Transact-SQL Stored Procedures

This document provides some simple examples of how to write Stored Procedures using the Transact-SQL language available with Azure SQL. To learn more about this topic and other constructs of this language, follow the below link:

- https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=azuresqldb-current

Table of Contents

Transact-SQL Stored procedure structure:

```
CREATE PROCEDURE <Procedure_Name>

AS

BEGIN

   <SQL Statement(s)>

END

GO
```

Transact-SQL Stored procedures provide benefits of code reuse, stronger security, reduced server/client network traffic, better performance, and easier maintenance. Feel free to read more about stored procedures benefits here:

- https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=azuresqldb-current

Examples:

We will execute a couple of stored procedures for demonstration (these code snippets available on Canvas in stored_procedure_examples.sql file). You can find more examples here:

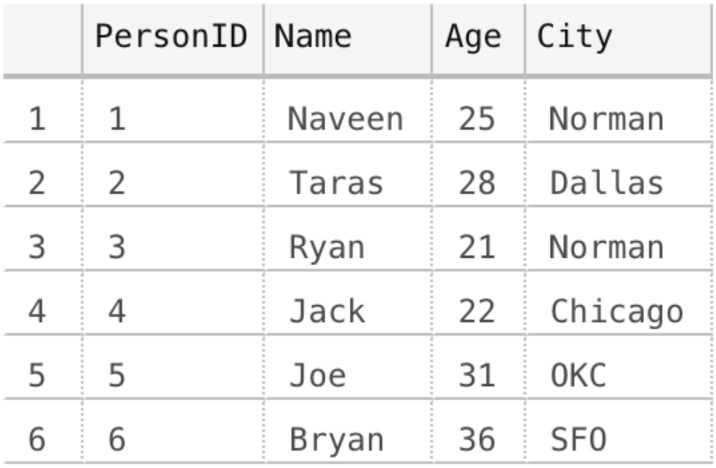
- https://docs.microsoft.com/en-us/sql/t-sql/statements/create-procedure-transact-sql?view=azuresqldb-current

First, let's create a table and insert some sample data into it:

```
DROP TABLE IF EXISTS Persons;

CREATE TABLE Persons(
    [PersonID] int PRIMARY KEY,
    [Name] varchar(25),
    [Age]int,
    [City]varchar(25));

INSERT INTO Persons(
    [PersonID],[Name],[Age],[City])
VALUES
    (1,'Naveen',25,'Norman'),
    (2,'Taras',28,'Dallas'),
    (3,'Ryan',21,'Norman'),
    (4,'Jack',22,'Chicago'),
    (5,'Joe',31,'OKC'),
    (6,'Bryan',36,'SFO');
```

EXAMPLE 1

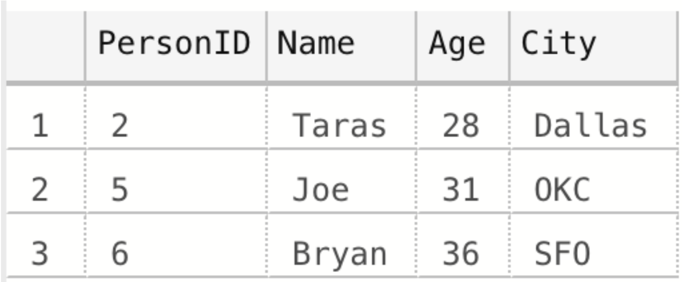
```
-- Procedure without any parameters
-- Creating a procedure sp_test1 which selects all records from Persons table

DROP PROCEDURE IF EXISTS sp_test1;

GO
CREATE PROCEDURE sp_test1
AS
BEGIN
    SELECT * FROM Persons
END

-- Executing the procedure sp_test1
GO
EXEC sp_test1;
```



	PersonID	Name	Age	City
1	1	Naveen	25	Norman
2	2	Taras	28	Dallas
3	3	Ryan	21	Norman
4	4	Jack	22	Chicago
5	5	Joe	31	OKC
6	6	Bryan	36	SFO

EXAMPLE 2

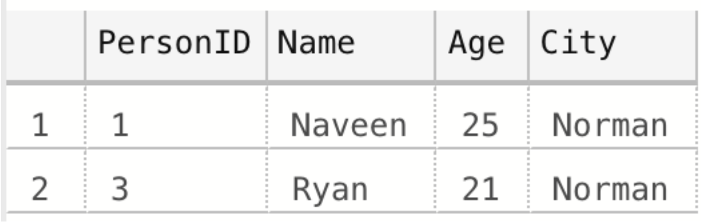
```
-- Procedure that uses one input parameter

DROP PROCEDURE IF EXISTS sp_test2;

GO
CREATE PROCEDURE sp_test2
   @age INT
AS
BEGIN
   SELECT * FROM Persons WHERE age > @age;
END

-- Executing the procedure sp_test2
GO
EXEC sp_test2 @age = 25;
```



	PersonID	Name	Age	City
1	2	Taras	28	Dallas
2	5	Joe	31	OKC
3	6	Bryan	36	SFO

EXAMPLE 3

```
-- Procedure which takes two input parameters

DROP PROCEDURE IF EXISTS sp_test3;

GO
CREATE PROCEDURE sp_test3
   @age INT,
   @city VARCHAR(20)
AS
BEGIN
   SELECT * FROM Persons WHERE age >= @age and city = @city;
END

-- Executing the procedure sp_test3
GO
EXEC sp_test3 @age = 20, @city = 'Norman';
```

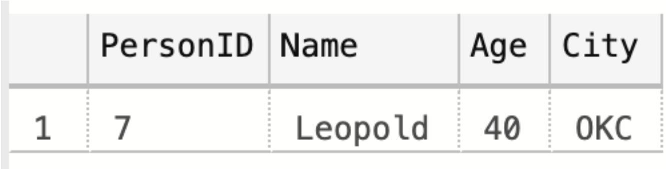


	PersonID	Name	Age	City
1	1	Naveen	25	Norman
2	3	Ryan	21	Norman

EXAMPLE 4

```
-- Procedure that uses a temporary variable and some conditional logic.
-- Insert a new person into the database. If they're an oldest person to date
--     set their city to OKC. Otherwise set it to Norman.

DROP PROCEDURE IF EXISTS sp_test4;
```

```
GO
CREATE PROCEDURE sp_test4
   @pid INT,
   @name VARCHAR(25),
   @age INT
AS
BEGIN
   DECLARE @max_age INT;
   SET @max_age = (SELECT max(age) FROM Persons);

   IF @age > @max_age
      INSERT INTO Persons VALUES (@pid, @name, @age, 'OKC');
   ELSE
      INSERT INTO Persons VALUES (@pid, @name, @age, 'Norman');
END
GO

EXEC sp_test4 @pid = 7, @name = "Leopold", @age = 40;
SELECT * FROM Persons WHERE PersonID = 7;
```



	PersonID	Name	Age	City
1	7	Leopold	40	OKC

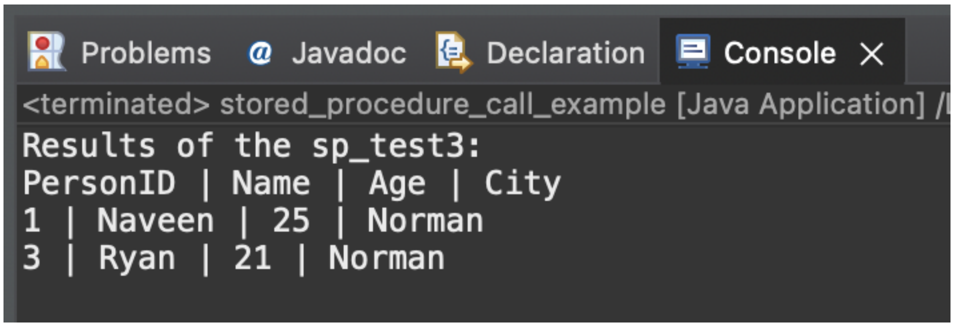
Calling a stored procedure from Java application

Please consider the below Java application that calls previously created stored procedure (code also available on canvas in stored_procedure_call_example.java file). Don’t forget to replace <your4x4> and <Your Password> to run it.

```
import java.sql.Connection;
import java.sql.ResultSet;
```

```
    import java.sql.SQLException;
    import java.sql.DriverManager;
    import java.sql.PreparedStatement;

    public class stored_procedure_call_example {

      // Database credentials
      final static String HOSTNAME = "<your4x4>-sql-server.database.windows.net";
      final static String DBNAME = "cs-dsa-4513-sql-db";
      final static String USERNAME = "<your4x4>";
      final static String PASSWORD = "<Your Password>";

      // Database connection string
      final static String URL =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;h
ostNameInCertificate=*.database.windows.net;loginTimeout=30;",
          HOSTNAME, DBNAME, USERNAME, PASSWORD);

      public static void main(String[] args) throws SQLException {
        // Connecting to the database
        try (final Connection connection = DriverManager.getConnection(URL)) {
          try (final PreparedStatement statement = connection.prepareStatement("EXEC sp_test3 @age = ?, @city =
?;")) {

              // Setting the storage procedure input parameter values
              statement.setInt(1, 20);
              statement.setString(2, "Norman");

              // Call the stored procedure
              ResultSet resultSet = statement.executeQuery();

              System.out.println("Results of the sp_test3:");
              System.out.println("PersonID | Name | Age | City ");

              while (resultSet.next()) {
                System.out.println(String.format("%s | %s | %s | %s ",
                  resultSet.getString(1),
                  resultSet.getString(2),
                  resultSet.getString(3),
                  resultSet.getString(4)));
              }
          }
        }

      }
    }
```

Execution Output:



```
<terminated> stored_procedure_call_example [Java Application] /
Results of the sp_test3:
PersonID | Name | Age | City
1 | Naveen | 25 | Norman
3 | Ryan | 21 | Norman
```

Other Useful References:

1. Another T-SQL tutorial:

   i. https://www.tutorialspoint.com/t_sql/