E-Commerce Database Management System

Project Overview:

In this project we are going to design an effective centralized E-commerce database which will maintain customer records, making the process of updating & retrieval of records convenient.

Background:

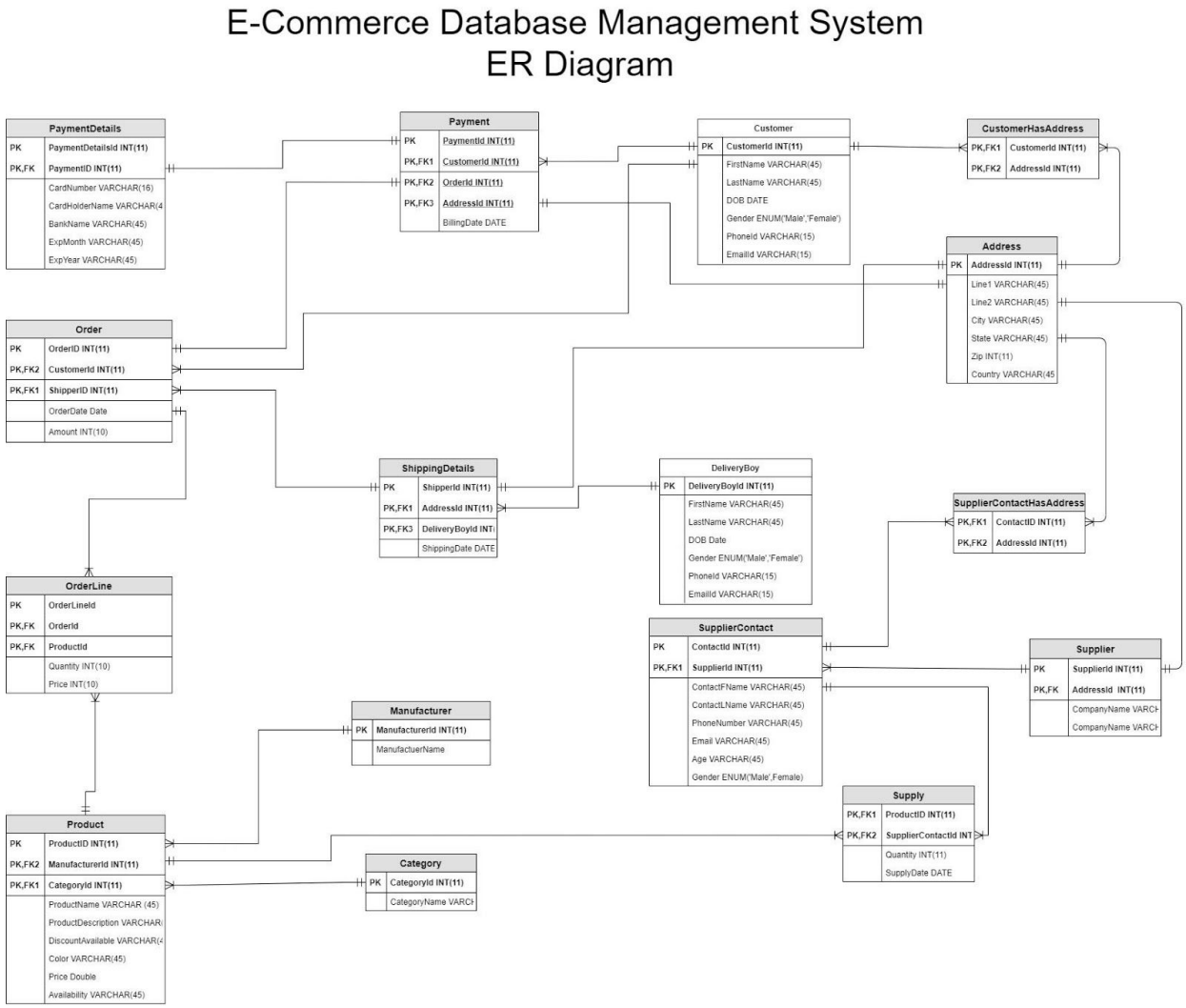
In today's world E-commerce data is fast gaining ground as an accepted and used business model. Which generates humongous data & it is a very tedious job to maintain customer records manually using paper based records. For any shopping store if we need to target a set of customers it is time consuming to go through multiple paper records & target a set of customers. Every year the amount of people on the planet are increasing and so are the basic demands of every human. People keep ordering multiple things from the internet and websites like Amazon are the highest in that regard. There are multiple orders being placed every second and to be able to keep a record is a huge task. The study on this database will help the customers retrieve their orders very fast.

Entities:

The entities that will be present in the database are as follows:

- **Customer:** Provides the basic information of a customer such as Name, Address, Phone Number
- **Payment:** Provides payment details of the Order of Customers and also the Address on which The order has been billed
- **PaymentDetails:** Provides Card details and Bank Details of the transaction
- **Orders:** Provides details about the Order that the customer has placed
- **OrderLine:** Provides the information of the items purchased by the customer
- **Product:** Provide product details like price, product description etc
- **Category:** Generalized category of products that belong to them
- **Manufacturer:** Provide the details of the manufacturer of a product.
- **Supply:** Provide details about the product that has been supplied by a supplier
- **Supplier:** Provide details about the supplier
- **SupplierContact:** Provides detail of a person who is associated with the supplier
- **DeliveryBoy:** Provide basic information of a person who is delivering the product.
- **ShippingDetails:** Provide basic information of the shipping details of an order. Sometimes the Shipping address and the billing address are different so this entity can prove to be beneficial for such cases.
- **Address:** This Entity contains all the addresses of the Customer, Supplier, Supplier Contact and Delivery Boy. Each Address has a unique id which refers to a particular entity so that there won't be any problem while specifying the address of a particular entity.

ER DIAGRAM:



Entity Relationship:

One to One Relationship: In one to one relationship, a row in either of the entities can be related to only one row of the other entity relation.

This kind of relation can be found between following entities:

- Address and Supplier
- Address and ShippingDetails
- Payment and PaymentDetails
- Payment and Orders
- Payment and Address

One to Many Relationship: In one to many relationships, a row in the first relation can be related to one or more rows in the second relation. But, the row in the second relation will have only a single relation with the row of the first relation.

This kind of relationship can be found between following entities:

- Orders and ShippingDetails
- Supplier and SupplierContact
- DeliveryBoy and ShippingDetails
- Product and Manufacturer
- Product and Category
- Customer and Payment
- Customer and Orders

Many to Many Relationship: In many to many relationships, one or more than one row in the first relation can be related to one or more rows in the second relation. Similarly, the row in the second relation will be related to one or more rows of the first relation.

This kind of relationship can be found between following entities:

- Customer and Address
- SupplierContact and Address
- SupplierContact and Product
- Product and Orders

VIEWS:

A view is a virtual table based that can be made by the needs of the user. We can join various tables and combine them as a view.

We have created 2 views which involves joining tables to view data on a single table

• **CustomerDetails:** In this view we are displaying all the details related to the Customer, we will be able to view the region from where the customer has ordered the product, we can also view the Payment Details Card Details, Products Ordered by the Customer.

```
CREATE VIEW CustomerDetails AS
SELECT (c.FirstName+ ' '+ c.LastName) as Fullname, a.State,a.City,a.Zip, p.PaymentId, pd.BankName,
        pd.Credit_card_number_encrypt, o.OrderId, ol.OrderLineId,o.OrderDate,prd.ProductName,
            prd.ProductDescription, ctg.CategoryName,prd.Price ,ol.Quantity,ol.TotalPrice, o.Amount
FROM Customer c
INNER JOIN Payment p
ON p.CustomerId = c.CustomerId
INNER JOIN PaymentDetails pd
ON pd.PaymentId = p.PaymentId
INNER JOIN Orders o
ON o.OrderId = p.OrderId
INNER JOIN OrderLine ol
ON ol.OrderId = o.OrderId
INNER JOIN Product prd
ON prd.ProductId = ol.ProductId
INNER JOIN Category ctg
ON ctg.CategoryId = prd.CategoryId
INNER JOIN CustomerHasAddress cha
ON c.CustomerId = cha.CustomerId
INNER JOIN Address a
ON a.AddressID = cha.AddressId;
```

Results | Messages

	Fullname	State	City	Zip	PaymentId	BankName	Credit_card_number_encrypt	OrderId	OrderLineId	OrderDate	ProductName	ProductDescription	CategoryName	Price	Quantity	TotalPrice	Amount
1	Lucius McKinty	Wisconsin	Madison	53705	100000	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000CA0530...	5000	1	2020-02-12	Headphones	Electronics	Technology	90.00	2	180.00	548.00
2	Lucius McKinty	Wisconsin	Madison	53705	100000	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000CA0530...	5000	2	2020-02-12	Mobile Phones	Electronics	Technology	52.00	1	52.00	548.00
3	Lucius McKinty	Wisconsin	Madison	53705	100000	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000CA0530...	5000	3	2020-02-12	Baby soap	Baby	Infants	79.00	4	316.00	548.00
4	Lucius McKinty	Wisconsin	Madison	53705	100011	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000611864D...	5002	4	2019-09-26	Grass	Garden	HomeDecor	59.00	1	59.00	195.00
5	Lucius McKinty	Wisconsin	Madison	53705	100011	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000611864D...	5002	5	2019-09-26	80 classic CDs	Music	Entertainment	68.00	2	136.00	195.00
6	Selie Rabidge	District of Columbia	Washington	20036	100022	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000B954926...	5004	6	2019-12-11	Lamp	Home	HomeDecor	64.00	3	192.00	592.00
7	Selie Rabidge	District of Columbia	Washington	20036	100022	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000B954926...	5004	7	2019-12-11	Bolt	Industrial	Hardware and Tools	100.00	4	400.00	592.00
8	Alis Wash	New York	Rochester	14683	100033	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F020000000BE2D3...	5006	8	2019-10-21	Fan	Home	HomeDecor	54.00	2	108.00	273.00
9	Alis Wash	New York	Rochester	14683	100033	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F020000000BE2D3...	5006	9	2019-10-21	Kids water pool	Kids	Kids	55.00	3	165.00	273.00
10	Carline Merrick	California	Fresno	93715	100044	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F020000001B95EC...	5008	10	2019-06-07	Rings	Jewelery	Diamonds & Jewel...	54.00	4	216.00	508.00
11	Carline Merrick	California	Fresno	93715	100044	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F020000001B95EC...	5008	11	2019-06-07	Avatar	Movies	Entertainment	73.00	4	292.00	508.00
12	Francisca Gio...	Missouri	Saint Louis	63167	100055	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000A57F143...	5010	12	2019-06-30	Keyboard	Computers	Technology	65.00	2	130.00	653.00
13	Francisca Gio...	Missouri	Saint Louis	63167	100055	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000A57F143...	5010	13	2019-06-30	ZARA T-shirt	Clothing	Clothing & Apparel	82.00	4	328.00	653.00
14	Francisca Gio...	Missouri	Saint Louis	63167	100055	Bank of America	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000A57F143...	5010	14	2019-06-30	Wireless Mou...	Computers	Technology	65.00	3	195.00	653.00
15	Timofei Menaul	District of Columbia	Washington	20370	100066	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000006BB7F87...	5012	15	2019-09-26	Badminton R...	Sports	Fitness	76.00	5	380.00	718.00
16	Timofei Menaul	District of Columbia	Washington	20370	100066	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000006BB7F87...	5012	16	2019-09-26	Perfume	Beauty	Personal Care	64.00	3	192.00	718.00
17	Timofei Menaul	District of Columbia	Washington	20370	100066	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000006BB7F87...	5012	17	2019-09-26	RC plane	Toys	Kids	73.00	2	146.00	718.00
18	Aurlia Willowby	Missouri	Columbia	65218	100077	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000045CC9E...	5014	18	2019-09-11	Nike runner 3.0	Shoes	Fitness	87.00	3	261.00	393.00
19	Aurlia Willowby	Missouri	Columbia	65218	100077	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000045CC9E...	5014	19	2019-09-11	Lee cooper f...	Shoes	Clothing & Apparel	53.00	1	53.00	393.00
20	Aurlia Willowby	Missouri	Columbia	65218	100077	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000045CC9E...	5014	20	2019-09-11	Plants	Garden	HomeDecor	79.00	1	79.00	393.00
21	Earle Folca	Michigan	Detroit	48242	100088	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F020000003AA5159...	5016	21	2019-04-12	Sanitizer	Natural Made	Health Care	25.00	2	50.00	215.00
22	Earle Folca	Michigan	Detroit	48242	100088	TD Citizen Bank	0x00F32DF89F6C1542BF7AFF314FF88F6F020000003AA5159...	5016	22	2019-04-12	Disposable G...	100pcs Per Pack	Health Care	15.00	3	45.00	215.00

• **SupplierDetails:**

This View provides the details of Supplier who has supplied the products with the help of their Supplier_contact and also the Quantity of the products supplied.

```
CREATE VIEW SupplierDetails AS

SELECT s.CompanyName,sc.ContactFName,sc.ContactLName,sc.CurrentAge,sc.PhoneNumber,a.State,p.ProductName,
        sp.Quantity,p.ProductDescription,c.CategoryName

FROM Supplier s
INNER JOIN SupplierContact sc
ON sc.SupplierId = s.SupplierId
INNER JOIN SupplierContactHasAddress sca
ON sc.SupplierContactId = sca.SupplierContactId
INNER JOIN Address a
ON a.AddressID = sca.AddressId
INNER JOIN Supply sp
ON sp.SupplierContactId = sc.SupplierContactId
INNER JOIN Product p
ON p.ProductId = sp.ProductId
INNER JOIN Category c
ON c.CategoryId = p.CategoryId;
```

	CompanyName	ContactFName	ContactLName	CurrentAge	PhoneNumber	State	ProductName	Quantity	ProductDescription	CategoryName
1	Klein LLC	Shelley	Lates	43	222-542-4405	Virginia	Headphones	10	Electronics	Technology
2	Hagenes-Grady	Babbette	Lehrmann	37	125-564-5427	California	Headphones	14	Electronics	Technology
3	Legros, Dicki and Homenick	Claiborn	Manuelli	44	650-798-3779	Indiana	Mobile Phones	11	Electronics	Technology
4	Howe-Kertzmann	Lissy	Tippler	34	141-697-4972	Florida	Mobile Phones	10	Electronics	Technology
5	Senger-Hansen	Laney	Cordsen	41	984-440-0225	Illinois	Baby soap	14	Baby	Infants
6	Johns LLC	Beck	Crimpe	40	256-986-8473	District of Columbia	Baby soap	10	Baby	Infants
7	Towne-Boehm	Shannon	Hampshire	35	580-372-6843	Florida	Grass	10	Garden	HomeDecor
8	Kreiger, Grady and Crist	Jerri	Gazey	34	421-210-1831	Pennsylvania	Grass	11	Garden	HomeDecor
9	Wolf, Dach and Becker	Raye	Corrigan	47	813-442-1460	North Carolina	80 classic CDs	10	Music	Entertainment
10	Lueilwitz-Jast	Terrill	Lancley	46	631-991-4316	Virginia	80 classic CDs	14	Music	Entertainment
11	Bailey Inc	Deck	Verheyden	46	454-353-3725	California	Lamp	11	Home	HomeDecor
12	Ritchie, Johnston and Rog...	Pru	Van Arsdall	43	204-758-2040	Georgia	Lamp	10	Home	HomeDecor
13	Hagenes-Grady	Babbette	Lehrmann	37	125-564-5427	California	Bolt	14	Industrial	Hardware an...
14	Howe-Kertzmann	Lissy	Tippler	34	141-697-4972	Florida	Fan	10	Home	HomeDecor

TRIGGERS:

Triggers are stored programs, which are automatically executed or fired when some events occur.

We have created 2 triggers in which we will be able to take a backup of the table product on update. The other trigger is created for the backup of the customer data on UPDATE, INSERT and DELETE.

TRIGGER1-ProductAuditTable:

This trigger is triggered on the update of the product table.

```
CREATE TRIGGER ProductAuditTable ON Product
AFTER UPDATE
AS BEGIN
        Declare @action VARCHAR(3)
        SET @action = 'U'

        INSERT INTO ProductAudit(
                ProductId, ManufactureId, CategoryId, ProductName, ProductDescription,
                DiscountAvailable, Color, Price, Availability, Action, ActionDate
        )
        SELECT
                ProductId, ManufactureId, CategoryId, ProductName, ProductDescription,
                DiscountAvailable, Color, Price, Availability, @action, GETDATE()
        FROM deleted

END
```

110 %

Results | Messages

	ProductAuditId	ProductId	ManufactureId	CategoryId	ProductName	ProductDescription	DiscountAvailable	Color	Price	Availability	Action	ActionDate
1	1	133	22	502	Face Masks	N95-3pc Per Pack	0	White	30.00	Yes	U	2020-04-07 23:53:17.933

TRIGGER2-CustomerBackup:

This trigger is triggered on INSERT, UPDATE and DELETE of the Customer Table.

```
CREATE TRIGGER CustomerBackup
  ON  Customer
  AFTER INSERT,DELETE,UPDATE
AS
BEGIN
        INSERT INTO CustomerAudit (
        CustomerId, FirstName, LastName, DOB, Gender, PhoneId, EmailId, updated_at,
        operation
         )
         SELECT
         i.CustomerId, FirstName, LastName, DOB, Gender, PhoneId, EmailId,
         GETDATE(), 'INSERT'
         from inserted as i
         UNION ALL
SELECT
         d.CustomerId, FirstName, LastName, DOB, Gender, PhoneId, EmailId, getdate(),
         'DELETE'
         FROM deleted as d;
```

```
END
```

110 %

Results | Messages

	CustomerAuditId	CustomerId	FirstName	LastName	DOB	Gender	PhoneId	EmailId	updated_at	operation
1	1	9999	ADISH	JOSHI	1997-11-26	Male	6179562274	adish@instagram.com	2020-04-08 12:48:52.530	INSERT
2	2	9999	ADISH	JOSHI	1997-11-26	Male	6179562274	adish@instagram.com	2020-04-08 13:02:46.510	DELETE

STORED PROCEDURES:

A stored procedure in SQL is a type of code in SQL that can be stored for later use and can be used many times. So, whenever you need to execute the query, instead of calling it you can just call the stored procedure. Values can be passed through stored procedures.

We have created 4 Stored Procedures.

Stored Procedure1- GetHighestOrderByGender

In this Stored Procedure we have 1 INPUT Parameter and 2 OUTPUT Parameters, we will pass the Gender of the Customer and the Procedure will return the Highest Order placed by the person according to the Gender.

```
ALTER PROCEDURE GetHighestOrderByGender
@Gender varchar(6),
@HighestOrder INT OUTPUT,
@CustomerName VARCHAR(45) OUTPUT
AS BEGIN
        SELECT top 5 sum(Amount), c.FirstName, c.LastName
        FROM Orders o
        INNER JOIN Customer c
        ON c.CustomerId = o.CustomerId
        GROUP BY c.FirstName, c.LastName
        ORDER BY sum(o.Amount) DESC
        HAVING c.Gender =
END
DECLARE @HighestOrderByGender INT, @FullName VARCHAR(45)
EXEC GetHighestOrderByGender 'Male', @HighestOrderByGender OUTPUT, @FullName OUTPUT
SELECT @HighestOrderByGender, @FullName
```

Results | Messages

	Amount$	FullName
1	718	Timofei Menaul

Stored Procedure 2- CustomerOrderHistory

The below stored procedure will give the Entire Order History of the customer when Customer Id is Passed as an INPUT

```
CREATE PROCEDURE CustomerOrderHistory
@CustomerId INT
AS BEGIN

        SELECT c.CustomerId, c.FirstName, c.LastName, o.OrderId,o.Amount,
                o.OrderDate, p.ProductId,p.ProductName
        FROM Customer c
        INNER JOIN Orders o
        ON c.CustomerId = o.CustomerId
        INNER JOIN OrderLine ol
        ON ol.OrderId = o.OrderId
        INNER  JOIN Product p
        ON p.ProductId = ol.ProductId
        WHERE c.CustomerId = @CustomerId
        ORDER BY o.OrderDate

END

EXEC CustomerOrderHistory @CustomerId = 3000
```

Results | Messages

	CustomerId	FirstName	LastName	OrderId	OrderDate	Amount	ProductId	ProductName
1	3000	Lucius	McKinty	5000	2020-02-12	548.00	111	Headphones
2	3000	Lucius	McKinty	5000	2020-02-12	548.00	112	Mobile Phones
3	3000	Lucius	McKinty	5000	2020-02-12	548.00	113	Baby soap

STORED PROCEDURE 3–TotalSalesPerDay

The below stored procedure can be used to find the total sales that are generated on a particular date. When we put the date into the input of the stored procedure, the total sales we want to find gets displayed .

```
CREATE PROCEDURE TotalSalesPerDay
@OrderDate DATE
AS BEGIN
        SELECT o.OrderDate, sum((p.Price-(p.Price*p.DiscountAvailable ))*ol.Quantity ) as TotalSale$
        FROM Orders o
        INNER JOIN OrderLine ol
        ON ol.OrderId = o.OrderId
        INNER JOIN Product p
        ON p.ProductId = ol.ProductId
        WHERE o.OrderDate = @OrderDate
        GROUP BY o.OrderDate
END

EXEC TotalSalesPerDay @OrderDate = '2019-04-12'
```

Results | Messages

	OrderDate	TotalSale$
1	2019-04-12	212.5000

STORED PROCEDURE 4-UpdateProductAvailibilty

I have created a stored procedure in which if we enter an input which is the id of the table 'Product', the attribute named 'Availablity' will be set to SOLD and the old details will be reflected on the back up table since the stored procedure will also fire the trigger

```
CREATE PROCEDURE UpdateProductAvailibilty
@ProductId INT
AS BEGIN
        UPDATE Product
        SET Availability = 'Sold'
        WHERE ProductId = @ProductId;
END
EXEC UpdateProductAvailibilty @ProductId = 133
```

Results | Messages

	ProductId	ManufactureId	CategoryId	ProductName	ProductDescription	DiscountAvailable	Color	Price	Availability
1	133	22	502	Face Masks	N95-3pc Per Pack	0	White	30.00	Sold

ProductAudit TABLE

Results | Messages

	ProductAuditId	ProductId	ManufactureId	CategoryId	ProductName	ProductDescription	DiscountAvailable	Color	Price	Availability	Action	ActionDate
1	1	133	22	502	Face Masks	N95-3pc Per Pack	0	White	30.00	Yes	U	2020-04-07 23:53:17.933

Computed Columns Using Functions:

In TABLE Customer, DeliveryBoy, SupplierContact we've Calculated the Current Age of the person by using their Date of Birth(DOB Column)

```
ALTER TABLE Customer ADD CurrentAge AS DATEDIFF(YEAR, DOB, GETDATE())

ALTER TABLE DeliveryBoy ADD CurrentAge AS DATEDIFF(YEAR, DOB, GETDATE())

ALTER TABLE SupplierContact ADD CurrentAge AS DATEDIFF(YEAR, DOB, GETDATE())
```

	CustomerId	FirstName	LastName	DOB	Gender	PhoneId	EmailId	CurrentAge
1	3000	Lucius	McKinty	1995-11-29	Male	948-942-9785	lmckinty0@instagram.com	25
2	3003	Selie	Rabidge	2002-10-17	Female	912-243-3632	srabidge1@etsy.com	18
3	3006	Nelly	Brandsma	1986-12-05	Female	284-583-4302	nbrandsma2@histats.com	34
4	3009	Alis	Wash	1971-05-31	Female	309-486-3045	awash3@cornell.edu	49
5	3012	Carline	Merrick	1993-01-15	Female	636-248-9056	cmerrick4@photobucket.com	27
6	3015	Francisca	Giovannoni	1998-01-31	Female	193-837-2569	fgiovannoni5@bloomberg.com	22
7	3018	Timofei	Menaul	1983-08-08	Male	413-995-7323	tmenaul6@symantec.com	37

ENCRYPTION COLUMN DATA:

We have Encrypted the Credit Card Number of the Customers

```
UPDATE PaymentDetails
SET Credit_card_number_encrypt = EncryptByKey (Key_GUID('SymmetricKey1'),CardNumber)
FROM Payment;
```

Results | Messages

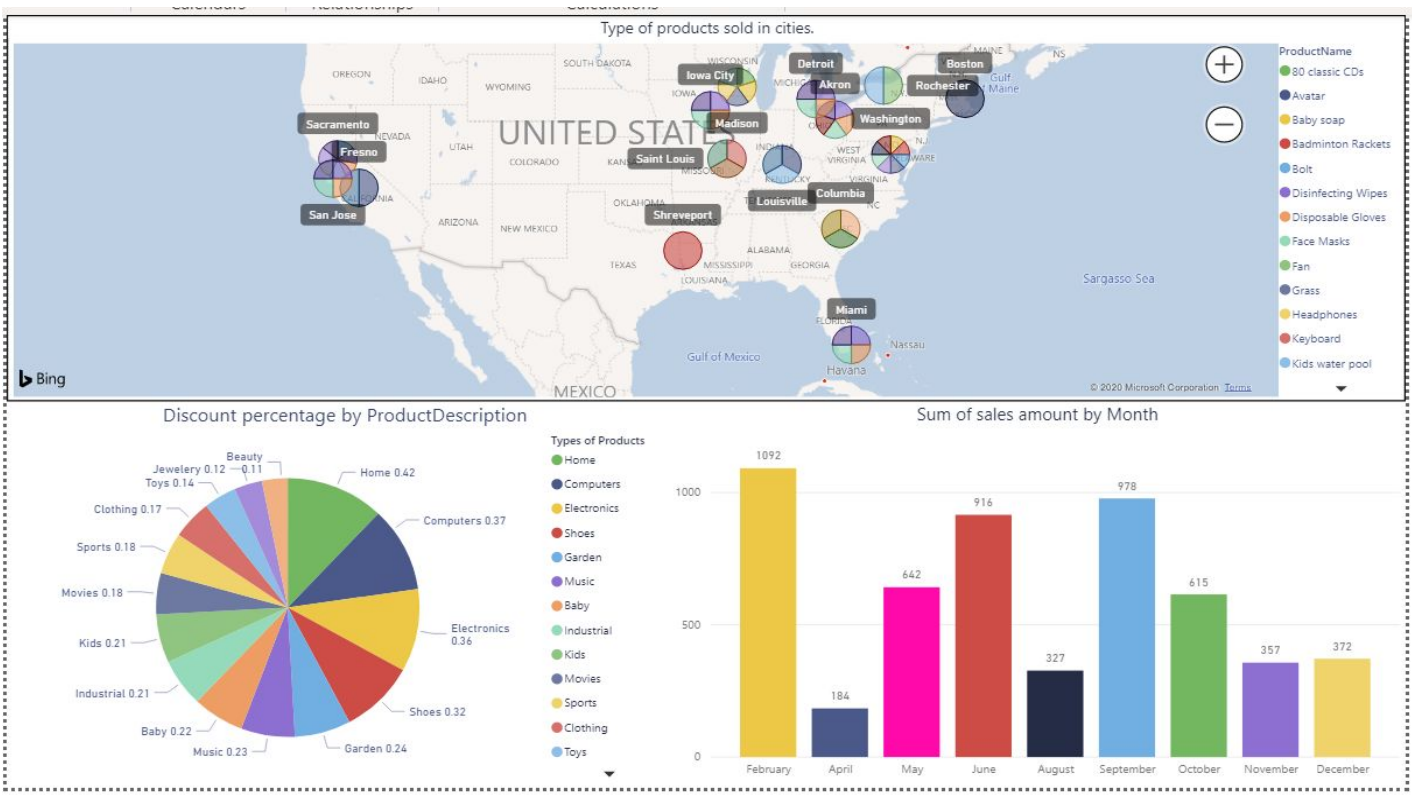
	PaymentDetailsId	PaymentId	CardHolderName	BankName	ExpMonth	ExpYear	Credit_card_number_encrypt
1	100	100000	Gage	Bank of America	January	2023	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000CA0530E...
2	101	100011	Sean	Bank of America	February	2023	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000611864D...
3	102	100022	Noble	Bank of America	March	2023	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000B954926...
4	103	100033	Amena	Bank of America	April	2019	0x00F32DF89F6C1542BF7AFF314FF88F6F020000000BE2D3...
5	104	100044	Ian	Bank of America	May	2024	0x00F32DF89F6C1542BF7AFF314FF88F6F020000001B95EC9...
6	105	100055	Tarik	Bank of America	June	2024	0x00F32DF89F6C1542BF7AFF314FF88F6F02000000A57F143...
7	106	100066	Cain	TD Citizen Bank	July	2021	0x00F32DF89F6C1542BF7AFF314FF88F6F020000006BB7F87...
8	107	100077	Lydia	TD Citizen Bank	August	2019	0x00F32DF89F6C1542BF7AFF314FF88F6F0200000045CC9E9...

Visualization of the Data and Creating a dynamic Dashboard using Microsoft Power BI:

Dashboards are interactive consoles which are used to represent the data. The dashboard is a collection of various interactive visual representations like bar-charts, pie diagram, Map visuals, etc. Here I have created 3 visuals for data representation where,

- The first is geographical representation in which we can very easily see the type of products which are sold in various cities in the states.
- The second is the pie chart of the discount on products which explains the discount in percentage for each type of product.
- The last one is a total amount of sales amount/ revenue generated by months which displays the sum of amount generated each month.

The screen shot of the visuals is attached below,



INFO 6210 Database Management and Database Design
Newton's Apple