Advanced Database Systems (CS60113)

Assignment 5. In-Memory and Column-store Databases (Due Date: **November 7** 2020)

NOTE: This is an individual assignment. Also, please check for Part B added on October 30th. The deadline has accordingly been extended.

Part A

Create two versions of the following four tables in MySQL: One for disk resident database and one for In-memory database. You may name those with a prefix dsk and inm for convenience. You will need to change the table names in the queries (given below) accordingly. For creating a table as In-memory, use the following clause in the crate table script: ENGINE=MEMORY. Note, any data inserted in In-memory tables will go once you restart your machine

storedim (scode integer, sname varchar(30), city varchar(30), state varchar(20), pincode integer)
itemdim (icode integer, iname varchar(30), category char(5), subcategory char(3), price integer)
dateofpurchasedim (dcode integer, purchasedate date, dayoftheweek char(3), month char(3), quarter char(2), year integer)
purchasefact (scode, icode, dcode, qty integer). scode, icode, dcode are foreign keys from the previous three tables

Descriptions of columns and possible values for the relevant ones are given below.

scode, icode, dcode, auto-generated unique serial numbers. You may also generate on your own.

sname, iname: Random strings

city: Mumbai, Chennai, Kolkata, Pune, Bengaluru, Guwahati, Kochi
state: Kerala, Maharashtra, West Bengal, Karnataka, Tamil Nadu, Assam (Ensure right city is included under right state
pincode: 123456, 234567, 345678, 456789, 567890, 678901, 789012, 890123, 901234. No constraint with respect to city and state.

category: GROCR, CONSM, FASHN, MEDCN, VGTBL
subcategory: PRS, OFF, MIL. No relationship with category
price: Integer

purchasedate: Date type column.
dayoftheweek: MON, TUE, WED, THR, FRI, SAT, SUN
month: JAN, FEB, …DEC
quarter: Q1, Q2, Q3, Q4 (Ensure correct matching of months to quarter, i.e., APR, MAY, JUN under Q1, JUL, AUG, SEP under Q2, etc.
year: 2015, 2016, 2017, 2018, 2019, 2020

Write programs in any language to generate appropriate “Insert into” statements for the 4 tables. Note, exactly the same rows should be entered in the two sets of tables – Disk resident and In-memory.

Generate 1000 rows each for storedim, itemdim and dateofpurchasedim.
Generate rows for purchasefact (100, 1,000, 100,000, 1,000,000, 10,000,000 number of rows in steps).

Save the two sets (one for each type of DB) of 4 “insert into” scripts (3 for the four dimension tables + 1 for entering 10,000,000 rows in the fact table). You will need to submit these for the assignment. Also, you will be able to use these scripts if you restart your machine and the In-memory data will be gone.

Execute the following queries:

Q1
Select s.state, s.city, s.pincode, sum(p.qty) from storedim s, purchasefact p where s.scode = p.scode group by s.state, s.city, s.pincode

Q2
Select s.sname, i.iname, d. purchasedate, p.qty from storedim s, itemdim i,dateofpurchasedim d, purchasefact p where s.scode = p.scode and i.icode=p.icode and d.dcode = p.dcode

Q3
Select s.city, i.category, sum(p.qty*i.price) from storedim s, itemdim i, purchasefact p where s.scode = p.scode and i.icode=p.icode group by s.city, i.category

Q4
Select s.city, i.subcategory, d.month, sum(p.qty*i. price) from storedim s, itemdim i, dateofpurchasedim d, purchasefact p where s.scode = p.scode and i.icode=p.icode and d.dcode = p.dcode group by s.city, i.subcategory, d.month

Q5
Select s.state, s.city, s.pincode, i.category, i.subcategory, d.year, d.quarter, d.month,d.dayoftheweek, sum(p.qty*i. price) from storedim s, itemdim i, dateofpurchasedim d, purchasefact p where s.scode = p.scode and i.icode=p.icode and d.dcode = p.dcode group by s.state, s.city, s.pincode, i.category, i.subcategory, d.year, d.quarter, d.month, d.dayoftheweek

Note down the time (in seconds) for executing the queries. Report your results in the following tabular form.

	100 rows	**1,000 rows**	**100,000 rows**	**1,000,000 rows**	**10,000,000 rows**
Q1	T1/T2	T1/T2	T1/T2	T1/T2	T1/T2
Q2	T1/T2	T1/T2	T1/T2	T1/T2	T1/T2
Q3	T1/T2	T1/T2	T1/T2	T1/T2	T1/T2
Q4	T1/T2	T1/T2	T1/T2	T1/T2	T1/T2
Q5	T1/T2	T1/T2	T1/T2	T1/T2	T1/T2

T1: Time for Disk resident database (in seconds)
T2: Time for In-memory database (in seconds)

Submit (i) a zip file containing the eight create table scripts (4 for Disk resident + 4 for In-memory) and eight insert into scripts (4 for Disk resident + 4 for In-memory) and (ii) a pdf file containing the results in the above tabular format. Note that, you do not need to submit the programs for generating the insert into scripts. Note, I should be able to simply execute these queries to recreate your environment and re-run the experiments if required.

PART B

Repeat the tasks given above using (i) MariaDB (in Linux) with ColumnStore as database engine. (ii) MonetDB. Report the results in the same format as mentioned above. You may include all the four results in a single table also.

Note: For MariaDB ColumnStore, you need to use the Linux version. The Windows version does not apparently support ColumnStore. For any issue with the installation, please contact Dbopriyo Banerjee (my senior Ph.D. student and TA for this course at: deb.ban89@gmail.com. He has prepared the instruction set given below.

```
##############################################################################
Installation of MariaDB with ColumnStore in Ubuntu
##############################################################################
Links
https://downloads.mariadb.org/mariadb/repositories/#distro=Ubuntu&distro_release=xenial--
ubuntu_xenial&mirror=harukasan&version=10.5
https://mariadb.com/docs/deploy/community-single-columnstore-cs105-ubuntu16/
##############################################################################
Execute the following commands:
sudo apt-get install software-properties-common gnupg-curl
sudo apt-key adv --fetch-keys 'https://mariadb.org/mariadb_release_signing_key.asc'
sudo add-apt-repository 'deb [arch=amd64,arm64,i386,ppc64el]
https://ftp.harukasan.org/mariadb/repo/10.5/ubuntu xenial main'

sudo apt update
sudo apt install mariadb-server mariadb-plugin-columnstore
sudo apt install libjemalloc1
##############################################################################
After installation check mysql status

service mysql status

if it is running, then restart the service

service mysql restart

else

service mysql start
##############################################################################
Now create a table with engine as columnstore:

MariaDB [debopriyo]> create table users (userId Integer, userName varchar(10))
ENGINE=columnstore;
Query OK, 0 rows affected (6.608 sec)

Drop Table:

MariaDB [debopriyo]> DROP TABLE users;
Query OK, 0 rows affected (3.537 sec)

##############################################################################
Create table with engine as innodb:

MariaDB [debopriyo]> create table users (userId Integer, userName varchar(10))
ENGINE=innodb;
Query OK, 0 rows affected (0.151 sec)

Drop Table:

MariaDB [debopriyo]> drop table users;
Query OK, 0 rows affected (0.079 sec)
##############################################################################
```

For MonetDB, installation is straightforward. Instructions for executing MonetDB was explained during the lecture on October 29th.