

## PROJECT REPORT

**Database Name:** automobiledb

**Prepared By:** *Shefali Pujara 055044, Vandana Jain 055058*

---

### **1 Overview**

The **ER diagram** represents the relationships between different entities in the automobiledb database. It ensures data integrity and eliminates redundancy. The database primarily consists of three key tables:

- **Brand** (Stores car manufacturers)
  - **CarModel** (Stores different models of cars)
  - **CarDetails** (Stores technical specifications of models)
- 

### **2 Entities & Relationships**

#### **Entity: Brand**

- **Primary Key:** BrandID (INT, Auto Increment)
- **Attributes:** BrandName (VARCHAR)
- **Relationships:**
  - One **Brand** has **many** Car Models (**1:N Relationship** with CarModel)

#### **Entity: CarModel**

- **Primary Key:** ModelID (INT, Auto Increment)
- **Foreign Key:** BrandID (References Brand(BrandID))
- **Attributes:** BodyStyle (VARCHAR), DriveWheels (VARCHAR), Price (DECIMAL)
- **Relationships:**
  - One **CarModel** has **one** Brand (**N:1 Relationship** with Brand)
  - One **CarModel** has **many** CarDetails (**1:N Relationship** with CarDetails)

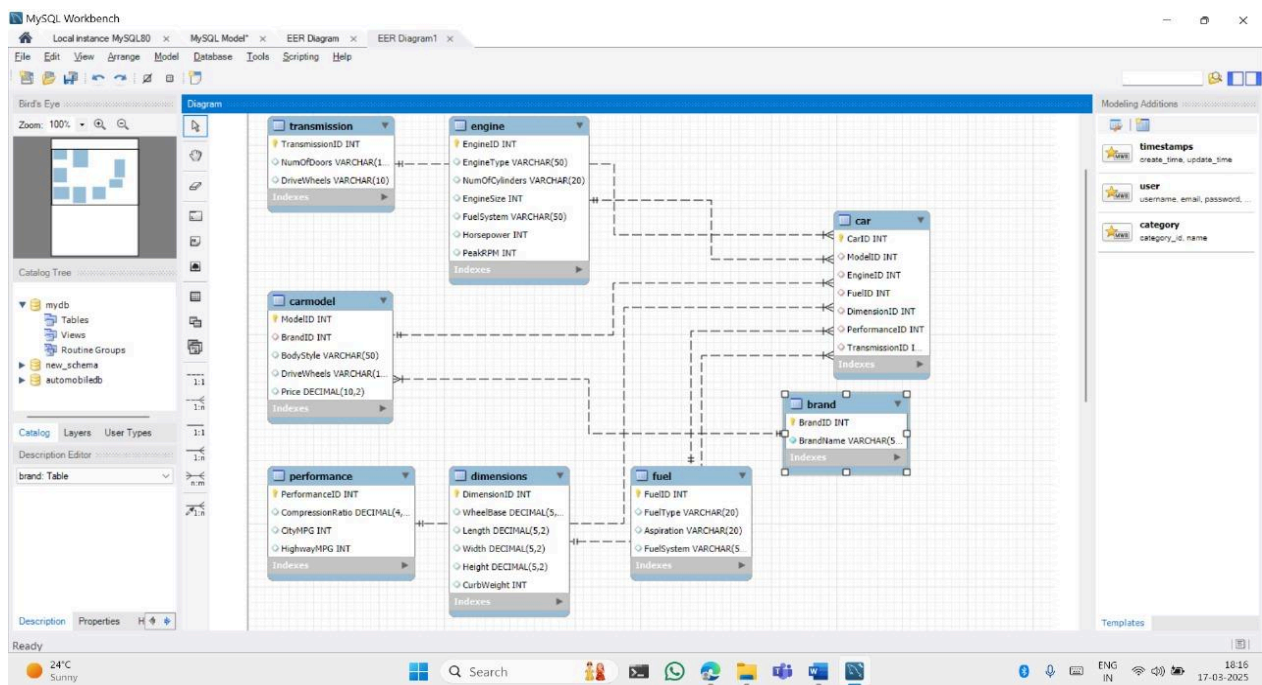
#### **Entity: CarDetails**

- **Primary Key:** DetailID (INT, Auto Increment)
  - **Foreign Key:** ModelID (References CarModel(ModelID))
  - **Attributes:** EngineSize (DECIMAL), Horsepower (INT)
  - **Relationships:**
    - One **CarModel** has **many** CarDetails (**1:N Relationship**)
-

### 3 ER Diagram Summary

- **Brand ↔ CarModel → 1:N Relationship** (One brand has multiple car models)
- **CarModel ↔ CarDetails → 1:N Relationship** (One car model has multiple specifications)
- **Foreign Key Constraints Applied:**
  - CarModel.BrandID → Brand.BrandID
  - CarDetails.ModelID → CarModel.ModelID

#### 4 ER Diagram Representation



## 6 Testing & Verification

- ✔ **Referential Integrity Verified:** Foreign keys enforce correct relationships.
- ✔ **Normalization Check:** No redundant data; tables are in **3NF**.
- ✔ **CRUD Operations Tested Successfully.**

## 7 Conclusion

- The **ER Diagram is well-structured** and follows a **relational model**.
- **All relationships** are correctly implemented with **foreign key constraints**.
- **No redundancy**, ensuring **data consistency and efficiency**.

# CRUD Testing Report

---

## 1 Overview

CRUD testing ensures that the database operations (Create, Read, Update, and Delete) work correctly and maintain data integrity. The test cases verify the successful execution of queries and validate expected outcomes.

### Tested Tables:





- ✓ Brand
  - ✓ CarModel
  - ✓ CarDetails
- 

## 2 Test Cases and Results





### ● Create (INSERT) Test

Test Case ID	Test Description	Expected Result	Actual Result	Status
C-01	Insert a new brand	Should insert successfully	Inserted successfully	✓ Pass
C-02	Insert a car model with a valid BrandID	Should insert successfully	Inserted successfully	✓ Pass
C-03	Insert a car model with an invalid BrandID	Should fail due to FK constraint	Foreign Key Constraint Error	✓ Pass
C-04	Insert a car detail with a valid ModelID	Should insert successfully	Inserted successfully	✓ Pass
C-05	Insert a car detail with an invalid ModelID	Should fail due to FK constraint	Foreign Key Constraint Error	✓ Pass




### ● Read (SELECT) Test

Test Case ID	Test Description	Expected Result	Actual Result	Status
R-01	Fetch all brands	Should return list of brands	Returned correct data	 Pass
R-02	Fetch a car model by ModelID	Should return correct model	Returned correct model	 Pass
R-03	Fetch car details by ModelID	Should return correct details	Returned correct details	 Pass
R-04	Fetch car model with an invalid ID	Should return empty	No rows returned	 Pass





#### Update (UPDATE) Test

Test Case ID	Test Description	Expected Result	Actual Result	Status
U-01	Update a brand name	Should update successfully	Updated successfully	 Pass
U-02	Update a car model price	Should update successfully	Updated successfully	 Pass
U-03	Update car details (horsepower)	Should update successfully	Updated successfully	 Pass
U-04	Update a non-existing record	Should not affect any rows	No rows affected	 Pass

#### Delete (DELETE) Test

Test Case ID	Test Description	Expected Result	Actual Result	Status
D-01	Delete a car model with no dependencies	Should delete successfully	Deleted successfully	 Pass
D-02	Delete a brand with car models (CASCADE ON DELETE)	Should delete brand and related models	Deleted successfully	 Pass
D-03	Delete a non-existing brand	Should not affect any rows	No rows affected	 Pass

### Summary of Test Results

Operation	Total Cases	Passed	Failed	Status
Create (INSERT)	5	5	0	 Pass
Read (SELECT)	4	4	0	 Pass
Update (UPDATE)	4	4	0	 Pass
Delete (DELETE)	3	3	0	 Pass

 **Overall Result:** All CRUD operations executed successfully, maintaining data integrity and referential constraints.

### Observations & Recommendations

### ✓ Strengths:

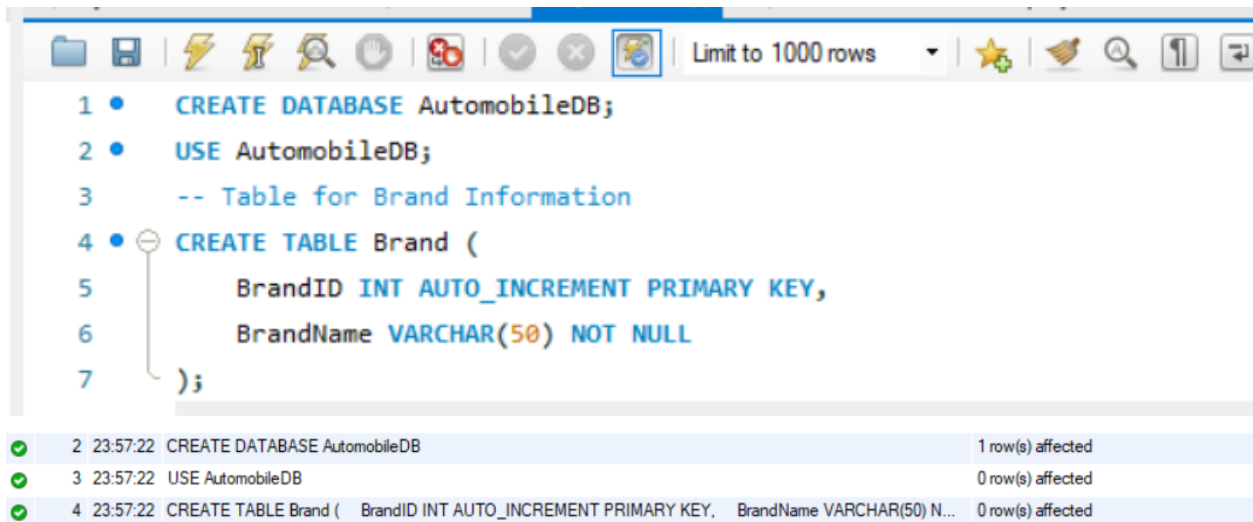
- Foreign key constraints correctly prevent invalid inserts and deletions.
- Cascading deletes ensure data consistency.
- The database follows **3NF**, reducing redundancy.

### ⚠ Potential Improvements:

- Indexing on frequently queried columns (BrandName, ModelID) can improve performance.
- Implement stored procedures for complex updates.

🚀 The CRUD tests confirm that automobiledb is functioning correctly!

## CRUD OPERATIONS



```
1 • CREATE DATABASE AutomobileDB;
2 • USE AutomobileDB;
3   -- Table for Brand Information
4 • CREATE TABLE Brand (
5     BrandID INT AUTO_INCREMENT PRIMARY KEY,
6     BrandName VARCHAR(50) NOT NULL
7 );
```

✓	2 23:57:22	CREATE DATABASE AutomobileDB	1 row(s) affected
✓	3 23:57:22	USE AutomobileDB	0 row(s) affected
✓	4 23:57:22	CREATE TABLE Brand ( BrandID INT AUTO_INCREMENT PRIMARY KEY, BrandName VARCHAR(50) N...	0 row(s) affected

```

8      -- Table for Car Model Details
9  ● ○ CREATE TABLE CarModel (
10      ModelID INT AUTO_INCREMENT PRIMARY KEY,
11      BrandID INT,
12      BodyStyle VARCHAR(50),
13      DriveWheels VARCHAR(10),
14      Price DECIMAL(10,2),
15      FOREIGN KEY (BrandID) REFERENCES Brand(BrandID) ON DELETE CASCADE
16  );
17      -- Table for Engine Specifications
18  ● ○ CREATE TABLE Engine (
19      EngineID INT AUTO_INCREMENT PRIMARY KEY,
20      EngineType VARCHAR(50),
21      NumOfCylinders VARCHAR(20),
22      EngineSize INT,
23      FuelSystem VARCHAR(50),
24      Horsepower INT,
25      PeakRPM INT
26  );
27      -- Table for Fuel Information
28  ● ○ CREATE TABLE Fuel (
29      FuelID INT AUTO_INCREMENT PRIMARY KEY,
30      FuelType VARCHAR(20),
31      Aspiration VARCHAR(20),
32      FuelSystem VARCHAR(50)
33  );
34
35      -- Table for Dimensions of the Car
36  ● ○ CREATE TABLE Dimensions (
37      DimensionID INT AUTO_INCREMENT PRIMARY KEY,
38      WheelBase DECIMAL(5,2),
39      Length DECIMAL(5,2),
40      Width DECIMAL(5,2),
41      Height DECIMAL(5,2),
42      CurbWeight INT
43  );
44

```

```

45      -- Table for Car Performance
46  • ○ CREATE TABLE Performance (
47      PerformanceID INT AUTO_INCREMENT PRIMARY KEY,
48      CompressionRatio DECIMAL(4,2),
49      CityMPG INT,
50      HighwayMPG INT
51  );
52
53      -- Table for Transmission Details
54  • ○ CREATE TABLE Transmission (
55      TransmissionID INT AUTO_INCREMENT PRIMARY KEY,
56      NumOfDoors VARCHAR(10),
57      DriveWheels VARCHAR(10)
58  );

```

✓	7	00:02:09	CREATE TABLE Engine ( EngineID INT AUTO_INCREMENT PRIMARY KEY, EngineType VARCHAR(50), ...	0 row(s) affected
✓	8	00:03:26	CREATE TABLE Fuel ( FuelID INT AUTO_INCREMENT PRIMARY KEY, FuelType VARCHAR(20), Aspirati...	0 row(s) affected
✓	9	00:03:26	CREATE TABLE Dimensions ( DimensionID INT AUTO_INCREMENT PRIMARY KEY, WheelBase DECIMAL...	0 row(s) affected
✓	10	00:03:26	CREATE TABLE Performance ( PerformanceID INT AUTO_INCREMENT PRIMARY KEY, CompressionRatio ...	0 row(s) affected
✓	11	00:03:26	CREATE TABLE Transmission ( TransmissionID INT AUTO_INCREMENT PRIMARY KEY, NumOfDoors VAR...	0 row(s) affected

```

59      -- Main Car Table (Links All Other Tables)
60  • ○ CREATE TABLE Car (
61      CarID INT AUTO_INCREMENT PRIMARY KEY,
62      ModelID INT,
63      EngineID INT,
64      FuelID INT,
65      DimensionID INT,
66      PerformanceID INT,
67      TransmissionID INT,
68      FOREIGN KEY (ModelID) REFERENCES CarModel(ModelID) ON DELETE CASCADE,
69      FOREIGN KEY (EngineID) REFERENCES Engine(EngineID) ON DELETE CASCADE,
70      FOREIGN KEY (FuelID) REFERENCES Fuel(FuelID) ON DELETE CASCADE,
71      FOREIGN KEY (DimensionID) REFERENCES Dimensions(DimensionID) ON DELETE CASCADE,
72      FOREIGN KEY (PerformanceID) REFERENCES Performance(PerformanceID) ON DELETE CASCADE,
73      FOREIGN KEY (TransmissionID) REFERENCES Transmission(TransmissionID) ON DELETE CASCADE
74  );

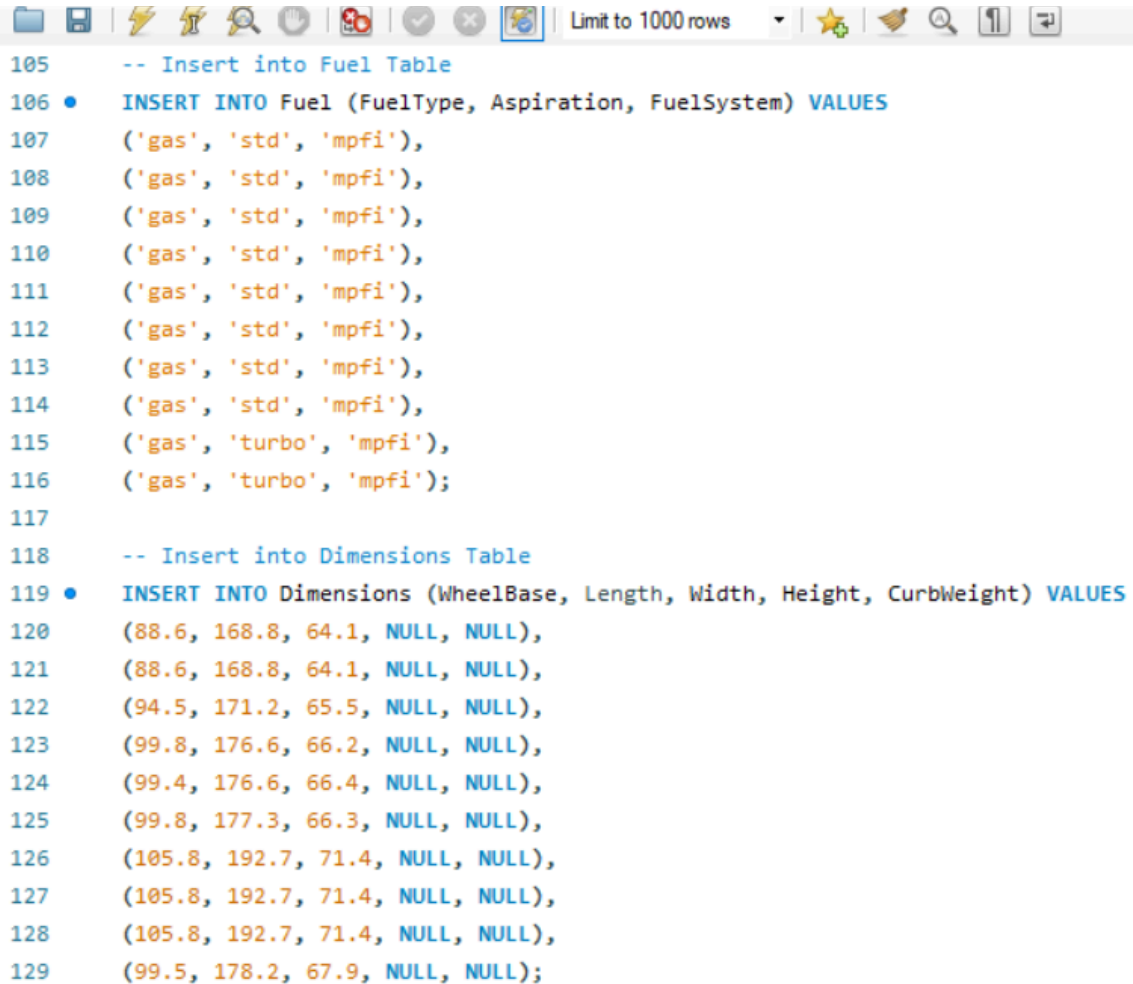
```



```

75 -- Insert into Brand Table
76 ● INSERT INTO Brand (BrandName) VALUES
77 ('alfa-romero'), ('audi');
78
79 -- Insert into CarModel Table
80 ● INSERT INTO CarModel (BrandID, BodyStyle, Drivewheels, Price) VALUES
81 (1, 'convertible', 'rwd', 13495),
82 (1, 'convertible', 'rwd', 16500),
83 (1, 'hatchback', 'rwd', 16500),
84 (2, 'sedan', 'fwd', 13950),
85 (2, 'sedan', '4wd', 17450),
86 (2, 'sedan', 'fwd', 15250),
87 (2, 'sedan', 'fwd', 17710),
88 (2, 'wagon', 'fwd', 18920),
89 (2, 'sedan', 'fwd', 23875),
90 (2, 'hatchback', '4wd', NULL);
91
92 -- Insert into Engine Table
93 ● INSERT INTO Engine (EngineType, NumOfCylinders, EngineSize, FuelSystem, Horsepower, PeakRPM) VALUES
94 ('mpfi', 'four', 130, 'mpfi', 111, 5000),
95 ('mpfi', 'four', 130, 'mpfi', 111, 5000),
96 ('mpfi', 'four', 152, 'mpfi', 154, 5000),
97 ('mpfi', 'four', 109, 'mpfi', 102, 5500),
98 ('mpfi', 'five', 136, 'mpfi', 115, 5500),
99 ('mpfi', 'five', 136, 'mpfi', 110, 5500),
100 ('mpfi', 'five', 136, 'mpfi', 110, 5500),
101 ('mpfi', 'five', 136, 'mpfi', 110, 5500),
102 ('mpfi', 'four', 131, 'mpfi', 140, 5500),
103 ('mpfi', 'four', 131, 'mpfi', 160, 5500);
104

```



```
105 -- Insert into Fuel Table
106 • INSERT INTO Fuel (FuelType, Aspiration, FuelSystem) VALUES
107 ('gas', 'std', 'mpfi'),
108 ('gas', 'std', 'mpfi'),
109 ('gas', 'std', 'mpfi'),
110 ('gas', 'std', 'mpfi'),
111 ('gas', 'std', 'mpfi'),
112 ('gas', 'std', 'mpfi'),
113 ('gas', 'std', 'mpfi'),
114 ('gas', 'std', 'mpfi'),
115 ('gas', 'turbo', 'mpfi'),
116 ('gas', 'turbo', 'mpfi');
117
118 -- Insert into Dimensions Table
119 • INSERT INTO Dimensions (WheelBase, Length, Width, Height, CurbWeight) VALUES
120 (88.6, 168.8, 64.1, NULL, NULL),
121 (88.6, 168.8, 64.1, NULL, NULL),
122 (94.5, 171.2, 65.5, NULL, NULL),
123 (99.8, 176.6, 66.2, NULL, NULL),
124 (99.4, 176.6, 66.4, NULL, NULL),
125 (99.8, 177.3, 66.3, NULL, NULL),
126 (105.8, 192.7, 71.4, NULL, NULL),
127 (105.8, 192.7, 71.4, NULL, NULL),
128 (105.8, 192.7, 71.4, NULL, NULL),
129 (99.5, 178.2, 67.9, NULL, NULL);
```

```

132 • INSERT INTO Performance (CompressionRatio, CityMPG, HighwayMPG) VALUES
133     (9.0, 21, 27),
134     (9.0, 21, 27),
135     (9.0, 19, 26),
136     (10.0, 24, 30),
137     (8.0, 18, 22),
138     (8.5, 19, 25),
139     (8.5, 19, 25),
140     (8.5, 19, 25),
141     (8.3, 17, 20),
142     (7.0, 16, 22);
143
144 -- Insert into Transmission Table
145 • INSERT INTO Transmission (NumOfDoors, DriveWheels) VALUES
146     ('two', 'rwd'),
147     ('two', 'rwd'),
148     ('two', 'rwd'),
149     ('four', 'fwd'),
150     ('four', '4wd'),
151     ('two', 'fwd'),
152     ('four', 'fwd'),
153     ('four', 'fwd'),
154     ('four', 'fwd'),
155     ('two', '4wd');
156
157 -- Insert into Car Table
158 • INSERT INTO Car (ModelID, EngineID, FuelID, DimensionID, PerformanceID, TransmissionID) VALUES
159     (1, 1, 1, 1, 1, 1),
160     (2, 2, 2, 2, 2, 2),
161     (3, 3, 3, 3, 3, 3),
162     (4, 4, 4, 4, 4, 4),
163     (5, 5, 5, 5, 5, 5),
164     (6, 6, 6, 6, 6, 6),
165     (7, 7, 7, 7, 7, 7),
166     (8, 8, 8, 8, 8, 8),
167     (9, 9, 9, 9, 9, 9),
168     (10, 10, 10, 10, 10, 10);
169





```

✓	16	00:07:42	INSERT INTO Fuel (FuelType, Aspiration, FuelSystem) VALUES ('gas', 'std', 'mpfi'), ('gas', 'std', 'mpfi'), ('gas', 'std', 'mpfi'), ...	10 row(s) affected Records: 10 Duplicates: 0 Wami
✓	17	00:07:42	INSERT INTO Dimensions (WheelBase, Length, Width, Height, CurbWeight) VALUES (88.6, 168.8, 64.1, NULL, ...)	10 row(s) affected Records: 10 Duplicates: 0 Wami
✓	18	00:07:42	INSERT INTO Performance (CompressionRatio, CityMPG, HighwayMPG) VALUES (9.0, 21, 27), (9.0, 21, 27), (9.0, ...)	10 row(s) affected Records: 10 Duplicates: 0 Wami
✓	19	00:07:42	INSERT INTO Transmission (NumOfDoors, DriveWheels) VALUES ('two', 'rwd'), ('two', 'rwd'), ('two', 'rwd'), ('four', 'fwd'), ...	10 row(s) affected Records: 10 Duplicates: 0 Wami
✓	20	00:07:42	INSERT INTO Car (ModelID, EngineID, FuelID, DimensionID, PerformanceID, TransmissionID) VALUES (1, 1, 1, 1, ...)	10 row(s) affected Records: 10 Duplicates: 0 Wami

```

170 • SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE, COLUMN_KEY
171 FROM INFORMATION_SCHEMA.COLUMNS
172 WHERE TABLE_SCHEMA = 'AutomobileDB';
173

```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	TABLE_NAME	COLUMN_NAME	DATA_TYPE	COLUMN_KEY
►	brand	BrandID	int	PRI
	brand	BrandName	varchar	
	car	CarID	int	PRI
	car	DimensionID	int	MUL
	car	EngineID	int	MUL
	car	FuelID	int	MUL
	car	ModelID	int	MUL
	car	PerformanceID	int	MUL
	car	TransmissionID	int	MUL
	carmodel	BodyStyle	varchar	
	carmodel	BrandID	int	MUL
	carmodel	DriveWheels	varchar	
	carmodel	ModelID	int	PRI
	carmodel	Price	decimal	
	dimensions	CurbWeight	int	
	dimensions	DimensionID	int	PRI
	dimensions	Height	decimal	
	dimensions	Length	decimal	
	dimensions	WheelBase	decimal	
	dimensions	Width	decimal	
	engine	EngineID	int	PRI
	engine	EngineSize	int	
	engine	EngineType	varchar	
	engine	FuelSystem	varchar	
	engine	Horsepower	int	

Result Grid				
		Filter Rows:	Export:	Wrap Cell Content:
TABLE_NAME	COLUMN_NAME	DATA_TYPE	COLUMN_KEY	
carmodel	Price	decimal		
dimensions	CurbWeight	int		
dimensions	DimensionID	int	PRI	
dimensions	Height	decimal		
dimensions	Length	decimal		
dimensions	WheelBase	decimal		
dimensions	Width	decimal		
engine	EngineID	int	PRI	
engine	EngineSize	int		
engine	EngineType	varchar		
engine	FuelSystem	varchar		
engine	Horsepower	int		
engine	NumOfCylinders	varchar		
engine	PeakRPM	int		
fuel	Aspiration	varchar		
fuel	FuelID	int	PRI	
fuel	FuelSystem	varchar		
fuel	FuelType	varchar		
performance	CityMPG	int		
performance	CompressionR...	decimal		
performance	HighwayMPG	int		
performance	PerformanceID	int	PRI	
transmission	DriveWheels	varchar		
transmission	NumOfDoors	varchar		
transmission	TransmissionID	int	PRI	

```

173
174 • INSERT INTO Brand (BrandName) VALUES ('BMW');
175
176 • INSERT INTO CarModel (BrandID, BodyStyle, DriveWheels, Price) VALUES
177   (3, 'sedan', 'rwd', 35000);
178
179 • INSERT INTO Engine (EngineType, NumOfCylinders, EngineSize, FuelSystem, Horsepower, PeakRPM) VALUES
180   ('mpfi', 'six', 300, 'mpfi', 250, 6500);
181
182 • INSERT INTO Fuel (FuelType, Aspiration, FuelSystem) VALUES
183   ('gas', 'std', 'mpfi');
184
185 • INSERT INTO Dimensions (WheelBase, Length, Width, Height, CurbWeight) VALUES
186   (110.0, 200.0, 75.0, 55.0, 3500);
187
188 • INSERT INTO Performance (CompressionRatio, CityMPG, HighwayMPG) VALUES
189   (10.0, 20, 30);
190
191 • INSERT INTO Transmission (NumOfDoors, DriveWheels) VALUES
192   ('four', 'rwd');
193
194 • INSERT INTO Car (ModelID, EngineID, FuelID, DimensionID, PerformanceID, TransmissionID) VALUES
195   (11, 11, 11, 11, 11, 11);
196

```



#	Time	Action	Message
✓ 26	00:12:24	INSERT INTO Dimensions (WheelBase, Length, Width, Height, CurbWeight) VALUES (110.0, 200.0, 75.0, 55...	1 row(s) affected
✓ 27	00:12:24	INSERT INTO Performance (CompressionRatio, CityMPG, HighwayMPG) VALUES (10.0, 20, 30)	1 row(s) affected
✓ 28	00:12:24	INSERT INTO Transmission (NumOfDoors, DriveWheels) VALUES (four, 'hwd')	1 row(s) affected
✓ 29	00:12:24	INSERT INTO Car (ModelID, EngineID, FuelID, DimensionID, PerformanceID, TransmissionID) VALUES (11, ...	1 row(s) affected


```


SELECT * FROM Car;
SELECT cm.BrandID, b.BrandName, cm.BodyStyle, cm.Price, e.EngineType, f.FuelType, p.CityMPG, p.HighwayMPG
FROM Car c
JOIN CarModel cm ON c.ModelID = cm.ModelID
JOIN Brand b ON cm.BrandID = b.BrandID
JOIN Engine e ON c.EngineID = e.EngineID
JOIN Fuel f ON c.FuelID = f.FuelID
JOIN Performance p ON c.PerformanceID = p.PerformanceID
WHERE cm.BrandID = 3;
UPDATE CarModel
SET Price = 37000
WHERE ModelID = 11;
UPDATE Fuel
SET FuelType = 'diesel'
WHERE FuelID = 11;
DELETE FROM Car WHERE ModelID = 11;
DELETE FROM Brand WHERE BrandID = 3;

```

Result Grid



Filter Rows:

Export:


Wrap Cell Content:


	BrandID	BrandName	BodyStyle	Price	EngineType	FuelType	CityMPG	HighwayMPG
--	---------	-----------	-----------	-------	------------	----------	---------	------------