**Slowly Changing Dimension**

Slowly Changing Dimension are Dimension which changes Slowly with the time,rather than changing in regular interval of time base in data warehouse

Data warehouse need to keep track on all the dimensions attributes in order to report the historical data

For example: In bank data base there are many people account whoes address might chnage with time and bank need to keep track of that hence we call address as SCD

There are many approaches how to deal with SCD. The most popular are:

* **Type 0** - The passive method
* **Type 1** - Overwriting the old value
* **Type 2** - Creating a new additional record
* **Type 3** - Adding a new column
* **Type 4** - Using historical table
* **Type 6** - Combine approaches of types 1,2,3 (1+2+3=6)

Passive method:

In passive method we try to keep all the dimension same as they where at time of insertion in the database (there are some dimension which need not to get updated with the time comes under passive method/)

Type1:

In Type 1 Types of handling SCD we simply Overwrite the exisiting data The old dimension value is simply overwritten be the new one. This type is easy to maintain and is often use for data which changes are caused by processing corrections(e.g. removal special characters, correcting spelling errors).

Before the change:

|  |  |  |
| --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Customer\_Type** |
| 1 | Cust\_1 | Corporate |

After the change:

|  |  |  |
| --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Customer\_Type** |
| 1 | Cust\_1 | Retail |

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.

Before the change:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Customer\_Type** | **Start\_Date** | **End\_Date** | **Current\_Flag** |
| 1 | Cust\_1 | Corporate | 22-07-2010 | 31-12-9999 | Y |

After the change:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Customer\_Type** | **Start\_Date** | **End\_Date** | **Current\_Flag** |
| 1 | Cust\_1 | Corporate | 22-07-2010 | 17-05-2012 | N |
| 2 | Cust\_1 | Retail | 18-05-2012 | 31-12-9999 | Y |

**Type 3** - Adding a new column. In this type usually only the current and previous value of dimension is kept in the database. The new value is loaded into 'current/new' column and the old one into 'old/previous' column. Generally speaking the history is limited to the number of column created for storing historical data. This is the least commonly needed technique.

Before the change:

|  |  |  |  |
| --- | --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Current\_Type** | **Previous\_Type** |
| 1 | Cust\_1 | Corporate | Corporate |

After the change:

|  |  |  |  |
| --- | --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Current\_Type** | **Previous\_Type** |
| 1 | Cust\_1 | Retail | Corporate |

**Type 4** - Using historical table. In this method a separate historical table is used to track all dimension's attribute historical changes for each of the dimension. The 'main' dimension table keeps only the current data e.g. customer and customer\_history tables.

Current table:

|  |  |  |
| --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Customer\_Type** |
| 1 | Cust\_1 | Corporate |

Historical table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Customer\_Type** | **Start\_Date** | **End\_Date** |
| 1 | Cust\_1 | Retail | 01-01-2010 | 21-07-2010 |
| 1 | Cust\_1 | Oher | 22-07-2010 | 17-05-2012 |
| 1 | Cust\_1 | Corporate | 18-05-2012 | 31-12-9999 |

**Type 6** - Combine approaches of types 1,2,3 (1+2+3=6). In this type we have in dimension table such additional columns as:

* current\_type - for keeping current value of the attribute. All history records for given item of attribute have the same current value.
* historical\_type - for keeping historical value of the attribute. All history records for given item of attribute could have different values.
* start\_date - for keeping start date of 'effective date' of attribute's history.
* end\_date - for keeping end date of 'effective date' of attribute's history.
* current\_flag - for keeping information about the most recent record.

In this method to capture attribute change we add a new record as in type 2. The current\_type information is overwritten with the new one as in type 1. We store the history in a historical\_column as in type 3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customer\_ID** | **Customer\_Name** | **Current\_Type** | **Historical\_Type** | **Start\_Date** | **End\_Date** | **Current\_Flag** |
| 1 | Cust\_1 | Corporate | Retail | 01-01-2010 | 21-07-2010 | N |
| 2 | Cust\_1 | Corporate | Other | 22-07-2010 | 17-05-2012 | N |
| 3 | Cust\_1 | Corporate | Corporate | 18-05-2012 | 31-12-9999 | Y |