# 

# Small Beer Ltd

# Solution Concept Document

|  |  |
| --- | --- |
| **Submission Date** | *07/25/2013* |
| **Requested By** | *Kyril Bucha* |
| **Business Owner** | *Tony Eastwood* |
| **Contact Info.** | *eastwood@sb.com* |

# Overview

Trading company engaged in the wholesale of beer from different manufacturers. Company’s business is spread among different counties. It has a number of customers and stocks in different places. According to existing strategy company serves its customers providing the opportunity to order products with different configuration.

## Business Background

The main idea of DWH is to build a system for analytic of the product sales and needs in products. The company is constantly deciding which product and how much to buy for resale. To do this, was decided to implement a system of ABC analysis. This analysis divides products in groups of interest. Each quarter company need new information about actual products. Besides, company constantly calculates others metrics, which can be realized in system.

Main entities used as business data:

1. Order

2. Order Item

3. Product

4. Product category

5. Product measure

6. Country

## Benefits

1. System is scalable and can easy take into account new stocks in different countries and new products.

# Requirements

## Business Requirements

|  |
| --- |
| Order can consist of different goods in any combination. |
| All calculations are carried out in U.S.$. |
| Data should be aggregated in different measures – countries, products, time intervals etc. |
| Order data can’t be removed – orders have been already done and registered. |
| Order data can’t be changed. |
| Product name can be updated, old name is not informative any more. |
| Product hierarchy is very stick. One product can belong only to one category. This belonging can change. |
| One product can have many configuration, so it can have different meashures. |

## Technical Requirements

|  |
| --- |
| Information in the system should be updated each hour and be available in work hours. |
| The system must be able to transfer the data into different languages. |

# Solution Sketch

## Source Tables structure

### ORDERS

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Comment |
| EVENT\_DT | **DATE** | **Date of contract conclusion** |
| ORDER\_ID | NUMBER (32) | Order unique code |
| CUSTOMER\_ID | NUMBER(32) | Code of the customer |
| CUSTOMER\_ID | NUMBER(32) | Organization made a deal |
| COUNTRY\_ID | NUMBER(32) | Country of the transaction |
| TOTAL\_SUM | NUMBER (12,2) | Order sum in U.S.$ |

### ORDER\_ITEMS

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Comment |
| ORDER\_ID | VARCHAR2 (32) | Order unique code |
| PRODUCT\_ID | NUMBER (32) | Identification of the sold product |
| QUANTITY | NUMBER(5) | Quantity of the sets of an item in the order |
| PRICE | NUMBER(12,2) | Price per one set of the item |

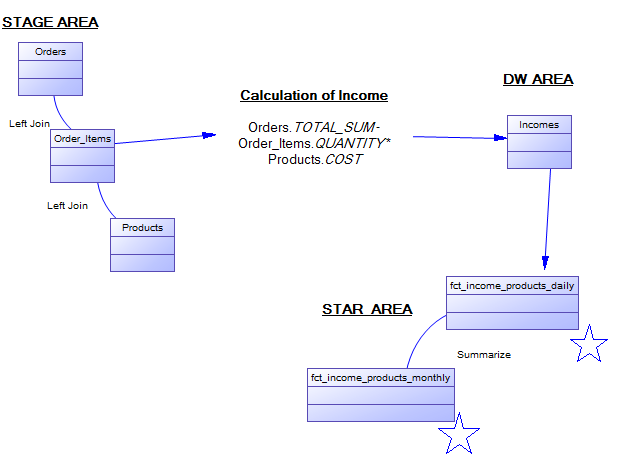
### PRODUCTS

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Comment |
| PRODUCT\_ID | NUMBER (32) | Product unique code |
| PRODUCT\_NAME | VARCHAR2 (40) | Full name of the product |
| PRODUCT\_CATEGORY | VARCHAR2 (40) | Category of a product |
| PRODUCT\_MEASURE | VARCHAR2(10) | Measure of set |
| MEASURE\_QUANTITY | NUMBER(15) | Quantity of the good in accurate measure |
| COST | NUMBER(12,2) | Cost per one set of product |

### COUNTRIES

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Comment |
| COUNTRY\_ID | NUMBER (32) | Country unique code |
| COUNTRY\_DESCRIPTION | VARCHAR2 (32) | Country Name |
| COUNTRY\_SHORT\_DESC | VARCHAR2 (5) | Country abbreviation |
| COUNTRY\_GROUP\_ID | VARCHAR2 (32) | Group code |
| COUNTRY\_GROUP\_DESC | NUMBER (32) | Group name |
| REGION | VARCHAR2 (32) | Region name |
| REGION\_CODE | NUMBER (32) | Region code |

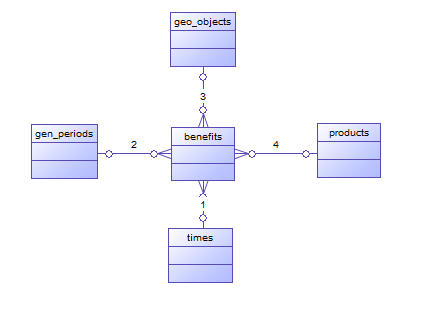
## Summarize Data Plan



# DWH Solution Concept

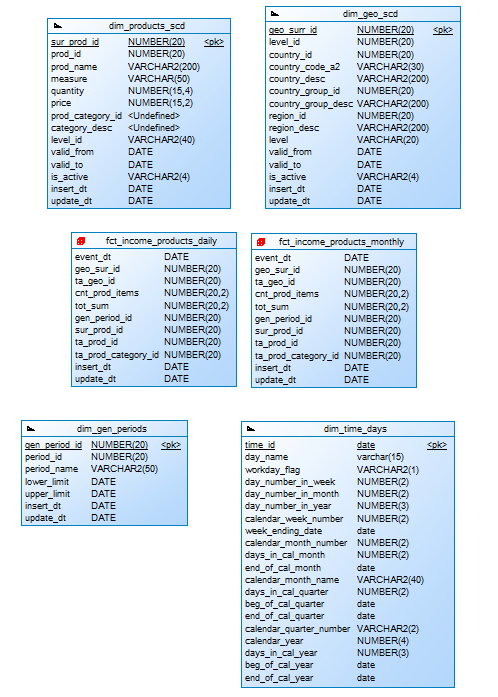
In DWH architecture will be presented layers from Data Flow Diagram. They are need for security and consistency of information. Cleansing layers are used by architects and administrators for loading and transformation of data. Other stages are used for storing data in accurate format.

## Logical Diagram



|  |  |
| --- | --- |
| Entity Name | Description |
| benefits | A fact table. Summarize the information about sold goods and income divided by this products |
| times | Dimension table. It contains the calendar with business activity rules |
| gen\_periods | Dimension table. Consists of definitions of business time periods and intervals in quality business features |
| products | Dimension table. It contain the information about products, its configurations and prices |
| geo\_objects | Dimension table. Contain the hierarchy of geographical objects. |

## Physical diagram



## Dimensions

### Dimension Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NAME | TYPE | SIZE | MERGED DIMENSIONS | DESCRIPTION |
| dim\_time\_days | SCD1 | BIG | dw.t\_days  dw.lc\_days  dw.t\_weeks  dw.t\_months  dw.lc\_months  dw.t\_quarters  dw.t\_years  dw.t\_localizations | Dimension consists of a calendar continuous list |
| dim\_geo\_scd | SCD2 | SMALL | dw.t\_countries  dw.lc\_countries  dw.t\_cntr\_groups  dw.lc\_cntr\_groups  dw.t\_geo\_regions  dw.lc\_geo\_regions  dw.t\_geo\_objects  dw.t\_geo\_types  dw.t\_geo\_object\_links  dw.t\_localizations  dw.t\_geo\_action | Dimension with geo information and hierarchy of countries, groups of countries and regions. |
| dim\_gen\_periods | SCD1 | SMALL | dw.gen\_periods  dw.lc\_gen\_periods | Dimension with information about business periods of customer’s activity |
| dim\_products\_scd | SCD2 | BIG | dw.t\_products  dw.lc\_measures  dw.lc\_prod\_categories  dw.prod\_action  dw.t\_localizations | Dimension with information about product sets, and categories. It includes the history of changing of price and names. |

### Dimension Hierarchies

DIM\_TIMES. Hierarchy :DAY-MONTH-QUARTER-YEAR

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| DAY | DAY | Store days at the calendar year | DAY\_ID |
| MONTH | MONTH | Store months at the calendar year | MONTH\_ID |
| QUARTER | QUARTER | Store quarters at the calendar year | QUARTER\_ID |
| YEAR | YEAR | Store years at the calendar year | YEAR\_ID |

DIM\_GEO. Hierarchy: COUNTRY-GROUPS\_OF\_COUNTRIES-REGIONS

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| COUNTRY | COUNTRY | Store country name at region | COUNTRY\_ID |
| CNTR\_GROUP | CNTR\_GROUP | Store country group name at the region | CNTR\_GROUP \_ID |
| REGION | REGION | Store regions | REGION \_ID |

DIM\_PRODUCTS. Hierarchy: PRODUCT-PRODUCT\_CATEGORY

|  |  |  |  |
| --- | --- | --- | --- |
| Name | LEVEL\_CODE | LEVEL\_DESC | LEVEL\_NATURAL\_KEY |
| PRODUCT | PRODUCT | Store product name at product category | PRODUCT \_ID |
| PROD\_CATEGORY | PROD\_CATEGORY | Store product category | PROD\_CATEGORY \_ID |

## Facts

The fact table contain the information about company’s incomes.

### Facts Aggregations

Table data has granularity by the day. On this level table has next aggregations:

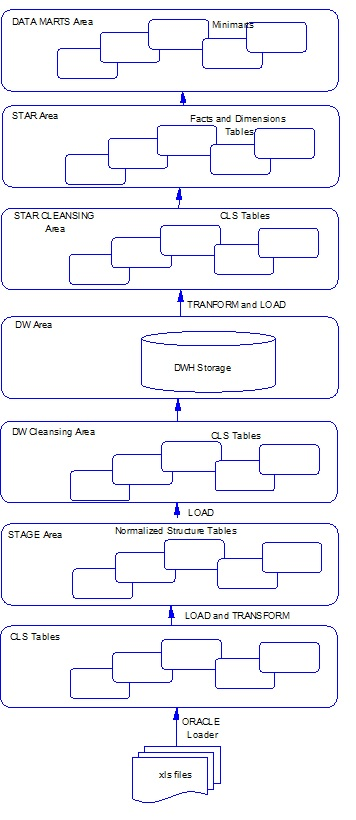
1. Income sum.
2. Number of sold product items.

This calculation described in the paragraph 3.2.

On the level Star Area should be placed two fact tables – FCT\_INCOME\_PRODUCTS\_DAILY and FCT\_INCOME\_PRODUCTA\_MONTHLY.

|  |  |
| --- | --- |
| Table Name | Aggregation Level |
| FCT\_INCOME\_PRODUCTS\_DAILY | Income and number of sold product calculated for each day of company’s activity. |
| FCT\_INCOME\_PRODUCTA\_MONTHLY | Income and number of sold product calculated for each month of company’s activity. |

## Dataflow Diagram



## Partitioning rules

As the company rapidly accumulates information about sold products t will be effective approach to store this information (FCT\_INCOME\_PRODUCT\_DAYLY) divided into partitions. As mostly requests for that information are built with the needs for a specific time period, then the table will store separated into partitions be the column EVENT\_DT.

Range partitioning will divide data by quarters. For each year there will be a tablespace.

|  |
| --- |
| CREATE TABLE FCT\_INCOME\_PRODUCT\_DAYLY  (  event\_dt DATE NOT NULL,  ...  )  PARTITION BY RANGE(event\_dt)  (  PARTITION q1y2010  VALUES LESS THAN (to\_date('04/01/2010', 'mm/dd/yyyy')) tablespace TS\_INC\_2010,  PARTITION q2y2010  VALUES LESS THAN (to\_date('07/01/2010', 'mm/dd/yyyy')) tablespace TS\_INC\_2010,  PARTITION q3y2010  VALUES LESS THAN (to\_date('10/01/2010', 'mm/dd/yyyy')) tablespace TS\_INC\_2010,  PARTITION q4y2010  VALUES LESS THAN (to\_date('01/01/2011', 'mm/dd/yyyy')) tablespace TS\_INC\_2010,  PARTITION q1y2011  VALUES LESS THAN (to\_date('04/01/2011', 'mm/dd/yyyy')) tablespace TS\_INC\_2011,  PARTITION q2y2011  VALUES LESS THAN (to\_date('07/01/2011', 'mm/dd/yyyy')) tablespace TS\_INC\_2011,  PARTITION q3y2011  VALUES LESS THAN (to\_date('10/01/2011', 'mm/dd/yyyy')) tablespace TS\_INC\_2011,  PARTITION q4y2011  VALUES LESS THAN (to\_date('07/01/2012', 'mm/dd/yyyy')) tablespace TS\_INC\_2011,  PARTITION q1y2012  VALUES LESS THAN (to\_date('04/01/2012', 'mm/dd/yyyy')) tablespace TS\_INC\_2012,  PARTITION q2y2012  VALUES LESS THAN (to\_date('07/01/2012', 'mm/dd/yyyy')) tablespace TS\_INC\_2012,  PARTITION q3y2012  VALUES LESS THAN (to\_date('10/01/2012', 'mm/dd/yyyy')) tablespace TS\_INC\_2012,  PARTITION q4y2012  VALUES LESS THAN (to\_date('07/01/2013', 'mm/dd/yyyy')) tablespace TS\_INC\_2012,  PARTITION q1y2013  VALUES LESS THAN (to\_date('04/01/2013', 'mm/dd/yyyy')) tablespace TS\_INC\_2013,  PARTITION q2y2013  VALUES LESS THAN (to\_date('07/01/2013', 'mm/dd/yyyy')) tablespace TS\_INC\_2013,  PARTITION q3y2013  VALUES LESS THAN (to\_date('10/01/2013', 'mm/dd/yyyy')) tablespace TS\_INC\_2013,  PARTITION q4y2013  VALUES LESS THAN (to\_date('07/01/2014', 'mm/dd/yyyy')) tablespace TS\_INC\_2013,  PARTITION INC\_OTHERS  VALUES LESS THAN (MAXVALUE) tablespace TS\_INC\_OTHERS  ); |

## Strategy of Parallel execution

The most actual operations with this DWH are downloading into DWH data about new orders, search of information about products and product analysis. Loading and transformation of order data consists of many steps, so there will be useful parallel execution of insert into tables with order data.