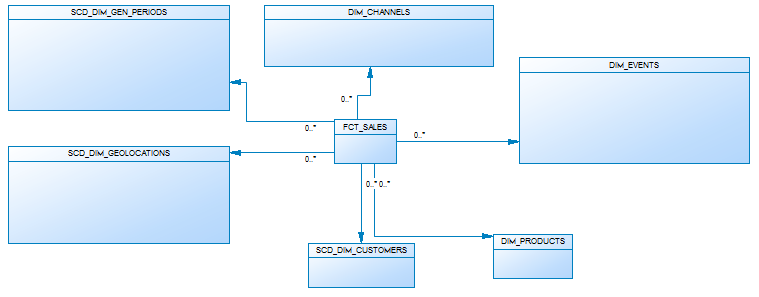


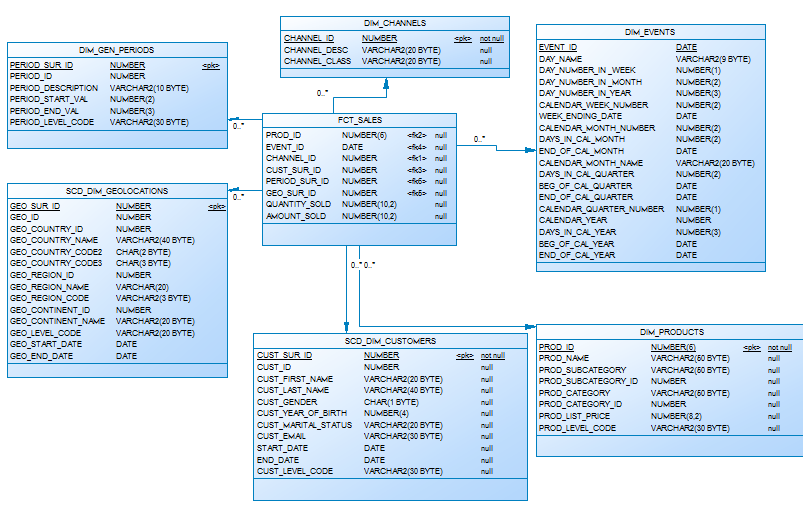
Electroforce Business Proposal

# DWH Solution Concept

## Star Logical Diagram



## Star Physical diagram



**Dimensions Types Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | DW – Merged Dimensions | Descriptions |
| DIM\_GEN\_TIME | SCD1 | BIG | DW.T\_DAYS, DW.T\_WEEKS, DW.T\_MONTHS, DW.T\_QUARTERS,  DW.T\_YEARS | Dimension describe time parameters of the business |
| SCD\_DIM\_GEOLOCATIONS | SCD2 | SMALL | DW.T\_COUNTRIES  DW.T\_CNTR\_GROUPS  DW.T\_CNTR\_SUB\_GROUPS  DW.LC\_CNTR\_GROUPS  DW.T\_GEO\_TYPES  DW.T\_GEO\_SYSTEMS  DW.LC\_GEO\_SYSTEMS  DW.T\_GEO\_PARTS  DW.T\_GEO\_REGIONS  DW.T\_GEO\_OBJECTS  DW.T\_CNTR\_GROUP\_SYSTEMS  DW. LC\_CNTR\_GROUP\_SYSTEMS  DW.LC\_CNTR\_SUB\_GROUPS  DW.LC\_ GEO\_PARTS  DW.LC\_COUNTRIES  DW.LC\_ GEO\_REGIONS | Describe locations of sales |
| SCD\_DIM\_CUSTOMERS | SCD2 | BIG | DW.T\_CUSTOMERS  DW.T\_GENDER  DW.T\_MARITAL\_STATUS  DW.T\_EMAIL | Describes customers |
| DIM\_CHANNELS | SCD1 | SMALL | DW.T\_CHANNEL\_DESC  DW.T\_CHANNEL\_CLASS | Describe channels of sales |
| SCD\_DIM\_GEN\_PERIOD | SCD2 | SMALL | DW.T\_PERIOD\_DESC  DW.T\_PER\_START  DW.T\_PER\_END  DW.T\_LEVEL\_CODE | Describe age period of customers. |
| DIM\_PRODUCTS | SCD1 | BIG | DW.T\_PRODUCT\_DESC  DW.T\_PROD\_SUBCAT  DW.T\_PROD\_CAT | Describe products of the company |

**Dimensions** **Hierarchies**

**DIM\_GEN\_TIME:**

**Hierarchy DAY-WEEK-MONTH-YEAR**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| DAYs | DAY | Store day at the calendar | DAY\_ID |
| WEEKs | WEEK | Store weeks at the calendar year | WEEK\_ID |
| MONTHs | MONTH | Store months at the calendar year | MONTH\_ID |
| YEARs | YEAR | Store years at the calendar year | YEAR\_ID |
|  |  |  |  |

**Hierarchy DAY--MONTH- QUARTER -YEAR**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| DAYs | DAY | Store day at the calendar year | DAY\_ID |
| MONTHSs | MONTH | Store months at the calendar year | WEEK\_ID |
| QUARTERs | QUARTER | Store quarters at the calendar year | QUARTER\_ID |
| YEARs | YEAR | Store years at the calendar year | YEAR\_ID |
|  |  |  |  |

**DIM\_PRODUCTS:**

**Hierarchy PRODUCTS –-SUBCATEGORY-- CATEGORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| PRODUCTS | PROD\_NAME | Store products for each category. | PROD\_ID |
| SUBCATEGORIES | PROD\_SUBCATEGORY | Store product subcategories for each category. | PROD\_SUBCATEGORY\_ID |
| CATEGORIES | PROD\_CATEGORY | Store product categories. | PROD\_CATEGORY\_ID |
|  |  |  |  |

**DIM\_GEO LOCATIONS:**

**Hierarchy COUNTRY –-SUBREGION--REGION**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| COUNTRIES | GEO\_COUNTRY\_NAME | Store countries for each region. | GEO\_COUNTRY\_ID |
| SUBREGIONS | GEO\_REGION\_NAME | Store sub regions for each region. | GEO\_REGION\_ID |
| REGIONS | GEO\_CONTINENT\_NAME | Store regions of the world. | GEO\_CONTINENT\_ID |
|  |  |  |  |

**Users, objects and tablespaces**

Create user for each object. The name of the user look like U\_”name of the object”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Level Type | Object Name | Tablespace | Description | Privileges |
| Storage level  SA\_\* | SA\_CUSTOMERS | ts\_sa\_customers\_data\_01 | Loading from Flat file storage system. Contains Customer information |  |
| SA\_SALES | ts\_sa\_sales\_data\_01 | Loading from Flat storage system. Contains Sales information |  |
|  |  |  |  |
| DW - Cleansing Level | DW\_CL | ts\_dw\_cl\_data\_01 | Clean raw data inserted from files and insert it into tables in the 3-rd normal form. | U\_DW\_CL clean raw data inserted from files and insert it into tables in the 3-rd normal form. To make this operations user U\_DW\_CL should have grants to create tables with raw data, to connect and select information from resources and load information, and create views, to create result views with clean information. |
| DW – Level | DW | ts\_dw\_data\_01 | Store information in the 3-rd normal form. |  |
| DW– Prepare Star Cleansing Level | SAL\_DW\_CL | ts\_sal\_dw\_cl\_data\_01 | Create views, which consist of merged tables from DW Level. | U\_SAL\_CL\_DW should have grants to connect and select information from DW Level, create tables, where he will make changes, and create any view to make the clean view visible to next level users, who will have grants on it. |
| STAR - Cleansing | SAL\_CL | ts\_sal\_cl\_data\_01 | Delete from views, from previous level information, which don’t touch our business analysis question. | U\_SAL\_CL should have grants to connect and select information from the view, which was created on U\_SAL\_CL\_DW level, create tables to insert into them the information, which relative only with future analytic plan and the grant to create views to make public clean information for future levels. |
| STAR – Level | FCT\_SALES | ts\_star\_fct\_data\_01 | Store information about facts |  |
| DIM\_TABLES | ts\_star\_dim\_data\_01 | Store information about dimensions |  |

Dataflow Diagram



**PARTITIONING**

The query execution speed, which works with fact table, can be increased with partitioning.

As business plan of the star is to show quickly the information about sales by channels and in section of time, so the composite partitioning should be done.

It will be used partitioning by quarters and every quarter will be partitioned by channels of sales.

Range Partition by EVENT\_ID. The sales are divided by quarters.

|  |  |  |
| --- | --- | --- |
| Quarter, number | Start date, dd/mm | End date, dd/mm |
| 1 | 01/01 | 31/03 |
| 2 | 01/04 | 30/06 |
| 3 | 01/07 | 30/09 |
| 4 | 01/10 | 31/12 |

Hash partition of every quarter by channels. Number of sub partitions is 4, because it should be the degree of 2 and because we have 6 channels.

The result table

|  |  |  |  |
| --- | --- | --- | --- |
| Sales | Quarter 1 | Hash\_1 | facts |
| facts |
| facts |
| … |
| Hash\_2 | …. |
| Hash\_3 |  |
| Hash\_4 |  |
| Quarter 2 | Hash\_1 |  |
| Hash\_2 |  |
| Hash\_3 |  |
| Hash\_4 |  |
| Quarter 3 | Hash\_1 |  |
| Hash\_2 |  |
| Hash\_3 |  |
| Hash\_4 |  |
| Quarter 4 | Hash\_1 |  |
| Hash\_2 |  |
| Hash\_3 |  |
| Hash\_4 |  |

# Strategy of Parallel execution

In my data warehouse system, large tables FCT\_Sales, SCD\_DIM\_PRODUCTS and SCD\_DIM\_CUSTOMERS need to be refreshed (updated) periodically with new or modified data from the production system. I can do this efficiently by using parallel DML combined with updatable join views. It also increase the speed of DDL.

The data that needs to be refreshed will be loaded into table before starting the refresh process. This table will contain either new rows or rows that have been updated since the last refresh of the data warehouse. After all changes will be done, the data will be pushed to public views.