



Database Management

BU.330.770

Session 6

Instructor: Changmi Jung, Ph.D.

Announcement



» Assignment #3 is due next week (before the class starts)

- Canvas > Week 7

» Final exam on week 8 (Dec. 18th)

- Cannot reschedule the exam for any reason & Cannot take the test online
- One cheat sheet allowed:
 - One sheet of letter-sized paper (not A4): 8.5 x 11 inches
 - Can fill both front and back
 - Any font size is acceptable
 - Can copy/paste from your lecture notes, but it must be your own work
 - Your name should be on the top right corner



Single-Row Functions



Session Objectives (1/3)

- » Use the UPPER, LOWER, and INITCAP functions to change the case of field values and character strings
- » Manipulate character substrings with the functions
- » Nest functions inside other functions SUBSTR and INSTR
- » 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 characters strings
- » Substitute character string values with the REPLACE function



Session Objectives (2/3)

- » Round and truncate numeric data using the ROUND and TRUNC functions
- » Return the remainder only of a division operation using the MOD function
- » Use the ABS function to set numeric values as positive
- » Calculate the number of months between two dates using the MONTHS_BETWEEN function
- » Manipulate date data using the ADD_MONTHS, NEXT_DAY, and TO_DATE functions



Session Objectives (3/3)

- » Identify and correct problems associated with calculations involving NULL values using the NVL function
- » Display dates and numbers in a specific format with the TO_CHAR function
- » Convert string values to numeric with the TO_NUMBER function
- » Use the DUAL table to test functions



Recap: DUAL Table

- » Dummy table
- » Consists of one column and one row
- » Can be used for table reference in the FROM clause

Worksheet		Query Builder	
		<pre>SELECT 25+5, sysdate, LENGTH('Hello') FROM dual;</pre>	
		Query Result x	
		SQL All Rows Fetched: 1 in 0.201 seconds	
	25+5	SYSDATE	LENGTH('HELLO')
1	30	25-FEB-23	5

Terminology



» Function – predefined block of code that accepts arguments and returns a single value as output

- Arguments: values listed inside parentheses of a function
- $F(\text{argument}) = \text{output}$



» Single-row function: accepts a single value as an input, and then returns one row of result for each record processed

» Multiple-row function: accepts multiple values as an input and returns one result per group of data processed

Types of Functions



Type of Function	Functions
Case conversion	UPPER, LOWER, INITCAP
Character manipulation	SUBSTR, INSTR, LENGTH, LPAD/RPAD, LTRIM/RTRIM, REPLACE
Numeric functions	ROUND, TRUNC, MOD, ABS
Date functions	MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY ...
Type conversion	TO_DATE, TO_CHAR, TO_NUMBER
Other functions	NVL, NVL2, DECODE, CASE Expression



Case Conversion Functions

Alter the case of a character string (either data stored in a column or character string)

- Used in a SELECT clause: they alter the appearance of the data in the results
- Used in a WHERE clause: they alter the value for comparison

» **LOWER(c)**: converts characters into lower case letters

» **UPPER(c)**: converts characters into upper case letters

» **INITCAP(c)**: converts characters into upper (first) + lower case (the rest) for each word

- Where *c* is the character string or column (field) to be converted into lowercase (or upper) characters



Use Cases

» Customer Data Entry

Ready to Build Better Demos?

Chat with our sales team to learn how Reprise can empower your team to build personalized, interactive demos that lead to greater conversions, shorter sales cycles and more revenue.

First name*

Last name*

Company name*

Company size*

Company email*

☐ I would like to receive marketing emails from Reprise to help me create winning demos!

By clicking submit you are agreeing to the Reprise [Privacy Policy](#).

[Book Your Demo](#)

» Data Search: Carey Faculty Directory

Examples in SELECT Clause




Worksheet

Query Builder

```
SELECT firstname, lastname, LOWER(lastname), UPPER(lastname), INITCAP(lastname)
FROM customers;
```

Query Result x

 | All Rows Fetched: 20 in 0.072 seconds

	FIRSTNAME	LASTNAME	LOWER(LASTNAME)	UPPER(LASTNAME)	INITCAP(LASTNAME)
1	BONITA	MORALES	moraless	MORALES	Moraless
2	RYAN	THOMPSON	thompson	THOMPSON	Thompson
3	LEILA	SMITH	smith	SMITH	Smith
4	THOMAS	PIERSON	pierson	PIERSON	Pierson
5	CINDY	GIRARD	girard	GIRARD	Girard
6	MESHIA	CRUZ	cruz	CRUZ	Cruz
7	TAMMY	GIANA	giana	GIANA	Giana
8	KENNETH	JONES	jones	JONES	Jones
9	JORGE	PEREZ	perez	PEREZ	Perez
10	JAKE	LUCAS	lucas	LUCAS	Lucas
11	REESE	MCGOVERN	mcgovern	MCGOVERN	Mcgovern
12	WILLIAM	MCKENZIE	mckenzie	MCKENZIE	Mckenzie
13	NICHOLAS	NGUYEN	nguyen	NGUYEN	Nguyen
14	JASMINE	LEE	lee	LEE	Lee
15	STEVE	SHELL	schell	SHELL	Schell
16	MICHELL	DAUM	daum	DAUM	Daum
17	BECCA	NELSON	nelson	NELSON	Nelson
18	DEBBIE	MONTAGNA	montagna	MONTAGNA	Montagna



LOWER Function Example in WHERE

» Used to convert characters to lowercase letters

The screenshot shows a database query builder interface. At the top, there are two tabs: 'Worksheet' and 'Query Builder'. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT firstname, lastname  
FROM customers  
WHERE LOWER(lastname) = 'nelson';
```

Below the query editor, there is a 'Query Result' window. It shows the execution status: 'All Rows Fetched: 1 in 0.035 seconds'. Below this, a table displays the results of the query:

	FIRSTNAME	LASTNAME
1	BECCA	NELSON

Converts data in the 'lastname' column into lower case

If the space and case of character strings do not match with the comparison values we specify in search condition, the search result will not contain any record.

UPPER Function Example







Worksheet

Query Builder

```
SELECT firstname, lastname
FROM customers
WHERE lastname = UPPER('nelson');
```

Query Result x



SQL | All Rows Fetched: 1 in 0.035 seconds

	FIRSTNAME	LASTNAME
1	BECCA	NELSON

Converts the character string 'nelson' into the upper case: the result is 'NELSON'





Worksheet

Query Builder

```
SELECT firstname, lastname
FROM customers
WHERE lastname = UPPER('&Last_Name');
```

▼

▶ Query Result x



SQL | All Rows Fetched: 1 in 0.02 seconds

	FIRSTNAME	LASTNAME
1	BECCA	NELSON

Enter Substitution Variable

Enter value for Last_Name:

Nelson

OKCancel



INITCAP Function

- » Converts characters to the mixed case: the first letter of each word in the character string to uppercase and the remaining letters into lowercase

Worksheet Query Builder	
<pre>SELECT title, INITCAP(title) "Book Title" FROM books WHERE ROWNUM <= 5 ORDER BY title;</pre>	
Query Result x	
SQL All Rows Fetched: 5 in 0.032 seconds	
TITLE	Book Title
1 BODYBUILD IN 10 MINUTES A DAY	Bodybuild In 10 Minutes A Day
2 BUILDING A CAR WITH TOOTHPICKS	Building A Car With Toothpicks
3 COOKING WITH MUSHROOMS	Cooking With Mushrooms
4 DATABASE IMPLEMENTATION	Database Implementation
5 REVENGE OF MICKEY	Revenge Of Mickey



Character Manipulation Functions

» Character manipulation functions manipulate data by extracting substrings, finding a position, counting the number of characters, replacing strings, etc.

» SUBSTR

» INSTR

» REPLACE

Sometimes we need to extract portions of a string or reposition a string.

» LENGTH

» LPAD/RPAD

» LTRIM/RTRIM



SUBSTR Function

» Returns a portion of a string (substring)

» Syntax:

`SUBSTR (c, p, l)`

where *c* = character string (or column name)

p = beginning character position for the extraction

l = length of the string to return

» Examples:

SUBSTR('carey', 3, 2) retrieves ?

SUBSTR('carey', 2, 4) retrieves ?

SUBSTR('carey', -5, 2) retrieves ?

SUBSTR Function Example



Worksheet		Query Builder	
		<pre>SELECT zip, SUBSTR(zip, 1, 3), SUBSTR(zip, -3, 2) FROM customers WHERE SUBSTR(zip, -3, 2) < 30;</pre>	
		Query Result x	
		SQL All Rows Fetched: 7 in 0.052 seconds	
	ZIP	SUBSTR(ZIP,1,3)	SUBSTR(ZIP,-3,2)
1	98115	981	11
2	12211	122	21
3	82003	820	00
4	02110	021	11
5	33111	331	11
6	49006	490	00
7	31206	312	20

Arguments: zip – column name that contains the character string
1 – beginning position is 1 (the very first character)
3 – the length of character to be retrieved



INSTR Function

- » Returns the first position of a set of characters within a character string: the output is a numeric value

INSTR(string1, string2, [p, n])

No...

where string1: string value (or column) to search

string2: a set of characters (substring) to locate

p: start searching from pth position in string1

n: nth occurrence of string2 (ex. 2: second occurrence)

- » Examples:

INSTR('database', 'b') = ?

INSTR('database', 'a', 1, 2) = ?

The default value of p is 1

1st 'a' 2nd 'a'

INSTR Function



Worksheet Query Builder

```
SELECT name, INSTR(name, ',') "First comma",  
        INSTR(name, ',', 10) "Start read position 10",  
        INSTR(name, ',', 1, 2) "Second comma"  
FROM contacts;
```

Query Result x

SQL | All Rows Fetched: 3 in 0.039 seconds

	NAME	First comma	Start read position 10	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

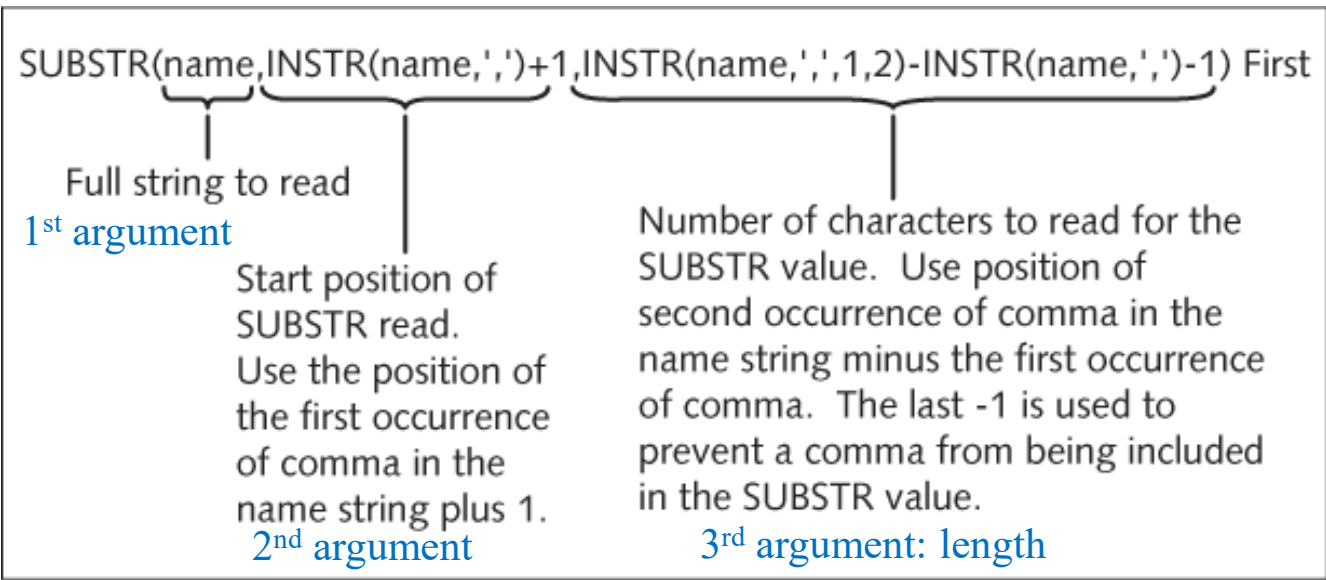
Position of comma
after 10th position

Position of the
second comma

What if you want to extract the first and last name from the NAME field?



Nesting Functions



We can extract the first name from the NAME column by using the Nesting functions.

e.g. if the *name* is 'Brown,Charlie,410-234-5678'

$\text{INSTR}(\text{name}, ',') + 1 = 6 + 1 = 7 \longrightarrow$ 2nd argument: starting position of the first name

$\text{INSTR}(\text{name}, ',', 1, 2) = 14$

Thus the last part = $14 - 6 - 1 = 7 \longrightarrow$ 3rd argument: length of the first name

This makes the above $\rightarrow \text{SUBSTR}(\text{name}, 7, 7)$ which extracts 'Charlie'

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
B	r	o	w	n	,	C	h	a	r	l	i	e	,	4	1	0	-	2	3	4



Nesting Functions Example

Worksheet

Query Builder

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

Query Result x

All Rows Fetched: 3 in 0.043 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

Nesting function simply means using one function as an argument of another function.



Practice

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

Display each person's telephone number only by using SUBSTR and INSTR.

LENGTH Function



» Determines the number of characters in a string

Worksheet Query Builder	
<pre>SELECT DISTINCT LENGTH(address) FROM customers ORDER BY LENGTH(address) DESC;</pre>	
Query Result x	
SQL All Rows Fetched: 7 in 0.077 seconds	
	LENGTH(ADDRESS)
1	20
2	18
3	17
4	16
5	13
6	12
7	11

Suppose JLDB needs to know the length of the customer address to create mailing labels.



146 Adela Avenue



LPAD and RPAD Functions

- » Pad, or fill in, a character string with a specific character (or blank space) to a fixed width

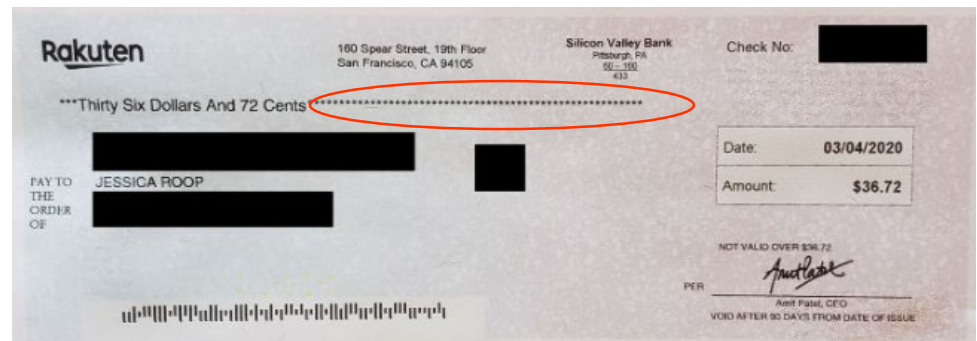
LPAD(c, l, s) or RPAD(c, l, s)

where c – character string (or column) to pad

l – the total length of the character string (c) after padding

s – the symbol or character to use as padding

Do you have an experience of receiving a check with the dollar amount followed by a series of asterisks?





LPAD and RPAD Example

- » LPAD: Filling in a space with blanks or asterisks to the left of the column (ex. aligns the column data to the right side).

Worksheet		Query Builder	
		<pre>SELECT firstname, LPAD(firstname, 12, ' ') "Space Pad", LPAD(firstname, 12, '*') "* Pad" FROM customers WHERE firstname LIKE 'J%';</pre>	
		Query Result x	
		SQL All Rows Fetched: 4 in 0.025 seconds	
	FIRSTNAME	Space Pad	* Pad
1	JORGE	JORGE	*****JORGE
2	JAKE	JAKE	*****JAKE
3	JASMINE	JASMINE	*****JASMINE
4	JENNIFER	JENNIFER	****JENNIFER

- » RPAD: fills in a space with a specified character to the right of the column.

Worksheet		Query Builder	
		<pre>SELECT firstname, RPAD(firstname, 12, ' ') "Space Pad", RPAD(firstname, 12, '*') "* Pad" FROM customers WHERE firstname LIKE 'J%';</pre>	
		Query Result x	
		SQL All Rows Fetched: 4 in 0.031 seconds	
	FIRSTNAME	Space Pad	* Pad
1	JORGE	JORGE	JORGE*****
2	JAKE	JAKE	JAKE*****
3	JASMINE	JASMINE	JASMINE*****
4	JENNIFER	JENNIFER	JENNIFER****



LTRIM and RTRIM Functions

» Remove a specific string of characters

LTRIM(c, s)

where c – character string (or column) to modify
s – the string to remove from the left end of c

prefix match.

Removes all characters that appear in s until reaching a character not in s and then returns the result.

Worksheet Query Builder	
<pre>SELECT address, LTRIM(address, 'P.O. BOX') FROM customers WHERE state = 'FL';</pre>	
Script Output x Query Result x	
SQL All Rows Fetched: 4 in 0.022 seconds	
ADDRESS	LTRIM(ADDRESS,'P.O.BOX')
1 P.O. BOX 651	651
2 P.O. BOX 66	66
3 357 WHITE EAGLE AVE.	357 WHITE EAGLE AVE.
4 P.O. BOX 677	677

Worksheet Query Builder	
<pre>SELECT name, RTRIM(name, 'INC.') FROM publisher;</pre>	
Script Output x Query Result x	
SQL All Rows Fetched: 5 in 0.019 seconds	
NAME	RTRIM(NAME,'INC.')
1 PRINTING IS US	PRINTING IS US
2 PUBLISH OUR WAY	PUBLISH OUR WAY
3 AMERICAN PUBLISHING	AMERICAN PUBLISHING
4 READING MATERIALS INC.	READING MATERIALS
5 REED-N-RITE	REED-N-RITE



REPLACE Function

» Substitutes a string with another specified string

REPLACE(c, s, r)

where c – character string (or column) to search → exact match
s – a string of characters to find (to be replaced)
r – the string of characters to substitute for s

Worksheet		Query Builder
		<pre>SELECT address, REPLACE(address, 'P.O.', 'POST OFFICE') FROM customers WHERE state = 'FL';</pre>
		Query Result x
		SQL All Rows Fetched: 4 in 0.02 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



Number Functions

» Allow for manipulation of numeric data

- **ROUND**
- **TRUNC**
- **MOD**
- **ABS**



ROUND Function

» Rounds a numeric column/value to a stated precision

ROUND(*n*, *p*)

- *n*: the numeric data (column) to round
- *p*: the position of the digits to which data should be rounded. Positive number refers to the right side of the decimal, and a negative value refers to the left side of the decimal position.

» Example: when *price* = 132.65

- ROUND(*price*, 1) = 132.7 (1 means round to one-tenth digit)
- ROUND(*price*, 0) = 133 (zero means round to ones digit)
- ROUND(*price*, -1) = 130 (-1 means round to tens digit)

ROUND Function Example



Worksheet Query Builder

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

Query Result x

SQL | All Rows Fetched: 14 in 0.054 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

TRUNC Function



» Truncates a numeric column/value to a specific position

Worksheet

Query Builder

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

Truncate after the tenths digit

Query Result x

SQL | All Rows Fetched: 14 in 0.023 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



FLOOR & CEIL Functions

- » FLOOR returns the largest integer less than or equal to a given numeric value/column
- » CEIL returns the smallest integer greater than or equal to a given numeric value/column

Worksheet		Query Builder	
		<pre>SELECT retail, FLOOR(retail), CEIL(retail) FROM books;</pre>	
		Script Output x Query Result x	
		SQL All Rows Fetched: 14 in 0.017 seconds	
	RETAIL	FLOOR(RETAIL)	CEIL(RETAIL)
1	30.95	30	31
2	22	22	22
3	59.95	59	60
4	55.95	55	56
5	19.95	19	20
6	75.95	75	76
7	25	25	25
8	54.5	54	55
9	89.95	89	90
10	28.75	28	29
11	8.95	8	9
12	29.95	29	30
13	31.95	31	32
14	39.95	39	40



MOD Function

» Returns only the remainder of a division operation

235 ounces of liquid is how many lbs and oz? (16 ounces = 1 lbs)

The screenshot shows a database query builder interface. At the top, there are tabs for 'Worksheet' and 'Query Builder'. The 'Query Builder' tab is active, displaying a SQL query: `SELECT 235/16, TRUNC(235/16, 0) LBS, MOD(235, 16) OZ FROM dual;`. Below the query, there is a 'Query Result' section. It includes a status bar that says 'All Rows Fetched: 1 in 0.049 seconds'. Below the status bar is a table with the following data:

	235/16	LBS	OZ
1	14.6875	14	11

→ 235 ounces of liquid is 14 lbs and 11 oz.

ABS Function







» Returns the absolute value of the numeric values

Worksheet Query Builder

```
SELECT title, cost - retail, ABS(cost - retail)
FROM books
WHERE title LIKE 'H%';
```

Query Result x

    SQL | All Rows Fetched: 4 in 0.033 seconds

	TITLE	COST-RETAIL	ABS(COST-RETAIL)
1	HOLY GRAIL OF ORACLE	-28.7	28.7
2	HANDCRANKED COMPUTERS	-3.2	3.2
3	HOW TO GET FASTER PIZZA	-12.1	12.1
4	HOW TO MANAGE THE MANAGER	-16.55	16.55



Date Functions





- » Used to perform date calculations or format date values
- » Subtract date for a number of days difference

Worksheet

Query Builder

```
SELECT order#, shipdate, orderdate, shipdate - orderdate "Shipping Days"
FROM orders
WHERE shipdate IS NOT NULL;
```

Query Result x

    SQL | All Rows Fetched: 15 in 0.061 seconds

	ORDER#	SHIPDATE	ORDERDATE	Shipping Days
1	1000	02-APR-09	31-MAR-09	2
2	1001	01-APR-09	31-MAR-09	1
3	1002	01-APR-09	31-MAR-09	1
4	1003	01-APR-09	01-APR-09	0
5	1004	05-APR-09	01-APR-09	4
6	1005	02-APR-09	01-APR-09	1
7	1006	02-APR-09	01-APR-09	1
8	1007	04-APR-09	02-APR-09	2
9	1008	03-APR-09	02-APR-09	1
10	1009	05-APR-09	03-APR-09	2
11	1010	04-APR-09	03-APR-09	1



MONTHS_BETWEEN Function

» Determines the number of months between two dates

Note

MONTHS_BETWEEN(*d1*, *d2*)

- subtract d2 from d1 and obtain the months between the two dates
- d1 is the later date

JustLee Books wants to know if customers order books that are recently released books or order books published many months ago.

Returns integer value if d1 and d2 are the same days of the month or both are the last days of the month.



MONTHS_BETWEEN Function Example

Worksheet Query Builder	
<pre>SELECT b.title, MONTHS_BETWEEN(o.orderdate, b.pubdate) "Months since Published" FROM books b JOIN orderitems oi ON b.isbn = oi.isbn JOIN orders o ON o.order# = oi.order# WHERE o.order# = 1012;</pre>	
Query Result x	
SQL All Rows Fetched: 4 in 0.023 seconds	
TITLE	Months since Published
1 REVENGE OF MICKEY	39.64516129032258064516129032258064516129
2 HANDCRANKED COMPUTERS	50.41935483870967741935483870967741935484
3 PAINLESS CHILD-REARING	56.5483870967741935483870967741935483871
4 BIG BEAR AND LITTLE DOVE	40.83870967741935483870967741935483870968

The non-integer part represents a portion of a month (based on 31 days).
Use TRUNC (xxx, 0) to eliminate those → Next page



Nested within TRUNC Function

Worksheet Query Builder

```
SELECT b.title, TRUNC(MONTHS_BETWEEN(o.orderdate, b.pubdate), 0) "Months since Published"
FROM books b JOIN orderitems oi ON b.isbn = oi.isbn
      JOIN orders o ON o.order# = oi.order#
WHERE o.order# = 1012;
```

Query Result x

SQL | All Rows Fetched: 4 in 0.07 seconds

	TITLE	Months since Published
1	REVENGE OF MICKEY	39
2	HANDCRANKED COMPUTERS	50
3	PAINLESS CHILD-REARING	56
4	BIG BEAR AND LITTLE DOVE	40

ADD_MONTHS Function



» Adds a specified number of months to a date

Notes

ADD_MONTHS (*d1*, *m*)

- add *m* months to date, *d1*

Suppose the management of JustLee Books renegotiates contract pricing for books every 18 months and stocks books for up to 17 years after publication.

Assume that the books in COMPUTER category were negotiated on Dec. 1, 2022

The screenshot shows a SQL Query Builder window with a query and its results. The query is:

```
SELECT title, pubdate,
       ADD_MONTHS('01-DEC-22', 18) "Renegotiation Date",
       ADD_MONTHS(pubdate, 12*17) "Drop Date"
FROM books
WHERE category = 'COMPUTER'
ORDER BY "Renegotiation Date";
```

The results are displayed in a table with 4 columns: TITLE, PUBDATE, Renegotiation Date, and Drop Date. The results are sorted by Renegotiation Date.

	TITLE	PUBDATE	Renegotiation Date	Drop Date
1	DATABASE IMPLEMENTATION	04-JUN-13	01-JUN-24	04-JUN-30
2	HOLY GRAIL OF ORACLE	31-DEC-15	01-JUN-24	31-DEC-32
3	HANDCRANKED COMPUTERS	21-JAN-15	01-JUN-24	21-JAN-32
4	E-BUSINESS THE EASY WAY	01-MAR-16	01-JUN-24	01-MAR-33



NEXT_DAY Function

- » Determines the next occurrence of a specified day of the week after a given date

Notes

NEXT_DAY(*d*, *DAY*)

- *d* is a given starting date, and *DAY* is a day of the week to identify

Suppose JustLee Books has a policy that books must be shipped by the first Monday after receiving a customer's order.

The screenshot shows a database query tool interface. The top bar has 'Worksheet' and 'Query Builder' tabs. The main area contains a SQL query: `SELECT order#, orderdate, NEXT_DAY(orderdate, 'MONDAY') "Shipped by"`
`FROM orders`
`WHERE order# = 1018;`

Below the query, there are tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab is active, showing a table with 1 row and 3 columns: 'ORDER #', 'ORDERDATE', and 'Shipped by'. The data in the row is: 1, 1018 05-APR-19, and 08-APR-19. Above the table, it says 'All Rows Fetched: 1 in 0.019 seconds'.

	ORDER #	ORDERDATE	Shipped by
1	1018	05-APR-19	08-APR-19



ROUND Function for Date

» Round(*d*, *u*) where *d* is a date to be rounded, and *u* is either month or year

- Cutoff point for ROUND(*d*, 'MONTH') is 16th of each month.
- Cutoff for ROUND(*d*, 'YEAR') is July 16th.

The screenshot shows a database query tool interface. At the top, there are tabs for 'Worksheet' and 'Query Builder'. Below them, a SQL query is entered in a text area:

```
SELECT pubdate, ROUND(pubdate, 'MONTH'), ROUND(pubdate, 'YEAR')
FROM books
WHERE category = 'CHILDREN';
```

Two green arrows point from the text labels 'MM' and 'YYYY' to the 'MONTH' and 'YEAR' arguments in the query, respectively.

Below the query editor, there is a section for 'Script Output' and 'Query Result'. The 'Query Result' tab is active, showing the results of the query. It includes a status bar that says 'All Rows Fetched: 2 in 0.024 seconds'.

	PUBDATE	ROUND(PUBDATE,'MONTH')	ROUND(PUBDATE,'YEAR')
1	18-MAR-16	01-APR-16	01-JAN-16
2	08-NOV-15	01-NOV-15	01-JAN-16



TO_DATE Function

- » Converts various date formats to the internal/default format (DD-MON-YY) used by Oracle

TO_DATE('d', 'f')

d – the date entered by users (entry date)

f – the format of the entry date, d

Worksheet

Query Builder

```
SELECT order#, customer#, orderdate, shipdate
FROM orders
WHERE orderdate = TO_DATE('March 31, 2019', 'Month DD, YYYY');
```

Script Output x

Query Result x

SQL

All Rows Fetched: 3 in 0.018 seconds

	ORDER#	CUSTOMER#	ORDERDATE	SHIPDATE
1	1000	1005	31-MAR-19	02-APR-19
2	1001	1010	31-MAR-19	01-APR-19
3	1002	1011	31-MAR-19	01-APR-19

TO_CHAR Function



» Converts dates and numbers to a formatted character string

TO_CHAR(*n*, *f*)

n – the date or number to format

f – the formatting instruction to use

```
Worksheet  Query Builder
SELECT title,
       TO_CHAR(pubdate, 'MON DD, YY') "Publication Date1",
       TO_CHAR(pubdate, 'mm/dd/yyyy') "Publication Date2",
       TO_CHAR(retail, '$999.99') "Retail Price"
FROM books
WHERE category = 'COMPUTER';
```

Use 'Mon' to display in mixed cases

Script Output x Query Result x				
All Rows Fetched: 4 in 0.022 seconds				
TITLE	Publication Date1	Publication Date2	Retail Price	
1 DATABASE IMPLEMENTATION	JUN 04, 13	06/04/2013	\$55.95	
2 HOLY GRAIL OF ORACLE	DEC 31, 15	12/31/2015	\$75.95	
3 HANDCRANKED COMPUTERS	JAN 21, 15	01/21/2015	\$25.00	
4 E-BUSINESS THE EASY WAY	MAR 01, 16	03/01/2016	\$54.50	



Format Arguments for Dates

Element	Description	Example
MONTH	Name of the month spelled out and padded with blank spaces to a total width of nine characters	APRIL
MON	Three-letter abbreviation for the name of the month	APR
MM	Two-digit numeric value for the month	04
RM	Roman numeral representing the 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	Dec. 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	2021
YYY or YY or Y	The last three, two, or single digits of the year	2021: 021, 21, 1
YEAR	Spelled out version of the year	TWO THOUSAND TWENTY ONE



Format Arguments for Time and Number

Time Element		
SS	Seconds	Values 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.
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



TO_NUMBER Function

» Converts a value to a numeric datatype, if possible

Worksheet Query Builder

```
SELECT title, pubdate,  
       TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) - TO_NUMBER(TO_CHAR(pubdate, 'yyyy')) "Years"  
FROM books  
WHERE category = 'COMPUTER';
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0.019 seconds

	TITLE	PUBDATE	Years
1	DATABASE IMPLEMENTATION	04-JUN-13	11
2	HOLY GRAIL OF ORACLE	31-DEC-15	9
3	HANDCRANKED COMPUTERS	21-JAN-15	9
4	E-BUSINESS THE EASY WAY	01-MAR-16	8

How old is each book in the computer category in years?

This function will return an error if the string being converted contains non-numeric characters.



Other Functions

- » NVL
- » NVL2
- » CASE expression (CASE... WHEN)



NVL Function

» **NVL (x, y)**: substitutes y for a NULL value in x

```
Worksheet | Query Builder
SELECT title, retail, discount, retail-discount,
       retail-NVL(discount, 0) "Sales Price"
FROM books;
```

→ If discount value is NULL, substitute 0 for the NULL

Script Output x

Query Result x

SQL | All Rows Fetched: 14 in 0.078 seconds

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

NVL can be used with numbers or dates



NVL2 Function

- » Allows different actions based on whether a value is NULL or not
- » **NVL2(x, y, z)**: if the value of x isn't null, the output is y; otherwise, the output is z

Worksheet | Query Builder

```
SELECT order#, orderdate, NVL2(shipdate, 'Shipped', 'Not Shipped') "Status"
FROM orders
WHERE shipstate = 'FL';
```

Output for Not Null Output for Null

Script Output x | Query Result x

SQL | All Rows Fetched: 5 in 0.026 seconds

	ORDER#	ORDERDATE	Status
1	1003	01-APR-19	Shipped
2	1006	01-APR-19	Shipped
3	1016	04-APR-19	Not Shipped
4	1017	04-APR-19	Shipped
5	1018	05-APR-19	Not Shipped



DECODE Function

» Determines action based upon values in a list

DECODE (V, L1, R1, L2, R2, ..., D)

V – column or values to be checked

Ln – list of specific values that leads to different results

Rn – result for Ln

D – default value

The screenshot shows a database query builder interface. The top tab is 'Worksheet' and the bottom tab is 'Query Builder'. The SQL query is as follows:

```
SELECT customer#, state,  
       DECODE (state, 'CA', .0725,  
                'FL', .06,  
                'GA', 0.04,  
                0) "Sales Tax Rate"  
FROM customers  
WHERE state IN ('CA', 'FL', 'GA');
```

Below the query, there is a 'Script Output' tab and a 'Query Result' tab. The 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 9 in 0.023 seconds'.

	CUSTOMER#	STATE	Sales Tax Rate
1	1001	FL	0.06
2	1002	CA	0.0725
3	1003	FL	0.06



CASE Expression

Suppose we need to determine the retirement level of each employee based on the number of years employed at JustLee Books.

Notes

```
SELECT empno, lname, fname,
       ROUND(MONTHS_BETWEEN(sysdate, hiredate)/12, 2) "Years",
       CASE WHEN MONTHS_BETWEEN(sysdate, hiredate)/12 < 5 THEN 'Level 1'
            WHEN MONTHS_BETWEEN(sysdate, hiredate)/12 < 9 THEN 'Level 2'
            WHEN MONTHS_BETWEEN(sysdate, hiredate)/12 < 13 THEN 'Level 3'
            WHEN MONTHS_BETWEEN(sysdate, hiredate)/12 < 17 THEN 'Level 4'
            ELSE 'Level 5'
       END "Retirement Level"
FROM employees;
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.02 seconds

	EMPNO	LNAME	FNAME	Years	Retirement Level
1	7839	KING	BEN	17.29	Level 5
2	8888	JONES	LARRY	14.95	Level 4
3	7344	SMITH	SAM	3.82	Level 1
4	7355	POTTS	JIM	8.36	Level 2
5	8844	STUART	SUE	11.82	Level 3



Summary (1/3)

- » Single-row functions return a result for each row or record processed
- » Case conversion functions such as UPPER, LOWER, and INITCAP can be used to alter the case of character strings
- » Character manipulation functions can be used to extract substrings (portions of a string), identify the position of a substring in a string, replace occurrences of a string with another string, determine the length of a character string, and trim spaces or characters from strings
- » Nesting one function within another allows multiple operations to be performed on data

Summary (2/3)



- » Simple number functions such as ROUND and TRUNC can round or truncate a number on both the left and right side of a decimal
- » The MOD function is used to return the remainder of a division operation
- » Date functions can be used to perform calculations with dates or to change the format of dates entered by a user
- » The NVL and NVL2 functions are used to address problems encountered with NULL values

Summary (3/3)



- » The TO_CHAR function lets a user present numeric data and dates in a specific format
- » The CASE expression enables you to evaluate conditions to determine the resulting value
- » The DUAL table can be helpful when testing functions