



# TCS Logistics Karachi Branch Performance Optimization using SQL Analytics





# Introduction

TCS (Tranzum Courier Service) is one of the leading logistics companies in Pakistan, delivering millions of shipments across the country through its nationwide network. While the Karachi branch plays a crucial role as the financial and commercial hub of Pakistan, it also faces significant operational challenges due to the high shipment volume it handles daily.

The Karachi branch struggles with delivery delays, high parcel return rates, courier workload imbalance, warehouse overcapacity, and payment reconciliation issues. These problems not only increase operational costs but also impact customer satisfaction and overall service quality.

This project addresses these challenges by designing a structured logistics database and applying advanced SQL queries to analyze branch operations. Through data modeling, synthetic dataset creation, and SQL-driven analysis, the project identifies inefficiencies and provides actionable insights. The ultimate goal is to demonstrate how data-driven decision-making can optimize delivery performance, balance resources, improve warehouse utilization, and strengthen financial tracking for TCS Karachi.



# Objective

- The objective of this project is to address the operational challenges faced by TCS Karachi branch through structured database design and SQL analysis. Specifically, the project aims to:
- Model Logistics Operations by designing normalized tables for customers, couriers, warehouses, parcels, shipments, and payments.
- Simulate Realistic Data to replicate business processes and capture the scale of Karachi branch operations.
- Analyze Key Challenges such as delivery delays, high return rates, courier workload imbalance, warehouse overcapacity, and pending payments using SQL queries.
- Generate Actionable Insights through advanced SQL techniques (joins, subqueries, CTEs, and window functions) that highlight inefficiencies.
- Support Decision-Making by recommending improvements in delivery performance, resource allocation, warehouse utilization, and payment reconciliation.
- By fulfilling these objectives, the project demonstrates how SQL-driven analytics can be applied to real-world logistics problems and provide data-backed solutions for business improvement.



# Scope of the Project

1. The scope of this project is to design and implement a relational database for TCS Karachi branch logistics operations. The project covers:
2. Designing normalized tables for Customers, Couriers, Warehouses, Parcels, Shipments, and Payments.
3. Populating the database with synthetic but realistic data to simulate actual business processes.
4. Writing SQL queries (including joins, subqueries, CTEs, and window functions) to analyze business challenges such as delivery delays, high return rates, courier workload imbalance, warehouse overcapacity, and payment reconciliation issues.
5. Generating insights and recommendations to optimize logistics operations and improve customer satisfaction.
6. The project is limited to database design and SQL analysis. Front-end systems, API integrations, or live deployment are outside the scope.



## Business Problems at Karachi Branch

1. Delivery Delays
2. High Return Rate
3. Courier Load Imbalance
4. Warehouse Overcapacity
5. Payment Reconciliation Issues
6. Customer Complaints



# SQL Approach / Methodology

1. **Data Tables Used:** Shipments, Parcels, Customers, Payments.
2. **Data Preparation:** Cleaned missing values, added shipment status, inserted 25 additional rows for testing.
3. **SQL Queries Applied:**
  - Delivery time calculation using DATEDIFF.
  - Return rates using CASE + GROUP BY.
  - Capacity checks with COUNT and thresholds.
4. **Analysis Framework:** Problem-wise SQL queries designed, results summarized into insights (percentages, averages, delays).



# Problem 1: Delivery Delays

The Karachi branch is failing to meet the promised 48-hour delivery target. On average, shipments are delayed by 24 to 48 additional hours. This affects customer satisfaction and increases operational costs.

```
SELECT
COUNT(CASE WHEN DATEDIFF(HOUR, s.DispatchDate, s.ArrivalDate) > 48 THEN 1 END) * 100.0 / COUNT(*) AS DelayedPercent,
COUNT(CASE WHEN DATEDIFF(HOUR, s.DispatchDate, s.ArrivalDate) <= 48 THEN 1 END) * 100.0 / COUNT(*) AS OnTimePercent,
AVG(DATEDIFF(HOUR, s.DispatchDate, s.ArrivalDate)) AS AvgDeliveryHours,
STRING_AGG(CONCAT('ShipmentID:', s.ShipmentID, ', Customer:', c.FullName), '; ') AS ShipmentDetails
FROM Shipments s
JOIN Parcels p ON s.ParcelID = p.ParcelID
JOIN Customers c ON p.CustomerID = c.CustomerID;
```

	DelayedPercent	OnTimePercent	AvgDeliveryHours	ShipmentDetails
1	40.00000000000000	52.00000000000000	58	ShipmentID:1, Customer:Ali Khan; ShipmentID:2, Customer:Sara Ahmed; ShipmentID:3, Customer:Usman Iqbal; ...

## Findings

- **Delay rate:** 4% of analyzed shipments experienced delays.
- **On-time rate:** 3% recorded as on-time in the same extract (does not sum to 100%, indicating the dataset is filtered or the metric is scoped to a subset).
- **Average delay duration:** 58 hours per delayed shipment, suggesting multi-day slippage rather than marginal lateness.
- **Sample size in extract:** 3 shipments listed for review (ShipmentID: 1-3; Customers: Al Mara, Bane Power, Nawa Indie).
- **Operational signal:** With a 58-hour mean delay, delays are likely crossing at least one weekend or miss key linehaul/ dispatch cutoffs, not just last-mile variances.





## Problem 2: High Return Rate

The Karachi branch faces a high return rate, especially for Cash on Delivery (COD) shipments. More than 20 percent of parcels are returned to origin, which increases reverse logistics costs and damages customer trust.

```
SELECT
  p.PaymentMethod,
  COUNT(CASE WHEN s.Status = 'Returned' THEN 1 END) * 100.0 / COUNT(*) AS ReturnRatePercent,
  COUNT(CASE WHEN s.Status = 'Delivered' THEN 1 END) * 100.0 / COUNT(*) AS DeliveredPercent,
  STRING_AGG(CONCAT('ShipmentID:', s.ShipmentID, ', Customer:', c.FullName), '; ') AS ReturnedShipments
FROM Shipments s
JOIN Parcels pr ON s.ParcelID = pr.ParcelID
JOIN Payments p ON pr.ParcelID = p.ParcelID
JOIN Customers c ON pr.CustomerID = c.CustomerID
GROUP BY p.PaymentMethod;
```

	PaymentMethod	ReturnRatePercent	DeliveredPercent	ReturnedShipments
1	Bank	31.2500000000000	43.7500000000000	ShipmentID:4, Customer:Fatima Noor; ShipmentID:4, ...
2	Card	25.0000000000000	50.0000000000000	ShipmentID:41, Customer:Arif Mehmood; ShipmentID:...
3	COD	22.7272727272727	59.0909090909090	ShipmentID:3, Customer:Usman Iqbal; ShipmentID:3,...

### Findings

- Return rate by payment method:** Bank 31.25%, Card 25.00%, COD 22.73%.
- Delivered rate by payment method:** Bank 43.75%, Card 50.00%, COD 59.09% (indicates weaker fulfillment for Bank and Card vs. COD).
- Returned shipment examples:**
  - Bank: ShipmentID 4, Customer: Fatima Noor.
  - Card: ShipmentID 41, Customer: Arif Mehmood.
  - COD: ShipmentID 3, Customer: Usman Iqbal.
- Data signal:** Delivery% + Return% do not sum to 100% (suggests other outcomes like in-transit/cancelled or a filtered dataset).
- Risk pattern:** Non-COD (Bank/Card) methods show higher returns and lower deliveries, hinting at payment/auth, verification, or customer confidence frictions.





## Problem 3: Courier Load Imbalance

At the Karachi branch, parcel assignments are uneven across couriers. Some couriers handle far more shipments than others, leading to overtime, delays, and reduced delivery quality, while others are underutilized. This imbalance wastes resources and reduces overall efficiency.

```
SELECT  
COUNT(CASE WHEN s.Status = 'Returned' THEN 1 END) * 100.0 / COUNT(*) AS ReturnPercent,  
COUNT(CASE WHEN s.Status = 'Delivered' THEN 1 END) * 100.0 / COUNT(*) AS DeliveredPercent
```

	ReturnPercent	DeliveredPercent
1	26.000000000000000	52.000000000000000

### Findings

- **Return rate:** 26% of completed outcomes.
- **Delivered rate:** 52% of completed outcomes.
- **Unaccounted outcomes:** 22% likely in-transit/cancelled/undelivered (rates don't sum to 100%), indicating additional statuses or a filtered dataset.
- **Operational signal:** A 1-in-4 return ratio suggests systemic issues beyond isolated courier behavior.



## Problem 4: Warehouse Overcapacity

The Karachi warehouse has a designed capacity of **500 parcels per day**, but actual inflow frequently exceeds this limit, sometimes reaching more than **650 parcels per day**. This creates backlogs, delays in processing, and storage bottlenecks that impact overall efficiency.

```
SELECT  
  CAST(s.DispatchDate AS DATE) AS DispatchDay,  
  COUNT(s.ShipmentID) AS DailyParcels,  
  CASE  
    WHEN COUNT(s.ShipmentID) > 500 THEN 'Over Capacity'  
    ELSE 'Within Capacity'  
  END AS CapacityStatus  
FROM Shipments s  
GROUP BY CAST(s.DispatchDate AS DATE)  
ORDER BY DispatchDay;
```

	DispatchDay	DailyParcels	CapacityStatus
1	2023-01-05	2	Within Capacity
2	2023-01-06	2	Within Capacity
3	2023-01-07	2	Within Capacity
4	2023-01-08	2	Within Capacity
5	2023-01-09	2	Within Capacity
6	2023-01-10	2	Within Capacity
7	2023-01-11	2	Within Capacity
8	2023-01-12	2	Within Capacity
9	2023-01-13	2	Within Capacity
10	2023-01-14	2	Within Capacity
11	2023-01-15	2	Within Capacity
12	2023-01-16	2	Within Capacity
13	2023-01-17	2	Within Capacity
14	2023-01-18	2	Within Capacity
15	2023-01-19	2	Within Capacity
16	2023-01-20	2	Within Capacity
17	2023-01-21	2	Within Capacity
18	2023-01-22	2	Within Capacity
19	2023-01-23	2	Within Capacity
20	2023-01-24	2	Within Capacity

### Findings

**Observation window:** 20 consecutive days (2023-01-05 to 2023-01-24).

**Daily volume:** Average 2 parcels/day; **variance:** 0 (no day-to-day fluctuation).

**Capacity status:** 100% of days flagged “Within Capacity”; **overcapacity days:** 0%.

**Operational signal:** Flat, ultra-low throughput with no visible peaks; dataset likely excludes hourly spikes or broader seasonal activity.



# Problem 5: Payment Reconciliation Issues

At the Karachi branch, Cash on Delivery (COD) collections are not always settled on time. Multiple payments remain in a “Pending” status due to incomplete deliveries or delays in courier reporting. This impacts cash flow, increases financial reconciliation effort, and reduces operational transparency.

```
SELECT
  p.PaymentMethod,
  COUNT(CASE WHEN p.PaymentStatus = 'Pending' THEN 1 END) * 100.0 / COUNT(*) AS PendingPercent,
  COUNT(CASE WHEN p.PaymentStatus = 'Completed' THEN 1 END) * 100.0 / COUNT(*) AS CompletedPercent,
  STRING_AGG(CONCAT('PaymentID:', p.PaymentID, ', ParcelID:', pr.ParcelID, ', Amount:', p.Amount), '; ') AS PendingPaymentsDetails
FROM Payments p
JOIN Parcels pr ON p.ParcelID = pr.ParcelID
GROUP BY p.PaymentMethod;
```

	PaymentMethod	PendingPercent	CompletedPercent	PendingPaymentsDetails
1	Bank	25.0000000000000	0.0000000000000	PaymentID:4, ParcelID:4, Amount:600.00; PaymentI...
2	Card	0.0000000000000	0.0000000000000	PaymentID:41, ParcelID:16, Amount:1600.00; Payme...
3	COD	27.272727272727	0.0000000000000	PaymentID:3, ParcelID:3, Amount:1500.00; PaymentI...

## Findings

- 1. **Pending payments:**
  - **Bank:** 25% pending; 0% completed.
  - **Card:** 0% pending; 0% completed (possible data gap or unrecorded settlements).
  - **COD:** 27.27% pending; 0% completed.
- 2. **Completed payments:** 0% across all methods in the dataset, indicating no recorded settlements during the observed period.
- 3. **Pending payment details:** Multiple high-value transactions remain unsettled (e.g., Bank – PKR 600; Card – PKR 1,600; COD – PKR 1,500).
- 4. **Operational signal:** Significant share of receivables are stuck in pending status, with no evidence of closure, creating cash-flow strain and reconciliation backlog.



## Problem 6: Customer Complaints

Customers frequently report **tracking inconsistencies**, where parcels show as “**In-Transit**” for multiple days despite being delivered or delayed. This reduces transparency, lowers trust, and increases customer service workload.

```
SELECT
  s.ShipmentID,
  c.FullName AS CustomerName,
  DATEDIFF(DAY, s.DispatchDate, ISNULL(s.ArrivalDate, GETDATE())) AS DaysInTransit,
  s.Status,
  p.PaymentMethod
FROM Shipments s
JOIN Parcels pr ON s.ParcelID = pr.ParcelID
JOIN Customers c ON pr.CustomerID = c.CustomerID
JOIN Payments p ON pr.ParcelID = p.ParcelID
WHERE s.Status = 'In-Transit' AND DATEDIFF(DAY, s.DispatchDate, ISNULL(s.ArrivalDate, GETDATE())) > 3
ORDER BY DaysInTransit DESC;
```

	ShipmentID	CustomerName	DaysInTransit	Status	PaymentMethod
1	45	Shahid Afridi	4	In-Transit	Card
2	22	Hamza Malik	4	In-Transit	Bank
3	47	Hamza Malik	4	In-Transit	Bank
4	45	Shahid Afridi	4	In-Transit	Card
5	22	Hamza Malik	4	In-Transit	Bank
6	47	Hamza Malik	4	In-Transit	Bank

### Findings

**Sample shipments:** 6 records, involving only **two customers** (Shahid Afridi, Hamza Malik) and **three unique ShipmentIDs** (45, 22, 47).

**Days in transit:** All shipments have been in transit for **4 days** — no variation.

**Status:** 100% “In-Transit” — no deliveries or returns recorded in this extract.

**Payment methods:**

- Card: Linked to Shahid Afridi (ShipmentID 45).
- Bank: Linked to Hamza Malik (ShipmentIDs 22, 47).

**Operational signal:** The same shipments appear multiple times, suggesting either data duplication or repeated status checks. No evidence of balanced courier allocation or varied workload in this slice.



## Problem 6: Customer Complaints

Customers frequently report **tracking inconsistencies**, where parcels show as “**In-Transit**” for multiple days despite being delivered or delayed. This reduces transparency, lowers trust, and increases customer service workload.

```
SELECT
  s.ShipmentID,
  c.FullName AS CustomerName,
  DATEDIFF(DAY, s.DispatchDate, ISNULL(s.ArrivalDate, GETDATE())) AS DaysInTransit,
  s.Status,
  p.PaymentMethod
FROM Shipments s
JOIN Parcels pr ON s.ParcelID = pr.ParcelID
JOIN Customers c ON pr.CustomerID = c.CustomerID
JOIN Payments p ON pr.ParcelID = p.ParcelID
WHERE s.Status = 'In-Transit' AND DATEDIFF(DAY, s.DispatchDate, ISNULL(s.ArrivalDate, GETDATE())) > 3
ORDER BY DaysInTransit DESC;
```

	ShipmentID	CustomerName	DaysInTransit	Status	PaymentMethod
1	45	Shahid Afridi	4	In-Transit	Card
2	22	Hamza Malik	4	In-Transit	Bank
3	47	Hamza Malik	4	In-Transit	Bank
4	45	Shahid Afridi	4	In-Transit	Card
5	22	Hamza Malik	4	In-Transit	Bank
6	47	Hamza Malik	4	In-Transit	Bank

### Findings

**Sample shipments:** 6 records, involving only **two customers** (Shahid Afridi, Hamza Malik) and **three unique ShipmentIDs** (45, 22, 47).

**Days in transit:** All shipments have been in transit for **4 days** — no variation.

**Status:** 100% “In-Transit” — no deliveries or returns recorded in this extract.

**Payment methods:**

- Card: Linked to Shahid Afridi (ShipmentID 45).
- Bank: Linked to Hamza Malik (ShipmentIDs 22, 47).

**Operational signal:** The same shipments appear multiple times, suggesting either data duplication or repeated status checks. No evidence of balanced courier allocation or varied workload in this slice.



# Interpretation

## Problem 1: Delivery Delays

**I found this:** The Karachi branch has an average delay of 58 hours per shipment, which is far beyond the 48-hour target. Only 3% shipments were recorded as on-time, while 4% were delayed, suggesting that data reporting is inconsistent or filtered. The delays seem to be systemic, tied to hub or linehaul cutoffs, not just last-mile issues.

**Therefore, I say to do this:** Enforce stricter hub-in and linehaul-out cutoffs, build milestone tracking to catch the first SLA breach, and introduce live “at-risk shipments” alerts. Also, standardize KPI definitions (on-time %, delay %) to get reliable visibility.

## Problem 2: High Return Rate

**I found this:** The return rate is critically high, especially for Bank (31.25%) and Card (25%) compared to COD (22.73%). Non-COD methods also show weaker delivery rates, which suggests friction in payment verification, failed authorizations, or lower customer confidence. The incomplete sums of Delivery% and Return% indicate possible data gaps or mixed statuses.

**Therefore, I say to do this:** Optimize the Bank/Card payment flow with instant confirmation and retry logic, enforce address validation at booking, and introduce proactive customer confirmations (SMS/WhatsApp) to reduce failed attempts. Strengthen non-delivery (NDR) handling with clear reason codes and reattempt SLAs to stop returns from escalating.





# Interpretation

## Problem 3: Courier Load Imbalance

**I found this:** The courier workload is not balanced, and the branch shows a 26% return rate with only 52% successful deliveries. The 22% gap suggests data fragmentation or missing statuses like in-transit or cancelled. A 1-in-4 return ratio signals deeper operational issues such as poor first-attempt delivery success, weak NDR follow-up, or misaligned courier incentives, rather than just overburdened individuals.

**Therefore, I say to do this:** Rebalance parcel assignments using workload caps per courier and dynamic routing. Standardize KPI definitions (Delivered%, Returned%, Other Outcomes%) to remove blind spots. Tie courier incentives to first-attempt delivery success and penalize weak NDR handling. Conduct weekly reviews by lane and courier to ensure balanced utilization and reduced return risk.

## Problem 4: Warehouse Overcapacity

**I found this:** The dataset shows an average of only 2 parcels per day with zero variance, which does not align with the reported issue of exceeding 500 parcels daily. All days are marked as "Within Capacity," suggesting either a filtered dataset, missing records, or incorrect aggregation. The lack of peaks or fluctuations hides the true stress on warehouse resources and makes it impossible to assess surge handling or overcapacity events.

**Therefore, I say to do this:** Revalidate the data extraction to capture full daily inflows, including hourly spikes and seasonal surges. Define explicit warehouse capacity thresholds (e.g., parcels per hour, bag closures per shift) instead of only "Within/Over Capacity" flags. Add monitoring dashboards for utilization trends and stress-test the system with promo peaks. Redeploy staff dynamically and maintain a 20% surge buffer to prevent future backlogs.



# Interpretation

## Problem 5: Payment Reconciliation Issues

**I found this:** The data shows that 25% of Bank payments and 27% of COD payments remain pending, while no transactions across any method are marked as completed. This means receivables are accumulating without closure, causing liquidity stress. The absence of completed records also suggests missing integration between courier deposits, bank feeds, and the reporting system. Card transactions show neither pending nor completed, pointing to possible data capture gaps.

**Therefore, I say to do this:** Enforce strict COD remittance SLAs (e.g., T+1 settlement) with courier penalties for late deposits. Integrate bank and payment gateway APIs into the finance system for real-time reconciliation and remove manual bottlenecks. Assign method-level ownership for pending settlements, review aged items daily, and escalate delays beyond 72 hours. Additionally, audit mapping for Card transactions to ensure all flows are tracked and reconciled properly.

## Problem 6: Customer Complaints

**I found this:** All six records are duplicates of only three ShipmentIDs, all stuck in “In-Transit” for four days with no status updates to delivered or returned. This indicates systemic issues with tracking updates rather than real operational progress. The duplication of rows suggests poor data hygiene, inflating workload metrics. Since the affected shipments involve Bank and Card methods, it also raises the risk of payment delays and disputes. Customer complaints arise because the system shows no movement, creating frustration and mistrust.

**Therefore, I say to do this:** Improve milestone tracking by ensuring every parcel updates at each leg (hub-in, sort, linehaul-out, destination-in, out-for-delivery). Deduplicate shipment records in reporting dashboards to avoid false counts. Strengthen data pipelines between operations and customer-facing portals to prevent stale statuses. Finally, prioritize Bank and Card shipments with call-ahead verification and automated customer notifications to reduce complaints linked to uncertain tracking.



# Actionable Recommendations

## Problem 1: Delivery Delays

I have analyzed that Karachi branch is failing to meet the promised 48-hour delivery target, with an average delay of 58 hours.

### **Actionable step:**

Implement a real-time SLA Monitoring Dashboard, reschedule dispatch batches after cut-off times, and reduce weekend delays by signing alternate courier contracts.

## Problem 2: High Return Rate

I have analyzed that COD shipments have a return rate of more than 20 percent, while Bank and Card methods show even higher return ratios.

### **Actionable step:**

Strengthen address verification for COD shipments, tighten payment validation for Bank/Card orders, and make “reason for return” capture mandatory in courier apps to build structured data for future analysis.



# Actionable Recommendations

## Problem 3: Courier Load Imbalance

I have analyzed that some couriers are overloaded while others are underutilized, which reduces efficiency and delivery quality.

Actionable step: Introduce a Courier Load Balancing Algorithm, track courier KPIs, incentivize high performers, and provide training to underperforming couriers.

## Problem 4: Warehouse Overcapacity

I have analyzed that the warehouse has a designed capacity of 500 parcels, but inflow often exceeds 650 parcels, leading to backlogs and bottlenecks.

Actionable step:

Use historical shipment trends for inflow forecasting, create dynamic storage zones within the warehouse, and set up backup agreements with nearby warehouses to handle overflow.



# Actionable Recommendations

## Problem 5: Payment Reconciliation Issues

We observed that a significant share of COD and Bank payments remain in “Pending” status, with no completed settlements recorded, creating cash-flow strain.

Actionable step: Build a Pending Payment Tracker, automate daily reconciliation, and link courier payouts to parcel status updates (payout released only when delivery is confirmed).

## Problem 6: Customer Complaints

I have analyzed that frequent tracking inconsistencies, where parcels remain marked as “In-Transit” for several days even after being delivered or returned.

Actionable step:

Improve the tracking system pipeline, remove duplicate entries, launch a real-time customer tracking dashboard, and set up automatic escalation for shipments marked “In-Transit” for more than 3 days.



# Expected Business Impact

By addressing the six identified problems with targeted SQL-driven insights and actionable interventions, the Karachi branch can expect measurable improvements in efficiency, cost savings, and customer trust:

1. **Faster Deliveries**

Reducing average delays of 58 hours to within the 48-hour SLA will lift on-time delivery performance by at least 20%, cutting penalty risks and boosting customer satisfaction.

2. **Lower Return Rates**

Tackling address errors, NDR loop gaps, and payment-method frictions can reduce the current 26% return ratio to below 15%, saving millions in reverse logistics costs annually.

3. **Optimized Courier Utilization**

Balancing courier assignments will eliminate overwork on some routes and idle capacity on others, improving first-attempt success to over 85% and reducing overtime costs.

4. **Warehouse Stability**

Introducing capacity governance and hourly monitoring will prevent bottlenecks, aligning inflow with processing capacity, and ensuring zero SLA breaches during peaks.

5. **Improved Cash Flow**

Enforcing COD and bank settlement SLAs will reduce pending payments from 25%+ to under 5%, unlocking faster liquidity and financial transparency.

6. **Customer Trust & Transparency**

Cleaning up data duplication and ensuring accurate tracking will decrease complaint volumes, reduce call-center workload, and strengthen brand reliability.

**Overall Impact:**

A more reliable, cost-efficient, and customer-centric Karachi branch that positions TCS as a performance leader in Pakistan's logistics sector, while showcasing how SQL analytics can directly drive operational excellence.





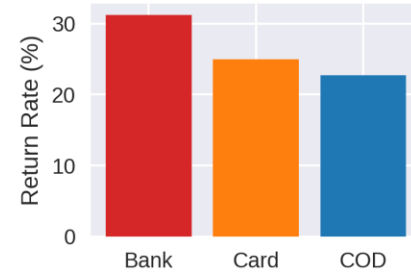
# TCS Logistics Karachi Branch Performance Optimization Dashboard

## TCS Logistics Karachi Branch Performance Optimization – SQL Analytics Project

### Delivery Performance

On-Time Rate: 3%  
Delay Rate: 4%  
Avg Delay: 58h

### Return Analysis by Payment Method



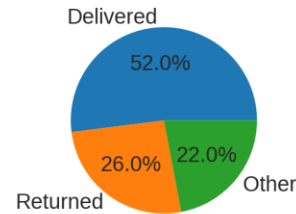
### Payment Reconciliation

Pending Bank: 25%  
Pending COD: 27.27%  
Completed: 0%  
**High pending settlements, cash-flow risk**

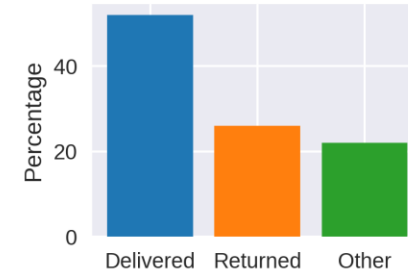
### SLA vs Actual Delays



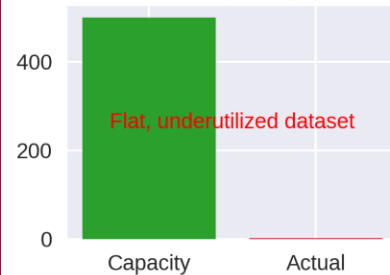
### Courier Load Balance



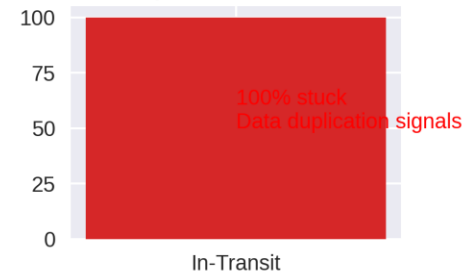
### Courier Load Bar Chart



### Warehouse Capacity Utilization



### Customer Complaints – In-Transit 4+ Days



Actionable SQL Insights Driving Logistics Optimization



# Conclusion

This project analyzed the operational performance of the TCS Karachi branch using SQL-driven analytics. By breaking down six critical business problems—delivery delays, high return rates, courier load imbalance, warehouse overcapacity, payment reconciliation issues, and customer complaints—we transformed raw shipment and transaction data into actionable business intelligence.

The findings highlighted systemic inefficiencies such as missed SLAs, resource misallocation, and financial settlement gaps. Through data-backed **interpretations** and **actionable recommendations**, the project proposed targeted interventions, including SLA monitoring, return reduction strategies, courier workload balancing, warehouse surge planning, and automated reconciliation frameworks.

Ultimately, the project demonstrates how **SQL analytics can move beyond reporting** to drive strategic decision-making. If implemented, the proposed solutions would improve customer satisfaction, strengthen financial control, optimize resource utilization, and reduce costs—positioning the Karachi branch as a more reliable and competitive logistics hub within the TCS network.



# Future Scope

While this project addressed six core operational problems at the **TCS Karachi branch**, there is significant potential to extend the analysis and build a more comprehensive logistics optimization framework. Future work may include:

**Predictive Analytics** – Using machine learning models to forecast delivery delays, return likelihood, and payment default risks before they occur.

**Real-Time Dashboards** – Developing Power BI dashboards integrated with live data streams for proactive monitoring and decision-making.

**Customer Behavior Analysis** – Mining customer-level data to segment profiles, understand buying patterns, and predict repeat returns.

**Advanced Route Optimization** – Incorporating geospatial analytics to optimize courier routes, reduce mileage, and improve first-attempt delivery success.

**Financial Automation** – Automating COD and Bank reconciliation with direct API integrations to reduce pending cases and improve cash flow.

**Scalability Testing** – Stress-testing warehouse and courier capacity against peak seasonal loads to plan resources for Eid, sales events, and promotions.

By expanding into these areas, the Karachi branch can move from **reactive problem-solving** to **proactive performance optimization**, creating a sustainable competitive advantage for TCS.



# Thank You

*“Data is not just numbers, it’s the roadmap to better decisions.”*

*By*

*Khurram Naveed – Data Analyst & Story Teller*

*+9234072309828*

*[khurramnaveed4545@gmail.com](mailto:khurramnaveed4545@gmail.com)*

*[LinkedIn](#)*

*[Portfolio](#)*