1 Practicing Query Fundamentals

1.1 Lab Introduction

1.1.1 Purpose

The purpose of this lab is to give you a chance to practice writing SQL statements in Snowflake. If you're experienced at writing SQL, you'll see that the SQL commands you may be familiar from other relational database management systems are also available in Snowflake. If you're new to SQL, you'll learn how to create SQL statements that you can use in your daily work.

1.1.2 What you'll learn

- How to write SELECT statements
- How to perform basic calculations
- How to insert data into a table
- How to write a pivot query

1.1.3 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.

1.2 SELECT Statements

The SELECT statement sets up a tabular view of data and/or calculations.

1.2.1 If you haven't created the class database or warehouse, do it now

```
CREATE WAREHOUSE IF NOT EXISTS LEARNER_WH;
CREATE DATABASE IF NOT EXISTS LEARNER_DB;
```

1.2.2 In this task you will run several SELECT statements, with and without conditional logic, to see what they return in the query results.

1.2.3 Navigate to **[Worksheets]** and create a new worksheet named *Query Data* and set the context:

```
USE ROLE TRAINING_ROLE;
USE WAREHOUSE LEARNER_WH;
USE DATABASE LEARNER_DB;
USE SCHEMA PUBLIC;
```

1.2.4 Perform basic calculations using SELECT, but with no table data:

```
SELECT (22+47+1184), 'My String', CURRENT_TIME();
```

1.2.5 Use AS on the previous query to rename the first column to SUM:

```
SELECT (22+47+1184) AS sum, 'My String', CURRENT_TIME();
```

1.2.6 Use AS to rename all the columns.

1.2.7 Use some conditional logic using CASE and WHEN:

```
SELECT
  CASE
     WHEN RANDOM() > 0 THEN 'POSITIVE'
     WHEN RANDOM() < 0 THEN 'NEGATIVE'
     ELSE 'Zero'
   END,
   CASE
      WHEN RANDOM(2) > 0 THEN 'POSITIVE'
      WHEN RANDOM(2) < 0 THEN 'NEGATIVE'
      ELSE 'Zero'
   END;
```

1.2.8 Run the following command to turn off query caching:

```
ALTER SESSION SET USE_CACHED_RESULT=FALSE;
```

1.2.9 Run the following command several times. Do you get the same result each time?

```
USE DATABASE SNOWFLAKE_SAMPLE_DATA;
USE SCHEMA TPCH_SF1;

SELECT * FROM lineitem LIMIT 3;
```

1.2.10 Run the following command several times. Do you get the same result each time?

```
SELECT * FROM nation LIMIT 3;
```

1.2.11 Run the SELECT on the `LINEITEM` table (step 9) 12 times. For each result, record the first three digits of the l_orderkey value.

Why do you think the two tables have different behaviors with LIMIT? What is different about the two tables?

1.3 PIVOT

1.3.1 Set your standard context in your worksheet.

1.3.2 Create a table and insert some data with the following statements:

```
USE DATABASE LEARNER_DB;
USE SCHEMA PUBLIC;

CREATE TABLE weekly_sales(name VARCHAR(10), day VARCHAR(3), amount
    NUMBER(8,2));

INSERT INTO weekly_sales VALUES
   ('Fred', 'MON',  913.24), ('Fred', 'WED', 1256.87), ('Rita', 'THU',
      10.45),
   ('Mark', 'TUE',  893.45), ('Mark', 'TUE', 2240.00), ('Fred', 'MON',
      43.99),
   ('Mark', 'MON',  257.30), ('Fred', 'FRI', 1000.27), ('Fred', 'WED',
      924.34),
   ('Rita', 'WED',  355.60), ('Rita', 'MON',  129.00), ('Fred', 'WED',
      3092.56),
   ('Fred', 'TUE',  449.00), ('Mark', 'MON',  289.12), ('Fred', 'FRI',
      900.57),
   ('Rita', 'THU', 1200.00), ('Fred', 'THU', 1100.95), ('Fred', 'MON',
      523.33),
   ('Fred', 'TUE',  972.33), ('Fred', 'MON', 4500.87), ('Fred', 'WED',
      35.90),
   ('Rita', 'MON',   28.90), ('Mark', 'FRI', 1466.02), ('Fred', 'MON',
      3022.45),
   ('Mark', 'TUE',  256.88), ('Fred', 'MON',  449.00), ('Rita', 'FRI',
      294.56),
   ('Fred', 'MON',  882.56), ('Fred', 'WED', 1193.20), ('Rita', 'WED',
      88.90),
   ('Mark', 'WED',   10.37), ('Fred', 'THU', 2345.00), ('Fred', 'TUE',
      2638.76),
   ('Rita', 'TUE',  988.26), ('Fred', 'THU', 3400.23), ('Fred', 'MON',
      882.45),
   ('Rita', 'THU', 734.527), ('Rita', 'MON', 6011.20), ('Fred', 'FRI',
      389.12),
   ('Fred', 'THU',  893.45), ('Mark', 'WED', 2900.13), ('Mark', 'MON',
      610.45),
   ('Fred', 'FRI',   45.69), ('Rita', 'FRI', 1092.35), ('Mark', 'MON',
      12.56);
```

1.3.3 Query the table to view the data.

1.3.4 Run a query with PIVOT that will list each employee in a row, and the total sales for each day of the week as columns:

```
SELECT * FROM weekly_sales
PIVOT (SUM(amount)FOR day in ('MON','TUE','WED','THU','FRI'));
```

1.3.5 Suspend and resize the warehouse

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