**Qdb::out**( estrarre i dati ) 1:1 **SINGOLO**

**Key Primaria** può essere una Key Base o una Key Combinata**KEY Base** 🡪 Qdb::out( ‘email**’** , ’rafcrack71@gmail.com’ ) Qdb::out( ‘punti’ , 47 , -2 )  
**KEY Combinata** 🡪 Qdb::out( ‘email.id**’** , ’rafcrack71@gmail.com’ , -1 ) Qdb::out( ‘iscritti.punti’ , 47, -3 )  
Ritorna un ***Array ( chiavi = valori )*** se esiste il valore della KEY Primaria oppure ***false***  in caso contrario.  
P.S.: se dopo il valore inserisci **-1** avrai il timestamp dell’inserimento di ogni singolo dato, se metti **-2** da differenza fra ora  
e i timestamp di ogni singolo dato, se metti **-3** il timestamp della creazione del valore associato alla Key Primaria.

Guardiamo il primo esempio dove ritorna il valore dell’ Array collegato all’ email = rafcrack71@gmail.com ( nessun timestamp )

Array

(

[K] => Array

(

[0] => id

[1] => password

[2] => stato

[3] => nome

[4] => cognome

[5] => #sito

[6] => @fattura

)

[#sito] => Array

(

[0] => 2

[1] => 1

)

[N.#sito] => 2

[T.#sito] => 2

[@fattura] => Array

(

[0] => 3.3

[1] => 2.2

[2] => 1.1

)

[N.@fattura] => 3

[T.@fattura] => 3

[id] => pASgkZaUJrE45zoL

[password] => 123456

[stato] => true

[nome] => Raffaele

[cognome] => Piazzi

[N] => 7

[T] => 4

)

**Key Associata** può essere una Key Base o una Key Combinata associata ad una Key Primaria  
**KEY Base** 🡪 Qdb::out( ‘email **nome’** , ’rafcrack71@gmail.com’ ) Qdb::out( ‘iscritti.punti **totale**’ , 47 )  
**KEY Base**  🡪 Qdb::out( ‘email **nome**,**cognome**,**stato’** , ’rafcrack71@gmail.com’ , -1 )  
**KEY Combinata** 🡪 Qdb::out( ‘email.id **nome’** , ’rafcrack71@gmail.com’ , -2 )  
Ritorna il ***Valore*** della KEY Associata se esiste oppure ***false***  in caso contrario.

Nel primo esempio ritorna il valore nel nome associato all’ email = rafcrack71@gmail.com ( **Raffaele** )  
Nel secondo esempio ritorna il valore nel totale associato agli iscritti.punti = 47 ( **23906** )  
Nel terzo esempio ritorna un **Array con più KEY** ( il separatore è la virgola ) con i relativi timestamp di inserimento ( , -1 )

Array

(

[K] => Array

(

[0] => nome

[1] => cognome

[2] => stato

)

[nome] => Raffaele

[t.nome] => 1417340675

[cognome] => Piazzi

[t.cognome] => 1417340681

[stato] => true

[t.stato] => 1417333345

[N] => 3

[T] => 4

)

Nel quarto esempio invece ritorna un **Array semplice** con la differenza fra ora e il timestamp di inserimento del dato ( , -2 )

Array

(

[K] => nome

[nome] => Raffaele

[t.nome] => 24545

)

**Qdb::out**( estrarre i dati ) 1:1 **LISTA**

**Key Primaria** può essere una Key Base o una Key Combinata**KEY Base** 🡪 Qdb::out( ‘email(3)**’** ) Qdb::out( ‘punti’ )  
**KEY Combinata** 🡪 Qdb::out( ‘email.id(2:3)**’** , -1 ) Qdb::out( ‘iscritti.punti’ , -2 )  
Ritorna un ***Array ( chiavi = valori )*** se esiste oppure ***false***  in caso contrario.  
P.S.: se dopo la KEY inserisci **-1** avrai il timestamp dell’inserimento dei dati, se metti **-2** da differenza fra ora e quei timestamp.

Guardiamo il primo esempio dove ritorna gli ultimi 3 valori in ordine di inserimento dell’ Array collegato alla KEY: email  
( nessun timestamp ) scrivere email(3) equivale a email(0:2) se invece li voglio invertiti esiste solo la sintassi: email(2:0)  
un esempio potentissimo di questa tecnica è la paginazione di una query: email(0:49) email(50:99) email(100:149)  
infine se NON mettiamo niente otteniamo il default ( 10 risultati ) se mettiamo 0 🡪 email(0) il massimo ( 100 risultati )

Array

Qdb::out( ‘email(3)**’** )  
Qdb::out( ‘email(0:2)**’** )

(

[0] => snik737@gmail.com

[1] => la\_tua\_email@gmail.com

[2] => piazzi.raffaele@libero.it

[N] => 3

[T] => 4

)

Oltre al riordino per posizione, I dati possono essere cercati per “ tempo ”, la query va eseguita fra parentesi quadre e il giorno (gg) può essere scambiato con l’anno (aaaa), quindi [gg-mm-aaaa] equivale a [aaaa-mm-gg] ecco tutte le combinazioni:

**per singolo giorno** [gg-mm-aaaa] : email[30-11-2014]  
**da giorno a giorno** diretto [gg-mm-aaaa:gg-mm-aaaa] : email[27-11-2014**:**30-11-2014]  
**da giorno a giorno** inverso [gg-mm-aaaa:gg-mm-aaaa] : email[30-11-2014**:**27-11-2014]

Ancora più interessante il fatto che si possano specificare le ore (hh) i minuti (mm) e i secondi (ss) ecco tutte le combinazioni:

**precisione al secondo** [ss] : email[26] => ultimi 26 secondi  
**precisione al minuto** [mm.ss] : email[15.03] => ultimi 15 minuti e 3 secondi ( 903 secondi )  
**precisione oraria** [hh.mm.ss] : email[9.4.28] => ultime 9 ore 4 minuti e 28 secondi ( 32668 secondi )  
**da ora a ora** diretto [hh.mm.ss:hh.mm.ss] : email[8.3.44**:**23.9.11] oppure misto email[8.03**:**21] => dalle 08.03.00 alle 21.59.59\*  
**da ora a ora** inverso [hh.mm.ss:hh.mm.ss] : email[23.9.11**:**8.3.44] oppure misto email[21.09**:**8] => dalle 21.09.59 alle 08.00.00\*

\*Se non viene completato l’orario - Quantico completerà quello maggiore con dei .59 e quello inferiore con dei .00  
  
  
Infine combiniamo i giorni con le ore unendoli con un = ( uguale ) si possono scambiare in molti modi ad esempio:   
[GG-MM-AAAA=hh.mm.ss:GG-MM-AAAA=hh.mm.ss] equivale a [hh.mm.ss:GG-MM-AAAA:hh.mm.ss=GG-MM-AAAA]  
oppure singolarmente [GG-MM-AAAA=hh.mm.ss] ma anche [hh.mm.ss:GG-MM-AAAA] le possibilità sono molte:

**precisione oraria** [GG-MM-AAAA=hh] : email[30-11-2014=15] => 30-11-2014 dalle 15.00.00 alle 15.59.59  
**precisione al minuto** [GG-MM-AAAA=hh.mm] : email[27-6-2014=15.20] => 27-06-2014 dalle 15.20.00 alle 15.20.59  
**precisione al secondo** [GG-MM-AAAA=hh.mm.ss] : email[18-10-2014=23.7.51] => 18-10-2014 dalle 23.07.51 alle 23.07.51  
**da giorno**=**ora a giorno**=**ora** diretto : email[12-10-2014=13.15**:**19-10-2014=8] => 12-10-2014 = 13.15.00 a 19-10-2014 = 08.59.59\*  
**da giorno**=**ora a giorno**=**ora** inverso : email[12-10-2014=13.15**:**11-10-2014=8] => 12-10-2014 = 13.15.59 a 11-10-2014 = 08.00.00\*

\*Se non viene completato l’orario - Quantico completerà quello del giorno maggiore con dei .59 e quello del giorno inferiore con dei .00

Bene non ci rimane che gestire il flusso dei dati scegliendo le posizioni che vogliamo vedere quindi come al solito se non mettiamo niente otteniamo il default ( 10 risultati ) se mettiamo 0 🡪 ( 100 risultati ) vediamo qualche esempio:

**precisione oraria** con risultati di default [GG-MM-AAAA=hh] : email[30-11-2014=15] ( normalmente 10 ) **precisione oraria** con risultati massimi [GG-MM-AAAA=hh](0) : email[30-11-2014=15](0) ( normalmente 100 )  
**precisione al minuto** con 20 risultati da 10 a 29 [GG-MM-AAAA=hh.mm](10:29) : email[27-06-2014=15.20](10:29)  
**precisione al minuto** con 20 risultati da 29 a 10 [GG-MM-AAAA=hh.mm](29:10) : email[27-06-2014=15.20](29:10)

**Qdb::out**( estrarre i dati ) 1:1 **LISTA**

**Key Associata** può essere una Key Base o una Key Combinata associata ad una Key Primaria  
**KEY Base** 🡪 Qdb::out( ‘email(3) nome**’** ) Qdb::out( ‘punti totali’ )  
**KEY Combinata** 🡪 Qdb::out( ‘email.id(1:3) nome,cognome**’** , -1 )  
Ritorna un ***Array ( chiavi = valori )*** se esiste oppure ***false***  in caso contrario.  
P.S.: se alla fine inserisci **-1** avrai il timestamp dell’inserimento dei dati, se metti **-2** da differenza fra ora e quei timestamp.

Array

(

[0] => la\_tua\_email@gmail.com

[t.0] => 1417279000

[1] => piazzi.raffaele@libero.it

[t.1] => 1417255352

[2] => rafcrack71@gmail.com

[t.2] => 1417199747

[K] => Array

(

[0] => nome

[1] => cognome

)

[nome] => Array

(

[0] =>

Qdb::out( ‘email.id(1:3) nome,cognome’ , -1)

[1] =>

[2] => RAF

)

[t.nome] => Array

(

[0] =>

[1] =>

[2] => 1417199772

)

[N.nome] => 1

[cognome] => Array

(

[0] => il tuo cognome

[1] => Piazzi

[2] =>

)

[t.cognome] => Array

(

[0] => 1417283184

[1] => 1417340681

[2] =>

)

[N.cognome] => 2

[N] => 3

[T] => 4

)

Oltre al riordino per tempo e per posizione esiste anche il riordino numerico, capiamo come funziona:  
  
**per valore** {numero} : Qdb::out( ‘email eta{30}**’** )  
**minore di** {numero} : Qdb::out( ‘email eta{<30}**’** )  
**maggiore di** {numero} : Qdb::out( ‘email eta{>30}**’** )  
**da valore a valore** diretto [minore:maggiore] : Qdb::out( ‘email eta{20:40}**’** )  
**da valore a valore** inverso [maggiore:minore] : Qdb::out( ‘email eta{40:20}**’** )

Con la possibilità di ordinare in modo crescente ( + ) decrescente ( - ) per quantità ( ! ) ecco tutte le combinazioni:  
  
**crescente** +{minore maggiore o intervallo} : Qdb::out( ‘email eta+{<25.67}**’** )  
**decrescente** -{minore maggiore o intervallo} : Qdb::out( ‘email eta-{>8.99}**’** )  
**per quantità** !{minore maggiore o intervallo} : Qdb::out( ‘email !eta{30:60}**’** )  
**per quantità crescente** !{minore maggiore o intervallo}+! : Qdb::out( ‘email !eta+!{>30}**’** )  
**per quantità decrescente** !{minore maggiore o intervallo}-! : Qdb::out( ‘email !eta-!{10:50}**’** )

Array

(

[0] => email\_8@libero.it

[1] => email\_6@aruba.it

[2] => email\_5@aruba.it

[3] => email\_1@gmail.com

[4] => email\_5@alice.it

[5] => email\_9@gmail.com

[6] => email\_2@libero.it

[7] => email\_7@libero.it

[8] => email\_4@gmail.com

[9] => email\_3@libero.it

[K] => Array

Qdb::out( ‘email eta{20:40}’ )

(

[0] => eta

)

[eta] => Array

(

[0] => 21

[1] => 21

[2] => 21

[3] => 27

[4] => 28

[5] => 29

[6] => 30

[7] => 30

[8] => 30

[9] => 30

)

[N.eta] => 10

[N] => 10

[T] => 22

)

Array

(

[K] => Array

(

[0] => eta

)

[eta] => Array

(

[0] => 21

[1] => 27

[2] => 28

[3] => 29

[4] => 30

[5] => 33

[6] => 39

Qdb::out( ‘email !eta{20:40}’ )

[7] => 40

)

[N.eta] => Array

(

[0] => 3

[1] => 1

[2] => 1

[3] => 1

[4] => 4

[5] => 1

[6] => 2

[7] => 1

)

[T.eta] => 14

[N] => 8

[T] => 22

)

Array

(

[K] => Array

(

[0] => eta

)

[eta] => Array

(

[0] => 30

[1] => 21

[2] => 39

[3] => 27

[4] => 28

[5] => 29

Qdb::out( ‘email !eta+!{20:40}’ )

[6] => 33

[7] => 40

)

[N.eta] => Array

(

[0] => 4

[1] => 3

[2] => 2

[3] => 1

[4] => 1

[5] => 1

[6] => 1

[7] => 1

)

[T.eta] => 14

[N] => 8

[T] => 22

)

Array

(

[K] => Array

(

[0] => eta

)

[eta] => Array

(

[0] => 27

[1] => 28

[2] => 29

[3] => 33

[4] => 40

Qdb::out( ‘email !eta-!{20:40}’ )

[5] => 39

[6] => 21

[7] => 30

)

[N.eta] => Array

(

[0] => 1

[1] => 1

[2] => 1

[3] => 1

[4] => 1

[5] => 2

[6] => 3

[7] => 4

)

[T.eta] => 14

[N] => 8

[T] => 22

)

Per completare questa sezione guardiamo l’ultimo esempio e capiamo la potenza dell’ordinamento ( + - ):  
  
**senza** : Qdb::out( ‘email(5) nome,sesso,eta**’** )  
**crescente** + : Qdb::out( ‘email(5) nome,sesso,eta+**’** )  
**decrescente** - : Qdb::out( ‘email(5) nome,sesso,eta-**’** )

Ovviamente si può combinare e ricercare per tempo [ ] , per posizione ( ) , sapere il momento esatto nel quale il dato è stato inserito ,-1 o da quanto tempo è stato inserito ,-2 come mostrato negli esempi precedenti.

P.S.: NON sto eseguendo un riordino NUMERICO , mancano le graffe { } , ma assolutamente Alfanumerico !!!

Array

(

[0] => email\_5@gmail.com

[1] => email\_10@gmail.com

[2] => email\_0@libero.it

[3] => email\_5@yahoo.it

[4] => email\_2@aruba.it

[K] => Array

(

[0] => nome

[1] => sesso

[2] => eta

)

[nome] => Array

(

[0] => nome 5

[1] => nome 10

[2] => nome 0

[3] => nome 5

[4] => nome 2

Qdb::out( ‘email(5) nome,sesso,eta’ )

)

[N.nome] => 5

[sesso] => Array

(

[0] => maschio

[1] => maschio

[2] => femmina

[3] => femmina

[4] => femmina

)

[N.sesso] => 5

[eta] => Array

(

[0] => 65

[1] => 33

[2] => 47

[3] => 56

[4] => 39

)

[N.eta] => 5

[N] => 5

[T] => 22

)

Array

(

[0] => email\_10@gmail.com

[1] => email\_2@aruba.it

[2] => email\_0@libero.it

[3] => email\_5@yahoo.it

[4] => email\_5@gmail.com

[K] => Array

(

[0] => nome

[1] => sesso

[2] => eta

)

[nome] => Array

(

[0] => nome 10

[1] => nome 2

[2] => nome 0

Qdb::out( ‘email(5) nome,sesso,eta+’ )

[3] => nome 5

[4] => nome 5

)

[N.nome] => 5

[sesso] => Array

(

[0] => maschio

[1] => femmina

[2] => femmina

[3] => femmina

[4] => maschio

)

[N.sesso] => 5  
 [eta] => Array

(

[0] => 33

[1] => 39

[2] => 47

[3] => 56

[4] => 65

)

[N.eta] => 5

[N] => 5

[T] => 22

)

Array

(

[0] => email\_5@gmail.com

[1] => email\_5@yahoo.it

[2] => email\_0@libero.it

[3] => email\_2@aruba.it

[4] => email\_10@gmail.com

[K] => Array

(

[0] => nome

[1] => sesso

[2] => eta

)

[nome] => Array

(

[0] => nome 5

[1] => nome 5

[2] => nome 0

[3] => nome 2

[4] => nome 10

Qdb::out( ‘email(5) nome,sesso,eta-’ )

)

[N.nome] => 5

[sesso] => Array

(

[0] => maschio

[1] => femmina

[2] => femmina

[3] => femmina

[4] => maschio

)

[N.sesso] => 5  
 [eta] => Array

(

[0] => 65

[1] => 56

[2] => 47

[3] => 39

[4] => 33

)

[N.eta] => 5

[N] => 5

[T] => 22

)