



Creating a Vehicle Management System through MySQL

Naveen R

PES University

ABSTRACT

This study was done to help build Vehicle Management System (VMS) software. Due to traditional manual vehicle management procedures, which created a number of complications in handling the cars sales and operation information, the establishment of this subscription management system was suggested. The main issues are that there is no database for vehicle and rental information, it takes a long time to handle reports, and incorrect vehicles are booked. Therefore, the main goals of this research initiative were to address the majority of the difficulties and complexity brought on by traditional manual vehicle management techniques. Overall, this development project implements an automated subscription management system that uses the concept of an interconnected (relational) database to store, manage, and organize data in a methodical manner. For this development project, MYSQL has been one of the primary development tools. In addition, the entire development project has been implemented using the Agile Unified Process methodology, which has been proven to be effective by numerous organizations. Since the user found it difficult to examine the data when the record was made manually and it was challenging to record quickly and efficiently, this solution will guarantee that using the application will be simple.

Introduction

The management techniques remain conservative and firmly rooted in tradition. Business transactions should be completed as quickly as possible, and manual management of the company leads to delayed transactions and a tendency for management to make simple mistakes while making reservations and even servicing the vehicle itself. Due to this, several consumers had a very negative experience with the booking procedure. When it comes to vehicle management systems, the entire process is carried out using a very conventional technique that requires a lot of manual labor. Due to these outdated recording techniques, management is having trouble overseeing their fleet and business operations.

Information about Vehicle Management System

A Vehicle The board Framework is a product application intended to help people or associations proficiently oversee different parts of their vehicles. This framework gives devices to follow, arrange, and keep up with data connected with vehicles, their proprietors, support, costs, from there, the sky is the limit. Here is an outline of the critical highlights and advantages of a vehicle the executives' framework:

Key Elements:

Vehicle Data: The framework permits clients to store insights concerning every vehicle, including make, model, year, variety, VIN (Vehicle Recognizable proof Number), enlistment number, and that's only the tip of the iceberg.

Customer Records: Clients can oversee data about vehicle proprietors, including their names, contact subtleties, and possession history.

Maintenance: The framework allows clients to record support exercises, for example, oil changes, fixes, and examinations. This incorporates support type, date, cost, and portrayal.

Fuel: Clients can follow fuel buys, including fuel type, amount, cost, and date. This assists in observing with filling utilization and costs.

Insurance: The framework can store data about vehicle protection, including the protection supplier, contract number, inclusion dates, and that's just the beginning.

Service and Examination: Clients can produce reports and investigation to dissect vehicle upkeep chronicles, eco-friendliness, costs, and that's just the beginning.

Easy to understand Connection point: The framework generally offers a natural connection point that improves on information section, recovery, and the board.

Benefits:

Efficient Data management: A vehicle the executives' framework incorporates all vehicle-related information, making it simple to sort out and get to data.

Cost Observing: Clients can follow costs connected with fuel, support, fixes, and protection, assisting with overseeing costs actually.

Evolved planning: Examination and reports give bits of knowledge into vehicle execution and costs, empowering better preparation and navigation.

Consistence and Guidelines: The framework can assist clients with remaining consistent with vehicle enlistment, protection, and support guidelines.

Accuracy of data: With organized information section, the framework diminishes the possibilities of blunders and irregularities in record-keeping.

Documentation: The framework gives a computerized record of vehicle history, making it more straightforward to keep up with documentation for reviews and resale.

Generally a Vehicle The executives Framework offers a helpful method for coordinating and track vehicle-related data, prompting more effective vehicle tasks, cost reserve funds, and further developed direction. The elements and advantages can fluctuate contingent upon the particular necessities and execution of the framework.

Problem Statement

There are three potential issues. The absence of a database management system for vehicle management information is the first issue. The second issue is that manually generating reports takes more time. The booking of incorrect automobiles is the third issue. For instance, managing their client and vehicle details in forms results in a lot of paperwork for them. Additionally, this takes time away from them when they look up the information about the customer and the car in question. Additionally, the approach they are using has no backup because the forms they have may be lost or damaged. To create a database management system that will allow users to maintain records or data utilizing the program to manage their vehicle management.

Objectives

1. To develop a database management system to control their car information.
2. To develop new management-helping features, such as reports.
3. To use the database system to enhance the administration of vehicles.

The client level and application scope are the two task extents. Client scopes the client as being prepared to handle the data in the Vehicle Management System. The features of the vehicle management system, including as reports and record-keeping of vehicle information, make up the system's scope. VMS is a standalone system for a single user that doesn't need a network or a sophisticated database system. There won't be any more blunders when consumers are given the wrong vehicles, according to the assumptions made about how fast and effectively the vehicle management and car rental processes will be completed. Since the system contains all the information on any servicing, the management won't forget about any crucial deadlines pertaining to the vehicle. The system can only be used with Windows OS, which is one of the project's restrictions. Since it is not an internet-based database system, it can only be viewed at the specific computer where it was installed.

Methodology

Designing a database schema, putting it into use with a DBMS, and creating an application for users to interact with the data are all steps in the creation of a database management system for vehicle management. An overview of the general steps you could take to design such a system is provided below:

Requirements

Define the precise specifications for your vehicle management system. Establish the sorts of data that must be saved, the user interface, and the required functionalities.

Designing

Designing the database's schema, which includes its entities, properties, relationships, and data types, is known as database design. To cut down on redundant information and boost data integrity, normalize the schema.

- Choosing the Software

Choose a DBMS: Decide on DBMS, to use to put your schema into practice. SQLite, Microsoft SQL Server, PostgreSQL, and MySQL are popular choices.

- Database Implementation:

Based on your database design, build the tables and relationships in the system you want DBMS. To design the schema and construct relationships, write SQL statements.

- Development of User Interface:

Design an interactive interface that enables users to communicate with the database. This might be a mobile app, desktop program, or both.

- Create, Read, Update, and Delete (CRUD)

Crud actions should be implemented to let users add, view, change, and delete car records.

- Data Validation and Error Handling:

Create validation rules to make sure the information entered by users satisfies specific requirements. Include tools for handling errors as well to deal with unforeseen circumstances.

- Data set Execution:

Make the tables and connections in your picked DBMS in light of your data set plan. Compose SQL articulations to characterize the composition and lay out connections.

- Client Confirmation and Approval:

Carry out client confirmation and approval to guarantee secure admittance to the framework. Characterize client jobs and authorizations.

- Information Approval and Mistake Taking care of:

Execute approval decides to guarantee that information entered by clients meets specific standards. Likewise, incorporate blunder dealing with components to oversee surprising situations.

- Reports and Examination:

Foster highlights that permit clients to produce reports and break down information, for example, upkeep chronicles, fuel utilization reports, and the sky is the limit from there.

- Testing:

Completely test the framework to guarantee that it works accurately, information is put away precisely, and client communications fill in true to form.

- Preparing and Documentation:

Give preparing materials and documentation to clients on the most proficient method to utilize the framework actually.

- Service, Maintenance and Updates:

Consistently keep up with and update the framework to address any bugs, add new elements, and guarantee its smooth activity after some time.

Designing a data base system:

Entities

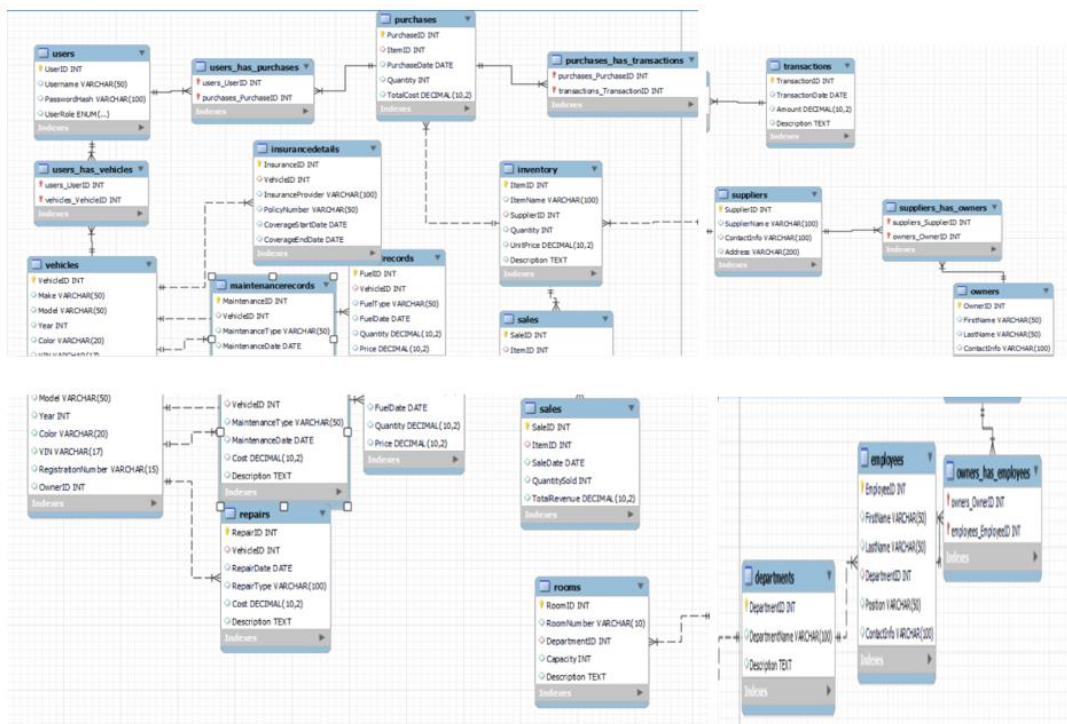
- Vehicles
- Owners
- Maintenance Records
- Fuel Records
- Insurance Details
- Repairs
- Users
- Departments
- Employees
- Rooms
- Suppliers

- Inventory
- Purchases
- Sales
- Transaction

Primary Keys and Foreign Keys

ENTITY	PRIMARY KEY	FOREIGN KEY
• Vehicles	Vehicle ID	Owner ID
• Owners	Owner ID	-
• Maintenance Records	Maintenance ID	Vehicle ID
• Fuel Records	Fuel ID	Vehicle ID
• Insurance Details	Insurance ID	Vehicle ID
• Repairs	RepairID	-
• Users	User ID	-
• Departments	Department ID	Vehicle ID
• Employees	Employee ID	Department ID
• Rooms	Room ID	-
• Suppliers	Supplier ID	Department ID
• Inventory	Item ID	Supplier ID
• Purchases	Purchase ID	Item ID
• Sales	Sale ID	Item ID
• Transaction	Transaction ID	-

ER diagram



ER diagram summary

The supplied Entity-Relationship (ER) diagram shows a network of interrelated entities and how they interact in a complete vehicle management system. This system is made to manage many aspects of cars, buildings, and related operations efficiently.

This system's foundation is centered on the idea of "Vehicles and Ownership." A VehicleID, which serves as the primary key, identifies each vehicle in a special way. OwnerID serves as the primary key for the storage of the Owners' data, which is essential for ownership monitoring. The OwnerID foreign key, which is contained in the cars table and establishes a connection between cars and their owners, links ownership information directly to the corresponding vehicles.

The system also takes thorough care of the important recordkeeping and maintenance tasks. In order to precisely record information about vehicle maintenance and fueling activities, the MaintenanceRecords and FuelRecords tables are introduced. The MaintenanceID and FuelID primary keys on these tables serve as independent identifiers. The VehicleID foreign key is used to preserve the connection between these entries and the related automobiles.

The system expands its ability to cover maintenance and insurance. Information on insurance and repairs is stored in the Insurance and Repairs tables, respectively. InsuranceID and RepairID serve as the primary keys that identify each table. The VehicleID is a foreign key maintains the link between these files and the relevant automobiles, serving as a crucial link between vehicle data and the insurance and repair information that go along with it. Another crucial component is user management, which handles user authentication and system-wide role management. The UserID serves as the fundamental key for the Users table, which contains data about users. The UserRole column stores user roles, which are classified according to "Admin" or "User." This specifies the level of access and responsibility for each user as well as the organizational hierarchy.

Employees are a part of these departments, which act as organizational units. A distinct DepartmentID is used to identify each department, while the EmployeeID serves as the main key for employee information. This organizational structure facilitates effective staff administration within designated departments. With the Supplier & Inventory entities, the system also takes into account the dynamics of the supply chain. The primary key and identifier for suppliers is their SupplierID. Items are tracked by the Inventory object and are recognized through the ItemID primary key. The SupplierID foreign key establishes the connection between suppliers and inventory, facilitating efficient supply chain management.

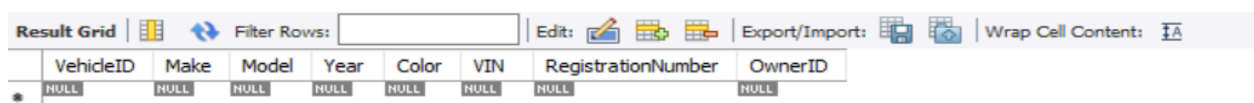
The ER diagram essentially depicts a strong and integrated structure for controlling vehicles, facilities, operations, and numerous operational elements. It provides a comprehensive approach to efficient data tracking, administration, and submitting reports, covering the various demands of a contemporary vehicle management system.

Queries that can help in decision making process in vehicle management system:

1. Retrieve vehicle information

```
#Retrieve Vehicle Information
SELECT * FROM Vehicles;
```

This search retrieves all vehicle-related data that is kept in the database.



VehicleID	Make	Model	Year	Color	VIN	RegistrationNumber	OwnerID
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

It allows users to examine information about each car registered in the system, including all attribute pertinent data.

2. Find total maintenance cost

```
#Find Total Maintenance Cost
SELECT SUM(Cost) AS TotalMaintenanceCost FROM MaintenanceRecords;
```

The overall cost of vehicle maintenance is determined by this query.

TotalMaintenanceCost
NULL

The Maintenance Records table's 'Cost' field is added together to create a consolidated value that represents the total amount spent on vehicle maintenance.

3. List of employees by department

```
-- List Employees by Department
SELECT E.FirstName, E.LastName, D.DepartmentName
FROM Employees AS E
JOIN Departments AS D ON E.DepartmentID = D.DepartmentID;
```

This query combines data from the Departments and Employees tables.

FirstName	LastName	DepartmentName
-----------	----------	----------------

It provides an understandable description of the organization's employment structure by listing the names of the staff together with the corresponding department titles they are linked with.

4. Get fuel records by vehicle

```
-- Get Fuel Records by Vehicle
SELECT F.*, V.Make, V.Model
FROM FuelRecords AS F
JOIN Vehicles AS V ON F.VehicleID = V.VehicleID;
```

This query offers a detailed picture of fuelling operations for each vehicle by connecting the Fuel Records and Cars tables.

FuelID	VehicleID	FuelType	FuelDate	Quantity	Price	Make	Model
--------	-----------	----------	----------	----------	-------	------	-------

5. Count the number of repairs by vehicle

The overall amount of repairs made for every car is determined by this query.

Make	Model	NumberOfRepairs
------	-------	-----------------

The number of fixes for each car, together with their Make and Model, are displayed thanks to an LEFT JOIN transaction between the Automobiles and Repairs tables.

```
-- Count the Number of Repairs per Vehicle
-- Retrieve Transactions by Date Range
SELECT *
FROM Transactions
WHERE TransactionDate BETWEEN '2023-01-01' AND '2023-12-31';
```

6. Retrieve transactions by date

This query retrieves all transactions that were logged over the time frame in question period.

This enables users to get financial and operational data over a specified time period by filtering their Transactions table dependent on the Transaction Date field.

7. Find highest selling item

This search reveals the stock of the item that sells the most units.

```
-- Find Highest Selling Item
SELECT I.ItemName, COUNT(S.SaleID) AS TotalSales
FROM Inventory AS I
LEFT JOIN Sales AS S ON I.ItemID = S.ItemID
GROUP BY I.ItemName
ORDER BY TotalSales DESC
LIMIT 1;
```

It identifies the product that had been sold most often along with the number of its sales by combining the Supply and Sales columns and grouping outcomes by ItemName.

ItemName	TotalSales
----------	------------

8. Get user information

```
-- Get User Information
SELECT UserID, Username, UserRole
FROM Users
WHERE UserRole = 'Admin';
```

The 'Admin' role users are the subject of this query, which is primarily concerned with user management.

UserID	Username	UserRole
NULL	NULL	NULL

```
-- List Rooms and Associated Department
SELECT R.RoomID, R.RoomNumber, D.DepartmentName
FROM Rooms AS R
JOIN Departments AS D ON R.DepartmentID = D.DepartmentID;
```

9. List rooms and associated department

To display details about bedrooms and the departments they belong to, this query integrates data from the Departments and Rooms tables.

RoomID	RoomNumber	DepartmentName
--------	------------	----------------

It provides information like RoomID, RoomNumber, and the name of the appropriate department.

10. Retrieve vehicle and maintenance records

```
-- Retrieve Vehicle and Maintenance Records
SELECT V.VehicleID, V.Make, V.Model, MR.MaintenanceType, MR.MaintenanceDate
FROM Vehicles AS V
JOIN MaintenanceRecords AS MR ON V.VehicleID = MR.VehicleID;
```

VehicleID	Make	Model	MaintenanceType	MaintenanceDate
-----------	------	-------	-----------------	-----------------

To offer an integrated overview of vehicle and upkeep informationIt provides details about each vehicle's maintenance history by displaying VehicleID, Make, Model, MaintenanceType, and MaintenanceDate.

References

<https://ieeexplore.ieee.org/document/8825030>

<https://www.slideshare.net/mohdasaddam/vehicle-management-system>

<https://www.inettutor.com/diagrams/vehicle-repair-and-maintenance-management-system-database-design/>

<http://www.diva-portal.org/smash/get/diva2:121283/FULLTEXT01.pdf>