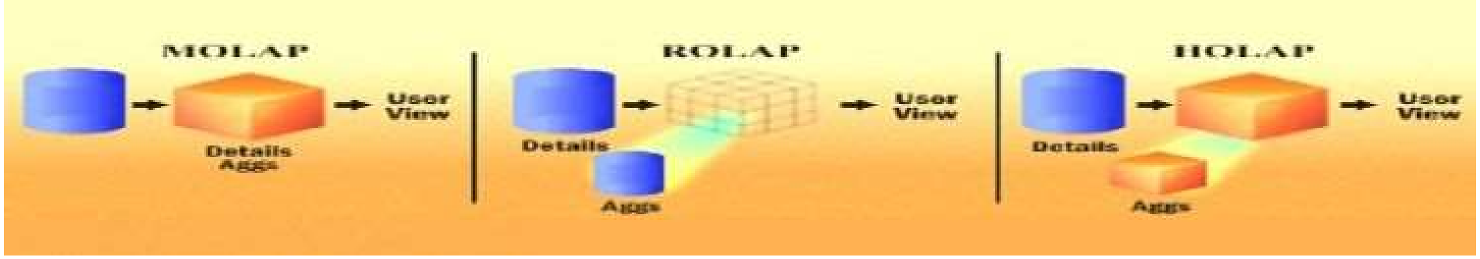
Data Warehouse po Polsku

Datawarehouse4u.Info



Data Warehouse info

Data Warehouse
OLTP vs. OLAP
What is Business Intelligence
Business Intelligence tools
ETL tools
ETL process
Magic Quadrant for Business Intelligence
Magic Quadrant for Data Integration Tools
DW Database Management Systems
BI market consolidation
Data Warehouse Schema Architecture
Slowly Changing Dimensions (SCD)
Magic Quadrant for Data Quality Tools
Business Intelligence Platforms
Teradata Certification

News

News
Archive

Recently added reviews

Talend - Talend Open Studio
Teradata Certification
QlikView
News

Contact Section

You may contact us by email:
datawarehouse4u.info[at]gmail.com

What are Slowly Changing Dimensions?

Slowly Changing Dimensions (SCD) - dimensions that change slowly over time, rather than changing on regular schedule, time-base. In Data Warehouse there is a need to track changes in dimension attributes in order to report historical data. In other words, implementing one of the SCD types should enable users assigning proper dimension's attribute value for given date. Example of such dimensions could be: customer, geography, employee.

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)

Type 0 - The passive method. In this method no special action is performed upon dimensional changes. Some dimension data can remain the same as it was first time inserted, others may be overwritten.

Type 1 - Overwriting the old value. In this method no history of dimension changes is kept in the database. 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

Type 2 - Creating a new additional record. In this methodology all history of dimension changes is kept in the database. You capture attribute change by adding a new row with a new surrogate key to the dimension table. Both the prior and new rows contain as attributes the natural key(or other durable identifier). Also 'effective date' and 'current indicator' columns are used in this method. There could be only one record with current indicator set to 'Y'. For 'effective date' columns, i.e. start_date and end_date, the end_date for current record usually is set to value 9999-12-31. Introducing 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 could be added in the future.

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 techinque.

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-	Y

					9999	