IST 659

Doggie Lifestyle

March 2021

Overview

A few years ago, we welcomed a new pup in our family named Boston. Boston is a medium-standard American Bulldog. American Bulldogs are a well-balanced athletic dog that demonstrate great strength, endurance, agility, and a friendly attitude. Their average weight ranges from 66-130 lbs. American bulldogs shed a lot due to his short haired breed. Dogs shed heavily twice a year. This shedding is triggered by light more than temperature. As the days get longer, your dog will start to shed, even though the weather may still be very cold outside. Whenever we would wash Boston, multiple towels were used to dry him, and his hair would get all over the towel and re-washing towels is a pain even with the de-shedding detergent. After speaking to multiple pet owners, I realized I am not the only one with this dilemma. Washing your dog can be time-consuming and a challenging task which is not always enjoyable for some dogs/dog parents. There are already doggy robes out there for such dogs, such as small and medium dogs. For large/extra-large dogs, there are not a lot of choices. Thus, came the idea of Doggie Lifestyle. Doggie robes for all sized dogs.

Goal

The primary objective of Doggie Lifestyle is to pamper your dog with a soft robe after a bath without the hassle of using multiple towels to dry your dog. In 2019, $95.7 billion was spent on our pets in the U.S. alone. According to the 2019-2020 APPA National Pet Owners Survey, 67% of U.S. households own a pet, dogs equate to 63.4 million homes[1]. Doggie Lifestyle robes are sold online, and the fabric used is absorbent/shed control. After a dog wash all the owner or groomer would have to do is put the robe on the pup and when the dog is dry. Air out the robe, and when the robe is completely dry. The dog's hair will fall right off for easy disposal.

EXECUTABLE CODE -- DATABASE CREATION

```
-- drop all database objects in reverse order of their dependencies
-- including views, stored procedures and tables; note the use of
-- OBJECT_ID to detect if the object exists and conditionally deletes
-- existing objects only. The 2nd parameter to OBJECT_ID is the object
-- type - P for procedure, V for view, U for table


-- drop all procedures
IF Object_ID ('dbo.InsertProduct', 'P') IS NOT NULL DROP PROCEDURE IF EXISTS
dbo.InsertProduct;
go

IF Object_ID ('dbo.InsertPayment', 'P') IS NOT NULL DROP PROCEDURE IF EXISTS
dbo.InsertPayment;
go

IF Object_ID ('dbo.InsertCustomer', 'P') IS NOT NULL DROP PROCEDURE IF EXISTS
dbo.InsertCustomer;
go

-- drop all views
IF OBJECT_ID ('dbo.CustomerPayment', 'V') IS NOT NULL DROP VIEW IF EXISTS
dbo.CustomerPayment;
go
--DROP VIEW IF EXISTS dbo.CustomerPayment;
go
```

[1] "Pet Industry Market Size and Ownership Statistics."*APPA,* Feb. 2021, *https://www.americanpetproducts.org/press_industrytrends.asp*.

```
-- drop all tables in reverse order of their dependencies
IF OBJECT_ID ('dbo.OrderDetail', 'U') IS NOT NULL DROP TABLE IF EXISTS
dbo.OrderDetail;
go

IF OBJECT_ID ('dbo.CustomerOrder', 'U') IS NOT NULL DROP TABLE IF EXISTS
dbo.CustomerOrder;
go

IF OBJECT_ID ('dbo.Employee', 'U') IS NOT NULL DROP TABLE IF EXISTS dbo.Employee;
go

IF OBJECT_ID ('dbo.Supplier', 'U') IS NOT NULL DROP TABLE IF EXISTS dbo.Supplier;
go

IF OBJECT_ID ('dbo.Product', 'U') IS NOT NULL DROP TABLE IF EXISTS dbo.Product;
go

IF OBJECT_ID ('dbo.Payment', 'U') IS NOT NULL DROP TABLE IF EXISTS dbo.Payment;
go

IF OBJECT_ID ('dbo.Customer', 'U') IS NOT NULL DROP TABLE IF EXISTS dbo.Customer;
go

-- create all tables in order of their dependencies

CREATE TABLE dbo.Customer
(
	CustomerID int Not Null IDENTITY,
	FirstName varchar(50) NOT NULL,
	LastName varchar(50) NOT NULL,
	ShipAddress varchar(255) NOT NULL,
	ShipCity varchar(30) NOT NULL,
	ShipState varchar(30) NOT NULL,
	ShipPostal varchar(30) NOT NULL,
	Email varchar(255) Null,
	CreditCard varchar(255) Null,
	CardExp int Null,
	DogSize varchar(30) Null,
	CONSTRAINT PK_Customer Primary Key (CustomerID),
	CONSTRAINT U1_Customer Unique(LastName)
);
go

CREATE TABLE dbo.Payment
```

```
(
	PaymentID int NOT Null IDENTITY,
	PaymentType varchar(50) Not Null,
	PaymentDate date Null,
	CustomerID int NOT Null,
	Allowed int Null,
	CONSTRAINT PK_Payment Primary Key (PaymentID),
	CONSTRAINT FK1_Payment FOREIGN KEY (CustomerID) references Customer(CustomerID)
);
go

CREATE TABLE dbo.Product
(
	ProductID int NOT NUll IDENTITY,
	ProductName varchar(50) NOT NULL,
	ProductDescription varchar(255) NOT Null,
	Fabric varchar(50) Null,
	Color varchar(50) Null,
	Size varchar(50) Null,
	CONSTRAINT PK_Product Primary Key (ProductID),
	CONSTRAINT U1_Product Unique(ProductName)
);
go

CREATE TABLE dbo.Supplier
(
	SupplierID int Not Null IDENTITY,
	CompanyName varchar(50) Not Null,
	ProductID int NOT Null,
	SupplierAddress varchar(255) Null,
	CONSTRAINT PK_Supplier Primary Key (SupplierID),
	CONSTRAINT FK1_Supplier FOREIGN KEY (ProductID) references dbo.Product (ProductID)
);
go

CREATE TABLE dbo.Employee
(
	EmployeeID int NOT NULL IDENTITY,
	FirstName varchar(50) NOT NULL,
	LastName varchar(50) NOT NULL,
	Department varchar(50) NULL,
	DateHired date Null,
	CONSTRAINT PK_Employee Primary Key (EmployeeID)
);
go
```

```
CREATE TABLE dbo.CustomerOrder
(
	OrderID int Not Null IDENTITY,
	CustomerID int Not Null,
	PaymentID int Not Null,
	OrderDate date Null,
	ProductID int Not Null,
	EmployeeID int Not Null,
	CONSTRAINT PK_CustomerOrder Primary Key (OrderID),
	CONSTRAINT FK1_CustomerOrder Foreign Key (CustomerID) references
Customer(CustomerID),
	CONSTRAINT FK2_CustomerOrder Foreign Key (PaymentID) references
Payment(PaymentID),
	CONSTRAINT FK3_CustomerOrder Foreign Key (EmployeeID) references
Employee(EmployeeID),
	CONSTRAINT FK4_CustomerOrder Foreign Key (ProductID) references Product(ProductID)
);
go

CREATE TABLE dbo.OrderDetail
(
	OrderDetailID int Not Null IDENTITY,
	ProductID int not Null,
	Quantity int Null,
	Discount int Null,
	Total money Null,
	Size varchar(20) Null,
	Color varchar(20) Null,
	Available int Null,
	OrderID int Not Null,
	ShipDate date Null,
	CONSTRAINT PK_OrderDetail Primary Key (OrderDetailID),
	CONSTRAINT FK_OrderDetailProduct Foreign Key (ProductID) references
Product(ProductID),
	CONSTRAINT FK_OrderDetailCustomer Foreign Key (OrderID) references
CustomerOrder(OrderID)
);
go


CREATE PROCEDURE dbo.InsertCustomer
	@FirstName varchar(50),
	@LastName varchar(50)
AS
```

```
	INSERT dbo.Customer (FirstName, LastName) VALUES (@FirstName, @LastName);
go

CREATE PROCEDURE dbo.InsertPayment
	@PaymentType varchar(50)
AS
	INSERT dbo.Payment(PaymentType) VALUES (@PaymentType);
go


CREATE PROCEDURE dbo.InsertProduct
	@ProductName varchar(50),
	@ProductDescription varchar(255)
AS
	INSERT dbo.Product(ProductName, ProductDescription) VALUES (@ProductName,
@ProductDescription);
go



-- insert records into tables
INSERT dbo.Customer(FirstName, LastName, ShipAddress, ShipCity, ShipState, ShipPostal,
DogSize, Email)
VALUES ('Sheev', 'Palpatine', '2324 Empire State', 'Los Angeles', 'CA', '90001', 'Large',
'sheeve.pal@gmail.com'),
		('Qui','Jinn', '1371 Darth State', 'Los Angeles', 'CA', '90010','XLarge',
'jinn09@gmail.com'),
		('Obi', 'Kenobi', '917 Ocean Jedi', 'Los Alamitos', 'CA', '90721',
'Medium', 'obik23@gmail.com'),
		('Mace', 'Windu', '3101 Blue Springs', 'Beverly Hills', 'CA',
'90209','Large', 'windumace@gmail.com'),
		('Maul', 'Darthomir', '8787 Sith Empire', 'Carlsbad', 'CA',
'92008','XLarge', 'darthstar@gmail.com'),
		('Chewy', 'Becca', '3012 Space Drive', 'Malibu', 'CA', '90263','Large',
'becca909@gmail.com');
go

INSERT dbo.Payment(PaymentType,PaymentDate, CustomerID)
VALUES ('VISA', '03/16/2021', (SELECT CUSTOMERID FROM dbo.Customer where LastName=
'Palpatine')),
	  ('MasterCard', '03/17/2021', (SELECT CustomerID from dbo.Customer where
LastName= 'Jinn')),
	  ('VISA', '3/18/2021', (SELECT CustomerID from dbo.Customer where LastName=
'Kenobi')),
```

```
          ('Discover', '3/19/2021', (SELECT CustomerID from dbo.Customer where
LastName= 'Windu')),
          ('MasterCard', '3/24/2021', (SELECT CustomerID from dbo.Customer where
LastName= 'Darthomir')),
          ('American Express', '3/25/2021', (SELECT CustomerID from dbo.Customer where
LastName= 'Becca'));
go

INSERT dbo.Product(ProductName, ProductDescription)
VALUES ('Blue Mid Rim', 'Blue Microfiber robe with galaxy trim'),
              ('Green Maze', 'Green All Cotton robe with abstract trim'),
              ('Orange Wild Space', 'Orange Microsuede robe with tie dye trim');
go

INSERT dbo.Supplier(CompanyName, ProductID)
VALUES ('Microfiber LLC',(SELECT ProductID from dbo.Product where ProductName= 'Blue Mid
Rim')),
          ('AllCotton',(SELECT ProductID from dbo.Product where ProductName= 'Green
Maze')),
          ('FabricsRUs', (SELECT ProductID from dbo.Product where ProductName= 'Orange
Wild Space'))
go

INSERT dbo.Employee(FirstName, LastName)
VALUES ('Boba','Fett'),
          ('Cara', 'Dune'),
          ('Fennec', 'Shand');
go

INSERT dbo.CustomerOrder(CustomerID, PaymentID, OrderDate, ProductID, EmployeeID)
VALUES
  (1, 1, '03/16/2021',1,1),
  (2, 2, '03/17/2021',2,1),
  (3, 3, '03/18/2021',2,2),
  (4, 4, '03/19/2021',3,2),
  (5, 5, '03/24/2021',1,3),
  (6, 6, '03/25/2021',1,1);
go


INSERT dbo.OrderDetail(ShipDate, Total, Quantity, ProductID, OrderID)
VALUES ('03/17/2021', '$80', 2, 1, (SELECT OrderID from dbo.CustomerOrder where OrderID=1
)),
          ('03/18/2021', '$56',1, 2, (SELECT OrderID from dbo.CustomerOrder where
OrderID=2)),
```

```
          ('03/19/2021', '$35', 1, 3, (SELECT OrderID from dbo.CustomerOrder where
OrderID=3)),
          ('03/20/2021', '$120', 3, 1, (SELECT OrderID from dbo.CustomerOrder where
OrderID=4)),
          ('03/25/2021', '$112', 2, 2, (SELECT OrderID from dbo.CustomerOrder where
OrderID=5)),
          ('03/26/2021', '$40', 1, 1, (SELECT OrderID from dbo.CustomerOrder where
OrderID=6));
go




-- Pull results from the tables
SELECT *
FROM   dbo.Customer;
go

SELECT *
FROM   dbo.Payment;
go

SELECT *
FROM   dbo.Product;
go

SELECT *
FROM   dbo.CustomerOrder
go

--Create a view with Customer and Payment

Create VIEW dbo.CustomerPayment
AS
SELECT PT.PaymentType, CUST.Email, CUST.LastName, PRD.ProductName, PRD.ProductDescription
          FROM dbo.Customer as CUST
          INNER JOIN dbo.Payment AS PT ON PT.CustomerID = CUST.CustomerID
          INNER JOIN dbo.CustomerOrder AS CO ON CO.PaymentID= PT.PaymentID
          INNER JOIN dbo.Product AS PRD ON PRD.ProductID = CO.ProductID
          ;
go

--Check the CustomerPayment View
SELECT * from dbo.CustomerPayment
```

```sql
--Data Question 1: How many orders each week so far?
Select DATEADD(Day, -DateDiff(Day, 0, [OrderDate])%7,[OrderDate]) AS [Weekof],
COUNT (*) AS Count
From dbo.CustomerOrder
Group By DATEADD(Day, -DateDiff(Day,0,[OrderDate]) %7, [OrderDate])

--Data Question 2: Which product is ordered the most?
Select ProductID, COUNT (ProductID) AS value_occurrence
From OrderDetail
Group By ProductID
Order By value_occurrence DESC

Select * from Product where ProductID= 1

--Data Question 3: What size are most commonly ordered?
Select DogSize, Count (*) from Customer
Group By DogSize
Order By DogSize asc

--Date Question 4: What is the monthly total? And weekly total?
SELECT
DATEPART (m, ShipDate) AS Month,
Sum(Total) AS Total
FROM OrderDetail
Group By DATEPART (m, ShipDate);

SELECT
DATEPART (wk, ShipDate) AS Week,
Sum(Total) AS Total
FROM OrderDetail
Group By DATEPART (wk, ShipDate);

--Data Question 5: What are the most ordered areas (zip code)?
Select ShipPostal from Customer
```

Appendix I-- Raw Data

Appendix II -- Maintenance Form

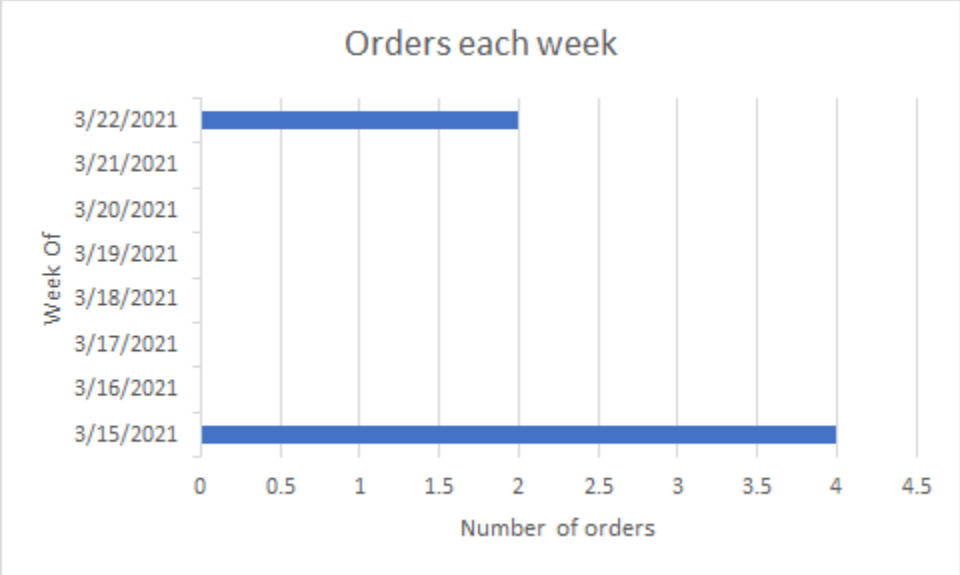
ANSWERING DATA QUESTIONS:

1. How many orders are made each week so far?

```sql
--Data Question 1: How many orders each week so far?
Select DATEADD(Day, -DateDiff(Day, 0, [OrderDate])%7,[OrderDate]) AS [Weekof],
COUNT (*) AS Count
From dbo.CustomerOrder
Group By DATEADD(Day, -DateDiff(Day,0,[OrderDate]) %7, [OrderDate])
```

Results | Messages

	Weekof	Count
1	2021-03-15	4
2	2021-03-22	2



2. Which product is ordered the most?

```sql
--Data Question 2: Which product is ordered the most?
Select ProductID, COUNT (ProductID) AS value_occurrence
From OrderDetail
Group By ProductID
Order By value_occurrence DESC

Select * from Product where ProductID= 1
```

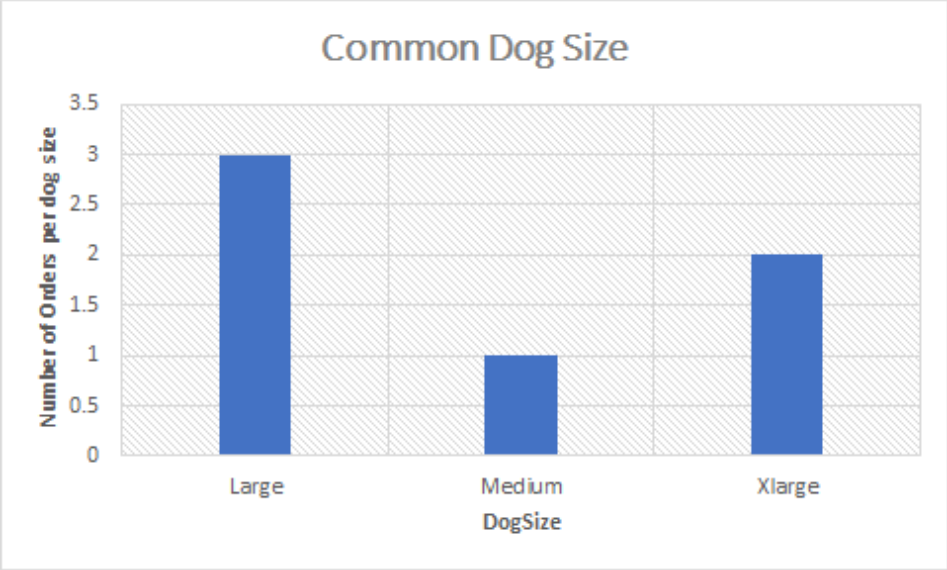
	ProductID	value_occurrence
1	1	3
2	2	2
3	3	1

	ProductID	ProductName	ProductDescription	Fabric	Color	Size
1	1	Blue Mid Rim	Blue Microfiber robe with galaxy trim	NULL	NULL	NULL

3. What size dogs are most ordered for?

```
--Data Question 3: What size are most commonly ordered?
Select DogSize, Count (*) from Customer
Group By DogSize
Order By DogSize asc
```

	DogSize	(No column name)
1	Large	3
2	Medium	1
3	XLarge	2



4. What is the monthly total? What is the weekly total?

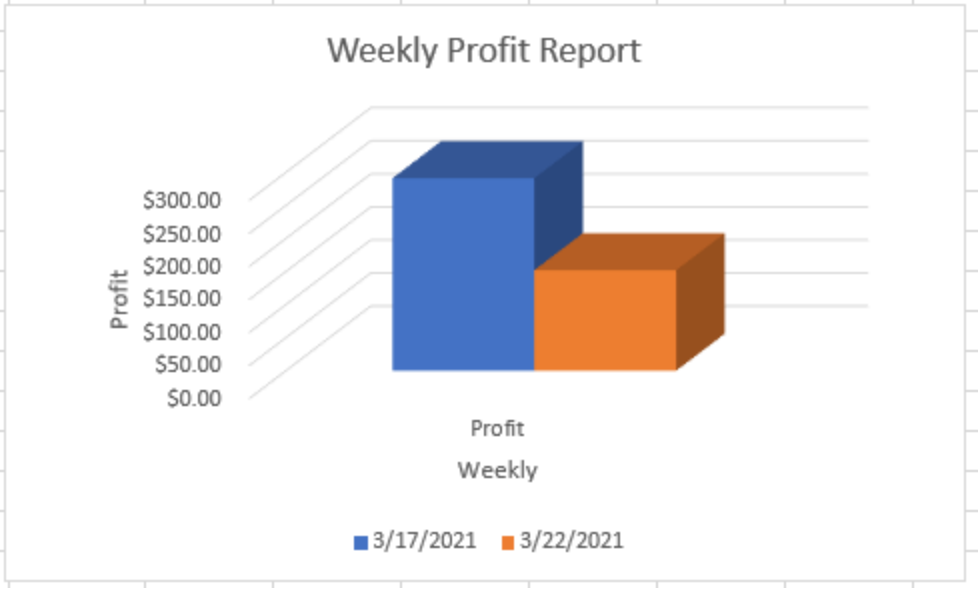
```
--Data Question 4: What is the monthly total? And weekly total?
SELECT
DATEPART (m, ShipDate) AS Month,
Sum(Total) AS Total
FROM OrderDetail
Group By DATEPART (m, ShipDate);

SELECT
DATEPART (wk, ShipDate) AS Week,
Sum(Total) AS Total
FROM OrderDetail
Group By DATEPART (wk, ShipDate);
```

	Month	Total
1	3	443.00

	Week	Total
1	12	291.00
2	13	152.00

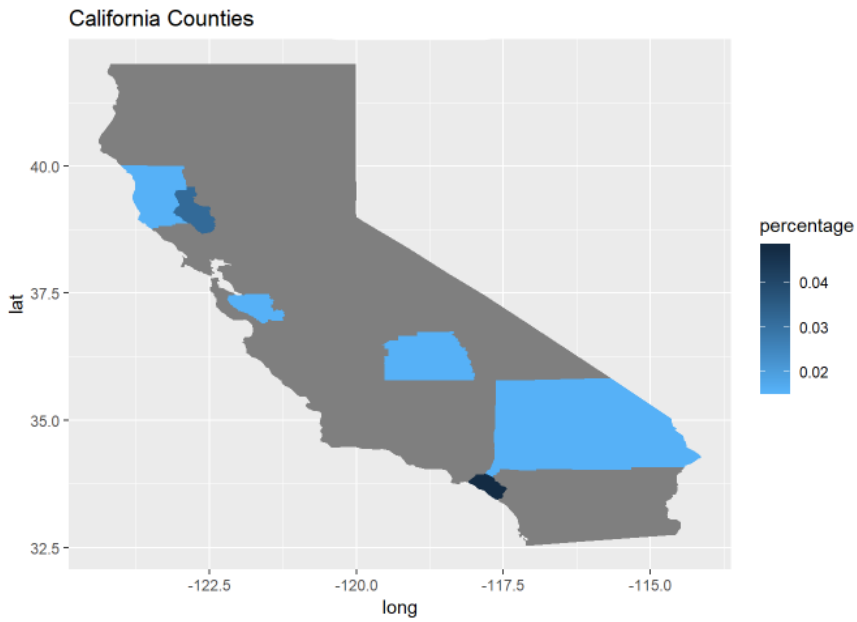
Week	Profit
3/17/2021	$291.00
3/22/2021	$152.00
Total	$443.00



5. What are the most ordered areas (zip code)?

```
--Data Question 5: What are the most ordered areas (zip code)?
Select ShipPostal from Customer
```

	ShipPostal
1	90001
2	90010
3	90721
4	90209
5	92008
6	90263



Appendix I - Raw Data:

3 Review items and shipping

Select FREE Amazon Day Delivery below to have orders delivered together in fewer boxes on a single day.

Delivery date: Mar. 31, 2021 If you order in the next 7 hours and 27 minutes (Details)
Items shipped from Amazon.com

THE SNUGGLY DOG Easy Wear Dog Towel. Luxuriously Soft, Fast Drying 400gsm Microfiber. Soft Belt Included for a Warm Plush Dog Robe Large Blue
$40.47 prime & FREE Returns
Qty: 1
Sold by: The Snuggly Dog
Add gift options

Choose your Prime delivery option:
- **Wednesday, Mar. 31**
  FREE Prime Delivery
- **Tuesday, April 6**
  FREE Amazon Day Delivery
  Get your orders together in fewer boxes and deliveries each Tuesday. Change delivery day
- **Sunday, April 4 - Monday, April 5**
  FREE No-Rush Shipping
  Get a $1 reward for select digital purchases. One reward per purchase. Details

Place your order
By placing your order, you agree to Amazon's privacy notice and conditions of use.

Order Summary

Items:	$40.47
Shipping & handling:	$0.00
Total before tax:	$40.47
Estimated tax to be collected:	$2.43
Order total:	**$42.90**

How are shipping costs calculated?
Prime shipping benefits have been applied to your order.

Place your order **Order total: $42.90**
By placing your order, you agree to Amazon's privacy notice and conditions of use.

Appendix II - Maintenance Form

Reflection

When I first started working on the project I did not understand the importance of the conceptual and logical model. After full completion of the project, I understand how helpful conceptual/logical models are. Once the models are completed the design of the database is a walk in the park. The only thing I would do differently is create a web interface using R to collect more data and show another visualization. Also, learning how to connect Access to SQL was very important because I can now apply what I have learned to my job.

Summary

This class was extremely helpful. I learned how to create my own database, connect sql to R and Access. Most of my data questions were answered by a combination of SQL SELECT

statements and creating a chart in either R and/or Excel. I now feel comfortable creating my own database and creating primary, foreign, and unique keys. I really enjoyed this course.