9 User-Defined Functions and Stored Procedures

9.1 Lab Introduction

9.1.1 Purpose

The purpose of this lab is to familiarize you with user-defined functions and stored procedures in Snowflake..

9.1.2 What you'll learn

- How to create a JavaScript user-defined function
- How to create a SQL user-defined function
- How to create a JavaScript stored procedure

9.1.3 How to complete this lab

In order to complete this lab, you can type the SQL commands below directly into a worksheet. It is not recommended that you cut and paste from the workbook pdf as that sometimes results in errors.

You can also use the SQL code file for this lab that was provided at the start of the class. You would simply need to open it in TextEdit (Mac) or Notepad (Windows), and then copy and paste the SQL code directly into a worksheet.

9.2 Create a JavaScript User-Defined Function

9.2.1 Open a new worksheet and name it *UDFs* and set the context:

```
USE ROLE TRAINING_ROLE;
CREATE WAREHOUSE IF NOT EXISTS LEARNER_WH;
USE WAREHOUSE LEARNER_WH;
CREATE DATABASE IF NOT EXISTS LEARNER_DB;
USE LEARNER_DB.PUBLIC;
```

9.2.2 Run the following to create a UDF named `Convert2Meters` :

```
CREATE OR REPLACE FUNCTION Convert2Meters(lengthInput double, InputScale string )
    RETURNS double
    LANGUAGE javascript
    AS
    $$
    /*
    *  convert English measurements to meters
    */
    var scale_UC =  INPUTSCALE.toUpperCase();
    switch(scale_UC) {
```



```
    case 'INCH':
        return(LENGTHINPUT * 0.0254)
        break;
    case 'FEET':
        return(LENGTHINPUT * 0.3048)
        break;
    case 'YARD':
        return(LENGTHINPUT * 0.9144)
        break;

    default:
        return null;
        break;
    }
  $$;
```

9.2.3 Call the UDF just created as part of a SELECT statement and return the value in meters:

```
SELECT Convert2Meters(10, 'yard');
```

9.3 Create a SQL User-defined Function

9.3.1 Create a function that returns the count of orders based on the customer number. It will be a scalar function because it will return a single value. After creating the UDF, use it in a query:

```
CREATE OR REPLACE FUNCTION order_cnt(custkey number(38,0))
  RETURNS number(38,0)
  AS
  $$
    SELECT COUNT(1) FROM "SNOWFLAKE_SAMPLE_DATA"."TPCH_SF1"."ORDERS"WHERE
       o_custkey = custkey
  $$;

SELECT C_name, C_address, order_cnt(C_custkey)
FROM "SNOWFLAKE_SAMPLE_DATA"."TPCH_SF1"."CUSTOMER" LIMIT 10;
```

9.4 Creating Stored Procedures

9.4.1 Create a Javascript stored procedure named `ChangeWHSize()` .

Remember that Javascript is case-sensitive, whereas SQL is not. The JavaScript appears between the $$ delimiters, so make sure case in the JavaScript portion is preserved.

The procedure demonstrates executing SQL statements in a stored procedure:

```
/* ***
*   ChangeWHSize(wh_name STRING, wh_size STRING)
*
```



```
* Description: Change a WH size to a new size. The size is limited to larger or
   below.

*/
create or replace procedure ChangeWHSize(wh_name STRING, wh_size STRING )
    returns string
    language javascript
    strict
    execute as owner
    as
    $$
    /*
    *  Change the named warehouse to a new size if the new size is LARGE OR smaller
    */
    var wh_size_UC = WH_SIZE.toUpperCase();
    switch(wh_size_UC) {
    case 'XSMALL':
    case 'SMALL':
    case 'MEDIUM':
    case 'LARGE':
        break;
    case 'XLARGE':
    case 'X-LARGE':
    case 'XXLARGE':
    case 'X2LARGE':
    case '2X-LARGE':
    case 'XXXLARGE':
    case 'X3LARGE':
    case '3X-LARGE':
    case 'X4LARGE':
    case '4X-LARGE':
        return "Size: " + WH_SIZE + " is too large";
        break;
    default:
        return "Size: " + WH_SIZE + " is not valid";
        break;
    }

    var sql_command =
     "ALTER WAREHOUSE IF EXISTS " + WH_NAME + " SET WAREHOUSE_SIZE = "+ WH_SIZE;
    try {
        snowflake.execute (
            {sqlText: sql_command}
            );
        return "Succeeded.";   // Return a success/error indicator.
        }
    catch (err)  {
        return "Failed: " + err;   // Return a success/error indicator.
        }
    $$
    ;
```

9.4.2 Call the stored procedure with a valid warehouse size:


```
CALL changewhsize ('LEARNER_wh', 'small');
```

9.4.3 Call the stored procedure with an invalid warehouse size:

```
CALL changewhsize ('LEARNER_wh', 'XLARGE');
```

9.4.4 Suspend and resize the warehouse

```
ALTER WAREHOUSE LEARNER_WH SET WAREHOUSE_SIZE=XSmall;
ALTER WAREHOUSE LEARNER_WH SUSPEND;
```

