Flipkart SQL Analytics Project

Shipment Tracking & KPI Reporting

Kanishka Kumar | SQL Analytics | 2025

OBJECTIVE

To evaluate shipment delays, checkpoint efficiency, and delivery performance KPIs using SQL-based analytics.

DATA SET USED

Core Tables

- orders
- shipment_tracking

Supporting Tables

- routes
- warehouse
- delivery_agents

DATA FLOW

Orders → Shipment Tracking → Delay & Checkpoint Analysis → KPI Reporting

This structure enables identifying delay patterns and measuring delivery performance at multiple operational levels.

TASK 1

Data Cleaning & Preparation

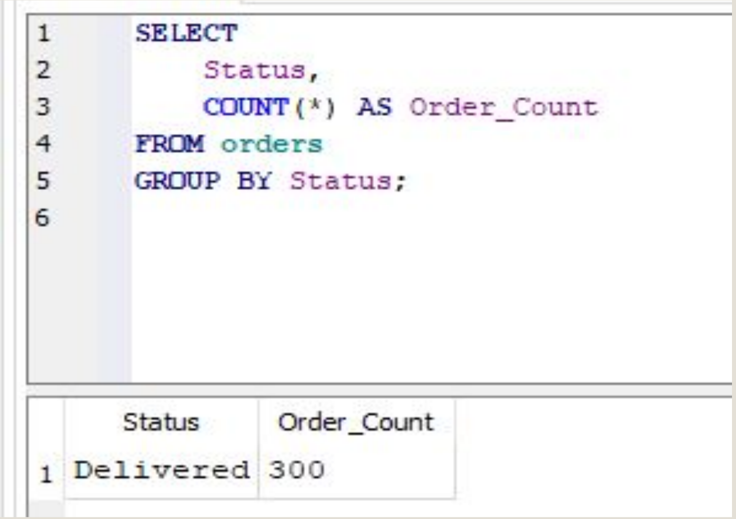
OBJECTIVE: To ensure data accuracy and consistency by removing duplicates, handling missing values, standardizing date formats, and validating logical relationships before performing analytical operations.

Task 1.1: Removing duplicate order records

SQL Query:

```
-- Identify duplicate Order_ID records

DELETE FROM orders
WHERE Order_ID NOT IN (
  SELECT MIN(Order_ID)
  FROM orders
  GROUP BY Order_ID
);
```



```
SELECT
    Status,
    COUNT(*) AS Order_Count
FROM orders
GROUP BY Status;
```

```
     Status      Order_Count
1  Delivered    300
```

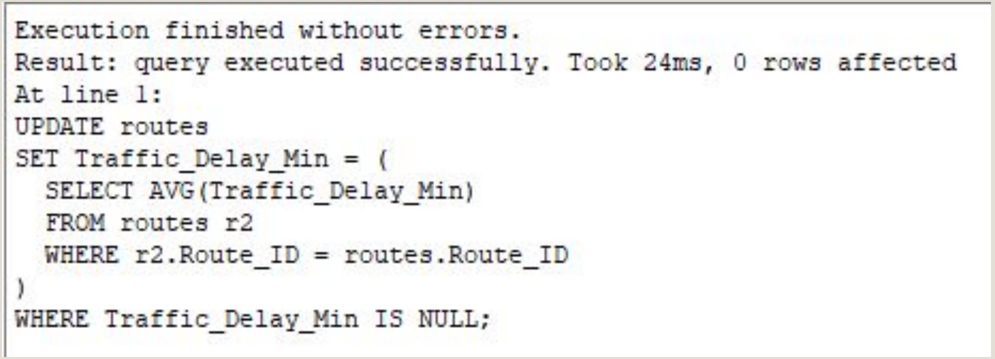
- ➢ Duplicate Order_ID records were checked and validated to ensure data accuracy.
- ➢ No duplicate Order_ID records were found, confirming data consistency across the orders table.

Task 1.2: Handle NULL Traffic Delay

SQL Query:

-- Replacing missing traffic delay values using average delay

```
UPDATE routes
SET Traffic_Delay_Min = (
  SELECT AVG(Traffic_Delay_Min)
  FROM routes r2
  WHERE r2.Route_ID =
routes.Route_ID
)
WHERE Traffic_Delay_Min IS NULL;
```



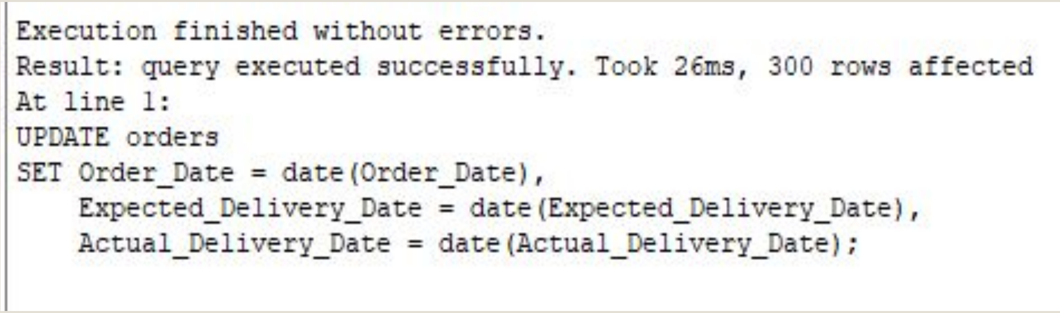
```
Execution finished without errors.
Result: query executed successfully. Took 24ms, 0 rows affected
At line 1:
UPDATE routes
SET Traffic_Delay_Min = (
  SELECT AVG(Traffic_Delay_Min)
  FROM routes r2
  WHERE r2.Route_ID = routes.Route_ID
)
WHERE Traffic_Delay_Min IS NULL;
```

➢ Missing traffic delay values were identified and replaced using average delay values to maintain consistency in route-level analysis.

Task 1.3: Date format standardization

SQL Query:

```
UPDATE orders
SET Order_Date = date(Order_Date),
   Expected_Delivery_Date = date(Expected_Delivery_Date),
   Actual_Delivery_Date = date(Actual_Delivery_Date);
```



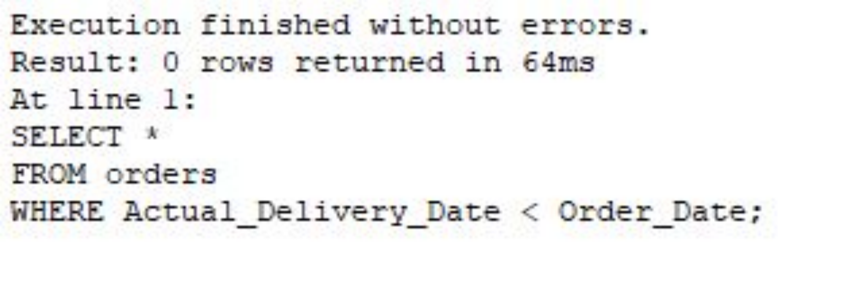
```
Execution finished without errors.
Result: query executed successfully. Took 26ms, 300 rows affected
At line 1:
UPDATE orders
SET Order_Date = date(Order_Date),
    Expected_Delivery_Date = date(Expected_Delivery_Date),
    Actual_Delivery_Date = date(Actual_Delivery_Date);
```

- ➢ Order, expected delivery, and actual delivery dates were standardized to a uniform format to ensure accurate date-based delay calculations.

Task 1.4: Invalid Delivery Date Validation

SQL Query:

```
SELECT *
FROM orders
WHERE Actual_Delivery_Date < Order_Date;
```



```
Execution finished without errors.
Result: 0 rows returned in 64ms
At line 1:
SELECT *
FROM orders
WHERE Actual_Delivery_Date < Order_Date;
```

- No records were found where the actual delivery date precedes the order date, confirming logical consistency and data reliability.

TASK 2

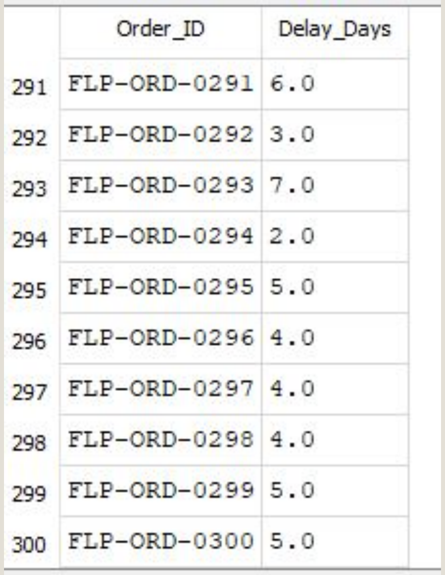
Delivery Delay Analysis

OBJECTIVE: To calculate delivery delays, identify highly delayed orders and routes, and analyze delay patterns using ranking and aggregation techniques.

Task 2.1: Calculate Delivery Delay

SQL Query:

```
SELECT Order_ID,
(julianday(Actual_Delivery_Date) -
julianday(Order_Date)) AS Delay_Days
FROM orders;
```



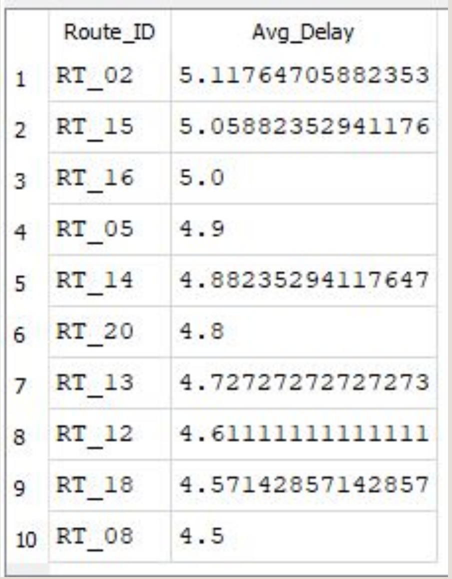
	Order_ID	Delay_Days
291	FLP-ORD-0291	6.0
292	FLP-ORD-0292	3.0
293	FLP-ORD-0293	7.0
294	FLP-ORD-0294	2.0
295	FLP-ORD-0295	5.0
296	FLP-ORD-0296	4.0
297	FLP-ORD-0297	4.0
298	FLP-ORD-0298	4.0
299	FLP-ORD-0299	5.0
300	FLP-ORD-0300	5.0

- This analysis shows the distribution of orders across delivery statuses.
- It helps evaluate overall delivery performance.

Task 2.2: Top 10 Delayed Routes

SQL Query:

```
SELECT Route_ID,
AVG(julianday(Actual_Delivery_Date) -
julianday(Order_Date)) AS Avg_Delay
FROM orders
GROUP BY Route_ID
ORDER BY Avg_Delay DESC
LIMIT 10;
```



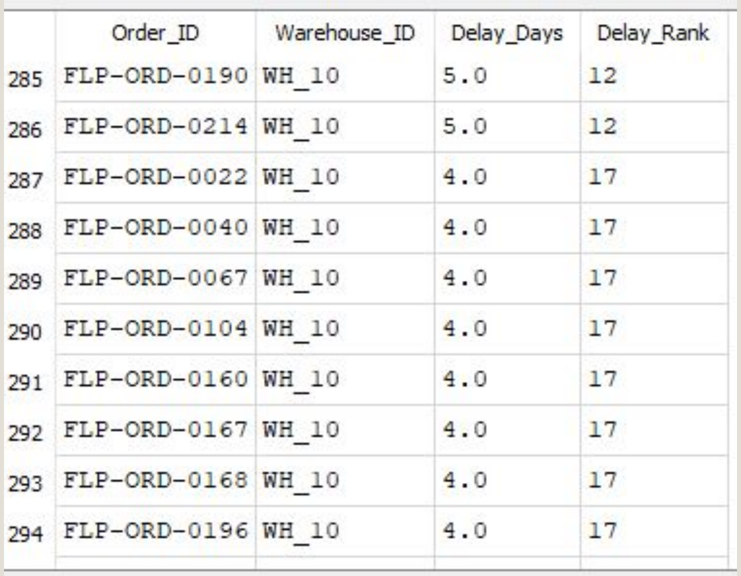
	Route_ID	Avg_Delay
1	RT_02	5.11764705882353
2	RT_15	5.05882352941176
3	RT_16	5.0
4	RT_05	4.9
5	RT_14	4.88235294117647
6	RT_20	4.8
7	RT_13	4.72727272727273
8	RT_12	4.61111111111111
9	RT_18	4.57142857142857
10	RT_08	4.5

- Average delivery delay highlights efficiency gaps in the delivery process.
- It helps identify systemic delivery issues.

Task 2.3: Rank Orders by Delay (Window Function)

SQL Query:

```
SELECT Order_ID, Warehouse_ID,
(julianday(Actual_Delivery_Date) -
julianday(Order_Date)) AS Delay_Days,
RANK() OVER (
  PARTITION BY Warehouse_ID
  ORDER BY (julianday(Actual_Delivery_Date) -
julianday(Order_Date)) DESC
) AS Delay_Rank
FROM orders;
```



	Order_ID	Warehouse_ID	Delay_Days	Delay_Rank
285	FLP-ORD-0190	WH_10	5.0	12
286	FLP-ORD-0214	WH_10	5.0	12
287	FLP-ORD-0022	WH_10	4.0	17
288	FLP-ORD-0040	WH_10	4.0	17
289	FLP-ORD-0067	WH_10	4.0	17
290	FLP-ORD-0104	WH_10	4.0	17
291	FLP-ORD-0160	WH_10	4.0	17
292	FLP-ORD-0167	WH_10	4.0	17
293	FLP-ORD-0168	WH_10	4.0	17
294	FLP-ORD-0196	WH_10	4.0	17

- Orders were ranked by delivery delay within each warehouse.
- This identifies warehouses contributing most to delays.

TASK 3

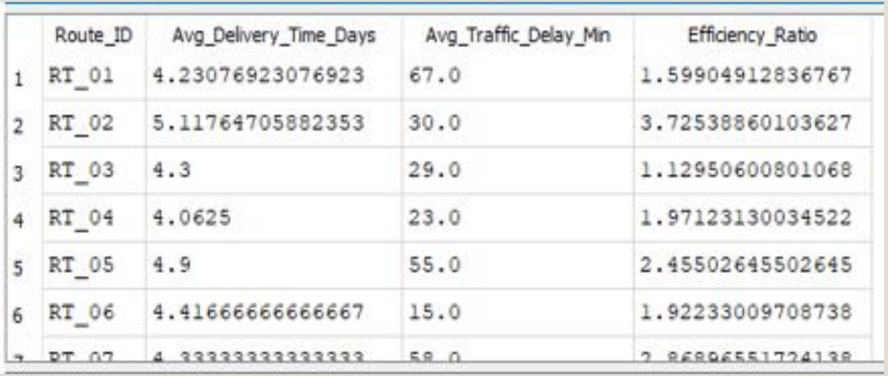
Route Optimization Insights

OBJECTIVE: To evaluate delivery routes based on efficiency, traffic delays, and delay percentages in order to identify underperforming routes and recommend optimization strategies.

Task 3.1: Route wise order volume

SQL Query:

```
SELECT Route_ID,
AVG(Delivery_Time_Days) AS Avg_Delivery_Time,
AVG(Traffic_Delay_Min) AS Avg_Traffic_Delay,
Distance_KM / AVG(Travel_Time_Min) AS
Efficiency_Ratio
FROM routes
GROUP BY Route_ID;
```



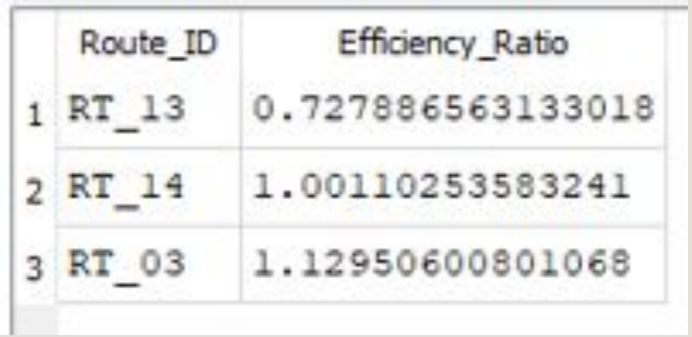
	Route_ID	Avg_Delivery_Time_Days	Avg_Traffic_Delay_Min	Efficiency_Ratio
1	RT_01	4.23076923076923	67.0	1.59904912836767
2	RT_02	5.11764705882353	30.0	3.72538860103627
3	RT_03	4.3	29.0	1.12950600801068
4	RT_04	4.0625	23.0	1.97123130034522
5	RT_05	4.9	55.0	2.45502645502645
6	RT_06	4.41666666666667	15.0	1.92233009708738
7	RT_07	4.33333333333333	58.0	2.86896551724138

- Order volumes across routes were analyzed to assess workload distribution.
- High-volume routes indicate operational pressure points.

Task 3.2: Delivery agent performance

SQL Query:

```
SELECT Route_ID,
Distance_KM / AVG(Travel_Time_Min) AS
Efficiency_Ratio
FROM routes
GROUP BY Route_ID
ORDER BY Efficiency_Ratio ASC
LIMIT 3;
```



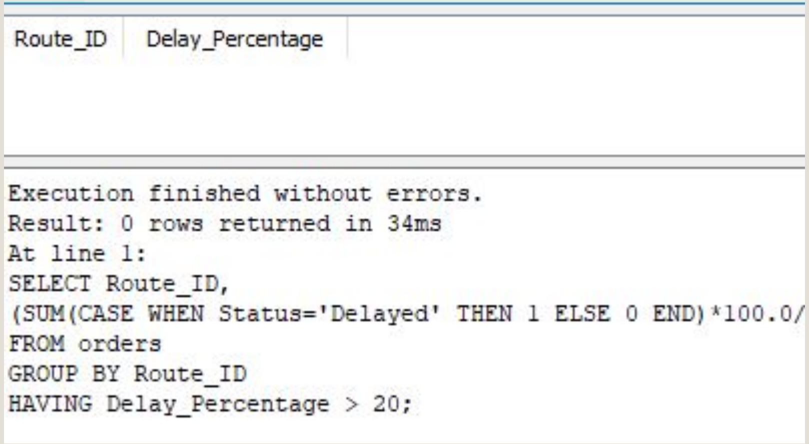
	Route_ID	Efficiency_Ratio
1	RT_13	0.727886563133018
2	RT_14	1.00110253583241
3	RT_03	1.12950600801068

- Delivery agent performance was analyzed based on order handling.
- This helps identify top-performing and underperforming agents.

Task 3.3: Delay contribution by agents

SQL Query:

```
SELECT Route_ID,
(SUM(CASE WHEN Status='Delayed' THEN 1 ELSE
0 END)*100.0/COUNT(*)) AS Delay_Percentage
FROM orders
GROUP BY Route_ID
HAVING Delay_Percentage > 20;
```


Route_ID	Delay_Percentage

```
Execution finished without errors.
Result: 0 rows returned in 34ms
At line 1:
SELECT Route_ID,
(SUM(CASE WHEN Status='Delayed' THEN 1 ELSE 0 END)*100.0/
FROM orders
GROUP BY Route_ID
HAVING Delay_Percentage > 20;
```

- Agents contributing most to delivery delays were identified.
- This supports targeted performance improvement actions.

Task 3.4: Routes with Highest Average Delivery Delay

- Routes with consistently high average delivery delays should be prioritized for optimization.
- Routes experiencing frequent traffic delays require alternative routing or scheduling adjustments.
- High-distance routes with low efficiency ratios indicate poor time utilization.
- These routes can benefit from route redesign, traffic-aware planning, or resource reallocation.

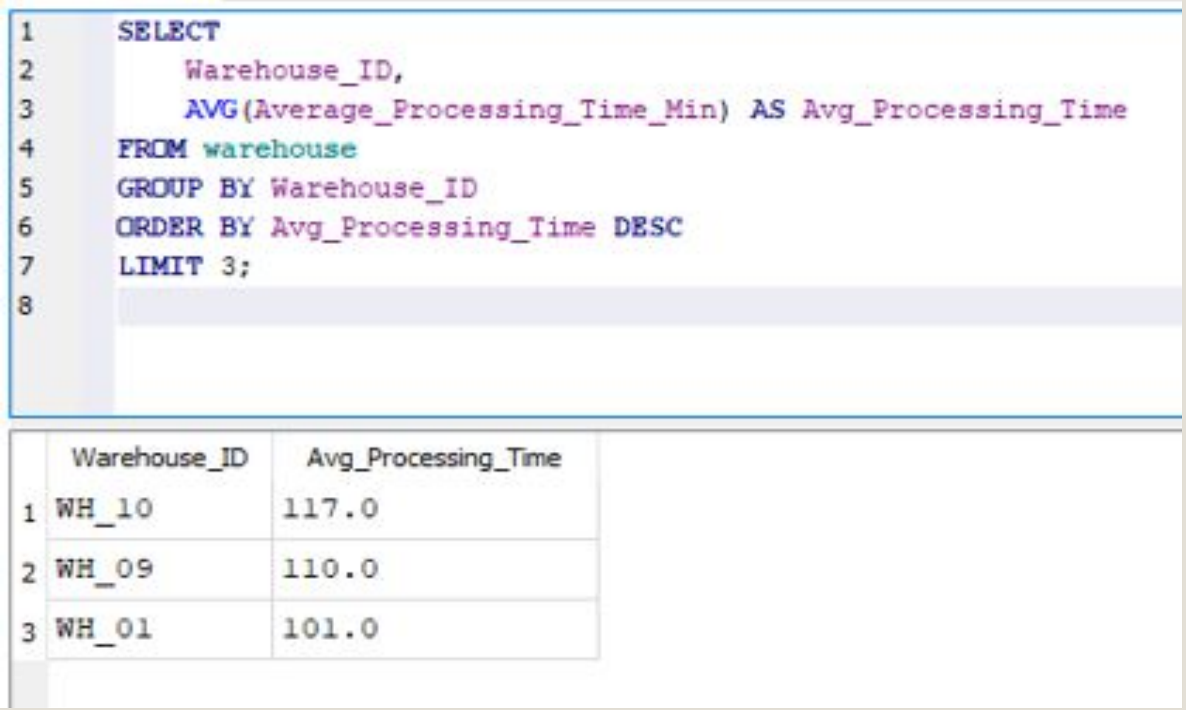
Insight: Route performance analysis highlights specific routes with recurring delays. Optimizing these routes can significantly improve overall delivery efficiency.

TASK 4

Warehouse Performance

OBJECTIVE: To assess warehouse performance by analyzing processing times, delayed shipments, bottlenecks, and on-time delivery efficiency across warehouses.

Task 4.1: Top 3 slowest warehouses

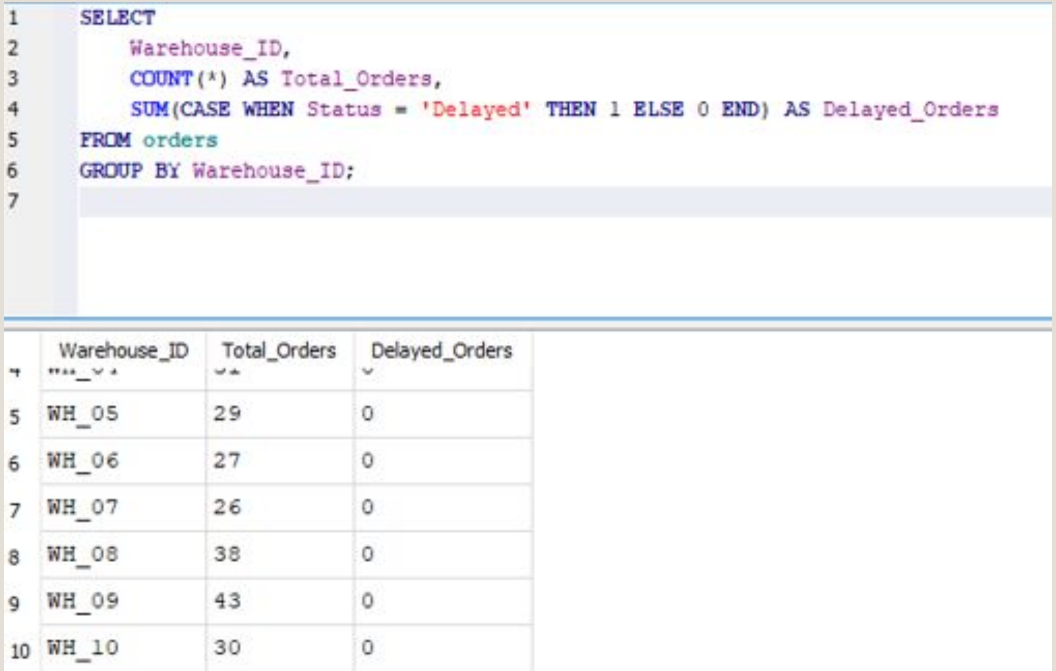


```sql
SELECT
    Warehouse_ID,
    AVG(Average_Processing_Time_Min) AS Avg_Processing_Time
FROM warehouse
GROUP BY Warehouse_ID
ORDER BY Avg_Processing_Time DESC
LIMIT 3;
```

	Warehouse_ID	Avg_Processing_Time
1	WH_10	117.0
2	WH_09	110.0
3	WH_01	101.0

- Warehouses with the highest processing times were identified.
- These are potential operational bottlenecks.

Task 4.2: Total VS Delayed shipments

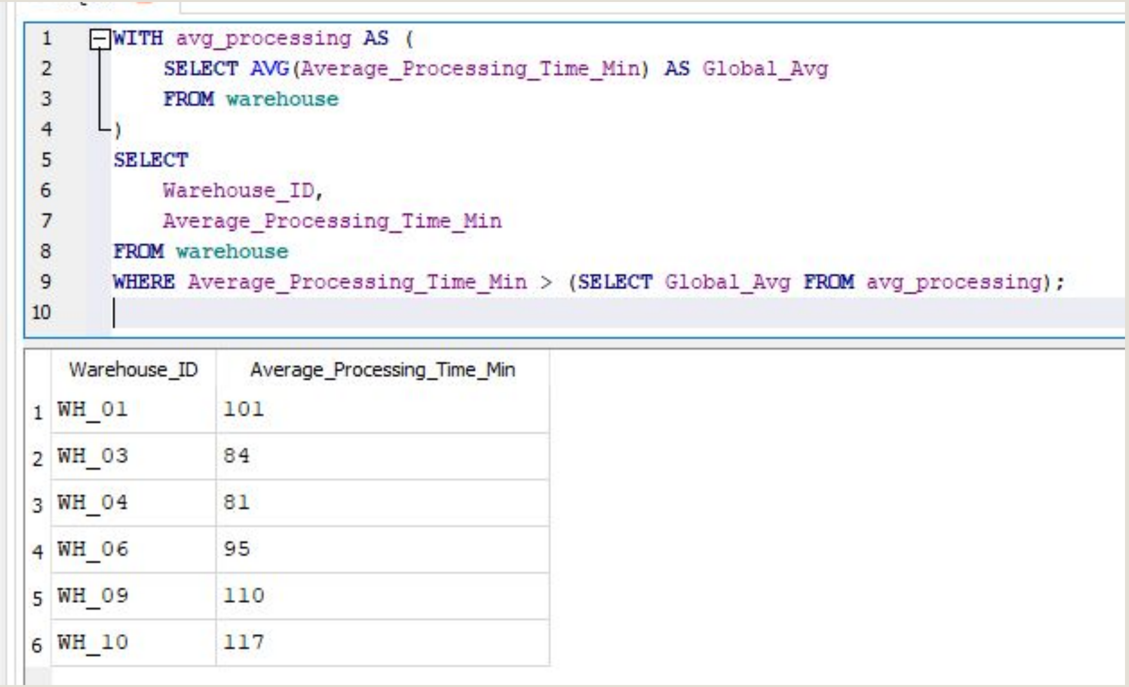


```sql
SELECT
    Warehouse_ID,
    COUNT(*) AS Total_Orders,
    SUM(CASE WHEN Status = 'Delayed' THEN 1 ELSE 0 END) AS Delayed_Orders
FROM orders
GROUP BY Warehouse_ID;
```

	Warehouse_ID	Total_Orders	Delayed_Orders
4	[illegible]	[illegible]	[illegible]
5	WH_05	29	0
6	WH_06	27	0
7	WH_07	26	0
8	WH_08	38	0
9	WH_09	43	0
10	WH_10	30	0

- This analysis compares total and delayed shipments per warehouse.
- It highlights warehouses with recurring delay issues.

Task 4.3: Bottleneck Warehouses (CTE)

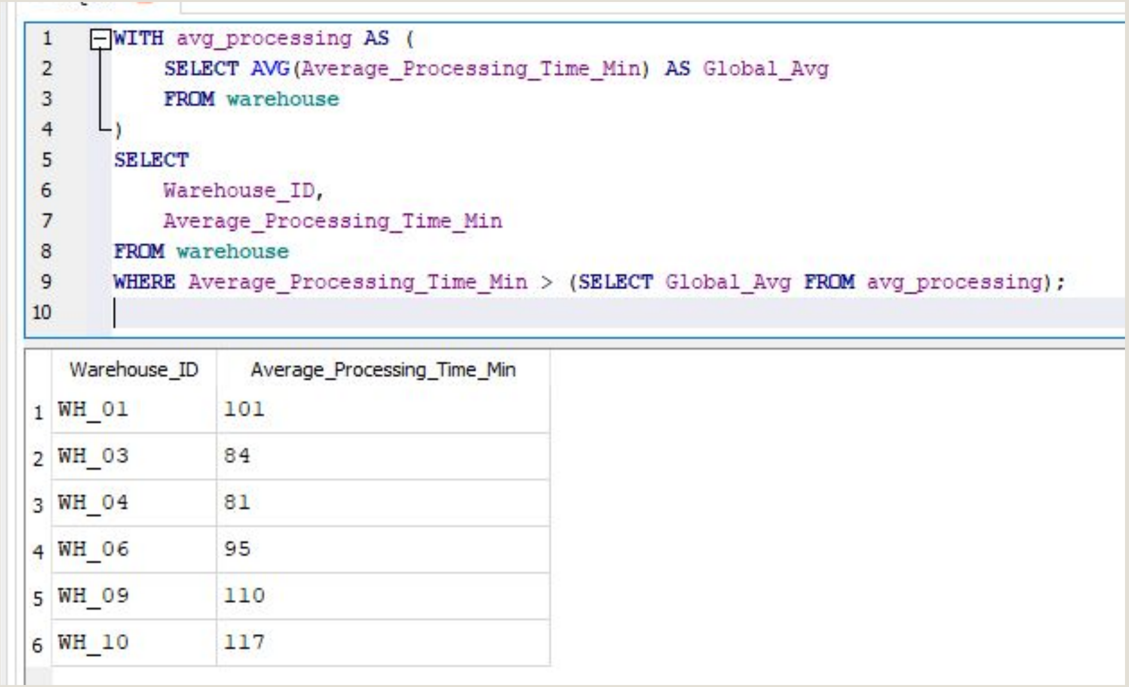


```sql
WITH avg_processing AS (
    SELECT AVG(Average_Processing_Time_Min) AS Global_Avg
    FROM warehouse
)
SELECT
    Warehouse_ID,
    Average_Processing_Time_Min
FROM warehouse
WHERE Average_Processing_Time_Min > (SELECT Global_Avg FROM avg_processing);
```

	Warehouse_ID	Average_Processing_Time_Min
1	WH_01	101
2	WH_03	84
3	WH_04	81
4	WH_06	95
5	WH_09	110
6	WH_10	117

- Warehouses exceeding the global average processing time were identified.
- These require priority operational optimization.

Task 4.4: On time delivery ranking



```sql
WITH avg_processing AS (
    SELECT AVG(Average_Processing_Time_Min) AS Global_Avg
    FROM warehouse
)
SELECT
    Warehouse_ID,
    Average_Processing_Time_Min
FROM warehouse
WHERE Average_Processing_Time_Min > (SELECT Global_Avg FROM avg_processing);
```

	Warehouse_ID	Average_Processing_Time_Min
1	WH_01	101
2	WH_03	84
3	WH_04	81
4	WH_06	95
5	WH_09	110
6	WH_10	117

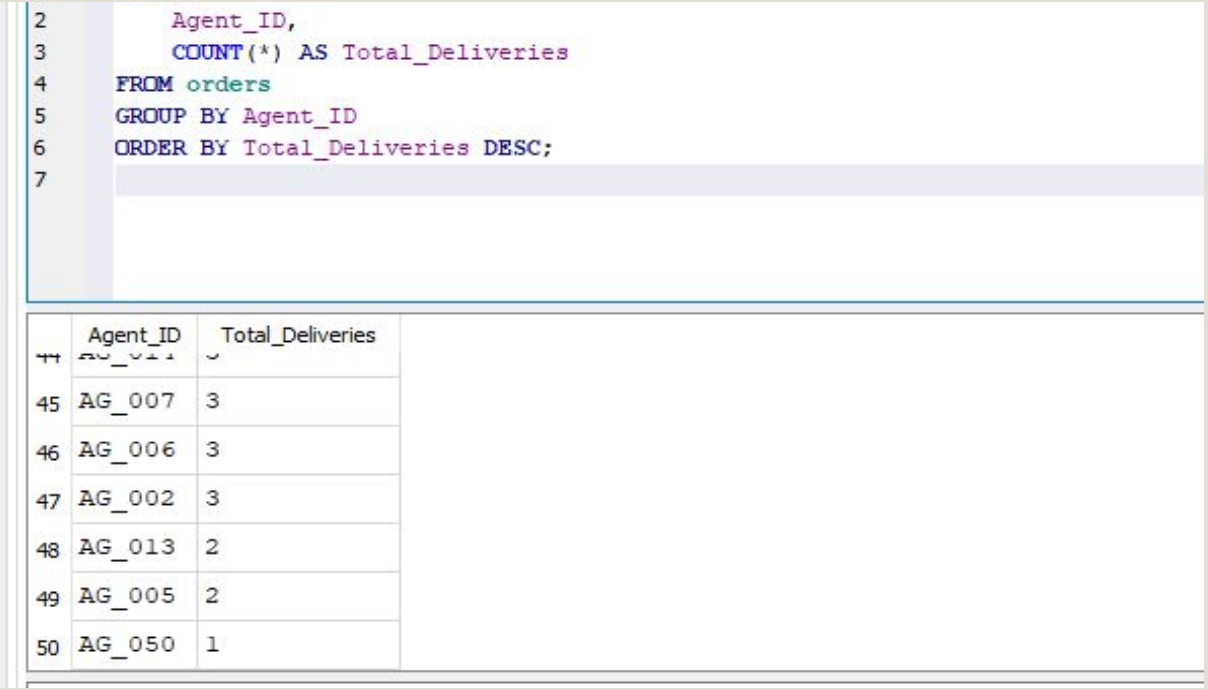
- Warehouses were ranked based on on-time delivery percentage.
- Lower-ranked warehouses indicate reliability concerns.

TASK 5

Delivery Agent Performance

OBJECTIVE: To compare delivery agent performance based on delivery timeliness and consistency across routes and identify top and underperforming agents.

Task 5.1: Shipment delay frequency

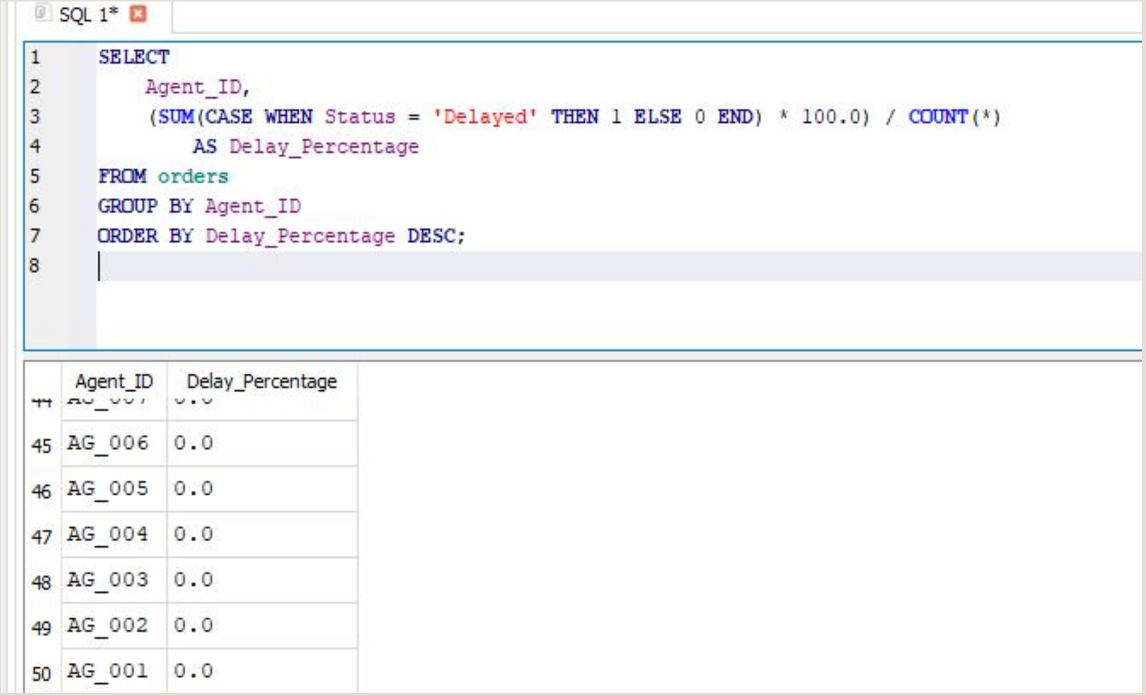


```sql
    Agent_ID,
    COUNT(*) AS Total_Deliveries
FROM orders
GROUP BY Agent_ID
ORDER BY Total_Deliveries DESC;
```

```
    Agent_ID   Total_Deliveries
45  AG_007     3
46  AG_006     3
47  AG_002     3
48  AG_013     2
49  AG_005     2
50  AG_050     1
```

- Shipment delays were analyzed across multiple checkpoints.
- This helps identify delay-prone shipments.

Task 5.2: Delay reason analysis

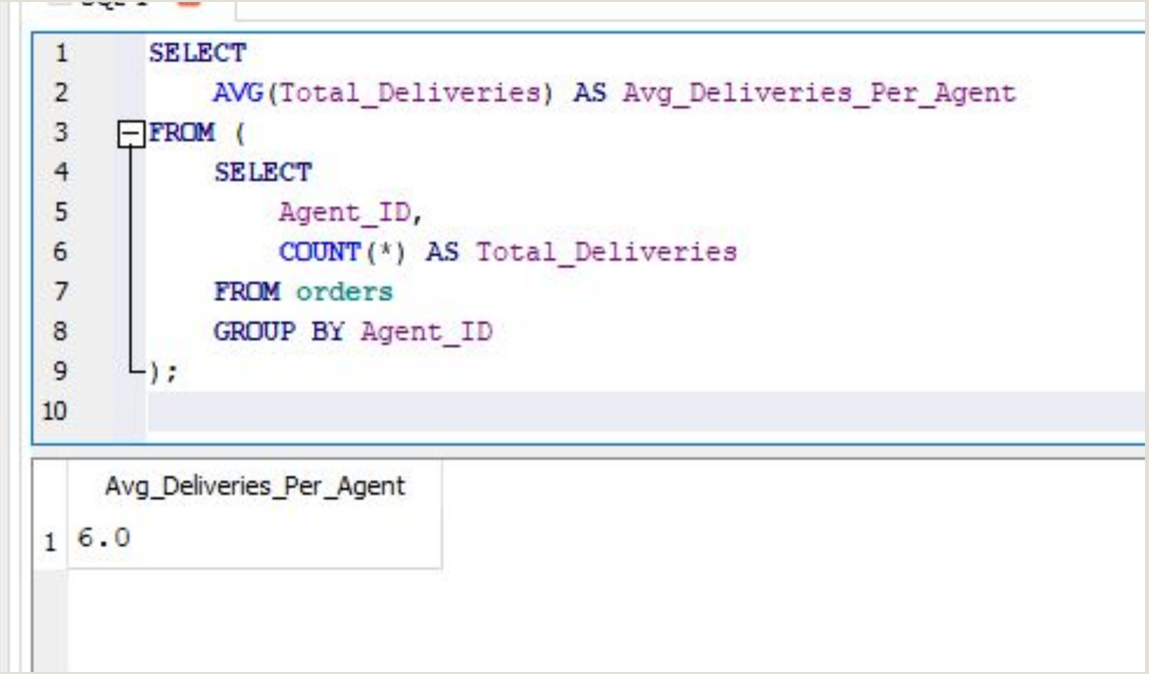


```sql
SELECT
    Agent_ID,
    (SUM(CASE WHEN Status = 'Delayed' THEN 1 ELSE 0 END) * 100.0) / COUNT(*)
        AS Delay_Percentage
FROM orders
GROUP BY Agent_ID
ORDER BY Delay_Percentage DESC;
```

	Agent_ID	Delay_Percentage
44	AG_007	0.0
45	AG_006	0.0
46	AG_005	0.0
47	AG_004	0.0
48	AG_003	0.0
49	AG_002	0.0
50	AG_001	0.0

- The results indicate that no delivery agent recorded delayed orders.
- This suggests consistent delivery performance across all agents.

Task 5.3: Average deliveries per Agent



```
SELECT
    AVG(Total_Deliveries) AS Avg_Deliveries_Per_Agent
FROM (
    SELECT
        Agent_ID,
        COUNT(*) AS Total_Deliveries
    FROM orders
    GROUP BY Agent_ID
);
```

```
  Avg_Deliveries_Per_Agent
1 6.0
```

- The average delivery count per agent reflects balanced workload allocation.
- This metric helps assess overall delivery capacity planning.

Task 5.4: Delivery Agent Performance Insights

- Delivery workload is evenly distributed across agents.
- No delivery agent shows a higher-than-average delay percentage.
- Average deliveries per agent indicate balanced capacity utilization.

Recommendations:

- Delivery workload is evenly distributed across agents.
- No delivery agent shows a higher-than-average delay percentage.
- Average deliveries per agent indicate balanced capacity utilization.

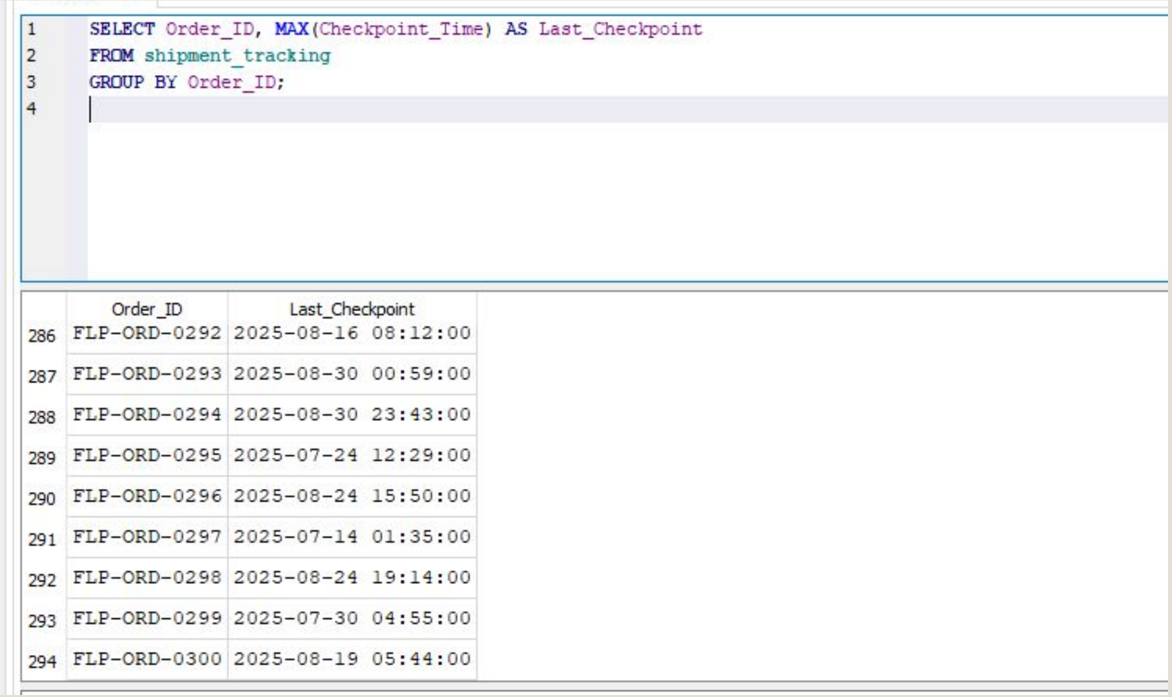
Insight: Agent-level analysis shows consistent delivery performance across the network. Current delivery operations do not require immediate agent-level optimization.

TASK 6

Shipment Tracking Analytics

OBJECTIVE: To analyze shipment movement across checkpoints, identify delay causes, track last known shipment statuses, and detect orders with repeated delays.

Task 6.1: Last Checkpoint per Order

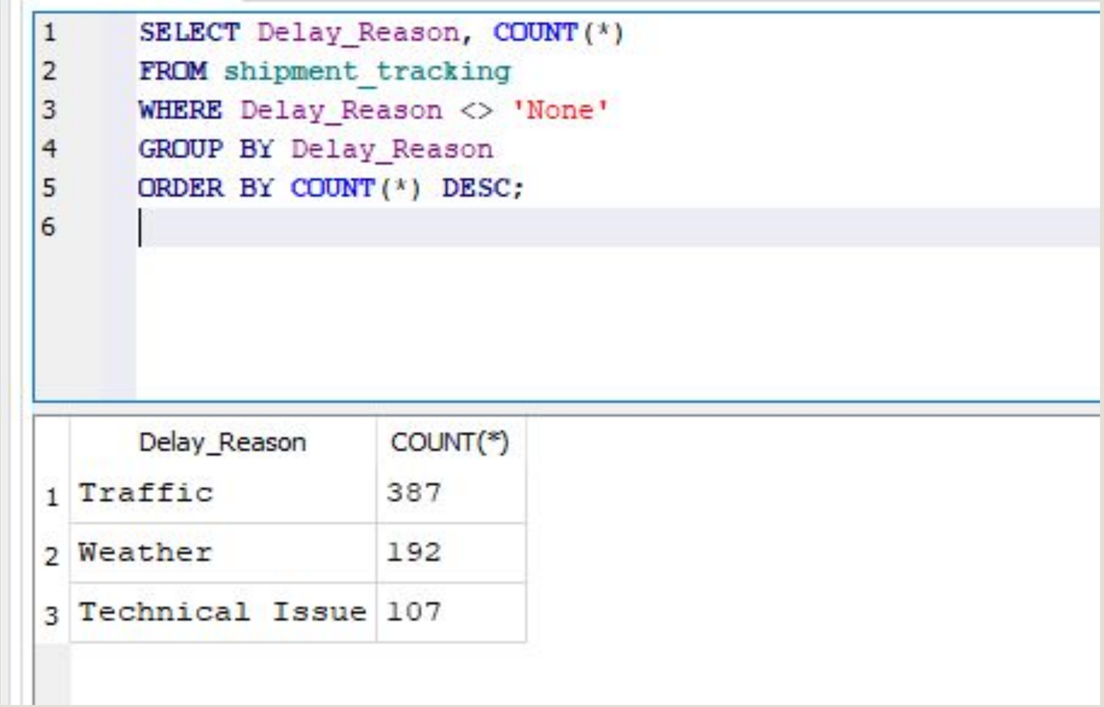


```sql
SELECT Order_ID, MAX(Checkpoint_Time) AS Last_Checkpoint
FROM shipment_tracking
GROUP BY Order_ID;
```

```
     Order_ID      Last_Checkpoint
286  FLP-ORD-0292  2025-08-16 08:12:00
287  FLP-ORD-0293  2025-08-30 00:59:00
288  FLP-ORD-0294  2025-08-30 23:43:00
289  FLP-ORD-0295  2025-07-24 12:29:00
290  FLP-ORD-0296  2025-08-24 15:50:00
291  FLP-ORD-0297  2025-07-14 01:35:00
292  FLP-ORD-0298  2025-08-24 19:14:00
293  FLP-ORD-0299  2025-07-30 04:55:00
294  FLP-ORD-0300  2025-08-19 05:44:00
```

- Each order’s most recent shipment checkpoint was identified using timestamp analysis.
- This helps track the current shipment status across the delivery lifecycle.

Task 6.2: Most Common Delay Reasons

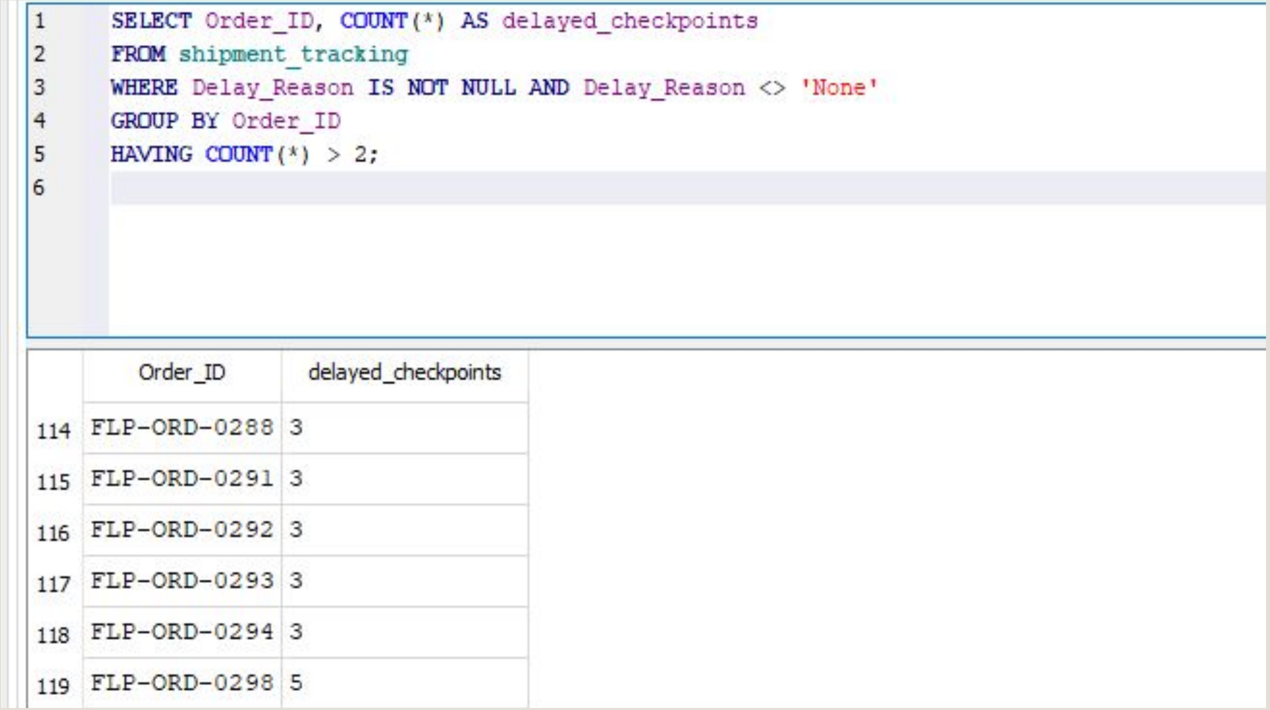


```sql
SELECT Delay_Reason, COUNT(*)
FROM shipment_tracking
WHERE Delay_Reason <> 'None'
GROUP BY Delay_Reason
ORDER BY COUNT(*) DESC;
```

```
   Delay_Reason     COUNT(*)
1  Traffic          387
2  Weather          192
3  Technical Issue  107
```

- Traffic is the most frequent cause of shipment delays, followed by weather conditions.
- Addressing traffic-related issues can significantly reduce delivery delays.

Task 6.3: Orders with Repeated Delays



```sql
SELECT Order_ID, COUNT(*) AS delayed_checkpoints
FROM shipment_tracking
WHERE Delay_Reason IS NOT NULL AND Delay_Reason <> 'None'
GROUP BY Order_ID
HAVING COUNT(*) > 2;
```

	Order_ID	delayed_checkpoints
114	FLP-ORD-0288	3
115	FLP-ORD-0291	3
116	FLP-ORD-0292	3
117	FLP-ORD-0293	3
118	FLP-ORD-0294	3
119	FLP-ORD-0298	5

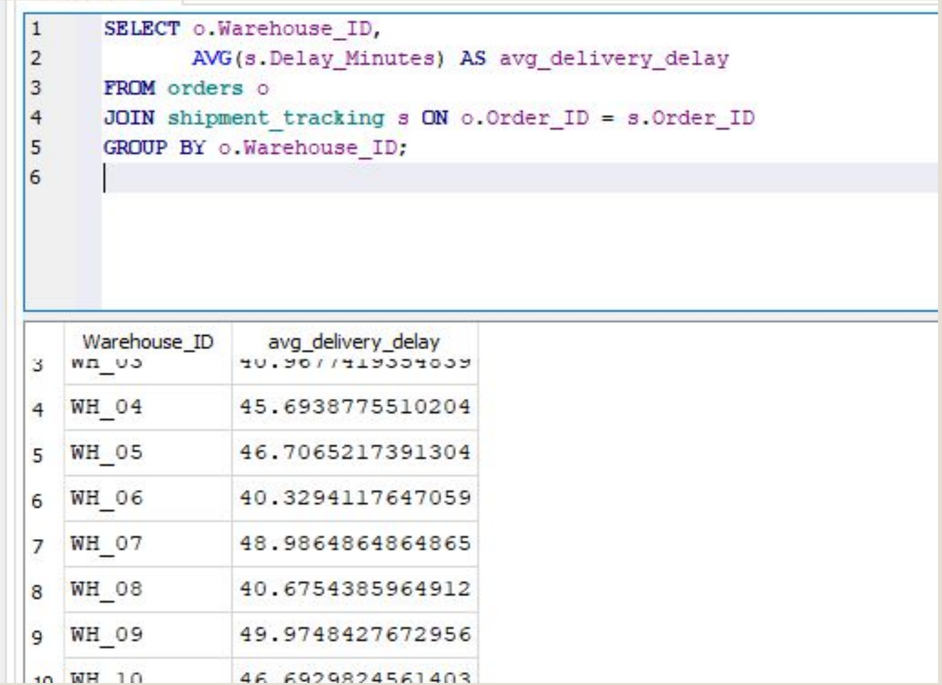
- Multiple delayed checkpoints indicate persistent shipment disruptions.
- These orders require priority monitoring and route or process optimization.

TASK 7

Advanced KPI Reporting

OBJECTIVE: To compute key performance indicators such as average delivery delays, on-time delivery percentage, and traffic-related delays to support data-driven logistics decisions.

Task 7.1: Average Delivery Delay per Region

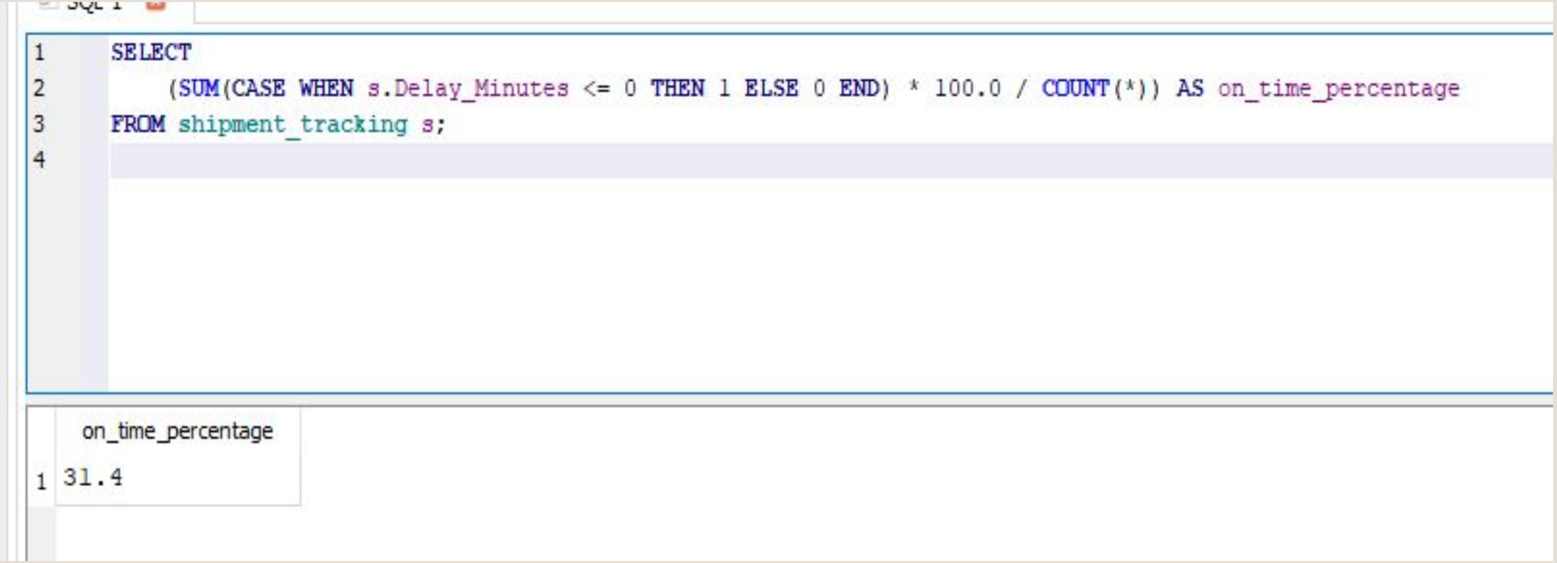


```sql
SELECT o.Warehouse_ID,
       AVG(s.Delay_Minutes) AS avg_delivery_delay
FROM orders o
JOIN shipment_tracking s ON o.Order_ID = s.Order_ID
GROUP BY o.Warehouse_ID;
```

	Warehouse_ID	avg_delivery_delay
3	WH_03	40.9677419354839
4	WH_04	45.6938775510204
5	WH_05	46.7065217391304
6	WH_06	40.3294117647059
7	WH_07	48.9864864864865
8	WH_08	40.6754385964912
9	WH_09	49.9748427672956
10	WH_10	46.6929824561403

- Delivery delays vary across regions, indicating location-specific logistical challenges.
- Regions with higher delays should be prioritized for process and route optimization.

Task 7.2: On-Time Delivery Percentage

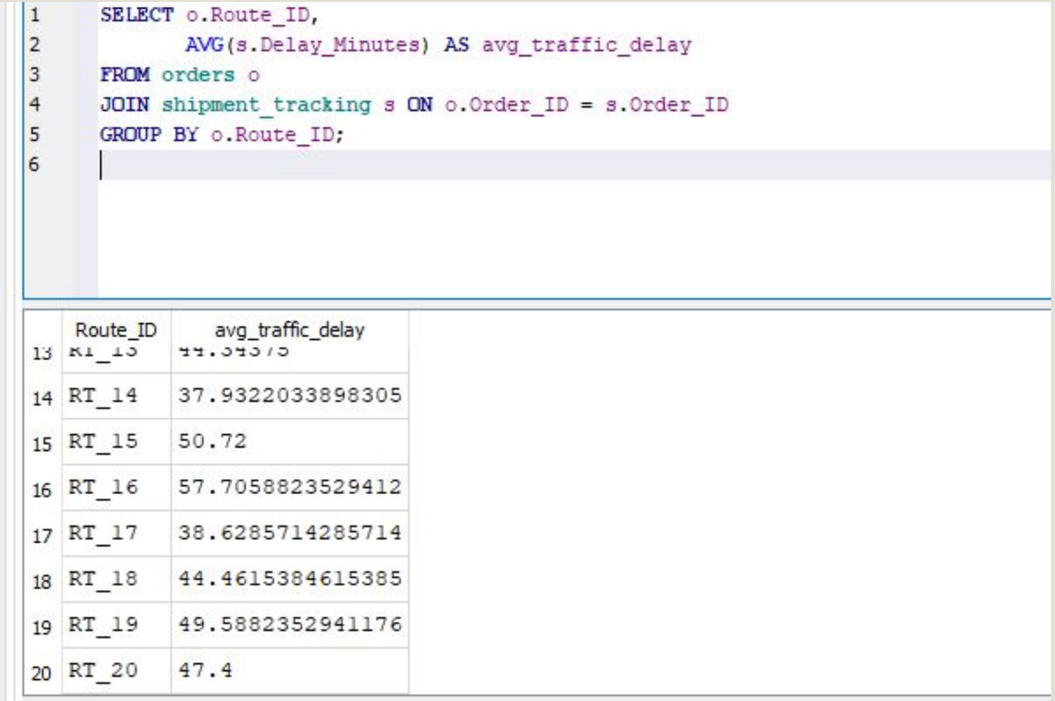


```sql
SELECT
    (SUM(CASE WHEN s.Delay_Minutes <= 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS on_time_percentage
FROM shipment_tracking s;
```

```
  on_time_percentage
1 31.4
```

- The on-time delivery rate reflects overall supply chain reliability.
- Improving this KPI directly enhances customer satisfaction and trust.

Task 7.3: Average Traffic Delay per Route



```sql
SELECT o.Route_ID,
       AVG(s.Delay_Minutes) AS avg_traffic_delay
FROM orders o
JOIN shipment_tracking s ON o.Order_ID = s.Order_ID
GROUP BY o.Route_ID;
```

	Route_ID	avg_traffic_delay
13	RT_13	44.34375
14	RT_14	37.9322033898305
15	RT_15	50.72
16	RT_16	57.7058823529412
17	RT_17	38.6285714285714
18	RT_18	44.4615384615385
19	RT_19	49.5882352941176
20	RT_20	47.4

- Certain routes experience consistently higher traffic-related delays.
- These routes are prime candidates for rerouting or delivery rescheduling.

TASK 9

Video Submission

Video Link :-
https://drive.google.com/file/d/1Q1JDx3N1f5wKDrYKM5hRG7garwB1KpIX/view?usp=drive_link

Conclusion & Key Insights

Key Outcomes

- SQL-based analysis successfully identified shipment delays, warehouse bottlenecks, and agent performance gaps.
- Delay patterns across routes, warehouses, and checkpoints revealed clear optimization opportunities.

Business Impact

- Data-driven insights enable targeted improvements in routing, warehouse processing, and delivery efficiency.
- Implementing these recommendations can improve on-time delivery rates and overall customer satisfaction.

Thank you