

ASSIGNMENT COVER SHEET

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course Code and Description | UECS 3203 ADVANCED DATABASE SYSTEMS | | | |
| Lecturer Name | Dr. Sugumaran a/l Nallusamy | | | |
| Assignment Title | Assignment 1 | | | |
| **DECLARATION** | | | | |
| We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person. | | | | |
| Programme | | Student ID Numbers | Student Names | Practical  Group |
| SE | | 2002759 | AVEN DING XIAN KAI | 1 |
| SE | | 2400056 | CHOO WEI XIANG | 1 |
| SE | | 2301374 | SIA LI ZE | 1 |
| SE | | 2003632 | TH’NG ZI QIN | 1 |

Table of Contents

[CHAPTER 1 DATA SELECTION 3](#_Toc175157247)

[CHAPTER 2 DATA IMPORT 4](#_Toc175157248)

[CHAPTER 3 TABLE CREATION & INDEXES 5](#_Toc175157249)

[3.1 Table Structure 5](#_Toc175157250)

[3.2 Indexes Suggestion 6](#_Toc175157251)

[CHAPTER 4 CRUD OPERATIONS 7](#_Toc175157252)

[4.1 Create 7](#_Toc175157253)

[4.2 Read 7](#_Toc175157254)

[4.3 Update 7](#_Toc175157255)

[4.4 Delete 7](#_Toc175157256)

[CHAPTER 5 TRANSACTION MANAGEMENT 8](#_Toc175157257)

[CHAPTER 6 TRIGGERS 9](#_Toc175157258)

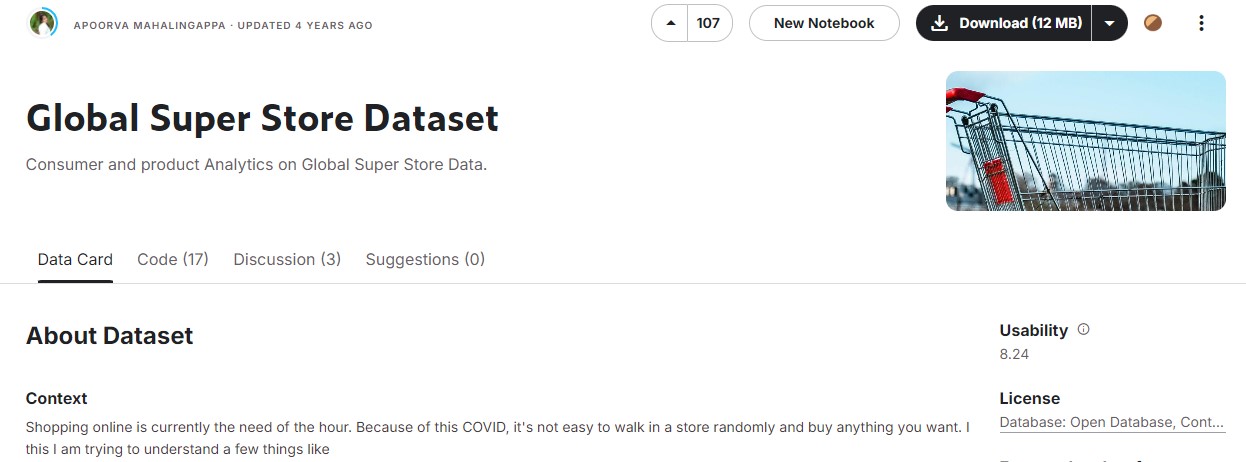
[CHAPTER 7 DATA ANALYSIS 10](#_Toc175157259)

[CHAPTER 8 REFERENCES 11](#_Toc175157260)

# DATA SELECTION

The Global Superstore Dataset was chosen by our group from Kaggle for this assignment. The dataset includes information on detailed order records, shipping details, customer profiles, market research, and product details. The data is contained in a single CSV file and is organized into 51,291 rows, each with 23 attributes.

The reason we decided on this dataset was that every single one of the columns had complete and rich data. The dataset is also appropriate from a variety of analytical angles because it provides a broad range of attributes. This versatility helps us achieve our goals for data analysis by enabling us to extract valuable insights from a variety of perspectives. This dataset is a great place for novices to start learning about e-commerce analytics in the field of data science. It offers lots of chances to extract substantial.



*Figure 1.1: Global Super Store Dataset* (*Global Super Store Dataset*, n.d.)

# DATA IMPORT

# TABLE CREATION & INDEXES

## Table Structure

The subsequent step after importing the dataset is to normalize the data into several tables to minimize redundancy and improve the efficiency of the analysis. The dataset was divided into four important tables, which we designated Customer, Shipping, Product, and OrderItem.

The Customer table contains customer-related information, including CustomerID as the primary key, and fields such as CustomerName, Segment, City, State, Country, Market, and Region. This table stores all the necessary details about each customer, ensuring that customer data is organized and easily accessible.

Details about each order's shipment are recorded in the Shipping table. It establishes a parent-child relationship between the two tables using ShippingID as the primary key and foreign key references to the CustomerID from the Customer table. It is simple to follow the logistics of each order with the help of this table, which also includes information like the OrderID, OrderDate, ShipDate, and ShipMode. Since a single shipment can cover multiple orders, OrderID is inappropriate for use as a foreign key in the Shipping table. It would be misleading to indicate that each shipment is associated with a single order if OrderID were a foreign key. In the OrderItem table, orders are tracked individually, but ShippingID continues to be the primary key in the Shipping table. By avoiding redundancy and keeping a flexible system that accurately reflects the many-to-one relationship between shipments and orders, this structure enables us to track shipments independently of specific orders.

The Product table stores data about the products, with ProductID as the primary key. It categorizes products into categories and subcategories through fields like Category and SubCategory, including the ProductName. This table provides a structure for managing product information independently from customer and order data.

Last but not least, every single order's specific products are stored in the OrderItem table. OrderID and ProductID make up its composite primary key, which guarantees the uniqueness of every order and product combination. Important metrics like SalesAmount, Quantity, Discount, Profit, ShippingCost, and OrderPriority are included in this table. Each order item is linked to its corresponding product through a foreign key reference from the OrderItem table to the Product table via ProductID.

## Indexes Suggestion

1. Index on CustomerID in the Shipping table (idx\_shipping\_customerid)

CustomerID column in the Shipping table act as a foreign key from the Customer table that connects each shipping record to a customer. The efficiency of queries that filter or join the Shipping table based on CustomerID can be increased by indexing on CustomerID. This is especially helpful for tasks that include merging shipment records with customer information or searching through all shipments for a particular customer. These operations would need a full table scan, which can be time-consuming, particularly for large datasets, if there was no index.

1. Index on CustomerName in the Customer table (idx\_customer\_name)

The CustomerName column in the Customer table is often used in queries for searching, filtering, or reporting purposes. Operations that involve looking up customers by name perform better when CustomerName is indexed. This index, as opposed to a full table scan, guarantees that operations such as finding a customer by name or generating reports that filter customers by name are completed more quickly if you regularly need to do so.

1. Index on OrderPriority in the OrderItem table (idx\_orderitem\_orderpriority)

Orders can be filtered or sorted according to their priority level using the OrderPriority column in the OrderItem table. The performance of queries involving filtering or sorting according to order priority is enhanced by creating an index on this column. This index is useful for operations that need to quickly retrieve orders with a certain priority level or for creating reports or analytics that classify orders according to their priority.

1. Index on ProductID in the OrderItem table (idx\_orderitem\_productid)

Each order item in the OrderItem table is associated with a unique product in the Product table via a foreign key found in the ProductID column. Queries that filter order items based on the product or join OrderItem with Product are sped up by indexing ProductID. When you have to look up every order item connected to a specific product or carry out operations involving product details, this is essential for performance.

# CRUD OPERATIONS

## Create

## Read

## Update

## Delete

# TRANSACTION MANAGEMENT

# TRIGGERS

# DATA ANALYSIS

# REFERENCES

* *Global Super Store Dataset*. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/apoorvaappz/global-super-store-dataset

‌