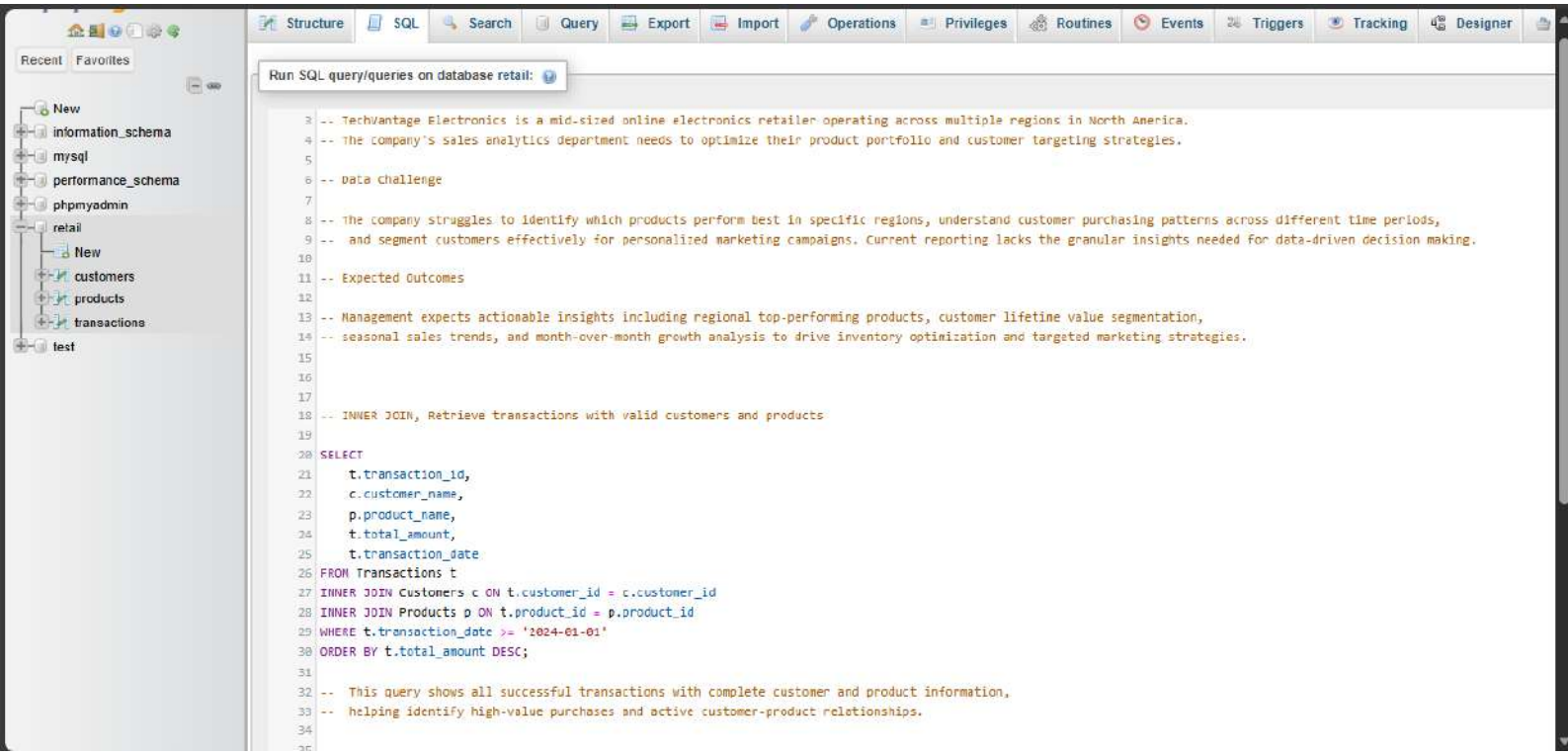
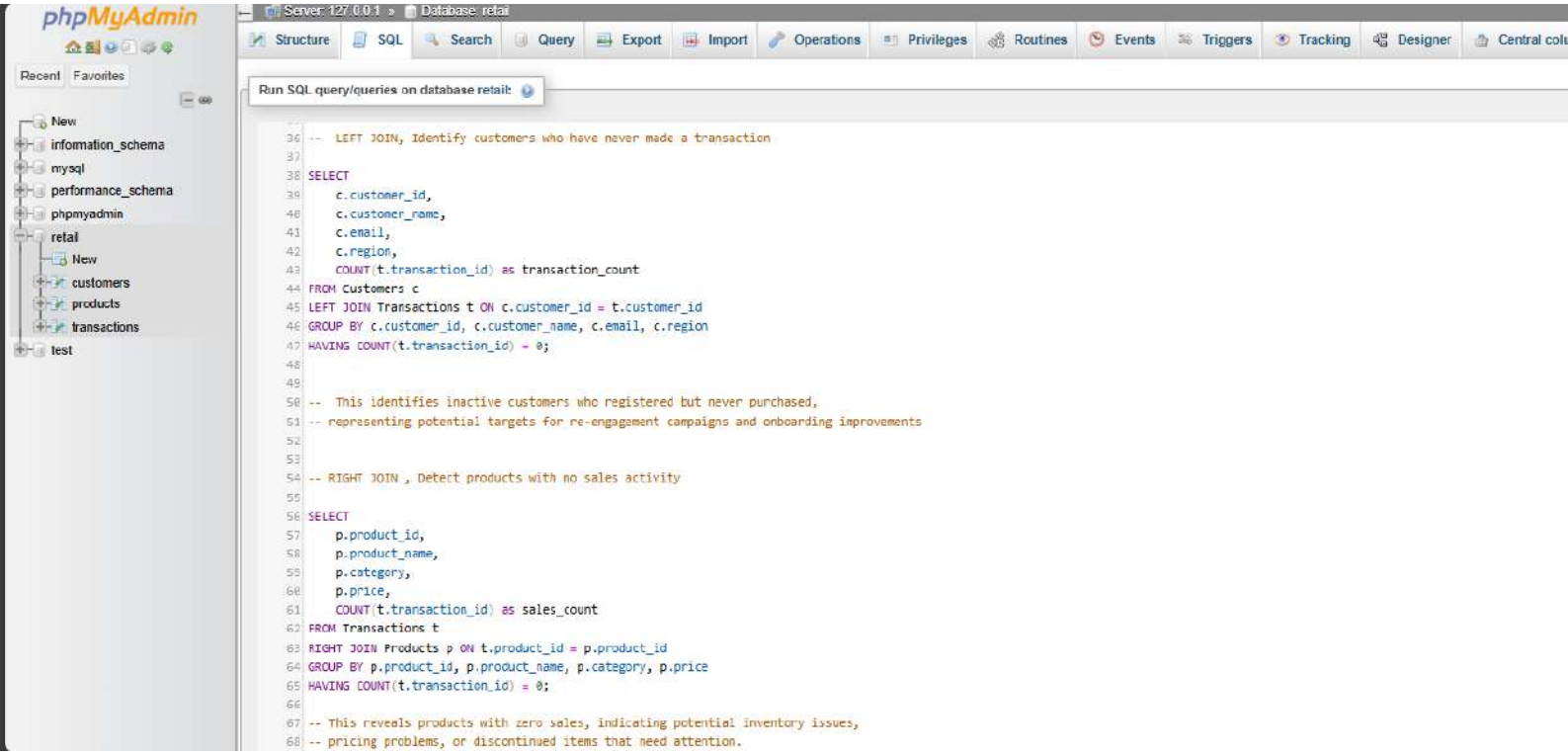
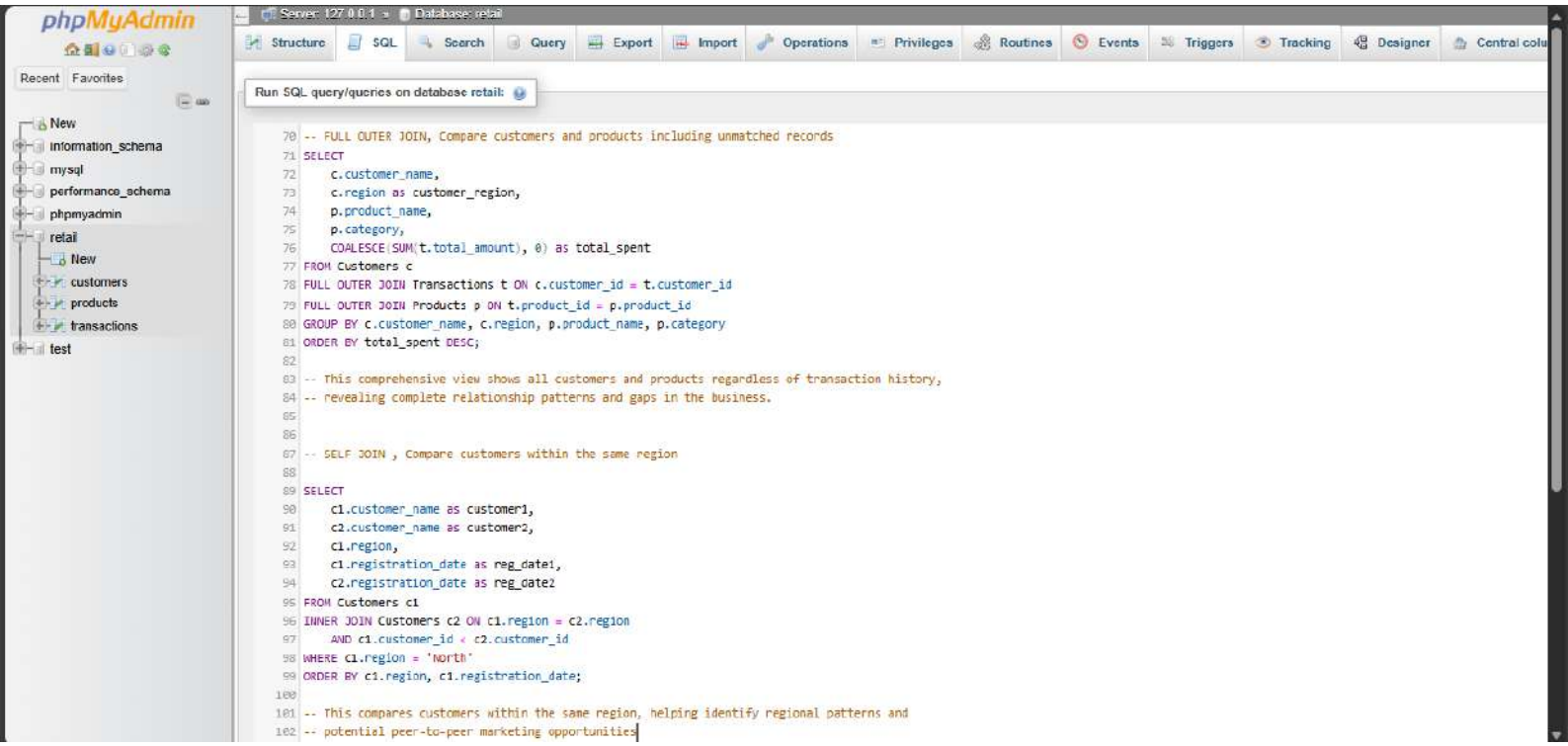

Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
retail
New
customers
products
transactions
test
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
Designer
Run SQL query/queries on database retail:
-- TechVantage Electronics is a mid-sized online electronics retailer operating across multiple regions in North America.
-- The company's sales analytics department needs to optimize their product portfolio and customer targeting strategies.
-- Data Challenge
-- The company struggles to identify which products perform best in specific regions, understand customer purchasing patterns across different time periods,
-- and segment customers effectively for personalized marketing campaigns. Current reporting lacks the granular insights needed for data-driven decision making.
-- Expected Outcomes
-- Management expects actionable insights including regional top-performing products, customer lifetime value segmentation,
-- seasonal sales trends, and month-over-month growth analysis to drive inventory optimization and targeted marketing strategies.
-- INNER JOIN, Retrieve transactions with valid customers and products
SELECT
t.transaction_id,
c.customer_name,
p.product_name,
t.total_amount,
t.transaction_date
FROM Transactions t
INNER JOIN Customers c ON t.customer_id = c.customer_id
INNER JOIN Products p ON t.product_id = p.product_id
WHERE t.transaction_date >= '2024-01-01'
ORDER BY t.total_amount DESC;
-- This query shows all successful transactions with complete customer and product information,
-- helping identify high-value purchases and active customer-product relationships.


phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
retail
New
customers
products
transactions
test
Server: 127.0.0.1 » Database: retail
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
Designer
Run SQL query/queries on database retail:
-- LEFT JOIN, Identify customers who have never made a transaction
SELECT
c.customer_id,
c.customer_name,
c.email,
c.region,
COUNT(t.transaction_id) as transaction_count
FROM Customers c
LEFT JOIN Transactions t ON c.customer_id = t.customer_id
GROUP BY c.customer_id, c.customer_name, c.email, c.region
HAVING COUNT(t.transaction_id) = 0;
-- This identifies inactive customers who registered but never purchased,
-- representing potential targets for re-engagement campaigns and onboarding improvements
-- RIGHT JOIN , Detect products with no sales activity
SELECT
p.product_id,
p.product_name,
p.category,
p.price,
COUNT(t.transaction_id) as sales_count
FROM Transactions t
RIGHT JOIN Products p ON t.product_id = p.product_id
GROUP BY p.product_id, p.product_name, p.category, p.price
HAVING COUNT(t.transaction_id) = 0;
-- This reveals products with zero sales, indicating potential inventory issues,
-- pricing problems, or discontinued items that need attention.


phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
retail
New
customers
products
transactions
test
Server: 127.0.0.1 » Database: retail
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
Designer
Run SQL query/queries on database retail:
-- FULL OUTER JOIN, Compare customers and products including unmatched records
SELECT
c.customer_name,
c.region as customer_region,
p.product_name,
p.category,
COALESCE(SUM(t.total_amount), 0) as total_spent
FROM Customers c
FULL OUTER JOIN Transactions t ON c.customer_id = t.customer_id
FULL OUTER JOIN Products p ON t.product_id = p.product_id
GROUP BY c.customer_name, c.region, p.product_name, p.category
ORDER BY total_spent DESC;
-- This comprehensive view shows all customers and products regardless of transaction history,
-- revealing complete relationship patterns and gaps in the business.
-- SELF JOIN , Compare customers within the same region
SELECT
c1.customer_name as customer1,
c2.customer_name as customer2,
c1.region,
c1.registration_date as reg_date1,
c2.registration_date as reg_date2
FROM Customers c1
INNER JOIN Customers c2 ON c1.region = c2.region
AND c1.customer_id < c2.customer_id
WHERE c1.region = 'North'
ORDER BY c1.region, c1.registration_date;
-- This compares customers within the same region, helping identify regional patterns and
-- potential peer-to-peer marketing opportunities