\

EPAM Systems, RD Dep.

Philip Morris International DWH

PHILIP MORRIS

Contents

1. Business Description

1.1. Business background

Philip Morris International Inc. (PMI) (NYSE: PM) is an American global cigarette and tobacco company, with products sold in over 180 countries outside the United States. The most recognized and best selling product of the company is Marlboro.

The company's operational headquarters are in Lausanne, Switzerland, although the corporate headquarters remain in New York. It does not operate in the United States, with Philip Morris brands there still owned by PMI's former owner Altria.

Because tobacco, the main constituent of cigarettes, is the single greatest cause of preventable death globally[3] and is addictive, the company's operations (and its competitors') are highly controversial and are increasingly the subject of litigation and restrictive legislation from governments concerned about the health impacts of its products.

Philip Morris International has six multi-billion US$ brands including:

1. Dji Sam Soe 234 was launched in 1913 and is a brand of Kretek cigarettes. It is the best seller of Kretek cigarettes in Indonesia.

2. L&M was launched by Liggett & Myers in 1953 with the tagline: "American cigarettes of the highest quality with the best filter." L&M variants include full flavor shorts, full flavor 100s, lights, ultra lights, menthol shorts, menthol 100s, menthol light shorts, menthol light 100s, Turkish Blend shorts, Turkish Blend 100s, and L&M Mild Kretek.

3. Longbeach include in Australia and Indonesia in 1999. Longbeach variant include: Longbeach Filter and Longbeach Mild.

4. Marlboro was launched in 1904. Marlboro is the premium brand. Marlboro variants include: Marlboro Special, Marlboro Menthol, Marlboro Lights, Marlboro Lights Menthol, Marlboro Mix-9 Filter Kretek, Marlboro Flavor Plus, Marlboro Black Menthol, and Heatsticks, a heated tobacco product.[10]

5. ST Dupont Paris is the brand cigarette designed by Simon Tissot Dupont in 1902. With the black packaging. ST Dupont Paris variants include: filter, lights, menthol, and menthol lights.

6. U Mild was launched in Indonesia in May 22, 1998 after Indonesian revolution. U Mild is a Mild Kretek cigarette sold in Indonesia.

In the end of 2016 Philip Morris presented the new product called IQOS - smokeless cigarette–one of the latest products from Philip Morris International, manufacturers of the legendary Marlboro cigarettes. The product is a hybrid between the benefits of electronic cigarettes and the design of traditional types – making it perfect for smokers looking to switch to vaping or even for occasional use. Given the known dangers of smoking traditional cigarettes, switching to vaping or just implementing a sporadic alternative is a good option.

The iQOS uses refills of real tobacco, but doesn’t burn it like the conventional cigarette which prevents the ingestion of combustive, charred materials. Rather, the iQOS ecig heats the tobacco just prior to the point of combustion, producing tobacco-flavored vapor.

1.2. Problems because of poor data management

A brand new product (iQOS) appeared in the last year, but due to the poor management, it didn’t get much popularity worldwide. As a result new production items had produced very little profit compared to the forecasts and the amount of resources that was spent on its production. All of this caused financial problems of the company and damaged the economic policy in 2017.

1.3. Benefits from implementing a Data Warehouse

So as to recover the company's brand name, the board of directors decided to build a detailed plan of the economic situation on the current date and to provide detailed information about all products and its sales amounts along with new company's production items. The main goal of the research is to get proper information about popularity and economic profitability of new products. Another task is to gather information about "old" production items. The research should be based on prebuild Data Warehouse. Data Warehouse should also be able to aggregate all information about sales for future years till 2030.

2. Dimensions of a Business

Starting with identifying 4 steps of creating data warehouse

1. Select the business process

   Getting detailed information about brand-new product contribution to the company's production process

2. Declare the Grain

   Total sales and quantity amount of new products compared

3. Identify the Dimensions

   - Products;
   - Departments;
   - Consumers;
   - Time;
   - Promotion;

4. Identify the Facts

"Total sales amount" as FACT_SALES

Entities:

- **Product Category**

  Categories of the products

  Examples: cigarette, tobacco heating products, vapor products

- **Product**

  Brands, produced by PMI company

  Examples: Marlboro, Most popular product of the company, global,

- **Location**

  Addresses, cities, countries and regions

  Example: 10-3 Hyde Park, NY, USA, North America

- **Consumers**

  Consumers of the PMI products

  Examples: Japan retail agency

- **Departments**

  PMI agency all over the world

  Example: St. Petersburg Main Agency, 2201

- **Promotions**

  Promotions, which take place in some parts of the year

  Example: Promotion #276421 Brand-new product

- **Sale**

  Sale of some product to the customer

  Example: PRODUCT, PRICE, QUANTITY, CONSUMER, DEPARTMENT, DATE

As a result, we get such **DIMENSIONS:**

3. Logical schema

Provided dimensions and fact table build the Star schema of data warehouse. Star schema was chosen for the following reasons:

- Maximum performance and ease of access;

  PMI Corporation is ready to invest much in local storage to get rid of redundant connections so as to make access to all data without extra joins and do not pay much attention about redundancy in data.

- Simplification of preparing data for reporting

  Using Star makes does not require special IT staff for ETL and building data marts for appropriate report because almost everything will be aggregated in dimensions and the only thing to do will be some simple grouping operations,

As far as dimensions were defined in the previous chapter, PMI data warehouse gets next model:

Star schema:

Picture – PMI Star schema model

List of tables in Star schema:

DIM_CONSUMERS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
CONSUMER_SURR_ID	NUMBER(8)	True	
CONSUMER_ID	NUMBER(8)	False	
CONSUMER_DESC	VARCHAR2(50)	False	
RATE_ID	NUMBER(8)	False	
RATE_DESC	VARCHAR2(50)	False	
RESPONSIBLE_ID	NUMBER(8)	False	
RESPONSIBLE_DESC	VARCHAR2(100)	False	
ADDRESS_ID	NUMBER(8)	False	
ADDRESS_DESC	VARCHAR2(50)	False	
CITY_ID	NUMBER(8)	False	
CITY_DESC	VARCHAR2(50)	False	
COUNTRY_ID	NUMBER(8)	False	
COUNTRY_DESC	VARCHAR2(50)	False	
SUBREGION_ID	NUMBER(8)	False	
SUBREGION_DESC	VARCHAR2(50)	False	
REGION_ID	NUMBER(8)	False	
REGION_DESC	VARCHAR2(50)	False	
POSTAL_CODE	VARCHAR2(50)	False	
PHONE	VARCHAR2(50)	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_DIM_CONSUMERS	CONSUMER_SURR_ID	

DIM_DATE

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
DATE_ID	DATE	True	
DAY_WEEK	NUMBER(8)	False	
DAY_SHORT	VARCHAR2(50)	False	
DAY_LONG	VARCHAR2(50)	False	
DAY_MONTH	NUMBER(8)	False	
LASTDAY_MONTH	DATE	False	
DAY_YEAR	NUMBER(8)	False	
MONTH_YEAR	NUMBER(8)	False	
MONTH_SHORT	VARCHAR2(50)	False	
MONTH_LONG	VARCHAR2(50)	False	
MONTH_DESC	VARCHAR2(50)	False	
QUARTER	NUMBER(8)	False	
YEAR_HALF	NUMBER(8)	False	
YEAR	NUMBER(8)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
	DATE_ID	

PK_DIM_DATE		

DIM_DEPARTMENTS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
DEPARTMENT_SURR_ID	NUMBER(8)	True	
DEPARTMENT_ID	NUMBER(8)	False	
DEPARTMENT_DESC	VARCHAR2(50)	False	
ADDRESS_ID	NUMBER(8)	False	
ADDRESS_DESC	VARCHAR2(50)	False	
PHONE	VARCHAR2(20)	False	
POSTAL_CODE	VARCHAR2(50)	False	
CITY_ID	NUMBER(8)	False	
CITY_DESC	VARCHAR2(50)	False	
COUNTRY_ID	NUMBER(8)	False	
COUNTRY_DESC	VARCHAR2(50)	False	
SUBREGION_ID	NUMBER(8)	False	
SUBREGION_DESC	VARCHAR2(50)	False	
REGION_ID	NUMBER(8)	False	
REGION_DESC	VARCHAR2(50)	False	
START_DT	DATE	False	
END_DT	DATE	False	
IS_ACTIVE	VARCHAR2(1)	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	
PRODUCT_ID	NUMBER	False	
CONSUMER_ID	NUMBER	False	
PROMO_ID	NUMBER	False	
DATE_ID	DATE	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_DIM_DEPARTMENTS	DEPARTMENT_SURR_ID	

DIM_PRODUCTS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PRODUCT_SURR_ID	NUMBER(8)	True	
PRODUCT_ID	NUMBER(8)	False	
PRODUCT_DESC	VARCHAR2(50)	False	
PRODUCT_PRICE	NUMBER(8,2)	False	
CATEGORY_ID	NUMBER(8)	False	
CATEGORY_DESC	VARCHAR2(50)	False	
SUBCATEGORY_ID	VARCHAR2(50)	False	
SUBCATEGORY_DESC	VARCHAR2(50)	False	
SUBCATEGORY_SIZE	NUMBER(8,2)	False	
SUBCATEGORY_TOBACCO	NUMBER(8,2)	False	
	NUMBER(8)	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
LOCALIZATION_ID			
LOCALIZATION_DESC	VARCHAR2(50)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_DIM_PRODUCTS	PRODUCT_SURR_ID	

DIM_PROMOTION

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PROMO_SURR_ID	NUMBER(8)	True	
PROMO_ID	NUMBER(8)	False	
PROMO_NAME	VARCHAR2(100)	False	
PROMO_DESC	VARCHAR2(100)	False	
PROMO_COST	NUMBER(8,2)	False	
START_DT	DATE	False	
END_DT	DATE	False	
IS_ACTIVE	VARCHAR(1)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_DIM_PROMOTIONS	PROMO_SURR_ID	

DIM_SALE_STATE

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
STATE_ID	NUMBER(8)	True	
STATE_DESC	VARCHAR2(50)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_SALE_STATE	STATE_ID	

FCT_SALES

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PRODUCT_ID	NUMBER(8)	True	
DEPARTMENT_ID	NUMBER(8)	True	
CONSUMER_ID	NUMBER(8)	True	
PROMO_ID	NUMBER(8)	True	
DATE_ID	DATE	True	
COST	NUMBER(8,2)	False	
AMOUNT	NUMBER(8,2)	False	

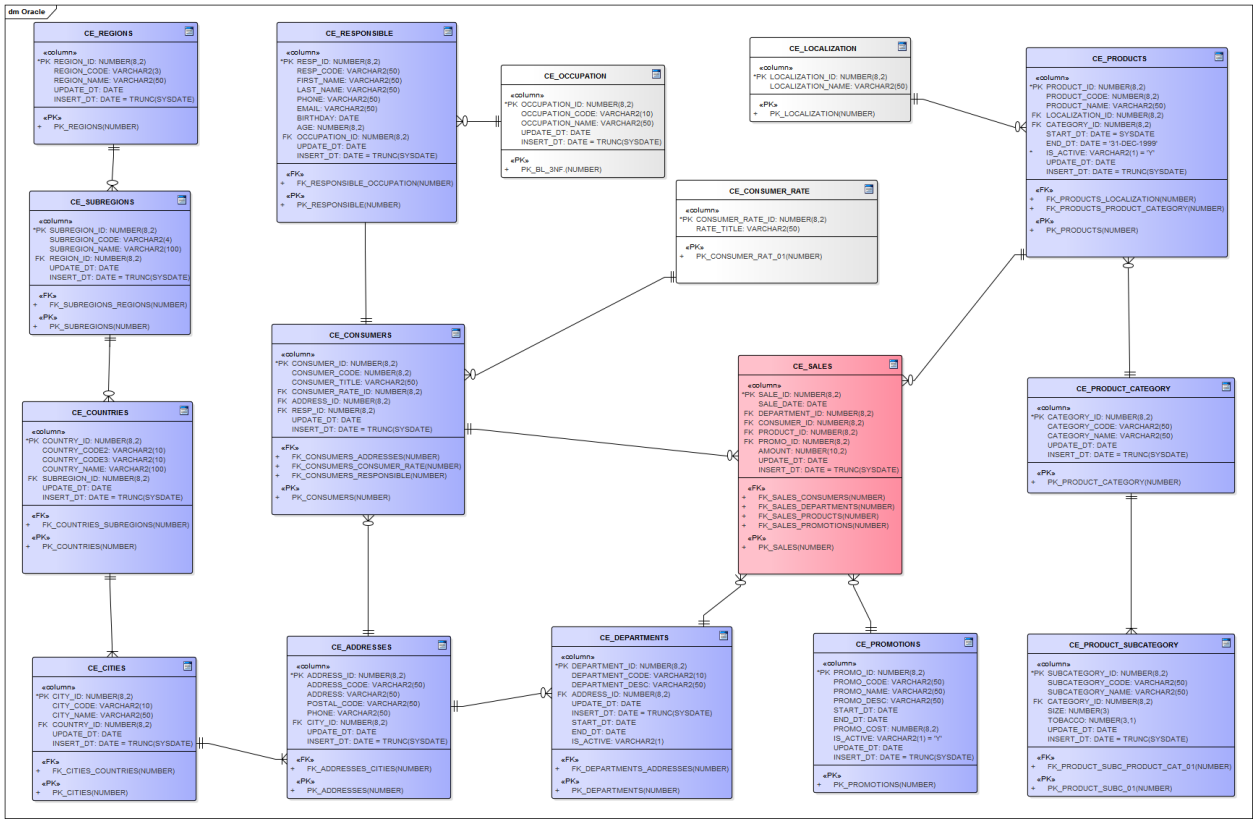
PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_FACT_SALES	PRODUCT_ID, DEPARTMENT_ID, CONSUMER_ID, PROMO_ID, DATE_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_FACT_SALES_DIM_DATE	DATE_ID	DIM_DATE(DATE_ID)
FK_FACT_SALES_DIM_PRODUCTS	PRODUCT_ID	DIM_PRODUCTS(PRODUCT_SURR_ID)
FK_FACT_SALES_DIM_CONSUMERS	CONSUMER_ID	DIM_CONSUMERS(CONSUMER_SURR_ID)

FK_FACT_SALES_DIM_PROMOTIONS	PROMO_ID	DIM_PROMOTIONS(PROMO_SURR_ID)
FK_FACT_SALES_DIM_DEPARTMENTS	DEPARTMENT_ID	DIM_DEPARTMENTS(DEPARTMENT_SURR_ID)

As far as star schema is built, the model of PMI database in 3rd normal form can be seen below.

3NF model:



Picture – PMI 3rd normal form model

Description of PMI tables can be seen below:

CE_ADDRESSES

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
ADDRESS_ID	NUMBER(8,2)	True	
ADDRESS_CODE	VARCHAR2(50)	False	
ADDRESS	VARCHAR2(50)	False	
POSTAL_CODE	VARCHAR2(50)	False	
PHONE	VARCHAR2(50)	False	
CITY_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_ADDRESSES	ADDRESS_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_ADDRESSES_CITIES	CITY_ID	CE_CITIES(CITY_ID)

CE_CITIES

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
CITY_ID	NUMBER(8,2)	True	
CITY_CODE	VARCHAR2(10)	False	
CITY_NAME	VARCHAR2(50)	False	
COUNTRY_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_CITIES	CITY_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_CITIES_COUNTRIES	COUNTRY_ID	CE_COUNTRIES(COUNTRY_ID)

CE_CONSUMERS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
CONSUMER_ID	NUMBER(8,2)	True	
CONSUMER_CODE	NUMBER(8,2)	False	
CONSUMER_TITLE	VARCHAR2(50)	False	
CONSUMER_RATE_ID	NUMBER(8,2)	False	
ADDRESS_ID	NUMBER(8,2)	False	
RESP_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_CONSUMERS	CONSUMER_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_CONSUMERS_CONSUMER_RATE	CONSUMER_RATE_ID	CE_CONSUMER_RATE(CONSUMER_RATE_ID)
FK_CONSUMERS_RESPONSIBLE	RESP_ID	CE_RESPONSIBLE(RESP_ID)

FK_CONSUMERS_ADDRESSES	ADDRESS_ID	CE_ADDRESSES(ADDRESS_ID)

CE_CONSUMER_RATE

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
CONSUMER_RATE_ID	NUMBER(8,2)	True	
RATE_TITLE	VARCHAR2(50)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_CONSUMER_RAT_01	CONSUMER_RATE_ID	

CE_COUNTRIES

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
COUNTRY_ID	NUMBER(8,2)	True	
COUNTRY_CODE2	VARCHAR2(10)	False	
COUNTRY_CODE3	VARCHAR2(10)	False	
COUNTRY_NAME	VARCHAR2(100)	False	
SUBREGION_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_COUNTRIES	COUNTRY_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES

FK_COUNTRIES_SUBREGIONS	SUBREGION_ID	CE_SUBREGIONS(SUBREGION_ID)

CE_DEPARTMENTS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
DEPARTMENT_ID	NUMBER(8,2)	True	
DEPARTMENT_CODE	VARCHAR2(10)	False	
DEPARTMENT_DESC	VARCHAR2(50)	False	
ADDRESS_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)
START_DT	DATE	False	
END_DT	DATE	False	
IS_ACTIVE	VARCHAR2(1)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_DEPARTMENTS	DEPARTMENT_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_DEPARTMENTS_ADDRESSES	ADDRESS_ID	CE_ADDRESSES(ADDRESS_ID)

CE_LOCALIZATION

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
LOCALIZATION_ID	NUMBER(8,2)	True	
LOCALIZATION_NAME	VARCHAR2(50)	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_LOCALIZATION	LOCALIZATION_ID	

CE_OCCUPATION

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
OCCUPATION_ID	NUMBER(8,2)	True	
OCCUPATION_CODE	VARCHAR2(10)	False	
OCCUPATION_NAME	VARCHAR2(50)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_	OCCUPATION_ID	

CE_PAYMENTS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PAYMENT_ID	NUMBER(8,2)	True	
PAYMENT_CODE	VARCHAR2(10)	False	
CONSUMER_ID	NUMBER(8,2)	False	
SALE_ID	NUMBER(8,2)	False	
DEPARTMENT_ID	NUMBER(8,2)	False	
COST	NUMBER(8,2)	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
AMOUNT	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: SYSDATE

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_PAYMENTS	PAYMENT_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_PAYMENTS_SALES	SALE_ID	CE_SALES(SALE_ID)

CE_PAY_STATE

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PAY_STATE_ID	NUMBER(8,2)	True	
PAY_STATE_NAME	VARCHAR2(50)	False	

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_PAY_STATE	PAY_STATE_ID	

CE_PRODUCTS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PRODUCT_ID	NUMBER(8,2)	True	
PRODUCT_CODE	NUMBER(8,2)	False	
PRODUCT_NAME	VARCHAR2(50)	False	
LOCALIZATION_ID	NUMBER(8,2)	False	
CATEGORY_ID	NUMBER(8,2)	False	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
START_DT	DATE	False	Initial value: SYSDATE
END_DT	DATE	False	Initial value: '31-DEC-1999'
IS_ACTIVE	VARCHAR2(1)	True	Initial value: 'Y'
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_PRODUCTS	PRODUCT_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_PRODUCTS_PRODUCT_CATEGORY	CATEGORY_ID	CE_PRODUCT_CATEGORY(CATEGORY_ID)
FK_PRODUCTS_LOCALIZATION	LOCALIZATION_ID	CE_LOCALIZATION(LOCALIZATION_ID)

CE_PRODUCT_CATEGORY

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
CATEGORY_ID	NUMBER(8,2)	True	
CATEGORY_CODE	VARCHAR2(50)	False	
CATEGORY_NAME	VARCHAR2(50)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS

PK_PRODUCT_CATEGORY	CATEGORY_ID	

CE_PRODUCT_SUBCATEGORY

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
SUBCATEGORY_ID	NUMBER(8,2)	True	
SUBCATEGORY_CODE	VARCHAR2(50)	False	
SUBCATEGORY_NAME	VARCHAR2(50)	False	
CATEGORY_ID	NUMBER(8,2)	False	
SIZE	NUMBER(3)	False	
TOBACCO	NUMBER(3,1)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_PRODUCT_SUBC_01	SUBCATEGORY_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_PRODUCT_SUBC_PRODUCT_CAT_01	CATEGORY_ID	CE_PRODUCT_CATEGORY(CATEGORY_ID)

CE_PROMOTIONS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PROMO_ID	NUMBER(8,2)	True	

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
PROMO_CODE	VARCHAR2(50)	False	
PROMO_NAME	VARCHAR2(50)	False	
PROMO_DESC	VARCHAR2(50)	False	
START_DT	DATE	False	
END_DT	DATE	False	
PROMO_COST	NUMBER(8,2)	False	
IS_ACTIVE	VARCHAR2(1)	False	Initial value: 'Y'
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_PROMOTIONS	PROMO_ID	

CE_REGIONS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
REGION_ID	NUMBER(8,2)	True	
REGION_CODE	VARCHAR2(3)	False	
REGION_NAME	VARCHAR2(50)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_REGIONS	REGION_ID	

CE_RESPONSIBLE

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
RESP_ID	NUMBER(8,2)	True	
RESP_CODE	VARCHAR2(50)	False	
FIRST_NAME	VARCHAR2(50)	False	
LAST_NAME	VARCHAR2(50)	False	
PHONE	VARCHAR2(50)	False	
EMAIL	VARCHAR2(50)	False	
BIRTHDAY	DATE	False	
AGE	NUMBER(8,2)	False	
OCCUPATION_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_RESPONSIBLE	RESP_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_RESPONSIBLE_OCCUPATION	OCCUPATION_ID	CE_OCCUPATION(OCCUPATION_ID)

CE_SALES

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
SALE_ID	NUMBER(8,2)	True	
SALE_DATE	DATE	False	
DEPARTMENT_ID	NUMBER(8,2)	False	
CONSUMER_ID	NUMBER(8,2)	False	
PRODUCT_ID	NUMBER(8,2)	False	
PROMO_ID	NUMBER(8,2)	False	
AMOUNT	NUMBER(10,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_SALES	SALE_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_SALES_PRODUCTS	PRODUCT_ID	CE_PRODUCTS(PRODUCT_ID)
		CE_PAY_STATE()
FK_SALES_PROMOTIONS	PROMO_ID	CE_PROMOTIONS(PROMO_ID)
FK_SALES_DEPARTMENTS	DEPARTMENT_ID	CE_DEPARTMENTS(DEPARTMENT_ID)
FK_SALES_CONSUMERS	CONSUMER_ID	CE_CONSUMERS(CONSUMER_ID)

CE_SUBREGIONS

COLUMN NAME	DATATYPE	NOT NULL	COMMENTS

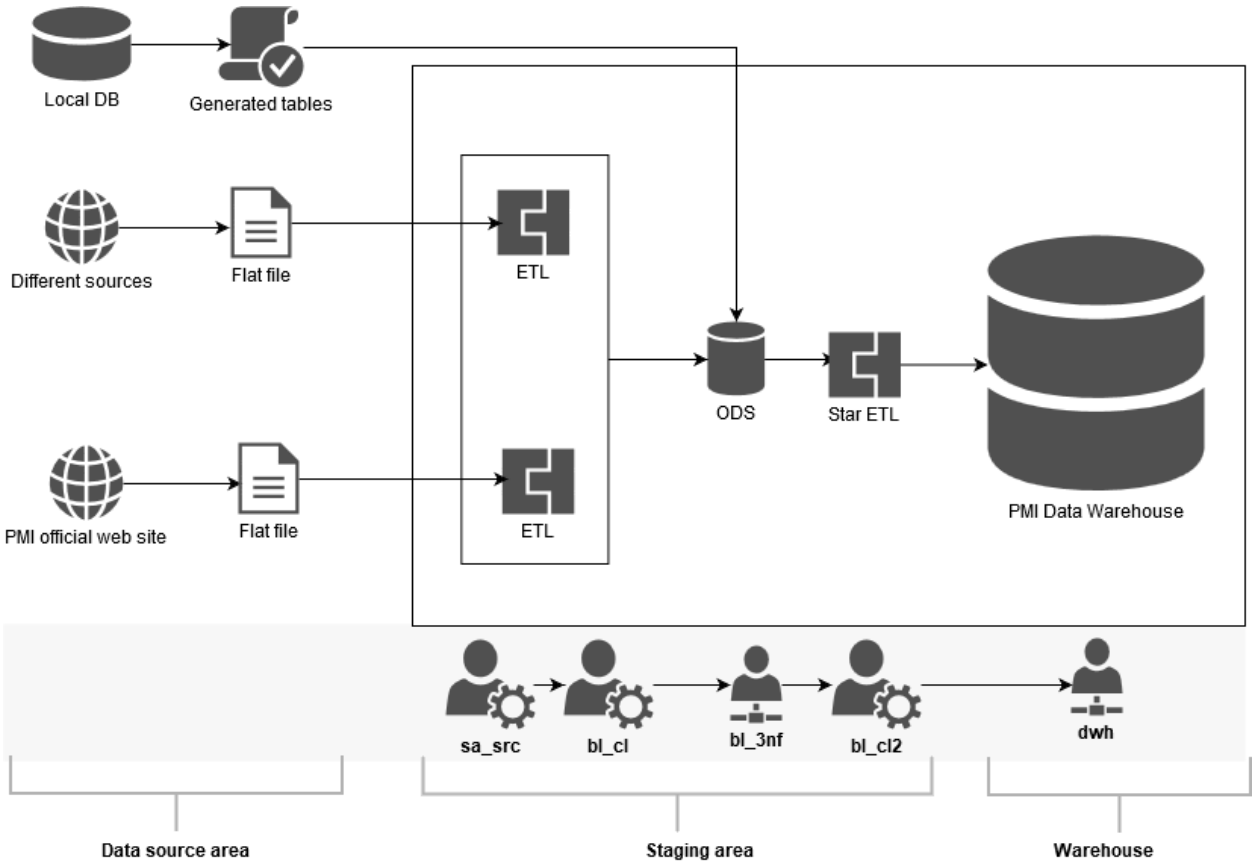
COLUMN NAME	DATATYPE	NOT NULL	COMMENTS
SUBREGION_ID	NUMBER(8,2)	True	
SUBREGION_CODE	VARCHAR2(4)	False	
SUBREGION_NAME	VARCHAR2(100)	False	
REGION_ID	NUMBER(8,2)	False	
UPDATE_DT	DATE	False	
INSERT_DT	DATE	False	Initial value: TRUNC(SYSDATE)

PRIMARY KEY NAME	COLUMNS	COMMENTS
PK_SUBREGIONS	SUBREGION_ID	

FOREIGN KEY NAME	COLUMNS	REFERENCES
FK_SUBREGIONS_REGIONS	REGION_ID	CE_REGIONS(REGION_ID)

4. Data flow

Here you can see the example of data flow that will be built during the process of building DWH:



Picture – PMI data flow model

Data flow has 3 areas:

1) Data source area;
2) Staging area;
3) Warehouse area.

Data source area

On the first step main task is to choose appropriate sources with PMI company data and combine it in flat files so as to prepare data for ETL processes. **Official website of PMI** Company is used as the main data source for product and department description. Some information about outdate products will be gathered from other **different sources** like Wikipedia, Numbeo.and some others. Some information like sales and date dimension will be generated manually in the **local database** and provided as .sql file that would be used in operational data store (ODS).

Staging area

Second step is to make **ETL** processes that will combine operations like:

- Extract;
- Clean;
- Conform;
- Transform
- Load.

External tables will be actively used on this stage. As the result of these operations, tables with cleaned data will be gained. Cleaned data will be sent to the ODS, where database in the 3rd normal form will be created.

Another step will be to transform 3rd normal form database into Star schema through some additional ETL processes.

Users:

SA_SRC

- starts the process of getting data from sources to external tables in the staging area;

BL_CL

- converts external (ext) tables to working (wrk) to get better performance in workink with data;
- truncates & inserts data to tables (cl);
- prepares data from external table to 3nf schema by cleaning data and allocation to appropriate tables;
- merges data with tables from 3nf layer (ce).

3NF

- operates with ready-build 3nf schema.

BL_CL2

- 2nd cleaning layer used for preparing data to DWH schema;
- truncates & inserts data to tables (CLS);
- prepares data from external table to 3nf schema by cleaning data and allocation to appropriate tables;
- merges data with tables from DWH.

DWH

- operates with ready-build DWH schema.

Warehouse area

The result of these processes will be ready-made Data Warehouse for PMI Corporation.

Methods, rules and the process of generation source data are the following:

- Product description and whole information will be taken from the PMI website and Wikipedia;
- Information about departments will be taken from additional sources, which provide appropriate information about PMI deps;
- Prices for products will be taken from the source https://www.numbeo.com/cost-of-living/prices_by_country.jsp?displayCurrency=USD&itemId=17 where there is an information about pricing in different countries. Other product pricing will be generated on the basis of this source so as to provide approximately true information about different brands of PMI;
- Pricing Index will be set up for location on the basis of source above;

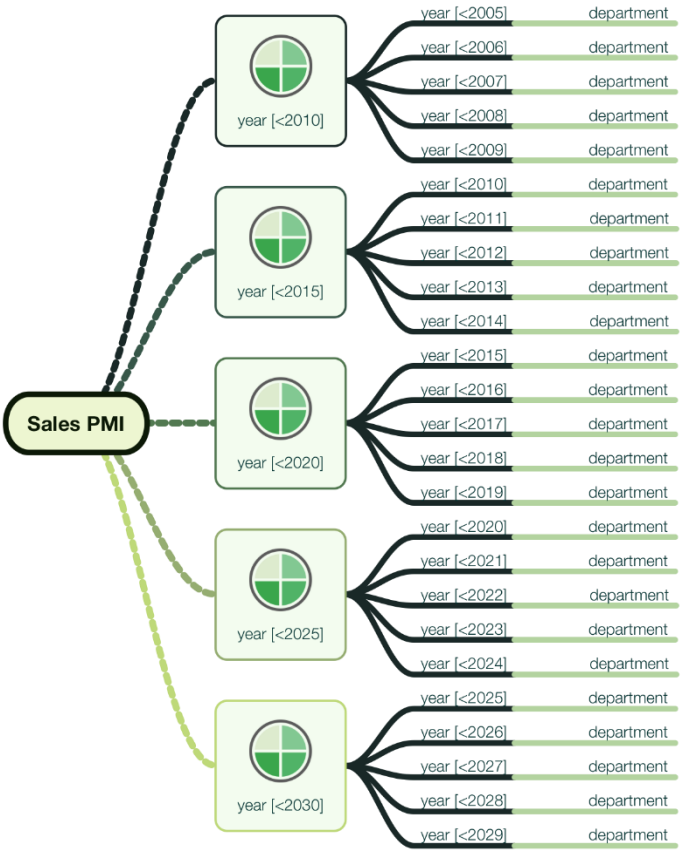
- Data on different years will be generated manually;
- Data on amount of sales will be generated manually (within the framework of global brand share);
- Data on quantity of products sold will be generated manually (within the framework of global brand share).

5. Fact Table Partitioning Strategy

Data Warehouse is going to have a great amount of records when it starts a production usage. And, moreover, the amount of records will be rising dramatically. It means that performance will be fallen every time when new sales appear and it will have been continuing till the end of 2030 year at least. So PMI Data Warehouse is needed to be correctly partitioned.

Approximately in every 5 years company make another big step in smoking industry and provides market with brand new products. And every 5 year company have a new economic plan that regulates it sales policy in the future years. So, it was decided to build partition on mentioned date (partition on every 5 years and subpartitions on every year) periods and it will be manually added range partitions. Despite it will make a great performance gain it can be not enough to manipulate with such data warehouse in a comfort way. So, additional department partitioning can be added so as to divided every 5 years by department sales in approximately equal proportions due to the techniques of hash partition.

As a result PMI data warehouse will have partition structure like it shown on the picture below:



6. Strategy of parallel load

Parallel load is an essential feature that will be used to insert data on cls level in table CLS_CONSUMERS. Consumers have been chosen because of great amount of records and large amount of records in table. So parallel insert helps to improve loading time a lot.

Screenshot below to shows the difference in executing equivalent statements:

Statistics	
643	CPU used by this session
643	CPU used when call started
1921	DB time
3	Requests to/from client
1820	enqueue releases
1822	enqueue requests
205	messages sent
3217	non-idle wait count
1279	non-idle wait time
13	opened cursors cumulative
2	opened cursors current
2122	physical read total IO requests
28729344	physical read total bytes
1	pinned cursors current
124	recursive calls
8	recursive cpu usage
4525270	session logical reads
-65536	session pga memory
203920	session uga memory
1907000	session uga memory max
2	user I/O wait time
5	user calls

Statistics	
9	CPU used by this session
9	CPU used when call started
796	DB time
3	Requests to/from client
17	enqueue conversions
20	enqueue releases
38	enqueue requests
781	in call idle wait time
10	messages sent
46	non-idle wait count
6	non-idle wait time
107	opened cursors cumulative
2	opened cursors current
2	physical read total IO requests
16384	physical read total bytes
1	pinned cursors current
221	recursive calls
3	recursive cpu usage
892	session logical reads
-393216	session pga memory
3342336	session pga memory max
-799104	session uga memory
1587096	session uga memory max
5	user calls

Insert without/with parallelism

It can be also seen the different ways of implementing same statement (different number of opened cursors and used memory)

7. Report layouts

One of the concepts of final report is shown on the screenshot below:

Picture – concept of final report

This report shows some main goals that were set on the first step of building DWH. One of the task was to find the difference between sales start of new product compared to the already known production. This question appeared because of the small profit after new production line had started.

This concept report was built in TIBCO Spotfire on the example-basis values and real production items. It includes:

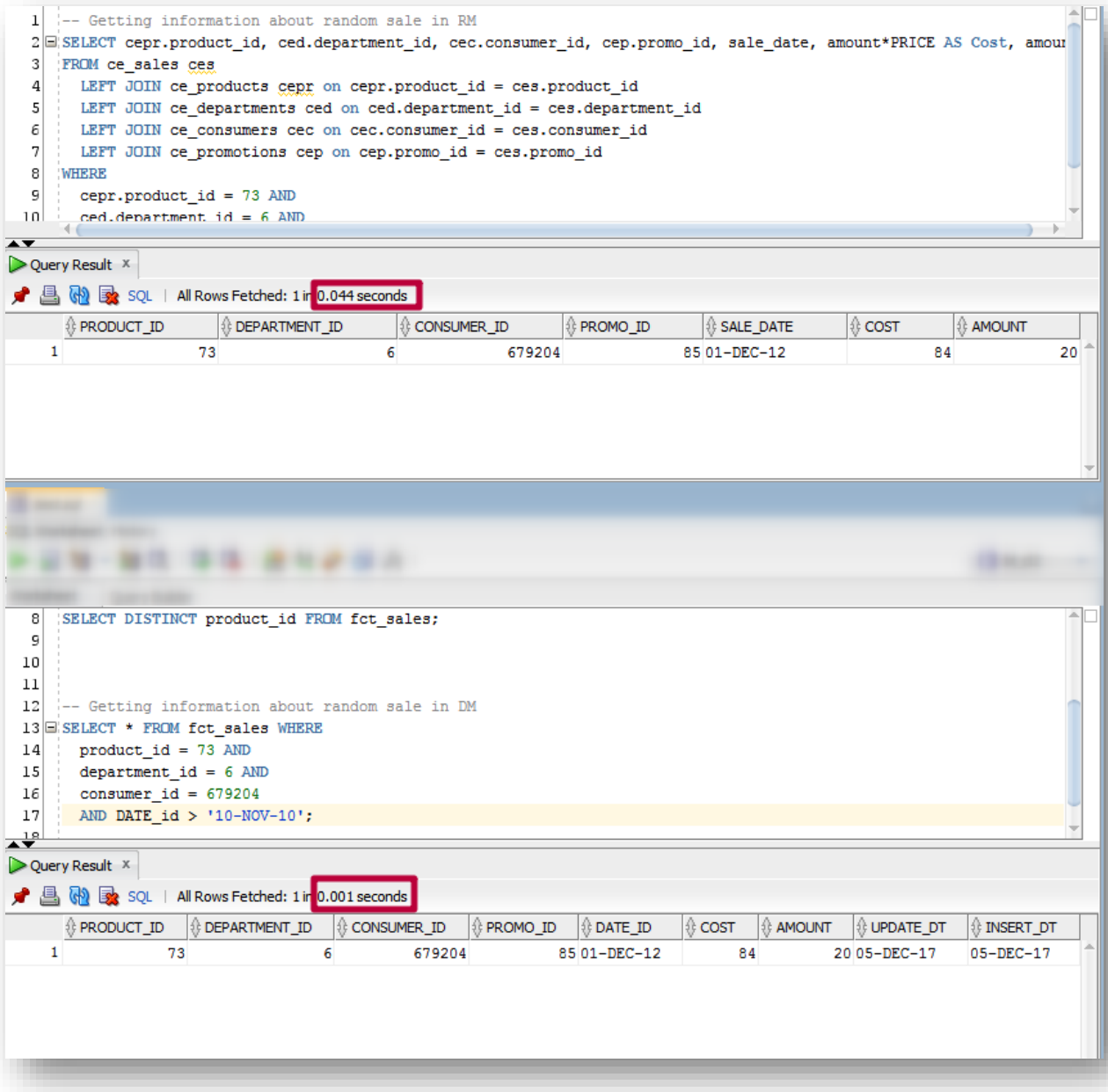
- 2 KPI charts (on the left of the screen) where
  - 1st chart shows total Quantity sold per category on its the first year on market;
  - 2nd chart shows total profit per category on its first year on market;

    * total profit is a calculated column where [quantity] was multiplied on [cost per item]
- Bar Chart (on the right of the screen) where there is a quantity and profit on value axis per products (grouped by categories). There is also a line that shows average quantity sold in the first year of the product on the market.

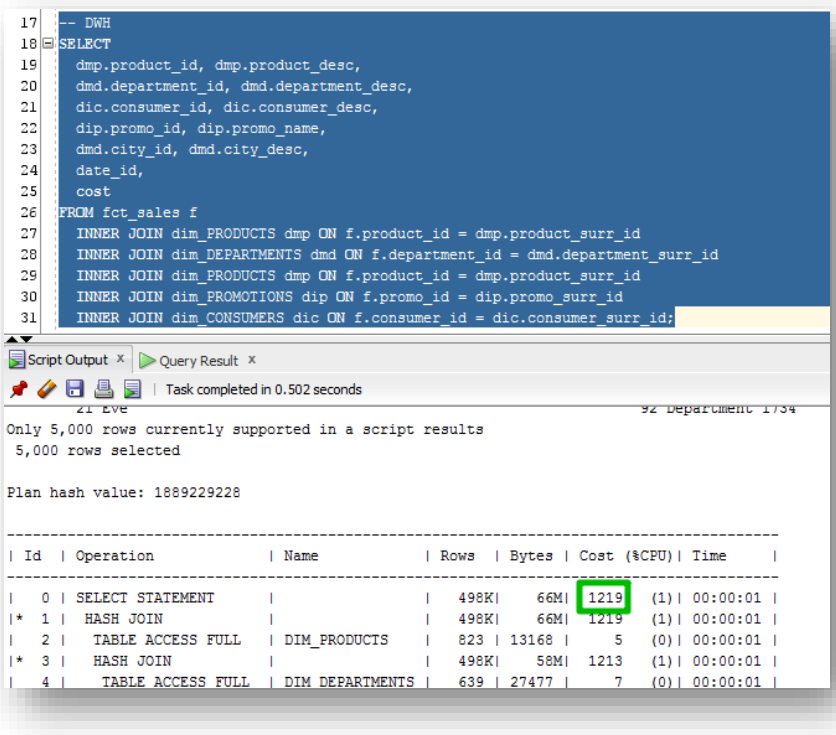
Such charts provides easy-to-understand access to the success of the products in their market sales starts.

8. Additional features

Data Warehouse brings company a great performance in getting results from database. Statements below shows that aggregate info about products in 3nf schema and dwh have the same result but great difference in speed of executing information.



Results of aggregate executions



```
-- DWH
SELECT
  dmp.product_id, dmp.product_desc,
  dmd.department_id, dmd.department_desc,
  dic.consumer_id, dic.consumer_desc,
  dip.promo_id, dip.promo_name,
  dmd.city_id, dmd.city_desc,
  date_id,
  cost
FROM fct_sales f
  INNER JOIN dim_PRODUCTS dmp ON f.product_id = dmp.product_surr_id
  INNER JOIN dim_DEPARTMENTS dmd ON f.department_id = dmd.department_surr_id
  INNER JOIN dim_PRODUCTS dmp ON f.product_id = dmp.product_surr_id
  INNER JOIN dim_PROMOTIONS dip ON f.promo_id = dip.promo_surr_id
  INNER JOIN dim_CONSUMERS dic ON f.consumer_id = dic.consumer_surr_id;
```

Script Output | Query Result

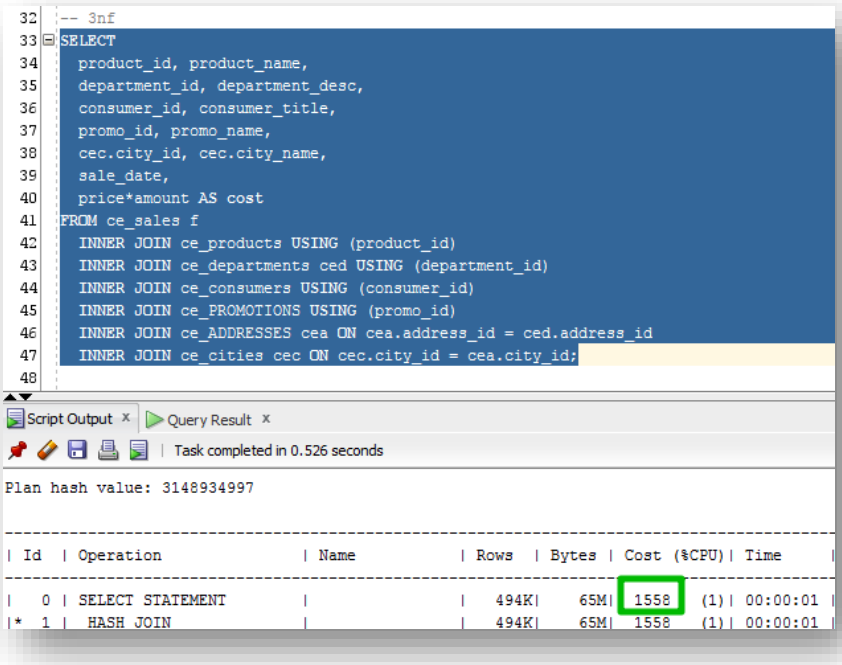
Task completed in 0.502 seconds

```
Only 5,000 rows currently supported in a script results
 5,000 rows selected

Plan hash value: 1889229228

-------------------------------------------------------------------------------------------
| Id  | Operation             | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                 |   498K|    66M|  1219   (1)| 00:00:01 |
|*  1 |  HASH JOIN            |                 |   498K|    66M|  1219   (1)| 00:00:01 |
|   2 |   TABLE ACCESS FULL   | DIM_PRODUCTS    |   823 | 13168 |     5   (0)| 00:00:01 |
|*  3 |   HASH JOIN           |                 |   498K|    58M|  1213   (1)| 00:00:01 |
|   4 |    TABLE ACCESS FULL  | DIM_DEPARTMENTS |   639 | 27477 |     7   (0)| 00:00:01 |
```

DWH cost



```
-- 3nf
SELECT
  product_id, product_name,
  department_id, department_desc,
  consumer_id, consumer_title,
  promo_id, promo_name,
  cec.city_id, cec.city_name,
  sale_date,
  price*amount AS cost
FROM ce_sales f
  INNER JOIN ce_products USING (product_id)
  INNER JOIN ce_departments ced USING (department_id)
  INNER JOIN ce_consumers USING (consumer_id)
  INNER JOIN ce_PROMOTIONS USING (promo_id)
  INNER JOIN ce_ADDRESSES cea ON cea.address_id = ced.address_id
  INNER JOIN ce_cities cec ON cec.city_id = cea.city_id;
```

Script Output | Query Result

Task completed in 0.526 seconds

```
Plan hash value: 3148934997

-------------------------------------------------------------------------------------------
| Id  | Operation             | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                 |   494K|    65M|  1558   (1)| 00:00:01 |
|*  1 |  HASH JOIN            |                 |   494K|    65M|  1558   (1)| 00:00:01 |
```

3NF cost

Other used techniques are:

- **Implicit cursor** [cls_Consumer_rate] & **Explicit cursor** [cls_Products, cls_Product_category, cls_Product_subcategory], which run on the small tables so as not to decrease performance much.

Implicit:

```
BEGIN
FOR cat IN (
  SELECT seq_product_category.nextval AS prod_category_id, prod_category FROM (
  SELECT prod_category
  FROM wrk_products
  WHERE PROD_category IS NOT NULL
  GROUP BY prod_category))
LOOP
  INSERT INTO cls_product_category (
    CATEGORY_id,
    CATEGORY_NAME)
  VALUES (
    cat.prod_category_id,
    cat.prod_category);
  END LOOP;
END;
/
```

Explicit:

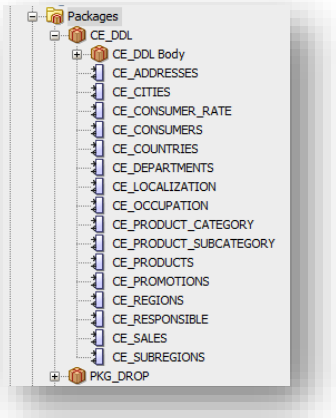
```
-- CLS_PRODUCTS
DECLARE CURSOR prod IS SELECT DISTINCT(product_code) FROM wrk_products;
p VARCHAR(10);

BEGIN
  OPEN prod;
  LOOP
    FETCH prod INTO p;
    EXIT WHEN prod%notfound;
    EXECUTE IMMEDIATE ('TRUNCATE TABLE CLS_PRODUCTS');
    INSERT INTO CLS_PRODUCTS (
      PRODUCT_CODE,
      PRODUCT_NAME,
      LOCALIZATION_CODE,
      CATEGORY_CODE,
      PRICE,
      UPDATE_DT,
      INSERT_DT)
    SELECT DISTINCT
      PRODUCT_CODE,
      PRODUCT,
      LOCALIZATION,
      PROD_CATEGORY_CODE,
      prod_price,
      sysdate,
      sysdate
    FROM WRK_PRODUCTS
      WHERE product IS NOT NULL;
  END LOOP;
  close prod;
  commit;
END;
/
```

- **Bulk collect & Forall**

```
-- CE_LOCALIZATION
PROCEDURE CE_LOCALIZATION AS
type TObjectTable is table of cls_localization%ROWTYPE;
t TObjectTable;
begin
  EXECUTE IMMEDIATE ('TRUNCATE TABLE ce_localization');
  select *
    BULK COLLECT INTO t
  from cls_localization;
forall x in t.First..t.Last
insert into ce_localization values t(x) ;
COMMIT;
end;
```

- **Partitioning**
- **Package data contribution:**



- **SCD1 & SCD2 tables**

```
-- CE_DEPARTMENTS
execute pkg_drop.DROP_proc('"BL_3NF"."CE_DEPARTMENTS"','table');
CREATE TABLE "BL_3NF"."CE_DEPARTMENTS"
(
  "DEPARTMENT_ID" NUMBER(8,2) NOT NULL,
  "DEPARTMENT_CODE" VARCHAR(10),
  "DEPARTMENT_DESC" VARCHAR2(50),
  "ADDRESS_ID" NUMBER(10),
  "START_DT" DATE,
  "END_DT" DATE,
  "IS_ACTIVE" VARCHAR(1),
  "UPDATE_DT" DATE,
  "INSERT_DT" DATE DEFAULT TRUNC(SYSDATE)
)
;

execute pkg_drop.DROP_proc('seq_departments','sequence');
CREATE SEQUENCE seq_departments
      INCREMENT BY 1
          START WITH 1
       MINVALUE 1
        NOCYCLE;
```

```
-- CE_PROMOTIONS
execute pkg_drop.DROP_proc('"BL_3NF"."CE_PROMOTIONS"','table');
CREATE TABLE "BL_3NF"."CE_PROMOTIONS"
(
  "PROMO_ID" NUMBER(8,2) NOT NULL,
  "PROMO_CODE" VARCHAR2(50),
  "PROMO_NAME" VARCHAR2(50),
  "PROMO_DESC" VARCHAR2(50),
  "PROMO_COST" NUMBER(8,2),
  "START_DT" DATE,
  "END_DT" DATE,
  "IS_ACTIVE" VARCHAR2(1),
  "UPDATE_DT" DATE,
  "INSERT_DT" DATE
)
;

execute pkg_drop.DROP_proc('seq_promotion','sequence');
CREATE SEQUENCE seq_promotion
      INCREMENT BY 1
          START WITH 1
       MINVALUE 1
        NOCYCLE;
```

- **Hierarchy**

```
SELECT * FROM dim_consumers;
SELECT * FROM dim_departments;

WITH t AS (
    SELECT
      department_id,
      region AS child_code,
      null AS parent_code
    FROM dim_departments
      UNION ALL
    SELECT
      department_id,
      country,
      region
    FROM dim_departments
      UNION ALL
    SELECT
      department_id,
      city,
      country
    FROM dim_departments)
SELECT
  ltrim(sys_connect_by_path(child_code,'==>'), '==>' ) AS path
FROM t
START WITH
  parent_code IS NULL
CONNECT BY
  PRIOR child_code = parent_code;
```

Script Output × Query Result ×

SQL | Fetched 50 rows in 0.029 seconds

	PATH
1	Americas
2	Americas==>Canada
3	Americas==>Canada==>Toronto
4	Americas==>United States of America
5	Americas==>United States of America==>Seattle