

## Project Report: Final SQL Task - Sales Database Analysis

### 1. Project Overview and Objectives

This document outlines the work completed for the final SQL project. The primary objective was to conduct an in-depth analysis of a relational database for a retail company to extract actionable business intelligence. The project required applying a wide range of standard and advanced SQL techniques to query the database, answer specific business questions, and create reusable database objects like views and stored procedures.

### 2. Database Schema

The analysis was performed on a relational database designed to manage sales operations. The schema consisted of eight interconnected tables:

- **Core Transactional Tables:** orders, order\_items
- **Business Entity Tables:** customers, products, staffs, stores
- **Product Organization Tables:** categories, brands

These tables are linked through primary and foreign key relationships, allowing for complex queries that join information across different business areas. For example, the **order\_items** table links **orders** and **products** to detail the contents of each sale, while the **orders** table connects to **customers**, **stores**, and **staffs** to provide context for each transaction.

### 3. Data Analysis, Queries, and Database Objects

The project was structured around a series of analytical questions that required progressively more advanced SQL solutions.

#### 3.1. Business Performance Queries

The initial set of queries focused on high-level business performance metrics:

- **Top 10 Best-Selling Products:** A query was developed to identify the ten best-selling products by joining **order\_items** and **products** and aggregating the total quantity sold for each product.
- **Total Sales per Store:** To analyze store performance, a query was written to calculate the total revenue generated by each store. This involved joining the **stores**, **orders**, and **order\_items** tables and summing the sales amount, grouped by store name.

- **Top 10 Customers:** A query was implemented to find the top 10 customers based on their total spending, providing insight into the most valuable customers.
- **Sales by Category:** Product sales were analyzed by category to understand which product lines are most popular. This required joining four tables (**categories, products, order\_items, orders**).
- **Sales by Staff:** The performance of sales staff was evaluated by calculating the total number of orders processed by each staff member.

### 3.2. Advanced SQL Implementation

To demonstrate mastery of advanced SQL features, the following objects and techniques were implemented:

- **View Creation:** A view named **daily\_sales** was created. This view simplifies the process of analyzing daily performance by pre-calculating the total number of orders and total revenue for each store on a daily basis. This abstracts away the complexity of the underlying joins and aggregations.
- **Stored Procedure:** A stored procedure named **get\_sales\_by\_year** was developed. This procedure accepts a year as an input parameter and returns the total sales revenue for that specific year, making it a reusable tool for annual reporting.
- **Window Functions:** To perform more sophisticated ranking, window functions were used to rank products within each category based on the number of units sold. The following functions were implemented and compared:
  - **ROW\_NUMBER():** Assigns a unique, sequential integer to each product.
  - **RANK():** Assigns a rank, leaving gaps in the sequence for ties.
  - **DENSE\_RANK():** Assigns a rank without any gaps in the sequence.
- **Common Table Expression (CTE):** A CTE was used to structure a complex query designed to find the average order value for each customer. The CTE first calculated the total value of each individual order before the outer query aggregated these values to compute the average for each customer.

## 4. Conclusion

This project successfully applied a broad set of SQL skills to analyze a sales database and derive meaningful business insights. The work progressed from fundamental data retrieval and aggregation to advanced techniques, including the creation of database objects like views and stored procedures, and the use of modern SQL features like window functions and CTEs. The project effectively demonstrated the ability to translate complex business questions into efficient and accurate SQL queries.