# *Course Project Report*

**DS-3003**

**DATA WAREHOUSING & BUSINESS INTELLIGENCE**


**Name:** **Hammad Javiad**

**Section:** **M**
**Roll number:** **i21-1661**

**Submitted to:** **Dr. Asif Naeem**

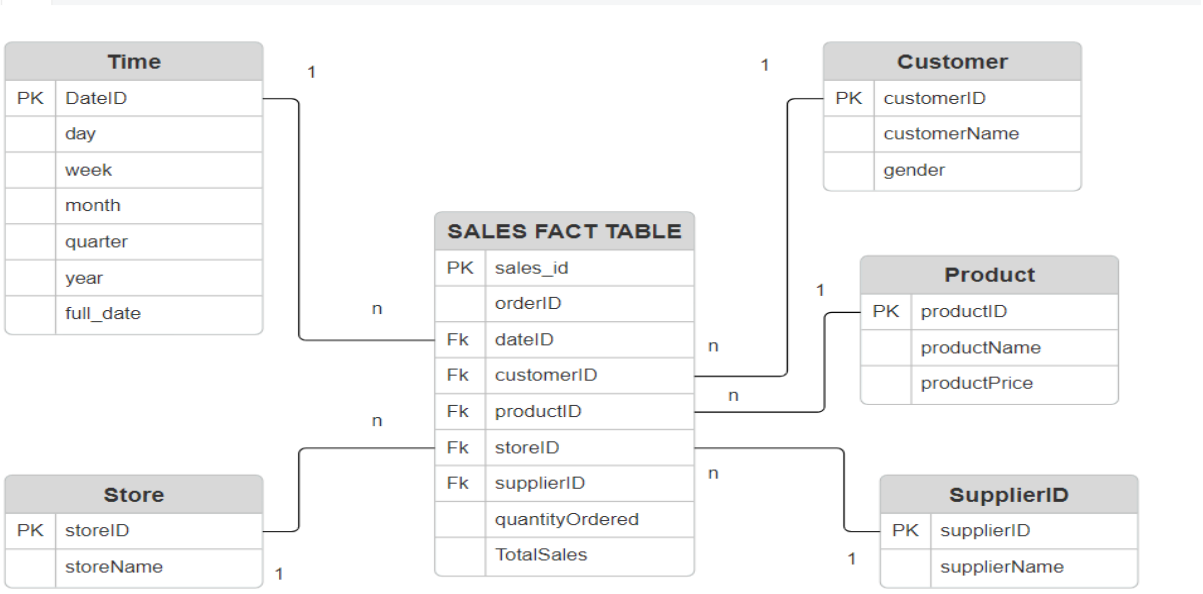
# Building and Analyzing Data Warehouse Prototype for Electronica Business Chain

## Problem Statement

This project involved designing and implementing a data warehouse (DW) for analyzing customer transactions for an electronic retail business. The main objective was to provide a near-real-time analytical solution to understand customer purchasing patterns. This involved extracting data from multiple sources, performing the ETL using the HYBRIDJOIN algorithm, and loading it into the DW.

## Star-Schema:

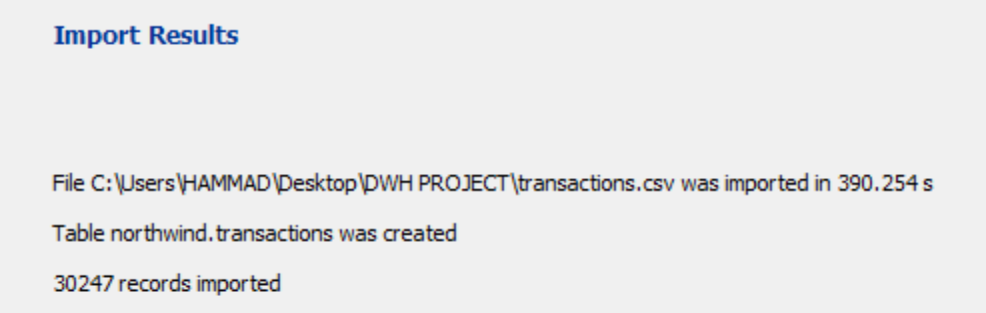
The schema designed for the DWH employs a star schema configuration, centralizing around a Sales fact table interconnected with several dimension tables: Date, Product, Supplier, Store, and Customer. Each dimension table is tailored to store specific attributes: Date captures time elements (day, week, month, quarter, year); Product details product-related information; Supplier, Store, and Customer maintain supplier, store, and customer details respectively. The Sales fact table, positioned at the core of this schema, records transactional data such as quantities ordered and total sales, linking to dimensions via foreign keys. This design facilitates efficient querying and analysis, essential for insights into sales patterns, customer behaviors, and operational efficiency, making it a robust solution for the data warehousing needs of a retail business environment.



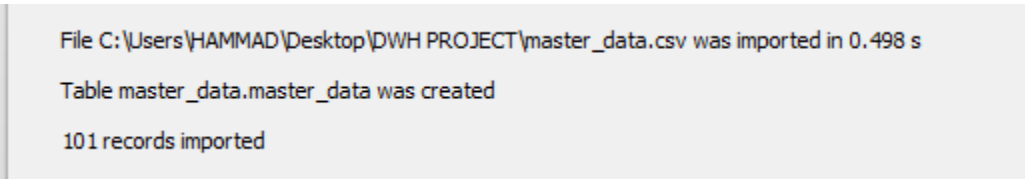
## Data Extraction:

The transaction and master_data was imported from csv file into mysql db.

1. Transactional Data: Obtained from a CSV file on port number




2. Master Data: Sourced from a different CSV file.




## HYBRIDJOIN Algorithm

The core of this project is the implementation of the HYBRIDJOIN algorithm within an ETL (Extract, Transform, Load) process, using Java and MySQL. The HYBRIDJOIN algorithm is a join method for merging transaction data with master data.

1. Transaction data is loaded into a multi-hash table and a custom queue.
2. Master data is loaded into a disk buffer according to the oldest node in the queue.
3. For each transaction, the algorithm finds a corresponding record in the master data, performs a join operation, and creates an enriched record.
4. The enriched data is then loaded into the DW, following the star schema whereas the corresponding nodes and tuples are removed from queue and hashtable and reloaded into streambuffer, hashtable and disk buffer. And these steps are repeated.

## DW Analysis:

– Query1 :Present total sales of all products supplied by each supplier with respect to quarter and month using drill down concept.

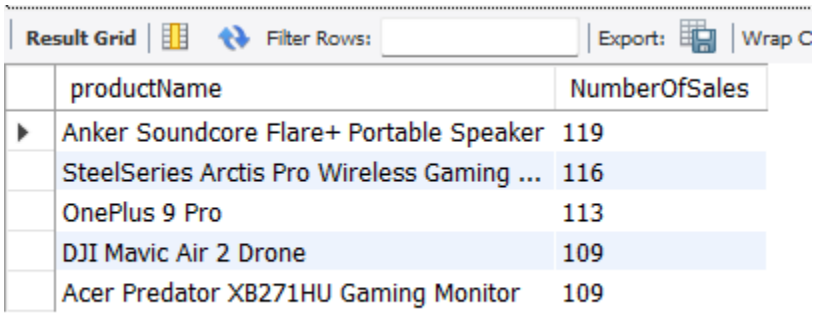
| supplierID | supplierName | Year | Quarter | Month | TotalSales |
|---|---|---|---|---|---|
| 1 | Apple Inc. | 2019 | 1 | 1 | 64289.14 |
| 1 | Apple Inc. | 2019 | 1 | 2 | 53799.23 |
| 1 | Apple Inc. | 2019 | 1 | 3 | 70549.05 |
| 1 | Apple Inc. | 2019 | 2 | 4 | 79059.00 |
| 1 | Apple Inc. | 2019 | 2 | 5 | 82399.03 |
| 1 | Apple Inc. | 2019 | 2 | 6 | 77919.00 |
| 1 | Apple Inc. | 2019 | 3 | 7 | 87039.04 |
| 1 | Apple Inc. | 2019 | 3 | 8 | 59539.04 |
| 1 | Apple Inc. | 2019 | 3 | 9 | 64969.07 |
| 1 | Apple Inc. | 2019 | 4 | 10 | 74569.03 |
| 1 | Apple Inc. | 2019 | 4 | 11 | 85969.02 |
| 1 | Apple Inc. | 2019 | 4 | 12 | 88459.07 |
| 2 | Dell Technologies | 2019 | 1 | 1 | 38999.65 |
| 2 | Dell Technologies | 2019 | 1 | 2 | 21999.80 |
| 2 | Dell Technologies | 2019 | 1 | 3 | 41499.65 |
| 2 | Dell Technologies | 2019 | 2 | 4 | 26099.78 |
| 2 | Dell Technologies | 2019 | 2 | 5 | 33199.71 |
| 2 | Dell Technologies | 2019 | 2 | 6 | 40499.65 |
| 2 | Dell Technologies | 2019 | 3 | 7 | 30299.74 |
| 2 | Dell Technologies | 2019 | 3 | 8 | 23399.77 |
| 2 | Dell Technologies | 2019 | 3 | 9 | 35099.68 |
| 2 | Dell Technologies | 2019 | 4 | 10 | 33499.70 |

Result 3

– Query2 :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".

| productID | month | totalSales |
|---|---|---|
| 21 | 1 | 9599.88 |
| 21 | 2 | 7999.90 |
| 21 | 3 | 7199.91 |
| 21 | 4 | 11999.85 |
| 21 | 5 | 8799.89 |
| 21 | 6 | 13599.83 |
| 21 | 7 | 6399.92 |
| 21 | 8 | 11199.86 |
| 21 | 9 | 10399.87 |
| 21 | 10 | 8799.89 |
| 21 | 11 | 15199.81 |
| 21 | 12 | 7999.90 |

– Query3: Find the 5 most popular products sold over the weekends.



Result Grid | Filter Rows: | Export: | Wrap C

| productName | NumberOfSales |
|---|---|
| Anker Soundcore Flare+ Portable Speaker | 119 |
| SteelSeries Arctis Pro Wireless Gaming ... | 116 |
| OnePlus 9 Pro | 113 |
| DJI Mavic Air 2 Drone | 109 |
| Acer Predator XB271HU Gaming Monitor | 109 |

– Query4: 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

| productName | Q1_Sales | Q2_Sales | Q3_Sales | Q4_Sales | Total_Yearly_Sales |
|---|---|---|---|---|---|
| Acer Aspire 5 Laptop | 13749.75 | 21449.61 | 15949.71 | 19799.64 | 70948.71 |
| Acer Predator Helios 300 Gaming Laptop | 39599.67 | 45599.62 | 41999.65 | 40799.66 | 167998.60 |
| Acer Predator X34 Curved Gaming Moni... | 17999.82 | 46999.53 | 26999.73 | 29999.70 | 121998.78 |
| Acer Predator XB271HU Gaming Monitor | 20399.66 | 17999.70 | 13799.77 | 16799.72 | 68998.85 |
| AirPods Pro | 8999.64 | 9999.60 | 6749.73 | 8999.64 | 34748.61 |
| Alienware Aurora Gaming PC | 52199.71 | 61199.66 | 43199.76 | 66599.63 | 223198.76 |
| Alienware AW2521HFL Gaming Monitor | 21499.57 | 14499.71 | 18999.62 | 15499.69 | 70498.59 |
| Amazon Echo Show 10 (3rd Gen) | 7999.68 | 7249.71 | 10249.59 | 7749.69 | 33248.67 |
| Anker Soundcore Flare+ Portable Speaker | 4099.59 | 3799.62 | 2899.71 | 4399.56 | 15198.48 |
| Anker Soundcore Liberty Air 2 Pro Earb... | 3379.74 | 4419.66 | 4289.67 | 4419.66 | 16508.73 |
| AOC CQ32G1 Curved Gaming Monitor | 12599.64 | 13649.61 | 13999.60 | 11899.66 | 52148.51 |
| Apple AirPods (3rd generation) | 6299.65 | 4859.73 | 6839.62 | 4499.75 | 22498.75 |
| Apple AirPods Max | 12099.78 | 21999.60 | 15949.71 | 20899.62 | 70948.71 |
| Apple HomePod Mini | 4099.59 | 3499.65 | 2999.70 | 3299.67 | 13898.61 |
| Apple iPad Pro (12.9-inch) | 31899.71 | 40699.63 | 30799.72 | 45099.59 | 148498.65 |
| Apple Watch SE | 6439.77 | 9519.66 | 11759.58 | 4199.85 | 31918.86 |
| Apple Watch Series 7 | 11199.72 | 9599.76 | 9599.76 | 9599.76 | 39999.00 |
| ASUS ROG Swift PG279Q Gaming Monitor | 29399.58 | 25199.64 | 20299.71 | 23799.66 | 98698.59 |
| ASUS TUF Gaming VG279QM Monitor | 12949.63 | 12599.64 | 6299.82 | 11199.68 | 43048.77 |
| Beats Powerbeats Pro Wireless Earphones | 11249.55 | 7999.68 | 6999.72 | 11499.54 | 37748.49 |
| Bose QuietComfort 35 II Wireless Headp... | 14099.53 | 10499.65 | 12599.58 | 10799.64 | 47998.40 |
| Bose SoundLink Revolve+ Bluetooth Spe... | 11699.61 | 10499.65 | 11399.62 | 9899.67 | 43498.55 |

Result 5

– Query5: Find an anomaly in the data warehouse dataset. write a query to show the anomaly and explain the anomaly in your project report.

One supplierID has multiple suppliernames.

| supplierID | supplierName |
|---|---|
| 19 | Canon Inc. |
| 19 | Canon Inc. |
| 19 | Nikon Corporation |
| 19 | Nikon Corporation |

Moreover, we can see Pakistan in the suppliers list:

| supplierID | supplierName |
|---|---|
| 39 | Sonos Inc. |
| 40 | NVIDIA Corporation |
| 41 | Sennheiser |
| 42 | Ring (Amazon) |
| 43 | Ultimate Ears (Logitech) |
| 44 | Beats by Dre (Apple Inc.) |
| 45 | Amazon.com, Inc. |
| 51 | Pakistan |
| NULL | NULL |

– Query6: Create a materialized view with the name “STOREANALYSIS_MV” that presents the product-wise sales analysis for each store.

| storeID | productID | storeTotal |
|---|---|---|
| 1 | 1 | 347596.84 |
| 2 | 2 | 400396.92 |
| 1 | 3 | 506996.62 |
| 3 | 4 | 134746.15 |
| 2 | 5 | 207596.54 |
| 4 | 6 | 159496.81 |
| 3 | 7 | 84996.60 |
| 5 | 8 | 570596.83 |
| 6 | 9 | 1266996.38 |
| 7 | 10 | 56156.88 |
| 1 | 11 | 817496.73 |
| 4 | 12 | 53546.43 |
| 2 | 13 | 226846.51 |
| 3 | 14 | 118196.06 |
| 2 | 15 | 408196.86 |
| 6 | 16 | 617996.91 |
| 7 | 17 | 123596.91 |
| 5 | 18 | 647996.40 |
| 6 | 19 | 932996.89 |

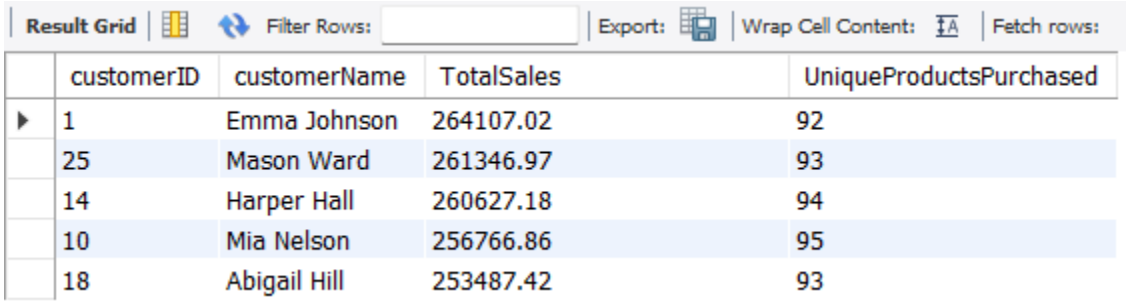
– Query7: Use the concept of Slicing to calculate the total sales for the store “Tech Haven”and product combination over the months.

| productName | month | TotalSales |
|---|---|---|
| Acer Aspire 5 Laptop | 7 | 10999.80 |
| Acer Aspire 5 Laptop | 8 | 12099.78 |
| Acer Aspire 5 Laptop | 9 | 9349.83 |
| Acer Aspire 5 Laptop | 10 | 14299.74 |
| Acer Aspire 5 Laptop | 11 | 12649.77 |
| Acer Aspire 5 Laptop | 12 | 7699.86 |
| Apple iPad Pro (12.9-inch) | 1 | 25299.77 |
| Apple iPad Pro (12.9-inch) | 2 | 28599.74 |
| Apple iPad Pro (12.9-inch) | 3 | 43999.60 |
| Apple iPad Pro (12.9-inch) | 4 | 34099.69 |
| Apple iPad Pro (12.9-inch) | 5 | 38499.65 |
| Apple iPad Pro (12.9-inch) | 6 | 32999.70 |
| Apple iPad Pro (12.9-inch) | 7 | 31899.71 |
| Apple iPad Pro (12.9-inch) | 8 | 16499.85 |
| Apple iPad Pro (12.9-inch) | 9 | 17599.84 |
| Apple iPad Pro (12.9-inch) | 10 | 21999.80 |
| Apple iPad Pro (12.9-inch) | 11 | 21999.80 |
| Apple iPad Pro (12.9-inch) | 12 | 25299.77 |
| Dell Inspiron 14 Laptop | 1 | 31199.61 |
| Dell Inspiron 14 Laptop | 2 | 27199.66 |
| Dell Inspiron 14 Laptop | 3 | 24799.69 |
| Dell Inspiron 14 Laptop | 4 | 23999.70 |

– Query8: Create a materialized view named "SUPPLIER_PERFORMANCE_MV" that presents the monthly performance of each supplier.

| supplierID | supplierName | month | monthlyPerformance |
|---|---|---|---|
| 1 | Apple Inc. | 8 | 127898.14 |
| 1 | Apple Inc. | 9 | 128348.07 |
| 1 | Apple Inc. | 10 | 142378.01 |
| 1 | Apple Inc. | 11 | 139518.15 |
| 1 | Apple Inc. | 12 | 153258.18 |
| 2 | Dell Technologies | 1 | 109199.01 |
| 2 | Dell Technologies | 2 | 98699.11 |
| 2 | Dell Technologies | 3 | 97599.13 |
| 2 | Dell Technologies | 4 | 98099.13 |
| 2 | Dell Technologies | 5 | 113399.02 |
| 2 | Dell Technologies | 6 | 114199.01 |
| 2 | Dell Technologies | 7 | 76599.33 |
| 2 | Dell Technologies | 8 | 58399.47 |
| 2 | Dell Technologies | 9 | 67499.40 |
| 2 | Dell Technologies | 10 | 72399.37 |
| 2 | Dell Technologies | 11 | 69299.39 |
| 2 | Dell Technologies | 12 | 64999.45 |
| 3 | Samsung Electronics | 1 | 70699.39 |
| 3 | Samsung Electronics | 2 | 70399.36 |
| 3 | Samsung Electronics | 3 | 70699.31 |
| 3 | Samsung Electronics | 4 | 79199.28 |

– Query9: Identify the top 5 customers with the highest total sales in 2019, considering the number of unique products they purchased.



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

| customerID | customerName | TotalSales | UniqueProductsPurchased |
|---|---|---|---|
| 1 | Emma Johnson | 264107.02 | 92 |
| 25 | Mason Ward | 261346.97 | 93 |
| 14 | Harper Hall | 260627.18 | 94 |
| 10 | Mia Nelson | 256766.86 | 95 |
| 18 | Abigail Hill | 253487.42 | 93 |

– Query10: Create a materialized view named "CUSTOMER_STORE_SALES_MV" that presents the monthly sales analysis for each store and then customers wise.

| storeID | customerID | customerName | month | monthlySales |
|---|---|---|---|---|
| 1 | 7 | Jackson Taylor | 1 | 499.98 |
| 1 | 4 | Liam Wilson | 1 | 14449.84 |
| 1 | 32 | Addison Cooper | 1 | 9249.94 |
| 1 | 46 | Savannah Allen | 1 | 7699.93 |
| 1 | 3 | Olivia Davis | 1 | 11699.86 |
| 1 | 12 | Amelia Thomas | 1 | 10299.91 |
| 1 | 5 | Ava Martinez | 1 | 9899.93 |
| 1 | 17 | Grayson Lewis | 1 | 12049.93 |
| 1 | 29 | Samuel Mitchell | 1 | 10749.93 |
| 1 | 42 | Hazel Turner | 1 | 20599.87 |
| 1 | 28 | Lily Moore | 1 | 11549.91 |
| 1 | 19 | Logan Adams | 1 | 5449.93 |
| 1 | 9 | Lucas Harris | 1 | 10249.89 |
| 1 | 14 | Harper Hall | 1 | 8799.90 |

## SHORTCOMINGS:

1. Since the algorithm involves loading chunks of data into memory (both the transaction data into the queue and a segment of master data into the disk buffer), it can be memory-intensive, especially with large datasets. If the dataset size significantly exceeds the available memory, it can lead to performance issues or even cause the system to run out of memory.
2. The performance of the algorithm can be significantly bottlenecked by the repeated disk I/O operations. This becomes more evident as the data size grows. In scenarios where the same or similar join operations are performed multiple times, caching can drastically reduce the need for these expensive I/O operations, thereby improving performance.

## What i learned from this project:

Throughout this project, I have significantly enhanced my understanding and practical skills in both data warehousing and Java programming. By meticulously applying data warehousing concepts, I developed a deeper appreciation for the intricacies of ETL processes, particularly in implementing the join algorithm: HYBRIDJOIN. This project solidified my grasp on the theoretical aspects of data warehousing and also challenged me to effectively address real-world problems, such as multi-threading and performance optimization. This hands-on experience was instrumental in deepening my knowledge of Java, particularly in database connectivity, data structures, and performance considerations. All in all, this project was equivalent to a semester project but based on both Data Structures and Database management, therefore, this was a significant step in my academic and professional journey, blending theoretical knowledge with practical application in a meaningful and challenging way.

Connectors needed:

1. mysql jdbc jar file
2. Apache set commons collections jar file

Environment Setup:

IntelliJ IDEA was used for the development with specific configurations for Java and Apache Spark 3.4. MySQL Connector/J 8.2.0 was integrated for JDBC connections.