

Query SQL CUC per conteggi messaggi e dimensioni cassette postali

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Database](#)

[Tabelle](#)

[Query SQL](#)

[Elenca il numero totale di messaggi con un alias noto](#)

[Elenca utenti con numero totale messaggi](#)

[Elenca utenti con numero totale messaggi in base al primo carattere nell'alias](#)

[Elenca utenti con totale messaggi Posta in arrivo](#)

[Elenca utenti con totale messaggi eliminati](#)

[Elenca utenti con totale, Posta in arrivo e Messaggi eliminati](#)

[Elenca conteggio messaggi utente con tempo di arrivo messaggi meno recente](#)

[Elenca conteggio messaggi utente con tempo di arrivo messaggio meno recente e dimensioni della cassetta postale / durata totale](#)

[Elenca la cartella Posta in arrivo degli utenti e il numero di messaggi eliminati con l'ora di arrivo del messaggio meno recente e le dimensioni della cassetta postale / durata totale](#)

[Elenca il numero totale di messaggi per tutte le cassette postali](#)

[Elenca dimensioni cassetta postale utente con limiti di invio e ricezione](#)

[Elenca tutte le dimensioni della cassetta postale dell'utente con limiti di invio e ricezione](#)

[Elenca le dimensioni totali di tutte le cassette postali combinate](#)

Introduzione

In questo documento viene descritto come ottenere il numero di messaggi e le dimensioni di una cassetta postale utente con query SQL (Structured Query Language) tramite la CLI. I dati possono essere recuperati anche con lo strumento [User Data Dump](#) dalla pagina [Strumenti di Cisco Unified Communications](#).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza di Cisco Unity Connection (CUC).

Componenti usati

Le informazioni fornite in questo documento si basano sulle versioni 8.X e successive di CUC, ma possono essere utilizzate anche nelle versioni precedenti.

Database

Le query SQL vengono formate con i dati di questi database:

- **UnityDirDB** - Questo database contiene le informazioni relative all'utente.
- **UnityMbxDB1** - Questo database contiene le informazioni sulla cassetta postale dell'utente.

Tabelle

Le query SQL vengono formate con i dati di queste viste. Una *vista* è una tabella costituita da una combinazione di due o più tabelle oppure dagli stessi dati in una singola tabella.

Nel database di **UnityDirDB** vengono utilizzate le seguenti visualizzazioni:

- **vw_mailbox** - Questa vista contiene il mapping tra l'utente e la cassetta postale.
- **vw_user** - Questa vista contiene le informazioni relative all'utente.

Nel database **UnityMbxDB1** vengono utilizzate le seguenti visualizzazioni:

- **vw_message** - Questa vista contiene un messaggio nel sistema. Questa tabella è semplicemente un contenitore delle proprietà del messaggio.
- **vw_mailbox** - Questa vista contiene una cassetta postale del sistema che contiene i messaggi in ingresso. Questa tabella contiene informazioni generali sulla singola cassetta postale, per includere la modalità di archiviazione dei messaggi nella cassetta postale.

Query SQL

In questa sezione vengono descritte le varie query SQL che è possibile utilizzare in CUC.

Elenca il numero totale di messaggi con un alias noto

Immettere questo comando per ottenere un elenco del numero totale di messaggi con un alias noto:

```
admin:run cuc dbquery unitymbxdb1 select count (*) as Messages from vw_message,
```

```
unitydirdb:vw_mailbox, unitydirdb:vw_user where mailboxobjectid in (select mailboxid from vw_mailbox where unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid and alias='Anirudh')
```

```
messages
```

```
-----
```

```
3
```

Questa query è una query di database doppio ad alta complessità che interessa più tabelle. Per i server con database e caselle di posta di dimensioni molto grandi, potrebbe trascorrere un periodo di tempo prolungato prima che venga visualizzato un output, anche oltre un'ora, il che non è la soluzione ideale. In questi scenari è possibile utilizzare la query seguente:

```
admin:run cuc dbquery unitymbxdb1 select count (*) as Messages from vw_message where mailboxobjectid in (select mailboxobjectid from vw_mailbox where description='Anirudh')
```

```
messages
```

```
-----
```

```
3
```

La prima query restituisce i dati quando viene specificato un **alias**, ovvero **Univoco**. La seconda query restituisce i dati quando viene indicata la **descrizione**, che **NON** è **univoca**.

Nota: Al momento della creazione della cassetta postale, la descrizione corrisponde all'alias; tuttavia, quando l'alias viene aggiornato, la descrizione non viene aggiornata. Per i database di piccole dimensioni, la prima query è ideale. Per dimostrare che la descrizione non viene modificata dopo la modifica dell'alias, **test3** viene modificato in **Atest3** e utilizzato per le sezioni successive.

Elenca utenti con numero totale messaggi

Immettere questo comando per ottenere un elenco di utenti con il numero totale di messaggi:

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as messages from vw_message, unitydirdb:vw_mailbox, unitydirdb:vw_user where mailboxobjectid in (select mailboxid from vw_mailbox where unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) group by alias order by messages desc
```

```
userid messages
```

```
-----
```

```
Anirudh 3
```

```
Atest3 2
```

```
undeliverablemessagesmailbox 1
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as Messages from vw_message, vw_mailbox where vw_mailbox.mailboxobjectid = vw_message.mailboxobjectid group by description order by messages desc
```

```
description messages
```

```
-----
```

```
Anirudh 3
```

test3 2
undeliverablemessagesmailbox 1

Nota: Nella seconda query, la descrizione non cambia da **test3** ad **Atest3** dopo la modifica dell'alias.

Elenca utenti con numero totale messaggi in base al primo carattere nell'alias

Immettere questo comando per ottenere un elenco di utenti con il numero totale di messaggi basato sul primo carattere di un alias:

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as messages from  
vw_message, unitydirdb:vw_mailbox, unitydirdb:vw_user where deleted='0' and  
mailboxobjectid in (select mailboxid from vw_mailbox where unitydirdb:  
vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) and alias like 'A%' group by  
alias order by messages
```

```
userid messages  
-----  
Atest3 2  
Anirudh 3
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as Messages from  
vw_message, vw_mailbox where vw_mailbox.mailboxobjectid =  
vw_message.mailboxobjectid and description like 'A%' group by description order  
by messages
```

Di seguito sono riportate alcune note importanti su questa query:

- L'operatore **A%** imposta la query in modo che raggiunga gli alias che iniziano con la lettera A.
- Il formato è **dove nomecolonna è simile a 'condizione'**. Il nome della colonna è **alias** per la prima query e **descrizione** per la seconda query.

Di seguito sono riportati alcuni esempi di condizioni:

- **_n%** - La prima lettera può essere qualsiasi carattere (un carattere jolly), seguita dalla lettera **n** e da qualsiasi numero di caratteri.
- **%s** - Consente di impostare la query in modo che raggiunga gli alias che terminano con le lettere **s**.

Le query menzionate finora vengono utilizzate per ottenere il totale dei messaggi (posta in arrivo ed elementi eliminati). Nella sezione successiva vengono descritte le query utilizzate per ottenere il numero totale di messaggi nella cartella Posta in arrivo e di elementi eliminati.

Elenca utenti con totale messaggi Posta in arrivo

Immettere questo comando per ottenere un elenco di utenti con il totale dei messaggi della posta in arrivo:

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as
inboxmessages from vw_message, unitydirdb:vw_mailbox, unitydirdb:vw_user
where deleted='0' and mailboxobjectid in (select mailboxid from vw_mailbox
where unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) and
alias like 'A%' group by alias order by inboxmessages
```

```
userid inboxmessages
```

```
-----
Atest3 2
Anirudh 3
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as InboxMessages
from vw_message, vw_mailbox where vw_mailbox.mailboxobjectid =
vw_message.mailboxobjectid and deleted = '0' and description like 'A%' group by
description order by InboxMessages
```

Di seguito sono riportate alcune note importanti su questa query:

- L'operatore **A%** imposta la query in modo che raggiunga gli alias che iniziano con la lettera **A**.
- Il formato è **dove nomecolonna è simile a 'condizione'**. Il nome della colonna è **alias** per la prima query e **descrizione** per la seconda query.

Di seguito sono riportati alcuni esempi di condizioni:

- **_n%** - La prima lettera può essere qualsiasi carattere (un carattere jolly), seguita dalla lettera **n** e da qualsiasi numero di caratteri.
- **%s** - Consente di impostare la query in modo che raggiunga gli alias che terminano con le lettere **s**.

Nota: Nell'esempio, viene utilizzata una condizione per limitare gli utenti con un alias o una descrizione che inizia con la lettera **A**.

Di seguito sono riportate alcune varianti della query:

- Per elencare tutti gli utenti, rimuovere **e** applicare l'**alias 'A%'** nella prima query o **e la descrizione 'A%'** nella seconda query.
- Per elencare un utente specifico (ad esempio, elencare il conteggio della cassetta postale di Anirudh), sostituire **e** utilizzare **alias quali 'A%'** con **e alias='Anirudh'** nella prima query oppure sostituire **e descrivere come 'A%'** con **e descrizione ='Anirudh'** nella seconda query. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.

Elenca utenti con totale messaggi eliminati

Immettere questo comando per ottenere un elenco di utenti con il totale dei messaggi eliminati:

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as
deletedmessages from vw_message, unitydirdb:vw_mailbox, unitydirdb:vw_user
```

```
where deleted='1' and mailboxobjectid in (select mailboxid from vw_mailbox
where unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) and
alias like 'A%' group by alias order by deletedmessages
```

No records found

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as
deletedmessages from vw_message, vw_mailbox where vw_mailbox.mailboxobjectid=
vw_message.mailboxobjectid and deleted = '1' and description like 'A%' group
by description order by deletedmessages
```

Nota: In questo esempio non sono presenti messaggi eliminati, quindi l'output verrà visualizzato come **Nessun record trovato**.

Di seguito sono riportate alcune note importanti su questa query:

- L'operatore **A%** imposta la query in modo che raggiunga gli alias che iniziano con la lettera A.
- Il formato è **dove nomecolonna è simile a 'condizione'**. Il nome della colonna è **alias** per la prima query e **descrizione** per la seconda query.

Di seguito sono riportati alcuni esempi di condizioni:

- **_n%** - La prima lettera può essere qualsiasi carattere (un carattere jolly), seguita dalla lettera **n** e da qualsiasi numero di caratteri.
- **%s** - Consente di impostare la query in modo che raggiunga gli alias che terminano con le lettere **s**.

Nota: Nell'esempio, viene utilizzata una condizione per limitare gli utenti con un alias o una descrizione che inizia con la lettera **A**.

Di seguito sono riportate alcune varianti della query:

- Per elencare tutti gli utenti, rimuovere **e** applicare l'**alias 'A%'** nella prima query o **e la descrizione 'A%'** nella seconda query.
- Per elencare un utente specifico (ad esempio, elencare il conteggio della cassetta postale di Anirudh), sostituire **e** utilizzare **alias quali 'A%'** con **e alias='Anirudh'** nella prima query oppure sostituire **e descrivere come 'A%'** con **e descrizione ='Anirudh'** nella seconda query. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.

Elenca utenti con totale, Posta in arrivo e Messaggi eliminati

Immettere questo comando per ottenere un elenco di utenti con il totale, la cartella Posta in arrivo e i messaggi eliminati:

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as messages,
sum(case when deleted='0' then 1 else 0 end) as Inboxmessages, sum(case when
```

```
deleted='1' then 1 else 0 end) as Deletedmessages from vw_message, unitydirdb:vw_mailbox, unitydirdb:vw_user where mailboxobjectid in (select mailboxid from vw_mailbox where unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) group by alias order by messages desc
```

```
userid messages inboxmessages deletedmessages
```

```
-----  
Anirudh 3 3 0
```

```
Atest3 2 2 0
```

```
undeliverablemessagesmailbox 1 1 0
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description as UserID, count (*) as messages, sum(case when deleted='0' then 1 else 0 end) as Inboxmessages, sum (case when deleted='1' then 1 else 0 end) as Deletedmessages from vw_mailbox join vw_message on vw_message.mailboxobjectid=vw_mailbox.mailboxobjectid group by description order by messages desc
```

Di seguito sono riportate alcune note importanti su questa query:

- L'operatore **A%** imposta la query in modo che raggiunga gli alias che iniziano con la lettera A.
- Il formato è **dove nomecolonna è simile a 'condizione'**. Il nome della colonna è **alias** per la prima query e **descrizione** per la seconda query.

Di seguito sono riportati alcuni esempi di condizioni:

- **_n%** - La prima lettera può essere qualsiasi carattere (un carattere jolly), seguita dalla lettera **n** e da qualsiasi numero di caratteri.
- **%s** - Consente di impostare la query in modo che raggiunga gli alias che terminano con le lettere **s**.

Nota: Nell'esempio, viene utilizzata una condizione per limitare gli utenti con un alias o una descrizione che inizia con la lettera **A**.

Di seguito sono riportate alcune varianti della query:

- Per elencare tutti gli utenti, rimuovere **e** applicare l'**alias 'A%'** nella prima query o **e la descrizione 'A%'** nella seconda query.
- Per elencare un utente specifico (ad esempio, elencare il conteggio della cassetta postale di Anirudh), sostituire **e** utilizzare **alias quali 'A%'** con **e alias='Anirudh'** nella prima query oppure sostituire **e descrivere come 'A%'** con **e descrizione ='Anirudh'** nella seconda query. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.

Elenca conteggio messaggi utente con tempo di arrivo messaggi meno recente

Questa query può essere utilizzata per determinare se i processi pianificati per la pulizia delle cassette postali hanno effetto:

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as messages,
```

```
min(arrivaltime) as OldestMessageTime from vw_message, unitydirdb:vw_mailbox,
unitydirdb:vw_user where mailboxobjectid in (select mailboxid from vw_mailbox
where unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) group
by alias order by messages desc
```

```
userid messages oldestmessagetime
```

```
-----
Anirudh 3 2013-03-19 14:38:14.459
Atest3 2 2013-01-18 05:49:45.355
undeliverablemessagesmailbox 1 2012-07-05 01:10:19.961
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as Messages,
min(arrivaltime) as OldestMessageTime from vw_message, vw_mailbox where
vw_mailbox.mailboxobjectid = vw_message.mailboxobjectid group by description
order by messages desc
```

Nota: Per ottenere l'ora dell'ultimo arrivo per le query nelle sezioni precedenti, aggiungere `min(arrivaltime)` come `OldestMessageTime` subito dopo `count(*)` come `Messaggi`.

Di seguito sono riportate alcune varianti della query:

- Per elencare tutti gli utenti con alias che iniziano con la lettera A, aggiungere un alias come `'A%'` nella prima query immediatamente prima della condizione `group by alias` oppure una descrizione come `'A%'` nella seconda query immediatamente prima della condizione `group by description`. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.
- Per elencare un utente specifico (ad esempio, elencare il conteggio per la cassetta postale di Anirudh), aggiungere `e alias='Anirudh'` nella prima query immediatamente prima della condizione `group by alias` oppure `and description ='Anirudh'` nella seconda query immediatamente prima della condizione `group by description`. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.

Elenca conteggio messaggi utente con tempo di arrivo messaggio meno recente e dimensioni della cassetta postale / durata totale

Immettere questo comando per ottenere un elenco del numero di messaggi utente con l'ora di arrivo del messaggio meno recente e le dimensioni della cassetta postale (senza durata totale):

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as messages,
min(arrivaltime) as OldestMessageTime, vw_mailbox.bytesize from vw_message,
vw_mailbox, unitydirdb:vw_mailbox, unitydirdb:vw_user where
vw_message.mailboxobjectid=vw_mailbox.mailboxobjectid and
vw_mailbox.mailboxobjectid in (select mailboxid from vw_mailbox where
unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) group by
alias, vw_mailbox.bytesize order by messages desc
```

```
userid messages oldestmessagetime bytesize
```

```
-----
Anirudh 3 2013-03-19 14:38:14.459 93319
Atest3 2 2013-01-18 05:49:45.355 59890
```


Nota: per ottenere la durata totale dei messaggi: aggiungere ", **sum(duration/1000) as TotalDuration_In_sec**" prima di "**from_vw_message**". Non dimenticare la virgola prima di sum. Questa opzione può essere utilizzata anche per le query nelle sezioni precedenti.

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as Messages,
min(arrivaltime) as OldestMessageTime, vw_mailbox.bytesize from vw_message,
vw_mailbox where vw_mailbox.mailboxobjectid = vw_message.mailboxobjectid
group by description, vw_mailbox.bytesize order by messages desc
```

Nota: Per ottenere la durata totale dei messaggi: aggiungere ", **sum(duration/1000) as TotalDuration_In_sec**" prima di "**from_vw_message**". Non dimenticare la virgola prima di sum. Questa opzione può essere utilizzata anche per le query nelle sezioni precedenti.

Di seguito sono riportate alcune varianti della query:

- Per elencare tutti gli utenti con alias che iniziano con la lettera A, aggiungere un **alias come 'A%'** nella prima query immediatamente prima della condizione **group by alias** oppure una **descrizione come 'A%'** nella seconda query immediatamente prima della condizione **group by description**. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.
- Per elencare un utente specifico (ad esempio, elencare il conteggio per la cassetta postale di Anirudh), aggiungere **e alias='Anirudh'** nella prima query immediatamente prima della condizione **group by alias** oppure **and description ='Anirudh'** nella seconda query immediatamente prima della condizione **group by description**. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.

Elenca la cartella Posta in arrivo degli utenti e il numero di messaggi eliminati con l'ora di arrivo del messaggio meno recente e le dimensioni della cassetta postale / durata totale

Immettere questo comando per ottenere un elenco della posta in arrivo dell'utente e del numero di messaggi eliminati con l'ora di arrivo del messaggio meno recente e le dimensioni della cassetta postale (senza durata totale):

```
admin:run cuc dbquery unitymbxdb1 select alias as UserID, count (*) as
TotalMessages, sum(case when deleted='0' then 1 else 0 end) as Inbox,
sum(case when deleted='1' then 1 else 0 end) as Deleted, min
(arrivaltime) as OldestMessageTime, vw_mailbox.bytesize from vw_message,
vw_mailbox, unitydirdb:vw_mailbox, unitydirdb:vw_user where
vw_message.mailboxobjectid=vw_mailbox.mailboxobjectid and
vw_mailbox.mailboxobjectid in (select mailboxid from vw_mailbox where
unitydirdb:vw_user.objectid = unitydirdb:vw_mailbox.userobjectid) group
by alias, vw_mailbox.bytesize order by TotalMessages desc
```

```
userid total inbox deleted oldestmessagetime byte
```

messages size

```
-----  
Anirudh 3 3 0 2013-03-19 14:38:14.459 93319  
Atest3 2 2 0 2013-01-18 05:49:45.355 59890  
undeliverable 1 1 0 2012-07-05 01:10:19.961 317003  
messagesmailbox
```

Nota: Per ottenere la durata totale dei messaggi: aggiungere ", **sum(duration/1000) as TotalDuration_In_sec**" prima di "**from_vw_message**". Non dimenticare la virgola prima di sum. Questa opzione può essere utilizzata anche per le query nelle sezioni precedenti.

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, count (*) as  
TotalMessages, sum(case when deleted='0' then 1 else 0 end) as Inbox,  
sum(case when deleted='1' then 1 else 0 end) as Deleted, min(arrivaltime)  
as OldestMessageTime, vw_mailbox.bytesize from vw_message, vw_mailbox  
where vw_mailbox.mailboxobjectid = vw_message.mailboxobjectid group by  
description, vw_mailbox.bytesize order by TotalMessages desc
```

Nota: per ottenere la durata totale dei messaggi: aggiungere ", **sum(duration/1000) as TotalDuration_In_sec**" prima di "**from_vw_message**". Non dimenticare la virgola prima di sum. Questa opzione può essere utilizzata anche per le query nelle sezioni precedenti.

Di seguito sono riportate alcune varianti della query:

- Per elencare tutti gli utenti con alias che iniziano con la lettera A, aggiungere un **alias come 'A%'** nella prima query immediatamente prima della condizione **group by alias** oppure una **descrizione come 'A%'** nella seconda query immediatamente prima della condizione **group by description**. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.
- Per elencare un utente specifico (ad esempio, elencare il conteggio per la cassetta postale di Anirudh), aggiungere **e alias='Anirudh'** nella prima query immediatamente prima della condizione **group by alias** oppure **and description ='Anirudh'** nella seconda query immediatamente prima della condizione **group by description**. Verificare che la sostituzione si trovi nella posizione esatta oppure che la query non riesca.

Elenca il numero totale di messaggi per tutte le cassette postali

Immettere questo comando per ottenere il numero totale di messaggi per tutte le cassette postali combinate:

```
admin:run cuc dbquery unitymbxdb1 select count(*) as messages  
from vw_message
```

messages

```
-----  
6
```

Elenca dimensioni cassetta postale utente con limiti di invio e ricezione

Immettere questo comando per ottenere le dimensioni della cassetta postale dell'utente con limiti di invio e ricezione:

```
admin:run cuc dbquery unitydirdb select alias as UserID,bytesize,send,receive,
warning from vw_user,unitymbxdb1:vw_mailbox where vw_user.objectid in (select
userobjectid from vw_usermailboxmap where
vw_usermailboxmap.mailboxid=unitymbxdb1:vw_mailbox.mailboxobjectid and
alias='Anirudh')
```

```
userid bytesize send receive warning
-----
Anirudh 93319 13000000 14745600 12000000
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, bytesize,send,receive,
warning from vw_mailbox where description ='Anirudh'
```

Elenca tutte le dimensioni della cassetta postale dell'utente con limiti di invio e ricezione

Immettere questo comando per ottenere tutte le dimensioni della cassetta postale dell'utente con limiti di invio e ricezione:

```
admin:run cuc dbquery unitydirdb select alias as UserID,bytesize,send,receive,
warning from vw_user,unitymbxdb1:vw_mailbox where vw_user.objectid in (select
userobjectid from vw_usermailboxmap where
vw_usermailboxmap.mailboxid=unitymbxdb1:vw_mailbox.mailboxobjectid) order by
bytesize desc
```

```
userid bytesize send receive warning
-----
undeliverablemessagesmailbox 317003 13000000 14745600 12000000
Anirudh 93319 13000000 14745600 12000000
Atest3 59890 13000000 14745600 12000000
Solomon 0 13000000 14745600 12000000
UnityConnection 0 50000000 50000000 45000000
Suvir 0 13000000 14745600 12000000
dsas 0 13000000 14745600 12000000
test1 0 13000000 14745600 12000000
Atest2 0 13000000 14745600 12000000
operator 0 13000000 14745600 12000000
```

Per gli stessi motivi indicati nella sezione [Elenca il numero totale di messaggi con un alias noto](#), è possibile utilizzare anche questa query:

```
admin:run cuc dbquery unitymbxdb1 select description, bytesize,send,receive,
warning from vw_mailbox order by bytesize desc
```

Come variante di questa query per elencare tutti gli utenti con alias che iniziano con la lettera A, aggiungere un alias come 'A%' nella prima query subito dopo vw_usermailboxmap.mailboxid=unitymbxdb1:vw_mailbox.mailboxobjectid e prima) ordina per condizione oppure aggiungere una **descrizione where come 'A%'** nella seconda query

immediatamente prima della **condizione order by** . Verificare che l'aggiunta sia eseguita nella posizione corretta o che la query non abbia esito positivo.

Elenca le dimensioni totali di tutte le cassette postali combinate

Immettere questo comando per ottenere le dimensioni totali di tutte le cassette postali combinate:

```
admin:run cuc dbquery unitymbxdb1 select sum (bytesize) from vw_mailbox
```

```
(sum)
```

```
-----  
2683210
```

```
admin:
```