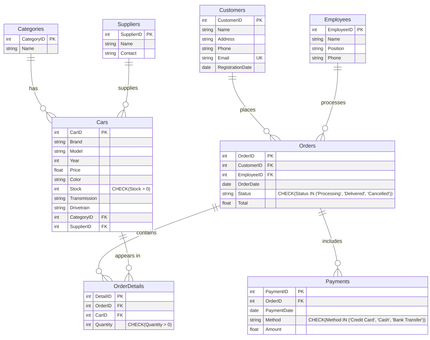
Car Dealership Database Documentation

Entity-Relationship Diagram



Relational Model

Table Name	Attributes	Keys and Constraints
Cars	CarID, Brand, Model, Year, Price, Color, Stock, Transmission, Drivetrain, CategoryID, SupplierID	**Primary Key:** CarID **Foreign Keys:** CategoryID, SupplierID **Constraints:** Stock >= 0
Categories	CategoryID, Name	**Primary Key:** CategoryID
Customers	CustomerID, Name, Address, Phone, Email, RegistrationDate	**Primary Key:** CustomerID **Unique:** Email
Employees	EmployeeID, Name, Position, Phone	**Primary Key:** EmployeeID
Suppliers	SupplierID, Name, Contact	**Primary Key:** SupplierID
Orders	OrderID, CustomerID, EmployeeID, OrderDate, Status, Total	**Primary Key:** OrderID **Foreign Keys:** CustomerID, EmployeeID **Constraints:** Status IN ('Processing', 'Delivered', 'Cancelled')
Payments	PaymentID, OrderID, PaymentDate, Method, Amount	**Primary Key:** PaymentID **Foreign Key:** OrderID **Constraints:** Method IN ('Credit Card', 'Cash', 'Bank Transfer')
OrderDetails	DetailID, OrderID, CarID, Quantity	**Primary Key:** DetailID **Foreign Keys:** OrderID, CarID **Constraint:** Quantity > 0

SQL Code

1. Creating the Database

```
CREATE DATABASE CarDealership;
USE CarDealership;
```

2. Creating the Tables

```
-- Create Categories table
CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY,
    Name VARCHAR(100)
);

-- Create Suppliers table
CREATE TABLE Suppliers (
    SupplierID INT PRIMARY KEY,
    Name VARCHAR(100),
    Contact VARCHAR(100)
);

-- Create Cars table
CREATE TABLE Cars (
    CarID INT PRIMARY KEY,
    Brand VARCHAR(100),
    Model VARCHAR(100),
    Year INT,
    Price FLOAT,
    Color VARCHAR(50),
    Stock INT CHECK (Stock >= 0),
    Transmission VARCHAR(50),
    Drivetrain VARCHAR(50),
    CategoryID INT,
    SupplierID INT,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID),
    FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)
);

-- Create Customers table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(200),
    Phone VARCHAR(15),
    Email VARCHAR(100) UNIQUE,
    RegistrationDate DATE
);

-- Create Employees table
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Position VARCHAR(50),
    Phone VARCHAR(15)
);

-- Create Orders table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    EmployeeID INT,
    OrderDate DATE,
    Status VARCHAR(20) CHECK (Status IN ('Processing', 'Delivered', 'Cancelled')),
    Total FLOAT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);

-- Create Payments table
CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY,
    OrderID INT,
    PaymentDate DATE,
    Method VARCHAR(20) CHECK (Method IN ('Credit Card', 'Cash', 'Bank Transfer')),
    Amount FLOAT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
);

-- Create OrderDetails table
CREATE TABLE OrderDetails (
    DetailID INT PRIMARY KEY,
    OrderID INT,
    CarID INT,
    Quantity INT CHECK (Quantity > 0),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (CarID) REFERENCES Cars(CarID)
);
```

3. Populating the Tables

```
-- Insert categories data first
INSERT INTO Categories VALUES
(1, 'SUV'),
(2, 'Sedan'),
(3, 'Truck');

-- Insert suppliers data
INSERT INTO Suppliers VALUES
(1, 'Toyota Supplier', 'contact@toyotasupplier.com'),
(2, 'Honda Supplier', 'contact@hondasupplier.com'),
(3, 'Ford Supplier', 'contact@fordsupplier.com');

-- Insert cars data (ensure CategoryID exists in Categories)
INSERT INTO Cars VALUES
(1, 'Toyota', 'RAV4', 2021, 30000, 'Blue', 10, 'Automatic', 'AWD', 1, 1),
(2, 'Honda', 'Civic', 2020, 20000, 'Red', 5, 'Manual', 'FWD', 2, 2),
(3, 'Ford', 'F-150', 2022, 35000, 'Black', 8, 'Automatic', '4WD', 3, 3),
(4, 'Toyota', 'Corolla', 2021, 22000, 'White', 12, 'Automatic', 'FWD', 2, 1),
(5, 'Honda', 'Accord', 2021, 28000, 'Silver', 7, 'Automatic', 'AWD', 2, 2);

-- Insert customers data
INSERT INTO Customers VALUES
(1, 'John Doe', '123 Elm St', '555-1234', 'john.doe@example.com', '2023-01-15'),
(2, 'Jane Smith', '456 Oak Ave', '555-5678', 'jane.smith@example.com', '2023-02-20'),
(3, 'Alice Johnson', '789 Pine Blvd', '555-8765', 'alice.johnson@example.com', '2023-03-10');

-- Insert employees data
INSERT INTO Employees VALUES
(1, 'Mark Adams', 'Sales Manager', '555-1122'),
(2, 'Sarah Miller', 'Sales Associate', '555-3344'),
(3, 'James Wilson', 'Customer Service', '555-5566');

-- Insert orders data
INSERT INTO Orders VALUES
(1, 1, 2, '2023-04-01', 'Processing', 20000),
(2, 2, 3, '2023-04-10', 'Delivered', 22000),
(3, 3, 1, '2023-04-15', 'Cancelled', 28000);

-- Insert payments data
INSERT INTO Payments VALUES
(1, 1, '2023-04-02', 'Credit Card', 20000),
(2, 2, '2023-04-12', 'Bank Transfer', 22000),
(3, 3, '2023-04-16', 'Cash', 28000);

-- Insert order details data (ensure CarID exists in Cars)
INSERT INTO OrderDetails VALUES
(1, 1, 1),  -- Order 1 with CarID 1
(2, 2, 1),  -- Order 2 with CarID 3
(3, 3, 2);  -- Order 3 with CarID 2
```

4. Example Queries

```
-- Query 1: Retrieve all cars and their categories
SELECT Cars.Brand, Cars.Model, Categories.Name AS Category
FROM Cars
JOIN Categories ON Cars.CategoryID = Categories.CategoryID;

-- Query 2: Get all orders with customer names
SELECT Orders.OrderID, Customers.Name, Orders.Total
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

-- Query 3: Total sales grouped by category
SELECT Categories.Name AS Category, SUM(Orders.Total) AS TotalSales
FROM Orders
JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
JOIN Cars ON OrderDetails.CarID = Cars.CarID
JOIN Categories ON Cars.CategoryID = Categories.CategoryID
GROUP BY Categories.Name;

-- Query 4: Orders sorted by date
SELECT * FROM Orders
ORDER BY OrderDate DESC;

-- Query 5: Retrieve all suppliers and the cars they supply
SELECT Suppliers.Name AS Supplier, Cars.Brand, Cars.Model
FROM Cars
JOIN Suppliers ON Cars.SupplierID = Suppliers.SupplierID;

-- Query 6: Get the total payments made by each customer
SELECT Customers.Name, SUM(Payments.Amount) AS TotalPayments
FROM Payments
JOIN Orders ON Payments.OrderID = Orders.OrderID
JOIN Customers ON Orders.CustomerID = Customers.CustomerID
GROUP BY Customers.Name;

-- Query 7: Find cars with stock below a certain threshold
SELECT Brand, Model, Stock
FROM Cars
WHERE Stock < 5;

-- Query 8: List employees and their positions
SELECT Name, Position FROM Employees;
```