**Name: Aaron Lim**

SQL Programming – Level 1 Programming Project 04

# Regular Expressions | Functions

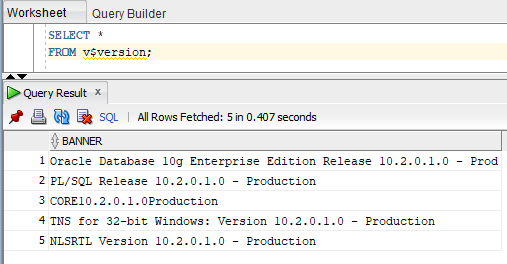
***Reminder: read the Project Guidelines document for instructions on how to format and submit your assignments.***

## Part 1 – use the Oracle 10g server for the following questions.

Demonstrate that you are using the Oracle 10gi server by issuing:

SELECT \*

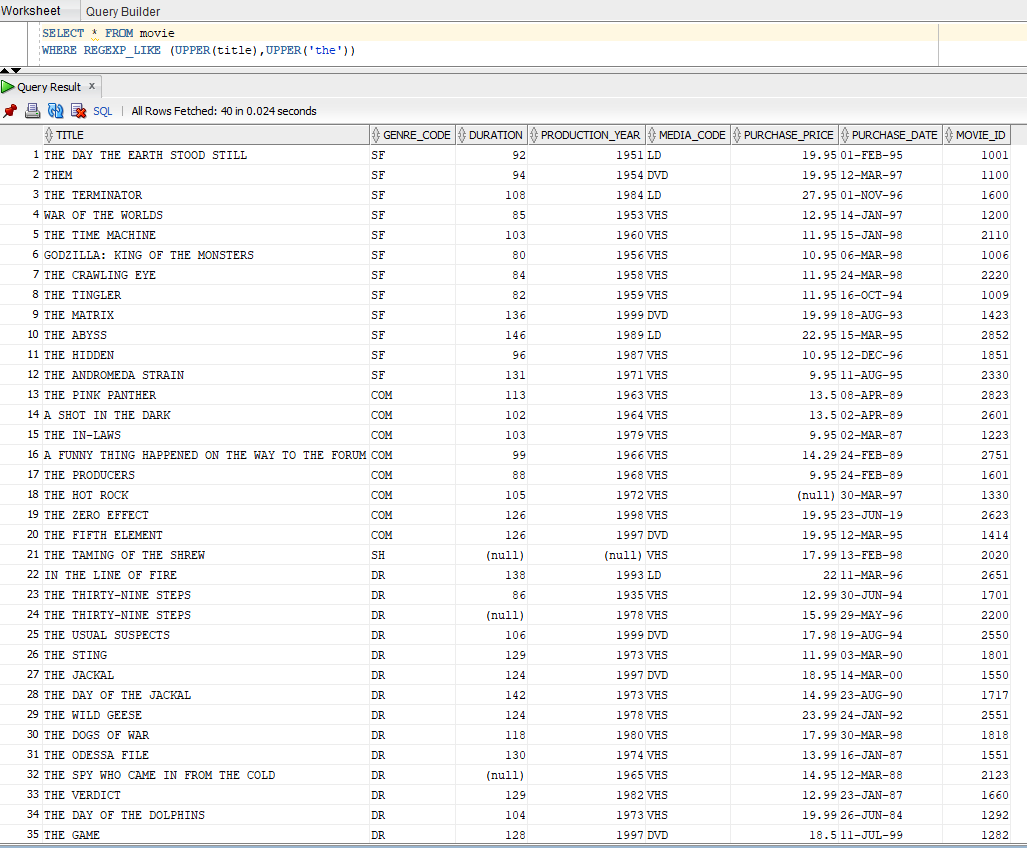
FROM v$version;

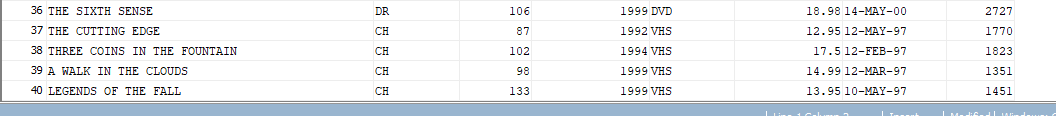


1. Use the regular expression feature of Oracle to find all movies whose title includes the **word** ‘the’.

SELECT \* FROM movie

WHERE REGEXP\_LIKE (UPPER(title),UPPER('the'))

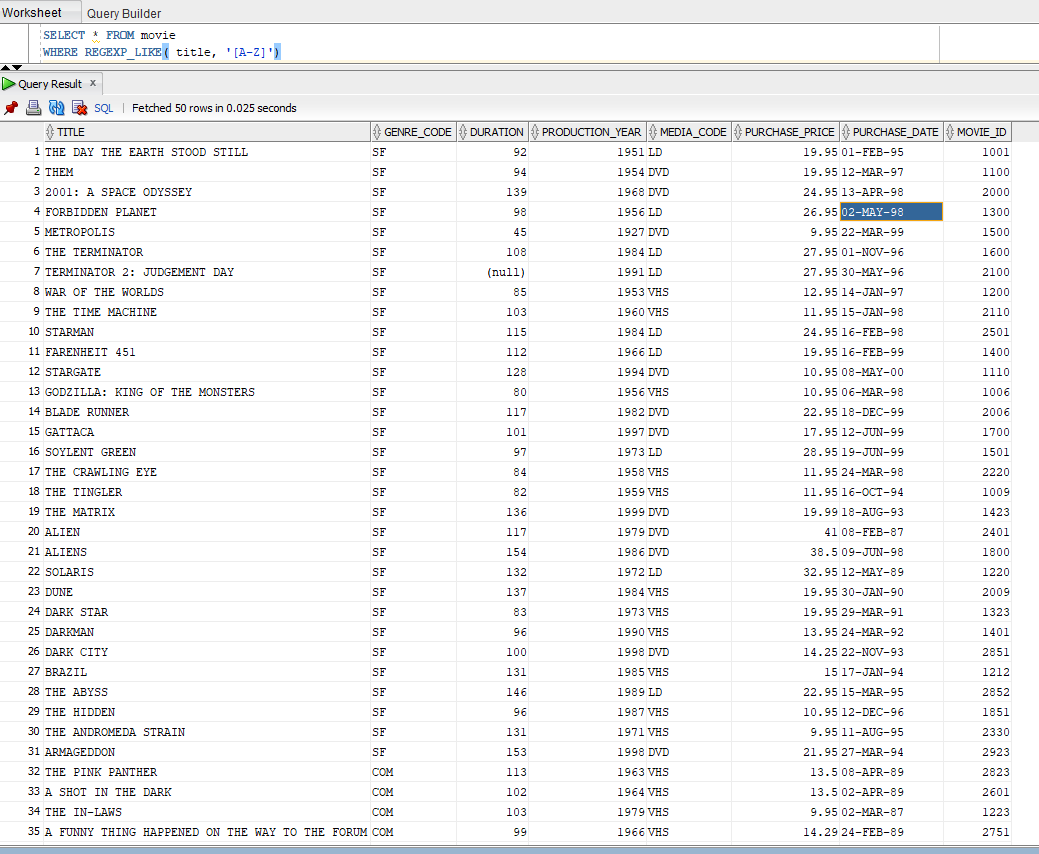


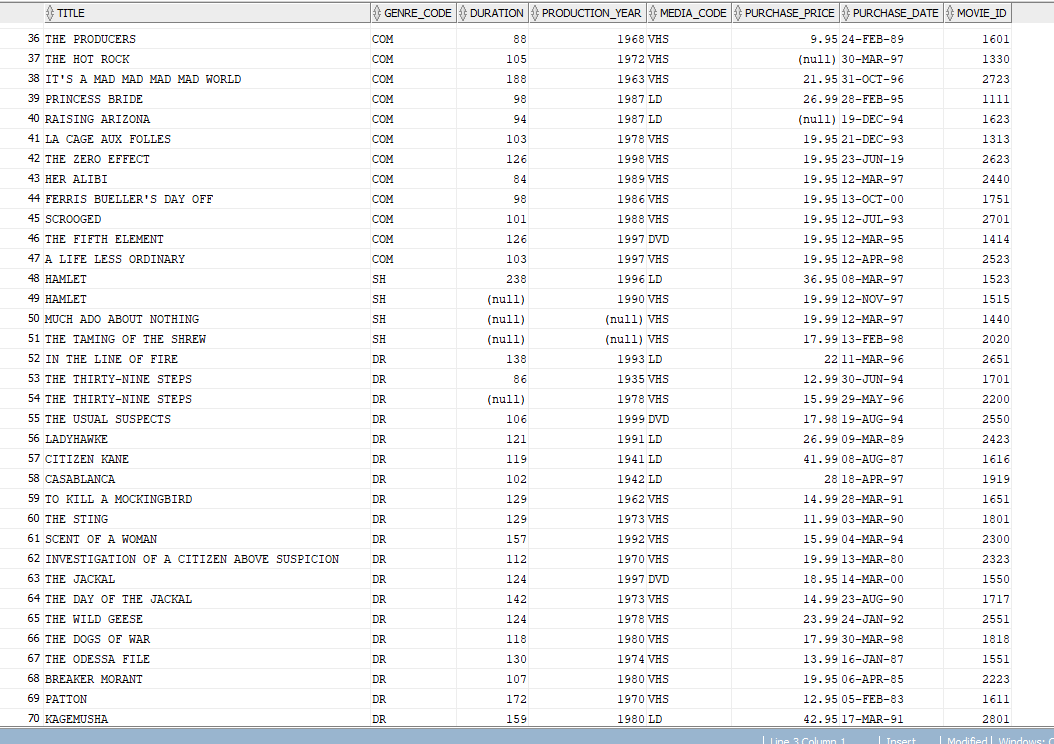


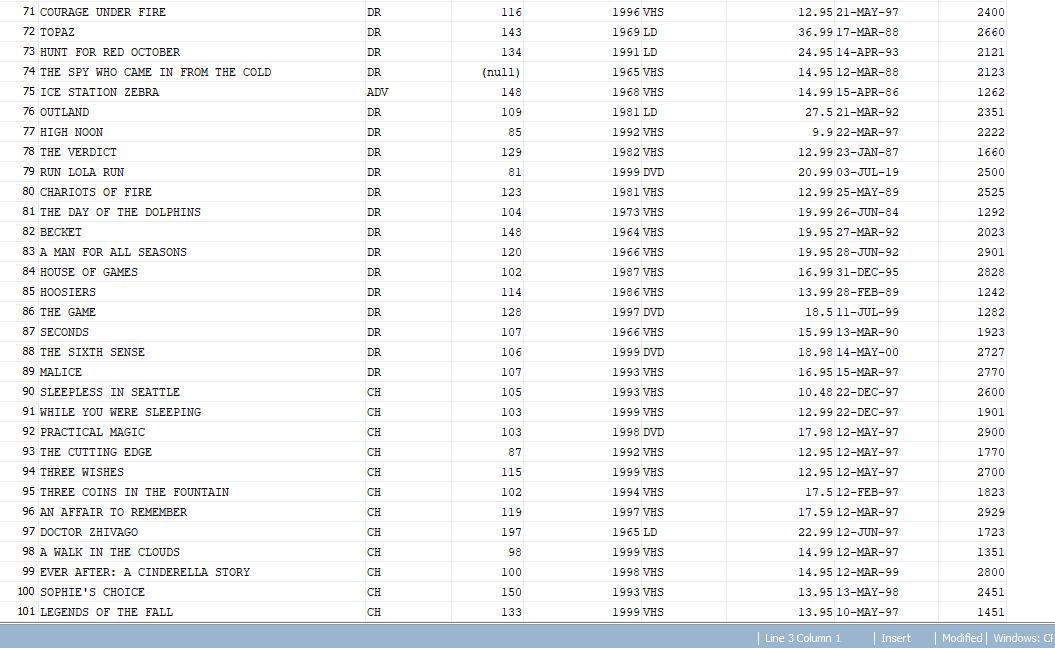
1. Use the regular expression feature of Oracle to find all movies whose title includes a single word.

SELECT \* FROM movie

WHERE REGEXP\_LIKE( title, '[A-Z]')



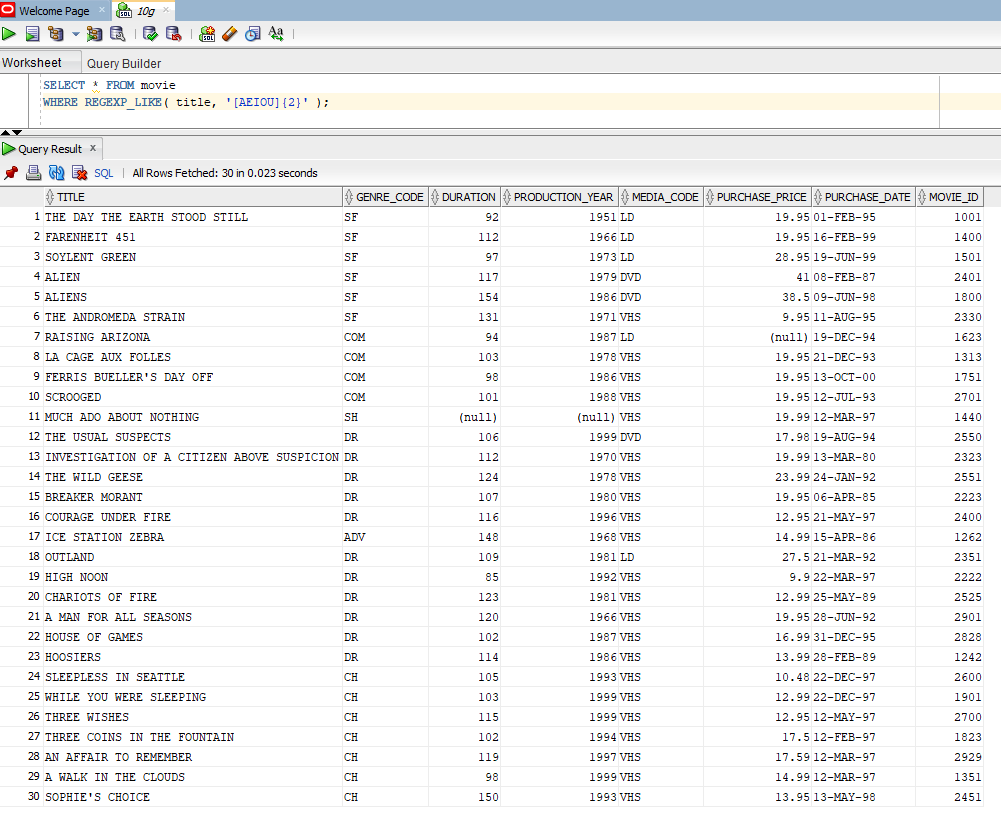




1. Use the regular expression feature of Oracle to find all movies whose title includes a word with a double vowel sequence, For example: ‘ie’, ‘ee’, ‘oa’ ….

SELECT \* FROM movie

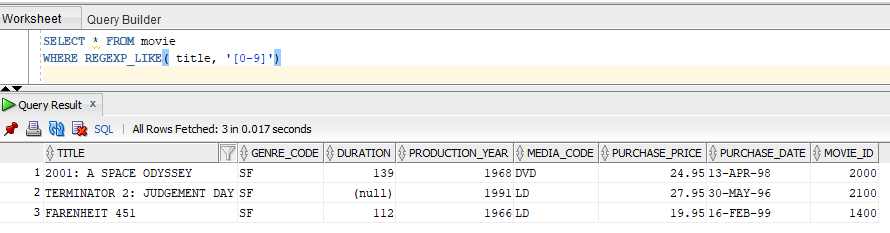
WHERE REGEXP\_LIKE( title, '[AEIOU]{2}' );



1. Use the regular expression feature of Oracle to find all movies that include a number (ie. any numeric digit) as part of the title.

SELECT \* FROM movie

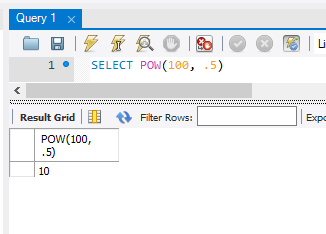
WHERE REGEXP\_LIKE( title, '[0-9]')



## Part 2 – use the MySQL server for the remaining questions.

1. Use the POW function to calculate the square root of 100 (ie. Raised to the ½ power)

SELECT POW(100, .5)



1. Use an appropriate MySQL string function to display the title and the single leftmost character of the genre for all of the movies in the collection.

SELECT title, LEFT(genre, 1)

FROM movies



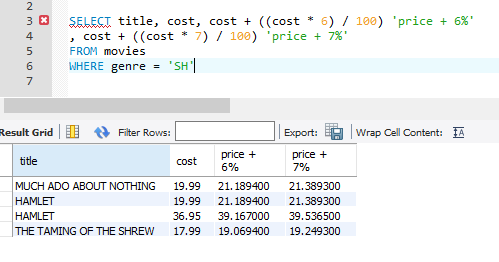


1. For all of the Shakespeare titles in the collection, show the title, purchase price, and price + 6% and price + 7%. Use an appropriate title for the calculated columns. [4 cols]

SELECT title, cost, cost + ((cost \* 6) / 100) 'price + 6%', cost + ((cost \* 7) / 100) 'price + 7%'

FROM movies

WHERE genre = 'SH



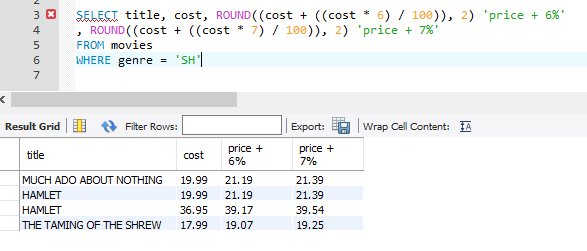
1. For all of the Shakespeare titles in the collection, show the title, purchase price, and price + 6% and price + 7%. Use an appropriate title for the calculated columns. Round all calculations to two decimal places. [4 cols]

SELECT title, cost, ROUND((cost + ((cost \* 6) / 100)), 2) 'price + 6%'

, ROUND((cost + ((cost \* 7) / 100)), 2) 'price + 7%'

FROM movies

WHERE genre = 'SH'

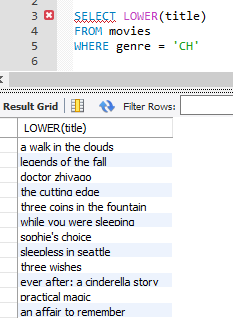


1. For each of the CH films (genre=CH), show the title in all lower case characters.

SELECT LOWER(title)

FROM movies

WHERE genre = 'CH'

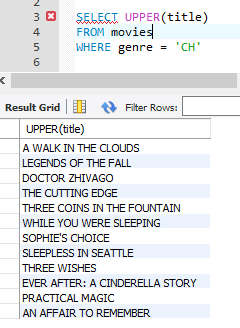


1. For each of the CH films (genre=CH), display the title in all upper case characters.

SELECT UPPER(title)

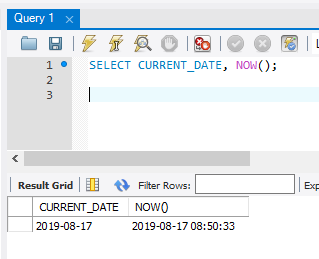
FROM movies

WHERE genre = 'CH'



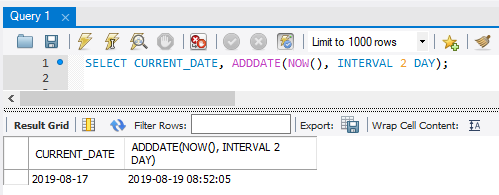
1. Show today’s date, as well as the timestamp for right now (right now being whenever it is you’re running your SQL program). [2 cols]

SELECT CURRENT\_DATE, NOW();



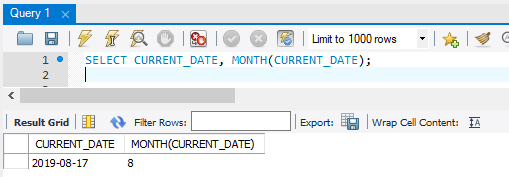
1. Show today’s date, as well as two days from now (use the ADDDATE function). [2 cols]

SELECT CURRENT\_DATE, ADDDATE(NOW(), INTERVAL 2 DAY);



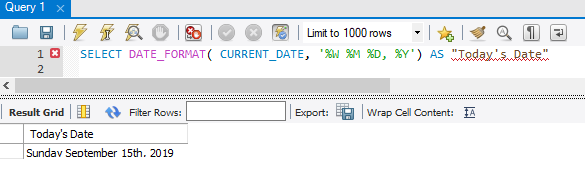
1. Display today’s date and the two digit representation of the month (For example, if today were April 13, 2005, the digits for the month would be ‘04’). [2 cols]

SELECT CURRENT\_DATE, MONTH(CURRENT\_DATE);



1. Display today’s date, and the name of the day, the name of the month, the day of the month (in digits), a comma, and the four digits of the year.

SELECT DATE\_FORMAT( CURRENT\_DATE, '%W %M %D, %Y') AS "Today's Date"



1. Display today’s date and the day of the year it is. [2 cols]

SELECT CURRENT\_DATE, DAYOFYEAR(CURRENT\_DATE)

