**Qdb::ver**( verificare i dati ) 1:1

**Key Primaria** può essere una Key Base o una Key Combinata**KEY Base** 🡪 Qdb::ver( ‘**email**’ , ’**rafcrack71@gmail.com**’ , -1 ) Qdb::ver( ‘**punti**’ , **47** , -2 )  
**KEY Combinata** 🡪 Qdb::ver( ‘**email.id**’ , ’**rafcrack71@gmail.com**’ ) Qdb::ver( ‘**iscritti.punti**’ , **47** )  
Se dopo il valore inserisci **-1** avrai il timestamp dell’inserimento del dato, se metti **-2** da differenza fra ora e quel timestamp.  
Ritorna **1** se Esiste ma NON ha Key Associate.  
Ritorna **2** se Esiste ed ha Key Associate.  
Ritorna **0** se NON esiste.

**Key Associata** può essere una Key Base o una Key Combinata associata ad una Key Primaria  
**KEY Base** 🡪 Qdb::ver( ‘email **nome’** , ’rafcrack71@gmail.com’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘email **nome.principale**’ , ’rafcrack71@gmail.com’ )  
Ritorna **1** se la Key Associata è Esatta.  
Ritorna **0** in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘email **nome’** , ’**Raffaele**’ , ’rafcrack71@gmail.com’ , -1 )  
**KEY Combinata** 🡪 Qdb::ver( ‘email **nome.principale**’ , ’**Raffaele**’ , ’rafcrack71@gmail.com’ )  
Se dopo il valore inserisci **-1** avrai il timestamp dell’inserimento del dato, se metti **-2** da differenza fra ora e quel timestamp.  
Ritorna **1** se il Valore della Key Associata è Esatto.  
Ritorna **0** in caso contrario.

**Qdb::ver**( verificare i dati ) 1:N

**Key Primaria** può essere una Key Base o una Key Combinata ( **#** ) **Link****KEY Base** 🡪 Qdb::ver( ‘email**#foto**’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘email**#foto.id**’ )  
Ritorna il ***TOTALE*** dei valori se la Key linkata è Esatta, ***False*** in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘email**#foto**’ , ’rafcrack71@gmail.com’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘email**#foto.id**’ , ’rafcrack71@gmail.com’ )  
Ritorna il ***TOTALE*** dei valori linkati alla Key Primaria ( rafcrack71@gmail.com ) , ***False*** in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘**#foto**’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘**#foto.id**’ )  
Ritorna ***True*** se la Key Primaria è stata linkata. Ritorna ***False*** in caso contrario.

**Key Primaria** può essere una Key Base o una Key Combinata ( **@** ) **Clone****KEY Base** 🡪 Qdb::ver( ‘email**@camicia**’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘email**@prodotti.scarpe.id**’ )  
Ritorna il ***Totale*** del Prodotto se la Key Primaria clonata è Esatta, ***False*** in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘email**@camicia**’ , ’rafcrack71@gmail.com’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘email**@prodotti.scarpe.id**’ , ’rafcrack71@gmail.com’ )  
Ritorna il ***Totale*** del Prodotto comperato dalla Key Primaria ( rafcrack71@gmail.com ) , ***False*** in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘email**@camicia**’ , ‘[**3.1.2.1.2**](javascript:guarda('s0-4-16','3.1.2.1.2@occhiale'))’ , ’rafcrack71@gmail.com’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘email**@prodotti.scarpe.id**’ , ’[**1.1.1.2.3**](javascript:guarda('s0-4-7','1.1.1.2.3@occhiale'))’ , ‘rafcrack71@gmail.com’ )  
Ritorna il ***Totale*** del Modello del Prodotto comperato dalla Key Primaria , ***False*** in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘[**3.1.2.1.2**](javascript:guarda('s0-4-16','3.1.2.1.2@occhiale'))**@camicia**’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘[**1.1.1.2.3**](javascript:guarda('s0-4-7','1.1.1.2.3@occhiale'))**@prodotti.scarpe.id**’ )  
Ritorna il ***Valore*** della Key Primaria corrispondente al modello generato in fase di clonazione ( prodotto comperato )  
Ritorna ***False***  in caso contrario.

**KEY Base** 🡪 Qdb::ver( ‘**@camicia**’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘**@prodotti.scarpe.id**’ )  
Ritorna ***True*** se la Key Primaria è stata clonata. Ritorna ***False*** in caso contrario.

**“** La clonazione ( @ ) genera modelli che hanno la proprietà di trasformare prodotti con Key Primarie 1:N  
 in modelli con Key Primarie 1:1 ---> lasciando inalterate le Key Associate. Nell’esempio la camicia 3  
 ( composta da Key 1:1 e Key 1:N ) viene trasformata nel modello @camicia 3.1.2.1.2 ( solo Key 1:1 )  
 Se vuoi vedere il modello 3.1.2.1.2 devi usare il comando: Qdb::out( ‘3.1.2.1.2@camicia’ )  
 Se vuoi vedere la camicia 3 devi usare il comando: Qdb::out( ‘camicia’ , 3 ) **“**

**Qdb::ver**( verificare i TOTALI )

**Key Primaria** può essere una Key Base o una Key Combinata**KEY Base** 🡪 Qdb::ver( ‘**email**’ ) Qdb::ver( ‘**punti**’ )  
**KEY Combinata** 🡪 Qdb::ver( ‘**email.id**’ ) Qdb::ver( ‘**iscritti.punti**’ )  
Ritorna un Array con 3 Totali:

[@] = numero di prodotti comperati 🡪 creato con: Qdb::ver( ‘**email@prodotti.camicie**’ )  
[ # ] = numero di incrementi automatici 🡪 creato con: Qdb::ver( ‘**#email.id**’ )  
[ T ] = numero totale dei valori della Key 🡪 creato con: Qdb::ver ( ‘**punti**’ , **47** )

Array

(

[K] => Array

(

[0] => nickname

[1] => eta

[2] => peso

[3] => altezza

[4] => #foto

)

[nickname] => 55

[eta] => 55

[peso] => 54

[altezza] => 55

[#foto] => 210

[@] => 0

[#] => 0

[N] => 5

[T] => 55

)

Qdb::ver( ‘**email altezza,sesso,!peso+!{40:50},eta{40:50}**’ )

Array

(

[K] => Array

(

[0] => altezza

[1] => sesso

[2] => peso

[3] => eta

)

[altezza] => 55

[sesso] => 24

[peso] => Array

(

[0] => 50

**!peso+!{40:50}**

[1] => 48

[2] => 46

[3] => 49

**Quantità ( MAX --> min )**

)

[N.peso] => Array

(

[0] => 7

[1] => 2

[2] => 1

[3] => 1

)

[T.peso] => 11

[eta] => Array

(

[0] => 40

[1] => 41

**eta{40:50}**

[2] => 42

[3] => 43

[4] => 48

[5] => 50

**Quantità ( Streaming )**

)

[N.eta] => Array

(

[0] => 6

[1] => 2

[2] => 4

[3] => 4

[4] => 2

[5] => 3

)

[T.eta] => 21

[@] => 0

[#] => 0

[N] => 4

[T] => 55

)

**Qdb::ver**( verificare i TOTALI ) **array()**

Qdb::ver( ‘**email**’, **array(**‘sesso’ **=> array(**‘maschio’,’femmina’**) ,** ‘nome’ **=> array(**‘raffaele’,’nicola’, ‘tiziana’,’jenny’**))** )

Array

(

[K] => Array

(

[0] => sesso

[1] => nome

)

[sesso] => Array

(

[0] => maschio

[1] => femmina

)

[N.sesso] => Array

(

[0] => 9

[1] => 6

)

[T.sesso] => 15

[nome] => Array

(

[0] => raffaele

[1] => nicola

[2] => tiziana

[3] => jenny

)

Case Insensitive

[N.nome] => Array

(

[0] => 5

[1] => 3

[2] => 2

[3] => 3

)

[T.nome] => 13  
 [@] => 0

[#] => 0

[N] => 2

[T] => 15

)

Qdb::ver( ‘**email[T]**’ ) ---> Ritorna il Totale dei valori della Key Primaria.

Qdb::ver( ‘**email[K]**’ ) ---> Ritorna un array con tutte le KEY della Key Primaria.

Qdb::ver( ‘**email[K]**’, ‘prova@libero.it’ ) ---> Ritorna un array con tutte le KEY  
collegate all’ email: ‘prova@libero.it’

Qdb::ver( ‘**email[ ]**’ ) ---> Ritorna un array tutte le KEY Associate della Key Primaria.

Qdb::ver( ‘**email[ ]**’, ‘prova@libero.it’ ) ---> Ritorna un array con tutte le KEY Associate  
collegate all’ email: ‘prova@libero.it’

Qdb::ver( ‘**email[#]**’ ) ---> Ritorna un array tutte le KEY Multiple della Key Primaria.

Qdb::ver( ‘**email[#]**’, ‘prova@libero.it’ ) ---> Ritorna un array con tutte le KEY Multiple  
collegate all’ email: ‘prova@libero.it’

Qdb::ver( ‘**email[@]**’ ) ---> Ritorna un array con tutte le KEY Clonate della Key Primaria.

Qdb::ver( ‘**email[@]**’, ‘prova@libero.it’ ) ---> Ritorna un array con tutte le KEY Clonate  
collegate all’ email: ‘prova@libero.it’

**Qdb::ver**( verificare le DIFFERENZE ) **array()**Solo **KEY Primaria** e **KEY Proprietaria** ( Base o Combinata )  
**1.** (niente) Qdb::ver( ‘**id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**)** )  
Qdb::ver( ‘**email#id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**),**‘rafcrack@alice.it’ )

**==>>>** Verifica i 4 valori della KEY Primaria ( id ) e ritorna un array contenente i valori che NON sono nel database.

**2.** ( , 1 ) Qdb::ver( ‘**id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**) , 1** )  
Qdb::ver( ‘**email#id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**),**‘rafcrack@alice.it’**, 1** )

**==>>>** Verifica i 4 valori della KEY Primaria ( id ) e ritorna un array contenente i valori che sono nel database.

**3.** ( , -1 ) Qdb::ver( ‘**id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**) , -1** )  
Qdb::ver( ‘**email#id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**),**‘rafcrack@alice.it’**, -1** )

**==>>>** Verifica i 4 valori della KEY Primaria ( id ) e ritorna un array contenente i valori e i timestamp che sono nel database.

**4.** ( , -2 ) Qdb::ver( ‘**id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**) , -2** )  
Qdb::ver( ‘**email#id**’, **array(**‘MF921E01F-H11’,‘PE122D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**),**‘rafcrack@alice.it’**, -2** )

**==>>>** Verifica i 4 valori della KEY Primaria ( id ) e ritorna un array contenente i valori e i timestamp che sono nel database.

**5.** ( , in ) Qdb::ver( ‘**id**’, **array(**‘MF921E01F-H11’,‘PE12D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**) ,** ‘**in**’ )  
Qdb::ver( ‘**email#id**’, **array(**‘MF921E01F-H11’,‘PE12D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**),**‘rafcrack@alice.it’**,** ‘**in**’ )

**==>>>** Verifica i 4 valori della KEY Primaria ( id ) ritorna un array contenente i Nuovi Valori che sono stati Inseriti nel database.

**6.** ( , del ) Qdb::ver( ‘**id**’, **array(**‘MF921E0F-H11’,‘PE12D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**) ,** ‘**del**’ )  
Qdb::ver( ‘**email#id**’, **array(**‘MF921E0F-H11’,‘PE12D07R-K11’,’DE122D041-Q11’**,**‘ AZ820E61F-I21’**),**‘rafcrack@alice.it’**,** ‘**del**’)

**==>>>** Verifica i 4 valori della KEY Primaria ( id ) ritorna un array contenente i Vecchi Valori che sono stati Cancellati dal database.

Case Sensitive --> MF921E01H e MF921e01H sono diversi