**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**KHO DỮ LIỆU VÀ OLAP**

**ĐỀ TÀI:**

**XÂY DỰNG KHO DỮ LIỆU PHÂN TÍCH**

**TAI NẠN GIAO THÔNG NƯỚC MỸ**

**GIAI ĐOẠN 2018 - 2023**

**Lớp**: IS207.Q13

**GVHD**: ThS. Đỗ Thị Minh Phụng

**Thành viên nhóm**: Ngô Tiến Sỹ - 23521367

Nguyễn Văn Nam – 23520982

**TP. HỒ CHÍ MINH, NĂM 2025**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2025*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**LỜI CẢM ƠN**

Kính gửi Cô Đỗ Thị Minh Phụng - giảng viên môn Kho Dữ liệu và OLAP,

Lời đầu tiên, nhóm thực hiện đề tài xin bày tỏ lòng biết ơn sâu sắc đến Cô. Chúng em vô cùng trân trọng những kiến thức quý báu mà Cô đã tận tâm truyền đạt trong suốt thời gian qua. Sự nhiệt huyết và phương pháp giảng dạy lôi cuốn của Cô đã khơi gợi niềm đam mê học hỏi trong chúng em.

Chúng em xin chân thành cảm ơn Cô vì những lời khuyên, định hướng và góp ý quý báu. Nhờ có sự chỉ dẫn tận tình của Cô, chúng em đã có thể xác định được hướng đi đúng đắn và hoàn thành đề tài này.

Trong quá trình thực hiện đồ án, chúng em đã cố gắng vận dụng tối đa những kiến thức được Cô trang bị trên lớp, đồng thời tích cực tìm tòi, nghiên cứu thêm từ nhiều nguồn tài liệu khác nhau. Mặc dù đã nỗ lực hết mình, do còn hạn chế về kinh nghiệm thực tế, đồ án của chúng em chắc chắn không tránh khỏi những thiếu sót.

Vì vậy, chúng em kính mong nhận được những nhận xét, đánh giá và góp ý chân thành từ Cô. Những ý kiến đóng góp của Cô sẽ là nguồn động lực to lớn giúp chúng em củng cố kiến thức, trau dồi kinh nghiệm và hoàn thiện hơn trong các dự án tương lai.

Xin chân thành cảm ơn Cô!

# LÝ DO CHỌN ĐỀ TÀI

Trong bối cảnh các vấn đề an toàn giao thông ngày càng được quan tâm, việc nghiên cứu và phân tích tình hình tai nạn giao thông tại Hoa Kỳ trong giai đoạn 2018 –2023 trở nên đặc biệt cần thiết. Đề tài này không chỉ cung cấp cái nhìn toàn diện về thực trạng tai nạn giao thông mà còn mang lại nguồn dữ liệu có giá trị cao cho các bài toán phân tích và hỗ trợ ra quyết định trong lĩnh vực quản lý giao thông.

Dataset được lựa chọn sở hữu cấu trúc đa chiều rõ ràng và phong phú, bao gồm các thông tin về nguồn thu thập dữ liệu, thời gian, địa điểm, điều kiện thời tiết cùng nhiều yếu tố môi trường khác. Đặc điểm này tạo điều kiện lý tưởng để xây dựng các mô hình kho dữ liệu theo Star Schema hoặc Snowflake Schema, từ đó triển khai hiệu quả các phép phân tích OLAP. Thông qua việc thực hiện các kỹ thuật phân tích như drill-down, roll-up và slice/dice, nhóm có thể khám phá sâu sắc các yếu tố tác động đến tai nạn giao thông, nhận diện các xu hướng tiềm ẩn và đưa ra những giải pháp khoa học để cải thiện an toàn giao thông.

Bên cạnh đó, khả năng tích hợp dataset với các nguồn dữ liệu bổ sung như thông tin dân số, mật độ giao thông và dữ liệu khí tượng sẽ mở rộng đáng kể phạm vi nghiên cứu và nâng cao chất lượng phân tích. Sự kết hợp đa chiều này không chỉ giúp làm sáng tỏ các mối quan hệ phức tạp giữa các yếu tố ảnh hưởng mà còn tạo ra nền tảng vững chắc cho việc đề xuất các chính sách và biện pháp cải thiện an toàn giao thông có tính khả thi cao.

Như vậy, việc lựa chọn đề tài này vừa thể hiện tính cấp thiết của vấn đề thực tiễn, vừa mang ý nghĩa quan trọng trong việc ứng dụng và phát triển các phương pháp phân tích dữ liệu hiện đại trong lĩnh vực giao thông, đồng thời phù hợp với mục tiêu học tập và nghiên cứu của môn Kho Dữ liệu và OLAP.

# MỤC LỤC

[LÝ DO CHỌN ĐỀ TÀI 4](#_Toc210152571)

[MỤC LỤC 5](#_Toc210152572)

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ DỮ LIỆU 6](#_Toc210152573)

[1.1 PHÁT BIỂU VỀ DỮ LIỆU 6](#_Toc210152574)

[1.1.1 Mô tả dữ liệu 6](#_Toc210152575)

[1.1.2 Tiền xử lý. 7](#_Toc210152576)

[1.1.3 Thuộc tính của dữ liệu. 12](#_Toc210152577)

[1.1.4 Hướng chủ đề. 14](#_Toc210152578)

[1.2 XÂY DỰNG KHO DỮ LIỆU 14](#_Toc210152579)

[1.2.1 Thiết kế lược đồ. 14](#_Toc210152580)

[1.2.2 Các bảng chiều. 15](#_Toc210152581)

[1.3 CÁC CÂU TRUY VẤN 18](#_Toc210152582)

[CHƯƠNG 2: QUÁ TRÌNH XÂY DỰNG KHO DỮ LIỆU (SSIS) 19](#_Toc210152583)

[2.1 CHUẨN BỊ CÁC CÔNG CỤ 19](#_Toc210152584)

[2.1.1 Cài đặt SQL Server 2022 Developer 19](#_Toc210152585)

[2.2.2 Cài đặt Visual Studio 2022 Community với Extension SQL Server Intergration Services Projects 2022, Microsoft Analysis Services Projects 25](#_Toc210152586)

# CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ DỮ LIỆU

## PHÁT BIỂU VỀ DỮ LIỆU

### 1.1.1 Mô tả dữ liệu

* Tên bộ dữ liệu: US Accidents (2016 – 2023).
* Tác giả: Sobhan Moosavi.
* Đây là Dataset về tai nạn xe hơi trên toàn nước Mỹ, bao phủ **49 bang**. Dữ liệu tai nạn được thu thập từ **tháng 2/2016 đến tháng 3/2023** cho toàn bộ các bang thuộc lục địa Hoa Kỳ, sử dụng nhiều API phát trực tuyến dữ liệu sự cố (hoặc sự kiện) giao thông. Các API này truyền dữ liệu giao thông được ghi nhận từ nhiều nguồn khác nhau, bao gồm:
  + Bộ Giao thông Vận tải Hoa Kỳ và các bang.
  + Cơ quan thực thi pháp luật.
  + Camera giao thông.
  + Cảm biến giao thông trong mạng lưới đường bộ.
* Kho dữ liệu gồm 7,728,394 dòng và 46 thuộc tính.
* Link dataset: [US Accidents (2016 - 2023)](https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents).

A screenshot of a web page

AI-generated content may be incorrect.

### 1.1.2 Tiền xử lý.

Tiền xử lý dữ liệu là bước quan trọng nhằm làm sạch, tối ưu và chuẩn bị dữ liệu trước khi phân tích và lưu trữ vào cơ sở dữ liệu. Giai đoạn này giúp loại bỏ các cột không cần thiết, xử lý giá trị thiếu, giảm kích thước dữ liệu, chuẩn hóa tên cột và kiểu dữ liệu, đồng thời tạo thêm các đặc trưng phục vụ cho phân tích như thời lượng tai nạn hay các thuộc tính thời gian. Quá trình được thực hiện theo nhiều pha liên tiếp để đảm bảo dữ liệu đầu ra có cấu trúc rõ ràng, chất lượng cao và sẵn sàng để import vào SQL Server.

#### 1.1.2.1 Pha 1: Tính Duration và xóa các cột không cần thiết.

Trong pha đầu tiên, hệ thống tính toán thời lượng xảy ra của mỗi vụ tai nạn dựa trên thời gian bắt đầu và kết thúc, đồng thời loại bỏ các cột không cần thiết hoặc không phục vụ mục tiêu phân tích nhằm giảm kích thước và đơn giản hóa dữ liệu.

|  |
| --- |
| def phase\_delete\_columns(df: pd.DataFrame, columns\_to\_delete: List[str]) -> pd.DataFrame:      """Pha 1: Xóa cột không cần thiết và tính DURATION"""      # Tính toán DURATION trước khi xóa End\_Time      if 'Start\_Time' in df.columns and 'End\_Time' in df.columns:          # Chuyển đổi về datetime nếu chưa phải          df['Start\_Time'] = pd.to\_datetime(df['Start\_Time'], errors='coerce')          df['End\_Time'] = pd.to\_datetime(df['End\_Time'], errors='coerce')            # Tính DURATION bằng giây (làm tròn thành số nguyên)          df['DURATION'] = (df['End\_Time'] - df['Start\_Time']).dt.total\_seconds()          df['DURATION'] = df['DURATION'].fillna(0).round().astype('int64')            # Xử lý giá trị âm (đặt về 0)          df.loc[df['DURATION'] < 0, 'DURATION'] = 0        # Xóa các cột không cần thiết      columns\_to\_drop = [col for col in columns\_to\_delete if col in df.columns]      return df.drop(columns=columns\_to\_drop) if columns\_to\_drop else df |

#### 1.1.2.2 Pha 2: Lọc dữ liệu theo mốc thời gian.

Dữ liệu được lọc để chỉ giữ lại các bản ghi từ năm 2018 trở về sau, đảm bảo tính cập nhật và phù hợp với phạm vi phân tích của đề tài.

|  |
| --- |
| def phase\_filter\_date(df: pd.DataFrame, time\_column: str = 'Start\_Time',                       date\_cutoff: str = "2018-01-01") -> pd.DataFrame:      """Pha 2: Lọc dữ liệu theo mốc thời gian"""      if time\_column not in df.columns:          return df        # Convert to datetime if needed      if df[time\_column].dtype != 'datetime64[ns]':          df[time\_column] = pd.to\_datetime(df[time\_column], errors='coerce')        cutoff\_date = pd.to\_datetime(date\_cutoff)      return df[df[time\_column] >= cutoff\_date] |

#### 1.1.2.3 Pha 3: Tạo đặc trưng thời gian.

Từ dữ liệu thời gian ban đầu, hệ thống trích xuất thêm các đặc trưng như năm, tháng, ngày, giờ, phút, giây, quý và cờ cuối tuần. Các đặc trưng này giúp hỗ trợ phân tích xu hướng và mô hình hóa dữ liệu theo thời gian.

|  |
| --- |
| def phase\_create\_time\_features(df: pd.DataFrame, time\_column: str = 'Start\_Time') -> pd.DataFrame:      """Pha 3: Tạo đặc trưng thời gian"""      if time\_column not in df.columns:          return df        # Chuyển đổi sang datetime nếu cần      df[time\_column] = pd.to\_datetime(df[time\_column], errors='coerce')        # Tạo đặc trưng thời gian cơ bản      df['YEAR'] = df[time\_column].dt.year.astype('int16')      df['QUARTER'] = df[time\_column].dt.quarter.astype('int8')      df['MONTH'] = df[time\_column].dt.month.astype('int8')      df['DAY'] = df[time\_column].dt.day.astype('int8')      df['HOUR'] = df[time\_column].dt.hour.astype('int8')      df['MINUTE'] = df[time\_column].dt.minute.astype('int8')      df['SECOND'] = df[time\_column].dt.second.astype('int8')      df['IS\_WEEKEND'] = df[time\_column].dt.dayofweek.isin([5, 6]).astype('bool')        # Loại bỏ cột thời gian gốc để tránh dư thừa      return df.drop(columns=[time\_column]) |

#### 1.1.2.4 Pha 4: Chuyển đổi kiểu dữ liệu

Các cột được chuyển đổi về kiểu dữ liệu phù hợp với SQL Server (như INT, FLOAT, NVARCHAR, BIT) để tối ưu hóa khả năng lưu trữ, truy vấn và đảm bảo tính toàn vẹn dữ liệu.

|  |
| --- |
| def phase\_sql\_data\_types(df: pd.DataFrame) -> pd.DataFrame:      """Pha 4: Chuyển đổi kiểu dữ liệu SQL Server"""        # Tọa độ: decimal(9,6)      coord\_cols = ['Start\_Lat', 'Start\_Lng', 'LATITUDE', 'LONGITUDE']      for col in coord\_cols:          if col in df.columns:              df[col] = df[col].round(6).astype('float64')        # Số thực khác: decimal(8,4)      float\_cols = df.select\_dtypes(include=['float64', 'float32']).columns      for col in float\_cols:          if col not in coord\_cols:              df[col] = df[col].round(4).astype('float64')        # Số nguyên: int/smallint/tinyint/bit      int\_cols = df.select\_dtypes(include=['int64', 'int32']).columns      for col in int\_cols:          if col in ['YEAR']:              df[col] = df[col].astype('int16')  # smallint          elif col in ['QUARTER', 'MONTH', 'DAY', 'HOUR', 'MINUTE', 'SECOND']:              df[col] = df[col].astype('int8')   # tinyint          elif col == 'DURATION':              df[col] = df[col].astype('int64')  # bigint cho DURATION (giây)          else:              df[col] = df[col].astype('int32')  # int32        # Danh sách các cột environment và IS\_WEEKEND luôn là Boolean (BIT)      environment\_boolean\_cols = [          'IS\_WEEKEND', 'AMENITY', 'BUMP', 'CROSSING', 'GIVE\_WAY', 'JUNCTION',          'NO\_EXIT', 'RAILWAY', 'ROUNDABOUT', 'STATION', 'STOP',          'TRAFFIC\_CALMING', 'TRAFFIC\_SIGNAL', 'TURNING\_LOOP'      ]        # Đảm bảo các cột environment được convert về bool nếu chúng là numeric      for col in environment\_boolean\_cols:          if col in df.columns:              if df[col].dtype in ['int8', 'int16', 'int32', 'int64', 'float32', 'float64']:                  # Convert về bool (0 -> False, non-zero -> True)                  df[col] = df[col].astype('bool')      # Chuỗi      string\_cols = df.select\_dtypes(include=['object']).columns      for col in string\_cols:          df[col] = df[col].astype('string')  # Sử dụng string type để mapping SQL      return df |

#### 1.1.2.5 Pha 5: Chuẩn hóa tên cột.

Tên các cột được chuẩn hóa về dạng chữ hoa và tuân theo quy tắc đặt tên nhất quán, giúp dễ dàng quản lý và tương thích với các quy ước trong hệ quản trị cơ sở dữ liệu.

|  |
| --- |
| def phase\_standardize\_columns(df: pd.DataFrame) -> pd.DataFrame:      """Pha 5: Chuẩn hóa tên cột"""      column\_mapping = {}      for col in df.columns:          # Chuyển tên cột thành chữ hoa và chuẩn hóa          new\_col = col.upper()          # Loại bỏ ký tự đặc biệt và thay thế khoảng trắng          new\_col = re.sub(r'\([^)]\*\)', '', new\_col)          # Thay thế khoảng trắng bằng dấu gạch dưới          new\_col = re.sub(r'\s+', '\_', new\_col.strip())          # Loại bỏ các ký tự không phải chữ cái, số, hoặc dấu gạch dưới          new\_col = re.sub(r'\_+', '\_', new\_col).strip('\_')          column\_mapping[col] = new\_col        # Đổi tên cột      df = df.rename(columns=column\_mapping)        # Đổi tên tọa độ cụ thể      coordinate\_mapping = {'START\_LAT': 'LATITUDE', 'START\_LNG': 'LONGITUDE'}      for old\_name, new\_name in coordinate\_mapping.items():          if old\_name in df.columns:              df = df.rename(columns={old\_name: new\_name})        return df |

#### 1.1.2.6 Pha 6: Sắp xếp lại thứ tự cột theo DDL SQL Server.

Cuối cùng, các cột được sắp xếp lại theo đúng thứ tự thiết kế của bảng dữ liệu trong SQL Server, giúp quá trình import và xử lý dữ liệu diễn ra thuận lợi và nhất quán.

|  |
| --- |
| def phase\_reorder\_columns(df: pd.DataFrame) -> pd.DataFrame:      """Pha 6: Sắp xếp lại thứ tự cột theo DDL SQL Server"""        # Định nghĩa thứ tự cột theo DDL - fact attributes lên đầu      fact\_columns = ['SEVERITY', 'DISTANCE', 'DURATION']        # Các nhóm cột dimension theo thứ tự DDL      source\_columns = ['SOURCE']        time\_columns = ['YEAR', 'QUARTER', 'MONTH', 'DAY', 'HOUR', 'MINUTE', 'SECOND', 'IS\_WEEKEND']        location\_columns = ['STATE', 'COUNTY', 'CITY', 'STREET', 'ZIPCODE', 'AIRPORT\_CODE', 'TIMEZONE', 'LATITUDE', 'LONGITUDE']        weather\_columns = ['TEMPERATURE', 'WIND\_CHILL', 'HUMIDITY', 'PRESSURE', 'VISIBILITY',                        'WIND\_DIRECTION', 'WIND\_SPEED', 'PRECIPITATION', 'WEATHER\_CONDITION',                        'SUNRISE\_SUNSET', 'CIVIL\_TWILIGHT', 'NAUTICAL\_TWILIGHT', 'ASTRONOMICAL\_TWILIGHT']        environment\_columns = ['AMENITY', 'BUMP', 'CROSSING', 'GIVE\_WAY', 'JUNCTION', 'NO\_EXIT',                            'RAILWAY', 'ROUNDABOUT', 'STATION', 'STOP', 'TRAFFIC\_CALMING',                            'TRAFFIC\_SIGNAL', 'TURNING\_LOOP']        # Tạo danh sách cột theo thứ tự mong muốn      desired\_order = []        # Thêm fact columns trước (những cột có trong DataFrame)      for col in fact\_columns:          if col in df.columns:              desired\_order.append(col)        # Thêm các dimension columns theo thứ tự DDL      for group in [source\_columns, time\_columns, location\_columns, weather\_columns, environment\_columns]:          for col in group:              if col in df.columns:                  desired\_order.append(col)        # Thêm các cột còn lại (nếu có)      remaining\_cols = [col for col in df.columns if col not in desired\_order]      desired\_order.extend(remaining\_cols)        # Sắp xếp lại DataFrame theo thứ tự mong muốn      return df[desired\_order] |

### 1.1.3 Thuộc tính của dữ liệu.

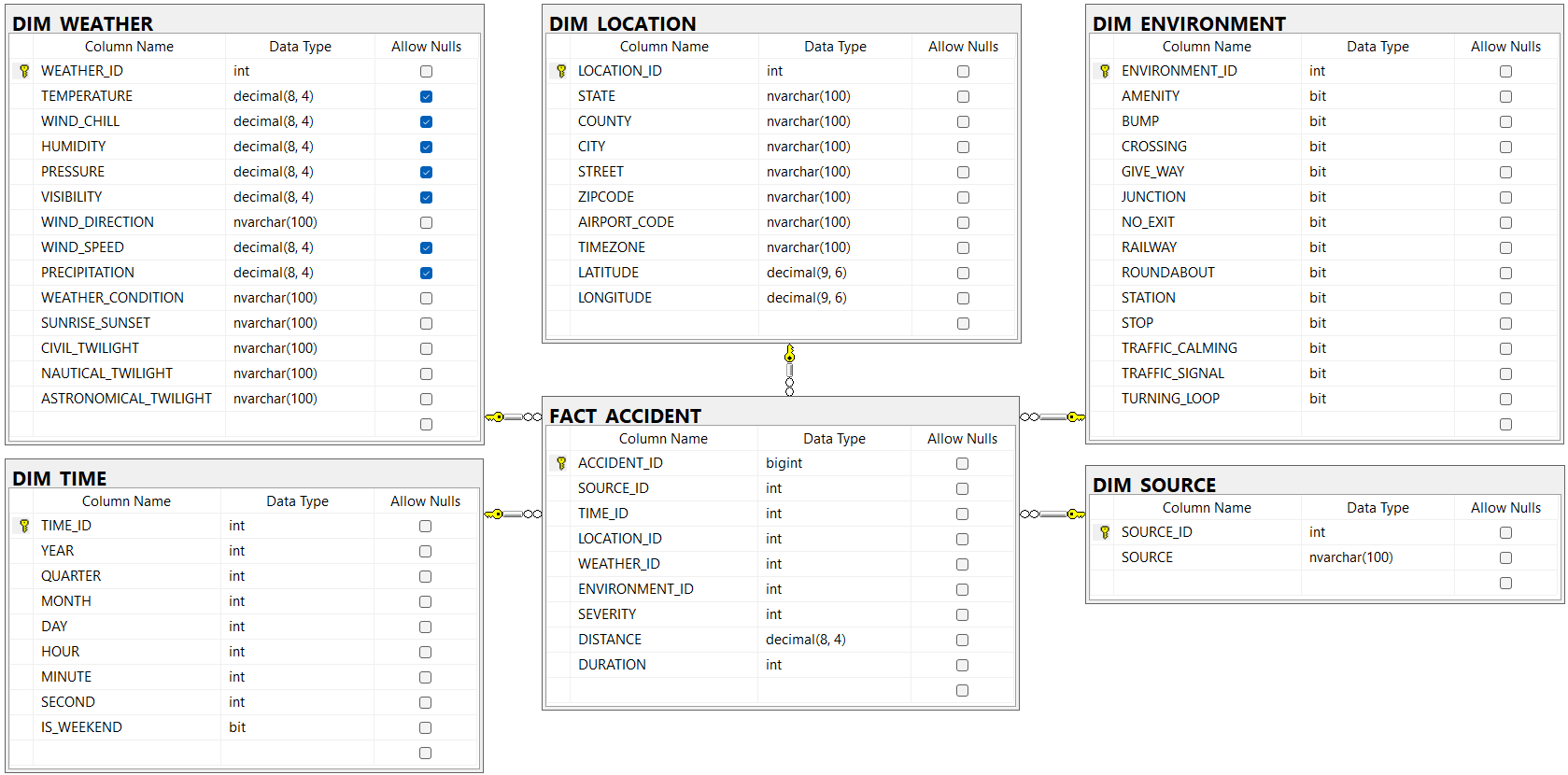
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | source | varchar(50) | Nguồn dữ liệu tai nạn thô (Source1, Source2, Other). |
| 2 | severity | int | Mức độ nghiêm trọng tai nạn (1-4), 1 = ít ảnh hưởng đến giao thông, 4 = ảnh hưởng nghiêm trọng. |
| 3 | latitude | float | Vĩ độ GPS của điểm bắt đầu tai nạn. |
| 4 | longitude | float | Kinh độ GPS của điểm bắt đầu tai nạn. |
| 5 | distance | float | Độ dài đoạn đường bị ảnh hưởng bởi tai nạn (đơn vị: dặm) |
| 6 | second | int | Thời điểm xảy ra tai nạn (giây). |
| 7 | minute | int | Thời điểm xảy ra tai nạn (phút). |
| 8 | hour | int | Thời điểm xảy ra tai nạn (giờ). |
| 9 | day | int | Thời điểm xảy ra tai nạn (ngày). |
| 10 | month | int | Thời điểm xảy ra tai nạn (tháng). |
| 11 | quarter | int | Thời điểm xảy ra tai nạn (quý). |
| 12 | year | int | Thời điểm xảy ra tai nạn (năm). |
| 13 | is\_weekend | bit | Có phải cuối tuần không. |
| 14 | street | varchar(255) | Tên đường trong trường địa chỉ. |
| 15 | city | varchar(100) | Tên thành phố trong trường địa chỉ. |
| 16 | county | varchar(100) | Tên quận trong trường địa chỉ. |
| 17 | state | varchar(10) | Tên tiểu bang trong trường địa chỉ. |
| 18 | zipcode | varchar(20) | Mã bưu chính trong trường địa chỉ. |
| 19 | timezone | varchar(50) | Múi giờ dựa trên vị trí tai nạn (US/Eastern, US/Pacific, etc.). |
| 20 | airport\_code | varchar(10) | Mã sân bay của trạm quan trắc thời tiết gần nhất. |
| 21 | temperature | float | Nhiệt độ (độ Fahrenheit). |
| 22 | wind\_chill | float | Chỉ số gió lạnh (độ Fahrenheit). |
| 23 | humidity | float | Độ ẩm (phần trăm). |
| 24 | pressure | float | Áp suất khí quyển (inch). |
| 25 | visibility | float | Tầm nhìn (dặm). |
| 26 | wind\_direction | varchar(10) | Hướng gió (CALM, S, N, etc.). |
| 27 | wind\_speed | float | Tốc độ gió (dặm/giờ). |
| 28 | precipitation | float | Lượng mưa (inch, nếu có). |
| 29 | weather\_condition | varchar(100) | Điều kiện thời tiết (Fair, Mostly Cloudy, rain, snow, etc.). |
| 30 | amenity | bit | Có tiện ích gần đó hay không (POI annotation). |
| 31 | bump | bit | Có gờ giảm tốc gần đó hay không. |
| 32 | crossing | bit | Có lối băng qua đường gần đó hay không. |
| 33 | give\_way | bit | Có biển nhường đường gần đó hay không. |
| 34 | junction | bit | Có ngã tư/giao lộ gần đó hay không. |
| 35 | no\_exit | bit | Có biển cấm ra gần đó hay không. |
| 36 | railway | bit | Có đường sắt gần đó hay không. |
| 37 | roundabout | bit | Có bùng binh gần đó hay không. |
| 38 | station | bit | Có trạm (xe buýt, tàu, etc.) gần đó hay không. |
| 39 | stop | bit | Có biển dừng gần đó hay không. |
| 40 | traffic\_calming | bit | Có biện pháp làm chậm giao thông gần đó hay không. |
| 41 | traffic\_signal | bit | Có đèn giao thông gần đó hay không. |
| 42 | turning\_loop | bit | Có vòng xuyến gần đó hay không (không có giá trị true nào). |
| 43 | sunrise\_sunset | varchar(10) | Thời điểm trong ngày dựa trên mặt trời mọc/lặn (Day/Night). |
| 44 | civil\_twilight | varchar(10) | Thời điểm trong ngày dựa trên hoàng hôn dân dụng (Day/Night). |
| 45 | nautical\_twilight | varchar(10) | Thời điểm trong ngày dựa trên hoàng hôn hàng hải (Day/Night). |
| 46 | astronomical\_twilight | varchar(10) | Thời điểm trong ngày dựa trên hoàng hôn thiên văn (Day/Night). |

### 1.1.4 Hướng chủ đề.

Khám phá cách dữ liệu tai nạn giao thông có thể được tổ chức và khai thác trong kho dữ liệu để phân tích xu hướng tai nạn theo thời gian, địa điểm và điều kiện môi trường. Qua đó hỗ trợ các cơ quan chức năng trong việc nhận diện điểm nóng tai nạn, đánh giá mức độ ảnh hưởng của thời tiết và các yếu tố liên quan, tối ưu hóa công tác quy hoạch giao thông, nâng cao hiệu quả cảnh báo và giảm thiểu rủi ro tai nạn.

## 1.2 XÂY DỰNG KHO DỮ LIỆU

### 1.2.1 Thiết kế lược đồ.



### 1.2.2 Các bảng chiều.

#### 1.2.2.1 Bảng DIM\_SOURCE.

|  |  |  |  |
| --- | --- | --- | --- |
| **Khóa chính** | **Tên thuộc tính** | **Kiểu** | **Mô tả thuộc tính** |
| Key with solid fill | source\_id | int | Mã nguồn |
|  | source | nvarchar(100) | Nguồn dữ liệu tai nạn thô (Source1, Source2, Other). |
|  |  |  |  |

#### 1.2.2.2 Bảng DIM\_TIME.

|  |  |  |  |
| --- | --- | --- | --- |
| **Khóa chính** | **Tên thuộc tính** | **Kiểu** | **Mô tả thuộc tính** |
| Key with solid fill | time\_id | int | Mã thời gian |
|  | year | int | Thời điểm xảy ra tai nạn (năm). |
|  | quarter | int | Thời điểm xảy ra tai nạn (quý). |
|  | month | int | Thời điểm xảy ra tai nạn (tháng). |
|  | day | int | Thời điểm xảy ra tai nạn (ngày). |
|  | hour | int | Thời điểm xảy ra tai nạn (giờ). |
|  | minute | int | Thời điểm xảy ra tai nạn (phút). |
|  | second | int | Thời điểm xảy ra tai nạn (giây). |
|  | is\_weekend | bit | Có phải cuối tuần không. |
|  |  |  |  |

#### 1.2.2.3 Bảng DIM\_LOCATION.

|  |  |  |  |
| --- | --- | --- | --- |
| **Khóa chính** | **Tên thuộc tính** | **Kiểu** | **Mô tả thuộc tính** |
| Key with solid fill | location\_id | int | Mã vị trí. |
|  | state | nvarchar(100) | Tên tiểu bang trong trường địa chỉ. |
|  | county | nvarchar(100) | Tên quận trong trường địa chỉ. |
|  | city | nvarchar(100) | Tên thành phố trong trường địa chỉ. |
|  | street | nvarchar(100) | Tên đường trong trường địa chỉ. |
|  | zipcode | nvarchar(100) | Mã bưu chính trong trường địa chỉ. |
|  | airport\_code | nvarchar(100) | Mã sân bay của trạm quan trắc thời tiết gần nhất. |
|  | timezone | nvarchar(100) | Múi giờ dựa trên vị trí tai nạn (US/Eastern, US/Pacific, etc.). |
|  | latitude | decimal(9,6) | Vĩ độ GPS của điểm bắt đầu tai nạn. |
|  | longitude | decimal(9,6) | Kinh độ GPS của điểm bắt đầu tai nạn. |
|  |  |  |  |

#### 1.2.2.4 Bảng DIM\_WEATHER.

|  |  |  |  |
| --- | --- | --- | --- |
| **Khóa chính** | **Tên thuộc tính** | **Kiểu** | **Mô tả thuộc tính** |
| Key with solid fill | weather\_id | int | Mã thời tiết |
|  | temperature | decimal(8,4) | Nhiệt độ (độ Fahrenheit). |
|  | wind\_chill | decimal(8,4) | Chỉ số gió lạnh (độ Fahrenheit). |
|  | humidity | decimal(8,4) | Độ ẩm (phần trăm). |
|  | pressure | decimal(8,4) | Áp suất khí quyển (inch). |
|  | visibility | decimal(8,4) | Tầm nhìn (dặm). |
|  | wind\_direction | nvarchar(100) | Hướng gió (CALM, S, N, etc.). |
|  | wind\_speed | decimal(8,4) | Tốc độ gió (dặm/giờ). |
|  | precipitation | decimal(8,4) | Lượng mưa (inch, nếu có). |
|  | weather\_condition | nvarchar(100) | Điều kiện thời tiết (Fair, Mostly Cloudy, rain, snow, etc.). |
|  | sunrise\_sunset | nvarchar(100) | Thời điểm trong ngày dựa trên mặt trời mọc/lặn (Day/Night). |
|  | civil\_twilight | nvarchar(100) | Thời điểm trong ngày dựa trên hoàng hôn dân dụng (Day/Night). |
|  | nautical\_twilight | nvarchar(100) | Thời điểm trong ngày dựa trên hoàng hôn hàng hải (Day/Night). |
|  | astronomical\_twilight | nvarchar(100) | Thời điểm trong ngày dựa trên hoàng hôn thiên văn (Day/Night). |
|  |  |  |  |

#### 1.2.2.5 Bảng DIM\_ENVIRONMENT.

|  |  |  |  |
| --- | --- | --- | --- |
| **Khóa chính** | **Tên thuộc tính** | **Kiểu** | **Mô tả thuộc tính** |
| Key with solid fill | environment\_id | int | Mã môi trường. |
|  | amenity | bit | Có tiện ích gần đó hay không (POI annotation). |
|  | bump | bit | Có gờ giảm tốc gần đó hay không. |
|  | crossing | bit | Có lối băng qua đường gần đó hay không. |
|  | give\_way | bit | Có biển nhường đường gần đó hay không. |
|  | junction | bit | Có ngã tư/giao lộ gần đó hay không. |
|  | no\_exit | bit | Có biển cấm ra gần đó hay không. |
|  | railway | bit | Có đường sắt gần đó hay không. |
|  | roundabout | bit | Có bùng binh gần đó hay không. |
|  | station | bit | Có trạm (xe buýt, tàu, etc.) gần đó hay không. |
|  | stop | bit | Có biển dừng gần đó hay không. |
|  | traffic\_calming | bit | Có biện pháp làm chậm giao thông gần đó hay không. |
|  | traffic\_signal | bit | Có đèn giao thông gần đó hay không. |
|  | turning\_loop | bit | Có vòng xuyến gần đó hay không (không có giá trị true nào). |
|  |  |  |  |

#### 1.2.2.6 Bảng FACT\_ACCIDENT.

|  |  |  |  |
| --- | --- | --- | --- |
| **Khóa chính** | **Tên thuộc tính** | **Kiểu** | **Mô tả thuộc tính** |
| Key with solid fill | accident\_id | int | Mã tai nạn. |
| Key outline | source\_id | int | Mã nguồn. |
| Key outline | time\_id | int | Mã thời gian. |
| Key outline | location\_id | int | Mã vị trí. |
| Key outline | weather\_id | int | Mã thời tiết. |
| Key outline | environment\_id | int | Mã môi trường. |
|  | severity | int | Mức độ nghiêm trọng tai nạn (1-4), 1 = ít ảnh hưởng đến giao thông, 4 = ảnh hưởng nghiêm trọng. |
|  | distance | decimal(8,4) | Độ dài đoạn đường bị ảnh hưởng bởi tai nạn (đơn vị: dặm) |
|  | duration | int | Thời gian giao thông bị ảnh hưởng bởi tai nạn. |

## 1.3 CÁC CÂU TRUY VẤN

1. Danh sách tai nạn theo trạng thái có tín hiệu giao thông hay không.
2. Khoảng cách trung bình của vụ tai nạn theo tiểu bang.
3. Phân tích tai nạn theo múi giờ.
4. Ảnh hưởng của tầm nhìn (VISIBILITY) đến độ nghiêm trọng.
5. Phân bố tai nạn theo quý và năm.
6. Phân tích mức độ nghiêm trọng tai nạn theo từng giai đoạn ánh sáng (dựa trên twilight).
7. Danh sách tai nạn gần đường sắt hoặc trạm xe lửa theo từng năm.
8. Tần suất và mức độ nghiêm trọng của tai nạn gần các tiện ích.
9. Top 10 con đường nguy hiểm nhất và các yếu tố môi trường liên quan.
10. Top 5 địa điểm tai nạn theo độ nghiêm trọng trung bình.
11. Phân tích mối liên hệ giữa điều kiện thời tiết khắc nghiệt và số lượng/mức độ tai nạn.
12. Xếp hạng 5 thành phố có số vụ tai nạn cao nhất trong mỗi tiểu bang.
13. Phân tích tốc độ tăng trưởng số vụ tai nạn qua từng năm theo tiểu bang.
14. Top 5 thành phố có tốc độ tăng trưởng tai nạn trung bình cao nhất giai đoạn 2019–2022.
15. So sánh mức độ nghiêm trọng trung bình của tai nạn giữa cuối tuần và ngày thường theo từng giờ trong ngày.

# CHƯƠNG 2: QUÁ TRÌNH XÂY DỰNG KHO DỮ LIỆU (SSIS)

## 2.1 CHUẨN BỊ CÁC CÔNG CỤ

Để thực hiện quá trình SSIS ta cần chuẩn bị và cài đặt các phần mềm và công cụ sau:

* SQL Server 2022 Developer (Phiên bản 16.0.1150.1)
* SQL Server Management Studio 21 (Phiên bản 21.5.14)
* Visual Studio 2022 Community (Phiên bản 17.14.16) cùng với:
  + Extensions: SQL Server Integration Services Projects 2022, Microsoft Analysis Services Projects.
  + Workloads: Data storage and processing.

Trước khi bắt đầu cài đặt, máy của cá nhân phải đảm bảo thỏa các tiêu chí:

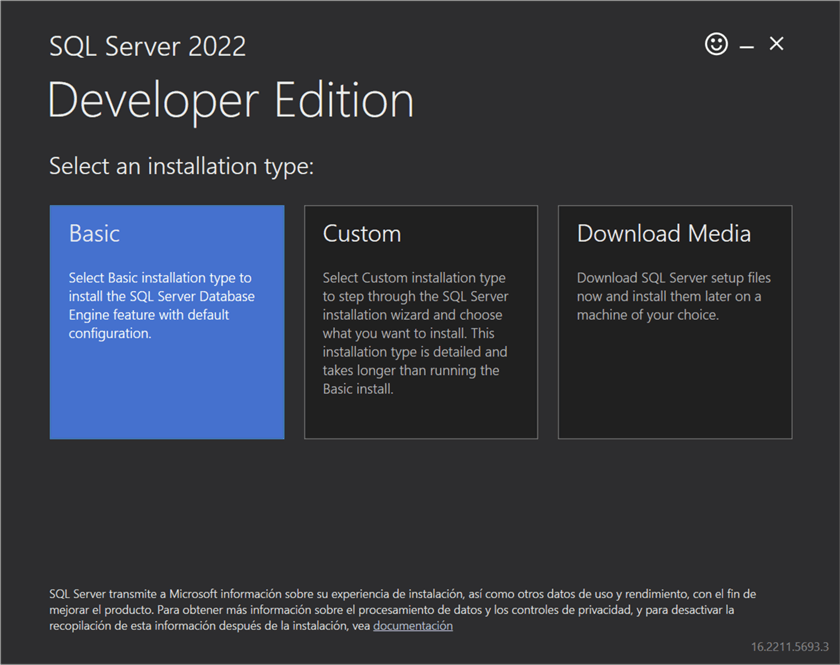
* Sử dụng hệ điều hành Windows 10/11 hoặc Windows Server từ 2016 tới 2022 (Standard and Datacenter).
* Tối thiểu 16GB RAM (Khuyến khích 32GB RAM để ổn định).
* Tối thiểu 30GB dung lượng lưu trữ trống (Khuyến khích 50GB dung lượng lưu trữ để ổn định).
* Đã có .NET Framework 4.8 trở lên. Nếu .NET Framework 4.8 chưa được cài đặt, chúng sẽ tự động được cài đặt trong quá trình cài đặt Visual Studio.

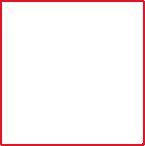
### 2.1.1 Cài đặt SQL Server 2022 Developer

Bước 1: Đầu tiên, tải file cài đặt về từ trang Microsoft.com: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Bước 2: Khởi chạy file cài đặt. Trong trình cài đặt sẽ xuất hiện 3 lựa chọn:

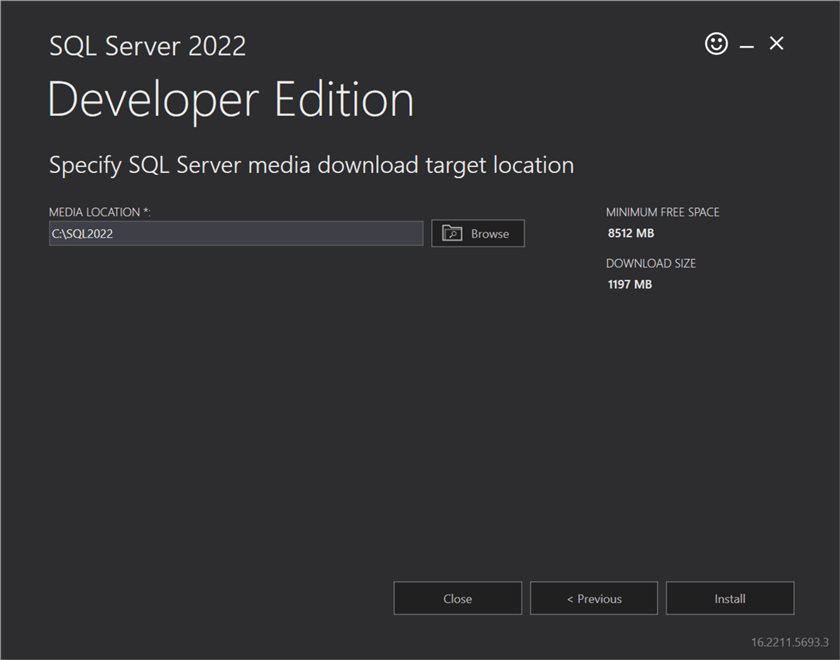
* Basic – Cài đặt SQL Server với thiết lập mặc định
* Custom – Tự chọn những thành phần muốn cài đặt
* Download Media – Tải trình cài đặt và chạy sau





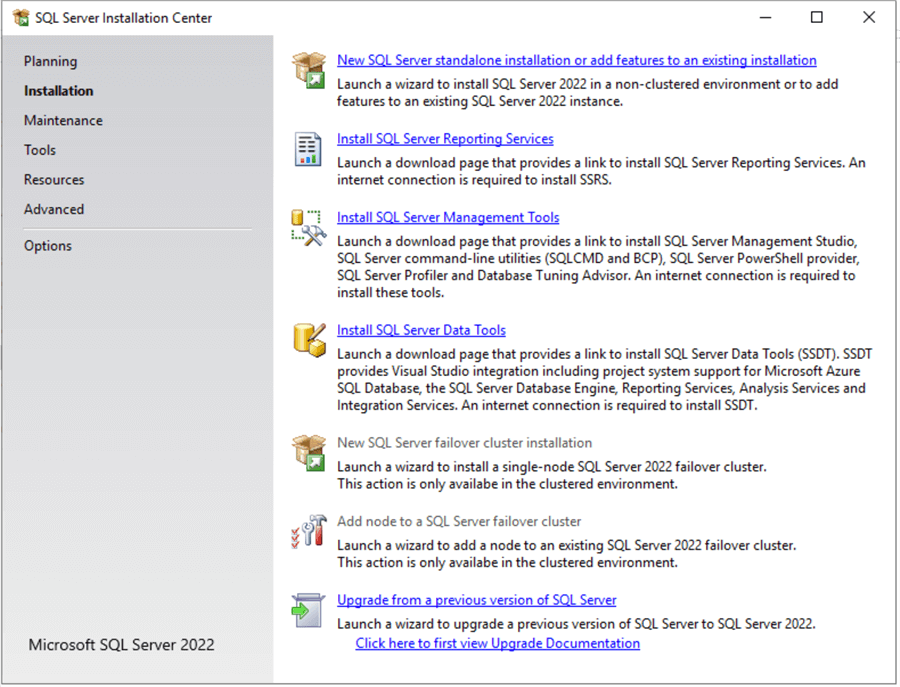
Để bắt đầu, người dùng chọn lựa chọn Custom

Bước 3: Trình cài đặt sẽ hỏi vị trí để lưu file tải về. Sau khi chọn xong, nhấn Install



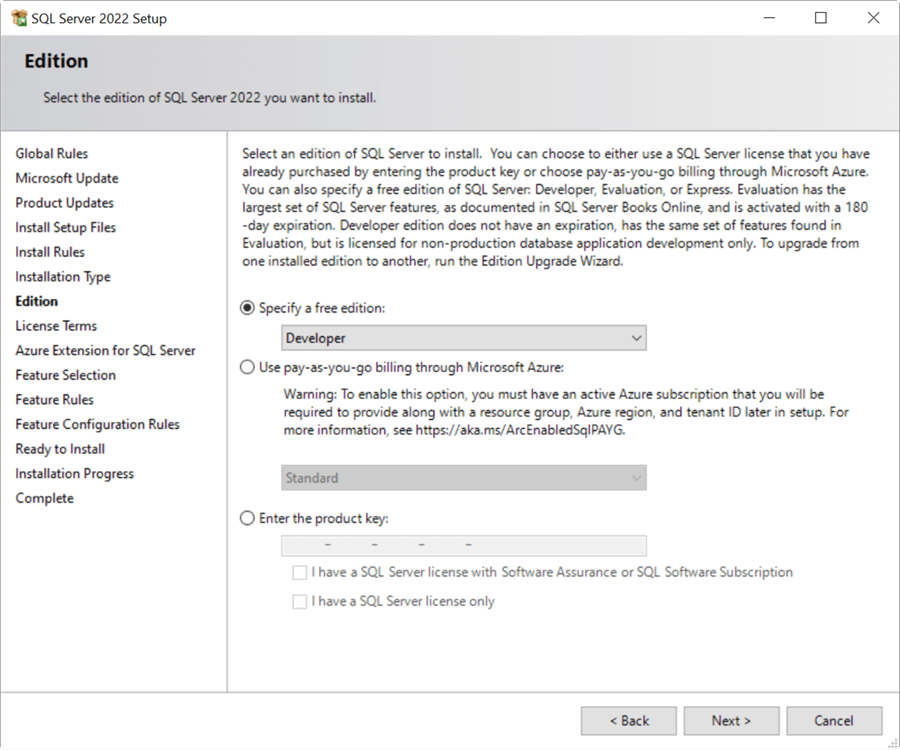


Bước 4: Sau khi tải về hoàn tất, SQL Server Installation Center sẽ mở lên. Chọn Installation, sau đó chọn New SQL Server stand-alone installation or add features to an existing installation.



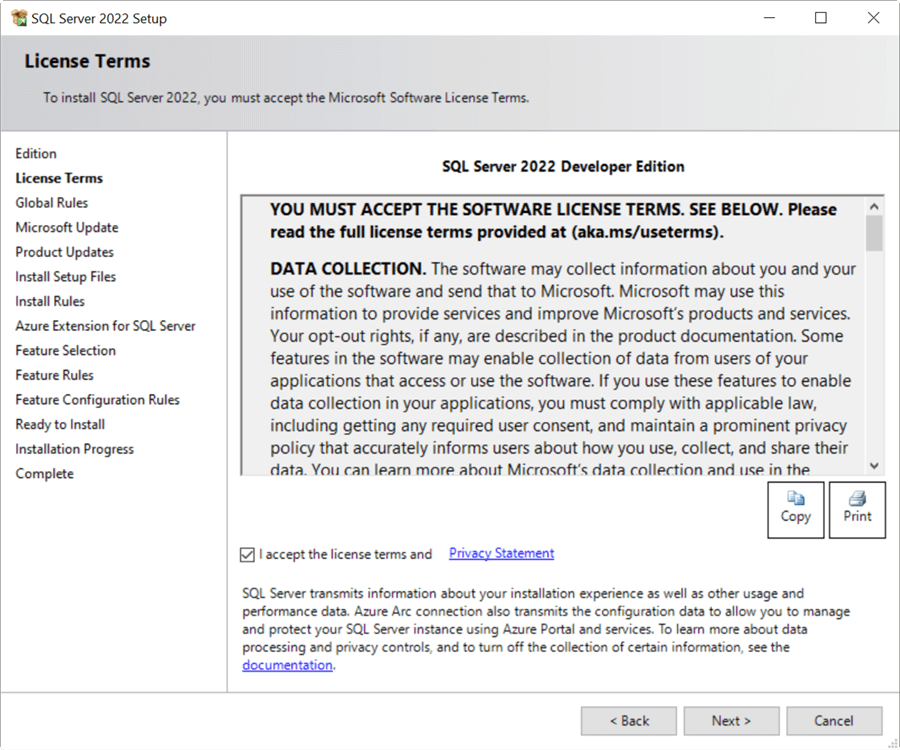


Bước 5 - Edition: Trên giao diện hiện ra, chọn Specify a free edition và chọn Developer và nhấn Next.



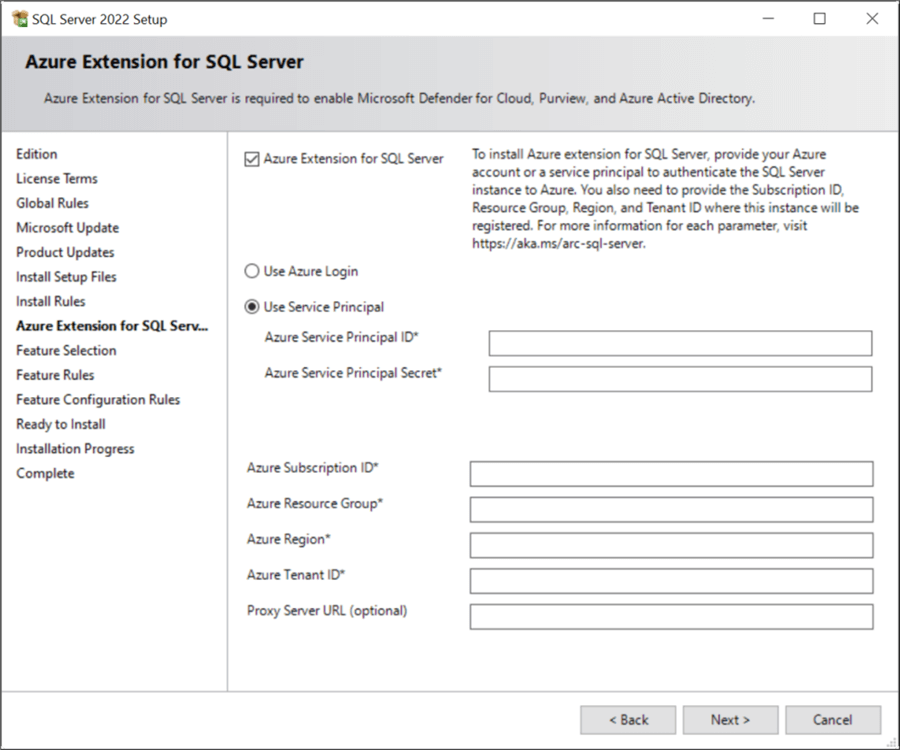


Bước 6 - License Terms: Tích vào ô I accept the license term and Privacy Statement và nhấn Next.



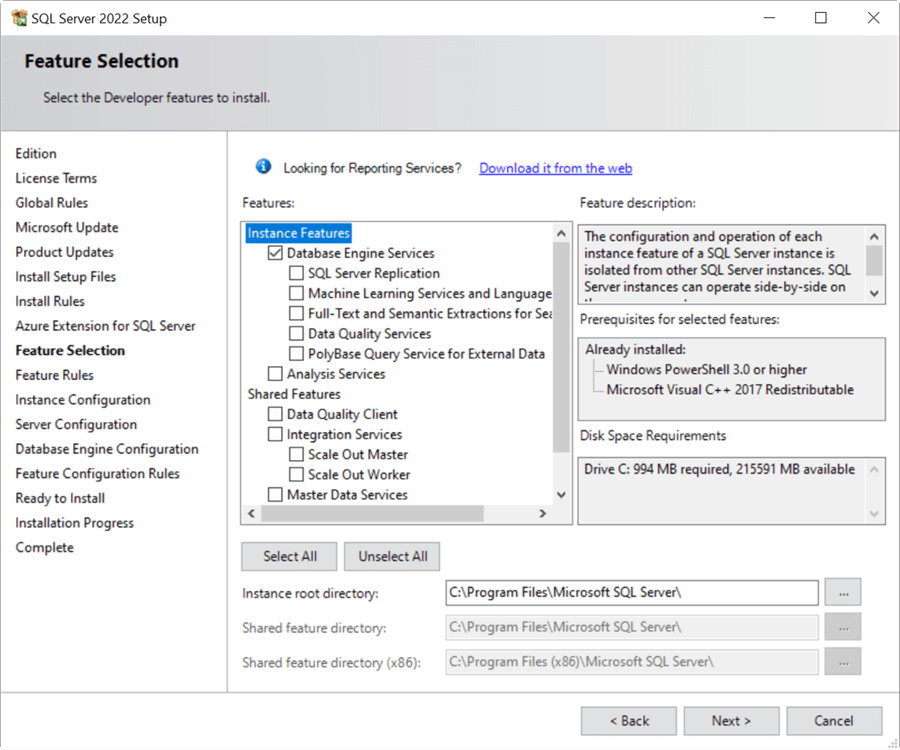


Bước 7 - Azure Extension for SQL Server: Sau khi qua bước Global Rules, Product Updates, Install Setup Files, Install Rules, sẽ tới bước Azure Extension for SQL Server. Hủy tích ở ô Azure Extension for SQL Server và nhấn Next



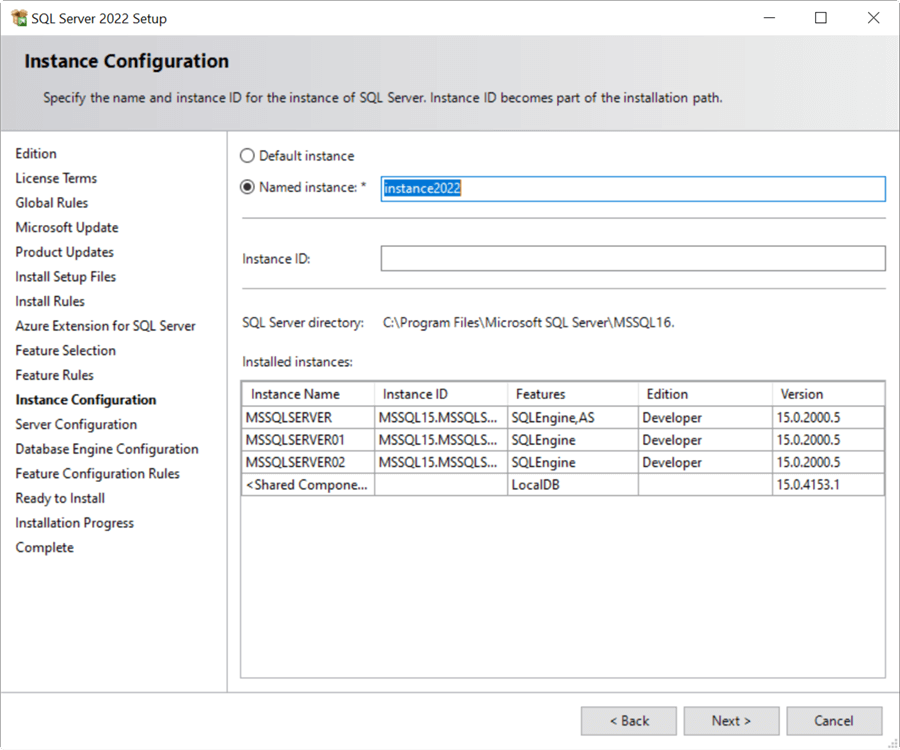


Bước 8 – Feature Selection: Chọn Database Engine Services và Analysis Services, sau đó nhấn Next



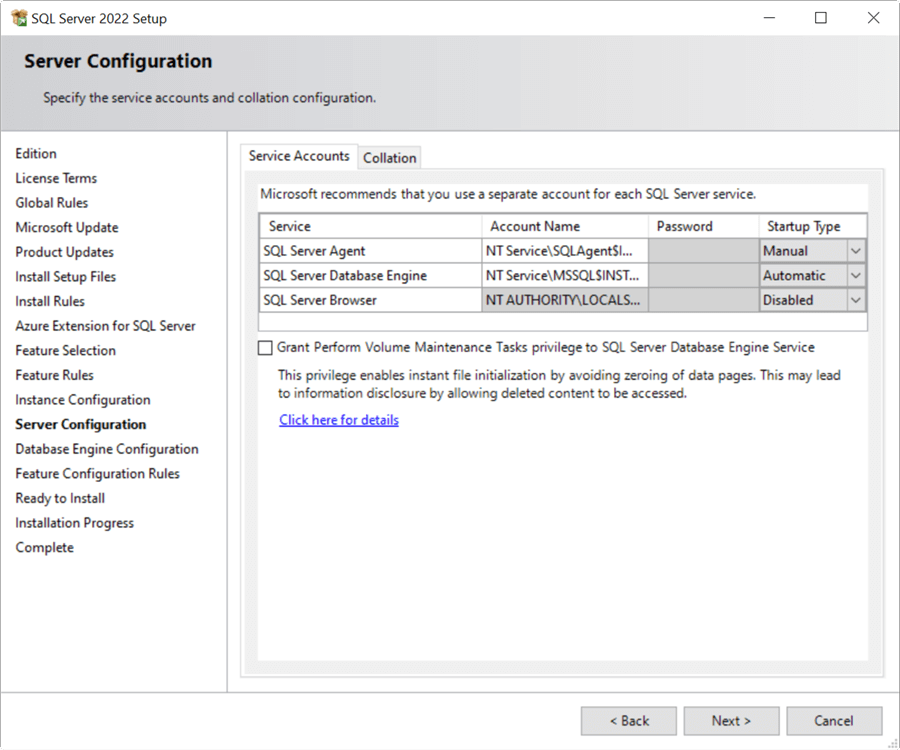


Bước 9 - Instance Configuration: Chọn Named instance và đặt tên là MSSQLDEV2025 và nhấn Next.



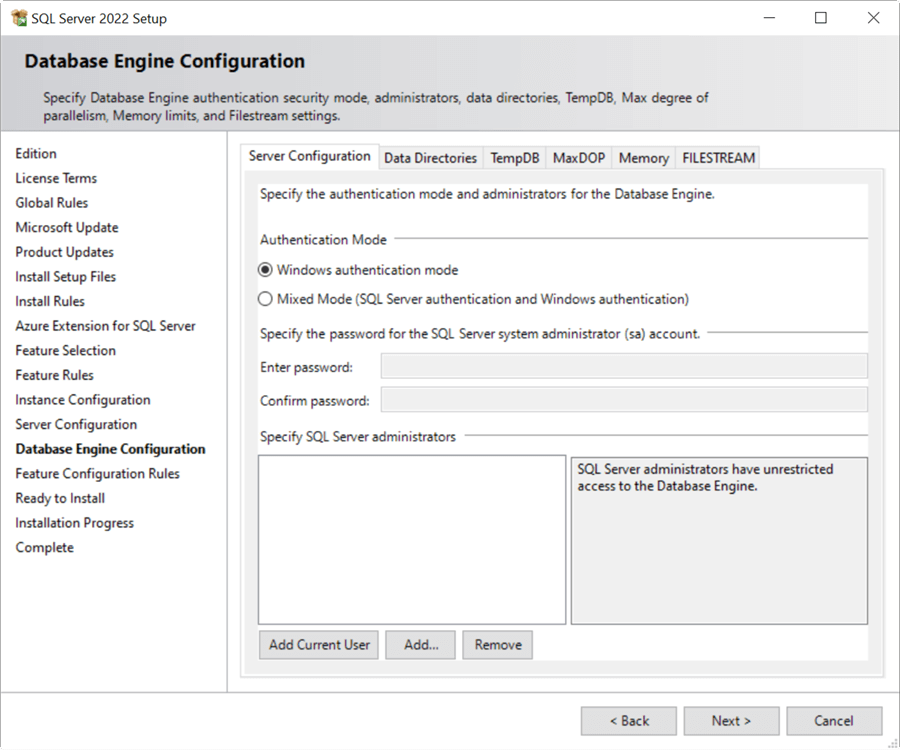


Bước 10 – Server Configuration: Chuyển tất cả Startup Type của tất cả Service thành Automatic và tích vào Grant Perform Volume Maintenance Tasks privilege to SQL Server Database Engine Service. Sau đó nhấn Next





Bước 11 - Database Engine Configuration: Tại tab Server Configuration, trong Authentication Mode, chọn Windows authentication mode, sau đó nhấn Add Current User và nhấn Next.





Bước 12 - Analysis Services Configuration: Tại tab Server Configuration, trong Server Mode, chọn Multipledimensional Mode, sau đó nhấn Add Current User và nhấn Next.

Bước 13 - Ready to Install: Sau khi mọi thứ được thiết lập, nhấn Install. Sau khi cài đặt xong thì nhấn Close để đóng trình cài đặt

### 2.2.2 Cài đặt Visual Studio 2022 Community với Extension SQL Server Intergration Services Projects 2022, Microsoft Analysis Services Projects

Bước 1: Đầu tiên, tải file cài đặt về từ trang Microsoft.com: <https://visualstudio.microsoft.com/>

Bước 2: Ở màn hình tiếp theo, nhấn vào Continue để bắt đầu cài đặt Visual Studio Installer

Bước 3 - Visual Studio Installer: Ở màn hình Installer, trong tab Workloads, chọn Data storage and processing và nhấn Install.

Bước 4: Sau khi cài đặt xong, mở Visual Studio, hoàn tất các thiết lập ban đầu. Tới màn hình Projects thì chọn Continue without code

Bước 5: Trong Visual Studio, trên thanh menu, chọn Extension > Manage extensions…, sau đó tìm kiếm SQL Server Integration Services Projects 2022 với Microsoft Analysis Services Projects. Sau đó người dùng tiến hành tải về và cài đặt theo hướng dẫn.

---to be continue---