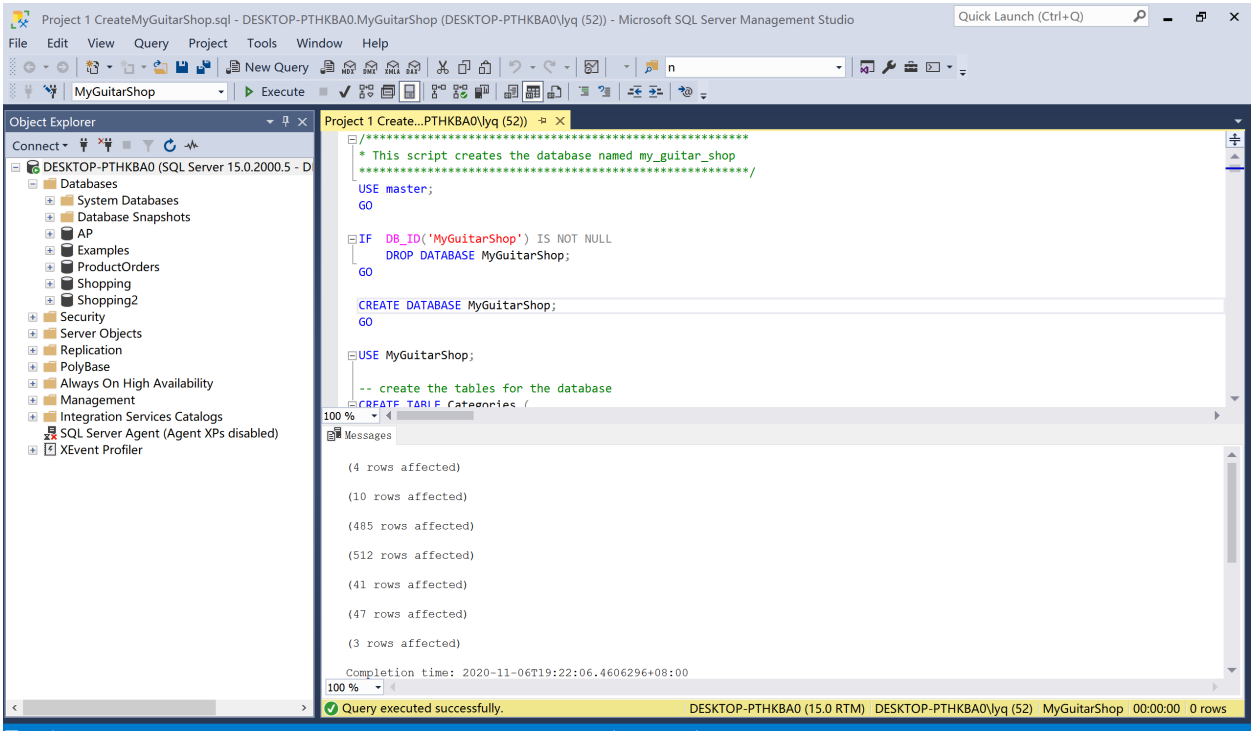
Project I

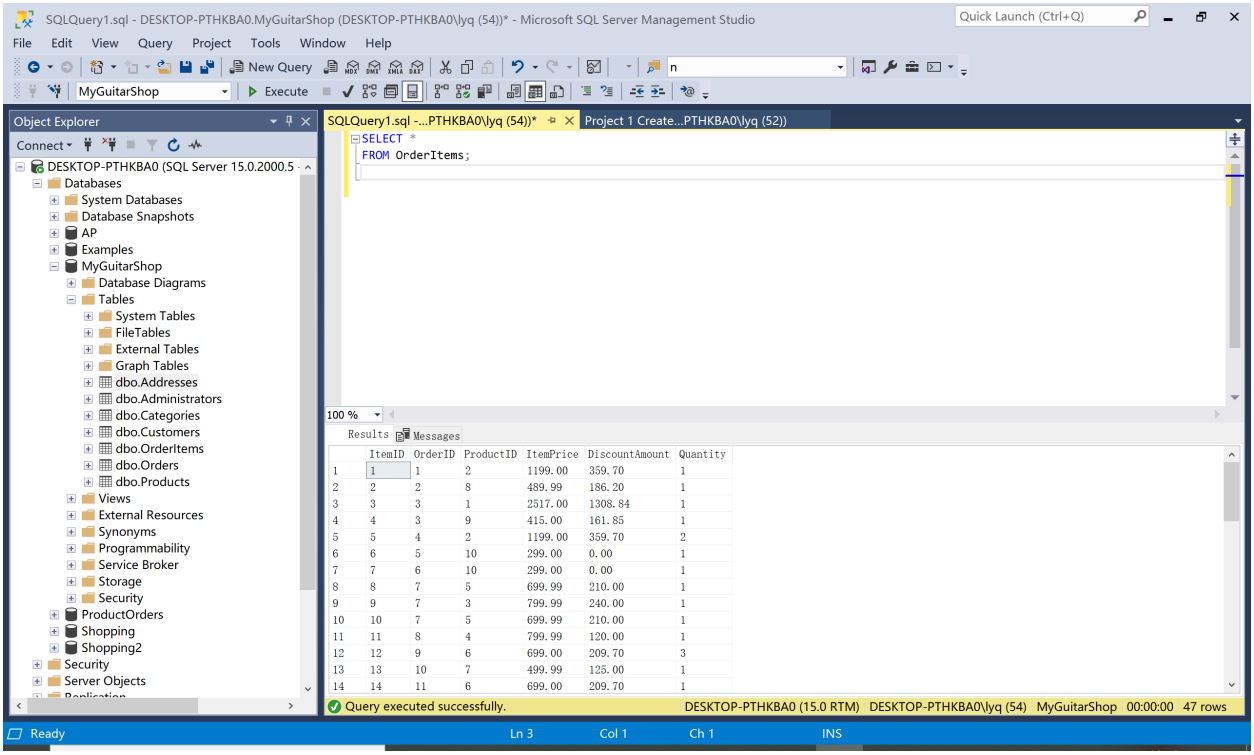
In part I, you will use SQL Server Management Studio to create the MyGuitarShop database, to review the tables in the MyGuitarShop database, and to enter SQL statements and run them against this database.

A. Database Setup [2 pts.]

1. (1) Download CreateMyGuitarShop.sql from Project 1 directory on Blackboard and open it in SQL server management studio. Execute the entire script and show the message in the Message tab, indicating the script is executed successfully. A complete screenshot of execution result is required. Your screenshot should show your entire SQL server window. You are not allowed to crop out any part and follow this for all the questions in this project.



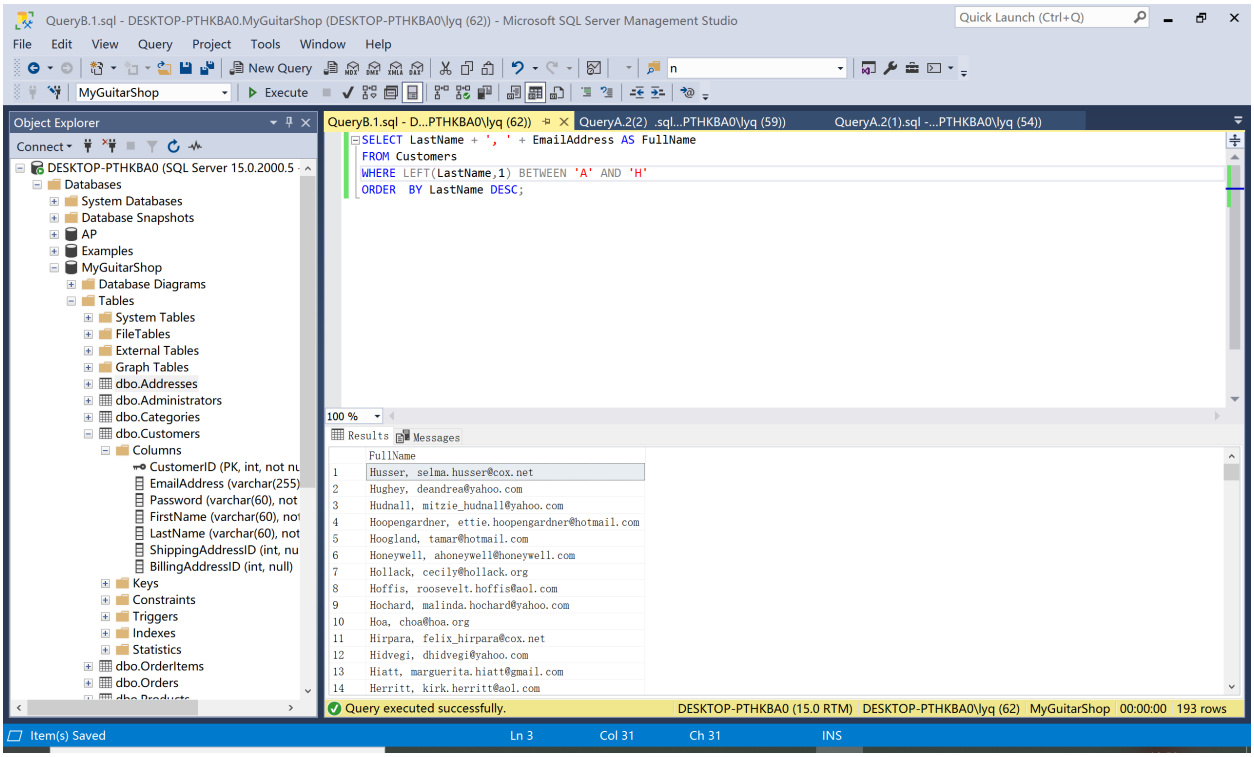
2. (2) Navigate through the database objects and view the column definitions for each table. Open a new Query Editor window. Show details in Orders table and OrdersItems table using SELECT statement.Full screenshots of execution results are required as mentioned.



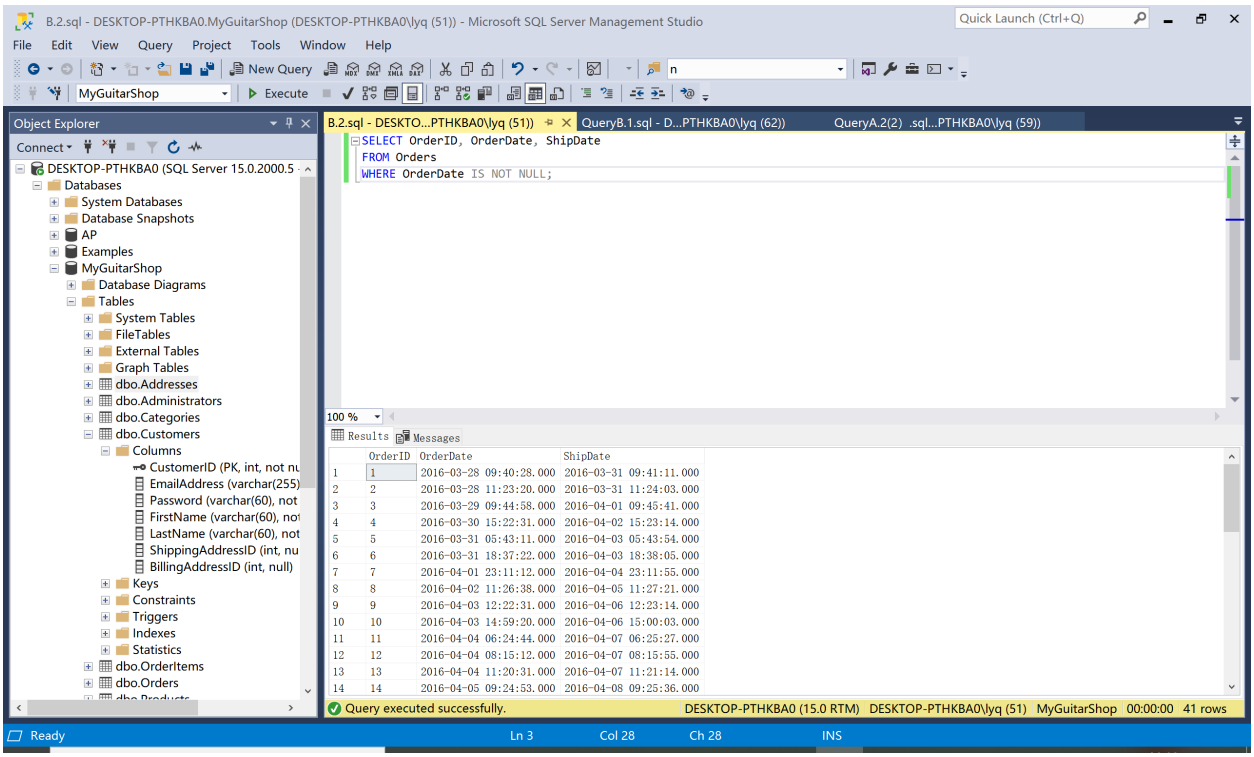
B. An Introduction to SQL [6 pts.]

1. [3] Write a SELECT statement that returns one column from the Customers table named FullName that joins the LastName and EmailAddress columns. Format this column with the last name, a comma, a space, and then Email Address like this: Abdallah, johnetta_abdallah@aol.com

Add an ORDER BY clause to this statement that sorts the result set by last name in descending sequence. Return only the contacts whose last name begins with a letter from A to H.

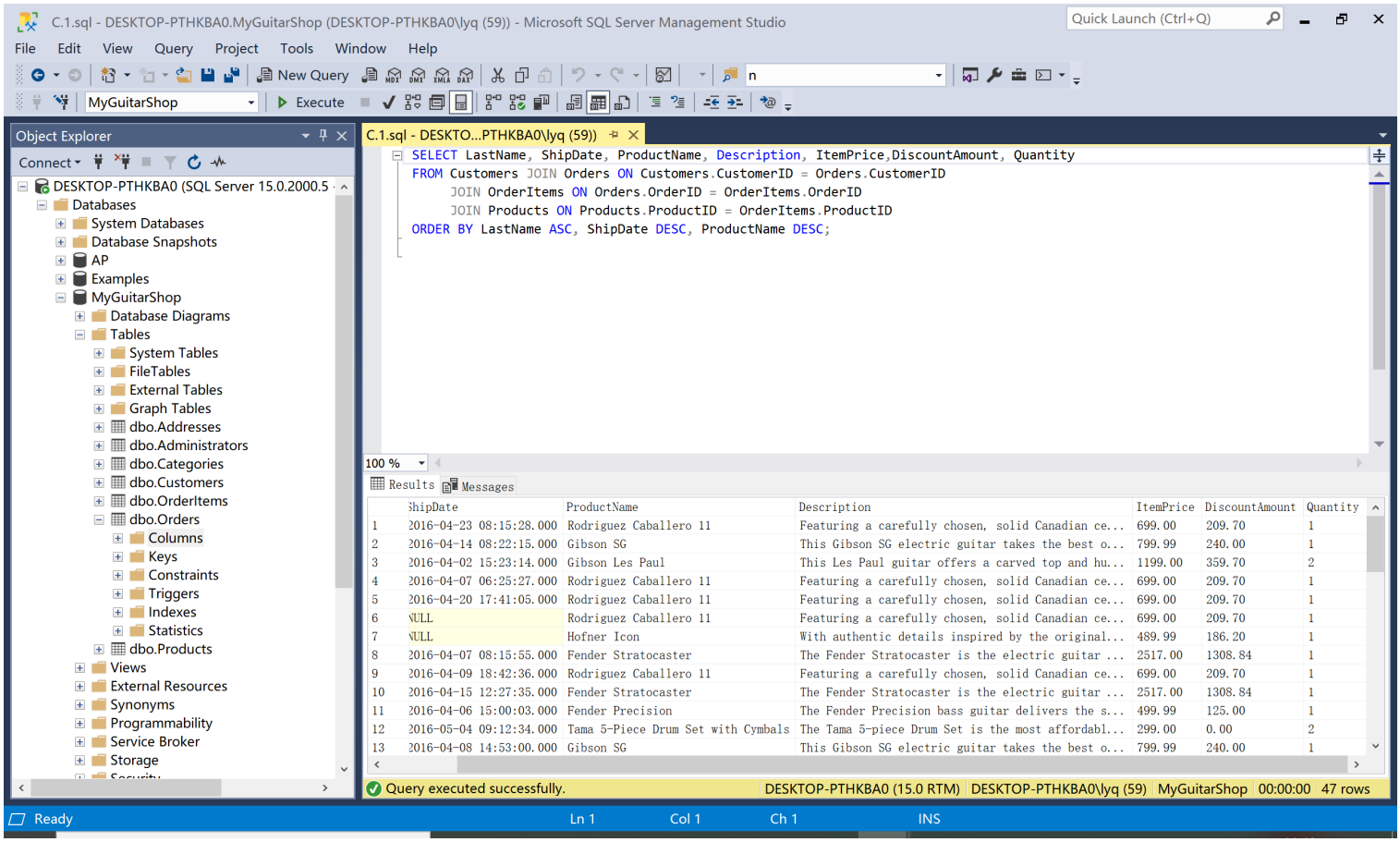


2. [3] Write a SELECT statement that returns these columns from the Orders table:
My Guitar Shop Database
OrderID OrderDate ShipDate
The OrderID column The OrderDate column The ShipDate column
Return only the rows where the OrderDate column does not contain a null value.

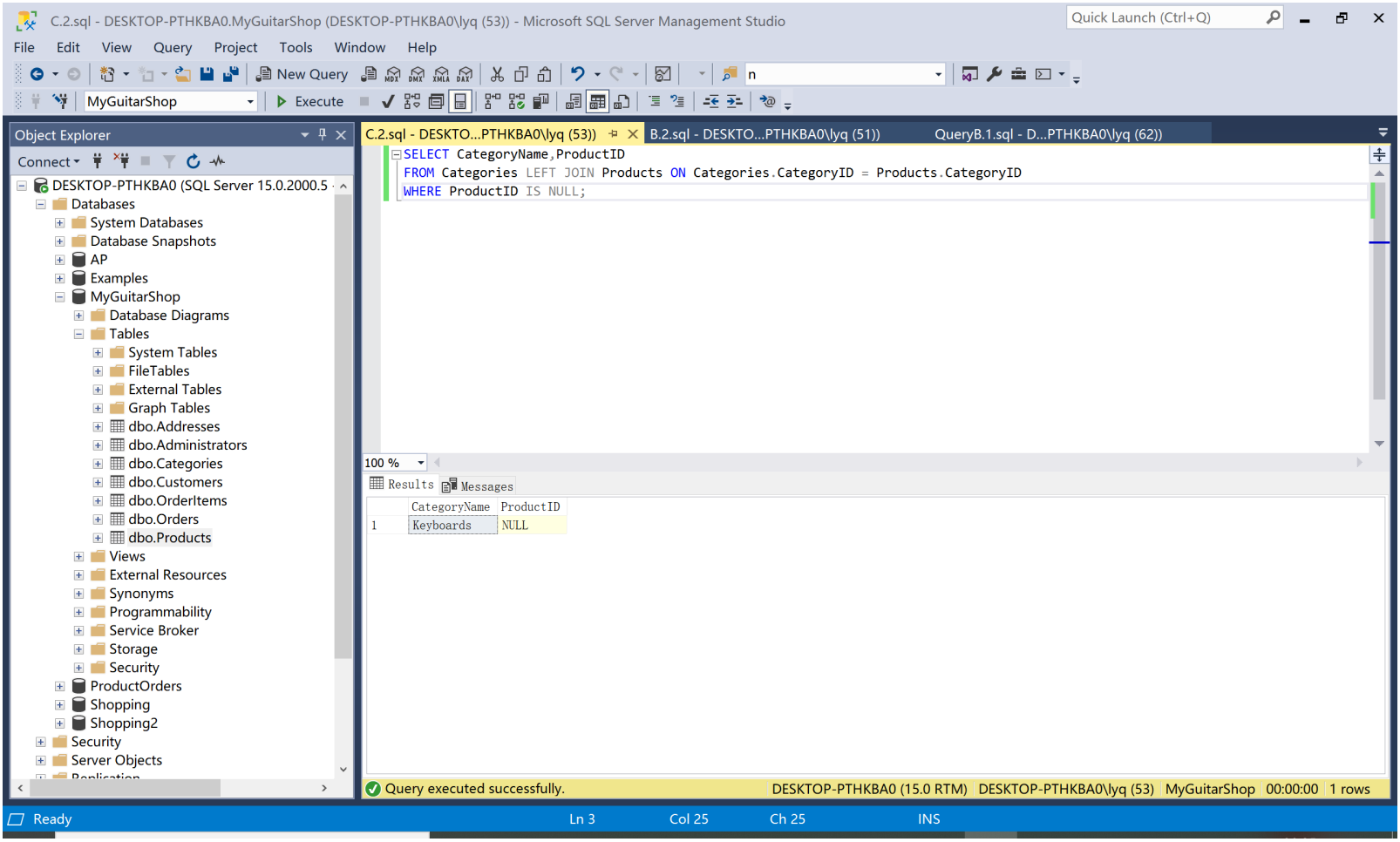


C. The essential SQL skills [44 pts.]

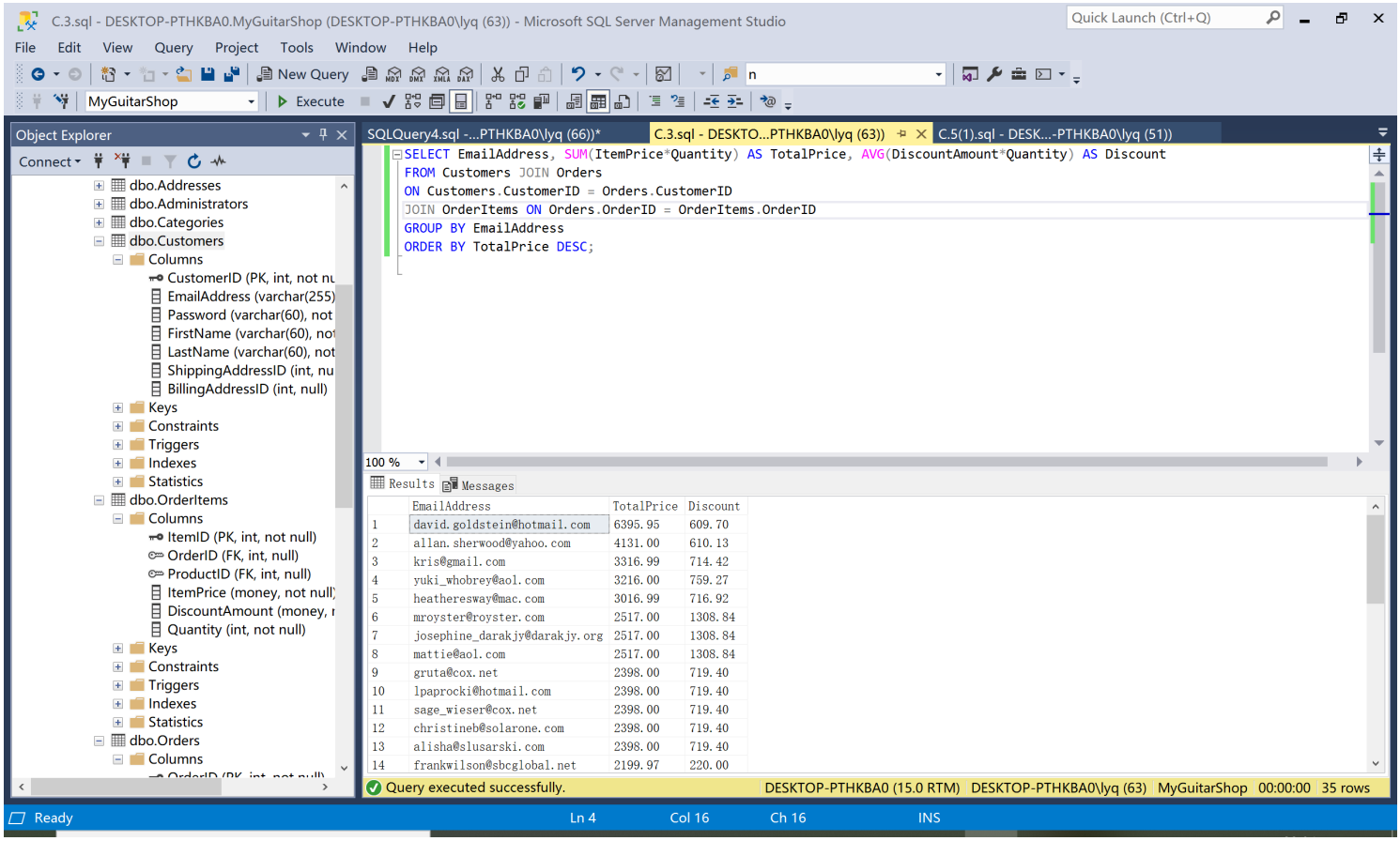
1. **[4] Write a SELECT statement that joins the Customers, Orders, OrderItems, and Products tables. This statement should return these columns: LastName, ShipDate, ProductName, Description, ItemPrice, DiscountAmount, and Quantity. Use aliases for the tables. Sort the final result set by LastName in ascending order, and in descending order for ShipDate, and ProductName.**



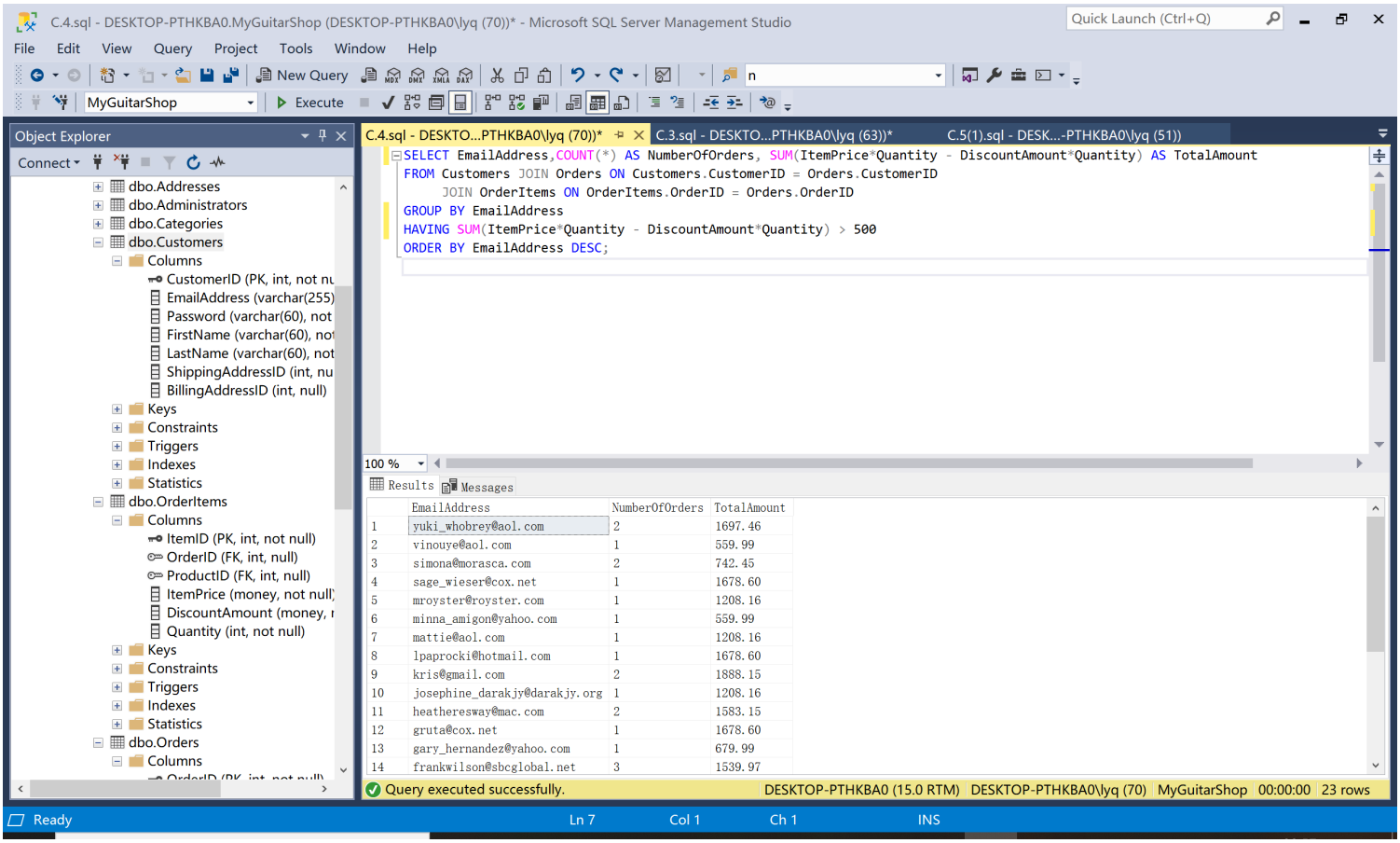
2. [4] Write a SELECT statement that returns CategoryName column from the Categories table. Return one row for each category that has never been used. (Hint: Use an outer join and only return rows where the ProductID column contains a null value.)



3. [4] Write a SELECT statement that returns one row for each customer that has orders with these columns:
a) The EmailAddress column from the Customers table
b) The sum of the ItemPrice in the OrderItems table multiplied by the quantity in the OrderItems table
c) The average of the DiscountAmount column in the OrderItems table multiplied by the quantity in the OrderItems table.
Sort the result set in descending sequence by the item price total for each customer.

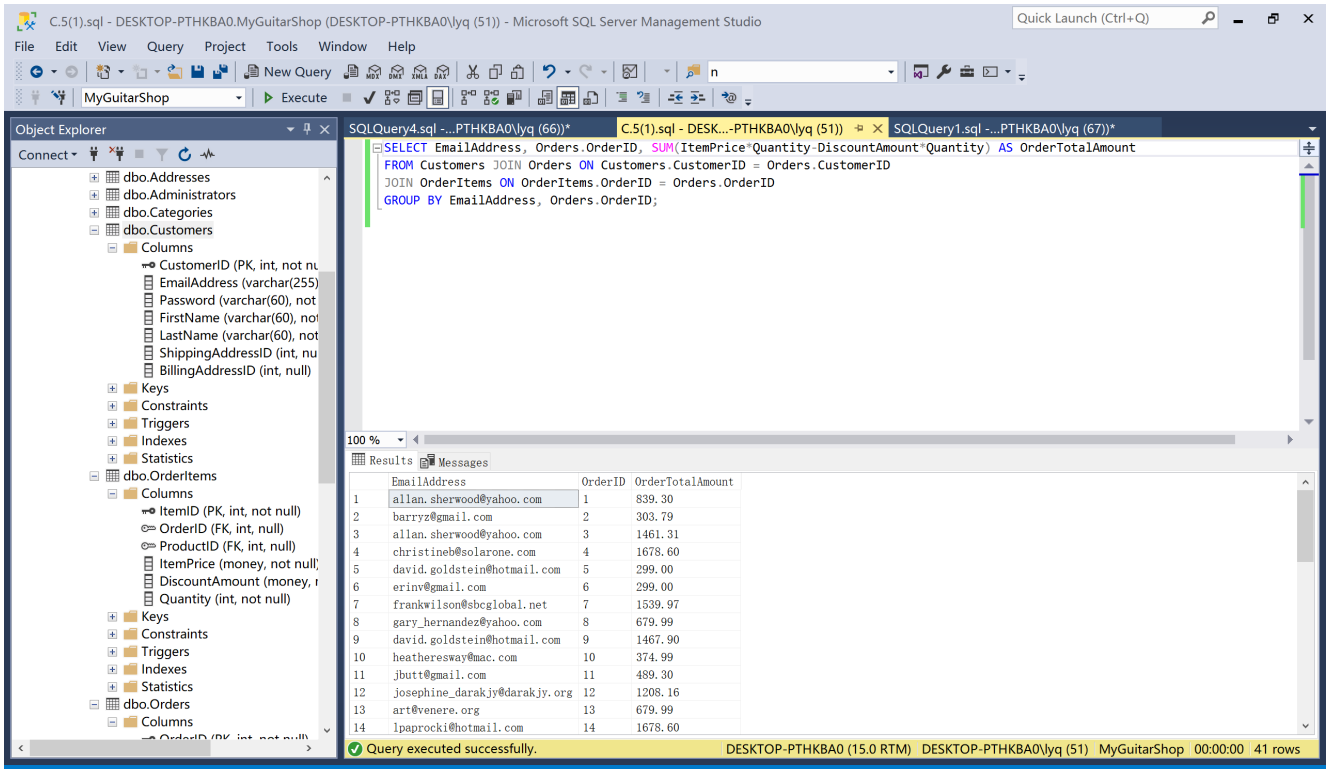


4. [4] Write a SELECT statement that returns one row for each customer that has orders with these columns:
a) The EmailAddress column from the Customers table
b) A count of the number of orders
c) The total amount for each order (Hint: First, subtract the discount amount from the price. Then, multiply by the quantity)
Return only those rows where items have a more than 500 ItemPrice value. Sort the result set in descending order of EmailAddress column.

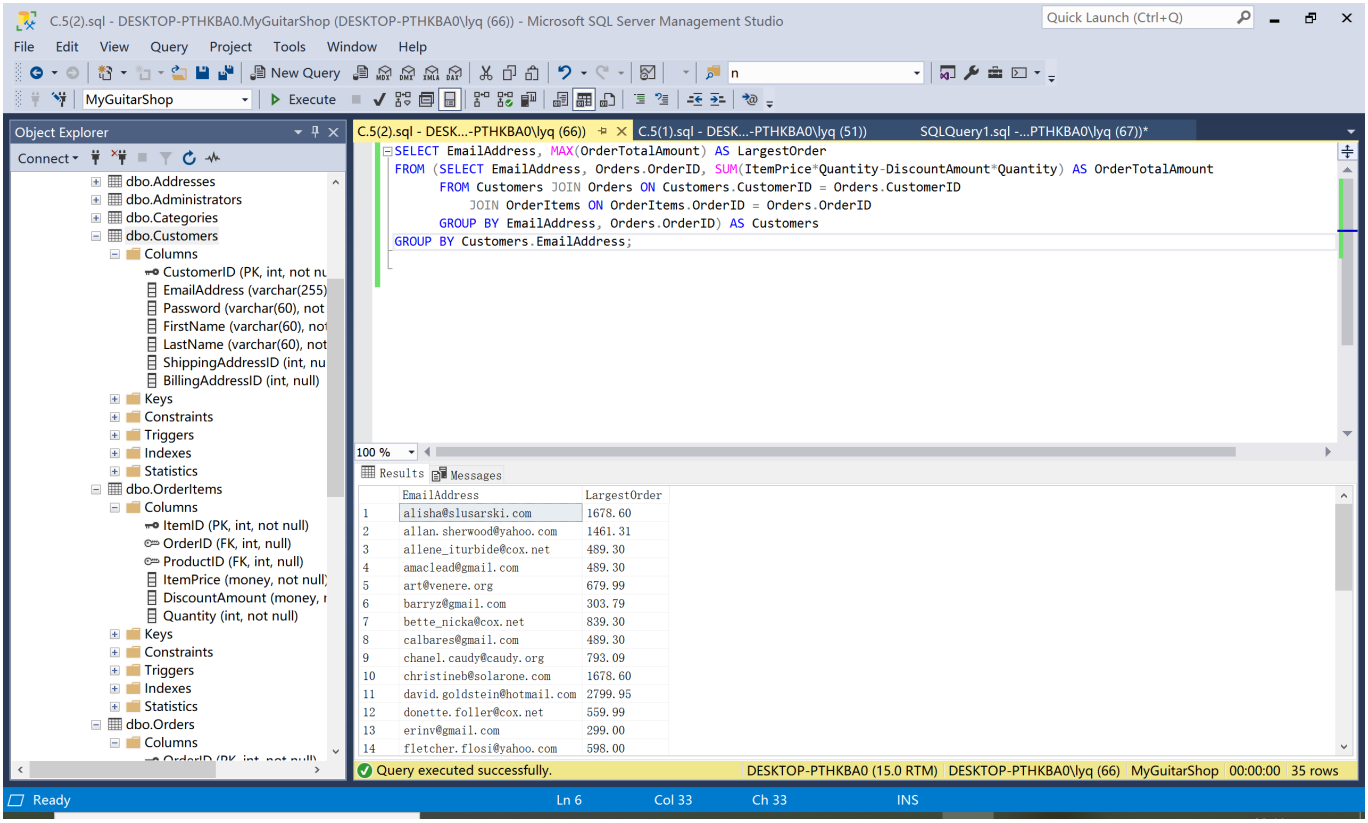


5. [4] (1) Write a SELECT statement that returns three columns: EmailAddress, OrderID, and the order total amount for each customer. To do this, you can group the result set by the EmailAddress and OrderID columns. In addition, you must calculate the order total amount from the columns in the OrderItems table.

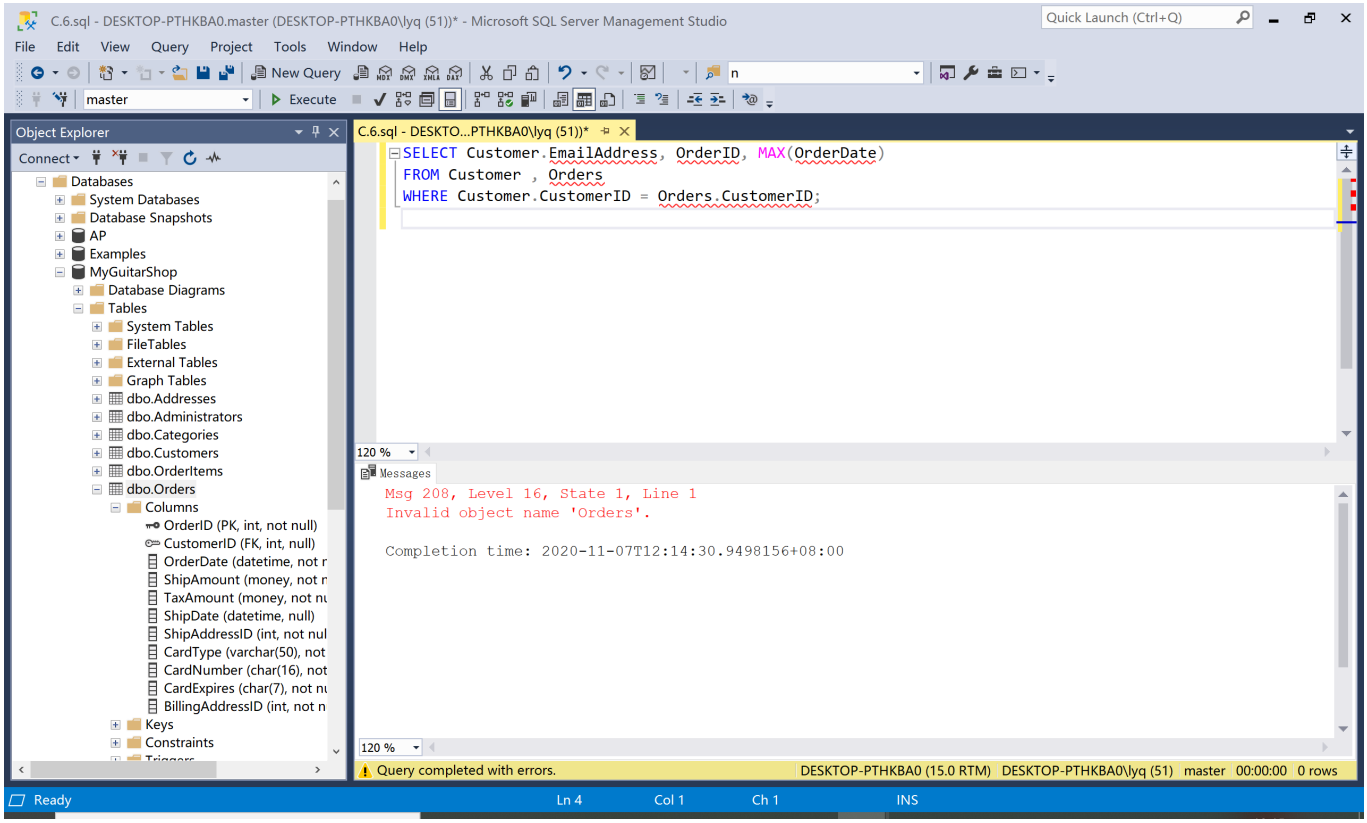
(1)



5(2) Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return two columns: the customer's email address and the largest order for that customer. To do this, you can group the result set by the EmailAddress column.

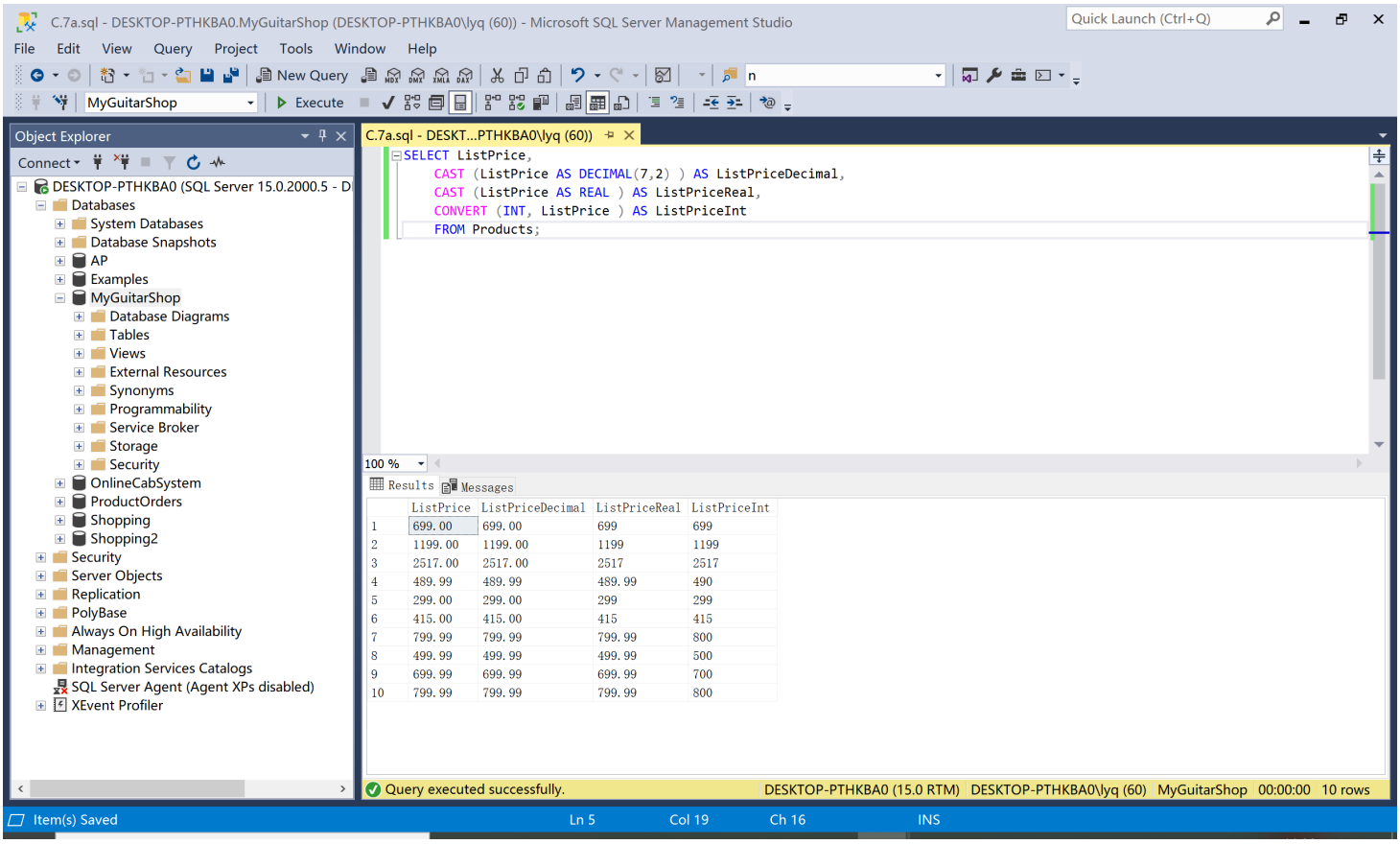


6. [4] Use a correlated subquery to return one row per customer, representing the customer's newest order (the one with the latest date). Each row should include these three columns: EmailAddress, OrderID, and OrderDate.

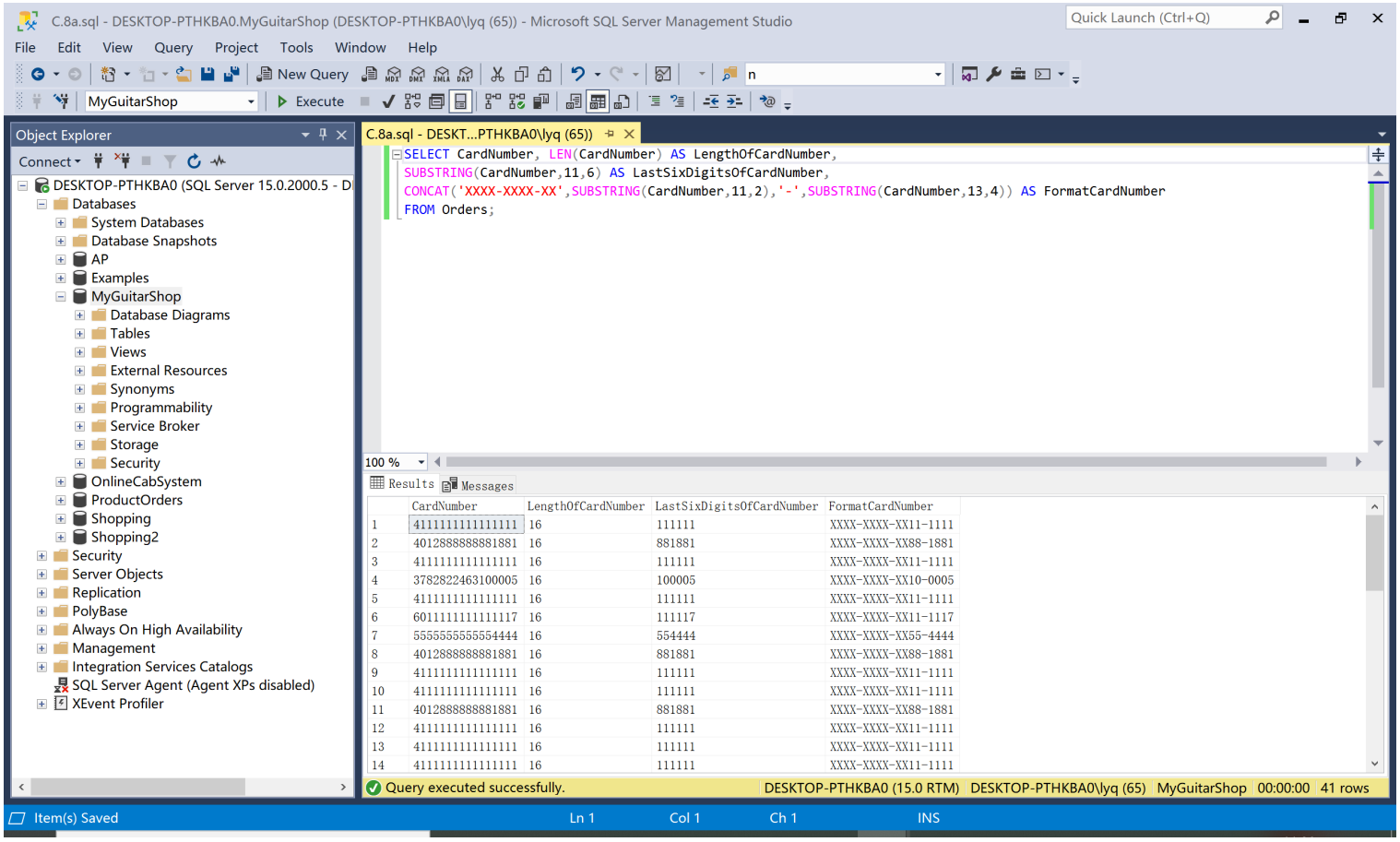


Sorry, I think I got the wrong syntax and I have tried to figure out but it didn't work.

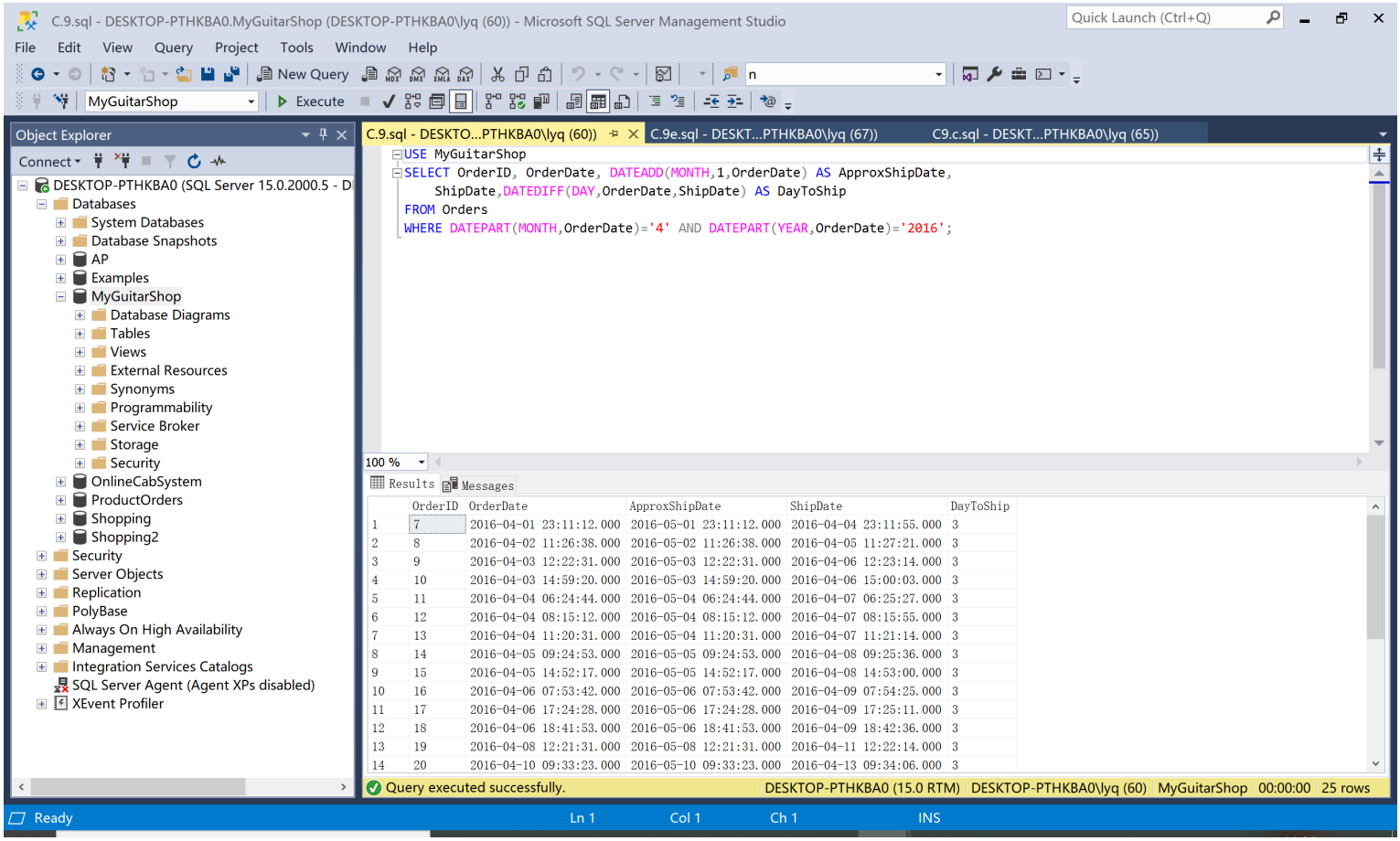
7. [4] Write a SELECT statement that returns these columns from the Products table:
a) The ListPrice column
b) b) A column that uses the CAST function to return the ListPrice column with 2 digits to
the right of the decimal point
c) A column that uses the CAST function to return the ListPrice column as a real number
d) A column that uses the CONVERT function to return the ListPrice column as an integer.



8. [4] Write a SELECT statement that returns these columns from the Orders table:
a) The CardNumber column
b) The length of the CardNumber column
c) The last six digits of the CardNumber column
d) A column that displays the last four digits of the CardNumber column in this format: XXXX-XXXX-XX12-3456. In other words, use X's for the first 10 digits of the card number and actual numbers for the last six digits of the number and include dash symbols as specified in the format.

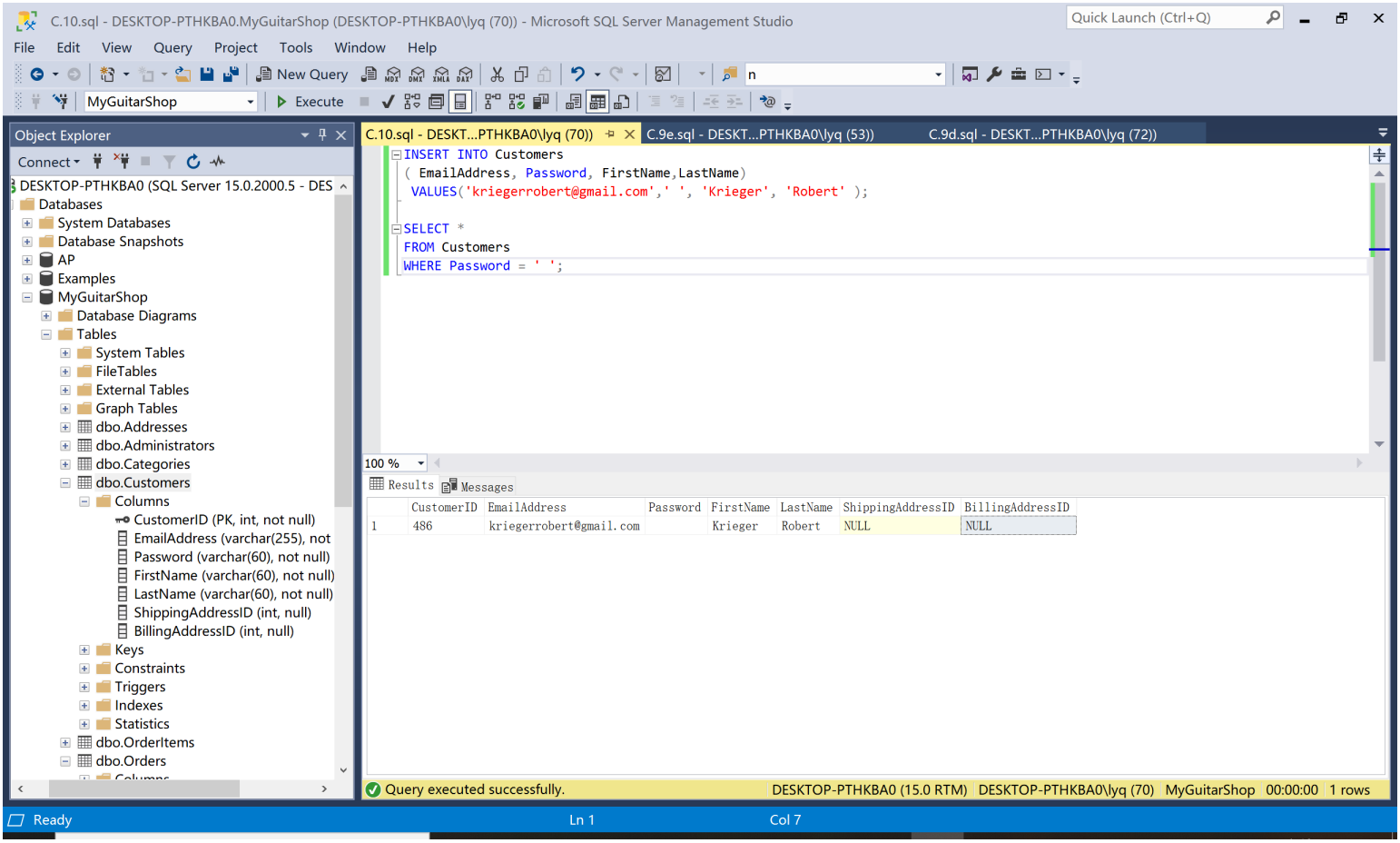


9. [4] Write a SELECT statement that returns these columns from the Orders table:
a) The OrderID column
b) The OrderDate column
c) A column named ApproxShipDate that's calculated by adding 1 month to the OrderDate column
d) The ShipDate column
e) A column named DaysToShip that shows the number of days between the order date
and the ship date
When you have this working, add a WHERE clause that retrieves just the orders for April 2016.



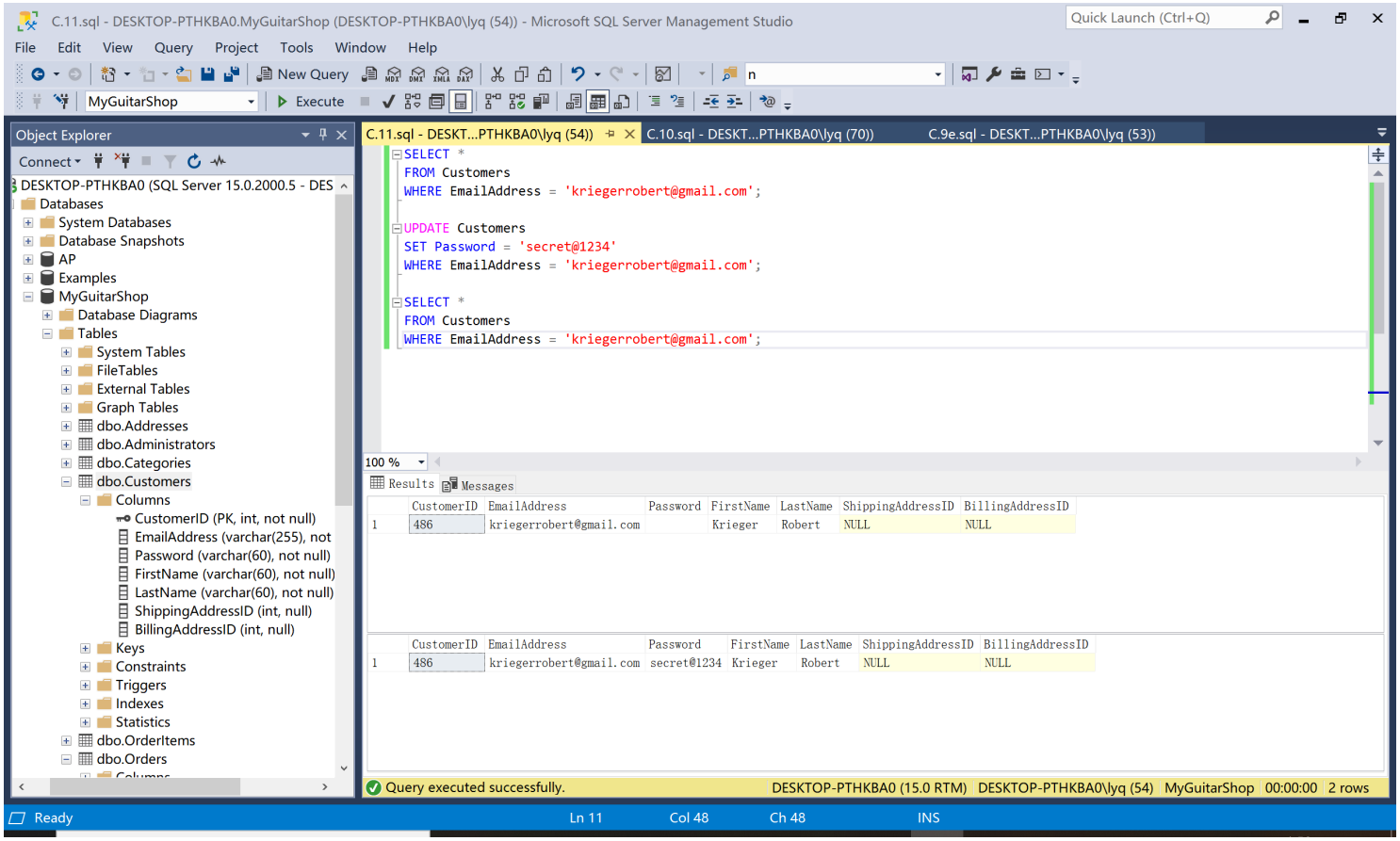
For question 10-11: To test whether a table has been modified correctly as you do these questions, please write and run an appropriate SELECT statement and take full screenshots of the verification without cropping any part of you SQL server window.
10. [4] Write an INSERT statement that adds this row to the Customers table:
EmailAddress: kriegerrobert@gmail.com
Password: FirstName: LastName:
(empty string) Krieger Robert
Use a column list for this statement..

Here are my test and we could found Krieger was inserted.



11. [4] Write an UPDATE statement that modifies the Customers table. Change the password column to "secret@1234" for the customer with an email address: kriegerrobert@gmail.com.

After the test, we could see as below that the password has been updated to" secret21234".



D. Advanced SQL skills (views/stores procedures/ functions / scripts) [24 pts.]

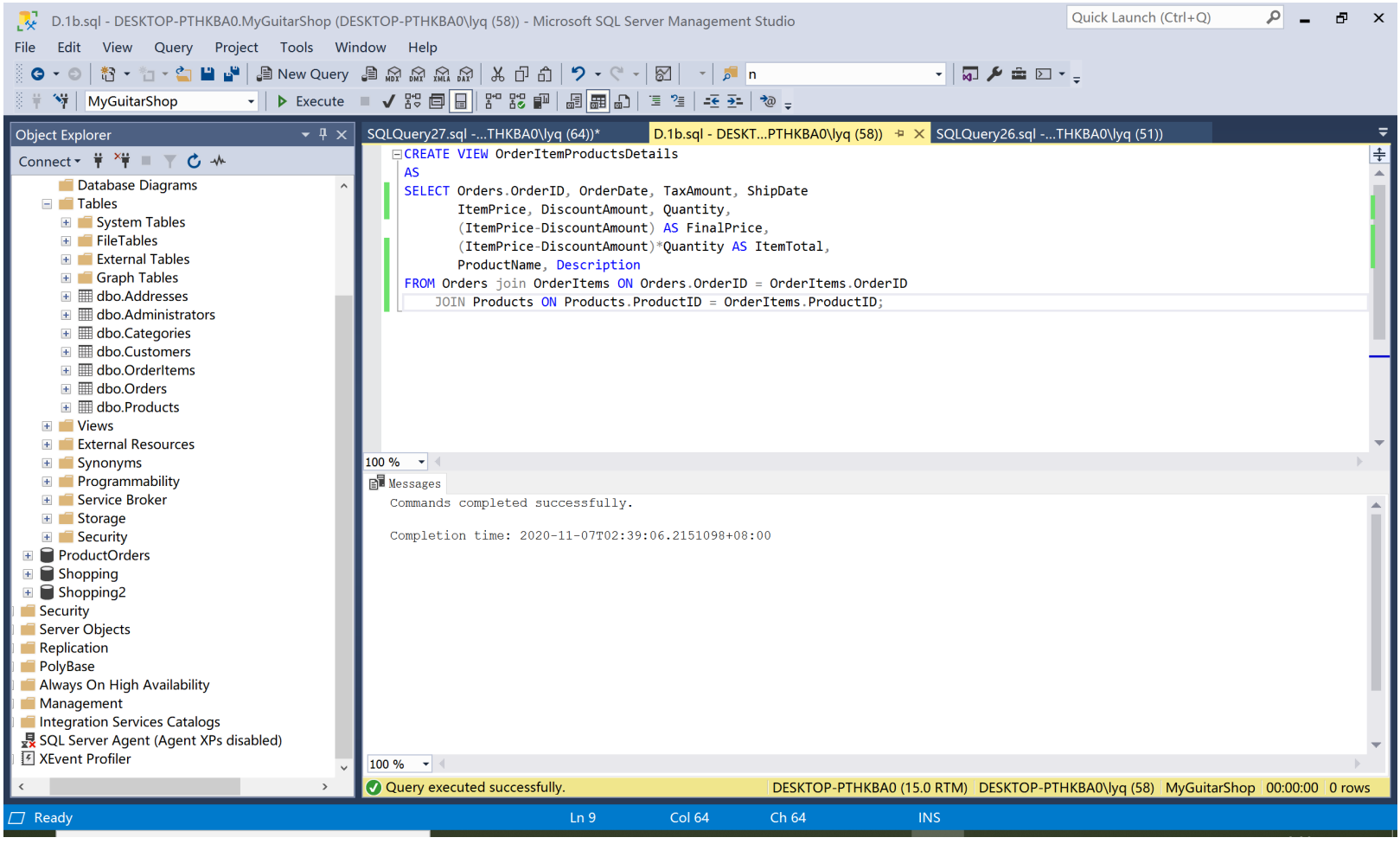
Open the script named CreateMyGuitarShop.sql and run this script again. That should restore the data that's in the database. Then complete questions in Section D. Screenshot of execution is required for each question. Please also use SELECT statement to verify results of your codes (Full Screenshots of verification are required).

1. [4] Create a view named OrderItemProductsDetails that returns columns from the Orders, OrderItems, and Products tables.

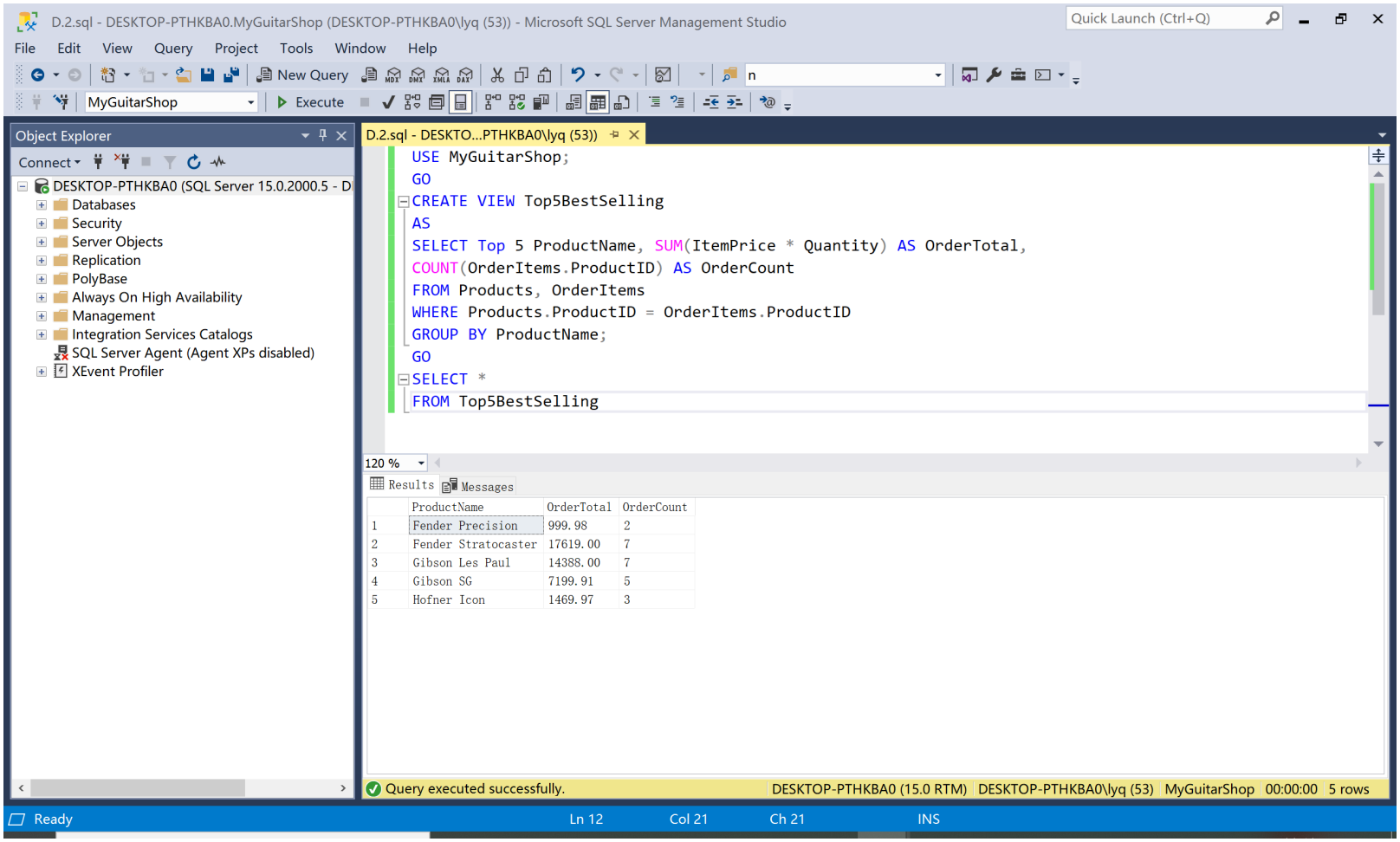
a) This view should return these columns from the Orders table: OrderID, OrderDate, TaxAmount, and ShipDate.

b) This view should return these columns from the OrderItems table: ItemPrice, DiscountAmount, FinalPrice (the discount amount subtracted from the item price), Quantity, and ItemTotal (the calculated total for the item).

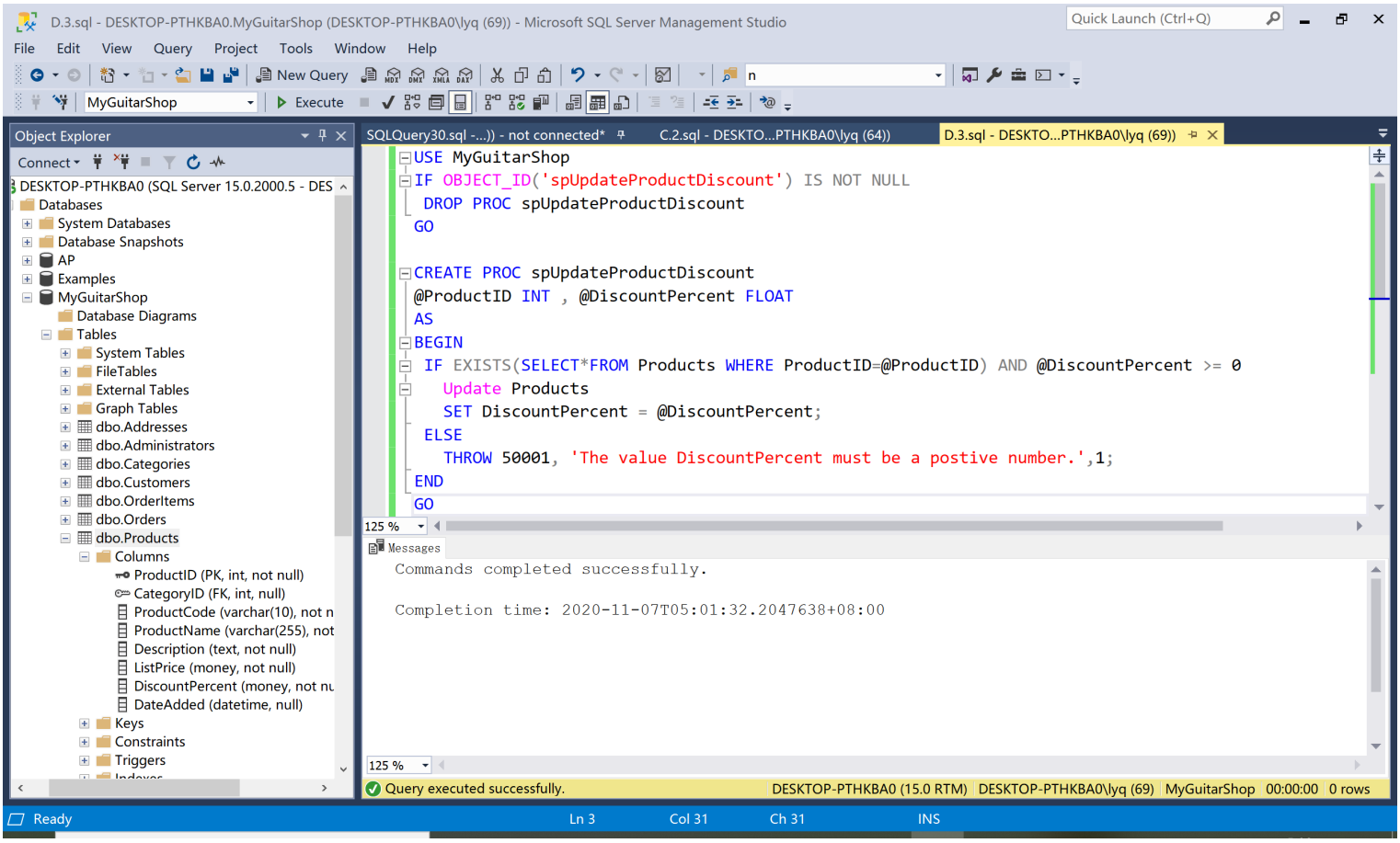
c) This view should return the ProductName and Description column from the Products table.



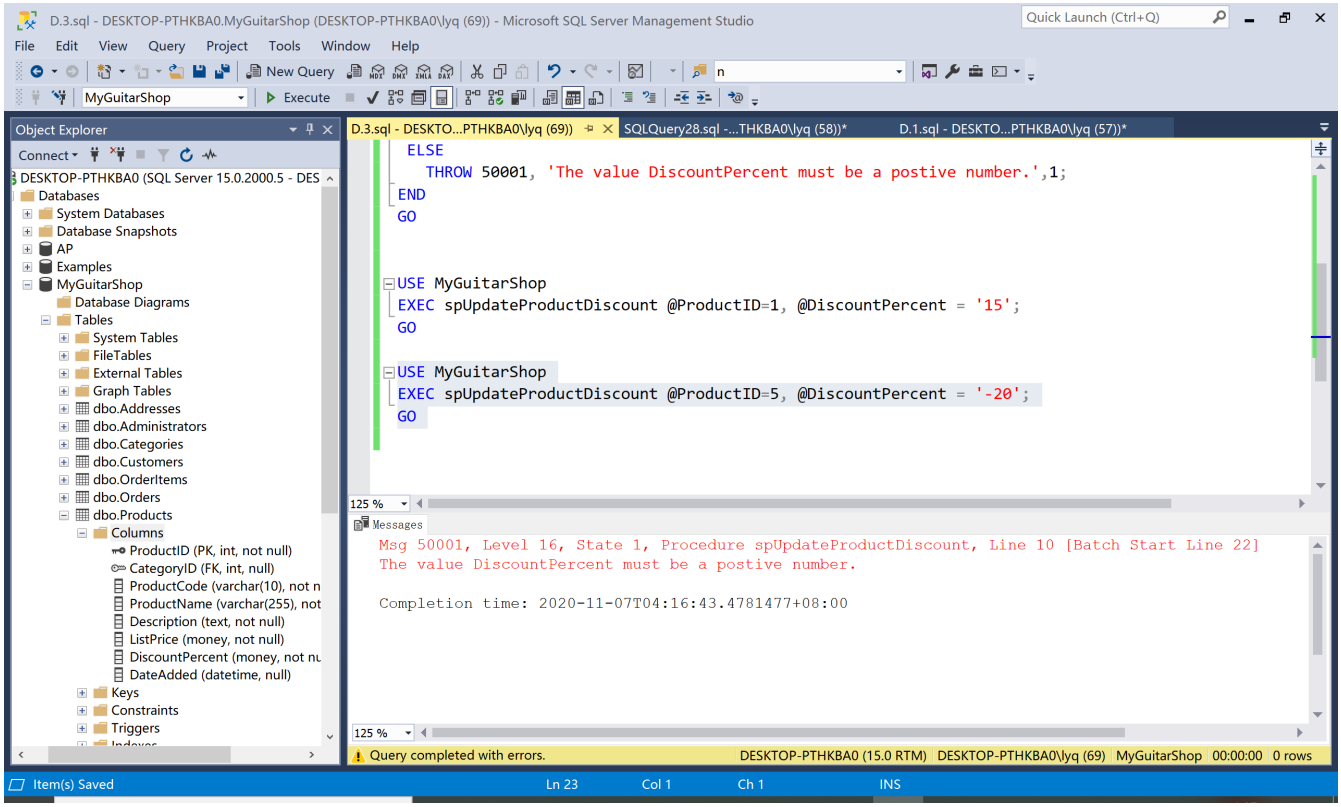
2. [5] Create a view named Top5BestSelling that uses the view you created in Section D Question 1. This view should return some summary information about five best selling products. Each row should include these columns: ProductName, OrderTotal (the total sales for the product) and OrderCount (the number of times the product has been ordered).

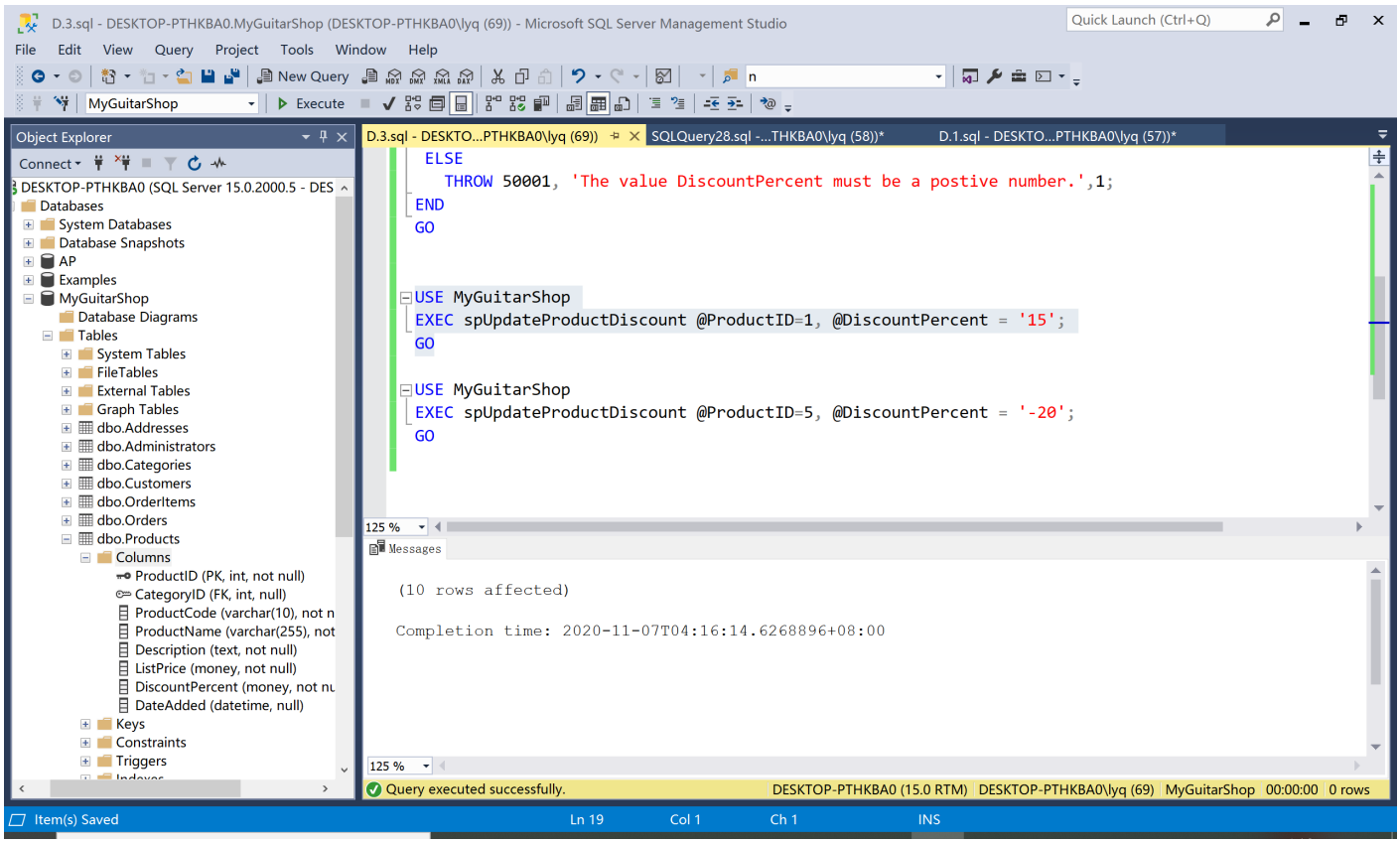


3. [5] Write a script that creates and calls a stored procedure named spUpdateProductDiscount that updates the DiscountPercent column in the Products table. This procedure should have one parameter for the product ID and another for the discount percent.
If the value for the DiscountPercent column is a negative number, the stored procedure should raise an error that indicates that the value for this column must be a positive number. Code at least two EXEC statements that test this procedure.



The EXEC statements are a below:





4. [5] Write a script that calculates the common factors between 15 and 30. To find a common factor, you can use the modulo operator (%) to check whether a number can be evenly divided into both numbers. Then, this script should print lines that display the common factors like this:

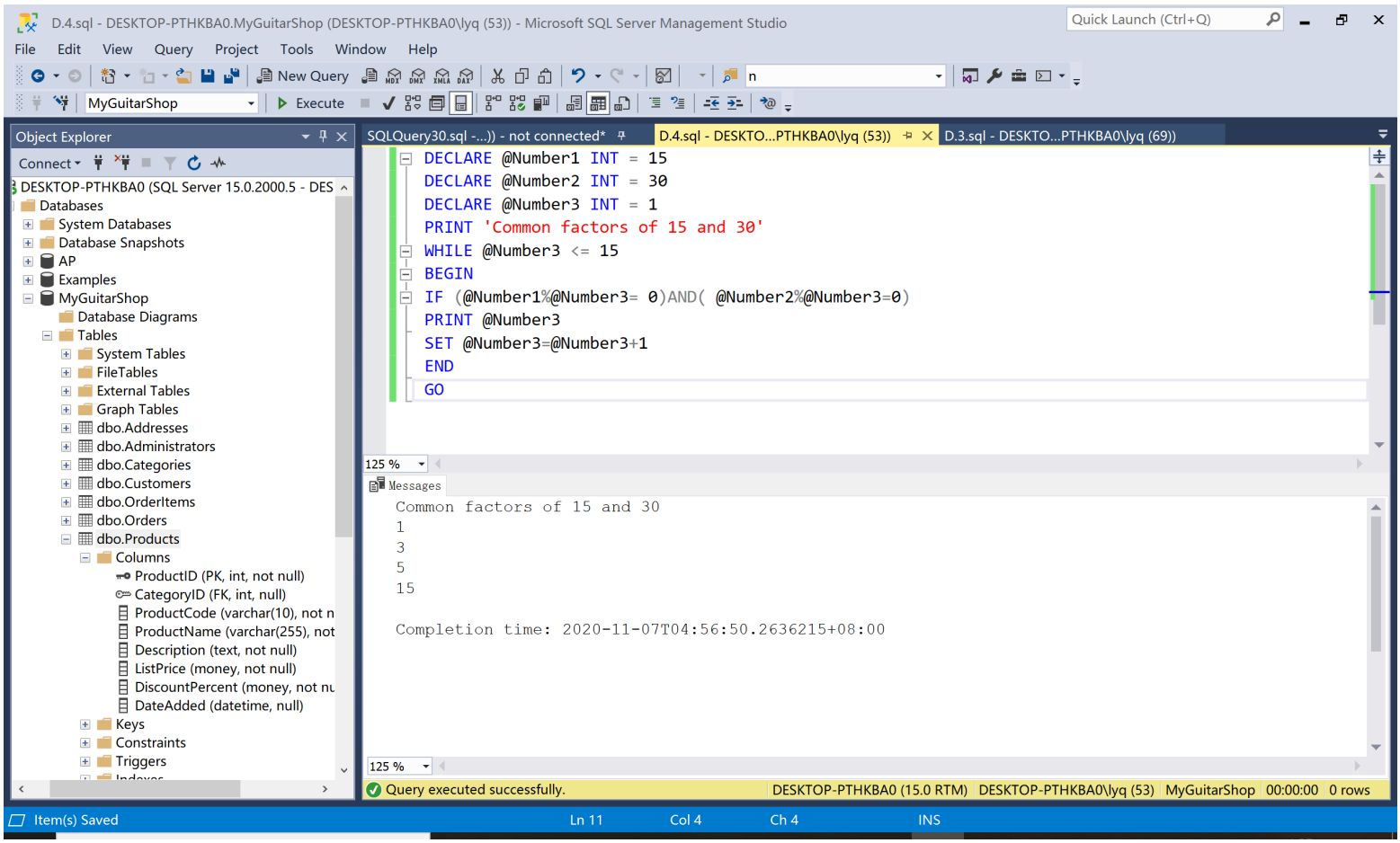
Common factors of 15 and 30

1

3

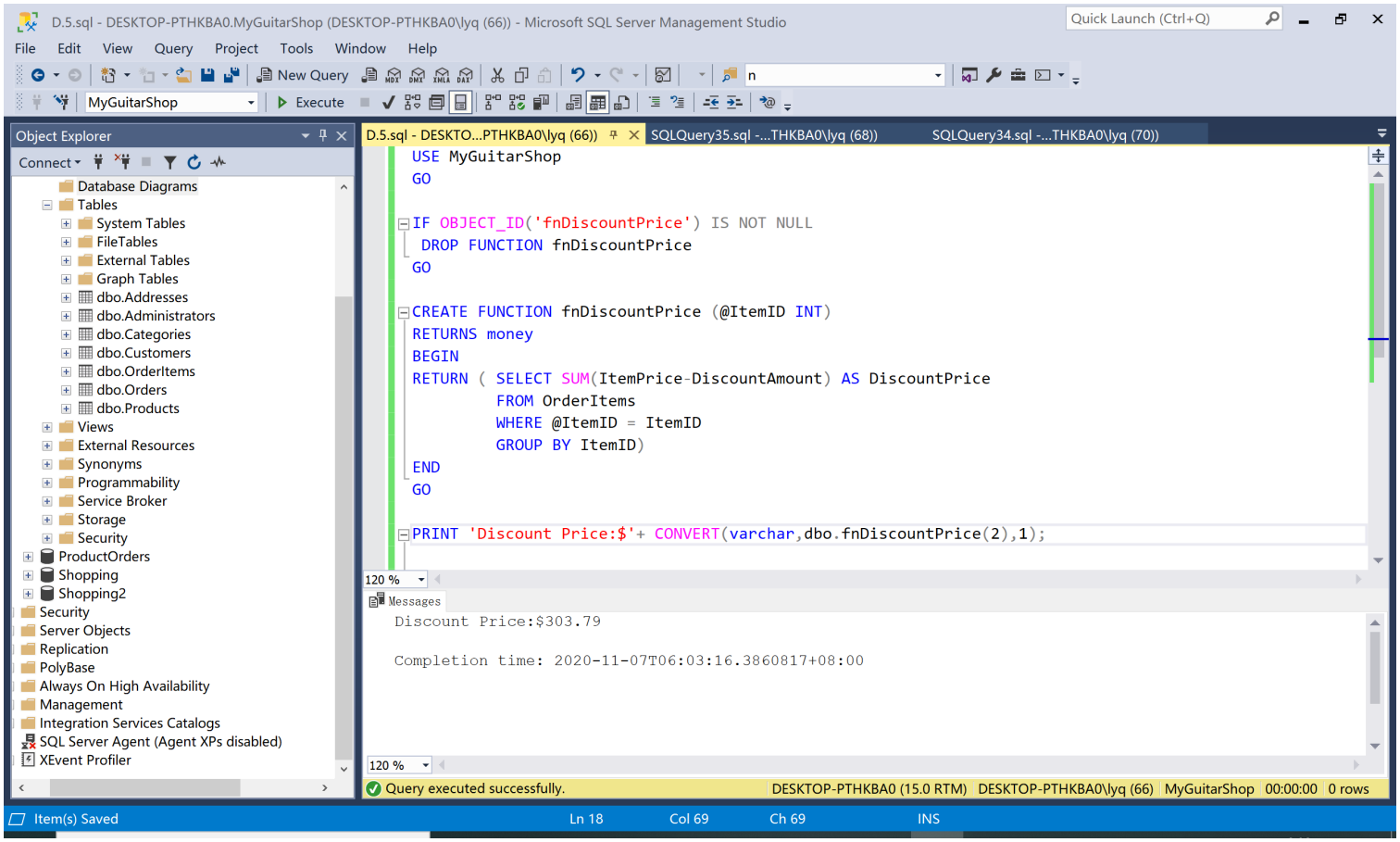
5

15

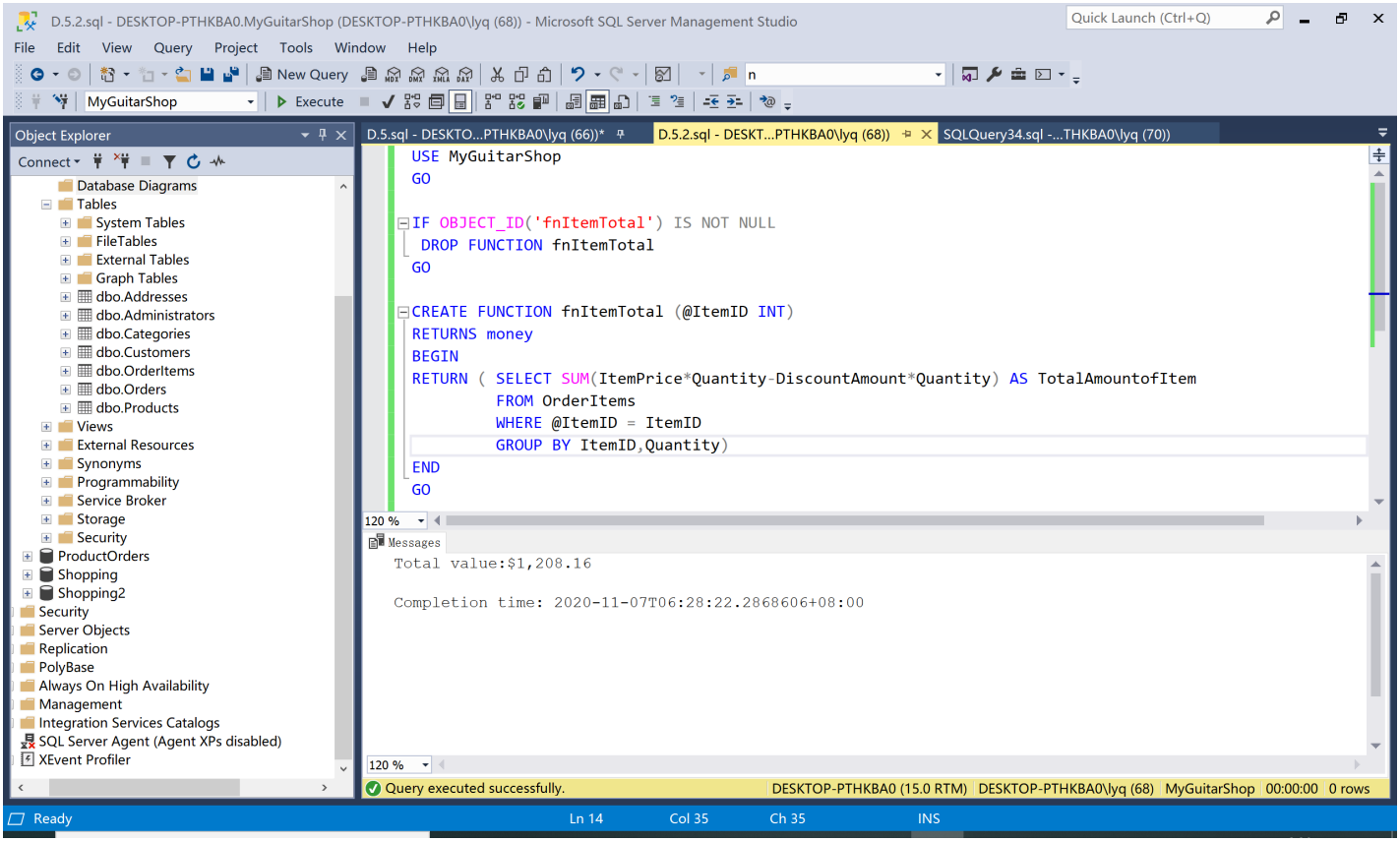


5.

[5] (1) Write a script that creates and calls a function named fnDiscountPrice that calculates the discount price of an item in the OrderItems table (discount amount subtracted from item price). To do that, this function should accept one parameter for the item ID, and it should return the value of the discount price for that item.



5.(2) Write a script that creates and calls a function named fnItemTotal that calculates the total amount of an item in the OrderItems table (discount price multiplied by quantity). To do that, this function should accept one parameter for the item ID, it should use the DiscountPrice function that you created in (1), and it should return the value of the total for that item.

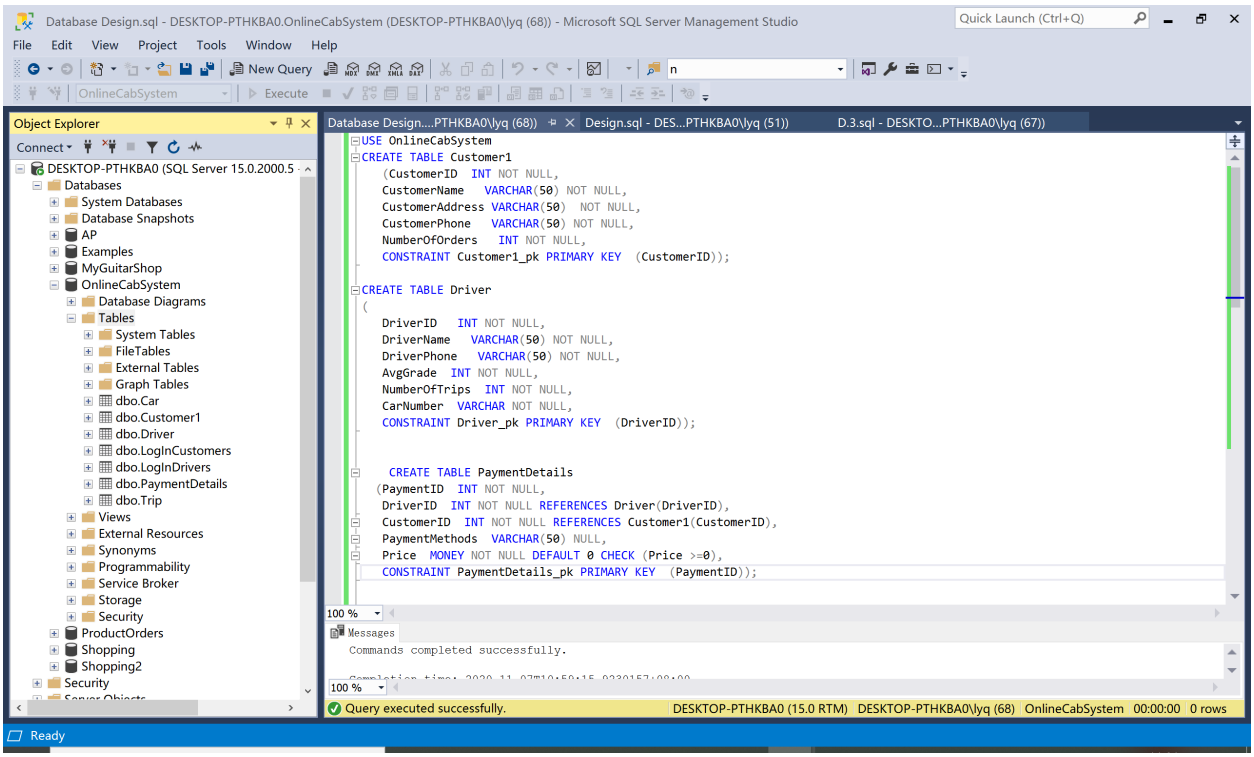


Database Design [20 pts.]

Create a sample database design for Online Cab System and draw one database model for it. It needs to keep track of Customer information, Driver information, login details of both customer and driver, Trip information, Payment details used by customers to pay for a trip, Car details, Rating information given for driver by customer and for customer given by driver and Insurance information(for both car and drivers if applicable). Your design could add more things to the existing requirements, but all the given requirements should be met.

1. **[3] Design the database that makes sense for the problem and select fields that make the most sense. A complete screenshot of your final design model is required.**

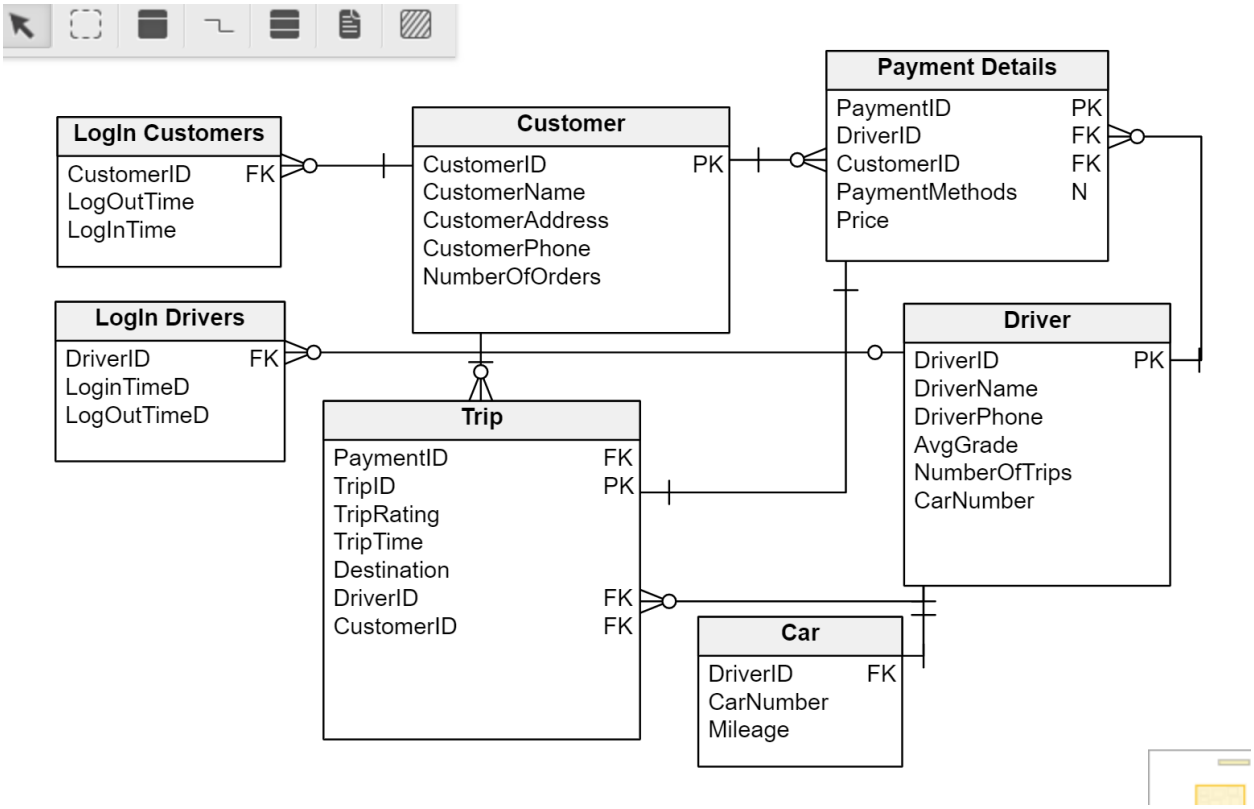
Here is my database design procedure and screenshot, As you could see I put 6 table in the new database for online cab system.



2. [10] Determine the tables, columns, primary keys, nullabilities and show relationships between tables (one -one/one-many/many-many).

The relationship between Customer table and Payment is one-many.
The relationship between Drivers table and Payment is one-many.
The relationship between Customer table and Trip is one-many.
The relationship between Car table and Driver is one-one.
The relationship between Customer table and LoginC is one-many.
The relationship between Driver table and LoginD is one-many.
The Primary keys and Foreign keys are as below.
And the nullabilities is Mileage.

3. [5] Normalize your design into 3rd Normal Form. Please use MS Visio, Vertabelo or any similar database design tool.



4. [2] Explain your design, including relationship between tables.

I just suspect one driver only have one car so the relationship between is one-one and I also think people with one account will log lots time during time.

My remarks : I think this course is a little bit hard to understand for me but I strongly think this course is very useful for us to deal with statistics by SQL, it help us gather the statistics and analyze them. I think it's interesting.

In your typed report (pdf or docx only), please include the following items:
1. Complete SQL source codes in typed format (not screenshots allowed for this).
2. Comments for SQL codes (or for Screenshots, if codes aren't to be used for a particular question)
3. Full screenshots of the requested actions or each question including the total number of rows in result set and your name. A complete screenshot of execution result is required. Your screenshot should show your entire SQL server window. You are not allowed to crop out any part and follow this for all the questions in this project
4. Your remarks on the project [4 pts.]. The remarks should be specific thoughts and references you learned that are related with knowledge points.
Submit your report on Blackboard.
5