Hadoop and Spark Developer - CCA 175

Problem Scenario 6 [Data Analysis]

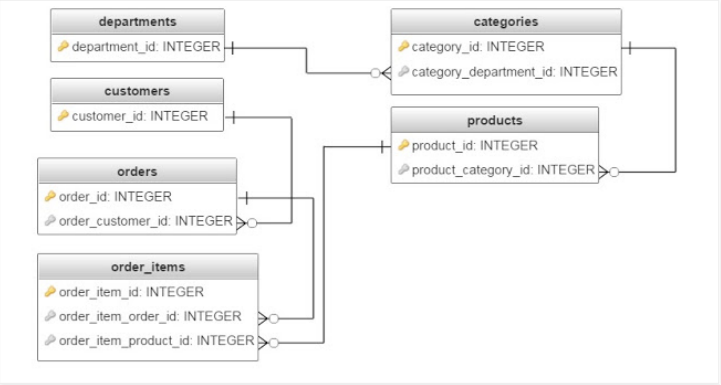
PLEASE READ THE INTRODUCTION TO THIS SERIES. CLICK ON HOME LINK AND READ THE INTRO BEFORE ATTEMPTING TO SOLVE THE PROBLEMS

Video walkthrough of this problem is available at ***[CLICK HERE]*** AND

Click here for the video version of this series. This takes you to the youtube playlist of videos.

This problem helps you strengthen and validate skills related to ***data analysis*** objective of the certification exam.

Data model in mysql on cloudera VM looks like this. [*Note: only primary and foreign keys are included in the relational schema diagram shown below*]



Problem 6: Provide two solutions for steps 2 to 7

- Using HIVE QL over Hive Context
- Using Spark SQL over Spark SQL Context or by using RDDs

1. create a hive meta store database named **problem6** and import all tables from mysql retail_db database into hive meta store.
2. On spark shell use data available on meta store as source and perform step 3,4,5 and 6. *[this proves your ability to use meta store as a source]*
3. Rank products within department by price and order by department ascending and rank descending *[this proves you can produce ranked and sorted data on joined data sets]*
4. find top 10 customers with most unique product purchases. if more than one customer has the same number of product purchases then the customer with the lowest customer_id will take precedence *[this proves you can produce aggregate statistics on joined datasets]*
5. On dataset from step 3, apply filter such that only products less than 100 are extracted *[this proves you can use subqueries and also filter data]*
6. On dataset from step 4, extract details of products purchased by top 10 customers which are priced at less than 100 USD per unit *[this proves you can use subqueries and also filter data]*
7. Store the result of 5 and 6 in new meta store tables within hive. *[this proves your ability to use metastore as a sink]*

Solution:

Try your best to solve the above scenario without going through the solution below. If you could then use the solution to compare your result. If you could not then I strongly recommend that you go through the concepts again (this time in more depth). Each step below provides a solution to the points mentioned in the Problem Scenario. Please go through the video for an indepth explanation of the solution.

NOTE: The same solution can be implemented using Spark SQL Context. Just replace Hive Context object with SQL Context object below. Rest of the solution remains the same. i.e same concept of querying, using temp table and storing the result back to hive.

Step 1:

```
sqoop import-all-tables --connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" --username retail_dba --password cloudera --warehouse-dir /user/hive/warehouse/problem6.db --hive-import --hive-database problem6 --create-hive-table --as-textfile;
```

Step 2:

```
var hc = new org.apache.spark.sql.hive.HiveContext(sc);
```

Step 3:

```
var hiveResult = hc.sql("select  d.department_id,  p.product_id,  p.product_name,  p.product_price,  rank() over (partition by d.department_id order by p.product_price) as product_price_rank,  dense_rank() over (partition by d.department_id order by p.product_price) as product_dense_price_rank  from products p  inner join categories c on c.category_id = p.product_category_id  inner join departments d on c.category_department_id = d.department_id  order by d.department_id, product_price_rank desc, product_dense_price_rank ");
```

Step 4:

```
var hiveResult2 = hc.sql("select c.customer_id, c.customer_fname, count(distinct(oi.order_item_product_id)) unique_products  from customers c  inner join orders o on o.order_customer_id = c.customer_id  inner join order_items oi on o.order_id = oi.order_item_order_id  group by c.customer_id, c.customer_fname  order by unique_products desc, c.customer_id  limit 10
")
```

Step 5:

```
hiveResult.registerTempTable("product_rank_result_temp");
hc.sql("select * from product_rank_result_temp where product_price < 100").show();
```

Step 6:

```
 var topCustomers = hc.sql("select c.customer_id, c.customer_fname, count(distinct(oi.order_item_product_id)) unique_products  from customers c  inner join orders o on o.order_customer_id = c.customer_id  inner join order_items oi on o.order_id = oi.order_item_order_id  group by c.customer_id, c.customer_fname  order by unique_products desc, c.customer_id  limit 10  ");

topCustomers.registerTempTable("top_cust");

var topProducts = hc.sql("select distinct p.* from products p inner join order_items oi on oi.order_item_product_id = p.product_id inner join orders o on o.order_id = oi.order_item_order_id inner join top_cust tc on o.order_customer_id = tc.customer_id where p.product_price <
```

100");

Step 7:

```
hc.sql("create table problem6.product_rank_result as select * from product_rank_result_temp where product_price < 100");


hc.sql("create table problem 6.top_products as select distinct p.* from products p inner join order_items oi on oi.order_item_product_id = p.product_id inner join orders o on o.order_id = oi.order_item_order_id inner join top_cust tc on o.order_customer_id = tc.customer_id where p.product_price < 100");
```

13 comments:

amitrock May 22, 2017 at 1:36 AM

Hi Arun , This Blog is being so awesome for me !!! When can we expect the next videos which will be related to Flume / Kafka / Spark Streaming and Configuration ??? Looking forward to learn more from you.

Reply

Replies

Arun Kumar Pasuparthi May 28, 2017 at 4:44 PM

Flume problem is out already. I dont expect anything asked in Kafka. Spark Streaming is a grey area as no one reported seeing a problem in spark streaming so far. I will post a problem on spark streaming shorting anyway.

Reply

Unknown July 10, 2017 at 11:51 AM

This comment has been removed by the author.

Reply

taoufik elk July 14, 2017 at 4:41 AM

hi Arun,

do you really need the --create-hive-table in the first question ?

Reply

ela hayder July 17, 2017 at 8:26 PM

Hi Arun,

I am getting error- "org.apache.spark.sql.AnalysisException: Table not found: products;" on while running the query in spark sql. I followed all the steps correctly till that point.

Reply

hammad zahid July 23, 2017 at 7:09 AM

Hi.arun first of all great effort? As I am pretty weak in hive but good in spark so I am confused whether what is replacement for dense rank in spark SQL.

Reply

Replies

hammad zahid July 23, 2017 at 7:11 AM

As part 3 or step 3 is pretty confusing to me.

Reply

Prateck Jain July 25, 2017 at 12:27 PM

Hi Arun,
Great Blog for preparing for CCA175.
I would like to know , that a single question in the certification contains the number of steps or queries like above? or they are fewer?? I am asking keeping in mind the duration of the exam. 2 hrs for 10 questions , gives us 12 mins per question. Would we be able to solve all the steps in this amount of time?

Reply

Replies

Arun Kumar Pasuparthi July 25, 2017 at 2:41 PM

you can safely consider each step as a single question in the exam. The exam questions should not be time consuming if you understand that approach to take.

Reply

subhashini balu October 5, 2017 at 7:46 AM

Such an useful blog!!
I have a query.Can i give answers to all the spark related questions in hive context/sql context?

Reply

Lakshmi Thiagarajan October 5, 2017 at 11:41 AM

This is more of a sql or hive QL question , In this query below for 'Rank products within department by price and order by department ascending and rank descending' - is there a way to list only the top 3 ranked products within each department ??

```
select d.department_id, p.product_id, p.product_name, p.product_price, rank() over (partition by d.department_id order by p.product_price) as product_price_rank, dense_rank() over (partition by d.department_id order by p.product_price) as product_dense_price_rank from products p inner join categories c on c.category_id = p.product_category_id inner join departments d on c.category_department_id = d.department_id order by d.department_id, product_price_rank desc, product_dense_price_rank
```

Reply

Replies

Lakshmi Thiagarajan October 5, 2017 at 12:07 PM

Never mind , I got it.

select * from (select product_id ,product_price , category_id , category_name , rank() over (partition by category_id order by product_price) price_rank , dense_rank() over (partition by category_id order by product_price) price_dense_rank from products join categories on product_category_id = category_id order by category_id,price_rank, price_dense_rank) tmp where price_dense_rank <= 3;

Thanks though
lakshmi

Reply

Ishtiaq Ahmad December 28, 2017 at 10:20 AM

HI Arun
great post, helped me a lot. One question i have is is there no other way to push data in hive without going through hiveContext?

I have tried DF.write.mode("append").saveAsTable("schema.table") and it works but still wondering if there is another better way.

Reply

Enter your comment...

Comment as: Select profile...

Publish Preview

Home

Subscribe to: Posts (Atom)

If you have landed on this page then you are most likely aspiring to learn Hadoop ecosystem of technologies and tools. Why not make you...

(no title)
If you have landed on this page then you are most likely aspiring to learn Hadoop ecosystem of technologies and tools. Why not make you...

Simple theme. Powered by Blogger.