

SCD

# Slowly Changing Dimension Type

The "**Slowly Changing Dimension**" is a common problem particular to data warehousing. In a nutshell, this applies to cases where the attribute for a record varies over time.

# Scenario of SCD

John is a customer with ABC Inc. company.

For the customer related details there is a table with columns CustID, CustName and CustCity.

Initially John's city is Delhi.

So originally the record of John looked like this:

CustID	CustName	CustCity
C1	John	Delhi

After some days John moved from Delhi to Mumbai.

Now how should ABC Inc. Company modify its customer table to reflect this change??

This is the "Slowly Changing Dimension" problem.

# Type 1 SCD

In Type 1 Slowly Changing Dimension, the new information simply overwrites the original information. In other words, no history is kept.

In our example, recall we originally have the

CustID	CustName	CustCity
C1	John	Delhi

# Type 1 SCD

After John moved from Delhi to Mumbai, the new information replaces the new record, and we have the following table:

CustID	CustName	CustCity
C1	John	<b>Mumbai</b>

# Advantages of Type 1 SCD

- This is the easiest way to handle the Slowly Changing Dimension problem, since there is no need to keep track of the old information.
- **Disk space required will be less.**

# Disadvantages of Type 1 SCD

**All history is lost.** By applying this methodology, it is not possible to trace back in history. For example, in this case, the company would not be able to know that John lived in Delhi before.



# Usage of Type1 SCD

- About 50% of the time.
- In **RDBMS model or OLTP systems** by using simple “Update” SQL statement we achieve Type 1 SCD !

# **When to use Type 1 SCD**

Type 1 slowly changing dimension should be used when it is not necessary for the data warehouse to keep track of historical changes.

# Type 2 SCD

In Type 2 Slowly Changing Dimension, a new record is added to the table to represent the new information. Therefore, both the original and the new record will be present. The new record gets its own primary key.

In our example, recall we originally have the following table:

CustID	CustName	CustCity
C1	John	Delhi

# Type 2 SCD

After John moved from Delhi to Mumbai, we add the new information as a new row into the table:

CustID	CustName	CustCity
C1	John	Delhi
<b>C1</b>	<b>John</b>	<b>Mumbai</b>

# Type 2 SCD

**Identification of old and new rows can be done by:**

1. Start Date & End Date Columns

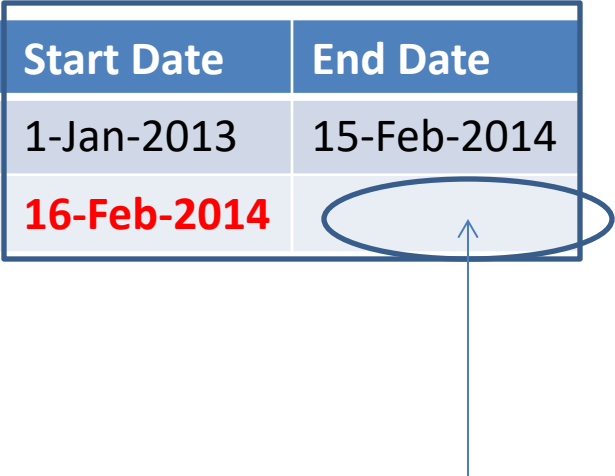
***Or***

2. One column having Status as Yes or No

# Type 2 SCD

Dates to Identify old row (s) & current row

CustID	CustName	CustCity	Start Date	End Date
C1	John	Delhi	1-Jan-2013	15-Feb-2014
<b>C1</b>	<b>John</b>	<b>Mumbai</b>	<b>16-Feb-2014</b>	



Null means current row

# Type 2 SCD

Status (*having Yes & No*) to Identify old row (s) & current row

CustID	CustName	CustCity	Status
C1	John	Delhi	No
<b>C1</b>	<b>John</b>	<b>Mumbai</b>	<b>Yes</b>

Yes means current row

# **Advantages of Type 2 SCD**

This allows us to accurately keep all historical information



# Disadvantages of Type 2 SCD

1. This will cause the size of the table to grow fast. In cases where the number of rows for the table is very high to start with, **storage and performance can become a concern.**
2. This necessarily **complicates** the ETL process.
3. It needs the **record identifier column** like Identity property, Sequence object or Auto number additionally to identify records of same customer id.

# Usage of Type 2 SCD

- About 50% of the time
- In **OLAP**
- This style can get used also in old Flat File Systems or in spreadsheets kind of data format.

# When to use Type 2 SCD

Type 2 slowly changing dimension should be used **when it is necessary for the data warehouse to track historical changes.**

# Type 3 SCD

- In Type 3 Slowly Changing Dimension, there will be two columns to indicate the particular attribute of interest, one indicating the original (or previous) value, and one indicating the current value.
- There will also be a column that indicates when the current value becomes active.

In our example, recall we originally have the

CustID	CustName	CustCity
C1	John	Delhi

# Type 3 SCD

To accommodate Type 3 Slowly Changing Dimension, we will now have the following columns:

1. Customer Key
2. Name
3. Original City
4. Current City
5. Effective Date

After John moved from Delhi to Mumbai, the original information gets updated, and we have the following table (assuming the effective date of change is January 15, 2010):

CustID	CustName	Original City	Current City	Effective Date
C1	John	Delhi	Mumbai	15-Jan-2010

# Advantages of Type 3 SCD

- This **does not increase the size of the table**, since new information is updated.
- This allows us to keep some part of history.

# Disadvantages of Type 3 SCD

- Type 3 will **not be able to keep all history** where an attribute is changed more than once.
- For example, if John later moves to Bangalore on December 15 2010, **the Delhi information will be lost.**



# Usage of Type 3 SCD

Type 3 is rarely used in actual practice.

# When to use Type 3 SCD

Type III slowly changing dimension should only be used when it is necessary for the data warehouse to track historical changes, and when such changes will only occur for a **finite number of time**.

# Solutions of first 3 types of SCD

There are in general three ways to solve this type of problem, and they are categorized as follows:

- **Type 1:** The new record replaces the original record. No trace of the old record exists.
- **Type 2:** A new record is added into the customer dimension table. Therefore, the customer is treated essentially as two people.
- **Type 3:** The original record is modified to reflect the change along with the old value .

## First 3 SCDs

CustKey	CustAltKey	Name	Phone
1	1002	Amy Alberts	555 123

Type 1



CustKey	CustAltKey	Name	Phone
1	1002	Amy Alberts	555 222

CustKey	CustAltKey	Name	City	Current	Start	End
1	1002	Amy Alberts	Vancouver	Yes	1/1/2000	

Type 2 ↓

CustKey	CustAltKey	Name	City	Current	Start	End
1	1002	Amy Alberts	Vancouver	No	1/1/2000	1/1/2012
4	1002	Amy Alberts	Toronto	Yes	1/1/2012	

CustKey	CustAltKey	Name	Cars
1	1002	Amy Alberts	0

Type 3



CustKey	CustAltKey	Name	Prior Cars	Current Cars
1	1002	Amy Alberts	0	1

## Type 4 SCD

- In Type 4 SCD all the changes are stored in a **History Table**. Latest value only saved in the original table. (Can be done by Triggers)