

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-12 Басараб Олег Андрійович

Перевірив Баришич Лука Маріянович

Київ 2023

## Хід роботи

### 1. Скрипти створення баз / таблиць

SQL-скрипт для створення бази даних columnstore\_bts і її таблиць:

```
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;

```

SQL-скрипт для створення бази даних innodb \_bts і її таблиць:

```
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;

```

BASH-скрипт для створення баз даних columnstore\_bts та innodb\_bts, таблиць цих БД і завантаження даних з файлів airports.csv, airlines.csv та flights.csv до таблиць цих БД:

```
set -euo pipefail # Enable strict mode
```

```
MARIADB=$(which mariadb)
```

```
CPIMPORT=$(which cpimport)
```

```
SCHEMA_DIR=$(readlink -f ./schema)
```

```
NAME1='airports'
```

```
NAME2='airlines'
```

```
NAME3='flights'
```

```
GREEN="\033[0;32m'
```

```
RED="\033[0;31m'
```

```
NC="\033[0m'
```

```
if $MARIADB <"${SCHEMA_DIR}"/columnstore_schema.sql &>/dev/null; then
```

```
    echo -e "Creating 'columnstore_bts' schema..." "${GREEN}done${NC}"
```

```
else
```

```
    echo -e "Creating 'columnstore_bts' schema..." "${RED}fail${NC}"
```

```

        exit 1
    fi

# Loading data into ColumnStore tables.
for CSV_FILE in ${NAME1} ${NAME2} ${NAME3}; do
    echo -e "\nLoading '${CSV_FILE}.csv' with cpimport ..."
    if ! $CPIMPORT -m 1 -s ',' -E "" columnstore_bts "${CSV_FILE}" -l
"${SCHEMA_DIR}/${CSV_FILE}.csv"; then
        echo -e "loading '${CSV_FILE}.csv' ... ${RED}fail${NC}"
        exit 1
    fi
done

printf "\nDo you want to include an InnoDB comparison schema? (Y/N) "
read -r COMPARE

if [[ $COMPARE == [yY] ]]; then
    if $MARIADB <"${SCHEMA_DIR}"/innodb_schema.sql &>/dev/null; then
        echo -e "\nCreating 'innodb_bts' schema..." "${GREEN}done${NC}\n"
        for CSV_FILE in "${NAME1}" "${NAME2}" "${NAME3}"; do
            echo -e "\nLoading '${CSV_FILE}.csv' with LDI ..."
            if ! $MARIADB --database="innodb_bts" --init-command="SET sql_mode=""
-vvv -e \
                "LOAD DATA INFILE '${SCHEMA_DIR}/${CSV_FILE}.csv'
                INTO TABLE ${CSV_FILE}
                FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY \"
                LINES TERMINATED BY '\n';" | awk '!/Bye/'; then
                echo -e "loading '${CSV_FILE}.csv' ... ${RED}fail${NC}"
                exit 1
            fi
        done
    fi
done

```

```
else
    echo -e "Creating 'innodb_bts' schema..." "${RED}fail${NC}"
    exit 1
fi
elif [[ $COMPARE == [nN] ]]; then
    exit 0
else
    echo -e "Invalid input. Please enter Y or N.\n"
    exit 1
fi
```



## 2. Запити

### 2.1. SQL-запит для розрахунку сумарної затримки по містах для БД columnstore\_bts:

```
select arr_delays.city, total_arr_delay + total_dep_delay as total_delay
from
  (select a.city, sum(f.dep_delay) as total_dep_delay
   from columnstore_bts.flights f
   join columnstore_bts.airports a on f.origin = a.iata_code
   group by a.city) as dep_delays
join
  (select a.city, sum(f.arr_delay) as total_arr_delay
   from columnstore_bts.flights f
   join columnstore_bts.airports a on f.dest = a.iata_code
   group by a.city) as arr_delays
on dep_delays.city = arr_delays.city
order by arr_delays.city;
```

Час виконання запиту – 0.402 с;

### SQL-запит для розрахунку сумарної затримки по містах для БД innodb\_bts:

```
select arr_delays.city, total_arr_delay + total_dep_delay as total_delay
from
  (select a.city, sum(f.dep_delay) as total_dep_delay
   from innodb_bts.flights f
   join innodb_bts.airports a on f.origin = a.iata_code
   group by a.city) as dep_delays
join
  (select a.city, sum(f.arr_delay) as total_arr_delay
   from innodb_bts.flights f
   join innodb_bts.airports a on f.dest = a.iata_code
   group by a.city) as arr_delays
on dep_delays.city = arr_delays.city
order by arr_delays.city;
```

Час виконання запиту – 5.555 с;

Результат виконання SQL-запитів для розрахунку сумарної затримки по містах:

	city	total_delay
1	Aberdeen	1775.00
2	Abilene	8458.00
3	Adak	-340.00
4	Agana	2044.00
5	Aguadilla	6261.00
6	Akron	15166.00
7	Albany	26060.00
8	Albuquerque	48599.00
9	Alexandria	7221.00
10	Allentown	7167.00
11	Alpena	3617.00
12	Amarillo	18638.00
13	Anchorage	-2393.00
14	Appleton	10458.00
15	Arcata/Eureka	7654.00
16	Arlington	205589.00
17	Asheville	8908.00
18	Aspen	43927.00
19	Atlanta	740512.00

## 2.2. SQL-запит для обрахунку кількості польотів по містах для БД columnstore\_bts:

```
select arrivals.city, total_departures + total_arrivals as total_flights
from
  (select a.city, count(*) as total_arrivals
   from columnstore_bts.flights f
   join columnstore_bts.airports a on f.dest = a.iata_code
   group by a.city) as arrivals
join
  (select a.city, count(*) as total_departures
   from columnstore_bts.flights f
   join columnstore_bts.airports a on f.origin = a.iata_code
   group by a.city) as departures
on departures.city = arrivals.city
order by arrivals.city;
```

Час виконання запиту – 0.320 с;

## SQL-запит для обрахунку кількості польотів по містах для БД innodb\_bts:

```
select arrivals.city, total_departures + total_arrivals as total_flights
from
  (select a.city, count(*) as total_arrivals
   from innodb_bts.flights f
   join innodb_bts.airports a on f.dest = a.iata_code
   group by a.city) as arrivals
join
  (select a.city, count(*) as total_departures
   from innodb_bts.flights f
   join innodb_bts.airports a on f.origin = a.iata_code
   group by a.city) as departures
on departures.city = arrivals.city
order by arrivals.city;
```

Час виконання запиту – 3.528 с;

Результат виконання SQL-запитів для обрахунку кількості польотів по містах:

	city	total_flights
1	Aberdeen	270
2	Abilene	1056
3	Adak	42
4	Agana	130
5	Aguadilla	446
6	Akron	2540
7	Albany	3012
8	Albuquerque	6768
9	Alexandria	1230
10	Allentown	777
11	Alpena	202
12	Amarillo	2122
13	Anchorage	5179
14	Appleton	1073
15	Arcata/Eureka	581
16	Arlington	30022
17	Asheville	923
18	Aspen	2757
19	Atlanta	133911

### 2.3. SQL-запит для знаходження міст з найменшою і найбільшою середніми затримками для БД columnstore\_bts:

```
(
    select arr_delays.city, total_arr_delay + total_dep_delay as total_delay
    from
        (select a.city, sum(f.dep_delay) as total_dep_delay
         from columnstore_bts.flights f
         join columnstore_bts.airports a on f.origin = a.iata_code
         group by a.city) as dep_delays
    join
        (select a.city, sum(f.arr_delay) as total_arr_delay
         from columnstore_bts.flights f
         join columnstore_bts.airports a on f.dest = a.iata_code
         group by a.city) as arr_delays
    on dep_delays.city = arr_delays.city
    order by total_arr_delay + total_dep_delay
    limit 1
)
union
(
    select arr_delays.city, total_arr_delay + total_dep_delay as total_delay
    from
        (select a.city, sum(f.dep_delay) as total_dep_delay
         from columnstore_bts.flights f
         join columnstore_bts.airports a on f.origin = a.iata_code
         group by a.city) as dep_delays
    join
        (select a.city, sum(f.arr_delay) as total_arr_delay
         from columnstore_bts.flights f
         join columnstore_bts.airports a on f.dest = a.iata_code
         group by a.city) as arr_delays
    on dep_delays.city = arr_delays.city
    order by total_arr_delay + total_dep_delay desc
    limit 1
);
```

Час виконання запиту – 0.657 с;

### SQL-запит для знаходження міст з найменшою і найбільшою середніми затримками для БД innodb\_bts:

```
(
    select arr_delays.city, total_arr_delay + total_dep_delay as total_delay
    from
        (select a.city, sum(f.dep_delay) as total_dep_delay
         from innodb_bts.flights f
         join innodb_bts.airports a on f.origin = a.iata_code
         group by a.city) as dep_delays
    join
        (select a.city, sum(f.arr_delay) as total_arr_delay
         from innodb_bts.flights f
         join innodb_bts.airports a on f.dest = a.iata_code
         group by a.city) as arr_delays
    on dep_delays.city = arr_delays.city
    order by total_arr_delay + total_dep_delay
    limit 1
)
union
(
    select arr_delays.city, total_arr_delay + total_dep_delay as total_delay
    from
        (select a.city, sum(f.dep_delay) as total_dep_delay
         from innodb_bts.flights f
         join innodb_bts.airports a on f.origin = a.iata_code
```

```

        group by a.city) as dep_delays
join
(select a.city, sum(f.arr_delay) as total_arr_delay
from innodb_bts.flights f
join innodb_bts.airports a on f.dest = a.iata_code
group by a.city) as arr_delays
on dep_delays.city = arr_delays.city
order by total_arr_delay + total_dep_delay desc
limit 1
);

```

);

Час виконання запиту – 11.075 с;

Результат виконання SQL-запитів для знаходження міст з найменшою і найбільшою середніми затримками:

	city	total_delay
1	Moab	-2485.00
2	Chicago	1990719.00

2.4. SQL-запит для знаходження всіх польотів з затримкою, більшою за середню, для БД columnstore\_bts:

```

select f.fl_num, (f.arr_delay + f.dep_delay) as total_delay
from columnstore_bts.flights f
where (f.arr_delay + f.dep_delay) > (select avg(arr_delay + dep_delay) from
columnstore_bts.flights);

```

Час виконання запиту – 0.098 с;

SQL-запит для знаходження всіх польотів з затримкою, більшою за середню, для БД innodb\_bts:

```

select f.fl_num, (f.arr_delay + f.dep_delay) as total_delay
from innodb_bts.flights f
where (f.arr_delay + f.dep_delay) > (select avg(arr_delay + dep_delay) from
innodb_bts.flights);

```

Час виконання запиту – 0.732 с;

Результат виконання SQL-запитів для знаходження всіх польотів з затримкою, більшою за середню:

	fl_num	total_delay
1	2601	129.00
2	2609	30.00
3	2611	132.00
4	2614	111.00
5	2618	50.00
6	2623	104.00
7	2625	44.00
8	2625	134.00

### 3. Порівняння розмірів і швидкодії звичайної і стовпчикової БД

#### 3.1. Порівняння розмірів БД

##### 3.1.1. SQL-запит для знаходження розміру стовпчикової БД columnstore\_bts:

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

Результат виконання запиту для знаходження розміру стовпчикової БД columnstore\_bts:

	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	52.23 MB	6.38 MB	58.61 MB

Розмір columnstore\_bts – 60.74 MB.

##### 3.1.2. BASH-запит для знаходження розміру звичайної БД innodb\_bts і результат його виконання:

```
sh-4.4# cd /var/lib/mysql
sh-4.4# du innodb_bts -h
413M    innodb_bts
```

Розмір innodb\_bts – 413 MB.

Відношення розміру innodb\_bts до columnstore\_bts – 6.80.

#### 3.2. Порівняння швидкості виконання запитів звичайною і стовпчиковою БД

Запит	Стовпчикова, с	Звичайна, с	Відношення швидкодії звичайної до стовпчикової
Запит для розрахунку сумарної затримки по містах	0.402	5.555	13.818
Запит для обрахунку кількості польотів по містах	0.320	3.528	11.025
Запит для знаходження міст з найменшою і найбільшою середніми затримками	0.657	11.075	16.857
Запит для знаходження всіх польотів з затримкою, більшою за середню	0.098	0.732	7.470

Отже, при виконанні даної задачі стовпчикова база даних продемонструвала кращі результати як у швидкодії, так і в зайнятому обсязі пам'яті.