

**Course Number:** MIS 6326.501

**Instructor:** Dr. Young U. Ryu

Spring, 2019

**Group Members:**

Mohit Agarwal

Atharva Bendre

Revati Dalvi

Pranesh Chitre

Geoffrey Golden

Krish Mehta

**Project Report Contents**

Part 1. Company Background

Part 2. Database Scope and Table Description

Part 3. Entity-Relationship Diagram

Part 4. Relationship Database Schema

Part 5. Data Input Screen Printout

Part 6. Sample Reports

Part 7. Menu Printouts

Part 8. Contribution Statement

**Part 1. Company Background**

Code Blue Jewelry is a premium medical ID jewelry company. The company offers personalized medical ID jewelry to patients with chronic illnesses. Their products include necklaces, bracelets, and stethoscope accessories, each with personalization added at the time of sale. These customizations include information about their condition that people might need in order to assist the wearer, but that the wearer themselves might be unable to communicate in a medical emergency.

Code Blue Jewelry’s marketing differentiator is style. Medical ID’s in the forms they offer already exist, but Code Blue Jewelry recognized the unmet need for these ID’s to be fashionable. Patients may not wear their ID’s or try to partially hide them if the ID would otherwise “brand” them as a “sick person”. With fashionable jewelry, patients can wear these ID’s proudly in places of prominence where they don’t immediately stick out as medical in nature, but where they would easily be found if needed. The medical need for these ID’s as well as the lack of competition in the market allows Code Blue Jewelry to charge a premium price for their products without lowering demand.

As Code Blue Jewelry is a new startup experiencing overwhelming demand, the company needs to build a database to track orders and maintain customer relationships. Currently, customer orders and fulfillment are completed through e-mail, which has become inadequate for the scale of their business. Sales Managers are spending more time combing their e-mail inboxes to track orders than they are in fulfilling these orders. Code Blue Jewelry needs a relational database that can interact with the website Code Blue Jewelry plans to build, and this database needs to be able to track customers, orders, shipments, and products sold.

**Part 2. Database Scope and Table Description**

The company’s database primary function is to track customer order fulfillment and product inventory sold. Customer orders need to be logged as they come in and updated as necessary after the order is placed. Customers need to be able to view transaction histories to manage billing and shipping, while sales managers need to be able to view these histories for order fulfillment, planning and marketing analytics. In order to track product inventory sold, product inventory needs to be searchable. Customers need to know if the base product for their medical ID is currently out of stock, and sales managers need to track this inventory to source for orders and plan replenishment schedules. This database uses seven tables, as recorded below.

**1. Customer Table**

When customers place an order, their personal information is stored within the customer table. This table contains vital customer statistics to be used in order fulfillment, marketing analytics and customer service. Eight attributes in addition to the primary key are used to record this information, allowing each to be searched and aggregated separately. When customers initiate their first order, they are automatically issued a random, unique ID number. This ID number becomes the Primary Key of their record, and the following attributes are also recorded:

1. Customer ID (CustID) - PK
2. First Name (CustFirst)
3. Last name (CustLast)
4. Address Line 1 (CustAddrL1)
5. Address Line 2 (CustAddrL2)
6. Customer City (CustCity)
7. Customer State (CustState)
8. Customer Zip (CustZip)
9. Customer Phone (CustPhone)
10. Customer E-mail (CustEmail)

**2. Recipient**

When orders are placed, recipient information must also be recorded, as often customers buy these products as gifts for their loved ones in need and thus the recipient is not always the same person as the customer who initiated the order. However, other than this differentiator and the

lack of a need for an e-mail address for the recipient, the recipient table has all the same needs as the customer table, and thus the same attributes. Recipients receive a unique, random recipient ID at the time of purchase, which becomes the primary key of their record, and the following attributes are recorded:

1. Recipient (RcptID) - PK
2. First Name (RcptFirst)
3. Last name (RcptLast)
4. Address Line 1 (RcptAddrL1)
5. Address Line 2 (RcptAddrL2)
6. Recipient City (RcptCity)
7. Recipient State (RcptState)
8. Recipient Zip (RcptZip)
9. Recipient Phone (RcptPhone)

**3. Sales Manager**

Sales Managers are the backbone of fulfillment, customizing each product to the specifications of the order and initiating the shipment. For the purposes of this order fulfillment database, only the Sales Manager name needs to be stored, since customers contact a central e-mail and phone number instead of each Sales Manager’s individual phone number or home address. Sales Managers are assigned a random, unique ID#, which becomes the primary key of their record, and the following attributes are recorded:

1. Sales Manager ID (SalesManID) - PK
2. First Name (SalesManFirst)
3. Last Name (SalesManLast)
4. Sales Manager E-mail (SalesManEmail)

**4. Carrier**

The order must be delivered by a carrier, with relevant contact info included. Both customers and sales managers need to have the ability to find the appropriate carrier contact information

and contact them in the event of an issue with the shipment. This information is stored in the Carrier table, with the Carrier Name becoming the primary ID, and the Carrier phone number stored as the sole relevant attribute.

1. Carrier Name (CarrName) – PK
2. Carrier Phone (CarrPhone)

**5. Order**

As a database whose primary purpose is order fulfillment, the order table forms the intersection of all previous tables. The order table contains the primary keys of the Customer, Recipient, Sales Manager and Carrier tables stored as foreign keys. It also records the date of the order, date of shipment for the order, and the tracking number of the shipment. This information is stored as the following attributes:

1. Order ID (OrderID) - PK
2. Customer ID (CustID) – FK
3. Recipient ID (RcptID) – FK
4. Sales Manager ID (SalesManID) – FK
5. Carrier Name (CarrName) – FK
6. Order Date (OrderDate)
7. Shipping Date (ShipDate)
8. Tracking Number (TrackNo)

**6. Order Line Item**

Each order may have many products, and each product may be assigned to many orders. This many-to-many relationship necessitates an Order Line Item table to facilitate that relationship, with this new table also serving many other functions. The primary keys OrderID from the Order table and JewelryID from the Jewelry table are foreign keys in the Order Line Item table, and together act as the composite primary key for each entry. The Order Line Item table contains the medical ID inscription unique to each order that communicates the person’s medical condition, providing the vital medical information that people need to know to meet the patient’s needs in an emergency. It also includes information on the custom decorations added to each product during the order process, as well as the unit price agreed to at the point of sale and the quantity of each item sold. This information is stored as the following attributes:

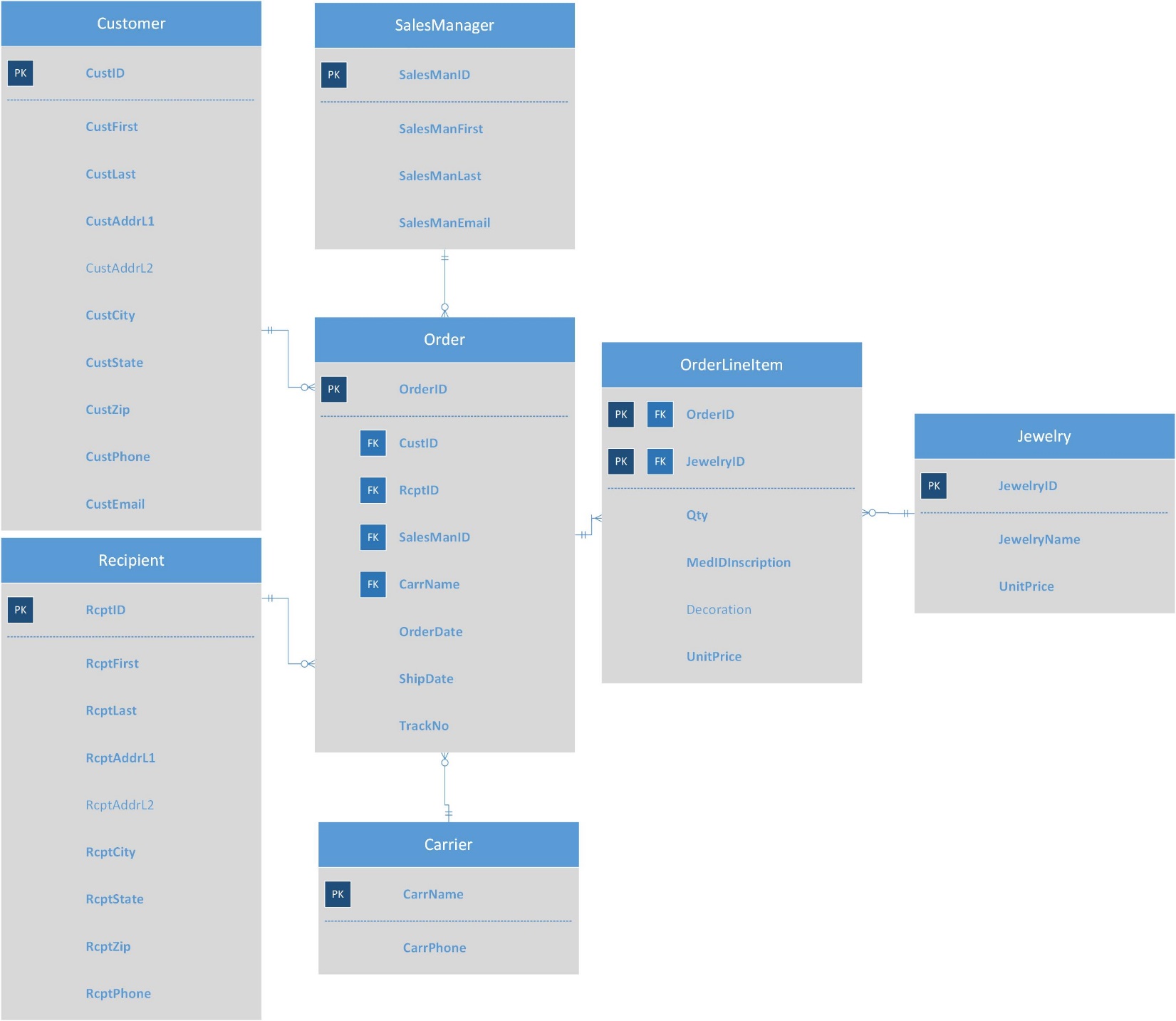
1. Jewelry ID, Order ID (JewelryID, OrderID) – FK’s, Composite PK
2. Quantity of Product Sold (Qty)
3. Medical ID Inscription (MedIDInscription)
4. Decoration (Decoration)
5. Unit Price (UnitPrice)

**7. Jewelry**

The Jewelry table tracks inventory of each product sold. The company currently only offers three products – necklaces, bracelets and stethoscope accessories. The customization occurs at the point of sale, so this information is not stored in the product table, instead being stored in the Order Line Item table. This leaves the name of the product and current price of the product to be stored in the Jewelry table, and a unique random Jewelry ID assigned to each product sold to act as the primary key for the table. Information is stored as follows:

1. Jewelry ID (JewelryID) – PK
2. Product Name (JewelryName)
3. Unit Price (UnitPrice)

**Part 3. Entity-Relationship Diagram**

****

**Part 4. Relational Database Schema**

Customer (CustId, CustFirst, CustLast, CustAddrL1, CustAddrL2, CustCity, CustState, CustZip, CustPhone, CustEmail)

Customer.CustLast NOT NULL, Customer.CustFirst NOT NULL, Customer.CustAddrL1 NOT NULL, Customer.CustCity NOT NULL, Customer.CustState NOT NULL, Customer.CustZip NOT NULL, Customer.CustPhone NOT NULL, Customer.CustEmail NOT NULL

Recipient (RcptID, RcptFirst, RcptLast, RcptAddrL1, RcptAddrL2, RcptCity, RcptState, RcptZip, RcptPhone)

Recipient.RcptFirst NOT NULL, Recipient.RcptLast NOT NULL,

Recipient.RcptAddrL1 NOT NULL, Recipient.RcptCity NOT NULL,

Recipient.RcptState NOT NULL, Recipient.RcptZip NOT NULL,

Recipient.RcptPhone NOT NULL

SalesManager (SalesManID, SalesManFirst, SalesManLast, SalesManEmail)

SalesManager.SalesManFirst NOT NULL, SalesManager.SalesManLast NOT NULL, SalesManager.SalesManEmail NOT NULL

Carrier (CarrName, CarrPhone)

Carrier.CarrPhone NOT NULL

Order (OrderID, CustID, RcptID, SalesManID, CarrName, OrderDate, ShipDate, TrackNo Order.CustID references Customer.CustID, Order.RcptID references Recipient.RcptID, Order.SalesManID references SalesManager.SalesManID, Order.CarrName references Carrier.CarrName)

Order.OrderDate NOT NULL, Order.ShipDate NOT NULL, Order.TrackNo NOT NULL

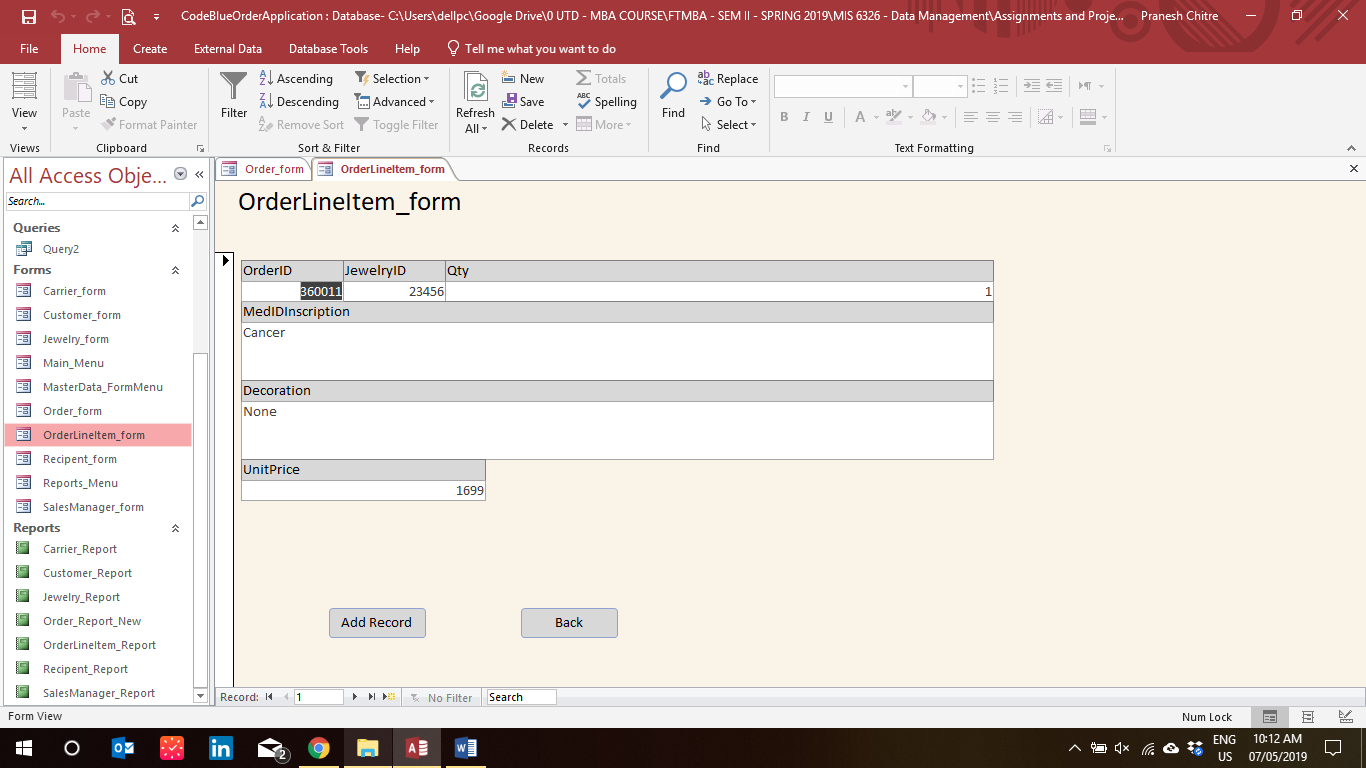
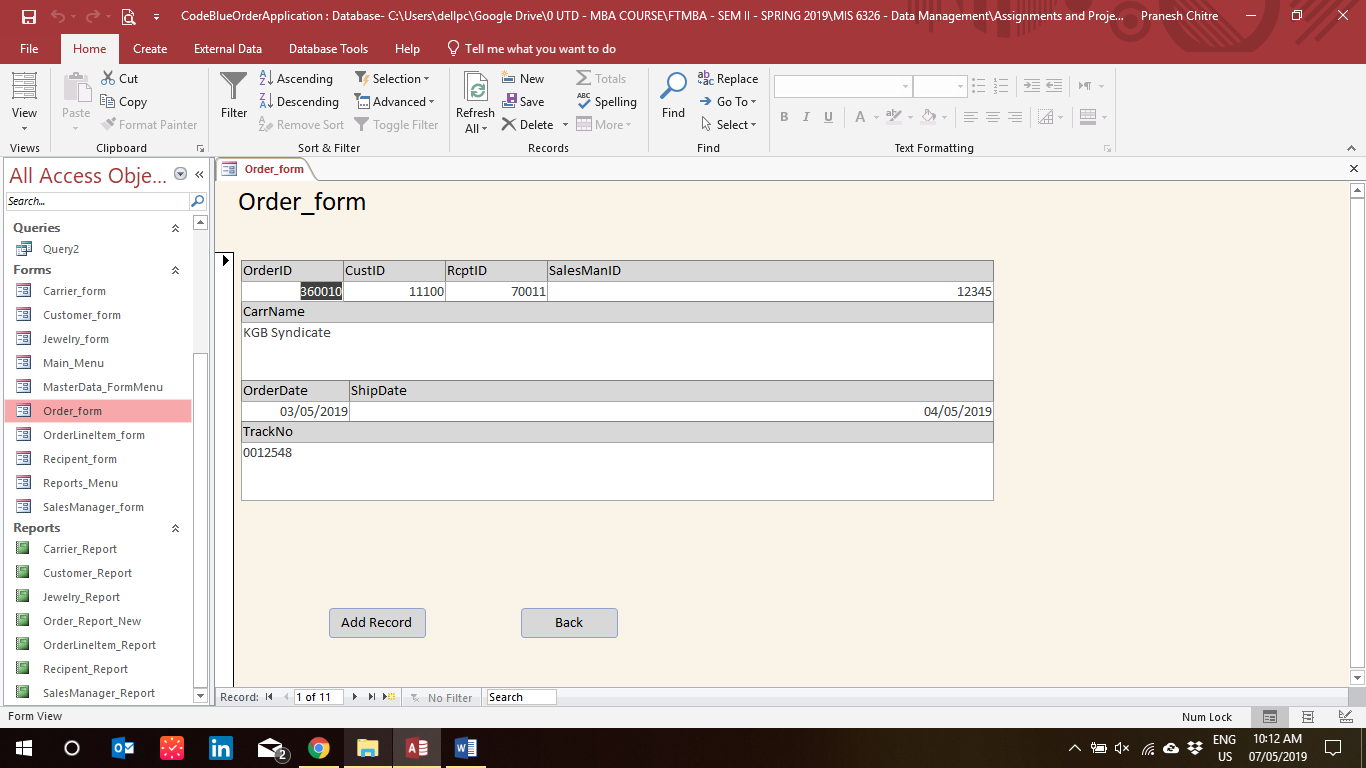
OrderLineItem (OrderID, JewelryID, Qty, MedIdInscription, Decoration, UnitPrice

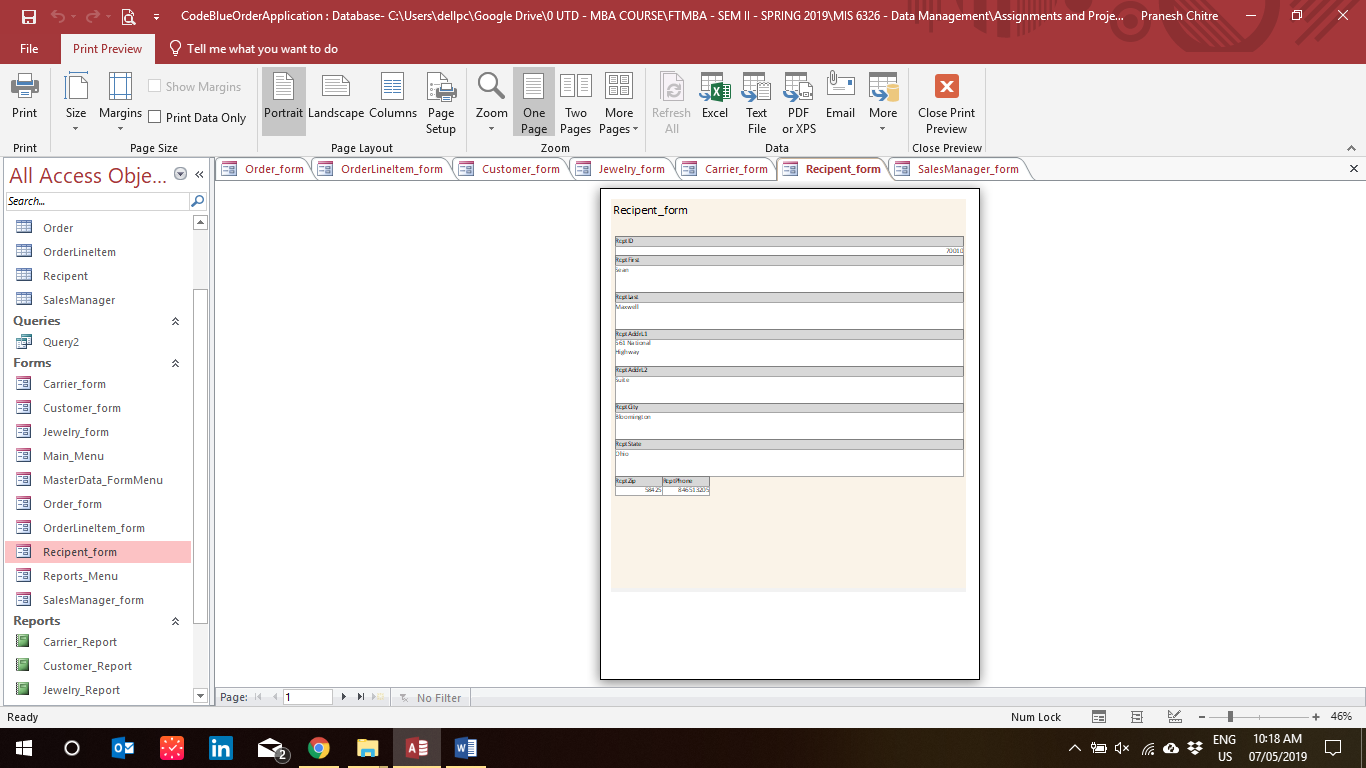
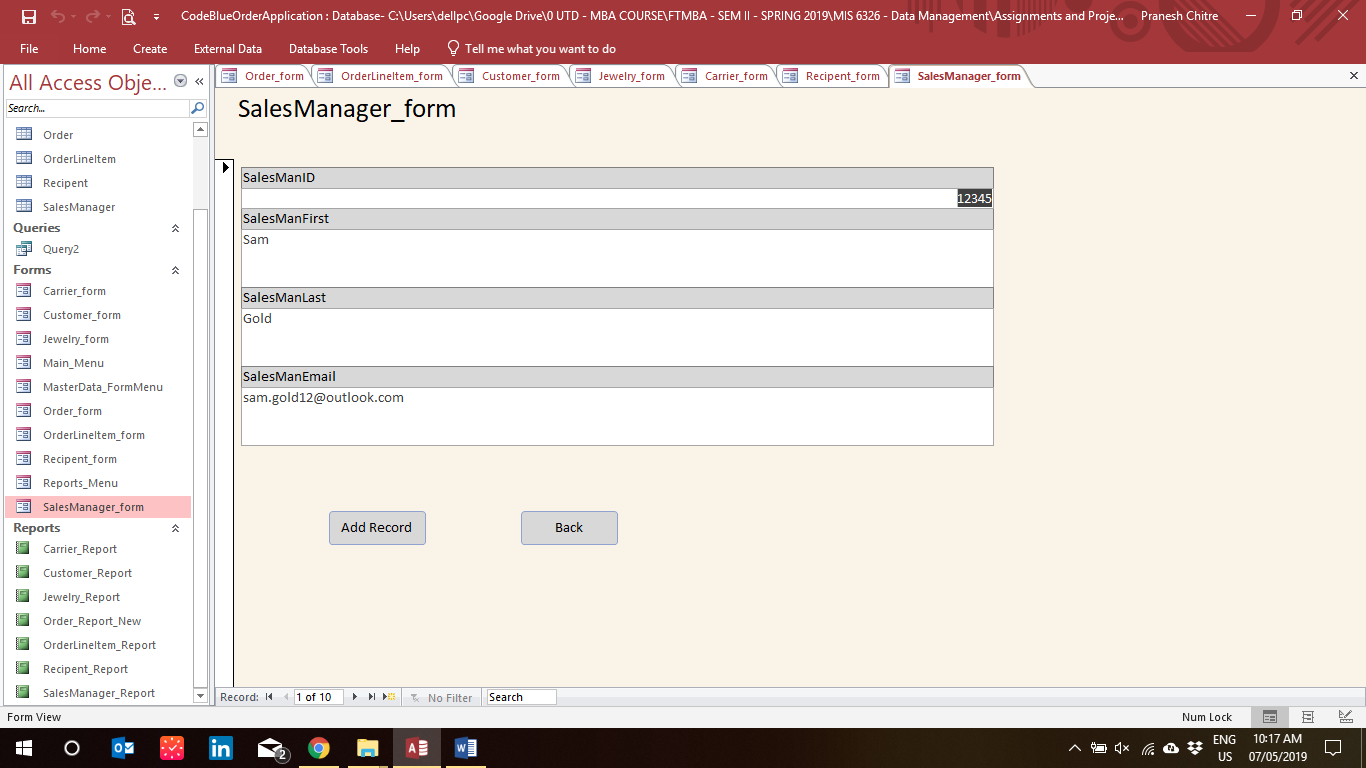
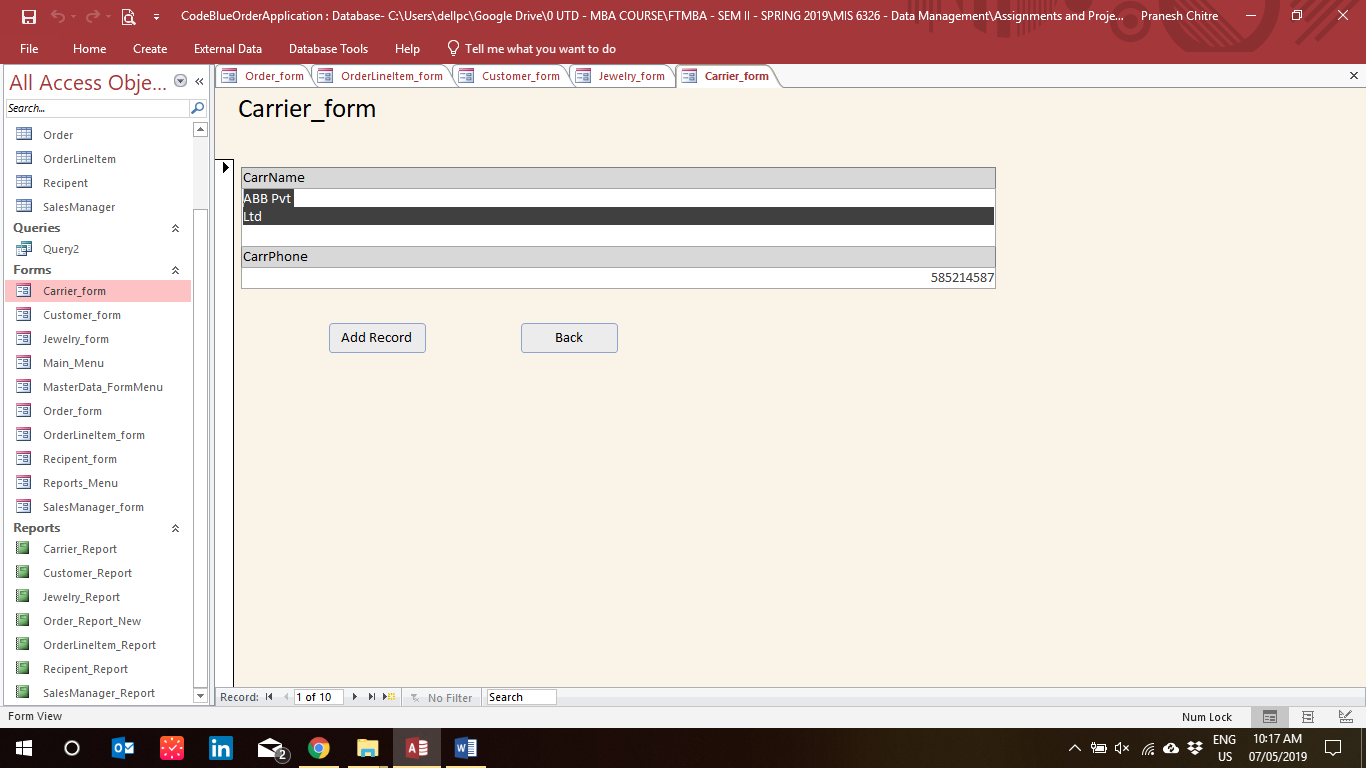
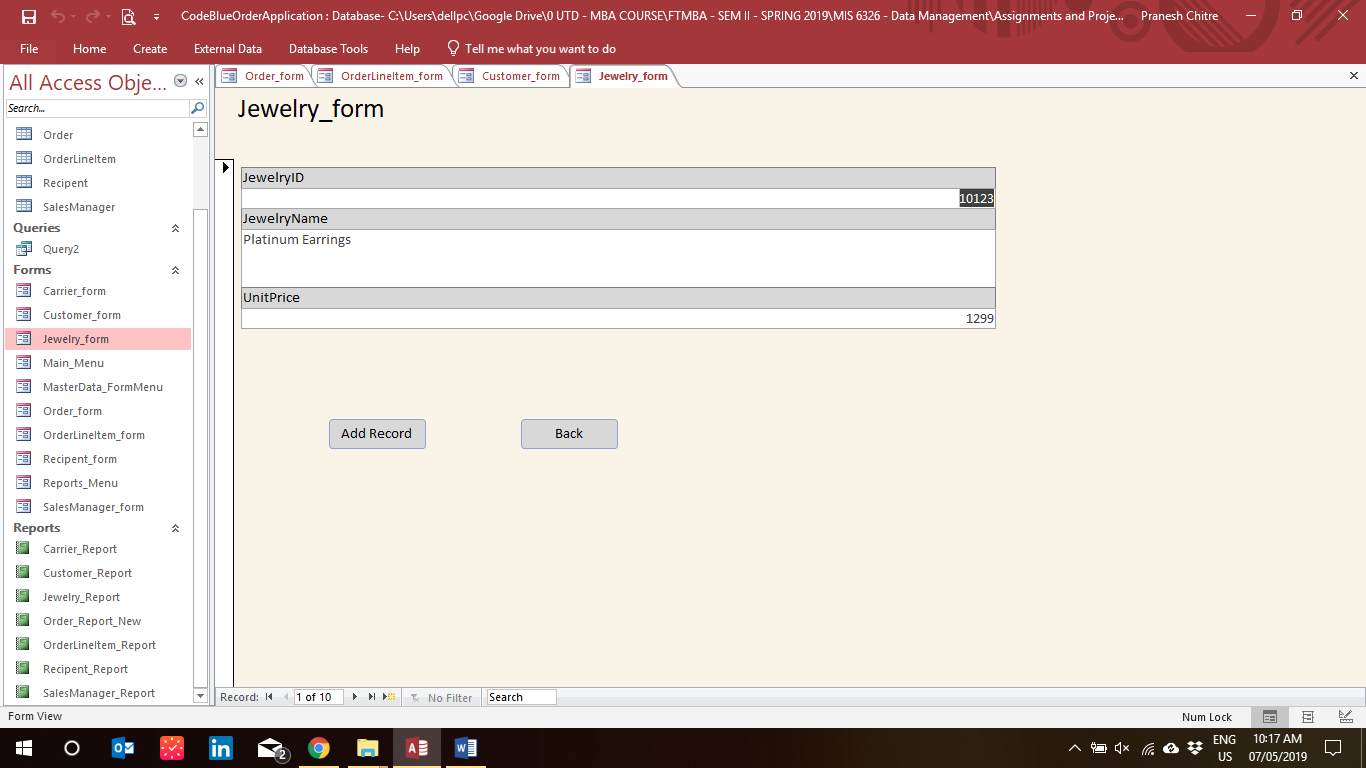
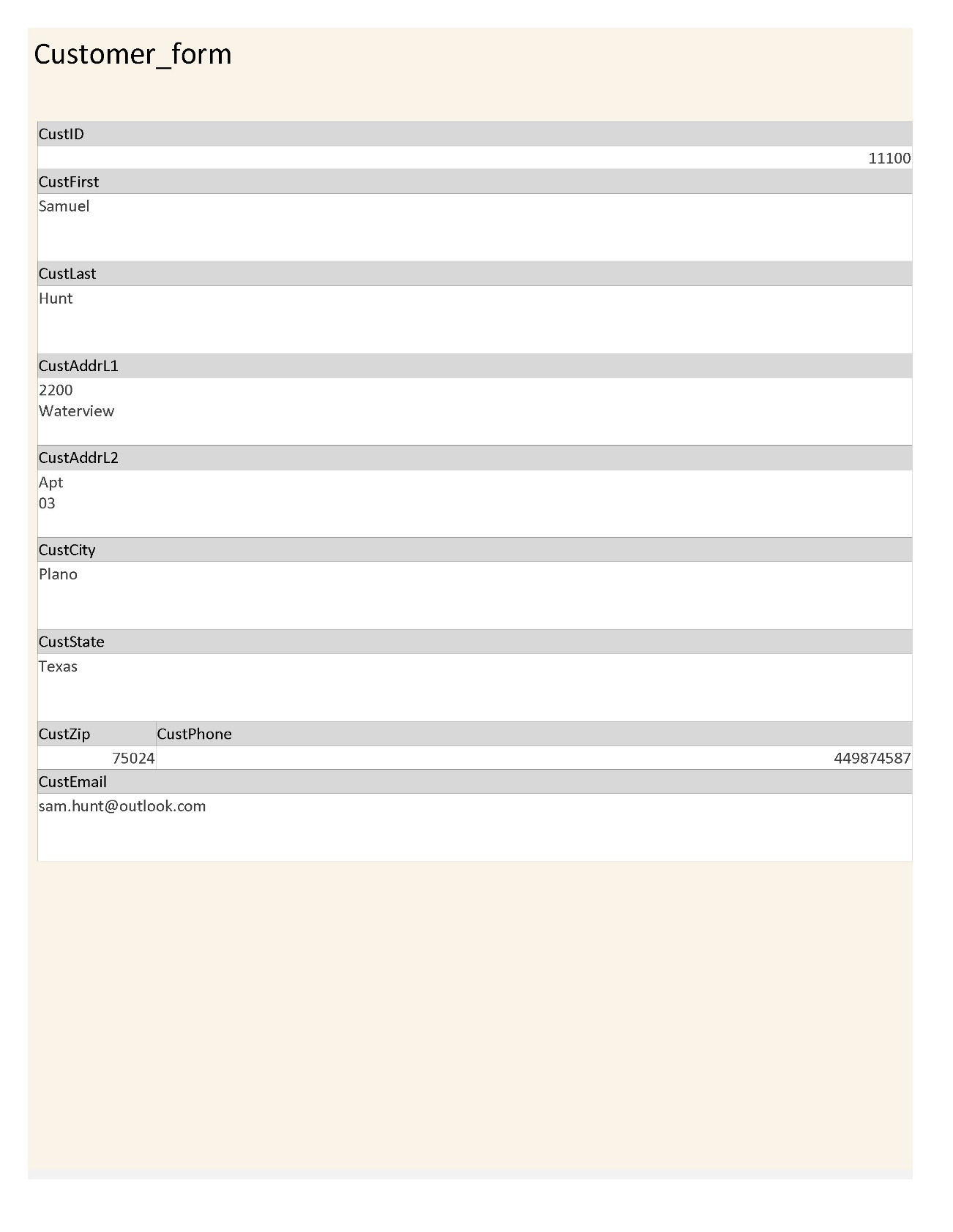
OrderLineItem.OrderID references Order.Order\_ID, OrderLineItem.JewelryID references Jewelry.JewelryID)

OrderLineItem.Qty NOT NULL, OrderLineItem.MedIDInscription NOT NULL, OrderLineItem.UnitPrice NOT NULL

Jewelry (JewelryID, JewelryName, UnitPrice)

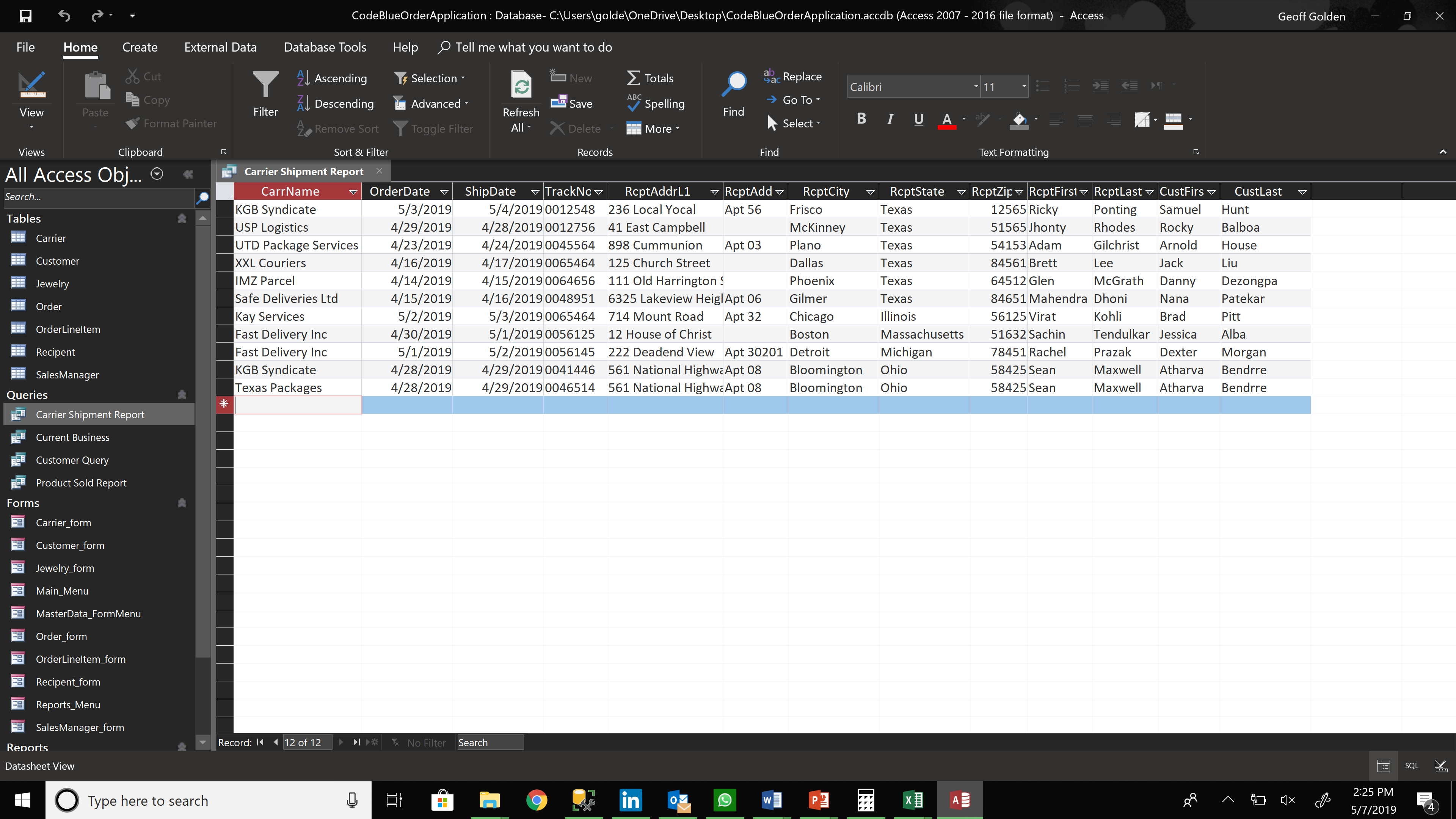
Jewelry.JewelryName NOT NULL, Jewelry.UnitPrice NOT NULL

**Part 5. Data Input Screen Printout**

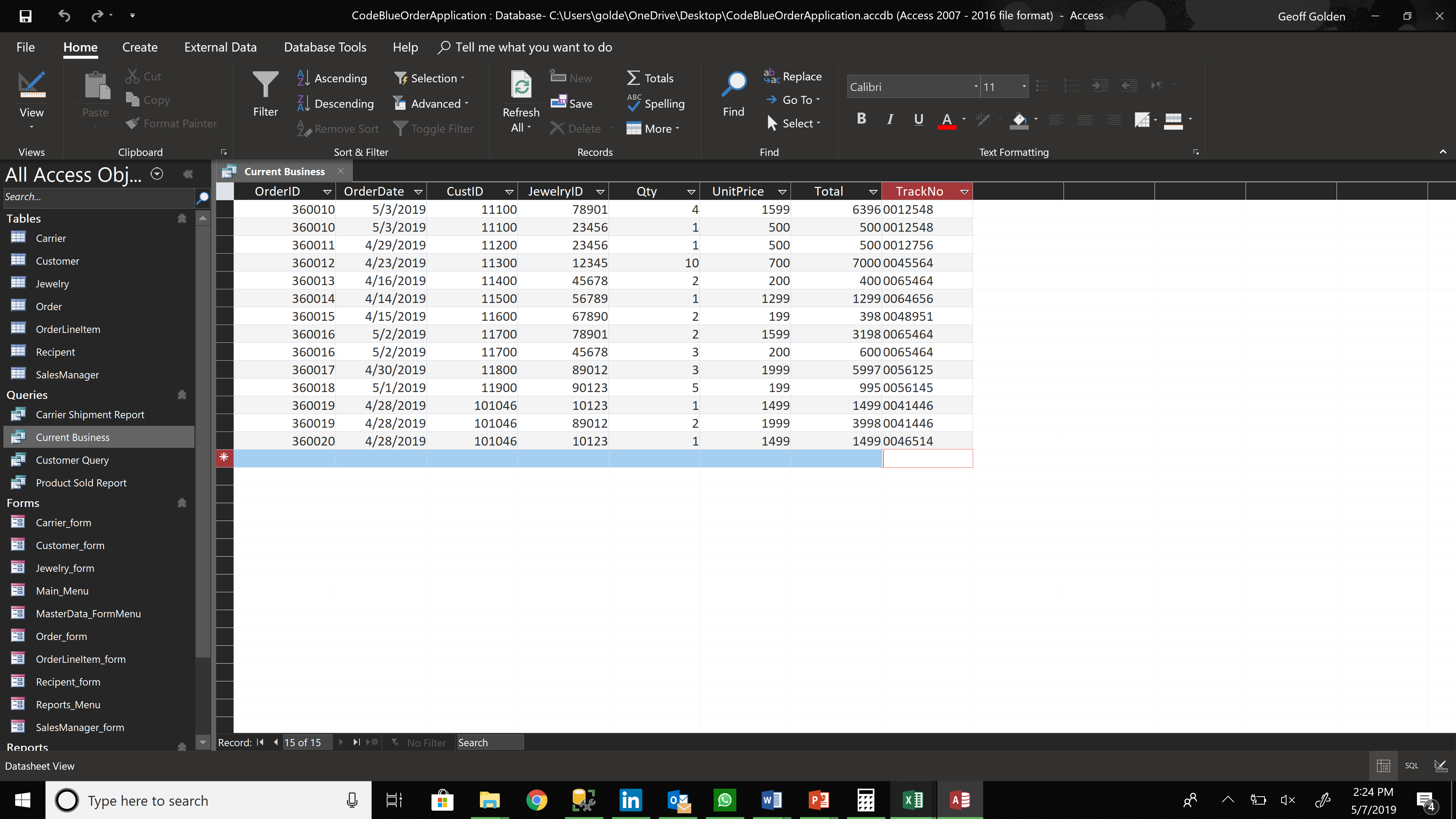
****

**Part 6. Sample Reports**

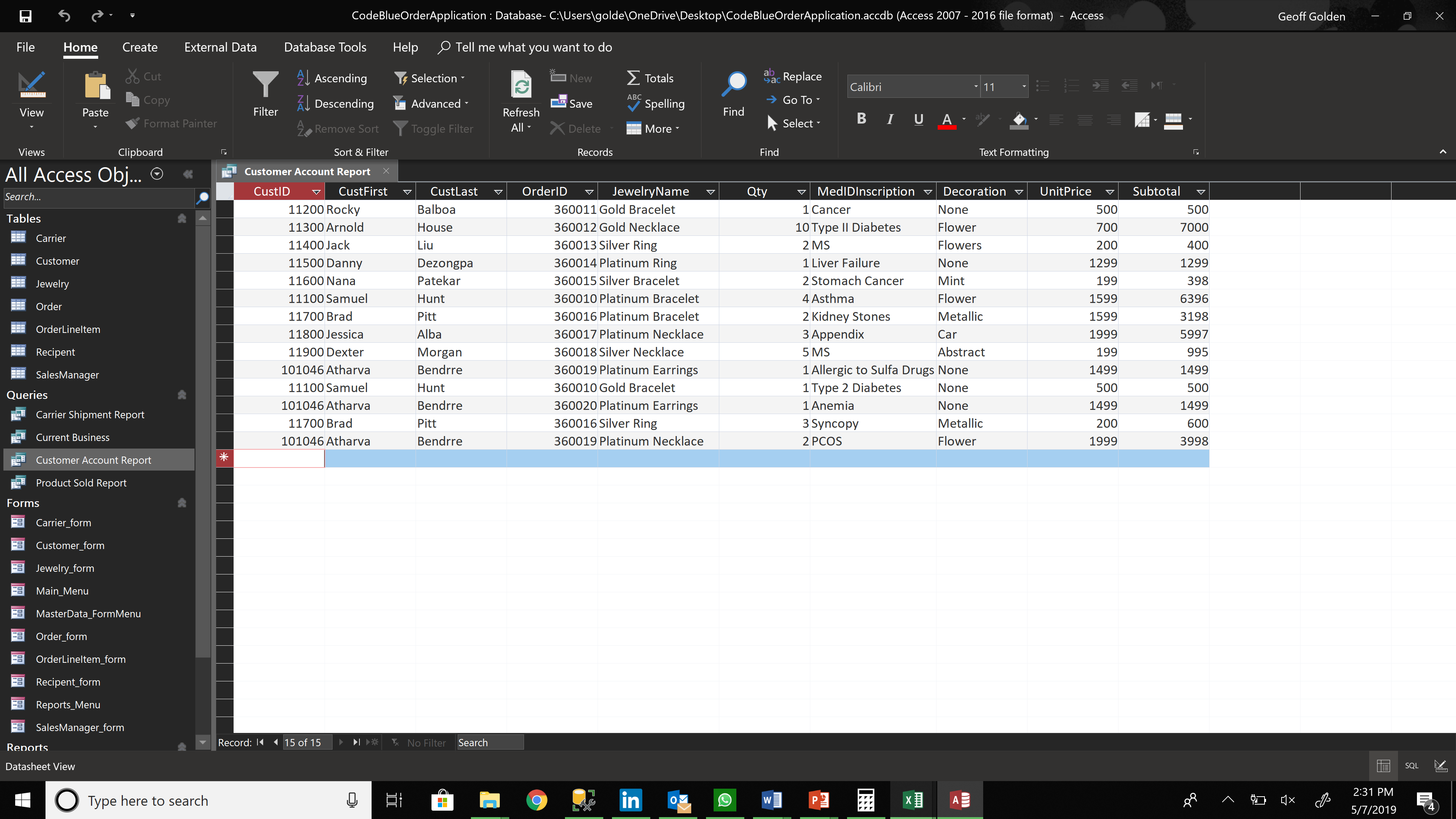
**Carrier Shipment Report**



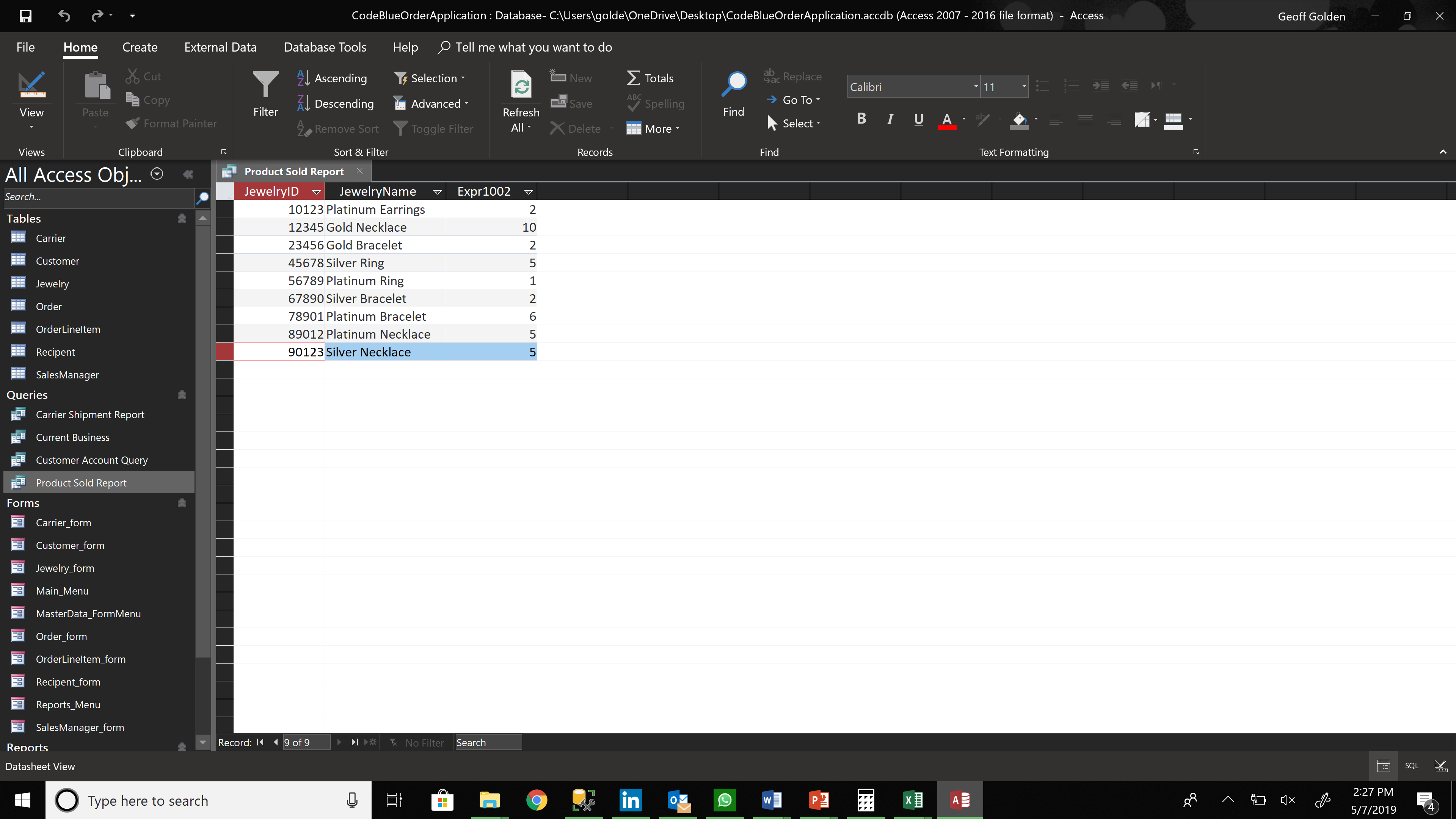
**Current Business Report**



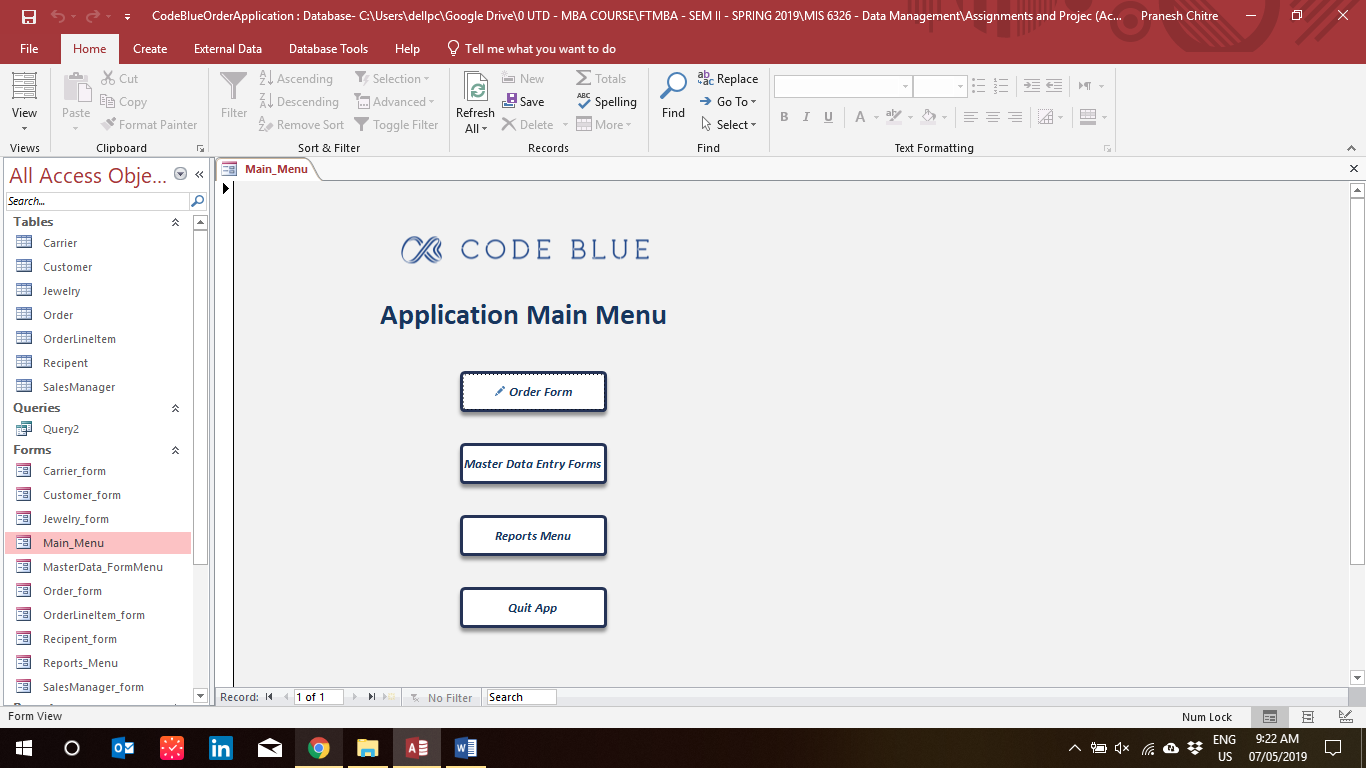
**Customer Account Report**

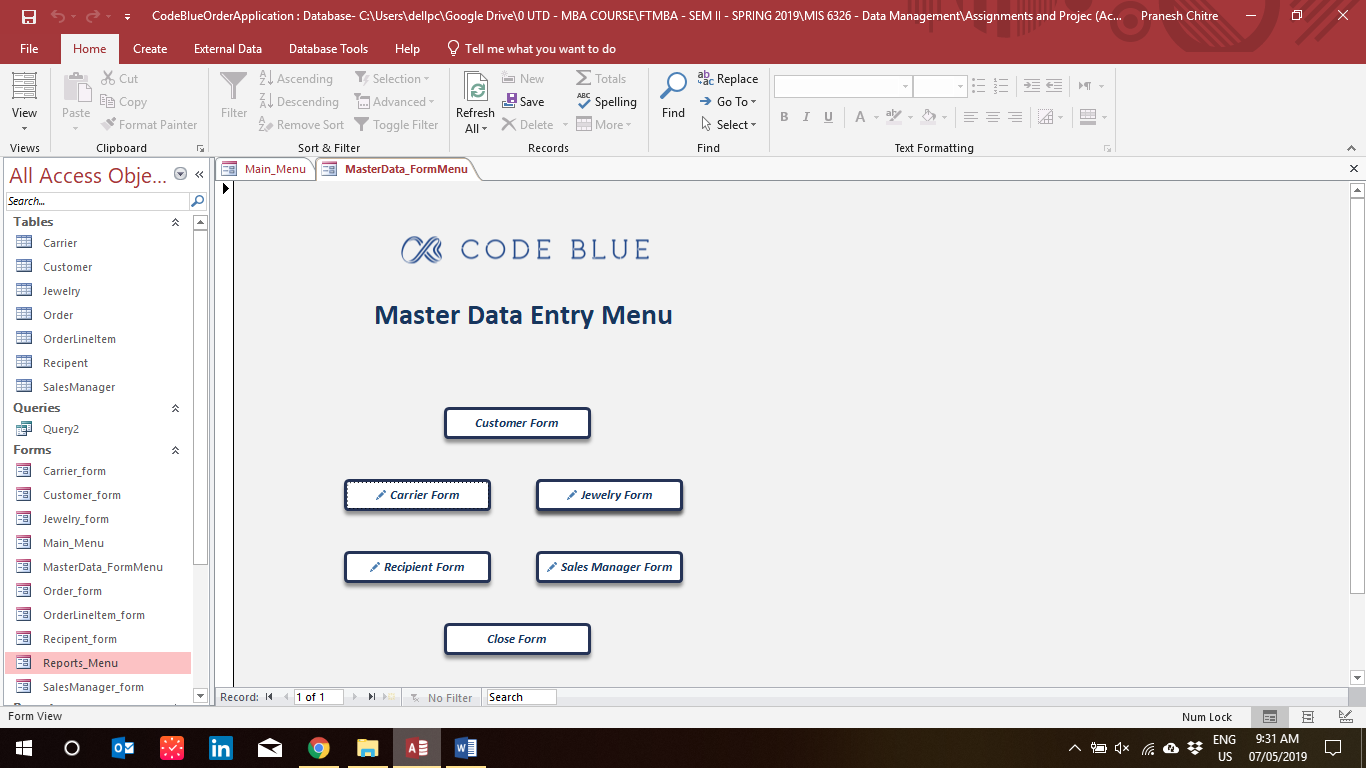


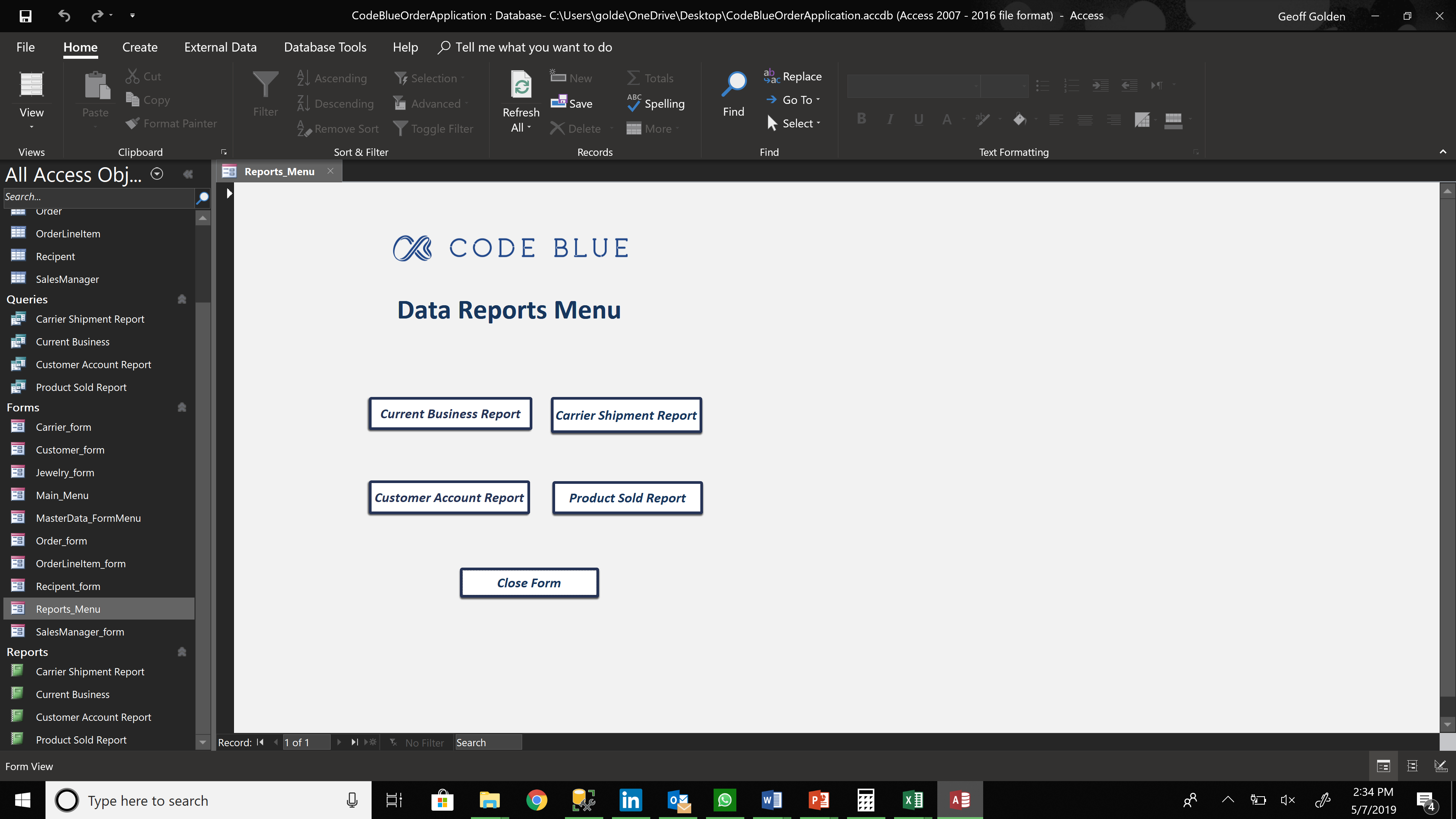
**Product Sold Report**



**Part 7. Menu Printouts**







**Part 8. Contribution Statement**

* Mohit Agarwal
  + - * + Data Entry into Tables
        + Documentation and Formatting
* Atharva Bendre
  + - * + Database Reports Creation
        + Report Formatting
* Revati Dalvi
  + - * + Entry Forms Creation
* Pranesh Chitre
  + - * + Menu Driven Environment Creation
        + Report Compilation
* Geoffrey Golden
  + - * + Background and Tables Description
        + ERD Creation
        + Relational Database Creation
* Krish Mehta
  + - * + Database Table Creation