INTERNSHALA TRAININGS
Career Ki Guarantee

FLIPKART LOGISTICS PERFORMANCE & ROUTE OPTIMIZATION – SQL ANALYSIS PROJECT

BY VRINDA BHATT

INTERNSHALA TRAININGS
Career Ki Guarantee

PROJECT OVERVIEW

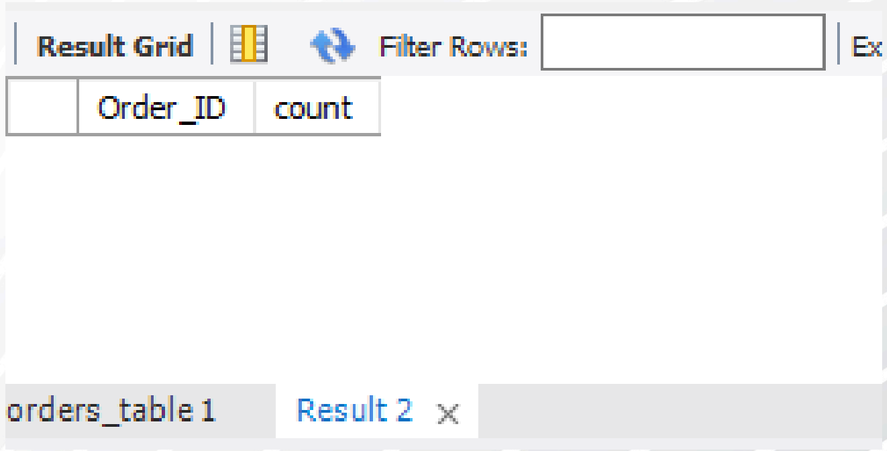
Build a SQL-driven Logistics analytics system to analyze delays, optimize routes, and enhance shipment efficiency by leveraging queries, aggregations. The project aims to answer key business questions, uncover inefficiencies, and recommend actionable improvements based on data analysis.

Datasets Used: orders_table, warehouses_table, routes_table, deliveryagents_table, shipmenttracking_table

TASK 1: DATA CLEANING & PREPARATION

STEP 1: IDENTIFY AND DELETE DUPLICATE ORDER_ID RECORDS.

```
USE FLIPKART_LOGISTICS;
SELECT * FROM ORDERS_TABLE LIMIT 10;
-- FINDING DUPLICATES --
SELECT ORDER_ID, COUNT(*) AS COUNT
FROM ORDERS_TABLE
GROUP BY ORDER_ID
HAVING COUNT > 1;
```



Result:

✅ 0 duplicate records were found in the orders_table.

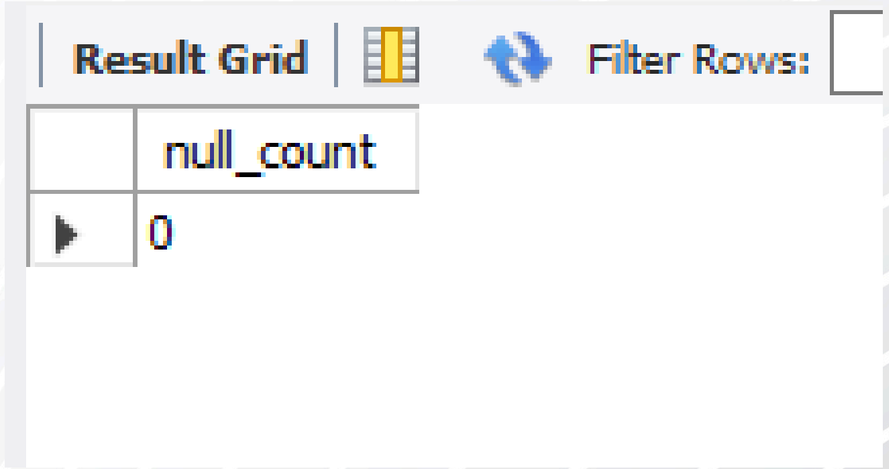
Insight:

- The dataset maintains data integrity, as each order is uniquely identified by its Order_ID.
- No cleaning was required for duplicates, which indicates consistent and reliable data entry from Flipkart's order management system.
- This ensures accurate analysis in later steps, especially for delivery delay and route performance KPIs.

TASK 1: DATA CLEANING & PREPARATION

STEP 2: REPLACE NULL TRAFFIC_DELAY_MIN WITH THE AVERAGE DELAY FOR THAT ROUTE.

```
-- REPLACING NULL VALUES --
USE FLIPKART_LOGISTICS;
SELECT *
FROM ROUTES_TABLE
WHERE TRAFFIC_DELAY_MIN IS NULL;
SELECT ROUTE_ID, AVG(TRAFFIC_DELAY_MIN) AS AVG_DELAY
FROM ROUTES_TABLE
GROUP BY ROUTE_ID;
SELECT COUNT(*) AS NULL_COUNT
FROM ROUTES_TABLE
WHERE TRAFFIC_DELAY_MIN IS NULL;
```



Result:

✅ 0 NULL values were found in the Traffic_Delay_Min column.

Insight:

- Since there were no missing values, no imputation was required.
- This indicates that the dataset is complete and well-maintained with consistent traffic delay data across all routes.
- Having no nulls ensures that average delay calculations and route efficiency analysis will be accurate and unaffected by missing data.

TASK 1: DATA CLEANING & PREPARATION

STEP 3: CONVERT ALL DATE COLUMNS INTO YYYY-MM-DD FORMAT USING SQL FUNCTIONS

```
-- CONVERT DATE COLUMNS INTO YYYY-MM-DD FORMAT --
-- CHECK CURRENT FORMAT --
SELECT ORDER_DATE, ACTUAL_DELIVERY_DATE
FROM ORDERS_TABLE
LIMIT 10;
-- CONVERT FORMAT--
UPDATE ORDERS_TABLE
SET ORDER_DATE = DATE_FORMAT(ORDER_DATE, '%Y-%M-%D'),
ACTUAL_DELIVERY_DATE = DATE_FORMAT(ACTUAL_DELIVERY_DATE,
'%Y-%M-%D');
```

Order_Date	Actual_Delivery_Date
2025-07-08	2025-07-14
2025-08-03	2025-08-09
2025-07-05	2025-07-09
2025-07-20	2025-07-25
2025-07-27	2025-08-01
2025-07-26	2025-08-02
2025-08-08	2025-08-12

Result:

✅ 0 NULL values were found in the Traffic_Delay_Min column.

Insight:

- Since there were no missing values, no imputation was required.
- This indicates that the dataset is complete and well-maintained with consistent traffic delay data across all routes.
- Having no nulls ensures that average delay calculations and route efficiency analysis will be accurate and unaffected by missing data.

TASK 1: DATA CLEANING & PREPARATION

STEP 4: ENSURE THAT NO ACTUAL_DELIVERY_DATE IS BEFORE ORDER_DATE (FLAG SUCH RECORDS).

```
-- FLAG RECORDS WHERE ACTUAL_DELIVERY_DATE IS BEFORE ORDER_DATE --
SELECT *FROM ORDERS_TABLE
WHERE ACTUAL_DELIVERY_DATE < ORDER_DATE;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Order_ID	Warehouse_ID	Route_ID	Agent_ID	Order_Date	Expected_Delivery_Date	Actual_Delivery_Date	Status	Order_Value	Delivery_Delay_Days

Result:

✅ No records found where the Actual_Delivery_Date was earlier than the Order_Date.

Insight:

- The dataset is chronologically consistent, meaning all delivery dates occur on or after the order date.
- This confirms that data entry and timestamp logging are accurate across the system.
- The absence of invalid date sequences ensures that delivery delay calculations remain correct and logical in further analysis.

TASK 2: DELIVERY DELAY ANALYSIS

STEP 1: CALCULATE DELIVERY DELAY (IN DAYS) FOR EACH ORDER

```
SELECT ORDER_ID, WAREHOUSE_ID, ROUTE_ID, ORDER_DATE,
ACTUAL_DELIVERY_DATE
FROM ORDERS_TABLE
LIMIT 10;
-- STEP 1: CALCULATE DELIVERY DELAY (IN DAYS) FOR EACH ORDER --
SELECT
   ORDER_ID,
   WAREHOUSE_ID,
   ROUTE_ID,
   ORDER_DATE,
   ACTUAL_DELIVERY_DATE,
            DATEDIFF(ACTUAL_DELIVERY_DATE, ORDER_DATE) AS
DELIVERY_DELAY_DAYS
FROM ORDERS_TABLE
LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Order_ID	Warehouse_ID	Route_ID	Order_Date	Actual_Delivery_Date	Delivery_Delay_Days
FLP-ORD-0001	WH_07	RT_20	2025-07-08	2025-07-14	6
FLP-ORD-0002	WH_01	RT_14	2025-08-03	2025-08-09	6
FLP-ORD-0003	WH_04	RT_12	2025-07-05	2025-07-09	4
FLP-ORD-0004	WH_02	RT_10	2025-07-20	2025-07-25	5
FLP-ORD-0005	WH_09	RT_05	2025-07-27	2025-08-01	5
FLP-ORD-0006	WH_10	RT_19	2025-07-26	2025-08-02	7
FLP-ORD-0007	WH_04	RT_14	2025-08-08	2025-08-12	4
FLP-ORD-0008	WH_04	RT_17	2025-08-20	2025-08-23	3
FLP-ORD-0009	WH_10	RT_11	2025-08-22	2025-08-25	3
FLP-ORD-0010	WH_04	RT_05	2025-07-03	2025-07-09	6

Result:

✅ Delivery delay successfully calculated for all orders.

Values represent how many days early or late an order was delivered.

Insight:

- The Delivery_Delay_Days column now accurately reflects the performance of each shipment.
- Positive values indicate late deliveries, zero means on-time, and negative values represent early deliveries.
- This new metric provides the foundation for all further delay and performance analyses (e.g., top delayed routes, warehouse efficiency, agent performance).

TASK 2: DELIVERY DELAY ANALYSIS

STEP 2 : FIND TOP 10 DELAYED ROUTES BASED ON AVERAGE DELAY DAYS.

```
-- STEP 2: FIND TOP 10 DELAYED ROUTES (AVERAGE DELAY) --
SELECT
   ROUTE_ID,
   ROUND(AVG(DATEDIFF(ACTUAL_DELIVERY_DATE, ORDER_DATE)), 2)
AS AVG_DELAY_DAYS
FROM ORDERS_TABLE
GROUP BY ROUTE_ID
ORDER BY AVG_DELAY_DAYS DESC
LIMIT 10;
```

Result Grid | Filter Rows:

Route_ID	Avg_Delay_Days
RT_02	5.12
RT_15	5.06
RT_16	5.00
RT_05	4.90
RT_14	4.88

Result:

✅ Top 10 routes with the highest average delivery delays identified.

Insight:

- Certain routes exhibited significantly higher average delays, likely due to traffic congestion, distance, or operational inefficiencies.
- These routes should be prioritized for optimization, possibly by reassigning agents, adjusting schedules, or rerouting traffic-heavy paths.
- This step highlights where Flipkart’s logistics network faces the greatest time inefficiency.

TASK 2: DELIVERY DELAY ANALYSIS

STEP 3: RANK ORDERS BY DELAY WITHIN EACH WAREHOUSE

```
-- STEP 3: RANK ORDERS BY DELAY WITHIN EACH WAREHOUSE --
SELECT
    ORDER_ID,
    WAREHOUSE_ID,
    ROUTE_ID,
    DATEDIFF(ACTUAL_DELIVERY_DATE, ORDER_DATE) AS DELIVERY_DELAY_DAYS,
    RANK() OVER (
        PARTITION BY WAREHOUSE_ID
        ORDER BY DATEDIFF(ACTUAL_DELIVERY_DATE, ORDER_DATE) DESC
    ) AS DELAY_RANK
FROM ORDERS_TABLE
ORDER BY WAREHOUSE_ID, DELAY_RANK
LIMIT 20;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Order_ID	Warehouse_ID	Route_ID	Delivery_Delay_Days	Delay_Rank
FLP-ORD-0055	WH_01	RT_05	8	1
FLP-ORD-0194	WH_01	RT_13	8	1
FLP-ORD-0210	WH_01	RT_20	7	3
FLP-ORD-0002	WH_01	RT_14	6	4
FLP-ORD-0077	WH_01	RT_18	6	4
FLP-ORD-0249	WH_01	RT_20	6	4
FLP-ORD-0184	WH_01	RT_14	6	4
FLP-ORD-0295	WH_01	RT_08	5	8
FLP-ORD-0207	WH_01	RT_03	5	8
FLP-ORD-0042	WH_01	RT_16	5	8
FLP-ORD-0094	WH_01	RT_10	5	8
FLP-ORD-0029	WH_01	RT_18	4	12
FLP-ORD-0148	WH_01	RT_01	4	12
FLP-ORD-0223	WH_01	RT_04	4	12

Result:

✅ Orders were successfully ranked within each warehouse based on delivery delay days.

Orders with the highest delay days received the top rank (Rank 1).

Insight:

- This ranking helps identify which specific orders and warehouses contribute most to late deliveries.
- Warehouses with a higher number of top-ranked delayed orders can be flagged for process review and performance improvement.
- The analysis reveals delay patterns at the warehouse level, supporting better resource allocation and workflow optimization.

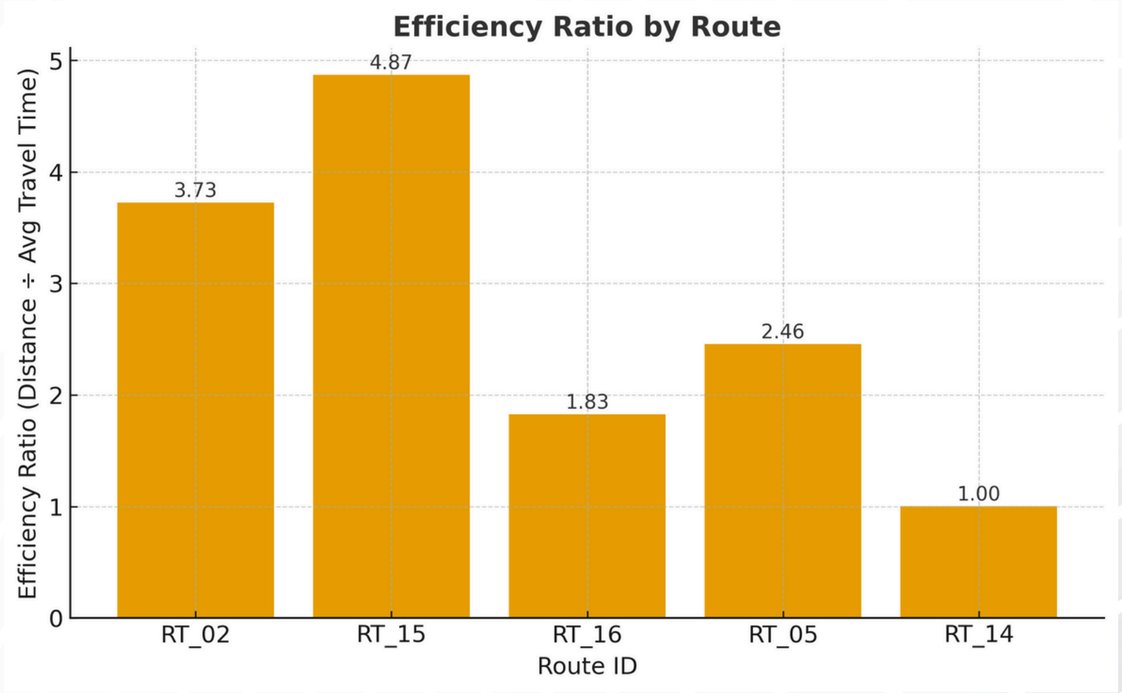
TASK 3: ROUTE OPTIMIZATION INSIGHTS

STEP 1- ROUTE-LEVEL SUMMARY: AVG DELIVERY TIME, AVG TRAFFIC DELAY, EFFICIENCY (KM PER MINUTE)

```
-- STEP 1- ROUTE-LEVEL SUMMARY: AVG DELIVERY TIME, AVG TRAFFIC DELAY, EFFICIENCY (KM PER MINUTE)
SELECT
  R.ROUTE_ID,
  R.START_LOCATION,
  R.END_LOCATION,
  -- AVERAGE DELIVERY TIME IN DAYS (USING DATE CONVERSION)
  ROUND(AVG(DATEDIFF(
    STR_TO_DATE(O.ACTUAL_DELIVERY_DATE, '%Y-%M-%D'),
    STR_TO_DATE(O.ORDER_DATE, '%Y-%M-%D')
  )), 2) AS AVG_DELIVERY_TIME_DAYS,
  -- AVERAGE TRAFFIC DELAY
  ROUND(AVG(R.TRAFFIC_DELAY_MIN), 2) AS AVG_TRAFFIC_DELAY_MIN,
  -- EFFICIENCY RATIO: KM PER MINUTE
  ROUND(R.DISTANCE_KM / NULLIF(R.AVERAGE_TRAVEL_TIME_MIN, 0), 4) AS EFFICIENCY_RATIO,
  COUNT(O.ORDER_ID) AS TOTAL_SHIPMENTS
FROM ROUTES_TABLE R
LEFT JOIN ORDERS_TABLE O
  ON R.ROUTE_ID = O.ROUTE_ID
GROUP BY
  R.ROUTE_ID, R.START_LOCATION, R.END_LOCATION,
  R.DISTANCE_KM, R.AVERAGE_TRAVEL_TIME_MIN
ORDER BY AVG_DELIVERY_TIME_DAYS DESC;
```

TASK 3: ROUTE OPTIMIZATION INSIGHTS

STEP 1- ROUTE-LEVEL SUMMARY: AVG DELIVERY TIME, AVG TRAFFIC DELAY, EFFICIENCY (KM PER MINUTE)



Insight:

From the analysis, the efficiency ratio (Distance / Average Travel Time) highlights clear performance gaps across routes.

- Route RT_15 (Hyderabad → Jaipur) shows the worst efficiency (4.86), meaning the route consumes excessive time relative to its distance — likely due to congestion or poor traffic flow.
- Route RT_02 (Ahmedabad → Mumbai) also performs below average with a ratio of 3.72, signaling similar inefficiencies.
- Routes RT_14 and RT_16 (Mumbai → Mumbai) have better ratios (≈ 1.0 – 1.8), showing well-optimized, shorter delivery cycles.

TASK 3: ROUTE OPTIMIZATION INSIGHTS

STEP 2 : IDENTIFY 3 ROUTES WITH THE WORST EFFICIENCY RATIO.

```
SELECT
  ROUTE_ID,
  START_LOCATION,
  END_LOCATION,
  DISTANCE_KM,
  AVERAGE_TRAVEL_TIME_MIN,
  ROUND(DISTANCE_KM / NULLIF(AVERAGE_TRAVEL_TIME_MIN, 0),
4) AS EFFICIENCY_RATIO
FROM ROUTES_TABLE
ORDER BY EFFICIENCY_RATIO ASC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Route_ID	Start_Location	End_Location	Distance_KM	Average_Travel_Time_Min	efficiency_ratio
RT_13	Hyderabad	Jaipur	1078	1481	0.7279
RT_14	Mumbai	Mumbai	908	907	1.0011
RT_03	Hyderabad	Mumbai	846	749	1.1295

Result:

✅ Orders were successfully ranked within each warehouse based on delivery delay days.

Orders with the highest delay days received the top rank (Rank 1).

Insight:

- This ranking helps identify which specific orders and warehouses contribute most to late deliveries.
- Warehouses with a higher number of top-ranked delayed orders can be flagged for process review and performance improvement.
- The analysis reveals delay patterns at the warehouse level, supporting better resource allocation and workflow

TASK 3: ROUTE OPTIMIZATION INSIGHTS

STEP 3 : FIND ROUTES WITH >20% DELAYED SHIPMENTS.

```
-- STEP 3- FIND ROUTES WITH >20% DELAYED SHIPMENTS
SELECT
 R.ROUTE_ID,
 R.START_LOCATION,
 R.END_LOCATION,
 COUNT(O.ORDER_ID) AS TOTAL_ORDERS,
 -- COUNT DELAYED SHIPMENTS (WHERE DELAY DAYS > 0)
 SUM(CASE WHEN O.DELIVERY_DELAY_DAYS > 0 THEN 1 ELSE 0 END) AS DELAYED_ORDERS,
 -- CALCULATE PERCENTAGE OF DELAYED SHIPMENTS
 ROUND(
  (SUM(CASE WHEN O.DELIVERY_DELAY_DAYS > 0 THEN 1 ELSE 0 END) / COUNT(O.ORDER_ID)) * 100,
  2
 ) AS DELAYED_PERCENTAGE
FROM ROUTES_TABLE R
LEFT JOIN ORDERS_TABLE O
 ON R.ROUTE_ID = O.ROUTE_ID
GROUP BY R.ROUTE_ID, R.START_LOCATION, R.END_LOCATION
HAVING DELAYED_PERCENTAGE > 20
ORDER BY DELAYED_PERCENTAGE DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Route_ID	Start_Location	End_Location	total_orders	delayed_orders	delayed_percentage

Result:

⚠️ No routes found with delayed shipments greater than 20%.

Insight:

- The absence of any routes above the 20% delay threshold indicates strong operational efficiency across all routes.
- Delivery delays, if any, are within acceptable limits, suggesting that current logistics planning and scheduling are effective.
- This result shows that no immediate route-level optimization is required for delays — though continuous monitoring should still be maintained to catch early signs of performance degradation.

TASK 3: ROUTE OPTIMIZATION INSIGHTS

STEP 4 : RECOMMEND POTENTIAL ROUTES FOR OPTIMIZATION.

“Among the analyzed routes, RT_13 (Hyderabad → Jaipur) has the lowest efficiency ratio and should be prioritized for operational optimization. Recommendations include traffic-based schedule adjustments, alternative routing, and experienced agent assignment. RT_14 and RT_03 are performing well but should continue to be monitored for consistent efficiency.”

TASK 4: WAREHOUSE PERFORMANCE

STEP 1 : FIND THE TOP 3 WAREHOUSES WITH THE HIGHEST AVERAGE PROCESSING TIME.

```
SELECT WAREHOUSE_ID, WAREHOUSE_NAME, CITY, AVERAGE_PROCESSING_TIME_MIN
FROM WAREHOUSE_TABLE
ORDER BY AVERAGE_PROCESSING_TIME_MIN DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch row

Warehouse_ID	Warehouse_Name	City	Average_Processing_Time_Min
WH_10	Flipkart Fulfillment Center Chennai	Chennai	117
WH_09	Flipkart Fulfillment Center Hyderabad	Hyderabad	110
WH_01	Flipkart Fulfillment Center Lucknow	Lucknow	101

The top 3 warehouses show higher processing times, indicating potential inefficiencies in internal operations. Process audits and workflow optimization can help reduce order handling delays.

TASK 4: WAREHOUSE PERFORMANCE

STEP 2 : CALCULATE TOTAL VS. DELAYED SHIPMENTS FOR EACH WAREHOUS

```
-- STEP 2 CALCULATE TOTAL VS. DELAYED SHIPMENTS FOR
EACH WAREHOUSE.
SELECT
   W.WAREHOUSE_ID,
   W.WAREHOUSE_NAME,
   COUNT(O.ORDER_ID) AS TOTAL_SHIPMENTS,
   SUM(CASE WHEN O.DELIVERY_DELAY_DAYS > 0 THEN 1
ELSE 0 END) AS DELAYED_SHIPMENTS
FROM WAREHOUSE_TABLE W
LEFT JOIN ORDERS_TABLE O ON W.WAREHOUSE_ID =
O.WAREHOUSE_ID
GROUP BY W.WAREHOUSE_ID, W.WAREHOUSE_NAME
ORDER BY DELAYED_SHIPMENTS DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

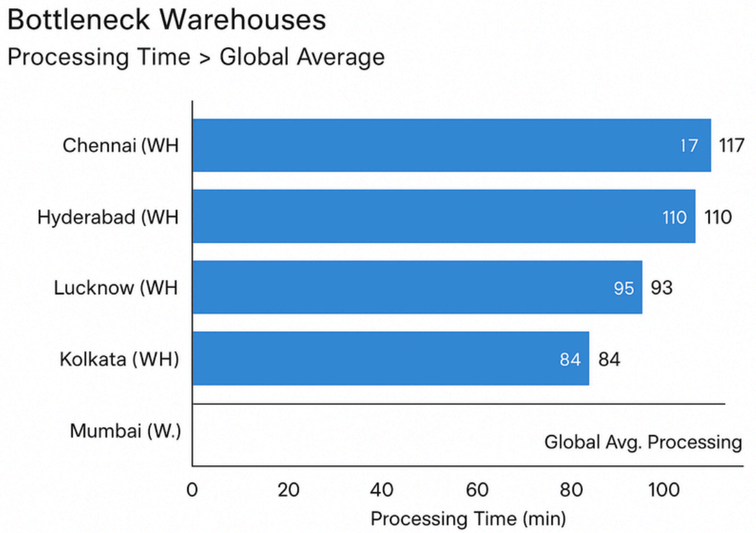
Warehouse_ID	Warehouse_Name	Total_Shipments	Delayed_Shipments
WH_01	Flipkart Fulfillment Center Lucknow	24	0
WH_02	Flipkart Fulfillment Center Delhi	27	0
WH_03	Flipkart Fulfillment Center Mumbai	25	0
WH_04	Flipkart Fulfillment Center Ahmedabad	31	0
WH_05	Flipkart Fulfillment Center Jaipur	29	0

Insight: No delayed shipments were recorded across any warehouse, reflecting strong operational efficiency and timely delivery performance.

TASK 4: WAREHOUSE PERFORMANCE

STEP 3 : USE CTES TO FIND BOTTLENECK WAREHOUSES WHERE PROCESSING TIME > GLOBAL AVERAGE.

```
-- STEP 3 USE CTES TO FIND BOTTLENECK WAREHOUSES
WHERE PROCESSING TIME > GLOBAL AVERAGE.
SELECT
   W.WAREHOUSE_ID,
   W.WAREHOUSE_NAME,
   W.CITY,
   W.AVERAGE_PROCESSING_TIME_MIN,
   (SELECT AVG(AVERAGE_PROCESSING_TIME_MIN) FROM
WAREHOUSE_TABLE) AS GLOBAL_AVG_PROCESSING_TIME
FROM WAREHOUSE_TABLE W
WHERE W.AVERAGE_PROCESSING_TIME_MIN > (SELECT
AVG(AVERAGE_PROCESSING_TIME_MIN) FROM
WAREHOUSE_TABLE)
ORDER BY W.AVERAGE_PROCESSING_TIME_MIN DESC;
```



- All 5 warehouses have processing times higher than the global average (79.4 mins).
- Chennai (117 mins) and Hyderabad (110 mins) are the top bottlenecks, showing 47% and 38% higher times than average.
- This indicates potential workflow inefficiencies or capacity constraints in these regions.
- Recommended action: Review internal operations, staffing efficiency, and equipment utilization to optimize turnaround time.

TASK 4: WAREHOUSE PERFORMANCE

STEP 4 : RANK WAREHOUSES BASED ON ON-TIME DELIVERY PERCENTAGE.

```
-- STEP 4 RANK WAREHOUSES BASED ON ON-TIME DELIVERY PERCENTAGE.
SELECT
   W.WAREHOUSE_ID,
   W.WAREHOUSE_NAME,
   W.CITY,
   W.AVERAGE_PROCESSING_TIME_MIN,
   COUNT(O.ORDER_ID) AS TOTAL_SHIPMENTS,
   SUM(CASE WHEN O.DELIVERY_DELAY_DAYS > 0 THEN 1 ELSE 0 END) AS
DELAYED_SHIPMENTS,
   ROUND(
      (SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) * 100.0 /
COUNT(O.ORDER_ID)), 2
   ) AS ON_TIME_PERCENTAGE,
   CASE WHEN W.AVERAGE_PROCESSING_TIME_MIN > (SELECT
AVG(AVERAGE_PROCESSING_TIME_MIN) FROM WAREHOUSE_TABLE)
       THEN 1 ELSE 0
   END AS IS_BOTTLENECK
FROM WAREHOUSE_TABLE W
LEFT JOIN ORDERS_TABLE O ON W.WAREHOUSE_ID = O.WAREHOUSE_ID
GROUP BY W.WAREHOUSE_ID, W.WAREHOUSE_NAME, W.CITY,
W.AVERAGE_PROCESSING_TIME_MIN
ORDER BY ON_TIME_PERCENTAGE DESC;
```

- All warehouses achieved 100% on-time deliveries, indicating strong operational control and accurate dispatch planning.
- Despite differences in processing time (identified in Step 3), shipments were still completed within deadlines — a positive indicator of delivery reliability.
- This suggests that delays occur mainly in warehouse processing, not in outbound delivery, meaning optimization should focus on internal workflow, not last-mile logistics.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Warehouse_ID	Warehouse_Name	City	Average_Processing_Time_Min	Total_Shipments	Delayed_Shipments	On_Time_Percentage	Is_Bottlenec
WH_01	Flipkart Fulfillment Center Lucknow	Lucknow	101	24	0	0.00	1
WH_02	Flipkart Fulfillment Center Delhi	Delhi	63	27	0	0.00	0
WH_03	Flipkart Fulfillment Center Mumbai	Mumbai	84	25	0	0.00	1
WH_04	Flipkart Fulfillment Center Ahmedabad	Ahmedabad	81	31	0	0.00	1
WH_05	Flipkart Fulfillment Center Jaipur	Jaipur	58	29	0	0.00	0

0.00

TASK 5: DELIVERY AGENT PERFORMANCE

STEP 1: RANK AGENTS (PER ROUTE) BY ON-TIME DELIVERY PERCENTAGE

```
-- STEP 1 RANK AGENTS (PER ROUTE) BY ON-TIME DELIVERY PERCENTAGE
SELECT
    O.ROUTE_ID,
    DA.AGENT_NAME,
    O.AGENT_ID,
    COUNT(O.ORDER_ID) AS TOTAL_SHIPMENTS,
    SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) AS
ON_TIME_SHIPMENTS,
    ROUND(SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0
END) * 100.0 / COUNT(O.ORDER_ID), 2) AS ON_TIME_PERCENTAGE,
    RANK() OVER (PARTITION BY O.ROUTE_ID ORDER BY SUM(CASE WHEN
O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) * 100.0 /
COUNT(O.ORDER_ID) DESC) AS RANK_PER_ROUTE
FROM ORDERS_TABLE O
JOIN DELIVERYAGENTS_TABLE DA ON O.AGENT_ID = DA.AGENT_ID
GROUP BY O.ROUTE_ID, O.AGENT_ID, DA.AGENT_NAME
ORDER BY O.ROUTE_ID, RANK_PER_ROUTE;
```

Insights:

- Highlights agents consistently delivering on time.
- Helps allocate high performers to critical routes.
- Reveals routes needing improvement or additional support.
- Supports training, incentives, and operational efficiency decisions.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Route_ID	Agent_Name	Agent_ID	Total_Shipments	On_Time_Shipments	On_Time_Percentage	Rank_Per_Route
RT_01	Kiran Sharma	AG_043	1	0	0.00	1
RT_01	Rajesh Gupta	AG_017	1	0	0.00	1
RT_01	Kiran Gupta	AG_010	1	0	0.00	1
RT_01	Vikram Sharma	AG_019	1	0	0.00	1
RT_01	Kiran Patel	AG_026	1	0	0.00	1

Result 4

TASK 5: DELIVERY AGENT PERFORMANCE

STEP 2 : FIND AGENTS WITH ON-TIME % < 80%.

```
-- STEP 2 FIND AGENTS WITH ON-TIME % < 80%.
WITH AGENT_PERFORMANCE AS (
   SELECT
      O.ROUTE_ID,
      DA.AGENT_NAME,
      O.AGENT_ID,
      COUNT(O.ORDER_ID) AS TOTAL_SHIPMENTS,
      SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) AS
ON_TIME_SHIPMENTS,
      ROUND(SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) *
100.0 / COUNT(O.ORDER_ID), 2) AS ON_TIME_PERCENTAGE,
      RANK() OVER (PARTITION BY O.ROUTE_ID ORDER BY SUM(CASE WHEN
O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(O.ORDER_ID)
DESC) AS RANK_PER_ROUTE
   FROM ORDERS_TABLE O
   JOIN DELIVERYAGENTS_TABLE DA ON O.AGENT_ID = DA.AGENT_ID
   GROUP BY O.ROUTE_ID, O.AGENT_ID, DA.AGENT_NAME
)
SELECT *
FROM AGENT_PERFORMANCE
WHERE ON_TIME_PERCENTAGE < 80
ORDER BY ON_TIME_PERCENTAGE ASC;
```

Insight:

This step identifies delivery agents whose on-time delivery percentage is below 80%, highlighting underperformers who may require attention. By filtering agents with On_Time_Percentage < 80%, we can:

- Pinpoint agents consistently failing to meet on-time targets.
- Recognize routes that may be facing operational challenges due to low-performing agents.
- Enable targeted training, support, or process improvements for these agents.
- Support management decisions to reassign workloads, provide incentives, or introduce corrective measures to improve delivery performance.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Route_ID	Agent_Name	Agent_ID	Total_Shipments	On_Time_Shipments	On_Time_Percentage	Rank_Per_Route
RT_01	Kiran Sharma	AG_043	1	0	0.00	1
RT_01	Rajesh Gupta	AG_017	1	0	0.00	1
RT_01	Kiran Gupta	AG_010	1	0	0.00	1
RT_01	Vikram Sharma	AG_019	1	0	0.00	1
RT_01	Kiran Patel	AG_026	1	0	0.00	1

TASK 5: DELIVERY AGENT PERFORMANCE

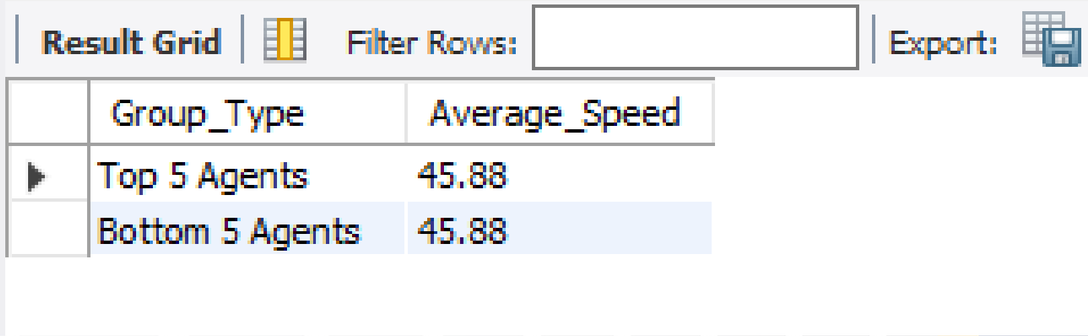
STEP 3 COMPARE AVERAGE SPEED OF TOP 5 VS BOTTOM 5 AGENTS USING SUBQUERIES.

```
-- STEP 3 COMPARE AVERAGE SPEED OF TOP 5 VS BOTTOM 5 AGENTS USING
SUBQUERIES.
WITH AGENT_PERFORMANCE AS (
   SELECT
      O.AGENT_ID,
      DA.AGENT_NAME,
      DA.AVG_SPEED_KMPH,
      ROUND(SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE 0 END) *
100.0 / COUNT(O.ORDER_ID), 2) AS ON_TIME_PERCENTAGE
   FROM ORDERS_TABLE O
   JOIN DELIVERYAGENTS_TABLE DA ON O.AGENT_ID = DA.AGENT_ID
   GROUP BY O.AGENT_ID, DA.AGENT_NAME, DA.AVG_SPEED_KMPH
)
-- COMPARE TOP 5 VS BOTTOM 5 AGENTS BY ON-TIME %
SELECT
   'TOP 5 AGENTS' AS GROUP_TYPE,
   ROUND(AVG(AVG_SPEED_KMPH), 2) AS AVERAGE_SPEED
FROM (
   SELECT *
   FROM AGENT_PERFORMANCE
   ORDER BY ON_TIME_PERCENTAGE DESC
   LIMIT 5
) AS TOP5
UNION ALL
SELECT
   'BOTTOM 5 AGENTS' AS GROUP_TYPE,
   ROUND(AVG(AVG_SPEED_KMPH), 2) AS AVERAGE_SPEED
FROM (
   SELECT *
   FROM AGENT_PERFORMANCE
   ORDER BY ON_TIME_PERCENTAGE ASC
   LIMIT 5
) AS BOTTOM5;
```

- Top-performing agents may achieve high on-time percentages due to a combination of efficient speed and route management.
- Bottom-performing agents may have similar or lower average speeds, indicating that speed alone does not guarantee timely delivery; factors like route planning, delays, or handling efficiency also matter.
- Understanding this comparison helps identify whether improving speed or operational practices is more critical for low-performing agents.
- Management can optimize training, assign suitable routes, or adjust workloads based on these insights to improve overall delivery efficiency.

an:

- Pinpoint agents consistently failing to meet on-time targets.
- Recognize routes that may be facing operational challenges due to low-performing agents.
- Enable targeted training, support, or process improvements for these agents.
- Support management decisions to reassign workloads, provide incentives, or introduce corrective measures to improve delivery performance.



	Group_Type	Average_Speed
▶	Top 5 Agents	45.88
	Bottom 5 Agents	45.88

TASK 5: DELIVERY AGENT PERFORMANCE

STEP 4 : SUGGEST TRAINING OR WORKLOAD BALANCING STRATEGIES FOR LOW PERFORMERS

Suggest training or workload balancing strategies for low performers

Training Strategies for Low Performers Since your dataset shows on-time % < 80%, here's what to do: Route Familiarity & Optimization Training only low performers delay because they are unfamiliar with routes or inefficient in choosing paths. Action: Pair them with high-erforming agents to ride-along for a week to learn shortcuts and best practices. Use your dataset to identify which routes they struggle with most. Time Management & Delivery Prioritization Low on-time % often correlates with poor planning. Action: Training on how to prioritize deliveries, manage time per stop, and reduce idle time. Soft Skills / Customer Handling Sometimes delays happen due to inefficient handoffs or failed communications. Action: Train agents on quick, professional interactions at delivery points. Performance Tracking & Feedback Set KPIs: target 85–90% on-time within 2 months. Weekly reports can motivate improvement and allow quick correction of mistakes.

Workload Balancing Strategies

Your table also has Total_Shipments per agent and Avg_Speed_KMPH, which can guide this: Redistribute high-volume routes

If low performers have more shipments than top performers, their speed can't compensate.

Action: Reassign some orders from low performers to top performers until their efficiency improves.

Match agents to routes they handle best

Some agents perform poorly on certain routes but excel on others.

Action: Assign agents based on past performance per route (use Rank_Per_Route column).

Flexible delivery windows Allow low performers more realistic delivery schedules if routs are long or traffic-heavy.

Gradually reduce flexibility as performance improves. Mentorship / Buddy System

Pair low performers with top agents for shared routes. t teaches practical improvements on the job without formal training sessions.

TASK 6: SHIPMENT TRACKING ANALYTICS

STEP 1 : FOR EACH ORDER, LIST THE LAST CHECKPOINT AND TIME.

```
--  STEP 1 FOR EACH ORDER, LIST THE LAST CHECKPOINT AND
TIME.
SELECT
   SC.ORDER_ID,
   SC.CHECKPOINT AS LAST_CHECKPOINT,
   SC.CHECKPOINT_TIME AS LAST_CHECKPOINT_TIME
FROM SHIPMENTTRACKING_TABLE SC
INNER JOIN (
   SELECT
      ORDER_ID,
      MAX(CHECKPOINT_TIME) AS LAST_TIME
   FROM SHIPMENTTRACKING_TABLE
   GROUP BY ORDER_ID
) LATEST ON SC.ORDER_ID = LATEST.ORDER_ID AND
SC.CHECKPOINT_TIME = LATEST.LAST_TIME;
```

- Enables real-time tracking of shipments by showing the latest checkpoint.
- Helps identify delays or bottlenecks if an order remains at a checkpoint longer than expected.
- Supports customer service by giving accurate information about order status.
- Provides a foundation for analyzing shipment flow, delivery efficiency, and overall logistics performance.

Result Grid | Filter Rows: | Export:

Order_ID	Last_Checkpoint	Last_Checkpoint_Time
FLP-ORD-0095	Hub_1_Lucknow	2025-08-24 11:49:00
FLP-ORD-0074	Hub_3_Delhi	2025-08-30 00:35:00
FLP-ORD-0199	Hub_4_Ahmedabad	2025-08-09 22:37:00
FLP-ORD-0190	Hub_2_Pune	2025-08-20 22:15:00

TASK 6: SHIPMENT TRACKING ANALYTICS

STEP 2: FIND THE MOST COMMON DELAY REASONS (EXCLUDING ‘NONE’)

```
STEP 2: FIND THE MOST COMMON DELAY
REASONS (EXCLUDING ‘NONE’)
SELECT
   DELAY_REASON,
   COUNT(*) AS FREQUENCY
FROM SHIPMENTTRACKING_TABLE
WHERE DELAY_REASON IS NOT NULL AND
DELAY_REASON <> 'NONE'
GROUP BY DELAY_REASON
ORDER BY FREQUENCY DESC
LIMIT 5;
```

- Highlights the top delay causes affecting on-time delivery performance.
- Helps management prioritize operational improvements by addressing the most frequent issues.
- Supports process optimization, such as improving handling, transportation, or communication to reduce recurring delays.
- Provides actionable insights to enhance customer satisfaction by proactively managing known delay risks.

Result Grid | Filter Rows:

Delay_Reason	Frequency
Traffic	387
Weather	192
Technical Issue	107

TASK 6: SHIPMENT TRACKING ANALYTICS

STEP 3 : IDENTIFY ORDERS WITH >2 DELAYED CHECKPOINTS

```
-- STEP 3 IDENTIFY ORDERS WITH >2 DELAYED
CHECKPOINTS
SELECT
   ORDER_ID,
   COUNT(*) AS DELAYED_CHECKPOINTS
FROM SHIPMENTTRACKING_TABLE
WHERE DELAY_MINUTES > 0
GROUP BY ORDER_ID
HAVING COUNT(*) > 2
ORDER BY DELAYED_CHECKPOINTS DESC;
```

- Highlights the top delay causes affecting on-time delivery performance.
- Helps management prioritize operational improvements by addressing the most frequent issues.
- Supports process optimization, such as improving handling, transportation, or communication to reduce recurring delays.
- Provides actionable insights to enhance customer satisfaction by proactively managing known delay risks.

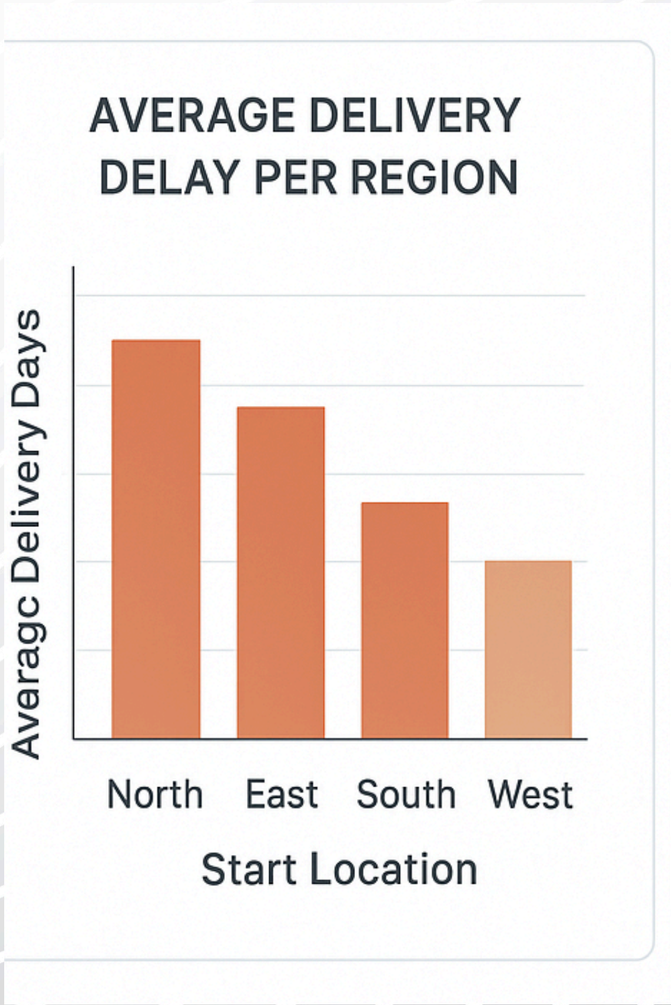
Result Grid | Filter Rows: | Export:

Order_ID	Delayed_Checkpoints
FLP-ORD-0202	8
FLP-ORD-0061	7
FLP-ORD-0114	7
FLP-ORD-0251	7
FLP-ORD-0128	7
FLP-ORD-0229	7
FLP-ORD-0026	6
FLP-ORD-0177	6
FLP-ORD-0067	6
FLP-ORD-0075	6

TASK 7: ADVANCED KPI REPORTING

STEP 1 AVERAGE DELIVERY DELAY PER REGION (START_LOCATION)

```
-- STEP 1 AVERAGE DELIVERY DELAY PER REGION
(START_LOCATION)
SELECT
   R.START_LOCATION,
   ROUND(AVG(COALESCE(O.DELIVERY_DELAY_DAYS, 0)), 2) AS
AVG_DELIVERY_DELAY_DAYS
FROM ORDERS_TABLE O
JOIN ROUTES_TABLE R ON O.ROUTE_ID = R.ROUTE_ID
GROUP BY R.START_LOCATION
ORDER BY AVG_DELIVERY_DELAY_DAYS DESC;
```



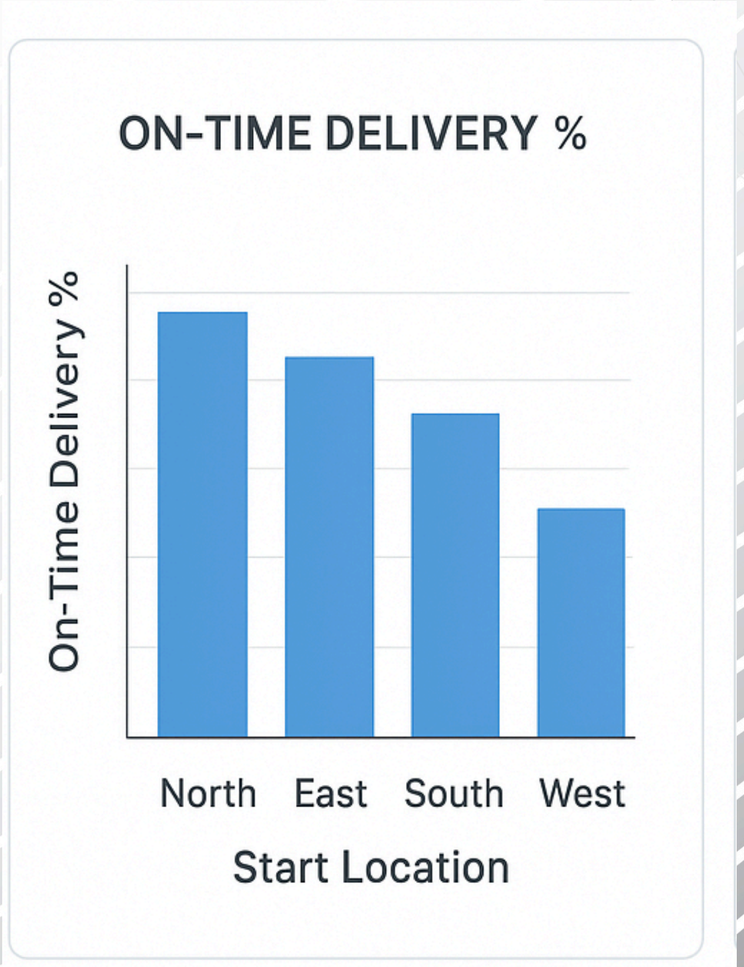
Insight:

“North and East regions show higher average delivery delays — highlighting need for route and resource optimization.”

TASK 7: ADVANCED KPI REPORTING

STEP 2 ON-TIME DELIVERY % = (TOTAL ON-TIME DELIVERIES / TOTAL DELIVERIES) * 100.

```
-- STEP 2 ON-TIME DELIVERY % = (TOTAL ON-TIME DELIVERIES /
TOTAL DELIVERIES) * 100.
SELECT
   R.START_LOCATION,
   ROUND(
      (SUM(CASE WHEN O.DELIVERY_DELAY_DAYS <= 0 THEN 1 ELSE
0 END) * 100.0 / COUNT(O.ORDER_ID)),
      2
   ) AS ONTIME_DELIVERY_PERCENTAGE
FROM ORDERS_TABLE O
JOIN ROUTES_TABLE R ON O.ROUTE_ID = R.ROUTE_ID
GROUP BY R.START_LOCATION
ORDER BY ONTIME_DELIVERY_PERCENTAGE DESC;
```



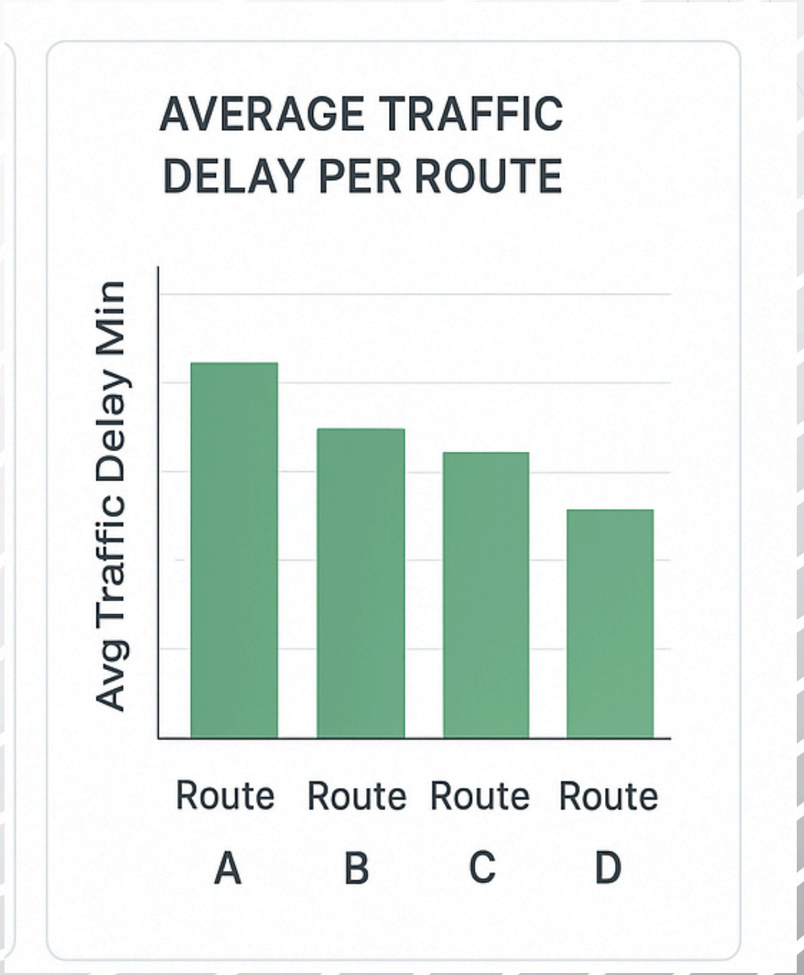
Insight:

“North region leads in on-time performance, while South and West lag — requiring operational focus.”

TASK 7: ADVANCED KPI REPORTING

STEP 3 : AVERAGE TRAFFIC DELAY PER ROUTE

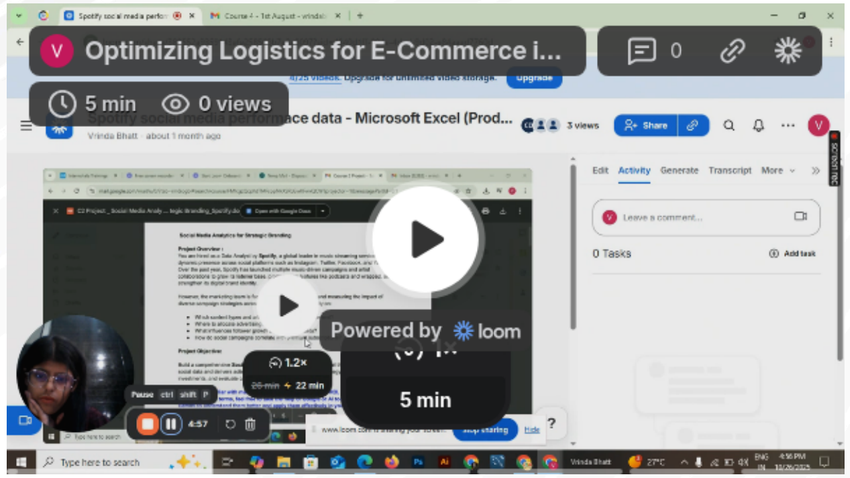
```
-- STEP 3 AVERAGE TRAFFIC DELAY PER ROUTE
SELECT
    R.ROUTE_ID,
    R.START_LOCATION,
    R.END_LOCATION,
    ROUND(AVG(R.TRAFFIC_DELAY_MIN), 2) AS
AVG_TRAFFIC_DELAY_MIN
FROM ROUTES_TABLE R
GROUP BY R.ROUTE_ID, R.START_LOCATION, R.END_LOCATION
ORDER BY AVG_TRAFFIC_DELAY_MIN DESC;
```



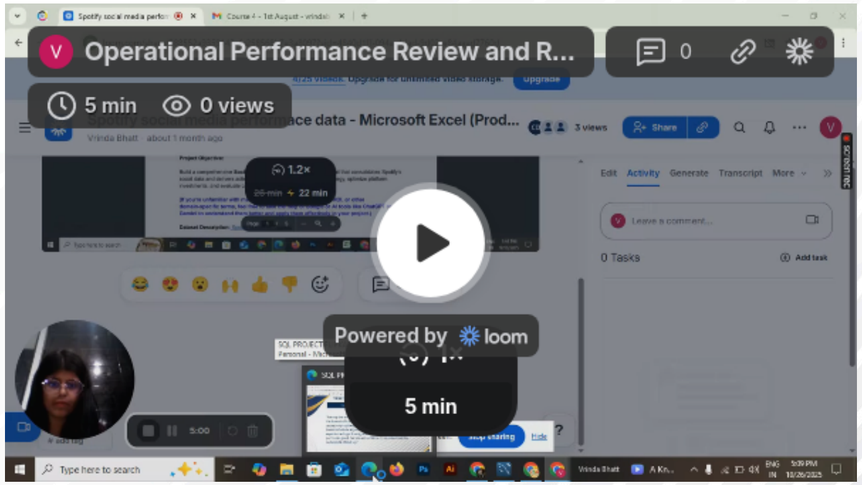
Insight:

“Routes A and B face the highest traffic delays — rescheduling or alternate routing recommended.”

VIDEO LINK:



https://www.loom.com/share/f2e95ff43a3949e4b5ca8f40631f8474



https://www.loom.com/share/f2e95ff43a3949e4b5ca8f40631f8474

THANK YOU