+1 0 More Next Blog» Create Blog Sign In

My Datastage Notes

SCD

WHAT IS SCD IN DATASTAGE ? TYPES OF SCD IN DATASTAGE.

SCD's are nothing but Slowly changing dimension.

Scd's are the dimensions that have the data that changes slowly. Rather than

changing in a time period. That is a regular schedule.

The Scd's are performed mainly into three types.

They are

Type-1 SCD

Type-2 SCD

Type-3 SCD

Type -1 SCD: In the type -1 SCD methodology, it will overwrites the older data

(Records) with the new data (Records) and therefore it will not maintain the

historical information.

This will used for the correcting the spellings of names, and for small updates of

customers.

TYpe -2 SCD: In the Type-2 SCS methodology, it will tracks the complete historical

information by creating the multilple records for the given natural key (Primary

key) in the dimension tables with a separate surrogate keys or a different

version numbers. We have a unlimited historical data preservation, as a new

record is inserted each time a change is made.

Here we use differet type of options inorder to track the historical data of

customers like

a) Active flag

b) Date functions

c) Version Numbers

d) Surrogate Keys

We use this to track all the historical data of the customer.

According to our input, we use required function to track.

Type-3 SCD: In the Type-2 SCD, it will maintain the partial historical

information.

HOW TO USE TYPE -2 SCD IN DATASTAGE

SCD'S is nothing but Slowly changing Dimensions.

Slowly Changing Dimensions are the dimensions that have the data that change slowly rather than changing in a time period, i.e regular schedule.

The most common Slowly Changing Dimensions are three types.
They are Type -1 , Type -2 , Type -3 SCD's

Type-2 SCD:-- The Type-2 methodology tracks the Complete Historical information by creating the multiple records for a given natural keys in the dimension tables with the separate surrogate keys or different version numbers.

And we have unlimited history preservation as every time new record is inserted each time a change is made.

SLOWLY CHANGING DIMENSIONS (SCD) - TYPES | DATA WAREHOUSE

Slowly Changing Dimensions: Slowly changing dimensions are the dimensions in which the data changes slowly, rather than changing regularly on a time basis.

For example, you may have a customer dimension in a retail domain. Let say the customer is in India and every month he does some shopping. Now creating the sales report for the customers is easy. Now assume that the customer is transferred to United States and he does shopping there. How to record such a change in your customer dimension?

You could sum or average the sales done by the customers. In this case you won't get the exact comparison of the sales done by the customers. As the customer salary is increased after the transfer, he/she might do more shopping in United States compared to in India. If you sum the total sales, then the sales done by the customer might look stronger even if it is good. You can create a second customer record and treat the transferred customer as the new customer. However this will create problems too.

Handling these issues involves SCD management methodologies which referred to as Type 1 to Type 3. The different types of slowly changing dimensions are explained in detail below.

SCD Type 1: SCD type 1 methodology is used when there is no need to store historical data in the dimension table. This method overwrites the old data in the dimension table with the new data. It is used to correct data errors in the dimension.

As an example, i have the customer table with the below data.

```
surrogate_key customer_id customer_name Location
-----------------------------------------------
1             1           Marspton       Illions
```

Here the customer name is misspelt. It should be Marston instead of Marspton. If you use type1 method, it just simply overwrites the data. The data in the updated table will be.

```
surrogate_key customer_id customer_name Location
-----------------------------------------------
1             1           Marston       Illions
```

The advantage of type1 is ease of maintenance and less space occupied. The disadvantage is that there is no historical data kept in the data warehouse.

SCD Type 3: In type 3 method, only the current status and previous status of the row is maintained in the table. To track these changes two separate columns are created in the table. The customer dimension table in the type 3 method will look as

```
surrogate_key customer_id customer_name Current_Location previous_locat
ion
--------------------------------------------------------------------------
---
1             1           Marston       Illions          NULL
```

Let say, the customer moves from Illions to Seattle and the updated table will look as

```
surrogate_key customer_id customer_name Current_Location previous_locat
ion
--------------------------------------------------------------------------
---
1             1           Marston       Seattle          Illions
```

Now again if the customer moves from seattle to NewYork, then the updated table will be

```
surrogate_key customer_id customer_name Current_Location previous_locat
ion
--------------------------------------------------------------------------
---
1             1           Marston       NewYork          Seattle
```

The type 3 method will have limited history and it depends on the number of columns you create.

SCD Type 2: SCD type 2 stores the entire history the data in the dimension table. With type 2 we can store unlimited history in the dimension table. In type 2, you can store the data in three different ways. They are

- Versioning
- Flagging

- Effective Date

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.

As an example, let's use the same example of customer who changes the location. Initially the customer is in Illions location and the data in dimension table will look as.

```
surrogate_key customer_id customer_name Location Version
-----------------------------------------------------------
1             1           Marston       Illions  1
```

The customer moves from Illions to Seattle and the version number will be incremented. The dimension table will look as

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

Now again if the customer is moved to another location, a new record will be inserted into the dimension table with the next version number.

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

Now for the first time, the customer dimension will look as.

```
surrogate_key customer_id customer_name Location flag
-----------------------------------------------------------
1             1           Marston       Illions  1
```

Now when the customer moves to a new location, the old records will be updated with flag value as 0 and the latest record will have the flag value as 1.

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

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-20
11
2             1           Marston       Seattle  21-Feb-2011  NULL
```

The NULL in the End_Date indicates the current version of the data and the remaining records indicate the past data.

SCD-2 Implementation in Datastage:

Slowly changing dimension Type 2 is a model where 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 and the fact table usually stores dimension key and version number.
SCD 2 implementation in Datastage
The job described and depicted below shows how to implement SCD Type 2 in Datastage. It is one of many possible designs which can implement this dimension.
For this example, we will use a table with customers data (it's name is D_CUSTOMER_SCD2) which has the following structure and data:
D_CUSTOMER dimension table before loading

Datastage SCD2 job design

The most important facts and stages of the CUST_SCD2 job processing:
• The dimension table with customers is refreshed daily and one of the data sources is a text file. For the purpose of this example the CUST_ID=ETIMAA5 differs from the one stored in the database and it is the only record with changed data. It has the following structure and data:
SCD 2 - Customers file extract:

• There is a hashed file (Hash_NewCust) which handles a lookup of the new data coming from

the text file.
• A T001_Lookups transformer does a lookup into a hashed file and maps new and old values to separate columns.
SCD 2 lookup transformer

• A T002_Check_Discrepacies_exist transformer compares old and new values of records and passes through only records that differ.
SCD 2 check discrepancies transformer

• A T003 transformer handles the UPDATE and INSERT actions of a record. The old record is updated with current indictator flag set to no and the new record is inserted with current indictator flag set to yes, increased record version by 1 and the current date.
SCD 2 insert-update record transformer

• ODBC Update stage (O_DW_Customers_SCD2_Upd) - update action 'Update existing rows only' and the selected key columns are CUST_ID and REC_VERSION so they will appear in the constructed where part of an SQL statement.
• ODBC Insert stage (O_DW_Customers_SCD2_Ins) - insert action 'insert rows without clearing' and the key column is CUST_ID.
D_CUSTOMER dimension table after Datawarehouse refresh

No comments:

Post a Comment

Home

Subscribe to: Posts (Atom)

Manohar. Simple template. Powered by Blogger.