

# CUC SQL-vragen voor berichtellingen en postvakgrootte

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[databases](#)

[Tabellen](#)

[SQL-vragen](#)

[Lijst van het totaal aantal berichten met een bekende alias](#)

[Gebruikers lijst met totaal aantal berichten](#)

[Lijstgebruikers met totaal aantal berichten gebaseerd op eerste teken in alias](#)

[Gebruikers lijst met totaal inbox berichten](#)

[Gebruikers lijst met volledig verwijderd berichten](#)

[Gebruikers lijst met berichten met Totaal, Inbox en verwijderd](#)

[Aantal deelnemende gebruikers van een lijst met oudste berichttijd](#)

[Lijstberichten tellen met oudste berichtgrootte en postvakgrootte / totale duur](#)

[Lijstgebruiker in vak en verwijderd berichten tellen met oudste berichtgrootte en postvakgrootte / totale duur](#)

[Geef het totale aantal berichten voor alle postvakjes op](#)

[Lijst de grootte van een gebruikershandleiding met Verzend en ontvang limieten](#)

[Lijst alle gebruikersmailboxformaten met Verzend en ontvang limieten](#)

[Geef de totale grootte van alle postvakjes samen op](#)

## Inleiding

Dit document beschrijft hoe de berichtelling en de grootte van een gebruikershandleiding met SQL-vragen (Structured Search Query Language) via de CLI te verkrijgen. Deze gegevens kunnen ook worden opgehaald met het [User Data Dump](#)-gereedschap van de [Cisco Unified Communications Gereedschappen](#) pagina.

## Voorwaarden

### Vereisten

Cisco raadt u aan kennis te hebben van Cisco Unity Connection (CUC).

## Gebruikte componenten

De informatie in dit document is gebaseerd op CUC versies 8.X en later, maar deze informatie kan ook voor eerdere versies werken.

## databases

De SQL query wordt gevormd met de gegevens uit deze databases:

- **UnityDirDB** - Deze database bevat de gebruikersgerelateerde informatie.
- **UnityMBDB1** - Deze database bevat de informatie over de gebruikershandleiding.

## Tabellen

De SQL query wordt gevormd met de gegevens in deze beelden. Een *weergave* is een tabel die een combinatie is van twee of meer tabellen, of dezelfde gegevens in één tabel.

Deze standpunten worden gebruikt in de database van **UnityDirDB**:

- **vw\_box** - Deze weergave bevat de mapping tussen de gebruiker en de postbus.
- **vw\_user** - Deze weergave bevat de gebruikersgerelateerde informatie.

Deze standpunten worden gebruikt in de **UnityMbxDB1** database:

- **vw\_message** - Deze weergave bevat een bericht item in het systeem. Deze tabel heeft alleen eigenschappen van een bericht.
- **vw\_box** - Deze weergave bevat een postvak op het systeem dat inkomende berichten bevat. Deze tabel bevat algemene informatie over de afzonderlijke postvak, om de manier te vermelden waarop berichten in de postvak worden opgeslagen.

## SQL-vragen

In deze sectie worden de verschillende SQL-vragen beschreven die u in CUC kunt gebruiken.

### Lijst van het totaal aantal berichten met een bekende alias

Voer deze opdracht in om een lijst te verkrijgen van de totale berichten tellen bij een bekend alias:

```
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

Deze query is een zeer complex, dubbel-database query dat meerdere tabellen omvat. Voor servers met een zeer grote database- en postvakgrootte kan een langere periode verlopen voordat er een uitvoer verschijnt, zelfs over een uur, wat niet ideaal is. In zulke scenario's kunt u deze query gebruiken in plaats daarvan:

```
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

De eerste query geeft de gegevens terug als **alias** wordt vermeld, wat **uniek** is. De tweede query geeft de gegevens terug als de **beschrijving** wordt vermeld, wat **NIET uniek** is.

**Opmerking:** Wanneer de brievenbus wordt gecreëerd, is de beschrijving dezelfde als de alias; wanneer het alias echter wordt bijgewerkt, is de beschrijving niet bijgewerkt. Voor kleine databases is de eerste query ideaal. Om aan te tonen dat de beschrijving niet wordt gewijzigd nadat de alias is gewijzigd, wordt **test3** gewijzigd in **Atest3** en gebruikt voor de volgende rubrieken.

## Gebruikers lijst met totaal aantal berichten

Voer deze opdracht in om een lijst met gebruikers te krijgen met de totale aantal berichten:

```
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

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

```
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

**Opmerking:** In de tweede query verandert de beschrijving niet van **test3** naar **Atest3** nadat

het alias is veranderd.

## Lijstgebruikers met totaal aantal berichten gebaseerd op eerste teken in alias

Voer deze opdracht in om een lijst van gebruikers te krijgen met de totale berichtstelling gebaseerd op het eerste teken van een 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
```

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

```
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
```

Hier volgen een paar belangrijke opmerkingen over deze vraag:

- The **A%** stelt de query in om aliases te starten die beginnen met de letter A.
- Het formaat is **waar columnname als 'conditie'**. Hier is de kolom naam **alias** voor de eerste query en **beschrijving** voor de tweede query.

Hier zijn een paar voorbeelden:

- **\_n%** - De eerste letter kan elk teken zijn (een jokerteken), gevolgd door de letter **n** en elk aantal tekens.
- **%s** - Dit stelt de query in om aliases te starten die eindigen met de letter **s**.

De vragen die tot nu toe zijn vermeld, worden gebruikt om de totale berichten te verkrijgen (in vak en verwijderde items). In de volgende sectie worden vragen beschreven die worden gebruikt om het totale aantal berichten in het inbox en de verwijderde punten te verkrijgen.

## Gebruikers lijst met totaal inbox berichten

Typ deze opdracht om een lijst met gebruikers te krijgen met de totale inbox-berichten:

```
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

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

```
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
```

Hier volgen een paar belangrijke opmerkingen over deze vraag:

- The **A%** stelt de query in om aliases te starten die beginnen met de letter A.
- Het formaat is **waar columnname als 'conditie'**. Hier is de kolom naam **alias** voor de eerste query en **beschrijving** voor de tweede query.

Hier zijn een paar voorbeelden:

- **\_n%** - De eerste letter kan elk teken zijn (een jokerteken), gevolgd door de letter **n** en elk aantal tekens.
- **%s** - Dit stelt de query in om aliases te starten die eindigen met de letter **s**.

**Opmerking:** In dit voorbeeld wordt een voorwaarde gebruikt om gebruikers te beperken met een alias/beschrijving die begint met de letter **A**.

Dit zijn een paar variaties van deze query:

- Om een lijst te maken van alle gebruikers, verwijder **en alias als 'A%'** in de eerste query, of **en beschrijving als 'A%'** in de tweede query.
- Om een bepaalde gebruiker op te nemen (lijst van de aantallen voor de brievenbus van Anirudh, bijvoorbeeld), vervang **en alias zoals "A%"** door **en alias="Anirudh"** in de eerste query, of vervang **en beschrijving als "A%"** door **en beschrijving = "Anirudh"** in de tweede query. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.

## Gebruikers lijst met volledig verwijderd berichten

Typ deze opdracht om een lijst met gebruikers te krijgen met de totale verwijderd berichten:

```
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

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende](#)

sectie [Alias](#), kan deze query ook worden gebruikt:

```
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
```

**Opmerking:** In dit voorbeeld zijn er geen verwijderde berichten, dus de uitvoer verschijnt als geen records gevonden.

Hier volgen een paar belangrijke opmerkingen over deze vraag:

- The **A%** stelt de query in om aliases te starten die beginnen met de letter A.
- Het formaat is **waar columnname als 'conditie'**. Hier is de kolom naam **alias** voor de eerste query en **beschrijving** voor de tweede query.

Hier zijn een paar voorbeelden:

- **\_n%** - De eerste letter kan elk teken zijn (een jokerteken), gevolgd door de letter **n** en elk aantal tekens.
- **%s** - Dit stelt de query in om aliases te starten die eindigen met de letter **s**.

**Opmerking:** In dit voorbeeld wordt een voorwaarde gebruikt om gebruikers te beperken met een alias/beschrijving die begint met de letter **A**.

Dit zijn een paar variaties van deze query:

- Om een lijst te maken van alle gebruikers, verwijder **en alias als 'A%'** in de eerste query, of **en beschrijving als 'A%'** in de tweede query.
- Om een bepaalde gebruiker op te nemen (lijst van de aantallen voor de brievenbus van Anirudh, bijvoorbeeld), vervang **en alias zoals "A%"** door **en alias="Anirudh"** in de eerste query, of vervang **en beschrijving als "A%"** door **en beschrijving = "Anirudh"** in de tweede query. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.

## Gebruikers lijst met berichten met Totaal, Inbox en verwijderd

Typ deze opdracht om een lijst met gebruikers te krijgen met de totaal-, inbox- en verwijderde berichten:

```
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

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

```
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
```

Hier volgen een paar belangrijke opmerkingen over deze vraag:

- The **A%** stelt de query in om aliases te starten die beginnen met de letter A.
- Het formaat is **waar columnname als 'conditie'**. Hier is de kolom naam **alias** voor de eerste query en **beschrijving** voor de tweede query.

Hier zijn een paar voorbeelden:

- **\_n%** - De eerste letter kan elk teken zijn (een jokerteken), gevolgd door de letter **n** en elk aantal tekens.
- **%s** - Dit stelt de query in om aliases te starten die eindigen met de letter **s**.

**Opmerking:** In dit voorbeeld wordt een voorwaarde gebruikt om gebruikers te beperken met een alias/beschrijving die begint met de letter **A**.

Dit zijn een paar variaties van deze query:

- Om een lijst te maken van alle gebruikers, verwijder **en alias als 'A%'** in de eerste query, of **en beschrijving als 'A%'** in de tweede query.
- Om een bepaalde gebruiker op te nemen (lijst van de aantallen voor de brievenbus van Anirudh, bijvoorbeeld), vervang **en alias zoals "A%"** door **en alias="Anirudh"** in de eerste query, of vervang **en beschrijving als "A%"** door **en beschrijving = "Anirudh"** in de tweede query. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.

## Aantal deelnemende gebruikers van een lijst met oudste berichtijd

Deze query kan worden gebruikt om te bepalen of de geplande banen voor de mailboxe schoonmaak effect hebben:

```
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
```

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

```
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
```

**Opmerking:** Om de laatste aankomsttijd voor de vragen in de vorige secties te verkrijgen, voeg **min (aankomsttijd) als OldestMessageTime** net na **count(\*)** toe als berichten.

Dit zijn een paar variaties van deze query:

- Om alle gebruikers te kunnen opsommen met aliases die beginnen met de letter A, **en alias zoals "A%"** in de eerste query net vóór de **groep door alias** conditie, of **en beschrijving zoals "A%"** in de tweede query net vóór de **groep door beschrijving** conditie. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.
- Om een bepaalde gebruiker te kunnen opsommen (geef bijvoorbeeld een lijst van de aantallen bij Anirudh's brievenbus), voeg **en alias='Anirudh'** toe in de eerste query vlak vóór de **groep door alias-voorwaarde**, of **en beschrijving ='Anirudh'** in de tweede query vlak vóór de **groep door beschrijving** voorwaarde. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.

## Lijstberichten tellen met oudste berichtgrootte en postvakgrootte / totale duur

Voer deze opdracht in om een lijst te krijgen van de tellen van het gebruikersbericht met de aankomsttijd van het oudste bericht en de postvakgrootte (zonder Totale Duur):

```
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
undeliverablemessagesmailbox 1 2012-07-05 01:10:19.961 317003
```

**Opmerking:** U krijgt de totale duur van de berichten als volgt: voeg **", sum (durf/1000) toe als TotalDuration\_In\_sec"** net vóór **"van vw\_message"**. Vergeet de komma niet vóór som. Dit kan ook worden gebruikt voor de vragen in de vorige afdelingen.

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:



```
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
```

**Opmerking:** Zo krijgt u de totale duur van de berichten: voeg ", sum (durf/1000) toe als **TotalDuration\_In\_sec**" net vóór "van vw\_message". Vergeet de komma niet vóór som. Dit kan ook worden gebruikt voor de vragen in de vorige afdelingen.

Dit zijn een paar variaties van deze query:

- Om alle gebruikers te kunnen opsommen met aliases die beginnen met de letter A, en alias zoals "A%" in de eerste query net vóór de **groep door alias** conditie, of en **beschrijving** zoals "A%" in de tweede query net vóór de **groep door beschrijving** conditie. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.
- Om een bepaalde gebruiker te kunnen opsommen (geef bijvoorbeeld een lijst van de aantallen bij Anirudh's brievenbus), voeg en **alias='Anirudh'** toe in de eerste query vlak vóór de **groep door alias-voorwaarde**, of en **beschrijving ='Anirudh'** in de tweede query vlak vóór de **groep door beschrijving** voorwaarde. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.

## Lijstgebruiker in vak en verwijderd berichten tellen met oudste berichtgrootte en postvakgrootte / totale duur

Voer deze opdracht in om een lijst van de gebruiker in-box te verkrijgen en de verwijderde berichten tellen met de aankomsttijd van het oudste bericht en de postvakgrootte (zonder Totale Duur):

```
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
```

**Opmerking:** Zo krijgt u de totale duur van de berichten: voeg ", sum (durf/1000) toe als **TotalDuration\_In\_sec**" net vóór "van vw\_message". Vergeet de komma niet vóór som. Dit kan ook worden gebruikt voor de vragen in de vorige afdelingen.

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

```
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
```

**Opmerking:** u krijgt de totale duur van de berichten als volgt: voeg ", sum (duur/1000) toe als TotalDuration\_In\_sec" net vóór "van vw\_message". Vergeet de komma niet vóór som. Dit kan ook worden gebruikt voor de vragen in de vorige afdelingen.

Dit zijn een paar variaties van deze query:

- Om alle gebruikers te kunnen opsommen met aliases die beginnen met de letter A, en alias zoals "A%" in de eerste query net vóór de groep door alias conditie, of en beschrijving zoals "A%" in de tweede query net vóór de groep door beschrijving conditie. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.
- Om een bepaalde gebruiker te kunnen opsommen (geef bijvoorbeeld een lijst van de aantallen bij Anirudh's brievenbus), voeg en alias='Anirudh' toe in de eerste query vlak vóór de groep door alias-voorwaarde, of en beschrijving ='Anirudh' in de tweede query vlak vóór de groep door beschrijving voorwaarde. Zorg ervoor dat de vervanging in de exacte positie staat, of de query faalt.

## Geef het totale aantal berichten voor alle postvakjes op

Typ deze opdracht om het totale aantal berichten voor alle gecombineerde postvakjes te verkrijgen:

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

```
messages
-----
6
```

## Lijst de grootte van een gebruikershandleiding met Verzend en ontvang limieten

Voer deze opdracht in om de grootte van de gebruikershandleiding te verkrijgen met verzendings- en ontvangstbeperingen:

```
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

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

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

## Lijst alle gebruikersmailboxformaten met Verzend en ontvang limieten

Voer deze opdracht in om alle gebruikerspostvakgrootte te verkrijgen met verzenden en ontvangen beperkingen:

```
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
```

Om de redenen die in de [Lijst](#) worden genoemd [de Totale Berichten Aantal met een bekende sectie Alias](#), kan deze query ook worden gebruikt:

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

Als een variatie van deze query om alle gebruikers op te sommen met aliassen die beginnen met de letter A, en alias zoals 'A%' in de eerste query net na de `vw_usermailboxmap.mailboxid=unitymbxdb1:vw_box.mailobject` en vóór `) order` door voorwaarde toevoegen, of je kunt waar toevoegen Beschrijving zoals "A%" in de tweede query vlak vóór de `order` door voorwaarde. Zorg ervoor dat dit wordt toegevoegd in de juiste positie, of de query faalt.

## Geef de totale grootte van alle postvakjes samen op

Typ deze opdracht om de totale grootte van alle gecombineerde postvakjes te verkrijgen:

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

```
(sum)
-----
2683210
```

```
admin:
```