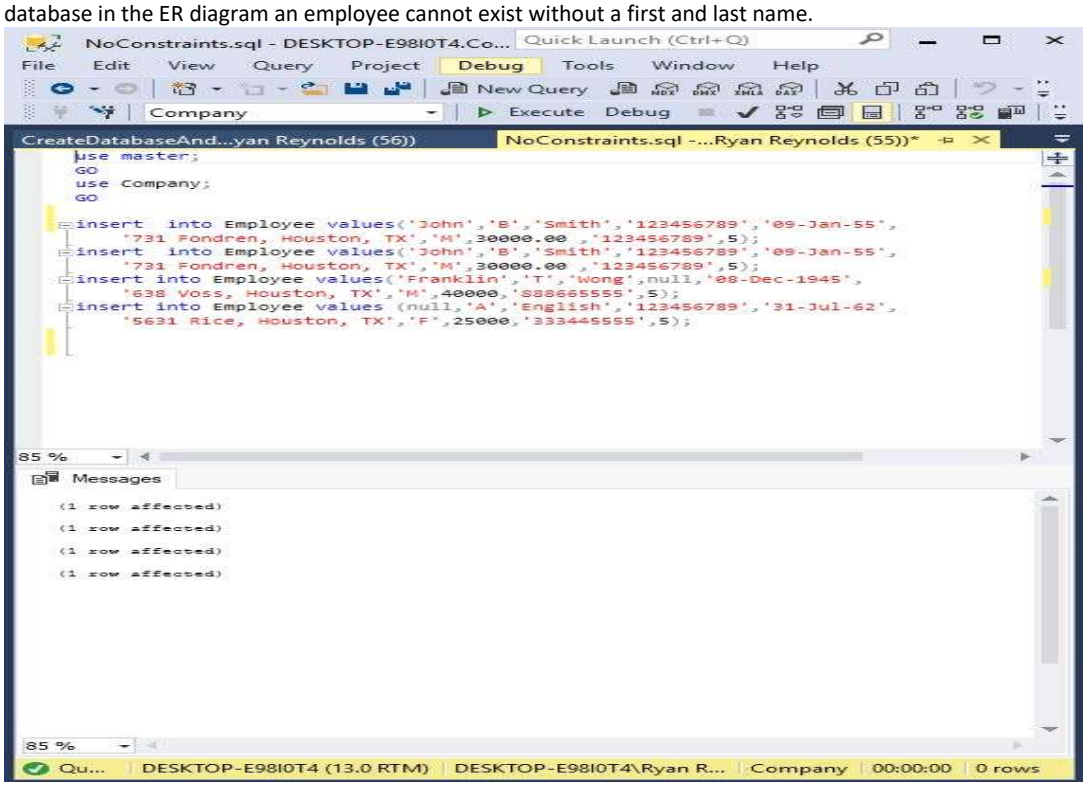
CIS 430: Lab 2
Name: Ryan Reynolds
ID# 2693018

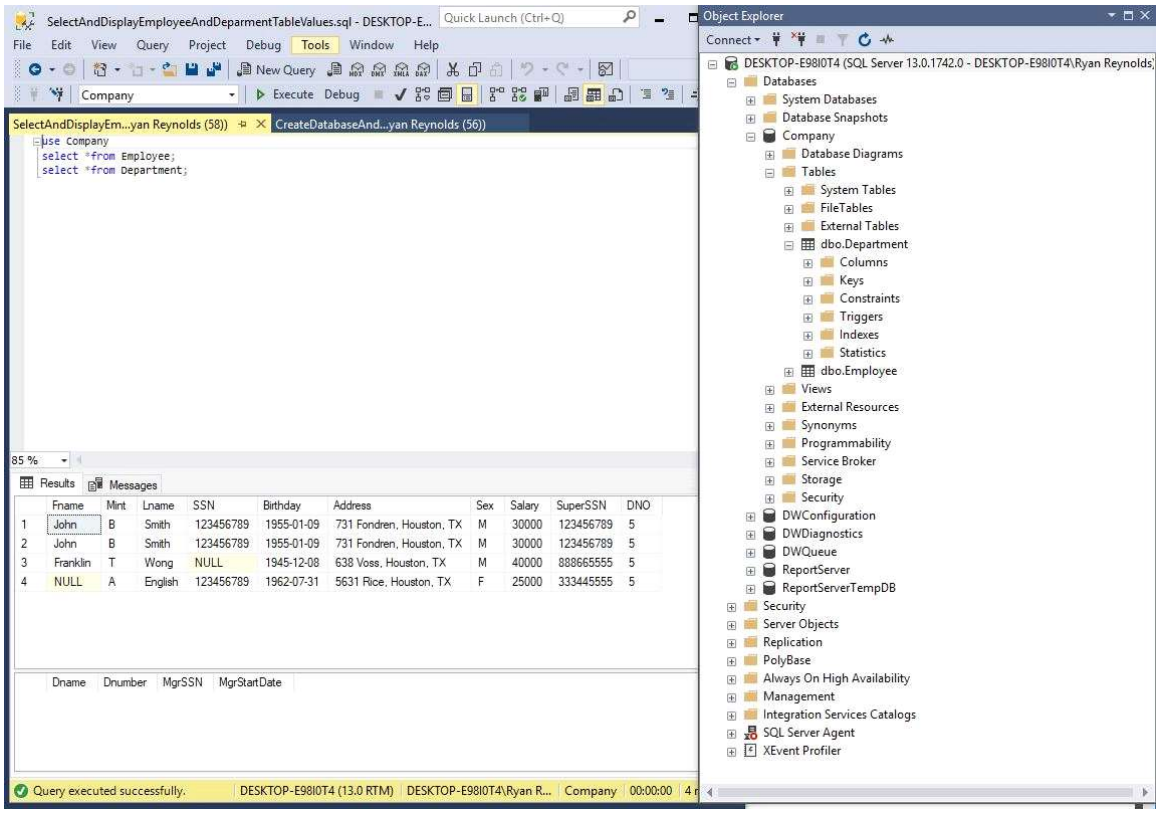
Part 1:

In order to demonstrate the need for constraints values that do not make any logical sense were entered into the database. For example, duplicate social security numbers were entered and without contraints the database stored the value.

```
insert  into Employee values('John','B','Smith','123456789','09-Jan-55',
        '731 Fondren, Houston, TX','M',30000.00 ,'123456789',5);
insert  into Employee values('John','B','Smith','123456789','09-Jan-55',
        '731 Fondren, Houston, TX','M',30000.00 ,'123456789',5);
insert into Employee values('Franklin','T','Wong',null,'08-Dec-1945',
        '638 Voss, Houston, TX','M',40000,'888665555',5);
insert into Employee values (null,'A','English','123456789','31-Jul-62',
        '5631 Rice, Houston, TX','F',25000,'333445555',5);
```

Additionally, null values were entered for employee first and last name. By the definition of the company database in the ER diagram an employee cannot exist without a first and last name.





Part 2:

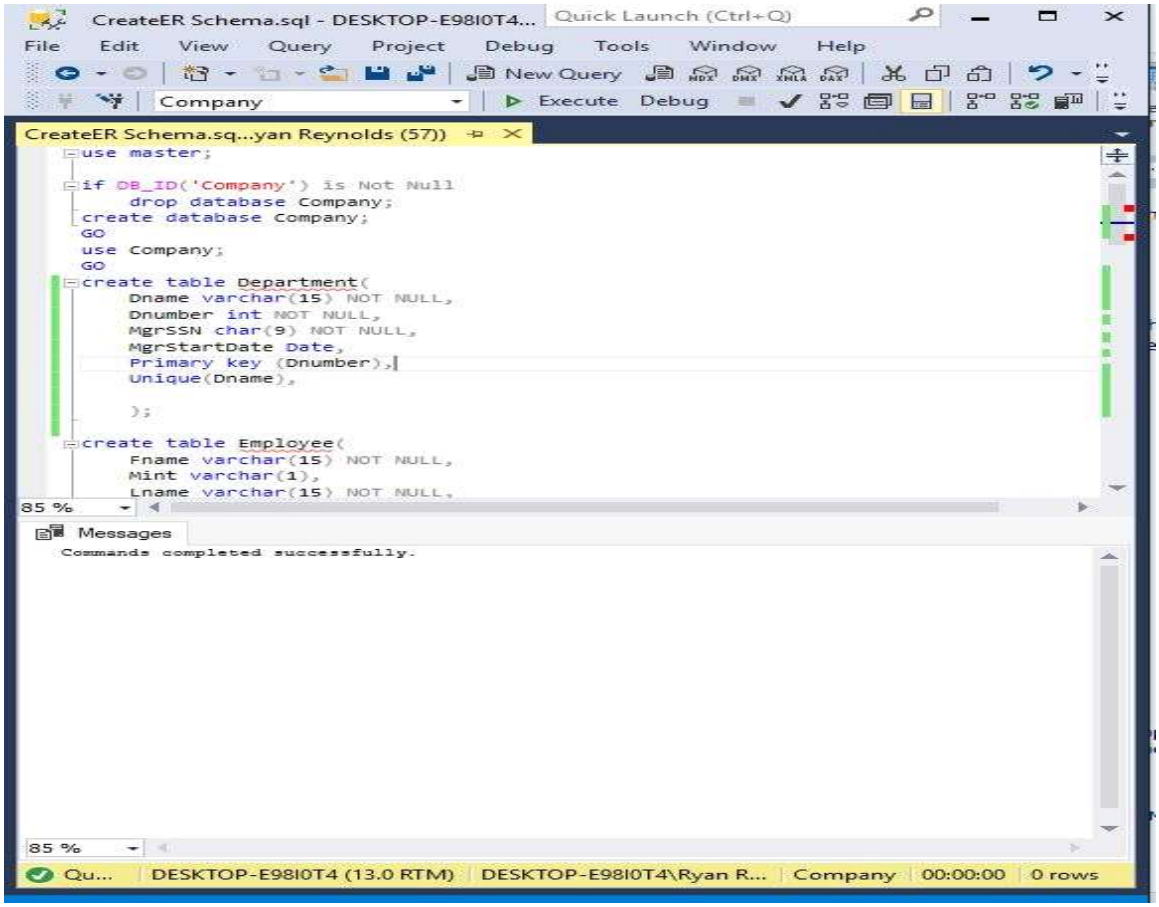
The database created for part one was dropped in order to set up the database with the correct constrained ER schema. The following command dropped the company database.

```
if DB_ID('Company') is Not Null
        drop database Company;
create database Company;
```

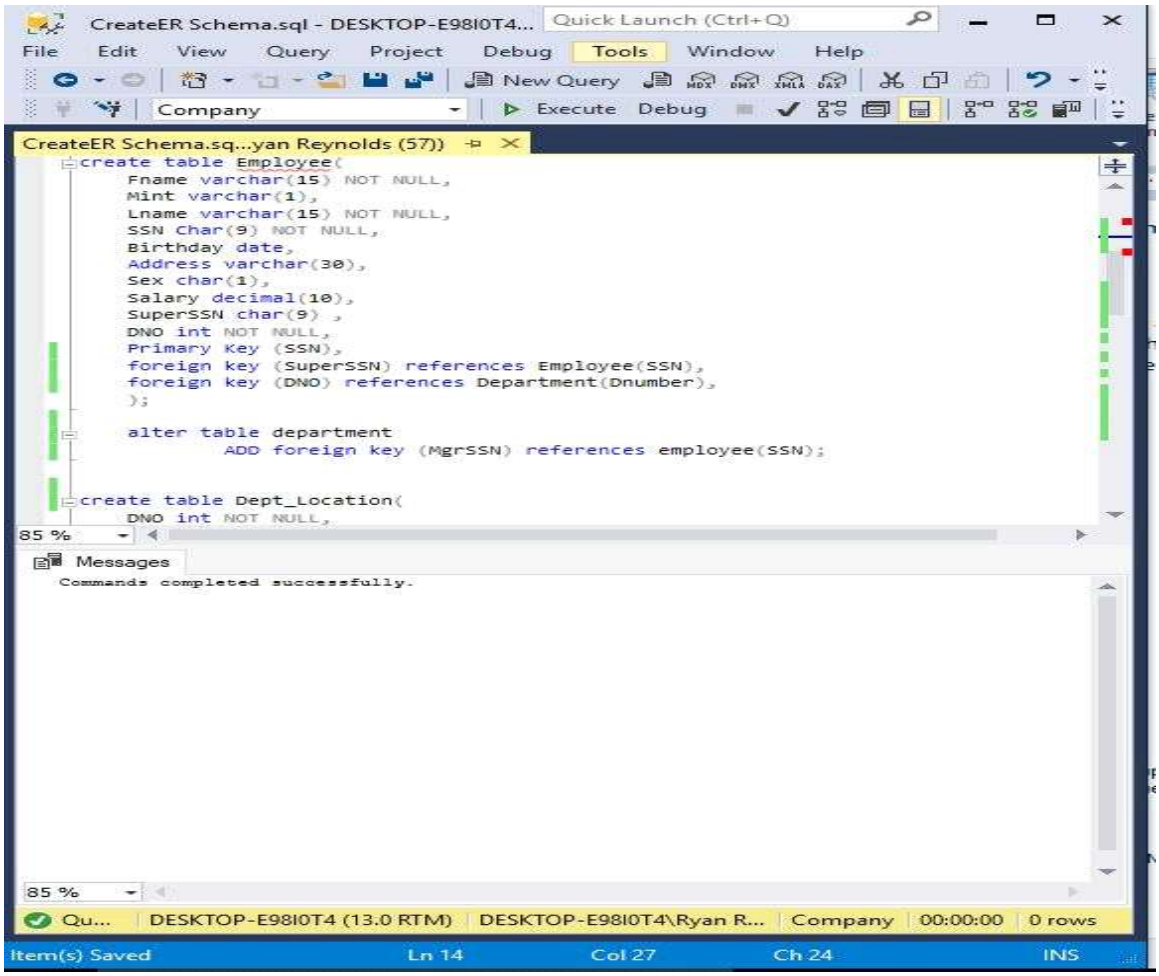
Furthermore, this line was implemented at the begining of the table creation query to allow the user to wipe the values in case of schema errors on future labs.

The employee and department tables were recreated with the corrected schema.

```
create table Department(
        Dname varchar(15) NOT NULL,
        Dnumber int NOT NULL,
        MgrSSN char(9) NOT NULL,
        MgrStartDate Date,
        Primary key (Dnumber),
        Unique(Dname),

        );
```




```
create table Employee(
        Fname varchar(15) NOT NULL,
        Mint varchar(1),
        Lname varchar(15) NOT NULL,
        SSN Char(9) NOT NULL,
        Birthday date,
        Address varchar(30),
        Sex char(1),
        Salary decimal(10),
        SuperSSN char(9) ,
        DNO int NOT NULL,
        Primary Key (SSN),
        foreign key (SuperSSN) references Employee(SSN),
        foreign key (DNO) references Department(Dnumber),
        );
```

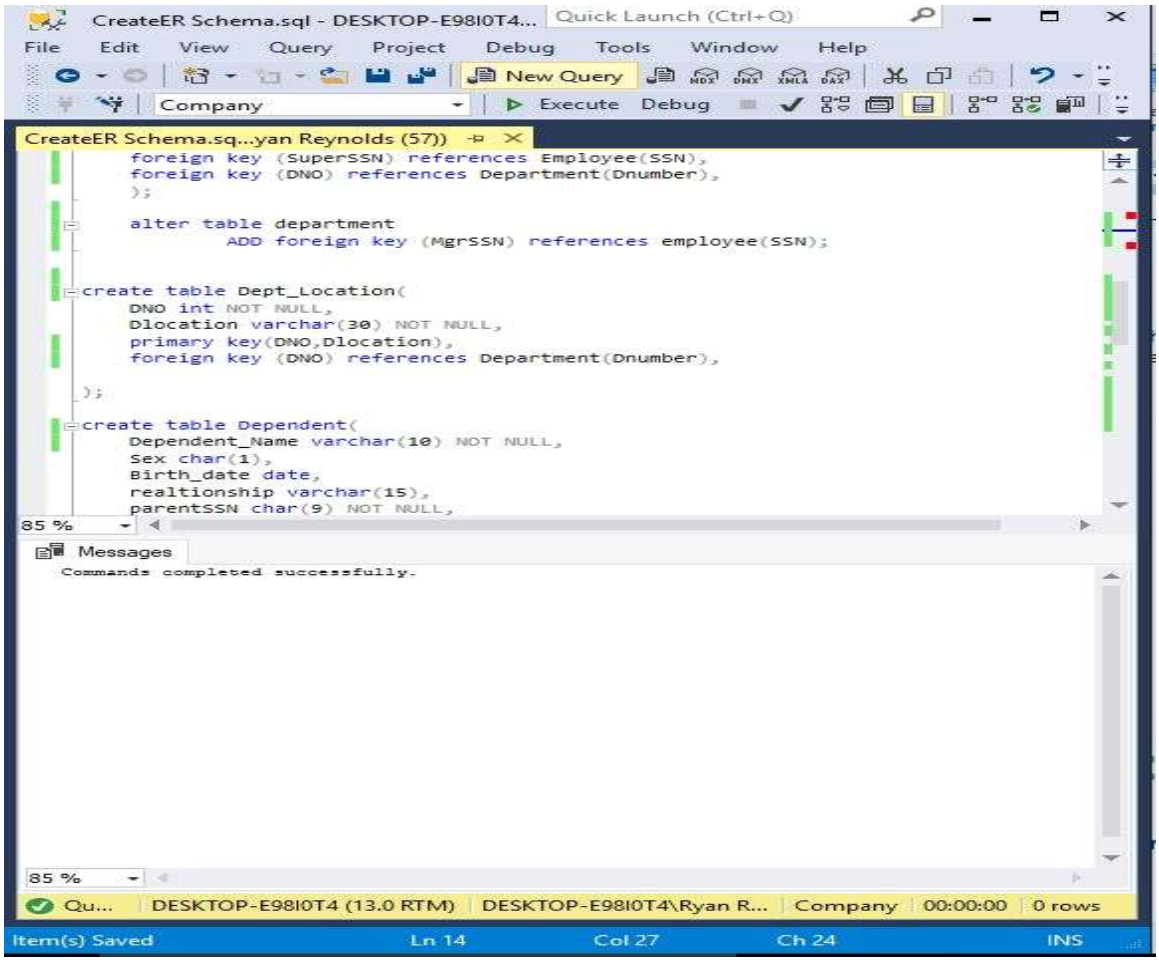


The foreign key that links a manager's social security number to the employee's social security number was implemented using the alter command. This was required because the employee table did not exist when the department table was created.

```
alter table department
                ADD foreign key (MgrSSN) references employee(SSN);
```

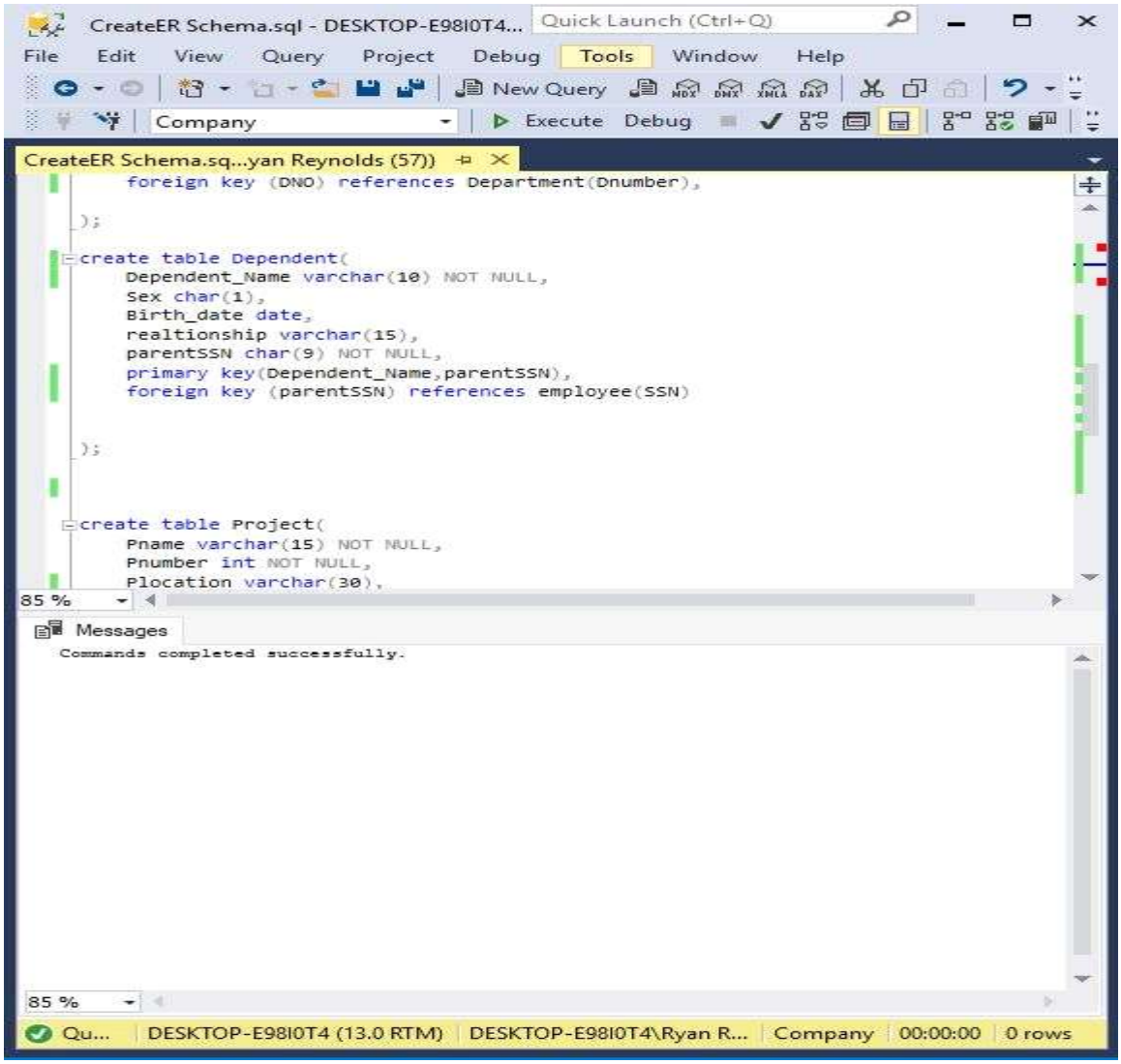
Next, the department location table was created. Departemtn location uses a primary key of the department number (DNO) and department location (Dlocation). Additionally, a foreign key of DNO was created to linke the department number stored here to the department number stored in the department table.

```
create table Dept_Location(
        DNO int NOT NULL,
        Dlocation varchar(30) NOT NULL,
        primary key(DNO,Dlocation),
        foreign key (DNO) references Department(Dnumber),

);
```



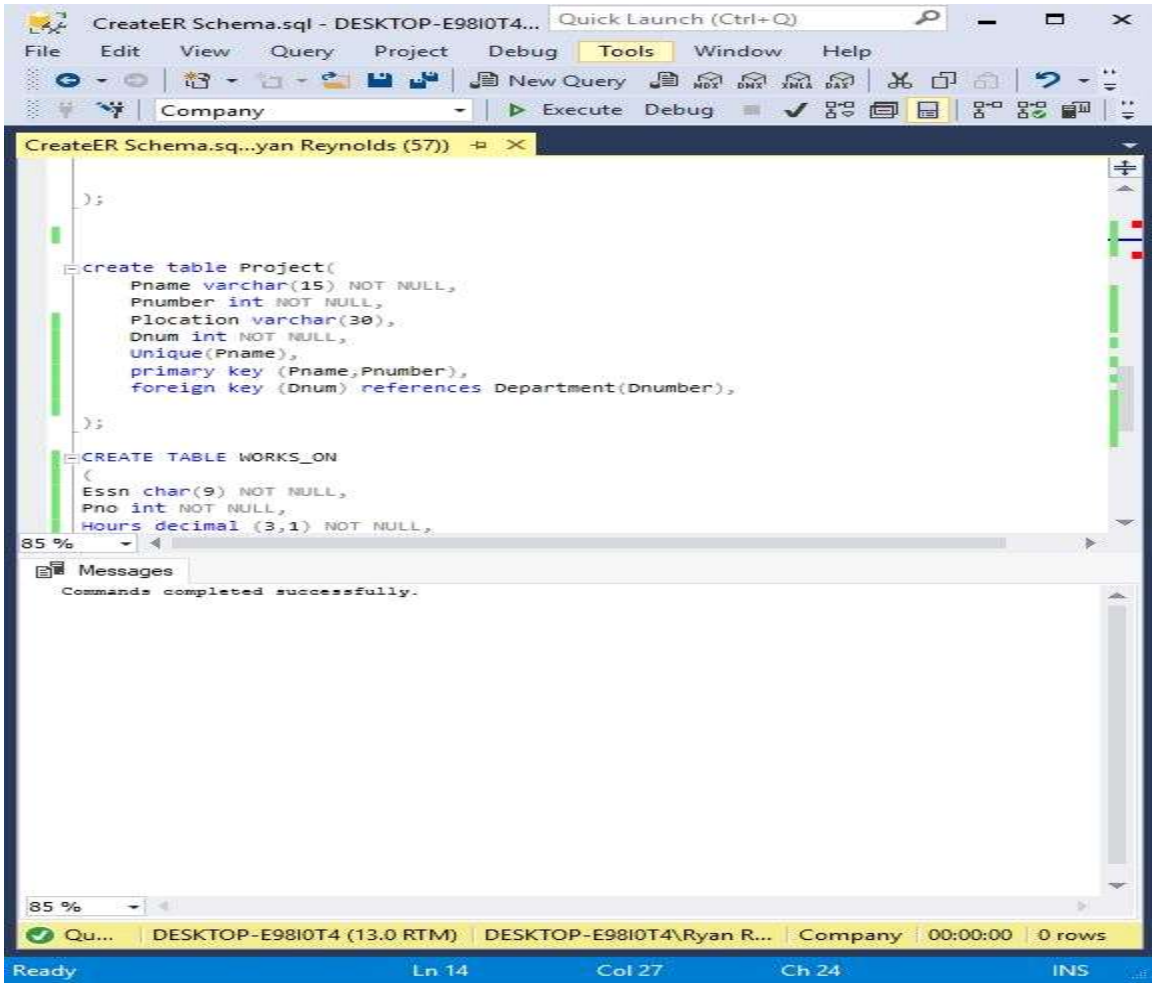
The dependent table was created after the dept_location table. The primary key for dependent consists of the combination of the dependent's name and their parent/guardian's social security number. Parent social security was linked to the employee social security number (SSN) within the employee table by creating a foreign key.

```
create table Dependent(
        Dependent_Name varchar(10) NOT NULL,
        Sex char(1),
        Birth_date date,
        realtionship varchar(15),
        parentSSN char(9) NOT NULL,
        primary key(Dependent_Name,parentSSN),
        foreign key (parentSSN) references employee(SSN)


);
```



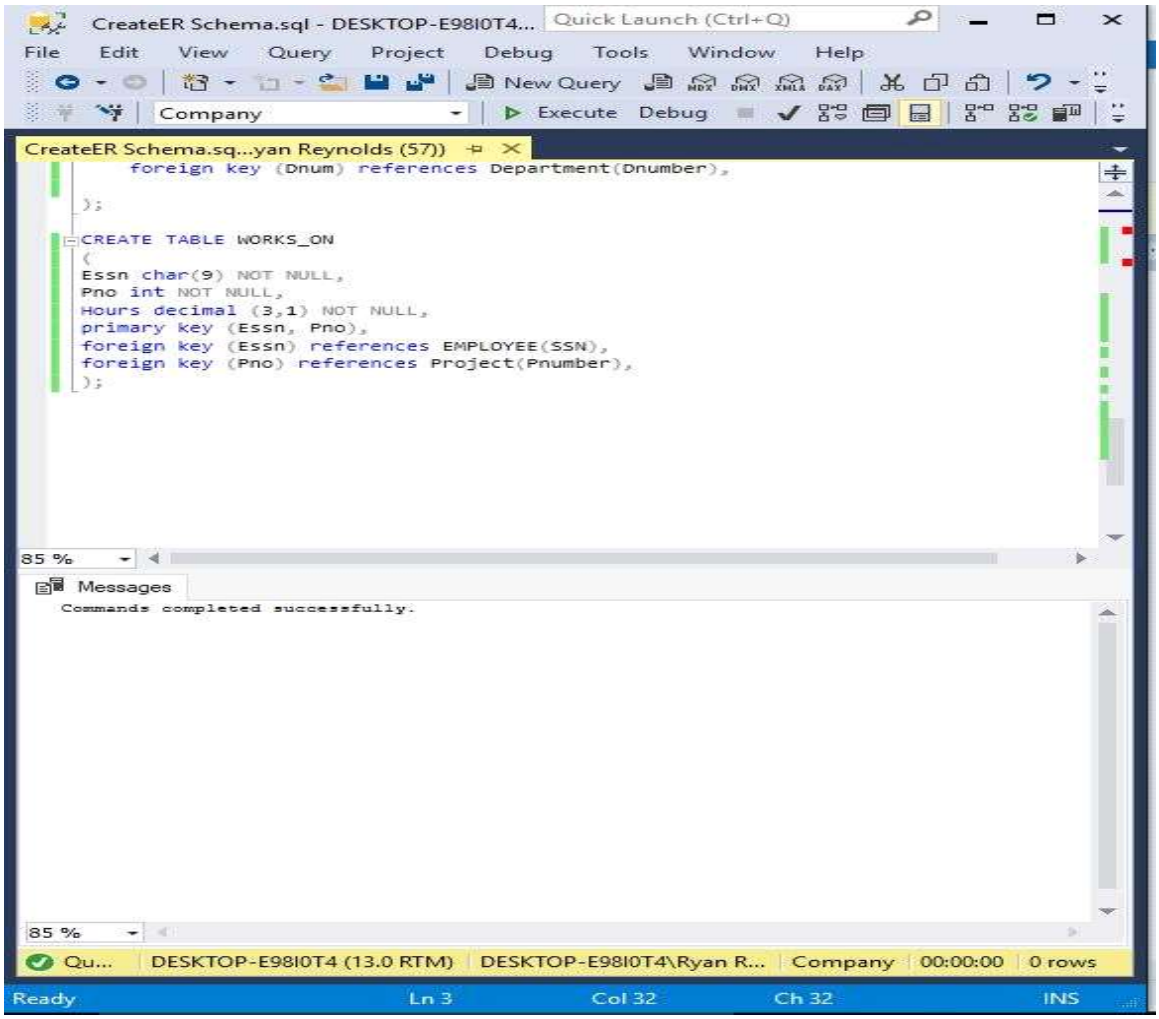
Following the dependent table was the creation of the project table. The primary key chosen was project number (Pnumber). Both Pname and Pnumber were given the constraint not null. Pname was further constrained to be unique to prevent duplicate projects of the same name. Dnum was defined as a foreign key referencing Dnumber in the department table. This relationship will be used to define what department controls the project.

```
create table Project(
        Pname varchar(15) NOT NULL,
        Pnumber int NOT NULL,
        Plocation varchar(30),
        Dnum int NOT NULL,
        Unique(Pname),
        primary key (Pname,Pnumber),
        foreign key (Dnum) references Department(Dnumber),

);
```



Finally, the Works_On table was created to help define the relationship between employee and project. Essn, Pno, and Hours were defined as not null. Pno was chosen as the primary key

```
CREATE TABLE WORKS_ON
        (
        Essn char(9) NOT NULL,
        Pno int NOT NULL,
        Hours decimal (3,1) NOT NULL,
        primary key (Essn, Pno),
        foreign key (Essn) references EMPLOYEE(SSN),
        foreign key (Pno) references Project(Pnumber),
        );
```

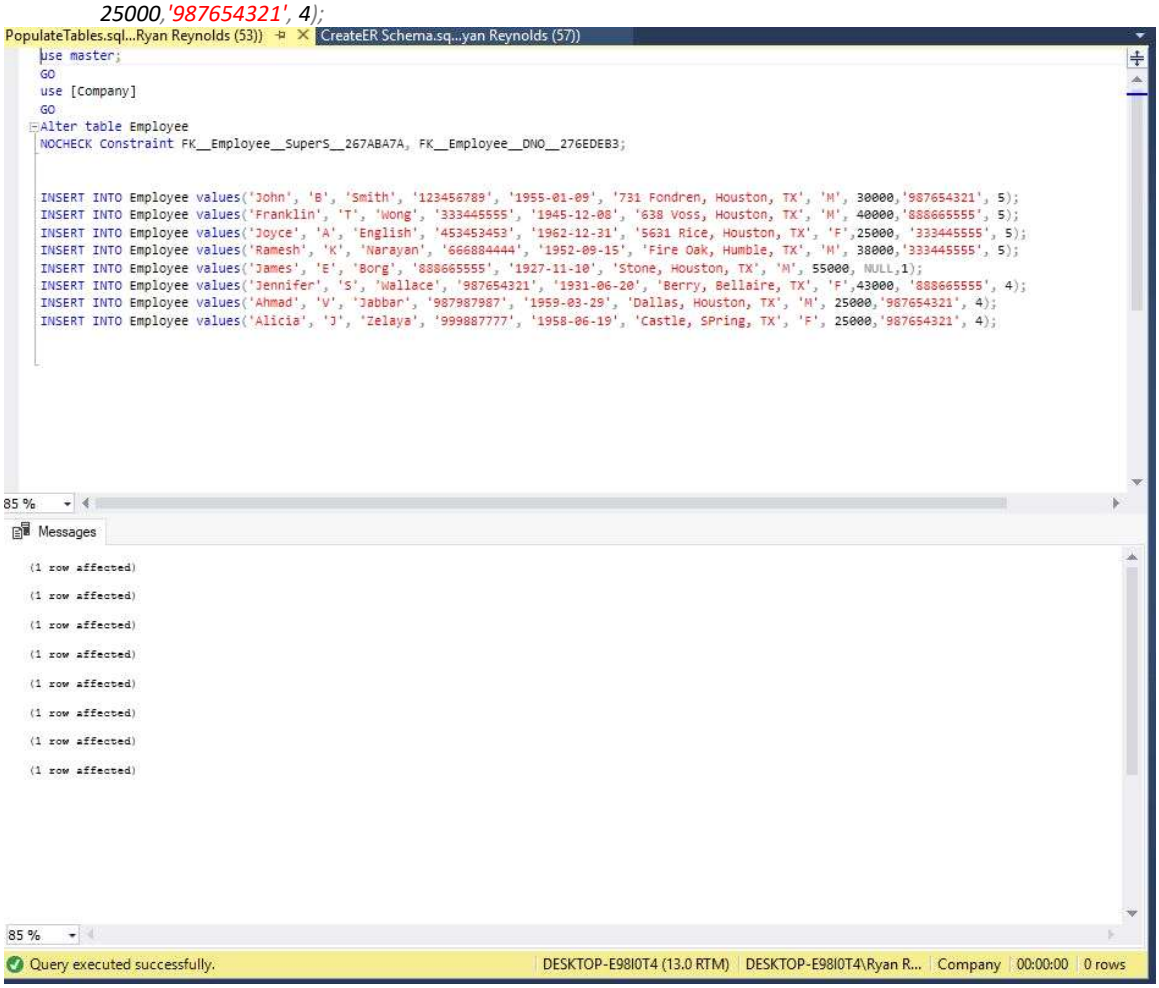


Part 3:

Populate the employee table with the values specified in the lab. The Nocheck constraint command was used to bypass the foreign keys as there is no existing table to check the references against. The foreign key constraints will apply to all data enter after these initial tuples.

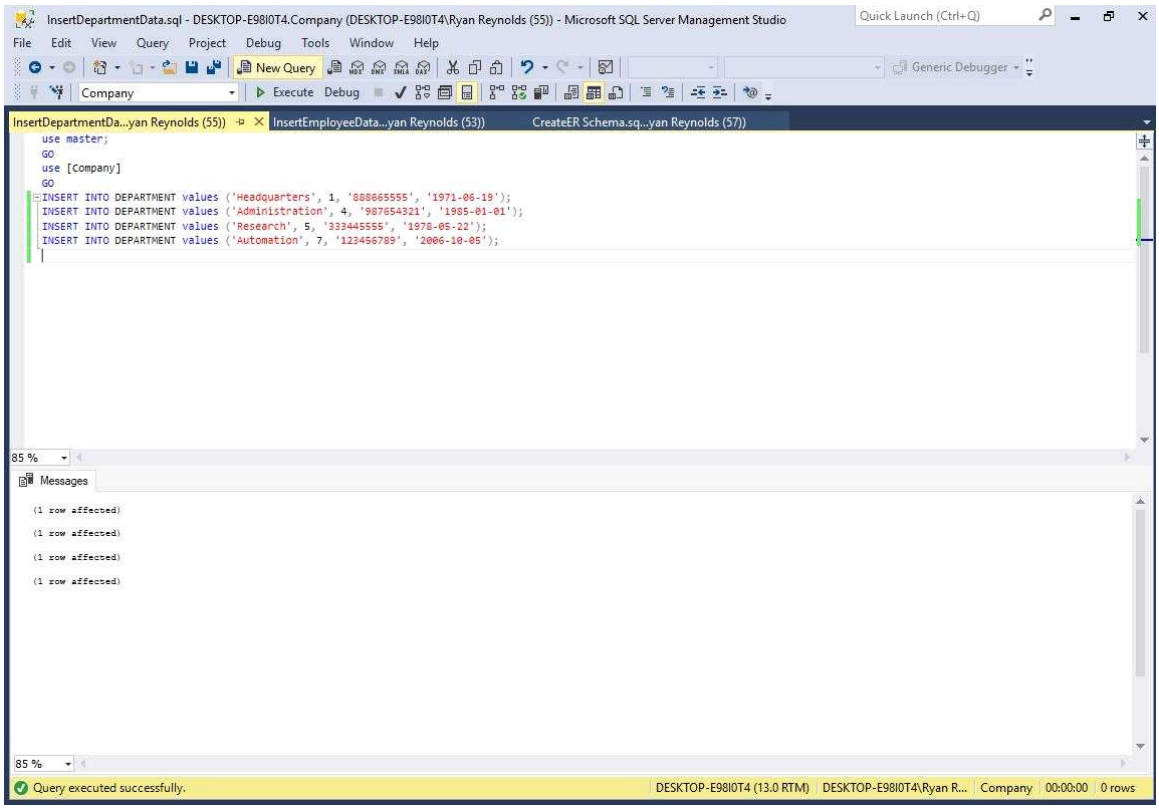
```
Alter table Employee
NOCHECK Constraint FK__Employee__SuperS__267ABA7A, FK__Employee__DNO__276EDEB3;


INSERT INTO Employee values('John', 'B', 'Smith', '123456789', '1955-01-09', '731 Fondren, Houston, TX',
'M', 30000,'987654321', 5);
INSERT INTO Employee values('Franklin', 'T', 'Wong', '333445555', '1945-12-08', '638 Voss, Houston, TX', 'M',
40000,'888665555', 5);
INSERT INTO Employee values('Joyce', 'A', 'English', '453453453', '1962-12-31', '5631 Rice, Houston, TX',
'F',25000, '333445555', 5);
INSERT INTO Employee values('Ramesh', 'K', 'Narayan', '666884444', '1952-09-15', 'Fire Oak, Humble, TX',
'M', 38000,'333445555', 5);
INSERT INTO Employee values('James', 'E', 'Borg', '888665555', '1927-11-10', 'Stone, Houston, TX', 'M',
55000, NULL,1);
INSERT INTO Employee values('Jennifer', 'S', 'Wallace', '987654321', '1931-06-20', 'Berry, Bellaire, TX',
'F',43000, '888665555', 4);
```

```
INSERT INTO Employee values('Ahmad', 'V', 'Jabbar', '987987987', '1959-03-29', 'Dallas, Houston, TX', 'M',
25000,'987654321', 4);
INSERT INTO Employee values('Alicia', 'J', 'Zelaya', '999887777', '1958-06-19', 'Castle, SPring, TX', 'F',
25000,'987654321', 4);
```



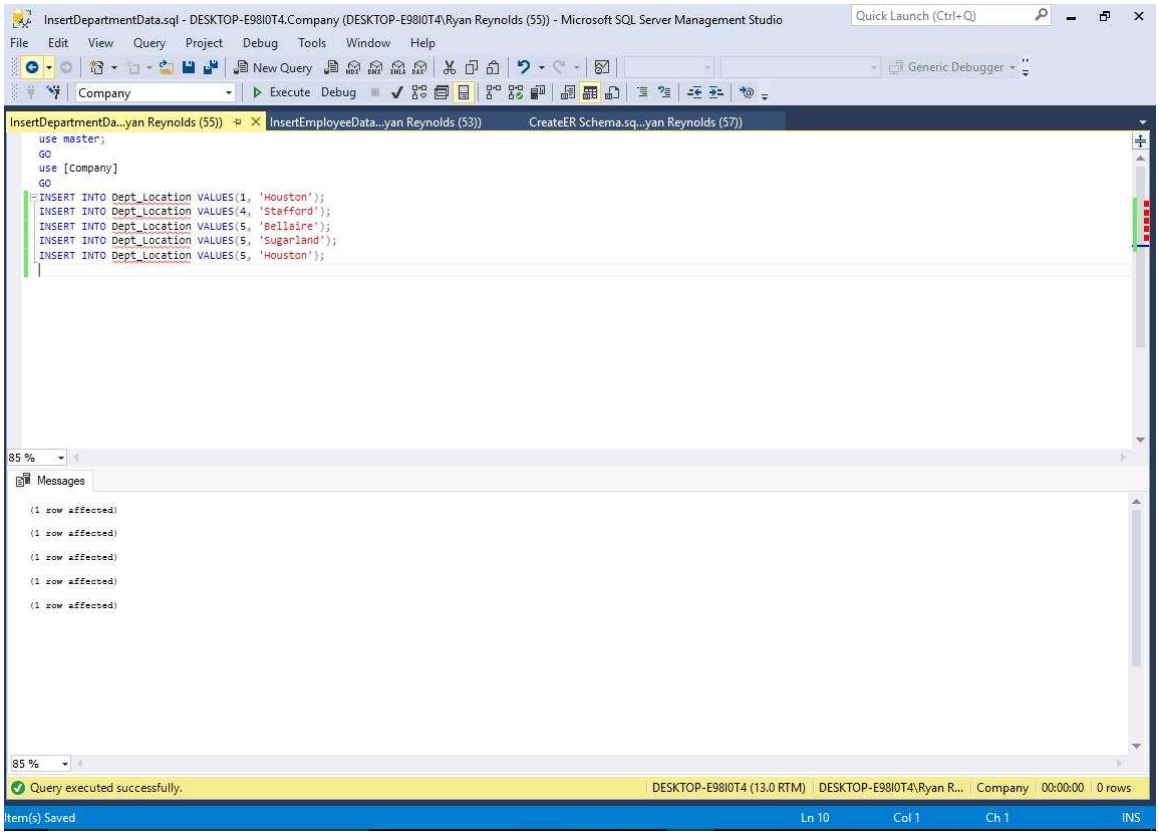
Populate the department table with the apporpriate values specified in the lab handout.

```
INSERT INTO DEPARTMENT values ('Headquarters', 1, '888665555', '1971-06-19');
INSERT INTO DEPARTMENT values ('Administration', 4, '987654321', '1985-01-01');
INSERT INTO DEPARTMENT values ('Research', 5, '333445555', '1978-05-22');
INSERT INTO DEPARTMENT values ('Automation', 7, '123456789', '2006-10-05');
```



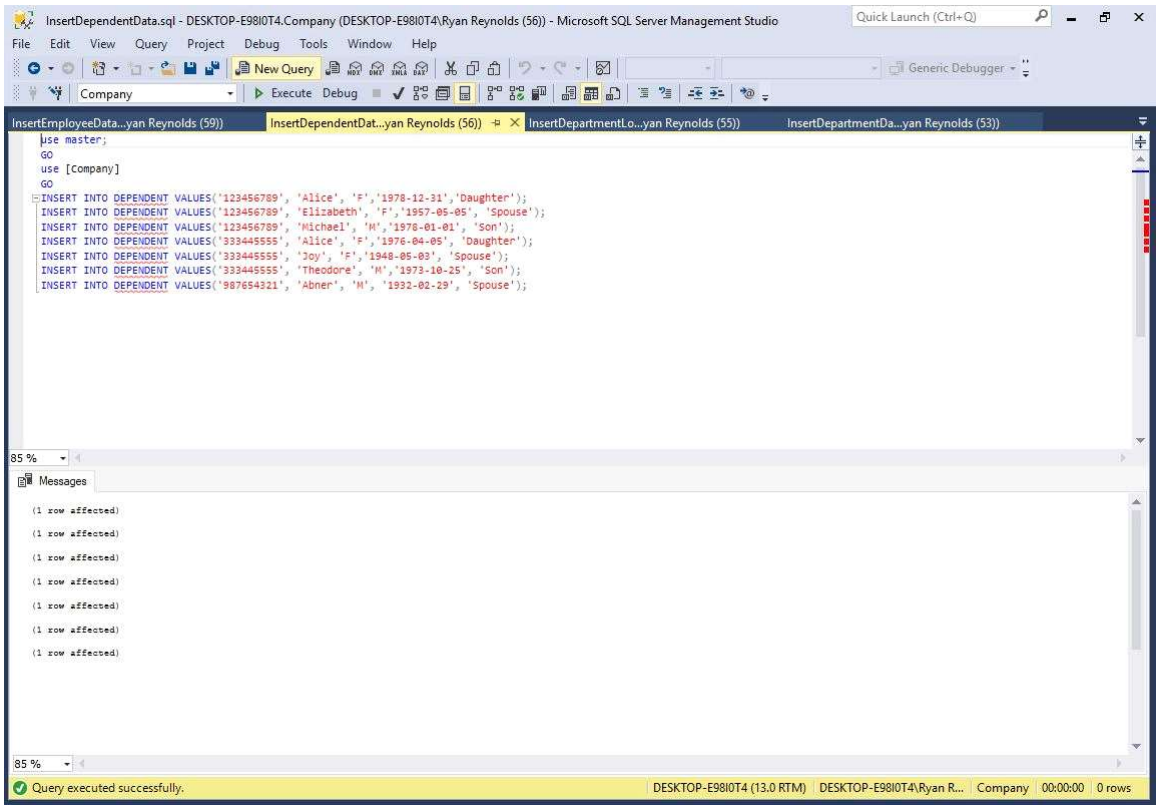
Populate the dept_location table with the values specified in the lab.

```
INSERT INTO Dept_Location VALUES(1, 'Houston');
INSERT INTO Dept_Location VALUES(4, 'Stafford');
INSERT INTO Dept_Location VALUES(5, 'Bellaire');
INSERT INTO Dept_Location VALUES(5, 'Sugarland');
INSERT INTO Dept_Location VALUES(5, 'Houston');
```



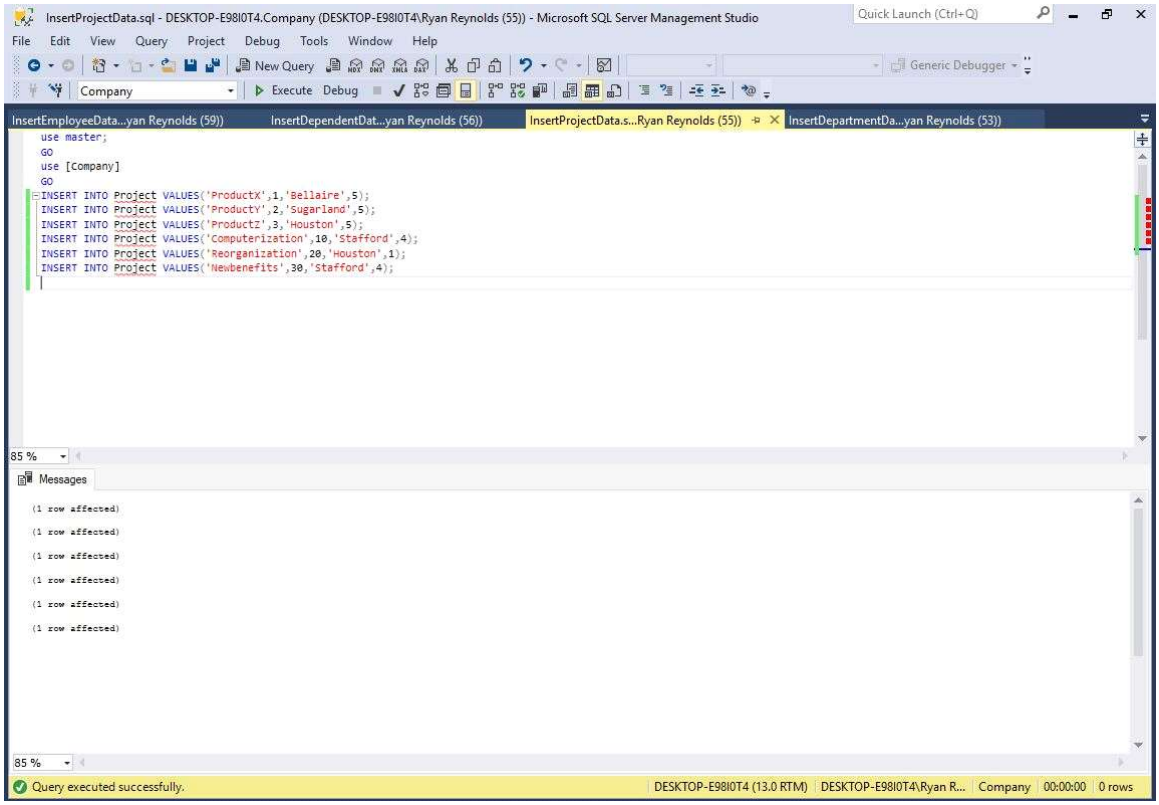
Populate the dependent tuples with the values specified in the lab.

```
INSERT INTO DEPENDENT VALUES('123456789', 'Alice', 'F','1978-12-31','Daughter');
INSERT INTO DEPENDENT VALUES('123456789', 'Elizabeth', 'F','1957-05-05', 'Spouse');
INSERT INTO DEPENDENT VALUES('123456789', 'Michael', 'M','1978-01-01', 'Son');
INSERT INTO DEPENDENT VALUES('333445555', 'Alice', 'F','1976-04-05', 'Daughter');
INSERT INTO DEPENDENT VALUES('333445555', 'Joy', 'F','1948-05-03', 'Spouse');
INSERT INTO DEPENDENT VALUES('333445555', 'Theodore', 'M','1973-10-25', 'Son');
INSERT INTO DEPENDENT VALUES('987654321', 'Abner', 'M', '1932-02-29', 'Spouse');
```



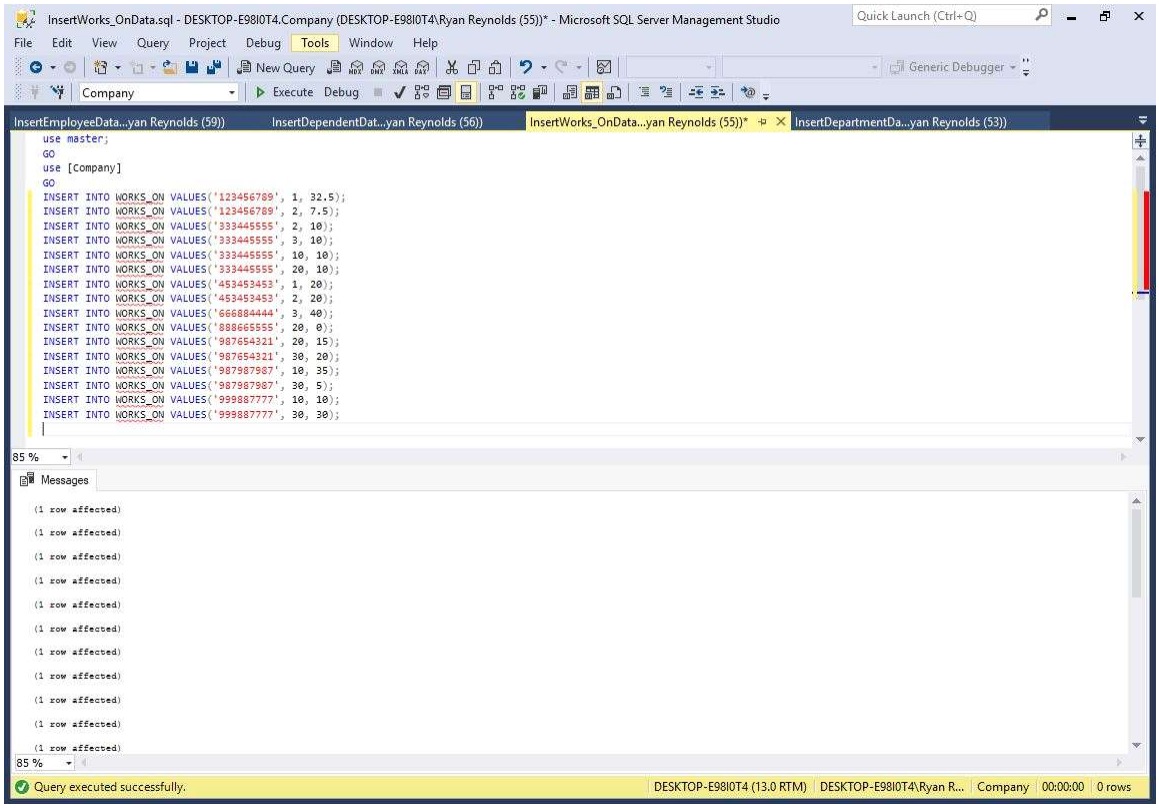
Populate the project table with the values specified in the lab table.

```
INSERT INTO Project VALUES('ProductX',1,'Bellaire',5);
INSERT INTO Project VALUES('ProductY',2,'Sugarland',5);
INSERT INTO Project VALUES('ProductZ',3,'Houston',5);
INSERT INTO Project VALUES('Computerization',10,'Stafford',4);
INSERT INTO Project VALUES('Reorganization',20,'Houston',1);
INSERT INTO Project VALUES('Newbenefits',30,'Stafford',4);
```



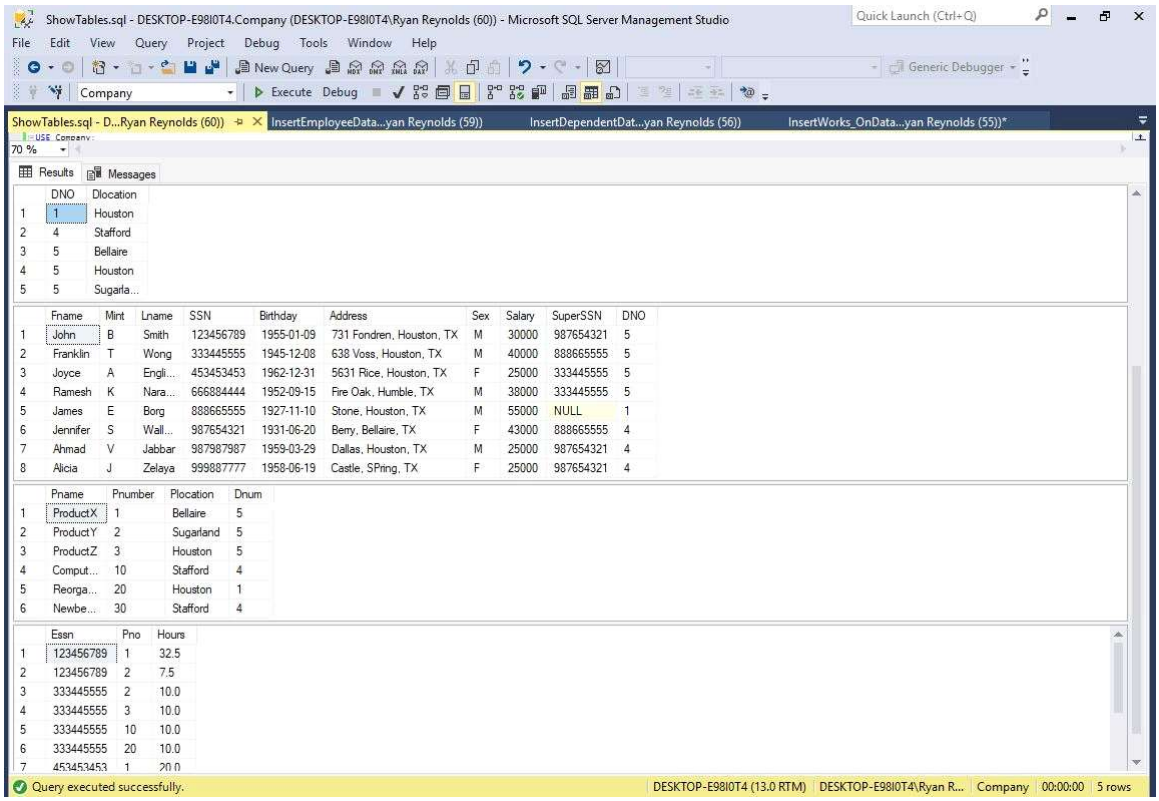
Populate the works_on table with the values specified in the lab.

```
INSERT INTO WORKS_ON VALUES('123456789', 1, 32.5);
INSERT INTO WORKS_ON VALUES('123456789', 2, 7.5);
INSERT INTO WORKS_ON VALUES('333445555', 2, 10);
INSERT INTO WORKS_ON VALUES('333445555', 3, 10);
INSERT INTO WORKS_ON VALUES('333445555', 10, 10);
INSERT INTO WORKS_ON VALUES('333445555', 20, 10);
INSERT INTO WORKS_ON VALUES('453453453', 1, 20);
INSERT INTO WORKS_ON VALUES('453453453', 2, 20);
INSERT INTO WORKS_ON VALUES('666884444', 3, 40);
INSERT INTO WORKS_ON VALUES('888665555', 20, 0);
INSERT INTO WORKS_ON VALUES('987654321', 20, 15);
INSERT INTO WORKS_ON VALUES('987654321', 30, 20);
INSERT INTO WORKS_ON VALUES('987987987', 10, 35);
INSERT INTO WORKS_ON VALUES('987987987', 30, 5);
INSERT INTO WORKS_ON VALUES('999887777', 10, 10);
INSERT INTO WORKS_ON VALUES('999887777', 30, 30);
```



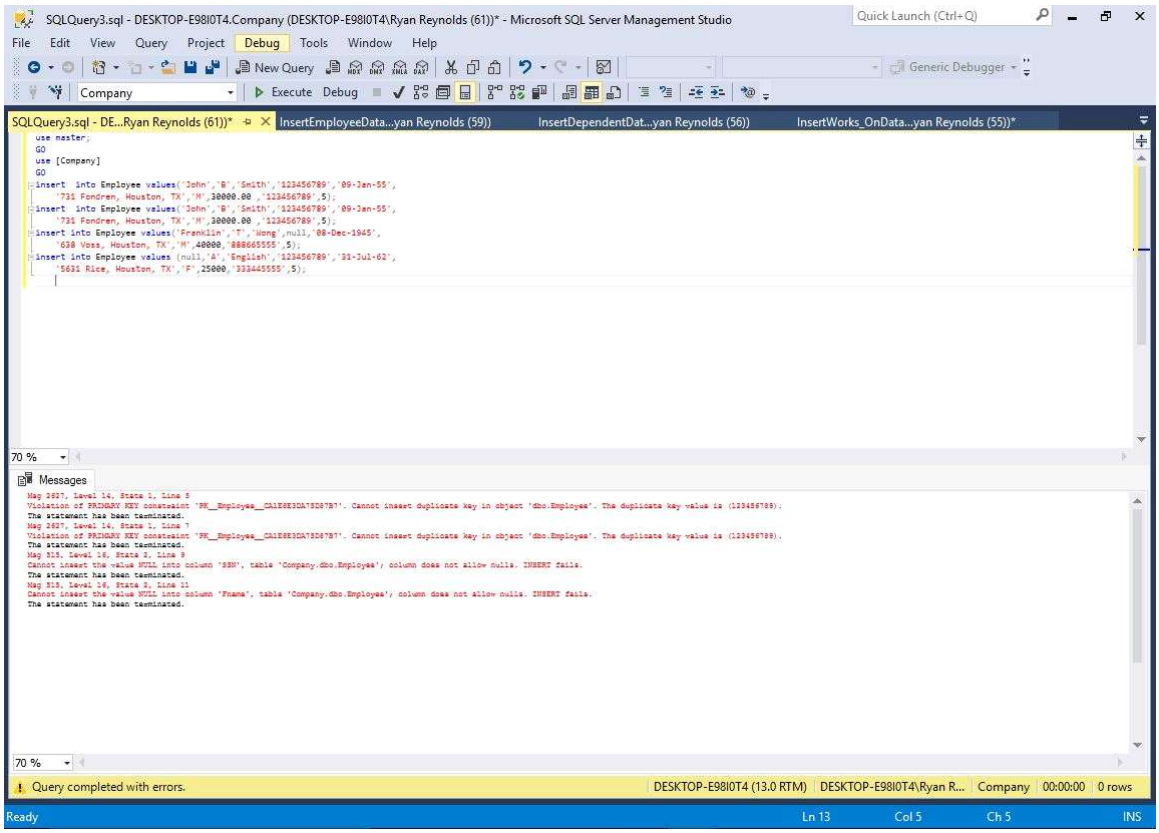
Display the tables showing the populated tuples.

```
USE Company;
select * from DEPARTMENT;
select * from DEPENDENT;
select * from Dept_location;
select * From EMPLOYEE;
select * from PROJECT;
select * from WORKS_ON;
```

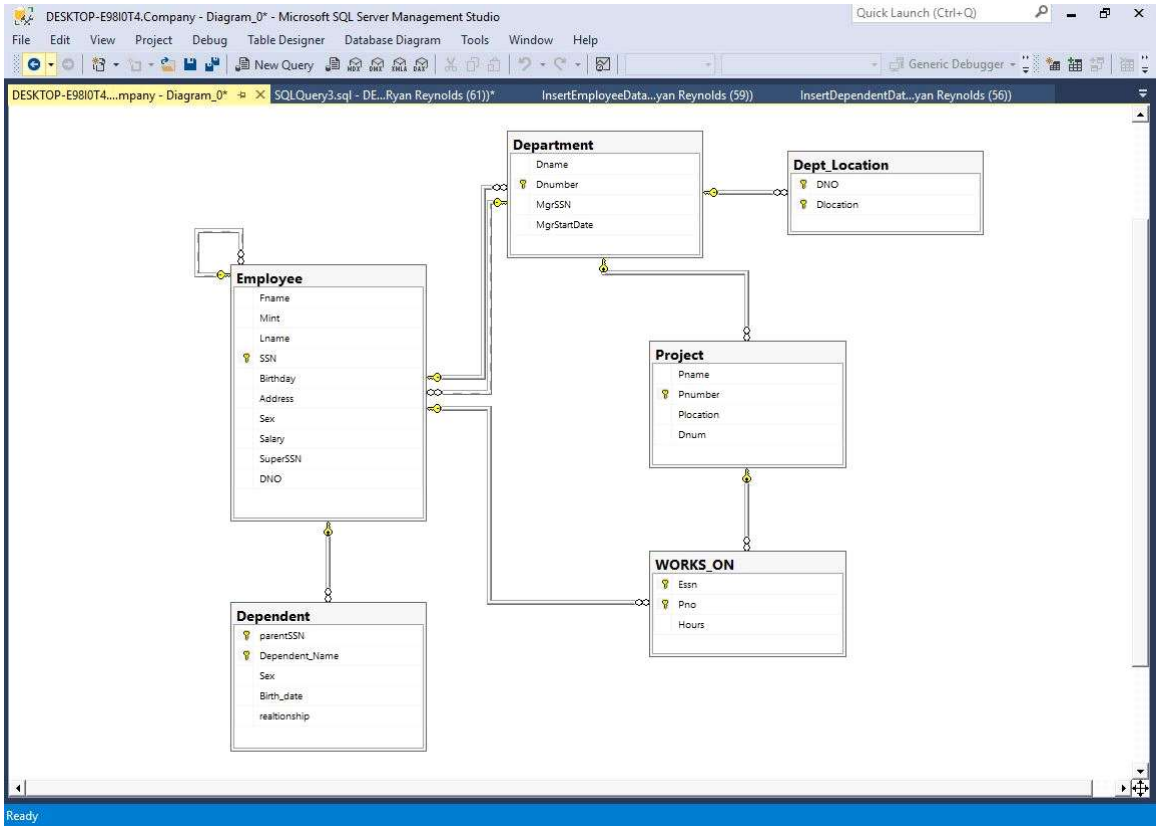


Re-entry of the values from part 1. Now the constraints catch the null and duplicate entries resulting in the line terminations shown in the output message.

```
insert  into Employee values('John','B','Smith','123456789','09-Jan-55',
        '731 Fondren, Houston, TX','M',30000.00 ,'123456789',5);
insert  into Employee values('John','B','Smith','123456789','09-Jan-55',
        '731 Fondren, Houston, TX','M',30000.00 ,'123456789',5);
insert into Employee values('Franklin','T','Wong',null,'08-Dec-1945',
        '638 Voss, Houston, TX','M',40000,'888665555',5);
insert into Employee values (null,'A','English','123456789','31-Jul-62',
        '5631 Rice, Houston, TX','F',25000,'333445555',5);
```



The generated ER Diagram showing the relationships between the different tables.


DESKTOP-E98I0T4.Company - Diagram_0* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Project Debug Table Designer Database Diagram Tools Window Help
New Query
Generic Debugger
DESKTOP-E98I0T4....mpany - Diagram_0*
SQLQuery3.sql - DE...Ryan Reynolds (61))*
InsertEmployeeData...yan Reynolds (59))
InsertDependentDat...yan Reynolds (56))
Department
Dname
Dnumber
MgrSSN
MgrStartDate
Dept_Location
DNO
Dlocation
Employee
Fname
Mint
Lname
SSN
Birthday
Address
Sex
Salary
SuperSSN
DNO
Project
Pname
Pnumber
Plocation
Dnum
WORKS_ON
Essn
Pno
Hours
Dependent
parentSSN
Dependent_Name
Sex
Birth_date
realtionship
Ready