AutoMart Sales and Service Management System (ASSMS)

Database Design and Development Document

1. Project Overview

The **AutoMart Sales and Service Management System (ASSMS)** is a comprehensive database solution for a car dealership that manages both vehicle sales and spare parts inventory. This system ensures seamless operations by keeping track of cars, spare parts, sales transactions, customers, and employees. The database will enable efficient data retrieval, sales tracking, and inventory management.

2. Database Requirements

Entities:

1. **Car Inventory**:
    a. Tracks all available cars for sale, their details, and their price.
2. **Spare Parts Inventory**:
    a. Manages the stock of spare parts available for sale, including descriptions and pricing.
3. **Customers**:
    a. Keeps records of customer information for tracking purchases, returns, and service requests.
4. **Sales Transactions**:
    a. Stores details about every car or spare part sale, associating it with the customer, employee, and date of sale.
5. **Employees**:
    a. Stores information about employees handling sales or working in inventory management.
6. **Suppliers**:

a. Manages information about suppliers providing cars and spare parts to the dealership.

3. Database Tables

Below are the primary tables for **ASSMS**, including the main attributes for each.

1. car Table

- **Description**: Stores details about the cars available for sale.
- **Attributes**:
  - car_id (Primary Key, Auto Increment)
  - make (e.g., Toyota, BMW)
  - model (e.g., Camry, X5)
  - year
  - price
  - status (e.g., available, sold)

2. spare_parts Table

- **Description**: Manages the details of spare parts in inventory.
- **Attributes**:
  - part_id (Primary Key, Auto Increment)
  - part_name
  - part_description
  - quantity_in_stock
  - price
  - supplier_id (Foreign Key)

3. customers Table

- **Description**: Holds details of customers.
- **Attributes**:
  - customer_id (Primary Key, Auto Increment)
  - first_name
  - last_name
  - phone_number
  - email
  - address

4. employees Table

- **Description**: Stores data of employees who handle sales and inventory.
- **Attributes**:
  - employee_id (Primary Key, Auto Increment)
  - employee_name
  - position
  - phone_number
  - email

5. suppliers Table

- **Description**: Holds details of suppliers for cars and spare parts.
- **Attributes**:
  - supplier_id (Primary Key, Auto Increment)
  - supplier_name
  - contact_number
  - email
  - address

6. car_sales Table

- **Description**: Records all car sales.
- **Attributes**:
  - sale_id (Primary Key, Auto Increment)
  - car_id (Foreign Key)
  - customer_id (Foreign Key)
  - employee_id (Foreign Key)
  - sale_date
  - sale_price

7. part_sales Table

- **Description**: Manages the sale of spare parts.
- **Attributes**:
  - sale_id (Primary Key, Auto Increment)
  - part_id (Foreign Key)
  - customer_id (Foreign Key)
  - employee_id (Foreign Key)
  - quantity

- sale_date
- total_price

4. Relationships Between Tables

1. **car_sales** has a many-to-one relationship with the **car**, **customers**, and **employees** tables.
    a. A car can be sold once, but many cars can be sold to one customer.
    b. Multiple employees can handle different sales.
2. **part_sales** has a many-to-one relationship with **spare_parts**, **customers**, and **employees**.
    a. Multiple parts can be sold in a single transaction, but each transaction is handled by one employee.
3. **suppliers** table has a one-to-many relationship with the **spare_parts** table.
    a. One supplier can provide multiple spare parts.

5. ER Diagram

An Entity-Relationship Diagram (ERD) will be generated to visually represent the relationships between tables. The following relationships will be depicted:

- **One-to-Many**: Customers to Sales, Employees to Sales, and Suppliers to Spare Parts.
- **Many-to-One**: Cars/Parts to Sales.