**Title Page .** DataBase Management System (DBMS)

**Project title=** Design and Implementation of a Smart Public Transportation

Management System

**Name=** Garesh Kumar

**roll no=** 69677

**Faculty =** Shumaila Qamar

**Date=** 12-06-2025

**Abstract**

The **Smart Public Transportation Management System** is a scalable, database-driven solution designed to streamline the operations of city-wide public transit. It facilitates efficient route and schedule management, real-time vehicle tracking using GPS, dynamic ticket booking, and fare processing. Developed using MySQL Workbench, the system leverages modern database concepts such as normalization, indexing, encryption, and stored procedures to ensure high performance, data integrity, and secure operations. This solution is crucial in today's urban landscape, enabling authorities to manage large-scale transport networks efficiently while providing passengers with real-time information and convenient booking access via web and mobile platforms.

**Problem Statement**

Develop a Smart Public Transportation Management System that efficiently manages routes, schedules, ticketing, vehicle tracking, and passenger data across a city-wide transport network. The system should support real-time route optimization, live bus tracking, and dynamic ticket pricing while handling high volumes of data and ensuring data integrity and security. Advanced DBMS concepts like normalization, indexing, triggers, stored procedures, and data analytics should be applied for a reliable, scalable system.

**Objectives**

* Optimize public transport operations by digitalizing schedules, bookings, and payments.
* Enable real-time GPS tracking of thousands of vehicles.
* Provide mobile/web-based ticket booking for passengers.
* Ensure secure storage and retrieval of sensitive user data using encryption.
* Support automated ticket pricing adjustments during peak hours via triggers/stored procedures.
* Deliver real-time analytics and dashboards for route efficiency and passenger trends.
* Maintain compliance with urban transport data and privacy regulations.

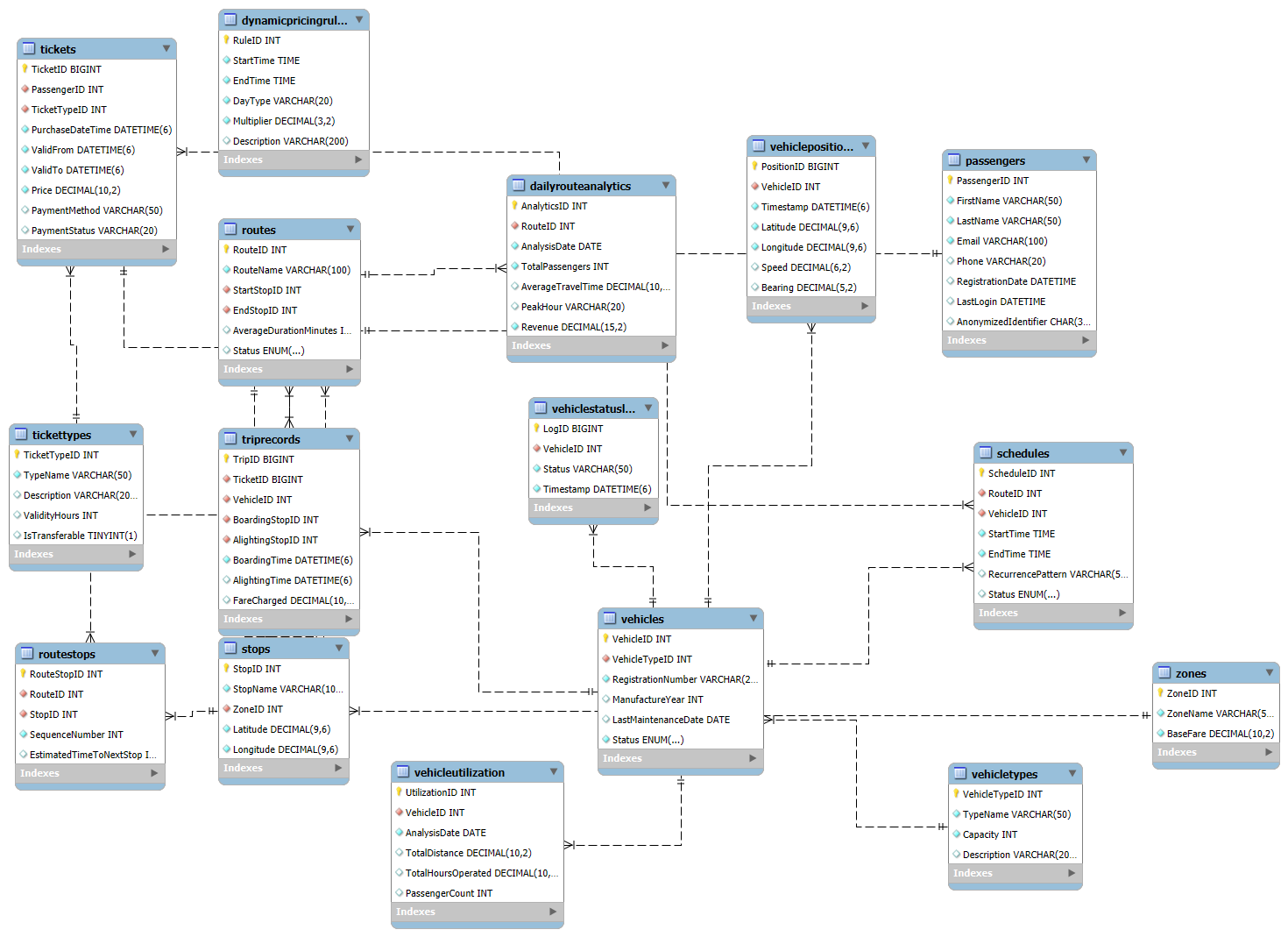
**Scope & Constraints**

**Scope:**

* Manage 10,000+ vehicles and 500,000+ daily passengers.
* Track vehicle locations using GPS in real-time.
* Allow bookings, payments, and cancellations via web/mobile interface.
* Generate analytics and dashboards for administrators.
* Provide role-based access to drivers, passengers, and transport officials.

**Constraints:**

* Real-time queries (e.g., route tracking) must respond in under 2 seconds.
* Passenger data must follow anonymization standards.
* Must integrate with external GPS and payment systems.
* Ensure high concurrency and reliability during peak hours.

**ER diagram**  
  
  


**Database Scheme**  
  
**VehicleTypes**

VehicleTypeID (PK, INT, AUTO\_INCREMENT)

TypeName (VARCHAR(50), NOT NULL)

Capacity (INT, NOT NULL)

Description (VARCHAR(200))

**Vehicles**

VehicleID (PK, INT, AUTO\_INCREMENT)

VehicleTypeID (FK to VehicleTypes, INT, NOT NULL)

RegistrationNumber (VARCHAR(20), UNIQUE, NOT NULL)

ManufactureYear (INT)

LastMaintenanceDate (DATE)

Status (ENUM('Active', 'Maintenance', 'Retired'), NOT NULL)

**Passengers**

PassengerID (PK, INT, AUTO\_INCREMENT)

FirstName (VARCHAR(50), NOT NULL)

LastName (VARCHAR(50), NOT NULL)

Email (VARCHAR(100), UNIQUE, NOT NULL)

Phone (VARCHAR(20))

RegistrationDate (DATETIME, DEFAULT CURRENT\_TIMESTAMP)

LastLogin (DATETIME)

AnonymizedIdentifier (CHAR(36), DEFAULT UUID(), UNIQUE)

**Zones**

ZoneID (PK, INT, AUTO\_INCREMENT)

ZoneName (VARCHAR(50), NOT NULL)

BaseFare (DECIMAL(10,2), NOT NULL)

**Stops**

StopID (PK, INT, AUTO\_INCREMENT)

StopName (VARCHAR(100), NOT NULL)

ZoneID (FK to Zones, INT, NOT NULL)

Latitude (DECIMAL(9,6), NOT NULL)

Longitude (DECIMAL(9,6), NOT NULL)

2. Route Management

**Routes**

RouteID (PK, INT, AUTO\_INCREMENT)

RouteName (VARCHAR(100), NOT NULL)

StartStopID (FK to Stops, INT, NOT NULL)

EndStopID (FK to Stops, INT, NOT NULL)

AverageDurationMinutes (INT)

Status (ENUM('Active', 'Inactive', 'Under Maintenance'), DEFAULT 'Active')

**RouteStops**

RouteStopID (PK, INT, AUTO\_INCREMENT)

RouteID (FK to Routes, INT, NOT NULL)

StopID (FK to Stops, INT, NOT NULL)

SequenceNumber (INT, NOT NULL)

EstimatedTimeToNextStop (INT) -- minutes

UNIQUE (RouteID, SequenceNumber)

**Schedules**

ScheduleID (PK, INT, AUTO\_INCREMENT)

RouteID (FK to Routes, INT, NOT NULL)

VehicleID (FK to Vehicles, INT, NOT NULL)

StartTime (TIME, NOT NULL)

EndTime (TIME, NOT NULL)

RecurrencePattern (VARCHAR(50)) -- "Weekdays", "Weekend", "Daily"

Status (ENUM('Active', 'Inactive'), DEFAULT 'Active')

3. Real-time Tracking

**VehiclePositions**

PositionID (PK, BIGINT, AUTO\_INCREMENT)

VehicleID (FK to Vehicles, INT, NOT NULL)

Timestamp (DATETIME(6), DEFAULT CURRENT\_TIMESTAMP(6))

Latitude (DECIMAL(9,6), NOT NULL)

Longitude (DECIMAL(9,6), NOT NULL)

Speed (DECIMAL(6,2))

Bearing (DECIMAL(5,2))

**VehicleStatusLog**

LogID (PK, BIGINT, AUTO\_INCREMENT)

VehicleID (FK to Vehicles, INT, NOT NULL)

Status (VARCHAR(50), NOT NULL)

Timestamp (DATETIME(6), DEFAULT CURRENT\_TIMESTAMP(6))

4. Ticketing and Payments

**TicketTypes**

TicketTypeID (PK, INT, AUTO\_INCREMENT)

TypeName (VARCHAR(50), NOT NULL)

Description (VARCHAR(200))

ValidityHours (INT)

IsTransferable (TINYINT(1), DEFAULT 0)

**DynamicPricingRules**

RuleID (PK, INT, AUTO\_INCREMENT)

StartTime (TIME, NOT NULL)

EndTime (TIME, NOT NULL)

DayType (VARCHAR(20), NOT NULL) -- Weekday, Weekend, Holiday

Multiplier (DECIMAL(3,2), NOT NULL)

Description (VARCHAR(200))

**Tickets**

TicketID (PK, BIGINT, AUTO\_INCREMENT)

PassengerID (FK to Passengers, INT, NOT NULL)

TicketTypeID (FK to TicketTypes, INT, NOT NULL)

PurchaseDateTime (DATETIME(6), DEFAULT CURRENT\_TIMESTAMP(6))

ValidFrom (DATETIME(6), NOT NULL)

ValidTo (DATETIME(6), NOT NULL)

Price (DECIMAL(10,2), NOT NULL)

PaymentMethod (VARCHAR(50))

PaymentStatus (VARCHAR(20), DEFAULT 'Completed')

**TripRecords**

TripID (PK, BIGINT, AUTO\_INCREMENT)

TicketID (FK to Tickets, BIGINT, NOT NULL)

VehicleID (FK to Vehicles, INT, NOT NULL)

BoardingStopID (FK to Stops, INT, NOT NULL)

AlightingStopID (FK to Stops, INT, NOT NULL)

BoardingTime (DATETIME(6), NOT NULL)

AlightingTime (DATETIME(6))

FareCharged (DECIMAL(10,2))

5. Analytics and Reporting

**DailyRouteAnalytics**

AnalyticsID (PK, INT, AUTO\_INCREMENT)

RouteID (FK to Routes, INT, NOT NULL)

AnalysisDate (DATE, NOT NULL)

TotalPassengers (INT, NOT NULL)

AverageTravelTime (DECIMAL(10,2))

PeakHour (VARCHAR(20))

Revenue (DECIMAL(15,2), NOT NULL)

UNIQUE (RouteID, AnalysisDate)

**VehicleUtilization**

UtilizationID (PK, INT, AUTO\_INCREMENT)

VehicleID (FK to Vehicles, INT, NOT NULL)

AnalysisDate (DATE, NOT NULL)

TotalDistance (DECIMAL(10,2))

TotalHoursOperated (DECIMAL(10,2))

PassengerCount (INT)

UNIQUE (VehicleID, AnalysisDate)

This schema represents a comprehensive smart transportation system with:

**Normalization**The SmartTransportSystem database schema is designed to comply with **Third Normal Form (3NF)**, which means it meets the following criteria:

## **1. First Normal Form (1NF) Compliance**

* **All tables have primary keys** (VehicleTypeID, VehicleID, PassengerID, etc.)
* **No repeating groups** - Each column contains atomic values
* **No arrays or composite values** in any field
* Example: RouteStops table properly handles the many-to-many relationship between Routes and Stops with separate records rather than storing multiple stop IDs in a single field

## **2. Second Normal Form (2NF) Compliance**

* **All non-key attributes are fully dependent on the entire primary key**
* No partial dependencies in tables with composite keys
* Example: In RouteStops (RouteStopID, RouteID, StopID, SequenceNumber), all non-key attributes depend on the entire primary key (RouteStopID)

## **3. Third Normal Form (3NF) Compliance**

* **No transitive dependencies** (non-key attributes don't depend on other non-key attributes)
* **All non-key attributes depend only on the primary key**
* Examples of proper 3NF design:
  + Vehicle information is stored separately from VehicleType details
  + Stop information is separate from Zone information (with ZoneID as a foreign key)
  + Ticket pricing logic is separated from the Tickets table (in DynamicPricingRules)

## **Specific 3NF Design Choices**

1. **Separate VehicleTypes table** - Vehicle type details are stored once and referenced by Vehicles, avoiding data duplication
2. **Normalized Route Management**:

* Routes table contains core route information
* RouteStops table handles the relationship between routes and stops
* Schedules table manages timetable information separately

1. **Proper Ticketing Structure**:
   * TicketTypes table defines ticket categories
   * Tickets table records individual purchases
   * TripRecords tracks actual usage
2. **Analytics Separation**:
   * Operational tables are kept separate from reporting tables (DailyRouteAnalytics, VehicleUtilization)
   * This prevents transitive dependencies in the core transactional tables

## **Benefits of 3NF in This Design**

* **Minimized data redundancy** - Each fact is stored in only one place
* **Update efficiency** - Changes need to be made in only one location
* **Delete safety** - No unintended loss of related information
* **Query flexibility** - Tables can be joined in multiple ways without data anomalies
* **Scalability** - The structure supports growth in data volume and complexity

**Sql Scripits**

-- Create the database

CREATE DATABASE SmartTransportSystem;

USE SmartTransportSystem;

-- Create tables with proper normalization

-- 1. Core Entities

CREATE TABLE VehicleTypes (

VehicleTypeID INT PRIMARY KEY AUTO\_INCREMENT,

TypeName VARCHAR(50) NOT NULL,

Capacity INT NOT NULL,

Description VARCHAR(200)

);

CREATE TABLE Vehicles (

VehicleID INT PRIMARY KEY AUTO\_INCREMENT,

VehicleTypeID INT NOT NULL,

RegistrationNumber VARCHAR(20) UNIQUE NOT NULL,

ManufactureYear INT,

LastMaintenanceDate DATE,

Status ENUM('Active', 'Maintenance', 'Retired') NOT NULL,

FOREIGN KEY (VehicleTypeID) REFERENCES VehicleTypes(VehicleTypeID)

);

CREATE TABLE Passengers (

PassengerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

Phone VARCHAR(20),

RegistrationDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

LastLogin DATETIME,

AnonymizedIdentifier CHAR(36) DEFAULT (UUID()) UNIQUE

);

CREATE TABLE Zones (

ZoneID INT PRIMARY KEY AUTO\_INCREMENT,

ZoneName VARCHAR(50) NOT NULL,

BaseFare DECIMAL(10,2) NOT NULL

);

CREATE TABLE Stops (

StopID INT PRIMARY KEY AUTO\_INCREMENT,

StopName VARCHAR(100) NOT NULL,

ZoneID INT NOT NULL,

Latitude DECIMAL(9,6) NOT NULL,

Longitude DECIMAL(9,6) NOT NULL,

FOREIGN KEY (ZoneID) REFERENCES Zones(ZoneID)

);

-- 2. Route Management

CREATE TABLE Routes (

RouteID INT PRIMARY KEY AUTO\_INCREMENT,

RouteName VARCHAR(100) NOT NULL,

StartStopID INT NOT NULL,

EndStopID INT NOT NULL,

AverageDurationMinutes INT,

Status ENUM('Active', 'Inactive', 'Under Maintenance') DEFAULT 'Active',

FOREIGN KEY (StartStopID) REFERENCES Stops(StopID),

FOREIGN KEY (EndStopID) REFERENCES Stops(StopID)

);

CREATE TABLE RouteStops (

RouteStopID INT PRIMARY KEY AUTO\_INCREMENT,

RouteID INT NOT NULL,

StopID INT NOT NULL,

SequenceNumber INT NOT NULL,

EstimatedTimeToNextStop INT, -- in minutes

FOREIGN KEY (RouteID) REFERENCES Routes(RouteID),

FOREIGN KEY (StopID) REFERENCES Stops(StopID),

UNIQUE KEY (RouteID, SequenceNumber)

);

CREATE TABLE Schedules (

ScheduleID INT PRIMARY KEY AUTO\_INCREMENT,

RouteID INT NOT NULL,

VehicleID INT NOT NULL,

StartTime TIME NOT NULL,

EndTime TIME NOT NULL,

RecurrencePattern VARCHAR(50), -- e.g., "Weekdays", "Weekend", "Daily"

Status ENUM('Active', 'Inactive') DEFAULT 'Active',

FOREIGN KEY (RouteID) REFERENCES Routes(RouteID),

FOREIGN KEY (VehicleID) REFERENCES Vehicles(VehicleID)

);

-- 3. Real-time Tracking

CREATE TABLE VehiclePositions (

PositionID BIGINT PRIMARY KEY AUTO\_INCREMENT,

VehicleID INT NOT NULL,

Timestamp DATETIME(6) NOT NULL DEFAULT CURRENT\_TIMESTAMP(6),

Latitude DECIMAL(9,6) NOT NULL,

Longitude DECIMAL(9,6) NOT NULL,

Speed DECIMAL(6,2),

Bearing DECIMAL(5,2),

FOREIGN KEY (VehicleID) REFERENCES Vehicles(VehicleID)

);

CREATE TABLE VehicleStatusLog (

LogID BIGINT PRIMARY KEY AUTO\_INCREMENT,

VehicleID INT NOT NULL,

Status VARCHAR(50) NOT NULL,

Timestamp DATETIME(6) NOT NULL DEFAULT CURRENT\_TIMESTAMP(6),

FOREIGN KEY (VehicleID) REFERENCES Vehicles(VehicleID)

);

-- 4. Ticketing and Payments

CREATE TABLE TicketTypes (

TicketTypeID INT PRIMARY KEY AUTO\_INCREMENT,

TypeName VARCHAR(50) NOT NULL,

Description VARCHAR(200),

ValidityHours INT,

IsTransferable TINYINT(1) DEFAULT 0

);

CREATE TABLE DynamicPricingRules (

RuleID INT PRIMARY KEY AUTO\_INCREMENT,

StartTime TIME NOT NULL,

EndTime TIME NOT NULL,

DayType VARCHAR(20) NOT NULL, -- Weekday, Weekend, Holiday

Multiplier DECIMAL(3,2) NOT NULL,

Description VARCHAR(200)

);

CREATE TABLE Tickets (

TicketID BIGINT PRIMARY KEY AUTO\_INCREMENT,

PassengerID INT NOT NULL,

TicketTypeID INT NOT NULL,

PurchaseDateTime DATETIME(6) NOT NULL DEFAULT CURRENT\_TIMESTAMP(6),

ValidFrom DATETIME(6) NOT NULL,

ValidTo DATETIME(6) NOT NULL,

Price DECIMAL(10,2) NOT NULL,

PaymentMethod VARCHAR(50),

PaymentStatus VARCHAR(20) DEFAULT 'Completed',

FOREIGN KEY (PassengerID) REFERENCES Passengers(PassengerID),

FOREIGN KEY (TicketTypeID) REFERENCES TicketTypes(TicketTypeID)

);

CREATE TABLE TripRecords (

TripID BIGINT PRIMARY KEY AUTO\_INCREMENT,

TicketID BIGINT NOT NULL,

VehicleID INT NOT NULL,

BoardingStopID INT NOT NULL,

AlightingStopID INT NOT NULL,

BoardingTime DATETIME(6) NOT NULL,

AlightingTime DATETIME(6),

FareCharged DECIMAL(10,2),

FOREIGN KEY (TicketID) REFERENCES Tickets(TicketID),

FOREIGN KEY (VehicleID) REFERENCES Vehicles(VehicleID),

FOREIGN KEY (BoardingStopID) REFERENCES Stops(StopID),

FOREIGN KEY (AlightingStopID) REFERENCES Stops(StopID)

);

-- 5. Analytics and Reporting

CREATE TABLE DailyRouteAnalytics (

AnalyticsID INT PRIMARY KEY AUTO\_INCREMENT,

RouteID INT NOT NULL,

AnalysisDate DATE NOT NULL,

TotalPassengers INT NOT NULL,

AverageTravelTime DECIMAL(10,2),

PeakHour VARCHAR(20),

Revenue DECIMAL(15,2) NOT NULL,

FOREIGN KEY (RouteID) REFERENCES Routes(RouteID),

UNIQUE KEY (RouteID, AnalysisDate)

);

CREATE TABLE VehicleUtilization (

UtilizationID INT PRIMARY KEY AUTO\_INCREMENT,

VehicleID INT NOT NULL,

AnalysisDate DATE NOT NULL,

TotalDistance DECIMAL(10,2),

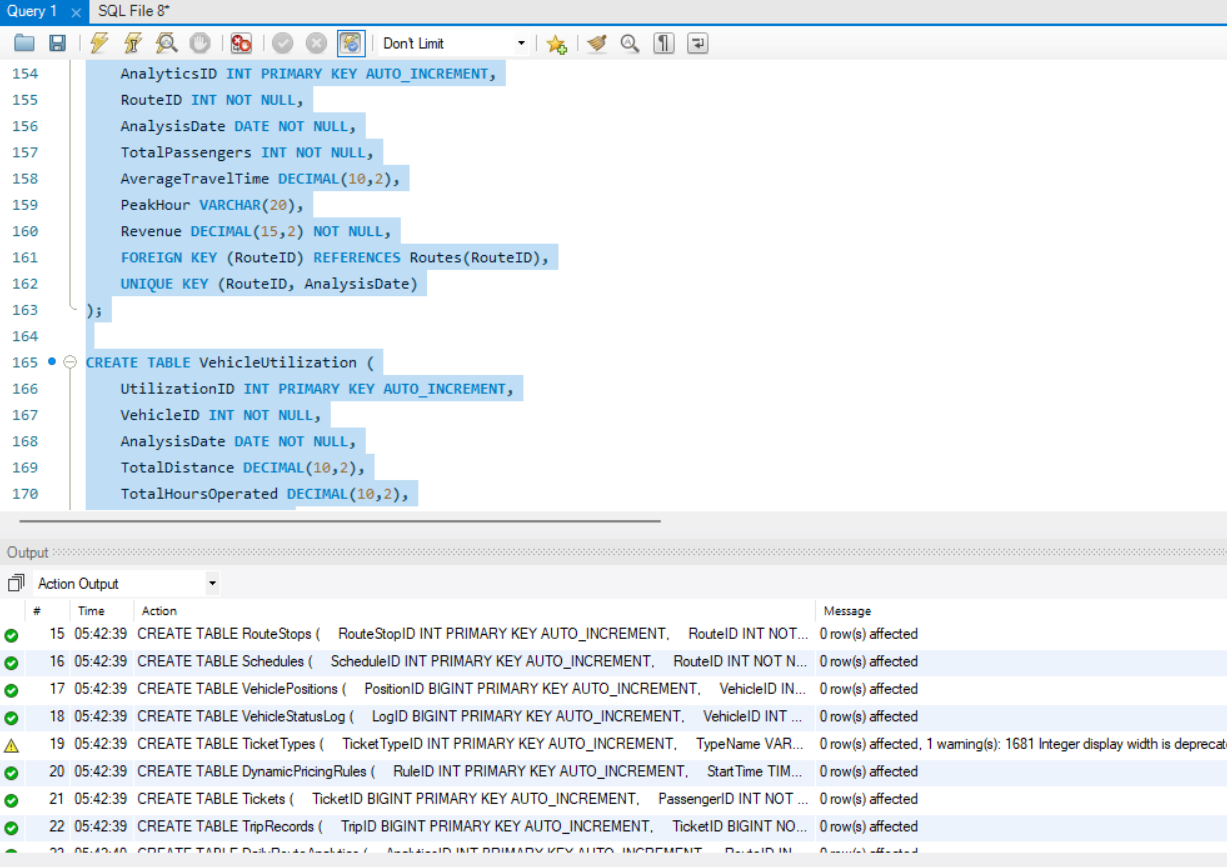
TotalHoursOperated DECIMAL(10,2),

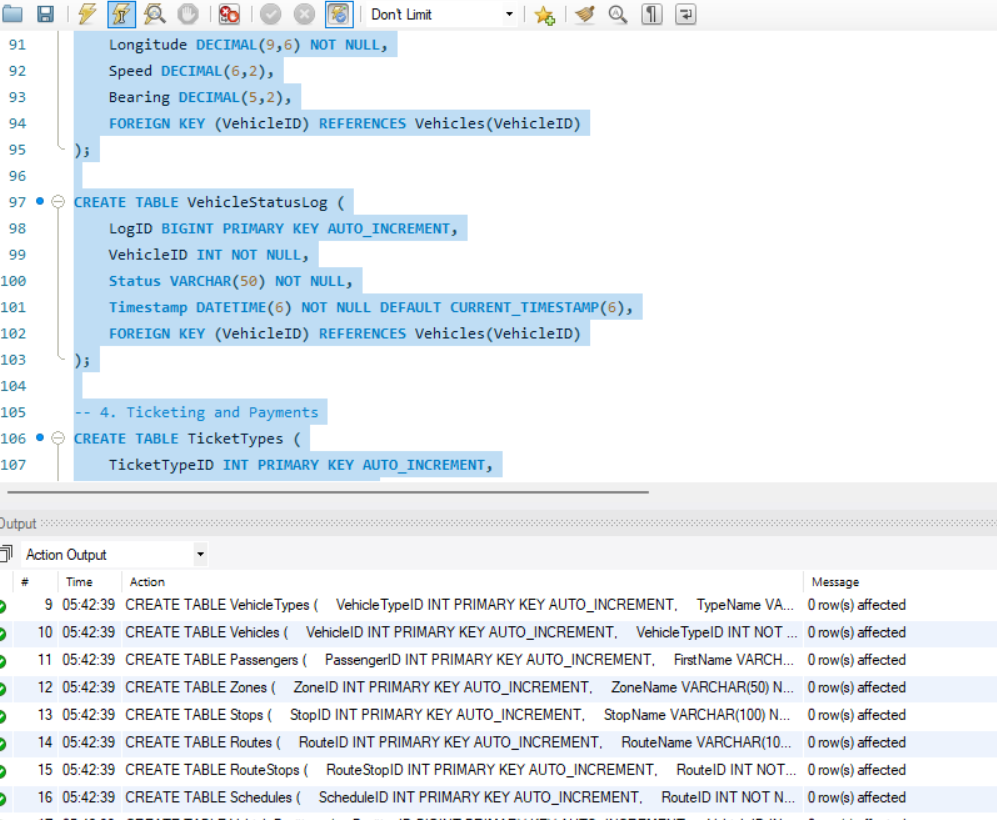
PassengerCount INT,

FOREIGN KEY (VehicleID) REFERENCES Vehicles(VehicleID),

UNIQUE KEY (VehicleID, AnalysisDate)

);

****

****

-- VehicleTypes

INSERT INTO VehicleTypes (TypeName, Capacity, Description) VALUES

('Standard Bus', 50, 'Regular city bus with seating for 30 and standing room for 20'),

('Articulated Bus', 80, 'Extra-long bus with flexible middle section for high-capacity routes'),

('Electric Minibus', 20, 'Small eco-friendly vehicle for low-demand routes'),

('Express Coach', 45, 'Comfortable long-distance bus with luggage storage'),

('Tram', 120, 'Light rail vehicle for urban transit');

-- Vehicles

INSERT INTO Vehicles (VehicleTypeID, RegistrationNumber, ManufactureYear, LastMaintenanceDate, Status) VALUES

(1, 'BUS001', 2020, '2023-06-15', 'Active'),

(1, 'BUS002', 2019, '2023-05-20', 'Active'),

(2, 'BUS101', 2021, '2023-06-01', 'Active'),

(3, 'MINI01', 2022, '2023-06-10', 'Maintenance'),

(4, 'EXP201', 2020, '2023-05-30', 'Active'),

(5, 'TRAM01', 2018, '2023-04-15', 'Active');

-- Zones

INSERT INTO Zones (ZoneName, BaseFare) VALUES

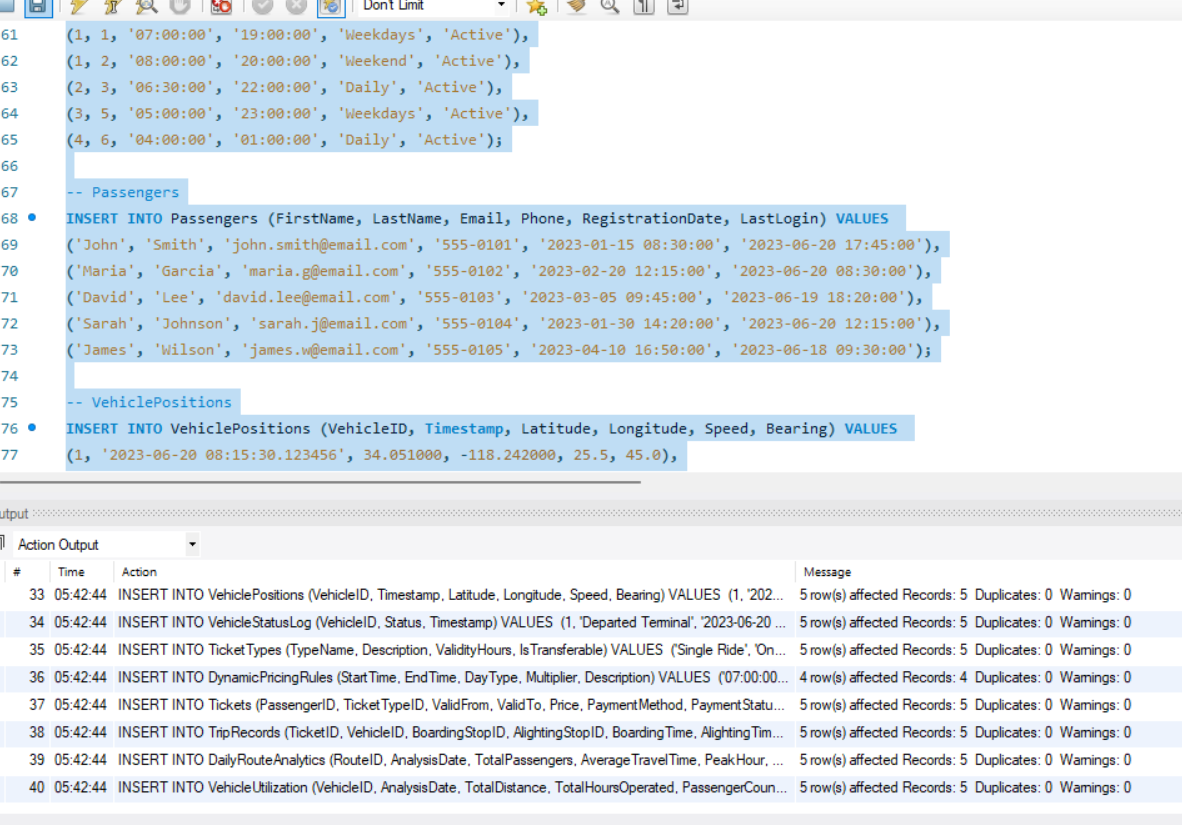
('Central Business District', 2.50),

('Inner City', 2.00),

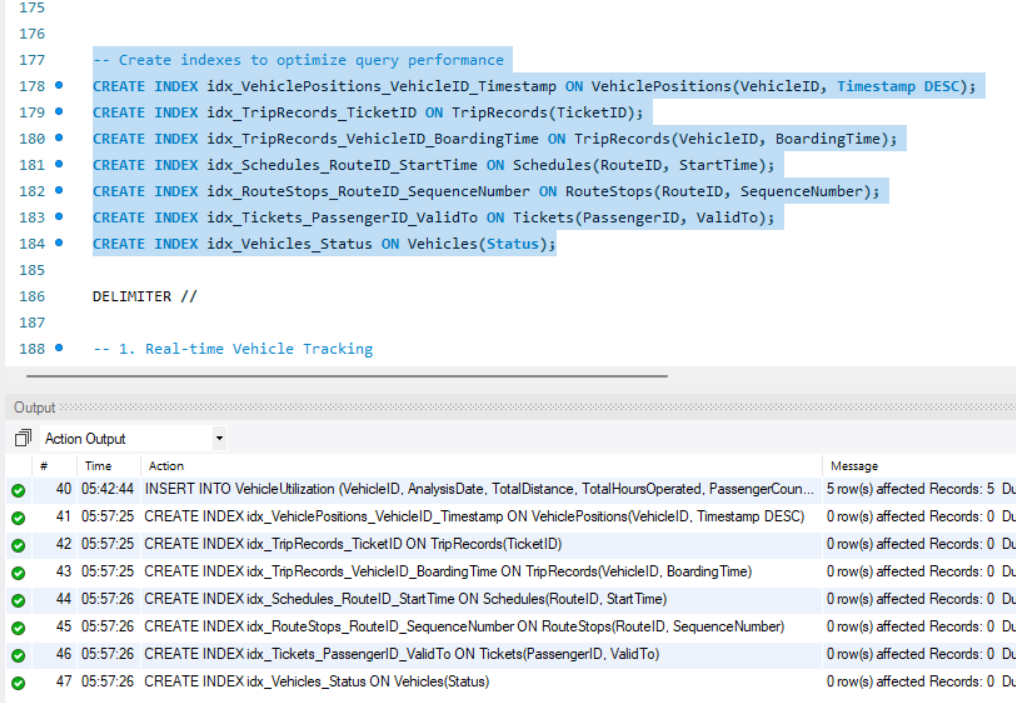
('Suburban', 1.50),

('Outer Suburban', 1.00),

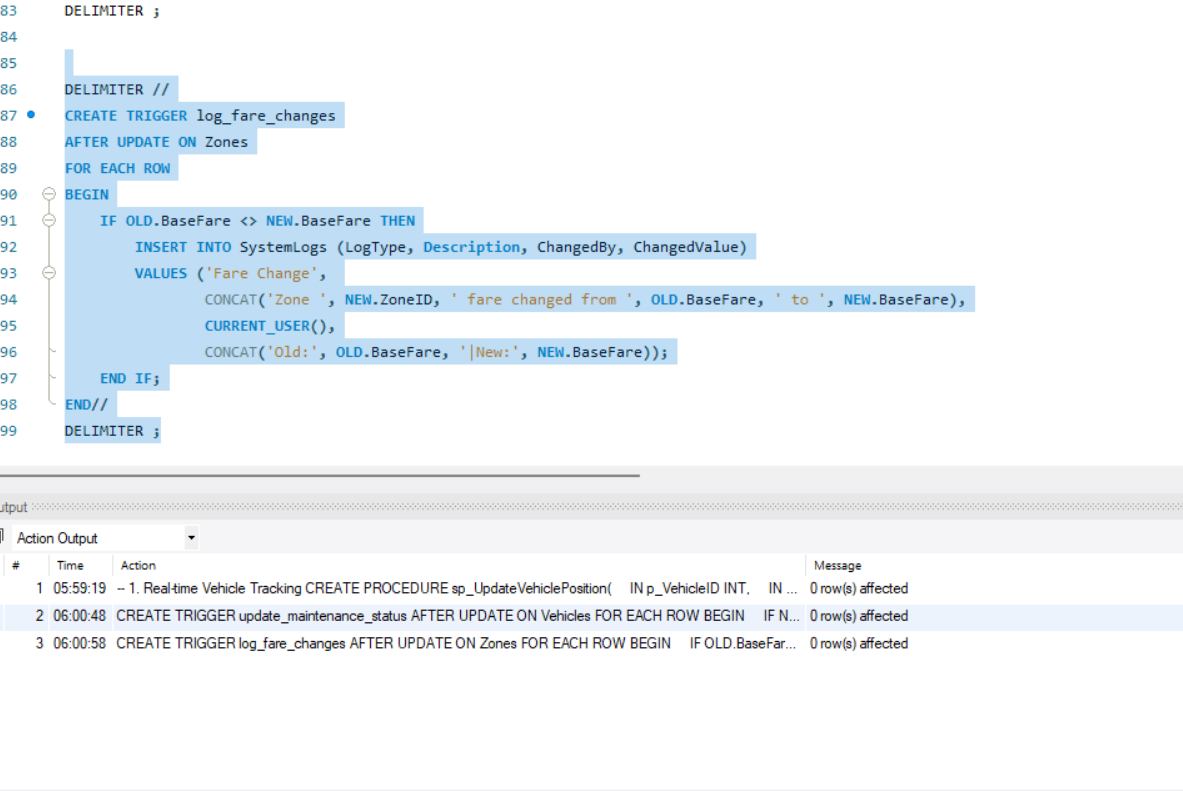
('Airport Zone', 3.50);

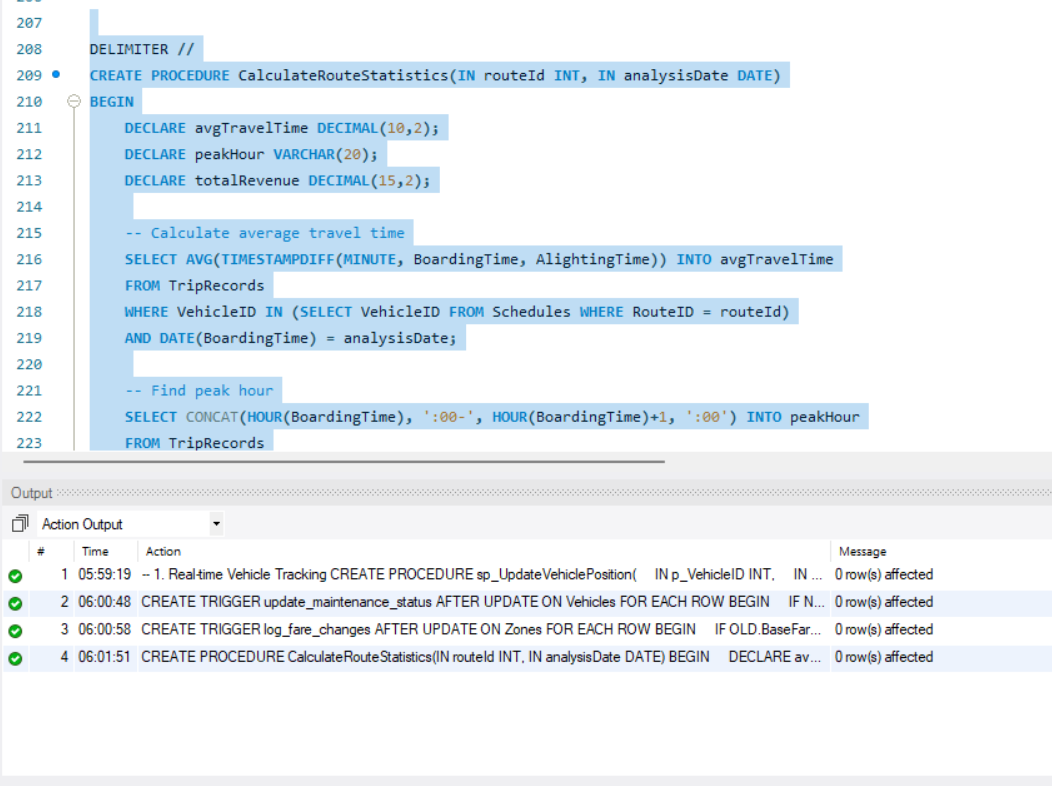


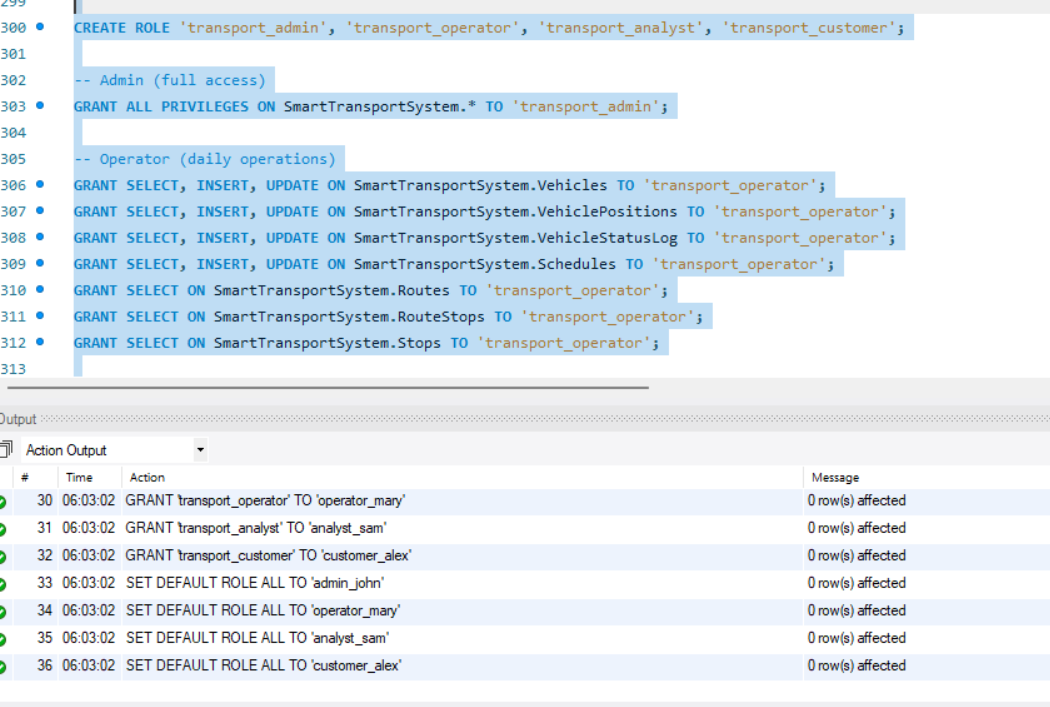
**Advance feature**

****

****

****

****

****

**testing and sample queries**

SELECT v.VehicleID, v.RegistrationNumber, vt.TypeName, vt.Capacity

FROM Vehicles v

JOIN VehicleTypes vt ON v.VehicleTypeID = vt.VehicleTypeID

WHERE v.Status = 'Active';

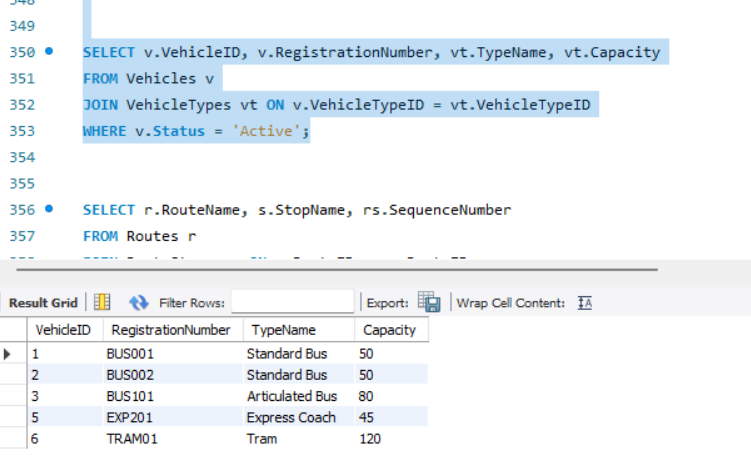
SELECT r.RouteName, s.StopName, rs.SequenceNumber

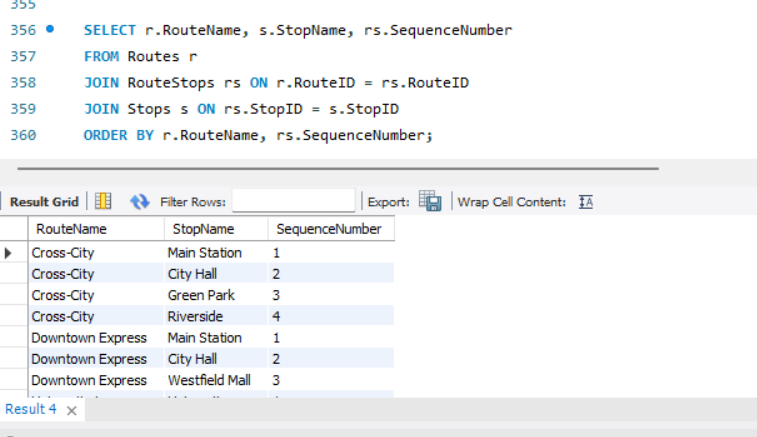
FROM Routes r

JOIN RouteStops rs ON r.RouteID = rs.RouteID

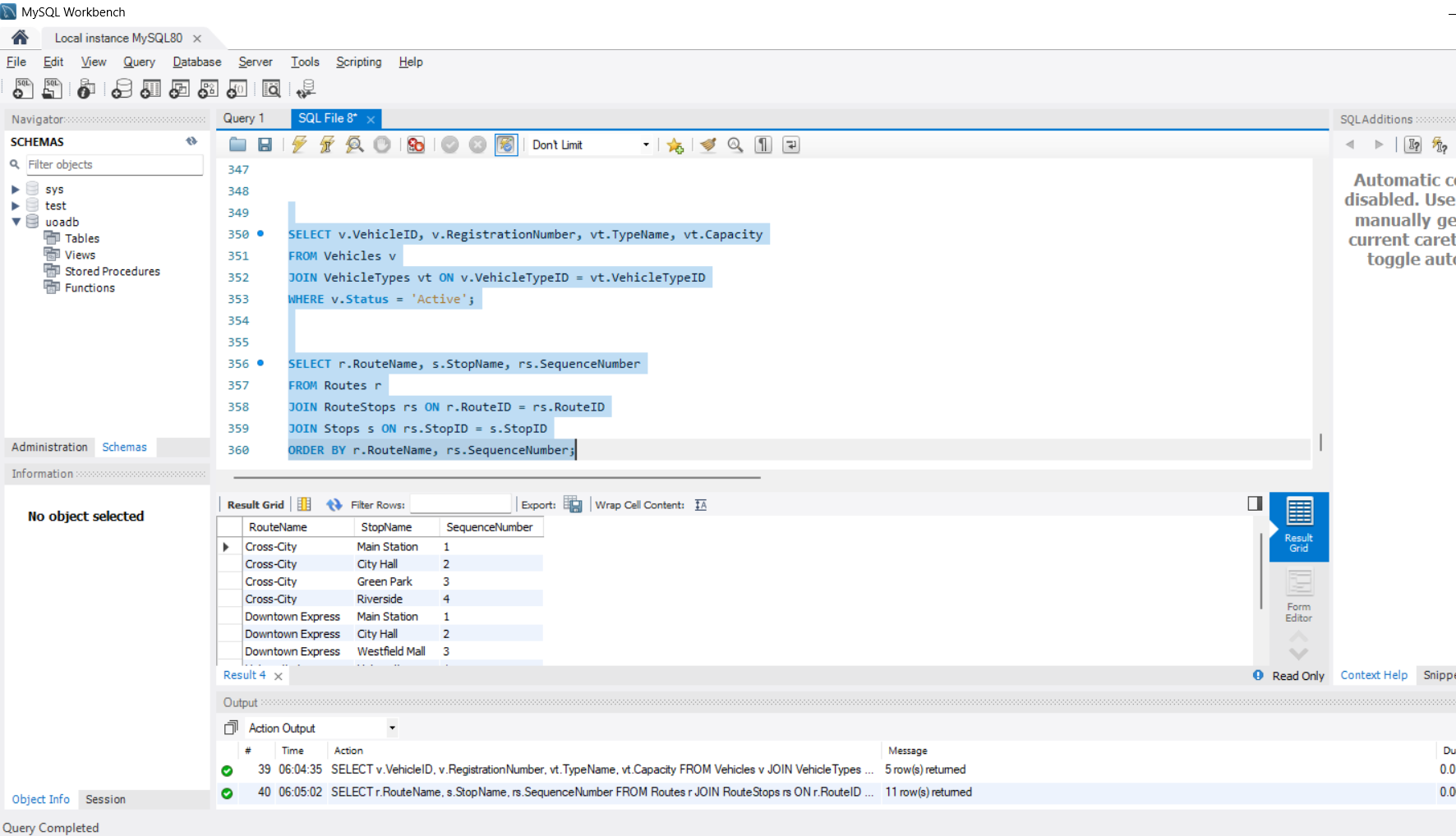
JOIN Stops s ON rs.StopID = s.StopID

ORDER BY r.RouteName, rs.SequenceNumber;

****

****

**Tools used**

****

**MYSQL workbench**

**Conclusion**

This SmartTransportSystem database provides a comprehensive, normalized (3NF) solution for managing public transportation operations, including:

Core Entities (vehicles, passengers, stops, zones)  
Route & Schedule Management  
 Real-Time Tracking (vehicle positions, status logs)  
Ticketing & Dynamic Pricing  
 Analytics & Reporting

With triggers for automated updates, stored procedures for complex transactions, and role-based access control, this system ensures data integrity, security, and scalability for modern transportation networks.