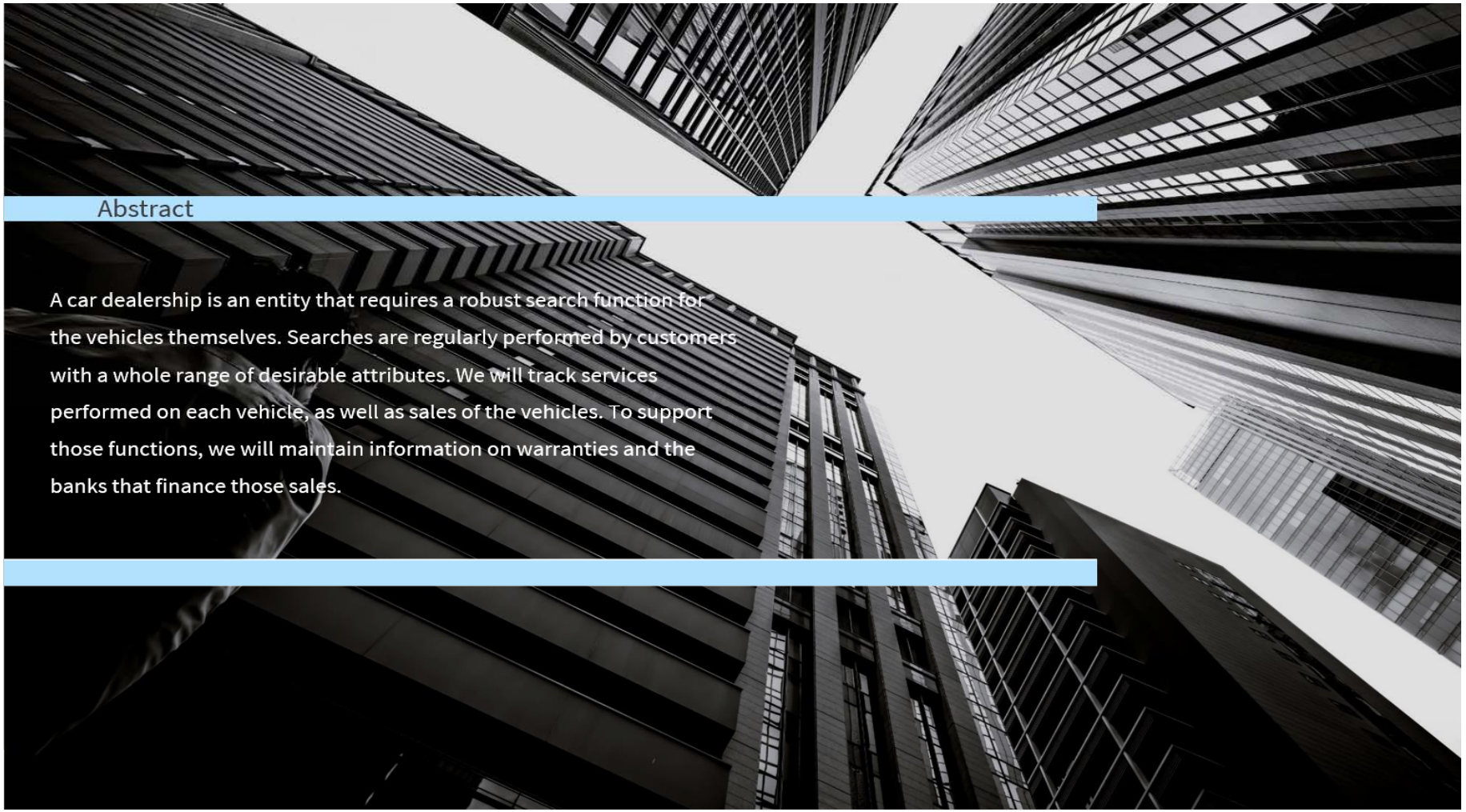
ARP Automotive Group

Alison Sturge, Ruhama Bekele, Bipul Paul



Abstract

A car dealership is an entity that requires a robust search function for the vehicles themselves. Searches are regularly performed by customers with a whole range of desirable attributes. We will track services performed on each vehicle, as well as sales of the vehicles. To support those functions, we will maintain information on warranties and the banks that finance those sales.

Mission Statement

Mission Statement: The purpose of the ARP Automotive Group database is to maintain the data that is used and generated to support vehicle sales and service at our multiple locations in Houston and surrounding areas.

Mission Objectives

To maintain (enter, update, delete) data on Locations
To maintain (enter, update, delete) data on Staff
To maintain (enter, update, delete) data on Customers
To maintain (enter, update, delete) data on Vehicles
To maintain (enter, update, delete) data on Services
To maintain (enter, update, delete) data on Service Codes
To maintain (enter, update, delete) data on Sale Transactions
To maintain (enter, update, delete) data on Banks
To maintain (enter, update, delete) data on Warranties

To perform searches on Locations
To perform searches on Staff
To perform searches on Customers
To perform searches on Vehicles
To perform searches on Services
To perform searches on Service Codes
To perform searches on Sale Transactions
To perform searches on Banks
To perform searches on Warranties

To report on Locations
To report on Staff
To report on Customers
To report on Vehicles
To report on Services
To report on Service Codes
To report on Sale Transactions
To report on Banks
To report on Warranties

Major User Views					
		GM	Dept Mgr	Staff	Customer
Location	Maintain	X			
	Query	X	X	X	
	Report	X			
Staff	Maintain		X		
	Query	X	X		
	Report	X	X		
Customer	Maintain		X	X	X
	Query	X	X	X	
	Report	X	X		
Vehicle	Maintain		X	X	
	Query	X	X	X	X
	Report	X	X	X	X
Service	Maintain		X	X	
	Query	X	X	X	
	Report	X	X	X	
ServiceCode	Maintain		X		
	Query	X	X	X	
	Report	X	X	X	
Sale Transaction	Maintain		X	X	
	Query	X	X		
	Report	X	X		
Warranty	Maintain		X		
	Query	X	X	X	
	Report	X	X	X	
Bank	Maintain		X		
	Query	X	X	X	
	Report	X	X	X	

USE CASES

Actors

- General Manager
- Department Manager
- Staff
- Customer

1 Entering new STAFF
Actor/User: Dept Manager
Steps:

User clicks "New Staff" button
New Staff ID is auto generated and displayed
Prompt user to enter the following:
- Last Name
- First Name
- Address
- Phone
- Location

Display information for confirmation
User clicks "Confirm" button

SQL

```
INSERT into Staff( LastName, FirstName, Address, Phone, Location, Department, Job Title)
VALUES ('Smith',' Mike', '3 Park Avenue Houston 77001', 8322434233, 'Main', 'Operations, 'Manager');
```

2 Modifying STAFF
Actor/User: Dept Manager
Steps:

User clicks "Modify Staff" button
Prompt user to enter Staff ID or Last Name
DB search using user entered search criteria
Display Staff that match search criteria
Prompt user to choose the correct employee
User clicks the correct employee
Display staff record for selected employee.
User clicks "Edit" button
User modifies necessary attribute
User clicks "Save" button

SQL

```
UPDATE Staff
SET JobTitle = 'Manager'
WHERE StaffID = 1;
```

3	**Deleting STAFF:** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Delete Staff" button Prompt user to enter Staff ID Display record for selected employee. Prompt to user: "Do you want to delete this Staff record?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Staff record?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Staff WHERE Staff ID = 1;
4	**Searching STAFF:** **Actor/User: General Manager/Dept Manager** **Steps:**	
		User clicks "Search Staff" button Prompt user to enter search criteria Display new view
	SQL	SELECT LastName, FirstName, Email FROM Staff WHERE Job Title = 'Manager';
5	**Entering new LOCATION** **Actor/User: General Manager** **Steps:**	
		User clicks "New Location" button New Location ID is auto generated and displayed Prompt user to enter the following: Location Name Phone Address Manager Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Location (Location Name, Phone, Address, Manager) VALUES ('MAIN', 7135551000, '100 Main Street Houston 77001', 1);
6	**Modifying LOCATION** **Actor/User: General Manager** **Steps:**	
		User clicks "Modify Location" button Prompt user to enter Location ID Display record for selected Location. User clicks "Edit" button User modifies necessary attribute User clicks "Save" button
	SQL	UPDATE Location SET Manager = 2 WHERE LocationID = 1;

7	**Deleting LOCATION** **Actor/User: General Manager** **Steps:**	
		User clicks "Delete Location" button Prompt user to enter Location ID Display record for selected Location Prompt to user: "Do you want to delete this Location record?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Location record?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Location WHERE LocationID = 1;
8	**Searching LOCATION:** **Actor/User: General Manager/Dept Manager/Staff** **Steps:**	
		User clicks "Search Locations" button Prompt user to enter search criteria Display new view
	SQL	SELECT LocName, Address, Phone, Manager FROM Location WHERE city != 'Houston'
9	**Entering new BANK** **Actor/User: Dept Manager** **Steps:**	
		User clicks "New Bank" button New BankNumber is auto generated and displayed Prompt user to enter the following: Bank Name Email Phone Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Bank(BankName, Email, Phone) VALUES ('Bank of Banking', 'admin@bank.com', 8885551212) ;
10	**Modifying BANK** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Modify Bank" button Prompt user to enter Bank Number Display record for selected Bank User clicks "Edit" button User modifies necessary attribute User clicks "Save" button
	SQL	UPDATE Bank SET Email = jsmith@bank.com WHERE BankCode = 1;

11	**Deleting BANK** **Actor/User:Dept Manager** **Steps:**	
		User clicks "Delete Bank" button Prompt user to enter BankNumber Display record for selected Bank. Prompt to user: "Do you want to delete this Bankrecord?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Bankrecord?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Bank WHERE BankCode = 7;
12	**Searching BANK** **Actor/User: Genera Manager/Dept Manager/Staff** **Steps:**	
		User clicks "Search Banks" button Prompt user to enter search criteria Display new view
	SQL	SELECT BankName, Email, Phone FROM Bank;
13	**Entering new VEHICLE** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "New Vehicle" button New Stock Number is auto generated and displayed Prompt user to enter the following: Year Make Model Color Mileage Price Location VIN New/Used Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Vehicle(Year, Make, Model, Color, Mileage, Price, Location, VIN, New/Used) VALUES(2022, 'Honda', 'Odyssey', 'Maroon', 51000, 31999, 'MAIN', '59JALB45IT987', 'New');

14	**Modifying VEHICLE** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "Modify Vehicle" button Prompt user to enter Stock Number Display record for selected vehicle User clicks "Edit" button User modifies necessary attribute User clicks "Save" button
	SQL	UPDATE Vehicle SET Price = 30999 WHERE StockNum = 124521;
15	**Deleting VEHICLE** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "Delete Vehicle" button Prompt user to enter Stock Number Display record for selected vehicle. Prompt to user: "Do you want to delete this Vehicle?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Vehicle?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Vehicle WHERE StockNum = 124521;
16	**Searching VEHICLE** **Actor/User: General Manager/Dept Manager/Staff/Customer** **Steps:**	
		User clicks "Search Vehicles" button Prompt user to enter search criteria Display new view
	SQL	SELECT Year, Make, Model, Mileage, Price, New/Used FROM Vehicles WHERE Price < 22000;
17	**Entering new SERVICE** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "New Service Record" button SvcDate is auto-generated Prompt user to enter the following: Stock Number Service Code Customer ID Technician Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Service(StockNum, SvcCode, CustID, EmpNum) VALUES(124564, 105, 16, 4);

18	**Modifying SERVICE** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "Modify Service Record" button Prompt user to enter Invoice Number Display selected invoice User clicks "Edit" button User modifies necessary attribute User clicks "Save" button
	SQL	UPDATE Service SET EmpNum = 6 WHERE StockNum = 124564 AND SvcCode = 105;
19	**Deleting SERVICE** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Delete Service Record" button Prompt user to enter Invoice Number Display selected invoice Prompt to user: "Do you want to delete this Invoice?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Invoice?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Service WHERE StockNum = 124564 AND SvcCode = 105;
20	**Searching SERVICE** **Actor/User: General Manager/Dept Manager/Staff** **Steps:**	
		User clicks "Search Service Records" button Prompt user to enter search criteria Display new view
	SQL	SELECT SvcDate, CustNum, FROM Service WHERE Tech = 10;
21	**Entering new SERVICE CODE** **Actor/User: Dept Manager** **Steps:**	
		User clicks "New Service Code" button New Service Code is auto generated and displayed Prompt user to enter the following: Service Description List Price Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Service Code(SvcDesc, Price) VALUES('Oil Change', 49.95)

22	**Modifying SERVICE CODE** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Modify Service Code" button Prompt user to enter Service Code Display record for selected Service Code User clicks "Edit" button User modifies necessary attribute User clicks "Save" button
	SQL	UPDATE ServiceCode SET Price = 59.95 WHERE SvcCode = 101;
23	**Deleting SERVICE CODE** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Delete Service Code" button Prompt user to enter Service Code Display record for selected Service Code Prompt to user: "Do you want to delete this Service Code record?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Service Code record?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from ServiceCode WHERE SvcCode = 108;
24	**Searching SERVICE CODE** **Actor/User: General Manager/Dept Manager/Staff** **Steps:**	
		User clicks "Search Service Codes" button Prompt user to enter search criteria Display new view
	SQL	SELECT SvcCode, SvcDescription, Price FROM SvcCode;
25	**Entering new SALES TRANSACTION** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "New Transaction" button New Transaction ID and Transaction Date are auto generated and displayed Prompt user to enter the following: Stock Number Customer ID Sale Price Salesperson Bank Warranty Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into SaleTrans (StockNum, CustID, SalePrice, Salesperson, BankID, WarrID) VALUES(111234, 5678, 35000, 2, 601, 801) ;

26	**Modifying SALES TRANSACTION** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "Modify Transaction" button Prompt user to enter Transaction ID Display record for selected Transaction. User clicks "Edit" button User modifies necessary attribute User clicks "Save" button
	SQL	UPDATE SaleTrans SET SalePrice = 34500 WHERE TransID = 2424756;
27	**Deleting SALES TRANSACTION** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Delete Transaction" button
		Prompt user to enter Transaction Number Display staff record for selected Transaction. Prompt to user: "Do you want to delete this Transaction record?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Transaction record?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from SaleTrans WHERE TransID = 2424756;
28	**Searching SALES TRANSACTION** **Actor/User: General Manager/Dept Manager** **Steps:**	
		User clicks "Search Customers" button Prompt user to enter search criteria Display new view
	SQL	SELECT TransID FROM SaleTrans WHERE Price >= 30000;
29	**Entering new CUSTOMER** **Actor/User: Dept Manager/Staff/Customer** **Steps:**	
		User clicks "New Customer" button New Customer ID is auto generated and displayed Prompt user to enter the following: Last Name First Name Phone Email Salesperson Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Customer (LastName, FirstName, Phone, Email, Salesperson) VALUES(Harris, Emily, 2814482555, 3) ;

30	**Modifying CUSTOMER** **Actor/User: Dept Manager/Staff/Customer** **Steps:**	
		User clicks "Modify Customer" button Prompt user to enter Customer ID Display record for selected Customer User clicks "Edit" button User modifies necessary attribute(s) User clicks "Save" button
	SQL	UPDATE Customer SET EmpID = 6 WHERE Customer = 23;
31	**Deleting CUSTOMER** **Actor/User: Dept Manager/Staff** **Steps:**	
		User clicks "Delete Customer" button Prompt user to enter Customer ID Display record for selected Customer Prompt to user: "Do you want to delete this Customer record?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Customer record?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Customer WHERE CustID = 23;
32	**Searching CUSTOMER** **Actor/User: General Manager/Dept Manager/Staff** **Steps:**	
		User clicks "Search Customers" button Prompt user to enter search criteria Display new view
	SQL	SELECT * FROM Customer WHERE city = 'Houston';
33	**Entering new WARRANTY** **Actor/User: Dept Manager** **Steps:**	
		User clicks "New Warranty" button New Warranty Code is auto-generated and displayed Prompt user to enter the following: Warranty Type Term Price Display information for confirmation User clicks "Confirm" button
	SQL	INSERT into Warranty (WarrType, Term, Price) VALUES (Basic, 12, 1000) ;

34	**Modifying WARRANTY** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Modify Warranty" button Prompt user to enter WarrantyCode Display record for selected Warranty User clicks "Edit" button User modifies necessary attribute(s) User clicks "Save" button
	SQL	UPDATE Warranty SET Price = 1500 WHERE WarrID = 804;
35	**Deleting WARRANTY** **Actor/User: Dept Manager** **Steps:**	
		User clicks "Delete Warranty" button Prompt user to enter WarrantyCode Display record for selected Warranty Prompt to user: "Do you want to delete this Warranty?" IF User clicks "Confirm" button THEN Prompt to user: "Are you sure you want to delete this Warranty?" User clicks "Confirm" button ELSE User clicks "Cancel" button
	SQL	DELETE from Warranty WHERE WarrID = 803;
36	**Searching WARRANTY** **Actor/User: General Manager/Dept Manager/Staff** **Steps:**	
		User clicks "Search Warranties" button Prompt user to enter search criteria Display new view
	SQL	SELECT WarrID FROM Warranty WHERE Price >= 3000;
37	**View Transactions by Customer** **Actor/User: General Manager/Dept Manager/Staff** **Steps:**	
	SQL	SELECT c.LastName, c.FirstName, t.StockNum, t.SalePrice FROM Transaction t JOIN Customer c ON c.CustID = t.Customer ORDER BY c.LastName;

38 **View Transactions by Salesperson**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

```
SELECT
    s.LastName,
    s.FirstName,
    t.SalePrice
    t.WarrID
FROM Transaction t
JOIN Staff s
ON t.Salesperson = s.EmpID
ORDER BY s.LastName;
```

39 **View Service by Technician**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

```
SELECT
    s.LastName,
    s.FirstName,
    v.SvcCode
FROM Service v
JOIN Staff s
ON v.Technician =  s.EmpID
ORDER BY s.LastName;
```

40 **View Service by Vehicle Year**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

```
SELECT
    s.InvoiceNum
    s.SvcCode
    v.Make
    v.Model
    v.Year
FROM Vehicle v
JOIN Service s
ON v.StockNum =  s.StockNum
ORDER BY v.Year;
```

41 **View Vehicles by Location**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

```
SELECT
    l.LocID,
    l.LocName,
    v.StockNum
    v.Price
FROM Vehicle v
JOIN Location l
ON v.Location =  l.LocID
ORDER BY l.LocID;
```

42 **View Transactions by Warranty**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

```
SELECT
    t.TransID,
    t.Price,
    w.WarrID,
    w.WarrType,
    w.Price,
FROM Transaction t
JOIN Warranty w
ON t.Warranty = w.WarrID
ORDER BY s.LastName;
```

43 **View Transactions by Bank**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

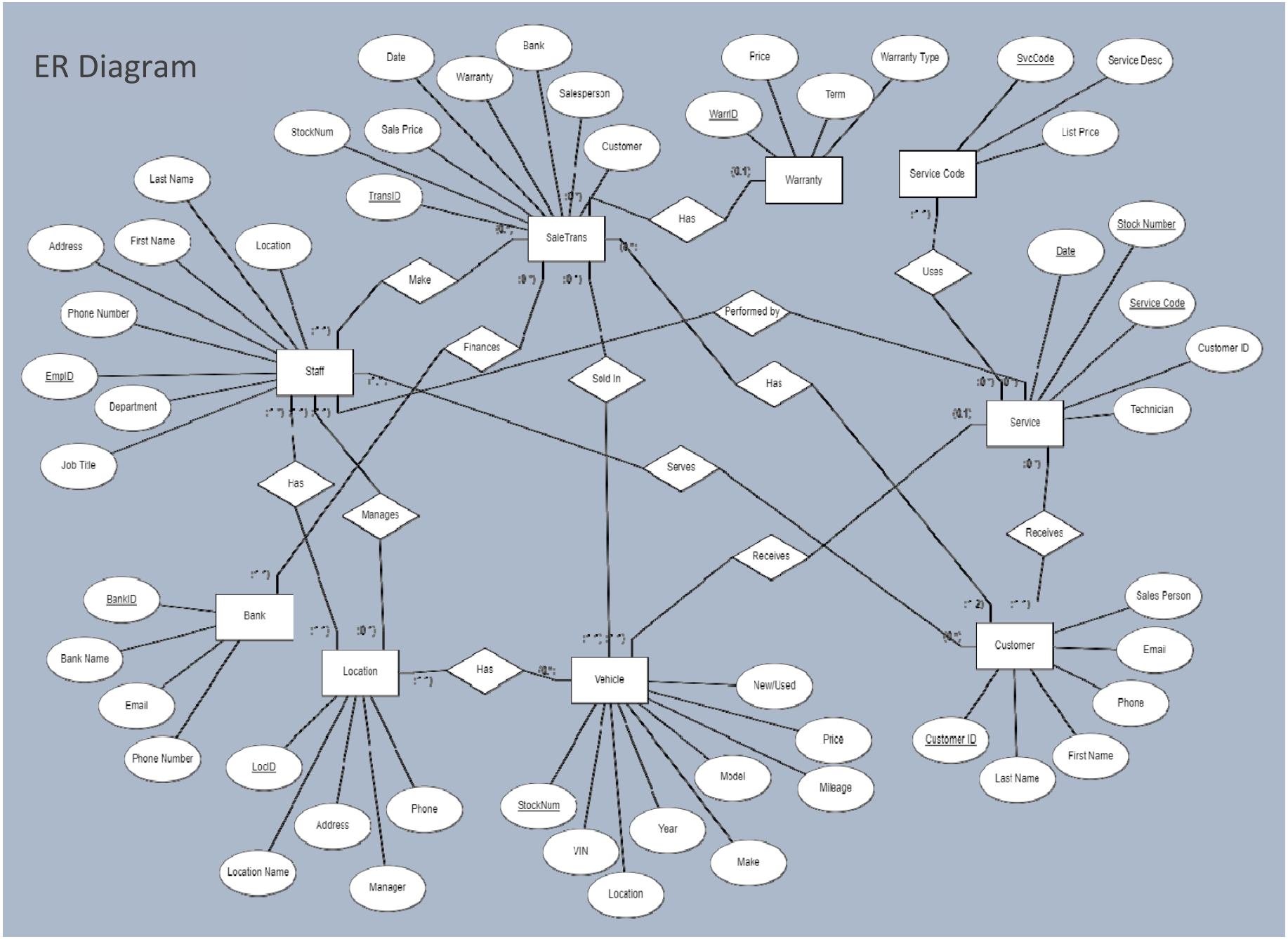
```
SELECT
    b.BankID,
    b.BankName,
    t.SalePrice,
FROM Transaction t
JOIN Bank b
ON  b.BankID =  t.Bank
ORDER BY b.BankName;
```

44 **View Service by Service Code**
Actor/User: General Manager/Dept Manager/Staff
Steps:

SQL

```
SELECT
    s.SvcDate
    s.SvcCode
    v.Price
FROM Service s
JOIN ServiceCode v
ON  s.SvcCode =  v.SvcCode
ORDER BY v.SvcCode;
```

ER Diagram


Date
Warranty
Bank
Salesperson
Customer
StockNum
Sale Price
TransID
SaleTrans
Has
Warranty
WarrID
Price
Term
Warranty Type
Service Code
SvcCode
Service Desc
List Price
Uses
Stock Number
Date
Service Code
Customer ID
Technician
Service
Performed by
Has
Sold In
Finances
Make
Staff
Last Name
First Name
Location
Address
Phone Number
EmpID
Department
Job Title
Has
Manages
Serves
Receives
Receives
Bank
BankID
Bank Name
Email
Phone Number
Location
LocID
Address
Phone
Location Name
Manager
Has
Vehicle
StockNum
VIN
Location
Year
Make
Model
Mileage
Price
New/Used
Customer
Customer ID
Last Name
First Name
Phone
Email
Sales Person

RELATIONAL MODELS AND NORMALIZATION

SERVICE CODE

```
CREATE TABLE ServiceCode (
        SvcCode INT (3),
        SvcDesc VARCHAR (15),
        Price DOUBLE PRECISION (6,2),
        PRIMARY KEY (SvcCode)
);
```

The primary key is SvcCode.

Functional Dependency:

SvcCode -> SvcDesc

SvcCode -> Price

This table is 1NF because each cell only has one data from the attribute domain.

This table is 2NF because SvcDesc and Price both fully depend on the primary key.

This table is 3 NF because Price does not depend on SvcDesc and SvcDesc does not depend on Price, so there is no transitive dependency.

This table is BCNF because it is 3NF and the primary key is a super key.

WARRANTY

```
CREATE TABLE Warranty (
        WarrID INT (3),
        WarrType VARCHAR (15),
        Term INT (2),
        Price INT (5),
        PRIMARY KEY (WarrID)
);
```

The primary key is WarrID.

Functional Dependency:

WarrID -> WarrType

WarrID -> Term

WarrID -> Price

This table is 1NF because each cell only has one data from the attribute domain.

This table is 2NF because WarrType, Term and Price all fully depend on the primary key.

This table is 3 NF because WarrType, Term and Price depend only on the primary key therefore there is no transitive dependency.

This table is BCNF because it is 3NF and the primary key is a super key.

RELATIONAL MODELS AND NORMALIZATION

BANK

```
CREATE TABLE Bank (
        BankID INT(3),
        BankName VARCHAR (15),
        Email VARCHAR (25),
        Phone DOUBLE (10,0),
        PRIMARY KEY (BankID)
);
```

The primary key is BankID.

Functional Dependency:

BankID -> BankName

BankID -> Email

BankID -> Phone

This table is 1NF because each cell only has one data from the attribute domain.

This table is 2NF because BankName, Email and Phone all fully depend on the primary key.

This table is 3 NF because BankName, Email and Phone depend only on the primary key therefore there is no transitive dependency.

This table is BCNF because it is 3NF and the primary key is a super key.

LOCATION

```
CREATE TABLE Location (
        LocID INT (2),
        LocName VARCHAR (10),
        StreetAddress VARCHAR (20),
        City VARCHAR (12),
        State VARCHAR (2),
        Zip INT (5),
        Phone DOUBLE (10,0),
        Manager INT (3),
        PRIMARY KEY (LocID)
);
```

The primary key is LocID.

Functional Dependency:

LocID -> LocName

LocID -> Address

LocID -> Phone

LocID -> Manager

This table is 1NF because each cell only has one data from the attribute domain.

This table is 2NF because LocName, Address, Phone and Manager all fully depend on the primary key.

This table is 3 NF because LocName, Address, Phone and Manager depend only on the primary key therefore there is no transitive dependency.

RELATIONAL MODELS AND NORMALIZATION	
VEHICLE	
CREATE TABLE Vehicle (StockNum INT(6), ManufYear INT(4), Make VARCHAR(10), Model VARCHAR(10), Color VARCHAR(10), Mileage INT(6), Price INT(6), Location INT(2), VIN VARCHAR(17), New_Used VARCHAR(4), PRIMARY KEY (StockNum), FOREIGN KEY (Location) REFERENCES Location (LocID));	The primary key is StockNum. Functional Dependency: StockNum -> ManufYear StockNum -> Make StockNum -> Model StockNum -> Mileage StockNum -> Location StockNum -> VIN StockNum -> New_Used This table is 1NF because each cell only has one data from the attribute domain. This table is 2NF because ManufYear, Make, Model, Mileage, Location, VIN and New_Used all fully depend on the primary key. This table is 3NF because ManufYear, Make, Model, Mileage, Location, VIN and New_Used depend only on the primary key therefore there is no transitive dependency. This table is BCNF because it is 3NF and the primary key is a super key.

RELATIONAL MODELS AND NORMALIZATION

STAFF

```
CREATE TABLE Staff (
        EmpID INT (4),
        LastName VARCHAR (15),
        FirstName VARCHAR (15),
        StreetAddress VARCHAR (20),
        City VARCHAR (12),
        State VARCHAR (2),
        Zip INT (5),
        Phone DOUBLE (10),
        Location INT (2),
        Department VARCHAR (12),
        JobTitle VARCHAR (12),
        PRIMARY KEY (EmpID),
        FOREIGN KEY (Location) REFERENCES Location (LocID)
);
```

The primary key is EmpID.

Functional Dependency:

EmpID -> LastName

EmpID -> FirstName

EmpID -> Address

EmpID -> Phone

EmpID -> Location

EmpID -> JobTitle

This table is 1NF because each cell only has one data from the attribute domain.

This table is 2NF because all non-primary key attributes are fully depend on the primary key.

This table is 3 NF because all non-primary attributes depend only on the primary key therefore there is no transitive dependency.

This table is BCNF because it is 3NF and the primary key is a super key.

CUSTOMER

```
CREATE TABLE Customer (
        CustID INT (7),
        LastName VARCHAR (15),
        FirstName VARCHAR (10),
        Phone DOUBLE (10),
        Email VARCHAR (25),
        Salesperson INT (4),
        PRIMARY KEY (CustID),
        FOREIGN KEY (Salesperson) REFERENCES Staff (EmpID)
);
```

The primary key is CustID.

Functional Dependency:

CustID -> FirstName

CustID -> LastName

CustID -> Email

CustID -> Phone

This table is 1NF because each cell only has one data from the attribute domain.

This table is 2NF because FirstName, LastName, Email and Phone all fully depend on the primary key.

This table is 3 NF because FirstName, LastName, Email and Phone depend only on the primary key therefore there is no transitive dependency.

This table is BCNF because it is 3NF and the primary key is a super key.

RELATIONAL MODELS AND NORMALIZATION	
SALE TRANSACTION	
CREATE TABLE SaleTrans (TransID INT (9), TransDate DATE, StockNum INT (6), Customer INT (7), SalePrice DOUBLE PRECISION (8,2), Salesperson INT (4), Bank INT (3), Warranty INT (3), PRIMARY KEY (TransID), FOREIGN KEY (StockNum) REFERENCES Vehicle (StockNum), FOREIGN KEY (Customer) REFERENCES Customer (CustID), FOREIGN KEY (Salesperson) REFERENCES Staff (EmpID), FOREIGN KEY (Bank) REFERENCES Bank (BankID), FOREIGN KEY (Warranty) REFERENCES Warranty (WarrID));	The primary key is TransID. Functional Dependency: TransID -> TransDate TransID -> StockNum TransID -> Customer TransID -> Salesperson TransID -> Bank TransID -> Warranty This table is 1NF because each cell only has one data from the attribute domain. This table is 2NF because all non-primary keys attributes are fully dependent on the primary key. This table is 3 NF because all non-primary attributes depend only on the primary key therefore there is no transitive dependency. This table is BCNF because it is 3NF and the primary key is a super key.

RELATIONAL MODELS AND NORMALIZATION

SERVICE RECORD

```
CREATE TABLE Service (
        SvcDate DATE,
        StockNum INT (6),
        ServiceCode INT (3),
        Customer INT (7),
        Technician INT (4),
        PRIMARY KEY (SvcDate, StockNum, ServiceCode),
        FOREIGN KEY (ServiceCode) REFERENCES ServiceCode (SvcCode),
        FOREIGN KEY (Customer) REFERENCES Customer (CustID),
        FOREIGN KEY (StockNum) REFERENCES Vehicle (StockNum),
        FOREIGN KEY (Technician) REFERENCES Staff (EmpID)
);
```

This table has a composite primary key of Date + StockNum + SvcCode.

Functional Dependency:

Date + StockNum + SvcCode -> Customer

Date + StockNum + SvcCode -> Technician

100101 in StockNum is associated with Customers 0 2, therefore Customer does not depend on StockNum.

100102 in StockNum is associated with Technicians 1003 and 1004, therefore Technician does not depend on StockNum.

105 in ServiceCode is associated with Customers 1 and 0, therefore Customer does not depend on ServiceCode.

105 in ServiceCode is associated with Technicians 1003 and 1004, therefore Technician does not depend on ServiceCode.

3-2-2020 in SvcDate is associated with Customers 1 and 3, therefore Customer does not depend on SvcCode.

3-a2-2020 in SvcDate is associated with Technicians 1003 and 1004, therefore Technician does not depend on ServiceCode.

This table is 2NF because Customer and Technician depend on the full primary key.

This table is 3NF because Customer and Technician depend only on the full primary key therefore there is no transitive dependency.

This table is BCNF because it is 3NF and the primary key of Date + StockNum + SvcCode is a super key.

TABLE DATA

Location

LocID	LocName	StreetAddress	City	State	Zip	Phone	Manager
10	Main	100 Main St	Houston	TX	77001	7132551000	1002
11	Stafford	3190 Ash Ave	Stafford	TX	77598	2818240090	1006
12	SciCtr	3270 Science Center	Houston	TX	77018	7135771000	1005
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Staff

EmpID	LastName	FirstName	StreetAddress	City	State	Zip	Phone	Location	Department	JobTitle
1000	Barryman	John	201 S. First St	Houston	TX	77223	2815559021	12	Sales	Salesperson
1001	Natarajan	Suresh	17201 Greenleaf Ct	Houston	TX	77495	2812775448	11	Sales	Salesperson
1002	MacKenzie	Sara	15201 Jameson St	Houston	TX	77032	2814445298	10	Sales	Manager
1003	Bombadil	Tom	675 McGowan St	Houston	TX	77006	7135269875	10	Service	Technician
1004	Davis	Calvin	661 Watson Street	Houston	TX	77106	2816966155	11	Service	Technician
1005	Kinard	Dona	1947 Arbor Court	Houston	TX	77055	7132888944	12	Operations	Manager
1006	Butler	John	120 George St	Houston	TX	77041	2815244698	11	Operations	Manager
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customer

CustID	LastName	FirstName	Phone	Email	Salesperson
0	House	In	NULL	NULL	NULL
1	Stover	Erica	7135788952	EricaIStover@teleworm.us	1001
2	Corry	Lani	9158317139	LaniWCorry@jourrapide.com	1001
3	Philpot	Thomas	2814908181	RonaldLPark@yahoo.com	1002
4	Park	Ronald	7133125488	rpark2378@gmail.com	1001
5	Douglas	Dennis	7175788952	supersailor12@gmail.com	1002
NULL	NULL	NULL	NULL	NULL	NULL

SaleTrans

TransID	TransDate	StockNum	Customer	SalePrice	Salesperson	Bank	Warranty
100100100	2019-11-19 00:00:00	100102	3	21500.00	1000	601	801
100100101	2019-10-20 00:00:00	100101	2	20000.00	1001	602	801
100100102	2019-09-19 00:00:00	100100	1	25000.00	1000	601	800
100100103	2019-12-19 00:00:00	100103	5	32750.00	1002	603	801
100100104	2019-10-25 00:00:00	100104	4	31000.00	1002	604	803
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

AGGREGATE QUERIES

City	Number Locations per City
Houston	2
Stafford	1

Location	Number of Employees by Location
10	2
11	3
12	2

Number of Customers
6

Average Car Sale
$26,050.00

TABLE DATA	AGGREGATE QUERIES

ServiceCode

SvcCode	SvcDesc	Price
100	Oil Change	43.95
101	Brake Pads	79.95
102	Batt Svc	109.95
103	Timing Belt	1299.95
105	Radiator Flush	229.95
NULL	NULL	NULL

Number of Available Services Under $100
2

Vehicle

StockNum	ManufYear	Make	Model	Color	Mileage	Price	Location	VIN	New_Used
100100	2020	Honda	Accord	Silver	31598	29995	10	JHMZE2H74BS003849	Used
100101	2020	Toyota	Rav4	Blue	38662	21995	10	5TELU4EN6AZ725438	Used
100102	2019	Toyota	Corolla	Silver	38662	21995	11	JTMRF4DV0CD045813	Used
100103	2018	Toyota	Tundra	Black	68001	32995	11	JT2EL32H8K0425561	Used
100104	2022	Toyota	Camry	Red	5	31596	12	JTEGD21A760143408	New
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Average Price
$27,715.20

Warranty

WarrID	WarrType	Term	Price
800	Basic	12	1500
801	Intermediate	24	2000
802	Full	36	2850
803	Luxury	48	3500
NULL	NULL	NULL	NULL

Warranty Price Range
$1500 - $3500

Bank

BankID	BankName	Email	Phone
600	Bank of America	manager@bofa.com	2125554000
601	Bank of America	manager@bofa.com	2125554000
602	Comerica Bank	mgr@comericabank.com	7135655020
603	TDECU	manager@tdecu.com	2815551234
604	Chase Bank	manager@chase.com	8885551212
605	Citibank	manager@citibank.com	8005556060
NULL	NULL	NULL	NULL

Number of Partner Banks
6

TABLE DATA	AGGREGATE QUERIES
Service	

svcdate	StockNum	ServiceCode	Customer	Technician
2020-01-27 00:00:00	100102	105	1	1003
2020-02-16 00:00:00	100101	103	2	1004
2020-03-02 00:00:00	100102	101	1	1004
2020-03-02 00:00:00	100102	102	1	1003
2020-04-12 00:00:00	100103	102	4	1004
2021-06-14 00:00:00	100101	100	0	1004
2021-06-14 00:00:00	100101	105	0	1004
2021-07-06 00:00:00	100102	100	3	1003
2021-08-17 00:00:00	100101	100	0	1004
NULL	NULL	NULL	NULL	NULL

Number Services, by Tech	Technician
3	1003
6	1004

JOINT QUERIES

Service, Customer, Vehicle, ServiceCode

Date	Customer	Vehicle	Service Performed
02/16/2020	Lani Corry	2020 Toyota Rav4	Timing Belt
06/14/2021	In House	2020 Toyota Rav4	Oil Change
08/17/2021	In House	2020 Toyota Rav4	Oil Change
06/14/2021	In House	2020 Toyota Rav4	Radiator Flush
04/12/2020	Ronald Park	2018 Toyota Tundra	Batt Svc
07/06/2021	Thomas Philpot	2019 Toyota Corolla	Oil Change
03/02/2020	Erica Stover	2019 Toyota Corolla	Brake Pads
03/02/2020	Erica Stover	2019 Toyota Corolla	Batt Svc
01/27/2020	Erica Stover	2019 Toyota Corolla	Radiator Flush

SaleTrans, Customer, Vehicle, Staff, Waranty, Bank

Date	Customer	Vehicle	Sale Price	Salesperson	Warranty	Financing Bank
09/19/2019	Erica Stover	2020 Honda Accord	$25,000.00	John Barryman	Basic 12 month	Bank of America
11/19/2019	Thomas Philpot	2019 Toyota Corolla	$21,500.00	John Barryman	Intermediate 24 month	Bank of America
12/19/2019	Dennis Douglas	2018 Toyota Tundra	$32,750.00	Sara MacKenzie	Intermediate 24 month	TDECU
10/25/2019	Ronald Park	2022 Toyota Camry	$31,000.00	Sara MacKenzie	Luxury 48 month	Chase Bank
10/20/2019	Lani Corry	2020 Toyota Rav4	$20,000.00	Suresh Natarajan	Intermediate 24 month	Comerica Bank

Customer, Staff

Salesperson	Number of Customers by Salesperson
Suresh Natarajan	3
Sara MacKenzie	2

Vehicle, Location

Stock Number	Vehicle	Price	Location
100100	2020 Honda Accord	$29,995.00	Main
100101	2020 Toyota Rav4	$21,995.00	Main
100102	2019 Toyota Corolla	$21,995.00	Stafford
100103	2018 Toyota Tundra	$32,995.00	Stafford
100104	2022 Toyota Camry	$31,596.00	SciCtr

JOINT QUERIES

Staff, Location

Employee	Location
Sara MacKenzie	Main
Tom Bombadil	Main
Suresh Natarajan	Stafford
Calvin Davis	Stafford
John Butler	Stafford
John Barryman	SciCtr
Dona Kinard	SciCtr

Location, Staff (Manager)

Location	Manager
Main	Sara MacKenzie
Stafford	John Butler
SciCtr	Dona Kinard

The database works as designed and is scalable to be able to add entities as needed for other departments, such as human resources and vehicle acquisitions. This would support the entire company, as opposed to merely sales and service.

CONCLUSION

*No references since data was made up, not sourced.