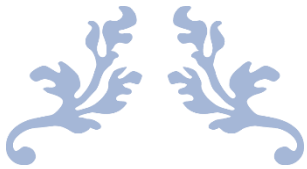


UBER COMPETITOR DATABASE DESIGN

Abstract

This is the project of designing,implementing and testing a database for uber competitor. All requirements listed are satisfied. It is aggregated all of knowledges I learned in Database management system class in 2019 Spring Semester.

NAME:TIANXIANG YI

SUID:312944880

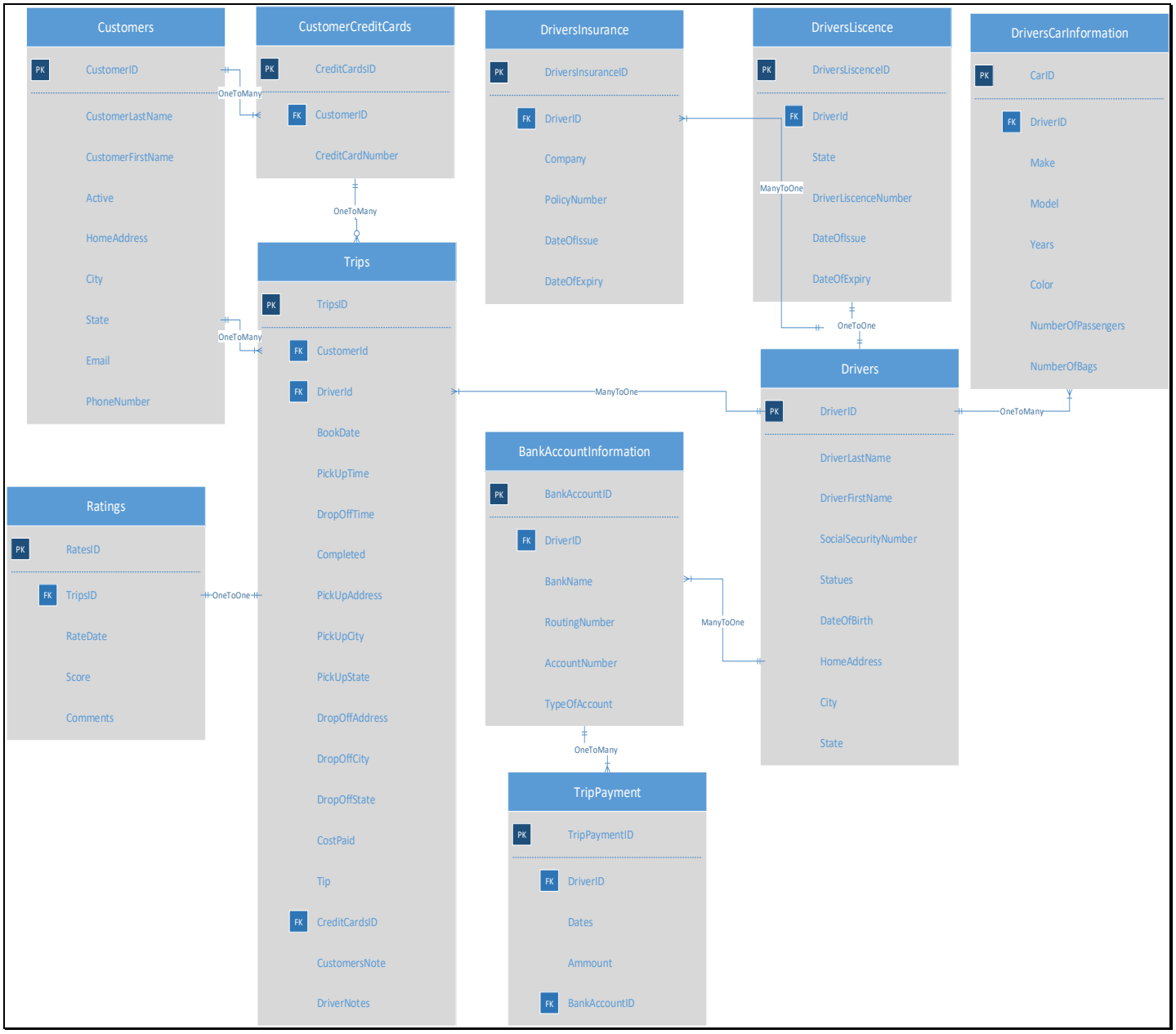
1.Design

1.1 Introduction

The Uber Competitor Database Design maps the logical data model to the target database management system with consideration to the system's performance requirements. The Database Design converts logical data like customer, driver, trips and rates to physical storage constructs (tables) of the target DBMS

1.2 Description

In this Uber Competitor Database Design Project, I implement 10 tables and each table have lots of attribute.



This is Entity/Relaionship Diagram of my Design. You could see that it store all of the necessary information need to store, each table has relationship between them and some tables get Foreign Key Constraints to satisfy business logic requirement, all the table names are satisfy the 'CamelCase' requirement. In the following code, you could see I build four business report with four Views:BadRatingTrip, DriverDetailedsInformation, CardInformation and HistoryTrips. I using scripts to estabilish the security level with Database Login 'TianxiangYi' and password '123456'. Also, I create two stored procedures to show 'Total number of customer oders' and 'Driver pickup records in one month' and two functions to show company income ,expendure and balance in one month.

2.Implementation

2.1 Brief Description

In the implementation part, first I normalize all the table to 3rd Normal Form. For example, I divide the address to address, city, state and divide the name to first name and last name to satisfy the attribute atomic. Then I decide which attribute is only denpend on the Customer, Driver and Trips to group them in different table to satisfy 2nd and 3rd Normal Form.

Second, I determines primary keys, each attribute/column datatype, nullabilities in table. Every table I generate primary key like 'XXX'ID for the purpose of quering fast. Then I consider the relationship between tables and tables.and set foreign key constrain. For example, the Relationship between Customers and Drivers in many to many relationship. So I implement a new table Trips which is relationship table to make each to Trips is one to many relationship and set CustomerID and DriverID is foreign key in Table Trips.

Third, I consider some potential integrity and security issues about the implementation I done. I find out that because I implment lots of foreign constraints, so when you drop talbes, you should consider it carefully. Also in some table like CustomerCreditCard, I should use the composite primary key composed of CustomerID and StoredCardSequence which is more safety for paying money. But I use the primary key CardID, it is more efficient in querying but lower safety level.

2.2 Code of creation of the database

Create 10 table with right datatypes, nullablilities and relationship in SQL

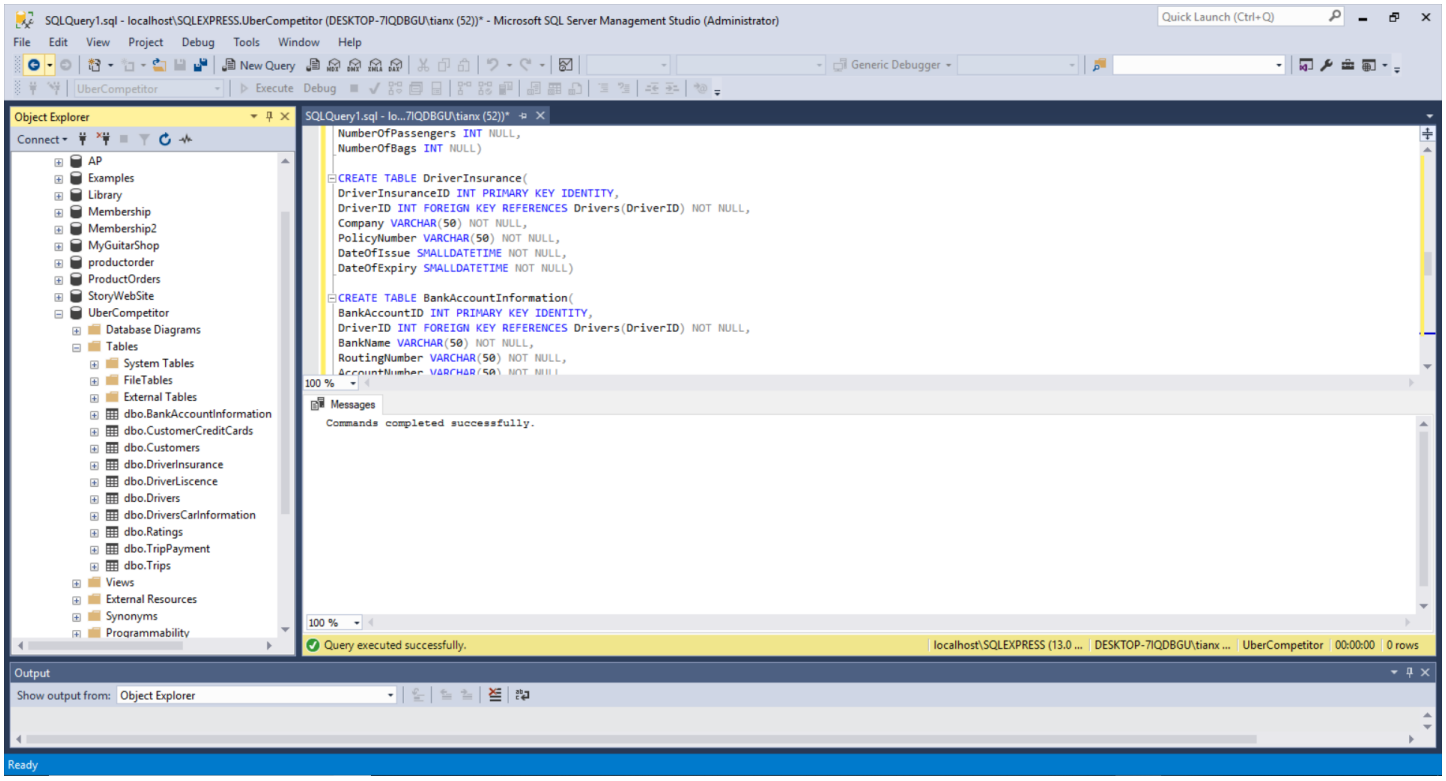
"

```
CREATE DATABASE UberCompetitor
GO

USE UberCompetitor
CREATE TABLE Customers
(CustomerID INT PRIMARY KEY IDENTITY,
CustomerLastName VARCHAR(50) NOT NULL,
CustomerFirstName VARCHAR(50) NOT NULL,
Active BIT NOT NULL,
HomeAddress VARCHAR(50) NOT NULL,
City VARCHAR(50) NOT NULL,
State VARCHAR(50) NOT NULL,
Email VARCHAR(50) NOT NULL,
PhoneNumber VARCHAR(50) NOT NULL,
)

CREATE TABLE CustomerCreditCards
(CreditCardID INT PRIMARY KEY IDENTITY,
```

```
CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID) NOT NULL,
CreditCardNumber  VARCHAR(50) NOT NULL)

CREATE TABLE Drivers(
DriverID INT PRIMARY KEY IDENTITY,
DriverLastName VARCHAR(50) NOT NULL,
DriverFirstName VARCHAR(50) NOT NULL,
SocialSecurityNumber VARCHAR(50) NOT NULL,
Statues VARCHAR(50) NOT NULL,
DateOfBirth SMALLDATETIME NOT NULL,

HomeAddress VARCHAR(50) NOT NULL ,
City VARCHAR(50) NOT NULL,
    State VARCHAR(50) NOT NULL
)

CREATE TABLE DriverLiscence(
DriverLiscenceID INT PRIMARY KEY IDENTITY,
DriverID INT FOREIGN KEY REFERENCES Drivers(DriverID) NOT NULL,
State VARCHAR(50) NOT NULL,
DriverLiscenceNumber VARCHAR(50) NOT NULL,
DateOfIssue SMALLDATETIME NOT NULL,
DateOfExpiry SMALLDATETIME NOT NULL
)

CREATE TABLE DriversCarInformation(
CarID INT PRIMARY KEY IDENTITY,
DriverID INT FOREIGN KEY REFERENCES Drivers(DriverID) NOT NULL,
Make VARCHAR(50) NOT NULL,
Model VARCHAR(50) NULL,
Years INT NOT NULL,
Color VARCHAR(50) NOT NULL,
NumberOfPassengers INT NULL,
NumberOfBags INT NULL)

CREATE TABLE DriverInsurance(
DriverInsuranceID INT PRIMARY KEY IDENTITY,
DriverID INT FOREIGN KEY REFERENCES Drivers(DriverID) NOT NULL,
Company VARCHAR(50) NOT NULL,
PolicyNumber VARCHAR(50) NOT NULL,
DateOfIssue SMALLDATETIME NOT NULL,
DateOfExpiry SMALLDATETIME NOT NULL)

CREATE TABLE BankAccountInformation(
```

```
BankAccountID INT PRIMARY KEY IDENTITY,
DriverID INT FOREIGN KEY REFERENCES Drivers(DriverID) NOT NULL,
BankName VARCHAR(50) NOT NULL,
RoutingNumber VARCHAR(50) NOT NULL,
AccountNumber VARCHAR(50) NOT NULL,
TypeOfAccount VARCHAR(50) NOT NULL
)

CREATE TABLE TripPayment(
TripPaymentID INT PRIMARY KEY IDENTITY,
DriverID INT FOREIGN KEY REFERENCES Drivers(DriverID) NOT NULL,
Dates SMALLDATETIME NOT NULL,
Ammount MONEY NOT NULL,
BankAccountID INT FOREIGN KEY REFERENCES BankAccountInformation(BankAccountID) NOT NULL
)

CREATE TABLE Trips(
TripsID INT PRIMARY KEY IDENTITY,
CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID) NOT NULL,
DriverID INT FOREIGN KEY REFERENCES Drivers(DriverID) NOT NULL,
BookDate SMALLDATETIME NOT NULL,
PickUpTime SMALLDATETIME  NULL,
DropOffTime SMALLDATETIME NULL,
Completed BIT NOT NULL,
PickUpAddress VARCHAR(50) NOT NULL,
PickUpCity VARCHAR(50) NOT NULL,
PickUPState VARCHAR(50) NOT NULL,
DropOffAddress VARCHAR(50) NOT NULL,
DropOffCity VARCHAR(50) NOT NULL,
DropOffState VARCHAR(50) NOT NULL,
CostPaid MONEY NULL,
Tip MONEY NULL,
CreditCardID INT FOREIGN KEY REFERENCES CustomerCreditCards(CreditCardID) NOT NULL,
CustomerNote Text NULL,
DriverNote Text NULL)

CREATE TABLE Ratings(
RatesID INT PRIMARY KEY IDENTITY,
TripsID INT FOREIGN KEY REFERENCES Trips(TripsID) NOT NULL,
RateDate SMALLDATETIME NULL,
Score INT NULL,
Comments TEXT NULL
)
"
```

Script result:



2.3 Code about data loading

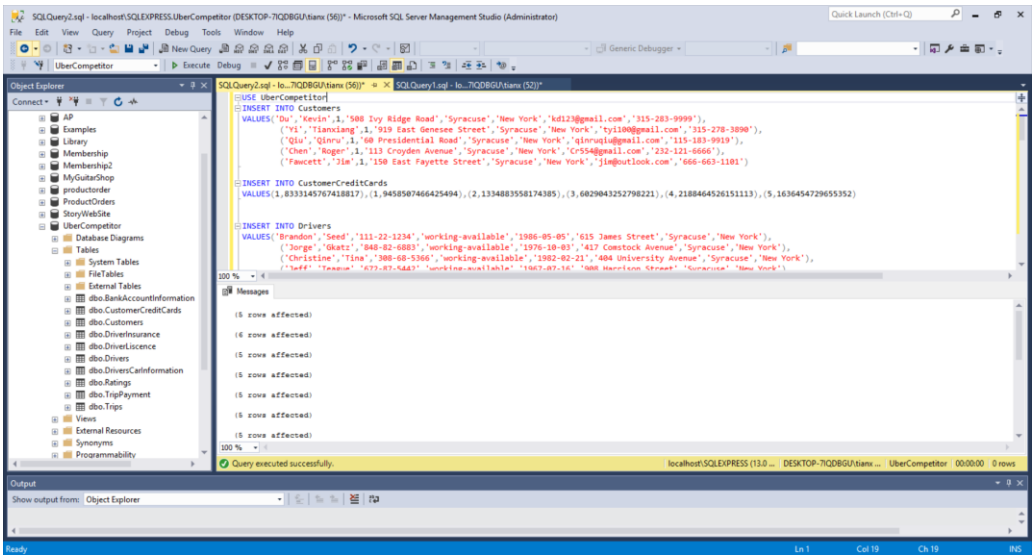
Each table insert 5 records data which has real meaning

```
“INSERT INTO Customers
VALUES('Du','Kevin',1,'508 Ivy Ridge Road','Syracuse','New York','kd123@gmail.com','315-283-9999'),
      ('Yi','Tianxiang',1,'919 East Genesee Street','Syracuse','New York','tyi100@gmail.com','315-278-3890'),
      ('Qiu','Qinru',1,'60 Presidential Road','Syracuse','New York','qinruqiu@gmail.com','115-183-9919'),
      ('Chen','Roger',1,'113 Croyden Avenue','Syracuse','New York','Cr554@gmail.com','232-121-6666'),
      ('Fawcett','Jim',1,'150 East Fayette Street','Syracuse','New York','jim@outlook.com','666-663-1101')

INSERT INTO CustomerCreditCards
VALUES(1,8333145767418817),(1,9458507466425494),(2,1334883558174385),(3,6029043252798221),(4,2188464526151113),(5,1636454729655352)

INSERT INTO Drivers
```

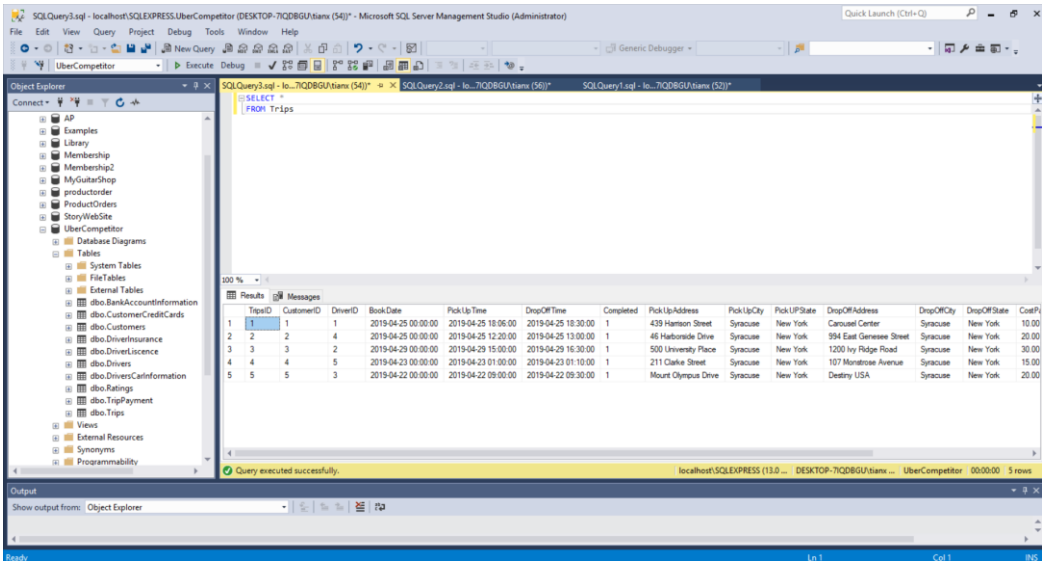
```
VALUES('Brandon','Seed','111-22-1234','working-available','1986-05-05','615 James
Street','Syracuse','New York'),
      ('Jorge','Gkatz','848-82-6883','working-available','1976-10-03','417 Comstock
Avenue','Syracuse','New York'),
      ('Christine','Tina','308-68-5366','working-available','1982-02-21','404 University
Avenue','Syracuse','New York'),
      ('Jeff','Teague','672-87-5442','working-available','1967-07-16','908 Harrison
Street','Syracuse','New York'),
      ('Rodney','Tawfek','602-73-4381','working-available','1988-08-07','205AHawley
Avenue','Syracuse','New York')

INSERT INTO DriversCarInformation
VALUES(1,'Toyotal','Camry',1,'Blue',4,2),
(2,'GMC','Terrian',1,'White',6,4),
(3,'Volkswage','Passat',1,'Black',4,2),
(4,'BMW','X5',1,'Black',4,2),
(5,'Audi','Q9',1,'Black',4,4)

INSERT INTO DriverLiscence
VALUES(1,'New York','A1270461','2015-08-01','2023-08-01'),
(2,'New York','A2272878','2012-10-01','2020-10-01'),
(3,'New York','C5456709','2019-05-01','2027-05-1'),
(4,'New York','A1330424','2017-2-20','2025-2-20'),
(5,'New York','A9626829','2011-11-11','2019-11-11')

INSERT INTO DriverInsurance
VALUES(1,'Progressive','65405181','2018-05-05','2020-05-05'),
(2,'QuoteWizard','26211066','2019-01-10','2022-01-10'),
(3,'State Farm','73026700','2019-02-01','2021-02-01'),
(4,'Esurance','15495412','2017-03-03','2020-03-03'),
(5,'Liberty Mutual','78411751','2018-10-25','2019-10-25')

INSERT INTO BankAccountInformation
VALUES(1,'JP Morgan Chase','322271627','229822663559','Checking'),
(2,'Bank of America','102001017','145492286476','Checking'),
(3,'Citigroup','021100361','856780521214','Saving'),
(4,'Wells Fargo','267084131','843192652788','Saving'),
(5,'Goldman Sachs','998100771','306596591043','Checking')

INSERT INTO TripPayment
VALUES(1,'2019-05-01',90,1),(2,'2019-05-01',80,2),(3,'2019-05-01',85,3),(4,'2019-05-
01',90,4),(5,'2019-05-01',95,5)

INSERT INTO Trips
```

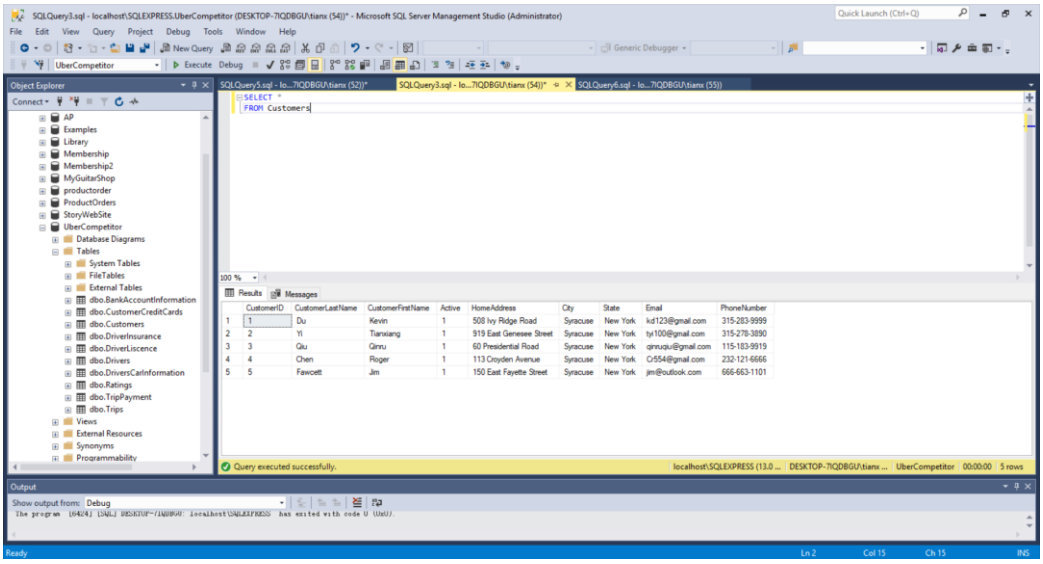
```
VALUES(1,1,'2019-04-25','2019-04-25 18:06','2019-04-25 18:30',1,'439 Harrison Street','Syracuse','New York','Carousel Center','Syracuse','New York',10,2,1,NULL,NULL),
(2,4,'2019-04-25','2019-04-25 12:20','2019-04-25 13:00',1,'46 Harborside Drive','Syracuse','New York','994 East Genesee Street','Syracuse','New York',20,2,3,NULL,NULL),
(3,2,'2019-04-29','2019-04-29 15:00','2019-04-29 16:30',1,'500 University Place','Syracuse','New York','1200 Ivy Ridge Road','Syracuse','New York',30,4.5,4,NULL,NULL),
(4,5,'2019-04-23','2019-04-23 1:00','2019-04-23 1:10',1,'211 Clarke Street','Syracuse','New York','107 Monstrose Avenue','Syracuse','New York',15,5,5,NULL,NULL),
(5,3,'2019-04-22','2019-04-22 9:00','2019-04-22 9:30',1,'Mount Olympus Drive','Syracuse','New York','Destiny USA','Syracuse','New York',20,4,6,NULL,NULL)

INSERT INTO Ratings
VALUES(1,'2019-04-25 18:35',100,NULL),
(2,'2019-04-25 14:30',90,'Smells bad!'),
(3,'2019-04-29 18:30',100,NULL),
(4,'2019-04-25 00:00',50,'Too slow'),
(5,'2019-04-23 09:30',100,NULL)
"
```

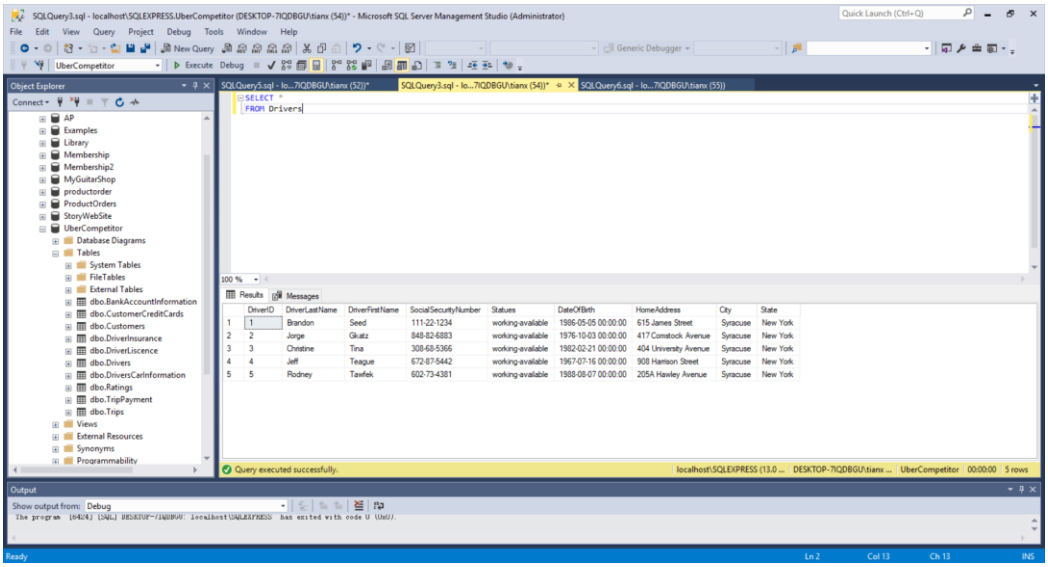
Here is the result of Code query:



Here is 3 important table demonstration:Trips,Customers,Drivers







3.Testing

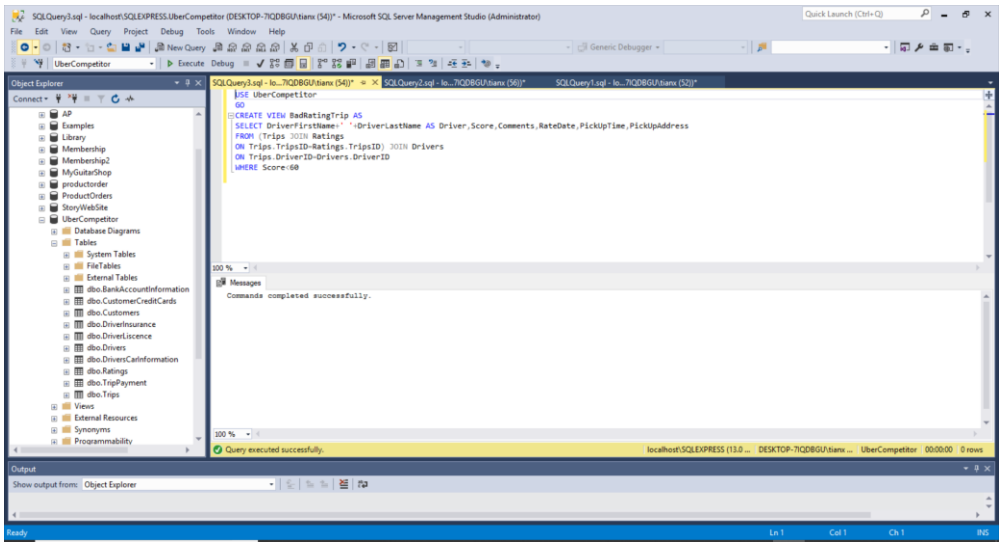
3.1 Views

(1)List of bad comments given by customer, I design not to show Customer Name which is supporting anonymous comments. This view helps customer to feedback to company and other customer to choose better driver.

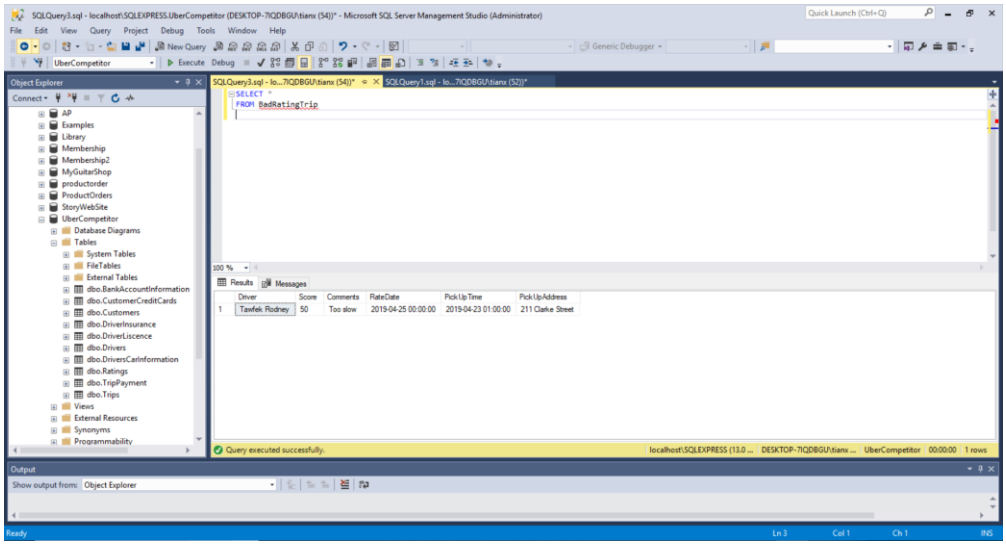
Code:

“
```
USE UberCompetitor
GO
CREATE VIEW BadRatingTrip AS
SELECT DriverFirstName+' '+DriverLastName AS Driver,Score,Comments,RateDate,PickUpTime,PickUpAddress
FROM (Trips JOIN Ratings
ON Trips.TripsID=Ratings.TripsID) JOIN Drivers
ON Trips.DriverID=Drivers.DriverID
WHERE Score<60
GO
SELECT *
FROM BadRatingTrip”
```

Execution Screenshot:



Result Verification

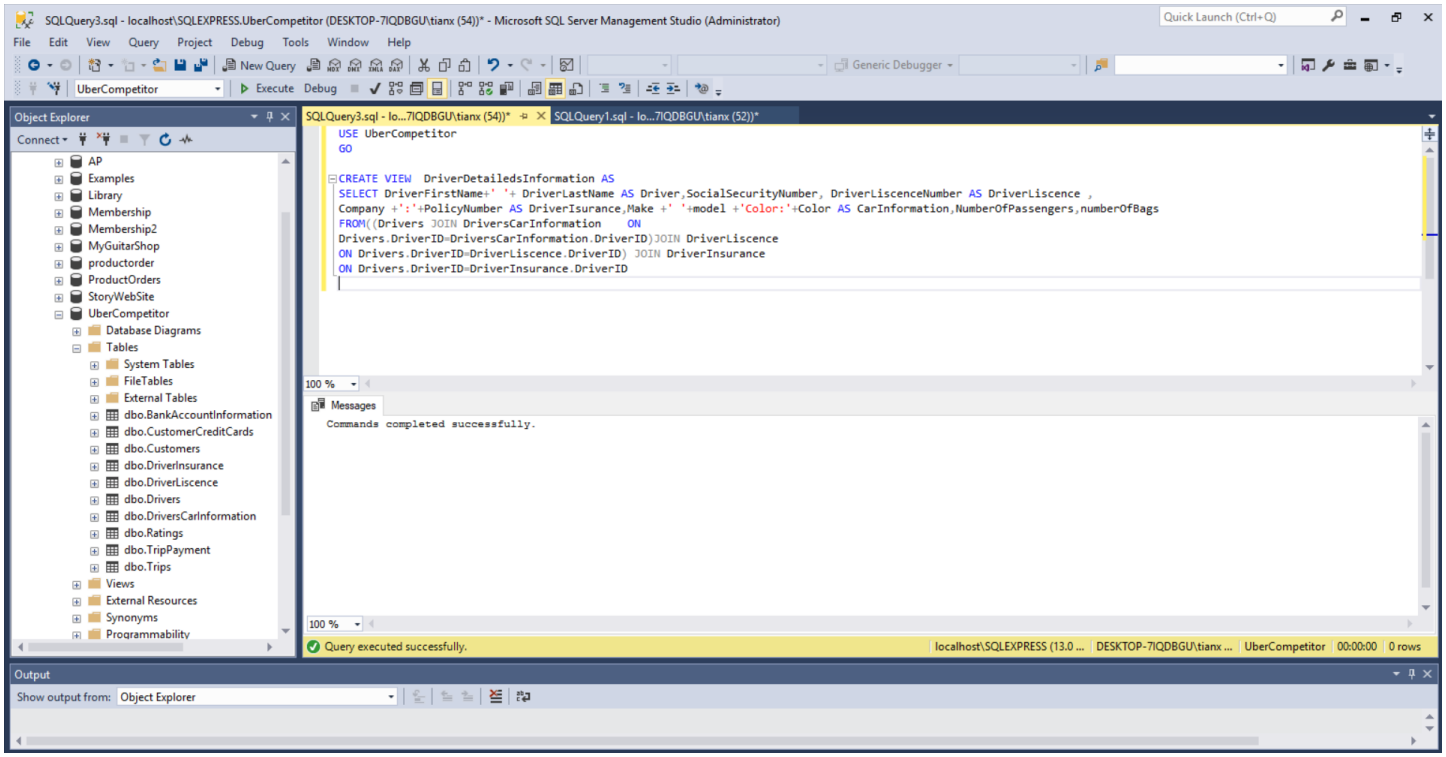


(2) This View help customers to choose the driver that satisfy their requirement(Seats and Bags) and descriminate the car when booked throgh color and make. What's more it could help company to manage driver information more easily.

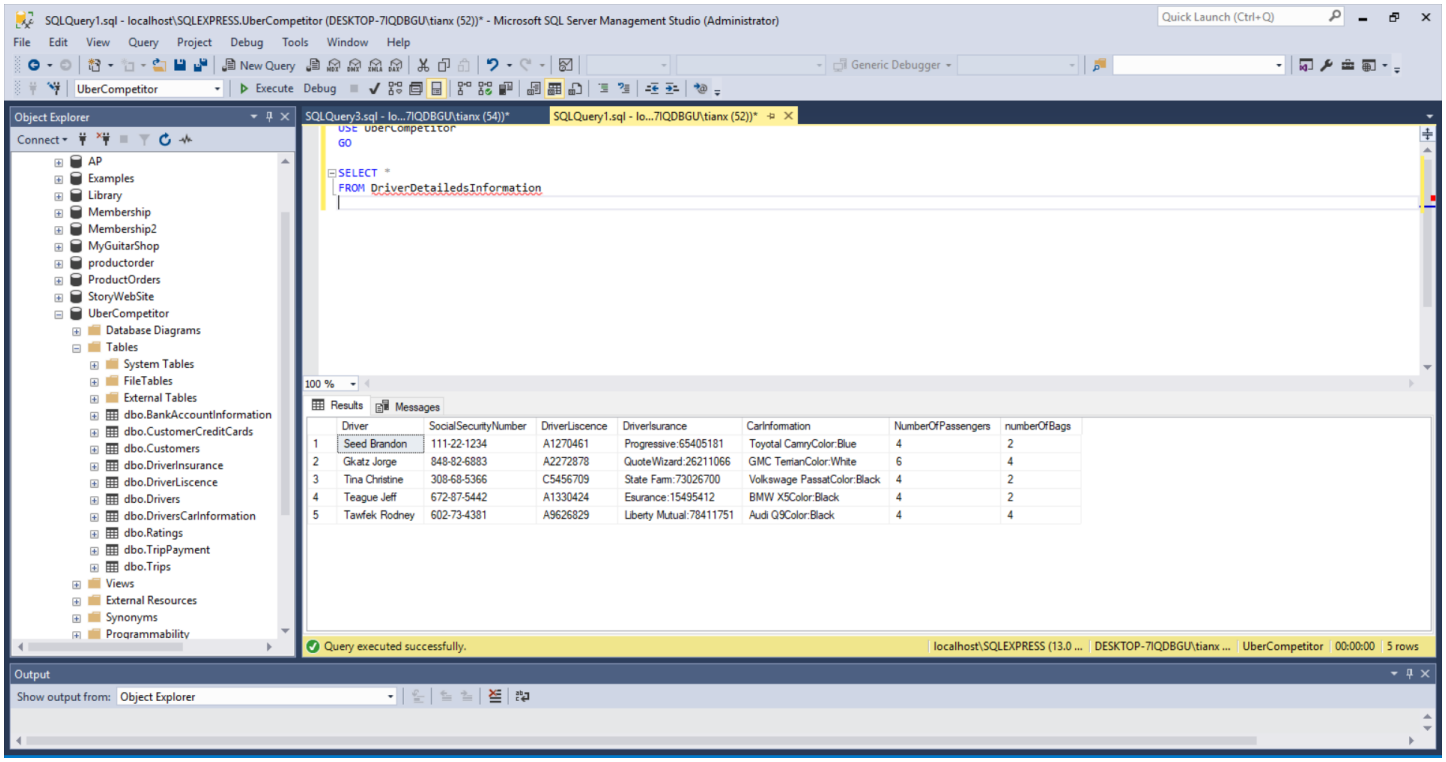
Code:

```
"USE UberCompetitor
GO
CREATE VIEW  DriverDetailedsInformation AS
SELECT DriverFirstName+' '+ DriverLastName AS Driver,SocialSecurityNumber, DriverLiscenceNumber AS
DriverLiscence ,
Company +':'+PolicyNumber AS DriverIsurance,Make +' '+model +'Color:'+Color AS
CarInformation,NumberOfPassengers,numberOfBags
FROM((Drivers JOIN DriversCarInformation ON
Drivers.DriverID=DriversCarInformation.DriverID)JOIN DriverLiscence
ON Drivers.DriverID=DriverLiscence.DriverID) JOIN DriverInsurance
ON Drivers.DriverID=DriverInsurance.DriverID
USE UberCompetitor
GO
SELECT *
FROM DriverDetailedsInformation
"
```

ExecutionScreenshot:



Result Verification:



(3)This View union the customer card information and driver account information.Help company to set its security level higher and carefully for the reason of its importance.

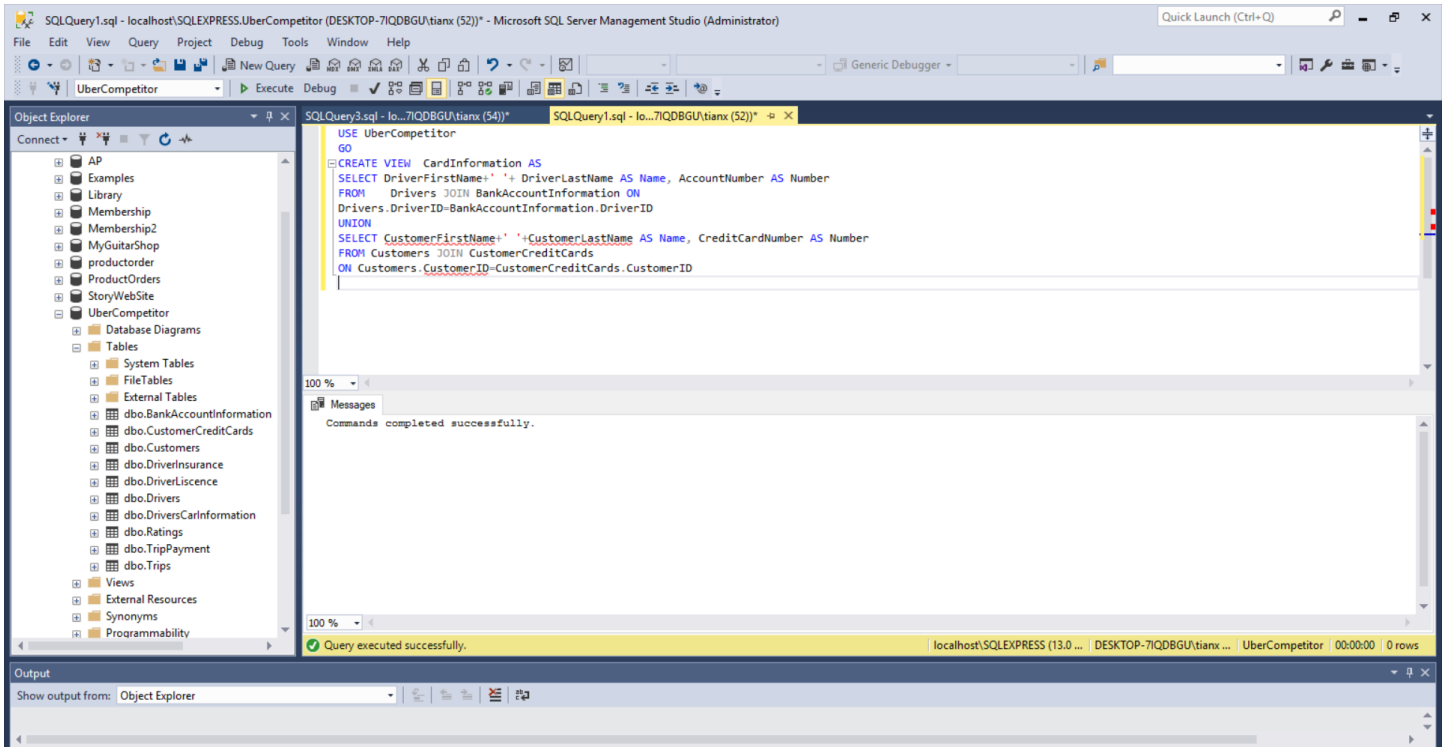
Code:

“USE UberCompetitor

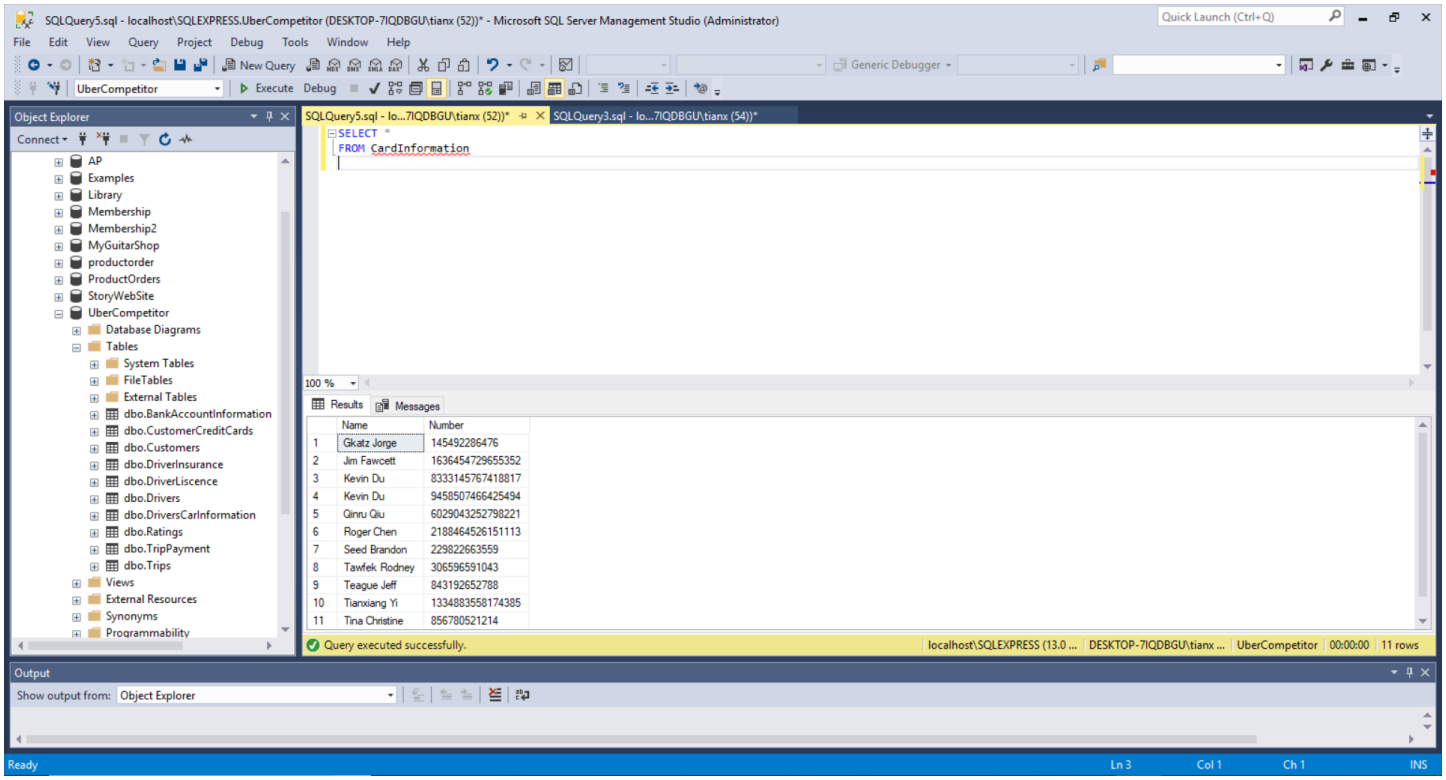
GO

```
CREATE VIEW  CardInformation AS
SELECT DriverFirstName+' '+ DriverLastName AS Name, AccountNumber AS Number
FROMDrivers JOIN BankAccountInformation ON
Drivers.DriverID=BankAccountInformation.DriverID
UNION
SELECT CustomerFirstName+' '+CustomerLastName AS Name, CreditCardNumber AS Number
FROM Customers JOIN CustomerCreditCards
ON Customers.CustomerID=CustomerCreditCards.CustomerID
GO
SELECT *
FROM CardInformation
”
```

Execution Screenshot:



Result Verification:

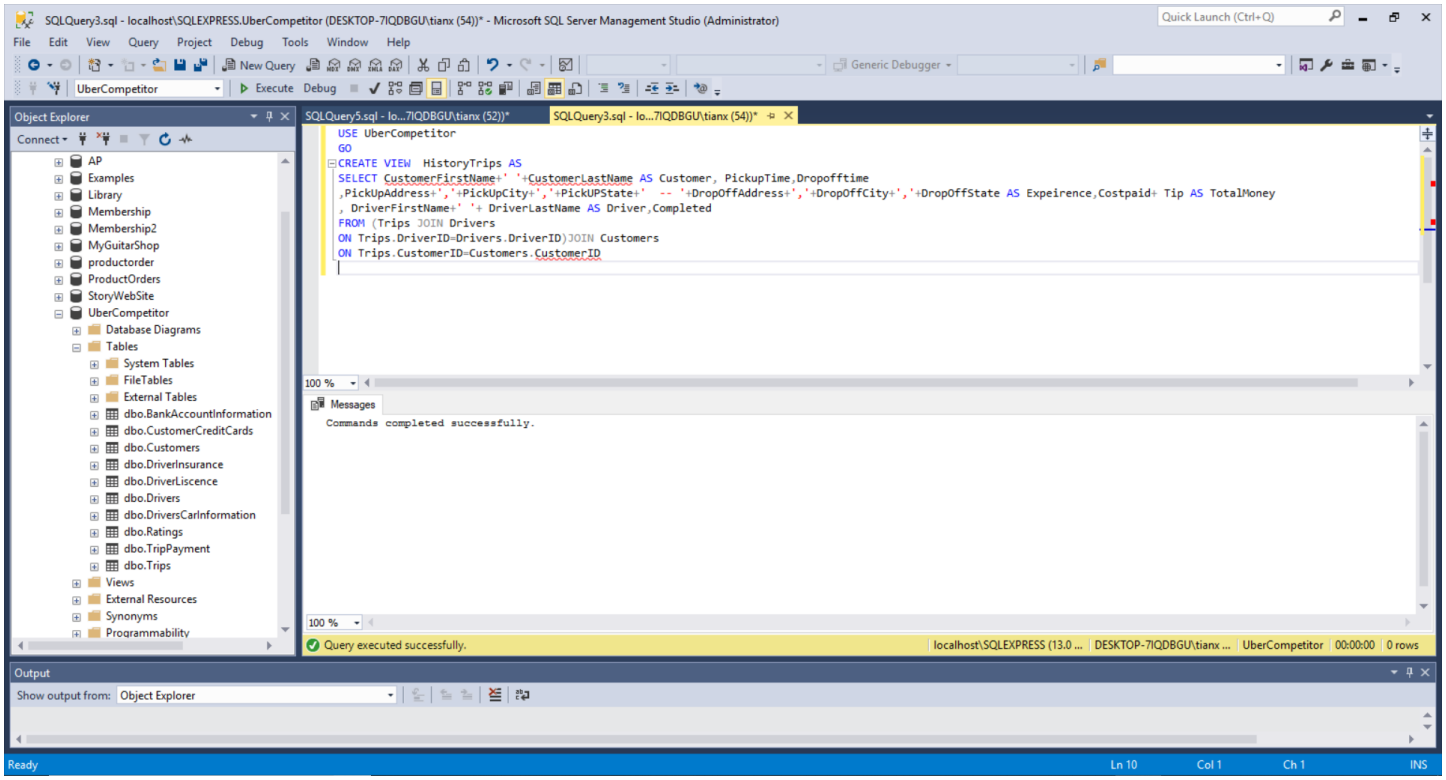


(4)This View is made for customer who want to check his past record of travel and for company to manage all the past record trips.

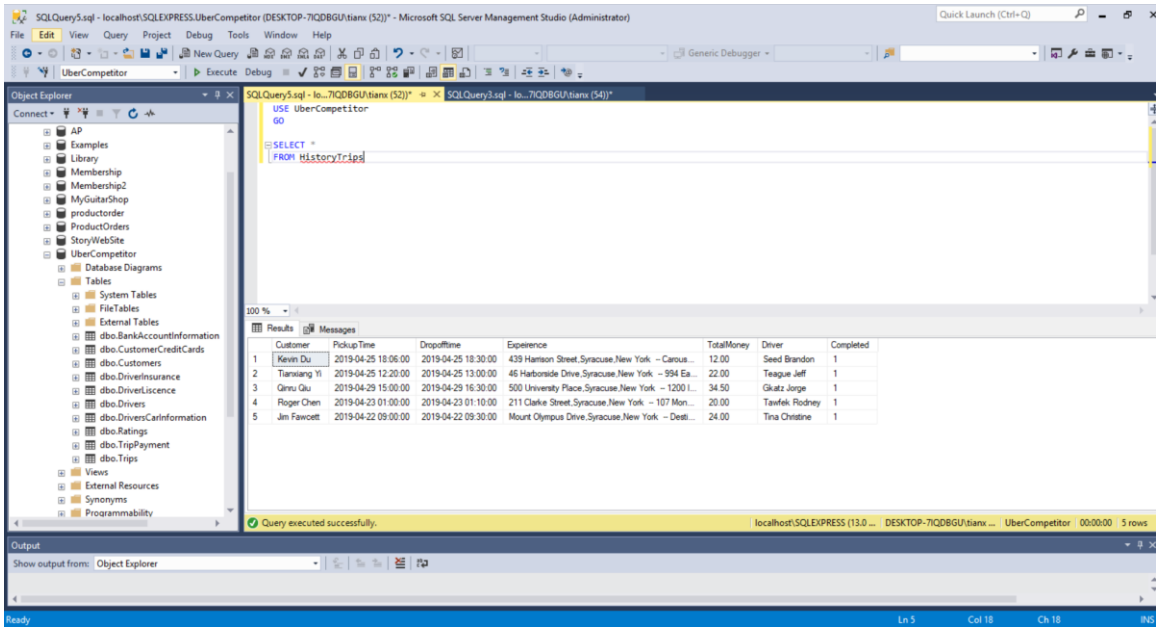
Code:

```
“USE UberCompetitor
GO
CREATE VIEW  HistoryTrips AS
SELECT CustomerFirstName+' '+CustomerLastName AS Customer, PickupTime,Dropofftime
,PickUpAddress+','+PickUpCity+','+PickUPState+'  -- '+DropOffAddress+','+DropOffCity+','+DropOffState
AS Expeirence,Costpaid+ Tip AS TotalMoney
, DriverFirstName+' '+ DriverLastName AS Driver,Completed
FROM (Trips JOIN Drivers
ON Trips.DriverID=Drivers.DriverID)JOIN Customers
ON Trips.CustomerID=Customers.CustomerID
USE UberCompetitor
GO
SELECT *
FROM HistoryTrips
”
```

Execution Screenshot:



Result Verification:



3.2 Functions

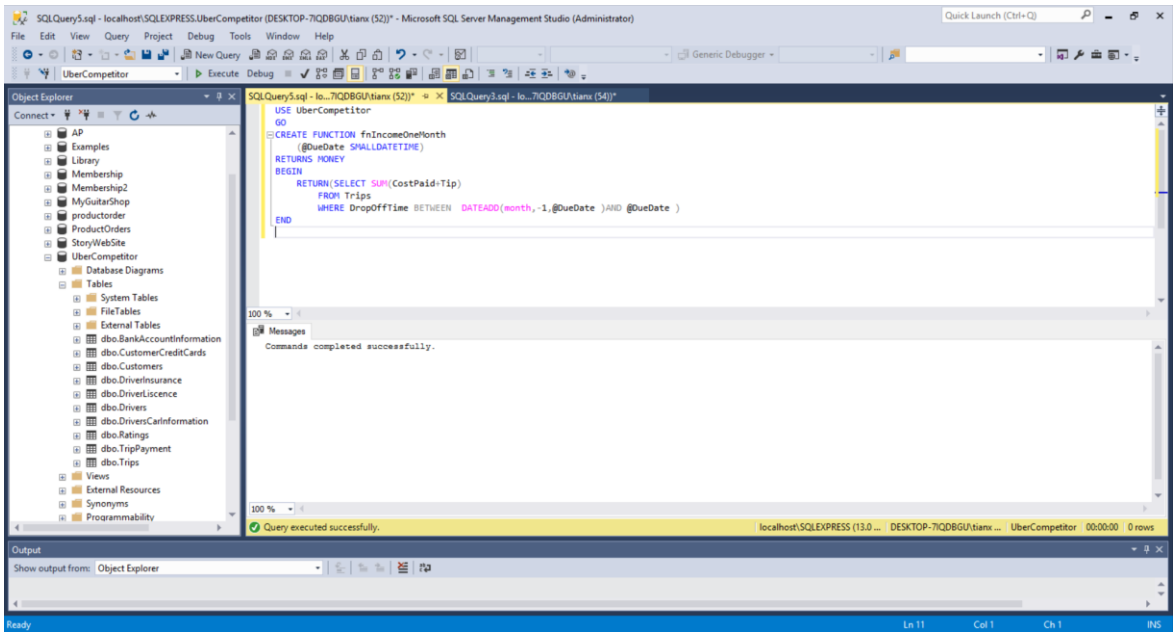
(1)One month income of company

Code:

```
“USE UberCompetitor
GO
CREATE FUNCTION fnIncomeOneMonth
    (@DueDate SMALLDATETIME)
```

```
RETURNS MONEY
BEGIN
    RETURN(SELECT SUM(CostPaid+Tip)
        FROM Trips
        WHERE DropOffTime BETWEEN  DATEADD(month,-1,@DueDate )AND @DueDate )
END”
```

Execution Screenshot:

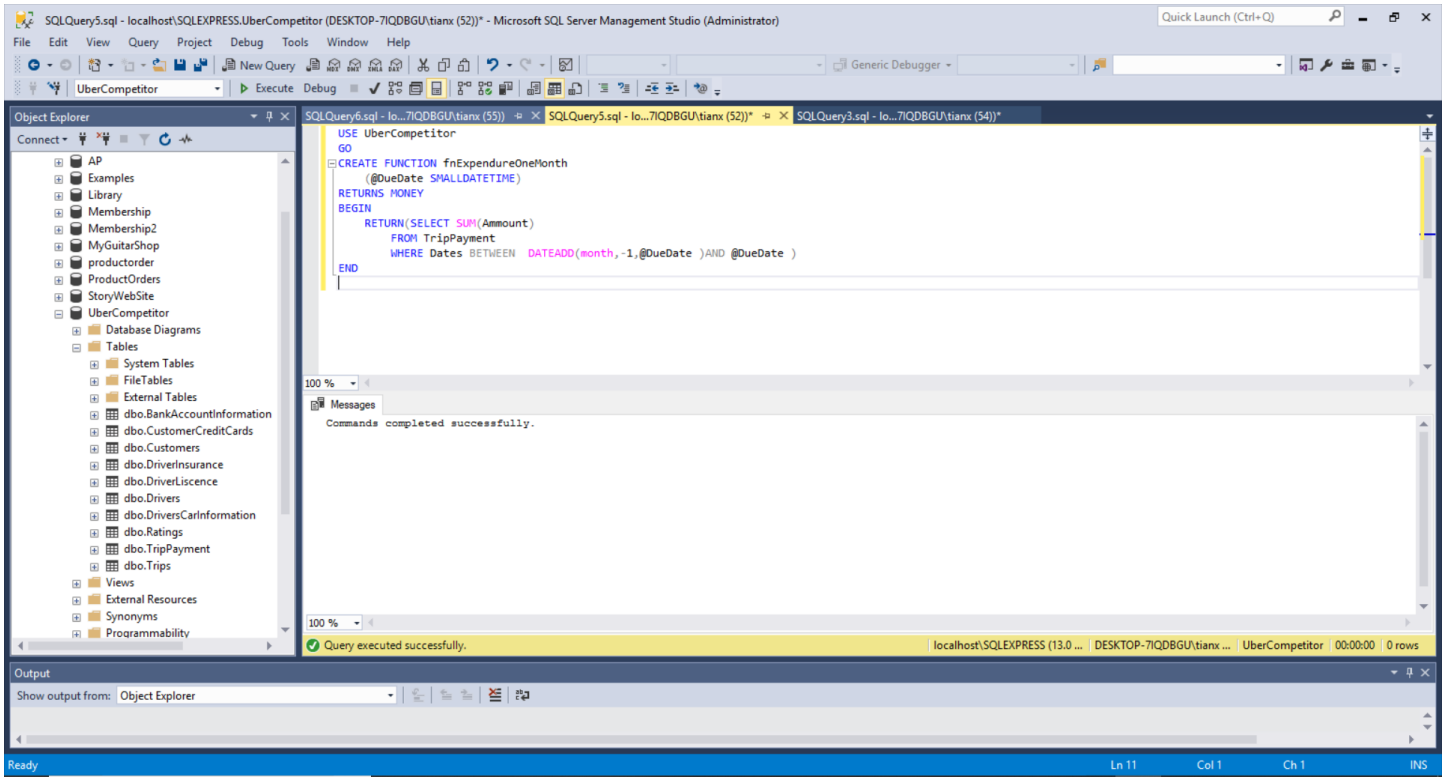


(2)Expendure (give salary to driver)

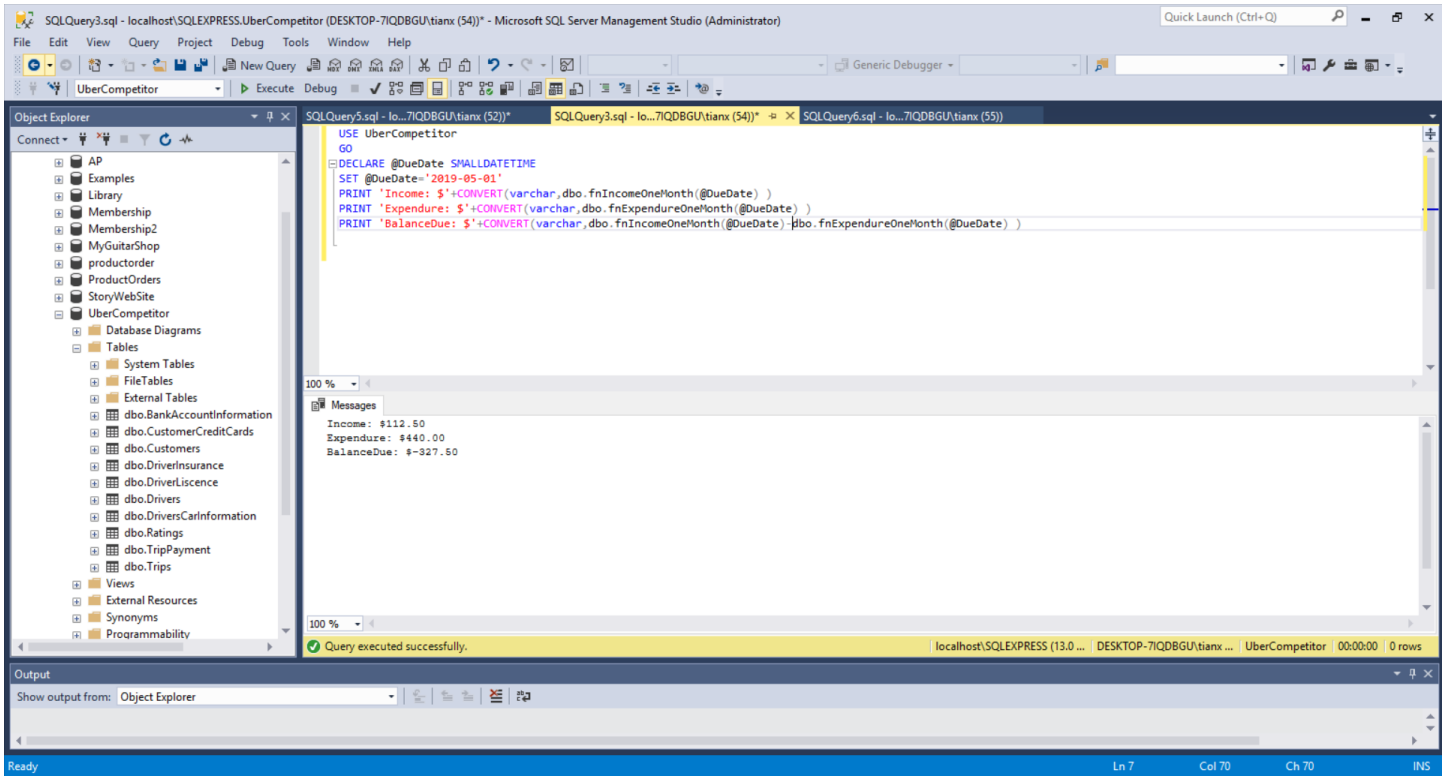
Code:

```
“USE UberCompetitor
GO
CREATE FUNCTION fnExpendureOneMonth
    (@DueDate SMALLDATETIME)
RETURNS MONEY
BEGIN
    RETURN(SELECT SUM(Ammount)
        FROM TripPayment
        WHERE Dates BETWEEN  DATEADD(month,-1,@DueDate )AND @DueDate )
END
”
```

Execution Screenshot:



Result Verification:



Explanation:

In my design of database, driver get their salary every first day of new month. So this function could help company to evaluate what's the total balance due of one month which is income from all tirps minus all drvier amount of the salary.

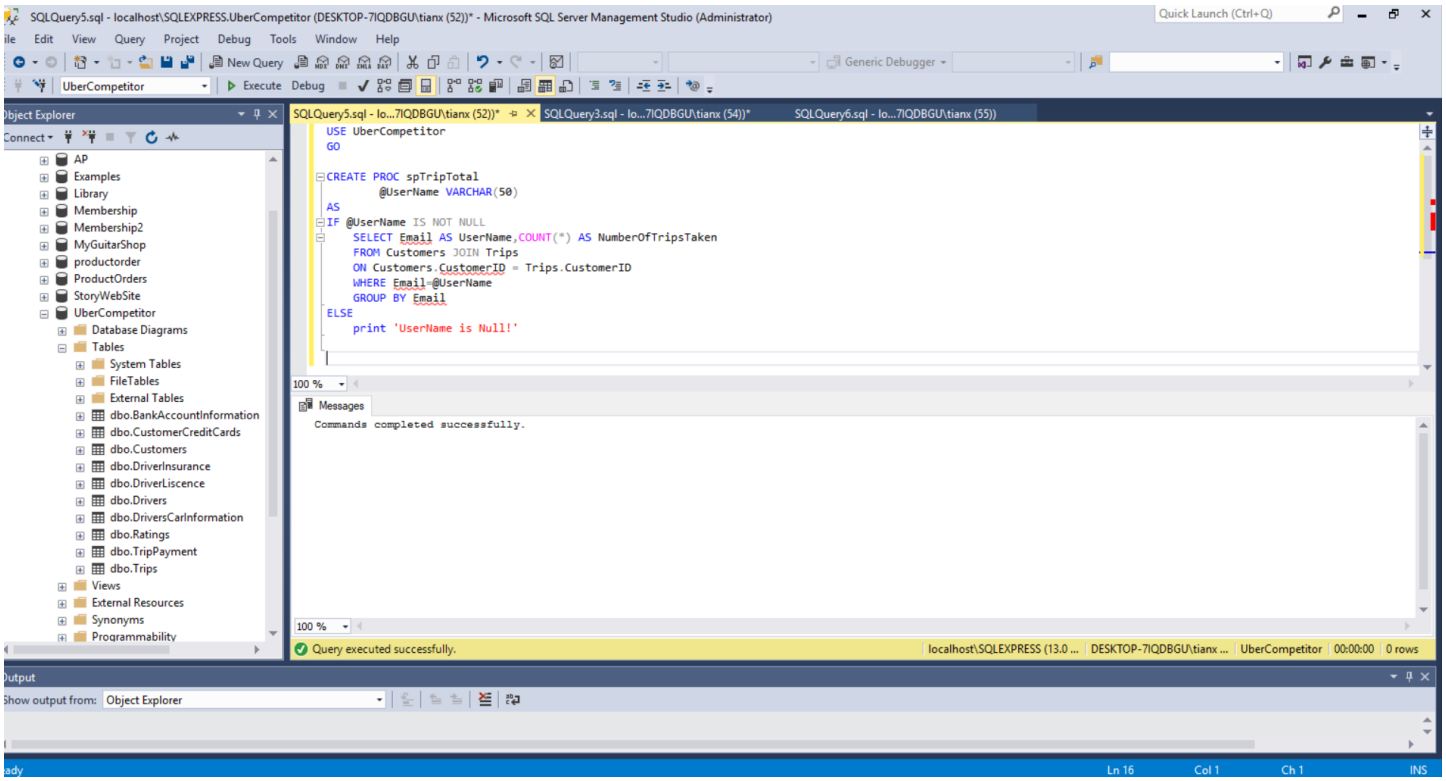
3.3 Stored procedure

(1) This is the procedure to specified customer to find his total number trips more quick and efficient.

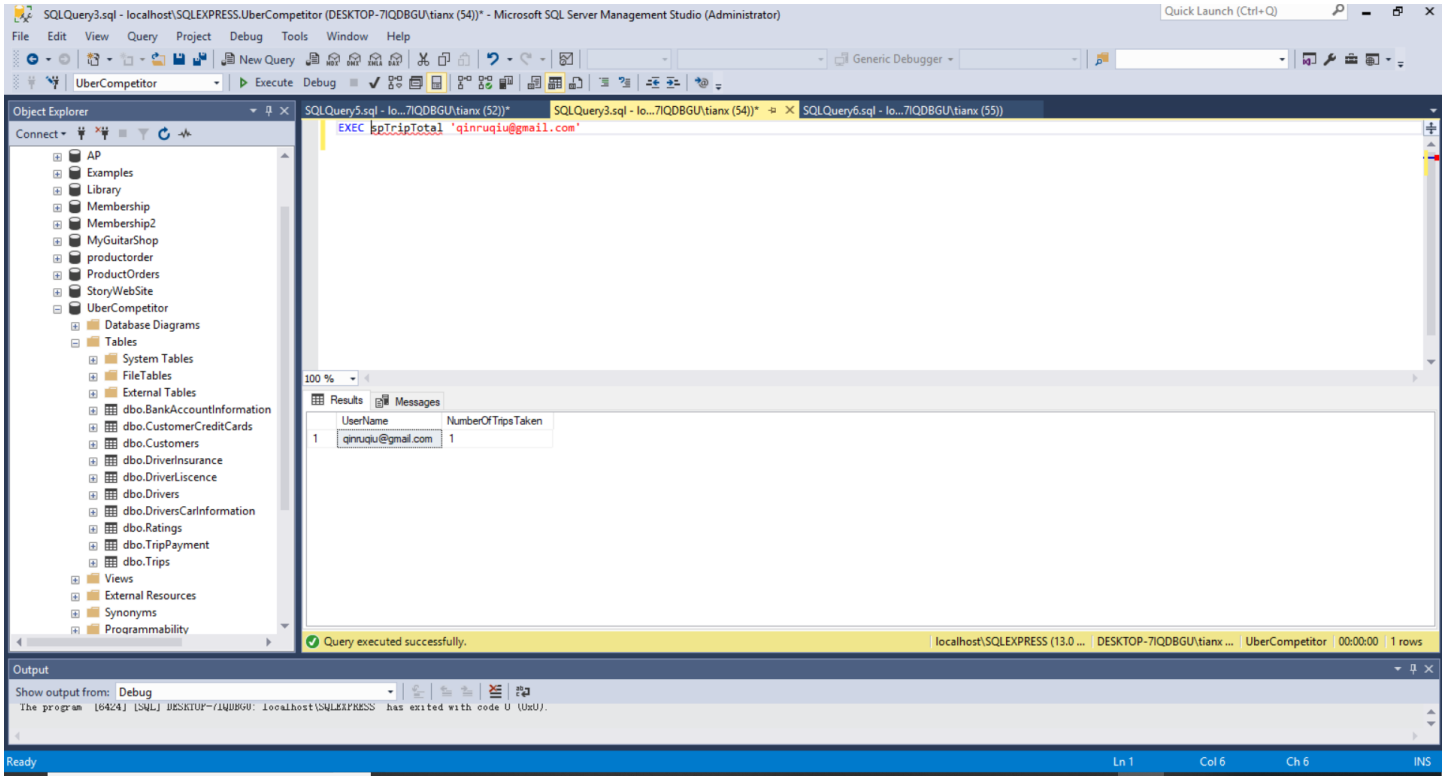
Code:

"USE UberCompetitor
GO
CREATE PROC spTripTotal
        @UserName VARCHAR(50)
AS
IF @UserName IS NOT NULL
    SELECT Email AS UserName,COUNT(*) AS NumberOfTripsTaken
    FROM Customers JOIN Trips
    ON Customers.CustomerID = Trips.CustomerID
    WHERE Email=@UserName
    GROUP BY Email
ELSE
    print 'UserName is Null!'
GO
EXEC spTripTotal 'qinruqiu@gmail.com'
"

Execution Screenshot:



Result Verification:



(2)This is procedure for driver to record his past one month trip. If this variable@Date is date driver got paid, this could be the record for all pickup of the pay.

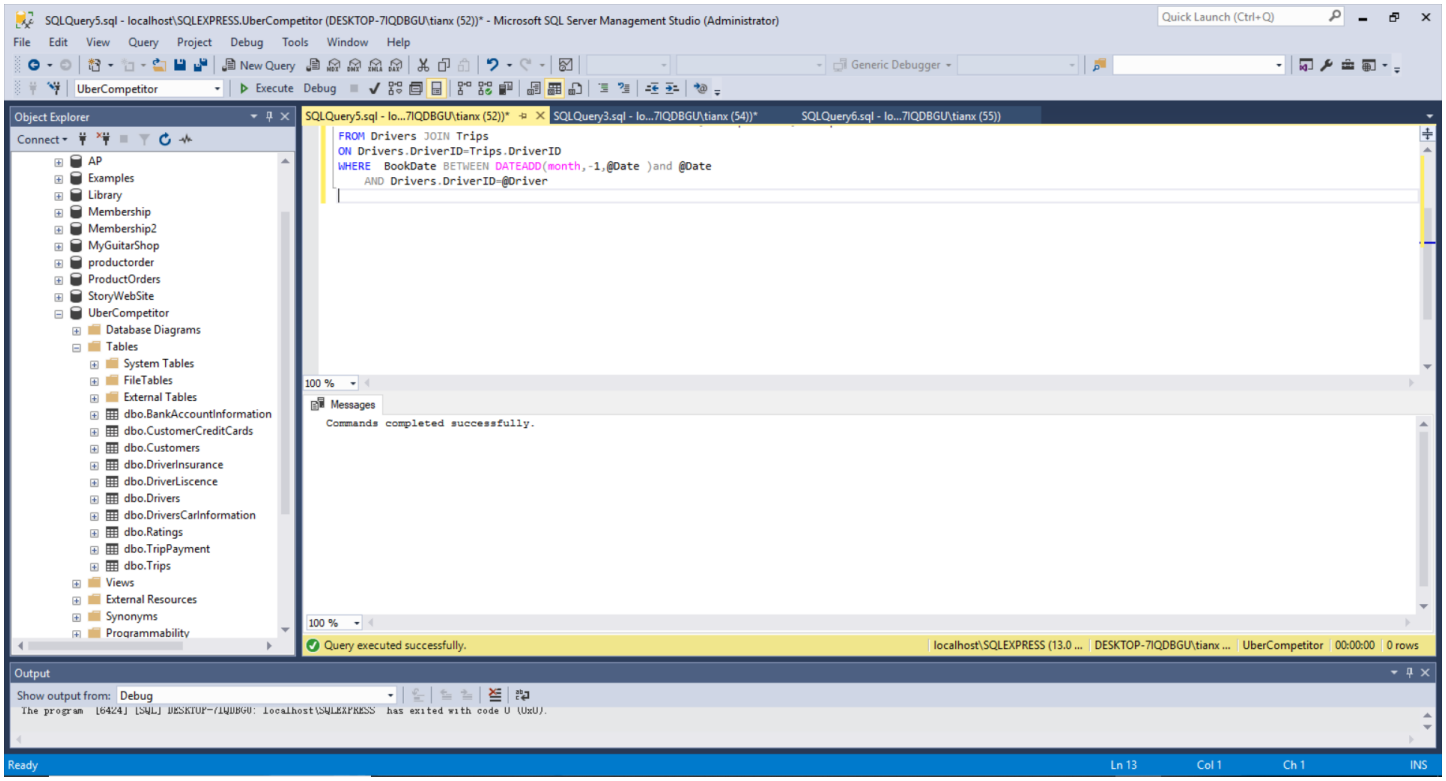
Code:

"USE UberCompetitor
GO

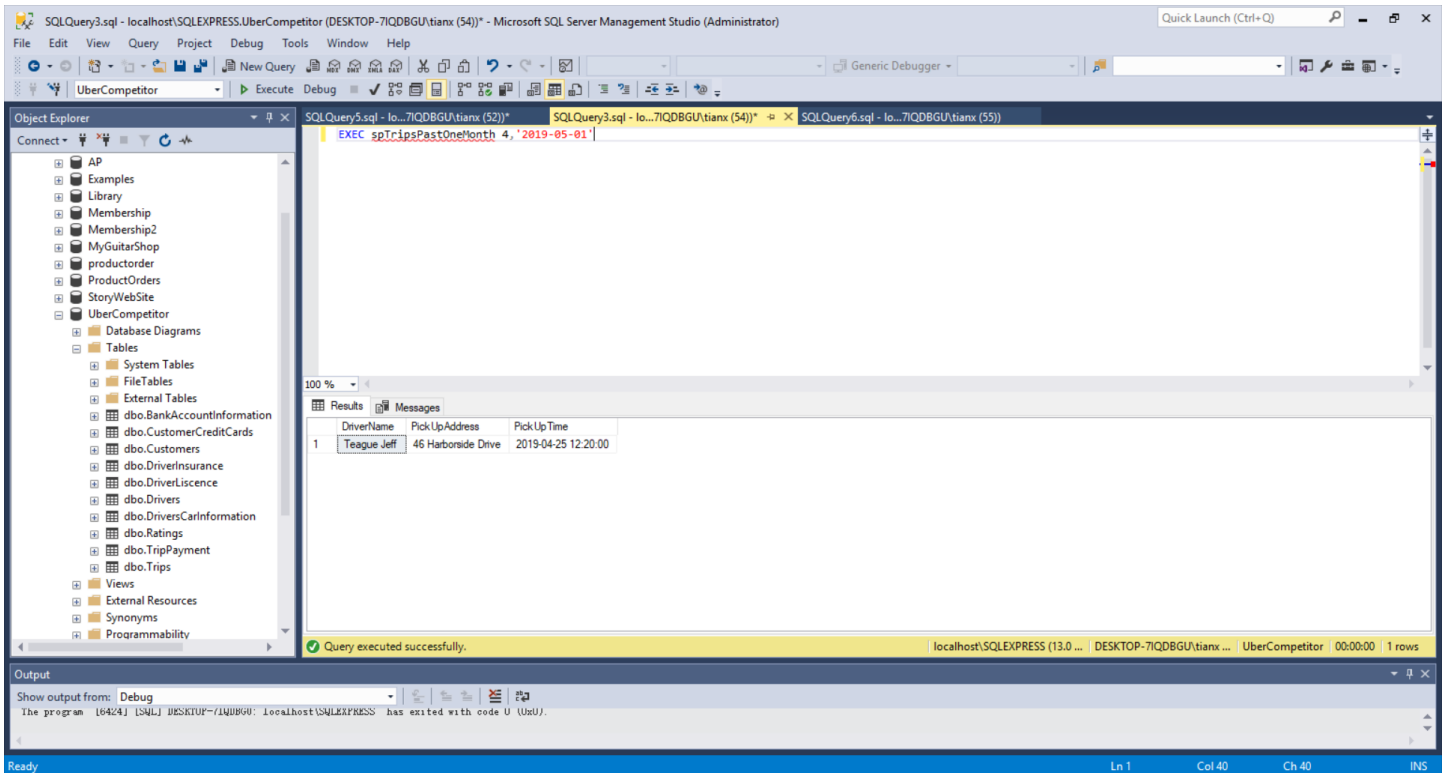
```
CREATE PROC spTripsPastOneMonth
    @Driver INT,
    @Date SMALLDATETIME
AS
SELECT DriverFirstName+' '+DriverLastName AS DriverName,PickUpAddress,PickUpTime
FROM Drivers JOIN Trips
ON Drivers.DriverID=Trips.DriverID
WHERE  BookDate BETWEEN DATEADD(month,-1,@Date )and @Date
    AND Drivers.DriverID=@Driver
GO
EXEC spTripsPastOneMonth 4,'2019-05-01'
```

"

Execution Screenshot:



Result Verification:



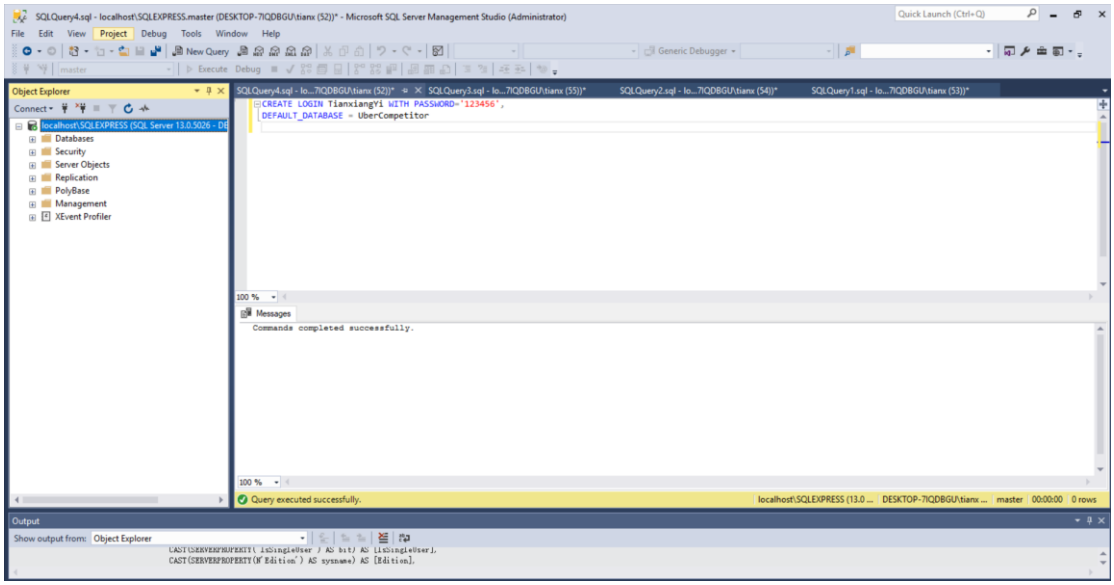
3.4 Scripts

(1)Password assignment

Code:

‘CREATE LOGIN TianxiangYi WITH PASSWORD='123456',
DEFAULT_DATABASE = UberCompetitor’

Execution Screenshot:

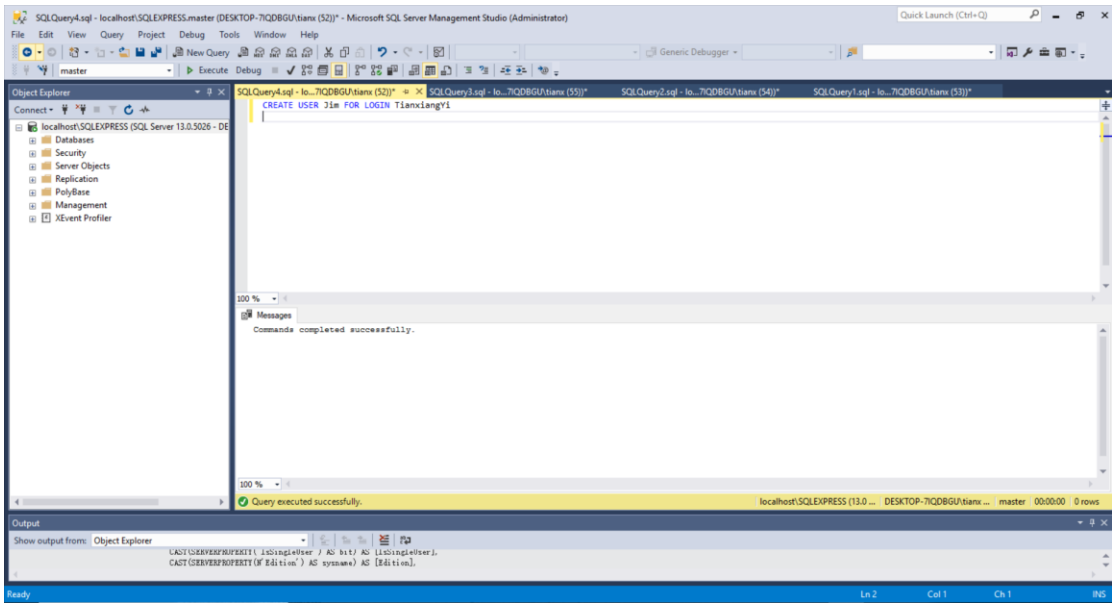


(2)User roles

Code:

```
CREATE USER Jim FOR LOGIN TianxiangYi
```

Execution Screenshot:



4. Conclusion

This is all the Uber Competitor Database I designed. It include everything provided in Project 2 Statement file.
It could access data from client side, driver side. The test situation looks well without error. The backward may be the data loading volum is too small. For futher modify and implementation should wait for it to use.

Remarks
The requirement of this Uber competitor database is all satisfied. It work well with test data.