Web: Inceptez.com Mail: info@inceptez.com Call: 7871299810, 7871299817

SQOOP INCREMENTAL USECASE

Import Scenarios:

1. **Import all data from the source by deleting target dir (Delete and load).**

Application:

- **When the source data is a rapidly growing small dimension table that undergoes several inserts/updates/deletes.**
- **For Eg. Product table that may change regularly and small.**

```
sqoop import --connect jdbc:mysql://127.0.0.1/custdb --username root -P -table customer -m 3 \
--split-by custid --target-dir sqoop_import --delete-target-dir --direct;
```

2. **Import all data from the source and append to the target dir (Append with all source data, may have duplicates).**

Application:

- **When the source data only grows by inserting new rows**
- **Pull only newly added data and append to the target directory.**
- **For Eg: Customer group table may be inserted with new customer groups.**

```
sqoop import --connect jdbc:mysql://127.0.0.1/custdb --username root --password root –table
customer -m 3 --split-by city --append --fetch-size 100;
```

CDC (Change Data Capture) and SCD (Slowly Changing Dimension) concepts implemented below

3. Import only (inserted) newly added data from the DB to HDFS and get it appended options used (check column, last value incremental append)
(Append only with newly added source data, No duplicates loaded in the target).

Application:

- **When the source data only grows which may have huge volume of data and it grows only by inserting new rows**
- **Append to the target directory, pull only newly added data on a daily basis to avoid pulling all rows again and again.**
- **For Eg: Transaction or customer events table when daily upto 1 million rows will be loaded with overall table contains nearly billion rows.**

```
sqoop import --connect jdbc:mysql://127.0.0.1/custdb --username root --password root -table customer
-m 1 --target-dir incrimport --incremental append --check-column custid --last-value 6
```

4. Import only (inserted and updated) newly added or modified data from the DB to HDFS and appended. (check column, last value, incremental lastmodified)
(Append only with newly added and updated source data, Duplicates/history loaded in the target- Slowly changing dimension type 2).

Sqoop Import Incremental last modified

Application:

- **When the source data which may have huge volume of data and it grows/modified either by inserting new rows or updating existing rows**
- **Append to the target directory (both inserted and updated data), pull only the updated/inserted data on a daily basis to avoid pulling all rows again and again.**
- **For Eg: Customer master table that gets inserted or updated with customer city, address etc, or adding new customers on a daily basis it maintains the history of the customer changes in the hdfs to understand the current and the previous cities of the customers.**

```
create table customer_lastmodified as select * from customer;

alter table customer_lastmodified add upddt date;

update customer_lastmodified set upddt='2018-10-10';
```

Importing all data for the first time
(Assume 2018-10-10 is the date from when the customer_lastmodified table is created and loaded with data)

```
sqoop import --connect jdbc:mysql://127.0.0.1/custdb --username root --password root --table customer_lastmodified -m 1 --target-dir incrimportlm --incremental lastmodified --check-column upddt --last-value 2018-10-10
```

To check incremental last modified, lets add 1 new row and update an existing row with upddt as 2018-10-11

```
insert into customer_lastmodified values(9,'inceptez1','tech','hyd',3,'2017-09-28',10000,'2018-10-11');

update customer_lastmodified set upddt='2018-10-11',city='Chennai' where custid =4;
```

Import only upddt greater or equal to 2018-10-11 and append in the HDFS location incrimportlm

```
sqoop import --connect jdbc:mysql://127.0.0.1/custdb --username root --password root --table customer_lastmodified -m 1 --target-dir incrimportlm --incremental lastmodified --check-column upddt --last-value 2018-10-11 --append
```

Note: *Above append option we are using to append the data in the targer hdfs, if we use –delete-target-dir rather than append it deletes and load, if you don't use any option the load will fail if the target directory already exists.*

hadoop fs -cat incrimportlm/*

5. Import only (inserted and updated) newly added or modified data from the DB to HDFS and get it merged in the target (insert else update).
(check column, last value incremental lastmodified --merge-key custid)
(insert else update - Slowly Changing Dimension Type 1).

Application:

- **When the source data which may have huge volume of data and it grows/modified either by inserting new rows or updating existing rows**
- **Append to the target directory (both inserted and updated data), pull only the updated/inserted data on a daily basis to avoid pulling all rows again and again.**
- **For Eg: Customer master table that gets inserted or updated with customer city, address etc, or adding new customers on a daily basis, it won't maintains the history of the customer changes in the hdfs.**

update customer_lastmodified set upddt='2018-10-12',city='hyderabad' where custid=8;

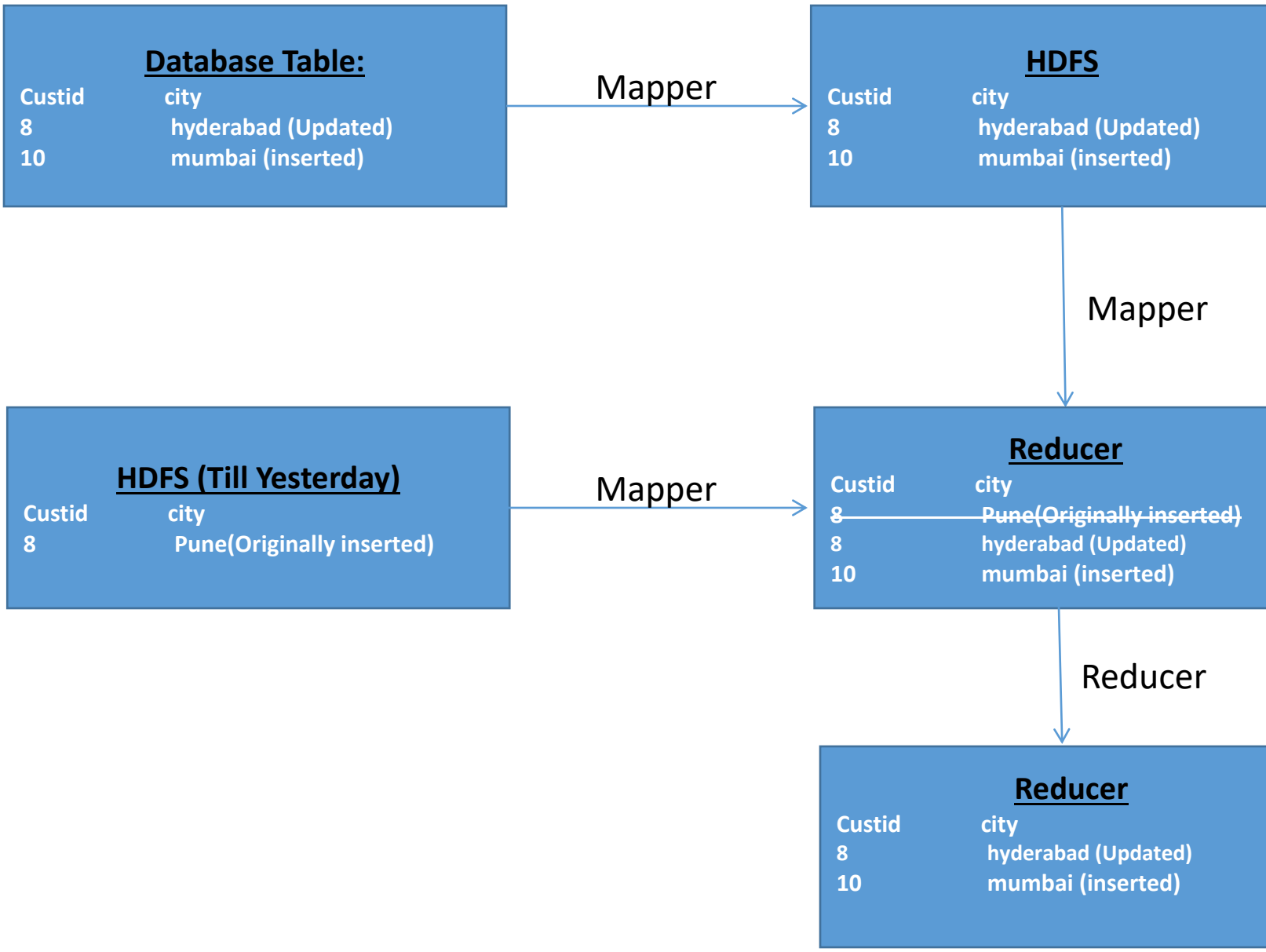
insert into customer_lastmodified values(10,'Interview','Important','mumbai',3,'2017-09-28',10000,'2018-10-12');

Sqoop uses Reducer in the below case (Import and update hdfs data if custid exist and insert if custid is not exist in the HDFS location incrimportlm, only upddt greater or equal to 2018-10-12)

sqoop import --connect jdbc:mysql://127.0.0.1/custdb --username root --password root -table customer_lastmodified -m 1 --target-dir incrimportlm --incremental lastmodified --check-column upddt --last-value 2018-10-12 --merge-key custid

Sqoop mapper pulls both inserted/updated data from the DB -> hdfs -> mapper reads from the above sqoop imported hdfs location today-> reducer reads from existing hdfs history data imported till yesterday, matches both new (today) and existing data (upto yesterday) and loads only new data replacing the existing one if the merge key matches -> hdfs

Refer the below image to understand what happens with sqoop merge that uses reducer also.


Database Table:
Custid city
8 hyderabad (Updated)
10 mumbai (inserted)
Mapper
HDFS
Custid city
8 hyderabad (Updated)
10 mumbai (inserted)
Mapper
HDFS (Till Yesterday)
Custid city
8 Pune(Originally inserted)
Mapper
Reducer
Custid city
8 Pune(Originally inserted)
8 hyderabad (Updated)
10 mumbai (inserted)
Reducer
Reducer
Custid city
8 hyderabad (Updated)
10 mumbai (inserted)