**Assignment 8 - Data Warehousing & Big Data Pipeline (Batch 5)**

Started: Sep 8 at 9:08pm

**Quiz Instructions**

This Assignment is based on the concepts that were discussed in the following modules:

* Introduction to Data Warehousing
* Large Scale Data Ingestion Using Sqoop

2. Please make sure that you are using the latest version of Google Chrome.

3. It's mandatory to disable all the browser extensions & enable Add-ons and then open the test.

4. If a test mandates you to use the webcam, we recommend you to give the required permissions and access.

5. Don't navigate to another page while taking this assignment. In case, if you want to access the lab, you can open it in a new tab.

6. By mistake, if you have navigated to a different page while taking the assignment, you can use the back button to return to the assignment.

7. Always keep a local copy of the programming questions and submit all the questions at once after solving all the questions.

8. To know the test results or figure out the next course of action, please contact your test administrator and they will guide you.

Best wishes from Edureka!

Top of Form

[Flag this Question](https://claas-canvas.corestack.io/courses/5/quizzes/335/take)

**Question 115 pts**

**Abstract:**

Every organization implements CRM as a strategy that integrates concepts of data mining and data warehousing in order to support an organization’s decision-making process to retain long-term and profitable relationships with its customers.

Retail Bazaar Inc. is a leading retail company in India with more than 5000 retail stores and hypermarkets across the country.

The company has a dedicated team to handle after-sales customer care services, which register customer complaints and resolves them in order to increase customer retention, loyalty and decrease customer attrition.

**Following is an abstract of end to end process:**

* Company has multiple call centers across India to handle the customer requests.
* Customers can reach to the customer care team over different channels i.e. via call or chat.
* CCR (customer care representative) will register the complaint with all the provided details.
* A case can have a status such as open, closed and reopened.
* A case can have states such as Escalated, Withdrawn, Abandoned, etc.
* Each case can be given a priority score and depending on the score, CCR has an SLA to close the case in a specific amount of hours/days.
* Once the case is closed, the customer receives a survey link to rate the overall process.
* Customer can take a survey or leave it unattended.
* Customer can score the survey from 1-10 on various questions about his experience regarding customer care process.
* The survey response is captured for that particular case.
* Some of the other dimensions of the CRM process are case category, product, language, channel, market, supplier etc.
* A case can belong to any of the categories such as ‘subscription, ‘Purchase, ‘shipment’ etc.

This complete CRM process generates data and stores it across multiple OLTP tables.

**The company wants to do some analysis around this data and capture the following KPIs to further enhance and optimize the CRM process.**

**As a Big Data Engineer, your task is to understand the KPIs, and you need to think of possible facts and dimensions to build a data warehouse that can help in deriving these KPIs.**

**KPIs:**

**Case KPIs:**

What is the total number of open/closed/reopened cases in a day/week/month and should be further sliced and diced by employee, product, priority, region, category and call center dimensions?

**Chat KPIs:**

If the case was handled through a chat by the representative, then, following are the KPIs for each employee:

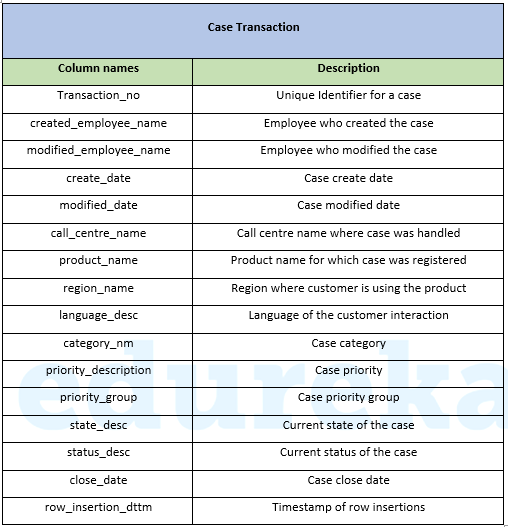
* The total number of chats handled by customer care representative in a day/week/month and should be further sliced and diced by product, priority, region, category and call center dimensions.
* Total chat time of a CCR for the above-mentioned dimensions.
* The total time when the representative was online but not chatting.
* Total wait time for the representative. Wait time can be calculated as the time when the customer sent first message minus time when representative sent the first response.
* The total time when the representative was offline.

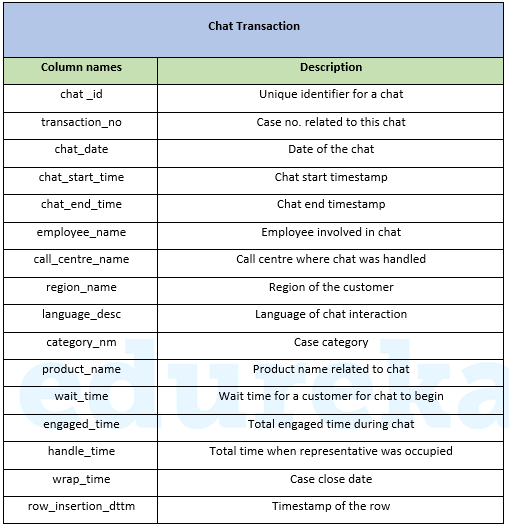
**Survey KPIs:**

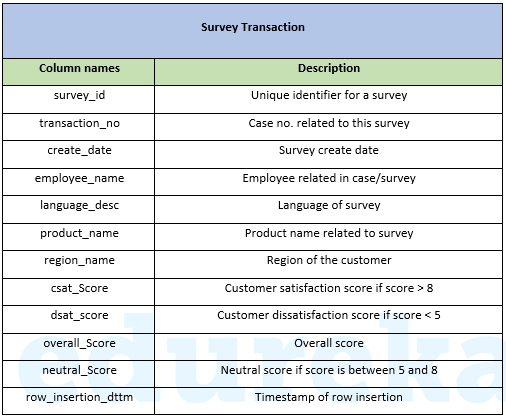
For surveys, if any question had a 1-4 score from the customer then it is a negative response, 5-8 score is neutral and 9-10 is a positive response.

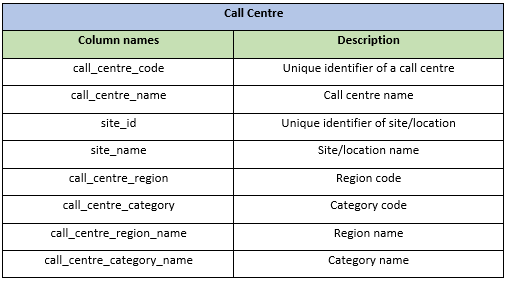
* The total number of positive/negative/neutral responses in a day/week/month and further sliced and diced by employee, product, priority, region, category and call center dimensions.

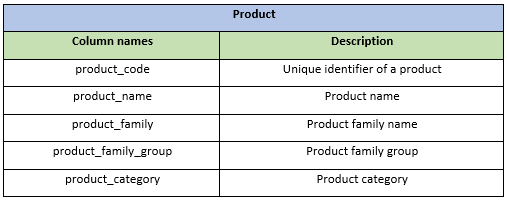
**OLTP Tables:**

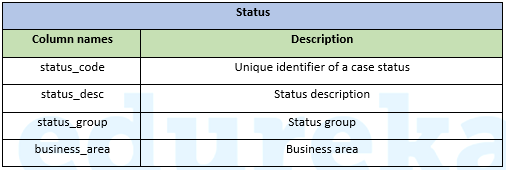


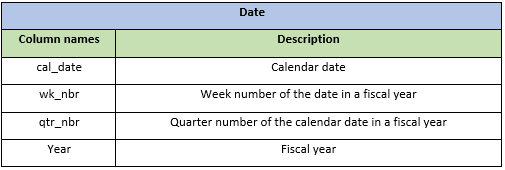


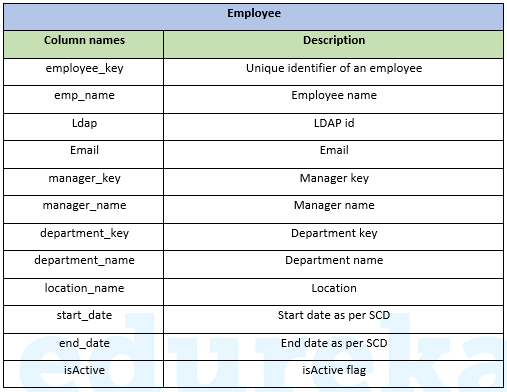


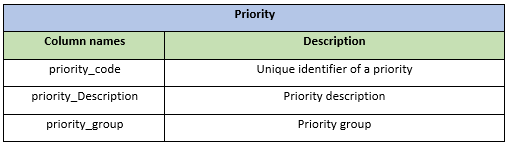


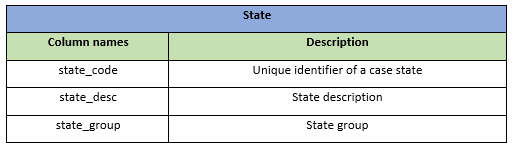






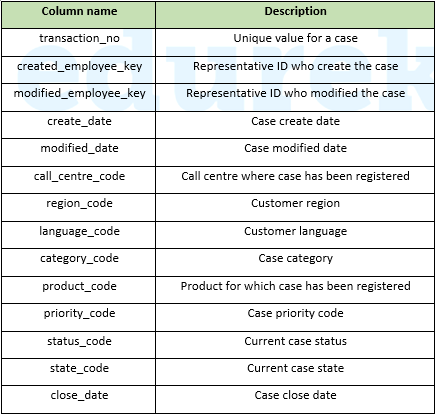






**Create an ER diagram to represent the facts & dimensions. Also, share your views for selecting a particular schema.**

**Hint:** A possible record in ‘case’ fact table could be:



**Output Format:**Create a folder structure 'Assignment8/DW\_Problem\_1' on your cloud lab and place the image of your ER diagram to the folder you created.

Mention file name and the path where you have uploaded the ER diagram image along with all your findings.

[HTML EditorKeyboard Shortcuts](https://claas-canvas.corestack.io/courses/5/quizzes/335/take)

12pt

Paragraph

0 words

[Flag this Question](https://claas-canvas.corestack.io/courses/5/quizzes/335/take)

**Question 215 pts**

As a Big Data Engineer, you need to build a Data Mart to find stores & different products, which are going out of stock on a day due to a smaller number of in-stock items and a greater number of sales.

**Hint:** if *fact\_instk\_dly.onhand\_each\_qty* < *total (fact\_scanv\_dly.sales\_unit\_qty)*for an item in a store on a day then item got out of stock in that store on that day.

**Pre-requisites:**

All the target tables in Hive must be partitioned on some columns with the ORC storage format. All the tables should have an additional *row\_insertion\_dttm* column which will store the timestamp of a record when it was loaded into the table. You can create stage tables in the hive if required.

Following are the details of dimensions & facts tables:

**Dimensions:**

* dim\_store (Table for Store-related information)
* dim\_region (Table for Geo-related Information)
* dim\_item (Table for Items-related Information)
* dim\_calendar (Date Dimension)

**Facts:**

* fact\_instk\_dly (Item in stocks related information at date level)

The information which we can derive from this table:

What is the onhand\_each\_qty (which is relatable to ‘in stocks’ in general terms) of an item(mds\_fam\_id) in a store (store\_nbr) of a geo (geo\_region\_cd) on a given date (bus\_dt).

* fact\_scanv\_dly (Items getting scanned and sold every visit day of customers)

The information which we can derive from this table:

What is the total sales amount, sales unit, return amount on a given customer visit dt

 This table is a sales transaction table and not an aggregated table. The table will have details for each sale transaction done from store POS.

**Basic EDA Details:**

* If any table has *geo\_region\_cd* column, then *geo\_region\_cd = ‘US’* are eligible records to work with. Filter out all other records. Dimension might have values for countries other than ‘US’ with column values in different languages that are not eligible for our use case.
* If any table has *op\_cmpny\_cd* column, *then op\_cmpny\_cd = ‘WMT-US’* are our eligible records to work with. Filter out all other records.
* If any table has both the columns, then we can filter on any of them with the values mentioned above.
* If Metric columns (e.g. *onhand\_each\_qty*, *sales\_unit\_qty*) have negative values, then we can ignore them as bad/corrupt data.
* Below is the list of columns in each table which is important for our use case. Other columns are either audit columns or columns which are not of our concern.

dim\_region:

                region\_nbr

                region\_nm

                region\_mgr\_nm

                geo\_region\_cd

                op\_cmpny\_cd

dim\_store:

                store\_nbr

                geo\_region\_cd

                store\_nm

                region\_nbr

                market\_nm

                 city\_nm

                 op\_cmpny\_cd

                dim\_item:

                                mds\_fam\_id

                                item\_nbr

                                item\_desc\_1

                                upc\_nbr

                                dept\_desc

                                brand\_nm

                                geo\_region\_cd

                dim\_calendar:

                                wm\_yr\_wk\_nbr

                                wm\_qtr\_nm

                                cal\_dt

                                geo\_region\_cd

                fact\_instk\_dly:

                                store\_nbr

                                geo\_region\_cd

                                mds\_fam\_id

                                onhand\_each\_qty

                                fcst\_dmand\_each\_qty

                                bus\_dt

                                op\_cmpny\_cd

                fact\_scanv\_dly:

                                store\_nbr

                                geo\_region\_cd

                                mds\_fam\_id

                                upc\_nbr

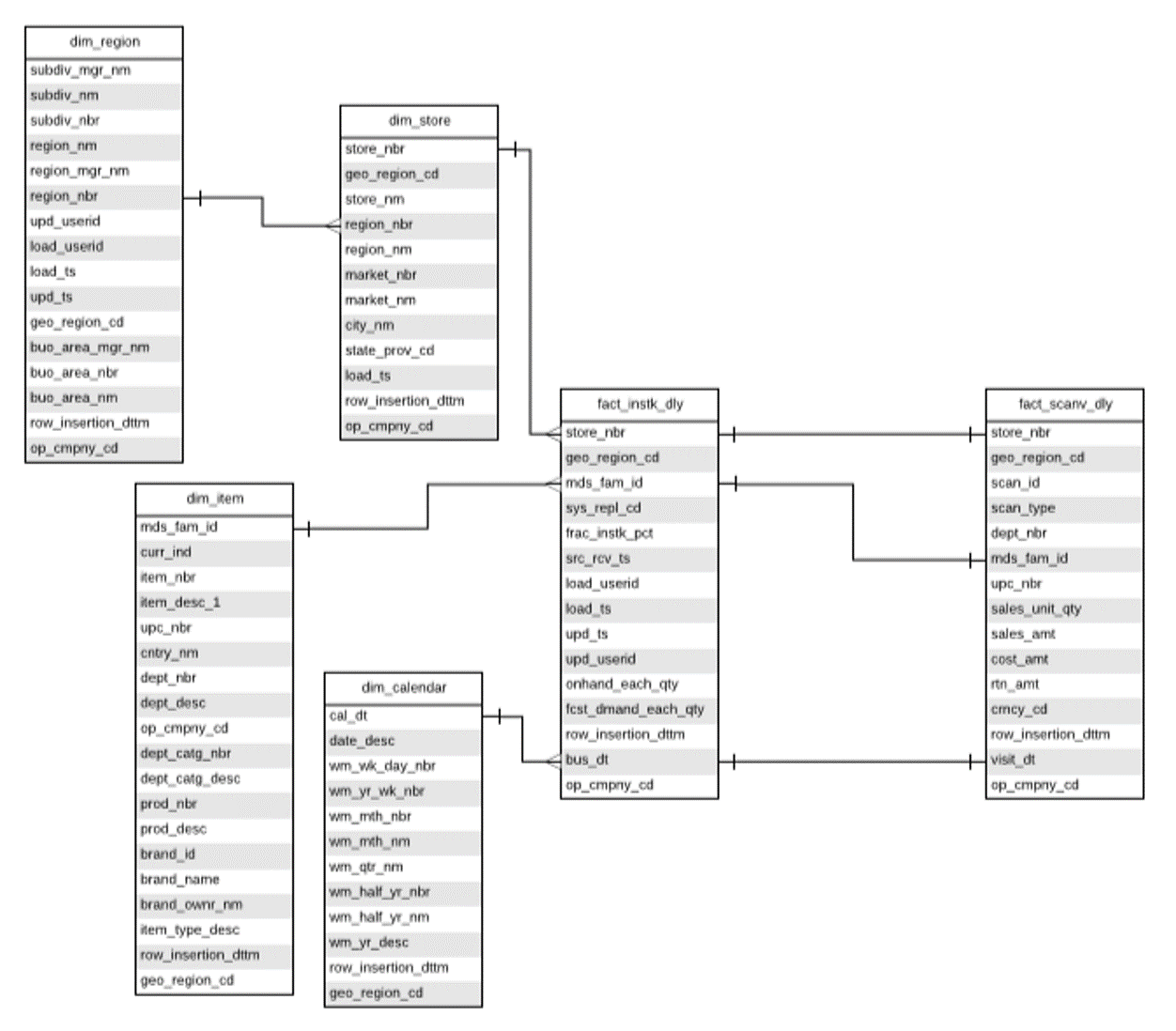
                                sales\_unit\_qty

                                sales\_amt

                                rtn\_amt

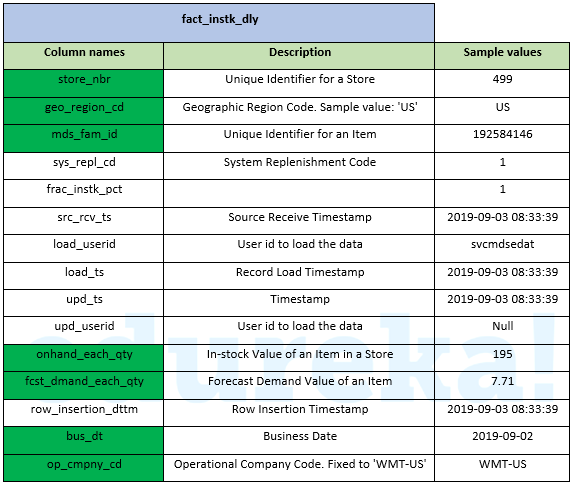
                                visit\_dt

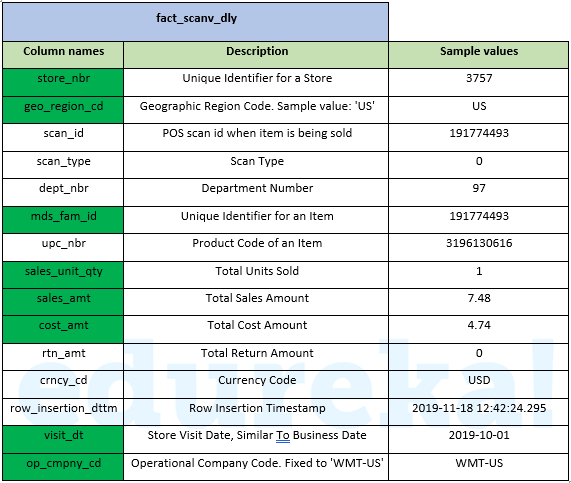
                                op\_cmpny\_cd

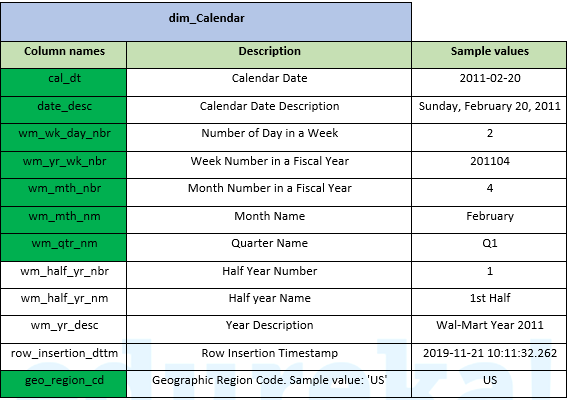


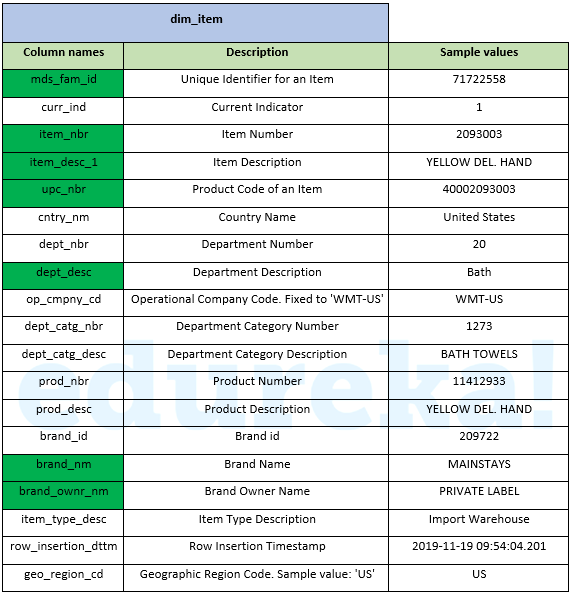
**Data Warehouse**

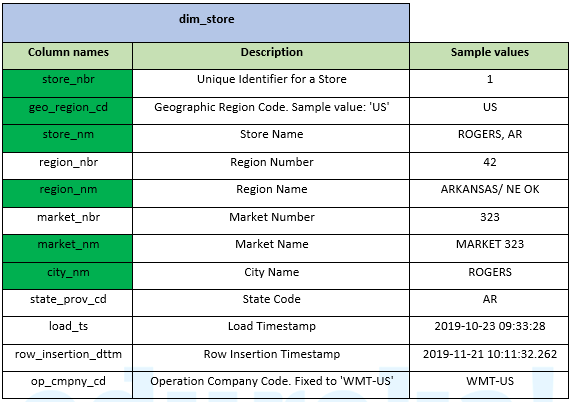
\* Important columns are highlighted in green which are required to solve the problem

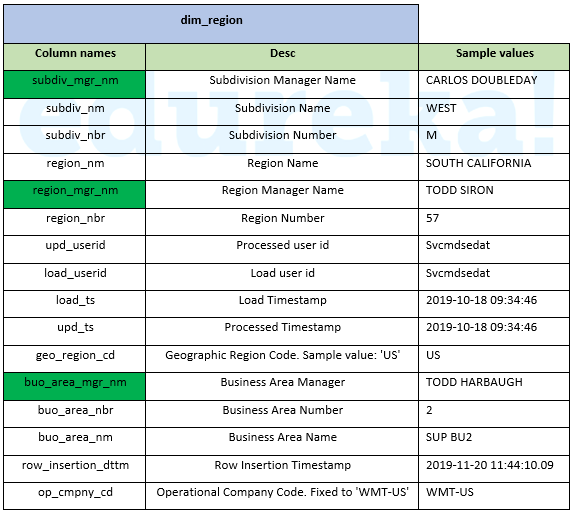






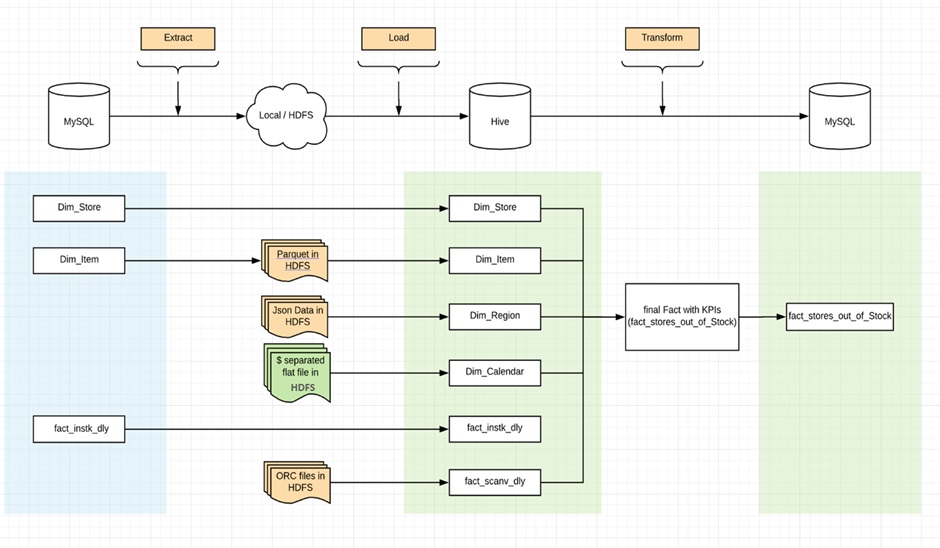




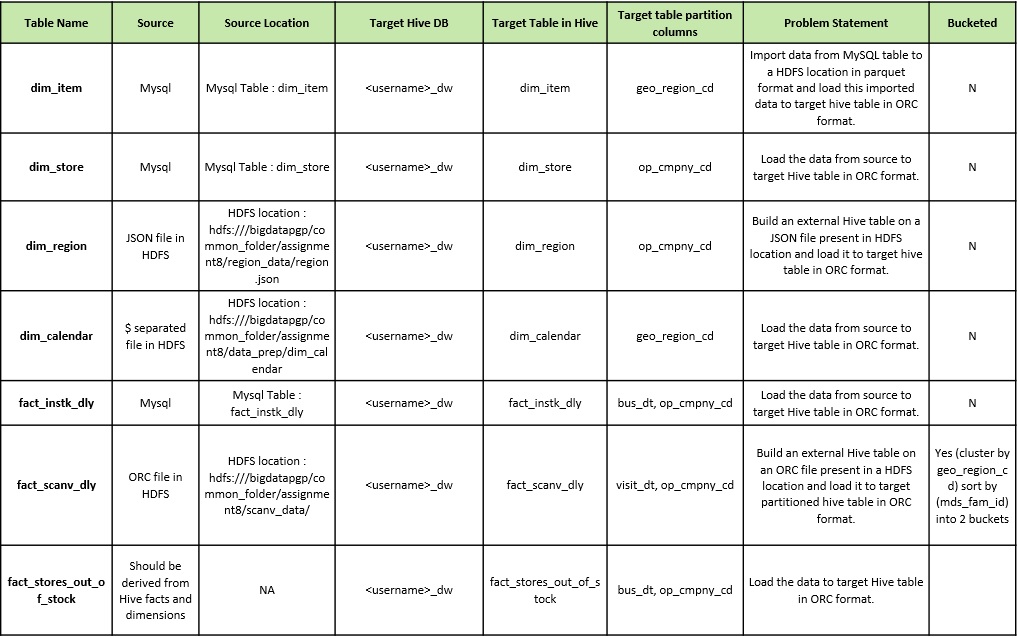


Following is the ETL flow of data:

**ETL Flow**



**Task Details**



**Output Format:**You need to paste the stepwise commands and the code used to solve this problem.

[HTML EditorKeyboard Shortcuts](https://claas-canvas.corestack.io/courses/5/quizzes/335/take)

12pt

Paragraph

0 words

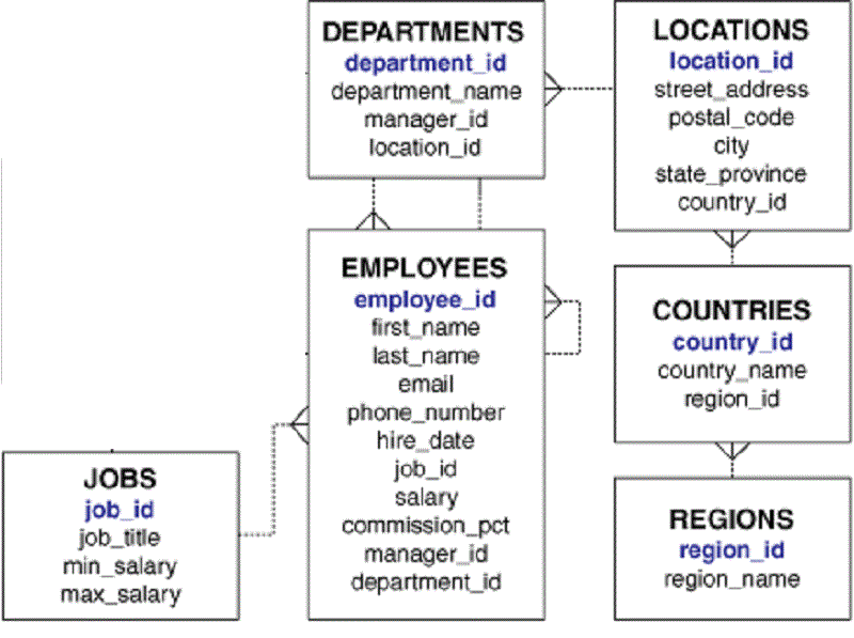
[Flag this Question](https://claas-canvas.corestack.io/courses/5/quizzes/335/take)

**Question 320 pts**

Consider the following employee-related tables in MySQL.

**Tables:**

* employees\_table (contains employee details such as id, name, email, phone, salary, department etc.)
* departments\_table (contains department details such as department id, name, location etc.)
* locations\_table (contains location information such as street name, city, pin, state, country)
* jobs\_table (contains job metadata such as job id, job name etc.)
* regions\_table (contains region details such as region id, region name)
* countries\_table (contains county details such as country id, country name)



As a Big Data Engineer, you have to perform the following tasks:

* Import all these MySQL tables to Hive tables. Hive DDLs can be inferred from table schemas in MySQL.
* Create a new table (i.e. employee\_scd) which implements a slowly changing dimension (type 2) for the employees table.
* SCD table should have additional columns, start\_date, end\_date, and active\_flag.
* Start\_date and end\_date behave as per the concept of SCD type 2. The active flag can be 0 or 1 depending on which row in the table contains the latest information of the employee.
* There is a delta file present at the HDFS path '**hdfs:///bigdatapgp/common\_folder/assignment8/emp\_details/Employeedelta.txt**' that contains the changes in employee information till the current date.
* Based on changes present in the delta file (on columns such as email, phone\_number, job\_id, department\_id, manager\_id, salary, commission\_pct) maintain the change history using start\_date, end\_date and active flag columns with old information being closed with end\_date = (current\_date -1).
* Create an updated employee table with the latest information on all the employees.
* Upload the updated employee details to a MySQL table.

**SCD Type 2 Example:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **Name** | **dept\_id** | **manager\_id** | **start\_date** | **end\_date** | **active\_flag** |
| 1 | Abc | 10 | 111 | 2019-01-01 | 9999-12-31 | 1 |
| 2 | Def | 10 | 111 | 2019-01-01 | 2019-12-08 | 0 |
| 2 | Def | 20 | 111 | 2019-12-09 | 9999-12-31 | 1 |

**For employee 2, the department got changed on 2019-12-09 from 10 to 20, hence in SCD type 2, there will be 2 records for employee id 2.**

**The schema of the output table is mentioned below. The SCD table needs to be present in the Hive whereas the updated employee table needs to be present in MySQL.**

**Output the result as per the schema mentioned in the below tables.**

1. **Employee\_scd Output Table in Hive:**

CREATE TABLE edureka\_dw.employees\_scd

   ( employee\_id int

   , first\_name VARCHAR(20)

   , last\_name VARCHAR(25)

   , email VARCHAR(25)

   , phone\_number VARCHAR(20)

   , hire\_date varchar(12)

   , job\_id VARCHAR(10)

   , salary decimal

   , commission\_pct decimal(2,2)

   , manager\_id int

   , department\_id int

   , start\_date string

   , end\_date string

   , active\_flag char(1)

   , row\_insertion\_dttm string

   ) ;

1. **Updated Employee Detail Table in MySQL:**

CREATE table edureka\_dw.employee\_details\_latest

   (employee\_id int,

   job\_id varchar(10),

   manager\_id int,

   department\_id int,

   location\_id int,

   country\_id char(2),

   first\_name varchar(20),

   last\_name varchar(25),

   salary decimal,

   commission\_pct decimal(2,2),

   department\_name varchar(30),

   job\_title varchar(35),

   city varchar(30),

   state\_province varchar(25),

   country\_name varchar(40),

   region\_name varchar(25)

   );

**Output Format:**You need to paste the commands and the code used to solve this problem.

[HTML EditorKeyboard Shortcuts](https://claas-canvas.corestack.io/courses/5/quizzes/335/take)

12pt

Paragraph

Bottom of Form