HURTOWNIE DANYCH

Projekt

Maciej Kopiński 254578

Projekt – etap II (12.05./24.05.2022 r.)

Proces ETL

1. Utworzone tabele w poprzednim punkcie wypełnić danymi zgodnie z ustalonymi założeniami projektowymi wykorzystując zapytania SQL lub inne narzędzia dostępne w Integration Services.

Przy ocenie będą brane następujące elementy pakietu(ów):

- właściwa struktura procesu ETL (odpowiednie rozbicie procesu ETL na zadania/pakiety, dobrze dobrane nazwy poszczególnych zadań, wprowadzona automatyzacja, obsługa błędów, itp.)
- stabilność i prawidłowe, bezbłędne wykonanie
- złożoność przeprowadzonych operacji. Przykładowo, jeżeli dane źródłowe już są w pełni zdenormalizowane proszę nie spodziewać się maksymalnej liczby punktów za ten element
- dokumentacja powinna zawierać krótki opis dotyczący każdego zadania, które pozwoli zorientować się, jaki jest jego cel (np. zadanie Z kopiuje dane z tabeli X i Y do tabeli T dokonując denormalizacji) oraz mapę logiczną procesu ETL.

Rozwiązania:

Kwerendy:

```
--DROP TABLES
IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'FACT_ACCIDENTS'
AND TABLE_SCHEMA = 'dbo')
DROP TABLE FACT_ACCIDENTS;

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'DIM_CONDITIONS'
AND TABLE_SCHEMA = 'dbo')
DROP TABLE DIM_CONDITIONS;

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'DIM_PLACE' AND
TABLE_SCHEMA = 'dbo')
DROP TABLE DIM_PLACE;

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'DIM_PLANE' AND
TABLE_SCHEMA = 'dbo')
DROP TABLE DIM_PLANE;

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'DIM_TIME' AND
TABLE_SCHEMA = 'dbo')
DROP TABLE DIM_TIME;
```

```sql
IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'DIM_ACCIDENT'
AND TABLE_SCHEMA = 'dbo')
DROP TABLE DIM_ACCIDENT;

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Dni' AND
TABLE_SCHEMA = 'dbo')
DROP TABLE Dni;

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Miesiace' AND
TABLE_SCHEMA = 'dbo')
DROP TABLE Miesiace;



--CREATE TABLES
CREATE TABLE [dbo].[DIM_ACCIDENT](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Investigation_Type] [nvarchar](50) NOT NULL,
	[Injury_Severity] [nvarchar](15) NULL,
	[Aircraft_Damage] [nvarchar](15) NULL,
	[FAR_Description] [nvarchar](200) NULL,
	[Schedule] [nvarchar](10) NULL,
	[Purpose_Of_Flight] [nvarchar](50) NULL,
	[Air_Carrier] [nvarchar](100) NULL,
	[Broad_Phase_Of_Flight] [nvarchar](20) NULL
);

CREATE TABLE [dbo].[DIM_CONDITIONS](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Weather_Condition] [nvarchar](50) NOT NULL,
	[Weather_Condition_Code] [nvarchar](10) NOT NULL
);

CREATE TABLE [dbo].[DIM_PLACE](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Location] [nvarchar](60) NOT NULL,
	[Country] [nvarchar](30) NOT NULL,
	[Region] [nvarchar](15) NULL,
	[Airport_Code] [nvarchar](10) NULL,
	[Airport_Name] [nvarchar](100) NULL,
	[Region_Code] [nvarchar](5) NULL
);
CREATE TABLE [dbo].[DIM_PLANE](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Make] [nvarchar](50) NULL,
	[Model] [nvarchar](50) NULL,
	[Amateur_Built] [nvarchar](50) NULL,
	[Number_Of_Engines] [int] NULL,
	[Engine_Type] [nvarchar](50) NULL,
	[Aircraft_Category] [nvarchar](15) NULL
);
CREATE TABLE [dbo].[DIM_TIME](
	[PK_TIME] [int] NOT NULL,
	[Year] [int] NOT NULL,
	[Quarter] [int] NOT NULL,
	[Month] [int] NOT NULL,
	[Month_In_Words] [nvarchar](15) NOT NULL,
	[Day] [int] NOT NULL,
	[Day_In_Words] [nvarchar](15) NOT NULL
)
```

```sql
CREATE TABLE [dbo].[FACT_ACCIDENTS](
	[Accident_Id] [int] NOT NULL,
	[Event_Date] [int] NOT NULL,
	[Place_Id] [int] NOT NULL,
	[Plane_Id] [int] NOT NULL,
	[Weather_Conditions_Id] [int] NOT NULL,
	[Total_Fatal_Injuries] [int] NOT NULL,
	[Total_Serious_Injuries] [int] NOT NULL,
	[Total_Minor_Injuries] [int] NOT NULL,
	[Total_Uninjured] [int] NOT NULL,
	[Total_Injured] [int] NOT NULL,
	[Mortality] [decimal](18, 15) NOT NULL
);

SET DATEFIRST 1;
SELECT DISTINCT DATEPART(dw, Event_Date) Numer,
	CASE
		WHEN DATEPART(dw, Event_Date) = 1 THEN 'Poniedzialek'
		WHEN DATEPART(dw, Event_Date) = 2 THEN 'Wtorek'
		WHEN DATEPART(dw, Event_Date) = 3 THEN 'Sroda'
		WHEN DATEPART(dw, Event_Date) = 4 THEN 'Czwartek'
		WHEN DATEPART(dw, Event_Date) = 5 THEN 'Piatek'
		WHEN DATEPART(dw, Event_Date) = 6 THEN 'Sobota'
		WHEN DATEPART(dw, Event_Date) = 7 THEN 'Niedziela'
	END Nazwa
INTO Dni
FROM Aviation_Data;

SELECT DISTINCT MONTH(Event_Date) Numer,
	CASE
		WHEN MONTH(Event_Date) = 1 THEN 'Styczen'
		WHEN MONTH(Event_Date) = 2 THEN 'Luty'
		WHEN MONTH(Event_Date) = 3 THEN 'Marzec'
		WHEN MONTH(Event_Date) = 4 THEN 'Kwiecien'
		WHEN MONTH(Event_Date) = 5 THEN 'Maj'
		WHEN MONTH(Event_Date) = 6 THEN 'Czerwiec'
		WHEN MONTH(Event_Date) = 7 THEN 'Lipiec'
		WHEN MONTH(Event_Date) = 8 THEN 'Sierpien'
		WHEN MONTH(Event_Date) = 9 THEN 'Wrzesien'
		WHEN MONTH(Event_Date) = 10 THEN 'Pazdziernik'
		WHEN MONTH(Event_Date) = 11 THEN 'Listopad'
		WHEN MONTH(Event_Date) = 12 THEN 'Grudzien'
	END Nazwa
INTO Miesiace
FROM Aviation_Data;


--REFERENCES
ALTER TABLE DIM_CONDITIONS
ADD CONSTRAINT CONDITIONS_PRIMARY_KEY UNIQUE(Id), PRIMARY KEY(Id);

ALTER TABLE DIM_PLACE
ADD CONSTRAINT PLACE_PRIMARY_KEY UNIQUE(Id), PRIMARY KEY(Id);
ALTER TABLE DIM_PLANE
ADD CONSTRAINT PLANE_PRIMARY_KEY UNIQUE(Id), PRIMARY KEY(Id);

ALTER TABLE DIM_TIME
ADD CONSTRAINT TIME_PRIMARY_KEY UNIQUE(PK_TIME), PRIMARY KEY(PK_TIME);

ALTER TABLE DIM_ACCIDENT
ADD CONSTRAINT ACCIDENT_PRIMARY_KEY UNIQUE(Id), PRIMARY KEY(Id);
```

```sql
ALTER TABLE FACT_ACCIDENTS
ADD CONSTRAINT CONDITIONS_FOREIGN_KEY FOREIGN KEY (Weather_Conditions_Id) REFERENCES
DIM_CONDITIONS(Id),
	CONSTRAINT PLANE_FOREIGN_KEY FOREIGN KEY (Plane_Id) REFERENCES DIM_PLANE(Id),
	CONSTRAINT PLACE_FOREIGN_KEY FOREIGN KEY (Place_Id) REFERENCES DIM_Place(Id),
	CONSTRAINT ACCIDENT_FOREIGN_KEY FOREIGN KEY (Accident_Id) REFERENCES
DIM_ACCIDENT(Id),
	CONSTRAINT EVENT_DATE_FOREIGN_KEY FOREIGN KEY (Event_Date) REFERENCES
DIM_TIME(PK_TIME);



--INSERT
WITH Accident (Investigation_Type, Injury_Severity, Aircraft_Damage, FAR_Description,
Schedule, Purpose_Of_Flight, Air_Carrier, Broad_Phase_Of_Flight)
AS (
	SELECT Investigation_Type, Injury_Severity, Aircraft_damage, FAR_Description,
Schedule, Purpose_of_flight, Air_carrier, Broad_phase_of_flight
FROM
(
	SELECT DISTINCT Investigation_Type, Injury_Severity, Aircraft_damage,
	FAR_Description, Schedule, Purpose_of_flight,
	Air_carrier,
	Broad_phase_of_flight
	FROM Aviation_Data
) P
) INSERT INTO DIM_ACCIDENT (Investigation_Type, Injury_Severity, Aircraft_Damage,
FAR_Description, Schedule, Purpose_Of_Flight, Air_Carrier, Broad_Phase_Of_Flight)
SELECT * FROM Accident;

WITH Conditions (Weather_Condition, Weather_Condition_Code)
AS (
	 SELECT CASE
		WHEN Weather_Conditions = 'VMC' THEN 'Good conditions'
		WHEN Weather_Conditions = 'UNK' OR Weather_Conditions='' THEN 'Unknown
conditions'
		WHEN Weather_Conditions = 'IMC' THEN 'Bad conditions'
	END,
	Weather_Conditions
	FROM
	(
		SELECT DISTINCT
		CASE
			WHEN Weather_Condition = 'VMC' THEN 'VMC'
			WHEN Weather_Condition = 'UNK' OR Weather_Condition='' THEN 'UNK'
			WHEN Weather_Condition = 'IMC' THEN 'IMC'
		END Weather_Conditions
		FROM Aviation_Data
	) A
) INSERT INTO DIM_CONDITIONS (Weather_Condition, Weather_Condition_Code)
SELECT * FROM Conditions;
```

```
WITH Place ([Location], Country, Region, Airport_Code, Airport_Name, Region_Code)
AS (
	SELECT DISTINCT
	 Location, Country, US_State, Airport_Code, Airport_Name,
CASE
	WHEN SUBSTRING([Location], LEN([Location]) - 2, 1)='-' THEN
SUBSTRING([Location], LEN([Location]) - 1, 2)
	END
	FROM Aviation_Data
	LEFT JOIN USState_Codes ON USState_Codes.Abbreviation = SUBSTRING([Location],
LEN([Location]) - 1, 2)
) INSERT INTO DIM_PLACE ([Location], Country, Region, Airport_Code, Airport_Name,
Region_Code)
SELECT * FROM Place;

WITH Plane (Make, Model, Amateur_Built, Number_Of_Engines, Engine_Type,
Aircraft_Category)
AS (
	SELECT Make, Model, Amateur_Built, Number_Of_Engines, Engine_Type,
Aircraft_Category
	FROM (
		SELECT DISTINCT Make, Model, Amateur_Built, Number_Of_Engines,
Engine_Type, Aircraft_Category
		FROM Aviation_Data
	) A
) INSERT INTO DIM_PLANE (Make, Model, Amateur_Built, Number_Of_Engines, Engine_Type,
Aircraft_Category)
SELECT * FROM Plane;

DECLARE @D INT;
SET @D = (SELECT TOP 1 DATEPART(YYYY, Event_Date) * 10000 + DATEPART(MM, Event_Date) *
100 + DATEPART(DD, Event_Date) FROM Aviation_Data ORDER BY 1);
DECLARE @COUNTER DATE;
SET @COUNTER = CONVERT(date, CAST(@D AS nvarchar));
DECLARE @END INT;
SET @END = (SELECT TOP 1 DATEPART(YYYY, Event_Date) * 10000 + DATEPART(MM, Event_Date)
* 100 + DATEPART(DD, Event_Date) FROM Aviation_Data ORDER BY 1 DESC);
WHILE (@D <= @END)
BEGIN
	INSERT INTO DIM_TIME VALUES
(
	@D,
	YEAR(@COUNTER),
	DATEPART(QQ, @COUNTER),
	MONTH(@COUNTER),
	(SELECT Nazwa FROM Miesiace WHERE Numer = MONTH(@COUNTER)),
	DAY(@COUNTER),
	(SELECT Nazwa FROM Dni WHERE Numer = DATEPART(DW, @COUNTER))
);
	SET @COUNTER = DATEADD(DAY, 1, @COUNTER);
	SET @D = CAST(CONVERT(varchar(8), @COUNTER, 112) AS INT);
END;
```

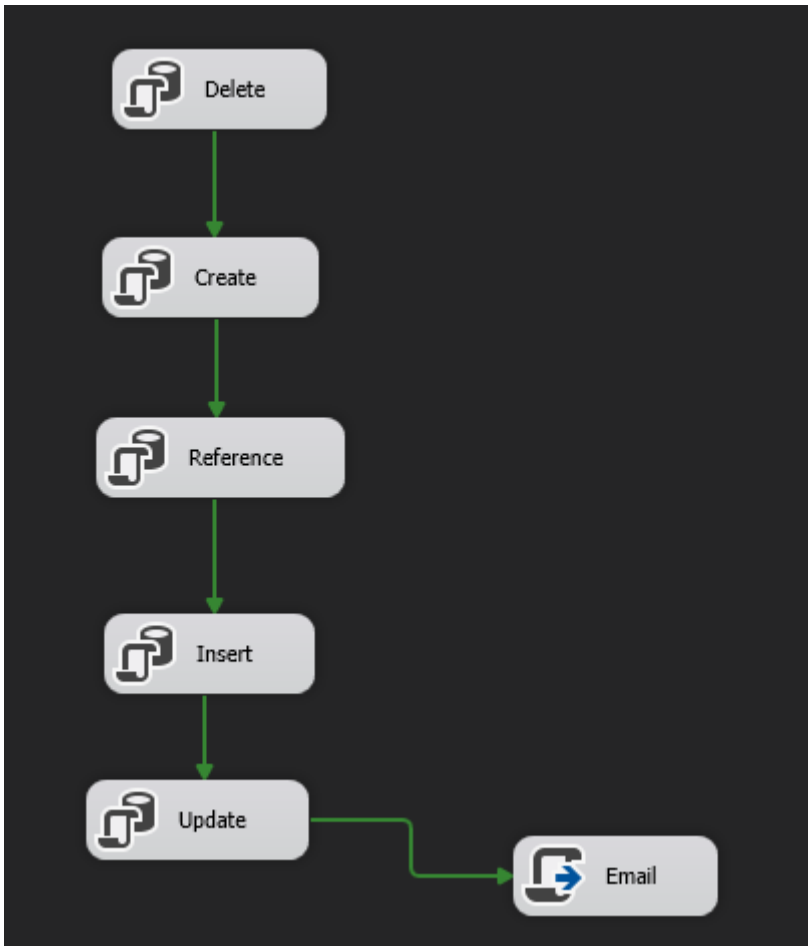
```
WITH FactAccidents (Accident_Id, Event_Date, Place_Id, Plane_Id,
Weather_Conditions_Id, Total_Fatal_Injuries, Total_Serious_Injuries,
Total_Minor_Injuries,
Total_Uninjured, Total_Injured, Mortality)
AS (
	SELECT DISTINCT DIM_ACCIDENT.Id, DIM_TIME.PK_TIME, DIM_PLACE.Id, DIM_PLANE.Id,
DIM_CONDITIONS.Id,
	CASE
		WHEN Aviation_Data.Total_Fatal_Injuries IS NULL THEN 0
		ELSE Aviation_Data.Total_Fatal_Injuries
	END Fatal,
	CASE
		WHEN Aviation_Data.Total_Serious_Injuries IS NULL THEN 0
		ELSE Aviation_Data.Total_Serious_Injuries
	END Serious,
	CASE
		WHEN Aviation_Data.Total_Minor_Injuries IS NULL THEN 0
		ELSE Aviation_Data.Total_Minor_Injuries
	END Minor,
	CASE
		WHEN Aviation_Data.Total_Uninjured IS NULL THEN 0
		ELSE Aviation_Data.Total_Uninjured
	END Uninjured,
	CASE WHEN Aviation_Data.Total_Fatal_Injuries IS NULL THEN 0
	ELSE Aviation_Data.Total_Fatal_Injuries END +
	CASE WHEN Aviation_Data.Total_Serious_Injuries IS NULL THEN 0
	ELSE Aviation_Data.Total_Serious_Injuries END +
	CASE WHEN Aviation_Data.Total_Minor_Injuries IS NULL THEN 0
	ELSE Aviation_Data.Total_Minor_Injuries END Injured,
	CASE
		WHEN Aviation_Data.Total_Fatal_Injuries IS NULL THEN 0
		WHEN (Aviation_Data.Total_Uninjured + Aviation_Data.Total_Fatal_Injuries
+ Aviation_Data.Total_Serious_Injuries + Aviation_Data.Total_Minor_Injuries) IS NULL
THEN 1
		WHEN (Aviation_Data.Total_Uninjured + Aviation_Data.Total_Fatal_Injuries
+ Aviation_Data.Total_Serious_Injuries + Aviation_Data.Total_Minor_Injuries) IS NULL
		AND Aviation_Data.Total_Fatal_Injuries IS NULL THEN 0
		WHEN (Aviation_Data.Total_Uninjured + Aviation_Data.Total_Fatal_Injuries
+ Aviation_Data.Total_Serious_Injuries + Aviation_Data.Total_Minor_Injuries)=0 THEN 1
		ELSE Aviation_Data.Total_Fatal_Injuries * 1.0
/((Aviation_Data.Total_Uninjured + Aviation_Data.Total_Serious_Injuries +
Aviation_Data.Total_Minor_Injuries + Aviation_Data.Total_Fatal_Injuries))
	END
	FROM Aviation_Data
	JOIN DIM_ACCIDENT ON CONCAT(Aviation_Data.Investigation_Type,
Aviation_Data.Injury_Severity, Aviation_Data.Aircraft_Damage,
Aviation_Data.FAR_Description, Aviation_Data.Schedule,
Aviation_Data.Purpose_Of_Flight, Aviation_Data.Air_Carrier,
Aviation_Data.Broad_Phase_Of_Flight) = CONCAT(DIM_ACCIDENT.Investigation_Type,
DIM_ACCIDENT.Injury_Severity, DIM_ACCIDENT.Aircraft_Damage,
DIM_ACCIDENT.FAR_Description, DIM_ACCIDENT.Schedule, DIM_ACCIDENT.Purpose_Of_Flight,
DIM_ACCIDENT.Air_Carrier, DIM_ACCIDENT.Broad_Phase_Of_Flight)
	JOIN DIM_TIME ON DIM_TIME.PK_TIME = DATEPART(YYYY, Event_Date) * 10000 +
DATEPART(MM, Event_Date) * 100 + DATEPART(DD, Event_Date)
	JOIN DIM_PLACE ON CONCAT(DIM_PLACE.Location, DIM_PLACE.Country,
DIM_PLACE.Airport_Code, DIM_PLACE.Airport_Name) = CONCAT(Aviation_Data.Location,
Aviation_Data.Country, Aviation_Data.Airport_Code, Aviation_Data.Airport_Name)
	JOIN DIM_PLANE ON CONCAT(DIM_PLANE.Make, DIM_PLANE.Model,
DIM_PLANE.Amateur_Built, CAST(DIM_PLANE.Number_Of_Engines AS nvarchar(2)),
DIM_PLANE.Engine_Type, DIM_PLANE.Aircraft_Category) = CONCAT(Aviation_Data.Make,
Aviation_Data.Model, Aviation_Data.Amateur_Built, CAST(Aviation_Data.Number_Of_Engines
AS nvarchar(2)), Aviation_Data.Engine_Type, Aviation_Data.Aircraft_Category)
```

```
      JOIN DIM_CONDITIONS ON DIM_CONDITIONS.Weather_Condition_Code =
Aviation_Data.Weather_Condition
) INSERT INTO FACT_ACCIDENTS (Accident_Id, Event_Date, Place_Id, Plane_Id,
Weather_Conditions_Id, Total_Fatal_Injuries, Total_Serious_Injuries,
Total_Minor_Injuries,
Total_Uninjured, Total_Injured, Mortality)
SELECT * FROM FactAccidents;
```

ETL:

Mapa:

Cel			Źródło			Przekształcenie
DIM_ACCIDENT	Id	int				Identity
DIM_ACCIDENT	Investigation_Type	nvarchar	Aviation_Data	Investigation_Type	nvarchar	---
DIM_ACCIDENT	Injury_Severity	nvarchar	Aviation_Data	Injury_Severity	nvarchar	---
DIM_ACCIDENT	Aircraft_Damage	nvarchar	Aviation_Data	Aircraft_Damage	nvarchar	---
DIM_ACCIDENT	FAR_Description	nvarchar	Aviation_Data	FAR_Description	nvarchar	---
DIM_ACCIDENT	Schedule	nvarchar	Aviation_Data	Schedule	nvarchar	---
DIM_ACCIDENT	Purpose_Of_Flight	nvarchar	Aviation_Data	Purpose_Of_Flight	nvarchar	---
DIM_ACCIDENT	Air_Carrier	nvarchar	Aviation_Data	Air_Carrier	nvarchar	---
DIM_ACCIDENT	Broad_Phase_Of_Flight	nvarchar	Aviation_Data	Broad_Phase_Of_Flight	nvarchar	---
DIM_CONDITIONS	Id	int				Identity
DIM_CONDITIONS	Weather_Condition	nvarchar	Aviation_Data		nvarchar	WHEN Weather_Conditions = 'VMC' THEN 'Good conditions' WHEN Weather_Conditions = 'UNK' OR Weather_Conditions='' T HEN 'Unknown conditions' WHEN Weather_Conditions = 'IMC' THEN 'Bad conditions'
DIM_CONDITIONS	Weather_Condition_Code	nvarchar	Aviation_Data	Weather_Condition_Code	nvarchar	---
DIM_PLACE	Id	int				Identity
DIM_PLACE	Location	nvarchar		Location	nvarchar	WHEN SUBSTRING([Location], LEN([Location]) - 3, 1)='-' THEN SUBSTRING([Location], LEN([Location]) - 2, 2)
DIM_PLACE	Country	nvarchar	Aviation_Data	Country	nvarchar	---
DIM_PLACE	Region	nvarchar	USState_Codes	Abbreviation	nvarchar	---
DIM_PLACE	Airport_Code	nvarchar	Aviation_Data	Airport_Code	nvarchar	---
DIM_PLACE	Airport_Name	nvarchar	Aviation_Data	Airport_Name	nvarchar	---
DIM_PLACE	Region_Code	nvarchar	Aviation_Data	Location	nvarchar	SUBSTRING([Location], LEN([Location]) - 1, 2)
DIM_PLANE	Id	int				Identity
DIM_PLANE	Make	nvarchar	Aviation_Data	Make	nvarchar	---
DIM_PLANE	Model	nvarchar	Aviation_Data	Model	nvarchar	---
DIM_PLANE	Amateur_Built	nvarchar	Aviation_Data	Amateur_Built	nvarchar	---
DIM_PLANE	Number_Of_Engines	int	Aviation_Data	Number_Of_Engines	int	---
DIM_PLANE	Engine_Type	nvarchar	Aviation_Data	Engine_Type	nvarchar	---
DIM_PLANE	Aircraft_Category	nvarchar	Aviation_Data	Aircraft_Category	nvarchar	---
DIM_TIME	PK_TIME	int	Aviation_Data	Event_Date	date	DATEPART(YYYY, Event_Date) * 10000 + DATEPART(MM, Event_Date) * 100 + DATEPART(DD, Event_Date) FROM Aviation_Data
DIM_TIME	Year	int	Aviation_Data	Event_Date	date	YEAR(Event_Date)
DIM_TIME	Quarter	int	Aviation_Data	Event_Date	date	DATEPART(QQ, Event_Date)
DIM_TIME	Month	int	Aviation_Data	Event_Date	date	MONTH(Event_Date)
DIM_TIME	Month_In_Words	nvarchar	Miesiace	Nazwa	date	---
DIM_TIME	Day	int	Aviation_Data	Event_Date	date	DAY(Event_Date)
DIM_TIME	Day_In_Words	nvarchar	Dni	Nazwa	date	---
FACT_ACCIDENTS	Accident_Id	int	DIM_ACCIDENT	Id	int	---
FACT_ACCIDENTS	Event_Date	int	DIM_TIME	Id	int	---
FACT_ACCIDENTS	Place_Id	int	DIM_PLACE	Id	int	---
FACT_ACCIDENTS	Plane_Id	int	DIM_PLANE	Id	int	---
FACT_ACCIDENTS	Weather_Conditions_Id	int	DIM_CONDITIONS	Id	int	---
FACT_ACCIDENTS	Total_Fatal_Injuries	int	Aviation_Data	Total_Fatal_Injuries	int	---
FACT_ACCIDENTS	Total_Serious_Injuries	int	Aviation_Data	Total_Serious_Injuries	int	---
FACT_ACCIDENTS	Total_Minor_Injuries	int	Aviation_Data	Total_Minor_Injuries	int	---
FACT_ACCIDENTS	Total_Uninjured	int	Aviation_Data	Total_Uninjured	int	---
FACT_ACCIDENTS	Total_Injured	int	Aviation_Data	Total_Fatal_Injuries, Total_Serious_Injuries, Total_Minor_Injuries	int	Aviation_Data.Total_Serious_Injuries + Aviation_Data.Total_Minor_Injuries + Aviation_Data.Total_Fatal_Injuries
FACT_ACCIDENTS	Mortality	decimal	Aviation_Data	Total_Uninjured, Total_Injured	int	Aviation_Data.Total_Fatal_Injuries * 1.0 /((Aviation_Data.Total_Uninjured + Aviation_Data.Total_Serious_Injuries + Aviation_Data.Total_Minor_Injuries + Aviation_Data.Total_Fatal_Injuries)



Wnioski:

Zadanie Delete sprawdza, czy dane tabele istnieją – jeżeli tak, usuwa je. Create tworzy schematy tabel wymiarów oraz tabel pomocniczych, natomiast reference dodaje klucze główne do tabeli wymiarów oraz klucze obce do tabeli faktów: klucz główny z tabel DIM_ACCIDENT, DIM_CONDITIONS, DIM_TIME, DIM_PLACE, DIM_PLANE.

Najbardziej złożoną instrukcją jest Insert:

- Dodaje do tabeli DIM_ACCIDENT ręcznie utworzone Id, Accident_Number, Investigation_Type, Injury_Severity, Aircraft_Damage, FAR_Description, Schedule, Purpose_Of_Flight, Air_Carrier oraz Broad_Phase_Of_Flight bezpośrednio z tabeli bazowej Aviation_Data
- Dodaje do tabeli DIM_CONDITIONS ręcznie utworzone Id, ręcznie rozwinięty skrót kodu warunków, oraz kod warunków pogodowych z tabeli Aviation_Data
- Do tabeli DIM_PLACE dodaje pola z tabeli Aviation_Data oraz tabeli USState_Codes
- Do tabeli DIM_PLANE dodaje pola bezpośrednio z tabeli Aviation_Data
- Do tabeli DIM_TIME dodaje wszystkie daty od pierwszej daty zdarzenia do ostatniej daty opublikowania wypadku
- Do tabeli FACT_ACCIDENTS dodaje numery Id z tabel wymiarów (przy użyciu ciekawych instrukcji złączenia ze względu na brak domyślnych Id dla wymiarów) oraz dokonuje walidacji miar – podmienia wartości NULL na 0.