Hive Usecases 2020

Lets learn about schema migration, Csv serde, json, fixed width , sqoop export/import with more options, if we know how to work on these we can boldly say I am one among very few know these advance concepts ☺

1. Create a usecase dir

mkdir /home/hduser/hiveusecase

Download from the link "inceptez.in -> more -> HIVE REALTIME USECASES FOR INTERVIEW PREPARATION -> Click here to download the data files for the usecases given below" the custpayments_ORIG.sql and payments.txt into /home/hduser/hiveusecase

2. Ensure Hadoop, MYSQL, Hive remote metastore is running.

Usecase 1:

****

1. Login to Mysql and execute the sql file to load the custpayments table:

source /home/hduser/hiveusecase/custpayments_ORIG.sql

2. Write sqoop command to import data from customerpayments table with 2 mappers, with enclosed by " (As we have ',' in the data itself we are importing in sqoop using --enclosed-by option into the location /user/hduser/custpayments).

3. Create a hive external table and load the sqoop imported data to the hive table called custpayments. As we have ',' in the data itself we are using quotedchar option below with the csv serde option as given below as example, create the table with all columns.

```
create external table custmaster (customerNumber int,customername string,contactlastname string,contactfirstname string)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

WITH SERDEPROPERTIES (

  "separatorChar" = ",",

  "quoteChar"     = "\"")

LOCATION '/user/hduser/custpayments/';
```

4. Copy the payments.txt into hdfs location /user/hduser/paymentsdata/ and Create an external table namely payments with customernumber, checknumber, paymentdate, amount columns to point the imported payments data.

5. Create an external table called cust_payments in avro format and load data by doing inner join of custmaster and payments tables, using insert select customernumber, contactfirstname,contactlastname,phone, creditlimit from custmaster and paymentdate, amount columns from payments table

6. Create a view called custpayments_vw to only display customernumber,creditlimit,paymentdate and amount selected from cust_payments.

7. Extract only customernumber,creditlimit,paymentdate and amount columns either using the above view/cust_payments table into hdfs location /user/hduser/custpaymentsexport with '|' delimiter.

8. Export the data from the /user/hduser/custpaymentsexport location to mysql table called cust_payments using sqoop export with staging table option using records per statement 100 and mappers 3.

Usecase 2:

Managing Fixed Width Data:

1. Copy the below fixed data into a linux file, load into a hive table called cust_fixed_raw in a column rawdata.

1 Lara chennai 55 2016-09-2110000

2 vasudevan banglore 43 2016-09-2390000

3 Paul chennai 33 2019-02-2020000

4 David Hanna New Jersey29 2019-04-22

2. Create a temporary table called cust_delimited_parsed_temp with columns such as id,name,city,age,dt,amt and load the cust_fixed_raw table using substr.

for eg to select id column : select trim(substr(rawdata,1,3)) from cust_fixed_raw;

3. Export only id, dt and amt column into a mysql table cust_fixed_mysql using sqoop export.

4. Load only chennai data to another table called cust_parsed_orc of type orc format partitioned based on dt.

5. Create a json table called cust_parsed_json (to load into json use the following steps).

cd /home/hduser/hiveusecase

wget https://repo1.maven.org/maven2/org/apache/hive/hcatalog/hive-hcatalog-core/1.2.1/hive-hcatalog-core-1.2.1.jar

```
add jar /home/hduser/hiveusecase/hive-hcatalog-core-1.2.1.jar;

create external table cust_parsed_json(id int, name string,city string, age int)

ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

stored as textfile

location '/user/hduser/custjson';
```

6. Insert into the cust_parsed_json only non chennai data using insert select of id,name,city, age from the cust_delimited_parsed_temp table.

7. Schema migration:

Convert the XML table called xml_bank created in the actual usecase to JSON data by the same way like step 5 using create table as select.

For eg:

```
create external table xml_json

ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

stored as textfile

location '/user/hduser/custxmljson'

as select * from xml_bank;
```

8. Import data from mysql directly creating static partition based on city=chennai as given below for additional knowledge.

```
sqoop import \

--connect jdbc:mysql://localhost:3306/custdb \

--username root \

--password root \
```

```
--query "select custid,firstname,age from customer where city='chennai' and \$CONDITIONS" \

--target-dir /user/hduser/hiveext/ \

--split-by custid \

--hive-overwrite \

--hive-import \

--create-hive-table \

--hive-partition-key city \

--hive-partition-value 'chennai' \

--fields-terminated-by ',' \

--hive-table default.custinfo \

--direct
```