THE UNIVERSITY OF MELBOURNE

ISYS90086 Data Warehousing

Summer Semester 2020

Assignment 2 – Data Warehouse Load Assignment

Jason Jia

Due date: 07-Feb-2020, 9.00pm AEDT

Content

1. Executive summary

This report offers a design of ETL process and a re-design of data warehouse for Overhill Winery based on their existing information systems to meet its business requirements. By Pentaho Data Integration Kettle, given data were cleaned, transformed and integrated in seven transformations. Throughout the ETL design process, data quality issues raised our attention that we made tremendous efforts to unify the format of data, remove duplicates and invalid records, etc.

Type 2 of slowly changing dimension (SCD) was designed to dimension 'Customer', 'Product' and 'Sales Agent', which could record future data changes. Fact table contains fields from dimension table as well as measures (dollar_sales, Commission, Total_Cost and Margin) to facilitate data analysis. All transformed data were read to the data warehouse, business users will be able to access all desired information in the fact table.

In the redesigned data warehouse model, the number of dimensions in the star schema was reduced to five. Primary keys of each dimension are replaced by surrogate keys instead of the business keys. To illustrate the purpose of each transformation job, data dictionaries were attached to the end of this report. After the successful implementation of transformations, we believe that the ETL processes and star-schema design can meet the client's requirements.

2. Design of the ETL Process

2.1 Date Dimension Transformation

The date dataset has consistent data and uniform format in each field. The date column should be highlighted that it is a “date” format that can be read by Pentaho directly. As the date dimension table can be commonly yielded from most systems and databases, which ensures the quality of data. We developed a rather simple transformation process.

According to the given data, the DataNum can be seen as the natural key to each date, which helps the Overhill business to manipulate for analysis. For each row, date derivatives were given for the convenience of analysis. However, the business requires us to analyze the most profitable products and key customers by season. Thus, we mapped four calendar quarter to four seasons which quarter one mapped to Autumn and so forth. The below picture shows the transformation of the date dimension.

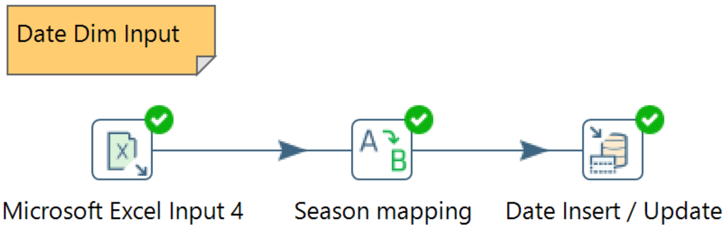
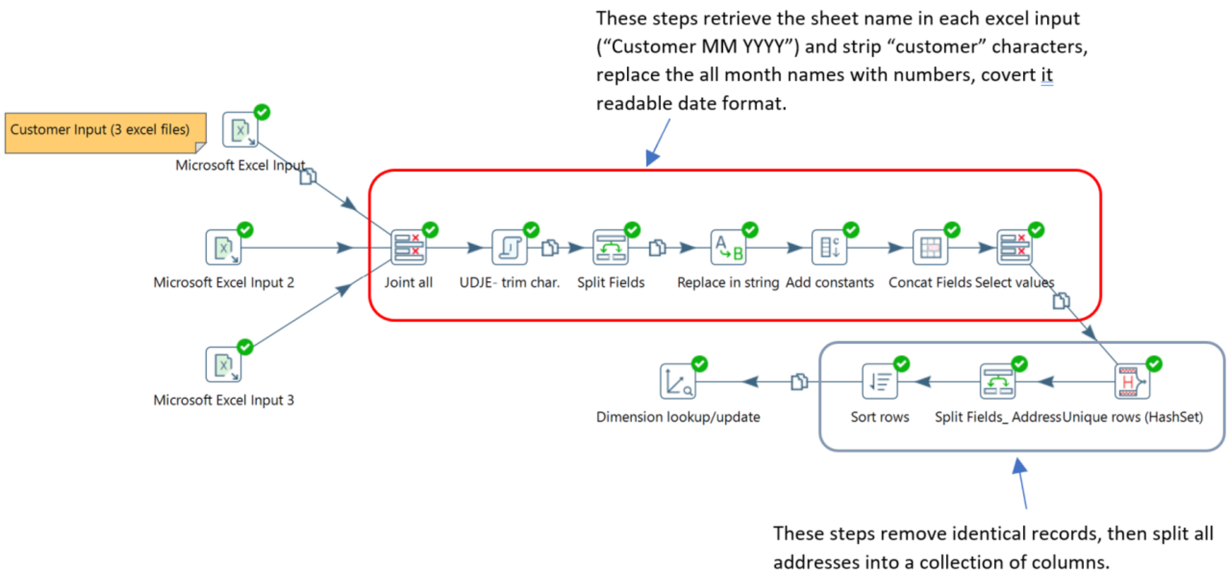


Figure 1 Date transformation

2.2 Customer Dimension Transformation

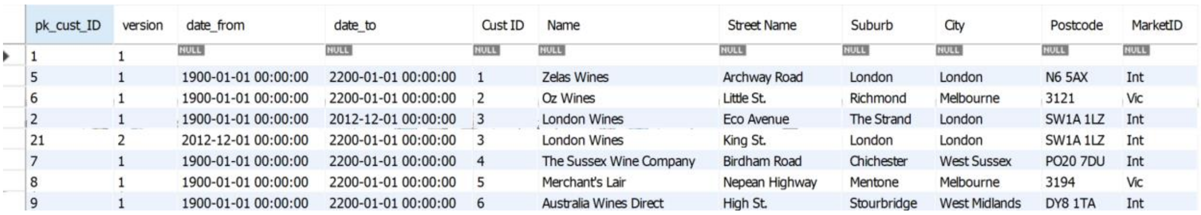


These steps retrieve the sheet name in each excel input (“Customer MM YYYY”) and strip “customer” characters, replace the all month names with numbers, covert it readable date format.

These steps remove identical records, then split all addresses into a collection of columns.

Figure 2 Customer transformation

Customer details were given in three datasets that ought to be aggregated and imported in ETL process. We used “Cust ID” as a natural to identify each customer. The customer address ought to be split into “street”, “suburb”, “city” and “postcode”. We observed identical customer records exist throughout all given files. To cleanse the identical data, we adopted two approaches. We firstly retrieved and appended time details from each sheet of excel file. The string-type time details were then stripped and converted to a readable date format. Additionally, these steps enable the ETL process to read string-type time details from future incremental batches without any manual effort. The second approach utilized the “dimension lookup” component to feature version number and date-from/date-to for each customer if any change occurred. The above picture describes the transformation of the customer dimension. Having this transformation implemented successfully, a snapshot of the customer dimensional table is attached below.



pk_cust_ID	version	date_from	date_to	Cust ID	Name	Street Name	Suburb	City	Postcode	MarketID
1	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	1	1900-01-01 00:00:00	2200-01-01 00:00:00	1	Zelas Wines	Archway Road	London	London	N6 5AX	Int
6	1	1900-01-01 00:00:00	2200-01-01 00:00:00	2	Oz Wines	Little St.	Richmond	Melbourne	3121	Vic
2	1	1900-01-01 00:00:00	2012-12-01 00:00:00	3	London Wines	Eco Avenue	The Strand	London	SW1A 1LZ	Int
21	2	2012-12-01 00:00:00	2200-01-01 00:00:00	3	London Wines	King St.	London	London	SW1A 1LZ	Int
7	1	1900-01-01 00:00:00	2200-01-01 00:00:00	4	The Sussex Wine Company	Birdham Road	Chichester	West Sussex	PO20 7DU	Int
8	1	1900-01-01 00:00:00	2200-01-01 00:00:00	5	Merchant's Lair	Nepean Highway	Mentone	Melbourne	3194	Vic
9	1	1900-01-01 00:00:00	2200-01-01 00:00:00	6	Australia Wines Direct	High St.	Stourbridge	West Midlands	DY8 1TA	Int

Figure 3 Customer dimensional table

2.3 Market Dimension Transformation

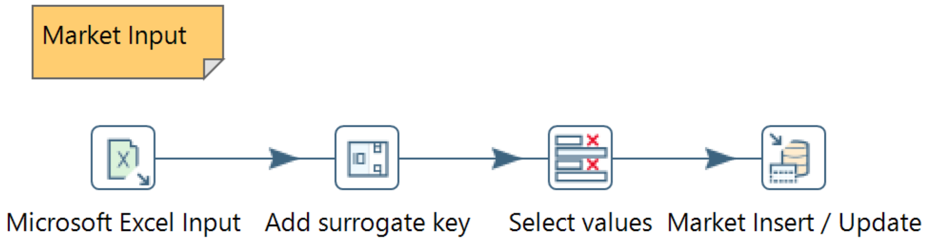
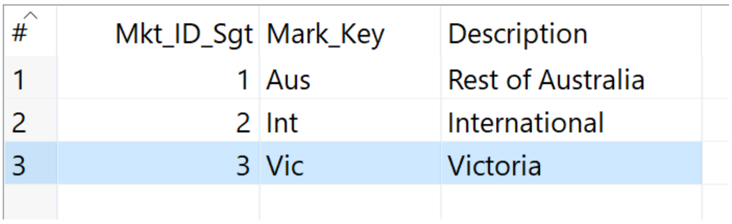


Figure 4 Market transformation

Transforming the market dimension only involves four steps since the given dataset is small. However, the business is facing remarkable growth which requires insert/updates frequently. Thus, integer surrogate-keys were added to the original dataset. The surrogate key and market key both used for look-up procedure in the insert/update procedure. The above picture shows the transformation process. The below picture shows the results of the implementation of transformation.



#	Mkt_ID_Sgt	Mark_Key	Description
1	1	Aus	Rest of Australia
2	2	Int	International
3	3	Vic	Victoria

Figure 5 Result of market transformation

2.4 Product Dimension Transformation

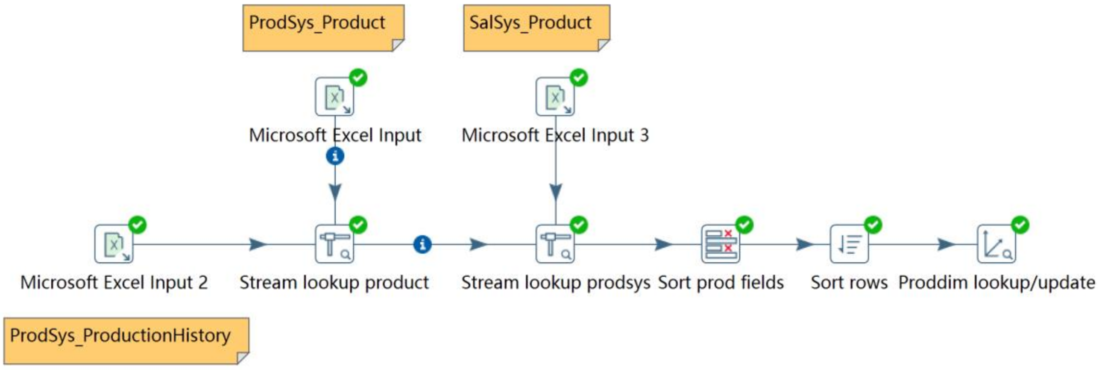
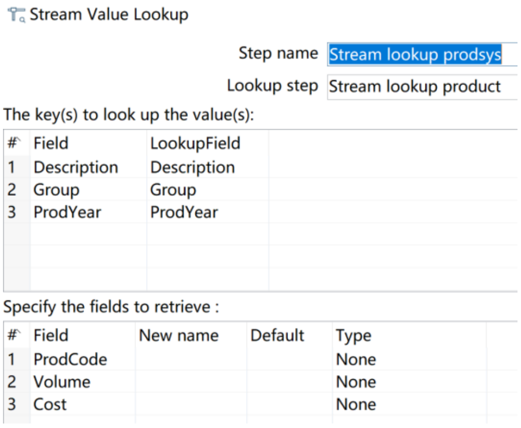


Figure 6 Product Transformation

The above figure describes the diagram of Product Table Transformation. Firstly the Production System is processed, the production information stored in table Product is assigned to corresponding types of wine produced in different years. According to the given tables, it is speculated that each type of wines produced in each year share one ProductionID. To align Production system with Sales System, ProductionID in table Production History is abandoned and other information will be joint into table Product in Sales System according to composite primary key, 'Description', 'Group' and 'ProdYear' (see the following figure).



Stream Value Lookup

Step name: Stream lookup prodsys

Lookup step: Stream lookup product

The key(s) to look up the value(s):

#	Field	LookupField
1	Description	Description
2	Group	Group
3	ProdYear	ProdYear

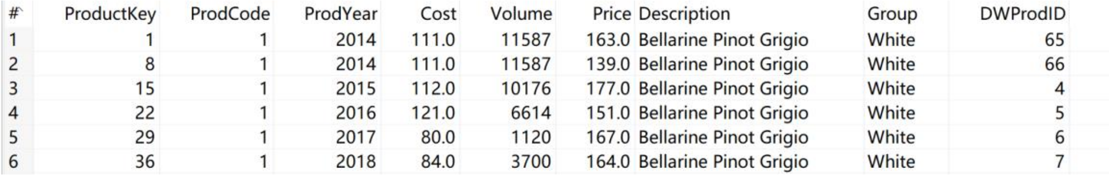
Specify the fields to retrieve :

#	Field	New name	Default	Type
1	ProdCode			None
2	Volume			None
3	Cost			None

Figure 7 stream lookup in the production dimension

Then after sorting, the slowly changing type is defined and the table is outputted to the database.

In the current table Product in Sales System, each product has one or two prices and they are identified by ProductKey. It may be caused by the change of vintage of red wine or discounting. For the production dimension, the price will change in the future. Thus in the Dimension lookup/update step, the SCD type is type 2 and the version field is added to record the changes. A new row will be added after the price adjustment. Part of output outcomes is showing in the next figure, in which DWProdID is the technical key, a surrogate key.



#	ProductKey	ProdCode	ProdYear	Cost	Volume	Price	Description	Group	DWProdID
1	1	1	2014	111.0	11587	163.0	Bellarine Pinot Grigio	White	65
2	8	1	2014	111.0	11587	139.0	Bellarine Pinot Grigio	White	66
3	15	1	2015	112.0	10176	177.0	Bellarine Pinot Grigio	White	4
4	22	1	2016	121.0	6614	151.0	Bellarine Pinot Grigio	White	5
5	29	1	2017	80.0	1120	167.0	Bellarine Pinot Grigio	White	6
6	36	1	2018	84.0	3700	164.0	Bellarine Pinot Grigio	White	7

Figure 8 Part of table Proddim

2.5 Sales Agent Dimension Transformation

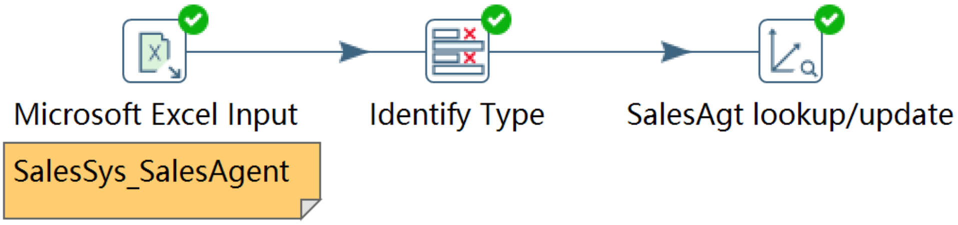
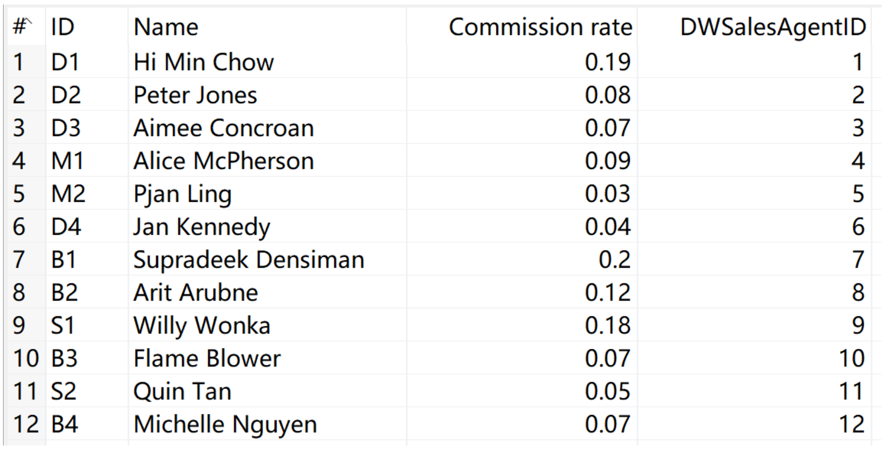


Figure 9 Sales Agent Transformation

In the sales agent dimension, the aim is to process sales agent information and all data come from table Sales Agent from Sales System. It is a simple transformation to retrieve attribution of sales agents, which including the step of identity data type.

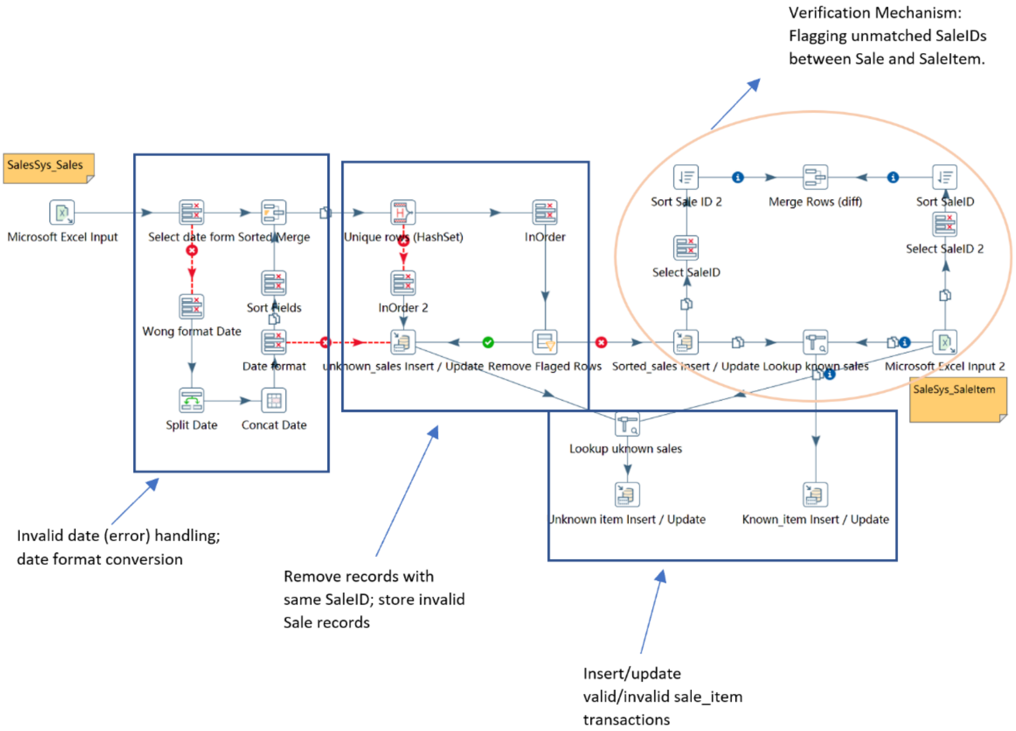
At last, considering the possible changes in the future, dimension lookup/update step is designed to record potential updating of commission rate even though there are no any changing record now. The SCD type is type 2. The output in table 'Sales_Agent_Dim' is following and DWSalesAgentID is the surrogate key.



#	ID	Name	Commission rate	DWSalesAgentID
1	D1	Hi Min Chow	0.19	1
2	D2	Peter Jones	0.08	2
3	D3	Aimee Concroan	0.07	3
4	M1	Alice McPherson	0.09	4
5	M2	Pjan Ling	0.03	5
6	D4	Jan Kennedy	0.04	6
7	B1	Supradeek Densiman	0.2	7
8	B2	Arit Arubne	0.12	8
9	S1	Willy Wonka	0.18	9
10	B3	Flame Blower	0.07	10
11	S2	Quin Tan	0.05	11
12	B4	Michelle Nguyen	0.07	12

Figure 10 Sales_Agent_Dim

2.6 Fact table: Sales data Transformation



Verification Mechanism: Flagging unmatched SaleIDs between Sale and SaleItem.

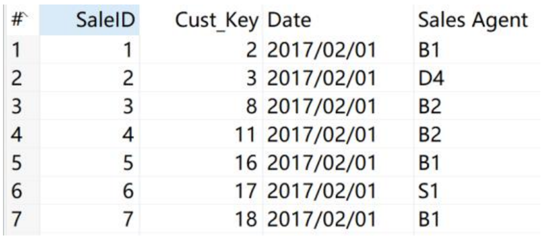
Invalid date (error) handling; date format conversion

Remove records with same SaleID; store invalid Sale records

Insert/update valid/invalid sale_item transactions

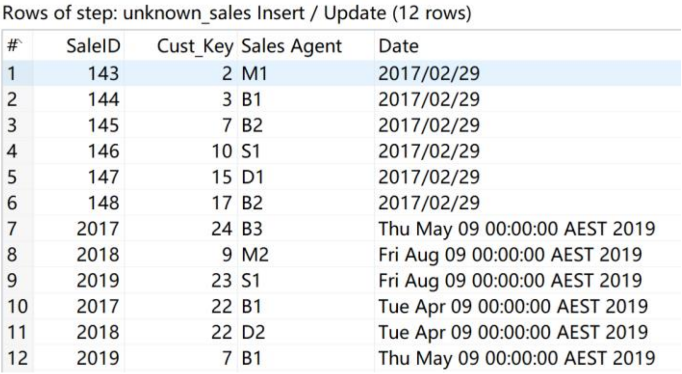
Figure 11 Sales data Transformation

In the sales fact table, the sales data need to be processed first. This transformation involves table Sales and SaleItem. The above picture dictates the process of transformation. It starts with error handling policy for the date data in table Sales as the input format is not standard. In the table Sales processing model, trough step ‘Select date form’, the date with different date format are separated. Rows with ‘DD/MM/YYYY’ are output to step ‘Wrong format Date’. By reforming the ‘Day’, ‘Month’ and ‘Year’ of date with the wrong format, all data are converted to uniform format ‘YYYY/MM/DD’ and outputted to table ‘Sorted_sales insert/update’. Meanwhile, there are six rows of record with a date that are invalid (2017/02/29) and they are treated as incorrect information and stored in table ‘unknown_sales lookup table’. We also observed records with same SaleID and stored them into “unknown_sales” table. The inconsistent records were also identified and removed as invalid records that exist in SaleItem but not Sale table. Result of the two output tables are following.



#	SaleID	Cust_Key	Date	Sales Agent
1	1	2	2017/02/01	B1
2	2	3	2017/02/01	D4
3	3	8	2017/02/01	B2
4	4	11	2017/02/01	B2
5	5	16	2017/02/01	B1
6	6	17	2017/02/01	S1
7	7	18	2017/02/01	B1

Figure 12 Part of Sorted_sales lookup table

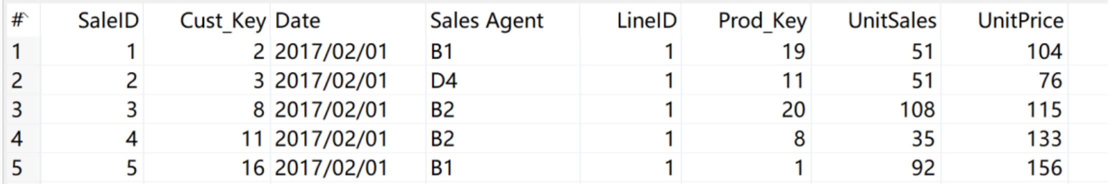


Rows of step: unknown_sales Insert / Update (12 rows)

#	SaleID	Cust_Key	Sales Agent	Date
1	143	2	M1	2017/02/29
2	144	3	B1	2017/02/29
3	145	7	B2	2017/02/29
4	146	10	S1	2017/02/29
5	147	15	D1	2017/02/29
6	148	17	B2	2017/02/29
7	2017	24	B3	Thu May 09 00:00:00 AEST 2019
8	2018	9	M2	Fri Aug 09 00:00:00 AEST 2019
9	2019	23	S1	Fri Aug 09 00:00:00 AEST 2019
10	2017	22	B1	Tue Apr 09 00:00:00 AEST 2019
11	2018	22	D2	Tue Apr 09 00:00:00 AEST 2019
12	2019	7	B1	Thu May 09 00:00:00 AEST 2019

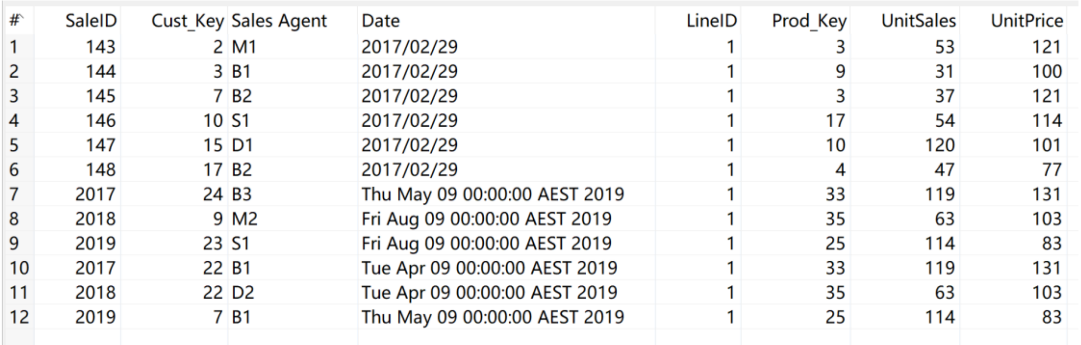
Figure 13 unknown_sales insert table

Then, table Sales and SaleItem are merged. Tables with wrong and right sales data retrieve data from table SaleItem and output table ‘Known_item lookup table’ and ‘unknown_item lookup table’ respectively. The former one will be used in the following analysis while the other will store incorrect data as a metadata table. The output is following.



#	SaleID	Cust_Key	Date	Sales Agent	LineID	Prod_Key	UnitSales	UnitPrice
1	1	2	2017/02/01	B1	1	19	51	104
2	2	3	2017/02/01	D4	1	11	51	76
3	3	8	2017/02/01	B2	1	20	108	115
4	4	11	2017/02/01	B2	1	8	35	133
5	5	16	2017/02/01	B1	1	1	92	156

Figure 14 Part of Known_item lookup table



#	SaleID	Cust_Key	Sales Agent	Date	LineID	Prod_Key	UnitSales	UnitPrice
1	143	2	M1	2017/02/29	1	3	53	121
2	144	3	B1	2017/02/29	1	9	31	100
3	145	7	B2	2017/02/29	1	3	37	121
4	146	10	S1	2017/02/29	1	17	54	114
5	147	15	D1	2017/02/29	1	10	120	101
6	148	17	B2	2017/02/29	1	4	47	77
7	2017	24	B3	Thu May 09 00:00:00 AEST 2019	1	33	119	131
8	2018	9	M2	Fri Aug 09 00:00:00 AEST 2019	1	35	63	103
9	2019	23	S1	Fri Aug 09 00:00:00 AEST 2019	1	25	114	83
10	2017	22	B1	Tue Apr 09 00:00:00 AEST 2019	1	33	119	131
11	2018	22	D2	Tue Apr 09 00:00:00 AEST 2019	1	35	63	103
12	2019	7	B1	Thu May 09 00:00:00 AEST 2019	1	25	114	83

Figure 15 unknown_item lookup table

2.7 Fact Table Transformation

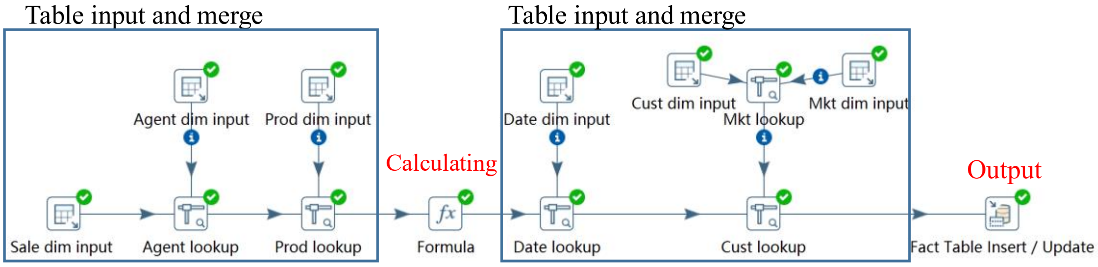
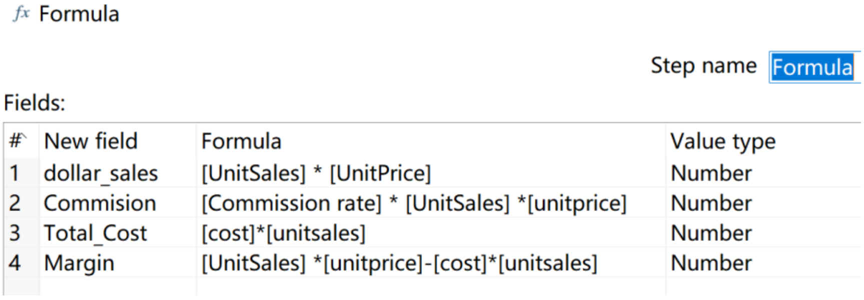


Figure 16 Transactional_sales_order_fact_transformation

The purpose of the fact table is to calculate margin, commission and other information and to output corresponding records in different tables with the grain of transaction requested in the business data analysis process.

As the given transformation diagram is shown, there are four steps in fact transactions. Firstly, all information required in calculating is inputted and merged. Secondly, through step 'formula', dollar_sales, Commission, Total_Cost and Margin of each sales order line are calculated (see the following figure).



fx Formula

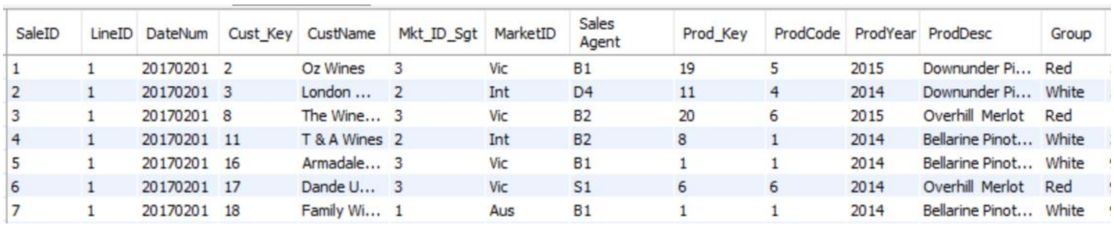
Step name Formula

Fields:

#	New field	Formula	Value type
1	dollar_sales	[UnitSales] * [UnitPrice]	Number
2	Commision	[Commission rate] * [UnitSales] *[unitprice]	Number
3	Total_Cost	[cost]*[unitsales]	Number
4	Margin	[UnitSales] *[unitprice]-[cost]*[unitsales]	Number

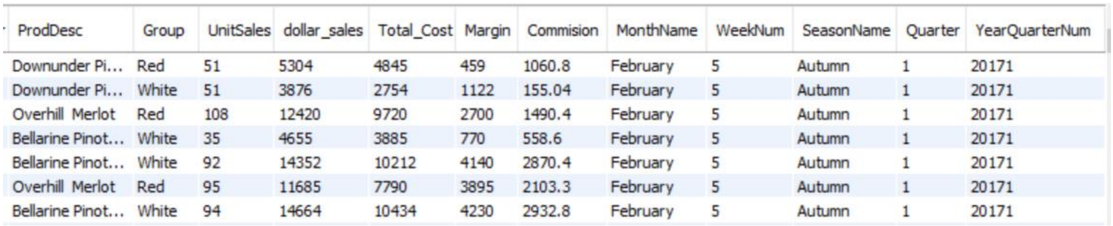
Figure 17 Fact transaction: Formula

Then, other information needed in a future analysis like DateNum is merged from three other dimensions. Those data will contribute to the process of querying and grouping in the future. At last, the result is outputted to the table 'transactional_sales_order_fact_table'. There are 4405 rows are outputted and part of the result in the database is the following.



SaleID	LineID	DateNum	Cust_Key	CustName	Mkt_ID_Sgt	MarketID	Sales Agent	Prod_Key	ProdCode	ProdYear	ProdDesc	Group
1	1	20170201	2	Oz Wines	3	Vic	B1	19	5	2015	Downunder Pi...	Red
2	1	20170201	3	London ...	2	Int	D4	11	4	2014	Downunder Pi...	White
3	1	20170201	8	The Wine...	3	Vic	B2	20	6	2015	Overhill Merlot	Red
4	1	20170201	11	T & A Wines	2	Int	B2	8	1	2014	Bellarine Pinot...	White
5	1	20170201	16	Armadale...	3	Vic	B1	1	1	2014	Bellarine Pinot...	White
6	1	20170201	17	Dande U...	3	Vic	S1	6	6	2014	Overhill Merlot	Red
7	1	20170201	18	Family Wi...	1	Aus	B1	1	1	2014	Bellarine Pinot...	White

Figure 18 Part of transactional_sales_order_fact_table-1



ProdDesc	Group	UnitSales	dollar_sales	Total_Cost	Margin	Commision	MonthName	WeekNum	SeasonName	Quarter	YearQuarterNum
Downunder Pi...	Red	51	5304	4845	459	1060.8	February	5	Autumn	1	20171
Downunder Pi...	White	51	3876	2754	1122	155.04	February	5	Autumn	1	20171
Overhill Merlot	Red	108	12420	9720	2700	1490.4	February	5	Autumn	1	20171
Bellarine Pinot...	White	35	4655	3885	770	558.6	February	5	Autumn	1	20171
Bellarine Pinot...	White	92	14352	10212	4140	2870.4	February	5	Autumn	1	20171
Overhill Merlot	Red	95	11685	7790	3895	2103.3	February	5	Autumn	1	20171
Bellarine Pinot...	White	94	14664	10434	4230	2932.8	February	5	Autumn	1	20171

Figure 19 Part of transactional_sales_order_fact_table-2

3. Design of the Data Warehouse

The star schema is redesigned to solve problems more easily and effectively. Following is the new design.

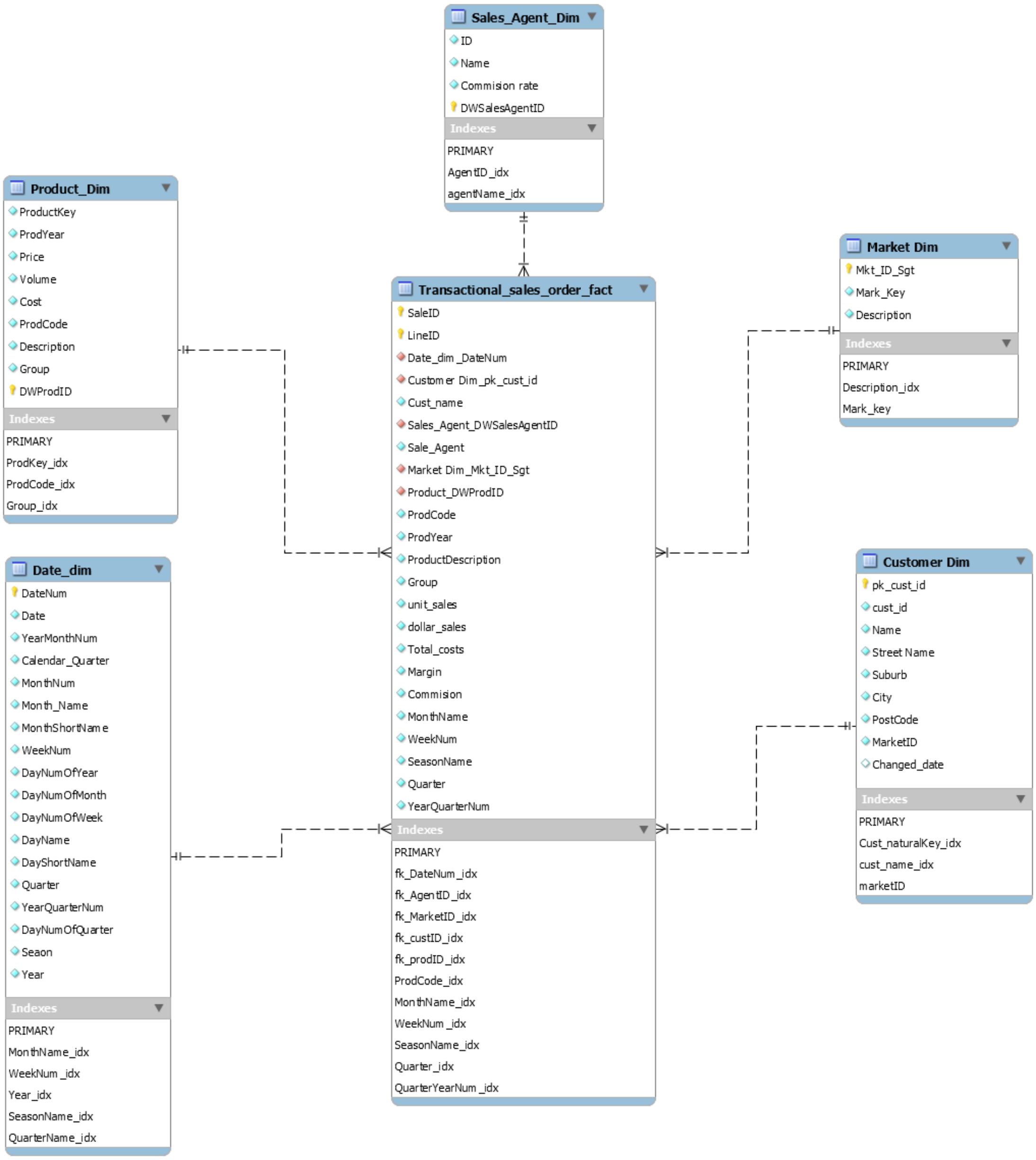


Figure 20 Dimension Model

The number of dimensions is reduced from seven to five. Dimension ‘product_type’ and ‘Sales_Order_Line’ are removed and their fields are incorporated into product dimension and fact table respectively. Also, based on the detailed data of Overhill Winery, the column names are adjusted. For the Date dimension, the data types are more varied owing to the complement of excel tables. Columns like ‘YearQuarterNum’ are added to simplify the query process and provide more options.

In addition, the primary key of dimensions is changed from nature key to the surrogate key.

These surrogate keys also indicate the SCD types illustrated in the process of ETL design. For example, the primary key of the Sales Agent dimension is pk_cust_ID. ‘Mkt_ID_Sgt’ is also designed to identify markets with a number. Considering the uniqueness and invariance of date, the Date dimension uses the nature key ‘DateNum’ as the primary key.

In the fact table, the foreign keys are changed with the primary keys in dimensions. More query conditions are added to help users to query and group the required information easily. Through operating towards the final output, users could realize querying within one database table. It is more user-friendly and will make the operation easier and the result more intuitionistic, to better assist users analyzing.

Indexes created for dimension and fact table aim to achieve a satisfactory performance of querying. Despite having more indexes takes more storages, it is more competitive to do so since storage is more cost-effective than computing. In this data warehouse, we created indexes based on the following principles:

- Created indexes for primary keys in dimension tables with slowly changing dimensions.
- Created indexes for those keys/attributes which will be used in the “where” clause frequently.
- Created indexes for attributes in dimension tables which are string-type data.
- Did not create an index for those values that will be updated frequently.

Over time, the data warehouse changes to accommodate changes in Overhill’s business structure, and the index structure must be changed. Since the data warehouses are directly connected to relational tables, we can use index tuning methods to modify indexes, such as evaluating queries and data mixing to adjust the index accordingly in the future.

4. Data Dictionary

It is recommended to check if there any NULL values or duplicates for all dimension tables as one procedure in pre-processing.

Market Dimension Table

Job Name	Mkt -transform
Job Purpose	Read market dataset to dimension table, add surrogate keys, export target schema
Source Table/Files	File Name: Market.xlsx Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData\SalesSystem
Target Table/Files	Table Name: Market lookup table Target Schema: dw
Frequency	Monthly
Data Quality Level	High

Date Dimension Table

Job Name	Date Dim-Transform
Job Purpose	Read date dataset to dimension table, add season to the table, export to target schema
Source Table/Files	File Name: DimDates.xlsx Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData
Target Table/Files	Table Name: Date Lookup table Target Schema: dw
Frequency	Monthly
Data Quality Level	High

Customer Dimension Table

Job Name	Cust -Transform
Job Purpose	Read date datasets to dimension table,merge them, read sheets' name as date, convert date string to date type, remove duplicates, export to dimension table to target schema
Source Table/Files	File Name: Customer Dec 2019.xlsx; Customer Feb 2019.xlsx; Customer Jan 2018.xlsx Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData\SalesSystem
Target Table/Files	Table Name: Customer Lookup table Target Schema: dw
Frequency	Quarterly
Data Quality Level	Medium

Product Transformation

Job Name	Product_table_transformation
Job Purpose	Merge 3 tables from 2 systems and define the SCD type of price as type 2
Source Tables/Files	File Name: Product.xlsx; ProductionHistory.xlsx Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData\ProductionSystem File Name: Product.xlsx Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData\SalesSystem
Target Tables/Files	Table Name: Proddim Target Schema: dw
Frequency	Monthly
Data Quality Level	High

Sales Agent Transformation

Job Name	SalesAgent_transformation
Job Purpose	Input data and define type 2 as SCD for future change of commission rate
Source Tables/Files	File Name: Sales Agent.xlsx; Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData\SalesSystem
Target Tables/Files	Table Name: Sales_Agent_Dim Target Schema: dw
Frequency	Monthly
Data Quality Level	High

Fact table: Sales data Transformation

Job Name	Fact_table_sales_data_transformation
Job Purpose	Processing and cleaning data and merge tables
Source Tables/Files	File Name: Sales.xlsx; SaleItem.xlsx Location: D:\UniMelb\ISYS90086 DW\A2\OLTPData\SalesSystem
Target Tables/Files	Table Name: Sorted_sales lookup table; Known_item lookup table Target Schema: dw
Rejected Data	Table Name: unknown_sales lookup table; unknown_item lookup table Target Schema: dw
Frequency	Weekly
Data Quality Level	Medium

Fact Table Transformation

Job Name	transactional_sales_order_fact_transformation
Job Purpose	Calculating required business analysis information
Source Tables/Files	Database Name: Sales_Agent_Dim; Known_item lookup table; Proddim Schema: dw
Target	Table Name: transactional_sales_order_fact_table

Tables/Files	Target Schema: dw
Frequency	Weekly
Data Quality Level	High