First query

```sql
CREATE DATABASE Bank
CREATE TABLE Clients (
Id INT PRIMARY KEY IDENTITY,
FirstName NVARCHAR(50) NOT NULL, LastName NVARCHAR(50) NOT NULL
)
CREATE TABLE AccountTypes (
Id INT PRIMARY KEY IDENTITY,
[Name] NVARCHAR(50) NOT NULL
)
CREATE TABLE Accounts (
Id INT PRIMARY KEY IDENTITY,
AccountTypeId INT FOREIGN KEY REFERENCES AccountTypes(Id),
Balance DECIMAL(15, 2) NOT NULL DEFAULT(0),
ClientId INT FOREIGN KEY REFERENCES Clients(Id)
)
INSERT INTO Clients (FirstName, LastName) VALUES
('Greta', 'Andersson'),
('Peter', 'Pettersson'),
('Mel', 'Gibson'),
('Maria', 'Danielsson')

INSERT INTO AccountTypes (Name) VALUES
('Checking'),
('Savings')
INSERT INTO Accounts (ClientId, AccountTypeId, Balance) VALUES
(1, 1, 175),
(2, 1, 275.56),
(3, 1, 138.01),
(4, 1, 40.30),
(4, 2, 375.50)

select * from Clients
SELECT * FROM v_ClientBalances

p_AddAccount 2, 2;
p_AddAccount 4, 3

SELECT * FROM Accounts
SELECT * FROM Transactions
```

Second query

```sql
CREATE VIEW v_ClientBalances AS
SELECT (FirstName + ' ' + LastName) AS [Name],
(AccountTypes.Name) AS [Account Type], Balance
FROM Clients
JOIN Accounts ON Clients.Id = Accounts.ClientId
JOIN AccountTypes ON AccountTypes.Id = Accounts.AccountTypeId
```

Third query

```
CREATE FUNCTION f_CalculateTotalBalance (@ClientID INT)
RETURNS DECIMAL(15, 2)
BEGIN
DECLARE @result AS DECIMAL(15, 2) = (
SELECT SUM(Balance)
FROM Accounts WHERE ClientId = @ClientID
)
RETURN @result
END
```

Fourth query

```
SELECT dbo.f_CalculateTotalBalance(4) AS Balance
```

Fifth query

```
CREATE PROC p_AddAccount @ClientId INT, @AccountTypeId INT AS
INSERT INTO Accounts (ClientId, AccountTypeId)
VALUES (@ClientId, @AccountTypeId)
```

Sixth query

```
CREATE PROC p_Deposit @AccountId INT, @Amount DECIMAL(15, 2) AS
UPDATE Accounts
SET Balance += @Amount
WHERE Id = @AccountId
```

Seventh query

```
CREATE PROC p_Deposit @AccountId INT, @Amount DECIMAL(15, 2) AS
UPDATE Accounts
SET Balance += @Amount
WHERE Id = @AccountId
```

Eight query

```
AFTER UPDATE
AS
INSERT INTO Transactions (AccountId, OldBalance, NewBalance, [DateTime])
SELECT inserted.Id, deleted.Balance, inserted.Balance, GETDATE() FROM inserted
JOIN deleted ON inserted.Id = deleted.Id
```

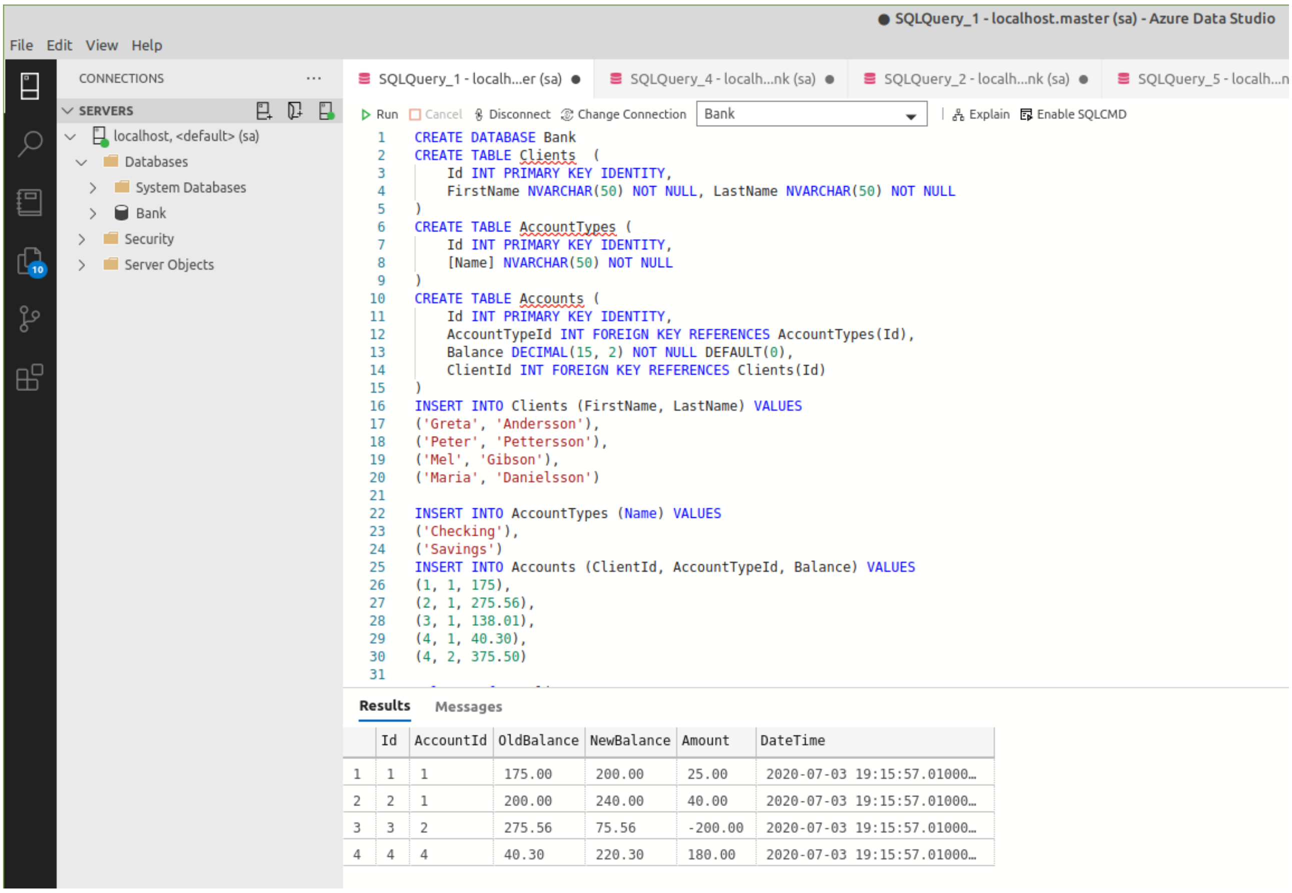
Ninth query

```
p_Deposit 1, 25.00
GO

p_Deposit 1, 40.00
GO

p_Withdraw 2, 200.00
GO
p_Deposit 4, 180.00
GO
```

Tenth query

```
CREATE TABLE Transactions (
Id INT PRIMARY KEY IDENTITY,
AccountId INT FOREIGN KEY REFERENCES Accounts(Id),
OldBalance DECIMAL(15, 2) NOT NULL,
NewBalance DECIMAL(15, 2) NOT NULL,
Amount AS NewBalance - OldBalance,
[DateTime] DATETIME2
)
```

Result:



```
CREATE DATABASE Bank
CREATE TABLE Clients  (
    Id INT PRIMARY KEY IDENTITY,
    FirstName NVARCHAR(50) NOT NULL, LastName NVARCHAR(50) NOT NULL
)
CREATE TABLE AccountTypes (
    Id INT PRIMARY KEY IDENTITY,
    [Name] NVARCHAR(50) NOT NULL
)
CREATE TABLE Accounts (
    Id INT PRIMARY KEY IDENTITY,
    AccountTypeId INT FOREIGN KEY REFERENCES AccountTypes(Id),
    Balance DECIMAL(15, 2) NOT NULL DEFAULT(0),
    ClientId INT FOREIGN KEY REFERENCES Clients(Id)
)
INSERT INTO Clients (FirstName, LastName) VALUES
('Greta', 'Andersson'),
('Peter', 'Pettersson'),
('Mel', 'Gibson'),
('Maria', 'Danielsson')

INSERT INTO AccountTypes (Name) VALUES
('Checking'),
('Savings')
INSERT INTO Accounts (ClientId, AccountTypeId, Balance) VALUES
(1, 1, 175),
(2, 1, 275.56),
(3, 1, 138.01),
(4, 1, 40.30),
(4, 2, 375.50)
```

	Id	AccountId	OldBalance	NewBalance	Amount	DateTime
1	1	1	175.00	200.00	25.00	2020-07-03 19:15:57.01000…
2	2	1	200.00	240.00	40.00	2020-07-03 19:15:57.01000…
3	3	2	275.56	75.56	-200.00	2020-07-03 19:15:57.01000…
4	4	4	40.30	220.30	180.00	2020-07-03 19:15:57.01000…