Imported_database_is_smaller_than_original_database

Last updated by | Vitor Pombeiro | Dec 23, 2021 at 5:29 AM PST

Contents

Issue

Imported database is smaller than original database

Investigation/Analysis

With Azure SQL Database, there are workload patterns where the allocation of underlying data files for databases can become larger than the amount of used data pages. This condition can occur when space used increases and data is subsequently deleted. The reason is because file space allocated is not automatically reclaimed when data is deleted.

If you export your database to a .bacpac file, and then you import it, it could be possible that the target database will be smaller than original one, but if you restore any of the available backups on Azure portal, the allocated size will the same than the original database.

The reason for this is simple, .bacpac file contains Schema and data, while .bak file contains a page-by-page copy of the database, and SQL Server database contains not only data pages, there are also pages with indexes that can be large.

If you are in front of this situation, you could think to shrink your original database to reduce its allocated space. But before shrink your database you need review the space used per table

```
SELECT
t.NAME AS TableName,
s.Name AS SchemaName,
p.rows AS RowCounts,
SUM(a.total_pages) * 8 AS TotalSpaceKB,
SUM(a.used_pages) * 8 AS UsedSpaceKB,
(SUM(a.total_pages) - SUM(a.used_pages)) * 8 AS UnusedSpaceKB
FROM
    sys.tables t
INNER JOIN
    sys.indexes i ON t.OBJECT_ID = i.object_id
INNER JOIN
    sys.partitions p ON i.object_id = p.OBJECT_ID AND i.index_id = p.index_id
INNER JOIN
    sys.allocation_units a ON p.partition_id = a.container_id
LEFT OUTER JOIN
    sys.schemas s ON t.schema_id = s.schema_id
WHERE t.is_ms_shipped = 0
    AND i.OBJECT_ID > 255
GROUP BY
    t.Name, s.Name, p.Rows
ORDER BY
    t.Name
```

If you see that your tables have to many Unused space, you need start recovering this unused space. You can do it for all tables using the following script that reorganize and rebuilt all indexes

https://docs.microsoft.com/en-us/sql/relational-databases/indexes/reorganize-and-rebuild-indexes?view=sql-server-ver15

or just rebuilt indexes for specific tables

```
ALTER INDEX ALL ON [dbo].[Table_name] REBUILD
```

But what happens if any of your tables don't have any index (Heap table). In this case query will not fail but will neither do anything. For this cases you will need rebuilt the Heap table using the following syntax.

```
ALTER TABLE [dbo].[ Table_name] REBUILD
```

https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes?view=sql-server-ver15

If you has finished the process correctly you will see that unused space for table has been decreased, and now you are ready to shrink your database. Do can do it executing this script.

https://github.com/yochananrachamim/AzureSQL/blob/master/Incremental Shrink.txt

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:

Root Cause: Azure SQL v3/Import/Export/Import issues/Other

How good have you found this content?