Apache Sqoop ile Veri Transferi Uygulaması

Görev 1

Adım 1: datasets klasörü içerisine indirdiğiniz retail_db klasöründe bulunan csv dosyalarını yükleyiniz.

- Eğer datasets içinde retail_db yoksa indiriniz.

```
wget https://raw.githubusercontent.com/erkansirin78/datasets/master/retail_db/categories.csv
wget https://raw.githubusercontent.com/erkansirin78/datasets/master/retail_db/customers.csv
wget https://raw.githubusercontent.com/erkansirin78/datasets/master/retail_db/departments.csv
wget https://raw.githubusercontent.com/erkansirin78/datasets/master/retail_db/order_items.csv
wget https://raw.githubusercontent.com/erkansirin78/datasets/master/retail_db/orders.csv
wget https://raw.githubusercontent.com/erkansirin78/datasets/master/retail_db/products.csv
```

Adım 2: Her dosyanın postgresql tablosunu oluşturunuz.

▼ From Terminal

```
psql -h localhost -d traindb -U train -c "create table if not exists categories(categoryId int, categoryDepartmentId int, categoryName VARCHAR(50));"
psql -h localhost -d traindb -U train -c "TRUNCATE TABLE categories;"
psql -h localhost -d traindb -U train -c "\copy categories FROM '/home/train/datasets/retail_db/categories.csv' DELIMITERS ',' CSV HEADER;"

psql -h localhost -d traindb -U train -c "create table if not exists customers(customerId int, customerFName varchar(50), customerLName varchar(50), customerEmail varchar(50), customerPassword varchar(20), customerStreet varchar(50), customerCity varchar(50), customerState varchar(10), customerZipcode int);"
psql -h localhost -d traindb -U train -c "TRUNCATE TABLE customers;"
psql -h localhost -d traindb -U train -c "\copy customers FROM '/home/train/datasets/retail_db/customers.csv' DELIMITERS ',' CSV HEADER;"

psql -h localhost -d traindb -U train -c "create table if not exists departments(customerIddepartmentId int, departmentName varchar(20));"
psql -h localhost -d traindb -U train -c "TRUNCATE TABLE departments;"
psql -h localhost -d traindb -U train -c "\copy departments FROM '/home/train/datasets/retail_db/departments.csv' DELIMITERS ',' CSV HEADER;"

psql -h localhost -d traindb -U train -c "create table if not exists order_items(orderItemName int,orderItemOrderId int,orderItemProductId int,orderItemQuantity int,orderItemSubTotal float8,orderItemProductPrice float8);"
psql -h localhost -d traindb -U train -c "TRUNCATE TABLE order_items;"
psql -h localhost -d traindb -U train -c "\copy order_items FROM '/home/train/datasets/retail_db/order_items.csv' DELIMITERS ',' CSV HEADER;"

psql -h localhost -d traindb -U train -c "create table if not exists orders(orderId int, orderDate timestamp,orderCustomerId int, orderStatus varchar(20));"
psql -h localhost -d traindb -U train -c "TRUNCATE TABLE orders;"
psql -h localhost -d traindb -U train -c "\copy orders FROM '/home/train/datasets/retail_db/orders.csv' DELIMITERS ',' CSV HEADER;"

psql -h localhost -d traindb -U train -c "create table if not exists products(productId int, productCategoryId int, productName varchar(50), productDescription varchar(50), productPrice float8, productImage varchar(255));"
psql -h localhost -d traindb -U train -c "TRUNCATE TABLE products;"
psql -h localhost -d traindb -U train -c "\copy products FROM '/home/train/datasets/retail_db/products.csv' DELIMITERS ',' CSV HEADER;"
```

▼ From PSQL

`psql -U train -d traindb`

```
create table if not exists categories(categoryId int, categoryDepartmentId int, categoryName VARCHAR(50));
TRUNCATE TABLE categories;
\copy categories FROM '/home/train/datasets/retail_db/categories.csv' DELIMITERS ',' CSV HEADER;

select * from categories c limit 5;

create table if not exists customers(customerId int, customerFName varchar(50), customerLName varchar(50), customerEmail varchar(50), customerPassword varchar(20), customerStreet varchar(50), customerCity
TRUNCATE TABLE customers;
\copy customers FROM '/home/train/datasets/retail_db/customers.csv' DELIMITERS ',' CSV HEADER;

create table if not exists departments(customerIddepartmentId int, departmentName varchar(20));
TRUNCATE TABLE departments;
\copy departments FROM '/home/train/datasets/retail_db/departments.csv' DELIMITERS ',' CSV HEADER;

create table if not exists order_items(orderItemName int,orderItemOrderId int,orderItemProductId int,orderItemQuantity int,orderItemSubTotal float8,orderItemProductPrice float8);
TRUNCATE TABLE order_items;
\copy order_items FROM '/home/train/datasets/retail_db/order_items.csv' DELIMITERS ',' CSV HEADER;

create table if not exists orders(orderId int, orderDate timestamp,orderCustomerId int, orderStatus varchar(20));
TRUNCATE TABLE orders;"
\copy orders FROM '/home/train/datasets/retail_db/orders.csv' DELIMITERS ',' CSV HEADER;

create table if not exists products(productId int, productCategoryId int, productName varchar(50), productDescription varchar(50), productPrice float8, productImage varchar(255));
TRUNCATE TABLE products;
\copy products FROM '/home/train/datasets/retail_db/products.csv' DELIMITERS ',' CSV HEADER;
```

Görev 2

Adım 1: Sqoop kullanarak hive veri tabanına aktarınız.

▼

Daha önce yapılmış ise overwrite yapılmalı, bunun için ise`--delete-target-dir` eklenmeli,

replace `--creata-hive-table` to `--hive-overwrite`

```
sqoop import --connect jdbc:postgresql://localhost/traindb \
--driver org.postgresql.Driver \
--username train --password-file file:///home/train/sqoop.password \
--table categories --delete-target-dir \
--m 1 --hive-import --hive-overwrite --hive-table test1.categories \
--target-dir /tmp/categories
```

```
sqoop import --connect jdbc:postgresql://localhost:5432/traindb  \
--driver org.postgresql.Driver \
--username train --password-file file:///home/train/sqoop.password \
--table  customers --delete-target-dir \
-m 1 --hive-import  --hive-overwrite --hive-table test1.customers \
--target-dir /tmp/customers
```

```
sqoop import --connect jdbc:postgresql://localhost/traindb  \
--driver org.postgresql.Driver \
--username train --password-file file:///home/train/sqoop.password \
--table  departments --delete-target-dir \
--m 1 --hive-import  -hive-overwrite --hive-table test1.departments \
--target-dir /tmp/departments
```

```
sqoop import --connect jdbc:postgresql://localhost/traindb  \
--driver org.postgresql.Driver \
--username train --password-file file:///home/train/sqoop.password \
--table  order_items \
--m 1 --hive-import  --create-hive-table --hive-table test1.order_items \
--target-dir /tmp/order_items
```

```
sqoop import --connect jdbc:postgresql://localhost/traindb  \
--driver org.postgresql.Driver \
--username train --password-file file:///home/train/sqoop.password \
--table  orders --delete-target-dir \
--m 1 --hive-import  --create-hive-table --hive-table test1.orders \
--target-dir /tmp/orders
```

```
sqoop import --connect jdbc:postgresql://localhost/traindb  \
--driver org.postgresql.Driver \
--username train --password-file file:///home/train/sqoop.password \
--table  products --delete-target-dir \
--m 1 --hive-import  --create-hive-table --hive-table test1.products \
--target-dir /tmp/products
```

Adım2: Beeline ile ORC formatına çeviriniz.

```
create table if not exists test1.categories_orc_snappy stored as orc TBLPROPERTIES ('orc.compress'='SNAPPY') as select * from test1.categories;
drop table test1.categories;
alter table test1.categories_orc_snappy rename to categories;
```

Tek aktarımda ORC formatı için kod örneği:

```
sqoop import --connect jdbc:postgresql://cloudera/retail  \
--driver org.postgresql.Driver \
--username kullanici_adi --password Şifre\
--query "select invoiceno, stockcode, description, quantity, invoicedate, \
 unitprice, customerid, country, id, adcools1 from online_retail WHERE invoiceday = '2010-12-03' AND \$CONDITIONS " \
--m 1 \
--hive-partition-key invoiceday --hive-partition-value '2010-12-03' --hive-table sqoop_works.online_retail \
--hcatalog-database sqoop_works --hcatalog-table online_retail --hcatalog-storage-stanza "stored as orcfile"
```