**Name: Tharindu Rehan Fernando Ponnahennedige**

**ID:22530478**

**Course: INFO601 Data Modelling & SQL/2401**

**TASK 2**

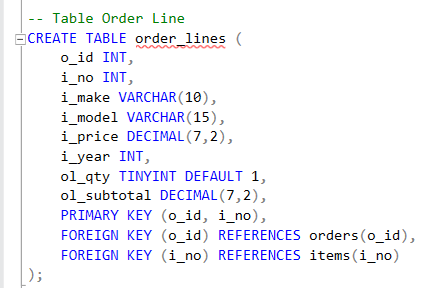
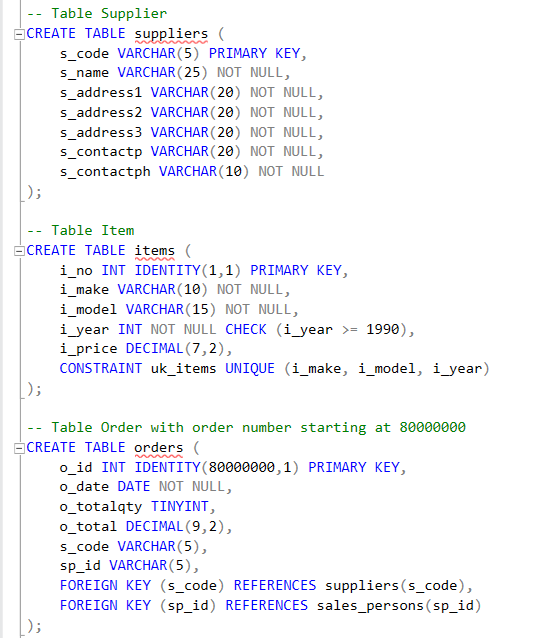
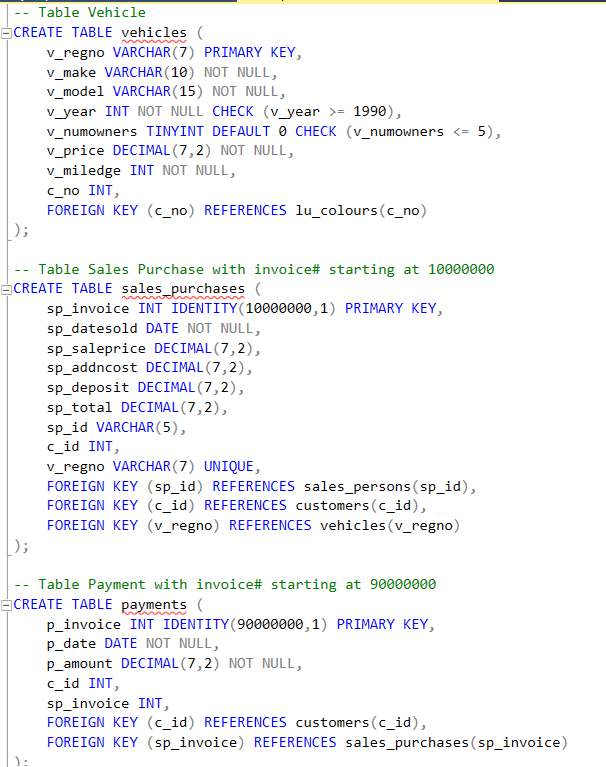
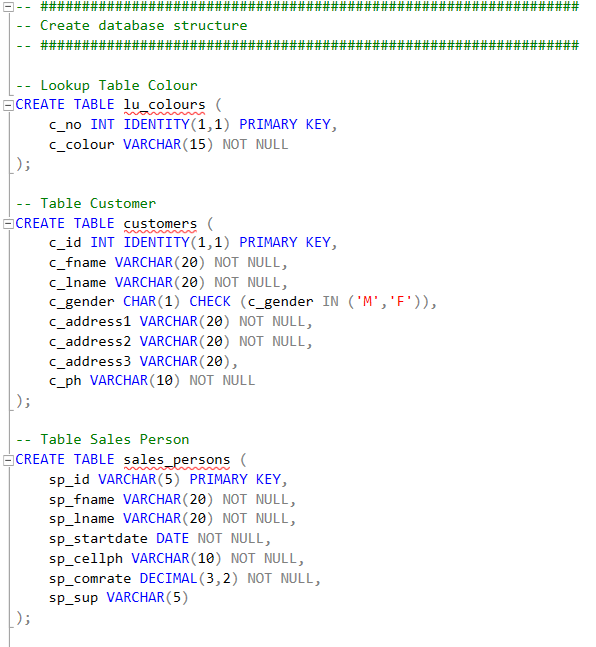
Your challenge is to create MSSQL Server scripts to perform the tasks, as outline below. The original Oracle scripts are included for you to use as a guide.

**PART A:**

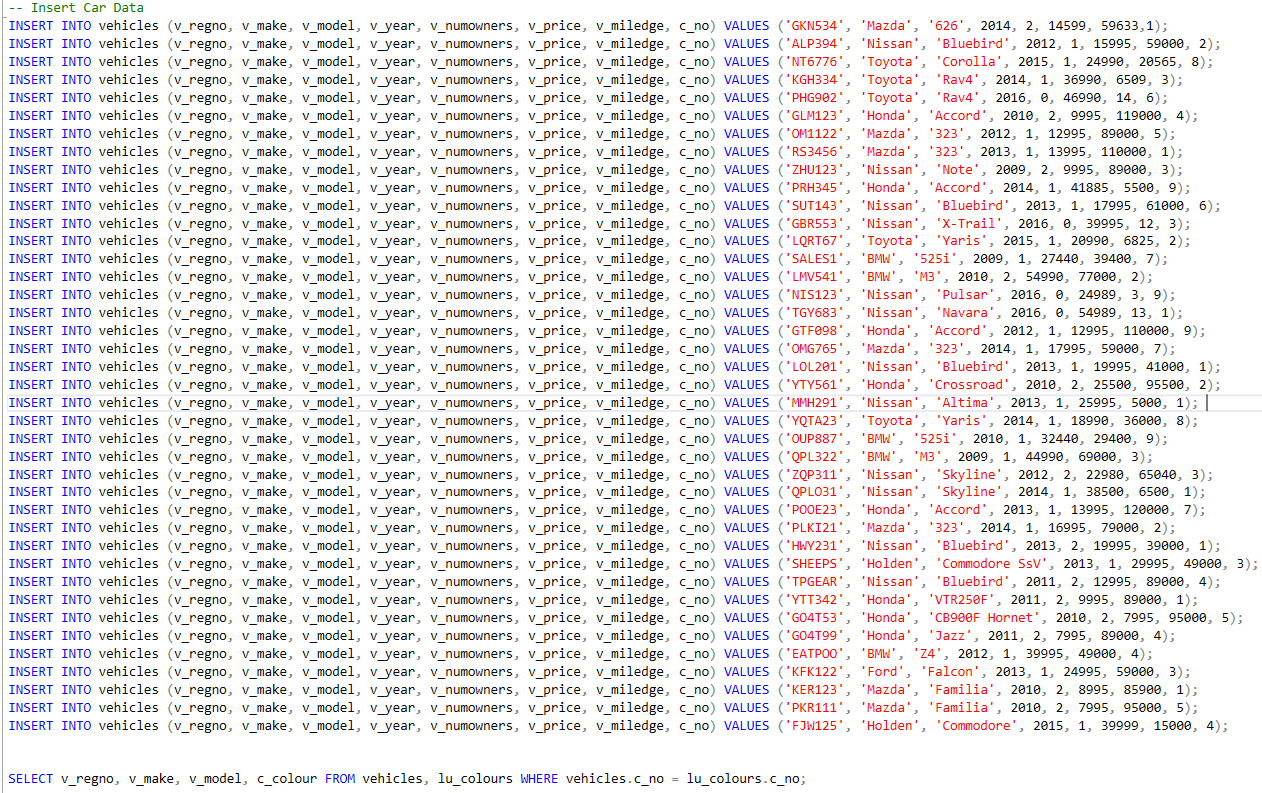
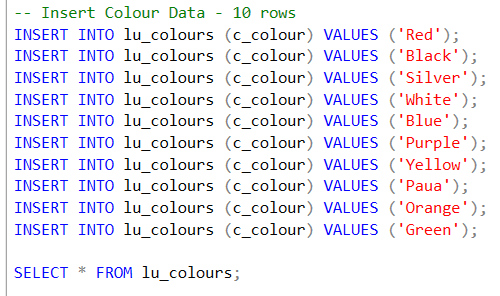
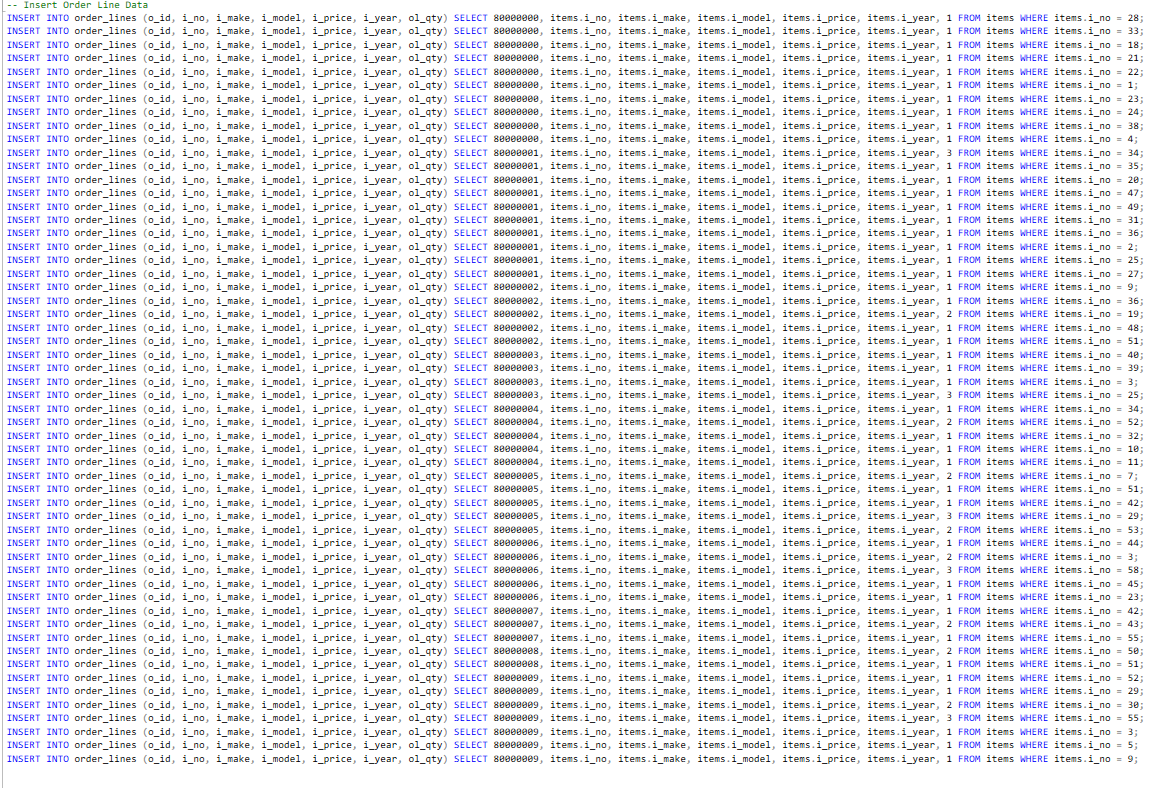
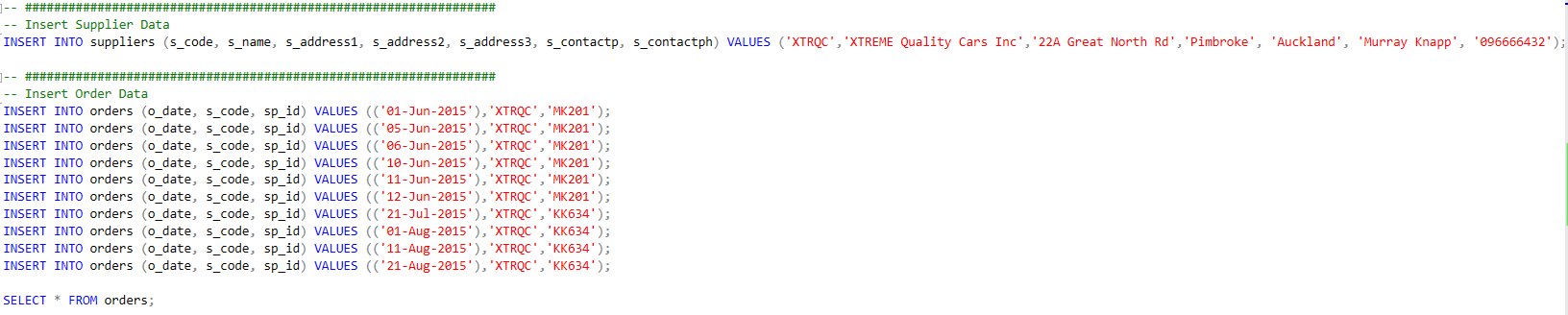
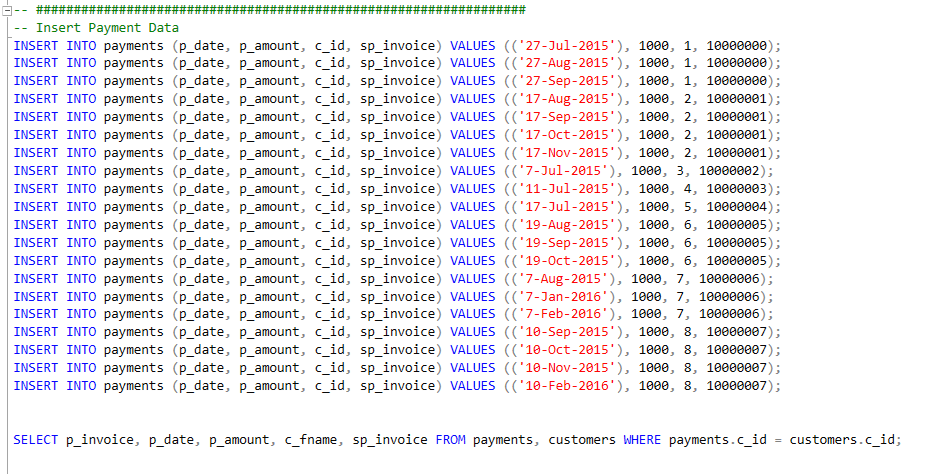
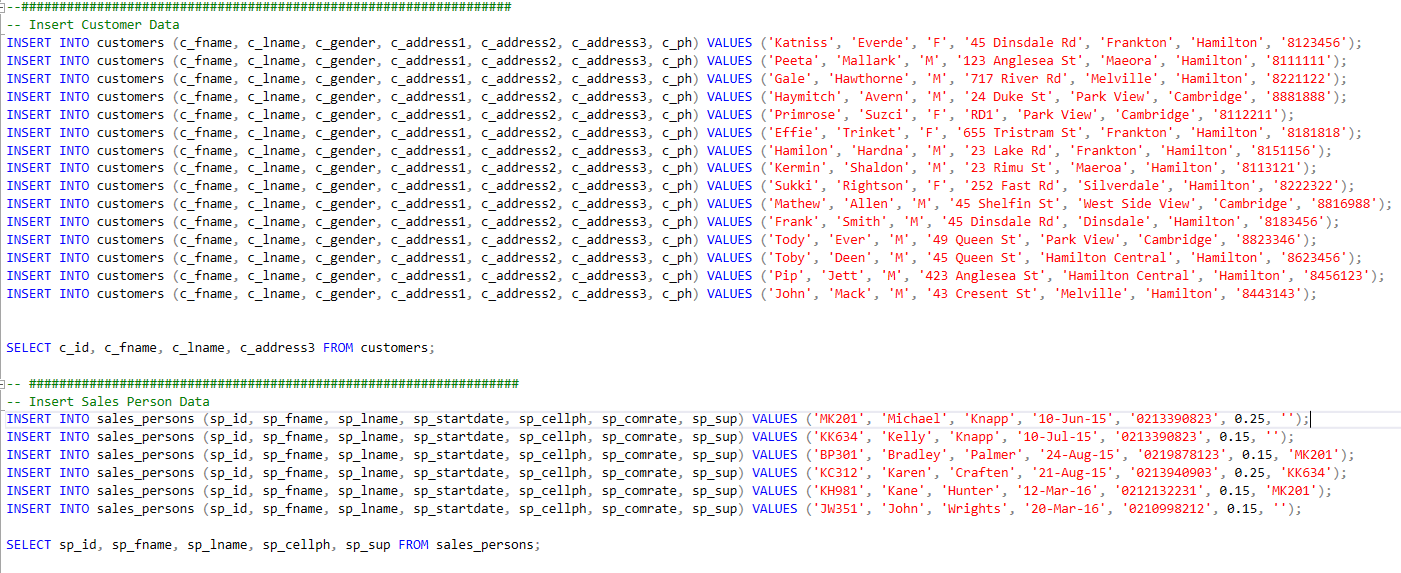
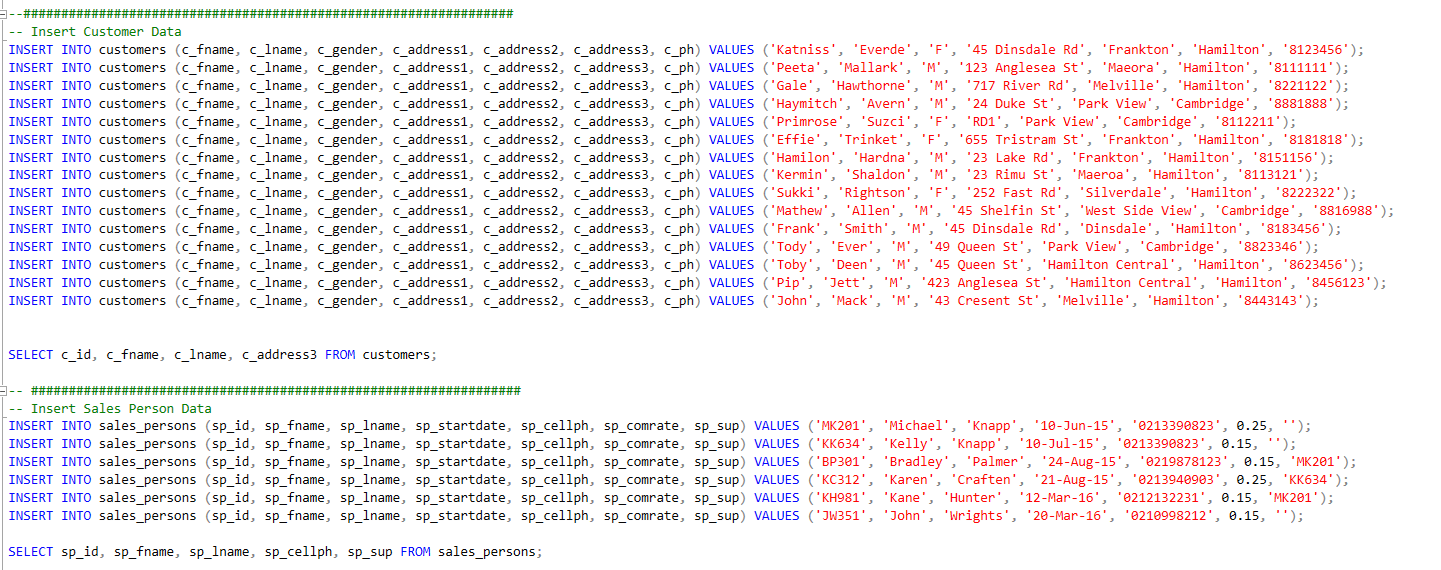
1. The first script is responsible for creating the tables and populating them with data.

1.1. Rewriting the Oracle Script in MSSQL Server and make changes to the data types, as required to make them MSSQL Server compatible.

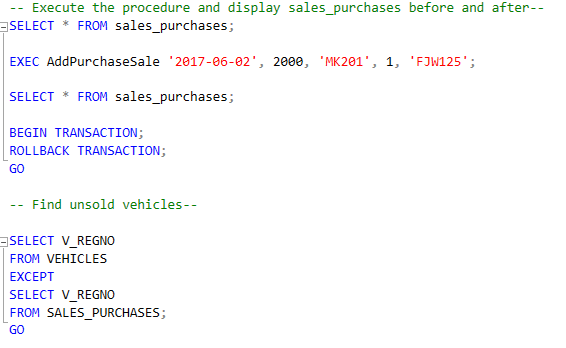
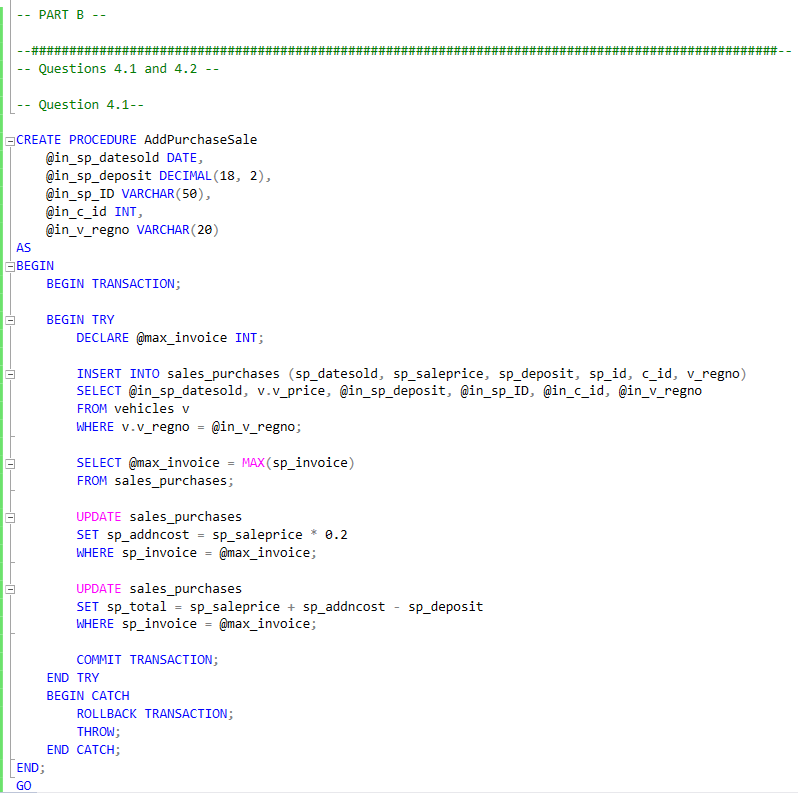
1.2. Creating each of the tables

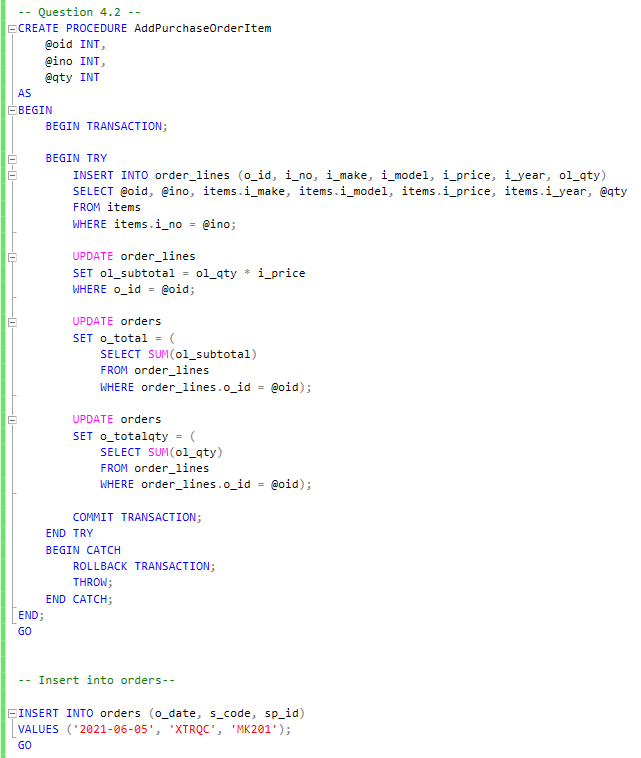


1.3. Modify the scripts to populate the tables.

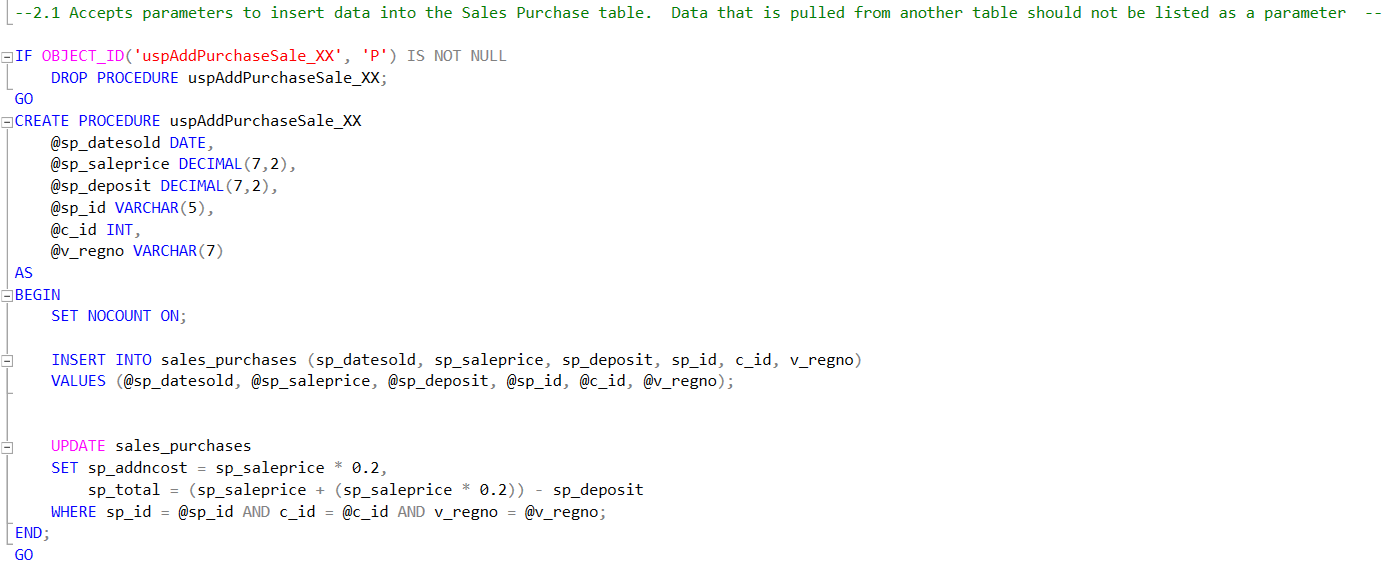


2.Modify the second script (4.1 and 4.2 Look for section - - Question 4.1) to work in MSSQL Server correctly. The original instructions given to create the Oracle script were: Create a procedure uspAddPurchaseSale\_XX() which inserts data into the Sales Purchases table.The procedure performs the following act.

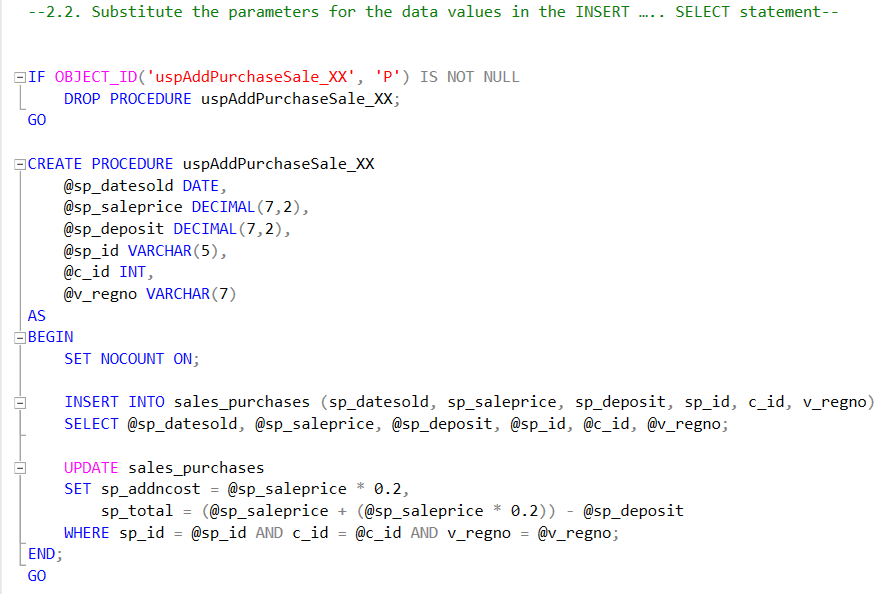




2.1. Accepts parameters to insert data into the Sales Purchase table. Data that is pulled from another table should not be listed as a parameter.

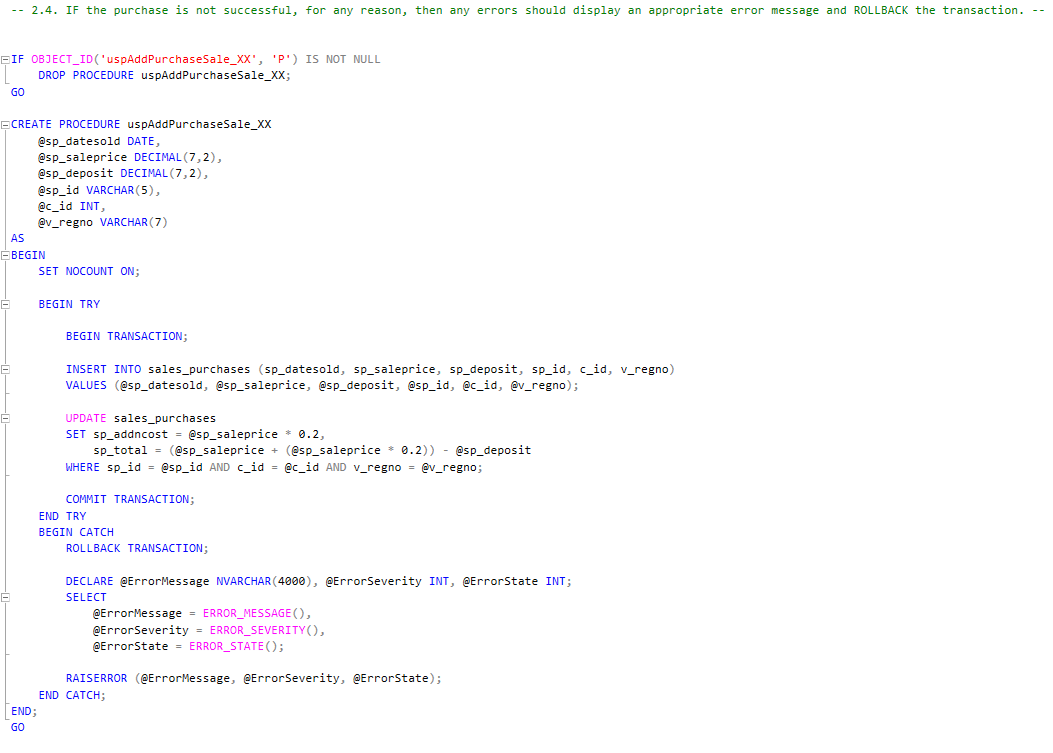
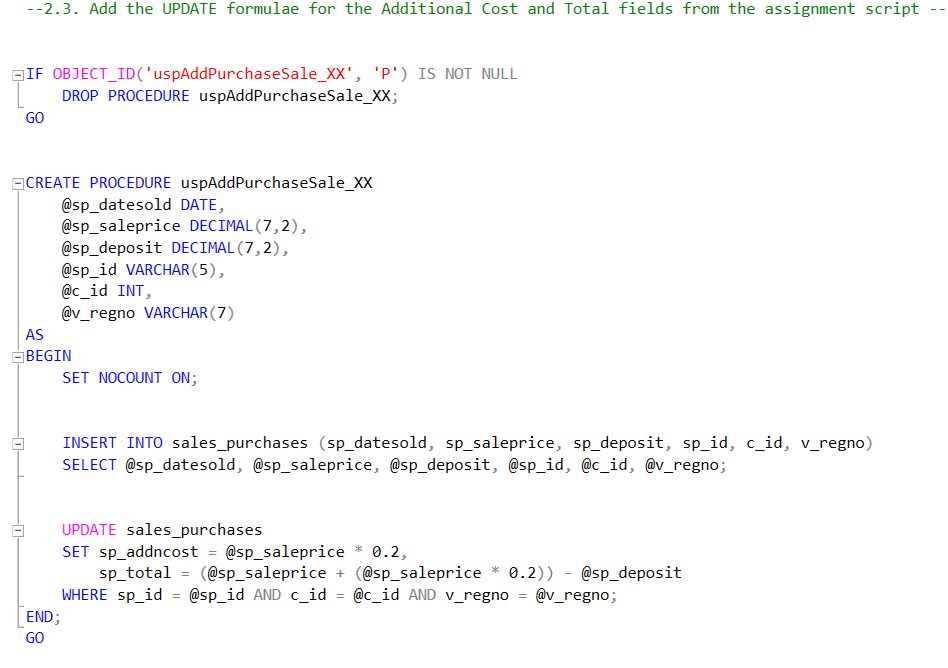


2.2. Substitute the parameters for the data values in the INSERT ….. SELECT statement



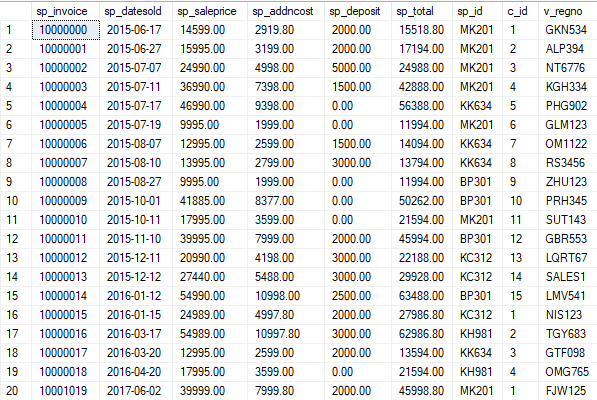
2.3. Add the UPDATE formulae for the Additional Cost and Total fields from the assignment script

2.4. IF the purchase is not successful, for any reason, then any errors should display an appropriate error message and ROLLBACK the transaction.

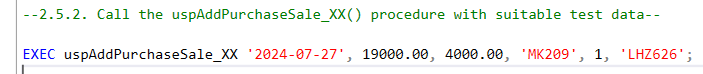


2.5. Test the procedure with test data to show it works.

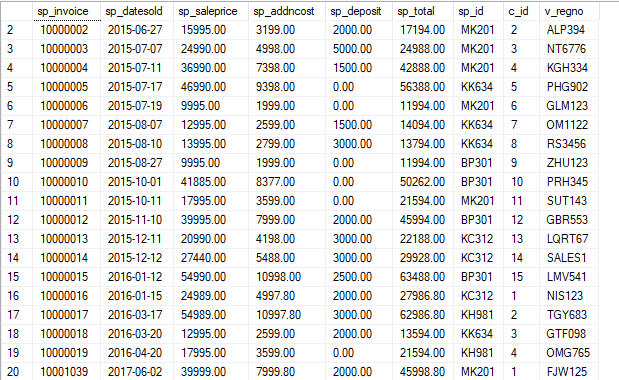
2.5.1. Perform a SELECT query on Sales\_Purchases



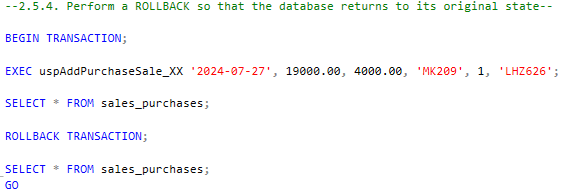
2.5.2. Call the uspAddPurchaseSale\_XX() procedure with suitable test data



2.5.3. Perform a SELECT query on Sales\_Purchases again

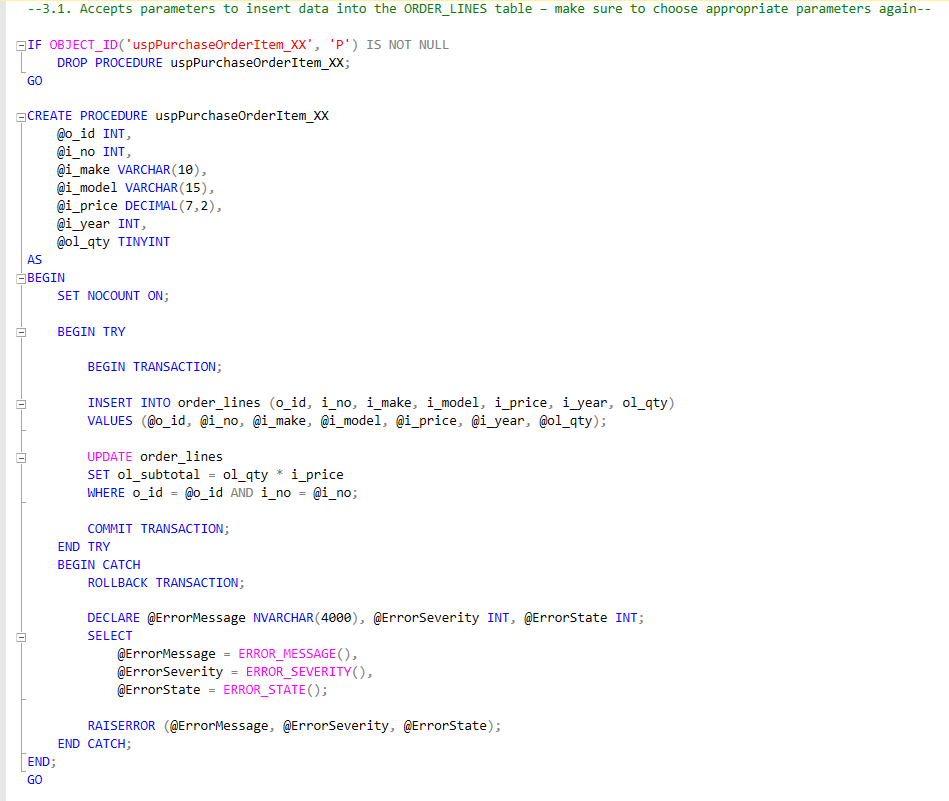


2.5.4. Perform a ROLLBACK so that the database returns to its original state

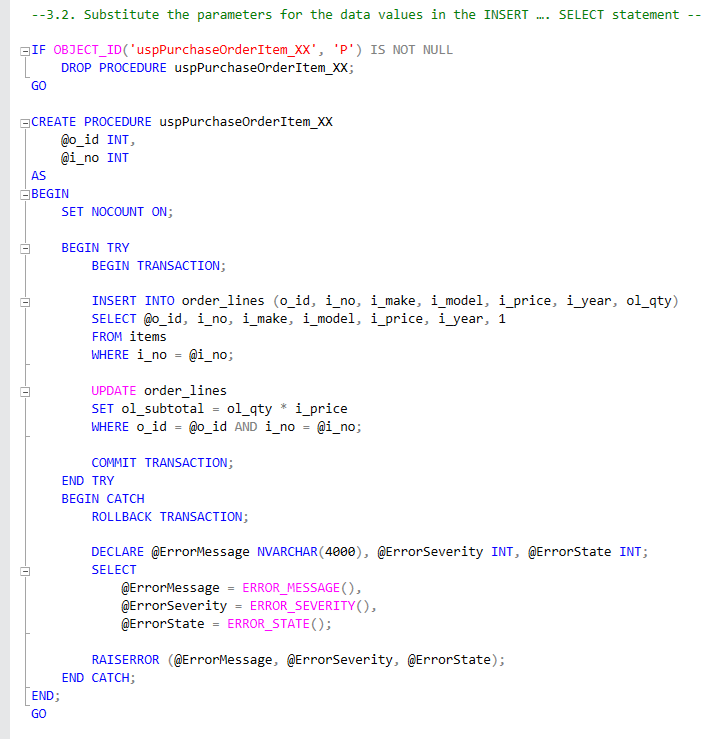


3. Modify the second script (4.1 and 4.2 Look for section - - Question 4.2) to work in MSSQL Server correctly. The original instructions given to create the Oracle script were: Create a procedure called uspPurchaseOrderItem\_XX().This procedure inserts data into the ORDER\_LINES table. The procedure performs the following:

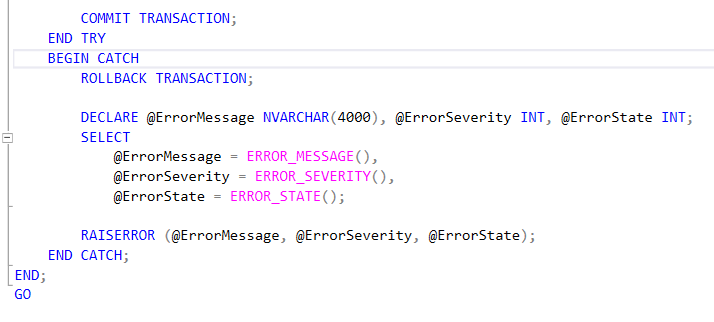
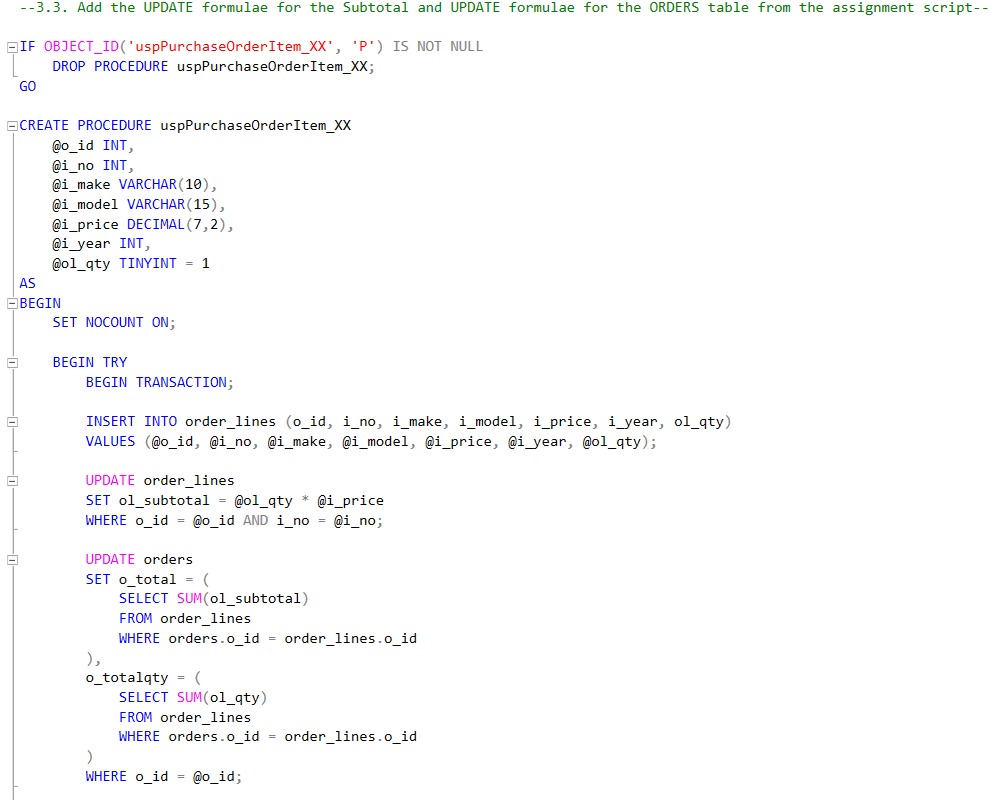
3.1. Accepts parameters to insert data into the ORDER\_LINES table – make sure to choose appropriate parameters again



3.2. Substitute the parameters for the data values in the INSERT …. SELECT statement



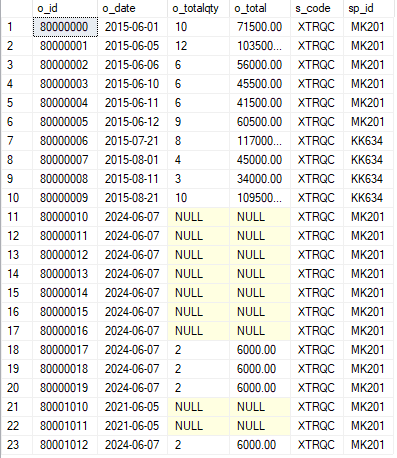
3.3. Add the UPDATE formulae for the Subtotal and UPDATE formulae for the ORDERS table from the assignment script.



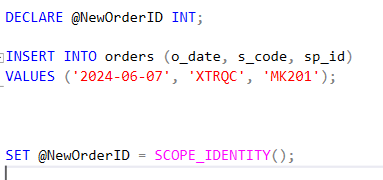
3.4. Test the procedure with test data to show it works – create an order first, and then test the procedure as such:

3.4.1. Perform a SELECT query on orders

C:\Users\rehan\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot 2024-06-07 223004.png



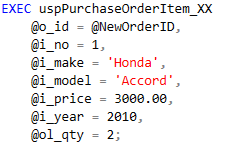
3.4.2. Create an ORDER so that you can get an order number



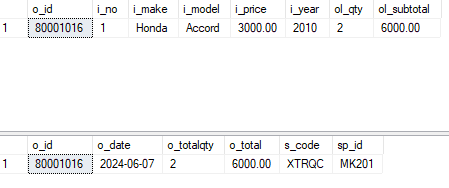
3.4.3. Perform a SELECT query on your order to get the order number



3.4.4. Call the uspAddPurchaseOrderItem\_XX() with your order number and other required parameters



3.4.5. Perform a SELECT query on ORDER\_LINES and ORDERS based on your order\_number



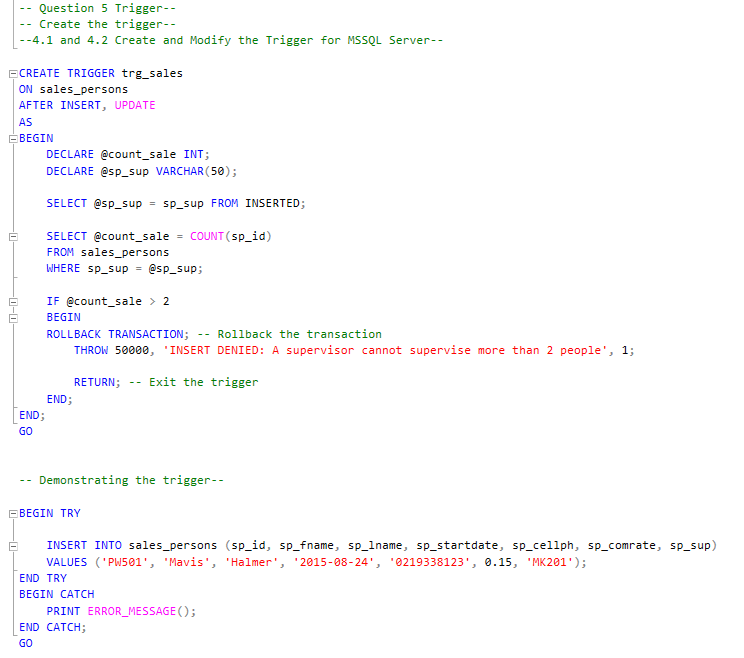
3.4.6. Perform a ROLLBACK so that the database returns to its original state



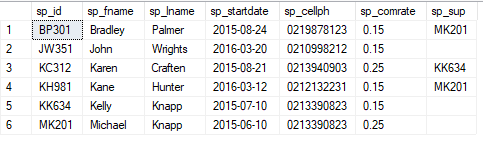
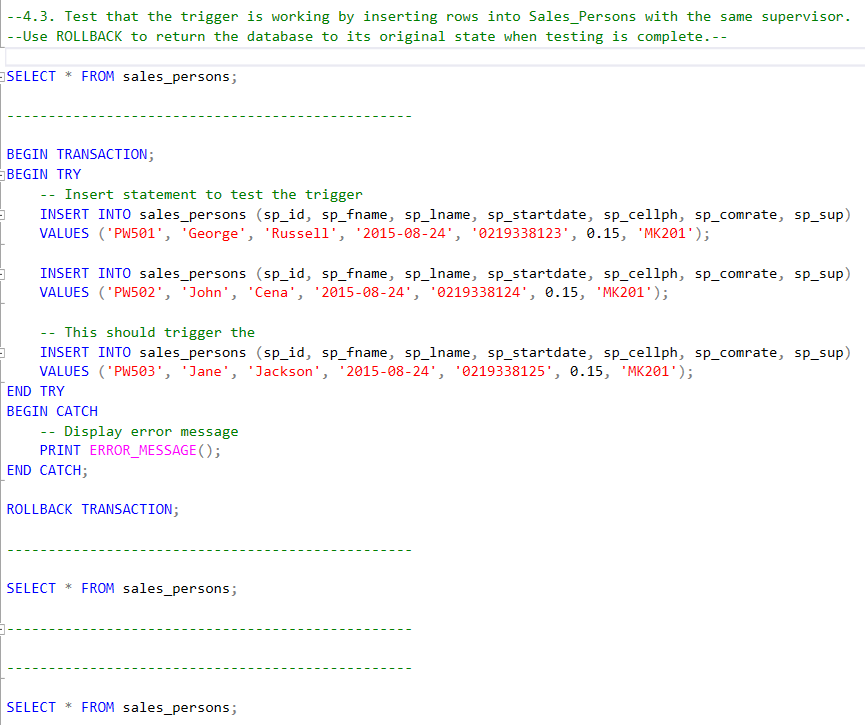
4. 5. Create the following business rules

4.1. A supervisor can supervise no more than 2 people. This trigger should occur on INSERT and UPDATE from the sales persons table.

4.2. Modify the script to work in MSSQL Server correctly.



4.3. Test that the trigger is working by inserting rows into Sales\_Persons with the same supervisor. Use ROLLBACK to return the database to its original state when testing is complete.



5.

Explanation for how the stored procedures work together for PurchaseSale:

Firstly the parameters procedure takes the parameters for the date of sale, amount, sale person ID customerID and rego number.

Then the insert operation inserts a new record into the Sales purchases table and using the provided parameters. And the Commit command is there so that the transaction is committed and if there is an error the transaction is rolled back.

Explanation for PurchaseorderItem:

Like the PurchaseSale procedure the PurchaseOrderItem procedure takes the parameters to accept the oderID, item number and quantity.

After that the insert operation takes the new records into the order\_Line table using the given parameters. The items are then fetched into the items table based on the item number. Like the purchasesale procedure the PurchaseorderItem takes the transaction committed if there is an error then the transaction is rolled back.

The update operation is there to update the items with the subtotal, total and total quantity of the purchase items.

**APA referencing**

* Chatgpt
* <https://www.youtube.com/watch?v=f6VWSlnHGCE&ab_channel=edureka%21>
* <https://www.youtube.com/watch?v=PkrjuQ4dNnk&ab_channel=InternetAuthoring>
* <https://www.mssqltips.com/sqlservertip/6132/create-alter-drop-and-execute-sql-server-stored-procedures/>