Karthik Garidepalli

Professor Huabo Lu

Introduction Database Systems

28 March 2025

Proposal for Car Rentals Management System

Introduction

In the rapidly evolving transportation industry, effective management of rental services is key to maintaining customer satisfaction and business efficiency. This proposal outlines the development of a Car Rentals Management System (CRMS) designed to simplify and optimize the operational needs of a fictional car rental organization. The system leverages MySQL as its database backend and Python with Tkinter for the front-end interface. The purpose of this database-driven application is to store, manage, and retrieve customer, vehicle, rental, and invoice information, enabling streamlined CRUD (Create, Read, Update, Delete) operations through a graphical user interface.

Business Story

s The fictional car rental company rents vehicles to customers on a daily basis. To maintain consistent service quality and financial oversight, the organization requires a reliable, normalized database system. The company needs to manage key information such as customer records, available cars, rental transactions, and invoice details. The current challenge lies in managing all of this data efficiently without redundancy, and ensuring accuracy when updating or deleting interrelated records. The CRMS is intended to automate and streamline these processes. It will enable users to monitor car availability, manage customer details, view rental

history, and generate invoices—all through a structured interface supported by a relational database

Database Software Choice: MySQL

For this system, MySQL Server has been chosen as the database software. MySQL is a widely-used, open-source relational database management system known for its stability, scalability, and support for ACID-compliant transactions. It integrates smoothly with Python applications via libraries such as mysql-connector-python, making it an ideal choice for academic and enterprise-grade solutions. In this system, MySQL facilitates the structured storage and retrieval of car rental data while ensuring data integrity via foreign key constraints and normalization.

Tentative Database Schema

The CRMS database schema includes four primary tables: Customers, Cars, Rentals, and Invoices.

- **Customers Table**
  - Attributes: customer_id (PK), first_name, last_name, email, phone
  - Functional Dependency: customer_id → all other attributes
  - Sample: (1, 'John', 'Doe', 'john.doe@example.com', '555-123-4567')
- **Cars Table**
  - Attributes: car_id (PK), car_type, car_color, car_price
  - Functional Dependency: car_id → all other attributes
  - Sample: (1, 'sports', 'Red', 25.00)

- **Rentals Table**
    - Attributes: rental_id (PK), customer_id (FK), car_id (FK), rental_start_date, rental_end_date
    - Constraints: ON DELETE CASCADE for both foreign keys
    - Functional Dependency: rental_id → all other attributes
    - Sample: (1, 1, 1, '2023-04-01', '2023-04-03')
- **Invoices Table**
    - Attributes: invoice_id (PK), rental_id (FK), invoice_amount
    - Constraints: ON DELETE CASCADE for rental_id
    - Functional Dependency: invoice_id → all other attributes
    - Sample: (1, 1, 50.00)

All tables follow the Third Normal Form (3NF), which ensures minimal redundancy and eliminates transitive dependencies. Foreign key constraints enforce data consistency and support referential integrity throughout the system.

Programming Tools Choice

The primary programming language used for the front-end application is Python. For GUI development, the Tkinter library is utilized, offering a simple yet effective way to build desktop applications. Python's versatility and readability make it a strong choice for database interaction and UI development. Additional libraries such as mysql-connector-python are used to establish a connection between the front end and the MySQL database. The choice of Python and Tkinter ensures that the system remains lightweight, easy to maintain, and accessible for future upgrades or enhancements.

Conclusion

The Car Rentals Management System provides a structured approach to managing car rental operations using Python and MySQL. By incorporating normalized tables, foreign key constraints, and a simple user interface, the application aims to provide an intuitive and robust platform for handling core business activities. This proposal outlines the foundation for building a practical solution that can be expanded in future iterations with additional features such as data analytics, authentication, and automated billing.