



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Лабораторна робота №1

Аналіз даних в інформаційних системах

Виконав
студент групи ІТ-03:

Чабан А.Є.

Перевірив:

Олійник Ю.О

Київ 2021

Завдання:

Приклад виконання завдання.

1) Самостійно обрати не менше 3-х джерел відкритих даних.

Наприклад:

- Реєстр адміністративно-територіального устрою

<https://data.gov.ua/dataset/a2d6c060-e7e6-4471-ac67-42cfa1742a19>

- Відомості про транспортні засоби та їх власників

[https://data.gov.ua/dataset/06779371-308f-42d7-895e-](https://data.gov.ua/dataset/06779371-308f-42d7-895e-5a39833375f0/resource/7a58e8f7-9323-47d4-a21d-19486e014eb4)

[5a39833375f0/resource/7a58e8f7-9323-47d4-a21d-19486e014eb4](https://data.gov.ua/dataset/06779371-308f-42d7-895e-5a39833375f0/resource/7a58e8f7-9323-47d4-a21d-19486e014eb4)

- План спроможної мережі надання первинної медичної

допомоги [https://data.gov.ua/dataset/2a2502f0-19f7-4552-9b94-](https://data.gov.ua/dataset/2a2502f0-19f7-4552-9b94-3b0db5399d8a/resource/99cf4cdd-dd45-4d8b-a706-65188b47efaf)

[3b0db5399d8a/resource/99cf4cdd-dd45-4d8b-a706-65188b47efaf](https://data.gov.ua/dataset/2a2502f0-19f7-4552-9b94-3b0db5399d8a/resource/99cf4cdd-dd45-4d8b-a706-65188b47efaf)

2) Спроекувати модель Stage зони для ETL процесів.

3) Спроекувати модель основного сховища за типом «зірка» [11, ст172]

або «сніжинка» . [11, ст189]

4) Створити ETL засоби:

- завантажити дані до Stage зони

- 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.

Виконання завдання:

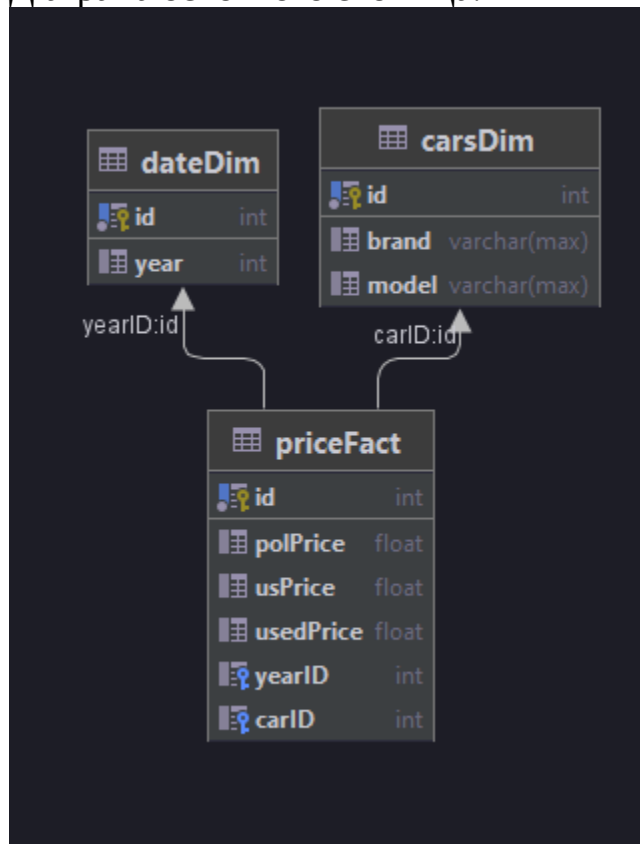
Діаграма Stage зони:

Car_prices	
ID	int
Price	int
Levy	varchar(max)
Manufacturer	varchar(max)
Model	varchar(max)
Prod. year	int
Category	varchar(max)
Leather interior	varchar(max)
Fuel type	varchar(max)
Engine volume	varchar(max)
Mileage	varchar(max)
Cylinders	float
Gear box type	varchar(max)
Drive wheels	varchar(max)
Doors	varchar(max)
Wheel	varchar(max)
Color	varchar(max)
Airbags	int

USA_cars_datasets	
ID	int
price	int
brand	varchar(max)
model	varchar(max)
year	int
title_status	varchar(max)
mileage	float
color	varchar(max)
vin	varchar(max)
lot	int
state	varchar(max)
country	varchar(max)
condition	varchar(max)

Car_Prices_Poland_Kaggle	
id	int
mark	varchar(max)
model	varchar(max)
generation_name	varchar(max)
year	int
mileage	int
vol_engine	int
fuel	varchar(max)
city	varchar(max)
province	varchar(max)
price	int

Діаграма основного сховища:



Використані скрипти:

```
create database adisDB

use adisDB

create schema stageSc
create schema modelSc

insert into modelSc.dateDim(year) (
    select distinct [Prod. year]
    from stageSc.Car_prices
    union
    select distinct USA_cars_datasets.year
    from stageSc.USA_cars_datasets
    union
    select distinct Car_Prices_Poland_Kaggle.year
    from stageSc.Car_Prices_Poland_Kaggle
)

UPDATE stageSc.Car_prices
SET Manufacturer = REPLACE(lower(Manufacturer), ' ', '')
UPDATE stageSc.Car_prices
SET Manufacturer = REPLACE(lower(Manufacturer), '-', '')
UPDATE stageSc.Car_prices
```

```

SET Model = REPLACE(lower(Model), ' ', '')
UPDATE stageSc.Car_prices
SET Model = REPLACE(lower(Model), '-', '')

UPDATE stageSc.USA_cars_datasets
SET brand = REPLACE(lower(brand), ' ', '')
UPDATE stageSc.USA_cars_datasets
SET brand = REPLACE(lower(brand), '-', '')
UPDATE stageSc.USA_cars_datasets
SET Model = REPLACE(lower(Model), ' ', '')
UPDATE stageSc.USA_cars_datasets
SET Model = REPLACE(lower(Model), '-', '')

UPDATE modelSc.carsDim
SET brand = REPLACE(lower(brand), ' ', '')
UPDATE modelSc.carsDim
SET brand = REPLACE(lower(brand), '-', '')
UPDATE modelSc.carsDim
SET model = REPLACE(lower(model), ' ', '')
UPDATE modelSc.carsDim
SET model = REPLACE(lower(model), '-', '')

--

insert into modelSc.carsDim(brand, model) (
    select distinct REPLACE(lower(Manufacturer), ' ', ''),
    REPLACE(lower(Model), ' ', '')
    from stageSc.Car_prices
    union
    select distinct REPLACE(lower(brand), ' ', ''), REPLACE(lower(Model), ' ', '')
    from stageSc.USA_cars_datasets
    union
    select distinct REPLACE(lower(mark), ' ', ''), REPLACE(lower(Model), ' ', '')
    from stageSc.Car_Prices_Poland_Kaggle
)

UPDATE stageSc.Car_Prices_Poland_Kaggle
SET mark = REPLACE(lower(mark), ' ', '')
UPDATE stageSc.Car_Prices_Poland_Kaggle
SET mark = REPLACE(lower(mark), '-', '')
UPDATE stageSc.Car_Prices_Poland_Kaggle
SET Model = REPLACE(lower(Model), ' ', '')
UPDATE stageSc.Car_Prices_Poland_Kaggle
SET Model = REPLACE(lower(Model), '-', '')

create function zlotyToUsd(@Price float)
    returns float
as
begin
    return @Price * 0.25;
end

--

with tmpPrices as (
    select usCD.brand as Brand,

```

```

        usCD.model as Model,
        usCD.year as ProdYear,
        usCD.price as USPrice,
        null      as PLPrice,
        null      as WorldPrice,
        dD.id     as YearID,
        cD.id     as CarID
    from stageSc.USA_cars_datasets usCD
        left join modelSc.dateDim dD on dD.year = usCD.year
        left join modelSc.carsDim cD on cD.brand = usCD.brand
    and cD.model = usCD.Model
union
select cP.Manufacturer as Brand,
       cP.model        as Model,
       cP.[Prod. year] as ProdYear,
       null            as USPrice,
       null            as PLPrice,
       cP.price        as WorldPrice,
       dD.id           as YearID,
       cD.id           as CarID
    from stageSc.Car_prices cP
        left join modelSc.dateDim dD on dD.year = cP.[Prod. year]
        left join modelSc.carsDim cD on cD.brand = cP.Manufacturer
    and cD.model = cP.Model
union
select cPPl.mark          as Brand,
       cPPl.model         as Model,
       cPPl.year          as ProdYear,
       null               as USPrice,
       dbo.zlotyToUsd(cPPl.price) as PLPrice,
       null               as WorldPrice,
       dD.id              as YearID,
       cD.id              as CarID
    from stageSc.Car_Prices_Poland_Kaggle cPPl
        left join modelSc.dateDim dD on dD.year = cPPl.year
        left join modelSc.carsDim cD on cD.brand = cPPl.mark
    and cD.model = cPPl.model
)
/*insert
into modelSc.priceFact(polPrice, usPrice, usedPrice, yearID, carID)*/
select PLPrice, USPrice, WorldPrice, YearID, CarID
from tmpPrices
order by CarID, YearID

--

select usCD.brand as Brand, usCD.model as Model, usCD.year as ProdYear,
usCD.price as Price, usCD.mileage as Mileage
from stageSc.USA_cars_datasets usCD
    left join modelSc.dateDim dD on dD.year = usCD.year
    left join modelSc.carsDim cD on cD.brand = usCD.brand
and cD.model = usCD.Model

select cP.Manufacturer as Brand,
       cP.model        as Model,

```

```

        cP.[Prod. year] as ProdYear,
        cP.price        as Price,
        cP.Mileage      as Mileage
from stageSc.Car_prices cP
    left join modelSc.dateDim dD on dD.year = cP.[Prod. year]
    left join modelSc.carsDim cD on cD.brand = cP.Manufacturer
and cD.model = cP.Model

select cPPl.mark          as Brand,
       cPPl.model         as Model,
       cPPl.year          as ProdYear,
       dbo.zlotyToUsd(cPPl.price) as Price,
       cPPl.Mileage       as Mileage
from stageSc.Car_Prices_Poland_Kaggle cPPl
    left join modelSc.dateDim dD on dD.year = cPPl.year
    left join modelSc.carsDim cD on cD.brand = cPPl.mark
and cD.model = cPPl.model

select cD.brand, cD.model, dD.year, polPrice, usPrice, usedPrice as
WorldPrice
from modelSc.priceFact pf
inner join modelSc.carsDim cD on cD.id = pf.carID
inner join modelSc.dateDim dD on dD.id = pf.yearID
where carID = 58

```

Таблиці Stage зони:

```
create table Car_Prices_Poland_Kaggle
(
    id            int,
    mark          varchar(max) ,
    model         varchar(max) ,
    generation_name varchar(max) ,
    year         int,
    mileage       int,
    vol_engine    int,
    fuel          varchar(max) ,
    city          varchar(max) ,
    province      varchar(max) ,
    price         int
)
go

create table Car_prices
(
    ID            int,
    Price         int,
    Levy          varchar(max) ,
    Manufacturer  varchar(max) ,
    Model         varchar(max) ,
    [Prod. year]  int,
    Category      varchar(max) ,
    [Leather interior] varchar(max) ,
    [Fuel type]   varchar(max) ,
    [Engine volume] varchar(max) ,
    Mileage       varchar(max) ,
    Cylinders     float,
    [Gear box type] varchar(max) ,
    [Drive wheels] varchar(max) ,
    Doors         varchar(max) ,
    Wheel         varchar(max) ,
    Color         varchar(max) ,
    Airbags       int
)
go

create table USA_cars_datasets
(
    ID            int,
    price         int,
    brand         varchar(max) ,
    model         varchar(max) ,
    year         int,
    title_status  varchar(max) ,
    mileage       float,
    color         varchar(max) ,
    vin          varchar(max) ,
    lot           int,
    state         varchar(max) ,
)
```



```
country    varchar(max),  
condition  varchar(max)  
)  
go
```

Таблиці основного сховища:

```
use adisDB
go

create table modelSc.carsDim
(
    id      int identity
           constraint carsDim_pk
           primary key nonclustered,
    brand  varchar(max),
    model  varchar(max)
)
go

create unique index carsDim_id_uindex
on modelSc.carsDim (id)
go

create table modelSc.dateDim
(
    id      int identity
           constraint dateDim_pk
           primary key nonclustered,
    year    int
)
go

create unique index dateDim_id_uindex
on modelSc.dateDim (id)
go

create table modelSc.priceFact
(
    id          int identity
               constraint priceFact_pk
               primary key nonclustered,
    polPrice    float,
    usPrice     float,
    usedPrice   float,
    yearID      int
               constraint priceFact_dateDim_id_fk
               references modelSc.dateDim,
    carID       int
               constraint priceFact_carsDim_id_fk
               references modelSc.carsDim
)
go

create unique index priceFact_id_uindex
on modelSc.priceFact (id)
go
```