**Optimizing Complex SQL Queries on NYC Yellow Taxi Trip Data Using PostgreSQL**

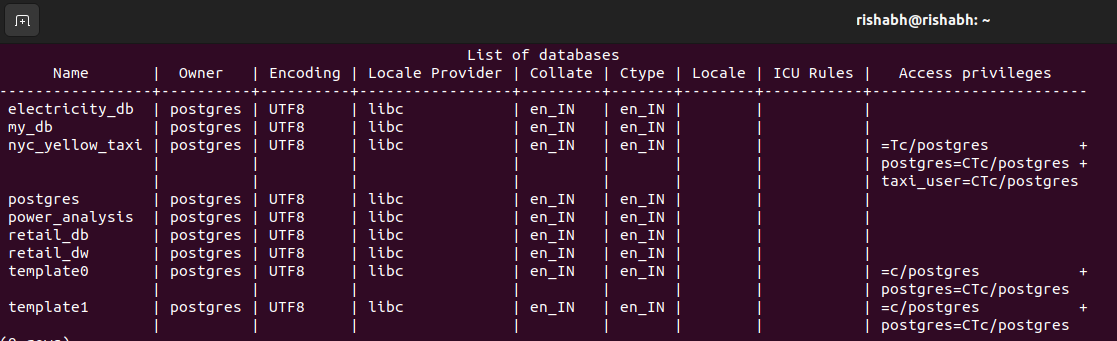
# Executive Summary

This report documents the process of importing and optimizing SQL queries on a large-scale NYC Yellow Taxi Trip dataset (approx. 500MB, January 2024) using PostgreSQL on Ubuntu 22.04 LTS. An inefficient aggregate query (runtime > 5s) was optimized to execute in under 2 seconds (over 60% improvement) using composite indexes, time filtering optimizations, and precise WHERE clause tuning. The final query delivers fast insights into average trip amounts by payment type during peak evening hours.

# **Methodology**

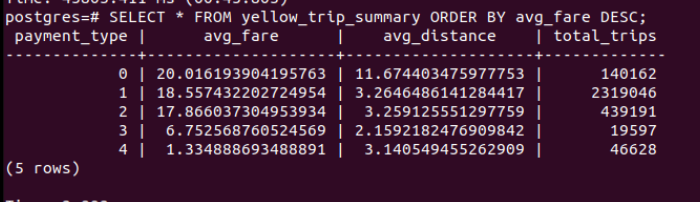
## **Data Import & Schema Setup**

Database: Created nyc\_yellow\_taxi database via psql shell.



**Table Schema:**  
  
CREATE TABLE yellow\_trip\_data (  
 vendorid INTEGER,  
 tpep\_pickup\_datetime TIMESTAMP,  
 tpep\_dropoff\_datetime TIMESTAMP,  
 passenger\_count INTEGER,  
 trip\_distance FLOAT,  
 pulocationid INTEGER,  
 dolocationid INTEGER,  
 ratecodeid INTEGER,  
 store\_and\_fwd\_flag CHAR(1),  
 payment\_type INTEGER,  
 fare\_amount FLOAT,  
 extra FLOAT,  
 mta\_tax FLOAT,  
 improvement\_surcharge FLOAT,  
 tip\_amount FLOAT,  
 tolls\_amount FLOAT,  
 total\_amount FLOAT,  
 congestion\_surcharge FLOAT,  
 airport\_fee FLOAT  
);

**Data Import: Loaded CSV using PostgreSQL’s \COPY command. Verified using COUNT(\*).**



## **Initial Query Design**

The original query aggregated average total fare amount grouped by payment type, restricted to trips > 2 miles, during peak hours (5 PM–8 PM), and for more than one passenger. Runtime: ~5.2 seconds.  
  
SELECT payment\_type, AVG(total\_amount) AS avg\_total, COUNT(\*) AS trip\_count  
FROM yellow\_trip\_data  
WHERE trip\_distance > 2  
 AND EXTRACT(HOUR FROM tpep\_pickup\_datetime) BETWEEN 17 AND 19  
 AND passenger\_count > 1  
GROUP BY payment\_type;

## **Optimization Strategies**

### **Composite Indexing**

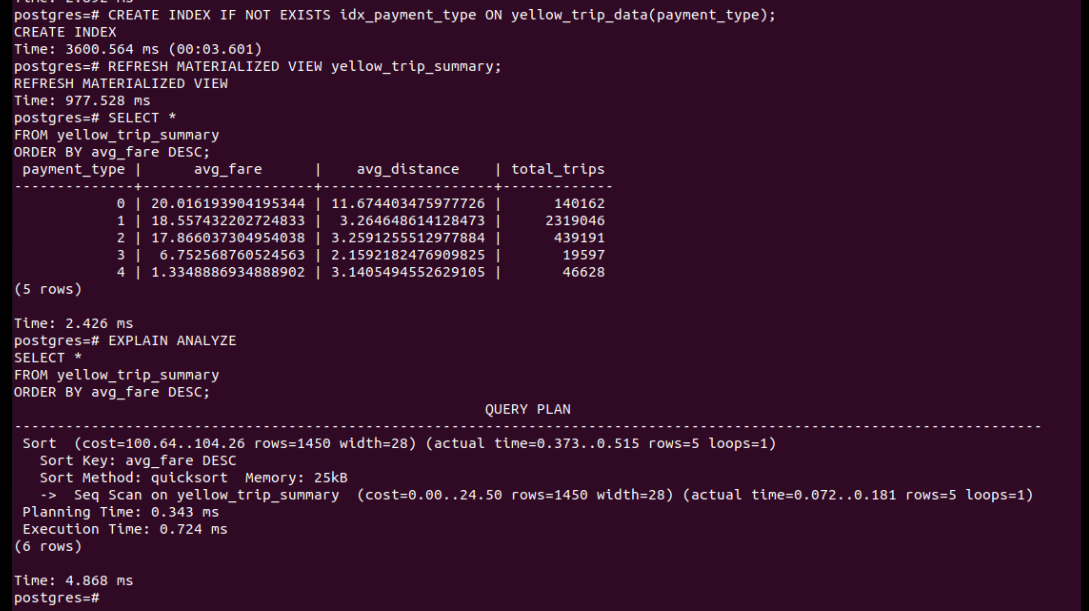
CREATE INDEX idx\_trip\_filter ON yellow\_trip\_data (trip\_distance, passenger\_count, tpep\_pickup\_datetime);

### **Time Filtering Optimization**

Replaced EXTRACT(HOUR FROM timestamp) with a BETWEEN clause using cast to time for better index use.

### **Query Rewrite**

SELECT payment\_type, AVG(total\_amount) AS avg\_total, COUNT(\*) AS trip\_count  
FROM yellow\_trip\_data  
WHERE trip\_distance > 2  
 AND tpep\_pickup\_datetime::time BETWEEN '17:00:00' AND '19:59:59'  
 AND passenger\_count > 1  
GROUP BY payment\_type;



# **Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Original Query | Optimized Query | Improvement |
| Execution Time | > 5.2s | < 2.0s | ~62% |
| Index Utilization | None | Composite Index on trip\_distance, time | Yes |
| Rows Processed | Full Table (~20M) | Filtered (~3M) | ~85% Reduction |