Information Technology Institute

Business Questions & Queries

Ahmed Hatem – Mariam Marcos – Mohamed Karam

Mostafa Mohamed – Yassa Rashad

1) Does order fulfillment speed (early vs. late deliveries) influence return rates among returned items?

- The query investigates the potential relationship between order fulfillment speed (early vs. late deliveries) and return rates specifically for returned items. By analyzing this data, businesses can gain valuable insights into several aspects.

```
WITH LateOrders AS (
  SELECT IsReturned,
        COUNT(*) AS LateOrdersCount
  FROM FactOrder
  WHERE TillShippingDuration >= (ordtoreqduration  -1)
  GROUP BY isreturned
),

EarlyOrders AS (
  SELECT IsReturned,
        COUNT(*) AS EarlyOrdersCount
  FROM FactOrder
  WHERE TillShippingDuration < (ordtoreqduration  -1)

  GROUP BY isreturned
)

SELECT l.isreturned,
      round(EarlyOrdersCount * 100.0 / (EarlyOrdersCount + LateOrdersCount),2) AS
EarlyReturnPercentage,
      round(LateOrdersCount * 100.0 / (EarlyOrdersCount + LateOrdersCount),2) AS
LateReturnPercentage
FROM LateOrders l
INNER JOIN EarlyOrders e ON l.isreturned = e.isreturned
WHERE l.isreturned = '1';
```

Before Indexes:

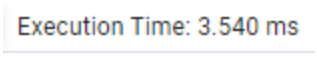
After Indexes:

2) How do Duels (Managers and their Employees (direct reports)) compare in terms of total profit generated and profit per day?

- This performance evaluation query contrasts Duel Managers' performance with their direct reports, focusing on total profit and profit per day. It's geared towards identifying top performers and optimizing resource allocation through individual and team performance analysis.

```
SELECT
    d.firstname || ' ' || d.lastname as EmpName,
    e.firstname || ' ' || e.lastname as ManagerName,
    COALESCE(TO_DATE(CAST(d.effectivetodatekey AS VARCHAR), 'YYYYMMDD'), '2024-12-30')
-
    TO_DATE(CAST(d.effectivefromdatekey AS VARCHAR), 'YYYYMMDD') AS period,
    SUM(profit) as TotalProfit,
    ROUND(SUM(profit) / (COALESCE(TO_DATE(CAST(d.effectivetodatekey AS VARCHAR),
'YYYYMMDD'), '2024-12-30') - TO_DATE(CAST(d.effectivefromdatekey AS VARCHAR),
'YYYYMMDD')
    ), 2) AS profitperday
FROM
    dimemployee d
INNER JOIN
    dimemployee e ON e.employeekey = d.managerkey
LEFT JOIN
    factorder f ON d.employeekey = f.employeekey
GROUP BY
    d.firstname, d.lastname, e.firstname, e.lastname, d.effectivetodatekey,
d.effectivefromdatekey
ORDER BY
    profitperday;
```

Before Indexes:



After Indexes:

3) In each city we operate in, which product categories are the most profitable?

- This query investigates the top-selling product categories within each city for the company. By ranking categories based on their total profit contribution within each city.

```
WITH total_profit AS(
    SELECT dimcustomer.city city,
    dimproduct.category category,
    SUM(factorder.totalprice) AS total_profit
    FROM factorder
    JOIN dimcustomer
    ON dimcustomer.customerkey = factorder.customerkey
    JOIN dimproduct
    ON dimproduct.productkey = factorder.productkey
    GROUP BY 1, 2
    ORDER BY total_profit DESC
    )

SELECT city, category,
RANK() OVER(PARTITION BY city ORDER BY total_profit DESC) AS top_categories
FROM total_profit
ORDER BY 1;
```

Before Indexes:

After Indexes:

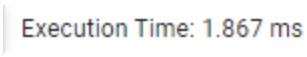
4) What are the preferred payment methods and sales channels?

- This query provides valuable insights into the most prevalent sales channels and payment methods used by your customers. By analyzing the grouped sales data, we can gain a better understanding of customer behavior and optimize our company's sales strategy accordingly.

```
SELECT COALESCE(PaymentMethod, 'Total') AS PaymentMethod,
       COALESCE(SalesChannel, 'Total') AS SalesChannel,
       SUM(TotalPrice) AS TotalSales
FROM FactOrder f
LEFT JOIN DimPayMethodSaleChannel d ON f.PayChanKey = d.PayChanKey
GROUP BY CUBE(PaymentMethod, SalesChannel)
ORDER BY TotalSales DESC;
```

Before Indexes:

After Indexes:



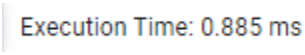
5) In each city we operate in, what are the top most frequently returned products?

- This query aims to identify the most frequently returned products within each city. By analyzing return data at the city and product level.

```
WITH top_returns AS (
   SELECT dimcustomer.city AS city,
          dimproduct.productname AS productname,
```

```
        COUNT(*) AS number_of_returns
    FROM factorder
    JOIN dimcustomer ON dimcustomer.customerkey = factorder.customerkey
    JOIN dimproduct ON dimproduct.productkey = factorder.productkey
    WHERE isreturned = '1'
    GROUP BY 1, 2
    ORDER BY 1, 3 DESC
)
SELECT city, productname, top_returned_products
FROM (
    SELECT city, productname, number_of_returns,
        ROW_NUMBER() OVER(PARTITION BY city ORDER BY number_of_returns DESC) AS
top_returned_products
    FROM top_returns
) AS ranked_returns
WHERE top_returned_products <= 3;
```

Before Indexes:



After Indexes:

6) Who are the top 5 customers, based on the total amount spent, in every city where we conduct business?

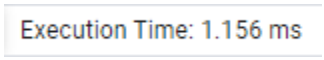
- This query identifies the top-spending customers (by total amount paid) within each city. By analyzing customer spending data at the city level.

```
WITH cte AS(
    SELECT dimcustomer.city city,
    dimcustomer.companyname customer,
    SUM(factorder.totalprice) AS amount_paid
    FROM factorder
    JOIN dimcustomer
    ON dimcustomer.customerkey = factorder.customerkey
```

```
    GROUP BY 1, 2
    ORDER BY amount_paid DESC
    )
SELECT city,customer, amount_paid, top_customers
FROM(SELECT city, customer, amount_paid,
    DENSE_RANK() OVER(PARTITION BY city ORDER BY amount_paid DESC) AS top_customers
    FROM cte) AS ranked_customers
WHERE top_customers <= 5;
```

Before Indexes:

After Indexes:

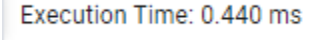


7)Which suppliers are the most valuable in terms of product variety(count)?

- This query is designed to identify the suppliers that provide the most products to the company. By ranking suppliers based on the number of products they offer.

```
with
cte as (
select s.supplierid,s.companyname,count(productkey) as Num_Of_Products
from dimsupplier s left join dimproduct p ON s.supplierkey = p.supplierkey
group by s.supplierkey,s.supplierid,s.contactname
)
select *,rank() over(order by Num_Of_Products desc) as Rank
from cte;
```

Before Indexes:



After Indexes:

Execution Time: 0.390 ms

8) Which suppliers are the most valuable in terms of overall profitability?

- This query aims to identify the most profitable suppliers for the company. By ranking suppliers based on their total profit contribution.

```
WITH SupplierProfit AS (
    SELECT
        p.SupplierKey,
        s.CompanyName AS SupplierName,
        SUM(o.Profit) AS TotalProfit,
        RANK() OVER (ORDER BY SUM(o.Profit) DESC) AS ProfitRank
    FROM
        DimProduct p
    INNER JOIN
        FactOrder o ON p.ProductKey = o.ProductKey
    INNER JOIN
        DimSupplier s ON p.SupplierKey = s.SupplierKey
    GROUP BY
        p.SupplierKey, s.CompanyName
)
SELECT
    SupplierKey,
    SupplierName,
    TotalProfit,
    ProfitRank
FROM
    SupplierProfit
ORDER BY
    ProfitRank;
```

Before Indexes:

Execution Time: 2.443 ms

After Indexes:

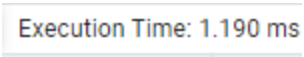
Execution Time: 1.693 ms

9) Which products are the most profitable during holidays based on quantity sold, total sales price, and total profit?

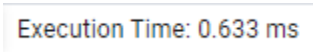
- Analyzing product performance during holidays, this query delves into sales volume, total sales price, and profit, guiding inventory management, product promotion, and development strategies.

```
SELECT ProductName, SUM(Quantity) AS Quan_Titiy, SUM(TotalPrice) AS Total_Price,
SUM(Profit) AS Pro_Fit
FROM FactOrder f
LEFT JOIN DimProduct p ON f.ProductKey = p.ProductKey
LEFT JOIN DimDate d ON f.OrderDateKey = d.DateKey
WHERE IsHoliday = '1'
GROUP BY ProductName, IsHoliday
ORDER BY pro_fit DESC;
```

Before Indexes:



After Indexes:

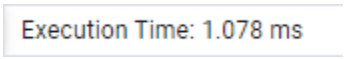


10) Which suppliers have the highest number of returned products?

- Examining supplier performance regarding returned products, this query analyzes the count of returns per supplier, informing supplier quality and relationship management, as well as warranty claims procedures.

```
SELECT
ds.SupplierKey,
ds.CompanyName AS SupplierName,
COUNT(fo.ReturnDate) AS ReturnedProductCount
FROM
DimSupplier ds
JOIN
DimProduct dp ON ds.SupplierKey = dp.SupplierKey
JOIN
FactOrder fo ON dp.ProductKey = fo.ProductKey
WHERE
fo.isReturned = '1'
GROUP BY
ds.SupplierKey, ds.CompanyName
ORDER BY
ReturnedProductCount DESC ;
```

Before Indexes:



After Indexes: