Error: Subquery returned more than 1 value

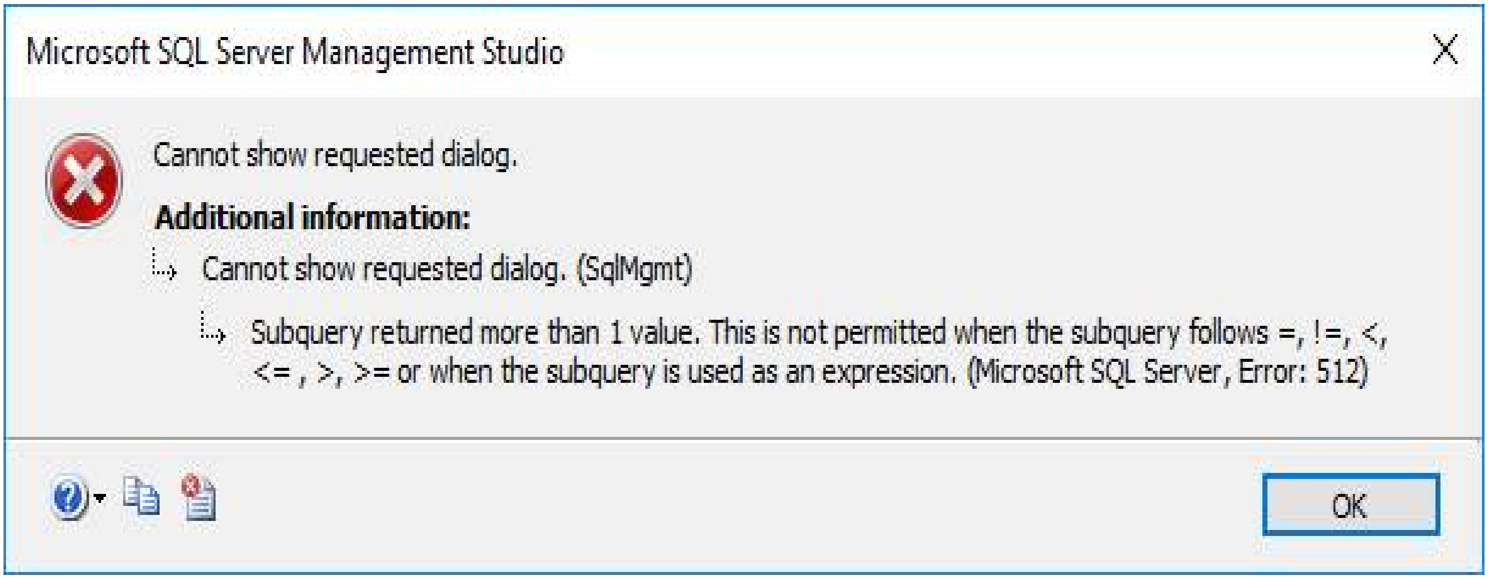
Last updated by | Vitor Tomaz | Mar 27, 2023 at 10:40 AM PDT

Contents

Issue

I am using SQL Server Management Studio to connect to my Managed Instance – no problem there.

However, when I try to access the properties of my database (right-click on database and select properties), I get the following error:



Microsoft SQL Server Management Studio

Cannot show requested dialog.

Additional information:

Cannot show requested dialog. (SqlMgmt)

Subquery returned more than 1 value. This is not permitted when the subquery follows =, !=, <, <= , >, >= or when the subquery is used as an expression. (Microsoft SQL Server, Error: 512)

Msg 512: Subquery returned more than 1 value. This is not permitted when the query follows =, !=, <, <=, >, >= or when the subquery is used as an expression.

Investigation/Analysis

The information for COPY_ONLY full backups initiated by user was stored in msdb.dbo.backupset table. Column database_name contained logical database name. In order to have better automatic backup transparency we have recently (Nov 2022) enabled storing information about automatic backups into msdb.dbo.backupset table as well. Due to a defect in the code instead of logical database name we stored a unique identifier of the managed database. That caused the issue with SSMS as the internal query expects to have one of these two.

Customer analysis

Customers can run below query to identify if the issue is due to msdb backupset:

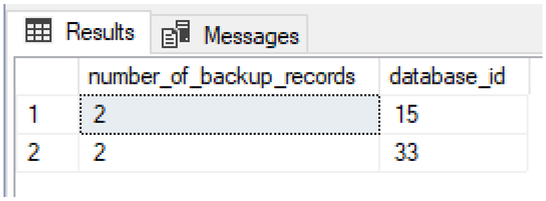
```
create table #tempbackup (database_name nvarchar(128), [type] char(1), backup_finish_date datetime)

insert into #tempbackup select database_name, [type], max(backup_finish_date) from msdb..backupset where [type


select count(db_id(database_name)) as number_of_backup_records, db_id(database_name) as database_id
from #tempbackup where type = 'D' and db_id(database_name) > 4
group by db_id(database_name)
having count(db_id(database_name)) > 1
```

This query should return no results in a normal case. But sometimes, it does because we store two rows per database - once with a user-friendly database name and once with logical name (GUID). This confuses SSMS as it rightfully expects only one row here.

Sample output:



	number_of_backup_records	database_id
1	2	15
2	2	33

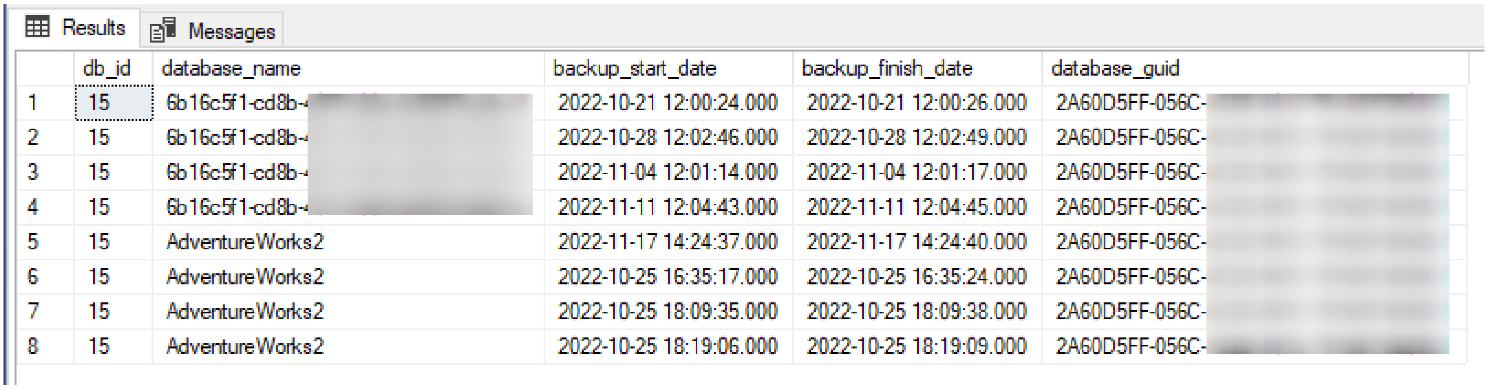
To identify which backup set is causing the issue, customer can run below query:

```
SELECT DB_ID(database_name) AS db_id, database_name, backup_start_date, backup_finish_date, database_guid
FROM msdb.dbo.backupset
WHERE type = 'D'
and DB_ID(database_name) = <database_id>  --database_id from above query where the count was > 1
ORDER BY 1, 2;
```

For example:

```
SELECT DB_ID(database_name) AS db_id, database_name, backup_start_date, backup_finish_date, database_guid
FROM msdb.dbo.backupset
WHERE type in ('D')
and DB_ID(database_name) = 15
ORDER BY 1, 2;
```

Output from above sample query:



	db_id	database_name	backup_start_date	backup_finish_date	database_guid
1	15	6b16c5f1-cd8b-[redacted]	2022-10-21 12:00:24.000	2022-10-21 12:00:26.000	2A60D5FF-056C-[redacted]
2	15	6b16c5f1-cd8b-[redacted]	2022-10-28 12:02:46.000	2022-10-28 12:02:49.000	2A60D5FF-056C-[redacted]
3	15	6b16c5f1-cd8b-[redacted]	2022-11-04 12:01:14.000	2022-11-04 12:01:17.000	2A60D5FF-056C-[redacted]
4	15	6b16c5f1-cd8b-[redacted]	2022-11-11 12:04:43.000	2022-11-11 12:04:45.000	2A60D5FF-056C-[redacted]
5	15	AdventureWorks2	2022-11-17 14:24:37.000	2022-11-17 14:24:40.000	2A60D5FF-056C-[redacted]
6	15	AdventureWorks2	2022-10-25 16:35:17.000	2022-10-25 16:35:24.000	2A60D5FF-056C-[redacted]
7	15	AdventureWorks2	2022-10-25 18:09:35.000	2022-10-25 18:09:38.000	2A60D5FF-056C-[redacted]
8	15	AdventureWorks2	2022-10-25 18:19:06.000	2022-10-25 18:19:09.000	2A60D5FF-056C-[redacted]

The output shows logical database name (AdvenutureWorks2) in msdb..backupset as a result of manual COPY_ONLY backup initiated on the database, while the unique identifier of the managed database (6b16c5f1-cd8b-xxxx-xxxx-xxxxxxxxxxxx) was stored as a result of newly introduced feature for automatic backup transparency (Nov 2022). Since the database_name column contains different values (logical database name and managed database unique id) for the same database_guid (2A60D5FF-056C-xxxx-xxxx-xxxxxxxxxxxx), this causes issue with SSMS, as the internal SSMS query expects to have one of these two.

Mitigation

Mitigation is to clear backup history from MSDB for the manual backups (backups that show in above example with Logical database name).

Customers can use sp_delete_backuphistory

```
EXEC msdb..sp_delete_backuphistory @oldest_date = '<mm/dd/yyyy>'
```

This command deletes the entries for backup sets older than the specified date. In the above example, the date would be '11/17/2022'.

For example:

```
EXEC msdb..sp_delete_backuphistory @oldest_date = '11/17/2022'
```

This will delete all backup history (irrespective of the database) that is older than 11/17/2022.

Skipping the date, will entirely purge the backup history. For example:

```
EXEC msdb..sp_delete_backuphistory
```

Alternatively, customers can delete the backup history based on database name using sp_delete_database_backuphistory

```
EXEC msdb..sp_delete_database_backuphistory @database_name = '<database_name>'
```

In the above example, the query would look like:

```
EXEC msdb..sp_delete_database_backuphistory @database_name = 'AdventureWorks2'
```

This will delete all backup reference from the manual backupset for the database (AdventureWorks2) and keep the backupset from the automated backups (appearing as guid under database_name in the sample output above).

Public Doc Reference

https://learn.microsoft.com/en-us/sql/ssms/release-notes-ssms?view=sql-server-ver16#known-issues-1902

Internal Reference:

Umbrella incident - 348945729

Classification

Root cause tree:

How good have you found this content?

🙂 🙁