

# 메시지 수 및 사서함 크기에 대한 CUC SQL 쿼리

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[데이터베이스](#)

[테이블](#)

[SQL 쿼리](#)

[알려진 별칭으로 총 메시지 수 나열](#)

[총 메시지 수가 있는 사용자 나열](#)

[별칭의 첫 문자를 기준으로 총 메시지 수가 있는 사용자 나열](#)

[총 받은 편지함 메시지가 있는 사용자 나열](#)

[총 삭제된 메시지가 있는 사용자 나열](#)

[총 메시지, 받은 편지함 및 삭제된 메시지의 사용자 나열](#)

[가장 오래된 메시지 도착 시간이 있는 사용자 메시지 수 나열](#)

[가장 오래된 메시지 도착 시간 및 사서함 크기/총 기간이 있는 사용자 메시지 수 나열](#)

[가장 오래된 메시지 도착 시간 및 사서함 크기/총 기간이 있는 사용자 받은 편지함 및 삭제된 메시지 수 나열](#)

[모든 사서함에 대한 총 메시지 수 나열](#)

[보내기 및 받기 제한이 있는 사용자 사서함 크기 나열](#)

[보내기 및 받기 제한이 있는 모든 사용자 사서함 크기 나열](#)

[결합된 모든 사서함의 총 크기 나열](#)

## 소개

이 문서에서는 CLI를 통해 SQL(Structured Query Language) 쿼리를 사용하여 사용자 사서함의 메시지 수 및 크기를 얻는 방법에 대해 설명합니다. 또한 [Cisco Unified Communications Tools](#) 페이지에서 [사용자 데이터 덤프](#) 도구를 사용하여 이 데이터를 검색할 수도 있습니다.

## 사전 요구 사항

### 요구 사항

Cisco에서는 Cisco Unity Connection(CUC)에 대한 지식을 보유하고 있는 것이 좋습니다.

### 사용되는 구성 요소

이 문서의 정보는 CUC 버전 8.X 이상을 기반으로 하지만 이 정보는 이전 버전에서도 사용할 수 있습니다.

## 데이터베이스

SQL 쿼리는 다음 데이터베이스의 데이터로 구성됩니다.

- **UnityDirDB** - 이 데이터베이스에는 사용자 관련 정보가 포함되어 있습니다.
- **UnityMbxDB1** - 이 데이터베이스에는 사용자 사서함 정보가 들어 있습니다.

## 테이블

SQL 쿼리는 이러한 뷰의 데이터로 구성됩니다. 뷰는 둘 이상의 테이블의 조합이거나 단일 테이블의 동일한 데이터인 테이블입니다.

이러한 뷰는 UnityDirDB 데이터베이스에서 사용됩니다.

- **vw\_mailbox** - 이 보기에는 사용자와 사서함 간의 매핑이 포함됩니다.
- **vw\_user** - 이 보기에는 사용자 관련 정보가 포함됩니다.

이러한 뷰는 UnityMbxDB1 데이터베이스에서 사용됩니다.

- **vw\_message** - 이 보기에는 시스템에 메시지 항목이 포함되어 있습니다. 이 테이블은 메시지 속성 소유자에 불과합니다.
- **vw\_mailbox** - 이 보기에는 인바운드 메시지가 있는 시스템의 사서함이 포함되어 있습니다. 이 표에는 개별 사서함에 대한 일반 정보가 포함되어 있으며, 이 정보에는 메시지를 사서함에 저장하는 방법이 포함됩니다.

## SQL 쿼리

이 섹션에서는 CUC에서 사용할 수 있는 다양한 SQL 쿼리에 대해 설명합니다.

### 알려진 별칭으로 총 메시지 수 나열

알려진 별칭으로 총 메시지 수 목록을 가져오려면 이 명령을 입력합니다.

```
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
```

```
-----
```

이 쿼리는 여러 테이블을 포함하는 복잡한 이중 데이터베이스 쿼리입니다. 데이터베이스 및 사서함 크기가 매우 큰 서버의 경우 출력이 표시되기 전에 시간이 1시간 이상 경과할 수 있으므로 적합하지 않습니다. 이러한 시나리오에서는 대신 이 쿼리를 사용할 수 있습니다.

```
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  
첫 번째 쿼리는 **별칭**이 언급될 때 데이터를 반환하며, 이는 **고유합니다**. 두 번째 쿼리는 **설명을** 언급할 때 데이터를 반환하며, 이는 **고유하지 않습니다**.

**참고:** 사서함을 만들 때 설명은 별칭과 동일합니다. 그러나 별칭이 업데이트되면 설명이 업데이트되지 않습니다. 소규모 데이터베이스의 경우 첫 번째 쿼리가 적합합니다. 별칭이 변경된 후 설명이 변경되지 않음을 보여주기 위해 **test3**이 **Test3**으로 수정되고 다음 섹션에 사용됩니다.

## 총 메시지 수가 있는 사용자 나열

총 메시지 수가 있는 사용자 목록을 가져오려면 다음 명령을 입력합니다.

```
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
```

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

**참고:** 두 번째 쿼리에서는 별칭이 변경된 후 **test3**에서 **Test3**로 설명이 변경되지 않습니다.

## 별칭의 첫 문자를 기준으로 총 메시지 수가 있는 사용자 나열

별칭의 첫 번째 문자를 기준으로 총 메시지 수가 포함된 사용자 목록을 가져오려면 다음 명령을 입력합니다.

```
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

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

다음은 이 쿼리에 대한 몇 가지 중요한 참고 사항입니다.

- **A%**는 문자 A로 시작하는 별칭에 대한 쿼리를 설정합니다.
- 형식은 'condition'과 같은 열 이름입니다. 여기서 열 이름은 첫 번째 쿼리의 별칭 및 두 번째 쿼리 설명입니다.

다음은 몇 가지 조건 예입니다.

- **\_n%** - 첫 번째 문자는 임의의 문자(와일드카드)가 될 수 있으며 그 뒤에 n자와 임의의 문자 수가 올 수 있습니다.
- **%s** - 문자로 끝나는 별칭에 대한 쿼리를 설정합니다.

지금까지 언급된 쿼리는 총 메시지(받은 편지함 및 삭제된 항목)를 가져오기 위해 사용됩니다. 다음 섹션에서는 받은 편지함 및 삭제된 항목의 총 메시지 수를 얻기 위해 사용되는 쿼리에 대해 설명합니다.

## 총 받은 편지함 메시지가 있는 사용자 나열

총 받은 편지함 메시지가 있는 사용자 목록을 가져오려면 다음 명령을 입력합니다.

```
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

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는

다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

다음은 이 쿼리에 대한 몇 가지 중요한 참고 사항입니다.

- **A%**는 문자 A로 시작하는 별칭에 대한 쿼리를 설정합니다.
- 형식은 'condition'과 같은 열 이름입니다. 여기서 열 이름은 첫 번째 쿼리의 별칭 및 두 번째 쿼리 설명입니다.

다음은 몇 가지 조건 예입니다.

- **\_n%** - 첫 번째 문자는 임의의 문자(와일드카드)가 될 수 있으며 그 뒤에 n자와 임의의 문자 수가 올 수 있습니다.
- **%s** - 문자로 끝나는 별칭에 대한 쿼리를 설정합니다.

**참고:** 이 예에서 조건은 문자 A로 시작하는 별칭/설명으로 사용자를 제한하기 위해 사용됩니다

다음은 이 쿼리의 몇 가지 변형입니다.

- 모든 사용자를 나열하려면 첫 번째 쿼리에서 'A%'와 같은 별칭 또는 두 번째 쿼리에서 'A%'와 같은 설명을 제거하고 별칭으로 지정합니다.
- 특정 사용자를 나열하려면(예: Anirudh의 사서함 수 나열) 및 별칭을 첫 번째 쿼리에서 'A%'와 alias='Anirudh'와 같이 바꾸거나, 'A%'와 같은 및 설명을 두 번째 쿼리에서 'A%'와 description='Anirudh'로 바꾸십시오. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.

## 총 삭제된 메시지가 있는 사용자 나열

총 삭제된 메시지의 사용자 목록을 가져오려면 다음 명령을 입력합니다.

```
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

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

**참고:** 이 예에서는 삭제된 메시지가 없으므로 출력이 **No records found(레코드 없음)**로 표시됩니다.

다음은 이 쿼리에 대한 몇 가지 중요한 참고 사항입니다.

- **A%**는 문자 A로 시작하는 별칭에 대한 쿼리를 설정합니다.
- 형식은 **'condition'**과 같은 열 이름입니다. 여기서 열 이름은 첫 번째 쿼리의 별칭 및 두 번째 쿼리 설명입니다.

다음은 몇 가지 조건 예입니다.

- **\_n%** - 첫 번째 문자는 임의의 문자(와일드카드)가 될 수 있으며 그 뒤에 n자와 임의의 문자 수가 올 수 있습니다.
- **%s** - 문자로 끝나는 별칭에 대한 쿼리를 설정합니다.

**참고:** 이 예에서 조건은 문자 A로 시작하는 별칭/설명으로 사용자를 제한하기 위해 사용됩니다

다음은 이 쿼리의 몇 가지 변형입니다.

- 모든 사용자를 나열하려면 첫 번째 쿼리에서 **'A%'**와 같은 별칭 또는 두 번째 쿼리에서 **'A%'**와 같은 설명을 제거하고 별칭으로 지정합니다.
- 특정 사용자를 나열하려면(예: Anirudh의 사서함 수 나열) 및 별칭을 첫 번째 쿼리에서 **'A%'**와 **alias='Anirudh'**와 같이 바꾸거나, **'A%'**와 같은 및 설명을 두 번째 쿼리에서 **'A%'**와 **description='Anirudh'**로 바꾸십시오. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.

## 총 메시지, 받은 편지함 및 삭제된 메시지의 사용자 나열

총 메시지, 받은 편지함 및 삭제된 메시지의 사용자 목록을 가져오려면 다음 명령을 입력합니다.

```
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
```

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

다음은 이 쿼리에 대한 몇 가지 중요한 참고 사항입니다.

- **A%**는 문자 A로 시작하는 별칭에 대한 쿼리를 설정합니다.
- 형식은 **'condition'**과 같은 열 이름입니다. 여기서 열 이름은 첫 번째 쿼리의 별칭 및 두 번째 쿼리 설명입니다.

다음은 몇 가지 조건 예입니다.

- **\_n%** - 첫 번째 문자는 임의의 문자(와일드카드)가 될 수 있으며 그 뒤에 n자와 임의의 문자 수가 올 수 있습니다.
- **%s** - 문자로 끝나는 별칭에 대한 쿼리를 설정합니다.

**참고:** 이 예에서 조건은 문자 A로 시작하는 별칭/설명으로 사용자를 제한하기 위해 사용됩니다.

다음은 이 쿼리의 몇 가지 변형입니다.

- 모든 사용자를 나열하려면 첫 번째 쿼리에서 **'A%'**와 같은 별칭 또는 두 번째 쿼리에서 **'A%'**와 같은 설명을 제거하고 별칭으로 지정합니다.
- 특정 사용자를 나열하려면(예: Anirudh의 사서함 수 나열) 및 별칭을 첫 번째 쿼리에서 **'A%'**와 **alias='Anirudh'**와 같이 바꾸거나, **'A%'**와 같은 및 설명을 두 번째 쿼리에서 **'A%'**와 **description='Anirudh'**로 바꾸십시오. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.

## 가장 오래된 메시지 도착 시간이 있는 사용자 메시지 수 나열

이 쿼리를 사용하여 사서함 정리에 대해 예약된 작업이 적용되는지 여부를 결정할 수 있습니다.

```
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
```

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

**참고:**이전 섹션의 쿼리에 대한 마지막 도착 시간을 가져오려면 `count(*)`를 `Messages`로 바로 뒤에 `min(arrivaltime)`을 `OldestMessageTime`으로 추가합니다.

다음은 이 쿼리의 몇 가지 변형입니다.

- 문자 A로 시작하는 별칭을 가진 모든 사용자를 나열하려면 별칭 조건별 그룹 바로 앞에 'A%'와 같은 별칭과 별칭을 추가하고 별칭을 설명 조건별 그룹 바로 앞에 있는 두 번째 쿼리에서 'A%'와 같은 설명을 추가합니다. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.
- 특정 사용자를 나열하려면(예: Anirudh의 사서함 수 나열) 별칭 조건으로 그룹 바로 앞에 있는 첫 번째 쿼리에서 `add` 및 `alias='Anirudh'`를 추가하고, 설명 조건별로 그룹 바로 앞에 있는 두 번째 쿼리에서 `description = 'Anurdh'`를 추가합니다. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.

## 가장 오래된 메시지 도착 시간 및 사서함 크기/총 기간이 있는 사용자 메시지 수 나열

가장 오래된 메시지 도착 시간 및 사서함 크기(총 기간 제외)와 함께 사용자 메시지 수 목록을 가져 오려면 다음 명령을 입력합니다.

```
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
```

**참고:** 메시지의 총 기간을 가져오려면 다음을 수행합니다. "from vw\_message" 바로 직전에 "sum(duration/1000)을 TotalDuration\_In\_sec"으로 추가합니다. sum 앞에 쉼표를 잊지 마십시오. 이전 섹션의 질의에도 사용할 수 있습니다.

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

**참고:**총 메시지 기간을 가져오려면 "from vw\_message" 바로 직전에 "sum(duration/1000)을 TotalDuration\_In\_sec"으로 추가합니다. sum 앞에 쉼표를 잊지 마십시오. 이전 섹션의 질의에도 사용할 수 있습니다.

다음은 이 쿼리의 몇 가지 변형입니다.



- 문자 A로 시작하는 별칭을 가진 모든 사용자를 나열하려면 별칭 조건별 그룹 바로 앞에 'A%'와 같은 별칭과 별칭을 추가하고 별칭을 설명 조건별 그룹 바로 앞에 있는 두 번째 쿼리에서 'A%'와 같은 설명을 추가합니다. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.
- 특정 사용자를 나열하려면(예: Anirudh의 사서함 수 나열) 별칭 조건으로 그룹 바로 앞에 있는 첫 번째 쿼리에서 add 및 alias='Anirudh'를 추가하고, 설명 조건별로 그룹 바로 앞에 있는 두 번째 쿼리에서 description='Anurdh'를 추가합니다. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.

## 가장 오래된 메시지 도착 시간 및 사서함 크기/총 기간이 있는 사용자 받은 편지함 및 삭제된 메시지 수 나열

가장 오래된 메시지의 도착 시간 및 사서함 크기(총 기간 제외)와 함께 사용자 받은 편지함 및 삭제된 메시지 수 목록을 가져오려면 다음 명령을 입력합니다.

```
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
```

**참고:** 총 메시지 기간을 가져오려면 "from vw\_message" 바로 직전에 "sum(duration/1000)을 TotalDuration\_In\_sec"으로 추가합니다. sum 앞에 쉼표를 잊지 마십시오. 이전 섹션의 질의에도 사용할 수 있습니다.

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

```
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
```

**참고:** 총 메시지 기간을 가져오려면 "from vw\_message" 바로 직전에 "sum(duration/1000)을 TotalDuration\_In\_sec"으로 추가합니다. sum 앞에 쉼표를 잊지 마십시오. 이전 섹션의 질의에도 사용할 수 있습니다.

다음은 이 쿼리의 몇 가지 변형입니다.

- 문자 A로 시작하는 별칭을 가진 모든 사용자를 나열하려면 별칭 조건별 그룹 바로 앞에 'A%'와 같은 별칭과 별칭을 추가하고 별칭을 설명 조건별 그룹 바로 앞에 있는 두 번째 쿼리에서 'A%'와 같은 설명을 추가합니다. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.
- 특정 사용자를 나열하려면(예: Anirudh의 사서함 수 나열) 별칭 조건으로 그룹 바로 앞에 있는 첫 번째 쿼리에서 add 및 alias='Anirudh'를 추가하고, 설명 조건별로 그룹 바로 앞에 있는 두 번째 쿼리에서 description ='Anurdh'를 추가합니다. 대체 항목이 정확한 위치에 있는지 또는 쿼리가 실패하는지 확인합니다.

## 모든 사서함에 대한 총 메시지 수 나열

결합된 모든 사서함에 대한 총 메시지 수를 얻으려면 이 명령을 입력합니다.

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

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

## 보내기 및 받기 제한이 있는 사용자 사서함 크기 나열

보내기 및 받기 제한이 있는 사용자 사서함 크기를 가져오려면 다음 명령을 입력합니다.

```
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
```

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 같이 사용할 수도 있습니다.

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

## 보내기 및 받기 제한이 있는 모든 사용자 사서함 크기 나열

보내기 및 받기 제한이 있는 모든 사용자 사서함 크기를 가져오려면 이 명령을 입력합니다.

```
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
```

List the Total Messages Count [with a Known Alias](#) 섹션에 언급된 것과 동일한 이유로, 이 쿼리는 다음과 [같이](#) 사용할 수도 있습니다.

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

문자 **A**로 시작하는 모든 사용자를 나열하기 위한 이 쿼리의 변형으로서 조건별 **vw\_usermailboxmap.mailboxid=unitymxd1:vw\_mailbox.mailboxobjectid** 및 이전) 순서를 기준으로 첫 번째 쿼리에서 'A%'와 같은 별칭을 추가하고 별칭을 추가하거나,, 조건에서 'A%'와 같은 설명을 추가할 수 있습니다. 올바른 위치 또는 쿼리가 실패합니다.

## 결합된 모든 사서함의 총 크기 나열

결합된 모든 사서함의 총 크기를 얻으려면 이 명령을 입력합니다.

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

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

```
admin:
```