Ride Operations and Revenue Optimization Analysis (Using SQL Server)

ANALYSIS OF UBER'S OPERATIONAL DATA USING SQL

1. PROJECT OVERVIEW

This project focuses on analyzing Uber's operational datasets (rides, drivers, payments, and cities) using SQL to extract meaningful business insights. The primary goal is to identify key operational challenges such as high cancellation rates, revenue leakage, and driver performance issues to support data-driven decision-making.

2. PROBLEM STATEMENT

Uber operates in multiple cities and relies heavily on data to optimize its services. However, challenges such as ride cancellations, payment discrepancies, and inconsistent driver performance impact operational efficiency. This SQL capstone project aims to analyze Uber's operational data to uncover patterns and provide strategic recommendations for improving performance and revenue.

3. OBJECTIVES

- Analyze city-wise ride performance

- Detect revenue leakage in completed rides

- Study cancellation patterns across cities and time

- Evaluate driver performance metrics

- Optimize database performance using indexes, views, and triggers

- Ensure data integrity through cleaning and relational design

4. DATASET DESCRIPTION

The project uses four relational datasets:

- Rides Dataset: Ride details including fare, ride status, timestamps, and cities

- Driver Dataset: Driver ratings, demographics, and experience

- Payment Dataset: Transaction status, payment method, and fare details

- City Dataset: Population and city-level attributes

Each dataset initially contained 500 rows and required preprocessing before analysis.

5. DATA CLEANING PROCESS

A comprehensive SQL-based data cleaning process was implemented:

- Removed NULL values in critical columns such as fare, ride_id, and population

- Deleted duplicate records using CTE

- Standardized data types and constraints

- Created Primary Keys for all tables

- Established Foreign Key relationships

6. DATABASE DESIGN & RELATIONSHIPS

Primary Keys:

- ride_id (rides table)

- driver_id (driver table)

- payment_id (payment table)

- city_id (city table)

Foreign Key:

- rides.driver_id references driver.driver_id

7. SQL FEATURES IMPLEMENTED

- Data Cleaning using SQL (DELETE, NULL handling)

- CTE for duplicate removal

- Views for analytics (avg_fare_by_city, driver_performance_metrics)

- Non-clustered indexes for performance optimization

- Trigger for ride status change logging

- Aggregation and analytical queries

8. BUSINESS ANALYSIS & INSIGHTS

8.1 City-Level Performance Analysis

Top cities were analyzed based on total rides, average driver ratings, and cancellation rates to support driver recruitment decisions.

8.2 Revenue Leakage Analysis

Detected rides marked as “Completed” but with pending payments, indicating financial leakage.

8.3 Cancellation Pattern Analysis

High cancellation patterns were observed across specific cities and peak time slots, impacting revenue and customer experience.

8.4 Time-Based Cancellation Trends

Evening and late-night hours showed the highest cancellation rates, suggesting driver shortages during peak demand.

8.5 Seasonal Fare Variation

Monthly fare analysis indicated fluctuations due to dynamic pricing and seasonal demand changes.

8.6 Driver Performance Evaluation

Driver efficiency was measured using:

- Total rides

- Average ratings

- Total earnings

- Average ride duration

9. QUERY PERFORMANCE OPTIMIZATION

- Non-clustered index created on ride_date

- Index on payment_method for faster queries

- Views implemented for reusable analytics

- Structured schema to enhance performance

10. KEY FINDINGS

- High cancellation rates significantly impact revenue

- Revenue leakage due to pending payments on completed rides

- Peak-hour demand leads to increased cancellations

- Driver performance varies across cities

- Fare trends indicate dynamic pricing effects

11. RECOMMENDATIONS

- Increase driver recruitment in high-demand cities

- Implement payment reconciliation checks

- Introduce peak-hour driver incentives

- Improve ride allocation algorithms

- Use performance-based driver reward systems

12. CONCLUSION

This SQL-based Uber Operational Data Analysis project demonstrates strong skills in data cleaning, database design, indexing, triggers, and business analytics. The insights derived can help improve operational efficiency, reduce revenue leakage, and enhance customer satisfaction through data-driven decision-making.