JWheels Dealership project

Made by: Jesus Nunez

December 2, 2022

CS 4318: Database Management Systems

Professor: Ling Xu

University of Houston - Downtown

Table of Contents

Introduction

Abstract:

The database application for this project will be based on simulating a dealership. The dealership database will contain two departments. The first department will be the sales department. The sales department will contain customer data, employee data, dealership data, vehicle sales data, and customer vehicle data if they want to do a trade in. The second department will be known as the service department. The service department will contain service type data, customer vehicle data, customer data, parts data, and service employee data. The sales department and service department will have different data that will be connected to a dealership.

Mission statement:

Our mission is to make an easy-access database that stores information securely and to ensure proper customer service. We strive to grow as a business and to become one of the most reliable dealerships in the world.

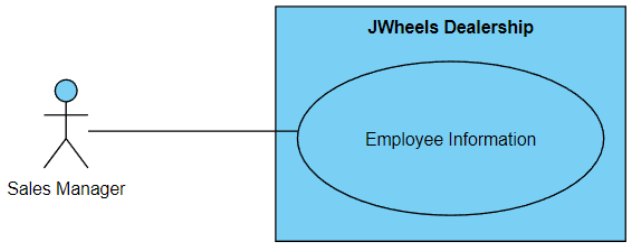
Mission Objectives:

- To maintain (enter, update, and delete) data on sales customer.
- To maintain (enter, update, and delete) data on sales employee.
- To maintain (enter, update, and delete) data on sales vehicles.
- To maintain (enter, update, and delete) data on dealerships.
- To maintain (enter, update, and delete) data on sales invoice.
- To maintain (enter, update, and delete) data on service employee.
- To maintain (enter, update, and delete) data on service customer.
- To maintain (enter, update, and delete) data on service invoice.
- To perform searches on customers.
- To perform searches on sales employees.
- To perform searches on sales vehicles.
- To perform searches on dealerships.
- To perform searches on sales invoice.
- To perform searches on service employee.

- To perform searches on service customer.
- To perform searches on service invoice.
- To track the status of dealership sales vehicles.
- To track the status of customer payments.
- To track the status of invoice.
- To track the status of parts.
- To report on sales customer.
- To report on sales employee.
- To report on dealership.
- To report on service customer.
- To report on service employee.
- To report on service parts.
- To report on service invoice.

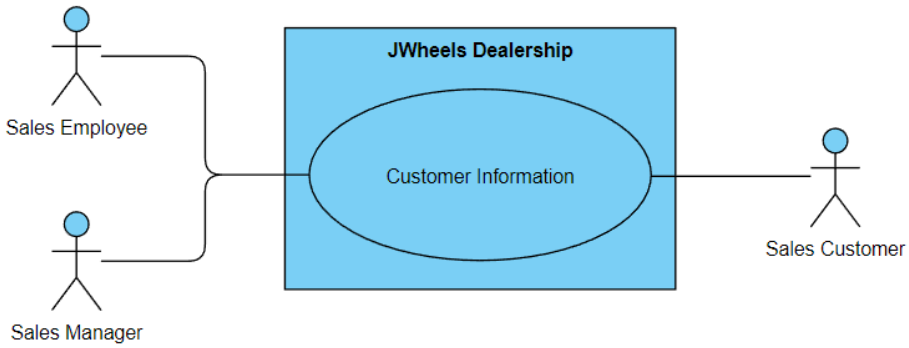
Sales Department Use cases:

Employee Information



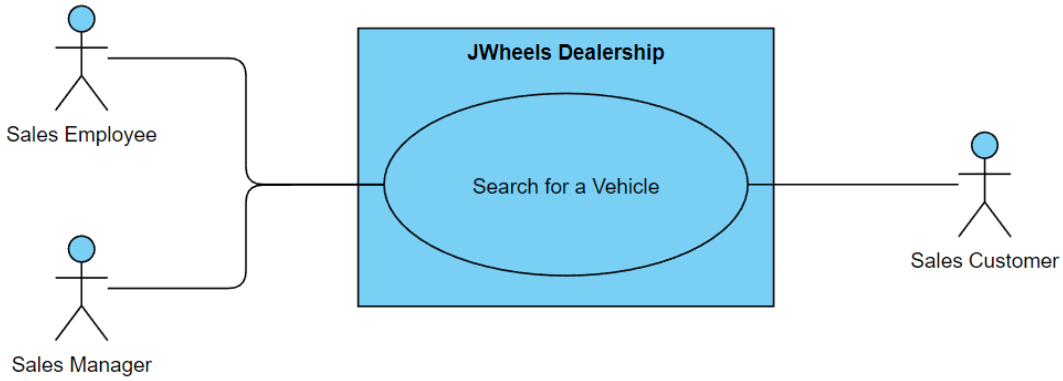
Brief Description: Dealership sales managers can add, edit, or delete employee information. Generally, managers will have more privileges when using the database.

Customer Information



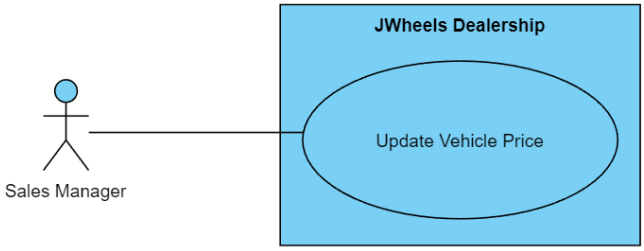
Brief Description: Dealerships normally get customers information when making sales. Therefore, sales employees and sales managers will have the privilege to add, edit, or delete customers information.

Search for a Vehicle



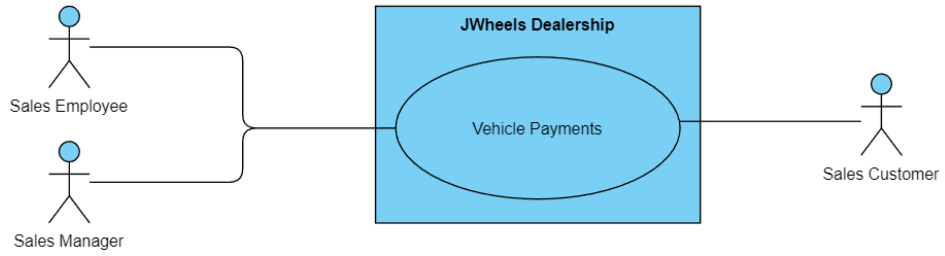
Brief Description: Managers or employees can assist a customer to find a new or used vehicle. In this case, the database will also contain the number of vehicles in stock.

Update Vehicle Price



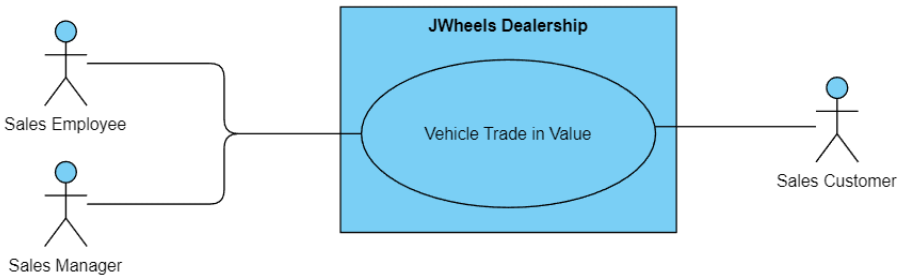
Brief Description: A manager only has the privilege to changing car prices. Employees must communicate with managers about any customer price change request such as discounts or other vehicle offers.

Vehicle Payments



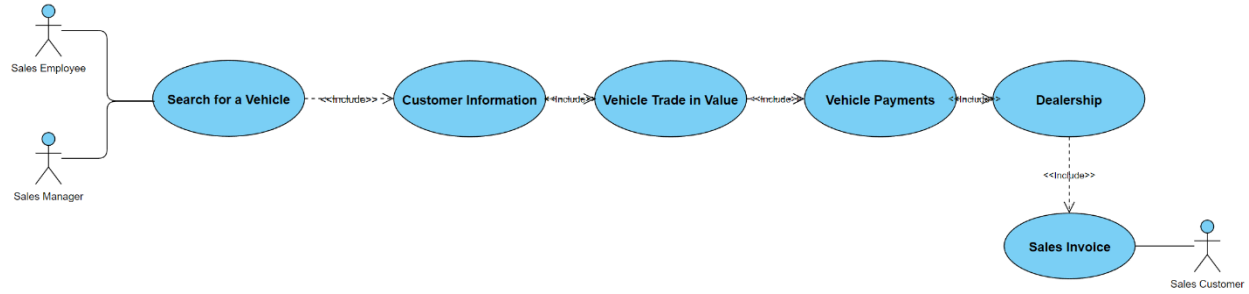
Brief Description: Both sales manager and employee can set a customer up for vehicle payments.

Vehicle Trade in Value



Brief Description: Both sales manager and employee can set a customer up for a vehicle trade-in value for customers that want to trade in their vehicles.

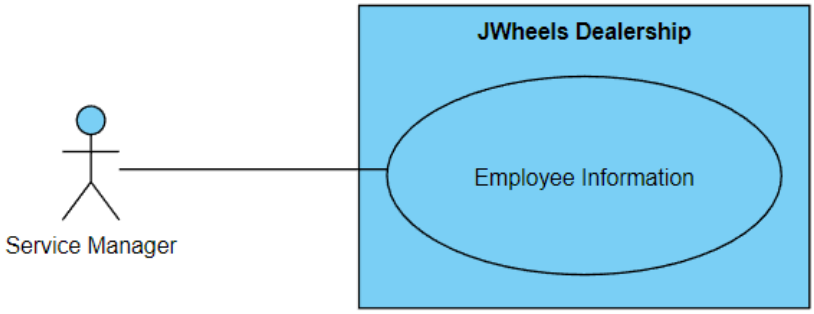
Vehicle Purchase Process



Brief Description: Either the employee or manager who is assisting a customer in making vehicle purchases must search for the selected customer vehicle. Customer information will also contain credit information to see how much annual percentage rate (APR) a customer will pay. The "Vehicle Trade in Value" is optional, if a customer does not want a vehicle to be traded in, then it could just be placed as zero. Vehicle payments will contain information about customers payments for their chosen purchased vehicle. The dealership branch and date of purchase will be included in the invoice as proof.

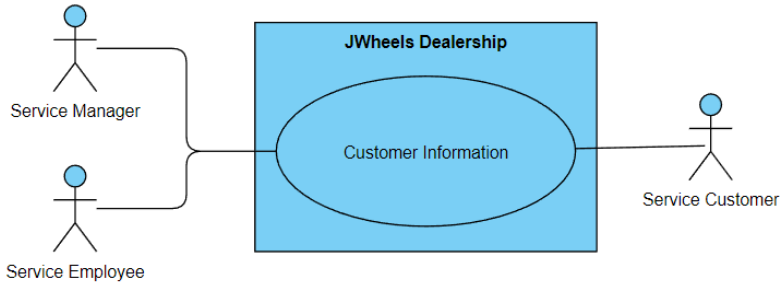
Service Department Use Cases

Employee Information



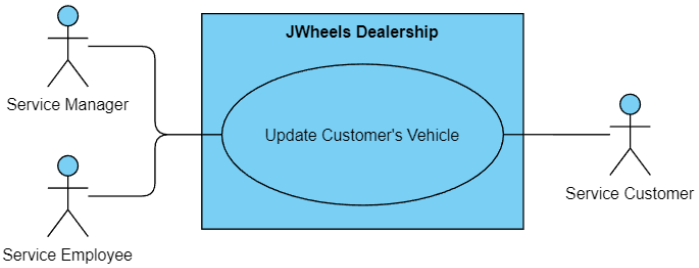
Brief Description: Service Managers will have the privilege of adding, editing, or deleting an employee's information.

Customer Information



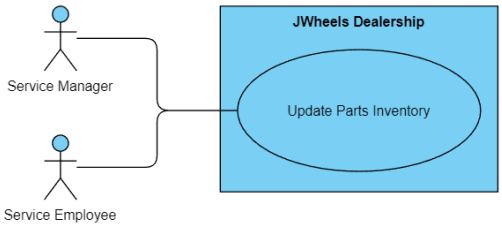
Brief Description: Both service manager and employee will have the privileges in adding, editing, or deleting customer data.

Update Customer’s Vehicle



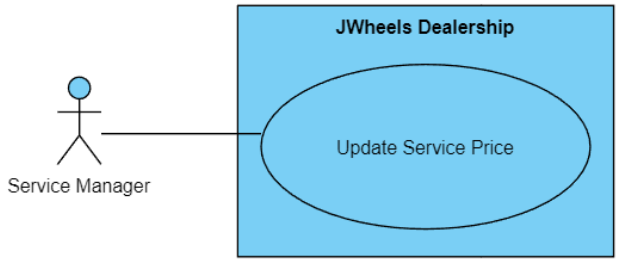
Brief Description: Both service manager and employee will have the privileges in adding, editing, or deleting a customer’s vehicle and its information such as date arrived, VIN, make, model, year, and vehicle descriptions for either service or vehicle issue.

Update Parts Inventory



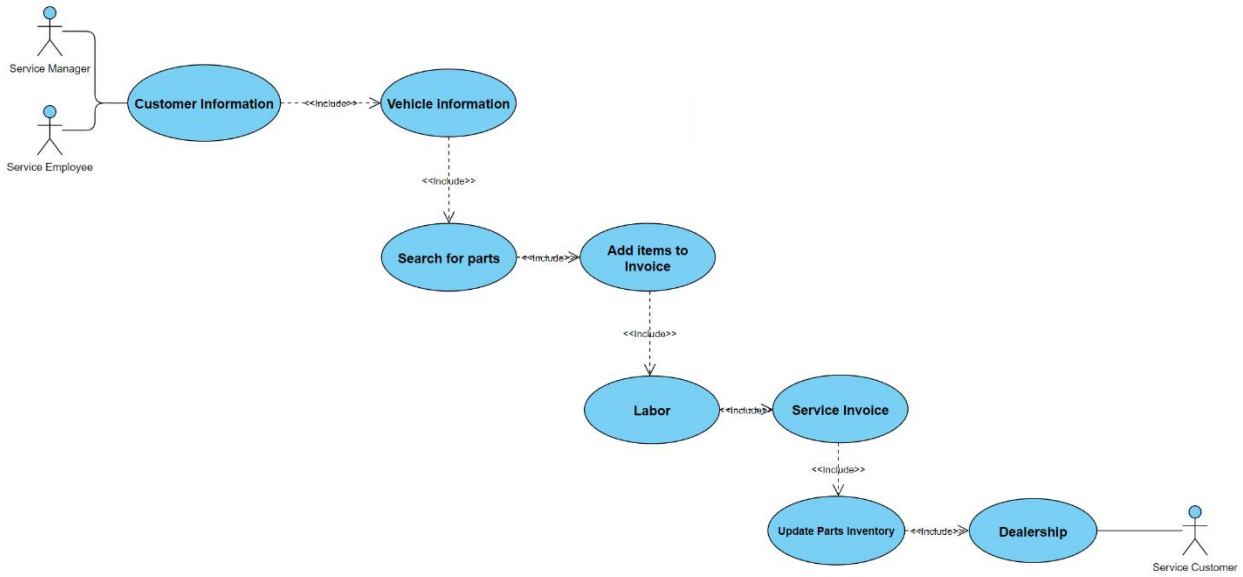
Brief Description: Both service employees and managers can add, edit, or delete parts inventory. There will be data that also includes the number of parts in stock.

Update Service Price



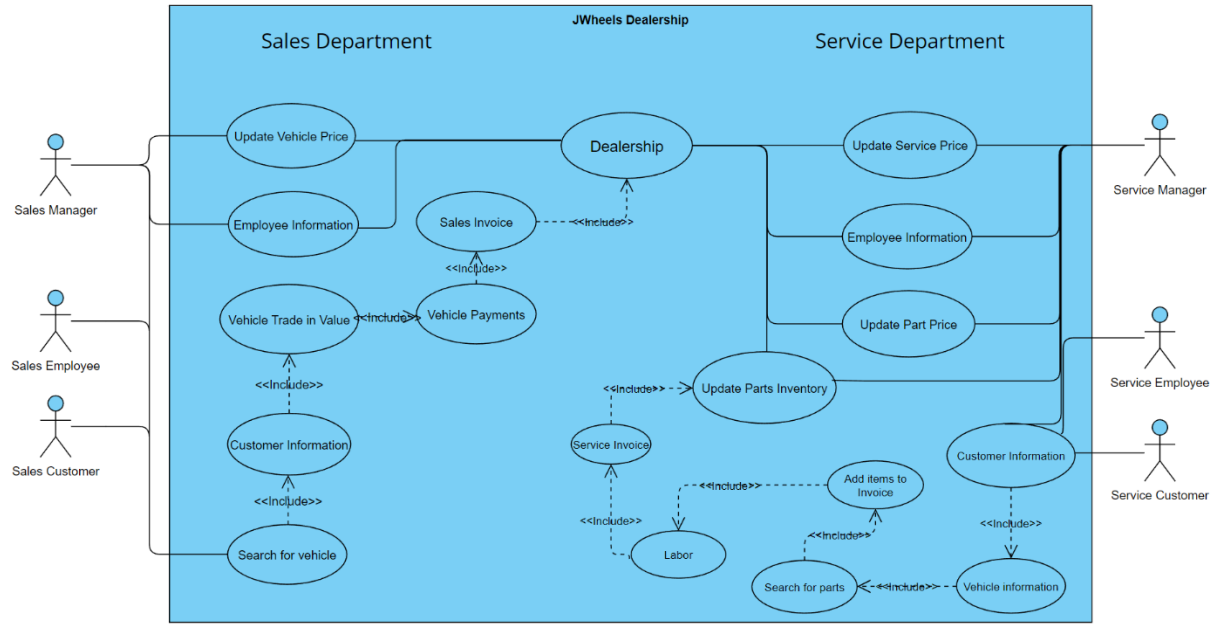
Brief Description: Service Manager has the privilege to add, edit, or delete service price.

Vehicle Servicing process

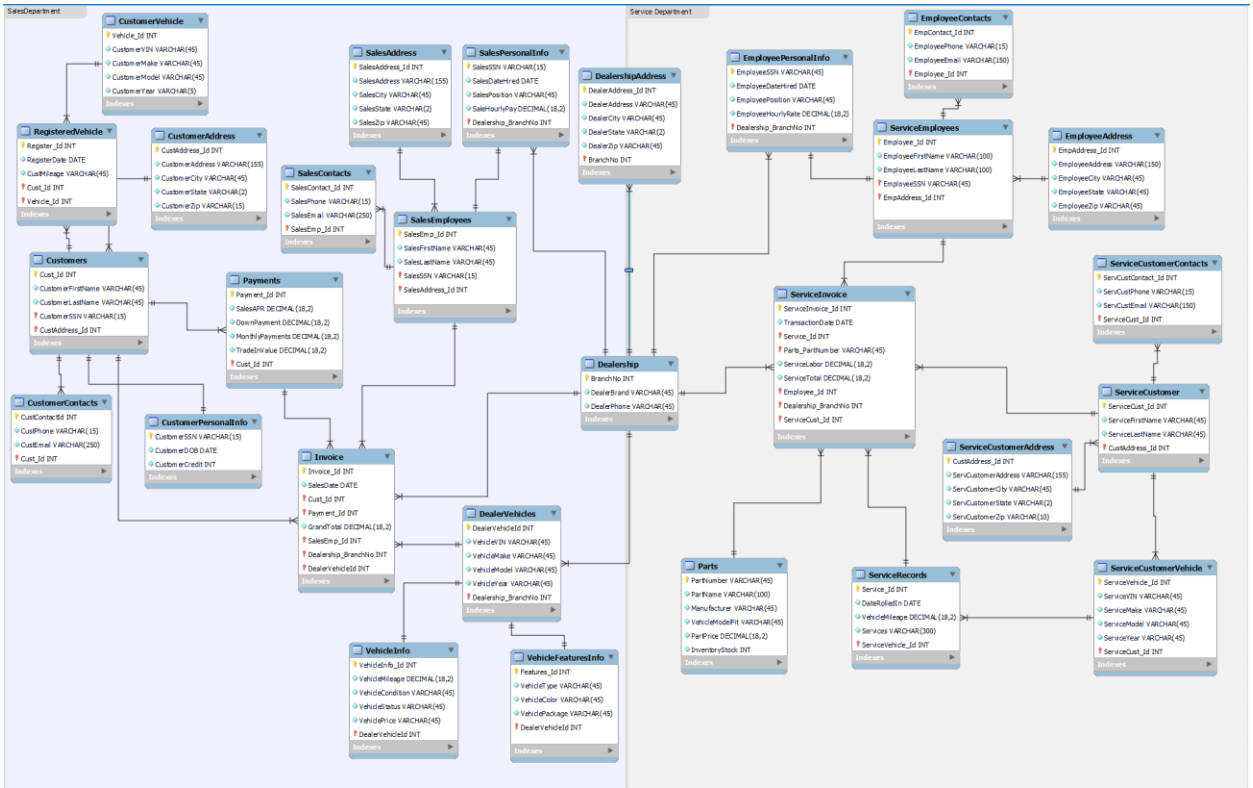


Brief Description: Either a service manager or an employee can assist a customer. When a customer rolls their vehicle in the shop, employees will search for customers information. Then they will get customer's vehicle info. When a technician sees that a part needs to be replaced, an employee or manager will search for parts. Then the parts selected will be added to a bill along with technician labor and print out an invoice along with the dealership branch number for the customer. An employee or manager can update parts inventory for every purchases. The database will add customer's service history invoice.

Full Use Case Diagram:



E/R diagram:



Schemas

Example: RELATION(Primary key → Attribute, **Foreign key**)

- **Dealership**(BranchNo → DealerBrand, DealerPhone)
- **DealershipAddress** (DealerAddress_Id → DealerAddress, DealerCity, DealerState, DealerZip, **BranchNo**)

Sales Department

- **Customers**(Cust_Id→ CustomerFirstName, CustomerLastName, **CustomerSSN, CustAddress_Id**)
- **CustomerPersonalInfo**(CustomerSSN→ CustomerDOB, CustomerCredit)
- **CustomerAddress**(CustAddress_Id → CustomerAddress, CustomerCity, CustomerState, CustomerZip)
- **CustomerContacts**(CustContactId → CustPhone, CustEmail, **Cust_Id**)
- **RegisteredVehicle**(Register_Id → RegisterDate, CustMileage, **Cust_Id**, **Vehicle_Id**)
- **CustomerVehicle** (Vehicle_Id → CustomerVIN, CustomerMake, CustomerModel, CustomerYear)
- **SalesEmployees**(SalesEmp_Id → SalesFirstName, SalesLastName, **SalesSSN**, **SalesAddress_Id**)
- **SalesPersonalInfo**(SalesSSN → SalesDateHired, SalesPosition, SaleHourlyPay, **Dealership_BranchNo**)
- **SalesAddress**(SalesAddress_Id → SalesAddress, SalesCity, SalesState, SalesZip)
- **SalesContacts**(SalesContact_Id → SalesPhone, SalesEmail, **SalesEmp_Id**)
- **DealerVehicles**(DealerVehicleId → VehicleVIN, VehicleMake, VehicleModel, VehicleYear, **Dealership_BranchNo**)
- **VehicleInfo** (VehicleInfo_Id→ VehicleMileage, VehicleCondition, VehicleStatus, VehiclePrice, **DealerVehicleId**)
- **VehicleFeaturesInfo**(Features_Id → VehicleType, VehicleColor, VehiclePackage, **DealerVehicleId**)

- **Payments**(Payment_Id → SalesAPR, DownPayment, MonthlyPayments, TradeInValue, **Cust_Id**)
- **Invoice**(Invoice_Id → SalesDate, **Cust_Id**, **Payment_Id**, GrandTotal, **SalesEmp_Id**, **Dealership_BranchNo**, **DealerVehicleId**)

Service Department

- **ServiceCustomer**(ServiceCust_Id → ServiceFirstName, ServiceLastName, **CustAddress_Id**)
- **ServiceCustomerContacts**(ServCustContact_Id → ServCustPhone, ServCustEmail, **ServiceCust_Id**)
- **ServiceCustomerVehicle**(ServiceVehicle_Id→ ServiceVIN, ServiceMake, ServiceModel, ServiceYear, **ServiceCust_Id**)
- **ServiceCustomerAddress**(CustAddress_Id → ServCustomerAddress, ServCustomerCity, ServCustomerState, ServCustomerZip)
- **ServiceEmployees**(Employee_Id → EmployeeFirstName, EmployeeLastName, **EmployeeSSN**, **EmpAddress_Id**)
- **EmployeePersonalInfo**(EmployeeSSN → EmployeeDateHired, EmployeePosition, EmployeeHourlyRate, **Dealership_BranchNo**)
- **EmployeeContacts**(EmpContact_Id → EmployeePhone, EmployeeEmail, **Employee_Id**)
- **EmployeeAddress**(EmpAddress_Id → EmployeeAddress, EmployeeCity, EmployeeState, EmployeeZip)
- **ServiceRecords**(Service_Id→ DateRolledIn, VehicleMileage, Services, **ServiceVehicle_Id**)
- **Parts**(PartNumber → PartName, Manufacturer, VehicleModelFit, PartPrice, InventoryStock)
- **ServiceInvoice**(ServiceInvoice_Id → TransactionDate, **Service_Id**, **Parts_PartNumber**, ServiceLabor, ServiceTotal, **Employee_Id**, **Dealership_BranchNo**, **ServiceCust_Id**)

Use case Implementation (SQL Statements)

Dealership Entity: Insert, Delete, Update commands

Insert command for “Dealership” table

```
select * from dealership;
insert into dealership
(BranchNo, DealerBrand, DealerPhone)
value
('787', 'Hyundai', '(622)-624-6666');
```

Delete command for “Dealership” table

```
select * from dealership;
delete from dealership
where BranchNo = 787;
```

Update Command for “Dealership” table

```
select * from dealership;
update dealership
set DealerBrand= 'TOYOTA'
where BranchNo = 126;
```

Sales Department: Insert, Update, Delete commands

Customer Entity: Insert, Delete, Update commands

Insert command for “Customers” table

```
select * from customers; insert into customers
(Cust_Id, CustomerFirstName, CustomerLastName, CustomerSSN, CustAddress_Id)
value ('20', 'Marcus', 'Bradley', '20XXXXXXX', '20');
```

Delete command for “Customers” table

```
select * from customers;
delete from customers
where Cust_id = 20;
```

Update command for “Customers” table

```
select * from customers;
update customers
set CustomerFirstName = 'Jamie'
where Cust_Id = 19;
```

Sales Employees Entity: insert, update, and delete commands

Insert command for “SalesEmployees” table

```
select * from salesemployees;
insert into salesemployees
(SalesEmp_Id, SalesFirstName, SalesLastName, SalesSSN, SalesAddress_Id)
value ('25', 'Kalil', 'Brown', '56XXXXXXX', '25')
```

Delete command for “SalesEmployees” table

```
select * from salesemployees;
delete from salesemployees
where SalesEmp_Id = 25;
```

Update command for “SalesEmployees” table

```
select * from salesemployees;
update salesemployees set SalesFirstName = 'Lu', SalesLastName = 'Xiao' where SalesEmp_Id = 13;
```

Sales Entity: insert, update, and delete commands

Insert command for “Invoice” table

```
select * from invoice;
insert into invoice
(Invoice_Id, SalesDate, Cust_Id, Payment_Id, GrandTotal, SalesEmp_Id, Dealership_BranchNo,
DealerVehicleId)
value ('20', '2022-03-15', '20', '20', '7538', '24', '847', '59');
```

Delete command for “Invoice” table

```
select * from invoice;
delete from invoice
where Invoice_Id = 20;
```

Update command for “Invoice” table

```
select * from invoice;
update invoice
set SalesDate = '2022-04-29'
where Invoice_Id = 19;
```

Dealership Vehicle Entity: insert, update, and delete commands

Insert command for “DealerVehicles” table

```
select * from dealervehicles;
insert into dealervehicles
(DealerVehicleId, VehicleVIN, VehicleMake, VehicleModel, VehicleYear)
value ('61', '2FMPK4J96NBA89376', 'Ford', 'Edge', '2022');
```

Delete command for “DealerVehicles” table

```
select * from dealervehicles;
delete from dealervehicles
where DealerVehicleId = 61;
```

Update command for “DealerVehicles” table

```
select * from dealervehicles;
update dealervehicles
set VehicleYear = '2021'    where DealerVehicleId = 60;
```

Service Department: insert, update, and delete commands

Service Customer Entity: insert, update, and delete commands

Insert command for “ServiceCustomer” table

```
select * from servicecustomer;
insert into servicecustomer
(ServiceCust_Id, ServiceFirstName, ServiceLastName, CustAddress_Id)
values ('19', 'Yesenia', 'Campos', '19');
```

Delete command for “ServiceCustomer” table

```
select * from ServiceCustomer;
delete from ServiceCustomer
where SalesEmp_Id = 25;
```

Update command for “ServiceCustomer” table

```
select * from servicecustomer;
update servicecustomer
set ServiceLastName = 'Vasquez'
where ServiceCust_Id = 11;
```

Service Employees Entity: insert, update, and delete commands

Insert command for “ServiceEmployees” table

```
select * from serviceemployees;
insert into serviceemployees
(Employee_Id, EmployeeFirstName, EmployeeLastName, EmployeeSSN, EmpAddress_Id)
values ('25', 'Kenya', 'Esparza', '80XXXXXXX', '25')
```

Delete command for “ServiceEmployees” table

```
select * from serviceemployees;
delete from serviceemployees
where Employee_Id = 25;
```

Update command for “ServiceEmployees” table

```
select * from serviceemployees;
update serviceemployees
set EmployeeFirstName = 'Yin', EmployeeLastName = 'Ten' where Employee_Id = 8;
```

Service Invoice Entity: insert, update, and delete commands

Insert command “ServiceInvoice” table

```
select * from serviceinvoice;
insert into serviceinvoice
(ServiceInvoice_Id, TransactionDate, Service_Id, Parts_PartNumber, ServiceLabor, ServiceTotal,
Employee_Id, Dealership_BranchNo, ServiceCust_Id)
value ('20', '2022-12-02', '20','N/A', '25.50', '25.50', '25', '515', '19');
```

Delete command “ServiceInvoice” table

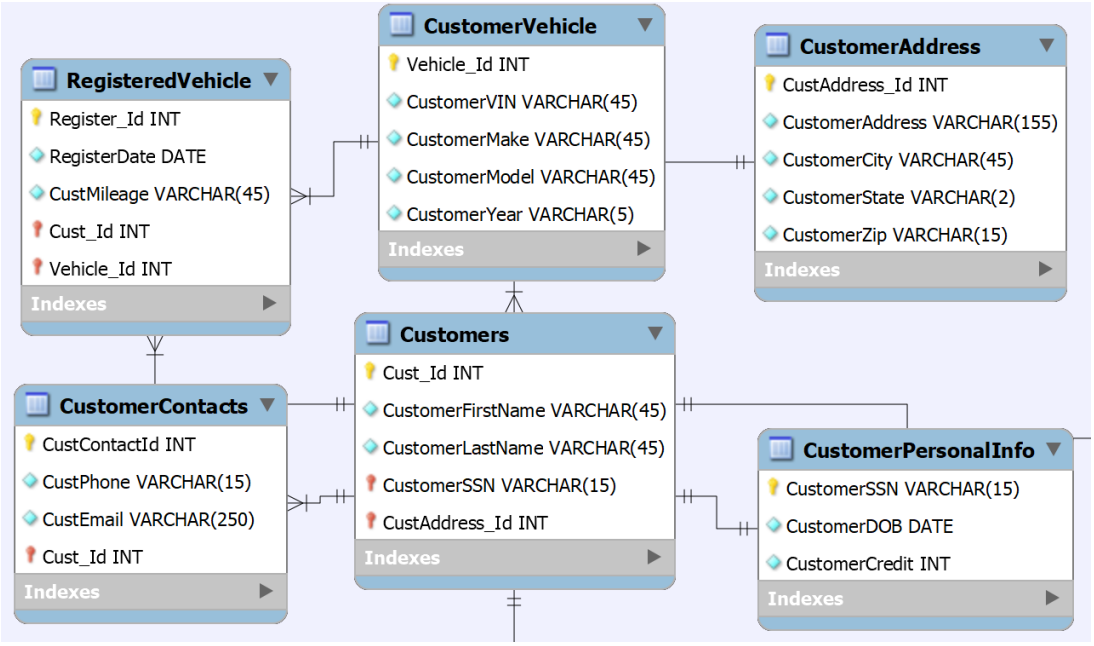
```
select * from serviceemployees;
delete from serviceemployees
where Employee_Id = 20;
```

Update command “ServiceInvoice” table

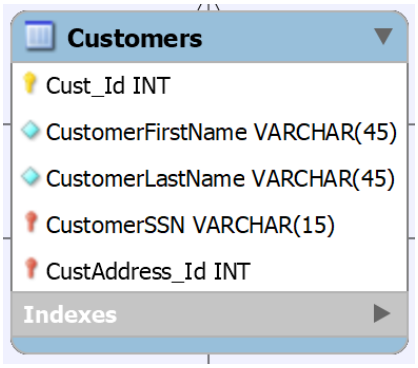
```
select * from serviceinvoice;
update serviceinvoice
set TransactionDate = '2022-11-12'
where ServiceInvoice_Id = 19;
```

Normalization: Sales Department

Customer entity data



The “Customers” entity data contains six tables are dedicated to store customer information. The entity includes “Customers”, “CustomerPersonalInfo”, “CustomerAddress”, “CustomerContacts”, “RegisteredVehicle”, and “CustomerVehicle” tables.

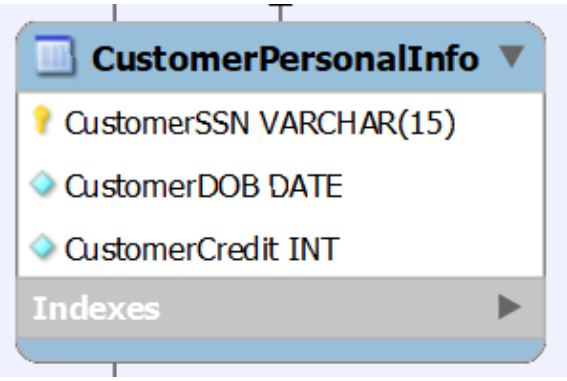


The “Customers” table contains basic customer information. The table contains a primary key that holds customer’s ID number (Cust_Id), customer’s first name (CustomerFirstName), customer’s last name (CustomerLastName), and two foreign keys that store customers social security number (CustomerSSN) and customer address (CustAddress_Id). The data is does not

contain any partial or transitive dependencies. The foreign keys only refer a primary key from another table, which in this case the “CustomerSSN” foreign key refers the “CustomerPersonalInfo” table, and “CustAddress” foreign key refers the “CustomerAddress” table. The “Customers” table is in a one-to-one relationship with “CustomerPersonalInfo” table because a customer can only have one personal information. The “Customers” table is in a many-to-one relationship with “CustomerAddress” table because many customers can live in an address. The “Customers” table is in a one-to-many relationship with “RegisterVehicle” table because a customer can have multiple registered vehicles. The “Customers” table is in a one-to-many relationship with “Payments” table because a customer can have multiple payments. The attributes are functionally dependent as there are no non-prime (non-key) attributes in the table.

- Cust_Id → CustomerFirstName, CustomerLastName, CustomerSSN, CustAddress

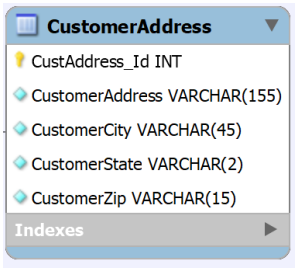
Therefore, “Customers” table is set in third normal form.



The “CustomerPersonalInfo” table contains a customer’s personal information. As stated before, “CustomerPersonalInfo” table is in a one-to-one relationship with “Customers” table. This table is in third normal form because there are no transitive and partial dependencies in the table. The table does have the proper functional dependencies that depend on a customer’s personal information such as the primary key that holds customer’s social security number (CustomerSSN) , customers date of birth (CustomerDOB), and customer’s credit score (CustomerCredit). In short:

- CustomerSSN → CustomerDOB, CustomerCredit

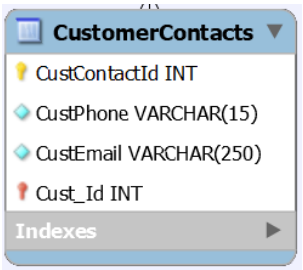
The table does not contain any non-prime attributes and transitive dependencies. Therefore, “CustomerPersonalInfo” table is in third normal form.



The “CustomerAddress” table contains a customer’s address information. As shown in the table above, there are five attributes that are functionally dependent of each other. The table includes a primary key that holds a customer address number ID (CustAddress_Id), customer’s address (CustomerAddress), customer’s city (CustomerCity), customer’s state (CustomerState), and customer’s zip code (CustomerZip). The “CustomerAddress” table is in a one-to-many relationship with “Customers” table since many an address could belong to multiple customers. The “CustomerAddress” table does not contain any non-prime attributes or transitive dependencies as every attribute is functionally dependent:

- CustAddress_Id → CustomerAddress, CustomerCity, CustomerState, CustomerZip

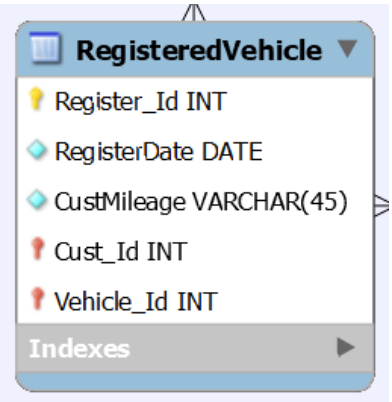
Therefore, the “CustomerAddress” table is in third normal form.



The “CustomerContacts” table contains customer’s contact information. The table contains three attributes and one foreign key that refers to “Customers” table. The table includes a primary key that holds the customer’s contact ID number (CustContactId), customer’s phone number (CustPhone), customer’s email (CustEmail), and the foreign key that holds the customer’s ID number (Cust_Id). The “CustomerContacts” table is in a many-to-one relationship with “Customers” table because a customer can have multiple contacts or use the same contact information. The attributes in the “CustomerContacts” table are functionally dependent and no non-prime attributes exist in the table. Also, the table does not contain any transitive dependencies.

- CustContactId → CustPhone, CustEmail, Cust_Id

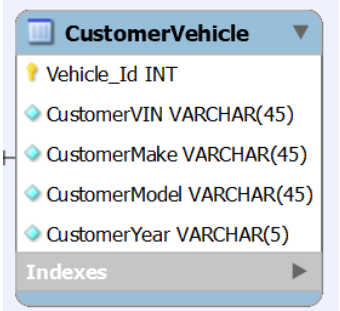
Therefore, “CustomerContacts” table is in third normal form.



The “RegisteredVehicle” table is also part of “Customers” entity because it contains information for customers that want to trade in their vehicles for a new one. The table contains three attributes and two foreign keys. The table includes a primary key that holds the registration ID number (Register_Id), date of registration (RegisterDate), the vehicle mileage (CustMileage), and two foreign keys that are known as “Cust_Id” and “Vehicle_Id”. The foreign key “Cust_Id” refers to the “Customers” table and “Vehicle_Id” refers to the “CustomerVehicle” table (which will be discussed next). The “RegisteredVehicle” table is in a many-to-one relationship with “Customers” table because a customer can have many registered vehicles. Also, the “RegisteredVehicle” table is in a many-to-one relationship with “CustomerVehicle” table because many registrations can contain a vehicle information. The attributes in the “RegisteredVehicle” table are functionally dependent since the table does not contain any non-prime attributes or transitive dependencies:

- Register_Id → RegisterDate, CustMileage, Cust_Id, Vehicle_Id

Note: CustMileage is functionally dependent with the attributes in this table because a vehicle will never have the same mileage from the day that it was registered. Hence, it is important that mileage is recorded properly based on date because it plays a determination of a vehicle’s value when it is traded in. Therefore, the “RegisteredVehicle” table is in third normal form.

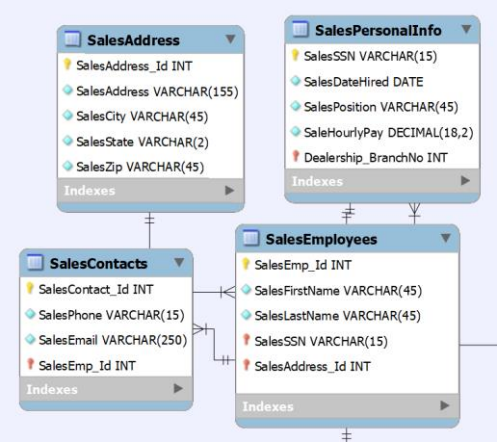


The “CustomerVehicle” table contains a customer’s vehicle information. There are five attributes that depend on each other. The table includes a primary key that holds the vehicle number ID (Vehicle_ID), customer’s vehicle identification number (CustomerVIN), customer’s make (CustomerMake), customer’s model (CustomerModel), and customer’s year (CustomerYear). As stated before, the “CustomerVehicle” table is in a one-to-many relationship with “RegisteredVehicle” table because a vehicle can have multiple registrations. The attributes in the “CustomerVehicle” table are functionally dependent since the table does not contain any non-prime attributes or transitive dependencies:

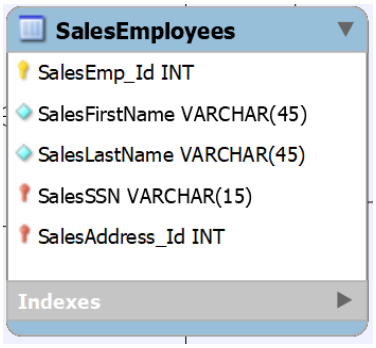
- Vehicle_Id → CustomerVIN, CustomerMake, CustomerModel, CustomerYear

Therefore, “CustomerVehicle” table in in third normal form.

Sales Employees entity



The “Sales Employees” entity contains four dedicated tables that store sales employees information. The entity includes “SalesEmployees”, “SalesContacts”, “SalesAddress”, and “SalesPersonalInfo” tables. The “Sales Entity” tables are in third normal form since none of the tables any partial or transitive dependencies. Here are the reasons why:

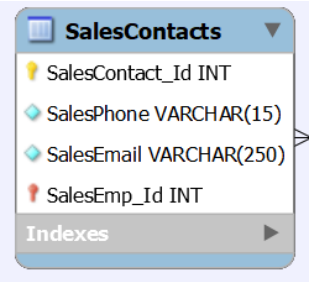


The “SalesEmployees” table contains the names of employees. As shown above, the table contains three attributes and two foreign keys. The table contains a primary key that holds sales employee’s ID number (SalesEmp_Id), sales employee’s first name (SalesFirstName), sales employee’s last name (SalesLastName), and two foreign keys that includes sales employees’ social security number (SalesSSN), and sales employees addresses ID number (SalesAddress_Id). The “SalesSSN” foreign key refers the “SalesPersonalInfo” table, and “SalesAddress_Id” refers the “SalesAddress” table. The “SalesEmployees” is in a many-to-one relationship with “SalesAddress” table because many employees can have an address. The “SalesEmployees” is in a one-to-one relationship with “SalesContacts” table because an

employee has one personal information. The “SalesEmployees” is in a one-to-many relationship with “Invoice” table because an employee can be included in multiple invoices. All these attributes are functionally dependent as there are no partial dependencies found in the table:

- SalesEmp_Id → SalesFirstName, SalesLastName, SalesSSN, SalesAddress

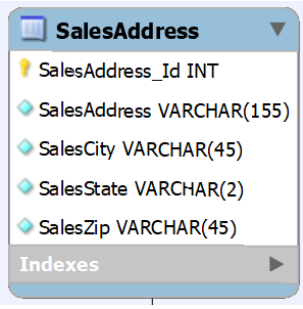
Therefore, “SalesEmployees” table is in third normal form.



The “SalesContacts” table contains information about a sales employee’s contact information. The table contains a sales employee contact ID number (SalesContact_Id) as a primary key, sales employee’s phone number (SalesPhone), sales employee’s email (SalesEmail), and a foreign key known as “SalesEmp_Id” that refers the “SalesEmployees” table. In addition, the “SalesContacts” table is in a many-to-one relationship with “SalesEmployees” table because many contacts information can belong to a sales employee. The attributes in the “SalesContacts” table are functionally dependent as there are no partial or transitive dependencies shown in the table:

- SalesContact_Id → SalesPhone, SalesEmail, SalesEmp_Id

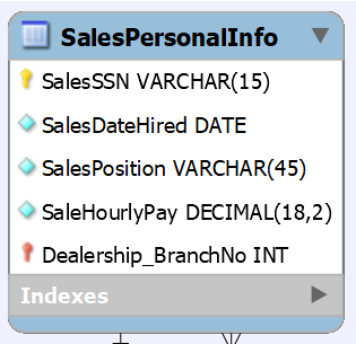
Therefore, the “SalesContacts” table is in third normal form.



The “SalesAddress” table’s purpose is to contain sales employees addresses. The table contains a primary key that is a salesperson address number ID (SalesAddress_Id), salesperson address (SalesAddress), salesperson city (SalesCity), salesperson state (SalesState), salesperson zip code (SalesZip), and a foreign key that displays salesperson ID number (SalesEmp_ID). The foreign key “SalesEmp_Id” refers to “SalesEmployees” table. The “SalesAddress” table is in a one-to-many relationship with “SalesEmployees” table because an employee could have multiple addresses or live in a same address. The attributes in the “SalesAddress” table are functionally dependent since the table does not contain any non-prime attributes or transitive dependencies:

- SalesAddress_Id → SalesAddress, SalesCity, SalesState, SalesZip

Therefore, “SalesAddress” table is in third normal form.



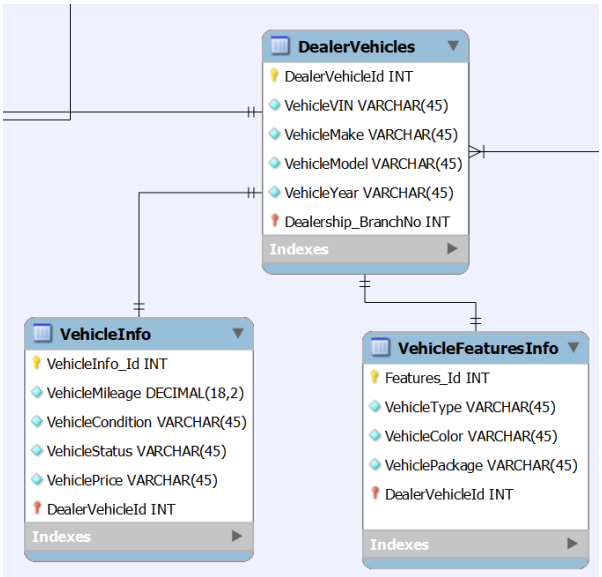
The “SalesPersonalInfo” is a table that contains a salesperson’s personal information table. The table includes four attributes and a foreign key. The table include a primary key that holds a sales employees’s social security number (SalesSSN), an employee’s hiring date (SalesDateHired), sales employee position (SalesPosition), sales employees’s hourly pay (SaleHourlyPay), and a foreign key that stores dealership branch number (Dealership_BranchNo). The “Dealership_BranchNo” foreign key refers the “Dealership” table. The “SalesPersonalInfo” table

is in a one-to-one relationship with “SalesEmployees” table because a sales employee’s personal information belongs to one sales employee. The “SalesPersonalInfo” table is in a many-to-one relationship with “Dealership” table (which will be discussed towards the end) because many sales employee’s personal information belongs to a dealership. The attributes in the “SalesPersonalInfo” table are functionally dependent since the table does not contain any non-prime attributes or transitive dependencies:

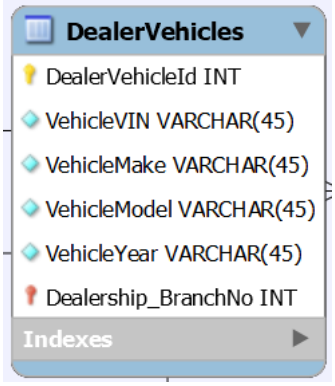
- SalesSSN → SalesDateHired, SalesPosition, SaleHourlyPay, Dealership_BranchNo

Therefore, the “SalesPersonalInfo” table is in third normal form.

Dealership Vehicle entity



The “Dealership Vehicle” entity contains three tables that are dedicated for the vehicles that are up to sale in a dealership. The entity includes “DealerVehicles”, “VehicleInfo”, and “VehicleFeaturesInfo” tables as shown above.

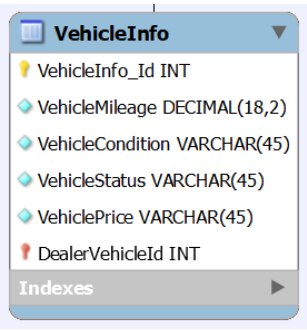


The “DealerVehicles” table contains five attributes and one foreign key that refers the “Dealership” table. The “DealerVehicles” table contains a vehicle ID number (DealerVehicleId) as a primary key, vehicle VIN number (VehicleVIN), vehicle manufacturer (VehicleMake), vehicle model (VehicleModel), vehicle year (VehicleYear), and a foreign key that contains the dealership’s branch number (Dealership_BranchNo). The “DealerVehicles” table is in a many-to-one relationship with the “Dealership” table because many dealer vehicles belong in a dealership. The “DealerVehicles” table is in a one-to-many relationship with the “Invoice” table because a dealer vehicle can be placed in many invoices. The “DealerVehicles” table is in a one-

to-one relationship with “VehicleInfo” table because a dealer vehicle contains an information. The “DealerVehicles” table is in a one-to-one relationship because each vehicle contains a feature. The table has functional dependencies since there are no partial or transitive dependencies in the table:

- DealerVehicleId → VehicleVIN, VehicleMake, VehicleModel, VehicleYear, Dealership_BranchNo

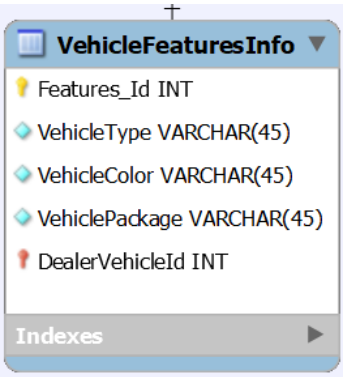
Therefore, the “DealerVehicles” table is in third normal form.



The “VehicleInfo” table contains information about the vehicle that is on sale. The table contains five attributes and one foreign key that refers the “DealerVehicles” table. The “VehicleInfo” table contains a primary key that holds the vehicle information ID number (VehicleInfo_Id), vehicle mileage (VehicleMileage), vehicle condition (VehicleCondition), vehicle status (VehicleStatus), vehicle price (VehiclePrice), and a foreign key known as “DealerVehicleId” that comes from “DealerVehicles” table. The “VehicleInfo” table in a one-to-one relationship with “DealerVehicles” table because a vehicle from the dealer can only have one information. The attributes in the “VehicleInfo” tables are functional dependent on each other. Also, there are no partial or transitive dependencies in this table:

- VehicleInfo_Id → VehicleMileage, VehicleCondition, VehicleStatus, VehiclePrice, DealerVehicleId

Therefore, “VehicleInfo” table is in third normal form.

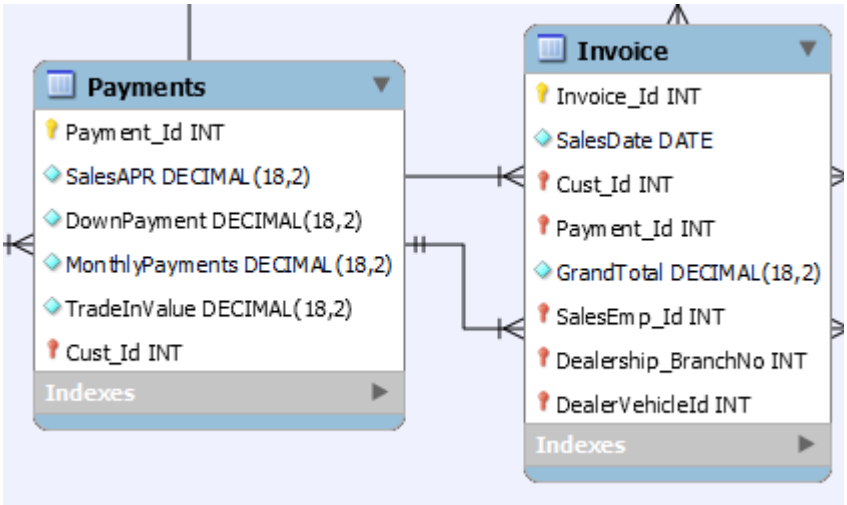


The “VehicleFeaturesInfo” table contains information about a vehicle’s features. The table contains four attributes and one foreign key from “DealerVehicles” table. The table includes primary key known as feature number ID (Features_Id), the type of vehicle (VehicleType), the color of a vehicle (VehicleColor), the name of the type of package a vehicle (VehiclePackage), and a foreign key known as the vehicle number ID (DealerVehicleId). The “VehicleFeaturesInfo” table is in a one-to-one relationship with “DealerVehicles” table because a vehicle feature information can only be applied to a vehicle. Partial and transitive dependencies do not exist in this table:

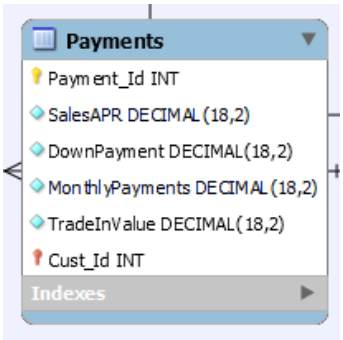
- Features_Id → VehicleType, VehicleColor, VehiclePackage, DealerVehicleId

Therefore, the table is in third normal form.

Sales entity



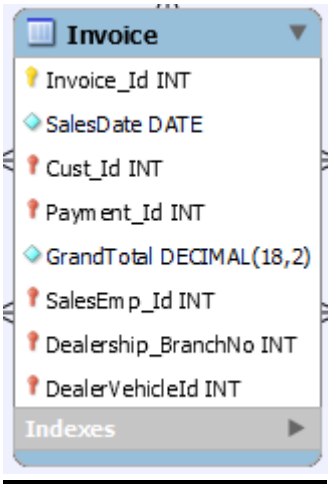
The sales entity contains two tables that are dedicated to sales data from each vehicle being sold. The names of the two tables are “Payments” and “Invoice”.



The “Payments” table contains five attributes and one foreign key. The “Payments” table includes primary key known as the payment ID number (Payment_Id), the annual percentage rate (SalesAPR), a down payment (DownPayment), monthly payments (MontlyPayments), a trade in value (TradeInValue), and a foreign key containing customer number ID (Cust_Id) that comes from “Customers” table. The “Payments” table is in a many-to-one relationship with the “Customers” table because many payments can come from a customer if they are purchasing multiple vehicles. The attributes functionally depend on each other as the “Payments” table does not contain any non-prime attributes or transitive dependencies:

- Payment_Id → SalesAPR, DownPayment, MontlyPayments, TradeInValue, Cust_Id

Therefore, the table is in third normal form.



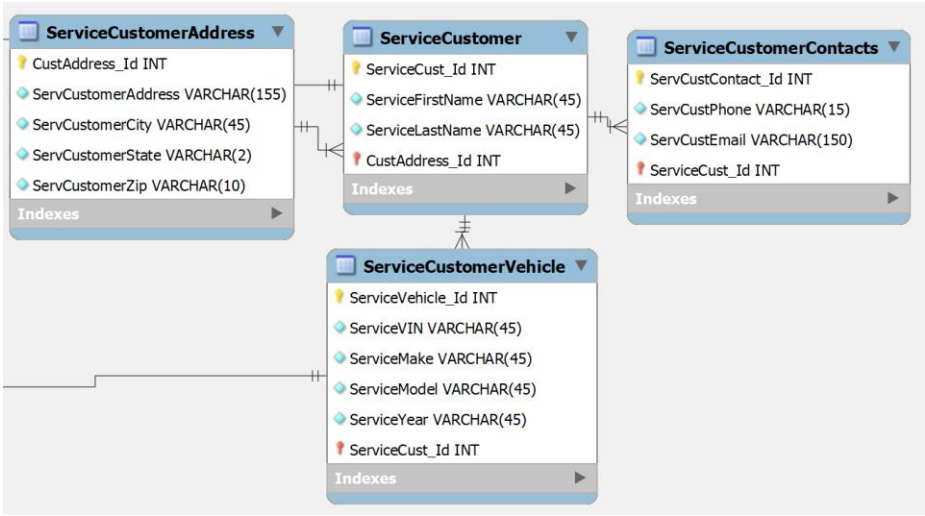
The “Invoice” table contains three attributes and five foreign keys. The “Invoice” table includes a primary key known as invoice ID number (Invoice_Id), the date of sale (SalesDate), the grand total (GrandTotal), and five foreign keys known as the “Cust_Id” that refers the “Customers” table, “Payment_Id” that refers the “Payments” table, “SalesEmp_Id” that refers the “SalesEmployees” table, “Dealership_BranchNo” that refers the “Dealership” table, and “DealerVehicleId” that refers the “DealerVehicles” table. The “Invoice” table is in a many-to-one relationship with “Customers” table because many invoices can contain a customer. The “Invoice” table is in a many-to-one relationship with “Payments” table because many invoices can contain a payment information. The “Invoice” table is in a many-to-one relationship with “SalesEmployees” table because many invoices can include an employee’s information for contributing to making sales. The “Invoice” table is in a many-to-one relationship with “DealerVehicles” table because many invoices can contain a vehicle. The “Invoice” table is in a many-to-one relationship with “Dealership” table because many invoices can contain a dealership’s information. The attributes in the “Invoice” table functionally depend on each other since the table does not contain any non-prime attributes or transitive dependencies:

- Invoice_Id → SalesDate, Cust_Id, Payment_Id, GrandTotal, SalesEmp_Id, Dealership_BranchNo, DealerVehicleId

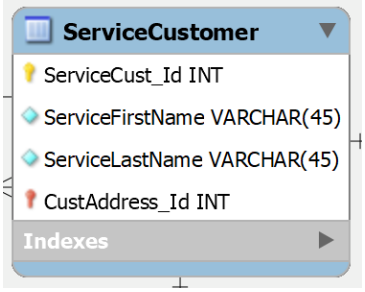
Therefore, the table is in third normal form.

Normalization Service Department

Service Customer Entity



The “Service Customer” entity contains four dedicated tables that contain information about a customer in service department. The names of the tables that are utilized in this entity are “ServiceCustomer”, “ServiceCustomerContacts”, “ServiceCustomerVehicle”, and “ServiceCustomerAddress”.

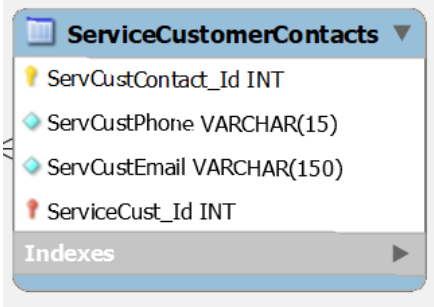


The “ServiceCustomer” table contains three attributes that are based on customer’s name. The “ServiceCustomer” table contains a primary that holds the service customer ID number (ServiceCust_Id), service customer’s first name (ServiceFirstName), and service customer’s last name (ServiceLastName). The “ServiceCustomer” table also contains a foreign key that stores customer’s address id (CustAddress_Id) and it refers the “ServiceCustomerAddress” table. The “ServiceCustomer” table is in a one-to-many relationship with “ServiceCustomerContacts” and “ServiceCustomerVehicle” tables. The reason why “ServiceCustomer” table is in a one-to-many

relationship with “ServiceCustomerContacts” table is because a customer can have multiple contacts. The reason why “ServiceCustomer” table is in a one-to-many relationship with “ServiceCustomerVehicle” table is because a customer in service can have many vehicles for service. The reason why “ServiceCustomer” table is in a one-to-many relationship with “ServiceCustomerAddress” is because a customer could in a same address. The attributes in the “ServiceCustomer” table functionally depend on each other since the table does not contain any non-prime attributes or transitive dependencies:

- ServiceCust_Id → ServiceFirstName, ServiceLastName, CustAddress_Id

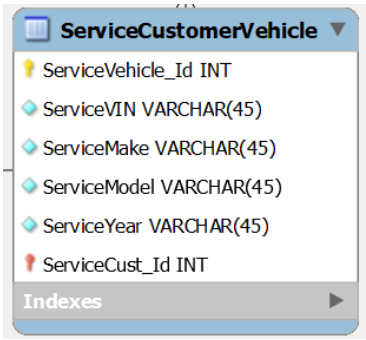
Therefore, the table is in third normal form.



The “ServiceCustomerContacts” table contains the address information from a customer in the service department. The table contains three attributes and a foreign key that refers the “ServiceCustomer” table. The table contains a primary key that holds the service customer contact ID number (ServCustContact_Id), service customer’s phone number (ServCustPhone), service customer’s email (ServCustEmail), and the foreign key known as the service customer ID number (ServiceCust_Id). The “ServiceCustomerContacts” table is in a many-to-one relationship with “ServiceCustomer” table because there can be multiple contacts for a customer in the service department. There are no transitive dependencies, partial dependencies, or non-prime attributes found in the table:

- ServCustContact_Id → ServCustPhone, ServCustEmail, ServiceCust_Id

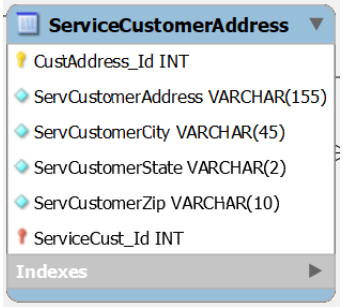
Therefore, the table is in third normal.



The “ServiceCustomerVehicle” table contains a service customer’s vehicle information. The table contains five attributes and a foreign key. The table includes a primary key that holds the customer’s service vehicle ID number (ServiceVehicle_Id), a service vehicle’s VIN number (ServiceVIN), the vehicle’s manufacturer (ServiceMake), the vehicle’s model (ServiceModel), vehicle’s year (ServiceYear), and a foreign key known as “ServiceCust_Id” and it refers “ServiceCustomer” table. The “ServiceCustomerVehicle” table is in a many-to-one relationship with “ServiceCustomer” table because many service vehicles can belong to a customer. The table’s attributes only relate to the service customer’s vehicle information. There are no attributes that are partially or transitively dependent as they are all functionally dependent in the “ServiceCustomerVehicle” table:

- ServiceVehicle_Id → ServiceVIN, ServiceMake, ServiceModel, ServiceYear, ServiceCust_Id

Therefore, the table is in third normal form.

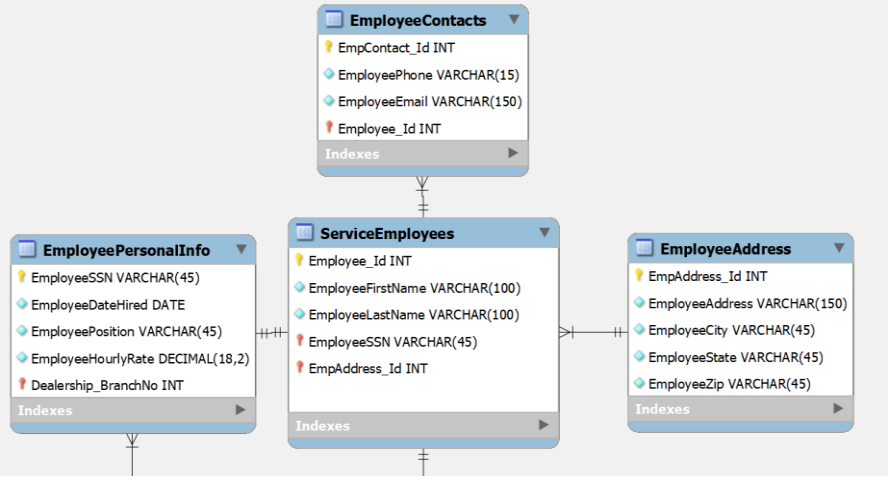


The “ServiceCustomerAddress” table contains information about a customer’s address for the service department. The table contains five attributes that includes a primary key that holds the service customer’s address ID number (CustAddress_Id), service customer’s address (ServCustomerAddress), service customer’s city (ServCustomerCity), service customer’s state (ServCustomerState), service customer’s zip (ServCustomerZip). The “ServiceCustomerAddress” table is in a one-to-many relationship with “ServiceCustomer” table because an address can belong to multiple customers in the service department, or a customer may live in a same address. The attributes are all functionally dependent since the table does not contain any non-prime attributes. In addition, there are no transitive or partial dependencies found in the table:

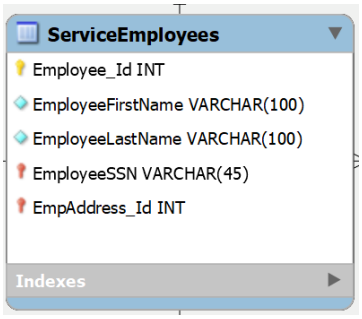
- CustAddress_Id → ServCustomerAddress, ServCustomerCity, ServCustomerState, ServCustomerZip

Therefore, the table is set in third normal form.

Service Employees entity



The “Service Employees” entity contains four tables that are dedicated to store service employee information in the database. The names of the tables that are utilized in this entity are “ServiceEmployees”, “EmployeeContacts”, “EmployeeAddress”, and “EmployeePersonalInfo”.

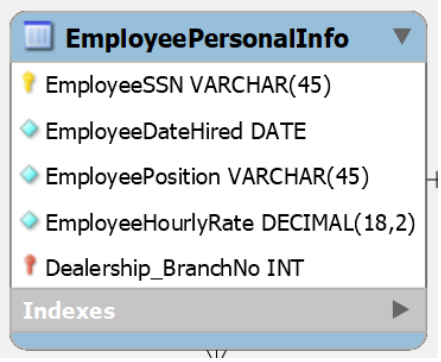


The “ServiceEmployees” table contains the names of the employees in the service department. The table contains three attributes that contain a service employee’s ID number (Employee_Id), service employee’s first name (EmployeeFirstName), and service employee’s last name (EmployeeLastName). The table also contains two foreign keys that store service employee social security number (EmployeeSSN), and service employee address (EmpAddress_Id). The “EmployeeSSN” foreign key refers the “SalesPersonalInfo” table, and “EmpAddress_Id” refers the “EmployeeAddress” table. The “ServiceEmployees” table has a one-to-one relationship with “EmployeePersonalInfo” table since an employee can only have one personal info. The “ServiceEmployees” table is in a many-to-one relationship with “EmployeeAddress” table

because many employees can live in an address. The “ServiceEmployees” table is in a one-to-many relationship with “EmployeeContacts” table because an employee can have multiple contacts. The attributes in the “ServiceEmployees” table does not contain any partial or transitive dependencies and there are no non-prime attributes found in the table:

- Employee_Id → EmployeeFirstName, EmployeeLastName, EmployeeSSN, EmpAddress_Id

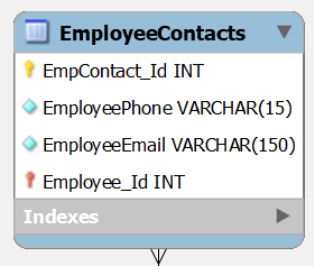
Therefore, the table is in third normal form.



The “EmployeePersonalInfo” contains service employee’s personal information. The table has four attributes and one foreign key. The table contains a primary key that holds the service employee’s social security number (EmployeeSSN), the service employee hiring date (EmployeeDateHired), the employee’s position (EmployeePosition), service employee’s hourly salary (EmployeeHourlyRate), and the foreign key known as “Dealership_BranchNo”. The “Dealership_BranchNo” foreign key refers to the “Dealership” table (comes from the “Dealership” entity and will be the last entity discussed towards the end). The “EmployeePersonalInfo” table is in a one-to-one relationship with “ServiceEmployees” because a personal information belongs to an employee. The “EmployeePersonalInfo” table is in a many-to-one relationship with “Dealership” table because many employees’ personal information belongs to a dealership. Looking at the table, the attributes are functionally dependent as there are no partial or transitive dependencies that exist in the table:

- EmployeeSSN → EmployeeDateHired, EmployeePosition, EmployeeHourlyRate, Dealership_BranchNo

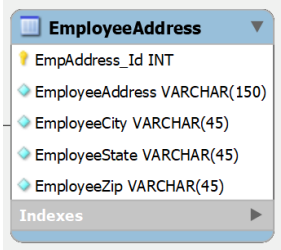
Therefore, the table is in third normal form.



The “EmployeeContacts” table contains the service employees contact information. The table contains three attributes and a foreign key that refers to “ServiceEmployees” table. The table has a primary key that is known to have a service employee’s ID number (EmpContact_Id), service employee’s phone number (EmployeePhone), service employee’s email (EmployeeEmail), and a foreign key known as “Employee_Id”. As stated before, the “EmployeeContacts” table is in a many-to-one relationship with “ServiceEmployees” table because there could be many contacts that come from an employee. The table does not have any partial or transitive dependencies since the attributes are functionally dependent. Also, non-prime attributes do not exist in the table:

- EmpContact_Id → EmployeePhone, EmployeeEmail, Employee_Id

Therefore, the table is in third normal form.

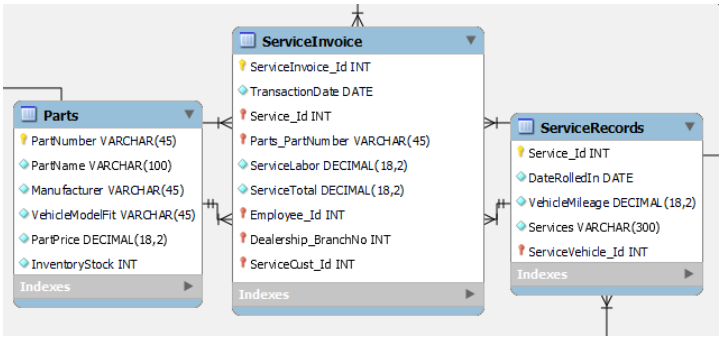


The “EmployeeAddress” table contains information about an employee’s address. The table has five attributes which contains a primary key that holds a service employee address ID number (EmpAddress_Id), service employee address (EmployeeAddress), service employee city (EmployeeCity), service employee state (EmployeeState), and service employee zip code (EmployeeZip). As stated before, the table is in a one-to-many relationship with “ServiceEmployees” table because there could be an address that belongs to many employees. The table does not have any partial or transitive dependencies since the attributes are functionally dependent:

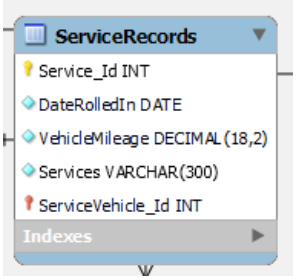
- EmpAddress_Id → EmployeeAddress, EmployeeCity, EmployeeState, EmployeeZip

Therefore, the table is in third normal form.

Service Sales entity



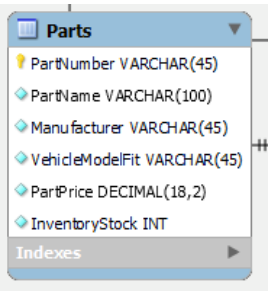
The “Service Sales” entity contains three tables that are dedicated on sales in the service department. The names of the tables that are used in this entity are “Parts”, “ServiceInvoice”, and “ServiceRecords”.



The “ServiceRecords” table contains records of a customer’s vehicle for any issues or requested services. The table contains four attributes and one foreign key that refers to “ServiceCustomerVehicle” table. The table contains a primary that is the service ID number (Service_Id), the date that customer’s vehicle arrived at the dealership (DateRolledIn), customer’s vehicle mileage (VehicleMileage), customer issues or requested services (Services), and the foreign key known as “ServiceVehicle_Id”. The “ServiceRecords” table has a many-to-one relationship with “ServiceCustomerVehicle” because there could be many services done for a vehicle. The “ServiceRecords” table is in a one-to-many relationship with “ServiceInvoice” table because a service record can be registered in many invoices. The “ServiceRecords” table does not contain any partial or transitive dependencies because all attributes are functional, and no non-prime attributes are found in the table:

- Service_Id → DateRolledIn, VehicleMileage, Services, ServiceVehicle_Id

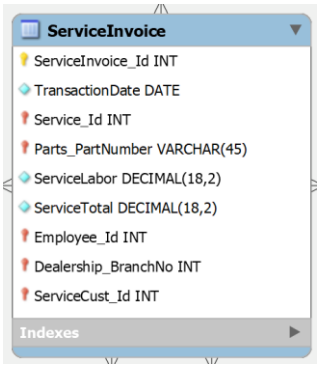
Therefore, the table is in third normal form.



The “Parts” table contains information about vehicle’s parts in the service department. The table contains six attributes. The table contains a primary key that gives a part number ID (PartNumber), part name (PartName), manufacturer (Manufacturer), parts compatibility (VehicleModelFit), part price (PartPrice), and inventory stock (InventoryStock). The “Parts” table has a one-to-many relationship with “ServiceInvoice” table because a part can be included in multiple invoices. The attributes in the “Parts” table functionally depend on each other since the table does not contain any non-prime attributes or transitive dependencies:

- PartNumber → PartName, Manufacturer, VehicleModelFit, PartPrice, InventoryStock

Therefore, the table is in third normal form.



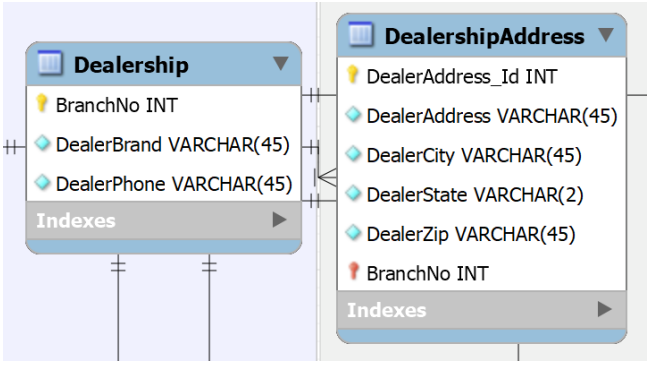
The “ServiceInvoice” table contains an invoice information from a serviced vehicle. The table contains four attributes and five foreign keys. The table contains primary key known as service invoice ID number (ServiceInvoice), the date of transaction (TransactionDate), a foreign key that refers the “ServiceRecords” table known as “Service_Id”, a foreign key that refers the “Parts” table known as “Parts_PartNumber”, service labor (ServiceLabor), service grand total (ServiceTotal), a foreign key that refers the “ServiceEmployees” table known as “Employee_Id”, a foreign key that refers the “Dealership” table known as “Dealership_BranchNo”, and a foreign

key that refers the “ServiceCustomer” table known as “ServiceCust_ID”. The “ServiceInvoice” table has a many-to-one relationship with five different tables. The “ServiceInvoice” table has a many-to-one relationship with “SalesEmployees” table because there can be many service invoices that contains a service employee information. The “ServiceInvoice” table has a many-to-one relationship with “ServiceRecords” table because there can be many invoices that contains a service record. The “ServiceInvoice” table has a many-to-one relationship with “Parts” table because there can be many invoices that contains a part. The “ServiceInvoice” table has a many-to-one relationship with “Dealership” table because because there can be many invoices that contains a dealership branch number. The “ServiceInvoice” table has a many-to-one relationship with “ServiceCustomer” table because there can be many service invoices that contains a customer’s information. The “ServiceInvoice” table does not contain any partial or transitive dependencies, and non-prime attributes since every attribute is functionally dependent:

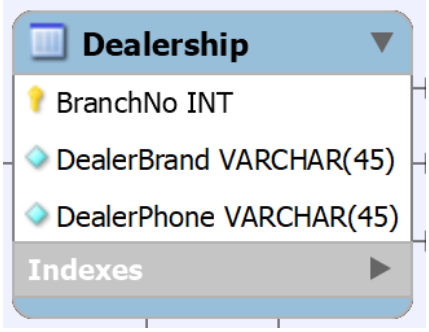
- ServiceInvoice → TransactionDate, Service_Id, Parts_PartNumber, ServiceLabor, ServiceTotal, Employee_Id, Dealership_BranchNo, ServiceCust_ID

Therefore, “ServiceInvoice” is in third normal form.

Normalization Dealership Entity



The “Dealership” entity contains two tables that are dedicated on dealership information. The names of the tables included in this entity are “Dealership” and “DealershipAddress” tables.

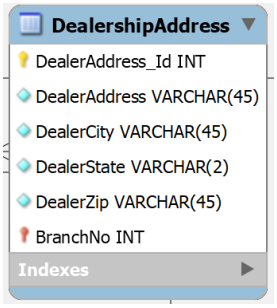


The “Dealership” table contains information about different dealerships. The table contains three attributes. The table contains a primary key known as the dealership branch number (BranchNo), the vehicle brands (DealerBrand), and dealership phone number (DealerPhone). The “Dealership” table is in a one-to-many relationship with “DealershipAddress” table because a dealership can contain many addresses. The “Dealership” table is in a one-to-many relationship with “DealerVehicle” table (from Sales department) because a dealership may contain many vehicles. The “Dealership” table is in a one-to-many relationship with “SalesPersonalInfo” table (from Sales department) because a dealership may contain many employees’ information in the sales department. The “Dealership” table is in a one-to-many relationship with “Invoice” table (from Sales department) because a dealership may contain many invoices. The “Dealership” table is in a one-to-many relationship with “EmployeePersonalInfo” table (from Service department) because a dealership may contain many employees’ information in the service

department. The “Dealership” table is in a one-to-many relationship with “ServiceInvoice” table (from Service department) because a dealership may contain many service invoices. The attributes in the “Dealership” table are functionally dependent and do not contain any transitive or partial dependencies, and non-prime attributes:

- BranchNo → DealerBrand, DealerPhone

Therefore, the table is in third normal form.



The “DealershipAddress” table contains information about a dealership’s location. The table contains five attributes and one foreign that refers the “Dealership” table. The “DealershipAddress” table contains a primary key known as the dealership address number ID (DealerAddress_Id), dealer’s address (DealerAddress), city (DealerCity), state (DealerState), zip code (DealerZip), and the dealership’s branch number foreign key (BranchNo). The “DealershipAddress” table is in a many-to-one relationship with “Dealership” table because there may be many addresses for a dealership. The table contains functional dependencies since it does not contain any partial or transitive dependencies, and non-prime attributes:

- DealerAddress_Id → DealerAddress, DealerCity, DealerState, DealerZip, BranchNo

Therefore, the table is in third normal form.

Test plan and records

Dealership Entity: Insert, Delete, Update, and Aggregate commands

Insert command “Dealership” table

select * from dealership;

insert into dealership

(BranchNo, DealerBrand, DealerPhone)

value

('787', 'Hyundai', '(622)-624-6666');

Expected Output

BranchNo	DealerBrand	DealerPhone
126	Toyota	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
787	Hyundai	(622)-624-6666
847	Chevrolet	(619)-800-0010

Actual Output

	BranchNo	DealerBrand	DealerPhone
▸	126	Toyota	(832)-450-8964
	225	Ford	(619)-619-6196
	255	Nissan	(832)-333-4570
	263	Mercedes	(619)-855-2306
	515	Honda	(832)-180-0018
	787	Hyundai	(622)-624-6666
	847	Chevrolet	(619)-800-0010
*	NULL	NULL	NULL

Delete Command “Dealership” table

select * from dealership;

delete from dealership

where BranchNo = 787;

Expected Output

BranchNo	DealerBrand	DealerPhone
126	Toyota	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
787	Hyundai	(622)-624-6666
847	Chevrolet	(619)-800-0010

BranchNo	DealerBrand	DealerPhone
126	Toyota	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
847	Chevrolet	(619)-800-0010

Actual Output

BranchNo	DealerBrand	DealerPhone
126	Toyota	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
787	Hyundai	(622)-624-6666
847	Chevrolet	(619)-800-0010
NULL	NULL	NULL

BranchNo	DealerBrand	DealerPhone
126	Toyota	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
847	Chevrolet	(619)-800-0010
NULL	NULL	NULL

Update Command “Dealership” table

select * from dealership;

update dealership

set DealerBrand= 'TOYOTA'

where BranchNo = 126;

Expected Output

BranchNo	DealerBrand	DealerPhone
126	TOYOTA	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
847	Chevrolet	(619)-800-0010

Actual Output

BranchNo	DealerBrand	DealerPhone
126	Toyota	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
847	Chevrolet	(619)-800-0010
NULL	NULL	NULL

BranchNo	DealerBrand	DealerPhone
126	TOYOTA	(832)-450-8964
225	Ford	(619)-619-6196
255	Nissan	(832)-333-4570
263	Mercedes	(619)-855-2306
515	Honda	(832)-180-0018
847	Chevrolet	(619)-800-0010
NULL	NULL	NULL

Dealership Entity: Aggregate Command for “Dealership” table

select * from dealership;

select count(*) from dealership;

Expected Output

count(*)
7

Actual Output

	count(*)
▶	7

Sales Department: Insert, Update, Delete commands

Customer Entity

"Customers" table

select * from customers; insert into customers

(Cust_Id, CustomerFirstName, CustomerLastName, CustomerSSN, CustAddress_Id)

value ('20', 'Marcus', 'Bradley', '20XXXXXXX', '20');

Expected Output

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
1	John	Doe	1XXXXXXXX	1
⋮	⋮	⋮	⋮	⋮
18	Johnny	Hansley	18XXXXXXX	18
19	William	Garrison	19XXXXXXX	19
20	Marcus	Bradley	20XXXXXXX	20

Actual Output

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
1	John	Doe	1XXXXXXXX	1
2	Kelly	Johnson	2XXXXXXXX	2
3	Kaleb	Kalisto	3XXXXXXXX	3
4	Lindsey	Lopez	4XXXXXXXX	4
5	Malik	Willis	5XXXXXXXX	5
6	George	Guillano	6XXXXXXXX	6
7	Michael	Smith	7XXXXXXXX	7
8	Sabrina	Johnson	8XXXXXXXX	8
9	Maria	Cabrera	9XXXXXXXX	9
10	Jose	Guitierrez	10XXXXXXX	10
11	Lucia	Mendez	11XXXXXXX	11
12	Aarav	Kumar	12XXXXXXX	12
13	Anastasia	Sidorov	13XXXXXXX	13
14	Mel	Jibson	14XXXXXXX	14
15	Joshua	Morrison	15XXXXXXX	15
16	Jordan	Jones	16XXXXXXX	16
17	Destiny	Aisha	17XXXXXXX	17
18	Johnny	Hansley	18XXXXXXX	18
19	William	Garrison	19XXXXXXX	19
20	Marcus	Bradley	20XXXXXXX	20
NULL	NULL	NULL	NULL	NULL

DELETE Command

Delete “Customers” table

select * from customers;

delete from customers

where Cust_id = 20;

Expected Output

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
⋮	⋮	⋮	⋮	⋮
18	Johnny	Hansley	18XXXXXXX	18
19	William	Garrison	19XXXXXXX	19
20	Marcus	Bradley	20XXXXXXX	20

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
⋮	⋮	⋮	⋮	⋮
18	Johnny	Hansley	18XXXXXXX	18
19	William	Garrison	19XXXXXXX	19

Actual Output

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
1	John	Doe	1XXXXXXX	1
2	Kelly	Johnson	2XXXXXXX	2
3	Kaleb	Kalisto	3XXXXXXX	3
4	Lindsey	Lopez	4XXXXXXX	4
5	Malik	Willis	5XXXXXXX	5
6	George	Guillano	6XXXXXXX	6
7	Michael	Smith	7XXXXXXX	7
8	Sabrina	Johnson	8XXXXXXX	8
9	Maria	Cabrera	9XXXXXXX	9
10	Jose	Guitierrez	10XXXXXX	10
11	Lucia	Mendez	11XXXXXX	11
12	Aarav	Kumar	12XXXXXX	12
13	Anastasia	Sidorov	13XXXXXX	13
14	Mel	Jibson	14XXXXXX	14
15	Joshua	Morrison	15XXXXXX	15
16	Jordan	Jones	16XXXXXX	16
17	Destiny	Aisha	17XXXXXX	17
18	Johnny	Hansley	18XXXXXX	18
19	William	Garrison	19XXXXXX	19
20	Marcus	Bradley	20XXXXXX	20

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
1	John	Doe	1XXXXXXX	1
2	Kelly	Johnson	2XXXXXXX	2
3	Kaleb	Kalisto	3XXXXXXX	3
4	Lindsey	Lopez	4XXXXXXX	4
5	Malik	Willis	5XXXXXXX	5
6	George	Guillano	6XXXXXXX	6
7	Michael	Smith	7XXXXXXX	7
8	Sabrina	Johnson	8XXXXXXX	8
9	Maria	Cabrera	9XXXXXXX	9
10	Jose	Guitierrez	10XXXXXX	10
11	Lucia	Mendez	11XXXXXX	11
12	Aarav	Kumar	12XXXXXX	12
13	Anastasia	Sidorov	13XXXXXX	13
14	Mel	Jibson	14XXXXXX	14
15	Joshua	Morrison	15XXXXXX	15
16	Jordan	Jones	16XXXXXX	16
17	Destiny	Aisha	17XXXXXX	17
18	Johnny	Hansley	18XXXXXX	18
19	William	Garrison	19XXXXXX	19

Update “Customers” table

```
select * from customers;

update customers

set CustomerFirstName = 'Jamie'

where Cust_Id = 19;
```

Expected Output

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
⋮	⋮	⋮	⋮	⋮
18	Johnny	Hansley	18XXXXXXX	18
19	Jamie	Garrison	19XXXXXXX	19

Actual result

Cust_Id	CustomerFirstName	CustomerLastName	CustomerSSN	CustAddress_Id
1	John	Doe	1XXXXXXXX	1
2	Kelly	Johnson	2XXXXXXXX	2
3	Kaleb	Kalisto	3XXXXXXXX	3
4	Lindsey	Lopez	4XXXXXXXX	4
5	Malik	Willis	5XXXXXXXX	5
6	George	Guillano	6XXXXXXXX	6
7	Michael	Smith	7XXXXXXXX	7
8	Sabrina	Johnson	8XXXXXXXX	8
9	Maria	Cabrera	9XXXXXXXX	9
10	Jose	Guitierrez	10XXXXXXX	10
11	Lucia	Mendez	11XXXXXXX	11
12	Aarav	Kumar	12XXXXXXX	12
13	Anastasia	Sidorov	13XXXXXXX	13
14	Mel	Jibson	14XXXXXXX	14
15	Joshua	Morrison	15XXXXXXX	15
16	Jordan	Jones	16XXXXXXX	16
17	Destiny	Aisha	17XXXXXXX	17
18	Johnny	Hansley	18XXXXXXX	18
19	Jamie	Garrison	19XXXXXXX	19
NULL	NULL	NULL	NULL	NULL

Sales Employees Entity

Insert command for “SalesEmployees” table

select * from salesemployees;

insert into salesemployees

(SalesEmp_Id, SalesFirstName, SalesLastName, SalesSSN, SalesAddress_Id)

value ('25', 'Kalil', 'Brown', '56XXXXXXX', '25')

Expected Output

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
⋮	⋮	⋮	⋮	⋮
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24
25	Kalil	Brown	56XXXXXXX	25

Actual Output

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
1	Claudia	Hartley	32XXXXXXX	1
2	Faizah	Murillo	33XXXXXXX	2
3	Nathan	Ellis	34XXXXXXX	3
4	Saad	Sheridan	35XXXXXXX	4
5	Tufor	Swift	36XXXXXXX	5
6	James	Carter	37XXXXXXX	6
7	Lilian	Power	38XXXXXXX	7
8	Jason	Mandela	39XXXXXXX	8
9	Cristiano	Rodriguez	40XXXXXXX	9
10	Robert	Johnson	41XXXXXXX	10
11	Crystal	Martinez	42XXXXXXX	11
12	James	Roberts	43XXXXXXX	12
13	Xiao	Lu	44XXXXXXX	13
14	Yukio	Yamamoto	45XXXXXXX	14
15	James	Carson	46XXXXXXX	15
16	Mike	Davidson	47XXXXXXX	16
17	Alice	Wonderland	48XXXXXXX	17
18	Marco	Polo	49XXXXXXX	18
19	Mike	Alliston	50XXXXXXX	19
20	Mark	Whatley	51XXXXXXX	20
21	Joseph	Micheals	52XXXXXXX	21
22	Jerry	Randy	53XXXXXXX	22
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24
25	Kalil	Brown	56XXXXXXX	25

Delete command for “SalesEmployees” table

select * from salesemployees;

delete from salesemployees

where SalesEmp_Id = 25;

Expected Output

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
⋮	⋮	⋮	⋮	⋮
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24
25	Kalil	Brown	56XXXXXXX	25

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
⋮	⋮	⋮	⋮	⋮
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24

Actual Output

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
1	Claudia	Hartley	32XXXXXXX	1
2	Faizah	Murillo	33XXXXXXX	2
3	Nathan	Ellis	34XXXXXXX	3
4	Saad	Sheridan	35XXXXXXX	4
5	Tufor	Swift	36XXXXXXX	5
6	James	Carter	37XXXXXXX	6
7	Lilian	Power	38XXXXXXX	7
8	Jason	Mandela	39XXXXXXX	8
9	Cristiano	Rodriguez	40XXXXXXX	9
10	Robert	Johnson	41XXXXXXX	10
11	Crystal	Martinez	42XXXXXXX	11
12	James	Roberts	43XXXXXXX	12
13	Xiao	Lu	44XXXXXXX	13
14	Yukio	Yamamoto	45XXXXXXX	14
15	James	Carson	46XXXXXXX	15
16	Mike	Davidson	47XXXXXXX	16
17	Alice	Wonderland	48XXXXXXX	17
18	Marco	Polo	49XXXXXXX	18
19	Mike	Alliston	50XXXXXXX	19
20	Mark	Whatley	51XXXXXXX	20
21	Joseph	Micheals	52XXXXXXX	21
22	Jerry	Randy	53XXXXXXX	22
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24
25	Kalil	Brown	56XXXXXXX	25

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
1	Claudia	Hartley	32XXXXXXX	1
2	Faizah	Murillo	33XXXXXXX	2
3	Nathan	Ellis	34XXXXXXX	3
4	Saad	Sheridan	35XXXXXXX	4
5	Tufor	Swift	36XXXXXXX	5
6	James	Carter	37XXXXXXX	6
7	Lilian	Power	38XXXXXXX	7
8	Jason	Mandela	39XXXXXXX	8
9	Cristiano	Rodriguez	40XXXXXXX	9
10	Robert	Johnson	41XXXXXXX	10
11	Crystal	Martinez	42XXXXXXX	11
12	James	Roberts	43XXXXXXX	12
13	Xiao	Lu	44XXXXXXX	13
14	Yukio	Yamamoto	45XXXXXXX	14
15	James	Carson	46XXXXXXX	15
16	Mike	Davidson	47XXXXXXX	16
17	Alice	Wonderland	48XXXXXXX	17
18	Marco	Polo	49XXXXXXX	18
19	Mike	Alliston	50XXXXXXX	19
20	Mark	Whatley	51XXXXXXX	20
21	Joseph	Micheals	52XXXXXXX	21
22	Jerry	Randy	53XXXXXXX	22
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24

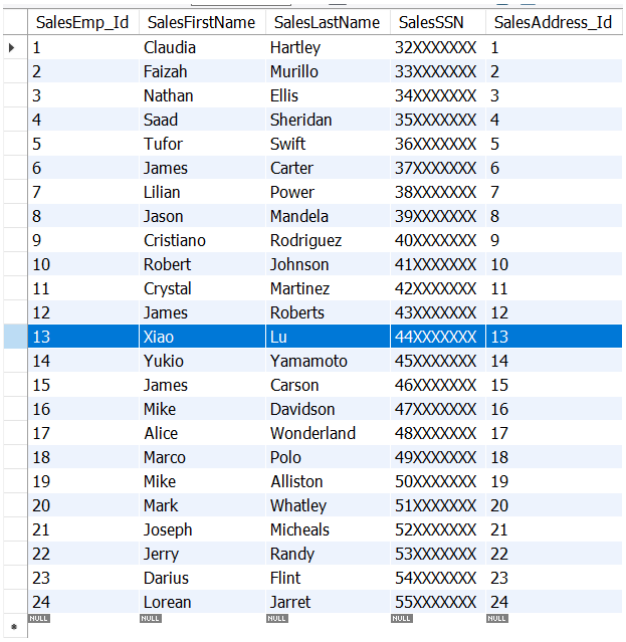
Update command for “SalesEmployees” table

```
select * from salesemployees;

update salesemployees

set SalesFirstName = 'Lu', SalesLastName = 'Xiao'

where SalesEmp_Id = 13;
```

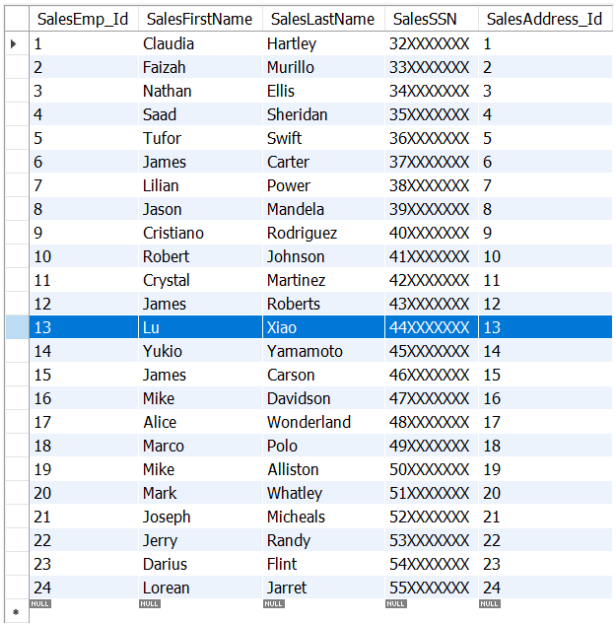
Expected Output

SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
⋮	⋮	⋮	⋮	⋮
11	Crystal	Martinez	42XXXXXXX	11
12	James	Roberts	43XXXXXXX	12
13	Lu	Xiao	44XXXXXXX	13

Actual Output



SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
1	Claudia	Hartley	32XXXXXXX	1
2	Faizah	Murillo	33XXXXXXX	2
3	Nathan	Ellis	34XXXXXXX	3
4	Saad	Sheridan	35XXXXXXX	4
5	Tufor	Swift	36XXXXXXX	5
6	James	Carter	37XXXXXXX	6
7	Lilian	Power	38XXXXXXX	7
8	Jason	Mandela	39XXXXXXX	8
9	Cristiano	Rodriguez	40XXXXXXX	9
10	Robert	Johnson	41XXXXXXX	10
11	Crystal	Martinez	42XXXXXXX	11
12	James	Roberts	43XXXXXXX	12
13	Xiao	Lu	44XXXXXXX	13
14	Yukio	Yamamoto	45XXXXXXX	14
15	James	Carson	46XXXXXXX	15
16	Mike	Davidson	47XXXXXXX	16
17	Alice	Wonderland	48XXXXXXX	17
18	Marco	Polo	49XXXXXXX	18
19	Mike	Alliston	50XXXXXXX	19
20	Mark	Whatley	51XXXXXXX	20
21	Joseph	Micheals	52XXXXXXX	21
22	Jerry	Randy	53XXXXXXX	22
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24



SalesEmp_Id	SalesFirstName	SalesLastName	SalesSSN	SalesAddress_Id
1	Claudia	Hartley	32XXXXXXX	1
2	Faizah	Murillo	33XXXXXXX	2
3	Nathan	Ellis	34XXXXXXX	3
4	Saad	Sheridan	35XXXXXXX	4
5	Tufor	Swift	36XXXXXXX	5
6	James	Carter	37XXXXXXX	6
7	Lilian	Power	38XXXXXXX	7
8	Jason	Mandela	39XXXXXXX	8
9	Cristiano	Rodriguez	40XXXXXXX	9
10	Robert	Johnson	41XXXXXXX	10
11	Crystal	Martinez	42XXXXXXX	11
12	James	Roberts	43XXXXXXX	12
13	Lu	Xiao	44XXXXXXX	13
14	Yukio	Yamamoto	45XXXXXXX	14
15	James	Carson	46XXXXXXX	15
16	Mike	Davidson	47XXXXXXX	16
17	Alice	Wonderland	48XXXXXXX	17
18	Marco	Polo	49XXXXXXX	18
19	Mike	Alliston	50XXXXXXX	19
20	Mark	Whatley	51XXXXXXX	20
21	Joseph	Micheals	52XXXXXXX	21
22	Jerry	Randy	53XXXXXXX	22
23	Darius	Flint	54XXXXXXX	23
24	Lorean	Jarret	55XXXXXXX	24

Sales Entity

Insert command for “Invoice” table

select * from invoice;

insert into invoice

(Invoice_Id, SalesDate, Cust_Id, Payment_Id, GrandTotal, SalesEmp_Id, Dealership_BranchNo, DealerVehicleId)

value ('20', '2022-03-15', '20', '20', '7538', '24', '847', '59');

Expected Output

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30
20	2022-03-15	20	20	7538.00	24	847	59

Actual Output

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
1	2022-09-24	1	1	44585.00	1	126	1
2	2022-04-04	2	2	21423.00	2	126	4
3	2022-12-01	3	3	31405.00	5	225	11
4	2022-08-07	4	4	14190.00	4	225	16
5	2022-02-12	5	5	43655.00	9	255	21
6	2022-07-07	6	6	14548.00	6	225	19
7	2022-03-20	7	7	447998.00	12	263	36
8	2022-08-27	8	8	25775.00	3	126	3
9	2022-09-15	9	9	21581.00	17	847	57
10	2022-12-11	10	10	73364.00	16	847	58
11	2022-05-13	11	11	29255.00	15	515	42
12	2022-08-20	12	12	31800.00	11	263	33
13	2022-05-01	13	13	42938.00	10	263	37
14	2022-12-29	14	14	35878.00	15	515	47
15	2022-08-05	15	15	25620.00	5	225	13
16	2022-05-28	16	16	20322.00	8	255	22
17	2022-11-10	17	17	16690.00	9	255	28
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30
20	2022-03-15	20	20	7538.00	24	847	59
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Delete command for “Invoice” table

select * from invoice;

delete from invoice

where Invoice_Id = 20;

Expected Output

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30
20	2022-03-15	20	20	7538.00	24	847	59

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30

Actual Output

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
1	2022-09-24	1	1	44585.00	1	126	1
2	2022-04-04	2	2	21423.00	2	126	4
3	2022-12-01	3	3	31405.00	5	225	11
4	2022-08-07	4	4	14190.00	4	225	16
5	2022-02-12	5	5	43655.00	9	255	21
6	2022-07-07	6	6	14548.00	6	225	19
7	2022-03-20	7	7	447998.00	12	263	36
8	2022-08-27	8	8	25775.00	3	126	3
9	2022-09-15	9	9	21581.00	17	847	57
10	2022-12-11	10	10	73364.00	16	847	58
11	2022-05-13	11	11	29255.00	15	515	42
12	2022-08-20	12	12	31800.00	11	263	33
13	2022-05-01	13	13	42938.00	10	263	37
14	2022-12-29	14	14	35878.00	15	515	47
15	2022-08-05	15	15	25620.00	5	225	13
16	2022-05-28	16	16	20322.00	8	255	22
17	2022-11-10	17	17	16690.00	9	255	28
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30
20	2022-03-15	20	20	7538.00	24	847	59

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
1	2022-09-24	1	1	44585.00	1	126	1
2	2022-04-04	2	2	21423.00	2	126	4
3	2022-12-01	3	3	31405.00	5	225	11
4	2022-08-07	4	4	14190.00	4	225	16
5	2022-02-12	5	5	43655.00	9	255	21
6	2022-07-07	6	6	14548.00	6	225	19
7	2022-03-20	7	7	447998.00	12	263	36
8	2022-08-27	8	8	25775.00	3	126	3
9	2022-09-15	9	9	21581.00	17	847	57
10	2022-12-11	10	10	73364.00	16	847	58
11	2022-05-13	11	11	29255.00	15	515	42
12	2022-08-20	12	12	31800.00	11	263	33
13	2022-05-01	13	13	42938.00	10	263	37
14	2022-12-29	14	14	35878.00	15	515	47
15	2022-08-05	15	15	25620.00	5	225	13
16	2022-05-28	16	16	20322.00	8	255	22
17	2022-11-10	17	17	16690.00	9	255	28
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30

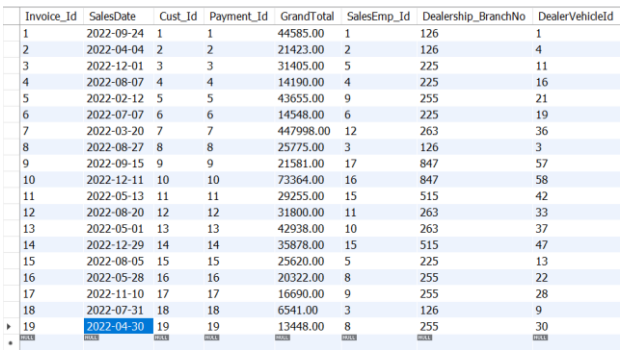
Update command for “Invoice” table

```
select * from invoice;

update invoice

set SalesDate = '2022-04-29'

where Invoice_Id = 19;
```

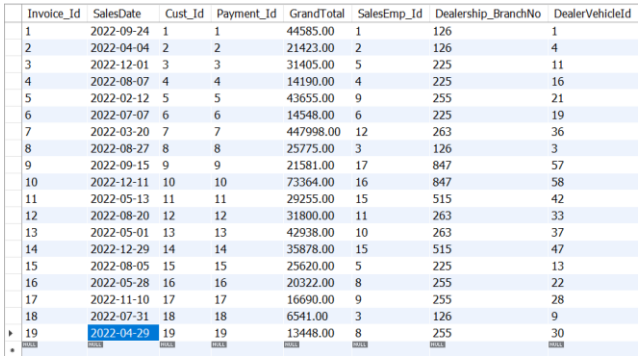
Expected Output

Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-29	19	19	13448.00	8	255	30

Actual Output



Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
1	2022-09-24	1	1	44585.00	1	126	1
2	2022-04-04	2	2	21423.00	2	126	4
3	2022-12-01	3	3	31405.00	5	225	11
4	2022-08-07	4	4	14190.00	4	225	16
5	2022-02-12	5	5	43655.00	9	255	21
6	2022-07-07	6	6	14548.00	6	225	19
7	2022-03-20	7	7	447998.00	12	263	36
8	2022-08-27	8	8	25775.00	3	126	3
9	2022-09-15	9	9	21581.00	17	847	57
10	2022-12-11	10	10	73364.00	16	847	58
11	2022-05-13	11	11	29255.00	15	515	42
12	2022-08-20	12	12	31800.00	11	263	33
13	2022-05-01	13	13	42938.00	10	263	37
14	2022-12-29	14	14	35878.00	15	515	47
15	2022-08-05	15	15	25620.00	5	225	13
16	2022-05-28	16	16	20322.00	8	255	22
17	2022-11-10	17	17	16690.00	9	255	28
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-30	19	19	13448.00	8	255	30



Invoice_Id	SalesDate	Cust_Id	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo	DealerVehicleId
1	2022-09-24	1	1	44585.00	1	126	1
2	2022-04-04	2	2	21423.00	2	126	4
3	2022-12-01	3	3	31405.00	5	225	11
4	2022-08-07	4	4	14190.00	4	225	16
5	2022-02-12	5	5	43655.00	9	255	21
6	2022-07-07	6	6	14548.00	6	225	19
7	2022-03-20	7	7	447998.00	12	263	36
8	2022-08-27	8	8	25775.00	3	126	3
9	2022-09-15	9	9	21581.00	17	847	57
10	2022-12-11	10	10	73364.00	16	847	58
11	2022-05-13	11	11	29255.00	15	515	42
12	2022-08-20	12	12	31800.00	11	263	33
13	2022-05-01	13	13	42938.00	10	263	37
14	2022-12-29	14	14	35878.00	15	515	47
15	2022-08-05	15	15	25620.00	5	225	13
16	2022-05-28	16	16	20322.00	8	255	22
17	2022-11-10	17	17	16690.00	9	255	28
18	2022-07-31	18	18	6541.00	3	126	9
19	2022-04-29	19	19	13448.00	8	255	30

Dealership Vehicle Entity

Insert command for “DealerVehicles” table

select * from dealervehicles;

insert into dealervehicles

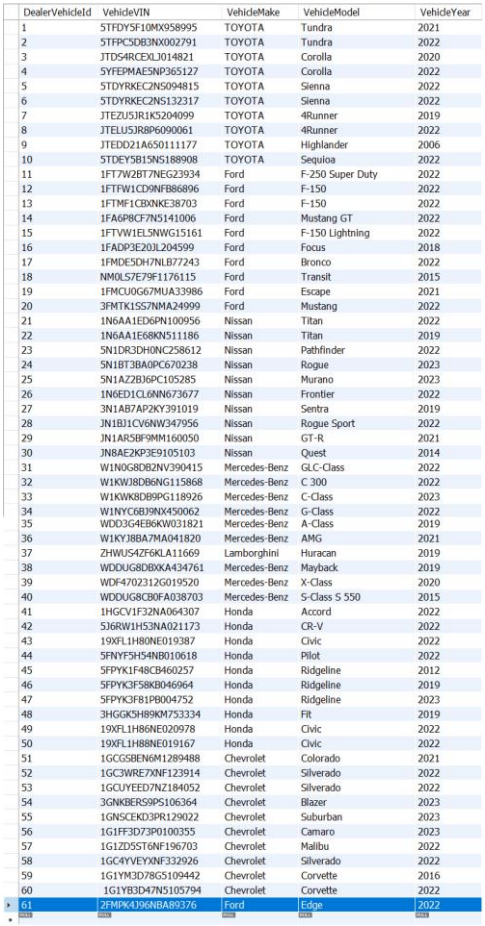
(DealerVehicleId, VehicleVIN, VehicleMake, VehicleModel, VehicleYear)

value ('61', '2FMPK4J96NBA89376', 'Ford', 'Edge', '2022');

Expected Output

DealerVehicleId	VehicleVIN	VehicleMake	VehicleModel	VehicleYear
⋮	⋮	⋮	⋮	⋮
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2022
61	2FMPK4J96NBA89376	Ford	Edge	2022

Actual Output



DealerVehicleId	VehicleVIN	VehicleMake	VehicleModel	VehicleYear
1	5TFDY5F10MX958995	TOYOTA	Tundra	2021
2	5TFPC5DB3NX002791	TOYOTA	Tundra	2022
3	JTDS4RCEXLJ014821	TOYOTA	Corolla	2020
4	5YFEPMAE5NP365127	TOYOTA	Corolla	2022
5	5TDYRKEC2NS094815	TOYOTA	Sienna	2022
6	5TDYRKEC2NS132317	TOYOTA	Sienna	2022
7	JTEZU5JR1K5204099	TOYOTA	4Runner	2019
8	JTELU5JR8P6090061	TOYOTA	4Runner	2022
9	JTEDD21A650111177	TOYOTA	Highlander	2006
10	5TDEY5B15NS188908	TOYOTA	Sequioa	2022
11	1FT7W2BT7NEG23934	Ford	F-250 Super Duty	2022
12	1FTFW1CD9NFB86896	Ford	F-150	2022
13	1FTMF1CBXNKE38703	Ford	F-150	2022
14	1FA6P8CF7N5141006	Ford	Mustang GT	2022
15	1FTVW1EL5NWG15161	Ford	F-150 Lightning	2022
16	1FADP3E20JL204599	Ford	Focus	2018
17	1FMDE5DH7NLB77243	Ford	Bronco	2022
18	NM0LS7E79F1176115	Ford	Transit	2015
19	1FMCU0G67MUA33986	Ford	Escape	2021
20	3FMTK1SS7NMA24999	Ford	Mustang	2022
21	1N6AA1ED6PN100956	Nissan	Titan	2022
22	1N6AA1E68KN511186	Nissan	Titan	2019
23	5N1DR3DH0NC258612	Nissan	Pathfinder	2022
24	5N1BT3BA0PC670238	Nissan	Rogue	2023
25	5N1AZ2BJ6PC105285	Nissan	Murano	2023
26	1N6ED1CL6NN673677	Nissan	Frontier	2022
27	3N1AB7AP2KY391019	Nissan	Sentra	2019
28	JN1BJ1CV6NW347956	Nissan	Rogue Sport	2022
29	JN1AR5BF9MM160050	Nissan	GT-R	2021
30	JN8AE2KP3E9105103	Nissan	Quest	2014
31	W1N0G8DB2NV390415	Mercedes-Benz	GLC-Class	2022
32	W1KWJ8DB6NG115868	Mercedes-Benz	C 300	2022
33	W1KWK8DB9PG118926	Mercedes-Benz	C-Class	2023
34	W1NYC6BJ9NX450062	Mercedes-Benz	G-Class	2022
35	WDD3G4EB6KW031821	Mercedes-Benz	A-Class	2019
36	W1KYJ8BA7MA041820	Mercedes-Benz	AMG	2021
37	ZHWUS4ZF6KLA11669	Lamborghini	Huracan	2019
38	WDDUG8DBXKA434761	Mercedes-Benz	Mayback	2019
39	WDF4702312G019520	Mercedes-Benz	X-Class	2020
40	WDDUG8CB0FA038703	Mercedes-Benz	S-Class S 550	2015
41	1HGCV1F32NA064307	Honda	Accord	2022
42	5J6RW1H53NA021173	Honda	CR-V	2022
43	19XFL1H80NE019387	Honda	Civic	2022
44	5FNYF5H54NB010618	Honda	Pilot	2022
45	5FPYK1F48CB460257	Honda	Ridgeline	2012
46	5FPYK3F58KB046964	Honda	Ridgeline	2019
47	5FPYK3F81PB004752	Honda	Ridgeline	2023
48	3HGGK5H89KM753334	Honda	Fit	2019
49	19XFL1H86NE020978	Honda	Civic	2022
50	19XFL1H88NE019167	Honda	Civic	2022
51	1GCGSBEN6M1289488	Chevrolet	Colorado	2021
52	1GC3WRE7XNF123914	Chevrolet	Silverado	2022
53	1GCUYEED7NZ184052	Chevrolet	Silverado	2022
54	3GNKBERS9PS106364	Chevrolet	Blazer	2023
55	1GNSCEKD3PR129022	Chevrolet	Suburban	2023
56	1G1FF3D73P0100355	Chevrolet	Camaro	2023
57	1G1ZD5ST6NF196703	Chevrolet	Malibu	2022
58	1GC4YVEYXNF332926	Chevrolet	Silverado	2022
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2022
61	2FMPK4J96NBA89376	Ford	Edge	2022

Zoomed it:

56	1G1FF3D73P0100355	Chevrolet	Camaro	2023
57	1G1ZD5ST6NF196703	Chevrolet	Malibu	2022
58	1GC4YVEYXNF332926	Chevrolet	Silverado	2022
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2022
61	2FMPK4J96NBA89376	Ford	Edge	2022
NULL	NULL	NULL	NULL	NULL

Delete Command for “DealerVehicles” table

select * from dealervehicles;

delete from dealervehicles

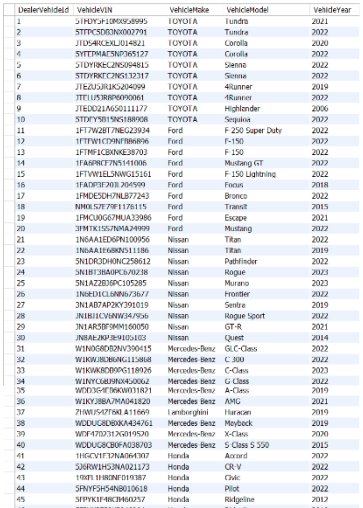
where DealerVehicleId = 61;

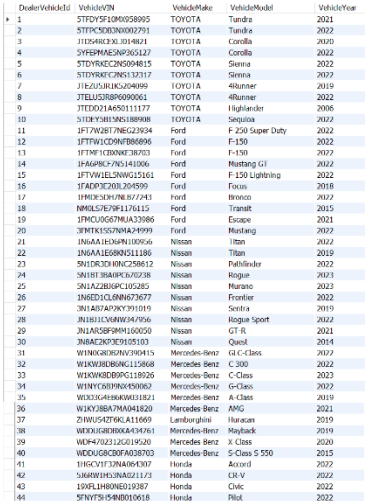
Expected Output

DealerVehicleId	VehicleVIN	VehicleMake	VehicleModel	VehicleYear
⋮	⋮	⋮	⋮	⋮
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2022
61	2FMPK4J96NBA89376	Ford	Edge	2022

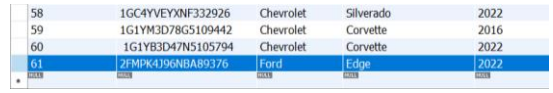
DealerVehicleId	VehicleVIN	VehicleMake	VehicleModel	VehicleYear
⋮	⋮	⋮	⋮	⋮
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2022

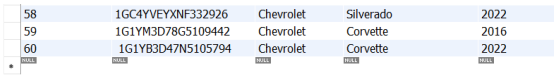
Actual Output





Zoomed it:





Update command for “DealerVehicles” table

```
select * from dealervehicles;

update dealervehicles

set VehicleYear = '2021'

where DealerVehicleId = 60;
```

Expected Output

DealerVehicleId	VehicleVIN	VehicleMake	VehicleModel	VehicleYear
⋮	⋮	⋮	⋮	⋮
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2020

Actual Output

DealerVehicleId	VehicleVIN	VehicleMake	VehicleModel	VehicleYear
1	5TFDY5F10MX958995	TOYOTA	Tundra	2021
2	5TFPC5DB3NX002791	TOYOTA	Tundra	2022
3	JTD54RCEXLJ014821	TOYOTA	Corolla	2020
4	5YFEPMAE5NP365127	TOYOTA	Corolla	2022
5	5TDYRKEC2NS094815	TOYOTA	Sienna	2022
6	5TDYRKEC2NS132317	TOYOTA	Sienna	2022
7	JTEZU5JR1K5204699	TOYOTA	4Runner	2019
8	JTELU5JR8P6090061	TOYOTA	4Runner	2022
9	JTEDD21A650111177	TOYOTA	Highlander	2006
10	5TDEY5B15NS188908	TOYOTA	Sequioa	2022
11	1FT7W2BT7NEG23934	Ford	F-250 Super Duty	2022
12	1FTFW1CD9NFB86896	Ford	F-150	2022
13	1FTMF1CB0NKE38703	Ford	F-150	2022
14	1FA6P8CF7N5141006	Ford	Mustang GT	2022
15	1FTVW1EL5NWG15161	Ford	F-150 Lightning	2022
16	1FADP3E203JL204599	Ford	Focus	2018
17	1FMDE5DH7NLB77243	Ford	Bronco	2022
18	NM0LS7E79F1176115	Ford	Transit	2015
19	1FMCU0G67MUA23986	Ford	Escape	2021
20	3FMTK1SS7NMA24999	Ford	Mustang	2022
21	1N6AA1ED6PN100956	Nissan	Titan	2022
22	1N6AA1E68KN511186	Nissan	Titan	2019
23	5N1DR3DH0NC258612	Nissan	Pathfinder	2022
24	5N1BT3BA0PC670238	Nissan	Rogue	2023
25	5N1AZ2BJ6PC105285	Nissan	Murano	2023
26	1N6ED1CL6NN673677	Nissan	Frontier	2022
27	3N1AB7AP2KY391019	Nissan	Sentra	2019
28	JN1BJ1CV6NW347956	Nissan	Rogue Sport	2022
29	JN1AR5BF9MM160050	Nissan	GT-R	2021
30	JN8AE2KP3E9105103	Nissan	Quest	2014
31	W1N0G8DB2NV390415	Mercedes-Benz	GLC-Class	2022
32	W1KWJ8DB6NG115868	Mercedes-Benz	C 300	2022
33	W1KWK8DB9PG118926	Mercedes-Benz	C-Class	2023
34	W1NYC6BJ9NX450062	Mercedes-Benz	G-Class	2022
35	WDD3G4EB6KW031821	Mercedes-Benz	A-Class	2019
36	W1KYJ8BA7MA041820	Mercedes-Benz	AMG	2021
37	ZHWUS4ZF6KLA11669	Lamborghini	Huracan	2019
38	WDDUG8DB0KA434761	Mercedes-Benz	Mayback	2019
39	WDF4702312G019520	Mercedes-Benz	X-Class	2020
40	WDDUG8CB0FA038703	Mercedes-Benz	S-Class S 550	2015
41	1HGCV1F32NA064307	Honda	Accord	2022
42	5J6RW1H53NA021173	Honda	CR-V	2022
43	19XFL1H80NE019387	Honda	Civic	2022
44	5FNYF5H54NB010618	Honda	Pilot	2022
45	5FPYK1F48CB460257	Honda	Ridgeline	2012
46	5FPYK3F58KB046964	Honda	Ridgeline	2019
47	5FPYK3F81PB004752	Honda	Ridgeline	2023
48	3HGGK5H89KM753334	Honda	Fit	2019
49	19XFL1H86NE020978	Honda	Civic	2022
50	19XFL1H88NE019167	Honda	Civic	2022
51	1GCGSBEN6M1289488	Chevrolet	Colorado	2021
52	1GC3WRE7XNF123914	Chevrolet	Silverado	2022
53	1GCUYEED7NZ184052	Chevrolet	Silverado	2022
54	3GNKBER59PS106364	Chevrolet	Blazer	2023
55	1GNSCEKD3PR129022	Chevrolet	Suburban	2023
56	1G1FF3D73P0100355	Chevrolet	Camaro	2023
57	1G1ZD5ST6NF196703	Chevrolet	Malibu	2022
58	1GC4YVEYXNF332926	Chevrolet	Silverado	2022
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2020

Zoomed it:

58	1GC4YVEYXNF332926	Chevrolet	Silverado	2022
59	1G1YM3D78G5109442	Chevrolet	Corvette	2016
60	1G1YB3D47N5105794	Chevrolet	Corvette	2020

Join Commands Sales Department

Join Customer and Sales Invoice

```
select customers.Cust_Id, customers.CustomerFirstName, customers.CustomerLastName,
        invoice.Invoice_Id, invoice.SalesDate, invoice.Payment_Id, invoice.GrandTotal,
        invoice.SalesEmp_Id, invoice.Dealership_BranchNo
from customers
inner join invoice
on customers.Cust_Id = invoice.Cust_Id;
```

Expected Output

Cust_Id	CustomerFirstName	CustomerLastName	Invoice_Id	SalesDate	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo
1	John	Doe	1	2022-09-24	1	44585.00	1	126
2	Kelly	Johnson	2	2022-04-04	2	21423.00	2	126
3	Kaleb	Kalisto	3	2022-12-01	3	31405.00	5	225
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

Cust_Id	CustomerFirstName	CustomerLastName	Invoice_Id	SalesDate	Payment_Id	GrandTotal	SalesEmp_Id	Dealership_BranchNo
1	John	Doe	1	2022-09-24	1	44585.00	1	126
2	Kelly	Johnson	2	2022-04-04	2	21423.00	2	126
3	Kaleb	Kalisto	3	2022-12-01	3	31405.00	5	225
4	Lindsey	Lopez	4	2022-08-07	4	14190.00	4	225
5	Malik	Willis	5	2022-02-12	5	43655.00	9	255
6	George	Guillano	6	2022-07-07	6	14548.00	6	225
7	Michael	Smith	7	2022-03-20	7	447998.00	12	263
8	Sabrina	Johnson	8	2022-08-27	8	25775.00	3	126
9	Maria	Cabrera	9	2022-09-15	9	21581.00	17	847
10	Jose	Guitierrez	10	2022-12-11	10	73364.00	16	847
11	Lucia	Mendez	11	2022-05-13	11	29255.00	15	515
12	Aarav	Kumar	12	2022-08-20	12	31800.00	11	263
13	Anastasia	Sidorov	13	2022-05-01	13	42938.00	10	263
14	Mel	Jibson	14	2022-12-29	14	35878.00	15	515
15	Joshua	Morrison	15	2022-08-05	15	25620.00	5	225
16	Jordan	Jones	16	2022-05-28	16	20322.00	8	255
17	Destiny	Aisha	17	2022-11-10	17	16690.00	9	255
18	Johnny	Hansley	18	2022-07-31	18	6541.00	3	126
19	William	Garrison	19	2022-04-30	19	13448.00	8	255
20	Marcus	Bradley	20	2022-03-15	20	7538.00	24	847

Join Sales Employees and invoice

```
select salesemployees.SalesEmp_Id, salesemployees.SalesFirstName, salesemployees.SalesLastName,
    invoice.Invoice_Id, invoice.SalesDate, invoice.Payment_Id, invoice.GrandTotal,
    invoice.Dealership_BranchNo
from salesemployees
inner join invoice
on salesemployees.SalesEmp_Id = invoice.SalesEmp_Id;
```

Expected Output

SalesEmp_Id	SalesFirstName	SalesLastName	Invoice_Id	SalesDate	Payment_Id	GrandTotal	Dealership_BranchNo
1	Claudia	Hartley	1	2022-09-24	1	44585.00	126
2	Faizah	Murillo	2	2022-04-04	2	21423.00	126
5	Tufor	Swift	3	2022-12-01	3	31405.00	225
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

SalesEmp_Id	SalesFirstName	SalesLastName	Invoice_Id	SalesDate	Payment_Id	GrandTotal	Dealership_BranchNo
1	Claudia	Hartley	1	2022-09-24	1	44585.00	126
2	Faizah	Murillo	2	2022-04-04	2	21423.00	126
5	Tufor	Swift	3	2022-12-01	3	31405.00	225
4	Saad	Sheridan	4	2022-08-07	4	14190.00	225
9	Cristiano	Rodriguez	5	2022-02-12	5	43655.00	255
6	James	Carter	6	2022-07-07	6	14548.00	225
12	James	Roberts	7	2022-03-20	7	447998.00	263
3	Nathan	Ellis	8	2022-08-27	8	25775.00	126
17	Alice	Wonderland	9	2022-09-15	9	21581.00	847
16	Mike	Davidson	10	2022-12-11	10	73364.00	847
15	James	Carson	11	2022-05-13	11	29255.00	515
11	Crystal	Martinez	12	2022-08-20	12	31800.00	263
10	Robert	Johnson	13	2022-05-01	13	42938.00	263
15	James	Carson	14	2022-12-29	14	35878.00	515
5	Tufor	Swift	15	2022-08-05	15	25620.00	225
8	Jason	Mandela	16	2022-05-28	16	20322.00	255
9	Cristiano	Rodriguez	17	2022-11-10	17	16690.00	255
3	Nathan	Ellis	18	2022-07-31	18	6541.00	126
8	Jason	Mandela	19	2022-04-30	19	13448.00	255
24	Lorean	Jarret	20	2022-03-15	20	7538.00	847

Join Dealership and invoice

```
select dealership.BranchNo, dealership.DealerBrand,
invoice.Invoice_Id, invoice.SalesDate, invoice.Payment_Id, invoice.GrandTotal
from dealership
inner join invoice
on dealership.BranchNo = invoice.Dealership_BranchNo;
```

Expected Output

BranchNo	DealerBrand	Invoice_Id	SalesDate	Payment_Id	GrandTotal
126	Toyota	1	2022-09-24	1	44585.00
126	Toyota	2	2022-04-04	2	21423.00
126	Toyota	8	2022-08-27	8	25775.00
⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

BranchNo	DealerBrand	Invoice_Id	SalesDate	Payment_Id	GrandTotal
126	Toyota	1	2022-09-24	1	44585.00
126	Toyota	2	2022-04-04	2	21423.00
126	Toyota	8	2022-08-27	8	25775.00
126	Toyota	18	2022-07-31	18	6541.00
225	Ford	3	2022-12-01	3	31405.00
225	Ford	4	2022-08-07	4	14190.00
225	Ford	6	2022-07-07	6	14548.00
225	Ford	15	2022-08-05	15	25620.00
255	Nissan	5	2022-02-12	5	43655.00
255	Nissan	16	2022-05-28	16	20322.00
255	Nissan	17	2022-11-10	17	16690.00
255	Nissan	19	2022-04-30	19	13448.00
263	Mercedes	7	2022-03-20	7	447998.00
263	Mercedes	12	2022-08-20	12	31800.00
263	Mercedes	13	2022-05-01	13	42938.00
515	Honda	11	2022-05-13	11	29255.00
515	Honda	14	2022-12-29	14	35878.00
847	Chevrolet	9	2022-09-15	9	21581.00
847	Chevrolet	10	2022-12-11	10	73364.00
847	Chevrolet	20	2022-03-15	20	7538.00

Join Dealer Vehicles and invoice

```
select dealervehicles.VehicleVIN, dealervehicles.VehicleMake, dealervehicles.VehicleModel,
       dealervehicles.VehicleYear,
       invoice.Invoice_Id, invoice.SalesDate, invoice.Payment_Id, invoice.GrandTotal,
      invoice.Dealership_BranchNo
from dealervehicles
inner join invoice
on dealervehicles.DealerVehicleId = invoice.DealerVehicleId;
```

Expected Output

VehicleVIN	VehicleMake	VehicleModel	VehicleYear	Invoice_Id	SalesDate	Payment_Id	GrandTotal	Dealership_BranchNo
5TFDY5F10MX958995	TOYOTA	Tundra	2021	1	2022-09-24	1	44585.00	126
5YFEPMAE5NP365127	TOYOTA	Corolla	2022	2	2022-04-04	2	21423.00	126
1FT7W2BT7NEG23934	Ford	F-250 Super Duty	2022	3	2022-12-01	3	31405.00	225
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

VehicleVIN	VehicleMake	VehicleModel	VehicleYear	Invoice_Id	SalesDate	Payment_Id	GrandTotal	Dealership_BranchNo
5TFDY5F10MX958995	TOYOTA	Tundra	2021	1	2022-09-24	1	44585.00	126
5YFEPMAE5NP365127	TOYOTA	Corolla	2022	2	2022-04-04	2	21423.00	126
1FT7W2BT7NEG23934	Ford	F-250 Super Duty	2022	3	2022-12-01	3	31405.00	225
1FADP3E20JL204599	Ford	Focus	2018	4	2022-08-07	4	14190.00	225
1N6AA1ED6PN100956	Nissan	Titan	2022	5	2022-02-12	5	43655.00	255
1FMCU0G67MUA33986	Ford	Escape	2021	6	2022-07-07	6	14548.00	225
W1KYJ8BA7MA041820	Mercedes-Benz	AMG	2021	7	2022-03-20	7	447998.00	263
JTDS4RCEXLJ014821	TOYOTA	Corolla	2020	8	2022-08-27	8	25775.00	126
1G1ZD5ST6NF196703	Chevrolet	Malibu	2022	9	2022-09-15	9	21581.00	847
1GC4YVEYXNF332926	Chevrolet	Silverado	2022	10	2022-12-11	10	73364.00	847
5J6RW1H53NA021173	Honda	CR-V	2022	11	2022-05-13	11	29255.00	515
W1KWK8DB9PG118926	Mercedes-Benz	C-Class	2023	12	2022-08-20	12	31800.00	263
ZHWUS4ZF6KLA11669	Lamborghini	Huracan	2019	13	2022-05-01	13	42938.00	263
5FPYK3F81PB004752	Honda	Ridgeline	2023	14	2022-12-29	14	35878.00	515
1FTMF1CBXNKE38703	Ford	F-150	2022	15	2022-08-05	15	25620.00	225
1N6AA1E68KN511186	Nissan	Titan	2019	16	2022-05-28	16	20322.00	255
JN1BJ1CV6NW347956	Nissan	Rogue Sport	2022	17	2022-11-10	17	16690.00	255
JTEDD21A650111177	TOYOTA	Highlander	2006	18	2022-07-31	18	6541.00	126
JN8AE2KP3E9105103	Nissan	Quest	2014	19	2022-04-30	19	13448.00	255
1G1YM3D78G5109442	Chevrolet	Corvette	2016	20	2022-03-15	20	7538.00	847

Aggregate Commands Sales Department

Aggregate Customers Entity

“Customers” table count

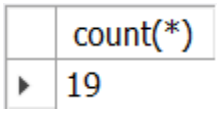
select * from customers;

select count(*) from customers;

Expected Output

count(*)
19

Actual result



	count(*)
▶	19

Aggregate Sales Employees Entity

select *

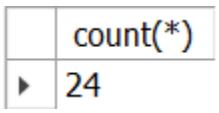
from salesemployees;

select count(*) from salesemployees;

Expected Output

count(*)
24

Actual result



	count(*)
▶	24

Aggregate Sales Entity

“Invoice” table sum

select * from invoice;

select sum(GrandTotal) from invoice;

Expected Output

sum(GrandTotal)
961016.00

Actual result

	sum(GrandTotal)
▸	961016.00

Aggregate Dealership Vehicle Entity

“DealerVehicles” table count

select * from dealervehicles;

select count(*) from dealervehicles;

Expected Output

count(*)
60

Actual result

	count(*)
▸	60

Service Department: Insert, Update, Delete commands

Service Customer Entity

Insert command for “ServiceCustomer” table

select * from servicecustomer;

insert into servicecustomer

(ServiceCust_Id, ServiceFirstName, ServiceLastName, CustAddress_Id)

values ('19', 'Yesenia', 'Campos', '19');

Expected Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
⋮	⋮	⋮	⋮
17	Eric	Doe	17
18	Yolanda	Vazquez	18
19	Yesenia	Campos	19

Actual Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
1	Mark	DeVellis	1
2	Carrie	Johnson	2
3	Jonathan	Tesla	3
4	Kyoshi	Sakamoto	4
5	Ronald	McDonald	5
6	Bobby	Ten	6
7	Yennifer	Castillo	7
8	Kelly	McDougal	8
9	Cristiano	Castellas	9
10	Mike	Cox	10
11	Wendy	Chabascos	11
12	Jason	Stark	12
13	Micheal	Dawnson	13
14	Mayra	Martinez	14
15	John	Rilley	15
16	Yenny	McRodgers	16
17	Eric	Doe	17
18	Yolanda	Vazquez	18
19	Yesenia	Campos	19
NULL	NULL	NULL	NULL

Delete command for “ServiceCustomer” table

select * from ServiceCustomer;

delete from ServiceCustomer

where SalesEmp_Id = 25;

Expected Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
⋮	⋮	⋮	⋮
17	Eric	Doe	17
18	Yolanda	Vazquez	18
19	Yesenia	Campos	19

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
⋮	⋮	⋮	⋮
17	Eric	Doe	17
18	Yolanda	Vazquez	18

Actual Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
1	Mark	DeVellis	1
2	Carrie	Johnson	2
3	Jonathan	Tesla	3
4	Kyoshi	Sakamoto	4
5	Ronald	McDonald	5
6	Bobby	Ten	6
7	Yennifer	Castillo	7
8	Kelly	McDougal	8
9	Cristiano	Castellas	9
10	Mike	Cox	10
11	Wendy	Chabascos	11
12	Jason	Stark	12
13	Micheal	Dawnson	13
14	Mayra	Martinez	14
15	John	Rilley	15
16	Yenny	McRodgers	16
17	Eric	Doe	17
18	Yolanda	Vazquez	18
19	Yesenia	Campos	19
NULL	NULL	NULL	NULL

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
1	Mark	DeVellis	1
2	Carrie	Johnson	2
3	Jonathan	Tesla	3
4	Kyoshi	Sakamoto	4
5	Ronald	McDonald	5
6	Bobby	Ten	6
7	Yennifer	Castillo	7
8	Kelly	McDougal	8
9	Cristiano	Castellas	9
10	Mike	Cox	10
11	Wendy	Chabascos	11
12	Jason	Stark	12
13	Micheal	Dawnson	13
14	Mayra	Martinez	14
15	John	Rilley	15
16	Yenny	McRodgers	16
17	Eric	Doe	17
18	Yolanda	Vazquez	18
NULL	NULL	NULL	NULL

Update command for “ServiceCustomer” table

```
select * from servicecustomer;

update servicecustomer

set ServiceLastName = 'Vasquez'

where ServiceCust_Id = 11;
```

Expected Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
⋮	⋮	⋮	⋮
9	Cristiano	Castellas	9
10	Mike	Cox	10
11	Wendy	Vasquez	11
⋮	⋮	⋮	⋮

Actual Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
1	Mark	DeVellis	1
2	Carrie	Johnson	2
3	Jonathan	Tesla	3
4	Kyoshi	Sakamoto	4
5	Ronald	McDonald	5
6	Bobby	Ten	6
7	Yennifer	Castillo	7
8	Kelly	McDougal	8
9	Cristiano	Castellas	9
10	Mike	Cox	10
11	Wendy	Chabascos	11
12	Jason	Stark	12
13	Micheal	Dawnson	13
14	Mayra	Martinez	14
15	John	Rilley	15
16	Yenny	McRodgers	16
17	Eric	Doe	17
18	Yolanda	Vazquez	18

ServiceCust_Id	ServiceFirstName	ServiceLastName	CustAddress_Id
1	Mark	DeVellis	1
2	Carrie	Johnson	2
3	Jonathan	Tesla	3
4	Kyoshi	Sakamoto	4
5	Ronald	McDonald	5
6	Bobby	Ten	6
7	Yennifer	Castillo	7
8	Kelly	McDougal	8
9	Cristiano	Castellas	9
10	Mike	Cox	10
11	Wendy	Vasquez	11
12	Jason	Stark	12
13	Micheal	Dawnson	13
14	Mayra	Martinez	14
15	John	Rilley	15
16	Yenny	McRodgers	16
17	Eric	Doe	17
18	Yolanda	Vazquez	18

Service Employees Entity

Insert command for “ServiceEmployees” table

select * from serviceemployees;

insert into serviceemployees

(Employee_Id, EmployeeFirstName, EmployeeLastName, EmployeeSSN, EmpAddress_Id)

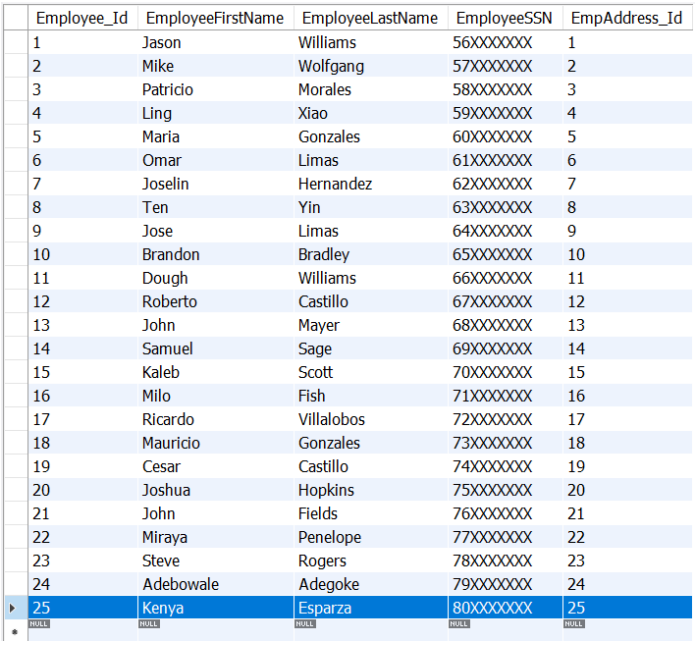
values

('25', 'Kenya', 'Esparza', '80XXXXXXX', '25')

Expected Output

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
⋮	⋮	⋮	⋮	⋮
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24
25	Kenya	Esparza	80XXXXXXX	25

Actual Output



Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
1	Jason	Williams	56XXXXXXX	1
2	Mike	Wolfgang	57XXXXXXX	2
3	Patricio	Morales	58XXXXXXX	3
4	Ling	Xiao	59XXXXXXX	4
5	Maria	Gonzales	60XXXXXXX	5
6	Omar	Limas	61XXXXXXX	6
7	Joselin	Hernandez	62XXXXXXX	7
8	Ten	Yin	63XXXXXXX	8
9	Jose	Limas	64XXXXXXX	9
10	Brandon	Bradley	65XXXXXXX	10
11	Dough	Williams	66XXXXXXX	11
12	Roberto	Castillo	67XXXXXXX	12
13	John	Mayer	68XXXXXXX	13
14	Samuel	Sage	69XXXXXXX	14
15	Kaleb	Scott	70XXXXXXX	15
16	Milo	Fish	71XXXXXXX	16
17	Ricardo	Villalobos	72XXXXXXX	17
18	Mauricio	Gonzales	73XXXXXXX	18
19	Cesar	Castillo	74XXXXXXX	19
20	Joshua	Hopkins	75XXXXXXX	20
21	John	Fields	76XXXXXXX	21
22	Miraya	Penelope	77XXXXXXX	22
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24
25	Kenya	Esparza	80XXXXXXX	25

Delete command for “ServiceEmployees” table

```
select * from serviceemployees;

delete from serviceemployees
where Employee_Id = 25;
```

Expected Output

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
⋮	⋮	⋮	⋮	⋮
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24
25	Kenya	Esparza	80XXXXXXX	25

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
⋮	⋮	⋮	⋮	⋮
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24

Actual Output

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
1	Jason	Williams	56XXXXXXX	1
2	Mike	Wolfgang	57XXXXXXX	2
3	Patricio	Morales	58XXXXXXX	3
4	Ling	Xiao	59XXXXXXX	4
5	Maria	Gonzales	60XXXXXXX	5
6	Omar	Limas	61XXXXXXX	6
7	Joselin	Hernandez	62XXXXXXX	7
8	Ten	Yin	63XXXXXXX	8
9	Jose	Limas	64XXXXXXX	9
10	Brandon	Bradley	65XXXXXXX	10
11	Dough	Williams	66XXXXXXX	11
12	Roberto	Castillo	67XXXXXXX	12
13	John	Mayer	68XXXXXXX	13
14	Samuel	Sage	69XXXXXXX	14
15	Kaleb	Scott	70XXXXXXX	15
16	Milo	Fish	71XXXXXXX	16
17	Ricardo	Villalobos	72XXXXXXX	17
18	Mauricio	Gonzales	73XXXXXXX	18
19	Cesar	Castillo	74XXXXXXX	19
20	Joshua	Hopkins	75XXXXXXX	20
21	John	Fields	76XXXXXXX	21
22	Miraya	Penelope	77XXXXXXX	22
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24
25	Kenya	Esparza	80XXXXXXX	25

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
1	Jason	Williams	56XXXXXXX	1
2	Mike	Wolfgang	57XXXXXXX	2
3	Patricio	Morales	58XXXXXXX	3
4	Ling	Xiao	59XXXXXXX	4
5	Maria	Gonzales	60XXXXXXX	5
6	Omar	Limas	61XXXXXXX	6
7	Joselin	Hernandez	62XXXXXXX	7
8	Ten	Yin	63XXXXXXX	8
9	Jose	Limas	64XXXXXXX	9
10	Brandon	Bradley	65XXXXXXX	10
11	Dough	Williams	66XXXXXXX	11
12	Roberto	Castillo	67XXXXXXX	12
13	John	Mayer	68XXXXXXX	13
14	Samuel	Sage	69XXXXXXX	14
15	Kaleb	Scott	70XXXXXXX	15
16	Milo	Fish	71XXXXXXX	16
17	Ricardo	Villalobos	72XXXXXXX	17
18	Mauricio	Gonzales	73XXXXXXX	18
19	Cesar	Castillo	74XXXXXXX	19
20	Joshua	Hopkins	75XXXXXXX	20
21	John	Fields	76XXXXXXX	21
22	Miraya	Penelope	77XXXXXXX	22
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24

Update command for “ServiceEmployees” table

```
select * from serviceemployees;

update serviceemployees
set EmployeeFirstName = 'Yin', EmployeeLastName = 'Ten'
where Employee_Id = 8;
```

Expected Output

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
⋮	⋮	⋮	⋮	⋮
7	Joselin	Hernandez	62XXXXXXX	7
8	Yin	Ten	63XXXXXXX	8
⋮	⋮	⋮	⋮	⋮
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24

Actual Output

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
1	Jason	Williams	56XXXXXXX	1
2	Mike	Wolfgang	57XXXXXXX	2
3	Patricio	Morales	58XXXXXXX	3
4	Ling	Xiao	59XXXXXXX	4
5	Maria	Gonzales	60XXXXXXX	5
6	Omar	Limas	61XXXXXXX	6
7	Joselin	Hernandez	62XXXXXXX	7
8	Ten	Yin	63XXXXXXX	8
9	Jose	Limas	64XXXXXXX	9
10	Brandon	Bradley	65XXXXXXX	10
11	Dough	Williams	66XXXXXXX	11
12	Roberto	Castillo	67XXXXXXX	12
13	John	Mayer	68XXXXXXX	13
14	Samuel	Sage	69XXXXXXX	14
15	Kaleb	Scott	70XXXXXXX	15
16	Milo	Fish	71XXXXXXX	16
17	Ricardo	Villalobos	72XXXXXXX	17
18	Mauricio	Gonzales	73XXXXXXX	18
19	Cesar	Castillo	74XXXXXXX	19
20	Joshua	Hopkins	75XXXXXXX	20
21	John	Fields	76XXXXXXX	21
22	Miraya	Penelope	77XXXXXXX	22
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24

Employee_Id	EmployeeFirstName	EmployeeLastName	EmployeeSSN	EmpAddress_Id
1	Jason	Williams	56XXXXXXX	1
2	Mike	Wolfgang	57XXXXXXX	2
3	Patricio	Morales	58XXXXXXX	3
4	Ling	Xiao	59XXXXXXX	4
5	Maria	Gonzales	60XXXXXXX	5
6	Omar	Limas	61XXXXXXX	6
7	Joselin	Hernandez	62XXXXXXX	7
8	Yin	Ten	63XXXXXXX	8
9	Jose	Limas	64XXXXXXX	9
10	Brandon	Bradley	65XXXXXXX	10
11	Dough	Williams	66XXXXXXX	11
12	Roberto	Castillo	67XXXXXXX	12
13	John	Mayer	68XXXXXXX	13
14	Samuel	Sage	69XXXXXXX	14
15	Kaleb	Scott	70XXXXXXX	15
16	Milo	Fish	71XXXXXXX	16
17	Ricardo	Villalobos	72XXXXXXX	17
18	Mauricio	Gonzales	73XXXXXXX	18
19	Cesar	Castillo	74XXXXXXX	19
20	Joshua	Hopkins	75XXXXXXX	20
21	John	Fields	76XXXXXXX	21
22	Miraya	Penelope	77XXXXXXX	22
23	Steve	Rogers	78XXXXXXX	23
24	Adebowale	Adegoke	79XXXXXXX	24

Service Invoice Entity

Insert command “ServiceInvoice” table

select * from serviceinvoice;

insert into serviceinvoice

(ServiceInvoice_Id, TransactionDate, Service_Id, Parts_PartNumber, ServiceLabor, ServiceTotal,

Employee_Id, Dealership_BranchNo, ServiceCust_Id)

value ('20', '2022-12-02', '20','N/A', '25.50', '25.50', '25', '515', '19');

Expected Output

ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2020-09-19	18	12696048	75	130.26	21	847	18
19	2022-10-12	19	90915YZZN1	100	257.14	2	126	1
20	2022-12-02	20	N/A	25.50	25.50	25	515	19

Actual Output

ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
1	2022-06-26	1	811500C211	150.00	571.21	1	126	1
2	2021-07-25	2	8155002460	100.00	257.14	2	126	2
3	2021-02-11	3	90915YZZN1	150.00	219.50	1	126	3
4	2022-06-11	4	14529	150.00	178.33	3	225	4
5	2022-08-25	5	10849	150.00	634.35	4	225	5
6	2021-02-16	6	5804320	1500.00	4295.35	4	225	6
7	2021-11-28	7	T99M6-9BU0A	100.00	180.00	5	255	7
8	2020-06-26	8	999L1-VZ001	150.00	425.00	6	255	8
9	2021-03-06	9	15208-9E01A	50.00	146.40	5	255	9
10	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	20	263	10
11	2021-09-04	11	256-140-99-00	1500.00	4419.30	7	263	11
12	2021-05-27	12	276-180-00-09	300.00	415.00	20	263	12
13	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	23	515	13
14	2022-11-02	14	08E16-TBA-100B	100.00	226.75	24	515	14
15	2022-06-03	15	15400-PLM-A02	150.00	251.20	24	515	15
16	2020-03-14	16	22832918	50.00	61.69	22	847	16
17	2022-09-15	17	96955193	100.00	174.97	21	847	17
18	2020-09-19	18	12696048	75.00	130.26	21	847	18
19	2022-10-12	19	90915YZZN1	100.00	257.14	2	126	1
20	2022-12-02	20	N/A	25.50	25.50	25	515	19
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Delete command “ServiceInvoice” table

select * from serviceemployees;

delete from serviceemployees

where Employee_Id = 20;

Expected Output

ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2020-09-19	18	12696048	75.00	130.26	21	847	18
19	2022-10-12	19	90915YZZN1	100.00	257.14	2	126	1
20	2022-12-02	20	N/A	25.50	25.50	25	515	19

Actual Output

	ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
▸	1	2022-06-26	1	811500C211	150.00	571.21	1	126	1
	2	2021-07-25	2	8155002460	100.00	257.14	2	126	2
	3	2021-02-11	3	90915YZZN1	150.00	219.50	1	126	3
	4	2022-06-11	4	14529	150.00	178.33	3	225	4
	5	2022-08-25	5	10849	150.00	634.35	4	225	5
	6	2021-02-16	6	5804320	1500.00	4295.35	4	225	6
	7	2021-11-28	7	T99M6-9BU0A	100.00	180.00	5	255	7
	8	2020-06-26	8	999L1-VZ001	150.00	425.00	6	255	8
	9	2021-03-06	9	15208-9E01A	50.00	146.40	5	255	9
	10	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	20	263	10
	11	2021-09-04	11	256-140-99-00	1500.00	4419.30	7	263	11
	12	2021-05-27	12	276-180-00-09	300.00	415.00	20	263	12
	13	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	23	515	13
	14	2022-11-02	14	08E16-TBA-100B	100.00	226.75	24	515	14
	15	2022-06-03	15	15400-PLM-A02	150.00	251.20	24	515	15
	16	2020-03-14	16	22832918	50.00	61.69	22	847	16
	17	2022-09-15	17	96955193	100.00	174.97	21	847	17
	18	2020-09-19	18	12696048	75.00	130.26	21	847	18
	19	2022-10-12	19	90915YZZN1	100.00	257.14	2	126	1
	20	2022-12-02	20	N/A	25.50	25.50	25	515	19
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
▸	1	2022-06-26	1	811500C211	150.00	571.21	1	126	1
	2	2021-07-25	2	8155002460	100.00	257.14	2	126	2
	3	2021-02-11	3	90915YZZN1	150.00	219.50	1	126	3
	4	2022-06-11	4	14529	150.00	178.33	3	225	4
	5	2022-08-25	5	10849	150.00	634.35	4	225	5
	6	2021-02-16	6	5804320	1500.00	4295.35	4	225	6
	7	2021-11-28	7	T99M6-9BU0A	100.00	180.00	5	255	7
	8	2020-06-26	8	999L1-VZ001	150.00	425.00	6	255	8
	9	2021-03-06	9	15208-9E01A	50.00	146.40	5	255	9
	10	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	20	263	10
	11	2021-09-04	11	256-140-99-00	1500.00	4419.30	7	263	11
	12	2021-05-27	12	276-180-00-09	300.00	415.00	20	263	12
	13	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	23	515	13
	14	2022-11-02	14	08E16-TBA-100B	100.00	226.75	24	515	14
	15	2022-06-03	15	15400-PLM-A02	150.00	251.20	24	515	15
	16	2020-03-14	16	22832918	50.00	61.69	22	847	16
	17	2022-09-15	17	96955193	100.00	174.97	21	847	17
	18	2020-09-19	18	12696048	75.00	130.26	21	847	18
	19	2022-10-12	19	90915YZZN1	100.00	257.14	2	126	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Update command “ServiceInvoice” table

```
select * from serviceinvoice;

update serviceinvoice

set TransactionDate = '2022-11-12'

where ServiceInvoice_Id = 19;
```

Expected Output

ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	2020-09-19	18	12696048	75.00	130.26	21	847	18
19	2022-11-12	19	90915YZZN1	100.00	257.14	2	126	1

Actual Output

ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
1	2022-06-26	1	811500C211	150.00	571.21	1	126	1
2	2021-07-25	2	8155002460	100.00	257.14	2	126	2
3	2021-02-11	3	90915YZZN1	150.00	219.50	1	126	3
4	2022-06-11	4	14529	150.00	178.33	3	225	4
5	2022-08-25	5	10849	150.00	634.35	4	225	5
6	2021-02-16	6	5804320	1500.00	4295.35	4	225	6
7	2021-11-28	7	T99M6-9BU0A	100.00	180.00	5	255	7
8	2020-06-26	8	999L1-VZ001	150.00	425.00	6	255	8
9	2021-03-06	9	15208-9E01A	50.00	146.40	5	255	9
10	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	20	263	10
11	2021-09-04	11	256-140-99-00	1500.00	4419.30	7	263	11
12	2021-05-27	12	276-180-00-09	300.00	415.00	20	263	12
13	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	23	515	13
14	2022-11-02	14	08E16-TBA-100B	100.00	226.75	24	515	14
15	2022-06-03	15	15400-PLM-A02	150.00	251.20	24	515	15
16	2020-03-14	16	22832918	50.00	61.69	22	847	16
17	2022-09-15	17	96955193	100.00	174.97	21	847	17
18	2020-09-19	18	12696048	75.00	130.26	21	847	18
19	2022-10-12	19	90915YZZN1	100.00	257.14	2	126	1

ServiceInvoice_Id	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo	ServiceCust_Id
1	2022-06-26	1	811500C211	150.00	571.21	1	126	1
2	2021-07-25	2	8155002460	100.00	257.14	2	126	2
3	2021-02-11	3	90915YZZN1	150.00	219.50	1	126	3
4	2022-06-11	4	14529	150.00	178.33	3	225	4
5	2022-08-25	5	10849	150.00	634.35	4	225	5
6	2021-02-16	6	5804320	1500.00	4295.35	4	225	6
7	2021-11-28	7	T99M6-9BU0A	100.00	180.00	5	255	7
8	2020-06-26	8	999L1-VZ001	150.00	425.00	6	255	8
9	2021-03-06	9	15208-9E01A	50.00	146.40	5	255	9
10	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	20	263	10
11	2021-09-04	11	256-140-99-00	1500.00	4419.30	7	263	11
12	2021-05-27	12	276-180-00-09	300.00	415.00	20	263	12
13	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	23	515	13
14	2022-11-02	14	08E16-TBA-100B	100.00	226.75	24	515	14
15	2022-06-03	15	15400-PLM-A02	150.00	251.20	24	515	15
16	2020-03-14	16	22832918	50.00	61.69	22	847	16
17	2022-09-15	17	96955193	100.00	174.97	21	847	17
18	2020-09-19	18	12696048	75.00	130.26	21	847	18
19	2022-11-12	19	90915YZZN1	100.00	257.14	2	126	1

Join Commands Service Department

Join "ServiceCustomer" table with "ServiceInvoice" table.

```
select servicecustomer.ServiceCust_Id, servicecustomer.ServiceFirstName,
servicecustomer.ServiceLastName,
serviceinvoice.TransactionDate, serviceinvoice.Service_Id, serviceinvoice.Parts_PartNumber,
serviceinvoice.ServiceLabor, serviceinvoice.ServiceTotal,
serviceinvoice.Employee_Id, serviceinvoice.Dealership_BranchNo
from servicecustomer
inner join serviceinvoice
on servicecustomer.ServiceCust_Id = serviceinvoice.ServiceCust_Id;
```

Expected Output

ServiceCust_Id	ServiceFirstName	ServiceLastName	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo
1	Mark	DeVellis	6/26/2022	1	811500C211	150	571.21	1	126
2	Carrie	Johnson	7/25/2021	2	8155002460	100	257.14	2	126
3	Jonathan	Tesla	2/11/2021	3	90915YZZN1	150	219.5	1	126
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

	ServiceCust_Id	ServiceFirstName	ServiceLastName	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Employee_Id	Dealership_BranchNo
▸	1	Mark	DeVellis	2022-06-26	1	811500C211	150.00	571.21	1	126
	2	Carrie	Johnson	2021-07-25	2	8155002460	100.00	257.14	2	126
	3	Jonathan	Tesla	2021-02-11	3	90915YZZN1	150.00	219.50	1	126
	4	Kyoshi	Sakamoto	2022-06-11	4	14529	150.00	178.33	3	225
	5	Ronald	McDonald	2022-08-25	5	10849	150.00	634.35	4	225
	6	Bobby	Ten	2021-02-16	6	5804320	1500.00	4295.35	4	225
	7	Yennifer	Castillo	2021-11-28	7	T99M6-9BU0A	100.00	180.00	5	255
	8	Kelly	McDougal	2020-06-26	8	999L1-VZ001	150.00	425.00	6	255
	9	Cristiano	Castellas	2021-03-06	9	15208-9E01A	50.00	146.40	5	255
	10	Mike	Cox	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	20	263
	11	Wendy	Vasquez	2021-09-04	11	256-140-99-00	1500.00	4419.30	7	263
	12	Jason	Stark	2021-05-27	12	276-180-00-09	300.00	415.00	20	263
	13	Micheal	Dawnson	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	23	515
	14	Mayra	Martinez	2022-11-02	14	08E16-TBA-100B	100.00	226.75	24	515
	15	John	Rilley	2022-06-03	15	15400-PLM-A02	150.00	251.20	24	515
	16	Yenny	McRodgers	2020-03-14	16	22832918	50.00	61.69	22	847
	17	Eric	Doe	2022-09-15	17	96955193	100.00	174.97	21	847
	18	Yolanda	Vazquez	2020-09-19	18	12696048	75.00	130.26	21	847
	1	Mark	DeVellis	2022-11-12	19	90915YZZN1	100.00	257.14	2	126

Join "ServiceEmployees" table with "ServiceInvoice" table.

```
select serviceemployees.Employee_Id, serviceemployees.EmployeeFirstName,
serviceemployees.EmployeeLastName,
serviceinvoice.TransactionDate, serviceinvoice.Service_Id, serviceinvoice.Parts_PartNumber,
serviceinvoice.ServiceLabor, serviceinvoice.ServiceTotal,
serviceinvoice.Dealership_BranchNo
from serviceemployees
inner join serviceinvoice
on serviceemployees.Employee_Id = serviceinvoice.Employee_Id;
```

Expected Output

Employee_Id	EmployeeFirstName	EmployeeLastName	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Dealership_BranchNo
1	Jason	Williams	6/26/2022	1	811500C211	150	571.21	126
2	Mike	Wolfgang	7/25/2021	2	8155002460	100	257.14	126
1	Jason	Williams	2/11/2021	3	90915YZZN1	150	219.5	126
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

Employee_Id	EmployeeFirstName	EmployeeLastName	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal	Dealership_BranchNo
1	Jason	Williams	2022-06-26	1	811500C211	150.00	571.21	126
2	Mike	Wolfgang	2021-07-25	2	8155002460	100.00	257.14	126
1	Jason	Williams	2021-02-11	3	90915YZZN1	150.00	219.50	126
3	Patricio	Morales	2022-06-11	4	14529	150.00	178.33	225
4	Ling	Xiao	2022-08-25	5	10849	150.00	634.35	225
4	Ling	Xiao	2021-02-16	6	5804320	1500.00	4295.35	225
5	Maria	Gonzales	2021-11-28	7	T99M6-9BU0A	100.00	180.00	255
6	Omar	Limas	2020-06-26	8	999L1-VZ001	150.00	425.00	255
5	Maria	Gonzales	2021-03-06	9	15208-9E01A	50.00	146.40	255
20	Joshua	Hopkins	2022-01-29	10	203-680-01-87-8H81	150.00	354.96	263
7	Joselin	Hernandez	2021-09-04	11	256-140-99-00	1500.00	4419.30	263
20	Joshua	Hopkins	2021-05-27	12	276-180-00-09	300.00	415.00	263
23	Steve	Rogers	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79	515
24	Adebowale	Adegoke	2022-11-02	14	08E16-TBA-100B	100.00	226.75	515
24	Adebowale	Adegoke	2022-06-03	15	15400-PLM-A02	150.00	251.20	515
22	Miraya	Penelope	2020-03-14	16	22832918	50.00	61.69	847
21	John	Fields	2022-09-15	17	96955193	100.00	174.97	847
21	John	Fields	2020-09-19	18	12696048	75.00	130.26	847
2	Mike	Wolfgang	2022-11-12	19	90915YZZN1	100.00	257.14	126

Join “Dealership” table with “ServiceInvoice” table.

```
select dealership.BranchNo, dealership.DealerBrand, dealership.DealerPhone,
serviceinvoice.TransactionDate, serviceinvoice.Service_Id, serviceinvoice.Parts_PartNumber,
serviceinvoice.ServiceLabor, serviceinvoice.ServiceTotal
from dealership
inner join serviceinvoice
on dealership.BranchNo = serviceinvoice.Dealership_BranchNo;
```

Expected Output

BranchNo	DealerBrand	DealerPhone	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal
126	Toyota	(832)-450-8964	6/26/2022	1	811500C211	150	571.21
126	Toyota	(832)-450-8964	7/25/2021	2	8155002460	100	257.14
126	Toyota	(832)-450-8964	2/11/2021	3	90915YZZN1	150	219.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Actual Output

	BranchNo	DealerBrand	DealerPhone	TransactionDate	Service_Id	Parts_PartNumber	ServiceLabor	ServiceTotal
▶	126	Toyota	(832)-450-8964	2022-06-26	1	811500C211	150.00	571.21
	126	Toyota	(832)-450-8964	2021-07-25	2	8155002460	100.00	257.14
	126	Toyota	(832)-450-8964	2021-02-11	3	90915YZZN1	150.00	219.50
	126	Toyota	(832)-450-8964	2022-11-12	19	90915YZZN1	100.00	257.14
	225	Ford	(619)-619-6196	2022-06-11	4	14529	150.00	178.33
	225	Ford	(619)-619-6196	2022-08-25	5	10849	150.00	634.35
	225	Ford	(619)-619-6196	2021-02-16	6	5804320	1500.00	4295.35
	255	Nissan	(832)-333-4570	2021-11-28	7	T99M6-9BU0A	100.00	180.00
	255	Nissan	(832)-333-4570	2020-06-26	8	999L1-VZ001	150.00	425.00
	255	Nissan	(832)-333-4570	2021-03-06	9	15208-9E01A	50.00	146.40
	263	Mercedes	(619)-855-2306	2022-01-29	10	203-680-01-87-8...	150.00	354.96
	263	Mercedes	(619)-855-2306	2021-09-04	11	256-140-99-00	1500.00	4419.30
	263	Mercedes	(619)-855-2306	2021-05-27	12	276-180-00-09	300.00	415.00
	515	Honda	(832)-180-0018	2022-02-13	13	04631-SYE-A00ZZ	150.00	505.79
	515	Honda	(832)-180-0018	2022-11-02	14	08E16-TBA-100B	100.00	226.75
	515	Honda	(832)-180-0018	2022-06-03	15	15400-PLM-A02	150.00	251.20
	847	Chevrolet	(619)-800-0010	2020-03-14	16	22832918	50.00	61.69
	847	Chevrolet	(619)-800-0010	2022-09-15	17	96955193	100.00	174.97
	847	Chevrolet	(619)-800-0010	2020-09-19	18	12696048	75.00	130.26

Aggregate Commands Service Department

Aggregate Command Service Customer Entity

"ServiceCustomer" table count

select * from servicecustomer;

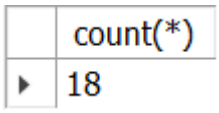
select count(*)

from servicecustomer;

Expected Output

count(*)
18

Actual Output



	count(*)
▸	18

Aggregate Command Service Employees Entity

"ServiceEmployees" table count

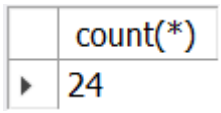
select * from serviceemployees;

select count(*) from serviceemployees;

Expected Output

count(*)
24

Actual Output



	count(*)
▸	24

Aggregate Command Service Invoice Entity

“ServiceInvoice” table count

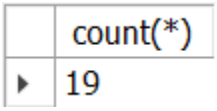
select * from serviceinvoice;

select count(*) from serviceinvoice;

Expected Output

count(*)
19

Actual Output



	count(*)
▸	19

Conclusion

In conclusion, this database was built for the purpose of maintaining data for dealerships. Originally, I wanted to simulate how a dealership typically holds data in a database. However, I assumed that all tables had to be in third normal form. Therefore, I went through many redesigns in my database in order to achieve third normal form for all tables. I went through extensive research in understanding the concept of normalization and attempted to make all my tables in third normal form, and as a result I ended up having a total of 28 tables. I was able to implement the use cases and SQL statements for every table in this database. Some the data in this database was made by me (such as fake social security numbers, phone numbers, addresses, etc.). Some data I had to pick online (such as VIN numbers, car model, etc.) to enhance the dealership experience. I have provided many functions for manipulating data and achieved all the listed objectives. Since there are too many tables in this database, there was a limited amount of time to implement all the functions for every table. However, in the future, I plan to implement more functions for every table in this database.

Resources

https://randomvin.com/

https://www.hausautogroup.com/documents-to-bring-when-buying-a-car.htm#:~:text=Proof%20of%20Residence,home's%20lease%20or%20mortgage%20agreement.

https://www.cars.com/

https://www.cargurus.com/Cars/inventorylisting/viewDetailsFilterViewInventoryListing.action?entitySelectingHelper.selectedEntity=d2361&zip=77090&pid=directVDPSimilarListing#listing=333675160

https://houston.craigslist.org/search/cta#search=1~gallery~0~0

https://www.classicchevysugarland.com/searchnew.aspx?search=new&utm_campaign=dds-branding-namekeywords&utm_content=branding-search&utm_medium=cpc&utm_source=dominion-google-ads

https://www.toyota.com/gst/deals-incentives/?viewAllOffers=1&srchid=SEM:700000002435463:GOOGLE:71700000089852642:58700007613107096:p68780760605:574633314601&gclid=Cj0KCQiAveebBhD_ARIsAFaAvrGguj4LzegI6hsC3cD6MiJWYKvdUmTa3bHMMPtmJpMobAnWtMmkpikaAnJLEALw_wcB&gclsrc=aw.ds

https://www.geeksforgeeks.org/third-normal-form-3nf/

https://parts.ford.com/en.html

https://www.sterlingmccalltoyota.com/parts/

https://www.hondapartsnow.com/

https://parts.chevrolet.com/

https://www.mbusa.com/en/owners/parts?sd_campaign_type=Search&sd_digadprov=Resolution&sd_campaign=Service_Google_Brand_Genuine+Parts&sd_channel=GOOGLE&sd_adid=General+Parts&sd_digadkeyword=mercedes+parts&gclid=Cj0KCQiAveebBhD_ARIsAFaAvrHLdy8QF8XGYcqReB17q1dzxsMu-8XItpCw_sHgopliUo-8a7nT1-kaAmvpEALw_wcB&gclsrc=aw.ds

https://parts.nissanusa.com/

https://www.mbusa.com/en/cpo/inventory?gclid=Cj0KCQiAveebBhD_ARIsAFaAvrEj2a_kfJ2soVLdCYwUTcvs5biIlI38YgrLc8iwEo4m0JXQsqWmwVwaAlSJEALw_wcB&gclsrc=aw.ds

https://uhd.photoshelter.com/galleries/C0000k1pYEiR6HDE/G0000Xd1IeCdCGY4/Main-UHD-Logos-Mark

https://www.uhd.edu/administration/university-relations/Pages/UHD-Logos.aspx

Video Links

Google Drive:

https://drive.google.com/file/d/1nImgQ7Yz7c-ashvd7jSTXH4FdGJmMWuQ/view?usp=share_link

One Drive:

https://1drv.ms/v/s!AsJmZPxfinREnA9pf-S8A1dY2GMI?e=xdrULB