

## Query Expressions Function Definitions and Examples

Functions in Query Expressions are special types of command words in the SQL command set, usually one-word commands which manipulate data items and return a single value which can be determined by various input parameters.

- A typical function appears like this on the Edit Expression Properties page:

- The functions displayed below are non-Wesleyan functions.
  - To read about Wesleyan Functions, go to the [Wesleyan Functions](#) page in the [SFIS Blog](#).
  - To view presentations on query expressions and functions, in the SFIS Blog go to the [QUERY ~ EXTERNAL REFERENCES](#) page and look under **Query Expressions and Functions**
- The functions shown below generally fall into these overall categories:
  - **Single-Row**
  - **Grouping/Aggregate**
  - **Analytic**
- See [Appendix A](#) for examples of expressions that use combinations of different functions as well as calculations, and, in some cases, text.
- See a list of queries with functions in [Appendix B](#) . *The queries are referenced in the tables below by number.*
- [Appendix C](#) contains additional Functions from a HEUG Presentation, **Expressions in Query: An In-Depth Exploration Into Function Statements**

## Query Expressions Function Definitions and Examples

### SINGLE-ROW FUNCTIONS

**This first set of functions are the single-row Functions**

- They operate on a single value and then return a single value.
- They can be used wherever an expression is valid.
- They can be divided into different logical categories.

The different types of Single-row functions are:

- Conversion
- String/Character
- Numeric
- Date and Time
- Advanced

### CONVERSION FUNCTIONS

**Conversion: Change or convert values from one data type to another (character to numeric, numeric to character, character to date or date to character)**

Function	Usage	Expression Example	Result	Field Header	Query
TO_CHAR	converts a number or date to a string - to_date(string1,[format_mask])	TO_CHAR((SYSDATE),'mm-dd-yyyy')	01-11-2016	TO_CHAR SYSDATE	1
TO_CHAR	(same as previous)	TO_CHAR((A.ACTION_DT),'mm-dd-yyyy')	June 05, 2012	TO_CHAR - Date- Month dd, yyyy	1

### STRING FUNCTIONS

**String (also referred to as Character): perform operations on a string (char/varchar) input value and return a string or numeric value**

Function	Usage	Example	Result	Field Header	Query
CONCATENATION	Appends/combines two or more literal expressions, column values or variables together into one string - (string1    string2    string_n)	'Major/Minor: '    A.WES_MAJORS     '/'     A.WES_MINORS	Major/Minor: MUSC,PHYS/FILM- MN	Major/Minor CONCATENATE	1, 2
INITCAP	converts a string to	INITCAP(A.WES_MAJORS)	Musc,Phys	Wes Majors	1

## Wesleyan University Student Faculty Information System

PeopleSoft 9.0  
PeopleTools 8.52

PS Query 8.52  
Expressions – SQL Functions

### Query Expressions Function Definitions and Examples

	initial capital letters - initcap(string1)			<i>INITCAP</i>	
LOWER	converts a string to all lowercase characters – lower(string1)	LOWER(A.WES_MAJORS)	<i>musc,phys</i>	<i>Wes Majors LOWER</i>	<i>1</i>
UPPER	converts a string to all uppercase characters – upper(string1)	UPPER(A.CITY)	<i>NEW YORK</i>	<i>City UPPER</i>	<i>1</i>
SUBSTR	extracts a portion of a string or field - SUBSTR(char, position [, substring_length ])	SUBSTR(A.WESPO,1,3) – <i>starting field is 93000</i>	<i>930</i>	<i>Wespo SUBSTR First 3 Chars</i>	<i>1</i>
LENGTH	returns the number of characters in a string or field – length(char)	LENGTH(A.WES_MAJORS)	<i>9</i>	<i>Major Field LENGTH</i>	<i>1</i>

### NUMERIC FUNCTIONS

**Numeric:** performs operations on numeric values and returns numeric values, accurate to 38 decimal points

Function	Usage	Example	Result	Field Header	Query
ROUND	returns a number rounded to x number of decimal points - round(number,[decimal places]) [default is 0]	ROUND(A.CUM_GPA,2) – <i>starting field is 93.635</i>	<i>93.64</i>	<i>GPA ROUND to 2 decimals</i>	<i>1</i>
TRUNC	returns a number truncated to x number of decimal points - trunc(number,[decimal places])	TRUNC(A.CUM_GPA,0) – <i>starting field is 93.635</i>	<i>93</i>	<i>GPA TRUNCATE to 0 decimals</i>	<i>1</i>
CEIL	returns the smallest integer value that is greater than or equal to a number - ceil(number)	CEIL(A.CUM_GPA) – <i>starting field is 93.635</i>	<i>94</i>	<i>GPA - CEIL (Ceiling)</i>	<i>1</i>
FLOOR	returns the largest integer value that is equal to or less than a number - floor(number)	FLOOR(A.CUM_GPA) – <i>starting field is 93.635</i>	<i>93</i>	<i>GPA - FLOOR</i>	<i>1</i>

**Query Expressions**  
**Function Definitions and Examples**

**DATE AND TIME FUNCTIONS**

**Date and Time: Perform operations on a date and time input values and return string, numeric, or date and time values**

Function	Usage	Example	Result	Field Header	Query
SYSDATE	returns the current system date and time on your local database - sysdate	SYSDATE - where today's date is 01/11/2016	2016-01-11	SYSDATE as Character Expr Type	1
ADD_MONTHS	returns a date plus n months - add_months(date1,n)	ADD_MONTHS(SYSDATE,3) - where today's date is 01/11/2016	2016-04-11	ADD_MONTHS to SYSDATE (+3)	1
MONTHS_BETWEEN	returns number of months between two dates - MONTHS_BETWEEN(date1, date2)	MONTHS_BETWEEN(SYSDATE,A.ACTION_DT) - where today's date is 01/11/2016 and ACTION_DT is 06/05/2012	43	MONTHS_BETWEEN 2 Dates	1

**ADVANCED FUNCTIONS**

Function	Usage	Example	Result	Field Header	Query
GREATEST	returns the greatest from a list of one or more expressions - GREATEST(expr [, expr ]...)	GREATEST(A.UNT_TAKEN_PGRSS ,A.UNT_INPROG_GPA,A.TOT_INPROG_NOGPA) – where UNT_TAKEN_PGRSS = 5.500 and UNT_INPROG_GPA = 4.500 and TOT_INPROG_NOGPA = 1.000	5.500	GREATEST example	2
LEAST	returns the least (the smallest) from a list of expressions - LEAST(expr [, expr ]...). This function is used for multiple values in the same row. See the MIN function if you want the smallest value from a group of rows.	LEAST(A.UNT_TAKEN_PGRSS,A.UNT_INPROG_GPA,A.TOT_INPROG_NOGPA) – where UNT_TAKEN_PGRSS = 5.500 and UNT_INPROG_GPA = 4.500 and TOT_INPROG_NOGPA = 1.000	1.000	LEAST example	2
NVL	allows substitution of a value when a null value is encountered - NVL(string1,	Example 1: NVL(A.WES_MINORS,'No Minor') – where WES_MINORS is null	No Minor	NVL Minors Null	2

## Wesleyan University Student Faculty Information System

PeopleSoft 9.0  
PeopleTools 8.52

PS Query 8.52  
Expressions – SQL Functions

### Query Expressions Function Definitions and Examples

	replace_with)				
NVL	(same as previous)	Example 2: NVL(A.WES_MINORS,'No Minor') – where WES_MINORS = FRST-MN	FRST-MN	NVL Minors Null	2
NVL2	allows the substitution of a value when a null value is encountered, as well as when a non-null value is encountered - NVL2(string1, value_if_not_null, value_if_null)	Example 1: NVL2(A.WES_MINORS,'Minor Yes', 'Minor No') – where WES_MINORS is null	Minor No	NVL2 Minor Null or Not Null	2
NVL2	(same as previous)	Example 2: NVL2(A.WES_MINORS,'Minor Yes', 'Minor No') – where WES_MINORS = FRST-MN	Minor Yes	NVL2 Minor Null or Not Null	2
ROWNUM	assigns a number indicating the order in which each row is returned by a query	ROWNUM	2931	ROWNUM	2
CASE	performs the functionality of an “IF-THEN-ELSE” statement ( <i>with greater possibilities than DECODE or COALESCE</i> ) – CASE WHEN condition_1 THEN result_1 WHEN condition_2 THEN result_2 WHEN condition_n THEN result_n ELSE result END	CASE WHEN A.ACAD_CAREER = 'UGRD' THEN 'Undergraduate Student' WHEN A.ACAD_CAREER = 'GRAD' THEN 'Graduate Student' WHEN A.ACAD_CAREER = 'GLSP' THEN 'GLSP Student' ELSE 'Check Career' END – where ACAD_CAREER is GRAD	Undergraduate Student	CASE WHEN A.ACAD_CAREER = UGRD	2

**Query Expressions**  
**Function Definitions and Examples**

**GROUPING/AGGREGATE FUNCTIONS**

Grouping/Aggregate Functions operate against a collection of values, but return a single, summarizing value. (SUM and AVG work with numeric values and ignore NULL values)

Function	Usage	Example	Result	Field Header	Query
AVG	computes the average of values in a column or an expression – avg(number)	AVG(A.TOT_CUMULATIVE)	16.7792	AVG(A.TOT_CUMULATIVE)	3
COUNT	counts all rows or all rows defined in an expression – count([expr]) or count(*)	COUNT(*)	3241	COUNT(*)	3
MIN	returns the smallest value from a set of rows	MIN(A.ADMIT_TERM)	1089	MIN(A.ADMIT_TERM)	3
MAX	identifies the maximum value in a column by the expression	MAX(A.ADMIT_TERM)	1160	MAX(A.ADMIT_TERM)	3
SUM	computes the sum	SUM(A.CUR_RESIDENT_TERMS)	6	SUM(A.CUR_RESIDENT_TERMS)	3
SUM	(same as previous)	SUM(A.CUM_RESIDENT_TERMS)	12209	SUM(A.CUM_RESIDENT_TERMS)	3

**Query Expressions**  
**Function Definitions and Examples**

**ANALYTIC FUNCTIONS**

**Analytic Functions:** Analytic functions compute an aggregate value based on a group of rows. There are many variations of Analytic Functions. Below are two versions of a function using “Over...Partition. For more information on Analytic Functions, see [Appendix C](#) which lists Analytic Functions discussed in the presentation [Expressions in Query: An In-Depth Exploration into Function Statements](#)

**Overview of COUNT (.....) OVER (PARTITION BY .....**

1. Example: COUNT(A.EMPLID) OVER (PARTITION BY A.WES\_CLASS)
2. Note: Function operation and grouping happens after all query criteria have been met
3. Calculation is independent of output
4. The value repeats for each row with that group/partition
5. To change Group by adding a new expression, add another Expression after the Count – the grouping changes

Function	Usage	Example	Result	Field Header	Query
COUNT (.....) OVER (PARTITION BY .....)	returns the number of rows in the set – the value repeats for that group/partition	COUNT(A.EXT_ORG_ID)OVER (PARTITION BY A.COUNTRY)	82	COUNT OVER PARTITION Country	4
COUNT (.....) OVER (PARTITION BY .....)	(same as previous)	COUNT(A.EXT_ORG_ID)OVER (PARTITION BY A.EXT_ORG_TYPE)	4772	COUNT OVER PARTITION Org Type	4

## Wesleyan University Student Faculty Information System

PeopleSoft 9.0  
PeopleTools 8.52

PS Query 8.52  
Expressions – SQL Functions

### Query Expressions Function Definitions and Examples

#### APPENDIX A

#### EXAMPLES - COMBINATIONS of FUNCTIONS and CALCULATIONS

Purpose	Example	Sample Field	Result	Field Header	Query
Age in Years - MONTHS BETWEEN Birthdate and Today	(MONTHS_BETWEEN(TO_DATE(TO_CHAR(SYSDATE, 'YYYY-MM-DD'), 'YYYY-MM-DD'), TO_DATE(A.BIRTHDATE, 'YYYY-MM-DD'))/12)	BIRTHDATE = 06/30/1994	20	Age in Years - MONTHS BETWEEN	1
Format SYSDATE	TO_CHAR((SYSDATE), 'mm-dd-yyyy')		01-11-2016	TO_CHAR SYSDATE	1
Add 14 to a Date and change format	TO_CHAR((ACTION_DT+14), 'mm/dd/yyyy')	ACTION_DT = 06/05/2012	06/19/2012	TO_CHAR Date +14 mm/dd/yyyy	1
Subtract 14 from a Date and change format to solid caps with embedded spaces	TO_CHAR((ACTION_DT-14), 'MONTH DD, YYYY')	ACTION_DT = 06/05/2012	MAY 22, 2012	TO_CHAR Date -14 CAPS Spaces	1
Subtract 14 from a Date and format with initial cap and removing spaces	TO_CHAR((ACTION_DT-14), 'fmMonth DD, YYYY')	ACTION_DT = 06/05/2012	May 22, 2012	TO_CHAR - Date +14 Mixed	1
Military time	TO_CHAR((ACTION_DT), 'hh24:mi')	ACTION_DT = 12:00 a.m.	00:00	TIME Military	1
Time as AM or PM	TO_CHAR((ACTION_DT), 'hh:miAM')	ACTION_DT = 12:00 a.m.	12:00AM	TIME AM/PM	1
Display Date as Year (Method 1)	TO_CHAR((ACTION_DT), 'yyyy')	ACTION_DT = 06/05/2012	2012	DATE Display as Year Method 1	1
Display Date as Year (Method 2)	SUBSTR(A.ACTION_DT,1,4)	ACTION_DT = 06/05/2012	2012	DATE Display as Year Method 2	1
Display text "Cumulative GPA: " and append field	'Cumulative GPA: '    TO_CHAR(A.CUM_GPA, '99.999')	CUM_GPA = 93.635	Cumulative GPA: 93.635	TO_CHAR Cum GPA plus Text	1
Display and format City, State Zip as one field (if all are USA)	A.CITY    ', '    A.STATE    ' '    A.POSTAL	Middletown CT 06457	Middletown, CT 06457	City, State Zip CONCATENATE	2
Display and format City, State and Postal as one field for those that are USA. Display and format City, Postal and Country for those that are not USA	CASE WHEN A.COUNTRY = 'USA' THEN A.CITY    ', '    A.STATE    ' '    A.POSTAL ELSE A.CITY    ' '    A.POSTAL    ' '    A.COUNTRY END CASE	Arlington VA 22207	Arlington, VA 22207	City, State Zip CASE & CONCAT	2
Same as previous	Same as previous	Bangkok	Bangkok	City, State Zip	2

## Wesleyan University Student Faculty Information System

PeopleSoft 9.0  
PeopleTools 8.52

PS Query 8.52  
Expressions – SQL Functions

### Query Expressions Function Definitions and Examples

		10110 THA	10110 THA	<i>CASE &amp; CONCAT</i>	
Display Postal Code as Zip Plus 4 if appropriate	CASE WHEN (A.COUNTRY = 'USA' AND LENGTH(TRIM(A.POSTAL)) = 9) THEN SUBSTR(A.POSTAL,1,5)    '-'	<i>064591234</i>	<i>06459-1234</i>	<i>CASE plus - Zip Code Plus 4</i>	<i>2</i>
Display Postal Code as Zip Plus 4 if appropriate	CASE WHEN (A.COUNTRY = 'USA' AND LENGTH(TRIM(A.POSTAL)) = 9) THEN SUBSTR(A.POSTAL,1,5)    '-'	<i>90610</i>	<i>90610</i>	<i>CASE plus - Zip Code Plus 4</i>	<i>2</i>
Display text "Major/Minor:" and append two fields with "/" between if WES_MINORS is not null. Otherwise only display WES.MAJORS	'Major/Minor: '     A.WES_MAJORS     CASE WHEN A.WES_MINORS > '' THEN '/'     A.WES_MINORS ELSE '' END case	<i>ENVS,FRST</i>	<i>Major/Minor: ENVS,FRST</i>	<i>Major/Minor CONCATENATE</i>	<i>1, 2</i>
Same as previous	Same as previous	<i>GOVT CEAS-MN</i>	<i>Major/Minor: GOVT/CEAS- MN</i>	<i>Major/Minor CONCATENATE</i>	<i>1, 2</i>

## APPENDIX B

### PEOPLESOFT QUERIES WITH EXAMPLES OF FUNCTIONS

To see examples of the above functions, you can take a look at these queries. These are public queries, and as such should not be changed by anyone except the person who created them. If you would like a copy of any query, please save the copy as a private query.

The designation for each query is shown next to the functions.

1. WES\_SFIS\_TRAINING\_EXPRESSIONS
2. WES\_SFIS\_TRAINING\_EXPRESS\_ADV2
3. WES\_SFIS\_TRAINING\_EXPRESS\_GRP
4. WES\_SFIS\_TRAINING\_EXPR\_ORGS

## Query Expressions Function Definitions and Examples

### APPENDIX C

#### Additional Functions from HEUG Presentation Expressions in Query

The HEUG Presentation [Expressions in Query: An In-Depth Exploration Into Function Statements](#) is densely packed with information on Functions. In addition to the Functions discussed above, it includes the following material as well as complex function statements, hints, tips and other advice on using Query. Click on the above link to see the presentation.

- *MOD* - returns the remainder of *m* divided by *n* (and returns *m* if *n* is 0) – *mod(m,n)*
- *REMAINDER* - returns the remainder of *m* divided by *n* – *remainder(m,n)*
- *INSTR* - returns the location of a substring in a string – *instr(string1,string2,[start\_position],[nth\_appearance])*
- *REPLACE* - replaces a sequence of characters in a string with another set of characters – *replace(string1,string\_to\_replace,[replacement\_string])*
- *SOUNDEX* - returns a string containing the phonetic representation (the way it sounds) of the string – *soundex(string1)*
- *TRANSLATE* - converts a string from one character set to another – *translate(string1,string\_to\_replace,[replacement\_string])*
- *TRIM* - removes leading characters, trailing characters
- or both from a character string – *trim([leading|trailing|both[trim\_character]]string1)*
- *NEXT\_DAY* - returns the date of the first weekday named that is later than the date specified - *NEXT\_DAY(date, char)*
- *Nested* - *CASE WHEN ..... THEN ..... ELSE ..... END*
- *RANK* - calculates the rank of a value in a group of values – *RANK( ) OVER ([ query\_partition\_clause ] order\_by\_clause)*
- *DENSE\_RANK* - computes the rank of a row in an ordered group of rows – *DENSE\_RANK( ) OVER([ query\_partition\_clause ] order\_by\_clause)*
- *PERCENT\_RANK* - calculates the rank of *r* minus 1, divided by 1 less than the number of rows being evaluated (the entire query result set or a partition) – *PERCENT\_RANK( ) OVER ([ query\_partition\_clause ] order\_by\_clause)*
- *LAG | LEAD* - provide access to more than one row of a table at the same time without a self join. Given a series of rows returned from a query and a position of the cursor, (*LAG|LEAD*) provides access to a row at a given physical offset (*prior|beyond*) that position – *LAG(value\_expr [, offset ] [, default ]) OVER ([ query\_partition\_clause ]order\_by\_clause)*
- *NTILE* - divides an ordered data set into the number of buckets as indicated and assigns the appropriate bucket number to each row – *NTILE(expr) OVER ([ query\_partition\_clause ] order\_by\_clause)*
- *ROW\_NUMBER* - assigns a unique number to each row within a group in the ordered sequence of rows specified in the *order-by-clause* – *ROW\_NUMBER( ) OVER ([ query\_partition\_clause ]order\_by\_clause)*
- *RATIO\_TO\_REPORT* - calculates the ratio of a value to the sum of a set of values – *ratio\_to\_report(expr) over ([query partition clause])* if *expr* is null, then *ratio\_to\_report* value is null as well value set is determined by the query partition clause (if the query partition clause is omitted, *ratio-to-report* is calculated over all returned rows)