**Sales Data Management and Analysis for Global Retail Solutions**

### **Problem Statement:**

Global Retail Solutions Company aims to improve its sales management system implementation by opting SQL for efficient storage, organisation, and retrieval of sales data. The company operates across multiple regions and provides products in various categories.

Your goal is to create the sales database, develop and implement the requisite tables and write SQL commands that will enable the management team to gain profitable insights. The purpose of this project will be to create tables, querying sales data operations, filtering records, grouping results by different categories, and sorting data based on specific criteria.

### **Project Requirements:**

1. **Database Structure**: Create tables in SQL to store sales, customer, and product information.
2. **Data Retrieval**: Write SQL queries to extract relevant sales data based on different conditions (e.g., region, product, date).
3. **Grouping and Aggregation**: Use SQL to group data by product category and region, and calculate key metrics like total sales, average order amount, and total quantity sold.
4. **Sorting Data**: Order the query results to identify top-selling products, most profitable regions, and the highest revenue generated.
5. **SQL Commands**: Apply SELECT, WHERE, GROUP BY, and ORDER BY clauses effectively to meet the company’s reporting needs.

### **SQL Tasks:**

1. **Create “Global Retail Solutions” Schema:** The following tables need to be created in SQL to store the data inside  **“Global Retail Solutions”** Schema:
2. **Products Table**:

* product\_id: Unique identifier for each product.
* product\_name: Name of the product.
* category: Category to which the product belongs (e.g., Electronics, Apparel).
* price: Price of each product.

1. **Customers Table**:

* customer\_id: Unique identifier for each customer.
* customer\_name: Name of the customer.
* region: Region where the customer is located.
* customer\_segment: Customer segmentation (e.g., Regular, Premium).

1. **Sales Table**:

* sale\_id: Unique identifier for each sale.
* order\_date: Date the order was placed.
* customer\_id: ID of the customer making the purchase (foreign key from Customers table).
* product\_id: ID of the product being purchased (foreign key from Products table).
* quantity: Number of units sold.
* total\_amount: Total amount of the sale (calculated as quantity \* price).

1. **Inserting Data into Tables**: After creating the tables, insert data into the **Products**, **Customers**, and **Sales** tables. Use the below dataset to be inserted into SQL tables.

**Dataset :**

[**Products**](https://drive.google.com/file/d/14WNMx38wvJFcDUAWrDqkEQ5Lgu-M7LfG/view?usp=sharing)

[**Customers**](https://drive.google.com/file/d/1EfgOMGjTl2Hejvy7tMncXEatjDu0m7Ou/view?usp=sharing)

[**Sales**](https://drive.google.com/file/d/1HMopAmAK_L84AprySn4Wrzu-g4F5cY9B/view?usp=sharing)

1. **Data Analysis Using SQL**: Once the tables are populated, perform the following SQL queries to retrieve meaningful insights.

* Retrieve total sales and the number of products sold in each region:
* Find the top 3 best-selling products based on total sales amount:
* Identify the customer segment that contributes the most to the total revenue:
* Find the average order value for each product category:
* List all sales in the "Electronics" category where the quantity sold is greater than 2.

### **Expected Outcomes:**

1. **Efficient Database Design**: You'll create normalized tables to store sales, customer, and product data in SQL.
2. **Effective SQL Queries**: You'll apply SQL commands to filter, group, and aggregate data.
3. **Data Insights**: You’ll retrieve key sales insights, such as total sales per region, top products, and customer segmentation analysis.
4. **Real-world Application**: You’ll practice designing databases and writing complex SQL queries for business analysis, which is vital for any data management role.