**Запит 1. Розрахувати сумарну затримку по містах**

SELECT origin AS city, SUM(arr\_delay + dep\_delay) AS total\_delay

FROM flights

GROUP BY origin

UNION ALL

SELECT dest AS city, SUM(arr\_delay + dep\_delay) AS total\_delay

FROM flights

GROUP BY dest;

**Колумнарна БД**:



**Звичайна**:



**Запит 2. Порахувати кількість польотів по містах**

SELECT origin AS city, COUNT(\*) AS flight\_count

FROM flights

GROUP BY origin

UNION ALL

SELECT dest AS city, COUNT(\*) AS flight\_count

FROM flights

GROUP BY dest;

**Колумнарна БД**:



**Звичайна:**



**Запит 3. Знайти місто з найменшою і найбільшою затримкою**

SELECT

city,

CASE

WHEN avg\_delay IS NULL THEN 'Немає даних'

ELSE avg\_delay

END AS avg\_delay

FROM (

SELECT

city,

AVG(arr\_delay) AS avg\_delay

FROM flights

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

GROUP BY city

ORDER BY avg\_delay DESC

LIMIT 1

) AS min\_delay\_city

UNION ALL

SELECT

city,

CASE

WHEN avg\_delay IS NULL THEN 'Немає даних'

ELSE avg\_delay

END AS avg\_delay

FROM (

SELECT

city,

AVG(arr\_delay) AS avg\_delay

FROM flights

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

GROUP BY city

ORDER BY avg\_delay ASC

LIMIT 1

) AS max\_delay\_city

**Колумнарна БД**:



**Звичайна**:



**Запит 4. Знайти всі польоти з затримкою більше за середній час затримки**

SELECT \*

FROM flights

WHERE (arr\_delay + dep\_delay) > (

SELECT AVG(arr\_delay + dep\_delay)

FROM flights

);

**Колумнарна БД**:



**Звичайна**:



|  |  |  |
| --- | --- | --- |
| Запит № | стовпчикова БД | Звичайна БД |
| 1 | 0.391 | 7.828 |
| 2 | 0,266 | 4,094 |
| 3 | 0,922 | 11,875 |
| 4 | 0,156 | 3,671 |

**Скріпт створення баз\таблиць**

#!/usr/bin/env bash

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

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

6)

  
