## Basic Information

Your Name: STEPHEN OMONDI  
Your SUID: 943934046  
Your Email: SOOMONDI@SYR.EDU  
Date Due: Friday, October 18, 2019  
Homework #: LABWORK 3

# 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.

---which database to use--

USE Demo;

GO

--drop table if it already exists, otherwise create it

DROP TABLE IF EXISTS shots;

GO

--drop table if it already exists, otherwise create it

DROP TABLE IF EXISTS players;

GO

---create tables in the demo database---

CREATE TABLE players(

player\_id INT IDENTITY PRIMARY KEY NOT NULL,

player\_name VARCHAR(255) NOT NULL,

shots\_attempted INT NOT NULL,

shots\_made INT NOT NULL

);

GO

--create table 'shots'--

CREATE TABLE shots(

shot\_id INT IDENTITY PRIMARY KEY NOT NULL,

player\_id INT FOREIGN KEY REFERENCES players(player\_id) NOT NULL,

clock\_time DATETIME NOT NULL

DEFAULT CURRENT\_TIMESTAMP,

shot\_made BIT

);

GO

--insert players into the players table--

INSERT INTO players(player\_name, shots\_attempted, shots\_made)

VALUES('Mary', 0, 0),

('Sue', 0, 0);

GO

1. 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.

---stored procudure for transactions---

DROP PROCEDURE IF EXISTS player\_record

GO

CREATE PROCEDURE player\_record

@player\_id INT,

@shot\_made BIT

AS

BEGIN TRY

BEGIN TRANSACTION

INSERT INTO shots(player\_id, shot\_made)

VALUES(@player\_id, @shot\_made);

IF @@ROWCOUNT <> 1 THROW 50001, 'Insert to players table affected 0 rows, expecting 1',0

UPDATE players

SET shots\_made = shots\_made + @shot\_made,

shots\_attempted = shots\_attempted + 1

WHERE player\_id = @player\_id;

IF @@ROWCOUNT <>1 THROW 5001, 'Update to shots table affected 0 rows, expecting at least 1',0

COMMIT

END TRY

BEGIN CATCH

PRINT error\_message()

ROLLBACK

END CATCH

GO

---set transaction isolation level-

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

GO

--call the player procedure for Mary---

EXEC player\_record @player\_id=1, @shot\_made=1;

GO

--call the player procedure for Sue

EXEC player\_record @player\_id=2, @shot\_made=0;

GO

----verify transactions---

Select \* from players;

Select \* from shots;

1. Alter the **players** table to be a system-versioned temporal table.

---ALTER PLAYERS TABLE---

ALTER TABLE players

ADD StartTime DATETIME2 GENERATED ALWAYS AS ROW START

HIDDEN DEFAULT GETUTCDATE(),

EndTime DATETIME2 GENERATED ALWAYS AS ROW END

HIDDEN DEFAULT

CONVERT(DATETIME2, '9999-12-31 23:59:59.9999999'),

PERIOD FOR SYSTEM\_TIME (StartTime, EndTime)

--ENABLE VERSIONING---

ALTER TABLE players

SET (SYSTEM\_VERSIONING = ON (HISTORY\_TABLE=dbo.Player\_history))

GO

1. 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.  
   --call the player procedure for Mary---

EXEC player\_record @player\_id=1, @shot\_made=1;

GO

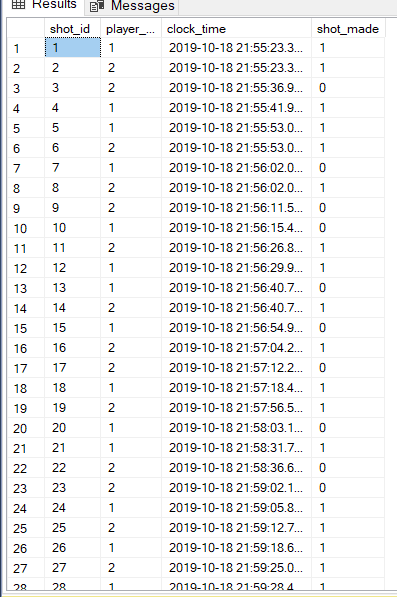
--call the player procedure for Sue

EXEC player\_record @player\_id=2, @shot\_made=0;

GO

----verify shots ---

Select \* from shots;



1. Write SQL queries to show:
   1. The player statistics at the end of the 5-minute period (current statistics).

Select \* from players;

* 1. The player statistics exactly 2 minutes and 30 seconds into the period.

--statistics in the first 2 mins and 30 seconds--

SELECT P.player\_name, S.shot\_made AS Made, P.shots\_attempted as Attempted FROM shots S

INNER JOIN players P ON P.player\_id = S.player\_id

WHERE S.clock\_time <= DATEADD(SECOND, 150, '2019-10-18 22:41:02.907')

ORDER BY S.shot\_made;

GO

* 1. The player statistics in the last minute of the period.

--statistics for the last minute into the game--

SELECT P.player\_name, S.shot\_made AS Made, P.shots\_attempted as Attempted FROM shots S

INNER JOIN players P ON P.player\_id = S.player\_id

WHERE S.clock\_time BETWEEN DATEADD(MINUTE, 4, '2019-10-18 22:41:02.907') AND DATEADD(MINUTE, 5, '2019-10-18 22:41:02.907')

ORDER BY S.shot\_made;

GO