Sql Project code for Car lifecycle management database with focus on manufacturing side

```
-- Create Customer table
CREATE TABLE Customer (
   C_ID VARCHAR(20) PRIMARY KEY NOT NULL,
   Name VARCHAR(50),
   Is_Company VARCHAR(50),
   Address VARCHAR(50),
   Phone BIGINT,
   Gender VARCHAR(10),
   Annual_income BIGINT,
   D_ID VARCHAR(20)
);

-- Insert values into Customer table
INSERT INTO Customer (C_ID, Name, Is_Company, Address, Phone, Gender,
Annual_income, D_ID)
VALUES
('C001', 'John Doe', 'Yes', '123 Main St', 1234567890, 'Male', 50000, 'D001'),
('C002', 'Jane Smith', 'No', '456 Oak Ave', 9876543210, 'Female', 75000, 'D002'),
('C003', 'Michael Johnson', 'Yes', '789 Elm Rd', 5551234567, 'Male', 60000, 'D003'),
('C004', 'Emily Brown', 'No', '321 Pine Blvd', 9998887777, 'Female', 80000, 'D001'),
('C005', 'David Wilson', 'Yes', '654 Cedar Ln', 1112223333, 'Male', 70000, 'D002');

-- Create Dealer table
CREATE TABLE Dealer (
   D_ID VARCHAR(20) PRIMARY KEY NOT NULL,
   Dname VARCHAR(20),
```

```
    Dlocation VARCHAR(50)
);

-- Insert values into Dealer table
INSERT INTO Dealer (D_ID, Dname, Dlocation)
VALUES
('D001', 'Dealer 1', 'D_City 1'),
('D002', 'Dealer 2', 'D_City 2'),
('D003', 'Dealer 3', 'D_City 3'),
('D004', 'Dealer 4', 'D_City 4'),
('D005', 'Dealer 5', 'D_City 5');

-- Create Sales table
CREATE TABLE Sales (
    M_ID VARCHAR(20) PRIMARY KEY NOT NULL,
    C_ID VARCHAR(20),
    D_ID VARCHAR(20),
    Date DATE,
    Price NUMERIC(10, 2),
    FOREIGN KEY (C_ID) REFERENCES Customer(C_ID),
    FOREIGN KEY (D_ID) REFERENCES Dealer(D_ID)
);

-- Insert values into Sales table
INSERT INTO Sales (M_ID, C_ID, D_ID, Date, Price)
VALUES
('M001', 'C001', 'D001', '2024-03-01', 2500000),
```

```
('M002', 'C002', 'D002', '2024-03-05', 3000000),
('M003', 'C003', 'D003', '2024-03-10', 3500000),
('M004', 'C004', 'D001', '2024-03-15', 4000000),
('M005', 'C005', 'D002', '2024-03-20', 4500000);

-- Create Inventory table
CREATE TABLE Inventory (
  In_ID VARCHAR(20) PRIMARY KEY NOT NULL,
  In_name VARCHAR(50),
  In_location VARCHAR(50),
  Dealer_ID VARCHAR(20),
  FOREIGN KEY (Dealer_ID) REFERENCES Dealer(D_ID)
);

-- Insert values into Inventory table
INSERT INTO Inventory (In_ID, In_name, In_location, Dealer_ID)
VALUES
('IN001', 'Inventory 1', 'I_Location 1', 'D001'),
('IN002', 'Inventory 2', 'I_Location 2', 'D002'),
('IN003', 'Inventory 3', 'I_Location 3', 'D003'),
('IN004', 'Inventory 4', 'I_Location 4', 'D001'),
('IN005', 'Inventory 5', 'I_Location 5', 'D002');

-- Create Brand table
CREATE TABLE Brand (
  Brand_name VARCHAR(50) PRIMARY KEY NOT NULL,
  Company_ID VARCHAR(20),
```

```
  Model_ID BIGSERIAL,
  Num_of_employees INTEGER
);

-- Insert values into Brand table
INSERT INTO Brand (Brand_name, Company_ID, Num_of_employees)
VALUES
('Tata Motors PV', '5001', 15000),
('Tata passenger EV', '5002', 7500);



-- Create Car Company table
CREATE TABLE Car_Company (
  COMID VARCHAR(20) PRIMARY KEY NOT NULL,
  Num_of_employees INTEGER,
  Brand_name VARCHAR(50),
  FOREIGN KEY (Brand_name) REFERENCES Brand(Brand_name)
);

-- Insert values into Car Company table
INSERT INTO Car_Company (COMID, Num_of_employees, Brand_name)
VALUES
(5001,15000, 'Tata Motors PV'),
(5002, 7500, 'Tata passenger EV');

-- Create Manufacturer table
CREATE TABLE Manufacturer (
```

```
    M_ID VARCHAR(20) PRIMARY KEY NOT NULL,
    Mname VARCHAR(50),
    Mlocation VARCHAR(50),
    Brand_name VARCHAR(30),
    FOREIGN KEY (Brand_name) REFERENCES Brand(Brand_name)
);

-- Insert values into Manufacturer table
INSERT INTO Manufacturer (M_ID, Mname, Mlocation, Brand_name)
VALUES
('M001', 'Manufacturer 1', 'M_Location 1', 'Tata Motors PV'),
('M002', 'Manufacturer 2', 'M_Location 2', 'Tata Motors PV'),
('M003', 'Manufacturer 3', 'M_Location 3', 'Tata passenger EV'),
('M004', 'Manufacturer 4', 'M_Location 4', 'Tata Motors PV'),
('M005', 'Manufacturer 5', 'M_Location 5', 'Tata passenger EV');


-- Create Supplier table
CREATE TABLE Supplier (
    SID BIGINT PRIMARY KEY NOT NULL,
    Sname VARCHAR(50),
    Slocation VARCHAR(50)
);

-- Insert values into Supplier table
INSERT INTO Supplier (SID, Sname, Slocation)
VALUES
```

```
(10001, 'Supplier 1', 'S_Location 1'),
(10002, 'Supplier 2', 'S_Location 2'),
(10003, 'Supplier 3', 'S_Location 3'),
(10004, 'Supplier 4', 'S_Location 4'),
(10005, 'Supplier 5', 'S_Location 5');

-- Create Car Model table
CREATE TABLE Car_Model (
  Model_ID VARCHAR(30) PRIMARY KEY NOT NULL,
  Model_name VARCHAR(30) NOT NULL,
  Model_year DATE,
  Body_style VARCHAR(50),
  Brand_name VARCHAR(30),
  Part_ID BIGINT, -- Added Part_ID as foreign key from Supplies table
  FOREIGN KEY (Brand_name) REFERENCES Brand(Brand_name)
);

-- Insert values into Car Model table
INSERT INTO Car_Model (Model_ID, Model_name, Model_year, Body_style, Brand_name, Part_ID)
VALUES
('Model001', 'Tata Tiago', '2023-01-01', 'Sedan', 'Tata Motors PV', 101), -- Assuming Part_ID for Tata Tiago is 1
('Model002', 'Tata Nexon', '2023-01-01', 'Small SUV', 'Tata Motors PV', 201), -- Assuming Part_ID for Tata Nexon is 2
('Model003', 'Tata Nexon EV', '2023-01-01', 'Hatchback', 'Tata passenger EV', 301), -- Assuming Part_ID for Tata Nexon EV is 3
('Model004', 'Tata Harrier', '2023-01-01', 'SUV', 'Tata Motors PV', 401), -- Assuming Part_ID for Tata Harrier is 4
```

('Model005', 'Tata Tigor EV', '2023-01-01', 'Sedan', 'Tata passenger EV', 501); -- Assuming Part_ID for Tata Tigor EV is 5

```
-- Create Supplies table
CREATE TABLE Supplies (
  Part_ID BIGSERIAL PRIMARY KEY NOT NULL,
  Part_type VARCHAR(50),
  Made_date DATE,
  Supply_date DATE,
  SID BIGINT,
  Model_ID VARCHAR(30),
  FOREIGN KEY (SID) REFERENCES Supplier(SID),
  FOREIGN KEY (Model_ID) REFERENCES Car_Model(Model_ID)
);

-- Insert values into Supplies table
INSERT INTO Supplies (Part_ID, Part_type, Made_date, Supply_date, SID, Model_ID)
VALUES
(101, 'Engine Parts', '2023-01-01', '2023-01-15', 10001, 'Model001'),
(201, 'Transmission Parts', '2023-01-02', '2023-01-16', 10002, 'Model002'),
(301, 'Body Parts', '2023-01-03', '2023-01-17', 10003, 'Model003'),
(401,'Electrical Parts', '2023-01-04', '2023-01-18', 10004, 'Model004'),
(501, 'Interior Parts', '2023-01-05', '2023-01-19', 10005, 'Model005');

-- Create Feature table
CREATE TABLE Feature (
```

```
    Feature_ID VARCHAR(20) PRIMARY KEY NOT NULL,
    Engine VARCHAR(30),
    Transmission VARCHAR(30),
    Color VARCHAR(30),
    Model_ID VARCHAR(30),
    VIN VARCHAR(20),
    FOREIGN KEY (Model_ID) REFERENCES Car_Model(Model_ID)
);

-- Insert values into Feature table
INSERT INTO Feature (Feature_ID, Engine, Transmission, Color, Model_ID, VIN)
VALUES
('F001', 'V6', 'Automatic', 'Red','Model001', 'VIN001'),
('F002', 'Inline-4', 'Manual', 'Blue', 'Model002', 'VIN002'),
('F003', 'V8', 'Automatic', 'Black', 'Model003', 'VIN003'),
('F004', 'Inline-6', 'Automatic', 'White', 'Model004', 'VIN004'),
('F005', 'V12', 'Automatic', 'Silver', 'Model005', 'VIN005');


-- Create Vehicle table
CREATE TABLE Vehicle (
    V_IN VARCHAR(20) PRIMARY KEY NOT NULL,
    Option_ID BIGINT,
    Model_name VARCHAR(30),
    Inventory_ID VARCHAR(20),
    Customer_ID VARCHAR(20),
    Manufacturing_ID VARCHAR(20),
```

```
    Feature_ID VARCHAR(20), -- Added Feature_ID as foreign key from Feature table
    FOREIGN KEY (Inventory_ID) REFERENCES Inventory(In_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(C_ID),
    FOREIGN KEY (Manufacturing_ID) REFERENCES Manufacturer(M_ID),
    FOREIGN KEY (Feature_ID) REFERENCES Feature(Feature_ID)
);

-- Insert values into Vehicle table
INSERT INTO Vehicle (V_IN, Option_ID, Model_name, Inventory_ID, Customer_ID,
Manufacturing_ID, Feature_ID)
VALUES
('VIN001', 1001, 'Tata Tiago', 'IN001', 'C001', 'M001', 'F001'),
('VIN002', 1002, 'Tata Nexon', 'IN002', 'C002', 'M002', 'F002'),
('VIN003', 1003, 'Tata Nexon EV', 'IN003', 'C003', 'M003', 'F003'),
('VIN004', 1004, 'Tata Harrier', 'IN004', 'C004', 'M004', 'F004'),
('VIN005', 1005, 'Tata Tigor EV', 'IN005', 'C005', 'M005', 'F005');

--show selected table
SELECT *
FROM Customer;

SELECT *
FROM Sales;

SELECT *
FROM Vehicle;
```

```
SELECT *
FROM Inventory;

SELECT *
FROM Dealer;

SELECT *
FROM car_model;

SELECT *
FROM Feature;

SELECT *
FROM Manufacturer;

SELECT *
FROM Brand;

SELECT *
FROM Car_company;

SELECT *
FROM Supplier;

SELECT *
FROM Supplies;
```

--table columns and data types

```
SELECT table_name, column_name, data_type

FROM information_schema.columns

WHERE table_schema = 'public';
```

To get these in post gres:

"host": "Your_Host",
"database": "Your_Database",
"user": "Your_User",
"password": "Your_Password",
"port" : "Your_Port"

```
-- Get current user

SELECT current_user;


-- Get server hostname and port

SELECT inet_server_addr() AS server_address, inet_server_port() AS server_port;


SELECT current_database();
```

Car Lifecycle management with focus on manufacturing side (Tables)

table_name	column_name	data_type
brand	company_id	character varying
brand	num_of_employee	integer
brand	brand_name	character varying
brand	model_id	bigint
car_company	num_of_employee	integer
car_company	comid	character varying
car_company	brand_name	character varying
car_model	brand_name	character varying
car_model	model_name	character varying
car_model	model_year	date
car_model	model_id	character varying
car_model	body_style	character varying
car_model	part_id	bigint
customer	gender	character varying
customer	d_id	character varying
customer	annual_income	bigint
customer	is_company	character varying
customer	c_id	character varying
customer	address	character varying
customer	name	character varying
customer	phone	bigint
dealer	dlocation	character varying
dealer	d_id	character varying
dealer	dname	character varying
employee	join_date	date
employee	name	character varying
employee	salary	numeric
employee	e_id	character varying
employee	position	character varying
employee	d_id	character varying
feature	engine	character varying
feature	model_id	character varying
feature	vin	character varying
feature	transmission	character varying
feature	color	character varying
feature	feature_id	character varying
inventory	in_location	character varying
inventory	dealer_id	character varying
inventory	in_id	character varying
inventory	in_name	character varying
manufacturer	mlocation	character varying
manufacturer	brand_name	character varying
manufacturer	m_id	character varying

manufacturer	mname	character varying
sales	c_id	character varying
sales	price	numeric
sales	d_id	character varying
sales	m_id	character varying
sales	date	date
supplier	sname	character varying
supplier	sid	bigint
supplier	slocation	character varying
supplies	model_id	character varying
supplies	supply_date	date
supplies	part_id	bigint
supplies	made_date	date
supplies	part_type	character varying
supplies	sid	bigint
vehicle	feature_id	character varying
vehicle	model_name	character varying
vehicle	v_in	character varying
vehicle	inventory_id	character varying
vehicle	manufacturing_id	character varying
vehicle	customer_id	character varying
vehicle	option_id	bigint

Customer table

	c_id [PK] character varying (20)	name character varying (50)	is_company character varying (50)	address character varying (50)	phone bigint	gender character varying (10)	annual_income bigint	d_id charact
1	C001	John Doe	Yes	123 Main St	1234567890	Male	50000	D001
2	C002	John Doe	Yes	123 Main St	1234567890	Male	50000	D002
3	C003	Michael Johnson	Yes	789 Elm Rd	5551234567	Male	60000	D003
4	C004	Emily Brown	No	321 Pine Blvd	9998887777	Female	80000	D008
5	C005	David Wilson	Yes	654 Cedar Ln	1112223333	Male	70000	D003
6	C006	Jessica Adams	No	789 Elm Rd	5559876543	Female	90000	D004
7	C007	Michael Smith	No	123 Oak St	5555555555	Male	80000	D005

Sales Table

	m_id [PK] character varying (20)	c_id character varying (20)	d_id character varying (20)	date date	price numeric (10,2)
1	M001	C001	D001	2024-03-01	2500000.00
2	M002	C002	D002	2024-03-05	3000000.00
3	M003	C003	D003	2024-03-10	3500000.00
4	M004	C004	D001	2024-03-15	4000000.00
5	M005	C005	D002	2024-03-20	4500000.00

Here I did error in column – column name should have been sales revenue and not price, and one zeroes added to each of the values

Vehicle table

	v_in [PK] character varying (20)	option_id bigint	model_name character varying (30)	inventory_id character varying (20)	customer_id character varying (20)	manufacturing_id character varying (20)	feature_id character
1	VIN001	1001	Tata Tiago	IN001	C001	M001	F001
2	VIN002	1002	Tata Nexon	IN002	C002	M002	F002
3	VIN003	1003	Tata Nexon EV	IN003	C003	M003	F003
4	VIN004	1004	Tata Harrier	IN004	C004	M004	F004
5	VIN005	1005	Tata Tigor EV	IN005	C005	M005	F005

Inventory Table

	in_id [PK] character varying (20)	in_name character varying (50)	in_location character varying (50)	dealer_id character varying (20)
1	IN001	Inventory 1	I_Location 1	D001
2	IN002	Inventory 2	I_Location 2	D002
3	IN003	Inventory 3	I_Location 3	D003
4	IN004	Inventory 4	I_Location 4	D001
5	IN005	Inventory 5	I_Location 5	D002

Dealer Table

	d_id [PK] character varying (20)	dname character varying (20)	dlocation character varying (50)
1	D001	Dealer 1	D_City 1
2	D002	Dealer 2	D_City 2
3	D003	Dealer 3	D_City 3
4	D004	Dealer 4	D_City 4
5	D005	Dealer 5	D_City 5

Car_model table

	model_id [PK] character varying (30)	model_name character varying (30)	model_year date	body_style character varying (50)	brand_name character varying (30)	part_id bigint
1	Model001	Tata Tiago	2023-01-01	Sedan	Tata Motors PV	101
2	Model002	Tata Nexon	2023-01-01	Small SUV	Tata Motors PV	201
3	Model003	Tata Nexon EV	2023-01-01	Hatchback	Tata passenger EV	301
4	Model004	Tata Harrier	2023-01-01	SUV	Tata Motors PV	401
5	Model005	Tata Tigor EV	2023-01-01	Sedan	Tata passenger EV	501

Feature table

	feature_id [PK] character varying (20)	engine character varying (30)	transmission character varying (30)	color character varying (30)	model_id character varying (30)	vin character varying (20)
1	F001	V6	Automatic	Red	Model001	VIN001
2	F002	Inline-4	Manual	Blue	Model002	VIN002
3	F003	V8	Automatic	Black	Model003	VIN003
4	F004	Inline-6	Automatic	White	Model004	VIN004
5	F005	V12	Automatic	Silver	Model005	VIN005

Manufacturer table

	m_id [PK] character varying (20)	mname character varying (50)	mlocation character varying (50)	brand_name character varying (30)
1	M001	Manufacturer 1	M_Location 1	Tata Motors PV
2	M002	Manufacturer 2	M_Location 2	Tata Motors PV
3	M003	Manufacturer 3	M_Location 3	Tata passenger EV
4	M004	Manufacturer 4	M_Location 4	Tata Motors PV
5	M005	Manufacturer 5	M_Location 5	Tata passenger EV

Employee Table

	e_id [PK] character varying (20)	name character varying (50)	position character varying (50)	salary numeric (10,2)	join_date date	d_id character varying (20)
1	E001	John Smith	Manager	80000.00	2023-01-01	D001
2	E002	Jane Doe	Salesperson	50000.00	2023-02-15	D002
3	E003	Michael Johnson	Technician	60000.00	2023-03-10	D003
4	E004	Emily Brown	Accountant	70000.00	2023-04-20	D001
5	E005	David Wilson	Engineer	75000.00	2023-05-30	D002

Brand table

	brand_name [PK] character varying (50)	company_id character varying (20)	model_id bigint	num_of_employees integer
1	Tata Motors PV	5001	1	15000
2	Tata passenger EV	5002	2	7500

Car_company table

	comid [PK] character varying (20)	num_of_employees integer	brand_name character varying (50)
1	5001	15000	Tata Motors PV
2	5002	7500	Tata passenger EV

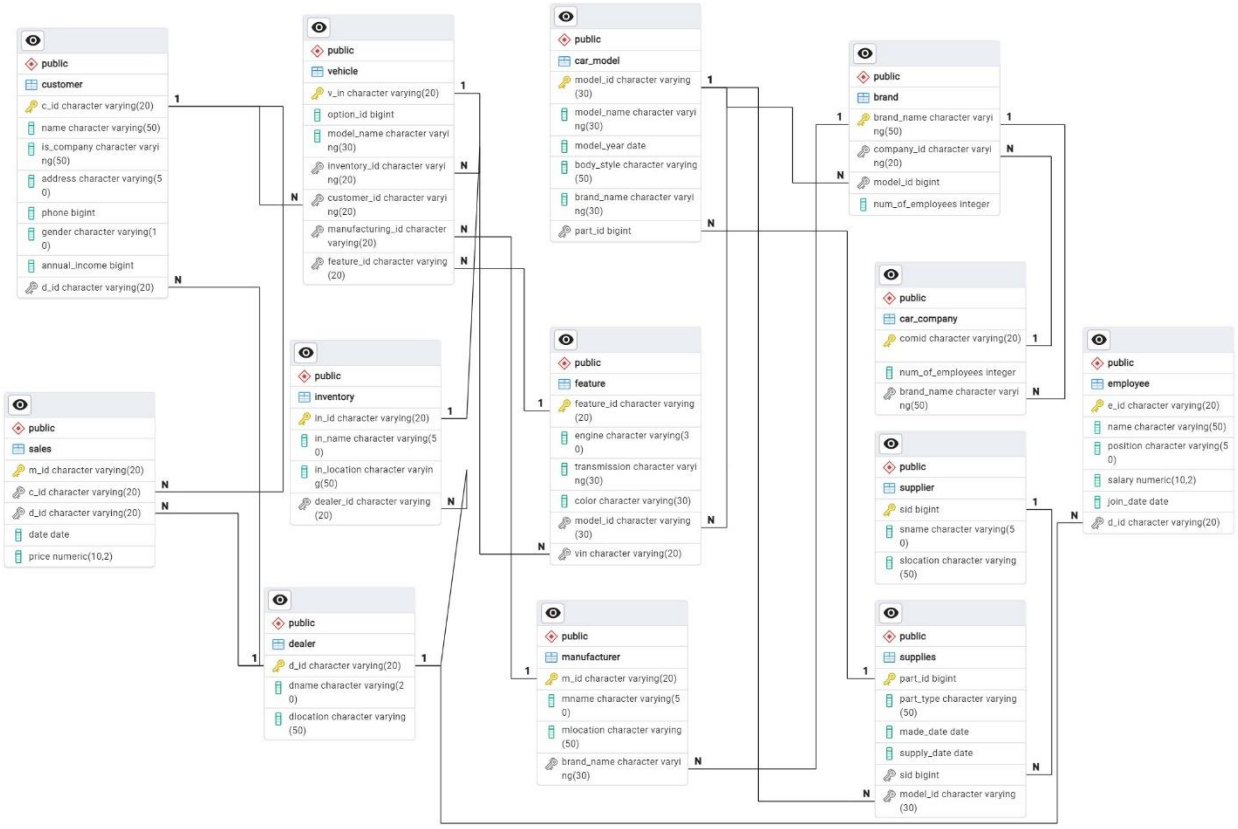
Supplier table

	sid [PK] bigint	sname character varying (50)	slocation character varying (50)
1	10001	Supplier 1	S_Location 1
2	10002	Supplier 2	S_Location 2
3	10003	Supplier 3	S_Location 3
4	10004	Supplier 4	S_Location 4
5	10005	Supplier 5	S_Location 5

Supplies table

	part_id [PK] bigint	part_type character varying (50)	made_date date	supply_date date	sid bigint	model_id character varying (30)
1	101	Engine Parts	2023-01-01	2023-01-15	10001	Model001
2	201	Transmission Parts	2023-01-02	2023-01-16	10002	Model002
3	301	Body Parts	2023-01-03	2023-01-17	10003	Model003
4	401	Electrical Parts	2023-01-04	2023-01-18	10004	Model004
5	501	Interior Parts	2023-01-05	2023-01-19	10005	Model005

ER diagram of TATA Motors Car lifecycle management with focus on manufacturing side



Test cases on DBMS Project

1. Sales Performance analysis of dealers

   SQL query:

   ```
   SELECT D.Dname, SUM (S.Price) AS Total_Sales
   FROM Sales S
   JOIN Dealer D ON S.D_ID = D.D_ID
   GROUP BY D.Dname
   ORDER BY Total_Sales DESC
   LIMIT 3;
   ```

   Output:

Query Query History

```
1  SELECT D.Dname, SUM(S.Price) AS Total_Sales
2  FROM Sales S
3  JOIN Dealer D ON S.D_ID = D.D_ID
4  GROUP BY D.Dname
5  ORDER BY Total_Sales DESC
6  LIMIT 3;
```

Data Output Messages Notifications

	dname character varying (20)	total_sales numeric
1	Dealer 2	7500000.00
2	Dealer 1	6500000.00
3	Dealer 3	3500000.00

2. **Test case 2: Manufacturer Comparison:**

SQL query:

```
SELECT M.Brand_name, COUNT(DISTINCT M.M_ID) AS Num_Manufacturers
FROM Manufacturer M
JOIN Car_Model CM ON M.Brand_name = CM.Brand_name
GROUP BY M.Brand_name
ORDER BY Num_Manufacturers DESC;
```

Output:

Query Query History

```
1  SELECT M.Brand_name, COUNT(DISTINCT M.M_ID) AS Num_Manufacturers
2  FROM Manufacturer M
3  JOIN Car_Model CM ON M.Brand_name = CM.Brand_name
4  GROUP BY M.Brand_name
5  ORDER BY Num_Manufacturers DESC;
6
```

Data Output Messages Notifications

	brand_name character varying (30)	num_manufacturers bigint
1	Tata Motors PV	3
2	Tata passenger EV	2

3. **Test case-3- Get the names of customers who made purchases from both Dealer 1 and Dealer 2**

SELECT Name
FROM Customer
WHERE C_ID IN (
  SELECT C_ID
  FROM Sales
  WHERE D_ID IN ('D001', 'D002')
);

Output:

```
16 --Test case-3- -- Get the names of customers who made purchases from both Dealer 1 and Dealer 2
17 SELECT Name
18 FROM Customer
19 WHERE C_ID IN (
20     SELECT C_ID
21     FROM Sales
22     WHERE D_ID IN ('D001', 'D002')
23 );
```

Data Output Messages Notifications

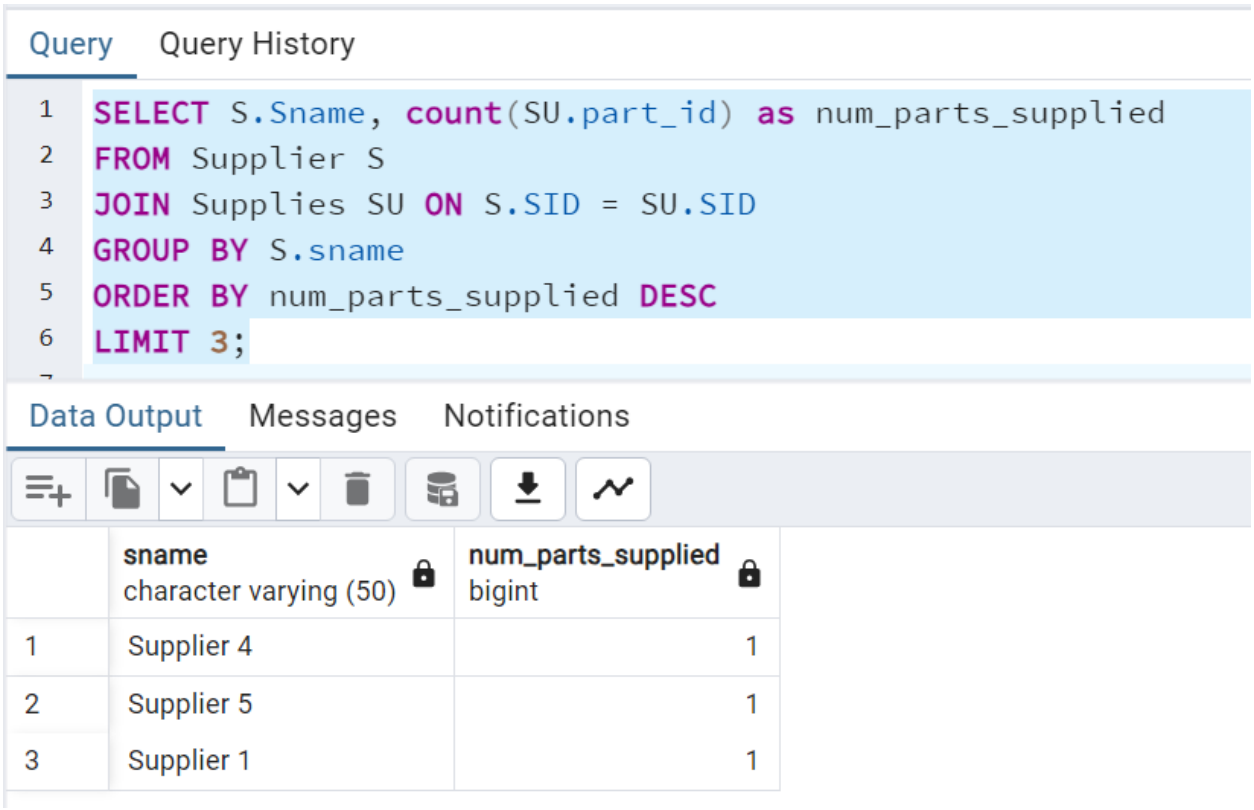
	name character varying (50)
1	John Doe
2	Jane Smith
3	Emily Brown
4	David Wilson

4. **Test Case 4: Supplier inventory check**

   SQL Query:

   SELECT S.name, count(SU.part_id) as num_parts_supplied
   FROM Supplier S
   JOIN Supplies SU ON S.SID = SU.SID
   GROUP BY S.sname
   ORDER BY num_parts_supplied DESC
   LIMIT 3;

   Output:



	sname character varying (50)	num_parts_supplied bigint
1	Supplier 4	1
2	Supplier 5	1
3	Supplier 1	1

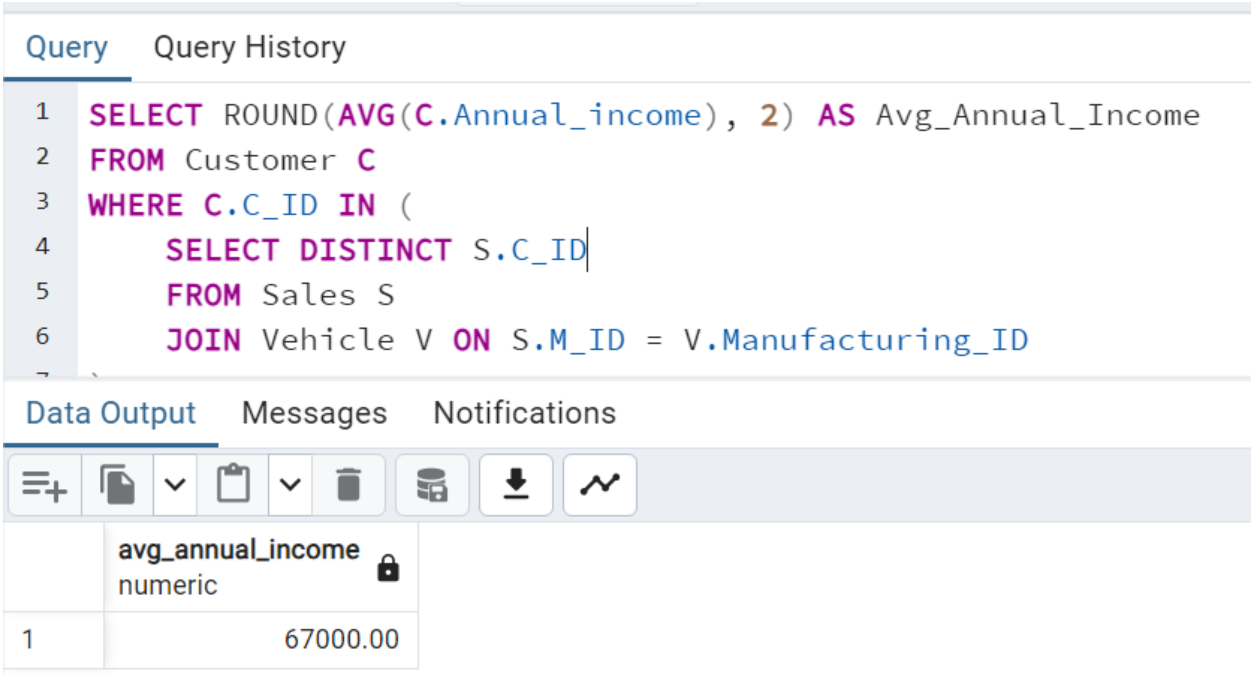
5. **Test Case 5:**

   SQL Query:

   SELECT ROUND(AVG(C.Annual_income), 2) AS Avg_Annual_Income
   FROM Customer C
   WHERE C.C_ID IN (
     SELECT DISTINCT S.C_ID

```
    FROM Sales S
    JOIN Vehicle V ON S.M_ID = V.Manufacturing_ID
);
```

Output:



Test case 6.