Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з комп’ютерного практикуму № 1 з дисципліни

«Аналіз даних в інформаційних системах»

«Створення сховища даних»

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Олійник Юрій Олександрович

Київ 2023

**Комп’ютерний практикум 1**

**Створення сховища даних**

**Постановка завдання.**

Навчитися створювати процедури завантаження даних до сховища.

1. Самостійно обрати не менше 3-х джерел відкритих даних;
2. Спроєктувати модель Stage зони для ETL процесів;
3. Спроєктувати модель основного сховища за типом «зірка» або «сніжинка»;
4. Створити ETL засоби:
   1. завантажити дані до Stage зони;
   2. створити набір процедур/функцій для перетворення та завантаження даних до основного сховища (або створити засобами програмних ETL засобів). Передбачити можливість завантаження змінених та додаткових даних;
5. Завантажити дані до основного сховища даних.

**Хід роботи.**

1. Для створення сховища даних я обрав джерела даних аеропортів, авіакомпаній та інформації про польоти у 2015 році в США. Джерела даних були обрані з ресурсу Kaggle: <https://www.kaggle.com/datasets/usdot/flight-delays> ;

Набори даних представлені у вигляді csv-файлів:

* airlines.csv – дані про авіалінії;
* airports.csv – дані про аеропорти;
* flights.csv – дані про польоти;

1. Предметною областю лабораторної роботи є авіарейси. В результаті розробки була спроєктована схема stage-зони, яка зображена на рисунку 1. Дана модель відображає таблиці для даних із вхідних джерел.

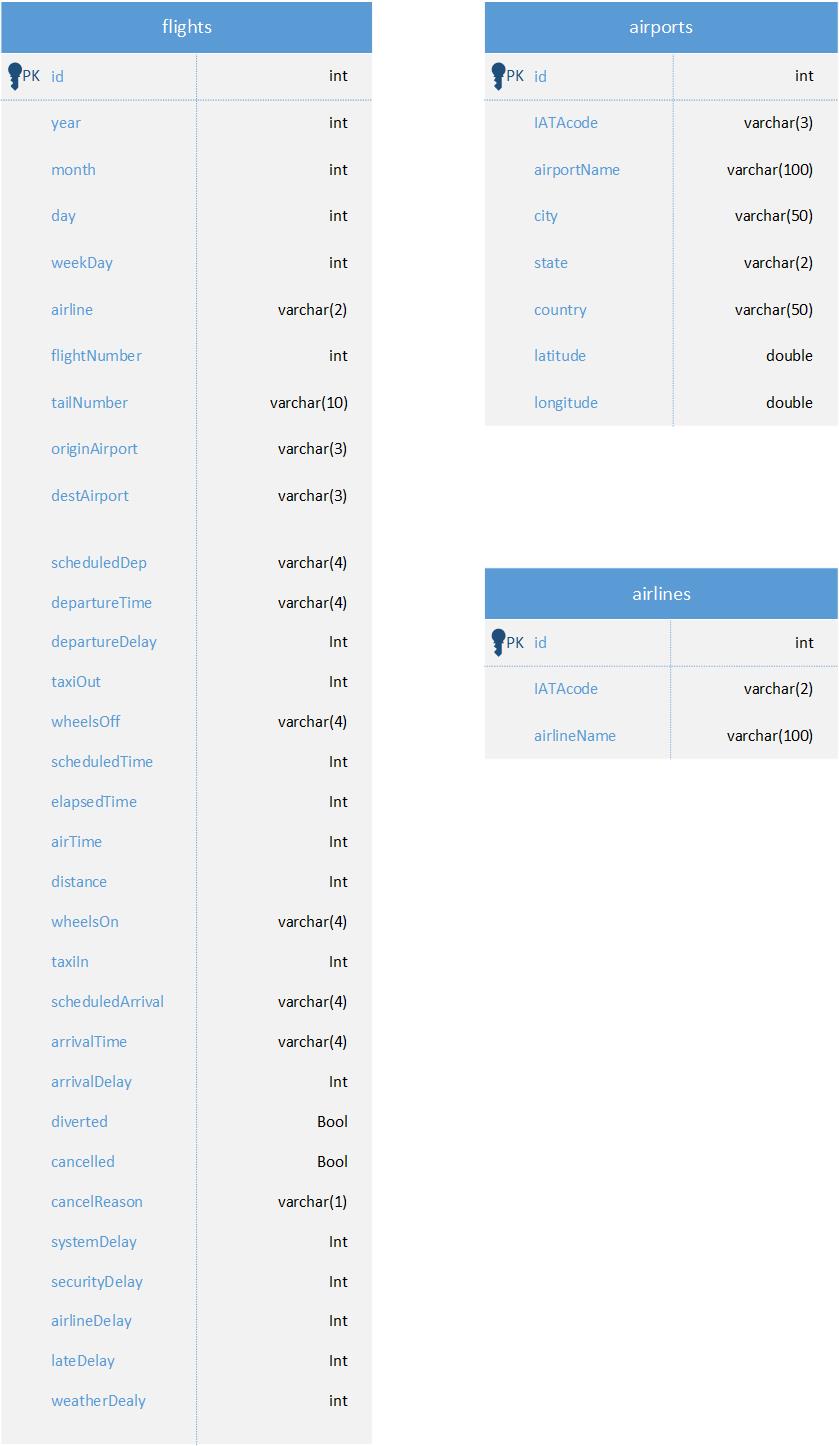


Рисунок 1 – Stage-зона

1. У розробленій моделі сховища даних присутні наступні таблиці:

* dim\_dates – дати польотів, вимір;
* dim\_aircraft – унікальні номери повітряних суден, вимір;
* dim\_cancelreason – причини скасування польотів, вимір;
* dim\_airlines – назви авіаліній, вимір;
* dim\_countries – назви країн, вимір;
* dim\_states – назви регіонів у країнах, вимір;
* dim\_cities – назви міст у регіонах, вимір;
* dim\_airports – назви аеропортів, їх місцезнаходження, вимір;
* fact\_flights – характеристики унікальних польотів, факти.

В результаті була спроєктована схема сховища даних, яка зображена на рисунку 2. Для представлення даних була вибрана модель за типом “зірка”.

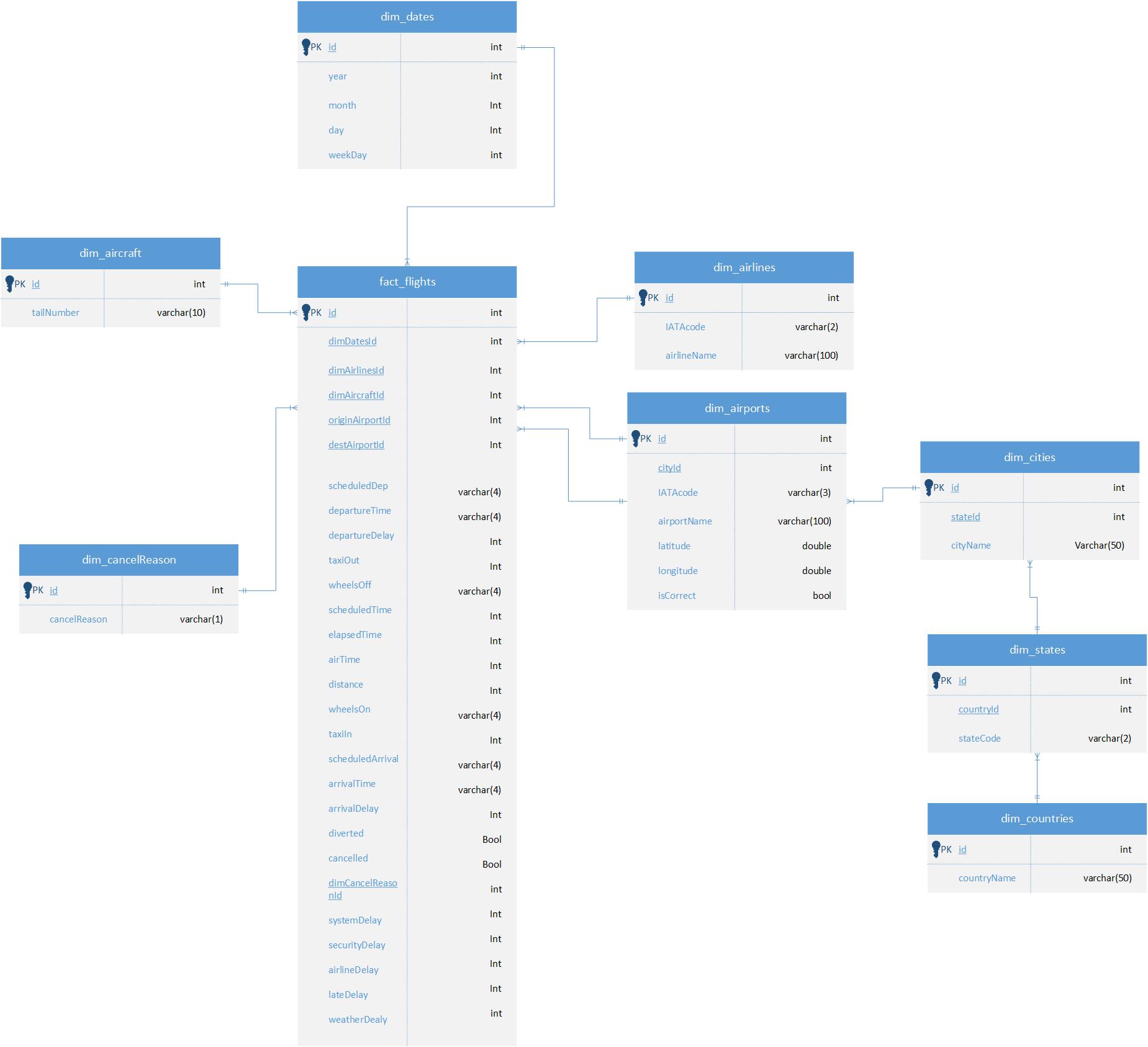


Рисунок 2 – модель основного сховища

1. Для створення сховища даних і реалізації ETL процесів було обрано скрипти SQL та СУБД PostgreSQL.
   1. Створення Stage зони відповідно до моделі і завантаження даних до Stage зони з джерел даних.

create schema stageZone;  
  
create table stagezone.airlines (  
id serial primary key,  
IATAcode *varchar*(2) unique not null,  
airlineName *varchar*(100) not null);  
  
create table stagezone.airports(  
id serial primary key,  
IATAcode *varchar*(3) unique not null,  
airportName *varchar*(100) not null,  
city *varchar*(50) not null,  
state *varchar*(2) not null,  
country *varchar*(50) not null,  
latitude *real*,  
longitude *real*);  
  
create table stagezone.flights(  
id serial primary key,  
year *integer* not null,  
month *integer* not null,  
day *integer* not null,  
weekday *integer*,  
airline *varchar*(2) not null,  
flightNumber *integer* not null,  
tailNumber *varchar*(10),  
originAirport *varchar*(3) not null,  
destAirport *varchar*(3) not null,  
  
scheduledDep *varchar*(4),  
departureTime *varchar*(4),  
 departureDelay *integer*,  
 taxiOut *integer*,  
 wheelsOff *varchar*(4),  
 scheduledTime *integer*,  
 elapsedTime *integer*,  
 airTime *integer*,  
 distance *integer* not null,  
 wheelsOn *varchar*(4),  
 taxiIn *integer*,  
 scheduledArrival *varchar*(4),  
 arrivalTime *varchar*(4),  
 arrivalDelay *integer*,  
 diverted boolean not null,  
 cancelled boolean not null,  
 cancelReason *varchar*(1),  
 systemDelay *integer*,  
 securityDelay *integer*,  
 airlineDelay *integer*,  
 lateDelay *integer*,  
 weatherDelay *integer*);  
   
copy stagezone.airlines(iatacode,airlinename)  
from 'C:\csvFiles\airlines.csv'  
delimiter ','  
csv header;  
  
copy stagezone.airports(iatacode,airportname,city,state,country,latitude,longitude)  
from 'C:\csvFiles\airports.csv'  
delimiter ','  
csv header;  
  
copy stagezone.flights(year,month,day,weekday,airline,flightnumber,tailnumber,originairport,  
 destairport,scheduleddep,departuretime,departuredelay,  
 taxiout,wheelsoff,scheduledtime,elapsedtime,airtime,distance,  
 wheelson,taxiin,scheduledarrival,arrivaltime,arrivaldelay,  
 diverted,cancelled,cancelreason,systemdelay,securitydelay,  
 airlinedelay,latedelay,weatherdelay)  
from 'C:\csvFiles\flights1.csv'  
delimiter ','  
csv header;

* 1. Створення основного сховища даних відповідно до моделі.

Таблицю warehouse.dim\_airports я створюю з підтримкою SCD. Маємо стовпець isCorrect для визначення актуальної назви аеропорту.

create schema warehouse;  
  
create table warehouse.dim\_Countries(  
id serial primary key,  
countryName varchar(50) unique not null);  
  
create table warehouse.dim\_States(  
id serial primary key,  
country\_id integer,  
stateCode varchar(2) unique,  
foreign key (country\_id) references warehouse.dim\_Countries(id));  
  
create table warehouse.dim\_Cities(  
id serial primary key,  
state\_id integer,  
cityName varchar(50) not null,  
foreign key (state\_id) references warehouse.dim\_States(id));  
  
create table warehouse.dim\_Airports(  
id serial primary key,  
city\_id integer,  
IATAcode varchar(3) unique not null,  
airportName varchar(100) unique not null,  
latitude real,  
longitude real,

isCorrect bool,  
foreign key (city\_id) references warehouse.dim\_Cities(id));  
  
create table warehouse.dim\_Airlines(  
id serial primary key,  
IATAcode varchar(2) unique not null,  
airlineName varchar(100) unique not null);  
  
create table warehouse.dim\_Dates(  
id serial primary key,  
year integer,  
month integer,  
day integer,  
weekday integer);  
  
create table warehouse.dim\_Aircraft(  
id serial primary key,  
tailnumber varchar(10) unique not null);  
  
create table warehouse.dim\_CancelReason(  
id serial primary key,  
cancelReason varchar(1) unique);  
  
create table warehouse.fact\_Flights(  
id serial primary key,  
dimDatesId integer,  
dimAirlinesId integer,  
dimAircraftId integer,  
originAirportId integer,  
destAirportId integer,  
dimCancelReasonId integer,  
  
scheduledDep varchar(4),  
departureTime varchar(4),  
 departureDelay integer,  
 taxiOut integer,  
 wheelsOff varchar(4),  
 scheduledTime integer,  
 elapsedTime integer,  
 airTime integer,  
 distance integer not null,  
 wheelsOn varchar(4),  
 taxiIn integer,  
 scheduledArrival varchar(4),  
 arrivalTime varchar(4),  
 arrivalDelay integer,  
 diverted boolean not null,  
 cancelled boolean not null,   
 systemDelay integer,  
 securityDelay integer,  
 airlineDelay integer,  
 lateDelay integer,  
 weatherDelay integer,  
  
 foreign key (dimDatesId) references warehouse.dim\_dates(id),  
 foreign key (dimAirlinesId) references warehouse.dim\_airlines(id),  
 foreign key (dimAircraftId) references warehouse.dim\_aircraft(id),  
 foreign key (originAirportId) references warehouse.dim\_airports(id),  
 foreign key (destAirportId) references warehouse.dim\_airports(id),  
 foreign key (dimCancelReasonId) references warehouse.dim\_cancelreason(id)  
);

* 1. Процедура для обробки таблиці airlines stage-зони і відповідного внесення даних таблиці до сховища даних.

CREATE OR REPLACE PROCEDURE stagezone.processAirlines()  
language plpgsql  
as $$  
  
declare   
 rec record;  
 lineCursor cursor for select \* from stagezone.airlines;  
   
begin  
 open lineCursor;  
 loop  
 fetch lineCursor into rec;  
 exit when not found;  
 if rec.iatacode is not null and rec.airlineName is not null   
 and rec.iatacode not in (SELECT iatacode FROM warehouse.dim\_airlines) then  
 insert into warehouse.dim\_airlines(iatacode,airlineName) values  
 (rec.iatacode,rec.airlineName);  
 end if;  
 end loop;  
 close lineCursor;  
end;$$;

* 1. Процедура для обробки таблиці airports stage-зони. Для таблиці warehouse.dim\_airports і коректної роботи SCD я перевіряю, чи вже є запис з кодом аеропортом, але іншою назвою, аніж пропонована в даному записі. Якщо попередня назва існує, то для того запису ставимо прапорець isCorrect:=False, і відповідно додаємо запис з актуальною назвою аеропорту і прапорцем isCorrect=True.

CREATE OR REPLACE PROCEDURE stagezone.processAirlines()  
language plpgsql  
as $$  
  
declare   
 rec record;  
 lineCursor cursor for select \* from stagezone.airlines;  
   
begin  
 open lineCursor;  
 loop  
 fetch lineCursor into rec;  
 exit when not found;  
 if rec.iatacode is not null and rec.airlineName is not null   
 and rec.iatacode not in (SELECT iatacode FROM warehouse.dim\_airlines) then  
 insert into warehouse.dim\_airlines(iatacode,airlineName) values  
 (rec.iatacode,rec.airlineName);  
 end if;  
 end loop;  
 close lineCursor;  
end;$$;  
  
CREATE OR REPLACE PROCEDURE stagezone.processAirports()  
language plpgsql  
as $$  
  
declare   
 rec record;  
 lineCursor cursor for select \* from stagezone.airports;  
 forID integer;  
 isCorrectVal bool;  
   
begin  
 open lineCursor;  
 loop  
 fetch lineCursor into rec;  
 exit when not found;  
   
   
 if rec.country is not null   
 and rec.country not in (SELECT countryname FROM warehouse.dim\_countries) then  
 insert into warehouse.dim\_countries(countryname) values  
 (rec.country);  
 end if;  
   
 if rec.state not in (SELECT statecode FROM warehouse.dim\_states) then  
 forid:=(SELECT id FROM warehouse.dim\_countries  
 WHERE warehouse.dim\_countries.countryname=rec.country);  
 insert into warehouse.dim\_states(country\_id,statecode) values  
 (forid, rec.state);  
 end if;  
   
   
   
 if rec.city is not null  
 and rec.city not in (SELECT cityname FROM warehouse.dim\_cities WHERE state\_id=(  
 select id from warehouse.dim\_states where statecode=rec.state)) then  
 forid:=(SELECT warehouse.dim\_states.id FROM warehouse.dim\_states, warehouse.dim\_countries  
 WHERE warehouse.dim\_states.statecode=rec.state  
 and warehouse.dim\_states.country\_id=(  
 SELECT warehouse.dim\_countries.id  
 WHERE warehouse.dim\_countries.countryname=rec.country));  
 insert into warehouse.dim\_cities(state\_id,cityname) values  
 (forid, rec.city);  
 end if;  
   
 if rec.iatacode is not null   
 and (rec.iatacode not in (SELECT iatacode FROM warehouse.dim\_airports) or  
 rec.airportName not in (SELECT airportName FROM warehouse.dim\_airports WHERE iatacode=rec.iatacode))  
 then  
 forid:=(SELECT warehouse.dim\_cities.id FROM warehouse.dim\_cities, warehouse.dim\_states  
 WHERE warehouse.dim\_cities.cityname=rec.city  
 and warehouse.dim\_cities.state\_id=(  
 SELECT warehouse.dim\_states.id  
 WHERE warehouse.dim\_states.statecode=rec.state));  
 if rec.iatacode in (SELECT iatacode FROM warehouse.dim\_airports)  
 then  
 update warehouse.dim\_airports  
 set isCorrect=False  
 where iatacode=rec.iatacode and isCorrect=True;  
 end if;  
 insert into warehouse.dim\_airports(city\_id,iatacode,airportName,  
 latitude,longitude,isCorrect) values  
 (forid,rec.iatacode,rec.airportName,rec.latitude,rec.longitude,True);  
 end if;  
 end loop;  
 close lineCursor;  
end;$$

* 1. Процедура для обробки таблиці flights stage зони і остаточного формування таблиць вимірів.

CREATE OR REPLACE PROCEDURE stagezone.processFlights()  
language plpgsql  
as $$  
  
declare   
 rec record;  
 lineCursor cursor for select \* from stagezone.flights;  
 forID integer;  
   
begin  
 open lineCursor;  
 loop  
 fetch lineCursor into rec;  
 exit when not found;   
   
 if rec.day not in (SELECT day FROM (  
 SELECT \* FROM (SELECT \* FROM warehouse.dim\_dates WHERE year=rec.year) as y WHERE month=rec.month) as mo) then  
 insert into warehouse.dim\_dates(year,month,day,weekday) values  
 (rec.year,rec.month,rec.day,rec.weekday);  
 end if;  
   
 if rec.tailnumber is not null and  
 rec.tailnumber not in (select tailnumber from warehouse.dim\_aircraft) then  
 insert into warehouse.dim\_aircraft(tailnumber) values  
 (rec.tailnumber);  
 end if;  
   
 if rec.cancelreason is not null and   
 rec.cancelreason not in (select cancelreason from warehouse.dim\_cancelreason) then  
 insert into warehouse.dim\_cancelreason(cancelreason) values  
 (rec.cancelreason);  
 end if;  
   
   
 end loop;  
 close lineCursor;  
end;$$

* 1. Процедура обробки таблиці flights stage-зони, пошук зв’язків з таблицями вимірів і формування таблиці фактів. Для таблиці warehouse.dim\_airports перевіряємо, щоб запис аеропорту для зовнішнього ключа містив актуальну назву, тобто isCorrect=True.

1. CREATE OR REPLACE PROCEDURE stagezone.processfacts()  
   language plpgsql  
   as $$  
     
   declare   
    rec record;  
    lineCursor cursor for select \* from stagezone.flights;  
    dateID integer;  
    airlineId integer;  
    aircraftId integer;  
    origAir integer;  
    destAir integer;  
    reasonID integer;  
      
   begin  
    open lineCursor;  
    loop  
    fetch lineCursor into rec;  
    exit when not found;  
      
      
    SELECT id from warehouse.dim\_dates where year=rec.year and month=rec.month and day=rec.day  
    into dateid;  
      
    SELECT id from warehouse.dim\_airlines where iatacode=rec.airline  
    into airlineid;  
      
    SELECT id from warehouse.dim\_aircraft where tailnumber=rec.tailnumber  
    into aircraftid;  
      
    SELECT id from warehouse.dim\_airports where iatacode=rec.originairport and isCorrect=True  
    into origair;  
      
    SELECT id from warehouse.dim\_airports where iatacode=rec.destairport  
    into destair;  
      
    SELECT id from warehouse.dim\_cancelreason where cancelreason=rec.cancelreason  
    into reasonId;  
      
    insert into warehouse.fact\_flights(dimDatesId,dimAirlinesId,dimAircraftId,  
    originAirportId,destAirportId,dimCancelReasonId,  
    scheduledDep,departureTime,departureDelay,taxiOut,  
    wheelsOff,scheduledTime,elapsedTime,airTime,distance,  
    wheelsOn,taxiIn,scheduledArrival,arrivalTime,  
    arrivalDelay,diverted,cancelled,systemDelay,  
    securityDelay,airlineDelay,lateDelay,weatherDelay)  
    values (dateID,airlineId,aircraftId,origAir,destAir,reasonID,  
    rec.scheduledDep,rec.departureTime,rec.departureDelay,  
    rec.taxiOut,rec.wheelsOff,rec.scheduledTime,rec.elapsedTime,  
    rec.airTime,rec.distance,rec.wheelsOn,rec.taxiIn,rec.scheduledArrival,  
    rec.arrivalTime,rec.arrivalDelay,rec.diverted,rec.cancelled,rec.systemDelay,  
    rec.securityDelay,rec.airlineDelay,rec.lateDelay,rec.weatherDelay);  
    end loop;  
    close lineCursor;  
   end;$$
2. Результати виконання скриптів:
   1. Створена Stage-зона:

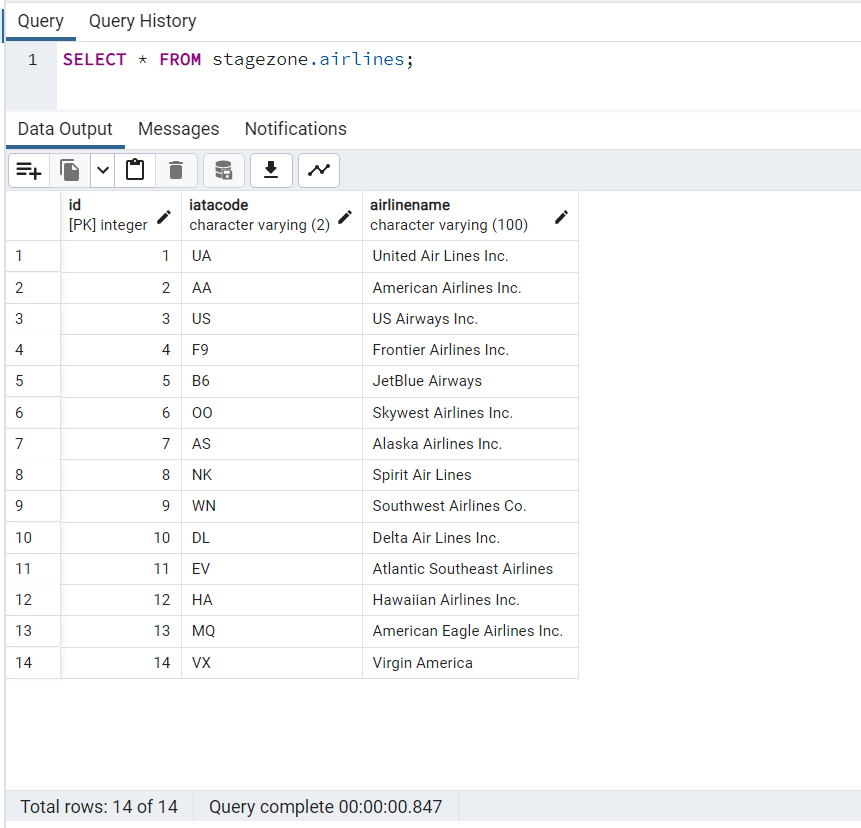


Рисунок 3 – таблиця airlines stage зони.

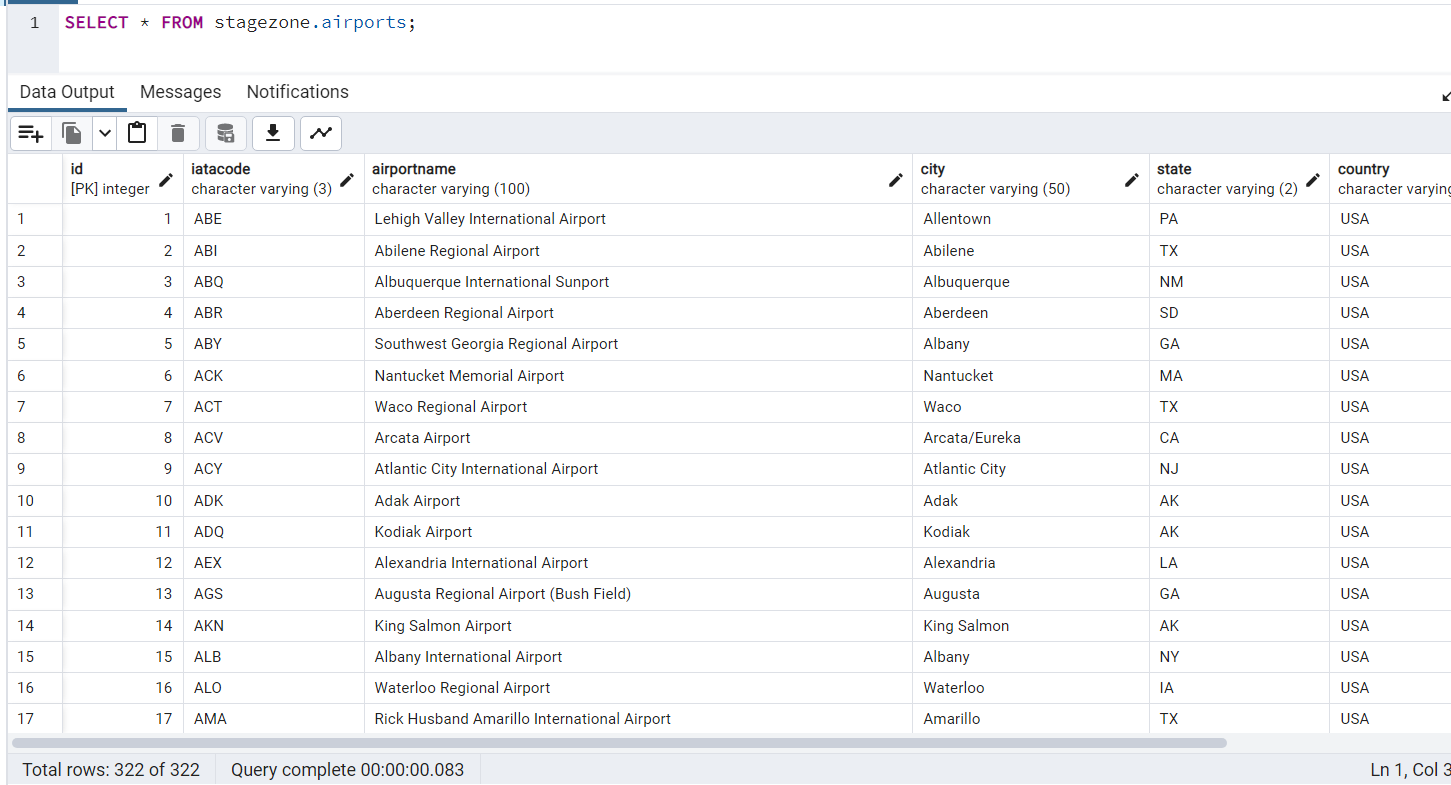


Рисунок 4 – таблиця airports stage зони.

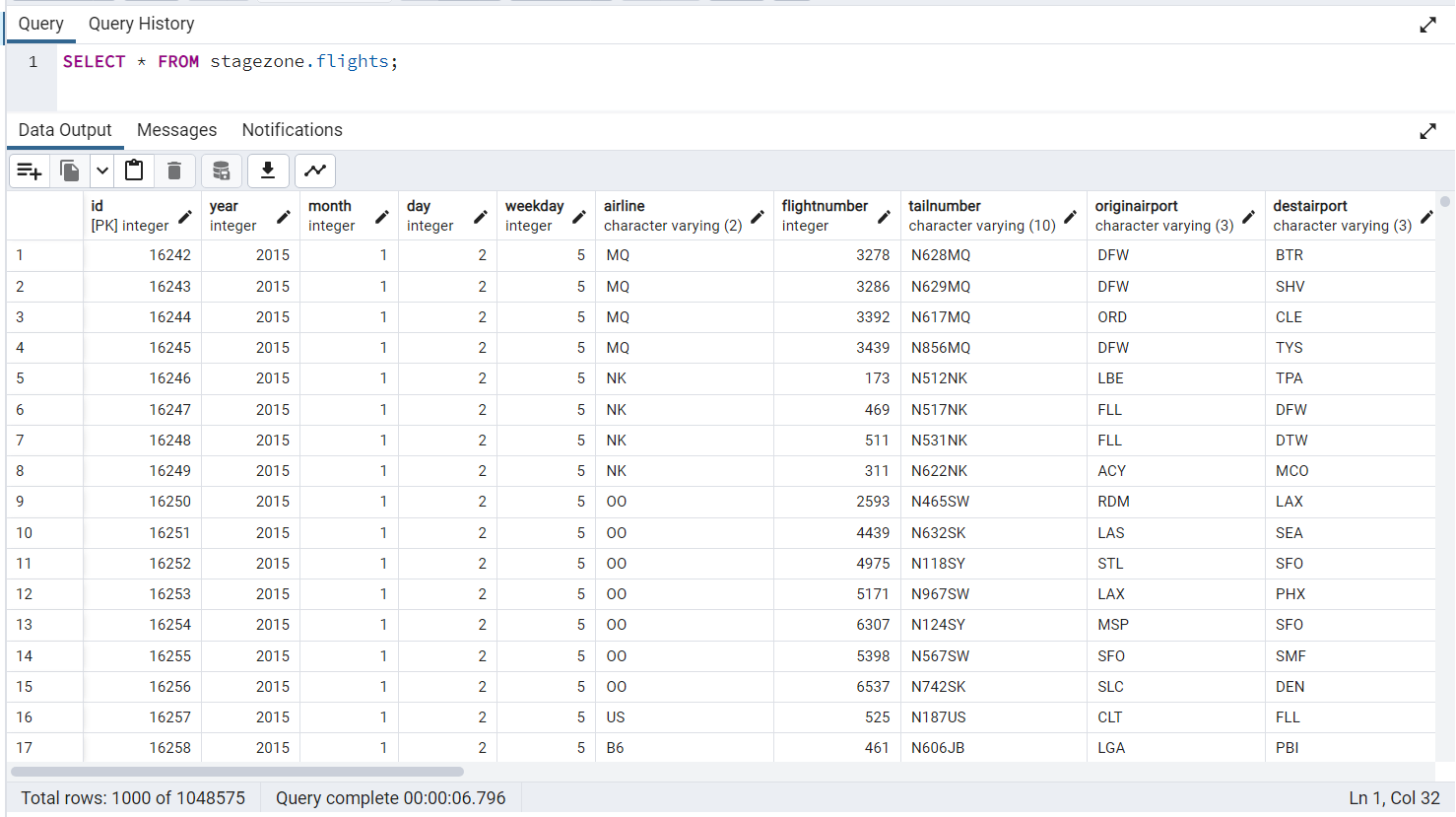


Рисунок 5 – таблиця flights stage зони.

* 1. Сховище даних після виконання завантаження даних до stage зони та виконання усіх процедур щодо перетворення й завантаження даних до основного сховища.

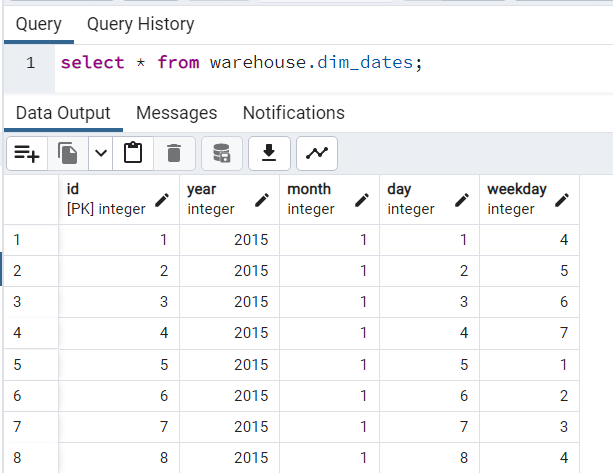


Рисунок 6 – таблиця dim\_dates основного сховища.

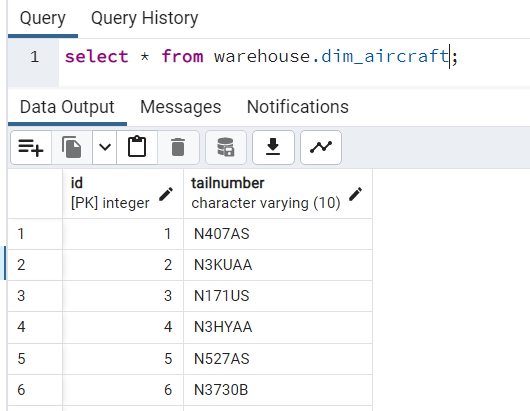


Рисунок 7 – таблиця dim\_aircraft основного сховища.

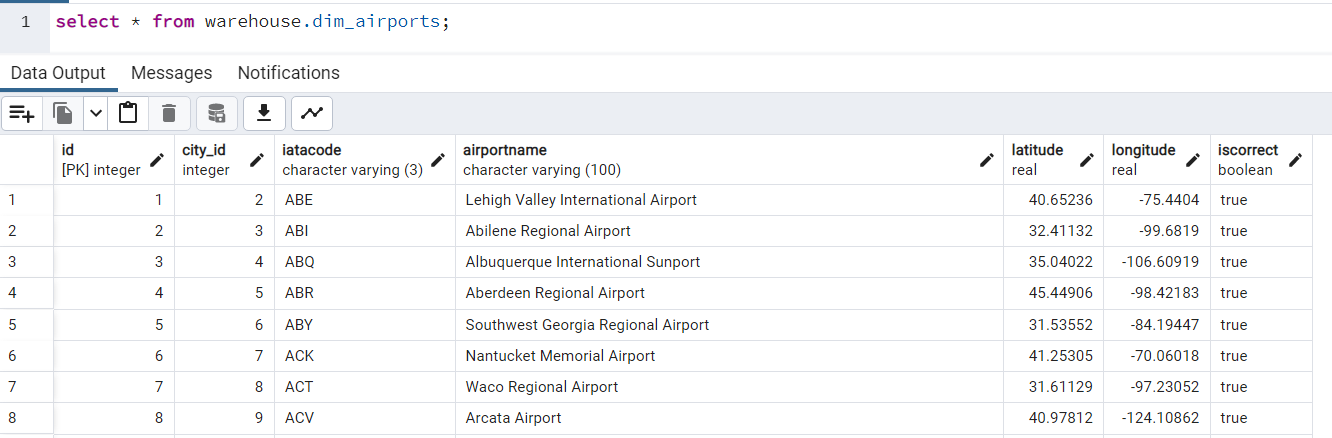


Рисунок 7 – таблиця dim\_airports основного сховища.

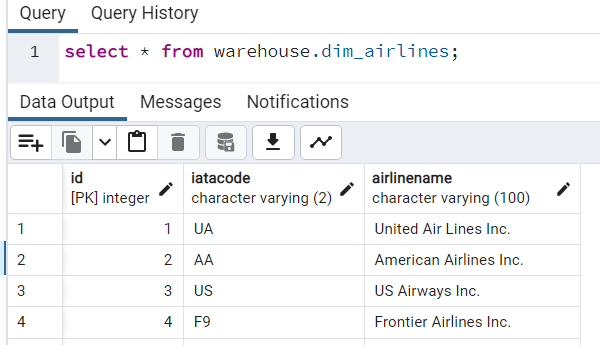


Рисунок 8 – таблиця dim\_airlines основного сховища.

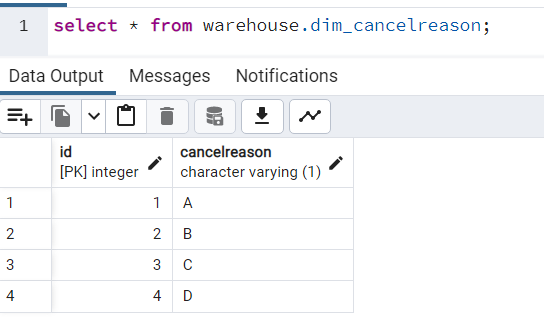


Рисунок 9 – таблиця dim\_cancelreason основного сховища.

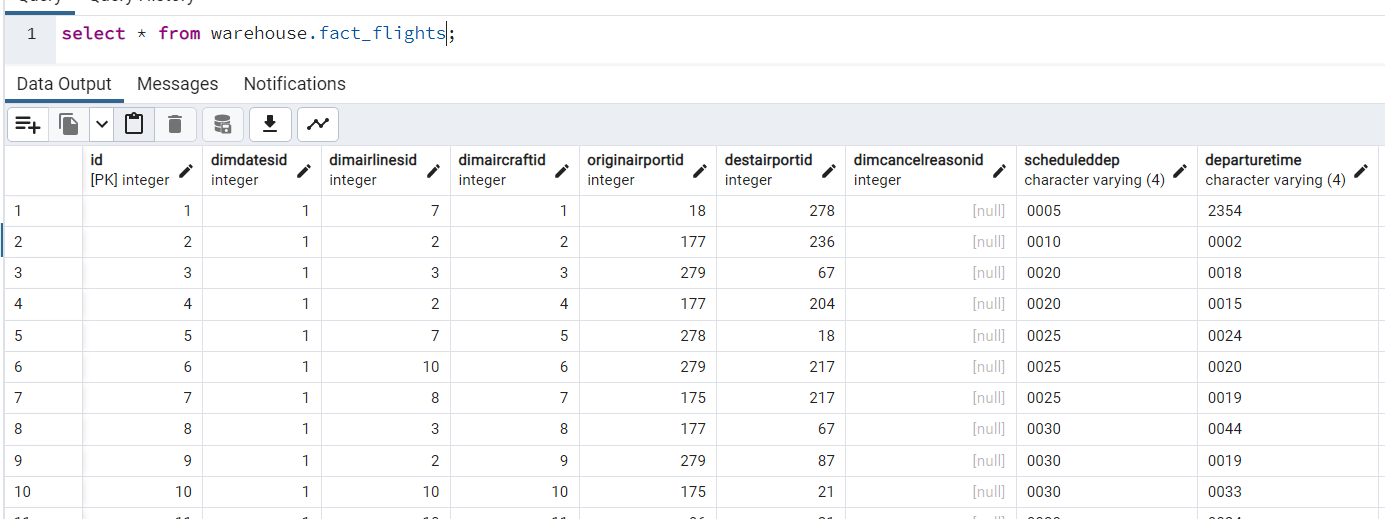


Рисунок 10 – таблиця fact\_flights основного сховища.

**Реалізація SCD.**

Таблицю warehouse.dim\_airports я створюю з підтримкою SCD. Маємо стовпець isCorrect для визначення, чи актуальна назва аеропорту. Якщо так, то isCorrect=True.

Під час заповнення даної таблиці я перевіряю, чи вже є запис з кодом поточного аеропорту, але іншою назвою, аніж пропонована в даному записі. Якщо попередня назва існує і не збігається з поточною, це означає, що назва змінилася, тож для того запису ставимо прапорець isCorrect:=False, і відповідно додаємо запис з актуальною назвою аеропорту і прапорцем isCorrect=True. Таким чином у таблиці зберігаються усі попередні назви аеропортів, а для визначення поточного використовується логічна змінна.

Під час додавання рейсів у фактову таблицю, задля коректності запису даних перевіряємо, щоб запис аеропорту для зовнішнього ключа містив актуальну назву, тобто isCorrect=True. Таким чином повністю реалізується SCD для таблиці аеропортів.

**Висновок.**

Отже, у цій роботі я ознайомився з підходами до створення сховищ даних та навчився створювати процедури завантаження даних до сховища. У результаті лабораторної роботи було обрано джерела даних, розроблено stage zone, сховище даних та ETL засоби для обробки даних. Використовуючи СУБД PostgreSQL, SQL-скрипти та MS Visio для побудови моделей, отримуємо коректний результат.