

Slowly Changing Dimensions (SCD) in Data Warehouse

📅 July 3, 2016 (<http://dwgeek.com/slowly-changing-dimensions-scd.html/>) 👤 Vithal Sampagar (<http://dwgeek.com/author/vithal-sampagargmail-com/>) 📁 Data Warehouse (<http://dwgeek.com/category/dwh/>)

Succeed With Big Data

Free Whitepaper! 7 Tips to Succeed With Big Data. Download Now. [Go to tableau.com](http://tableau.com)



Slowly changing dimensions or SCD are dimensions that changes slowly over time, rather than regular bases. In data warehouse environment, there may be a requirement to keep track of the change in dimension values and are used to report historical data at any given point of time.

Read:

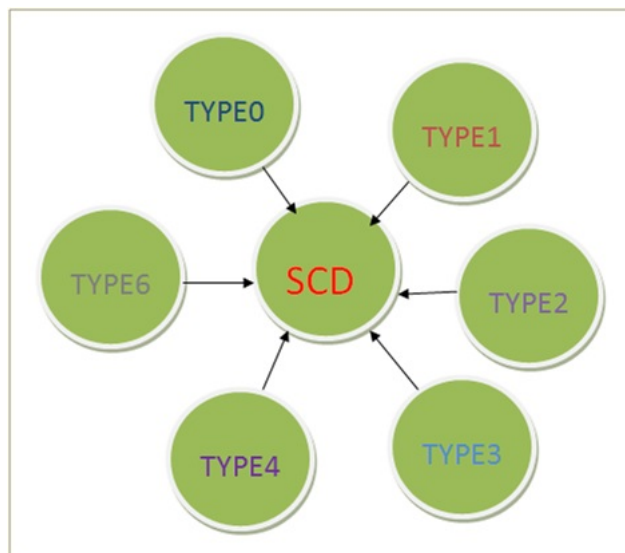


BMO

Smart online investing is simple with BMO SmartFolio. Set your objectives and our experts take care of the rest.

[Learn more](#)

- Star Schema model in Data Warehouse (<http://dwgeek.com/star-schema-model-data-darehouse.html/>)
- Snowflake Schem Model in Data Warehouse (<http://dwgeek.com/snowflake-schema-model-data-warehouse.html/>)
- Step by Step Guide to Dimensional Data Modeling (<http://dwgeek.com/guide-dimensional-modeling.html/>)
- Slowly Changing Dimensions (SCD) in Data Warehouse (<http://dwgeek.com/slowly-changing-dimensions-scd.html/>)
- Rapidly Changing Dimension in Data Warehouse (<http://dwgeek.com/rapidly-changing-dimension-data-warehouse.html/>)
- Data Warehouse Fact Constellation Schema and Design (<http://dwgeek.com/data-warehouse-fact-constellation-schema-design.html/>)



We can implement slowly changing dimensions (SCD) using various approaches.

- Type 0: Always retains original

- Type 1 : Keeps latest data, old data is overwritten
- Type 2 : Keeps the history of old data by adding new row
- Type 3 : Adds new attribute to store changed value
- Type 4 : Uses separate history table
- Type 6 : combination of type 1, 2 and 3

Type 0: Always retains original

The type 0 is passive method. And therefore, perform no actions on dimension changes. Values remain as they were at the time the dimension table record was first inserted, others may be overwritten.

Type 1: Keeps latest data, old data is overwritten

This method overwrites the old data, hence doesn't keep historical data.



Consider the patient table

SK	Pat_ID	Name	Cellphone
100	1	ABC	1234567890

In the above table, pat_ID is the natural key and SK is surrogate key column. Technically, SK column is not required as natural key uniquely identifies the records. Join tables using this key column and this column is unique key.



BMO

Sit back. Relax. Enjoy your coffee. BMO SmartFolio makes investing online easy because our experts do all the work.

[Learn more](#)

If the patient changes his/her cellphone number, overwrite the old data.

SK	Pat_ID	Name	Cellphone
100	1	ABC	9876543210

One of the Type 1 method disadvantage is, there is no historical data in data warehouse. However, Type 1 maintenance is very easy and advantage is reduced the data warehouse size.

Type 2: Keeps the history of old data by adding new row

This method keeps track of historical data by adding multiple rows of given key columns in dimension table. Some SCD type 2 implementations use effective from and to date with flag indicating latest record.

- Learn: SCD Type2 Design and Development (<http://dwgeek.com/scd-type-2-sql.html/>)

SK	Pat_ID	Name	Cellphone	EFF_FR_DT	EFF_TO_DT	flag
100	1	ABC	1234567890	2016-06-01	2016-06-10	0

101	1	ABC	9876543210	2016-06-10	NULL	1
-----	---	-----	------------	------------	------	---

Flag = 1 and EFF_TO_DT = NULL indicates current or latest tuple versions. In some implementation, data modeller uses future date (9999-12-31) as effective to date. In case if you want to track the historical records, you have to select all records which are non-null and flag = 0 values.

Adding new records in changes to the dimensional model in type 2 could be very expensive database operation so it is not recommended to use it in dimensions where a new attribute/columns could be added to table in the future.

Type 3: Adds new attribute to store changed value

This method of slowly changing dimensions keeps the limited history of the changed data in form of new column. The history preservation is limited to the number of columns designed for storing historical data. The original table structure in type 1 and 2 is same, but type 3 adds new column to store current data.

For example, in following example, add additional column to deal with cell phone number changes.

SK	Pat_ID	Name	Previous_Cellphone	Current_Cellphone
100	1	ABC	1234567890	9876543210

The disadvantage with this method is that it keeps limited history about changed data. In above table, new columns are added to hold new and changed data i.e. current_cellphone and previous_cellphones respectively.

Type 4: Uses separate history table

Type 4 usually uses the separate table to hold history data, where as original table keeps the current data, and create other table to store the changed data. Add surrogate key to both tables to and is used in fact table to identify original and history data.

Patient table

SK	Pat_ID	Name	Cellphone
101	1	ABC	9876543210

Patient History table

SK	Pat_ID	Name	Cellphone	CRT_DT
100	1	ABC	1234567890	2016-06-01
101	1	ABC	9876543210	2016-06-10

Type 6: Combination of type 1, 2 and 3

Finally, the Type 6 method combines the approaches of types 1, 2 and 3. Therefore, this type is called hybrid method. In this type, below are the changes to dimension table:

Current_record – keeping the current data

Historical_record – keeping historical or changed data

Effective_from_date – keeping start date of record, null for new records

Effective_to_date –keeping end of expired record

Curr_flag – indicates current records

Attribute change are captured in this type of SCD, a new table record is added as in SCD type 2. Overwrite the old information with the new data as in type 1. And preserve history in a historical_record as in type 3.

Below is type 6 table of the following example

Pat_ID	Name	Current_Cellphone	Historical_Cellphone	EFF_FR_DT	EFF_TO_DT	flag
1	ABC	9876543210	2345678901	2015-01-01	2016-01-20	0
1	ABC	9876543210	1234567890	2016-01-21	Null	1