# Shrink - Monitor database size in Kusto

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:27 AM PST

Contents



## Issue

This TSG provides you with Kusto queries to show you the current database size and individual table sizes, e.g. for monitoring the progress of a customer's shrink attempts.

## Investigation / Analysis

If you are looking for a quick overview on the customer's database sizes and storage consumption, then rather go to ASC and run the troubleshooter for detailed information.

If you are looking for troubleshooting steps about what to do about issues while shrinking a database, look into these articles:

- Shrink Database Best Practices
- Shrink Database and shrink file with Low Priority
- Shrink failed due to backups
- Shrink - Incremental Shrink

## Mitigation

### MonDmIoVirtualFileStats

This Kusto query will show you the change in space consumption in your user database and in tempdb over time. The granularity is 1 hour, so if there is a massive growth and shrink between the capture times, you won't see it on the telemetry. It will however give you a good overview about trends and patterns.

```
let startTime = datetime(2022-10-12 06:30:00Z);
let endTime = datetime(2022-10-12 08:00:00Z);
let srv = "servername";
let db = "databasename";
MonDmIoVirtualFileStats
| where  TIMESTAMP >= startTime
| where  TIMESTAMP <= endTime
| where LogicalServerName =~ srv
| where db_name in ("tempdb", db)
| extend size_on_disk_mb=(size_on_disk_bytes*1.0/1024/1024)
| extend max_space_used_percent = toint((1.0 * spaceused_mb / max_size_mb) * 100)
| project TIMESTAMP, NodeName, AppName, LogicalServerName, db_name, database_id, file_id, type_desc, is_primar
| order by TIMESTAMP asc nulls last, db_name, file_id asc nulls last
```

Sample output:

| TIMESTAMP | Node... | AppName | LogicalServer... | db_name | database_id | file_id | type_desc | is_primary_... | is_remote | storage_type | growth | spaceused_mb | size_on_disk_mb | max_size_mb | max_space_used_percent |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2022-10-12 06:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | tempdb | 2 | 1 | ROWS | 0 | 0 | 8 | 8192 | 9 | 16 | 14225 | 0 |
| 2022-10-12 06:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | tempdb | 2 | 2 | LOG | 0 | 0 | 8 | 8192 | 3 | 16 | 6758 | 0 |
| 2022-10-12 06:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | AdventureWorks | 9 | 1 | ROWS | 1 | 1 | 1 | 2048 | 240 | 368 | 2048 | 11 |
| 2022-10-12 06:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | AdventureWorks | 9 | 2 | LOG | 1 | 1 | 1 | 2048 | 2 | 264 | 5120 | 0 |
| 2022-10-12 07:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | tempdb | 2 | 1 | ROWS | 0 | 0 | 8 | 8192 | 10 | 16 | 14225 | 0 |
| 2022-10-12 07:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | tempdb | 2 | 2 | LOG | 0 | 0 | 8 | 8192 | 4 | 16 | 6758 | 0 |
| 2022-10-12 07:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | AdventureWorks | 9 | 1 | ROWS | 1 | 1 | 1 | 2048 | 241 | 368 | 2048 | 11 |
| 2022-10-12 07:55:13.... | _DB_12 | e4aeb2328e63 | weholgerl | AdventureWorks | 9 | 2 | LOG | 1 | 1 | 1 | 2048 | 1 | 264 | 5120 | 0 |

## MonWiDmDbPartitionStats

This Kusto query lists all tables in the database with the granularity of 1 day. It allows you to check how individual objects change their page count and row count between days. By setting the start and end date, you can narrow down on specific periods, e.g. over a weekend where the customer was running a shrink operation.

```
//Check table size
let srv = "servername";
let db = "databasename";
let startTime = datetime(2022-11-19 06:00:00Z);
let endTime = datetime(2022-11-20 23:00:00Z);
let timeRange = ago(7d);
let AppNames = MonAnalyticsDBSnapshot
 | where  TIMESTAMP >= startTime
 | where  TIMESTAMP <= endTime
// | where  TIMESTAMP >= timeRange
 | where logical_server_name =~ srv
 | where logical_database_name =~ db
 | extend AppName = sql_instance_name
 | distinct AppName;
let PartitionStats=materialize(MonWiDmDbPartitionStats
 | where  TIMESTAMP >= startTime
 | where  TIMESTAMP <= endTime
// | where  TIMESTAMP >= timeRange
 | where AppName in (AppNames) and logical_database_name !='master' and index_id in (0,1)
 | summarize used_page_count=max(used_page_count), row_count=max(row_count) by database_id, logical_database_n
let FilteredResults=materialize(MonDatabaseMetadata
 | where AppName in (AppNames) and logical_db_name != 'master'
 | where (table_name=='sysclsobjs' and class==50) or (table_name=='sysschobjs' and ['type']=='U ') or (table_n
 | project TIMESTAMP, table_name, class, ['type'], id, name, nsid, indid);
let schemas=FilteredResults
 | where (table_name=='sysclsobjs' and class==50)
 | summarize by schema_id=id, schema_name=tolower(name);
let tables=FilteredResults
 | where (table_name=='sysschobjs' and ['type']=='U ')
 | summarize by schema_id=nsid, object_id=id, table_name=name;
let indexes=FilteredResults
 | where (table_name=='sysidxstats' and indid in (0,1))
 | extend index_type_desc=iff(['type']==0, 'HEAP', iff(['type']==1, 'CLUSTERED', iff(['type']==5, 'CCI', tostr
 | summarize by object_id=id,index_id=indid,index_type=type,index_type_desc;
tables
| join kind=inner (schemas) on schema_id
| join kind=inner (indexes) on object_id
| join kind=inner (PartitionStats) on object_id
| extend logical_server_name = srv
| project logical_server_name, database_id, logical_database_name, schema_id, schema_name, object_id, table_na
| sort by schema_name asc, table_name asc, object_id asc, data_date asc
//| sort by size_kb desc , schema_name asc, table_name asc, object_id asc, data_date asc
```

| server_name | database_id | database_name | schema_id | schema_name | object_id | table_name | table_t |
|---|---|---|---|---|---|---|---|
| servername | 9 | databasename | 5 | humanresources | 2101582525 | Shift | CLUSTER |
| servername | 9 | databasename | 5 | humanresources | 2101582525 | Shift | CLUSTER |
| servername | 9 | databasename | 6 | person | 2133582639 | Address | CLUSTER |
| servername | 9 | databasename | 6 | person | 2133582639 | Address | CLU |
| servername | 9 | databasename | 6 | person | 18099105 | AddressType | CLUSTER |
| servername | 9 | databasename | 6 | person | 18099105 | AddressType | CLUSTER |
| servername | 9 | databasename | 6 | person | 50099219 | BusinessEntity | CLUSTER |
| servername | 9 | databasename | 6 | person | 50099219 | BusinessEntity | CLUSTER |
| servername | 9 | databasename | 6 | person | 82099333 | BusinessEntityAddress | CLUSTER |
| servername | 9 | databasename | 6 | person | 82099333 | BusinessEntityAddress | CLUSTER |

## How good have you found this content?