PROJECT REPORT

Database Name: automobiledb
Prepared By: KARTIK TALWAR (055020), CHIRAG SINGLA (055060)

---

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**)

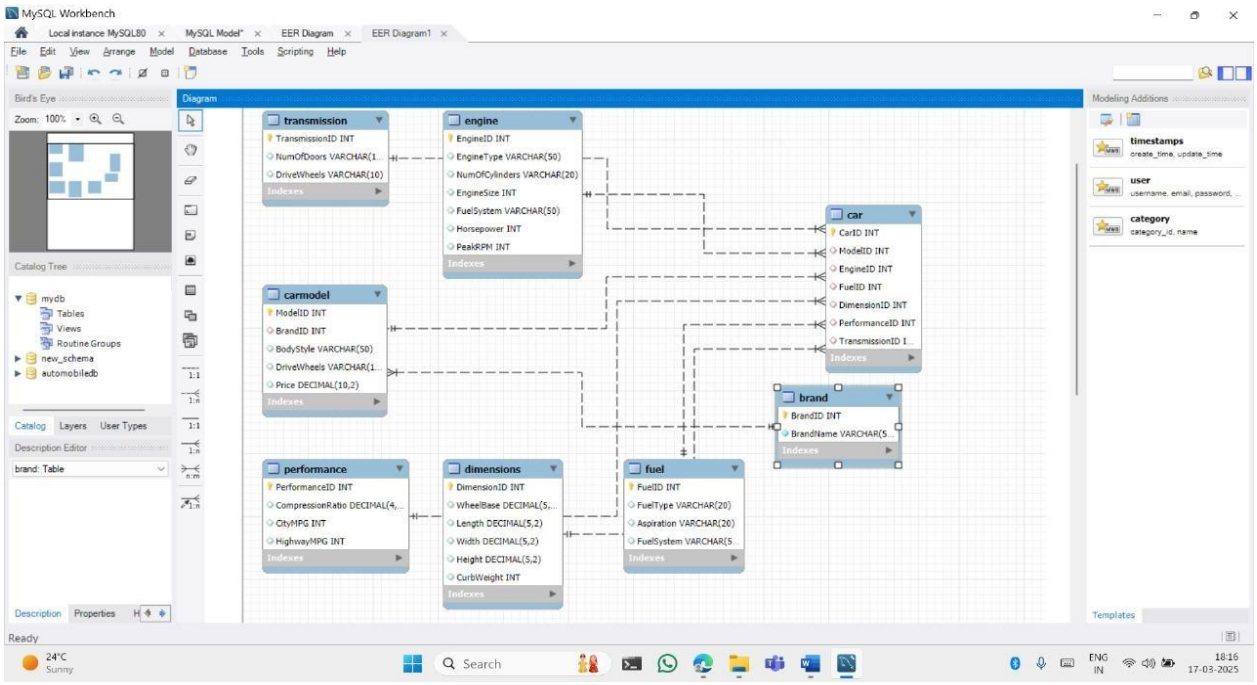
---

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) 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

---

3️⃣ 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.

4️⃣ Observations & Recommendations

✅ **Strengths:**

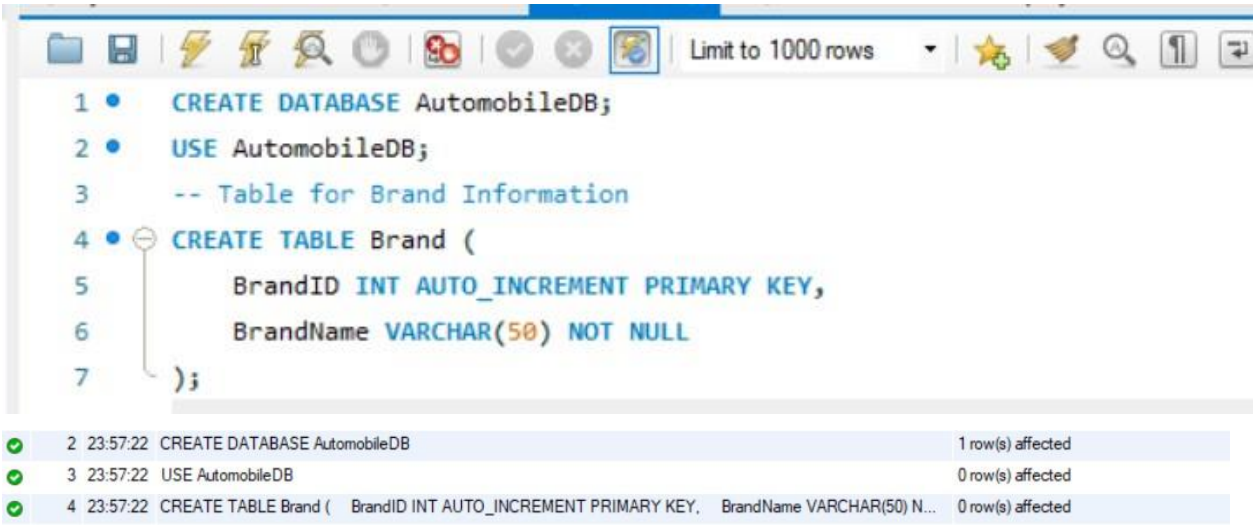
- Foreign key constraints correctly prevent invalid inserts and deletions.
- Cascading deletes ensure data consistency.
- The database follows **3NF**, reducing redundancy.

⚠ **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

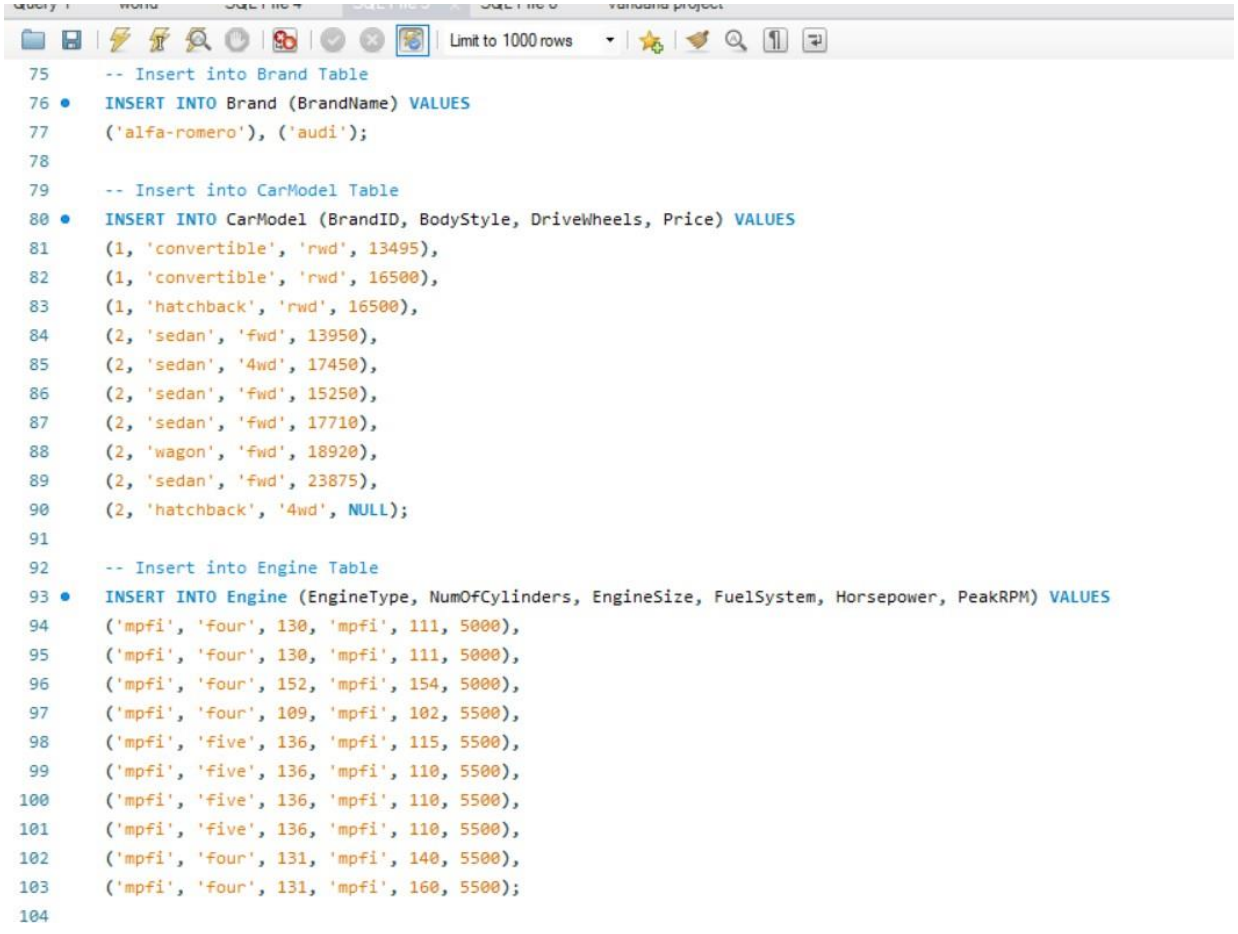


```sql
-- Table for Car Model Details
CREATE TABLE CarModel (
    ModelID INT AUTO_INCREMENT PRIMARY KEY,
    BrandID INT,
    BodyStyle VARCHAR(50),
    DriveWheels VARCHAR(10),
    Price DECIMAL(10,2),
    FOREIGN KEY (BrandID) REFERENCES Brand(BrandID) ON DELETE CASCADE
);
-- Table for Engine Specifications
CREATE TABLE Engine (
    EngineID INT AUTO_INCREMENT PRIMARY KEY,
    EngineType VARCHAR(50),
    NumOfCylinders VARCHAR(20),
    EngineSize INT,
    FuelSystem VARCHAR(50),
    Horsepower INT,
    PeakRPM INT
);
-- Table for Fuel Information
CREATE TABLE Fuel (
    FuelID INT AUTO_INCREMENT PRIMARY KEY,
    FuelType VARCHAR(20),
    Aspiration VARCHAR(20),
    FuelSystem VARCHAR(50)
);

-- Table for Dimensions of the Car
CREATE TABLE Dimensions (
    DimensionID INT AUTO_INCREMENT PRIMARY KEY,
    WheelBase DECIMAL(5,2),
    Length DECIMAL(5,2),
    Width DECIMAL(5,2),
    Height DECIMAL(5,2),
    CurbWeight INT
);
```

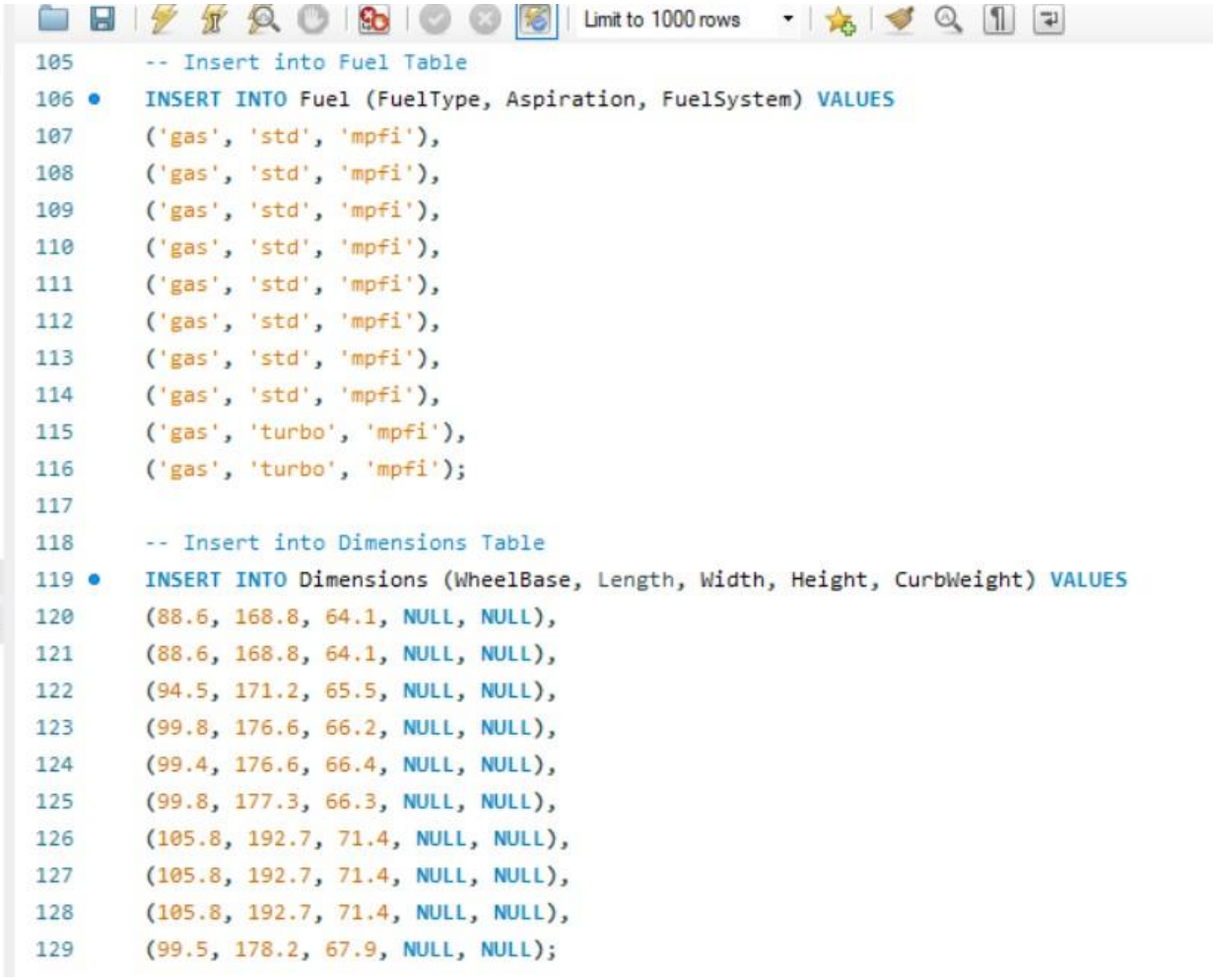
```sql
-- Table for Car Performance
CREATE TABLE Performance (
    PerformanceID INT AUTO_INCREMENT PRIMARY KEY,
    CompressionRatio DECIMAL(4,2),
    CityMPG INT,
    HighwayMPG INT
);

-- Table for Transmission Details
CREATE TABLE Transmission (
    TransmissionID INT AUTO_INCREMENT PRIMARY KEY,
    NumOfDoors VARCHAR(10),
    DriveWheels VARCHAR(10)
);
```

✔	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

```sql
-- Main Car Table (Links All Other Tables)
CREATE TABLE Car (
    CarID INT AUTO_INCREMENT PRIMARY KEY,
    ModelID INT,
    EngineID INT,
    FuelID INT,
    DimensionID INT,
    PerformanceID INT,
    TransmissionID INT,
    FOREIGN KEY (ModelID) REFERENCES CarModel(ModelID) ON DELETE CASCADE,
    FOREIGN KEY (EngineID) REFERENCES Engine(EngineID) ON DELETE CASCADE,
    FOREIGN KEY (FuelID) REFERENCES Fuel(FuelID) ON DELETE CASCADE,
    FOREIGN KEY (DimensionID) REFERENCES Dimensions(DimensionID) ON DELETE CASCADE,
    FOREIGN KEY (PerformanceID) REFERENCES Performance(PerformanceID) ON DELETE CASCADE,
    FOREIGN KEY (TransmissionID) REFERENCES Transmission(TransmissionID) ON DELETE CASCADE
);
```



```sql
-- Insert into Brand Table
INSERT INTO Brand (BrandName) VALUES
('alfa-romero'), ('audi');

-- Insert into CarModel Table
INSERT INTO CarModel (BrandID, BodyStyle, DriveWheels, Price) VALUES
(1, 'convertible', 'rwd', 13495),
(1, 'convertible', 'rwd', 16500),
(1, 'hatchback', 'rwd', 16500),
(2, 'sedan', 'fwd', 13950),
(2, 'sedan', '4wd', 17450),
(2, 'sedan', 'fwd', 15250),
(2, 'sedan', 'fwd', 17710),
(2, 'wagon', 'fwd', 18920),
(2, 'sedan', 'fwd', 23875),
(2, 'hatchback', '4wd', NULL);

-- Insert into Engine Table
INSERT INTO Engine (EngineType, NumOfCylinders, EngineSize, FuelSystem, Horsepower, PeakRPM) VALUES
('mpfi', 'four', 130, 'mpfi', 111, 5000),
('mpfi', 'four', 130, 'mpfi', 111, 5000),
('mpfi', 'four', 152, 'mpfi', 154, 5000),
('mpfi', 'four', 109, 'mpfi', 102, 5500),
('mpfi', 'five', 136, 'mpfi', 115, 5500),
('mpfi', 'five', 136, 'mpfi', 110, 5500),
('mpfi', 'five', 136, 'mpfi', 110, 5500),
('mpfi', 'five', 136, 'mpfi', 110, 5500),
('mpfi', 'four', 131, 'mpfi', 140, 5500),
('mpfi', 'four', 131, 'mpfi', 160, 5500);
```


Limit to 1000 rows
-- Insert into Fuel Table
INSERT INTO Fuel (FuelType, Aspiration, FuelSystem) VALUES
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'std', 'mpfi'),
('gas', 'turbo', 'mpfi'),
('gas', 'turbo', 'mpfi');
-- Insert into Dimensions Table
INSERT INTO Dimensions (WheelBase, Length, Width, Height, CurbWeight) VALUES
(88.6, 168.8, 64.1, NULL, NULL),
(88.6, 168.8, 64.1, NULL, NULL),
(94.5, 171.2, 65.5, NULL, NULL),
(99.8, 176.6, 66.2, NULL, NULL),
(99.4, 176.6, 66.4, NULL, NULL),
(99.8, 177.3, 66.3, NULL, NULL),
(105.8, 192.7, 71.4, NULL, NULL),
(105.8, 192.7, 71.4, NULL, NULL),
(105.8, 192.7, 71.4, NULL, NULL),
(99.5, 178.2, 67.9, NULL, NULL);

```sql
INSERT INTO Performance (CompressionRatio, CityMPG, HighwayMPG) VALUES
(9.0, 21, 27),
(9.0, 21, 27),
(9.0, 19, 26),
(10.0, 24, 30),
(8.0, 18, 22),
(8.5, 19, 25),
(8.5, 19, 25),
(8.5, 19, 25),
(8.3, 17, 20),
(7.0, 16, 22);

-- Insert into Transmission Table
INSERT INTO Transmission (NumOfDoors, DriveWheels) VALUES
('two', 'rwd'),
('two', 'rwd'),
('two', 'rwd'),
('four', 'fwd'),
('four', '4wd'),
('two', 'fwd'),
('four', 'fwd'),
('four', 'fwd'),
('four', 'fwd'),
('two', '4wd');

-- Insert into Car Table
INSERT INTO Car (ModelID, EngineID, FuelID, DimensionID, PerformanceID, TransmissionID) VALUES
(1, 1, 1, 1, 1, 1),
(2, 2, 2, 2, 2, 2),
(3, 3, 3, 3, 3, 3),
(4, 4, 4, 4, 4, 4),
(5, 5, 5, 5, 5, 5),
(6, 6, 6, 6, 6, 6),
(7, 7, 7, 7, 7, 7),
(8, 8, 8, 8, 8, 8),
(9, 9, 9, 9, 9, 9),
(10, 10, 10, 10, 10, 10);
```

✓	16	00:07:42	INSERT INTO Fuel (FuelType, Aspiration, FuelSystem) VALUES ('gas', 'std', 'mpfi'), ('gas', 'std', 'mpfi'), ('gas', 'std', '...	10 row(s) affected Records: 10	Duplicates: 0 Warni
✓	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 Warni
✓	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 Warni
✓	19	00:07:42	INSERT INTO Transmission (NumOfDoors, DriveWheels) VALUES ('two', 'rwd'), ('two', 'rwd'), ('two', 'rwd'), ('four', 'f...	10 row(s) affected Records: 10	Duplicates: 0 Warni
✓	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 Warni

```sql
SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE, COLUMN_KEY
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_SCHEMA = 'AutomobileDB';
```

Result Grid | Filter Rows: | 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

COLUMNS 1 ×

```
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);
```

#	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', 'rwd')	1 row(s) affected
29	00:12:24	INSERT INTO Car (ModelID, EngineID, FuelID, DimensionID, PerformanceID, TransmissionID) VALUES (11, ...	1 row(s) affected

```sql
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