**Report: SQL Analysis and Optimization**

**Requirement 1: Total Sales and Customer Insights**

* **Objective**: Calculate the total spending of each customer and rank the top 10 customers by total sales.
* **Approach**:
  + Used a WITH clause to create a Common Table Expression (CTE) named invoiceCountItems to count items per invoice.
  + Created a second CTE named profit\_per\_customer to calculate the number of invoices, total spending, and items for each customer.
  + Ordered results by total spending in descending order and limited the output to the top 10 customers.
* **Top 10 Customers**:

| **Customer ID** | **Full Name** | **Number of Invoices** | **Total Per Customer** | **Number of Items** |
| --- | --- | --- | --- | --- |
| 6 | Helena Holý | 7 | 49.62 | 38 |
| 26 | Richard Cunningham | 7 | 47.62 | 38 |
| 57 | Luis Rojas | 7 | 46.62 | 38 |

**Requirement 2: Ranking Products by Sales**

* **Objective**: Rank products by total sales and identify the top-selling products.
* **Approach**:
  + Used a CTE invoiceline\_trackname to join InvoiceLine and Track tables, calculating total sales (unitprice \* quantity) and the total quantity sold.
  + Applied the RANK() window function to rank products by quantity sold in descending order.
  + Limited the result to the top 10 products.
* **Top-Selling Products**:

| **Track ID** | **Product Name** | **Total Sales** | **Quantity Sold** | **Rank** |
| --- | --- | --- | --- | --- |
| 2 | Balls to the Wall | 1.98 | 2 | 1 |
| 8 | Inject The Venom | 1.98 | 2 | 1 |
| 9 | Snowballed | 1.98 | 2 | 1 |

**Requirement 3: Indexing and Performance Optimization**

* **Objective**: Optimize the query to calculate total sales per customer using indexing.
* **Steps**:
  1. **Before Index**:
     + Observed a **full table scan** during the aggregation process.
  2. **Index Creation**:
     + Created an index on Invoice.CustomerId using:

sql

Copy code

CREATE INDEX IF NOT EXISTS idx\_invoice\_customerid ON Invoice(CustomerId);

* 1. **After Index**:
     + Execution plan shows optimization with aggregation (AggStep, AggFinal) using the index.
* **Key Metrics**:
  1. **Before Index**: Full table scan (OpenRead, Rewind operations observed).
  2. **After Index**: Use of aggregation steps (AggStep, AggFinal), indicating optimized query execution.