

CUC SQL-Abfragen für Nachrichtenanzahl und Mailbox-Größe

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Datenbanken](#)

[Tabellen](#)

[SQL-Abfragen](#)

[Auflisten der Gesamtzahl der Nachrichten mit einer bekannten Alias](#)

[Benutzer mit Gesamtzahl der Nachrichten auflisten](#)

[Liste von Benutzern mit der Gesamtzahl der Nachrichten basierend auf dem ersten Zeichen in Alias](#)

[Benutzer mit Gesamtzahl der Posteingänge auflisten](#)

[Liste von Benutzern mit insgesamt gelöschten Nachrichten](#)

[Benutzer mit Gesamtzahl, Posteingang und gelöschten Nachrichten auflisten](#)

[Liste der Anzahl der Benutzermeldungen mit der ältesten Ankunftszeit der Nachricht](#)

[Anzahl der Benutzermeldungen mit der ältesten Ankunftszeit und Größe des Postfachs anzeigen/Gesamtdauer der Nachricht](#)

[Liste der Benutzer-Posteingänge und der Anzahl der gelöschten Nachrichten mit der ältesten Ankunftszeit und der Größe des Posteingangs/Gesamtdauer der Nachricht](#)

[Geben Sie die Gesamtzahl der Nachrichten für alle Mailboxen an.](#)

[Mailbox-Größe für Benutzer mit Sende- und Empfangsbeschränkungen auflisten](#)

[Listen Sie alle Mailbox-Größen für Benutzer mit Sende- und Empfangsbeschränkungen auf.](#)

[Geben Sie die Gesamtgröße aller kombinierten Mailboxen an.](#)

Einführung

In diesem Dokument wird beschrieben, wie die Nachrichtenanzahl und die Größe einer Benutzermailbox mit SQL-Abfragen (Structured Query Language) über die CLI abgerufen werden. Diese Daten können auch mit dem [User Data Dump](#)-Tool von der [Cisco Unified Communications-Tools](#)-Seite abgerufen werden.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Cisco Unity Connection (CUC) verfügen.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf CUC-Versionen 8.x und höher, können aber auch für ältere Versionen verwendet werden.

Datenbanken

Die SQL-Abfragen werden mit den Daten aus diesen Datenbanken gebildet:

- **UnityDirDB** - Diese Datenbank enthält die benutzerbezogenen Informationen.
- **UnityMbxDB1** - Diese Datenbank enthält die Informationen für die Benutzermailbox.

Tabellen

Die SQL-Abfragen werden mit den Daten in diesen Ansichten gebildet. Eine *Ansicht* ist eine Tabelle, die entweder eine Kombination aus zwei oder mehr Tabellen oder die gleichen Daten in einer einzigen Tabelle darstellt.

Diese Ansichten werden in der **UnityDirDB**-Datenbank verwendet:

- **vw_mailbox** - Diese Ansicht enthält die Zuordnung zwischen Benutzer und Mailbox.
- **vw_user**: Diese Ansicht enthält die benutzerbezogenen Informationen.

Diese Ansichten werden in der **UnityMbxDB1**-Datenbank verwendet:

- **vw_message**: Diese Ansicht enthält ein Nachrichtenelement im System. Diese Tabelle ist lediglich ein Inhaber von Nachrichteneigenschaften.
- **vw_mailbox** - Diese Ansicht enthält eine Mailbox im System, die eingehende Nachrichten enthält. Diese Tabelle enthält allgemeine Informationen über die einzelnen Mailboxen, einschließlich der Art und Weise, wie Nachrichten in der Mailbox gespeichert werden.

SQL-Abfragen

In diesem Abschnitt werden die verschiedenen SQL-Abfragen beschrieben, die Sie in CUC verwenden können.

Auflisten der Gesamtzahl der Nachrichten mit einer bekannten Alias

Geben Sie diesen Befehl ein, um eine Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias zu erhalten:

```
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

Diese Abfrage ist eine komplexe Doppeldatenbankabfrage, die mehrere Tabellen umfasst. Bei Servern mit einer sehr großen Datenbank- und Mailbox-Größe kann es vorkommen, dass ein längerer Zeitraum verstrichen ist, bevor eine Ausgabe angezeigt wird, sogar über eine Stunde, was nicht ideal ist. In solchen Szenarien können Sie stattdessen folgende Abfrage verwenden:

```
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

Die erste Abfrage gibt die Daten zurück, wenn **Alias** erwähnt wird, was **eindeutig** ist. Die zweite Abfrage gibt die Daten zurück, wenn eine **Beschreibung** erwähnt wird, was **NICHT eindeutig** ist.

Hinweis: Beim Erstellen der Mailbox ist die Beschreibung mit dem Alias identisch. Wenn der Alias jedoch aktualisiert wird, wird die Beschreibung nicht aktualisiert. Für kleine Datenbanken ist die erste Abfrage ideal. Um zu zeigen, dass die Beschreibung nach dem Ändern des Alias nicht geändert wird, wird **test3** in **Atest3** geändert und für die nächsten Abschnitte verwendet.

Benutzer mit Gesamtzahl der Nachrichten auflisten

Geben Sie diesen Befehl ein, um eine Liste von Benutzern mit der Gesamtzahl der Nachrichten zu erhalten:

```
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

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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
```

Hinweis: In der zweiten Abfrage ändert sich die Beschreibung nicht von **test3** auf **Atest3**, nachdem der Alias geändert wurde.

Liste von Benutzern mit der Gesamtzahl der Nachrichten basierend auf dem ersten Zeichen in Alias

Geben Sie diesen Befehl ein, um eine Liste von Benutzern abzurufen, deren Gesamtzahl auf dem ersten Zeichen eines Alias basiert:

```
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
```

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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 einige wichtige Hinweise zu dieser Abfrage:

- Der **A%** legt die Abfrage auf Aliase fest, die mit dem Buchstaben A beginnen.
- Im Format kann **Spaltenname wie 'condition' sein**. Hier ist der Spaltenname **Alias** für die erste Abfrage und **Beschreibung** für die zweite Abfrage.

Hier einige Beispielbedingungen:

- **_n%** - Der erste Buchstabe kann ein beliebiges Zeichen (ein Platzhalter) sein, gefolgt vom Buchstaben n und einer beliebigen Anzahl von Zeichen.
- **%s** - Legt die Abfrage so fest, dass Aliase aufgerufen werden, die mit den Buchstaben **s** enden.

Die bisher erwähnten Abfragen werden zum Abrufen der gesamten Nachrichten (Posteingang und gelöschte Elemente) verwendet. Im nächsten Abschnitt werden Abfragen beschrieben, die verwendet werden, um die Gesamtzahl der Nachrichten im Posteingang und gelöschten Elementen abzurufen.

Benutzer mit Gesamtzahl der Posteingänge auflisten

Geben Sie diesen Befehl ein, um eine Liste der Benutzer mit den gesamten Posteingang-Nachrichten zu erhalten:

```
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
```

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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 einige wichtige Hinweise zu dieser Abfrage:

- Der **A%** legt die Abfrage auf Aliase fest, die mit dem Buchstaben A beginnen.
- Im Format **kann Spaltenname wie 'condition' sein**. Hier ist der Spaltenname **Alias** für die erste Abfrage und **Beschreibung** für die zweite Abfrage.

Hier einige Beispielbedingungen:

- **_n%** - Der erste Buchstabe kann ein beliebiges Zeichen (ein Platzhalter) sein, gefolgt vom Buchstaben n und einer beliebigen Anzahl von Zeichen.
- **%s** - Legt die Abfrage so fest, dass Aliase aufgerufen werden, die mit den Buchstaben **s** enden.

Hinweis: In diesem Beispiel wird eine Bedingung verwendet, um Benutzer mit einem Alias/einer Beschreibung zu beschränken, die mit dem Buchstaben **A** beginnt.

Hier einige Varianten dieser Abfrage:

- Um alle Benutzer aufzulisten, entfernen **und Alias wie 'A%'** in der ersten Abfrage, oder **und Beschreibung wie 'A%'** in der zweiten Abfrage.
- Um einen bestimmten Benutzer aufzulisten (z. B. die Anzahl für Anirudh's Mailbox aufzulisten), ersetzen Sie **und Alias wie 'A%'** durch **und alias='Anirudh'** in der ersten Abfrage, oder ersetzen **und beschreiben Sie wie 'A%'** durch **und description ='Anirudh'** in der zweiten Abfrage. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.

Liste von Benutzern mit insgesamt gelöschten Nachrichten

Geben Sie diesen Befehl ein, um eine Liste der Benutzer mit den insgesamt gelöschten Nachrichten zu erhalten:

```
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

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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
```

Hinweis: In diesem Beispiel gibt es keine gelöschten Nachrichten, sodass die Ausgabe als **Keine Datensätze gefunden** angezeigt wird.

Hier einige wichtige Hinweise zu dieser Abfrage:

- Der **A%** legt die Abfrage auf Aliase fest, die mit dem Buchstaben A beginnen.
- Im Format **kann Spaltenname wie 'condition' sein**. Hier ist der Spaltenname **Alias** für die erste Abfrage und **Beschreibung** für die zweite Abfrage.

Hier einige Beispielbedingungen:

- **_n%** - Der erste Buchstabe kann ein beliebiges Zeichen (ein Platzhalter) sein, gefolgt vom Buchstaben n und einer beliebigen Anzahl von Zeichen.
- **%s** - Legt die Abfrage so fest, dass Aliase aufgerufen werden, die mit den Buchstaben **s** enden.

Hinweis: In diesem Beispiel wird eine Bedingung verwendet, um Benutzer mit einem Alias/einer Beschreibung zu beschränken, die mit dem Buchstaben A beginnt.

Hier einige Varianten dieser Abfrage:

- Um alle Benutzer aufzulisten, entfernen **und Alias wie 'A%'** in der ersten Abfrage, oder **und Beschreibung wie 'A%'** in der zweiten Abfrage.
- Um einen bestimmten Benutzer aufzulisten (z. B. die Anzahl für Anirudh's Mailbox aufzulisten), ersetzen Sie **und Alias wie 'A%'** durch **und alias='Anirudh'** in der ersten Abfrage, oder ersetzen **und beschreiben Sie wie 'A%'** durch **und description ='Anirudh'** in der zweiten Abfrage. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.

Benutzer mit Gesamtzahl, Posteingang und gelöschten Nachrichten auflisten

Geben Sie diesen Befehl ein, um eine Liste der Benutzer mit der Gesamtzahl, dem Posteingang und den gelöschten Nachrichten zu erhalten:

```
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
```

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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 einige wichtige Hinweise zu dieser Abfrage:

- Der **A%** legt die Abfrage auf Aliase fest, die mit dem Buchstaben A beginnen.
- Im Format kann Spaltenname wie 'condition' sein. Hier ist der Spaltenname **Alias** für die erste Abfrage und **Beschreibung** für die zweite Abfrage.

Hier einige Beispielbedingungen:

- **_n%** - Der erste Buchstabe kann ein beliebiges Zeichen (ein Platzhalter) sein, gefolgt vom Buchstaben n und einer beliebigen Anzahl von Zeichen.
- **%s** - Legt die Abfrage so fest, dass Aliase aufgerufen werden, die mit den Buchstaben s enden.

Hinweis: In diesem Beispiel wird eine Bedingung verwendet, um Benutzer mit einem Alias/einer Beschreibung zu beschränken, die mit dem Buchstaben A beginnt.

Hier einige Varianten dieser Abfrage:

- Um alle Benutzer aufzulisten, entfernen **und Alias wie 'A%'** in der ersten Abfrage, oder **und Beschreibung wie 'A%'** in der zweiten Abfrage.
- Um einen bestimmten Benutzer aufzulisten (z. B. die Anzahl für Anirudh's Mailbox aufzulisten), ersetzen Sie **und Alias wie 'A%'** durch **und alias='Anirudh'** in der ersten Abfrage, oder ersetzen **und beschreiben Sie wie 'A%'** durch **und description ='Anirudh'** in der zweiten Abfrage. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die

Abfrage fehlschlägt.

Liste der Anzahl der Benutzermeldungen mit der ältesten Ankunftszeit der Nachricht

Diese Abfrage kann verwendet werden, um zu bestimmen, ob die geplanten Aufträge für die Bereinigung von Mailboxen wirksam werden:

```
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
```

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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
```

Hinweis: Um die Uhrzeit der letzten Ankunft für die Abfragen in den vorherigen Abschnitten abzurufen, fügen Sie **min(arrivaltime) als OldestMessageTime** unmittelbar nach **count(*) als Messages** hinzu.

Hier einige Varianten dieser Abfrage:

- Um alle Benutzer mit Aliasen aufzulisten, die mit dem Buchstaben A beginnen, fügen Sie in der ersten Abfrage unmittelbar vor der **Gruppe** die **Alias-Bedingung und Alias wie 'A%'** hinzu, oder **und in der zweiten Abfrage** unmittelbar vor der **Gruppe nach** Bedingung "A%" in **Beschreibungsbedingungen**. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.
- Um einen bestimmten Benutzer aufzulisten (z. B. die Anzahl für Anirudh's Mailbox aufzulisten), fügen Sie **und alias='Anirudh'** in der ersten Abfrage kurz vor der **Gruppe** nach Aliasbedingung hinzu, oder **und description ='Anirudh'** in der zweiten Abfrage unmittelbar vor der **Gruppe nach** Beschreibungsbedingung. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.

Anzahl der Benutzermeldungen mit der ältesten Ankunftszeit und Größe des Postfachs anzeigen/Gesamtdauer der Nachricht

Geben Sie diesen Befehl ein, um eine Liste der Benutzernachrichten mit der Ankunftszeit der ältesten Nachricht und der Mailbox-Größe (ohne Gesamtdauer) zu erhalten:


```
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
```

Hinweis: So rufen Sie die Gesamtdauer der Nachrichten ab: Fügen Sie ",
sum(duration/1000) als TotalDuration_In_sec" kurz vor "**from vw_message**"
hinzu. Vergessen Sie nicht das Komma vor der Summe. Sie kann auch für die Abfragen in
den vorherigen Abschnitten verwendet werden.

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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
```

Hinweis: So rufen Sie die Gesamtdauer der Nachrichten ab: Fügen Sie ",
sum(duration/1000) als TotalDuration_In_sec" kurz vor "**from vw_message**"
hinzu. Vergessen Sie nicht das Komma vor der Summe. Sie kann auch für die Abfragen in
den vorherigen Abschnitten verwendet werden.

Hier einige Varianten dieser Abfrage:

- Um alle Benutzer mit Aliasen aufzulisten, die mit dem Buchstaben A beginnen, fügen Sie in der ersten Abfrage unmittelbar vor der **Gruppe** die **Alias-Bedingung und Alias wie 'A%'** hinzu, oder **und in der zweiten Abfrage** unmittelbar vor der **Gruppe nach** Bedingung "A%" in **Beschreibungsbedingungen**. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.
- Um einen bestimmten Benutzer aufzulisten (z. B. die Anzahl für Anirudh's Mailbox aufzulisten), fügen Sie **und alias='Anirudh'** in der ersten Abfrage kurz vor der **Gruppe** nach Aliasbedingung hinzu, oder **und description ='Anirudh'** in der zweiten Abfrage unmittelbar vor der **Gruppe nach** Beschreibungsbedingung. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.

Liste der Benutzer-Posteingänge und der Anzahl der gelöschten Nachrichten mit der ältesten Ankunftszeit und der Größe des Posteingangs/Gesamtdauer der Nachricht

Geben Sie diesen Befehl ein, um eine Liste der Benutzer-Posteingang und der Anzahl gelöschter Nachrichten mit der Ankunftszeit der ältesten Nachricht und der Mailbox-Größe (ohne

Gesamtdauer) zu erhalten:

```
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
```

Hinweis: So rufen Sie die Gesamtdauer der Nachrichten ab: Fügen Sie ",
sum(duration/1000) als TotalDuration_In_sec" kurz vor "**from vw_message**"
hinzu. Vergessen Sie nicht das Komma vor der Summe. Sie kann auch für die Abfragen in
den vorherigen Abschnitten verwendet werden.

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

```
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
```

Hinweis: So rufen Sie die Gesamtdauer der Nachrichten ab: Fügen Sie ",
sum(duration/1000) als TotalDuration_In_sec" kurz vor "**from vw_message**"
hinzu. Vergessen Sie nicht das Komma vor der Summe. Sie kann auch für die Abfragen in
den vorherigen Abschnitten verwendet werden.

Hier einige Varianten dieser Abfrage:

- Um alle Benutzer mit Aliasen aufzulisten, die mit dem Buchstaben A beginnen, fügen Sie in der ersten Abfrage unmittelbar vor der **Gruppe** die **Alias-Bedingung und Alias wie 'A%'** hinzu, oder **und in der zweiten Abfrage** unmittelbar vor der **Gruppe nach** Bedingung "A%" in **Beschreibungsbedingungen**. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.
- Um einen bestimmten Benutzer aufzulisten (z. B. die Anzahl für Anirudh's Mailbox aufzulisten), fügen Sie **und alias='Anirudh'** in der ersten Abfrage kurz vor der **Gruppe** nach Aliasbedingung hinzu, oder **und description ='Anirudh'** in der zweiten Abfrage unmittelbar vor der **Gruppe nach** Beschreibungsbedingung. Stellen Sie sicher, dass der Austausch in der richtigen Position ist, oder dass die Abfrage fehlschlägt.

Geben Sie die Gesamtzahl der Nachrichten für alle Mailboxen an.

Geben Sie diesen Befehl ein, um die Gesamtzahl der Nachrichten für alle kombinierten Mailboxen abzurufen:

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

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

Mailbox-Größe für Benutzer mit Sende- und Empfangsbeschränkungen auflisten

Geben Sie diesen Befehl ein, um die Mailbox-Größe des Benutzers mit Sende- und Empfangsbeschränkungen abzurufen:

```
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
```

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

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

Listen Sie alle Mailbox-Größen für Benutzer mit Sende- und Empfangsbeschränkungen auf.

Geben Sie diesen Befehl ein, um alle Mailboxgrößen für Benutzer mit Sende- und Empfangsbeschränkungen zu erhalten:

```
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
```

Aus den gleichen Gründen, die im Abschnitt [Liste der Gesamtzahl der Nachrichten mit einem bekannten Alias](#) erwähnt werden, kann diese Abfrage auch verwendet werden:

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

Um alle Benutzer mit Aliasen aufzulisten, die mit dem Buchstaben **A** beginnen, fügen Sie **und Alias wie "A%"** in der ersten Abfrage kurz nach der **vw_usermailboxmap.mailboxid=unitymbxdb1:vw_mailbox.mailboxobjektid** und vor der **Bestellung nach** Bedingung hinzu, oder Sie können **eine Bestellung hinzufügen** wobei eine Beschreibung wie 'A%' in der zweiten Abfrage kurz vor der Bestellung durch Bedingung. Stellen Sie sicher, dass diese in der richtigen Position hinzugefügt wird, oder die Abfrage schlägt fehl.

Geben Sie die Gesamtgröße aller kombinierten Mailboxen an.

Geben Sie diesen Befehl ein, um die Gesamtgröße aller kombinierten Mailboxen abzurufen:

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

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

```
admin:
```