ITC 5104 DATABASE DESIGN AND SQL

Lecture 7

Chapter 10 Oracle 12c SQL

Selected Single-Row Functions

OBJECTIVES

- Use the UPPER, LOWER, and INITCAP functions to change the case of field values and character strings
- Manipulate character substrings with the SUBSTR and INSTR functions
- Determine the length of a character string using the LENGTH function
- Use the LPAD and RPAD functions to pad a string to a certain width
- Use the LTRIM and RTRIM functions to remove specific character strings
- Substitute character string values with the REPLACE
- Round and truncate numeric data, using the ROUND and TRUNC functions

OBJECTIVES

- Combine character strings with the CONCAT function
- Round and truncate numeric and date data with the ROUND and TRUNC functions
- Return only the remainder of a division operation with the MOD function
- Use the ABS function to set numeric values as positive
- Use the POWER function to raise a number to a specified power
- Manipulate date data with the ADD_MONTHS, NEXT_DAY, LAST_DAY, and TO_DATE functions
- Differentiate between the CURRENT_DATE and SYSDTAE functions
- Calculate the number of months between two dates using the MONTHS_BETWEEN function

OBJECTIVES

- Identify and correct problems associated with calculations involving NULL values using the NVL function
- Manipulate NULL values with the NVL2 function
- Display dates and numbers in a specific format using the TO_CHAR function
- Perform condition processing similar to an IF statement with the DECODE function and the CASE expression
- Convert string values to numeric with the TO_NUMBER function
- Use the DUAL table to test functions

A FUNCTION

- A function is a predefined block of code that accepts one or more arguments
- An argument is a value that is listed within parentheses
- A function returns a single value as output
- Single-row functions return one row of results for each record processed
- Multiple-row functions return only one result per group or category of rows processed, such as counting the number of books published by each publisher (Chapter 11)

FUNCTIONS

- Different categories of single-row functions:
  - Case conversion functions
  - Character manipulation functions
  - Numeric functions
  - Date functions
  - Other functions

CASE CONVERSION FUNCTIONS

- You can use character functions to change the case of characters, for example
  - Upper to lower
  - Lower to upper
- Used most of the time by developers, rarely by database administrators
- Case conversion functions alter the case of data stored in a field
- It is only temporary, it does not affect how data are stored, only how data are viewed during the execution of a query
- There are three functions supported by Oracle 12c for this purpose, LOWER, UPPER and INITCAP

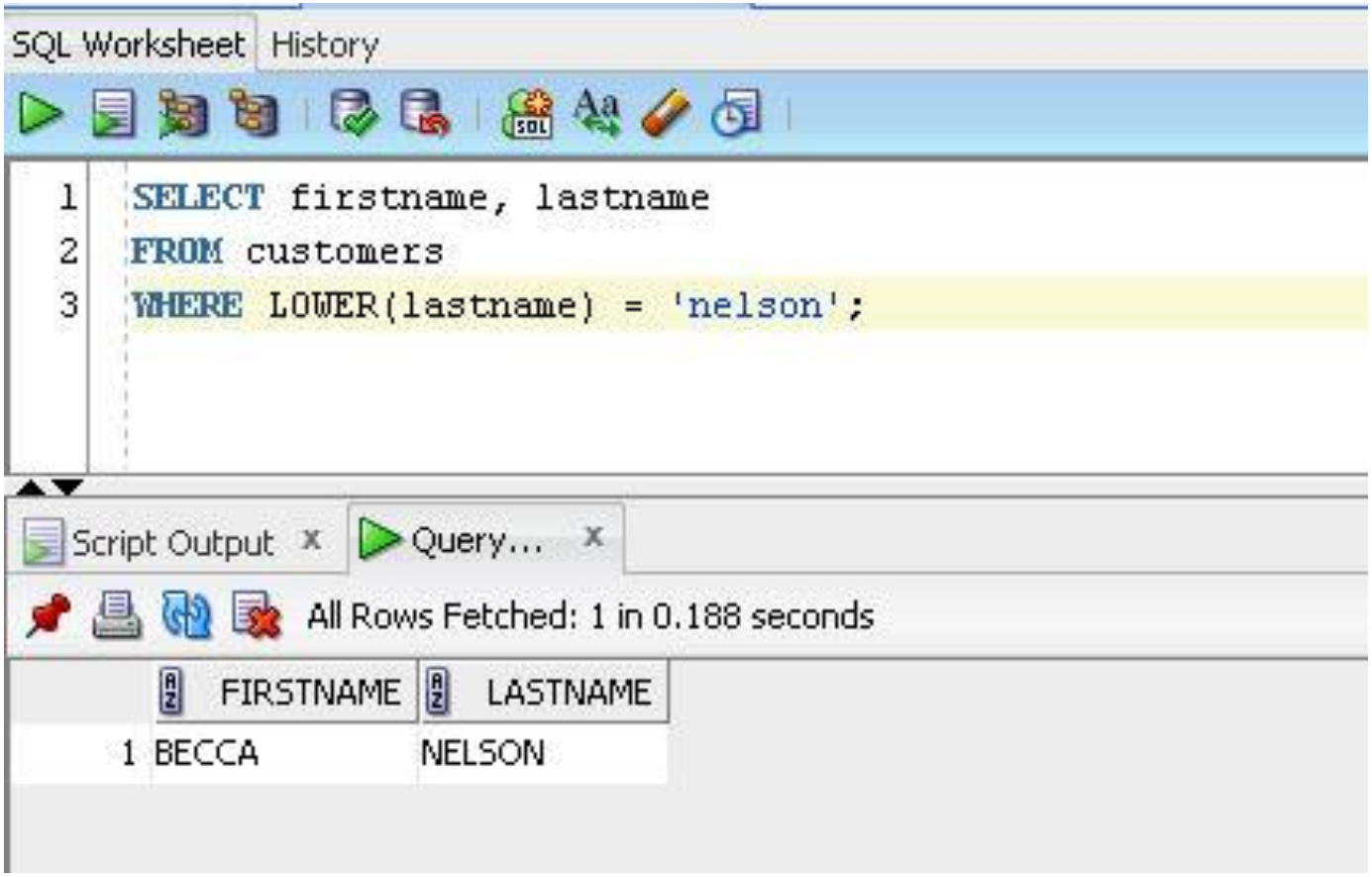
CASE CONVERSION FUNCTIONS

- These are used to change the case of characters
- These alter the case of character stings
- They are used in a query to display data in a different case but they do not affect how he data is stored
- They can be used with an INSERT statement to set the case of a stored value
- There are the following functions:
  - LOWER
  - UPPER
  - INITCAP

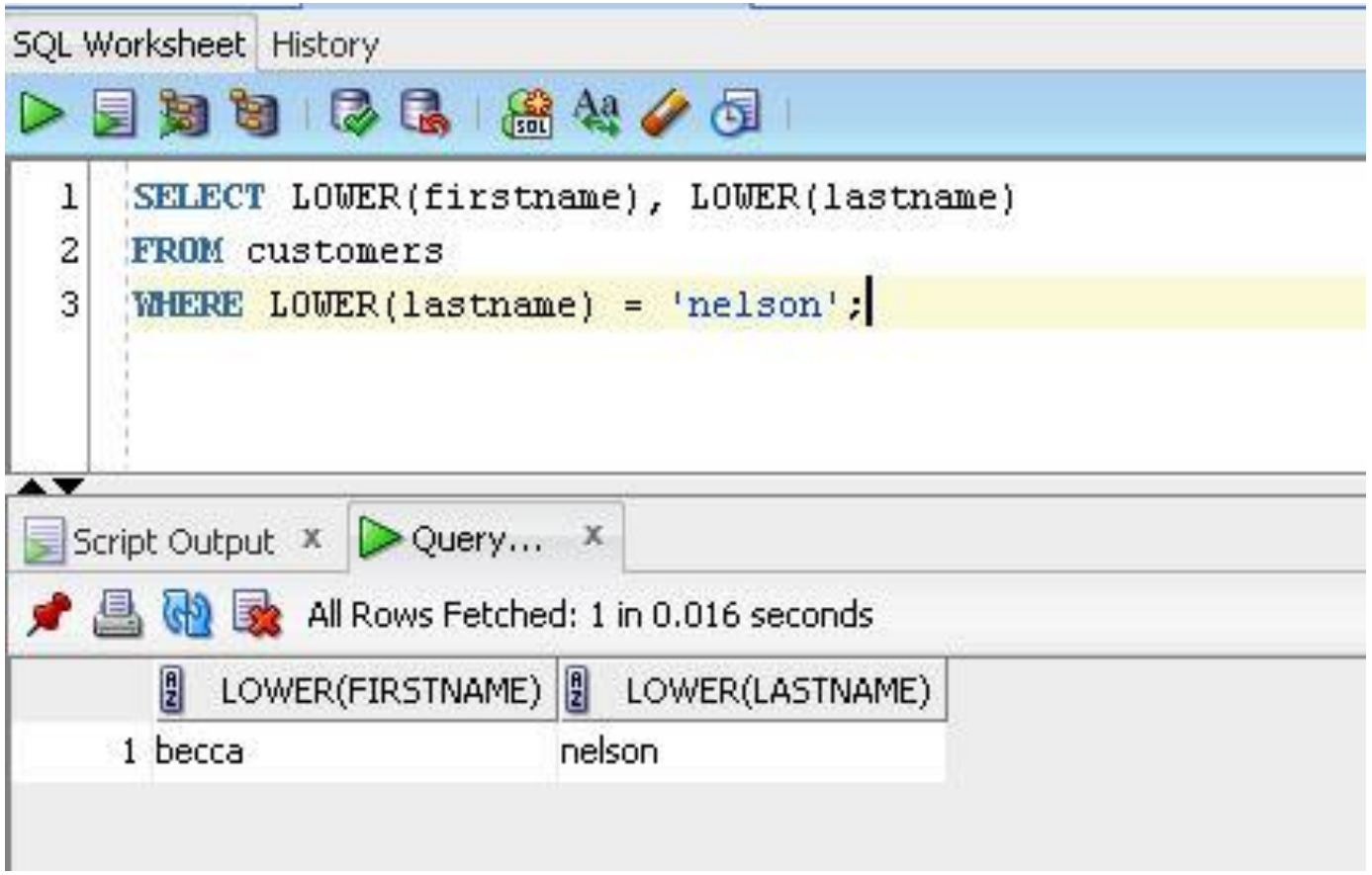
LOWER FUNCTION

- When the LOWER function is used in the SELECT clause, it affects only how the data in the results will be displayed
- When the LOWER function is used in the WHERE clause, it is used for the specific comparison operation
  - Just Lee Books has data stored in upper case
  - LOWER(c), where (c) is the field or string to convert

LOWER FUNCTION



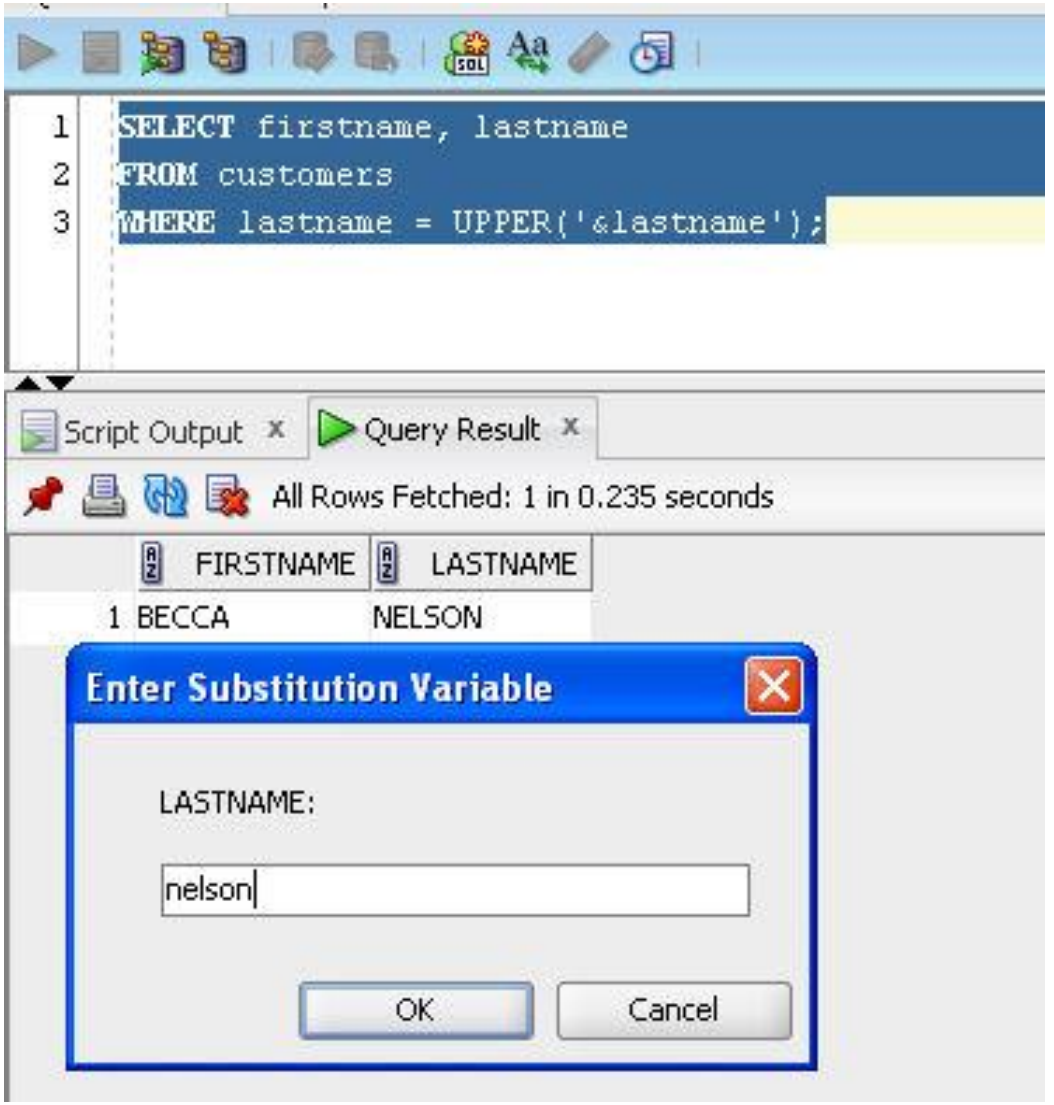
LOWER FUNCTION



UPPER FUNCTION

- This is another function used to make searches more "user friendly"
- The UPPER function converts the given string to *upper-case*
- The syntax is:
  - UPPER(c)
- Again, (c) is the character string to be converted to *upper-case*

UPPER FUNCTION



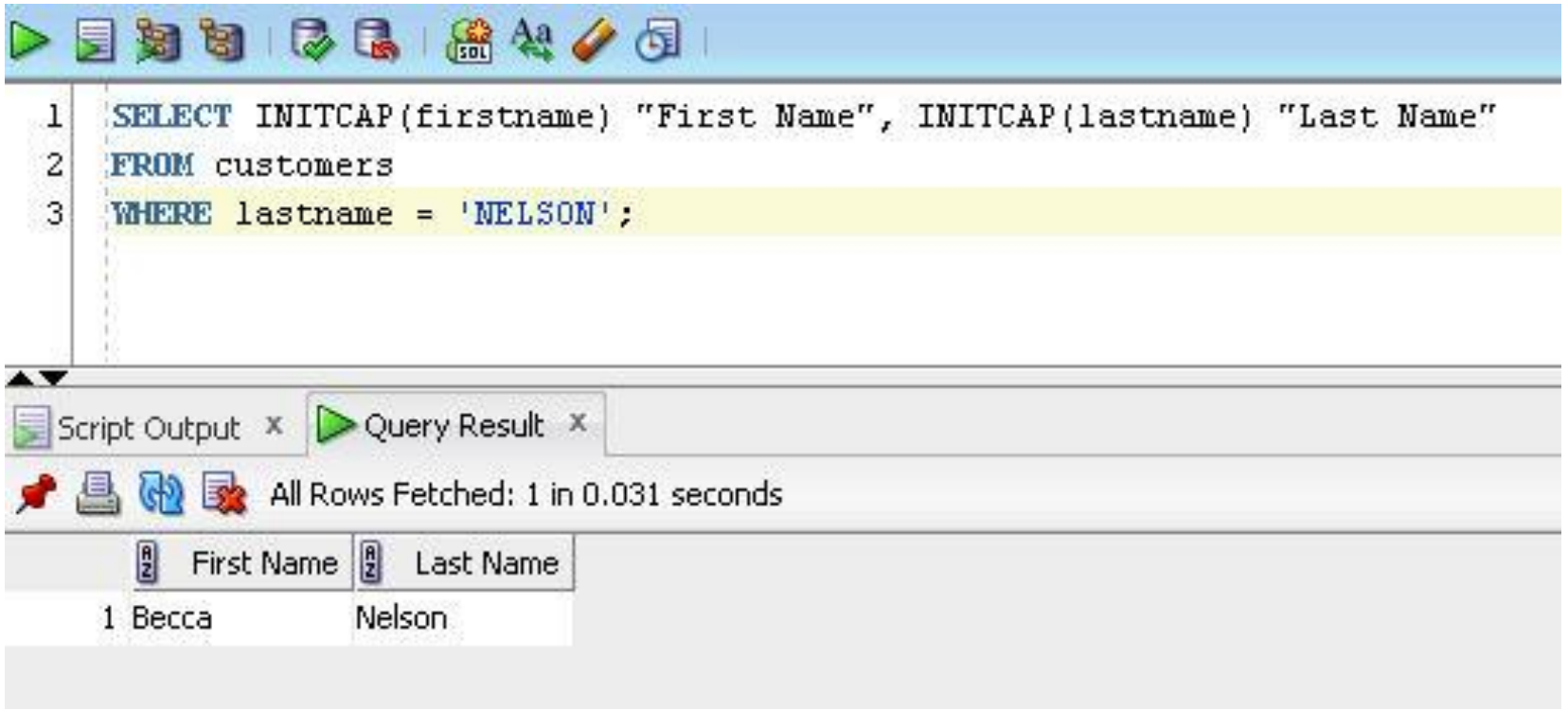
UPPER FUNCTION

- The UPPER function is included with the condition portion of the WHERE clause.
- It instructs Oracle 12c to convert the user-entered character string of 'nelson' to *upper-case* characters when the SELECT statement is executed
- It is more efficient to convert the search condition to the same case as the data stored in the table for the following reasons:
  - You don't need to worry whether the user knows the exact case to use when entering search strings
  - Oracle 12c does not need to convert the case of all the data contained within the field during execution of the query, thus reducing the processing burden placed on the Oracle server

INITCAP FUNCTION

- Having the table data and the search criteria in the same case enables you to find the records you seek. But, the output may not be presented in a very appealing way
- It is generally easier for people to read displayed data when it is presented in *mixed-case* letters rather than in all *upper-case* or all *lower-case* letters
- Oracle 12c uses the INITCAP function to convert character strings to mixed-case, placing a capital letter at the beginning of each word, and the balance of the word in *lower-case*
- Syntax is:
  - INITCAP(c)
- Where (c) is the string to be converted

INITCAP FUNCTION



CHARACTER MANIPULATION FUNCTIONS

- There may be times when you need to manipulate a character string in order to yield the desired query output
- You may need to:
  - Know the length of a string
  - Extract portions of a string
  - Reposition a string
- These tasks are performed by using the manipulation functions

SUBSTR FUNCTION

- The SUBSTR function is used to return a *substring* or a portion of a string
- This is an excellent way to extract portions of a string that has been created on fields where a coding scheme has been implemented
- For example, in the American Zip Code, the first three numbers indicate a geographical distribution area within each state
- The SUBSTR function could be used to extract the first three digits of the zip code stored for each customer

SUBSTR FUNCTION

- The syntax for this function is:
  - SUBSTR(c, p, l)
    - Where C represents the character string
    - P represents the starting location for extraction (the first character is at position 1)
    - L represents the length of the string to return in the results
- The DISTINCT keyword is used to eliminate duplication
- On the next slide, the arguments *(zip, 1, 3)* instruct Oracle 12c to use the zip field starting with the first digit and to extract the first 3 characters and return them in the results

SUBSTR FUNCTION

Worksheet | Query Builder

```
SELECT DISTINCT SUBSTR(zip, 1, 3), zip
FROM customers;
```

Query Result

SQL | All Rows Fetched: 20 in 0.109 seconds

	SUBSTR(ZIP,1,3)	ZIP
1	904	90404
2	787	78710
3	079	07962
4	981	98115
5	915	91510
6	323	32306
7	824	82414
8	331	33111
9	915	91508
10	312	31206
11	122	12211
12	820	82003
13	303	30314
14	606	60606
15	490	49006
16	323	32328
17	837	83707
18	021	02110
19	347	34711
20	086	08607

SUBSTR FUNCTION

- The SUBSTR function can also be used to extract *substrings* from the end of the data stored in the field
- For example, if a negative 3 (-3) were used to indicate the starting position, the software would count backwards (from the end of the string) three positions, then extract the required number of characters
- Again, *column aliases* could be used to make the results more readable

SUBSTR FUNCTION

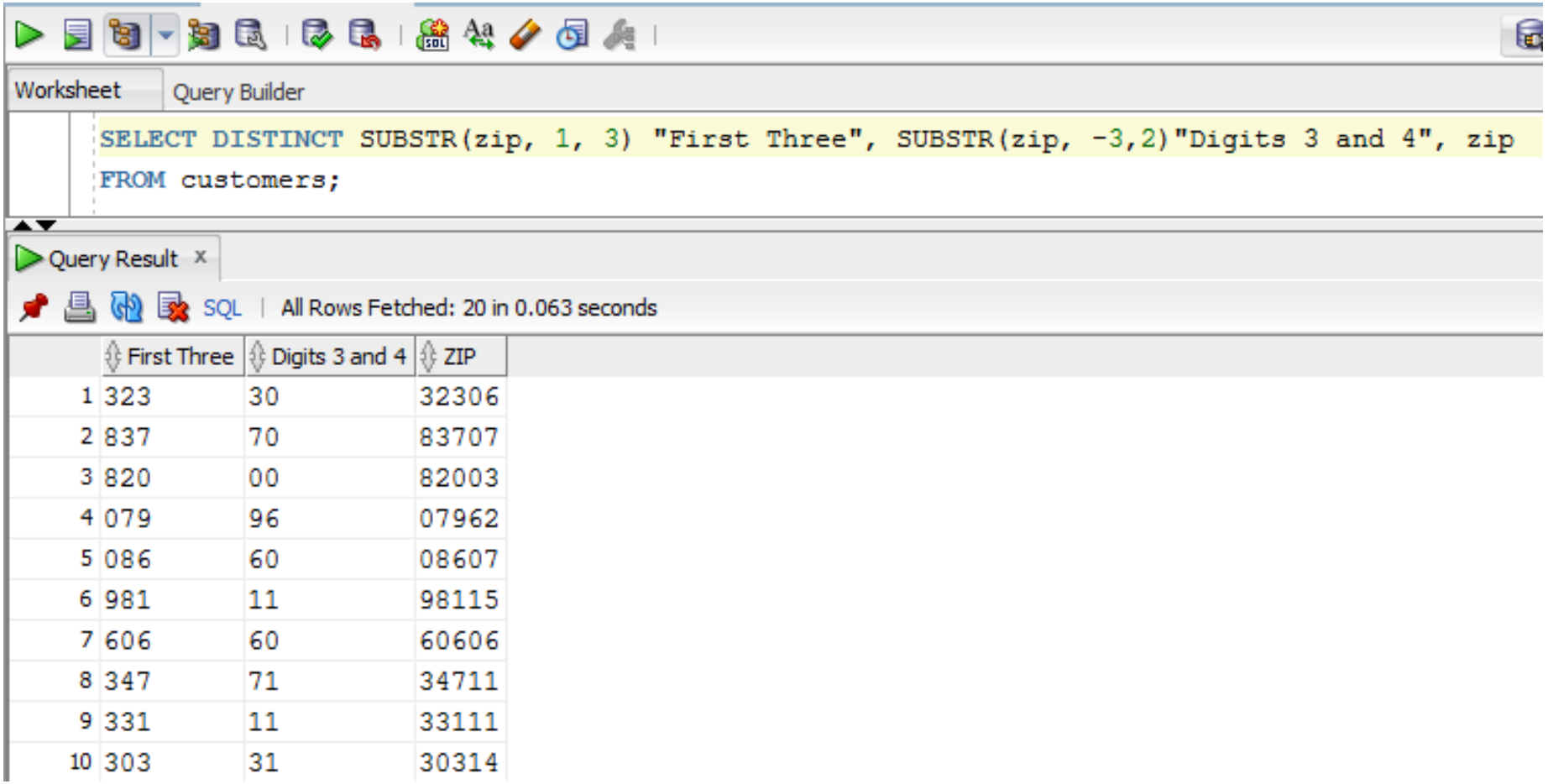
```
1  SELECT DISTINCT SUBSTR(zip, 1, 3)
2  FROM customers;
```

Script Output | Query...

All Rows Fetched: 18 in 0.015 seconds

	SUBSTR(ZIP,1,3)
1	331
2	079
3	323
4	837
5	122
6	312
7	904
8	820
9	915
10	490
11	787
12	303
13	347
14	606
15	021
16	824
17	086
18	981

SUBSTR FUNCTION



Worksheet | Query Builder

```
SELECT DISTINCT SUBSTR(zip, 1, 3) "First Three", SUBSTR(zip, -3,2)"Digits 3 and 4", zip
FROM customers;
```

Query Result

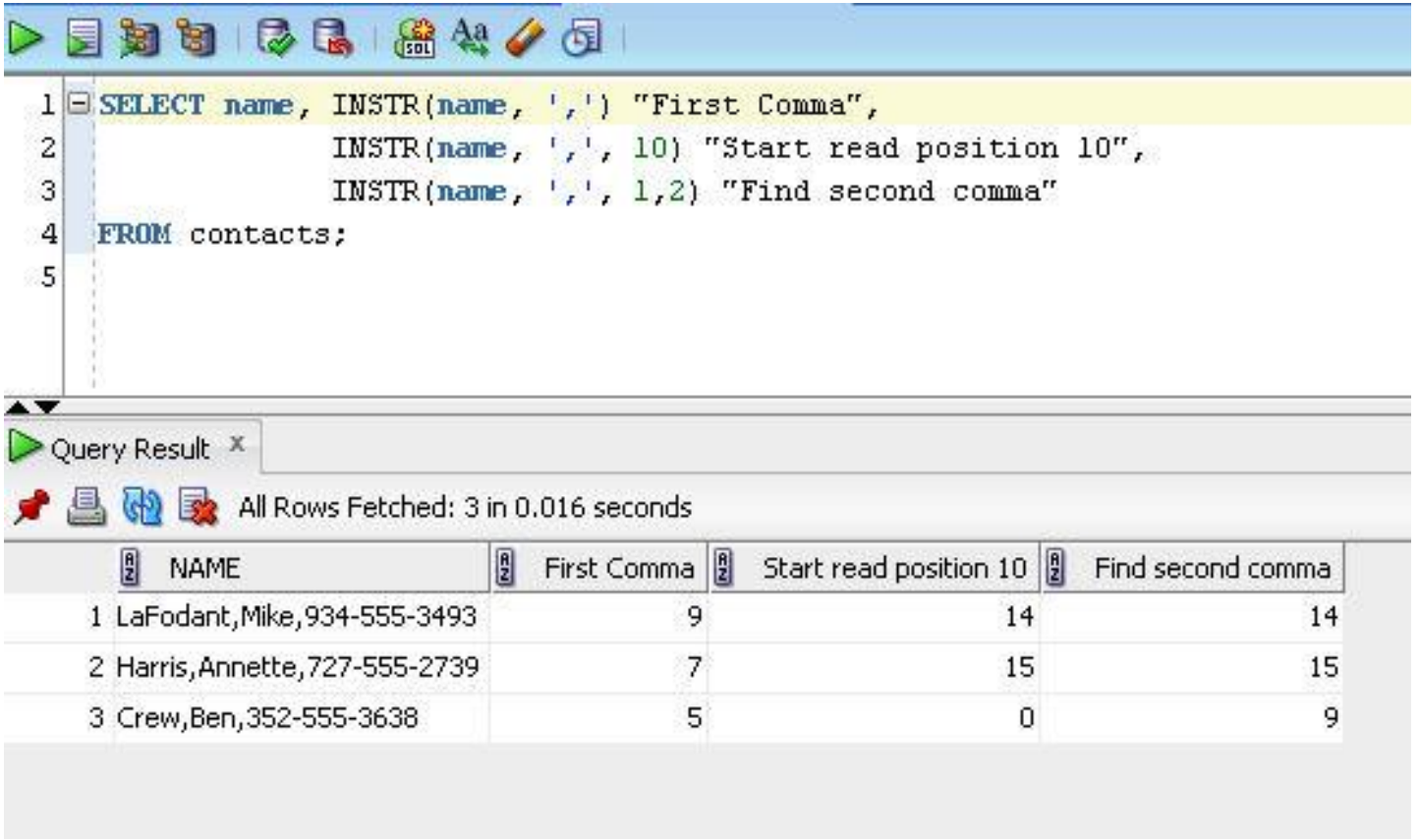
SQL | All Rows Fetched: 20 in 0.063 seconds

	First Three	Digits 3 and 4	ZIP
1	323	30	32306
2	837	70	83707
3	820	00	82003
4	079	96	07962
5	086	60	08607
6	981	11	98115
7	606	60	60606
8	347	71	34711
9	331	11	33111
10	303	31	30314

INSTR FUNCTION

- The INSTR function searches a string for a specified set of characters or a substring, hen returns a numeric value representing the first character position in which the substring is found
- If the substring does not exist a 0 (zero) is returned
- Two arguments are provided to the INSTR function, the string that is to be searched, and the characters or substring enclosed in quotes to locate
- Two optional arguments are also available, the start position, indicating which character of he string value the search should begin with, and occurrence, which is the instance of the search value to locate, is it the first occurrence, or the second occurrence and so on
- By default is it starts at the beginning of string value, and the beginning of the first occurrence is located

INSTR FUNCTION



```
1  SELECT name, INSTR(name, ',') "First Comma",
2               INSTR(name, ',', 10) "Start read position 10",
3               INSTR(name, ',', 1,2) "Find second comma"
4  FROM contacts;
5
```

Query Result

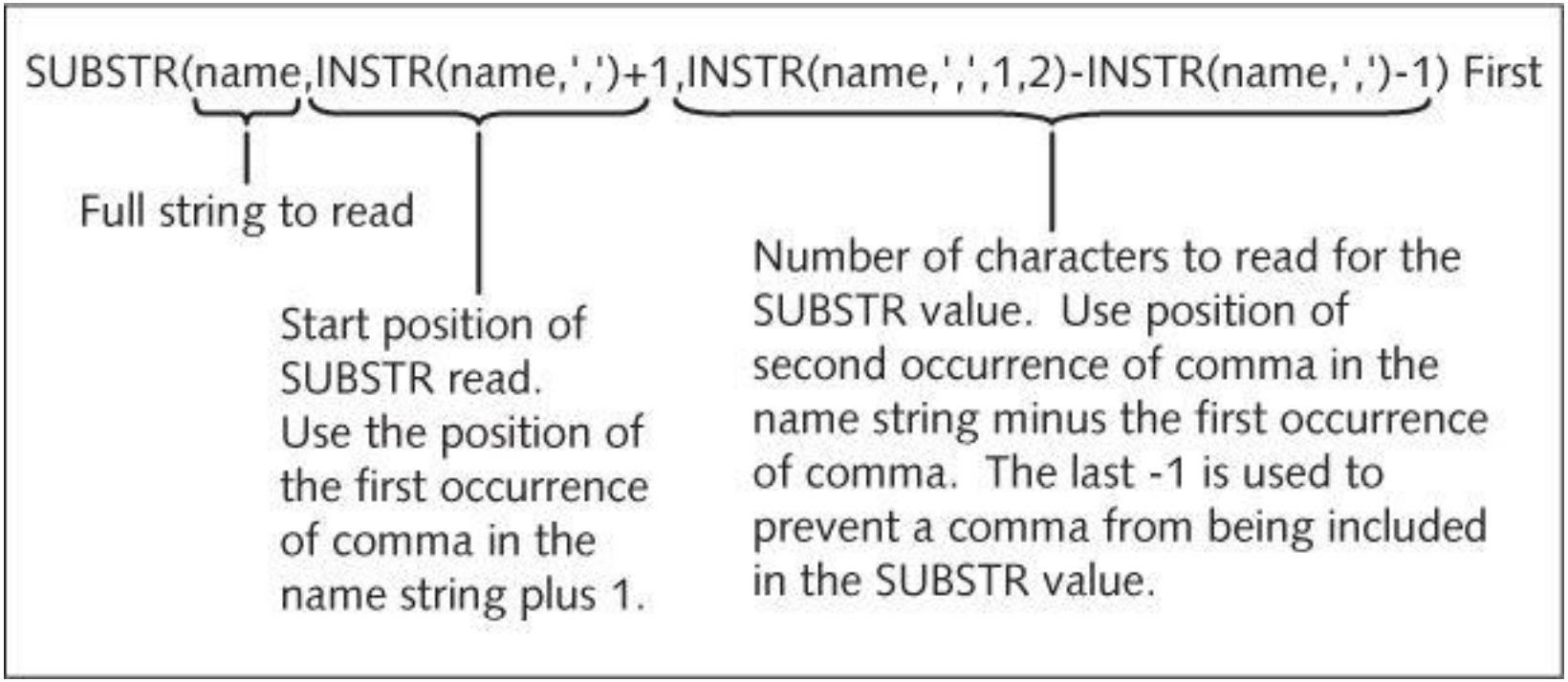
All Rows Fetched: 3 in 0.016 seconds

	NAME	First Comma	Start read position 10	Find second comma
1	LaFodant,Mike,934-555-3493	9	14	14
2	Harris,Annette,727-555-2739	7	15	15
3	Crew,Ben,352-555-3638	5	0	9

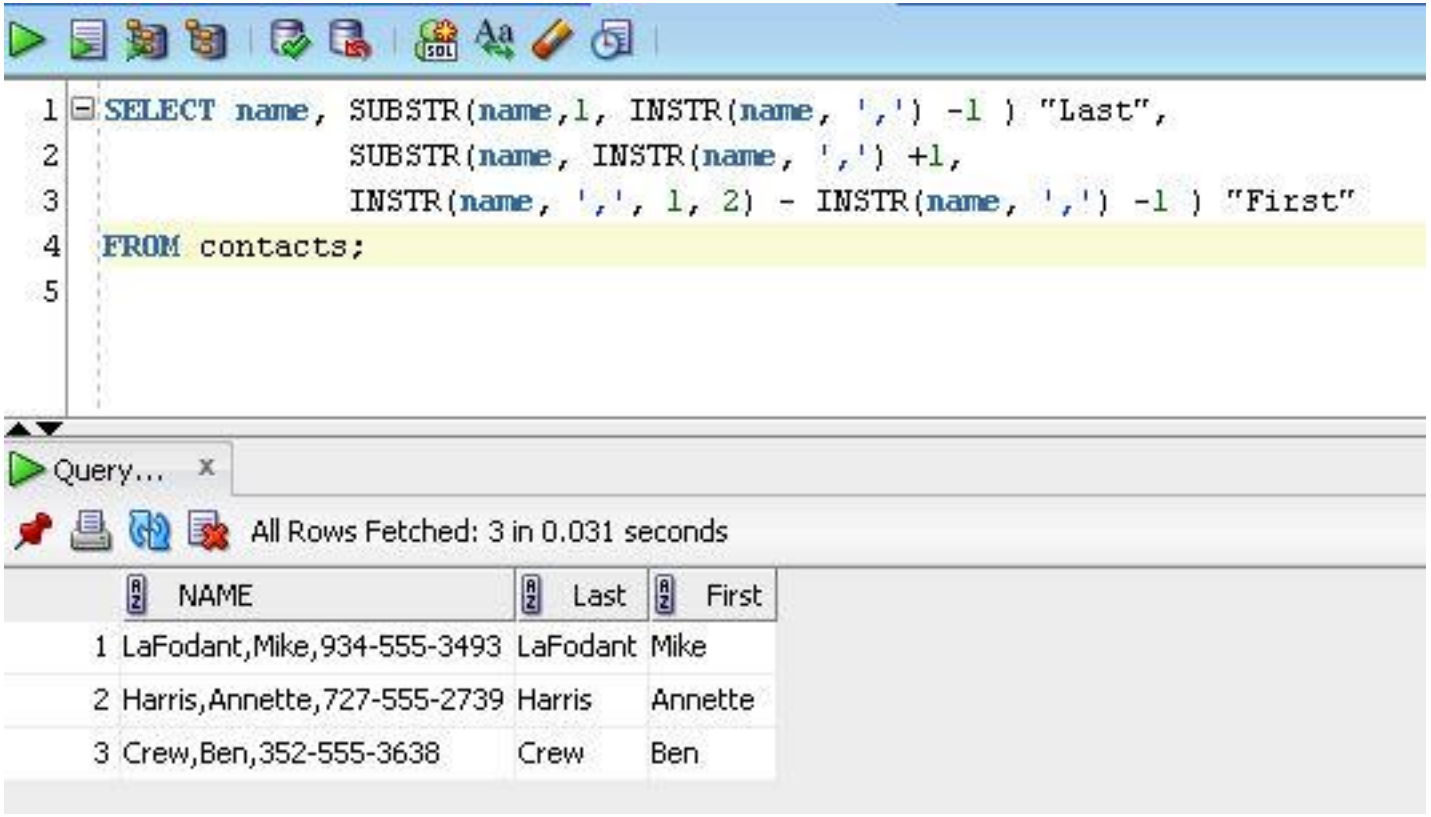
NESTED FUNCTIONS

- Any of the single-row functions can be nested inside other single-row functions
- When nesting functions you need to remember the following rules:
  - All arguments required for each function must be provided
  - For every opening parenthesis there must be a corresponding closing parenthesis
  - The nested or inner function is evaluated first, then the result of the inner function is passed to the outer function and then the outer function is executed
- See the nested function on slide 30, you may have to study it for awhile in order to figure out what is happening

INSTR NESTED INSIDE SUBSTR



INSTR NESTED INSIDE SUBSTR



```
1  SELECT name, SUBSTR(name,1, INSTR(name, ',') -1 ) "Last",
2               SUBSTR(name, INSTR(name, ',') +1,
3               INSTR(name, ',', 1, 2) - INSTR(name, ',') -1 ) "First"
4  FROM contacts;
5
```

Query...

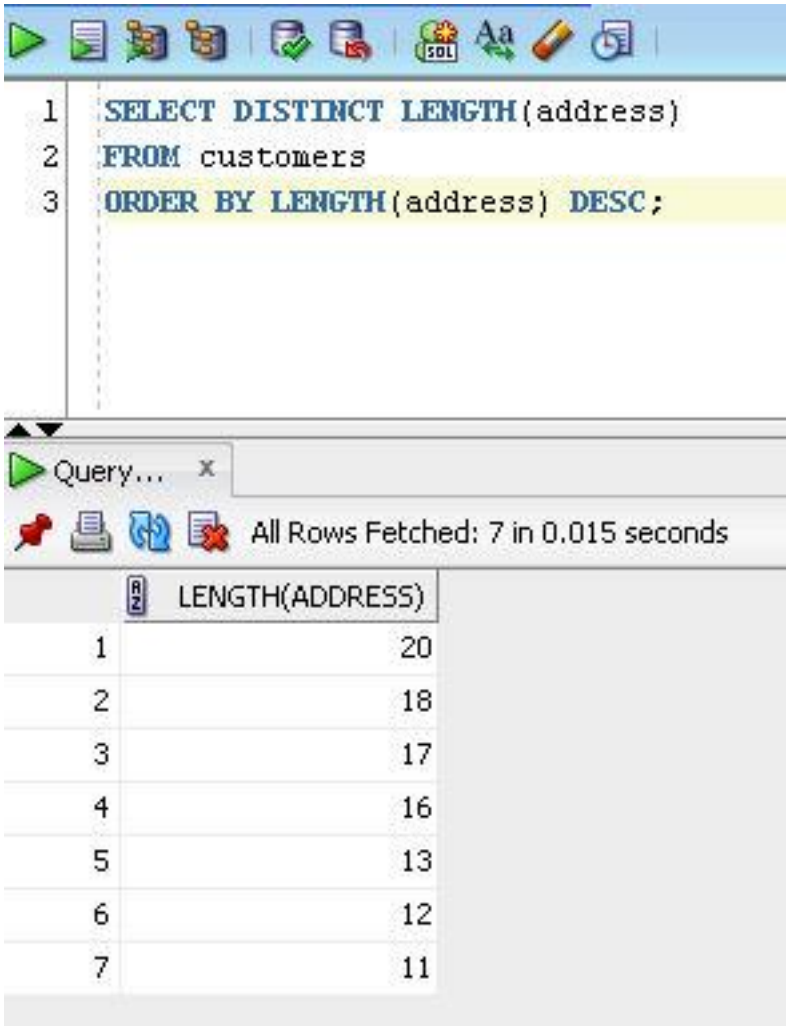
All Rows Fetched: 3 in 0.031 seconds

	NAME	Last	First
1	LaFodant,Mike,934-555-3493	LaFodant	Mike
2	Harris,Annette,727-555-2739	Harris	Annette
3	Crew,Ben,352-555-3638	Crew	Ben

LENGTH FUNCTION

- To determine the number of characters in a string, the LENGTH function is used
- The syntax of the LENGTH function is:
  - LENGTH(c)
    - Where (c) represents the character string whose length is to be determined

LENGTH FUNCTION



```
1  SELECT DISTINCT LENGTH(address)
2  FROM customers
3  ORDER BY LENGTH(address) DESC;
```

Query...

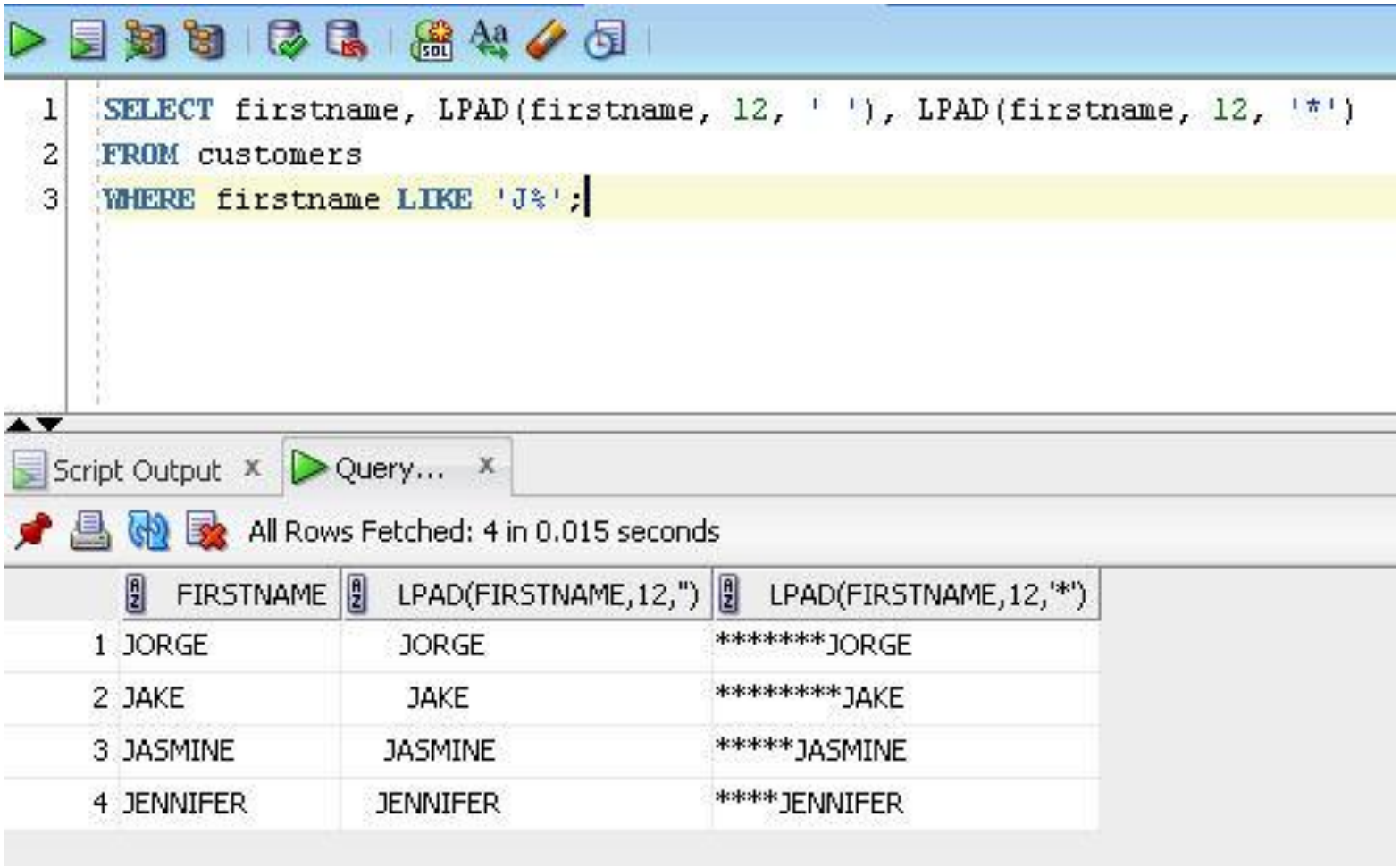
All Rows Fetched: 7 in 0.015 seconds

	LENGTH(ADDRESS)
1	20
2	18
3	17
4	16
5	13
6	12
7	11

LPAD AND RPAD FUNCTIONS

- The LPAD function can be used to pad or fill in the area to the left of a character string with a specific character or blank space
- For example, in some cheques that are sent out, blank spaces to the left of the digits on the cheque are padded with the asterisk character to make altering the cheque more difficult
- The syntax of the LPAD function is:
  - LPAD(c, l, s)
  - Where c represents the string to be padded, l represents the length of the character string after being padded and s represents the symbol or character to be used as padding

LPAD AND RPAD FUNCTIONS



```
1  SELECT firstname, LPAD(firstname, 12, ' '), LPAD(firstname, 12, '*')
2  FROM customers
3  WHERE firstname LIKE 'J%';
```

Script Output | Query...

All Rows Fetched: 4 in 0.015 seconds

	FIRSTNAME	LPAD(FIRSTNAME,12,'')	LPAD(FIRSTNAME,12,'*')
1	JORGE	JORGE	*******JORGE
2	JAKE	JAKE	********JAKE
3	JASMINE	JASMINE	*****JASMINE
4	JENNIFER	JENNIFER	****JENNIFER

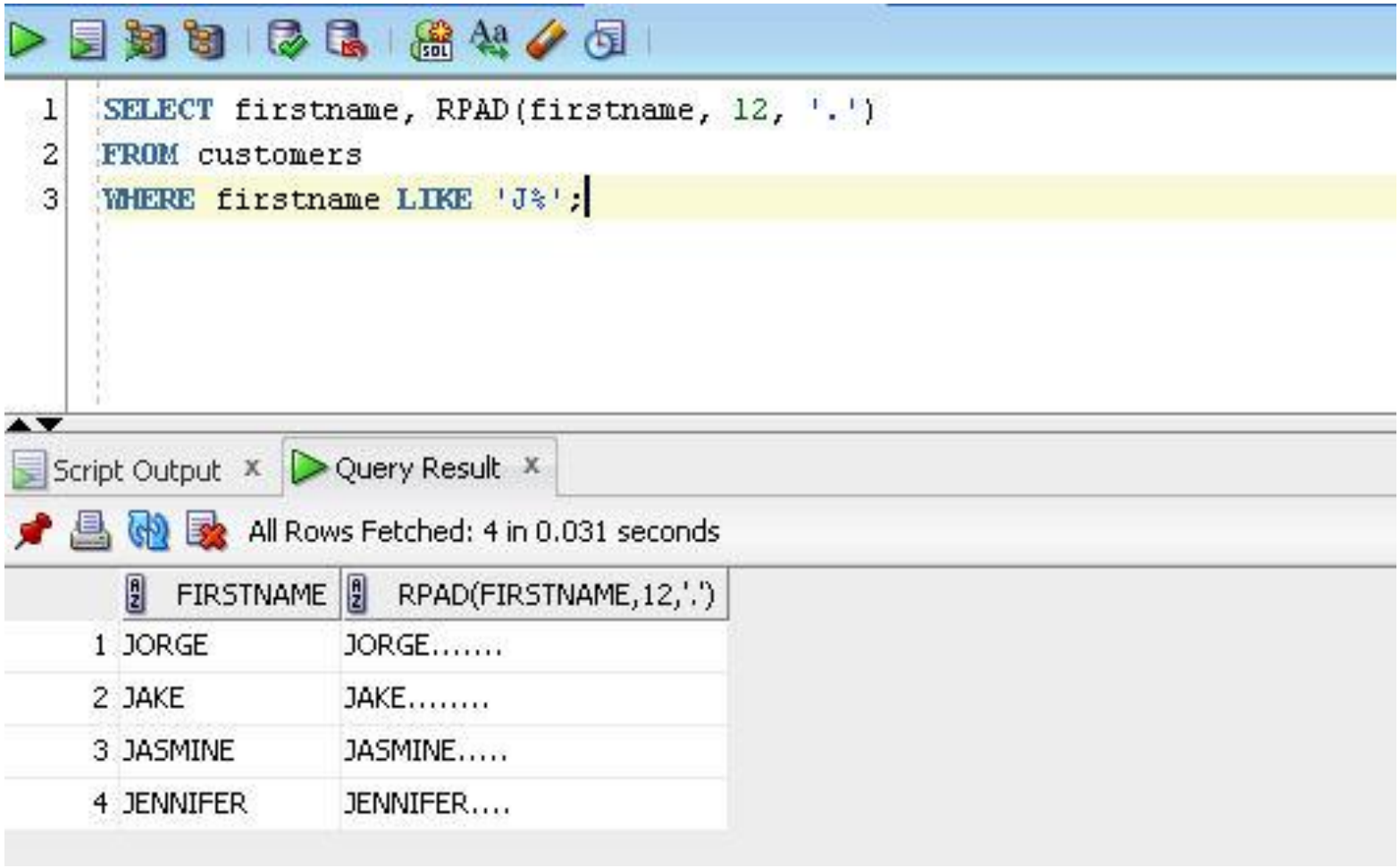
LPAD AND RPAD FUNCTIONS

- The arguments of the LPAD function (firstname, 12, ' ') indicate the following:
  - The firstname column is to be padded
  - The data contained in the firstname column is to be padded to a length of 12 characters. 12 is the total length, which includes both the data and the padding
  - A blank space is to be used as the padding symbol. The blank spaces are placed to the left of the customer's first name until the total length of 12 characters is reached
  - In our output, this actually right aligns the customer's first name in the second column of the output

LPAD AND RPAD FUNCTIONS

- The RPAD function uses a symbol to pad the right side of a character string to a specific width
- The syntax of the RPAD function is:
  - RPAD(c, l, s)
    - Where c represents the string to be padded, l represents the length of the character string after being padded and s represents the symbol or character to be used as padding

LPAD AND RPAD FUNCTIONS



```
1  SELECT firstname, RPAD(firstname, 12, '.')
2  FROM customers
3  WHERE firstname LIKE 'J%';
```

Script Output | Query Result

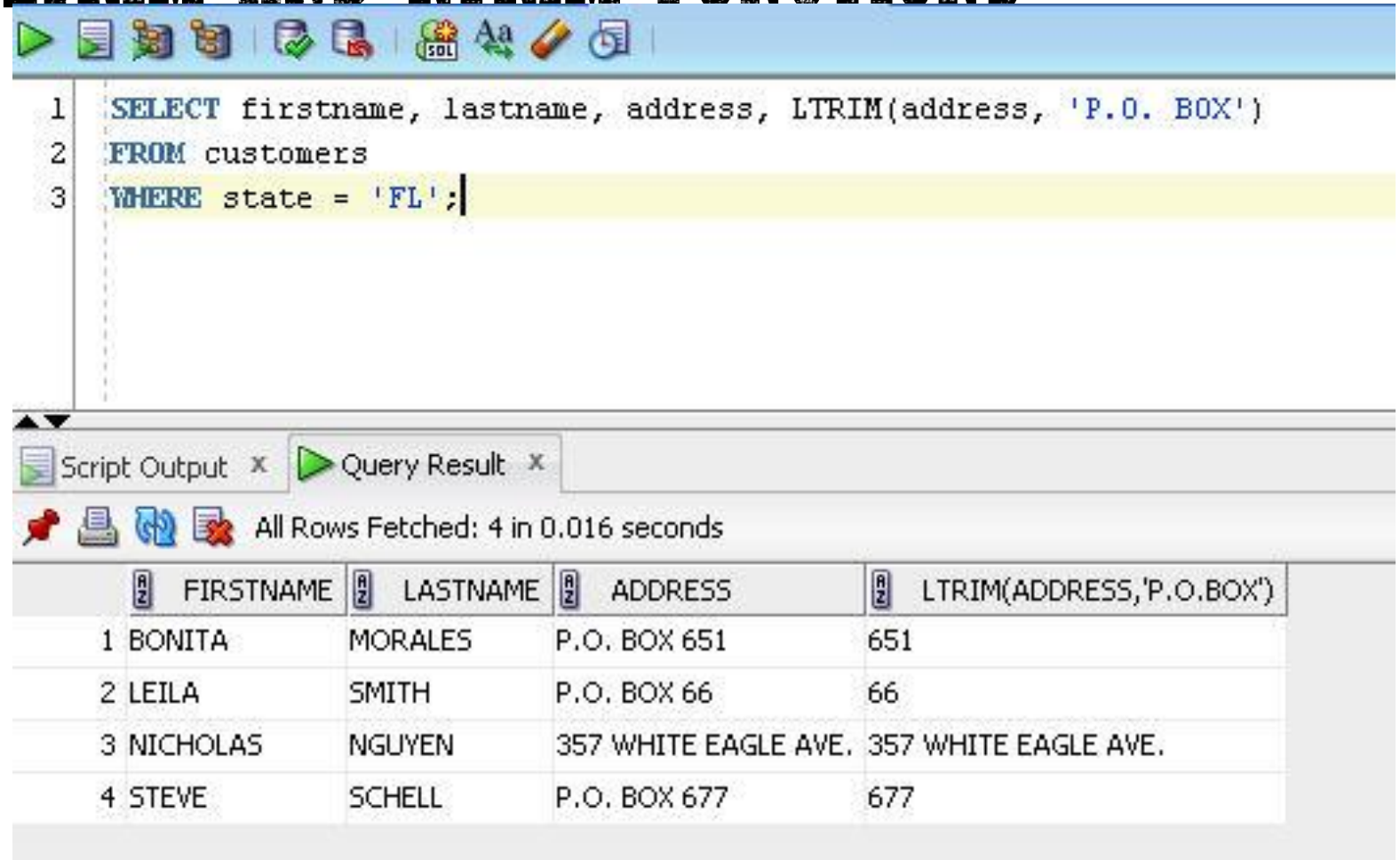
All Rows Fetched: 4 in 0.031 seconds

	FIRSTNAME	RPAD(FIRSTNAME,12,'.')
1	JORGE	JORGE.....
2	JAKE	JAKE.....
3	JASMINE	JASMINE.....
4	JENNIFER	JENNIFER....

LTRIM AND RTRIM FUNCTIONS

- The LTRIM function can be used to remove a specific string of characters from the left side of a set of data
- The syntax for the LTRIM function is:
  - LTRIM(c, s)
    - Where c represents the data to be affected and s represents the string to be removed from the left side of the data
- For example, a preprinted form is being used to send a mailing to customers. The preprinted form already contains the string P. O. Box. Any customers in the CUSTOMERS table that contain P. O. Box as part of their address need to have the string removed from the data so only the actual Box number itself will display

LTRIM AND RTRIM FUNCTIONS



```
1  SELECT firstname, lastname, address, LTRIM(address, 'P.O. BOX')
2  FROM customers
3  WHERE state = 'FL';
```

Script Output | Query Result

All Rows Fetched: 4 in 0.016 seconds

	FIRSTNAME	LASTNAME	ADDRESS	LTRIM(ADDRESS,'P.O.BOX')
1	BONITA	MORALES	P.O. BOX 651	651
2	LEILA	SMITH	P.O. BOX 66	66
3	NICHOLAS	NGUYEN	357 WHITE EAGLE AVE.	357 WHITE EAGLE AVE.
4	STEVE	SCHELL	P.O. BOX 677	677

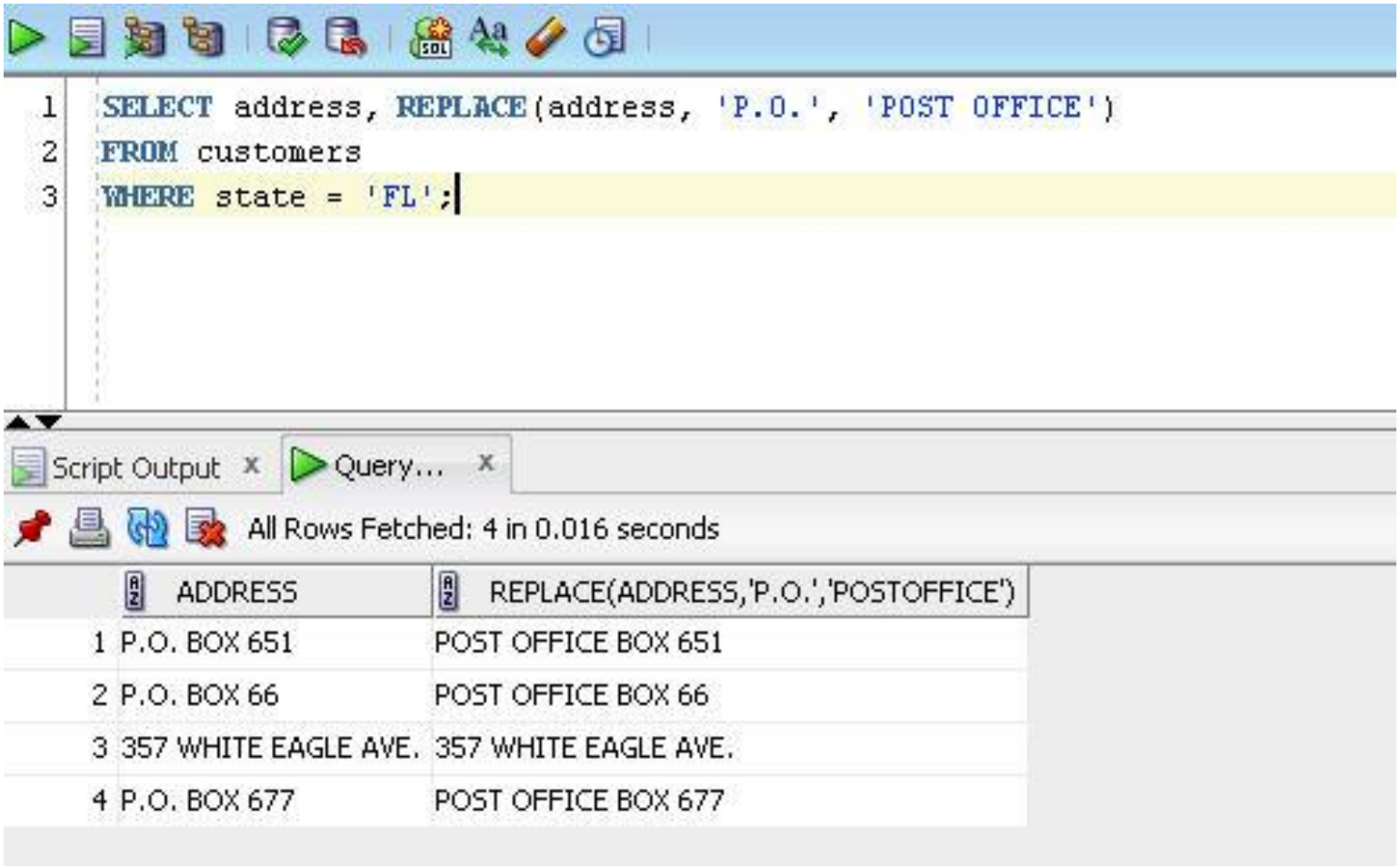
LTRIM AND RTRIM FUNCTIONS

- The RTRIM function removes specific characters from the right side of a set of data
- The syntax for the RTRIM function is:
  - RTRIM(c, s)
    - Where c represents the data to be affected and s is the string to be removed from the right side of the data

REPLACE FUNCTION

- The REPLACE function is similar to a "*search and replace*" function found is some application programs
- The REPLACE function looks for the occurrence of the specified string of characters, and, if it is found, replaces it with another set of specified characters
- The syntax for the REPLACE function is:
  - REPLACE(c, s, r)
    - Where c represents the data or column to be searched, s represents the string of characters to be replaced and r represents the string of characters to be substituted

REPLACE FUNCTION



```
1  SELECT address, REPLACE(address, 'P.O.', 'POST OFFICE')
2  FROM customers
3  WHERE state = 'FL';
```

Script Output | Query...

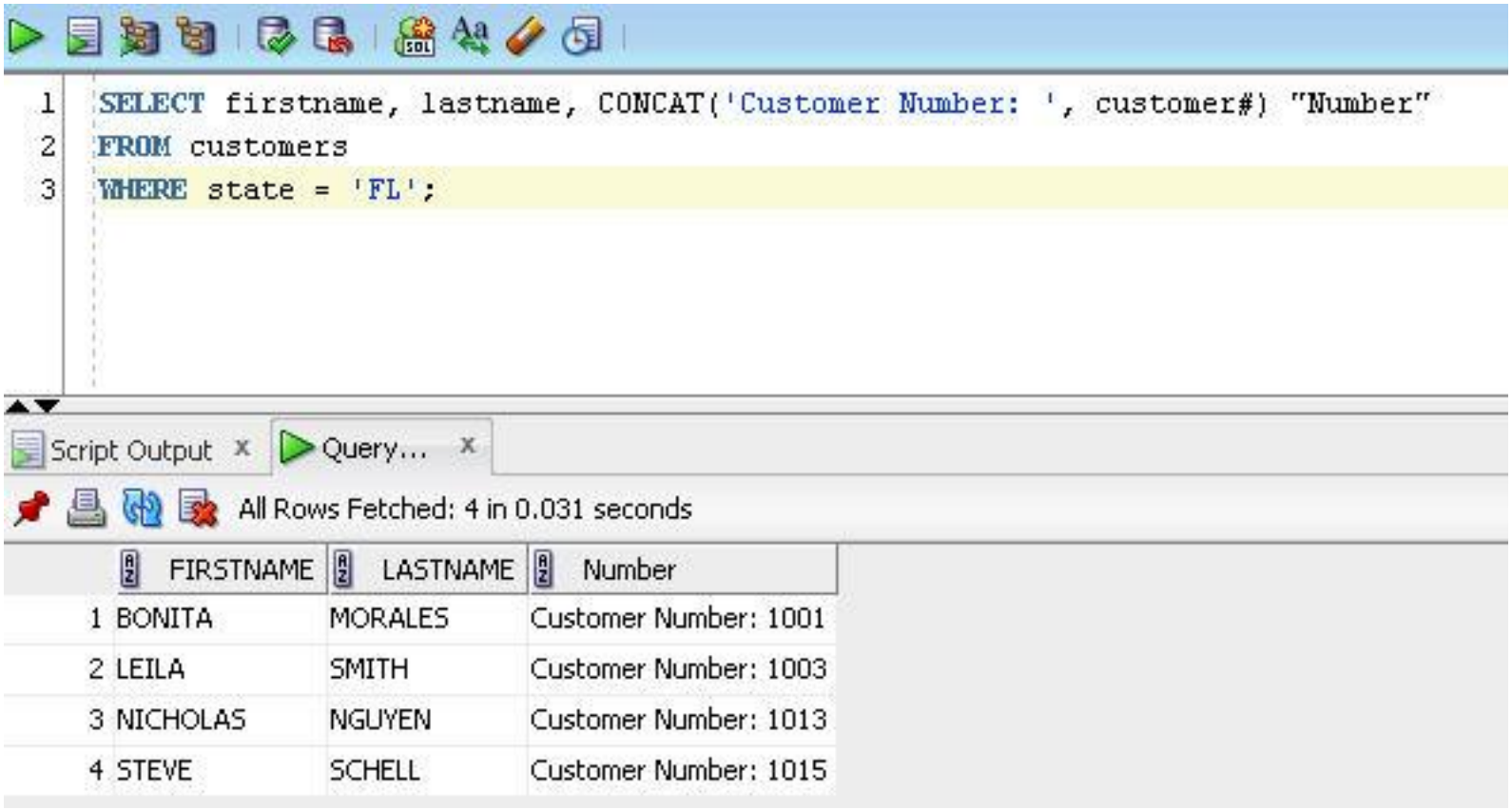
All Rows Fetched: 4 in 0.016 seconds

	ADDRESS	REPLACE(ADDRESS,'P.O.','POSTOFFICE')
1	P.O. BOX 651	POST OFFICE BOX 651
2	P.O. BOX 66	POST OFFICE BOX 66
3	357 WHITE EAGLE AVE.	357 WHITE EAGLE AVE.
4	P.O. BOX 677	POST OFFICE BOX 677

CONCAT FUNCTION

- We have already seen one method to concatenate strings, the (||) operator
- CONCAT function can be used to concatenate data from two columns
- The concatenation operator is usually preferred since it is not limited to two items
- If you need more than two items with the CONCAT function you would have nest a CONCAT function inside another CONCAT function
- Syntax is CONCAT(c1, c2)
- Both c1 and c2 can be a column name or a string literal

CONCAT FUNCTION



```
1  SELECT firstname, lastname, CONCAT('Customer Number: ', customer#) "Number"
2  FROM customers
3  WHERE state = 'FL';
```

Script Output | Query...

All Rows Fetched: 4 in 0.031 seconds

	FIRSTNAME	LASTNAME	Number
1	BONITA	MORALES	Customer Number: 1001
2	LEILA	SMITH	Customer Number: 1003
3	NICHOLAS	NGUYEN	Customer Number: 1013
4	STEVE	SCHELL	Customer Number: 1015

NUMBER FUNCTIONS

- Oracle 12c provides a set of functions that are specifically designed for the manipulation of numeric data
- In daily operations, there are two popular number functions, ROUND, TRUNC, MOD, ABS, and POWER

ROUND FUNCTION

- The ROUND function is designed to round numeric values to the given precision
- The syntax of the ROUND function is:
  - ROUND(n, p)
    - Where n represents the numeric data or column to be rounded and p represents the position of the digit to which the data should be rounded
    - If p is a positive number then round on the right side of the decimal position
    - If p is a negative number then round on the left side of the decimal position

ROUND FUNCTION

```
1  SELECT title, retail, ROUND(retail, 1)
2  FROM books;
```

Script Output | Query Result

All Rows Fetched: 14 in 0.094 seconds

	TITLE	RETAIL	ROUND(RETAIL,1)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	31
2	REVENGE OF MICKEY	22	22
3	BUILDING A CAR WITH TOOTHPICKS	59.95	60
4	DATABASE IMPLEMENTATION	55.95	56
5	COOKING WITH MUSHROOMS	19.95	20
6	HOLY GRAIL OF ORACLE	75.95	76
7	HANDCRANKED COMPUTERS	25	25
8	E-BUSINESS THE EASY WAY	54.5	54.5
9	PAINLESS CHILD-REARING	89.95	90
10	THE WOK WAY TO COOK	28.75	28.8
11	BIG BEAR AND LITTLE DOVE	8.95	9
12	HOW TO GET FASTER PIZZA	29.95	30
13	HOW TO MANAGE THE MANAGER	31.95	32
14	SHORTEST POEMS	39.95	40

Most are rounded to the nearest dollar. They round up to an even dollar because of the 95 in the original data.

Look at The Wok Way to Cook. It is rounded up to 8 from .75

ROUND FUNCTION

```
1  SELECT title, retail, ROUND(retail, 0)
2  FROM books;
```

Script Output | Query Result

All Rows Fetched: 14 in 0 seconds

	TITLE	RETAIL	ROUND(RETAIL,0)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	31
2	REVENGE OF MICKEY	22	22
3	BUILDING A CAR WITH TOOTHPICKS	59.95	60
4	DATABASE IMPLEMENTATION	55.95	56
5	COOKING WITH MUSHROOMS	19.95	20
6	HOLY GRAIL OF ORACLE	75.95	76
7	HANDCRANKED COMPUTERS	25	25
8	E-BUSINESS THE EASY WAY	54.5	55
9	PAINLESS CHILD-REARING	89.95	90
10	THE WOK WAY TO COOK	28.75	29
11	BIG BEAR AND LITTLE DOVE	8.95	9
12	HOW TO GET FASTER PIZZA	29.95	30
13	HOW TO MANAGE THE MANAGER	31.95	32
14	SHORTEST POEMS	39.95	40

A value of 0 for p indicates that the number should be rounded to the nearest whole number with no decimal places

ROUND FUNCTION

```
1  SELECT title, retail, ROUND(retail, -1)
2  FROM books;
```

Script Output | Query Result

All Rows Fetched: 14 in 0.015 seconds

	TITLE	RETAIL	ROUND(RETAIL,-1)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	30
2	REVENGE OF MICKEY	22	20
3	BUILDING A CAR WITH TOOTHPICKS	59.95	60
4	DATABASE IMPLEMENTATION	55.95	60
5	COOKING WITH MUSHROOMS	19.95	20
6	HOLY GRAIL OF ORACLE	75.95	80
7	HANDCRANKED COMPUTERS	25	30
8	E-BUSINESS THE EASY WAY	54.5	50
9	PAINLESS CHILD-REARING	89.95	90
10	THE WOK WAY TO COOK	28.75	30
11	BIG BEAR AND LITTLE DOVE	8.95	10
12	HOW TO GET FASTER PIZZA	29.95	30
13	HOW TO MANAGE THE MANAGER	31.95	30
14	SHORTEST POEMS	39.95	40

The –1 indicates rounding on the left side of the decimal. This would round to the nearest tens of dollars

ROUND FUNCTION

```
1  SELECT title, retail, ROUND(retail, 1), ROUND(retail, 0), ROUND(retail, -1)
2  FROM books;
```

Script Output | Query...

All Rows Fetched: 14 in 0.016 seconds

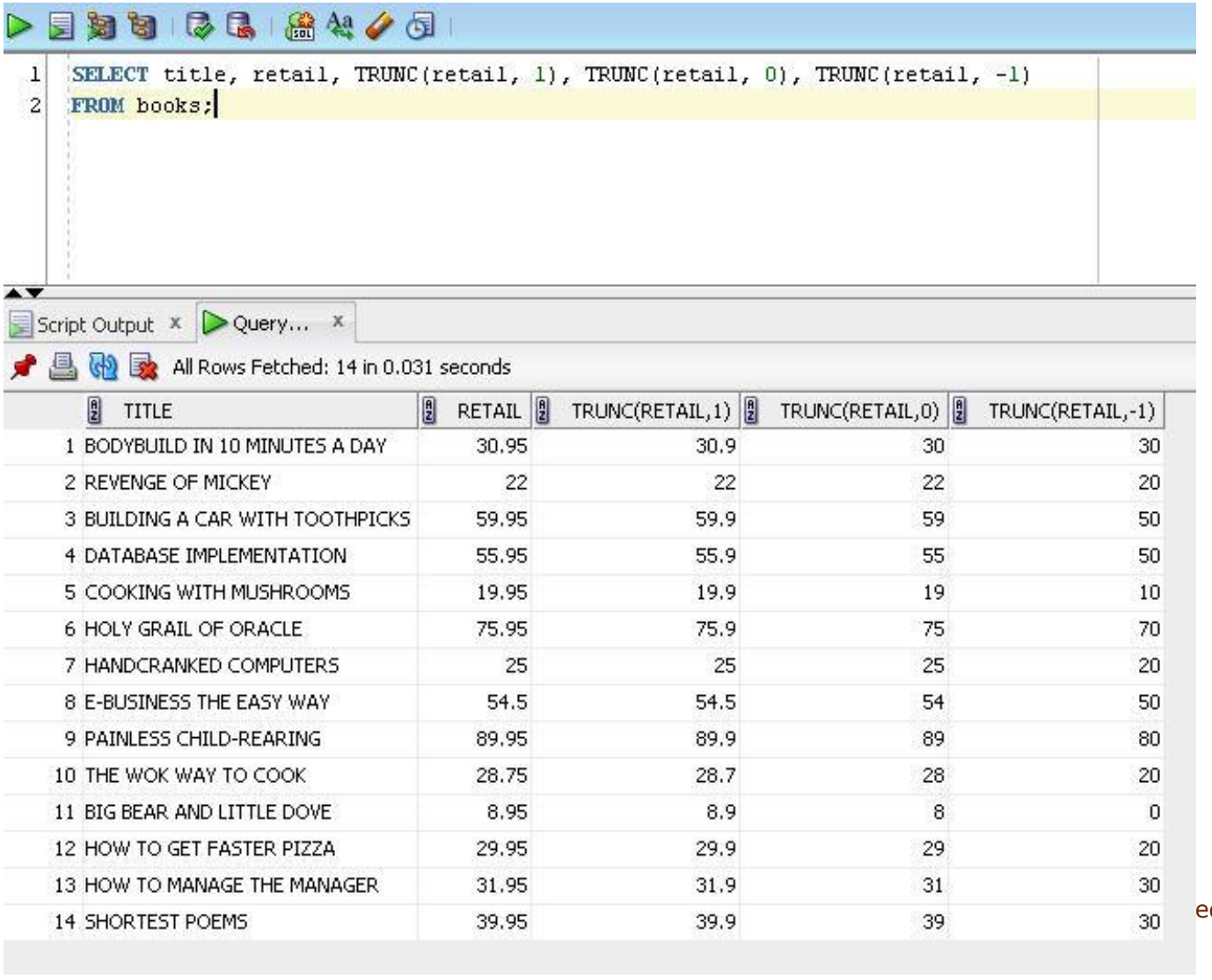
	TITLE	RETAIL	ROUND(RETAIL,1)	ROUND(RETAIL,0)	ROUND(RETAIL,-1)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	31	31	30
2	REVENGE OF MICKEY	22	22	22	20
3	BUILDING A CAR WITH TOOTHPICKS	59.95	60	60	60
4	DATABASE IMPLEMENTATION	55.95	56	56	60
5	COOKING WITH MUSHROOMS	19.95	20	20	20
6	HOLY GRAIL OF ORACLE	75.95	76	76	80
7	HANDCRANKED COMPUTERS	25	25	25	30
8	E-BUSINESS THE EASY WAY	54.5	54.5	55	50
9	PAINLESS CHILD-REARING	89.95	90	90	90
10	THE WOK WAY TO COOK	28.75	28.8	29	30
11	BIG BEAR AND LITTLE DOVE	8.95	9	9	10
12	HOW TO GET FASTER PIZZA	29.95	30	30	30
13	HOW TO MANAGE THE MANAGER	31.95	32	32	30
14	SHORTEST POEMS	39.95	40	40	40

Comparison of the three methods beside each other

TRUNC FUNCTION

- The TRUNC function is used to truncate numeric data rather than round it
- The TRUNC function can be used to truncate numeric data to a specific position. Any numbers after that position are simply removed or dropped off
- The syntax for the TRUNC function is:
  - TRUNC(n, p)
    - Where n represents the numeric data or field to be truncated and p represents the position from the digit from which the data should be removed
    - Whether p is *positive* or *negative,* it follows the same rules as it did for the ROUND function

TRUNC FUNCTION



```
1  SELECT title, retail, TRUNC(retail, 1), TRUNC(retail, 0), TRUNC(retail, -1)
2  FROM books;
```

Script Output | Query...

All Rows Fetched: 14 in 0.031 seconds

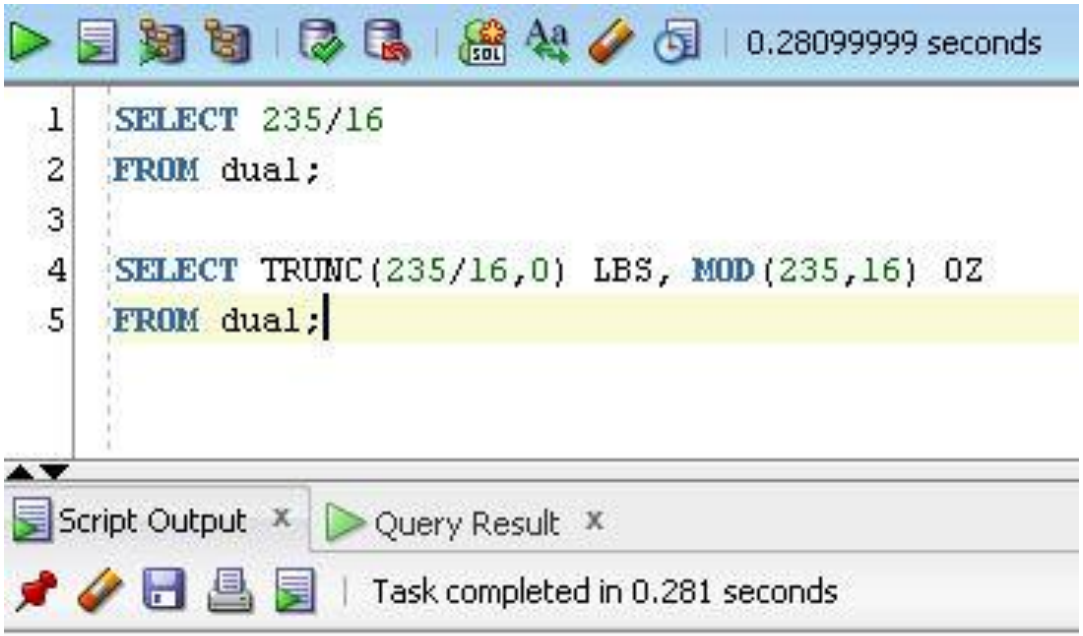
	TITLE	RETAIL	TRUNC(RETAIL,1)	TRUNC(RETAIL,0)	TRUNC(RETAIL,-1)
1	BODYBUILD IN 10 MINUTES A DAY	30.95	30.9	30	30
2	REVENGE OF MICKEY	22	22	22	20
3	BUILDING A CAR WITH TOOTHPICKS	59.95	59.9	59	50
4	DATABASE IMPLEMENTATION	55.95	55.9	55	50
5	COOKING WITH MUSHROOMS	19.95	19.9	19	10
6	HOLY GRAIL OF ORACLE	75.95	75.9	75	70
7	HANDCRANKED COMPUTERS	25	25	25	20
8	E-BUSINESS THE EASY WAY	54.5	54.5	54	50
9	PAINLESS CHILD-REARING	89.95	89.9	89	80
10	THE WOK WAY TO COOK	28.75	28.7	28	20
11	BIG BEAR AND LITTLE DOVE	8.95	8.9	8	0
12	HOW TO GET FASTER PIZZA	29.95	29.9	29	20
13	HOW TO MANAGE THE MANAGER	31.95	31.9	31	30
14	SHORTEST POEMS	39.95	39.9	39	30

TRUNC function used with a p value of 1, 0 and –1. With –1, the digit to the left of the decimal reverts to a zero, it is still needed as a placeholder

MOD FUNCTION

- The MOD function returns the remainder only of a division operation
- The two arguments are needed for the function are the numerator and denominator of the division operation
- We want to convert the total ounces into pounds and ounces
- The decimal amount might not be easily used by someone needing to know the weight in pounds and ounces

MOD FUNCTION




```
235/16
----------------------
14.6875

LBS                    OZ
---------------------- ----------------------
14                     11
```

We will discuss the use of the DUAL table later in this chapter

MOD FUNCTION

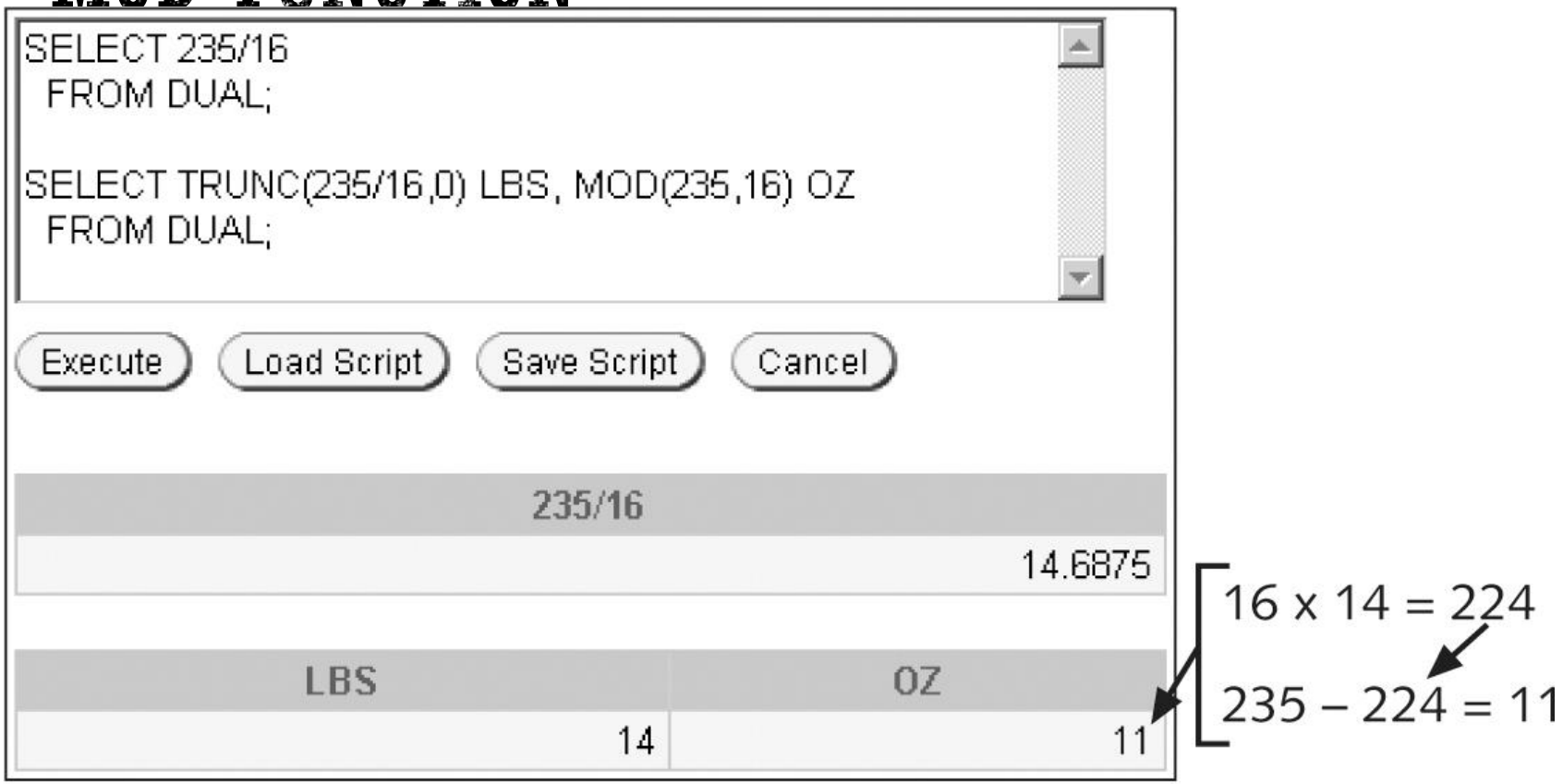
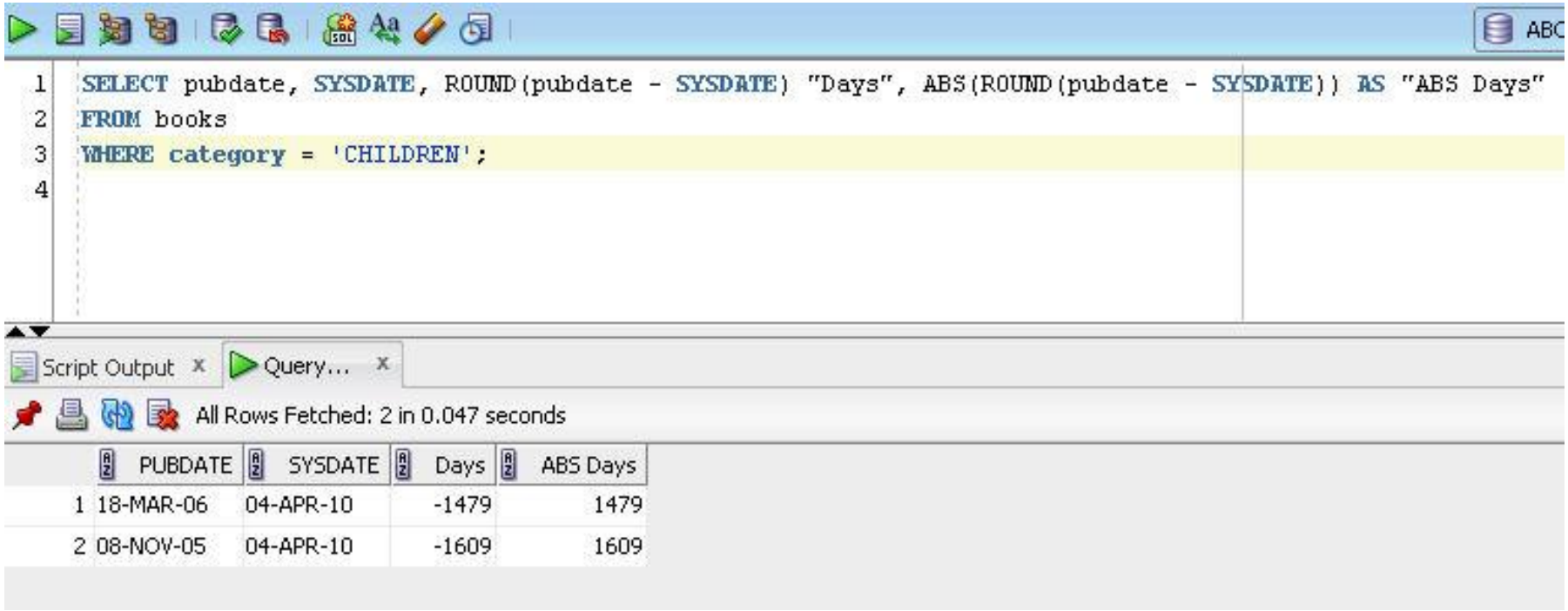


FIGURE 10-19 Using the MOD function to return the remainder

ABS FUNCTION

- The ABS (Absolute) function returns the absolute or positive value of the numeric value provided as the argument
- If you are only concerned with the difference of two date values and not the order of the date values used in the calculation use the ABS function to always return the absolute value of the result

ABS FUNCTION



```
1  SELECT pubdate, SYSDATE, ROUND(pubdate - SYSDATE) "Days", ABS(ROUND(pubdate - SYSDATE)) AS "ABS Days"
2  FROM books
3  WHERE category = 'CHILDREN';
4
```

Script Output | Query...

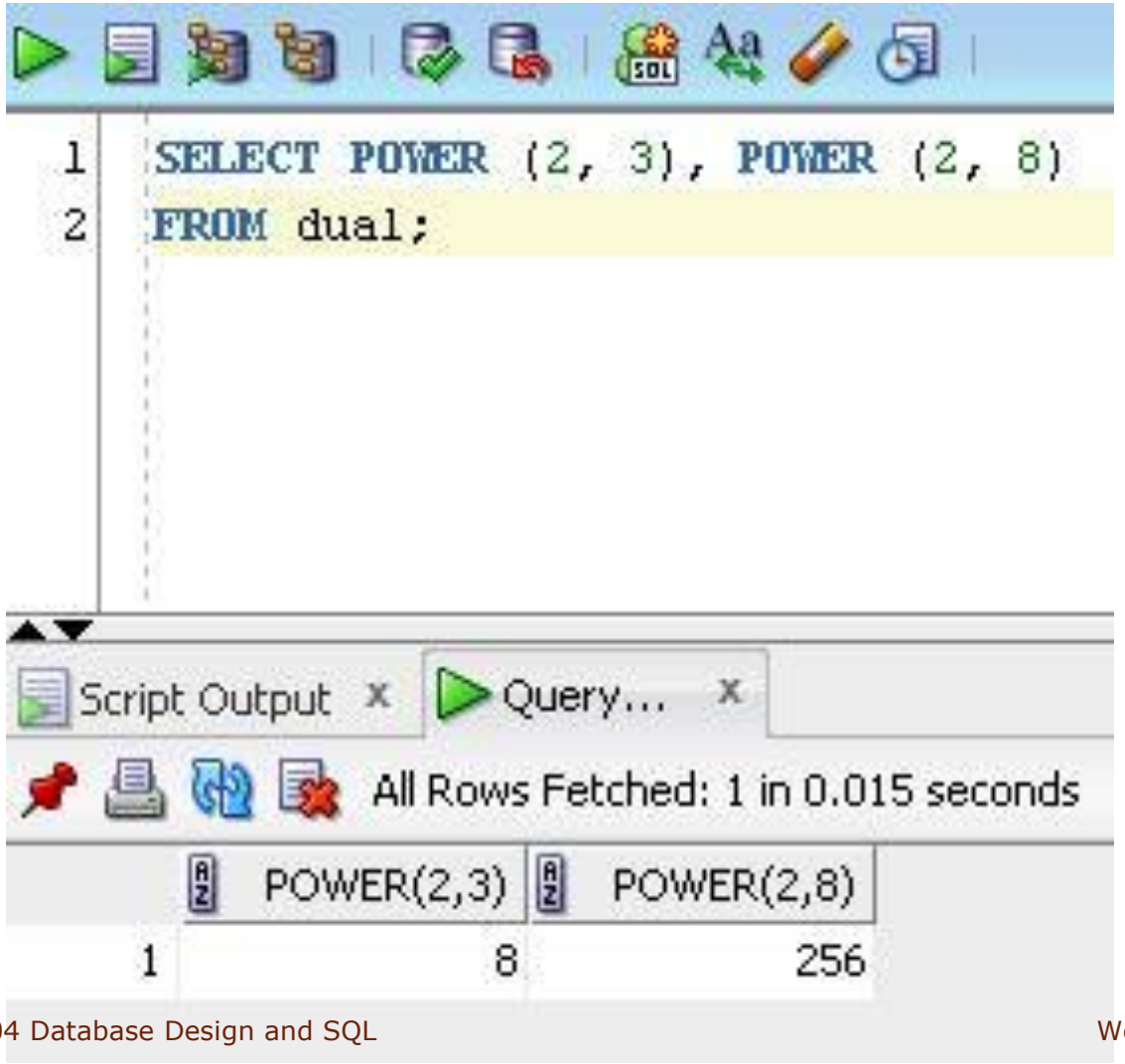
All Rows Fetched: 2 in 0.047 seconds

	PUBDATE	SYSDATE	Days	ABS Days
1	18-MAR-06	04-APR-10	-1479	1479
2	08-NOV-05	04-APR-10	-1609	1609

POWER FUNCTION

- The POWER function raises the number in the first argument to the power indicated in the second argument
- The syntax of the POWER function is:
  - POWER(x, y)
- Where x is the number you are raising and y represents the power you are raising it to

POWER FUNCTION



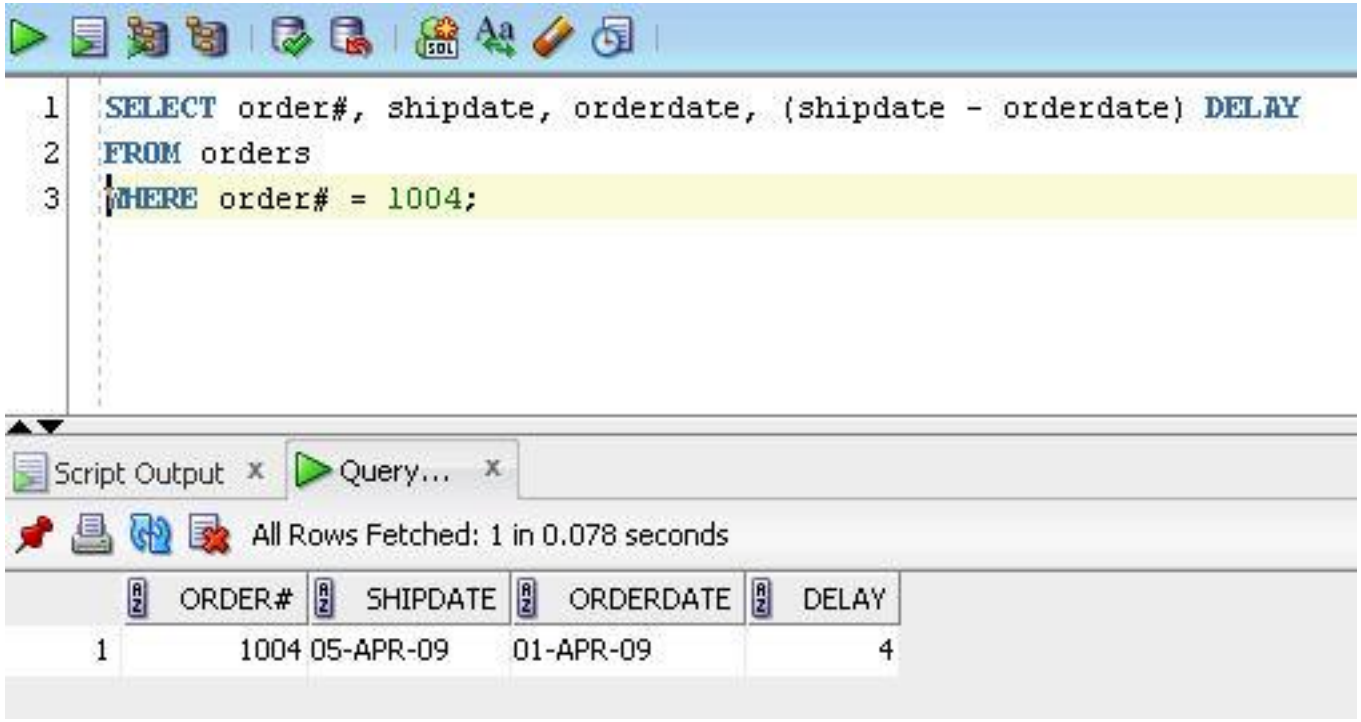
DATE FUNCTIONS

- The DATE function of Oracle 12c displays date values in a *DD-MON-RR* format that represents a 2-digit day, 3-letter month abbreviation and a 2-digit year
- March 5, 2003 would be stored as 05-MAR-03
- A date is referenced as a non-numeric field and is enclosed in single quotation marks
- Internally it is stored as a numeric value representing the *century*, *year*, *month*, *day*, *hours*, *minutes*, and *seconds*
- The valid range for a date is *4712 B.C.* to *9999 A.D.*
- Although a date appears as a non-numeric field, users can perform calculations on dates because they are stored internally as numbers

DATE FUNCTIONS

- The internal numeric version of the date used by Oracle 12c is a Julian date
- A Julian date represents the number of days that have passed from a specific date and 4712 B.C.
- If you need to calculate the number of days between two dates
- Oracle first converts the dates to the Julian date format, then determines the difference between the two dates

DATE FUNCTIONS



Results display the number of days that occur between the 2 dates specified

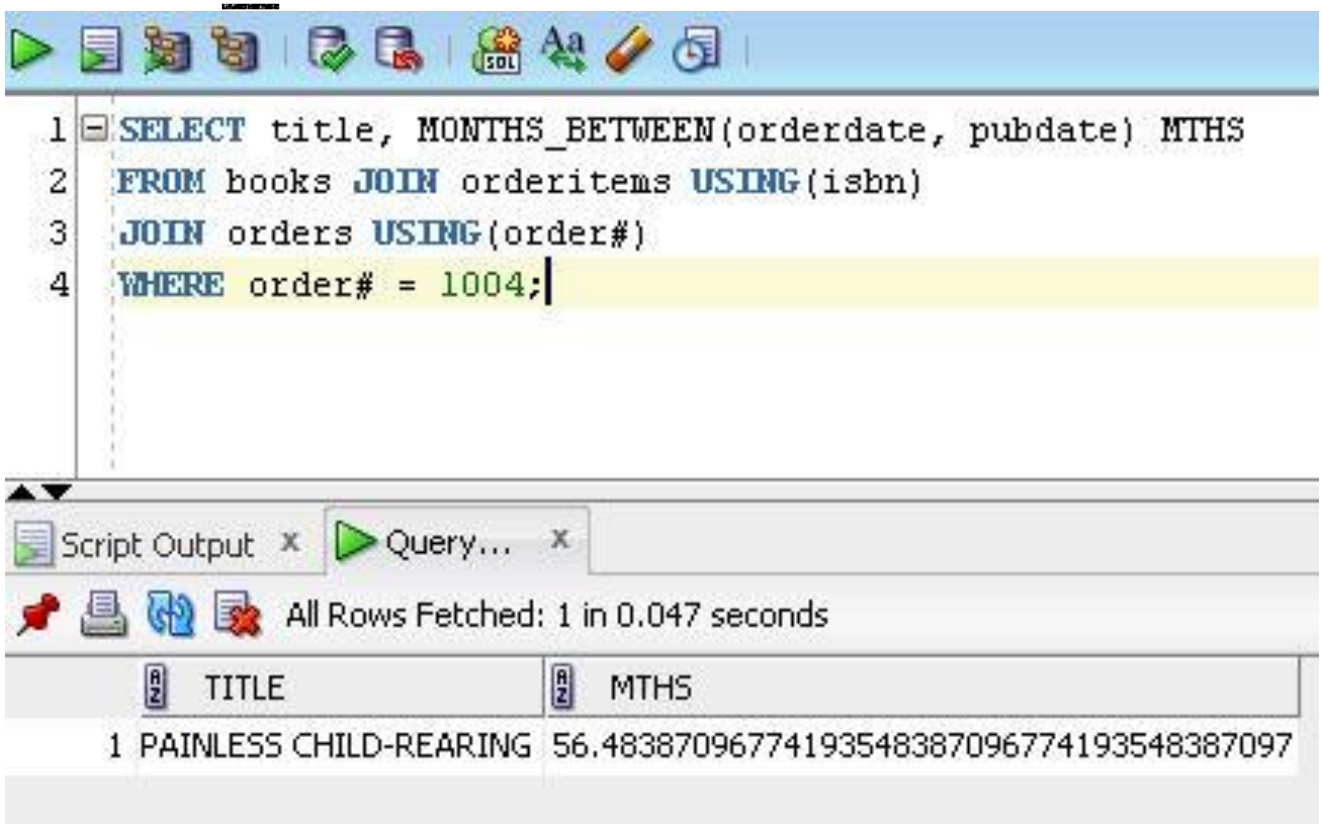
DATE FUNCTIONS

- If you wanted to know the number of months between two dates, it becomes a little tricky since not all months have the same number of days
- Oracle 12c provides several functions to assist in these kinds of calculations:
  - MONTHS_BETWEEN
  - ADD_MONTHS
  - NEXT_DAY
  - LAST_DAY
  - TO_DATE

MONTHS_BETWEEN FUNCTION

- Management would like to know whether customers are ordering books that have recently been released, or books that were published several months or even several years ago
- You could do as we saw previously; subtract publication date from order date to show the number of days or even weeks
- Instead, we can use the MONTHS_BETWEEN function to determine the number of months between two dates
- Syntax is:
  - MONTHS_BETWEEN(d1, d2)
    - Where d1 and d2 are dates and d2 is subtracted from d1

MONTHS_BETWEEN FUNCTION



```
1 SELECT title, MONTHS_BETWEEN(orderdate, pubdate) MTHS
2 FROM books JOIN orderitems USING(isbn)
3 JOIN orders USING(order#)
4 WHERE order# = 1004;
```

Script Output | Query...

All Rows Fetched: 1 in 0.047 seconds

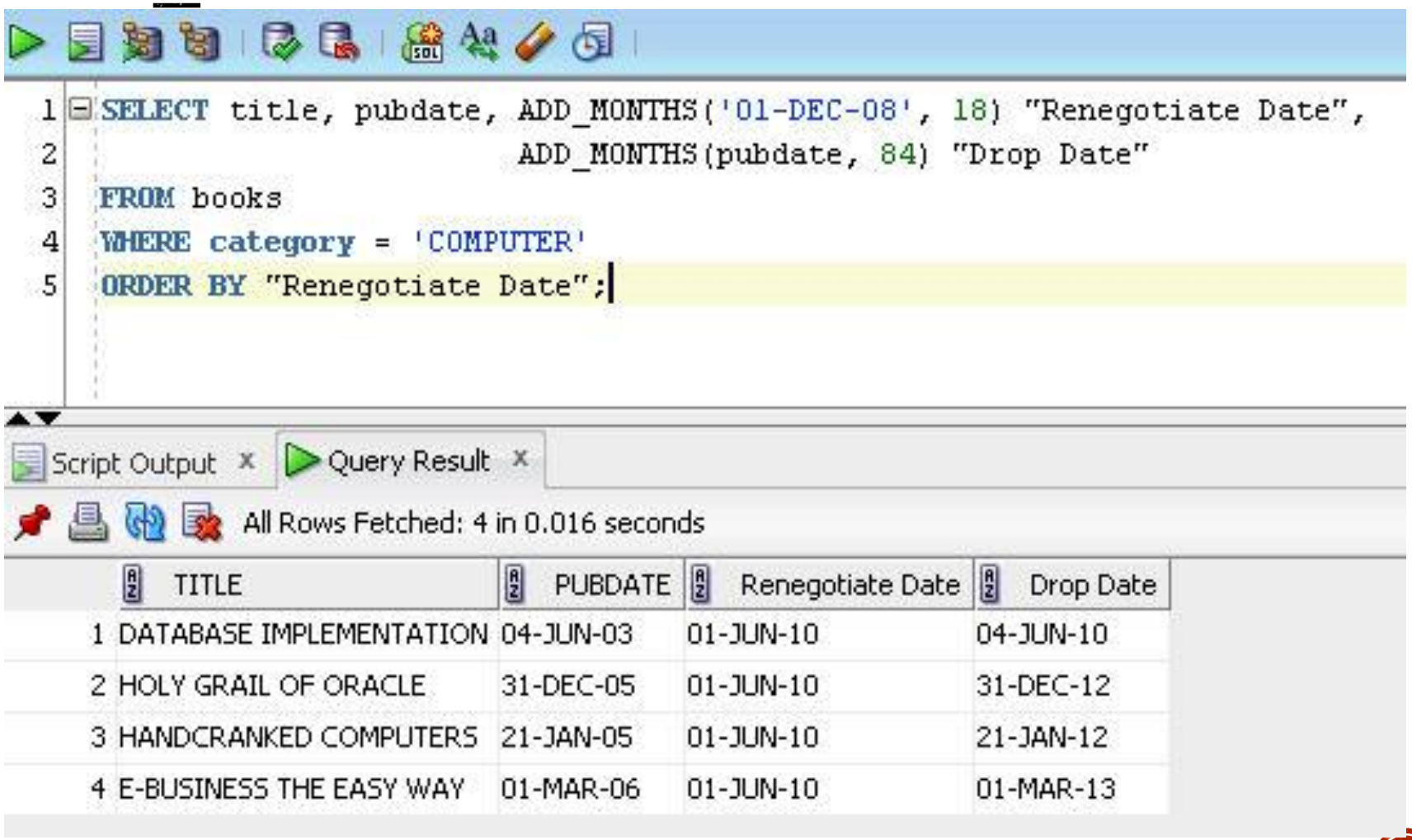
	TITLE	MTHS
1	PAINLESS CHILD-REARING	56.483870967741935483870967741935483870967

If you wanted to remove the decimal portion of the output, you could truncate the result so only the integer portion is seen. More on this later

ADD_MONTHS FUNCTION

- Just Lee Books would like to drop any inventory that has been in stock for longer than 5 years
- We need to produce a list showing the publication date and the drop date for all books in inventory
- We could simply add (365*5) to the publication date or could use the ADD_MONTHS function which is a better solution
- The syntax for the ADD_MONTHS function is:
  - ADD_MONTHS(d, m)
    - Where d represents the beginning date and m represents the number of months to add to d

ADD_MONTHS FUNCTION



```
1  SELECT title, pubdate, ADD_MONTHS('01-DEC-08', 18) "Renegotiate Date",
2                         ADD_MONTHS(pubdate, 84) "Drop Date"
3  FROM books
4  WHERE category = 'COMPUTER'
5  ORDER BY "Renegotiate Date";
```

Script Output | Query Result

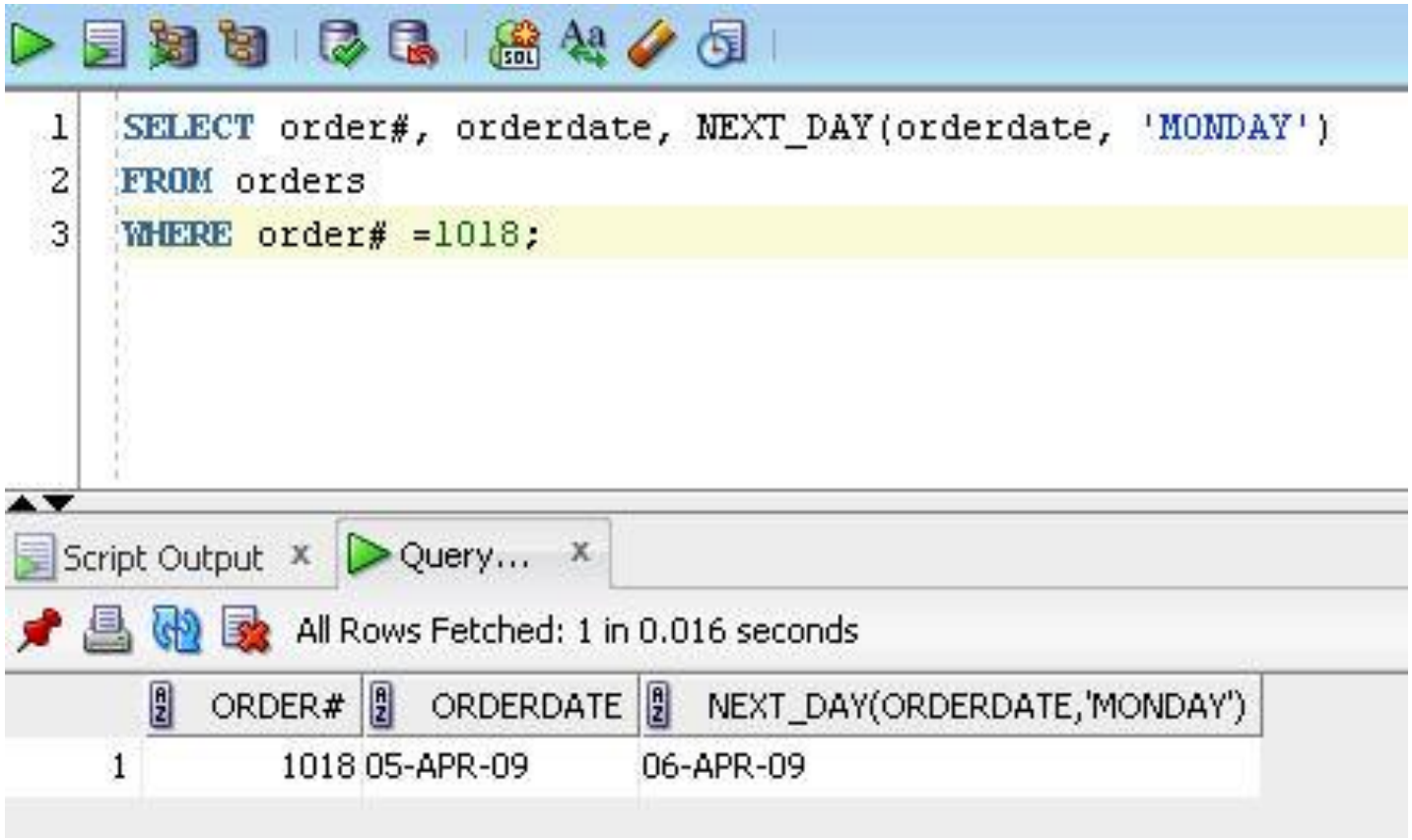
All Rows Fetched: 4 in 0.016 seconds

	TITLE	PUBDATE	Renegotiate Date	Drop Date
1	DATABASE IMPLEMENTATION	04-JUN-03	01-JUN-10	04-JUN-10
2	HOLY GRAIL OF ORACLE	31-DEC-05	01-JUN-10	31-DEC-12
3	HANDCRANKED COMPUTERS	21-JAN-05	01-JUN-10	21-JAN-12
4	E-BUSINESS THE EASY WAY	01-MAR-06	01-JUN-10	01-MAR-13

NEXT_DAY FUNCTION

- Just Lee Books has a policy that books must be shipped by the first Monday after they receive a customer's order
- The NEXT_DAY function can be used to determine the next occurrence of a specific day of the week and the give you the actual date of that day
- The syntax for the NEXT_DAY function is:
  - NEXT_DAY(d, DAY)
    - Where d represents the starting date and DAY represents the day of the week to be identified

NEXT_DAY FUNCTION

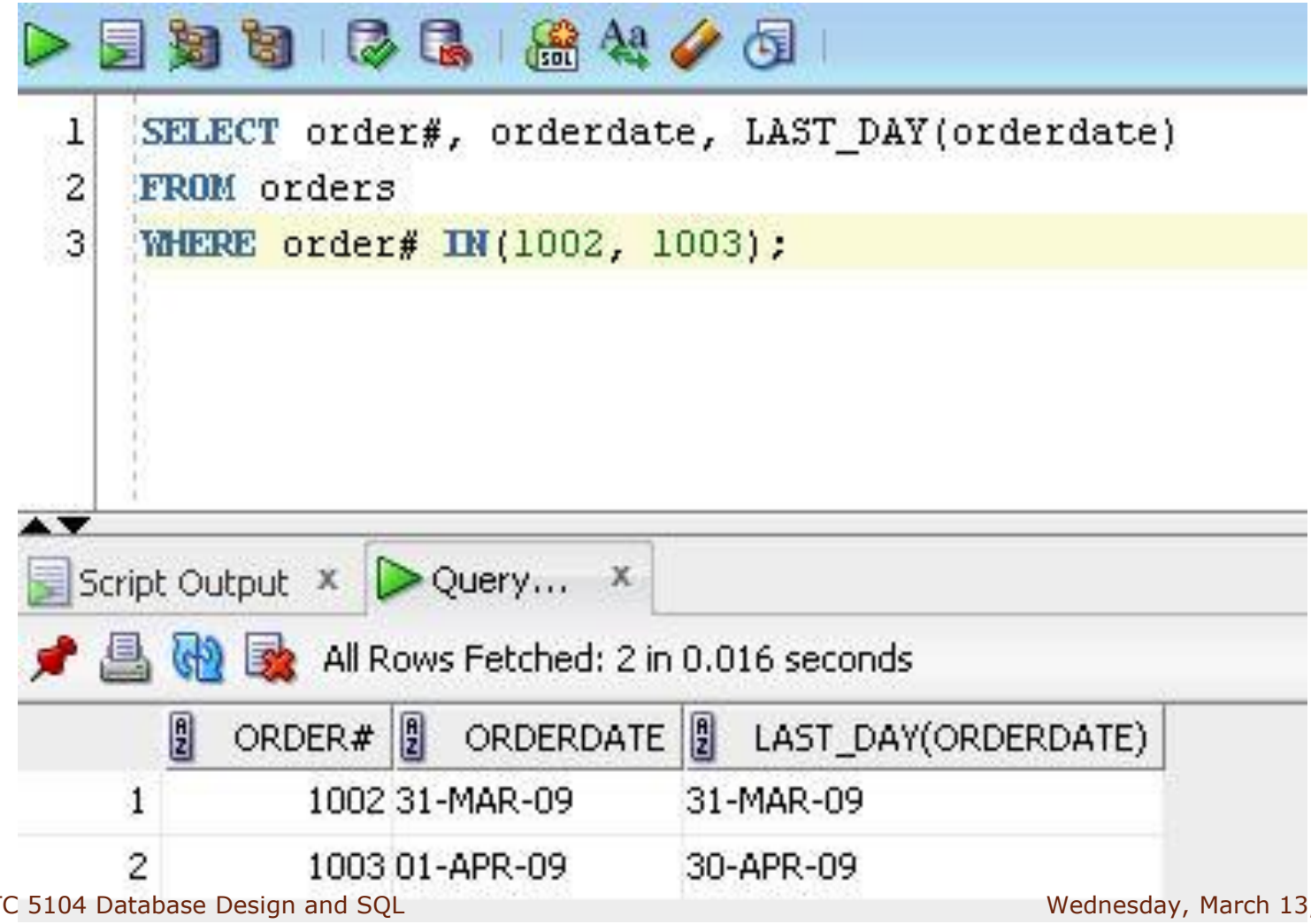


The next Monday from the order date is April 6th

LAST_DAY FUNCTION

- Similar to NEXT_DAY function except it always determines the last day of the month for the month of two selected order dates

LAST_DAY FUNCTION



```
1  SELECT order#, orderdate, LAST_DAY(orderdate)
2  FROM orders
3  WHERE order# IN(1002, 1003);
```

Script Output | Query...

All Rows Fetched: 2 in 0.016 seconds

	ORDER#	ORDERDATE	LAST_DAY(ORDERDATE)
1	1002	31-MAR-09	31-MAR-09
2	1003	01-APR-09	30-APR-09

TO_DATE FUNCTION

- The TO_DATE function is of particular interest to application developers since it permits the user to enter the date in any format
- This date is then converted to the standard Oracle 12c default format for storage
- The syntax for the TO_DATE function:
  - TO_DATE(d,'f')
    - Where d represents the date being entered by the user and f represents the format for the date entered

TO_DATE FUNCTION

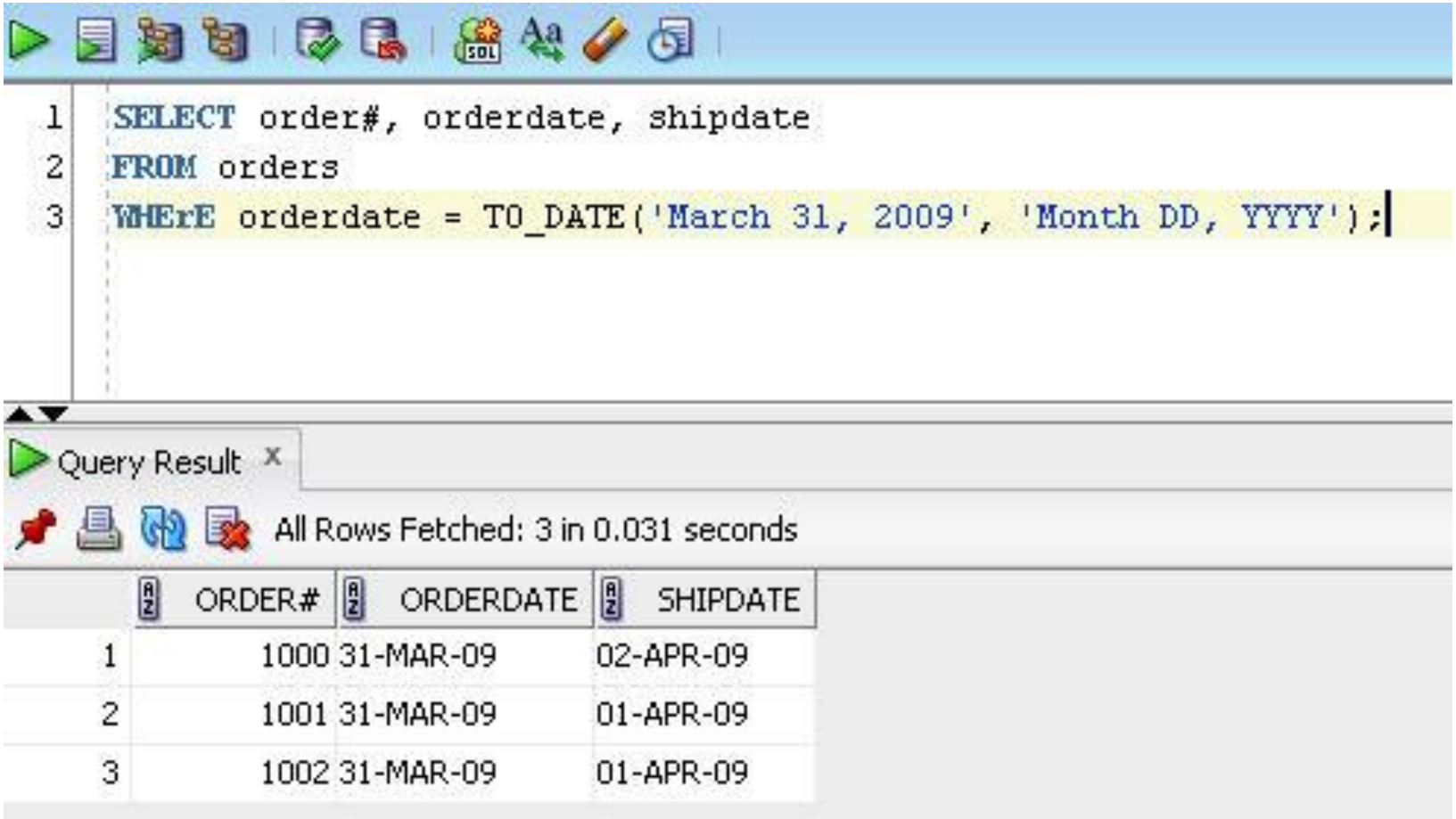
DATE FORMATS		
Element	Description	Example
MONTH	Name of the month spelled out—padded with blank spaces to a total width of nine spaces	APRIL
MON	Three-letter abbreviation for the name of the month	APR
MM	Two-digit numeric value of the month	04
RM	Roman numeral month	IV
D	Numeric value for the day of the week	Wednesday = 4
DD	Numeric value for the day of the month	28
DDD	Numeric value for the day of the year	December 31 = 365
DAY	Name of the day of the week—padded with blank spaces to a length of nine characters	Wednesday
DY	Three-letter abbreviation for the day of the week	WED
YYYY	Displays the Four-digit numeric value of the year	2004
YYY or YY or Y	The last three, two, or single digit(s) of the year	2004 = 004; 2004 = 04; 2004 = 4
YEAR	Spelled-out version of the year	TWO THOUSAND FOUR
B.C. or A.D.	Value indicating B.C. or A.D.	2004 A.D.

FIGURE 10-25 Data format elements

TO_DATE FUNCTION

- When working with the TO_DATE function, the user enters the date as the first argument and a format mask or model that allows Oracle to distinguish the date components entered
- Both the date and the format of the date are character strings so each argument is enclosed in *single quotes*
- In our example, the user will enter the date in the format of MONTH, DD, YYYY, this will be indicated in the format mask
- The TO_DATE function will be specified in the WHERE clause

TO_DATE FUNCTION



```
1  SELECT order#, orderdate, shipdate
2  FROM orders
3  WHErE orderdate = TO_DATE('March 31, 2009', 'Month DD, YYYY');
```

Query Result

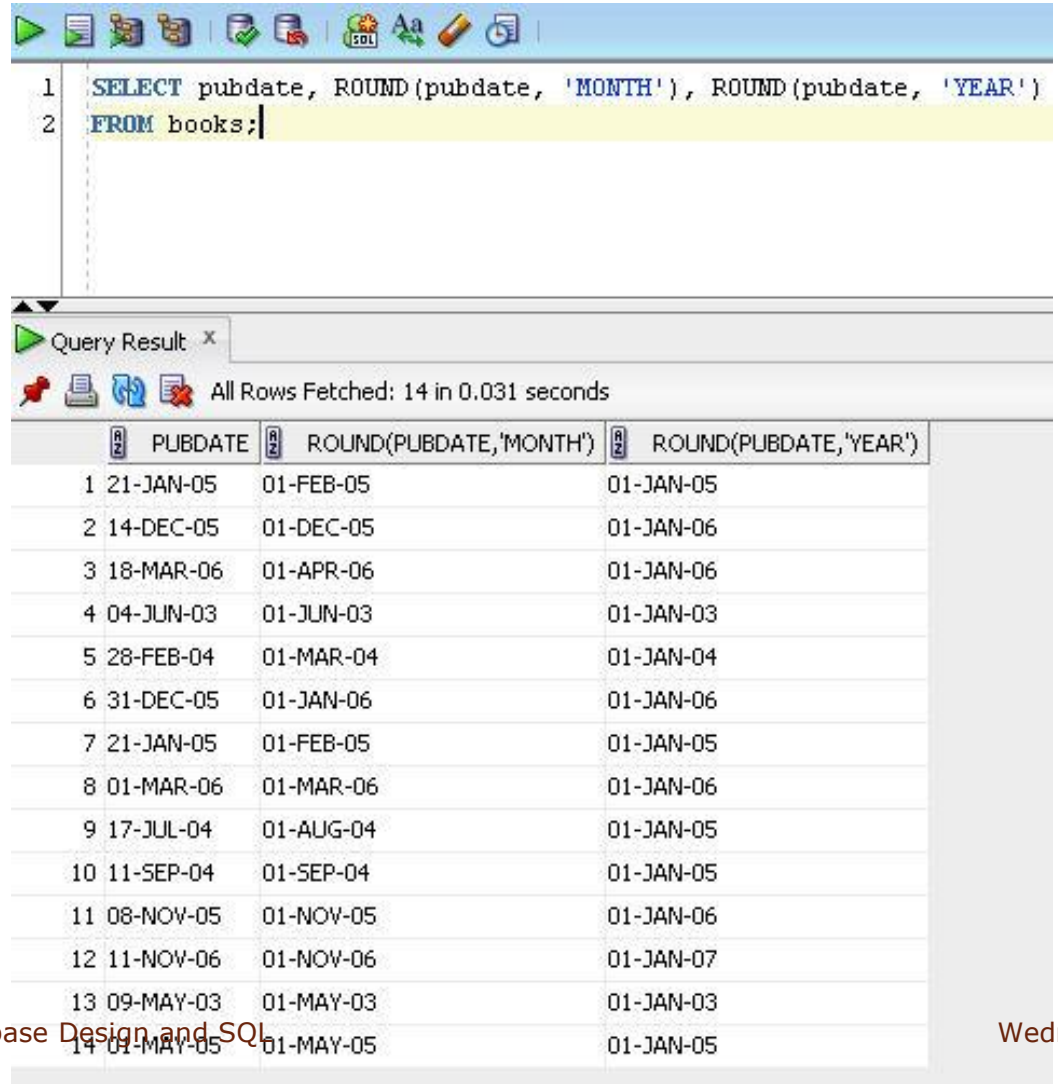
All Rows Fetched: 3 in 0.031 seconds

	ORDER#	ORDERDATE	SHIPDATE
1	1000	31-MAR-09	02-APR-09
2	1001	31-MAR-09	01-APR-09
3	1002	31-MAR-09	01-APR-09

ROUNDING DATE VALUES

- Even though the ROUND function is typically associated with numeric data you can use it on date values as well
- The syntax of the ROUND function is:
  - ROUND(d, u) where d is the date data and u represents the unit to use for rounding
- The date can be rounded by the unit month and year
- For month rounding if the day of the month is 16 or later, the date is rounded to the first of the next month
- If the day of the month is less than 16 the date is rounded to the first of that month
- Year rounding is similar with July 16th as the cutoff to determine rounding
- The 'YEAR' can also be represented as 'YYYY' and 'MONTH can be represented as 'MM'

ROUNDING DATE VALUES



```
1  SELECT pubdate, ROUND(pubdate, 'MONTH'), ROUND(pubdate, 'YEAR')
2  FROM books;
```

Query Result

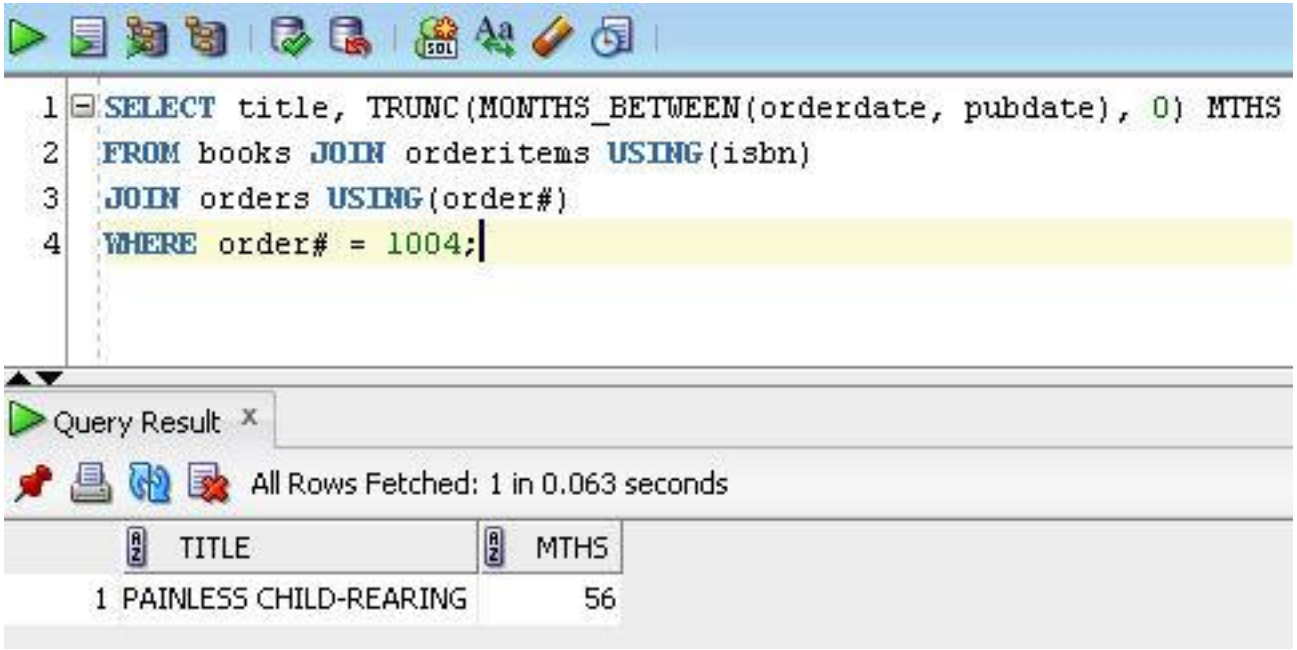
All Rows Fetched: 14 in 0.031 seconds

	PUBDATE	ROUND(PUBDATE,'MONTH')	ROUND(PUBDATE,'YEAR')
1	21-JAN-05	01-FEB-05	01-JAN-05
2	14-DEC-05	01-DEC-05	01-JAN-06
3	18-MAR-06	01-APR-06	01-JAN-06
4	04-JUN-03	01-JUN-03	01-JAN-03
5	28-FEB-04	01-MAR-04	01-JAN-04
6	31-DEC-05	01-JAN-06	01-JAN-06
7	21-JAN-05	01-FEB-05	01-JAN-05
8	01-MAR-06	01-MAR-06	01-JAN-06
9	17-JUL-04	01-AUG-04	01-JAN-05
10	11-SEP-04	01-SEP-04	01-JAN-05
11	08-NOV-05	01-NOV-05	01-JAN-06
12	11-NOV-06	01-NOV-06	01-JAN-07
13	09-MAY-03	01-MAY-03	01-JAN-03
14	01-MAY-05	01-MAY-05	01-JAN-05

TRUNCATING DATE VALUES

- The TRUNC function that is used for truncating numeric results can also be quite useful in date calculations and is applied to dates by using the unit of month or year
- The query shown has been used before on slide 67 where a large decimal portion was left
- To eliminate this decimal portion of the output s only the number of whole months is left the TRUNC function can be used
- The two functions being used together is an example of nesting functions, the value returned by the MONTHS_BETWEEN function is used as an argument for the TRUNC function, TRUNC is done so the entire decimal portion is removed and only the whole number of months remains

TRUNCATING DATE VALUES



NESTING FUNCTIONS

- Any single-row function can be nested inside other single-row functions
- Nesting means that one function is used as an argument inside another function
- There are some rules to follow:
  1. All arguments required for each function must be provided
  2. For every open parenthesis, there must be a corresponding close parenthesis
  3. The nested or inner function is solved first, the result of the inner function is passed to the outer function, and then the outer function is executed

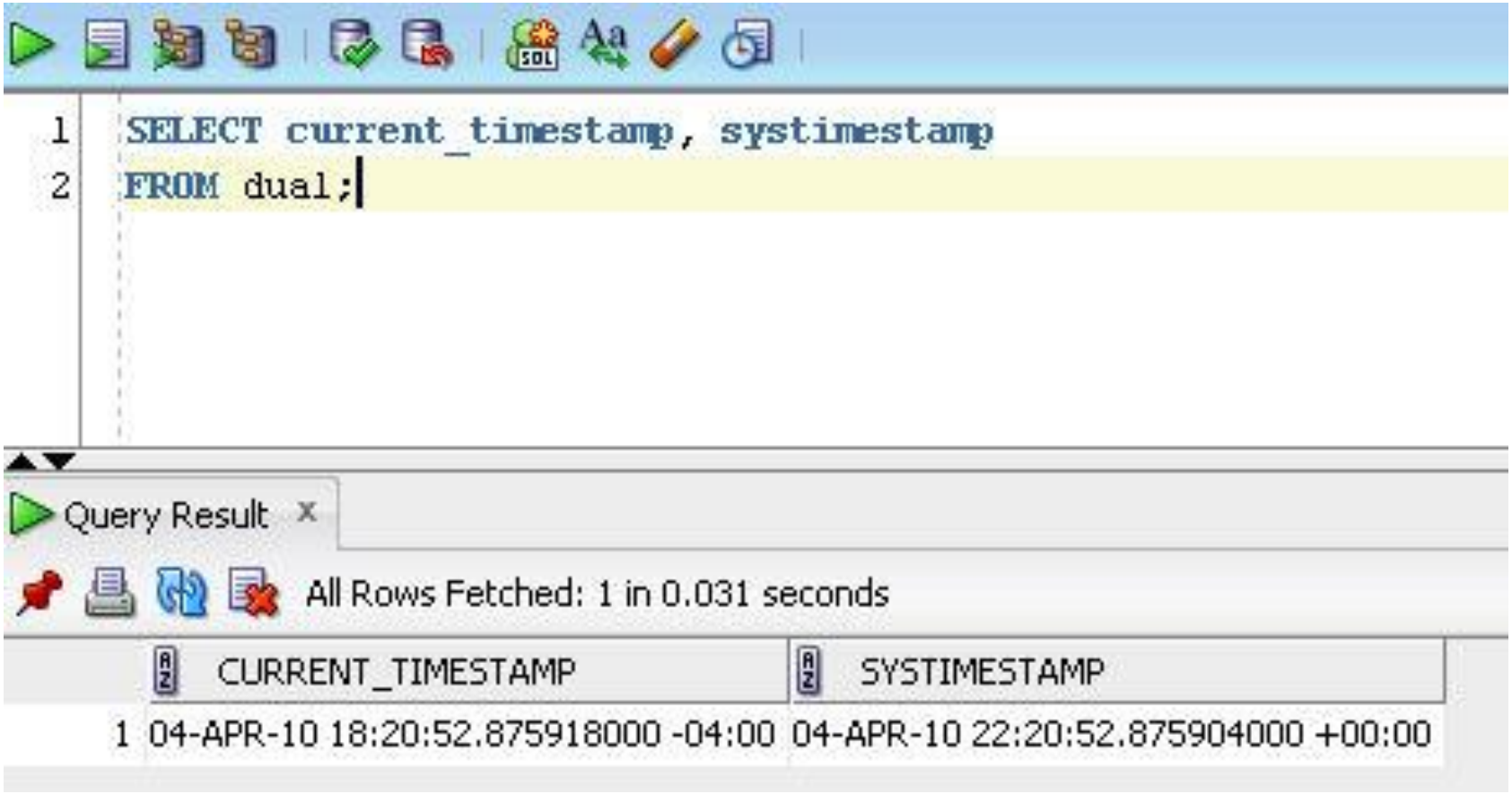
NESTING FUNCTIONS

- TRUNC function has *two arguments*, the value to be truncated and the position the truncation should occur
- The value to truncate is the result of the MONTHS_BETWEEN function
- This requires that the the MONTHS_BETWEEN function be entered as the first argument of the TRUNC function
- The 0 indicates that there are to be no decimal places in the output after truncation occurs
- The value computed by the *inner function* is used to compute the *outer function*
- The result is a whole number

CURRENT_DATE VS. SYSDATE

- Many companies conduct transactions in different parts of he world so time zone recognition is essential
- Both CURRENT_DATE and SYSDATE functions identify the current date and time
- SYSDATE returns the current date and time set on the operating system where the database resides
- The CURRENT_DATE function returns the current date and time from the user session
- To demonstrate this our database is residing on Munro while SQL Developer is running locally on your computer
- Look at the output of the next slide, you will notice the two values are different, server is on GMT and the terminal is EST

CURRENT_DATE VS. SYSDATE



```
1  SELECT current_timestamp, systimestamp
2  FROM dual;
```

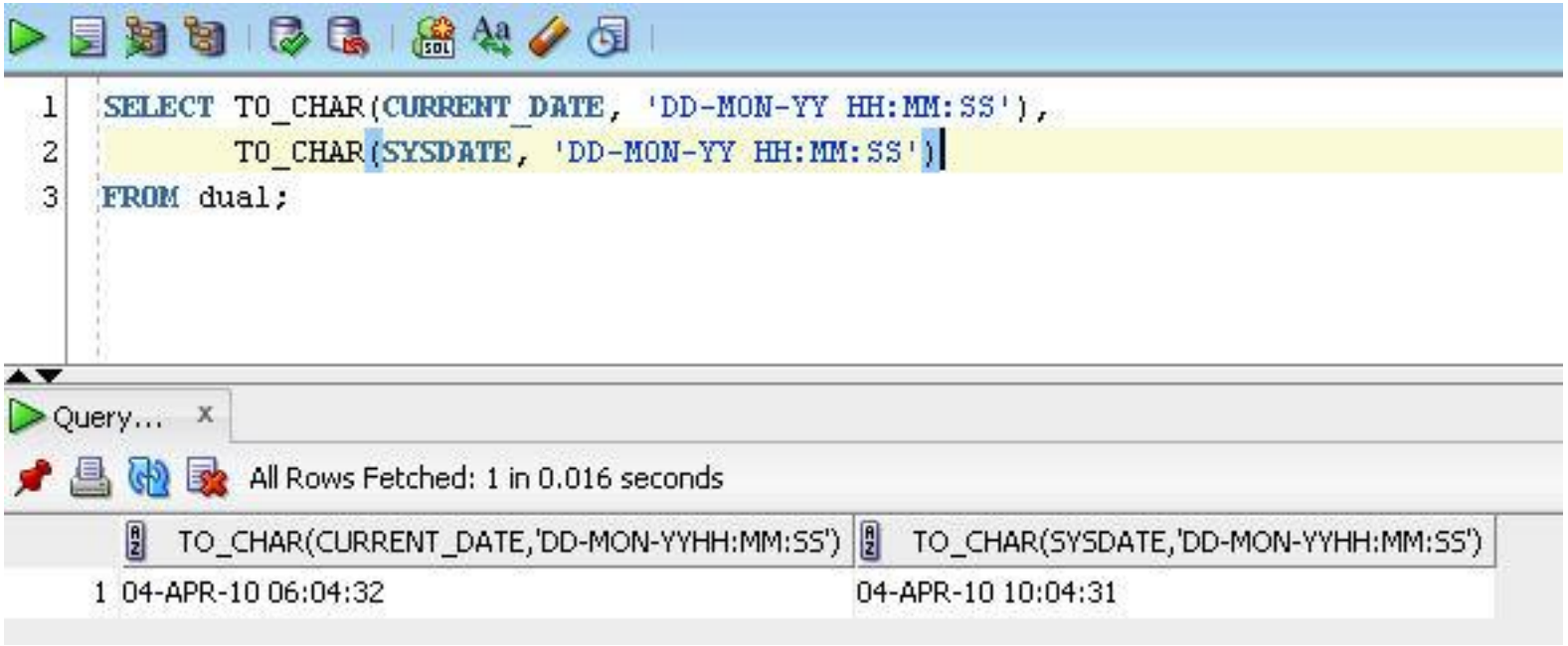
Query Result

All Rows Fetched: 1 in 0.031 seconds

	CURRENT_TIMESTAMP	SYSTIMESTAMP
1	04-APR-10 18:20:52.875918000 -04:00	04-APR-10 22:20:52.875904000 +00:00

CURRENT_DATE VS. SYSDATE

The time I entered this command was 6:04 PM, CURRENT_DATE reflects my computer's time, SYSDATE is Munro's time



OTHER FUNCTIONS

- Some functions provided with Oracle 12c do not fall into character, numeric or date categories
- We will discuss five additional functions:
  - NVL
  - NVL2
  - TO_CHAR
  - DECODE
  - CASE

NVL FUNCTION

- The NVL function can be used to address problems that arise when performing arithmetic operations on numeric fields that contain NULL values
- As you recall, a NULL value is an absence of data or an unknown value
- If a NULL value is used in a calculation, the result is also a NULL value
- The NVL function allows you to you to substitute a value for an existing NULL value
- The syntax for NVL function is:
  - NVL(x, y)
    - Where y represents the value to be substituted if x is NULL
    - In many cases the substitute for a NULL value is 0

NVL FUNCTION

```
1  SELECT title, retail, discount, retail - discount "Sale Price"
2  FROM books;
```

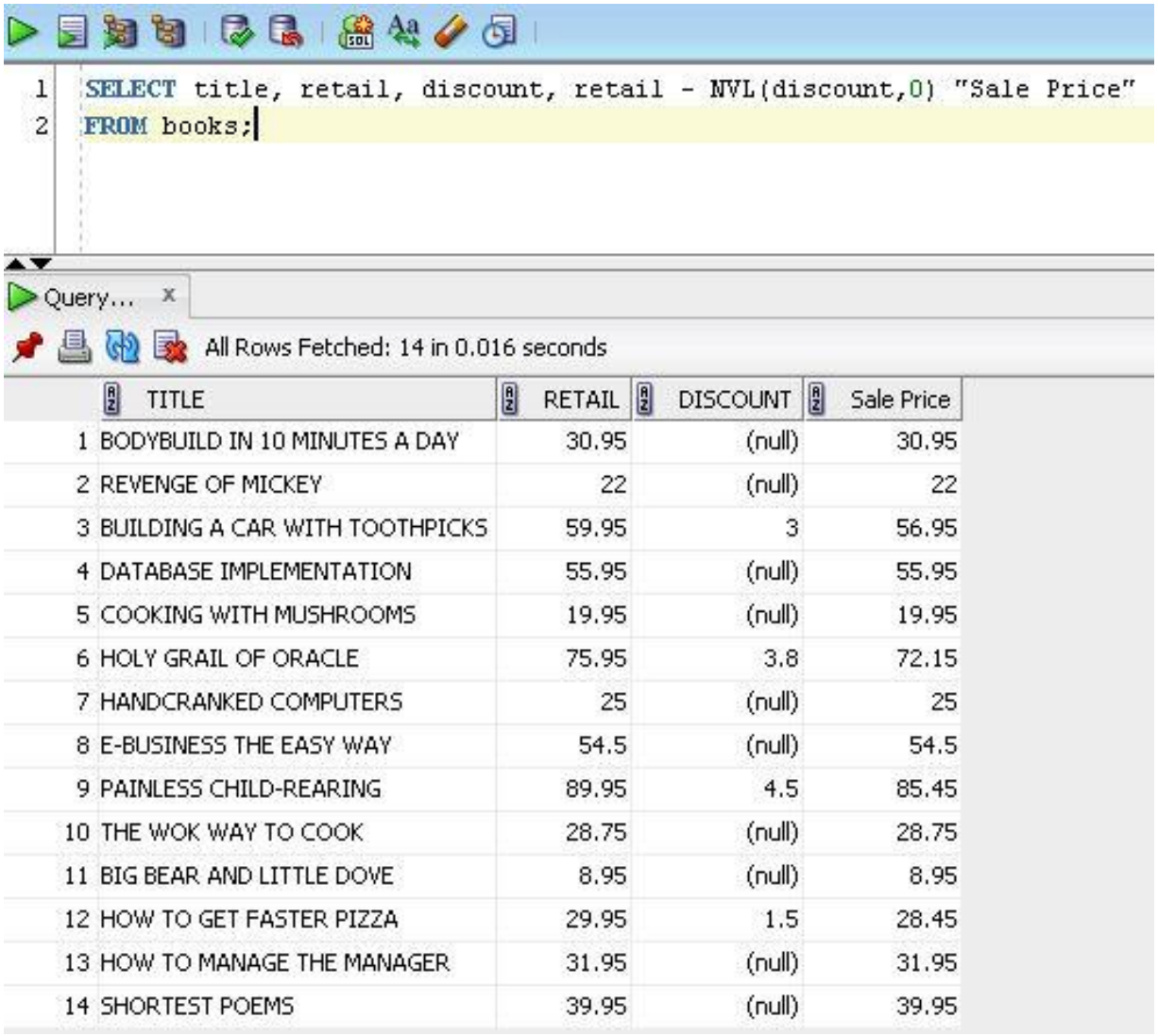
Query... x

All Rows Fetched: 14 in 0.015 seconds

	TITLE	RETAIL	DISCOUNT	Sale Price
1	BODYBUILD IN 10 MINUTES A DAY	30.95	(null)	(null)
2	REVENGE OF MICKEY	22	(null)	(null)
3	BUILDING A CAR WITH TOOTHPICKS	59.95	3	56.95
4	DATABASE IMPLEMENTATION	55.95	(null)	(null)
5	COOKING WITH MUSHROOMS	19.95	(null)	(null)
6	HOLY GRAIL OF ORACLE	75.95	3.8	72.15
7	HANDCRANKED COMPUTERS	25	(null)	(null)
8	E-BUSINESS THE EASY WAY	54.5	(null)	(null)
9	PAINLESS CHILD-REARING	89.95	4.5	85.45
10	THE WOK WAY TO COOK	28.75	(null)	(null)
11	BIG BEAR AND LITTLE DOVE	8.95	(null)	(null)
12	HOW TO GET FASTER PIZZA	29.95	1.5	28.45
13	HOW TO MANAGE THE MANAGER	31.95	(null)	(null)
14	SHORTEST POEMS	39.95	(null)	(null)

Notice that each book with a NULL discount value shows a null sales price as a result of the calculation

NVL FUNCTION



```
1  SELECT title, retail, discount, retail - NVL(discount,0) "Sale Price"
2  FROM books;
```

Query...

All Rows Fetched: 14 in 0.016 seconds

	TITLE	RETAIL	DISCOUNT	Sale Price
1	BODYBUILD IN 10 MINUTES A DAY	30.95	(null)	30.95
2	REVENGE OF MICKEY	22	(null)	22
3	BUILDING A CAR WITH TOOTHPICKS	59.95	3	56.95
4	DATABASE IMPLEMENTATION	55.95	(null)	55.95
5	COOKING WITH MUSHROOMS	19.95	(null)	19.95
6	HOLY GRAIL OF ORACLE	75.95	3.8	72.15
7	HANDCRANKED COMPUTERS	25	(null)	25
8	E-BUSINESS THE EASY WAY	54.5	(null)	54.5
9	PAINLESS CHILD-REARING	89.95	4.5	85.45
10	THE WOK WAY TO COOK	28.75	(null)	28.75
11	BIG BEAR AND LITTLE DOVE	8.95	(null)	8.95
12	HOW TO GET FASTER PIZZA	29.95	1.5	28.45
13	HOW TO MANAGE THE MANAGER	31.95	(null)	31.95
14	SHORTEST POEMS	39.95	(null)	39.95

If the discount value is now a 0 (zero) the NVL function converts to a 0 then does the calculation

The result is a value for the Sale Price

For every NULL discount price it is made into a 0 for the calculation, the actual data is not modified

NVL FUNCTION

- Another example to demonstrate the use of the NVL function is to look at the orders table for Just Lee Books
- Some orders have not been shipped, so the value of shipdate is NULL
- Since every order is to be shipped on the following Monday, the NVL function could be used to substitute Monday's date so a value will appear for all records

NVL FUNCTION

```
1  SELECT order#, orderdate, shipdate,
2         NVL(shipdate, '06-APR-09') - orderdate "Ship Days"
3  FROM orders
4  WHERE orderdate >= '03-APR-09';
```

Query...

All Rows Fetched: 12 in 0.031 seconds

	ORDER#	ORDERDATE	SHIPDATE	Ship Days
1	1009	03-APR-09	05-APR-09	2
2	1010	03-APR-09	04-APR-09	1
3	1011	03-APR-09	05-APR-09	2
4	1012	03-APR-09	(null)	3
5	1013	03-APR-09	04-APR-09	1
6	1014	04-APR-09	05-APR-09	1
7	1015	04-APR-09	(null)	2
8	1016	04-APR-09	(null)	2
9	1017	04-APR-09	05-APR-09	1
10	1018	05-APR-09	(null)	1
11	1019	05-APR-09	(null)	1
12	1020	05-APR-09	(null)	1

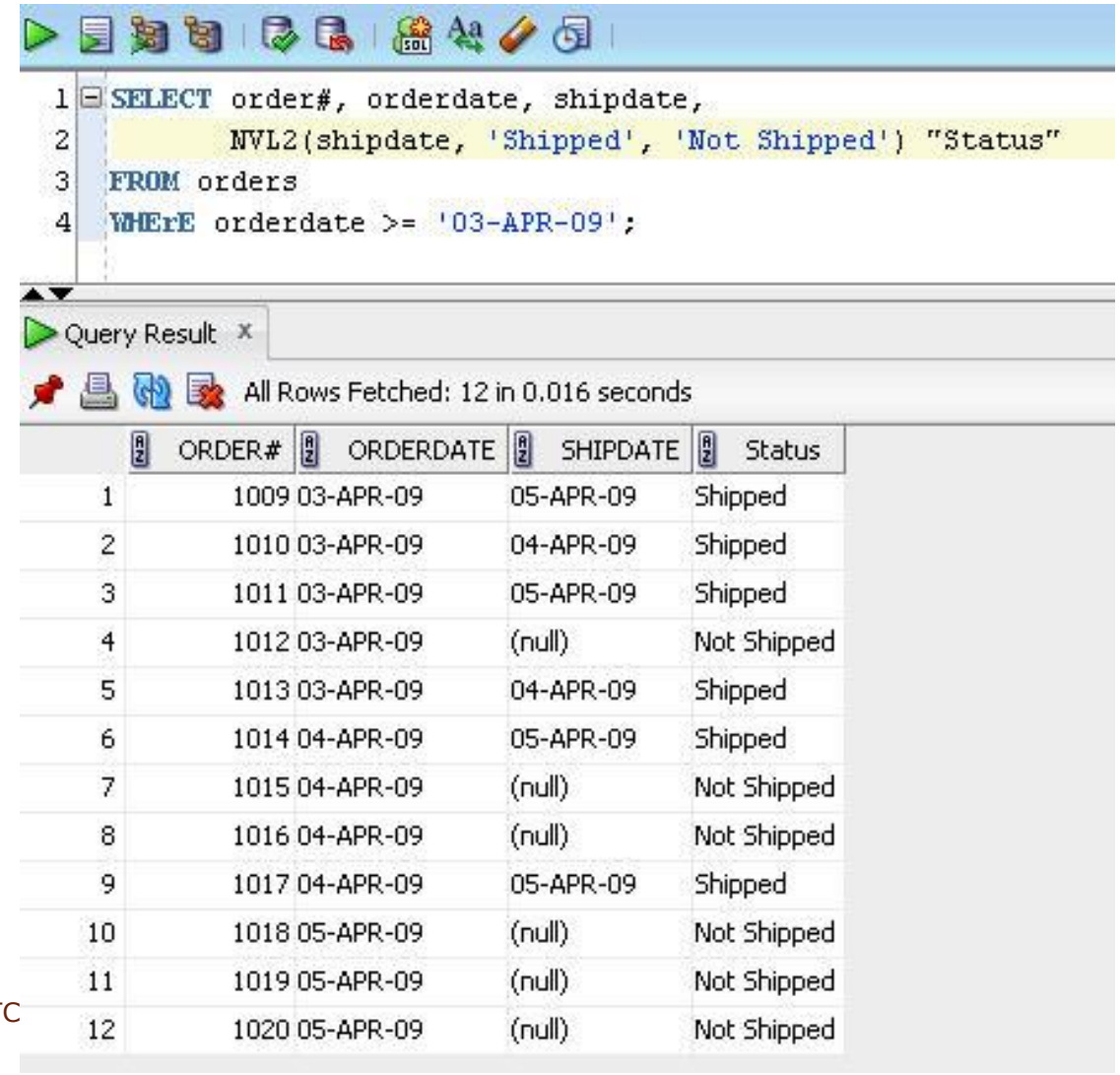
Since the shipdate for some orders is a NULL value, the delay value is also NULL

The anticipated ship date is now included using the NVL function, so a value for Delay appears for each record

NVL2 FUNCTION

- The NVL2 function is a variation of the NVL function that allows different options based on whether a NULL value exits
- The syntax of the NVL2 function is:
  - NVL2(x, y, z)
    - Where y represents what should be substituted is x is not NULL and z represents what should be substituted if x is NULL
    - This allows the user a little more flexibility when working with NULL values

NVL2 FUNCTION



```
1  SELECT order#, orderdate, shipdate,
2         NVL2(shipdate, 'Shipped', 'Not Shipped') "Status"
3  FROM orders
4  WHERE orderdate >= '03-APR-09';
```

Query Result

All Rows Fetched: 12 in 0.016 seconds

	ORDER#	ORDERDATE	SHIPDATE	Status
1	1009	03-APR-09	05-APR-09	Shipped
2	1010	03-APR-09	04-APR-09	Shipped
3	1011	03-APR-09	05-APR-09	Shipped
4	1012	03-APR-09	(null)	Not Shipped
5	1013	03-APR-09	04-APR-09	Shipped
6	1014	04-APR-09	05-APR-09	Shipped
7	1015	04-APR-09	(null)	Not Shipped
8	1016	04-APR-09	(null)	Not Shipped
9	1017	04-APR-09	05-APR-09	Shipped
10	1018	05-APR-09	(null)	Not Shipped
11	1019	05-APR-09	(null)	Not Shipped
12	1020	05-APR-09	(null)	Not Shipped

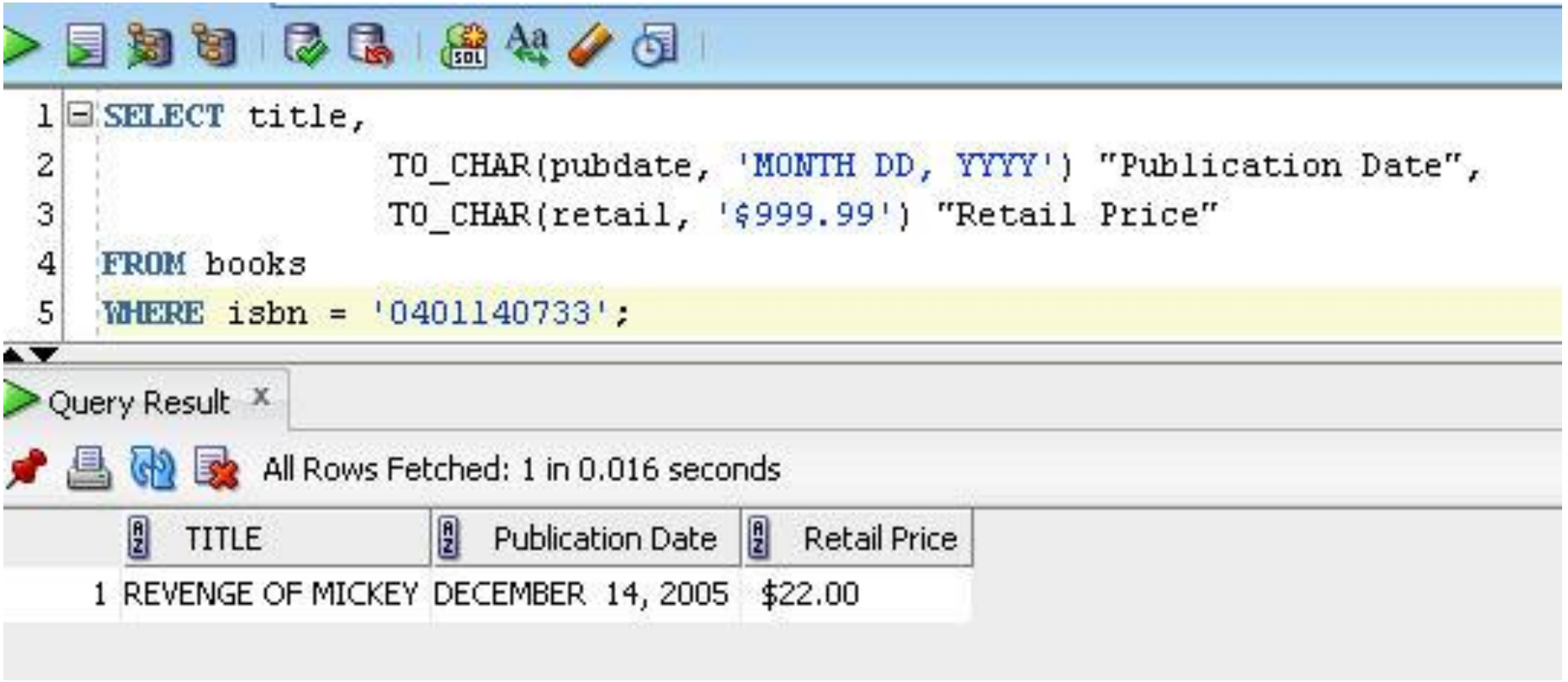
If the SHIPDATE is not null, display Shipped

If the SHIPDATE is a null value, display the string Not Shipped

TO_CHAR FUNCTION

- The TO_CHAR function is widely used to convert dates and numbers to a formatted character string
- It is the opposite of the TO_DATE function
- The TO_DATE function allows a user to enter a date in any type of format, whereas the TO_CHAR function is used to have Oracle 12c display dates in a particular format
- The syntax of the TO_CHAR function is:
  - TO_CHAR(n,'f')
    - Where n represents the date or number to be formatted and f represents the *format mask* or model to be used
    - The *format mask* consists of a series of elements that represent how the date should appear when displayed

TO_CHAR FUNCTION



```
1  SELECT title,
2                 TO_CHAR(pubdate, 'MONTH DD, YYYY') "Publication Date",
3                 TO_CHAR(retail, '$999.99') "Retail Price"
4  FROM books
5  WHERE isbn = '0401140733';
```

Query Result

All Rows Fetched: 1 in 0.016 seconds

	TITLE	Publication Date	Retail Price
1	REVENGE OF MICKEY	DECEMBER 14, 2005	$22.00

FORMATS		
Element	Description	Example
Date Elements		
MONTH	Name of the month spelled out—padded with blank spaces to a total width of nine spaces	APRIL
MON	Three-letter abbreviation for the name of the month	APR
MM	Two-digit numeric value of the month	04
RM	Roman numeral month	IV
D	Numeric value for the day of the week	Wednesday = 4
DD	Numeric value for the day of the month	28
DDD	Numeric value for the day of the year	December 31 = 365
DAY	Name of the day of the week—padded with blank spaces to a length of nine characters	Wednesday
DY	Three-letter abbreviation for the day of the week	WED
YYYY	Numeric value for the four-digit year	2004
YYY or YY or Y	Numeric value for the last three, two, or single digit(s) of year	2004 = 004; 2004=04; the 2004 = 4
YEAR	Spelled-out version of the year	TWO THOUSAND FOUR
BC or AD	Value indicating B.C. or A.D.	2004 A.D.
Time Elements		
SS	Seconds	Value between 0–59
SSSS	Seconds past midnight	Value between 0–86399
MI	Minutes	Value between 0–59
HH or HH12	Hours	Value between 1–12
HH24	Hours	Value between 0–23
A.M. or P.M.	Value indicating morning or evening hours	A.M. (before noon) or P.M. (after noon)
Number Elements		
9	Indicates width of display with a series of 9s, but insignificant leading zeros are not displayed	99999
0	Displays insignificant leading zeros	0009999
$	Displays a floating dollar sign	$99999
.	Indicates number of decimals to display	999.99
,	Displays a comma in the position indicated	9,999

FIGURE 10-38 Format model elements

TO_CHAR FUNCTION

FORMATS		
Element	Description	Example
Other Elements		
, . (punctuation symbols)	Displays indicated punctuation	DD, YYYY = 24, 2001
"string"	Displays the exact character string inside the double quotation marks	"of the year" YYYY = of the year 2001
TH	Displays the ordinal number	DDTH = 8th
SP	Spells out the number	DDSP = EIGHT
SPTH	Spell out the ordinal number	DDSPTH = EIGHTH

FIGURE 10-38 Format model elements (continued)

DECODE FUNCTION

- The DECODE function takes a specified value and compares it to values in a list
- If a match is found the specified result is returned
- The DECODE function enables you to specify different actions to take depending on the circumstances
- It saves you from having to enter multiple statements for each possible situation
- The syntax of the DECODE is:
  - DECODE(V, LI, R1, L2, R2, … D)
  - Where V is the value you are searching for
  - L1 represents the first value in the list
  - R1 represents the result to return if L1 and V are equivalent
  - D is the default value to return if no match is found

DECODE FUNCTION

```
1  SELECT customer#, state,
2         DECODE (state, 'CA', .08,
3                        'FL', .07,
4                              0) "Sales Tax Rate"
5  FROM customers
6  WHERE state IN ('CA', 'FL', 'GA', 'TX')
7  ORDER BY state;
```

Query Result

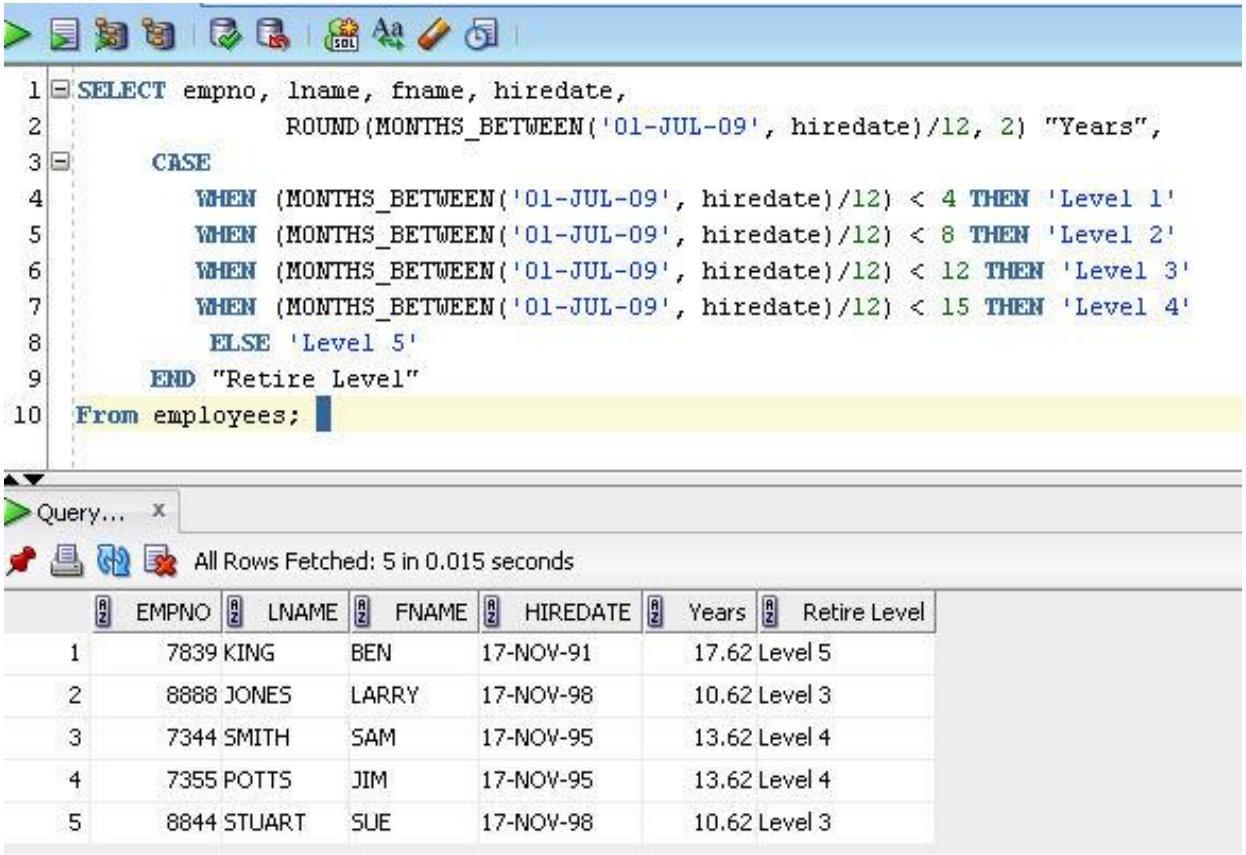
All Rows Fetched: 10 in 0.016 seconds

	CUSTOMER#	STATE	Sales Tax Rate
1	1009	CA	0.08
2	1016	CA	0.08
3	1002	CA	0.08
4	1013	FL	0.07
5	1003	FL	0.07
6	1015	FL	0.07
7	1001	FL	0.07
8	1018	GA	0
9	1010	GA	0
10	1007	TX	0

CASE EXPRESSION

- The CASE expression is similar to the DECODE function in that it can perform IF … THEN … ELSE conditional processing
- Both the simple CASE expression and the DECODE function evaluate equality conditions
- However the searched CASE expression gives you more flexibility because it allows other comparisons besides equality
- You might need to determine the retirement level assigned to employees based on the number of years employed at Just Lee Books
- These levels are determined using ranges of years

CASE EXPRESSION



```
1  SELECT empno, lname, fname, hiredate,
2                 ROUND(MONTHS_BETWEEN('01-JUL-09', hiredate)/12, 2) "Years",
3       CASE
4          WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 4 THEN 'Level 1'
5          WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 8 THEN 'Level 2'
6          WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 12 THEN 'Level 3'
7          WHEN (MONTHS_BETWEEN('01-JUL-09', hiredate)/12) < 15 THEN 'Level 4'
8           ELSE 'Level 5'
9       END "Retire Level"
10 From employees;
```

Query...

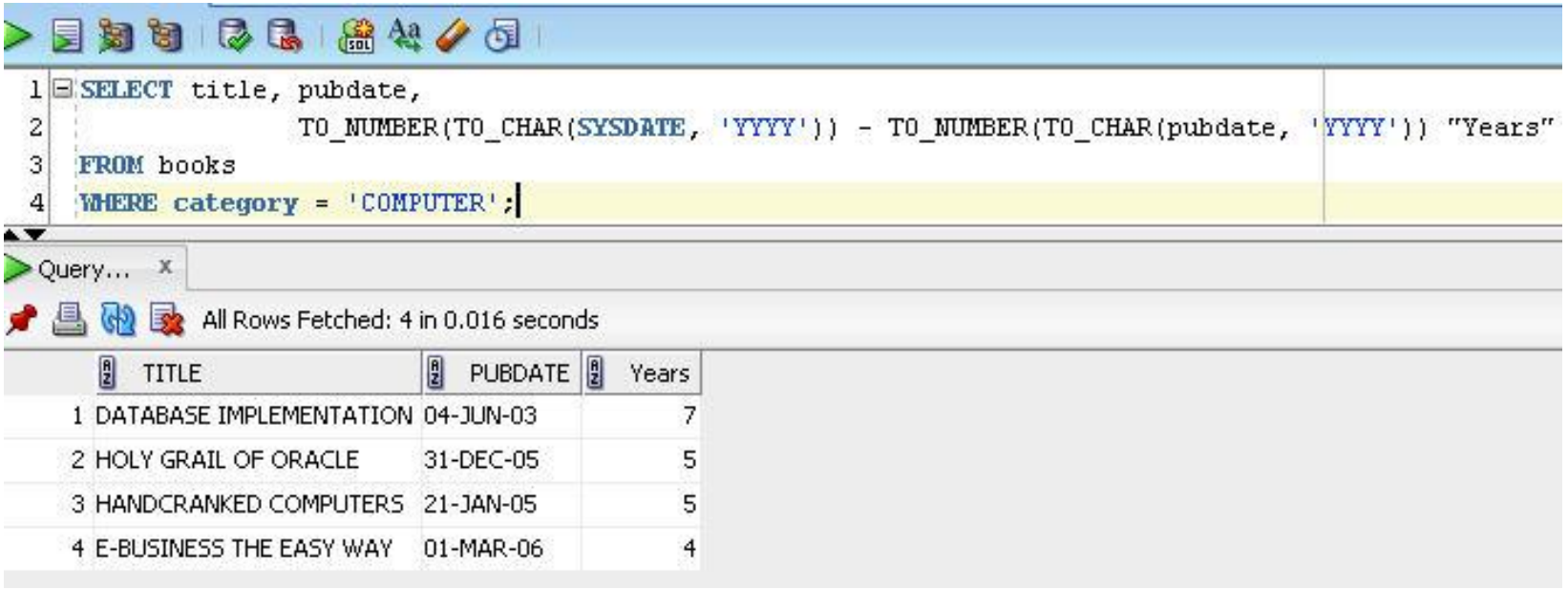
All Rows Fetched: 5 in 0.015 seconds

	EMPNO	LNAME	FNAME	HIREDATE	Years	Retire Level
1	7839	KING	BEN	17-NOV-91	17.62	Level 5
2	8888	JONES	LARRY	17-NOV-98	10.62	Level 3
3	7344	SMITH	SAM	17-NOV-95	13.62	Level 4
4	7355	POTTS	JIM	17-NOV-95	13.62	Level 4
5	8844	STUART	SUE	17-NOV-98	10.62	Level 3

THE TO_NUMBER FUNCTION

- The TO_NUMBER function converts a value to a numeric datatype, if possible
- The string value of 2009 stored in a date or character string could be converted to a numeric datatype to use in calculation

THE TO_NUMBER FUNCTION



```
1  SELECT title, pubdate,
2                 TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - TO_NUMBER(TO_CHAR(pubdate, 'YYYY')) "Years"
3  FROM books
4  WHERE category = 'COMPUTER';
```

Query...

All Rows Fetched: 4 in 0.016 seconds

	TITLE	PUBDATE	Years
1	DATABASE IMPLEMENTATION	04-JUN-03	7
2	HOLY GRAIL OF ORACLE	31-DEC-05	5
3	HANDCRANKED COMPUTERS	21-JAN-05	5
4	E-BUSINESS THE EASY WAY	01-MAR-06	4

THE DUAL TABLE

- Any of the single-row functions covered in this chapter can be used with the DUAL table
- Although DUAL is rarely used in industry it can be valuable for someone learning how to work with functions or testing new function
- Oracle has the mandatory FROM clause to specify a table name in a query, DUAL can be that table

THE DUAL TABLE