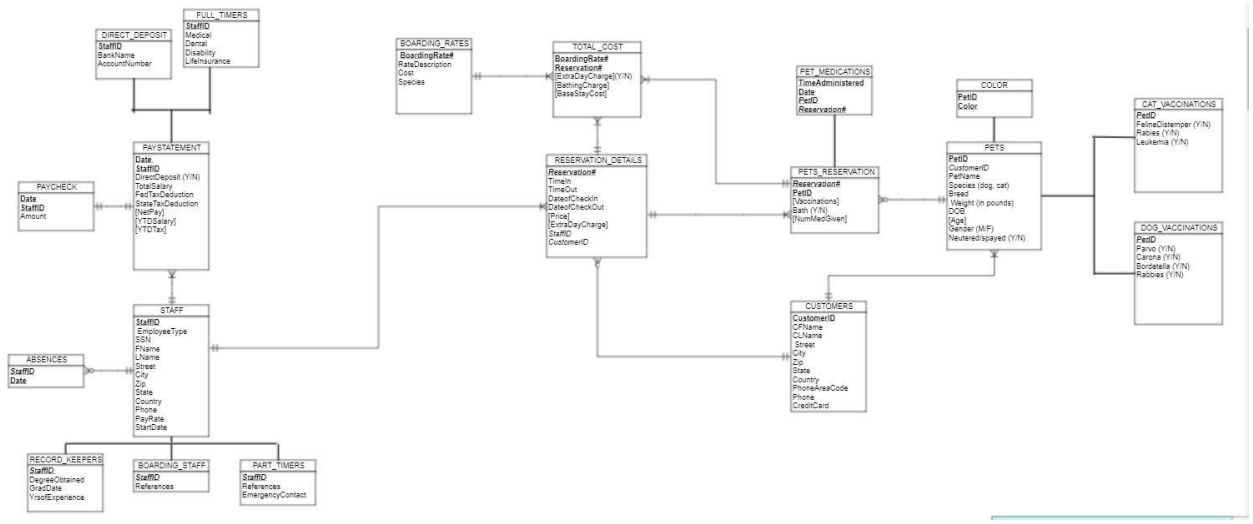
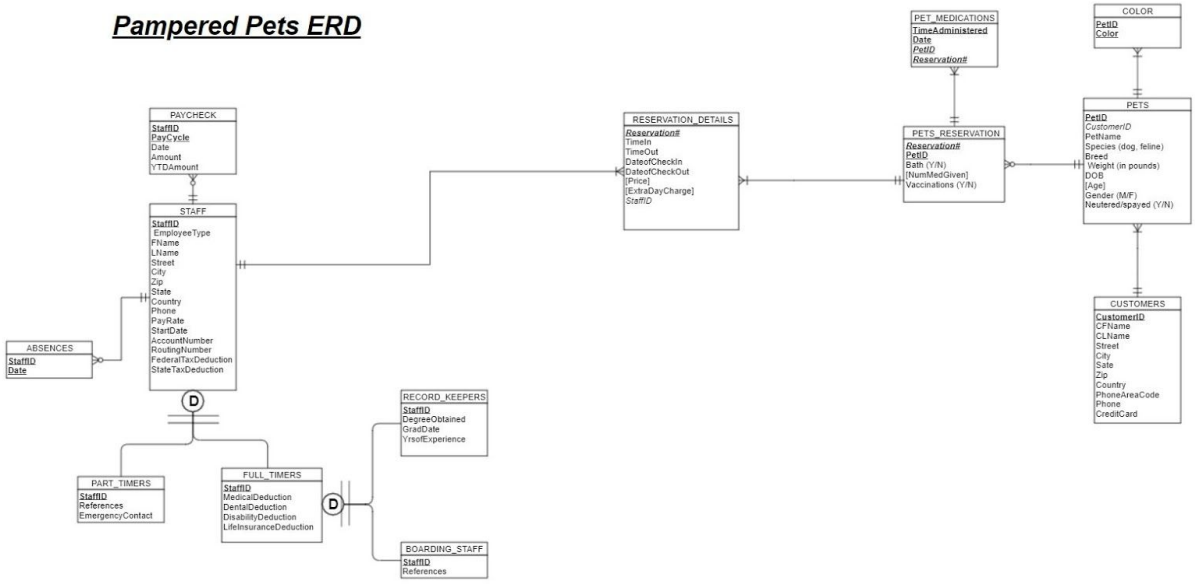
Pampered Pets ERD



Revised ERD for Pampered Pets (After Speaking With Client)



Data Dictionary Entries

Entity	Attribute	Definition
CUSTOMERS *(PK)*	CustomerID	The letter C plus an 8 -digit number uniquely identifying the customer, assigned by Pampered Pets, not based on SSN or other ID.
CUSTOMERS	CFName	(Required), character data, max-length 30, Customers First Name.
CUSTOMERS	CLName	(Required), character data, max-length 30,

		Customers Last Name.
CUSTOMERS	Street	Digit and character data, max-length 30.
CUSTOMERS	City	Character data, max-length 20.
CUSTOMERS	Zip	5 digit number.
CUSTOMERS	State	Character data, max-length 20.
CUSTOMERS	Country	Character data, max-length 20.
CUSTOMERS	PhoneAreaCode	3 digit number that tells phone number area code.
CUSTOMERS	Phone	7 digit number, customer phone number.
CUSTOMERS	CreditCard	16 digit number, customer card number for payment purposes.

PETS *(PK)*	PetID	The letter “C” or “D” to identify the data as pertaining to either a cat or a dog respectively followed by 8 consecutive digits to uniquely identify each pet (example C89714589 for a cat, and D89714512) Number string after the character C or D may not be re-used for the other type of pet not previously used.
PETS	CustomerID	Foreign Key.
PETS	PetName	Character data - Max 20 characters to write down the pets name.
PETS	Species	Character data- Max 3 characters to state whether cat or dog.
PETS	Breed	Character data - Max 30 characters to write down the breed of animal.
PETS	Weight	LBS.
PETS	DOB	Date.
PETS	Age	Derived from DOB.
PETS	Gender	M/F

PETS	Neutered/Spayed	Y/N

COLOR *(FT)*	PetID	Foreign Key
COLOR	Color	Character data, max-length 8.

RESERVATIONS *(PK)*	Reservation#	An 8 digit number uniquely identifying the reservation of stay, assigned by Pampered Pets, not based on SSN or other ID.
RESERVATIONS	TimeIn	Five character data, consisting of four digits with a ":" as the third character (12:11).
RESERVATIONS	TimeOut	Five character data, consisting of four digits with a ":" as the third character (12:11).
RESERVATIONS	DateOfCheckIn	Date.
RESERVATIONS	DateOfCheckOut	Date.
RESERVATIONS	Price	Decimal (8,2).
RESERVATIONS	ExtraDayCharge	Smallint, derived from TimeCheckout and DailyCost to add additional charge to bill if pet is checked out after 12PM.
RESERVATIONS	StaffID	Foreign Key

RESERVATION_DETAILS *(PK)*	Reservation#	An 8 digit number uniquely identifying the reservation of stay, assigned by Pampered Pets, not based on SSN or other ID.
RESERVATION_DETAILS	PetID	The letter "C" or "D" to identify the data as pertaining to either a cat or a dog respectively followed by 8 consecutive digits to uniquely identify each pet (example C89714589 for a cat, and D89714512) number string after the

		character C or D may not be re-used for the other type of pet not previously used.
RESERVATION_DETAILS	Vaccinations	Character Data. If all vaccinations have been administered and are still valid, this value will show as true “Y”. If not, then this value will show as false “N”.
RESERVATION_DETAILS	Bath	Character Data. The customer will specify if they want their pet to be bathed while staying with the company. Values of Yes “Y” or No “N”.
RESERVATION_DETAILS	NumMedGiven	Integer. To show how many medications were administered to the pet while staying there.

PET_MEDICATIONS *(PK)*	TimeAdministered	Character data, max- length 5. Two digit numbers followed by a colon with two more digits, describes the time the medication was administered.
PET_MEDICATIONS	Date	Date.
PET_MEDICATIONS	PetID	Foreign Key.
PET_MEDICATIONS	Reservation#	Foreign Key.

STAFF *(PK)*	StaffID	Character data, max-length 9. The letter S plus 8 digit number uniquely identifying the staff, assigned by Pampered Pets, not based on SSN or other ID.
STAFF	EmployeeType	(Required), character data, max-length 30 to specify what the staff position is in the company.
STAFF	FName	(Required), character data, max-length 30.
STAFF	LName	(Required), character data, max-length 30.
STAFF	Street	Character and Digit, max-length 30.
STAFF	City	Character, max-length 30.

STAFF	Zip	5 digit number unique to a section of the U.S.
STAFF	State	Character data, max-length 20.
STAFF	Country	Character data, max-length 20.
STAFF	Phone	10 digit number, customer phone number.
STAFF	PayRate	$35.00
STAFF	StartDate	Date
STAFF	AccountNumber	Decimal (12,0). Bank Account Number.
STAFF	RoutingNumber	Decimal (12,0). Bank Account Routing Number.
STAFF	FederalTaxDeduction	Decimal (3,2).
STAFF	StateTaxDeduction	Decimal (3,2).

FULL_TIMERS *(FT)*	StaffID	Foreign Key.
FULL_TIMERS	MedicalDeduction	Decimal (7,2).
FULL_TIMERS	DentalDeduction	Decimal (7,2).
FULL_TIMERS	DisabilityDeduction	Decimal (7,2).
FULL_TIMERS	LifeInsuranceDeduction	Decimal (7,2).

RECORD_KEEPERS (*RK*)	StaffID	Foreign Key.
RECORD_KEEPERS	DegreeObtained	Character data, max-length 30, describes and identifies the schooling the record keeper has completed, typically bachelors or masters.
RECORD_KE	GradDate	Date.

EPERS		
RECORD_KEEPERS	YrsOfExperience	Tinyint, representing years of experience.

BORARDING_STAFF (*FK*)	StaffID	Foreign Key.
BORARDING_STAFF	References	Character data, max-length 20, name of individual who referred the part-time employee (if applicable).

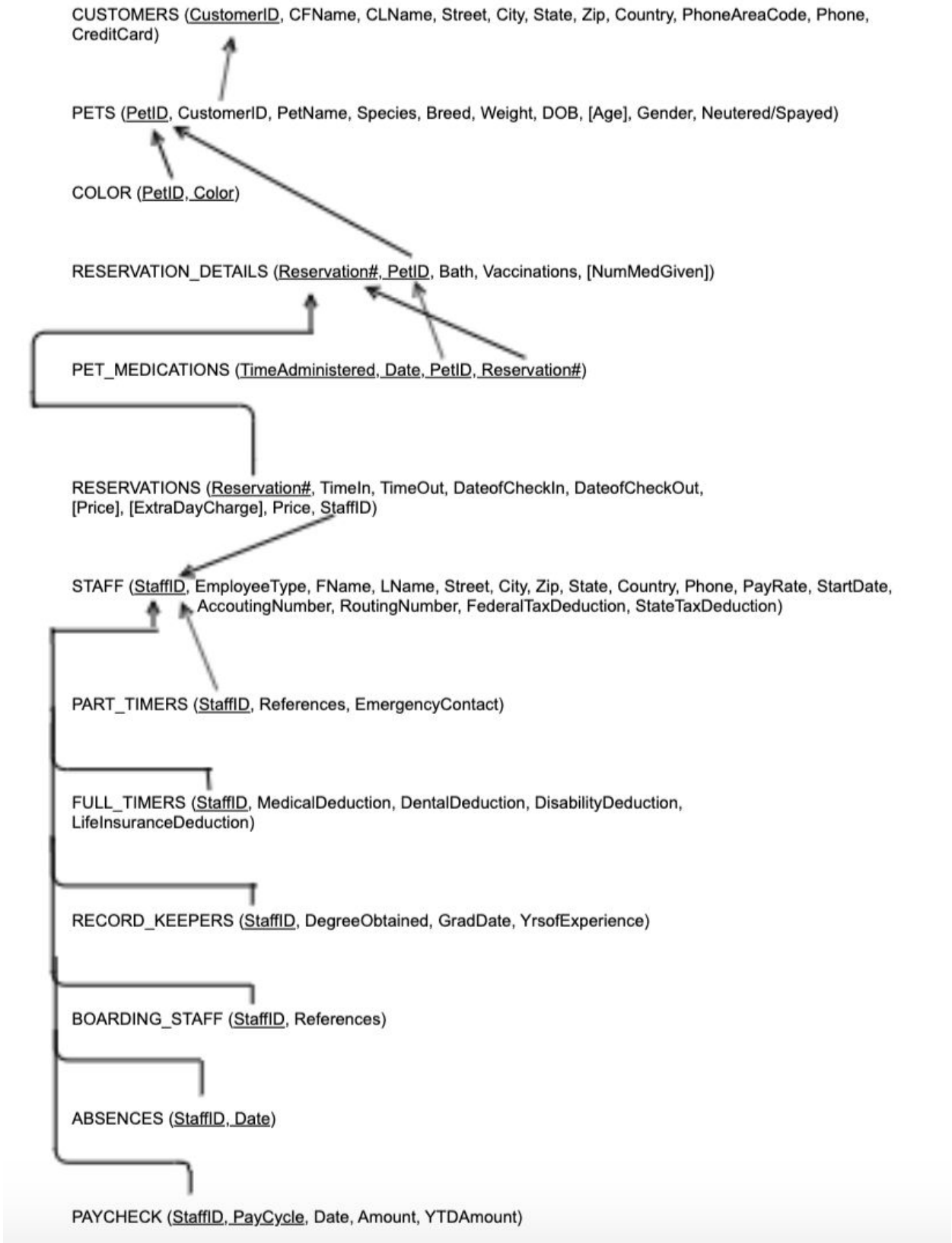
PART_TIMERS (*FK*)	StaffID	Foreign Key.
PART_TIMERS	RFName	Character data, max-length 50, first name of individual who referred the part-time employee (if applicable).
PART_TIMERS	RLName	Character data, max-length 50, last name of individual who referred the part-time employee (if applicable).
PART_TIMERS	EmergencyContact	Character data, max-length 50, name of individual who will be called in case of an emergency involving the part-time employee.
PART_TIMERS	ECPhone	Digit data, max-length 10, phone number of individual who will be called in case of an emergency involving the part-time employee.

ABSENCES (*FK*)	StaffID	Foreign Key.
ABSENCES	Date	Date.

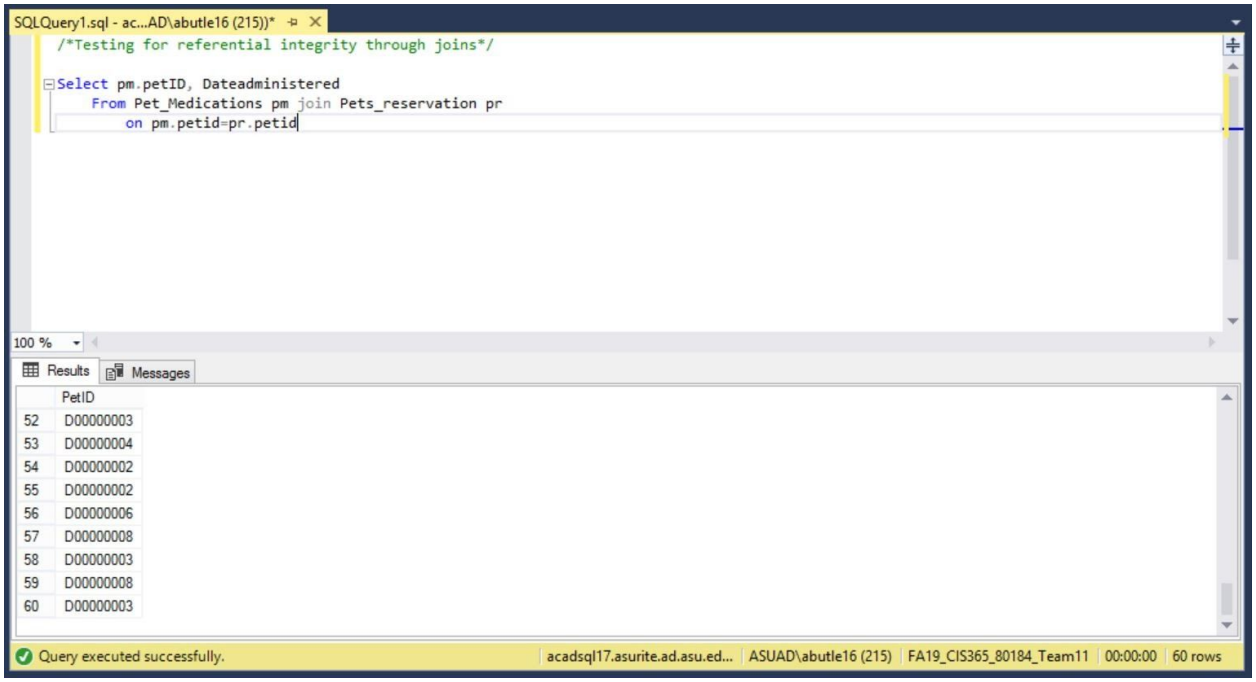
PAYCHECK (*PK*)	StaffID	Foreign Key.
PAYCHECK	PayCycle	BigInt. first, second, third… pay periods

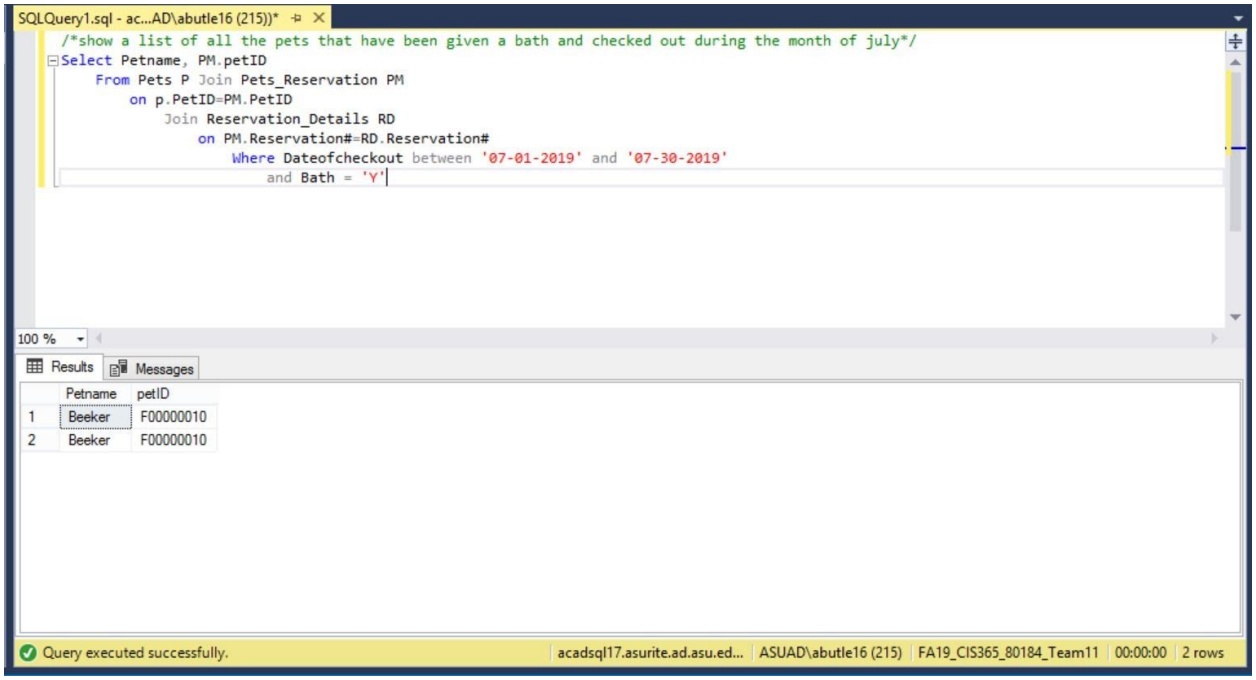
PAYCHECK	Date	Date.
PAYCHECK	Amount	Decimal (8,2). To signify pay amount for the chosen pay period.
PAYCHECK	YTDSalary	Decimal (8,2). Sum of all paycheck amounts per staff.

Database Schema



Database/ Integrity Testing & Results





SQLQuery1.sql - ac...AD\abutle16 (215))*

```
Select CFname, Petname
    From Customers C join Pets P
        On C.CustomerID=P.CustomerID
            Order by CFname
```

100 %

Results | Messages

	CFname	Petname
1	Abigail	Hamsta
2	Anthony	Ophelia
3	Aubrey	Polly
4	Barack	Omai
5	Carl	TiaoWen
6	Chloe	Elroy
7	Dominic	Dream
8	Elvis	Nacho
9	Emma	Chase
10	Frank	Swizzle

Query executed successfully. | acadsql17.asurite.ad.asu.ed... | ASUAD\abutle16 (215) | FA19_CIS365_80184_Team11 | 00:00:00 | 30 rows

SQLQuery1.sql - ac...AD\abutle16 (215))*

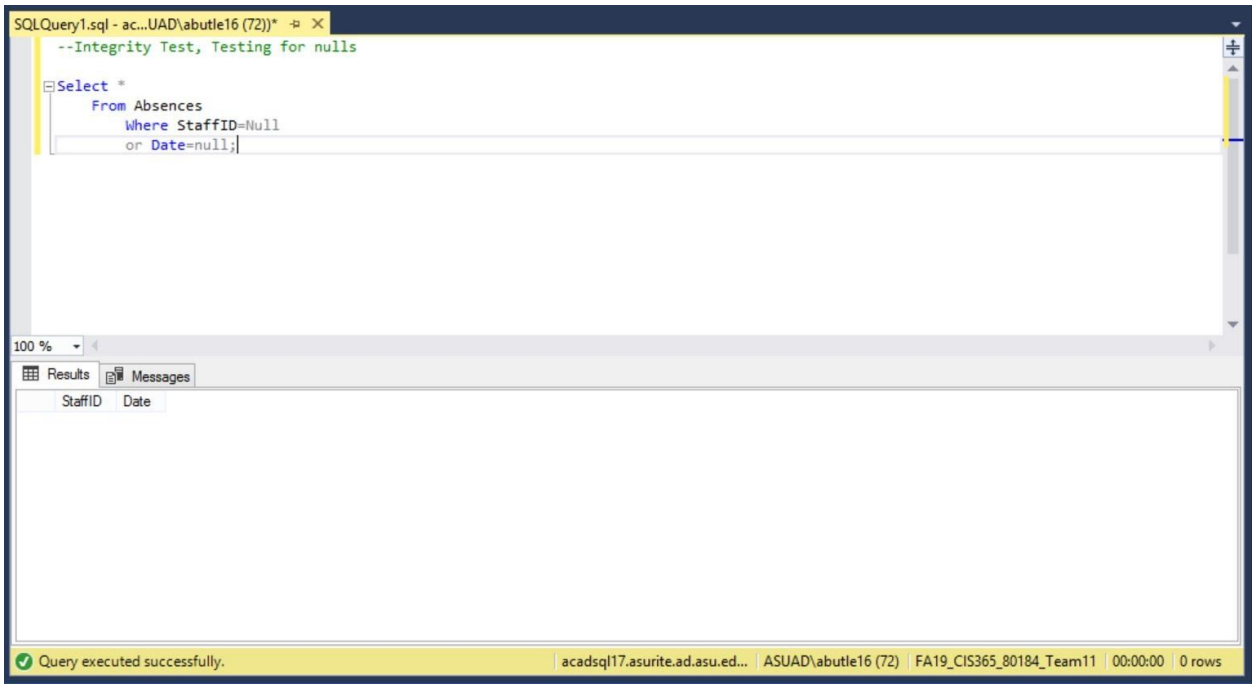
```
Select PR.PetID
    From Pets_Reservation PR Join Pet_Medications PM
        On PR.PetID = PM.PetID
```

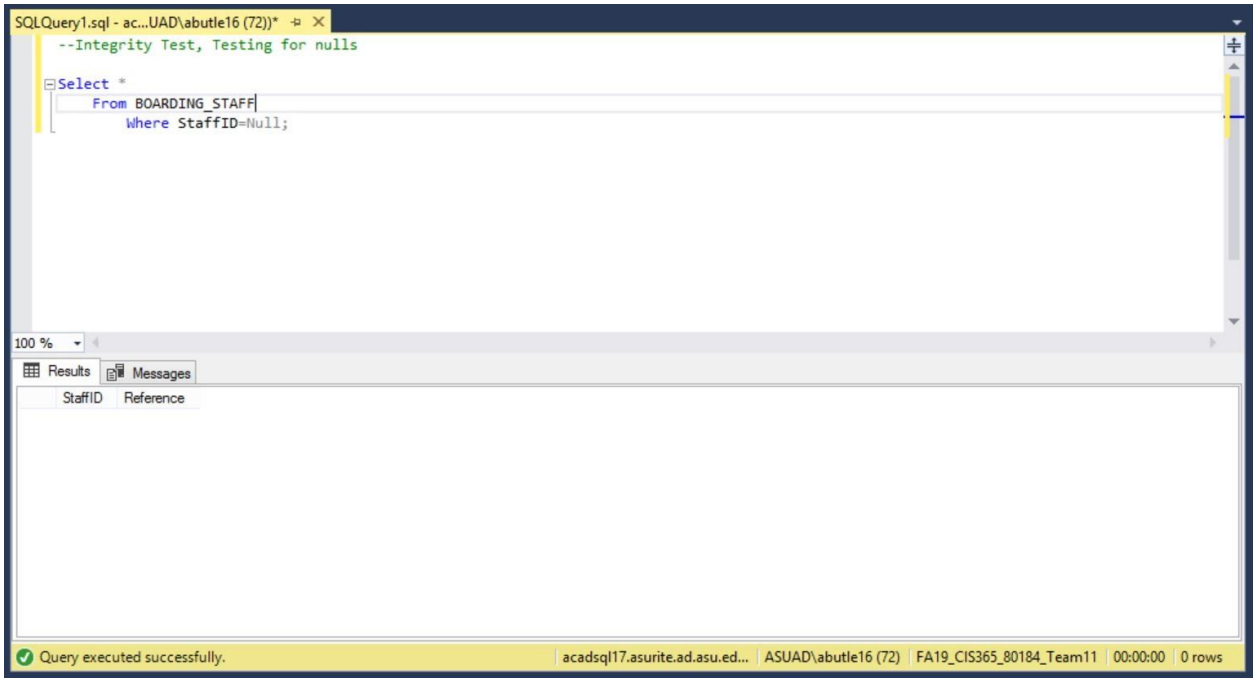
100 %

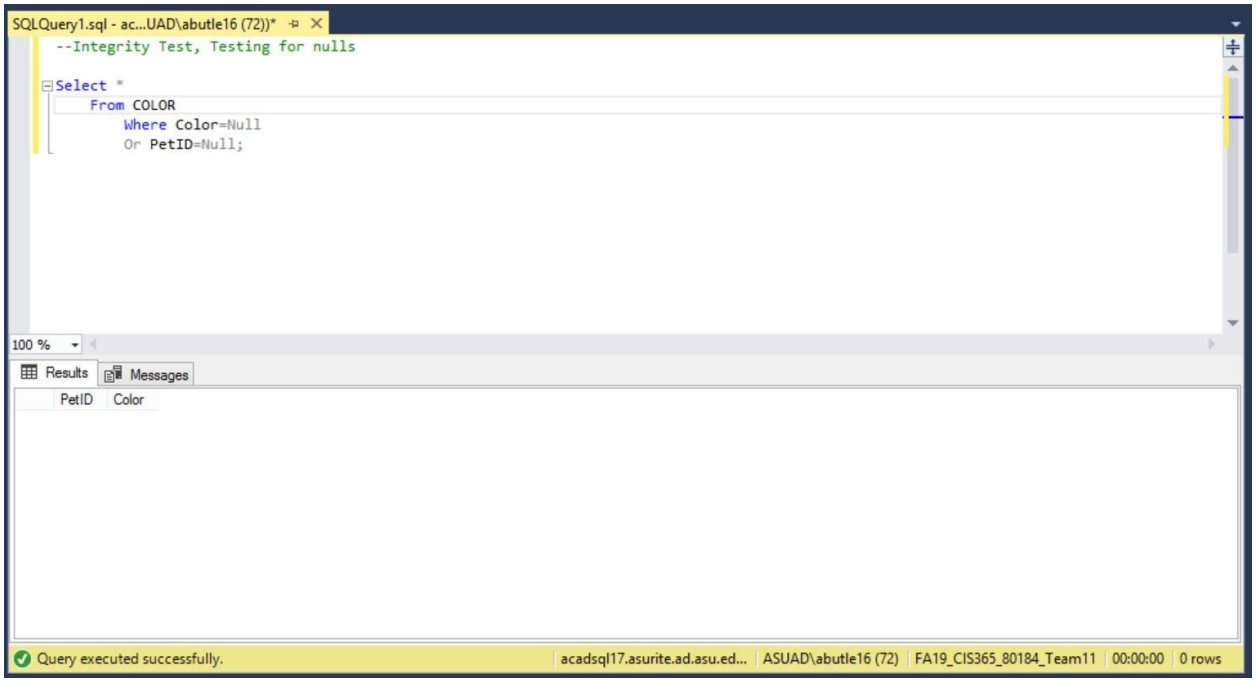
Results | Messages

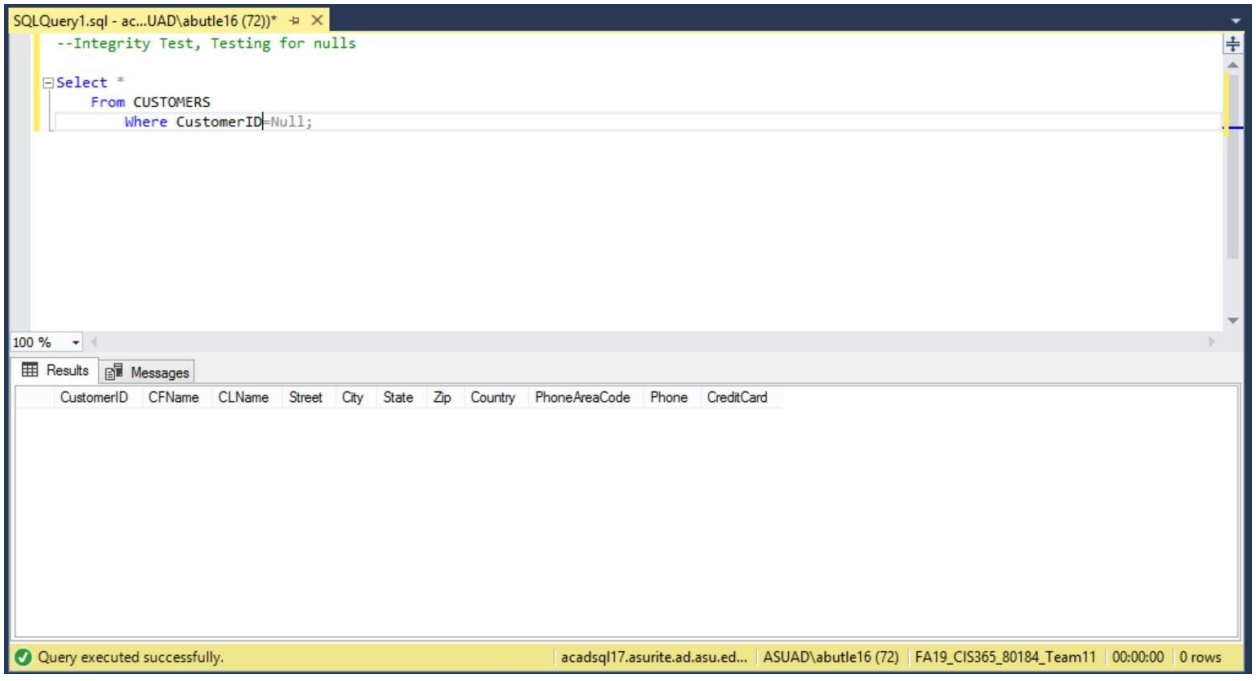
	PetID
1	F00000001
2	F00000001
3	F00000002
4	F00000002
5	F00000003
6	F00000005
7	F00000006
8	F00000006
9	F00000007
10	F00000008

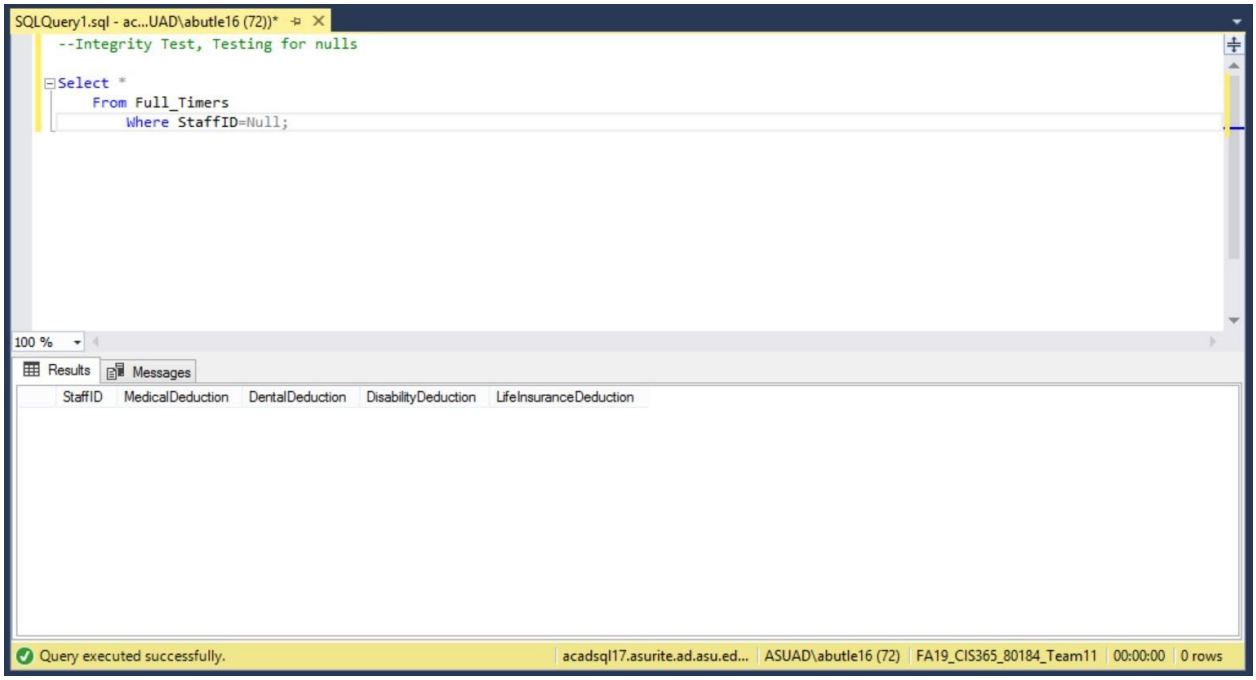
Query executed successfully. | acadsql17.asurite.ad.asu.ed... | ASUAD\abutle16 (215) | FA19_CIS365_80184_Team11 | 00:00:00 | 60 rows

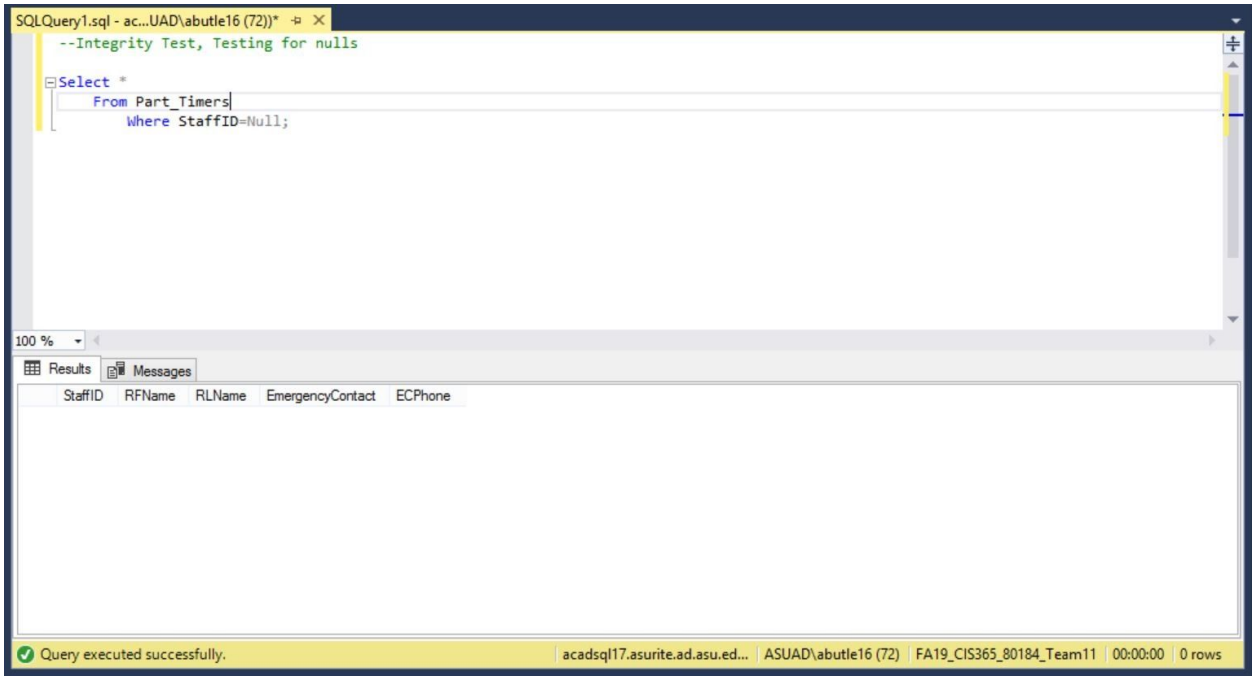

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From Absences
Where StaffID=Null
or Date=null;
100 %
Results
Messages
StaffID
Date
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

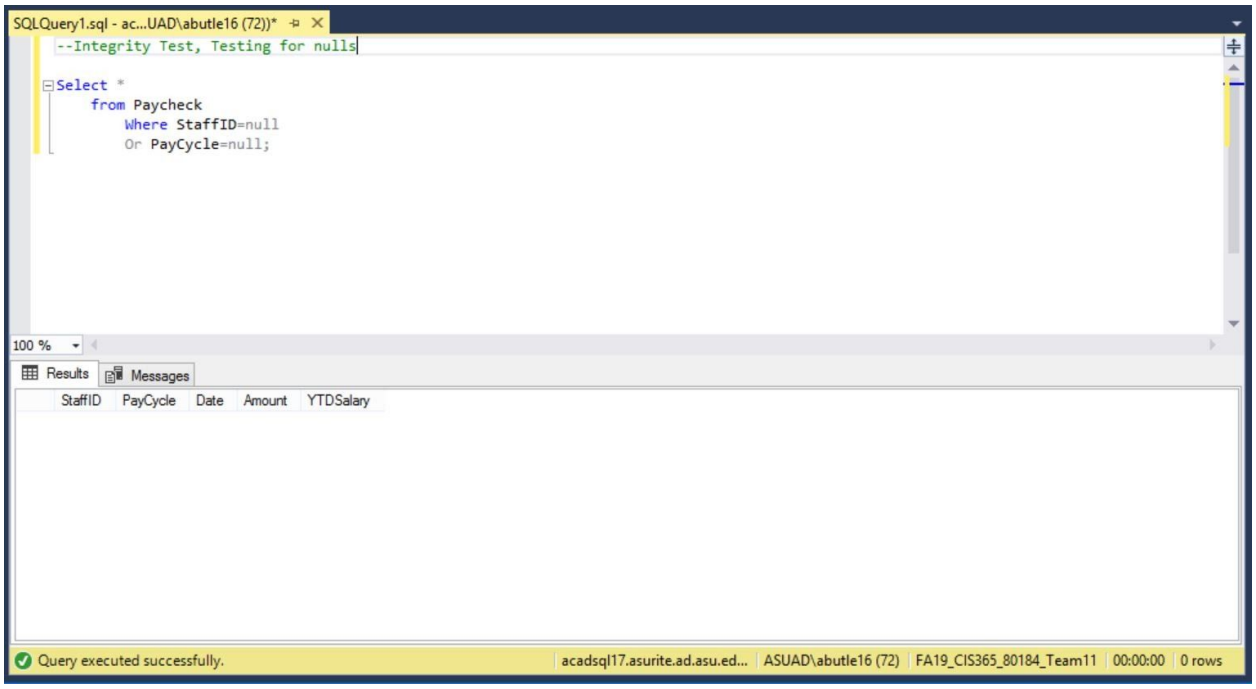

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From BOARDING_STAFF
Where StaffID=Null;
100 %
Results
Messages
StaffID
Reference
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

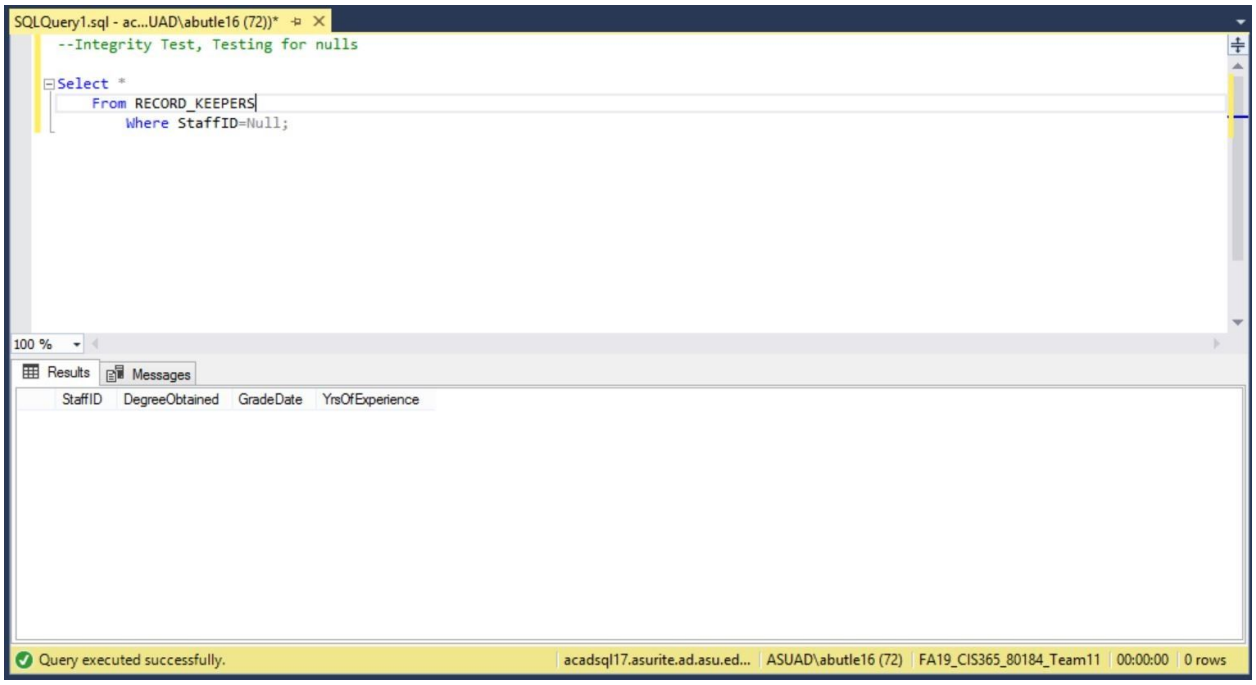

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From COLOR
Where Color=Null
Or PetID=Null;
100 %
Results
Messages
PetID
Color
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

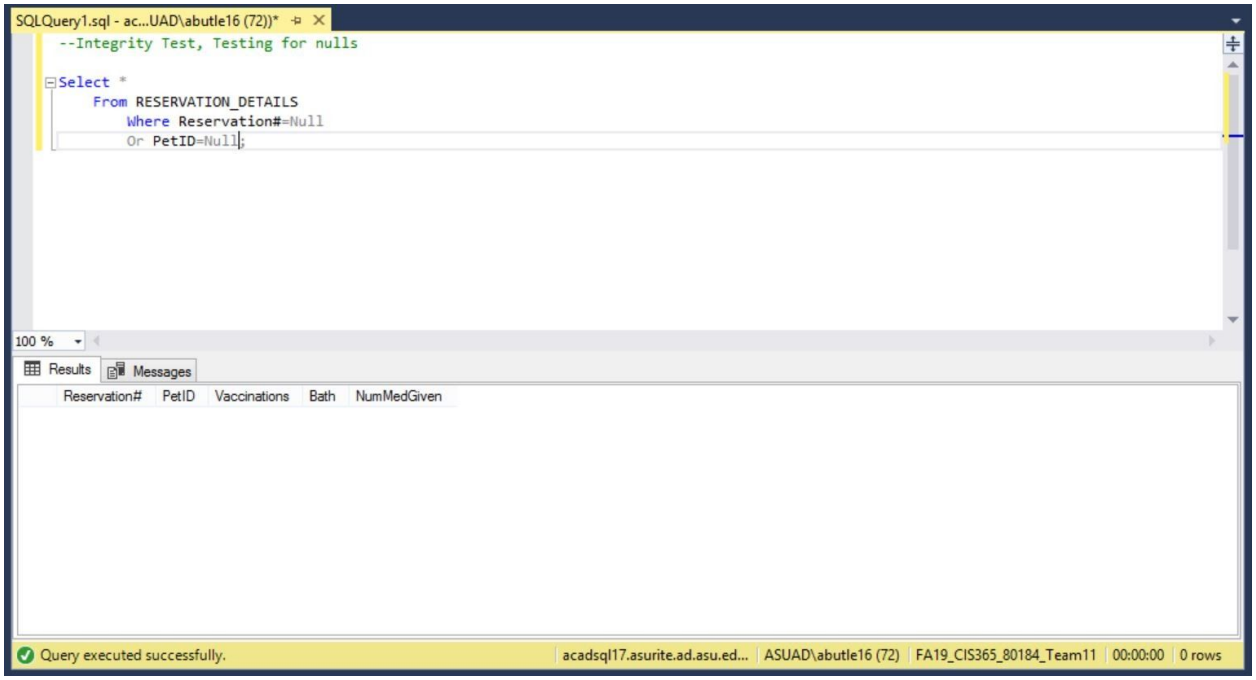

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From CUSTOMERS
Where CustomerID=Null;
100 %
Results
Messages
CustomerID
CFName
CLName
Street
City
State
Zip
Country
PhoneAreaCode
Phone
CreditCard
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

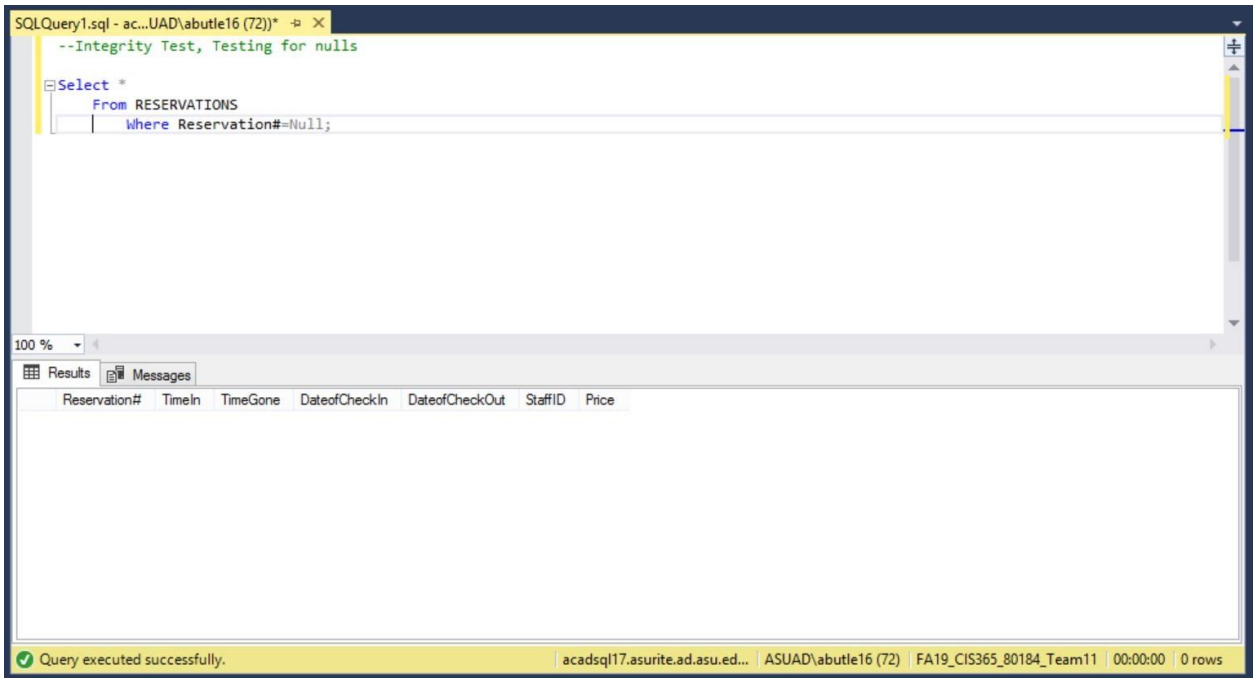

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From Full_Timers
Where StaffID=Null;
100 %
Results
Messages
StaffID
MedicalDeduction
DentalDeduction
DisabilityDeduction
LifeInsuranceDeduction
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

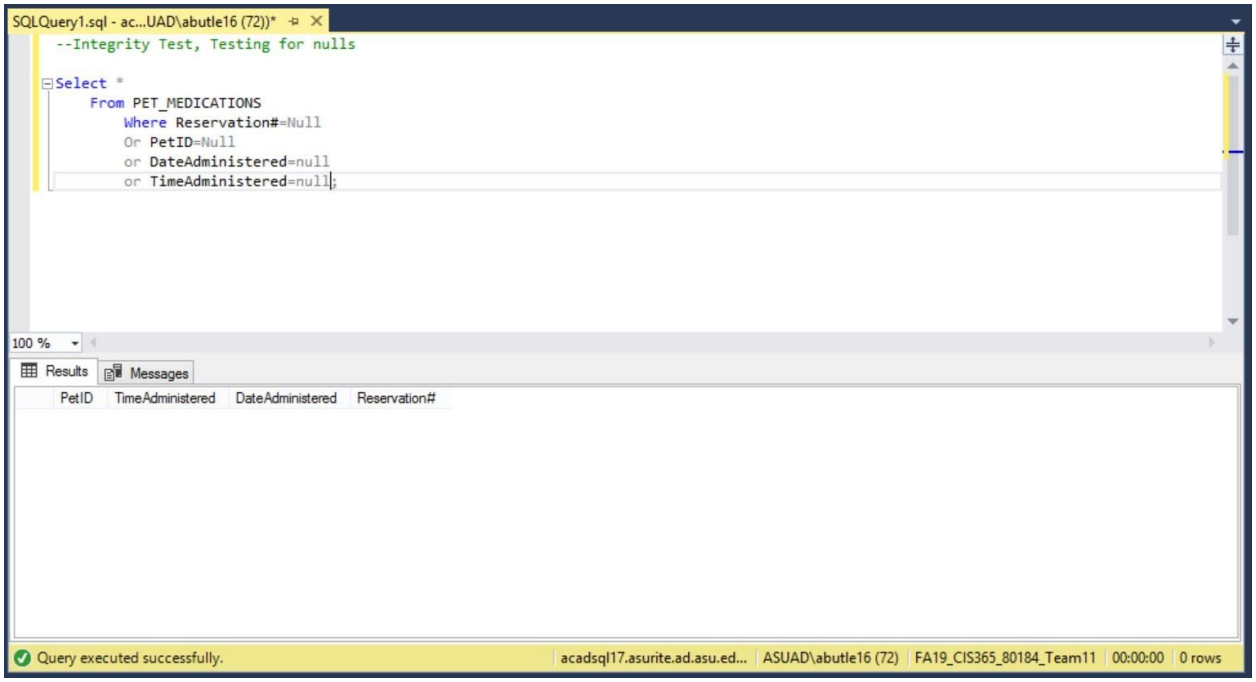

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From Part_Timers
Where StaffID=Null;
100 %
Results
Messages
StaffID
RFName
RLName
EmergencyContact
ECPhone
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

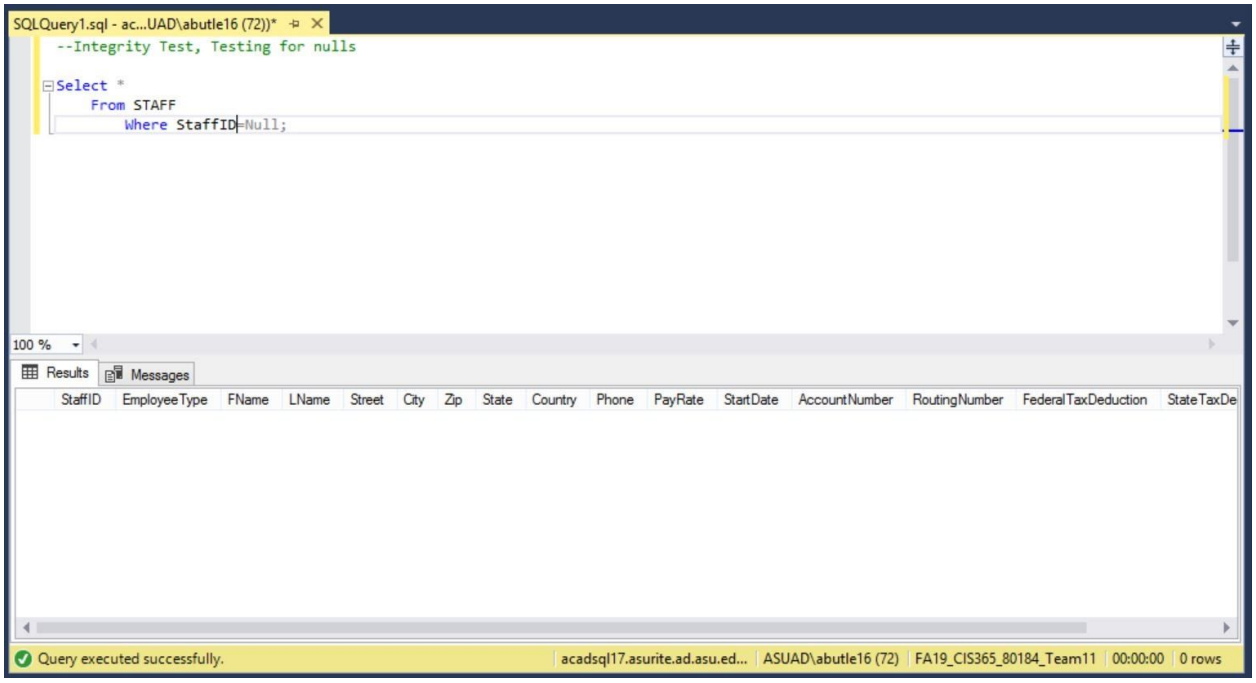

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
from Paycheck
Where StaffID=null
Or PayCycle=null;
100 %
Results
Messages
StaffID
PayCycle
Date
Amount
YTDSalary
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

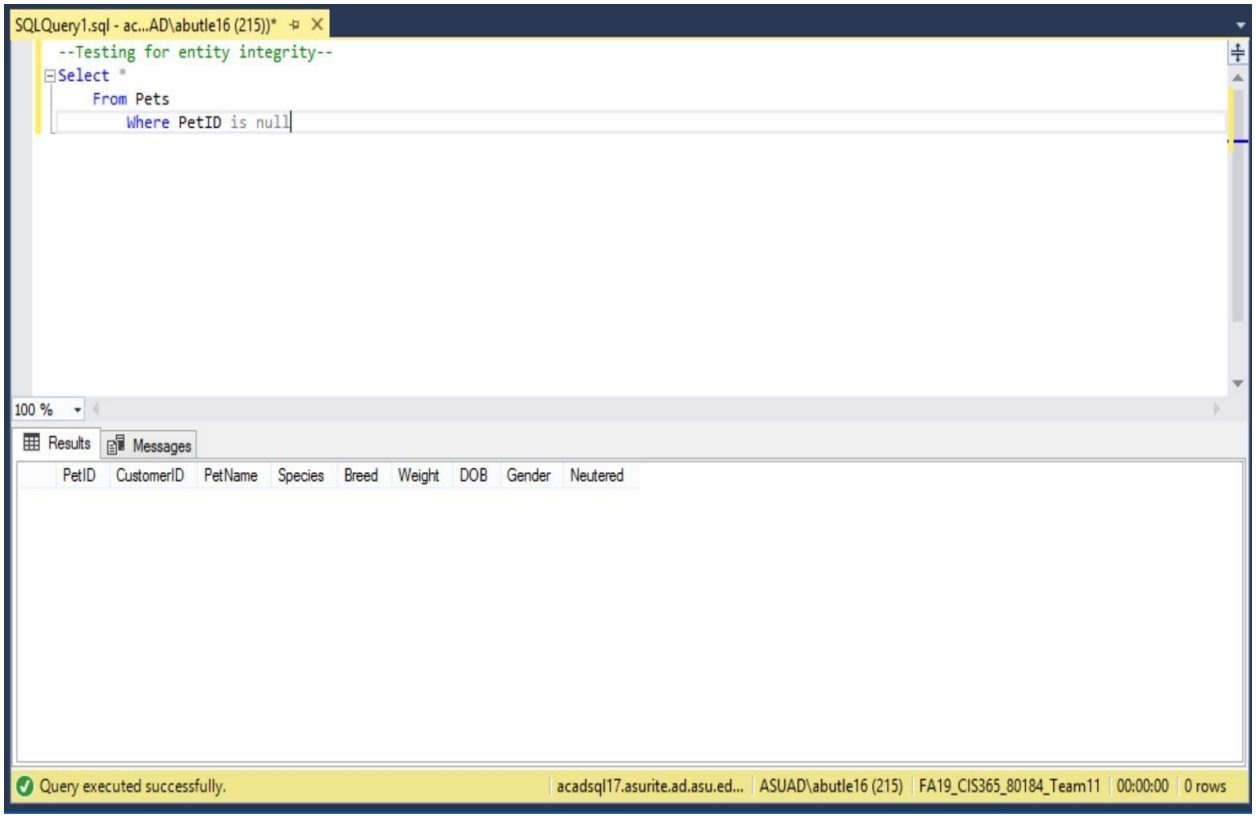

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From RECORD_KEEPERS
Where StaffID=Null;
100 %
Results
Messages
StaffID
DegreeObtained
GradeDate
YrsOfExperience
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows


SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From RESERVATION_DETAILS
Where Reservation#=Null
Or PetID=Null;
100 %
Results
Messages
Reservation#
PetID
Vaccinations
Bath
NumMedGiven
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows

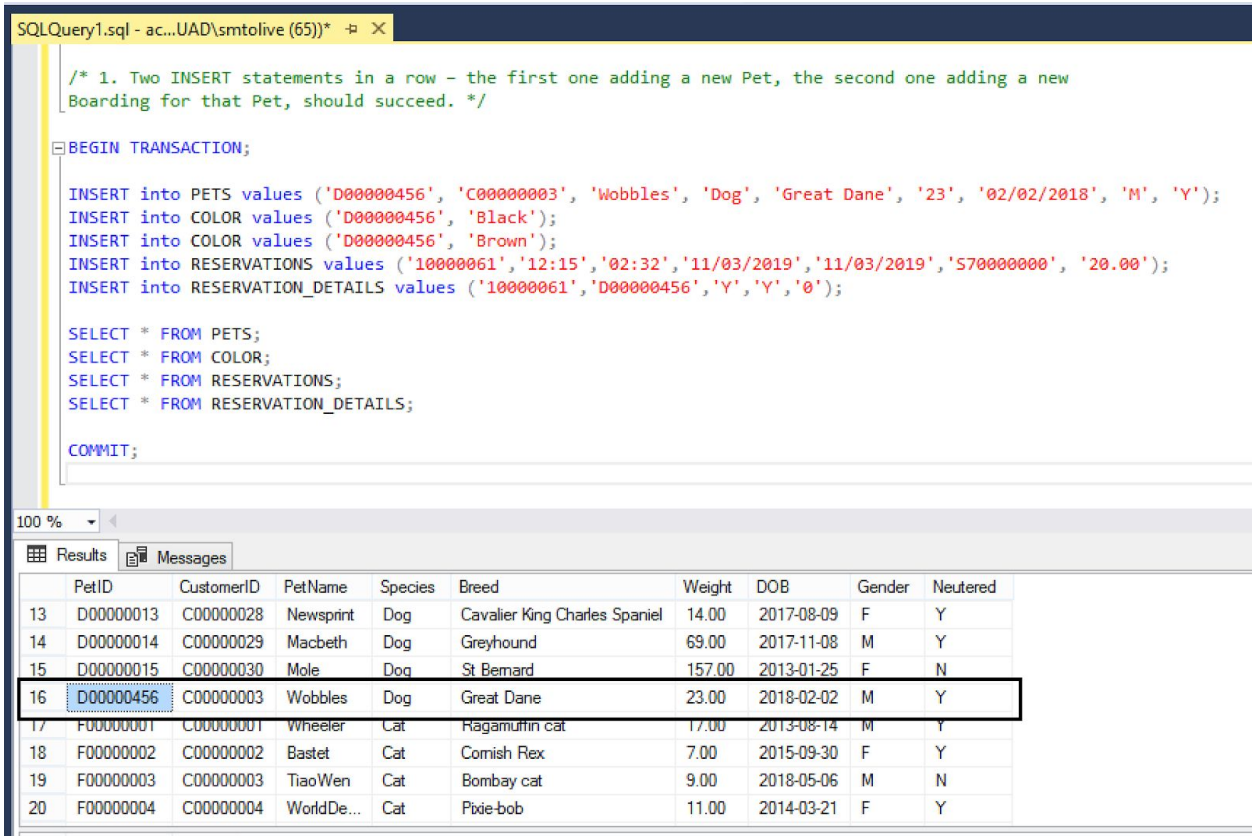

SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From RESERVATIONS
Where Reservation#=Null;
100 %
Results
Messages
Reservation#
TimeIn
TimeGone
DateofCheckIn
DateofCheckOut
StaffID
Price
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows


SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From PET_MEDICATIONS
Where Reservation#=Null
Or PetID=Null
or DateAdministered=null
or TimeAdministered=null;
100 %
Results
Messages
PetID
TimeAdministered
DateAdministered
Reservation#
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows


SQLQuery1.sql - ac...UAD\abutle16 (72))*
--Integrity Test, Testing for nulls
Select *
From STAFF
Where StaffID=Null;
100 %
Results
Messages
StaffID
EmployeeType
FName
LName
Street
City
Zip
State
Country
Phone
PayRate
StartDate
AccountNumber
RoutingNumber
FederalTaxDeduction
StateTaxDe
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (72)
FA19_CIS365_80184_Team11
00:00:00
0 rows


SQLQuery1.sql - ac...AD\abutle16 (215))*
--Testing for entity integrity--
Select *
From Pets
Where PetID is null
100 %
Results
Messages
PetID
CustomerID
PetName
Species
Breed
Weight
DOB
Gender
Neutered
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (215)
FA19_CIS365_80184_Team11
00:00:00
0 rows

Basic Queries

1. 

```
/* 1. Two INSERT statements in a row - the first one adding a new Pet, the second one adding a new
Boarding for that Pet, should succeed. */

BEGIN TRANSACTION;

INSERT into PETS values ('D00000456', 'C00000003', 'Wobbles', 'Dog', 'Great Dane', '23', '02/02/2018', 'M', 'Y');
INSERT into COLOR values ('D00000456', 'Black');
INSERT into COLOR values ('D00000456', 'Brown');
INSERT into RESERVATIONS values ('10000061','12:15','02:32','11/03/2019','11/03/2019','S70000000', '20.00');
INSERT into RESERVATION_DETAILS values ('10000061','D00000456','Y','Y','0');

SELECT * FROM PETS;
SELECT * FROM COLOR;
SELECT * FROM RESERVATIONS;
SELECT * FROM RESERVATION_DETAILS;

COMMIT;
```

100 %

Results | Messages

	PetID	CustomerID	PetName	Species	Breed	Weight	DOB	Gender	Neutered
13	D00000013	C00000028	Newsprint	Dog	Cavalier King Charles Spaniel	14.00	2017-08-09	F	Y
14	D00000014	C00000029	Macbeth	Dog	Greyhound	69.00	2017-11-08	M	Y
15	D00000015	C00000030	Mole	Dog	St Bernard	157.00	2013-01-25	F	N
16	D00000456	C00000003	Wobbles	Dog	Great Dane	23.00	2018-02-02	M	Y
17	F00000001	C00000001	Wheeler	Cat	Ragamuffin cat	17.00	2013-08-14	M	Y
18	F00000002	C00000002	Bastet	Cat	Cornish Rex	7.00	2015-09-30	F	Y
19	F00000003	C00000003	TiaoWen	Cat	Bombay cat	9.00	2018-05-06	M	N
20	F00000004	C00000004	WorldDe...	Cat	Pixie-bob	11.00	2014-03-21	F	Y

(1. 2ND RESULT)

	PetID	Color
30	D00000015	White
31	D00000456	Black
32	D00000456	Bro...
33	F00000001	Grey
34	F00000001	White
35	F00000002	Back
36	F00000002	Black
37	F00000003	Bro...

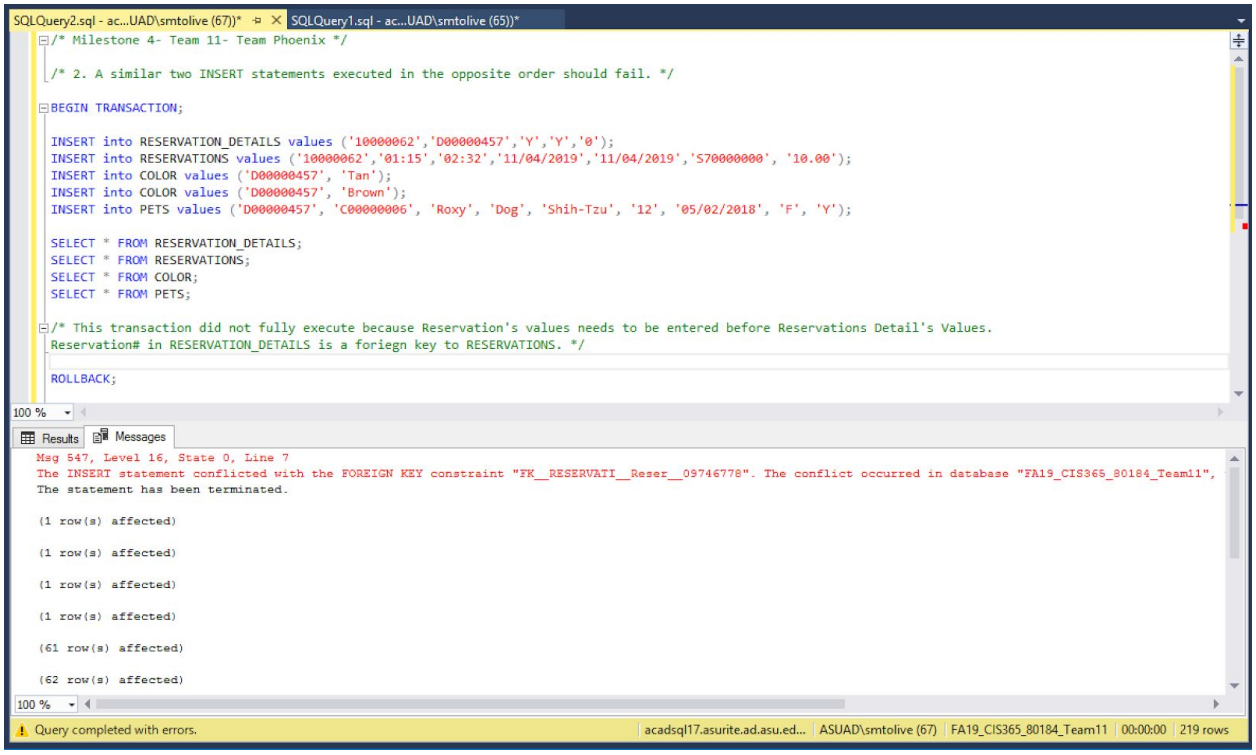
(1. 3RD RESULT)

	Reservation#	TimeIn	TimeGone	DateofCheckIn	DateofCheckOut	StaffID
54	10000054	11:45	13:13	2019-01-23	2019-01-26	S40000000
55	10000055	11:13	13:34	2019-09-30	2019-10-04	S50000000
56	10000056	16:30	17:56	2019-08-01	2019-08-04	S80000000
57	10000057	11:18	13:44	2019-02-04	2019-02-05	S30000000
58	10000058	15:00	16:07	2019-09-28	2019-09-30	S40000000
59	10000059	13:29	15:15	2019-08-11	2019-08-14	S80000000
60	10000060	10:57	16:41	2019-01-14	2019-01-18	S50000000
61	10000061	12:15	02:32	2019-11-03	2019-11-03	S70000000

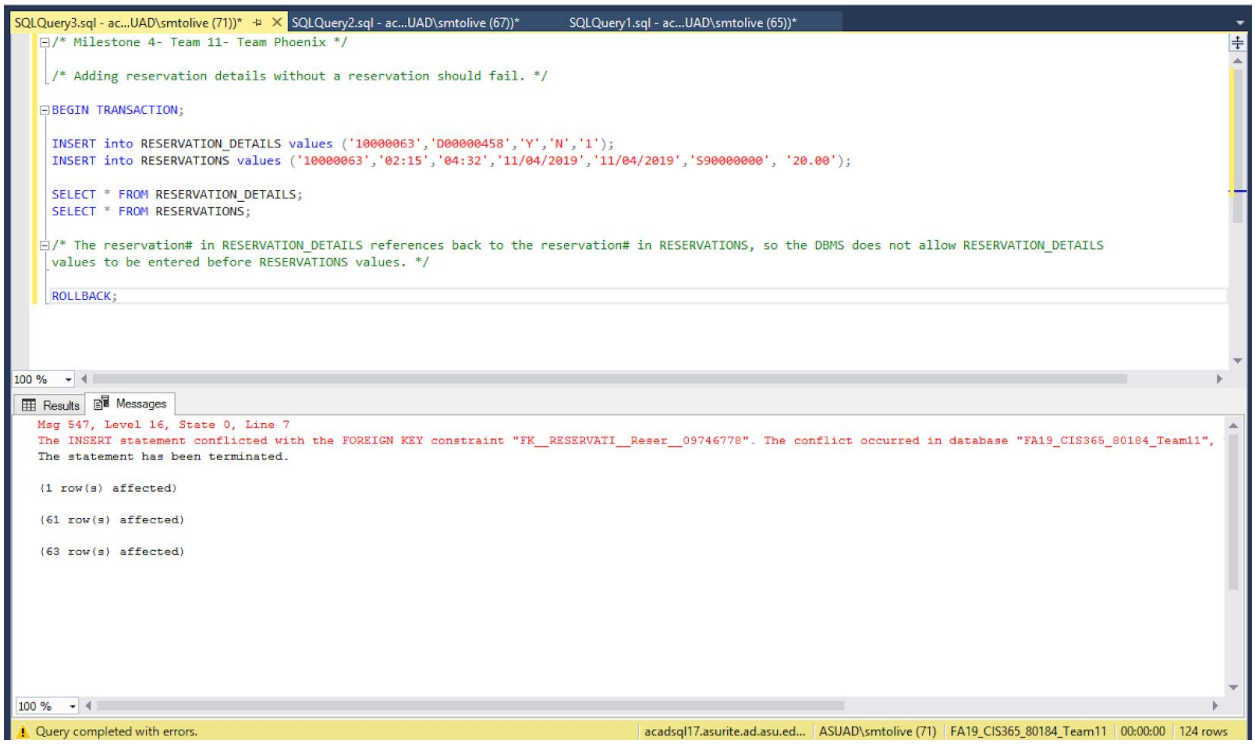
(1. 4TH RESULT)

	Reservation#	PetID	Vaccinations	Bath	NumMedGiven
54	10000054	D000000...	Y	N	4
55	10000055	D000000...	Y	Y	3
56	10000056	D000000...	Y	N	2
57	10000057	D000000...	Y	N	1
58	10000058	D000000...	Y	N	1
59	10000059	D000000...	Y	Y	1
60	10000060	D000000...	Y	N	0
61	10000061	D000004...	Y	Y	0

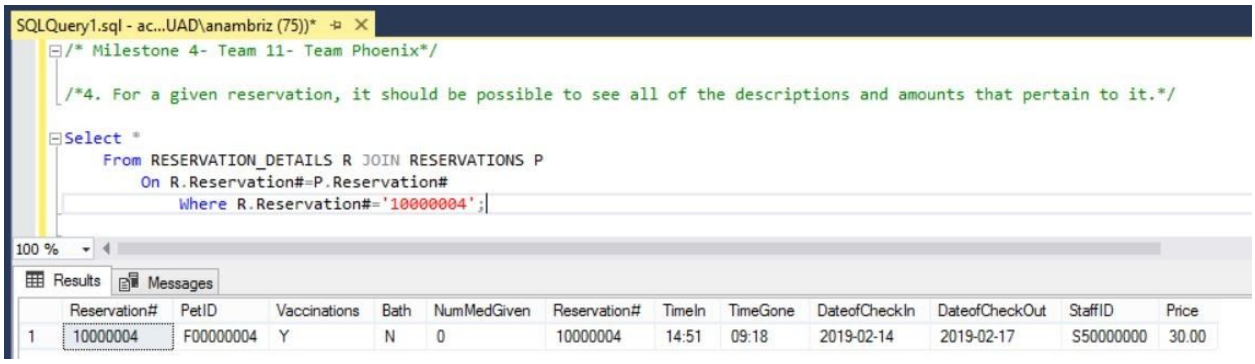
2.



3.


SQLQuery3.sql - ac...UAD\smtolive (71))*
SQLQuery2.sql - ac...UAD\smtolive (67))*
SQLQuery1.sql - ac...UAD\smtolive (65))*
/* Milestone 4- Team 11- Team Phoenix */
/* Adding reservation details without a reservation should fail. */
BEGIN TRANSACTION;
INSERT into RESERVATION_DETAILS values ('10000063','D00000458','Y','N','1');
INSERT into RESERVATIONS values ('10000063','02:15','04:32','11/04/2019','11/04/2019','S90000000', '20.00');
SELECT * FROM RESERVATION_DETAILS;
SELECT * FROM RESERVATIONS;
/* The reservation# in RESERVATION_DETAILS references back to the reservation# in RESERVATIONS, so the DBMS does not allow RESERVATION_DETAILS
values to be entered before RESERVATIONS values. */
ROLLBACK;
100 %
Results
Messages
Msg 547, Level 16, State 0, Line 7
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__RESERVATI__Reser__09746778". The conflict occurred in database "FA19_CIS365_80184_Team11",
The statement has been terminated.
(1 row(s) affected)
(61 row(s) affected)
(63 row(s) affected)
100 %
Query completed with errors.
acadsql17.asurite.ad.asu.ed...
ASUAD\smtolive (71)
FA19_CIS365_80184_Team11
00:00:00
124 rows

4.


SQLQuery1.sql - ac...UAD\anambriz (75))*
/* Milestone 4- Team 11- Team Phoenix*/
/*4. For a given reservation, it should be possible to see all of the descriptions and amounts that pertain to it.*/
Select *
From RESERVATION_DETAILS R JOIN RESERVATIONS P
On R.Reservation#=P.Reservation#
Where R.Reservation#='10000004';
100 %
Results
Messages
Reservation# PetID Vaccinations Bath NumMedGiven Reservation# TimeIn TimeGone DateofCheckIn DateofCheckOut StaffID Price
1 10000004 F00000004 Y N 0 10000004 14:51 09:18 2019-02-14 2019-02-17 S50000000 30.00

5.


SQLQuery3.sql - ac...UAD\anambriz (53))*
SQLQuery2.sql - ac...AD\anambriz (107))*
SQLQuery1.sql - ac...AD\anambriz (100))*
/*Milestone 4-Team 11-Team Phoennix*/
/*5. Trying to delete a Pet who has past on should fail*/
begin transaction;
delete from PETS where PetID ='F00000001'
select * from pets
rollback;
100 %
Messages
Msg 547, Level 16, State 0, Line 2
The DELETE statement conflicted with the REFERENCE constraint "FK__PET_MEDIC__PetID__224015A2". The conflict occurred in database "FA19_CIS365_80184_Team11", table "dbo.PET_MEDICATIONS", column 'PetID'.
The statement has been terminated.

```
SQLQuery5.sql - ac...UAD\anambriz (75))*   SQLQuery4.sql - ac...UAD\anambriz (65))*

select * from STAFF;
```

100 %

Results | Messages

	StaffID	EmployeeType	FName	LName	Street	City	Zip	State	Country	Phone	PayRate	StartDate	AccountNumber	RoutingNumber	FederalTaxDeduction	StateTaxDeduction
1	S10000000	recordkeeper	Shelby	Johnson	1234 S. Side Trl.	Anthem	85086	AZ	US	6029202277	12.50	2018-12-12	22223333	44445678	0.06	0.05
2	S11000000	accountant	Abraham	Mohammad	3390 Here to There Blvd.	Surprise	12234	AZ	US	623078999	11.00	2015-04-18	89898673	12677793	0.08	0.03
3	S12000000	accountant	David	Rajkstan	2345 Rockley	Surprise	56772	AZ	US	6230246629	11.00	2014-08-13	89838673	78544893	0.06	0.06
4	S13000000	accountant	Romand	Ramada	40408 Marco Polo Ln.	Glendale	33390	AZ	US	6020437890	11.00	2017-10-21	88838673	12904893	0.05	0.05
5	S20000000	recordkeeper	Ashley	Mackenzie	5678 N. Asher St.	Phoenix	85087	AZ	US	4909726689	12.50	2019-01-02	12900876	90875678	0.06	0.04
6	S30000000	boardingstaff	Alecxander	Ambriz	7880 Dead End Ln.	Mesa	82023	AZ	US	4802997168	11.00	2017-02-20	12349764	12897654	0.04	0.03
7	S40000000	boarding staff	Slim	Shady	7881 Dead End Ln.	Mesa	82023	AZ	US	5408990998	11.00	2017-02-02	89308673	12674893	0.05	0.03
8	S50000000	boarding staff	Tony	Stark	40409 N. Cross Timbers Trl.	Anthem	85087	AZ	US	4800000090	16.00	2014-01-01	89948673	12679893	0.04	0.03
9	S60000000	boarding staff	Haylie	Burg	72266 So Many Words St.	Phoenix	87765	AZ	US	7809986790	11.00	2019-09-09	89558673	12609763	0.07	0.03
10	S70000000	boarding staff	Matthew	Mackles	23469 Apple Rd.	Chandler	90128	AZ	US	4808888888	11.00	2018-12-31	56782773	126904893	0.06	0.03
11	S80000000	boarding staff	Jackson	Roberts	1114 South Side Serpents	Chandler	81007	AZ	US	4803333333	11.00	2018-11-29	89339876	16642893	0.05	0.03
12	S90000000	accountant	John	Reynolds	9100 W. Washer Ave.	Glendale	76953	AZ	US	4809997658	12.50	2019-01-08	8986473	12678893	0.05	0.06

6.

```
SQLQuery4.sql - ac...UAD\anambriz (61))*   SQLQuery2.sql - ac...UAD\anambriz (63))*   SQLQuery1.sql - ac...UAD\anambriz (60))*

/* Milestone 4- Team 11-Team Phoenix*/

/* Entering a new employee without any days absent from work should succeed.*/
begin transaction;
Insert into STAFF
 values ('S11110000', 'recordkeeper', 'Alyssa', 'Toliver', '3560 Sawgrass Dr.', 'Anthem', '85086', 'AZ', 'US', '6309853333', '12.50', '12-13-18','12345678','91123456','0.06','.05')

 select * from STAFF;
 rollback;
```

100 %

Results | Messages

	StaffID	EmployeeType	FName	LName	Street	City	Zip	State	Country	Phone	PayRate	StartDate	AccountNumber	RoutingNumber	FederalTaxDeduction	StateTaxDeduction
1	S10000000	recordkeeper	Shelby	Johnson	1234 S. Side Trl.	Anthem	85086	AZ	US	6029202277	12.50	2018-12-12	22223333	44445678	0.06	0.05
2	S11000000	accountant	Abraham	Mohammad	3390 Here to There Blvd.	Surprise	12234	AZ	US	623078999	11.00	2015-04-18	89898673	12677793	0.08	0.03
3	S11110000	recordkeeper	Alyssa	Toliver	3560 Sawgrass Dr.	Anthem	85086	AZ	US	6309853333	12.50	2018-12-13	12345678	91123456	0.06	0.05
4	S12000000	accountant	David	Rajkstan	2345 Rockley	Surprise	56772	AZ	US	6230246629	11.00	2014-08-13	89838673	78544893	0.06	0.06
5	S13000000	accountant	Romand	Ramada	40408 Marco Polo Ln.	Glendale	33390	AZ	US	6020437890	11.00	2017-10-21	88838673	12904893	0.05	0.05
6	S20000000	recordkeeper	Ashley	Mackenzie	5678 N. Asher St.	Phoenix	85087	AZ	US	4909726689	12.50	2019-01-02	12900876	90875678	0.06	0.04
7	S30000000	boardingstaff	Alecxander	Ambriz	7880 Dead End Ln.	Mesa	82023	AZ	US	4802997168	11.00	2017-02-20	12349764	12897654	0.04	0.03
8	S40000000	boarding staff	Slim	Shady	7881 Dead End Ln.	Mesa	82023	AZ	US	5408990998	11.00	2017-02-02	89308673	12674893	0.05	0.03
9	S50000000	boarding staff	Tony	Stark	40409 N. Cross Timbers Trl.	Anthem	85087	AZ	US	4800000090	16.00	2014-01-01	89948673	12679893	0.04	0.03
10	S60000000	boarding staff	Haylie	Burg	72266 So Many Words St.	Phoenix	87765	AZ	US	7809986790	11.00	2019-09-09	89558673	12609763	0.07	0.03
11	S70000000	boarding staff	Matthew	Mackles	23469 Apple Rd.	Chandler	90128	AZ	US	4808888888	11.00	2018-12-31	56782773	126904893	0.06	0.03
12	S80000000	boarding staff	Jackson	Roberts	1114 South Side Serpents	Chandler	81007	AZ	US	4803333333	11.00	2018-11-29	89339876	16642893	0.05	0.03
13	S90000000	accountant	John	Reynolds	9100 W. Washer Ave.	Glendale	76953	AZ	US	4809997658	12.50	2019-01-08	8986473	12678893	0.05	0.06

7.

```
SQLQuery1.sql - ac...UAD\abutle16 (75))*

-- (7.) adding a pet to the database, and putting in a customerID that does not exist in the database --

Insert into PETS Values ('F00000046','C00000089','Snowflake','Cat','Tabby','6.5','08/14/2009','F','Y');
```

100 %

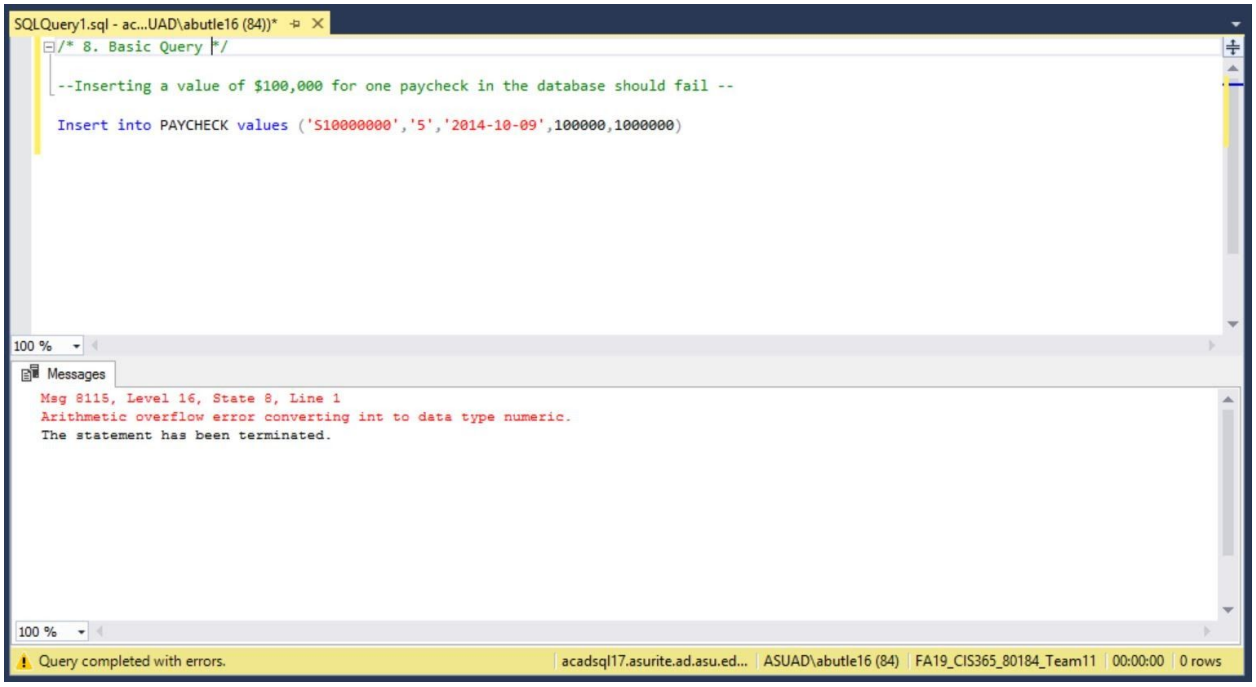
Messages

```
Msg 547, Level 16, State 0, Line 3
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__PETS__CustomerID__151B244E". The conflict occurred in database "FA19_
The statement has been terminated.
```

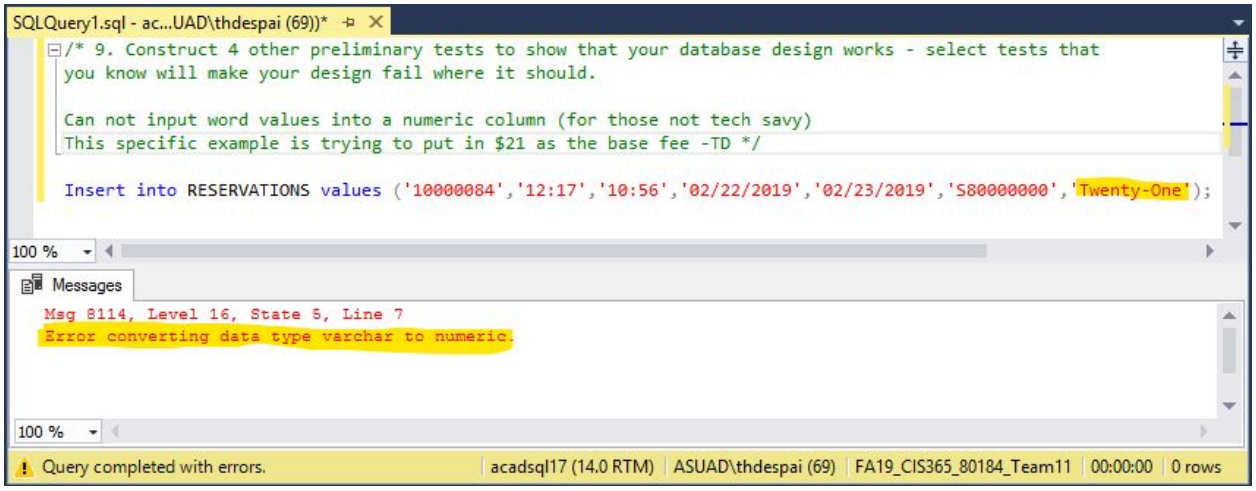
100 %

Query completed with errors. | acadsql17.asurite.ad.asu.ed... | ASUAD\abutle16 (75) | FA19_CIS365_80184_Team11 | 00:00:00 | 0 rows

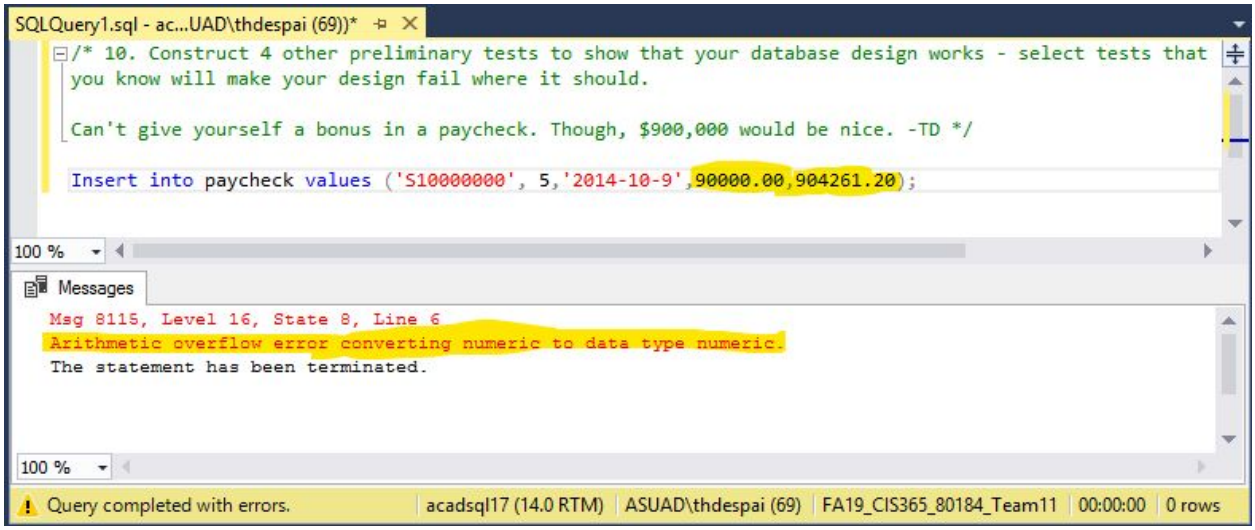
8.


SQLQuery1.sql - ac...UAD\abutle16 (84))*
/* 8. Basic Query */
--Inserting a value of $100,000 for one paycheck in the database should fail --
Insert into PAYCHECK values ('S10000000','5','2014-10-09',100000,1000000)
100 %
Messages
Msg 8115, Level 16, State 8, Line 1
Arithmetic overflow error converting int to data type numeric.
The statement has been terminated.
100 %
Query completed with errors.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (84)
FA19_CIS365_80184_Team11
00:00:00
0 rows

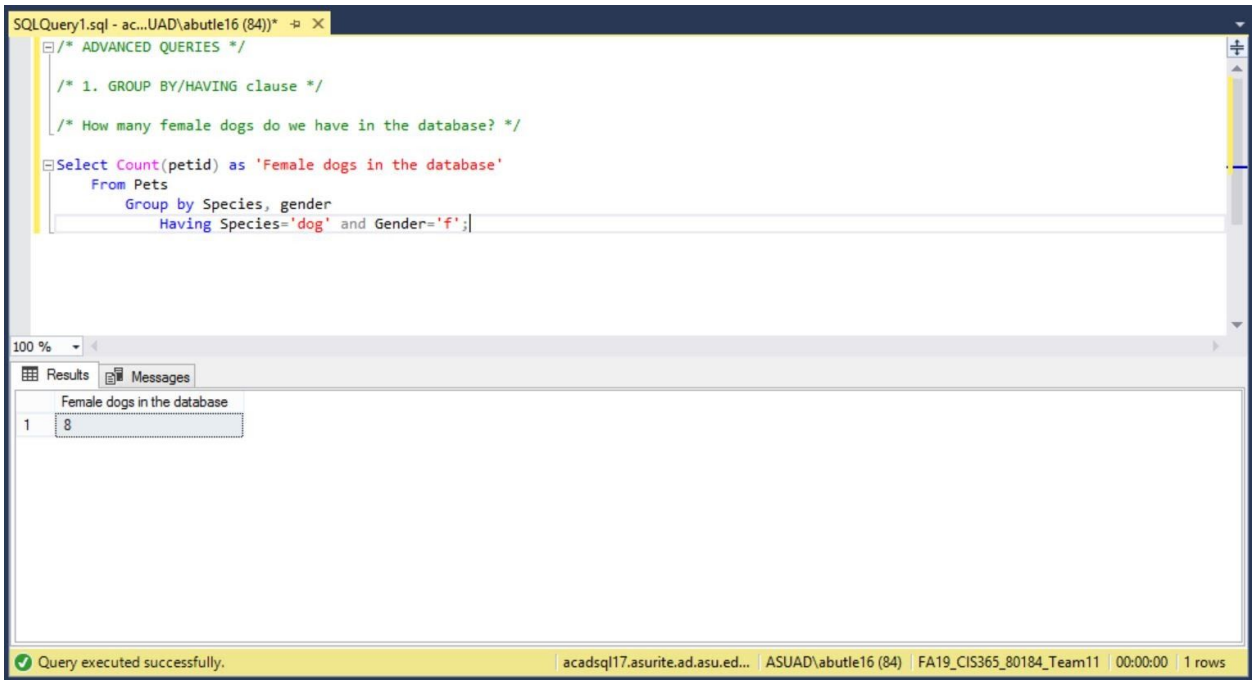
9.

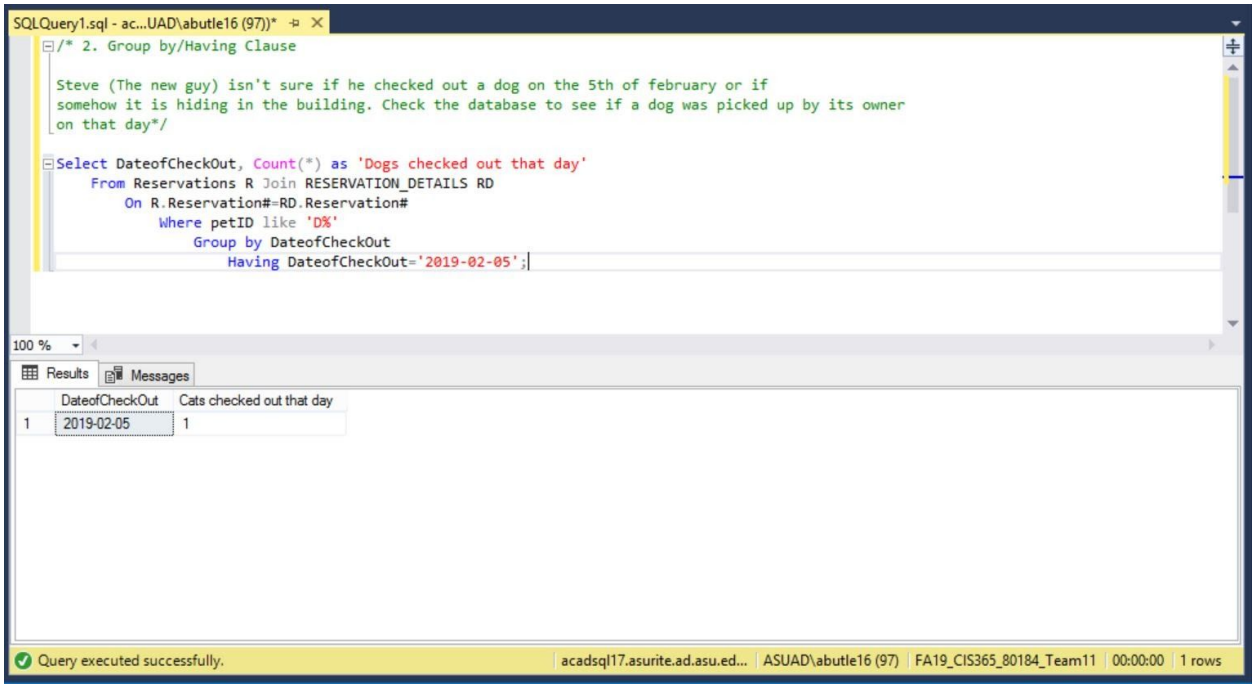

SQLQuery1.sql - ac...UAD\thdespai (69))*
/* 9. Construct 4 other preliminary tests to show that your database design works - select tests that
you know will make your design fail where it should.
Can not input word values into a numeric column (for those not tech savy)
This specific example is trying to put in $21 as the base fee -TD */
Insert into RESERVATIONS values ('10000084','12:17','10:56','02/22/2019','02/23/2019','S80000000','Twenty-One');
100 %
Messages
Msg 8114, Level 16, State 5, Line 7
Error converting data type varchar to numeric.
100 %
Query completed with errors.
acadsql17 (14.0 RTM)
ASUAD\thdespai (69)
FA19_CIS365_80184_Team11
00:00:00
0 rows

10.


SQLQuery1.sql - ac...UAD\thdespai (69))*
/* 10. Construct 4 other preliminary tests to show that your database design works - select tests that you know will make your design fail where it should.
Can't give yourself a bonus in a paycheck. Though, $900,000 would be nice. -TD */
Insert into paycheck values ('S10000000', 5,'2014-10-9',90000.00,904261.20);
100 %
Messages
Msg 8115, Level 16, State 8, Line 6
Arithmetic overflow error converting numeric to data type numeric.
The statement has been terminated.
100 %
Query completed with errors.
acadsql17 (14.0 RTM)
ASUAD\thdespai (69)
FA19_CIS365_80184_Team11
00:00:00
0 rows

Advanced Queries

1. 

2. 

3.

SQLQuery1.sql - ac...AD\thdespai (106))*

```sql
/* 3. GROUP BY/HAVING clause -TD */

/* How many Pets have had more than one dosage of medication while staying with us? */

Select petid as "Pets with more than 1 Administration of Medication", count(petid) as "Number of Dosages"
    from PET_MEDICATIONS
    group by petid
    having count(petid)>1;
```

100 %

Results | Messages

	Pets with more than 1 Administration of Medication	Number of Dosages
1	D00000002	2
2	D00000009	2
3	F00000001	2
4	F00000002	2
5	F00000006	2
6	F00000009	2

Query executed successfully. | acadsql17 (14.0 RTM) | ASUAD\thdespai (106) | FA19_CIS365_80184_Team11 | 00:00:00 | 6 rows

4.

SQLQuery1.sql - ac...UAD\thdespai (69))*

```sql
/* 4. Fully NESTED query (no joins) that combines 3 or more data tables. -TD */

/* Who had reservation# 10000026? They forgot to pay! */

Select CFName, CLName, PhoneAreaCode, Phone
    from Customers
        where customerid in (Select CustomerID
                    from Pets
                        where petid in(Select Petid
                                from RESERVATION_DETAILS
                                    where Reservation# = '10000026'));
```

100 %

Results | Messages

	CFName	CLName	PhoneAreaCode	Phone
1	Kevin	Short	480	106-0310

Query executed successfully. | acadsql17 (14.0 RTM) | ASUAD\thdespai (69) | FA19_CIS365_80184_Team11 | 00:00:00 | 1 rows

5.

```
/* Milestone 4- Team 11-Team Phoenix*/
/* fully nested query & a view-AA */

/*3.Create view Tempe_Pets_Colors and write the fully nested statement to list the PetID and color of the pet for a pet whose owners address is in the
city of Tempe.*/

Create View Tempe_Pets_Colors AS

select PetID,Color
    from COLOR
        where PetID IN
    (select  PetID
        from PETS
            where CustomerID in
                (select CustomerID
                    from CUSTOMERS
                        where City='Tempe'));
select *
    From Tempe_Pets_Colors
```

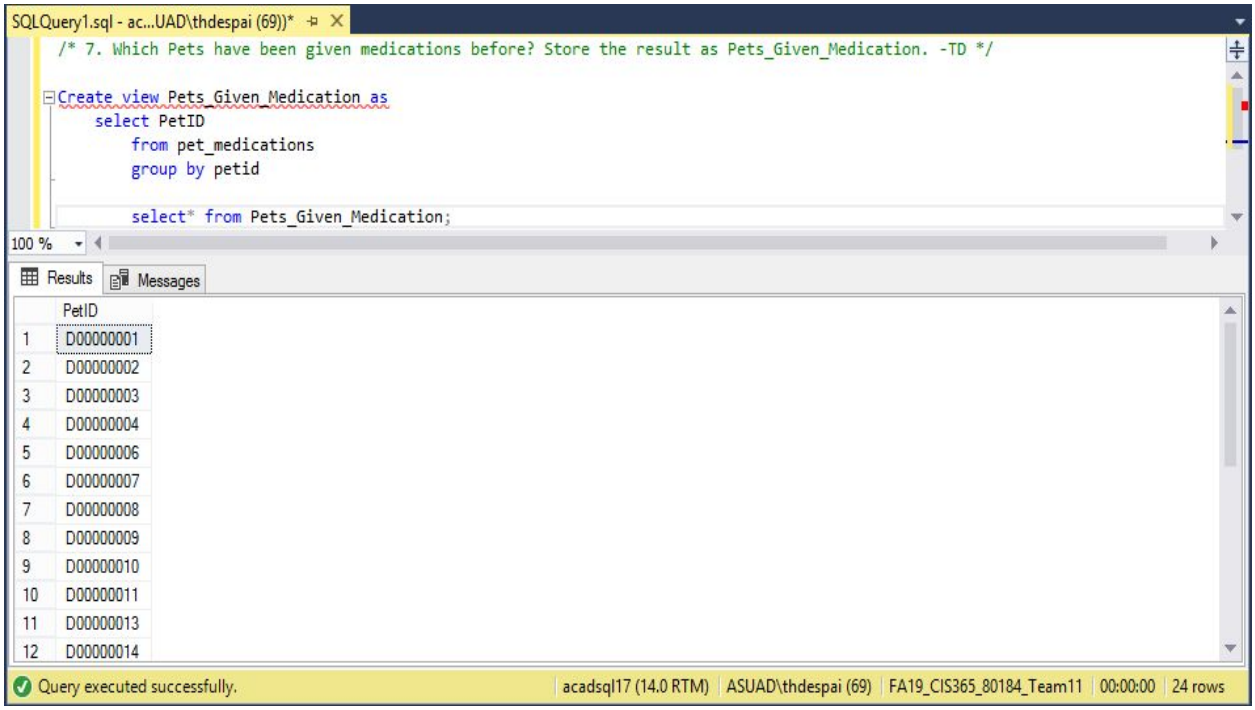
	PetID	Color
1	D00000005	Black
2	D00000005	Grey
3	D00000008	Gold
4	D00000008	Grey
5	D00000012	Brown
6	D00000012	White
7	F00000001	Grey
8	F00000001	White

6.

```
/* Milestone 4- Team 11-Team Phoenix*/
/* fully nested query */

/* 4. Write the fully nested query to display the city and state of all guests who
checked in a pet who did get a bath .*/


Select C.City, C.State, C.CustomerID
    from Customers C
        where CustomerID in
        (select CustomerID
            from PETS
                where PetID in
            (select PetID
                from RESERVATION_DETAILS
                    where Bath='Y'));
```

	PetID	Color
1	D00000005	Black
2	D00000005	Grey
3	D00000008	Gold
4	D00000008	Grey
5	D00000012	Brown
6	D00000012	White
7	F00000001	Grey
8	F00000001	White

7.


SQLQuery1.sql - ac...UAD\thdespai (69))*
/* 7. Which Pets have been given medications before? Store the result as Pets_Given_Medication. -TD */
Create view Pets_Given_Medication as
select PetID
from pet_medications
group by petid
select* from Pets_Given_Medication;
100 %
Results
Messages
PetID
1 D00000001
2 D00000002
3 D00000003
4 D00000004
5 D00000006
6 D00000007
7 D00000008
8 D00000009
9 D00000010
10 D00000011
11 D00000013
12 D00000014
Query executed successfully.
acadsql17 (14.0 RTM)
ASUAD\thdespai (69)
FA19_CIS365_80184_Team11
00:00:00
24 rows

8.

SQLQuery2.sql - ac...UAD\smtolive (55))* | SQLQuery1.sql - ac...UAD\smtolive (53))*

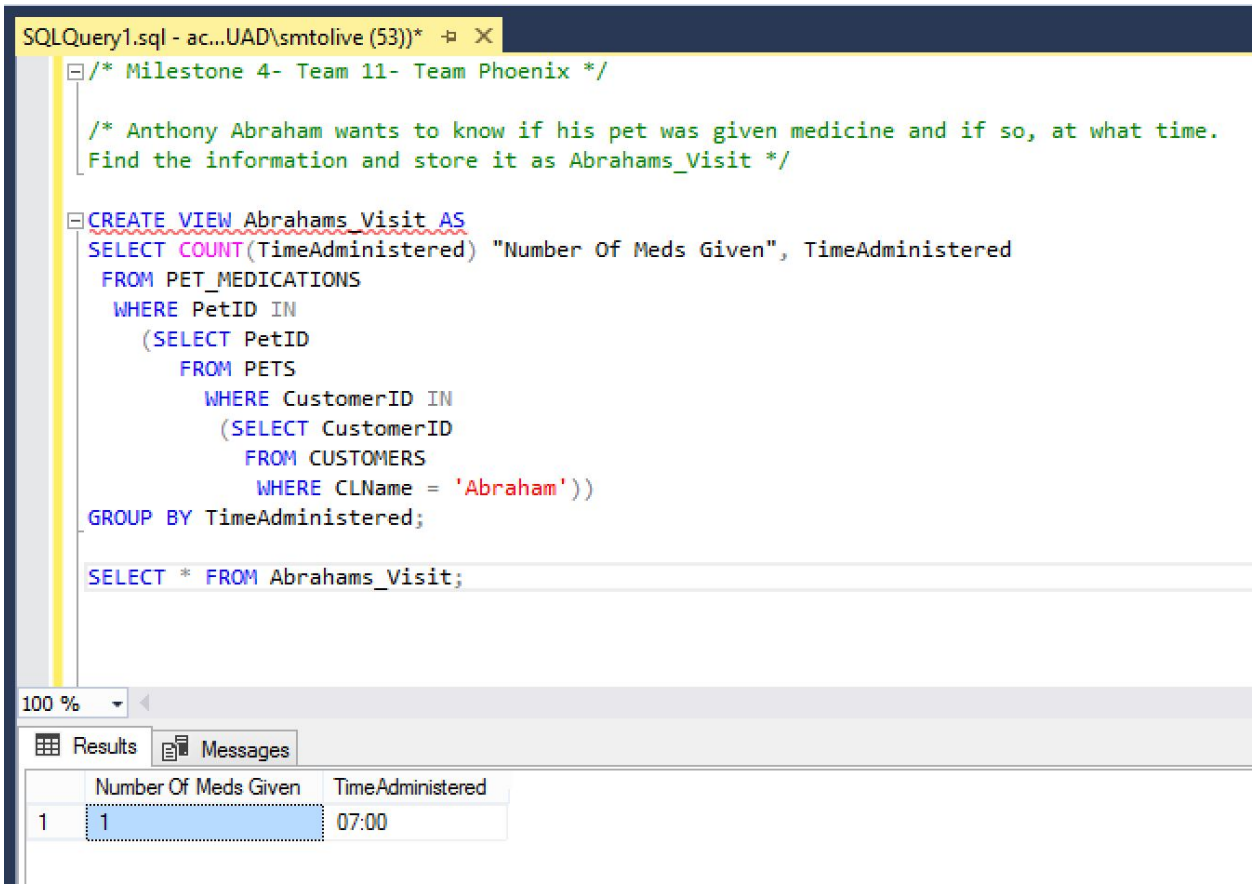
```
/* Milestone 4- Team 11- Team Phoenix */

/* Management want to know how the second half of the year's third quarter went.
List the dates of check in and prices paid. Store the information in 2NDhalf_3RDquarter.
Order by oldest to newest. */

CREATE VIEW "2NDhalf_3RDquarter" AS
SELECT DateOfCheckIn, Price
 FROM RESERVATIONS
  WHERE DateOfCheckIn Between '09/01/19' and '10/31/19'
GROUP BY DateOfCheckIn, Price;

SELECT *
 FROM [2NDhalf_3RDquarter]
  ORDER BY DateOfCheckIN;
```

100 %

Results | Messages

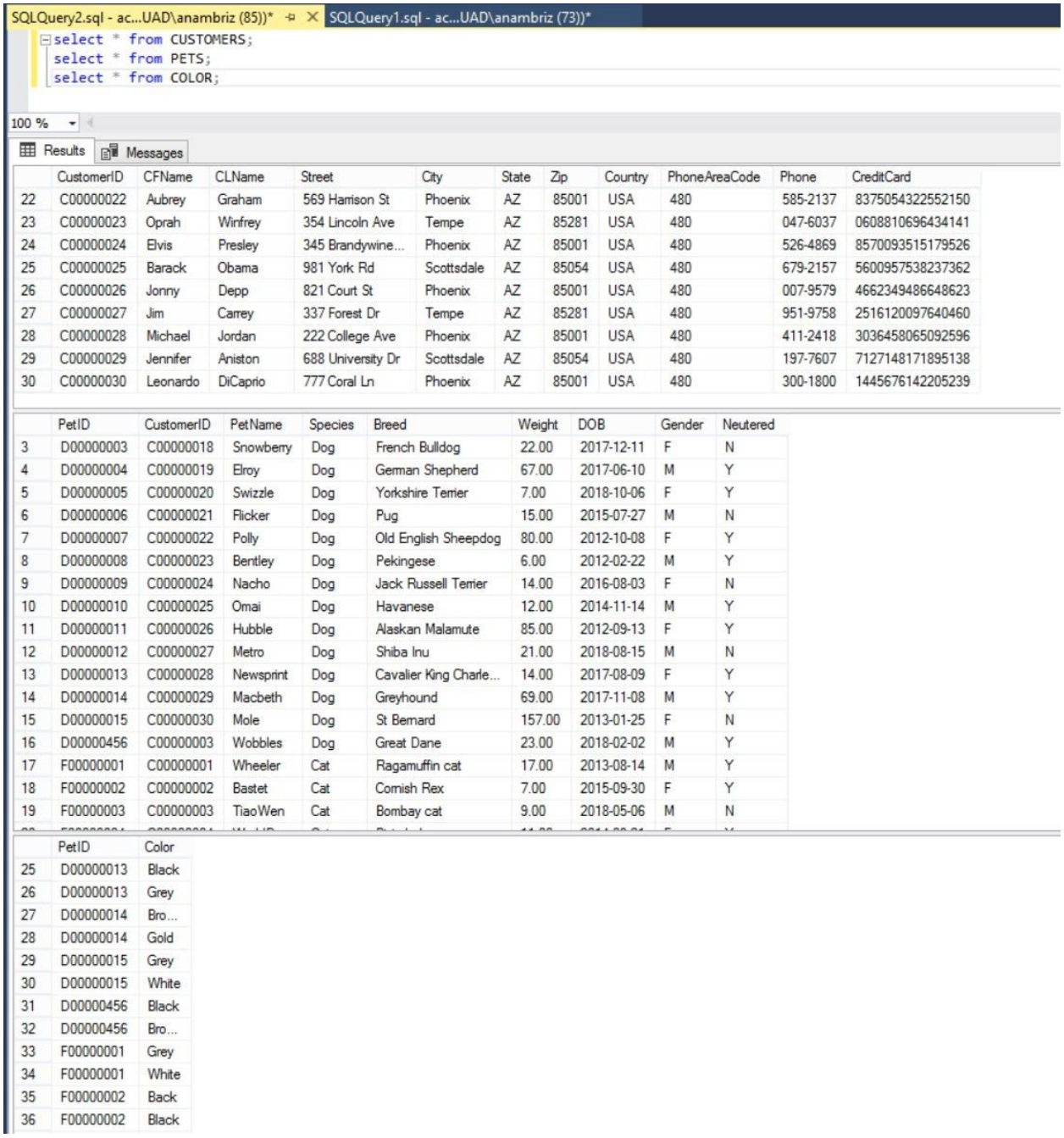
	DateOfCheckIn	Price
1	2019-09-03	63.00
2	2019-09-15	21.00
3	2019-09-19	63.00
4	2019-09-21	20.00
5	2019-09-21	63.00
6	2019-09-28	42.00
7	2019-09-30	84.00

9.


SQLQuery1.sql - ac...UAD\smtolive (53))*
/* Milestone 4- Team 11- Team Phoenix */
/* Anthony Abraham wants to know if his pet was given medicine and if so, at what time.
Find the information and store it as Abrahams_Visit */
CREATE VIEW Abrahams_Visit AS
SELECT COUNT(TimeAdministered) "Number Of Meds Given", TimeAdministered
FROM PET_MEDICATIONS
WHERE PetID IN
(SELECT PetID
FROM PETS
WHERE CustomerID IN
(SELECT CustomerID
FROM CUSTOMERS
WHERE CLName = 'Abraham'))
GROUP BY TimeAdministered;
SELECT * FROM Abrahams_Visit;
100 %
Results
Messages
Number Of Meds Given
TimeAdministered
1
1
07:00

Transactions

1.Before



SQLQuery2.sql - ac...UAD\anambriz (85))* | SQLQuery1.sql - ac...UAD\anambriz (73))*

```
select * from CUSTOMERS;
select * from PETS;
select * from COLOR;
```

100 %

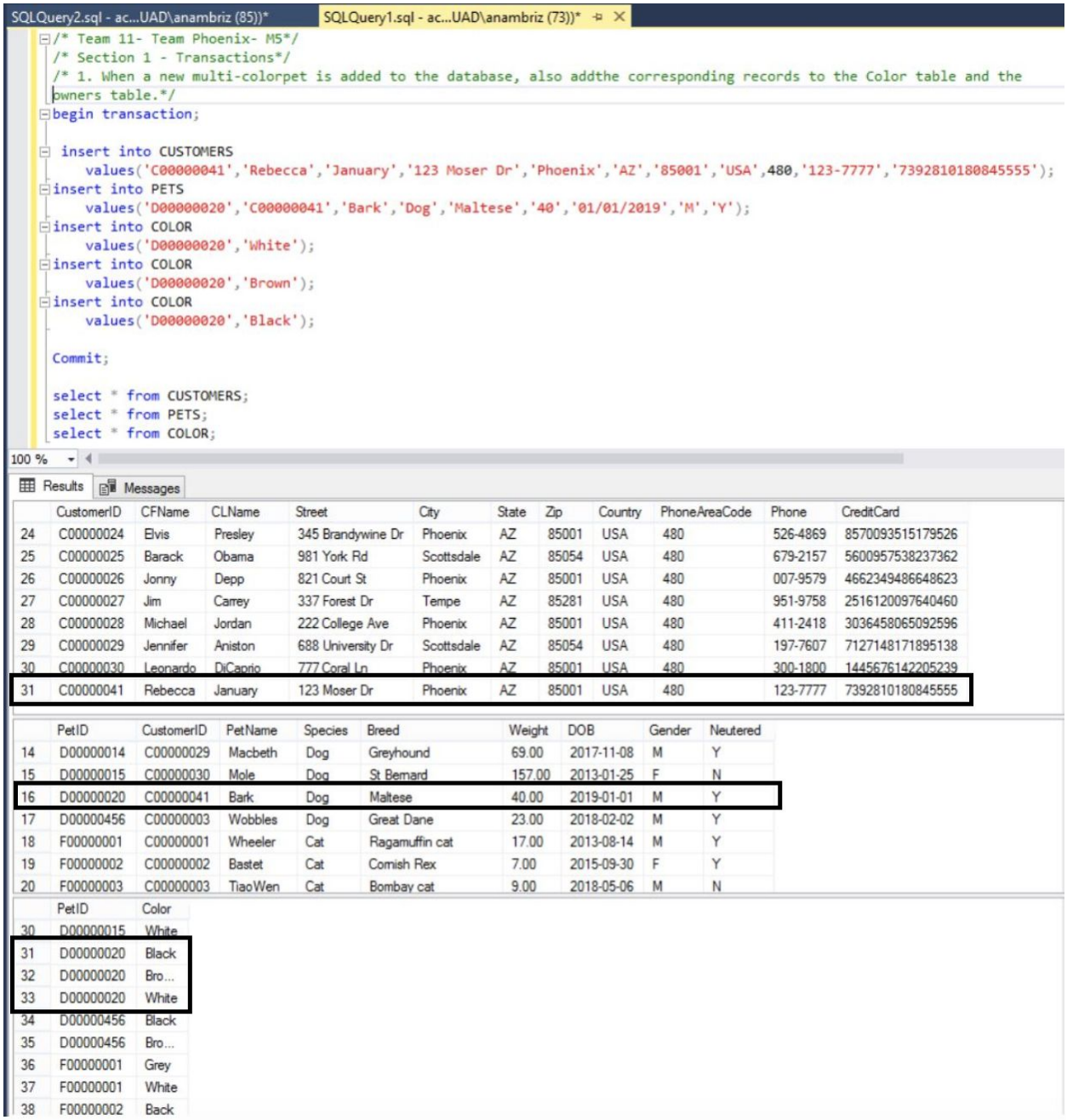
Results | Messages

	CustomerID	CFName	CLName	Street	City	State	Zip	Country	PhoneAreaCode	Phone	CreditCard
22	C00000022	Aubrey	Graham	569 Harrison St	Phoenix	AZ	85001	USA	480	585-2137	8375054322552150
23	C00000023	Oprah	Winfrey	354 Lincoln Ave	Tempe	AZ	85281	USA	480	047-6037	0608810696434141
24	C00000024	Elvis	Presley	345 Brandywine...	Phoenix	AZ	85001	USA	480	526-4869	8570093515179526
25	C00000025	Barack	Obama	981 York Rd	Scottsdale	AZ	85054	USA	480	679-2157	5600957538237362
26	C00000026	Jonny	Depp	821 Court St	Phoenix	AZ	85001	USA	480	007-9579	4662349486648623
27	C00000027	Jim	Carrey	337 Forest Dr	Tempe	AZ	85281	USA	480	951-9758	2516120097640460
28	C00000028	Michael	Jordan	222 College Ave	Phoenix	AZ	85001	USA	480	411-2418	3036458065092596
29	C00000029	Jennifer	Aniston	688 University Dr	Scottsdale	AZ	85054	USA	480	197-7607	7127148171895138
30	C00000030	Leonardo	DiCaprio	777 Coral Ln	Phoenix	AZ	85001	USA	480	300-1800	1445676142205239

	PetID	CustomerID	PetName	Species	Breed	Weight	DOB	Gender	Neutered
3	D00000003	C00000018	Snowberry	Dog	French Bulldog	22.00	2017-12-11	F	N
4	D00000004	C00000019	Elroy	Dog	German Shepherd	67.00	2017-06-10	M	Y
5	D00000005	C00000020	Swizzle	Dog	Yorkshire Terrier	7.00	2018-10-06	F	Y
6	D00000006	C00000021	Flicker	Dog	Pug	15.00	2015-07-27	M	N
7	D00000007	C00000022	Polly	Dog	Old English Sheepdog	80.00	2012-10-08	F	Y
8	D00000008	C00000023	Bentley	Dog	Pekingese	6.00	2012-02-22	M	Y
9	D00000009	C00000024	Nacho	Dog	Jack Russell Terrier	14.00	2016-08-03	F	N
10	D00000010	C00000025	Omai	Dog	Havanese	12.00	2014-11-14	M	Y
11	D00000011	C00000026	Hubble	Dog	Alaskan Malamute	85.00	2012-09-13	F	Y
12	D00000012	C00000027	Metro	Dog	Shiba Inu	21.00	2018-08-15	M	N
13	D00000013	C00000028	Newsprint	Dog	Cavalier King Charle...	14.00	2017-08-09	F	Y
14	D00000014	C00000029	Macbeth	Dog	Greyhound	69.00	2017-11-08	M	Y
15	D00000015	C00000030	Mole	Dog	St Bernard	157.00	2013-01-25	F	N
16	D00000456	C00000003	Wobbles	Dog	Great Dane	23.00	2018-02-02	M	Y
17	F00000001	C00000001	Wheeler	Cat	Ragamuffin cat	17.00	2013-08-14	M	Y
18	F00000002	C00000002	Bastet	Cat	Cornish Rex	7.00	2015-09-30	F	Y
19	F00000003	C00000003	TiaoWen	Cat	Bombay cat	9.00	2018-05-06	M	N

	PetID	Color
25	D00000013	Black
26	D00000013	Grey
27	D00000014	Bro...
28	D00000014	Gold
29	D00000015	Grey
30	D00000015	White
31	D00000456	Black
32	D00000456	Bro...
33	F00000001	Grey
34	F00000001	White
35	F00000002	Back
36	F00000002	Black

1. After



```
/* Team 11- Team Phoenix- M5*/
/* Section 1 - Transactions*/
/* 1. When a new multi-colorpet is added to the database, also addthe corresponding records to the Color table and the
owners table.*/
begin transaction;

 insert into CUSTOMERS
    values('C00000041','Rebecca','January','123 Moser Dr','Phoenix','AZ','85001','USA',480,'123-7777','7392810180845555');
insert into PETS
    values('D00000020','C00000041','Bark','Dog','Maltese','40','01/01/2019','M','Y');
insert into COLOR
    values('D00000020','White');
insert into COLOR
    values('D00000020','Brown');
insert into COLOR
    values('D00000020','Black');

Commit;

select * from CUSTOMERS;
select * from PETS;
select * from COLOR;
```

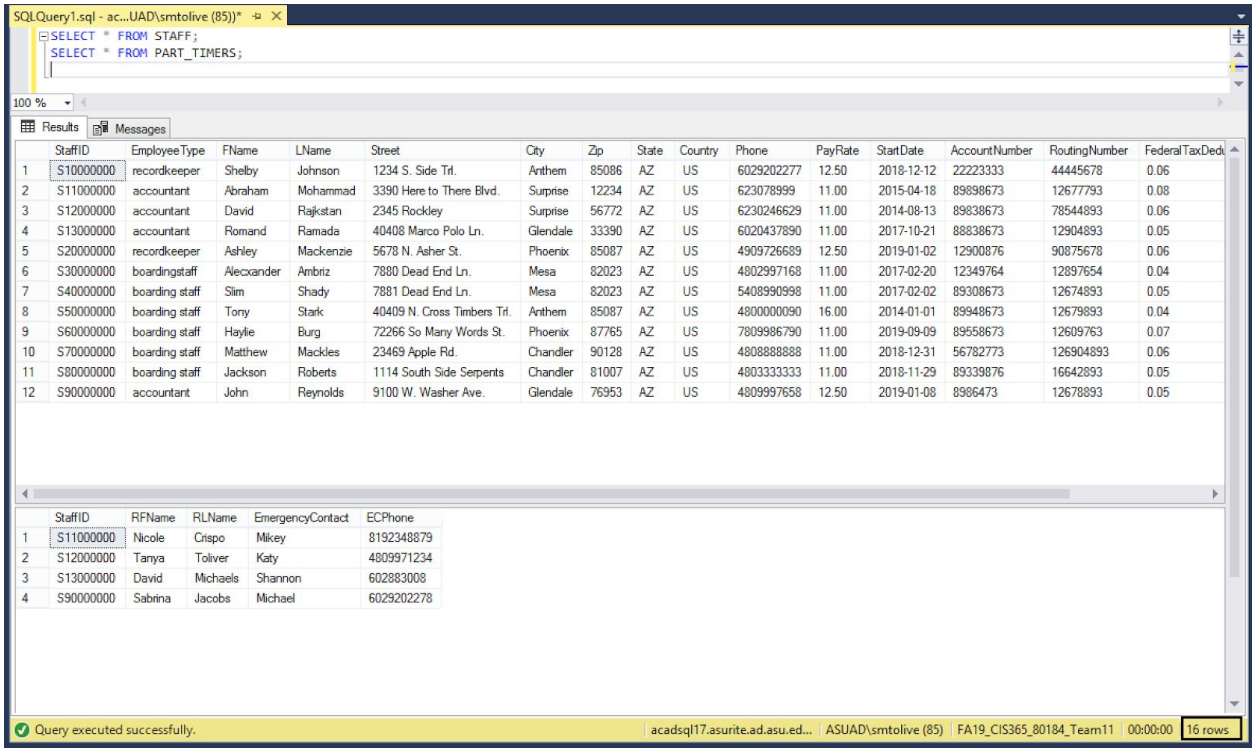
Results | Messages

	CustomerID	CFName	CLName	Street	City	State	Zip	Country	PhoneAreaCode	Phone	CreditCard
24	C00000024	Elvis	Presley	345 Brandywine Dr	Phoenix	AZ	85001	USA	480	526-4869	8570093515179526
25	C00000025	Barack	Obama	981 York Rd	Scottsdale	AZ	85054	USA	480	679-2157	5600957538237362
26	C00000026	Jonny	Depp	821 Court St	Phoenix	AZ	85001	USA	480	007-9579	4662349486648623
27	C00000027	Jim	Carrey	337 Forest Dr	Tempe	AZ	85281	USA	480	951-9758	2516120097640460
28	C00000028	Michael	Jordan	222 College Ave	Phoenix	AZ	85001	USA	480	411-2418	3036458065092596
29	C00000029	Jennifer	Aniston	688 University Dr	Scottsdale	AZ	85054	USA	480	197-7607	7127148171895138
30	C00000030	Leonardo	DiCaprio	777 Coral Ln	Phoenix	AZ	85001	USA	480	300-1800	1445676142205239
31	C00000041	Rebecca	January	123 Moser Dr	Phoenix	AZ	85001	USA	480	123-7777	7392810180845555

	PetID	CustomerID	PetName	Species	Breed	Weight	DOB	Gender	Neutered
14	D00000014	C00000029	Macbeth	Dog	Greyhound	69.00	2017-11-08	M	Y
15	D00000015	C00000030	Mole	Dog	St Bernard	157.00	2013-01-25	F	N
16	D00000020	C00000041	Bark	Dog	Maltese	40.00	2019-01-01	M	Y
17	D00000456	C00000003	Wobbles	Dog	Great Dane	23.00	2018-02-02	M	Y
18	F00000001	C00000001	Wheeler	Cat	Ragamuffin cat	17.00	2013-08-14	M	Y
19	F00000002	C00000002	Bastet	Cat	Cornish Rex	7.00	2015-09-30	F	Y
20	F00000003	C00000003	TiaoWen	Cat	Bombay cat	9.00	2018-05-06	M	N

	PetID	Color
30	D00000015	White
31	D00000020	Black
32	D00000020	Bro...
33	D00000020	White
34	D00000456	Black
35	D00000456	Bro...
36	F00000001	Grey
37	F00000001	White
38	F00000002	Back

2.Before



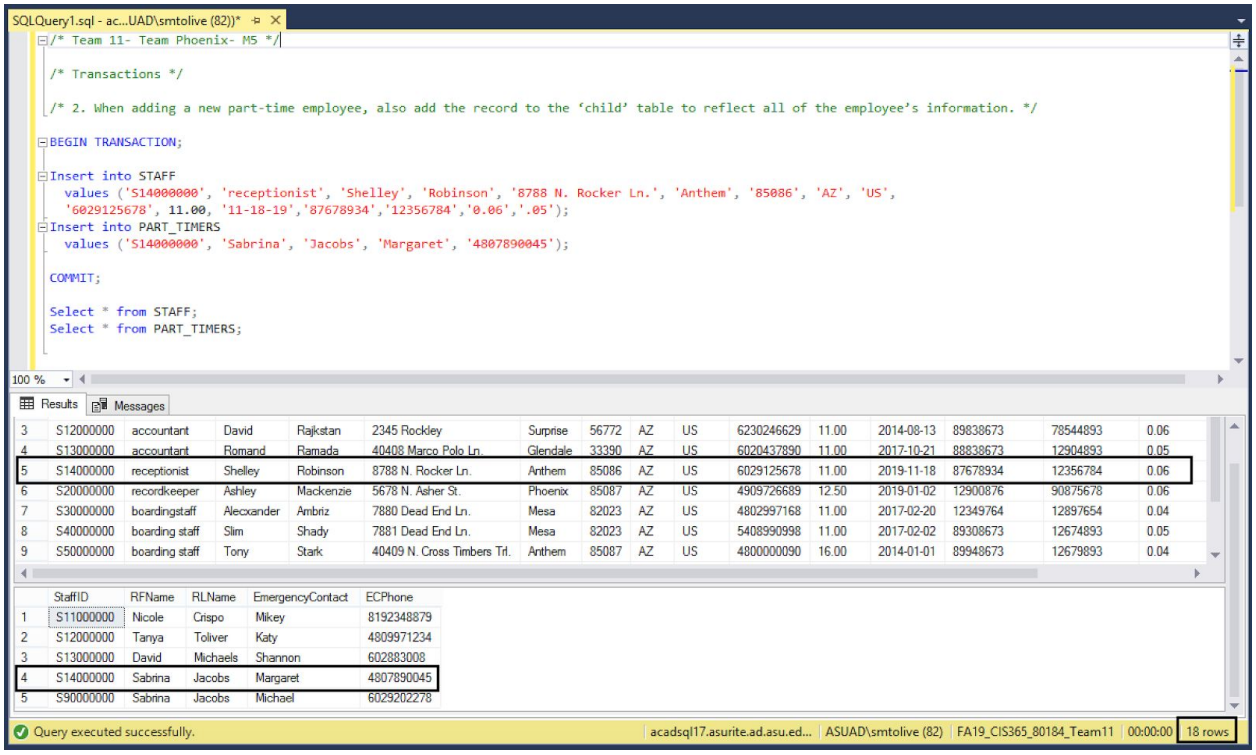
```
SELECT * FROM STAFF;
SELECT * FROM PART_TIMERS;
```

	StaffID	EmployeeType	FName	LName	Street	City	Zip	State	Country	Phone	PayRate	StartDate	AccountNumber	RoutingNumber	FederalTaxDed
1	S10000000	recordkeeper	Shelby	Johnson	1234 S. Side Trl.	Anthem	85086	AZ	US	6029202277	12.50	2018-12-12	22223333	44445678	0.06
2	S11000000	accountant	Abraham	Mohammad	3390 Here to There Blvd.	Surprise	12234	AZ	US	623078999	11.00	2015-04-18	89898673	12677793	0.08
3	S12000000	accountant	David	Rajkstan	2345 Rockley	Surprise	56772	AZ	US	6230246629	11.00	2014-08-13	89838673	78544893	0.06
4	S13000000	accountant	Romand	Ramada	40408 Marco Polo Ln.	Glendale	33390	AZ	US	6020437890	11.00	2017-10-21	88838673	12904893	0.05
5	S20000000	recordkeeper	Ashley	Mackenzie	5678 N. Asher St.	Phoenix	85087	AZ	US	4909726689	12.50	2019-01-02	12900876	90875678	0.06
6	S30000000	boardingstaff	Alecxander	Ambriz	7880 Dead End Ln.	Mesa	82023	AZ	US	4802997168	11.00	2017-02-20	12349764	12897654	0.04
7	S40000000	boarding staff	Slim	Shady	7881 Dead End Ln.	Mesa	82023	AZ	US	5408990998	11.00	2017-02-02	89308673	12674893	0.05
8	S50000000	boarding staff	Tony	Stark	40409 N. Cross Timbers Trl.	Anthem	85087	AZ	US	4800000090	16.00	2014-01-01	89948673	12679893	0.04
9	S60000000	boarding staff	Haylie	Burg	72266 So Many Words St.	Phoenix	87765	AZ	US	7809986790	11.00	2019-09-09	89558673	12609763	0.07
10	S70000000	boarding staff	Matthew	Mackles	23469 Apple Rd.	Chandler	90128	AZ	US	4808888888	11.00	2018-12-31	56782773	126904893	0.06
11	S80000000	boarding staff	Jackson	Roberts	1114 South Side Serpents	Chandler	81007	AZ	US	4803333333	11.00	2018-11-29	89339876	16642893	0.05
12	S90000000	accountant	John	Reynolds	9100 W. Washer Ave.	Glendale	76953	AZ	US	4809997658	12.50	2019-01-08	8986473	12678893	0.05

	StaffID	RFName	RLName	EmergencyContact	ECPhone
1	S11000000	Nicole	Crispo	Mikey	8192348879
2	S12000000	Tanya	Toliver	Katy	4809971234
3	S13000000	David	Michaels	Shannon	602883008
4	S90000000	Sabrina	Jacobs	Michael	6029202278

Query executed successfully. acadsql17.asurite.ad.asu.ed... | ASUAD\smtolive (85) | FA19_CIS365_80184_Team11 | 00:00:00 | 16 rows

2.After



```
/* Team 11- Team Phoenix- M5 */

/* Transactions */

/* 2. When adding a new part-time employee, also add the record to the 'child' table to reflect all of the employee's information. */

BEGIN TRANSACTION;

Insert into STAFF
  values ('S14000000', 'receptionist', 'Shelley', 'Robinson', '8788 N. Rocker Ln.', 'Anthem', '85086', 'AZ', 'US',
  '6029125678', 11.00, '11-18-19','87678934','12356784','0.06','.05');
Insert into PART_TIMERS
  values ('S14000000', 'Sabrina', 'Jacobs', 'Margaret', '4807890045');

COMMIT;

Select * from STAFF;
Select * from PART_TIMERS;
```

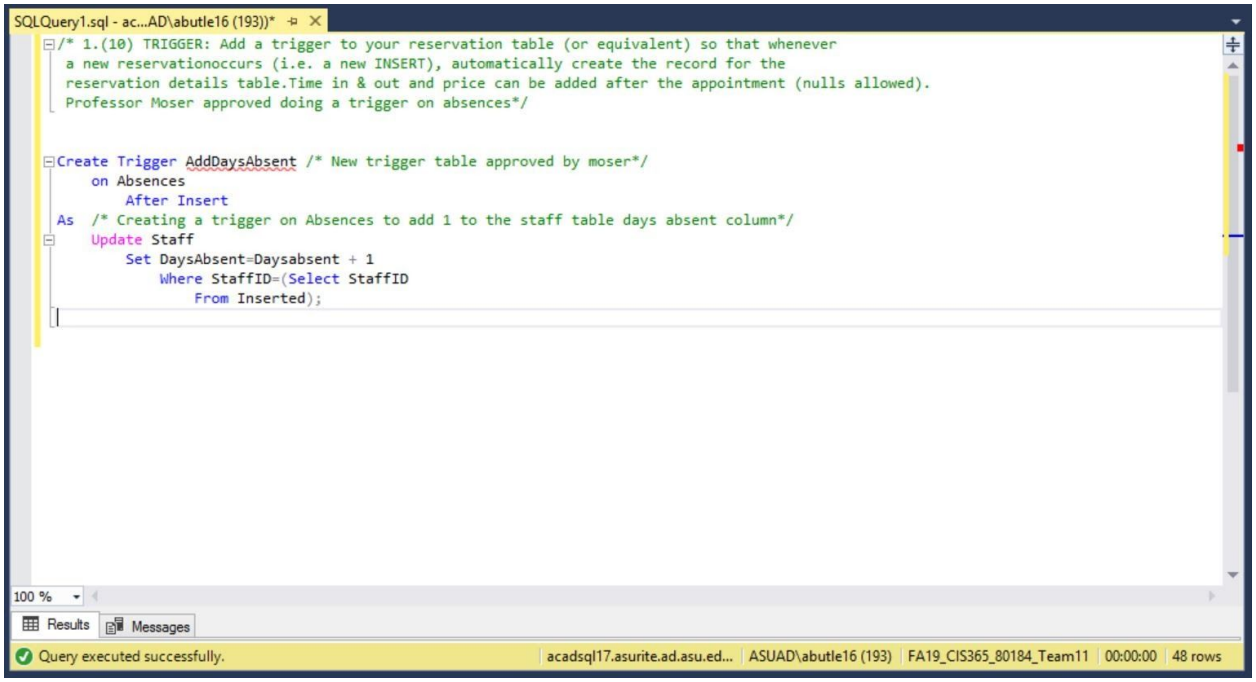
	StaffID	EmployeeType	FName	LName	Street	City	Zip	State	Country	Phone	PayRate	StartDate	AccountNumber	RoutingNumber	FederalTaxDed
3	S12000000	accountant	David	Rajkstan	2345 Rockley	Surprise	56772	AZ	US	6230246629	11.00	2014-08-13	89838673	78544893	0.06
4	S13000000	accountant	Romand	Ramada	40408 Marco Polo Ln.	Glendale	33390	AZ	US	6020437890	11.00	2017-10-21	88838673	12904893	0.05
5	S14000000	receptionist	Shelley	Robinson	8788 N. Rocker Ln.	Anthem	85086	AZ	US	6029125678	11.00	2019-11-18	87678934	12356784	0.06
6	S20000000	recordkeeper	Ashley	Mackenzie	5678 N. Asher St.	Phoenix	85087	AZ	US	4909726689	12.50	2019-01-02	12900876	90875678	0.06
7	S30000000	boardingstaff	Alecxander	Ambriz	7880 Dead End Ln.	Mesa	82023	AZ	US	4802997168	11.00	2017-02-20	12349764	12897654	0.04
8	S40000000	boarding staff	Slim	Shady	7881 Dead End Ln.	Mesa	82023	AZ	US	5408990998	11.00	2017-02-02	89308673	12674893	0.05
9	S50000000	boarding staff	Tony	Stark	40409 N. Cross Timbers Trl.	Anthem	85087	AZ	US	4800000090	16.00	2014-01-01	89948673	12679893	0.04

	StaffID	RFName	RLName	EmergencyContact	ECPhone
1	S11000000	Nicole	Crispo	Mikey	8192348879
2	S12000000	Tanya	Toliver	Katy	4809971234
3	S13000000	David	Michaels	Shannon	602883008
4	S14000000	Sabrina	Jacobs	Margaret	4807890045
5	S90000000	Sabrina	Jacobs	Michael	6029202278

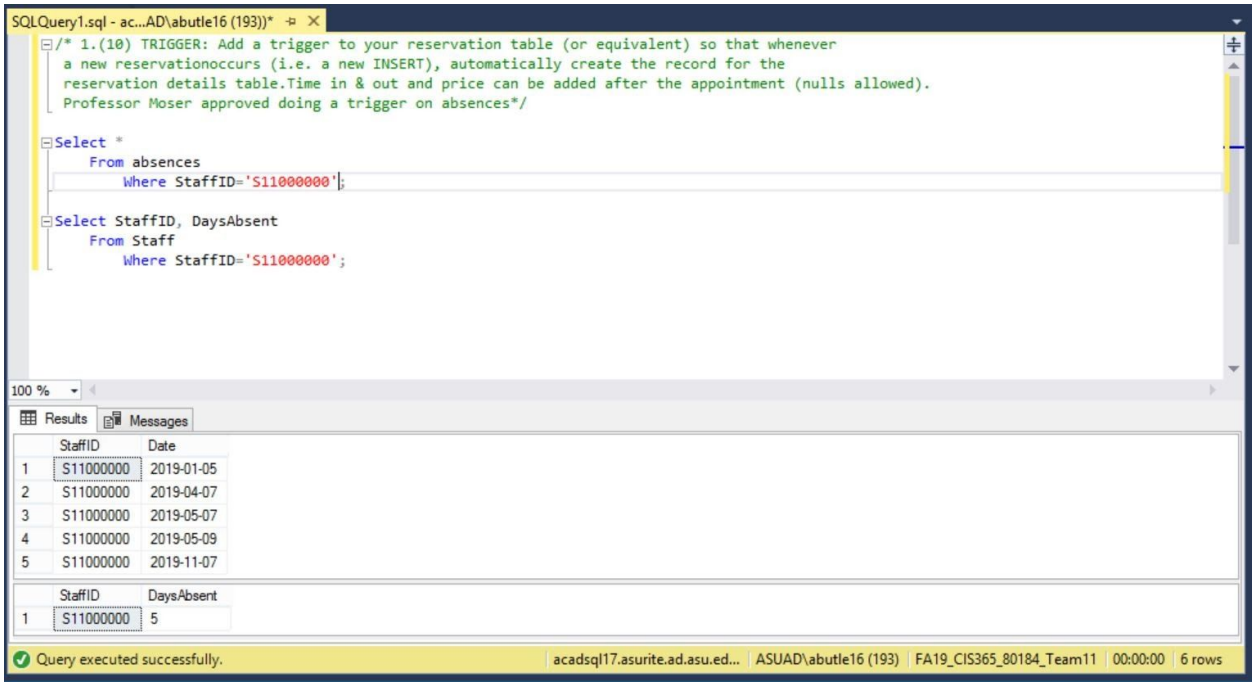
Query executed successfully. acadsql17.asurite.ad.asu.ed... | ASUAD\smtolive (82) | FA19_CIS365_80184_Team11 | 00:00:00 | 18 rows

Triggers

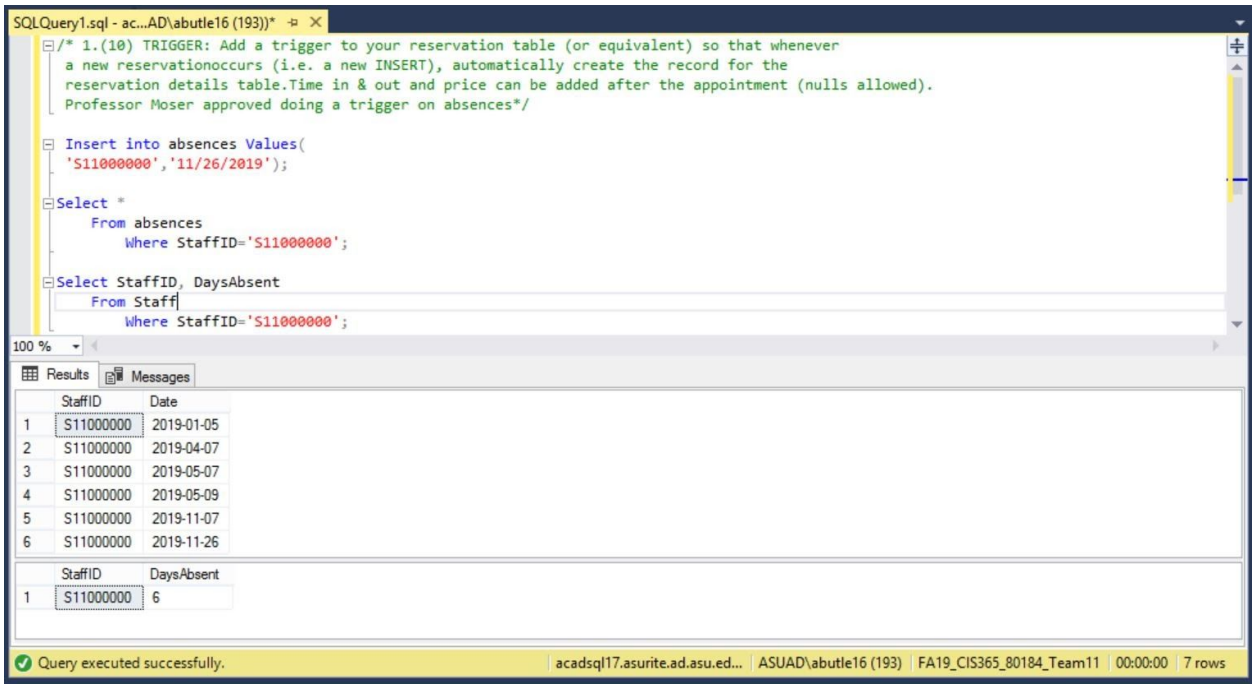
Code:



Before:



After:


SQLQuery1.sql - ac...AD\abutle16 (193))*
/* 1.(10) TRIGGER: Add a trigger to your reservation table (or equivalent) so that whenever
a new reservationoccurs (i.e. a new INSERT), automatically create the record for the
reservation details table.Time in & out and price can be added after the appointment (nulls allowed).
Professor Moser approved doing a trigger on absences*/
Insert into absences Values(
'S11000000','11/26/2019');
Select *
From absences
Where StaffID='S11000000';
Select StaffID, DaysAbsent
From Staff
Where StaffID='S11000000';
100 %
Results
Messages
StaffID Date
1 S11000000 2019-01-05
2 S11000000 2019-04-07
3 S11000000 2019-05-07
4 S11000000 2019-05-09
5 S11000000 2019-11-07
6 S11000000 2019-11-26
StaffID DaysAbsent
1 S11000000 6
Query executed successfully.
acadsql17.asurite.ad.asu.ed...
ASUAD\abutle16 (193)
FA19_CIS365_80184_Team11
00:00:00
7 rows

Procedure (Procedure query script in .txt file)

Before:

SQLQuery1.sql - ac...AD\thdespai (118))*

```
/* (10) PROCEDURE:  When an employee has a sick day, add a new record to the SickDaysLog
to record the employee'sID, name, date sick, and the total number of sick days thus far
including the one addedfor this employee. Hint:You will need to create a SickDaysLogto
keep track of this data.  The procedure will execute after a new sick day has been recorded
for the employee. */

Create Table SickDaysLog(
    StaffID  Char(9) Not Null,
    Fname  Varchar(20),
    SickDate   Date,
    TotalSickdays VarChar(3),
    Primary Key (StaffID, SickDate),
    foreign key (StaffID) References Staff(StaffID));

Create Procedure SickDay
    @StaffID Char(9),
    @Date Date as
    Insert Into SickDaysLog
        Values (@StaffID,
            (Select FName from Staff where staffID = @staffid),
            @Date, (Select COUNT(*) From SickDaysLog Where StaffID=@StaffID)+1); /* Look at most recent
            date for the correct amount of sick days */

Exec SickDay 'S14000000','11/26/19';

Select * from SickDaysLog;
```

100 %

Results | Messages

	StaffID	Fname	SickDate	TotalSickdays
1	S10000000	Shelby	2019-11-23	1
2	S10000000	Shelby	2019-11-24	2
3	S10000000	Shelby	2019-11-25	3
4	S10000000	Shelby	2019-11-26	4
5	S11000000	Abra...	2019-11-22	1

After:

SQLQuery1.sql - ac...AD\thdespai (118))*

```
/* (10) PROCEDURE:  When an employee has a sick day, add a new record to the SickDaysLog
to record the employee'sID, name, date sick, and the total number of sick days thus far
including the one addedfor this employee. Hint:You will need to create a SickDaysLogto
keep track of this data.  The procedure will execute after a new sick day has been recorded
for the employee. */

Create Table SickDaysLog(
    StaffID  Char(9) Not Null,
    Fname  Varchar(20),
    SickDate   Date,
    TotalSickdays VarChar(3),
    Primary Key (StaffID, SickDate),
    foreign key (StaffID) References Staff(StaffID));

Create Procedure SickDay
    @StaffID Char(9),
    @Date Date as
    Insert Into SickDaysLog
        Values (@StaffID,
            (Select FName from Staff where staffID = @staffid),
            @Date, (Select COUNT(*) From SickDaysLog Where StaffID=@StaffID)+1); /* Look at most recent
            date for the correct amount of sick days */

Exec SickDay 'S14000000','11/26/19';

Select * from SickDaysLog;
```

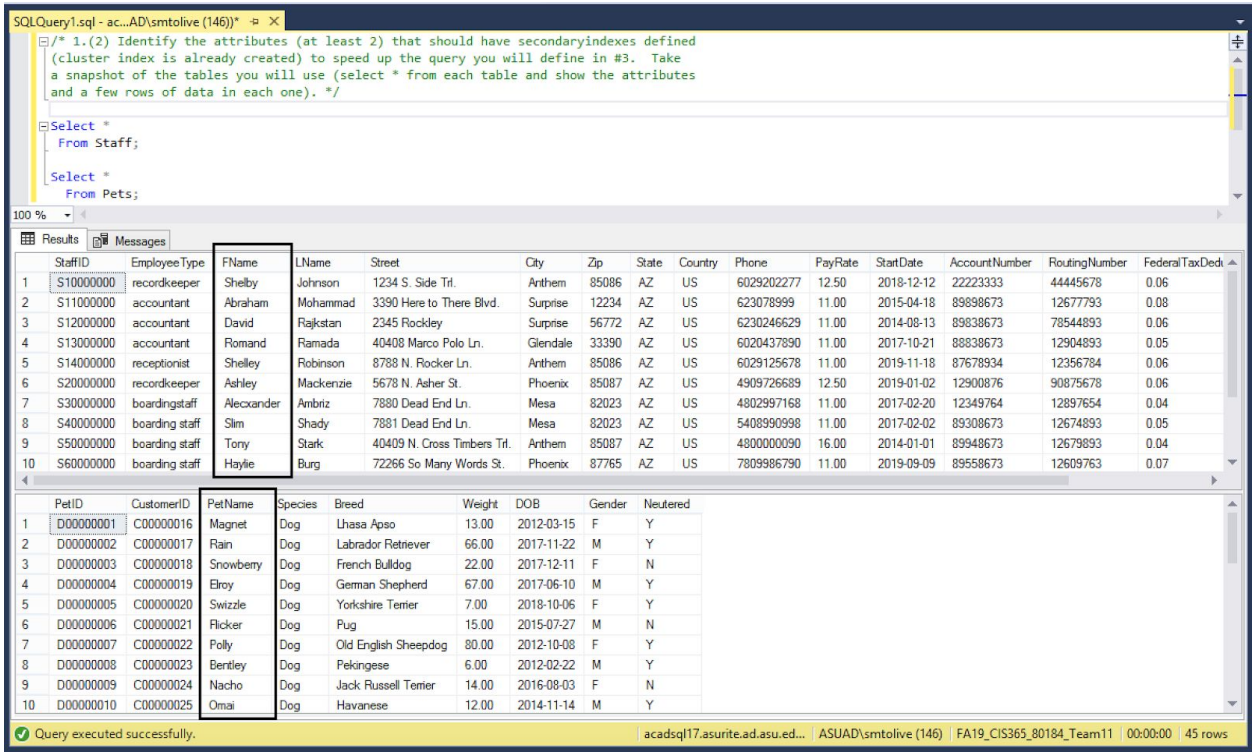
100 %

Results | Messages

	StaffID	Fname	SickDate	TotalSickdays
1	S10000000	Shelby	2019-11-23	1
2	S10000000	Shelby	2019-11-24	2
3	S10000000	Shelby	2019-11-25	3
4	S10000000	Shelby	2019-11-26	4
5	S11000000	Abraham	2019-11-22	1
6	S14000000	Shelley	2019-11-26	1

Indexes

1. Identifying Attributes To Put Secondary Indexes On:



```
/* 1.(2) Identify the attributes (at least 2) that should have secondaryindexes defined
(cluster index is already created) to speed up the query you will define in #3.  Take
a snapshot of the tables you will use (select * from each table and show the attributes
and a few rows of data in each one). */

Select *
 From Staff;

Select *
  From Pets;
```

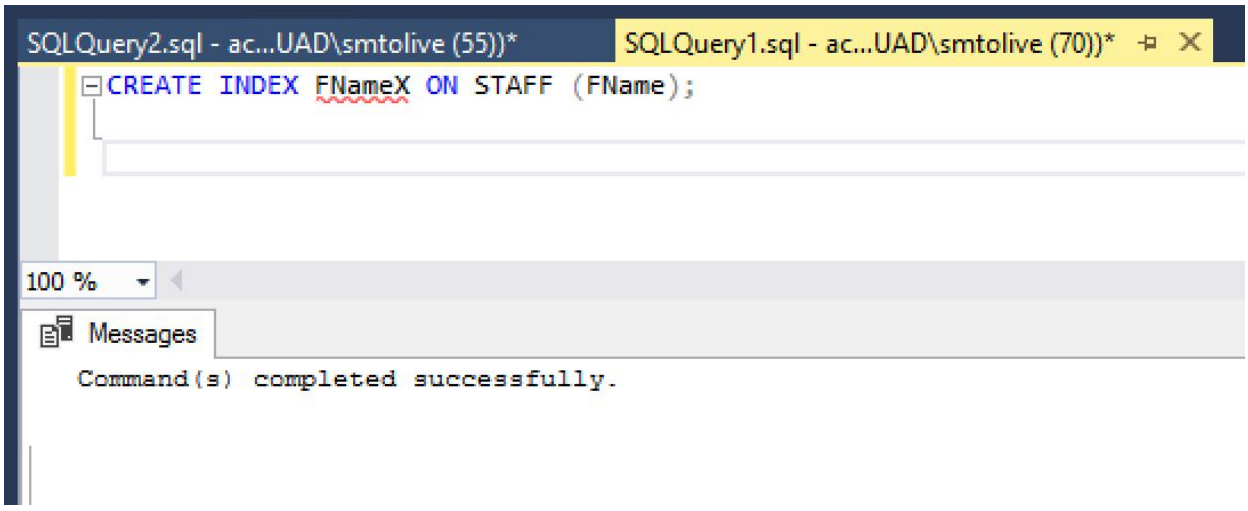
Results | Messages

	StaffID	EmployeeType	FName	LName	Street	City	Zip	State	Country	Phone	PayRate	StartDate	AccountNumber	RoutingNumber	FederalTaxDed
1	S10000000	recordkeeper	Shelby	Johnson	1234 S. Side Trl.	Anthem	85086	AZ	US	6029202277	12.50	2018-12-12	22223333	44445678	0.06
2	S11000000	accountant	Abraham	Mohammad	3390 Here to There Blvd.	Surprise	12234	AZ	US	623078999	11.00	2015-04-18	89898673	12677793	0.08
3	S12000000	accountant	David	Rajkstan	2345 Rockley	Surprise	56772	AZ	US	6230246629	11.00	2014-08-13	89838673	78544893	0.06
4	S13000000	accountant	Romand	Ramada	40408 Marco Polo Ln.	Glendale	33390	AZ	US	6020437890	11.00	2017-10-21	88838673	12904893	0.05
5	S14000000	receptionist	Shelley	Robinson	8788 N. Rocker Ln.	Anthem	85086	AZ	US	6029125678	11.00	2019-11-18	87678934	12356784	0.06
6	S20000000	recordkeeper	Ashley	Mackenzie	5678 N. Asher St.	Phoenix	85087	AZ	US	4909726689	12.50	2019-01-02	12900876	90875678	0.06
7	S30000000	boardingstaff	Alecxander	Ambriz	7880 Dead End Ln.	Mesa	82023	AZ	US	4802997168	11.00	2017-02-20	12349764	12897654	0.04
8	S40000000	boarding staff	Slim	Shady	7881 Dead End Ln.	Mesa	82023	AZ	US	5408990998	11.00	2017-02-02	89308673	12674893	0.05
9	S50000000	boarding staff	Tony	Stark	40409 N. Cross Timbers Trl.	Anthem	85087	AZ	US	4800000090	16.00	2014-01-01	89948673	12679893	0.04
10	S60000000	boarding staff	Haylie	Burg	72266 So Many Words St.	Phoenix	87765	AZ	US	7809986790	11.00	2019-09-09	89558673	12609763	0.07

	PetID	CustomerID	PetName	Species	Breed	Weight	DOB	Gender	Neutered
1	D00000001	C00000016	Magnet	Dog	Lhasa Apso	13.00	2012-03-15	F	Y
2	D00000002	C00000017	Rain	Dog	Labrador Retriever	66.00	2017-11-22	M	Y
3	D00000003	C00000018	Snowberry	Dog	French Bulldog	22.00	2017-12-11	F	N
4	D00000004	C00000019	Elroy	Dog	German Shepherd	67.00	2017-06-10	M	Y
5	D00000005	C00000020	Swizzle	Dog	Yorkshire Terrier	7.00	2018-10-06	F	Y
6	D00000006	C00000021	Flicker	Dog	Pug	15.00	2015-07-27	M	N
7	D00000007	C00000022	Polly	Dog	Old English Sheepdog	80.00	2012-10-08	F	Y
8	D00000008	C00000023	Bentley	Dog	Pekingese	6.00	2012-02-22	M	Y
9	D00000009	C00000024	Nacho	Dog	Jack Russell Terrier	14.00	2016-08-03	F	N
10	D00000010	C00000025	Omai	Dog	Havanese	12.00	2014-11-14	M	Y

Query executed successfully. | acadsql17.asurite.ad.asu.ed... | ASUAD\smtolive (146) | FA19_CIS365_80184_Team11 | 00:00:00 | 45 rows

2.



SQLQuery2.sql - ac...UAD\smtolive (55))* | SQLQuery1.sql - ac...UAD\smtolive (70))*

```
CREATE INDEX FNameX ON STAFF (FName);
```

100 %

Messages

```
Command(s) completed successfully.
```

SQLQuery2.sql - ac...UAD\smtolive (55))* | SQLQuery1.sql - ac...UAD\smtolive (70))*

```
CREATE INDEX PetNameX ON PETS (PetName);
```

100 %

Messages

```
Command(s) completed successfully.
```

3. Before:

SQLQuery1.sql - ac...AD\thdespai (105))*

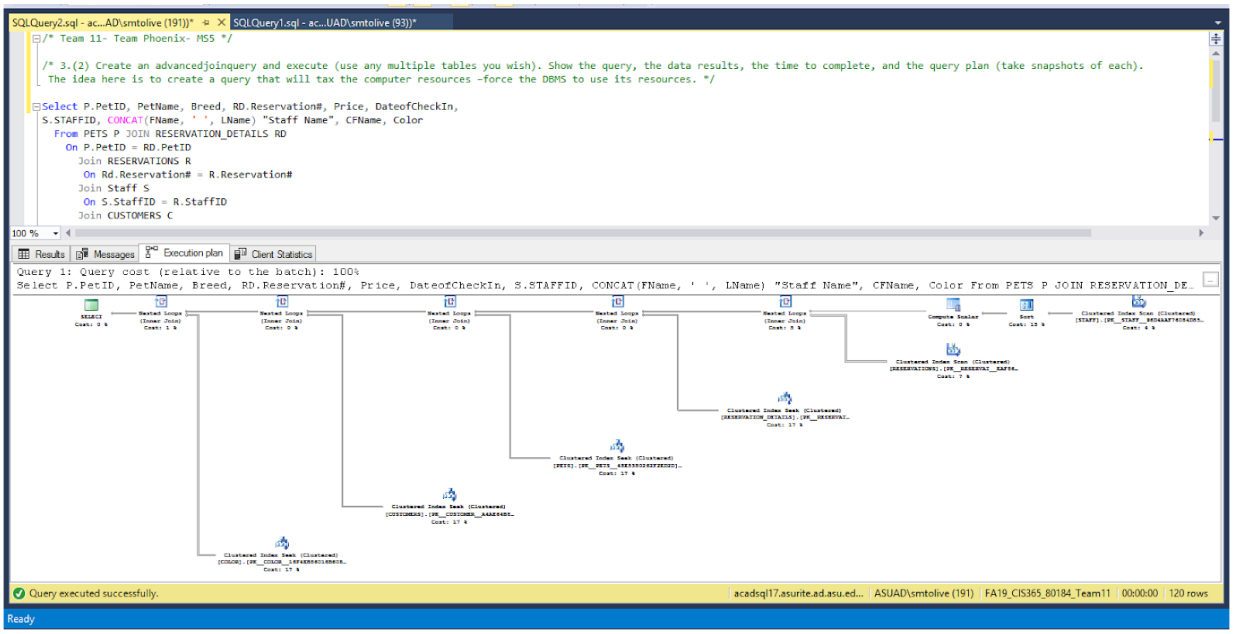
```
/* Team 11- Team Phoenix- M5 */

/* Indexes */

/* 3.(2) Create an advanced join query and execute (use any multiple tables you wish).
Show the query, the data results, the time to complete, and the query plan
(take snapshots of each).  The idea here is to create a query that will tax the
computer resources -force the DBMS to use its resources.    */

SELECT P.PetID, PetName, Breed, RD.Reservation#, Price, DateofCheckIn, DateofCheckOut, S.StaffID, CONCAT(FName,' ', LName) "Staff Name", CFName, Color
  FROM PETS P Join RESERVATION_DETAILS RD
    on P.PetID=RD.PetID
        Join RESERVATIONS R
            On RD.Reservation#=R.Reservation#
        Join Staff S
            on S.StaffID=R.StaffID
        Join Customers C
            on C.CustomerID=P.CustomerID
        Join Color Co
            On Co.PetID=P.PetID
    Order by Fname;
```

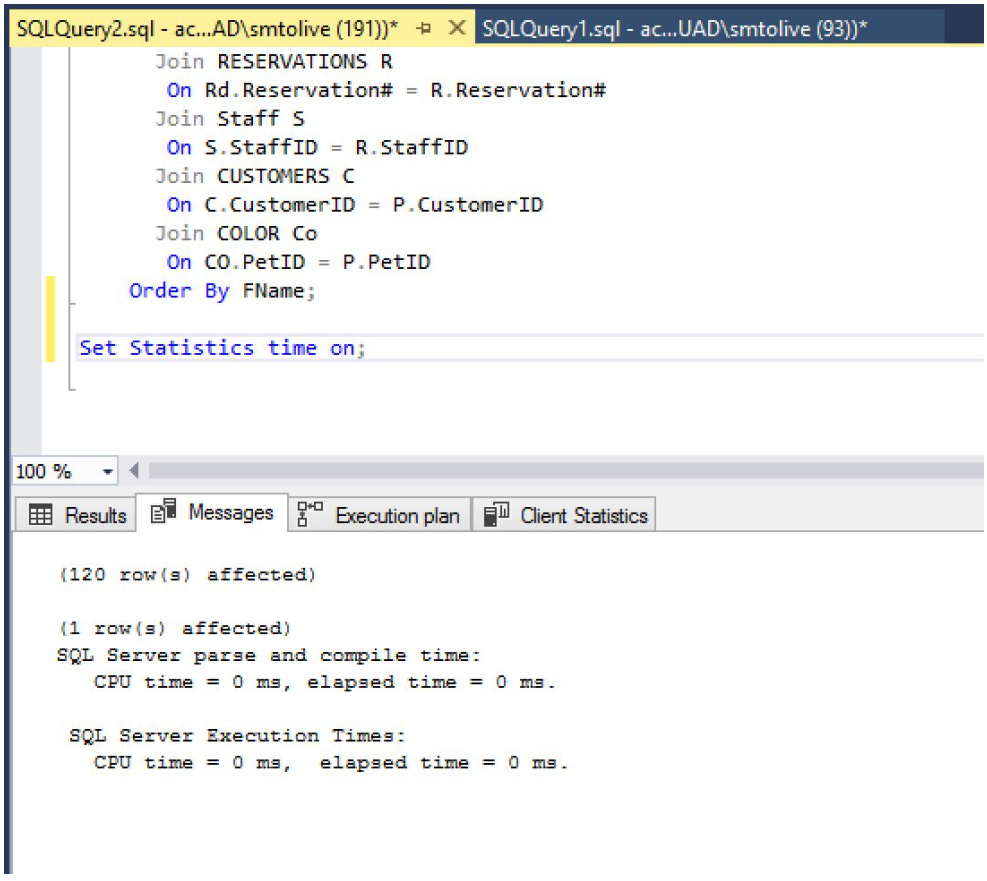
100 %

Results | Messages | Execution plan

	PetID	PetName	Breed	Reservation#	Price	DateofCheckIn	DateofCheckOut	StaffID	Staff Name	CFName	Color
1	F00000001	Wheeler	Ragamuffin cat	10000001	30.00	2019-05-26	2019-05-29	S30000000	Alecxander Ambriz	Rebecca	Grey
2	F00000001	Wheeler	Ragamuffin cat	10000001	30.00	2019-05-26	2019-05-29	S30000000	Alecxander Ambriz	Rebecca	White
3	F00000007	Chase	Japanese Bobtail	10000007	10.00	2019-02-23	2019-02-24	S30000000	Alecxander Ambriz	Emma	Gold
4	F00000007	Chase	Japanese Bobtail	10000007	10.00	2019-02-23	2019-02-24	S30000000	Alecxander Ambriz	Emma	Grey
5	F00000013	Wrath	Norwegian Forest	10000013	20.00	2019-02-06	2019-02-08	S30000000	Alecxander Ambriz	Kevin	Brown
6	F00000013	Wrath	Norwegian Forest	10000013	20.00	2019-02-06	2019-02-08	S30000000	Alecxander Ambriz	Kevin	Gold
7	F00000011	Hamsta	Exotic Shorthair	10000019	30.00	2019-08-15	2019-08-18	S30000000	Alecxander Ambriz	Abigail	Brown
8	F00000011	Hamsta	Exotic Shorthair	10000019	30.00	2019-08-15	2019-08-18	S30000000	Alecxander Ambriz	Abigail	White
9	F00000004	WorldDestroyer	Poe-bob	10000023	20.00	2019-06-18	2019-06-20	S30000000	Alecxander Ambriz	Lillian	Black

3. Before (Cont.)





4. After Optimization