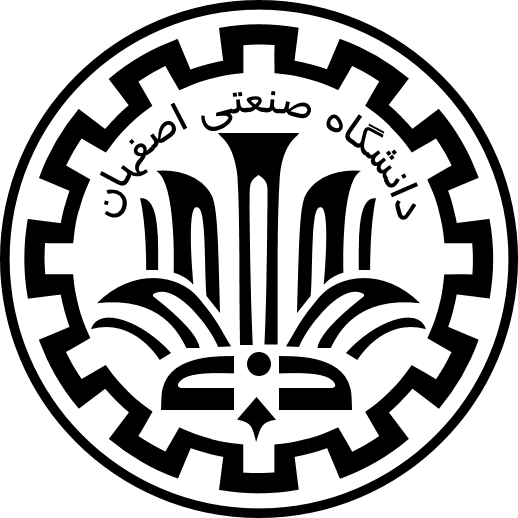
به نام خدا



دانشکده مهندسی برق و کامپیوتر

پایگاه داده دو

**طراحی انبار داده کلینیک درمانی**

اعضای تیم:

وانیا کرد – 40129043

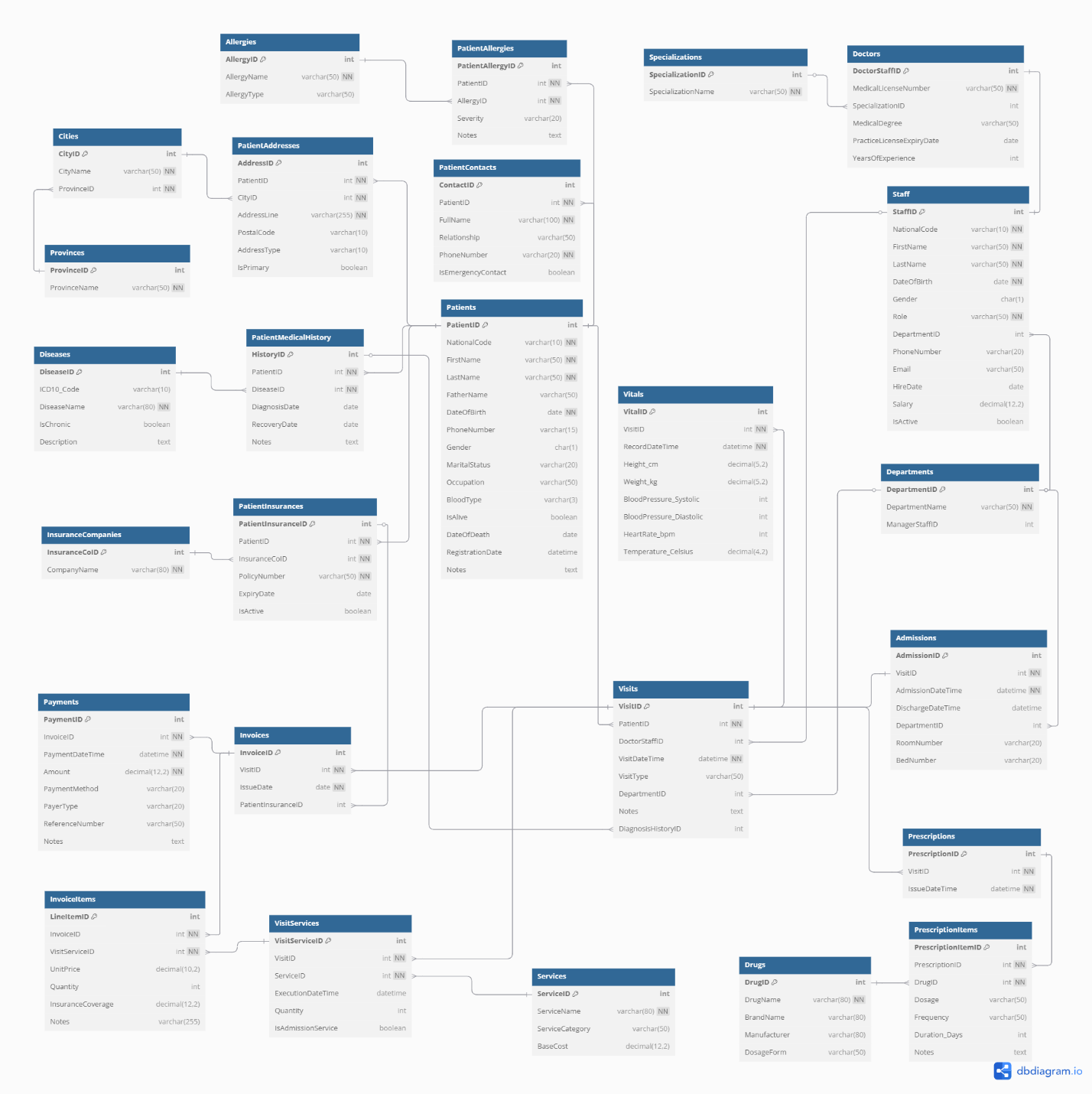
زهرا ابوطالبی – 40116943

استاد:

دکتر علیرضا بصیری

بهار 1404

**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 ارائه‌شده 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 نرم‌افزار مدیریتی کلینیک عمل کرده و با فراهم آوردن بستری برای ثبت دقیق اطلاعات، به اهداف کلیدی زیر دست می‌یابد:

* **افزایش کیفیت خدمات به بیمار:** از طریق دسترسی سریع به سوابق کامل پزشکی، حساسیت‌ها و تاریخچه درمان.
* **بهبود بهره‌وری عملیاتی:** با خودکارسازی فرآیندهای ثبت اطلاعات ویزیت، صدور نسخه و صورتحساب.
* **مدیریت مالی دقیق:** با ثبت تمام خدمات ارائه‌شده و پیگیری صورتحساب‌ها و پرداخت‌ها.
* **فراهم‌سازی داده‌های دقیق و منسجم:** به عنوان زیرساخت اصلی برای تحلیل‌های مدیریتی و گزارش‌گیری‌های آتی در انبار داده.

در ادامه، ساختار جداول، به همراه شرح کوتاهی از نقش و مهم‌ترین فیلدهای هرکدام آورده شده است.

**تشریح ساختار دیتابیس و جداول**

دیتابیس به صورت یک مدل رابطه‌ای (Relational) طراحی شده و جداول آن در چند بخش منطقی دسته‌بندی می‌شوند:

**بخش اول: مدیریت اطلاعات پایه بیماران**

این بخش شامل اطلاعات هویتی، تماس و بیمه‌ای 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.
* Provinces و Cities (استان‌ها و شهرها): جداول پایه برای مدیریت اطلاعات جغرافیایی.
* PatientAddresses (آدرس‌های بیمار): نگهداری آدرس‌های مختلف بیمار (مثلاً منزل، محل کار).
* InsuranceCompanies (شرکت‌های بیمه): لیست شرکت‌های بیمه طرف قرارداد با کلینیک.
* PatientInsurances (بیمه‌های بیمار): برای ثبت اطلاعات بیمه‌نامه هر بیمار و ارتباط آن با شرکت بیمه مربوطه.

**بخش دوم: سوابق بالینی و کاتالوگ‌های پزشکی**

این بخش به مدیریت تاریخچه پزشکی بیماران و اطلاعات مرجع پزشکی می‌پردازد.

* Allergies (حساسیت‌ها): یک کاتالوگ از انواع حساسیت‌های دارویی، غذایی و محیطی.
* PatientAllergies (حساسیت‌های بیمار): ثبت حساسیت‌های مشخص هر بیمار با ارجاع به جدول Allergies.
* Diseases (بیماری‌ها): کاتالوگ بیماری‌ها بر اساس کدهای استاندارد مانند ICD-10.
* PatientMedicalHistory (سوابق پزشکی بیمار): مهم‌ترین جدول این بخش که تاریخچه بیماری‌های هر فرد (مزمن 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 مثلا 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**

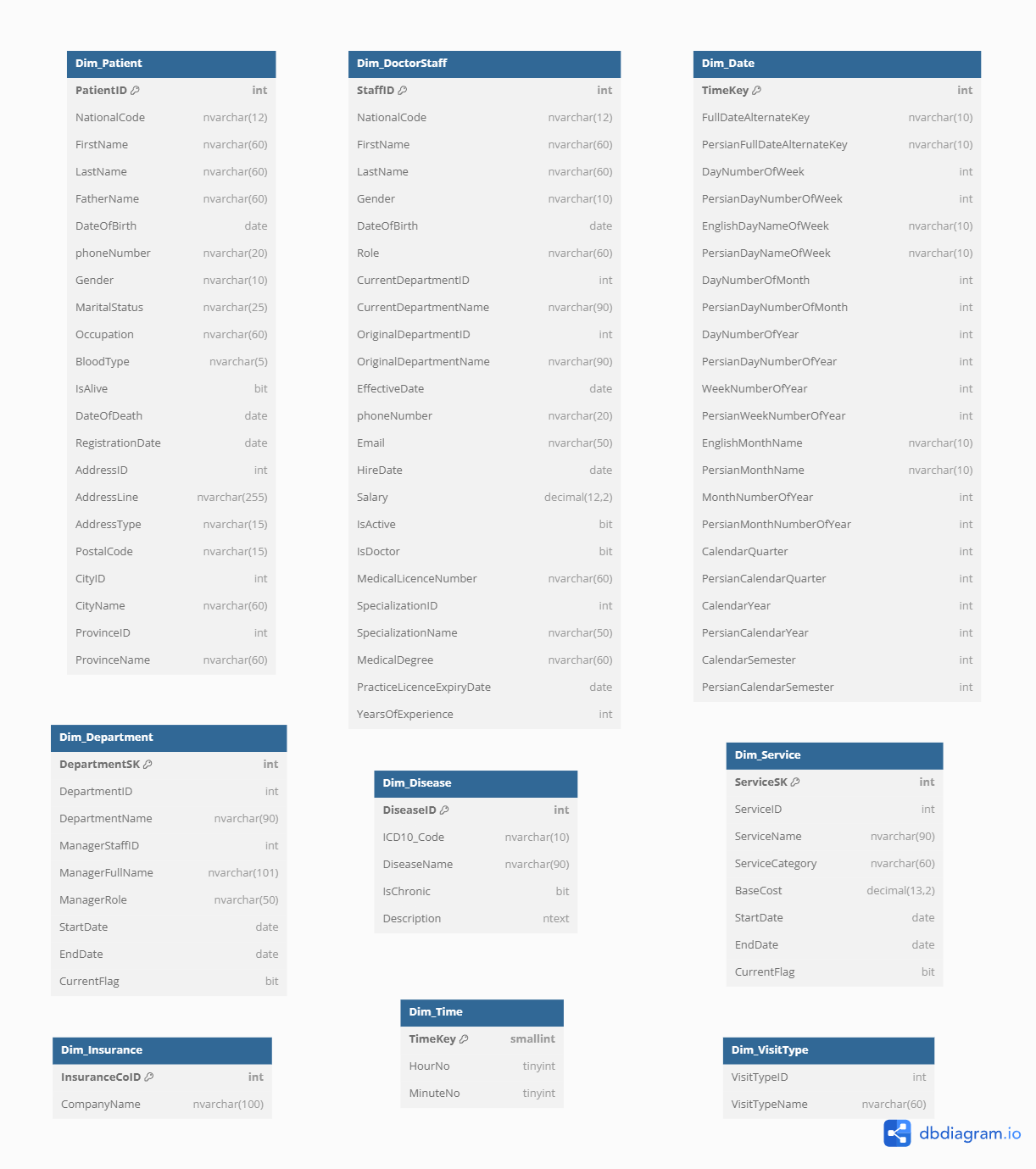
شامل اطلاعات تحلیلی در مورد رفتار بیماران، مراجعه های مکرر، گروه های سنی و بیماری ها

1. **Financial Data Mart**

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

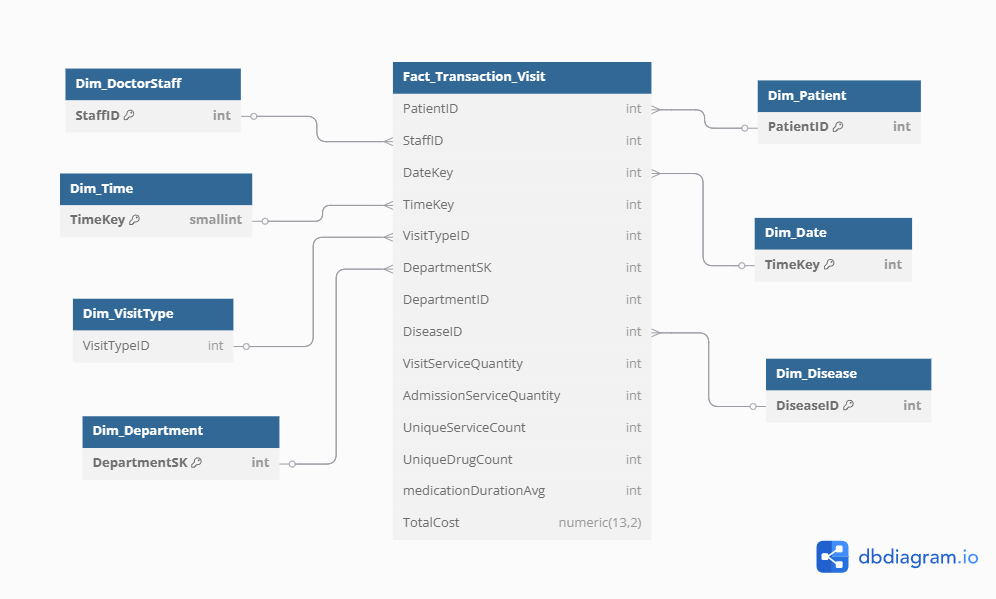
**انبار داده – مدل استار**

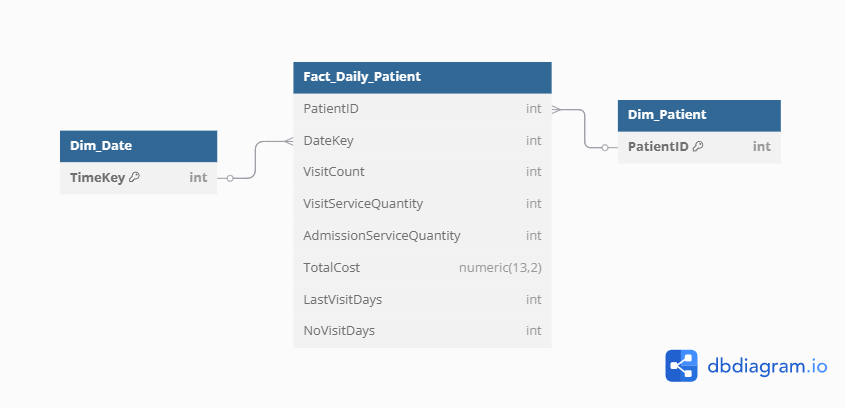
1. **Dimensions**

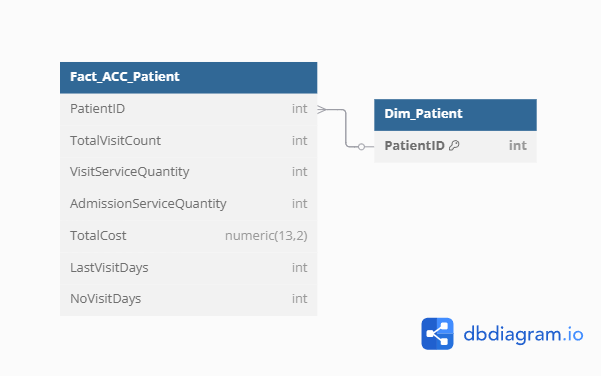
****

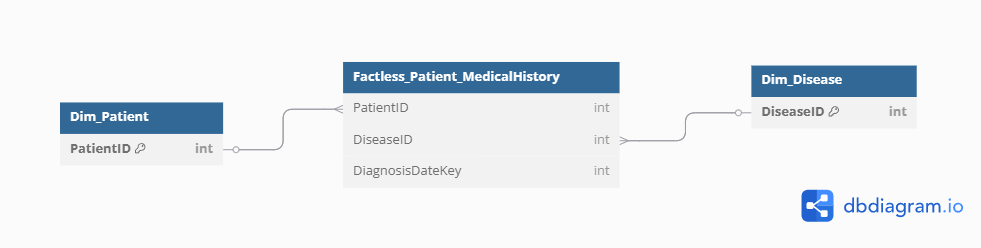
1. **Patient Data Mart**

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

****

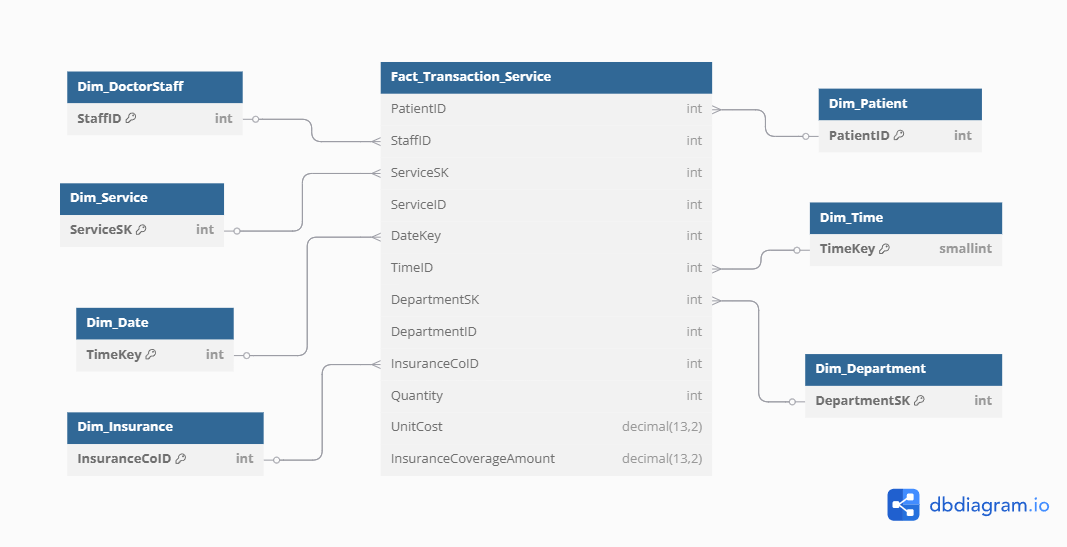


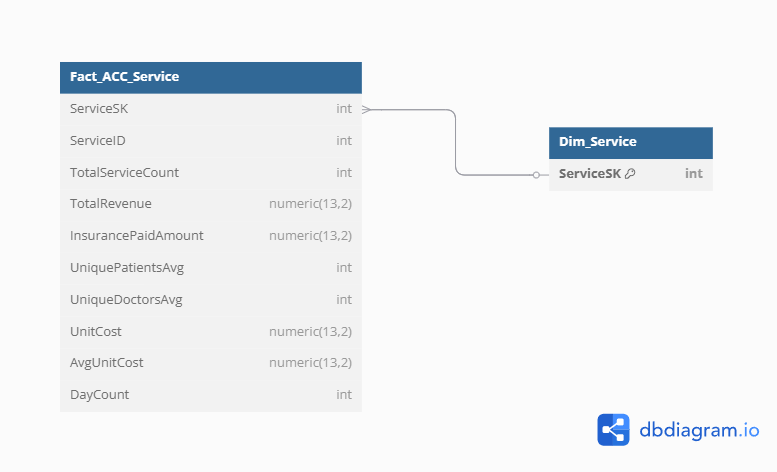
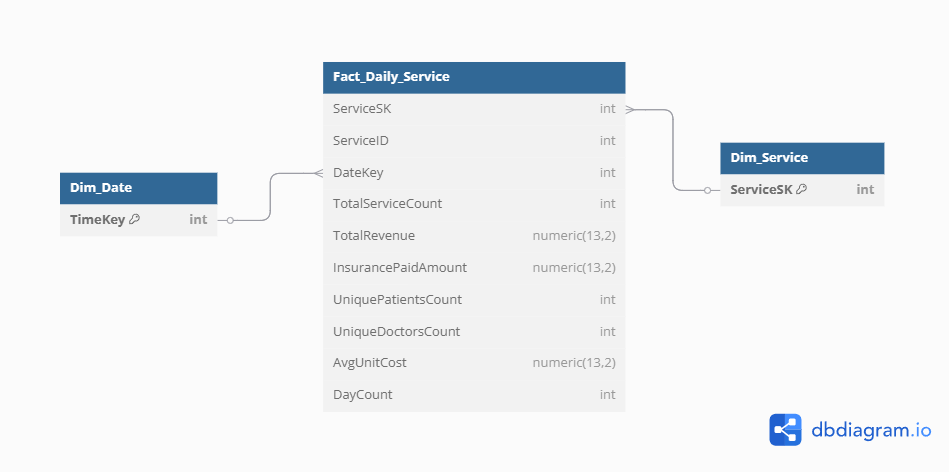


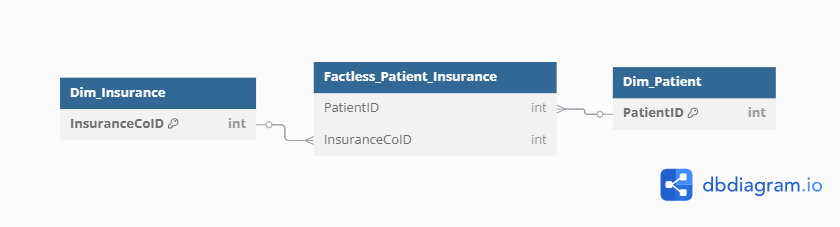
****

1. **Financial Data Mart**

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 |  |  |

1. **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 |

1. **بعد بیمار**

اطلاعات کلی بیمار در این بعد قرار دارد. لازم است جداول Patient، PatientAddresses، 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 می‌شود. | INT | PatientID |
| کد ملی بیمار؛ از ستون NationalCode در جدول Patient استخراج می‌شود. | VARCHAR(12) | NationalCode |
| نام بیمار؛ از ستون FirstName جدول Patient استخراج می‌شود. | VARCHAR(60) | FirstName |
| نام خانوادگی بیمار؛ از ستون LastName جدول Patient استخراج می‌شود. | VARCHAR(60) | LastName |
| نام پدر بیمار؛ از ستون FatherName جدول Patient استخراج می‌شود. | VARCHAR(60) | FatherName |
| تاریخ تولد بیمار؛ از ستون DateOfBirth جدول Patient استخراج می‌شود. | DATE | DateOfBirth |
| شماره تماس بیمار؛ از ستون PhoneNumber جدول Patient استخراج می‌شود. | VARCHAR(20) | phoneNumber |
| جنسیت بیمار (مانند Male یا Female)؛ از ستون Gender جدول Patient استخراج می‌شود. | VARCHAR(10) | Gender |
| وضعیت تأهل بیمار (مانند Single یا Married)؛ از ستون MaritalStatus جدول Patient استخراج می‌شود. | VARCHAR(25) | MaritalStatus |
| شغل بیمار؛ از ستون Occupation جدول Patient استخراج می‌شود. | VARCHAR(60) | Occupation |
| گروه خونی بیمار (مانند )؛ از ستون BloodType جدول Patient استخراج می‌شود. | VARCHAR(5) | BloodType |
| مشخص می‌کند که آیا بیمار زنده است یا خیر؛ از ستون IsAlive جدول Patient استخراج می‌شود. | BIT | IsAlive |
| در صورت فوت بیمار، تاریخ فوت او؛ از ستون DateOfDeath جدول Patient استخراج می‌شود. | DATE | DateOfDeath |
| تاریخ ثبت‌نام یا ایجاد پرونده بیمار در سامانه؛ از ستون RegistrationDate جدول PatientAddresses استخراج می‌شود. | DATE | RegistrationDate |
| شناسه آدرس مرتبط با بیمار؛ از ستون AddressID موجود در جدول PatientAddresses استخراج می‌شود. | INT | AddressID |
| آدرس کامل بیمار به صورت خطی؛ از ستون AddressLine جدول PatientAddresses استخراج می‌شود. | VARCHAR(255) | AddressLine |
| نوع آدرس (مانند منزل یا محل کار)؛ از ستون AddressType جدول PatientAddresses استخراج می‌شود. | VARCHAR(15) | AddressType |
| کد پستی آدرس بیمار؛ از ستون PostalCode جدول PatientAddresses استخراج می‌شود. | VARCHAR(15) | PostalCode |
| شناسه شهر مرتبط با آدرس؛ از ستون CityID جدول PatientAddresses استخراج می‌شود. | INT | CityID |
| نام شهر مربوط به آدرس بیمار؛ از ستون CityName جدول Cities استخراج می‌شود. | VARCHAR(60) | CityName |
| شناسه استان مربوط به شهر آدرس؛ از ستون ProvinceID جدول Cities استخراج می‌شود. | INT | ProvinceID |
| نام استان مربوط به شهر آدرس؛ از ستون ProvinceName جدول Provinces استخراج می‌شود. | VARCHAR(60) | ProvinceName |

1. **بعد کارکنان**

اطلاعات کلی کارکنان کلینیک از جمله پزشکان در این بعد قرار دارد. لازم است جداول Staff، Doctors، Departments و Specialization در پایگاه داده منبع بر روی فیلد های مناسب جوین شوند تا اطلاعات این جدول به دست بیاید.

در صورتی که فردی پزشک نباشد در فیلد های 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 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  SCD نوع 3 | INT | CurrentDepartmentID |
| نام دپارتمان؛ از ستون DepartmentName در جدول Departments به دست می‌آید، با استفاده از DepartmentID  SCD نوع 3 | VARCHAR(60) | CurrentDepartmentName |
| شناسه دپارتمانی که کارمند قبلا 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 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 دکتری عمومی، تخصص، فوق‌تخصص) 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 |

1. **بعد سرویس**

اطلاعات کلی سرویس های 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  SCD نوع 2 | 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 |

1. **بعد دپارتمان**

اطلاعات هر بخش (دپارتمان) در این بعد قرار میگیرد. اطلاعات این بخش از جوین 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  SCD نوع 2 | 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 |

1. **بعد بیمه**

اطلاعات مربوط به شرکت های بیمه در این بعد قرار می‌گیرد. اطلاعات این بعد از جدول InsuranceCompanies در پایگاه داده منبع به دست می‌آید.

|  |  |  |
| --- | --- | --- |
| Dim\_Insurance | | |
| description | **Data type** | **field name** |
| شناسه شرکت بیمه؛ از فیلد InsuranceCoID موجود در جدول InsuranceCompanies به دست می‌آید. | INT | InsuranceCoID |
| نام شرکت بیمه؛ از فیلد CompanyName موجود در جدول InsuranceCompanies به دست می‌آید. | VARCHAR(100) | CompanyName |

**جداول مارت مالی**

* **جداول فکت**

1. **فکت تراکنشی سرویس**

به ازای هر سرویس که یک بیمار دریافت می‌کند و صورتحساب برای آن صادر می‌شود، یک رکورد در این فکت قرار می‌گیرد. برای این کار از جدول invoiceItems استفاده میشود و برای به دست آوردن اطلاعات مربوطه این جدول با جداول visitServises، Visits، Patients، Invoices و PatientInsurances موجود در پایگاه داده منبع جوین می‌شود.

|  |  |  |
| --- | --- | --- |
| Fact\_Transaction\_Service | | |
| description | **Data type** | **field name** |
| کلید بعد بیمار؛  جدول invoiceItems با visitServises و دوباره با Visits جوین میشود. از PatientID در جدول Visits به دست می‌آید. | int | PatientID |
| کلید بعد کارکنان؛  مشابه کلید بعد بیمار از جدول Visits و ستون DoctorStaffID به دست می‌آید. | int | StaffID |
| کلید بعد سرویس؛  از ستون ServiceID در جدول visitServices استفاده میشود و با جوین روی بعد Dim\_service در انبار داده ServiceSK این جدول در این قسمت قرار می‌گیرد. | int | ServiceSK |
| کلید طبیعی سرویس  از ستون ServiceID در جدول visitServices به دست می‌آید | int | ServiceID |
| کلید بعد تاریخ  از ستون ExecutionDateTime در جدول visitServices به دست می‌آید. | int | DateKey |
| کلید بعد زمان  از ستون ExecutionDateTime در جدول visitServices به دست می‌آید. | int | TimeKey |
| کلید بعد دپارتمان  از جدول Visits و ستون DoctorStaffID استفاده میشود و با جوین 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 می‌‌آید. | int | DepartmentID |
| کلید بعد بیمه  جدول InvoiceItems 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 invoiceItems به دست می‌آید | int | Quantity |
| قیمت سرویس به ازای هر واحد  از ستون UnitCost جدول invoiceItems به دست می‌آید | numeric(13,2) | UnitCost |
| مبلغ پوشش داده شده از سمت بیمه  از ستون InsuranceCoverageAmount جدول invoiceItems به دست می‌آید. | numeric(13,2) | InsuranceCoverageAmount |

1. **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 UnitCost × Quantityمحاسبه میکنیم. | numeric(13,2) | TotalRevenue |
| مبلغی که بیمه‌ها بابت این سرویس در این روز پرداخت کردند.  در جدول Fact\_Transaction\_Service بر اساس روز و ServiceSK گروپ بای و مجموع InsuranceCoverageAmount را محاسبه میکنیم. | numeric(13,2) | InsurancePaidAmount |
| 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 |

1. **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 اخرین روز | numeric(13,2) | AvgUnitCost |
| ستون DayCount در جدول Fact\_Daily\_Service در انبار داده برای اخرین روز | int | DayCount |

1. **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 می‌آید.

|  |  |  |
| --- | --- | --- |
| Dim\_ Disease | | |
| description | **Data type** | **field name** |
| کلید – شناسه یکتای بیماری که از جدول Disease به دست می‌آید | INT | Disease ID |
| کد بیماری؛ از جدول Disease به دست می‌آید | VARCHAR(10) | ICD10-Code |
| نام بیماری؛ از جدول Disease به دست می‌آید | VARCHAR(90) | DiseaseName |
| مشخص میکند که این بیماری مزمن است؛ از جدول Disease به دست می‌آید | BIT | IsCronic |
| توضیحات بیماری؛ از جدول Disease به دست می‌آید | TEXT | Desciption |

1. **نوع ویزیت**

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

|  |  |  |
| --- | --- | --- |
| 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 می‌گیرد. برای این کار از جدول Visits و جوین با جداول Patients، Admissions، PatientMedicalHistory، visitServises، Prescription و PrescriptionItems به دست می‌آید.

|  |  |  |
| --- | --- | --- |
| Fact\_Transaction\_Visit | | |
| description | **Data type** | **field name** |
| کلید بعد بیمار؛  از ستون PatientID در جدول Visits در پایگاه داده منبع به دست می‌آید. | int | PatientID |
| کلید بعد کارکنان؛ شناسه پزشک ویزیت کننده  از ستون DoctorStaffID در جدول Visits در پایگاه داده منبع به دست می‌آید. | int | StaffID |
| کلید بعد تاریخ  از ستون VisitDateTime در جدول Visits به دست می‌آید. | int | DateKey |
| کلید بعد زمان  از ستون VisitDateTime در جدول Visits به دست می‌آید. | int | TimeKey |
| کلید بعد نوع ویزیت  از ستون VisitType در جدول Visits به دست می‌آید. | int | VisitTypeID |
| کلید بعد دپارتمان  از جدول Visits و ستون DepartmentID استفاده میشود و با جوین 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 می‌‌آید. | int | DepartmentID |
| کلید بعد بیماری  با استفاده از ستون DiagnosisHistoryID جداول Visits و PatientMedicalHistory جوین می‌شوند و ستون DiseaseID به دست می‌‌آید.  در صورتی که بیماری تشخیص داده نشود و DiagnosisHistoryID خالی باشد، کلید -1 قرار می‌گیرد. | int | DiseaseID |
| تعداد سرویس های سرپایی دریافت شده در این ویزیت  از جمع ستون Quantity در جدول visitServises به دست می‌آید در صورتی که IsAdmissionService صفر باشد | int | VisitServiceQuantity |
| تعداد سرویس های بستری دریافت شده در این ویزیت  از جمع ستون Quantity در جدول visitServises 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 visitServises 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 VisitServises و BaseCost در جدول Services به دست می‌آید. | numeric(13,2) | TotalCost |

1. **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 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 گروپ بای انجام میدهیم و تعداد را محاسبه میکنیم. | int | VisitCount |
| تعداد سرویس های سرپایی دریافت شده در این روز  در جدول Fact\_Transaction\_Visit در انبار داده بر اساس روز و PatientID گروپ بای انجام میدهیم و جمع ستون VisitServiceQuantity را محاسبه میکنیم. | int | VisitServiceQuantity |
| تعداد سرویس های بستری دریافت شده در این روز  در جدول Fact\_Transaction\_Visit در انبار داده بر اساس روز و PatientID گروپ 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 گروپ 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  درغیر 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 |

1. **فکت تجمعی بیماران**

در این فکت اطلاعات هر بیمار به صورت تجمیعی 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 گروپ بای انجام میدهیم و جمع VisitCount را محاسبه میکنیم. | int | TotalVisitCount |
| تعداد سرویس های سرپایی دریافت شده در کل  در جدول Fact\_Daily\_Visit در انبار داده بر اساس PatientID گروپ بای انجام میدهیم و جمع ستون VisitServiceQuantity را محاسبه میکنیم. | int | VisitServiceQuantity |
| تعداد سرویس های بستری دریافت شده در کل  در جدول Fact\_Daily\_Visit در انبار داده بر اساس PatientID گروپ بای انجام میدهیم 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 گروپ 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 |

1. **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 می‌آید. | int | PatientID |
| کلید بعد بیماری از ستون DiseaseID در جدول PatientMedicalHistory در پایگاه داده منبع به دست می‌آید. | int | DiseaseID |
| کلید بعد تاریخ، تاریخ تشخیص بیماری  از ستون DiagnosisDate در جدول PatientMedicalHistory در پایگاه داده منبع به دست می‌آید. | int | DiagnosisDate |