Fall 2019

Brendan Ellison

Final Project CIS276DB

Putting it all together

Brendan

Contents

[Definition of the problem 2](#_Toc25152525)

[Develop a plan to solve the problem 2](#_Toc25152526)

[Collect and Analyze the appropriate information to create the database 3](#_Toc25152527)

[Interpret findings to determine plausible solutions for normalization 3](#_Toc25152528)

[Evaluate the effectiveness of the reports and the results obtained 7](#_Toc25152529)

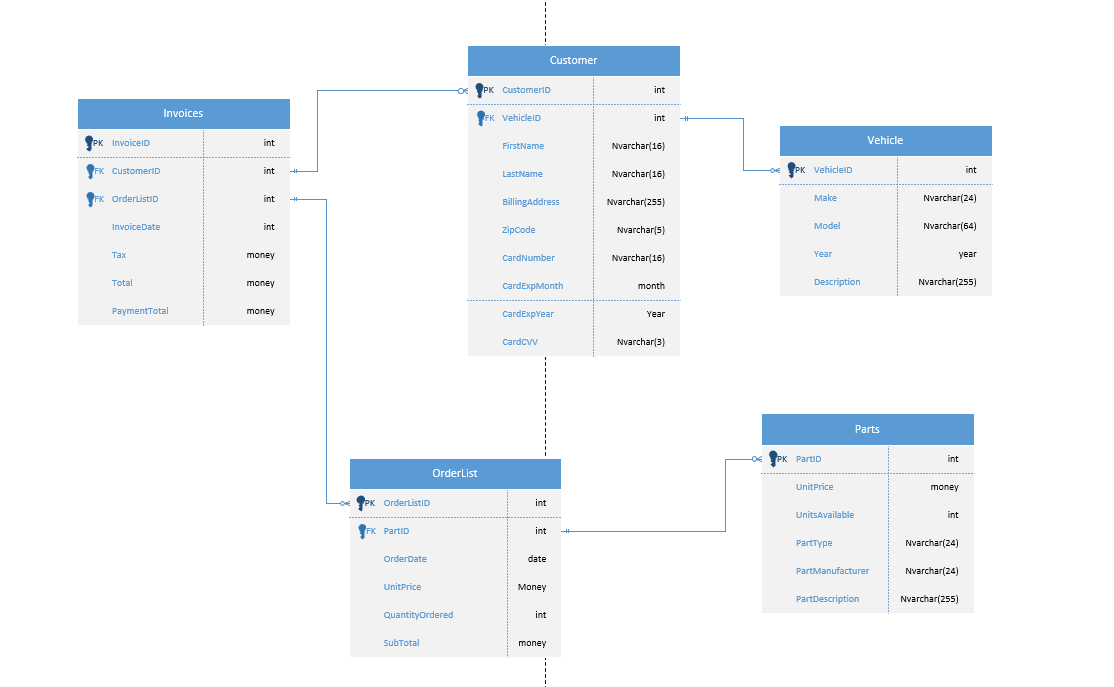
[Communicate the Results 11](#_Toc25152530)

# Definition of the problem

I have been asked to make a database for an auto repair shop to help them manage their business

# Develop a plan to solve the problem

I will make a database that stores data about the customer, their vehicle, details about parts ordered for their vehicle, and capable of getting invoice information.



# Collect and Analyze the appropriate information to create the database

/\*

Author: Brendan Ellison

Date: 11/23/2021

This can be executed all at once but I want to keep this consolidated,

so if anything is added it should be highlighted then executed

\*/

USE MASTER

GO

IF DB\_ID('BrendansAutoShop') IS NOT NULL

DROP DATABASE BrendansAutoShop;

GO

CREATE DATABASE BrendansAutoShop

GO

USE BrendansAutoShop

GO

CREATE TABLE Vehicle

(

VehicleID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Producer NVARCHAR(32) NOT NULL,

Model NVARCHAR(64) NOT NULL,

Year SMALLINT NULL, -- I feel that not everyone remembers their model year

Description NVARCHAR(255) NOT NULL, -- Must have a description of the work needed to repair

CHECK (Producer = 'Ford' OR Producer = 'Honda' OR Producer = 'Toyota' OR Producer = 'Volkswagen' OR Producer = 'Chevrolet' OR Producer = 'GM' OR Producer = 'BMW' OR Producer = 'Hyundai' OR Producer = 'Nissan' OR Producer = 'Volvo' OR Producer = 'Other')

--The Check is to avoid bad entries/user error/stupid entries, it is more of a concept then a complete list

--I did not add a check for models because that list would be too big and not really add much

);

GO

CREATE TABLE Parts

(

PartID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

UnitsAvailable INT NOT NULL, -- Storage/Spares

PartType NVARCHAR(24) NOT NULL, -- Alternator, Battery, Cylinder, etc...

PartManufacturer NVARCHAR(24) NOT NULL, -- I think it is useful to know reliable sellers

PartDescription NVARCHAR(255) NULL, -- Encouraged but not required

GO

CREATE TABLE Customer

(

CustomerID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

VehicleID INT NOT NULL REFERENCES Vehicle (VehicleID),

FirstName NVARCHAR(32) NOT NULL,

LastName NVARCHAR(32) NOT NULL,

BillingAddress NVARCHAR(255) NOT NULL,

ZipCode NVARCHAR(5) NOT NULL,

CardNumber NVARCHAR(16) NOT NULL,

CardExpMonth SMALLINT NOT NULL,

CardExpYear SMALLINT NOT NULL,

CardCVV NVARCHAR(3) NOT NULL

);

GO

CREATE TABLE OrderList

(

OrderListID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

CustomerID INT NOT NULL REFERENCES Customer (CustomerID),

PartID INT NOT NULL REFERENCES Parts (PartID),

OrderDate DATE NOT NULL,

UnitPrice MONEY NOT NULL,

QuantityOrdered INT NOT NULL,

OrderTotal AS UnitPrice \* QuantityOrdered

);

GO

-- DO NOT have the orders tied to the invoices, input with a query

CREATE TABLE Invoices

(

InvoiceID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

CustomerID INT NOT NULL REFERENCES Customer (CustomerID),

SubTotal MONEY NULL,

Tax AS SubTotal \* 0.08, --I think tax is 8%...

Service AS SubTotal \* 0.15,

Total AS SubTotal \* 1.23,

PaymentTotal MONEY NULL DEFAULT 0

CONSTRAINT Payment\_Validator CHECK (PaymentTotal >= 0) --AND (PaymentTotal <= Total)) Does not like the computed column in here

);

GO

The data that I made up to add to the tables goes as follows…

/\*

Author: Brendan Ellison

Date: 11/25/2021

\*/

USE BrendansAutoShop

GO

INSERT INTO Vehicle (Producer, Model, Year, Description)

VALUES ('Toyota', 'Tacoma', 2014, 'Owner had his catalytic converter stolen and needs it replaced as well as a protective strap'),--True story btw

('Chevrolet', 'Malibu', 2014,'Car needs a replacement secondary battery in the trunk, car runs fine otherwise'),

('Ford','F-250', 2015,'Nail found in the tire, needs a new wheel'),

('Chevrolet','Silverado 2500HD',2021,'Needs an oil change'),

('Honda','Civic',2016,'Needs new brake pads and ac filter'),

('Ford','F-150',2020,'Alternator needs to be replaced and battery as well'),

('BMW','328i',2011,'Thermostate needs to be replaced'),

('Nissan','NV Cargo',2021,'Both side mirrors missing, needs replacing'),

('Chevrolet','Malibu',2014,'Brake lights stopped working');

GO

SELECT \*

FROM Vehicle

GO

INSERT INTO Customer (VehicleID, FirstName, LastName, BillingAddress, ZipCode, CardNumber, CardExpMonth, CardExpYear, CardCVV)

VALUES (1,'Brendan','Ellison','2220 E Beardsley Road', '85024','9330093598667675',04,23,123),

(2,'Marianne','Ellison','2220 E Beardsley Road','85024','6990504036642041',12,25,596),

(3,'Skye','Berry','105 New Saddle Street Stroudsburg','43017','4064645536577681',05,24,670),

(4,'Quentin','Smith','104 S. Primrose Street Murfreesboro','22468','8388647097946058',07,27,099),

(5,'Greyson','Smith','726 Warren Ave. South Richmond Hill','55582','1664849122693541',09,29,433),

(6,'Jack','Neal','417 Walnut Street Saint Albans','53158','3906897658221940',11,21,888),

(7,'Cheyenne','Cowan','118 Devonshire St. Rahway','44155','1157808329225578',01,26,671),

(8,'John','Lee','9557 Carriage Ave. Tualatin','95809','0515829402171149',04,24,263),

(9,'Wesley','Mcdaniel','9528 Halifax Drive Wantagh','37821','8528535821184297',02,27,397);

GO

SELECT \*

FROM Customer

GO

INSERT INTO Parts (UnitsAvailable, PartType, PartManufacturer, PartDescription)

VALUES (0,'Catalytic Converter','ExtremeTerrain.com','Reduces emmissions and sound from truck engine'),

(0,'Strap','getcatsecurity.com','A big metal plate that covers parts from being stole off the undercarriage'),

(1,'Auxiliary Battery','AutoZone','A secondary battery found in some cars'),

(2,'Tire','Discount Tire','Large truck tire, 18"'),

(2,'Synthetic Oil','Amsoil','Synthetic Oil for mid sized truck'),

(0,'Brake Pad','BrakeProformance','Brake pads for 2015 honda civic'),

(5,'AC Filter','Amazon.com','Replacment air filter'),

(0,'Alternator','Amazon.com','Alternator for Ford vehicles'),

(1,'Battery','Advanced Auto Parts','Battery for a mid sized truck, Size: 34'),

(0,'Thermostat','FCP Euro','Thermostat used for BMW'),

(0,'Side Mirror','1AAuto.com','Side mirror for NV Cargo'),

(4,'Brake Light','Amazon.com','2013-2015 Malibu brake light');

GO

SELECT \*

FROM Parts

GO

INSERT INTO OrderList (CustomerID, PartID, OrderDate, UnitPrice, QuantityOrdered)

VALUES (1,1,'2021-10-20',650.99,1),

(1,2,'2021-10-20',299.99,1),

(2,3,'2021-09-10',92.99,1),

(3,4,'2021-09-26',283,1),

(4,5,'2021-11-11',55.29,1),

(5,6,'2021-11-14',78.40,4),

(5,7,'2021-11-14',15.76,1),

(6,8,'2021-10-29',72.95,1),

(6,9,'2021-10-29',201.99,1),

(7,10,'2021-11-10',58.51,1),

(8,11,'2021-12-01',35.95,2),

(9,12,'2021-11-04',98.98,2);

GO

SELECT \*

FROM OrderList

GO

INSERT INTO Invoices (CustomerID, SubTotal)

SELECT CustomerID, SUM(OrderTotal)

FROM OrderList

GROUP BY CustomerID

GO

SELECT \*

FROM Invoices

GO

-- These are just to simulate some payments

UPDATE Invoices

SET PaymentTotal = 1169.71

WHERE CustomerID = 1

UPDATE Invoices

SET PaymentTotal = 114.38

WHERE CustomerID = 2

UPDATE Invoices

SET PaymentTotal = 338.18

WHERE CustomerID = 6

UPDATE Invoices

SET PaymentTotal = 88.44

WHERE CustomerID = 8

UPDATE Invoices

SET PaymentTotal = 200

WHERE CustomerID = 5

UPDATE Invoices

SET PaymentTotal = 100

WHERE CustomerID = 9

SELECT \*

FROM Invoices

GO

# Interpret findings to determine plausible solutions for normalization

Ill be honest, I am not 100% sure what normalized means but there are no duplicate tables, and the only key that does not get used is InvoiceID. I think overtime it would be realistic to assume more data will be added and structure could be changed to better fit the new data.

# Evaluate the effectiveness of the reports and the results obtained

I think for as simple as this database is, the reports I made get the job done.

View\_FindDebt is a simple view that is to show Customers that owe money.

/\*

Author: Brendan Ellison

Date: 12/2/21

Objective: Simple view to find customers with debt

\*/

USE BrendansAutoShop

GO

DROP VIEW IF EXISTS View\_FindDebt

GO

CREATE VIEW View\_FindDebt AS

SELECT FirstName, LastName, -- Full Name (I hate doing Last, First)

Year, Model, -- The Customers Vehicle

FORMAT(Total, 'C') AS [Total Due], FORMAT(PaymentTotal, 'C') AS [Ammount Paid],

FORMAT((Total - PaymentTotal), 'C') AS [Ammount Due] -- Useful Payment Info

FROM Invoices JOIN Customer

ON Invoices.CustomerID = Customer.CustomerID

JOIN Vehicle

ON Customer.VehicleID = Vehicle.VehicleID

WHERE PaymentTotal < Total

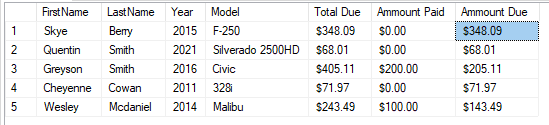
GO

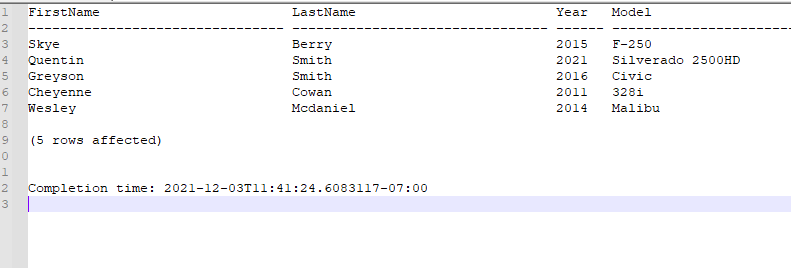
SELECT \*

FROM View\_FindDebt

GO

The results:





The second View I made for use in Procedures:

/\*

Author: Brendan Ellison

Date: 12/2/21

Objective: View to find customer Info

\*/

USE BrendansAutoShop

GO

DROP VIEW IF EXISTS View\_FindCustomer

GO

CREATE VIEW View\_FindCustomer AS

SELECT CustomerID,

FirstName, LastName,

BillingAddress, ZipCode,

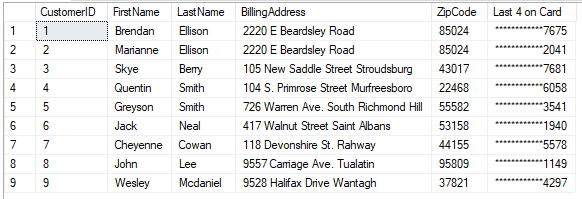
'\*\*\*\*\*\*\*\*\*\*\*\*' + RIGHT(CardNumber,4) AS [Last 4 on Card] -- For verification if needed

FROM Customer

GO

SELECT \* FROM View\_FindCustomer GO

The Results:





I made two procedures for this view, the first one being:

/\*

Author: Brendan Ellison

Date: 12/2/21

Objective: Procedure to find info on customer

\*/

USE BrendansAutoShop

GO

DROP PROCEDURE IF EXISTS GetCustomerInfo\_ID

GO

CREATE PROCEDURE GetCustomerInfo\_ID

@CustomerID INT -- I just want the meaning to be very clear

AS

SELECT \*

FROM View\_FindCustomer

WHERE CustomerID = @CustomerID

GO

EXEC GetCustomerInfo\_ID @CustomerID = 1

GO

The Results:



The second Procedure I made was the same thing but with customer name.

You can see they get the same result just in a different way:

/\*

Author: Brendan Ellison

Date: 12/2/21

Objective: Procedure to find info on customer With First and Last name

I feel in a larger Database this is a more realistic way of getting info,

this would be easy to set up in an application

\*/

USE BrendansAutoShop

GO

DROP PROCEDURE IF EXISTS GetCustomerInfo\_Name

GO

CREATE PROCEDURE GetCustomerInfo\_Name

@CustomerFName NVARCHAR(32),

@CustomerLName NVARCHAR(32)

AS

SELECT \*

FROM View\_FindCustomer

WHERE FirstName = @CustomerFName AND

LastName = @CustomerLName

GO

EXEC GetCustomerInfo\_Name @CustomerFName = 'Brendan', @CustomerLName = 'Ellison'

The Results:



I made one final procedure:

/\*

Author: Brendan Ellison

Date: 12/3/21

Objective: Procedure to insert into vehicle table via app

\*/

USE BrendansAutoShop

GO

DROP PROC IF EXISTS Proc\_AddVehicle

GO

CREATE PROC Proc\_AddVehicle

@Producer NVARCHAR(32),

@Model NVARCHAR(64),

@Year SMALLINT NULL,

@Description NVARCHAR(255)

AS

INSERT INTO Vehicle (Producer, Model, Year, Description)

VALUES (@Producer, @Model, @Year, @Description)

GO

SELECT \* FROM Vehicle

GO

EXEC Proc\_AddVehicle 'Ford','F-250',NULL,'Truck has a flat tire'

GO

SELECT \* FROM Vehicle

GO

--Clean up

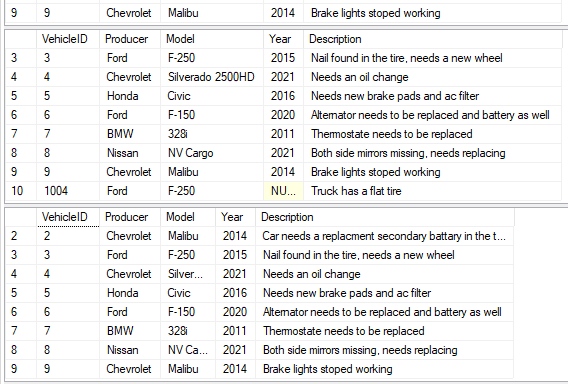
DELETE FROM Vehicle WHERE VehicleID > 9;

GO

SELECT \* FROM Vehicle

GO

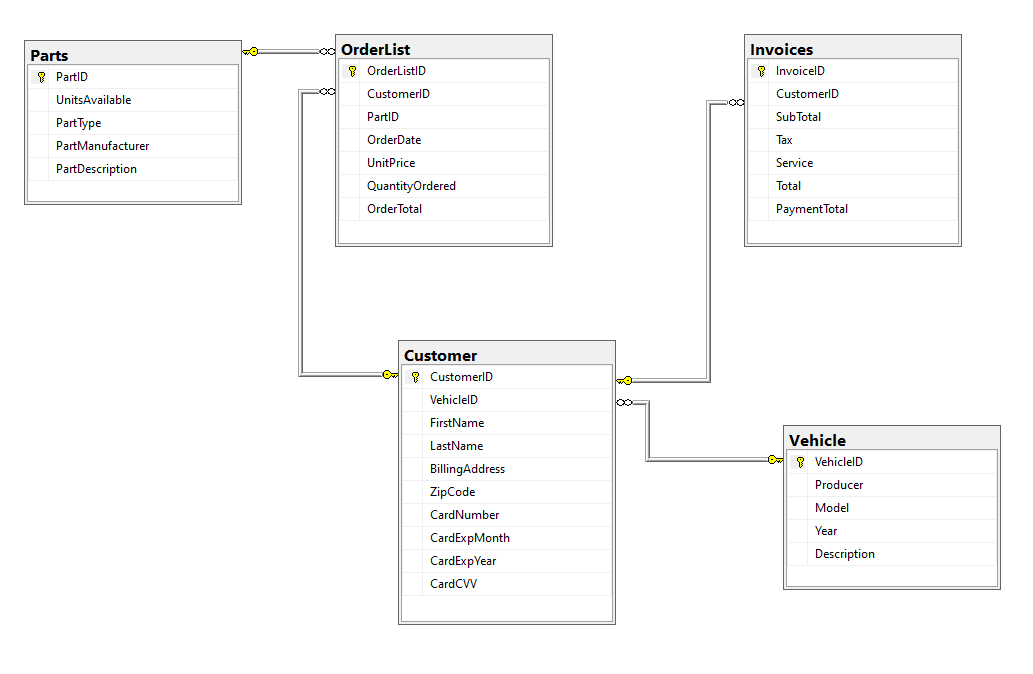
The Results:



# Communicate the Results

I think I learned a lot doing this project and I would like to learn more about making database. I hope that I get that chance out in the field, but I have no idea what is really waiting for me out there. Overall, I learned how to make a database, a basic understanding of database structure, how to run queries, how to use said quires, how and why to make procedures, as well as some other details.

The final result of by database structure:



The .sql and .rpt files hopefully in order: