MARMARA UNIVERSITY

FACULTY OF ENGINEERING
COMPUTER SCIENCE & ENGINEERING DEPARTMENT

CSE3055
DATABASE SYSTEMS

Project Step #3
Logical Database Design
&
Mapping & Physical Design
&
Database Implementation

A.Tunahan Cinsoy – 150117062
Enver Aslan – 150115851
Abdullah Gülçür – 150116014

a) Project description: explain what your database project is about.

In our project, we are trying to create a database that is aimed to act as a “Patient Admission System” for our client, who is “Istanbul Bolge Hastanesi”. With the help of this database system, hospital authorities will be able to track their ongoing patient flow, create relationships among entities such as patient-rooms, patient-doctors, patient-staff etc.

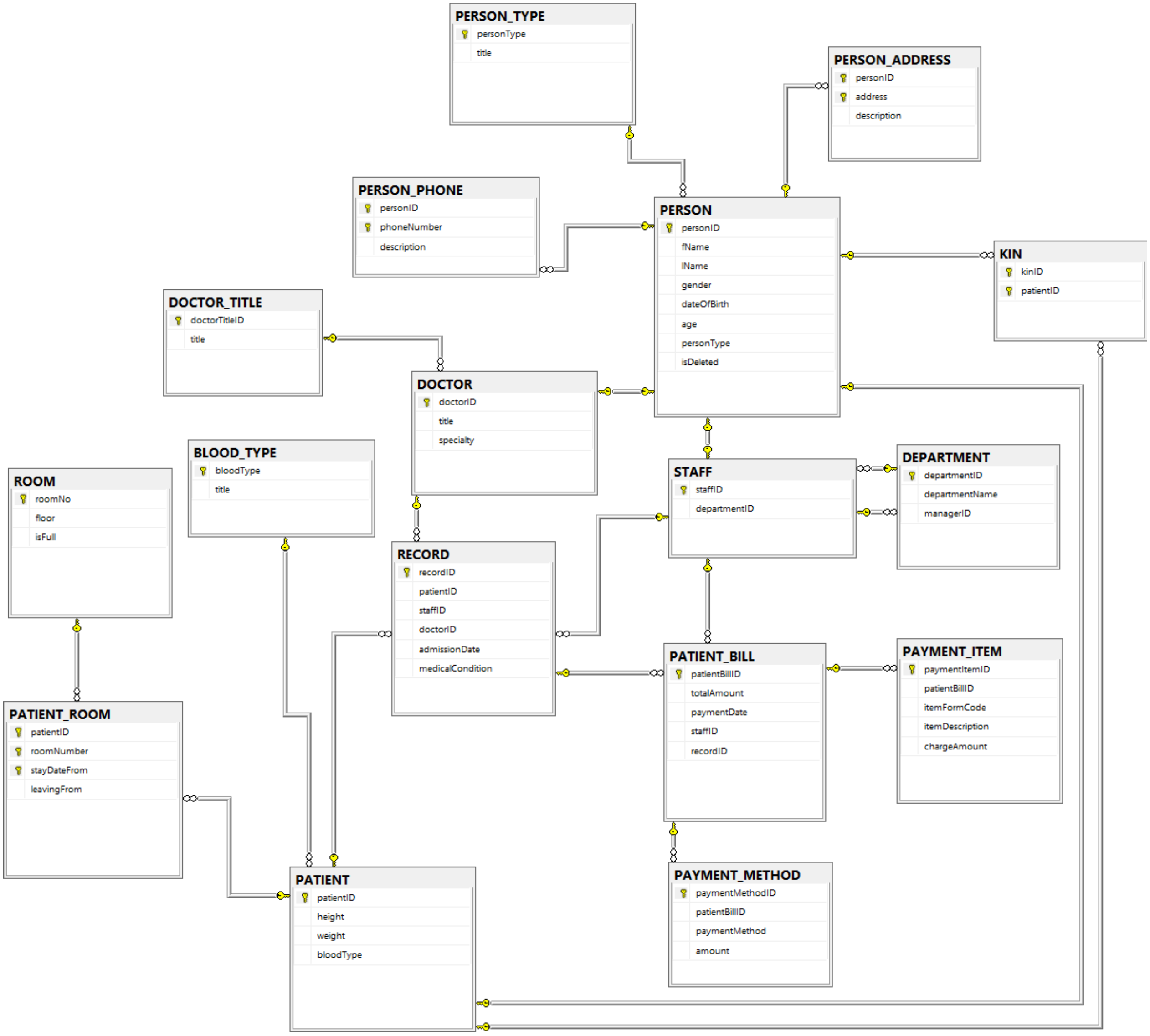
b) Scope: what is included/exclude? Which processes are supported, which ones are not?

In our database implementation, we’ve created various entities that are related with patient admission. Instead of explaining all of them bullet point by bullet point, we can shortly summarize the context as follows: When patient arrives at hospital, one of the staff members registers him/her, and according to his/her condition, staff member attends one of our doctors to our patient. Based on the diagnosis of doctor, patient may stay in one the of the rooms of the hospital. After the medication process, patient gets discharged by paying his/her fee which can be paid with 3 different ways (Credit Card, Cash, Insurance Company).

c) Data and requirements analysis for the database and business processes.

Our client asked us to implement a ‘Patient Admission System’ that is robust, scalable, improvable and efficient. To satisfy their requests, we’ve created several constraints, relationships and also stored procedures. Detailed explanation of these various assets will be explained at further titles.

b) Diagram of whole database.



c) Tables

Our database consists tables called as “DOCTOR, PATIENT, STAFF, KIN” and their supertype table “PERSON”. In addition to these tables, we have: “BLOOD_TYPE, PERSON_PHONE, PERSON_ADDRESS, DOCTOR_TITLE, ROOM, PATIENT_ROOM, RECORD, PATIENT_BILL, PAYMENT_ITEM, PAYMENT_METHOD, DEPARTMENT”.

PERSON

personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted
Bigint	Nvarchar(50)	Nvarchar(50)	Bit	Date	int	Nvarchar(5)	bit

PERSON entity holds information about all human beings that is stored in our database. This entity is a supertype of tables called “DOCTOR, KIN, PATIENT, STAFF”. Primary key of PERSON table is “personID”. “Age” field is computed column based on “dateOfBirth” attribute.

DOCTOR

doctorID	title	specialty
Bigint	Nvarchar(10)	Nvarchar(50)

DOCTOR entity holds information about doctors that are working under our client’s hospital. Primary Key of DOCTOR table is “doctorID” which is derived from PERSON’s “personID”.

STAFF

staffID	departmentID
Bigint	smallint

STAFF entity holds information about staff members that are working under our client’s hospital. Primary Key of STAFF table is “staffID” which is derived from PERSON’s “personID”. Foreign Key of STAFF table is “departmentID” that is coming from DEPARTMENT table.

PATIENT

patientID	Height	Weight	bloodType
Bigint	Smallint	Smallint	Nvarchar(4)

PATIENT entity holds information about patients that have visited our client’s hospital. Primary Key of PATIENT table is “patientID” which is derived from PERSON’s “personID”. Foreign Key of PATIENT table is “bloodType” that is coming from BLOOD_TYPE table.

KIN

kinID	patientID	description
Bigint	bigint	Nvarchar(50)

KIN entity holds information about kin that have 1st or 2nd degree relationship with patients. Primary Key of KIN table is “kinID” which is derived from PERSON’s “personID”. Foreign Key of KIN table is “patientID” that is coming from PATIENT table.

RECORD

recordID	**patientID**	**staffID**	**doctorID**	**admissionDate**	**medCondition**
Bigint	Bigint	Bigint	Bigint	Date	Nvarchar(50)

RECORD entity holds information about medical records of patients. Primary Key of RECORD table is “recordID”. Foreign Keys of RECORD table are “patientID, staffID, doctorID” that are coming from PATIENT, STAFF and DOCTOR table respectively. RECORD entity also has a default constraint for its “date” attribute and trigger named as “INSERT_RECORD_TRIGGER” that inserts information to PATIENT_BILL table.

PATIENT_BILL

patientBillID	**totalAmount**	**paymentDate**	**staffID**	**recordID**
Bigint	Decimal	Datetime	Bigint	Bigint

PATIENT_BILL entity holds information about billing details of patients. Primary Key of PATIENT_BILL table is “patientBillID”. Foreign Keys of RECORD table are “staffID, recordID” that are coming from STAFF and RECORD table respectively. PATIENT_BILL entity also has a default constraint for its “totalAmount” attribute.

PAYMENT_ITEM

paymentItemID	**patientBillID**	**itemFormCode**	**itemDescription**	**chargedAmount**
Bigint	Bigint	Nvarchar(50)	Nvarchar(100)	decimal

PAYMENT_ITEM entity holds information about payment details of patients. Primary Key of PAYMENT_ITEM table is “paymentItemID”. Foreign Keys of RECORD table is “patientBillID” that is coming from PATIENT_BILL table. PAYMENT_ITEM entity also has a unique constraint for its “itemFormCode” attribute, a trigger named as “update_totalAmount” and index constraint for its “itemFormCode” attribute.

PAYMENT_METHOD

paymentMethodID	**patientBillID**	**paymentMethod**	**amount**
Bigint	Bigint	Nvarchar(100)	decimal

PAYMENT_METHOD entity holds information about payment method details of patients. Primary Key of PAYMENT_ITEM table is “paymentMethodID”. Foreign Keys of RECORD table is “patientBillID” that is coming from PATIENT_BILL table. PAYMENT_METHOD entity also has a default constraint for its “amount” attribute.

DEPARTMENT

departmentID	departmentName	managerID
smallint	Nvarchar(50)	bigint

DEPARTMENT entity holds information about various departments of our client’s hospital. Primary Key of DEPARTMENT table is “departmentID”. Foreign Keys of DEPARTMENT table is “managerID” that is coming from STAFF table.

PERSON_ADDRESS

personID	address	description
Bigint	Nvarchar(255)	Nvarchar(50)

PERSON_ADDRESS entity holds information about addresses that are given by patients or kin. Primary Keys of PERSON_ADDRESS table are “personID, address”.

PERSON_PHONE

personID	phoneNumber	description
Bigint	Nvarchar(50)	Nvarchar(50)

PERSON_PHONE entity holds information about phone numbers that are given by people of hospital. Primary Keys of PERSON_PHONE table are “personID, phoneNumber”.

PERSON_TYPE

personType	Title
Nvarchar(5)	Nvarchar(50)

PERSON_TYPE entity holds additional identity information. Primary Key of PERSON_TYPE table is “personType”.

BLOOD_TYPE

bloodType	Title
Nvarchar(4)	Nvarchar(20)

PERSON_TYPE entity holds information about blood types of patients. Primary Keys of BLOOD_TYPE table is “bloodType”.

DOCTOR_TITLE

doctorTitleID	Title
Nvarchar(10)	Nvarchar(50)

DOCTOR_TITLE entity holds information about titles of doctors. Primary Key of DOCTOR_TITLE table is “doctorTitleID”.

ROOM

roomNo	Floor	isFull
Smallint	Nvarchar(50)	bit

ROOM entity holds information about rooms of hospital. Primary Key of ROOM table is “roomNo”.

PATIENT_ROOM

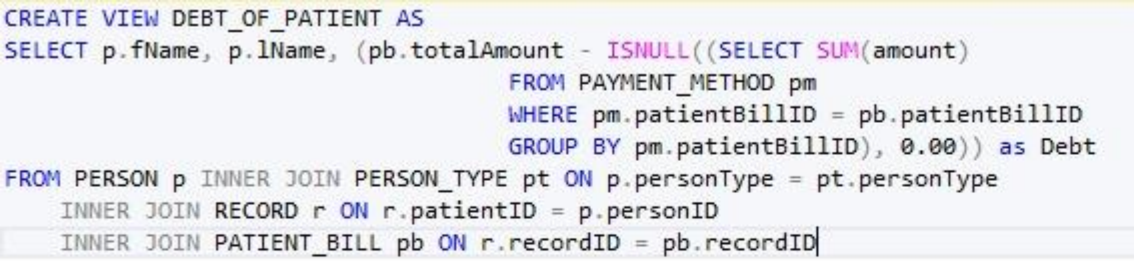
patientID	roomNumber	stayDateFrom	leavingFrom
Bigint	Smallint	Datetime	datetime

ROOM entity holds information about rooms of patients of hospital. Primary Keys of ROOM table are “patientID, roomNumber and stayDateFrom”.

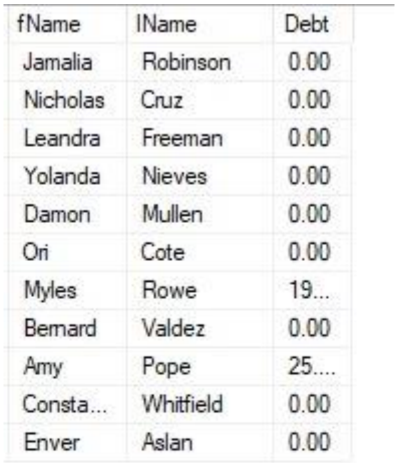
d) Views

We have totally 5 views for our database which are named as “DEBT_OF_PATIENT, DOCTOR_PHONE_ADDRESS, INMATES_WITH_THEIR_DOCTORS, PATIENT_THAT_USES_CREDIT_CARD, PATIENT_KIN_PHONE_NUMBER”.

DEBT_OF_PATIENT: We simply show the debt amount of all patients.

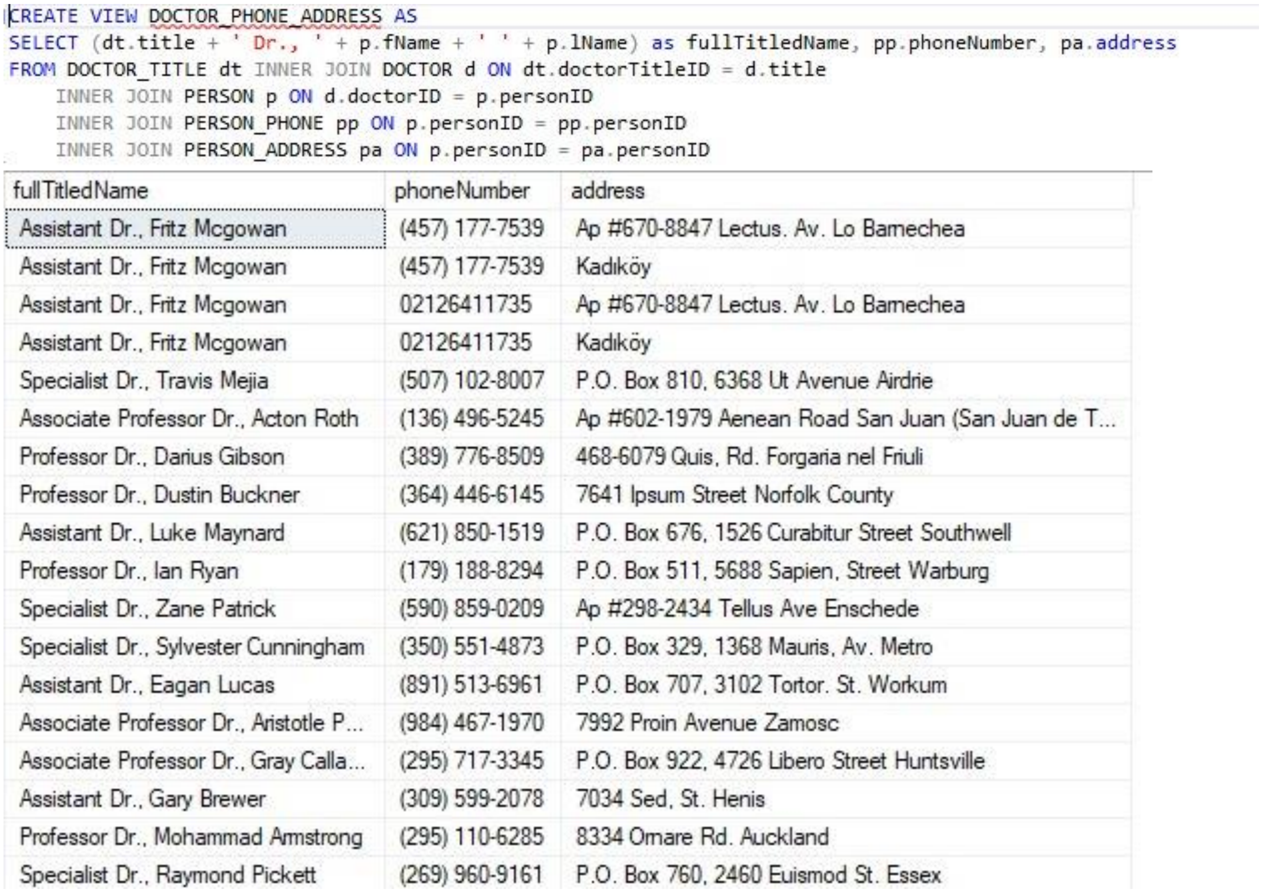


```
CREATE VIEW DEBT_OF_PATIENT AS
SELECT p.fName, p.lName, (pb.totalAmount - ISNULL((SELECT SUM(amount)
                                    FROM PAYMENT_METHOD pm
                                    WHERE pm.patientBillID = pb.patientBillID
                                    GROUP BY pm.patientBillID), 0.00)) as Debt
FROM PERSON p INNER JOIN PERSON_TYPE pt ON p.personType = pt.personType
    INNER JOIN RECORD r ON r.patientID = p.personID
    INNER JOIN PATIENT_BILL pb ON r.recordID = pb.recordID
```



fName	lName	Debt
Jamalia	Robinson	0.00
Nicholas	Cruz	0.00
Leandra	Freeman	0.00
Yolanda	Nieves	0.00
Damon	Mullen	0.00
Ori	Cote	0.00
Myles	Rowe	19...
Bernard	Valdez	0.00
Amy	Pope	25....
Consta...	Whitfield	0.00
Enver	Aslan	0.00

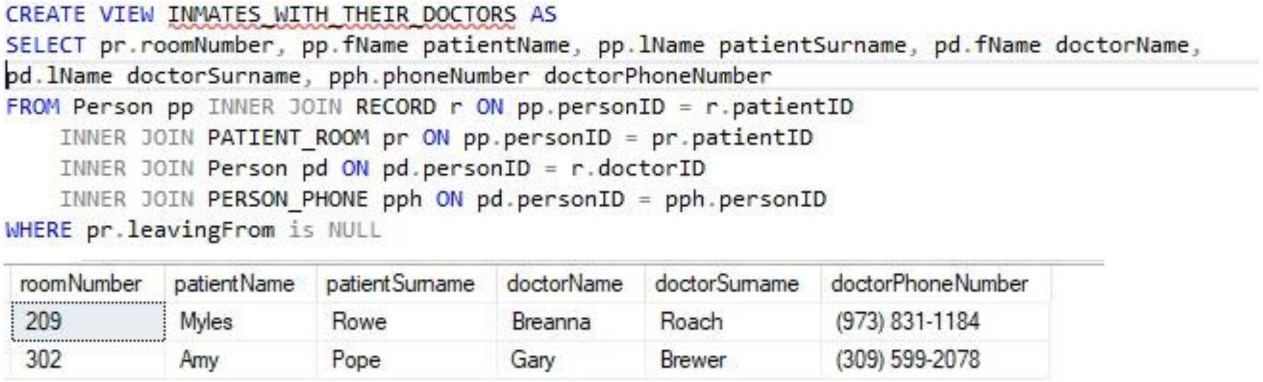
DOCTOR_PHONE_ADDRESS: Used for retrieving phone number and address information of doctors.



```
CREATE VIEW DOCTOR_PHONE_ADDRESS AS
SELECT (dt.title + ' Dr., ' + p.fName + ' ' + p.lName) as fullTitledName, pp.phoneNumber, pa.address
FROM DOCTOR_TITLE dt INNER JOIN DOCTOR d ON dt.doctorTitleID = d.title
    INNER JOIN PERSON p ON d.doctorID = p.personID
    INNER JOIN PERSON_PHONE pp ON p.personID = pp.personID
    INNER JOIN PERSON_ADDRESS pa ON p.personID = pa.personID
```

fullTitledName	phoneNumber	address
Assistant Dr., Fritz Mcgowan	(457) 177-7539	Ap #670-8847 Lectus. Av. Lo Barnechea
Assistant Dr., Fritz Mcgowan	(457) 177-7539	Kadıköy
Assistant Dr., Fritz Mcgowan	02126411735	Ap #670-8847 Lectus. Av. Lo Barnechea
Assistant Dr., Fritz Mcgowan	02126411735	Kadıköy
Specialist Dr., Travis Mejia	(507) 102-8007	P.O. Box 810, 6368 Ut Avenue Airdrie
Associate Professor Dr., Acton Roth	(136) 496-5245	Ap #602-1979 Aenean Road San Juan (San Juan de T...
Professor Dr., Darius Gibson	(389) 776-8509	468-6079 Quis, Rd. Forgaria nel Friuli
Professor Dr., Dustin Buckner	(364) 446-6145	7641 Ipsum Street Norfolk County
Assistant Dr., Luke Maynard	(621) 850-1519	P.O. Box 676, 1526 Curabitur Street Southwell
Professor Dr., Ian Ryan	(179) 188-8294	P.O. Box 511, 5688 Sapien, Street Warburg
Specialist Dr., Zane Patrick	(590) 859-0209	Ap #298-2434 Tellus Ave Enschede
Specialist Dr., Sylvester Cunningham	(350) 551-4873	P.O. Box 329, 1368 Mauris, Av. Metro
Assistant Dr., Eagan Lucas	(891) 513-6961	P.O. Box 707, 3102 Tortor. St. Workum
Associate Professor Dr., Aristotle P...	(984) 467-1970	7992 Proin Avenue Zamosc
Associate Professor Dr., Gray Calla...	(295) 717-3345	P.O. Box 922, 4726 Libero Street Huntsville
Assistant Dr., Gary Brewer	(309) 599-2078	7034 Sed, St. Henis
Professor Dr., Mohammad Armstrong	(295) 110-6285	8334 Ornare Rd. Auckland
Specialist Dr., Raymond Pickett	(269) 960-9161	P.O. Box 760, 2460 Euismod St. Essex

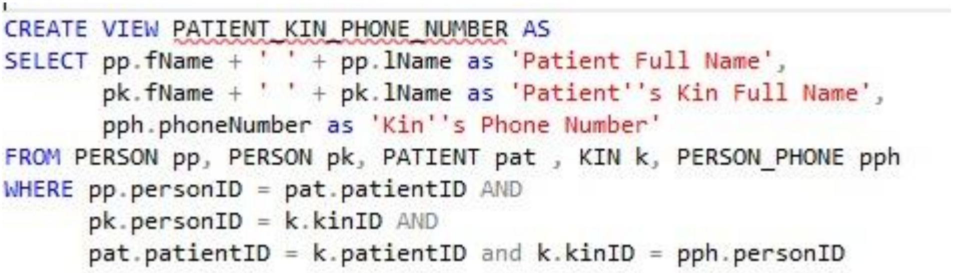
INMATES_WITH_THEIR_DOCTORS: Getting personal information of patients who are currently staying at one of hospital’s rooms.



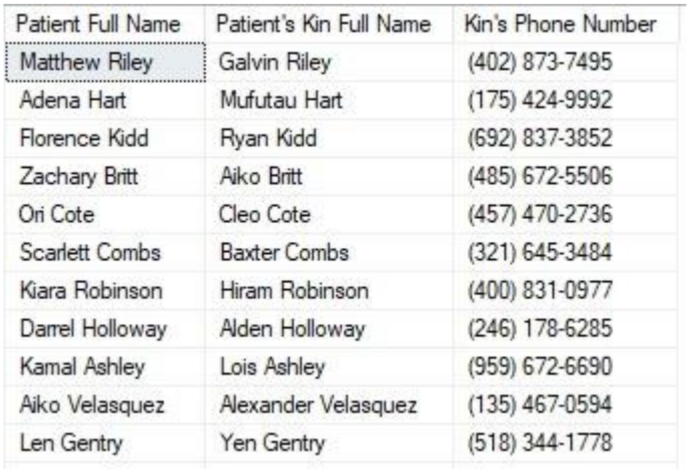
```
CREATE VIEW INMATES_WITH_THEIR_DOCTORS AS
SELECT pr.roomNumber, pp.fName patientName, pp.lName patientSurname, pd.fName doctorName,
pd.lName doctorSurname, pph.phoneNumber doctorPhoneNumber
FROM Person pp INNER JOIN RECORD r ON pp.personID = r.patientID
    INNER JOIN PATIENT_ROOM pr ON pp.personID = pr.patientID
    INNER JOIN Person pd ON pd.personID = r.doctorID
    INNER JOIN PERSON_PHONE pph ON pd.personID = pph.personID
WHERE pr.leavingFrom is NULL
```

roomNumber	patientName	patientSurname	doctorName	doctorSurname	doctorPhoneNumber
209	Myles	Rowe	Breanna	Roach	(973) 831-1184
302	Amy	Pope	Gary	Brewer	(309) 599-2078

PATIENT_KIN_PHONE_NUMBER: Retrieving phone number of patient’s kin members.

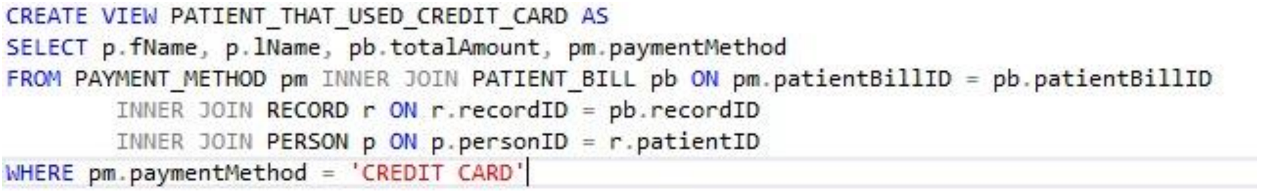


```
CREATE VIEW PATIENT_KIN_PHONE_NUMBER AS
SELECT pp.fName + ' ' + pp.lName as 'Patient Full Name',
       pk.fName + ' ' + pk.lName as 'Patient''s Kin Full Name',
       pph.phoneNumber as 'Kin''s Phone Number'
FROM PERSON pp, PERSON pk, PATIENT pat , KIN k, PERSON_PHONE pph
WHERE pp.personID = pat.patientID AND
      pk.personID = k.kinID AND
      pat.patientID = k.patientID and k.kinID = pph.personID
```

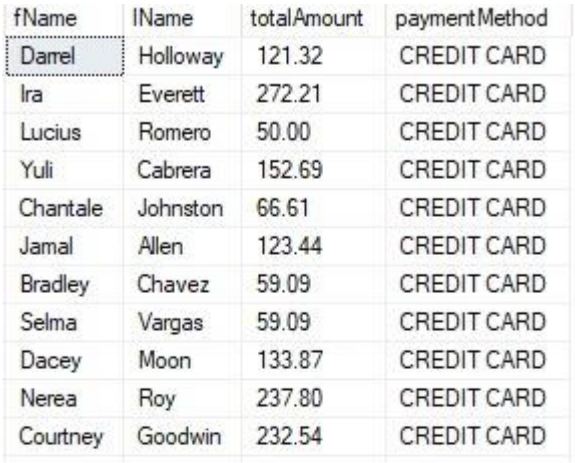


Patient Full Name	Patient's Kin Full Name	Kin's Phone Number
Matthew Riley	Galvin Riley	(402) 873-7495
Adena Hart	Mufutau Hart	(175) 424-9992
Florence Kidd	Ryan Kidd	(692) 837-3852
Zachary Britt	Aiko Britt	(485) 672-5506
Ori Cote	Cleo Cote	(457) 470-2736
Scarlett Combs	Baxter Combs	(321) 645-3484
Kiara Robinson	Hiram Robinson	(400) 831-0977
Darrel Holloway	Alden Holloway	(246) 178-6285
Kamal Ashley	Lois Ashley	(959) 672-6690
Aiko Velasquez	Alexander Velasquez	(135) 467-0594
Len Gentry	Yen Gentry	(518) 344-1778

PATIENT_THAT_USED_CREDIT_CARD: Views patients that have used credit card option for payment.



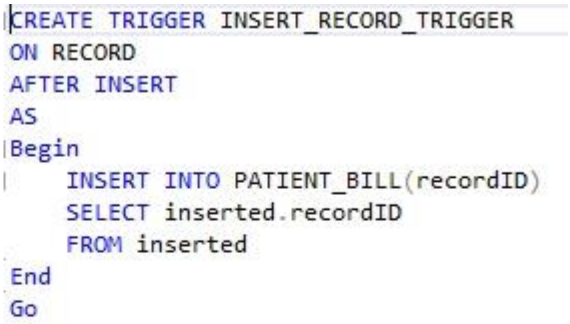
```
CREATE VIEW PATIENT_THAT_USED_CREDIT_CARD AS
SELECT p.fName, p.lName, pb.totalAmount, pm.paymentMethod
FROM PAYMENT_METHOD pm INNER JOIN PATIENT_BILL pb ON pm.patientBillID = pb.patientBillID
        INNER JOIN RECORD r ON r.recordID = pb.recordID
        INNER JOIN PERSON p ON p.personID = r.patientID
WHERE pm.paymentMethod = 'CREDIT CARD'
```



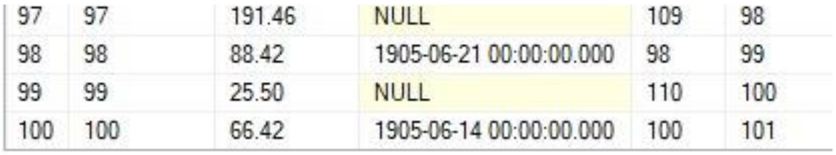
fName	lName	totalAmount	paymentMethod
Darrel	Holloway	121.32	CREDIT CARD
Ira	Everett	272.21	CREDIT CARD
Lucius	Romero	50.00	CREDIT CARD
Yuli	Cabrera	152.69	CREDIT CARD
Chantale	Johnston	66.61	CREDIT CARD
Jamal	Allen	123.44	CREDIT CARD
Bradley	Chavez	59.09	CREDIT CARD
Selma	Vargas	59.09	CREDIT CARD
Dacey	Moon	133.87	CREDIT CARD
Nerea	Roy	237.80	CREDIT CARD
Courtney	Goodwin	232.54	CREDIT CARD

e) Triggers
   We have two triggers in our database named as "INSERT_RECORD_TRIGGER, UPDATE_TOTAL_AMOUNT".

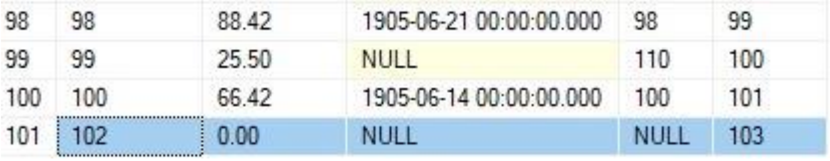
INSERT_RECORD_TRIGGER: When a creation process occurs at RECORD table, that change also affects PATIENT_BILL table. In other words, when we create a registration for a patient, database automatically creates a bill record for that patient.



```
CREATE TRIGGER INSERT_RECORD_TRIGGER
ON RECORD
AFTER INSERT
AS
Begin
    INSERT INTO PATIENT_BILL(recordID)
    SELECT inserted.recordID
    FROM inserted
End
Go
```

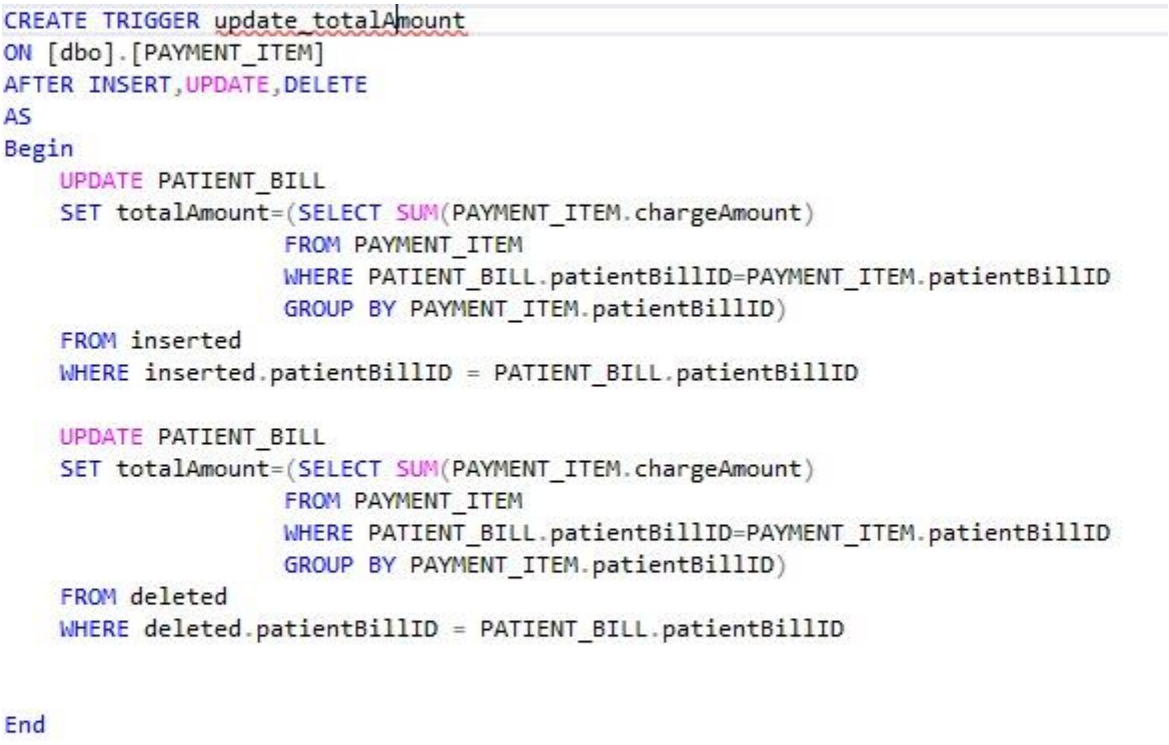


97	97	191.46	NULL	109	98
98	98	88.42	1905-06-21 00:00:00.000	98	99
99	99	25.50	NULL	110	100
100	100	66.42	1905-06-14 00:00:00.000	100	101

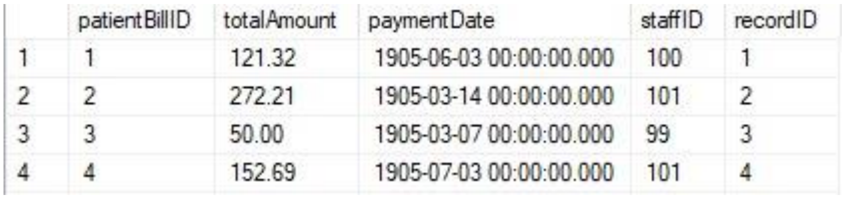


98	98	88.42	1905-06-21 00:00:00.000	98	99
99	99	25.50	NULL	110	100
100	100	66.42	1905-06-14 00:00:00.000	100	101
101	102	0.00	NULL	NULL	103

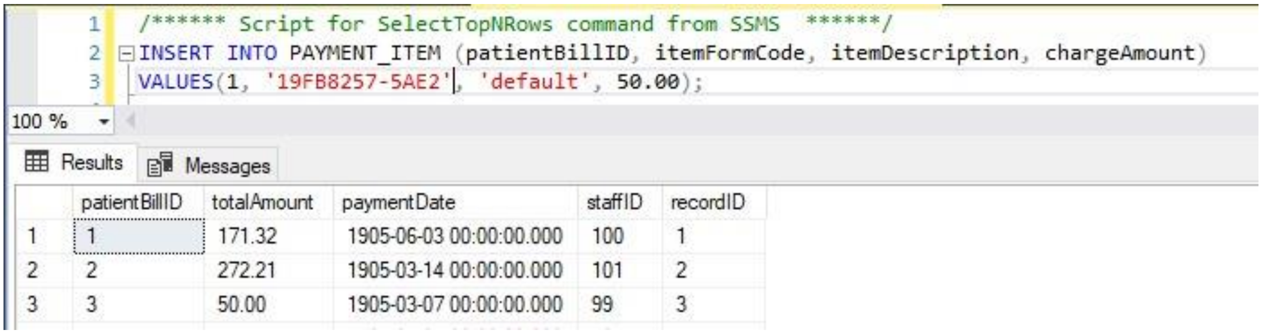
UPDATE_TOTAL_AMOUNT: When there is a change occurs at PAYMENT_ITEM, that change also affects PATIENT_BILL table. In other words, total amount that patient is going to pay will be updated.



```
CREATE TRIGGER update_totalAmount
ON [dbo].[PAYMENT_ITEM]
AFTER INSERT,UPDATE,DELETE
AS
Begin
    UPDATE PATIENT_BILL
    SET totalAmount=(SELECT SUM(PAYMENT_ITEM.chargeAmount)
                    FROM PAYMENT_ITEM
                    WHERE PATIENT_BILL.patientBillID=PAYMENT_ITEM.patientBillID
                    GROUP BY PAYMENT_ITEM.patientBillID)
    FROM inserted
    WHERE inserted.patientBillID = PATIENT_BILL.patientBillID

    UPDATE PATIENT_BILL
    SET totalAmount=(SELECT SUM(PAYMENT_ITEM.chargeAmount)
                    FROM PAYMENT_ITEM
                    WHERE PATIENT_BILL.patientBillID=PAYMENT_ITEM.patientBillID
                    GROUP BY PAYMENT_ITEM.patientBillID)
    FROM deleted
    WHERE deleted.patientBillID = PATIENT_BILL.patientBillID


End
```



	patientBillID	totalAmount	paymentDate	staffID	recordID
1	1	121.32	1905-06-03 00:00:00.000	100	1
2	2	272.21	1905-03-14 00:00:00.000	101	2
3	3	50.00	1905-03-07 00:00:00.000	99	3
4	4	152.69	1905-07-03 00:00:00.000	101	4



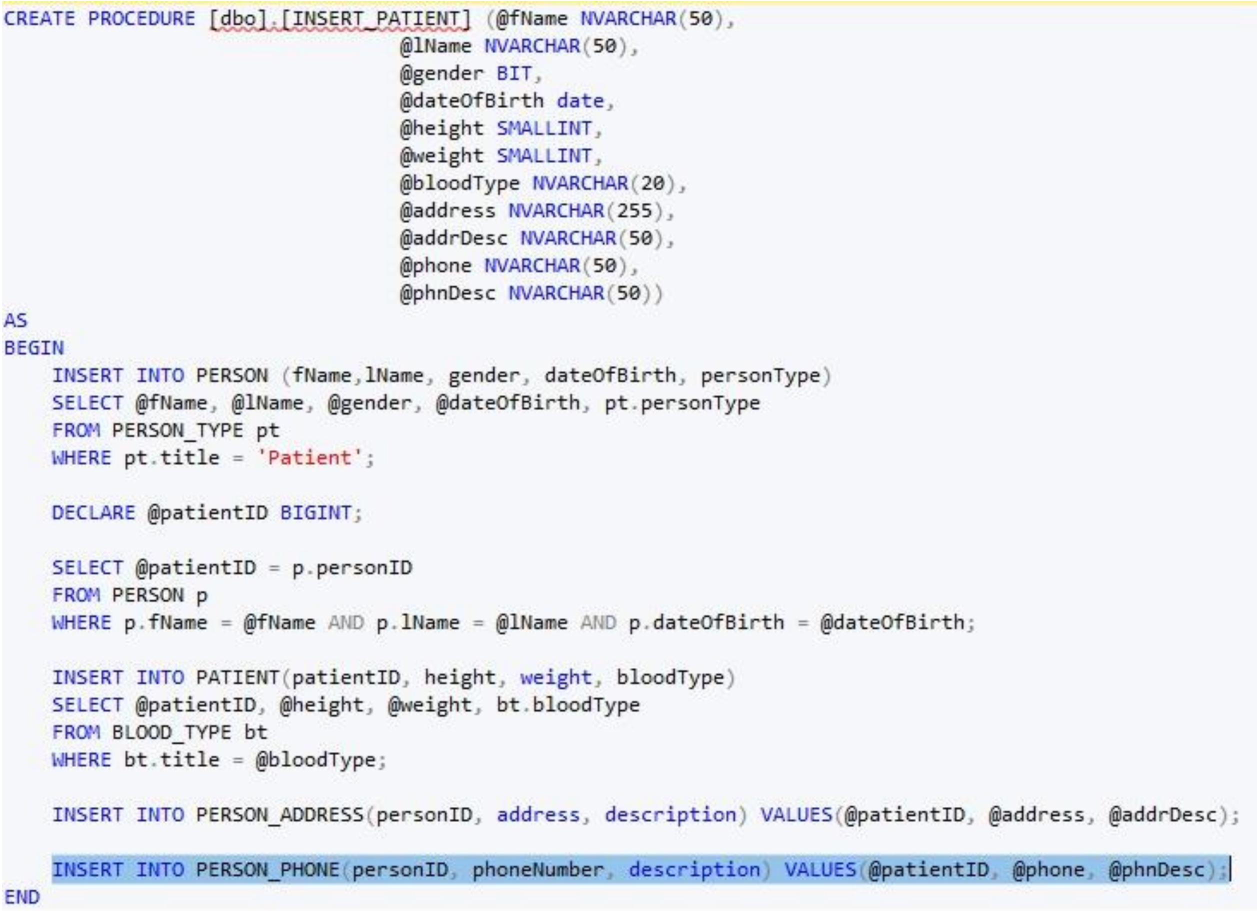
```
/****** Script for SelectTopNRows command from SSMS  ******/
INSERT INTO PAYMENT_ITEM (patientBillID, itemFormCode, itemDescription, chargeAmount)
VALUES(1, '19FB8257-5AE2', 'default', 50.00);
```

	patientBillID	totalAmount	paymentDate	staffID	recordID
1	1	171.32	1905-06-03 00:00:00.000	100	1
2	2	272.21	1905-03-14 00:00:00.000	101	2
3	3	50.00	1905-03-07 00:00:00.000	99	3

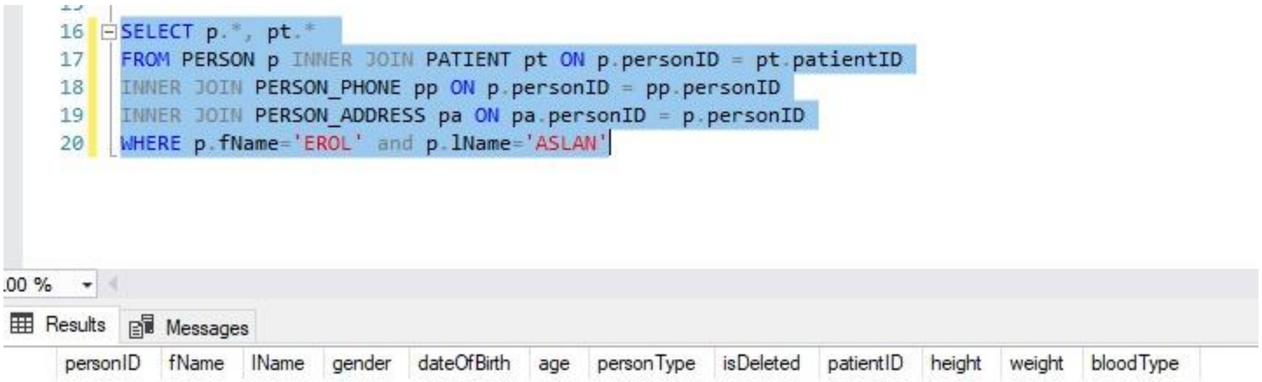
f) Stored procedures

We have totally 11 stored procedures called as “INSERT_PATIENT, INSERT_DOCTOR, INSERT_STAFF, INSERT_KIN, DELETE_PERSON, INSERT_PAYMENT_ITEM, INSERT_PAYMENT_METHOD, DELETE_KIN, UPDATE_DEPARTMENT_MANAGER, INSERT_PHONE, INSERT_ADDRESS”.

INSERT_PATIENT: Procedure for inserting a patient to database.

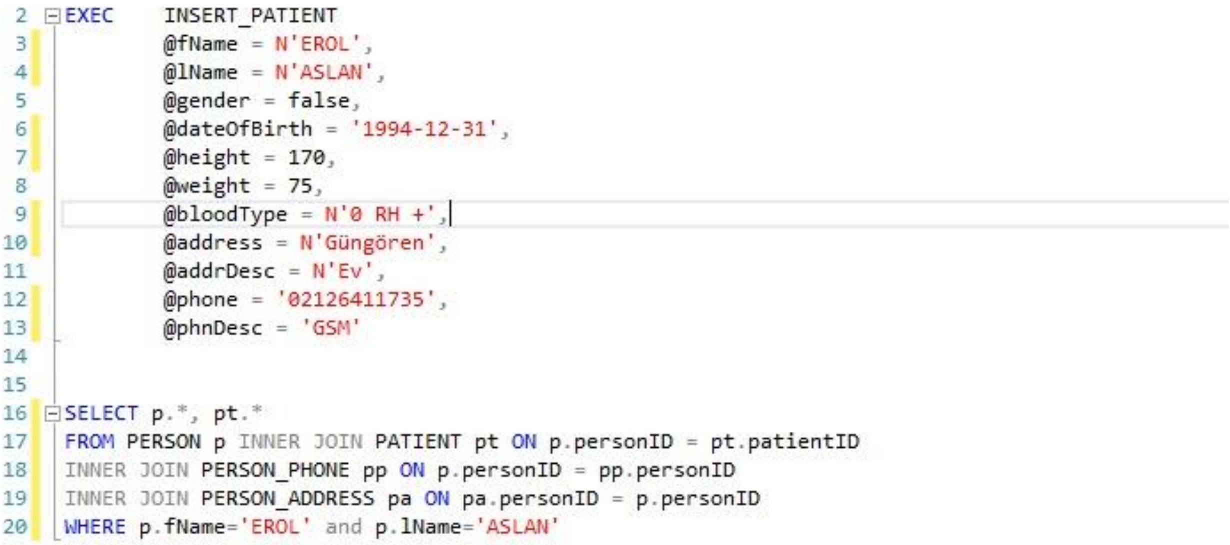


```
CREATE PROCEDURE [dbo].[INSERT_PATIENT] (@fName NVARCHAR(50),
                                 @lName NVARCHAR(50),
                                 @gender BIT,
                                 @dateOfBirth date,
                                 @height SMALLINT,
                                 @weight SMALLINT,
                                 @bloodType NVARCHAR(20),
                                 @address NVARCHAR(255),
                                 @addrDesc NVARCHAR(50),
                                 @phone NVARCHAR(50),
                                 @phnDesc NVARCHAR(50))
AS
BEGIN
    INSERT INTO PERSON (fName,lName, gender, dateOfBirth, personType)
    SELECT @fName, @lName, @gender, @dateOfBirth, pt.personType
    FROM PERSON_TYPE pt
    WHERE pt.title = 'Patient';

    DECLARE @patientID BIGINT;

    SELECT @patientID = p.personID
    FROM PERSON p
    WHERE p.fName = @fName AND p.lName = @lName AND p.dateOfBirth = @dateOfBirth;

    INSERT INTO PATIENT(patientID, height, weight, bloodType)
    SELECT @patientID, @height, @weight, bt.bloodType
    FROM BLOOD_TYPE bt
    WHERE bt.title = @bloodType;

    INSERT INTO PERSON_ADDRESS(personID, address, description) VALUES(@patientID, @address, @addrDesc);

    INSERT INTO PERSON_PHONE(personID, phoneNumber, description) VALUES(@patientID, @phone, @phnDesc);
END
```

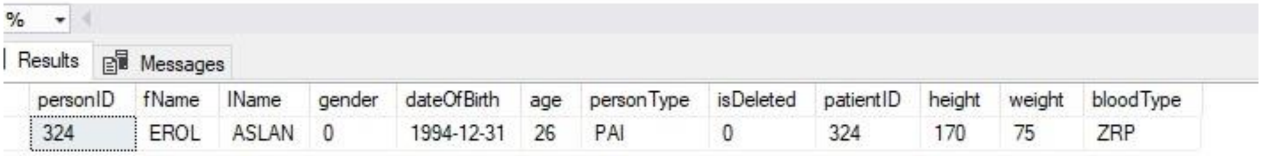


```
SELECT p.*, pt.*
FROM PERSON p INNER JOIN PATIENT pt ON p.personID = pt.patientID
INNER JOIN PERSON_PHONE pp ON p.personID = pp.personID
INNER JOIN PERSON_ADDRESS pa ON pa.personID = p.personID
WHERE p.fName='EROL' and p.lName='ASLAN'
```

personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	patientID	height	weight	bloodType

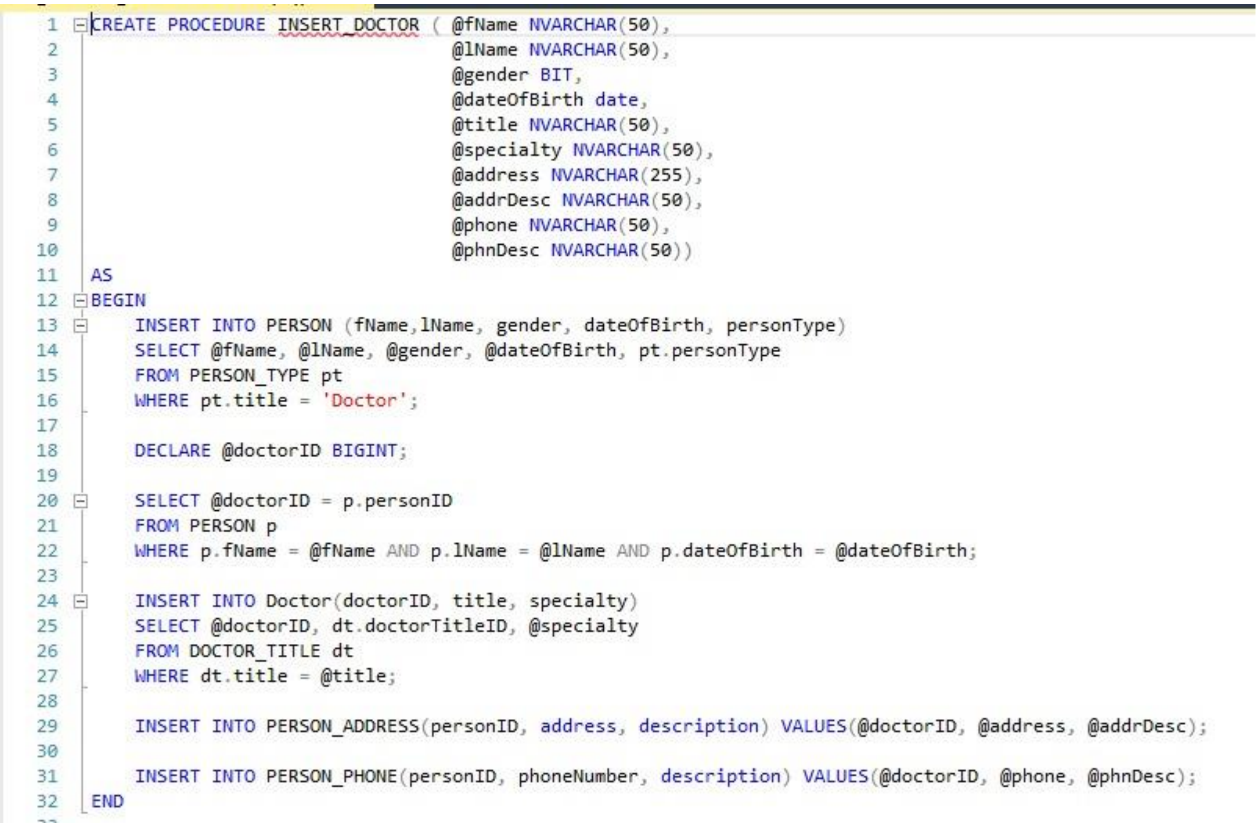


```sql
EXEC    INSERT_PATIENT
        @fName = N'EROL',
        @lName = N'ASLAN',
        @gender = false,
        @dateOfBirth = '1994-12-31',
        @height = 170,
        @weight = 75,
        @bloodType = N'0 RH +',
        @address = N'Güngören',
        @addrDesc = N'Ev',
        @phone = '02126411735',
        @phnDesc = 'GSM'


SELECT p.*, pt.*
FROM PERSON p INNER JOIN PATIENT pt ON p.personID = pt.patientID
INNER JOIN PERSON_PHONE pp ON p.personID = pp.personID
INNER JOIN PERSON_ADDRESS pa ON pa.personID = p.personID
WHERE p.fName='EROL' and p.lName='ASLAN'
```

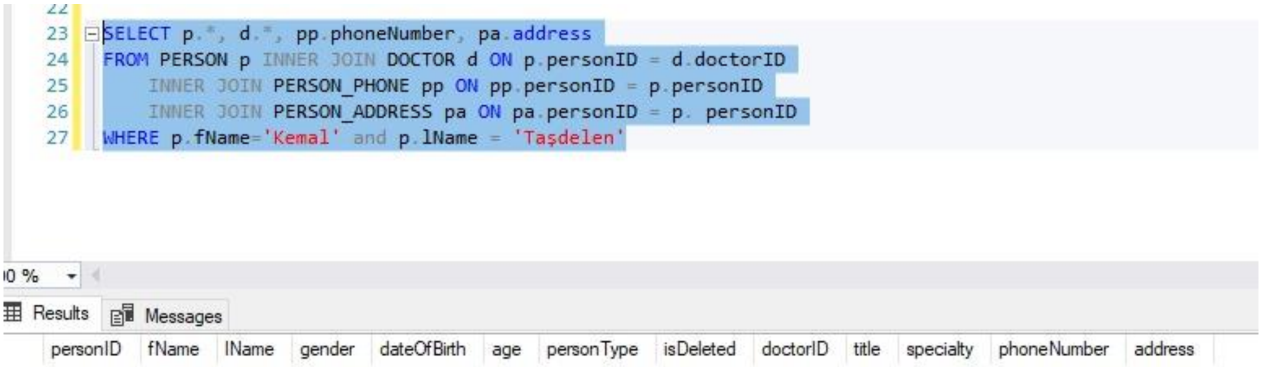


personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	patientID	height	weight	bloodType
324	EROL	ASLAN	0	1994-12-31	26	PAI	0	324	170	75	ZRP

INSERT_DOCTOR: Procedure for inserting a doctor to database.

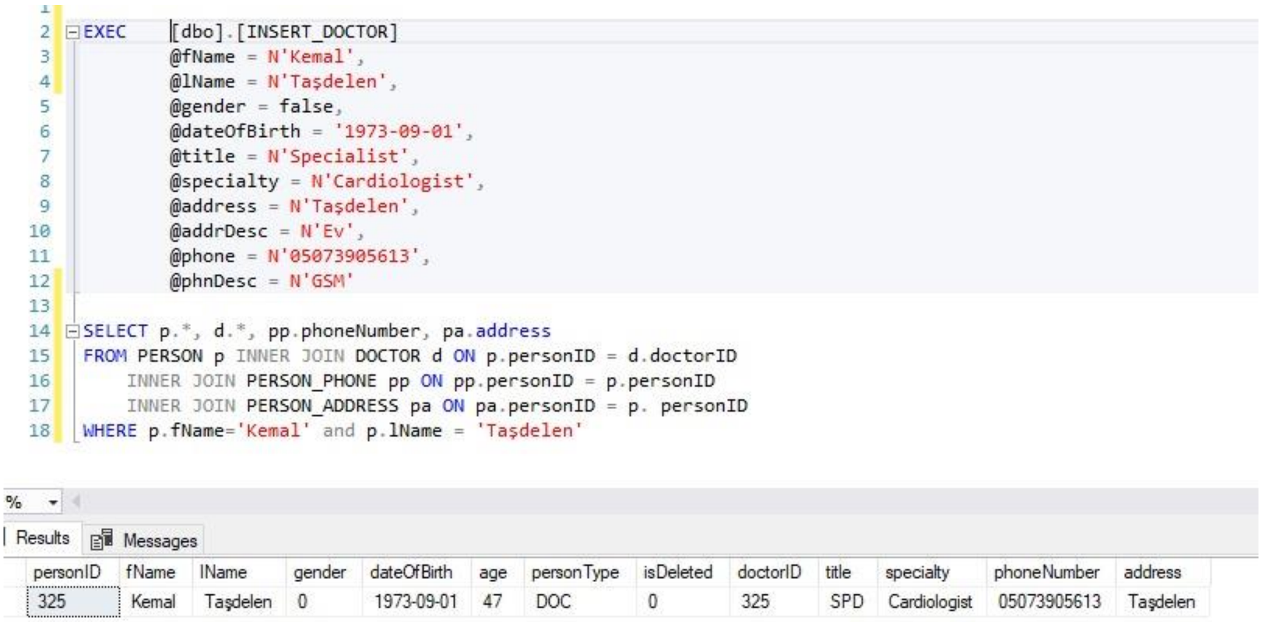


```sql
CREATE PROCEDURE INSERT_DOCTOR ( @fName NVARCHAR(50),
                                 @lName NVARCHAR(50),
                                 @gender BIT,
                                 @dateOfBirth date,
                                 @title NVARCHAR(50),
                                 @specialty NVARCHAR(50),
                                 @address NVARCHAR(255),
                                 @addrDesc NVARCHAR(50),
                                 @phone NVARCHAR(50),
                                 @phnDesc NVARCHAR(50))
AS
BEGIN
    INSERT INTO PERSON (fName,lName, gender, dateOfBirth, personType)
    SELECT @fName, @lName, @gender, @dateOfBirth, pt.personType
    FROM PERSON_TYPE pt
    WHERE pt.title = 'Doctor';

    DECLARE @doctorID BIGINT;

    SELECT @doctorID = p.personID
    FROM PERSON p
    WHERE p.fName = @fName AND p.lName = @lName AND p.dateOfBirth = @dateOfBirth;

    INSERT INTO Doctor(doctorID, title, specialty)
    SELECT @doctorID, dt.doctorTitleID, @specialty
    FROM DOCTOR_TITLE dt
    WHERE dt.title = @title;

    INSERT INTO PERSON_ADDRESS(personID, address, description) VALUES(@doctorID, @address, @addrDesc);

    INSERT INTO PERSON_PHONE(personID, phoneNumber, description) VALUES(@doctorID, @phone, @phnDesc);
END
```



```sql
SELECT p.*, d.*, pp.phoneNumber, pa.address
FROM PERSON p INNER JOIN DOCTOR d ON p.personID = d.doctorID
    INNER JOIN PERSON_PHONE pp ON pp.personID = p.personID
    INNER JOIN PERSON_ADDRESS pa ON pa.personID = p. personID
WHERE p.fName='Kemal' and p.lName = 'Taşdelen'
```

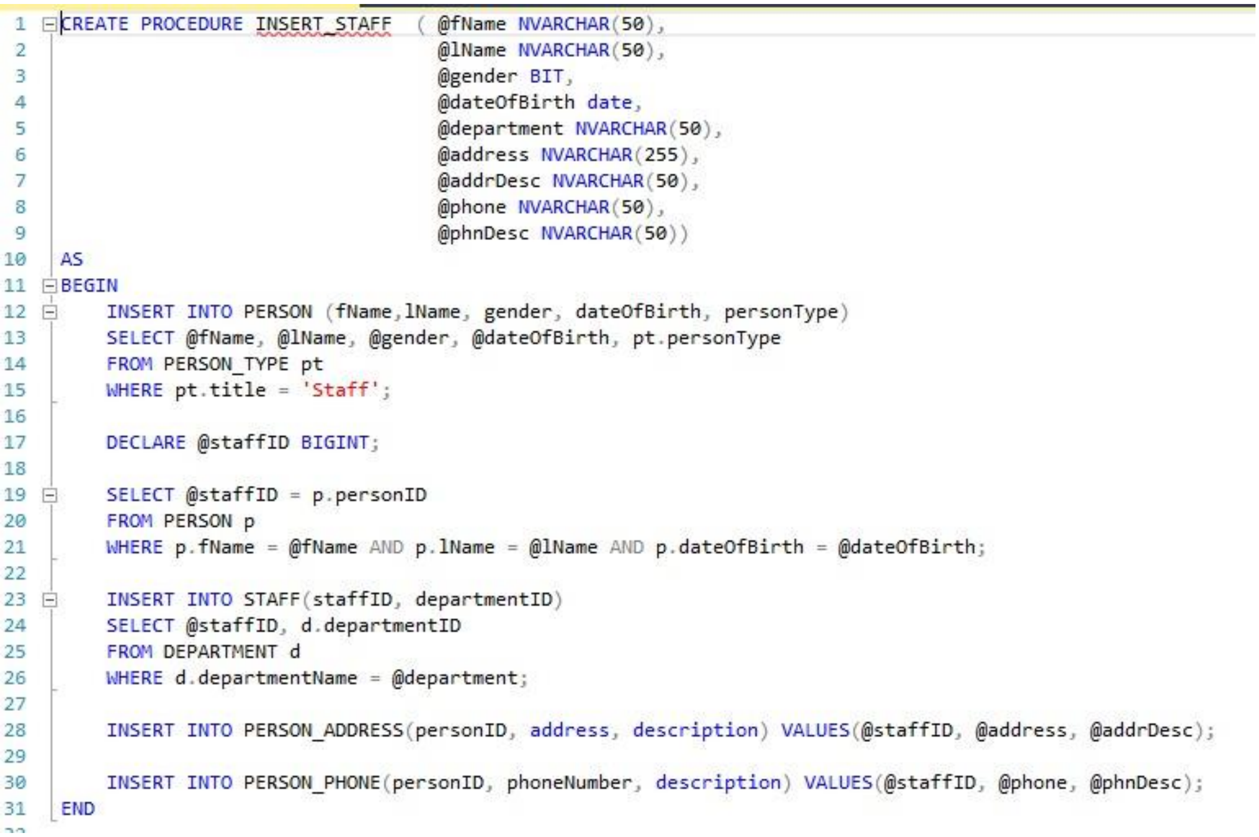
personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	doctorID	title	specialty	phoneNumber	address



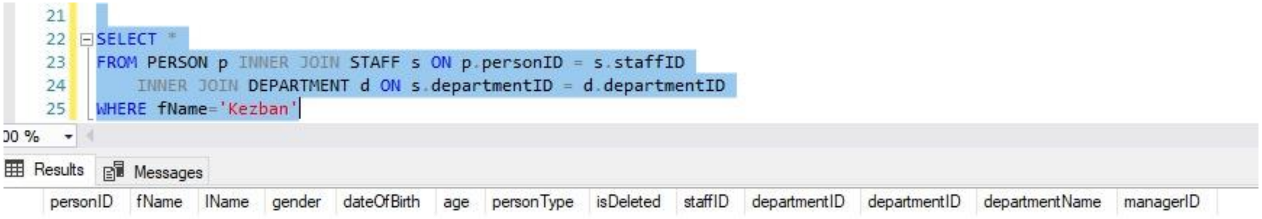
```sql
EXEC    [dbo].[INSERT_DOCTOR]
        @fName = N'Kemal',
        @lName = N'Taşdelen',
        @gender = false,
        @dateOfBirth = '1973-09-01',
        @title = N'Specialist',
        @specialty = N'Cardiologist',
        @address = N'Taşdelen',
        @addrDesc = N'Ev',
        @phone = N'05073905613',
        @phnDesc = N'GSM'

SELECT p.*, d.*, pp.phoneNumber, pa.address
FROM PERSON p INNER JOIN DOCTOR d ON p.personID = d.doctorID
    INNER JOIN PERSON_PHONE pp ON pp.personID = p.personID
    INNER JOIN PERSON_ADDRESS pa ON pa.personID = p. personID
WHERE p.fName='Kemal' and p.lName = 'Taşdelen'
```

personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	doctorID	title	specialty	phoneNumber	address
325	Kemal	Taşdelen	0	1973-09-01	47	DOC	0	325	SPD	Cardiologist	05073905613	Taşdelen

INSERT_STAFF: Procedure for inserting a staff member to database.

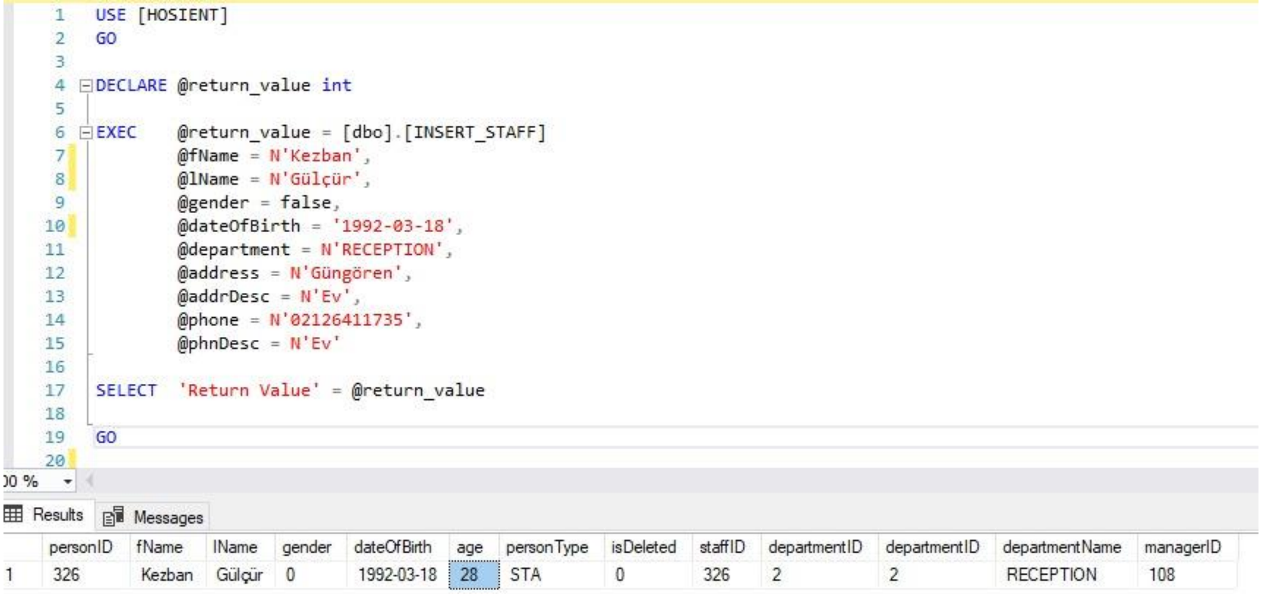


```sql
CREATE PROCEDURE INSERT_STAFF  ( @fName NVARCHAR(50),
                                 @lName NVARCHAR(50),
                                 @gender BIT,
                                 @dateOfBirth date,
                                 @department NVARCHAR(50),
                                 @address NVARCHAR(255),
                                 @addrDesc NVARCHAR(50),
                                 @phone NVARCHAR(50),
                                 @phnDesc NVARCHAR(50))
AS
BEGIN
    INSERT INTO PERSON (fName,lName, gender, dateOfBirth, personType)
    SELECT @fName, @lName, @gender, @dateOfBirth, pt.personType
    FROM PERSON_TYPE pt
    WHERE pt.title = 'Staff';

    DECLARE @staffID BIGINT;

    SELECT @staffID = p.personID
    FROM PERSON p
    WHERE p.fName = @fName AND p.lName = @lName AND p.dateOfBirth = @dateOfBirth;

    INSERT INTO STAFF(staffID, departmentID)
    SELECT @staffID, d.departmentID
    FROM DEPARTMENT d
    WHERE d.departmentName = @department;

    INSERT INTO PERSON_ADDRESS(personID, address, description) VALUES(@staffID, @address, @addrDesc);

    INSERT INTO PERSON_PHONE(personID, phoneNumber, description) VALUES(@staffID, @phone, @phnDesc);
END
```



```sql

SELECT *
FROM PERSON p INNER JOIN STAFF s ON p.personID = s.staffID
    INNER JOIN DEPARTMENT d ON s.departmentID = d.departmentID
WHERE fName='Kezban'
```

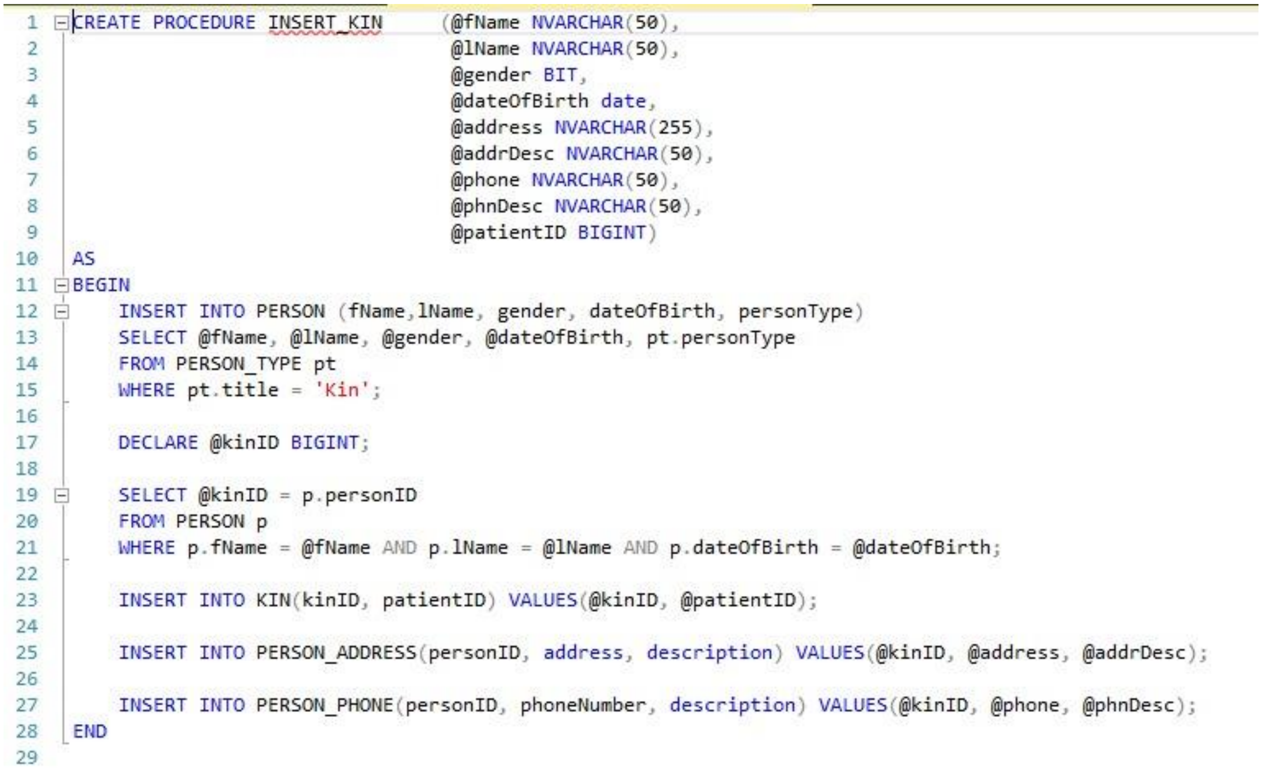
personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	staffID	departmentID	departmentID	departmentName	managerID



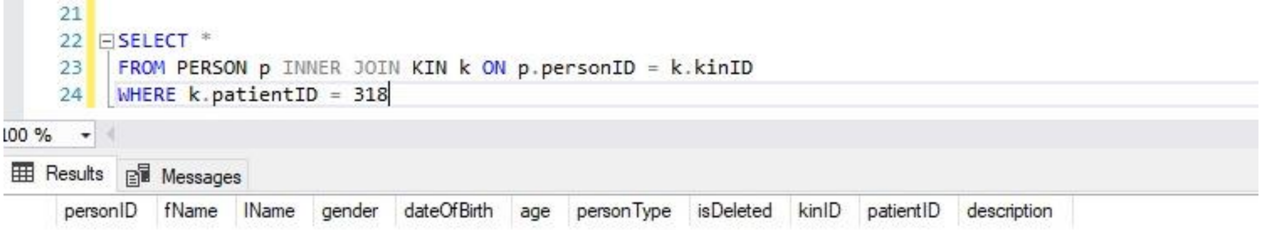
```sql
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[INSERT_STAFF]
        @fName = N'Kezban',
        @lName = N'Gülçür',
        @gender = false,
        @dateOfBirth = '1992-03-18',
        @department = N'RECEPTION',
        @address = N'Güngören',
        @addrDesc = N'Ev',
        @phone = N'02126411735',
        @phnDesc = N'Ev'

SELECT  'Return Value' = @return_value

GO
```

	personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	staffID	departmentID	departmentID	departmentName	managerID
1	326	Kezban	Gülçür	0	1992-03-18	28	STA	0	326	2	2	RECEPTION	108

INSERT_KIN: Procedure for inserting a kin to database.

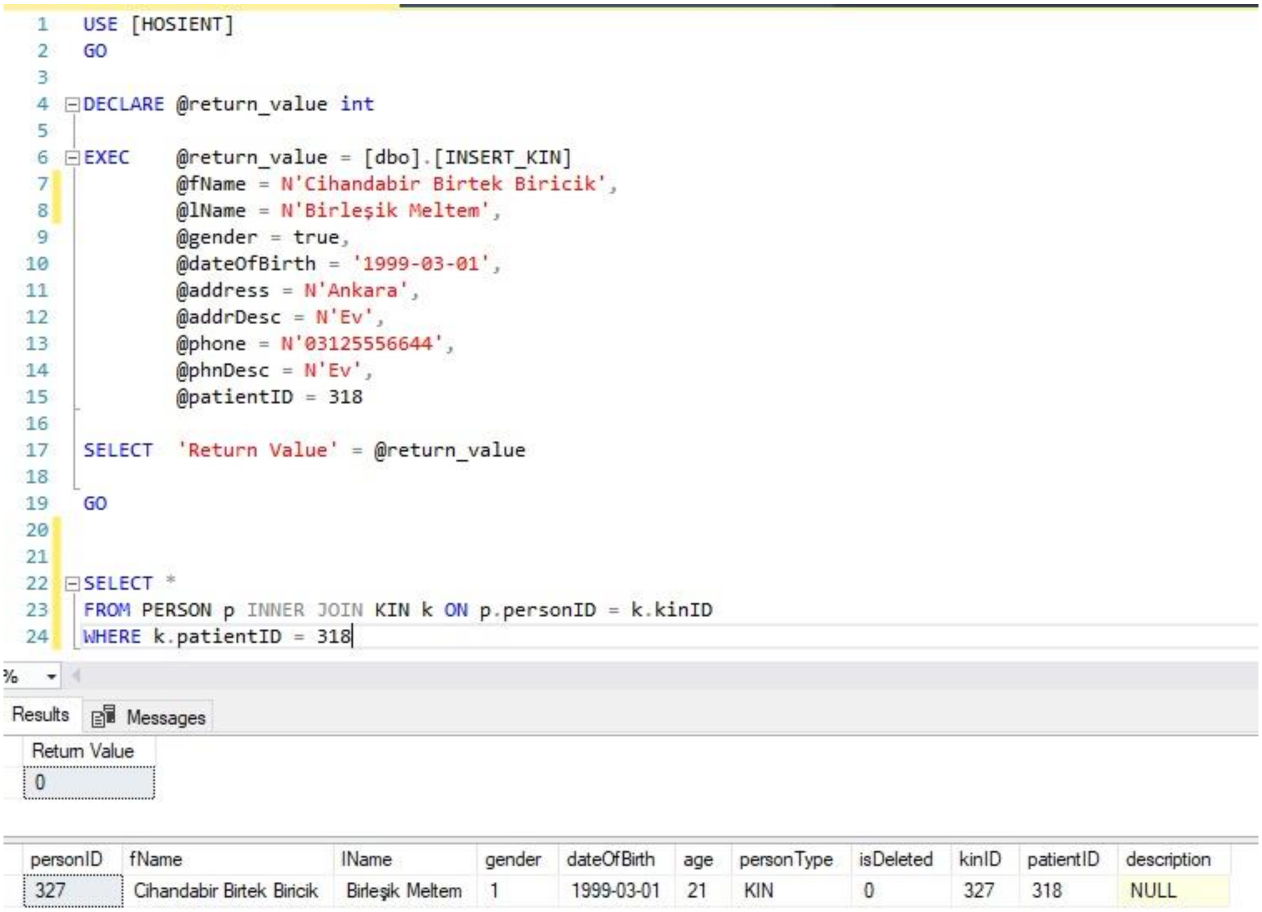


```sql
CREATE PROCEDURE INSERT_KIN     (@fName NVARCHAR(50),
                                 @lName NVARCHAR(50),
                                 @gender BIT,
                                 @dateOfBirth date,
                                 @address NVARCHAR(255),
                                 @addrDesc NVARCHAR(50),
                                 @phone NVARCHAR(50),
                                 @phnDesc NVARCHAR(50),
                                 @patientID BIGINT)
AS
BEGIN
    INSERT INTO PERSON (fName,lName, gender, dateOfBirth, personType)
    SELECT @fName, @lName, @gender, @dateOfBirth, pt.personType
    FROM PERSON_TYPE pt
    WHERE pt.title = 'Kin';

    DECLARE @kinID BIGINT;

    SELECT @kinID = p.personID
    FROM PERSON p
    WHERE p.fName = @fName AND p.lName = @lName AND p.dateOfBirth = @dateOfBirth;

    INSERT INTO KIN(kinID, patientID) VALUES(@kinID, @patientID);

    INSERT INTO PERSON_ADDRESS(personID, address, description) VALUES(@kinID, @address, @addrDesc);

    INSERT INTO PERSON_PHONE(personID, phoneNumber, description) VALUES(@kinID, @phone, @phnDesc);
END
```



```sql

SELECT *
FROM PERSON p INNER JOIN KIN k ON p.personID = k.kinID
WHERE k.patientID = 318
```

personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	kinID	patientID	description

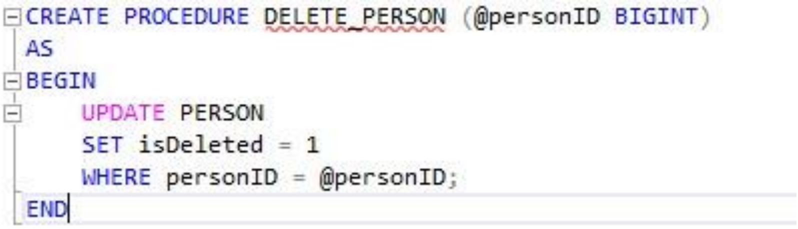


```sql
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[INSERT_KIN]
        @fName = N'Cihandabir Birtek Biricik',
        @lName = N'Birleşik Meltem',
        @gender = true,
        @dateOfBirth = '1999-03-01',
        @address = N'Ankara',
        @addrDesc = N'Ev',
        @phone = N'03125556644',
        @phnDesc = N'Ev',
        @patientID = 318

SELECT  'Return Value' = @return_value

GO


SELECT *
FROM PERSON p INNER JOIN KIN k ON p.personID = k.kinID
WHERE k.patientID = 318
```

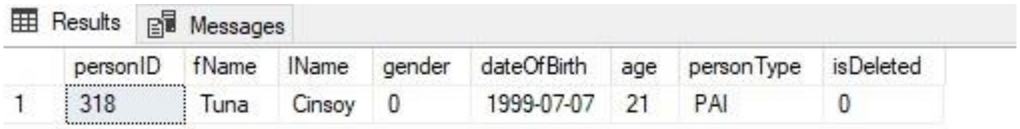
Return Value
0

personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted	kinID	patientID	description
327	Cihandabir Birtek Biricik	Birleşik Meltem	1	1999-03-01	21	KIN	0	327	318	NULL

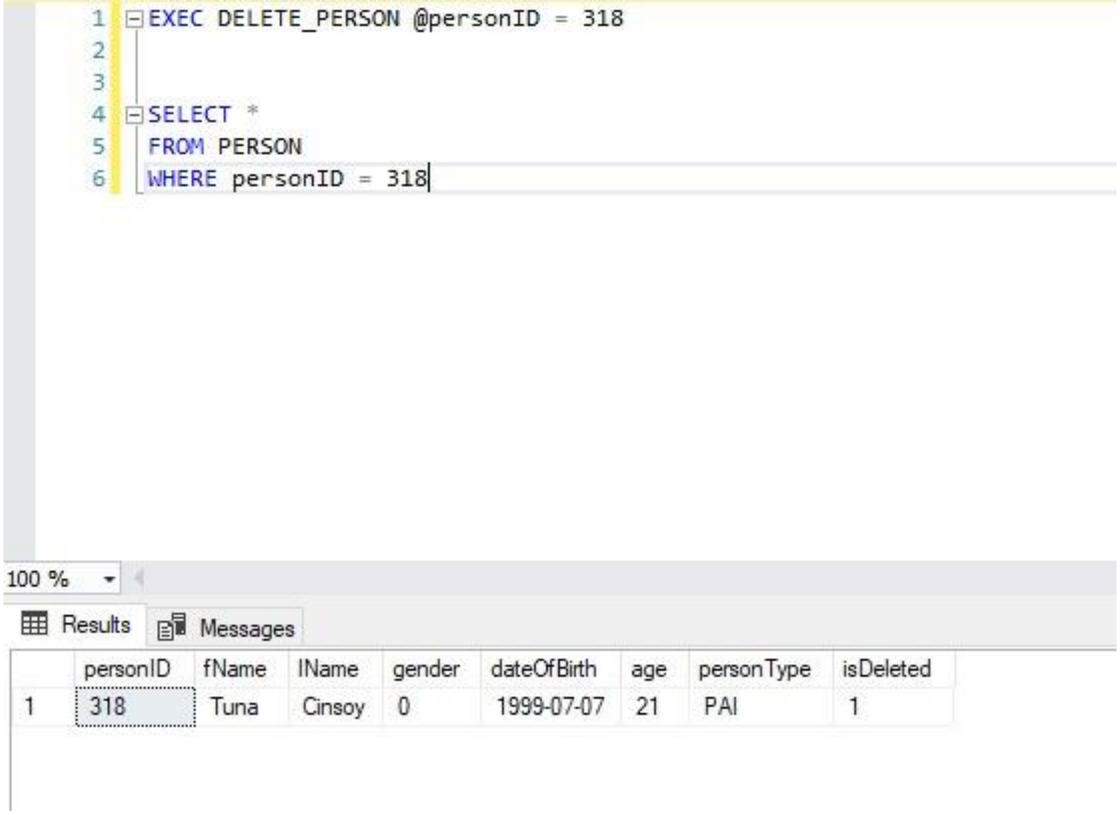
DELETE_PERSON: Procedure for updating "isDeleted" attribute.



```sql
CREATE PROCEDURE DELETE_PERSON (@personID BIGINT)
AS
BEGIN
    UPDATE PERSON
    SET isDeleted = 1
    WHERE personID = @personID;
END
```



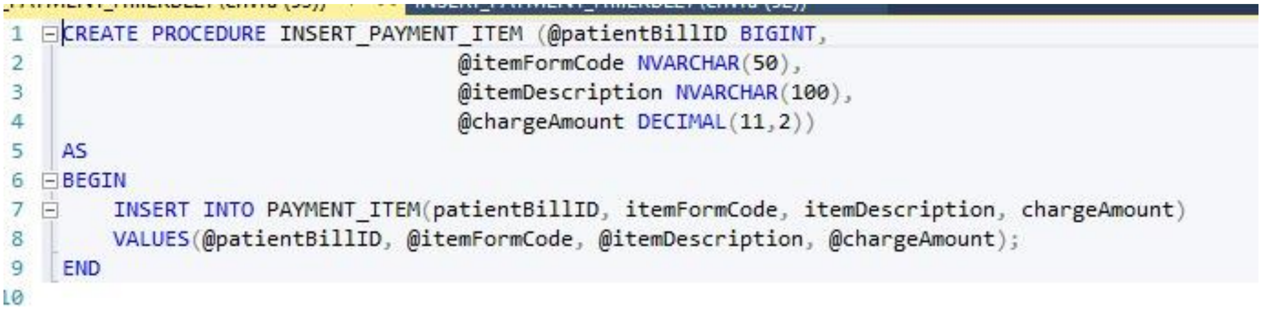
	personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted
1	318	Tuna	Cinsoy	0	1999-07-07	21	PAI	0



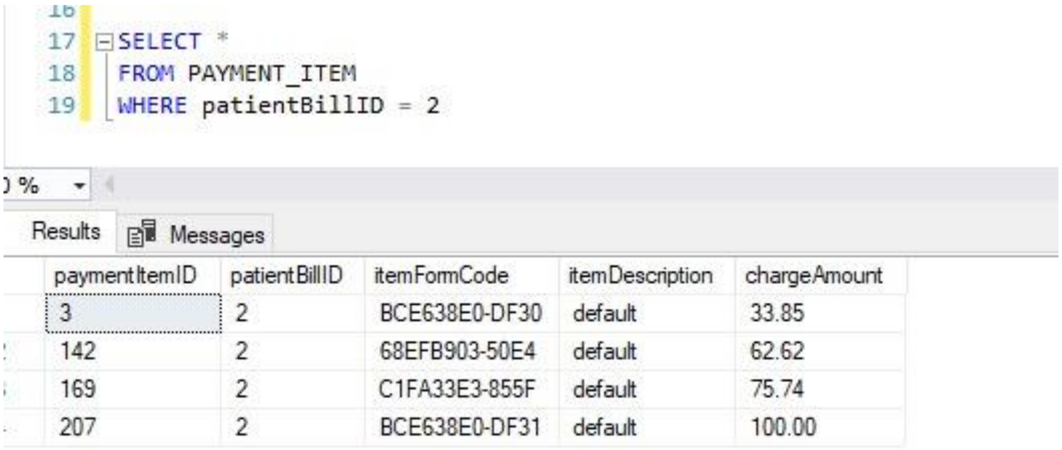
```
EXEC DELETE_PERSON @personID = 318


SELECT *
FROM PERSON
WHERE personID = 318
```

	personID	fName	lName	gender	dateOfBirth	age	personType	isDeleted
1	318	Tuna	Cinsoy	0	1999-07-07	21	PAI	1

INSERT_PAYMENT_ITEM: Procedure for inserting payment item to database.

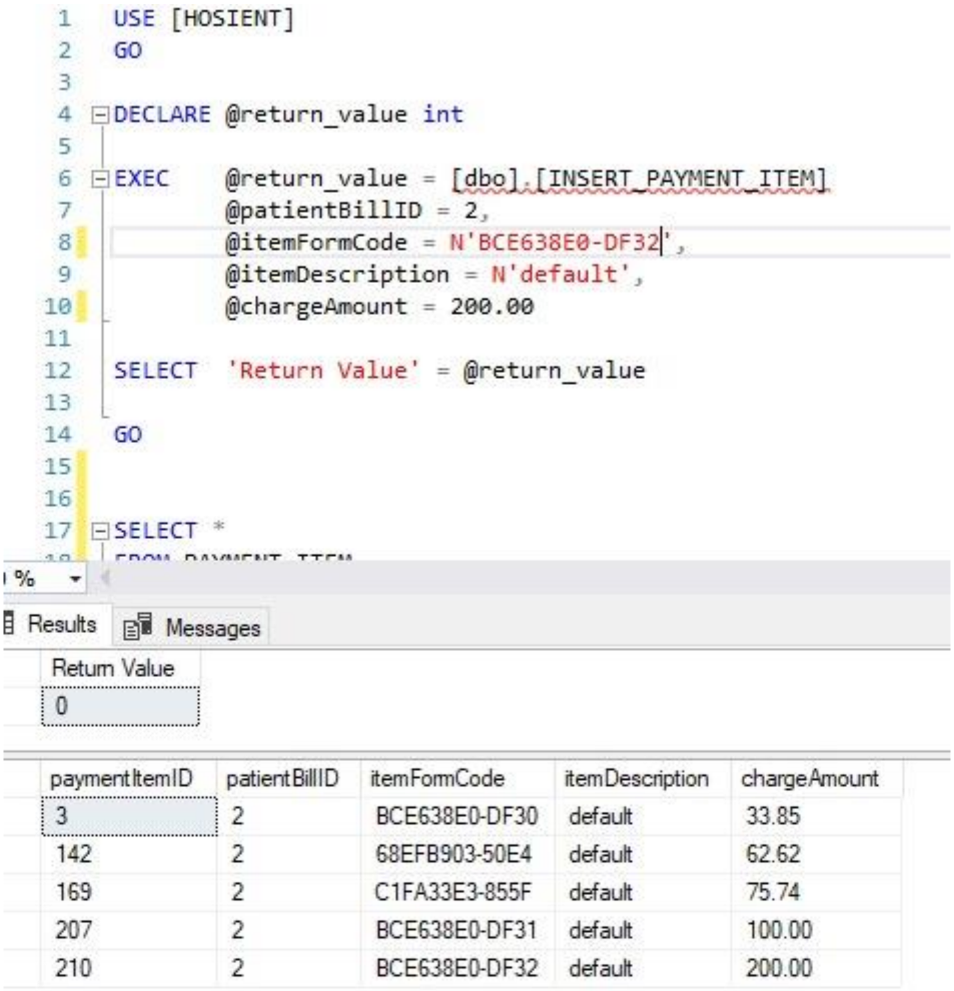


```
CREATE PROCEDURE INSERT_PAYMENT_ITEM (@patientBillID BIGINT,
                                     @itemFormCode NVARCHAR(50),
                                     @itemDescription NVARCHAR(100),
                                     @chargeAmount DECIMAL(11,2))
AS
BEGIN
    INSERT INTO PAYMENT_ITEM(patientBillID, itemFormCode, itemDescription, chargeAmount)
    VALUES(@patientBillID, @itemFormCode, @itemDescription, @chargeAmount);
END
```



```
SELECT *
FROM PAYMENT_ITEM
WHERE patientBillID = 2
```

paymentItemID	patientBillID	itemFormCode	itemDescription	chargeAmount
3	2	BCE638E0-DF30	default	33.85
142	2	68EFB903-50E4	default	62.62
169	2	C1FA33E3-855F	default	75.74
207	2	BCE638E0-DF31	default	100.00

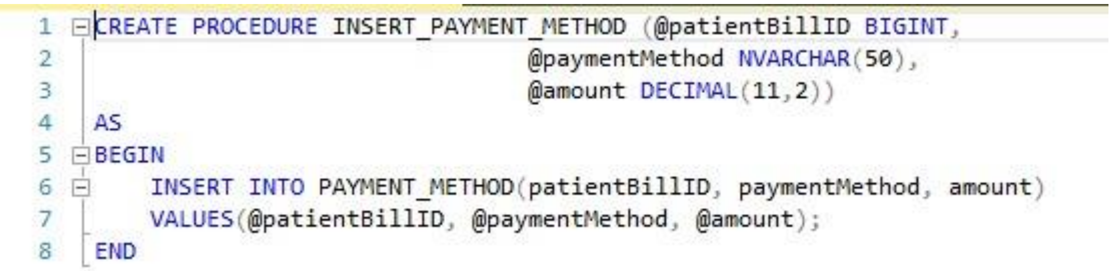


```
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[INSERT_PAYMENT_ITEM]
        @patientBillID = 2,
        @itemFormCode = N'BCE638E0-DF32',
        @itemDescription = N'default',
        @chargeAmount = 200.00

SELECT  'Return Value' = @return_value

GO


SELECT *
FROM PAYMENT_ITEM
```

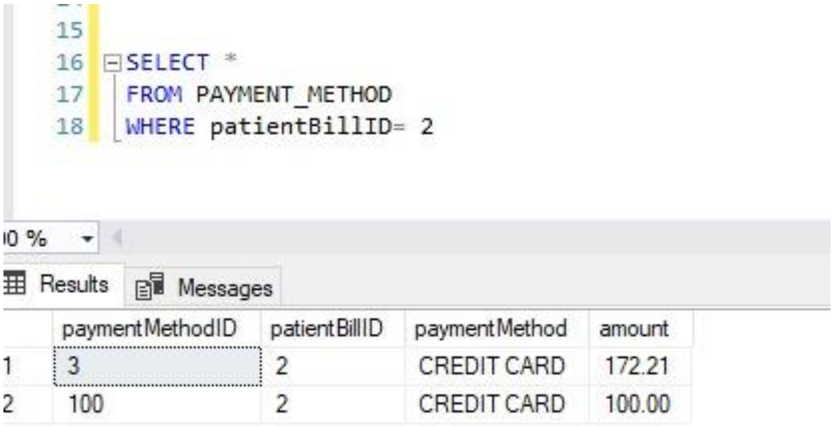
Return Value
0

paymentItemID	patientBillID	itemFormCode	itemDescription	chargeAmount
3	2	BCE638E0-DF30	default	33.85
142	2	68EFB903-50E4	default	62.62
169	2	C1FA33E3-855F	default	75.74
207	2	BCE638E0-DF31	default	100.00
210	2	BCE638E0-DF32	default	200.00

INSERT_PAYMENT_METHOD: Procedure for inserting a payment method record to database.

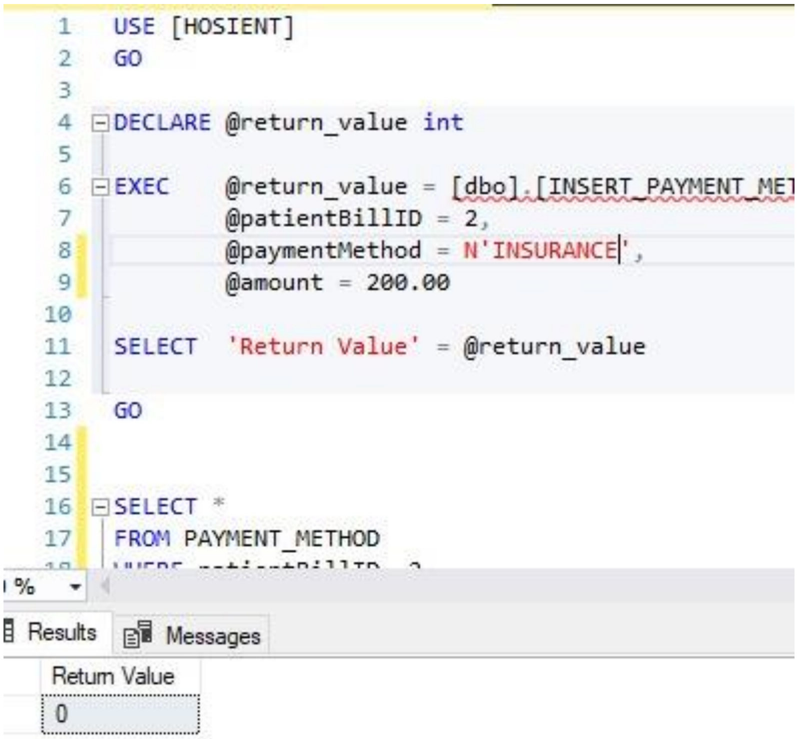


```
CREATE PROCEDURE INSERT_PAYMENT_METHOD (@patientBillID BIGINT,
                                        @paymentMethod NVARCHAR(50),
                                        @amount DECIMAL(11,2))
AS
BEGIN
    INSERT INTO PAYMENT_METHOD(patientBillID, paymentMethod, amount)
    VALUES(@patientBillID, @paymentMethod, @amount);
END
```



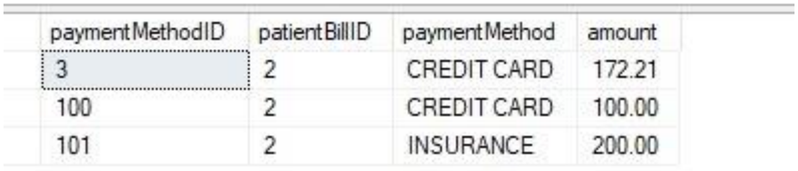
```
SELECT *
FROM PAYMENT_METHOD
WHERE patientBillID= 2
```

	paymentMethodID	patientBillID	paymentMethod	amount
1	3	2	CREDIT CARD	172.21
2	100	2	CREDIT CARD	100.00



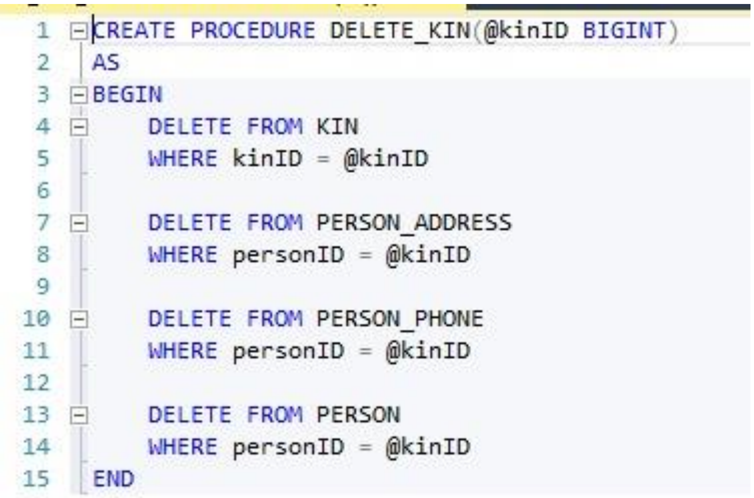
```sql
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[INSERT_PAYMENT_MET
        @patientBillID = 2,
        @paymentMethod = N'INSURANCE',
        @amount = 200.00

SELECT  'Return Value' = @return_value

GO


SELECT *
FROM PAYMENT_METHOD
WHERE patientBillID = 2
```

Return Value
0

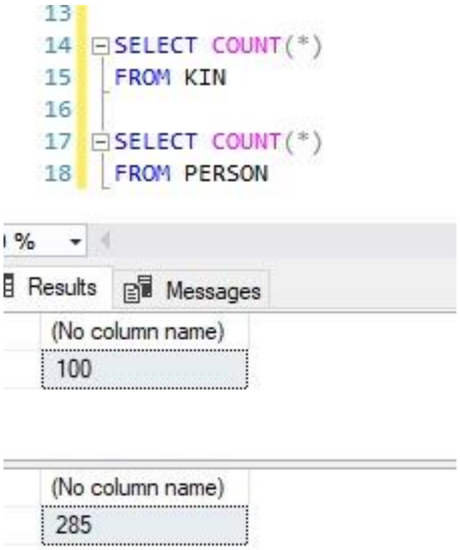


paymentMethodID	patientBillID	paymentMethod	amount
3	2	CREDIT CARD	172.21
100	2	CREDIT CARD	100.00
101	2	INSURANCE	200.00

DELETE_KIN: Procedure for deleting kin from database.



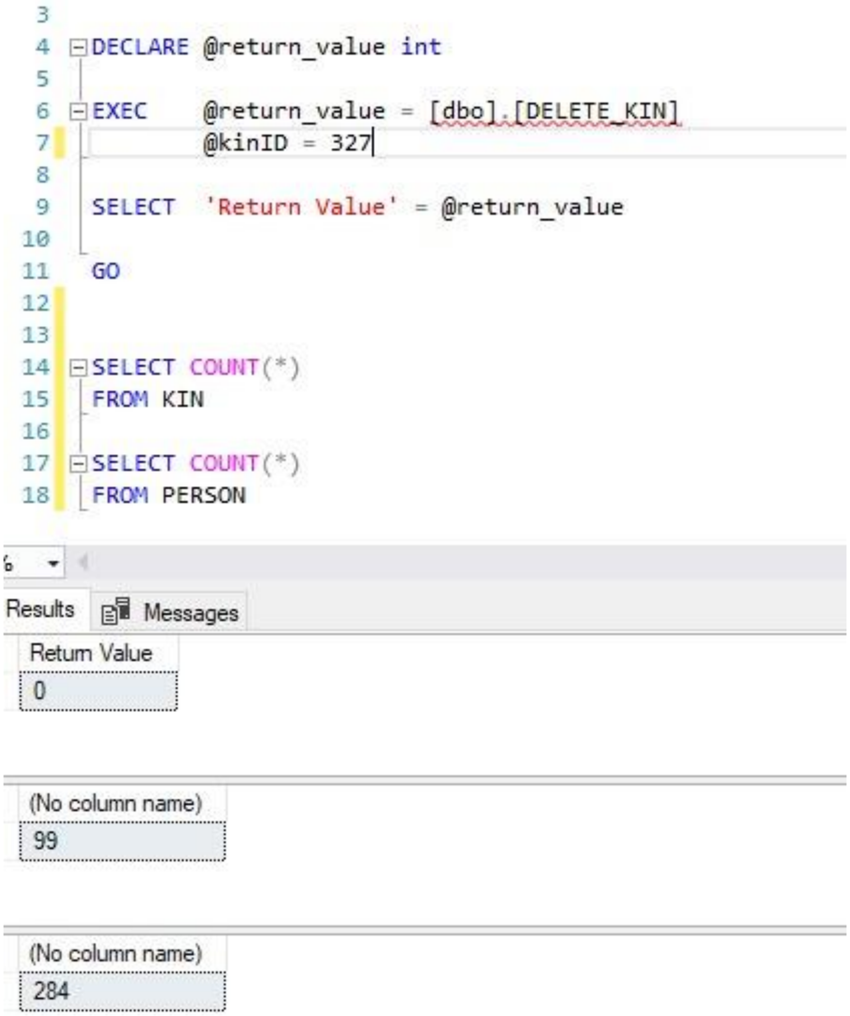
```sql
CREATE PROCEDURE DELETE_KIN(@kinID BIGINT)
AS
BEGIN
    DELETE FROM KIN
    WHERE kinID = @kinID

    DELETE FROM PERSON_ADDRESS
    WHERE personID = @kinID

    DELETE FROM PERSON_PHONE
    WHERE personID = @kinID

    DELETE FROM PERSON
    WHERE personID = @kinID
END
```



```
SELECT COUNT(*)
FROM KIN

SELECT COUNT(*)
FROM PERSON
```

```
(No column name)
100
```

```
(No column name)
285
```



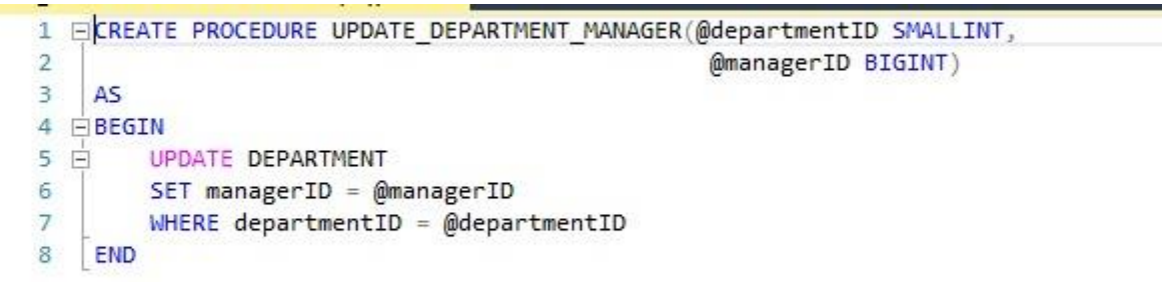
```

DECLARE @return_value int

EXEC    @return_value = [dbo].[DELETE_KIN]
        @kinID = 327

SELECT  'Return Value' = @return_value

GO


SELECT COUNT(*)
FROM KIN

SELECT COUNT(*)
FROM PERSON
```

```
Return Value
0
```

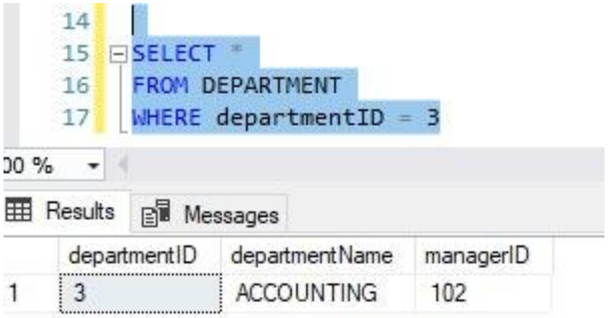
```
(No column name)
99
```

```
(No column name)
284
```

UPDATE_DEPARTMENT_MANAGER: Procedure for updating manager of any department.

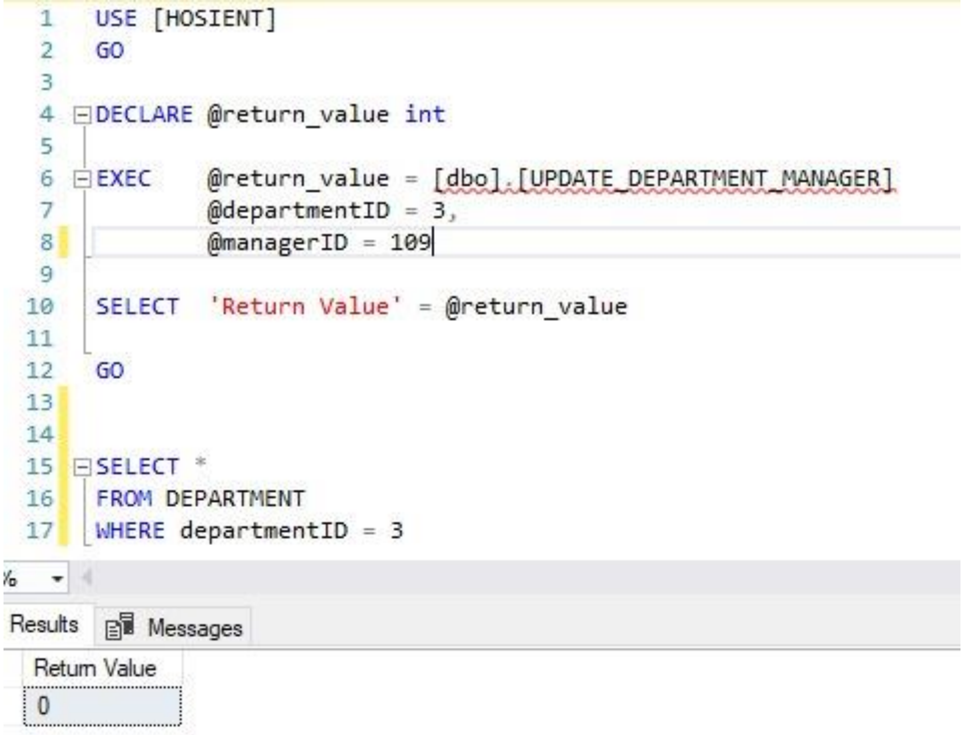


```
CREATE PROCEDURE UPDATE_DEPARTMENT_MANAGER(@departmentID SMALLINT,
                                          @managerID BIGINT)
AS
BEGIN
    UPDATE DEPARTMENT
    SET managerID = @managerID
    WHERE departmentID = @departmentID
END
```



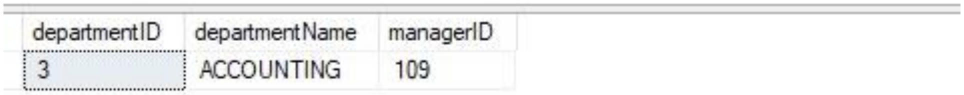
```

SELECT *
FROM DEPARTMENT
WHERE departmentID = 3
```

	departmentID	departmentName	managerID
1	3	ACCOUNTING	102



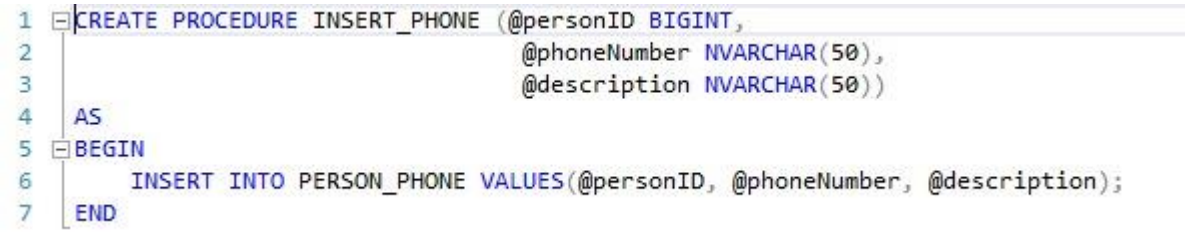
```
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[UPDATE_DEPARTMENT_MANAGER]
        @departmentID = 3,
        @managerID = 109

SELECT  'Return Value' = @return_value

GO


SELECT *
FROM DEPARTMENT
WHERE departmentID = 3
```

Return Value
0

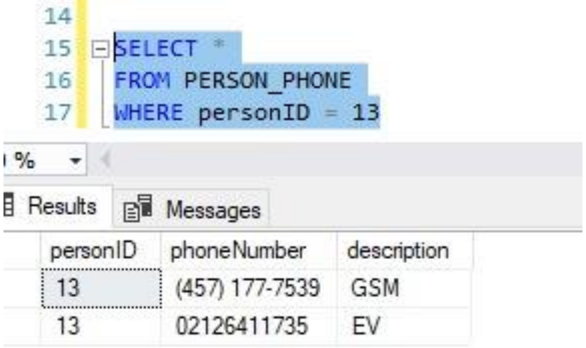


departmentID	departmentName	managerID
3	ACCOUNTING	109

INSERT_PHONE: Procedure for inserting information about phoneNumber of people.

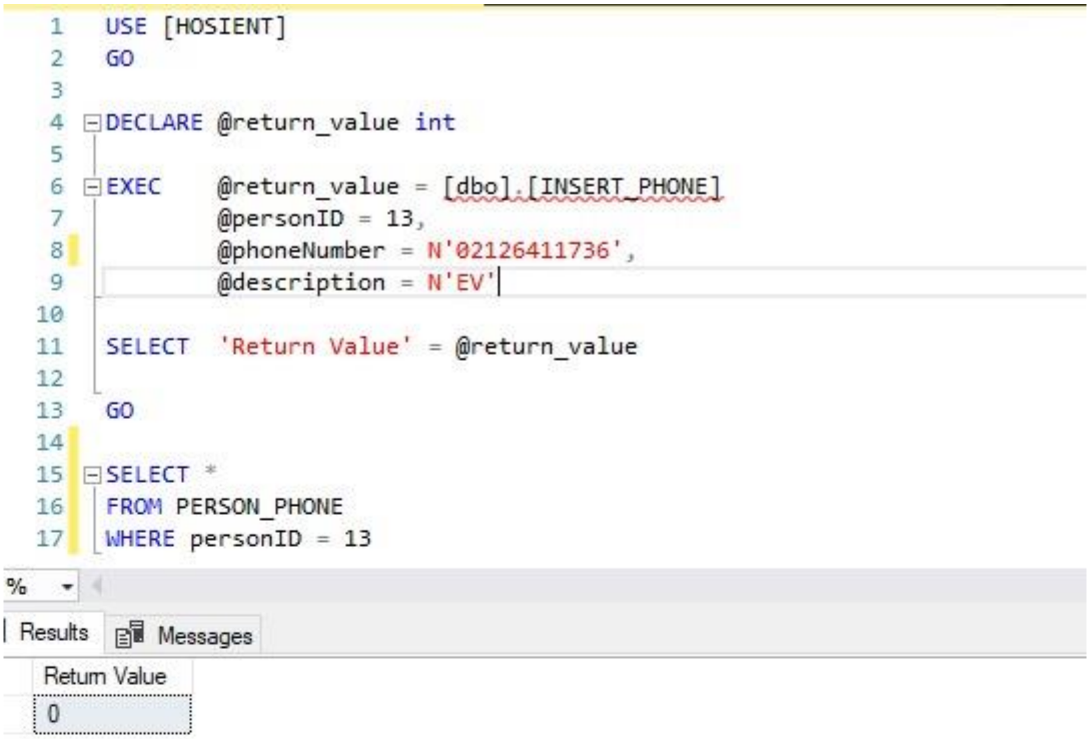


```
CREATE PROCEDURE INSERT_PHONE (@personID BIGINT,
                               @phoneNumber NVARCHAR(50),
                               @description NVARCHAR(50))
AS
BEGIN
    INSERT INTO PERSON_PHONE VALUES(@personID, @phoneNumber, @description);
END
```



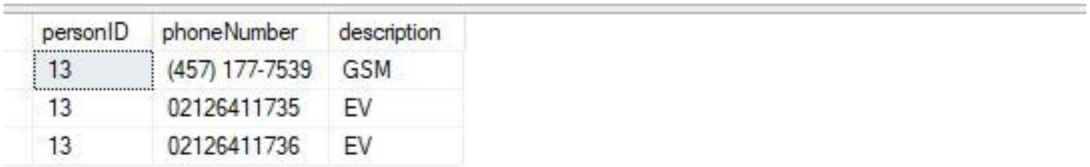
```
SELECT *
FROM PERSON_PHONE
WHERE personID = 13
```

personID	phoneNumber	description
13	(457) 177-7539	GSM
13	02126411735	EV



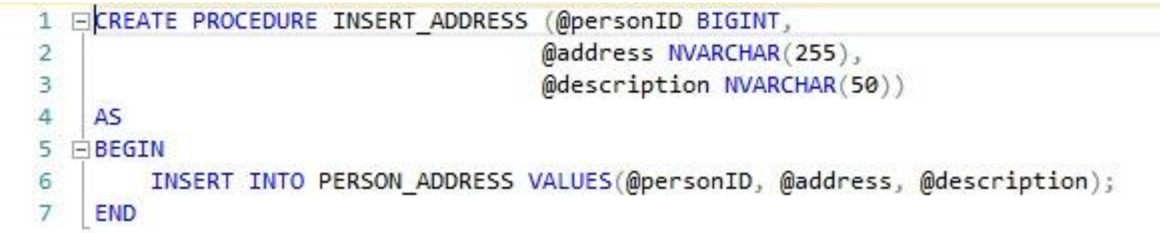
```
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[INSERT_PHONE]
        @personID = 13,
        @phoneNumber = N'02126411736',
        @description = N'EV'

SELECT  'Return Value' = @return_value

GO

SELECT *
FROM PERSON_PHONE
WHERE personID = 13
```

Return Value
0

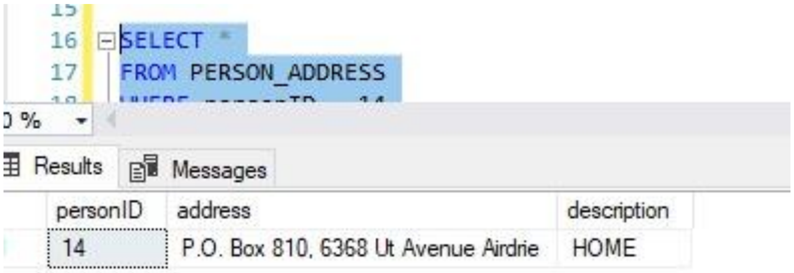


personID	phoneNumber	description
13	(457) 177-7539	GSM
13	02126411735	EV
13	02126411736	EV

INSERT_ADDRESS: Procedure for inserting address to database.

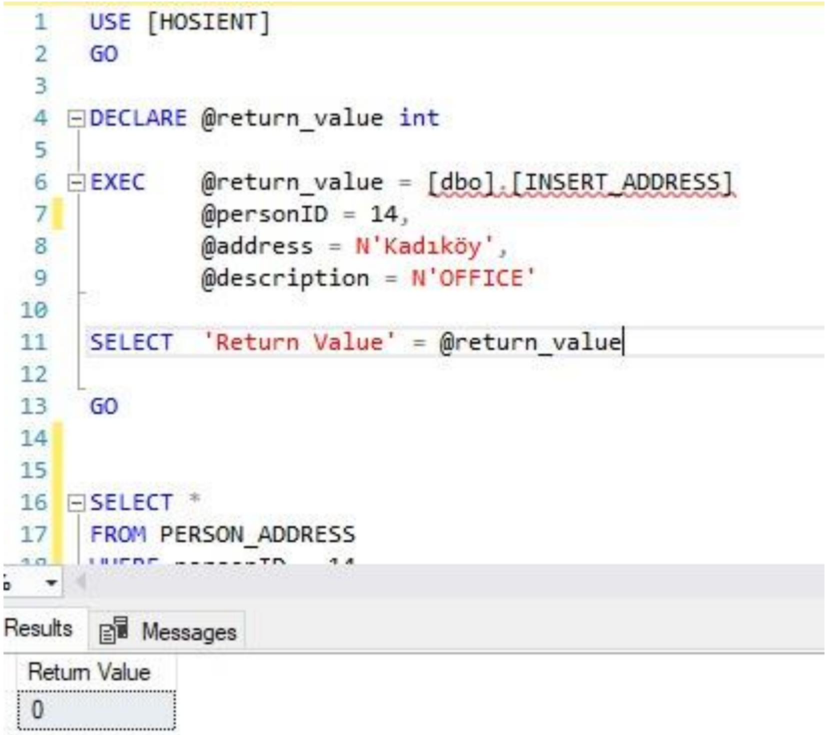


```
CREATE PROCEDURE INSERT_ADDRESS (@personID BIGINT,
                                 @address NVARCHAR(255),
                                 @description NVARCHAR(50))
AS
BEGIN
    INSERT INTO PERSON_ADDRESS VALUES(@personID, @address, @description);
END
```



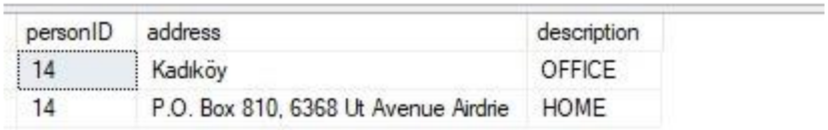
```
SELECT *
FROM PERSON_ADDRESS
```

personID	address	description
14	P.O. Box 810, 6368 Ut Avenue Airdrie	HOME



```
USE [HOSIENT]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[INSERT_ADDRESS]
        @personID = 14,
        @address = N'Kadıköy',
        @description = N'OFFICE'

SELECT  'Return Value' = @return_value

GO


SELECT *
FROM PERSON_ADDRESS
```

Return Value
0



personID	address	description
14	Kadıköy	OFFICE
14	P.O. Box 810, 6368 Ut Avenue Airdrie	HOME