```
/* ************************************************************************
                    DATABASE ADMINISTRATION FUNDAMENTALS:
                  INTRODUCTION TO STRUCTURED QUERY LANGUAGE
                    SF21SQL1001, 2021/11/02 - 2021/12/09
                 https://folvera.commons.gc.cuny.edu/?cat=29
 ************************************************************************

  SESSION #9 (2021/12/02):  CREATING DATABASE OBJECTS

  1. Parameters, user-defined functions and stored procedures
 ************************************************************************

 1. LAB #10 (Procedures)
    1.1. Write a procedure `strings2_sp` in schema `lab10` in database `ace` to
         concatenate two (2) strings with an empty space (` `) between the two
         strings.

         HINT: two (2) input parameters to produce one (1) output parameter
         with the minimal size of the sum of the sizes of the first input
         parameter and the second input parameter

    1.2. To test that procedure `lab10.strings2_sp` works, execute the
         procedure passing first name and last name.

         HINT: EXEC procedure_name(@in_param1, @in_param2)
 ************************************************************************ */

CREATE SCHEMA lab10;

CREATE PROCEDURE lab10.string2_sp @in_string1 VARCHAR(50),
  @in_string2 VARCHAR(50)
AS
BEGIN
  DECLARE @out_string VARCHAR(101)              -- to accept VARCHAR(50) for
                                                -- `@in_string1`, VARCHAR(1)
                                                -- for a space + VARCHAR(50)
                                                -- for `@in_string2`
  SET @out_string = CONCAT (
      @in_string1,
      ' ',
      @in_string2
      )
  PRINT @out_string
END;


/* ************************************************************************
    1.3. Then we execute procedure `lab10.string2_sp` passing two (2) values.
         Passing more or fewer values will return an error.

                    Msg 201, Level 16, State 4, Procedure lab10.string2_sp,
                    Line 0 [Batch Start Line 53]
```

```
                    Procedure or function 'string2_sp' expects parameter
                    '@in_string2', which was not supplied.
 ********************************************************************** */

EXEC lab10.string2_sp 'John', 'Smith';


/* **********************************************************************
 2. LAB #11 (Functions)
    2.1. Write a function `strings2_udf()` in schema `lab11` in database `ace`
         to concatenate two (2) strings with an empty space (` `) between the
         two strings.

         HINT: two (2) input parameters to produce one (1) output parameter
         with the minimal size of the sum of the sizes of the first input
         parameter and the second input parameter

    2.2. To test that function `lab11.strings2_udf()` works, write a query
         calling all values from `AP1.ContactUpdates` using function
         `lab11.strings2_udf()` on `first_name` and `last_name`.

         HINT: SELECT function_name(@in_param1, @in_param2)
 ********************************************************************** */

CREATE SCHEMA lab11;

CREATE FUNCTION lab11.string2_udf (
  @in_string1 VARCHAR(50),
  @in_string2 VARCHAR(50)
  )
RETURNS VARCHAR(101)                            -- same datatype and size as
                                                -- `@out_string`, in this case
AS
BEGIN
  DECLARE @out_string VARCHAR(101)              -- to accept VARCHAR(50) for
                                                -- `@in_string1`, VARCHAR(1)
                                                -- for a space + VARCHAR(50)
                                                -- for `@in_string2`
  SET @out_string = CONCAT (
      @in_string1,
      ' ',
      @in_string2
      )
  RETURN @out_string
END;

/* **********************************************************************
    2.3. Then we use function `lab11.string2_udf` passing two (2) values.  Note
         that passing more or fewer values will return an error.

                    Msg 313, Level 16, State 2, Line 101
                    An insufficient number of arguments were supplied for the
```

```
105                     procedure or function lab11.string2_udf.
106   ************************************************************************ */
107
108  SELECT lab11.string2_udf('John', 'Smith');
109
110
111  /* ************************************************************************
112   3. In the example below, we make a function to dress up phone numbers as
113      `(xxx) xxx-xxxx` in schema `lab12`.
114   ************************************************************************ */
115
116  CREATE SCHEMA lab12;
117
118  CREATE FUNCTION lab12.phones_udf (@in_phone VARCHAR(15))
119  RETURNS VARCHAR(15)                              -- need to remember that a
120                                                   -- function RETURNS a value
121  AS
122  BEGIN
123    DECLARE @out_phone VARCHAR(15)
124    SET @out_phone = CASE                          -- `CASE` clause to check if
125        WHEN @in_phone IS NOT NULL                 -- `CONCAT` needs to be run
126          OR @in_phone <> ''
127          OR @in_phone NOT LIKE ('(%)%-%')        -- checking for formatted phone
128          THEN CONCAT (
129              '(',
130              LEFT(@in_phone, 3),
131              ') ',
132              SUBSTRING(@in_phone, 4, 3),
133              '-',
134              RIGHT(@in_phone, 4)
135              )
136        ELSE @in_phone
137        END                                       -- ending/closing `CASE` clause
138    RETURN @out_phone
139  END;                                            -- ending/closing function
140
141
142  /* ************************************************************************
143      3.1 Then we use function `lab11.string2_udf` passing two (2) values when
144           querying `SF21SQL1001.AP1.Vendors`.
145
146           Since accessing another objects in another database, you need to call
147           the full name the function (`labs.lab11.string2_udf`) and/or the table
148           (`SF21SQL1001.AP1.Vendors`) depending in which database you are in.
149   ************************************************************************ */
150
151  SELECT VendorID,
152    SF21SQL1001.AP1.Vendors.VendorName,
153    labs.lab11.strings2_udf(SF21SQL1001.AP1.Vendors.VendorAddress1,
154      SF21SQL1001.AP1.Vendors.VendorAddress2)
155      AS VendorAddress,                            -- `labs.lab11.strings2_udf`
156                                                   -- on `VendorAddress1` and
```

```
157                                                 -- `VendorAddress2`
158     SF21SQL1001.AP1.Vendors.VendorCity,
159     SF21SQL1001.AP1.Vendors.VendorState,
160     SF21SQL1001.AP1.Vendors.VendorZipCode,
161     labs.lab12.phones_udf(VendorPhone)
162       AS VendorPhone                            -- using function
163                                                 -- `labs.lab12.phones_udf`
164  FROM SF21SQL1001.AP1.Vendors;
165
166
167  /* ***************************************************************************
168   4. This marks the end of new material.
169
170      4.1. As a developer, you should have a list of resources -- websites, books
171           or people whom you can contact for help.  The following is only a list
172           of resources -- not a recommendation of goods and/or services.
173
174           Analytics Vidhya (data science community)
175           https://analyticsvidhya.com/
176
177           Apache Spark - Unified Analytics Engine
178           https://spark.apache.org/
179
180           Apache Spark - Unified Analytics Engine - Spark SQL & DataFrames
181           https://spark.apache.org/sql/
182
183           Azure Cosmos DB
184           https://azure.microsoft.com/en-us/services/cosmos-db/
185
186           Azure SQL - Azure SQL documentation - Microsoft Docs
187           https://docs.microsoft.com/en-us/azure/azure-sql/
188
189           Cockroach Labs - CockroachDB
190           https://cockroachlabs.com/
191
192           DBeaver - Universal Database Tool
193           https://dbeaver.com/
194
195           DBeaver Community (client)
196           https://dbeaver.io/
197
198           EverSQL - SQL Query Optimizer Tool Online
199           https://eversql.com/sql-syntax-check-validator/
200
201           HeidiSQL (client)
202           https://heidisql.com/
203
204           Infrastructure as SQL (iaSQL)
205           https://iasql.com/
206
207           MariaDB (MySQL fork, not related to Oracle)
208           https://mariadb.org/
```

```
209
210           Microsoft Azure
211           https://portal.azure.com/
212
213           Microsoft Azure - Quickstart Templates
214           https://azure.microsoft.com/en-us/resources/templates/
215
216           Microsoft Power Automate
217           https://flow.microsoft.com/en-us/desktop/
218
219           Microsoft Power BI (business intelligence)
220           https://powerbi.microsoft.com/
221
222           Microsoft SQL Server
223           https://microsoft.com/en-us/sql-server/
224
225           Microsoft SQL Server - Get Started
226           https://microsoft.com/en-us/sql-server/developer-get-started/
227
228           MongoDB
229           https://mongodb.com/
230
231           MongoDB Blog
232           https://mongodb.com/blog
233
234           mycli (CLI MariaDB, MySQL & Percona)
235           https://mycli.net/
236
237           MySQL (Oracle)
238           https://mysql.com/
239
240           Oracle
241           http://oracle.com/
242
243           phpMyAdmin (administration tool for MySQL/MariaDB)
244           https://phpmyadmin.net/
245
246           Poor SQL (code formatter)
247           https://poorsql.com/
248
249           PostgreSQL
250           https://postgresql.org/
251
252           Slack - codebar - sql
253           https://app.slack.com/client/T08CJBA82/CHPE04RU7
254
255           SQLite
256           https://sqlite.org/
257
258           SQLZOO
259           https://sqlzoo.net/
260
```

```
261           Tech on the Net - SQL Server
262           https://techonthenet.com/sql_server/
263
264           Vespa (big data AI, Oath/Yahoo)
265           http://vespa.ai/
266
267   ********************************************************************************
268   https://folvera.commons.gc.cuny.edu/?p=1056
269   ********************************************************************************/
```