



FATIMA JINNAH WOMEN UNIVERSITY
Department of Software Engineering

DATABASE PROJECT

Normalization and Functional Dependencies

Submitted To: Dr. Irum Matloob

Submitted By: Laiba Imran(2021-BSE-072)
Laiba Shahid(2021-BSE-051)
Hira Shakeel(2021-BSE-049)

Date: 12th -June-2023

Structured Query Language

Queries typically fall under the category of Data Definition Language (DDL) and Data Manipulation Language (DML) in the context of database management systems. Let's break down the two categories:

1. Data Definition Language (DDL):

- DDL statements are used to define or modify the structure of database objects, such as tables, indexes, or views.
- Examples of DDL statements include CREATE, ALTER, and DROP.
- DDL statements do not directly manipulate the data within the tables; they focus on defining the database schema.

2. Data Manipulation Language (DML):

- DML statements are used to manipulate the data stored in the database.
- Examples of DML statements include SELECT, INSERT, UPDATE, and DELETE.
- DML statements allow you to retrieve, insert, modify, and delete data from tables.

Created Tables using queries

In SQL, a CREATE query is used to create new database objects such as tables, views, indexes, or procedures.

Following queries are used to create the following tables:

CAR_comm,COMPANY_comm,Sale_comm,CONSUMER_comm,FEATURES and RATINGS_comm.

- Attributes in CAR_comm Table are VIN,condition,price,mileage,stock_#,Drivetrain.
- Attributes in COMPANY_comm Table are Make,State,Seller_Type.
- Attributes in Sale_comm are Stock_#,Year.
- Attributes in Consumer_comm Table are Consumer_Rating, Consumer_Review_#.
- Attributes in FEATURES Table are Model, Exterior_Color,Interior_Color,Engine,MPG,Fuel_Type,Transmission.
- Attributes in RATINGS_comm are Consumer_Rating, Confort_Rating,Interior_Design_Rating,Performance_Rating,Value_For_Money_Rating, Exterior_Styling_Rating,Reliability_Rating.

Table - 1

- **CAR TABLE**

Query:

```
CREATE TABLE CAR_comm  
(
```

```
VIN INT PRIMARY KEY,  
condition VARCHAR(50) NOT NULL,  
price INT NOT NULL,  
mileage INT,  
stock_# INT,  
Drivetrain VARCHAR(50),  
);
```

Table – 2

- COMPANY TABLE

Query:

```
CREATE TABLE COMPANY_comm  
(  
Make VARCHAR(50),  
State VARCHAR(50),  
Seller_Type VARCHAR(50) NOT NULL,  
  
);
```

Table – 3

- SALE TABLE

Query:

```
CREATE TABLE Sale_comm  
(  
Stock_# VARCHAR,  
Year INT,  
);
```

Table – 4

- CONSUMER TABLE

Query:

```
CREATE TABLE Consumer_comm  
(  
Consumer_Rating VARCHAR(50),  
Consumer_Review_# INT,  
);
```

Table – 5

- FEATURES TABLE

Query:

```
CREATE TABLE FEATURES  
(  
Model VARCHAR PRIMARY KEY,  
Exterior_Color VARCHAR(50) NOT NULL,  
Interior_Color VARCHAR(50) NOT NULL,  
Engine VARCHAR(100),  
MPG INT,
```

```
Fuel_Type VARCHAR(50),  
Transmission VARCHAR(50),  
);
```

Table – 6

- **RATINGS TABLE**

Query:

```
CREATE TABLE RATINGS_comm  
(  
Consumer_Rating VARCHAR(50) NOT NULL,  
Confort_Rating VARCHAR(50) NOT NULL,  
Interior_Design_Rating VARCHAR(50) NOT NULL,  
Performance_Rating VARCHAR(100),  
Value_For_Money_Rating VARCHAR(50),  
Exterior_Styling_Rating VARCHAR(50),  
Reliability_Rating VARCHAR(50),  
);
```

Inserted Data Using queries

In SQL, an INSERT query is used to insert data into a table.

By the below given queries the data has been inserted into above created tables.

Table – 1

- **CONSUMER TABLE INSERTION**

Query:

```
USE [Honda_sell_database]
```

```
GO
```

```
INSERT INTO [dbo].[Consumer_comm]  
    ([Consumer_Rating]  
    ,[Consumer_Review_#])  
SELECT  
    [Consumer_Rating]  
    ,[Consumer_Review_#]  
FROM honda_sell_data$
```

Table – 2

- **SALE TABLE INSERTION**

Query:

GO

USE [Honda_sell_database]

GO

INSERT INTO [dbo].[Sale]

 ([Stock_#]

 ,[Year])

SELECT

 [Stock_#]

 ,[Year]

 FROM honda_sell_data\$

GO

USE [Honda_sell_database]

GO

Table – 3

- **RATINGS TABLE INSERTION**

Query:

INSERT INTO [dbo].[RATINGS_comm]

 ([Consumer_Rating]

 ,[Comfort_Rating]

 ,[Interior_Design_Rating]

 ,[Performance_Rating]

 ,[Value_For_Money_Rating]

 ,[Exterior_Styling_Rating]

 ,[Reliability_Rating])

SELECT

 [Consumer_Rating]

 ,[Comfort_Rating]

```

,[Interior_Design_Rating]
,[Performance_Rating]
,[Value_For_Money_Rating]
,[Exterior_Styling_Rating]
,[Reliability_Rating]
FROM honda_sell_data$

```

GO

CREATE TABLE COMPANY_comm

```

(
Make VARCHAR(50),
State VARCHAR(50),
Seller_Type VARCHAR(50) NOT NULL,

);

```

Table – 4

- **COMPANY TABLE INSERTION**

Query:

```

USE [Honda_sell_database]

GO

INSERT INTO [dbo].[COMPANY_comm]
    ([Make]
    ,[State]
    ,[Seller_Type])
SELECT
    [Make]
    ,[State]
    ,[Seller_Type]
FROM honda_sell_data$

GO

```

DESIGN VIEW OF TABLES AND PRIMARY KEYS

Primary Key:

In a relational database, a primary key is a column or a set of columns that uniquely identifies each row in a table. Its purpose is to ensure data integrity and provide a way to uniquely identify and retrieve individual records in a table. Here are some key characteristics of a primary key:

- **Uniqueness:** Every value in the primary key column(s) must be unique. No two rows can have the same value(s) in the primary key column(s).
- **Non-nullability:** The primary key column(s) cannot contain null values. Each row must have a valid, non-null value in the primary key column(s).
- **Immutability:** The value(s) of the primary key column(s) should generally remain unchanged over the lifetime of a row. Modifying the primary key value(s) is not recommended, as it can lead to data inconsistencies and potential issues with referential integrity.
- **Indexing:** By default, most database systems automatically create an index on the primary key column(s). This helps improve the performance of searching, joining, and retrieving data from the table.
- **Relationship establishment:** A primary key can be used to establish relationships (such as foreign keys) with other tables. This enables the creation of logical connections between related data across multiple tables.

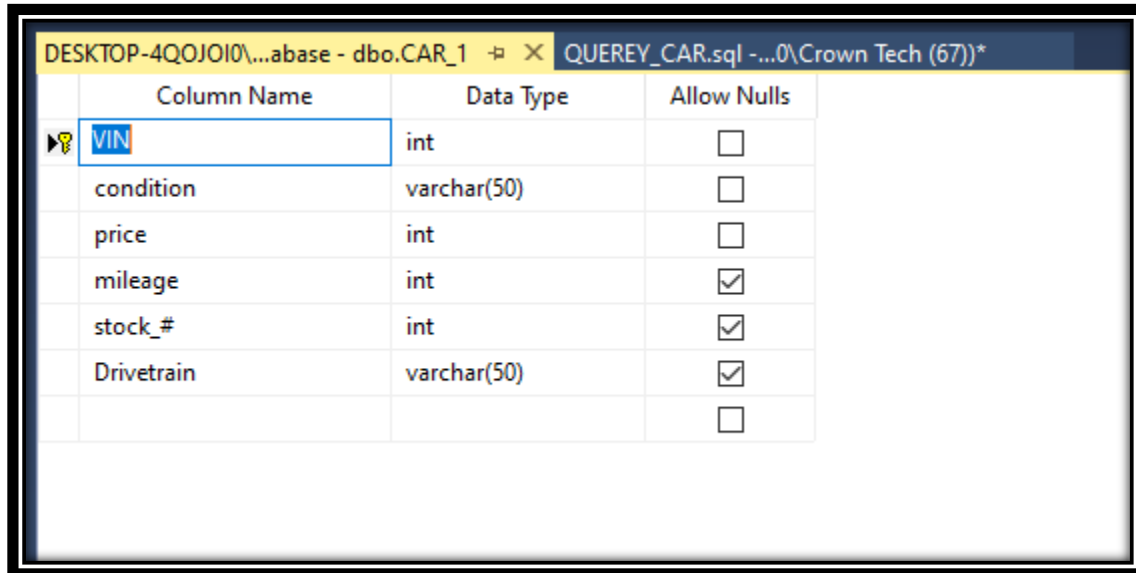
Primary Keys in Tables:

- In **CAR_comm** table VIN (Vehicle Identification Number) is the Primary key.
- In **COMPANY_comm** table ID is the primary key.
- In **CONSUMER_comm** table Consumer_ID is the primary key.
- In **RATINGS_comm** table RATING_ID is the primary key.
- In **Sale_comm** table sale_ID is the primary key.

Table – 1

- CAR TABLE

Design View:




Column Name	Data Type	Allow Nulls
VIN	int	<input type="checkbox"/>
condition	varchar(50)	<input type="checkbox"/>
price	int	<input type="checkbox"/>
mileage	int	<input checked="" type="checkbox"/>
stock_#	int	<input checked="" type="checkbox"/>
Drivetrain	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Table – 2

- COMPANY TABLE

Design View:



Column Name	Data Type	Allow Nulls
Make	varchar(50)	<input checked="" type="checkbox"/>
State	varchar(50)	<input checked="" type="checkbox"/>
Seller_Type	varchar(50)	<input checked="" type="checkbox"/>
ID	int	<input type="checkbox"/>
		<input type="checkbox"/>

Table – 3

- CONSUMER TABLE

Design View:

DESKTOP-4QOJOI0...o.Consumer_comm			DESKTOP-4QOJOI0...o.COMPANY_comm	DESKTOP-4QOJOI0...
	Column Name	Data Type	Allow Nulls	
▶	Consumer_Rating	varchar(50)	<input checked="" type="checkbox"/>	
	Consumer_Review_#	int	<input checked="" type="checkbox"/>	
🔑	Consumer_ID	int	<input type="checkbox"/>	
			<input type="checkbox"/>	

Table – 4

- RATINGS TABLE

Design View:

DESKTOP-4QOJOI0...bo.RATINGS_comm			DESKTOP-4QOJOI0...se - dbo.FEATURES	DESKTOP-4QOJOI0...
	Column Name	Data Type	Allow Nulls	
▶	Consumer_Rating	varchar(50)	<input type="checkbox"/>	
	Comfort_Rating	int	<input checked="" type="checkbox"/>	
	Interior_Design_Rating	int	<input checked="" type="checkbox"/>	
	Performance_Rating	varchar(100)	<input checked="" type="checkbox"/>	
	Value_For_Money_Rating	varchar(50)	<input checked="" type="checkbox"/>	
	Exterior_Styling_Rating	varchar(50)	<input checked="" type="checkbox"/>	
	Reliability_Rating	varchar(50)	<input checked="" type="checkbox"/>	
🔑	RATING_ID	int	<input type="checkbox"/>	
			<input type="checkbox"/>	

Table – 5

- SALE TABLE

Design View:

DESKTOP-4QOJOI0\S...atabase - dbo.Sale - X DESKTOP-4QOJOI0...bo.RATINGS_comm DESI			
	Column Name	Data Type	Allow Nulls
▶	Stock_#	varchar(20)	<input checked="" type="checkbox"/>
	Year	int	<input checked="" type="checkbox"/>
🔑	sale_ID	int	<input type="checkbox"/>
			<input type="checkbox"/>

REMOVING REDUNDANCY FROM TABLES

By running below query redundancy has been removed from CAR_comm table.

Similarly, the same procedure is applied to all other tables accordingly to remove redundancy.

Table – 1

- **CAR TABLE**

Duplication of data:

No duplication.

Table – 2

- **COMPANY TABLE**

Duplication of data:

Removing redundancy.

Query:

Select Distinct Make,State,Seller_Type

From COMPANY_comm;

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'DESKTOP-4QOJ010\SQLEXPRESS01 (SQL Se...'. The 'Tables' folder is expanded, and 'dbo.COMPANY_comm' is selected. The 'Columns' folder is also expanded, showing 'PK_COMPANY_comm'. The 'Messages' pane at the bottom right displays the results of a query. The query is: `SELECT DISTINCT Make, State, Seller_Type FROM COMPANY_comm;`. The results are as follows:

	Make	State	Seller_Type
48	Honda	NV	Dealer
49	Honda	OK	Dealer
50	Honda	IN	Dealer
51	Honda	AL	Dealer
52	Honda	Glens	Dealer
53	Honda	PA	Dealer
54	Honda	IA	Dealer
55	Honda	MN	Dealer
56	Honda	SC	Dealer
57	Honda	AR	Dealer
58	Honda	NH	Dealer

Table – 3

- **CONSUMER TABLE**

Duplication of data:

Removing redundancy.

Query:

```
DELETE FROM CONSUMER_COMM
WHERE CONSUMER_ID NOT IN(
SELECT MIN(CONSUMER_ID)
FROM CONSUMER_COMM
GROUP BY CONSUMER_RATING,CONSUMER_REVIEW_#
);
SELECT * FROM CONSUMER_COMM;
```

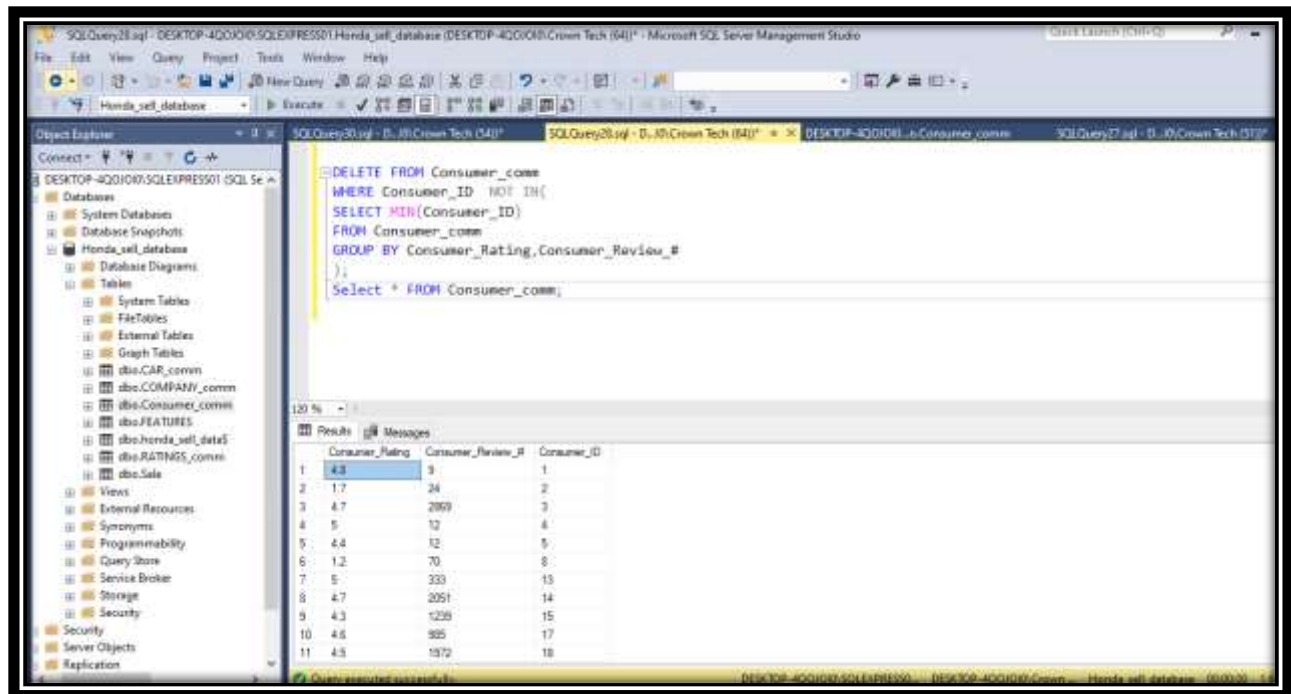


Table – 4

- RATINGS TABLE

Duplication of data:

Removing redundancy.

Query:

```
DELETE FROM RATINGS_comm
WHERE RATING_ID NOT IN(
SELECT MIN(RATING_ID)
FROM RATINGS_comm
GROUP BY
Consumer_Rating,Comfort_Rating,Interior_Design_Rating,Performance_Rating,Value_For_Money_
Rating,Exterior_Styling_Rating,Reliability_Rating
);
Select * FROM RATINGS_comm;
```

SQL Query34.sql - D:\Crown Tech (860)* - Microsoft SQL Server Enterprise Manager

Object Explorer: Connect -> Database Diagrams -> Tables -> System Tables -> External Tables -> Graph Tables -> dbo.CAR_comm -> dbo.COMPANY_comm -> dbo.Consumer_comm -> dbo.FEATURES -> Columns -> Keys -> Constraints -> Triggers -> Indexes -> Statistics -> dbo.honda_sell_data5 -> dbo.RATINGS_comm -> dbo.Sale -> Views -> External Resources -> Synonyms -> Programmability -> Query Store -> Service Broker -> Storage -> Security -> Security -> Server Objects

SQL Query34.sql - D:\Crown Tech (860)*

```

DELETE FROM RATINGS_comm
WHERE RATING_ID NOT IN(
SELECT MIN(RATING_ID)
FROM RATINGS_comm
GROUP BY Consumer_Rating,Comfort_Rating,Interior_Design_Rating,Performance_Rating,Value_For_Money_Rating,Exterior_Style
);
Select * FROM RATINGS_comm;

```

Results: 100% - 4

	Consumer_Rating	Comfort_Rating	Interior_Design_Rating	Performance_Rating	Value_For_Money_Rating	Exterior_Styling_Rating	Reliability_Rating	RATING_ID
1	4.8	5	4	4.8	4.2	5	5	49
2	3.7	5	3	4	4	5	5	50
3	4.7	5	3	4	4	5	5	51
4	5	5	5	5	5	5	5	52
5	4.4	5	3	4	4	5	5	53
6	4.4	5	3	4	4	4	5	54
7	3.2	5	3	4	4	5	5	55
8	3.2	NULL	NULL	NULL	NULL	NULL	NULL	56
9	4.7	5	5	5	4	4	5	57
10	4.3	NULL	NULL	NULL	NULL	NULL	NULL	58
11	4.6	NULL	NULL	NULL	NULL	NULL	NULL	59

Table – 5

- SALE TABLE

Duplication of data:

Removing redundancy.

Query:

```

DELETE FROM Sale
WHERE sale_ID NOT IN(
SELECT MIN(sale_ID)
FROM Sale
GROUP BY Stock_#,Year
);
Select * FROM Sale;

```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure, including tables like `dbo.CAR_comm`, `dbo.COMPANY_comm`, `dbo.Consumer_comm`, `dbo.FEATURES`, `dbo.honda_sell_data$`, `dbo.RATINGS_comm`, and `dbo.Sale`. The main query window shows the following SQL code:

```
DELETE FROM Sale
WHERE sale_ID NOT IN(
SELECT MIN(sale_ID)
FROM Sale
GROUP BY Stock_#,Year
);
Select * FROM Sale;
```

Below the query window, the Results grid displays the following data:

	Stock_#	Year	sale_ID
1	830164	2023	3
2	6.40295e+006	2023	4
3	NULL	2023	5
4	NULL	2022	6
5	5.70595e+006	2023	7
6	5.72427e+006	2023	8
7	2.90379e+006	2023	12
8	30444	2023	14
9	78543	2022	15
10	44695	2023	17
11	44677	2023	18

Constraints in Tables

Foreign Keys:

Foreign keys serve to establish relationships between tables in a relational database. They ensure referential integrity by enforcing constraints that maintain consistency and prevent inconsistencies in the data.

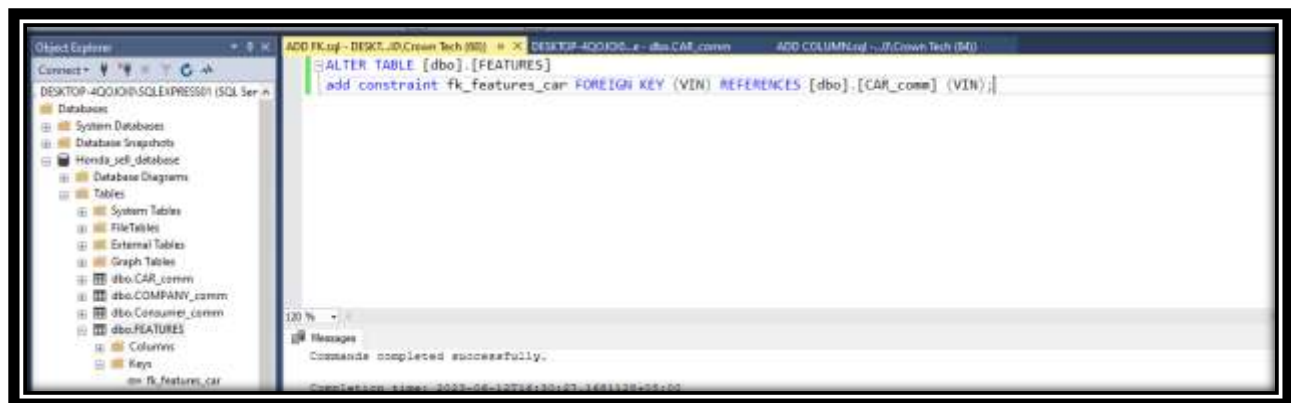
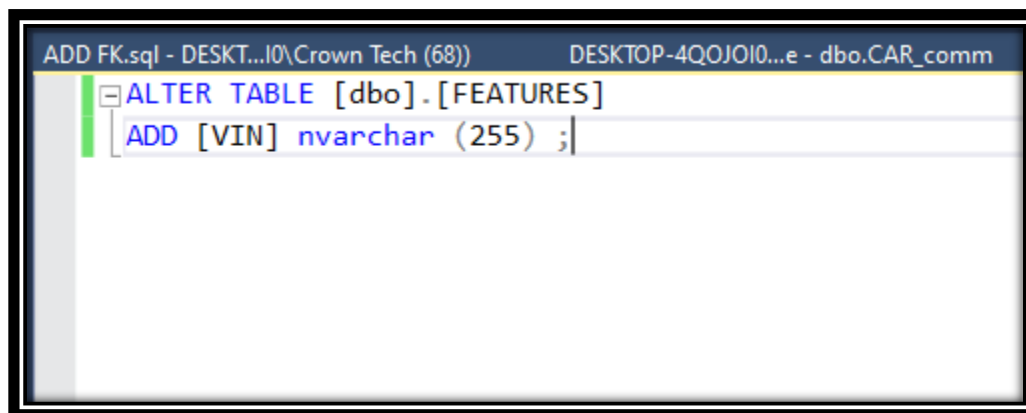
1. Foreign Key of COMPANY AND CAR (COMPANY HAS A CAR)

The screenshot shows the SQL Server Enterprise Manager interface with a query window containing the following SQL code:

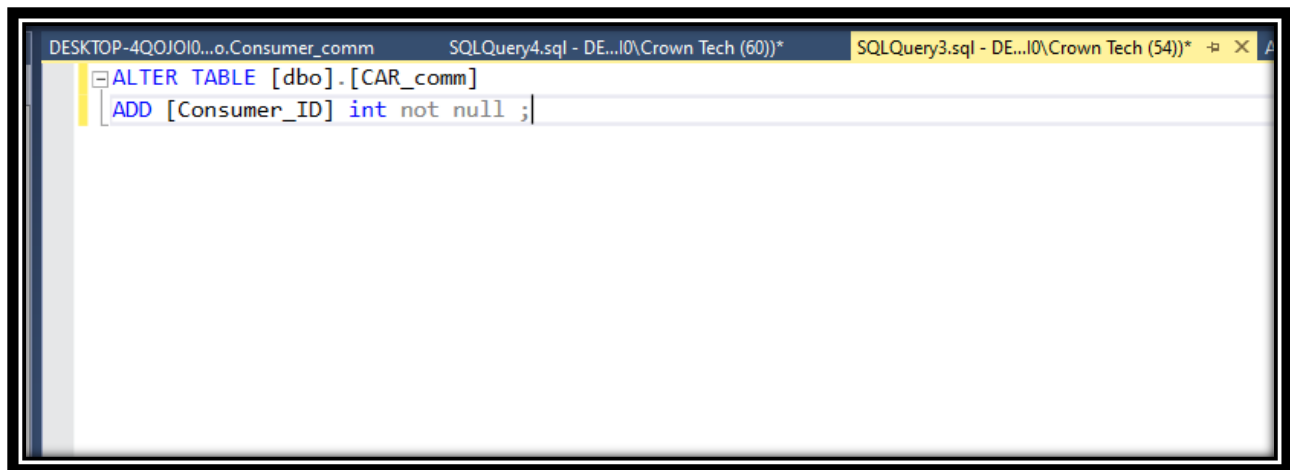
```
alter table [dbo].[CAR_comm]
add constraint FK_Comp_car FOREIGN key(ID) REFERENCES [dbo].[COMPANY_comm](ID);
```



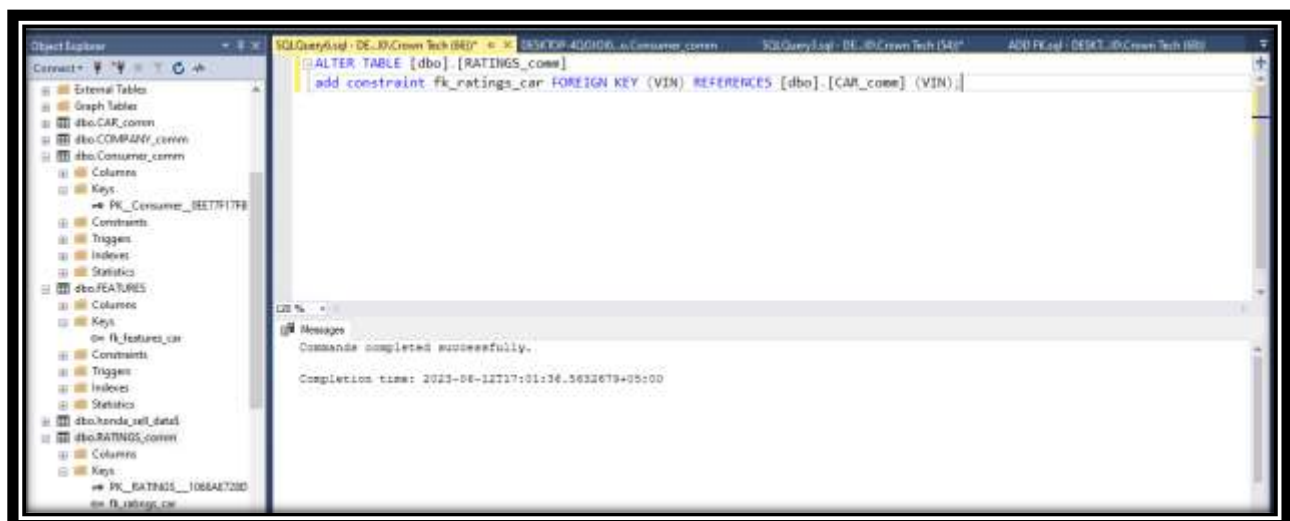
2. Foreign key of FEATURES (CAR HAS FEATURES)



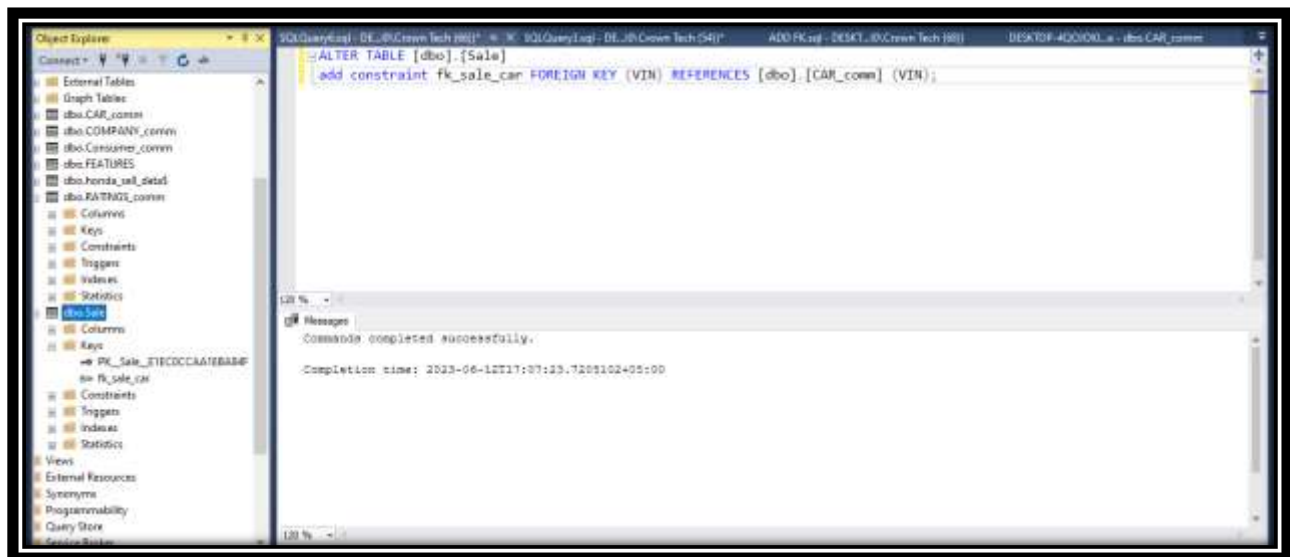
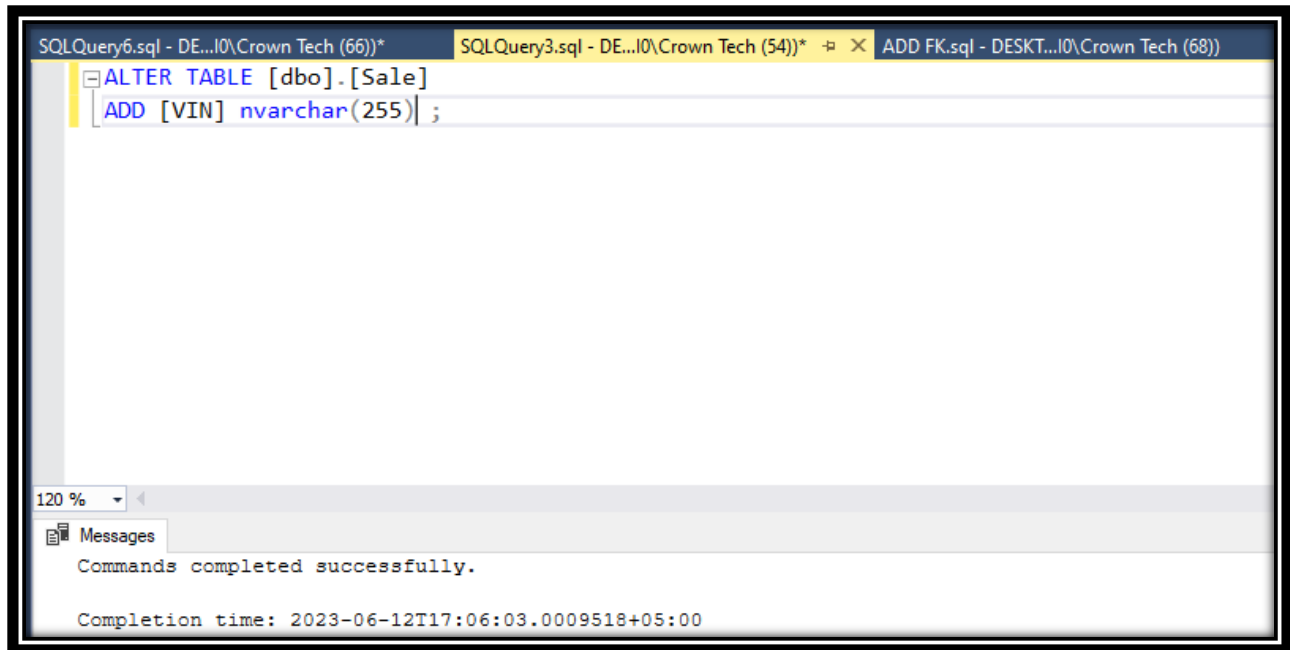
3. Foreign key of CONSUMER AND CAR (CONSUMER BUYS A CAR)



4. Foreign key of car and rating (car has rating)



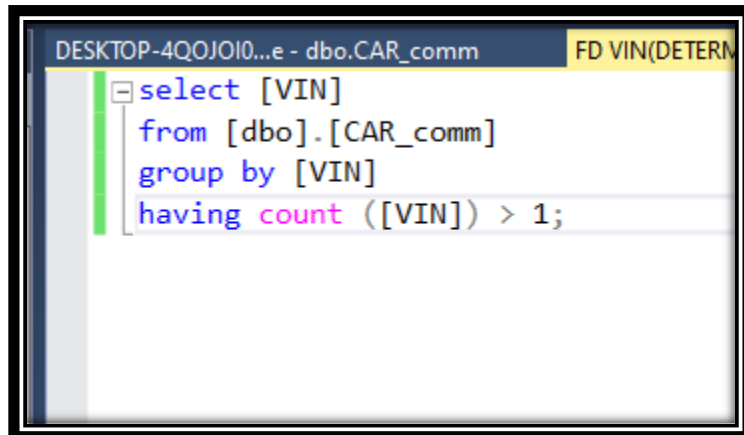
5. Foreign key of CAR AND SALE (CAR IS SALED OUT)



FUNCTIONAL DEPENDENCIES

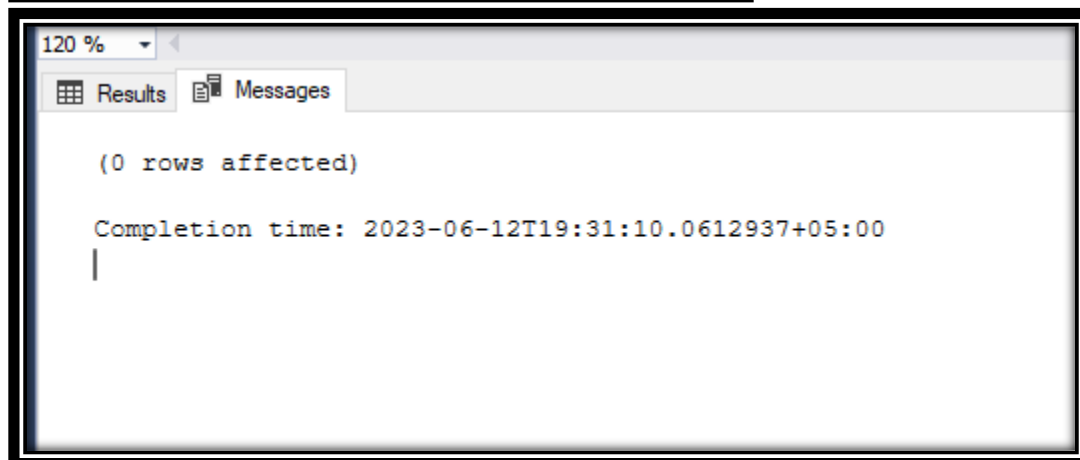
Functional dependencies describe the relationships between attributes (columns) within a database table. They establish a dependency relationship where the value of one or more attributes determines the value of another attribute.

CAR_comm



DESKTOP-4QOJOI0...e - dbo.CAR_comm FD VIN(DETERM

```
select [VIN]
from [dbo].[CAR_comm]
group by [VIN]
having count ([VIN]) > 1;
```



120 %

Results Messages

(0 rows affected)

Completion time: 2023-06-12T19:31:10.0612937+05:00

```
DESKTOP-4QOJOI0...e - dbo.CAR_comm    FD VIN(DETERMININ... \Crown Te
select [condition]
from [dbo].[CAR_comm]
group by [condition]
having count ([condition]) > 1;
```

```
120 %
Results Messages
(0 rows affected)
Completion time: 2023-06-12T19:36:20.2681096+05:00
```

```
DESKTOP-4QOJOI0...e - dbo.CAR_comm    FD VIN(DETERMININ...\Crown Tech (52))*  
select [mileage]  
from [dbo].[CAR_comm]  
group by [mileage]  
having count ([mileage]) > 1
```

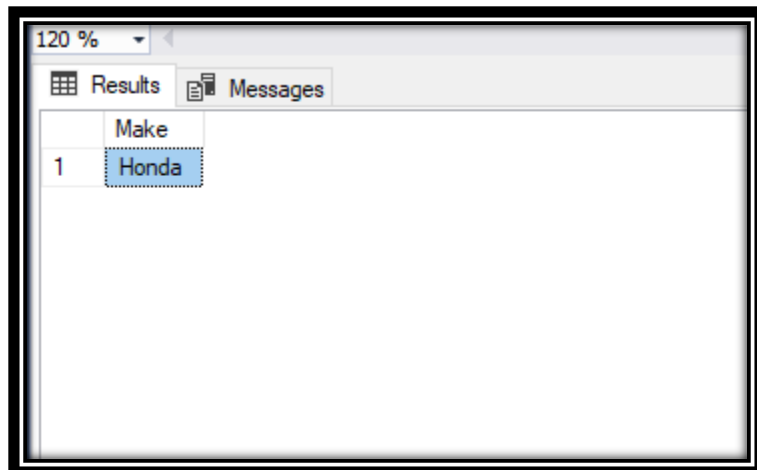
```
120 %  
Results Messages  
  
(0 rows affected)  
  
Completion time: 2023-06-12T19:48:33.3214668+05:00  
|
```

```
DESKTOP-4QOJOI0...e - dbo.CAR_comm  FD VIN(DETERMININ...\Crown Tech (
select [stock_#]
from [dbo].[CAR_comm]
group by [stock_#]
having count ([stock_#]) > 1
```

```
120 %
Results Messages
(0 rows affected)

Completion time: 2023-06-12T19:48:33.3214668+05:00
|
```

```
select [Make]
from [dbo].[COMPANY_comm]
group by [Make]
having count ([Make]) > 1
```

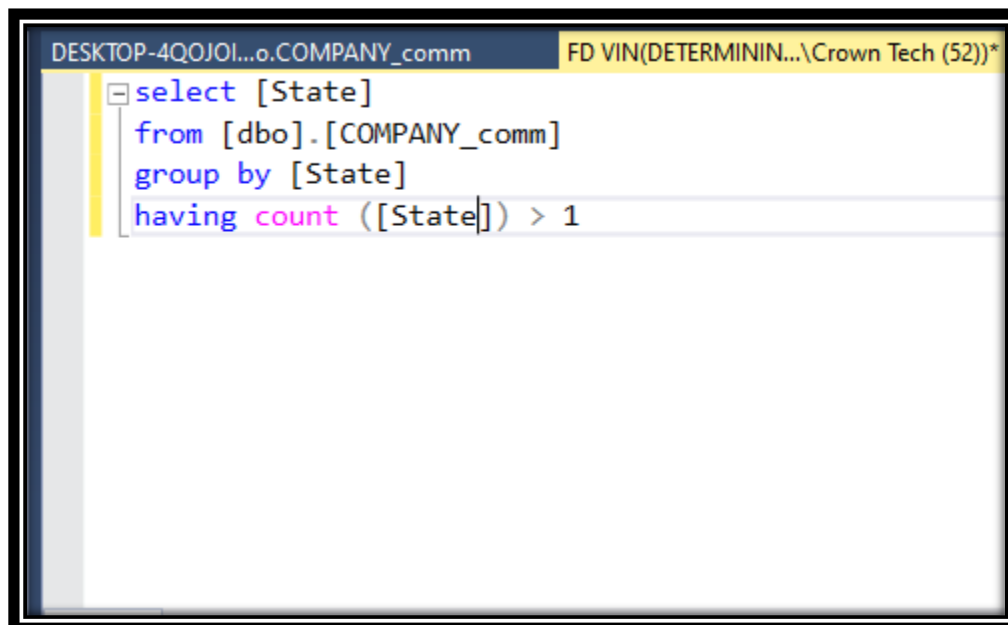


120 %

Results Messages

	Make
1	Honda

COMPANY_comm



DESKTOP-4QOJOI...o.COMPANY_comm FD VIN(DETERMININ... \Crown Tech (52))*

```
select [State]
from [dbo].[COMPANY_comm]
group by [State]
having count ([State]) > 1
```

120 %

Results Messages

	State
1	UT
2	WI
3	NC
4	MA
5	MI
6	TN
7	NH
8	AK
9	OK
10	KY
11	CO

DESKTOP-4QOJOI...o.COMPANY_comm FD VIN(DETERMININ...\Crown T

```

select [Seller_Type]
from [dbo].[COMPANY_comm]
group by [Seller_Type]
having count ([Seller_Type]) > 1
  
```

120 %

Results Messages

	Seller_Type
1	Dealer
2	Individual

Consumer_comm

```
DESKTOP-4QOJO10...o.Consumer_comm  FD VIN(DETERMININ...\Crown Tech (52))*  
select [Consumer_Rating]  
from [dbo].[Consumer_comm]  
group by [Consumer_Rating]  
having count ([Consumer_Rating]) > 1
```

120 %

Results Messages

	Consumer_Rating
1	3
2	3.1
3	4.2
4	3.9
5	1.6
6	2.7
7	3.8
8	4.5
9	3.3
10	2.2
11	4.4


```
DESKTOP-4QOJOI0...o.Consumer_comm  FD VIN(DETERMININ...\Crown Tech (52))*
select [Consumer_Review_#]
from [dbo].[Consumer_comm]
group by [Consumer_Review_#]
having count ([Consumer_Review_#]) > 1
```

120 %

Results Messages

	Consumer_Review_#
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10

```
DESKTOP-4QOJOI0...o.Consumer_comm  FD VIN(DETERMININ...\Crown Tech
select [Consumer_ID]
from [dbo].[Consumer_comm]
group by [Consumer_ID]
having count ([Consumer_ID]) > 1
```

120 %

Results Messages

Consumer_ID

RATINGS_comm

```
DESKTOP-4QOJOI0...bo.RATINGS_comm  FD VIN(DETERMININ...\Crown Tech (52))*
select [Consumer_Rating]
from [dbo].[RATINGS_comm]
group by [Consumer_Rating]
having count ([Consumer_Rating]) > 1
```

Results Messages	
	Consumer_Rating
1	3
2	3.1
3	4.2
4	3.9
5	1.6
6	1.2
7	2.7
8	3.8
9	4.5
10	3.3
11	2.2

```

DESKTOP-4QOJOI0...bo.RATINGS_comm  FD VIN(DETERMININ...\Crown Tech
select [Comfort_Rating]
from [dbo].[RATINGS_comm]
group by [Comfort_Rating]
having count ([Comfort_Rating]) > 1

```

Results Messages	
	Comfort_Rating
1	3
2	4
3	5

```
DESKTOP-4QOJOI0...bo.RATINGS_comm  FD VIN(DETERMININ...\Crown Tech (52))*  X
select [Interior_Design_Rating]
from [dbo].[RATINGS_comm]
group by [Interior_Design_Rating]
having count ([Interior_Design_Rating]) > 1
```

120 %

Results Messages

	Interior_Design_Rating
1	3
2	4
3	5

```
DESKTOP-4QOJOI0...bo.RATINGS_comm  FD VIN(DETERMININ...\Crown Tech (52))* -p X
select [Performance_Rating]
from [dbo].[RATINGS_comm]
group by [Performance_Rating]
having count ([Performance_Rating]) > 1
```

Results		Messages
	Performance_Rating	
1	4.2	
2	3.9	
3	3.8	
4	4.5	
5	4.4	
6	4.1	
7	4.8	
8	4.7	
9	4.3	
10	3.6	
11	4.9	

```
DESKTOP-4QOJOI0...bo.RATINGS_comm  FD VIN(DETERMININ...\Crown Tech (52))*
select [Value_For_Money_Rating]
from [dbo].[RATINGS_comm]
group by [Value_For_Money_Rating]
having count ([Value_For_Money_Rating]) > 1
```

120 %

Results Messages

	Value_For_Money_Rating
1	4.2
2	3.9
3	3.8
4	4.5
5	4.4
6	4.1
7	4.8
8	4.7
9	4.3
10	3.6
11	4.9

```
DESKTOP-4QOJOI0...bo.RATINGS_comm  FD VIN(DETERMININ...\Crown Tech (52))
select [RATING_ID]
from [dbo].[RATINGS_comm]
group by [RATING_ID]
having count ([RATING_ID]) > 1
```

120 %

Results Messages

RATING_ID

Sale_comm

```
DESKTOP-4QOJOI0\S...atabase - dbo.Sale  DESKTOP-4QOJOI0...I
select [Stock_#]
from [dbo].[Sale]
group by [Stock_#]
having count ([Stock_#]) > 1
```

120 %

Results Messages

	Stock_#
1	1.38007e+009
2	1298
3	12990
4	16044
5	2.00195e+009
6	2.00201e+009
7	2.00202e+009
8	2.00208e+009
9	2.00209e+009
10	2.00211e+009
11	2.00213e+009

DESKTOP-4QOJOI0\S...atabase - dbo.Sale DESKTOP-4QOJOI0...

```
select [Year]
from [dbo].[Sale]
group by [Year]
having count ([Year]) > 1
```


120 %

Results		Messages
	Year	
1	2010	
2	2007	
3	2001	
4	2021	
5	2013	
6	2008	
7	2002	
8	2022	
9	2016	
10	2005	
11	2019	

DESKTOP-4QOJOI0\S...atabase - dbo.Sale DESKTOP-4QOJOI0...bo.RATINGS_comm

```

select [sale_ID]
from [dbo].[Sale]
group by [sale_ID]
having count ([sale_ID]) > 1
  
```

120 %

Results

Messages

sale_ID

NORMALIZATION

1. First Normal Form: (1NF)

- **Atomic Values:**

Each column in a table should contain only atomic (indivisible) values. This means that each value in a column should represent a single piece of data. If a column contains multiple values or a list of values, it should be split into separate columns.

- **Unique Column Names:**

Each column in a table should have a unique name, and no two columns should have the same name. This ensures that each column is uniquely identifiable.

- **Rows and Columns:**

Each row in a table should represent a unique record or entity, and each column should contain a single attribute of that entity. There should be no duplicate rows, and each column should have a distinct purpose.

- **Primary Key:**


Each table should have a primary key that uniquely identifies each row in the table. The primary key should consist of one or more columns that have unique values for each row. This helps in distinguishing one row from another.

- **No Repeating Groups:**

A table should not contain repeating groups or arrays of values. Each column should contain only single values, and if there are multiple values related to an entity, they should be placed in separate rows.

- **Key Points:**

In short there should not be multi valued attribute for any attribute and Unique record should exist.


	Column Name	Data Type	Allow Nulls
	VIN	nvarchar(255)	<input type="checkbox"/>
	condition	varchar(50)	<input type="checkbox"/>
	price	int	<input type="checkbox"/>
	mileage	int	<input checked="" type="checkbox"/>
	stock_#	int	<input checked="" type="checkbox"/>
	Drivetrain	varchar(50)	<input checked="" type="checkbox"/>
	ID	int	<input type="checkbox"/>
	Consumer_ID	int	<input type="checkbox"/>
			<input type="checkbox"/>


DESKTOP-4QOJOI0...o.COMPANY_comm X DESKTOP-4QOJOI0...e - dbo.CAR_

	Column Name	Data Type	Allow Nulls
►	Make	varchar(50)	<input checked="" type="checkbox"/>
	State	varchar(50)	<input checked="" type="checkbox"/>
	Seller_Type	varchar(50)	<input checked="" type="checkbox"/>
🔑	ID	int	<input type="checkbox"/>
			<input type="checkbox"/>

DESKTOP-4QOJOI0...o.Consumer_comm X DESKTOP-4QOJOI0...o.COMPANY_com

	Column Name	Data Type	Allow Nulls
►	Consumer_Rating	varchar(50)	<input checked="" type="checkbox"/>
	Consumer_Review_#	int	<input checked="" type="checkbox"/>
🔑	Consumer_ID	int	<input type="checkbox"/>
			<input type="checkbox"/>

DESKTOP-4QOJOI0...o.Consumer_comm		DESKTOP-4QOJOI0...o.COMPA	
Column Name	Data Type	Allow Nulls	
Consumer_Rating	varchar(50)	<input type="checkbox"/>	
Comfort_Rating	int	<input checked="" type="checkbox"/>	
Interior_Design_Rating	int	<input checked="" type="checkbox"/>	
Performance_Rating	varchar(100)	<input checked="" type="checkbox"/>	
Value_For_Money_Rating	varchar(50)	<input checked="" type="checkbox"/>	
Exterior_Styling_Rating	varchar(50)	<input checked="" type="checkbox"/>	
Reliability_Rating	varchar(50)	<input checked="" type="checkbox"/>	
 RATING_ID	int	<input type="checkbox"/>	
VIN	nvarchar(255)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

DESKTOP-4QOJOI0...o.Consumer_comm		DESKTOP-4QOJOI0...o.COMPANY_con	
Column Name	Data Type	Allow Nulls	
Stock_#	varchar(20)	<input checked="" type="checkbox"/>	
Year	int	<input checked="" type="checkbox"/>	
 sale_ID	int	<input type="checkbox"/>	
VIN	nvarchar(255)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Results:

So, in our database there is no multivalued attribute so, it is in *first normal form*.

2. Second Normal form:(2NF)

- **Meet the criteria for 1NF:**

The table must already satisfy the criteria for 1NF, including having atomic values, unique column names, rows and columns, and a primary key.

- **Single-Attribute Primary Key:**






The table should have a single-attribute primary key, meaning it consists of only one column. If the primary key is composite (consisting of multiple columns), it should be broken down into separate tables, each with a single-column primary key.

- **Partial Dependencies:**

Ensure that each non-key column in the table depends on the entire primary key, not just a part of it. If any non-key column depends on only a subset of the primary key, it should be moved to a separate table along with the subset of the primary key it depends on.

Results:

So, as the tables given above there is *no partial dependencies* in this database because no composite primary key exists.

	Consumer_ID	int	<input type="checkbox"/>
	ID	int	<input type="checkbox"/>
	RATING_ID	int	<input type="checkbox"/>
	sale_ID	int	<input type="checkbox"/>
	VIN	nvarchar(255)	<input type="checkbox"/>

3. Third normal form: (3NF)

- **Meet the criteria for 2NF:**

Table should be in 2nf.

- **Transitive Dependency:**

No transitive dependency exists.

- **Non-Prime Attributes:**

Non-prime attributes should not depend on other non-prime

Results:

All the tables in this Honda data sell *only prime attributes can determine all other attributes*.

4. (Boyce-Codd Normal Form): BCNF

BCNF stands for Boyce-Codd Normal Form, which is a higher level of database normalization.

- **Check on Prime attributes:**

it is strictly checked that only prime attributes or the candidate can determine all the other non-prime attributes.

As this database is in 3nf and also in BCNF.

L.H.S = ck, sk