**Project report**

**Overview:**

I started the project by first converting the CSV files into SQL by using a converter. By doing this I loaded the two CSVs in SQL where I had previously created a database called electronica-dw.

Then I made the necessary dimension and fact tables, by taking a look at the master and transactional data.

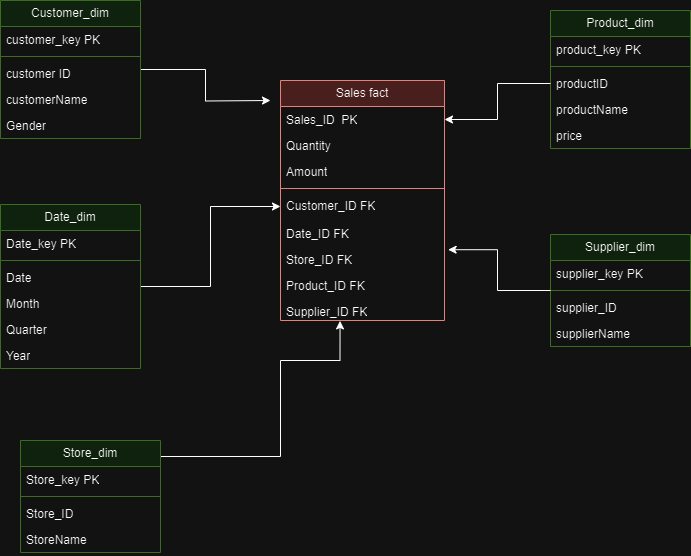
Then I set up my eclipse environment, downloaded the necessary jar files required for the libraries.

Afterwards I applied the hybrid join algorithm and loaded the new data into the dimension table.

The given queries were applied then.

**Star schema for data warehouse:**

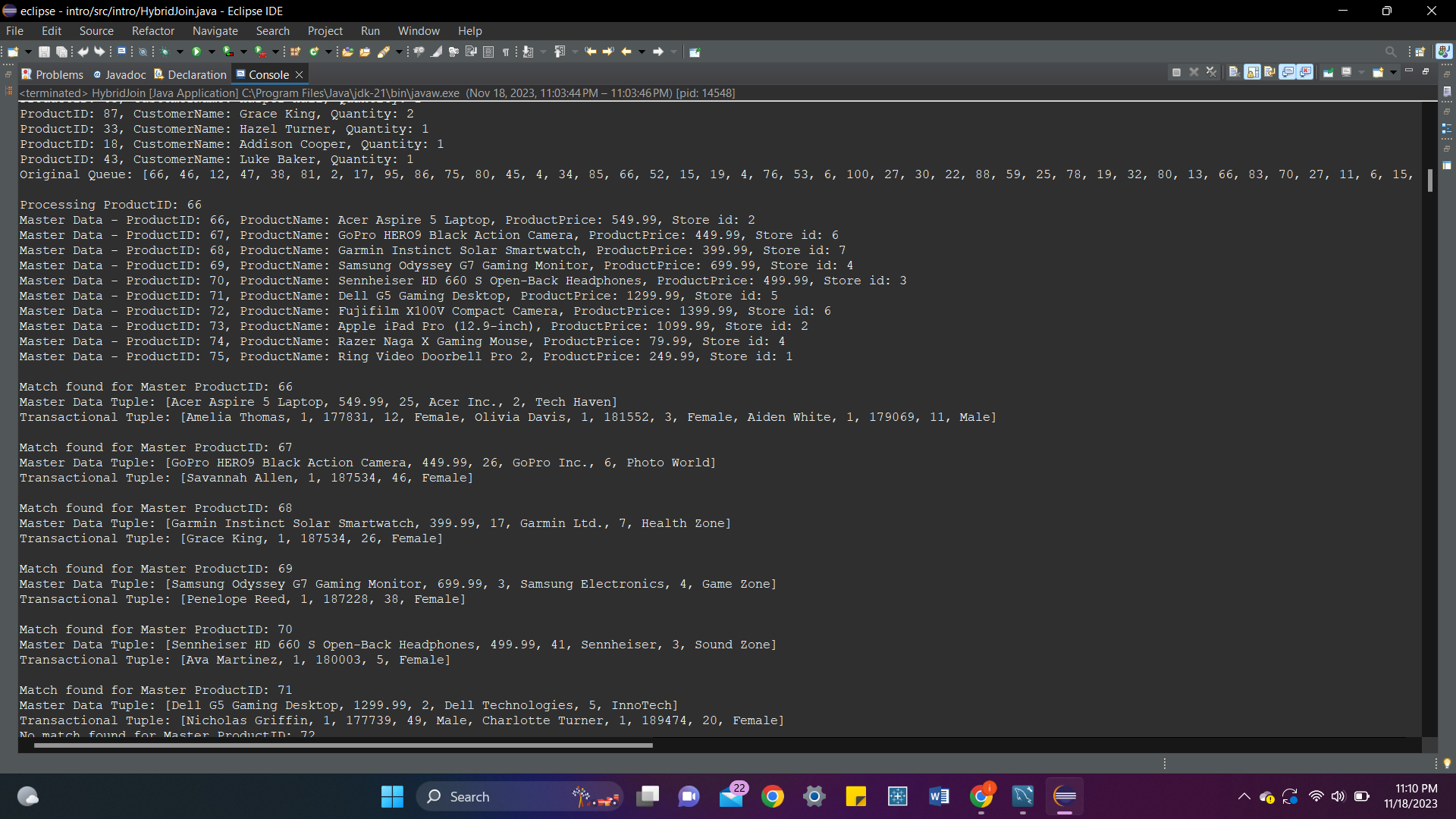
The following is the star schema keeping master and transactional data in view:



**Hybrid join algorithm:**

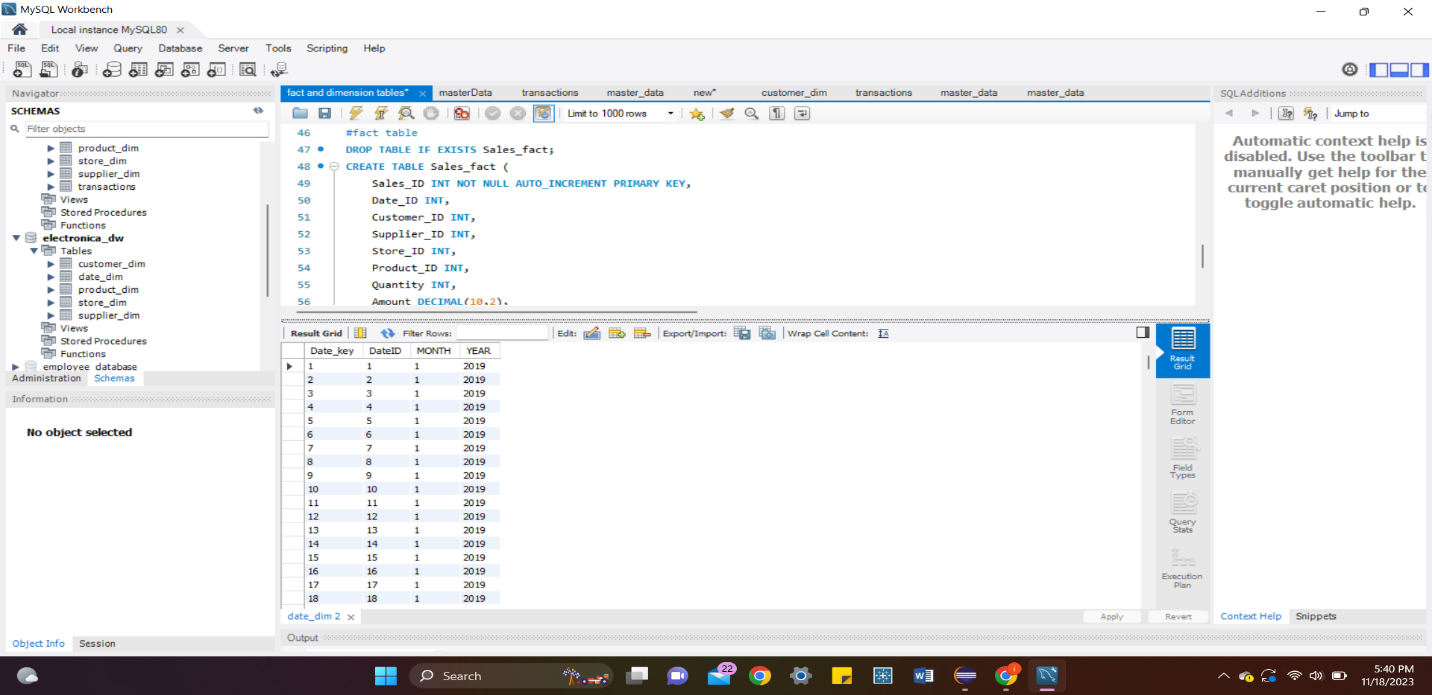
The hybrid join algorithm works in the following way:

* First of all, 1000 columns are loaded from the transactional data into the multi valued hash table with product IDs as their keys. As I am low on disk space I am loading 100 columns instead of 1000.
* Then these product IDs are stored in a doubly linked list called queue that follows the basic principal of FIFO.
* Then the head (first value inserted into the queue) is retrieved and its corresponding next 10 product IDs are loaded into the master data hash table. Here the product IDs are keys and the remaining tuple is stored as the multi values. E.g. if dequeued product ID is 10 the master data table holds product IDs from 10….20.
* Then each product ID is checked from the transactional hash table, if it exists there and as well as in the queue, the tuples extracted from master and transactional are immediately inserted into the dimension tables previously created in SQL.
* This happens in a loop until the queue is empty.

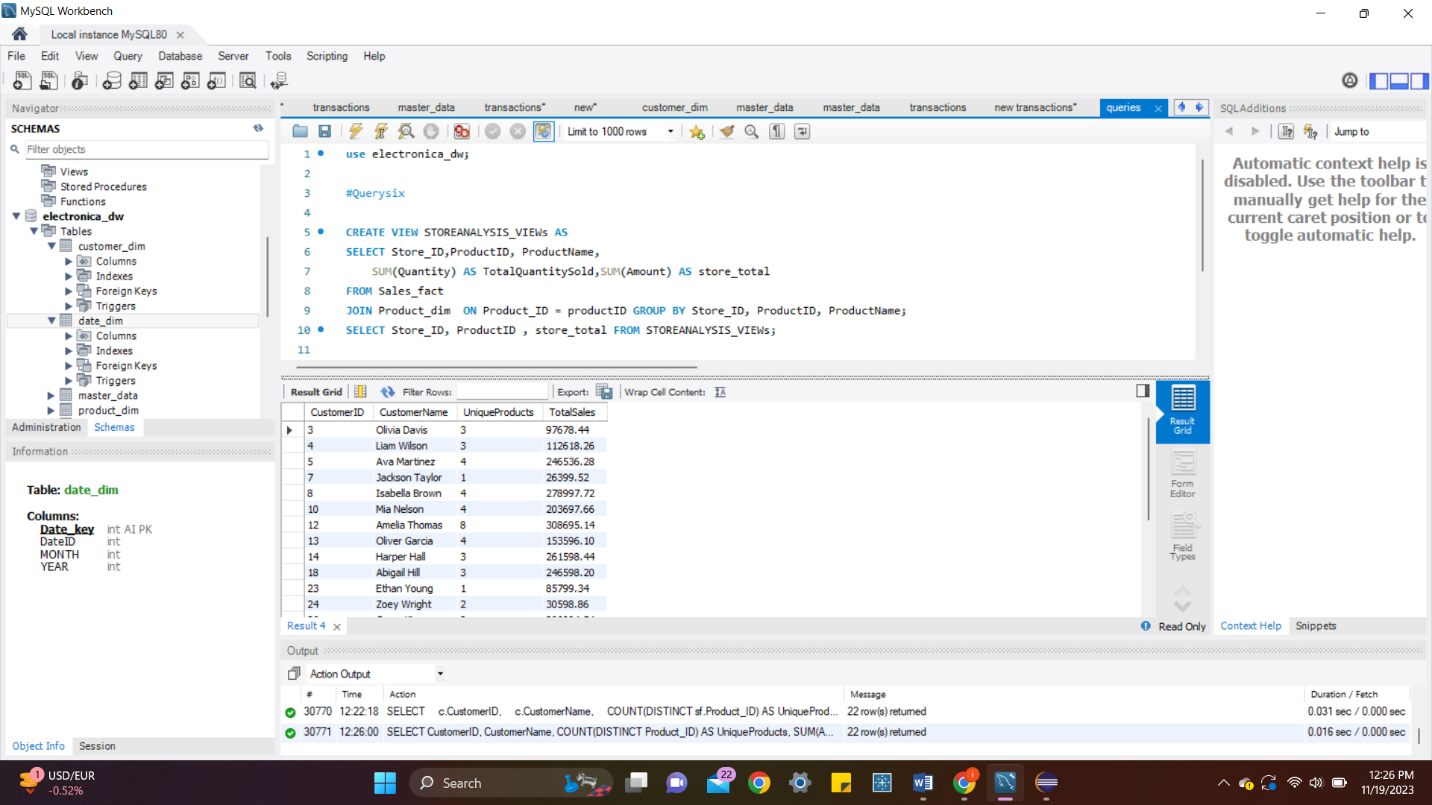
****

**OLAP queries:**

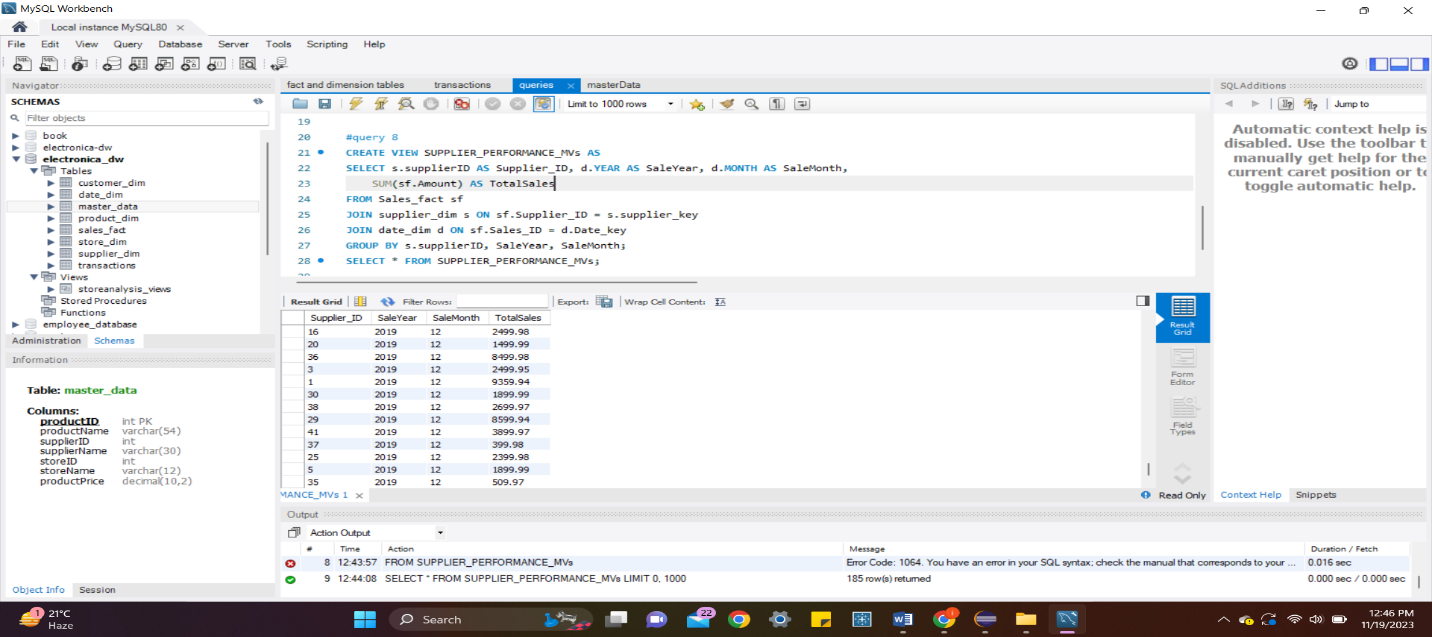
**6**



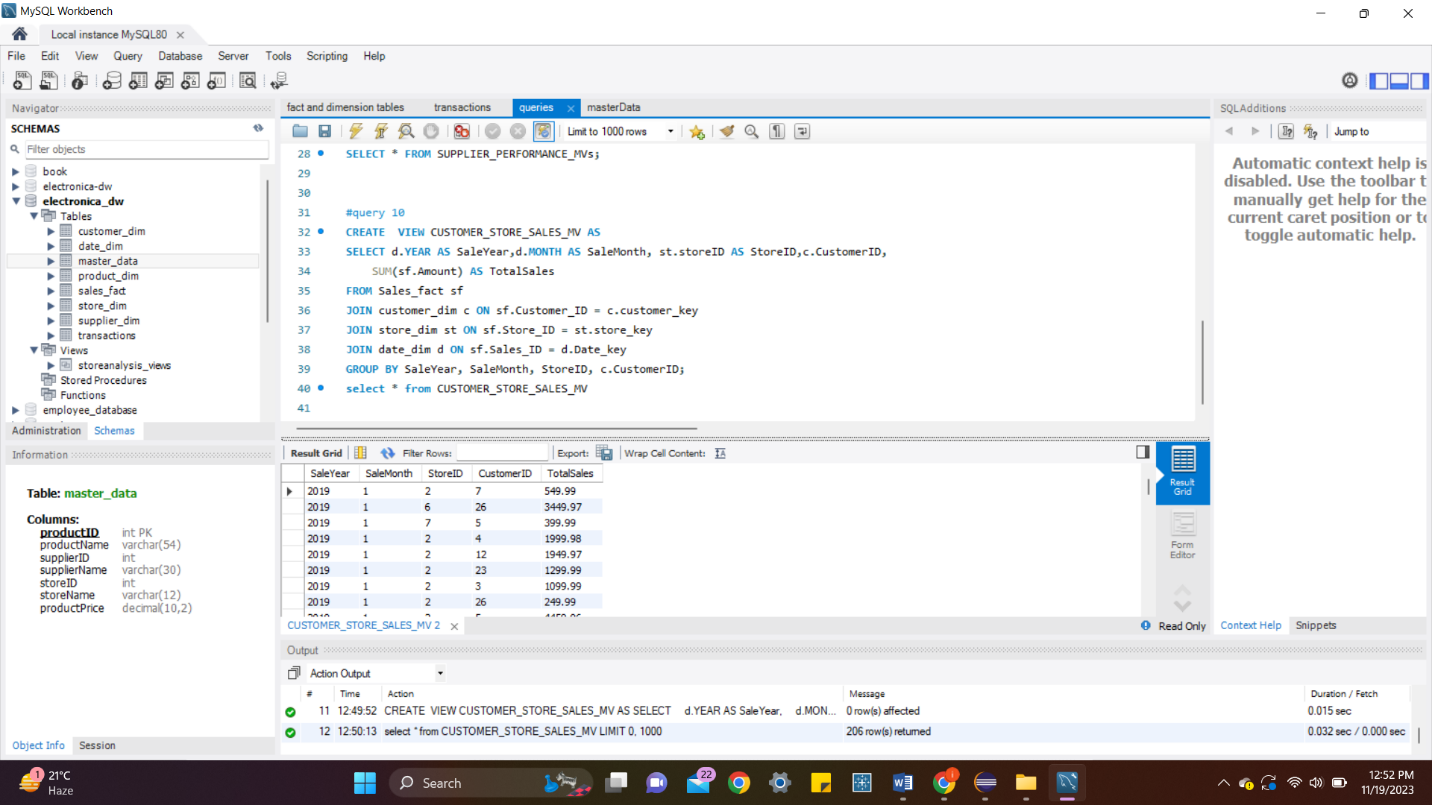
9



8



10



**Shortcomings of hybrid join:**

Initial Loading Overhead: The process of loading 1000 tuples into the Hash table during the initial iteration could potentially introduce a significant overhead, particularly when dealing with extensive customers' transaction tables. This preliminary loading phase may demand a considerable amount of time and computational resources.

Dependency on External Libraries: The reliance on external libraries, such as the Apache multi-hash-map, introduces a reliance on third-party components. This could lead to dependencies and compatibility issues, potentially restricting the algorithm's portability across diverse computing environments.

Dependency on Join Attribute: The algorithm's efficacy is closely tied to the join attribute for both indexing and joining. In cases where the join attribute exhibits low selectivity, indicating that numerous tuples share the same join attribute value, the algorithm might encounter performance challenges. Processing a larger number of tuples with identical join attribute values could adversely impact overall efficiency.

**Learning outcomes:**

In this project, I learned a lot of things. First I got to learn the JAVA language and was introduced to the complexities and functions there.

I learnt to set up the eclipse environment for java and install the necessary libraries, where finding the correct version which is compatible was a deal in itself.

I learned how to connect SQL with java and populate it.

Also learnt how to use link list and hash maps in java.

.. 1000

… root input

…date

..commenting