**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;

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

)

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);

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) |

Obraz zawierający tekst, sprzęt elektroniczny

Opis wygenerowany automatycznie

**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.