



# SCD in Hive

...

Slowly changing Dimension

# IMPORTANT

## **Copyright Infringement and Illegal Content Sharing Notice**

All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

**Trendytech Insights** legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.



# SCD

**Stands for slowly changing dimension.**

**Also called as change data capture.**

**In data warehousing, slowly-changing dimensions (SCDs) capture data that changes at irregular and unpredictable intervals.**

**There are several common approaches for managing SCDs, corresponding to different business needs.**

**For example you may want to track full history in a table, allowing you to track the evolution of a customer over time.**

**In other cases you don't care about history but need an easy way to synchronize reporting systems with source operational databases.**



# most common SCD Types

**SCD Type 1: Overwrite old data with new data. The advantage of this approach is that it is extremely simple, and is used any time you want an easy way to synchronize reporting systems with operational systems. The disadvantage is you lose history any time you do an update.**

**SCD Type 2: Add new rows with version history. The advantage of this approach is that it allows you to track full history. The disadvantage is that your dimension tables grow without limit and may become very large. When you use Type 2 SCD you will also usually need to create additional reporting views to simplify the process of seeing only the latest dimension values.**



# Steps to implement SCD 1

**Let us first create the two tables:**

**table2 is old table**

**table3 is new table**

**We want to sync the table2 based on changes in table3**

**create table table2 (col1 String, col2 int) row format delimited fields terminated by ',';**

**create table table3 (col1 String, col2 int) row format delimited fields terminated by ',';**



# Steps to implement SCD 1

Let us now load the data in both the tables

load data local inpath '/home/cloudera/Downloads/table2\_data.csv' into table table2;

load data local inpath '/home/cloudera/Downloads/table3\_data.csv' into table table3;

```
Time taken: 0.051 seconds
hive> create table table2 (col1 String, col2 int) row format delimited fields terminated by ',';
OK
Time taken: 0.06 seconds
hive> create table table3 (col1 String, col2 int) row format delimited fields terminated by ',';
OK
Time taken: 0.066 seconds
hive> load data local inpath '/home/cloudera/Downloads/table2_data.csv' into table table2;
Loading data to table trendytech.table2
Table trendytech.table2 stats: [numFiles=1, totalSize=129]
OK
Time taken: 0.253 seconds
hive> load data local inpath '/home/cloudera/Downloads/table3_data.csv' into table table3;
Loading data to table trendytech.table3
Table trendytech.table3 stats: [numFiles=1, totalSize=115]
OK
```



# Steps to implement SCD 1

let us see the data in both the tables.

```
hive> select * from table2;  
OK  
John      1300  
Albert    1200  
Mark      1000  
Frank     1150  
Loopa     1100  
Lui       1300  
Lesa      900  
Pars      800  
leo       700  
lock      650  
pars      900  
jack      700  
fransis   1000
```

```
hive> select * from table3;  
OK  
John      1500  
Albert    1900  
Mark      1000  
Frank     1150  
Loopa     1100  
Lui       1300  
Lesa      900  
Pars      800  
leo       700  
lock      650  
Bhut      800  
Lio       500
```



# Query for SCD 1

```
select
  case when cdc_codes ='Update' Then table3s
    when cdc_codes = 'NoChange' then table2s
    when cdc_codes = 'New' then table3s
    when cdc_codes = 'Deletes' then table2s
  end
from (select
  case  when table2.col1=table3.col1 and table2.col2=table3.col2 then 'NoChange'
    when table2.col1=table3.col1 and table2.col2<>table3.col2 then 'Update'
    when table2.col1 is null then 'New'
    when table3.col1 is null then 'Deletes'
  end as cdc_codes,
  concat(table2.col1,',',table2.col2) as table2s,
  concat(table3.col1,',',table3.col2) as table3s
from table2
full outer join table3 on table2.col1=table3.col1) as b1
```





# Steps to implement SCD 1

```
hive> select
>     case when cdc_codes = 'Update' Then table3s
>           when cdc_codes = 'NoChange' then table2s
>           when cdc_codes = 'New' then table3s
>           when cdc_codes = 'Deletes' then table2s
>     end
> from (select
>     case
>         when table2.col1=table3.col1 and table2.col2=table3.col2 then 'N
oChange'
>         when table2.col1=table3.col1 and table2.col2<>table3.col2 then '
Update'
>         when table2.col1 is null then 'New'
>         when table3.col1 is null then 'Deletes'
>     end as cdc_codes,
>     concat(table2.col1,',',table2.col2) as table2s,
>     concat(table3.col1,',',table3.col2) as table3s
> from table2
> full outer join table3 on table2.col1=table3.col1) as b1
> ;
Query ID = cloudera_20200430135757_ed52d26b-35b8-4952-940b-7fb8e8129a7c
Total jobs = 1
```



# Steps to implement SCD 1

Let us see the results now (we can then insert overwrite these results in table2)

```
Wrote: 178 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 700 msec
OK
Albert,1900
Bhut,800
Frank,1150
John,1500
Lesa,900
Lio,500
Loopa,1100
Lui,1300
Mark,1000
Pars,800
fransis,1000
jack,700
leo,700
lock,650
pars,900
Time taken: 29.907 seconds, Fetched: 15 row(s)
hive> █
```



# What are various strategies in SCD 2

**SCD Type 2 Versioning:** In versioning method, a sequence number is used to represent the change. The latest sequence number always represents the current row and the previous sequence numbers represents the past data.

surrogate_key	customer_id	customer_name	Location	Version
1	1	Marston	Illions	1
2	1	Marston	Seattle	2



# What are various strategies in SCD 2

**SCD Type 2 Flagging:** In flagging method, a flag column is created in the table. The current record will have the flag value as 1 and the previous records will have the flag as 0.

surrogate_key	customer_id	customer_name	Location	Version
1	1	Marston	Illions	0
2	1	Marston	Seattle	1



# What are various strategies in SCD 2

**SCD Type 2 Effective Date:** In Effective Date method, the period of the change is tracked using the start\_date and end\_date columns in the dimension table.

surrogate_key	customer_id	customer_name	Location	Start_date	End_date
1	1	Marston	Illions	01-Mar-2010	20-Fdb-2011
2	1	Marston	Seattle	21-Feb-2011	NULL



# Steps to implement SCD 2 (out of scope of this course)

Here's the detailed implementation of slowly changing dimension type 2 in Hive

Assuming that the source is sending a complete data file i.e. old, updated and new records.

**Steps:**

**STEP 1: Load the recent file data to STG table**

**STEP 2: Select all the expired records from HIST table**

**select \* from HIST\_TAB where exp\_dt != '2099-12-31'**

**STEP 3: Select all the records which are not changed from STG and HIST using inner join and filter on HIST.column = STG.column as below**

**select hist.\* from HIST\_TAB hist inner join STG\_TAB stg on hist.key = stg.key where hist.column = stg.column**



# Steps to implement SCD 2

**STEP 4:** Select all the new and updated records which are changed from STG\_TAB using exclusive left join with HIST\_TAB and set expiry and effective date as below

```
select stg.*, eff_dt (yyyy-MM-dd), exp_dt (2099-12-31) from STG_TAB stg left join (select *  
from HIST_TAB where exp_dt = '2099-12-31') hist  
on hist.key = stg.key where hist.key is null or hist.column != stg.column
```

**STEP 5:** Select all updated old records from the HIST table using exclusive left join with STG table and set their expiry date as shown below:

```
select hist.*, exp_dt(yyyy-MM-dd) from (select * from HIST_TAB where exp_dt =  
'2099-12-31') hist left join STG_TAB stg  
on hist.key= stg.key where hist.key is null or hist.column!= stg.column
```

**STEP 6:** unionall queries from 2-5 and insert overwrite result to HIST table



# References/Additional reading

1. <https://www.folkstalk.com/2012/03/slowly-changing-dimensions-scd-types.html>
2. <https://blog.cloudera.com/update-hive-tables-easy-way-2/>
3. <https://community.cloudera.com/t5/Support-Questions/Best-and-Easy-way-to-implement-and-create-SCD2-in-Hive-and/td-p/182059>
4. <https://dwgeek.com/impala-hive-slowly-changing-dimension-scd-type-2.html/>





**We have learnt about SCD in Hive**

**Happy Learning!!!**



**5** Star Google Rated  
Big Data Course

**LEARN FROM THE EXPERT**



**9108179578**

**Call for more details**



# Follow US

**Trainer** Mr. Sumit Mittal

**Phone** 9108179578

**Email** trendytech.sumit@gmail.com

**Website** <https://trendytech.in/courses/big-data-online-training/>

**LinkedIn** <https://www.linkedin.com/in/bigdatabysumit/>

**Twitter** @BigdataBySumit

**Instagram** bigdatabysumit

**Facebook** <https://www.facebook.com/trendytech.in/>

**Youtube** [https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C\\_gUGSg](https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C_gUGSg)