**DDS**

**Dương Khánh An - 20120424**

1. **Các Measure có cần lưu ở DDS không?**

* **DDS hiện tại** lưu mean\_aqi và std\_aqi, nhưng:
  + mean\_aqi = min\_aqi = max\_aqi tại mức chi tiết thấp nhất.
  + std\_aqi = 0 vì không có sự tổng hợp.
* **Giải pháp**:
  + **Loại bỏ** mean\_aqi và std\_aqi khỏi DDS.
  + **Giữ lại** các Measure hỗ trợ: sum\_aqi, sum\_squares\_aqi, count\_aqi.

1. **Có cần thêm Measure hỗ trợ nào không?**

* Đã đủ các Measure cần thiết để tính:
  + Trung bình: mean\_aqi = sum\_aqi / count\_aqi.
  + Độ lệch chuẩn:
* Ko cần thêm Measure hỗ trợ.

1. **Có nên lưu dữ liệu tổng hợp thay vì mức chi tiết?**

* **Không**:
  + Mức chi tiết thấp nhất cho phép phục vụ tất cả các truy vấn, từ chi tiết đến tổng hợp.
  + Tổng hợp dữ liệu sẽ được thực hiện trên **Cube SSAS**.

**4. Làm gì với tổ hợp Dimension không tồn tại?**

* Một số tổ hợp (county\_id, category\_id, parameter\_id) không có dữ liệu thực tế.
* Giải pháp khả thi
  + **Sparse Storage**: Chỉ lưu các tổ hợp thực sự có dữ liệu trong bảng Fact.
  + **Cube Handling**: Sử dụng Null Handling trên Cube để ẩn các tổ hợp không tồn tại

**Gợi ý cải tiến DDS**

**1: Loại bỏ Measure dư thừa**

ALTER TABLE AQI\_Fact

DROP COLUMN mean\_aqi,

DROP COLUMN std\_aqi;

**2: Sử dụng Sparse Storage**

* Thay vì join full các Dimension, chỉ lưu các tổ hợp có dữ liệu từ các file đầu vào.
* **SQL để lọc dữ liệu thực tế**:

INSERT INTO AQI\_Fact (county\_id, date\_id, category\_id, parameter\_id, sum\_aqi, sum\_squares\_aqi, count\_aqi, count\_day, min\_aqi, max\_aqi)

SELECT

c.county\_id,

d.date\_id,

cat.category\_id,

p.parameter\_id,

SUM(a.AQI) AS sum\_aqi,

SUM(POWER(a.AQI, 2)) AS sum\_squares\_aqi,

COUNT(a.AQI) AS count\_aqi,

COUNT(DISTINCT a.Date) AS count\_day,

MIN(a.AQI) AS min\_aqi,

MAX(a.AQI) AS max\_aqi

FROM raw\_aqi\_data a

JOIN Dim\_County c ON a.county\_name = c.county\_name

JOIN Dim\_Date d ON a.Date = d.date

JOIN Dim\_Category cat ON a.Category = cat.category\_name

JOIN Dim\_DefiningParameter p ON a.Defining\_Parameter = p.parameter\_name

GROUP BY c.county\_id, d.date\_id, cat.category\_id, p.parameter\_id;

**3: Partitioning để tăng hiệu năng**

CREATE PARTITION FUNCTION pfYear (INT)

AS RANGE LEFT FOR VALUES (2021, 2022, 2023);

CREATE PARTITION SCHEME psYear

AS PARTITION pfYear ALL TO ([PRIMARY]);

CREATE TABLE AQI\_Fact\_Partitioned (

fact\_id INT IDENTITY(1,1) PRIMARY KEY,

county\_id INT NOT NULL,

date\_id INT NOT NULL,

category\_id INT NOT NULL,

parameter\_id INT NOT NULL,

sum\_aqi INT NOT NULL,

sum\_squares\_aqi FLOAT NOT NULL,

count\_aqi INT NOT NULL,

count\_day INT NOT NULL,

min\_aqi INT NOT NULL,

max\_aqi INT NOT NULL

) ON psYear(date\_id);

**Tóm tắt**

1. **DDS hiện tại**: Đáp ứng tốt, nhưng cần loại bỏ mean\_aqi và std\_aqi.
2. **Cải tiến đề xuất**:
   * Loại bỏ Measure dư thừa.
   * Chỉ lưu dữ liệu thực tế bằng Sparse Storage.
   * Partition bảng Fact để cải thiện hiệu năng.
3. **Dữ liệu mức chi tiết**: Đủ để phục vụ mọi truy vấn khi kết hợp với Cube SSAS.

**Trần Thái Vỹ - 20120632**

Với thiết kế DDS ở mức chi tiết thấp nhất, việc lưu các measure như std\_aqi trực tiếp trong bảng Fact không mang lại giá trị thực sự vì tính chất của dữ liệu ở mức này không cho phép tổng hợp. Ở mức này std\_aqi sẽ luôn bằng 0 vì chỉ có một điểm dữ liệu duy nhất trong mỗi nhóm, do đó k có sự phân tán.

### ****Xử lý Measure trên Cube SSAS****

Trong trường hợp này, ta có thể ưu tiên việc lưu các measure hỗ trợ tổng hợp trên Cube SSAS thay vì lưu trực tiếp các kết quả như mean\_aqi hoặc std\_aqi trong DDS. Các measure hỗ trợ như sum\_aqi, sum\_squares\_aqi, và count\_aqi sẽ cung cấp những thông tin cần thiết để có thể tính toán mean\_aqi và std\_aqi trên Cube SSAS với mức tổng hợp cao hơn.

### ****Vài lợi ích của việc lưu các Measure hỗ trợ:****

1. sum\_aqi: Tổng của AQI giúp tính toán trung bình (mean\_aqi) ở mức tổng hợp cao hơn.
2. sum\_squares\_aqi: Tổng bình phương của AQI, cần thiết để tính độ lệch chuẩn (std\_aqi).
3. count\_aqi: Số lượng mẫu dữ liệu, cần thiết để tính cả trung bình và độ lệch chuẩn.

### ****Đề xuất cho DDS:****

* **Loại bỏ** std\_aqi **và** mean\_aqi **khỏi bảng Fact**, vì các giá trị này không có ý nghĩa ở mức chi tiết nhất và sẽ được tính toán lại một cách hiệu quả hơn ở mức độ tổng hợp cao trên Cube.
* **Thêm các cột hỗ trợ** như đã đề cập để hỗ trợ tính toán tổng hợp trên Cube.

### ****SQL Đề xuất để sửa đổi bảng Fact:****

ALTER TABLE AQI\_Fact

DROP COLUMN mean\_aqi,

DROP COLUMN std\_aqi;

ALTER TABLE AQI\_Fact

ADD sum\_aqi FLOAT NOT NULL DEFAULT 0,

sum\_squares\_aqi FLOAT NOT NULL DEFAULT 0,

count\_aqi INT NOT NULL DEFAULT 1;

**Bổ sung phân tích cho DDS**

**1. Về mặt tổng quan**

DDS được thiết kế để cung cấp nền tảng dữ liệu cho việc thu thập, lưu trữ và phân tích dữ liệu Chỉ số chất lượng không khí (AQI). Bao gồm bảng facts và dimensions, được xây dựng để hỗ trợ các truy vấn phân tích phức tạp và báo cáo trong thời gian thực.

**2. Mục Đích**

Cung cấp một giải pháp lưu trữ dữ liệu hiệu quả và một nền tảng phân tích dữ liệu chuyên sâu, giúp các nhà quản lý và phân tích viên hiểu rõ hơn về các xu hướng và yếu tố ảnh hưởng đến chất lượng không khí.

**3. Mô hình dữ liệu**

**Bảng Dimension**

* **Dim\_State**
  + state\_id: INT, Khóa chính.
  + state\_code: VARCHAR(2), Mã bang duy nhất.
  + state\_name: VARCHAR(255), Tên đầy đủ của bang.
  + state\_abbr: VARCHAR(2), Tên viết tắt của bang.
* **Dim\_County**
  + county\_id: INT, Khóa chính.
  + county\_name: VARCHAR(255), Tên quận/huyện.
  + county\_fips: VARCHAR(5), Mã FIPS duy nhất của quận/huyện.
  + state\_id: INT, Khóa ngoại tham chiếu đến Dim\_State.
* **Dim\_Date**
  + date\_id: INT, Khóa chính.
  + date: DATE, Ngày thực tế.
  + year, quarter, month, day: INT, Thông tin thời gian chi tiết.
  + daylightsaving: BIT, Chỉ thị giờ tiết kiệm ánh sáng.
* **Dim\_Category**
  + category\_id: INT, Khóa chính.
  + category\_name: VARCHAR(255), Tên danh mục AQI.
  + min\_value, max\_value: INT, Giá trị nhỏ nhất và lớn nhất cho danh mục.
  + aqi\_color: VARCHAR(255), Màu sắc đại diện cho AQI.
  + description: VARCHAR(255), Mô tả chi tiết về danh mục.
* **Dim\_DefiningParameter**
  + parameter\_id: INT, Khóa chính tự tăng.
  + parameter\_name: VARCHAR(255), Tên của tham số đo lường.

**Bảng Fact (AQI\_Fact)**

* fact\_id: INT, Khóa chính tự tăng.
* county\_id, date\_id, category\_id, parameter\_id: INT, Khóa ngoại liên kết với các bảng dimensions.
* mean\_aqi, std\_aqi, min\_aqi, max\_aqi: FLOAT, Các chỉ số AQI.
* sum\_aqi, sum\_squares\_aqi, count\_aqi, count\_day: INT, Các chỉ số hỗ trợ tính toán tổng hợp.

**4. Ý tưởng xử lý thay đổi dữ liệu**

DDS sử dụng cả ba loại Slowly Changing Dimension (SCD) để quản lý thay đổi:

* **Overwrite:** Đối với các thay đổi không cần giữ lịch sử.
* **Track History:** Giữ lại lịch sử thay đổi, đặc biệt cho Dim\_County và Dim\_State để theo dõi sự thay đổi về địa lý và hành chính.
* **Store Previous Value:** Lưu trữ giá trị trước và sau khi thay đổi trong cùng một bản ghi, áp dụng cho Dim\_Category khi các thông số như min\_value và max\_value thay đổi.

**OPTIONAL DDS:**

**Các measure có cần phải bỏ hết hay không?**

Quyết định bỏ mean\_aqi và std\_aqi khỏi DDS và thay vào đó sử dụng các thuộc tính hỗ trợ tính toán trên Cube SSAS là có hợp lý vì các giá trị này không cung cấp ý nghĩa ở mức chi tiết thấp nhất của dữ liệu. Tuy nhiên, việc có nên loại bỏ min\_aqi, max\_aqi, và count\_day khỏi bảng Fact hay không phụ thuộc vào nhu cầu phân tích cho từng mục đích khác nhau.

**Xem xét giữ min\_aqi, max\_aqi, và count\_day:**

**1. min\_aqi và max\_aqi:**

* **Giữ lại**: Nếu các phân tích thường xuyên cần đến các giá trị cực trị của AQI để hiểu biên độ dao động của chất lượng không khí trong một khoảng thời gian nhất định, thì việc giữ lại min\_aqi và max\_aqi trong bảng Fact là cần thiết. Việc này cho phép truy vấn trực tiếp các giá trị này mà không cần phải tính toán chúng từ dữ liệu chi tiết, nhờ đó giảm thời gian truy vấn và tăng hiệu quả.
* **Loại bỏ**: Nếu các giá trị này chỉ được sử dụng để hỗ trợ tính toán tổng hợp hoặc không cần thiết cho các báo cáo hàng ngày, ta có thể loại bỏ chúng và chỉ tính chúng khi cần thiết trên Cube.

**2. count\_day:**

* **Giữ lại**: Nếu count\_day (số ngày được đo) có ý nghĩa quan trọng trong việc phân tích, ví dụ như để xác định mức độ độ tin cậy của dữ liệu hoặc để phân tích sự biến đổi AQI qua các ngày trong một khoảng thời gian, thì nên giữ lại thuộc tính này. Nó cũng hữu ích trong việc xác định tổng số ngày có dữ liệu trong một khoảng thời gian nhất định.
* **Loại bỏ**: Nếu tính năng này không mang lại giá trị thêm trong báo cáo hoặc phân tích, ta có thể loại bỏ nó để đơn giản hóa mô hình.

**Gợi ý đề xuất:**

Nếu quyết định loại bỏ mean\_aqi và std\_aqi, nhưng giữ lại min\_aqi, max\_aqi, và count\_day, ta phải đảm bảo các thuộc tính này thực sự hỗ trợ cho các yêu cầu phân tích của bạn trước khi quyết định giữ chúng. Đồng thời, xem xét việc thêm các thuộc tính hỗ trợ như sum\_aqi, sum\_squares\_aqi, và count\_aqi để tối ưu việc tổng hợp và phân tích trên Cube SSAS.

Đây là một quá trình cân nhắc giữa việc đơn giản hóa mô hình dữ liệu và đáp ứng nhu cầu phân tích dữ liệu. Quyết định cuối cùng vẫn là nên dựa trên các mục tiêu cụ thể của tổ chức và các yêu cầu phân tích từ người dùng.

**Giải thích lý do sử dụng partition, số lượng dữ liệu có đủ nhiều để cần tối ưu không?**

Partitioning trong csdl, đặc biệt trong một hệ thống dữ liệu lớn như Data Warehouse (DW), là một chiến lược quan trọng để tối ưu hóa hiệu suất truy vấn và quản lý dữ liệu hiệu quả. Dưới đây là các lý do cụ thể và việc đánh giá xem có cần áp dụng cho dữ liệu hiện tại hay không.

**Lý do sử dụng**

1. **Cải thiện hiệu suất truy vấn:**
   * **Partition Pruning:** Khi truy vấn chỉ tác động đến một phần nhất định của dữ liệu, hệ thống có thể bỏ qua các phân vùng không liên quan, giảm đáng kể lượng dữ liệu cần xử lý.
   * **Phân phối I/O:** Dữ liệu được phân tán trên nhiều đĩa, giúp cải thiện hiệu suất đọc/ghi vì các truy vấn có thể được xử lý song song trên nhiều đĩa.
2. **Quản lý dữ liệu dễ dàng hơn:**
   * **Bảo trì:** Các hoạt động bảo trì như xây dựng chỉ mục, sao lưu và khôi phục có thể được thực hiện ở cấp phân vùng, giúp giảm thiểu thời gian và tài nguyên cần thiết.
   * **Quản lý dung lượng:** Các phân vùng có thể được lưu trữ trên các hệ thống lưu trữ khác nhau tùy theo mức độ quan trọng và tần suất truy cập.
3. **Load và Unload dữ liệu hiệu quả:**
   * Có thể nhanh chóng load hoặc unload dữ liệu ở cấp phân vùng mà không ảnh hưởng đến phần còn lại của bảng.

**Đánh giá Số lượng Dữ liệu và Nhu cầu Partitioning cho đồ án.**

1. **Kích thước và Độ phức tạp của Dữ liệu:**
   * Tệp dữ liệu về chất lượng không khí của các bang (AQI) cho năm 2022 và 2023, và dữ liệu về quận/hạt của Hoa Kỳ.
   * Nếu mỗi bảng chứa hàng triệu bản ghi (như trong trường hợp các bảng dữ liệu AQI tổng hợp hàng năm cho mỗi quận), việc áp dụng partitioning là hợp lý.
2. **Tần suất Truy cập:**
   * Dữ liệu được truy cập thường xuyên không? Nếu dữ liệu cũ ít được truy cập, có thể lưu trữ chúng trong phân vùng lạnh để giảm chi phí và tối ưu hóa hiệu suất truy cập dữ liệu phổ biến hơn.
3. **Chiến lược Phân vùng:**
   * Dựa trên **Dim\_Date** để phân vùng theo thời gian (ví dụ, phân vùng hàng năm hoặc hàng quý).
   * Có thể áp dụng phân vùng dựa trên **state\_id** hoặc **county\_id** để phân chia dữ liệu theo địa lý.

**Kết luận**

Nếu kích thước và tần suất truy cập của dữ liệu trong các tệp lớn và đòi hỏi tối ưu hóa hiệu suất, thì việc sử dụng partitioning là một lựa chọn thích hợp. Điều này không chỉ giúp cải thiện tốc độ truy vấn mà còn đơn giản hóa quản lý dữ liệu.

**Tìm hiểu cách thiết kế DDS với nhiều aggregare view để đáp ứng các yêu cầu phân tích khi bảng Fact đang ở mức chi tiết thấp nhất, có cần bỏ bảng Fact đó luôn hay không vì không có ý nghĩa tổng hợp**

Khi xây dựng một Data Warehouse (DW) và thiết kế bảng Fact ở mức chi tiết thấp nhất, một trong những thách thức là cân nhắc giữa việc lưu trữ dữ liệu chi tiết với nhu cầu tối ưu hóa truy vấn thông qua việc sử dụng aggregate views. Aggregate views là cấu trúc dữ liệu được thiết kế để tổng hợp sẵn dữ liệu dựa trên các tiêu chí phân tích cụ thể, nhằm giảm thiểu thời gian truy vấn và cải thiện hiệu suất.

**Tại sao cần Aggregate Views?**

1. **Hiệu Suất Truy Vấn:**
   * Cải thiện đáng kể thời gian truy vấn bằng cách giảm số lượng dữ liệu cần xử lý.
   * Trả lời nhanh các truy vấn phân tích tổng hợp thường gặp.
2. **Giảm Tải Xử Lý:**
   * Giảm bớt gánh nặng xử lý cho hệ thống DW khi thực hiện các phân tích phức tạp và đòi hỏi tính toán trên lượng lớn dữ liệu.
3. **Tối Ưu Hóa Chi Phí Lưu Trữ:**
   * Giảm bớt nhu cầu lưu trữ dữ liệu chi tiết thô có thể không cần thiết cho mọi loại phân tích.

**Xây dựng Aggregate Views:**

1. **Xác Định Nhu Cầu Phân Tích:**
   * Phân tích các truy vấn phân tích thường gặp để xác định các dim và measure cần tổng hợp.
   * Ví dụ: Tổng hợp AQI theo bang, quý, loại AQI.
2. **Thiết kế Cấu trúc Aggregate:**
   * **Aggregate theo Thời Gian**: Tổng hợp dữ liệu theo năm, quý, tháng.
   * **Aggregate theo Địa Lý**: Tổng hợp dữ liệu theo khu vực, bang, thành phố.
   * **Aggregate theo Category**: Tổng hợp dữ liệu theo các phân loại chất lượng không khí.
3. **Triển Khai Aggregate Tables:**
   * Tạo các bảng tổng hợp từ bảng Fact chi tiết.
   * Ví dụ:
   * CREATE TABLE Agg\_AQI\_By\_State\_Quarter AS
   * SELECT state\_id, date\_id, AVG(mean\_aqi) as avg\_aqi, SUM(sum\_aqi) as total\_aqi
   * FROM AQI\_Fact
   * GROUP BY state\_id, date\_id;
4. **Lựa chọn Giữa Bảng Fact và Aggregate Views:**
   * **Bảo Lưu Bảng Fact Chi Tiết**: Nếu vẫn cần truy cập dữ liệu chi tiết cho các phân tích sâu hơn hoặc để đáp ứng các yêu cầu phân tích đặc biệt.
   * **Chỉ sử dụng Aggregate Views**: Nếu các truy vấn phân tích đều có thể được trả lời hiệu quả qua các bảng tổng hợp và không yêu cầu dữ liệu chi tiết.
5. **Cập nhật và Bảo trì:**
   * Thiết lập quy trình cập nhật định kỳ cho các bảng tổng hợp để đảm bảo dữ liệu luôn chính xác và cập nhật.

**Kết luận và gợi ý đề xuất:**

* **Không nên bỏ Bảng Fact chi tiết ngay lập tức**: Dữ liệu chi tiết cung cấp khả năng phân tích sâu và linh hoạt cho nhu cầu phân tích đặc biệt. Bảng Fact có thể được giữ lại và chỉ sử dụng khi cần truy vấn chi tiết.
* **Sử dụng Aggregate Views cho Phân tích hàng ngày**: Đáp ứng nhanh các yêu cầu phân tích phổ thông và thường xuyên, giảm tải cho DW.
* **Đánh giá thường xuyên**: Đánh giá hiệu suất, chi phí và nhu cầu phân tích để quyết định tối ưu giữa bảng Fact chi tiết và các Aggregate Views.

**SSAS**

**Mô phỏng về cách xây dựng khối cubes trong SSAS**

**Bước 1: Xác định yêu cầu phân tích**

Xây dựng các cubes để phân tích:

1. Min và max AQI theo từng quý, từng bang.
2. Trung bình và độ lệch chuẩn của AQI theo từng quý, từng bang.
3. Số ngày và trung bình AQI trong trạng thái "very unhealthy" hoặc tệ hơn cho mỗi bang và quận.
4. Đếm số ngày trong mỗi danh mục chất lượng không khí cho một số bang cụ thể.
5. Tính trung bình AQI theo từng quý cho một số bang cụ thể.
6. Biểu diễn xu hướng AQI theo năm cho một số bang cụ thể.

**Bước 2: Thiết kế và xây dựng CUBE**

**Khởi tạo Project SSAS**

1. Mở SQL Server Data Tools và tạo một dự án mới dưới loại "Analysis Services Multidimensional and Data Mining".
2. Đặt tên cho dự án và chọn vị trí lưu trữ.

**Tạo Data Source**

Kết nối tới cơ sở dữ liệu DDSDB chứa các bảng facts và dimensions.

**Tạo Data Source View**

Chọn các bảng facts và dimensions cần thiết từ cơ sở dữ liệu để tham gia vào cube.

**Tạo Dimensions**

Dùng Data Source View để tạo các dimensions:

* + **State Dimension**: Với state\_id làm khóa, bao gồm các thuộc tính như state\_code, state\_name, state\_abbr.
  + **County Dimension**: Với county\_id làm khóa, bao gồm county\_name, county\_fips.
  + **Date Dimension**: Với date\_id làm khóa, bao gồm date, year, quarter, month, day.
  + **Category Dimension**: Với category\_id làm khóa, bao gồm category\_name, min\_value, max\_value, aqi\_color, description.
  + **Defining Parameter Dimension**: Với parameter\_id làm khóa, bao gồm parameter\_name.

**Tạo Cube**

1. Sử dụng các bảng facts và đã liệt kê để tạo cube.
2. Thêm các measures như mean\_aqi, min\_aqi, max\_aqi, sum\_aqi, sum\_squares\_aqi, count\_aqi.
3. Thiết lập các relationships giữa các dimensions và facts.
4. Xác định các aggregation và calculated measures nếu cần.

**Deploy và Process Cube (chưa làm được)**

1. Deploy cube lên server Analysis Services.
2. Thực hiện process cube để chuẩn bị cho việc phân tích.

**Bước 3: Kiểm tra và phân tích dữ liệu (chưa)**

1. Sử dụng SQL Server Management Studio hoặc Excel để kết nối với cube và thực hiện các truy vấn OLAP.
2. Kiểm tra các yêu cầu phân tích đã được đề cập bằng cách sử dụng các truy vấn MDX hoặc các công cụ trực quan hóa dữ liệu.

**Kết Luận (Thêm hay k cũng được)**

Cubes trong SSAS sẽ cho phép ta nhanh chóng truy vấn và phân tích dữ liệu phức tạp, cung cấp cái nhìn sâu sắc về chất lượng không khí dựa trên dữ liệu từ nhiều bang và quận/huyện. Các báo cáo và trực quan hóa dữ liệu có thể được tạo ra dựa trên dữ liệu này để hỗ trợ các quyết định môi trường và y tế công cộng.

Trong SSAS, các thành phần chính được tạo để xây dựng một giải pháp phân tích bao gồm: Databases, Data Sources, Data Source Views, Cubes, Dimensions, Measures, và Measure Groups. Giải thích chi tiết về mỗi thành phần:

**1. Data Source**

Đây là kết nối giữa Project SSAS và cơ sở dữ liệu nguồn mà ta sẽ sử dụng để tạo cubes. Data source chứa thông tin cần thiết để kết nối tới cơ sở dữ liệu như tên server, tên cơ sở dữ liệu, phương thức xác thực, và các thông tin kết nối khác.

**2. Data Source View (DSV)**

Là một tầng trừu tượng giữa các objects trong Project SSAS và cơ sở dữ liệu nguồn. Cho phép ta định nghĩa, xem và quản lý các bảng, views, và relationships mà ta muốn sử dụng trong cube. DSV giúp lọc và chỉ định dữ liệu cụ thể nào từ data source sẽ được sử dụng trong cube.

**3. Dimensions**

Là các đối tượng mô tả các thông tin dạng thuộc tính mà theo đó bạn có thể phân tích dữ liệu trong cubes. Ví dụ, một Dimension có thể là "Time" với các thuộc tính là năm, tháng, quý, ngày. Mỗi Dimension được xây dựng từ một hoặc nhiều bảng hoặc views trong DSV và chứa một hoặc nhiều hierarchies và thuộc tính mà người dùng cuối có thể truy vấn.

**4. Cubes**

Cube là thành phần chính trong SSAS, nơi tổng hợp dữ liệu từ một hoặc nhiều bảng facts (bảng sự kiện) và được tổ chức xung quanh các dimensions. Cube cho phép phân tích dữ liệu trong nhiều chiều (Dimensions) và có khả năng tính toán và lưu trữ các tổng hợp (aggregations) để tăng tốc độ truy vấn.

**5. Measures và Measure Groups**

* **Measures** là các dữ liệu số mà ta muốn phân tích, thường là dữ liệu từ các cột trong bảng fact. Ví dụ: tổng doanh thu, số lượng bán, trung bình AQI.
* **Measure Groups** là nhóm các measures có liên quan đến nhau và thường đến từ cùng một bảng fact. Mỗi measure group tương ứng với một bảng fact trong cơ sở dữ liệu nguồn.

**6. Aggregations**

Là các tổng hợp dữ liệu được tính toán trước và lưu trữ trong cube để tăng tốc độ truy vấn. Đây là một phần quan trọng của cube vì nó cho phép phân tích dữ liệu nhanh chóng mà không cần phải truy vấn và tính toán từ cơ sở dữ liệu nguồn..

**7. Hierarchies**

Trong mỗi dimension, có thể thiết lập các hierarchies để định nghĩa cách dữ liệu được tổ chức và truy vấn. Hierarchies cho phép người dùng phân tích dữ liệu theo cấp độ từ tổng quát đến chi tiết.

**8. Roles và Security**

Roles trong SSAS cho phép định nghĩa quyền truy cập và hạn chế đối với dữ liệu trong cube. Ta có thể kiểm soát người dùng nào có thể truy cập cube và những phần nào của cube mà họ có thể xem.

Từng bước trong quá trình thiết kế và triển khai SSAS đều đóng vai trò quan trọng trong việc đảm bảo cube hoạt động hiệu quả và đáp ứng được yêu cầu phân tích của người dùng.