AUT UNIVERSITY COMPUTER + MATHEMATICAL SCIENCES

Master of Computer and Information Sciences

Paper: COMP810 – Data Warehousing and Big Data

Semester-2, 2018

Assessment 2 –Data Warehousing Project

Weight in grade: 60%

NEW WORLD

Building and Analysing a DW prototype for New World, NZ

1. Assessment task

To design, implement and analyse a Data Warehouse (DW) prototype for New World, one of the biggest supermarket chains in NZ.

2. Project overview

New World is one of the biggest supermarket chains in NZ. The stores are located all over the country. New World has thousands of customers and therefore it is important for the organisation to analyse the shopping behaviour of their customers. Based on such analysis the organisation can optimise their selling techniques e.g. by having relevant promotions on different products.

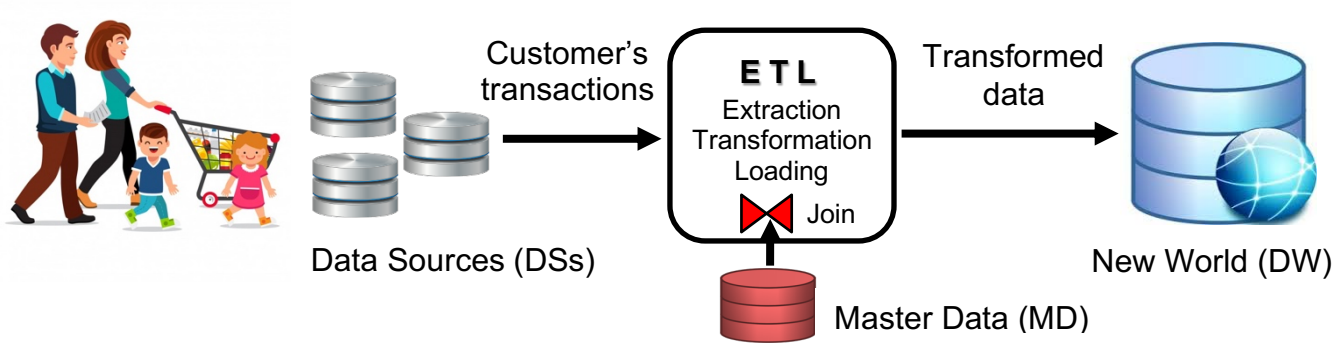

Figure 1: An overview of data integration in New World DW scenario

The company desires to analyse sopping behaviour of their customers that requires a DW in place. The customers' transactions from Data Sources (DSs) are required to be reflected in the DW on a daily basis. This process of reflecting the customers' data into DW is called Data Integration (DI) as shown in Figure 1. To implement DI we usually need ETL (Extraction, Transformation, and Loading) tools. Since the data generated by customers is not in the format required by DW, it needs to be processed in the transformation layer of ETL. This processing usually involves the enriching of transactional data with information from Master Data (MD) as shown in Figure 2.

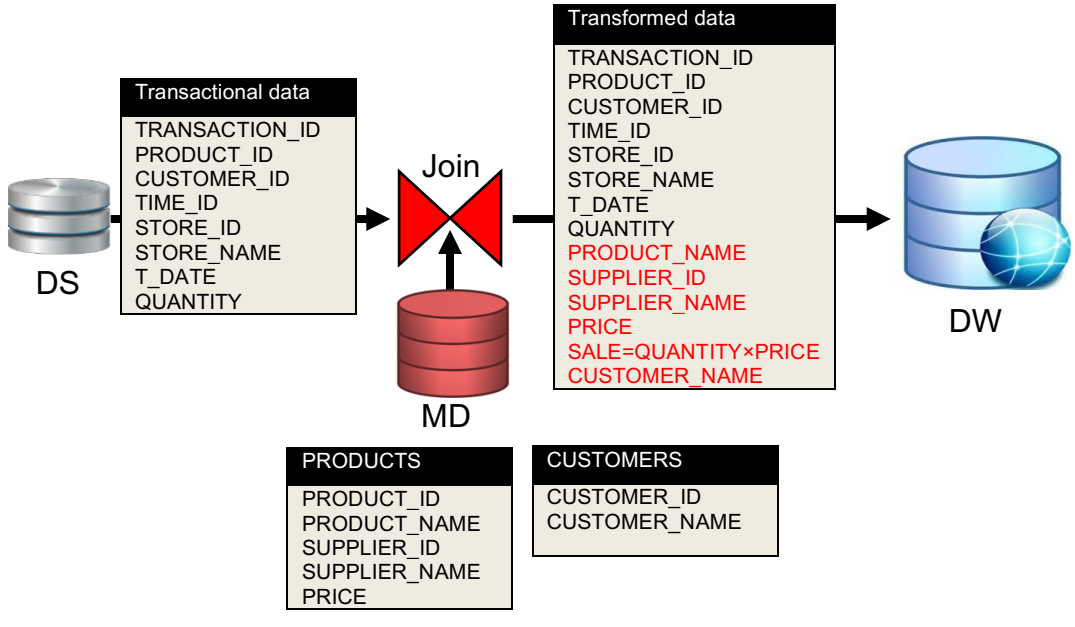

Figure 2: An example of enriching transactional data with information from MD

To implement this enrichment feature in the transformation phase of ETL we need a join operator typically called Semi-Stream Join (SSJ). There are a number of algorithms available to implement this join operation however, the simplest one is Index Nested Loop Join (INLJ) which is explained in the next section and you are required to implement it in this project.

3. Index Nested Loop Join (INLJ)

The INLJ is a traditional join operator to implement the join operation between DS and MD. In INLJ, DS is scanned in batches of tuples and based on each tuple in the batch, the disk-based MD is accessed using an index on the join attribute. A graphical overview of an INLJ is shown in Figure 3 where tuple s_i from DS's batch is joined with tuple r_j from MD.

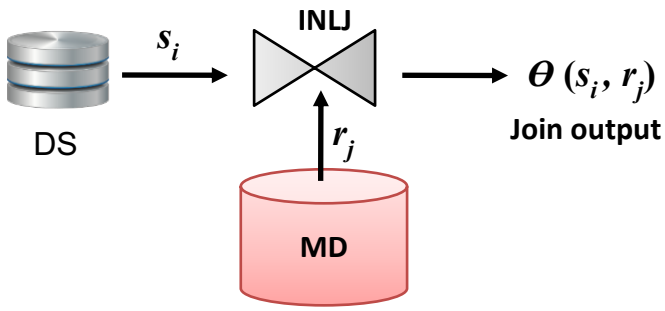

Figure 3: Execution architecture of INLJ

The crux of INLJ is that with every loop step, the algorithm reads a batch of DS tuples and joins them one-by-one with the relevant tuples from MD. To access the relevant tuple from MD the algorithm uses index on the join attribute in MD. For example, for the scenario presented in Figure 2 the join attributes are PRODUCT_ID and CUSTOMER_ID in both transactional and MD.

4. Star-schema

The star schema (which you will use in this project) is a data modelling technique that is used to map multidimensional decision support data into a relational database. Star-schema yields an easily implemented model for multidimensional data analysis while still preserving the relational structures on which the operational database is built.

The star schema represents aggregated data for specific business activities. Using this schema, one can store aggregated data from multiple sources that will represent different aspects of business operations. For example, the aggregation may involve total sales by selected time periods, by products, by stores, and so on. Aggregated totals can be *total number of product sold*, *total sales values by products*, and so on. The basic star schema has three main components: *facts, dimensions,* and *attributes.* Usually in case of star-schema for sales the dimension tables are: customer, *product*, *time*, *store,* and *supplier* while the fact table is *sales_table*. However, for your ease the structure of the star schema you will used in this project is described in the next section.

5. Data and structure specifications

The assessment provides a scripts file named "Transactional_masterData_Generator.sql". By executing the script it will create following three tables in your account. One is TRANSACTIONS

table in DS with 10,000 records populated in it. This data will be generated randomly based on 100 products, 50 customers, 10 stores, and one year time period as a date - from 01-Jan-17 to 31-Dec-17. The values for the quantity attribute will be random between 1 and 10. The other two tables named PRODUCTS and CUSTOMERS in MD with 100 and 50 records respectively. The structure for transaction data table, MD tables, and DW tables with their attribute names and data types is given in Figure 4.

TRANSACTIONS								
Attributes	TRANSACTION_ID (PK)	PRODUCT_ID	CUSTOMER_ID	TIME_ID	STORE_ID	STORE_NAME	T_DATE	QUANTITY
Data type and size	NUMBER(8,0)	VARCHAR2(6)	VARCHAR2(4)	VARCHAR2(8)	VARCHAR2(4)	VARCHAR2(20)	DATE	NUMBER(3,0)

(a) Transactional Data table

PRODUCTS					
Attributes	PRODUCT_ID (PK)	PRODUCT_NAME	SUPPLIER_ID	SUPPLIER_NAME	PRICE
Data type and size	VARCHAR2(6)	VARCHAR2(30)	VARCHAR2(5)	VARCHAR2(30)	NUMBER(5,2) DEFAULT 0.0

CUSTOMERS		
Attributes	CUSTOMER_ID (PK)	CUSTOMER_NAME
Data type and size	VARCHAR2(4)	VARCHAR2(30)

(b) MD tables

D_CUSTOMERS		
Attributes	CUSTOMER_ID (PK)	CUSTOMER_NAME
Data type and size	VARCHAR2(4)	VARCHAR2(30)

D_PRODUCTS		
Attributes	PRODUCT_ID (PK)	PRODUCT_NAME
Data type and size	VARCHAR2(6)	VARCHAR2(30)

D_STORES		
Attributes	STORE_ID (PK)	STORE_NAME
Data type and size	VARCHAR2(4)	VARCHAR2(30)

D_SUPPLIERS		
Attributes	SUPPLIER_ID (PK)	SUPPLIER_NAME
Data type and size	VARCHAR2(5)	VARCHAR2(30)

D_TIME						
Attributes	TIME_ID (PK)	CAL_DATE	CAL_DAY	CAL_MONTH	CAL_QUARTER	CAL_YEAR
Data type and size	VARCHAR2(8)	DATE	VARCHAR2(9)	VARCHAR2(9)	VARCHAR2(1)	NUMBER(4,0)

W_FACTS									
Attributes	TRANSACTION_ID (PK)	CUSTOMER_ID (FK)	PRODUCT_ID (FK)	STORE_ID (FK)	SUPPLIER_ID (FK)	TIME_ID (FK)	QUANTITY	PRICE	SALE
Data type and size	NUMBER(8,0)	VARCHAR2(5)	VARCHAR2(8)	VARCHAR2(4)	VARCHAR2(5)	VARCHAR2(8)	NUMBER(2,0)	NUMBER(5,2)	NUMBER(6,2)

(c) DW tables

Figure 4: Structure for transactional data, MD, and DW tables

.

6. Implementation of INLJ

To implement INLJ algorithm you will implement the following steps.

1. Read a batch of 100 tuples from TRANSACTIONS table as input data into a cursor. The cursor is a user defined data type in PLSQL which works as a list and is used to store multiple records in memory for processing.
2. Read the cursor tuple by tuple and for each tuple retrieve the relevant tuples from PRODUCTS and CUSTOMERS tables of MD using PRODUCT_ID and CUSTOMER_ID as indexes on the both tables respectively and add the required attributes (mentioned in colour red in Figure 2) into the transaction tuple (in memory).
3. The transaction tuple with new attributes is to be loaded into DW. Before loading the tuple into DW you will check whether the dimension tables already contain this information. If yes, then only update the fact table otherwise update the required dimension tables and the fact table.
4. Repeat steps 1 to 3 until you load all the data from TRANSACTIONS table to DW.

7. DW analysis

Once the entire data has been loaded into DW, apply the following analysis to your DW using OLAP queries.

1. Which product generated maximum sales in September, 2017?

PRODUCT NAME	SALE

   OR

PRODUCT NAME	SALE	RANK

2. Determine top three supplier names based on highest sales of their products.

RANK	SUPPLIER NAME	SALE

3. Determine the top 3 store names who generated highest sales in September, 2017.

RANK	STORE NAME	SALE

4. Presents the quarterly sales analysis for all stores using drill down query concepts.

STORE_NAME	Q1_2017	Q2_2017	Q3_2017	Q4_2017

5. Create a materialised view with name “STORE_PRODUCT_ANALYSIS” that presents store and product wise sales. The results should be ordered by store name and then product name.

STORE NAME	PRODUCT NAME	SALE

6. Create a materialised view with name “MONTH_STORE_ANALYSIS” that presents month and store wise sales. The results should be ordered by month name and then store name.

MONTH	STORE NAME	SALE

8. Tasks break-up

Following is a list of tasks that you need to complete in this project.

1. Creating star schema using the tables' structures described in Figure 4(c). Make sure your SQL script should drop any pre-exiting table(s) with the same name(s). The script should also apply all primary and foreign keys on the right attributes.
2. Implementing the INLJ algorithm (as described in Section 6) to load transactional data into DW after joining it with MD.
3. Applying the analytic queries (described in Section 7) on DW using slicing, dicing, drill down, and materialising view concepts.
4. Writing the project report that should include project overview, Pseudocode for INLJ algorithm, your OLAP queries with outputs and a summary of 2 pages stating what you have learnt from the project.

9. What to submit

Each student has to submit the following files:

1. *createDW* –SQL script file to create star-schema for DW
   Note: your scripts should drop the table(s) if they already exist in the database.
2. *INLJ* – PLSQL file that implements the INLJ algorithm
3. *queriesDW* – SQL script file containing all of your OLAP queries
4. *projectReport* – a doc file containing all contents described in point 4 under the task break-up section
5. *readMe* – a text file describing the step-by-step instructions to operate your project

Note: all files need to be submitted in a zipped folder named by your familyName_studentID e.g. *Smith_1612345*.

10. When to submit

Due date: Tuesday, 23rd Oct 2017, 4:00 p.m.

Late penalty: maximum late submissions time is 24 hours after the due date. In this case 5% late penalty will be applied.

11. Where to submit

The project should be submitted through AUT Blackboard.

NOTE: Every student has to complete the project individually. Each student's project source and report materials should be unique and done on his/her own. All assessments will be assessed through TurnItIn system and in case of finding of any duplication or identical material, AUT plagiarism policy will be applied.

---------- E N D ----------

Appendix

Marking guide

Project Component	Marks
CreateDW –SQL script file to create star-schema for DW	/15
The script should create all dimension and fact tables table in DW and if any table with same name already exists, the table should be dropped. The script should also apply all primary and foreign keys on the right attributes.	
Implementing of INLJ	/30
INLJ procedure should implement all three phases of ETL – it should extract records from TRANSACTIONS table, transform these with MD and then load these records to DW successfully.	
queriesDW – SQL script file containing of all your OLAP queries	/36
The file should include OLAP queries for all tasks presented in Section 7.	
projectReport – a doc file containing all contents described in point 4 under the task break-up section.	/15
Report must contain project overview, Pseudocode for INLJ algorithm, your OLAP queries with outputs and a summary of 2 pages stating what was learnt from the project.	
readMe – a text file describing the step-by-step instructions to operate your project	/4
readMe file should contain a step-by-step guide to operate the project.	
Late submission penalty	-/5
TOTAL MARKS	/100