

# Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Bases de datos](#)

[Tablas](#)

[Consultas SQL](#)

[Enumere la cuenta total de los mensajes con sabido alias](#)

[Enumere a los usuarios con la cuenta total de los mensajes](#)

[Enumere a los usuarios con los mensajes totales cuentan basado en el primer carácter en alias](#)

[Enumere a los usuarios con los mensajes de entrada totales](#)

[Enumere a los usuarios con los mensajes borrados totales](#)

[Enumere a los usuarios con el total, el inbox, y los mensajes borrados](#)

[Enumere el recuento de mensajes del usuario con la hora de llegada del más viejo mensaje](#)

[Enumere el recuento de mensajes del usuario con la hora de llegada del más viejo mensaje y el tamaño del buzón/la duración del total](#)

[Enumere el inbox del usuario y los mensajes borrados cuentan con la hora de llegada del más viejo mensaje y el tamaño del buzón/la duración del total](#)

[Enumere el número total de mensajes para todos los buzones](#)

[Enumere un tamaño de la casilla de correo del usuario con envían y reciben los límites](#)

[Enumere todos los tamaños de la casilla de correo del usuario con envían y reciben los límites](#)

[Enumere el tamaño total de todos los buzones combinados](#)

## Introducción

Este documento describe cómo obtener el recuento de mensajes y el tamaño de una casilla de correo del usuario con las interrogaciones del Lenguaje de consulta estructurado (SQL) vía el CLI. Estos datos se pueden también extraer con la herramienta del [volcado de los datos del usuario](#), de la página de las [herramientas de las Comunicaciones unificadas de Cisco](#).

## Prerrequisitos

### Requisitos

Cisco recomienda que usted tiene conocimiento del Cisco Unity Connection (CUC).

## Componentes Utilizados

La información en este documento se basa en las versiones 8.X CUC y posterior, pero esta información pudo trabajar para las versiones anteriores también.

## Bases de datos

Las consultas SQL se forman con los datos de estas bases de datos:

- **UnityDirDB** - Esta base de datos contiene la información usuario-relacionada.
- **UnityMbxDB1** - Esta base de datos contiene la información de la casilla de correo del usuario.

## Tablas

Las consultas SQL se forman con los datos en estas opiniones. *Una visión* es una tabla que es una combinación de dos o más tablas, o los mismos datos en una sola tabla.

Estas opiniones se utilizan en la base de datos de **UnityDirDB**:

- **vw\_mailbox** - Esta visión contiene la asignación entre el usuario y el buzón.
- **vw\_user** - Esta visión contiene la información usuario-relacionada.

Estas opiniones se utilizan en la base de datos **UnityMbxDB1**:

- **vw\_message** - Esta visión contiene un elemento del mensaje en el sistema. Esta tabla es simplemente un tenedor de las propiedades del mensaje.
- **vw\_mailbox** - Esta visión contiene un buzón en el sistema ese los mensajes entrantes de los controles. Esta tabla contiene la información general sobre la casilla de correo individual, para incluir la manera en la cual los mensajes se salvan en el buzón.

## Consultas SQL

Esta sección describe las diversas consultas SQL que usted puede utilizar en CUC.

### Enumere los mensajes totales cuentan con sabido alias

Ingrese este comando para obtener una lista de los mensajes totales cuentan con sabido 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
```

Esta interrogación es una complejidad alta, la interrogación de la doble-base de datos que implica

las varias tablas. Para los servidores con una base de datos y un tamaño muy grandes del buzón, un período ampliado pudo transcurrir antes de que una salida aparezca, incluso durante una hora, que no es ideal. En tales escenarios, usted puede utilizar esta interrogación en lugar de otro:

```
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 primera interrogación devuelve los datos cuando se menciona el **alias**, que es **único**. La segunda interrogación devuelve los datos cuando se menciona la **descripción**, que no es **única**.

Nota: Cuando se crea el buzón, la descripción es lo mismo que el alias; sin embargo, cuando el alias es actualizado, la descripción no es actualizada. Para las pequeñas bases de datos, la primera interrogación es ideal. Para demostrar que la descripción no está cambiada después de que se altere el alias, **test3** se modifica a **Atest3** y se utiliza para las siguientes secciones.

## Usuarios de la lista con la cuenta total de los mensajes

Ingrese este comando para obtener una lista de usuarios con la cuenta total de los mensajes:

```
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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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: En la segunda interrogación, la descripción no cambia de **test3** a **Atest3** después de que se cambie el alias.

## Los usuarios de la lista con los mensajes totales cuentan basado en el primer carácter en alias

Ingrese este comando para obtener una lista de usuarios con los mensajes totales cuentan basado en el primer carácter de 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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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
```

Aquí están algunas NOTAS IMPORTANTES sobre esta interrogación:

- El **A%** fija la interrogación para golpear los alias que comienzan con la letra **A**.
- El formato es **donde el columnname le gusta la "condición"**. Aquí, el nombre de columna está **alias** para la primera interrogación y **descripción** para la segunda interrogación.

Aquí están algunas condiciones del ejemplo:

- el **\_n%** - La primera carta puede ser cualquier carácter (comodín), seguido por la letra **n** y cualquier número de caracteres.
- **%s** - Esto fija la interrogación para golpear los alias que terminan con la letra **S**.

Las interrogaciones que se mencionan hasta el momento se utilizan para obtener los mensajes totales (inbox y los elementos eliminados). La siguiente sección describe las interrogaciones que se utilizan para obtener el número total de mensajes en el inbox y los elementos eliminados.

## Usuarios de la lista con los mensajes de entrada totales

Ingrese este comando para obtener una lista de usuarios con los mensajes de entrada totales:

```
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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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
```

Aquí están algunas NOTAS IMPORTANTES sobre esta interrogación:

- El **A%** fija la interrogación para golpear los alias que comienzan con la letra **A**.
- El formato es **donde el columnname le gusta la “condición”**. Aquí, el nombre de columna está **alias** para la primera interrogación y **descripción** para la segunda interrogación.

Aquí están algunas condiciones del ejemplo:

- **el \_n%** - La primera carta puede ser cualquier carácter (comodín), seguido por la letra **n** y cualquier número de caracteres.
- **%s** - Esto fija la interrogación para golpear los alias que terminan con la letra **S**.

Nota: En este ejemplo, una condición se utiliza para limitar a los usuarios con un alias/una descripción que comience con la letra **A**.

Éstas son algunas variaciones de esta interrogación:

- Para enumerar a todos los usuarios, quite **y alias como el “A%”** en la primera interrogación, o **y la descripción como el “A%”** en la segunda interrogación.
- Para enumerar a un usuario determinado (enumere la cuenta para el buzón de Anirudh, por ejemplo), substituirlo **y alias como el “A%”** por **y el alias='Anirudh** en la primera interrogación, o substituirlo **y la descripción como el “A%”** por **y la descripción = ' Anirudh** en la segunda interrogación. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

## Enumere a los usuarios con los mensajes borrados totales

Ingrese este comando para obtener una lista de usuarios con los mensajes borrados totales:

```
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

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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: En este ejemplo no hay mensajes borrados, así que la salida aparece pues **ningunos expedientes encontraron**.

Aquí están algunas NOTAS IMPORTANTES sobre esta interrogación:

- El **A%** fija la interrogación para golpear los alias que comienzan con la letra A.
- El formato es **donde el columnname le gusta la “condición”**. Aquí, el nombre de columna está **alias** para la primera interrogación y **descripción** para la segunda interrogación.

Aquí están algunas condiciones del ejemplo:

- el **\_n%** - La primera carta puede ser cualquier carácter (comodín), seguido por la letra n y cualquier número de caracteres.
- **%s** - Esto fija la interrogación para golpear los alias que terminan con la letra S.

Nota: En este ejemplo, una condición se utiliza para limitar a los usuarios con un alias/una descripción que comience con la letra **A**.

Éstas son algunas variaciones de esta interrogación:

- Para enumerar a todos los usuarios, quite **y alias como el “A%”** en la primera interrogación, o **y la descripción como el “A%”** en la segunda interrogación.
- Para enumerar a un usuario determinado (enumere la cuenta para el buzón de Anirudh, por ejemplo), sustituirlo **y alias como el “A%”** por **y el alias='Anirudh'** en la primera interrogación, o sustituirlo **y la descripción como el “A%”** por **y la descripción = 'Anirudh'** en la segunda interrogación. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

## Enumere a los usuarios con el total, el inbox, y los mensajes borrados

Ingrese este comando para obtener una lista de usuarios con el total, el inbox, y los mensajes borrados:

```
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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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
```

Aquí están algunas NOTAS IMPORTANTES sobre esta interrogación:

- El **A%** fija la interrogación para golpear los alias que comienzan con la letra **A**.
- El formato es **donde el columnname le gusta la “condición”**. Aquí, el nombre de columna está **alias** para la primera interrogación y **descripción** para la segunda interrogación.

Aquí están algunas condiciones del ejemplo:

- **el \_n%** - La primera carta puede ser cualquier carácter (comodín), seguido por la letra **n** y cualquier número de caracteres.
- **%s** - Esto fija la interrogación para golpear los alias que terminan con la letra **S**.

Nota: En este ejemplo, una condición se utiliza para limitar a los usuarios con un alias/una descripción que comience con la letra **A**.

Éstas son algunas variaciones de esta interrogación:

- Para enumerar a todos los usuarios, quite **y alias como el “A%”** en la primera interrogación, o **y la descripción como el “A%”** en la segunda interrogación.
- Para enumerar a un usuario determinado (enumere la cuenta para el buzón de Anirudh, por ejemplo), sustituirlo **y alias como el “A%”** por **y el alias='Anirudh** en la primera interrogación, o sustituirlo **y la descripción como el “A%”** por **y la descripción = ' Anirudh** en la segunda interrogación. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

## Enumere el recuento de mensajes del usuario con la hora de llegada del más viejo mensaje

Esta interrogación se puede utilizar para determinar si los trabajos programados para la limpieza del mailboxe toman el efecto:

```
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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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: Para obtener la hora de llegada más pasada para las interrogaciones en las secciones anteriores, agregue el **min(arrivaltime) como OldestMessageTime** enseguida después de la **cuenta (\*) como mensajes**.

Éstas son algunas variaciones de esta interrogación que comienzan con la letra **A**, agregue **y**

alias como el "A%" en la primera interrogación momentos antes del grupo por la condición del alias, o y de la descripción como el "A%" en la segunda interrogación momentos antes del grupo por la condición de la descripción. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

- Para enumerar a un usuario determinado (enumere la cuenta para el buzón de Anirudh, por ejemplo), agregue y alias='Anirudh en la primera interrogación momentos antes del grupo por la condición del alias, o y la descripción = ' Anirudh en la segunda interrogación momentos antes del grupo por la condición de la descripción. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

## Enumere el recuento de mensajes del usuario con la hora de llegada del más viejo mensaje y el tamaño del buzón/la duración del total

Ingrese este comando para obtener una lista del recuento de mensajes del usuario con la hora de llegada del más viejo mensaje y el tamaño del buzón (sin la duración total):

```
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
```

Nota: , sum(duration/1000) como TotalDuration\_In\_sec" momentos antes de "del vw\_message". No olvide la coma antes de la suma. Esto se puede también utilizar para las interrogaciones en las secciones anteriores.

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

```
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: En la orden para conseguir la duración total de los mensajes: agregue " , sum(duration/1000) como TotalDuration\_In\_sec" momentos antes de "del vw\_message". No olvide la coma antes de la suma. Esto se puede también utilizar para las interrogaciones en las secciones anteriores.

Éstas son algunas variaciones de esta interrogación:

- Para enumerar a todos los usuarios con los alias que comienzan con la letra A, agregue y alias como el "A%" en la primera interrogación momentos antes del grupo por la condición del alias, o y de la descripción como el "A%" en la segunda interrogación momentos antes del



**grupo por la** condición de la **descripción**. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

- Para enumerar a un usuario determinado (enumere la cuenta para el buzón de Anirudh, por ejemplo), agregue **y alias='Anirudh'** en la primera interrogación momentos antes del **grupo por la** condición del **alias**, o **y la descripción = ' Anirudh'** en la segunda interrogación momentos antes del **grupo por la** condición de la **descripción**. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

## Enumere el inbox del usuario y los mensajes borrados cuentan con la hora de llegada del más viejo mensaje y el tamaño del buzón/la duración del total

Ingrese este comando para obtener una lista del inbox del usuario y los mensajes borrados cuentan con la hora de llegada del más viejo mensaje y el tamaño del buzón

```
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: Para , **sum(duration/1000) como TotalDuration\_In\_sec** momentos antes de “del **vw\_message**”. No olvide la coma antes de la suma. Esto se puede también utilizar para las interrogaciones en las secciones anteriores.

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de \*\*alias sabida\*\*](#), esta interrogación puede también ser utilizada:

```
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: Para , **sum(duration/1000) como TotalDuration\_In\_sec** momentos antes de “del **vw\_message**”. No olvide la coma antes de la suma. Esto se puede también utilizar para las interrogaciones en las secciones anteriores.

Éstas son algunas variaciones de esta interrogación:

- Para enumerar a todos los usuarios con los alias que comienzan con la letra A, agregue **y alias como el "A%"** en la primera interrogación momentos antes del **grupo por la** condición del

**alias**, o **y de la descripción como el "A%"** en la segunda interrogación momentos antes del **grupo por** la condición de la **descripción**. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

- Para enumerar a un usuario determinado (enumere la cuenta para el buzón de Anirudh, por ejemplo), agregue **y alias='Anirudh'** en la primera interrogación momentos antes del **grupo por** la condición del **alias**, o **y la descripción = ' Anirudh'** en la segunda interrogación momentos antes del **grupo por** la condición de la **descripción**. Asegúrese de que el reemplazo esté en la posición exacta, o la interrogación falla.

## Enumere el número total de mensajes para todos los buzones

Ingrese este comando para obtener el número total de mensajes para todos los buzones combinados:

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

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

## Enumere un tamaño de la casilla de correo del usuario con envían y reciben los límites

Ingrese este comando para obtener el tamaño de la casilla de correo del usuario con envían y reciben los límites:

```
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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de \*\*alias sabida\*\*](#), esta interrogación puede también ser utilizada:

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

## Enumere todos los tamaños de la casilla de correo del usuario con envían y reciben los límites

Ingrese este comando para obtener todo el los tamaños de la casilla de correo del usuario con envían y reciben los límites:

```
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
```

Por las mismas razones que se mencionan en la [lista que los mensajes totales cuentan con una sección de alias sabida](#), esta interrogación puede también ser utilizada:

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

Como variación de esta interrogación para enumerar a todos los usuarios con los alias que comienzan con la letra **A**, agregue **y alias como el "A%"** en la primera interrogación enseguida después del **vw\_usermailboxmap.mailboxid=unitymbxdb1:vw\_mailbox.mailboxobjectid** y antes) **orden por la** condición, o usted puede agregar **donde descripción como el "A%"** en la segunda interrogación momentos antes de la **orden por la** condición. Asegúrese de que esto esté agregada en la posición correcta, o la interrogación falla.

## Enumere el tamaño total de todos los buzones combinados

Ingrese este comando para obtener el tamaño total de todos los buzones combinados:

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

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

```
admin:
```