



Amazon Competitor Database

Design and Implementation and Testing

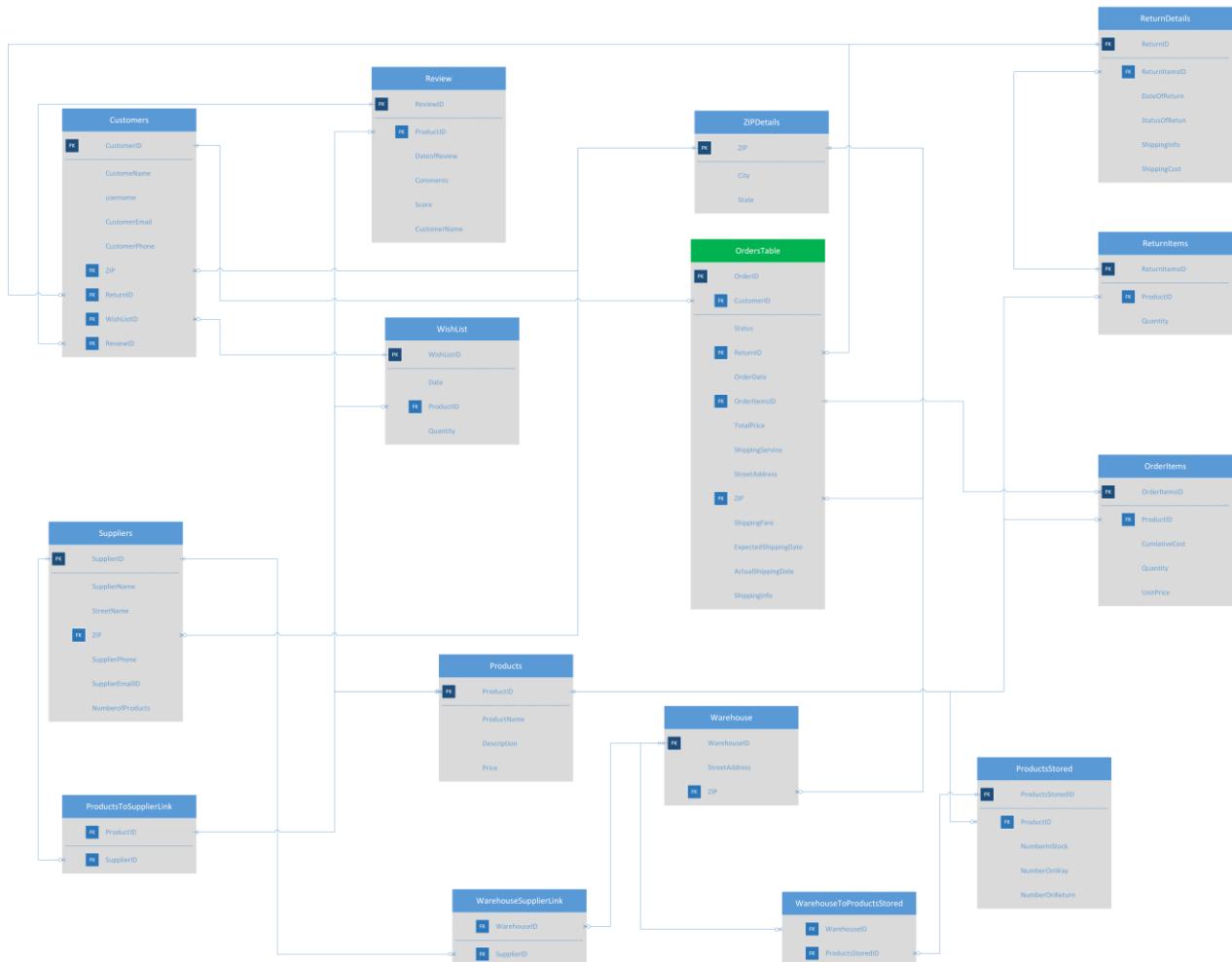
Project2

Parikshit Dubey
265002524

ABSTRACT

In this project we have created a database named “MyAmazonCompetitor” which basically has the similar functionality as the Amazon database. The database has been designed and implemented from scratch and all the steps are properly labelled here in this report. The design of database has been done in third normal form. All the tables and columns created are thoroughly discussed and have been explained in this report. Screenshots are provided for each step which gives the step by step procedure of this Project. The database design has been carried out in MS Visio and the necessary codes are written in Microsoft SQL Server Management studio. For complete review on codes used for the implementation please review the Appendix provided at the end of this report.

Database Design



Database Diagram

Project2

Parikshit Dubey
265002524

The above provided diagram is the design of “MyAmazonCompetitor” database. The design is discussed in detail below:

1. OrdersTable: This is the central part of this database design. It has numerous columns namely OrderID which is uniquely assigned to every order which is created on the system by any individual. CustomerID is generated for every order which is created and it is referenced to the Customers table which has all the customer related data. Status column provides the status of the order which can be either Processing, Preparing to dispatch, Dispatched, Out for delivery, Delivered or Return requested. If there is a Return requested on the order, then ReturnID gets a value which is NOT NULL and it is in turn connected to the ReturnDetails table. OrderDetails column stores the date on which order was created. OrderItemsID has the list of items present in the particular order and it is connected to the OrderItems table for further processing. TotalPrice column has the total amount which is payable by the customer against the order been placed. ShippingService keeps the record of shipping company which has been assigned the shipment. Shipping fare is the extra cost the which is incurred on the order shipping. ExpectedShipping date column provides the expected date on which the order will be shipped. This column can't be null as we have to provide an estimated shipping date for the order being placed by the customer. ActualShipping date is the date on which the shipment was actually shipped. This column can be null initially. ShippingInfo Column provides the extra shipment information which is required for the order being shipped.

2. ReturnDetails: The ReturnDetails table has all the necessary of the items in the product being returned by the customer. Since, this table is dependent of the OrdersTable, it has a connected with the foreign key ReturnID on that table. The primary key for ReturnDetails table is ReturnID. This table has a foreign key ReturnItemID which is connected to the ReturnItems table. The column DateOfReturn has the date on which return will be initiated and it will be a default value of null. StatusOfReturn column has the status of the return for the items being returned, it is null by default which can be modified when item to be returned is shipped or any extra comments are added. ShippingInfo column has the information about the shipping of the items on return, it also has a default value of null. ShippingCost column stores the cost of shipment of item being returned.

3.ReturnItems: This table contains information about the products which are requested to be returned. It is connected to the ReturnDetails column which has the necessary information about the return of the items. This table has a column ProductID which is connected to the Products table for the information about products being returned. The Quantity column has the total number of particular items being returned. It is not null by default.

Project2

Parikshit Dubey
265002524

4.OrderItems: This table contains the information about items that have been placed in order. It is connected to the orders table and also the Products table to garner information about the items whose orders are placed. It has a column named CumulativeCost which gives the total cost of all the products in the Order cart. Quantities of products per order is given in the Quantity column. The price for individual items on order is given by the UnitPrice column which is not null by default.

5.ZipDetails: This table has been used to normalize the other tables to third normal form. It has a primary key of ZIP and the other columns in this table are city and state which are not null by default.

6.Products: Products table store the necessary information about the products. It has a primary key named ProductID which is connected to various table to provide the information about product. The column ProductName in this table store the name of the individual products and it is not null. The Description column provides description about the products being ordered. Price column has the price information about the individual products and it is not null.

7.Customers: This table contains all the information about the customers who placed the order on system. The OrderTable has the orders that are being placed and it is in turn connected to the customers table for the information about customers. It has a column CustomerName which has the name of Customer who placed the order. The username column has the unique username assigned to each customer and it is not null. CustomerEmail column has the customer's email information and it is not null. CustomerPhone has the phone number of customer and it is not null. ZIP column stores the ZIP of customer and it is connected to the ZIPDetails column. The ReturnID column is connected to the ReturnDetails table. The WishListID column is connected to the WishList table. The ReviewID column is connected to the ReviewsTable which is used to gather reviews of the particular products placed by the particular customer.

8. Suppliers: The supplier table has all the information about the suppliers who ship the orders to the customer from the warehouses. The primary key for this table is SupplierID. The SupplierName column has the name of the supplier. The StreetName column contains the information about the street name in the suppliers address and it can't be null. The ZIP column of the supplier is connected to the ZIPDetails column which provides the information about the city and state of supplier. SupplierPhone column provides the phone number of the supplier and it can't be null. SupplierEmailID column provies the email id of supplier and it can't be null. NumberOfProducts column provides the information about the total number of products that are present with the supplier and this column can't be null.

9. Review: Review table provides the review details of a particular product. It is connected to the customer and products table with two individual foreign keys for each table which make up the connection between these tables. It has a column DateOfReview which stores the date on which

Project2

Parikshit Dubey
265002524

review was posted and it can't be null. Comments column in this table has the comments posted against the individual reviews provided by the customers. Score column provides the ratings provided by the customers on a scale of 1 to 10, this column can't be null. CustomerName column provides the name of customer who posted the review and it can be null.

10. WishList: This table provides the wishlist details which is created by the customer. This table is connected to the customers table and the products table with two individual foreign keys. The date column provides the detail on which the wishlist was created. The Quantity column provides the information about the total quantity of items in WishList and it can be null.

11. Warehouse: This table provides information about the location of products which are stored in individual warehouses. From this part the products will be shipped to the customers by sellers. The Warehouse table has many-to-many relations with Suppliers table and Products Stored table.

12. ProductsStore: This table has the information about the products stored in the warehouse. It has many-to-many relationship with the warehouse and is connected to it via the linking table named WarehouseToProductsStored. This table is also connected to the products table. The total number of products in stock is stored in the NumberInStock column. The NumberOnWay column provides the information about the total number of products that are on way to customer. The NumberOnReturn column provides the information about the total products which is in its way return from the customer to the warehouse, it can be null.

13. ProductsToSupplierLink: Since Products and Suppliers table have a many-to-many relationship, we create this table to properly link the two tables.

14. WarehouseSupplierLink: Since warehouse to suppliers have a many-to-many relationship, we use this table as a link between these two tables.

15. WarehouseToProductsStored: Since warehouse and Products stored table have a many-to-many relation. We use this table as a linking table.

Project2

Parikshit Dubey
265002524

Table Name	Primary Key	Foreign Key
Customers	CustomerID	ZIP
		ReturnID
		WishListID
		ReviewID
Review	ReviewID	ProductID
ZIPDetails	ZIP	-
ReturnDetails	ReturnID	ReturnItemsID
ReturnItems	ReturnItemsID	ProductID
OrdersTable	OrderID	ReturnID
		OrderItemsID
		CustomerID
		ZIP
WishList	WishListID	ProductID
OrderItems	OrderItemsID	ProductID
Suppliers	SupplierID	ZIP
Products	ProductID	-
Warehouse	Warehouseld	ZIP
ProductsStored	ProductsStoredID	ProductID

Project2

Parikshit Dubey
265002524

Database Implementation

- Creating the database named MyAmazonCompetitor.

```

use master
GO

/**Object: Database MyAmazonCompetitor****/
IF DB_ID('MyAmazonCompetitor') IS NOT NULL
    DROP DATABASE MyAmazonCompetitor
GO

Create DATABASE MyAmazonCompetitor
Go

USE MyAmazonCompetitor
Go

/**Object: Zip Details Table ****/
CREATE TABLE ZIPDetails(
    ZIP INT PRIMARY KEY IDENTITY NOT NULL,
    City VARCHAR(60) NOT NULL,
    State VARCHAR(2) NOT NULL
);

/**Object: Products table******/
CREATE TABLE Products(
    ProductID INT PRIMARY KEY IDENTITY NOT NULL,
    ProductName VARCHAR(255) NOT NULL,
    Price DECIMAL(10, 2)
);

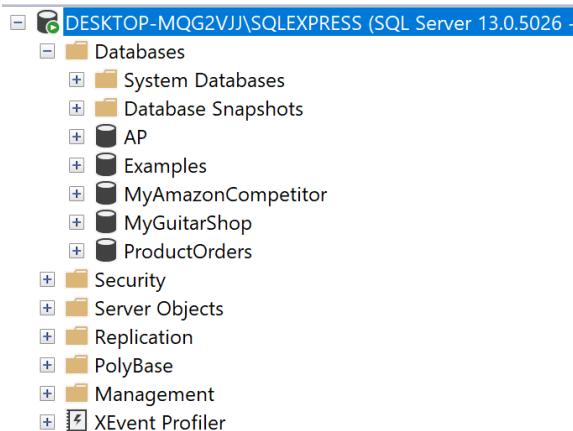
```

80 %

Messages

Commands completed successfully.

Completion time: 2019-12-07T16:38:05.7131089-05:00

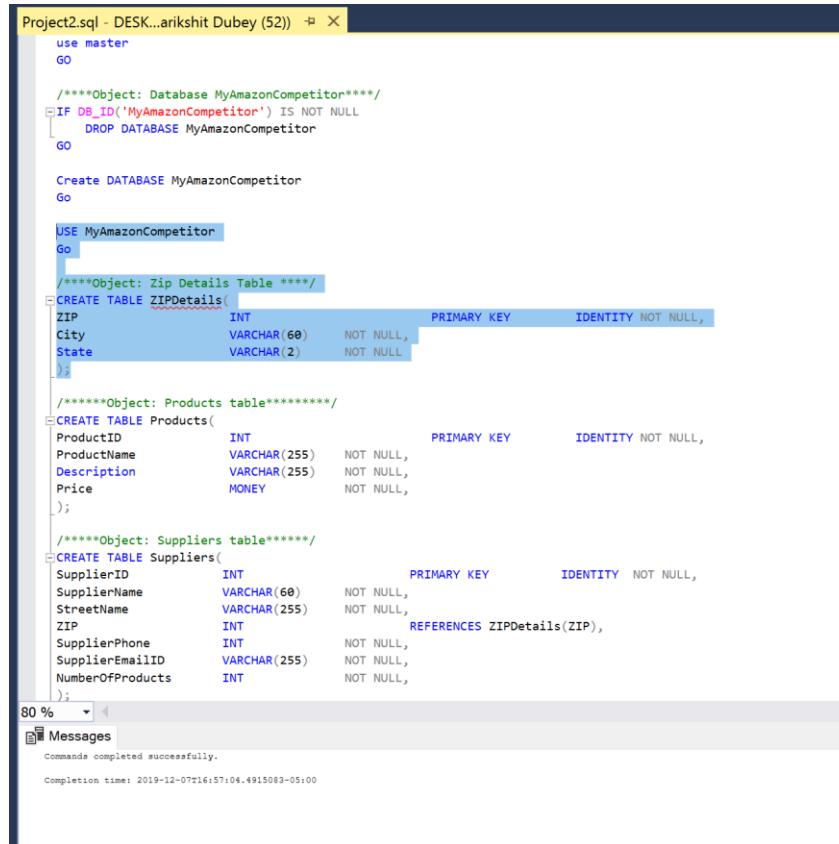


The database has been created.

Project2

Parikshit Dubey
265002524

2. Creating the table named ZIPDetails



```

Project2.sql - DESK...parikshit Dubey (52)  X
use master
GO

/**Object: Database MyAmazonCompetitor***/
IF DB_ID('MyAmazonCompetitor') IS NOT NULL
    DROP DATABASE MyAmazonCompetitor
GO

Create DATABASE MyAmazonCompetitor
Go

USE MyAmazonCompetitor
Go

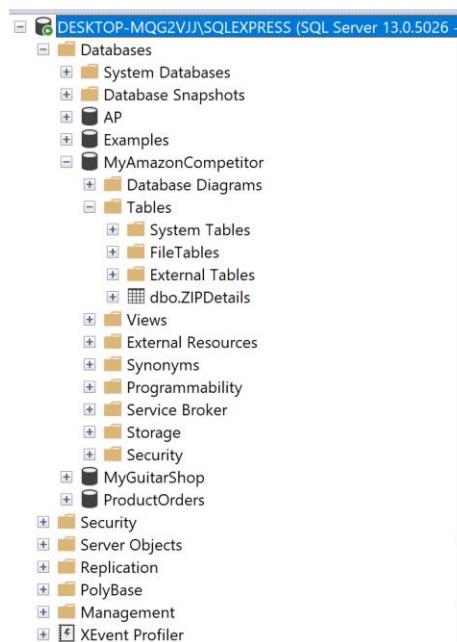
/**Object: Zip Details Table ***/
CREATE TABLE ZIPDetails(
    ZIP INT PRIMARY KEY IDENTITY NOT NULL,
    City VARCHAR(60) NOT NULL,
    State VARCHAR(2) NOT NULL
);

/**Object: Products table****/
CREATE TABLE Products(
    ProductID INT PRIMARY KEY IDENTITY NOT NULL,
    ProductName VARCHAR(255) NOT NULL,
    Description VARCHAR(255) NOT NULL,
    Price MONEY NOT NULL,
);

/**Object: Suppliers table****/
CREATE TABLE Suppliers(
    SupplierID INT PRIMARY KEY IDENTITY NOT NULL,
    SupplierName VARCHAR(60) NOT NULL,
    StreetName VARCHAR(255) NOT NULL,
    ZIP INT REFERENCES ZIPDetails(ZIP),
    SupplierPhone INT NOT NULL,
    SupplierEmailID VARCHAR(255) NOT NULL,
    NumberOfProducts INT NOT NULL
);

80 %  ◀
Messages
Commands completed successfully.
Completion time: 2019-12-07t16:57:04.4915083-05:00

```



The ZIPDetails table has been created.

Project2

Parikshit Dubey
265002524

3. Creating the table named Products

```

use master
GO

/**Object: Database MyAmazonCompetitor****/
IF DB_ID('MyAmazonCompetitor') IS NOT NULL
    DROP DATABASE MyAmazonCompetitor
GO

Create DATABASE MyAmazonCompetitor
Go

USE MyAmazonCompetitor
Go

/**Object: Zip Details Table ****/
CREATE TABLE ZIPDetails(
    ZIP INT PRIMARY KEY IDENTITY NOT NULL,
    City VARCHAR(60) NOT NULL,
    State VARCHAR(2) NOT NULL
);

/**Object: Products table*****/
CREATE TABLE Products(
    ProductID INT PRIMARY KEY IDENTITY NOT NULL,
    ProductName VARCHAR(255) NOT NULL,
    Description VARCHAR(255) NOT NULL,
    Price MONEY NOT NULL
);

/**Object: Suppliers table****/
CREATE TABLE Suppliers(
    SupplierID INT PRIMARY KEY IDENTITY NOT NULL,
    SupplierName VARCHAR(60) NOT NULL,
    StreetName VARCHAR(255) NOT NULL,
    ZIP INT REFERENCES ZIPDetails(ZIP),
    SupplierPhone INT NOT NULL,
    SupplierEmailID VARCHAR(255) NOT NULL,
    NumberOfProducts INT NOT NULL
);

Commands completed successfully.

Completion time: 2019-12-07T16:58:26.3386504-05:00

```

Project2

Parikshit Dubey
265002524

4. Creating the table named Suppliers

```

Project2.sql - DESK...arikshit Dubey (52)  ✎ X

Create DATABASE MyAmazonCompetitor
Go

USE MyAmazonCompetitor
Go

/**Object: Zip Details Table ****/
CREATE TABLE ZIPDetails(
    ZIP           INT          PRIMARY KEY      IDENTITY NOT NULL,
    City          VARCHAR(60)   NOT NULL,
    State         VARCHAR(2)    NOT NULL
);

/**Object: Products table*****/
CREATE TABLE Products(
    ProductID     INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductName   VARCHAR(255)  NOT NULL,
    Description   VARCHAR(255)  NOT NULL,
    Price         MONEY        NOT NULL,
);

/**Object: Suppliers table*****/
CREATE TABLE Suppliers(
    SupplierID    INT          PRIMARY KEY      IDENTITY NOT NULL,
    SupplierName  VARCHAR(60)   NOT NULL,
    StreetName    VARCHAR(255)  NOT NULL,
    ZIP          INT          REFERENCES ZIPDetails(ZIP),
    SupplierPhone INT          NOT NULL,
    SupplierEmailID VARCHAR(255) NOT NULL,
    NumberOfProducts INT        NOT NULL,
);

/**Object: Products Stored table *****/
CREATE TABLE ProductsStored(
    ProductsStoredID INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID       INT          REFERENCES Products(ProductID),
    NumberInStock   INT          NOT NULL,
    NumberOnWay     INT          DEFAULT NULL,
    NumberOnReturn  INT          DEFAULT NULL,
);

80 %  ◀
Messages
Commands completed successfully.
Completion time: 2019-12-07T16:58:26.3386504-05:00

```

Project2

Parikshit Dubey
265002524

5. Creating the table named ProductsStored

```

Project2.sql - DESK...arikshit Dubey (52)) ✎ X

*****Object: Products table*****
CREATE TABLE Products(
    ProductID          INT           PRIMARY KEY      IDENTITY NOT NULL,
    ProductName        VARCHAR(255)   NOT NULL,
    Description         VARCHAR(255)   NOT NULL,
    Price               MONEY          NOT NULL,
);

*****Object: Suppliers table*****
CREATE TABLE Suppliers(
    SupplierID          INT           PRIMARY KEY      IDENTITY NOT NULL,
    SupplierName        VARCHAR(60)   NOT NULL,
    StreetName          VARCHAR(255)   NOT NULL,
    ZIP                 INT           REFERENCES ZIPDetails(ZIP),
    SupplierPhone       INT           NOT NULL,
    SupplierEmailID    VARCHAR(255)   NOT NULL,
    NumberOfProducts    INT           NOT NULL,
);

*****Object: Products Stored table *****
CREATE TABLE ProductsStored(
    ProductsStoredID    INT           PRIMARY KEY      IDENTITY NOT NULL,
    ProductID          INT           REFERENCES Products(ProductID),
    NumberInStock       INT           NOT NULL,
    NumberOnWay         INT           DEFAULT NULL,
    NumberOnReturn      INT           DEFAULT NULL,
);

*****Object: Warehouse table *****
CREATE TABLE Warehouse(
    WarehouseID         INT           PRIMARY KEY      IDENTITY NOT NULL,
    StreetAddress        VARCHAR(255)   NOT NULL,
    ZIP                 INT           REFERENCES ZIPdetails(ZIP),
    ProductStoredID     INT           REFERENCES ProductsStored(ProductsStoredID),
);

*****Object: Products to Suppliers table *****
CREATE TABLE ProductsToSupplierLink(
);

```

80 % ▾ ◀

Messages
Clear

Commands completed successfully.
 Completion time: 2019-12-07T17:02:57.5719443-05:00

Project2

Parikshit Dubey
265002524

6. Creating the table named Warehouse

```

Project2.sql - DESK...arikshit Dubey (52)  X

*****Object: Suppliers table*****
CREATE TABLE Suppliers(
    SupplierID           INT          PRIMARY KEY      IDENTITY NOT NULL,
    SupplierName         VARCHAR(60)   NOT NULL,
    StreetName           VARCHAR(255)  NOT NULL,
    ZIP                  INT          REFERENCES ZIPDetails(ZIP),
    SupplierPhone        INT          NOT NULL,
    SupplierEmailID     VARCHAR(255) NOT NULL,
    NumberOfProducts    INT          NOT NULL,
);

*****Object: Products Stored table *****
CREATE TABLE ProductsStored(
    ProductsStoredID    INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID           INT          REFERENCES Products(ProductID),
    NumberInStock       INT          NOT NULL,
    NumberOnWay          INT          DEFAULT NULL,
    NumberOnReturn       INT          DEFAULT NULL,
);

*****Object: Warehouse table *****
CREATE TABLE Warehouse(
    WarehouseID          INT          PRIMARY KEY      IDENTITY NOT NULL,
    StreetAddress        VARCHAR(255) NOT NULL,
    ZIP                  INT          REFERENCES ZIPdetails(ZIP),
    ProductStoredID     INT          REFERENCES ProductsStored(ProductsStoredID),
);

*****Object: Products to Suppliers table *****
CREATE TABLE ProductsToSupplierLink(
    ProductID            INT          REFERENCES Products(ProductID),
    SupplierID           INT          REFERENCES Suppliers(SupplierID),
);

*****Object: Warehouse to Supplier link *****
CREATE TABLE WarehouseSupplierLink(
    WarehouseID          INT          REFERENCES Warehouse(WarehouseID),
    SupplierID           INT          REFERENCES Suppliers(SupplierID),
);

80 %  ▾
Messages
Commands completed successfully.

Completion time: 2019-12-07T17:05:12.5792792-05:00

```

Project2

Parikshit Dubey
265002524

7. Creating the table named ProductsToSupplierLink

```

Project2.sql - DESK...arikshit Dubey (52)  X

NumberInStock      INT          NOT NULL,
NumberOnWay        INT          DEFAULT NULL,
NumberOnReturn     INT          DEFAULT NULL,
);

/*********Object: Warehouse table *****/
CREATE TABLE Warehouse(
WarehouseID        INT          PRIMARY KEY      IDENTITY NOT NULL,
StreetAddress      VARCHAR(255) NOT NULL,
ZIP                INT          REFERENCES ZIPdetails(ZIP),
ProductStoredID   INT          REFERENCES ProductsStored(ProductsStoredID),
);

/*********Object: Products to Suppliers table *****/
CREATE TABLE ProductsToSupplierLink(
ProductID         INT          REFERENCES Products(ProductID),
SupplierID        INT          REFERENCES Suppliers(SupplierID),
);

/*********Object: Warehouse to Supplier link *****/
CREATE TABLE WarehouseSupplierLink(
WarehouseID       INT          REFERENCES Warehouse(WarehouseID),
SupplierID        INT          REFERENCES Suppliers(SupplierID),
);

/*********Object: Review to Supplier link *****/
CREATE TABLE Review(
ReviewID           INT          PRIMARY KEY      IDENTITY NOT NULL,
ProductID          INT          REFERENCES Products(ProductID),
DateOfReview       DATETIME    NOT NULL,
Comments           VARCHAR(255) DEFAULT NULL,
Score              INT          DEFAULT NULL,
CustomerName      VARCHAR(255) DEFAULT NULL,
);

/*********Object: Wishlist Table *****/
CREATE TABLE WishList(
);

80 %  ▾
Messages
Commands completed successfully.

Completion time: 2019-12-07T17:06:37.7378542-05:00

```

Project2

Parikshit Dubey
265002524

8. Creating the table named WarehouseSupplierLink

```

Project2.sql - DESK...arikshit Dubey (52)  X

NumberInStock      INT          NOT NULL,
NumberOnWay        INT          DEFAULT NULL,
NumberOnReturn     INT          DEFAULT NULL,
);

/*********Object: Warehouse table *****/
CREATE TABLE Warehouse(
WarehouseID        INT          PRIMARY KEY          IDENTITY NOT NULL,
StreetAddress      VARCHAR(255) NOT NULL,
ZIP                INT          REFERENCES ZIPdetails(ZIP),
ProductStoredID   INT          REFERENCES ProductsStored(ProductsStoredID),
);

/*********Object: Products to Suppliers table *****/
CREATE TABLE ProductsToSupplierLink(
ProductID         INT          REFERENCES Products(ProductID),
SupplierID        INT          REFERENCES Suppliers(SupplierID),
);

/*********Object: Warehouse to Supplier link *****/
CREATE TABLE WarehouseSupplierLink(
WarehouseID        INT          REFERENCES Warehouse(WarehouseID),
SupplierID        INT          REFERENCES Suppliers(SupplierID),
);

/*********Object: Review to Supplier link *****/
CREATE TABLE Review(
ReviewID           INT          PRIMARY KEY          IDENTITY NOT NULL,
ProductID          INT          REFERENCES Products(ProductID),
DateOfReview       DATETIME    NOT NULL,
Comments           VARCHAR(255) DEFAULT NULL,
);

100 %  <  Messages
Commands completed successfully.

Completion time: 2019-12-07T17:07:50.6636991-05:00

```

Project2

Parikshit Dubey
265002524

9. Creating the table named Review

```

Project2.sql - DESK...arikshit Dubey (52)*  ✎ X

*****Object: Products to Suppliers table *****/
CREATE TABLE ProductsToSupplierLink(
    ProductID      INT          REFERENCES Products(ProductID),
    SupplierID     INT          REFERENCES Suppliers(SupplierID),
);

*****Object: Warehouse to Supplier link *****/
CREATE TABLE WarehouseSupplierLink(
    WarehouseID    INT          REFERENCES Warehouse(WarehouseID),
    SupplierID     INT          REFERENCES Suppliers(SupplierID),
);

*****Object: Review table *****/
CREATE TABLE Review(
    ReviewID       INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID     INT          REFERENCES Products(ProductID),
    DateOfReview   DATETIME     NOT NULL,
    Comments       VARCHAR(255)  DEFAULT NULL,
    Score          INT          DEFAULT NULL,
    CustomerName   VARCHAR(255)  DEFAULT NULL,
);

*****Object: Wishlist Table *****/
CREATE TABLE WishList(
    WishListID     INT          PRIMARY KEY      IDENTITY NOT NULL,
    Date           DATETIME     DEFAULT NULL,
    ProductID      INT          REFERENCES PRODUCTS(ProductID),
    Quantity        INT          NOT NULL
);

100 %  ◀
Messages
Commands completed successfully.

Completion time: 2019-12-07T17:10:05.4950205-05:00

```

Project2

Parikshit Dubey
265002524

10. Creating the table named WishList

```

Project2.sql - DESK...arikshit Dubey (52)*  ✎ X

ProductID      INT          REFERENCES Products(ProductID),
DateOfReview   DATETIME    NOT NULL,
Comments       VARCHAR(255) DEFAULT NULL,
Score          INT          DEFAULT NULL,
CustomerName   VARCHAR(255) DEFAULT NULL,
);

/* ****Object: Wishlist Table ****/
CREATE TABLE Wishlist(
WishListID     INT          PRIMARY KEY IDENTITY NOT NULL,
Date           DATETIME    DEFAULT NULL,
ProductID      INT          REFERENCES PRODUCTS(ProductID),
Quantity       INT          NOT NULL
);

/* ****Object: Customers Table ****/
CREATE TABLE Customers(
CustomerID     INT          PRIMARY KEY IDENTITY NOT NULL,
CustomerName   VARCHAR(60)  NOT NULL,
username        VARCHAR(60)  NOT NULL UNIQUE,
CustomerEmail  VARCHAR(255) NOT NULL UNIQUE,
CustomerPhone  INT          NOT NULL,
ZIP            INT          REFERENCES ZIPDetails(ZIP),
ReturnID       INT          NOT NULL, /*ADD REFERENCE*/
WishListID     INT          REFERENCES Wishlist(WishListID),
ReviewID       INT          REFERENCES Review(ReviewID)
);

100 %  ◀
Messages
Commands completed successfully.

Completion time: 2019-12-07T17:11:03.0215086-05:00

```

Project2

Parikshit Dubey
265002524

11. Creating table named Customers

```

Project2.sql - DESK...parikshit Dubey (52)* ➔ X

****Object: Wishlist Table ****/
CREATE TABLE Wishlist(
    WishlistID      INT          PRIMARY KEY      IDENTITY      NOT NULL,
    Date            DATETIME     DEFAULT NULL,
    ProductID       INT          REFERENCES PRODUCTS(ProductID),
    Quantity        INT          NOT NULL
);

****Object: Customers Table ****/
CREATE TABLE Customers(
    CustomerID      INT          PRIMARY KEY      IDENTITY NOT NULL,
    CustomerName    VARCHAR(60)   NOT NULL,
    username        VARCHAR(60)   NOT NULL UNIQUE,
    CustomerEmail   VARCHAR(255)  NOT NULL UNIQUE,
    CustomerPhone   INT          NOT NULL,
    ZIP             INT          REFERENCES ZIPDetails(ZIP),
    ReturnID        INT          NOT NULL, /*ADD REFERENCE*/
    WishListID      INT          REFERENCES Wishlist(WishListID),
    ReviewID        INT          REFERENCES Review(ReviewID)
);

*****Object: OrderItems table*****
CREATE TABLE OrderItems(
    OrderItemsID    INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID       INT          REFERENCES Products(ProductID),
    Quantity        INT          NOT NULL,
    UnitPrice       MONEY        NOT NULL,
);

100 % ▶
Messages
Commands completed successfully.
Completion time: 2019-12-07T17:12:28.7574223-05:00

```

Project2

Parikshit Dubey
265002524

12. Creating table named OrderItems

```

Project2.sql - DESK...parikshit Dubey (52)*  X

CustomerEmail      VARCHAR(255)    NOT NULL  UNIQUE,
CustomerPhone      INT            NOT NULL,
ZIP                INT            REFERENCES ZIPDetails(ZIP),
ReturnID           INT            NOT NULL,          /*ADD REFERENCE*/
WishListID         INT            REFERENCES WishList(WishListID),
ReviewID           INT            REFERENCES Review(ReviewID)
);

/*********Object: OrderItems table*****/
CREATE TABLE OrderItems(
OrderItemsID        INT            PRIMARY KEY      IDENTITY NOT NULL,
ProductID          INT            REFERENCES Products(ProductID),
Quantity            INT            NOT NULL,
UnitPrice           MONEY          NOT NULL,
CumulativeCost     MONEY          NOT NULL
);

/*********Object: Return Items table*****/
CREATE TABLE ReturnItems(
ReturnItemsID       INT            PRIMARY KEY      IDENTITY NOT NULL,
ProductID          INT            REFERENCES Products(ProductID),
Quantity           INT            DEFAULT NULL
);

/*********Object: Return details*****/
CREATE TABLE ReturnDetails(
ReturnID            INT            PRIMARY KEY      IDENTITY NOT NULL,
ReturnItemsID       INT            REFERENCES ReturnItems(ReturnItemsID),
DateOfReturn        DATETIME        DEFAULT NULL,
StatusOfReturn      VARCHAR(255)   DEFAULT NULL,
);

100 %  ◀
Messages
Commands completed successfully.

Completion time: 2019-12-07T17:13:31.7694646-05:00

```

Project2

Parikshit Dubey
265002524

13. Creating table named ReturnItems

```

Project2.sql - DESK...parikshit Dubey (52)* X

/*********Object: OrderItems table*****/
CREATE TABLE OrderItems(
    OrderItemsID      INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID        INT          REFERENCES Products(ProductID),
    Quantity          INT          NOT NULL,
    UnitPrice         MONEY        NOT NULL,
    CumulativeCost   MONEY        NOT NULL
);

/*********Object: Return Items table*****/
CREATE TABLE ReturnItems(
    ReturnItemsID     INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID        INT          REFERENCES Products(ProductID),
    Quantity          INT          DEFAULT NULL
);

/*********Object: Return details*****/
CREATE TABLE ReturnDetails(
    ReturnID          INT          PRIMARY KEY      IDENTITY NOT NULL,
    ReturnItemsID     INT          REFERENCES ReturnItems(ReturnItemsID),
    DateOfReturn      DATETIME     DEFAULT NULL,
    StatusOfReturn    VARCHAR(255) DEFAULT NULL,
    ShippingInfo      VARCHAR(255) DEFAULT NULL,
    ShippingCost      Money        DEFAULT NULL
);

****Object: OrdersTable ****/
CREATE TABLE OrdersTable(
    OrderID           INT          PRIMARY KEY      IDENTITY NOT NULL,
    CustomerId        INT          REFERENCES Customers(CustomerID),
);

```

100 %

Messages

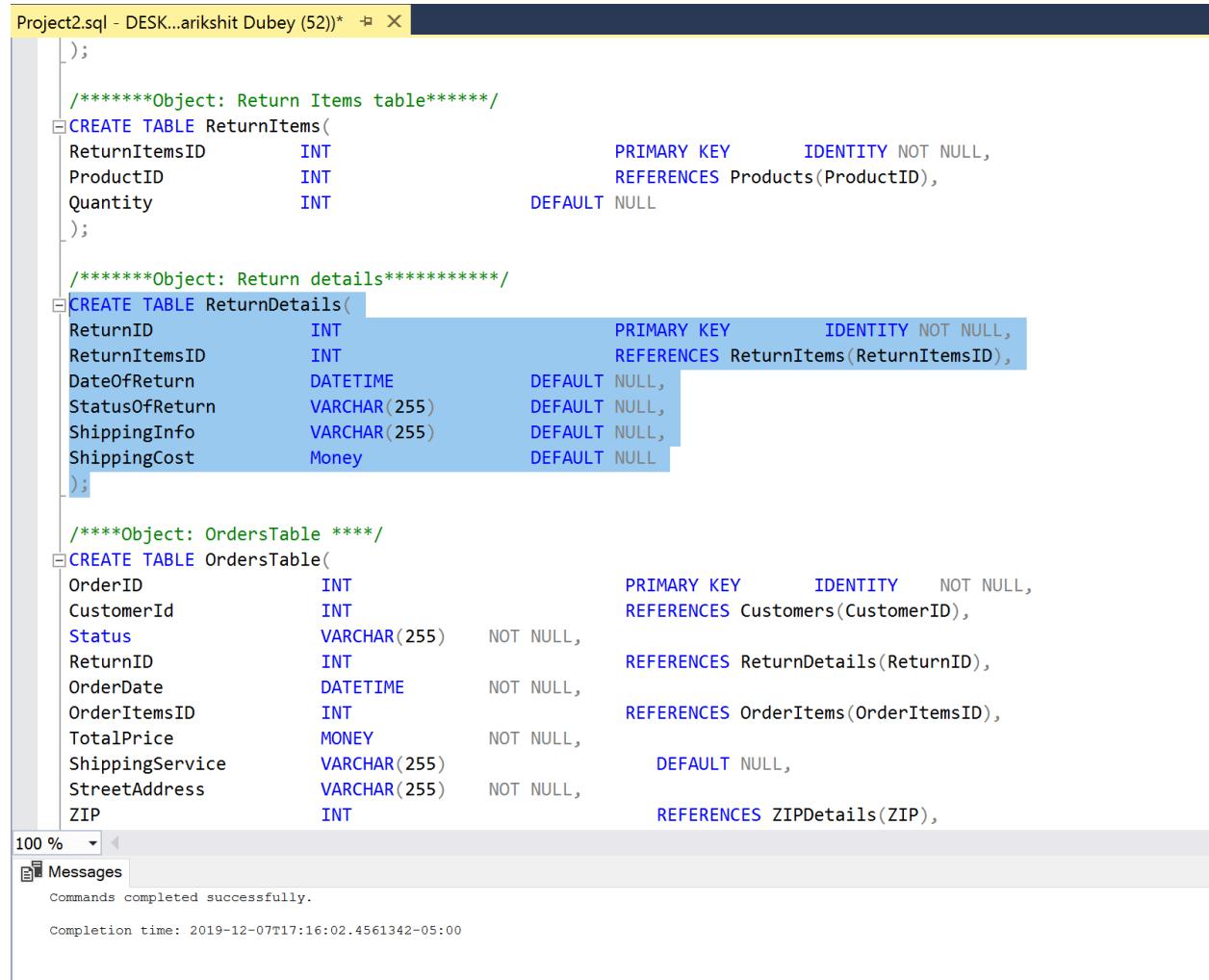
Commands completed successfully.

Completion time: 2019-12-07T17:14:33.5099779-05:00

Project2

Parikshit Dubey
265002524

14. Creating table named ReturnDetails



```

Project2.sql - DESK...parikshit Dubey (52)* X

);
/*********Object: Return Items table*******/
CREATE TABLE ReturnItems(
    ReturnItemsID      INT          PRIMARY KEY      IDENTITY NOT NULL,
    ProductID         INT          REFERENCES Products(ProductID),
    Quantity          INT          DEFAULT NULL
);

/*********Object: Return details*******/
CREATE TABLE ReturnDetails(
    ReturnID          INT          PRIMARY KEY      IDENTITY NOT NULL,
    ReturnItemsID     INT          REFERENCES ReturnItems(ReturnItemsID),
    DateOfReturn      DATETIME    DEFAULT NULL,
    StatusOfReturn    VARCHAR(255) DEFAULT NULL,
    ShippingInfo      VARCHAR(255) DEFAULT NULL,
    ShippingCost      MONEY       DEFAULT NULL
);

****Object: OrdersTable ****/
CREATE TABLE OrdersTable(
    OrderID           INT          PRIMARY KEY      IDENTITY NOT NULL,
    CustomerId        INT          REFERENCES Customers(CustomerID),
    Status             VARCHAR(255) NOT NULL,
    ReturnID          INT          REFERENCES ReturnDetails(ReturnID),
    OrderDate         DATETIME    NOT NULL,
    OrderItemsID      INT          REFERENCES OrderItems(OrderItemsID),
    TotalPrice        MONEY       NOT NULL,
    ShippingService   VARCHAR(255) DEFAULT NULL,
    StreetAddress     VARCHAR(255) NOT NULL,
    ZIP               INT          REFERENCES ZIPDetails(ZIP),
);

100 % ▾
Messages
Commands completed successfully.
Completion time: 2019-12-07T17:16:02.4561342-05:00

```

Project2

Parikshit Dubey
265002524

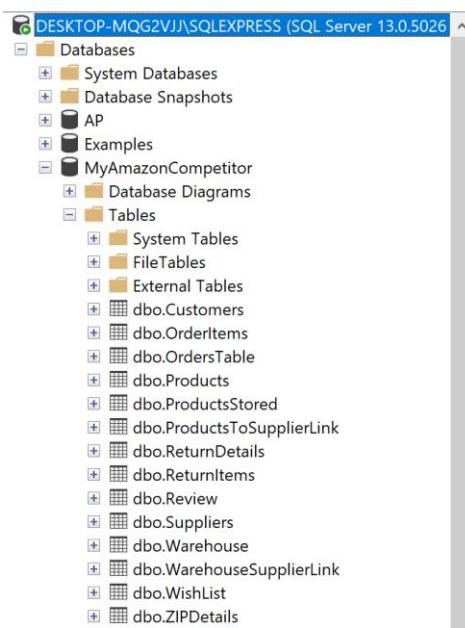
15. Creating table named OrdersTable

```

Project2.sql - DESK..arikshit Dubey (52)  ↗ X
DateOfReturn      DATETIME          DEFAULT NULL,
StatusOfReturn    VARCHAR(255)      DEFAULT NULL,
ShippingInfo      VARCHAR(255)      DEFAULT NULL,
ShippingCost      Money             DEFAULT NULL
);

/***/Object: OrdersTable ****/
CREATE TABLE OrdersTable(
    OrderID          INT               PRIMARY KEY      IDENTITY NOT NULL,
    CustomerId       INT               REFERENCES Customers(CustomerID),
    Status           VARCHAR(255)     NOT NULL,
    ReturnID         INT               REFERENCES ReturnDetails(ReturnID),
    OrderDate        DATETIME        NOT NULL,
    OrderItemsID     INT               REFERENCES OrderItems(OrderItemsID),
    TotalPrice       MONEY            NOT NULL,
    ShippingService  VARCHAR(255)      DEFAULT NULL,
    StreetAddress    VARCHAR(255)      NOT NULL,
    ZIP              INT               REFERENCES ZIPDetails(ZIP),
    ShippingFare     MONEY            NOT NULL,
    ExpectedShippingDate DATETIME      NOT NULL,
    ActualShippingDate DATETIME      DEFAULT NULL,
    ShippingInfo     VARCHAR(255)      DEFAULT NULL
);

```



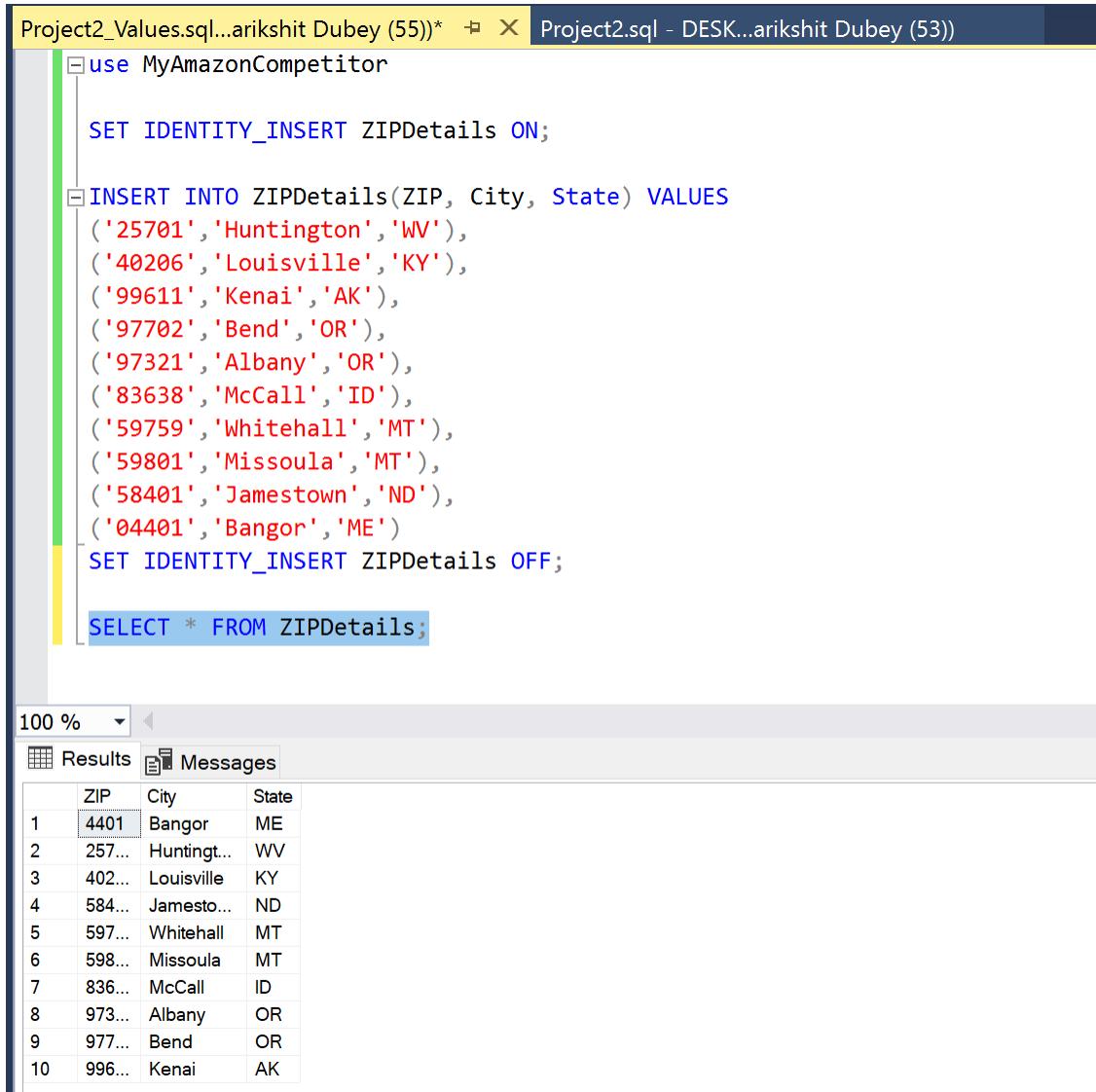
The above screenshot shows all the tables were created successfully in the MyAmazonCompetitor database.

Project2

Parikshit Dubey
265002524

Populating the tables

1. Populating the ZIPDetails table



```

Project2_Values.sql...arikshit Dubey (55)*  X  Project2.sql - DESK...arikshit Dubey (53))

use MyAmazonCompetitor

SET IDENTITY_INSERT ZIPDetails ON;

INSERT INTO ZIPDetails(ZIP, City, State) VALUES
('25701', 'Huntington', 'WV'),
('40206', 'Louisville', 'KY'),
('99611', 'Kenai', 'AK'),
('97702', 'Bend', 'OR'),
('97321', 'Albany', 'OR'),
('83638', 'McCall', 'ID'),
('59759', 'Whitehall', 'MT'),
('59801', 'Missoula', 'MT'),
('58401', 'Jamestown', 'ND'),
('04401', 'Bangor', 'ME')
SET IDENTITY_INSERT ZIPDetails OFF;

SELECT * FROM ZIPDetails;

```

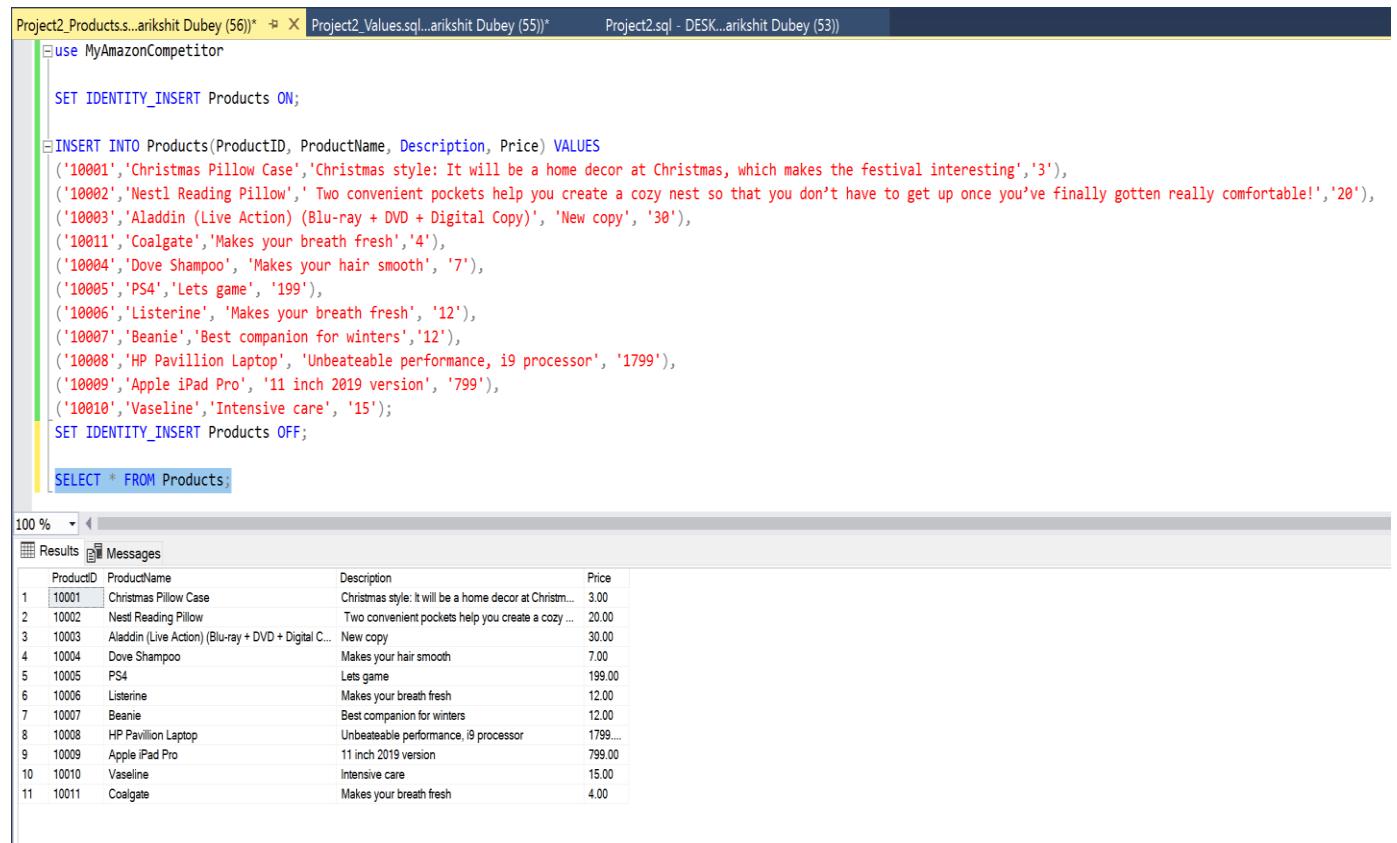
The screenshot shows the results of the executed SQL script. The 'Results' tab is selected, displaying a table with three columns: ZIP, City, and State. The data consists of 10 rows, each representing a ZIP code, its corresponding city, and its state.

	ZIP	City	State
1	4401	Bangor	ME
2	257...	Huntingt...	WV
3	402...	Louisville	KY
4	584...	Jamesto...	ND
5	597...	Whitehall	MT
6	598...	Missoula	MT
7	836...	McCall	ID
8	973...	Albany	OR
9	977...	Bend	OR
10	996...	Kenai	AK

Project2

Parikshit Dubey
265002524

2. Populating the Products table



```

Project2_Products.s...arkshit Dubey (56)*  Project2_Values.sql...arkshit Dubey (55)*  Project2.sql - DESK...arkshit Dubey (53)
use MyAmazonCompetitor

SET IDENTITY_INSERT Products ON;

INSERT INTO Products(ProductID, ProductName, Description, Price) VALUES
('10001', 'Christmas Pillow Case', 'Christmas style: It will be a home decor at Christmas, which makes the festival interesting', '3'),
('10002', 'Nestl Reading Pillow', ' Two convenient pockets help you create a cozy nest so that you don\'t have to get up once you\'ve finally gotten really comfortable!', '20'),
('10003', 'Aladdin (Live Action) (Blu-ray + DVD + Digital Copy)', 'New copy', '30'),
('10011', 'Coalgate', 'Makes your breath fresh', '4'),
('10004', 'Dove Shampoo', 'Makes your hair smooth', '7'),
('10005', 'PS4', 'Lets game', '199'),
('10006', 'Listerine', 'Makes your breath fresh', '12'),
('10007', 'Beanie', 'Best companion for winters', '12'),
('10008', 'HP Pavilion Laptop', 'Unbeatable performance, i9 processor', '1799'),
('10009', 'Apple iPad Pro', '11 inch 2019 version', '799'),
('10010', 'Vaseline', 'Intensive care', '15');

SET IDENTITY_INSERT Products OFF;

SELECT * FROM Products;

```

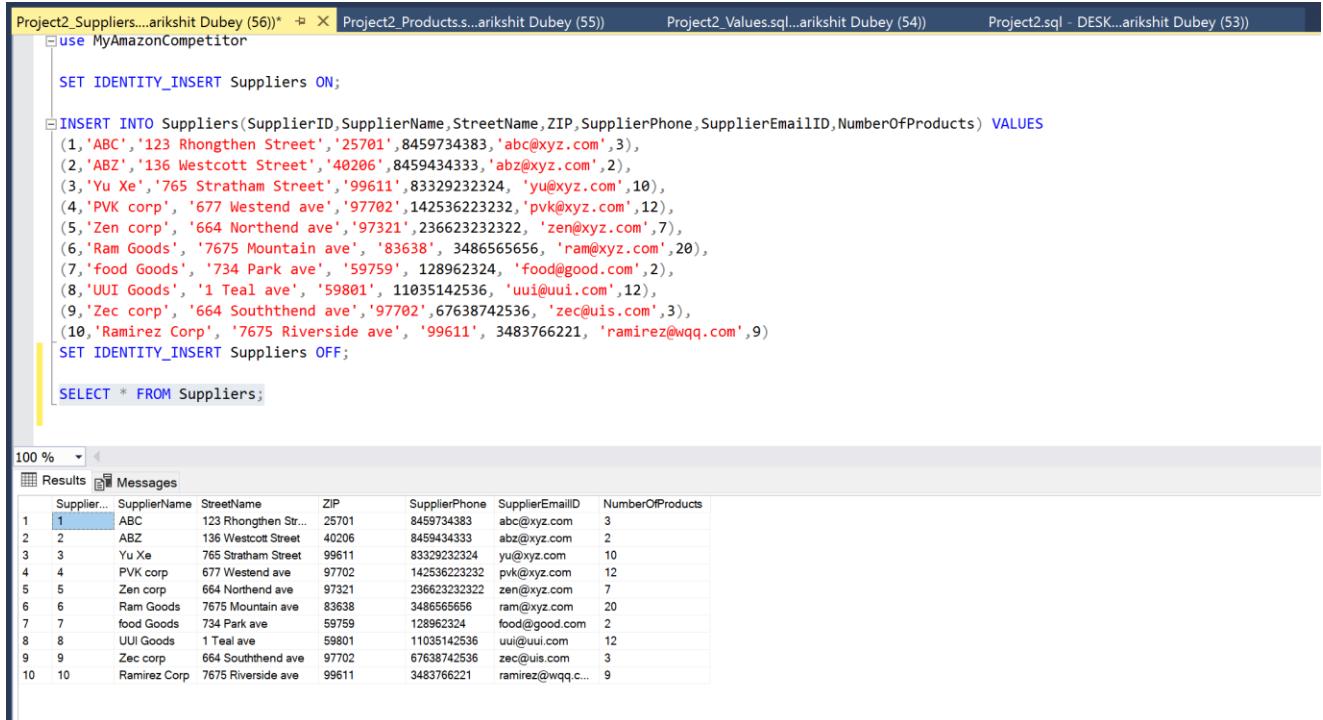
Results

	ProductID	ProductName	Description	Price
1	10001	Christmas Pillow Case	Christmas style: It will be a home decor at Christm...	3.00
2	10002	Nestl Reading Pillow	Two convenient pockets help you create a cozy ...	20.00
3	10003	Aladdin (Live Action) (Blu-ray + DVD + Digital C...	New copy	30.00
4	10004	Dove Shampoo	Makes your hair smooth	7.00
5	10005	PS4	Lets game	199.00
6	10006	Listerine	Makes your breath fresh	12.00
7	10007	Beanie	Best companion for winters	12.00
8	10008	HP Pavilion Laptop	Unbeatable performance, i9 processor	1799...
9	10009	Apple iPad Pro	11 inch 2019 version	799.00
10	10010	Vaseline	Intensive care	15.00
11	10011	Coalgate	Makes your breath fresh	4.00

Project2

Parikshit Dubey
265002524

3. Populating the Suppliers table



```

Project2_Suppliers...arkshit Dubey (56)* -> X Project2_Products...arkshit Dubey (55) Project2_Values.sql...arkshit Dubey (54) Project2.sql - DESK...arkshit Dubey (53)
use MyAmazonCompetitor

SET IDENTITY_INSERT Suppliers ON;

INSERT INTO Suppliers(SupplierID,SupplierName,StreetName,ZIP,SupplierPhone,SupplierEmailID,NumberOfProducts) VALUES
(1,'ABC','123 Rhongthen Street','25701',8459734383,'abc@xyz.com',3),
(2,'ABZ','136 Westcott Street','40206',8459434333,'abz@xyz.com',2),
(3,'Yu Xe','765 Stratham Street','99611',83329232324,'yu@xyz.com',10),
(4,'PVK corp','677 Westend ave','97702',1425362232322,'pvk@xyz.com',12),
(5,'Zen corp','664 Northend ave','97321',236623232322,'zen@xyz.com',7),
(6,'Ram Goods','7675 Mountain ave','83638',3486565656,'ram@xyz.com',20),
(7,'Food Goods','734 Park ave','59759',128962324,'food@good.com',2),
(8,'UUI Goods','1 Teal ave','59801',11035142536,'uui@uui.com',12),
(9,'Zec corp','664 Souththend ave','97702',67638742536,'zec@uis.com',3),
(10,'Ramirez Corp','7675 Riverside ave','99611',3483766221,'ramirez@wqq.com',9)

SET IDENTITY_INSERT Suppliers OFF;

SELECT * FROM Suppliers;

```

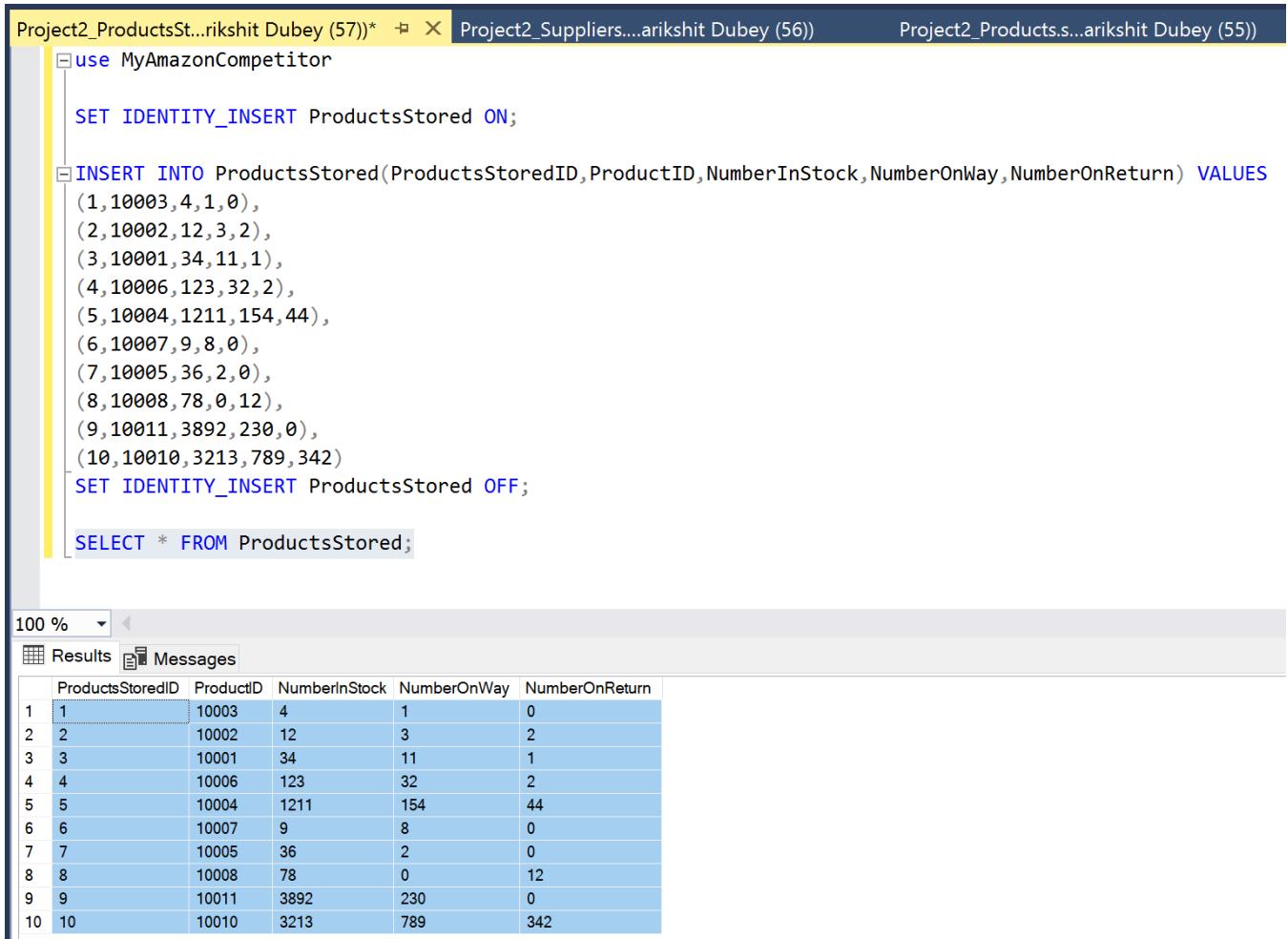
The screenshot shows the results of the executed query. The table has columns: SupplierID, SupplierName, StreetName, ZIP, SupplierPhone, SupplierEmailID, and NumberOfProducts. The data is as follows:

SupplierID	SupplierName	StreetName	ZIP	SupplierPhone	SupplierEmailID	NumberOfProducts
1	ABC	123 Rhongthen Str...	25701	8459734383	abc@xyz.com	3
2	ABZ	136 Westcott Street	40206	8459434333	abz@xyz.com	2
3	Yu Xe	765 Stratham Street	99611	83329232324	yu@xyz.com	10
4	PVK corp	677 Westend ave	97702	1425362232322	pvk@xyz.com	12
5	Zen corp	664 Northend ave	97321	236623232322	zen@xyz.com	7
6	Ram Goods	7675 Mountain ave	83638	3486565656	ram@xyz.com	20
7	Food Goods	734 Park ave	59759	128962324	food@good.com	2
8	UUI Goods	1 Teal ave	59801	11035142536	uui@uui.com	12
9	Zec corp	664 Souththend ave	97702	67638742536	zec@uis.com	3
10	Ramirez Corp	7675 Riverside ave	99611	3483766221	ramirez@wqq.c...	9

Project2

Parikshit Dubey
265002524

4. Populating the ProductsStored table



```

Project2_ProductsSt...rikshit Dubey (57)*  X  Project2_Suppliers....arikshit Dubey (56)          Project2_Products.s...arikshit Dubey (55)

use MyAmazonCompetitor

SET IDENTITY_INSERT ProductsStored ON;

INSERT INTO ProductsStored(ProductsStoredID,ProductID,NumberInStock,NumberOnWay,NumberOnReturn) VALUES
(1,10003,4,1,0),
(2,10002,12,3,2),
(3,10001,34,11,1),
(4,10006,123,32,2),
(5,10004,1211,154,44),
(6,10007,9,8,0),
(7,10005,36,2,0),
(8,10008,78,0,12),
(9,10011,3892,230,0),
(10,10010,3213,789,342)
SET IDENTITY_INSERT ProductsStored OFF;

SELECT * FROM ProductsStored;

```

The screenshot shows a SQL query being run in SSMS. The code uses the `SET IDENTITY_INSERT` command to enable identity insert mode for the `ProductsStored` table. It then inserts 10 rows of data with specific values for ProductID, NumberInStock, NumberOnWay, and NumberOnReturn. Finally, it selects all columns from the `ProductsStored` table.

	ProductsStoredID	ProductID	NumberInStock	NumberOnWay	NumberOnReturn
1	1	10003	4	1	0
2	2	10002	12	3	2
3	3	10001	34	11	1
4	4	10006	123	32	2
5	5	10004	1211	154	44
6	6	10007	9	8	0
7	7	10005	36	2	0
8	8	10008	78	0	12
9	9	10011	3892	230	0
10	10	10010	3213	789	342

Project2

Parikshit Dubey
265002524

5. Populating the Warehouse table

The screenshot shows a SQL query window titled "Warehouse.sql - DE...rikshit Dubey (58)*" with a status bar indicating "Project2_ProductsSt...rikshit Dubey (57)". The code in the window is:

```
use MyAmazonCompetitor

SET IDENTITY_INSERT Warehouse ON;
INSERT INTO Warehouse(WarehouseID,StreetAddress,ZIP) VALUES
(1,'123 gthen Street','25701'),
(2,'136 cott Street','40206'),
(3,'765 Tham Street','99611'),
(4,'677 Endstreet ave','97702'),
(5,'664 EndRoad ave','97321'),
(6,'7675 Mountain Clark ave', '83638'),
(7,'734 Park ave', '59759'),
(8,'1 Teal ave', '59801'),
(9,'664 Souththend ave', '97702'),
(10,'7675 Riverside ave', '99611')
SET IDENTITY_INSERT Warehouse OFF;

SELECT * FROM Warehouse;
```

The results pane below shows the data inserted into the Warehouse table:

	Warehouse...	StreetAddress	ZIP
1	1	123 gthen Street	25701
2	2	136 cott Street	40206
3	3	765 Tham Street	99611
4	4	677 Endstreet ave	97702
5	5	664 EndRoad ave	97321
6	6	7675 Mountain Clark ...	83638
7	7	734 Park ave	59759
8	8	1 Teal ave	59801
9	9	664 Souththend ave	97702
10	10	7675 Riverside ave	99611

Project2

Parikshit Dubey
265002524

6. Populating the WarehouseToProductsStored table

Warehouse.sql - DE...rikshit Dubey (55) WarehouseProductSt...ikshit Dubey (54)* X Project

```
USE MyAmazonCompetitor

INSERT INTO WarehouseToProductsStored(WarehouseID,ProductsStoredID) VALUES
(1,1),
(1,2),
(1,10),
(1,8),
(2,9),
(2,7),
(1,5),
(3,5),
(7,3),
(7,1),
(3,6),
(10,3),
(10,10),
(9,4),
(8,7);

SELECT * From WarehouseToProductsStored;
```

100 %

	Warehouse...	ProductsStoredID
1	1	1
2	1	2
3	1	10
4	1	8
5	2	9
6	2	7
7	1	5
8	3	5
9	7	3
10	7	1
11	3	6
12	10	3
13	10	10
14	9	4
15	8	7

Project2

Parikshit Dubey
265002524

7. Populating the ProductsToSupplierLink table

The screenshot shows a SQL query being run in a database context. The top bar indicates the current database is 'Project2' and the schema is 'Products'. The query itself is an INSERT INTO statement for the 'ProductsToSupplierLink' table, followed by a SELECT statement to verify the data.

```
USE MyAmazonCompetitor
INSERT INTO ProductsToSupplierLink(ProductID,SupplierID) VALUES
(10001,1),
(10004,2),
(10002,10),
(10001,8),
(10002,9),
(10002,7),
(10001,5),
(10003,5),
(10007,3),
(10007,1),
(10003,6),
(10010,3),
(10010,10),
(10009,4),
(10008,7);
SELECT * From ProductsToSupplierLink;
```

The results pane shows a table with two columns: ProductID and SupplierID. The data consists of 15 rows, each mapping a product ID to a supplier ID.

	ProductID	SupplierID
1	10001	1
2	10004	2
3	10002	10
4	10001	8
5	10002	9
6	10002	7
7	10001	5
8	10003	5
9	10007	3
10	10007	1
11	10003	6
12	10010	3
13	10010	10
14	10009	4
15	10008	7

Project2

Parikshit Dubey
265002524

8. Populating the WarehouseSupplierLink table

The screenshot shows a SQL query being run in SSMS. The code is as follows:

```
USE MyAmazonCompetitor
INSERT INTO WarehouseSupplierLink(WarehouseID,SupplierID) VALUES
(1,1),
(1,2),
(1,10),
(1,8),
(2,9),
(2,4),
(1,7),
(3,8),
(7,2),
(7,5),
(3,6),
(10,9),
(10,10),
(9,4),
(8,7);
SELECT * From WarehouseSupplierLink;
```

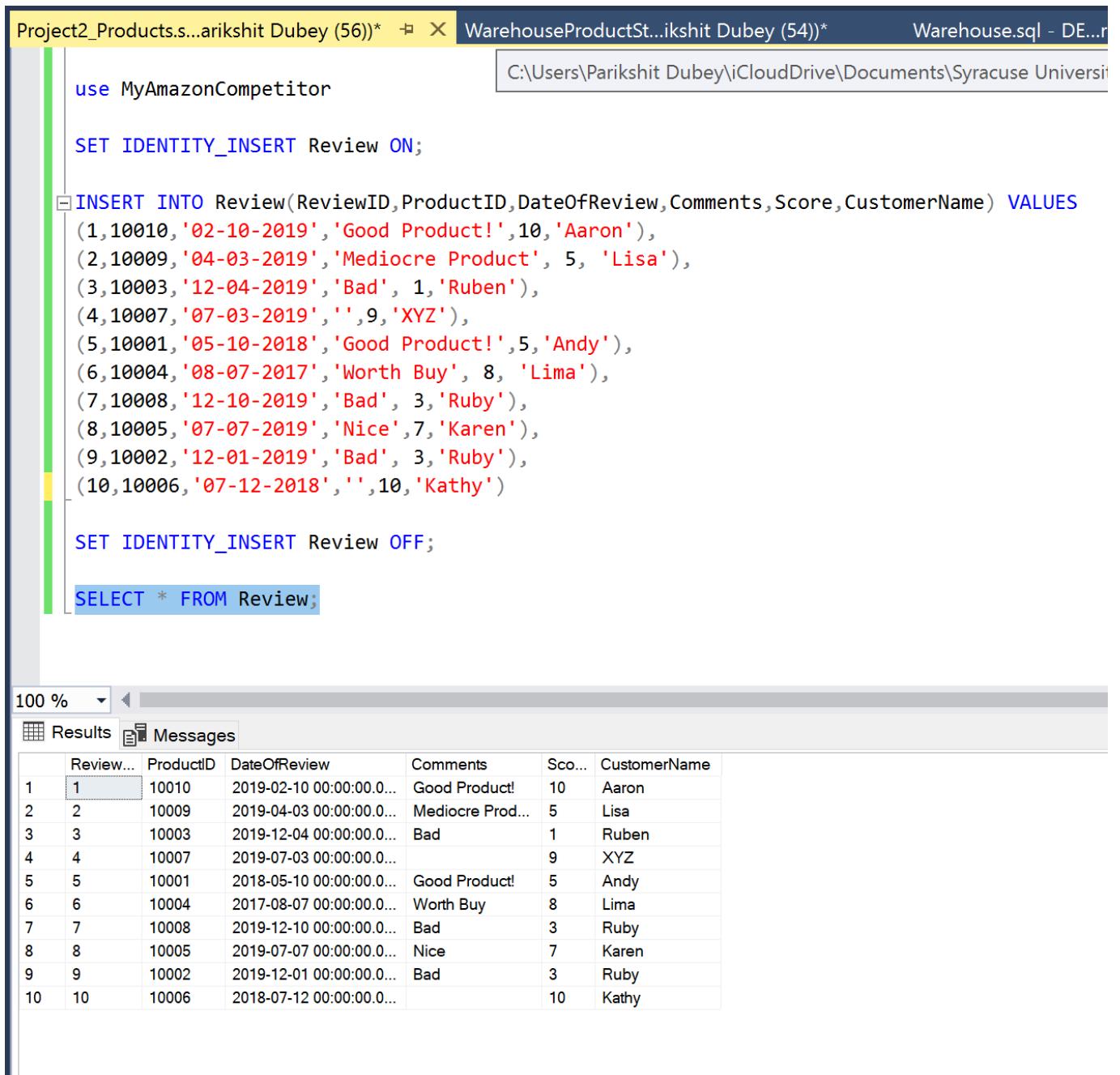
The results pane shows the following data:

	WarehouseID	SupplierID
1	1	1
2	1	2
3	1	10
4	1	8
5	2	9
6	2	4
7	1	7
8	3	8
9	7	2
10	7	5
11	3	6
12	10	9
13	10	10
14	9	4
15	8	7

Project2

Parikshit Dubey
265002524

9. Populating the Review table



```

Project2_Products.s...arikshit Dubey (56)*  WarehouseProductSt...ikshit Dubey (54))*
Warehouse.sql - DE...r

use MyAmazonCompetitor
SET IDENTITY_INSERT Review ON;

INSERT INTO Review(ReviewID, ProductID, DateOfReview, Comments, Score, CustomerName) VALUES
(1,10010, '02-10-2019', 'Good Product!', 10, 'Aaron'),
(2,10009, '04-03-2019', 'Mediocre Product', 5, 'Lisa'),
(3,10003, '12-04-2019', 'Bad', 1, 'Ruben'),
(4,10007, '07-03-2019', '', 9, 'XYZ'),
(5,10001, '05-10-2018', 'Good Product!', 5, 'Andy'),
(6,10004, '08-07-2017', 'Worth Buy', 8, 'Lima'),
(7,10008, '12-10-2019', 'Bad', 3, 'Ruby'),
(8,10005, '07-07-2019', 'Nice', 7, 'Karen'),
(9,10002, '12-01-2019', 'Bad', 3, 'Ruby'),
(10,10006, '07-12-2018', '', 10, 'Kathy')

SET IDENTITY_INSERT Review OFF;

SELECT * FROM Review;

```

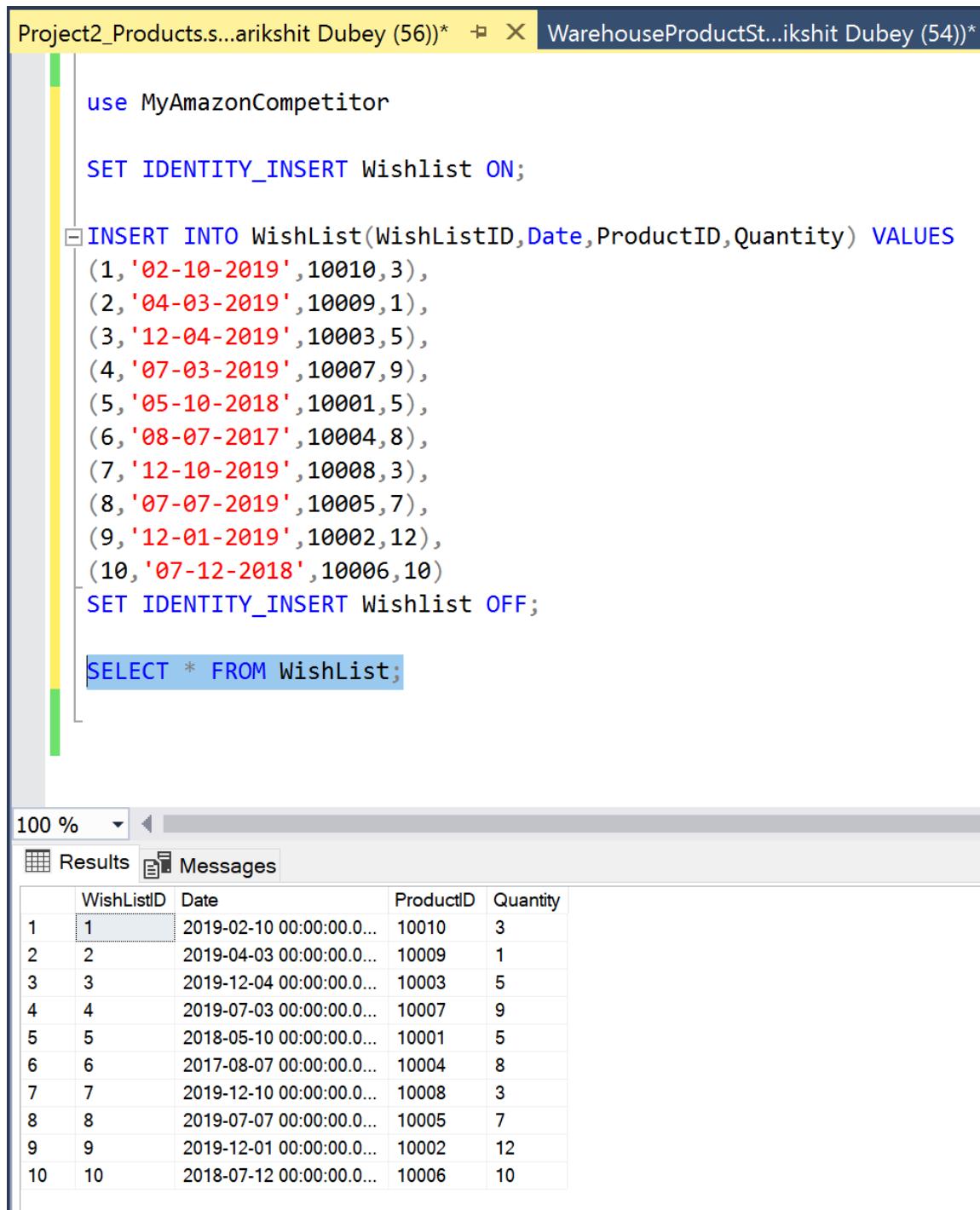
The screenshot shows the results of the executed SQL script. The 'Results' tab is selected, displaying a table with 10 rows of data inserted into the 'Review' table. The columns are: ReviewID, ProductID, DateOfReview, Comments, Score, and CustomerName.

	ReviewID	ProductID	DateOfReview	Comments	Score	CustomerName
1	1	10010	2019-02-10 00:00:00.000	Good Product!	10	Aaron
2	2	10009	2019-04-03 00:00:00.000	Mediocre Prod...	5	Lisa
3	3	10003	2019-12-04 00:00:00.000	Bad	1	Ruben
4	4	10007	2019-07-03 00:00:00.000		9	XYZ
5	5	10001	2018-05-10 00:00:00.000	Good Product!	5	Andy
6	6	10004	2017-08-07 00:00:00.000	Worth Buy	8	Lima
7	7	10008	2019-12-10 00:00:00.000	Bad	3	Ruby
8	8	10005	2019-07-07 00:00:00.000	Nice	7	Karen
9	9	10002	2019-12-01 00:00:00.000	Bad	3	Ruby
10	10	10006	2018-07-12 00:00:00.000		10	Kathy

Project2

Parikshit Dubey
265002524

10. Populating the WishList table



```

use MyAmazonCompetitor

SET IDENTITY_INSERT Wishlist ON;

INSERT INTO Wishlist(WishListID, Date, ProductID, Quantity) VALUES
(1, '02-10-2019', 10010, 3),
(2, '04-03-2019', 10009, 1),
(3, '12-04-2019', 10003, 5),
(4, '07-03-2019', 10007, 9),
(5, '05-10-2018', 10001, 5),
(6, '08-07-2017', 10004, 8),
(7, '12-10-2019', 10008, 3),
(8, '07-07-2019', 10005, 7),
(9, '12-01-2019', 10002, 12),
(10, '07-12-2018', 10006, 10)

SET IDENTITY_INSERT Wishlist OFF;

SELECT * FROM Wishlist;

```

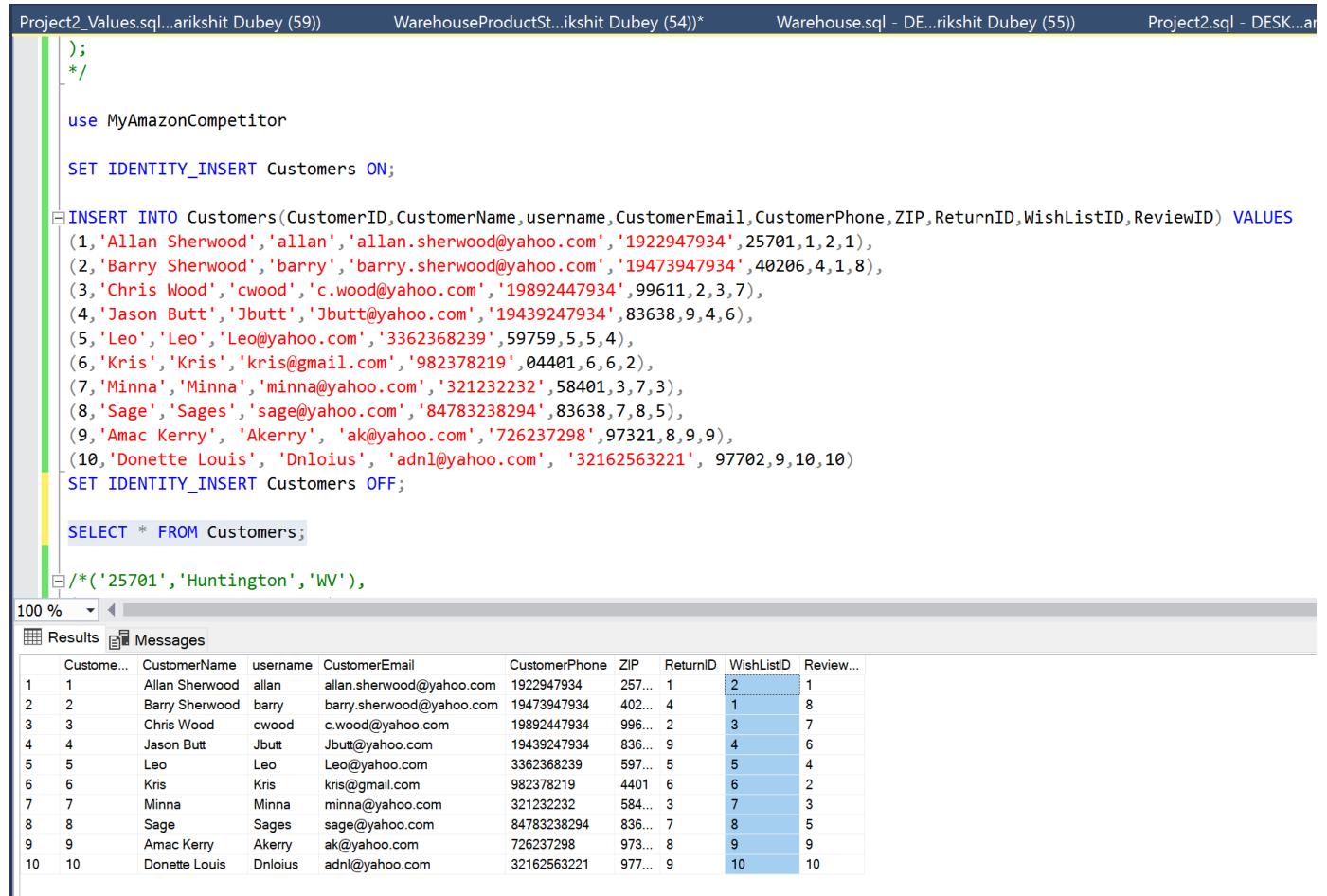
The screenshot shows the execution of an SQL script in SSMS. The script uses the `MyAmazonCompetitor` database and the `Wishlist` table. It inserts 10 rows of data into the table, each containing a WishListID, a Date, a ProductID, and a Quantity. After the insertions, it turns off `IDENTITY_INSERT`. Finally, it selects all columns from the `Wishlist` table. The results are displayed in a grid below the script.

	WishListID	Date	ProductID	Quantity
1	1	2019-02-10 00:00:00.0...	10010	3
2	2	2019-04-03 00:00:00.0...	10009	1
3	3	2019-12-04 00:00:00.0...	10003	5
4	4	2019-07-03 00:00:00.0...	10007	9
5	5	2018-05-10 00:00:00.0...	10001	5
6	6	2017-08-07 00:00:00.0...	10004	8
7	7	2019-12-10 00:00:00.0...	10008	3
8	8	2019-07-07 00:00:00.0...	10005	7
9	9	2019-12-01 00:00:00.0...	10002	12
10	10	2018-07-12 00:00:00.0...	10006	10

Project2

Parikshit Dubey
265002524

11. Populating the Customers table



```

Project2_Values.sql...arkshit Dubey (59)          WarehouseProductSt...ikshit Dubey (54)*          Warehouse.sql - DE...rikshit Dubey (55))          Project2.sql - DESK...ar
);
/*
use MyAmazonCompetitor
SET IDENTITY_INSERT Customers ON;

INSERT INTO Customers(CustomerID,CustomerName,username,CustomerEmail,CustomerPhone,ZIP,ReturnID,WishListID,ReviewID) VALUES
(1,'Allan Sherwood','allan','allan.sherwood@yahoo.com','1922947934',25701,1,2,1),
(2,'Barry Sherwood','barry','barry.sherwood@yahoo.com','19473947934',40206,4,1,8),
(3,'Chris Wood','cwood','c.wood@yahoo.com','19892447934',99611,2,3,7),
(4,'Jason Butt','Jbutt','Jbutt@yahoo.com','19439247934',83638,9,4,6),
(5,'Leo','Leo','Leo@yahoo.com','3362368239',59759,5,5,4),
(6,'Kris','Kris','kris@gmail.com','982378219',04401,6,6,2),
(7,'Minna','Minna','minna@yahoo.com','321232232',58401,3,7,3),
(8,'Sage','Sages','sage@yahoo.com','84783238294',83638,7,8,5),
(9,'Amac Kerry','Akerry','ak@yahoo.com','726237298',97321,8,9,9),
(10,'Donette Louis','Dnloius','adnl@yahoo.com','32162563221',97702,9,10,10)
SET IDENTITY_INSERT Customers OFF;

SELECT * FROM Customers;

/*('25701','Huntington','WV'),

```

The screenshot shows the execution of a SQL script named 'Project2_Values.sql'. The script uses the 'MyAmazonCompetitor' database and sets the identity insert option for the 'Customers' table. It then inserts 10 rows of customer data into the table. Finally, it selects all columns from the 'Customers' table. The results pane displays the 10 inserted rows, each with a unique CustomerID from 1 to 10, corresponding names, and various contact details.

	CustomerID	CustomerName	username	CustomerEmail	CustomerPhone	ZIP	ReturnID	WishListID	ReviewID
1	1	Allan Sherwood	allan	allan.sherwood@yahoo.com	1922947934	257...	1	2	1
2	2	Barry Sherwood	barry	barry.sherwood@yahoo.com	19473947934	402...	4	1	8
3	3	Chris Wood	cwood	c.wood@yahoo.com	19892447934	996...	2	3	7
4	4	Jason Butt	Jbutt	Jbutt@yahoo.com	19439247934	836...	9	4	6
5	5	Leo	Leo	Leo@yahoo.com	3362368239	597...	5	5	4
6	6	Kris	Kris	kris@gmail.com	982378219	04401	6	6	2
7	7	Minna	Minna	minna@yahoo.com	321232232	584...	3	7	3
8	8	Sage	Sages	sage@yahoo.com	84783238294	836...	7	8	5
9	9	Amac Kerry	Akerry	ak@yahoo.com	726237298	973...	8	9	9
10	10	Donette Louis	Dnloius	adnl@yahoo.com	32162563221	977...	9	10	10

Project2

Parikshit Dubey
265002524

12. Populating the OrderItems table

```

Project2_Values.sql...arikshit Dubey (59))          WarehouseProductSt...ikshit Dubey (54))*          Warehouse.sql - DE...
UnitPrice           MONEY      NOT NULL,
CumulativeCost     MONEY      NOT NULL
);*/
use MyAmazonCompetitor
SET IDENTITY_INSERT OrderItems ON;
INSERT INTO OrderItems(OrderItemsID,ProductID,Quantity,UnitPrice,CumulativeCost) VALUES
(1,10010,3,4,12),
(2,10002,4,12,48),
(3,10001,8,2,16),
(10,10004,10,3,30),
(4,10003,32,2,64),
(5,10006,21,3,63),
(6,10005,54,2,108),
(7,10008,15,3,45),
(8,10007,1200,2,2400),
(9,10009,2100,1,2100)
SET IDENTITY_INSERT OrderItems OFF;
SELECT * from OrderItems;
/*('25701','Huntington','WV'),
('40206','Louisville','KY'),
('100011','Knoxville','TN')*/

```

100 %

	OrderItemsID	ProductID	Quant...	UnitPrice	CumulativeCost
1	1	10010	3	4.00	12.00
2	2	10002	4	12.00	48.00
3	3	10001	8	2.00	16.00
4	4	10003	32	2.00	64.00
5	5	10006	21	3.00	63.00
6	6	10005	54	2.00	108.00
7	7	10008	15	3.00	45.00
8	8	10007	1200	2.00	2400.00
9	9	10009	2100	1.00	2100.00
10	10	10004	10	3.00	30.00

Project2

Parikshit Dubey
265002524

13. Populating the ReturnItems table

```

Project2_Values.sql...arikshit Dubey (59)          WarehouseProductSt...ikshit Dubey (54))*
ReturnItemsID           INT          PRIMARY KEY
ProductID                INT          REFERENCES Product
Quantity                 INT          DEFAULT NULL
);*/
use MyAmazonCompetitor

SET IDENTITY_INSERT ReturnItems ON;
INSERT INTO ReturnItems(ReturnItemsID,ProductID,Quantity) VALUES
(1,10001,3),
(2,10002,4),
(3,10003,2),
(4,10004,5),
(5,10005,10),
(6,10006,21),
(7,10007,14),
(8,10008,56),
(9,10009,25),
(10,10010,54)
SET IDENTITY_INSERT ReturnItems OFF;

SELECT * FROM ReturnItems;

/*('25701','Huntington','WV'),
('40206','Louisville','KY'),
('99611','Kenai','AK'),
('97702','Bend','OR'),
('97321','Albany','OR'),
('83638','McCall','ID'),
```

100 %

	ReturnItemsID	ProductID	Quantity
1	1	10001	3
2	2	10002	4
3	3	10003	2
4	4	10004	5
5	5	10005	10
6	6	10006	21
7	7	10007	14
8	8	10008	56
9	9	10009	25
10	10	10010	54

Project2

Parikshit Dubey
265002524

14. Populating the ReturnDetails table

```

Project2 Values.sql..arikshit Dubey (59)          WarehouseProductSt...ikshit Dubey (54))*      Warehouse.sql - DE..rikshit Dubey (55))      Project
StatusOfReturn        VARCHAR(255)           DEFAULT NULL,
ShippingInfo          VARCHAR(255)           DEFAULT NULL,
ShippingCost          Money                 DEFAULT NULL
),*/
use MyAmazonCompetitor

SET IDENTITY_INSERT ReturnDetails ON;
INSERT INTO ReturnDetails(ReturnID,ReturnItemsID,DateOfReturn,StatusOfReturn,ShippingInfo,ShippingCost) VALUES
(1,1,'01-04-2019','Returned','FedEx',3),
(2,2,'01-05-2019','Returned','FedEx',4),
(3,3,'12-04-2018','Not Returned','','0'),
(4,4,'04-04-2019','Returned','FedEx',2),
(5,5,'05-12-2019','Returned','FedEx',12),
(6,6,'11-01-2019','Not Returned','','0'),
(7,7,'11-05-2019','Not Returned','','0'),
(8,8,'07-14-2018','Returned','USPS',4),
(9,9,'04-06-2019','Returned','FedEx',1),
(10,10,'10-12-2019','Returned','UPS',0)
SET IDENTITY_INSERT ReturnDetails OFF;

SELECT * FROM ReturnDetails;

/*
(*('25701','Huntington','WV'),
('40206','Louisville','KY'),
('99611','Kenai','AK'),
('97702','Bend','OR'),
)

```

100 %

	ReturnID	ReturnItemsID	DateOfReturn	StatusOfRet...	ShippingI...	ShippingCost
1	1	1	2019-01-04 00:00:00.0...	Returned	FedEx	3.00
2	2	2	2019-01-05 00:00:00.0...	Returned	FedEx	4.00
3	3	3	2018-12-04 00:00:00.0...	Not Returned		0.00
4	4	4	2019-04-04 00:00:00.0...	Returned	FedEx	2.00
5	5	5	2019-05-12 00:00:00.0...	Returned	FedEx	12.00
6	6	6	2019-11-01 00:00:00.0...	Not Returned		0.00
7	7	7	2019-11-05 00:00:00.0...	Not Returned		0.00
8	8	8	2018-07-14 00:00:00.0...	Returned	USPS	4.00
9	9	9	2019-04-06 00:00:00.0...	Returned	FedEx	1.00
10	10	10	2019-10-12 00:00:00.0...	Returned	UPS	0.00

Project2

Parikshit Dubey
265002524

15. Populating the OrdersTable table

```

Project2_Values.sql..arkshit Dubey (59)      WarehouseProductSt...ikshit Dubey (54)*      Warehouse.sql - DE...rikshit Dubey (55))      Project2.sql - DESK...arkshit Dubey (5
TotalPrice          MONEY      NOT NULL,          DEFAULT NULL,
ShippingService     VARCHAR(255)          NOT NULL,
StreetAddress       VARCHAR(255)          NOT NULL,
ZIP                INT                   REFERENCES ZIPDetails(ZIP),
ShippingFare        MONEY      NOT NULL,
ExpectedShippingDate DATETIME          NOT NULL,
ActualShippingDate DATETIME          DEFAULT NULL,
ShippingInfo        VARCHAR(255)          DEFAULT NULL
),*/
use MyAmazonCompetitor

SET IDENTITY_INSERT OrdersTable ON;
INSERT INTO OrdersTable(OrderID,CustomerID,Status,ReturnID,OrderDate,OrderItemsID,TotalPrice,ShippingService,StreetAddress,ZIP,ShippingF
(1,1,'Processing',1,'06-07-2019',1,14,'UPS','111 ABC Street',25701,3,'06-08-2019','',''),
(2,2,'Processing',2,'06-09-2019',2,16,'USPS','222 ABC Street',25701,4,'06-09-2019','',''),
(3,3,'Processing',3,'07-07-2019',3,14,'FedEx','333 PYS Street',40206,4,'07-08-2019','',''),
(4,4,'Processing',4,'07-09-2019',4,213,'USPS','276 RTT Street',97702,2,'07-09-2019','',''),
(5,5,'Processing',5,'06-07-2019',6,148,'UPS','111 RON Street',25701,3,'06-08-2019','',''),
(6,6,'Processing',6,'06-09-2019',5,1676,'USPS','222 West Street',04401,4,'06-09-2019','',''),
(7,7,'Processing',7,'07-07-2019',8,1400,'FedEx','333 North Street',40206,4,'07-08-2019','',''),
(8,8,'Processing',8,'07-09-2019',7,2130,'USPS','276 Euclid Street',58401,2,'07-09-2019','',''),
(9,9,'Processing',9,'07-09-2019',10,243,'USPS','276 RTT Street',97702,2,'07-09-2019','07-09-2019','Reached'),
(10,10,'Processing',10,'06-07-2019',9,187,'UPS','111 RON Street',25701,5,'06-08-2019','06-09-2019','Reached')
SET IDENTITY_INSERT OrdersTable OFF;

SELECT * FROM OrdersTable;

```

100 % ▾

	OrderID	CustomerID	Status	ReturnID	OrderDate	OrderItemsID	TotalPrice	ShippingService	StreetAddress	ZIP	ShippingFare	ExpectedShippingDate	ActualShippingDate	ShippingInfo
1	1	1	Proce...	1	2019-06-07 00:00:00.0...	1	14.00	UPS	111 ABC Street	257...	3.00	2019-06-08 00:00:00.0...	1900-01-01 00:00:00.0...	
2	2	2	Proce...	2	2019-06-09 00:00:00.0...	2	16.00	USPS	222 ABC Street	257...	4.00	2019-06-09 00:00:00.0...	1900-01-01 00:00:00.0...	
3	3	3	Proce...	3	2019-07-07 00:00:00.0...	3	14.00	FedEx	333 PYS Street	402...	4.00	2019-07-08 00:00:00.0...	1900-01-01 00:00:00.0...	
4	4	4	Proce...	4	2019-07-09 00:00:00.0...	4	213.00	USPS	276 RTT Street	977...	2.00	2019-07-09 00:00:00.0...	1900-01-01 00:00:00.0...	
5	5	5	Proce...	5	2019-06-07 00:00:00.0...	6	148.00	UPS	111 RON Street	257...	3.00	2019-06-08 00:00:00.0...	1900-01-01 00:00:00.0...	
6	6	6	Proce...	6	2019-06-09 00:00:00.0...	5	1676.00	USPS	222 West Street	4401	4.00	2019-06-09 00:00:00.0...	1900-01-01 00:00:00.0...	
7	7	7	Proce...	7	2019-07-07 00:00:00.0...	8	1400.00	FedEx	333 North Street	402...	4.00	2019-07-08 00:00:00.0...	1900-01-01 00:00:00.0...	
8	8	8	Proce...	8	2019-07-09 00:00:00.0...	7	2130.00	USPS	276 Euclid Str...	584...	2.00	2019-07-09 00:00:00.0...	1900-01-01 00:00:00.0...	
9	9	9	Proce...	9	2019-07-09 00:00:00.0...	10	243.00	USPS	276 RTT Street	977...	2.00	2019-07-09 00:00:00.0...	2019-07-09 00:00:00.0...	Reached
10	10	10	Proce...	10	2019-06-07 00:00:00.0...	9	187.00	UPS	111 RON Street	257...	5.00	2019-06-08 00:00:00.0...	2019-06-09 00:00:00.0...	Reached

Project2

Parikshit Dubey
265002524

Views, Stored Procedures, Transactions, Triggers, Constraints, Scripts

1. A view is created to show ProductID, ProductName, ProductDescription and Price from Products table.

The screenshot shows a SQL query window in SSMS. The code is:

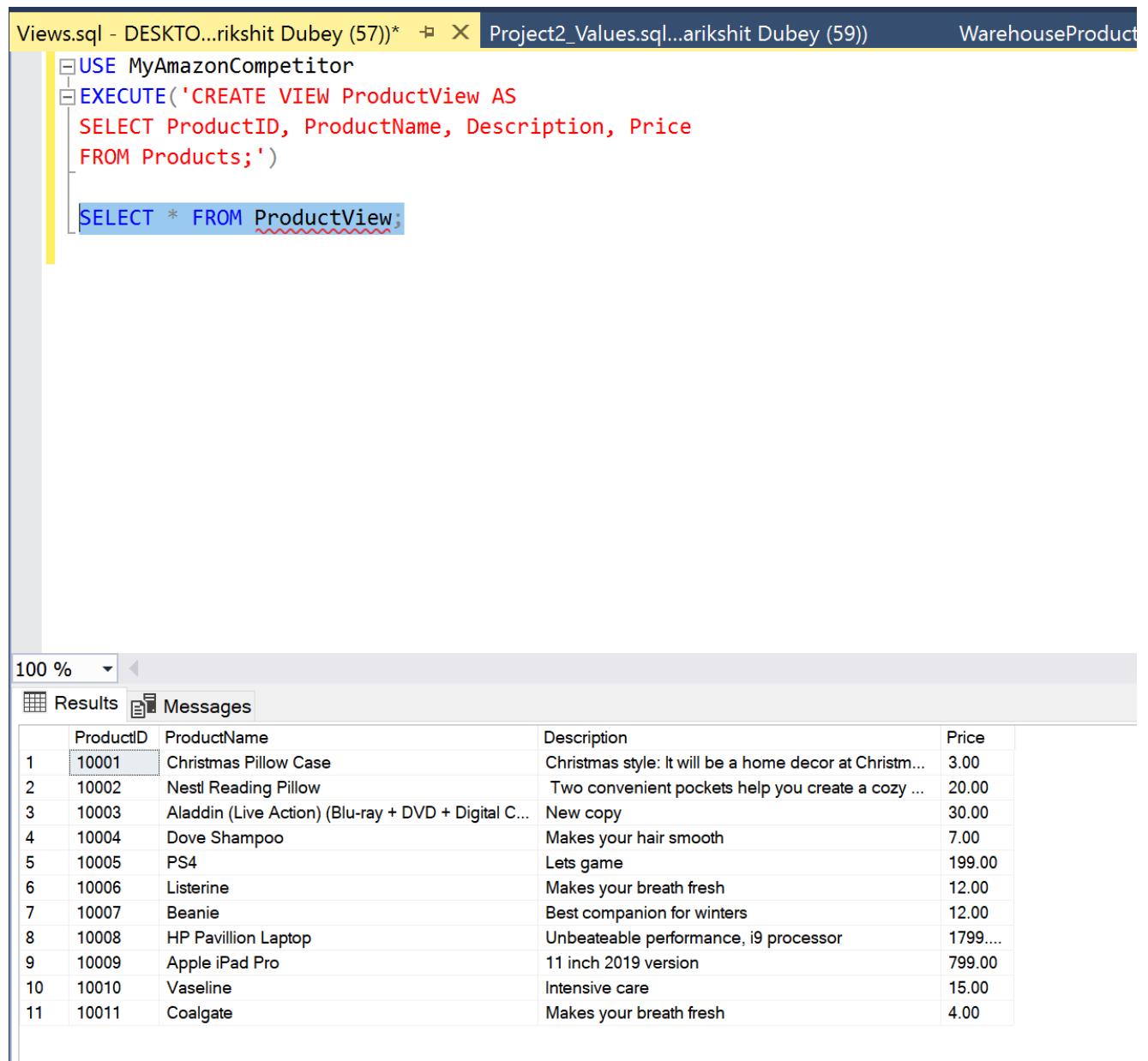
```
USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ProductView AS
    SELECT ProductID, ProductName, Description, Price
    FROM Products;')
```

The execution message at the bottom says:

Commands completed successfully.
Completion time: 2019-12-07T23:46:59.3621402-05:00

Project2

Parikshit Dubey
265002524



The screenshot shows a SQL Server Management Studio (SSMS) window. The title bar indicates the connection is to 'DESKTO...rikshit Dubey (57)*' and the current file is 'Views.sql'. The query pane displays the following T-SQL code:

```

USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ProductView AS
SELECT ProductID, ProductName, Description, Price
FROM Products;')

SELECT * FROM ProductView;

```

The 'Results' tab is selected, showing the output of the last query. The data is presented in a table:

	ProductID	ProductName	Description	Price
1	10001	Christmas Pillow Case	Christmas style: It will be a home decor at Christm...	3.00
2	10002	Nestl Reading Pillow	Two convenient pockets help you create a cozy ...	20.00
3	10003	Aladdin (Live Action) (Blu-ray + DVD + Digital C...	New copy	30.00
4	10004	Dove Shampoo	Makes your hair smooth	7.00
5	10005	PS4	Lets game	199.00
6	10006	Listerine	Makes your breath fresh	12.00
7	10007	Beanie	Best companion for winters	12.00
8	10008	HP Pavillion Laptop	Unbeatable performance, i9 processor	1799...
9	10009	Apple iPad Pro	11 inch 2019 version	799.00
10	10010	Vaseline	Intensive care	15.00
11	10011	Coalgate	Makes your breath fresh	4.00

Project2

Parikshit Dubey
265002524

2. A view is created to show ProductID, Score and CustomerName from Review table.

The screenshot shows a SQL query editor window with two distinct sections of T-SQL code. The top section creates a view named 'ProductView' that selects ProductID, ProductName, Description, and Price from the 'Products' table. The bottom section creates a view named 'ReviewView' that selects ProductID, Score, and CustomerName from the 'Review' table. Both sections include a final 'SELECT * FROM' statement to demonstrate the view's functionality. The code is syntax-highlighted, and the 'Messages' pane at the bottom indicates that the commands completed successfully.

```
Views.sql - DESKTOP-RIKSHIT DUBEY (57)* X Project2_Values.sql...ariKshit DUBEY (59)

USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ProductView AS
    SELECT ProductID, ProductName, Description, Price
    FROM Products;')

SELECT * FROM ProductView;

USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ReviewView AS
    SELECT ProductID, Score, CustomerName
    FROM Review;')

SELECT * FROM ReviewView;

100 % <
Messages
Commands completed successfully.

Completion time: 2019-12-07T23:52:21.3543507-05:00
```

Project2

Parikshit Dubey
265002524

The screenshot shows a SQL Server Management Studio window with two queries in the script pane and a results grid in the results pane.

```
USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ProductView AS
SELECT ProductID, ProductName, Description, Price
FROM Products;')

SELECT * FROM ProductView;

USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ReviewView AS
SELECT ProductID, Score, CustomerName
FROM Review;')

SELECT * FROM ReviewView;
```

The results pane displays the data from the `ReviewView` query:

	ProductID	Score	CustomerNa...
1	10010	10	Aaron
2	10009	5	Lisa
3	10003	1	Ruben
4	10007	9	XYZ
5	10001	5	Andy
6	10004	8	Lima
7	10008	3	Ruby
8	10005	7	Karen
9	10002	3	Ruby
10	10006	10	Kathy

Project2

Parikshit Dubey
265002524

3. A view is created using Join to view ratings on the products

The screenshot shows a SQL query editor window with two tabs: 'Views.sql' and 'Project2_Values.sql'. The 'Views.sql' tab is active, displaying the following SQL code:

```
USE MyAmazonCompetitor
EXECUTE('CREATE VIEW RatingView AS
SELECT ProductName, Score, CustomerName
FROM Review JOIN Products
ON Review.ProductID = Products.ProductID;')

SELECT * FROM RatingView;
```

Below the code, the message pane shows:

100 % ▶

Messages Client Statistics

Commands completed successfully.

Completion time: 2019-12-07T23:57:51.7955742-05:00

Project2

Parikshit Dubey
265002524

Views.sql - DESKTOP-RIKSHIT DUBEY (57)* X Project2_Values.sql...arikshit Dubey

```
USE MyAmazonCompetitor
EXECUTE('CREATE VIEW RatingView AS
SELECT ProductName, Score, CustomerName
FROM Review JOIN Products
ON Review.ProductID = Products.ProductID;')

SELECT * FROM RatingView;
```

100 %

Results Messages Client Statistics

	ProductName	Sco...	CustomerNa...
1	Vaseline	10	Aaron
2	Apple iPad Pro	5	Lisa
3	Aladdin (Live Action) (Blu-ray + DVD + Digital C...	1	Ruben
4	Beanie	9	XYZ
5	Christmas Pillow Case	5	Andy
6	Dove Shampoo	8	Lima
7	HP Pavilion Laptop	3	Ruby
8	PS4	7	Karen
9	Nestl Reading Pillow	3	Ruby
10	Listerine	10	Kathy

Project2

Parikshit Dubey
265002524

4. A view is created to view the items whose price is greater than \$500.

The screenshot shows a SQL Server Management Studio (SSMS) window. The title bar indicates two tabs: 'Views.sql - DESKTOP-RIKSHIT DUB...' (active) and 'Project2_Values.sql - DESKTOP-RIKSHIT DUB...'. The main pane displays the following T-SQL code:

```
EXECUTE('CREATE VIEW ExpensiveItems AS
SELECT ProductName, Description, Price
FROM Products
WHERE Price > 500;')
```

The code is highlighted in red. Below the code, the message pane shows:

100 %

Messages

Commands completed successfully.

Completion time: 2019-12-08T00:06:02.8417615-05:00

Project2

Parikshit Dubey
265002524

The screenshot shows a SQL Server Management Studio (SSMS) window. The title bar indicates two tabs: 'Views.sql - DESKTOP...rikshit Dubey (57)*' and 'Project2_Values.sql...arikshit Dubey (59))'. The main pane displays the following SQL code:

```
EXECUTE('CREATE VIEW ExpensiveItems AS
SELECT ProductName, Description, Price
FROM Products
WHERE Price > 500;')

SELECT * FROM ExpensiveItems;
```

Below the code, the results pane is visible, showing a table with two rows of data. The table has columns: ProductName, Description, and Price.

	ProductName	Description	Price
1	HP Pavilion Laptop	Unbeatable performance, i9 processor	1799....
2	Apple iPad Pro	11 inch 2019 version	799.00

Project2

Parikshit Dubey
265002524

5. A view is created to view the Orders which are in Processing.

The screenshot shows a SQL Server Management Studio (SSMS) window. The title bar indicates the current database is 'Project2'. The main pane displays the following T-SQL code:

```
CREATE VIEW OrdersInProcessing AS
SELECT OrderID, OrderDate, Status, TotalPrice, ShippingInfo, ExpectedShippingDate
FROM OrdersTable
WHERE Status = 'Processing';
```

In the bottom right corner of the main pane, there is a small yellow status bar showing '100 %' completion. Below the main pane is a 'Messages' window with the following content:

Commands completed successfully.
Completion time: 2019-12-08T00:14:28.9427172-05:00

Project2

Parikshit Dubey
265002524

The screenshot shows a SQL Server Management Studio window with two queries and their results.

Query 1:

```
Views.sql - DESKTOP-RIKSHIT DUBHEY (57)* Project2_Values.sql...arikshit Dubey (59) WarehouseProductSt...ikshit Dubey (54)* W
SELECT OrderID, OrderDate, Status, TotalPrice, ShippingInfo, ExpectedShippingDate
FROM OrdersTable
WHERE Status = 'Processing';
```

Query 2:

```
SELECT * FROM OrdersInProcessing;
```

Results:

	OrderID	OrderDate	Status	TotalPrice	ShippingInfo	ExpectedShippingDate
1	1	2019-06-07 00:00:00.0...	Processing	14.00		2019-06-08 00:00:00.0...
2	2	2019-06-09 00:00:00.0...	Processing	16.00		2019-06-09 00:00:00.0...
3	3	2019-07-07 00:00:00.0...	Processing	14.00		2019-07-08 00:00:00.0...
4	4	2019-07-09 00:00:00.0...	Processing	213.00		2019-07-09 00:00:00.0...
5	5	2019-06-07 00:00:00.0...	Processing	148.00		2019-06-08 00:00:00.0...
6	6	2019-06-09 00:00:00.0...	Processing	1676.00		2019-06-09 00:00:00.0...
7	7	2019-07-07 00:00:00.0...	Processing	1400.00		2019-07-08 00:00:00.0...
8	8	2019-07-09 00:00:00.0...	Processing	2130.00		2019-07-09 00:00:00.0...

Project2

Parikshit Dubey
265002524

- A trigger is created on the products table, whenever there is an Insert, Update or Delete in the table. An error flag is raised.

The screenshot shows a SQL Server Management Studio window with three tabs: 'Trigger.sql - DESKT...arikshit Dubey (58)*' containing the trigger creation script, 'Views.sql - DESKTO...rikshit Dubey (57)' which is empty, and 'Project2_Values.sql...arikshit Dubey (59)' which is also empty. Below the tabs is a code editor window with the following content:

```

CREATE TRIGGER TRIGGER1
ON Products
AFTER INSERT,UPDATE,DELETE
AS RAISERROR('Product table has some updates, Please review',16,10)
GO

```

At the bottom of the screen, the 'Messages' pane displays the following output:

```

Commands completed successfully.

Completion time: 2019-12-08T00:35:51.4667364-05:00

```

Error flag is raised below:

The screenshot shows a SQL Server Management Studio window with several tabs: 'Trigger.sql - DESKT...arikshit Dubey (58)*', 'Project2_Values.sql...arikshit Dubey (59)', 'WarehouseProduct5...ikshit Dubey (54)*', 'Warehouse.sql - DE...rikshit Dubey (55)', and 'Project2_Products...arikshit Dubey (56)'. The code editor window contains the trigger definition and an insert statement:

```

ON Products
AFTER INSERT,UPDATE,DELETE
AS RAISERROR('Product table has some updates, Please review',16,10)
GO

SET IDENTITY_INSERT Products ON;

INSERT INTO Products(ProductID, ProductName, Description, Price) VALUES
('1011','Phone Case','Christmas style: It will be a home decor at Christmas, which makes the festival interesting','200')
SET IDENTITY_INSERT Products OFF;

```

The 'Messages' pane at the bottom shows the following output:

```

Msg 50000, Level 16, State 10, Procedure TRIGGER1, Line 4 [Batch Start Line 6]
Product table has some updates, Please review

(1 row affected)

Completion time: 2019-12-08T00:37:40.0951153-05:00
| 

```

Project2

Parikshit Dubey
265002524

The updated Products table from the previous insert statement can be verified below:

The screenshot shows a SQL Server Management Studio window with four tabs at the top: 'Trigger.sql - DESKT...arikshit Dubey (58)*' (selected), 'Project2_Values.sql...arikshit Dubey (59)', 'WarehouseProductSt...ikshit Dubey (54)*', and 'Warehouse.sql - DE...rikshit Dubey (55)'. The 'Trigger.sql' tab contains the following T-SQL code:

```

ON Products
AFTER INSERT,UPDATE,DELETE
AS RAISERROR('Product table has some updates, Please review',16,10)
GO

SET IDENTITY_INSERT Products ON;

INSERT INTO Products(ProductID, ProductName, Description, Price) VALUES
('1011', 'Phone Case', 'Christmas style: It will be a home decor at Christmas, which makes the festival interesting', '200')
SET IDENTITY_INSERT Products OFF;

SELECT * FROM Products;

```

The 'Results' tab displays the contents of the 'Products' table:

	ProductID	ProductName	Description	Price
1	1011	Phone Case	Christmas style: It will be a home decor at Christmas, which makes the festival interesting	200.00
2	10001	Christmas Pillow Case	Christmas style: It will be a home decor at Christmas, which makes the festival interesting	3.00
3	10002	Nestl Reading Pillow	Two convenient pockets help you create a cozy reading space	20.00
4	10003	Aladdin (Live Action) (Blu-ray + DVD + Digital C...	New copy	30.00
5	10004	Dove Shampoo	Makes your hair smooth	7.00
6	10005	PS4	Lets game	199.00
7	10006	Listerine	Makes your breath fresh	12.00
8	10007	Beanie	Best companion for winters	12.00
9	10008	HP Pavilion Laptop	Unbeatable performance, i9 processor	1799.00
10	10009	Apple iPad Pro	11 inch 2019 version	799.00
11	10010	Vaseline	Intensive care	15.00
12	10011	Coalgate	Makes your breath fresh	4.00

Project2

Parikshit Dubey
265002524

7. A new role NewRole is created in which Update permission is granted to Products, Update and Insert permission is granted to ZIPDetails table and Update, Insert and Delete permission is granted to Warehouse table.

The screenshot shows the SQL Server Management Studio interface. In the top tab bar, the active window is 'part.sql - DESKTOP...rikshit Dubey (60)*'. Below the tabs, there is a code editor pane containing the following T-SQL script:

```
USE MyAmazonCompetitor;
Create Role NewRole
Grant Update on Products to NewRole
Grant Update, Insert on ZIPDetails to NewRole
Grant Update, Insert, Delete on Warehouse to NewRole
Exec sp_addrolemember db_datareader , NewRole
```

At the bottom of the screen, the 'Messages' pane displays the following output:

100 % ▶

Messages

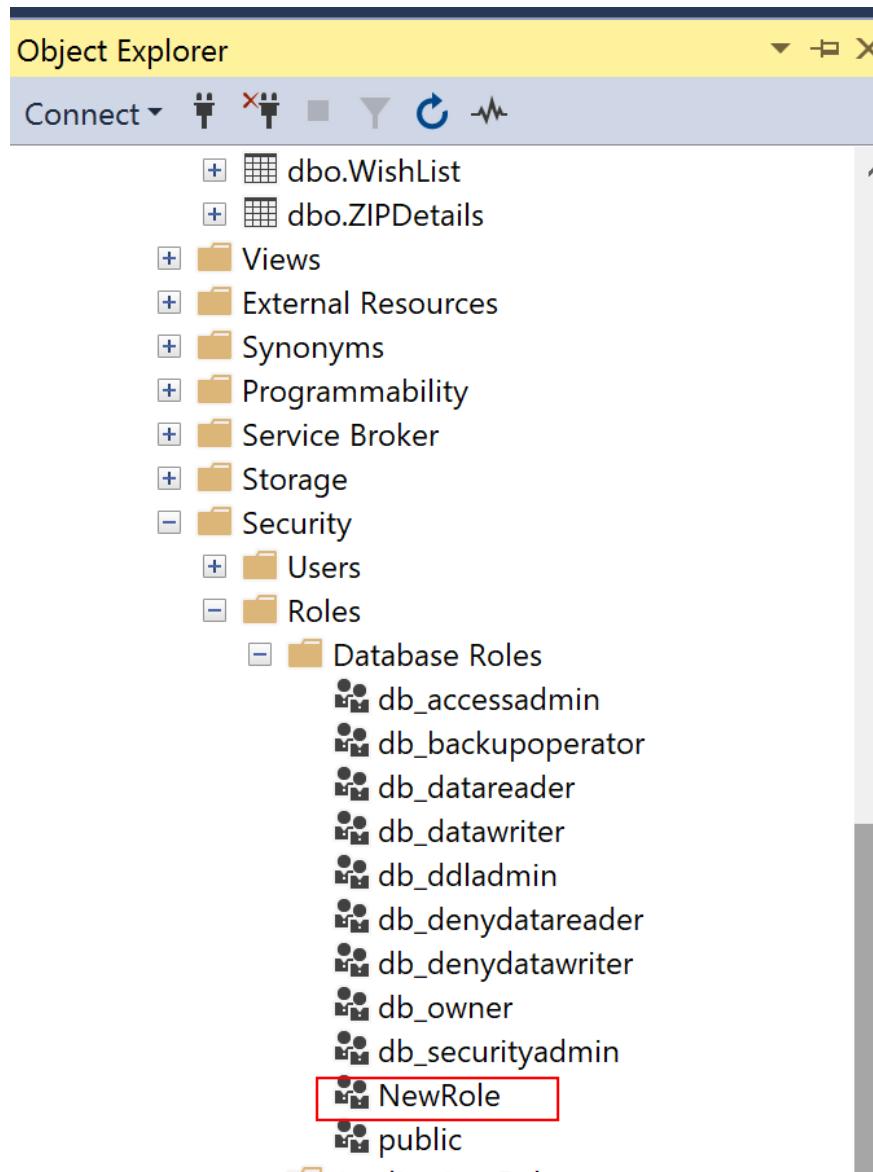
Commands completed successfully.

Completion time: 2019-12-08T00:43:40.0186971-05:00

Project2

Parikshit Dubey
265002524

New role creation can be verified below:



Project2

Parikshit Dubey
265002524

8. A new Login is created on the database with User Parikshit Dubey and NewRole has been assigned to the user.

The screenshot shows a SQL Server Management Studio (SSMS) interface. The title bar indicates three open windows: 'epart.sql - DESKTOP...', 'Trigger.sql - DESKTOP...', and 'Project2_Values.sql'. The main pane displays T-SQL code for creating a login and a user:

```
CREATE LOGIN AmazonCompetitorDatabase01 with password = 'zzz123',
Default_Database = MyAmazonCompetitor
Create USER ParikshitDubey for login AmazonCompetitorDatabase01
Exec sp_addrolemember NewRole, ParikshitDubey;
```

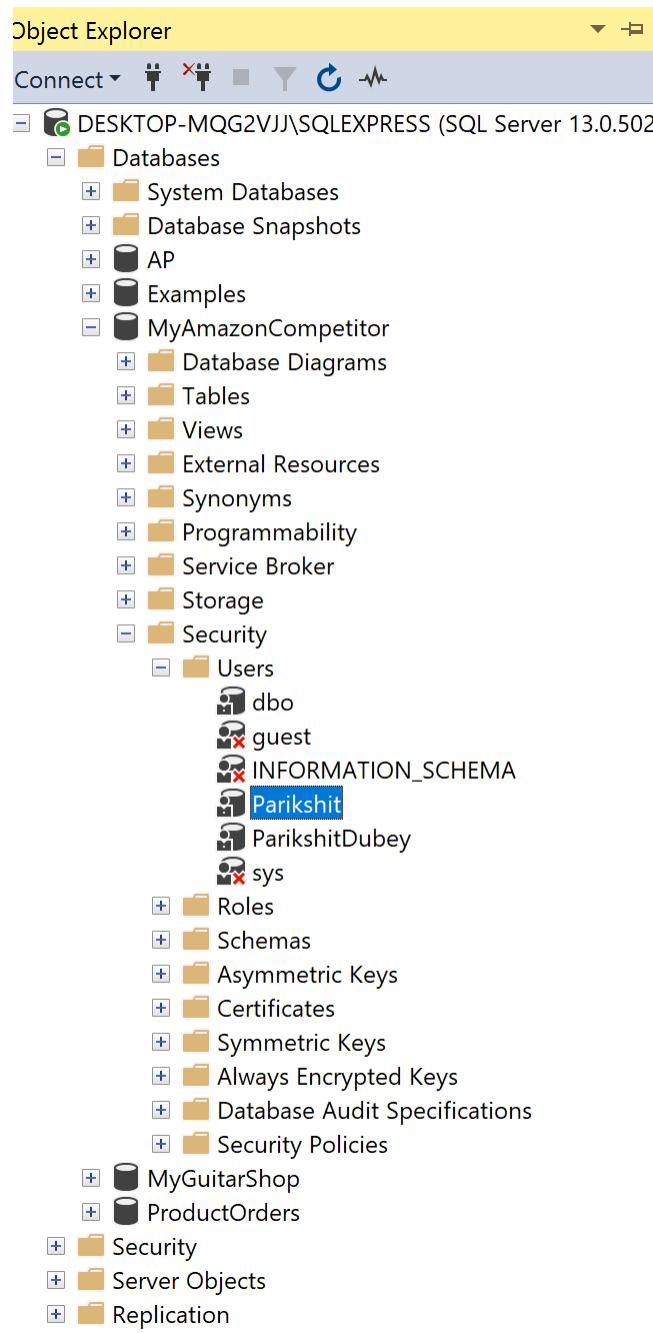
The code is highlighted with syntax coloring. The 'Messages' tab in the bottom right shows the execution results:

Commands completed successfully.
Completion time: 2019-12-08T00:52:04.4669804-05:00

Project2

Parikshit Dubey
265002524

Creation of users Parikshit and Parikshit Dubey can be verified below:



Project2

Parikshit Dubey
265002524

9. A new schema ParikshitDubs is created and Products Delete,Select,Update,Insert and Execute permissions on schema has been granted from ParikshitDubs to Parikshit Dubey

The screenshot shows a SQL Server Management Studio (SSMS) window with three tabs at the top: 'eprt.sql - DESKTOP...rikshit Dubey (60)*' (highlighted in yellow), 'Trigger.sql - DESKT...arikshit Dubey (58))', and 'Project2_Values.sql...arikshit Dubey (59))'. The main pane displays T-SQL code:

```
GO
Create Schema ParikshitDubs
Go
Alter Schema ParikshitDubs Transfer dbo.Products
Alter User ParikshitDubey with Default_Schema= ParikshitDubs
Grant Delete,Select,Update, Insert, Execute on Schema :: ParikshitDubs to ParikshitDubey
```

The 'Messages' pane at the bottom shows the execution results:

```
Commands completed successfully.
Completion time: 2019-12-08T00:55:26.7895849-05:00
```

Project2

Parikshit Dubey
265002524

10.A stored procedure named procdateSelect is created that accepts two parameters, @DateMin and @DateMax. If the procedure is called with no parameters or with the null values, an error is raised to describe the problem. If the procedure is called with null values, the parameters are validated. If all the parameters are valid the result set has ReviewID, DateOfReview, and Score for each of the review is within the date range, it is sorted with the earliest DateOfReview first.

The screenshot shows the SQL Server Management Studio interface with three tabs at the top: 'eapart.sql - DESKTOP...rikshit Dubey (60)*' (highlighted in yellow), 'Trigger.sql - DESKT...ariikshit Dubey (58)', and 'Project2_Values.sql...ariikshit Du'. The main pane displays the T-SQL code for creating the stored procedure:

```
USE MyAmazonCompetitor;
GO
CREATE PROC procdateSelect
    @DateMin varchar(30) = NULL,
    @DateMax varchar(30) = NULL
AS
IF @DateMin IS NULL OR @DateMax IS NULL
    THROW 50001, 'Please enter the minimum and maximum date', 1;
IF NOT(ISDATE(@DateMin) = 1 AND ISDATE(@DateMax) = 1)
    THROW 50001, 'Please enter a valid date format', 1;
IF CAST(@DateMin AS datetime) > CAST(@DateMax AS datetime)
    THROW 50001, 'DateMin cannot be greater than DateMax', 1;
SELECT ReviewID, DateOfReview, Score
FROM Review
WHERE DateOfReview BETWEEN @DateMin AND @DateMax
ORDER BY DateOfReview;
```

The code includes validation logic to check for null parameters, valid date formats, and ensure DateMin is not greater than DateMax. The final output is a SELECT statement filtering reviews by the specified date range and ordering them by DateOfReview.

At the bottom, the 'Messages' pane shows:

- Commands completed successfully.
- Completion time: 2019-12-08T01:20:04.7553727-05:00

Project2

Parikshit Dubey
265002524

11. The query is used to call the procedure created in step 10. The query designed in this question returns the reviews between December10 and December12, 2019. Also for the second part of this query a call is designed which can cause any raised errors by the procedure and the error number with description is printed.

The screenshot shows a SQL Server Management Studio window with two tabs: 'epart.sql' and 'Trigger.sql'. The 'epart.sql' tab contains the following code:

```
BEGIN TRY
EXEC procdateSelect '12/10/2019';
END TRY
BEGIN CATCH
PRINT 'Error Number - ' + CONVERT(varchar(30), ERROR_NUMBER());
PRINT 'Error Message - ' + CONVERT(varchar(50), ERROR_MESSAGE());
END CATCH
```

The 'Messages' pane at the bottom displays the following output:

```
Error Number - 50001
Error Message - Please enter the minimum and maximum date
Completion time: 2019-12-08T01:21:10.7113992-05:00
```

Project2

Parikshit Dubey
265002524

Conclusion

- ❖ This project was carried out using Microsoft SQL Management Studio and MS Visio.
- ❖ This project was performed on a windows 10 machine running on a virtual machine
- ❖ All the questions in the project were answered in the report with proper screenshots
- ❖ Database creation and implementation is carried out in this project
- ❖ Join and unions are used to obtain results
- ❖ Queries and Subqueries are also used to obtain results
- ❖ Data Manipulation is carried out in this project
- ❖ All the database diagrams for questions are properly drawn
- ❖ The relationships between the tables are properly shown the screenshots with connections
- ❖ All the codes used to create and implement the database is given in the appendix below

Project2

Parikshit Dubey
265002524

Appendix A(Database Creation)

```

use master
GO

/**Object: Database MyAmazonCompetitor****/
IF DB_ID('MyAmazonCompetitor') IS NOT NULL
    DROP DATABASE MyAmazonCompetitor
GO

Create DATABASE MyAmazonCompetitor
Go

USE MyAmazonCompetitor
Go

/**Object: Zip Details Table ****/
CREATE TABLE ZIPDetails(
    ZIP           INT
                PRIMARY KEY
    City          VARCHAR(60)      NOT NULL,
    State         VARCHAR(2)       NOT NULL
);

/**Object: Products table*****
CREATE TABLE Products(
    ProductID      INT
                PRIMARY KEY
    ProductName    VARCHAR(255)   NOT NULL,
    Description     VARCHAR(255)   NOT NULL,
    Price          MONEY         NOT NULL,
);

/**Object: Suppliers table*****
CREATE TABLE Suppliers(
    SupplierID      INT
                PRIMARY KEY
    IDENTITY        NOT NULL,
    SupplierName    VARCHAR(60)      NOT NULL,
    StreetName      VARCHAR(255)     NOT NULL,
    ZIP             INT
                REFERENCES ZIPDetails(ZIP),
    SupplierPhone   VARCHAR(20)       NOT NULL,
    SupplierEmailID VARCHAR(255)     NOT NULL,
    NumberOfProducts INT            NOT NULL,
);

/**Object: Products Stored table ****/
CREATE TABLE ProductsStored(
    ProductsStoredID INT
                PRIMARY KEY
    IDENTITY        NOT NULL,
    ProductID       INT
                REFERENCES Products(ProductID),
    NumberInStock   INT            NOT NULL,
    NumberOnWay     INT            DEFAULT NULL,
    NumberOnReturn  INT            DEFAULT NULL,
);

```

Project2

Parikshit Dubey
265002524

```

);

*****Object: Warehouse table *****/
CREATE TABLE Warehouse(
WarehouseID           INT          PRIMARY KEY
IDENTITY   NOT NULL,
StreetAddress        VARCHAR(255) NOT NULL,
ZIP                  INT          REFERENCES ZIPdetails(ZIP),
);

*****Object: Warehouse to products stored table *****/
CREATE TABLE WarehouseToProductsStored(
WarehouseID           INT          REFERENCES Warehouse(WarehouseID),
ProductsStoredID      INT          REFERENCES ProductsStored(ProductsStoredID)
);

*****Object: Products to Suppliers table *****/
CREATE TABLE ProductsToSupplierLink(
ProductID             INT          REFERENCES Products(ProductID),
SupplierID            INT          REFERENCES Suppliers(SupplierID),
);

*****Object: Warehouse to Supplier link *****/
CREATE TABLE WarehouseSupplierLink(
WarehouseID           INT          REFERENCES Warehouse(WarehouseID),
SupplierID            INT          REFERENCES Suppliers(SupplierID),
);

*****Object: Review table *****/
CREATE TABLE Review(
ReviewID               INT          PRIMARY KEY
IDENTITY   NOT NULL,
ProductID              INT          REFERENCES Products(ProductID),
DateOfReview           DATETIME    NOT NULL,
Comments               VARCHAR(255) DEFAULT NULL,
Score                  INT          DEFAULT NULL,
CustomerName           VARCHAR(255) DEFAULT NULL,
);

*****Object: Wishlist Table *****/
CREATE TABLE WishList(
WishListID             INT          PRIMARY KEY
IDENTITY   NOT NULL,
Date                  DATETIME    DEFAULT NULL,
);

```

Project2

Parikshit Dubey
265002524

```

ProductID          INT             REFERENCES
PRODUCTS(ProductID),
Quantity           INT             NOT NULL
);

*****Object: Customers Table *****
CREATE TABLE Customers(
CustomerID         INT             PRIMARY KEY
                               IDENTITY NOT NULL,
CustomerName       VARCHAR(60)    NOT NULL,
username           VARCHAR(60)    NOT NULL      UNIQUE,
CustomerEmail     VARCHAR(255)   NOT NULL      UNIQUE,
CustomerPhone     VARCHAR(20)    NOT NULL,
ZIP                INT             REFERENCES ZIPDetails(ZIP),
ReturnID           INT             NOT NULL,
WishListID         INT             /*ADD REFERENCE*/
REFERENCES WishList(WishListID),
ReviewID           INT             REFERENCES Review(ReviewID)
);

*****Object: OrderItems table*****
CREATE TABLE OrderItems(
OrderItemsID        INT             PRIMARY KEY
                               IDENTITY NOT NULL,
ProductID          INT             REFERENCES
Products(ProductID),
Quantity            INT             NOT NULL,
UnitPrice           MONEY          NOT NULL,
CumulativeCost     MONEY          NOT NULL
);

*****Object: Return Items table*****
CREATE TABLE ReturnItems(
ReturnItemsID       INT             PRIMARY KEY
                               IDENTITY NOT NULL,
ProductID          INT             REFERENCES
Products(ProductID),
Quantity            INT             DEFAULT NULL
);

*****Object: Return details*****
CREATE TABLE ReturnDetails(
ReturnID            INT             PRIMARY KEY
                               IDENTITY NOT NULL,
ReturnItemsID       INT             REFERENCES
ReturnItems(ReturnItemsID),
DateOfReturn        DATETIME       DEFAULT NULL,
StatusOfReturn      VARCHAR(255)  DEFAULT NULL,
ShippingInfo        VARCHAR(255)  DEFAULT NULL,
);

```

Project2

Parikshit Dubey
265002524

```

ShippingCost           Money          DEFAULT NULL
);

/**Object: OrdersTable ***/
CREATE TABLE OrdersTable(
OrderID                INT
PRIMARY KEY      IDENTITY    NOT NULL,
CustomerId            INT
REFERENCES Customers(CustomerID),
Status                 VARCHAR(255)  NOT NULL,
ReturnID               INT
REFERENCES ReturnDetails(ReturnID),
OrderDate              DATETIME     NOT NULL,
OrderItemsID           INT
OrderItems(OrderItemsID),
TotalPrice              MONEY        NOT NULL,
ShippingService         VARCHAR(255)
StreetAddress           VARCHAR(255) NOT NULL,
ZIP                    INT
REFERENCES ZIPDetails(ZIP),
ShippingFare             MONEY        NOT NULL,
ExpectedShippingDate   DATETIME    NOT NULL,
ActualShippingDate     DATETIME
ShippingInfo             VARCHAR(255)
);

```

Project2

Parikshit Dubey
265002524

Appendix B(Database population)

```

use MyAmazonCompetitor

SET IDENTITY_INSERT Products ON;

INSERT INTO Products(ProductID, ProductName, Description, Price) VALUES
('10001', 'Christmas Pillow Case', 'Christmas style: It will be a home decor at Christmas, which makes the festival interesting', '3'),
('10002', 'Nestl Reading Pillow', ' Two convenient pockets help you create a cozy nest so that you don't have to get up once you've finally gotten really comfortable!', '20'),
('10003', 'Aladdin (Live Action) (Blu-ray + DVD + Digital Copy)', 'New copy', '30'),
('10011', 'Coalgate', 'Makes your breath fresh', '4'),
('10004', 'Dove Shampoo', 'Makes your hair smooth', '7'),
('10005', 'PS4', 'Lets game', '199'),
('10006', 'Listerine', 'Makes your breath fresh', '12'),
('10007', 'Beanie', 'Best companion for winters', '12'),
('10008', 'HP Pavillion Laptop', 'Unbeatable performance, i9 processor', '1799'),
('10009', 'Apple iPad Pro', '11 inch 2019 version', '799'),
('10010', 'Vaseline', 'Intensive care', '15');

SET IDENTITY_INSERT Products OFF;

SELECT * FROM Products;

```

```

/*
CREATE TABLE Review(
ReviewID          INT           PRIMARY KEY
    IDENTITY NOT NULL,
ProductID        INT           REFERENCES
Products(ProductID),
DateOfReview     DATETIME      NOT NULL,
Comments         VARCHAR(255)   DEFAULT NULL,
Score            INT           DEFAULT NULL,
CustomerName    VARCHAR(255)   DEFAULT NULL,
);
*/

```

```

use MyAmazonCompetitor

SET IDENTITY_INSERT Review ON;

INSERT INTO Review(ReviewID, ProductID, DateOfReview, Comments, Score, CustomerName) VALUES
(1,10010, '02-10-2019', 'Good Product!', 10, 'Aaron'),
(2,10009, '04-03-2019', 'Mediocre Product', 5, 'Lisa'),
(3,10003, '12-04-2019', 'Bad', 1, 'Ruben'),
(4,10007, '07-03-2019', '', 9, 'XYZ'),
(5,10001, '05-10-2018', 'Good Product!', 5, 'Andy'),
(6,10004, '08-07-2017', 'Worth Buy', 8, 'Lima'),
(7,10008, '12-10-2019', 'Bad', 3, 'Ruby'),
(8,10005, '07-07-2019', 'Nice', 7, 'Karen'),
(9,10002, '12-01-2019', 'Bad', 3, 'Ruby'),
(10,10006, '07-12-2018', '', 10, 'Kathy')

```

Project2

Parikshit Dubey
265002524

```

SET IDENTITY_INSERT Review OFF;

SELECT * FROM Review;

/**Object: Wishlist Table ***
CREATE TABLE WishList(
WishListID           INT          PRIMARY KEY          IDENTITY NOT NULL,
Date                 DATETIME      NOT NULL,
ProductID            INT          REFERENCES PRODUCTS(ProductID),
Quantity              INT          NOT NULL
);*/
use MyAmazonCompetitor

SET IDENTITY_INSERT Wishlist ON;

INSERT INTO Wishlist(WishListID,Date,ProductID,Quantity) VALUES
(1,'02-10-2019',10010,3),
(2,'04-03-2019',10009,1),
(3,'12-04-2019',10003,5),
(4,'07-03-2019',10007,9),
(5,'05-10-2018',10001,5),
(6,'08-07-2017',10004,8),
(7,'12-10-2019',10008,3),
(8,'07-07-2019',10005,7),
(9,'12-01-2019',10002,12),
(10,'07-12-2018',10006,10)
SET IDENTITY_INSERT Wishlist OFF;

SELECT * FROM Wishlist;

/**Object: Customers Table ***
CREATE TABLE Customers(
CustomerID           INT          PRIMARY KEY          IDENTITY NOT NULL,
CustomerName          VARCHAR(60)   NOT NULL,
username               VARCHAR(60)   NOT NULL UNIQUE,
CustomerEmail          VARCHAR(255)  NOT NULL UNIQUE,
CustomerPhone          VARCHAR(20)    NOT NULL,
ZIP                   INT          REFERENCES ZIPDetails(ZIP),
ReturnID               INT          NOT NULL /*ADD REFERENCE*/,
WishListID             INT          REFERENCES WishList(WishListID),
ReviewID               INT          REFERENCES Review(ReviewID)
);
*/
use MyAmazonCompetitor

```

Project2

Parikshit Dubey
265002524

```

SET IDENTITY_INSERT Customers ON;

INSERT INTO
Customers(CustomerID,CustomerName,username,CustomerEmail,CustomerPhone,ZIP,ReturnID,WishListID,ReviewID) VALUES
(1,'Allan Sherwood','allan','allan.sherwood@yahoo.com','1922947934',25701,1,2,1),
(2,'Barry Sherwood','barry','barry.sherwood@yahoo.com','19473947934',40206,4,1,8),
(3,'Chris Wood','cwood','c.wood@yahoo.com','19892447934',99611,2,3,7),
(4,'Jason Butt','Jbutt','Jbutt@yahoo.com','19439247934',83638,9,4,6),
(5,'Leo','Leo','Leo@yahoo.com','3362368239',59759,5,5,4),
(6,'Kris','Kris','kris@gmail.com','982378219',04401,6,6,2),
(7,'Minna','Minna','minna@yahoo.com','321232232',58401,3,7,3),
(8,'Sage','Sages','sage@yahoo.com','84783238294',83638,7,8,5),
(9,'Amac Kerry','Akerry','ak@yahoo.com','726237298',97321,8,9,9),
(10,'Donette Louis','Dnloius','adnl@yahoo.com','32162563221',97702,9,10,10)
SET IDENTITY_INSERT Customers OFF;

```

```
SELECT * FROM Customers;
```

```
*****Object: OrderItems table*****
```

```

CREATE TABLE OrderItems(
OrderItemsID          INT                               PRIMARY KEY
                      IDENTITY NOT NULL,
ProductID              INT                               REFERENCES
Products(ProductID),
Quantity                INT                               NOT NULL,
UnitPrice               MONEY                            NOT NULL,
CumulativeCost         MONEY                            NOT NULL
);*/

```

```
use MyAmazonCompetitor
```

```

SET IDENTITY_INSERT OrderItems ON;
INSERT INTO OrderItems(OrderItemsID,ProductID,Quantity,UnitPrice,CumulativeCost) VALUES
(1,10010,3,4,12),
(2,10002,4,12,48),
(3,10001,8,2,16),
(10,10004,10,3,30),
(4,10003,32,2,64),
(5,10006,21,3,63),
(6,10005,54,2,108),
(7,10008,15,3,45),
(8,10007,1200,2,2400),
(9,10009,2100,1,2100)
SET IDENTITY_INSERT OrderItems OFF;

```

```
SELECT * from OrderItems;
```

```

/*
CREATE TABLE ReturnItems(
ReturnItemsID          INT                               PRIMARY KEY
                      IDENTITY NOT NULL,
ProductID              INT                               REFERENCES
Products(ProductID),
Quantity                INT                               DEFAULT NULL
);*/

```

Project2

Parikshit Dubey
265002524

```

);*/
use MyAmazonCompetitor

SET IDENTITY_INSERT ReturnItems ON;
INSERT INTO ReturnItems(ReturnItemsID,ProductID,Quantity) VALUES
(1,10001,3),
(2,10002,4),
(3,10003,2),
(4,10004,5),
(5,10005,10),
(6,10006,21),
(7,10007,14),
(8,10008,56),
(9,10009,25),
(10,10010,54)
SET IDENTITY_INSERT ReturnItems OFF;

SELECT * FROM ReturnItems;

/******Object: Return details*****/
CREATE TABLE ReturnDetails(
ReturnID           INT                               PRIMARY KEY
                      IDENTITY NOT NULL,
ReturnItemsID      INT                               REFERENCES
ReturnItems(ReturnItemsID),
DateOfReturn       DATETIME,
StatusOfReturn     VARCHAR(255)                    DEFAULT NULL,
ShippingInfo        VARCHAR(255)                    DEFAULT NULL,
ShippingCost        Money                           DEFAULT NULL
);*/
use MyAmazonCompetitor

SET IDENTITY_INSERT ReturnDetails ON;
INSERT INTO
ReturnDetails(ReturnID,ReturnItemsID,DateOfReturn,StatusOfReturn,ShippingInfo,ShippingCos
t) VALUES
(1,1,'01-04-2019','Returned','FedEx',3),
(2,2,'01-05-2019','Returned','FedEx',4),
(3,3,'12-04-2018','Not Returned','','0'),
(4,4,'04-04-2019','Returned','FedEx',2),
(5,5,'05-12-2019','Returned','FedEx',12),
(6,6,'11-01-2019','Not Returned','','0'),
(7,7,'11-05-2019','Not Returned','','0'),
(8,8,'07-14-2018','Returned','USPS',4),
(9,9,'04-06-2019','Returned','FedEx',1),
(10,10,'10-12-2019','Returned','UPS',0)
SET IDENTITY_INSERT ReturnDetails OFF;

SELECT * FROM ReturnDetails;

/*
****Object: OrdersTable ****/
CREATE TABLE OrdersTable(

```

Project2

Parikshit Dubey
265002524

```

OrderID           INT
PRIMARY KEY      IDENTITY NOT NULL,
CustomerId        INT
REFERENCES Customers(CustomerID),
Status            VARCHAR(255) NOT NULL,
ReturnID          INT
REFERENCES ReturnDetails(ReturnID),
OrderDate         DATETIME    NOT NULL,
OrderItemsID     INT
OrderItems(OrderItemsID),
TotalPrice        MONEY      NOT NULL,
ShippingService   VARCHAR(255)
StreetAddress     VARCHAR(255) NOT NULL,
ZIP               INT
REFERENCES ZIPDetails(ZIP),
ShippingFare      MONEY      NOT NULL,
ExpectedShippingDate DATETIME NOT NULL,
ActualShippingDate DATETIME
ShippingInfo       VARCHAR(255)
DEFAULT NULL,
DEFAULT NULL
);*/

```

```
use MyAmazonCompetitor
```

```

SET IDENTITY_INSERT OrdersTable ON;
INSERT INTO
OrdersTable(OrderID,CustomerID,Status,ReturnID,OrderDate,OrderItemsID,TotalPrice,Shipping
Service,StreetAddress,ZIP,ShippingFare,ExpectedShippingDate,ActualShippingDate,ShippingIn
fo) VALUES
(1,1,'Processing',1,'06-07-2019',1,14,'UPS','111 ABC Street',25701,3,'06-08-2019','',''),
(2,2,'Processing',2,'06-09-2019',2,16,'USPS','222 ABC Street',25701,4,'06-09-
2019','',''),
(3,3,'Processing',3,'07-07-2019',3,14,'FedEx','333 PYS Street',40206,4,'07-08-
2019','',''),
(4,4,'Processing',4,'07-09-2019',4,213,'USPS','276 RTT Street',97702,2,'07-09-
2019','',''),
(5,5,'Processing',5,'06-07-2019',6,148,'UPS','111 RON Street',25701,3,'06-08-
2019','',''),
(6,6,'Processing',6,'06-09-2019',5,1676,'USPS','222 West Street',04401,4,'06-09-
2019','',''),
(7,7,'Processing',7,'07-07-2019',8,1400,'FedEx','333 North Street',40206,4,'07-08-
2019','',''),
(8,8,'Processing',8,'07-09-2019',7,2130,'USPS','276 Euclid Street',58401,2,'07-09-
2019','',''),
(9,9,'Processing',9,'07-09-2019',10,243,'USPS','276 RTT Street',97702,2,'07-09-2019','07-
09-2019','Reached'),
(10,10,'Processing',10,'06-07-2019',9,187,'UPS','111 RON Street',25701,5,'06-08-
2019','06-09-2019','Reached')
SET IDENTITY_INSERT OrdersTable OFF;

```

```
SELECT * FROM OrdersTable;
```

```
/*('25701','Huntington','WV'),
('40206','Louisville','KY'),
('99611','Kenai','AK'),
```

Project2

Parikshit Dubey
265002524

```

('97702','Bend','OR'),
('97321','Albany','OR'),
('83638','McCall','ID'),
('59759','Whitehall','MT'),
('59801','Missoula','MT'),
('58401','Jamestown','ND'),
('04401','Bangor','ME')*/



/*
ProductsStoredID      INT          PRIMARY KEY
IDENTITY      NOT NULL,
ProductID       INT          REFERENCES
Products(ProductID),
NumberInStock    INT          NOT NULL,
NumberOnWay     INT          DEFAULT NULL,
NumberOnReturn   INT          DEFAULT NULL,


*/
use MyAmazonCompetitor

SET IDENTITY_INSERT ProductsStored ON;

INSERT INTO
ProductsStored(ProductsStoredID,ProductID,NumberInStock,NumberOnWay,NumberOnReturn)
VALUES
(1,10003,4,1,0),
(2,10002,12,3,2),
(3,10001,34,11,1),
(4,10006,123,32,2),
(5,10004,1211,154,44),
(6,10007,9,8,0),
(7,10005,36,2,0),
(8,10008,78,0,12),
(9,10011,3892,230,0),
(10,10010,3213,789,342)
SET IDENTITY_INSERT ProductsStored OFF;

SELECT * FROM ProductsStored;

use MyAmazonCompetitor

SET IDENTITY_INSERT Suppliers ON;

INSERT INTO
Suppliers(SupplierID,SupplierName,StreetName,ZIP,SupplierPhone,SupplierEmailID,NumberOfProducts) VALUES
(1,'ABC','123 Rhongthen Street','25701',8459734383,'abc@xyz.com',3),
(2,'ABZ','136 Westcott Street','40206',8459434333,'abz@xyz.com',2),
(3,'Yu Xe','765 Stratham Street','99611',83329232324,'yu@xyz.com',10),
(4,'PVK corp','677 Westend ave','97702',142536223232,'pvk@xyz.com',12),
(5,'Zen corp','664 Northend ave','97321',236623232322,'zen@xyz.com',7),
(6,'Ram Goods','7675 Mountain ave','83638',3486565656,'ram@xyz.com',20),
(7,'Food Goods','734 Park ave','59759',128962324,'food@good.com',2),
(8,'UUI Goods','1 Teal ave','59801',11035142536,'uui@uui.com',12),
(9,'Zec corp','664 Souththend ave','97702',67638742536,'zec@uis.com',3),
(10,'Ramirez Corp','7675 Riverside ave','99611',3483766221,'ramirez@wqq.com',9)

```

Project2

Parikshit Dubey
265002524

```

SET IDENTITY_INSERT Suppliers OFF;
SELECT * FROM Suppliers;

use MyAmazonCompetitor

SET IDENTITY_INSERT ZIPDetails ON;

INSERT INTO ZIPDetails(ZIP, City, State) VALUES
('25701','Huntington','WV'),
('40206','Louisville','KY'),
('99611','Kenai','AK'),
('97702','Bend','OR'),
('97321','Albany','OR'),
('83638','McCall','ID'),
('59759','Whitehall','MT'),
('59801','Missoula','MT'),
('58401','Jamestown','ND'),
('04401','Bangor','ME')
SET IDENTITY_INSERT ZIPDetails OFF;

/*CREATE TABLE WarehouseToProductsStored(
WarehouseID          INT                               REFERENCES
Warehouse(WarehouseID),
ProductsStoredID      INT                               REFERENCES
ProductsStored(ProductsStoredID)
)

CREATE TABLE ProductsToSupplierLink(
ProductID            INT                               REFERENCES
Products(ProductID),
SupplierID           INT                               REFERENCES
Suppliers(SupplierID),
);

CREATE TABLE WarehouseSupplierLink(
WarehouseID          INT                               REFERENCES
Warehouse(WarehouseID),
SupplierID           INT                               REFERENCES
Suppliers(SupplierID),
);
;*/

USE MyAmazonCompetitor

INSERT INTO WarehouseToProductsStored(WarehouseID,ProductsStoredID) VALUES
(1,1),
(1,2),
(1,10),
(1,8),
(2,9),
(2,7),
(1,5),
(3,5),
(7,3),

```

Project2

Parikshit Dubey
265002524

```
(7,1),
(3,6),
(10,3),
(10,10),
(9,4),
(8,7);
```

```
SELECT * From WarehouseToProductsStored;
```

```
USE MyAmazonCompetitor
INSERT INTO ProductsToSupplierLink(ProductID,SupplierID) VALUES
(10001,1),
(10004,2),
(10002,10),
(10001,8),
(10002,9),
(10002,7),
(10001,5),
(10003,5),
(10007,3),
(10007,1),
(10003,6),
(10010,3),
(10010,10),
(10009,4),
(10008,7);
SELECT * From ProductsToSupplierLink;
```

```
USE MyAmazonCompetitor
```

```
INSERT INTO WarehouseSupplierLink(WarehouseID,SupplierID) VALUES
(1,1),
(1,2),
(1,10),
(1,8),
(2,9),
(2,7),
(1,5),
(3,5),
(7,3),
(7,1),
(3,6),
(10,3),
(10,10),
(9,4),
(8,7);
```

```
SELECT * From WarehouseSupplierLink;
```

```
use MyAmazonCompetitor

SET IDENTITY_INSERT Warehouse ON;
INSERT INTO Warehouse(WarehouseID,StreetAddress,ZIP) VALUES
```

Project2

Parikshit Dubey
265002524

```
(1,'123 gthen Street','25701'),  
(2,'136 cott Street','40206'),  
(3,'765 Tham Street','99611'),  
(4,'677 Endstreet ave','97702'),  
(5,'664 EndRoad ave','97321'),  
(6,'7675 Mountain Clark ave', '83638'),  
(7,'734 Park ave', '59759'),  
(8,'1 Teal ave', '59801'),  
(9,'664 Souththend ave','97702'),  
(10,'7675 Riverside ave', '99611')  
SET IDENTITY_INSERT Warehouse OFF;
```

```
SELECT * FROM Warehouse;
```

Project2

Parikshit Dubey
265002524

Appendix C(DatabaseManipulation)

```

USE MyAmazonCompetitor;
Create Role NewRole
Grant Update on Products to NewRole
Grant Update, Insert on ZIPDetails to NewRole
Grant Update, Insert, Delete on Warehouse to NewRole
Exec sp_addrolemember db_datareader , NewRole

CREATE LOGIN AmazonCompetitorDatabase01 with password = 'zzz123',
Default_Database = MyAmazonCompetitor
Create USER ParikshitDubey for login AmazonCompetitorDatabase01
Exec sp_addrolemember NewRole, ParikshitDubey;

GO
Create Schema ParikshitDubs
Go
Alter Schema ParikshitDubs Transfer dbo.Products
Alter User ParikshitDubey with Default_Schema= ParikshitDubs
Grant Delete,Select,Update, Insert, Execute on Schema :: ParikshitDubs to ParikshitDubey

USE MyAmazonCompetitor;
GO
CREATE PROC procdateSelect
@DateMin varchar(30) = NULL,
@DateMax varchar(30) = NULL
AS
IF @DateMin IS NULL OR @DateMax IS NULL
THROW 50001, 'Please enter the minimum and maximum date', 1;

IF NOT(ISDATE(@DateMin) = 1 AND ISDATE(@DateMin) = 1)
THROW 50001, 'Please enter a valid date format', 1;

IF CAST(@DateMin AS datetime) > CAST(@DateMax AS datetime)
THROW 50001, 'DateMin cannot be greater than DateMax', 1;

SELECT ReviewID, DateOfReview, Score
FROM Review
WHERE DateOfReview BETWEEN @DateMin AND @DateMax
ORDER BY DateOfReview;

BEGIN TRY
EXEC procdateSelect '12/12/2019';

END TRY

```

Project2

Parikshit Dubey
265002524

```
BEGIN CATCH
PRINT 'Error Number - ' + CONVERT(varchar(30), ERROR_NUMBER());
PRINT 'Error Message - ' + CONVERT(varchar(50), ERROR_MESSAGE());
END CATCH
```

```
USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ProductView AS
SELECT ProductID, ProductName, Description, Price
FROM Products;')

SELECT * FROM ProductView;
```

```
USE MyAmazonCompetitor
EXECUTE('CREATE VIEW ReviewView AS
SELECT ProductID, Score, CustomerName
FROM Review;')
```

```
SELECT * FROM ReviewView;

USE MyAmazonCompetitor
EXECUTE('CREATE VIEW RatingView AS
SELECT ProductName, Score, CustomerName
FROM Review JOIN Products
ON Review.ProductID = Products.ProductID;')

SELECT * FROM RatingView;
```

```
EXECUTE('CREATE VIEW ExpensiveItems AS
SELECT ProductName, Description, Price
FROM Products
WHERE Price > 500;')
```

```
SELECT * FROM ExpensiveItems;
```

```
CREATE VIEW OrdersInProcessing AS
SELECT OrderID, OrderDate, Status, TotalPrice, ShippingInfo, ExpectedShippingDate
FROM OrdersTable
WHERE Status = 'Processing';
```

```
SELECT * FROM OrdersInProcessing;
```

```
CREATE TRIGGER TRIGGER1
ON Products
AFTER INSERT,UPDATE,DELETE
AS RAISERROR('Product table has some updates, Please review',16,10)
GO
```

```
SET IDENTITY_INSERT Products ON;
```

Project2

Parikshit Dubey
265002524

```
INSERT INTO Products(ProductID, ProductName, Description, Price) VALUES
('1011', 'Phone Case', 'Christmas style: It will be a home decor at Christmas, which makes
the festival interesting', '200')
SET IDENTITY_INSERT Products OFF;

SELECT * FROM Products;
```