**Steps to Create a Dimensional Data Model**

**1. Understand Business Objectives**

* Determine the metrics (facts) and dimensions that are most relevant to the analysis (e.g., predicting diabetes, analyzing health factors, etc.).
* Key question: What analyses or reports will this model support?

**2. Identify Facts**

Facts are numerical measures that are the focus of analysis. In the context of the dataset:

* **Facts**:
  + Whether an individual has diabetes (diabetes column, as a binary fact).
  + Other metrics such as weight, BMI, and pregnancies.

**3. Identify Dimensions**

Dimensions are descriptive attributes that provide context for facts. These often correspond to categorical variables. Dimensions might include:

* **Person Dimension**: Gender, age group, diet type.
* **Lifestyle Dimension**: Physical activity level, stress level, social media usage.
* **Health Metrics Dimension**: BMI categories, diabetes pedigree function.
* **Time Dimension** (if applicable): Could be derived if the data includes timestamps or other temporal information.

**4. Design Fact Table**

The fact table typically contains:

* **Keys to Dimension Tables**: Foreign keys referencing dimensions.
* **Measures/Attributes**: The numerical facts being analyzed (e.g., diabetes status, weight, BMI).

**Example Fact Table:**

| **Person\_Key** | **Lifestyle\_Key** | **Health\_Metrics\_Key** | **Diabetes** |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 1 |

**5. Design Dimension Tables**

Dimension tables contain descriptive attributes. They are typically "denormalized" for faster querying.

**Examples**:

1. **Person Dimension**:

| **Person\_Key** | **Gender** | **Age\_Group** | **Diet\_Type** |
| --- | --- | --- | --- |
| 1 | Male | 40-50 | Vegan |

1. **Lifestyle Dimension**:

| **Lifestyle\_Key** | **Physical\_Activity\_Level** | **Stress\_Level** | **Social\_Media\_Usage** |
| --- | --- | --- | --- |
| 1 | Lightly Active | Low | Occasionally |

1. **Health Metrics Dimension**:

| **Health\_Metrics\_Key** | **BMI\_Category** | **Diabetes\_Pedigree\_Function** |
| --- | --- | --- |
| 1 | Overweight | 0.65 |

**6. Define a Schema**

The dimensional model can be structured in one of the following schemas:

* **Star Schema**: A central fact table is surrounded by dimension tables.
* **Snowflake Schema**: Dimension tables are normalized into sub-dimensions.
* **Galaxy Schema**: Multiple fact tables share dimension tables.

**7. Populate the Model**

* Use SQL or ETL tools to transform and load the cleaned dataset into the dimensional model.
* Create surrogate keys for dimensions to replace natural keys (e.g., Person\_Key).

**8. Optimize for Queries**

* **Indexing**: Add indexes on foreign keys in the fact table for faster joins.
* **Pre-Aggregations**: Precompute common aggregations (e.g., average BMI by diet type).

Would you like an example implementation of creating fact and dimension tables from the current dataset? If so, I can outline how to extract and structure specific dimensions and the fact table.

4o

**You said:**

Show SQL for creating star schema.

**ChatGPT said:**

ChatGPT

Here’s an example of SQL to create a **Star Schema** for a dimensional data model from the cleaned dataset. This schema includes a central fact table (fact\_diabetes) and related dimension tables.

**1. Fact Table: fact\_diabetes**

The fact table contains keys referencing dimension tables and measures for analysis.

sql

Copy code

CREATE TABLE fact\_diabetes (

fact\_id INT AUTO\_INCREMENT PRIMARY KEY,

person\_key INT,

lifestyle\_key INT,

health\_metrics\_key INT,

diabetes\_status TINYINT, -- 0 for no diabetes, 1 for diabetes

weight FLOAT,

BMI FLOAT,

pregnancies INT,

FOREIGN KEY (person\_key) REFERENCES dim\_person(person\_key),

FOREIGN KEY (lifestyle\_key) REFERENCES dim\_lifestyle(lifestyle\_key),

FOREIGN KEY (health\_metrics\_key) REFERENCES dim\_health\_metrics(health\_metrics\_key)

);

**2. Dimension Table: dim\_person**

The dim\_person table provides demographic details.

sql

Copy code

CREATE TABLE dim\_person (

person\_key INT AUTO\_INCREMENT PRIMARY KEY,

gender VARCHAR(10),

age\_group VARCHAR(20), -- e.g., "20-30", "30-40"

diet\_type VARCHAR(50)

);

**3. Dimension Table: dim\_lifestyle**

The dim\_lifestyle table provides lifestyle-related attributes.

sql

Copy code

CREATE TABLE dim\_lifestyle (

lifestyle\_key INT AUTO\_INCREMENT PRIMARY KEY,

physical\_activity\_level VARCHAR(50),

stress\_level VARCHAR(50),

social\_media\_usage VARCHAR(50)

);

**4. Dimension Table: dim\_health\_metrics**

The dim\_health\_metrics table provides health-related metrics.

sql

Copy code

CREATE TABLE dim\_health\_metrics (

health\_metrics\_key INT AUTO\_INCREMENT PRIMARY KEY,

BMI\_category VARCHAR(20), -- e.g., "Underweight", "Normal", "Overweight", "Obese"

diabetes\_pedigree\_function FLOAT

);

**5. Insert Data into Dimension Tables**

**Example: dim\_person**

sql

Copy code

INSERT INTO dim\_person (gender, age\_group, diet\_type)

VALUES

('Male', '30-40', 'Vegetarian'),

('Female', '40-50', 'Vegan'),

('Male', '50-60', 'Mediterranean');

**Example: dim\_lifestyle**

sql

Copy code

INSERT INTO dim\_lifestyle (physical\_activity\_level, stress\_level, social\_media\_usage)

VALUES

('Sedentary', 'Low', 'Occasionally'),

('Lightly Active', 'Moderate', 'Never'),

('Very Active', 'High', 'Frequently');

**Example: dim\_health\_metrics**

sql

Copy code

INSERT INTO dim\_health\_metrics (BMI\_category, diabetes\_pedigree\_function)

VALUES

('Normal', 0.55),

('Overweight', 0.65),

('Obese', 0.75);

**6. Insert Data into Fact Table**

Assume the dimension table keys are generated based on specific mappings.

sql

Copy code

INSERT INTO fact\_diabetes (person\_key, lifestyle\_key, health\_metrics\_key, diabetes\_status, weight, BMI, pregnancies)

VALUES

(1, 1, 1, 1, 180.5, 29.4, 2),

(2, 2, 2, 0, 145.3, 24.8, 0),

(3, 3, 3, 1, 200.1, 32.5, 1);

**Querying the Star Schema**

You can now perform analytical queries using joins on the schema.

**Example: Average BMI by Age Group**

sql

Copy code

SELECT dp.age\_group, AVG(fd.BMI) AS avg\_bmi

FROM fact\_diabetes fd

JOIN dim\_person dp ON fd.person\_key = dp.person\_key

GROUP BY dp.age\_group;

**Example: Diabetes Distribution by Physical Activity Level**

sql

Copy code

SELECT dl.physical\_activity\_level, COUNT(fd.diabetes\_status) AS total\_cases

FROM fact\_diabetes fd

JOIN dim\_lifestyle dl ON fd.lifestyle\_key = dl.lifestyle\_key

WHERE fd.diabetes\_status = 1

GROUP BY dl.physical\_activity\_level;