
Proposed Data Warehouse Schema

We'll use a **Star Schema** for simplicity and query performance.

Fact Tables (storing measurable events):

- **Fact_Appointments**
- **Fact_MedicalRecords**
- **Fact_Billing**

Dimension Tables (describing entities):

- **Dim_Patients**
- **Dim_Doctors**
- **Dim_Date** (useful for time-based analysis)

Fact Tables Design

Fact_Appointments

Tracks all patient appointments, including status.

```
CREATE TABLE Fact_Appointments (
    AppointmentID INT PRIMARY KEY,
    PatientID INT,
    DoctorID INT,
    DateID INT,
    Status VARCHAR(20),
    FOREIGN KEY (PatientID) REFERENCES Dim_Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Dim_Doctors(DoctorID),
    FOREIGN KEY (DateID) REFERENCES Dim_Date(DateID)
);
```

Fact_MedicalRecords

Stores patient visits, diagnoses, and prescriptions.

```
CREATE TABLE Fact_MedicalRecords (
    RecordID INT PRIMARY KEY,
    PatientID INT,
    DoctorID INT,
    DateID INT,
    Diagnosis TEXT,
    Prescription TEXT,
    FOREIGN KEY (PatientID) REFERENCES Dim_Patients(PatientID),
    FOREIGN KEY (DoctorID) REFERENCES Dim_Doctors(DoctorID),
    FOREIGN KEY (DateID) REFERENCES Dim_Date(DateID)
);
```

Fact_Billing

Captures billing transactions.

```
CREATE TABLE Fact_Billing (
    BillID INT PRIMARY KEY,
    PatientID INT,
    DateID INT,
    Amount DECIMAL(10, 2),
    PaymentStatus VARCHAR(20),
    FOREIGN KEY (PatientID) REFERENCES Dim_Patients(PatientID),
    FOREIGN KEY (DateID) REFERENCES Dim_Date(DateID)
);
```

Dimension Tables Design

Dim_Doctors

Stores doctor details.

```
CREATE TABLE Dim_Doctors (
    DoctorID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    FullName VARCHAR(101) GENERATED ALWAYS AS (CONCAT(FirstName, ' ', LastName)),
    Specialization VARCHAR(100),
    ContactNumber VARCHAR(15),
    Email VARCHAR(100)
);
```

Dim_Patients

Contains patient demographic data.

```
CREATE TABLE Dim_Patients (
   PatientID INT PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   FullName VARCHAR(101) GENERATED ALWAYS AS (CONCAT(FirstName, ' ', LastName)),
   DOB DATE,
   Gender VARCHAR(10),
   ContactNumber VARCHAR(15),
   Email VARCHAR(100),
   Address TEXT
);
```

Dim_Date

Standard date dimension to enable time-based analysis.

```
CREATE TABLE Dim_Date (
   DateID INT PRIMARY KEY,
   Date DATE,
   Day INT,
   Month INT,
   Year INT,
   Quarter INT,
   DayOfWeek VARCHAR(10)
);
```

Data Transformation (ETL/ELT) Process

1. **Extraction:**
   - Pull data from the OLTP system (e.g., Patients, Appointments, Billing).
2. **Transformation:**
   - Cleanse and standardize data (e.g., format names, normalize phone numbers).
   - Create surrogate keys for dimensions (if needed).

    - Derive useful fields (like full names, age from DOB).

3. **Loading:**
    - Load dimension tables first (slowly changing dimensions as needed).
    - Load fact tables afterward, linking to dimension tables using foreign keys.

Analytical Use Cases

- **Patient Appointment Analytics:** How many appointments were completed, canceled, or rescheduled over time?
- **Doctor Performance Metrics:** How many patients does each doctor see, and what are the most common diagnoses?
- **Billing Insights:** What's the total revenue, and how much is still overdue?
- **Patient History Tracking:** View a patient's complete medical history, including visits, diagnoses, prescriptions, and payments.

Scalability & Future Enhancements

- **Scalability:** Cloud data warehouses like Snowflake or BigQuery handle growing data volumes.
- **Security:** Implement row-level security for sensitive patient data.
- **Extensions:** Add dimensions like **Hospital Locations** or **Insurance Providers**.