**CSIT 2520 – Lab 7 Practice with Simple Regular Expressions (25 pts.)**

This lab provides practice using regular expressions as described in Chapter 4 of your Price textbook. From the online course, you will also need to download the script that creates movie rental tables (Movie Rental Schema), extract the script from the zip file and run it in SQL Plus or SQL Developer. Review the script file to familiarize yourself with the new tables you’ve created (recommended: create an ERD of the relationship between tables).

**To receive credit: Submit responses in electronic form. Copy this document file to your class disk. Put your name at the top. Do all requested tasks; answer all questions in this document in bold and submit when completed.**

1. Using the Movie Rental Schema, run the following queries.

a.

select phone, regexp\_substr (phone, '([[:digit:]]{3})([[:digit:]]{3})([[:digit:]]{4})')

as "Phone Number" from video\_customer;

b.

select phone, regexp\_substr (phone, '705([[:digit:]]{3})([[:digit:]]{4})')

as "Phone Number" from video\_customer;

What is the difference in the result returned in b. versus a. (be very precise)?

**Query a is getting any first 3 digits for the first 3 numbers while query b is getting only digits 705 for the first 3 numbers. However both queries are neglecting the formatting for the first 4 rows returned. These rows have parentheses around the first 3 digits and a space afterwards and then a hyphen after the next the digits. The parentheses, space, and hyphen are not digits and therefore are not registered by the regexp\_substr. Also, neither one of the queries is doing anything about nulls.**

c. Now try

select regexp\_replace

(phone, '([[:digit:]]{3})([[:digit:]]{3})([[:digit:]]{4})', '(\1) (\2)-(\3)')

as "Modifed Phone#"

from video\_customer;

Look at the end of the second line of this query. Which clause in the statement caused the replacement to occur?

**'(\1) (\2)-(\3)'**

Did this replacement change the address table in any way? Why or why not?

No because regexp\_replace just replaces the returned data with a different format. It is not actually replacing any data.

Why do the first 4 phone numbers look different from the other modified phone numbers?

Because they are formatted differently originally. The regexp picks them up as strings because of the parentheses, space, and hyphen, and not numbers like in the 7 rows which contain only numbers.

d. Modify query c. to replace the middle three numbers with the prefix 694.

Paste your query and results here. (hint: focus on the \2 part of the last regexp\_replace argument)

> select regexp\_replace

(phone, '([[:digit:]]{3})([[:digit:]]{3})([[:digit:]]{4})', '(\1) (694)-(\3)')

as "Modifed Phone#"

from video\_customer

Modifed Phone#

----------------------------------------------------------------(202) 234-3223

(202) 111-2222

(202) 222-1111

(202) 212-1212

(705) (694)-(7712)

(703) (694)-(3725)

(703) (694)-(7854)

(705) (694)-(5712)

(703) (694)-(3324)

(703) (694)-(2154)

(703) (694)-(9856)

14 rows selected

2. Now try the following:

a.

select phone from video\_customer

where regexp\_instr(Phone,'7',4,2) > 0;

Explain what this query has done for us?

It's getting the position where the pattern '7' occurs twice in the PHONE column starting at position 4 and if that's greater than 0 it will display that phone number.

b. This next query builds on example 2a. Try it.

select regexp\_replace(Phone, '7','#',4,2) "Match Test"

from video\_customer

where regexp\_instr(Phone, '7',4,2) > 0;

Paste a copy of your query results here. Explain what you are seeing in the results of this query?

> select regexp\_replace(Phone, '7','#',4,2) "Match Test"

from video\_customer

where regexp\_instr(Phone, '7',4,2) > 0

Match Test

----------------------------------------------------------------------

7053567#12

7037213#25

Regexp\_replace is finding the number 7 in the Phone column starting at position 4 and replacing it with the # sign. It is also looking to make sure that there are 2 occurences of ‘7’. It seems to be placing the pound sign at the last occurrence of ‘7’ but I’m not sure how to explain why.

3. **Use the REGEXPR\_LIKE function** to perform the following queries. Paste your working query under each specification.

1. Display the last name and phone number of all video rental customers that have an area code of ‘202’. Be sure your query won’t accidentally change the value 202 if it appears in the middle of the phone number.

> SELECT last\_name, first\_name, phone

FROM video\_customer

WHERE REGEXP\_LIKE (phone, '(202)')

LAST\_NAME FIRST\_NAME PHONE

------------------------------ ------------------------------ --------------

Moore John (202) 234-3223

Campbell Karen (202) 111-2222

Smith Robert (202) 222-1111

Richard Joanne (202) 212-1212

I’m not sure if this is how you intended this to be done but I didn’t really know how to get if to just match ‘202’ starting at position 2 using REGEXP\_LIKE.

1. Display the last name, first name and phone number of all video rental customers who have the string ‘ON’ in their last name. Use a case insensitive match.

> SELECT last\_name, first\_name, phone

FROM video\_customer

WHERE REGEXP\_LIKE (last\_name, 'on', 'i')

LAST\_NAME FIRST\_NAME PHONE

------------------------------ ------------------------------ --------------

Milton Henry 7037213725

Jackson Cindy 7038547854

Johnson Michelle 7038659856

1. Referring to the example in the middle of page 123 of your Price text (REGEXP\_LIKE function example using dates), display the customer id, full name and date of birth for all video customers born between 1964 and 1978.

> SELECT id, last\_name, first\_name, dob

FROM video\_customer

WHERE REGEXP\_LIKE (TO\_CHAR(dob, 'YYYY'), '^19[6-7][0-9]$')

ID LAST\_NAME FIRST\_NAME DOB

---------- ------------------------------ ------------------------------ ---------

10 Moore John 12-MAR-65

30 Smith Robert 31-MAY-66

40 Richard Joanne 17-FEB-60

230 Andrews Rebecca 12-MAR-69

329 Jeffrey Denning 15-MAR-64

330 Mayer Arthur 11-JUN-73

331 Baker Paul 19-DEC-69

332 Johnson Michelle 21-MAR-66

8 rows selected