CSE 581 SPRING 2020 Intro to Database Management Systems

Project – 2

Design and Implementation of Recruitment Database

Abstract:

In this project we design and implement Recruitment Database for 'xyz' company. I've written all the tables which are required during the design and then I've written its associated columns. I've designed the database using ER diagrams and then we normalize it so that to improve efficiency and reducing redundancy. After designing the database, we create tables and constraints. I've added triggers according to the problem statement. I've populated data and inserted into the tables. I've used database concepts like Stored Procedures, Server Roles, Security, Views, Functions. After Implementing the database, we run few test scripts which shows that database is correct and useful. Recruitment is important part of the company and with this project we can maintain the recruitment hiring of any company.

VENKATA SAI PAWAN KOMARAVOLU

SUID : 425210282

Table of Contents

1. DATABASE NAME

I've named my database as XYZRecruitment Database

2. DESIGN PHASE OF PROJECT

2.1INTRODUCTION

In today's world, almost most of the companies have HR department which develops and administers programs which increase an organization effectiveness. The HR department focuses on Recruitment, Compensation and benefits, Training and Learning, Labor and Employee Relations and Organization Development. In this project we design, implement and test Recruitment branch of a company 'xyz' . The recruitment Database keeps track of steps beginning from the job opening to Onboarding process of the selected candidates. In between there are multiple steps where we need to keep track of. Our first step is to design a database for that we have taken tables and their columns, and we use ER diagrams to design our database.

We design the database in way that we can retrieve records of all the candidates who have applied and who's interview process in on going or those which are completed. We keep track of candidate status in the who process of the Interview.

The Interview is either online or onsite. Depending on the location of the interview we maintain tables for Hotel reservation, Airline reservation and Car rentals and we maintain amount for each candidate so that it can be reimbursed by the candidate. We also keep track of the candidate applying to specific job role and job type.

We maintain an onboarding table where a selected candidate training is done and all his/her documents are verified and with this our database management is done as a database admin.

2.2TABLES USED

The following are the tables used in my project. I've added appropriate columns to each table so that we can meet all the conditions which were asked in the problem statement.

1) Address Table :
   - AddressID
   - AddressLine1
   - AddressLine2
   - City
   - State
   - ZipCode

2) ContactDetails Table
   - ContactID
   - EmailAddress
   - PhoneNumber

3) Department Table
   - DepartmentID
   - DepartmentName
   - DepartmentDescription

4) CandidateQualification Table
   - CandidateQualificationID
   - GradGpa
   - WorkExperienceYears

5) Interviewer Table
   - InterviewerID
   - InterviewerName
   - InterviewerStatus

6) For InterviewLocation Table
   - BuildingName
   - InterviewLocationID
   - City
   - State
   - ZipCode

7) JobOpening Table
   - JobOpeningID
   - JobRole

- JobDescription
- DepartmentID
- JobType
- NoOfJobOpenings

8) CandidateDetails Table
- CandidateDetailsID
- CandidateName
- AddressID
- ContactID
- CandidateQualificationID
- CandidateQualification(CandidateQualificationID)

9) JobApplication Table
- CandidateID
- JobOpeningID
- CandidateDetailsID
- CandidateStatus

10) Interview Table
- InterviewerID
- InterviewID
- CandidateID
- InterviewResult
- InterviewType

11) ComplaintHandling Table
- CandidateID
- ComplaintID
- InterviewerID
- ComplaintDescription

12) AirlineReservation Table
- TicketID
- InterviewID
- CandidateID
- InterviewLocationID
- FlightDetails
- AirlineFare money

13) Onboarding Table
- OnboardingID
- CandidateID

- JobOpeningID
- DepartmentID
- CandidateDocuments

14) HotelReservation Table

- ReservationID
- InterviewID
- CandidateID
- InterviewLocationID
- HotelDetails
- HotelFare money

15) CarRental Table

- RentalID
- InterviewID
- CandidateID
- InterviewLocationID
- CarDetails
- RentalFare money

16) For Reimbursement Table

- ReimbursementID
- InterviewID
- CandidateID
- RentalID
- TicketID
- ReservationID
- ReimbursementAmount

17) JobOfferNegotitation Table

- NegotiationID
- CandidateID
- NegotitationStatus

2.3 LOGIC

First Starting from JobApplication table with CandidateID as the primary key. The Candidate who will apply he needs to fill his details and this will be stored in CandidateDetails table. In JobApplication table we will have CandidateStatus column where the status of the interview is stored and updated when ever there is change in the interview status for that particular Candidate

In CanddidateDetails table we have CanddidateDetailsID as the primary key which stores the CanddiateDetails

A candidate apply to a particular Jobrole and Jobtype for that I have created JobOpening table with JobOpeningID as the primary key. In this table we keep track of JobRole , DepartmentID and number of JobOpening. When a candidate gets selected and the opening aren't there then his status is done as On call ofr next Interview.

After Applying the Interview can be Online and Onsite and we update these in Interview table with InterviewID as primary key.

Depending on the InterviewType and InterviewLocation if its onsite then company pays for the Candidate. Company allocates AirlineFare , HotelFare and CarRental Fare which are three tables in the database with TicketID , ReservationID and RentalID as the primary key which keeps track of travel expenses , Living expenses of the candidate. We have InterviewLocationID as the primary key for the InterviewLocation.

These fare are later reimbursed and thus for this we have reimbursement table with ReiumbursementID as the primary key.

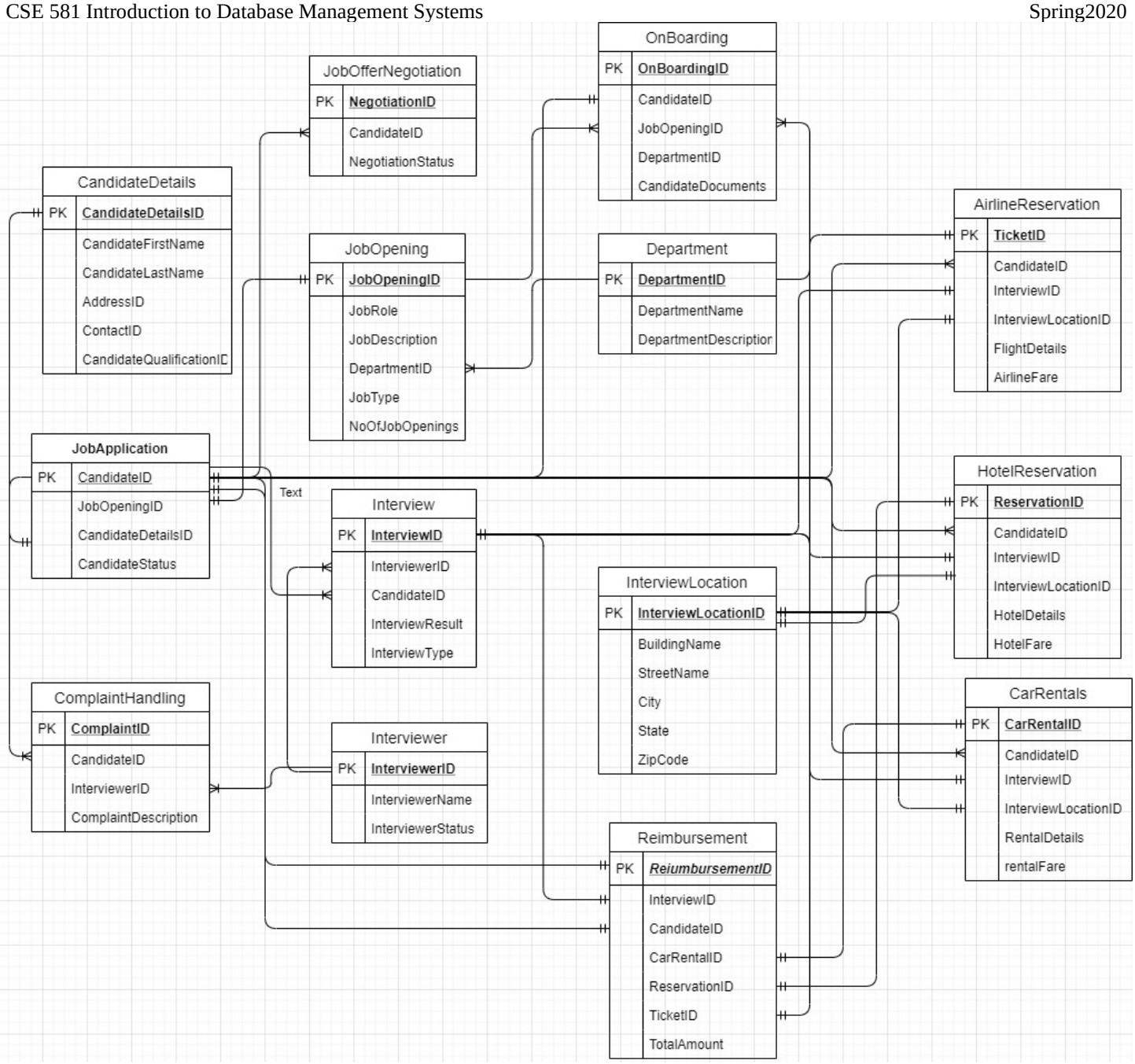
If a candidate gets selected then offer is made to him but the candidate can negotiate with the company and therefore we create OfferNegotiation table with OfferNegotiationID as the primary key.

After accepting the offer the candidate has to do Onboarding process for this we have a table with OnboardingID as the primary key. After the Onboarding process is completed the status of the Candidate is updated and with th

2.4 E/R DIAGRAM

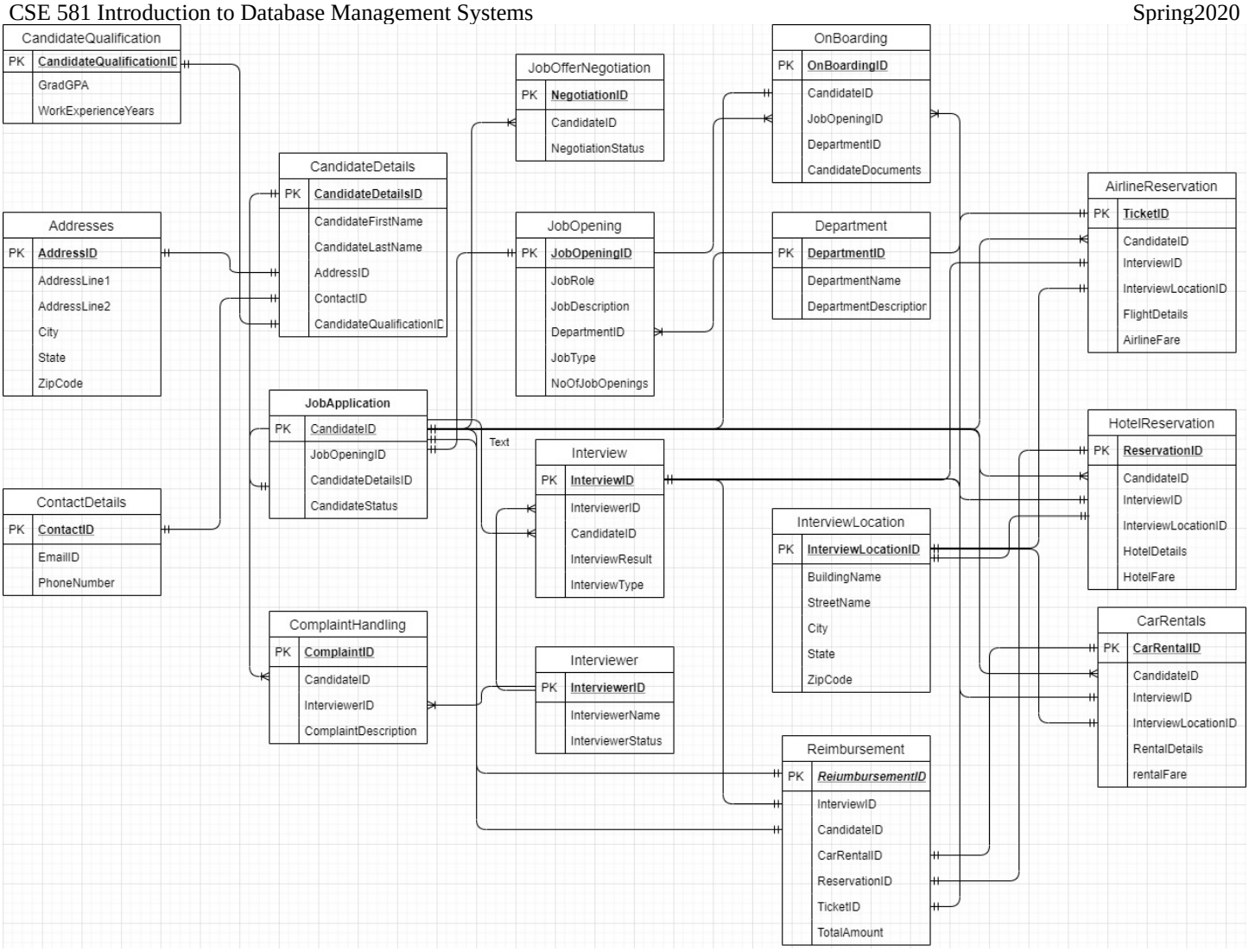
The Entity Relationship(E/R) diagram mainly focusses on how entities in the database are related to each other and how to link them.

Here is the E/R diagram given below that is used for my project.



NORMALIZED DATABASE DESIGN

For a database design to be in 3rd Normal form, it must be in First Normal Form i.e. it must not contain any non repeating columns and should contain scalar values in all rows, and it must also be in 2nd Normal form(i.e. no partial dependencies) and also all the columns must only depend on the primary key(i.e. no transitive dependencies). So here is my database design in 3rd Normal form given below:



2.5 Potential Integrity and Security issues

The potential security and integrity is reduced by transforming the database into the 3NF format.

For referential integrity to hold in a relational database, any column in a base table that is declared a foreign key can only contain either null values or values from a parent table's primary key or a candidate key. In other words, when a foreign key value is used it must reference a valid, existing primary key in the parent table. For instance, deleting a record that contains a value referred to by a foreign key in another table would break referential integrity.

3. IMPLEMENTATION PHASE OF THE PROJECT

We implement the database of this project by firstly creating database, creating tables and columns which we have discussed in the design phase and adding the trigger to solve the problem statement.

3.1 DATABASE CREATION

We create Database and name it 'XYZRecruitmentDatabase'.

3.2 Constraints

There are two types of constraints will be there. They are:

1. Normal Constraints:- Normal constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted. Various types of normal constraints are: NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECH, DEFAULT, UNIQUE, INDEX.
2. Business Constraints: They include the relationships between the tables such as One-to-One, Many-to-One, Many-to-One.

Following are few Constraints like primary key , foreign key , Nullable and relationships which are used in this project.

1) For Address Table

AddressID INT NOT NULL PRIMARY KEY

AddressLine1 VARCHAR(50) NOT NULL

AddressLine2 VARCHAR(50) Default NULL

City CHAR(50) NOT NULL

State CHAR(50) NOT NULL

ZipCode INT NOT NULL

2) For ContactDetails Table

ContactID INT NOT NULL PRIMARY KEY

EmailAddress VARCHAR(50) NOT NULL

PhoneNumber VARCHAR(12) NOT NULL

3) For Department Table

DepartmentID INT NOT NULL PRIMARY KEY

DepartmentName CHAR(30) NOT NULL

DepartmentDescription varchar(50) NULL

4) For CandidateQualification Table

CandidateQualificationID INT NOT NULL PRIMARY KEY

GradGpa INT NOT NULL

WorkExperienceYears INT NOT NULL

5) For Interviewer Table

InterviewerID INT NOT NULL PRIMARY KEY

InterviewerName CHAR(50) NOT NULL

InterviewerStatus Varchar(50) NULL

6) For InterviewLocation Table

InterviewLocationID INT NOT NULL PRIMARY KEY

BuildingName VARCHAR(50) NOT NULL

City CHAR(50) NOT NULL

State CHAR(50) NOT NULL

ZipCode INT NOT NULL

7) For JobOpening Table

JobOpeningID INT NOT NULL PRIMARY KEY

JobRole VARCHAR(50) NOT NULL

JobDescription VARCHAR(50) NOT NULL

DepartmentID INT NOT NULL REFERENCES Department(DepartmentID)

JobType VARCHAR(50) NOT NULL

NoOfJobOpenings INT NOT NULL

8) For CandidateDetails Table

CandidateDetailsID INT NOT NULL PRIMARY KEY

CandidateName CHAR(50)

AddressID INT NOT NULL REFERENCES Addresses(AddressID)

ContactID INT NOT NULL REFERENCES ContactDetails(ContactID)

CandidateQualificationID INT NOT NULL REFERENCES CandidateQualification(CandidateQualificationID)

9) JobApplication Table

CandidateID INT NOT NULL PRIMARY KEY

JobOpeningID INT NOT NULL REFERENCES JobOpening(JobOpeningID

CandidateDetailsID INT NOT NULL REFERENCES CandidateDetails(CandidateDetailsID)

CandidateStatus VARCHAR(50) NOT NULL

10) For Interview Table

InterviewID INT NOT NULL PRIMARY KEY

InterviewerID INT NOT NULL References Interviewer(InterviewerID

CandidateID INT NOT NULL References JobApplication(CandidateID)

InterviewResult CHAR(50) NOT NULL

InterviewType VARCHAR(20) NOT NULL

11) For ComplaintHandling Table

ComplaintID INT NOT NULL PRIMARY KEY

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

InterviewerID INT NOT NULL References Interviewer(InterviewerID)

ComplaintDescription VARCHAR(100)

12) For AirlineReservation Table

TicketID INT NOT NULL PRIMARY KEY

InterviewID INT NOT NULL REFERENCES Interview(InterviewID)

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID),

FlightDetails VARCHAR(100)

AirlineFare money

13) Onboarding Table

OnboardingID INT NOT NULL PRIMARY KEY

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

JobOpeningID INT NOT NULL REFERENCES JobOpening(JobOpeningID)

DepartmentID INT NOT NULL REFERENCES Department(DepartmentID)

CandidateDocuments Varchar(50) NOT NULL

14) HotelReservation Table

ReservationID INT NOT NULL PRIMARY KEY

InterviewID INT NOT NULL REFERENCES Interview(InterviewID)

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID)

HotelDetails VARCHAR(100) NOT NULL

HotelFare money NOT NULL

15) For CarRental Table

RentalID INT NOT NULL PRIMARY KEY

InterviewID INT NOT NULL REFERENCES Interview(InterviewID)

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID)

CarDetails VARCHAR(50) NOT NUL

RentalFare money NOT NULL

16) For Reimbursement Table

ReimbursementID INT NOT NULL PRIMARY KEY

InterviewID INT NOT NULL REFERENCES Interview(InterviewID)

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

RentalID INT NOT NULL REFERENCES CarRental(RentalID)

TicketID INT NOT NULL REFERENCES AirlineReservation(TicketID)

ReservationID INT NOT NULL REFERENCES HotelReservation(ReservationID)

ReimbursementAmount MONEY

17) For JobOfferNegotitation Table

NegotiationID INT NOT NULL PRIMARY KEY

CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID)

NegotitationStatus varchar(50) NOT NULL

Here are the list of relations that are used between the tables in my project.

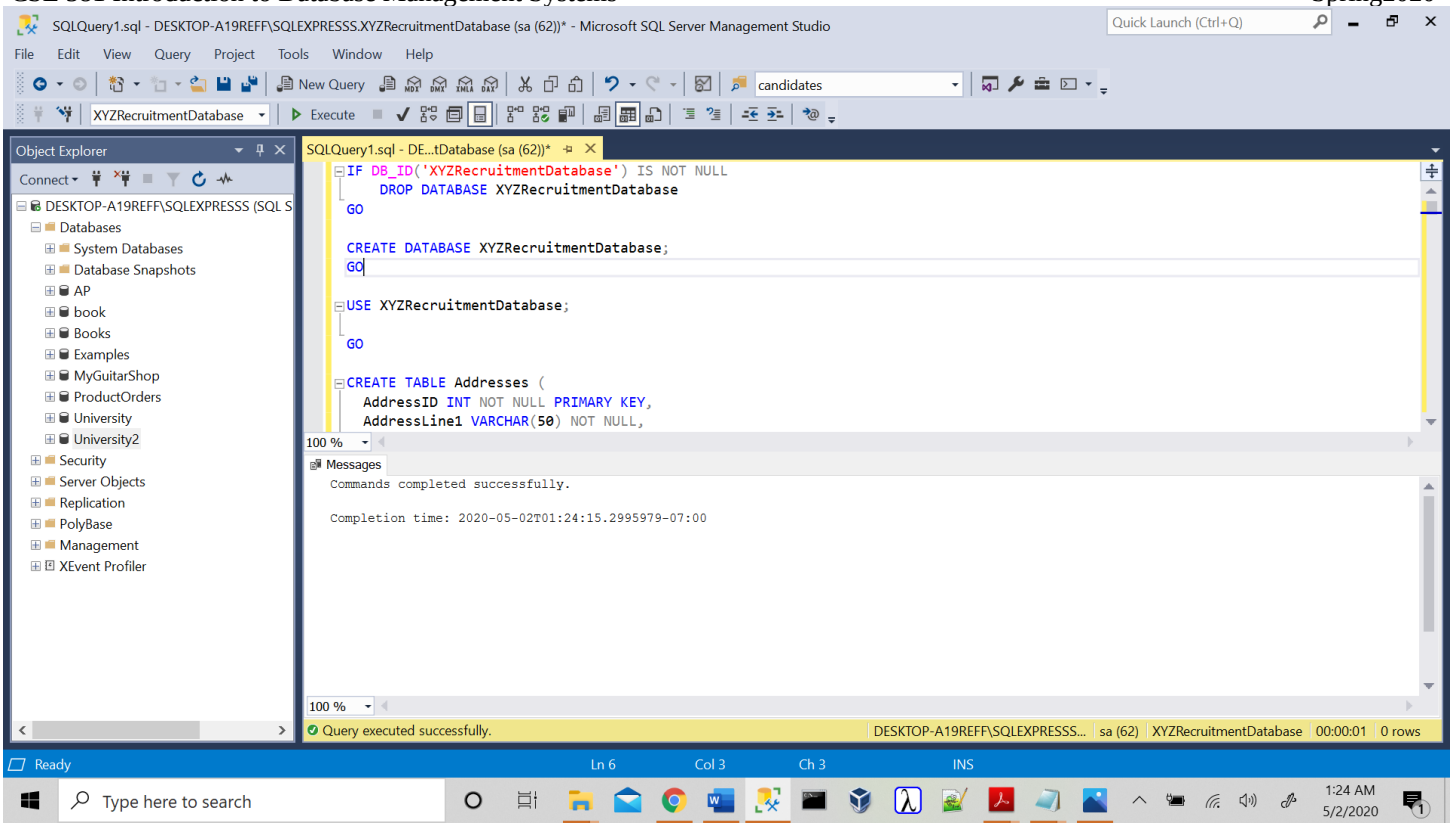
In this project I have used 17 One-to-One relations, 11 Many-to-One relations

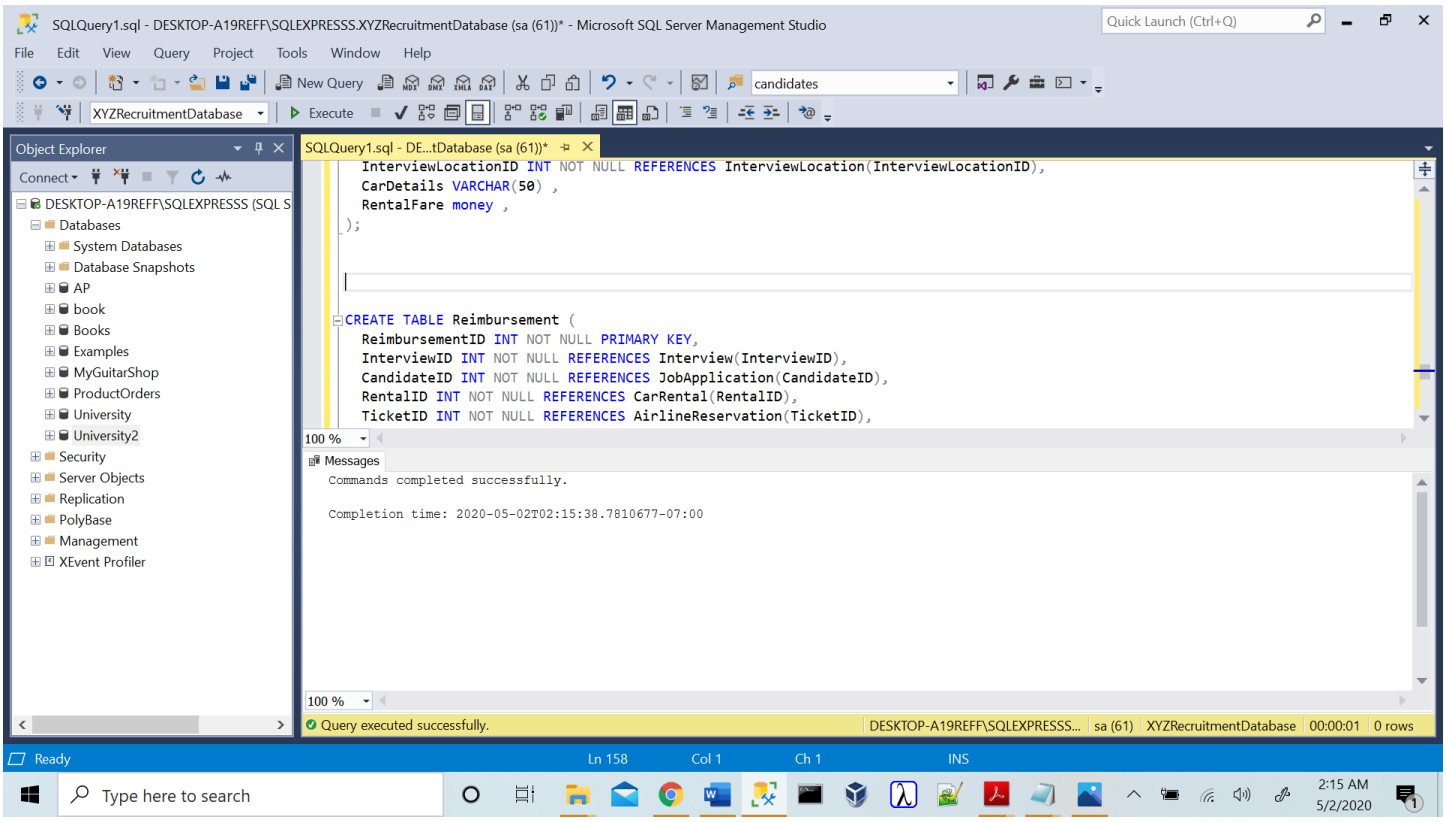
- There is One to One relation between JobApplication and CandidateDetails on CandidateDetailsID
- There is One to Many relation between JobApplication and ComplaintHandling on CandidateID
- There is One to One relation between CandidateDetails and CandidateQualification on CandidateQualificationID
- There is One to One relation between CandidateDetails and Addresses on AddressID
- There is One to One relation between CandidateDetails and ContactDetails on ContactID
- There is One to Many relation between JobApplication and JobOfferNegotiation on CandidateID

- There is One to One relation between JobApplication and JobOpening on JobOpeningID
- There is One to Many relation between Interviewer and Interview on InterviewerID
- There is One to Many relation between Interviewer and ComplaintHandling on InterviewerID
- There is One to Many relation between JobApplication and Interview on CandidateID
- There is One to One relation between JobApplication and OnBoarding on CandidateID
- There is One to Many relation between JobOpening and OnBoarding on JobOpeningID
- There is One to Many relation between Department and OnBoarding on JobOpeningID
- There is One to Many relation between Department and JobOpening on DepartmentID
- There is One to One relation between InterviewLocation and AirlineReservation on InterviewLocationID
- There is One to One relation between InterviewLocation and HotelReservation on InterviewLocationID
- There is One to One relation between InterviewLocation and CarRentals on InterviewLocationID
- There is One to One relation between Reimbursement and AirlineReservation on TicketID
- There is One to One relation between Reimbursement and HotelReservation on ReservationID
- There is One to One relation between Reimbursement and CarRentals on CarRentalID
- There is One to One relation between Reimbursement and JobApplication on CandidateID
- There is One to One relation between Reimbursement and Interview on InterviewID
- There is One to Many relation between JobApplication and AirlineReservation on CandidateID
- There is One to One relation between Interview and AirlineReservation on InterviewID
- There is One to Many relation between JobApplication and HotelReservation on CandidateID
- There is One to One relation between Interview and HotelReservation on InterviewID
- There is One to Many relation between JobApplication and CarRentals on CandidateID
- There is One to One relation between Interview and CarRentals on InterviewID

3.3 TABLE CREATING

WE CREATE DATABASE AND THEN WE CREATE TABLES WITH ABOVE CONSTRAINTS


SQLQuery1.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (62))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery1.sql - DE...tDatabase (sa (62))*
IF DB_ID('XYZRecruitmentDatabase') IS NOT NULL
DROP DATABASE XYZRecruitmentDatabase
GO
CREATE DATABASE XYZRecruitmentDatabase;
GO
USE XYZRecruitmentDatabase;
GO
CREATE TABLE Addresses (
AddressID INT NOT NULL PRIMARY KEY,
AddressLine1 VARCHAR(50) NOT NULL,
100 %
Messages
Commands completed successfully.
Completion time: 2020-05-02T01:24:15.2995979-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (62) XYZRecruitmentDatabase 00:00:01 0 rows
Ready
Ln 6 Col 3 Ch 3 INS
Type here to search
1:24 AM
5/2/2020

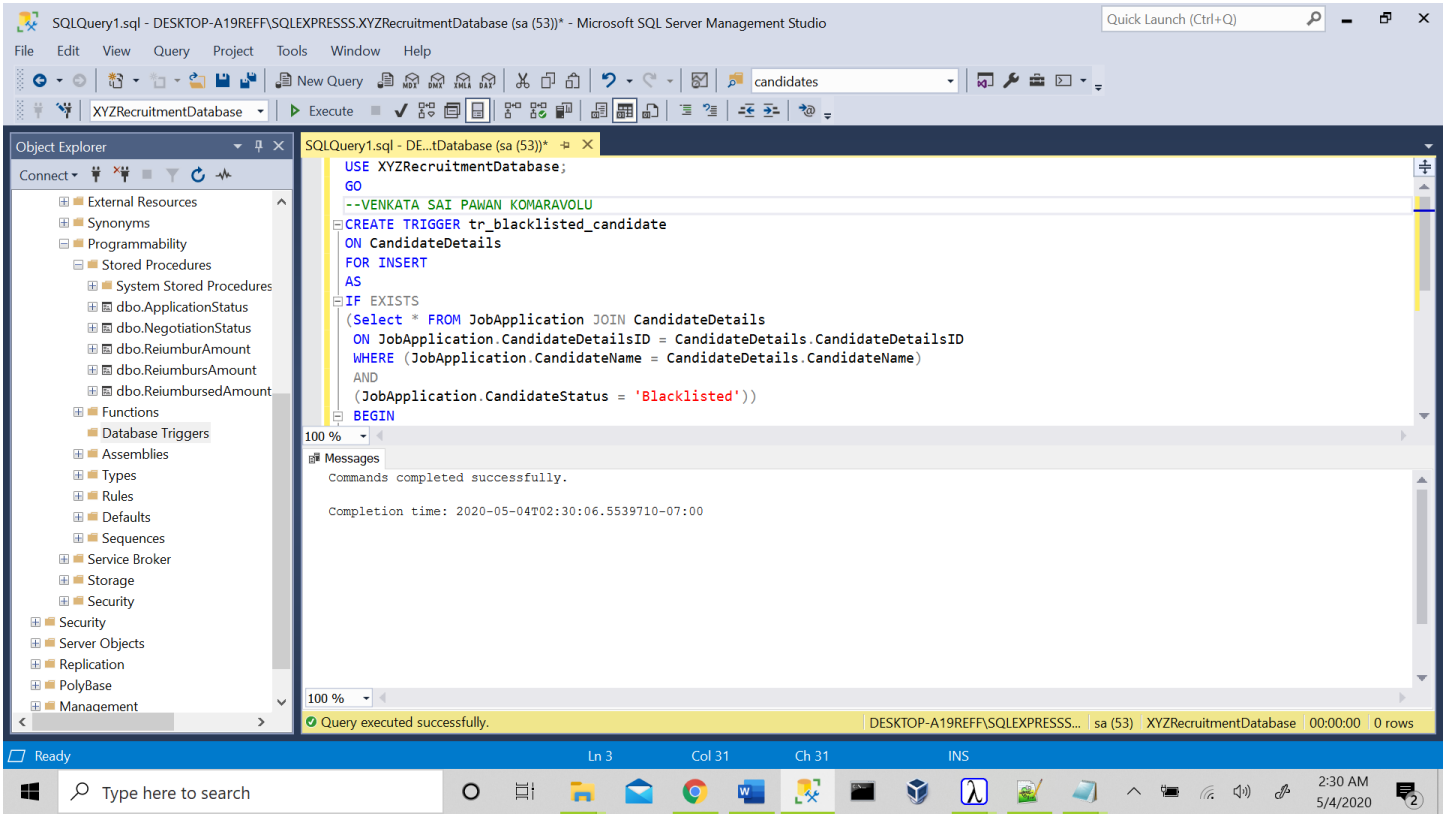

SQLQuery1.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (61))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery1.sql - DE...tDatabase (sa (61))*
InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID),
CarDetails VARCHAR(50) ,
RentalFare money ,
);
CREATE TABLE Reimbursement (
ReimbursementID INT NOT NULL PRIMARY KEY,
InterviewID INT NOT NULL REFERENCES Interview(InterviewID),
CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
RentalID INT NOT NULL REFERENCES CarRental(RentalID),
TicketID INT NOT NULL REFERENCES AirlineReservation(TicketID),
100 %
Messages
Commands completed successfully.
Completion time: 2020-05-02T02:15:38.7810677-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (61) XYZRecruitmentDatabase 00:00:01 0 rows
Ready
Ln 158 Col 1 Ch 1 INS
Type here to search
2:15 AM
5/2/2020

3.4 TRIGGERS

In this project I've used 2 triggers. According to the problem statement whenever a blacklisted candidate applies It should say that the candidate is blacklisted therefore the candidate is prohibited to apply. For this I've created a trigger. When ever a candidate applies we update the candidate details column and when ever the candidate provides his name we check if the name is already present and if the name is matched with the JobApplication table's name we see the status of the candidate and if its Blacklisted the trigger works and we the the message stating he cant apply.

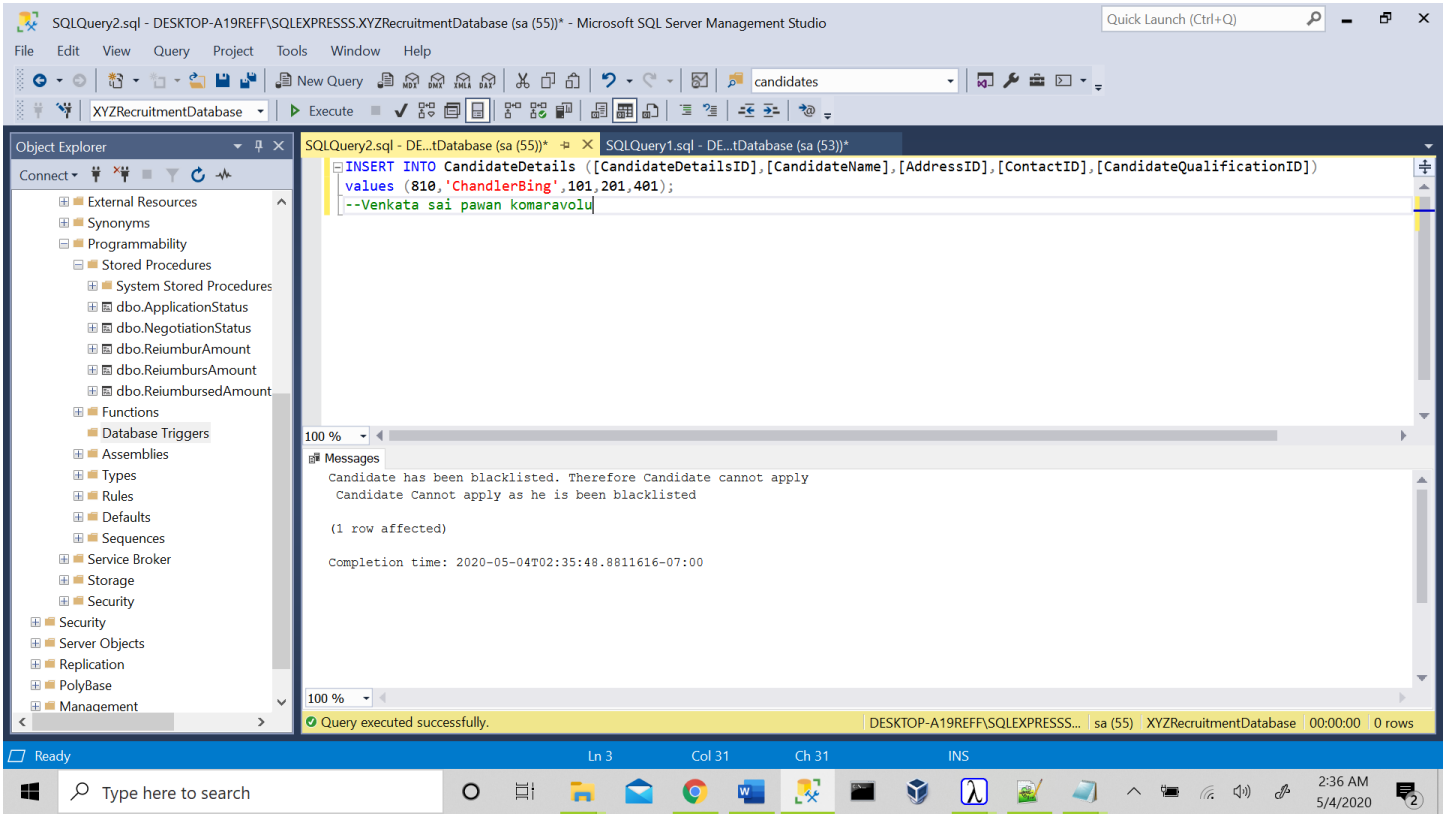
```
USE XYZRecruitmentDatabase;
GO

CREATE TRIGGER tr_blacklisted_candidate
ON CandidateDetails
FOR INSERT
AS
IF EXISTS
(Select * FROM JobApplication JOIN CandidateDetails
 ON JobApplication.CandidateDetailsID = CandidateDetails.CandidateDetailsID
 WHERE (JobApplication.CandidateName = CandidateDetails.CandidateName)
 AND
 (JobApplication.CandidateStatus = 'Blacklisted'))
 BEGIN
   PRINT 'Candidate has been blacklisted. Therefore Candidate cannot apply';
 END;
```



INSERT INTO CandidateDetails
([CandidateDetailsID],[CandidateName],[AddressID],[ContactID],[CandidateQualificationID])
values (810,'ChandlerBing',101,201,401);

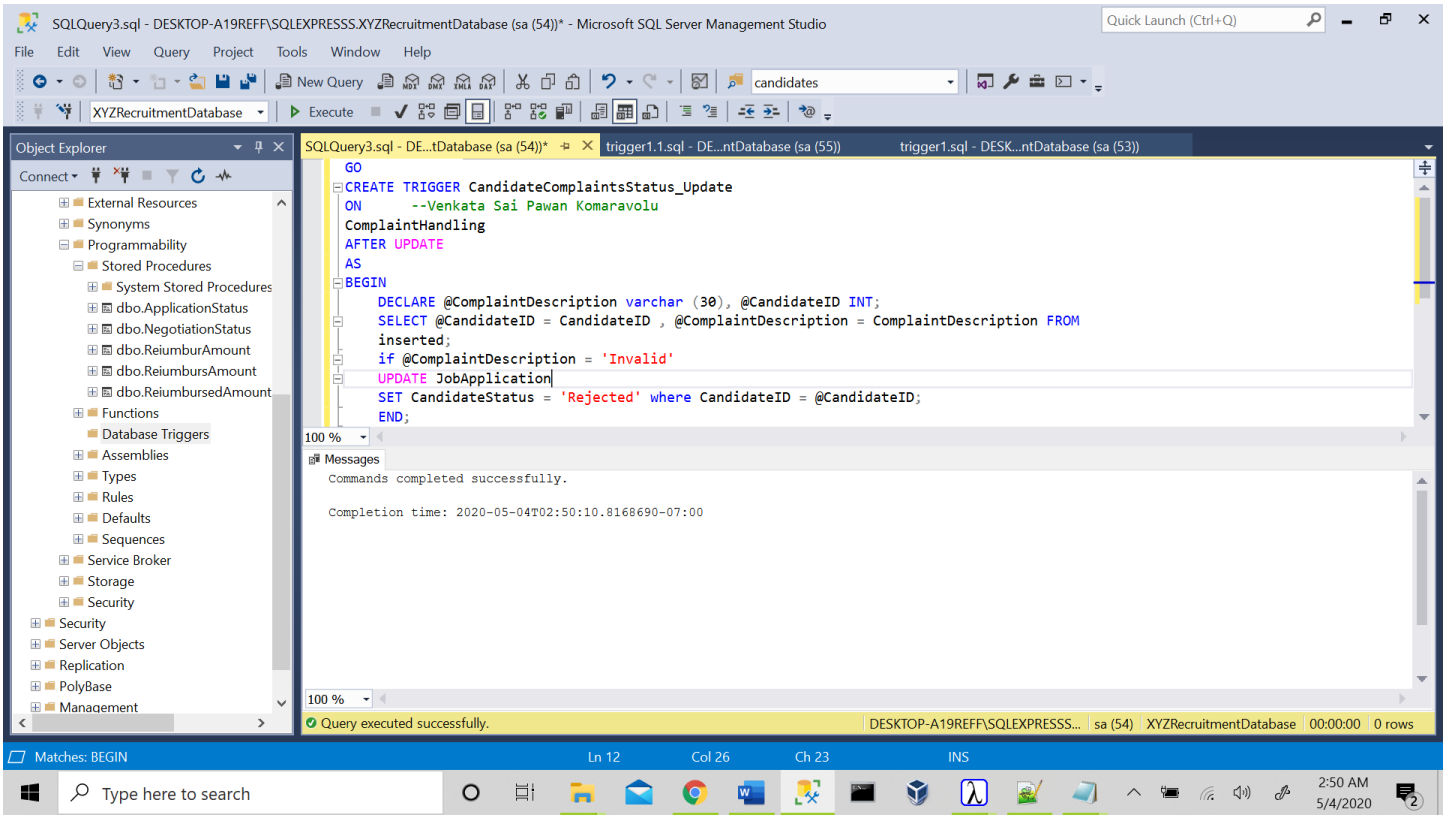
Here ChandlerBing is blacklisted and when he applies again name is checked and if the status is blacklisted then we get the message.



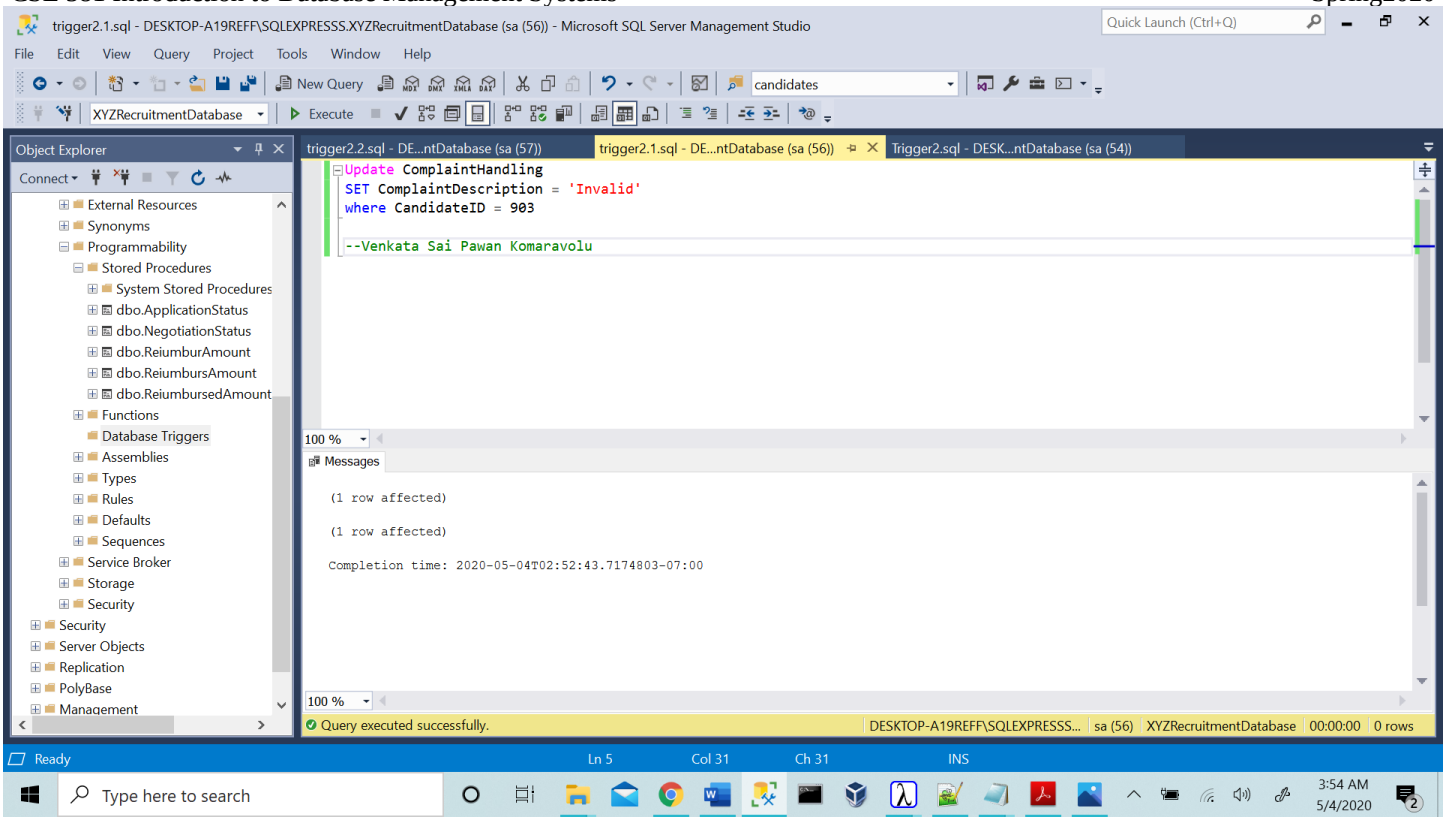
Another condition mentioned is that when ever a candidate gets rejected he can complain and if the complaint is approved he will be kept in the interview list otherwise he will be rejected. For this I've created a trigger and when ever there is update in the Complaint Description and it says Invalid then the candidate is rejected and his status is update to rejected.

```
GO
CREATE TRIGGER CandidateComplaintsStatus_Update
ON
ComplaintHandling
AFTER UPDATE
AS
BEGIN
        DECLARE @ComplaintDescription varchar (30), @CandidateID INT;
        SELECT @CandidateID = CandidateID , @ComplaintDescription = ComplaintDescription FROM
        inserted;
        if @ComplaintDescription = 'Invalid'
        UPDATE JobApplication
        SET CandidateStatus = 'Rejected' where CandidateID = @CandidateID;
```

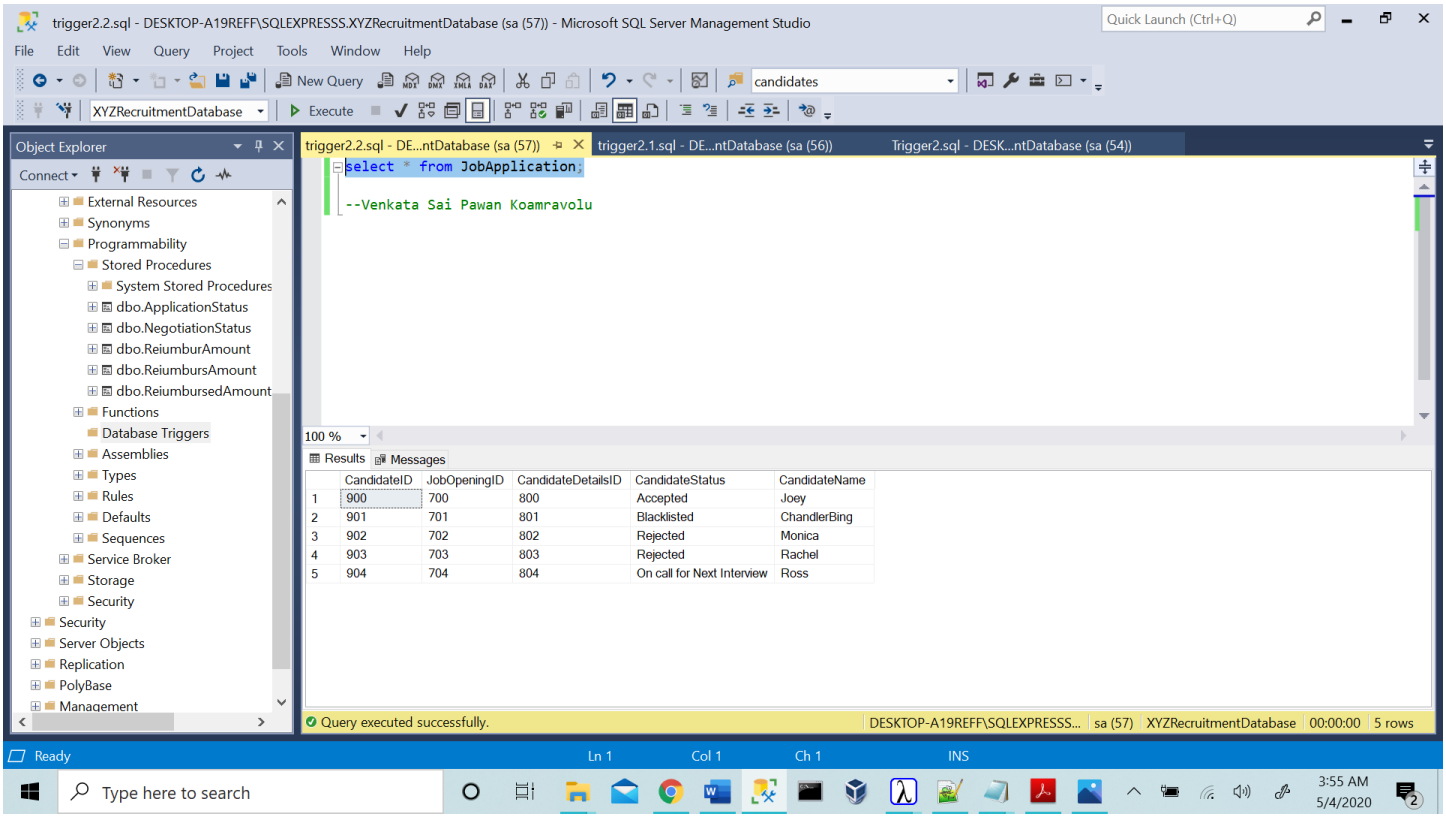
```
END;
```



Update ComplaintHandling
SET ComplaintDescription = 'Invalid'
where CandidateID = 903



select * from JobApplication;



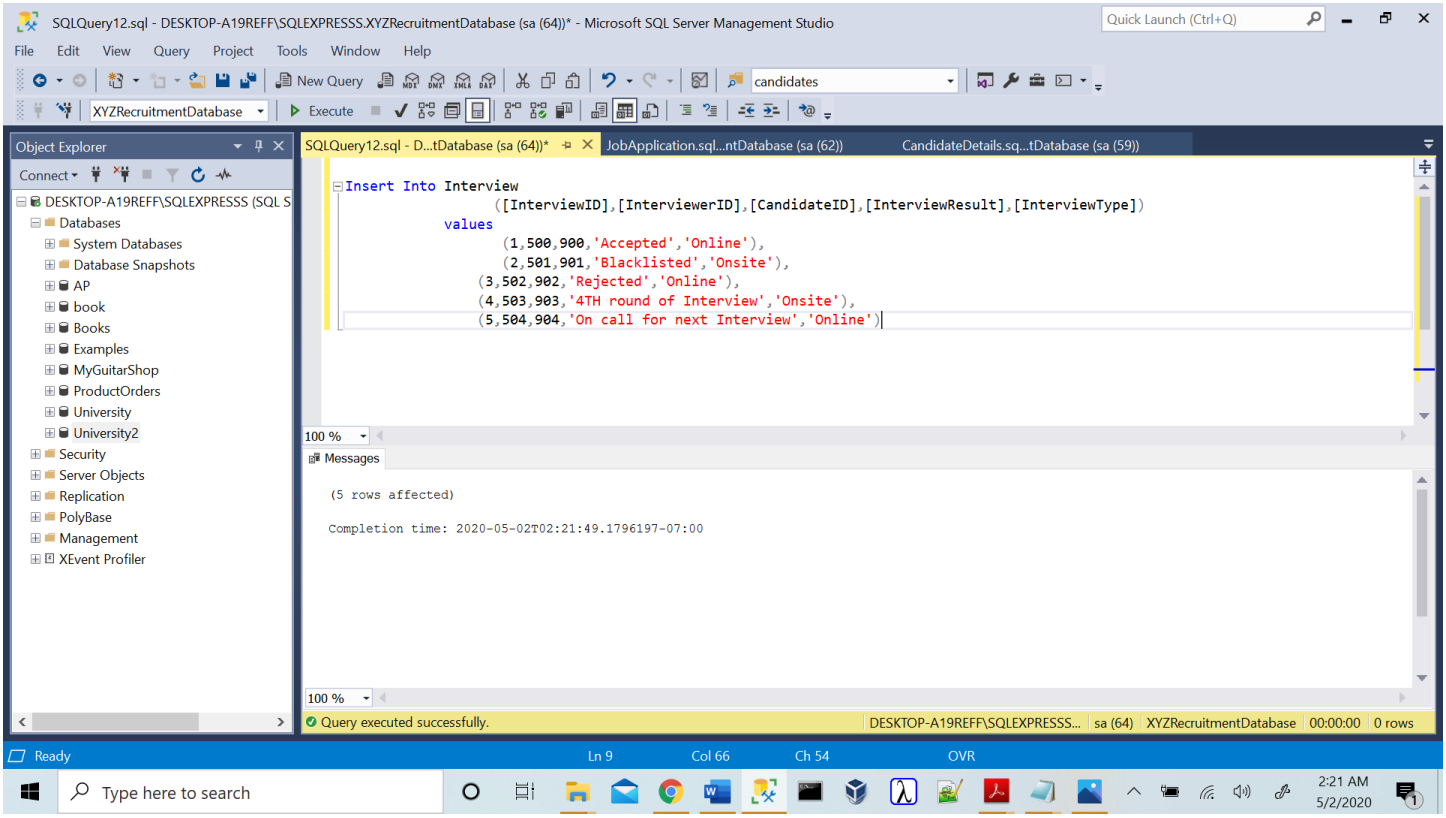
	CandidateID	JobOpeningID	CandidateDetailsID	CandidateStatus	CandidateName
1	900	700	800	Accepted	Joey
2	901	701	801	Blacklisted	ChandlerBing
3	902	702	802	Rejected	Monica
4	903	703	803	Rejected	Rachel
5	904	704	804	On call for Next Interview	Ross

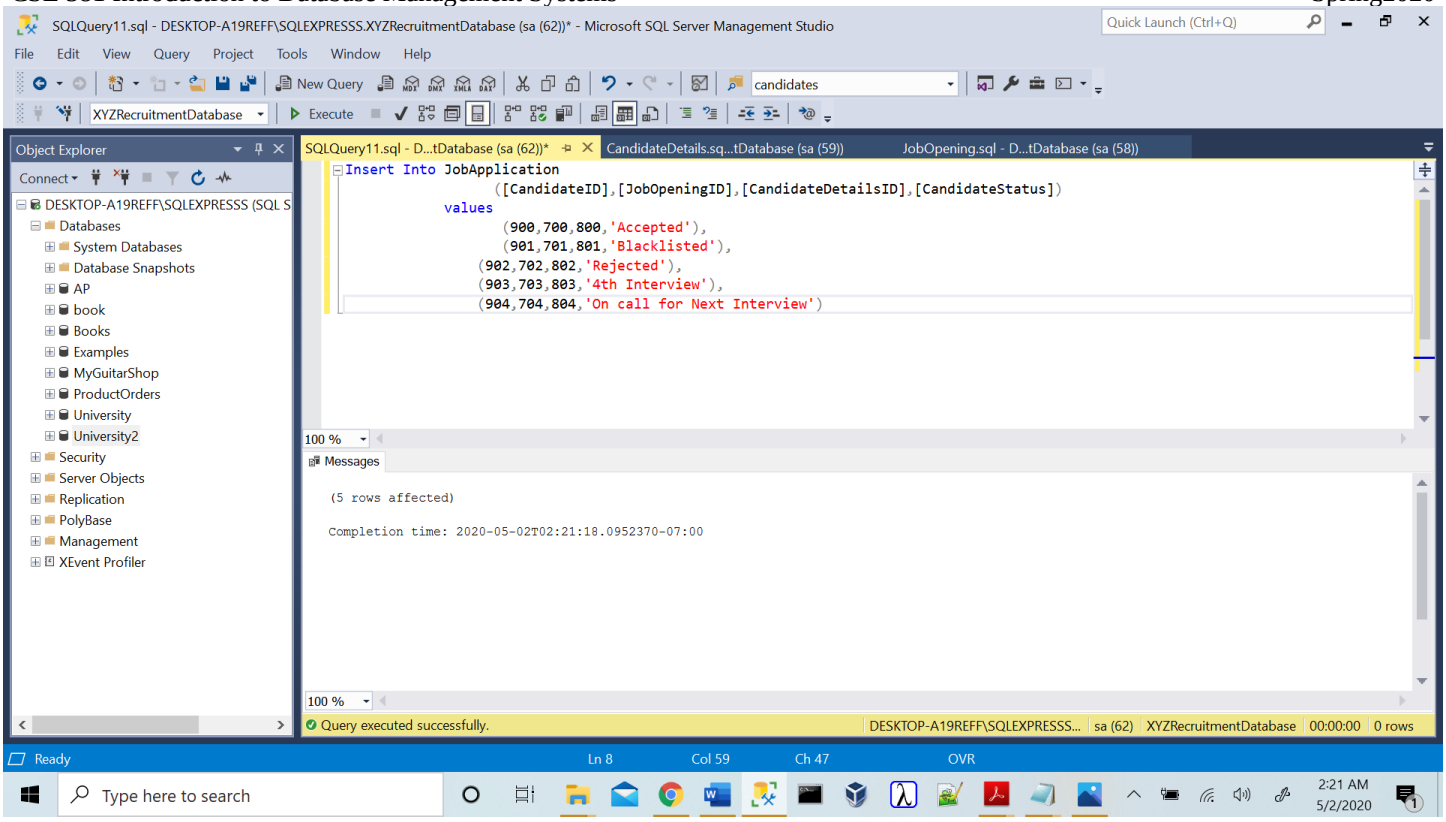
4. TESTING PHASE

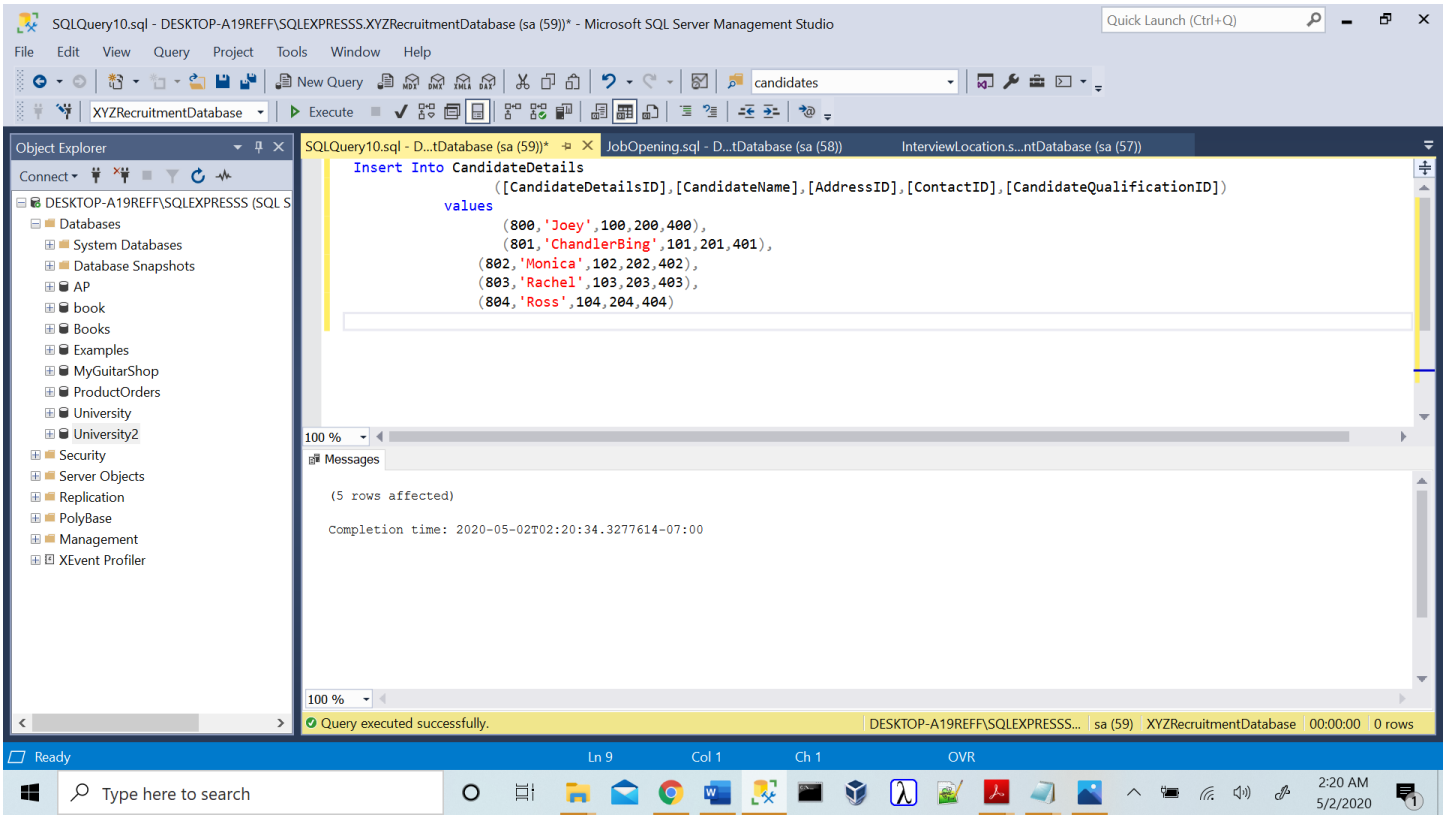
In testing phase we populate the database by inserting values into tables of the database which we have created and then to test the scenarios we have used views, stored procedure, Functions, Scripts and transaction.

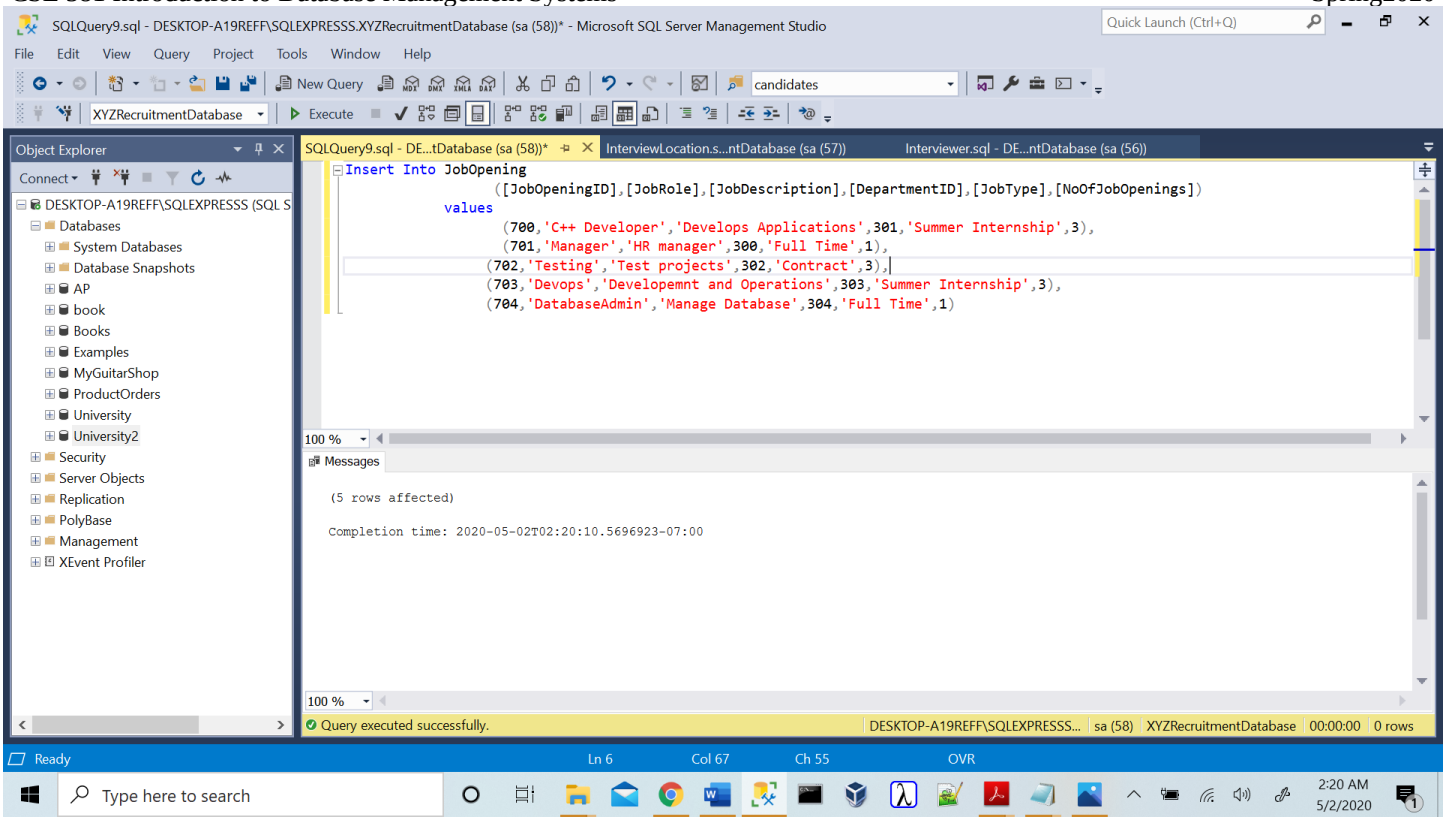
4.1 POPULATING DATABASE

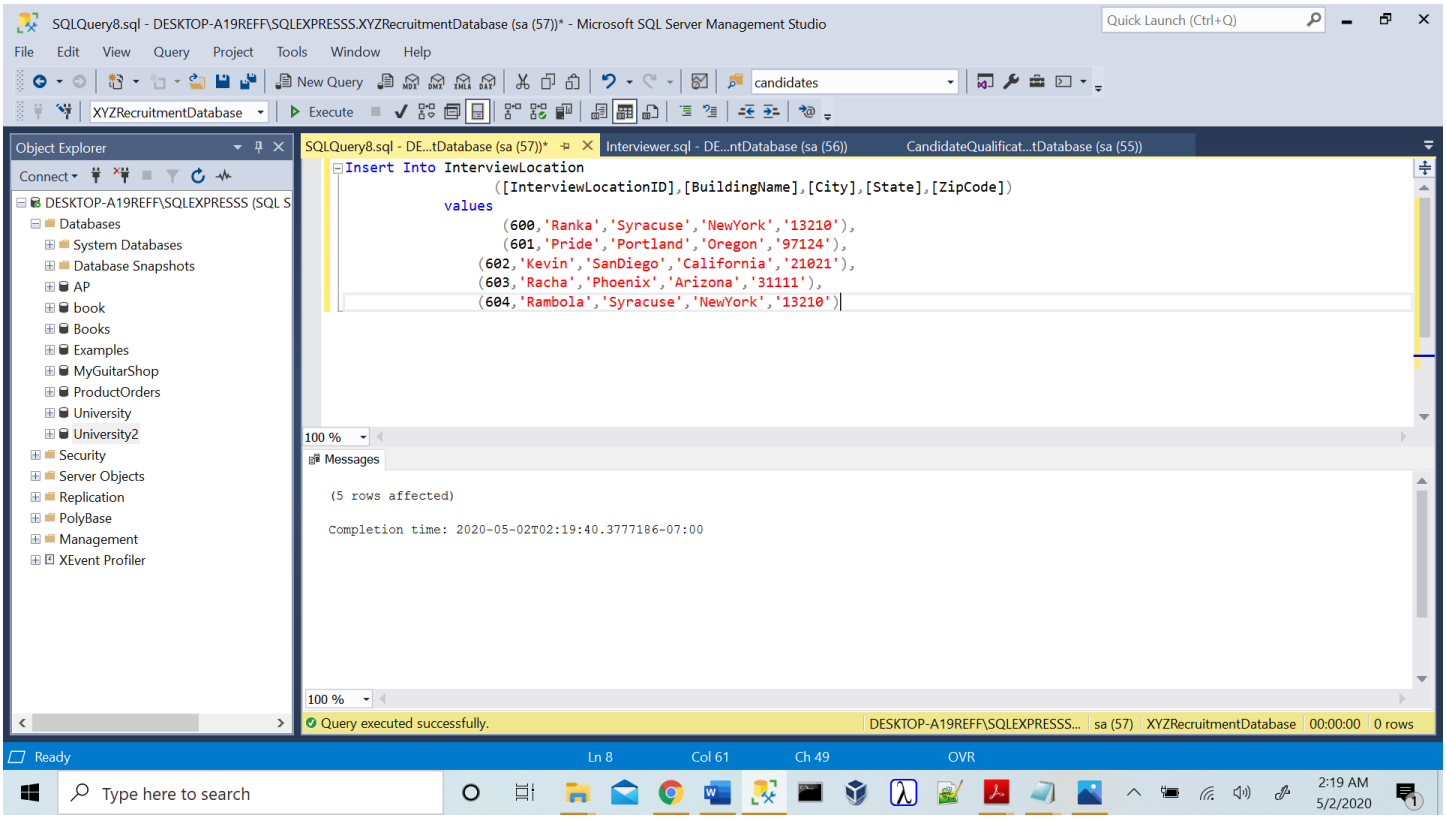
We now insert values into the tables and columns using the INSERT INTO statement.

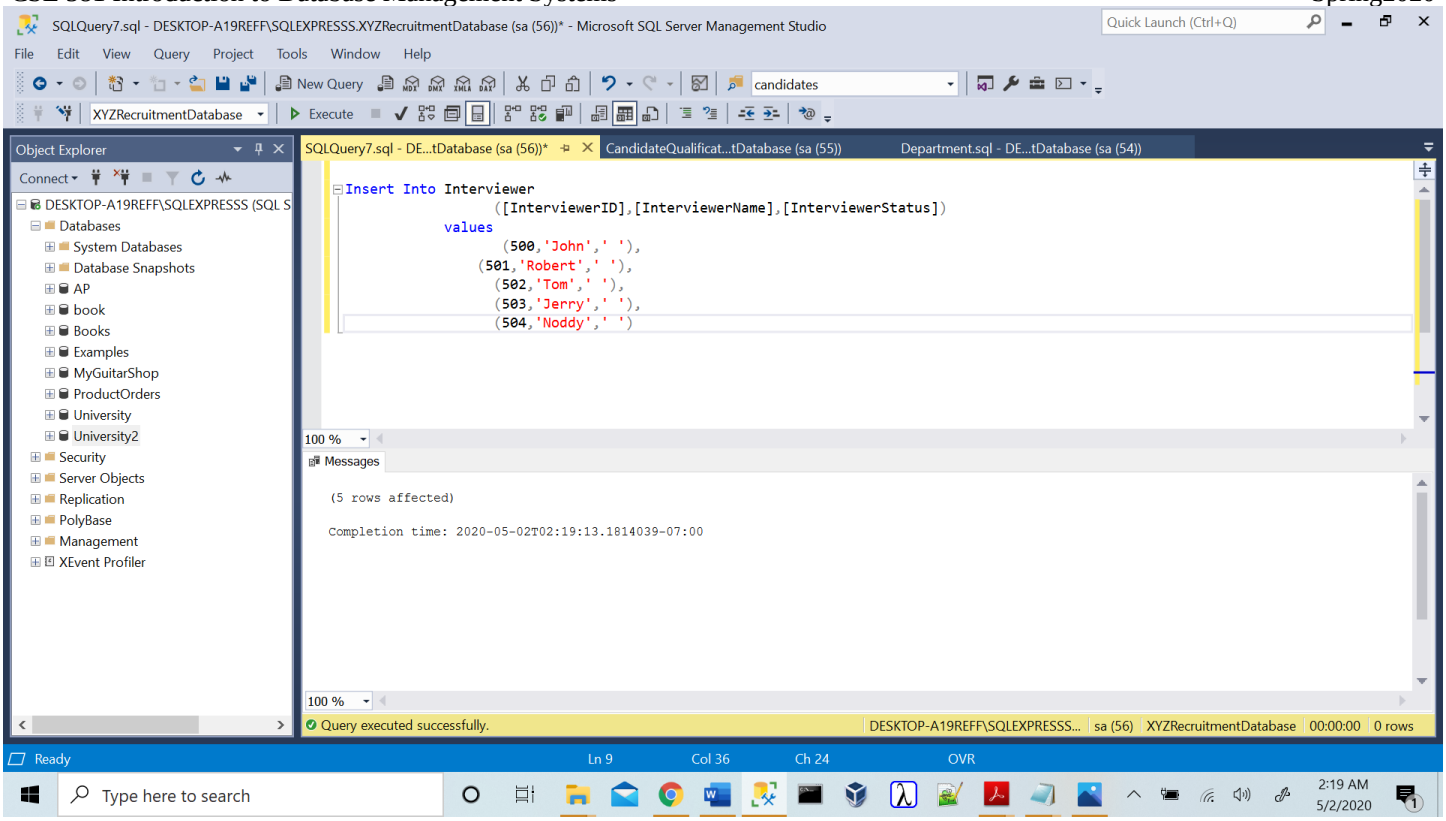


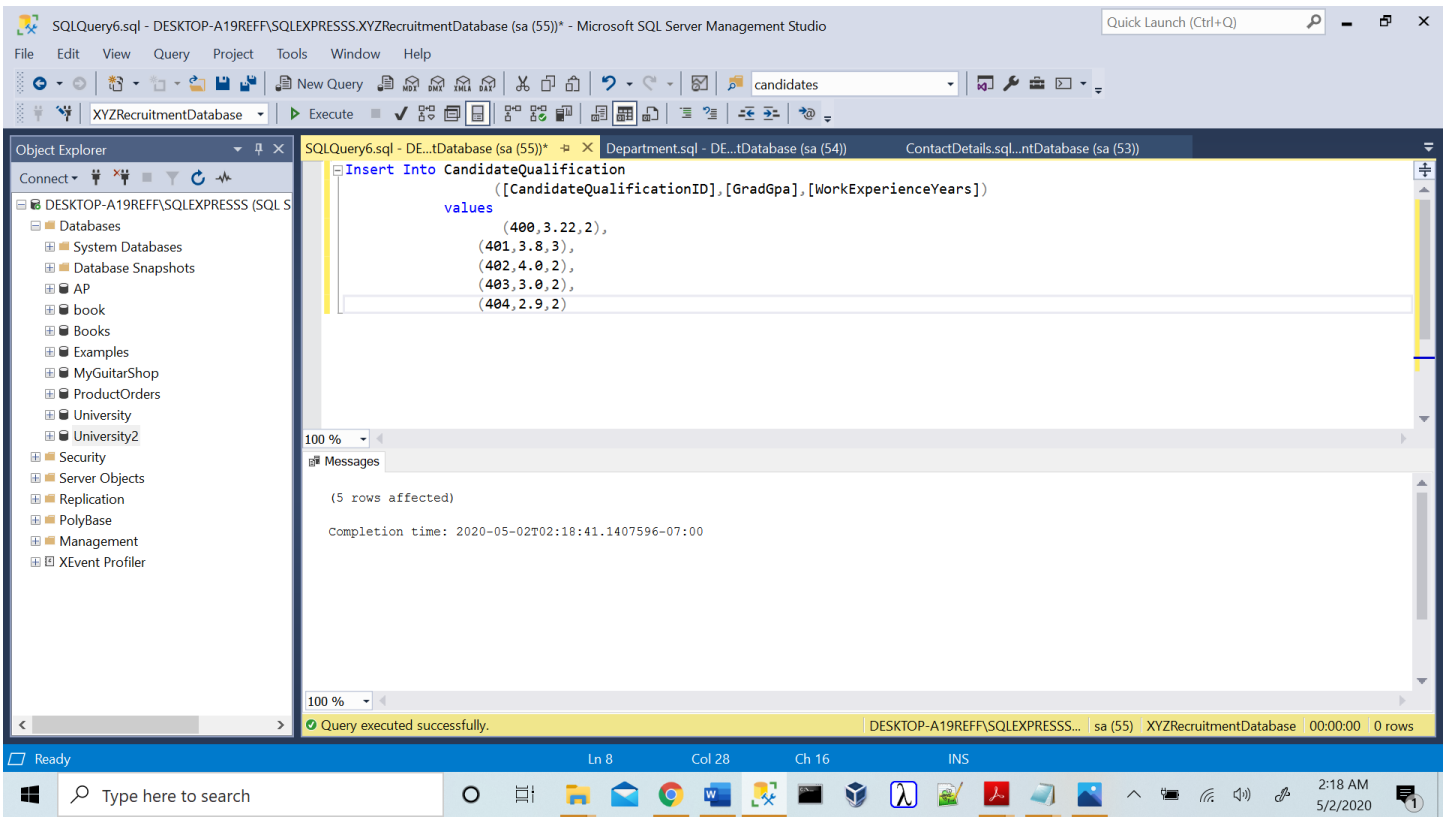

SQLQuery11.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (62))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery11.sql - D...tDatabase (sa (62))*
CandidateDetails.sq...tDatabase (sa (59))
JobOpening.sql - D...tDatabase (sa (58))
Insert Into JobApplication
([CandidateID],[JobOpeningID],[CandidateDetailsID],[CandidateStatus])
values
(900,700,800,'Accepted'),
(901,701,801,'Blacklisted'),
(902,702,802,'Rejected'),
(903,703,803,'4th Interview'),
(904,704,804,'On call for Next Interview')
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:21:18.0952370-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (62) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 59 Ch 47 OVR
Type here to search
2:21 AM
5/2/2020

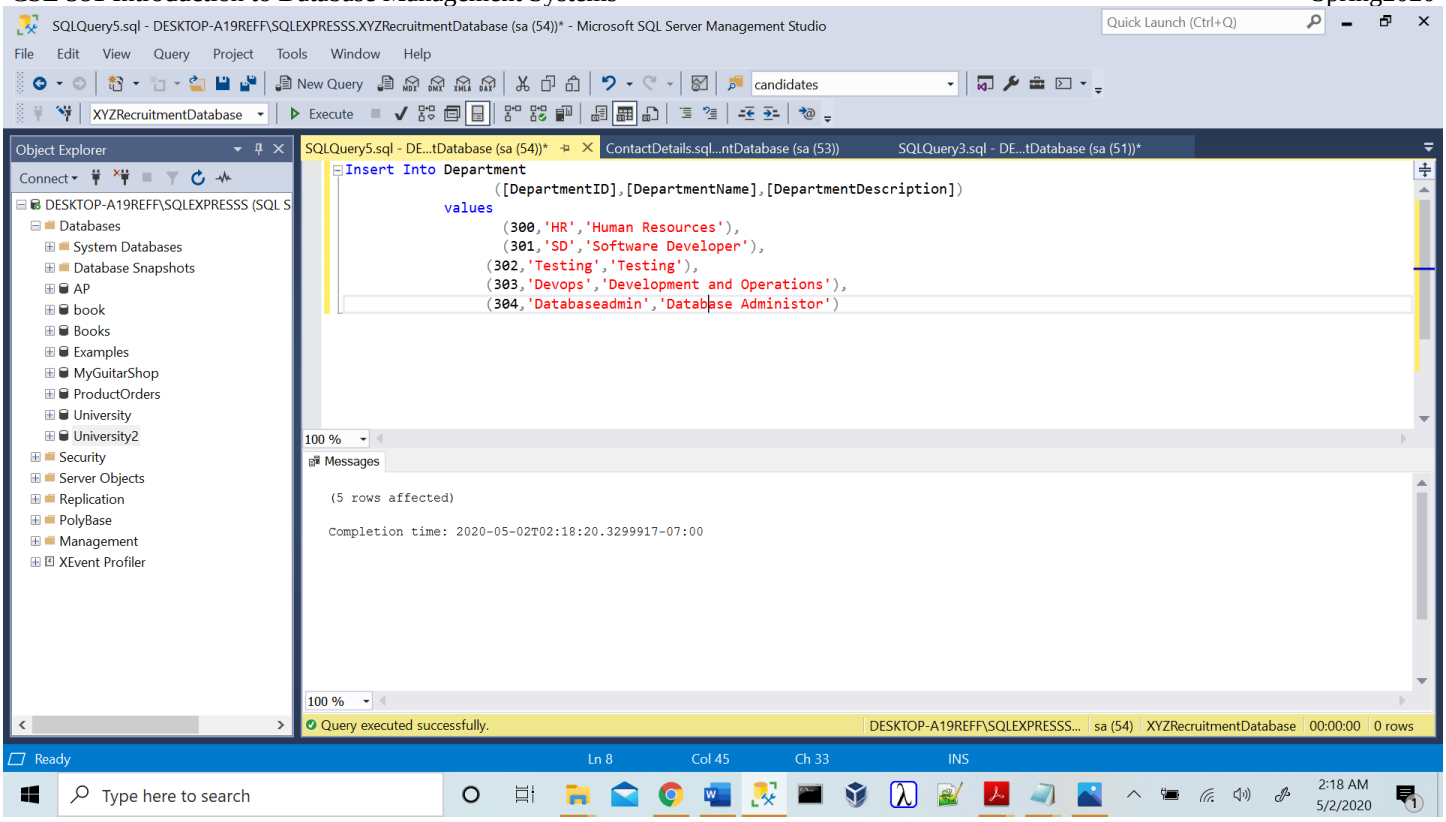

SQLQuery10.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (59))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery10.sql - D...tDatabase (sa (59))*
JobOpening.sql - D...tDatabase (sa (58))
InterviewLocation.s...ntDatabase (sa (57))
Insert Into CandidateDetails
([CandidateDetailsID],[CandidateName],[AddressID],[ContactID],[CandidateQualificationID])
values
(800,'Joey',100,200,400),
(801,'ChandlerBing',101,201,401),
(802,'Monica',102,202,402),
(803,'Rachel',103,203,403),
(804,'Ross',104,204,404)
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:20:34.3277614-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (59) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 9 Col 1 Ch 1 OVR
Type here to search
2:20 AM
5/2/2020

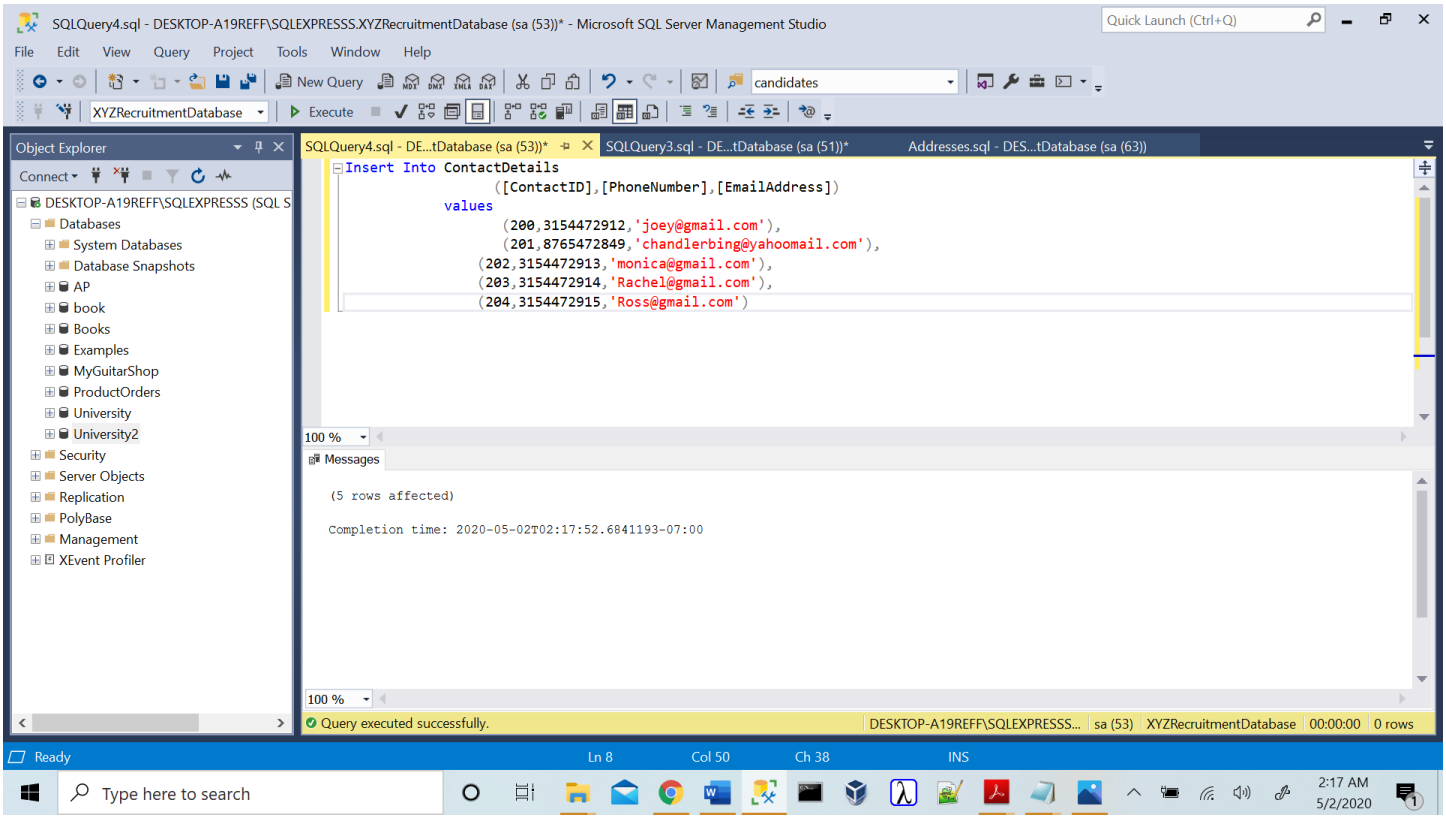

SQLQuery9.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery9.sql - DE...tDatabase (sa (58))*
InterviewLocation.s...ntDatabase (sa (57))
Interviewer.sql - DE...ntDatabase (sa (56))
Insert Into JobOpening
([JobOpeningID],[JobRole],[JobDescription],[DepartmentID],[JobType],[NoOfJobOpenings])
values
(700,'C++ Developer','Develops Applications',301,'Summer Internship',3),
(701,'Manager','HR manager',300,'Full Time',1),
(702,'Testing','Test projects',302,'Contract',3),
(703,'Devops','Developemnt and Operations',303,'Summer Internship',3),
(704,'DatabaseAdmin','Manage Database',304,'Full Time',1)
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:20:10.5696923-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (58) XYZRecruitmentDatabase 00:00:00 0 rows
Ready Ln 6 Col 67 Ch 55 OVR
Type here to search
2:20 AM
5/2/2020

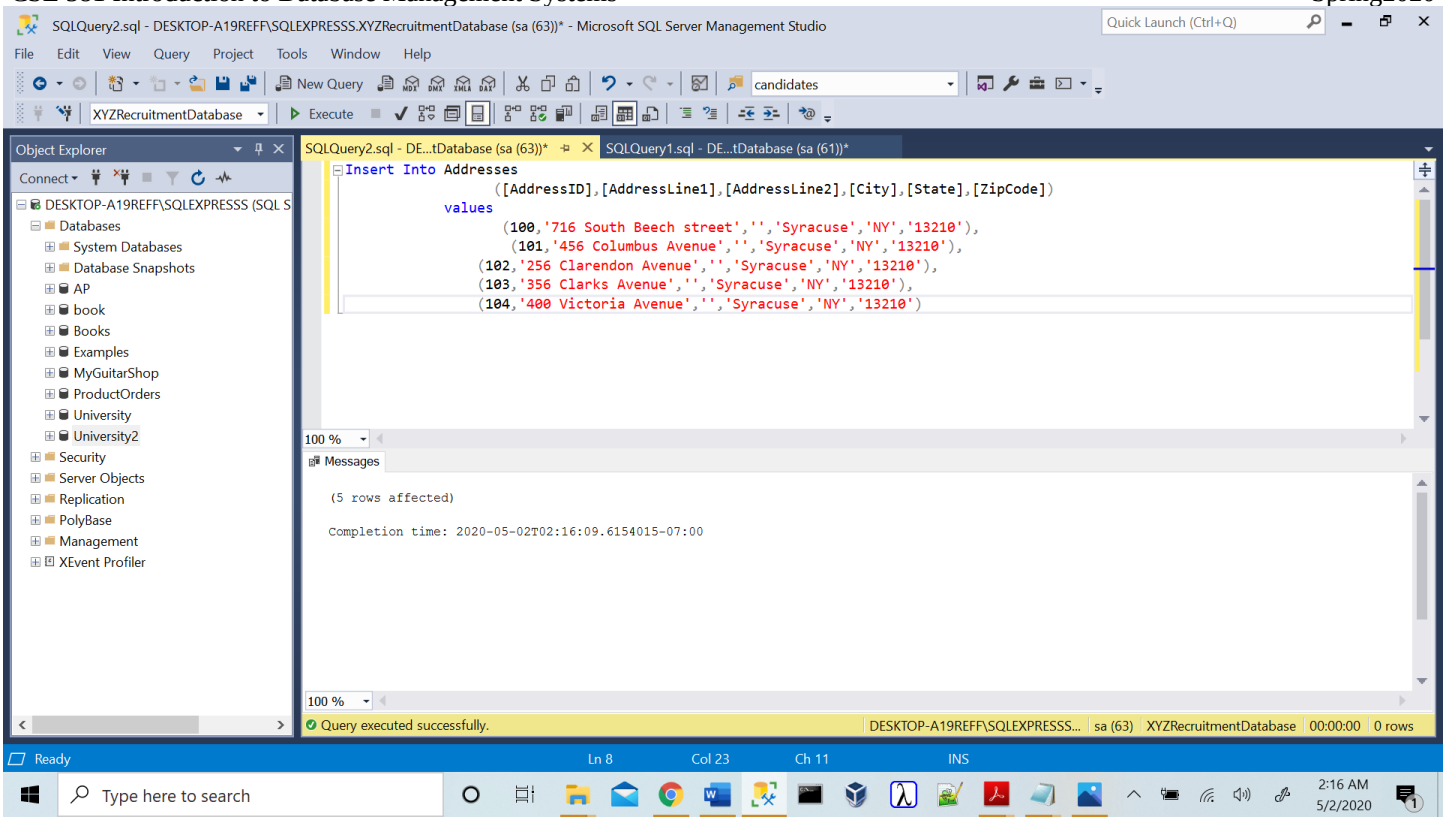

SQLQuery8.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (57))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery8.sql - DE...tDatabase (sa (57))*
Interviewer.sql - DE...ntDatabase (sa (56))
CandidateQualificat...tDatabase (sa (55))
Insert Into InterviewLocation
([InterviewLocationID],[BuildingName],[City],[State],[ZipCode])
values
(600,'Ranka','Syracuse','NewYork','13210'),
(601,'Pride','Portland','Oregon','97124'),
(602,'Kevin','SanDiego','California','21021'),
(603,'Racha','Phoenix','Arizona','31111'),
(604,'Rambola','Syracuse','NewYork','13210')
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:19:40.3777186-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (57) XYZRecruitmentDatabase 00:00:00 0 rows
Ready Ln 8 Col 61 Ch 49 OVR
Type here to search
2:19 AM
5/2/2020

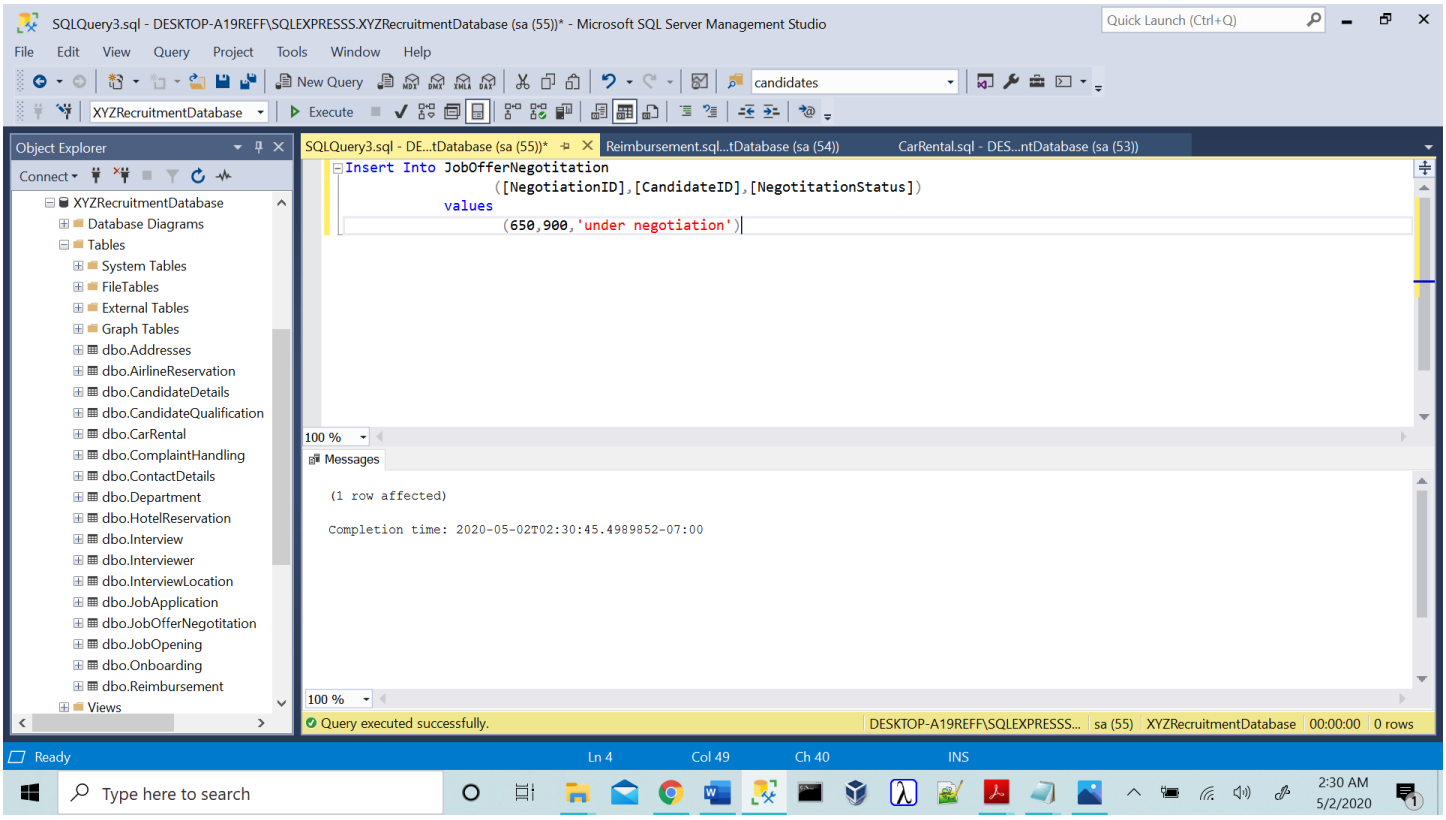

SQLQuery7.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (56))* - Microsoft SQL Server Management Studio
Insert Into Interviewer
([InterviewerID],[InterviewerName],[InterviewerStatus])
values
(500,'John',' '),
(501,'Robert',' '),
(502,'Tom',' '),
(503,'Jerry',' '),
(504,'Noddy',' ')
(5 rows affected)
Completion time: 2020-05-02T02:19:13.1814039-07:00
Query executed successfully.
Ready
Ln 9
Col 36
Ch 24
OVR
2:19 AM
5/2/2020

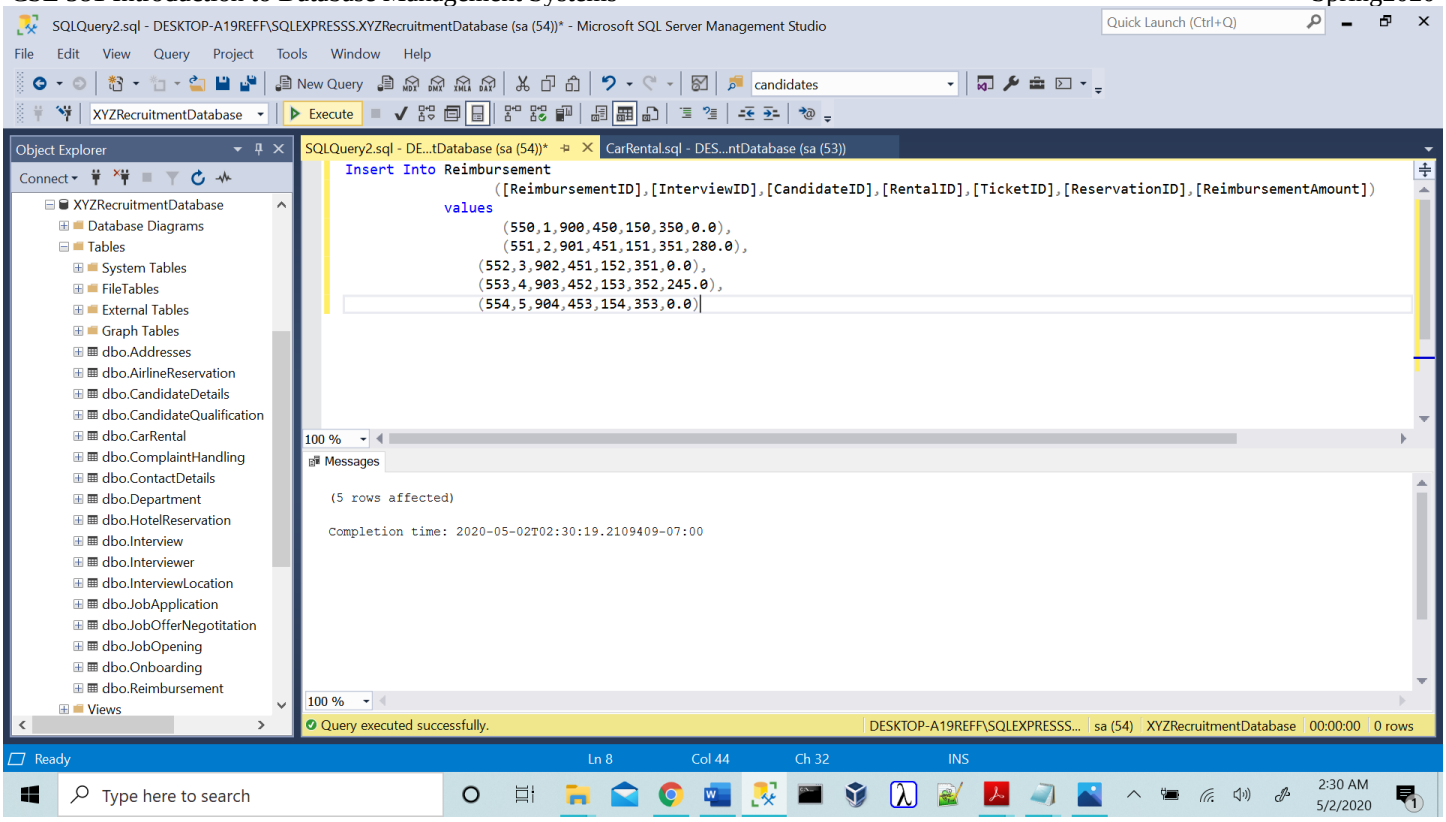

SQLQuery6.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (55))* - Microsoft SQL Server Management Studio
Insert Into CandidateQualification
([CandidateQualificationID],[GradGpa],[WorkExperienceYears])
values
(400,3.22,2),
(401,3.8,3),
(402,4.0,2),
(403,3.0,2),
(404,2.9,2)
(5 rows affected)
Completion time: 2020-05-02T02:18:41.1407596-07:00
Query executed successfully.
Ready
Ln 8
Col 28
Ch 16
INS
2:18 AM
5/2/2020

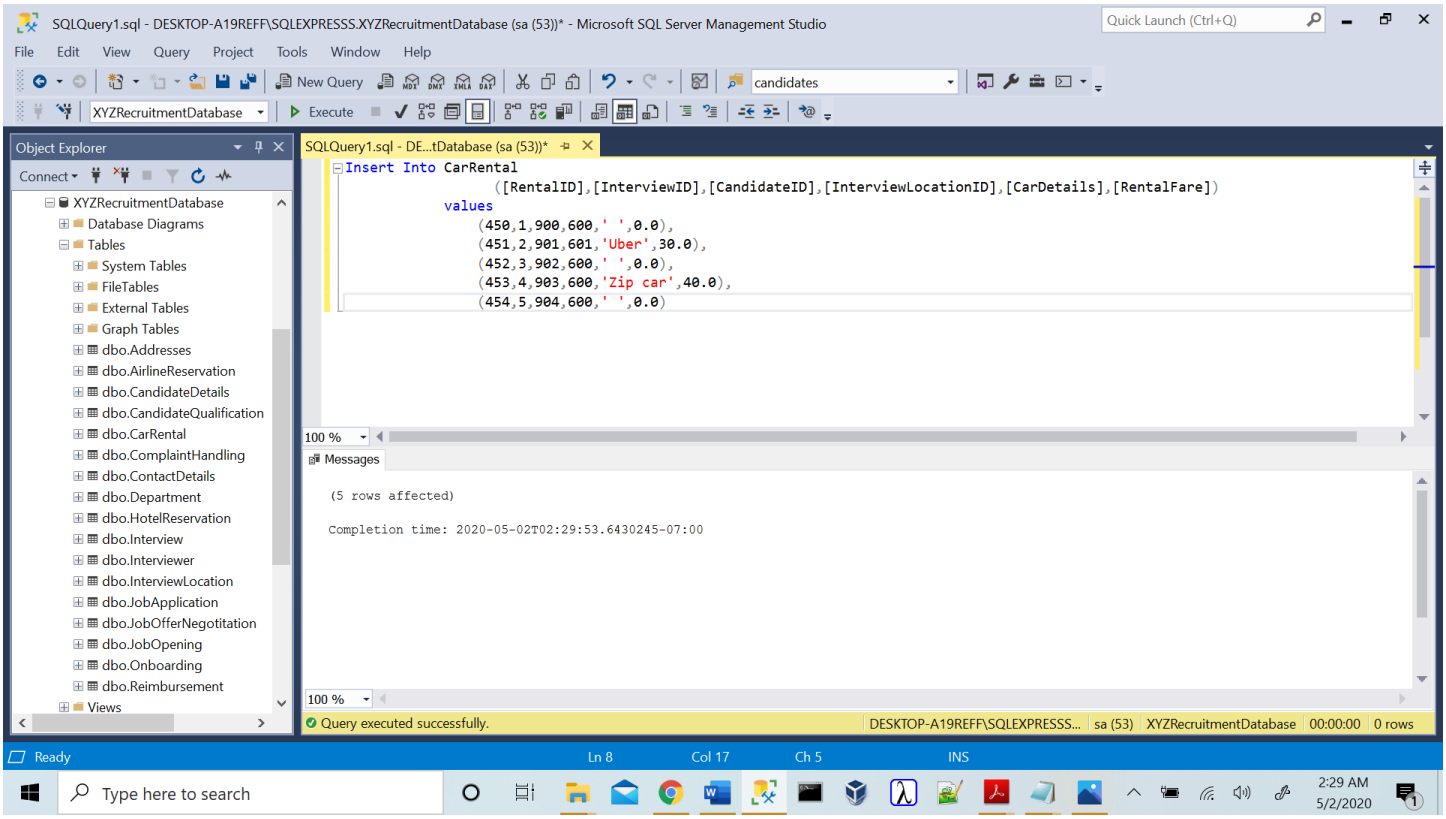

SQLQuery5.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (54))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery5.sql - DE...tDatabase (sa (54))*
ContactDetails.sql...ntDatabase (sa (53))
SQLQuery3.sql - DE...tDatabase (sa (51))*
Insert Into Department
([DepartmentID],[DepartmentName],[DepartmentDescription])
values
(300,'HR','Human Resources'),
(301,'SD','Software Developer'),
(302,'Testing','Testing'),
(303,'Devops','Development and Operations'),
(304,'Databaseadmin','Database Administor')
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:18:20.3299917-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (54) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 45 Ch 33 INS
Type here to search
2:18 AM
5/2/2020

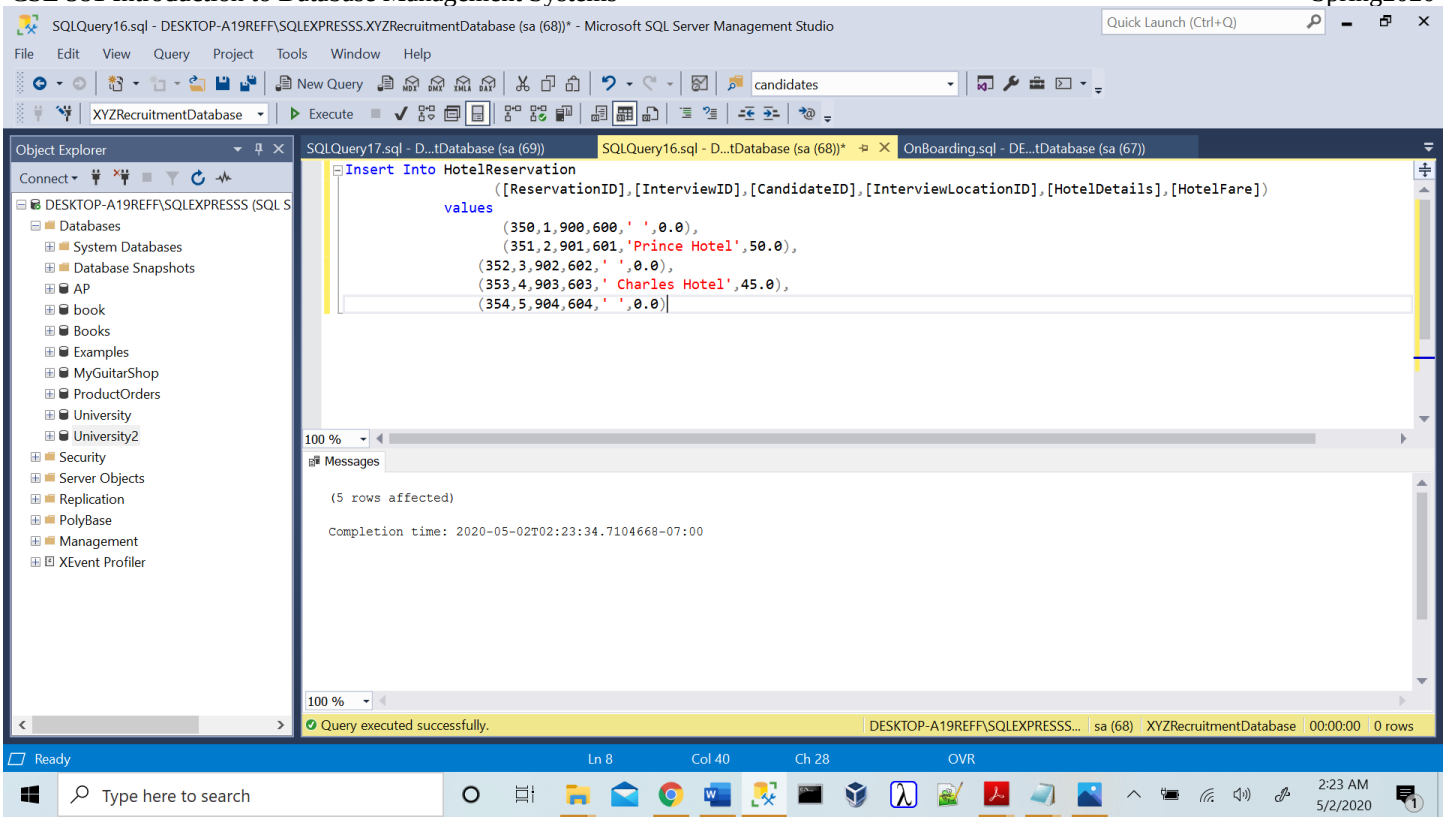

SQLQuery4.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery4.sql - DE...tDatabase (sa (53))*
SQLQuery3.sql - DE...tDatabase (sa (51))*
Addresses.sql - DES...tDatabase (sa (63))
Insert Into ContactDetails
([ContactID],[PhoneNumber],[EmailAddress])
values
(200,3154472912,'joey@gmail.com'),
(201,8765472849,'chandlerbing@yahoomail.com'),
(202,3154472913,'monica@gmail.com'),
(203,3154472914,'Rachel@gmail.com'),
(204,3154472915,'Ross@gmail.com')
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:17:52.6841193-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (53) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 50 Ch 38 INS
Type here to search
2:17 AM
5/2/2020

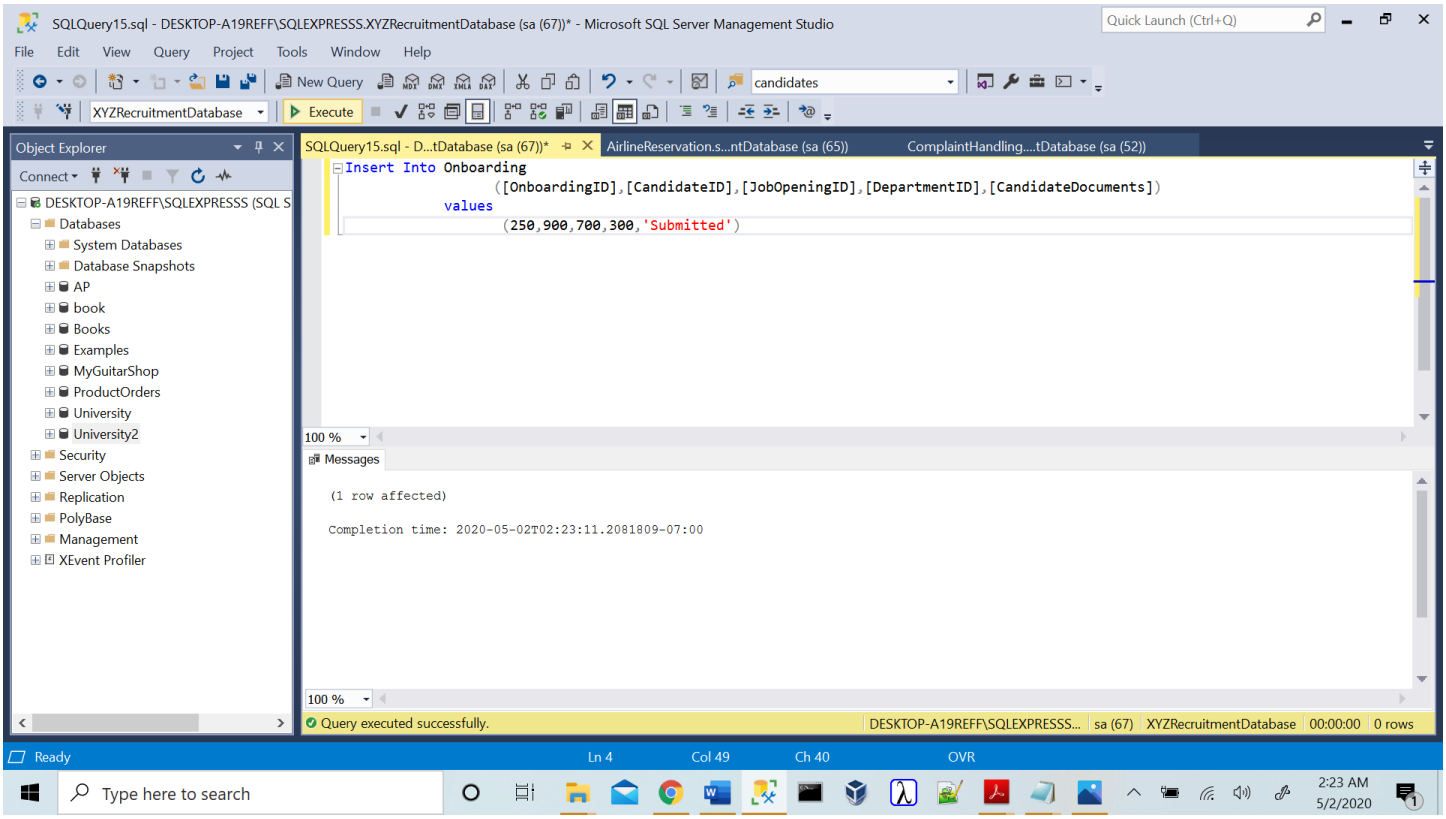

SQLQuery2.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (63))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery2.sql - DE...tDatabase (sa (63))*
SQLQuery1.sql - DE...tDatabase (sa (61))*
Insert Into Addresses
([AddressID],[AddressLine1],[AddressLine2],[City],[State],[ZipCode])
values
(100,'716 South Beech street','','Syracuse','NY','13210'),
(101,'456 Columbus Avenue','','Syracuse','NY','13210'),
(102,'256 Clarendon Avenue','','Syracuse','NY','13210'),
(103,'356 Clarks Avenue','','Syracuse','NY','13210'),
(104,'400 Victoria Avenue','','Syracuse','NY','13210')
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:16:09.6154015-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (63) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 23 Ch 11 INS
Type here to search
2:16 AM
5/2/2020

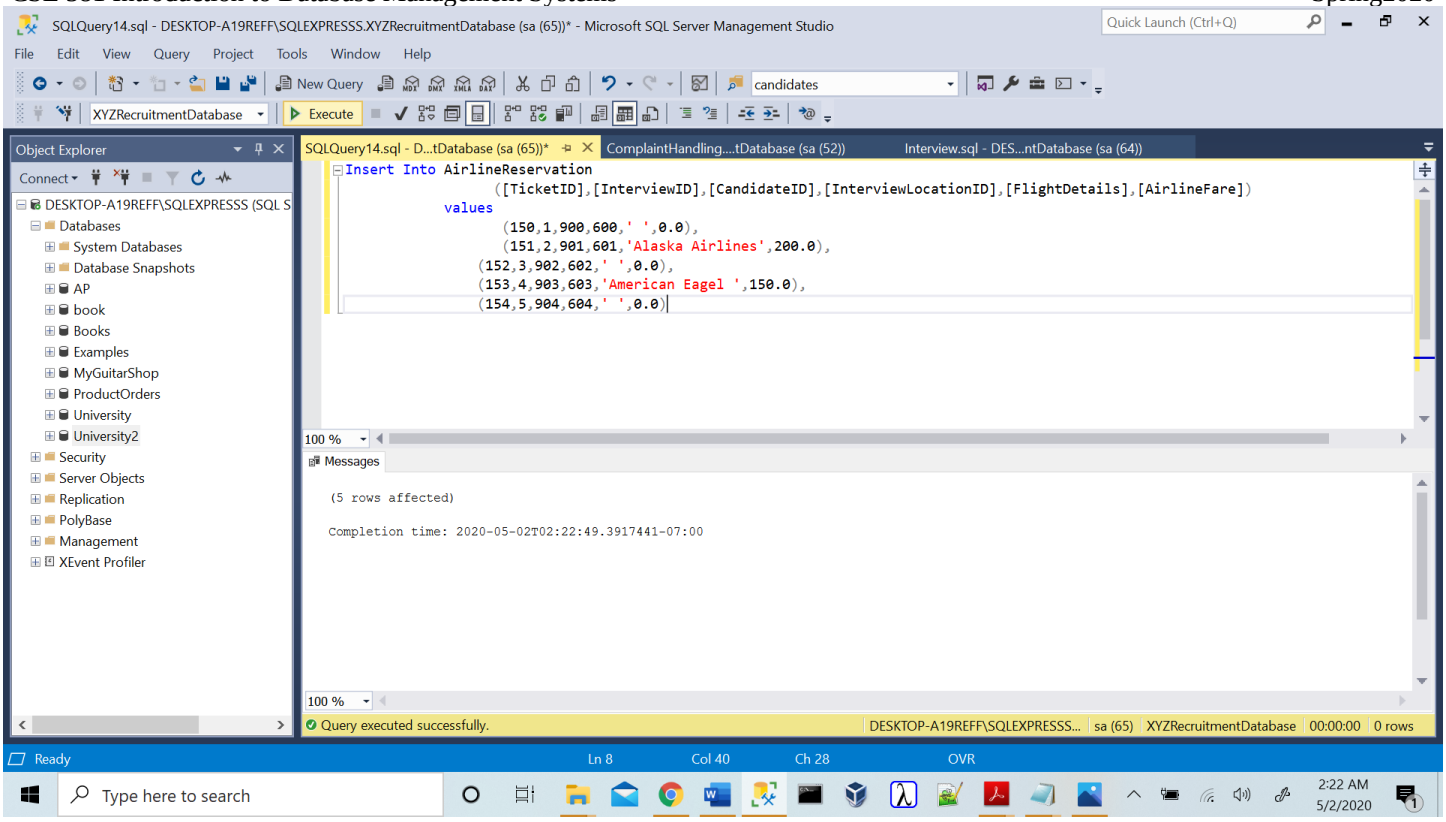

SQLQuery3.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (55))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
XYZRecruitmentDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Addresses
dbo.AirlineReservation
dbo.CandidateDetails
dbo.CandidateQualification
dbo.CarRental
dbo.ComplaintHandling
dbo.ContactDetails
dbo.Department
dbo.HotelReservation
dbo.Interview
dbo.Interviewer
dbo.InterviewLocation
dbo.JobApplication
dbo.JobOfferNegotitation
dbo.JobOpening
dbo.Onboarding
dbo.Reimbursement
Views
SQLQuery3.sql - DE...tDatabase (sa (55))*
Reimbursement.sql...tDatabase (sa (54))
CarRental.sql - DES...ntDatabase (sa (53))
Insert Into JobOfferNegotitation
([NegotiationID],[CandidateID],[NegotitationStatus])
values
(650,900,'under negotiation')
100 %
Messages
(1 row affected)
Completion time: 2020-05-02T02:30:45.4989852-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (55) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 4 Col 49 Ch 40 INS
Type here to search
2:30 AM
5/2/2020

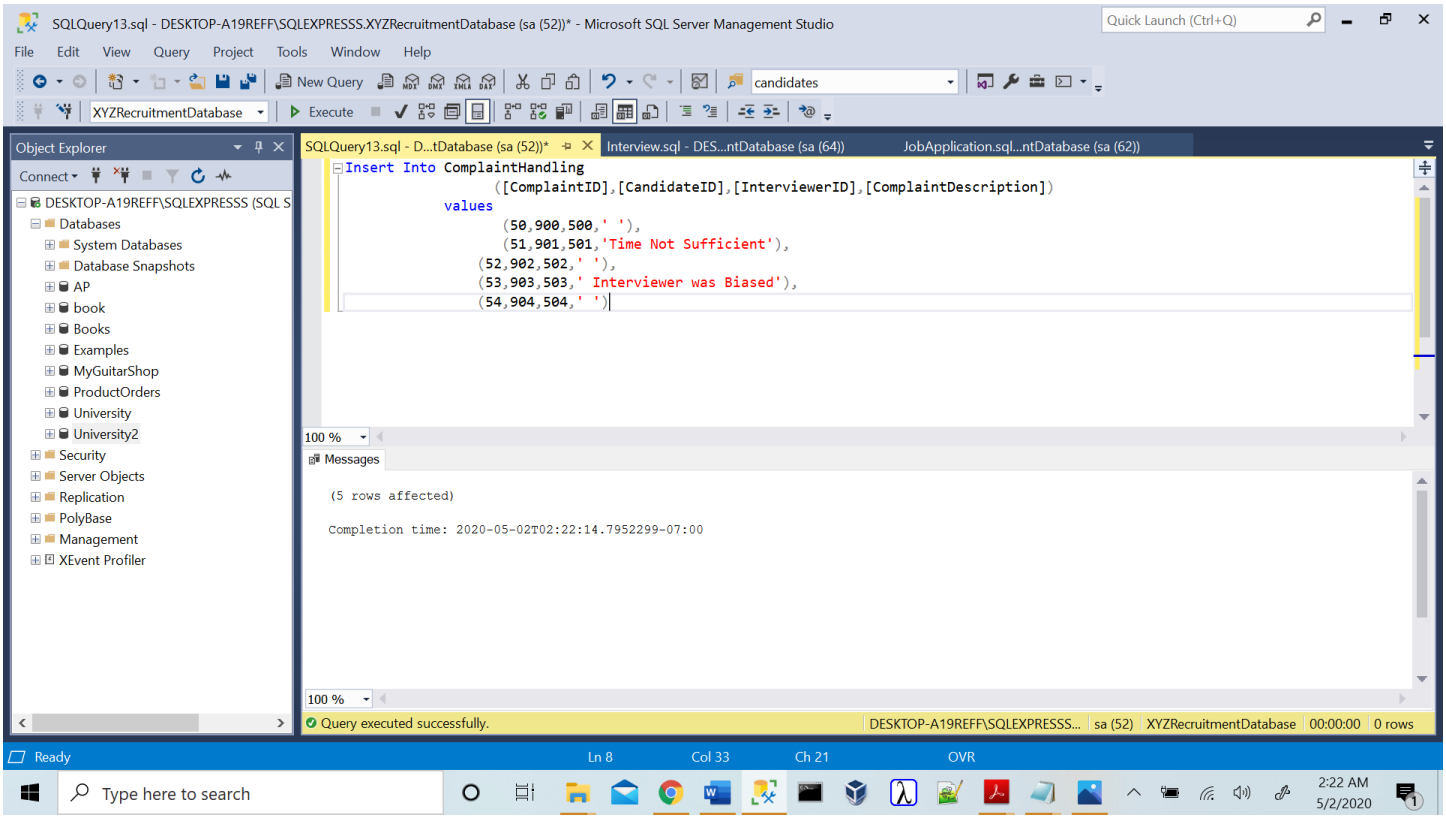

SQLQuery2.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (54))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
XYZRecruitmentDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Addresses
dbo.AirlineReservation
dbo.CandidateDetails
dbo.CandidateQualification
dbo.CarRental
dbo.ComplaintHandling
dbo.ContactDetails
dbo.Department
dbo.HotelReservation
dbo.Interview
dbo.Interviewer
dbo.InterviewLocation
dbo.JobApplication
dbo.JobOfferNegotitation
dbo.JobOpening
dbo.Onboarding
dbo.Reimbursement
Views
SQLQuery2.sql - DE...tDatabase (sa (54))*
CarRental.sql - DES...ntDatabase (sa (53))
Insert Into Reimbursement
([ReimbursementID],[InterviewID],[CandidateID],[RentalID],[TicketID],[ReservationID],[ReimbursementAmount])
values
(550,1,900,450,150,350,0.0),
(551,2,901,451,151,351,280.0),
(552,3,902,451,152,351,0.0),
(553,4,903,452,153,352,245.0),
(554,5,904,453,154,353,0.0)
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:30:19.2109409-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (54) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 44 Ch 32 INS
Type here to search
2:30 AM
5/2/2020


SQLQuery1.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (53))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
XYZRecruitmentDatabase
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Addresses
dbo.AirlineReservation
dbo.CandidateDetails
dbo.CandidateQualification
dbo.CarRental
dbo.ComplaintHandling
dbo.ContactDetails
dbo.Department
dbo.HotelReservation
dbo.Interview
dbo.Interviewer
dbo.InterviewLocation
dbo.JobApplication
dbo.JobOfferNegotitation
dbo.JobOpening
dbo.Onboarding
dbo.Reimbursement
Views
SQLQuery1.sql - DE...tDatabase (sa (53))*
Insert Into CarRental
([RentalID],[InterviewID],[CandidateID],[InterviewLocationID],[CarDetails],[RentalFare])
values
(450,1,900,600,' ',0.0),
(451,2,901,601,'Uber',30.0),
(452,3,902,600,' ',0.0),
(453,4,903,600,'Zip car',40.0),
(454,5,904,600,' ',0.0)
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:29:53.6430245-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (53) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 17 Ch 5 INS
Type here to search
2:29 AM
5/2/2020


SQLQuery16.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (68))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery17.sql - D...tDatabase (sa (69))
SQLQuery16.sql - D...tDatabase (sa (68))*
OnBoarding.sql - DE...tDatabase (sa (67))
Insert Into HotelReservation
([ReservationID],[InterviewID],[CandidateID],[InterviewLocationID],[HotelDetails],[HotelFare])
values
(350,1,900,600,' ',0.0),
(351,2,901,601,'Prince Hotel',50.0),
(352,3,902,602,' ',0.0),
(353,4,903,603,' Charles Hotel',45.0),
(354,5,904,604,' ',0.0)
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:23:34.7104668-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (68) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 40 Ch 28 OVR
Type here to search
2:23 AM
5/2/2020


SQLQuery15.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (67))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery15.sql - D...tDatabase (sa (67))*
AirlineReservation.s...ntDatabase (sa (65))
ComplaintHandling....tDatabase (sa (52))
Insert Into Onboarding
([OnboardingID],[CandidateID],[JobOpeningID],[DepartmentID],[CandidateDocuments])
values
(250,900,700,300,'Submitted')
100 %
Messages
(1 row affected)
Completion time: 2020-05-02T02:23:11.2081809-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (67) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 4 Col 49 Ch 40 OVR
Type here to search
2:23 AM
5/2/2020


SQLQuery14.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (65))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery14.sql - D...tDatabase (sa (65))*
ComplaintHandling....tDatabase (sa (52))
Interview.sql - DES...ntDatabase (sa (64))
Insert Into AirlineReservation
([TicketID],[InterviewID],[CandidateID],[InterviewLocationID],[FlightDetails],[AirlineFare])
values
(150,1,900,600,' ',0.0),
(151,2,901,601,'Alaska Airlines',200.0),
(152,3,902,602,' ',0.0),
(153,4,903,603,'American Eagel ',150.0),
(154,5,904,604,' ',0.0)
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:22:49.3917441-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (65) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 40 Ch 28 OVR
Type here to search
2:22 AM
5/2/2020


SQLQuery13.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (52))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
System Databases
Database Snapshots
AP
book
Books
Examples
MyGuitarShop
ProductOrders
University
University2
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery13.sql - D...tDatabase (sa (52))*
Interview.sql - DES...ntDatabase (sa (64))
JobApplication.sql...ntDatabase (sa (62))
Insert Into ComplaintHandling
([ComplaintID],[CandidateID],[InterviewerID],[ComplaintDescription])
values
(50,900,500,' '),
(51,901,501,'Time Not Sufficient'),
(52,902,502,' '),
(53,903,503,' Interviewer was Biased'),
(54,904,504,' ')
100 %
Messages
(5 rows affected)
Completion time: 2020-05-02T02:22:14.7952299-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (52) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 8 Col 33 Ch 21 OVR
Type here to search
2:22 AM
5/2/2020

4.2 VIEW

Views in SQL are virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition. We create a view to show current status of the candidate interview . For this we have view that joins CandidateDetails table and JobApplication table and return the candidate name and his/her job status.

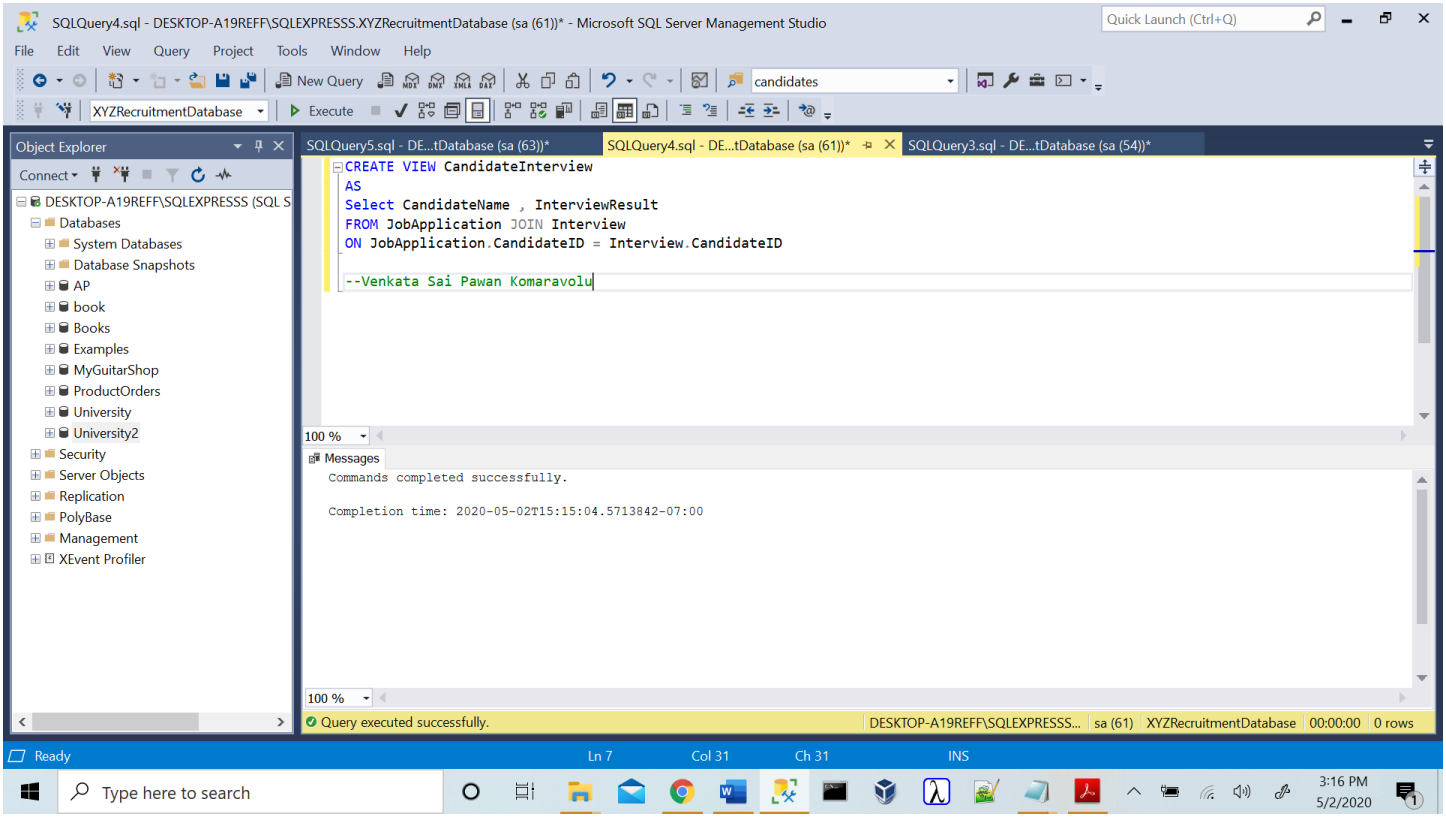
1st view : CREATE VIEW CandidateInterview

AS

Select CandidateName , InterviewResult

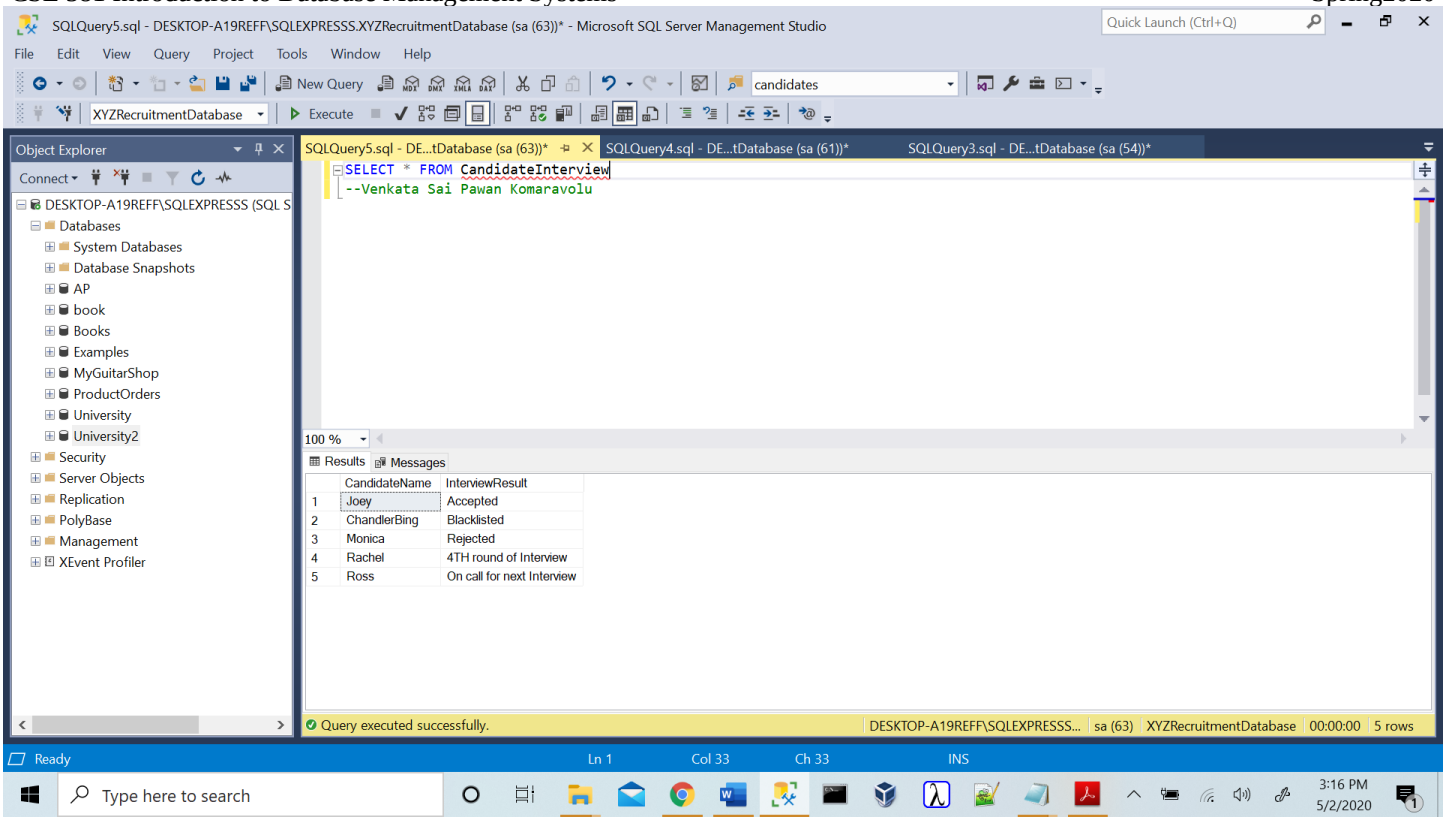
FROM JobApplication JOIN Interview

ON JobApplication.CandidateID = Interview.CandidateID



We now write select statement for selecting everything from the view

Select * from CandidateInterview



We create another view to get the interview location of the candidate which is onsite and all details of the travel i.e. Airline details, Hotels details and car rentals details are combined into a table using views. We create view named CandidateInterviewLocation and get CandidateName, CandidateID , Airline details, hotel details and car rental details using join and on common CandidateID.

2nd view: CREATE VIEW CandidateInterviewLocation

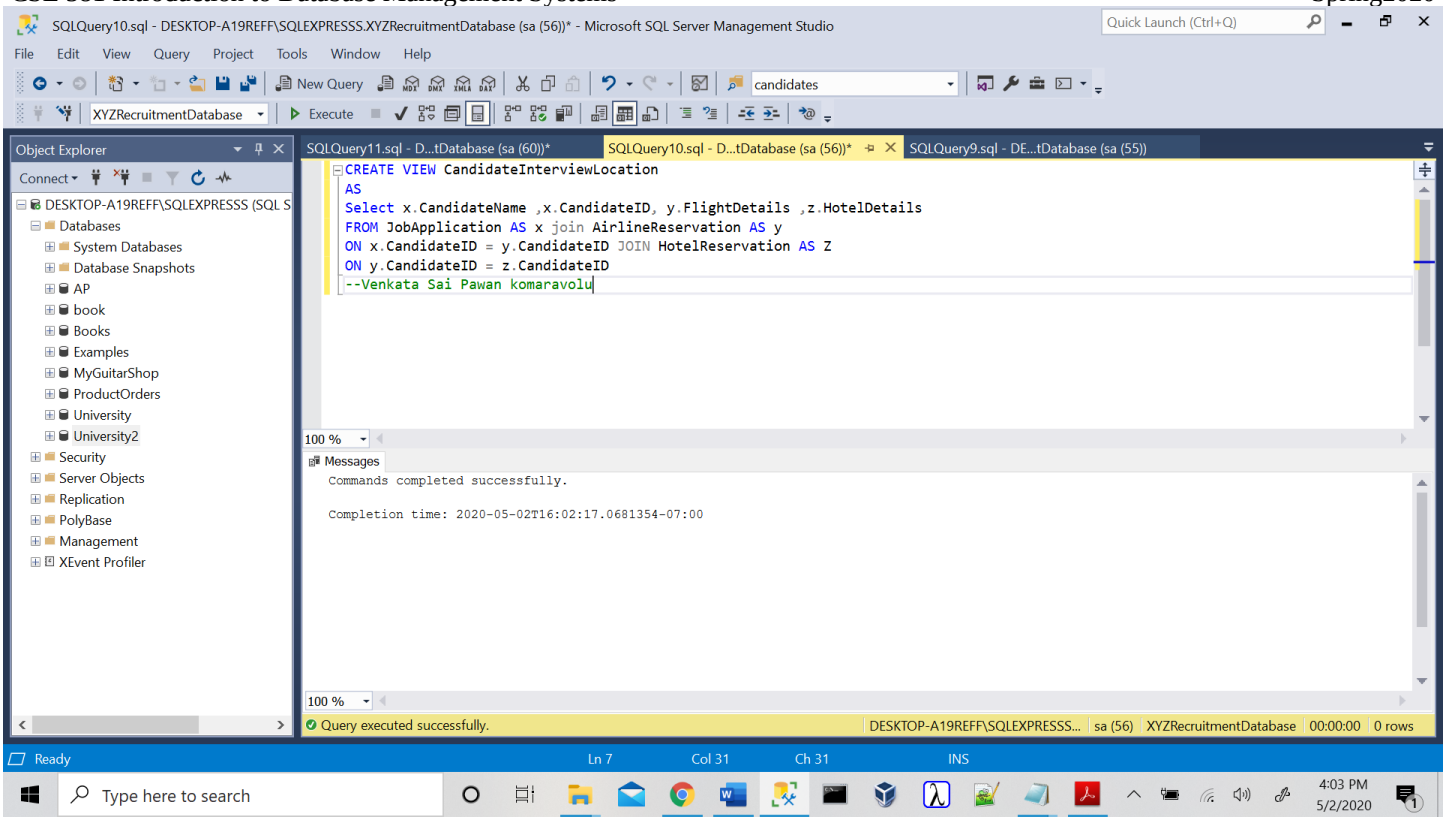
AS

Select x.CandidateName ,x.CandidateID, y.FlightDetails ,z.HotelDetails

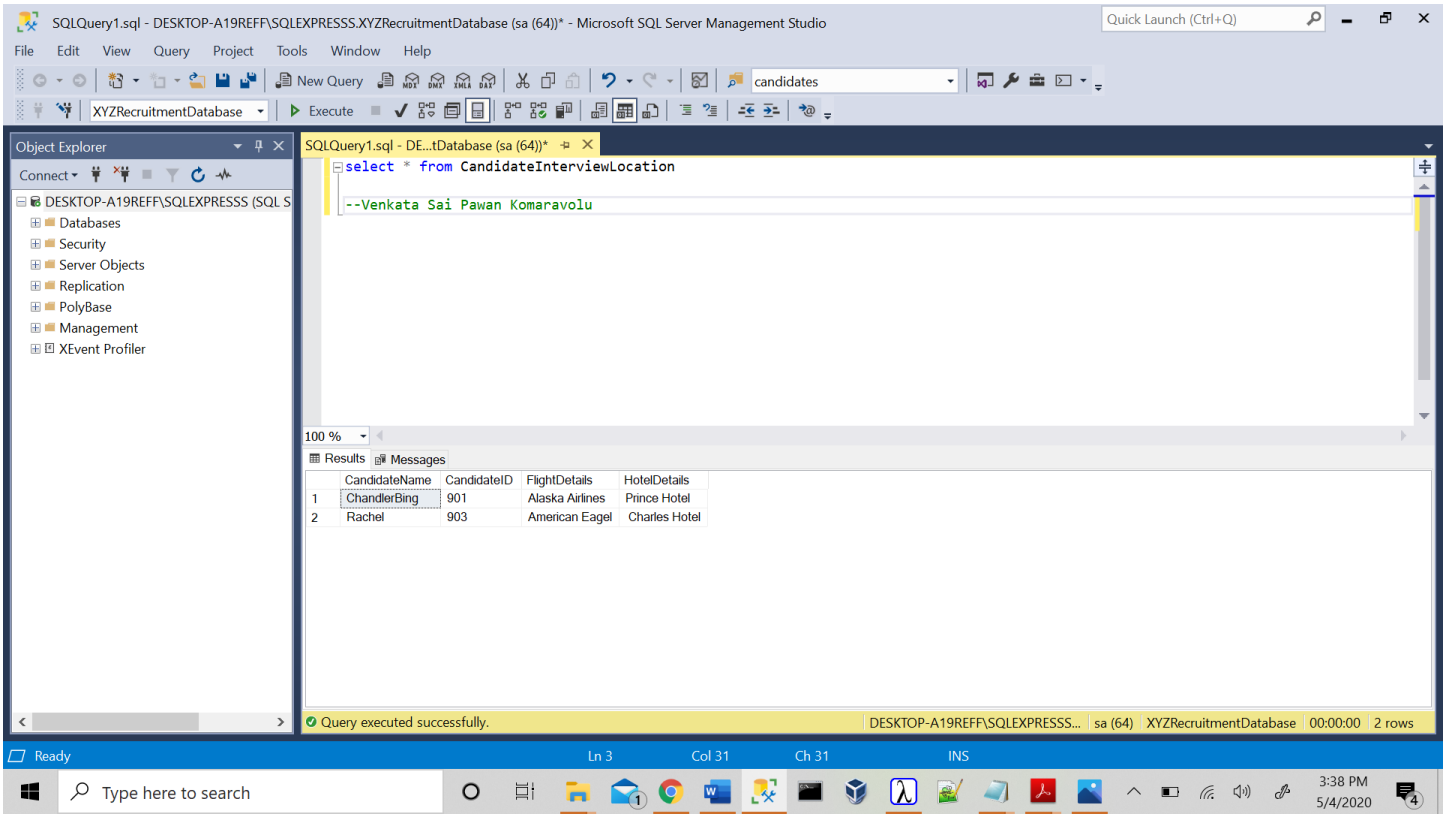
FROM JobApplication AS x join AirlineReservation AS y

ON x.CandidateID = y.CandidateID JOIN HotelReservation AS Z

ON y.CandidateID = z.CandidateID



We display results of this view



We create another view to get the details of the reimbursement amount from each of the tables: Airlines, hotel and rental on common CadidateID.

3rd view: view: CREATE VIEW ReiumbursementAmount

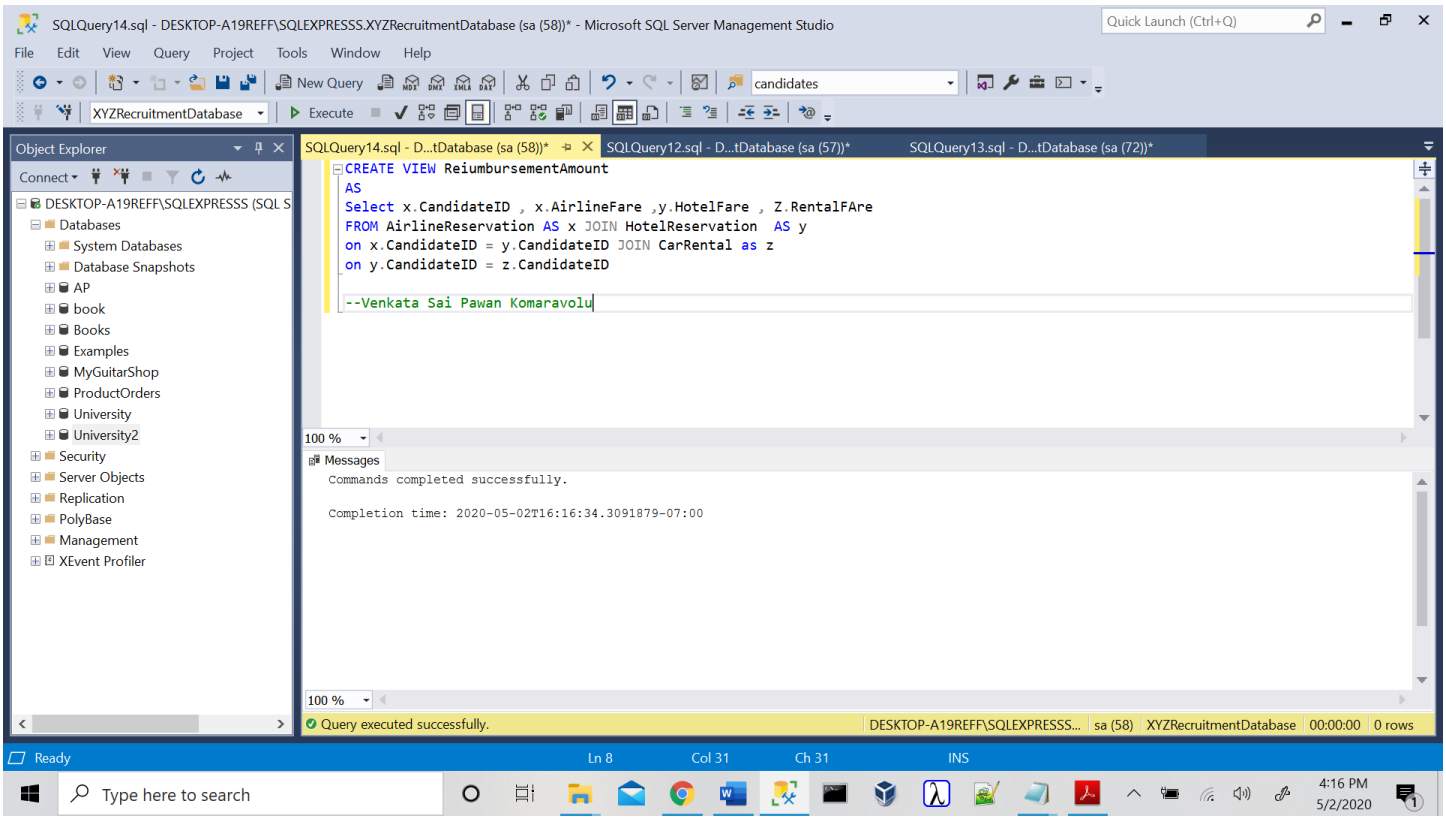
AS

Select x.CandidateID , x.AirlineFare ,y.HotelFare , Z.RentalFAre

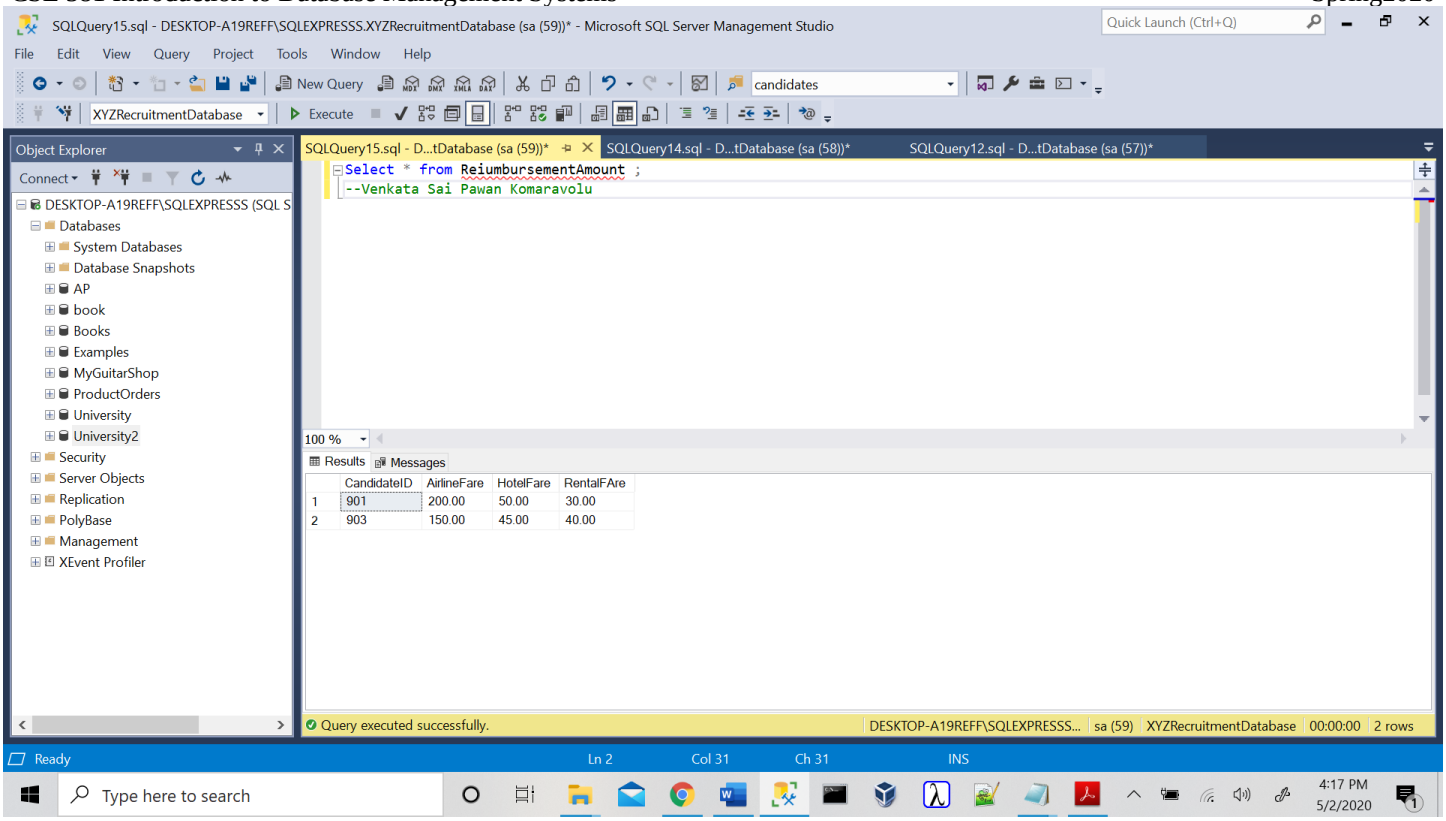
FROM AirlineReservation AS x JOIN HotelReservation AS y

on x.CandidateID = y.CandidateID JOIN CarRental as z

on y.CandidateID = z.CandidateID



We now get the result of candidateID and the fares which need to be reiumbursed



We create another view for getting the complaint status for candidate and interviewer with complaint description. We get the CaniddiateID, InterviewerID and the complaint description.

4th view : CREATE VIEW ComplaintCandidate

AS

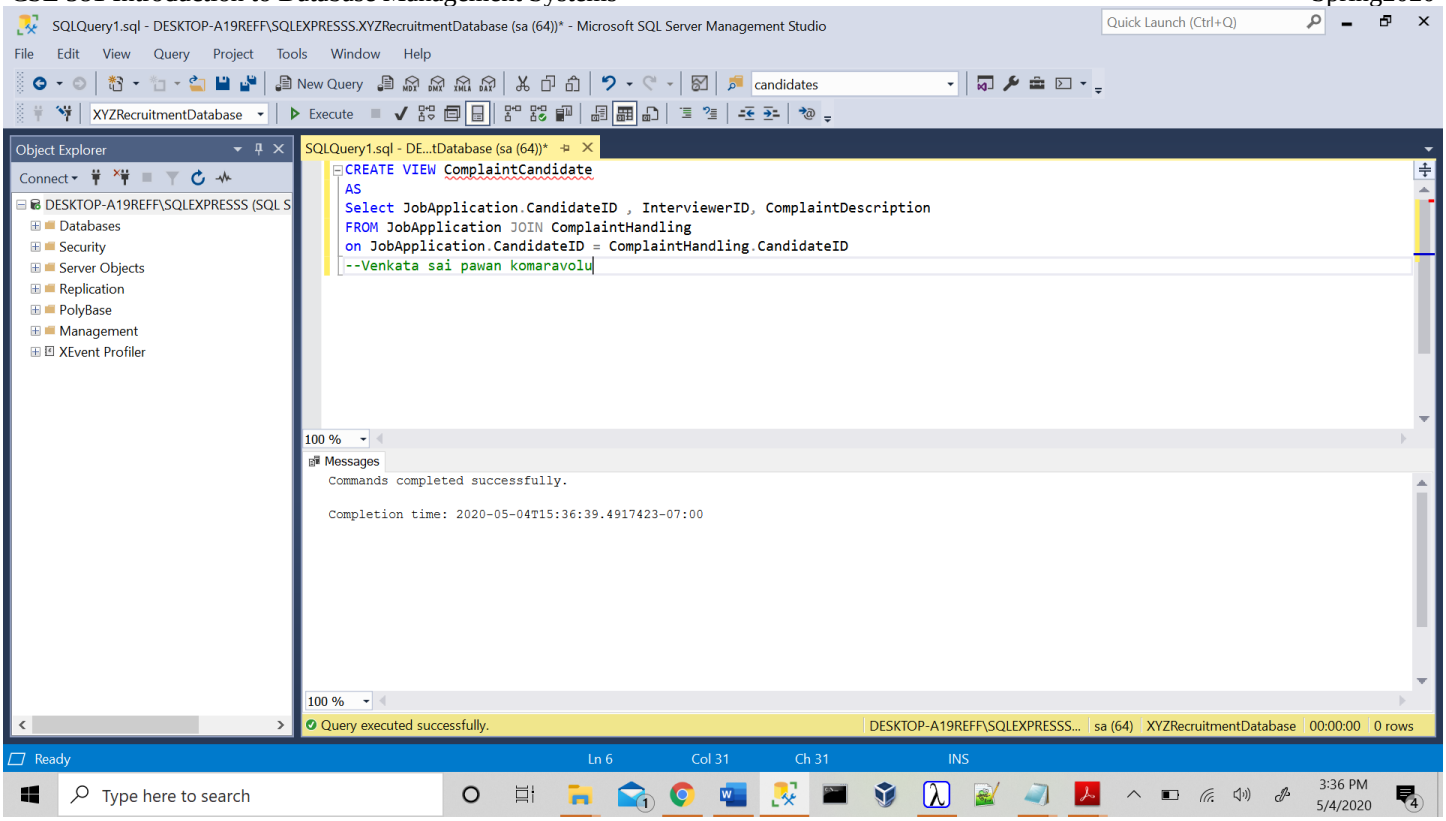
Select JobApplication.CandidateID , InterviewerID, ComplaintDescription

FROM JobApplication JOIN ComplaintHandling

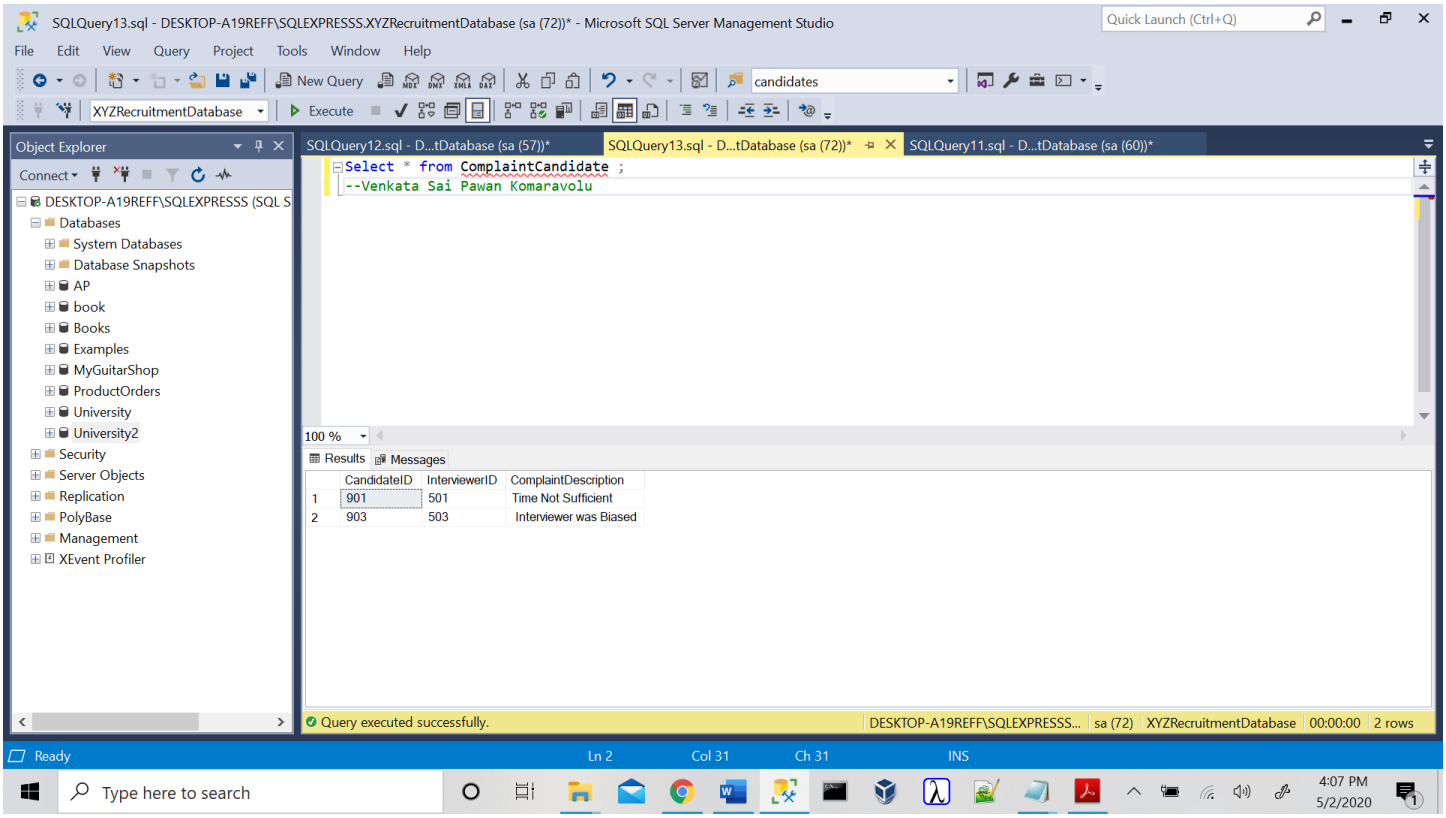
on JobApplication.CandidateID = ComplaintHandling.CandidateID

Select * from ComplaintCandidate ;

Select * from CandidateInterviewLocation



We select everything from the ComplaintCandidate view.



We create another view to get the contact details of the candidate for this we join ContactDetails table and CandidateDetails table on common CandidateDetailsID.

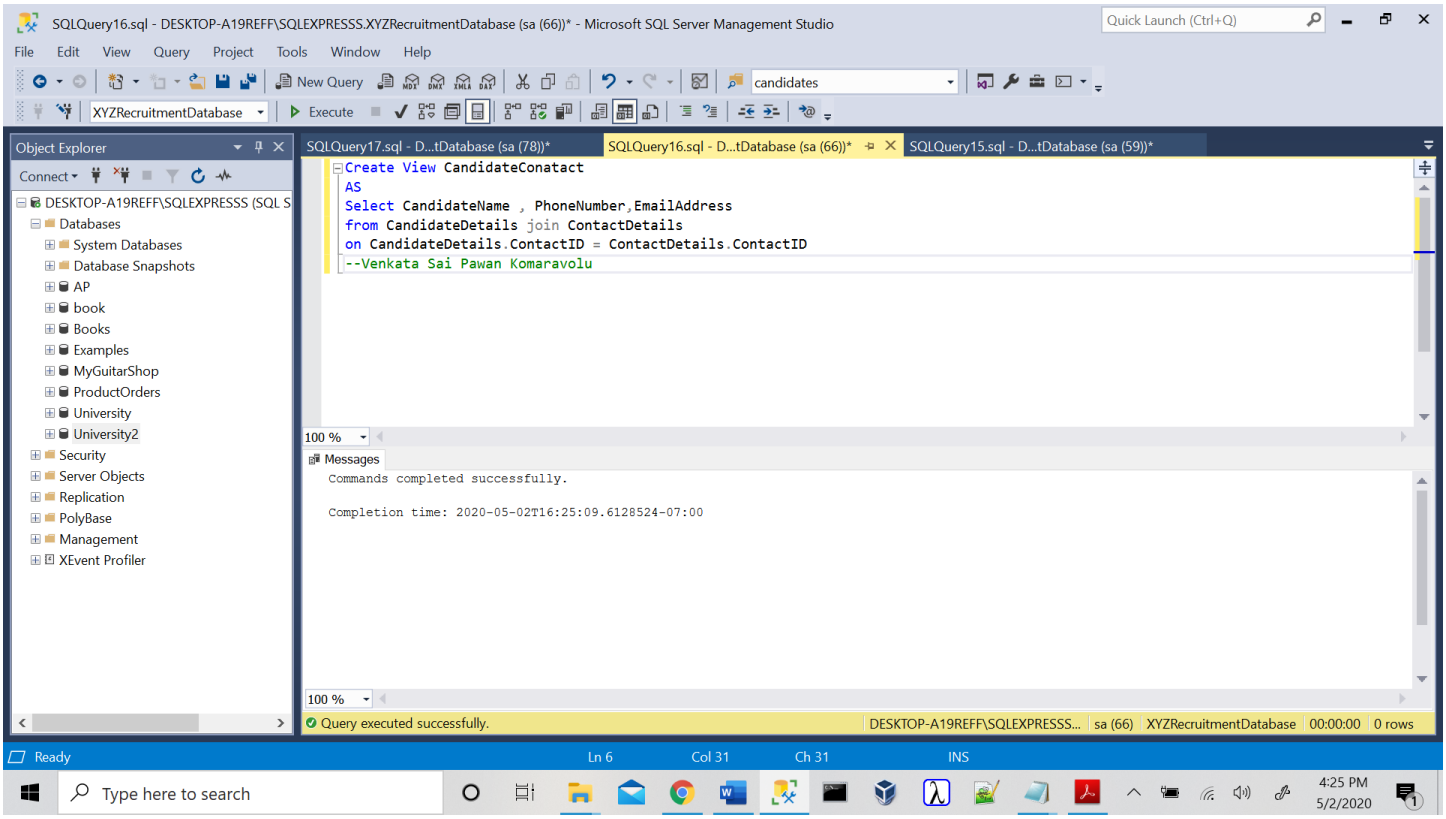
5th view: Create View CandidateConatact

AS

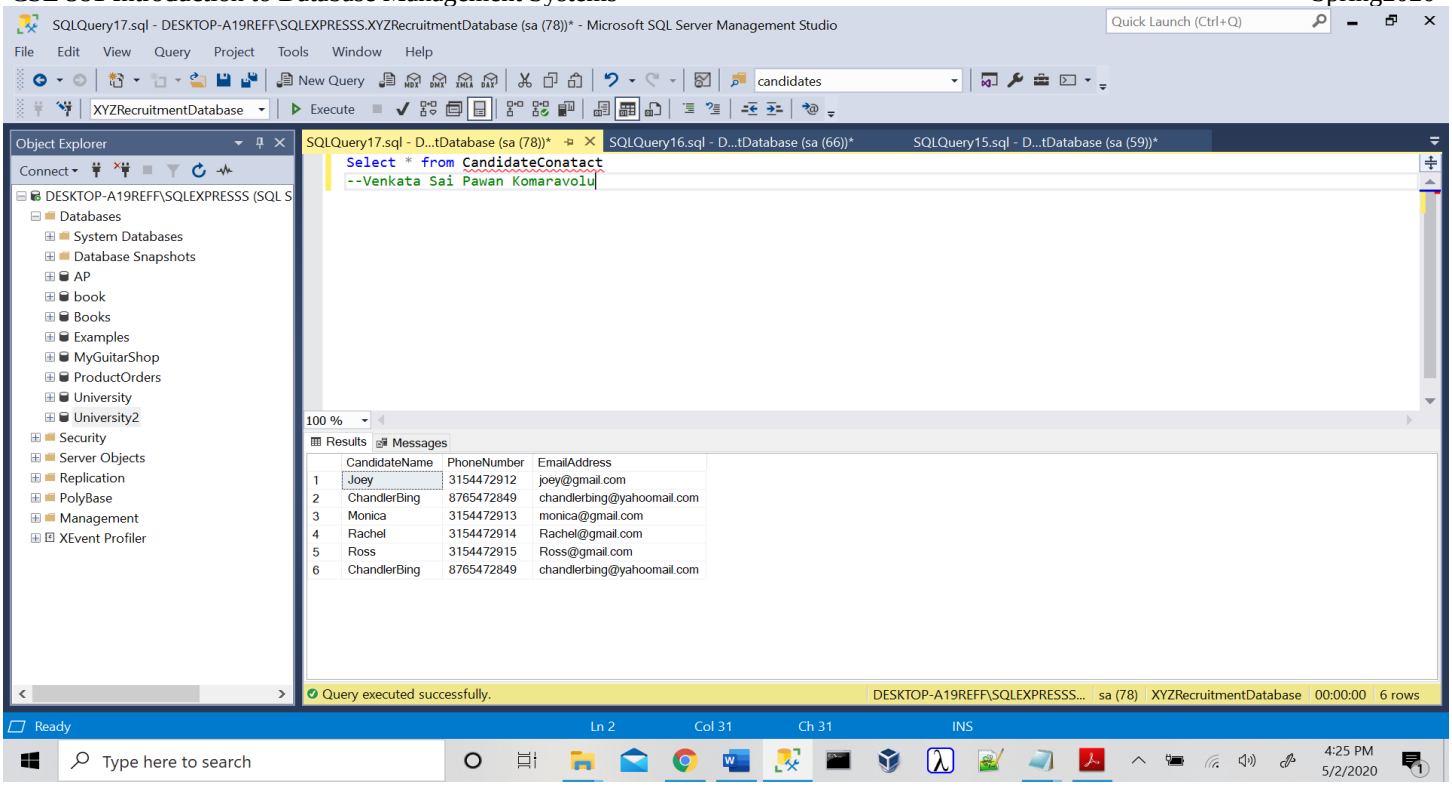
Select CandidateName , PhoneNumber,EmailAddress

from CandidateDetails join ContactDetails

on CandidateDetails.ContactID = ContactDetails.ContactID



We select everything from CandidateContact view



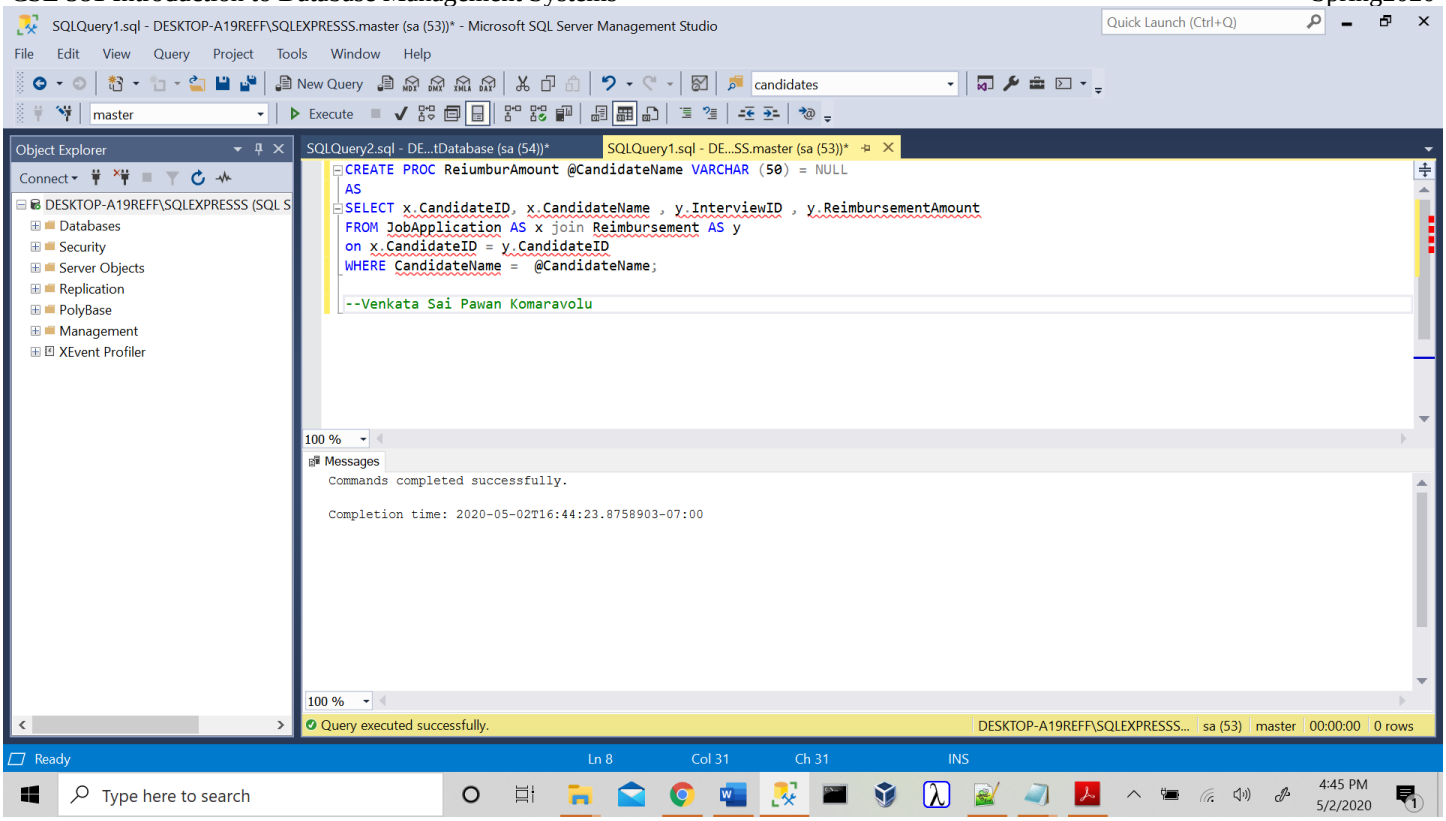
4.3 STORED PROCEDURE

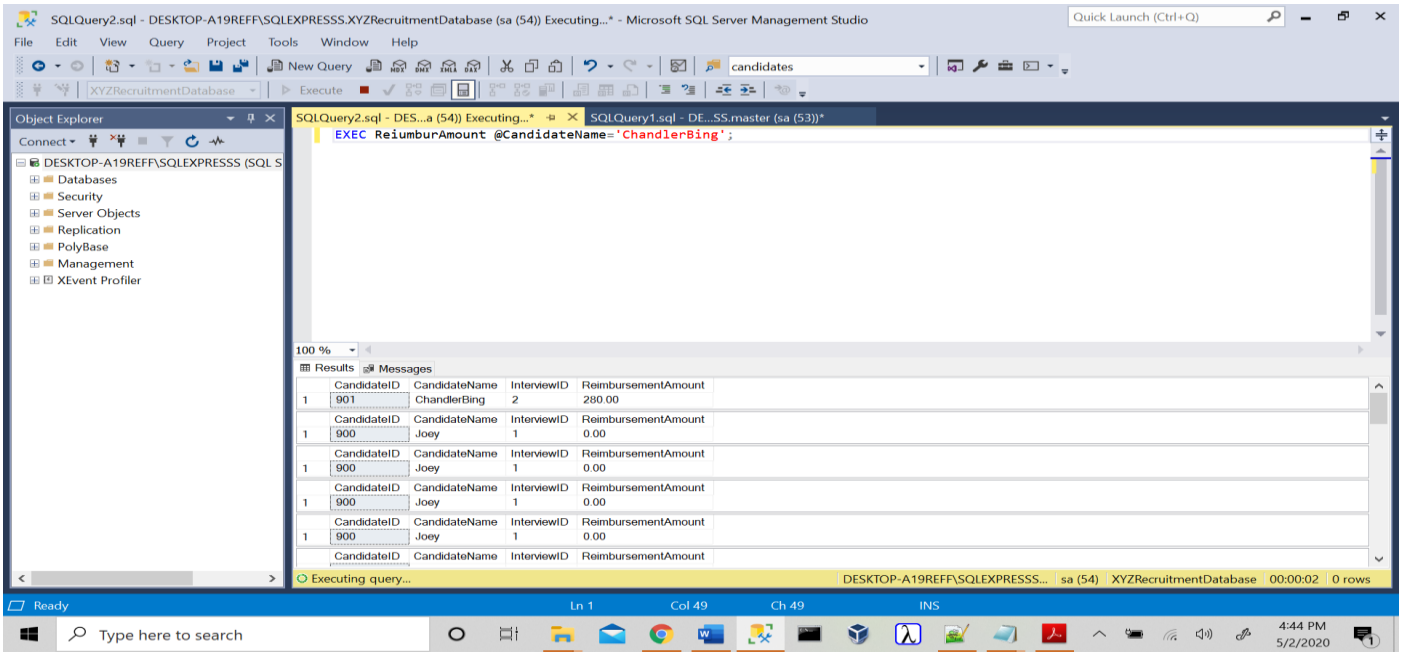
A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it. You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed We create a stored procedure to get the reimbursed amount for a candidate. If we give the procedure name and name of the candidate, we get the reimbursed amount for that candidate.

1st Procedure : CREATE PROC ReiumbursedAmount @CandidateName VARCHAR (50) = NULL
AS
SELECT x.CandidateID, x.CandidateName , y.InterviewID , y.ReimbursementAmount
FROM Candidate AS x join Reimbursement AS y
on x.CandidateID = y.CandidateID
WHERE CandidateName = @CandidateName;

We execute the procedure using the query

EXEC ReiumbursementAmount @ CandidateName = ‘ChandlerBing’ will give results of the reiumbursement amount for the CandidateName ChandlerBing.

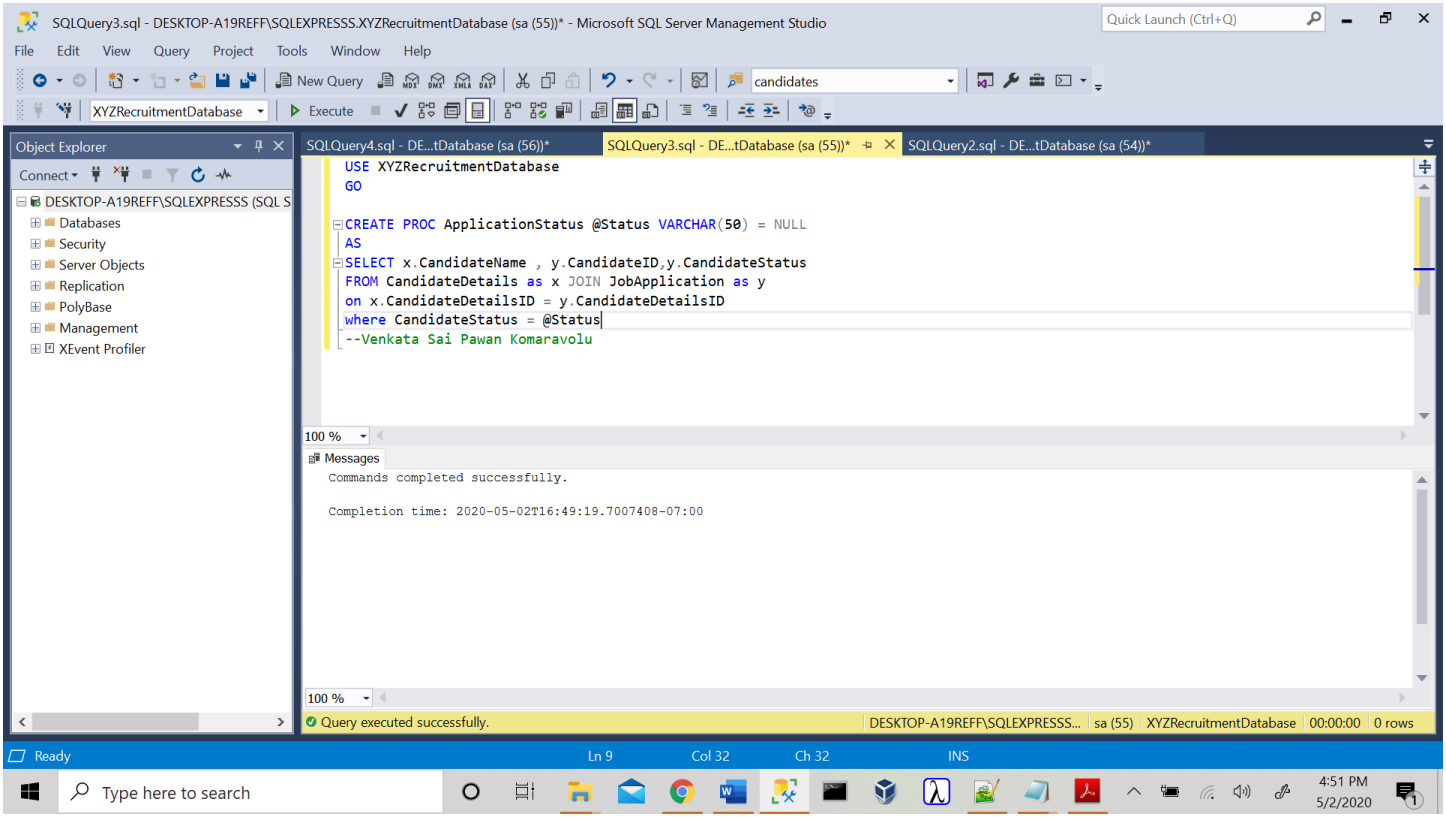


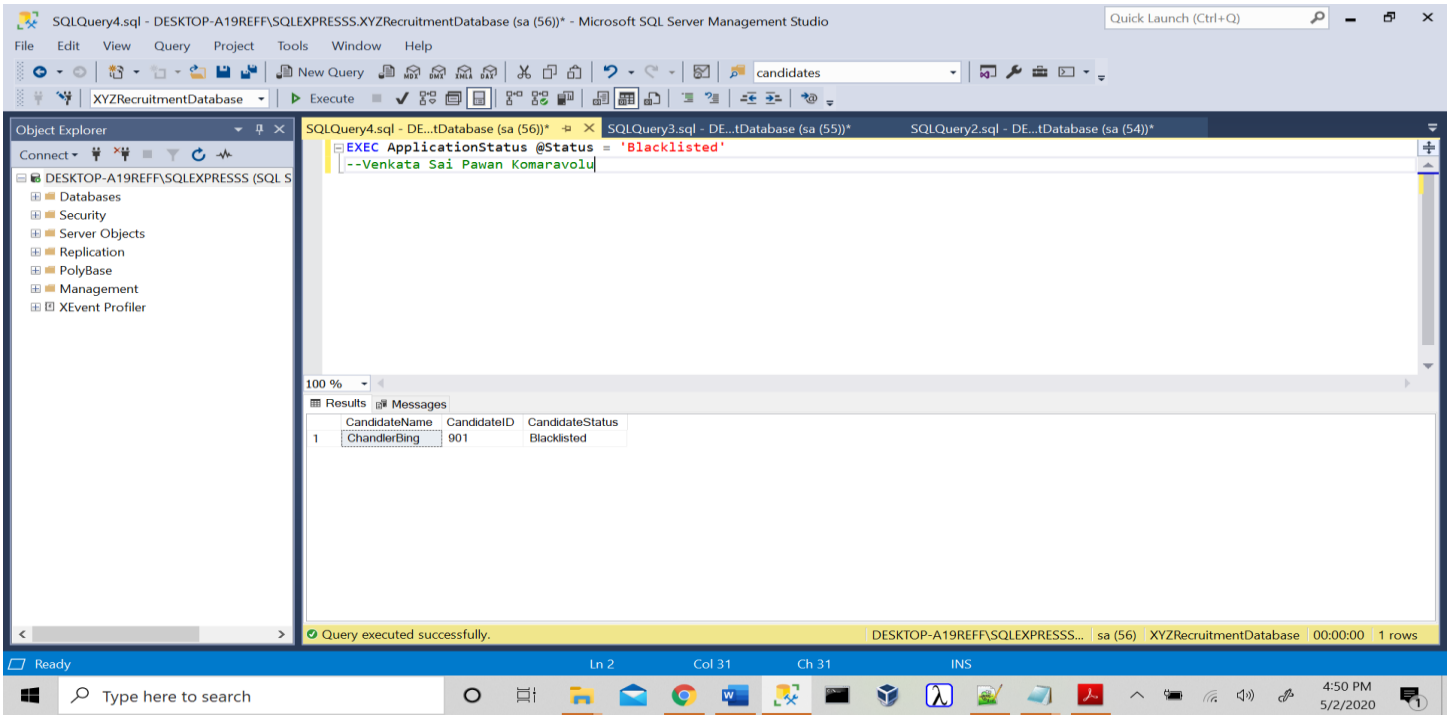


We create another procedure for the application status. If have application status in stored procedure and when we execute this procedure with candidate name we get the status of the candidate

2nd procedure: CREATE PROC [dbo].[ApplicationStatus] @Status VARCHAR(50) = NULL
AS
SELECT x.CandidateName , y.CandidateID,y.CandidateStatus
FROM CandidateDetails as x JOIN JobApplication as y

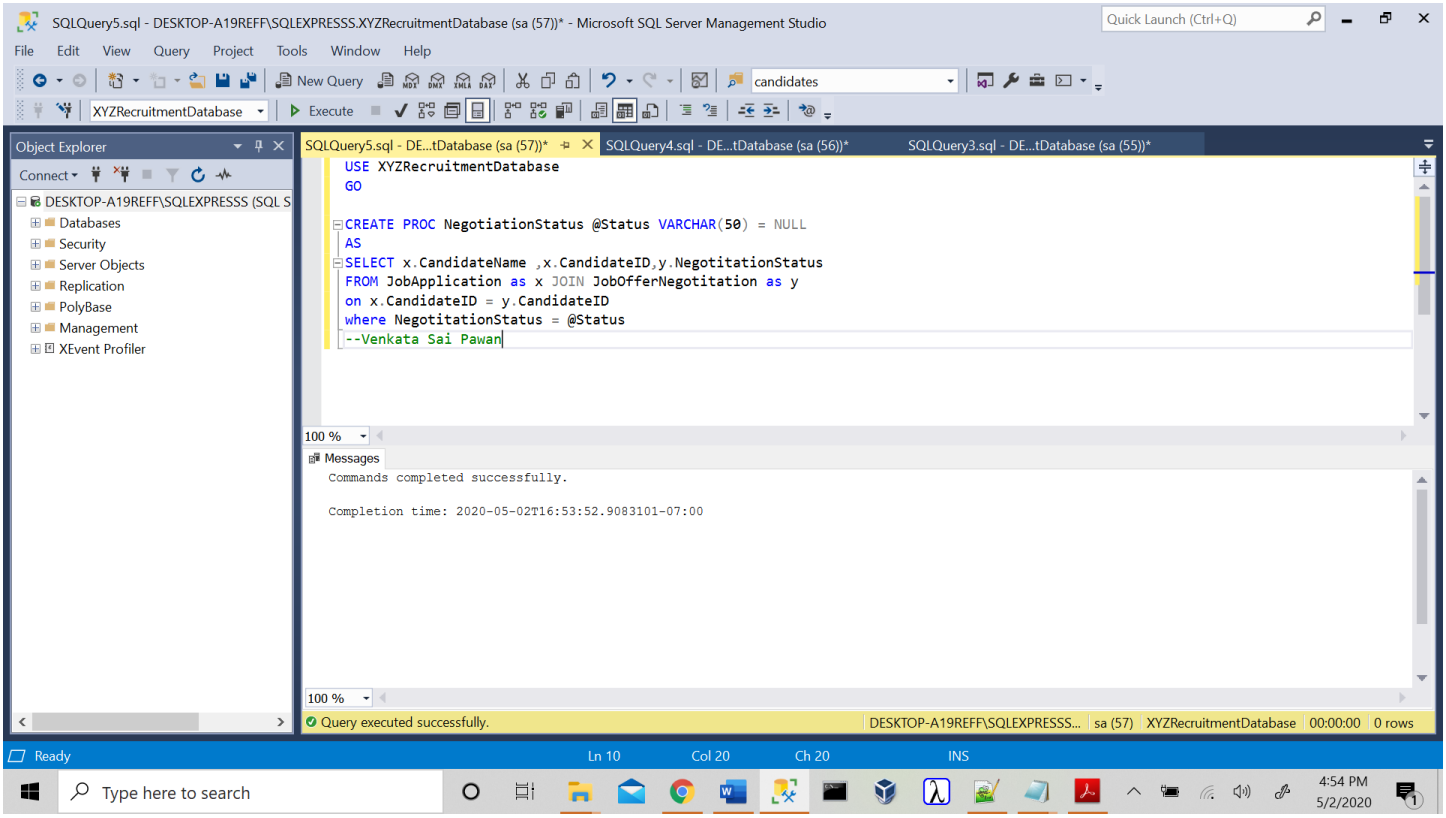
on x.CandidateDetailsID = y.CandidateDetailsID
where CandidateStatus = @Status

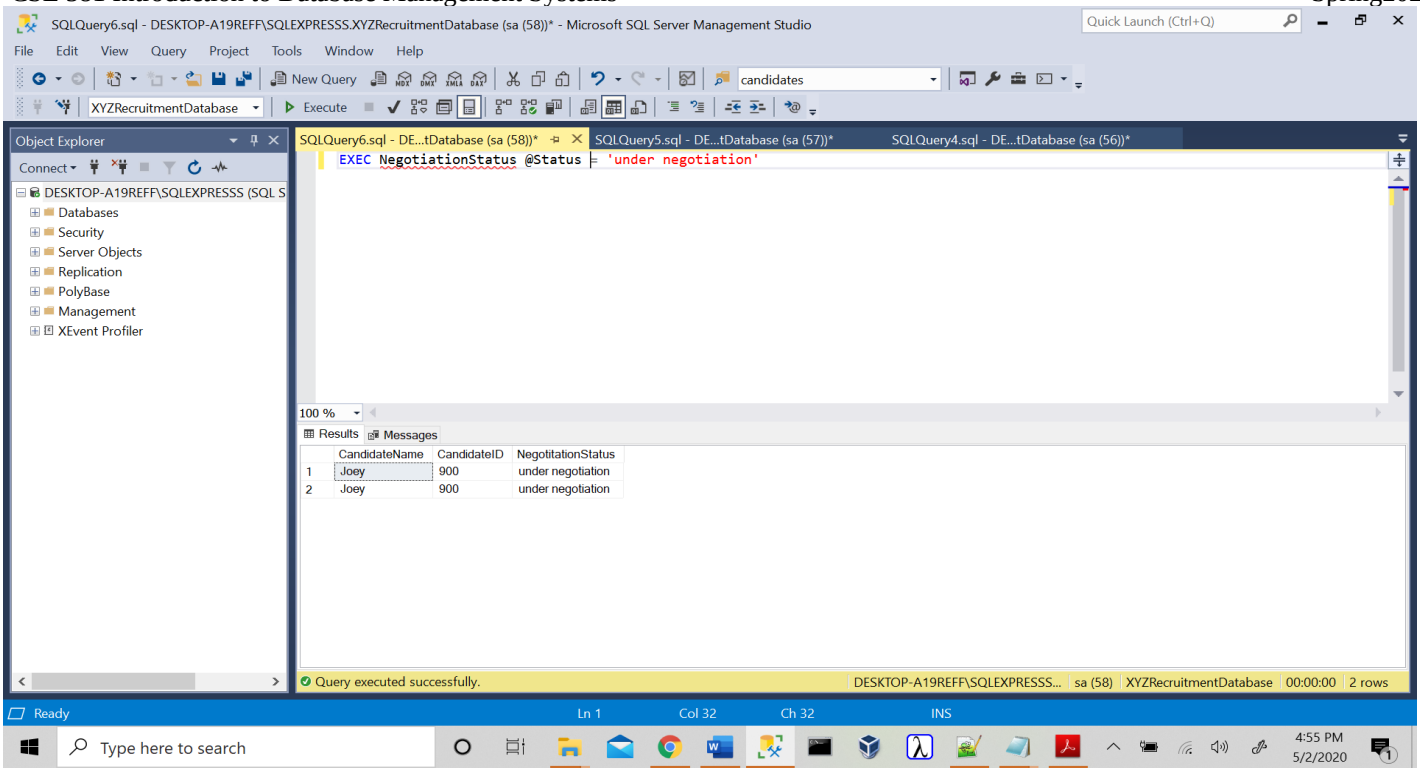




We create another procedure to get the status of the negotiation which is still under negotiation

3rd procedure: CREATE PROC [dbo].[NegotiationStatus] @Status VARCHAR(50) = NULL
AS
SELECT x.CandidateName ,x.CandidateID,y.NegotitationStatus
FROM JobApplication as x JOIN JobOfferNegotitation as y
on x.CandidateID = y.CandidateID
where NegotitationStatus = @Status
GO



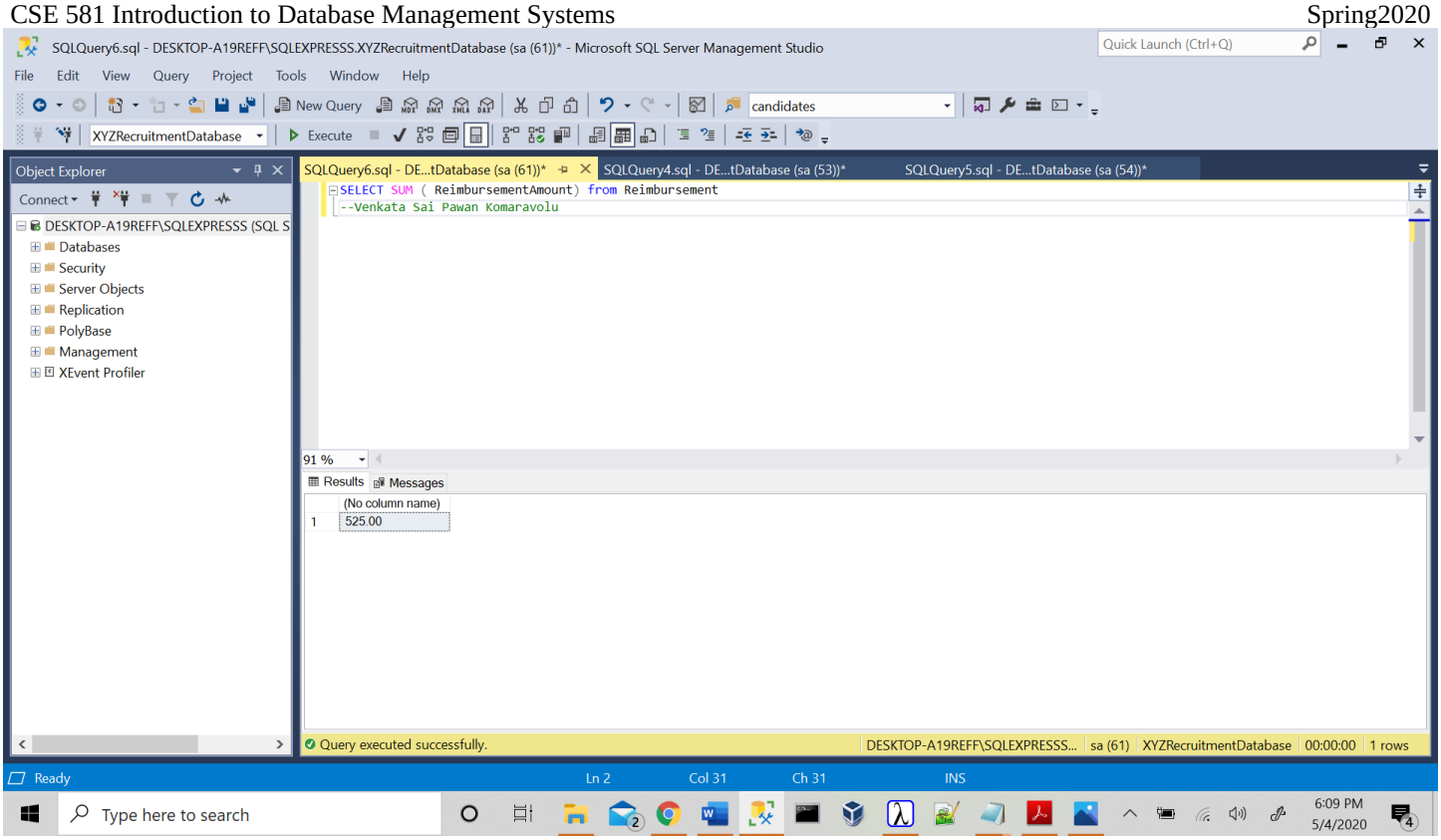


4.4 FUNCTIONS

We have 2 types of functions 1. Built in functions such as aggregate functions and 2. UDF : User defined function. Below are the functions used in this project

SELECT SUM (ReimbursementAmount) from Reimbursement

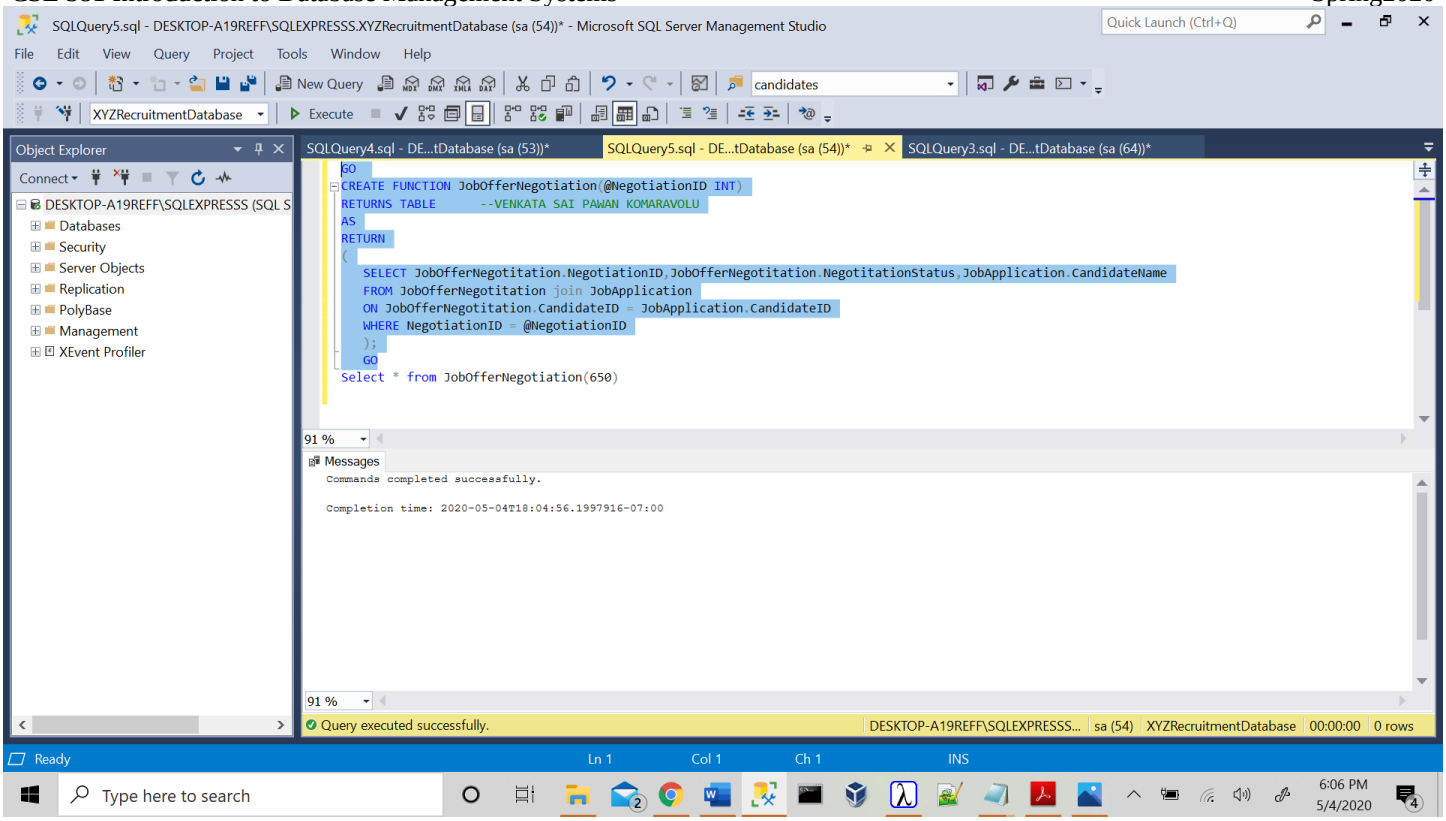
We select the sum of the reimbursement amount in the reimbursement table.

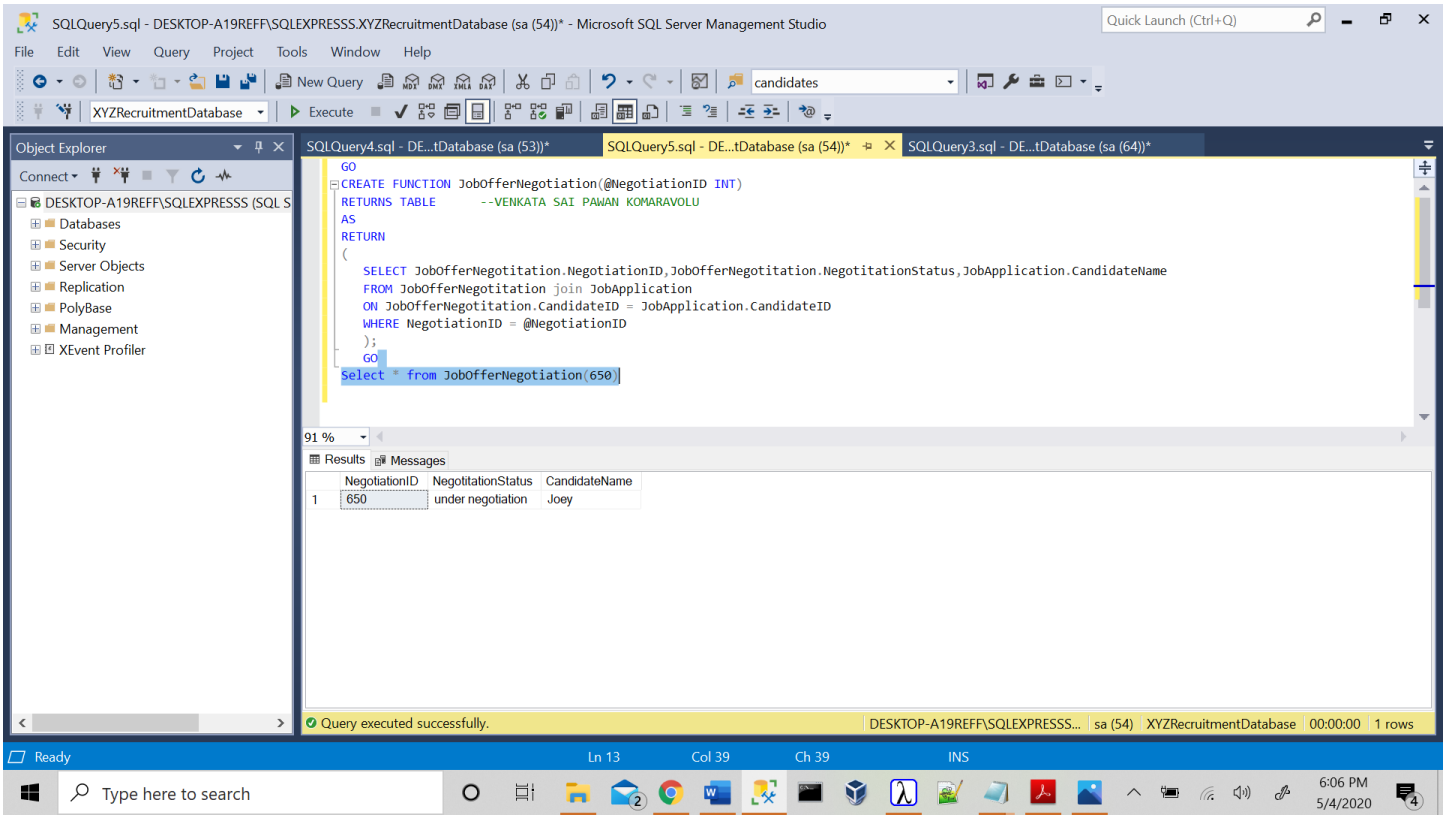


2nd Function :

```
GO
CREATE FUNCTION JobOfferNegotiation(@NegotiationID INT)
RETURNS TABLE
AS
RETURN
(
  SELECT
JobOfferNegotitation.NegotiationID,JobOfferNegotitation.NegotitationStatus,JobApplication.CandidateName
  FROM JobOfferNegotitation join JobApplication
  ON JobOfferNegotitation.CandidateID = JobApplication.CandidateID
  WHERE NegotiationID = @NegotiationID
  );
  GO
Select * from JobOfferNegotiation(650)
```

We create a function named JobOfferNegotiation and in this we select NegotiationID, NegotiationStatus from JobOfferNegotiation table and Candidate name from JobApplication table on Common CandidateID and we have NegoitationID as the parameter and the final result we get is the candidate name and his negotiation status from both the tables and we pass the NegotiationID which is 650 in this case.



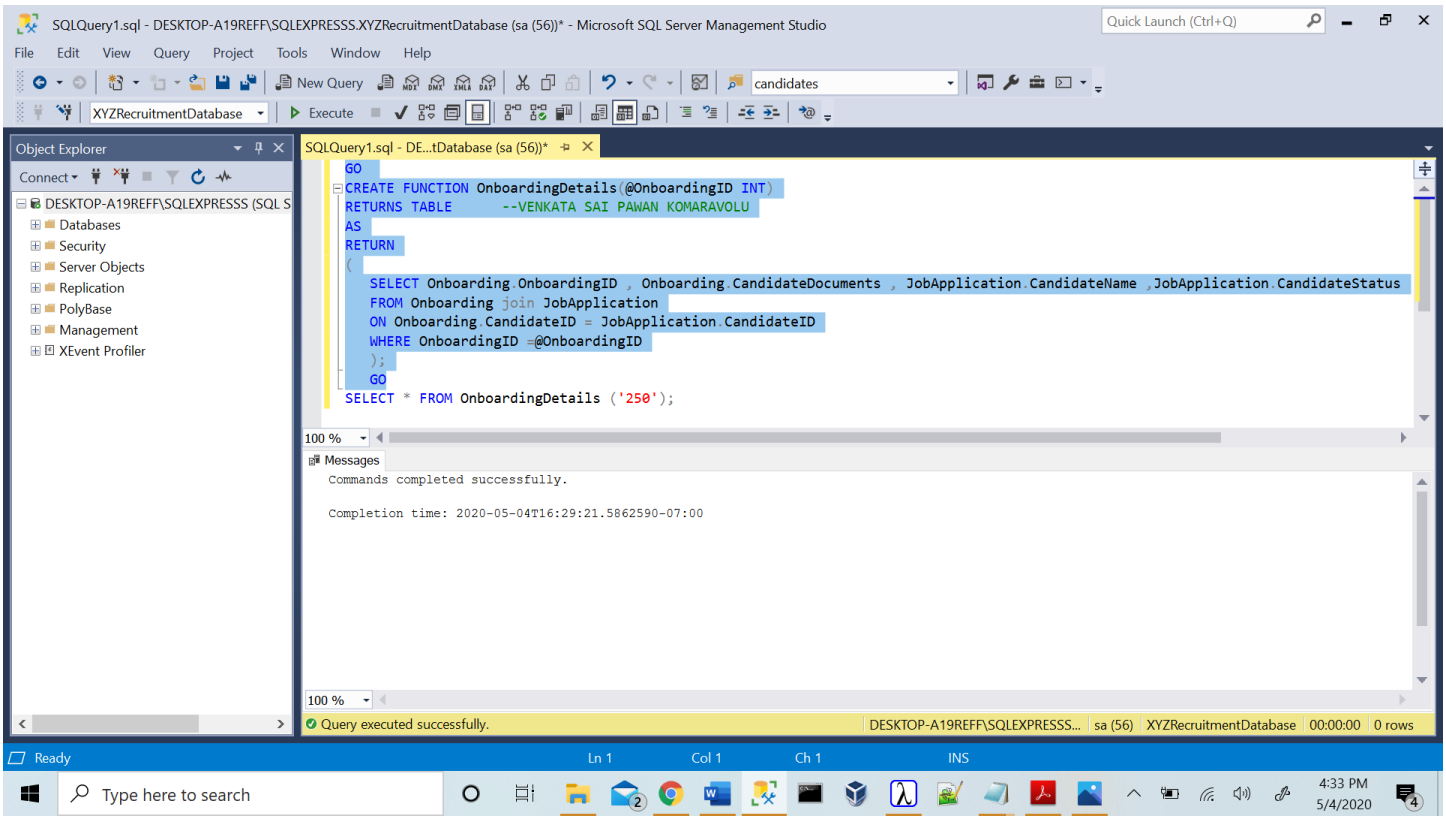


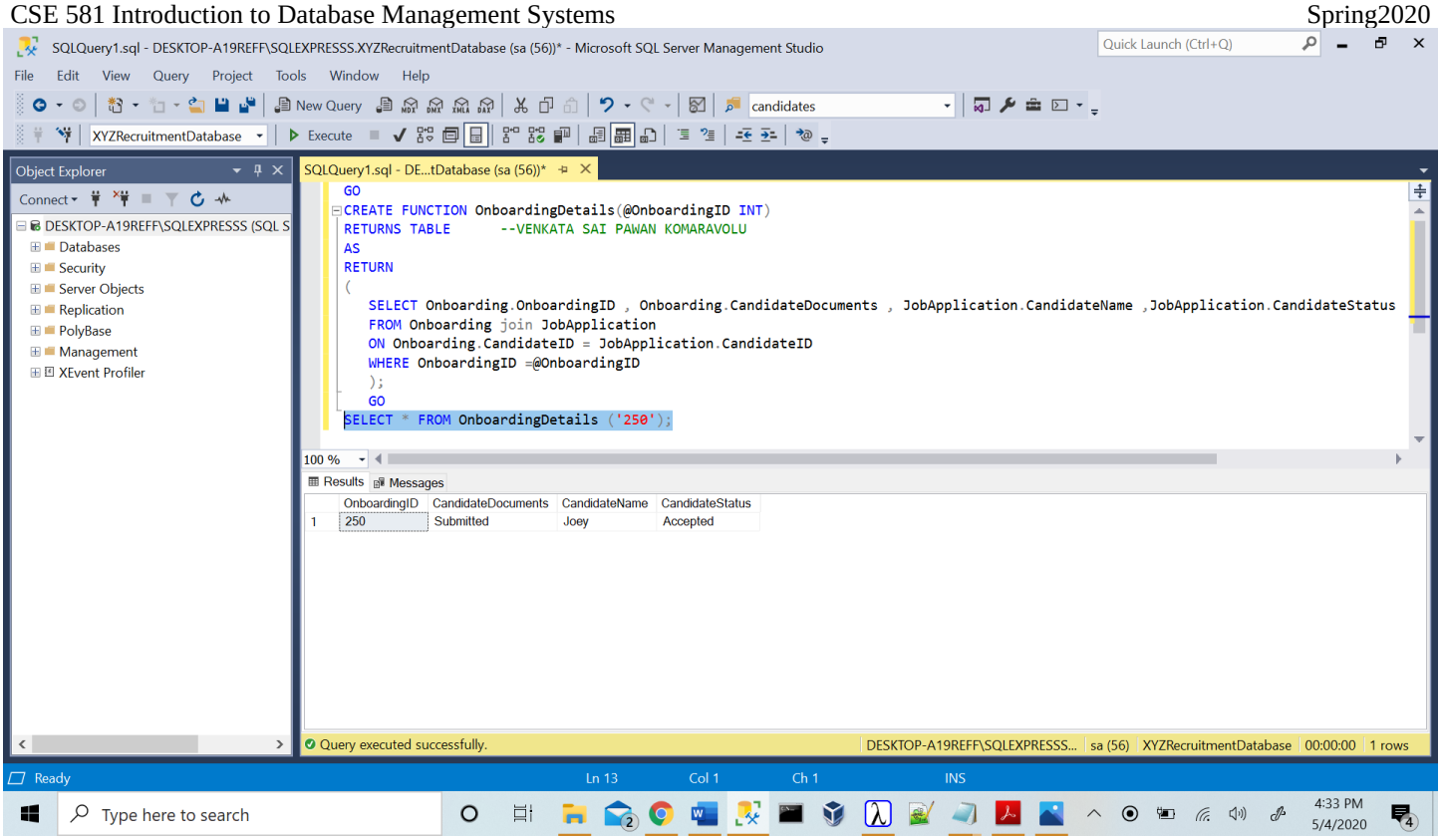
3rd Function :

We now create a function to get the Onboarding status of the candidate along with his/her name and for this we create the function named OnBoardingDetails and Onboarding ID as the object and returns the table where we have selected OnboardingID , CandidateDocuments and Candidatename and CandidateStatus from the join of

Onboarding and JobApplication table on common CandidateID and later we select everything from this function where Onboarding ID is 250.

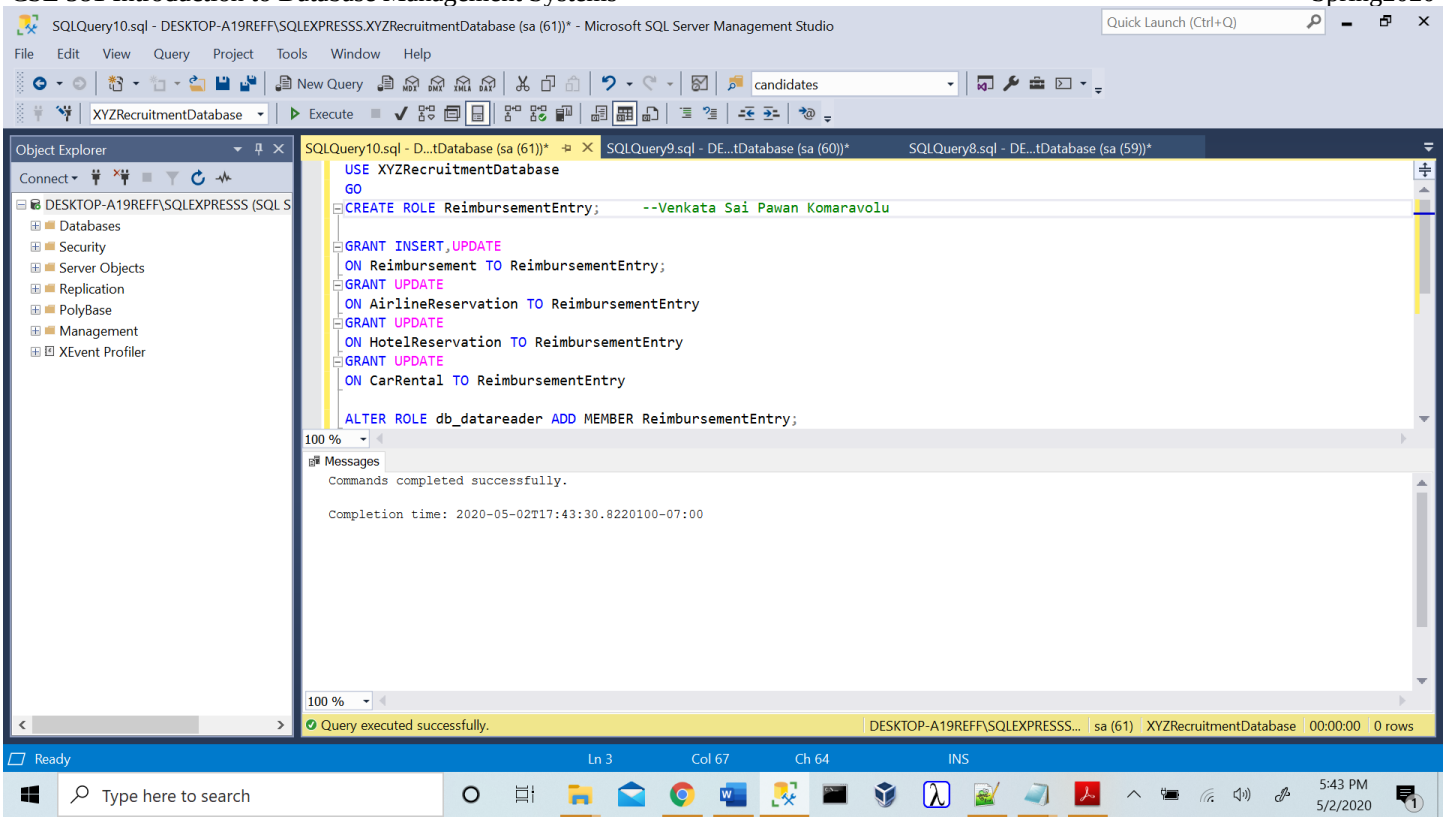
```
GO
CREATE FUNCTION OnboardingDetails(@OnboardingID INT)
RETURNS TABLE
AS
RETURN
(
  SELECT Onboarding.OnboardingID , Onboarding.CandidateDocuments , JobApplication.CandidateName
,JobApplication.CandidateStatus
  FROM Onboarding join JobApplication
  ON Onboarding.CandidateID = JobApplication.CandidateID
  WHERE OnboardingID =@OnboardingID
  );
  GO
```

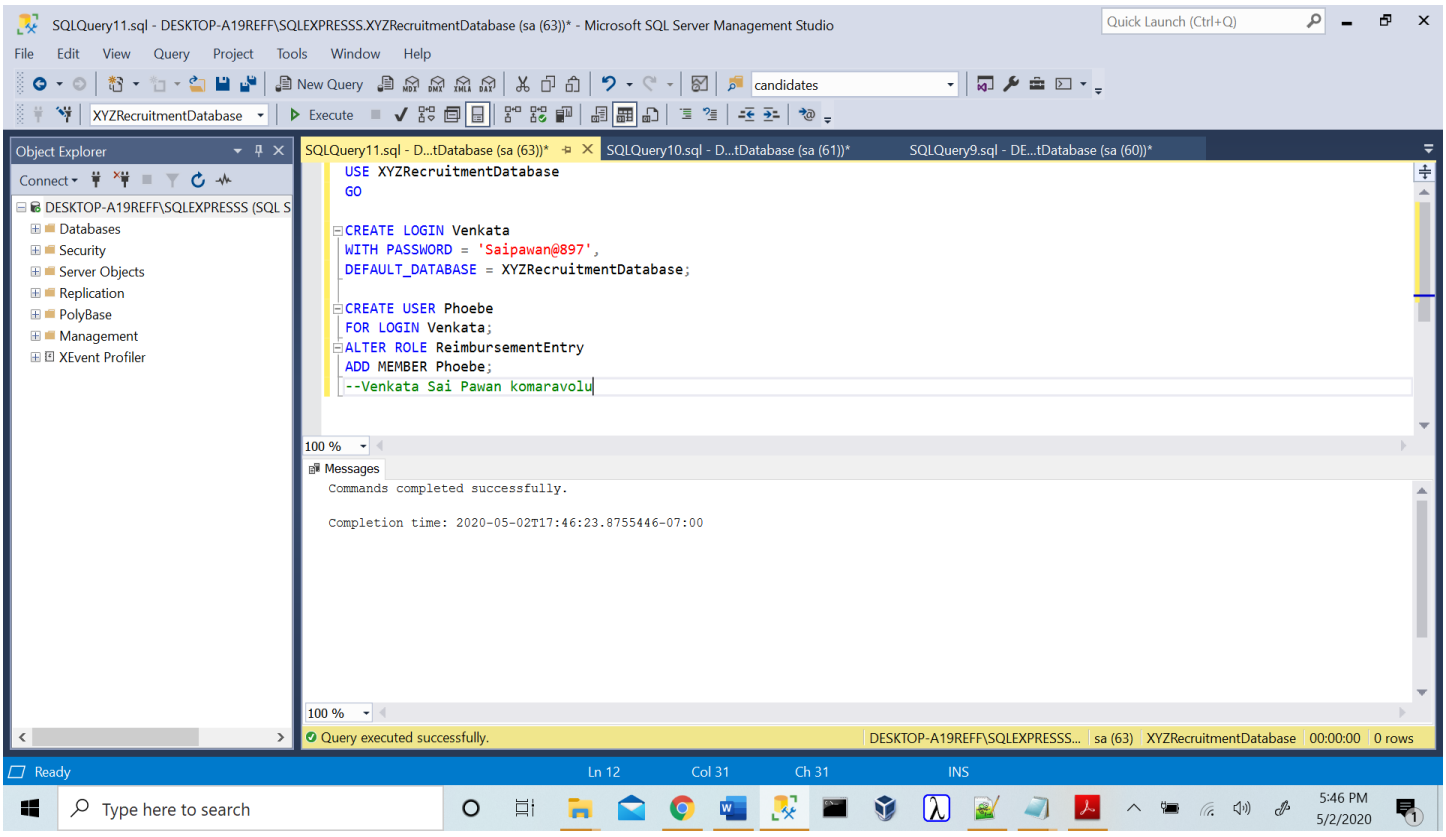




4.5 SCRIPTS

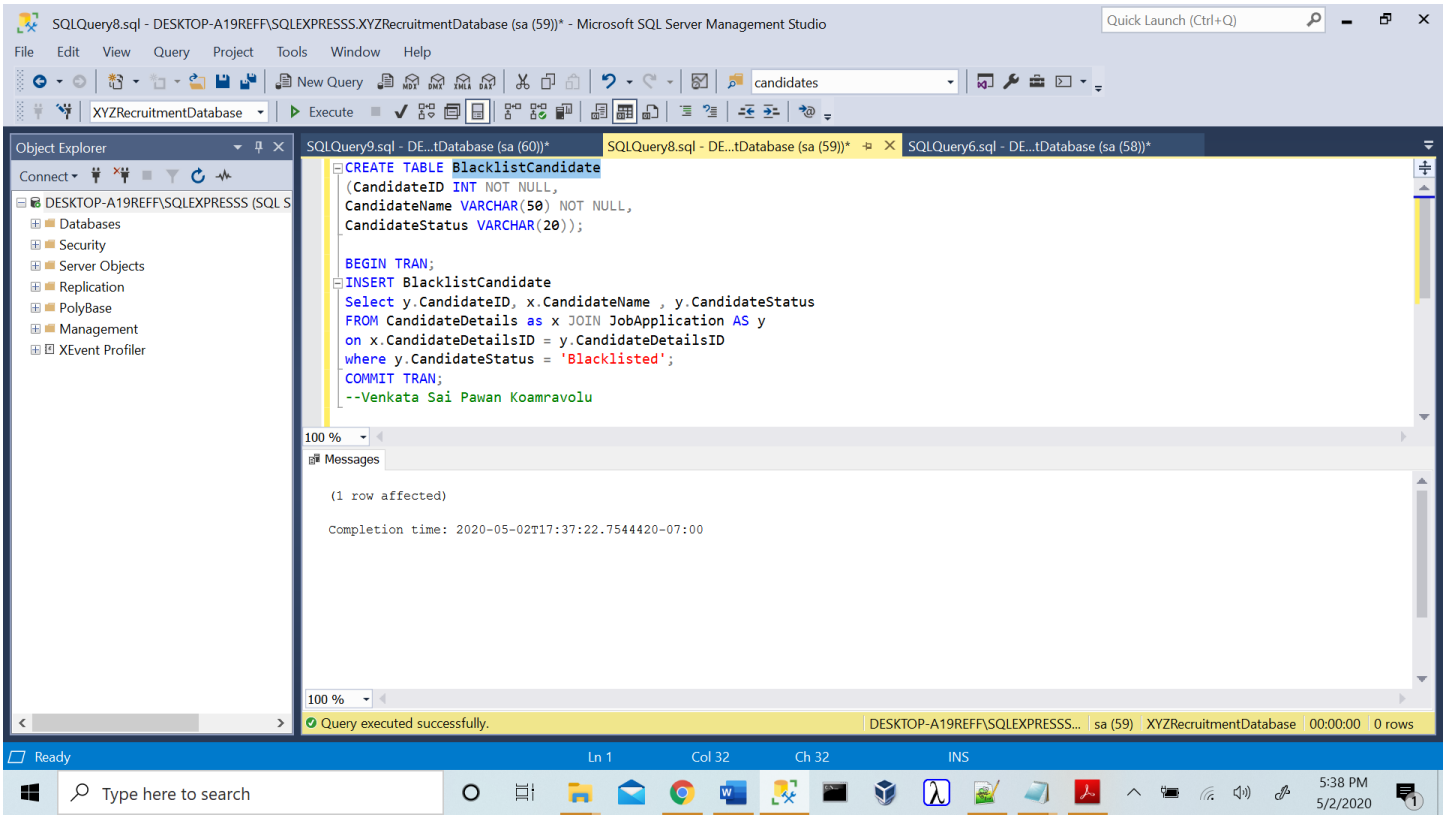
I've created a role ReiumburseEntry and I've granted INSERT and UPDATE on Reiumbursement table to the roleReiumbursementEntry and also to the role we grant UPDATE permission on the tables AirlineReservation, HotelReservation , CarRental to the role ReiumbursementEntry and later alter the role db_datareader and add the member reiumbursementEntry. I have created a Login named 'Venkata' with password as 'Saipawan@897' and with default database as XYZRecruitmentDatabase. Also I have created a User named 'Phoebe' for Login named 'Venkata' I've created a user for this and for the role. I've added the member Phoebe.


SQLQuery10.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (61))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery10.sql - D...tDatabase (sa (61))*
SQLQuery9.sql - DE...tDatabase (sa (60))*
SQLQuery8.sql - DE...tDatabase (sa (59))*
USE XYZRecruitmentDatabase
GO
CREATE ROLE ReimbursementEntry; --Venkata Sai Pawan Komaravolu
GRANT INSERT,UPDATE
ON Reimbursement TO ReimbursementEntry;
GRANT UPDATE
ON AirlineReservation TO ReimbursementEntry
GRANT UPDATE
ON HotelReservation TO ReimbursementEntry
GRANT UPDATE
ON CarRental TO ReimbursementEntry
ALTER ROLE db_datareader ADD MEMBER ReimbursementEntry;
100 %
Messages
Commands completed successfully.
Completion time: 2020-05-02T17:43:30.8220100-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (61) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 3 Col 67 Ch 64 INS
Type here to search
5:43 PM
5/2/2020

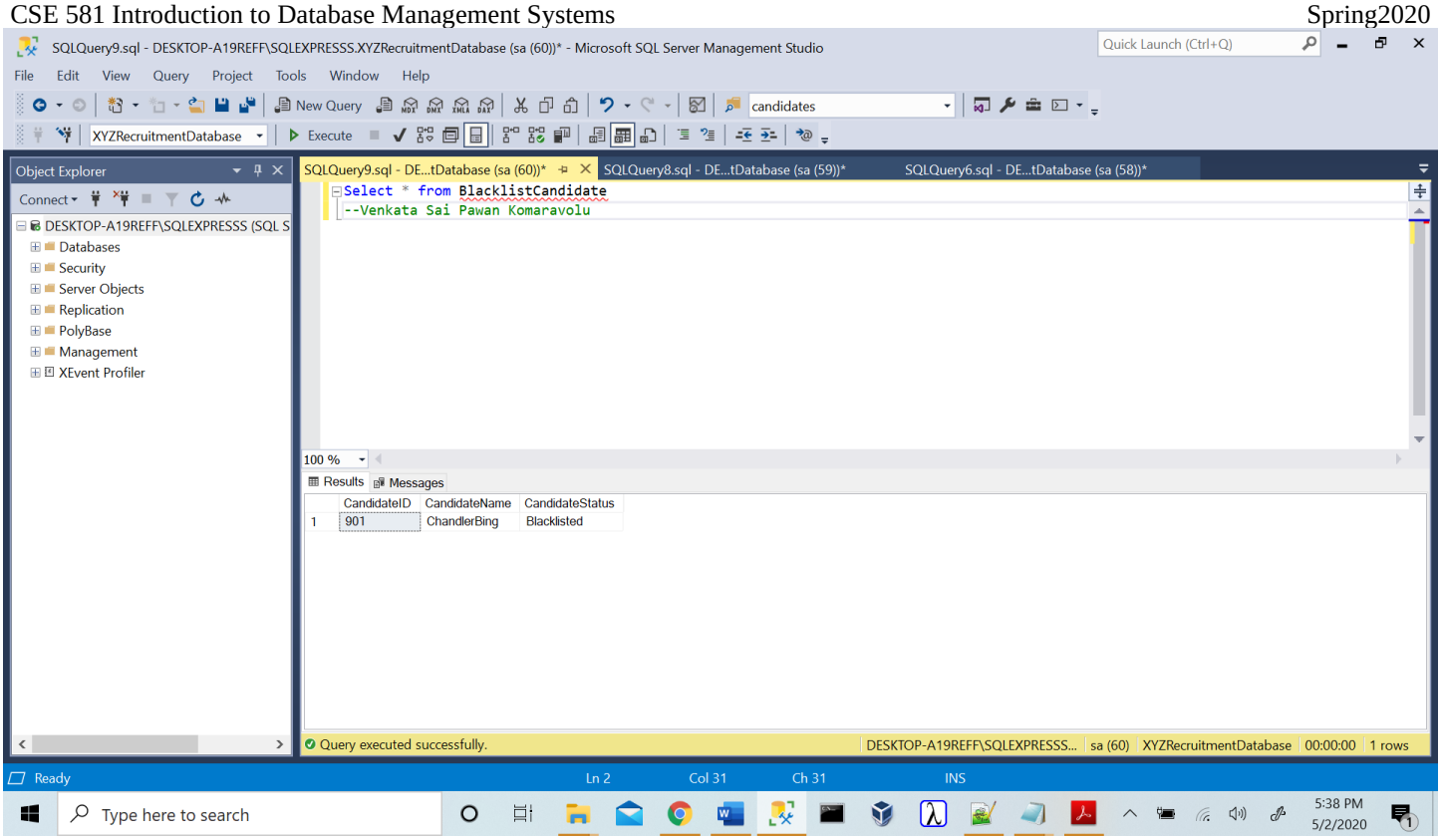

SQLQuery11.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (63))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery11.sql - D...tDatabase (sa (63))*
SQLQuery10.sql - D...tDatabase (sa (61))*
SQLQuery9.sql - DE...tDatabase (sa (60))*
USE XYZRecruitmentDatabase
GO
CREATE LOGIN Venkata
WITH PASSWORD = 'Saipawan@897',
DEFAULT_DATABASE = XYZRecruitmentDatabase;
CREATE USER Phoebe
FOR LOGIN Venkata;
ALTER ROLE ReimbursementEntry
ADD MEMBER Phoebe;
--Venkata Sai Pawan komaravolu
100 %
Messages
Commands completed successfully.
Completion time: 2020-05-02T17:46:23.8755446-07:00
100 %
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (63) XYZRecruitmentDatabase 00:00:00 0 rows
Ready
Ln 12 Col 31 Ch 31 INS
Type here to search
5:46 PM
5/2/2020

4.6 TRANSACTION

We create a transaction whenever the status of the candidate is changed to blacklisted we send that candidate details to a table which is BlacklistCandidate. For this we create a table and declare the table. We begin the transaction and insert into the BlacklistCandidate the candidateID,CandidateName,CandidateStatus from Candidate details table join JobApplication table on common CandidateDetailsID and where the CandidateStatus is blacklisted and then we Commit the transaction.



We get the new table for Blacklisted Candidate and details are mentioned in the below screenshot


SQLQuery9.sql - DESKTOP-A19REFF\SQLEXPRESSS.XYZRecruitmentDatabase (sa (60))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
candidates
XYZRecruitmentDatabase
Execute
Object Explorer
Connect
DESKTOP-A19REFF\SQLEXPRESSS (SQL S
Databases
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
SQLQuery9.sql - DE...tDatabase (sa (60))*
SQLQuery8.sql - DE...tDatabase (sa (59))*
SQLQuery6.sql - DE...tDatabase (sa (58))*
Select * from BlacklistCandidate
--Venkata Sai Pawan Komaravolu
100 %
Results
Messages
CandidateID CandidateName CandidateStatus
1 901 ChandlerBing Blacklisted
Query executed successfully.
DESKTOP-A19REFF\SQLEXPRESSS... sa (60) XYZRecruitmentDatabase 00:00:00 1 rows
Ready
Ln 2 Col 31 Ch 31 INS
Type here to search
5:38 PM
5/2/2020

5. CONCLUSION

The Implementation and Design of Recruitment Database for a company ‘xyz’ is done in this project. I’ve created database from scratch. In this project, I’ve used all the basic database concepts are implemented i.e. creation of tables, creation of columns, constraints, relationships between the tables used, Entity Relationship diagram, populating data, Normalizing the database, usage of Triggers ,creation of Views, usage of functions, Stored Procedures and transactions. By doing this project I have got great command over various skills on all the database concepts also got experience on working with the real-world project from scratch. We now have a recruitment database and can we implemented as all the test cases are successful and meets the problem statement.

Source Code for creating tables:

```
IF DB_ID('XYZRecruitmentDatabase') IS NOT NULL
    DROP DATABASE XYZRecruitmentDatabase
GO

CREATE DATABASE XYZRecruitmentDatabase;
GO

USE XYZRecruitmentDatabase;

GO

CREATE TABLE Addresses (
  AddressID INT NOT NULL PRIMARY KEY,
  AddressLine1 VARCHAR(50) NOT NULL,
  AddressLine2 VARCHAR(50) Default NULL,
  City CHAR(50) NOT NULL,
  State CHAR(50) NOT NULL,
  ZipCode INT NOT NULL
);

CREATE TABLE ContactDetails (
  ContactID INT NOT NULL PRIMARY KEY,
  EmailAddress VARCHAR(50) NOT NULL,
  PhoneNumber VARCHAR(12)  NOT NULL,
);
```

```
CREATE TABLE Department (
  DepartmentID INT NOT NULL PRIMARY KEY,
  DepartmentName CHAR(30) NOT NULL,
  DepartmentDescription varchar(50) NULL
);

CREATE TABLE CandidateQualification (
  CandidateQualificationID INT NOT NULL PRIMARY KEY,
  GradGpa INT NOT NULL,
  WorkExperienceYears INT NOT NULL,
);

CREATE TABLE Interviewer (
  InterviewerID INT NOT NULL PRIMARY KEY,
  InterviewerName CHAR(50) NOT NULL,
  InterviewerStatus Varchar(50) NULL
);

CREATE TABLE InterviewLocation (
  InterviewLocationID INT NOT NULL PRIMARY KEY,
  BuildingName VARCHAR(50) NOT NULL,
  City CHAR(50) NOT NULL,
  State CHAR(50) NOT NULL,
  ZipCode INT NOT NULL,
);

CREATE TABLE JobOpening (
  JobOpeningID INT NOT NULL PRIMARY KEY,
  JobRole VARCHAR(50) NOT NULL,
  JobDescription VARCHAR(50) NOT NULL,
```

```
  DepartmentID INT NOT NULL REFERENCES Department(DepartmentID),
  JobType VARCHAR(50) NOT NULL,
  NoOfJobOpenings INT NOT NULL,
);

CREATE TABLE CandidateDetails (
  CandidateDetailsID INT NOT NULL PRIMARY KEY,
  CandidateName CHAR(50),
  AddressID INT NOT NULL REFERENCES Addresses(AddressID),
  ContactID INT NOT NULL REFERENCES ContactDetails(ContactID),
  CandidateQualificationID INT NOT NULL REFERENCES
CandidateQualification(CandidateQualificationID),
);
CREATE TABLE JobApplication (
  CandidateID INT NOT NULL PRIMARY KEY,
  JobOpeningID INT NOT NULL REFERENCES JobOpening(JobOpeningID),
  CandidateDetailsID INT NOT NULL REFERENCES CandidateDetails(CandidateDetailsID),
  CandidateStatus VARCHAR(50) NOT NULL,
  CandidateName VARCHAR(50) NOT NULL,
);

CREATE TABLE Interview (
  InterviewID INT NOT NULL PRIMARY KEY,
  InterviewerID INT NOT NULL References Interviewer(InterviewerID),
  CandidateID INT NOT NULL References JobApplication(CandidateID),
  InterviewResult CHAR(50) NOT NULL,
  InterviewType VARCHAR(20) NOT NULL,
);
CREATE TABLE ComplaintHandling (
  ComplaintID INT NOT NULL PRIMARY KEY,
```

```
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  InterviewerID INT NOT NULL References Interviewer(InterviewerID),
   ComplaintDescription VARCHAR(100),
);

CREATE TABLE AirlineReservation (
  TicketID INT NOT NULL PRIMARY KEY,
  InterviewID INT NOT NULL REFERENCES Interview(InterviewID),
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID),
  FlightDetails VARCHAR(100),
  AirlineFare money
);

CREATE TABLE Onboarding (
  OnboardingID INT NOT NULL PRIMARY KEY,
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  JobOpeningID INT NOT NULL REFERENCES JobOpening(JobOpeningID),
  DepartmentID INT NOT NULL REFERENCES Department(DepartmentID),
  CandidateDocuments Varchar(50) NOT NULL
);

CREATE TABLE HotelReservation (
  ReservationID INT NOT NULL PRIMARY KEY,
  InterviewID INT NOT NULL REFERENCES Interview(InterviewID),
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID),
  HotelDetails VARCHAR(100) NOT NULL,
  HotelFare money NOT NULL,
);
```

```
CREATE TABLE CarRental (
  RentalID INT NOT NULL PRIMARY KEY,
  InterviewID INT NOT NULL REFERENCES Interview(InterviewID),
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  InterviewLocationID INT NOT NULL REFERENCES InterviewLocation(InterviewLocationID),
  CarDetails VARCHAR(50) ,
  RentalFare money ,
);

CREATE TABLE Reimbursement (
  ReimbursementID INT NOT NULL PRIMARY KEY,
  InterviewID INT NOT NULL REFERENCES Interview(InterviewID),
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  RentalID INT NOT NULL REFERENCES CarRental(RentalID),
  TicketID INT NOT NULL REFERENCES AirlineReservation(TicketID),
  ReservationID INT NOT NULL REFERENCES HotelReservation(ReservationID),
  ReimbursementAmount MONEY
);
CREATE TABLE JobOfferNegotitation (
  NegotiationID INT NOT NULL PRIMARY KEY,
  CandidateID INT NOT NULL REFERENCES JobApplication(CandidateID),
  NegotitationStatus varchar(50) NOT NULL,
);
```