THE UNIVERSITY OF MELBOURNE

ISYS90086
Data Warehousing

Summer Semester 2019

Assignment 2-ETL PROCESS

Dinghao Yong	**868878**
Min Xue	**897082**

Content

Executive Summary

This report mainly focuses on the ETL process which is generally considered as one of the most time-consuming tasks during the establishment of the data warehousing. All procedures in terms of extracting data from multiple sources, transforming data into required types and formats and loading data into the data warehouse will be described in detail.

During the transformation process, some issues like enabling data quality, calculating derived values of some specific attributes, integrating data from different sources into one table and solving the slow changing dimensions (SCD) problem should be considered so as to reduce the processing time and get optimal performance. A useful tool named Pentaho is used for dealing with ETL transformations on each fact table and dimension table and addressing all of those issues mentioned above. When it comes to addressing the SCD problem, there are several ways given by Kimball to handle it. One of the basic types is chosen as the solution under this circumstance which described as creating a new row whenever the value changes and keeping the old values for later analysis.

In addition, some adjustments need to be conducted on the data warehouse to fit operational requirements. Several additional metadata are listed in the data dictionary with their types and reasons for adding them.

Design of the ETL Process

2.1 Date Table Transformation

Figure 1. Date Transformation Process

As the figure 1 show, the transformation process for Date Dimension includes two steps. The first step is to input the source data form the source *xlsx* file *Date.xlsx* and the second one is to output the data into the target table.
The mapping fields are listed blew:

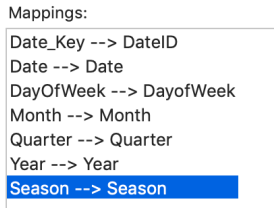


Figure 2. Fields Mapping of Output Step in Date Transformation

Part of the final output:

DateID	Date	DayofWeek	Month	Season	Quarter	Year
1	2015-01-01	Thursday	January	Summer	1	2015
2	2015-01-02	Friday	January	Summer	1	2015
3	2015-01-03	Saturday	January	Summer	1	2015
4	2015-01-04	Sunday	January	Summer	1	2015
5	2015-01-05	Monday	January	Summer	1	2015
6	2015-01-06	Tuesday	January	Summer	1	2015
7	2015-01-07	Wednesday	January	Summer	1	2015
8	2015-01-08	Thursday	January	Summer	1	2015
9	2015-01-09	Friday	January	Summer	1	2015
10	2015-01-10	Saturday	January	Summer	1	2015
11	2015-01-11	Sunday	January	Summer	1	2015
12	2015-01-12	Monday	January	Summer	1	2015
13	2015-01-13	Tuesday	January	Summer	1	2015
14	2015-01-14	Wednesday	January	Summer	1	2015
15	2015-01-15	Thursday	January	Summer	1	2015
16	2015-01-16	Friday	January	Summer	1	2015
17	2015-01-17	Saturday	January	Summer	1	2015
18	2015-01-18	Sunday	January	Summer	1	2015
19	2015-01-19	Monday	January	Summer	1	2015
20	2015-01-20	Tuesday	January	Summer	1	2015
21	2015-01-21	Wednesday	January	Summer	1	2015

Figure 3. Partial final output of Date Transformation

2.2 Employee Table Transformation

Figure 4. Employee Table Transformation

In the transformation process of employee table, there are 7 steps. The first step is to input the source CSV file, *SalesPerson.csv*. The step 2 to step 5 is to transform the data type of commission rate. After input step, the format of commission rate is 'XX%' and the data type is String. With step2 to step 7, we transform the field type into Decimal(4,2). In these steps, we remove the symbol '%', transform

the String value to Number and then divide it by 100. After sorting the table by ID, we output the final table.

The calculation step:

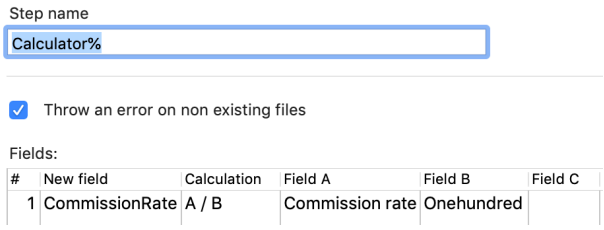


Figure 5. The calculation step in Employee Transforamtion

The output table:

EmployeeID	Name	CommisionRate
B1	Supradeek Densiman	0.20
B2	Arit Arubne	0.12
B3	Flame Blower	0.07
B4	Michelle Nguyen	0.07
D1	Hi Min Chow	0.19
D2	Peter Jones	0.08
D3	Aimee Concroan	0.07
D4	Jan Kennedy	0.04
M1	Alice McPherson	0.09
M2	Pjan Ling	0.03
S1	Willy Wonka	0.18
S2	Quin Tan	0.05

Figure 6. Employee Table Result

2.3 Product Table Transformation

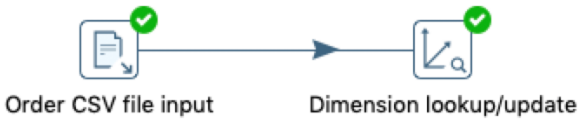


Figure 7. Product Table Transformation

In the transformation process of product table, the main step is the dimension lookup/update process. As the order price of each product is different at every time when purchasing products from the suppliers, the product dimension is a slow changing dimension. In this case, we need to add the version field to record the change. After the input of source CSV file, *ProductOrder.csv*, we perform the dimension update step.

In the dimension update step, we set a ValidFromDate and a ValidUntilDate. When a purchasing happens, new versions of corresponding items will be added. The product_rid, a surrogate key, is an auto increment key generated by the system in this step.

The details of the dimension lookup/update step:

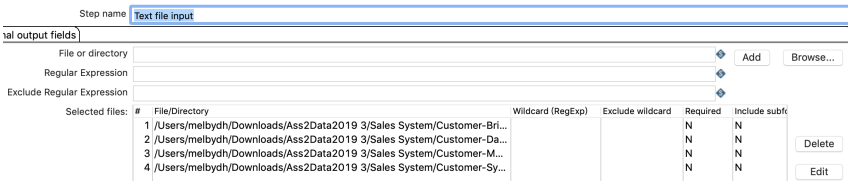


Figure 11. Input of four txt files

As we need to record the age of all customers to solve the business problem, a Calculator step is added to calculate the age value. The calculation method is as below:

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type
1	BirthYear	Year of date A	Date_of_Birth			Integer
2	ThisYear	Set field to constant val...	2019			Integer
3	Age	A - B	ThisYear	BirthYear		Integer

Figure 12. Calculation of Age value in Customer Dimension

In the transformation process of customer table, the main step is the dimension lookup/update process. As the customers may change their postcode, the customer dimension is a slow changing dimension. In this case, we need to add the version field to record the change. In the dimension lookup/update step, we set a ValidUntilDate. When one customer move their houses, new versions of corresponding customers will be added. Customer_rid, a surrogate key, is an auto increment key generated by the system in this step.

After finishing the Dimension Lookup/update, we output the data to another table as we do not need some values such as *date_from* and *date_to* in the result of update step. The final output table is as below:

Customer_...	CustCode	Name	Date_Of_Birth	Postcode	Validuntil	Age
2	C1959Bris	Chee Jorge Samson	1997-02-10	4053	NULL	22
3	C1961Bris	Robert Sacks Romero	1974-04-14	4010	NULL	45
4	C1969Bris	Aleck Chia Saha	1957-10-18	4117	NULL	62
5	C196Bris	Matthew Tang Chan	1966-08-09	4058	NULL	53
6	C1973Bris	Tsz Munajat Saha	1952-04-18	4130	NULL	67
7	C1976Bris	Woo-Jung Cao Salim	1958-03-13	4080	NULL	61
8	C1977Bris	Yingshuang John Sangsurane	1951-06-19	4066	NULL	68
9	C1978Bris	Bin Kin Sangsurane	1963-05-31	4155	NULL	56
10	C1979Bris	Eddie Francis Salim	1999-06-11	4108	NULL	20
11	C197Bris	Shenn Moon Chan	1999-09-07	4089	NULL	20
12	C1980Bris	Maree Cao Sau	1954-06-25	4002	NULL	65
13	C1981Bris	Miu Gerald Sau	1993-10-08	4076	NULL	26
14	C1982Bris	Robert Ling Sangsurane	1996-08-16	4047	NULL	23
15	C1984Bris	Charles Mulki Sam	1949-03-27	4114	NULL	70
16	C1985Bris	Hong-Lim Jun Sanjaya	1971-03-28	4028	NULL	48
17	C1986Bris	Lingjie David Schmidt	1949-02-04	4044	NULL	70
18	C1989Bris	Lillian Sung-Chit Schmidt	1998-07-25	4071	NULL	21
19	C198Bris	Yu Sie Chai	1952-11-06	4063	NULL	67
20	C1992Bris	Jingyi Zhai Sankranti	1978-08-27	4026	NULL	41
21	C1996Bris	Lye Kooy Septina	1960-05-04	4141	NULL	59
22	C1997Bris	Ruichen Ruth Scully	1965-10-19	4118	NULL	54
23	C199Bris	Anne Paul Chan	1995-01-30	4036	NULL	24
24	C2000Bris	Ankur Raso Setiawan	1953-06-07	4058	NULL	66
25	C2002Bris	Bei Yu Shah	1999-12-25	4073	NULL	20
26	C2003Bris	Serey Meng Seah	1999-12-08	4121	NULL	20
27	C2005Bris	Tou Li Shah	1982-04-02	4044	NULL	37
28	C2008Bris	Elvie Elizabeth Shakespeare	1990-11-30	4011	NULL	29
29	C2009Bris	Zhenya Ngu Shek	1956-10-30	4013	NULL	63
30	C200Bris	Brianna Sachdeva Chan	1954-12-08	4068	2017-05-...	65
31	C200Bris	Brianna Sachdeva Chan	1954-12-08	4020	NULL	65

Figure 12. The Output of Customer Table Transformation

2.5 Store Table Transformation

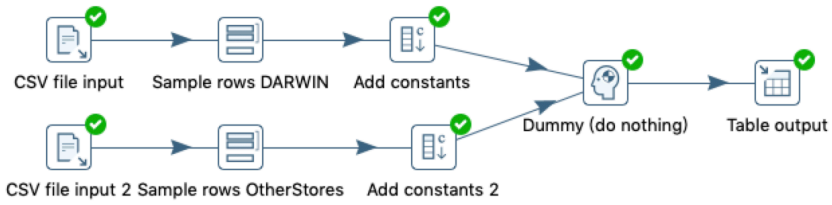


Figure 13. The Process of Store Table Transformation

As the Case Study document of Assignment 1 states, only one store, which locates in Darwin, opens on Sundays. We need to record this information in the store table. In order to reach this, we input two times and then select lines of Darwin or Other Stores respectively. After that, we add the AvaOnSunday values. We set this value in line of Darwin as 'Y' and set the value in lines of other stores as 'N'. Then we use a dummy step to combine the rows into one table. The last step is the output.

The Select Step and Add Constants Step of Store in Darwin:

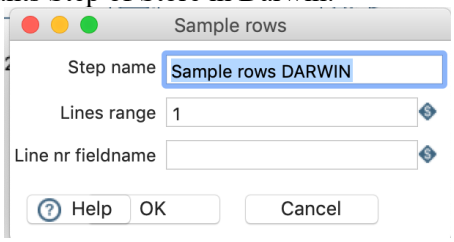


Figure 14. Select Darwin Line from the Input

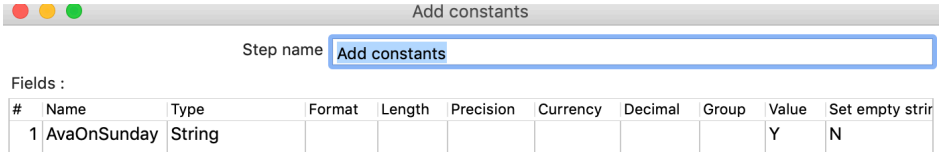


Figure 15. Set the Value in Darwin Line as 'Y'

The Select Step and Add Constants Step of Stores in Other Cities:

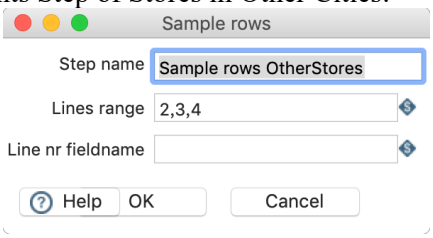


Figure 16. Select Lines of Other Stores from the Input

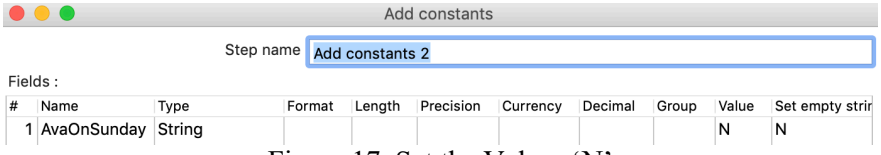


Figure 17. Set the Values 'N'

The output table of this transformation:

StoreID	City	Address	AvaOnSun
1	DARWIN	19 Finniss St, Darwin, NT 0800	Y
2	BRISBANE	23 Wellington St, Brisbane, QLD 4000	N
3	SYDNEY	233 Macquarie St, Sydney, NSW 2000	N
4	MELBOURNE	123 Latrobe St, Melbourne, VIC 3000	N

Figure 18. The Output of Store Table Transformation

2.6 Sales_Commission Table Transformation

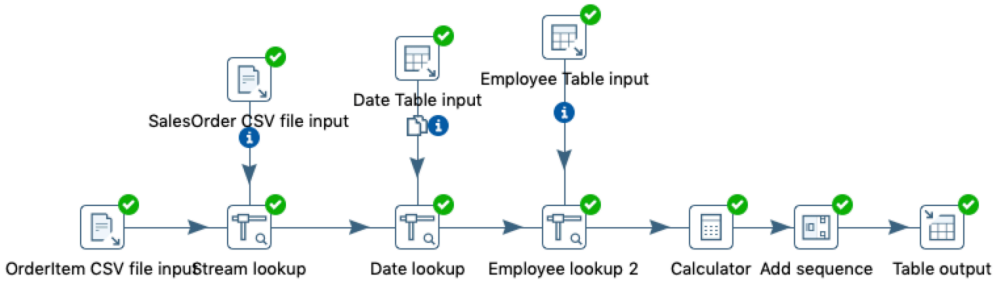


Figure 19. The Process of Sales_Commission Table Transformation

In our design, a row in the table Sales_commission represents that one paticular employee brings some amount dollar sales to the shop when selling out something. In this case, this fact table has two dimensions, which are corresponding to the existed table in the data warehouse, Date and Employee. At the very beginning, we need to input the sales information, which is recorded in two CSV files, *Order.csv* and *OrderItem.csv*. In order to make them one table, a Stream Lookup step is performed. The key to look up the value is 'OrderID' in this lookup step.
The details are as below in the first lookup step:

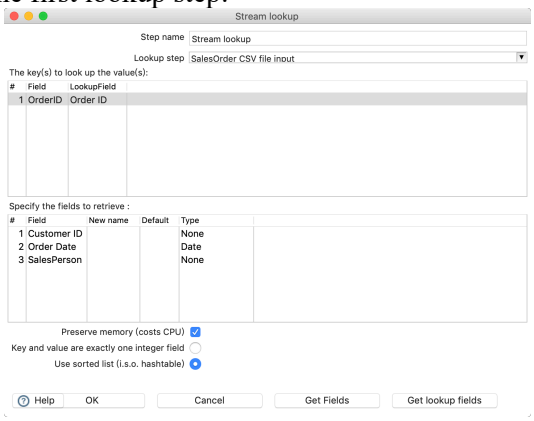


Figure 20. The First Lookup step in Sales_commission Transformtaion

Then we perform other two Stream Lookup steps to get the DateID and Commission Rate respectively. We get the DateID by looking up the Order Date and get the Commission Rate by looking up the SalesPerson. The details are as below:

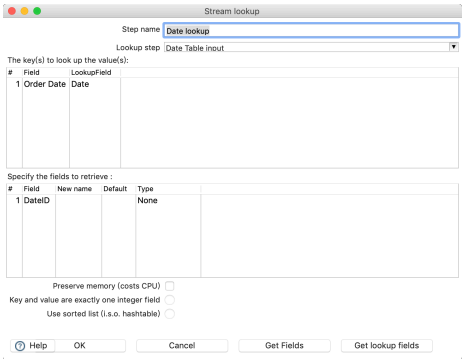


Figure 21. The Date Lookup Step

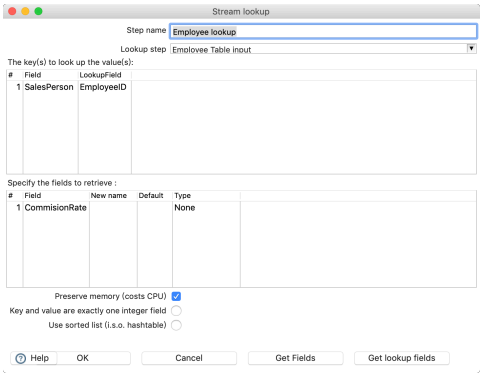


Figure 22. The Employee Lookup Step

After finishing all the Stream Lookup steps, we calculate the dollar sales and the commission. In this step, *DollarSales* is calculated by *Units* multiplies *UnitPrice* and *Commission* equals to *DollarSales* multiplies *Commission Rate.* The details are as below:

Step name

Calculator

Throw an error on non existing files

Fields:

#	New field	Calculation	Field A	Field B	Field C
1	DollarSales	A * B	UnitPrice	Units	
2	Money	A * B	DollarSales	CommisionRate	

Figure 23. Calculation of DollarSales and Commission

Before output, an Add Sequence Step is performed to add incremental values to work as primary keys in output table.
Figure 24 is the screenshot of output table.

SalesID	Employee_EmployeeID	Date_DateID	Dollar_Sales	Unit_Sales	Commission
1	B1	1	23.98	11	4.80
2	B1	1	117.00	20	23.40
3	B1	1	80.25	15	16.05
4	B1	1	151.36	4	30.27
5	B1	1	4619.04	48	923.81
6	B1	1	1240.80	15	248.16
7	B1	1	178.64	22	35.73
8	B1	1	23.10	7	4.62
9	B1	1	227.48	2	45.50
10	B1	1	23.08	1	4.62
11	D4	1	759.92	14	30.40
12	D4	1	21.72	3	0.87
13	D4	1	204.33	7	8.17
14	B2	1	127.84	4	15.34
15	B2	1	38.94	3	4.67
16	B2	1	281.52	51	33.78
17	B2	1	347.90	14	41.75
18	B2	1	125.19	9	15.02
19	B2	1	1734.56	74	208.15
20	B2	1	52.50	6	6.30
21	B2	1	190.19	13	22.82
22	B2	1	978.56	32	117.43
23	B2	1	558.36	27	67.00
24	B2	1	0.48	1	0.06
25	B2	1	93.84	8	11.26
26	B2	1	614.57	11	73.75
27	B2	1	78.48	12	9.42
28	B2	1	499.20	13	59.90
29	B2	1	302.72	8	36.33
30	B2	1	1334.76	12	160.17

Figure 24. The output of Sales_commission Transformation

2.7 Sales_margin Table Transformation

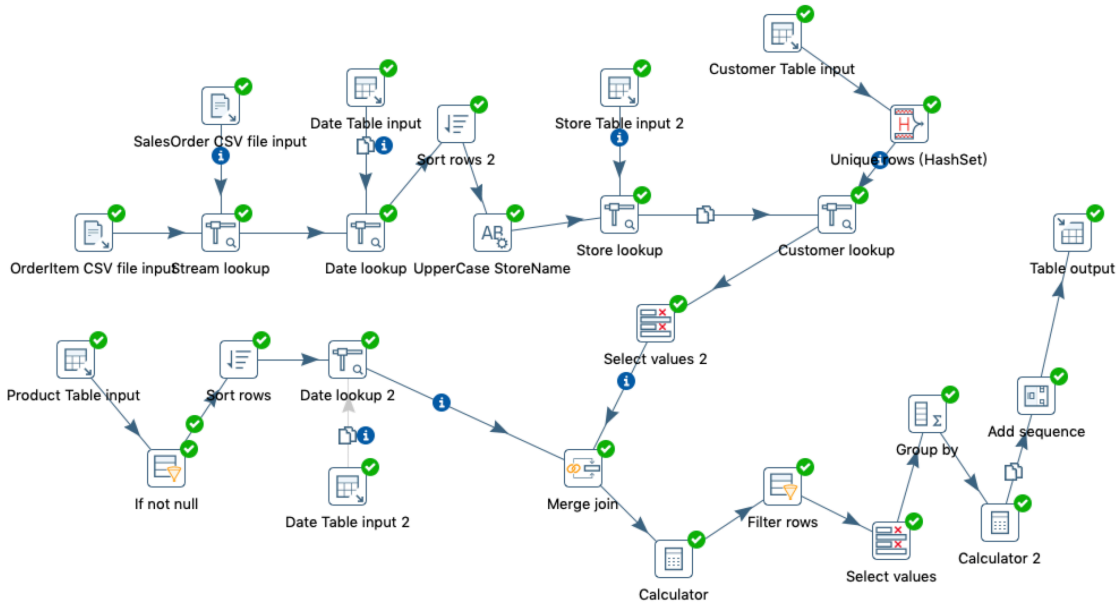


Figure 25. The Process of Sales_Margin Table Transformation

In this transformation, like the Sales_commission Transformation in section 2.6, some Stream Lookup steps are needed to get the *DateID*, *StoreID* and *Customer_rid*. The special one is the Customer part. As the Customer dimension is a slow changing dimension, a *CustCode* may be corresponding to several *Customer_rid*s. In this case, an Unique Rows(HashSet) step is performed to remove the lines, in which versions are not the latest.

In order to calculate the margin, cost of the products is needed. As the Product dimension is a slow changing dimension, which means the same products may have different cost value at different time, we need to calculate the cost of the items according to the *Sales Date.* To reach this, at first, we perform a date lookup to get the DateID corresponding with the Date value in the product table. Then we perform a Merge Join step to merge the two parts, sales part and the product part . The details are as below:

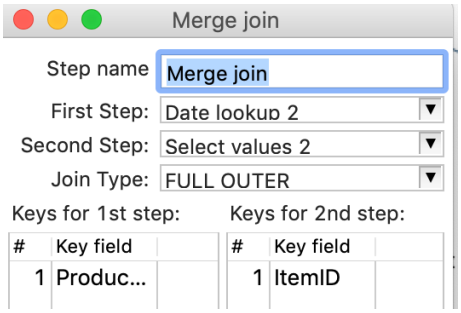


Figure 26. Merge Join Step

Then we calculate the date difference by subtracting DateID of inventory from DateID of sales:

Fields:

#	New field	Calculation	Field A	Field B	Field C
1	DateDifference	A - B	SalesDateID	DateID	

Figure 27. Calculator

A Filter Rows step is performed to remove the lines with negative values:

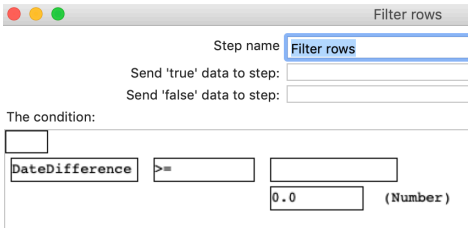


Figure 28. Filter Rows

After removing the negative values, we group the lines by *OrderID* and *ProdcutID*, and keep the row with the minimum value of *DateDifference* for every group. With the above steps, we get the cost value for all the rows in the OrderItem table.

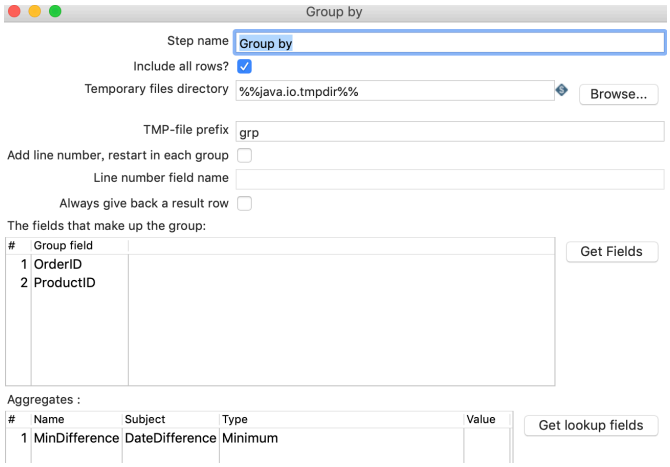


Figure 29. Group By Step

Then we calculate the *DollarSales* by multiplying *Units* and *Unitprice,* and calculate the *MarginPerItem* by subtracting *Cost* from *Unitprice* and calculate *Margin* by multiplying *Units* and *MarginPerItem.*

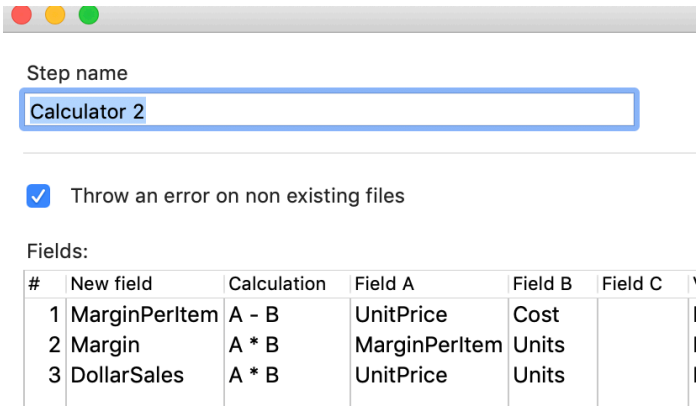


Figure 30. Calculator 2

After adding ID sequence, the output step is performed. The fields mapping:

Mappings:

Margin --> Margin
Customer_rid --> Customer_Customer_rid
Product_rid --> Product_Product_rid
StoreID --> Store_StoreID
SalesDateID --> Date_DateID
Units --> Unit_sales
DollarSales --> Dollar_Sales
id --> SalesID

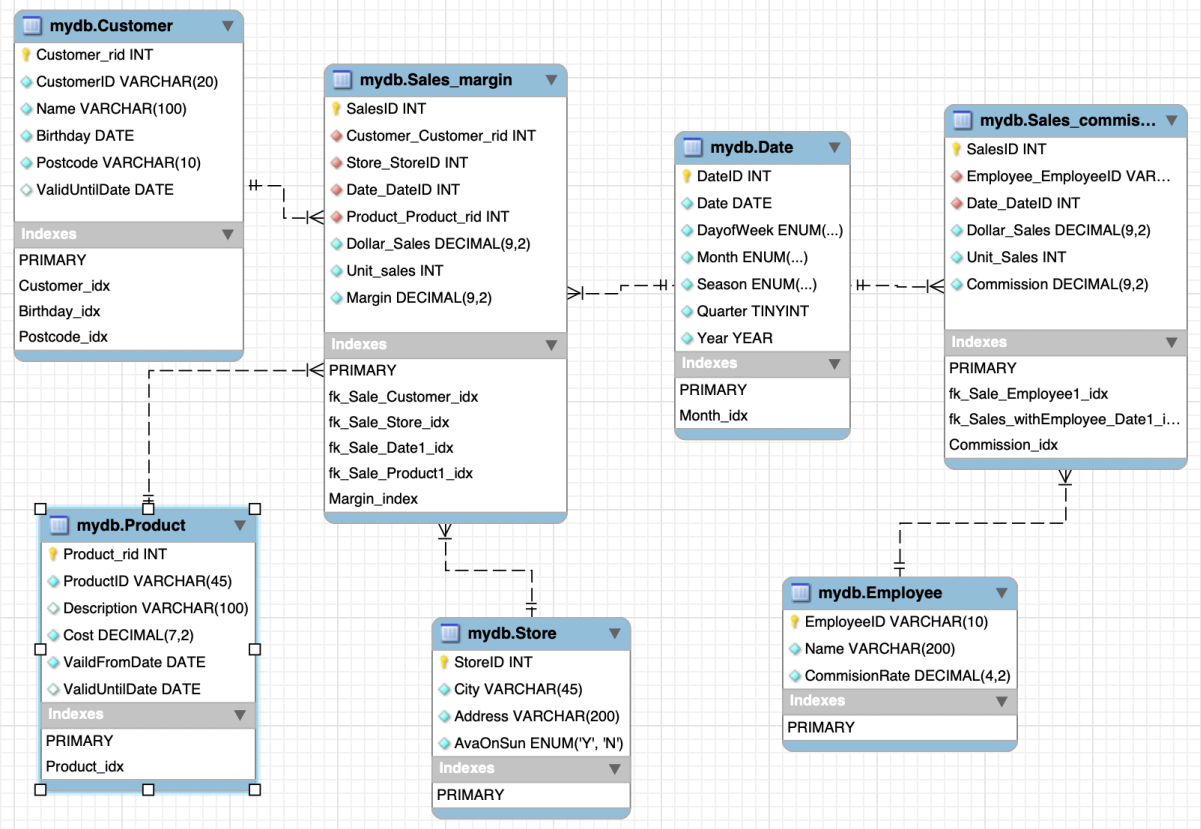
Figure 31. Fields Mapping

The output:

SalesID	Customer_Customer_rid	Store_StoreID	Date_DateID	Product_Product_rid	Dollar_Sales	Unit_sales	Margin
1	1608	2	170	1452	277.27	7	49.28
2	3413	3	175	1452	176.04	4	45.76
3	4	2	184	1452	1320.30	30	343.20
4	2051	1	187	1452	561.15	15	72.60
5	2706	1	200	1452	484.11	11	125.84
6	1402	2	209	1452	710.79	19	91.96
7	2790	1	217	1452	660.15	15	171.60
8	2356	1	218	1452	1188.30	30	211.20
9	189	2	220	1452	176.04	4	45.76
10	1491	2	224	1452	308.07	7	80.08
11	1519	2	227	1452	448.92	12	58.08
12	166	2	229	1452	572.13	13	148.72
13	2047	1	235	1452	118.83	3	21.12
14	504	2	239	1452	79.22	2	14.08
15	221	2	247	1452	1100.25	25	286.00
16	3099	4	250	1452	299.28	8	38.72
17	3174	4	253	1452	594.15	15	105.60
18	374	2	260	1452	44.01	1	11.44
19	2207	1	268	1452	1197.12	32	154.88
20	2765	1	274	1452	176.04	4	45.76
21	2133	1	286	1452	396.10	10	70.40
22	1864	2	286	1452	88.02	2	22.88
23	1255	2	290	1452	132.03	3	34.32
24	3652	3	300	1452	336.69	9	43.56
25	1382	2	303	1452	2200.50	50	572.00
26	1547	2	308	1452	352.08	8	91.52
27	2055	1	312	1452	277.27	7	49.28
28	1932	1	332	1452	264.06	6	68.64
29	1976	1	337	1452	220.05	5	57.20

Figure 32. The output of Slaes_margin Transformation

Design of the data warehouse



Redesign of the data warehouse

Sales_margin fact table

The original design of the Sales_margin fact table is relatively reasonable for addressing business problems in terms of the key customers and the profits. The adjustment is only applied to the indexes among which an index relates to the margin attribute is added. Since the values of the margin attribute are used frequently in solving business problems, the append of index on the margin improves the efficiency in querying and also leads to better performance.

Customer dimension table

Only several adjustments are conducted in the customer dimension table, including merging both FirstName and LastName attributes into one Name attribute of the customer which reduces the complexity of transformation in the following ETL process. In addition, the Address and Suburb attributes are deleted as they do not contribute to addressing business problems, the remaining of the Postcode attribute is quite enough.

Product dimension table

Based on the data stored in sales and inventory system, both the Name of the product and the Supplier attribute are removed from the product dimension table since there is no relevant information recorded and they have nothing to do with problems solving.

Date dimension table

One attribute named Date is added into the date dimension table which stores the date of each sale transmission. It is difficult to do the ETL without the date information because the date values are used to integrate tables during the transformation process. Besides, the attribute HalfYear is moved from the table which is a useless grain to measure the date.

Store dimension table

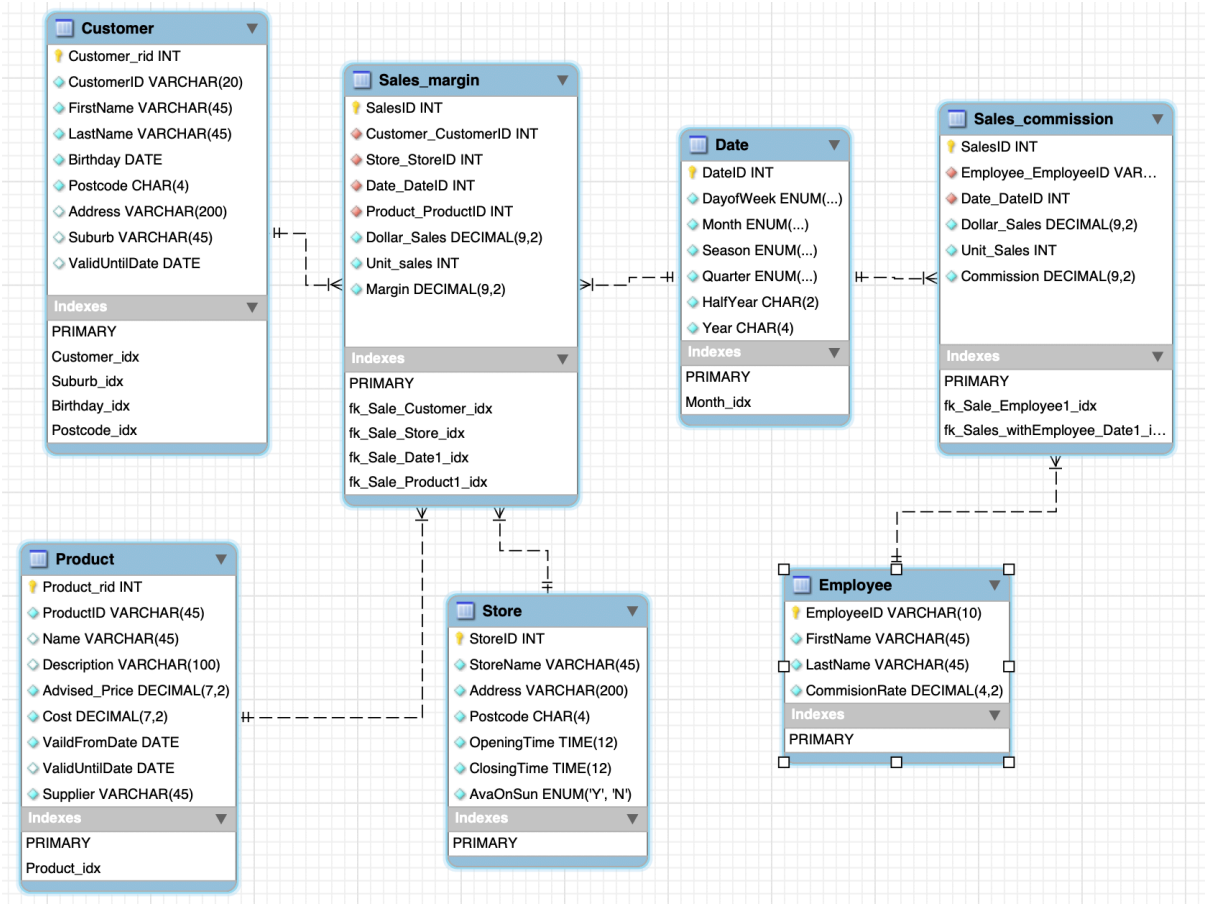
Some details about stores are deleted in the store dimension table, such as the StoreName, the Postcode and business hours. Not only are them not provided in the source system but also they are not relevant to the purpose of designing the data warehouse. In contrary, the City attribute is added as a general description for the location of all stores.

Employee dimension table

The whole name of each employee is stored in an attribute Name instead of separating them as FirstName and LastName and the reason for this integration is the same as the operation in customer dimension table.

Sales_commission fact table

Since the commission attribute is used frequently during the evaluation of the key employees, an index is appended to the commission attribute stored in the Sales_commission fact table.



Original version of the data warehouse

Data Dictionary

Sales_margin fact table

Data Type	Description
Aliases	Sales record
Definition	A record of each business transaction
Remarks	The Sales_margin fact table includes all historical and current transactional data
Source	The source of the Sales_margin fact table is coming from both the Sales system and the Inventory system, including tables of customer, store, order and product.
Update Cycle	The data of the Sales_margin fact table is updated when a new business transaction is added.
Responsible Users	Digger (Sparky) Lightfinger owns the data stored in the Sales system and Kim (Firery) Ng owns the data stored in the Inventory system

Customer dimension table

Data Type	Description
Aliases	client
Definition	A person or an organization that purchases fireworks from the Fantastic Fireworks store
Remarks	The information about customers are stored separately based on their location
Source	The source of the customer dimension table is coming from customer tables stored in the Sales system
Update Cycle	The data of the customer dimension table is updated every 6 months
Responsible Users	Digger (Sparky) Lightfinger

Product dimension table

Data Type	Description
Definition	An item that is for sale in the Fantastic Fireworks stores
Remarks	The cost of each product may vary in different purchases, hence timestamp should be added for each buying
Source	The source of the product dimension table is coming from ProductOrder table stored in the Inventory system

Update Cycle	The data of the product dimension table is updated for each buying
Responsible Users	Kim (Firery) Ng

Date dimension table

Data Type	Description
Definition	The record of time for each sale and buying and is described daily, monthly, seasonally, quarterly and yearly
Source	The source of the date dimension table is coming from the Date table
Update Cycle	The data of the date dimension table is updated daily

Store dimension table

Data Type	Description
Definition	A store belongs to the Fantastic Fireworks company
Source	The source of the store dimension table is coming from Store table stored in the Sales system
Update Cycle	The data of the store dimension table is updated if a new store is added
Responsible Users	Digger (Sparky) Lightfinger

Employee dimension table

Data Type	Description
Aliases	Sales Person
Definition	A person that is hired by the Fantastic Fireworks stores for selling products
Source	The source of the employee dimension table is coming from Sales Person table stored in the Sales system
Update Cycle	The data of the employee dimension table is updated if a new sales person is hired
Responsible Users	Digger (Sparky) Lightfinger

Sales_commission fact table

Data Type	Description

Aliases	Employee sales record
Definition	A record of employee sales and commission rate of each employee
Remarks	The Sales_commission fact table includes all employee sales
Source	The source of the Sales_commission fact table is coming from the Sales system
Update Cycle	The data of the Sales_margin fact table is updated when a new business transaction is added.
Responsible Users	Digger (Sparky) Lightfinger

Word Counts: 2236

Appendix 1-Work Breakdown

Both members participate in the redesign of the data warehouse, the ETL process and the accomplishment of the report.

To be more specific, Dinghao Yong (868878) completes following parts:

1. Redesign of the data warehouse;
2. Conduct the ETL process;
3. Complete the report in terms of the ETL process.

Min Xue (897082) finishes tasks below:

1. Redesign of the data warehouse;
2. Conduct the ETL process;
3. Complete the report in terms of executive summary, redesign of the data warehouse and data dictionary.