Swiggy Case Study — Refunds (Project)

Author: Tushar Kalra
Role: Business Analyst / Product Analyst (Portfolio Project)

---

1) Project Summary

This case study tackles the challenge of high refund rates on Swiggy. I analyzed potential root causes such as restaurant errors, delivery delays, and opportunistic customer fraud, and proposed a phased, proof-backed solution that balances user trust with operational accountability.

---

2) Deliverables

- `Swiggy_Case_Study_Refunds_Tushar_Kalra.pdf` — One-page case study + PRD snippet.
- `PRD.md` — Detailed Product Requirement Document.
- `SQL/diagnostics_queries.sql` — Diagnostic SQL queries.
- `Dashboards/` — Excel/PowerBI visuals (KPIs, trends, restaurant-level breakdowns).

---

3) README draft (for GitHub)

```
# Swiggy Refunds — Root-Cause & Proof-backed Refunds

**Author:** Tushar Kalra

## Overview
Refund requests were rising and hurting both customer satisfaction and unit
economics. This project investigates the root causes and proposes a phased
solution supported by data diagnostics, SQL queries, and PRD-driven
recommendations.

## Key Components
- Case study PDF (executive summary)
- Detailed PRD document
- SQL queries for diagnostic analysis
- Dashboard mockups (Excel/PowerBI)

## Key Metrics & Targets
- Refund rate drop >= 20% (pilot cities)
- Avg resolution time reduced by 50%
- False refund rate decreased significantly

## Contact
📧 tusharkalra94613@gmail.com
```

4) PRD.md Suggested Sections

- Title
- Author & Date
- Objective
- Problem statement
- Scope
- User stories
- Acceptance criteria
- Metrics & dashboards
- Phased implementation (diagnostics → quick wins → proof/verification → enforcement)
- Risks & trade-offs

---

5) Diagnostic SQL Examples

```sql
-- Refund rate by restaurant
SELECT r.restaurant_id, r.name,
       COUNT(*) FILTER (WHERE o.refund_requested = 1) AS refunds,
       COUNT(*) AS total_orders,
       ROUND(100.0 * COUNT(*) FILTER (WHERE o.refund_requested = 1) /
COUNT(*), 2) AS refund_pct
FROM orders o
JOIN restaurants r ON o.restaurant_id = r.restaurant_id
GROUP BY r.restaurant_id, r.name
ORDER BY refund_pct DESC;

-- Refunds by order value band
SELECT CASE
         WHEN total_amount <= 300 THEN '<=300'
         WHEN total_amount <= 700 THEN '301-700'
         ELSE '>700' END AS band,
       COUNT(*) AS refunds
FROM orders
WHERE refund_requested = 1
GROUP BY band;

-- Repeat refund customers
SELECT customer_id, COUNT(*) AS refund_count
FROM orders
WHERE refund_requested = 1
GROUP BY customer_id
HAVING COUNT(*) > 1
ORDER BY refund_count DESC;
```

---

6) Dashboard Outline

- KPIs: Refund rate per 1,000 orders, Avg resolution time, % auto-approved refunds, Repeat refund customers %, Restaurants flagged.
- Charts: Refund % by restaurant (bar), refund band distribution (pie), refund trend over time (line).
- Filters: city, order_value_band, date range.

---

*Prepared and authored by Tushar Kalra***