Requêtes SQL CUC pour des nombres de messages et des tailles de boîte aux lettres

Contenu

Introduction

Ce document décrit comment obtenir le nombre de messages et la taille d'une boîte aux lettres d'utilisateur avec des requêtes du SQL (SQL) par l'intermédiaire du CLI. Ces données peuvent également être récupérées avec l'outil de vidage de la mémoire d'utilisateur, de la page Outils de Cisco Unified Communications.

Conditions préalables

Conditions requises

Cisco recommande que vous ayez la connaissance du Cisco Unity Connection (CUC).

Composants utilisés

Les informations dans ce document sont basées sur des versions 8.X et ultérieures CUC, mais ces informations pourraient fonctionner pour des versions antérieures aussi bien.

Bases de données

Les requêtes SQL sont formées avec les données de ces bases de données :

- **UnityDirDB** - Cette base de données contient les informations utilisateur utilisateur.
- **UnityMbxDB1** - Cette base de données contient les informations de boîte aux lettres d'utilisateur.

Tableaux

Les requêtes SQL sont formées avec les données dans ces vues. *Une vue* est une table qui est une combinaison de deux tables ou plus, ou les mêmes données dans une table simple.

Ces vues sont utilisées dans la base de données d'**UnityDirDB** :

- **vw_mailbox** - Cette vue contient le mappage entre l'utilisateur et la boîte aux lettres.
- **vw_user** - Cette vue contient les informations utilisateur utilisateur.

Ces vues sont utilisées dans la base de données **UnityMbxDB1** :

- **vw_message** - Cette vue contient un élément de message dans le système. Cette table est simplement un titulaire des propriétés du message.
- **vw_mailbox** - Cette vue contient une boîte aux lettres sur le système ce les messages d'arrivée d'attentes. Cette table contient les informations générales au sujet de la boîte aux lettres individuelle, pour inclure la manière dans laquelle des messages sont enregistrés dans la boîte aux lettres.

Requêtes SQL

Cette section décrit les diverses requêtes SQL que vous pouvez utiliser dans CUC.

Répertoriez tout le compte de messages avec connu alias

Sélectionnez cette commande afin d'obtenir une liste de tous les messages comptent avec connu 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
```

Cette requête est une complexité élevée, la requête de double-base de données qui implique de plusieurs tables. Pour des serveurs avec une base de données et une taille de boîte aux lettres très grandes, une longue période pourrait s'écouler avant qu'un résultat apparaisse, même au-dessus d'une heure, qui n'est pas idéale. Dans de tels scénarios, vous pouvez utiliser cette requête à la place :

```
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 première requête renvoie les données quand le **pseudonyme** est mentionné, qui est **seul**. La deuxième requête renvoie les données quand la **description** est mentionnée, qui n'est pas **seule**.

> **Note**: Quand la boîte aux lettres est créée, la description est identique que le pseudonyme ; cependant, quand le pseudonyme est mis à jour, la description n'est pas mise à jour. Pour de petites bases de données, la première requête est idéale. Afin d'expliquer que la description n'est pas changée après que le pseudonyme soit modifié, **test3** est modifié à **Atest3** et utilisé pour les sections suivantes.

Utilisateurs de liste avec le compte total de messages

Sélectionnez cette commande afin d'obtenir une liste d'utilisateurs avec tout le compte de messages :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

> **Note**: Dans la deuxième requête, la description ne change pas de **test3** en **Atest3** après que le pseudonyme soit changé.

Utilisateurs de liste avec le compte total de messages basé sur le premier caractère dans alias

Sélectionnez cette commande afin d'obtenir une liste d'utilisateurs avec tous les messages comptent basé sur le premier caractère d'un pseudonyme :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

Voici quelques informations importantes au sujet de cette requête :

- **L'A%** place la requête pour frapper les pseudonymes qui commencent par la lettre R.
- Le format est **où le columnname aiment la « condition »**. Ici, le nom de colonne est **alias** pour la premières requête et **description** pour la deuxième requête.

Voici quelques états d'exemple :

- **_n%** - La première lettre peut être n'importe quel caractère (un masque), suivi de la lettre **n** et d'un certain nombre de caractères.
- **%s** - Ceci place la requête pour frapper les pseudonymes qui finissent avec la lettre **S.**

Les requêtes qui sont mentionnées jusqu'ici sont utilisées afin d'obtenir tous les messages (boîte de réception et éléments supprimés). La section suivante décrit les requêtes qui sont utilisées afin d'obtenir le nombre total de messages dans la boîte de réception et les éléments supprimés.

Utilisateurs de liste avec les messages totaux de boîte de réception

Sélectionnez cette commande afin d'obtenir une liste d'utilisateurs avec tous les messages de boîte de réception :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

Voici quelques informations importantes au sujet de cette requête :

- **L'A%** place la requête pour frapper les pseudonymes qui commencent par la lettre R.
- Le format est **où le columnname aiment la « condition »**. Ici, le nom de colonne est **alias** pour la premières requête et **description** pour la deuxième requête.

Voici quelques états d'exemple :

- **_n%** - La première lettre peut être n'importe quel caractère (un masque), suivi de la lettre **n** et d'un certain nombre de caractères.
- **%s** - Ceci place la requête pour frapper les pseudonymes qui finissent avec la lettre **S.**

  Note: Dans cet exemple, une condition est utilisée afin de limiter des utilisateurs avec un pseudonyme/description qui commence par la lettre **R.**

Ce sont quelques variations de cette requête :

- Afin de répertorier tous les utilisateurs, retirez **et alias comme « A% »** dans la première requête, ou **et la description comme « A% »** dans la deuxième requête.
- Afin de répertorier un utilisateur particulier (répertoriez le compte pour la boîte aux lettres d'Anirudh, par exemple), le remplacer **et alias comme « A% »** par **et alias='Anirudh** dans la première requête, ou le remplacer **et la description comme « A% »** par **et description = ' Anirudh** dans la deuxième requête. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

Utilisateurs de liste avec les messages supprimés par total

Sélectionnez cette commande afin d'obtenir une liste d'utilisateurs avec les messages supprimés par total :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

> **Note**: Dans cet exemple il n'y a aucun message supprimé, ainsi la sortie apparaît en tant qu'**aucun enregistrements trouvés**.

Voici quelques informations importantes au sujet de cette requête :

- **L'A%** place la requête pour frapper les pseudonymes qui commencent par la lettre R.
- Le format est **où le columnname aiment la « condition »**. Ici, le nom de colonne est **alias** pour la premières requête et **description** pour la deuxième requête.

Voici quelques états d'exemple :

- **_n%** - La première lettre peut être n'importe quel caractère (un masque), suivi de la lettre **n** et d'un certain nombre de caractères.
- **%s** - Ceci place la requête pour frapper les pseudonymes qui finissent avec la lettre **S.**

  > **Note**: Dans cet exemple, une condition est utilisée afin de limiter des utilisateurs avec un pseudonyme/description qui commence par la lettre **R.**

Ce sont quelques variations de cette requête :

- Afin de répertorier tous les utilisateurs, retirez **et alias comme « A% »** dans la première requête, ou **et la description comme « A% »** dans la deuxième requête.
- Afin de répertorier un utilisateur particulier (répertoriez le compte pour la boîte aux lettres d'Anirudh, par exemple), le remplacer **et alias comme « A% »** par **et alias='Anirudh** dans la première requête, ou le remplacer **et la description comme « A% »** par **et description = ' Anirudh** dans la deuxième requête. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

Utilisateurs de liste avec le total, la boîte de réception, et les messages supprimés

Sélectionnez cette commande afin d'obtenir une liste d'utilisateurs avec le total, la boîte de réception, et les messages supprimés :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

Voici quelques informations importantes au sujet de cette requête :

- **L'A%** place la requête pour frapper les pseudonymes qui commencent par la lettre R.
- Le format est **où le columnname aiment la « condition »**. Ici, le nom de colonne est **alias** pour la premières requête et **description** pour la deuxième requête.

Voici quelques états d'exemple :

- **_n%** - La première lettre peut être n'importe quel caractère (un masque), suivi de la lettre **n** et d'un certain nombre de caractères.
- **%s** - Ceci place la requête pour frapper les pseudonymes qui finissent avec la lettre **S.**

  Note: Dans cet exemple, une condition est utilisée afin de limiter des utilisateurs avec un pseudonyme/description qui commence par la lettre **R.**

Ce sont quelques variations de cette requête :

- Afin de répertorier tous les utilisateurs, retirez **et alias comme « A% »** dans la première requête, ou **et la description comme « A% »** dans la deuxième requête.
- Afin de répertorier un utilisateur particulier (répertoriez le compte pour la boîte aux lettres d'Anirudh, par exemple), le remplacer **et alias comme « A% »** par **et alias='Anirudh** dans la première requête, ou le remplacer **et la description comme « A% »** par **et description = ' Anirudh** dans la deuxième requête. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

Nombre de messages d'utilisateur de liste avec l'heure d'arrivée du message le plus

ancien

Cette requête peut être utilisée afin de déterminer si les travaux planifiés pour le nettoyage de mailboxe les prennent effet :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

> **Note**: Afin d'obtenir la dernière heure d'arrivée pour les requêtes dans les sections précédentes, ajoutez le **min(arrivaltime) comme OldestMessageTime** juste après le **compte (*) comme messages.**

Ce sont quelques variations de cette requête :

- Afin de répertorier tous les utilisateurs avec les pseudonymes qui commencent par la lettre A, ajoutez **et alias comme « A% »** dans la première requête juste avant le **groupe par** état de **pseudonyme**, ou **et la description comme « A% »** dans la deuxième requête juste avant le **groupe par** état de **description**. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

- Afin de répertorier un utilisateur particulier (répertoriez le compte pour la boîte aux lettres d'Anirudh, par exemple), ajoutez **et alias='Anirudh** dans la première requête juste avant le **groupe par** état de **pseudonyme**, ou **et description = ' Anirudh** dans la deuxième requête juste avant le **groupe par** état de **description**. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

Répertoriez le nombre de messages d'utilisateur avec l'heure d'arrivée du message le plus ancien et la durée de taille de boîte aux lettres/total

Sélectionnez cette commande afin d'obtenir une liste du nombre de messages d'utilisateur avec l'heure de l'arrivée de message le plus ancien et la taille de boîte aux lettres (sans durée 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
undeliverablemessagesmailbox 1 2012-07-05 01:10:19.961 317003
```

> **Note**: Pour obtenir toute la durée des messages : ajoutez « **, sum(duration/1000) comme TotalDuration_In_sec »** juste avant « de **vw_message ».** N'oubliez pas la virgule avant somme. Ceci peut également être utilisé pour les requêtes dans les sections précédentes.

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

> **Note**: Dans la commande pour obtenir toute la durée des messages : ajoutez « **, sum(duration/1000) comme TotalDuration_In_sec »** juste avant « de **vw_message ».** N'oubliez pas la virgule avant somme. Ceci peut également être utilisé pour les requêtes dans les sections précédentes.

Ce sont quelques variations de cette requête :

- Afin de répertorier tous les utilisateurs avec les pseudonymes qui commencent par la lettre A, ajoutez **et alias comme « A% »** dans la première requête juste avant le **groupe par** état de **pseudonyme**, ou **et la description comme « A% »** dans la deuxième requête juste avant le **groupe par** état de **description**. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

- Afin de répertorier un utilisateur particulier (répertoriez le compte pour la boîte aux lettres d'Anirudh, par exemple), ajoutez **et alias='Anirudh** dans la première requête juste avant le **groupe par** état de **pseudonyme**, ou **et description = ' Anirudh** dans la deuxième requête juste avant le **groupe par** état de **description**. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

Répertoriez la boîte de réception d'utilisateur et les messages supprimés comptent avec l'heure d'arrivée du message le plus ancien et la durée de taille de boîte aux lettres/total

Sélectionnez cette commande afin d'obtenir une liste de la boîte de réception d'utilisateur et les messages supprimés comptent avec l'heure de l'arrivée de message le plus ancien et la taille de boîte aux lettres (sans durée 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
```

> **Note**: Afin d'obtenir toute la durée des messages : ajoutez « **, sum(duration/1000) comme TotalDuration_In_sec »** juste avant « de **vw_message ».** N'oubliez pas la virgule avant somme. Ceci peut également être utilisé pour les requêtes dans les sections précédentes.

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

```
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
```

> **Note**: Pour obtenir toute la durée des messages : ajoutez « **, sum(duration/1000) comme TotalDuration_In_sec »** juste avant « de **vw_message ».** N'oubliez pas la virgule avant somme. Ceci peut également être utilisé pour les requêtes dans les sections précédentes.

Ce sont quelques variations de cette requête :

- Afin de répertorier tous les utilisateurs avec les pseudonymes qui commencent par la lettre A, ajoutez **et alias comme « A% »** dans la première requête juste avant le **groupe par** état de **pseudonyme**, ou **et la description comme « A% »** dans la deuxième requête juste avant le **groupe par** état de **description**. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

- Afin de répertorier un utilisateur particulier (répertoriez le compte pour la boîte aux lettres d'Anirudh, par exemple), ajoutez **et alias='Anirudh** dans la première requête juste avant le **groupe par** état de **pseudonyme**, ou **et description = ' Anirudh** dans la deuxième requête juste avant le **groupe par** état de **description**. Assurez-vous que le remplacement est en position précise, ou la requête échoue.

Répertoriez le nombre total de messages pour toutes les boîtes aux lettres

Sélectionnez cette commande afin d'obtenir le nombre total de messages pour toutes les boîtes aux lettres combinées :

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

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

Répertoriez un utilisateur que la taille de boîte aux lettres avec envoient et recevez les limites

Sélectionnez cette commande afin d'obtenir l'utilisateur que la taille de boîte aux lettres avec envoient et recevez les limites :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

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

Répertoriez tout l'utilisateur que les tailles de boîte aux lettres avec envoient et recevez les limites

Sélectionnez cette commande afin d'obtenir tout les l'utilisateur que les tailles de boîte aux lettres avec envoient et recevez les limites :

```
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
```

Pour les mêmes raisons qui sont mentionnés dans la liste que tous les messages comptent avec une section de pseudonyme connue, cette requête peut également être utilisée :

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

Comme variation de cette requête afin de répertorier tous les utilisateurs avec les pseudonymes qui commencent par la lettre **A**, ajoutez **et alias comme « A% »** dans la première requête juste après le **vw_usermailboxmap.mailboxid=unitymbxdb1:vw_mailbox.mailboxobjectid** et avant **) commande par** condition, ou vous pouvez ajouter **où description comme « A% »** dans la deuxième requête juste avant la **commande par** condition. Assurez-vous que ceci est ajouté en position correcte, ou la requête échoue.

Répertoriez la taille totale de toutes les boîtes aux lettres combinées

Sélectionnez cette commande afin d'obtenir la taille totale de toutes les boîtes aux lettres combinées :

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

(sum)
-------
2683210

admin:
```