**DATA MANAGEMENT AND DATABASE DESIGN**

**BOSTON TRAIN OPERATIONS MANAGEMENT SYSTEM**

**TEAM MAVERICKS**

**Team Members**

* Additya Dharangaonkar : NUID – 001052304
* Samarth Hadawale : NUID – 001053811
* Rohan Kapadnis : NUID – 001342161
* Priyanka Pal : NUID – 001059632
* Parthiv Shah : NUID – 001059325

**Background**

The Massachusetts Bay Transport Authority (popularly known as the T) is a public transport agency responsible for operating most public transportation services in Greater Boston, Massachusetts. It operates various major types of mass transit vehicles such as Light Rail, Heavy Rail, Regional Rail Trains and Electric Trolley Buses running on total length of 1193 miles & serving daily ridership of 1.3 million. As there is a huge number of passengers using the T, especially in Boston city, between 5 major lines there needs to be a smooth interaction between the MBTA database and the passengers. As there is a large amount of train services and passenger data, it is important to merge and store the data in database. This database system will facilitate easy viewing of the schedule and booking of the tickets as well as reduce the time consumed in the process by the passenger by providing a quicker service.

**Mission Statement**

Developing a centralized Train Management Database System for the City of Boston, this will facilitate functionalities like Searching schedule, Booking and Cancellation of tickets with the aim of improving upon the current system by saving passenger’s time & efforts in the process.

**VIEW NO. 1**

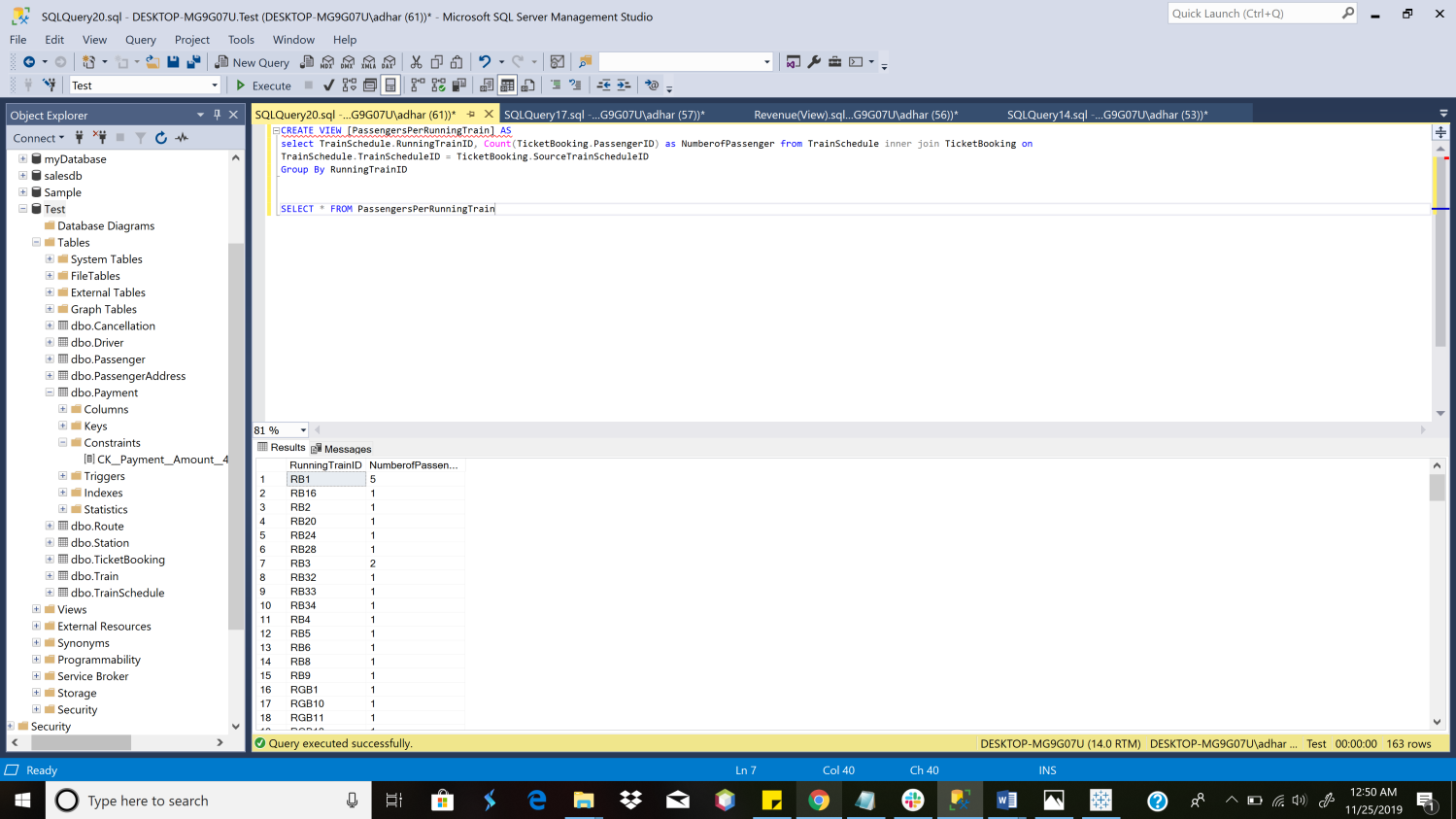
CREATEVIEW [PassengersPerRunningTrain] AS

selectTrainSchedule.RunningTrainID,Count(TicketBooking.PassengerID)asNumberofPassengerfromTrainScheduleinnerjoinTicketBookingon

TrainSchedule.TrainScheduleID=TicketBooking.SourceTrainScheduleID

GroupBy RunningTrainID

SELECT\*FROM PassengersPerRunningTrain



**VIEW [PassengersPerRunningTrain]**

**View No. 2**

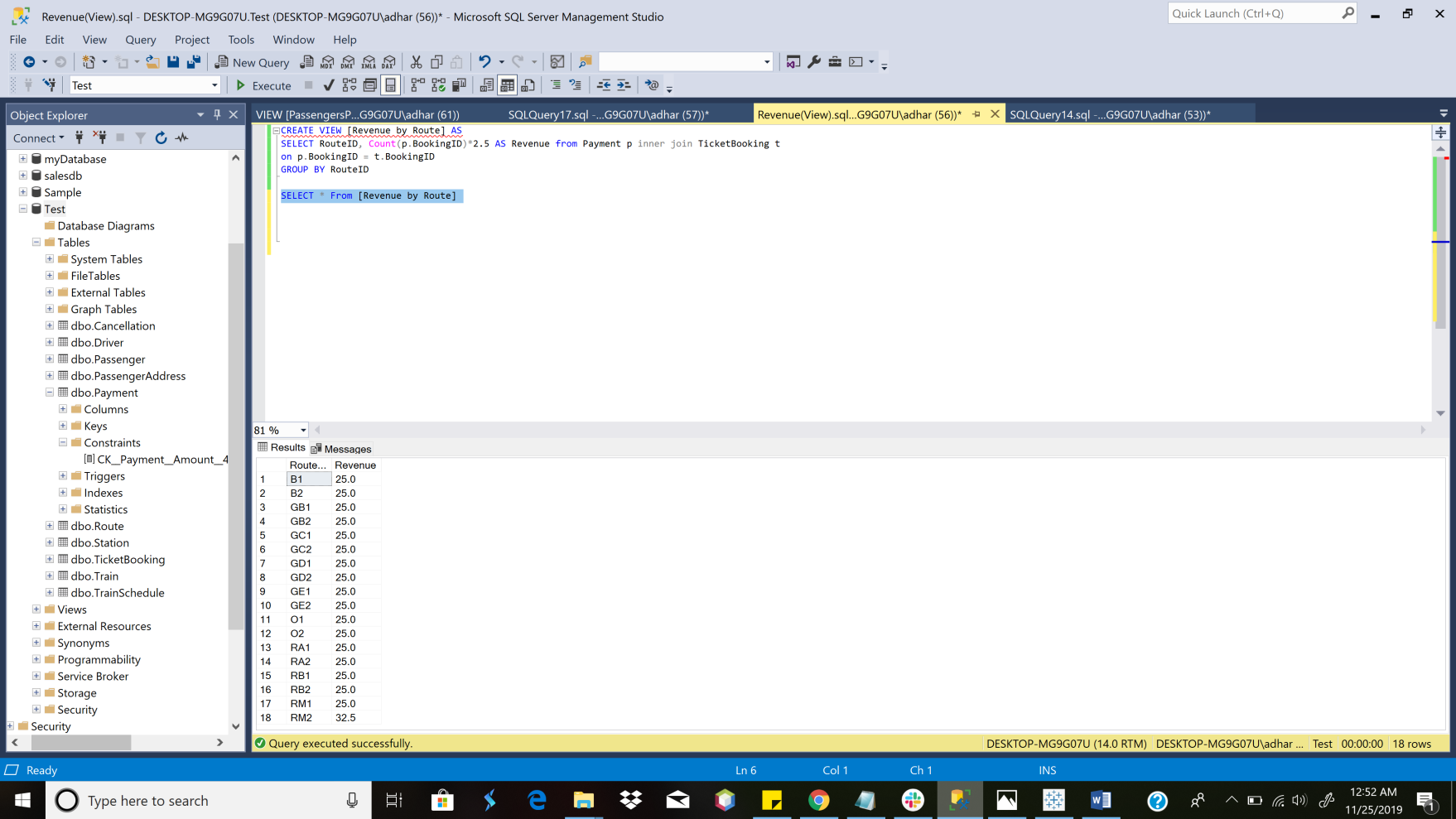
CREATEVIEW [Revenue by Route] AS

SELECTRouteID,Count(p.BookingID)\*2.5 AS Revenue from Payment p innerjoinTicketBooking t

onp.BookingID=t.BookingID

GROUPBYRouteID

SELECT\*From [Revenue by Route]



**VIEW [Revenue Per Route]**

**Column Data Encryption**

USEmaster;

GO

SELECT\*

FROMsys.symmetric\_keys

WHEREname='##MS\_ServiceMasterKey##';

GO

-- Create database Key

USE Test;

GO

CREATEMASTERKEYENCRYPTIONBYPASSWORD='Password123';

GO

-- Create self signed certificate

USE Test;

GO

CREATECERTIFICATE Certificate1

WITHSUBJECT='Protect Data';

GO

-- Create symmetric Key

USE Test;

GO

CREATESYMMETRICKEY SymmetricKey1

WITHALGORITHM=AES\_128

ENCRYPTIONBYCERTIFICATE Certificate1;

GO

USE Test;

GO

ALTERTABLE Payment

ADDCredit\_card\_number\_encryptvarbinary(MAX)NULL

GO

-- Populating encrypted data into new column

USE Test;

GO

-- Opens the symmetric key for use

OPENSYMMETRICKEY SymmetricKey1

DECRYPTIONBYCERTIFICATE Certificate1;

GO

UPDATE Payment

SETCredit\_card\_number\_encrypt=EncryptByKey(Key\_GUID('SymmetricKey1'),CardNumber)

FROM Payment;

GO

-- Closes the symmetric key

CLOSESYMMETRICKEY SymmetricKey1;

GO

SELECT\*FROM Payment

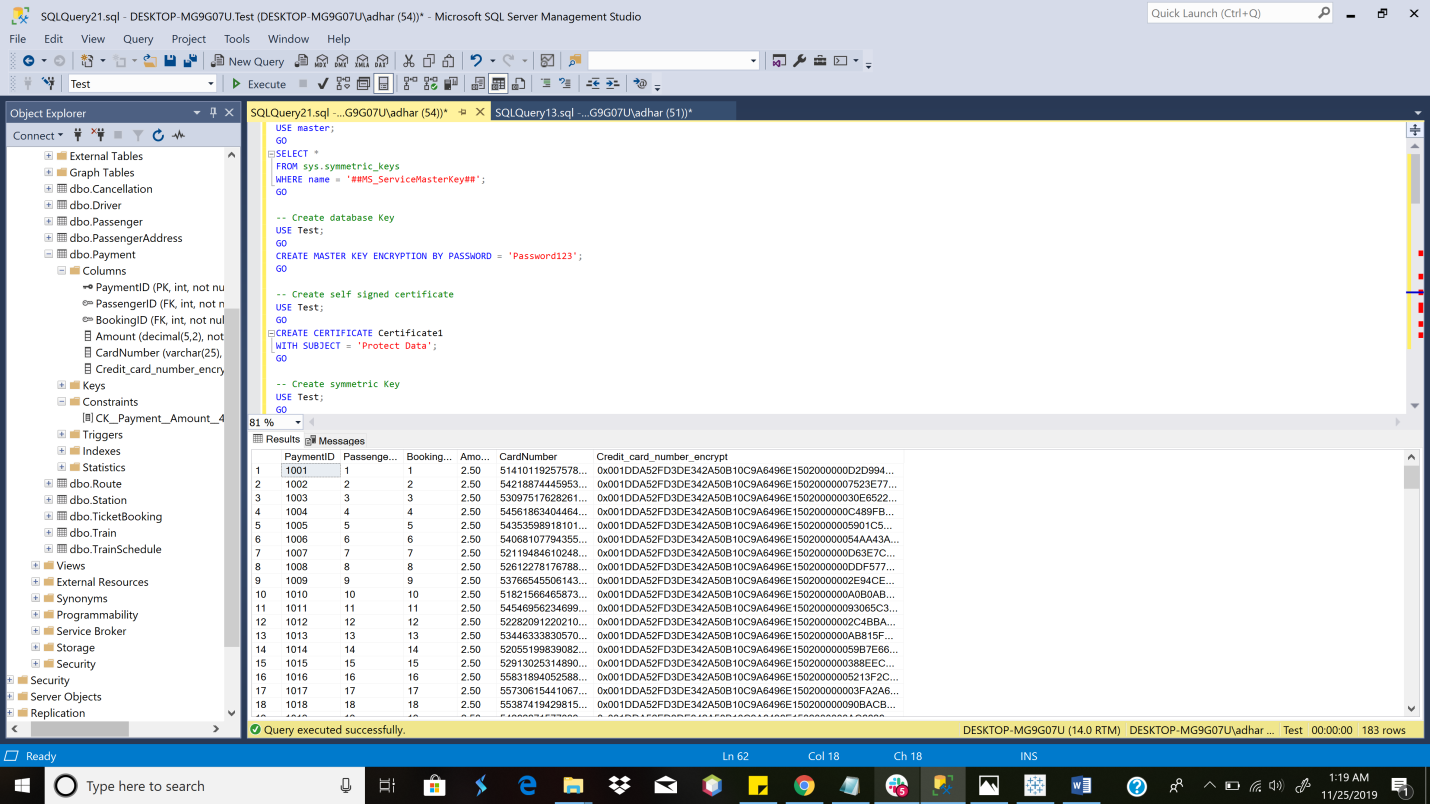
USE Test;

GO

ALTERTABLECustomer\_data

DROPCOLUMNCredit\_card\_number;

GO

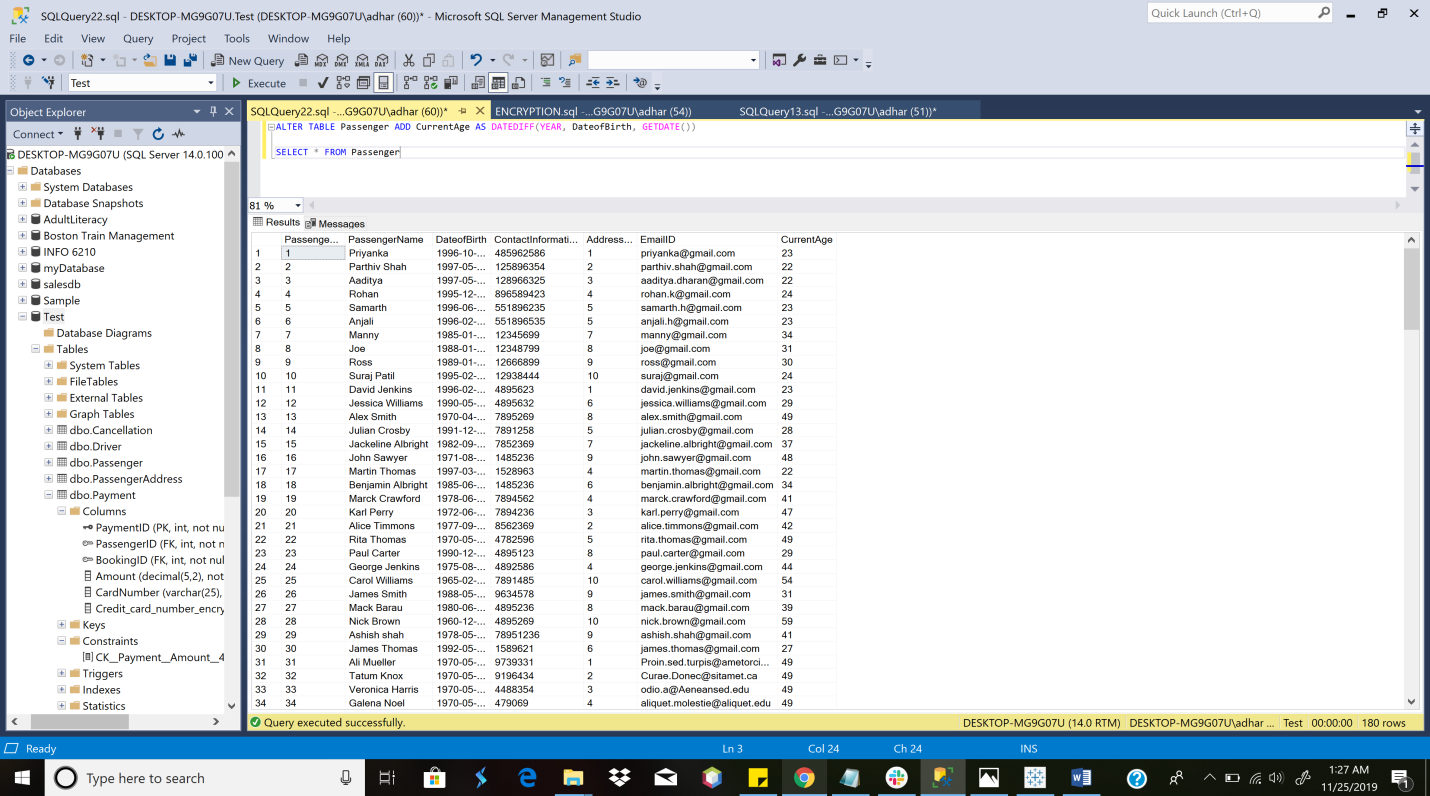


**COLUMN DATA ENCRYPTION**

**Computed Columns based on a Function**

ALTERTABLE Passenger ADDCurrentAgeASDATEDIFF(YEAR,DateofBirth,GETDATE())

SELECT\*FROM Passenger

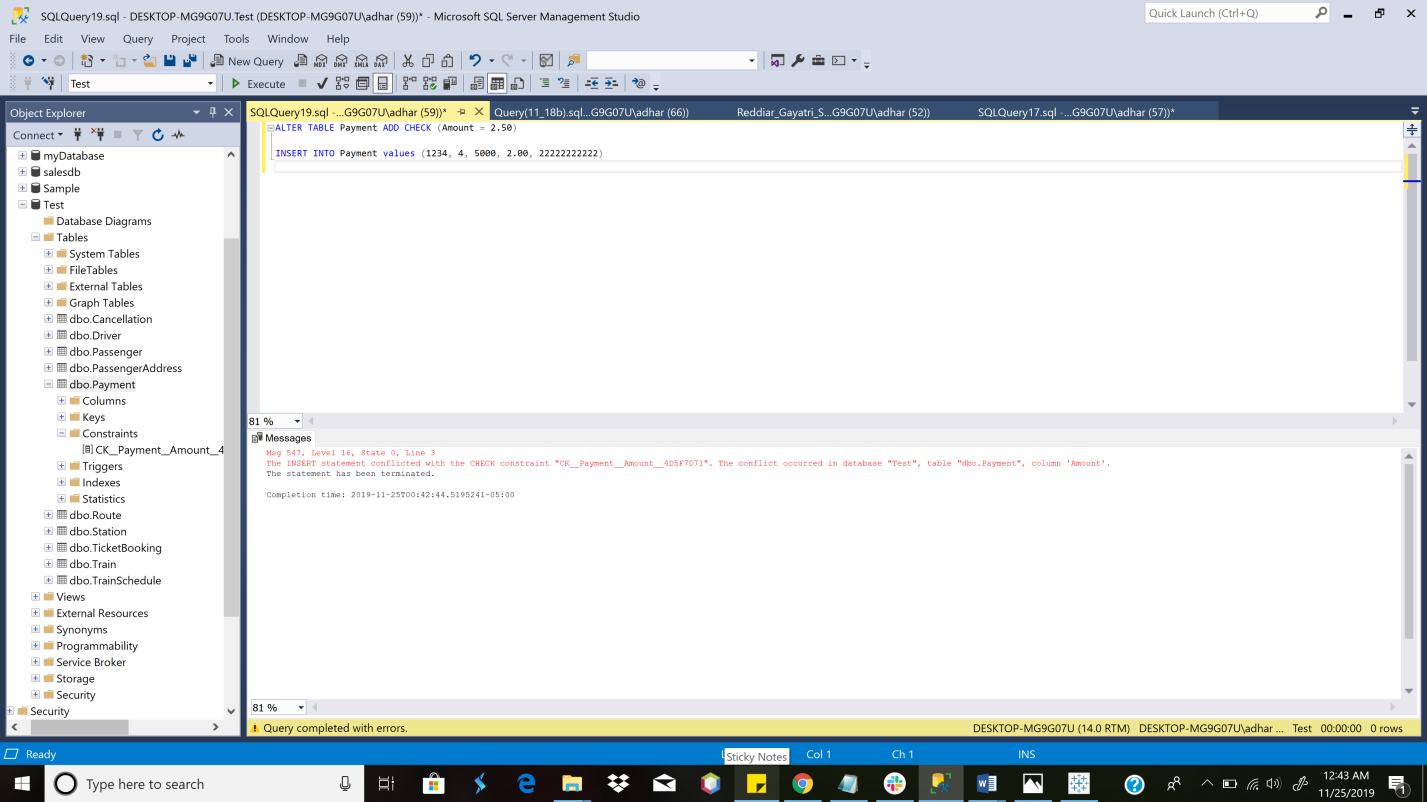


**Computed Columns [based on a Function]**

**TABLE LEVEL CHECK CONSTRAINT**

ALTERTABLE Payment ADDCHECK (Amount = 2.50)

INSERTINTO Payment values (1234, 4, 5000, 2.00, 22222222222)



**TABLE LEVEL CHECK CONSTRAINT ON PAYMENT TABLE**

**STORED PROCEDURE TrainSchedule OUTBOUND**

CREATE PROCEDURE TrainScheduleOutbound @Source VARCHAR(25), @Destination VARCHAR(25), @StartTimeTIME, @EndTimeTIME

AS

BEGIN

SELECT t1.trainID, s1.StationName,t1.Arrival, s2.StationName, t1.[Date]

FROMTrainSchedule t1 innerjoinTrainSchedule t2 on t1.RunningTrainID= t2.RunningTrainID and t1.RouteID = t2.RouteID

innerjoin Station s1 on s1.StationStopID=t1.StationStopIDinnerjoin Station s2 on s2.StationStopID=t2.StationStopID

innerjoinRoute r on s1.RouteID=r.RouteID

where (t1.Arrivalbetween @StartTimeand @EndTime)and s1.StationName = @Source and s2.StationName=@Destination and(t1.RouteID='GE1'OR t1.RouteID='GD1'OR t1.RouteID='GC1'OR t1.RouteID='GB1'

OR t1.RouteID='B1'OR t1.RouteID='RA1'OR t1.RouteID='O1'OR t1.RouteID='RB1'OR t1.RouteID='RM1')

END

EXECUTETrainScheduleOutbound

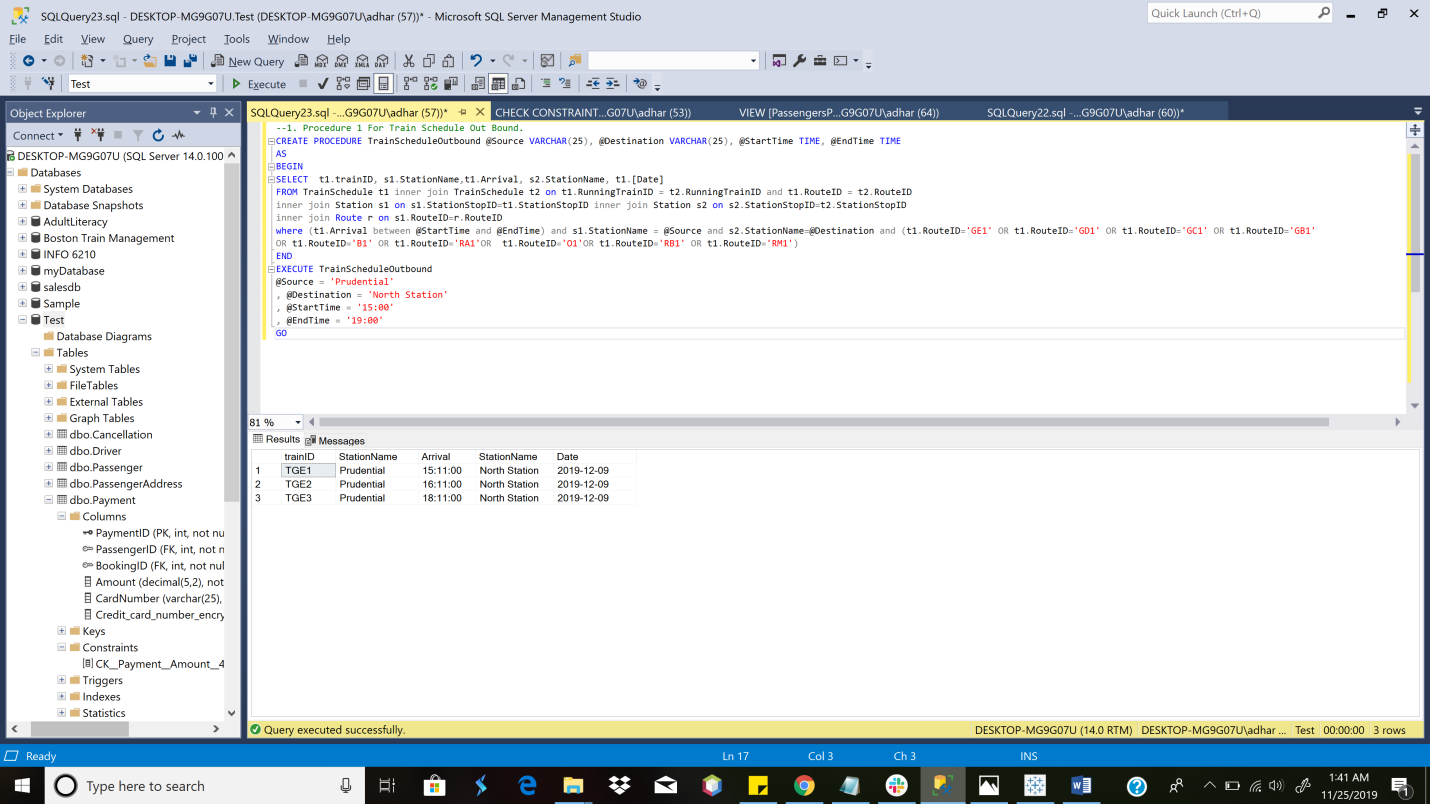
@Source ='Prudential'

, @Destination ='North Station'

, @StartTime='15:00'

, @EndTime='19:00'

GO



**Stored Procedure For Train Schedule Out Bound**

**STORED PROCEDURE TrainScheduleInbound**

CREATE PROCEDURE TrainScheduleInbound @Source VARCHAR(25), @Destination VARCHAR(25), @StartTimeTIME, @EndTimeTIME

AS

BEGIN

SELECT t1.trainID, s1.StationName,t1.Arrival, s2.StationName, t1.[Date]

FROMTrainSchedule t1 innerjoinTrainSchedule t2 on t1.RunningTrainID= t2.RunningTrainID and t1.RouteID = t2.RouteID

innerjoin Station s1 on s1.StationStopID=t1.StationStopIDinnerjoin Station s2 on s2.StationStopID=t2.StationStopID

innerjoinRoute r on s1.RouteID=r.RouteID

where (t1.Arrivalbetween @StartTimeand @EndTime)and s1.StationName = @Source and s2.StationName=@Destination and(t1.RouteID='GE2'OR t1.RouteID='GD2'OR t1.RouteID='GC2'OR t1.RouteID='GB2'

OR t1.RouteID='B2'OR t1.RouteID='RA2'OR t1.RouteID='O2'OR t1.RouteID='RB2'OR t1.RouteID='RM2')

END

EXECUTETrainScheduleInbound

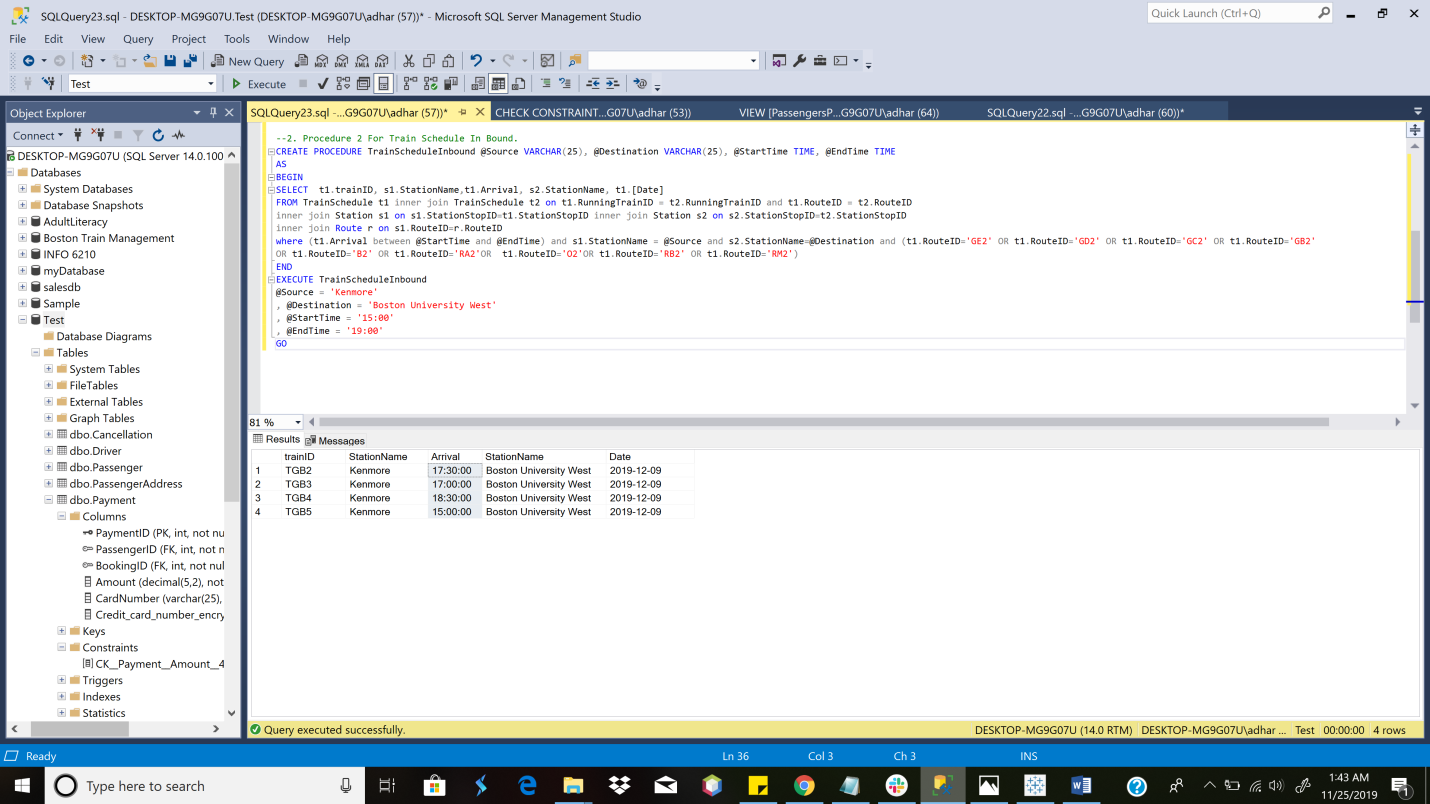
@Source ='Kenmore'

, @Destination ='Boston University West'

, @StartTime='15:00'

, @EndTime='19:00'

GO



**Stored Procedure For Train Schedule InBound**

**Stored Procedure to add Passenger Details**

CREATE PROCEDURE AddPassengerDetails @passengerIDint, @passengerNamevarchar(25), @DOB date, @contactInfoint, @addressIDint, @emailIDVARCHAR(225), @address VARCHAR(255), @state VARCHAR(225), @country VARCHAR(225), @zipcodeint

AS

BEGIN

INSERTINTOPassengerAddress(AddressID,[Address],[State],Country,ZipCode)VALUES (@addressID, @address ,@state, @country, @zipcode);

INSERTINTOPassenger(PassengerID,PassengerName,DateOfBirth,ContactInformation,AddressID,EmailID)VALUES (@passengerID, @passengerName, @DOB, @contactInfo,@addressID,@emailID);

SELECT\*FROM Passenger

END

EXECUTEAddPassengerDetails

@passengerID= 182

,@passengerName='Milan Yewale'

,@DOB ='1996-12-12'

,@contactInfo= 13847333

,@addressID= 111

,@emailID='ry@gmail.com'

,@address='69 South Huntington'

,@state='MA'

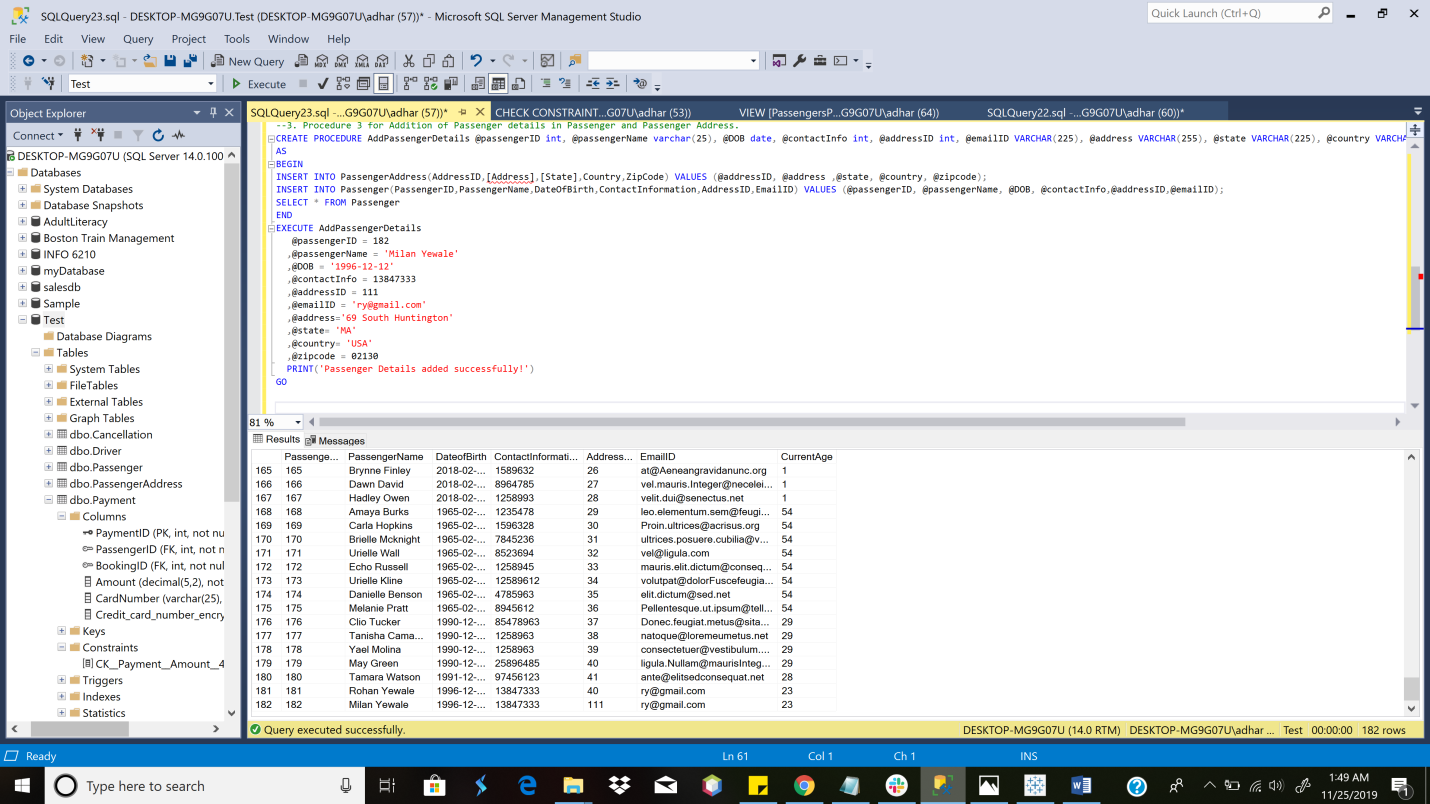
,@country='USA'

,@zipcode= 02130

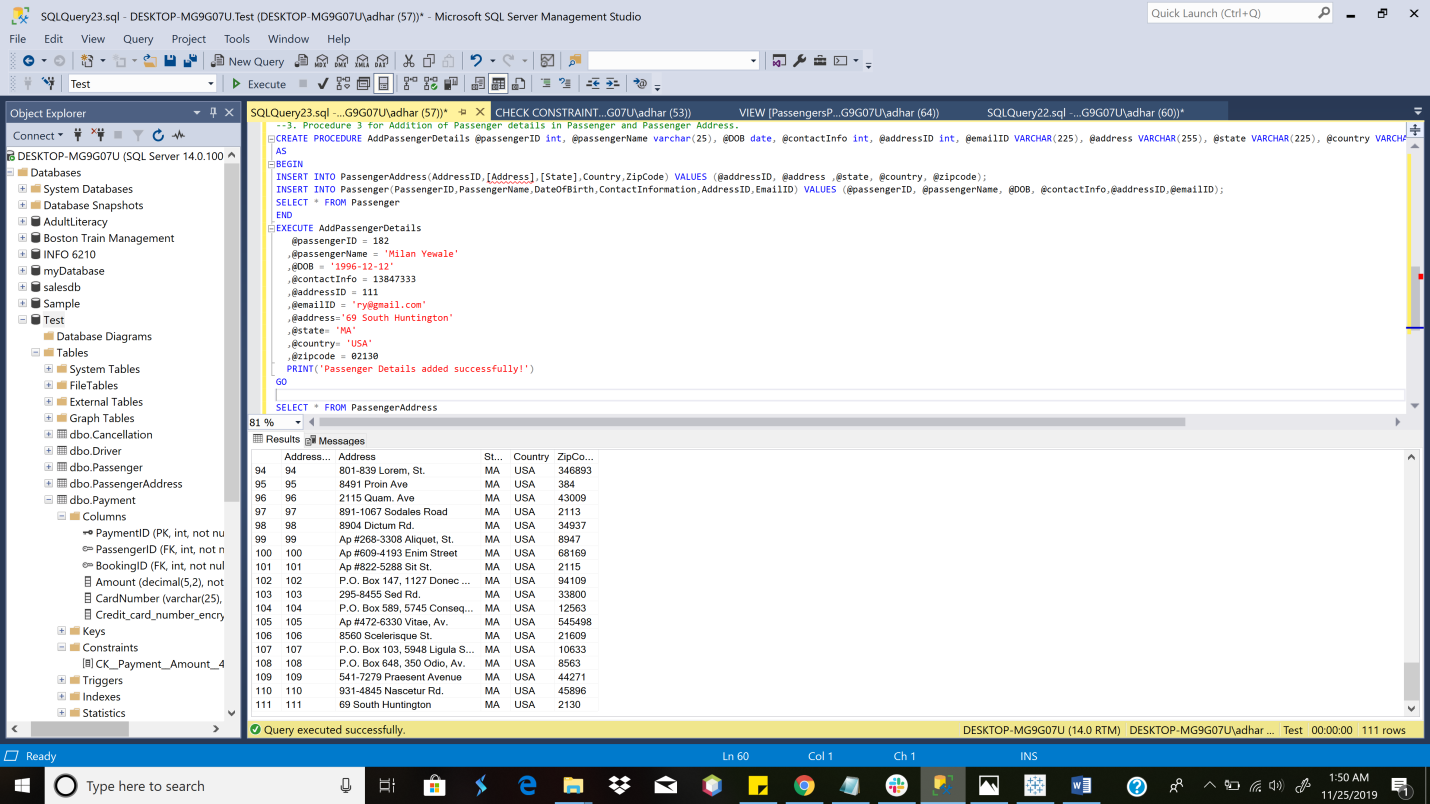
PRINT('Passenger Details added successfully!')

GO

SELECT\*FROMPassengerAddress



**Screenshot of added Passenger Details in Passenger Table**



**Screenshot of added Passenger Details in PassengerAddress Table using Same Stored Procedure**

**Stored Procedure to get Reaching Time at Desired Location**

CREATEPROCEDUREReachingTimeForDesiredDestination @SOURCE VARCHAR(50), @DESTINATION VARCHAR (50), @RunningTrainID VARCHAR (20)

AS

BEGIN

SELECTTrainID,RunningTrainID,s.StationName, Arrival,

(select Arrival from Station joinTrainScheduleonStation.StationStopID=TrainSchedule.StationStopIDwhere (StationName=@DESTINATION andRunningTrainID=@RunningTrainID))asReachingTimeAtDesiredDestination

FROMTrainSchedule t INNERJOIN Station s on

t.StationStopID=s.StationStopIDwhere (s.StationName= @SOURCE )and(RunningTrainID = @RunningTrainID)

END

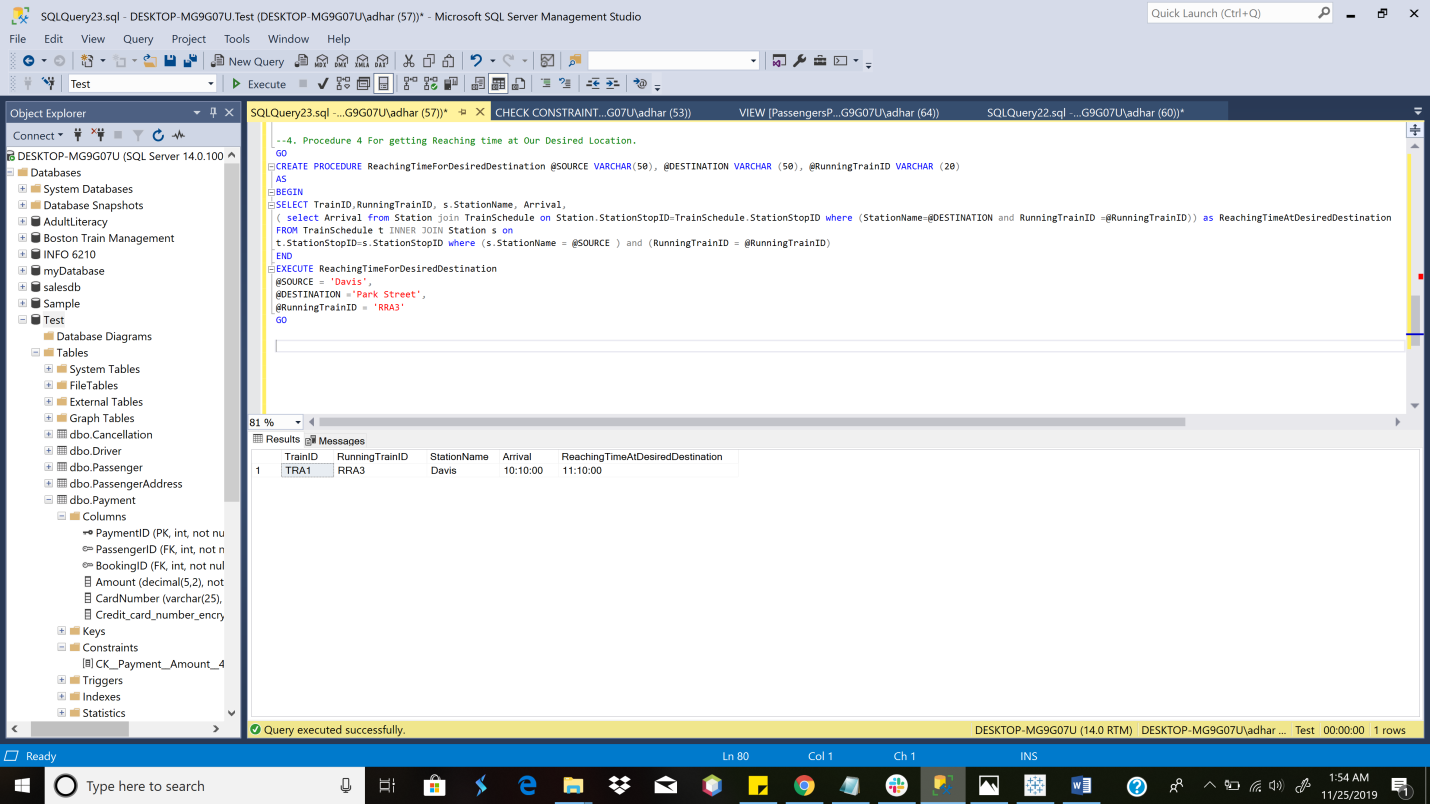
EXECUTEReachingTimeForDesiredDestination

@SOURCE ='Davis',

@DESTINATION ='Park Street',

@RunningTrainID ='RRA3'

GO



**Stored Procedure for getting Detailed Driver Schedule**

CREATE PROCEDURE DriverSchedule @DriverIDINT

AS

BEGIN

SELECTDriver.DriverID,Driver.DriverName,TrainSchedule.Arrival,TrainSchedule.TrainID,TrainSchedule.RouteID,TrainSchedule.[Date]

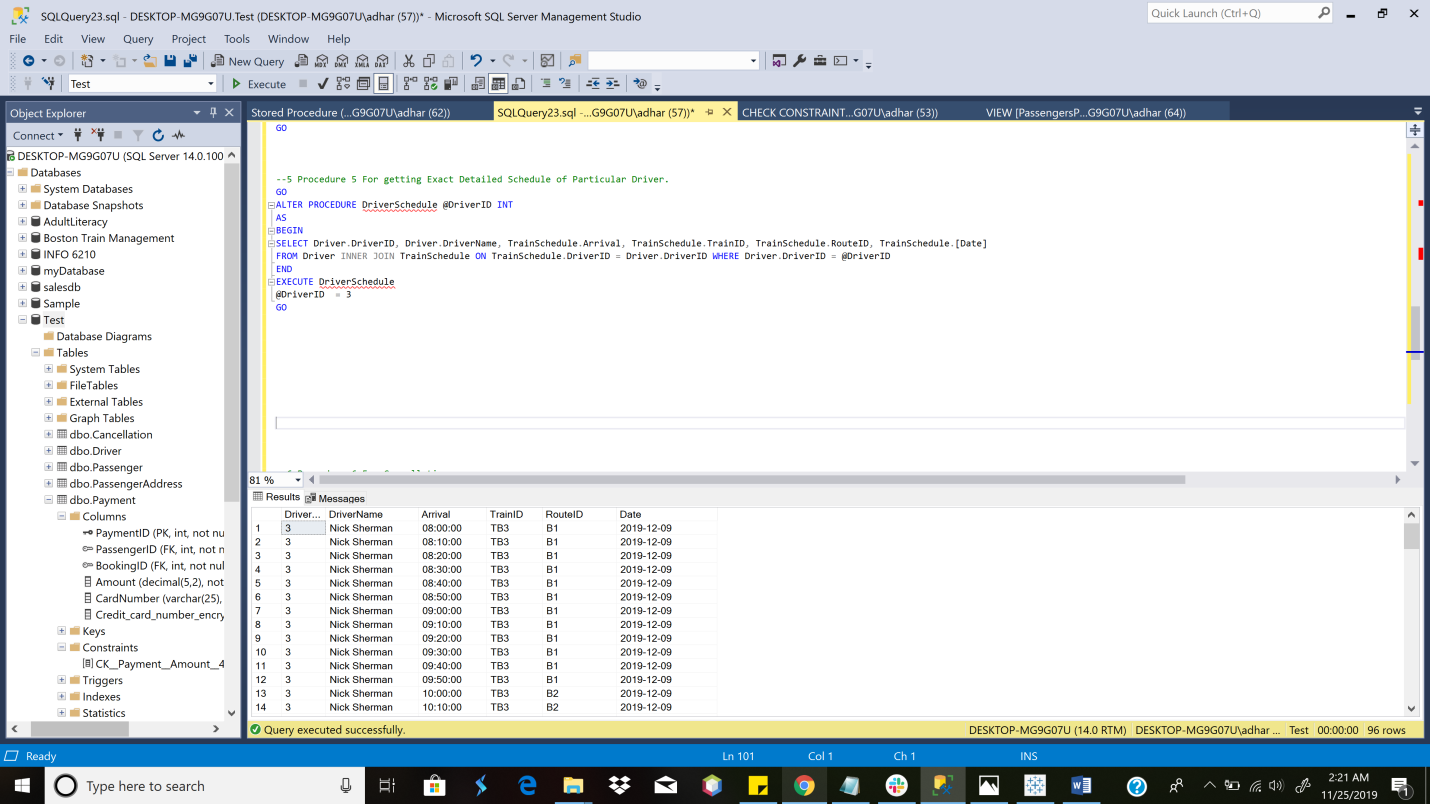
FROM Driver INNERJOINTrainScheduleONTrainSchedule.DriverID=Driver.DriverIDWHEREDriver.DriverID= @DriverID

END

EXECUTEDriverSchedule

@DriverID= 3

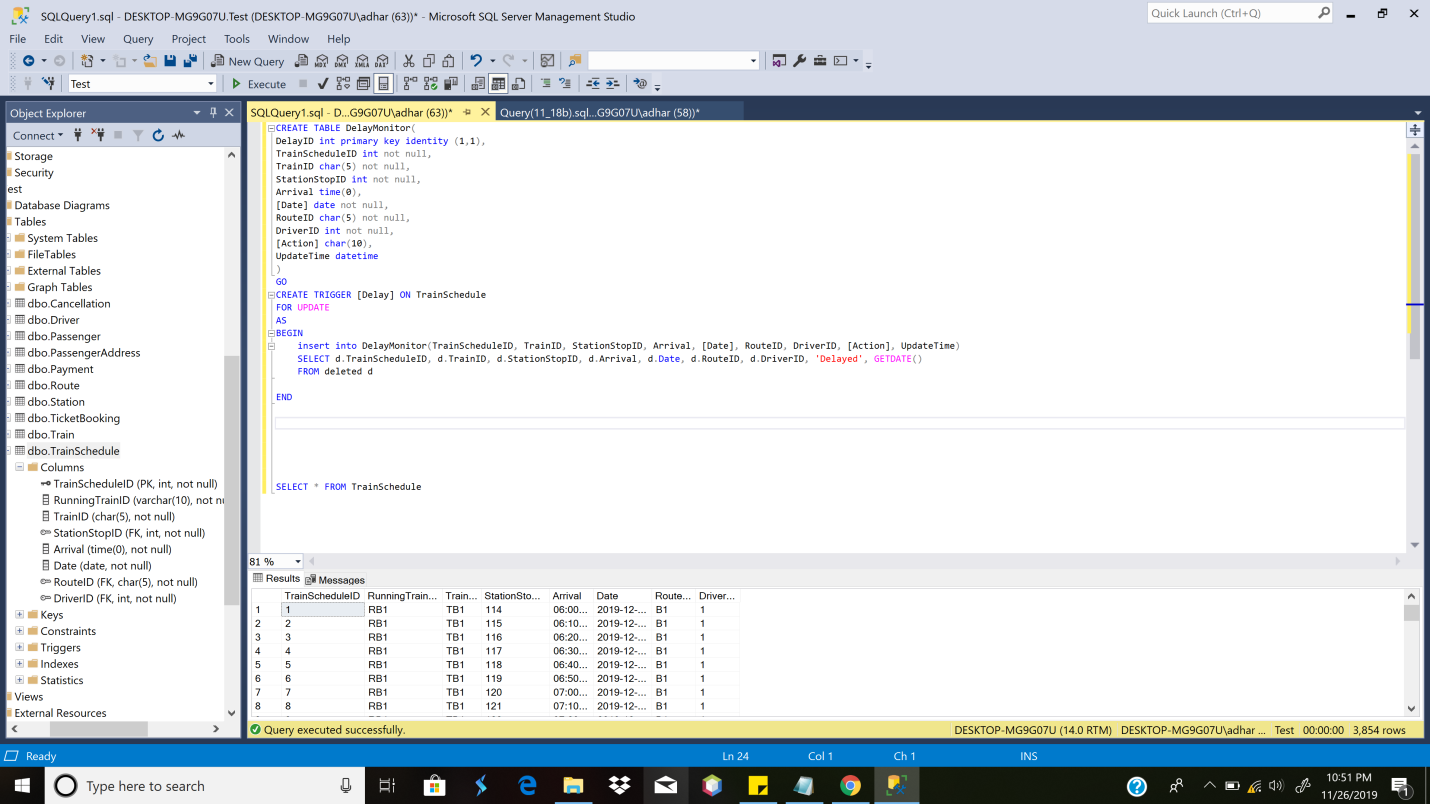
GO



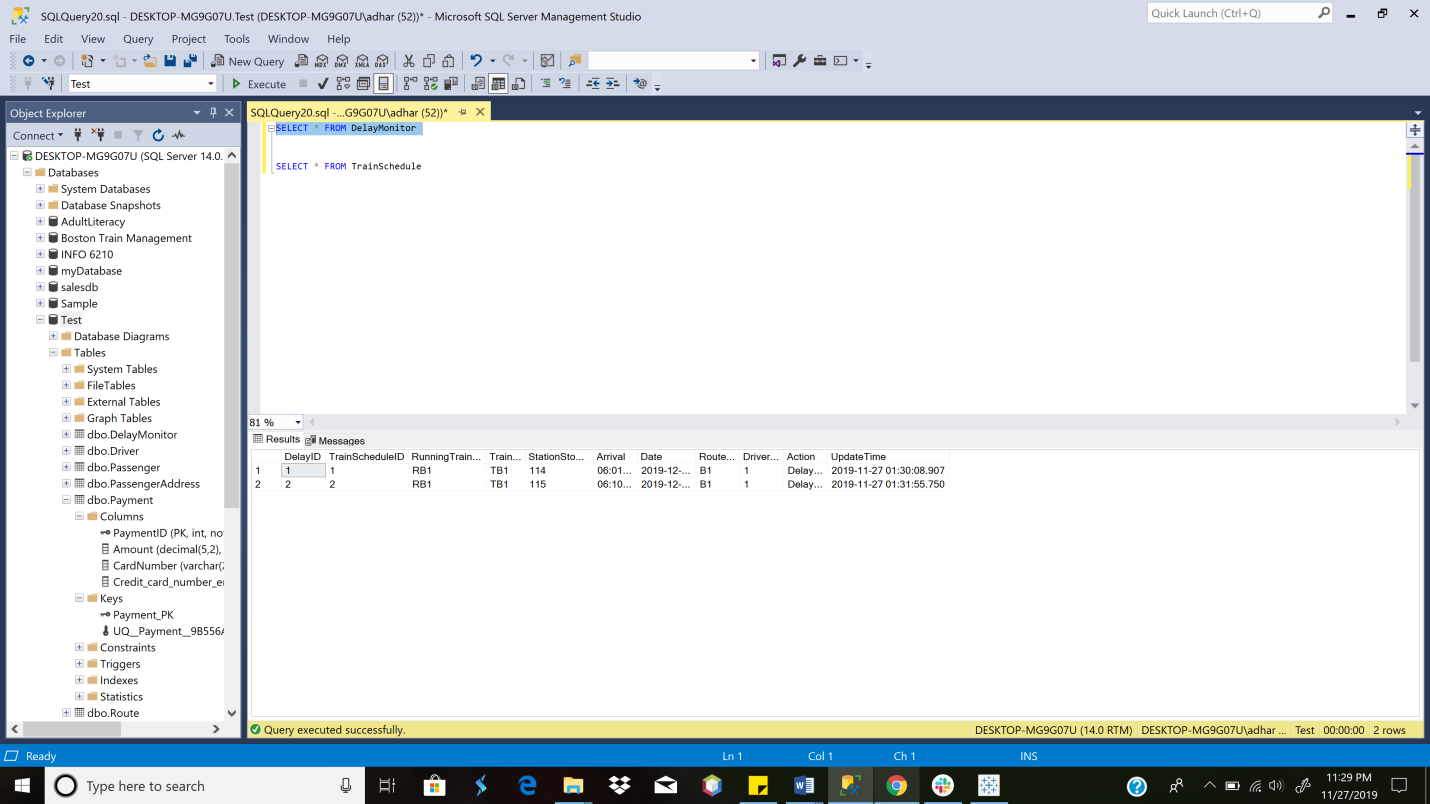
**Stored Procedure to obtain Detailed Driver Schedule**

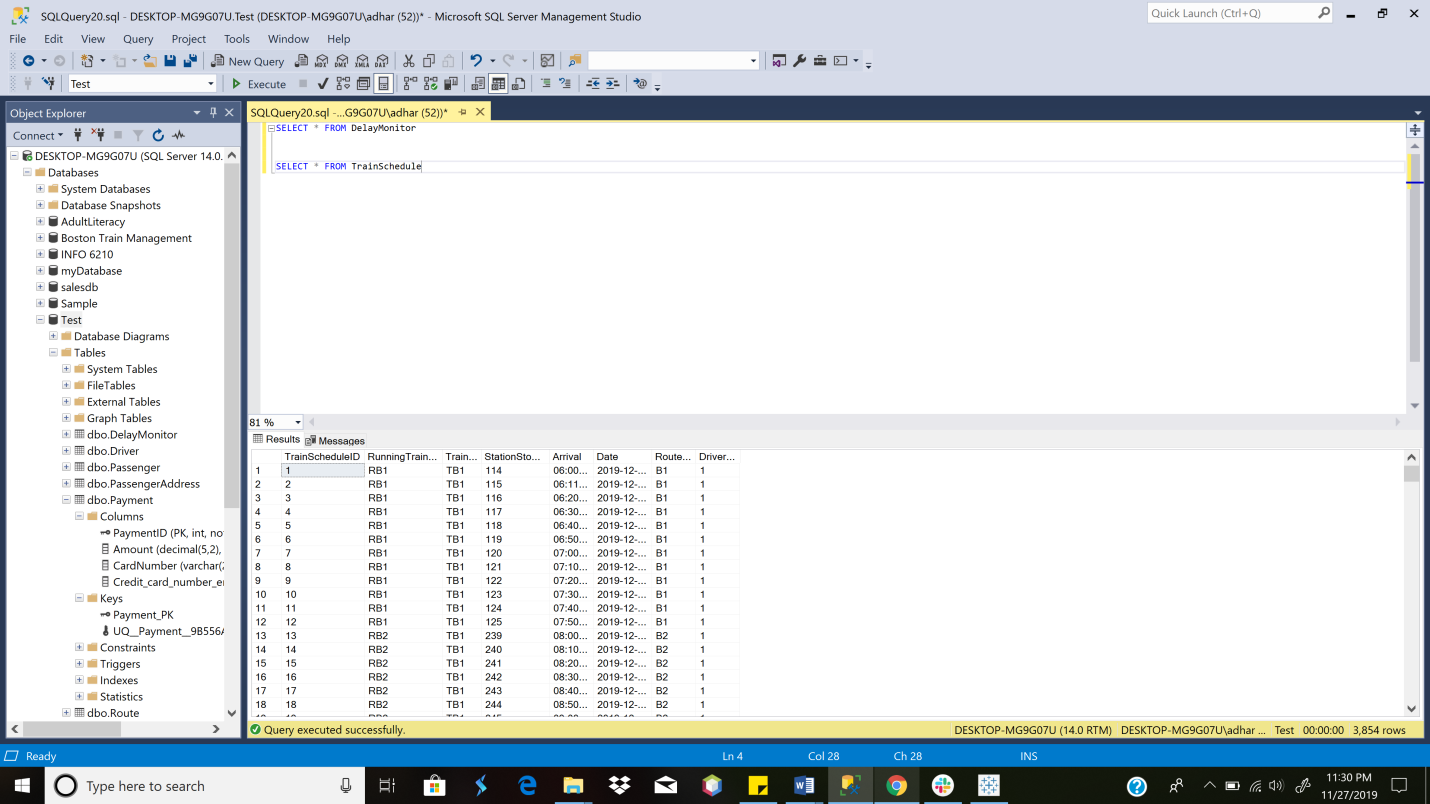
**TRIGGER [Delay Monitor]**

**INITIAL TRIGGER**



**TABLE DelayMonitor**





**Changes in TrainSchedule (after Trigger)**

CREATETABLEDelayMonitor(

DelayIDintprimarykeyidentity (1,1),

TrainScheduleIDintnotnull,

RunningTrainIDvarchar(10)notnull,

TrainIDchar(5)notnull,

StationStopIDintnotnull,

Arrival time(0),

[Date] datenotnull,

RouteIDchar(5)notnull,

DriverIDintnotnull,

[Action] char(10),

UpdateTimedatetime

)

GO

ALTERTRIGGER [Delay] ONTrainSchedule

AFTERUPDATE

AS

BEGIN

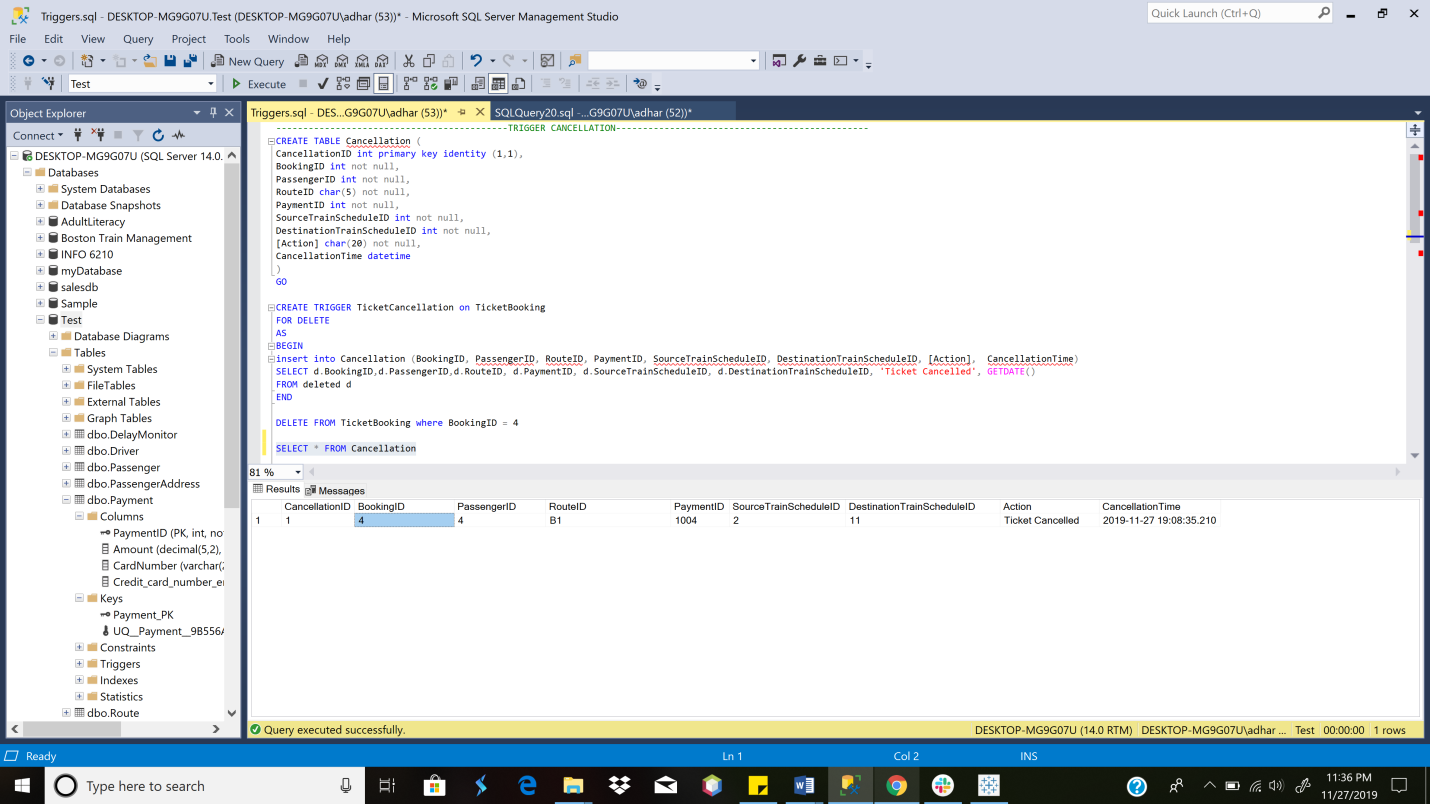
insertintoDelayMonitor(TrainScheduleID,RunningTrainID,TrainID,StationStopID, Arrival, [Date],RouteID,DriverID, [Action],UpdateTime)

SELECTd.TrainScheduleID,d.RunningTrainID,d.TrainID,d.StationStopID,d.Arrival,d.Date,d.RouteID,d.DriverID,'Delayed',GETDATE()

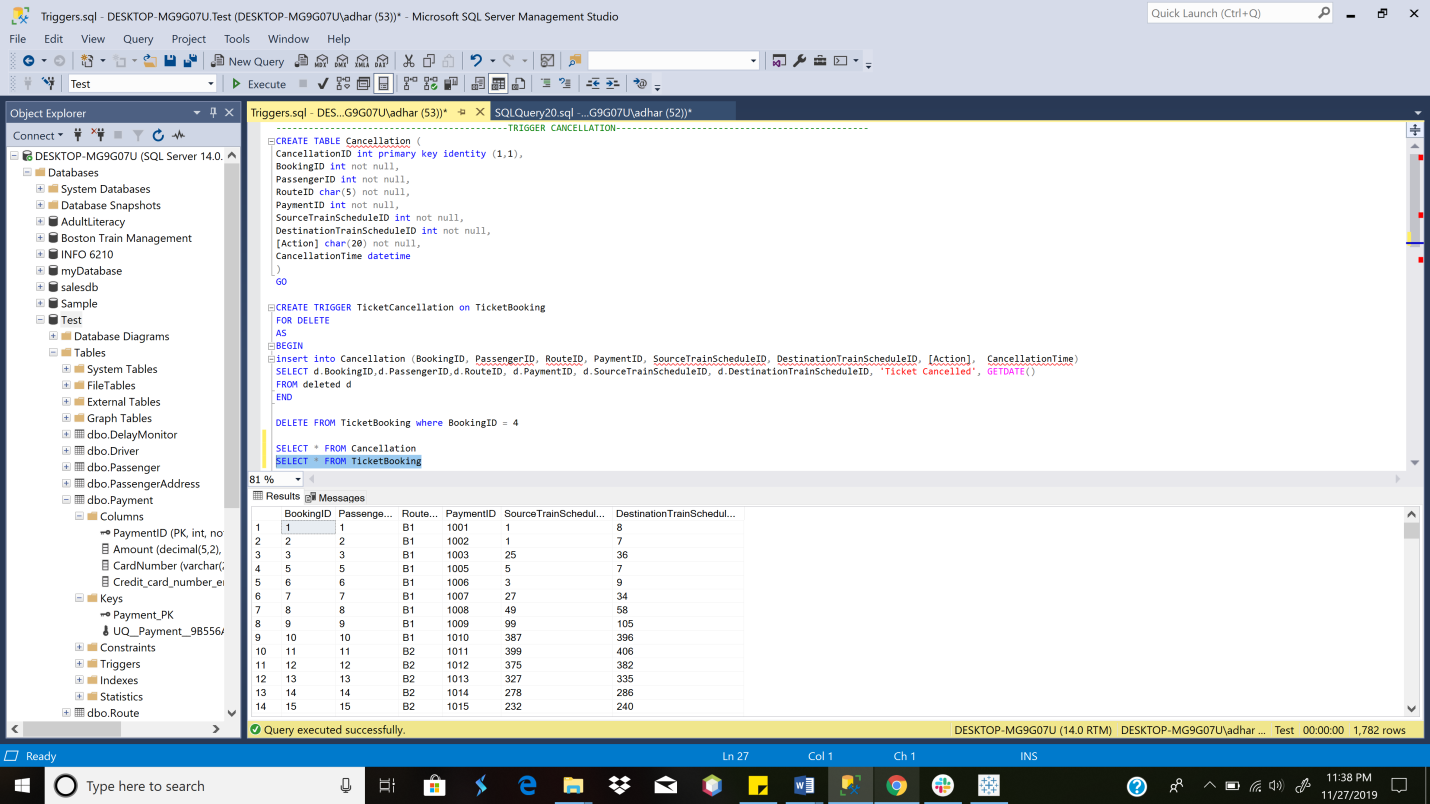
FROM deleted d

END

**TRIGGER [Cancellation]**



**Cancellation Table after TicketCancellation Trigger is fired**



**TicketBooking Table after TicketCancellation Trigger is fired**

CREATETABLE Cancellation(

CancellationIDintprimarykeyidentity (1,1),

BookingIDintnotnull,

PassengerIDintnotnull,

RouteIDchar(5)notnull,

PaymentIDintnotnull,

SourceTrainScheduleIDintnotnull,

DestinationTrainScheduleIDintnotnull,

[Action] char(20)notnull,

CancellationTimedatetime

)

GO

CREATETRIGGERTicketCancellationonTicketBooking

FORDELETE

AS

BEGIN

insertinto Cancellation(BookingID,PassengerID,RouteID,PaymentID,SourceTrainScheduleID,DestinationTrainScheduleID, [Action],CancellationTime)

SELECTd.BookingID,d.PassengerID,d.RouteID,d.PaymentID,d.SourceTrainScheduleID,d.DestinationTrainScheduleID,'Ticket Cancelled',GETDATE()

FROM deleted d

END

DELETEFROMTicketBookingwhereBookingID= 4

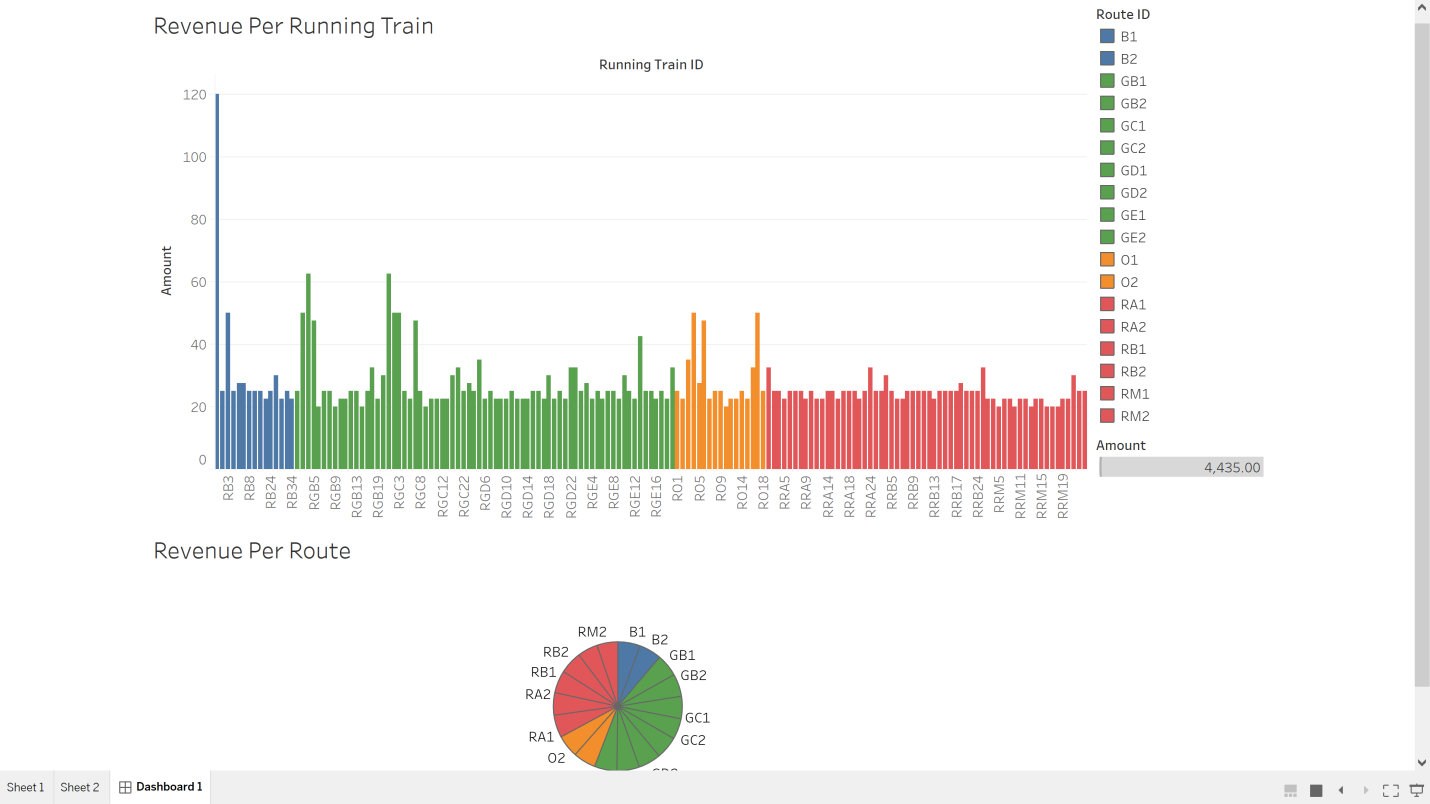
SELECT\*FROM Cancellation

SELECT\*FROMTicketBooking

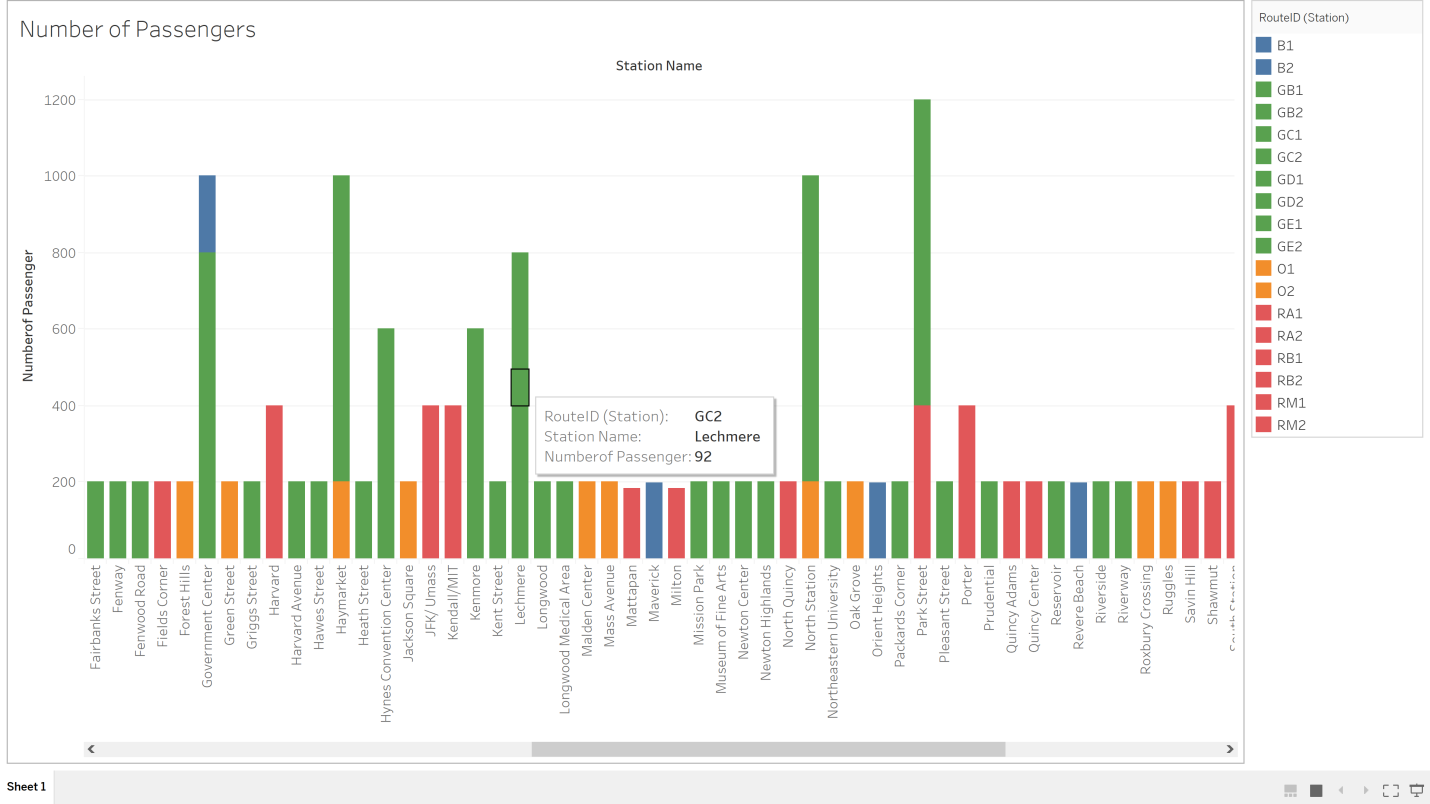
**DASHBOARDS**

-Bar Graph for **Revenue per Train Route** which is further divided into **Individual Running Trains**.

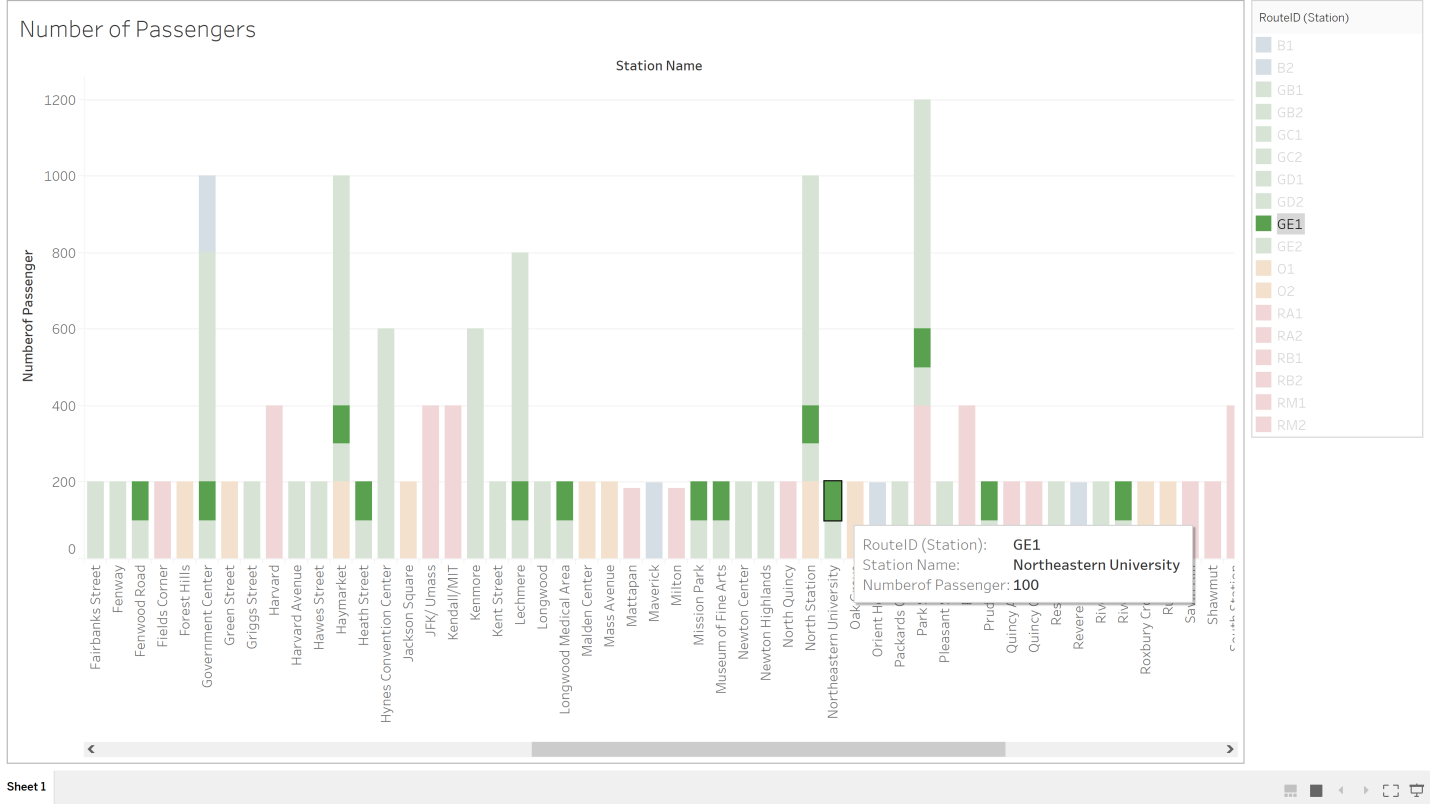
- Pie Chart for **Revenue per Train Route**



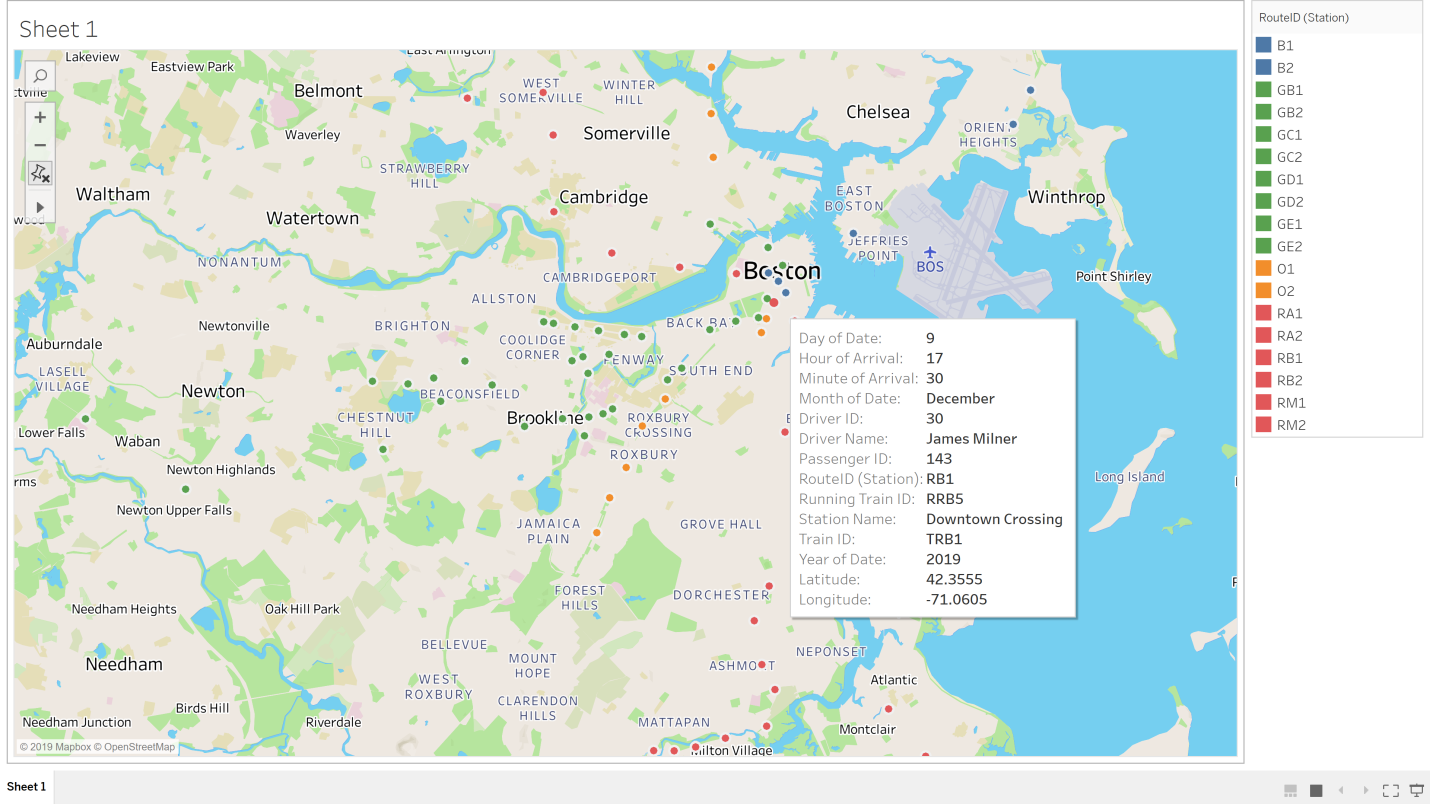
-Bar Graph for **Number of Passengers boarding** at a particular **Station (Over Multiple Lines)**



-Bar Graph for **Number of Passengers**at a **particular station(Northeastern)**on a **particular route(GE1)**



**-Data Points on a map** indicating details about :**Onboarding Passenger on a particulartrain, route, station, date and time**



**-Data points on a map** indicating details about: **Onboarding passenger** on a **particular Route, Station, Date**

**-Size of the datapoint** represents the **number of onboarding passengers** at a **particular Station**

