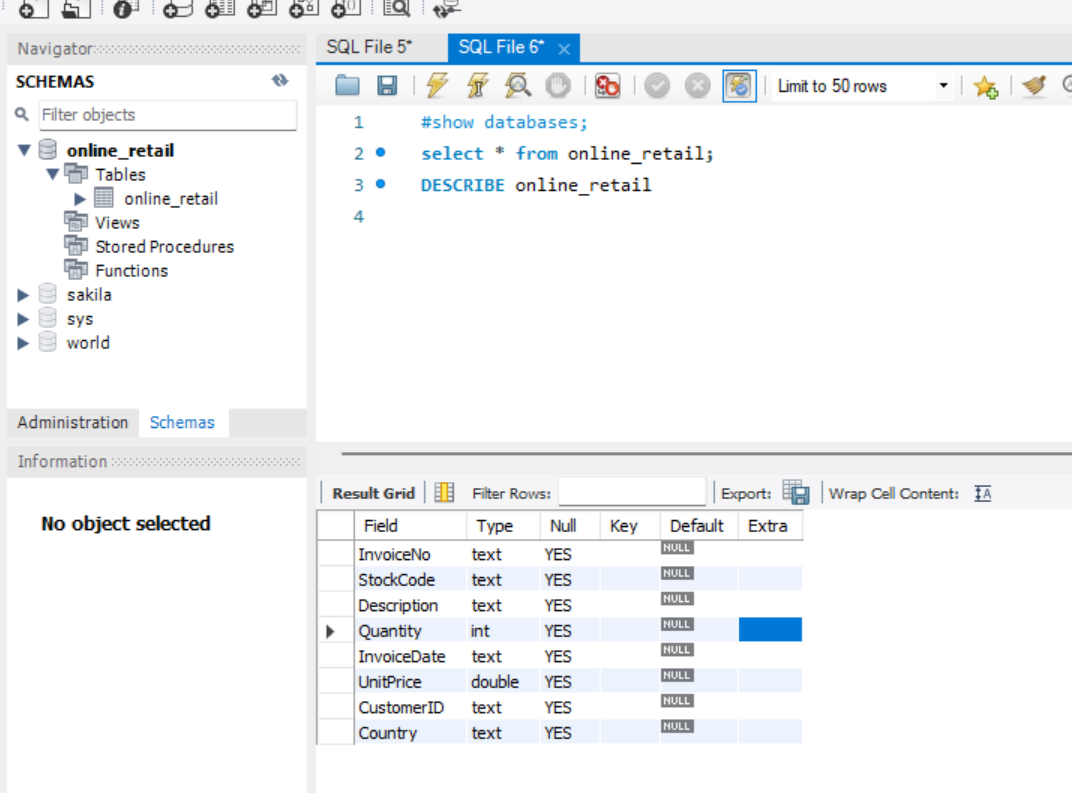
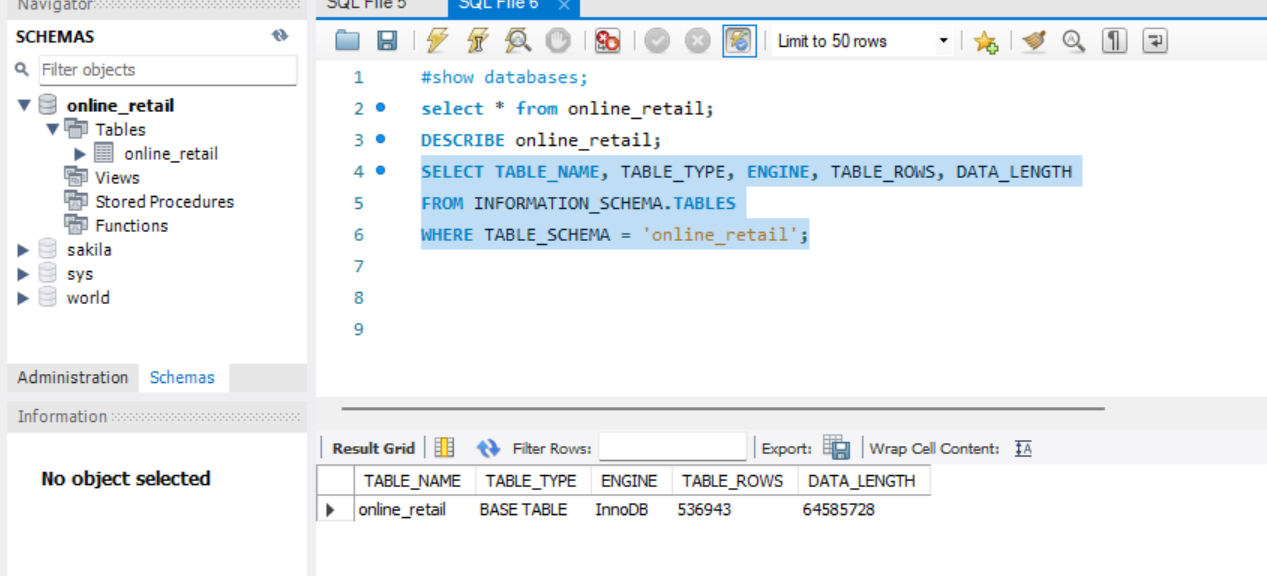
**SQL Project Idea: Use SQL queries to answer the following questions:**

Define meta data in mysql workbench or any other SQL tool

DESCRIBE online\_retail;





What is the distribution of order values across all customers in the dataset?

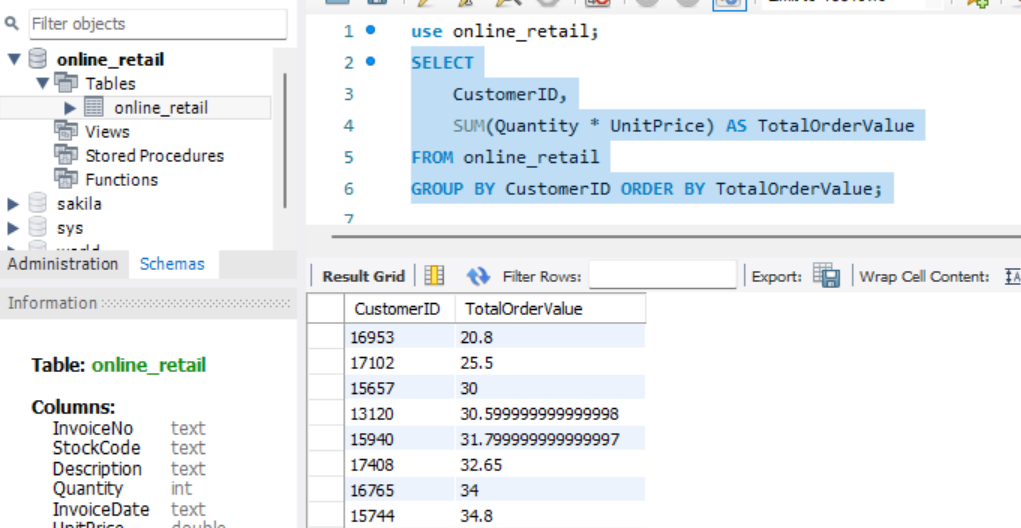
SELECT

CustomerID,

SUM(Quantity \* UnitPrice) AS TotalOrderValue

FROM online\_retail

GROUP BY CustomerID ORDER BY TotalOrderValue;



How many unique products has each customer purchased?

SELECT

CustomerID,

COUNT(DISTINCT StockCode) AS UniqueProductsPurchased

FROM

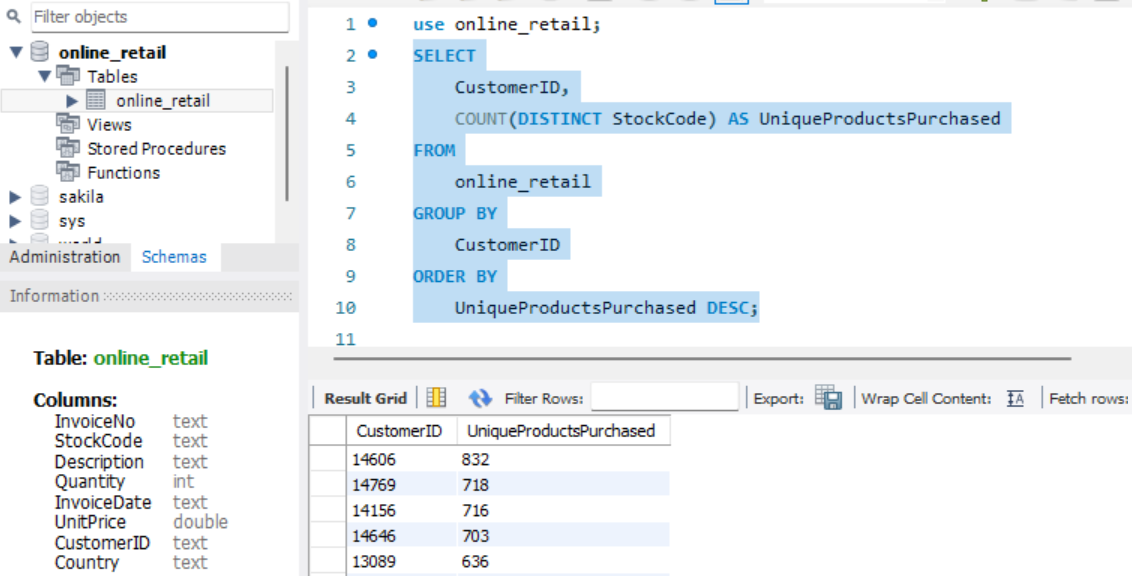
online\_retail

GROUP BY

CustomerID

ORDER BY

UniqueProductsPurchased DESC;



Which customers have only made a single purchase from the company?

SELECT

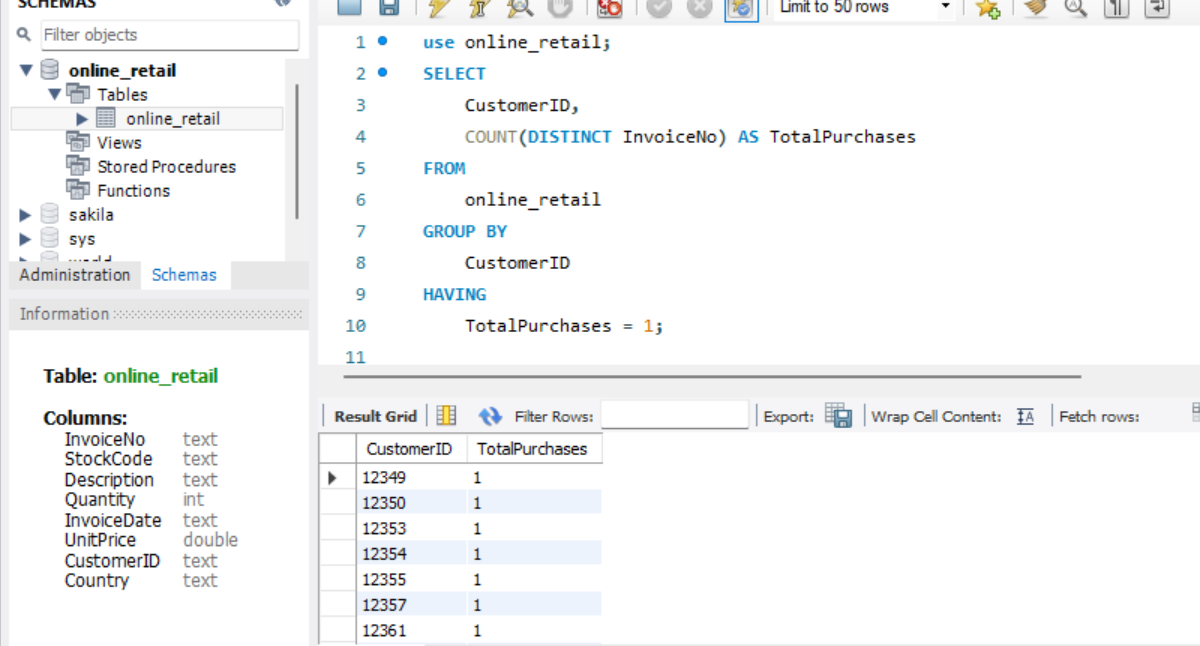
CustomerID,

COUNT(DISTINCT InvoiceNo) AS TotalPurchases

FROM

YourTableName

GROUP BY CustomerID HAVING TotalPurchases = 1;



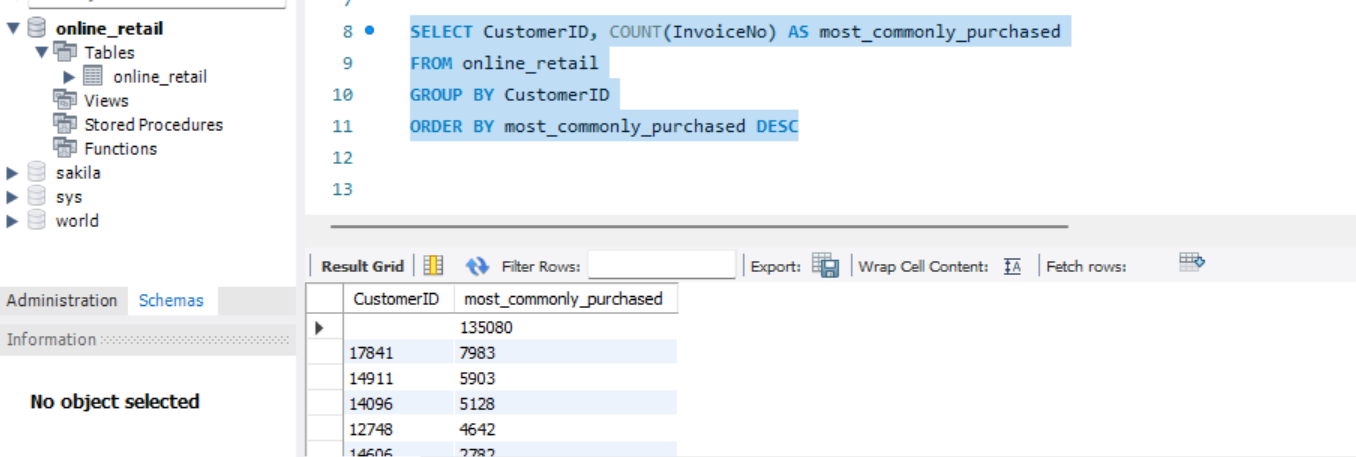
Which products are most commonly purchased together by customers in the dataset?

SELECT CustomerID, COUNT(InvoiceNo) AS most\_commonly\_purchased

FROM online\_retail

GROUP BY CustomerID

ORDER BY most\_commonly\_purchased DESC



1. **Customer Segmentation by Purchase Frequency**

Group customers into segments based on their purchase frequency, such as high, medium, and low frequency customers. This can help you identify your most loyal customers and those who need more attention.

SELECT CustomerID,

CASE

WHEN frequency >= 10 THEN 'High Frequency'

WHEN frequency >= 5 THEN 'Medium Frequency'

ELSE 'Low Frequency'

END AS frequency\_segment

FROM (

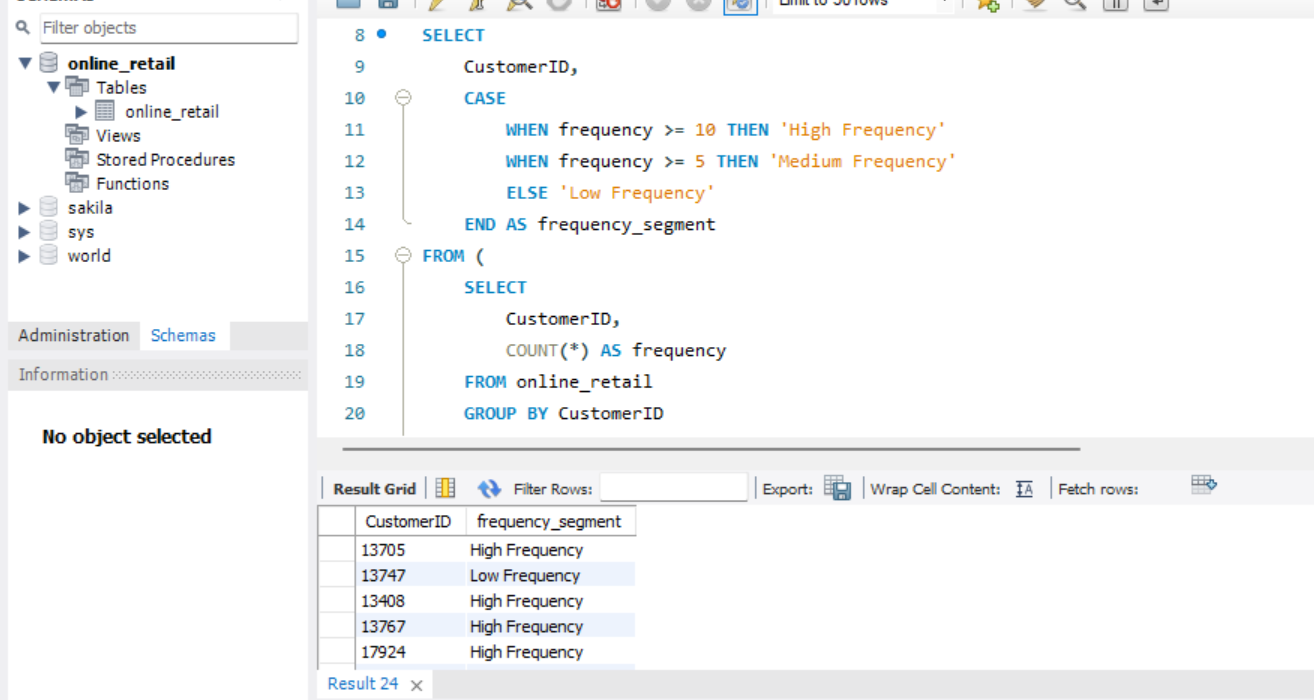
SELECT CustomerID,

COUNT(\*) AS frequency

FROM online\_retail

GROUP BY CustomerID

) AS customer\_frequency;



1. **Average Order Value by Country**

Calculate the average order value for each country to identify where your most valuable customers are located.

SELECT

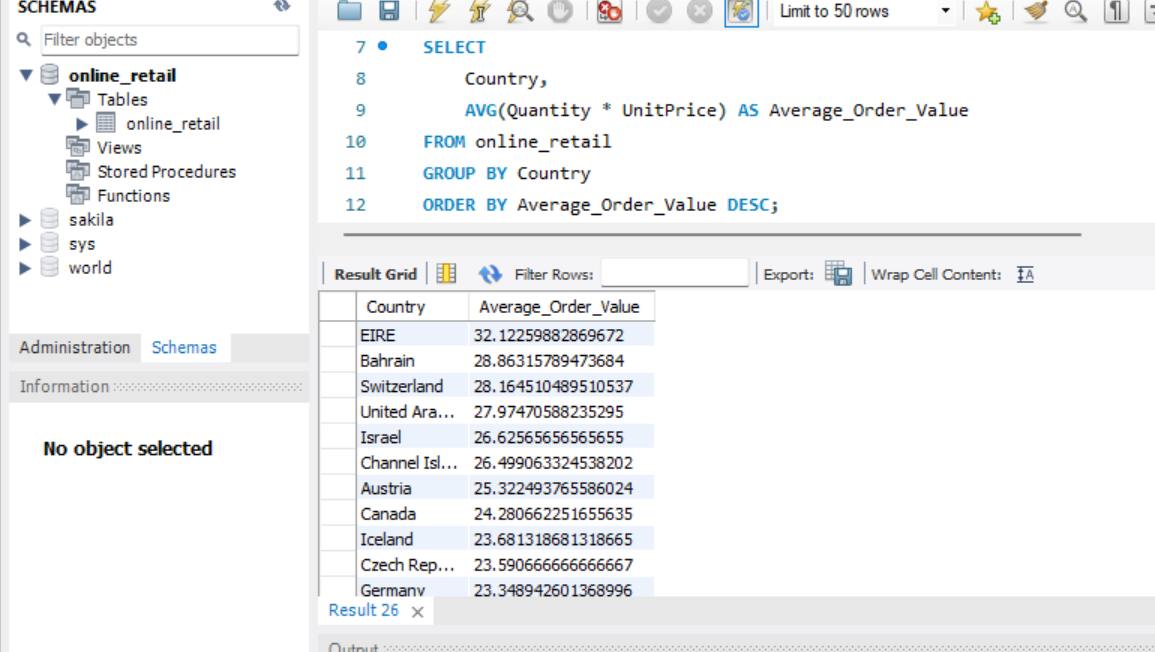
Country,

AVG(Quantity \* UnitPrice) AS Average\_Order\_Value

FROM online\_retail

GROUP BY Country

ORDER BY Average\_Order\_Value DESC;



1. **Customer Churn Analysis**

Identify customers who haven't made a purchase in a specific period (e.g., last 6 months) to assess churn.

SELECT

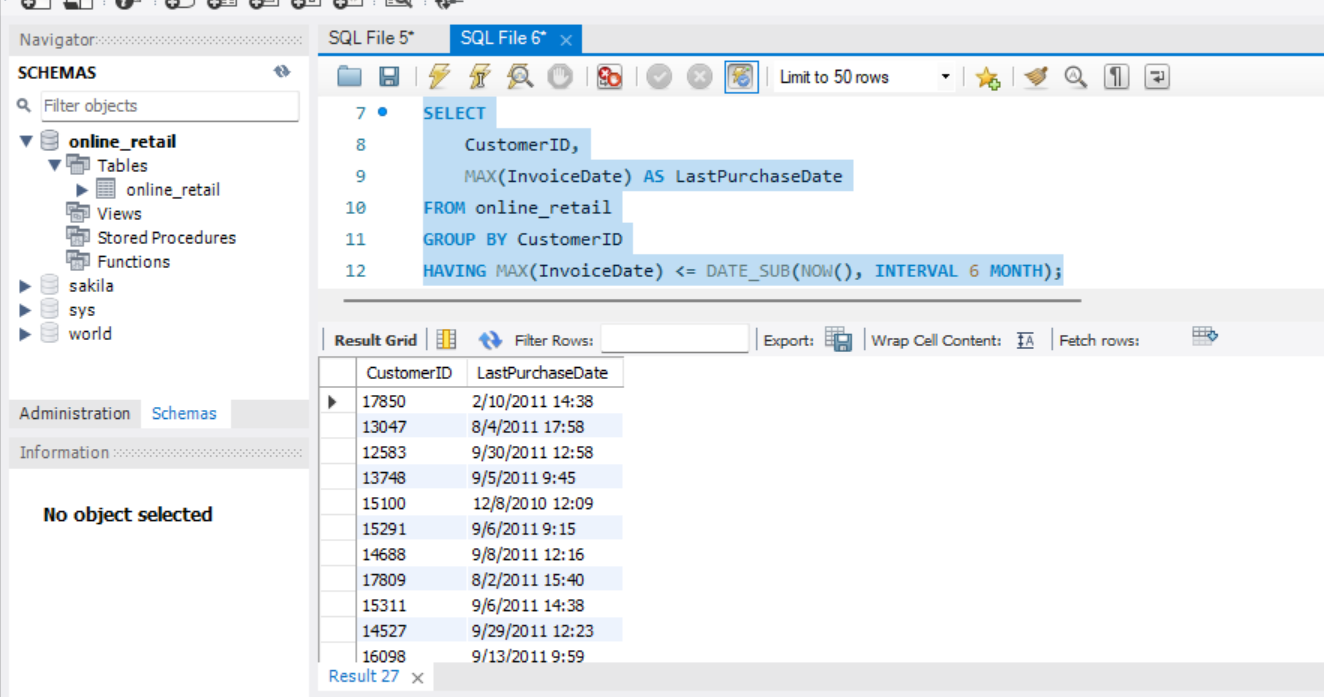
CustomerID,

MAX(InvoiceDate) AS LastPurchaseDate

FROM online\_retail

GROUP BY CustomerID

HAVING MAX(InvoiceDate) <= DATE\_SUB(NOW(), INTERVAL 6 MONTH);



1. **Product Affinity Analysis**

Determine which products are often purchased together by calculating the correlation between product purchases.

SELECT

p1.StockCode AS Product1,

p2.StockCode AS Product2,

COUNT(DISTINCT p1.InvoiceNo) AS CommonInvoices

FROM

online\_retail p1

JOIN

online\_retail p2 ON p1.InvoiceNo = p2.InvoiceNo AND p1.StockCode < p2.StockCode

WHERE

p1.InvoiceDate BETWEEN '12/1/2010 8:26' AND '12/1/2010 9:02'

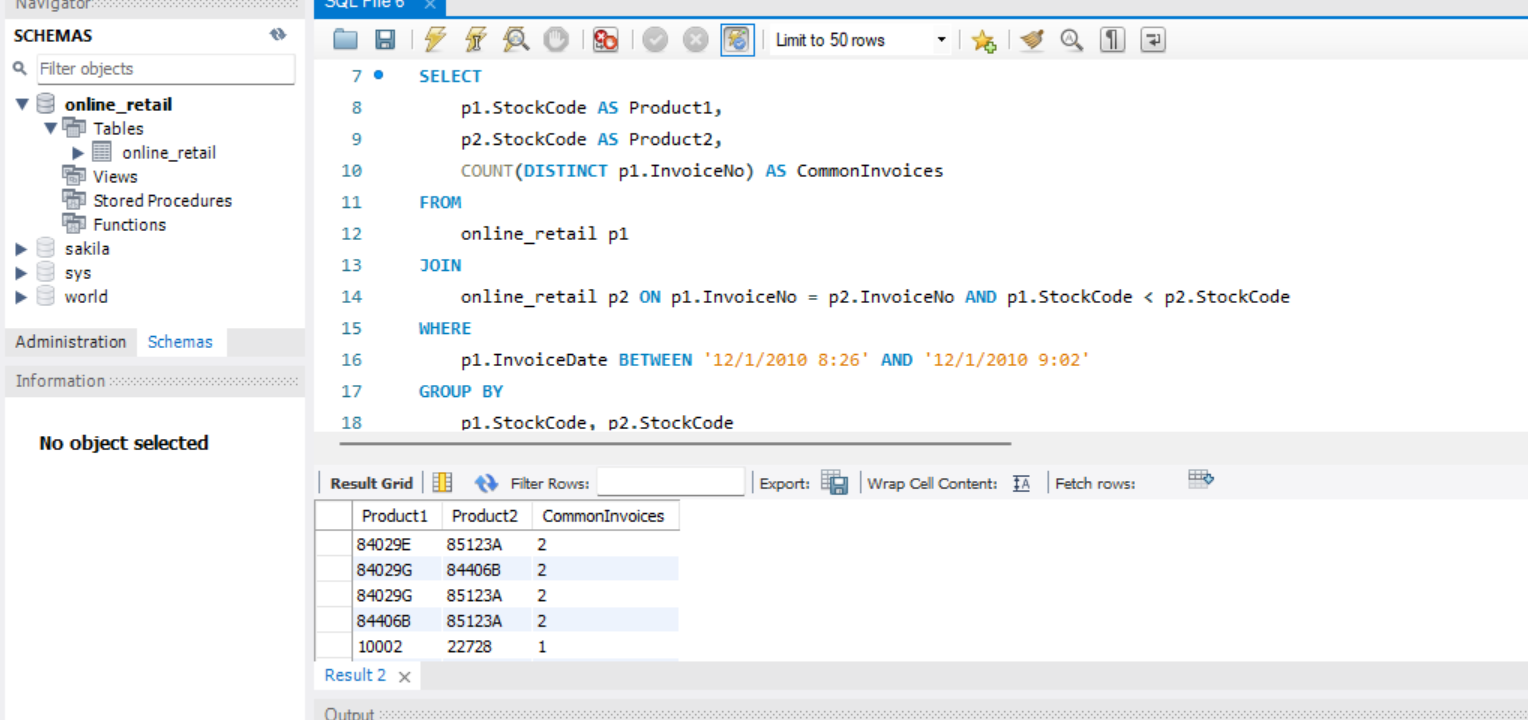
GROUP BY

p1.StockCode, p2.StockCode

ORDER BY

CommonInvoices DESC

LIMIT 0, 50;



1. **Time-based Analysis**

  Explore trends in customer behavior over time, such as monthly or quarterly sales patterns.

SELECT

InvoiceDate AS YearMonth,

SUM(Quantity) AS TotalQuantity,

SUM(UnitPrice \* Quantity) AS TotalRevenue

FROM

online\_retail

GROUP BY

YearMonth

ORDER BY

YearMonth;

