Bachelor of Science in Data Science

Course Code: DS-3003

Course Name: Data Warehousing

Semester-Fall, 2023

Course Project

Building and Analysing Data Warehouse Prototype for Electronica Business Chain

Marks: 100

Weight in grade: 15%

ChatGPT or any other Large Language Models are strictly not allowed.

1. Assessment task

The student has to design, implement, and analyse a Data Warehouse (DW) prototype for an Electronica Business Chain in Pakistan.

2. Project overview

Electronica is one of the biggest Electronics Business chain in Pakistan and worldwide. The stores has thousands of customers and therefore it is important for the business to online analyse the shopping behaviour of their customers. Based on that the business can optimise their selling techniques e.g. giving promotions on different products.

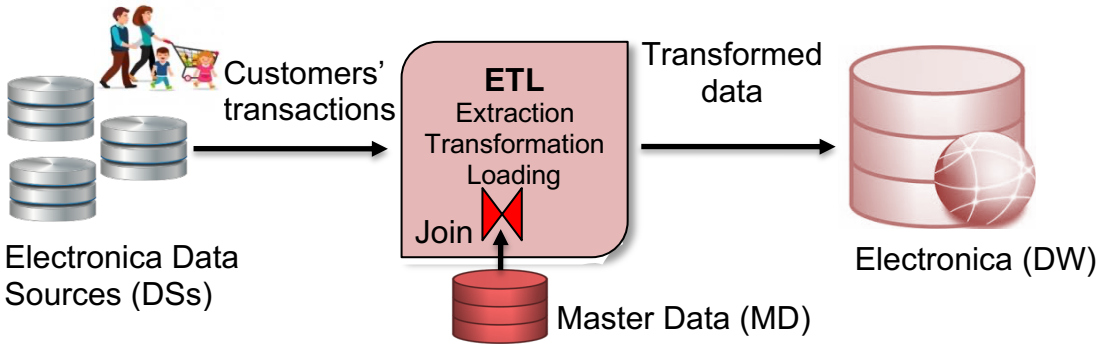

Figure 1: An overview of Electronica DW

Now, to make this analysis of shopping behaviour practical there is a need of building a near-real-time DW and customers' transactions from Data Sources (DSs) are required to reflect into DW as soon as they appear in DSs. The overview of Electronica DW is presented in Figure 1. To build a near-real-time DW we need to implement a near-real-time ETL (Extraction, Transformation, and Loading) tools. Since the data generated by customers is not in the format required by DW therefore, it needs to process in the transformation layer of ETL. For example enriching of some information e.g. attributes in colour red from disk-based Master Data (MD) as shown in Figure 2.

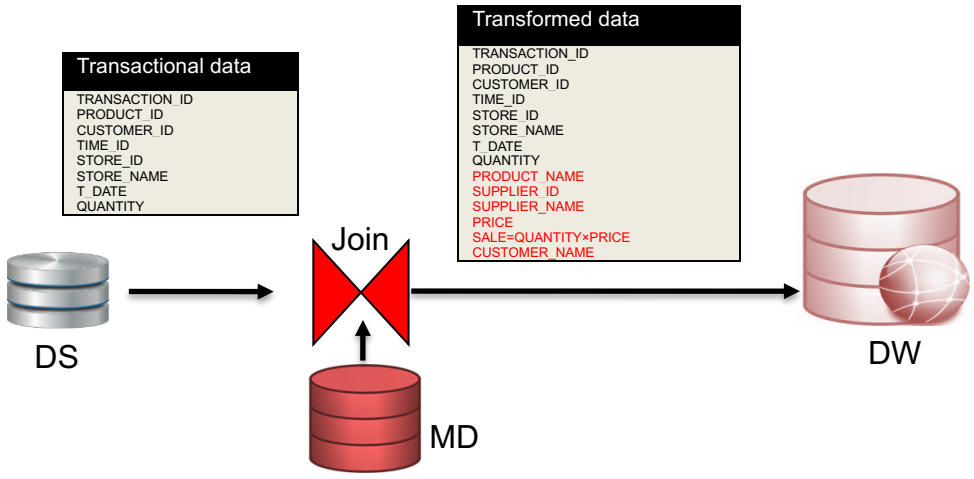

Figure 2: Enrichment example

To implement this enrichment feature in the transformation phase of ETL we need a join operator. There are a number of algorithms available to implement this join operation however, the most popular one is HYBRIDJOIN (Hybrid Join) which is explained in the next section and you will implement it in this project using **Java Eclipse**.

3. HYBRIDJOIN (Hybrid Join)

The HYBRIDJOIN algorithm has been introduced by Naeem et. al. in 2011 with the objective of implementing the join operation in the transformation phase of ETL.

The main components of HYBRIDJOIN are: **the disk-buffer** which is used to load data from MD in memory using the join attribute as an index. **The multi-hash table** (provided by Apache multi-hash-map) which stores the customers' transactions (tuples) and a pointer to the queue node**. The queue** (based on doubly linked list) is used to store the join attribute values which are also used as indexes on MD. Each queue node only stores one join attribute value for customers' transaction tuples. The reason for using a double-link list is to facilitate random deletion of nodes from the queue. **The stream-buffer** is used to hold the customer transaction meanwhile the algorithm completes one iteration. However, you don't need the stream buffer in this project as we are not considering the stream of customers' transactions.

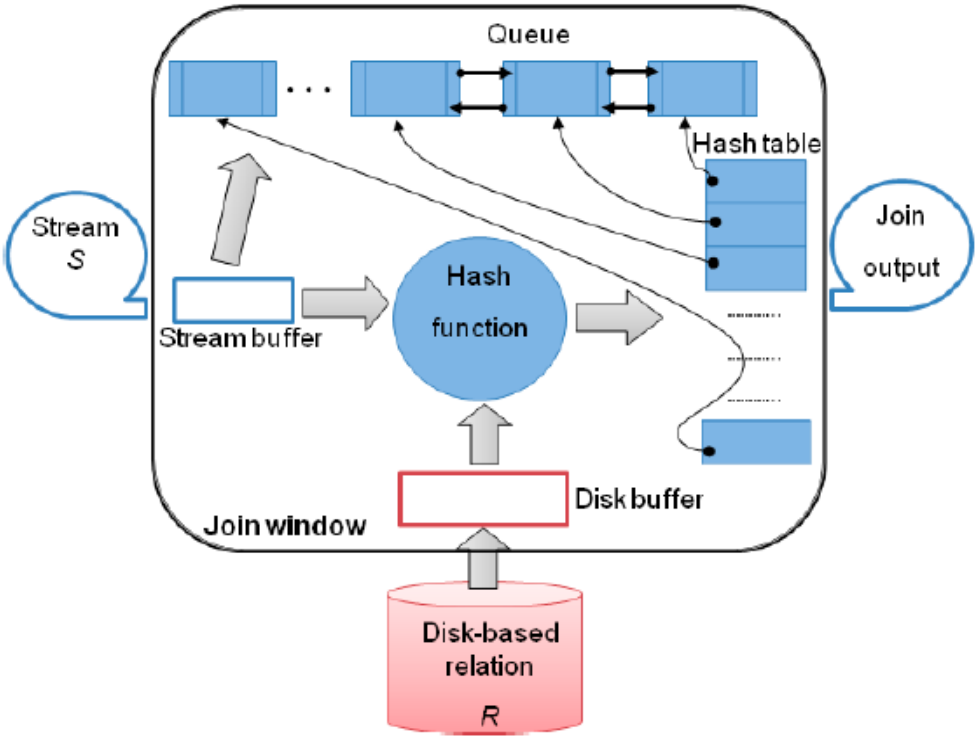

Figure 3: Execution Architecture of HYBRIDJOIN

The crux of HYBRIDJOIN is that with every loop step a new input chunk of customers' transactions is read into main memory (the Hash table and the Queue) from DS. The size of the input chunk depends on the number of tuples deleted from the Hash table in the previous transaction. However, initially for the first time you will load 1000 tuples from customers' transactions table to the Hash table along with their join attributes values to the Queue. Also for every loop step a segment of MD tuples is loaded into the disk-buffer using the join attribute value stored in the queue as an index. The size of the MD segment you will use in this project would be 10. The MD tuples loaded in the disk buffer are then probed into the multi-hash table. If the tuple matches in the multi-hash table then the matched record is joined with the relevant MD tuple and the join output is produced. The matched tuple is then removed from the multi-hash table and the queue. Finally, the joined output tuple is loaded to DW.

4. Multi-Threading

You are required to use a multithreaded approach to implement the project. Following thread should be implemented.

1. Thread1 (StreamGenerator): This thread is responsible for generating a stream of sales data stored in DSs.
2. Thread2 (HybdridJoin): This thread is dedicated to implementing the HYBRIDJOIN algorithm.
3. Thread 3 (Controller): This thread is responsible to monitor the stream arrival rate (**λ**) and service rate (***μ***). The thread will control the speed of StreamGenerator based on ***μ***. It will ensure that the HYBRIDJOIN algorithm should not be underload and there should not be an unnecessary backlog in the stream buffer.

This multithreaded design enhances efficiency by allowing concurrent execution of different aspects of the join operation, ultimately optimizing the performance of the entire system.

5. 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 the schema, one can create multiple aggregated data 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. The basic star schema has four main components: *facts, dimensions, attributes,* and *classification levels.* 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 need to carefully study the dataset. The name of your data warehouse should be ELECTRONICA-DW.

6. Data specifications

The assessment provides two csv files “transactions.csv” and “master_data.csv”. The transactional data is basically a customers’ sales data for the year **2019**. The transaction data will be joined with master data. You need to import the data from both CSV files to your MySQL databases. The attributes of the both datasets are given below.

Transactional data
Order ID, Order Date, ProductID, CustomerID, CustomerName, Gender ,Quantity Ordered
Master data
productID, productName, productPrice, supplierID, supplierName, storeID, storeName

7. Implementation of HYBRIDJOIN

To implement the HYBRIDJOIN algorithm you will implement following steps using **Java Eclipse**.

1. Read a new input chunk of sales data from the stream buffer and load it into the multi-hash table with their join attribute values in the queue. Each node in the queue should be pointed by the multi-hash table entry.
2. Select the oldest node from the queue and based on that a segment of MD will be loaded into the disk buffer.

3. Read each record from the disk buffer and match this to the multi-hash table. If a tuple match is found then add the required attributes of MD into the transaction tuple, produce the output tuple, and remove the matched tuple from the multi-hash table along with its join attribute value from the queue. It is important to note that the attribute TOTAL_SALE does not exist in MD while the attribute PRICE is there so you will calculate TOTAL_SALE using QUANTITY and PRICE attributes. You need to print the output of the first 50 tuples at the console and copy this to your report as well.
4. The transaction tuple with new attributes will then be loaded into DW. Before loading the tuple into DW you will check whether the dimensions tables already contain this information. If yes then only update the fact table otherwise update both dimensions and fact tables. Make sure to enter foreign keys in the fact table.
5. Repeat Step 3 to 4 until all loaded data is exhausted to probe the hash table.
6. Repeat steps 1 to 5 until you load all the data from the TRANSACTIONS table to DW.

8. 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 Present total sales of all products supplied by each supplier with respect to quarter and month using drill down concept.

Q2 Find total sales of product with respect to month using feature of rollup on month and feature of dicing on supplier with name "**DJI**" and Year as "2019". You will use the grouping sets feature to achieve rollup. Your output should be sequentially ordered according to product and month.

Q3 Find the 5 most popular products sold over the weekends.

Q4 Present the quarterly sales of each product for 2019 along with its total yearly sales. Note: each quarter sale must be a column and yearly sale as well. Order result according to product

Q5 Find an anomaly in the data warehouse dataset. write a query to show the anomaly and explain the anomaly in your project report.

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

STORE_ID	PROD_ID	STORE_TOTAL

Q7 Use the concept of Slicing calculate the total sales for the store “**Tech Haven**”and product combination over the months.

Q8 Create a materialized view named "SUPPLIER_PERFORMANCE_MV" that presents the monthly performance of each supplier.

Q9 Identify the top 5 customers with the highest total sales in 2019, considering the number of unique products they purchased.

Q10 Create a materialized view named "CUSTOMER_STORE_SALES_MV" that presents the monthly sales analysis for each store and then customers wise.

9. Tasks break-up

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

1. Identifying appropriate dimension tables, fact tables, and their attributes for the sales scenario. Based on that, creating a star-schema for DW with appropriate primary and foreign keys. To keep the attribute name and their data types consistent in DW, consult the attributes list of both transactional and master data. Also use "ELECTRONICA-DW'' as a name for the database where star-schema will be created.
2. Loading date Dimension table in the form of calendar using Java. i.e. all dates from year 2019.
3. Implementing the HYBRIDJOIN algorithm using Java and successfully loading transactional data into DW after joining it with MD.
4. Applying different analysis (described in Section 8) on DW using slicing, dicing, drill down, and materialized view concepts.
5. Writing a project report that should include project overview, schema for DW, HYBRIDJOIN algorithm, output of your OLAP queries, two shortcomings of HYBRIDJOIN, and what did you learn from the project?

10. What to submit

Each student has to submit the following files:

1. *createDW* –SQL script file to create star-schema for DW
   Note: your script should drop the table(s) if they already exist in the database.
2. *ETL* – Eclipse project with all Java files (StreamGenerator, HybdridJoin, and Controller) that implements all ETL operations including the HYBRIDJOIN algorithm. Make sure your Java Project should include all external libraries that are required to run the project. You will set both username and password as "root" for MySQL DB connection.
   Note: Your program should take the database credentials from the user at execution time.
3. *queriesDW* – SQL script file containing of all your OLAP queries
4. *projectReport* – a doc file containing all contents described in point 4 under the tasks break-up section.
5. *readMe* – a text file describing the step-by-step instructions to operate your project. In case of no submission of *readMe* file 5 marks will be deducted.

Note: all above files need to submit in a zipped folder named by your family name, student ID, and assessment version e.g. Bilal-12345v1.

11. When to submit

Due date: 17th November 2023, 5pm PST

Late penalty: maximum late submissions time is 24 hours after the due date. In this case a 10% late penalty will be applied.

12. Who to submit

The project should be submitted through Google Classroom.

NOTE: Every student has to complete the project individually. Each student's project source and report materials should be unique and done by his/her own. All assessments will be assessed through the turnitin and code checker systems and in case of finding any duplication or identical material 0 marks will be marked for the whole project.

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

Appendix

Marking guide

Project Component	Marks
createDW –SQL script file to create star-schema for DW	/20
The script should create all dimensions' and fact tables in DW and if any table with the same name exists already the script should drop that. It should also apply all primary and foreign keys on the right attributes.	
Implementing ETL	/30
The Java project 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. **MULTI-THREADING should be implemented.**	
queriesDW – SQL script file containing of all your OLAP queries	/30
The file should include OLAP queries for all tasks presented in Section 8.	
projectReport – a doc file containing all contents described in point 4 under the task break-up section.	/20
Report must contain a project overview, schema for DW, HYBRIDJOIN algorithm, output of your OLAP queries, two shortcomings of HYBRIDJOIN, and what did you learn from the project?	
readMe – a text file describing the step-by-step instructions to operate your project	-/5
The readMe file should contain a step-by-step guide to operate the project. In case of no submission of readMe file 5 marks will be deducted.	
Late submission penalty	-/10
TOTAL MARKS	/100