Project Overview – Part B1: Denormalized Star Schema Design

After completing Part A of the project—where I designed a fully normalized 3NF logical model for a heart attack risk prediction database with over ten relational tables, including a many-to-many PatientRiskFactor relationship—I moved on to Part B with the aim of optimizing the model for analytics.

In this stage, I developed a denormalized star schema to facilitate efficient analytical querying and business intelligence reporting. This approach supports fast data retrieval and simplifies the process of generating insights from patient health data.

1. Rationale Behind My Star Schema Design

Simplified Reporting & Analytics

The star schema places all quantitative data in a centralized fact table and organizes descriptive attributes into surrounding dimension tables. This structure reduces join complexity, making it easier to query and report on the data.

Enhanced Performance

By denormalizing the structure, the model reduces the number of joins needed for common OLAP queries. This boosts performance for read-heavy operations typically used in dashboards and reporting.

Clear Separation of Concerns

I've separated numeric metrics (like cholesterol levels and BMI) from contextual details (like patient demographics and diet), which makes drill-downs and aggregations more intuitive and powerful.

2. Star Schema Components Overview

Fact Table: FactHeartAttackRisk

This central table stores all measurable data points related to a patient's heart health.

Foreign Keys: PatientID, CountryID, DietID, and (optionally) RiskFactorID reference their respective dimensions.

Measures: Includes fields like CholesterolLevel, SystolicBP, DiastolicBP, BMI, Triglycerides, and HeartAttackRiskScore.

Optional: TimeID can reference a DimTime table to support historical analysis.

DimPatient

Attributes: Stores age, sex, stress level, and additional patient details.

Design: Uses VARCHAR(n) for textual values to ensure flexibility and portability.

DimCountry

Attributes: Captures country names and links to continent and hemisphere references from the original model.

Purpose: Encodes geographical hierarchy for drill-down analytics.

DimDiet

Attributes: Categorizes diet as "Healthy", "Average", or "Unhealthy".

Design: Ensures standardized input using foreign key constraints.

DimRiskFactor

Attributes: Lists individual risk factors (e.g., Smoking, Hypertension, Diabetes).

Structure: Represents a flattened many-to-many relationship; a bridge table may be used if required.

(Optional) DimTime

Attributes: Includes date, month, quarter, and year fields.

Purpose: Assists in analyzing trends over time (e.g., rising cholesterol levels annually).

All dimension tables feature clear primary keys and meaningful, query-friendly data types.

3. Relationships & Schema Marking

The schema follows a classic star layout:

Central Fact Table: FactHeartAttackRisk.

Surrounding Dimension Tables: DimPatient, DimCountry, DimDiet, DimRiskFactor, and optionally DimTime.

Each dimension is connected to the fact table through clearly defined foreign keys. The schema notations precisely reflect the cardinality and direction of these relationships.

4. Addressing Feedback from Part A

Foreign Key Normalization

In Part A, descriptive strings like CountryName and DietType were stored directly in the Patient table. In this version, I replaced them with foreign key references, improving normalization and data integrity.

Use of VARCHAR(n) Types

All textual fields are now explicitly defined using VARCHAR(n) to avoid unbounded strings and enhance schema compatibility.

Preserved Constraints

Despite the denormalization, I retained critical constraints (e.g., non-negative values for numeric fields and limited categories for diet types) to maintain data quality and enforce business rules.

Conclusion

This denormalized star schema represents a robust analytical foundation for the heart attack risk prediction system. It supports efficient OLAP operations, enables rapid aggregation, and remains consistent with the normalized model from Part A. I have carefully addressed prior feedback, enforced data integrity, and designed the schema to meet real-world analytical needs.