

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
з дисципліни
Програмування інтелектуальних інформаційних систем

Виконав:

студент групи ІІІ-13:
Ал Хадам М.Р.

Перевірила:

ст. викладач.
Баришич Л.М.

Київ 2023

Завдання

1. Створити стовпчикову і звичайну бд
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. Виміряти вбудованими методами об'єм БД та швидкість виконання запитів. Порівняти звичайну і стовпчикову

Хід виконання

Створення таблиць для бази даних columnstore_bts:

```
DROP DATABASE IF EXISTS `columnstore_bts`;

CREATE DATABASE `columnstore_bts`;

USE `columnstore_bts`;

CREATE TABLE `airlines`
(
  `iata_code` varchar(2) DEFAULT NULL,
  `airline` varchar(30) DEFAULT NULL
) ENGINE = Columnstore
DEFAULT CHARSET = utf8mb3
COLLATE = utf8mb3_general_ci;

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
) ENGINE = Columnstore
DEFAULT CHARSET = utf8mb3
```

```
COLLATE = utf8mb3_general_ci;

CREATE TABLE `flights`
(
  `year` smallint(6) DEFAULT NULL,
  `month` tinyint(4) DEFAULT NULL,
  `day` tinyint(4) DEFAULT NULL,
  `day_of_week` tinyint(4) DEFAULT NULL,
  `fl_date` date DEFAULT NULL,
  `carrier` varchar(2) DEFAULT NULL,
  `tail_num` varchar(6) DEFAULT NULL,
  `fl_num` smallint(6) 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
) ENGINE = Columnstore
```

```
DEFAULT CHARSET = utf8mb3  
COLLATE = utf8mb3_general_ci;
```

Створення таблиць для бази даних innodb_bts:

```
DROP DATABASE IF EXISTS `innodb_bts`;  
  
CREATE DATABASE `innodb_bts`;  
  
USE `innodb_bts`;  
  
CREATE TABLE `airlines`  
(  
  `iata_code` varchar(2) NOT NULL,  
  `airline` varchar(30) DEFAULT NULL,  
  PRIMARY KEY (`iata_code`),  
  KEY `airline` (`airline`)  
) ENGINE = InnoDB  
DEFAULT CHARSET = utf8mb3  
COLLATE = utf8mb3_general_ci;  
  
CREATE TABLE `airports`  
(  
  `iata_code` varchar(3) NOT 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,  
  PRIMARY KEY (`iata_code`),  
  KEY `state` (`state`)  
) ENGINE = InnoDB  
DEFAULT CHARSET = utf8mb3  
COLLATE = utf8mb3_general_ci;  
  
CREATE TABLE `flights`  
(
```

```
`year` smallint(6) DEFAULT NULL,
`month` tinyint(4) DEFAULT NULL,
`day` tinyint(4) DEFAULT NULL,
`day_of_week` tinyint(4) DEFAULT NULL,
`fl_date` date DEFAULT NULL,
`carrier` varchar(2) DEFAULT NULL,
`tail_num` varchar(6) DEFAULT NULL,
`fl_num` smallint(6) 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,
KEY `carrier` (`carrier`),
KEY `year` (`year`),
KEY `carrier_delay` (`carrier_delay`),
KEY `weather_delay` (`weather_delay`),
KEY `nas_delay` (`nas_delay`),
```

```
KEY `security_delay` (`security_delay`),
KEY `late_aircraft_delay` (`late_aircraft_delay`),
KEY `arr_delay` (`arr_delay`),
KEY `month` (`month`),
KEY `dest` (`dest`)
) ENGINE = InnoDB

DEFAULT CHARSET = utf8mb3

COLLATE = utf8mb3_general_ci;
```

Завантаження даних до таблиць у створених базах:

```
USE columnstore_bts;

LOAD DATA INFILE '../.../data/airlines.csv'
INTO TABLE airlines
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

LOAD DATA INFILE '../.../data/airports.csv'
INTO TABLE airports
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

SET SESSION sql_mode = '';

LOAD DATA INFILE '../.../data/flights.csv'
INTO TABLE flights
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

USE innodb_bts;

LOAD DATA INFILE '../.../data/airlines.csv'
INTO TABLE airlines
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';
```

```

LOAD DATA INFILE '../.../data/airports.csv'
INTO TABLE airports
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA INFILE '../.../data/flights.csv'
INTO TABLE flights
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n';

```

Запити завдання

Розрахувати сумарну затримку по містах:

```

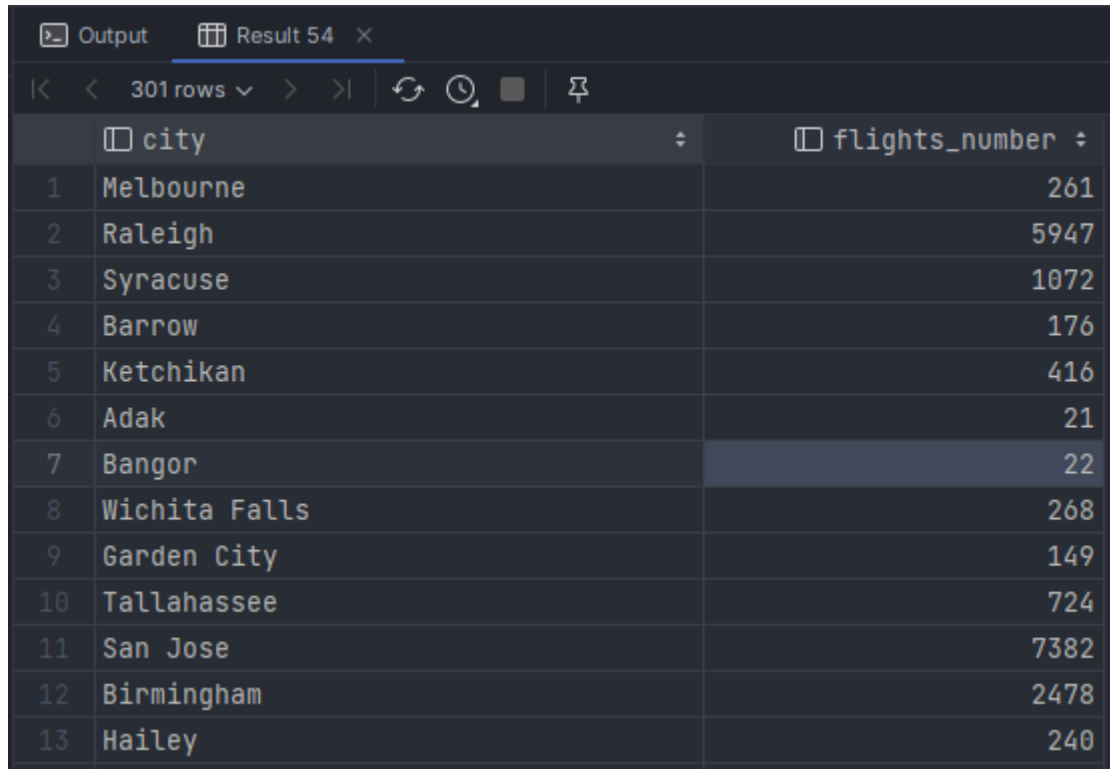
DROP VIEW IF EXISTS delays;
CREATE VIEW delays AS
SELECT a.city AS city,
sum(f.arr_delay + f.dep_delay) AS total_delay
FROM flights f
JOIN airports a ON f.dest = a.iata_code
GROUP BY a.city;

```

Output Result 53		
301 rows		
	city	total_delay
1	Raleigh	98745.00
2	Syracuse	28424.00
3	Melbourne	1732.00
4	Barrow	2197.00
5	Ketchikan	4207.00
6	Adak	-298.00
7	Bangor	560.00
8	Birmingham	43075.00
9	Tallahassee	12093.00
10	San Jose	68481.00
11	Pasco	8053.00
12	Hailey	1486.00
13	Crescent City	2306.00

Порахувати кількість польотів по містах:

```
SELECT a.city AS city,
count(*) AS flights_number
FROM flights f
JOIN airports a ON f.dest = a.iata_code
GROUP BY a.city;
```

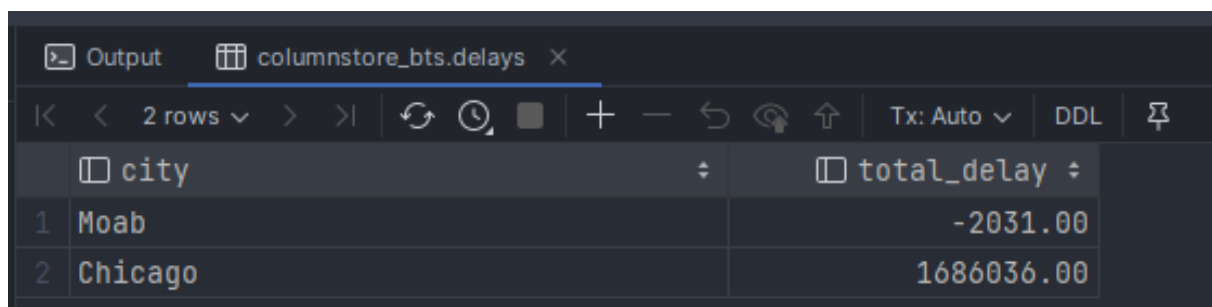


The screenshot shows a database interface with a query result table. The table has two columns: 'city' and 'flights_number'. There are 13 rows of data. The row for 'Bangor' is highlighted.

	city	flights_number
1	Melbourne	261
2	Raleigh	5947
3	Syracuse	1072
4	Barrow	176
5	Ketchikan	416
6	Adak	21
7	Bangor	22
8	Wichita Falls	268
9	Garden City	149
10	Tallahassee	724
11	San Jose	7382
12	Birmingham	2478
13	Hailey	240

Знайти місто з найменшою і найбільшою затримкою:

```
SELECT *
FROM delays
WHERE total_delay = (SELECT MIN(total_delay) FROM delays)
OR total_delay = (SELECT MAX(total_delay) FROM delays);
```

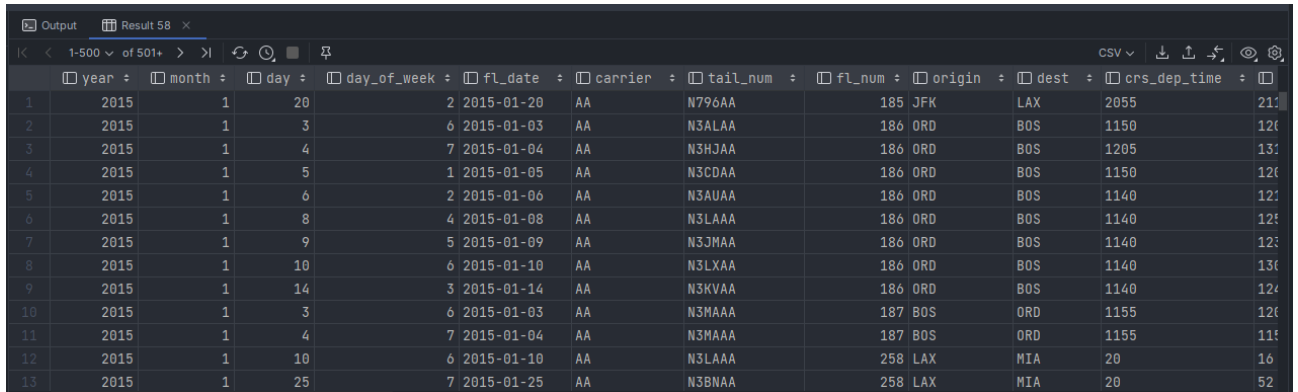


The screenshot shows a database interface with a query result table. The table has two columns: 'city' and 'total_delay'. There are 2 rows of data. The row for 'Chicago' is highlighted.

	city	total_delay
1	Moab	-2031.00
2	Chicago	1686036.00

Знайти всі польоти з затримкою більше за середній час затримки:

```
SELECT *, f.arr_delay + f.dep_delay AS total_delay
FROM flights f
HAVING total_delay > (SELECT avg(f.arr_delay + f.dep_delay)
FROM flights f);
```



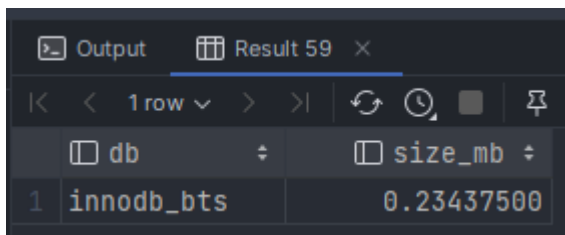
	year	month	day	day_of_week	fl_date	carrier	tail_num	fl_num	origin	dest	crs_dep_time	
1	2015	1	20	2	2015-01-20	AA	N796AA	185	JFK	LAX	2055	21
2	2015	1	3	6	2015-01-03	AA	N3ALAA	186	ORD	BOS	1150	12
3	2015	1	4	7	2015-01-04	AA	N3HJAA	186	ORD	BOS	1205	13
4	2015	1	5	1	2015-01-05	AA	N3CDAA	186	ORD	BOS	1150	12
5	2015	1	6	2	2015-01-06	AA	N3AUAA	186	ORD	BOS	1140	12
6	2015	1	8	4	2015-01-08	AA	N3LAAA	186	ORD	BOS	1140	12
7	2015	1	9	5	2015-01-09	AA	N3JMAA	186	ORD	BOS	1140	12
8	2015	1	10	6	2015-01-10	AA	N3LXAA	186	ORD	BOS	1140	13
9	2015	1	14	3	2015-01-14	AA	N3KVAA	186	ORD	BOS	1140	12
10	2015	1	3	6	2015-01-03	AA	N3MAAA	187	BOS	ORD	1155	12
11	2015	1	4	7	2015-01-04	AA	N3MAAA	187	BOS	ORD	1155	11
12	2015	1	10	6	2015-01-10	AA	N3LAAA	258	LAX	MIA	20	16
13	2015	1	25	7	2015-01-25	AA	N3BNAA	258	LAX	MIA	20	52

Заміряти вбудованими методами об’єм БД та швидкість виконання запитів.

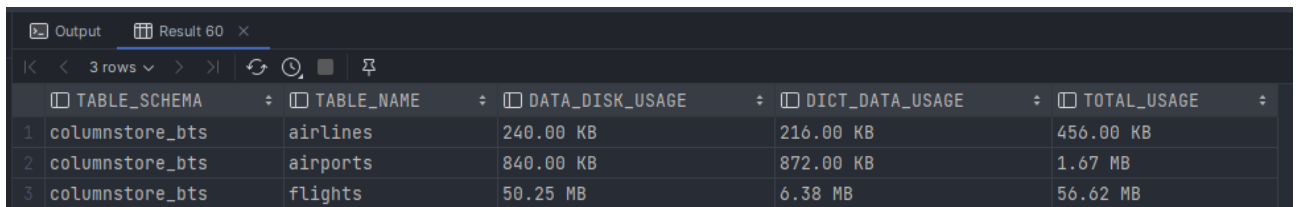
Порівняти звичайну і стовпчикову:

```
SELECT table_schema AS db,
SUM(data_length + index_length) / 1024 / 1024 AS size_mb
FROM information_schema.tables
WHERE table_schema = 'innodb_bts';

CALL columnstore_info.table_usage('columnstore_bts', NULL);
```



db	size_mb
innodb_bts	0.23437500



	TABLE_SCHEMA	TABLE_NAME	DATA_DISK_USAGE	DICT_DATA_USAGE	TOTAL_USAGE
1	columnstore_bts	airlines	240.00 KB	216.00 KB	456.00 KB
2	columnstore_bts	airports	840.00 KB	872.00 KB	1.67 MB
3	columnstore_bts	flights	50.25 MB	6.38 MB	56.62 MB

Висновок

На лабораторній роботі ми успішно створили два типи баз даних: стовпчикову та звичайну. Ми використали вбудовані інструменти для імпорту даних з файлів `airlines.csv`, `airports.csv` та `flights.csv`, і відповідно до наших поставлених завдань створили SQL запити. Далі ми провели порівняльний аналіз двох різних систем управління базами даних - `columnstore` та `innodb` - враховуючи такі показники, як обсяг даних та швидкість виконання запитів. Отримані результати були систематизовані та представлені у вигляді таблиці для подальшого аналізу.

Таблиця порівняння часу виконання кожного з запитів на різних типах баз даних та витраченої пам'яті представлено далі.

	Columnstore	Innodb
Task 1	347 ms	3 s
Task 2	251 ms	1 s 309 ms
Task 3	649 ms	10 s
Task 4	339 ms	2 s 83 ms
Memory	59 mb	0.234 mb