ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

*

MÔN HỌC: HỆ THỐNG THÔNG TIN
PHỤC VỤ TRÍ TUỆ KINH DOANH

BÁO CÁO ĐỒ ÁN MÔN HỌC

Giảng viên hướng dẫn:
Hồ Thị Hoàng Vy
Nguyễn Ngọc Minh Châu
Tiết Gia Hồng

Nhóm thực hiện:
CLC_BI#11
Thành viên:
22127026 - Ôn Gia Bảo
22127091 - Phạm Mai Duyên
22127447 - Phùng Tố Uyên

Thành phố Hồ Chí Minh – 2025

A. Phân công công việc

Công việc	Phân công	Mức độ hoàn thành
Thiết kế CSDL cho DDS	Cả nhóm	100%
Viết script tạo CSDL cho DDS	Phùng Tố Uyên	100%
Thiết kế CSDL cho NDS	Cả nhóm	100%
Viết script tạo CSDL cho NDS	Phạm Mai Duyên	100%
Viết script tạo CSDL cho Stage	Phạm Mai Duyên, Phùng Tố Uyên	100%
Viết script tạo CSDL Metadata	Ôn Gia Bảo	100%
Viết xây dựng ETL pipeline: Source > Stage	Ôn Gia Bảo	100%
Viết xây dựng ETL pipeline: Stage > NDS, có log/kiểm soát lỗi	Ôn Gia Bảo	100%
Viết xây dựng ETL pipeline: NDS > DDS (Dim_Airline, Dim_Airport, Dim_Reason)	Phạm Mai Duyên	100%
Viết xây dựng ETL pipeline: NDS > DDS (Dim_Date, Dim_Time, Fact_Flight)	Phùng Tố Uyên	100%
Cài đặt tự động hoá cho ETL	Phạm Mai Duyên	100%
Xây dựng OLAP cube	Ôn Gia Bảo	100%
Viết truy vấn MDX	Ôn Gia Bảo	100%
Xây dựng dashboard Power BI	Phạm Mai Duyên, Phùng Tố Uyên	100%

Thực hiện data mining	Phùng Tố Uyên	100%
Quay video demo ETL Source to Stage	Phạm Mai Duyên	100%
Quay video demo ETL Stage to NDS	Ôn Gia Bảo	100%
Quay video demo ETL NDS to DDS	Phùng Tố Uyên	100%
Quay video demo thực hiện truy vấn MDX	Ôn Gia Bảo	100%
Quay video demo dashboard PowerBI	Phạm Mai Duyên	100%
Quay video demo data mining	Phùng Tố Uyên	100%

B. Data Modeling

I. Mô tả mô hình DDS

1. Mục tiêu nghiệp vụ

Người dùng nghiệp vụ (business user) cần phân tích hoạt động chuyến bay để:

- Theo dõi tổng số chuyến bay theo ngày, tuyến bay, hãng.
- Đánh giá tình trạng đúng giờ:
    - Tỷ lệ chuyến bay khởi hành/đến đúng giờ.
    - Mức độ trễ trung bình.
- Giám sát tình trạng hủy chuyến và chuyển hướng (diverted).
- Phân tích chi tiết nguyên nhân gây trễ/hủy.
- Phân tích theo nhiều chiều:
    - Hãng hàng không
    - Sân bay đi / đến
    - Ngày, tháng, quý, năm, mùa, cuối tuần
    - Khung giờ trong ngày
    - Loại lý do (Delay/Cancel)

Mục tiêu cuối cùng:

- Hỗ trợ ra quyết định: tối ưu lịch bay, phân bổ nguồn lực, cải thiện chất lượng dịch vụ.

- Theo dõi KPI chuẩn như On-Time Performance (OTP), tỉ lệ delay theo nguyên nhân, tỉ lệ cancel theo sân bay/hãng.

2. **Sự kiện (Event)**

Mỗi chuyến bay (Flight) được xem là một sự kiện trong hệ thống phân tích với

- Hãng hàng không xác định.
- Cặp sân bay xuất phát – sân bay đến xác định.
- Ngày và giờ khởi hành, ngày và giờ đến (theo lịch & thực tế).
- Tập các thông tin trễ, hủy, chuyển hướng, lý do trễ/hủy (nếu có).

Hệ thống ghi nhận lại toàn bộ dữ liệu này trong bảng Fact_Flight, mỗi chuyến bay là 1 dòng trong Fact_Flight.

3. **Bối cảnh sự kiện (Context of Event)**

Câu hỏi ngữ cảnh	Thành phần DDS	Ý nghĩa
Ai?	Dim_Airline	Hãng chịu trách nhiệm vận hành chuyến bay
Ở đâu?	Dim_Airport	Sân bay xuất phát (Origin) và sân bay đến (Destination)
Cái gì? (Event)	Fact_Flight	Bản thân chuyến bay: số hiệu, đuôi máy bay, delay, cancel,…
Khi nào?	Dim_Date, Dim_Time	Ngày/giờ cất cánh – hạ cánh, dùng để phân tích theo ngày, tháng, quý, giờ, mùa, cuối tuần
Tại sao?	Dim_Reason	Nhóm lý do giải thích việc trễ/hủy chuyến

4. **Đo lường (Measures) – Dữ kiện trong Fact**

Fact_Flight lưu trữ các chỉ số định lượng để phân tích:

- **Độ trễ tổng quát**
  - Total_depart_delay: số phút trễ tại thời điểm cất cánh.
  - Total_arrive_delay: số phút trễ tại thời điểm hạ cánh.
- **Độ trễ theo nguyên nhân**
  - Air_system_delay
  - Weather_delay
  - Security_delay
  - Late_aircraft_delay

- Airline_delay

→ Cho phép phân rã tổng delay theo từng nhóm nguyên nhân.

- **Trạng thái chuyến bay**
  - Canceled_Flag: 1 nếu chuyến bay bị hủy.
  - Diverted_Flag: 1 nếu chuyến bay chuyển hướng (không hạ cánh tại sân bay dự kiến).
- **Lý do hủy**
  - Canceled_Reason: khóa ngoại tham chiếu Dim_Reason (chỉ áp dụng khi Canceled_Flag = 1).

Những measures này phục vụ trực tiếp các phân tích:

- OTP (On-time performance)
- Tỷ lệ trễ theo hãng / sân bay / khung giờ / mùa.
- Tỷ lệ hủy theo lý do.
- Xác định điểm nghẽn.

5. Cấu trúc mô hình DDS (Star Schema)

a. Fact Table

Fact_Flight là trung tâm, chứa dữ liệu giao dịch (event), với:

Các khóa ngoại (FK) liên kết tới các bảng dimension:

- Departure_Date_ID, Arrival_Date_ID**:** tham chiếu Dim_Date
- Departure_Time_ID, Arrival_Time_ID: tham chiếu Dim_Time
- Origin_Airport_ID, Destination_Airport_ID: tham chiếu Dim_Airport
- Airline_ID: tham chiếu Dim_Airline
- Canceled_Reason: tham chiếu Dim_Reason

b. Dimension Tables

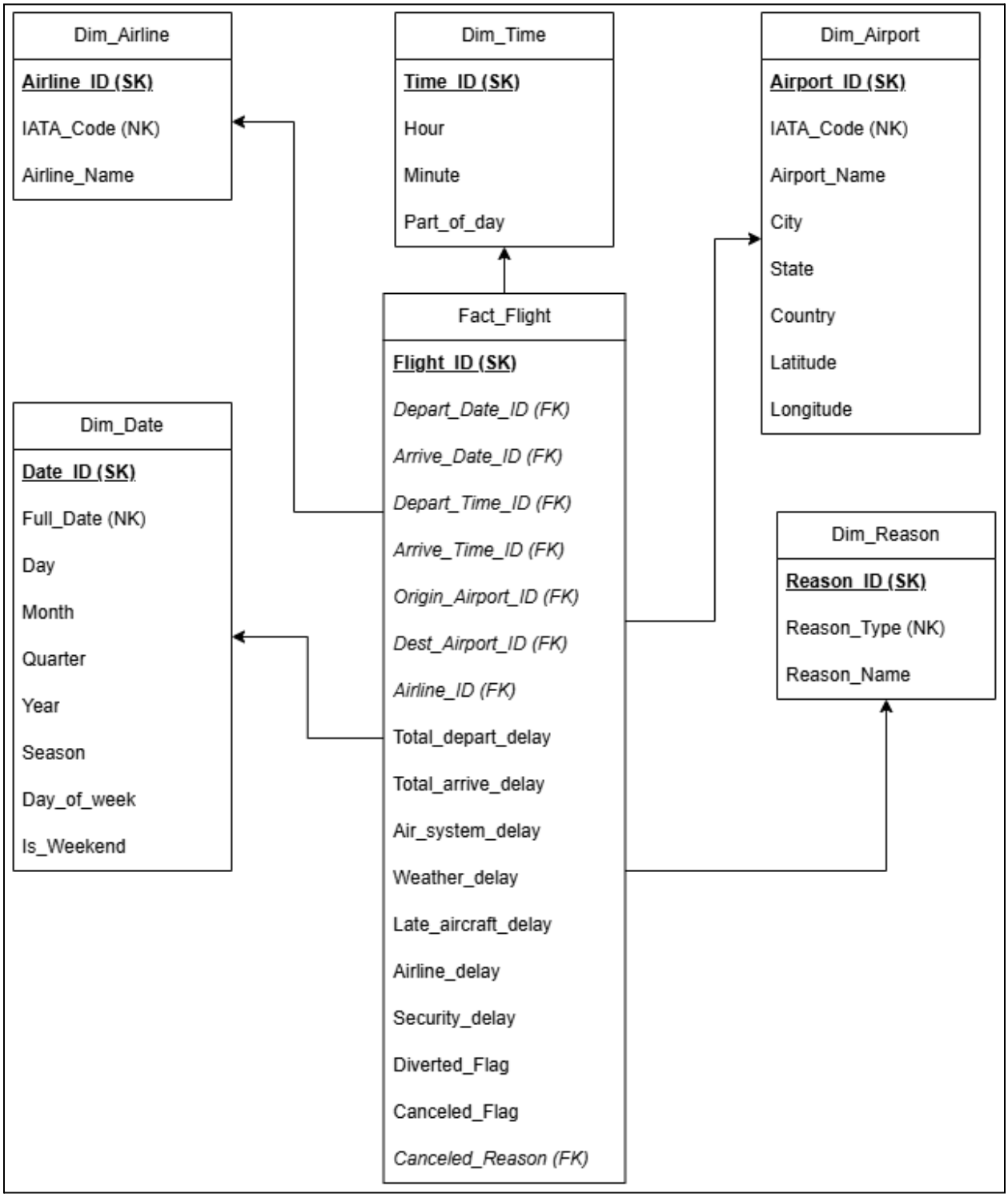
- **Dim_Date**
  - Mục đích: phân tích theo trục thời gian lịch.
  - Thuộc tính chính:
    - Full_Date, Day, Month, Quarter, Year
    - Season (Xuân/Hạ/Thu/Đông hoặc theo business rule)
    - Day_of_week (Thứ 2–CN)
    - Is_Weekend (0/1)
  - Ứng dụng:

        - So sánh theo tháng, quý, năm.
        - Phân tích ảnh hưởng mùa vụ đến delay/hủy.

- **Dim_Time**
    - Mục đích: phân tích theo khung giờ.
    - Thuộc tính:
        - Hour, Minute
        - Part_of_day (Night/Morning/Afternoon/Evening)
    - Ứng dụng:
        - Xác định "giờ cao điểm" dễ trễ.
        - So sánh hiệu suất giờ sáng vs giờ tối.
- **Dim_Airline**
    - Mục đích: phân tích theo hãng.
    - Thuộc tính:
        - IATA_Code, Airline_Name
    - Ứng dụng:
        - So sánh OTP giữa các hãng.
        - Xác định hãng có tỷ lệ trễ/hủy cao.
- **Dim_Airport**
    - Mục đích: phân tích theo sân bay/vùng địa lý.
    - Thuộc tính:
        - IATA_Code, Airport_Name, City, Country, Latitude, Longitude
    - Ứng dụng:
        - Tìm sân bay thường xuyên bị delay.
        - Phân tích theo khu vực (bắc/nam, nội địa/quốc tế,…).
- **Dim_Reason**
    - Mục đích: chuẩn hóa danh mục lý do trễ/hủy.
    - Thuộc tính:
        - Reason_Type (Delay / Cancel)
        - Reason_Name (Weather, NAS, Security, Carrier,…)

- Ứng dụng:
  - Thống kê số chuyến trễ/hủy theo từng nhóm nguyên nhân.
  - Hỗ trợ báo cáo chuẩn hóa, không phụ thuộc text tự do.

6. Lược đồ cơ sở dữ liệu DDS



II. Mô tả mô hình NDS

1. Mô tả mô hình NDS (Normalized Data Store)

1.1. Mục tiêu thiết kế

Mô hình **NDS (Normalized Data Store)** được xây dựng nhằm lưu trữ dữ liệu đã được làm sạch, chuẩn hóa và tổ chức có quan hệ chặt chẽ giữa các thực thể.
NDS đóng vai trò trung gian giữa **hệ thống nguồn (Source System)** và **tầng phân tích (DDS – Dimensional Data Store)**.
Mục tiêu của NDS là đảm bảo:

- Dữ liệu **nhất quán và không dư thừa**.
- Các quan hệ khóa ngoại thể hiện **mối liên kết thực giữa các thực thể**.
- Chuẩn bị dữ liệu **sẵn sàng để chuyển đổi sang mô hình chiều (Star Schema)** tại tầng DDS.

1.2. Cấu trúc các bảng trong NDS

Hệ thống NDS gồm 4 bảng chính:

Nhóm	Bảng	Chức năng
Sân bay & hãng bay	**NDS_Airport, NDS_Airline**	Lưu trữ thông tin chi tiết về sân bay và hãng hàng không.
Lý do chậm/hủy	**NDS_Reason**	Mô tả các loại lý do chậm hoặc hủy chuyến.
Chuyến bay	**NDS_Flight**	Lưu thông tin chi tiết về mỗi chuyến bay (thời gian, trễ, lý do, v.v.).

1.3. Mối quan hệ giữa các bảng

- **NDS_Flight là bảng trung tâm, liên kết với các bảng khác qua khóa ngoại:**
  - Airline (FK) → NDS_Airline.IATA_Code
  - Origin_Airport (FK) → NDS_Airport.IATA_Code
  - Dest_Airport (FK) → NDS_Airport.IATA_Code
  - Canceled_Reason (FK) → NDS_Reason.Reason_Type

Cấu trúc này đảm bảo:

- Dữ liệu có **tính toàn vẹn quan hệ**.
- Giảm thiểu **trùng lặp địa lý và hãng bay**.
- Dễ dàng **tổng hợp, phân tích, và chuyển hóa sang DDS**.

1.4. Đánh giá dạng chuẩn

Bảng: NDS_Airport

- **Lược đồ:**
  NDS_Airport(**IATA_Code**, Airport_Name, City, State, Country, Latitude, Longitude, Created, Modified)
- **Phụ thuộc hàm:**
  IATA_Code → tất cả thuộc tính còn lại
- **Khóa:**
  IATA_Code
- **Đánh giá dạng chuẩn:**
  - Thuộc tính nguyên tử → 1NF.
  - Mọi thuộc tính không khóa phụ thuộc đầy đủ vào khóa → 2NF.
  - Không có phụ thuộc bắc cầu → 3NF.
  - Tất cả phụ thuộc hàm có vế trái là khóa → BCNF.

Bảng: NDS_Airline

- **Lược đồ:**
  NDS_Airline(**IATA_Code**, Airline_Name, Created, Modified)
- **Phụ thuộc hàm:**
  IATA_Code→ tất cả thuộc tính còn lại
- **Khóa:**
  IATA_Code
- **Đánh giá dạng chuẩn:**
  - Thuộc tính nguyên tử → 1NF.
  - Mọi thuộc tính không khóa phụ thuộc đầy đủ vào khóa → 2NF.
  - Không có phụ thuộc bắc cầu → 3NF.
  - Tất cả phụ thuộc hàm có vế trái là khóa → BCNF.

Bảng: NDS_Reason

- **Lược đồ:**
  NDS_Reason(**Reason_Type**, Reason_Name, Created, Modified)
- **Phụ thuộc hàm:**
  Reason_Type →tất cả thuộc tính còn lại
- **Khóa:**
  Reason_ID
- **Đánh giá dạng chuẩn:**
  - Thuộc tính nguyên tử → 1NF.

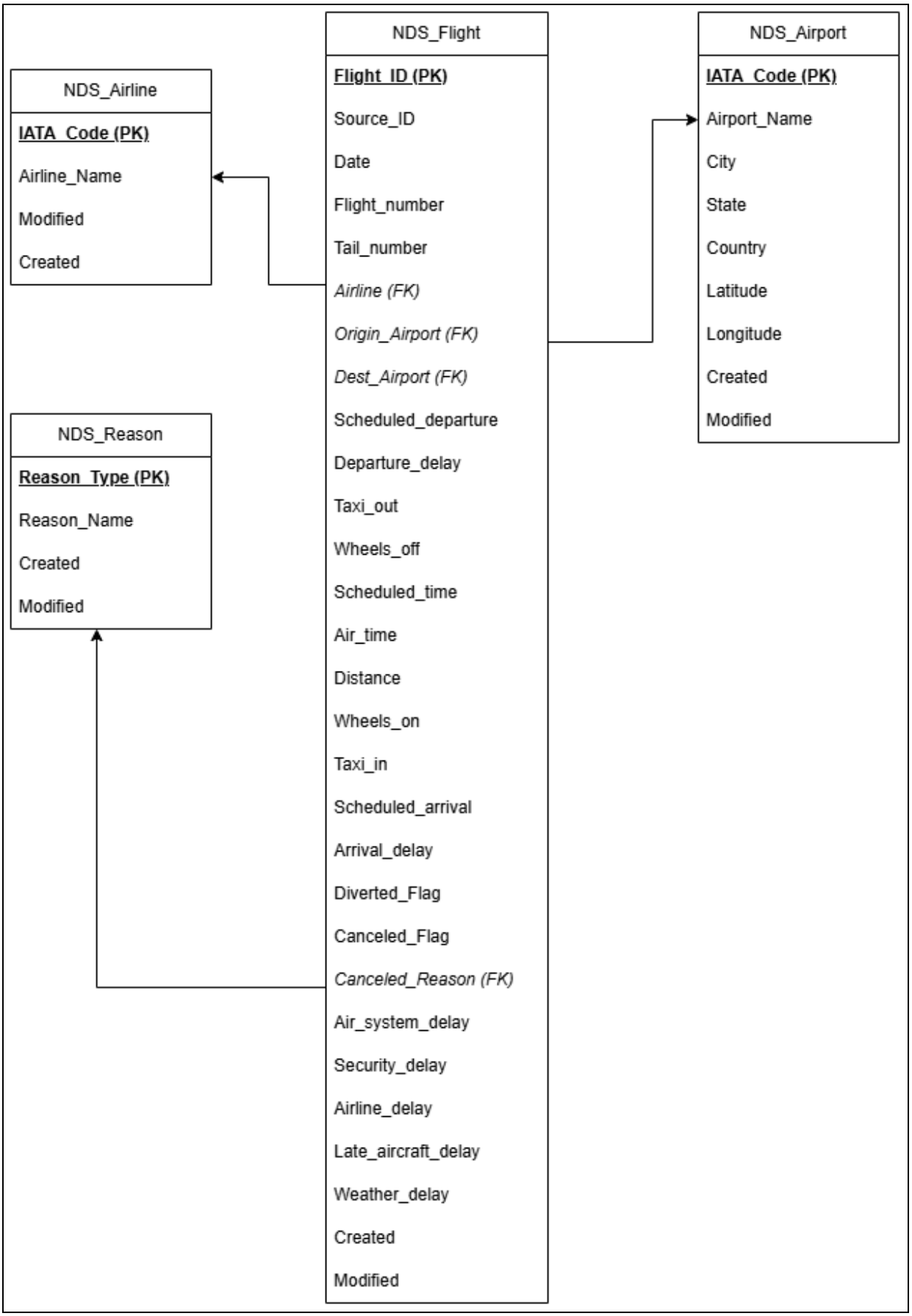
- Mọi thuộc tính không khóa phụ thuộc đầy đủ vào khóa → 2NF.
- Không có phụ thuộc bắc cầu → 3NF.
- Tất cả phụ thuộc hàm có vế trái là khóa → BCNF.

Bảng: NDS_Flight

- **Lược đồ:**
  NDS_Flight(**Flight_ID**, Source_ID, Date, Flight_number, Tail_number, Airline,
  Origin_Airport, Dest_Airport, Scheduled_departure, Departure_delay, Taxi_out,
  Wheels_off, Scheduled_time, Air_time, Distance, Wheels_on, Taxi_in, Scheduled_arrival, Arrival_delay, Diverted_Flag, Canceled_Flag, Canceled_Reason, Air_system_delay, Security_delay, Airline_delay, Late_aircraft_delay, Weather_delay, Created, Modified)
- **Phụ thuộc hàm:**
  - Flight_ID → tất cả các thuộc tính còn lại.
  - (Date, Flight_number, Origin_Airport, Dest_Airport, Scheduled_departure) → tất cả các thuộc tính còn lại.
- **Khóa:**
  - Flight_ID
  - Date, Flight_number, Origin_Airport, Dest_Airport, Scheduled_departure
- **Đánh giá dạng chuẩn:**
  - Các thuộc tính nguyên tử → 1NF
  - Mọi thuộc tính mô tả chuyến bay phụ thuộc đầy đủ vào khoá → 2NF
  - Không có phụ thuộc bắc cầu → 3NF
  - Tất cả phụ thuộc hàm có vế trái là khóa → BCNF

Tất cả các bảng trong mô hình NDS đều đạt đến BCNF, đảm bảo dữ liệu không dư thừa, duy trì toàn vẹn khóa ngoại, và tối ưu cho việc chuyển đổi sang mô hình chiều (DDS).

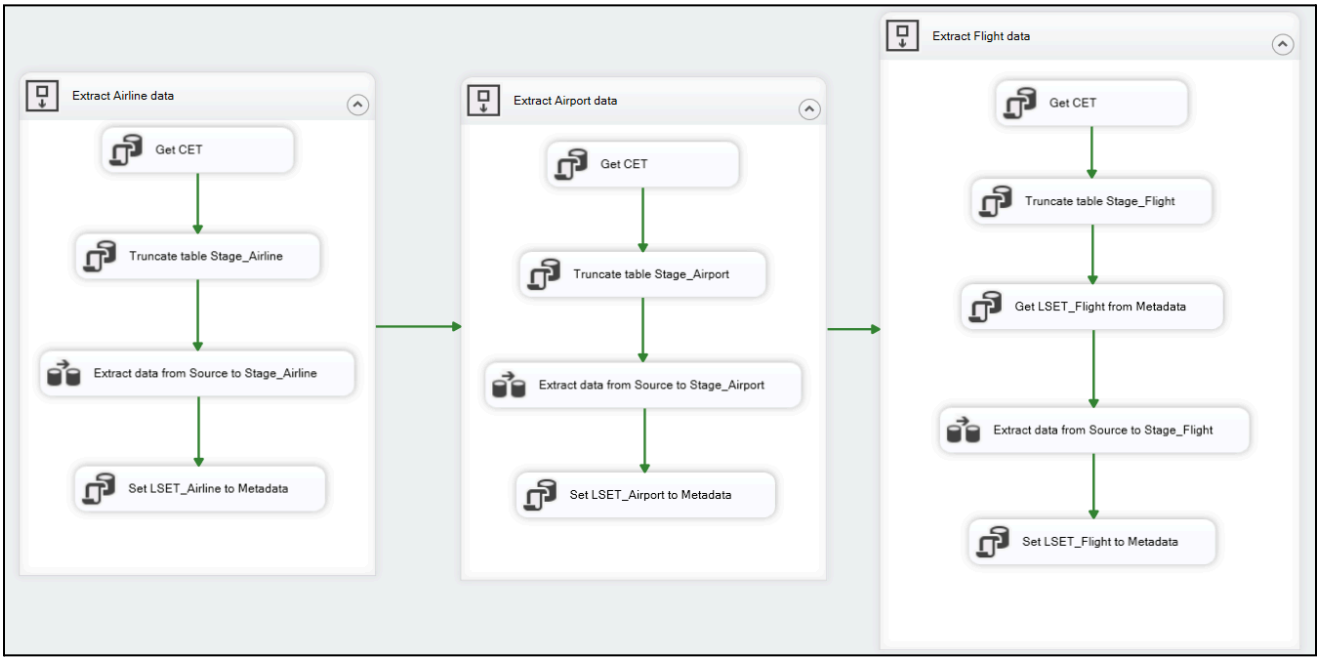
2. Lược đồ CSDL NDS


NDS_Flight
Flight_ID (PK)
Source_ID
Date
Flight_number
Tail_number
Airline (FK)
Origin_Airport (FK)
Dest_Airport (FK)
Scheduled_departure
Departure_delay
Taxi_out
Wheels_off
Scheduled_time
Air_time
Distance
Wheels_on
Taxi_in
Scheduled_arrival
Arrival_delay
Diverted_Flag
Canceled_Flag
Canceled_Reason (FK)
Air_system_delay
Security_delay
Airline_delay
Late_aircraft_delay
Weather_delay
Created
Modified
NDS_Airline
IATA_Code (PK)
Airline_Name
Modified
Created
NDS_Airport
IATA_Code (PK)
Airport_Name
City
State
Country
Latitude
Longitude
Created
Modified
NDS_Reason
Reason_Type (PK)
Reason_Name
Created
Modified

C. ETL Process

I. Source to Stage

1. Control Flow



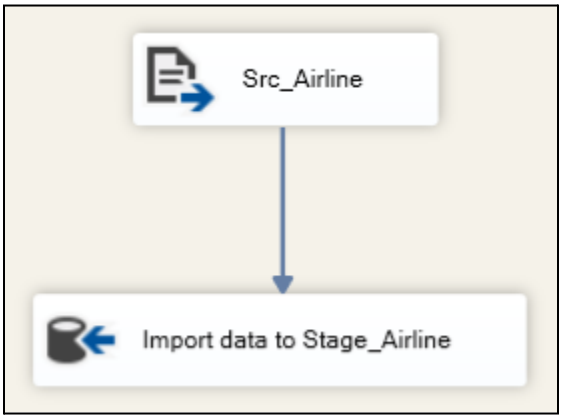
Được chia làm 3 luồng chính trong mỗi Sequence container, thực thi theo thứ tự:

- Container: Extract Airline data
    - Mục tiêu: Trích xuất dữ liệu hãng hàng không (Airline) từ nguồn về bảng Stage.
    - Luồng xử lý:
        - Get CET: Lấy Current Extraction Time (thời điểm ETL hiện tại). Thường gán giá trị này vào biến User::CET.
        - Truncate table Stage_Airline: Xóa toàn bộ dữ liệu cũ trong bảng tạm Stage_Airline trước khi nạp dữ liệu mới.
        - Extract data from Source to Stage_Airline: Dùng Data Flow Task để trích xuất dữ liệu từ nguồn (Source) sang bảng Stage_Airline.
        - Set LSET_Airline to Metadata: Sau khi load xong, cập nhật giá trị Last Successful Extraction Time (LSET) cho bảng Airline vào metadata table, để các lần chạy sau có thể dùng làm mốc trích xuất gia tăng.
- Container: Extract Airport data
    - Mục tiêu: Trích xuất dữ liệu sân bay (Airport) từ nguồn về bảng Stage.
    - Luồng xử lý:
        - Get CET: Lấy thời điểm chạy ETL hiện tại.

        - Truncate table Stage_Airport: Xóa dữ liệu cũ trong bảng Stage_Airport.
        - Extract data from Source to Stage_Airport (Data Flow Task): nạp dữ liệu Airport từ nguồn sang Stage.
        - Set LSET_Airport to Metadata: Ghi lại thời gian trích xuất cuối cùng vào bảng metadata.
- Container: Extract Flight data
    - Mục tiêu: Trích xuất dữ liệu chuyến bay (Flight) từ nguồn về bảng Stage.
    - Luồng xử lý:
        - Get CET: Lấy thời điểm hiện tại của lần ETL.
        - Truncate table Stage_Flight: Xóa dữ liệu cũ trong Stage_Flight.
        - Get LSET_Flight from Metadata: Đọc giá trị Last Successful Extraction Time trước đó từ bảng metadata, để chỉ lấy dữ liệu mới hơn mốc đó.
        - Extract data from Source to Stage_Flight (Data Flow Task): truy vấn dữ liệu từ nguồn (chỉ lấy record có LastModified > LSET_Flight) và ghi vào Stage_Flight.
        - Set LSET_Flight to Metadata: Cập nhật lại giá trị LSET bằng thời điểm CET sau khi ETL thành công.

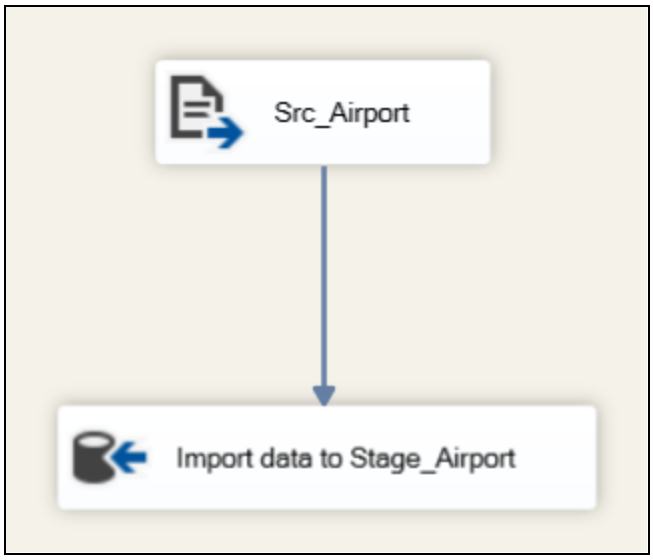
2. Data Flow

a. Extract data from Source to Stage_Airline



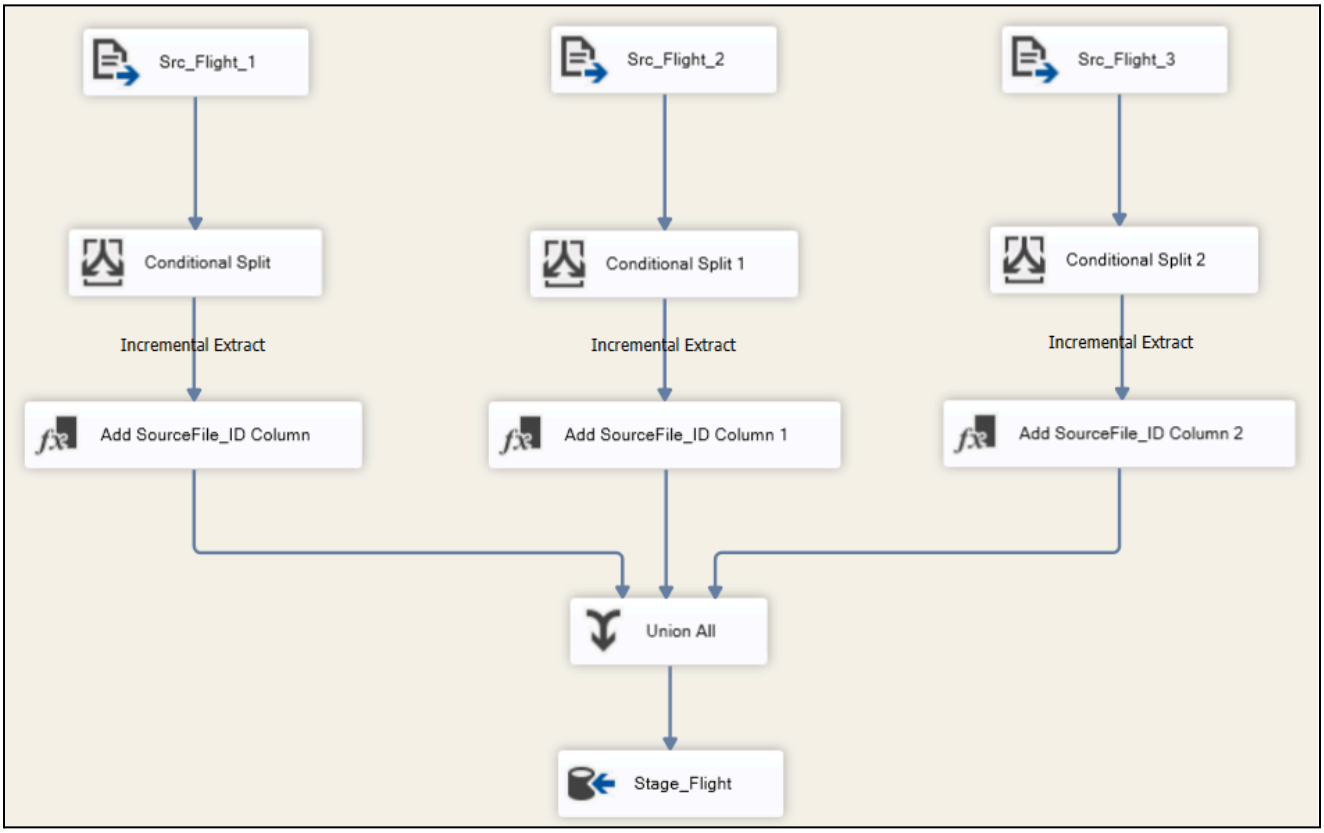
- Src_Airline (Flat File Source component):
  Kết nối đến file nguồn dữ liệu (*airlines.csv*) để đọc dữ liệu Airlines.
- Import data to Stage_Airline (OLE DB Destination component):
  Ghi dữ liệu nhận được từ Src_Airline vào bảng đích Stage_Airline trong database Stage_DB.

b. Extract data from Source to Stage_Airport



- Src_Airport (Flat File Source component):
  Kết nối đến file nguồn dữ liệu (*filtered_airport.csv*) để đọc dữ liệu Airports.
- Import data to Stage_Airline (OLE DB Destination component):
  Ghi dữ liệu nhận được từ Src_Airport vào bảng đích Stage_Airport trong database Stage_DB.

c. Extract data from Source to Stage_Flight



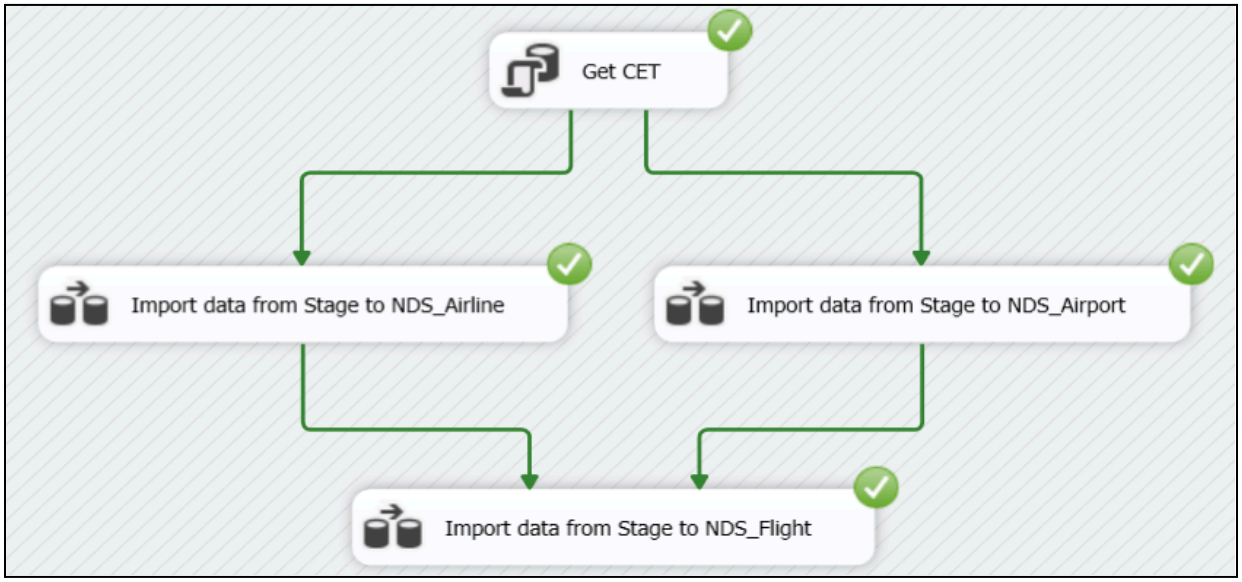
- Src_Flight_x:

Kết nối đến 3 file nguồn dữ liệu (*filtered_flights_x.csv*) để đọc dữ liệu Flights.

- Conditional Split:
  Thực hiện incremental extract: Chỉ load các dòng dữ liệu có ngày khởi tạo (CREATED) hoặc ngày chỉnh sửa (MODIFIED) nằm trong khoảng (LSET, CET].
- Add SourceFile_ID Column:
  Thêm cột SourceFile_ID chứ giá trị số nguyên vào các dòng dữ liệu để đánh dấu dữ liệu thuộc nguồn của file nào:
  - 1: filtered_flights_1.csv
  - 2: filtered_flights_2.csv
  - 3: filtered_flights_3.csv
- Union All:
  - Dữ liệu từ TẤT CẢ các luồng Derived Column (đã được gán SourceFile_ID) được đưa vào chung một thành phần Union All.
  - Mục đích: Thành phần này "xếp chồng" (append) tất cả các dòng từ các nguồn khác nhau lên nhau, tạo thành một luồng dữ liệu duy nhất, khổng lồ.
- Stage_Flight (Destination):
  Tải toàn bộ dữ liệu đã được gộp vào bảng Stage_Flight trong cơ sở dữ liệu.

II. Stage to NDS

1. Control Flow



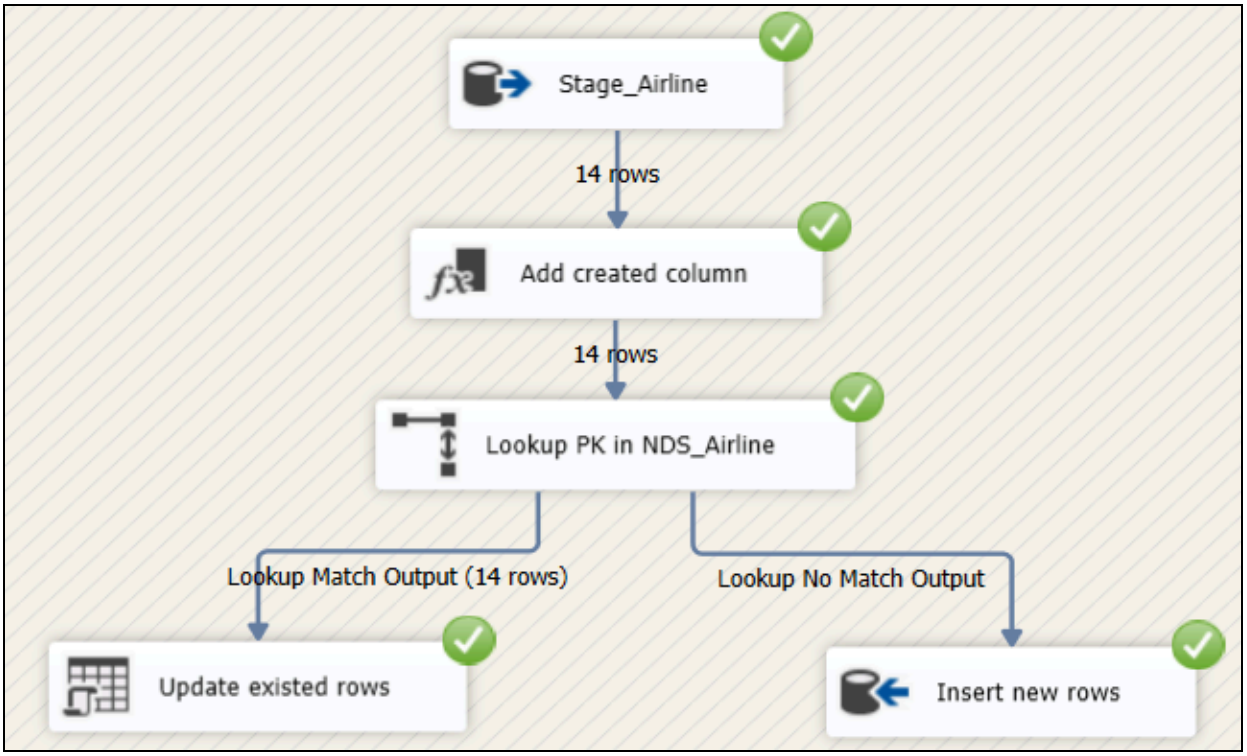
- Bước 0:
  - Trước khi thực hiện ETL cho giai đoạn từ Stage vào NDS, cần chạy riêng đoạn script insert dữ liệu vào bảng NDS_Reason.
  - Mục đích là để bảng dữ liệu giao dịch phụ thuộc (NDS_Flight) có thể tham chiếu các mã lý do này thông qua khóa ngoại.

```sql
INSERT INTO NDS_Reason (Reason_Type, Reason_Name, Created)
VALUES
('A', 'Airline/Carrier', GETDATE()),
('B', 'Weather', GETDATE()),
('C', 'National Air System', GETDATE()),
('D', 'Security', GETDATE());
```

- Bước 1:
  - Tác vụ "Get CET" có chức năng chính là xác định và lưu trữ mốc thời gian mà quá trình trích xuất dữ liệu hiện tại bắt đầu.
  - Thiết lập một biến số toàn cục chứa giá trị thời gian chính xác khi package bắt đầu chạy.
  - Mốc thời gian này (CET) sau đó sẽ được sử dụng trong các tác vụ nhập dữ liệu tiếp theo (NDS_Airline, NDS_Airport, NDS_Flight) để thêm cột "CREATED" (đánh dấu mốc thời gian mà dữ liệu này được thêm vào bảng) hoặc "MODIFIED" (đánh dấu mốc thời gian gần nhất mà dòng dữ liệu này được cập nhật trong bảng) phục vụ cho việc đổ dữ liệu vào DDS (dùng incremental extract).
- Bước 2:
  - Sau khi "Get CET" hoàn tất, luồng điều khiển phân nhánh thành hai tác vụ nhập dữ liệu chạy song song (đồng thời):
    - Import data from Stage to NDS_Airline: Nhập dữ liệu về Hãng hàng không từ Stage_Airline vào bảng NDS_Airline.
    - Import data from Stage to NDS_Airport: Nhập dữ liệu về Sân bay từ Stage_Airport vào bảng NDS_Airport.
  - Vì cả hai luồng dữ liệu Airline và Airport không phụ thuộc vào nhau tại thời điểm nhập ban đầu nên có thể thiết lập chạy song song để tiết kiệm thời gian.
- Bước 3:
  - Chỉ khi cả hai tác vụ nhập dữ liệu song song (Airline và Airport) hoàn thành thành công (và sau khi chạy script insert dữ liệu cho bảng NDS_Reason), tác vụ cuối cùng mới được kích hoạt:
    - Import data from Stage to NDS_Flight: Nhập dữ liệu về Chuyến bay (Flight) từ Stage vào bảng/kho NDS_Flight.
  - Sự phụ thuộc: Việc nhập dữ liệu NDS_Flight cần dữ liệu NDS_Airline và NDS_Airport đã được nhập xong vì thông tin chuyến bay chứa các khóa ngoại tham chiếu đến hãng hàng không phụ trách chuyến bay và sân bay đi/đến.

2. Data Flow

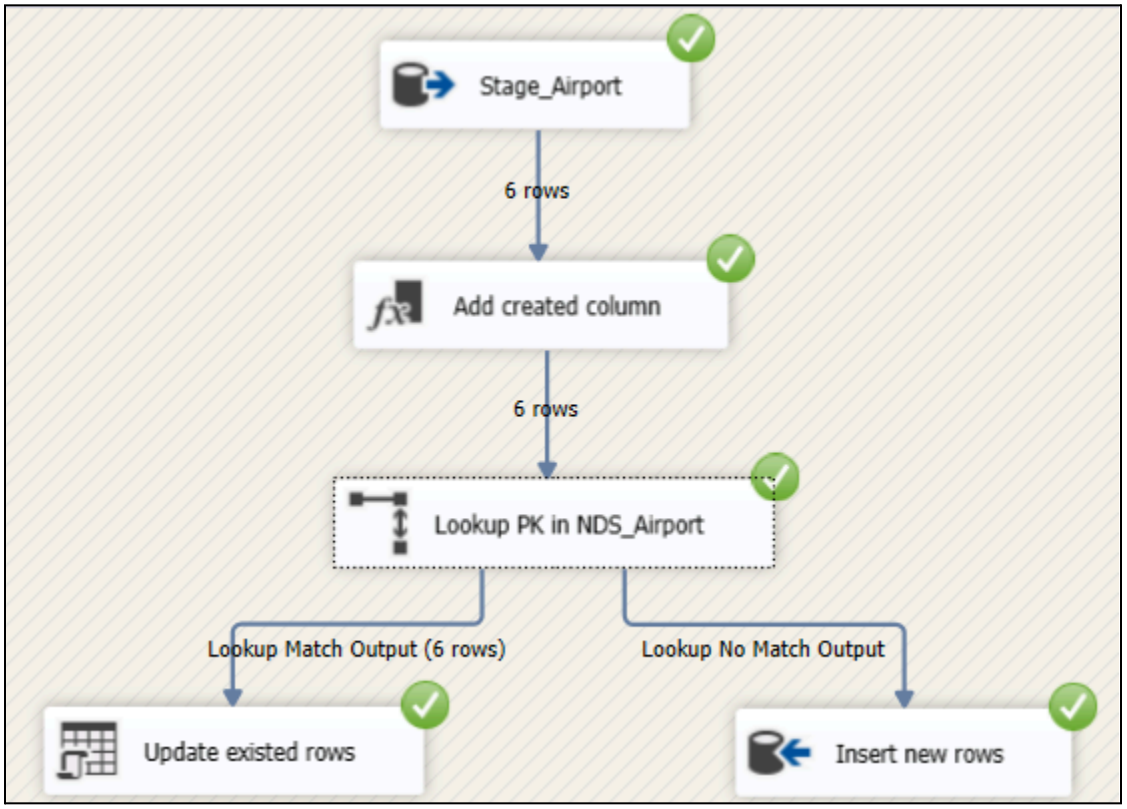
a. Import data from Stage to NDS_Airline



- Stage_Airline: Nguồn đầu vào, lấy dữ liệu từ bảng Stage_Airline.
- Add created column:
    - Mô tả: Thêm một cột mới (CREATED) vào luồng dữ liệu. Cột này được đặt giá trị là thời gian trích xuất hiện tại (CET) được lấy từ Control Flow.
    - Mục đích: Ghi lại thời điểm dữ liệu được xử lý. Cột này được ánh xạ vào trường Created trong bảng đích NDS_Airline để phục vụ việc incremental extract vào DDS.
- Lookup PK in NDS_Airline:
    - Mô tả: Đây là bước quan trọng nhất để xác định xem các bản ghi từ Stage đã tồn tại trong bảng đích NDS_Airline hay chưa. Nó thực hiện việc tra cứu dựa trên Khoá Chính (là IATA_Code).
    - Mục đích: Phân tách luồng dữ liệu thành:
        - Các bản ghi đã tồn tại (cần cập nhật).
        - Các bản ghi mới (cần chèn).
- Sau bước Lookup sẽ được chia thành 2 luồng đích riêng biệt:
    - Luồng Lookup Match Output:
        - Update existed rows: OLE DB Command.
        - Các bản ghi đã được tìm thấy trong NDS_Airline (dựa trên PK). Dữ liệu cũ trong NDS_Airline sẽ được cập nhật bằng dữ liệu mới từ Stage.
    - Luồng Lookup No Match Output:
        - Insert new rows: OLE DB Destination.

        - Các bản ghi không được tìm thấy trong NDS_Airline. Đây là các bản ghi mới hoàn toàn và sẽ được chèn vào bảng NDS_Airline.

b. Import data from Stage to NDS_Airport

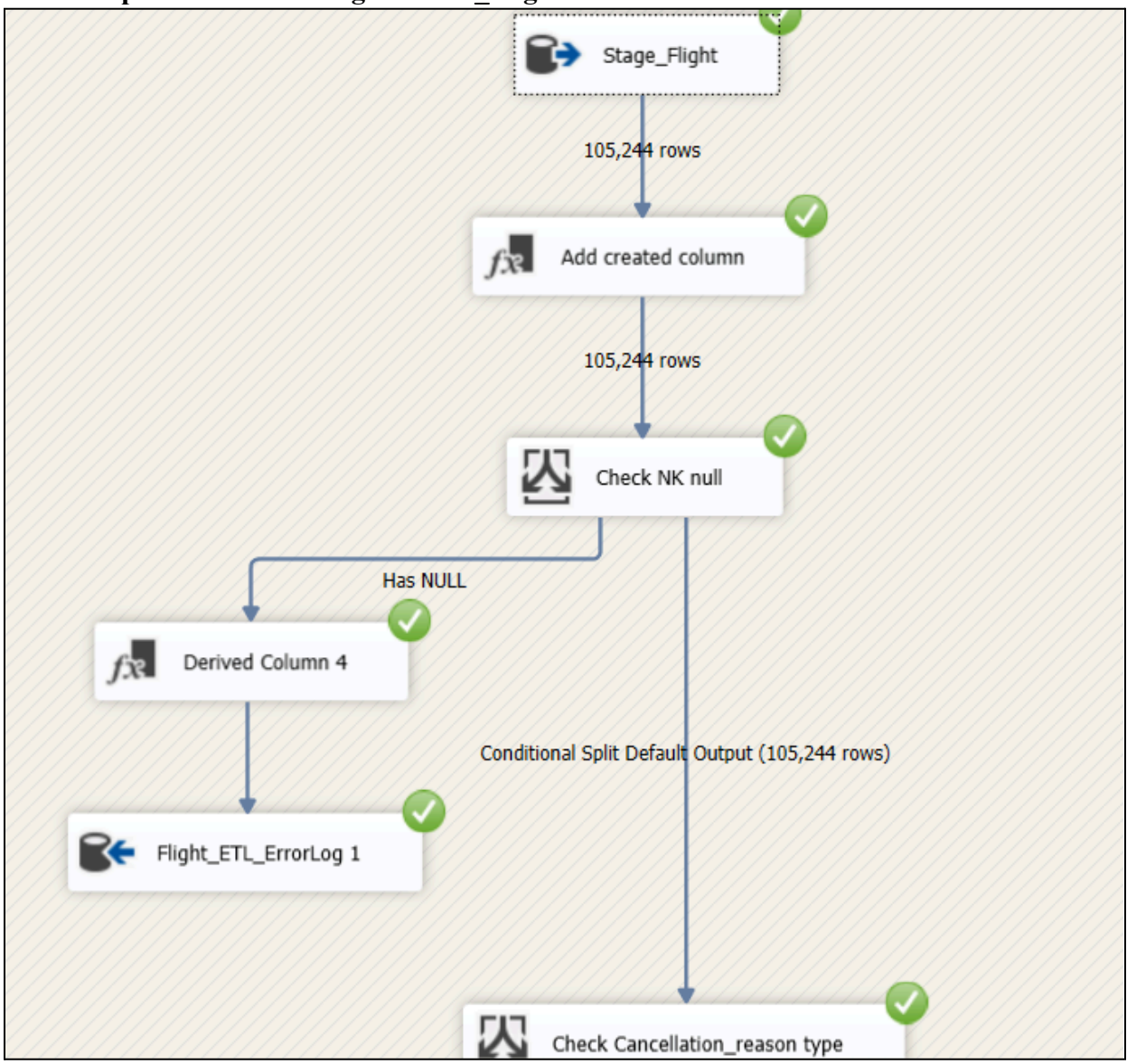


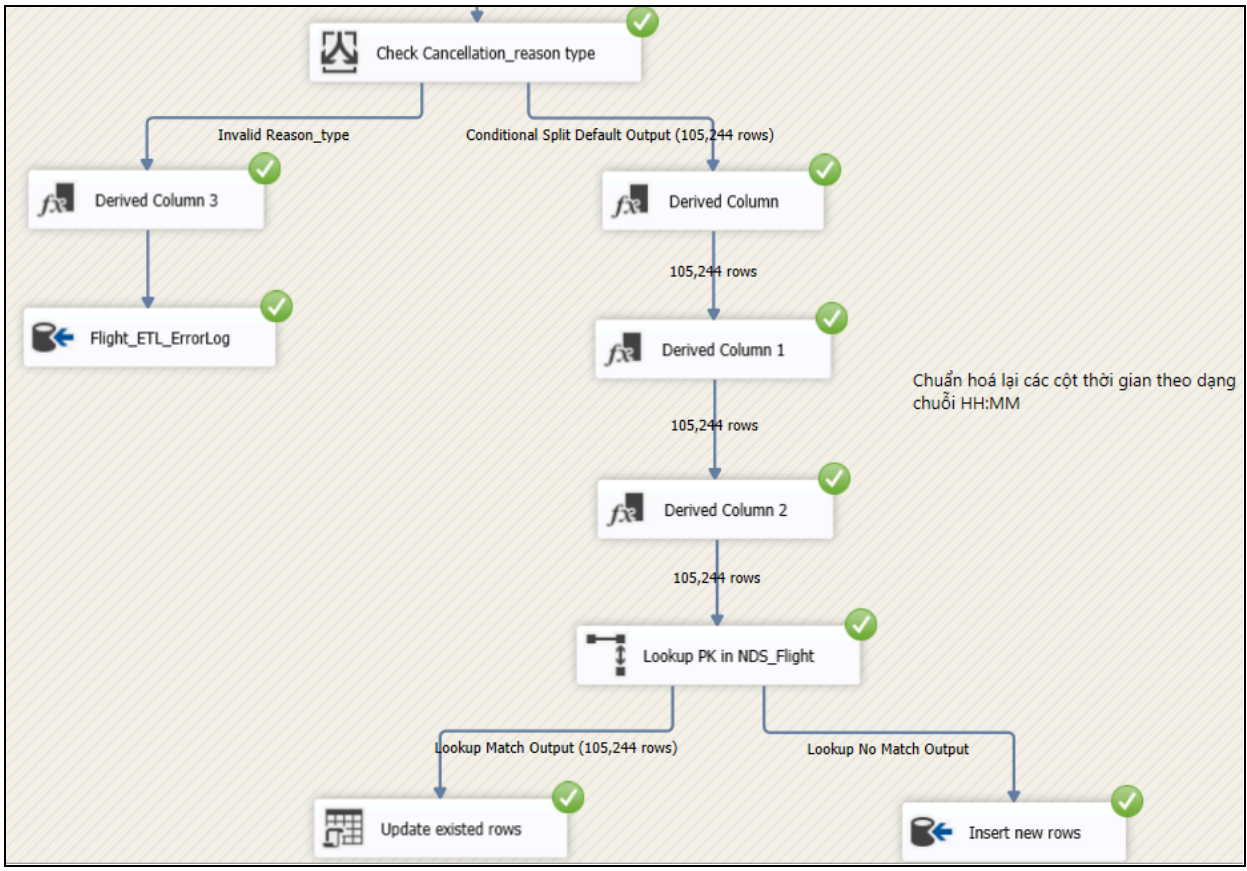
- Stage_Airport: Nguồn đầu vào, lấy dữ liệu từ bảng Stage_Airport.
- Add created column:
    - Mô tả: Thêm một cột mới (CREATED) vào luồng dữ liệu. Cột này được đặt giá trị là thời gian trích xuất hiện tại (CET) được lấy từ Control Flow.
    - Mục đích: Ghi lại thời điểm dữ liệu được xử lý. Cột này được ánh xạ vào trường Created trong bảng đích NDS_Airport để phục vụ việc incremental extract vào DDS.
- Lookup PK in NDS_Airport:
    - Mô tả: Đây là bước quan trọng nhất để xác định xem các bản ghi từ Stage đã tồn tại trong bảng đích NDS_Airport hay chưa. Nó thực hiện việc tra cứu dựa trên Khoá Chính (là IATA_Code).
    - Mục đích: Phân tách luồng dữ liệu thành:
        - Các bản ghi đã tồn tại (cần cập nhật).
        - Các bản ghi mới (cần chèn).
- Sau bước Lookup sẽ được chia thành 2 luồng đích riêng biệt:
    - Luồng Lookup Match Output:
        - Update existed rows: OLE DB Command.

- Các bản ghi đã được tìm thấy trong NDS_Airport (dựa trên PK). Dữ liệu cũ trong NDS_Airport sẽ được cập nhật bằng dữ liệu mới từ Stage.

- Luồng Lookup No Match Output:
  - Insert new rows: OLE DB Destination.
  - Các bản ghi không được tìm thấy trong NDS_Airport. Đây là các bản ghi mới hoàn toàn và sẽ được chèn vào bảng NDS_Airport.

c. Import data from Stage to NDS_Flight





- Stage_Flight: Nguồn dữ liệu đầu vào, lấy từ bảng Stage_Flight.
- Add created column: Thêm cột ghi lại thời điểm xử lý (tương tự như NDS_Airline và NDS_Airport, dùng giá trị CET).
- Check NK null:
    - Mô tả: Kiểm tra xem Khóa Tự nhiên (Natural Key - NK) của bản ghi chuyến bay (bao gồm: Date, Flight_number, Scheduled_departure, Origin_airport, Destination_airport) có bị NULL hay không. Khóa này là bắt buộc để xác định bản ghi là duy nhất.
    - Luồng lỗi (Has NULL): Nếu có ít nhất 1 thành phần trong Khóa Tự nhiên bị NULL, các bản ghi này được chuyển qua Derived Column 4 (thêm thông tin lỗi) và ghi vào bảng *Flight_ETL_ErrorLog*. Đây là cơ chế xử lý lỗi (Error Handling) và loại bỏ dữ liệu kém chất lượng khỏi luồng chính.

```sql
CREATE TABLE Flight_ETL_ErrorLog (
    ErrorLog_ID BIGINT IDENTITY(1,1) PRIMARY KEY,

    -- 1. Thông tin định danh chuyến bay bị lỗi (để truy vết)
    Date DATE,
    Flight_number VARCHAR(10),
    Scheduled_departure VARCHAR(10),
    Origin_Airport CHAR(3),
    Dest_Airport CHAR(3),
    Source_ID INT,

    -- 2. Thông tin chi tiết về lỗi
    Error_Column VARCHAR(50) NULL,      -- Trường bị lỗi (ví dụ: 'Canceled_Reason',
    Error_Description NVARCHAR(512) NOT NULL, -- Chi tiết mô tả lỗi
    Error_Value VARCHAR(200) NULL, -- Lưu trữ giá trị gây lỗi

    -- 3. Thời gian và quá trình
    SSIS_Package VARCHAR(100) NULL, -- Tên package trong SSIS gây ra lỗi
    Error_Timestamp DATETIME NOT NULL DEFAULT GETDATE()
);
```

Bảng ghi các dòng dữ liệu gây lỗi trong Flight

- Luồng chính (Conditional Split Default Output): Các bản ghi hợp lệ (không bị NULL NK) tiếp tục đi vào luồng xử lý chính.

- Check Cancellation_reason type (Phân tách điều kiện):
  - Mô tả: Kiểm tra xem trường Loại Lý do Hủy chuyến (Cancellation_reason type) có phải là một trong các giá trị đã được khởi tạo trong bảng NDS_Reason (A, B, C, D) hay không.
  - Luồng lỗi (Invalid Reason_type): Các bản ghi có loại lý do hủy chuyến không hợp lệ được chuyển qua Derived Column 3 (thêm thông tin lỗi) và ghi vào bảng Flight_ETL_ErrorLog.
  - Luồng chính (Conditional Split Default Output): Các bản ghi hợp lệ tiếp tục đi vào luồng chuẩn hóa (105,244 rows).
- Derived Column, Derived Column 1, Derived Column 2:
  - Mô tả: Ba bước Biến đổi (Derived Column) liên tiếp.
  - Mục đích: Đảm bảo tất cả các cột thời gian (ví dụ: Scheduled_Departure, Scheduled_Arrival, Wheels_on, Wheels_off) từ dạng chuỗi mất các số 0 ở bên trái thành một chuỗi đầy đủ dạng HH:MM, ví dụ: từ '600' thành '06:00', loại bỏ các vấn đề định dạng có thể xảy ra trong dữ liệu nguồn Stage.
- Lookup PK in NDS_Flight:
  - Mô tả: Dùng Khóa Tự nhiên (NK) đã được kiểm tra ở trên để lookup trong bảng đích NDS_Flight.
  - Mục đích: Phân tách luồng dữ liệu thành:
    - Các bản ghi đã tồn tại (cần cập nhật).
    - Các bản ghi mới (cần chèn).
  - Sau bước Lookup sẽ được chia thành 2 luồng đích riêng biệt:
    - Update existed rows (Đích):

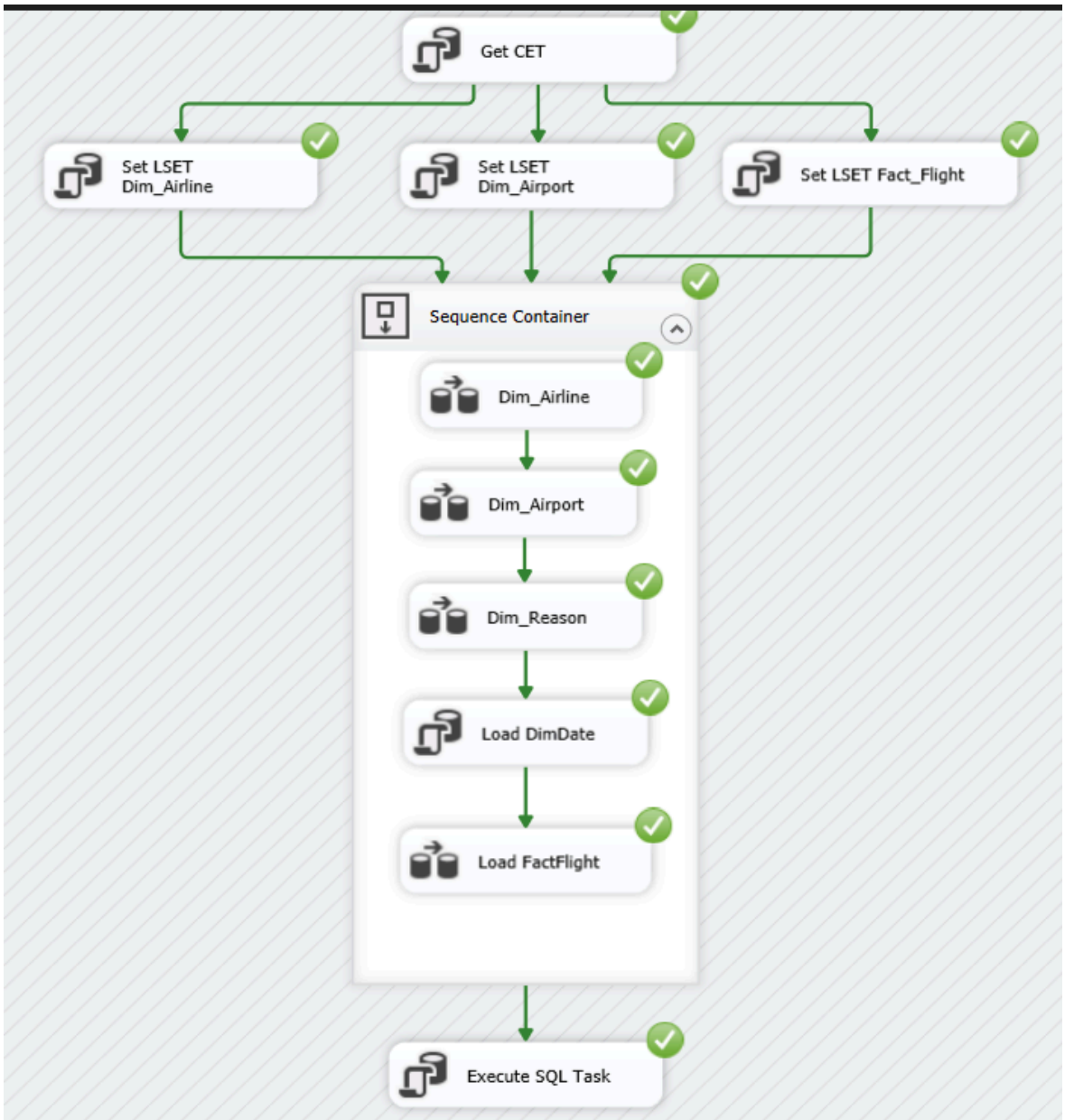
- Mô tả: Xử lý các bản ghi trùng khớp.
- Mục đích: Thực hiện Cập nhật các cột thay đổi (trừ các cột làm NK) trong NDS_Flight bằng dữ liệu mới nhất từ Stage.

■ Insert new rows (Đích):

- Mô tả: Xử lý các bản ghi không trùng khớp.
- Mục đích: Thực hiện Chèn (Insert) các bản ghi mới vào NDS_Flight.

III. NDS to DDS

1. Control Flow



Các bước:

Bước 1: Khởi tạo

Quy trình bắt đầu với một tác vụ thiết lập ban đầu:

Get CET: Đây là tác vụ đầu tiên được thực hiện, có khả năng là để lấy hoặc thiết lập một giá trị biến toàn cục nào đó (có thể là Current Execution Time - Thời gian thực thi hiện tại, hoặc một tham số quan trọng khác).

Bước 2: Tải bảng chiều

a. Nhánh Chiều phụ thuộc (Kết nối với Get CET)

Sau khi khởi tạo, hệ thống tiến hành tải các bảng chiều (Dim) một cách song song và độc lập để chuẩn bị dữ liệu tham chiếu:

- **Set LSET Dim_Airline**: Thiết lập hoặc tải dữ liệu cho bảng Dim_Airline.
- **Set LSET Dim_Airport**: Thiết lập hoặc tải dữ liệu cho bảng Dim_Airport.
- **Set LSET Dim_Reason**: Thiết lập hoặc tải dữ liệu cho bảng Dim_Reason.

Ba tác vụ tải bảng chiều này chạy song song, có liên kết với tác vụ **Get CET**.

b. Nhánh Chiều thời gian (Độc lập)

Các tác vụ này chạy tuần tự để chuẩn bị các khóa thời gian/ngày tháng:

- **Load DimTime**: Tải hoặc cập nhật bảng Chiều Thời Gian.
- **Load DimDate**: Tải hoặc cập nhật bảng Chiều Ngày.

Bước 3: Tải bảng sự kiện

Tác vụ quan trọng nhất, **Load FactFlight** (Tải Bảng Sự kiện Chuyến bay), chỉ bắt đầu khi **DimTime** và **DimDate** đã được tải thành công.

Load FactFlight: Đây là nơi dữ liệu sự kiện cốt lõi về chuyến bay được tải. Sự phụ thuộc vào **DimTime** và **DimDate** là cần thiết để đảm bảo các khóa ngoại cho thời gian và ngày đã sẵn sàng trước khi tải dữ liệu sự kiện.

Bước 4: Tác vụ trong Sequence Container

Sequence Container được sử dụng để nhóm và thực hiện một chuỗi các tác vụ theo thứ tự xác định. Thùng chứa này chỉ bắt đầu khi tất cả các bảng chiều khác đã được tải thành công (Dim_Airline, Dim_Airport, Dim_Reason).

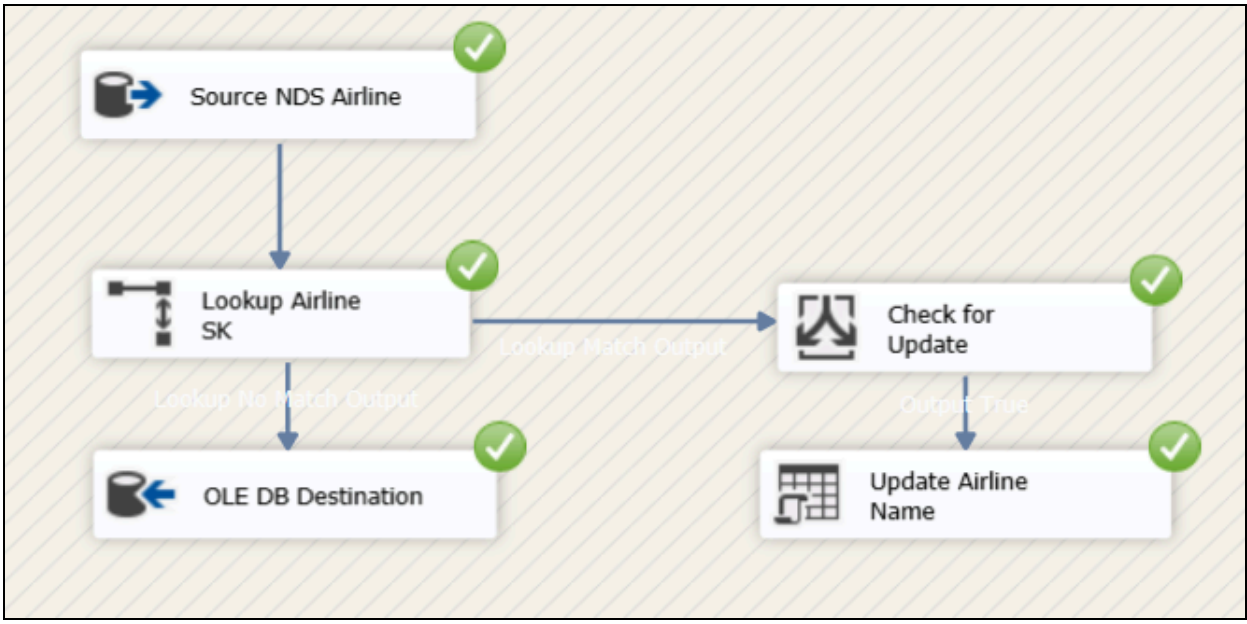
Bước 5: Kết thúc quy trình

Bước cuối cùng của luồng dữ liệu là Execute SQL Task, có mục đích chính là ghi nhận và xác nhận sự thành công của toàn bộ chu trình ETL.

Ghi nhận Metadata và thiết lập điểm bắt đầu cho lần chạy tiếp theo (Incremental Load). Việc cập nhật LSET = CET đảm bảo rằng trong lần chạy ETL tiếp theo, hệ thống sẽ chỉ xử lý và tải những bản ghi dữ liệu mới được tạo ra **sau thời điểm LSET** này, từ đó tối ưu hóa hiệu suất và giảm tải dữ liệu dư thừa.

2. Data Flow

a. Dim Airline



- **Source NDS Airline**

  Đây là điểm khởi đầu, lấy dữ liệu gốc, đã được làm sạch từ khu vực **Staging/NDS (Nomination Data Store)**. Dữ liệu này chứa các bản ghi hãng hàng không cần được đồng bộ hóa với Kho dữ liệu.

- **Lookup Airline SK**
  - Tra cứu trong bảng Dim_Airline (DDS) bằng Khóa Nghiệp vụ (NK): IATA_Code.
  - **Phân luồng:** Thành phần này phân tách dữ liệu thành hai luồng chính:
    - **Lookup No Match Output:**Luồng này chứa các bản ghi mới hoàn toàn. Logic khẳng định rằng bản ghi này chưa có Surrogate Key nào và cần được chèn.
    - **Lookup Match Output:** Luồng này chứa các bản ghi đã tồn tại trong DDS. Dữ liệu DDS liên quan (ví dụ: Airline_Name cũ) được mang vào luồng để so sánh.
- **OLE DB Destination**

  Nhận luồng **No Match Output**. Thực hiện lệnh INSERT khối (Bulk Load) vào Dim_Airline. Hệ thống tự động tạo Khóa Surrogate mới cho các bản ghi này.

- **Check for Update**

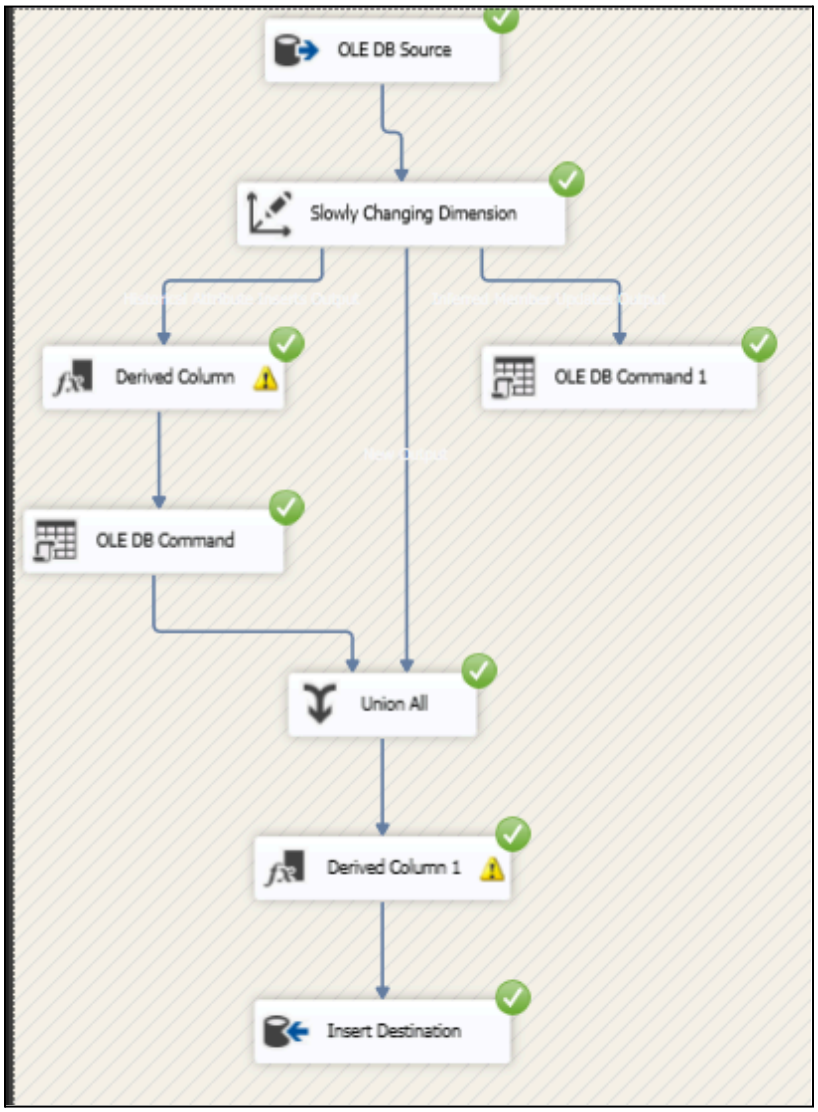
  Nhận dữ liệu từ luồng **Lookup Match Output**. Đây là thành phần logic quyết định xem có cần thực hiện thao tác cập nhật hay không. Nó so sánh

các thuộc tính được theo dõi giữa dữ liệu nguồn và dữ liệu đích (So sánh: [NDS.Airline_Name] != [DDS.Airline_Name]).

- **Đầu ra:** Chỉ chuyển tiếp dữ liệu qua **Output True** nếu phát hiện có sự **thay đổi** về giá trị thuộc tính.

- Update Airline Name
    1. Nhận dữ liệu từ **Output True**. Thực hiện thao tác **UPDATE** (Cập nhật) để **ghi đè** Tên Hãng Hàng Không cũ bằng giá trị mới.
    2. **Chiến lược SCD:** Thao tác này khẳng định việc áp dụng **SCD Loại 1** (Slowly Changing Dimension Type 1), nghĩa là dữ liệu lịch sử không được duy trì, chỉ có thông tin hiện tại nhất là được lưu trữ.

b. Dim Airport



- **OLE DB Source**

  Điểm khởi đầu. Thành phần này kết nối với cơ sở dữ liệu nguồn (ví dụ: OLTP hoặc Staging Area) để trích xuất dữ liệu mới hoặc dữ liệu đã thay đổi của bảng chiều cần xử lý.

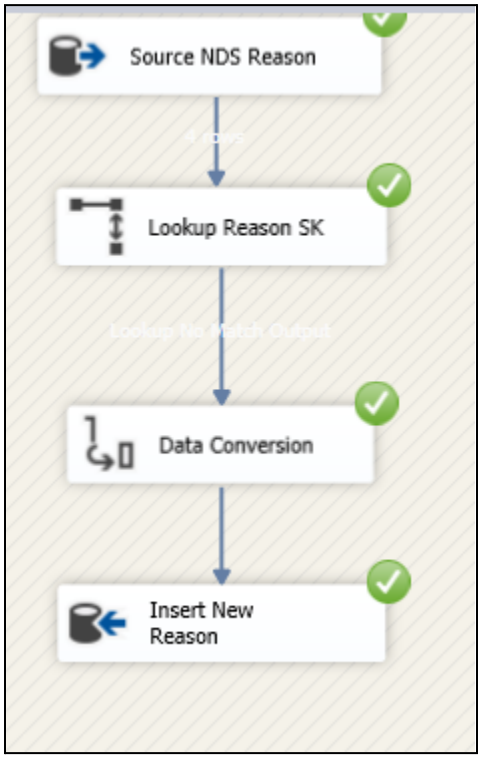
- **Slowly Changing Dimension (SCD Component)**

  Đây là thành phần logic cốt lõi. Nó tự động thực hiện thao tác Lookup (Tra cứu) dựa trên khóa nghiệp vụ của dữ liệu nguồn so với bảng chiều đích.

    - **Phân luồng:** Thành phần SCD chia dữ liệu đầu ra thành các luồng xử lý khác nhau dựa trên kết quả đối chiếu và cấu hình của các cột:
        - **Luồng Thay đổi:** Dành cho các bản ghi có sự thay đổi về thuộc tính cần được theo dõi. Luồng này sẽ được tách thành các luồng nhỏ hơn cho các loại SCD (Loại 1 và Loại 2).
        - **Luồng Bản ghi Mới:** Dành cho các bản ghi chưa tồn tại trong bảng đích.
        - **Luồng Không thay đổi:** Dành cho các bản ghi đã tồn tại nhưng không có sự khác biệt (thường bị loại bỏ).
- **Derived Column & OLE DB Command (Xử lý SCD Loại 1 - Ghi đè)**
    - **Derived Column:**
        - Nhận luồng dữ liệu cần áp dụng SCD Loại 1 (ghi đè). Thành phần này có thể được sử dụng để thêm hoặc sửa đổi các cột cần thiết trước khi cập nhật
    - **OLE DB Command:**
        - Nhận đầu ra từ **Derived Column**. Thực hiện lệnh **UPDATE** để ghi đè dữ liệu cũ bằng dữ liệu mới. Đây là việc áp dụng **SCD Loại 1** cho các thuộc tính đã cấu hình.
- **OLE DB Command 1 (Xử lý SCD Loại 2 - Vô hiệu hóa bản ghi cũ)**
    - Nhận luồng dữ liệu cần áp dụng SCD Loại 2 (duy trì lịch sử). Thành phần này được sử dụng để thực hiện lệnh **UPDATE** trên các bản ghi cũ.
    - Nó cập nhật bản ghi cũ bằng cách thiết lập **cờ hiệu current_flag= False** (Vô hiệu hóa bản ghi cũ) trong bảng chiều đích.
- **Union All (Kết hợp các luồng INSERT)**
    - Kết hợp lại các luồng dữ liệu được tạo ra cho thao tác **INSERT** (Chèn).
    - **Các luồng kết hợp:** Thường là:
        - Luồng Bản ghi Mới ban đầu (từ SCD Component).
        - Luồng Bản ghi SCD Loại 2 Mới (Tức là bản ghi mới được tạo ra để thay thế bản ghi cũ bị vô hiệu hóa).

- **Derived Column 1 (Cột Dẫn xuất 1)**
    - Nhận đầu ra đã kết hợp từ **Union All**. Thành phần này có thể được dùng để chuẩn bị hoặc tính toán các cột cuối cùng
- **Insert Destination**
    - Nhận dữ liệu đã chuẩn bị từ **Derived Column 1**. Thực hiện lệnh **INSERT** để thêm các bản ghi mới (bao gồm cả bản ghi mới hoàn toàn và bản ghi thay thế SCD Loại 2) vào bảng chiều đích.

c. Dim Reason



- **Source NDS Reason**

  Lấy dữ liệu Reason từ bảng nguồn NDS. Đây là điểm bắt đầu của luồng dữ liệu, cung cấp các Khóa Tự nhiên như Reason_Type và Reason_Name.

- **Lookup Reason SK (Tra cứu Khóa Thay thế Lý do)**

  Thực hiện thao tác **tra cứu** bản ghi nguồn so với bảng Dim_Reason đích bằng cách sử dụng **Khóa Nghiệp vụ** (Business Key) của Lý do.
    - **Mục đích:** Đối chiếu dữ liệu nguồn (Reason_Type + Reason_Name) với các bản ghi hiện có trong Dim_Reason (DDS).
    - **Phân luồng:**

        - **Lookup No Match Output:** Chứa các bản ghi có Khóa Nghiệp vụ **chưa tồn tại**. Đây là các **Reason mới** cần được chèn (INSERT).
        - **Lookup Match Output:** Chứa các bản ghi đã tồn tại trong DDS. Trong chiến lược Insert Only này, luồng này bị loại bỏ (discarded) khỏi Data Flow.
- **Data Conversion (Chuyển đổi Dữ liệu)**
    - Nhận dữ liệu từ luồng **Lookup No Match Output**. Đảm bảo kiểu dữ liệu đầu vào (Input) khớp chính xác với kiểu dữ liệu của bảng đích (ví dụ: chuyển từ DT_WSTR sang DT_NVARCHAR và đảm bảo độ dài chuỗi là chính xác).
- **Insert New Reason**
    - Nhận dữ liệu **đã chuyển đổi** từ luồng **No Match Output**. Thực hiện lệnh INSERT vào bảng Dim_Reason
    - **Thao tác:** SQL Server tự động tạo Khóa Surrogate (Reason_ID) mới cho mỗi Lý do mới được chèn.

d. DimTime

```sql
USE Project_DDS_DB;
GO

IF NOT EXISTS (SELECT 1 FROM Dim_Time)
BEGIN
    ;WITH T AS (
        SELECT CAST(0 AS INT) AS n
        UNION ALL
        SELECT n + 1 FROM T WHERE n < 1439   -- 0..1439 = 1440 phút
    )
    INSERT INTO Dim_Time ([Hour], [Minute], Part_of_day)
    SELECT
        n / 60 AS [Hour],           -- 0..23
        n % 60 AS [Minute],         -- 0..59
        CASE
            WHEN n / 60 BETWEEN 0  AND 5  THEN N'Night'      -- hoặc N'Đêm'
            WHEN n / 60 BETWEEN 6  AND 11 THEN N'Morning'    -- hoặc N'Sáng'
            WHEN n / 60 BETWEEN 12 AND 17 THEN N'Afternoon'  -- hoặc N'Chiều'
            ELSE N'Evening'                                   -- hoặc N'Tối'
        END AS Part_of_day
    FROM T
    OPTION (MAXRECURSION 0);
END
```

74 % No issues found

Results Messages

	Time_ID	Hour	Minute	Part_of_day
1	1	0	0	Night
2	2	0	1	Night
3	3	0	2	Night
4	4	0	3	Night
5	5	0	4	Night
6	6	0	5	Night
7	7	0	6	Night
8	8	0	7	Night
9	9	0	8	Night
10	10	0	9	Night
11	11	0	10	Night
12	12	0	11	Night
13	13	0	12	Night
14	14	0	13	Night

Nguồn đầu vào: Bảng Dim_Time được xây dựng bằng cách sinh ra toàn bộ 1440 mốc thời gian (24 giờ × 60 phút).

Kiểm tra dữ liệu tồn tại trong Dim_Time

- Trước khi sinh dữ liệu, hệ thống kiểm tra xem bảng Dim_Time đã có dữ liệu hay chưa bằng mệnh đề IF NOT EXISTS.
- Mục đích:
    - Tránh insert trùng dữ liệu Dim_Time
    - Đảm bảo Dim_Time chỉ được tạo một lần và giữ vai trò bảng dimension tĩnh

Generate minute index values (CTE T)

- Sử dụng Common Table Expression (CTE) đệ quy để sinh ra dãy số liên tục từ 0 đến 1439, tương ứng với tổng số phút trong một ngày.
- Mục đích:
    - Tạo nguồn dữ liệu thời gian mà không cần bảng vật lý
    - Mỗi giá trị n đại diện cho một mốc phút trong ngày

Chuyển đổi minute index thành Hour và Minute:

- Từ giá trị n trong CTE:
    - Hour = n / 60 → giá trị từ 0 đến 23
    - Minute = n % 60 → giá trị từ 0 đến 59
- Mục đích:
    - Tách mỗi mốc phút thành cặp (Giờ, Phút)
    - Đảm bảo Dim_Time có đầy đủ tất cả các thời điểm trong ngày với độ chi tiết từng phút

Phân loại Part_of_day (Night, Morning, Afternoon, Evening):

- Dựa trên giá trị Hour, hệ thống phân loại thời điểm trong ngày theo các khoảng:
    - Night: 00:00 – 05:59
    - Morning: 06:00 – 11:59
    - Afternoon: 12:00 – 17:59
    - Evening: 18:00 – 23:59
- Mục đích:
    - Hỗ trợ phân tích theo khung thời gian trong ngày
    - Phục vụ báo cáo và KPI như: chuyến bay buổi sáng, buổi tối, giờ cao điểm

Insert into Dim_Time:

- Toàn bộ kết quả sinh ra từ CTE (Hour, Minute, Part_of_day) được insert vào bảng Dim_Time.
- Mục đích:
    - Hoàn thiện bảng Dim_Time với 1440 bản ghi cố định
    - Cung cấp dimension thời gian chi tiết cho các bảng fact (Fact_Flight) thông qua Time_ID

e. DimDate

```sql
IF OBJECT_ID('dbo.usp_Extend_Dim_Date', 'P') IS NOT NULL
    DROP PROCEDURE dbo.usp_Extend_Dim_Date;
GO

CREATE PROCEDURE dbo.usp_Extend_Dim_Date
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @NeedMin DATE, @NeedMax DATE;
    DECLARE @CurMin  DATE, @CurMax  DATE;
    DECLARE @BuffDay INT = 30;

    SELECT
        @NeedMin = MIN([Date]),
        @NeedMax = MAX([Date])
    FROM Project_NDS_DB.dbo.NDS_Flight;

    IF @NeedMin IS NULL OR @NeedMax IS NULL
    BEGIN
        PRINT 'No source date in NDS. Nothing to extend for Dim_Date.';
        RETURN;
    END;

    SET @NeedMin = DATEADD(DAY, -@BuffDay, @NeedMin);
    SET @NeedMax = DATEADD(DAY,  @BuffDay, @NeedMax);

    -- 2. Lấy range hiện tại Dim_Date (trong DDS)
    SELECT
        @CurMin = MIN(Full_Date),
        @CurMax = MAX(Full_Date)
    FROM dbo.Dim_Date;

    IF @CurMax IS NULL
```

61 % No issues found

Results Messages

	Date_ID	Full_Date	Day	Month	Quarter	Year	Season	Day_of_week	Is_Weekend
407	407	2016-01-12	12	1	1	2016	Winter	3	0
408	408	2016-01-13	13	1	1	2016	Winter	4	0
409	409	2016-01-14	14	1	1	2016	Winter	5	0
410	410	2016-01-15	15	1	1	2016	Winter	6	0
411	411	2016-01-16	16	1	1	2016	Winter	7	1
412	412	2016-01-17	17	1	1	2016	Winter	1	1
413	413	2016-01-18	18	1	1	2016	Winter	2	0
414	414	2016-01-19	19	1	1	2016	Winter	3	0
415	415	2016-01-20	20	1	1	2016	Winter	4	0

Nguồn đầu vào:

- Dim_Date không lấy dữ liệu trực tiếp từ các bảng dimension khác, mà được tự động mở rộng bằng Stored Procedure dbo.usp_Extend_Dim_Date.
- Dim_Date được xây dựng theo kiểu "extend when needed": dựa trên khoảng ngày thực tế xuất hiện trong Project_NDS_DB.dbo.NDS_Flight, sau đó mở rộng thêm một khoảng đệm (buffer) 30 ngày để đảm bảo lookup không bị thiếu ngày.

Xác định khoảng ngày cần có từ NDS_Flight (NeedMin, NeedMax):

- Procedure lấy ngày nhỏ nhất và ngày lớn nhất từ cột [Date] trong Project_NDS_DB.dbo.NDS_Flight:
    - @NeedMin = MIN([Date])
    - @NeedMax = MAX([Date])
- Nếu không có dữ liệu ngày (NeedMin/NeedMax = NULL) thì procedure dừng và in thông báo "Nothing to extend".
- Mục đích:
    - Lấy đúng phạm vi ngày thực tế phát sinh trong dữ liệu chuyến bay
    - Tránh tạo thừa dữ liệu ngày khi NDS chưa có dữ liệu
    - Đảm bảo Dim_Date phục vụ đúng nhu cầu phân tích theo dữ liệu hiện có

Mở rộng phạm vi ngày bằng Buffer (±30 ngày):

- Sau khi có NeedMin/NeedMax, procedure mở rộng thêm:
    - NeedMin = NeedMin - 30 ngày
    - NeedMax = NeedMax + 30 ngày
- Mục đích:
    - Tránh lỗi lookup khi dữ liệu có thể phát sinh ngày cận biên (đầu tháng/cuối tháng)
    - Cho phép Fact_Flight load ổn định khi dữ liệu nguồn tăng dần theo thời gian
    - Tạo "vùng an toàn" để DW không thiếu ngày khi chạy incremental

Lấy phạm vi hiện tại Dim_Date (CurMin, CurMax):

- Procedure kiểm tra Dim_Date hiện đang phủ được đến ngày nào bằng:
    - @CurMin = MIN(Full_Date)
    - @CurMax = MAX(Full_Date)
- Nếu Dim_Date đang rỗng (CurMax is NULL) thì set CurMax = NeedMin - 1 ngày để chuẩn bị insert từ NeedMin.
- Mục đích:
    - Biết Dim_Date hiện có tới đâu để chỉ insert phần thiếu
    - Tránh insert trùng ngày đã tồn tại
    - Hỗ trợ cơ chế mở rộng tăng dần (incremental extend)

Điều kiện mở rộng: chỉ insert khi NeedMax > CurMax:

- Procedure chỉ thực hiện insert khi ngày cần có (NeedMax) lớn hơn ngày hiện có lớn nhất (CurMax).
- Nếu Dim_Date đã bao phủ đủ range thì chỉ in: "Dim_Date already covers needed range."
- Mục đích:
    - Tối ưu ETL: không rebuild toàn bộ bảng mỗi lần chạy
    - Đảm bảo Dim_Date luôn đủ để fact lookup nhưng không tốn tài nguyên dư thừa

Vòng lặp sinh từng ngày cần thêm (CTE D)

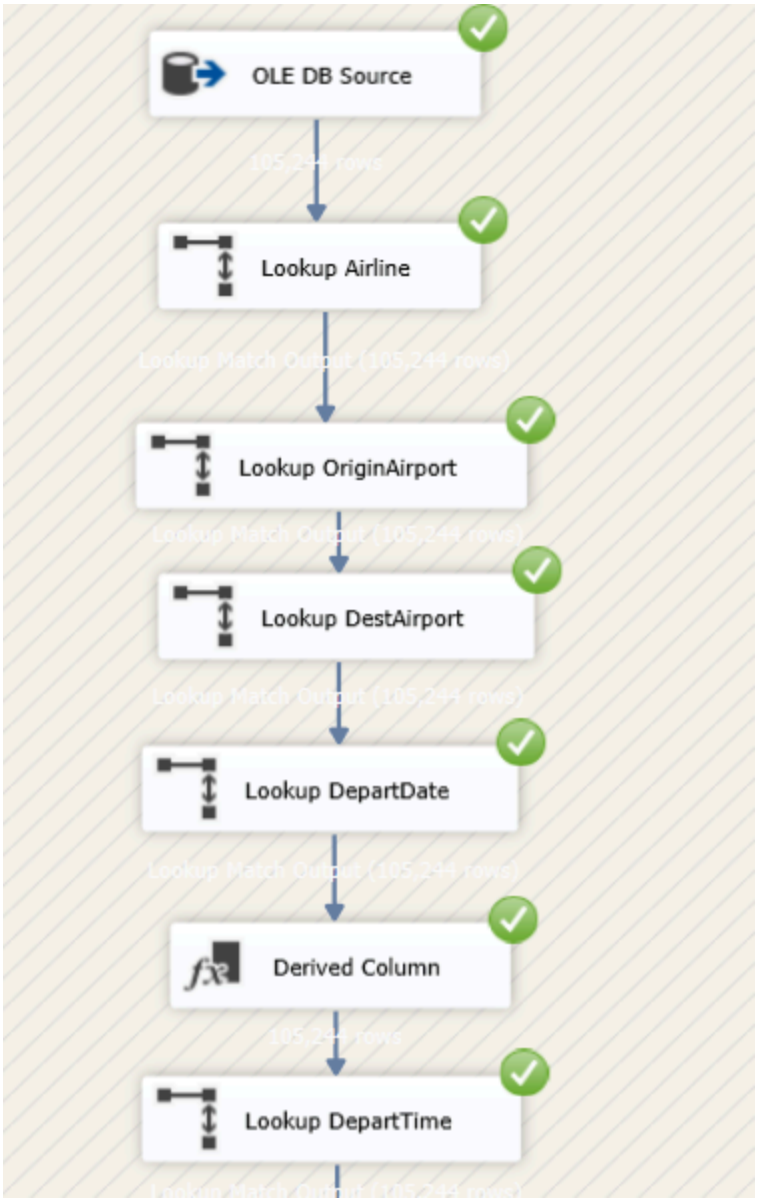
- Sử dụng CTE đệ quy để sinh lần lượt từng ngày từ CurMax + 1 đến NeedMax:
  - Bắt đầu: DATEADD(DAY, 1, @CurMax)
  - Tăng dần từng ngày cho đến khi chạm NeedMax
- **Mục đích:**
  - Tự động tạo đầy đủ mọi ngày trong khoảng thiếu mà không cần bảng nguồn
  - Đảm bảo không thiếu ngày nào (no gaps) trong phần mở rộng
  - Hỗ trợ ETL chạy nhiều lần mà Dim_Date vẫn tăng dần đúng

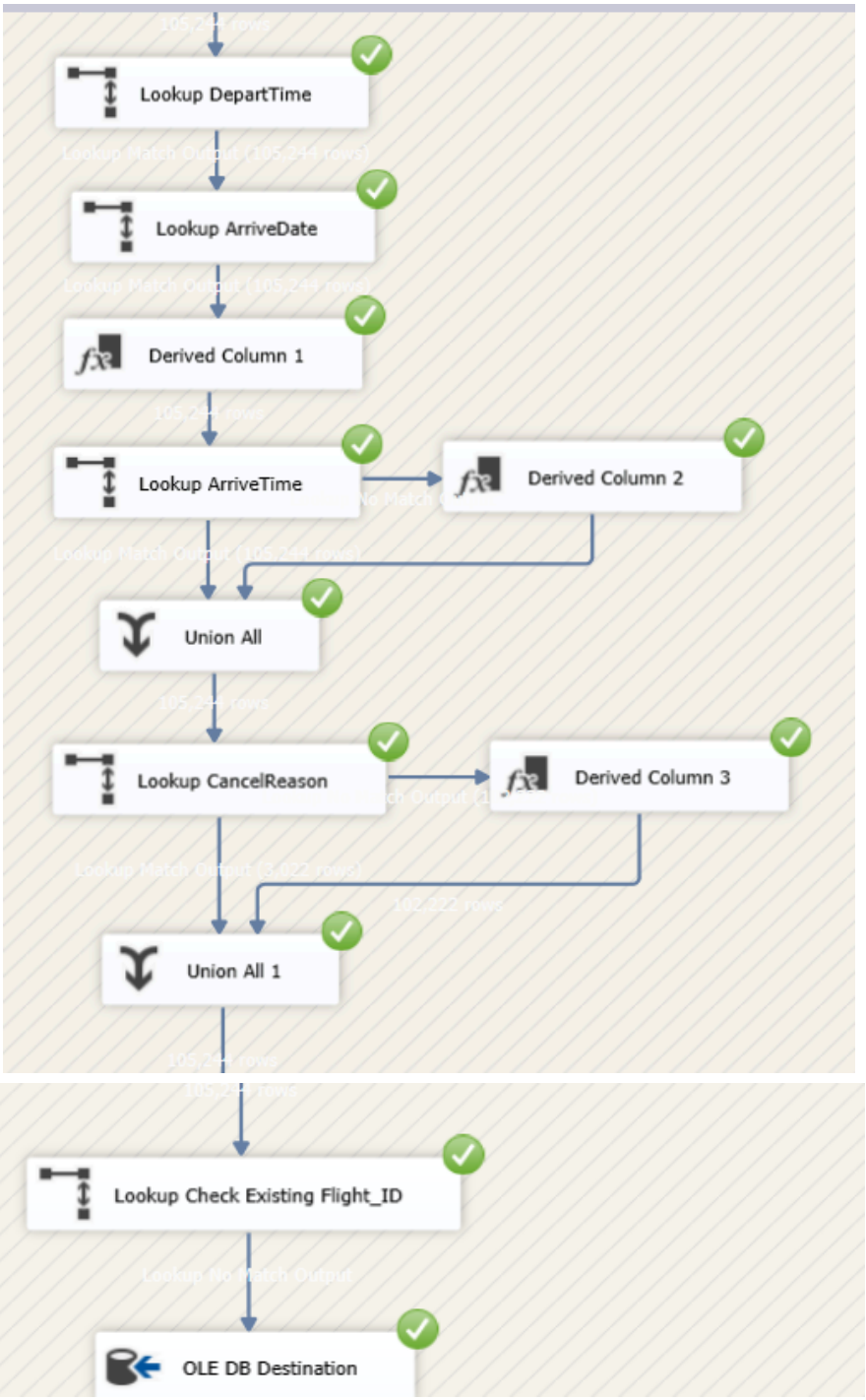
Sinh các thuộc tính phân tích theo thời gian: Trong mỗi ngày được sinh ra, hệ thống tính toán và tạo các thuộc tính chuẩn của Date Dimension:

- Full_Date: ngày đầy đủ (YYYY-MM-DD), là trường chính để join/hiển thị thời gian.
- Day: ngày trong tháng (1–31), hỗ trợ báo cáo theo ngày.
- Month: tháng trong năm (1–12), dùng để nhóm dữ liệu theo tháng.
- Quarter: quý trong năm (1–4), phục vụ phân tích theo quý.
- Year: năm, hỗ trợ phân tích dài hạn.
- Season: phân loại mùa theo tháng: Winter (12,1,2), Spring (3,4,5), Summer (6,7,8), Autumn (9,10,11)
  Mục đích: phân tích mùa vụ, xu hướng du lịch, ảnh hưởng thời tiết.
- Day_of_week: thứ trong tuần bằng DATEPART(WEEKDAY, Full_Date).
- Is_Weekend: 1 nếu là Thứ Bảy/Chủ Nhật, ngược lại 0.
- Mục đích tổng quát:
  - Cho phép phân tích KPI theo ngày, tháng, quý, năm
  - So sánh hiệu suất Weekday với Weekend
  - Dùng cho drill-down time hierarchy trong BI

Insert vào Dim_Date: Sau khi sinh ngày và tính đủ thuộc tính, procedure insert vào dbo.Dim_Date các cột Full_Date, Day, Month, Quarter, Year, Season, Day_of_week, Is_Weekend.

f. Fact Flight


OLE DB Source
Lookup Airline
Lookup OriginAirport
Lookup DestAirport
Lookup DepartDate
Derived Column
Lookup DepartTime



OLE DB Source – Lấy dữ liệu từ NDS_Flight

- Chức năng: Đây là nguồn dữ liệu đầu vào, lấy từ bảng NDS_Flight, vốn đã được chuẩn hoá từ Staging.
- Trong bảng này, các trường vẫn ở dạng Natural Key như:
    - Mã hãng bay (AirlineCode)
    - Sân bay đi (Origin)
    - Sân bay đến (Dest)

    - Ngày cất cánh / hạ cánh (DepartDate, ArriveDate)
    - Giờ cất cánh / hạ cánh
    - Lý do hủy chuyến dạng text
- Ý nghĩa nghiệp vụ: NDS_Flight là bản ghi thực tế (transactional event), phản ánh mỗi chuyến bay và thông tin hoạt động tương ứng. Đây là thông tin nền tảng để phân tích và gán SK từ hệ thống DDS.

Lookup AirlineSK – Chuẩn hóa Hãng bay

- Chức năng: Đối chiếu mã hãng bay (IATA_Code) từ NDS_Flight với Dim_Airline để tìm AirlineSK.
- Ý nghĩa: Mỗi chuyến bay thuộc một hãng bay duy nhất → Fact phải liên kết với Dim_Airline.
- Vai trò:
    - Thay thế Natural Key (IATA Code) bằng Surrogate Key.
    - Cho phép Fact_Flight tham gia join với Dim_Airline.

Lookup OriginAirportSK – Chuẩn hóa Sân bay xuất phát

- Chức năng: Tra cứu mã sân bay đi (OriginAirport) trong Dim_Airport.
- Lý do nghiệp vụ: Sân bay đi là thành phần quan trọng để phân tích:
    - Mật độ khai thác theo sân bay
    - Tình trạng delay tại từng origin
    - Hiệu suất hoạt động vùng miền

Lookup DestinationAirportSK – Chuẩn hóa Sân bay đến

- Chức năng: Tra cứu mã sân bay đến (DestAirport) trong Dim_Airport.
- Ý nghĩa phân tích:
    - Đo lường thời gian hạ cánh
    - Phân tích chất lượng khai thác tại điểm đến
    - So sánh tình trạng delay giữa điểm đi và điểm đến

Lookup DepartDateSK – Chuẩn hóa Ngày cất cánh:

- Chức năng: Lookup ngày bay trong Dim_Date để lấy DepartDateSK.
- Lý do nghiệp vụ:
- Giúp phân tích theo:
    - Ngày / tháng / quý / năm
    - Mùa vụ
    - Ngày trong tuần
    - Ngày nghỉ lễ

Derived Column – Chuẩn hóa giờ/phút cất cánh:

- Chức năng: Tách dữ liệu thời gian thành:
    - Hour
    - Minute

- Lý do nghiệp vụ:
    - Dim_Time được chuẩn hóa theo 24h × 60 phút → 1440 giá trị.
    - NDS_Flight thường chứa thời gian dạng datetime → phải chuẩn hóa để Lookup Time SK.
- Cho phép phân tích theo:
    - Khung giờ cao điểm
    - Buổi sáng / trưa / chiều / tối
    - Hiệu suất theo giờ

Lookup DepartTimeSK – Chuẩn hóa thời gian cất cánh:

- Chức năng: Tra cứu giờ/phút cất cánh trong Dim_Time.
- Ý nghĩa:
    - Thời điểm cất cánh ảnh hưởng lớn đến delay pattern.
    - Chuyến bay sáng thường ít delay hơn chiều tối.

Lookup ArriveDateSK – Chuẩn hóa Ngày hạ cánh:

- Chức năng: Lookup ngày hạ cánh trong Dim_Date để lấy ArriveDateSK.
- Lý do nghiệp vụ: Một chuyến bay có thể hạ cánh sang ngày hôm sau, đặc biệt với chuyến bay đêm. Dim_Date giúp phân tích chênh lệch ngày bay.

Derived Column 1 – Chuẩn hóa giờ/phút hạ cánh:

- Chức năng: Tách hoặc chuẩn hóa thông tin thời gian để phục vụ Lookup tiếp theo:
    - Arr_Hour
    - Arr_Min

Lookup ArriveTimeSK – Chuẩn hóa thời gian hạ cánh:

- Chức năng: Tra cứu giờ/phút hạ cánh với Dim_Time.
- Ý nghĩa phân tích:
- Các báo cáo có thể trả lời:
    - Chuyến bay hạ cánh đúng giờ?
    - Arrival delay nhiều trong khung giờ nào?
    - Chuyến bay đêm có bị trễ nhiều hơn không?

Derived Column 2 + Union All – Xử lý trường hợp không lookup được Time:

- Chức năng:
    - Với các bản ghi không lookup được Arrive_Time, hệ thống gán giá trị mặc định (ví dụ Time_ID = 0).
    - Sau đó sử dụng Union All để hợp nhất:
        - Dữ liệu lookup thành công
        - Dữ liệu đã được gán mặc định

- Mục đích:
    - Đảm bảo không mất dữ liệu chuyến bay
    - Giữ Fact_Flight luôn đầy đủ record

Lookup CancelReasonSK – Chuẩn hóa lý do hủy chuyến:

- Chức năng: Đối chiếu mã lý do hủy (Reason_Type) với bảng Dim_Reason.
- Ý nghĩa phân tích:
- Giúp thống kê:
    - % hủy chuyến theo từng loại lý do
    - Ảnh hưởng thời tiết
    - Sự cố kỹ thuật
    - Vấn đề hành khách
    - Quy định an ninh hoặc NAS
- Vai trò: Làm sạch dữ liệu, chuẩn hóa lý do thành SK thay vì text rời rạc.

Derived Column 3 + Union All 1 – Xử lý Reason không match:

- Chức năng:
    - Với các bản ghi không lookup được lý do hủy, gán Reason_ID mặc định (ví dụ "Unknown").
    - Sử dụng Union All để gộp dữ liệu match và no match.
- Mục đích:
    - Tránh lỗi Foreign Key
    - Đảm bảo Fact_Flight luôn load được toàn bộ dữ liệu

Lookup Check Existing Flight_ID – Chống trùng dữ liệu:

- Chức năng: Kiểm tra Flight_ID đã tồn tại trong Fact_Flight hay chưa.
- Vai trò:
    - Nếu đã tồn tại → loại bỏ, không insert lại
    - Nếu chưa tồn tại → cho phép insert
- Mục đích:
    - Tránh trùng dữ liệu khi ETL chạy nhiều lần
    - Đảm bảo Fact_Flight chỉ chứa dữ liệu mới

OLE DB Destination – Nạp dữ liệu vào Fact_Flight:

- Chức năng: Chèn toàn bộ dữ liệu đã chuẩn hóa SK vào bảng Fact_Flight.
- Mục đích:
    - Hoàn thiện quá trình nạp Fact
    - Đảm bảo Fact chỉ chứa Surrogate Key và các số đo (measures) như:
        - Thời gian delay
        - Thời gian thực tế

- Khoảng cách
- Số phút trễ cất cánh/hạ cánh

IV. Tự động hoá ETL theo schedule

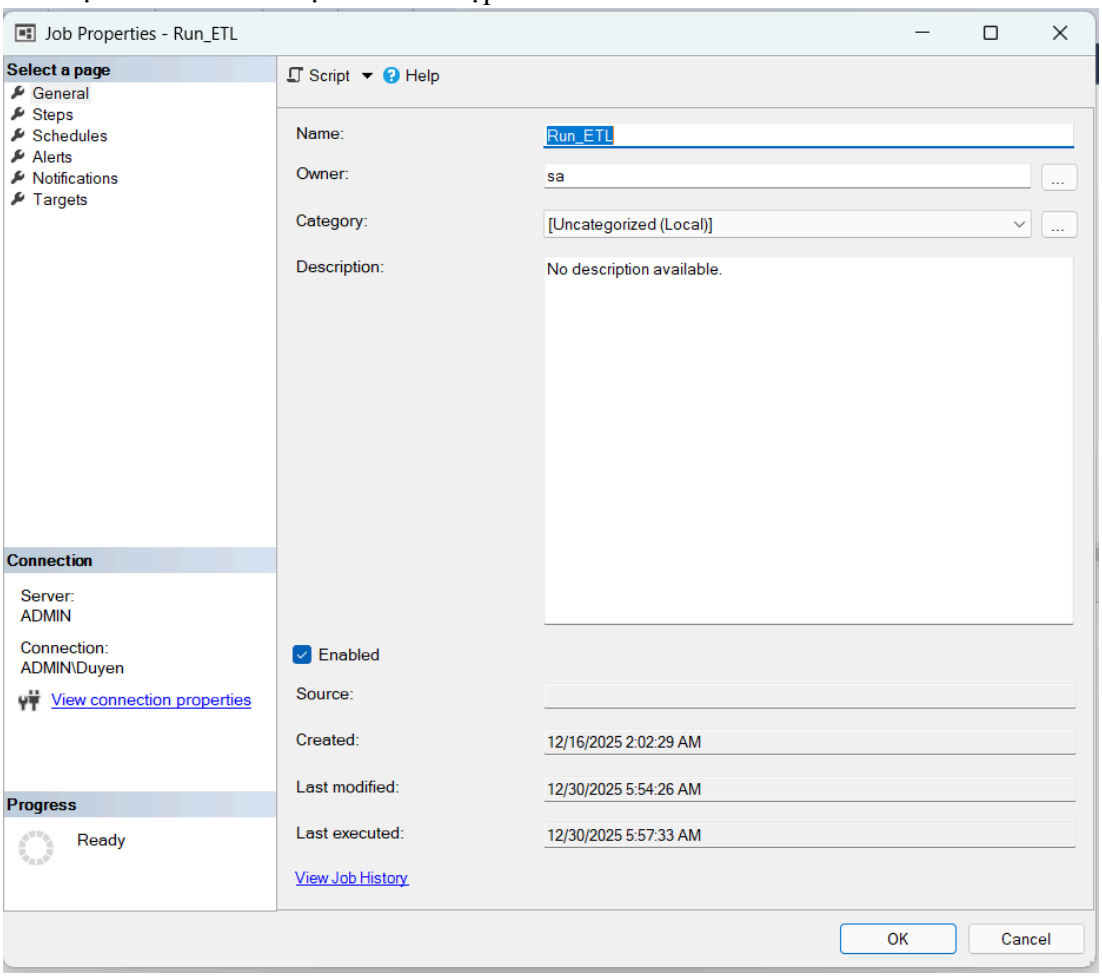
1. Giải pháp tự động hóa

Sử dụng SQL Server Agent để lập lịch và quản lý các tác vụ ETL. Các Job được cấu hình để tự động thực thi các bước nạp, xử lý và cập nhật dữ liệu từ nguồn vào kho dữ liệu theo chu kỳ định sẵn.

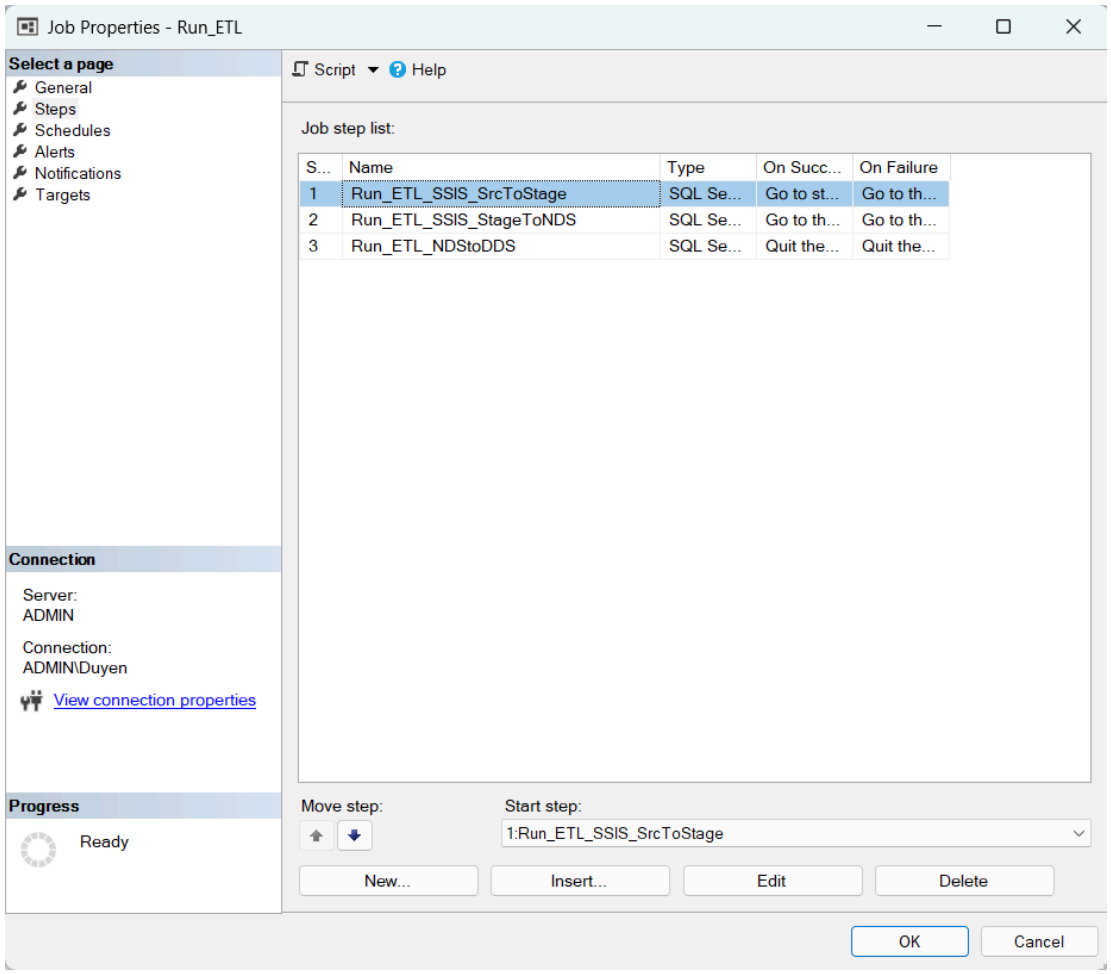
2. Quy trình

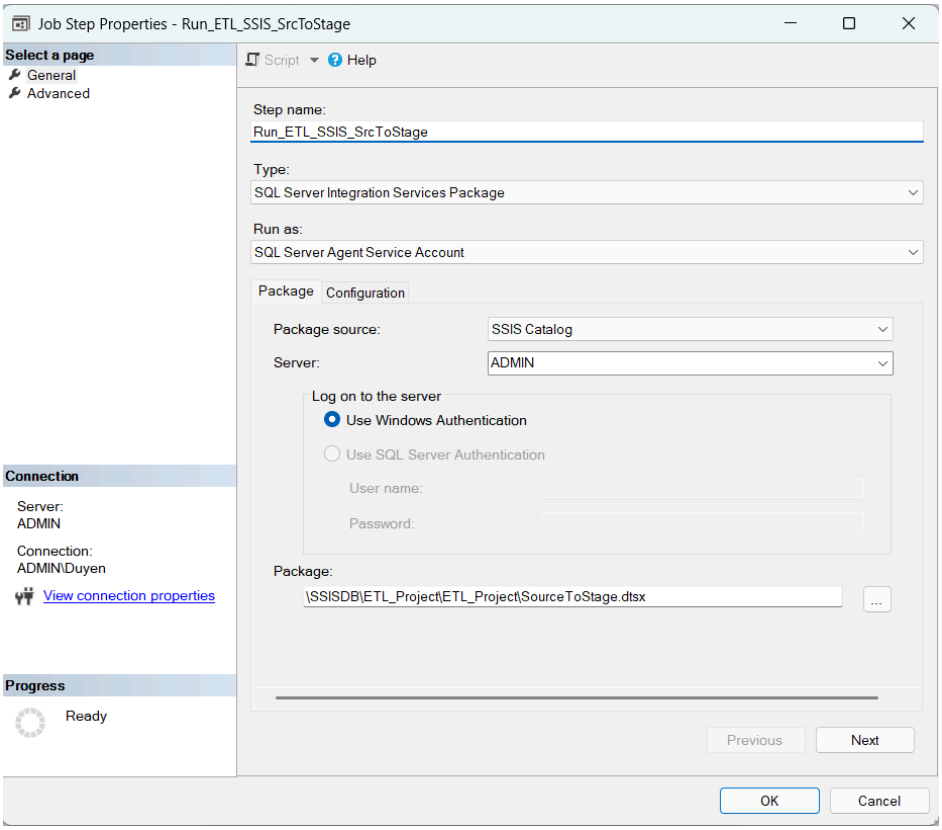
Quy trình tự động hóa bao gồm các bước chính:

- Khởi tạo Job ETL theo lịch đã thiết lập.

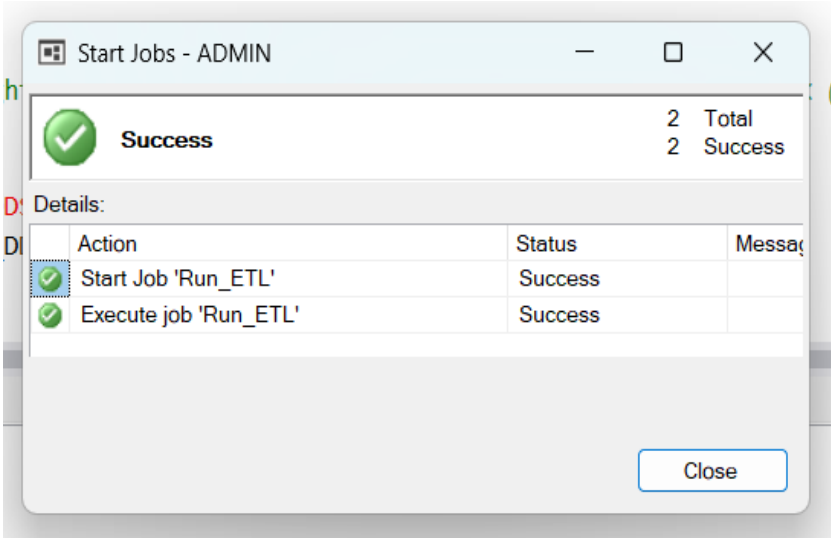


- Thực thi các gói ETL để trích xuất dữ liệu từ nguồn.


Job Properties - Run_ETL
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Job step list:
S...
Name
Type
On Succ...
On Failure
1
Run_ETL_SSIS_SrcToStage
SQL Se...
Go to st...
Go to th...
2
Run_ETL_SSIS_StageToNDS
SQL Se...
Go to th...
Go to th...
3
Run_ETL_NDStoDDS
SQL Se...
Quit the...
Quit the...
Connection
Server:
ADMIN
Connection:
ADMIN\Duyen
View connection properties
Progress
Ready
Move step:
Start step:
1:Run_ETL_SSIS_SrcToStage
New...
Insert...
Edit
Delete
OK
Cancel

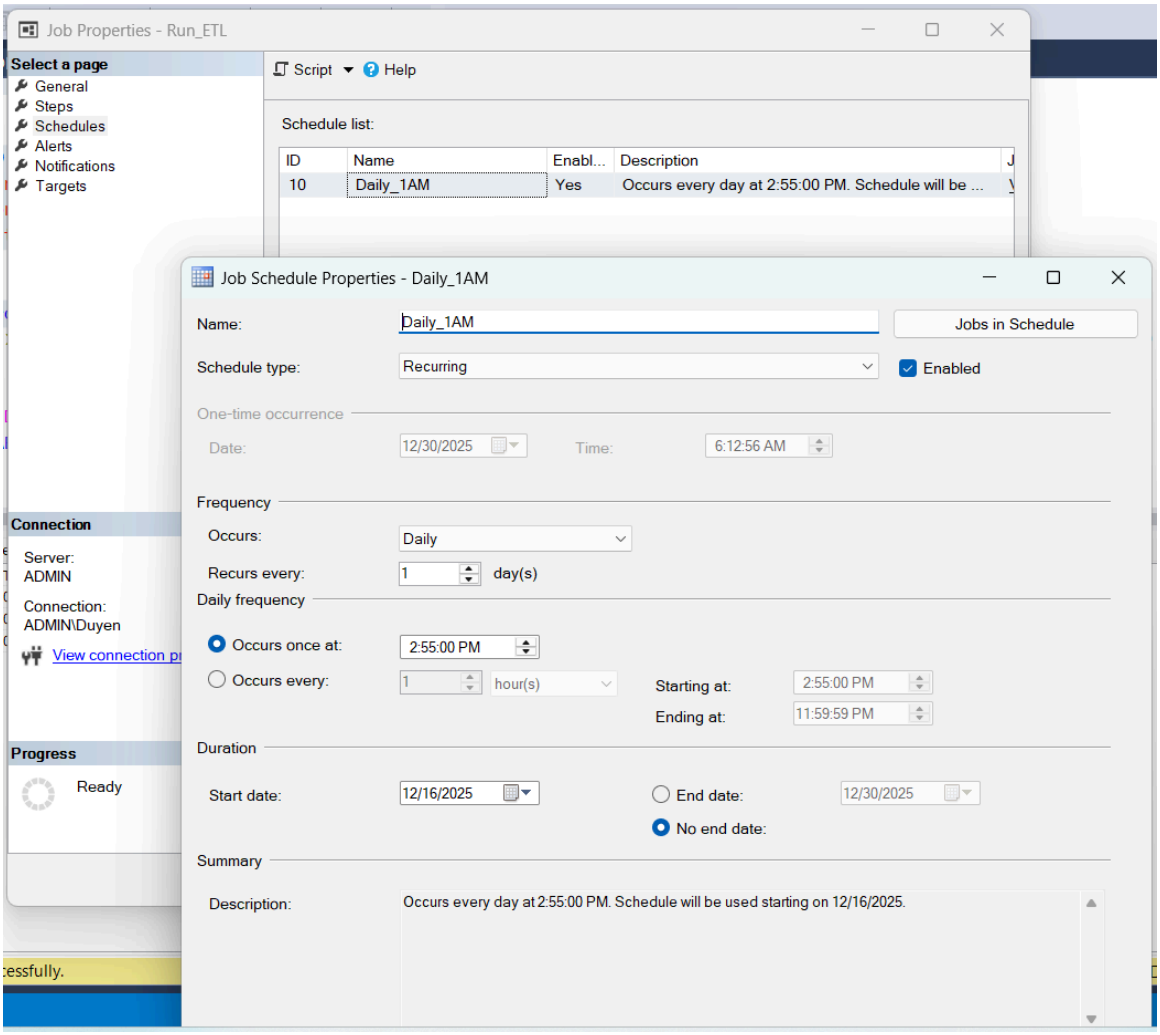


- Ghi nhận trạng thái thực thi để phục vụ theo dõi và kiểm soát lỗi.



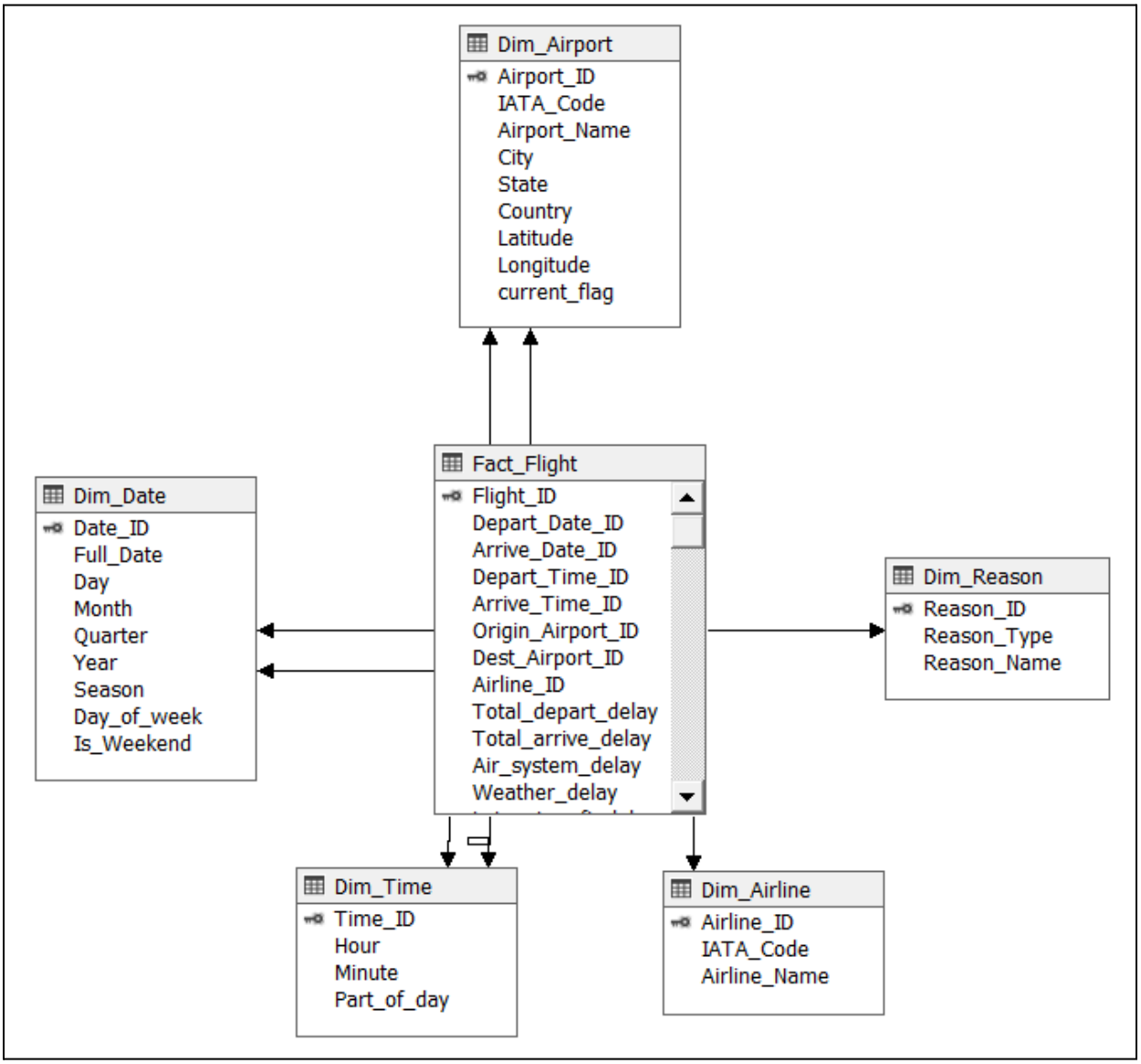
3. Cấu hình lập lịch

Job ETL được cấu hình chạy tự động theo lịch, ví dụ mỗi ngày vào khung giờ thấp điểm nhằm hạn chế ảnh hưởng đến hiệu năng hệ thống. Việc lập lịch giúp đảm bảo dữ liệu luôn được cập nhật kịp thời cho các dashboard và báo cáo BI.


Job Properties - Run_ETL
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Schedule list:
ID
Name
Enabl...
Description
10
Daily_1AM
Yes
Occurs every day at 2:55:00 PM. Schedule will be ...
Job Schedule Properties - Daily_1AM
Name:
Daily_1AM
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
12/30/2025
Time:
6:12:56 AM
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
2:55:00 PM
Occurs every:
1
hour(s)
Starting at:
2:55:00 PM
Ending at:
11:59:59 PM
Duration
Start date:
12/16/2025
End date:
12/30/2025
No end date:
Summary
Description:
Occurs every day at 2:55:00 PM. Schedule will be used starting on 12/16/2025.
Connection
Server:
ADMIN
Connection:
ADMIN\Duyen
View connection pr
Progress
Ready
essfully.

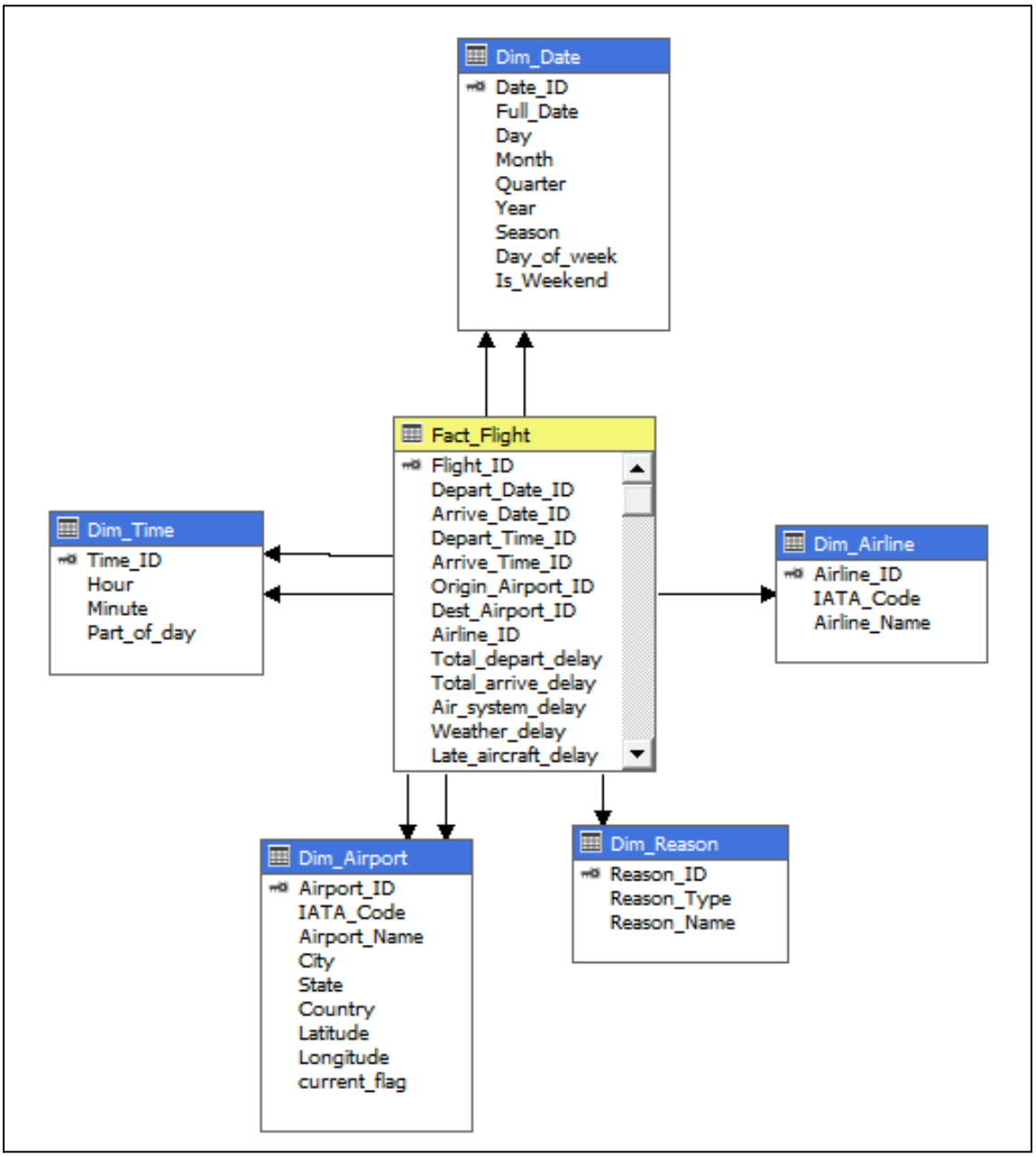
D. OLAP

I. Data Source View



Data Source View (DSV) được thiết kế nhằm mô hình hóa dữ liệu phục vụ cho việc phân tích và khai thác thông tin về hoạt động bay, đặc biệt tập trung vào các yếu tố ảnh hưởng đến tình trạng trễ chuyến. DSV được xây dựng theo mô hình Star Schema như trong DDS, trong đó Fact_Flight đóng vai trò là bảng sự kiện trung tâm, liên kết với các bảng chiều xung quanh.

II. Cube



Cube được thiết kế theo mô hình Star Schema với tâm điểm là bảng Fact_Flight kết nối với các bảng chiều (Dimensions) qua các khóa ngoại. Mô hình này tối ưu hóa việc truy vấn đa chiều và cho phép phân tích dữ liệu từ nhiều góc độ như thời gian (gồm ngày và khung giờ), địa điểm, hãng bay và nguyên nhân sự cố.

1. Dim_Date

- Đây là Role-playing Dimension (Một bảng đóng hai vai trò: Ngày đi và Ngày đến).
- Thuộc tính: Date ID (Key), Full Date, Year, Quarter (QuartersOfYear), Month (MonthsOfYear), Day (DaysOfMonth), Season, Is Weekend.
- Phân cấp chiều:

    - Hierarchy: Year -> Quarter -> Month -> Day. Cấu trúc này cho phép thao tác Drill-down từ tổng thể năm xuống chi tiết từng quý cho đến tháng hoặc ngày.
    - Hierarchy 1: Year -> Season -> Month -> Day. Cấu trúc này tương tự Hierarchy nhưng thay vì Roll-up theo Quý ta sẽ tổng hợp theo mùa (xuân/hạ/thu/đông).

- Ý nghĩa: Giúp phân tích hiệu suất bay theo thời gian.

2. **Dim_Airline**
    - Thuộc tính: Airline ID (Key), Airline Name, IATA Code.
    - Ý nghĩa: Cho phép so sánh hiệu suất vận hành giữa các hãng hàng không khác nhau.
3. **Dim_Reason**
    - Thuộc tính: Reason ID (Key), Reason Name, Reason Type.
    - Ý nghĩa: Giải thích lý do cụ thể của các sự cố (Thời tiết, Hãng bay, An ninh...).
4. **Dim_Airport**
    - Đây cũng là Role-playing Dimension lấy dữ liệu từ bảng Dim_Airport, đóng hai vai trò sân bay đi và sân bay đến.
    - Thuộc tính: Airport ID (Key), Airport Name, City, State, Country, Latitude, Longitude, IATA Code.
    - Ý nghĩa: Cho phép so sánh hiệu suất vận hành giữa các sân bay khác nhau.
5. **Dim_Time**
    - Đây cũng là Role-playing Dimension lấy dữ liệu từ bảng Dim_Time, đóng hai vai trò khung giờ khởi hành và khung giờ hạ cánh.
    - Thuộc tính: Hour, Minute, Part Of Day (Sáng, Trưa, Chiều, Tối).
    - Ý nghĩa: Phân tích xem các khung giờ nào trong ngày thường xuyên xảy ra tình trạng trễ chuyến nhất.
6. **Fact_Flight**
    - Các thuộc tính khoá:
        - Flight_ID: Khóa chính của bảng fact.
        - Depart_Date_ID, Arrive_Date_ID: Liên kết đến bảng Dim_Date.
        - Depart_Time_ID, Arrive_Time_ID: Liên kết đến bảng Dim_Time.
        - Origin_Airport_ID, Dest_Airport_ID: Liên kết đến bảng Dim_Airport.
        - Airline_ID: Liên kết đến bảng Dim_Airline.
        - Reason_ID: Liên kết đến bảng Dim_Reason.
    - Các measures:

- Fact Flight Count:
  - Tổng số chuyến bay (Hàm Count).
  - Addictive (có thể sum theo mọi chiều)
- Total Depart Delay:
  - Tổng số phút trễ khởi hành.
  - Addictive
- Total Arrive Delay:
  - Tổng số phút trễ hạ cánh.
  - Addictive
- OnTime Flight Count:
  - Số chuyến bay đúng giờ (tính từ Named Calculation).
  - Addictive
- Air System Delay:
  - Số phút trễ do hệ thống không lưu.
  - Addictive
- Airline Delay:
  - Số phút trễ do các hãng hàng không.
  - Addictive
- Weather Delay:
  - Số phút trễ do thời tiết.
  - Addictive
- Late Aircraft Delay:
  - Số phút trễ do vấn đề về phi cơ.
  - Addictive
- Security Delay:
  - Số phút trễ do vấn đề an ninh.
  - Addictive

III. MDX Queries

1. Truy vấn tổng số chuyến bay theo tháng, quý, năm.

a. Mô tả

Truy vấn này thực hiện thống kê tổng lượng chuyến bay thông qua các cấp độ phân cấp (Hierarchy) của chiều ngày (Depart_Date). Sử dụng các thao tác Roll-up để xem cái nhìn tổng quan theo Năm/Quý và Drill-down để chi tiết hóa biến động theo từng Tháng.

b. Ý nghĩa

Kết quả này đóng vai trò quan trọng trong việc xác định các giai đoạn cao điểm và hỗ trợ lập kế hoạch điều phối nguồn lực tại sân bay.

c. Kết quả

			Fact Flight Count
2015	1	1	9276
2015	1	2	8439
2015	1	3	9537
2015	2	4	9330
2015	2	5	9722
2015	2	6	10031
2015	3	7	10368
2015	3	8	10298
2015	3	9	9707
2015	4	11	9328
2015	4	12	9208

2. **Top 5 sân bay bận rộn nhất (có nhiều chuyến bay đi và đến nhất).**
    a. **Mô tả**

Truy vấn này thực hiện tạo một calculated measure là "Total Traffic" để tính toán tổng lưu lượng khai thác của từng sân bay bằng cách tổng hợp dữ liệu từ hai vai trò của chiều sân bay: điểm đi (Origin Airport) và điểm đến (Dest Airport). Sử dụng hàm TOPCOUNT, hệ thống lọc ra 5 thực thể có tổng lượng chuyến bay cao nhất.

    b. **Ý nghĩa**

Giúp nhận diện các trung tâm hàng không (Hubs) có mật độ khai thác lớn nhất, từ đó phân tích áp lực hạ tầng và khả năng chịu tải của các sân bay trọng điểm.

    c. **Kết quả**

	Total Traffic
Chicago O'Hare International Airport	56704
Dallas/Fort Worth International Airport	41422
Charlotte Douglas International Airport	39590
LaGuardia Airport (Marine Air Terminal)	35638
Newark Liberty International Airport	21412

3. **Tỉ lệ chuyến bay đúng giờ (On-Time Performance - OTP) ± 5 theo sân bay.**
    a. **Mô tả**

Truy vấn sử dụng chỉ số tính toán OTP Rate để đo lường hiệu suất đúng giờ bằng cách chia số lượng chuyến bay có thời gian trễ trong ngưỡng cho phép cho tổng số chuyến bay phục vụ. Dữ liệu được phân tách theo chiều Dest Airport để đánh giá hiệu quả tiếp nhận tàu bay của từng địa điểm.

    b. **Ý nghĩa**

Cung cấp KPI then chốt để đánh giá độ tin cậy của lịch trình bay, giúp các nhà quản lý xác định các khu vực thường xuyên xảy ra tình trạng chậm trễ.

    c. **Kết quả**

	OTP Rate
All	22.58%
Baltimore-Washington International Airport	25.99%
Charlotte Douglas International Airport	25.62%
Chicago O'Hare International Airport	20.31%
Dallas/Fort Worth International Airport	20.22%
LaGuardia Airport (Marine Air Terminal)	23.35%
Newark Liberty International Airport	23.67%

4. **Tỉ lệ hủy chuyến theo nguyên nhân.**
    a. **Mô tả**

Phân tích tỷ lệ phần trăm các lý do gây hủy chuyến (Thời tiết, Hãng bay...) bằng cách kết hợp bảng Fact_Flight và bảng chiều Dim_Reason.

    b. **Ý nghĩa**

Cho thấy cấu trúc các yếu tố gây ảnh hưởng tiêu cực đến hoạt động bay (như Thời tiết, Hãng bay, An ninh), hỗ trợ đưa ra các kịch bản ứng phó rủi ro phù hợp để hạn chế tỷ lệ huỷ chuyến.

    c. **Kết quả**

	Fact Flight Count	Cancellation Rate
Airline/Carrier	681	0.65%
National Air System	521	0.50%
Security	(null)	(null)
Weather	1820	1.73%

5. **Trung bình thời gian delay theo sân bay đi/đến**
    a. **Mô tả**

Truy vấn tính toán thời gian trễ trung bình tại hai đầu bến: khởi hành (Avg Depart Delay) và hạ cánh (Avg Arrive Delay).

    b. **Ý nghĩa**

Cho phép so sánh sự chênh lệch về thời gian trễ giữa các sân bay đi và đến, giúp nhận diện liệu sự chậm trễ bắt nguồn từ quản lý mặt đất tại điểm đi hay do điều hành không lưu tại điểm đến.

    c. **Kết quả**

	Avg Depart Delay	Avg Arrive Delay
Baltimore-Washington International Airport	10.4108891998473	7.3713268032057
Charlotte Douglas International Airport	10.4198029805506	8.10750189441778
Chicago O'Hare International Airport	15.9681151241535	9.50754796839729
Dallas/Fort Worth International Airport	11.5921008159915	6.15513495244073
LaGuardia Airport (Marine Air Terminal)	10.0219428699703	2.41326673775184
Newark Liberty International Airport	11.9824397534093	4.50046702783486

6. **Trung bình thời gian delay theo sân bay đi/đến**
    a. **Mô tả**

Câu truy vấn này tính và hiển thị xu hướng độ trễ đến trung bình (Average Arrival Delay) của các chuyến bay theo từng mùa và từng tháng.

    b. **Ý nghĩa**

Giúp nhận diện tính chu kỳ của sự chậm trễ. Ví dụ: Xác định xem các mùa thời tiết xấu (như mùa hè thường trời mưa nhiều) có gây ra mức độ trễ cao hơn đáng kể so với các tháng/mùa khác hay không.

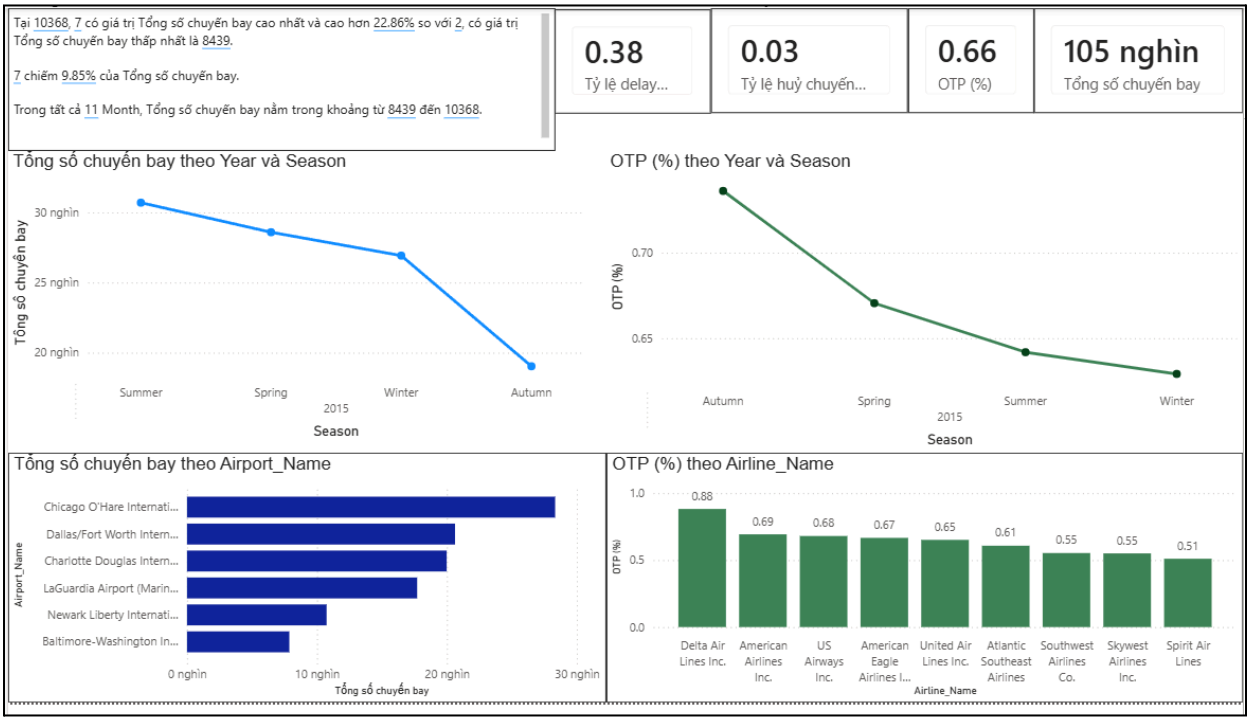
c. Kết quả

		Avg Delay
Autumn	9	-0.935922530132894
Autumn	10	0
Autumn	11	3.59423241852487
Spring	3	8.88707140610255
Spring	4	3.51521972132905
Spring	5	5.36659123637112
Summer	6	15.7709101784468
Summer	7	10.0928819444444
Summer	8	4.91444940765197
Winter	1	8.03438982319966
Winter	2	10.1187344472094
Winter	12	4.24815377932233

E. Dashboards

I. Báo cáo phân tích hoạt động chuyến bay và hiệu suất đúng giờ

1. Dashboard visualize



2. Tổng quan

Dashboard thể hiện bức tranh tổng thể về hoạt động khai thác chuyến bay, bao gồm khối lượng chuyến bay, mức độ đúng giờ (OTP), tỷ lệ trễ và hủy chuyến. Dữ liệu được phân tích theo thời gian (Year, Season), sân bay và hãng hàng không, giúp đánh giá cả quy mô hoạt động lẫn chất lượng vận hành.

3. Nhận xét chung

- Tổng số chuyến bay ở mức cao, tuy nhiên OTP chỉ đạt 66%, cho thấy còn dư địa cải thiện về đúng giờ.
- Lưu lượng chuyến bay biến động theo mùa, trong đó Summer có số chuyến cao nhất nhưng lại không có OTP tốt, phản ánh áp lực vận hành khi nhu cầu tăng.
- Autumn có ít chuyến bay hơn nhưng OTP cao hơn, cho thấy hiệu quả khai thác tốt khi mật độ bay giảm.
- Các sân bay trung tâm như Chicago O'Hare và Dallas/Fort Worth đóng vai trò then chốt, ảnh hưởng lớn đến tổng hiệu suất toàn hệ thống.
- Giữa các hãng hàng không tồn tại sự chênh lệch rõ ràng về OTP, trong đó Delta Air Lines thể hiện năng lực vận hành vượt trội so với phần còn lại.

4. Kết luận

Hoạt động khai thác chuyến bay chịu ảnh hưởng mạnh bởi mùa vụ và quy mô vận hành. Khi số chuyến bay tăng cao, hiệu suất đúng giờ có xu hướng giảm. Để nâng cao chất lượng dịch vụ, cần tập trung tối ưu vận hành trong mùa cao điểm, đặc biệt tại các sân bay lớn, đồng thời học hỏi mô hình quản lý và khai thác từ những hãng có OTP cao nhằm cải thiện hiệu quả toàn diện.

II. Báo cáo phân tích tình trạng delay chuyến bay

1. Dashboard visualize

2. Tổng quan

Dashboard tập trung phân tích tình trạng delay chuyến bay, phản ánh cả tần suất delay và mức độ delay thông qua tỷ lệ delay và thời gian delay trung bình. Dữ liệu được phân tích theo ngày và theo sân bay nhằm xác định các điểm phát sinh delay nổi bật.

3. Nhận xét chung

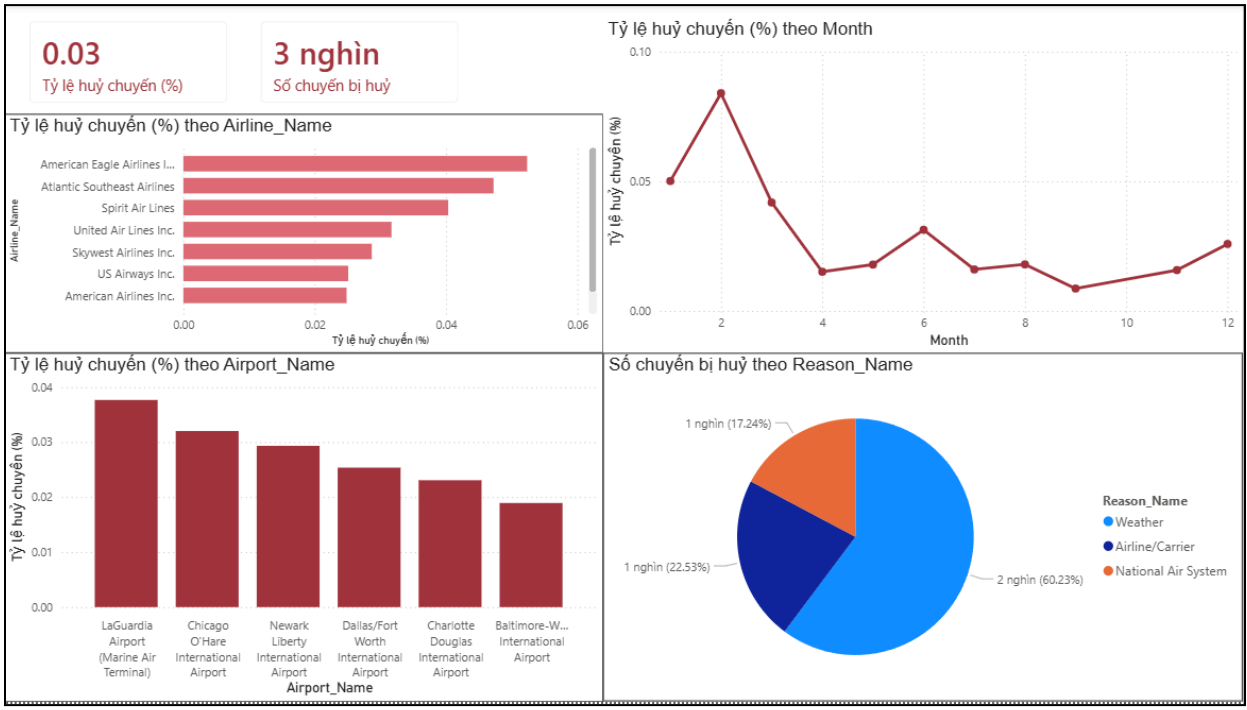
- Tỷ lệ delay >15 phút ở mức trung bình, cho thấy delay vẫn là vấn đề đáng chú ý trong vận hành.
- Delay không phân bố đều theo ngày mà có biến động rõ, cho thấy khả năng chịu ảnh hưởng từ lịch bay hoặc năng lực khai thác theo thời điểm.
- Một số sân bay lớn vừa có số chuyến delay cao vừa có thời gian delay dài, đặc biệt là Newark Liberty và Chicago O'Hare.
- Có sự khác biệt rõ ràng giữa các sân bay, cho thấy nguyên nhân delay mang tính cục bộ hơn là toàn hệ thống.

4. Kết luận

Dashboard cung cấp đầy đủ thông tin để đánh giá hiệu quả vận hành liên quan đến delay. Kết quả cho thấy cần ưu tiên cải thiện tại các sân bay có tỷ lệ và thời gian delay cao, đồng thời theo dõi sát các ngày cao điểm để giảm thiểu ảnh hưởng dây chuyền đến toàn mạng lưới chuyến bay.

III. Báo cáo phân tích tình trạng hủy chuyến bay

1. Dashboard visualize



2. Tổng quan

Dashboard thể hiện tình hình hủy chuyến bay thông qua các KPI tổng quan (tỷ lệ hủy, số chuyến bị hủy) và các biểu đồ phân tích theo thời gian (Month), hãng hàng không, sân bay và nguyên nhân hủy chuyến. Điều này giúp nhìn rõ cả quy mô lẫn nguyên nhân của vấn đề hủy chuyến.

3. Nhận xét chung

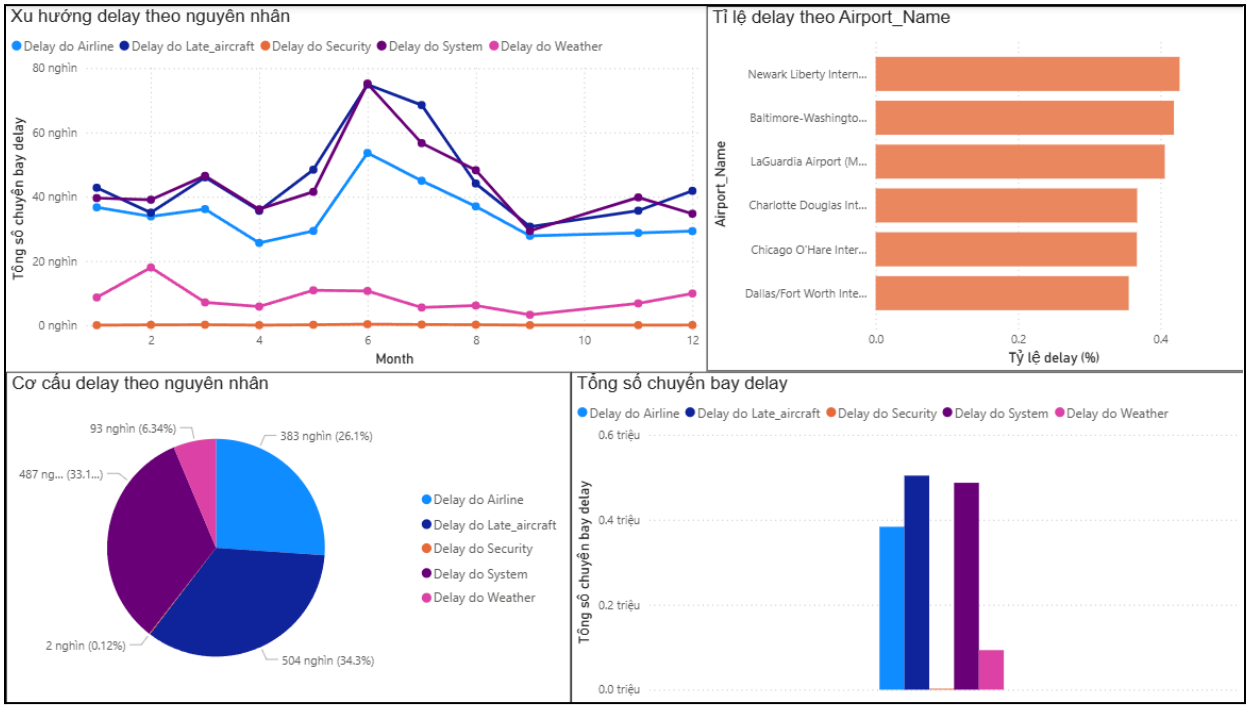
- Tỷ lệ hủy chuyến thấp (khoảng 3%), cho thấy hủy chuyến không xảy ra thường xuyên.
- Hủy chuyến không đồng đều theo tháng, có một số tháng cao điểm rõ rệt.
- Một số hãng hàng không có tỷ lệ hủy cao hơn mặt bằng chung, phản ánh sự khác biệt về năng lực vận hành.
- Các sân bay lớn như LaGuardia và Chicago O'Hare có tỷ lệ hủy cao hơn, có thể do mật độ bay cao.
- Nguyên nhân thời tiết chiếm tỷ trọng lớn nhất trong các chuyến bị hủy, vượt xa các nguyên nhân còn lại.

4. Kết luận

Hủy chuyến bay tuy chiếm tỷ lệ nhỏ nhưng chịu ảnh hưởng mạnh từ yếu tố thời tiết và điều kiện khai thác tại các sân bay lớn. Việc theo dõi sát xu hướng hủy chuyến theo thời gian và tập trung quản lý rủi ro thời tiết sẽ giúp giảm thiểu tác động đến hoạt động và trải nghiệm hành khách.

IV. Báo cáo phân tích nguyên nhân gây delay chuyến bay

1. Dashboard visualize



2. Tổng quan

Dashboard phân tích delay theo nguyên nhân gồm: Airline, Late Aircraft, System, Weather và Security. Các biểu đồ thể hiện cả xu hướng theo tháng, cơ cấu tỷ trọng, so sánh tỉ lệ delay theo sân bay và mức độ đóng góp của từng nguyên nhân, giúp xác định nguyên nhân trọng yếu gây delay.

3. Nhận xét chung

- Late Aircraft và System là hai nguyên nhân chiếm tỷ trọng lớn nhất trong tổng delay.
- Delay do Airline cũng chiếm tỷ lệ đáng kể, phản ánh vấn đề trong khâu vận hành nội bộ.
- Weather có tỷ trọng nhỏ hơn nhưng biến động theo mùa, tăng vào một số thời điểm nhất định.
- Security gần như không đáng kể so với các nguyên nhân còn lại.
- Các sân bay lớn như Chicago O'Hare và Dallas/Fort Worth đóng góp phần lớn tổng thời gian delay, cho thấy vai trò then chốt của các hub lớn trong phát sinh delay.

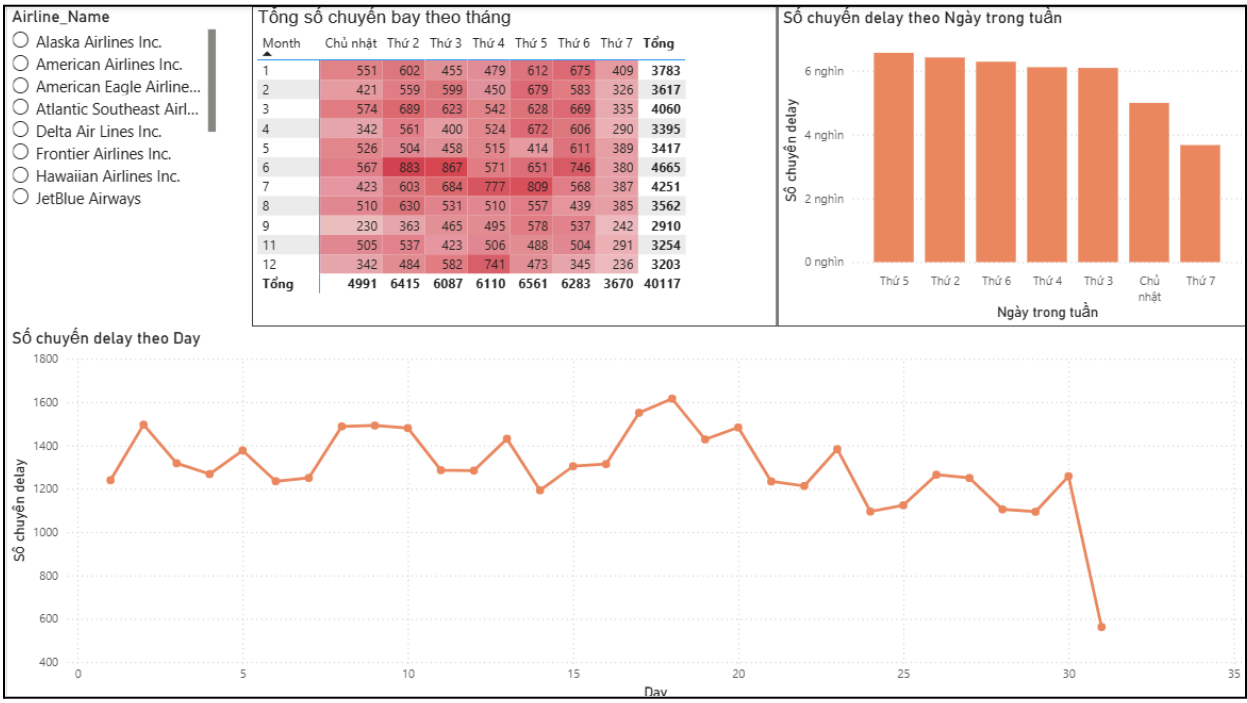
4. Kết luận

Phần lớn delay chuyến bay xuất phát từ nguyên nhân nội tại của hệ thống và dây chuyền khai thác, đặc biệt là trễ máy bay quay đầu và vấn đề hệ thống, hơn là các yếu tố bên ngoài. Do đó, việc cải thiện khả năng điều phối, tối ưu lịch bay và

giảm hiệu ứng dây chuyền tại các sân bay lớn sẽ mang lại hiệu quả cao trong việc giảm delay.

V. Báo cáo phân tích xu hướng delay theo ngày

1. Dashboard visualize



2. Tổng quan

Dashboard phân tích số chuyến delay theo ba mức thời gian: tháng, ngày trong tuần và ngày trong tháng. Ngoài ra, dashboard cho phép lọc theo hãng hàng không, giúp quan sát chi tiết hành vi delay của từng hãng theo thời gian.

3. Nhận xét chung

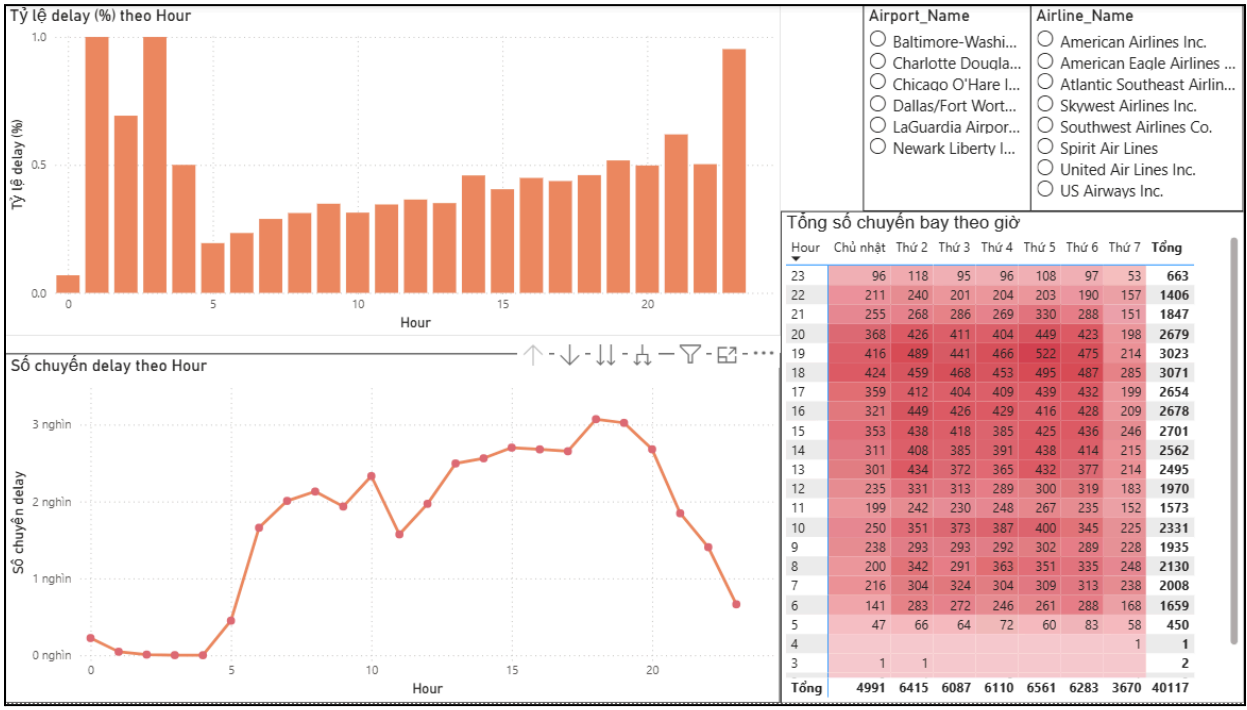
- Số chuyến delay phân bố không đều theo tháng, trong đó các tháng giữa năm có xu hướng cao hơn.
- Theo ngày trong tuần, delay tập trung nhiều vào các ngày trong tuần làm việc (Thứ 2 đến Thứ 6).
- Cuối tuần, đặc biệt là Chủ nhật, có số chuyến delay thấp hơn rõ rệt.
- Theo ngày trong tháng, delay dao động liên tục và xuất hiện một số ngày cao điểm, cho thấy ảnh hưởng từ lịch bay và mật độ khai thác.
- Việc có slicer theo hãng giúp linh hoạt phân tích và so sánh hành vi delay giữa các hãng hàng không.

4. Kết luận

Delay chuyến bay chịu ảnh hưởng rõ rệt từ yếu tố thời gian, đặc biệt là sự khác biệt giữa ngày thường và cuối tuần. Các giai đoạn có mật độ bay cao thường đi kèm với số chuyến delay nhiều hơn. Do đó, việc tối ưu lịch bay và phân bổ nguồn lực hợp lý theo từng thời điểm sẽ giúp giảm đáng kể tình trạng delay.

VI. Báo cáo phân tích xu hướng delay theo giờ

1. Dashboard visualize



Hour	Chủ nhật	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7	Tổng
23	96	118	95	96	108	97	53	663
22	211	240	201	204	203	190	157	1406
21	255	268	286	269	330	288	151	1847
20	368	426	411	404	449	423	198	2679
19	416	489	441	466	522	475	214	3023
18	424	459	468	453	495	487	285	3071
17	359	412	404	409	439	432	199	2654
16	321	449	426	429	416	428	209	2678
15	353	438	418	385	425	436	246	2701
14	311	408	385	391	438	414	215	2562
13	301	434	372	365	432	377	214	2495
12	235	331	313	289	300	319	183	1970
11	199	242	230	248	267	235	152	1573
10	250	351	373	387	400	345	225	2331
9	238	293	293	292	302	289	228	1935
8	200	342	291	363	351	335	248	2130
7	216	304	324	304	309	313	238	2008
6	141	283	272	246	261	288	168	1659
5	47	66	64	72	60	83	58	450
4							1	1
3	1	1						2
Tổng	4991	6415	6087	6110	6561	6283	3670	40117

2. Tổng quan

Biểu đồ "Tỷ lệ delay (%) theo Hour" thể hiện xác suất xảy ra delay tại từng khung giờ trong ngày. Khác với biểu đồ số chuyến delay, biểu đồ này phản ánh mức độ rủi ro delay, không phụ thuộc vào số lượng chuyến bay nhiều hay ít.

3. Nhận xét chung

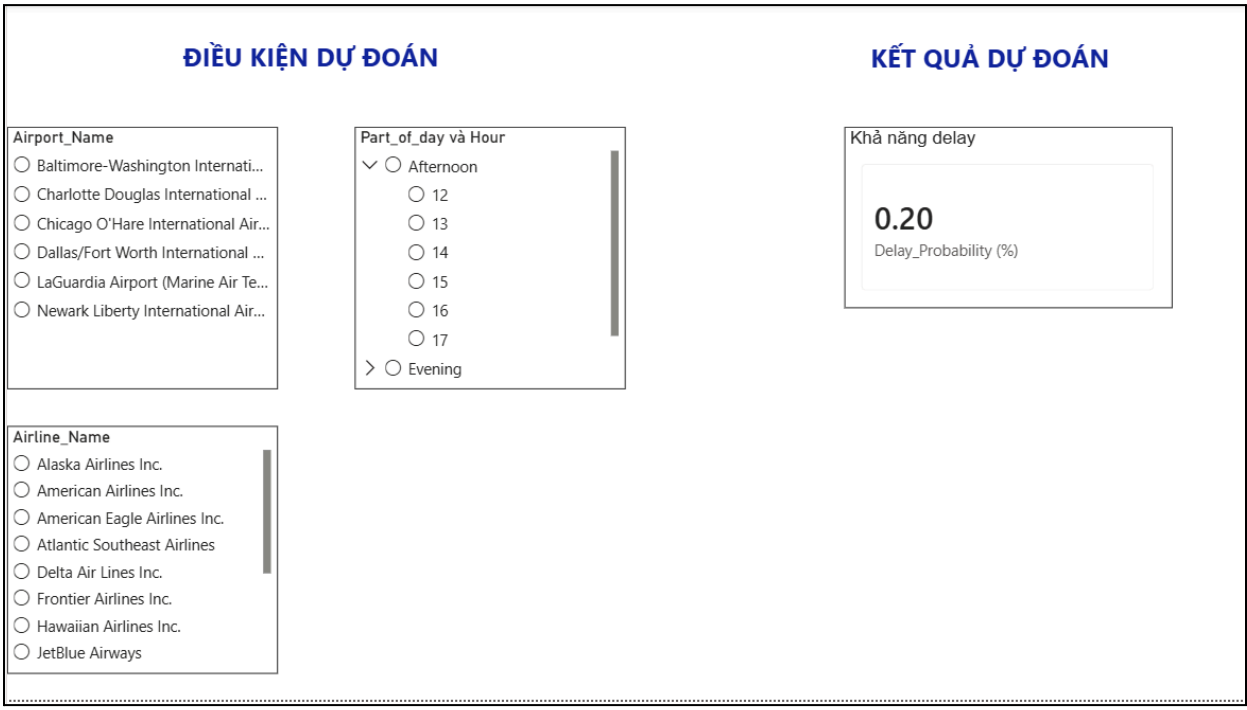
- Tỷ lệ delay cao vào các khung giờ rất sớm (0–2h) và cuối ngày (sau 20h).
- Trong khung giờ sáng sớm đến trưa (5h–12h), tỷ lệ delay ở mức thấp và ổn định hơn.
- Từ chiều đến đầu tối (13h–19h), tỷ lệ delay tăng dần, cho thấy ảnh hưởng của việc dồn chuyến và delay dây chuyền.
- Mặc dù một số giờ có ít chuyến bay, tỷ lệ delay vẫn cao, phản ánh rủi ro vận hành theo thời điểm, không chỉ phụ thuộc vào lưu lượng bay.

4. Kết luận

Biểu đồ cho thấy delay chuyến bay không chỉ phụ thuộc vào số lượng chuyến mà còn chịu ảnh hưởng lớn từ thời điểm trong ngày. Các khung giờ sớm và muộn có rủi ro delay cao hơn, do đó cần được ưu tiên theo dõi và điều phối nguồn lực để giảm khả năng phát sinh delay.

VII. Báo cáo dự đoán khả năng delay

1. Dashboard visualize



2. Mục tiêu

Dashboard được xây dựng nhằm **ước lượng khả năng một chuyến bay bị delay** dựa trên dữ liệu lịch sử, hỗ trợ người dùng **chủ động đánh giá rủi ro** trước khi chuyến bay diễn ra.

3. Phương pháp

Khả năng delay được ước lượng dựa trên tỷ lệ delay đã xảy ra trong quá khứ đối với các chuyến bay có cùng điều kiện khai thác. Phương pháp này thuộc nhóm predictive analytics trong Business Intelligence, không sử dụng mô hình machine learning phức tạp mà tập trung vào xác suất có điều kiện.

Các điều kiện được sử dụng bao gồm:

- Sân bay xuất phát (Airport)
- Hãng hàng không (Airline)
- Thời điểm bay (buổi trong ngày và giờ cụ thể) (Part of day, hour)

Delay_Probability (%) được dùng để ước lượng khả năng chuyến bay bị delay dựa trên dữ liệu lịch sử. Công thức tính bằng tỷ lệ giữa số chuyến bay bị delay và tổng số chuyến bay, sau khi áp dụng các điều kiện lọc như sân bay, thời điểm bay hoặc hãng hàng không.

```
Delay_Probability (%) =
DIVIDE (
```

```
    CALCULATE (
        COUNTROWS ( Fact_Flight ),
        Fact_Flight[IsDelay] = 1
    ),
    COUNTROWS ( Fact_Flight )
)
```

Giá trị thu được nằm trong khoảng từ 0 đến 1 và được hiển thị dưới dạng phần trăm để thể hiện xác suất chuyến bay bị delay.

4. Nhận xét

Khả năng delay thay đổi rõ rệt theo sân bay và thời điểm bay, cho thấy đây là các yếu tố ảnh hưởng mạnh đến rủi ro delay.

Dashboard cho phép người dùng linh hoạt thay đổi điều kiện để quan sát sự biến động của xác suất delay, từ đó so sánh giữa các kịch bản khác nhau.

Cách trình bày đơn giản, tập trung vào kết quả giúp người dùng nhanh chóng nắm bắt thông tin quan trọng.

Dashboard dự đoán khả năng delay giúp chuyển hoạt động phân tích từ **mô tả** sang **dự báo**, hỗ trợ ra quyết định trong công tác điều phối và lập kế hoạch chuyến bay. Phương pháp dựa trên dữ liệu lịch sử tuy đơn giản nhưng phù hợp với mục tiêu Business Intelligence và đáp ứng yêu cầu của đề bài.

F. Data mining

1. Nguồn dữ liệu DDS:

Dữ liệu được sử dụng trong bài toán Data Mining là dữ liệu từ DDS (Dimensional Data Store) – tầng dữ liệu đã được ETL, làm sạch và chuẩn hóa theo mô hình sao (Star Schema).

Trong mô hình sao này:

- Fact_Flight là bảng trung tâm, mỗi dòng tương ứng với một chuyến bay thực tế.
- Fact_Flight chứa các chỉ số trễ quan trọng như:
    - Total_depart_delay
    - Total_arrive_delay
    - Các nguyên nhân trễ: Weather_delay, Air_system_delay, Late_aircraft_delay, Airline_delay, Security_delay

Ngoài ra, dữ liệu được mở rộng bằng cách join thêm các bảng dimension:

- Dim_Airline: lấy tên hãng bay
- Dim_Airport: lấy sân bay đi và sân bay đến
- Dim_Date: lấy tháng, thứ trong tuần, thông tin cuối tuần
- Dim_Time: lấy giờ cất cánh

Việc sử dụng DDS thay vì dữ liệu CSV giúp:

- Dữ liệu đã sạch và nhất quán nghiệp vụ
- Định nghĩa khóa rõ ràng, join ổn định
- Dễ mở rộng và tái sử dụng cho BI và Data Mining

2. Mục tiêu bài toán và định nghĩa "chuyến bay trễ"

Bài toán Data Mining tập trung vào phân tích và dự đoán hiện tượng chậm chuyến bay (Flight Delay).

Một chuyến bay được định nghĩa là trễ khi thỏa mãn đồng thời các điều kiện:

- Total_depart_delay > 15 phút
- Không bị hủy (Canceled_Flag = 0)
- Không bị chuyển hướng (Diverted_Flag = 0)

Dựa trên định nghĩa này, biến mục tiêu cho bài toán phân loại được xây dựng:

- Delay_Flag = 1: chuyến bay trễ
- Delay_Flag = 0: chuyến bay không trễ

Bài toán Data Mining bao gồm:

- Classification: dự đoán trễ / không trễ
- Regression: dự đoán số phút trễ
- Clustering: phân nhóm hành vi trễ chuyến bay

3. Trích xuất dữ liệu từ SQL Server

Dữ liệu được trích xuất trực tiếp từ SQL Server thông qua câu lệnh SQL.
Các nhóm cột được lựa chọn bao gồm:

- Nhóm định danh: Flight_ID
- Nhóm hãng bay: Airline_Name (từ Dim_Airline)
- Nhóm sân bay: Origin_Airport, Dest_Airport (từ Dim_Airport)
- Nhóm thời gian:
    - Month

    - Day_of_week
    - Is_Weekend (từ Dim_Date)
    - Depart_Hour (từ Dim_Time)
- Nhóm biến trễ:
    - Total_depart_delay
    - Total_arrive_delay
    - Các nguyên nhân trễ
- Biến mục tiêu: Delay_Flag (theo rule 15 phút)

Ngoài ra, các cột Canceled_Flag và Diverted_Flag vẫn được giữ lại để xử lý nhãn đúng và phục vụ các bước lọc dữ liệu sau này.

4. Load dữ liệu và tiền xử lý ban đầu

Dữ liệu được load vào Python bằng pd.read_sql() và lưu dưới dạng DataFrame.

- df.head() được sử dụng để kiểm tra nhanh cấu trúc dữ liệu, bao gồm hãng bay, sân bay, thời gian và các cột delay.
- df.shape cho biết kích thước tập dữ liệu là (105,244 dòng × 18 cột), tương ứng với 105,244 chuyến bay.

5. Làm sạch dữ liệu dạng text

Trong dữ liệu thực tế, các cột text như tên hãng bay hoặc mã sân bay có thể chứa khoảng trắng dư hoặc ký tự xuống dòng.

Nếu không xử lý, các giá trị như:

- "American Airlines"
- "American Airlines "

sẽ bị coi là hai category khác nhau khi one-hot encoding, gây sai lệch mô hình.

Do đó, các cột:

- Airline_Name
- Origin_Airport
- Dest_Airport

được xử lý bằng: .astype(str).str.strip()

Sau khi làm sạch, dữ liệu được kiểm tra lại bằng df.head() để đảm bảo giá trị nhất quán.

6. Xây dựng tập đặc trưng (Feature Engineering)

Các biến đầu vào được chia thành hai nhóm:

a. Biến phân loại (Categorical)

- Airline_Name
- Origin_Airport
- Dest_Airport
- Day_of_week

Các biến này được xử lý bằng One-Hot Encoding.

b. Biến số (Numerical)

- Month
- Is_Weekend
- Depart_Hour
- Các biến delay theo nguyên nhân

Các biến số được:

- Ép kiểu bằng pd.to_numeric(errors="coerce")
- Xử lý missing bằng median để tránh ảnh hưởng của outlier

Các dòng thiếu biến phân loại hoặc thiếu Delay_Flag được loại bỏ vì không thể dùng để huấn luyện mô hình.

7. Classification – Random Forest

a. Chia train / test

- X: toàn bộ feature
- y: Delay_Flag
- Tỷ lệ chia: 80% train – 20% test
- Sử dụng stratify=y để đảm bảo tỷ lệ trễ/không trễ đồng đều

Kết quả:

- 84,195 chuyến bay dùng để train
- 21,049 chuyến bay dùng để test
- 15 feature đầu vào

b. Pipeline tiền xử lý và huấn luyện

Pipeline bao gồm:

- Categorical:

    - Imputer: most_frequent
    - OneHotEncoder
- Numerical:
    - Imputer: median
- Model: RandomForestClassifier

Pipeline giúp đảm bảo dữ liệu mới khi dự đoán sẽ được xử lý giống hệt lúc train.

c. Đánh giá mô hình Classification

Sau khi huấn luyện:

- Mô hình sinh ra xác suất Delay_Flag = 1
- Threshold mặc định ban đầu là 0.5

Kết quả đánh giá: ROC-AUC ≈ 0.969, cho thấy mô hình phân biệt rất tốt giữa hai lớp trễ và không trễ.

Confusion Matrix được dùng để phân tích:

- True Negative: không trễ và đoán đúng
- False Positive: báo động giả
- False Negative: bỏ sót chuyến bay trễ
- True Positive: trễ và đoán đúng

```
ROC-AUC: 0.9686552185829252
Confusion matrix @0.5:
 [[16547   242]
 [  889  3371]]
```

d. Classification Report & Threshold tối ưu

Classification report cung cấp:

- Precision
- Recall
- F1-score
- Support

Do bài toán ưu tiên không bỏ sót chuyến bay trễ, threshold được tinh chỉnh theo F1-score. Best threshold ≈ 0.53, cho kết quả cân bằng tốt hơn so với 0.5.

```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     16789
           1       0.93      0.79      0.86      4260

    accuracy                           0.95     21049
   macro avg       0.94      0.89      0.91     21049
weighted avg       0.95      0.95      0.94     21049

Best threshold: 0.530790690154376
```

e. Xuất kết quả Classification

Toàn bộ dataset được dự đoán và xuất ra CSV với các cột:

- Delay_Probability
- Delay_Pred

File này được dùng cho báo cáo và Power BI.

8. Regression – Dự đoán số phút trễ

Mục tiêu hồi quy là dự đoán Total_depart_delay.

- Loại bỏ các chuyến bay bị hủy hoặc diverted
- Sử dụng RandomForestRegressor

Kết quả đánh giá:

- MAE ≈ 5.55 phút
- MSE ≈ 87
- $R^2 \approx 0.945$

Mô hình giải thích được khoảng 94.5% biến động độ trễ, cho thấy khả năng dự đoán rất tốt.

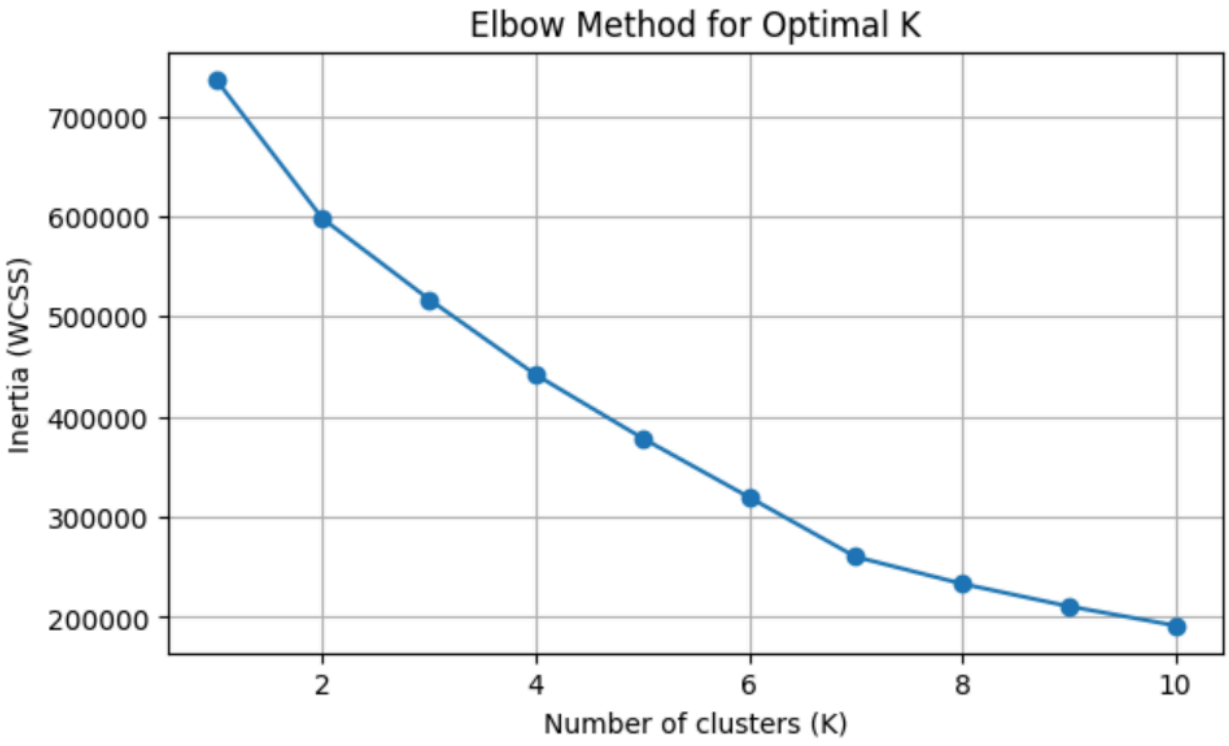
9. Clustering – Phân nhóm hành vi trễ

a. Chuẩn hóa và chọn K

Các biến delay theo nguyên nhân và giờ bay được chuẩn hóa bằng StandardScaler.

Số cụm được xác định bằng Elbow Method, chạy K từ 1 đến 10.

- Điểm gãy rõ nhất tại K = 3
- Dễ diễn giải nghiệp vụ



b. Ý nghĩa các cụm

Cluster 1 – Nhóm không trễ hoặc trễ rất nhẹ:

- Số chuyến bay: 98.978
- Total_depart_delay trung bình: ≈ 4.3 phút
- Các loại delay (Weather, Air System, Airline, Security): rất thấp, gần bằng 0
- Giờ khởi hành trung bình: ≈ 12.6 giờ
- Mức trễ trung bình khoảng 4 phút được xem là không đáng kể trong vận hành hàng không, thường phát sinh từ các yếu tố kỹ thuật nhỏ hoặc quy trình mặt đất thông thường. Việc tất cả các thành phần delay đều ở mức rất thấp cho thấy các chuyến bay trong cụm này không gặp sự cố đáng kể về thời tiết, hệ thống hay an ninh.

- Cluster này chiếm phần lớn dữ liệu, phản ánh trạng thái hoạt động bình thường của hệ thống hàng không.
- Cluster 1 đại diện cho các chuyến bay hoạt động ổn định, đúng giờ hoặc chỉ trễ rất nhẹ. Đây là nhóm có kích thước lớn nhất và phản ánh hành vi vận hành chuẩn của hệ thống.

Cluster 0 – Nhóm trễ nặng do nhiều nguyên nhân cộng dồn

- Số chuyến bay: 6.240
- Total_depart_delay trung bình: ≈ 138 phút
- Late_aircraft_delay: ≈ 60 phút
- Airline_delay: ≈ 34 phút
- Air_system_delay: ≈ 33 phút
- Weather_delay: ≈ 11 phút
- Giờ khởi hành trung bình: ≈ 15.0 giờ
- Với tổng thời gian trễ trung bình hơn 2 giờ, đây là nhóm các chuyến bay bị trễ nghiêm trọng. Đặc điểm nổi bật của cluster này là không có một nguyên nhân trễ duy nhất chi phối, mà là sự cộng dồn của nhiều yếu tố.
- Trong đó, Late Aircraft Delay chiếm tỷ trọng lớn nhất, cho thấy hiện tượng hiệu ứng dây chuyền trong lịch bay: chuyến bay trước đến muộn kéo theo trễ cho các chuyến sau. Bên cạnh đó, các yếu tố từ hãng bay và hệ thống không lưu cũng đóng vai trò quan trọng, phản ánh các vấn đề vận hành và kỹ thuật phức tạp.
- Cluster 0 đại diện cho nhóm chuyến bay trễ nghiêm trọng, thường do nhiều sự cố vận hành và kỹ thuật xảy ra đồng thời, gây ảnh hưởng lớn đến lịch trình bay và trải nghiệm của hành khách.

Cluster 2 – Nhóm trễ đặc biệt do an ninh

- Số chuyến bay: 26
- Total_depart_delay trung bình: ≈ 59 phút
- Security_delay: ≈ 42 phút
- Các nguyên nhân trễ khác: gần bằng 0
- Giờ khởi hành trung bình: ≈ 13.7 giờ
- Cluster này có số lượng chuyến bay rất nhỏ, cho thấy đây là các trường hợp hiếm gặp. Phần lớn thời gian trễ xuất phát từ Security Delay, trong khi các yếu tố khác hầu như không đáng kể. Điều này cho thấy các chuyến bay trong nhóm này bị ảnh hưởng bởi các sự cố an ninh đặc thù, chẳng hạn như kiểm tra an ninh tăng cường hoặc sự kiện bất thường.

- Do tần suất xuất hiện thấp và nguyên nhân mang tính ngoại lệ, cluster này không phản ánh hành vi vận hành chung, mà được xem là các outlier trong dữ liệu.
- Cluster 2 đại diện cho các trường hợp trễ đặc biệt liên quan đến an ninh, mang tính hiếm và chiếm tỷ lệ rất nhỏ trong toàn bộ tập dữ liệu.

```
=== Cluster Summary (Mean) ===
               Total_depart_delay  Weather_delay  Air_system_delay  \
Delay_Cluster
0                         138.136         11.322            33.176
1                           4.286          0.227             2.826
2                          59.346          0.000            11.577

               Late_aircraft_delay  Airline_delay  Security_delay  Depart_Hour
Delay_Cluster
0                           59.952         33.941           0.000       15.017
1                            1.309          1.734           0.007       12.619
2                            8.577          1.154          42.346       13.731

=== Cluster Size ===
Delay_Cluster
0     6240
1    98978
2       26
Name: count, dtype: int64
```