Code Test - Data Operations

Table of Contents

SQL

```
You have a database with the following tables:

 - tblDimDate             - a table of dates, i.e., a calendar
 - tblOrder               - a table of 'Orders', also referred to as Campaigns
 - tblAdvertiserLineItem  - a table of 'Advertiser Line Items' (ALI for short).

Each ALI is a component of a campaign.
Therefore, the relation of tblAdvertiserLineItem to tblOrder is many-to-one, with the foriegn key relationship described below.

Use the sample data and schema descriptions below to provide the following queries:
```

QUERY_1)
Write an SQL query to return all months in the current year for which there are exactly 30 days.

QUERY_2)
tblDimDate should have one row (one date) for every date between the first date and the last date in the table. Write a SQL query to determine how many dates are missing, if any, between the first date and last date. You do not need to supply a list of the missing dates.

QUERY_3)
Write an SQL query to identify all orders scheduled to run in November 2021, for which there are not yet any records in tblAdvertiserLineItem.

QUERY_4)
Write an SQL query to count total number of campaigns in tblOrder grouped by campaign duration.
Campaign duration would be the number of days between dateStart and dateEnd.

QUESTION_1)
Database design:
What are the advantages and disadvantages of creating and using normalized tables?
What are the advantages and disadvantages of creating and using non-normalized tables?

```
# SAMPLE DATA

select * from tblDimDate limit 1\G
```

```
*************************** 1. row ***************************
       dateDay: 2008-01-01
        iMonth: 01
        sMonth: January
         iYear: 2008
      iYearDay: 1
     iMonthDay: 01
      iWeekDay: 02
      sWeekDay: Tuesday
    iYearMonth: 200801
    iUSHoliday: 1
         iWeek: 0
sUSHolidayName: New Years Day

select * from tblOrder limit 1\G
*************************** 1. row ***************************
                    id: 12
          idAdvertiser: 12
               sStatus: ACTIVE
             dateStart: 2009-01-06 00:00:00
               dateEnd: 2009-02-28 00:00:00
            dateCreate: 2009-01-06 16:13:09
           dateUpdated: 2019-01-09 18:30:11

select id, idOrder, sstatus, datestart, dateend, sratetype, fclientrate, datecreate, dateupdated, fbudget from tblAdvertiserLineItem order by id desc limit 1\G
*************************** 1. row ***************************
         id: 248049
    idOrder: 112011
    sstatus: PRELAUNCH
  datestart: 2021-10-01 00:00:00
    dateend: 2021-11-30 23:59:59
  sratetype: CPM
fclientrate: 4.2
 datecreate: 2021-09-29 00:05:06
dateupdated: 2021-09-29 00:20:44
    fbudget: 5009.5
1 row in set (0.00 sec)

# SCHEMA DESCRIPTIONS

desc tblDimDate ;
+----------------+-------------------------------+------+-----+------------+-------+
| Field          | Type                          | Null | Key | Default    | Extra |
+----------------+-------------------------------+------+-----+------------+-------+
| dateDay        | date                          | NO   | PRI | 0000-00-00 |       |
| iMonth         | smallint(2) unsigned zerofill | YES  |     | NULL       |       |
| sMonth         | varchar(20)                   | NO   |     | NULL       |       |
| iYear          | smallint(5) unsigned          | NO   |     | NULL       |       |
| iYearDay       | smallint(6) unsigned          | YES  |     | NULL       |       |
| iMonthDay      | smallint(2) unsigned zerofill | YES  |     | NULL       |       |
| iWeekDay       | smallint(2) unsigned zerofill | YES  |     | NULL       |       |
| sWeekDay       | char(10)                      | YES  |     | NULL       |       |
| iYearMonth     | int(10) unsigned              | NO   |     | NULL       |       |
| iUSHoliday     | tinyint(4)                    | YES  |     | NULL       |       |
| iWeek          | smallint(5) unsigned          | YES  |     | NULL       |       |
| sUSHolidayName | varchar(60)                   | YES  |     | NULL       |       |
+----------------+-------------------------------+------+-----+------------+-------+

desc tblOrder ;
+-------------------------+---------------------------------------------+------+-----+-------------------+----------------+
| Field                   | Type                                        | Null | Key | Default           | Extra          |
+-------------------------+---------------------------------------------+------+-----+-------------------+----------------+
| id                      | int(10) unsigned                            | NO   | PRI | NULL              | auto_increment |
| idAdvertiser            | int(10) unsigned                            | NO   | MUL | NULL              |                |
| sStatus                 | enum('ACTIVE','INACTIVE','PROPOSED','ENDED') | YES  |     | NULL              |                |
| dateStart               | datetime                                    | YES  |     | NULL              |                |
| dateEnd                 | datetime                                    | YES  |     | NULL              |                |
| dateCreate              | datetime                                    | NO   |     | NULL              |                |
| dateUpdated             | timestamp                                   | NO   | MUL | CURRENT_TIMESTAMP |                |
+-------------------------+---------------------------------------------+------+-----+-------------------+----------------+

desc tblAdvertiserLineItem ;
+---------------------------+---------------------+------+-----+-------------------+-----------------------------+
| Field                     | Type                | Null | Key | Default           | Extra                       |
+---------------------------+---------------------+------+-----+-------------------+-----------------------------+
| id                        | int(10) unsigned    | NO   | PRI | NULL              | auto_increment              |
| idOrder                   | int(10) unsigned    | NO   | MUL | NULL              |                             | <== this is a foreign key to tblOrder.id
| sStatus                   | varchar(45)         | NO   |     | ACTIVE            |                             |
| dateStart                 | datetime            | YES  |     | NULL              |                             |
| dateEnd                   | datetime            | YES  |     | NULL              |                             |
| sRateType                 | varchar(255)        | YES  |     | NULL              |                             |
| fClientRate               | float               | YES  |     | NULL              |                             |
| dateCreate                | datetime            | NO   |     | NULL              |                             |
| dateUpdated               | timestamp           | NO   | MUL | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| fBudget                   | double              | YES  |     | NULL              |                             |
+---------------------------+---------------------+------+-----+-------------------+-----------------------------+
```

Hive, HDFS

```
QUESTION_1) Given a table in Hive, how do we identify where the data is stored?
QUESTION_2) How can we see what partitions the table may have?
QUESTION_3) How can we see if the table is missing any partitions?

Given a table, 'auctions' with the following DDL:

      CREATE TABLE auctions (
            auctionid string COMMENT 'the unique identifier for the auction',
            idlineitem bigint COMMENT 'the Line Item ID associated with the auction',
            arysegments array<string> COMMENT 'the array of segments associated with the auction'
      )
      PARTITIONED BY (utc_date string)
      ROW FORMAT DELIMITED
      FIELDS TERMINATED BY '\t'
      STORED AS TEXTFILE
      LOCATION '/bidder_data/auctions' ;

      What would be the queries to answer the following questions:

      QUERY_1) Provide an HQL query to provide a distribution of the number of auctions and line items, grouped by the number of segments within each auction record.
      QUERY_2) Provide an HQL query to provide the distinct count of auctions and line items, associated to each segment within arysegments. (HINT: You will need to expand the
      segment array first.)
```

Linux, Bash

You need to run a hive query for a sequence of 30 dates (all dates in September 2021)..
The size of the table data and cluster bandwidth require that the query be run for each date individually.
The query string will accept 'DATE' as an argument, as follows:

```
HQL_QUERY="select utc_date, sum(1) as num_rows from my_table where utc_date = '${DATE}' group by utc_date"
```

Using bash, write a script which will execute this query for all dates in September 2021 and store the result to a single file.
The file will have 2 columns and up to 30 rows.

Python

You need to run a hive query for a sequence of 30 dates (all dates in September 2021)..
The size of the table data and cluster bandwidth require that the query be run for each date individually.
The hive query will accept 'DATE' as an argument, as follows.

```
HQL_QUERY = "select utc_date, sum(1) as num_rows from my_table where utc_date = '${DATE}' group by utc_date"
```

Using Python, write a script which will execute this query for all dates in September 2021 and store the result to a single file.
The file will have 2 columns and up to 30 rows.