SQOOP COMMANDS

Sqoop:
-->It is a tool which is used to transfer data from relational database to hadoop(hdfs, hive, hbase) and vice versa.
It is used for data ingestion/migration.

Architecture: It is the system design which is used to build a product/software.
Framework: It is used on top of architecture to provide functionality and build the product. It is a combination of tools and components which work together.
Tool: It is an instrument which is used to handle a particular functionality in a software

-->Sqoop IMPORT is used to transfer data from database to hadoop
Sqoop EXPORT is used to transfer data from hadoop to database
Sqoop EVAL is used to get a feel of data ie run queries on tables of databases

-->**To import all the tables**

```
Sqoop-import-all-tables \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--as-sequencialfile \  (we mention the file format here default is text file)
--warehouse-dir \output (this directory shdn't exist before)
```

Target-dir   vs   warehouse-dir:
-->target-dir : It is used when we have a single table to be imported which gets created in one particular folder

```
--target-dir  /user/cloudera/data
--warehouse-dir  /user/cloudera/data/orders (if warehouse used for single table)
```

-->warehouse-dir: it is used to create sub-directories of multiple tables that are imported

```
--warehouse-dir /user/cloudera/sqoop/employee
--warehouse-dir /user/cloudera/sqoop/persons
--warehouse-dir /user/cloudera/sqoop/retail
--warehouse-dir /user/cloudera/sqoop/billing
```

Sqoop Guide:
-->sqoop help : Lists all related commands being used in sqoop
-->sqoop help eval : Lists all related commands in "eval"

Redirecting Logs:
How to view the job logs later?
-->If suppose we want to see the logs of the import later we can redirect them to a path where they will be saved and can be viewed later.

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--warehouse-dir /user/cloudera/output1 1>output.log(execution messages)
2>others.log (complete processing log)
```

**Note: The log files can be with any name and extension and stored in local path(/home/cloudera)

-->to view the logs we have to give
-->cat output.log ()
-->cat others.log (where we have complete processing)
-->These log files are stored in local

Boundary query:
How mappers divide the work?
-->To know how 4 mappers divide the work among themselves.
-->It is based on the range defined by the primary key.
[(max(primary_key) - min(primary_key) /4] ~ no. of records taken by each mapper.

-->you can view the runtime log generated where the splitting is taking place
-->based on the range obtained mappers will process data within the range

-->ex. 1-100 : records in range of 1-100 (primary key) is taken up by mapper 1
100-200 : mapper 2

200-300: mapper 3
300-400: mapper 4

How our data can be compressed?

How we can compress the data ?

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--compress \
--warehouse-dir /user/cloudera/output1
```

How to import only particular columns?

-->To view top 10 records

-->hadoop fs -ls /output/partm-00000 | head

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--columns customer_id,customer_fname \
--warehouse-dir /user/cloudera/output1
```

Putting it all together

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--compress \
--columns customer_id,customer_fname \
--where "customer_city in ("new york") " \
--warehouse-dir /user/cloudera/output1
```

How to set customized boundary value when we have outliers?
-->suppose we inserted a record with primary key 800000 where total records is 65000(max primary key)
So system creates boundary value of (800000 - 0) which is wrong
Due to this major effort falls on mapper 1 as it processes ~2lakh records in our case all the 65000 records
To avoid this we set the boundary query value

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--boundary-query "select 1, 65000" (here we r specifying the range of primary key)
--warehouse-dir /user/cloudera/output1
```

Split-by USE CASE:
What if primary key is not present or unevenly distributed?
-->We can use spilt-by when a table doesn't have primary keys
-->And when the primary is unevenly distributed
-->we can set mappers to 1 (not recommended as parallelism doesn't exist)

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--split-by "customer_zipcode" \
--warehouse-dir /user/cloudera/no_pk
```

-->here we have used customer zipcode as index to split data into 4 mappers

-->It creates splits based on max , min values
-->if the primary key is a text column (customer name) based on ACSII values data will be split.

When primary key is text field?

sqoop import \
-Dorg.apache.sqoop.splitter.allow_text_splitter=true \ (optional as it works without this syntax as well)
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--split-by "customer_state" \
--target-dir /user/cloudera/no_pk1

How to auto-reset mappers?

-->It means if there is no primary key then auto set mappers to 1
-->if there is primary key then set mappers to 8
-->by default mappers will be 4

sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--autoreset-to-one-mapper (or) --m 1 \
--num-mappers 8 \
--target-dir /user/cloudera/no_pk1

How to delimit fields and rows in sqoop?

-->It means we can separate fields by a delimiter character and separate rows by a delimiter character

sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \

```
--password cloudera \
--table customers \
--fields-terminated-by '|' \
--lines-terminated-by ';' \
--target-dir /user/cloudera/no_pk1
```

How to create a hive table using sqoop?
-->It creates an empty table in hive using sqoop and imports the schema of table from mysql

```
sqoop create-hive-table \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--hive-table customers
```

How to view detailed logs in sqoop?
-->"verbose" command is used to view detailed logs of data imported through sqoop which can be used for debugging

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--verbose
--target-dir /user/cloudera/no_pk1
```

How to store output files in already existing folder?
-->"append" command is used to add output files to a pre-existing folder

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--target-dir /user/cloudera/no_pk1
```
--append

How to overwrite an output folder?
-->It causes the existing folder to be deleted (if any) and creates a new folder in its place.

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table customers \
--target-dir /user/cloudera/no_pk1
```
--delete-target-dir

How to deal with null values?
-->we can automatically append null values with a predefined value

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_db
--password cloudera \
--table customers \
--target-dir /user/cloudera/no_pk1
```
--null-non-string "-1"
--null-as-string "NA"

Sqoop Export:

Sqoop Export:
How to export data from Hdfs to Rdbms using sqoop?
Step 1: create a table having same schema in rdbms
Step 2: Use sqoop cmd to export the data

```
sqoop export \
--connect "jdbc:mysql://quickstart.cloudera:3306/banking" \
--username root \
--password cloudera \
--table card_transactions \
--export-dir /user/cloudera/card_trans.csv
```

-->data from csv in hadoop is exported to table in msql

What happens when we have partial export?
-->in some use cases when the export fails, then we see that there is partial export of data which is bad for our application.
-->we make sure the database transaction is atomic ie it either succeeds or fails completely.
-->we make use of **staging table** for this purpose which acts like an intermediator.
-->when the export starts records are first fetched into staging table, when there is any error data will not be migrated to main table.
-->if the export succeeds data will be migrated from staging table to main table

```
sqoop export \
--connect "jdbc:mysql://quickstart.cloudera:3306/banking" \
--username root \
--password cloudera \
--table card_transactions \
--staging-table card_transactions_stage \
--export-dir /user/cloudera/card_trans.csv
```

Sqoop Incremental:
What to do when we want to insert new data into hdfs on daily basis?
There are two methods to do it:
1.incremental append
2.incremental lastmodified

-->we can insert new data records into hdfs through sqoop regularly.
-->It can done using the sqoop command

--incremental-append : It is used to insert the new data records into hdfs using sqoop

--check-column : based on which column we can decide on new records to be added (generally primary key)
--last-value : signifies after this value(primary key) new records can be inserted

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/incremental \
--incremental append \
--check-column order_id \
--last-value 0
```

-->After first import is done and 68883 records were added. Now for the next newly added records

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table orders \
--incremental-append \
--check-column order_id \
--target-dir /user/cloudera/no_pk1 \
--last-value 68883
```

```
  sqoop import \
 --connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
 --username retail_dba \
 --password cloudera \
 --table orders \
 --where "order_id > 68000" \
 --warehouse-dir /user/cloudera/incremental \
 --incremental append \
 --check-column order_customer_id \
 --last-value 8440
```

How to update and insert records using sqoop?
-->"incremental lastmodified" is a method used to update the records and insert new records as well.
-->Based on the last modified date we will perform update and insert
-->check for order_date and update/insert records last modified after '2021-06-25 12:00:00'
-->"append" is used to write the files into given directory without creating new one

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/incremental \
--incremental lastmodified \
--check-column order_date \
--last-value '2021-06-25 12:00:00' \
--append
```

-->we have to make note of last value so that in our next import we can use it to make new updations/insertion

How we can have only latest records in our table?

-->we get duplicates in our table with both old and new records after updating them.

-->To make sure we have only the latest records in our table we use "merge-key" field in sqoop
-->based on "order_id" records will be merged and we have only the latest record in our table

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/incremental \
--incremental lastmodified \
--check-column order_date \
--last-value 0 \  (right from the beginning)
--merge-key order_id
```

Sqoop job:
How we can automate sqoop import/export?
-->we can automate running of sqoop jobs using airflow or oozie tools.
-->As seen previously we have to keep track of last value which has to be passed to the next import which is a tiring task.
-->we can automate this by creating a sqoop job which will save the state of job and pass it to the next sqoop import of same job.

Job creation:

```
sqoop job \
--create job_orders \
-- import \   (there is a space between "--" and "import"
--connect "jbdc:mysql://quickstart.cloudera:3306/retail_db" \
--username root \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/sqoop_job \
--incremental append \
--check-column order_id \
--last-value 0
```

View list of jobs:
-->sqoop job --list

Run Job:
-->sqoop job --exec job_orders

To view metadata of job:
-->sqoop job --show job_orders

Where is this metadata stored?
-->it is stored in /home/cloudera (local path) in a hidden directory (.sqoop)
-->ls -altr

****Note:**
-->If the split-by column is a varchar then based on ASCII values the data is divided among the mappers.
-->If the primary key contains randomized values (not exist in real scenario) then u can insert a column with sequential key and make is primary key.
-->If data contains duplicate records even then data is divided based on boundary vals query and few mappers may get more data which might be duplicates.
-->How boundary value works for outliers
Suppose ur records are 1,100,200 only three records
Now 200-1/4 approximately 50 records
1 mapper will only the records between 1 to 50 ie only one record
2nd mapper also holds one
3 rd mapper is empty
4 th one one record
Records goes to mapper based on range of primary key