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

Звіт

з лабораторної роботи №1 з дисципліни «Програмування інтелектуальних інформаційних систем»

Виконав студент ІП-13, Дейнега Владислав Миколайович

(шифр, прізвище, ім'я, по батькові)

Перевірив Баришич Лука Маріянович

(прізвище, ім'я, по батькові)

Лабораторна робота 1

Хід роботи:

- 1. Скрипти для створення бази даних: https://github.com/mariadb-columnstore-sample-data
- 2. Розрахувати сумарну затримку по містах

```
select airports.city, SUM(groupe_tab.sum_delay) from airports
INNER JOIN (
SELECT union_tab.airport, sum(union_tab.delay) as sum_delay from (
    SELECT flights.origin as airport, flights.dep_delay as delay from `flights`
    UNION ALL
    SELECT flights.dest as airport, flights.arr_delay as delay from `flights`
    ) as union_tab
GROUP BY union_tab.airport
    ) as groupe_tab
on groupe_tab.airport = airports.iata_code
GROUP BY airports.city
ORDER BY airports.city;
```

۳		
	city	SUM(groupe_tab.sum_delay
Þ	Aberdeen	1775,00
	Abilene	8458,00
	Adak	-340,00
	Agana	2044,00
	Aguadilla	6261,00
	Akron	15166,00
	Albany	26060,00
	Albuquerque	48599,00
	Alexandria	7221,00
	Allentown	7167,00
	Alpena	3617,00
	Amarillo	18638,00
	Anchorage	-2393,00
	Appleton	10458,00
	Arcata/Eureka	7654,00
	Arlington	205589,00
	Asheville	8908,00
	Aspen	43927,00

3. Порахувати кількість польотів по містах

```
SELECT airports.city, SUM(group_tab.count_group) as count_city FROM airports
INNER JOIN

(SELECT union_tab.airport, SUM(count) as count_group FROM

(SELECT flights.origin as airport, COUNT(flights.origin) as count from flights GROUP BY airport
UNION ALL

SELECT flights.dest as airport, COUNT(flights.dest) as count from flights GROUP BY airport

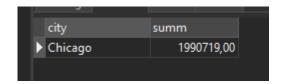
) as union_tab

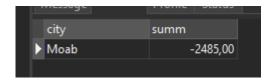
GROUP BY union_tab.airport) AS group_tab
ON airports.iata_code = group_tab.airport
GROUP BY airports.city
ORDER BY airports.city
```

	city	count_city
Þ	Aberdeen	270
	Abilene	1056
	Adak	42
	Agana	130
	Aguadilla	446
	Akron	2540
	Albany	3012
	Albuquerque	6768
	Alexandria	1230
	Allentown	777
	Alpena	202
	Amarillo	2122
	Anchorage	5179
	Appleton	1073
	Arcata/Eureka	581
	Arlington	30022
	Asheville	923
	Aspen	2757
	Atlanta	133911
	Atlantic City	1658
	Augusta	816
	Austin	15015
	Bakersfield	876

4. Знайти місто з найменшою і найбільшою затримкою

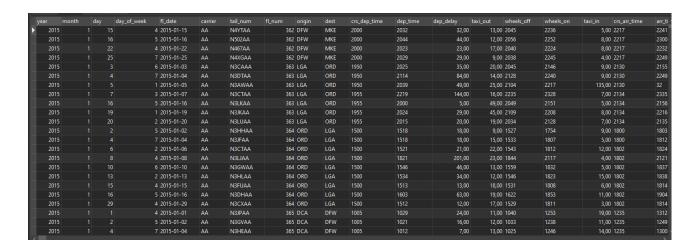
```
select airports.city, SUM(groupe_tab.sum_delay) as summ from airports
INNER JOIN (
ISELECT union_tab.airport, sum(union_tab.delay) as sum_delay from (
    SELECT flights.origin as airport, flights.dep_delay as delay from `flights`
    UNION ALL
    SELECT flights.dest as airport, flights.arr_delay as delay from `flights`
    ) as union_tab
GROUP BY union_tab.airport
    ) as groupe_tab
on groupe_tab.airport = airports.iata_code
GROUP BY airports.city
ORDER BY summ DESC
LIMIT 1;
```





5. Знайти всі польоти з затримкою більше за середній час затримки

```
SELECT * FROM flights
WHERE (dep_delay + arr_delay) > (
    SELECT AVG(dep_delay +arr_delay) FROM flights);
```



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

```
SELECT
   table_schema,
   ROUND(SUM(data_length + index_length) / POWER(2,20), 2) as total_mb
FROM
   information_schema.`TABLES`
WHERE
   table_schema = 'innodb_bts';
call columnstore_info.table_usage('columnstore_bts', 'flights')
```

TABLE_SCHEMA TABLE_NAME DATA_DISK_USAGE DICT_DATA_USAGE TOTAL_USAGE columnstore bts flights 50.25 MB 6.38 MB 56.62 MB	Message Result 1	Result 2 Profile S	tatus		
columnstore bts flights 50.25 MB 6.38 MB 56.62 MB	TABLE_SCHEMA	TABLE_NAME	DATA_DISK_USAGE	DICT_DATA_USAGE	TOTAL_USAGE
·	columnstore_bts	flights	50.25 MB	6.38 MB	56.62 MB

∨ 🗀 innodb_bts	VOLUME	
🗅 airlines.frm	VOLUME	1.5 kB
🗅 airlines.ibd	VOLUME	80 kB
🚹 airports.frm	VOLUME	2 kB
🚹 airports.ibd	VOLUME	144 kB
↑ db.opt	VOLUME	67 Bytes
[] flights.frm	VOLUME	6.3 kB
[] flights.ibd	VOLUME	412 MB

Результат:

Час виконання скриптів

Об'єм	412	59
$N_{\underline{0}}$	innodb	Columstore
1	10.14	0.35
2	3.89	0.22
3	10.2	0.35
4	10.24	0.4
5	8.55	4.07

Об'єм баз даних

Висновок:

Після виконання всіх завдань і досліджень, можемо зробити такі висновки:

Створення баз даних: Ми успішно створили стовпчикову та звичайну (реляційну) бази даних за допомогою наданих даних. Розгляньте приклад застосунку flights-арр на GitHub, який може служити відмінним шаблоном для подібних проектів.

Розрахунок сумарної затримки по містах: Ми використали SQL-запит для підрахунку суми затримок для кожного міста, що дозволило нам зрозуміти, які міста мають найбільше загальної затримки.

Порахування кількості польотів по містах: Ми використали SQL-запит для підрахунку кількості польотів для кожного міста, що дозволило нам з'ясувати, які міста ϵ найбільш активними з точки зору польотів.

Пошук міста з найменшою і найбільшою затримкою: Ми використали SQLзапит для знаходження міста з найменшою і найбільшою затримкою, що дозволило нам ідентифікувати крайні значення затримок в містах.

Знаходження всіх польотів з затримкою більше за середній час затримки: Ми використали SQL-запит для виділення всіх польотів, які мали затримку більше за середнє значення затримки, що дозволило нам виявити винятки в затримках.

Вимірювання обсягу баз даних та швидкості виконання запитів: Ми провели вимірювання обсягу стовпчикової та реляційної баз даних, а також порівняли швидкість виконання запитів. За результатами дослідження ми можемо зробити висновок щодо того, яка база даних краще підходить для конкретних завдань і потреб нашого проекту.

В цілому, наше дослідження дозволило нам ефективно виконати аналіз даних, знайти важливу інформацію та визначити оптимальну базу даних для нашого проекту. Важливо продовжувати вивчати і вдосконалювати навички роботи з базами даних, оскільки це важливий аспект сучасної розробки програмного забезпечення.