Requêtes SQL CUC pour le nombre de messages et la taille des boîtes aux lettres

Contenu

Introduction
Conditions préalables
Conditions requises
Components Used
Bases de données
Tableaux
Requêtes SQL
Répertorier le nombre total de messages avec un alias connu
Liste des utilisateurs avec nombre total de messages
Répertorier les utilisateurs dont le nombre total de messages est basé sur le premier caractère de l'alias
Répertorier les utilisateurs avec le nombre total de messages de boîte de réception
Répertorier les utilisateurs avec le nombre total de messages supprimés
Répertorier les utilisateurs avec le nombre total, la boîte de réception et les messages supprimés
Répertorier le nombre de messages utilisateur avec l'heure d'arrivée la plus ancienne des messages
Répertorier le nombre de messages utilisateur avec l'heure d'arrivée du message le plus ancien et la taille de la boîte aux lettres / la durée totale
Répertorier le nombre de messages supprimés et de boîtes de réception utilisateur avec l'heure d'arrivée des messages la plus ancienne et la taille de la boîte aux lettres / durée totale
Répertorier le nombre total de messages pour toutes les boîtes aux lettres
Indiquez la taille d'une boîte aux lettres utilisateur avec les limites d'envoi et de réception
Répertorier toutes les tailles de boîte aux lettres utilisateur avec les limites d'envoi et de réception
Répertorier la taille totale de toutes les boîtes aux lettres combinées

Introduction

Ce document décrit comment obtenir le nombre et la taille des messages d'une boîte aux lettres utilisateur avec des requêtes SQL (Structured Query Language) via l'interface de ligne de commande. Ces données peuvent également être récupérées à l'aide de l'outil de vidage des données utilisateur, à partir de la page Outils de communications unifiées Cisco.

Conditions préalables

Conditions requises

Cisco vous recommande de connaître Cisco Unity Connection (CUC).

Components Used

Les informations de ce document sont basées sur CUC versions 8.X et ultérieures, mais ces informations peuvent également fonctionner pour les versions antérieures.

Bases de données

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

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

Tableaux

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

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

- **vw_mailbox** - Cet affichage contient le mappage entre l'utilisateur et la boîte aux lettres.
- **vw_user** - Cet affichage contient les informations relatives à l'utilisateur.

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

- **vw_message** - Cet affichage contient un élément de message dans le système. Cette table n'est qu'un détenteur de propriétés de message.
- **vw_mailbox** - Cette vue contient une boîte aux lettres sur le système qui contient des messages entrants. Cette table contient des informations générales sur chaque boîte aux lettres, pour inclure la manière dont les messages sont stockés dans la boîte aux lettres.

Requêtes SQL

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

Répertorier le nombre total de messages avec un alias connu

Entrez cette commande afin d'obtenir une liste du nombre total de messages avec un alias connu :

```
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 requête double base de données complexe qui implique plusieurs tables. Pour les serveurs dont la base de données et la taille de la boîte aux lettres sont très volumineuses, une période prolongée peut s'écouler avant qu'une sortie n'apparaisse, même sur une heure, ce qui n'est pas idéal. 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 retourne les données lorsque l'**alias** est mentionné, ce qui est **unique**. La deuxième requête retourne les données lorsque la **description** est mentionnée, ce qui n'est **PAS unique**.

> **Note**: Lorsque la boîte aux lettres est créée, la description est identique à l'alias ; cependant, lorsque l'alias est mis à jour, la description n'est pas mise à jour. Pour les petites bases de données, la première requête est idéale. Afin de démontrer que la description n'est pas modifiée après la modification de l'alias, **test3** est modifié en **Atest3** et utilisé pour les sections suivantes.

Liste des utilisateurs avec nombre total de messages

Entrez cette commande afin d'obtenir une liste d'utilisateurs avec le nombre total 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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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 passe pas de **test3** à **Atest3** après la modification de l'alias.

Répertorier les utilisateurs dont le nombre total de messages est basé sur le premier caractère de l'alias

Entrez cette commande afin d'obtenir une liste d'utilisateurs avec le nombre total de messages basé sur le premier caractère d'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
```

Pour les mêmes raisons que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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 remarques importantes sur cette requête :

- Le **A%** définit la requête pour qu'elle touche les alias commençant par la lettre A.
- Le format est **où nom de colonne comme 'condition'**. Ici, le nom de colonne est **alias** pour la première requête et la **description** pour la deuxième requête.

Voici quelques exemples de conditions :

- **_n%** - La première lettre peut être n'importe quel caractère (un caractère générique), suivie de la lettre **n** et de n'importe quel nombre de caractères.
- **%s** - Définit la requête pour qu'elle touche les alias qui se terminent par les lettres **s**.

Les requêtes mentionnées jusqu'à présent sont utilisées afin d'obtenir le total des messages (boîtes de réception et éléments supprimés). La section suivante décrit les requêtes utilisées pour obtenir le nombre total de messages dans la boîte de réception et les éléments supprimés.

Répertorier les utilisateurs avec le nombre total de messages de boîte de réception

Entrez cette commande afin d'obtenir une liste d'utilisateurs avec le total des messages de la 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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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 remarques importantes sur cette requête :

- Le **A%** définit la requête pour qu'elle touche les alias commençant par la lettre A.
- Le format est **où nom de colonne comme 'condition'**. Ici, le nom de colonne est **alias** pour la première requête et la **description** pour la deuxième requête.

Voici quelques exemples de conditions :

- **_n%** - La première lettre peut être n'importe quel caractère (un caractère générique), suivie de la lettre **n** et de n'importe quel nombre de caractères.
- **%s** - Définit la requête pour qu'elle touche les alias qui se terminent par les lettres **s**.

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

Voici quelques variantes de cette requête :

- Afin de répertorier tous les utilisateurs, supprimez **et alias comme 'A%'** dans la première requête, ou **et description comme 'A%'** dans la deuxième requête.
- Afin de lister un utilisateur particulier (liste le nombre de boîtes aux lettres d'Anirudh, par exemple), remplacez **et alias comme 'A%'** par **et alias='Anirudh'** dans la première requête, ou remplacez **et description comme 'A%'** par **et description ='Anirudh'** dans la deuxième requête. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

Répertorier les utilisateurs avec le nombre total de messages supprimés

Entrez cette commande afin d'obtenir une liste d'utilisateurs avec le nombre total de messages

supprimés :

```
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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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, de sorte que la sortie apparaît comme **Aucun enregistrement trouvé**.

Voici quelques remarques importantes sur cette requête :

- Le **A%** définit la requête pour qu'elle touche les alias commençant par la lettre A.
- Le format est **où nom de colonne comme 'condition'**. Ici, le nom de colonne est **alias** pour la première requête et la **description** pour la deuxième requête.

Voici quelques exemples de conditions :

- **_n%** - La première lettre peut être n'importe quel caractère (un caractère générique), suivie de la lettre **n** et de n'importe quel nombre de caractères.
- **%s** - Définit la requête pour qu'elle touche les alias qui se terminent par les lettres **s**.

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

Voici quelques variantes de cette requête :

- Afin de répertorier tous les utilisateurs, supprimez **et alias comme 'A%'** dans la première requête, ou **et description comme 'A%'** dans la deuxième requête.
- Afin de lister un utilisateur particulier (liste le nombre de boîtes aux lettres d'Anirudh, par exemple), remplacez **et alias comme 'A%'** par **et alias='Anirudh'** dans la première requête, ou remplacez **et description comme 'A%'** par **et description ='Anirudh'** dans la deuxième requête. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

Répertorier les utilisateurs avec le nombre total, la boîte de réception et les messages supprimés

Entrez 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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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 remarques importantes sur cette requête :

- Le **A%** définit la requête pour qu'elle touche les alias commençant par la lettre A.
- Le format est **où nom de colonne comme 'condition'**. Ici, le nom de colonne est **alias** pour la première requête et la **description** pour la deuxième requête.

Voici quelques exemples de conditions :

- **_n%** - La première lettre peut être n'importe quel caractère (un caractère générique), suivie de la lettre **n** et de n'importe quel nombre de caractères.
- **%s** - Définit la requête pour qu'elle touche les alias qui se terminent par les lettres **s**.

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

Voici quelques variantes de cette requête :

- Afin de répertorier tous les utilisateurs, supprimez **et alias comme 'A%'** dans la première requête, ou **et description comme 'A%'** dans la deuxième requête.
- Afin de lister un utilisateur particulier (liste le nombre de boîtes aux lettres d'Anirudh, par exemple), remplacez **et alias comme 'A%'** par **et alias='Anirudh'** dans la première requête, ou remplacez **et description comme 'A%'** par **et description ='Anirudh'** dans la deuxième requête. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

Répertorier le nombre de messages utilisateur avec l'heure d'arrivée la plus

ancienne des messages

Cette requête peut être utilisée afin de déterminer si les travaux planifiés pour le nettoyage de boîte aux lettres 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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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 l'Heure d'arrivée dernière pour les requêtes des sections précédentes, ajoutez **min(arrivaltime) en tant qu'OldestMessageTime** juste après **count(*) en tant que Messages.**

Voici quelques variantes de cette requête :

- Afin de lister tous les utilisateurs avec des alias qui commencent par la lettre A, ajoutez **et alias comme 'A%'** dans la première requête juste avant le **groupe par** condition **d'alias**, ou **et description comme 'A%'** dans la deuxième requête juste avant le **groupe par** condition de **description**. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

- Afin de lister un utilisateur particulier (liste le nombre de boîtes aux lettres d'Anirudh, par exemple), ajoutez **et alias='Anirudh'** dans la première requête juste avant le **groupe par** condition **d'alias**, ou **et description ='Anirudh'** dans la deuxième requête juste avant le **groupe par** condition **de description**. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

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

Entrez cette commande afin d'obtenir une liste du nombre de messages utilisateur avec l'heure d'arrivée du message le plus ancien et la taille de la 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
```

> **Remarque** : Pour obtenir la durée totale des messages : ajouter **", sum(term/1000) as TotalDuration_In_sec »** juste avant "**from vw_message ».** N'oubliez pas la virgule avant la somme. Ceci peut également être utilisé pour les requêtes des sections précédentes.

Pour les mêmes raisons que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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**: Afin d'obtenir la durée totale des messages : ajouter **", sum(term/1000) as TotalDuration_In_sec »** juste avant "**from vw_message ».** N'oubliez pas la virgule avant la somme. Ceci peut également être utilisé pour les requêtes des sections précédentes.

Voici quelques variantes de cette requête :

- Afin de lister tous les utilisateurs avec des alias qui commencent par la lettre A, ajoutez **et alias comme 'A%'** dans la première requête juste avant le **groupe par** condition **d'alias**, ou **et description comme 'A%'** dans la deuxième requête juste avant le **groupe par** condition de **description**. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

- Afin de lister un utilisateur particulier (liste le nombre de boîtes aux lettres d'Anirudh, par exemple), ajoutez **et alias='Anirudh'** dans la première requête juste avant le **groupe par** condition **d'alias**, ou **et description ='Anirudh'** dans la deuxième requête juste avant le **groupe par** condition **de description**. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

Répertorier le nombre de messages supprimés et de boîtes de réception utilisateur avec l'heure d'arrivée des messages la plus ancienne et la taille de la boîte aux lettres / durée totale

Entrez cette commande afin d'obtenir une liste de la boîte de réception de l'utilisateur et du nombre de messages supprimés avec l'heure d'arrivée du message le plus ancien et la taille de la 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 la durée totale des messages : ajouter **", sum(term/1000) as TotalDuration_In_sec »** juste avant "**from vw_message ».** N'oubliez pas la virgule avant la somme. Ceci peut également être utilisé pour les requêtes des sections précédentes.

Pour les mêmes raisons que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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
```

> **Remarque** : pour obtenir la durée totale des messages : ajouter **", sum(term/1000) as TotalDuration_In_sec »** juste avant "**from vw_message ».** N'oubliez pas la virgule avant la somme. Ceci peut également être utilisé pour les requêtes des sections précédentes.

Voici quelques variantes de cette requête :

- Afin de lister tous les utilisateurs avec des alias qui commencent par la lettre A, ajoutez **et alias comme 'A%'** dans la première requête juste avant le **groupe par** condition **d'alias**, ou **et description comme 'A%'** dans la deuxième requête juste avant le **groupe par** condition de **description**. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

- Afin de lister un utilisateur particulier (liste le nombre de boîtes aux lettres d'Anirudh, par exemple), ajoutez **et alias='Anirudh'** dans la première requête juste avant le **groupe par** condition **d'alias**, ou **et description ='Anirudh'** dans la deuxième requête juste avant le **groupe par** condition **de description**. Assurez-vous que le remplacement se trouve dans la position exacte ou que la requête échoue.

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

Entrez 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
```

Indiquez la taille d'une boîte aux lettres utilisateur avec les limites d'envoi et de réception

Entrez cette commande afin d'obtenir la taille de la boîte aux lettres utilisateur avec les limites d'envoi et de réception :

```
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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, cette requête peut également être utilisée :

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

Répertorier toutes les tailles de boîte aux lettres utilisateur avec les limites d'envoi et de réception

Entrez cette commande afin d'obtenir toutes les tailles de boîtes aux lettres utilisateur avec les limites d'envoi et de réception :

```
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 que celles mentionnées dans la section Liste du nombre total de messages avec un alias connu, 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 une variante de cette requête afin de lister tous les utilisateurs avec des alias qui commencent par la lettre A, ajoutez et alias comme 'A%' dans la première requête juste après la vw_usermailboxmap.mailboxid=unitymbxdb1:vw_mailbox.mailboxobjectid et avant) ordre par condition, ou vous pouvez ajouter **où description comme 'A%** dans la deuxième requête juste avant la **commande par** condition. Assurez-vous qu'il est ajouté à la bonne position, sinon la requête échoue.

Répertorier la taille totale de toutes les boîtes aux lettres combinées

Entrez 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:
```

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.