Case Study Data Description

Table of Contents

[1 Data Overview 2](#_Toc504054089)

[1.1 Data Tables 2](#_Toc504054090)

[2 Data Model 2](#_Toc504054091)

[3 Attributes Information of Cleaned Data Tables 3](#_Toc504054092)

[3.1 Product Dimension 3](#_Toc504054093)

[3.2 Date Dimension 3](#_Toc504054094)

[3.3 Site Dimension 4](#_Toc504054095)

[3.4 Planning Data 4](#_Toc504054096)

[3.5 Planning Data at Higher Granularity 5](#_Toc504054097)

[4 Raw Data Cleaning and Transformation 5](#_Toc504054098)

[4.1 Product Dimension 5](#_Toc504054099)

[4.1.1 Attributes in Raw Data 5](#_Toc504054100)

[4.1.2 Data Cleaning 5](#_Toc504054101)

[4.1.3 Data Transformation 6](#_Toc504054102)

[4.2 Date Dimension 6](#_Toc504054103)

[4.2.1 Attributes in Raw Data 6](#_Toc504054104)

[4.2.2 Data Cleaning 6](#_Toc504054105)

[4.3 Site Dimension 7](#_Toc504054106)

[4.3.1 Attributes in Raw Data 7](#_Toc504054107)

[4.3.2 Data Cleaning 7](#_Toc504054108)

[4.4 Planning Data 7](#_Toc504054109)

[4.4.1 Attributes in Raw Data 7](#_Toc504054110)

[4.4.2 Data Cleaning 7](#_Toc504054111)

[4.4.3 Data Transformation 7](#_Toc504054112)

[4.4.4 Aggregation to Per Article, Per Day Granularity 8](#_Toc504054113)

[4.4.5 Aggregation to Per Article, Per Week Granularity 8](#_Toc504054114)

[4.4.6 Aggregation to Per Article, Per Month Granularity 8](#_Toc504054115)

# Data Overview

## Data Tables

The data contains transactional information of selected facial/skin cosmetics products, from 2007-07-30 to 2016-09-11, in a regional market. It also includes attributes of articles (products), sites (stores) and dates. The data was extracted in Dec-2016.

The **raw data set** includes the following data tables:

|  |  |
| --- | --- |
| **File** | **Description** |
| dim\_product\_raw | Product data table |
| dim\_date\_raw | Date data table |
| dim\_site\_raw | Site data table |
| planning\_raw | Transactional data table |

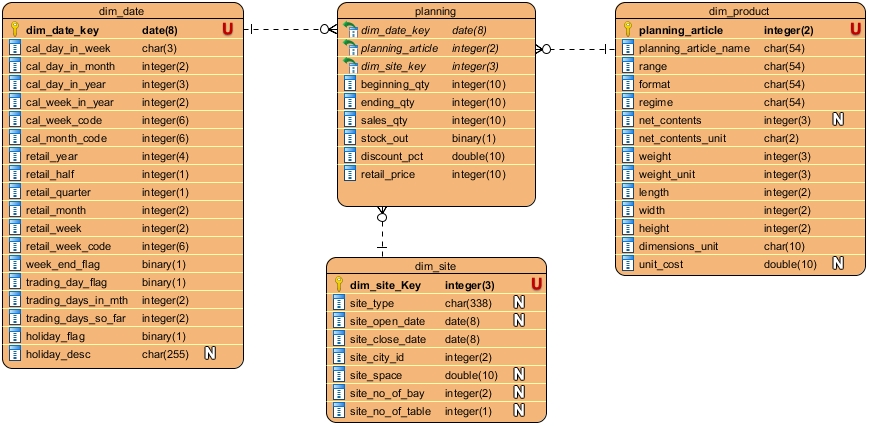
A **cleaned data set** was compiled, which includes the following files:

|  |  |
| --- | --- |
| **File** | **Description** |
| dim\_product\_cleaned | Product data table |
| dim\_date\_cleaned | Date data table |
| dim\_site\_cleaned | Site data table |
| planning\_cleaned | Transactional data table |
| PA\_d | Aggregated transactional data at per article, per day granularity |
| PA\_w | Aggregated transactional data at per article, per week granularity |
| PA\_m | Aggregated transactional data at per article, per month granularity |

The following sections will describe the details of the cleaned data set, the raw data set, as well as the data cleaning and transformation to generate the cleaned data set from the raw data set.

# Data Model

There are three dimensions and one transactional data tables. The three dimensions are *dim\_product*, *dim\_date* and *dim\_site*. The *planning* data table contains historical sales and on-hand quantity information. Below is the ER diagram of the cleaned data tables.



*(ER Diagram generated by Visual Paradigm)*

# Attributes Information of Cleaned Data Tables

## Product Dimension

*dim\_product* data table maps the primary key *‘planning\_article’* to the article attributes.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Example** |
| *planning\_article* | Primary key in format of integer of length 2 | 2 |
| *planning\_article\_name* | Name of the article | PROTECTIVE FACE SERUM |
| *range* | Brand name | VITAMIN |
| *format* | Format and packaging type | FACE SERUMS |
| *regime* | Function | Treatments Skincare |
| *net\_contents* | Net contents | 30 |
| *net\_contents\_unit* | Unit of net contents in “#”, “G”, or “ML”.  “#” is specific for face wipes. | ML |
| *weight* | Weight for logistics | 57 |
| *weight\_unit* | Unit of weight | Grams |
| *length* | Length for logistics | 3 |
| *width* | Width for logistics | 3 |
| *height* | Height for logistics | 11 |
| *dimension\_unit* | Units of the logistics dimensions | Centimeter |
| *unit\_cost* | Unit cost at the time of data extraction | 495 |

## Date Dimension

*dim\_date* data table maps the primary date key *‘dim\_date\_key’* to the date attributes.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Example** |
| *dim\_date\_key* | Primary key in calendar date | 2012-01-01 |
| *cal\_day\_in\_week* | Weekday (Mon – Sun) | Sun |
| *cal\_day\_in\_month* | Day index in calendar month (1 – 31) | 1 |
| *cal\_day\_in\_year* | Day index in calendar year (1 – 366) | 1 |
| *cal\_week\_in\_year* | Calendar week index in calendar year (1 – 53) | 1 |
| *cal\_week\_code* | Calendar week in yyyyww format | 201201 |
| *cal\_month\_code* | Calendar month in yyyymm format | 201201 |
| *retail\_year* | Year index for commercial activities | 2011 |
| *retail\_half* | Half index in retail year | 2 |
| *retail\_quarter* | Quarter index in retail year | 4 |
| *retail\_month* | Month index in retail year | 12 |
| *retail\_week* | Week index in retail year | 52 |
| *retail\_week\_code* | Week in yyyyww format in retail year | 201152 |
| *week\_end\_flag* | N = Mon – Fri; Y = Sat, Sun | Y |
| *trading\_day\_flag* | N = weekend or holiday; Y = not weekend nor holiday | N |
| *trading\_days\_in\_mth* | Number of trading days in the calendar month (5 – 23) | 21 |
| *trading\_days\_so\_far* | Number of trading days passed since the beginning of the calendar month (0 – 23) | 0 |
| *holiday\_flag* | N = not holiday; Y = holiday | Y |
| *holiday\_desc* | Name of the holiday.  Note the holiday information is only available up to 2017-12-31. | New Year’s Day |

Note that there are two date systems: one is calendar (­*cal\_*), the other is retail (*retail\_*). The calendar attributes follow the local calendar, while the retail attributes provide index for commercial activities.

One important note is that, retail weeks always have 7 days, while the first or last calendar week of the year may have fewer than 7 days.

## Site Dimension

*dim\_site* data table maps the primary site key *‘dim\_site\_key’* to the site attributes.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Example** |
| *dim\_site\_key* | Primary key in format of integer of length 3 | 10 |
| *site\_type* | Type of site | MALL |
| *site\_open\_date* | Open date of the site | 2009-03-31 |
| *site\_close\_date* | Close (or target close) date of the site | 2049-12-31 |
| *site\_city\_id* | City ID of the site | 1 |
| *site\_space* | Area of site | 70.320 |
| *site\_no\_of\_bay* | Number of bays  Primary measurement of site size. | 17 |
| *site\_no\_of\_table* | Number of tables | 1 |

Below is a reference photo to illustrate the concept of bays and tables of a site.



## Planning Data

*planning* data table contains historical sales and on-hand information of 54 selected facial/skin cosmetics articles. The observations present historical data at per article, per site, per day granularity.

This data has been cleaned to ensure:

* Each selected article has at least 365 days of data
* Each selected article has relatively low average stock-out level.
* For each selected article, every week has completely 7 days of data.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Description** | **Example** |
| *planning\_article* | Planning article key | 22 |
| *dim\_date\_key* | Date key | 2011-09-11 |
| *dim\_site\_key* | Site key | 26 |
| *beginning\_qty* | On-hand quantity at beginning of the day | 12 |
| *ending\_qty* | On-hand quantity at the end of the day | 10 |
| *sales\_qty* | Total sales quantity of the day. Note that *sales\_qty* can be different from *beginning\_qty - ending\_qty*, because of other non-sales transactions. | 2 |
| *stock\_out* | 0: *beginning\_qty>0 & ending\_qty>0*  1: *beginning\_qty0 | ending\_qty0* | 0 |
| *discount\_pct* | Average discount% of the product sold. | 17.5 |
| *retail\_price* | Original unit price (IDR) of the product sold. | 600 |

“PA\_d\_s.jpeg” presents a time-series overview of sales quantity, discount% and stock out%, at per article, per day granularity.

## Planning Data at Higher Granularity

Several additional data tables were generated through aggregation of the planning data to enable analysis at higher granularity:

* *PA\_d*: aggregate over all sites to present regional total values of each article on each day
* *PA\_w*: aggregate *PA\_d* over days to present total values of each article in each week
* *PA\_m*: aggregate *PA\_d* over days to present total values of each article in each month

Table below summarizes the attributes of the aggregated data different from the cleaned planning data:

|  |  |
| --- | --- |
| **Different Attribute** | **Description** |
| *retail\_week\_code* | Retail weeks in yyyyww format; always have 7 days in a week |
| *cal\_month\_code* | Calendar months in yyyymm format |
| *dim\_date\_key* | Date of the 1st day of the retail week/calendar month |
| *beginning\_qty* | Sum of *beginning\_qty* of all sites and all days in the period |
| *ending\_qty* | Sum of *ending\_qty* of all sites and all days in the period |
| *stock\_out\_pct* | Average *stock\_out* of all sites and all days in the period |
| *sites\_num* | Average number of sites per day selling the article in the period |

# Raw Data Cleaning and Transformation

The following sections introduce the attributes in the raw data, as well as the data cleaning and transformation to generate the cleaned data.

## Product Dimension

### Attributes in Raw Data

The raw data contains following attributes with the same meanings as in the cleaned data:

* *planning\_article, planning\_article\_name, range, format, regime, net\_contents, net\_contents\_unit, weight, weight\_unit, length, width, height, dimension\_unit, unit\_cost*

It contains one additional attribute:

* *dim\_product\_key*: Product key in format of integer of length 3.

Relationship between planning\_article and dim\_product\_key is 1-to-many.

The reason for one article to have multiple product keys is that, product keys are used to track minor changes of the article, which is not concerned in this case study.

### Data Cleaning

Table below summarizes the data cleaning details of the corresponding attributes.

|  |  |
| --- | --- |
| **Attributes** | **Data Cleaning** |
| *format, regime, net\_contents, net\_contents\_unit, weight\_unit, dimension\_unit* | Remove head and tail blank spaces from the string and convert “” to NA.  Example: |
|  |  |
| *format* | Change ‘LIQUIDS’ to ‘FACE TONERS’ |
| *regime* | Change ‘Moisturisers Skcare’ to ‘Moisturisers Skincare’ |
| *length, width, height* | If length, width and height of a product are all equal to 1, these dimension values are considered as abnormal. Replace these values with NA.  Example: |
| *weight* | Weight value 1 Grams is considered as abnormal; replace with NA.  Example: |

### Data Transformation

Because the analysis with be based on *planning\_article*, the raw product data is aggregated over

*dim\_product\_key* to use *planning\_article* as the primary key of the transformed data.

Below are the steps:

1. Arrange the data by *dim\_product\_key* in descending order.
2. Filter out records without *planning\_article* value.
3. Group by each *planning\_article*:
   1. Select on set of *length, width, height* and *unit\_cost* values:
      1. Get index of the first record with valid *unit\_cost* value.
      2. Use the *length, width, height* and *unit\_cost* values at index as the selected values, if the values are not missing; If any value at index is missing, use the value at index 1.
   2. Keep the first row of the group.
4. Remove the *dim\_product\_key* column.

## Date Dimension

### Attributes in Raw Data

The raw data contains following attributes with the same meanings as in the cleaned data:

* *dim\_date\_key, cal\_day\_in\_week, cal\_day\_in\_month, cal\_day\_in\_year, cal\_week\_in\_year, retail\_year, retail\_half, retail\_quarter, retail\_month, retail\_week, week\_end\_flag, trading\_day\_flag, trading\_days\_in\_mth, trading\_days\_so\_far, holiday\_flag, holiday\_desc*

Some attributes in the raw data are in different formats from the cleaned data, which will be explained in the data cleaning section.

### Data Cleaning

Table below summarizes the data cleaning details of the corresponding attributes.

|  |  |
| --- | --- |
| **Attributes** | **Data Cleaning** |
| *retail\_half* | Remove the leading “H”. Example: change “H1” to 1. |
| *retail\_quarter* | Remove the leading “Q”. Example: change “Q1” to 1. |
| *retail\_month* | Remove the tailing “-Mmm”. Example: change “04-Apr” to 4. |
| *retail\_week* | Remove the leading “W”. Example: change “W01” to 1. |
| *cal\_week\_code* | Generate by “(year of *dim\_date\_key*) \* 100 + *cal\_week\_in\_year*”. |
| *cal\_month\_code* | Generate by converting *dim\_date\_key* to “yyyymm” format. |
| *retail\_week\_code* | Generate by “*retail\_year* \* 100 + *retail\_week*”. |
| *holiday\_flag, holiday\_desc* | Add missing “Labour Day” (1-May) in year 2011, 2012 and 2013. |

## Site Dimension

### Attributes in Raw Data

The raw data contains following attributes with the same meanings as in the cleaned data:

* *dim\_site\_key, site\_type, site\_open\_date, site\_close\_date, site\_city\_id, site\_space, site\_no\_of\_bay, site\_no\_of\_table*

### Data Cleaning

Table below summarizes the data cleaning details of the corresponding attributes.

|  |  |
| --- | --- |
| **Attributes** | **Data Cleaning** |
| *site\_type, site\_space, site\_no\_of\_bay, site\_no\_of\_table* | Remove head and tail blank spaces from the string and convert “” to NA.  Example: |
| *site\_open\_date, site\_close\_date* | Change from “yyyy-mm-dd 00:00:00.000000000” to “yyyy-mm-dd” format. |

## Planning Data

### Attributes in Raw Data

The raw data contains following attributes with the same meanings as in the cleaned data:

* *planning\_article, dim\_date\_key, dim\_site\_key, beginning\_qty, ending\_qty, sales\_qty, retail\_price*

It contains additional attributes:

* *dim\_product\_key*: Same as the *dim\_product\_key* in *dim\_product* raw data.
* *adj\_qty*: Manual input to adjust the *beginning\_qty*
* unit\_markdown: Average price reduction for each product sold. Note that it is presented in negative values in the raw data.

### Data Cleaning

Table below summarizes the data cleaning details of the corresponding attributes.

|  |  |
| --- | --- |
| **Attributes** | **Data Cleaning** |
| *sales\_qty* | Adjust negative values to 0. |
| *beginning\_qty* | Adjust *beginning\_qty = beginning\_qty + adj\_qty* |
| *unit\_markdown* | Inverse the sign; then adjust negative values to 0.  For example, -1 adjust to 1, while 1 adjust to -1 then to 0. |
| *retail\_price* | Fill up missing values and replace abnormal values (0 or 1):   1. Sort the table by *dim\_product\_key, dim\_site\_key, dim\_date\_key*; 2. Replace each missing/abnormal value by the closest valid previous value. |

### Data Transformation

Following transformation was conducted to generate the cleaned planning data at per article, per day, per site granularity:

1. Filter out all records after the close date of the site:
   1. Merge *site\_close\_date* from *dim\_site* by *dim\_site\_key*.
   2. Remove all records with *dim\_date\_key* *site\_close\_date*.
2. Generate total sales in money value: *total\_sales\_amt = retail\_price \* sales\_qty*
3. Generate total discount amount in money value:
   1. *total\_disc\_amt = unit\_markdown \* sales\_qty*
   2. Adjust those *total\_disc\_amt* above *total\_sales\_amt* to be equal to *total\_sales\_amt*.
4. Aggregate over *dim\_product\_key* to per article, per day, per site level:  
   (Note the following sequence matters)
   1. Group by *planning\_article, dim\_date\_key, dim\_site\_key.*
   2. Sum up *beginning\_qty.*
   3. Sum up *ending\_qty.*
   4. Generate *stock\_out*: 1 if *ending\_qty0 or beginning\_qty0*; otherwise 0.
   5. Generate *discount\_pct = sum(total\_disc\_amt)/sum(total\_sales\_amt)\*100* if *sum(total\_sales\_amt)>0*; otherwise 0.
   6. Sum up *sales\_qty*.
   7. *retail\_price = sum(total\_sales\_amt)/sales\_qty* if *sales\_qty>0*; otherwise use *mean(retail\_price)*. Note that the *sales\_qty* in the step has been summed up in the previous step.
5. For each article, filter out beginning/ending week without 7 days of data:
   1. Merge *retail\_week\_code* from *dim\_date* by *dim\_date\_key*.
   2. Group by *planning\_article, retail\_week\_code*.
   3. Filter out groups without 7 distinct *dim\_date\_key*.
6. Remove *retail\_week\_code* column.

### Aggregation to Per Article, Per Day Granularity

Below are the steps to aggregate the cleaned planning data to per article, per day granularity (*PA\_d*):

1. Group by *planning\_article* and *dim\_date\_key*.
2. Generate the following columns (note that the sequence matters):
   1. Sum up *beginning\_qty*.
   2. Sum up *ending\_qty*.
   3. *retail\_price =* weighted average of *retail\_price* over *sales\_qty* if *sales\_qty*>0; otherwise use *mean(retail\_price)*.
   4. *stock\_out\_pct = mean(stock\_out) \* 100*
   5. *discount\_pct =* weighted average of *discount\_pct* over *sales\_qty* if *sales\_qty*>0; otherwise 0.
   6. Sum up *sales\_qty*.
   7. Number of sites selling the article on the day: *sites\_num* = distinct count of *dim\_site\_key*.

### Aggregation to Per Article, Per Week Granularity

Below are the steps to aggregate *PA\_d* to per article, per week granularity (*PA\_w*):

1. Merge *retail\_week\_code* from *dim\_date* by *dim\_date\_key*.
2. Group by *planning\_article* and *retail\_week\_code*.
3. Generate the following columns (note that the sequence matters):
   1. *dim\_date\_key*: first value in the group.
   2. Sum up *beginning\_qty*.
   3. Sum up *ending\_qty*.
   4. *retail\_price =* weighted average of *retail\_price* over *sales\_qty* if *sales\_qty*>0; otherwise use *mean(retail\_price)*.
   5. *stock\_out\_pct =* weighted average of *stock\_out\_pct* over *sites\_num*.
   6. *discount\_pct =* weighted average of *discount\_pct* over *sales\_qty* if *sales\_qty*>0; otherwise 0.
   7. Sum up *sales\_qty*.
   8. Take mean of *sites\_num*.

### Aggregation to Per Article, Per Month Granularity

Below are the steps to aggregate *PA\_d* to per article, per month granularity (*PA\_m*):

1. Merge *cal\_month\_code* from *dim\_date* by *dim\_date\_key*.
2. Group by *planning\_article* and *cal\_month\_code*.
3. Filter out those months with fewer than 28 days of data.
4. Generate the following columns (note that the sequence matters):
   1. *dim\_date\_key*: first value in the group.
   2. Sum up *beginning\_qty*.
   3. Sum up *ending\_qty*.
   4. *retail\_price =* weighted average of *retail\_price* over *sales\_qty* if *sales\_qty*>0; otherwise use *mean(retail\_price)*.
   5. *stock\_out\_pct =* weighted average of *stock\_out\_pct* over *sites\_num*.
   6. *discount\_pct =* weighted average of *discount\_pct* over *sales\_qty* if *sales\_qty*>0; otherwise 0.
   7. Sum up *sales\_qty*.
   8. Take mean of *sites\_num*.