

*AD Store Contracts*

*Buissness Intelegence Proposal*

Version Number: 1.0

Version Date: 03/29/2012

General Information

|  |  |
| --- | --- |
| **Submission Date** | *03/29/2012* |
| **Requested By** | *Kyril Bulcha* |
| **Business Owner** | *Aliaksandr Dabradzei* |
| **Contact Info.** | *Aliaksandr\_Dabradzei@epam.com* |
|  |  |

# Overview

## Business Background

# AlexDobrodey Store (AD Store) is a small, Minsk-based automobile racing parts brick and mortar and mail order organization. AD Store will remain small in order to offer unprecedented customer attention. AlexDobrodey Store will become the premier destination for entry level and novice racing parts.

# AD Store needs system of tracing signed contracts all over the world. Also this system will keep information about customers.

# DWH Solution Concept

## Logical Diagram



## 

## Physical diagram



# Dimensions

## Dimension Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Size | DW – Merged Dimensions | Descriptions |
| DIM\_ TIME\_DAY | SCD1 | BIG | DW.T\_DAYS, DW.T\_WEEKS, DW.T\_MONTHS, DW.T\_QUARTERS,  DW.T\_YEARS | It is a specific type of dimension. Appears at the DWR and contains all the time since the introduction of the project until its completion. |
| DIM\_CURRENCY | SCD1 | SMALL | DW.T\_CURRENCY | Small dimension with available currencies |
| DIM\_CUSTOMER\_SCD | SCD2 | BIG | DW.T\_CUSTOMERS,  DW.T\_CUST\_ADDRESSES,  DW.T\_CUST\_TECHN\_INF | Big dimension with customer references. |
| DIM\_DEPARTMENT | SCD1 | SMALL | DW.T\_DEPARTMENTS | Small dimension with departments’ references. |
| DIM\_GEN\_PERIODS | SCD1 | SMALL | DW.T\_PERIODS | Small dimension with available periods. |
| DIM\_GEO\_LOCATIONS | SCD1 | BIG | DW.T\_COUNTRIES  DW.T\_CNTR\_GROUPS  DW.T\_CNTR\_SUB\_GROUPS  DW.T\_GEO\_TYPES  DW.T\_GEO\_SYSTEMS  DW.T\_GEO\_PARTS  DW.T\_GEO\_REGIONS  DW.T\_GEO\_OBJECTS  DW.T\_CNTR\_GROUP\_SYSTEMS | Dimension, which stores available geo-locations. |
| DIM\_PRODUCT\_SCD | SCD2 | BIG | DW.PRODUCTS,  DW.PRODUCT\_SUBCAT  DW.PRODUCT\_CAT  DW.PRODUCT\_SALES | Big dimension with products references. |

### Dimension Hierarchies

## Hierarchies

DIM\_GEN\_TIME:

Hierarchy DAY – WEEK – MONTH – QUARTER – YEAR

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

DIM\_PRODUCTS\_SCD:

Hierarchy PRODUCTS – SUBCATEGORY – CATEGORY

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

DIM\_GEO LOCATIONS:

Hierarchy COUNTRY – REGION – CONTINENT

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| COUNTRIES | COUNTRY\_NAME | Store all countries for each region. | COUNTRY\_ID |
| REGIONs | COUNTRY\_REGION | Store all regions of the world. | COUNTRY\_REGION\_ID |
| CONTINENTs | COUNTRY\_CONTINENT | Store all continents of the world. | COUNTRY\_CONTINENT\_ID |

# Facts

## Facts Aggregations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Code | Table Name | Additive | Descriptions |
| Counts amount sold products by day | AMOUNT\_SOLD | FCT\_CONTRACTS\_DD | + | Calculate distinct values of Contracts at the *event\_dt* period. |
| Counts quantity sold products by day | QUANTITY\_SOLD | FCT\_ CONTRACTS\_DD | + | Calculate distinct values of Contracts Products at the *event\_dt* period. |

# Data Warehouse Architecture

## Tablespaces

|  |  |  |  |
| --- | --- | --- | --- |
| Level Type | Object Name | Tablespace | Description |
| Storage level  SA\_\* | SA\_CUSTOMERS | ts\_sa\_customers\_data\_01 | Loading from xls storage system. Contains Customer information |
| SA\_PRODUCTS | ts\_sa\_products\_data\_01 | Loading from xls storage system. Contains Products information |
| SA\_CONTRACTS | ts\_sa\_contracts\_data\_01 | Loading from xls storage system. Contains Contracts information |
| SA\_DEPARTMENTS | ts\_sa\_departments\_data\_01 | Loading from flat file storage system. Contains Departments information |
| SA\_CURRENCY | ts\_sa\_currency\_data\_01 | Loading from flat file storage system. Contains Currency information |
| DW - Cleansing Level | DW\_CL\_REFERENCES | ts\_dw\_cl\_ref\_01 | Cleansing data for references |
| DW\_CL\_DATA | ts\_dw\_cl\_data\_01 | Cleansing data for facts |
|  | DW\_CL\_PKG | ts\_dw\_cl\_pkg\_01 | Packages for extracting and loading data. |
| DW – Level | DW\_DATA | ts\_dw\_data\_01 | Storage for aligned facts |
| DW\_IDX | ts\_dw\_idx\_01 | Storage for indexes |
| DW\_REFERENCES | ts\_dw\_references\_01 | Storage for dimension data |
| DW– Prepare Star Cleansing Level | SAL\_DW\_CL | ts\_sal\_dw\_cl\_01 | Data for prepare star. |
| STAR - Cleansing | SAL\_CL\_DATA | ts\_sal\_cl\_data\_01 | Cleansing tables for facts |
| SAL\_CL\_REFS | ts\_sal\_cl\_refs\_01 | Cleansing tables for references |
| STAR – Level | SAL\_DATA | ts\_sal\_data\_01 | Storage for facts in star |
| SAL\_IDX | ts\_sal\_idx\_01 | Storage for indexes in star |
| SAL\_REFS | ts\_sal\_refs\_01 | Storage for references in star |
| DM\_CUSTOMERS | ts\_dm\_customers\_01 | Data Mart storage for analytics by customers |
| DM\_PRODUCTS | ts\_dm\_products\_01 | Data Mart storage for analytics by products |
| DM\_LOCATIONS | ts\_dm\_locations\_01 | Data Mart storage for analytics by locations |
| DM\_DEPARTMENTS | ts\_dm\_departments\_01 | Data Mart storage for analytics by departments |
| DM\_PERIODS | ts\_dm\_periods\_01 | Data Mart storage for analytics by periods |

## Schemas

|  |  |
| --- | --- |
| **SCHEMAs** | **Comment** |
| **SA** | |
| *u\_sa\_common* | Owner of Stage area objects |
| **DW CL AREA** | |
| *u\_dw\_cl\_common* | Owner of Data Warehouse Cleansing area packages, have rights to read *u\_sa\_common* objects and update *u\_dw\_cl\_data, u\_dw\_data*, *u\_dw\_references* data objects |
| *u\_dw\_cl\_data* | Owner of Data Warehouse Cleansing area data objects |
| **DW AREA** | |
| *u\_dw\_common* | Owner of Data Warehouse area packages, have rights to read *u\_dw\_cl\_data* objects and update *u\_dw\_data*, *u\_dw\_references* and *u\_dw\_sal\_cl\_common* data objects |
| *u\_dw\_data* | Owner of Data Warehouse area data objects |
| *u\_dw\_references* | Owner of Data Warehouse area data references objects |
| **DW SAL CL AREA** | |
| *u\_dw\_sal\_cl\_common* | Owner of Data Warehouse Star Aggregation Level Cleansing area packages, have rights to read *u\_dw\_data*, *u\_dw\_references* objects and update *u\_dw\_sal\_cl\_common* and *u\_sal\_cl\_common* data objects |
| **SAL CL AREA** | |
| *u\_sal\_cl\_common* | Owner of Star Aggregation Level Cleansing area packages, have rights to read *u\_dw\_sal\_cl\_common* data objects and update *u\_sal\_common*, *u\_sal\_data*, *u\_sal\_references* data objects |
| **SAL AREA** | |
| *u\_sal\_common* | Owner of Star Aggregation Level area packages, have rights to read *u\_sal\_cl\_common* data objects and update *u\_sal\_data*, *u\_sal\_references* and *u\_dm\_data* data objects |
| *u\_sal\_data* | Owner of Star Aggregation Level area data objects |
| *u\_sal\_references* | Owner of Star Aggregation Level area data references objects |
| **DM AREA** | |
| *u\_dm\_common* | Owner of Data Marts area packages, have rights to read *u\_sal\_data*, *u\_sal\_references* and update *u\_dm\_data* data objects |
| *u\_dm\_data* | Owner of Data Marts area data objects |

# Dataflow Diagram

# 

# Partitioning rules

I will partition facts table by *event\_dt* by years. Each partition will have 4 sub partitions by hash of *product\_surr\_id* column.

PARTITION BY RANGE(event\_dt)

INTERVAL (NUMTOYMINTERVAL(1,'YEAR'))

SUBPARTITION BY HASH(product\_surr\_id)

SUBPARTITIONS 4

( PARTITION p1 VALUES LESS THAN (to\_date('2010/01/01','yyyy/mm/dd'))

STORAGE (INITIAL 100M NEXT 50M)

(SUBPARTITION sp\_01 TABLESPACE ts\_sal\_data\_011),

(SUBPARTITION sp\_02 TABLESPACE ts\_sal\_data\_012),

(SUBPARTITION sp\_03 TABLESPACE ts\_sal\_data\_013),

(SUBPARTITION sp\_04 TABLESPACE ts\_sal\_data\_014));

# Strategy of Parallel execution

In my data warehouse system, large tables FCT\_CONTRACTS\_DD, DIM\_PRODUCTS\_SCD and DIM\_CUSTOMERS\_SCD 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. Also I can use the MERGE statement.

The data that needs to be refreshed will be loaded into a temporary 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. I could use an updatable join view with parallel UPDATE to refresh the updated rows, and I can use an anti-hash join with parallel INSERT to refresh the new rows.

All updates will be performed by update jobs.