Wei Cheng Wu, Eric Rodriquez, Alejandro Bermudez

Slide 4: Trend Quantities by Year

```
SELECT YEAR(s.Order_Date) AS year, SUM(s.Quantity) AS total_quantity FROM
proj_sales s GROUP BY YEAR(s.Order_Date) ORDER BY year;
```

Slide 5: Trend in Last Order Dates

```
SELECT FORMAT(Order_Date, 'yyyy-MM') AS Year_Month, COUNT(DISTINCT
CustomerKey) AS Number_of_Customers FROM proj_sales WHERE Order_Date IS
NOT NULL GROUP BY FORMAT(Order_Date, 'yyyy-MM') ORDER BY Year_Month;
```

Slide 6: Total Revenue by Customer Type and Year

```
WITH FirstOrderYear AS ( SELECT CustomerKey, MIN(YEAR(Order_Date)) AS
first_purchase_year FROM proj_sales GROUP BY CustomerKey ) SELECT
YEAR(s.Order_Date) AS year, CASE WHEN f.first_purchase_year =
YEAR(s.Order_Date) THEN 'New Customer' ELSE 'Existing Customer' END AS
customer_type, SUM(s.Quantity * p.Unit_Price_USD) AS total_revenue FROM
proj_sales s JOIN FirstOrderYear f ON s.CustomerKey = f.CustomerKey JOIN
proj_products p ON s.ProductKey = p.ProductKey GROUP BY YEAR(s.Order_Date),
CASE WHEN f.first_purchase_year = YEAR(s.Order_Date) THEN 'New Customer'
ELSE 'Existing Customer' END ORDER BY year, customer_type;
```

Slide 7: Sales Trend vs Cost Trend by Region

```
SELECT st.State AS region, SUM(s.Quantity * p.Unit_Price_USD) AS total_revenue,
SUM(s.Quantity * p.Unit_Cost_USD) AS total_cost FROM proj_sales s JOIN
proj_stores st ON s.StoreKey = st.StoreKey JOIN proj_products p ON s.ProductKey =
p.ProductKey GROUP BY st.State ORDER BY total_revenue DESC;
```

Slide 8: Sales by month for 2019

```
SELECT YEAR(s.Order_Date) AS year, MONTH(s.Order_Date) AS month,
SUM(s.Quantity * p.Unit_Price_USD) AS total_sales FROM proj_sales s JOIN
proj_products p ON s.ProductKey = p.ProductKey WHERE YEAR(s.Order_Date) =
2019 -- Filter for the year 2019 GROUP BY YEAR(s.Order_Date),
MONTH(s.Order_Date) ORDER BY month;
```

Slide 8: Sales by month for 2020

```
SELECT YEAR(s.Order_Date) AS year, MONTH(s.Order_Date) AS month,
SUM(s.Quantity * p.Unit_Price_USD) AS total_sales FROM proj_sales s JOIN
proj_products p ON s.ProductKey = p.ProductKey WHERE YEAR(s.Order_Date) =
2020 -- Filter for the year 2020 GROUP BY YEAR(s.Order_Date),
MONTH(s.Order_Date) ORDER BY month;
```

Slide 9: Late Delivery Rate by Product

```
SELECT p.Product_Name AS product_name, SUM(s.Quantity) AS
total_ordered_quantity, COUNT(s.Order_Number) AS total_orders, SUM(CASE WHEN
DATEDIFF(day, s.Order_Date, s.Delivery_Date) > 3 THEN 1 ELSE 0 END) AS
delayed_orders, (SUM(CASE WHEN DATEDIFF(day, s.Order_Date, s.Delivery_Date) >
3 THEN 1 ELSE 0 END) * 100.0 / COUNT(s.Order_Number)) AS late_delivery_rate
FROM proj_sales s JOIN proj_products p ON s.ProductKey = p.ProductKey GROUP BY
p.Product_Name HAVING (SUM(CASE WHEN DATEDIFF(day, s.Order_Date,
s.Delivery_Date) > 3 THEN 1 ELSE 0 END) * 100.0 / COUNT(s.Order_Number)) > 50 --
Late delivery rate > 50% ORDER BY late_delivery_rate DESC;
```

Slide 10: Order status Distribution

```
SELECT CASE WHEN s.Delivery_Date IS NOT NULL THEN 'Fulfilled' ELSE 'Unfulfilled'
END AS order_status, COUNT(*) AS total_orders FROM proj_sales s GROUP BY
CASE WHEN s.Delivery_Date IS NOT NULL THEN 'Fulfilled' ELSE 'Unfulfilled' END;
```