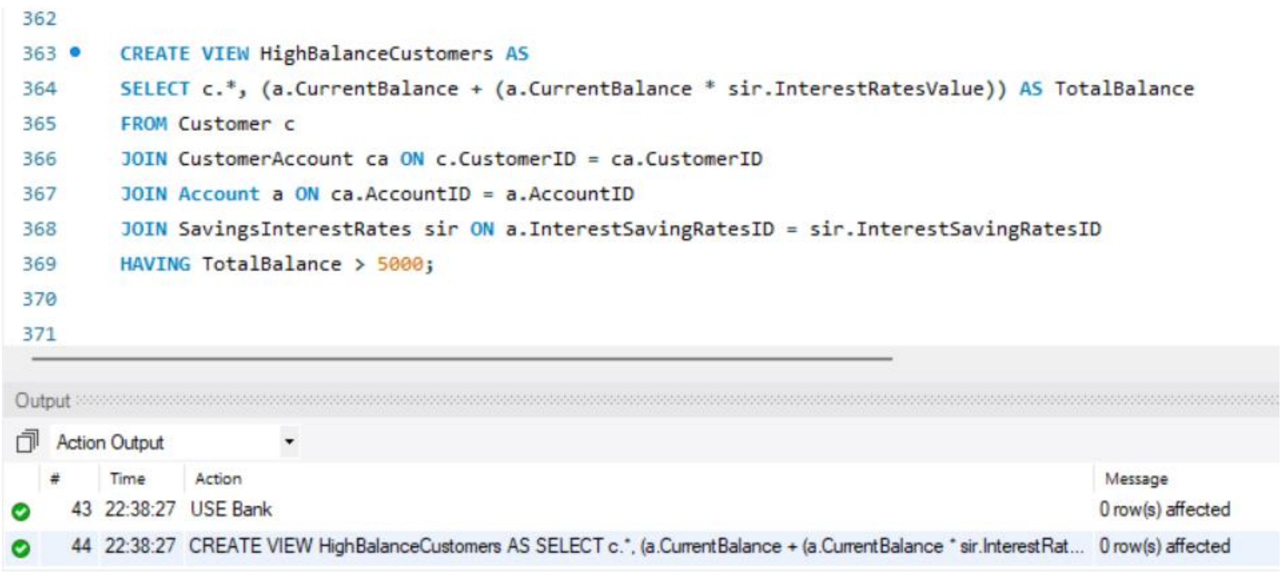
1. **Create a view to get all customers with checking account from ON province**.



```
CREATE VIEW ONProvinceCheckingCustomers AS
SELECT c.*
FROM Customer c
JOIN CustomerAccount ca ON c.CustomerID = ca.CustomerID
JOIN Account a ON ca.AccountID = a.AccountID
JOIN AccountType at ON a.AccountTypeID = at.AccountTypeID
WHERE at.AccountTypeDescription = 'Checking' AND c.State = 'ON';
```

```
Output
Action Output
   #   Time      Action                                                                                   Message
   41  22:31:49  USE Bank                                                                                 0 row(s) affected
   42  22:31:49  CREATE VIEW ONProvinceCheckingCustomers AS SELECT c.* FROM Customer c JOIN CustomerAccount ca ...  0 row(s) affected
```

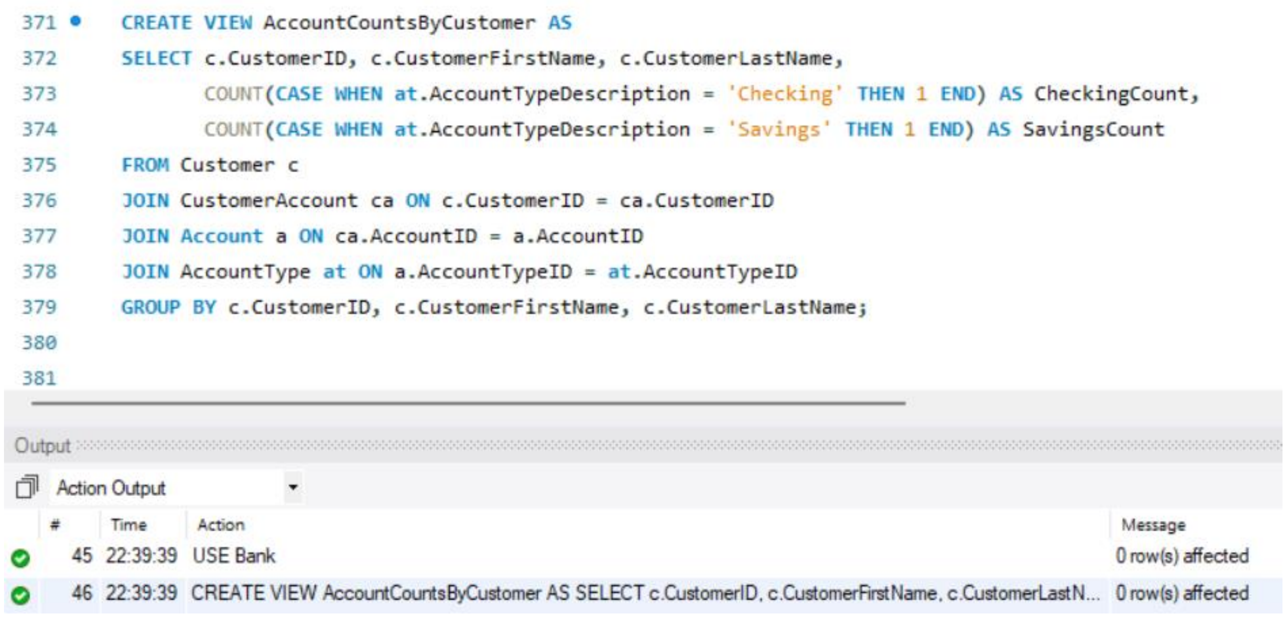
2. Create a view to get all customers with total account balance (including interest rate) greater than5000.



```
CREATE VIEW HighBalanceCustomers AS
SELECT c.*, (a.CurrentBalance + (a.CurrentBalance * sir.InterestRatesValue)) AS TotalBalance
FROM Customer c
JOIN CustomerAccount ca ON c.CustomerID = ca.CustomerID
JOIN Account a ON ca.AccountID = a.AccountID
JOIN SavingsInterestRates sir ON a.InterestSavingRatesID = sir.InterestSavingRatesID
HAVING TotalBalance > 5000;
```

```
Output
Action Output
   #   Time      Action                                                                                        Message
   43  22:38:27  USE Bank                                                                                      0 row(s) affected
   44  22:38:27  CREATE VIEW HighBalanceCustomers AS SELECT c.*, (a.CurrentBalance + (a.CurrentBalance * sir.InterestRat...  0 row(s) affected
```

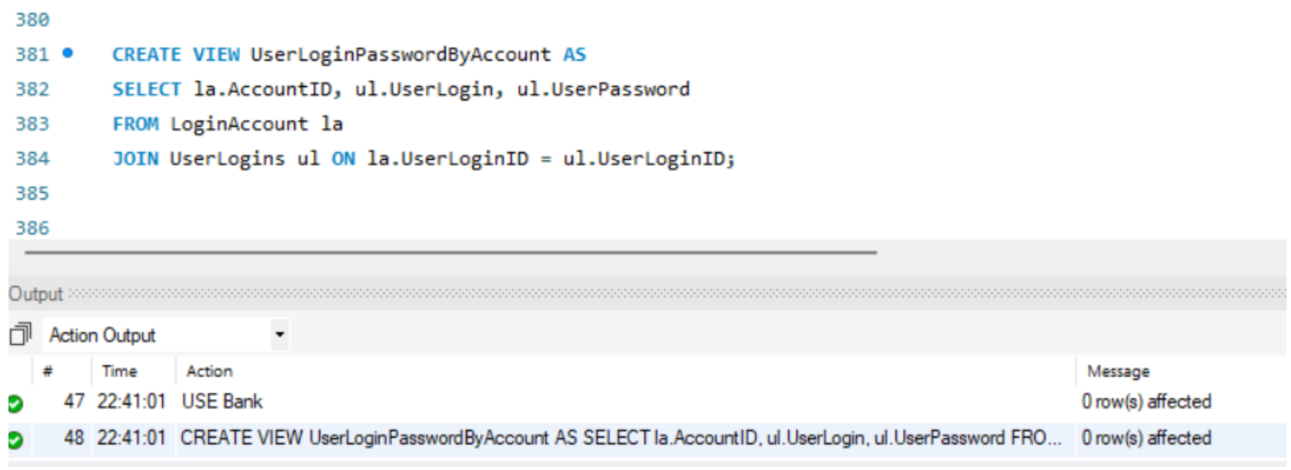
3. **Create a view to get counts of checking and savings accounts by customer.**



```sql
CREATE VIEW AccountCountsByCustomer AS
SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLastName,
       COUNT(CASE WHEN at.AccountTypeDescription = 'Checking' THEN 1 END) AS CheckingCount,
       COUNT(CASE WHEN at.AccountTypeDescription = 'Savings' THEN 1 END) AS SavingsCount
FROM Customer c
JOIN CustomerAccount ca ON c.CustomerID = ca.CustomerID
JOIN Account a ON ca.AccountID = a.AccountID
JOIN AccountType at ON a.AccountTypeID = at.AccountTypeID
GROUP BY c.CustomerID, c.CustomerFirstName, c.CustomerLastName;
```

```
Output
Action Output
 #   Time      Action                                                                                   Message
 45  22:39:39  USE Bank                                                                                 0 row(s) affected
 46  22:39:39  CREATE VIEW AccountCountsByCustomer AS SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLastN...  0 row(s) affected
```

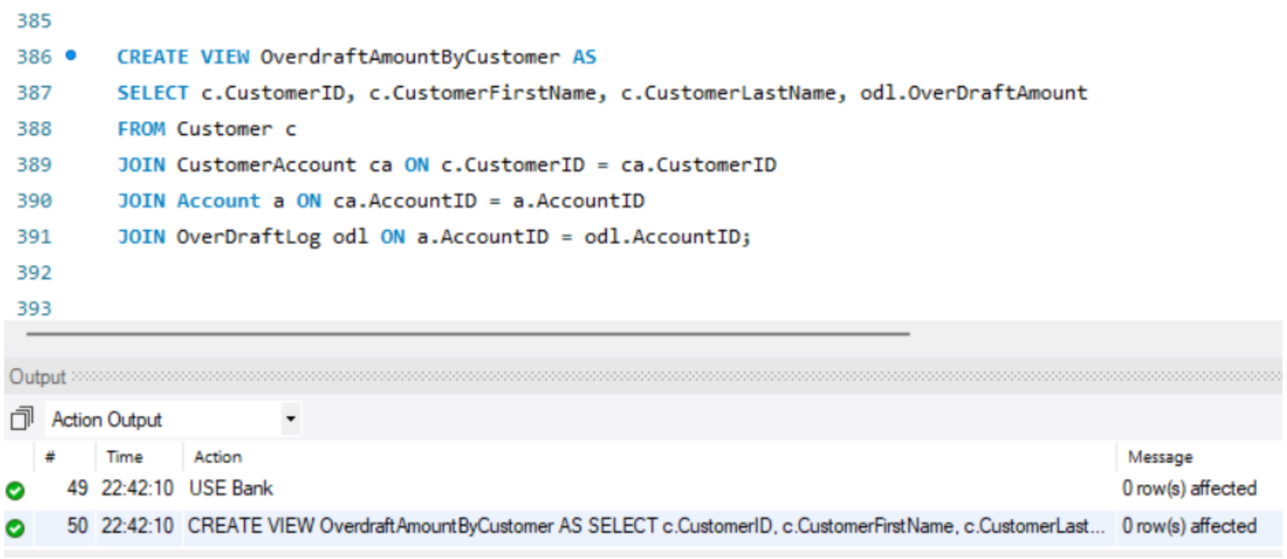
4. **Create a view to get any particular user's login and password using AccountId.**



```sql
CREATE VIEW UserLoginPasswordByAccount AS
SELECT la.AccountID, ul.UserLogin, ul.UserPassword
FROM LoginAccount la
JOIN UserLogins ul ON la.UserLoginID = ul.UserLoginID;
```

```
Output
Action Output
 #   Time      Action                                                                                   Message
 47  22:41:01  USE Bank                                                                                 0 row(s) affected
 48  22:41:01  CREATE VIEW UserLoginPasswordByAccount AS SELECT la.AccountID, ul.UserLogin, ul.UserPassword FRO...  0 row(s) affected
```

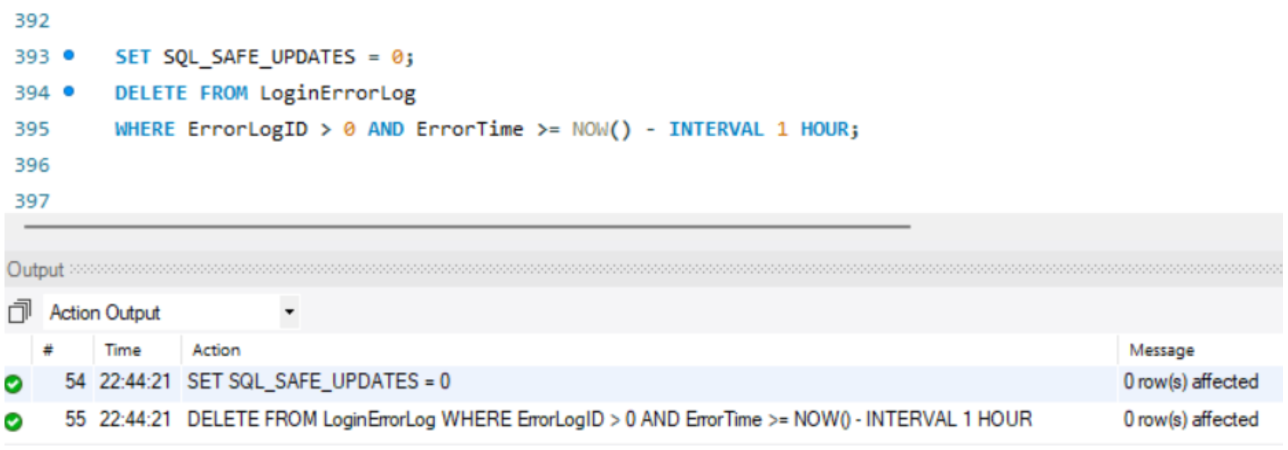
5. **Create a view to get all customers' overdraft amount.**



```sql
CREATE VIEW OverdraftAmountByCustomer AS
SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLastName, odl.OverDraftAmount
FROM Customer c
JOIN CustomerAccount ca ON c.CustomerID = ca.CustomerID
JOIN Account a ON ca.AccountID = a.AccountID
JOIN OverDraftLog odl ON a.AccountID = odl.AccountID;
```

```
Output
Action Output
 #   Time      Action                                                                                   Message
 49  22:42:10  USE Bank                                                                                 0 row(s) affected
 50  22:42:10  CREATE VIEW OverdraftAmountByCustomer AS SELECT c.CustomerID, c.CustomerFirstName, c.CustomerLast...  0 row(s) affected
```

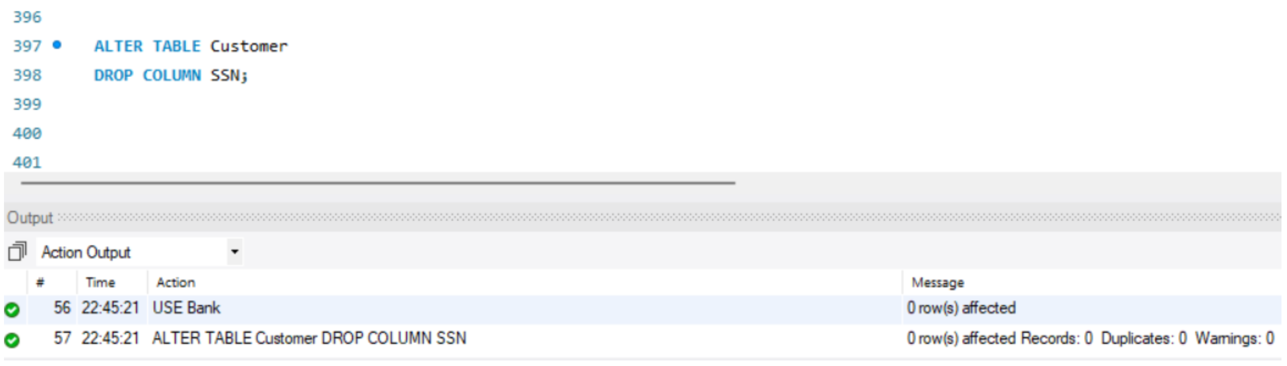
6. **Delete all error logs created in the last hour.**



```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM LoginErrorLog
WHERE ErrorLogID > 0 AND ErrorTime >= NOW() - INTERVAL 1 HOUR;
```

```
Output
Action Output
   #   Time      Action                                                                   Message
✓  54  22:44:21  SET SQL_SAFE_UPDATES = 0                                                 0 row(s) affected
✓  55  22:44:21  DELETE FROM LoginErrorLog WHERE ErrorLogID > 0 AND ErrorTime >= NOW() - INTERVAL 1 HOUR   0 row(s) affected
```

7. **Write a query to remove SSN column from Customer table.**



```
ALTER TABLE Customer
DROP COLUMN SSN;
```

```
Output
Action Output
   #   Time      Action                                  Message
✓  56  22:45:21  USE Bank                                0 row(s) affected
✓  57  22:45:21  ALTER TABLE Customer DROP COLUMN SSN    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```