ABSTRACT:
Many companies use a data warehouse successfully for storing their historical data to the data warehouse than a database. As database is going to be used for the store operational data rather than using a data warehouse. So, in the data warehouse also how the data is going to be managed for the current and historical data is also a complex problem. Many times the past historical data is required to make out a decision on the particular point of view and that’s why to manage a historical as well as current data in to the database is much more important.

KEYWORDS: SCD 1, SCD 2, SCD 3, SCD 4, SCD 6, Slowly Changing Dimensions

INTRODUCTION
The most current data’s direct access is the biggest problem in the many data warehouse design. The complex issue is to keep track of the dimensional table’s columns over time and how to resolve it. Standard data warehouse modeling techniques is used by the initial iteration of these dimensional models. Slowly changing dimensions type 2 or 3 is address to resolve above mentioned problem.

SCD stands for slowly changing dimensions
SCD1: SCD1 keeps the most recent data in to the target. It does not keep history. It keeps only current data
SCD2: SCD2 keeps full history in to the target. Any updates done in the OLTP database, it inserts the new record in the target
SCD3: SCD3 keeps current data and previous data in to the target. i.e., it keeps partial history

DIFFERENT SCD TYPES
For most people in the BI community there are nominally 3 different types of SCD, each with a different method of handling change

- **Type 1 SCD** where change is handled by overwriting old data, hence you always get the latest view.

Type 1 SCD DW architecture applies when no history is kept in the database. The new, changed data simply overwrites old entries. This approach is used quite often with data which change over the time and it is caused by correcting data quality errors (misspells, data consolidations, trimming spaces, language specific characters).

Type 1 SCD is easy to maintain and used mainly when losing the ability to track the old history is not an issue.

- **The Type 2 SCD** where change is stored using a combination of effective dates and new rows of data.

In the Type 2 SCD model the whole history is stored in the database. An additional dimension record is created and the segmenting between the old record values and the new (current) value is easy to extract and the history is clear. The fields 'effective date' and 'current indicator' are very often used in that dimension.

- **Type 3 SCD** where change is handled by adding a new column!

Type 3 SCD - only the information about a previous value of a dimension is written into the database. An 'old' or 'previous' column is created which stores the immediate previous attribute. In Type 3 SCD users are able to describe history immediately and can report both forward and backward from the change. However, that model can't track all historical changes, such as when a dimension changes twice or more. It would require creating next columns to store historical data and could make the whole data warehouse schema very complex.

Wikipedia also defines Types 0, 4 and 6

- **Type 0 SCD** -where nothing is done at all.

Type 0 SCD is not used frequently, as it is classified as when no effort has been made to deal with the changing dimensions issues. So, some dimension data may be overwritten and other may stay unchanged over the time and it can result in confusing end-users.

- **Type 4 SCD** – where “history tables” are used to store old records

Type 4 SCD idea is to store all historical changes in a separate historical data table for each of the dimensions.

- **Type 6** – which is a hybrid (invented by Kimball) of Types 1 + 2 + 3 = 6

EXPLANATION OF THE SCD 0 TO SCD 3
Comparison is going to be done in between the previously loaded production key of the dimensional table and the newly loaded cycle data. Changes to relevant columns in the new load data for matching production keys are loaded with new surrogate key assignments.

In this if the operational meta data columns are incorporated into the design of the table, newly assigned surrogate keys can be identified as current while records previously loaded for a matching production key can be flagged as historical. The technique used here is solved half of the problem in the business requirements. The use of the surrogate keys on both table , fact table and dimension table, make easy to work on the queries run against dimension table using the SCD 2 method will automatically associate the proper view of history to the corresponding fact table row through use of the surrogate keys.

Structured query language (SQL) queries would produce fact information grouped, over time, according to the number iterations the production key of the dimensions had gone through. This technique is not as practical when the business requirement is to ignore the historical changes made to a dimension...
table and show all facts associated with the most current view of the production key. The operational Meta data flag on the table can easily identified the current rows in the dimensions and ignore other view or other historical data. But, the fact table’s rows are still associated to the historical surrogate keys on the dimension. Simply constraining the most current dimension rows and retrieving fact table information would result in incomplete or incorrectly interrupted reporting results.

**SLOWLY CHANGING DIMENSION TYPE 1 (SCD TYPE 1)**

Slowly Changing Dimension Type 1 does not maintain history data and overrides the old data with new data and therefore does not track historical data at all. This is most appropriate when you don’t need to track the history of dimension and just want to update the correcting certain types of data errors, such as the spelling of a name or a date.

e.g. Consider SCD of Company.

<table>
<thead>
<tr>
<th>Company ID</th>
<th>Company Name</th>
<th>Company Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC Supply Company</td>
<td>Maharashtra</td>
</tr>
</tbody>
</table>

If Company moves to new location and since its SCD Type 1, we can simply update the table to overwrite data.

<table>
<thead>
<tr>
<th>Company ID</th>
<th>Company Name</th>
<th>Company Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC Supply Company</td>
<td>Karnataka</td>
</tr>
</tbody>
</table>

The big disadvantage to this method of managing SCDs is that there is no historical record kept in the data warehouse. However an advantage to this is that they are easy to maintain.

**SLOWLY CHANGING DIMENSION TYPE 2**

SCD 2 is used when need to model a dimensional table when the changes to relevant columns required to be monitored over time and time. Basically this technique is depend on the use of the production or natural keys in the dimensional table that may not going to be changed over time. New surrogate keys are assigned to the dimensional table when changes to the relevant columns are find out while batch load cycle.

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 newer record gets its own primary key.

**Example: 1)**

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

<table>
<thead>
<tr>
<th>Customer Key</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Christina</td>
<td>Illinois</td>
</tr>
</tbody>
</table>

After Christina moved from Illinois to California, we add the new information as a new row into the table:

<table>
<thead>
<tr>
<th>Customer Key</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003</td>
<td>Christina</td>
<td>Illinois</td>
</tr>
<tr>
<td>1005</td>
<td>Christina</td>
<td>California</td>
</tr>
</tbody>
</table>

**ADVANTAGES:**

This allows us to accurately keep all historical information.

**DISADVANTAGES:**

- 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.
- This necessarily complicates the ETL process.
- Maintaining huge history with SCD 2 can make your data warehouse huge and it should be used after evaluating business use cases and data requirement for reporting.

SCD 2 ETL transformations are bit complicated to develop however now a day’s most of the commercial tool are coming up with wizards to develop SCD Type 2

**EXAMPLE: 2) slowly Changing Dimension Type 2**

Slowly Changing Dimension Type 2 is used for tracking historical data by maintaining a version of records. CD type 2 tracks the data history by creating multiple versions of records and using either date or flag to identify the active record. This method allows tracking any number of histories as each time new record is inserted in table with its version identifier.

Let’s take an example as we discussed earlier in Type 1

<table>
<thead>
<tr>
<th>Company ID</th>
<th>Company Name</th>
<th>Company Location</th>
<th>is_active</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC</td>
<td>Maharashtra</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ABC</td>
<td>Karnataka</td>
<td>1</td>
</tr>
</tbody>
</table>

In the same example, if the Company moves to Karnataka, the table would look like this:

<table>
<thead>
<tr>
<th>Company ID</th>
<th>Company Name</th>
<th>Company Location</th>
<th>Start_Date</th>
<th>End_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC</td>
<td>Maharashtra</td>
<td>1-Jan-00</td>
<td>22-Dec-04</td>
</tr>
<tr>
<td>2</td>
<td>ABC</td>
<td>Karnataka</td>
<td>22-Dec-04</td>
<td></td>
</tr>
</tbody>
</table>

In above example is_active is the version identifier which helps to identify the current Location, however instead of using flag dates can be used which can also help to find out number of period the record was active.

NULL end date can be used to identify the current active record, instead of using NULL if some standard date is used e.g. 1111-11-11 this column can be used for an Index

**Slowly Changing Dimension Type 3 (SCD Type 3)**

tracks the history my adding separate columns for each version of record. As opposed to Slowly Changing Dimension Type 2 (SCD Type 2) where we can maintain any number of history of record SCD Type 3 allows limited history of record as adding a separate column for each version is not a good practice. Generally it allows maintaining history up to 2 levels.

**EXAMPLE: 1)**

<table>
<thead>
<tr>
<th>CompanyID</th>
<th>Company Name</th>
<th>Company Last Location</th>
<th>Company Current Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ABC</td>
<td>Maharashtra</td>
<td>1-Jan-00</td>
</tr>
<tr>
<td>2</td>
<td>ABC</td>
<td>Karnataka</td>
<td>22-Dec-04</td>
</tr>
</tbody>
</table>

After looking at above example it’s clear that with SCD Type 3 we cannot track the history of Company movement if it moves again to new Location.

**CONCLUSION:**

Tracking of the different records in the historical database is critical. But still we can maintain it as there are numbers of SCD types are used to carry out solution for the problems in the business industries to maintain different problems associated with the historical data changes and keep track on that and how to store them. Above different methods from
SCD 0 to SCD 3 are generally used for the keeping track of historical updated data.

REFERENCES:
1. Implementing a Data Warehouse by Bruce Ullrey
2. Building the Data Warehouse by W. H. Inmon
3. Data Warehousing 101 by Arshad Khan
4. Agile Data Warehousing: Delivering World-Class Business Intelligence Systems Using Scrum and Xp