

Structure keywords		Structure keywords (cont)	
from	Point to a table for the query. Specify a table name using <code>from</code> ☒ <code>from my.app.web.auth</code>	decode	switch/case equivalent clause to set conditionally column values ☐ <code>decode (column, checkValue, value, [checkValue])</code> ☐ Each pair of arguments <code>[check Value, value]</code> ☐ <code>decode (statusCode, 200, "Successful", 400, "Error") as desc</code>
where	Where clause to filter results. ☐ <code>where [filter1 expression], [filter2 expression]</code> ☐ String values in expressions have to be surrounded by double quotes, single quotes are not	nvl	Null-Coalescing operator. Allow to set an alternate value when null ☒ <code>nvl(column, alternate_value_when_null)</code>
select	Add column to result set: ☒ <code>select source_column or column operation as destination_column</code> ☒ <code>select uriHost(uri) as host</code>	Aggregation functions and operators	
group every / by	Group clause with an optional server and client aggregation period filter: ☒ <code>group [every server_period] [by column]</code> ☐ <code>group by statusCode</code> ☒ <code>group every 10m</code> ☒ <code>group every 10m by statusCode every 1m</code>	avg	Returns the average of a range of values <code>[avg]</code> or only over the results on each group: ☐ <code>avg (column) ; nnavg (column)</code>
every	Client aggregation filter: ☒ <code>every period</code> ☐ Period syntax: an integer number followed by a symbol indicating the time period. <code>[seconds]</code> ☐ <code>every 0</code> means no client period. ☐ <code>every 5m by serverIp</code>	count	Returns the count of results on each group: ☐ <code>count ([column])</code> ☐ With argument, include only not null entries
ifthenelse	if/then/else equivalent clause to set conditionally column values: ☐ <code>ifthenelse (condition, errorValue, successValue)</code> ☐ <code>select ifthenelse (statusCode != 200, "Error", "Successful") as statusCodeDesc</code>	first / nnfirst	Returns the first or the not null first entry of the results on each group: ☐ <code>first (column) ; nnfirst (column)</code>



By [Rafa Hernández \(elpluto\)](#)
cheatography.com/elpluto/

Published 2nd January, 2023.
Last updated 2nd January, 2023.
Page 1 of 8.

Sponsored by [ApolloPad.com](#)
Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

Aggregation functions and operators (cont)

last / nnlast Returns the **last** or the **not null last** entry of the results on each group:

☐ **last**(column); **nnlast**(column)

max / min Returns the **maximum** or the **minimum** value for the columns provided, on each group:

☐ **max/min**(col1, [col2], [col3]...)

median Returns the statistical **median** for a column on each group:

☐ **median**(column)

☐ **Restricted to columns of integer type**

sum Returns the **sum** of the results on each group:

☐ **sum**(column)

sum2 Returns the **sum** of the squares of the results on each group:

☐ **sum2**(column)

percentile5 Returns the specific statistic **percentileN**, using linear interpolation, of the results on each group:

percentile10

percentile25

percentile75

percentile90

percentile95

☐ **percentile[N]**(column)

stddev / nnstddev Returns the **biased standard deviation** [stddev] of the values or not null values [nnstddev] of the results on each group:

☐ **stddev / nnstddev**(column)

☐ **Biased**

Aggregation functions and operators (cont)

stddev / nnstddev Returns the **unbiased standard deviation** [ustddev] of the values or not null values [unnstddev] of the results on each group:

☐ **ustddev / unnstddev**(column)

☐ **Unbiased**

var / nnvar Returns the **biased variance** [var] of the values or not null values [nnvar] of the results on each group:

☐ **var/nnvar**(column)

☐ **Biased**

uvar / nnuvar Returns the **unbiased variance** [uvar] of the values or not null values [nnuvar] of the results on each group:

☐ **uvar/nnuvar**(column)

☐ **Unbiased**

hllpp Returns the estimated count of distinct values of each group using the **HyperLogLog++** algorithm:

☐ **hllpp**(column)

☐ **Applies on DC (distinct count) data**

hllppcount Returns the estimated count of distinct values of each group using the **HyperLogLog++** algorithm:

☐ **hllppcount**(column)

☐ **Applies on float or integer data**



By [Rafa Hernández \(elpluto\)](#)
cheatography.com/elpluto/

Published 2nd January, 2023.
Last updated 2nd January, 2023.
Page 2 of 8.

Sponsored by [ApolloPad.com](#)
Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

String operators and functions		String operators and functions (cont)	
has, [->]	Case sensitive contains comparison. Using the operator -> only allows check one value: ☐ has (column, value1, [value2],...) ☐ column -> value1	replace	Replaces only first occurrence of a string with a substitute: <code>replace(c column, string ToSearch, string ToReplace)</code>
weakhas	Case insensitive contains comparison: <code>weakhas(c column, value)</code>	replaceall	Replaces all occurrences of a search string with a substitute: <code>replaceall(c column, string ToSearch, string ToReplace)</code>
in, [<-]	Case sensitive is contained comparison. Using the operator '<-' allows only one value: ☒ in (value1, [value 2], [...], column) ☒ value1 <- column	split	Returns a specific piece of splitting operation by a separator: ☒ pieceNumber begin at 0. <code>split(c column, separator String, pieceNumber)</code>
weakin	Case insensitive is contained comparison: <code>weakin(value, column)</code>	splitre	Returns a specific piece of splitting operation by a regular expression: ☒ pieceNumber begin at 0. <code>splitre(c column, re(string) or regexp, pieceNumber)</code>
startswith	Returns strings that start with specific value: <code>startswith(c column, value)</code>	substring	Returns a substring beginning at specific index with the provided length: <code>substring(c column, index, length)</code>
endswith	Returns strings that end with a specific value: <code>endswith(c column, value)</code>	subs	Returns a string replacing first substring occurrence based on template string as substitution value: ☒ FailValue is returned when is provided and no match is found. ☒ <code>subs(c column, regexp, template, [failValue])</code> ☒ <code>subs(c column, re(string), template, [failValue])</code>
toktakes	Specialized contains function for ASCII delimited tokens: <code>toktakes(c column, value, [bool_left], [bool_right])</code>		
length	Returns the length of a string value: <code>length(c column)</code>		
locate	Returns the position of a substring, indexOf function: <code>locate(c column, substring_to Locate)</code>		
lower	Returns the transformation to lower case: <code>lower(c column)</code>		
upper	Returns the transformation to upper case: <code>upper(c column)</code>		



By **Rafa Hernández (elpluto)**
cheatography.com/elpluto/

Published 2nd January, 2023.
 Last updated 2nd January, 2023.
 Page 3 of 8.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopad.com>

String operators and functions (cont)

subsall	Returns a string replacing all substring occurrences based on a regular expression and a template string as substitution value: 👁 FailValue is returned when is provided and no occurrences found ✔ subs(c o l u m n, regexp, template, [failValue]) ✔ subs(c o l u m n, re(str ing), templa te(str ing), [failV alue])
trim	Returns the result of trimming both sides : trim(c o l u m n)
ltrim	Returns the result of trimming left side : ltrim(c o l u m n)
rtrim	Returns the result of trimming right side : rtrim(c o l u m n)
matches, [~]	Matches function that finds occurrences in a column using a regular expression: ✔ matche s(c o l u m n, re(string) or regexp value) ✔ column ~ re(string) or regexp value
peek	Returns the part of a string based on a regular expression, optionally indicating a specific part occurrence: 👁 If no partNumber is provided then returns first part occurrence peek(column, re(string) or regexp, [partN umber])
formatnumber	Format a number with a specific mask and locale: format num ber (nu mbe rCo l u m n, mask, locale) 👁 format num ber (to tal Amount, " ###.## ", " en- GB")

String operators and functions (cont)

damerau	Returns Damerau distance: damera u(c o l u m n, value)
hamming	Returns Hamming distance: hammin g(c o l u m n, value)
levenshtein	Returns Levenshtein distance: levens hte in(c o l u m n, value)
osa	Returns osa distance: osa(co l u m n, value)
publicsuffix	Returns the main public suffix of a hostname: publicsuffix(hostnameColumn) 👁 'www.m y.s ite.co.uk' = 'co.uk'
rootdomain	Returns the root domain of a hostname part of an url: rootdomain(hostnameUrlColumn) 👁 'www.m y.s ite.com' = 'site'
rootprefix	Returns the root prefix of a hostname part of an url: rootprefix(hostnameUrlColumn) 👁 'www.m y.s ite.com' = 'www.m y.s ite'
rootsuffix	Returns the root suffix of a hostname part of an url: rootsuffix(hostnameUrlColumn) 👁 'www.m y.s ite.com' = 'my.si te.c om'
subdomain	Returns the subdomain of a hostname part of an url: subdomain(hostnameUrlColumn) 👁 'www.m y.s ite.com' = 'www'
topleveldomain	Returns the top level domain of a hostname part of an url: topleveldomain(hostnameUrlColumn) 👁 'www.m y.s ite.co.uk' = 'uk'
shannonentropy	



By **Rafa Hernández (elpluto)**
cheatography.com/elpluto/

Published 2nd January, 2023.
 Last updated 2nd January, 2023.
 Page 4 of 8.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

Cryptographic functions

md5

sha1

sha256

sha512

Web functions

urischeme

urihost

uriport

uripath

urifragment

uriquery

uriuser

urissp

uriauthority

absoluteuri

opaqueuri

urldecode

uauri

uaname

uatype

uaversion

uaicon

uarobot

uainfouri

uafamily

uacompany

uacompanyuri

uadeviceicon

uadeviceinfouri

uadevicetype

uaosuri

uaosname

uaosicon

uaosfamily

uaoscompany

uaoscompanyuri

uaosversion Possible missing function to filter by OS version.

Meta functions

pragmavalue

tablename

Data Types

str String

int Integer number: 1, 58, 12598

float Floating point number: 24.256

boolean Boolean: true, false

timestamp Timestamp date in format: yyyy-MM-dd HH:mm:ss.

boxar(int) Byte array in hexadecimal string format

duration Amount of time: an integer following by a letter
[d]ays, [h]ours, [m]inutes, [s]econds, [ms]econds

geocord Geographic coordinates set:
✔ Latitude/ longitude sexagesimal value
✔ Hash representation of coordinates

ip IPv4 address format: 192.168.5.56

ip6 IPv6 address format: 2001:0db8:85a3:0007:334

net4 IPv4 address in format: {x.x.x.x}/0

net6 IPv6 address in format: x.x.x.x.x.x/s

regexp Regular expression: [^\w]

template Represents a substitution string mask.

dc Represents a estimated count of distinct elements in a

image Image as Base64 encoding image.

mac MAC address in format: 00:0a:95:9d:68:16

namepattern Represents a part of a table name: my.app, demo

set(name) Represents a set of table names: {my.app.test

json String in json format: {"id":345, "name":



By Rafa Hernández (elpluto)
cheatography.com/elpluto/

Published 2nd January, 2023.
Last updated 2nd January, 2023.
Page 5 of 8.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>

Data Types (cont)

jq Represents a **jq** filter, **jq** is a command line json processor.
 .email

Common comparison functions and operators

eq, [=] **Equals to** function and operator:
 ✓ `eq(column, value or column)`
 ✓ `column1 = value or column`

eqic Case insensitive **Equals to** function:
`eqic(column, value or column)`

ge, [>=] **Greater or equal** function and operator:
 ✓ `ge(column, value or column)`
 ✓ `column >= value or column`

gt, [>] **Greater than** function and operator:
 ✓ `gt(column, value or column)`
 ✓ `column > value or column`

le, [≤] **Less or equal** function and operator:
 ✓ `le(column, value or column)`
 ✓ `column ≤ value or column`

lt, [<] **Less than** function and operator:
 ✓ `lt(column, value or column)`
 ✓ `column < value or column`

ne, [≠] **Not equal** function and operator:
 ✓ `ne(column, value or column)`
 ✓ `column ≠ value or column`

isnull Check if **is null** function:
`isnull (column)`

isnotnull Check if **is not null** function:
`isnotnull (column)`

Logig Functions

and

or

not

JSON related functions

jqeval

label

jsonparse

Math functions and operators

abs

add / [+]

sub / [-]

mul / [*]

div / [/]

rdiv / [/] Real division function and operator:

mod / [%%] Module function:

rem / [%] Return the remain of a division operation:

pow Power function:

cbrt Cube root function:

sqrt Square root function:

ceil

floor

round

signum

Statistical and specialised statistical functions

estimation

pack

unpackhllpp

Network functions

ispublic

isprivate

ipip4

ipprotocol

purpose

host

routing

httpstatusdescription

httpstatustype

reputation

score

sbl



By **Rafa Hernández (elpluto)**
cheatography.com/elpluto/

Published 2nd January, 2023.
 Last updated 2nd January, 2023.
 Page 6 of 8.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

Conversion functions

int

str

bool

float

image

ip4

net4

ip6

net6

mac

to16

from16

to64

from64

toutf8

fromutf8

toz85

fromz85

compatible

mapped

translated

template

timestamp

duration

re

parsedate

formatdate

humansize

mkboxar

Special comparison functions

matches Matches function finds occurrences in a column using a regular expression:

- ✓ `matche s(c olumn, re(string) or regexp valu e)`
- ✓ `column ~ re(string) or regexp value`

Special comparison functions (cont)

anymatches Find occurrences in a **set of names** type column against

```
anymat che s(s etO fNames, namegl ob( str: ttern))
```

👁 `anymat che s(t ables, namegl ob( " my.:`

nameglob Return a formatted string as a **namepattern** to use with

```
namegl ob( string)
```

Date and time functions

day

dayofweek

dayofyear

month

year

epoch

hour

minute

second

millisecond

today

tomorrow

yesterday

period

Packet functions

hasio4

hastcp

hasudp

hasether

ip4proto

ip4src

ip4dst

ip4status

ip4ttl

ip4len

ip4payload

ip4flags



By **Rafa Hernández (elpluto)**
cheatography.com/elpluto/

Published 2nd January, 2023.
Last updated 2nd January, 2023.
Page 7 of 8.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopadd.com>

Packet functions (cont)

ip4fragment

ip4cs

ip4hl

ip4ds

ip4ecn

ip4tos

etherdst

ethersrc

etherpayload

etherstatus

ethertag

ethertype

tcpdst

tcpsrc

tcpstatus

tcpflags

tcppack

tcpcs

tcpseq

tcpnl

tcppayload

tcpurg

tcpwin

udpsrc

udpport

udpstatus

udpcs

udplen

udppayload



By **Rafa Hernández** (elpluto)
cheatography.com/elpluto/

Published 2nd January, 2023.
Last updated 2nd January, 2023.
Page 8 of 8.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopapad.com>