Israel Martinez

May 11, 2020

INFO 330

Assignment 06

Database Administration

# Introduction

The purpose of database administration (from what I have learned from studying everything up to Module 06) is to promote easy access and contribution to a database. It is important to know what constraints each table of data has and how those constraints should be interacted with by outsiders. If this is achieved, the data itself can promote information and knowledge to its greatest extent.

# Creating Tables

To create the tables, I first began by looking back at the Module 01 - Database Design. After looking over my first assignment, I determined that it was too simplistic to use as a template for creating tables because I remembered there were integrity constraints that were essential in the beginning stages of creating columns, such as “Not Null” and “Identity” for the primary key. So, I looked further into Module 02 – Database Options to get better examples. It was in Assignment 02 where I found a table creation statement that I felt was the best one to go off from (Figure 1).

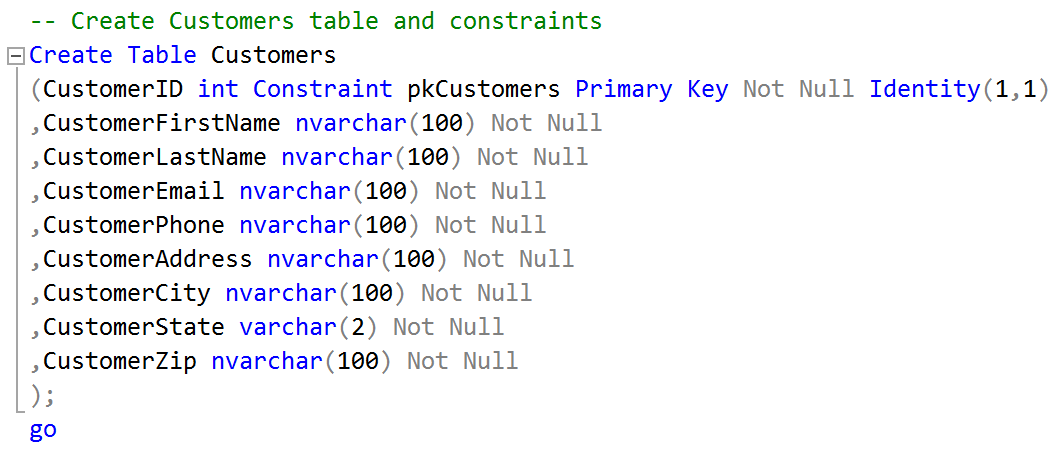


Figure 1: Example of creating a table called “Customers.”

# Creating Constraints

I continued looking at the content of Module 02 to find examples of constraint creation statements. Referencing Assignment 02 was not enough to exemplify the complex constraints requested in this assignment, so I looked further into the Lab sections content of the module. In there, I came across Lab Session 02, where there were numerous examples of unique, check, default, and foreign key constraints. I focused more on the check constraint examples (Figure 2). This is because they were good examples on how to format phone numbers.

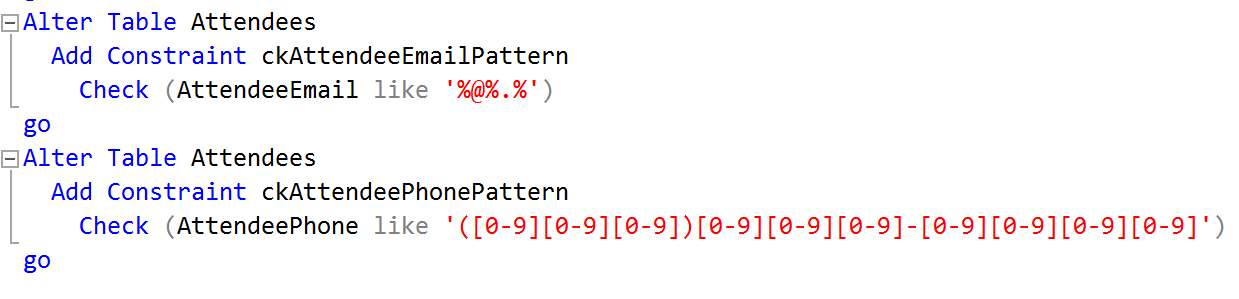


Figure 2: Example of creating check constraints for the email and phone number columns.

## Using UDF for check constraints

Creating the constraints for most of the tables did not take long. It was not until I started creating the “EnrollmentDateTime” constraint in the Enrollments table where I was stumped. This was because of the requirement of referencing the CourseStartDate column in the Courses table.

Thankfully, this problem was addressed in the Module 6 discussion board and it had an answer (Figure 3). After reading Randal Root’s response, I opened the Assignment 04 document and found the UDF YouTube video that was referenced in the reply. It was a very easy-to-follow video covering how to approach the exact problem I was having trouble on.

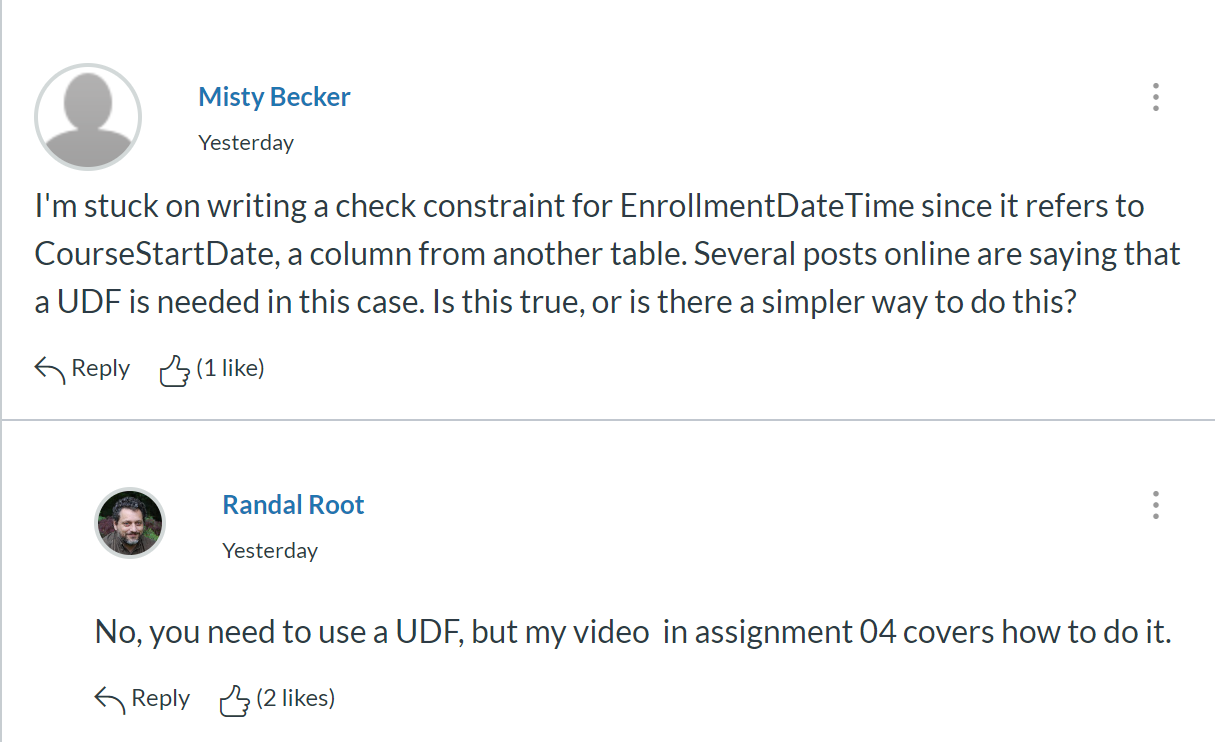


Figure 3: Randal Root referring to his Functions-05 Using a UDF for a Check Constraint [video](https://www.youtube.com/watch?v=NxNJJvG7FzU&feature=youtu.be&list=PLfycUyp06LG_ShpiOSTkQmu69sZD2Ng13) (External Site).

# Creating Views

To create the views of the tables, I again referred to my solutions in Assignment 02 (Figure 4). I followed the same format by titling the views with a “v” in front the table names and selecting all the columns. It took less than 2 minutes to complete them.

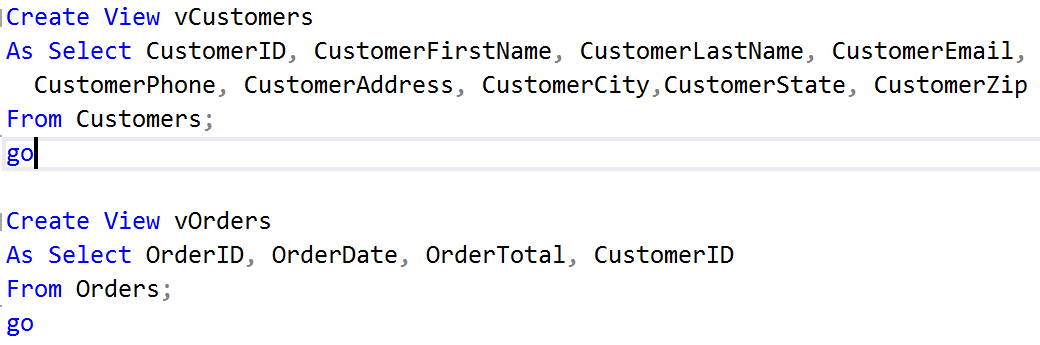


Figure 4: Example of creating a view of tables of Customers table.

## Creating Reporting View

At first, I did not understand what was meant by a “reporting view.” I looked on the Module 06 Discussion Board and saw that someone brought it up and a student referenced the Tuesday lecture in week 6. I spent about 30 minutes searching through the transcript and the chat messages trying to fully understand what was expected. Then, I finally realized that the reporting view is just the same as the Enrollment Tracker table in the MetaDataWorksheet.

To complete this task, I looked my solutions for the tasks in Assignment 04. I chose the very last one as my template because it was the most complex and resembled what I needed to do to create the reporting view (Figure 5).

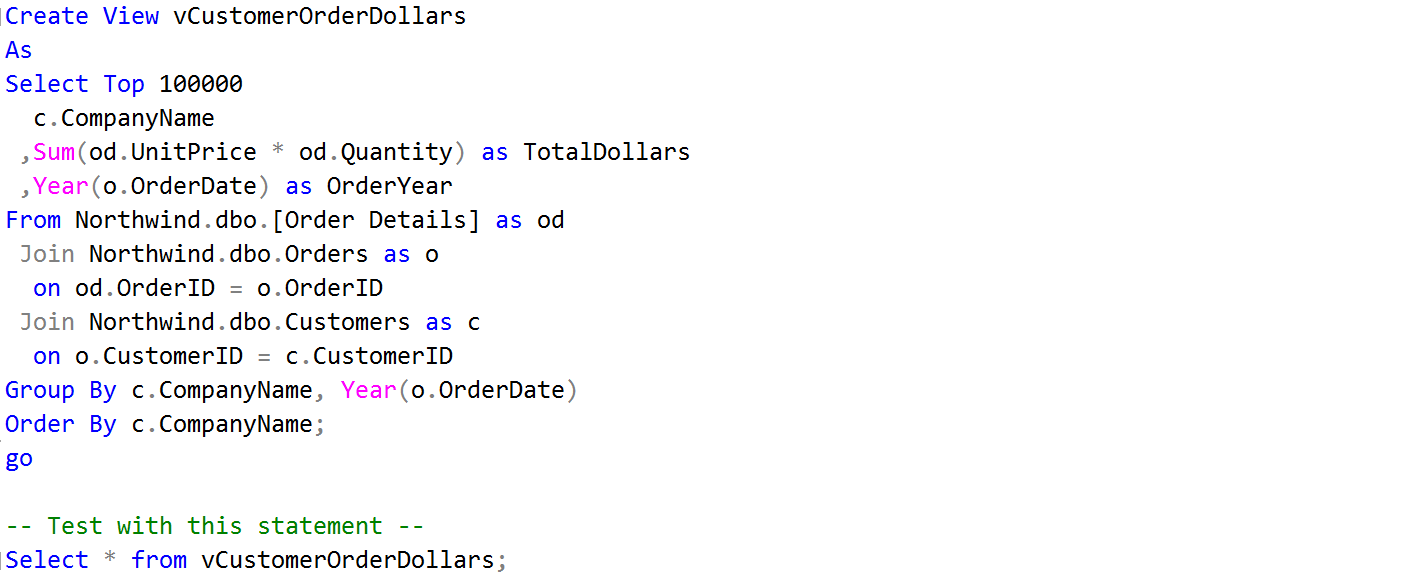


Figure 5: Example of creating a reporting view of total dollars customers placed each year.

# Creating Procedures

To create the procedures, I just followed the solutions that I made in Assignment 05. It was not very complicated to apply those solutions to the tables for Assignment 06. However, it did take some time to replace the columns to align with the ones I made for this assignment. I made an effort to be cognizant about the variables and columns I entered in so that I did not have to go back and correct mistakes which would take some time to fix.

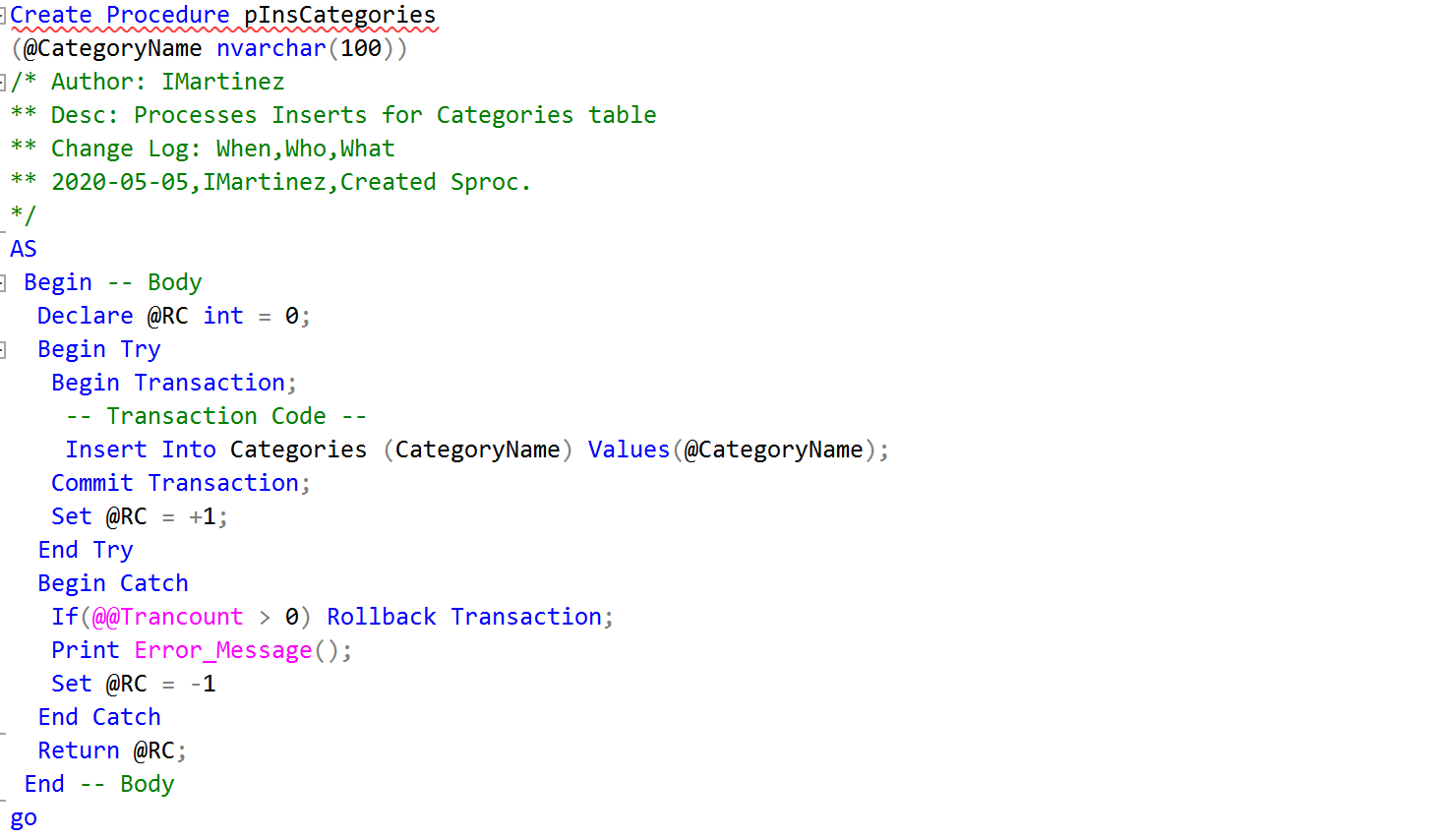


Figure 6: Example of creating an insert stored procedure for Categories table.

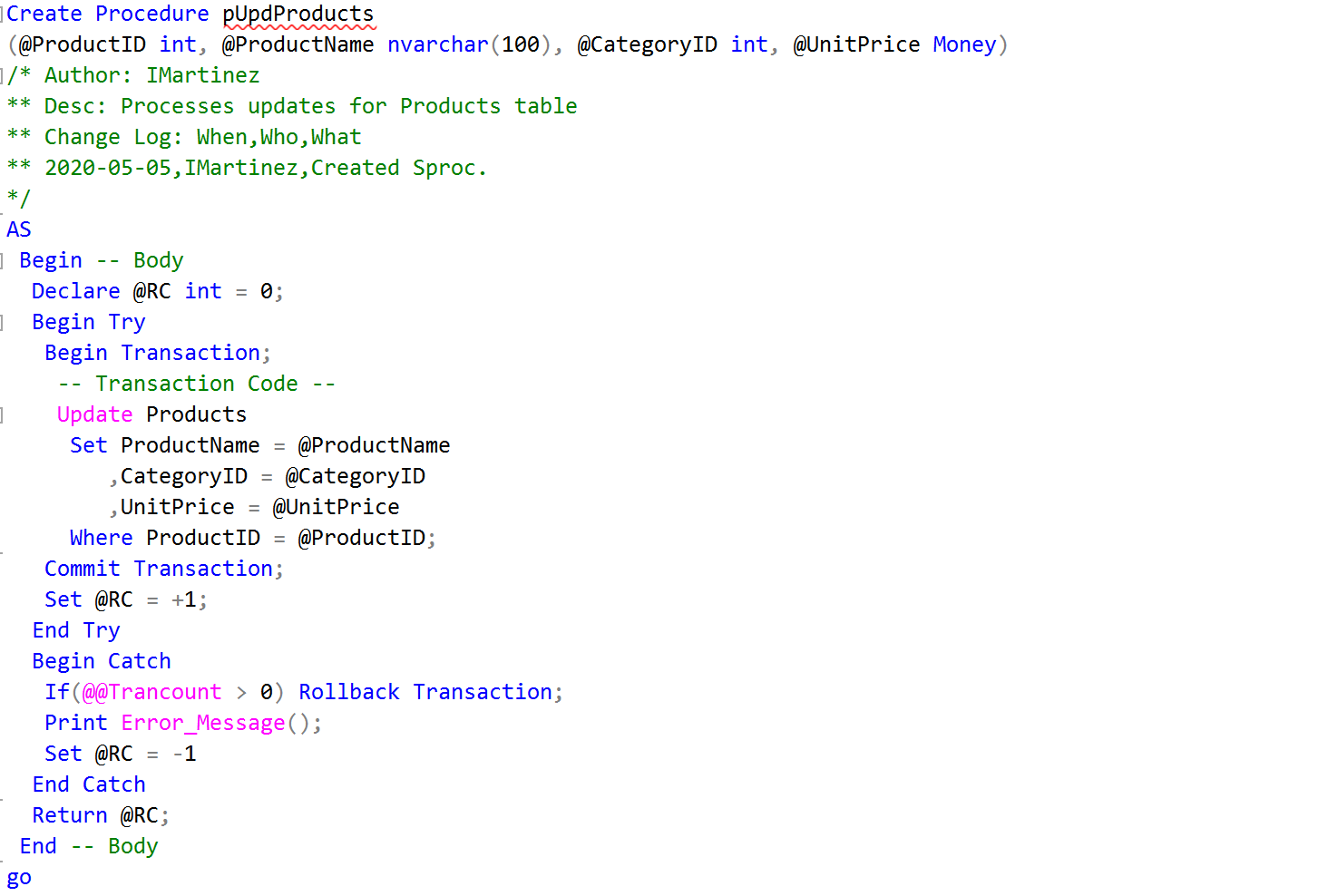


Figure 7: Example of creating an update stored procedure for Products table

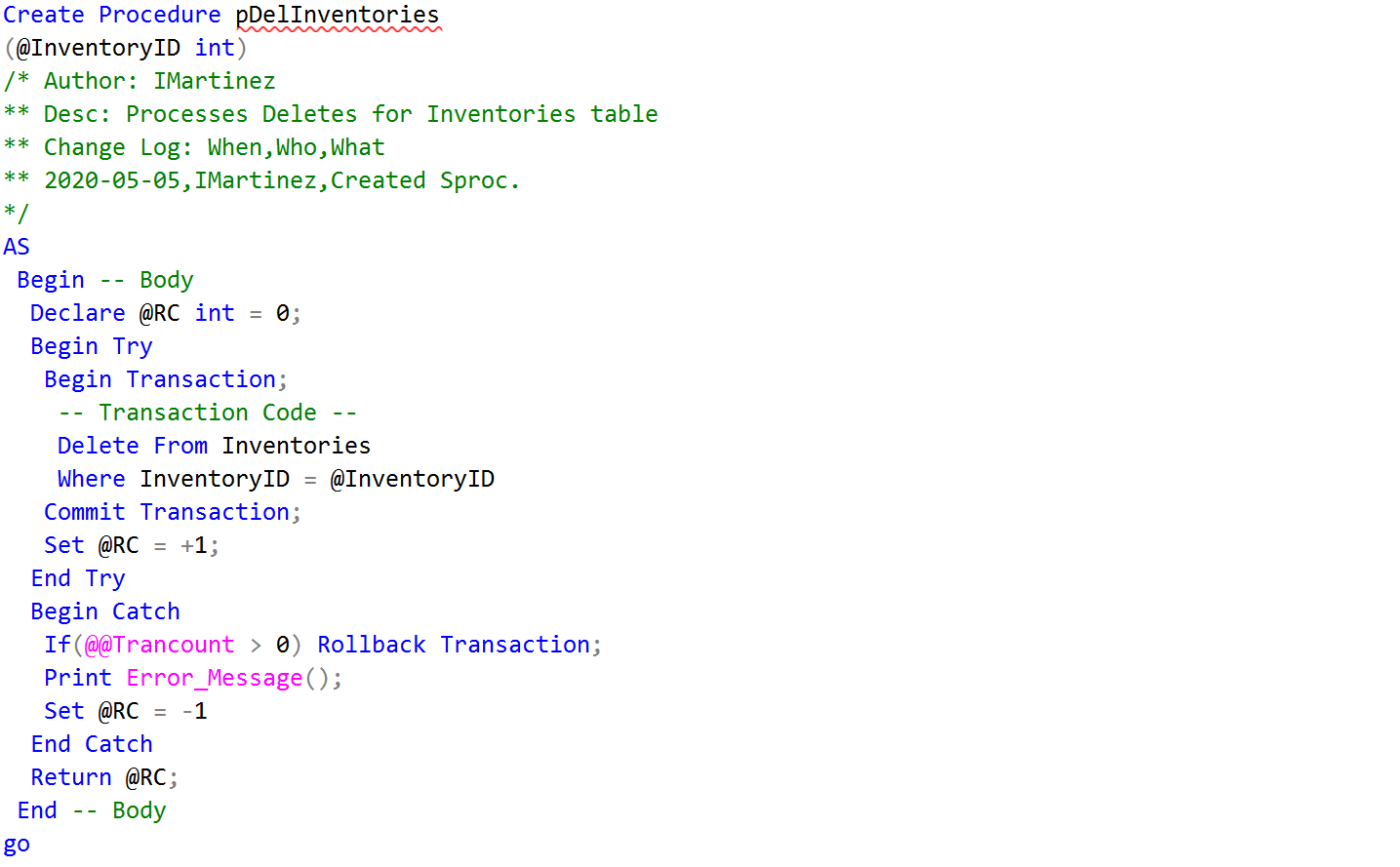


Figure 8: Example of creating a delete stored procedure for Inventories table

# Testing Views and Sprocs

I referenced the transactional stored procedures from my Assignment 05. While I created insert, update, and delete procedures in my assignment query, I only executed the insert procedures because it was not requested to update or delete any information going forward. That is why I share here only insert statements since that is the command I used the most at the end of the assignment.



Figure 9: Example of executing an insert procedure for several tables.

# Summary

After attempting to complete the tasks of Assignment 06, I have learned that Database Administration can be a very rigorous and tedious practice. One little misinput or mistake can disrupt the entire data infrastructure that is so difficult to build up in the first place. But by utilizing the content that I have learned and created in this class thus far, it was a smoother process than I thought it would be.