—---

- Section 1: Customer and Order Overview

-- Questions Q1 to Q5

-- Focus: Total order count, total revenue, unique customer count, order frequency, and most recent purchase activity

-- Note: Establishes foundational KPIs to support downstream analysis in product performance and customer retention

—--

```
-- Q1: Total Revenue from Delivered Orders
-- Objective: Calculate total revenue from all successfully delivered
orders.

SELECT
    SUM(sale_price) AS total_revenue
FROM orders
WHERE status = 'Delivered';

--------------------------------------------------
```

```
-- Q2: Revenue by Product Category
-- Objective: Determine which product categories generate the most
revenue.

SELECT
    p.product_category,
    SUM(o.sale_price) AS total_revenue
FROM orders o
JOIN product p
ON o.productid = p.productid
WHERE o.status = 'Delivered'
GROUP BY p.product_category
ORDER BY total_revenue DESC;

--------------------------------------------------
```

```sql
-- Q3: Top 10 Best-Selling Products by Sales Value
-- Objective: Identify the top 10 products that brought in the highest
revenue.

SELECT TOP 10
    p.product_name,
    SUM(o.sale_price) AS total_sales
FROM orders o
JOIN product p ON
o.productid = p.productid
WHERE o.status = 'Delivered'
GROUP BY p.product_name
ORDER BY total_sales DESC;

--------------------------------------------------

-- Q4: Total Orders and Revenue by Zone
-- Objective: Compare order volume and revenue across zones.

SELECT
    o.zone,
    COUNT(o.orderid) AS total_orders,
    SUM(o.sale_price) AS total_revenue
FROM orders o
WHERE o.status = 'Delivered'
GROUP BY o.zone
ORDER BY total_revenue DESC;

--------------------------------------------------

-- Q5: Month-over-Month Revenue Trend (last 6 months)
-- Objective: Identify recent revenue trends and detect any seasonality.

SELECT
    FORMAT(orderdate, 'yyyy-MM') AS month_year,
    SUM(sale_price) AS monthly_revenue
FROM orders
```

```sql
WHERE status = 'Delivered'
AND orderdate >= DATEADD(MONTH, -6, GETDATE())
GROUP BY FORMAT(orderdate, 'yyyy-MM')
ORDER BY month_year;

--Here as the latest date in the dataset is 31st DEC 2020
--so in general last 6 months from the current date will not give us the exact output, so
--to modify the question little bit , identify the last 6 month revenue trend from the latest year

--"Since the dataset ends on 31st Dec 2020, using GETDATE() wouldn't give relevant results.
--Instead, I dynamically anchored the last 6 months based on the MAX(orderdate) in the dataset.
--I used > instead of >= to include exactly 6 full months, from July to December 2020."

SELECT
    FORMAT(orderdate, 'yyyy-MM') AS month_year,
    SUM(sale_price) AS monthly_revenue
FROM orders
WHERE status = 'Delivered'
AND orderdate > DATEADD(MONTH, -6, (select max(orderdate) from orders))
GROUP BY FORMAT(orderdate, 'yyyy-MM')
ORDER BY month_year;

-- -------------------------------------------------------
-- Section 2: Customer Demographics and Category Analysis
-- Questions Q6 to Q10
-- Focus: Unique customer count, age group trends, gender patterns,
--        zone-wise demand, and average order quantity by category
-- Note: All queries consider only 'Delivered' orders for cleaner insights
-- -------------------------------------------------------


-- Q6: Number of Unique Customers by Category
-- Objective: Understand customer engagement across product categories
-- Note: Filtered only 'Delivered' orders to ensure analysis is based on successful transactions.
```

```
SELECT
    p.Product_Category,
    COUNT(DISTINCT o.CustomerID) AS unique_customers
FROM Product p
JOIN Orders o
  ON p.ProductID = o.ProductID
WHERE o.Status = 'Delivered'
GROUP BY p.Product_Category
ORDER BY unique_customers DESC;


-- Q7: Customer Age Group Distribution by Category
-- Objective: Identify age segments active in different product categories
-- Note: Focused only on delivered orders to reflect actual customer
behavior.

SELECT
    p.Product_Category,
    CASE
        WHEN o.Customer_Age < 20 THEN 'Below 20'
        WHEN o.Customer_Age BETWEEN 20 AND 29 THEN '20-29'
        WHEN o.Customer_Age BETWEEN 30 AND 39 THEN '30-39'
        WHEN o.Customer_Age BETWEEN 40 AND 49 THEN '40-49'
        ELSE '50+'
    END AS age_group,
    COUNT(DISTINCT o.CustomerID) AS customer_count
FROM Orders o
JOIN Product p
  ON o.ProductID = p.ProductID
WHERE o.Status = 'Delivered'
GROUP BY p.Product_Category,
         CASE
            WHEN o.Customer_Age < 20 THEN 'Below 20'
            WHEN o.Customer_Age BETWEEN 20 AND 29 THEN '20-29'
            WHEN o.Customer_Age BETWEEN 30 AND 39 THEN '30-39'
            WHEN o.Customer_Age BETWEEN 40 AND 49 THEN '40-49'
            ELSE '50+'
         END
ORDER BY p.Product_Category, age_group;
```

```sql
-- Q8: Product Category Purchase by Gender
-- Objective: Analyze gender-based purchase behavior per category
-- Note: Only delivered orders included to avoid distortion by returns or
cancellations.

SELECT
    p.Product_Category,
    o.Customer_Gender,
    COUNT(*) AS total_orders
FROM Orders o
JOIN Product p
  ON o.ProductID = p.ProductID
WHERE o.Status = 'Delivered'
GROUP BY p.Product_Category, o.Customer_Gender
ORDER BY p.Product_Category;


-- Q9: Customer Zone-wise Category Demand
-- Objective: Understand regional trends in category purchases
-- Note: Only completed deliveries are considered to reflect true
geographic demand.

SELECT
    o.Zone,
    p.Product_Category,
    COUNT(*) AS total_orders
FROM Orders o
JOIN Product p
  ON o.ProductID = p.ProductID
WHERE o.Status = 'Delivered'
GROUP BY o.Zone, p.Product_Category
ORDER BY o.Zone, total_orders DESC;


-- Q10: Average Order Quantity per Category
-- Objective: Determine how much customers typically order per category
-- Note: Analyzed only delivered orders for an accurate average.
```

```sql
SELECT
    p.Product_Category,
    ROUND(AVG(o.Order_Quantity), 2) AS avg_order_quantity
FROM Orders o
JOIN Product p
  ON o.ProductID = p.ProductID
GROUP BY p.Product_Category
ORDER BY avg_order_quantity DESC;

-- --------------------------------------------------------
-- Section 3: Product Returns and Rating Analysis
-- Questions Q11 to Q15
-- Focus: Return rates by category and product, top return reasons,
--         fulfillment success, and the relationship between ratings and
returns
-- Note: Most queries focus on 'Returned' and 'Delivered' statuses.
--        Output is formatted using CAST/ROUND for clean readability.
-- --------------------------------------------------------

--Q11. Return Rate by Product Category
--Identify product categories with high return rates to investigate
product quality or expectation gaps.

select p.product_category,
cast(Round(count(case when od.status='returned' then 1
end)*100.0/count(*), 2) as decimal(5,2))
as return_rate
from orders od
join product p
on od.productid=p.productid
group by p.product_category
order by return_rate;

-- Objective: Identify product categories with high return rates
-- Note: Return rate = (Returned Orders / Total Orders) * 100
--Using CTEs for readability

with return_cte as(
select p.product_category, count(od.status) as return_orders
from orders od join product p
on od.productid=p.productid
where od.status='returned'
group by p.Product_Category),
count_cte as(select p.product_category,count(od.status) as order_count
from orders od
join product p on p.productid=od.productid
group by p.Product_Category
)
select count_cte.product_category,
CAST( (return_orders)*100.0/order_count as Decimal (5,2)) as Return_rate
from return_cte join count_cte
on return_cte.Product_Category=count_cte.product_Category
order by return_rate desc;

--Comment
-- Used CTEs to split return and total orders for better readability
-- Applied CAST to format return rate percentage to two decimal places
-- Compared return orders against total orders for each category

--Q12. Top 5 Products with Highest Return Rate
--Spot individual products being returned frequently to review listing or
QA.

select top 5 p.product_name,
CAST(COUNT(CASE WHEN od.Status = 'Returned' THEN 1 END) * 100.0 / COUNT(*)
AS DECIMAL(5,2))
AS return_rate
from orders od join product p
on p.productid=od.productid
group by p.product_name
order by return_rate desc;

--Q13. Top 5 Return Reasons
--Understand the most common reasons customers return products.
select top 5 Reason, count(*) as return_count
from orders
where status='Returned'
group by Reason
order by return_count desc;
```

```sql
--Using Coalesce to Handle Null values(avoid reason as Null)

select top 5 Coalesce (Reason,'Not mentioned') as Return_reason, count(*)
as return_count
from orders
where status='Returned'
group by Coalesce (Reason,'Not mentioned')
order by return_count desc;

--Q14. Compare Returned vs Delivered Count per Category
--Compare fulfillment success rate by category.

select p.product_category,
count(case when od.status='delivered' then 1 end) as
delivery_category_count,
count(case when od.status='returned' then 1 end) as return_category_count
from orders od join product p
on od.productid=p.productid
group by p.product_category;

--Q15. Rating vs Return Rate Relationship
--Check if lower ratings are linked to more returns for product
improvement.

select rating,
case
when rating < 3 then 'below avg'
when rating =3 then 'avg'
else 'above_avg'
end as rating_cat,
CAST(count(case when status='returned' then 1 end)*100.0/count(*) as
decimal(5,2))as return_rate
from orders
group by rating,
case
when rating < 3 then 'below avg'
when rating =3 then 'avg'
else 'above_avg'
end
order by rating desc;
```

```sql
--other method without categorizing

select rating,
CAST(count(case when status='returned' then 1 end)*100.0/count(*) as
decimal(5,2)) return_rate
from orders
group by rating
order by return_rate desc;

-- ---------------------------------------------------------------------
-- Section 4: Delivery and Operational Analysis (Q16 to Q20)
-- ---------------------------------------------------------------------



-- Q16: Avg Delivery Time by Delivery Type
-- Objective: Identify average delivery timelines to assess SLA adherence

SELECT
    delivery_type,
    AVG(DATEDIFF(DAY, orderdate, delivery_date)) AS
avg_deliverytime_in_days
FROM orders
GROUP BY delivery_type
ORDER BY avg_deliverytime_in_days DESC;


-- Further Analysis: Return Rate for Each Delivery Type

SELECT
    delivery_type,
    CAST(
        COUNT(CASE WHEN status = 'returned' THEN 1 END) * 100.0 / COUNT(*)
        AS DECIMAL(5, 2)
    ) AS return_rate
FROM orders
GROUP BY delivery_type
ORDER BY return_rate DESC;
```

```sql
--  Insight Summary (Q16 Extension):

-- Among the three delivery types, Shipped from Abroad had the longest
-- average delivery time of approximately 15 days and also the highest
-- return rate at 30.92%, making it a major operational red flag.
-- Despite being the fastest, Express delivery (avg ~3 days) showed a
-- return rate of 26.38%, possibly indicating product quality issues or
-- customer dissatisfaction.
-- Standard Delivery averaged ~9 days and had the lowest return rate
-- (23.73%), making it the most balanced fulfillment method with
-- optimization potential.


-- Q17: Delayed Deliveries Count (If Delivery > 5 Days)
-- Objective: Count and analyze delayed orders to reduce future delays

-- Method 1: Subquery
SELECT COUNT(*) AS delay_delivery
FROM (
    SELECT DATEDIFF(DAY, orderdate, delivery_date) AS days_diff_count
    FROM orders
    WHERE DATEDIFF(DAY, orderdate, delivery_date) > 5
) delay_count;


-- Method 2: Simplified (Same Output)

SELECT COUNT(*) AS delay_delivery
FROM orders
WHERE DATEDIFF(DAY, orderdate, delivery_date) > 5;

-- Breakdown by Delivery Type (Optional Deep-Dive)
SELECT
    delivery_type,
    COUNT(*) AS delayed_orders
FROM orders
WHERE DATEDIFF(DAY, orderdate, delivery_date) > 5
GROUP BY delivery_type
```

```sql
ORDER BY delayed_orders DESC;


-- Q18: Fastest and Slowest Zones (Average Delivery Days)
-- Objective: Identify zone-wise delivery efficiency or bottlenecks

SELECT
    zone,
    AVG(DATEDIFF(DAY, orderdate, delivery_date)) AS avg_delivery_days
FROM orders
GROUP BY zone;


-- Q19: Product Categories with Highest Express Orders
-- Objective: Analyze which product types rely heavily on fast shipping

SELECT TOP 1
    p.product_category,
    COUNT(CASE WHEN od.delivery_type = 'Express' THEN 1 END) AS
express_count
FROM orders od
JOIN product p ON od.productid = p.productid
GROUP BY p.product_category
ORDER BY express_count DESC;


-- Q20: Delivery Type Preference by Zone
-- Objective: Determine the most preferred delivery method per zone

-- Method 1: Without Window Function (Simple Aggregation)
SELECT
    delivery_type,
    zone,
    COUNT(*) AS delivery_count
FROM orders
WHERE status = 'delivered'
GROUP BY delivery_type, zone
ORDER BY delivery_count DESC;

-- Use Case: Visualize full breakdown of delivery preferences by zone
-- Method 2: With Window Function (Dense Rank)

WITH Ranked_zone AS (
    SELECT
        delivery_type,
        zone,
        COUNT(*) AS delivery_count,
        DENSE_RANK() OVER (
            PARTITION BY zone
            ORDER BY COUNT(*) DESC
        ) AS delivery_zone_rank
    FROM orders
    WHERE status = 'delivered'
    GROUP BY delivery_type, zone
)
SELECT *
FROM Ranked_zone
WHERE delivery_zone_rank = 1
ORDER BY delivery_count DESC;

-- Explanation:
-- DESC ensures the most-used delivery type gets rank 1 within each zone.
-- This helps identify delivery-type popularity at a zonal level for
-- operational alignment.
```