Bachelor of Science in Computer Sciences

Course Code: CS408

Course Name: Data Warehousing

Semester-Fall, 2019

Practical Project

Building and Analysing a DW Prototype for Metro Store in Islamabad

Marks: 100

Weight in grade: 30%

1. Assessment task

To design, implement and analyse a Data Warehouse (DW) prototype for Metro store, one of the biggest retail store in Islamabad.

2. Project overview

Metro is one of the biggest retail store in Islamabad. The store has thousands of customers 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.

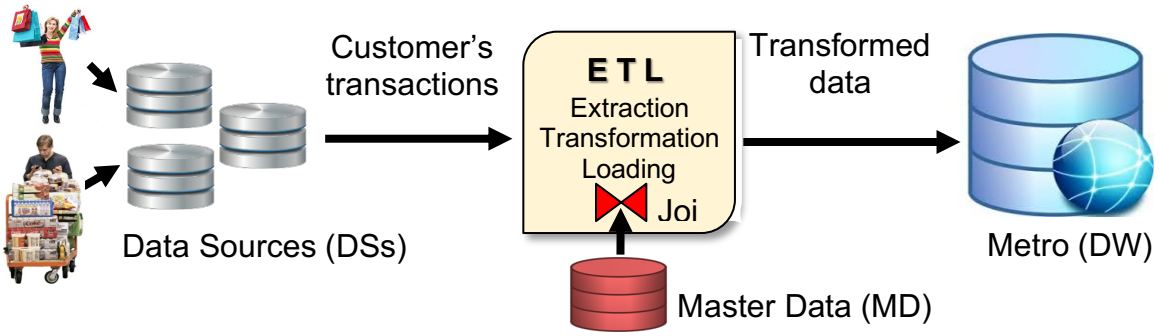

Figure 1: An overview of data integration in Metro store

There is a need to build a DW to make the analysis of shopping behaviour practical; customers' transactions from Data Sources (DSs) are required to be reflected in the DW on a daily basis. This process of reflecting the customer 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 will involve the enriching of transactional data with information from Master Data (MD) as shown in Figure 2.

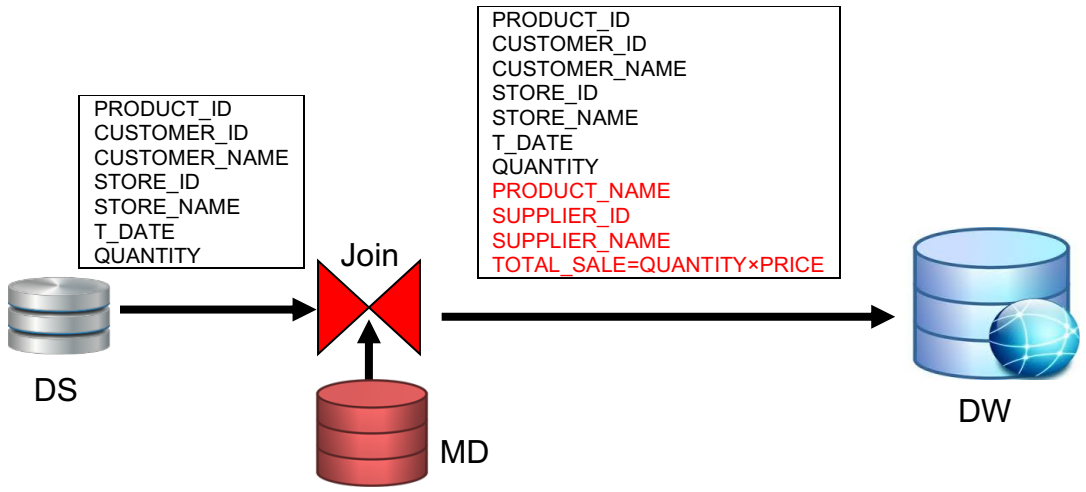

Figure 2: Enrichment example

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 tuple by tuple and based on this, 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 DS tuple s_i is joined with MD tuple r_j.

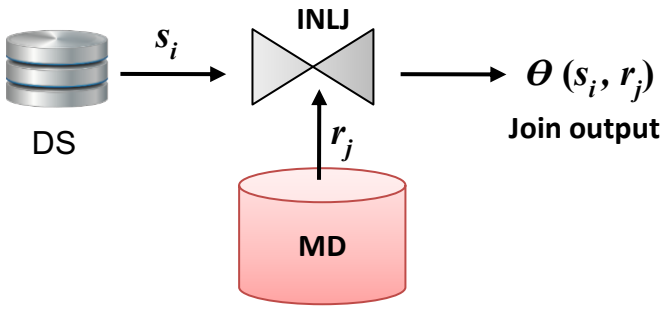

Figure 3: Execution architecture of INLJ

The crux of INLJ is that with every loop step, the algorithm reads a chunk of DS tuples and joins them one-by-one with the relevant tuples in 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 attribute is PRODUCT_ID which should be considered as the Primary Key (PK) in MD while a Foreign Key (FK) in DS and therefore, the join would be between PK and FK.

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 four main components: *facts, dimensions, attributes,* and *classification hierarchy.* Usually in case of star-schema for sales the dimension tables are: *product*, *date*, *store,* and *supplier* while the fact table is *sales*. However, to determine the right attributes you will need to consider Figure 2.

5. Data specifications

The assessment provides a scripts file named “Transaction_and_MasterData_Generator.sql”. By executing the script it will create two tables in your account. One is TRANSACTIONS table 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-16 to 31-Dec-16. The values for the quantity attribute will be random between 1 and 10. The other is MASTERDATA table with 100 records in it. The structure of both tables with their attribute names and data types is given in Figure 4. The attributes TRANSACTION_ID and PRODUCT_ID are primary keys in TRANSACTIONS and MASTERDATA tables respectively.

TRANSACTIONS								
Attributes	TRANSACTION_ID	PRODUCT_ID	CUSTOMER_ID	CUSTOMER_NAME	STORE_ID	STORE_NAME	T_DATE	QUANTITY
Data type and size	VARCHAR2(8)	VARCHAR2(6)	VARCHAR2(4)	VARCHAR2(30)	VARCHAR2(4)	VARCHAR2(20)	DATE	NUMBER(3,0)

MASTERDATA					
Attributes	PRODUCT_ID	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

Figure 4: Structures for TRANSACTIONS and MASTERDATA tables

6. Implementation of INLJ

To implement INLJ algorithm you will implement the following steps.

1. Read 50 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 tuple from MASTERDATA table using PRODUCT_ID as an index and add the required attributes (mentioned 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, you will be required to analyse your DW by applying following OLAP queries.

Q1 Which product produced highest sales in the whole year?

Q2 Determine the top 3 supplier names in Aug 2016 in terms of total sales.

Q3 Determine the top 3 store names in Aug 2016 in terms of total sales.

Q4 How many sales transactions were there for the product that generated maximum sales revenue in 2016? Also present the product quantity sold.

Q5 Present the quarterly sales analysis for all products using drill down query concepts, resulting in a report that looks like:

PRODUCT_NAME	Q1_2016	Q2_2016	Q3_2016	Q4_2016

Q6 Create a materialised view with name "STOREANALYSIS_MV" that presents the product-wise sales analysis for each store.

STORE_ID	PROD_ID	STORE_TOTAL

8. Tasks break-up

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

1. Identification of appropriate dimension tables, fact table, and their attributes for the sales scenario presented in Figure 2. Based on that creation of star-schema for DW with appropriate primary and foreign keys. Consult the structure of tables TRANSACTIONS and MASTERDATA provided in Figure 4 in order to keep the attribute name and their data types consistent in DW.
2. Implementation of the INLJ algorithm and loading of transactional data into DW after joining it with MD.
3. Applying of different analysis (described in Section 7) on DW using slicing, dicing, drill down, and materialising view concepts.
4. Writing of the project report that should include project overview, INLJ algorithm, schema for DW, your OLAP queries with outputs and a summary (1-2 pages) of what you have learned 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 submited in a zipped folder named by your family name_student ID e.g. *Ahmed_1612345*.

10. When to submit

Due date: Wed, 20th Nov 2019, 11:00 a.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 Google Class Room.

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, **0 marks will be assigned**.

---------- 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 droped. It 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	/30
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.	/20
Report must contain project overview, INLJ algorithm, schema for DW, your OLAP queries with outputs and a summary of what was learnt from the project.	
readMe – a text file describing the step-by-step instructions to operate your project	/5
readMe file should contain a step-by-step guide to operate the project.	
Late submission penalty	-/5
TOTAL MARKS	/100