

به نام خدا



دانشکده مهندسی برق و کامپیوتر

پایگاه داده دو

طراحی انبار داده کلینیک درمانی

اعضای تیم:

وانیا کرد – 40129043

زهرا ابوطالبی – 40116943

استاد:

دکتر علیرضا بصیری

پایگاه داده Source



معرفی پایگاه داده‌ی کلینیک درمانی (Clinic_DB)

پایگاه داده‌ی Clinic_DB برای مدیریت اطلاعات یک کلینیک درمانی خصوصی طراحی شده است. این کلینیک خدماتی جامع نظیر معاینه، درمان‌های سرپایی، بستری‌های کوتاه‌مدت، و همچنین فرآیندهای مالی و اداری مرتبط را ارائه می‌دهد. هدف اصلی این پایگاه داده، ایجاد یک سیستم مرکزی برای مدیریت کارآمد داده‌های بیماران، پرسنل، مراجعات، سوابق پزشکی، صورتحساب‌ها و

خدمات ارائه شده است. در دنیای امروز، ارائه خدمات درمانی باکیفیت و رقابتی، نیازمند یک زیرساخت اطلاعاتی دقیق و منسجم است تا فرآیندهای اداری، مالی و درمانی با حداکثر دقت و سرعت انجام شوند.

این پایگاه داده به عنوان هسته مرکزی نرم افزار مدیریتی کلینیک عمل کرده و با فراهم آوردن بستری برای ثبت دقیق اطلاعات، به اهداف کلیدی زیر دست می یابد:

- **افزایش کیفیت خدمات به بیمار:** از طریق دسترسی سریع به سوابق کامل پزشکی، حساسیت ها و تاریخچه درمان.
- **بهبود بهره وری عملیاتی:** با خودکارسازی فرآیندهای ثبت اطلاعات ویزیت، صدور نسخه و صورتحساب.
- **مدیریت مالی دقیق:** با ثبت تمام خدمات ارائه شده و پیگیری صورتحساب ها و پرداخت ها.
- **فراهم سازی داده های دقیق و منسجم:** به عنوان زیرساخت اصلی برای تحلیل های مدیریتی و گزارش گیری های آتی در انبار داده.

در ادامه، ساختار جداول، به همراه شرح کوتاهی از نقش و مهم ترین فیلدهای هر کدام آورده شده است.

تشریح ساختار دیتابیس و جداول

دیتابیس به صورت یک مدل رابطه ای (Relational) طراحی شده و جداول آن در چند بخش منطقی دسته بندی می شوند:

بخش اول: مدیریت اطلاعات پایه بیماران

این بخش شامل اطلاعات هویتی، تماس و بیمه ای بیماران است.

- **Patients (بیماران):** جدول مرکزی برای نگهداری اطلاعات دموگرافیک هر بیمار مانند نام، کد ملی، تاریخ تولد و اطلاعات تماس اولیه.
- **PatientContacts (مخاطبین بیمار):** برای ثبت اطلاعات یک یا چند مخاطب اضطراری برای هر بیمار.
- **Cities و Provinces (استان ها و شهرها):** جداول پایه برای مدیریت اطلاعات جغرافیایی.
- **PatientAddresses (آدرس های بیمار):** نگهداری آدرس های مختلف بیمار (مثلاً منزل، محل کار).
- **InsuranceCompanies (شرکت های بیمه):** لیست شرکت های بیمه طرف قرارداد با کلینیک.
- **PatientInsurances (بیمه های بیمار):** برای ثبت اطلاعات بیمه نامه هر بیمار و ارتباط آن با شرکت بیمه مربوطه.

بخش دوم: سوابق بالینی و کاتالوگ های پزشکی

این بخش به مدیریت تاریخچه پزشکی بیماران و اطلاعات مرجع پزشکی می پردازد.

- **Allergies (حساسیت ها):** یک کاتالوگ از انواع حساسیت های دارویی، غذایی و محیطی.
- **PatientAllergies (حساسیت های بیمار):** ثبت حساسیت های مشخص هر بیمار با ارجاع به جدول Allergies.
- **Diseases (بیماری ها):** کاتالوگ بیماری ها بر اساس کدهای استاندارد مانند ICD-10.

- PatientMedicalHistory (سوابق پزشکی بیمار): مهم‌ترین جدول این بخش که تاریخچه بیماری‌های هر فرد (مزمین یا حاد) را ثبت می‌کند.
- Drugs (داروها): لیست داروهای مورد استفاده در کلینیک.
- Services (خدمات): لیست تمام خدمات قابل ارائه در کلینیک (مانند ویزیت، نوار قلب، تزریقات) به همراه هزینه پایه.

بخش سوم: مدیریت کارکنان و ساختار کلینیک

اطلاعات مربوط به پرسنل و بخش‌های مختلف کلینیک در این قسمت قرار دارد.

- Departments (بخش‌ها): تعریف بخش‌های مختلف کلینیک (مانند داخلی، اطفال، زنان).
- Staff (کارکنان): اطلاعات تمام پرسنل کلینیک اعم از پزشک، پرستار و پذیرش.
- Specializations (تخصص‌ها): لیست تخصص‌های پزشکی (مانند قلب، پوست).
- Doctors (پزشکان): جدولی برای نگهداری اطلاعات تکمیلی پزشکان (که خود نوعی از Staff هستند) مانند شماره نظام پزشکی و تخصص آن‌ها.

بخش چهارم: مدیریت ویزیت و درمان

این بخش، هسته عملیاتی دیتابیس است و تمام رویدادهای مربوط به مراجعه بیمار را ثبت می‌کند.

- VisitTypes: انواع مختلف ویزیت مثلاً اورژانسی ...
- Visits (ویزیت‌ها): جدول مرکزی که هر رکورد آن نشان‌دهنده یک مراجعه بیمار به کلینیک است. این جدول به بیمار، پزشک و بخش مربوطه متصل است.
- Vitals (علائم حیاتی): ثبت اطلاعاتی مانند فشار خون، ضربان قلب و دمای بیمار در هر ویزیت.
- Prescriptions (نسخه‌ها): ثبت اطلاعات کلی هر نسخه که در یک ویزیت صادر می‌شود.
- PrescriptionItems (اقلام نسخه): جزئیات هر نسخه، شامل داروهای تجویز شده، دوز و دستور مصرف.
- VisitServices (خدمات ویزیت): ثبت خدماتی که در طول یک ویزیت به بیمار ارائه شده است (مثلاً نوار قلب یا تزریقات).
- admissions (بستری): در صورتی که مراجعه فرد به بستری منجر شود در این جدول ثبت می‌شود.

بخش پنجم: مدیریت مالی و صورتحساب

این بخش مسئولیت مدیریت جنبه‌های مالی مراجعات را بر عهده دارد.

- Invoices (صورتحساب‌ها): برای هر ویزیت، یک صورتحساب کلی صادر می‌شود.
- InvoiceItems (اقلام صورتحساب): جزئیات هر صورتحساب که به • VisitServices ارجاع می‌دهد و هزینه نهایی را مشخص می‌کند.

- Payments (پرداخت‌ها): ثبت پرداخت‌های انجام‌شده توسط بیمار یا شرکت بیمه برای تسویه یک صورتحساب.

ارتباطات کلیدی و جریان داده

قدرت این دیتابیس در ارتباطات منطقی و مستحکم بین جداول آن نهفته است. جریان داده یک مراجعه بیمار به کلینیک به صورت زیر است:

- ثبت بیمار: اطلاعات یک فرد در جدول Patients ثبت می‌شود.
- ایجاد ویزیت: بیمار به یک پزشک (Doctors) در یک بخش (Departments) مراجعه کرده و یک رکورد در جدول Visits برای او ایجاد می‌شود.
- فرآیند درمان:
 - علائم حیاتی در Vitals ثبت می‌شود.
 - پزشک ممکن است یک بیماری را تشخیص دهد که به عنوان یک رکورد جدید در PatientMedicalHistory ثبت شده و به ویزیت فعلی ارجاع داده می‌شود.
 - یک نسخه در Prescriptions و اقلام آن در PrescriptionItems ثبت می‌گردد.
 - خدمات جانبی در VisitServices ثبت می‌شوند.
- فرآیند مالی:
 - تمام VisitServices مربوط به آن ویزیت، در جدول InvoiceItems جمع‌آوری شده و به یک Invoice واحد متصل می‌شوند.
 - در نهایت، بیمار یا بیمه، صورتحساب را از طریق یک یا چند Payments تسویه می‌کند.

جمع‌بندی

این پایگاه داده طراحی شده تا اطلاعات بالینی، مالی و مدیریتی یک کلینیک درمانی کوچک را به صورت منسجم ذخیره کرده و زمینه‌ای برای تحلیل، گزارش‌گیری و مدیریت مؤثرتر فراهم کند. ساختار جدول‌ها و ارتباط بین آن‌ها به گونه‌ای انتخاب شده که هم نرمال‌سازی داده‌ها رعایت شود و هم امکان توسعه‌ی آینده در سیستم وجود داشته باشد.

طراحی انبار داده برای کلینیک درمانی

دیتا مارت ها:

1. Patient Data Mart

شامل اطلاعات تحلیلی در مورد رفتار بیماران، مراجعه های مکرر، گروه های سنی و بیماری ها

2. Financial Data Mart

شامل اطلاعات درآمد کلینیک به تفکیک زمان، پزشک، بیمار و خدمات درمانی.

انبار داده – مدل استار

Dimensions .1

Dim_Patient	
PatientID	int
NationalCode	nvarchar(12)
FirstName	nvarchar(60)
LastName	nvarchar(60)
FatherName	nvarchar(60)
DateOfBirth	date
phoneNumber	nvarchar(20)
Gender	nvarchar(10)
MaritalStatus	nvarchar(25)
Occupation	nvarchar(60)
BloodType	nvarchar(5)
IsAlive	bit
DateOfDeath	date
RegistrationDate	date
AddressID	int
AddressLine	nvarchar(255)
AddressType	nvarchar(15)
PostalCode	nvarchar(15)
CityID	int
CityName	nvarchar(60)
ProvinceID	int
ProvinceName	nvarchar(60)

Dim_DoctorStaff	
StaffID	int
NationalCode	nvarchar(12)
FirstName	nvarchar(60)
LastName	nvarchar(60)
Gender	nvarchar(10)
DateOfBirth	date
Role	nvarchar(60)
CurrentDepartmentID	int
CurrentDepartmentName	nvarchar(90)
OriginalDepartmentID	int
OriginalDepartmentName	nvarchar(90)
EffectiveDate	date
phoneNumber	nvarchar(20)
Email	nvarchar(50)
HireDate	date
Salary	decimal(12,2)
IsActive	bit
IsDoctor	bit
MedicalLicenceNumber	nvarchar(60)
SpecializationID	int
SpecializationName	nvarchar(50)
MedicalDegree	nvarchar(60)
PracticeLicenceExpiryDate	date
YearsOfExperience	int

Dim_Date	
TimeKey	int
FullDateAlternateKey	nvarchar(10)
PersianFullDateAlternateKey	nvarchar(10)
DayNumberOfWeek	int
PersianDayNumberOfWeek	int
EnglishDayNameOfWeek	nvarchar(10)
PersianDayNameOfWeek	nvarchar(10)
DayNumberOfMonth	int
PersianDayNumberOfMonth	int
DayNumberOfYear	int
PersianDayNumberOfYear	int
WeekNumberOfYear	int
PersianWeekNumberOfYear	int
EnglishMonthName	nvarchar(10)
PersianMonthName	nvarchar(10)
MonthNumberOfYear	int
PersianMonthNumberOfYear	int
CalendarQuarter	int
PersianCalendarQuarter	int
CalendarYear	int
PersianCalendarYear	int
CalendarSemester	int
PersianCalendarSemester	int

Dim_Department	
DepartmentSK	int
DepartmentID	int
DepartmentName	nvarchar(90)
ManagerStaffID	int
ManagerFullName	nvarchar(101)
ManagerRole	nvarchar(50)
StartDate	date
EndDate	date
CurrentFlag	bit

Dim_Disease	
DiseaseID	int
ICD10_Code	nvarchar(10)
DiseaseName	nvarchar(90)
IsChronic	bit
Description	ntext

Dim_Service	
ServiceSK	int
ServiceID	int
ServiceName	nvarchar(90)
ServiceCategory	nvarchar(60)
BaseCost	decimal(13,2)
StartDate	date
EndDate	date
CurrentFlag	bit

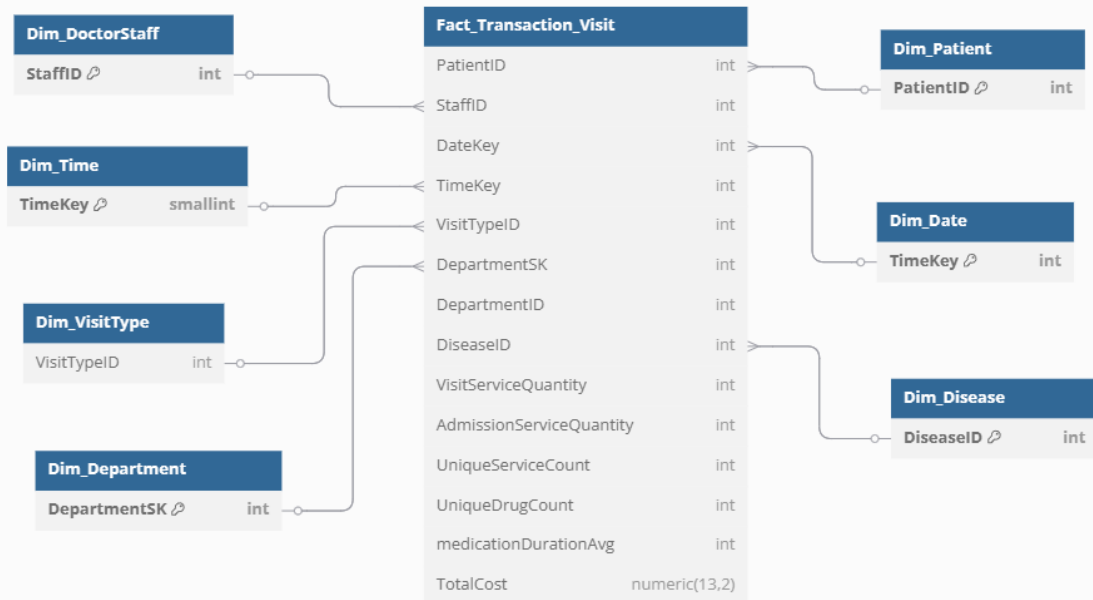
Dim_Insurance	
InsuranceCoID	int
CompanyName	nvarchar(100)

Dim_Time	
TimeKey	smallint
HourNo	tinyint
MinuteNo	tinyint

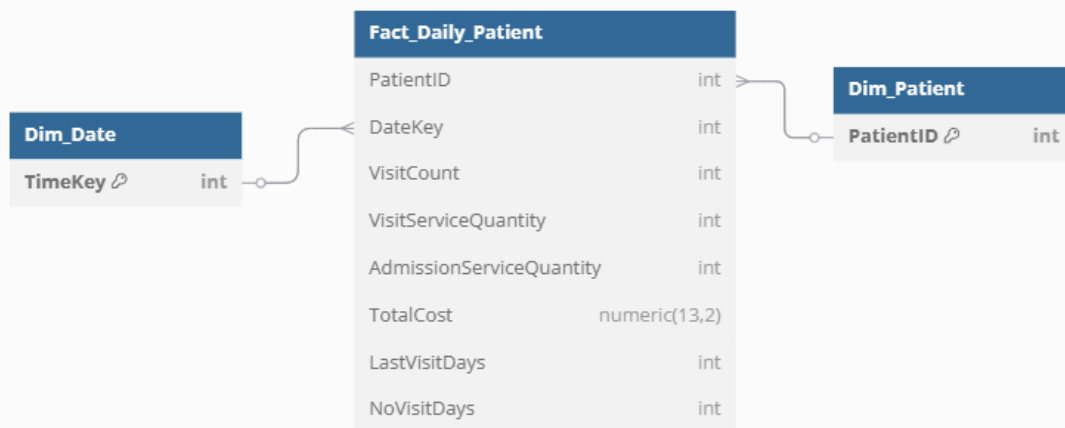
Dim_VisitType	
VisitTypeID	int
VisitTypeName	nvarchar(60)

Patient Data Mart .2

شامل اطلاعات تحلیلی در مورد رفتار بیماران، مراجعه های مکرر و بیماری ها

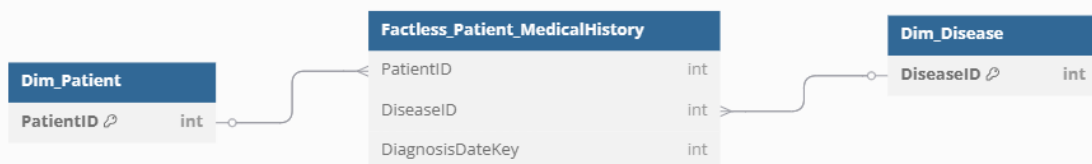


dbdiagram.io



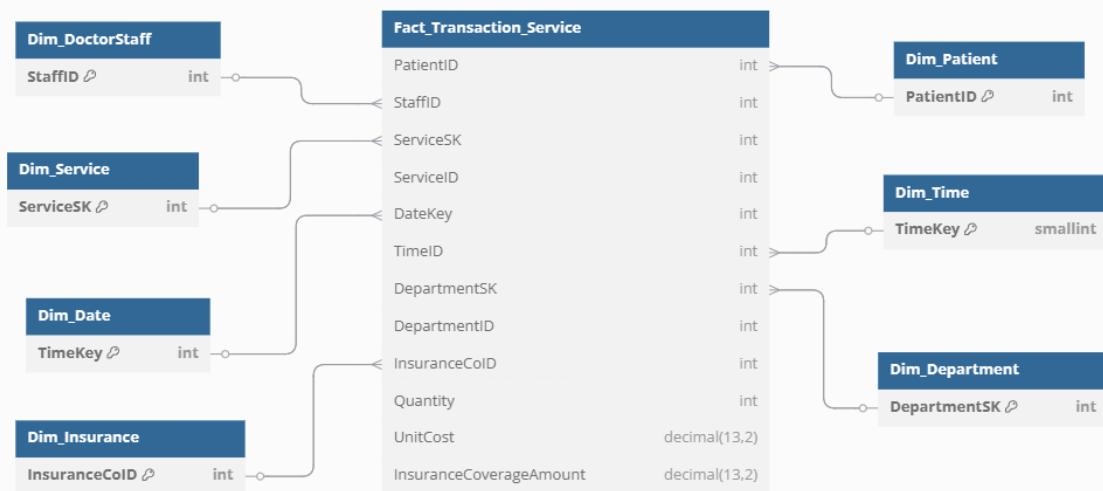
dbdiagram.io

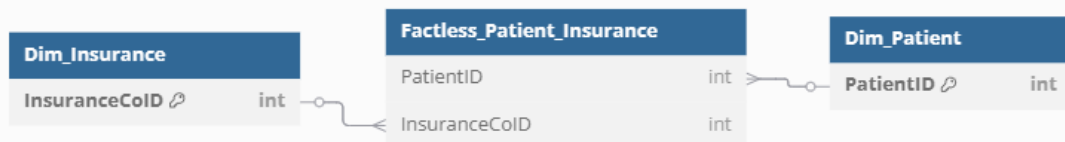
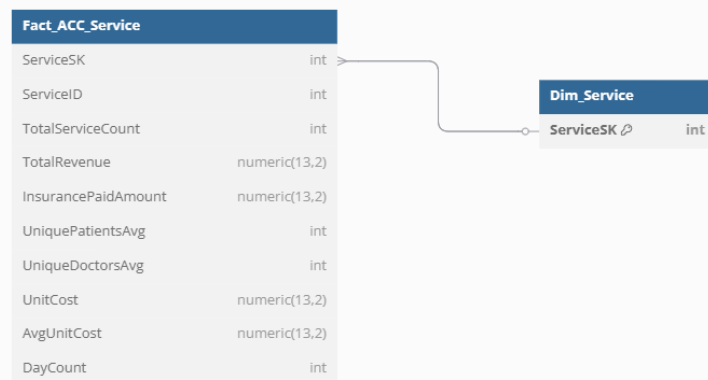
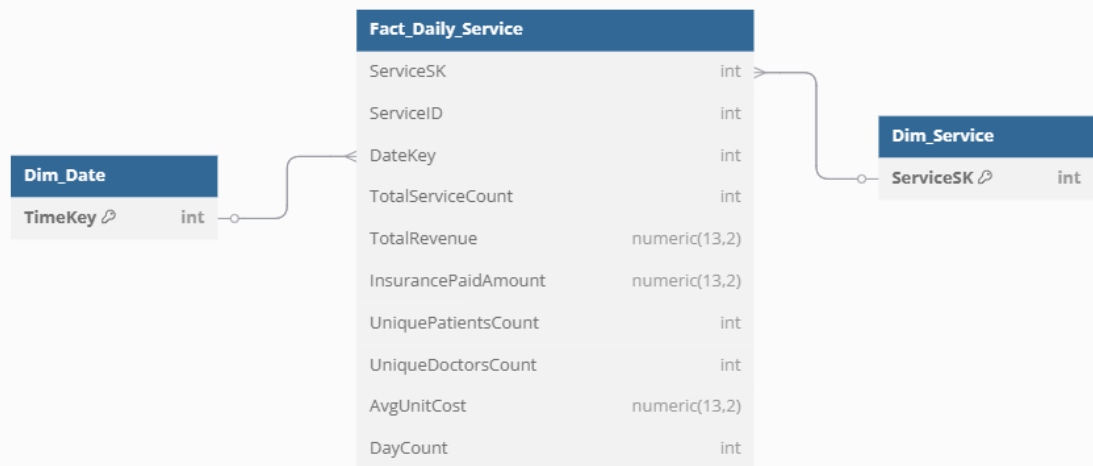
Fact_ACC_Patient	
PatientID	int
TotalVisitCount	int
VisitServiceQuantity	int
AdmissionServiceQuantity	int
TotalCost	numeric(13,2)
LastVisitDays	int
NoVisitDays	int



Financial Data Mart .3

شامل اطلاعات درآمد کلینیک به تفکیک زمان، پزشک، بیمار و خدمات درمانی.





مستند ETL

جداول مشترک

1. بعد تاریخ

این بعد یک بار در ابتدا به صورت مستقل ساخته میشود و در مراحل بعد فقط استفاده می‌شود.

Dim_Date			
Data type	field name	Data type	field name
INT	WeekNumberOfYear	INT	DateKey
INT	PersianWeekNumberOfYear	VARCHAR(10)	FullDateAlternateKey
VARCHAR(10)	EnglishMonthName	VARCHAR(10)	PersianFullDateAlternateKey
VARCHAR(10)	PersianMonthName	INT	DayNumberOfWeek
INT	MonthNumberOfYear	INT	PersianDayNumberOfWeek
INT	PersianMonthNumberOfYear	VARCHAR(10)	EnglishDayNameOfWeek
INT	CalendarQuarter	VARCHAR(10)	PersianDayNameOfWeek
INT	PersianCalendarQuarter	INT	DayNumberOfMonth
INT	CalendarYear	INT	PersianDayNumberOfMonth
INT	PersianCalendarYear	INT	DayNumberOfYear
INT	CalendarSemester	INT	PersianDayNumberOfYear
INT	PersianCalendarSemester		

2. بعد زمان

این بعد شامل ساعت و دقیقه است و فقط یکبار ساخته می‌شود.

Dim_Patient		
description	Data type	field name
کلید زمان، HHMM ساعت × 100 + دقیقه	INT	TimeKey
ساعت (از 0 تا 24)	TINYINT	HourNo
دقیقه (از 0 تا 59)	TINYINT	MinuteNo

3. بعد بیمار

اطلاعات کلی بیمار در این بعد قرار دارد. لازم است جداول PatientAddresses، Patient، Cities و Provinces در پایگاه داده منبع بر روی فیلد های مناسب جوین شوند تا اطلاعات این جدول به دست بیاید.

Dim_Patient		
description	Data type	field name
شناسه یکتای بیمار که از ستون PatientID در جدول Patient در پایگاه داده منبع استخراج می شود.	INT	PatientID
کد ملی بیمار؛ از ستون NationalCode در جدول Patient استخراج می شود.	VARCHAR(12)	NationalCode
نام بیمار؛ از ستون FirstName در جدول Patient استخراج می شود.	VARCHAR(60)	FirstName
نام خانوادگی بیمار؛ از ستون LastName در جدول Patient استخراج می شود.	VARCHAR(60)	LastName
نام پدر بیمار؛ از ستون FatherName در جدول Patient استخراج می شود.	VARCHAR(60)	FatherName
تاریخ تولد بیمار؛ از ستون DateOfBirth در جدول Patient استخراج می شود.	DATE	DateOfBirth
شماره تماس بیمار؛ از ستون PhoneNumber در جدول Patient استخراج می شود.	VARCHAR(20)	phoneNumber
جنسیت بیمار (مانند Male یا Female)؛ از ستون Gender در جدول Patient استخراج می شود.	VARCHAR(10)	Gender
وضعیت تأهل بیمار (مانند Single یا Married)؛ از ستون MaritalStatus در جدول Patient استخراج می شود.	VARCHAR(25)	MaritalStatus
شغل بیمار؛ از ستون Occupation در جدول Patient استخراج می شود.	VARCHAR(60)	Occupation
گروه خونی بیمار (مانند $A+, O-$)؛ از ستون BloodType در جدول Patient استخراج می شود.	VARCHAR(5)	BloodType
مشخص می کند که آیا بیمار زنده است یا خیر؛ از ستون IsAlive در جدول Patient استخراج می شود.	BIT	IsAlive
در صورت فوت بیمار، تاریخ فوت او؛ از ستون DateOfDeath در جدول Patient استخراج می شود.	DATE	DateOfDeath
تاریخ ثبت نام یا ایجاد پرونده بیمار در سامانه؛ از ستون RegistrationDate در جدول PatientAddresses استخراج می شود.	DATE	RegistrationDate
شناسه آدرس مرتبط با بیمار؛ از ستون AddressID موجود در جدول PatientAddresses استخراج می شود.	INT	AddressID
آدرس کامل بیمار به صورت خطی؛ از ستون AddressLine در جدول PatientAddresses استخراج می شود.	VARCHAR(255)	AddressLine
نوع آدرس (مانند منزل یا محل کار)؛ از ستون AddressType در جدول PatientAddresses استخراج می شود.	VARCHAR(15)	AddressType

PostalCode	VARCHAR(15)	کد پستی آدرس بیمار؛ از ستون PostalCode جدول PatientAddresses استخراج می‌شود.
CityID	INT	شناسه شهر مرتبط با آدرس؛ از ستون CityID جدول PatientAddresses استخراج می‌شود.
CityName	VARCHAR(60)	نام شهر مربوط به آدرس بیمار؛ از ستون CityName جدول Cities استخراج می‌شود.
ProvinceID	INT	شناسه استان مربوط به شهر آدرس؛ از ستون ProvinceID جدول Cities استخراج می‌شود.
ProvinceName	VARCHAR(60)	نام استان مربوط به شهر آدرس؛ از ستون ProvinceName جدول Provinces استخراج می‌شود.

4. بعد کارکنان

اطلاعات کلی کارکنان کلینیک از جمله پزشکان در این بعد قرار دارد. لازم است جداول Doctors، Staff و Departments و Specialization در پایگاه داده منبع بر روی فیلدهای مناسب جوین شوند تا اطلاعات این جدول به دست بیاید. در صورتی که فردی پزشک نباشد در فیلدهای مربوط به اطلاعات پزشک NULL قرار داده می‌شود. (علت یکی کردن این بعد این بود که ممکن است برخی خدمات توسط افراد دیگر مانند پرستار ارائه شود) SCD نوع 3 در این بعد لحاظ شده است (روی فیلد دپارتمان کارمند/پزشک)

Dim_DoctorStaff		
description	Data type	field name
شناسه یکتای کارمند یا ارائه‌دهنده خدمات درمانی از ستون StaffID در جدول Staff به دست می‌آید	INT	StaffID
کد ملی؛ از ستون NationalCode در جدول Staff به دست می‌آید	VARCHAR(12)	NationalCode
نام کارمند یا پزشک؛ از ستون FirstName در جدول Staff به دست می‌آید	VARCHAR(60)	FirstName
نام خانوادگی؛ از ستون LastName در جدول Staff به دست می‌آید	VARCHAR(60)	LastName
جنسیت؛ از ستون Gender در جدول Staff به دست می‌آید	VARCHAR(10)	Gender
تاریخ تولد؛ از ستون DateOfBirth در جدول Staff به دست می‌آید	DATE	DateofBirth
سمت یا عنوان شغلی کارمند (مثل پزشک، پرستار ...) از ستون Role در جدول Staff به دست می‌آید	VARCHAR(60)	Role
شناسه دپارتمانی که کارمند در آن مشغول است؛ از ستون DepartmentID در جدول Staff به دست می‌آید	INT	CurrentDepartmentID
<u>SCD نوع 3</u>		

نام دپارتمان؛ از ستون DepartmentName در جدول Departments به دست می‌آید، با استفاده از DepartmentID	VARCHAR(60)	CurrentDepartmentName
SCD نوع 3		
شناسه دپارتمانی که کارمند قبلاً در آن بوده است	INT	OriginalDepartmentID
نام دپارتمان قبلی	VARCHAR(60)	OriginalDepartmentName
تاریخ تغییر دپارتمان (برای SCD نوع 3)	Date	EffectiveDate
شماره تماس؛ از ستون phoneNumber در جدول Staff به دست می‌آید	VARCHAR(20)	phoneNumber
آدرس ایمیل؛ از ستون Email در جدول Staff به دست می‌آید	VARCHAR(60)	Email
تاریخ استخدام؛ از ستون HireDate در جدول Staff به دست می‌آید	DATE	HireDate
میزان حقوق ماهیانه؛ از ستون Salary در جدول Staff به دست می‌آید	Decimal(12,2)	Salary
وضعیت فعال بودن کارمند در سیستم (۱: فعال، ۰: غیرفعال) از ستون IsActive در جدول Staff به دست می‌آید	BIT	IsActive
نشان می‌دهد که آیا کارمند پزشک است یا خیر؛ از ستون Role در جدول Staff به دست می‌آید. در صورتی که برابر با doctor باشد.	BIT	IsDoctor
شماره نظام پزشکی پزشک؛ از ستون MedicalLicenceNumber در جدول Doctors به دست می‌آید، در صورت پزشک بودن	VARCHAR(60)	MedicalLicenceNumber
شناسه تخصص پزشکی؛ از ستون SpecializationID در جدول Doctors به دست می‌آید، در صورت پزشک بودن	INT	SpecializationID
نام تخصص پزشکی؛ از جدول Specialization با استفاده از SpecializationID به دست می‌آید، در صورت پزشک بودن	VARCHAR(60)	SpecializationName
مدرک تحصیلی پزشکی (مثل دکترای عمومی، تخصص، فوق تخصص) از ستون MedicalDegree در جدول Doctors به دست می‌آید در صورت پزشک بودن	VARCHAR(60)	MedicalDegree
تاریخ انقضای مجوز طبابت؛ ستون PracticeLicenceExpiryDate در جدول Doctors به دست می‌آید، در صورت پزشک بودن	DATE	PracticeLicenceExpiryDate
تعداد سال‌های تجربه کاری پزشک از ستون YearsOfExperience در جدول Doctors به دست می‌آید و مربوط به سوابق کاری قبلی است، در صورت پزشک بودن	INT	YearsOfExperience

5. بعد سرویس

اطلاعات کلی سرویس‌های ارائه شده در کلینیک در این بعد قرار دارد. اطلاعات این بعد از جدول Services در پایگاه داده منبع به دست می‌آید. همچنین SCD نوع 2 در این بعد بر روی ستون BaseCost لحاظ شده است.

Dim_Service		
description	Data type	field name
surrogate key - کلید عددی جایگزین که در انبار داده استفاده می‌شود. این کلید برای لحاظ کردن SCD نوع 2 ایجاد شده است.	INT	ServiceSK
کلید طبیعی - شناسه یکتای سرویس که از جدول Services به دست می‌آید	INT	ServiceID
نام سرویس ارائه‌شده؛ از جدول Services به دست می‌آید	VARCHAR(90)	ServiceName
دسته‌بندی سرویس؛ از جدول Services به دست می‌آید	VARCHAR(60)	ServiceCategory
هزینه پایه سرویس؛ از جدول Services به دست می‌آید <u>SCD نوع 2</u>	DECIMAL(13,2)	BaseCost
تاریخ ایجاد این رکورد. برای SCD نوع 2	DATE	StartDate
تاریخ پایان این رکورد. برای SCD نوع 2	DATE	EndDate
نشان می‌دهد که ان رکورد معتبر است یا خیر	BIT	CurrentFlag

6. بعد دپارتمان

اطلاعات هر بخش (دپارتمان) در این بعد قرار می‌گیرد. اطلاعات این بخش از جویین جداول Departments و Staff روی فیلد ManagerStaffID به دست می‌آید. همچنین SCD نوع 2 بر روی ستون ManagerStaffID لحاظ شده است.

Dim_Department		
description	Data type	field name
surrogate key - کلید عددی جایگزین که در انبار داده استفاده می‌شود. این کلید برای لحاظ کردن SCD نوع 2 ایجاد شده است.	INT	DepartmentSK
کلید طبیعی - شناسه یکتای بخش که از جدول Departments به دست می‌آید	INT	DepartmentID
نام بخش؛ از جدول Departments به دست می‌آید	VARCHAR(90)	DepartmentName
شناسه ی مدیر بخش؛ از جدول Departments به دست می‌آید <u>SCD نوع 2</u>	VARCHAR(60)	ManagerStaffID
نام کامل مدیر بخش؛ از ستون های FirstName و LastName در جدول Staff به دست می‌آید.	DECIMAL(13,2)	ManagerFullName
نقش مدیر بخش؛ از ستون Role در جدول Staff به دست می‌آید.	DATE	ManagerRole
تاریخ ایجاد این رکورد. برای SCD نوع 2		StartDate
تاریخ پایان این رکورد. برای SCD نوع 2	DATE	EndDate
نشان می‌دهد که ان رکورد معتبر است یا خیر	BIT	CurrentFalg

7. بعد بیمه

اطلاعات مربوط به شرکت های بیمه در این بعد قرار می گیرد. اطلاعات این بعد از جدول InsuranceCompanies در پایگاه داده منبع به دست می آید.

Dim_Insurance		
description	Data type	field name
شناسه شرکت بیمه؛ از فیلد InsuranceCoID موجود در جدول InsuranceCompanies به دست می آید.	INT	InsuranceCoID
نام شرکت بیمه؛ از فیلد CompanyName موجود در جدول InsuranceCompanies به دست می آید.	VARCHAR(100)	CompanyName

جداول مارت مالی

• جداول فکت

1. فکت تراکنشی سرویس

به ازای هر سرویس که یک بیمار دریافت می کند و صورتحساب برای آن صادر می شود، یک رکورد در این فکت قرار می گیرد. برای این کار از جدول invoiceItems استفاده میشود و برای به دست آوردن اطلاعات مربوطه این جدول با جدول visitServices، Patients، Invoices و PatientInsurances موجود در پایگاه داده منبع جویین می شود.

Fact_Transaction_Service		
description	Data type	field name
کلید بعد بیمار؛ جدول invoiceItems با visitServices و دوباره با Visits جویین میشود. از PatientID در جدول Visits به دست می آید.	int	PatientID
کلید بعد کارکنان؛ مشابه کلید بعد بیمار از جدول Visits و ستون DoctorStaffID به دست می آید.	int	StaffID
کلید بعد سرویس؛ از ستون ServiceID در جدول visitServices استفاده میشود و با جویین روی بعد Dim_service در انبار داده ServiceSK این جدول در این قسمت قرار می گیرد.	int	ServiceSK
کلید طبیعی سرویس از ستون ServiceID در جدول visitServices به دست می آید	int	ServiceID
کلید بعد تاریخ از ستون ExecutionDateTime در جدول visitServices به دست می آید.	int	DateKey
کلید بعد زمان از ستون ExecutionDateTime در جدول visitServices به دست می آید.	int	TimeKey
کلید بعد دپارتمان از جدول Visits و ستون DoctorStaffID استفاده میشود و با جویین روی بعد Dim_Department در انبار داده، ستون DepartmentSK در این قسمت قرار میگیرد.	int	DepartmentSK
کلید طبیعی دپارتمان	int	DepartmentID

از جدول Visits و ستون DepartmentID به دست می‌آید.		
کلید بعد بیمه جدول Invoiceltems با Invoices و دوباره با PatientInsurances در پایگاه داده منبع جوین می‌شوند. از ستون InsuranceCoID در جدول PatientInsurances به دست می‌آید.	int	InsuranceCoID
تعداد واحد سرویس استفاده شده از ستون Quantity جدول Invoiceltems به دست می‌آید	int	Quantity
قیمت سرویس به ازای هر واحد از ستون UnitCost جدول Invoiceltems به دست می‌آید	numeric(13,2)	UnitCost
مبلغ پوشش داده شده از سمت بیمه از ستون InsuranceCoverageAmount جدول Invoiceltems به دست می‌آید.	numeric(13,2)	InsuranceCoverageAmount

2. فکت دوره ای روزانه خدمات (سرویس ها)

در این فکت در هر روز یک رکورد برای هر سرویس ثبت می‌شود و وضعیت کلی سرویس در آن روز را مشخص می‌کند.

Fact_Daily_Service		
description	Data type	field name
کلید بعد سرویس از ستون ServiceSK در جدول Dim_Service در انبار داده به دست می‌آید.	int	ServiceSK
کلید طبیعی سرویس از ستون ServiceID در جدول Dim_Service در انبار داده به دست می‌آید.	int	ServiceID
کلید تاریخ از ستون DateKey جدول Dim_Date در انبار داده بدست می‌آید	int	DateKey
مجموع دفعات ارائه این سرویس در این روز. در جدول Fact_Transaction_Service در انبار داده بر اساس روز و ServiceSK گروپ بای انجام می‌دهیم و تعداد را محاسبه می‌کنیم.	int	TotalServiceCount
مجموع درآمد حاصل از این سرویس در روز در جدول Fact_Transaction_Service در انبار داده بر اساس روز و ServiceSK گروپ بای و مجموع $Quantity \times UnitCost$ محاسبه می‌کنیم.	numeric(13,2)	TotalRevenue
مبلغی که بیمه‌ها بابت این سرویس در این روز پرداخت کردند.	numeric(13,2)	InsurancePaidAmount

در جدول Fact_Transaction_Service بر اساس روز و ServiceSK گروپ بای و مجموع InsuranceCoverageAmount را محاسبه میکنیم.		
تعداد بیماران یکتایی که این سرویس را دریافت کردند. در جدول Fact_Transaction_Service در انبار داده بر اساس روز و ServiceSK گروپ بای می‌کنیم و تعداد یکتای PatientID را محاسبه میکنیم.	int	UniquePatientsCount
تعداد پزشکان یکتایی که این سرویس را ارائه کردند. در جدول Fact_Transaction_Service در انبار داده بر اساس روز و ServiceSK گروپ بای می‌کنیم و تعداد یکتای StaffID را محاسبه میکنیم.	int	UniqueDoctorsCount
میانگین مبلغ پایه سرویس از گذشته تا کنون در جدول Fact_Daily_Service در انبار داده با استفاده از میانگین روز قبل و BaseCost امروز برای این سرویس میانگین BaseCost سرویس را محاسبه میکنیم	numeric(13,2)	AvgUnitCost
تعداد روز های این سرویس در در جدول Fact_Daily_Service در انبار داده DayCount روز قبل را بدست آورده و به علاوه یک می‌کنیم. در صورتی که رکوردی وجود نداشت 1 می‌گذاریم	int	DayCount

3. فکت تجمیعی خدمات (سرویس ها)

در این فکت اطلاعات هر سرویس به صورت تجمیعی ذخیره می‌شود.

Fact_ACC_Service		
description	Data type	field name
کلید بعد سرویس از ستون ServiceSK در جدول Dim_Service در انبار داده به دست می‌آید.	int	ServiceSK
کلید طبیعی سرویس از ستون ServiceID در جدول Dim_Service در انبار داده به دست می‌آید.	int	ServiceID
مجموع دفعات ارائه این سرویس در کل در جدول Fact_Daily_Service در انبار داده بر اساس ServiceSK گروپ بای انجام می‌دهیم و تعداد را محاسبه می‌کنیم.	int	TotalServiceCount

مجموع درآمد حاصل از این سرویس در کل در جدول Fact_Daily_Service در انبار داده بر اساس ServiceSK گروپ بای کرده و مجموع TotalRevenue را محاسبه میکنیم.	numeric(13,2)	TotalRevenue
مبلغی که بیمه‌ها بابت این سرویس در کل پرداخت کرده اند. در جدول Fact_Daily_Service بر اساس روز و ServiceSK گروپ بای و مجموع InsurancePaidAmount را محاسبه میکنیم.	numeric(13,2)	InsurancePaidAmount
میانگین روزانه تعداد بیماران یکتایی که این سرویس را دریافت کرده اند. در جدول Fact_Daily_Service در انبار داده بر اساس ServiceSK گروپ بای می‌کنیم و میانگین UniquePatientsCount را محاسبه میکنیم.	int	UniquePatientsAvg
میانگین روزانه تعداد پزشکان یکتایی که این سرویس را ارائه کرده اند در جدول Fact_Daily_Service در انبار داده بر اساس ServiceSK گروپ بای می‌کنیم و میانگین UniqueDoctorsCount را محاسبه میکنیم.	int	UniqueDoctorsAvg
مبلغ پایه سرویس (آخرین قیمت) از جدول Dim_Service انبار داده به دست می‌آید.	numeric(13,2)	UnitCost
میانگین مبلغ پایه سرویس از گذشته تا کنون ستون AvgUnitCost در جدول Fact_Daily_Service در انبار داده برای آخرین روز	numeric(13,2)	AvgUnitCost
ستون DayCount در جدول Fact_Daily_Service در انبار داده برای آخرین روز	int	DayCount

4. فکت لس بیمار-بیمه

در این فکت ارتباط بیمار و شرکت های بیمه ذخیره می‌شود.

Factless_Patient_Insurance		
description	Data type	field name
کلید بعد بیمار از ستون PatientID در جدول PatientInsuranceS در پایگاه داده منبع به دست می‌آید.	int	PatientID
کلید بعد بیمه از ستون InsuranceCoID در جدول PatientInsuranceS در پایگاه داده منبع به دست می‌آید.	int	InsuranceCoID

جداول مارت بیمار

جداول بعد

1. بعد بیماری

این بعد شامل اطلاعات مربوط به بیماری های مختلف است و از جدول Disease در پایگاه داده منبع به دست می آید.

Dim_Disease		
description	Data type	field name
کلید - شناسه یکتای بیماری که از جدول Disease به دست می آید	INT	Disease ID
کد بیماری؛ از جدول Disease به دست می آید	VARCHAR(10)	ICD10-Code
نام بیماری؛ از جدول Disease به دست می آید	VARCHAR(90)	DiseaseName
مشخص میکند که این بیماری مزمن است؛ از جدول Disease به دست می آید	BIT	IsCronic
توضیحات بیماری؛ از جدول Disease به دست می آید	TEXT	Description

2. نوع ویزیت

این بعد شامل نوع های مختلف ویزیت است. و از جدول VisitType در پایگاه داده منبع به دست می آید.

Dim_VisitType		
description	Data type	field name
کلید - شناسه یکتای نوع ویزیت	INT	VisitTypeID
نوع ویزیت (مثلا ... emergency, scheduled, follow-up, ...)	VARCHAR(60)	VisitTypeName

جداول فکت مارت بیمار

1. فکت تراکنشی ویزیت ها

به ازای هر فردی که به کلینیک مراجعه میکند و ویزیت برای او ثبت میشود، یک رکورد در این فکت قرار می گیرد. برای این کار از جدول Visits و جویین با جداول Patients، Admissions، PatientMedicalHistory، visitServices، Prescription و PrescriptionItems به دست می آید.

Fact_Transaction_Visit		
description	Data type	field name
کلید بعد بیمار؛ از ستون PatientID در جدول Visits در پایگاه داده منبع به دست می آید.	int	PatientID
کلید بعد کارکنان؛ شناسه پزشک ویزیت کننده از ستون DoctorStaffID در جدول Visits در پایگاه داده منبع به دست می آید.	int	StaffID
کلید بعد تاریخ از ستون VisitDateTime در جدول Visits به دست می آید.	int	DateKey
کلید بعد زمان از ستون VisitDateTime در جدول Visits به دست می آید.	int	TimeKey
کلید بعد نوع ویزیت از ستون VisitType در جدول Visits به دست می آید.	int	VisitTypeID
کلید بعد دپارتمان از جدول Visits و ستون DepartmentID استفاده میشود و با جویین روی بعد Dim_Department در انبار داده، ستون DepartmentSK در این قسمت قرار میگیرد.	int	DepartmentSK
کلید طبیعی دپارتمان از جدول Visits و ستون DepartmentID به دست می آید.	int	DepartmentID
کلید بعد بیماری با استفاده از ستون DiagnosisHistoryID جداول Visits و PatientMedicalHistory جویین می شوند و ستون DiseaseID به دست می آید. در صورتی که بیماری تشخیص داده نشود و DiagnosisHistoryID خالی باشد، کلید -1 قرار می گیرد.	int	DiseaseID
تعداد سرویس های سرپایی دریافت شده در این ویزیت	int	VisitServiceQuantity

از جمع ستون Quantity در جدول visitServices به دست می‌آید در صورتی که IsAdmissionService صفر باشد		
تعداد سرویس های بستری دریافت شده در این ویزیت از جمع ستون Quantity در جدول visitServices به دست می‌آید در صورتی که IsAdmissionService یک باشد	int	AdmissionServiceQuantity
تعداد سرویس های متمایز دریافت شده در این ویزیت از شمارش یکتای ServiceID در جدول visitServices به دست می‌آید	int	UniqueServiceCount
تعداد دارو های متمایز دریافت شده در این ویزیت جدول Visits با Prescription و با PrescriptionItems جوین می‌شوند و از شمارش یکتای DrugID در جدول PrescriptionItems به دست می‌آید	int	UniqueDrugCount
میانگین زمان لازم برای مصرف دارو ها از میانگین DurationDays در جدول PrescriptionItems به دست می‌آید	int	medicationDurationAvg
هزینه کل سرویس های استفاده شده از ضرب Quantity در جدول VisitServices و BaseCost در جدول Services به دست می‌آید.	numeric(13,2)	TotalCost

2. فکت دوره روزانه بیماران

در این فکت در هر روز یک رکورد برای هر بیمار ثبت می‌شود و وضعیت کلی بیمار در آن روز را مشخص می‌کند.

Fact_Daily_Patient		
description	Data type	field name
کلید بعد بیمار از ستون PatientID در جدول Dim_Patient در انبار داده به دست می‌آید.	int	PatientID
کلید تاریخ از ستون DateKey در جدول Dim_Date در انبار داده بدست می‌آید	int	DateKey
تعداد دفعات ویزیت در روز در جدول Fact_Transaction_Visit در انبار داده بر اساس روز و PatientID گروه بای انجام می‌دهیم و تعداد را محاسبه می‌کنیم.	int	VisitCount
تعداد سرویس های سرپایی دریافت شده در این روز	int	VisitServiceQuantity

در جدول Fact_Transaction_Visit در انبار داده بر اساس روز و PatientID گروه بای انجام می‌دهیم و جمع ستون VisitServiceQuantity را محاسبه می‌کنیم.		
تعداد سرویس های بستری دریافت شده در این روز در جدول Fact_Transaction_Visit در انبار داده بر اساس روز و PatientID گروه بای انجام می‌دهیم و جمع ستون AdmissionServiceQuantity را محاسبه می‌کنیم.	int	AdmissionServiceQuantity
هزینه کل سرویس های استفاده شده در این روز در جدول Fact_Transaction_Visit در انبار داده بر اساس روز و PatientID گروه بای کرده و مجموع TotalCost را محاسبه می‌کنیم.	int	TotalCost
چند روز از آخرین مراجعه میگذرد در همین جدول (Fact_Daily_Service) اگر VisitCount بزرگتر از 0 باشد صفر در غیر اینصورت رکورد روز قبل برای این بیمار را بدست آورده LastVisitDays + 1 را محاسبه می‌کنیم.	int	LastVisitDays
تعداد روز های بدون مراجعه در همین جدول (Fact_Daily_Service) NoVisitDays روز قبل برای این بیمار را بدست آورده: اگر VisitCount امروز بزرگتر از صفر بود NoVisitDays و در غیر اینصورت NoVisitDays + 1 قرار می‌دهیم.	int	NoVisitDays

3. فکت تجمعی بیماران

در این فکت اطلاعات هر بیمار به صورت تجمعی ذخیره می‌شود.

Fact_ACC_Patient		
description	Data type	field name
کلید بعد بیمار از ستون PatientID در جدول Dim_Patient در انبار داده به دست می‌آید.	int	PatientID
تعداد دفعات ویزیت کل در جدول Fact_Daily_Visit در انبار داده بر اساس PatientID گروه بای انجام می‌دهیم و جمع VisitCount را محاسبه می‌کنیم.	int	TotalVisitCount
تعداد سرویس های سرپایی دریافت شده در کل	int	VisitServiceQuantity

در جدول Fact_Daily_Visit در انبار داده بر اساس PatientID گروه بای انجام میدهم و جمع ستون VisitServiceQuantity را محاسبه میکنیم.		
تعداد سرویس های بستری دریافت شده در کل در جدول Fact_Daily_Visit در انبار داده بر اساس PatientID گروه بای انجام میدهم و جمع ستون VisitServiceQuantity را محاسبه میکنیم.	int	AdmissionServiceQuantity
هزینه کل سرویس های استفاده شده در جدول Fact_Daily_Visit در انبار داده بر اساس PatientID گروه بای کرده و مجموع TotalCost را محاسبه میکنیم.	int	TotalCost
چند روز از آخرین مراجعه میگذرد در جدول Fact_Daily_Visit ستون LastVisitDays را برای امروز محاسبه میکنیم.	int	LastVisitDays
تعداد روز های بدون مراجعه در جدول Fact_Daily_Visit ستون NoVisitDays را برای امروز محاسبه میکنیم.	int	NoVisitDays

4. فکت لس بیمار و بیماری

سوابق بیماری بیماران در این قسمت قرار میگیرد.

Factless_Patient_MedicalHistory		
description	Data type	field name
کلید بعد بیمار از ستون PatientID در جدول PatientMedicalHistory در پایگاه داده منبع به دست می آید.	int	PatientID
کلید بعد بیماری از ستون DiseaseID در جدول PatientMedicalHistory در پایگاه داده منبع به دست می آید.	int	DiseaseID
کلید بعد تاریخ، تاریخ تشخیص بیماری از ستون DiagnosisDate در جدول PatientMedicalHistory در پایگاه داده منبع به دست می آید.	int	DiagnosisDate