EAST TENNESSEE STATE UNIVERSITY

CSCI 2020 – Database Fundamentals

Lab 3 – Advanced SQL (Part 1)

Header Comments

Include the standard header comments into your SQL Worksheet in SQL Developer. If you do not enter the header comments, I will take 10% off your grade. Save your file as LastnameFirstinitial_Lab3.sql (e.g., RezwanaT_Lab3.sql).

A. Lesson: Advanced Queries

This portion of the assignment is a built-in lesson that introduces you to SQL concepts explaining how and why certain clauses and keywords are used. To better understand the example queries, copy and paste to your SQL Developer worksheet. To run the query, highlight the SELECT statement and click on “Run Statement” (big green Play button).

Pattern Matches

In our previous queries, we have been searching for records that meet a specific criterion, such as City = ‘Bristol’ or State = ‘NY’, etc. What if, however, we do not know exactly how a particular item is spelled? Instead of searching for an “exact” match, we can look for a “pattern”. For example, to find customers whose first name begins with an ‘R’ you can use the following query:

```
SELECT *
FROM ALLPOWDER.Customer
WHERE firstname LIKE 'R%';
```

Note: The use of the percent (%) sign. This symbol stands for any character or sequence of characters. You can use an underscore (_) to represent a single character.

ORDER BY Clause

Assume you want to sort your results in a particular order. You can use the ORDER BY clause to do this. You can specify whether you want the results to be in ascending order (A-Z) with the “ASC” keyword or in descending order (Z-A) with the “DESC” keyword. **If you leave out the argument, SQL assumes you mean ascending order.**

```
SELECT *
FROM ALLPOWDER.Customer
WHERE State = 'TN'
ORDER BY lastname ASC;
```

You can even sort by more than one field at a time. Because there are multiple cities and countries listed in the Customers table, you could sort first by State in descending order and then by City in ascending order with the following query:

```
SELECT *
FROM ALLPOWDER.Customer
ORDER BY State DESC, City ASC;
```

Note: ASC is not required in the above queries because SQL assumes that if there is no argument specified, it will sort that field in ascending order.

String Manipulation

SQL queries will often require formatting existing data to present them in a specific format to users. In SQL we can use various **functions** to apply formatting to the values in a column. **SQL Functions have a name and accept parameters.** Some parameters are mandatory which means your code will not work if you miss these parameters while some are optional. A few of these functions are described below with examples of queries using the function.

SUBSTR Function

The SUBSTR function returns a portion of a string.

Syntax: SUBSTR (char, position [, substring_length]).

- **char (mandatory):** a base string from which the substring is created. This can be a hard-coded string or a column name where the column accepts character values
- **position (mandatory):** the position at which the first character of the returned string begins.
    - When *position* is positive, the function counts from the beginning of the base *string* to find the first character.
    - When *position* is negative, the function counts backward from the end of base *string*.
- substring_length (optional): length of the returned string. When you do not specify a value for this argument, then the function returns all characters to the end of the base *string*.

```
SELECT SUBSTR('abcdefgh', 5, 3)
FROM DUAL;

SELECT SUBSTR(lastname, 1, 1) || SUBSTR(firstname, 1, 1) AS "Customer Initials"
FROM ALLPOWDER.Customer
```

Note: DUAL is a dummy table that exists in Oracle databases. We use this when we are not querying tables from real databases.

LPAD/RPAD Fucntion

The LPAD function returns an expression, left-padded to a specified length with the specified characters. To right-pad a text expression, use RPAD.

Syntax: LPAD (text-exp , length [, pad-exp]).

- **text-exp (mandatory):** A text expression or column name that you want to pad.
- **length (mandatory):** The total length of the return value as it is displayed on your screen. In most character sets, this is also the number of characters in the return value.
- pad-exp (optional): A text expression that specifies the padding characters. The default value of pad-exp is a single blank.

```
SELECT LPAD('Page 1',15,'*.') "LPAD example"
   FROM DUAL;
```

You can use functions within functions. For example, the query below uses both LPAD, RPAD as well as the LENGTH function in the same statement.

```
SELECT RPAD(LPAD(name, LENGTH(name)+5, '$'), LENGTH(name)+10, '#')
FROM ALLPOWDER.Manufacturer;
```

TO_CHAR Function

The purpose of the Oracle TO_CHAR function is to convert either a number or a date value to a string value.

Syntax: TO_CHAR(input_value, [format_mask], [nls_parameter])

- **input_value (mandatory):** DATE or Number value to convert to String type or column that contains said value.
- format_mask: the format that the input_value should be displayed as. If this is omitted, the function will use a default format, as mentioned below.
- nls_parameter (optional): This value is used to determine how the output value is displayed. For example, when converting a date format or applying a currency format for different countries and regions.

```
SELECT TO_CHAR(SYSDATE, 'YYYY_MM_DD') FROM DUAL;
```

Querying multiple tables

JOINING Tables

In the previous examples we have been querying a single table. However, querying a single table does not always give us the information we need and in order to get the “bigger picture” we need to be able to combine data from multiple tables. This is accomplished by **“joining” tables based on fields that they have in common.**

For example, click on the Department table in the HR schema. Notice that it has the following fields:

```
Department_ID, Department_Name, Manager_ID, Location_ID
```

Now click on the Location table. Notice that it has the following fields in it:

```
Location_ID, Street_Address, Postal_Code, City, State_Province, Country_ID
```

What fields exist in both tables? The Location_ID field. Therefore, we can write a query that will allow us to retrieve data from both tables by joining them using the common field:

```
SELECT * FROM HR.departments
INNER JOIN HR.locations
ON HR.departments.location_ID = HR.locations.location_ID
```

There are other types of joins that we will discuss in the coming week.

Self JOIN

A self join allows you to join a table to itself. Since the query references the same table more than once, **table aliases** are used to assign different temporary names to the same table. One classic example is querying an employee table to get the names of managers since managers are employees too.

```
SELECT emp.first_name || ' ' || emp.last_name "Employee Name",
man.first_name || ' ' || man.last_name "Manager Name"
FROM HR.Employees emp
INNER JOIN HR.Employees man
ON emp.manager_ID = man.employee_ID;
```

Multiple JOINS

A multiple join is a query that contains the same or different join types, which are used more than once**.** Thus, we gain the ability to combine data from multiple tables within one query.

For example, get the first and last name of customers that are beginners in cross-country skating.

```
SELECT cust.firstname, cust.lastname, sk.description, cs.style
FROM ALLPOWDER.Customer cust
INNER JOIN ALLPOWDER.CustomerSkill cs
ON cust.customerID = cs.customerID
INNER JOIN ALLPOWDER.SkillLevel sk
ON cs.skillLevel = sk.skillLevel
WHERE sk.description = 'Beginner' AND cs.style = 'Cross-Country-Skate';
```

B. String Matching [18 Total Points]

For this section, you will be using several tables within the HR schema. The ERD for this schema is on the last page. You have SELECT access to these tables.

1. Write a select statement that returns all records from the HR.Employees table that have a phone number that begins with ‘515’. Insert the statement into your SQL file. Format your answer as follows. (3 points)

```
-----------------------------------
-- B1. Employees from 515 Area Code

<Insert Select Statement Here>
```

2. Write a select statement that returns all records from the HR.Employees table that have a phone number with the digits 121 within the phone number. Format your answer as follows. (3 points)

```
---------------------------------------------
-- B2. Employees with 121 in the phone number

<Insert Select Statement Here>
```

3. Write a select statement that returns all records from the HR.Employees table that have a phone number that ends with the digits 34. Format your answer as follows. (3 points)

```
-------------------------------------------------------
-- B3. Employees with 34 at the end of the phone number

<Insert Select Statement Here>
```

4. Write a select statement that returns all records from the HR.Employees table that have a phone number that matches this pattern: ###.###.####. Use the LIKE keyword. Format your answer as follows. (3 points)

```
--------------------------------------------------------
-- B4. Employees phone number in the format ###.###.####

<Insert Select Statement Here>
```

5. Write a select statement that returns all records from the HR.Employees table that have a phone number that matches this pattern: ###.###.###. Use Oracle's REGEXP_LIKE function. You may need to use Google here to find the right regular expression. Format your answer as follows. (3 points)

```
--------------------------------------------------------
-- B5. Employees phone number in the format ###.###.####

<Insert Select Statement Here>
```

6. Write a select statement that returns all records from the HR.Employees table that have a phone number that matches this pattern: ###.##.####.######. Use Oracle's REGEXP_LIKE function. Format your answer as follows. (3 points)

```
--------------------------------------------------------------
-- B6. Employees phone number in the format ###.##.####.######

<Insert Select Statement Here>
```

C. String Manipulation [18 Total Points]

1. Write a select statement that returns the employee name in the format "Last, First", concatenated together, from the HR.Employees table. You will need to use Oracle's Concatenation operator (||) for this exercise. Format your answer as follows. (3 points)

```
--------------------------------------------------------
-- C1. Employees Last and first names

<Insert Select Statement Here>
```

2. Write a select statement that returns a specially formatted employee ID from the HR.Employees table. Your format should be the first character of the last_name, the first character of the first_name, a hyphen, and the employee_id (E.g. CK-119). You will need to use the Concatenation operator (||) and the SUBSTR function here. Format your answer as follows. (3 points)

```
--------------------------------------------------------
-- C2. Special Employee IDs in the format ZZ-###

<Insert Select Statement Here>
```

3. Write a select statement that uses LPAD and RPAD to surround the HR.Locations table's city field with asterisks. Your total width should allow the city name to both begin and end with an asterisk. Order the list by the length of the city name so the shortest city name appears first (Hint: Use the LENGTH function in the ORDER BY clause). (3 points)

```
--------------------------------------------------------
-- C3. City names surrounded by asterisks

<Insert Select Statement Here>
```

4. Write a select statement that displays the full address of each record in HR.Locations. The first line must include the street_address, and the second line must include the city, state_province postal_code and country_id. Use CHR(13) as a newline character. Indent the second line 5 spaces. To see the results properly, you must use "Run Script" rather than "Run Statement" in SQL Developer. (3 points)

```
--------------------------------------------------------
-- C4. Formatted Location Addresses

<Insert Select Statement Here>
```

5. Write a select statement that displays today's date in the following formats. Your result will be one row with six columns that look like the following examples.

042014	APR-2014	2014/04/08	April 08, 2014	Monday, April 08, 2014	April 08, 2014 05:22:58 PM

To do this, use the TO_CHAR function and the SYSDATE keyword (gives you the current date). You must also SELECT from the DUAL "dummy" table in order to do this properly. (6 points)

```
--------------------------------------------------------
-- C5. Formatted Dates

<Insert Select Statement Here>
```

D. Simple Join Statements [24 Total Points]

1. In the HR database, you can see that HR.Employees relates to zero or one HR.Departments records. To relate the two tables, you must use an INNER JOIN where the Employee's Department ID matches the Department's Department ID. In your SELECT script, display the Employee_ID, Last_Name, First_Name, Phone_Number and the Department_Name. Format your script as follows. (5 points)

```
------------------------------------------------
-- D1. Simple Join of Employees and Departments

<Insert Select Statement Here>
```

2. Copy and paste the statement you created for D1. Alter the statement so that you also display the manager's first name and last name. Note that this requires an INNER JOIN to Employee. Format your script as follows. (5 points)

```
---------------------------------------------------
-- D2. Select with two joins, including a self-join

<Insert Select Statement Here>
```

3. Count the number of records in the HR.Employees table using the COUNT function. Count the number of records in the HR.Departments table. Multiply the two numbers together in an SQL statement, selecting from the DUAL dummy table. Finally, select all columns, performing a CROSS JOIN between the HR.Employees table and the HR.Jobs table. How many records are returned in the CROSS JOIN? (8 points)

```
------------------------------------
-- D3. Cross Join Example
```

```
<Insert Select Statement for HR.Employees COUNT Here>

<Insert Select Statement for HR.Departments COUNT Here>

<Insert Select Statement for Multiplying the Result Here>

<Insert Select Statement for CROSS JOIN here>
```

4. Copy your CROSS JOIN statement and paste it in the location specified below. Add a WHERE clause to the statement that matches the Department's Department ID to the Employee's Department ID. How many records are returned in the CROSS JOIN? How does this compare to the result of D1 and why? Place your answers in the following format. (6 points)

```
----------------------------------------
-- D4. Cross Join as Inner Join Example

<Paste and Alter Select Statement Here>

/*
Number of Records Returned:

How does this compare and why?

*/
```

Submit your file, named LastnameFirstinitial_Lab2.sql, to the Lab 2 Dropbox area on D2L by **Wednesday, 2/9/2022 at 11:59 PM**. **Late Submission Accepted** (please refer to late submission policy in the syllabus if you are not sure what this means).

References

Jesse, G. (2019, October). SQL: An Introduction to SQL Lesson and Hands-On Lab. *Journal of Computing Sciences in Colleges, 35*(October 2019), 143–156.

Oracle. (2008, August). Oracle OLAP DML Reference, 11g Release 1 (11.1). Redwood City, CA, USA. Retrieved February 3, 2021, from https://docs.oracle.com/cd/B28359_01/olap.111/b28126/dml_functions_2.htm#OLADM576