**ADVANCED DATABASE PROJECT**

**Vehicle Repair and Maintenance Database Documentation**

This document outlines the design, schema, SQL queries, and functionalities implemented for your vehicle repair and maintenance database system. It includes explanations and rationales behind the design decisions and implementation choices.

**1. Database Design**

The database utilizes a relational model with tables linked through foreign keys to maintain data integrity and facilitate efficient data retrieval. Here's a breakdown of the tables:

* **Customers:** Stores customer information such as name, phone number, and email address.
* **Vehicles:** Stores vehicle details like make, model, year, license plate, VIN, and a foreign key referencing the customer who owns the vehicle (linking it to the Customers table).
* **Services:** Stores information about services offered by the company, including service name, description, and labor cost.
* **Repairs:** Records repair events, with foreign keys referencing the Vehicles table (vehicle serviced) and the Services table (service performed).
* **Payments:** Tracks payments made for repairs, with a foreign key referencing the Repairs table (to link the payment to a specific repair) and potentially a foreign key referencing the Customers table (to directly link the payment to the customer).
* **Technicians:** Stores information about technicians performing repairs, including name, hire date, and hourly rate.
* **Inventory:** Manages parts used in repairs, with details like part name, description (optional), cost per unit, quantity in stock, and reorder point.

**Rationale:**

* This design separates customer and vehicle information, allowing a customer to own multiple vehicles.
* Linking services and repairs allows tracking services performed on specific vehicles.
* Payments are associated with repairs, ensuring proper financial record-keeping.
* The optional tables provide additional functionalities if needed, such as managing technicians and parts inventory.

**2. Database Schema**

The schema visually represents the database structure, including tables, columns, data types, and relationships between tables using primary keys and foreign keys.

**Here, you can include a visual schema diagram similar to what we discussed earlier.**

**Explanation:**

* The schema diagram provides a clear understanding of how data is organized and connected within the database.
* Foreign keys ensure data consistency by referencing primary keys in other tables.

**3. SQL Queries**

The system utilizes SQL queries for various purposes, including:

* **Inserting data:** Adding new customers, vehicles, services, repairs, payments, technicians (optional), and inventory items (optional).
* **Updating data:** Modifying existing customer, vehicle, service, repair, payment, technician (optional), or inventory (optional) information.
* **Deleting data:** Removing customer, vehicle, service, repair, payment, technician, or inventory data (considering referential integrity constraints).
* **Selecting data:** Retrieving specific customer information, vehicle details, service descriptions, repair history, payment records, technician information, or parts inventory.

**Examples:**

We previously discussed the CalculateTotalRepairCost stored procedure that demonstrates how to automate complex calculations. You can include additional examples of SQL queries relevant to your specific needs.

**4. Additional Features**

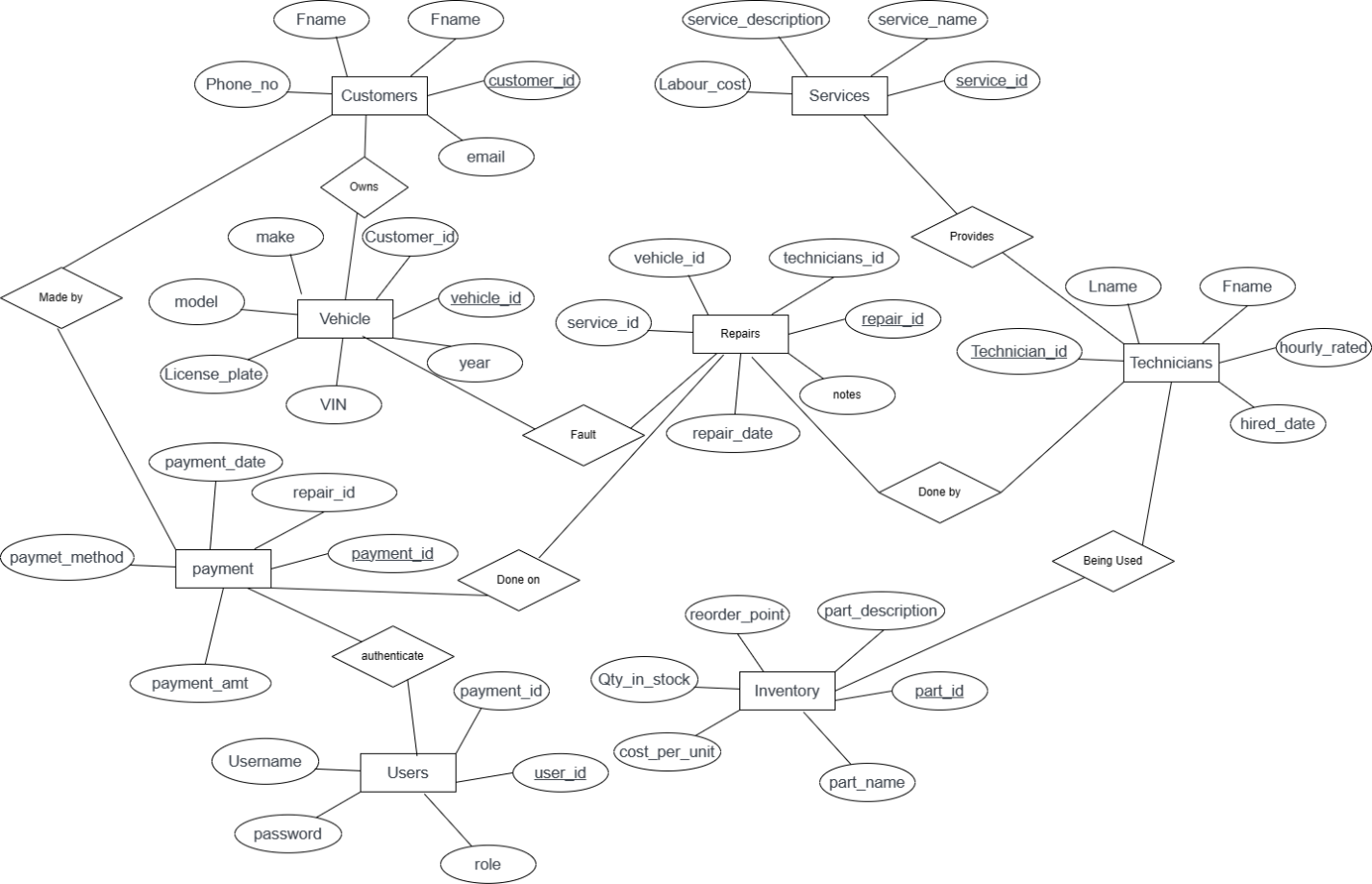
The database can be enhanced with functionalities beyond the core tables and queries. Here are some examples:

* **Triggers:** Enforce data integrity and business rules. For instance, a trigger could prevent deleting a vehicle if it has open repairs.
* **Functions:** Perform calculations or validations as needed. A function could validate a customer's phone number format before inserting it into the database.
* **Stored procedures:** Automating frequently executed tasks, such as calculating total repair cost.

**Rationale:**

* These features improve data integrity, automate tasks, and enhance overall system usability.

### Entity Relational Model



**5. Updated Section**

* Added a **user authentication** table to authenticate users during payment.

It keep truck of the person who made the record of payment after repair, thus either

the database administrator or a database user or the customer.

* Updated ER-Diagram.

**6. Conclusion**

This documentation provides a comprehensive overview of your vehicle repair and maintenance database design, schema, SQL functionalities, and potential additional features. The design choices and implementation decisions were made to ensure data integrity, efficient data retrieval, and scalability to accommodate future needs.