1 Створити стовпчикову і звичайну бд

<https://github.com/mariadb-corporation/mariadb-columnstore-sample-data>

(приклад застосунку <https://github.com/orgs/mariadb-developers/repositories?q=flights-app&language=>)

2 Розрахувати сумарну затримку по містах

3 Порахувати кількість польотів по містах

4 Знайти місто з найменшою і найбільшою затримкою

5 Знайти всі польоти з затримкою більше за середній час затримки

6 Заміряти вбудованими методами об'єм БД та швидкість виконання запитів. Порівняти звичайну і стовпчикову

Для виконання цієї лабораторної роботи я використовував Microsoft SQL Server.

**Створення стовпчикової бази даних:**

USE columnstore\_bts;

CREATE TABLE airlines (

iata\_code varchar(2) DEFAULT NULL,

airline varchar(30) DEFAULT NULL

)

CREATE CLUSTERED COLUMNSTORE INDEX MyColumnstoreIndex ON airlines;

CREATE TABLE airports (

iata\_code varchar(3) DEFAULT NULL,

airport varchar(80) DEFAULT NULL,

city varchar(30) DEFAULT NULL,

state varchar(2) DEFAULT NULL,

country varchar(30) DEFAULT NULL,

latitude decimal(11,4) DEFAULT NULL,

longitude decimal(11,4) DEFAULT NULL

)

CREATE CLUSTERED COLUMNSTORE INDEX MyColumnstoreIndex ON airports;

CREATE TABLE flights (

year smallint DEFAULT NULL,

month tinyint DEFAULT NULL,

day tinyint DEFAULT NULL,

day\_of\_week tinyint DEFAULT NULL,

fl\_date date DEFAULT NULL,

carrier varchar(2) DEFAULT NULL,

tail\_num varchar(6) DEFAULT NULL,

fl\_num smallint DEFAULT NULL,

origin varchar(5) DEFAULT NULL,

dest varchar(5) DEFAULT NULL,

crs\_dep\_time varchar(4) DEFAULT NULL,

dep\_time varchar(4) DEFAULT NULL,

dep\_delay decimal(13,2) DEFAULT NULL,

taxi\_out decimal(13,2) DEFAULT NULL,

wheels\_off varchar(4) DEFAULT NULL,

wheels\_on varchar(4) DEFAULT NULL,

taxi\_in decimal(13,2) DEFAULT NULL,

crs\_arr\_time varchar(4) DEFAULT NULL,

arr\_time varchar(4) DEFAULT NULL,

arr\_delay decimal(13,2) DEFAULT NULL,

cancelled decimal(13,2) DEFAULT NULL,

cancellation\_code varchar(20) DEFAULT NULL,

diverted decimal(13,2) DEFAULT NULL,

crs\_elapsed\_time decimal(13,2) DEFAULT NULL,

actual\_elapsed\_time decimal(13,2) DEFAULT NULL,

air\_time decimal(13,2) DEFAULT NULL,

distance decimal(13,2) DEFAULT NULL,

carrier\_delay decimal(13,2) DEFAULT NULL,

weather\_delay decimal(13,2) DEFAULT NULL,

nas\_delay decimal(13,2) DEFAULT NULL,

security\_delay decimal(13,2) DEFAULT NULL,

late\_aircraft\_delay decimal(13,2) DEFAULT NULL

)

CREATE CLUSTERED COLUMNSTORE INDEX MyColumnstoreIndex ON flights;

**Створення звичайної бази даних:**

USE innodb\_bts;

CREATE TABLE airlines (

iata\_code varchar(2) NOT NULL PRIMARY KEY,

airline varchar(30)

);

CREATE TABLE airports (

iata\_code varchar(3) NOT NULL PRIMARY KEY,

airport varchar(80) DEFAULT NULL,

city varchar(30) DEFAULT NULL,

state varchar(2) DEFAULT NULL,

country varchar(30) DEFAULT NULL,

latitude decimal(11,4) DEFAULT NULL,

longitude decimal(11,4) DEFAULT NULL

);

CREATE TABLE flights (

year smallint DEFAULT NULL,

month tinyint DEFAULT NULL,

day tinyint DEFAULT NULL,

day\_of\_week tinyint DEFAULT NULL,

fl\_date date DEFAULT NULL,

carrier varchar(2) DEFAULT NULL,

tail\_num varchar(6) DEFAULT NULL,

fl\_num smallint DEFAULT NULL,

origin varchar(5) DEFAULT NULL,

dest varchar(5) NOT NULL,

crs\_dep\_time varchar(4) DEFAULT NULL,

dep\_time varchar(4) DEFAULT NULL,

dep\_delay decimal(13,2) DEFAULT NULL,

taxi\_out decimal(13,2) DEFAULT NULL,

wheels\_off varchar(4) DEFAULT NULL,

wheels\_on varchar(4) DEFAULT NULL,

taxi\_in decimal(13,2) DEFAULT NULL,

crs\_arr\_time varchar(4) DEFAULT NULL,

arr\_time varchar(4) DEFAULT NULL,

arr\_delay decimal(13,2) DEFAULT NULL,

cancelled decimal(13,2) DEFAULT NULL,

cancellation\_code varchar(20) DEFAULT NULL,

diverted decimal(13,2) DEFAULT NULL,

crs\_elapsed\_time decimal(13,2) DEFAULT NULL,

actual\_elapsed\_time decimal(13,2) DEFAULT NULL,

air\_time decimal(13,2) DEFAULT NULL,

distance decimal(13,2) DEFAULT NULL,

carrier\_delay decimal(13,2) DEFAULT NULL,

weather\_delay decimal(13,2) DEFAULT NULL,

nas\_delay decimal(13,2) DEFAULT NULL,

security\_delay decimal(13,2) DEFAULT NULL,

late\_aircraft\_delay decimal(13,2) DEFAULT NULL

);

**Для імпорту даних з csv-файлів використовувався наступний код:**

BULK INSERT airlines

FROM 'C:\tmp\mariadb-columnstore-sample-data\schema\airlines.csv'

WITH

(

FIRSTROW = 1,

FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n'

);

BULK INSERT airports

FROM 'C:\tmp\mariadb-columnstore-sample-data\schema\airports.csv'

WITH

(

FIRSTROW = 2,

FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n'

);

BULK INSERT flights

FROM 'C:\tmp\mariadb-columnstore-sample-data\schema\flights.csv'

WITH

(

FIRSTROW = 1,

FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n'

);

**Розрахунок сумарної затримки по містах:**

SELECT

airports.city AS city,

SUM(flights.arr\_delay) AS total\_delay

FROM

flights

JOIN

airports

ON flights.origin = airports.iata\_code

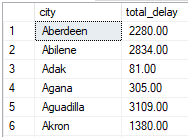
JOIN

airports AS a\_dest

ON flights.dest = a\_dest.iata\_code

GROUP BY airports.city

ORDER BY airports.city;



Час виконання для стовпчикової БД: 485 мс

Час виконання для звичайної БД: 1709 мс

**Розрахунок кількості польотів по містах:**

SELECT airports.city,

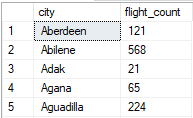
COUNT(\*) AS flight\_count

FROM airports

JOIN flights ON airports.iata\_code = flights.dest

GROUP BY airports.city

ORDER BY airports.city;



Час виконання для стовпчикової БД: 118 мс

Час виконання для звичайної БД: 716 мс

**Знаходження міста з найменшою та найбільшою затримкою:**

SELECT TOP(1) airports.city, AVG(flights.arr\_delay) AS avg\_arr\_delay

FROM airports

JOIN flights ON airports.iata\_code = flights.dest

GROUP BY airports.city

ORDER BY avg\_arr\_delay ASC;

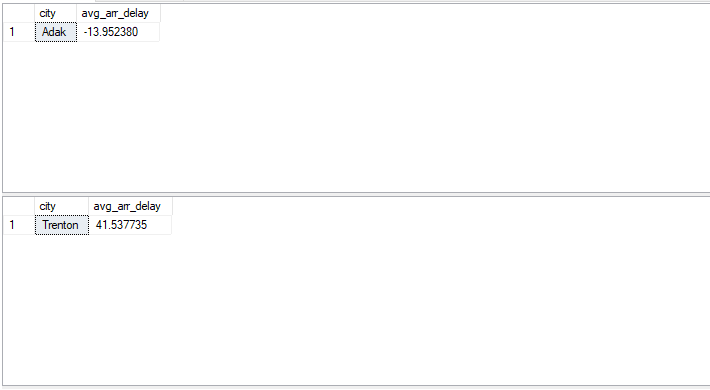
SELECT TOP(1) airports.city, AVG(flights.arr\_delay) AS avg\_arr\_delay

FROM airports

JOIN flights ON airports.iata\_code = flights.dest

GROUP BY airports.city

ORDER BY avg\_arr\_delay DESC;



Час виконання для стовпчикової БД: 146 мс, 118 мс

Час виконання для звичайної БД: 1036 мс, 939 мс

**Знаходження всіх польотів з затримкою більше за середній час затримки:**

SELECT \* FROM flights WHERE arr\_delay > (SELECT AVG(arr\_delay) FROM flights);



Час виконання для стовпчикової БД: 12368 мс

Час виконання для звичайної БД: 12457 мс

Розмір стовпчикової БД: 80 мб

Розмір звичайної БД: 848 мб

**Висновки:**

Створено стовпчикову та звичайну БД та декілька запитів до них. В результаті можна зробити висновок, що стовпчикова база даних значно краще підходить для зберігання великих об’ємів даних, адже вона займає набагато менше місця та швидше обробляє запити.