Lab Homework 3: Transactions and Temporal Tables

# Learning Objectives

In this exercise, you will practice programming in Microsoft SQL Server:

* Writing transaction safe code and encapsulating it into a stored procedure
* Creating a temporal table
* Writing Temporal queries

# Setup

To complete this homework or follow along with the instructor’s class demos from your coursework, you must first complete the steps in this setup section. For this exercise we require the SQL Server instance in the Docker environment. For more details on the how’s and why’s of managing the database environments used in this course, consult Lab 1. Instructions for this assignment:

1. Open the PowerShell prompt on Windows or the Terminal on MacOS.
2. Type:  
   cd adv-db-labs  
   to change the working directory to the repository folder. If you are in the correct spot, your command prompt should have adv-db-labs in it, for instance: PS adv-db-labs>
3. Change into the mssql folder, type:  
   cd mssql  
   if you are in the correct folder, your command prompt should have mssql in it.
4. Bring up the MSSQL environment, type:  
   docker-compose up -d
5. Check to make sure the environment is running, type:  
   docker-compose ps

The state of the **mssql** should be **Up** on **port 1433**.

1. You are now ready to connect to the running instance. Open **SQL Server Management Studio** (on Windows) or **SQL Server Operations Studio** (if you’re on a Mac), a.k.a the SQL Client. Use the SQL Client to connect to SQL server with:
   1. Server name: **localhost**
   2. Authentication: **SQL Server Authentication**
   3. Login: **sa**
   4. Password: **SU2Orange!**
2. After you’ve connected, open a new query window by pressing CTRL+n.
3. You are ready to begin the lab!

# Exercises

Complete each of the following exercises. If you are unsure how to accomplish the task, please consult the coursework videos where there are explanations and demos.

In this example we will create a series of database objects in the **Demo** database to track Basketball player shooting statistics across a 5-minute period.

1. In the demo database, create two tables:
   1. The first table **players** should have columns player id (int pk), player name (varchar), shots attempted (int) shots made (int)
   2. The second table **shots** should have columns shot id (int pk), player id (int fk to players), clock time (datetime) shot made (bit)
   3. Add two players to the players table. Mary and Sue initialize the players with 0 shots attempted and made.
2. Write transaction safe code as a stored procedure which when given a player id, clock time, and whether the shot was made (bit value) will add the record to the **shots** table and update the player record in the **players** table. For example, If Mary takes a shot and makes it, then misses the next one, there would be two records in the **shots** table and her row in the **players** table should have 2 attempt and 1 shot made. Execute the stored procedure to demonstrate the transaction is ACID compliant.
3. Alter the **players** table to be a system-versioned temporal table.
4. Execute your stored procedure from part 2 to create at least 15 shot records over a 5-minute period. Make sure there are records in the first ½ of the 5-minute period and at few in the last minute of the 5-minute period.
5. Write SQL queries to show:
   1. The player statistics at the end of the 5-minute period (current statistics).
   2. The player statistics exactly 2 minutes and 30 seconds into the period.
   3. The player statistics in the last minute of the period.

# Turning it in

Take your copy and paste each of the solutions to the exercises into the submission template file included with this assignment. Make sure your name and SU email are at the top and turn in your work through the course learning management system.

# Tear-Down

When you are finished with the homework you should stop the environment:

1. From the terminal window where you typed docker-compose up -d type in the following:  
   docker-compose stop