- **SQL Scrips to Create tables and output screens-**

First, I show a screenshot of the MySQL Workbench environment with the script (Figures 8 and 14). Then, I present the script for each table followed by a screenshot of the table created (Figures 9, 10, 11, 12, 13, 15, 16, 17 and 18).

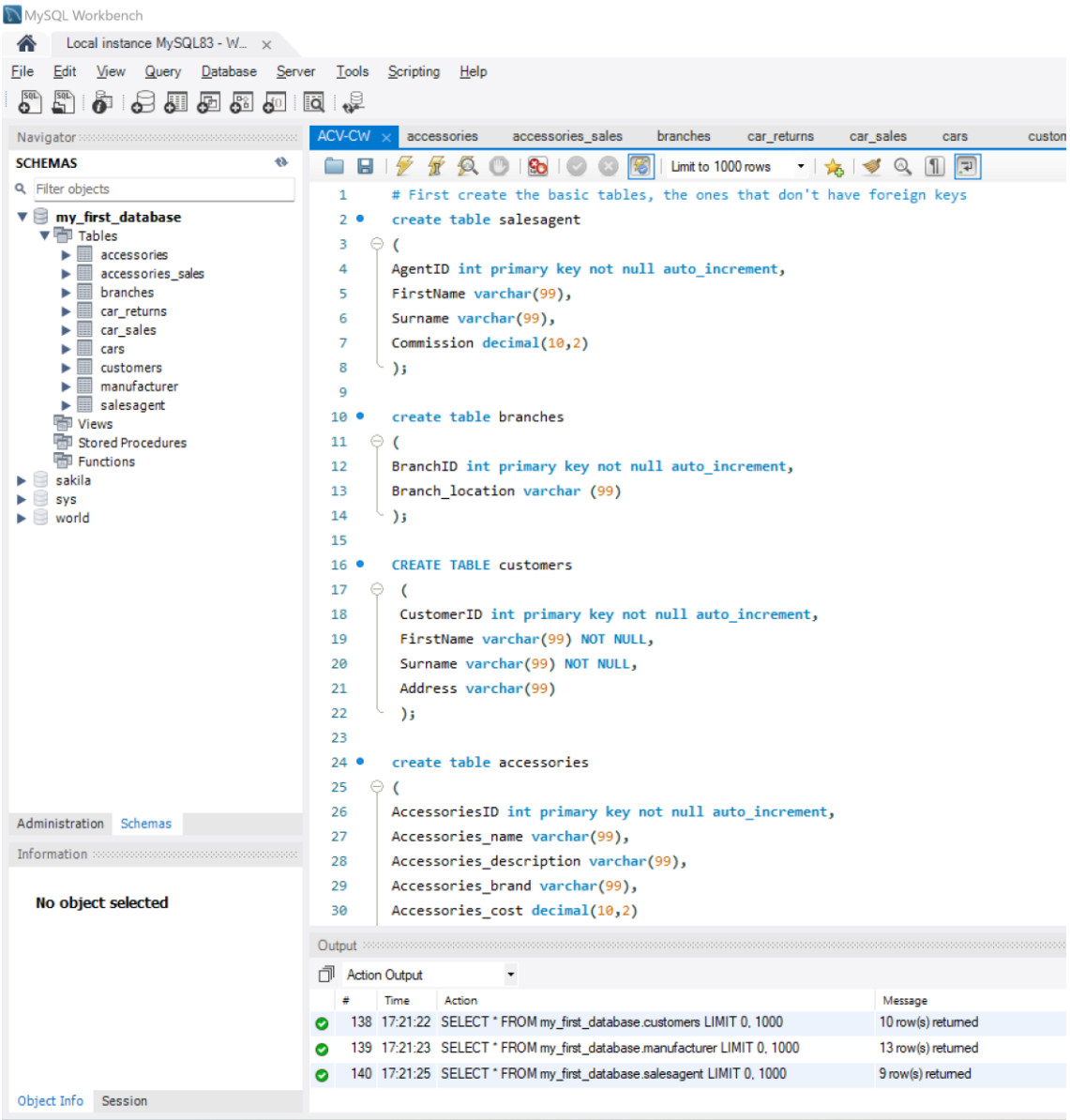

Figure 1. Screenshot of the script after running in MySQL Workbench

Script to create table salesagent:

```
create table salesagent
(
AgentID int primary key not null auto_increment,
FirstName varchar(99),
Surname varchar(99),
Commission decimal(10,2)
);
```

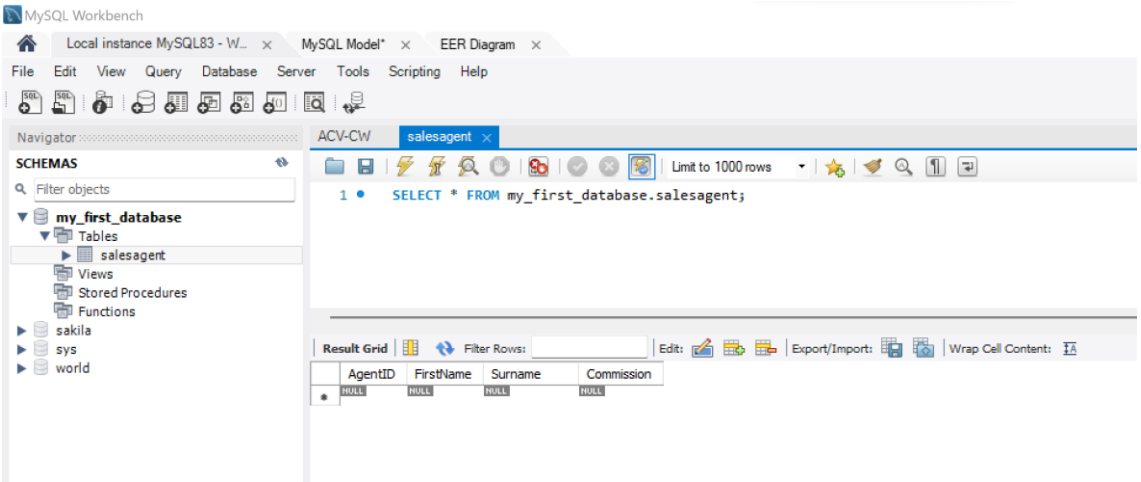

Figure 2. Screenshot of table Sales Agent in MySQL Workbench

Script to create table branches:

```
create table branches
(
BranchID int primary key not null auto_increment,
Branch_location varchar (99)
);
```

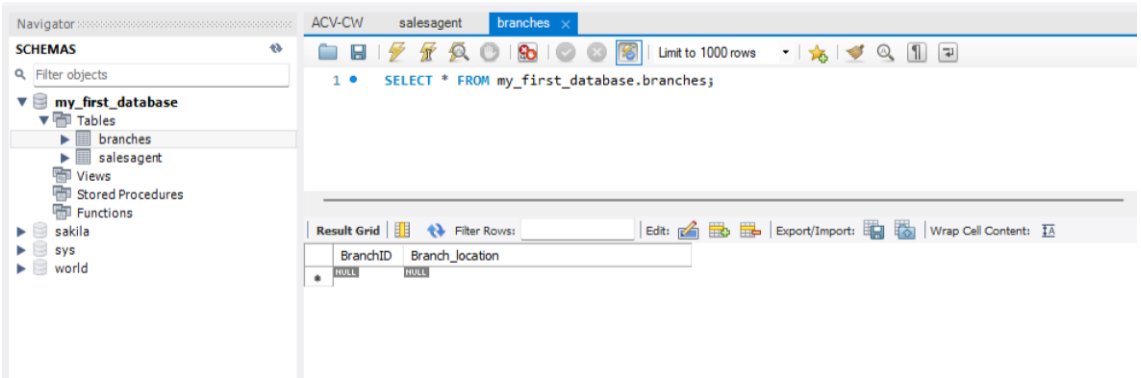

Figure 3. Screenshot of table Branches in MySQL Workbench

Script to create table customers:

```
CREATE TABLE customers
 (
 CustomerID int primary key not null auto_increment,
 FirstName varchar(99) NOT NULL,
 Surname varchar(99) NOT NULL,
 Address varchar(99)
 );
```

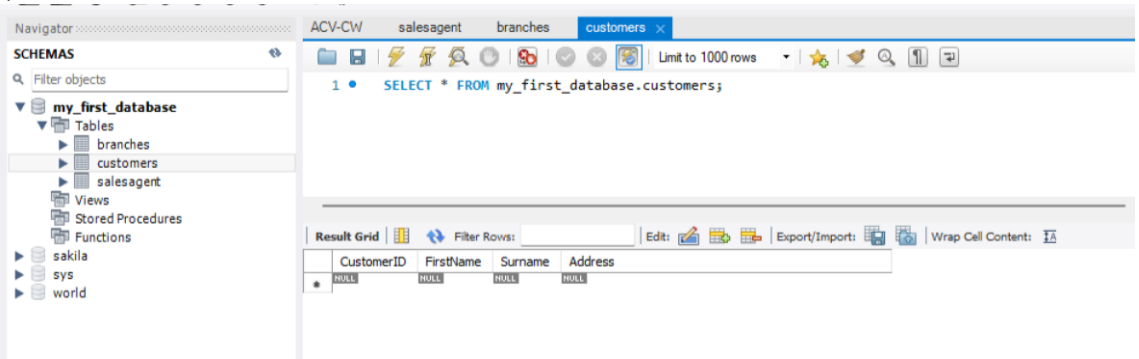

Figure 4. Screenshot of table Customers in MySQL Workbench

Script to create table accessories:

```
create table accessories
(
AccessoriesID int primary key not null auto_increment,
Accessories_name varchar(99),
Accessories_description varchar(99),
Accessories_brand varchar(99),
Accessories_cost decimal(10,2)
 );
```

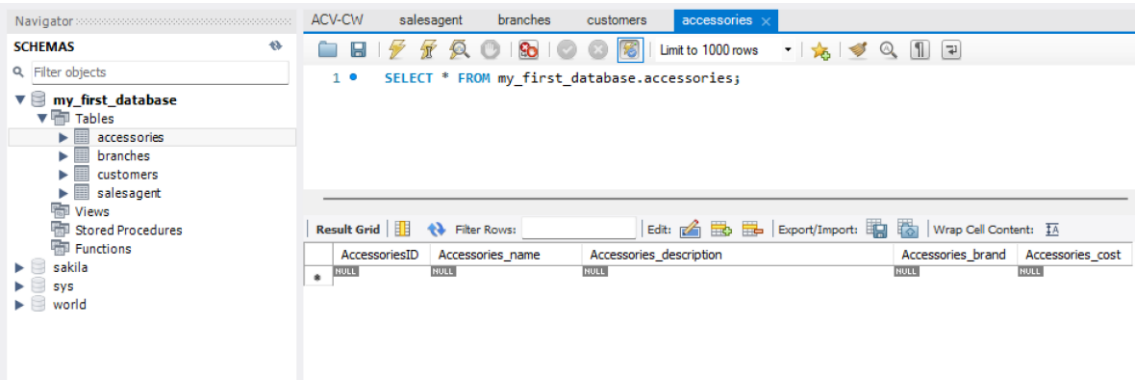

Figure 5. Screenshot of table Accessories in MySQL Workbench

Script to create table manufacturer:

```
CREATE TABLE manufacturer
 (
 ManufacturerID int primary key not null auto_increment,
 Manufacturer varchar(99),
 Manufacturing_date date
 );
```

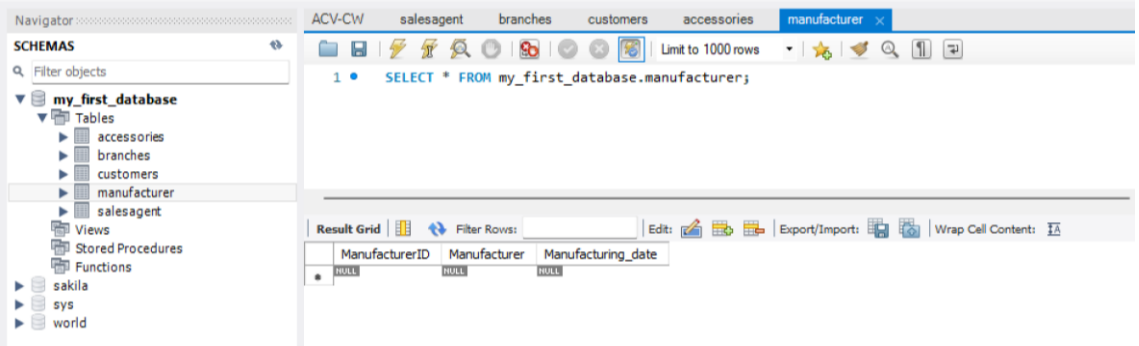

Figure 6. Screenshot of table Manufacturer in MySQL Workbench

Now the tables with foreign keys

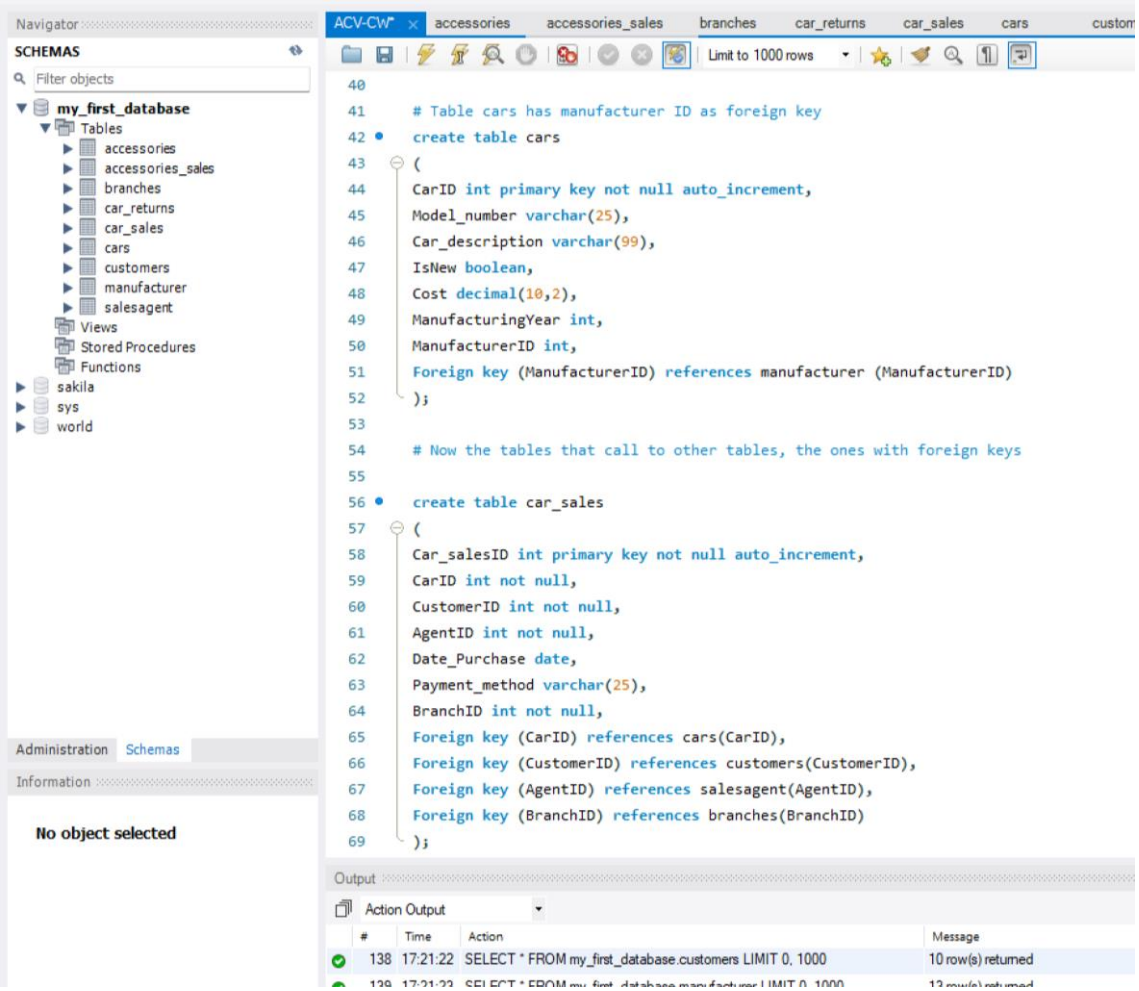

Figure 7. Screenshot of the script to create the tables after running in MySQL Workbench

Script to create table cars:

```
create table cars
(
CarID int primary key not null auto_increment,
Model_number varchar(25),
Car_description varchar(99),
IsNew boolean,
Cost decimal(10,2),
ManufacturingYear int,
ManufacturerID int,
Foreign key (ManufacturerID) references manufacturer (ManufacturerID)
);
```

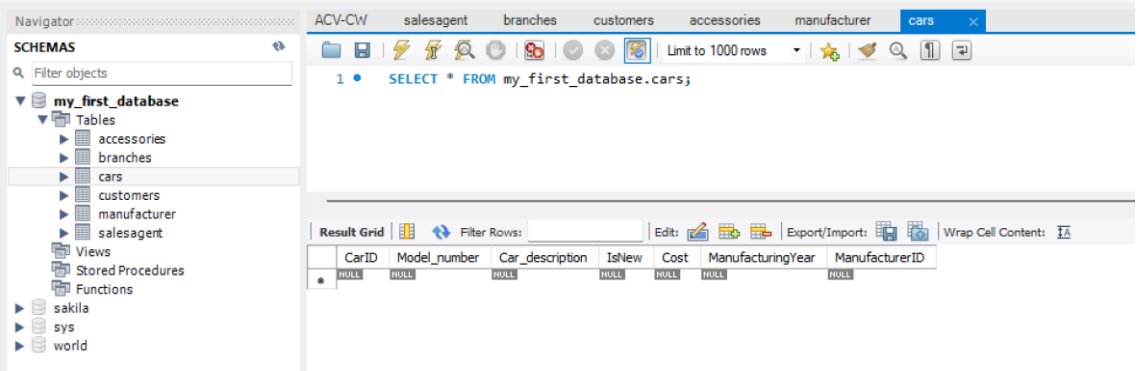

Figure 8. Screenshot of table Cars in MySQL Workbench

Script to create table car_sales:

```
create table car_sales
(
Car_salesID int primary key not null auto_increment,
CarID int not null,
CustomerID int not null,
AgentID int not null,
Date_Purchase date,
Payment_method varchar(25),
BranchID int not null,
Foreign key (CarID) references cars(CarID),
Foreign key (CustomerID) references customers(CustomerID),
Foreign key (AgentID) references salesagent(AgentID),
Foreign key (BranchID) references branches(BranchID)
);
```

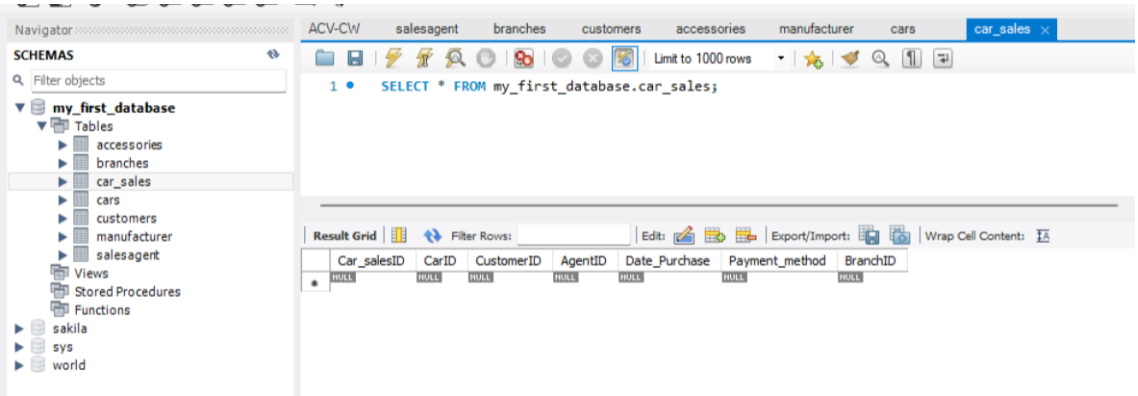

Figure 9. Screenshot of table Car Sales in MySQL Workbench

Script to create table accessories_sales:

```
create table accessories_sales
 (
 Accessories_salesID int primary key not null auto_increment,
 AccessoriesID int not null,
 CustomerID int not null,
 AgentID int not null,
 Date_Purchase date,
 Payment_method varchar(25),
 BranchID int not null,
 Foreign key (AccessoriesID) references accessories(AccessoriesID),
 Foreign key (CustomerID) references customers(CustomerID),
 Foreign key (AgentID) references salesagent(AgentID),
 Foreign key (BranchID) references branches(BranchID)
 );
```

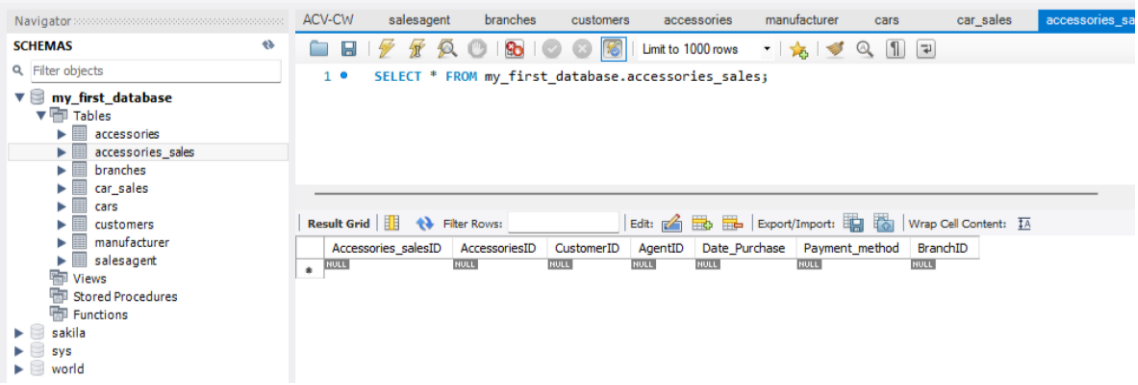

Figure 10. Screenshot of table Accessories Sales in MySQL Workbench

Script to create table car_returns:

```
create table car_returns
(
car_returnID int primary key not null auto_increment,
CarID int not null,
CustomerID int not null,
AgentID int not null,
Date_return date,
Payment_method varchar(25),
BranchID int not null,
Foreign key (CarID) references cars(CarID),
Foreign key (CustomerID) references customers(CustomerID),
Foreign key (AgentID) references salesagent(AgentID),
Foreign key (BranchID) references branches(BranchID)
);
```

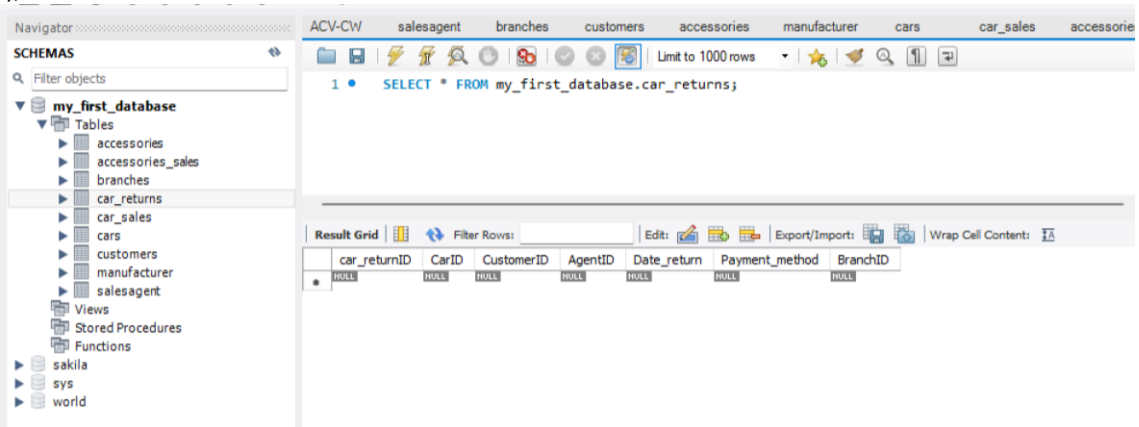

Figure 11. Screenshot of table Car Returns in MySQL Workbench

- **SQL Scripts to Populate the database-**

Figure 19 shows a screenshot of the MySQL Workbench environment with the script of this section. Then, I copied the segment of the script to populate each table, accompanied by the screenshot of the corresponding table.

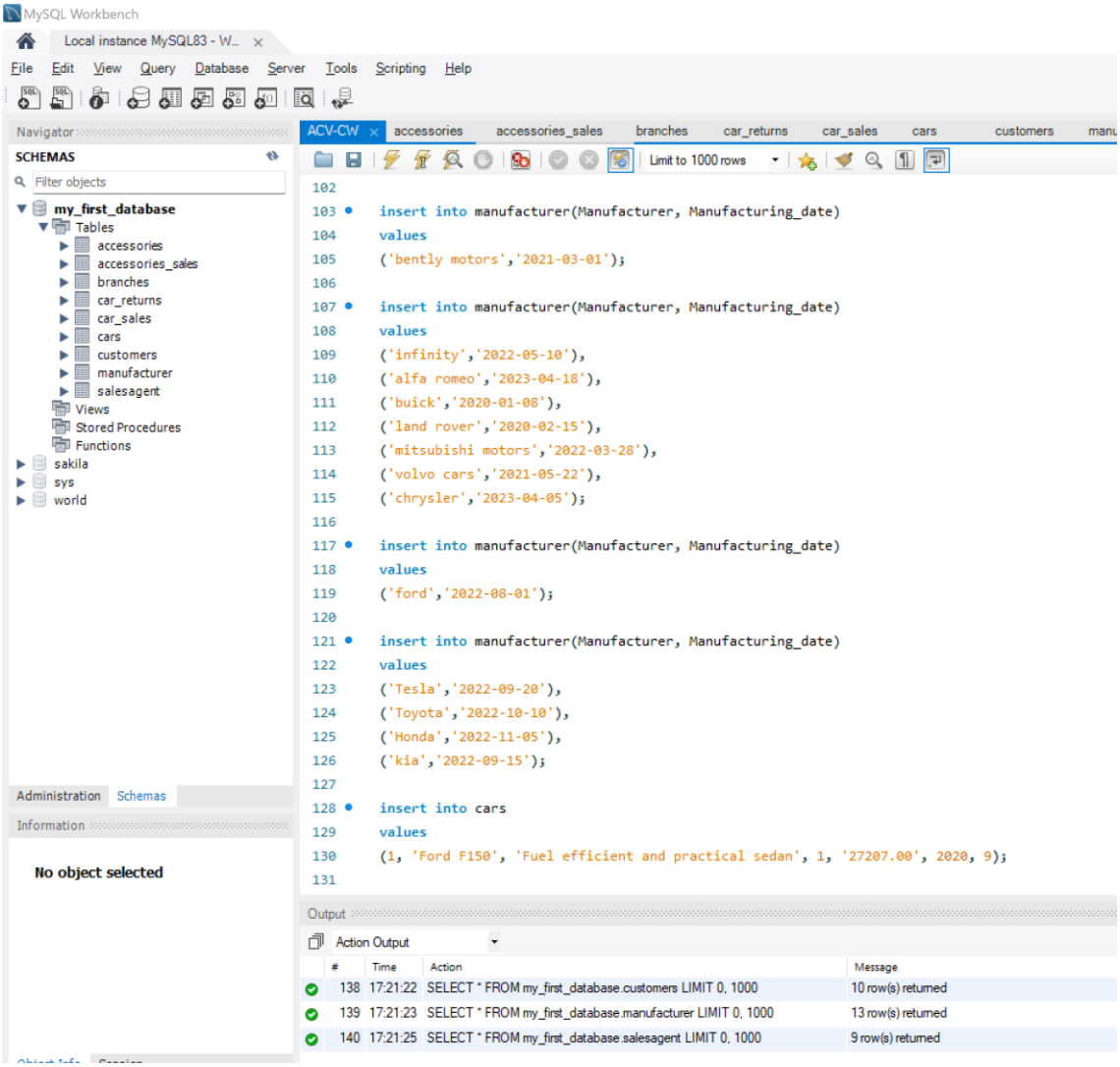

Figure 12. Screenshot of the script in MySQL workbench, populating the tables

script to populate table manufacturer

```
insert into manufacturer(Manufacturer, Manufacturing_date)
values
('bently motors','2021-03-01');

insert into manufacturer(Manufacturer, Manufacturing_date)
values
('infinity','2022-05-10'),
('alfa romeo','2023-04-18'),
('buick','2020-01-08'),
('land rover','2020-02-15'),
('mitsubishi motors','2022-03-28'),
```

```
('volvo cars','2021-05-22'),
('chrysler','2023-04-05');

insert into manufacturer(Manufacturer, Manufacturing_date)
values
('ford','2022-08-01');

insert into manufacturer(Manufacturer, Manufacturing_date)
values
('Tesla','2022-09-20'),
('Toyota','2022-10-10'),
('Honda','2022-11-05'),
('kia','2022-09-15');
```

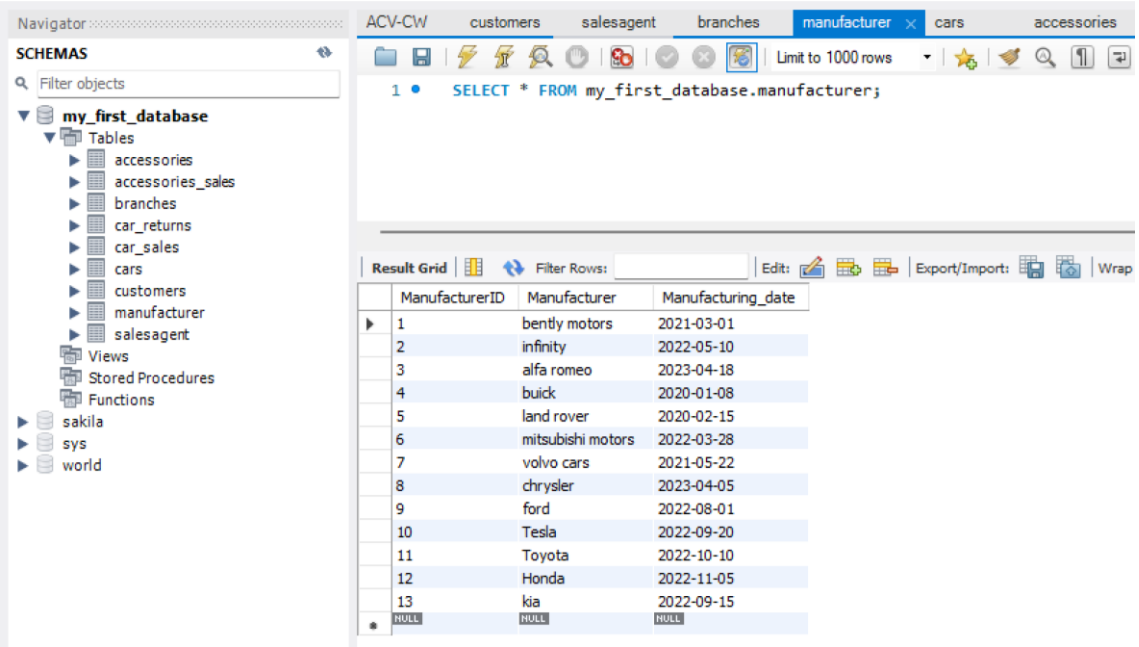

Figure 13. Screenshot of Manufacturers table with data

script to populate table cars

```
insert into cars
values
(1, 'Ford F150', 'Fuel efficient and practical sedan', 1, '27207.00', 2020, 9);

insert into cars(Model_number, Car_description, IsNew, Cost, ManufacturingYear, ManufacturerID)
values
('Tesla Model 3', 'Sleek and reliable sedan', 1, '22090.00', 2021, 10),
('Toyota Camry', 'Luxurious and high-tech electric car', 1, '33090.00',2019, 11),
('Honda Accord', 'Safe and reliable SUV', 0, '20500.00', 2021, 12),
('Honda Accord', 'Fuel-efficient and practical sedan', 0, '22490.00', 2022, 12),
('Honda Accord', 'Spacious and comfortable family car', 0, '20200.00', 2020, 12),
('Toyota Camry', 'Fuel-efficient and practical sedan', 1, '42090.00', 2020, 11),
('Kia Telluride', 'Spacious and comfortable family car', 0, '22090.00', 2019, 13);
```

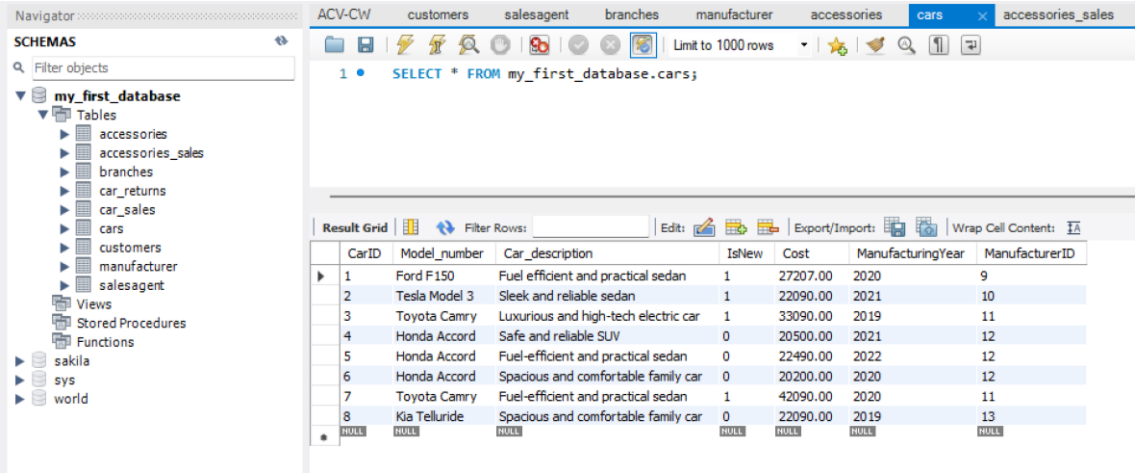

Figure 14. Screenshot of Cars table with data

script to populate table customers

```
insert into customers
values
(10001, 'Eric', 'Carl', '33a, Peckham Road, London, SE5 8BA');

insert into customers(FirstName, Surname, Address)
values
('Roald', 'Dahl', '147, The Quays, London, SE1 2LZ'),
('Quentin', 'Blake', '242, Deptford High Street, London, SE8 5DH'),
('Tim', 'Minchin', '173, Lewisham High Street, London, SE13 6JN'),
('Kate', 'Pankhurst', '27, The Cut, London, SE1 8LF'),
('Michael', 'Rosen', '12, Creek Road, Greenwich, London, SE8 3RJ'),
('Julia', 'Donaldson', '107, Queen Elizabeth Street, London, SE1 9NE'),
('Lynley', 'Dodd', '27, Royal Hill, Greenwich, London, SE10 8RF'),
('Oliver', 'Jeffers', '152, Stansted Road, London, SE23 1EW'),
('Jill', 'Murphy', '41, Blackheath Hill, London, SE10 8DJ');
```

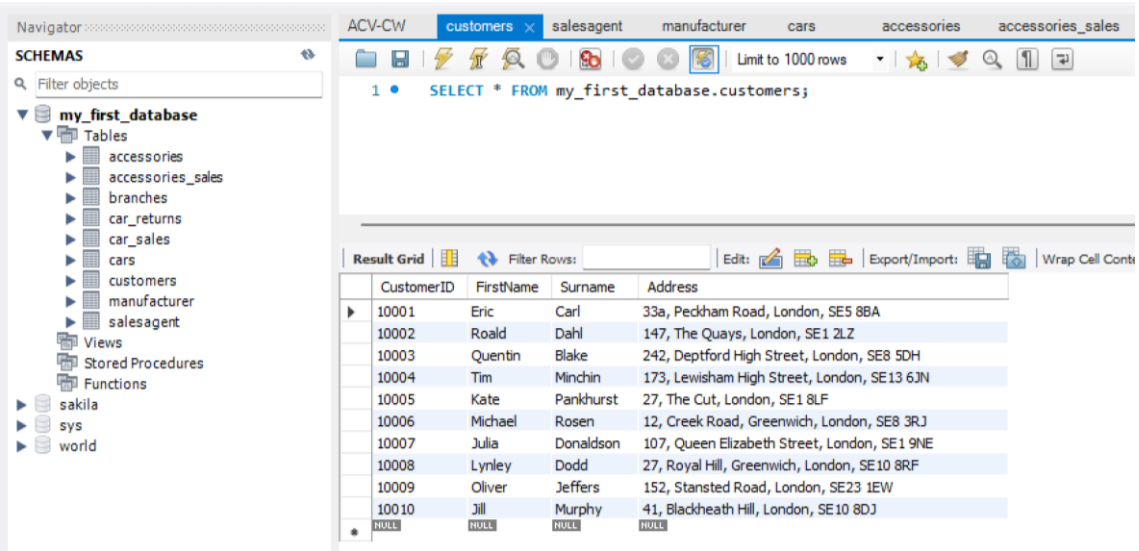

Figure 15. Screenshot of Customers table with data

script to populate table salesagent

```
insert into salesagent
values
(001, 'Harry', 'Potter', 0.03);

insert into salesagent(FirstName, Surname, Commission)
values
('Mildred', 'Hubble', 0.04),
('Matilda', 'Wormwood', 0.05),
('Danny', 'Williams', 0.04),
('Willy', 'Wonka', 0.03),
('Sophie', 'Giant', 0.04),
('Maisie', 'Mouse', 0.03),
('Harry', 'Caterpillar', 0.03),
('Elmer', 'Elephant', 0.04);
```

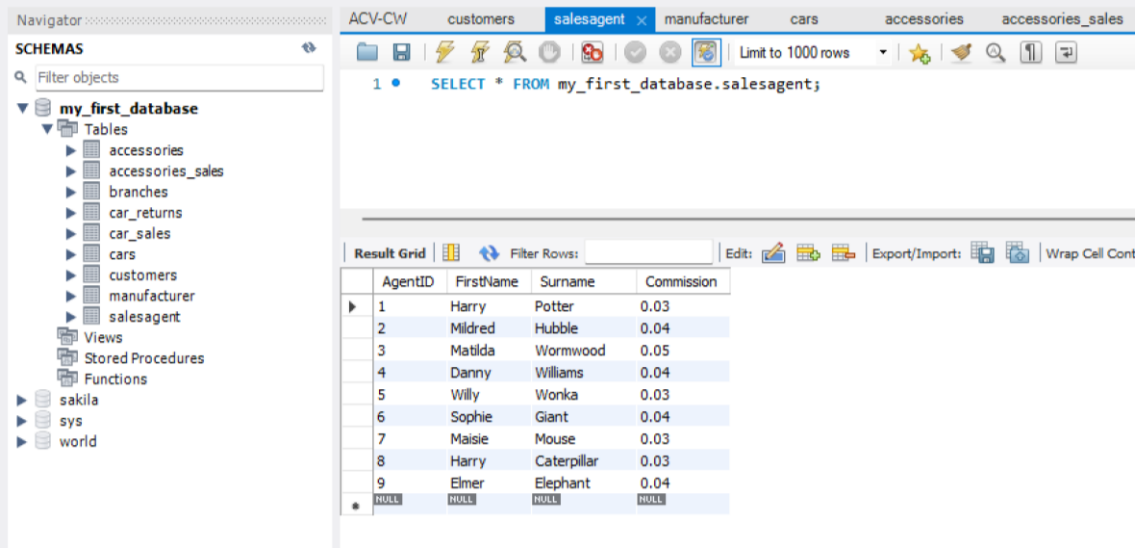

Figure 16. Screenshot of sales agents table with data

script to populate table branches

```
insert into branches
values
(10, '21 Rolt Street, Deptford, London SE8 4NF');

insert into branches(Branch_location)
values
('N Woolwich Rd, London, E16 2HP'),
('1 Midnight Ave, London, SE5 0SE'),
('173, Lewisham High Street, London, SE13 6JN'),
('27, The Cut, London, SE1 8LF'),
('31 Elmira Rd, London, "SE13 7DW');
```

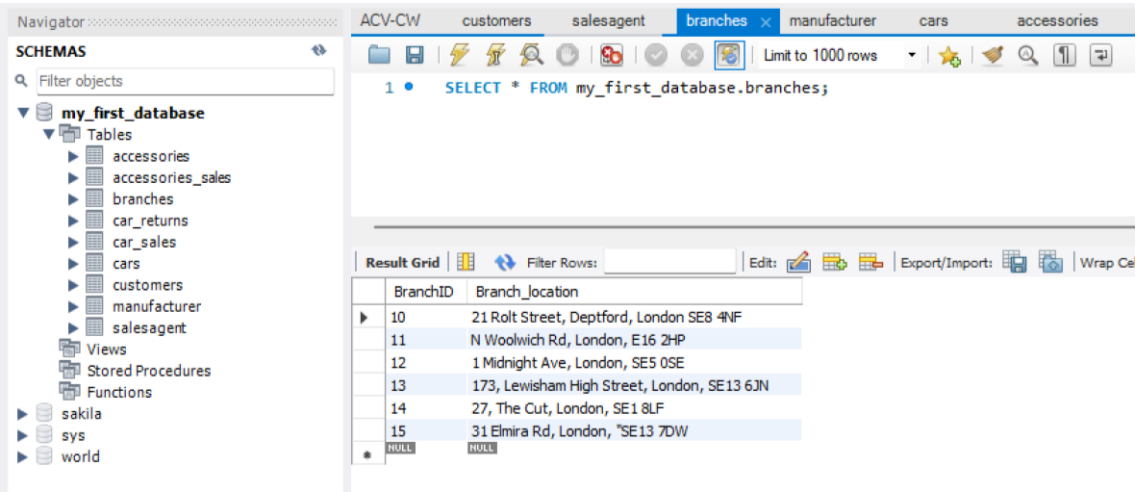

Figure 17. Screenshot of branches table with data

script to populate table accessories

insert into accessories
values
(1000, 'Phone mount', "It attaches to your car's air vent or dashboard and is adjustable to fit most phones", 'Anker', 12.99);

insert into accessories(Accessories_name, Accessories_description, Accessories_brand, Accessories_cost)
values
('Dash cam', "Mounts to your car's windshield. It has a 1080p resolution and a wide field of view", 'Vantrue', 69.99),
('Car trunk organiser', "Organizer with multiple compartments and dividers to keep everything neat and tidy", "TrunkSpace", 24.99),
('Wireless car charger', "Mounts to your car's air vent or dashboard and charges your phone through its case.","Anker", 16.99),
('Car windows sunshade', "Made of a breathable mesh fabric that allows air to circulate while still blocking the sun.", "ShadePro", 9.99),
('Steering wheel cover', "Made of a soft, breathable material, protects your steering wheel from wear and tear", "FHGroup", 12.99);

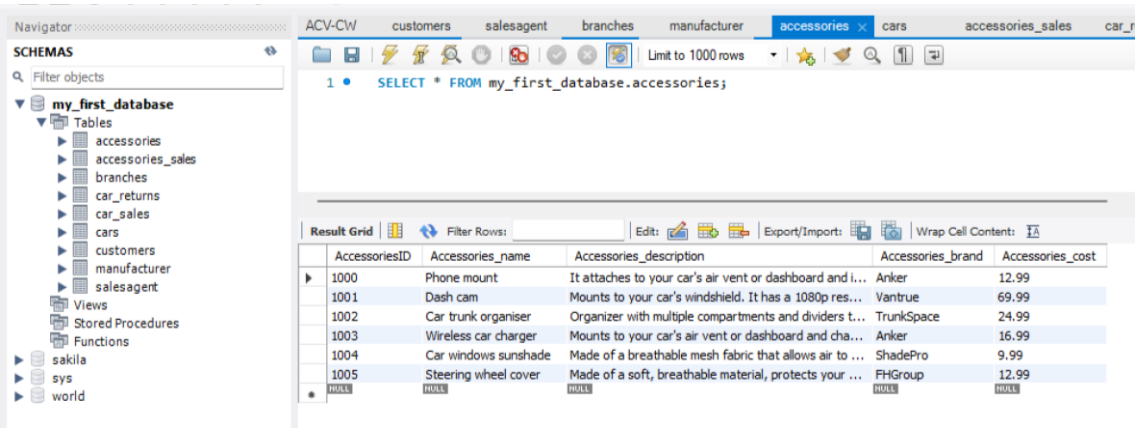

Figure 18. Screenshot of accessories table with data

script to populate table car_sales

insert into car_sales
values
(240000, 2, 10003, 004, '2024-01-22', 'Credit', 11);

insert into car_sales(CarID, CustomerID, AgentID, Date_Purchase, Payment_method, BranchID)
values
(3, 10004, 005, '2024-01-23', 'Bank Transfer', 12),
(4, 10005, 003, '2024-02-24', 'Lease', 13),
(5, 10002, 002, '2024-03-25', 'Trade-in', 14);

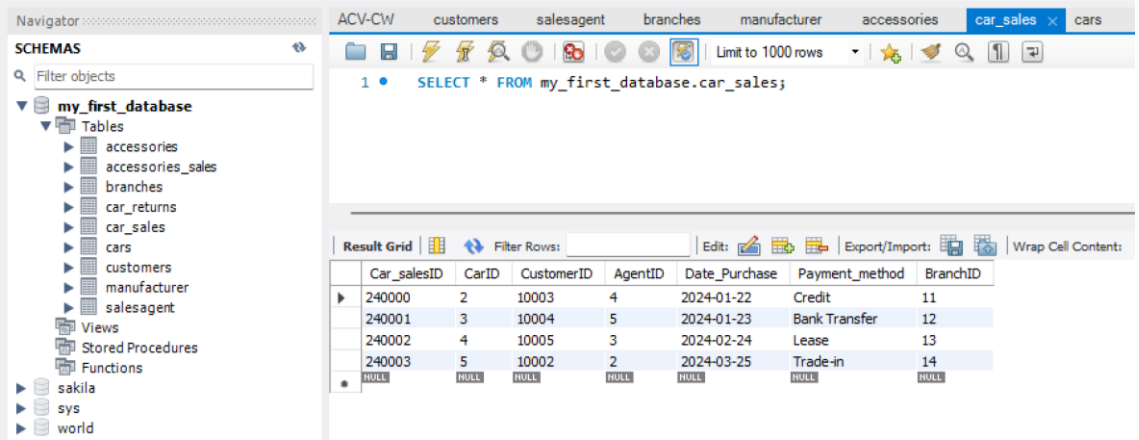

Figure 19. Screenshot of Car sales table with data

script to populate table accessories_sales

insert into accessories_sales
values
(2400000, 1000, 10001, 001, '2024-02-22', 'Bank Transfer', 10);

insert into accessories_sales(AccessoriesID, CustomerID, AgentID, Date_Purchase, Payment_method, BranchID)
values
(1001, 10010, 009, '2024-02-23', 'Bank Transfer', 11),
(1002, 10009, 008, '2024-03-24', 'Cash', 12),
(1003, 10008, 007, '2024-03-25', 'Credit', 13);

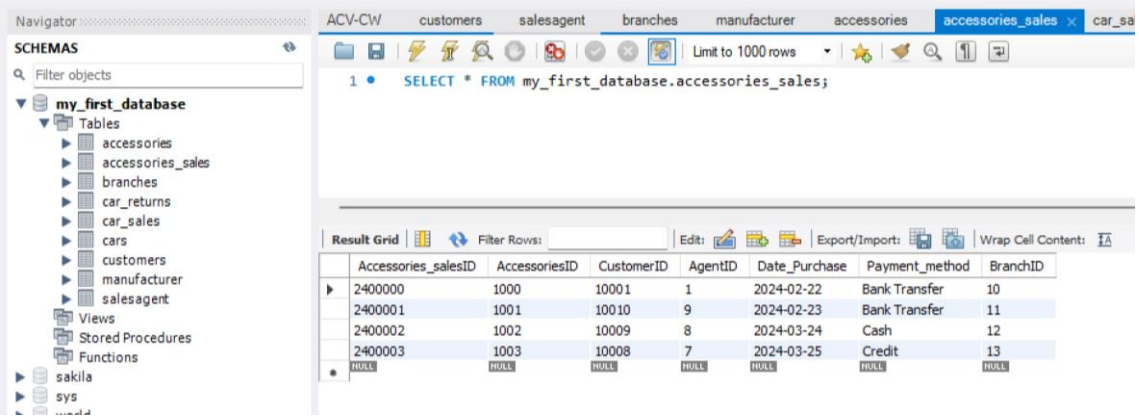

Figure 20. Screenshot of Accessories sales table with data

script to populate table car_returns

insert into car_returns
values
(24000, 6, 10008, 007, '2022-11-15', 'Trade-in', 10);

insert into car_returns(CarID, CustomerID, AgentID, Date_return, Payment_method, BranchID)

values
(5, 10007, 006, '2022-11-16', 'Trade-in', 11),
(4, 10006, 005, '2022-11-17', 'Trade-in', 12);

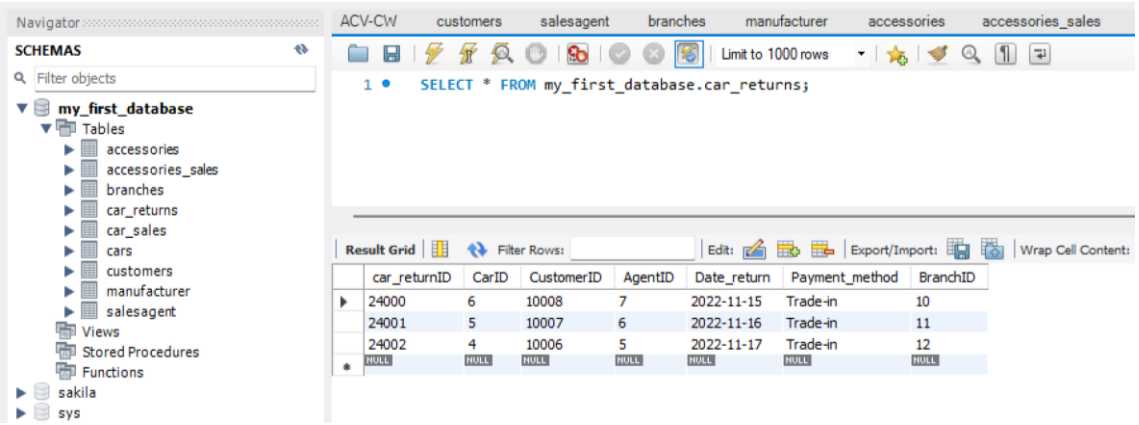

Figure 21. Screenshot of Car returns table with data

2. Demonstration of functionality.

- **SQL Scripts to manipulate the database**

To see all the registers in the car_sales table:

select * from car_sales;

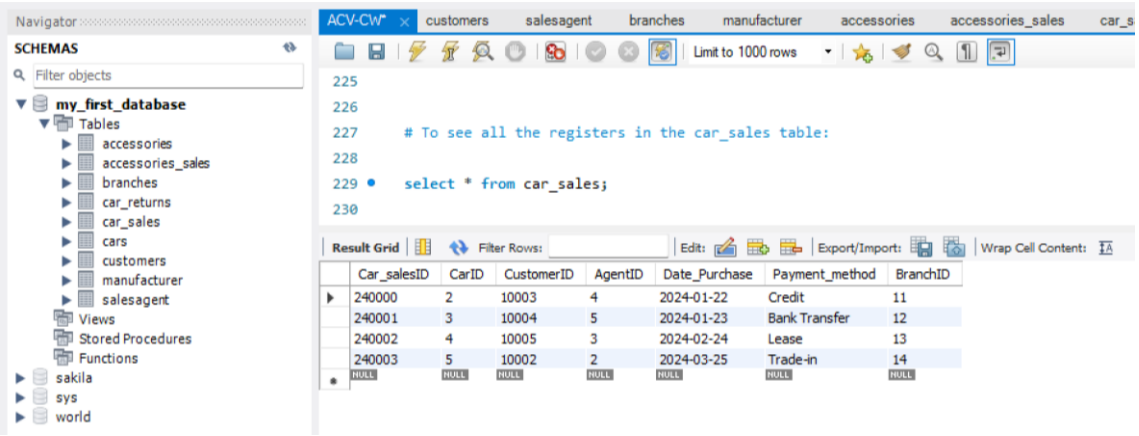

Figure 22. Screenshot of query select *and output.*

To show in the cars table all the cars manufactured in the year 2020

select * from cars where ManufacturingYear=2020;

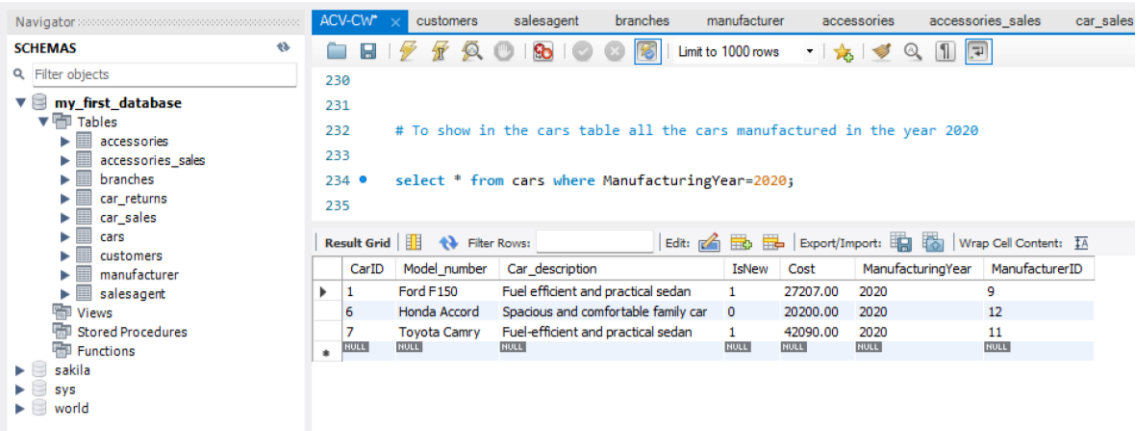

Figure 23. Screenshot of query select... where *and output.*

To show all the sales agents sorted by surname in ascending order:

select * from salesagent
order by Surname asc;

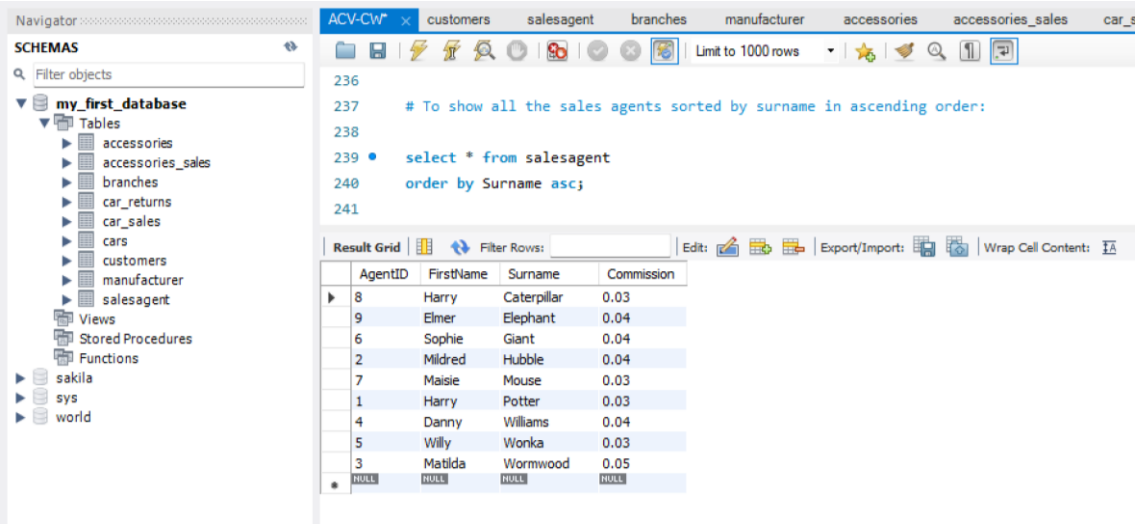

Figure 24. Screenshot of query select... order by *and output.*

- **SQL Codes to join two or more table and output screens**

To show the names of the customers from the accessories_sales table, and the accessories they bought

```
select Accessories_name, FirstName, Surname
from customers inner join accessories_sales
on customers.CustomerID = accessories_sales.CustomerID
inner join accessories
on accessories_sales.AccessoriesID = accessories.AccessoriesID;
```

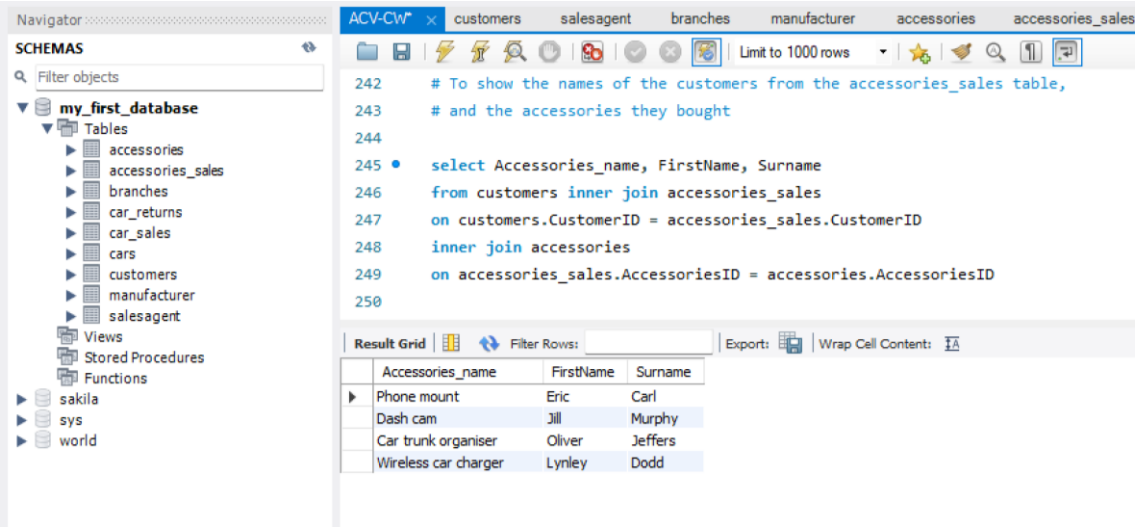

Figure 25. Screenshot of query with inner join and output.

To show the information of the cars that appear in both the car_returns table and the car_sales table

```
select * from car_returns inner join car_sales
on car_returns.CarID = car_sales.CarID;
```

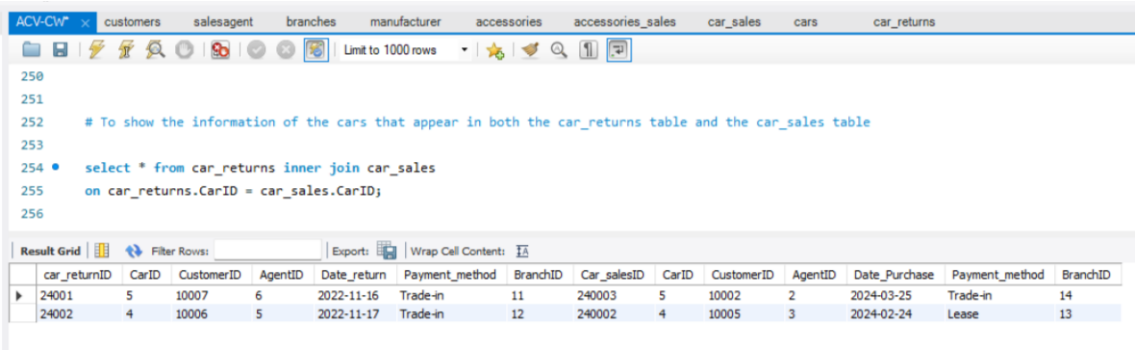

Figure 26. Screenshot of query with inner join and output.