Adventure Works Cycle Analysis

Introduction

Adventure Works Cycles is a fictional company that sells bicycles and related accessories. In this project, we analyse various facets of Adventure Works Cycles' business using a comprehensive dataset. The dataset includes sales data, product details, customer demographics, and product returns, providing insights into both operational and customer aspects of the business.

Problem Statements

1. Identifying Customers Eligible for Special Coupon
2. Identifying Blockbuster Territories for Each Subcategory

Data Dictionary

1. Sales:
   - ✓ OrderDate: Date when the order was placed.
   - ✓ StockDate: Date when the stock was updated.
   - ✓ OrderNumber: Unique identification number for each order.
   - ✓ Product ID: Unique ID of the product ordered.
   - ✓ Customer_ID: Unique ID of the customer who placed the order.
   - ✓ Territory_ID: Unique ID of the sales territory:
   - ✓ OrderLineltem: Order line item number.
   - ✓ OrderQuantity: Quantity of products ordered.
2. Calender:
   - ✓ Date: Dates listed, potentially representing a range of dates of interest.
3. Territory:
   - ✓ Sales Territory_Id: Unique ID for each sales territory.
   - ✓ Region: The region of the sales territory.
   - ✓ Country: The country of the sales territory.
   - ✓ Continent: The continent of the sales territory.
4. Subcategory:
   - ✓ Subcategory_Id: Unique ID for each product subcategory.
   - ✓ SubcategoryName: Name of the product subcategory.
   - ✓ Category_id: The category to which the subcategory belongs.
5. Returns:
   - ✓ ReturnDate: Date when the product was returned.
   - ✓ Territory_Id: ID of the territory from which the product was returned.
   - ✓ Product_Id: ID of the product that was returned.
   - ✓ ReturnQuantity: Quantity of the returned product.
6. Products:
   - ✓ Product_Id: Unique ID for each product.
   - ✓ Subcategory_Id: ID of the subcategory to which the product belongs.
   - ✓ ProductSKU: Stock Keeping Unit for the product.
   - ✓ ProductName: Name of the product.
   - ✓ ProductColor: Color of the product.
   - ✓ ProductSize: Size of the product.
   - ✓ ProductStyle: Style of the product.

- ✓ ProductCost: Cost of the product.
- ✓ ProductPrice: Selling price of the product.

7. Customers:
    - ✓ Customer_Id: Unique ID for each customer.
    - ✓ Prefix: Prefix for the customer's name (e.a. Mr.. Mrs.).
    - ✓ FirstName: First name of the customer.
    - ✓ LastName: Last name of the customer.
    - ✓ BirthDate: Birthdate of the customer.
    - ✓ MaritalStatus: Marital status of the customer.
    - ✓ Gender: Gender of the customer.
    - ✓ EmallAddress: Email address of the customer.
    - ✓ Annualincome: Annual income of the customer.
    - ✓ TotalChildren: Total number of children the customer has.
    - ✓ EducationLevel: Education level of the customer.
    - ✓ Occupation: Occupation of the customer.
    - ✓ HomeOwner: Indicates whether the customer owns a home (Y/N).
8. Categories:
    - ✓ Category_Id: Unique ID for each product category.
    - ✓ CategoryName: Name of the product category.

Data Analysis, SQL Queries and Results

Problem Statement 1

The organization intends to offer a special coupon for every customer's third purchase. You are to generate a table displaying the customer ID, the customer's full name in the format of first name followed by last name, and the ID of their third order. Please ensure the table follows the exact sequence of customer ID, full name, and third order ID. The final table should be arranged in ascending order by customer ID

Solution

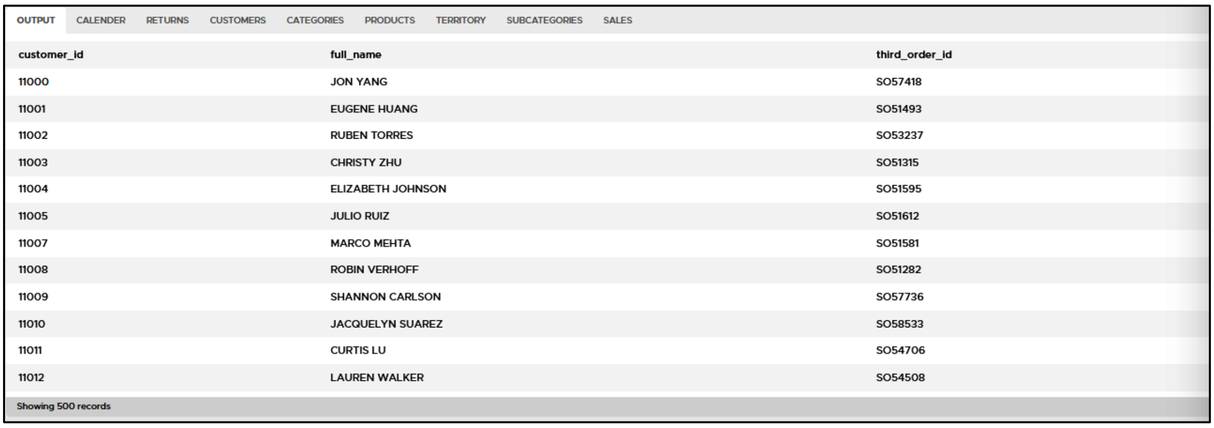
We create a table named 'CustomerOrders' that includes the customer ID, full name, and order number, with a sequence number assigned based on the order date. Filtering the table by the third order sequence, we get the required information.

Query

```
WITH CustomerOrders AS (
    SELECT
        c.customer_id,
        CONCAT(c.firstname, ' ', c.lastname) AS full_name,
        s.ordernumber,
        ROW_NUMBER() OVER(PARTITION BY c.customer_id ORDER BY s.orderdate) AS OrderSequence
    FROM
        customers c
    JOIN
        sales s ON c.customer_id::text = s.customer_id::text
)

SELECT
    co.customer_id,
    co.full_name,
    co.ordernumber AS Third_Order_ID
FROM
    CustomerOrders co
WHERE
    co.OrderSequence = 3
ORDER BY
    co.customer_id;
```

Output



customer_id	full_name	third_order_id
11000	JON YANG	SO57418
11001	EUGENE HUANG	SO51493
11002	RUBEN TORRES	SO53237
11003	CHRISTY ZHU	SO51315
11004	ELIZABETH JOHNSON	SO51595
11005	JULIO RUIZ	SO51612
11007	MARCO MEHTA	SO51581
11008	ROBIN VERHOFF	SO51282
11009	SHANNON CARLSON	SO57736
11010	JACQUELYN SUAREZ	SO58533
11011	CURTIS LU	SO54706
11012	LAUREN WALKER	SO54508

Showing 500 records

Problem Statement 2

Your task is to identify the 'blockbuster territory'-the territory with the highest sales volume-for each subcategory. Since this territory already demonstrates high sales, additional marketing efforts there might be redundant. You're expected to provide the territory ID that corresponds to the peak of sales for every subcategory. The desired output should list the subcategory ID, subcategory name, and the respective blockbuster territory ID, all sorted in ascending order by subcategory ID.

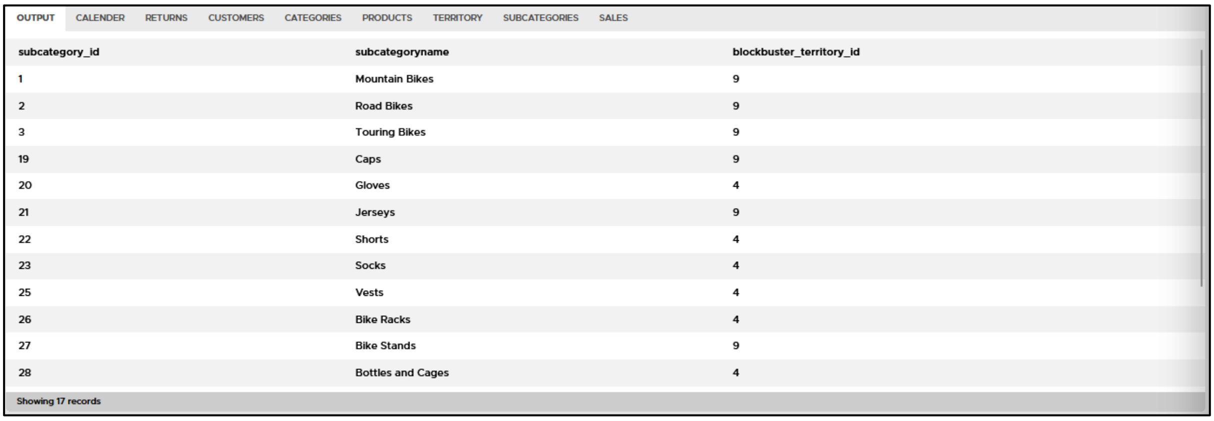
Solution

We create a table named 'SubcategorySales' to calculate the total sales volume for each subcategory in each territory. We rank the territories based on sales volume within each subcategory. Finally, we filter the table to get the blockbuster territory for each subcategory.

Query

```sql
WITH SubcategorySales AS (
    SELECT
        sc.subcategory_id,
        sc.subcategoryname,
        t.salesterritory_id,
        SUM(p.ProductPrice * s.OrderQuantity) AS TotalSalesVolume,
        ROW_NUMBER() OVER(PARTITION BY sc.subcategory_id ORDER BY
SUM(p.ProductPrice * s.OrderQuantity) DESC) AS TerritoryRank
    FROM
        Sales s
    JOIN
        Products p ON s.Product_ID = p.Product_Id
    JOIN
        Subcategories sc ON p.subcategory_id = sc.subcategory_id
    JOIN
        Territory t ON s.Territory_Id = t.salesterritory_id
    GROUP BY
        sc.subcategory_id, sc.subcategoryname, t.salesterritory_id
)
SELECT
    subcategory_id, subcategoryname,
    salesterritory_id AS Block_buster_Territory_ID
FROM SubcategorySales
WHERE TerritoryRank = 1
ORDER BY
    subcategory_id;
```

Output



subcategory_id	subcategoryname	blockbuster_territory_id
1	Mountain Bikes	9
2	Road Bikes	9
3	Touring Bikes	9
19	Caps	9
20	Gloves	4
21	Jerseys	9
22	Shorts	4
23	Socks	4
25	Vests	4
26	Bike Racks	4
27	Bike Stands	9
28	Bottles and Cages	4

Showing 17 records

Conclusion

In this project, we successfully addressed the organization's requirements by identifying customers eligible for a special coupon and determining the blockbuster territory for each product subcategory. These insights can guide targeted marketing efforts and improve overall business strategies for Adventure Works Cycles.

BY – DIVYA GATTUGARI