

# Oracle SQL Function Cheat Sheet

http://www.databasestar.com/oracle-sql-functions

#### **Aggregate Functions**

AVG ( [DISTINCT|ALL] expression ) [OVER ( analytic\_clause )] COUNT ([\*|[DISTINCT|ALL]expression)[OVER (analytic clause)] MAX ([DISTINCT|ALL] expression) [OVER (analytic\_clause)] MIN ([DISTINCT|ALL] expression) [OVER ( analytic clause )] SUM ([DISTINCT|ALL] expression) [OVER (analytic\_clause)]

#### **Conversion Functions**

BIN\_TO\_NUM ( expression\_list ) CAST (expression AS type name) CAST (MULTISET (subquery) AS type\_name ) COALESCE (expr1, expr2, [expr...]) CHARTOROWID (input\_char) FROM\_TZ ( timestamp\_value, timezone\_value ) HEXTORAW (charvalue) NUMTODSINTERVAL ( number, interval\_unit ) NUMTOYMINTERVAL ( number, interval\_unit ) RAWTOHEX (charvalue) RAWTONHEX (raw) ROWIDTOCHAR (rowid) ROWIDTONCHAR (rowid) SCN TO TIMESTAMP (number) TIMESTAMP\_TO\_SCN (timestamp) TO\_BINARY\_DOUBLE ( expression [, format [, nlsparam ]]) TO\_CHAR( input\_value, [format\_mask], [nls\_parameter]) TO\_CLOB (input\_string) TO DATE( charvalue, [format mask], [nls date language]) TO\_DSINTERVAL (input\_string[, nlsparam]) TO\_LOB ( long\_value ) TO\_MULTI\_BYTE ( string ) TO\_NCHAR (input\_string) TO NCHAR (input datetime [, format [,nlsparam ]]) TO\_NCHAR ( number [, format [, nlsparam ] ] ) TO\_NCLOB (lob\_value) TO\_NUMBER (input\_value, [format\_mask], [nls\_parameter]) TO\_SINGLE\_BYTE (input\_string) TO\_TIMESTAMP (input\_string, [format\_mask], ['nlsparam']) TO\_TIMESTAMP\_TZ ( input\_string [, format\_mask] [, nls\_param]) TO\_YMINTERVAL (input\_string)

### **Date and Time Functions**

UNISTR (string)

ADD\_MONTHS (input\_date, number\_months) CURRENT\_DATE CURRENT\_TIMESTAMP ([precision]) **DBTIMEZONE** LAST\_DAY (input\_date) LOCALTIMESTAMP (timestamp\_precision) MONTHS\_BETWEEN (date1, date2) NEW\_TIME ( input\_date, timezone1, timezone2 ) NEXT\_DAY ( input\_date, weekday ) **SESSIONTIMEZONE** SYS\_EXTRACT\_UTC ( datetime\_with\_timezone\_value) SYSDATE SYSTIMESTAMP TZ\_OFFSET ( timezone\_name | time\_value | SESSIONTIMEZONE | DBTIMEZONE )

# **Environment Functions**

CON\_DBID\_TO\_ID ( container\_dbid ) CON GUID TO ID (container guid) CON\_NAME\_TO\_ID ( container\_name ) CON\_UID\_TO\_ID ( container\_uid ) ORA INVOKING USER ORA\_INVOKING\_USERID() SYS\_CONTEXT (namespace, parameter [, length]) SYS GUID() SYS\_TYPEID ( object\_type\_value ) **USER** USERENV (parameter) **SQLCODE** SQLERRM ( error\_number )

# **NLS Functions**

NLS\_CHARSET\_DECL\_LEN ( byte\_count, char\_set\_id) NLS\_CHARSET\_ID ( string\_value ) NLS\_CHARSET\_NAME ( number )

#### **Numeric and Maths Functions**

ABS (number) ACOS (number) ASIN (number) ATAN2 (number1 [/|,] number2) BITAND (expr1, expr2) CEIL (input\_val) CORR (expression1, expression2) COS ( number ) COSH (number) COVAR\_POP (expression1, expression2) [OVER (analytic\_clause)] COVAR\_SAMP (expression1, expression2) [OVER (analytic\_clause)] CUME\_DIST (expression1, ... expression\_n) WITHIN GROUP (ORDER BY expression\_order1, ... expression\_order\_n) CUME\_DIST() OVER ( [query\_partition\_clause] ORDER BY order\_clause) DENSE\_RANK ( expr, [expr(n)] ) WITHIN GROUP ( ORDER BY (order\_expr [ASC|DESC] [NULLS FIRST|LAST]) DENSE\_RANK() OVER ( [query\_partition\_clause] order\_by\_clause) EXP (number) EXTRACT ( date\_component FROM expression ) FLOOR (input\_number) GREATEST (expr1, [expr\_n]) LEAST (expr1, [expr\_n]) LN ( number ) LOG ([base, ] expression) MEDIAN ( expr ) [OVER (query\_partition\_clause)] MOD ( numerator, denominator ) ORA HASH (expression [, max bucket [,

seed\_value ]]) PERCENT\_RANK (expression) WITHIN GROUP ( ORDER BY (expression\_n [. DESC | ASC ] [NULLS FIRST|LAST]) PERCENT\_RANK () OVER ( [query\_partition\_clause] order\_by\_clause) PERCENTILE\_CONT (expression) WITHIN GROUP (ORDER BY expression [ASC | DESC] [OVER ( query\_partition\_clause) PERCENTILE\_DISC (expression) WITHIN GROUP ( ORDER BY expression [ ASC | DESC ] [OVER ( query\_partition\_clause)

POWER (n2, n1) RANK (expr) WITHIN GROUP (ORDER BY ( order\_expr [NULLS FIRST/LAST])) RANK () OVER ( [query\_partition\_clause]

order\_by\_clause) REMAINDER (n2, n1)

ROUND (input, roundto) **ROWNUM** 

ROW\_NUMBER () OVER ( [ query\_partition\_clause] order\_by\_clause)

SIGN (number) SIN (number) SINH (number) SQRT (number)

STANDARD\_HASH (expression [, method]) STDDEV ([DISTINCT | ALL] expression)[OVER

(analytical\_clause) ] STDDEV\_POP (expression) [OVER (

analytic clause)]

STDDEV\_SAMP (expression) [OVER ( analytic\_clause)]

TAN (number) TANH ( number ) TRUNC (date, fmt)

TRUNC(number, decimals)

VAR\_POP ( expression) [OVER ( analytic\_clause )] VAR\_SAMP ( expression) [OVER ( analytic\_clause )] VARIANCE ([DISTINCT | ALL] expression) [OVER ( analytic\_clause ) ]

WIDTH\_BUCKET (expression, min\_value, max\_value, num\_buckets)

#### **String and Character Functions**

ASCII (charvalue) ASCIISTR (charvalue) CHR ( number\_code [USING NCHAR\_CS] ) COMPOSE (input\_value) CONCAT( string1, string2 ) CONVERT (input\_char, dest\_char\_set, [source\_char\_set]) DECODE (expression, search, result [, search, result]... [,default]) DECOMPOSE (input\_string [CANONICAL|COMPATIBILITY]) DUMP (expression [, return\_format] [, start\_position] [, length])

INSTR (string, substring, [start\_position], [occurrence]) INSTR2 (string, substring, [start\_position], [occurrence]) INSTR4 (string, substring, [start\_position], [occurrence]) INSTRB (string, substring, [start\_position], [occurrence]) INSTRC (string, substring, [start\_position], [occurrence]) LISTAGG (measure\_expr [, delimiter]) WITHIN GROUP (order\_by\_clause) [OVER query\_partition\_clause] LENGTH (string\_value) LENGTH2 ( string\_value ) LENGTH4 (string value) LENGTHB ( string\_value ) LENGTHC (string\_value) LOWER (input\_string) LPAD( expr, length [, pad\_expr] ) LTRIM( input\_string, [trim\_string] ) LNNVL (condition) NCHR (number\_code) NLS\_INITCAP ( input\_char [, nlsparam ] ) NLS\_LOWER (input\_char[, nlsparam]) NLS\_UPPER ( input\_char [, nlsparam ] ) NLSSORT ( input\_char [, nlsparam ] ) NANVL (check value, replace value) NVL (check\_value, replace\_value) NVL2 (value\_to\_check, value\_if\_not\_null, value\_if\_null) NULLIF (expr1, expr2) REGEXP\_COUNT (source\_char, pattern [, position [, match\_pattern [, subexpression ]]]) REGEXP\_INSTR (source\_char, pattern [, position [, occurrence [, return\_option [, match\_pattern [, subexpression ]]]]]) REGEXP\_REPLACE (source\_char, pattern [, replace\_string [, position [, occurrence [, match parameter ]]]]) REGEXP\_SUBSTR (source\_char, pattern [, position [, occurrence [, match\_parameter ] ] ] ] ) REPLACE ( whole\_string, string\_to\_replace, [replacement\_string]) RPAD (expr, length [, pad\_expr]) RTRIM (input\_string, [trim\_character]) SOUNDEX (string) SUBSTR (string, start\_position, [length]) TRANSLATE (source, from\_string, to\_string) TRANSLATE (charvalue USING {CHAR\_CS|NCHAR\_CS}) TREAT (expression AS [REF][schema.]type) TRIM ([[LEADING | TRAILING | BOTH] trim\_character FROM ] trim\_source ) UPPER (input\_string) VSIZE (expression) **Analytic Functions** 

INITCAP (input\_string)

FIRST\_VALUE (expression [IGNORE NULLS]) OVER (analytic\_clause) LAST\_VALUE ( expression [ IGNORE NULLS ] ) OVER ( analytic\_clause ) LAG (expression [, offset [, default]]) OVER ([ query\_partition\_clause ] order\_by\_clause ) LEAD (expression [, offset [, default]]) OVER ([ query\_partition\_clause ] order\_by\_clause ) NTILE (expression) OVER ( [query\_partition\_clause] order\_by\_clause) RATIO\_TO\_REPORT( expression ) OVER ( [query\_partition\_clause])

# **Other Functions**

CASE [expression] WHEN condition\_1 THEN result\_1 WHEN condition\_n THEN result\_n **ELSE** result END case\_name

SYS\_CONNECT\_BY\_PATH (column, character\_separator)

# **Grouping Functions**

GROUP ID() GROUPING (expression) GROUPING\_ID ( expression1 [, expression\_n ] )

# Large Object Functions

BFILENAME (directory, filename) EMPTY\_BLOB() EMPTY\_CLOB ()