به نام خدا

پروژه پایگاه داده

سیستم مدیریت خدمات پزشکی

نام استاد:

زهرا علی نیا

اعضای گروه:

فرناز فتاحی ۱۴۰۰۰۱۲۲۶۸۱۱۱

امیرحسین مرتضایی ۱۴۰۰۰۱۲۲۶۸۰۱۳

پاییز ۱۴۰۳

فاز اول – نیاز سنجی

برای انجام مرحله نیازسنجی، ابتدا باید سناریوی پروژه را تحلیل کنیم و نیازمندی‌های اصلی را استخراج کنیم. بر اساس توضیحات موجود در صورت پروژه، سیستم قرار است برای مدیریت خدمات بیمارستان و مراکز درمانی طراحی شود. در ادامه نیازمندی‌های این پروژه را فهرست می‌کنیم:

هدف پروژه:

- طراحی یک سیستم پایگاه داده برای مدیریت اطلاعات و خدمات بیمارستانی
- افزایش کارایی در ذخیره، مدیریت، و پیگیری اطلاعات بیماران و خدمات پزشکی

ویژگی های سامانه خدمات درمانی:

- مدیریت بیماران: مدیریت اطلاعات بیماران شامل پیشینه پزشکی، اطلاعات دموگرافیک و اطلاعات مرتبط با بیمه (نام، کد ملی، سن، جنسیت، آدرس، شماره تماس) (تاریخچه درمان، داروها، آزمایش‌ها)
- مدیریت ویزیت: زمان بندی و مدیریت ویزیت ها (رزرو و تغییر زمان ملاقات بیماران با پزشکان)
- مدیریت مالی: مدیریت صورت حساب ها و خدمات بیمه برای بیماران
- مدیریت دکتران: مدیریت اطلاعات دکتران شامل تخصص، برنامه ریزی و اطلاعات تماس

کاربران سیستم:

- مدیر بیمارستان: برای نظارت کلی بر خدمات و عملکرد سیستم
- پزشکان: برای مشاهده و به‌روزرسانی سوابق بیماران
- بیماران: برای رزرو وقت ملاقات و مشاهده سوابق پزشکی
- کارکنان پذیرش: برای ثبت اطلاعات و مدیریت وقت ملاقات‌ها

نیازمندی‌های عملکردی:

- ثبت اطلاعات بیماران (نام، کد ملی، سن، جنسیت، آدرس، شماره تماس)

- ثبت و مدیریت سوابق پزشکی بیماران (بیماری‌ها، آزمایش‌ها، درمان‌ها)
- مدیریت وقت ملاقات‌ها (ثبت وقت جدید، تغییر وقت، حذف وقت)
- تولید صورت‌حساب و مدیریت پرداخت‌ها
- ارائه گزارش‌های مدیریتی (مثل تعداد بیماران، وضعیت پرداخت‌ها)

نیازمندی‌های غیرعملکردی:

- امنیت بالا برای حفاظت از اطلاعات حساس بیماران
- قابلیت دسترسی آسان برای کاربران مختلف با رابط کاربری ساده
- قابلیت مقیاس‌پذیری برای اضافه کردن کاربران و داده‌ها در آینده
- سرعت مناسب در پاسخ‌گویی به درخواست‌ها

محدودیت‌ها:

- اطلاعات باید فقط برای کاربران مجاز (پزشکان، مدیران، کارکنان پذیرش) قابل دسترسی باشد
- زمان‌بندی ملاقات‌ها باید با زمان‌بندی پزشکان مطابقت داشته باشد

ویژگی های سامانه خدمات درمانی:

۱. مدیریت بیماران: مدیریت اطلاعات بیماران شامل پیشینه پزشکی، اطلاعات دموگرافیک و اطلاعات مرتبط با بیمه
۲. مدیریت ویزیت: زمان بندی و مدیریت ویزیت ها (رزرو و تغییر زمان ملاقات بیماران با پزشکان)
۳. مدیریت مالی: مدیریت صورت حساب ها و خدمات بیمه برای بیماران
۴. مدیریت دکتران: مدیریت اطلاعات دکتران شامل تخصص، برنامه ریزی و اطلاعات تماس

فاز ۲ - طراحی معنایی

پایگاه داده این سامانه شامل موجودیت هایی نظیر بیمار، ویزیت، صورت حساب و دکترهاست. هر کدام از این موجودیت ها اطلاعاتی را در مورد بیماران، ویزیت ها، صورت حساب ها و اطلاعات دکترها ذخیره می کنند.

موجودیت ها و صفت ها

۱. *موجودیت بیمار*

این موجودیت اطلاعات بیمار شامل نام، کد ملی، سن، جنسیت، آدرس، شماره تماس، اطلاعات بیمه و همچنین تاریخچه درمان، داروها، آزمایش‌ها را نگه داری می کند.

صفت ها:

- شناسه بیمار (کلید اصلی که برای هر بیمار منحصر به فرد است)
- نام (شامل نام و نام خانوادگی)
- تاریخ تولد
- جنسیت
- آدرس
- شماره تماس

۲. *موجودیت دکتر*

این موجودیت اطلاعات مربوط به دکترها شامل تخصص آن ها و زمانبندی کاری آن ها را مدیریت می کند.

صفت ها:

- شناسه دکتر (کلید اصلی)
- نام (شامل نام و نام خانوادگی)
- آدرس
- شماره تماس
- تخصص
- برنامه کاری (روزها و ساعت هایی که حضور دارد)

۳. *موجودیت بیمه*

صفت ها:

- شناسه بیمه (کلید اصلی)
- نام شرکت بیمه
- اطلاعات تماس
- جزئیات پوشش

۴. *موجودیت ویزیت*

این موجودیت ویزیت های بیماران با دکترها را مدیریت می کند و بیماران را به دکتر موردنظرشان لینک می کند و وضعیت ویزیت ها را دنبال می کند.

صفت ها:

- شناسه ویزیت (کلید اصلی)
- وضعیت (رزرو شده، کنسل شده و ...)

۵. *موجودیت صورت حساب*

این موجودیت صورت حساب ها و خدمات بیمه را مدیریت می کند.

صفت ها:

- شماره صورت حساب (کلید اصلی)
- مبلغ صورتحساب
- تاریخ صورتحساب
- وضعیت پرداخت (پرداخت شده، پرداخت نشده)

۶. *موجودیت نسخه*

صفت ها:

- شماره نسخه
- تاریخ
- دوز (متن)

۷. *موجودیت سابقه پذیرش*

صفت ها:

- شماره پذیرش (به همراه شناسه بیمار کلید اصلی)
- تاریخ پذیرش
- تاریخ مرخصی

۸. *موجودیت دارو*

صفت ها:

- کد دارو (کلید اصلی)
- نام دارو
- برند دارو
- توصیف دارو

۹. *موجودیت بخش*

صفت ها:

- شناسه بخش (کلید اصلی)
- نام بخش
- مسئول بخش

۱۰. *موجودیت تخت*

صفت ها:

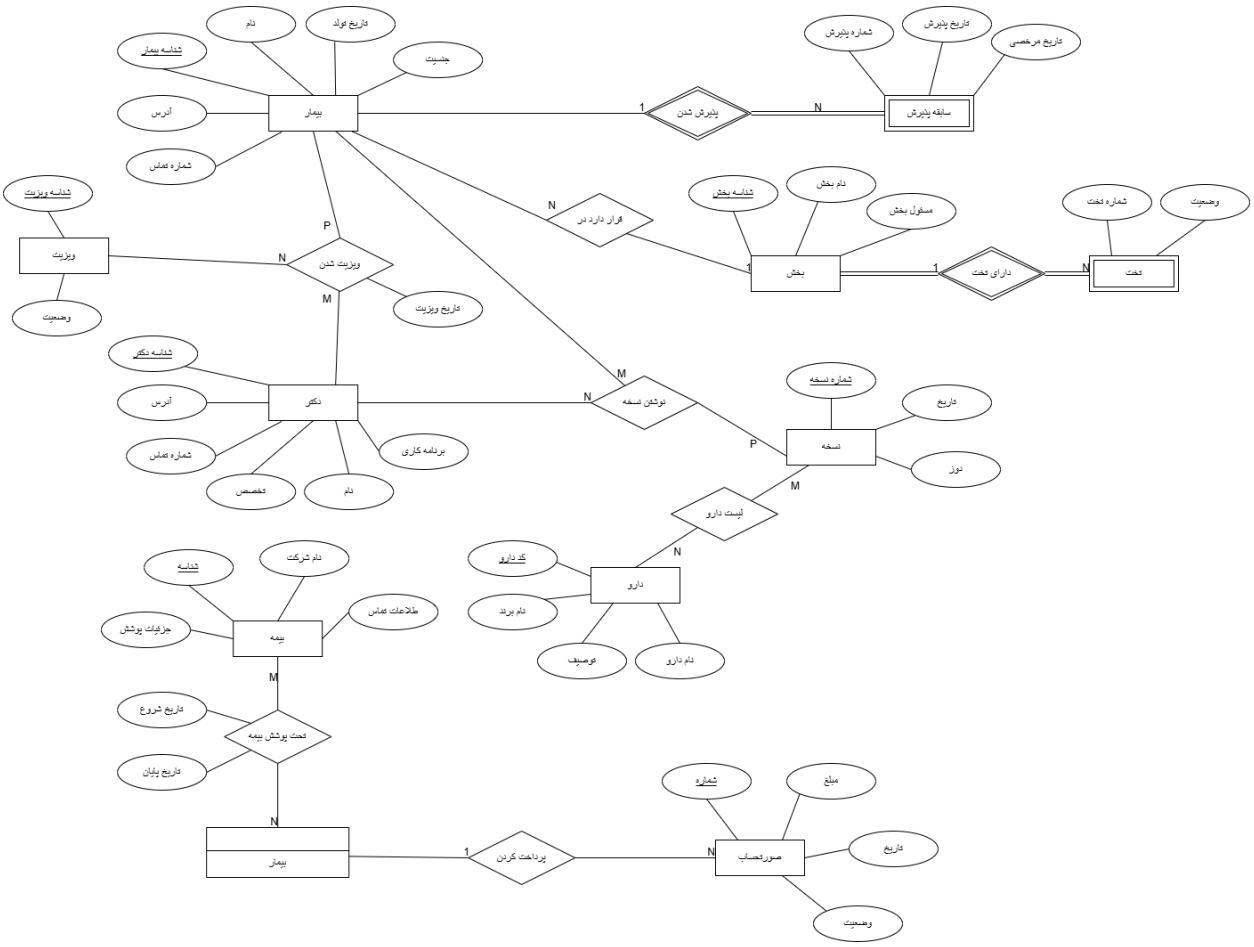
- شماره تخت (به همراه شماره بخش کلید اصلی)
- وضعیت (اشغال یا خالی)

رابطه بین موجودیت ها

۱. *ویزیت شدن:* رابطه چند به چند بین بیمار، دکتر و ویزیت
۲. *تحت پوشش بیمه:* رابطه چند به چند بین بیمه و بیمار

۳. *پرداخت کردن:* رابطه یک به چند بین بیمار و صورتحساب
۴. *قرار دارد در:* رابطه یک به چند بین بیمار و بخش
۵. *دارای تخت:* رابطه یک به چند موجودیت ضعیف تخت با بخش
۶. *پذیرش شدن:* رابطه یک به چند بیمار با سابقه پذیرش
۷. *نوشتن نسخه:* رابطه چند به چند بین بیمار و دکتر
۸. *لیست دارو:* رابطه چند به چند بین نسخه و دارو

نمودار ERD



فاز سوم – طراحی منطقی

فاز طراحی منطقی یکی از مراحل کلیدی در فرآیند طراحی پایگاه داده است که در آن مدل معنایی داده‌ها) مثل (ERDبه ساختار منطقی قابل‌اجرا در یک سیستم مدیریت پایگاه داده (DBMS) تبدیل می‌شود. در این فاز، ارتباط بین موجودیت‌ها به جداول و ستون‌های واقعی در پایگاه داده تبدیل می‌شود و نوع داده‌ها و محدودیت‌ها برای هر ستون تعریف می‌گردد.

جداول منطقی

Patient (PateintID, Pname, PBD, PGender, PAddress, PPhone, SectionNum)

Doctor (DoctorID, DName, DAddress, DPhone, DSpecialty, DSchedule)

Appointment (AppointmentID, AStatus)

ReceptionHistory (ReceptionNum, PatientID, RDate, RDischarge)

Section (SectionNum, SName, SHead)

Bed (BedNum, SectionNum, BStatus)

Prescription (PrescriptionNum, PDate, PDose)

Drug (DrugCode, DrugName, DBrand, DPrescription)

Bill (BillNum, BAmount, BDate, BStatus, PatientID)

Insurance (InsuranceID, ICompany, IDetails, IContact)

MakingAppointment (PatientID, DoctorID, AppointmentID, Date)

InsuranceCoverage (InsuranceID, PatientID, StartDate, EndDate)

WritingPrescription (PatientID, DoctorID, PrescriptionNum)

DrugsList (DrugCode, PrescriptionNum)

تعریف نوع داده‌ها و محدودیت ها

نوع داده برای هر ستون بر اساس ماهیت داده (مثل عددی، متنی، تاریخ، و غیره) و همچنین محدودیت از نظر هیچ مقدارناپذیری، یکتا بودن و ... نیز تعیین می شود.

Patient:

ستون	نوع داده	محدودیت ها
PatientID	INT	NOT NULL, PRIMARY KEY
PName	VARCHAR(100)	NOT NULL
PBD	DATE	NOT NULL
PGender	CHAR(1)	NOT NULL, CHECK (PGender IN ('M', 'F'))
PAddress	VARCHAR(255)	-
PPhone	VARCHAR(15)	NOT NULL
SectionNum	INT	NOT NULL, FOREIGN KEY

Doctor:

ستون	نوع داده	محدودیت ها
DoctorID	INT	NOT NULL, PRIMARY KEY
DName	VARCHAR(100)	NOT NULL
DAddress	VARCHAR(255)	-
DPhone	VARCHAR(15)	NOT NULL
DSpecialty	VARCHAR(50)	NOT NULL
DSchedule	VARCHAR(255)	-

Appointment:

ستون	نوع داده	محدودیت ها
AppointmentID	INT	NOT NULL, PRIMARY KEY
AStatus	VARCHAR(50)	NOT NULL, CHECK (AStatus IN ('Scheduled', 'Canceled', 'Completed'))

ReceptionHistory:

ستون	نوع داده	محدودیت ها
ReceptionNum	INT	NOT NULL, PRIMARY KEY
PatientID	INT	NOT NULL, FOREIGN KEY
RDate	DATE	NOT NULL
RDischarge	DATE	-

Section:

ستون	نوع داده	محدودیت ها
SectionNum	INT	NOT NULL, PRIMARY KEY
SName	VARCHAR(50)	NOT NULL, UNIQUE
SHead	VARCHAR(100)	NOT NULL

Bed:

ستون	نوع داده	محدودیت ها
BedNum	INT	NOT NULL, PRIMARY KEY
SectionNum	INT	NOT NULL, FOREIGN KEY
BStatus	VARCHAR(50)	NOT NULL, CHECK (BStatus IN ('Available', 'Occupied'))

Prescription:

ستون	نوع داده	محدودیت ها
PrescriptionNum	INT	NOT NULL, PRIMARY KEY
PDate	DATE	NOT NULL
PDose	VARCHAR(255)	NOT NULL

Drug:

ستون	نوع داده	محدودیت ها
DrugCode	INT	NOT NULL, PRIMARY KEY
DrugName	VARCHAR(100)	NOT NULL
DBrand	VARCHAR(50)	NOT NULL
DPrescription	VARCHAR(255)	-

Bill:

ستون	نوع داده	محدودیت ها
BillNum	INT	NOT NULL, PRIMARY KEY
BAmount	DECIMAL(10,2)	NOT NULL, CHECK (BAmount >= 0)
BDate	DATE	NOT NULL
BStatus	VARCHAR(50)	NOT NULL, CHECK (BStatus IN ('Paid', 'Unpaid', 'Pending'))
PatientID	INT	NOT NULL, FOREIGN KEY

Insurance:

ستون	نوع داده	محدودیت ها
InsuranceID	INT	NOT NULL, PRIMARY KEY
ICompany	VARCHAR(100)	NOT NULL
IDetails	VARCHAR(255)	-
IContact	VARCHAR(50)	-

MakingAppointment:

ستون	نوع داده	محدودیت ها
PatientID	INT	NOT NULL, FOREIGN KEY
DoctorID	INT	NOT NULL, FOREIGN KEY
AppointmentID	INT	NOT NULL, FOREIGN KEY
Date	DATE	NOT NULL

InsuranceCoverage:

ستون	نوع داده	محدودیت ها
InsuranceID	INT	NOT NULL, FOREIGN KEY
PatientID	INT	NOT NULL, FOREIGN KEY
StartDate	DATE	NOT NULL
EndDate	DATE	NOT NULL

WritingPrescription:

ستون	نوع داده	محدودیت ها
PatientID	INT	NOT NULL, FOREIGN KEY
DoctorID	INT	NOT NULL, FOREIGN KEY
PrescriptionNum	INT	NOT NULL, FOREIGN KEY

DrugsList:

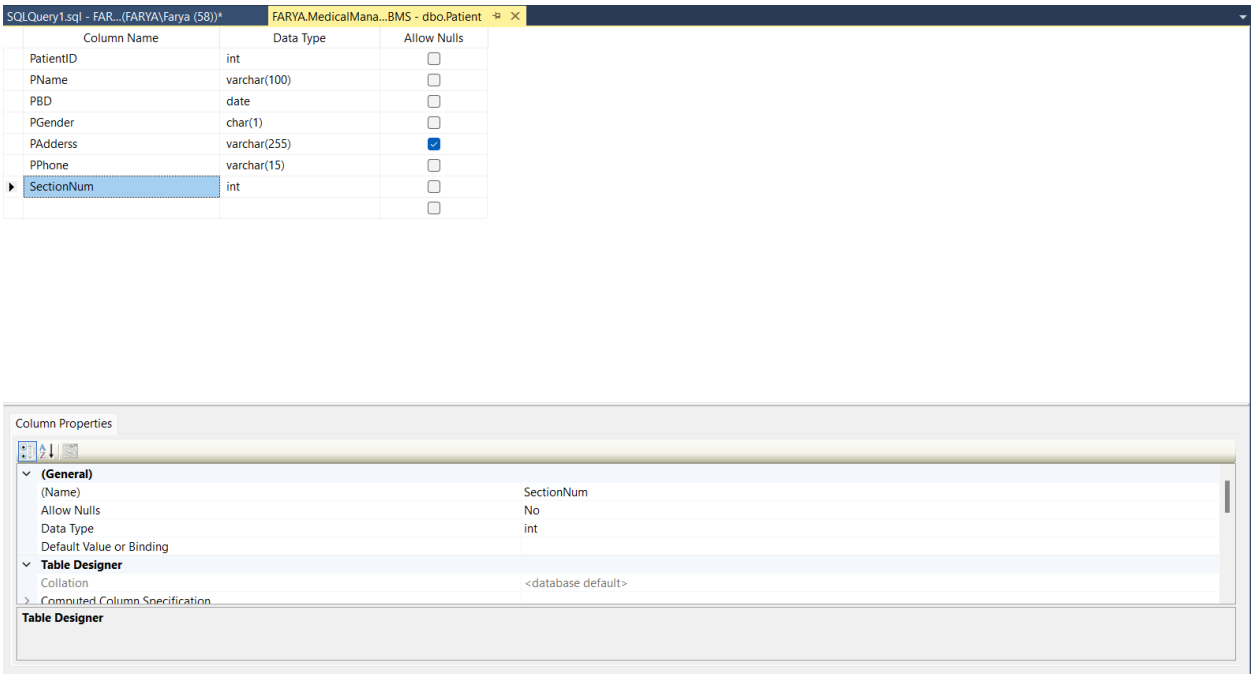
ستون	نوع داده	محدودیت ها
DrugCode	INT	NOT NULL, FOREIGN KEY
PrescriptionNum	INT	NOT NULL, FOREIGN KEY

پیاده سازی در SQL:

برای پیاده سازی در SQL از SQL Server Management Studio استفاده شد. با کلیک بر روی database یک پایگاه داده جدید ایجاد می کنیم. و با کلیک بر روی tables جدول های موردنظرمان را می سازیم.

ایجاد جدول ها:

هم به صورت دستی و هم به صورت کوئری می توان جدول ها را ساخت. ایجاد جدول به صورت دستی به صورت زیر است:



برای ایجاد جدول ها به صورت کوئری باید ابتدا یک کوئری در همین کانکشن ایجاد کنیم و سپس دستورات را نوشته و execute را بزنیم. دستور ایجاد جدول های گفته شده به صورت زیر است و در فایل CreateTables در فولدر پروژه قرار دارد:

```
-- Create the Section table
CREATE TABLE Section (
    SectionNum INT PRIMARY KEY,
    SName VARCHAR(50) NOT NULL UNIQUE,
    SHead VARCHAR(100)
);

-- Create the Patient table
CREATE TABLE Patient (
    PatientID INT PRIMARY KEY,
    PName VARCHAR(100) NOT NULL,
    PBD DATE NOT NULL,
    PGender CHAR(1) NOT NULL CHECK (PGender IN ('M', 'F')),
    PAddress VARCHAR(255),
    PPhone VARCHAR(15) NOT NULL,
    SectionNum INT NOT NULL,
    FOREIGN KEY (SectionNum) REFERENCES Section(SectionNum)
);

-- Create the Doctor table
CREATE TABLE Doctor (
    DoctorID INT PRIMARY KEY,
    DName VARCHAR(100) NOT NULL,
    DAddress VARCHAR(255),
    DPhone VARCHAR(15) NOT NULL UNIQUE,
```

```
    DSpecialty VARCHAR(50) NOT NULL,
    DSchedule VARCHAR(255)
);

-- Create the Appointment table
CREATE TABLE Appointment (
    AppointmentID INT PRIMARY KEY,
    AStatus VARCHAR(50) NOT NULL CHECK (AStatus IN ('Scheduled', 'Canceled',
'Completed'))
);

-- Create the ReceptionHistory table
CREATE TABLE ReceptionHistory (
    ReceptionNum INT PRIMARY KEY,
    PatientID INT NOT NULL,
    RDate DATE NOT NULL,
    RDischarge DATE,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)
);

-- Create the Bed table
CREATE TABLE Bed (
    BedNum INT PRIMARY KEY,
    SectionNum INT NOT NULL,
    BStatus VARCHAR(50) NOT NULL CHECK (BStatus IN ('Available', 'Occupied')),
    FOREIGN KEY (SectionNum) REFERENCES Section(SectionNum)
);

-- Create the Prescription table
CREATE TABLE Prescription (
    PrescriptionNum INT PRIMARY KEY,
    PDate DATE NOT NULL,
    PDose VARCHAR(255)
);

-- Create the Drug table
CREATE TABLE Drug (
    DrugCode INT PRIMARY KEY,
    DrugName VARCHAR(100) NOT NULL,
    DBrand VARCHAR(50),
    DPrescription VARCHAR(255)
);

-- Create the Bill table
CREATE TABLE Bill (
    BillNum INT PRIMARY KEY,
    BAmount DECIMAL(10, 2) NOT NULL CHECK (BAmount >= 0),
    BDate DATE NOT NULL,
    BStatus VARCHAR(50) NOT NULL CHECK (BStatus IN ('Paid', 'Unpaid', 'Pending')),
    PatientID INT NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)
);

-- Create the Insurance table
CREATE TABLE Insurance (
    InsuranceID INT PRIMARY KEY,
    ICompany VARCHAR(100) NOT NULL,
    IDetails VARCHAR(255),
```

```
    IContact VARCHAR(50)
);

-- Create the MakingAppointment table
CREATE TABLE MakingAppointment (
    PatientID INT NOT NULL,
    DoctorID INT NOT NULL,
    AppointmentID INT NOT NULL,
    Date DATE NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID),
    FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID)
);

-- Create the InsuranceCoverage table
CREATE TABLE InsuranceCoverage (
    InsuranceID INT NOT NULL,
    PatientID INT NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    FOREIGN KEY (InsuranceID) REFERENCES Insurance(InsuranceID),
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)
);

-- Create the WritingPrescription table
CREATE TABLE WritingPrescription (
    PatientID INT NOT NULL,
    DoctorID INT NOT NULL,
    PrescriptionNum INT NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID),
    FOREIGN KEY (PrescriptionNum) REFERENCES Prescription(PrescriptionNum)
);

-- Create the DrugsList table
CREATE TABLE DrugsList (
    DrugCode INT NOT NULL,
    PrescriptionNum INT NOT NULL,
    FOREIGN KEY (DrugCode) REFERENCES Drug(DrugCode),
    FOREIGN KEY (PrescriptionNum) REFERENCES Prescription(PrescriptionNum)
);
```

در این مرحله تعدادی داده به پایگاه داده اضافه می کنیم:

```
INSERT INTO Section (SectionNum, SName, SHead)
VALUES
(1, 'Cardiology', 'Dr. Smith'),
(2, 'Neurology', 'Dr. Brown'),
(3, 'Pediatrics', 'Dr. Johnson');

INSERT INTO Patient (PatientID, PName, PBD, PGender, PAddress, PPhone, SectionNum)
VALUES
(1, 'John Doe', '1985-06-15', 'M', '123 Main St', '1234567890', 1),
```

```
(2, 'Jane Smith', '1990-03-25', 'F', '456 Elm St', '0987654321', 2),
(3, 'Emily Brown', '2005-08-10', 'F', '789 Oak St', '1122334455', 3);

INSERT INTO Doctor (DoctorID, DName, DAddress, DPhone, DSpecialty, DSchedule)
VALUES
(1, 'Dr. Adam White', '12 Park Ave', '5551234567', 'Cardiologist', 'Mon-Fri 9am-5pm'),
(2, 'Dr. Sarah Green', '34 Forest Rd', '5559876543', 'Neurologist', 'Tue-Thu 10am-4pm'),
(3, 'Dr. Chris Black', '56 Maple St', '5556789123', 'Pediatrician', 'Mon-Wed 8am-2pm');

INSERT INTO Appointment (AppointmentID, AStatus)
VALUES
(1, 'Scheduled'),
(2, 'Completed'),
(3, 'Canceled');

INSERT INTO ReceptionHistory (ReceptionNum, PatientID, RDate, RDischarge)
VALUES
(1, 1, '2025-01-01', '2025-01-05'),
(2, 2, '2025-01-03', '2025-01-06'),
(3, 3, '2025-01-07', NULL);

INSERT INTO Bed (BedNum, SectionNum, BStatus)
VALUES
(1, 1, 'Available'),
(2, 1, 'Occupied'),
(3, 2, 'Occupied');

INSERT INTO Prescription (PrescriptionNum, PDate, PDose)
VALUES
(1, '2025-01-02', 'Take 1 tablet daily'),
(2, '2025-01-03', 'Apply twice daily'),
(3, '2025-01-04', 'Take 2 tablets after meals');

INSERT INTO Drug (DrugCode, DrugName, DBrand, DPrescription)
VALUES
(1, 'Aspirin', 'Bayer', 'Pain relief'),
(2, 'Metformin', 'Teva', 'Diabetes treatment'),
(3, 'Amoxicillin', 'Pfizer', 'Antibiotic');

INSERT INTO Bill (BillNum, BAmount, BDate, BStatus, PatientID)
VALUES
(1, 500.00, '2025-01-05', 'Paid', 1),
(2, 300.00, '2025-01-06', 'Unpaid', 2),
(3, 200.00, '2025-01-08', 'Pending', 3);

INSERT INTO Insurance (InsuranceID, ICompany, IDetails, IContact)
VALUES
(1, 'Blue Cross', 'Covers 80% of medical expenses', 'contact@bluecross.com'),
(2, 'United Health', 'Covers 70% of outpatient expenses', 'info@unitedhealth.com'),
(3, 'Aetna', 'Covers 90% of inpatient expenses', 'support@aetna.com');

INSERT INTO MakingAppointment (PatientID, DoctorID, AppointmentID, Date)
VALUES
(1, 1, 1, '2025-01-01'),
(2, 2, 2, '2025-01-03'),
(3, 3, 3, '2025-01-07');

INSERT INTO InsuranceCoverage (InsuranceID, PatientID, StartDate, EndDate)
```

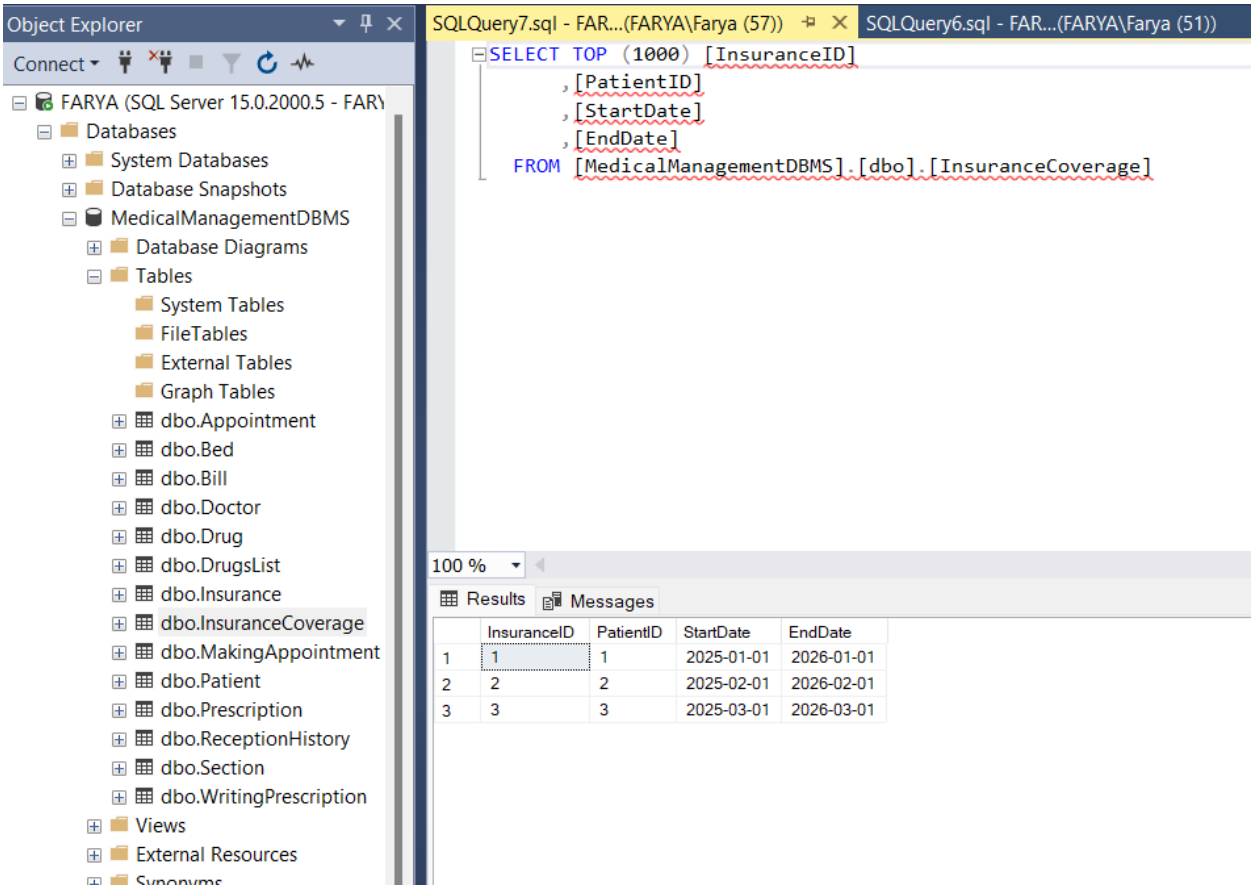
```
VALUES
(1, 1, '2025-01-01', '2026-01-01'),
(2, 2, '2025-02-01', '2026-02-01'),
(3, 3, '2025-03-01', '2026-03-01');

INSERT INTO WritingPrescription (PatientID, DoctorID, PrescriptionNum)
VALUES
(1, 1, 1),
(2, 2, 2),
(3, 3, 3);

INSERT INTO DrugsList (DrugCode, PrescriptionNum)
VALUES
(1, 1),
(2, 2),
(3, 3);
```

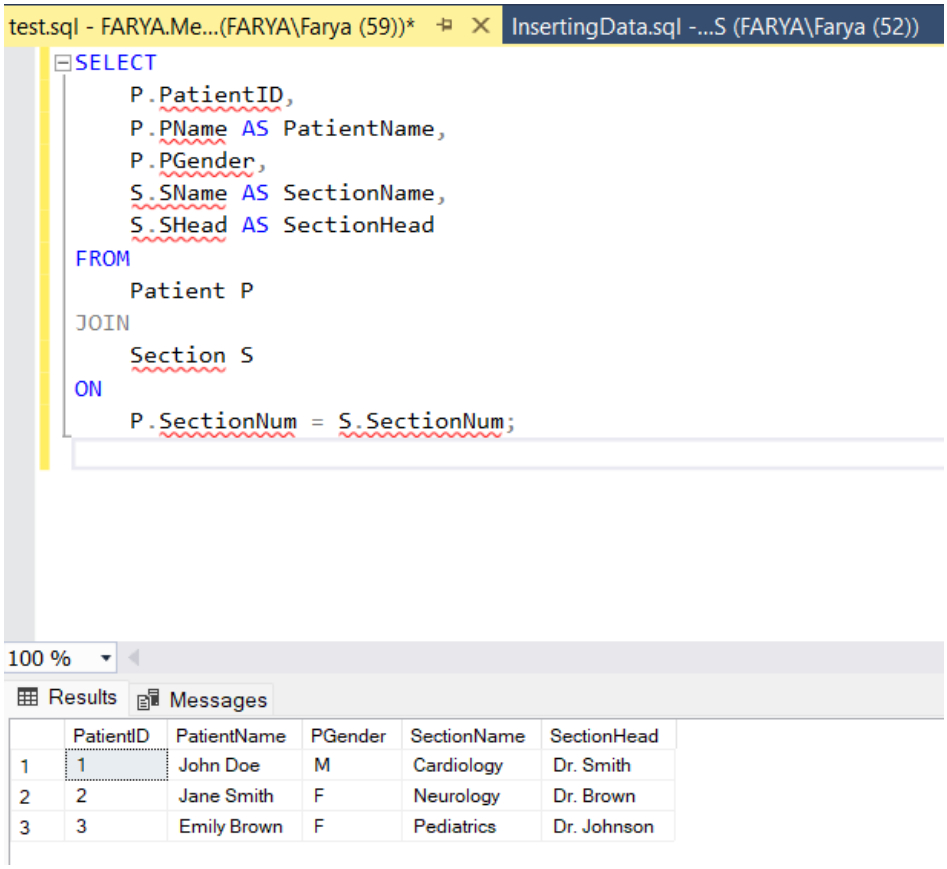
این دستورات در فایل InsertData قابل مشاهده است.

با کلیک راست بر روی هر جدول و زدن گزینه مشاهده ۱۰۰ ردیف اول جدول محتویات جدول قابل مشاهده است. البته با کوئری نوشتن هم میتوان محتویات جدول ها را مشاهده کرد.



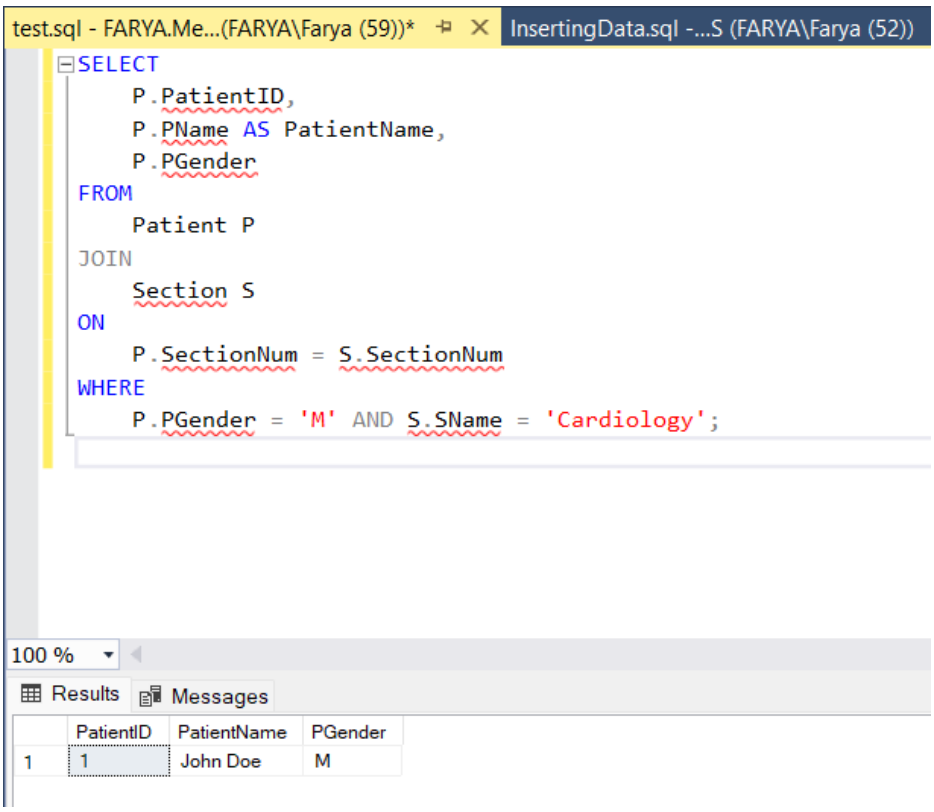
در ادامه چندین کوئری دلخواه را روی پایگاه داده مان مینویسیم و نتیجه را برمیگردانیم:

۱. نمایش تمام بیماران و اطلاعات بخش مربوطه

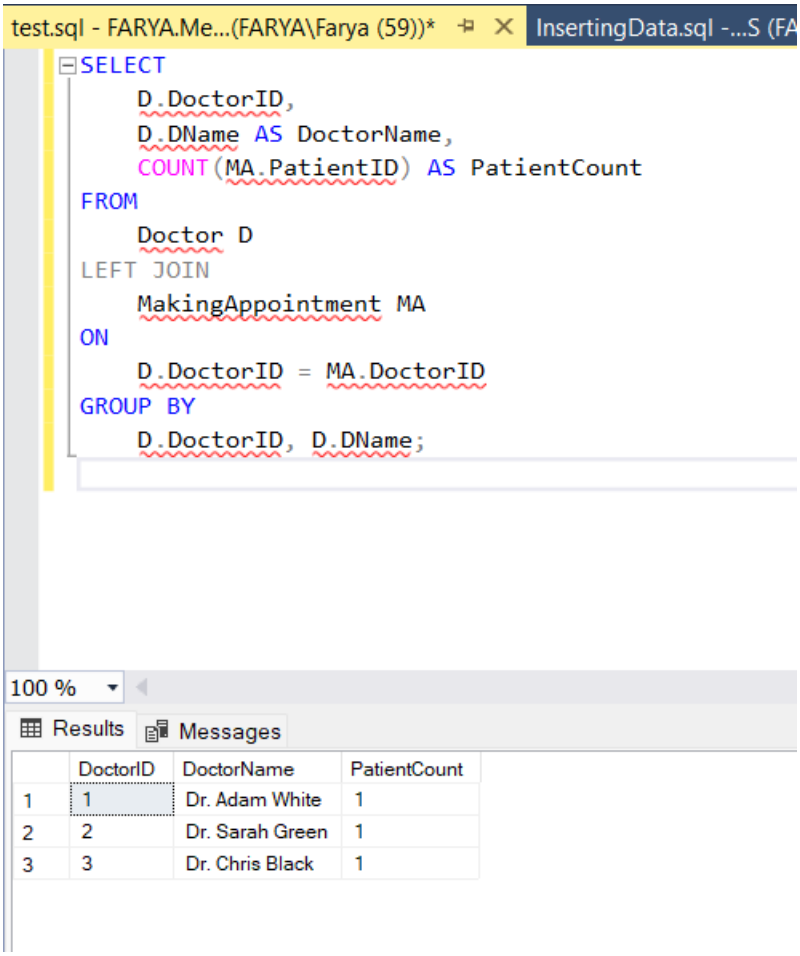


	PatientID	PatientName	PGender	SectionName	SectionHead
1	1	John Doe	M	Cardiology	Dr. Smith
2	2	Jane Smith	F	Neurology	Dr. Brown
3	3	Emily Brown	F	Pediatrics	Dr. Johnson

۲. بیماران مرد در بخش قلب‌شناسی(Cardiology)



۳. لیست پزشکان و تعداد بیماران مرتبط با هر پزشک



۴. نمایش صورتحساب بیماران به همراه وضعیت پرداخت

test.sql - FARYA.Me...(FARYA\Farya (59))* | InsertingData.sql -...S (FARY

```sql
SELECT
    B.BillNum,
    P.PName AS PatientName,
    B.BAmount AS Amount,
    B.BStatus AS PaymentStatus
FROM
    Bill B
JOIN
    Patient P
ON
    B.PatientID = P.PatientID
ORDER BY
    B.BStatus DESC;
```

100 %

Results | Messages

	BillNum	PatientName	Amount	PaymentStatus
1	2	Jane Smith	300.00	Unpaid
2	3	Emily Brown	200.00	Pending
3	1	John Doe	500.00	Paid

۵. نمایش بیمه‌های فعال بیماران

test.sql - FARYA.Me...(FARYA\Farya (59))* | InsertingData.sql -...S (FARYA\Farya (52))

```
    IC.PatientID,
    P.PName AS PatientName,
    I.ICompany AS InsuranceCompany,
    IC.StartDate,
    IC.EndDate
FROM
    InsuranceCoverage IC
JOIN
    Insurance I
ON
    IC.InsuranceID = I.InsuranceID
JOIN
    Patient P
ON
    IC.PatientID = P.PatientID
WHERE
    IC.EndDate > GETDATE();
```

100 %

Results | Messages

	PatientID	PatientName	InsuranceCompany	StartDate	EndDate
1	1	John Doe	Blue Cross	2025-01-01	2026-01-01
2	2	Jane Smith	United Health	2025-02-01	2026-02-01
3	3	Emily Brown	Aetna	2025-03-01	2026-03-01

۶. نمایش نسخه‌های تجویز شده برای هر بیمار