

Database corruption on Azure SQL Database

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

Contents

- [Issue](#)
- [Investigation / Analysis](#)
- [Mitigation](#)
 - [General guidance](#)
 - [Example](#)
 - [Heap indid=0 or Clustered Index indid=1 - Open IcM](#)
 - [Nonclustered index indid=2 or larger - Consider an index r...](#)
 - [Indexed Views - drop and create the view's clustered index](#)
- [More Information](#)
 - [Sample output of a DBCC CHECKDB with corruption messa...](#)
 - [Sample output of a good DBCC CHECKDB](#)
- [Internal Doc Reference](#)
- [Public Doc Reference](#)

Issue

Database corruption was more common in earlier SQL Server versions on-premise, but have almost disappeared in Azure SQL Database. Related support cases have become very rare, but when the issue occurs, it usually causes a lot of stress to the customer. Database corruption can occur either on the data files or the transaction log. It might either be just a minor issue or can result in a data loss disaster.

For all who are wondering why there are a lot less data corruption cases compared to on-premise SQL Server - see article [Data Integrity in Azure SQL Database](#) for some details about the reason.

Typical error messages are:

Error: 823, Severity: 24, State: 2

The operating system returned error %ls to SQL Server during a %S_MSG at offset %#016l64x in file '%ls'. Additional messages in the SQL Server error log and system event log may provide more detail. This is a severe system-level error condition that threatens database integrity and must be corrected immediately. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.

[Documentation about error 823](#) 

Error: 824, Severity: 24, State: 2.

SQL Server detected a logical consistency-based I/O error: %ls. It occurred during a %S_MSG of page %S_PGID in database ID %d at offset %#016l64x in file '%ls'. Additional messages in the SQL Server error log or system event log may provide more detail.

[Documentation about error 824](#) 

Error 825

A read of the file '%ls' at offset %#016l64x succeeded after failing %d time(s) with error: %ls. Additional messages in the SQL Server error log and system event log may provide more detail. This error condition threatens database integrity and must be corrected. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.

[Documentation about error 825](#) 

Error: 8646, Severity: 21, State: 1. Unable to find index entry in index ID %1, of table %2, in database '%3'. The indicated index is corrupt or there is a problem with the current update plan. Run DBCC CHECKDB or DBCC CHECKTABLE. If the problem persists, contact product support.

Error: 9003, Severity: 20, State: 1. The log scan number (40588348:10536:19) passed to log scan in database '30638c6d-b100-4cc8-b319-ce309a209194' is not valid. This error may indicate data corruption or that the log file (.ldf) does not match the data file (.mdf). If this error occurred during replication, re-create the publication. Otherwise, restore from backup if the problem results in a failure during startup.

[Documentation about error 9003](#) 

Investigation / Analysis

The initial troubleshooting should include the following 3 steps:

1. Confirm the error on the telemetry and find out the exact point in time when the corruption started to occur.
2. Ask the customer to initiate a Point-in-Time restore for shortly before the corruption occurred first. Note that this is not for troubleshooting, but to save time for any data rescue operation that might be needed.
3. Ask the customer to run a DBCC CHECKDB against the affected database.

Regarding 1: confirm corruption on the telemetry

Run the following Kusto query for the customer server. Make sure to cover a larger period to identify the starting point of the issue. If needed, go back as far as the earliest cut-off point in the telemetry.

```
// adapt the filter on 'message' if the customer reported a different suspicious error
let srv = "servername";
let db = "databasename";
let startTime = datetime(2022-09-13 00:30:00Z);
let endTime = datetime(2022-09-30 00:00:00Z);
let timeRange = ago(7d);
let AppNames = MonAnalyticsDBSnapshot
| where logical_server_name =~ srv
| where logical_database_name =~ db
| extend AppName = sql_instance_name
| distinct AppName;
MonSQLSystemHealth
//| where TIMESTAMP > timeRange
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
| where AppName in~ ( AppNames )
| where
    message contains "Error: 823" or message contains "Error: 824" or message contains "Error: 825" or
    message contains "Error: 8646" or message contains "Error: 9003"
| summarize min(message) by AppName, AppTypeName
```

Sample output:

| AppName | AppTypeName | min_message |
|--------------|---------------------|---|
| d3c496492e69 | Worker.Vldb.Storage | 2022-09-24 07:05:36.59 spid103s Error: 824, Severity: 24, State: 2. 2022-09-24 07:05:36.59 spid103s SQL Server detected a logical consistency-based I/O error: incorrect pageid (expected 3:4427757; actual 0:0). It occurred during a read of pag |
| ce6a475db806 | Worker.Vldb.Compute | 2022-09-28 06:52:53.61 spid546 Error: 823, Severity: 24, State: 2. 2022-09-28 06:52:53.61 spid546 The operating system returned error 1460(This operation returned because the timeout period expired,) to SQL Server during a read at offset |
| e410158ae1e5 | Worker.Vldb.Storage | 2022-09-28 06:34:03.66 spid111s getRemoteForeignRedoInfo: DbId [56]. Reading the GFH failed with error: 823, severity 24, state 19. |
| a62b87258a82 | Worker.Vldb.Storage | 2022-09-28 06:34:08.09 spid96s getRemoteForeignRedoInfo: DbId [56]. Reading the GFH failed with error: 823, severity 24, state 19. |
| e49c4abb830c | Worker.Vldb.Storage | 2022-09-28 06:34:01.71 spid95s getRemoteForeignRedoInfo: DbId [56]. Reading the GFH failed with error: 823, severity 24, state 19. |
| e78e05419a56 | Worker.Vldb.Storage | 2022-09-28 06:34:06.52 spid41s getRemoteForeignRedoInfo: DbId [56]. Reading the GFH failed with error: 823, severity 24, state 19. |
| ea1b3b56ecb2 | Worker.Vldb.Storage | 2022-09-28 06:34:05.23 spid100s getRemoteForeignRedoInfo: DbId [56]. Reading the GFH failed with error: 823, severity 24, state 19. |

This specific customer was extremely unlucky to run into a corruption twice (related to a Hyperscale issue). The first occurrence of error 824 was reported for 2022-09-24 07:05:36.59 UTC, whereas error 823 started to occur at 2022-09-28 06:34:01.71 UTC. These would be the dates to give to the customer for the PITR restore in step 2. Note that the output might be unordered.

More sample output for error messages:

2022-10-07 11:03:53.57 spid107 Error: 823, Severity: 24, State: 2.

2022-10-07 11:03:53.57 spid107 The operating system returned error 32(The process cannot access the file because it is being used by another process.) to SQL Server during a read at offset 0x00000000f0000 in file 'https://wasdprodeus1apdw4721.blob.core.windows.net/data-tsmo-synapse-ws-6233ad5f-af17-4d8d-a06e-81293d7c0ed4/data/8b653389-9776-407b-bfa5-1fc144f863bb_1.mdf'. Additional messages in the SQL Server error log and operating system error log may provide more detail. This is a severe system-level error condition that threatens database integrity and must be corrected immediately. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.

2022-09-28 06:52:53.61 spid546 Error: 823, Severity: 24, State: 2.

2022-09-28 06:52:53.61 spid546 The operating system returned error 1460(This operation returned because the timeout period expired,) to SQL Server during a read at offset 0x000000004f2000 in file 'C:\WFRoot\DB.40\Fabric\work\Applications\Worker.Vldb.Compute_App170631\work\data\e5bd6e03-4926-4f95-8f35-22be7b88d1bc_4.ndf'. Additional messages in the SQL Server error log and operating system error log may provide more detail. This is a severe system-level error condition that threatens database integrity and must be corrected immediately. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.

2022-09-05 23:21:51.50 spid89s Error: 824, Severity: 24, State: 2.

2022-09-05 23:21:51.50 spid89s SQL Server detected a logical consistency-based I/O error: incorrect pageid (expected 1:1483; actual 0:0). It occurred during a read of page (1:1483) in database ID 6 at offset 0x00000000b96000 in file 'wasd2prodweu1apfs7155.file.core.windows.net \1r631c5e9bea64604c5386accd95c2fe97d5\data\8789d5f7-56db-47c1-9c63-e95310108a72_1.mdf'. Additional messages in the SQL Server error log or operating system error log may provide more detail. This is a severe error condition that threatens database integrity and must be corrected immediately. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.

2022-09-24 07:05:36.59 spid103s Error: 824, Severity: 24, State: 2.

2022-09-24 07:05:36.59 spid103s SQL Server detected a logical consistency-based I/O error: incorrect pageid (expected 3:4427757; actual 0:0). It occurred during a read of page (3:4427757) in database ID 56 at offset 0x00000871fda000 in file 'https://wasdprodeus1ahspp223.blob.core.windows.net/pageserver-ragrs-f76dbc53-e72561ad-a351-436d-a6dc-64ca2bd17778/data/f76dbc53-55e5-43c9-ad8f-a9fbd9ec7df5_287_2.ndf'. Additional messages in the SQL Server error log or operating system error log may provide more detail. This is a severe error condition that threatens database integrity and must be corrected immediately. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.

Regarding 2: initiate a Point-in-Time restore for shortly before the corruption occurred first

The purpose of this step is to save time for any data rescue operation that might be needed later. Depending on the size of the database, the restore might take a long time. It can run in the background while we continue the troubleshooting.

Ask the customer to initiate a point-in-time restore of the database to a new name as described in [Restore a database from a backup in Azure SQL Database](#). See the Kusto query and sample output from step 1 for the appropriate restore datetime values.

Consider opening an ICM if the first occurrence of the corruption is beyond the PITR retention period (e.g. retention is 7 days but corruption started 15 days ago).

Regarding 3: run a DBCC CHECKDB against the affected database

Run DBCC CHECKDB in the affected customer database to determine the full scope of the database corruption.

```
DBCC CHECKDB
```

You can configure the CPU consumption by specifying a `MAXDOP` value, thus either decreasing or increasing its priority over other workloads:

```
DBCC CHECKDB WITH MAXDOP = 1; -- run slower
DBCC CHECKDB WITH MAXDOP = 8; -- run faster
```

If the database is very big and the customer suspects that only one or few tables are affected, you can consider checking just this table or an individual index:

```
-- check single table
DBCC CHECKTABLE ('HumanResources.Employee');

-- check only one index in the table, together with the heap or clustered index
SELECT index_id, name FROM sys.indexes WHERE object_id = OBJECT_ID('Production.Product');
DECLARE @indid int;
SET @indid = (SELECT index_id FROM sys.indexes
              WHERE object_id = OBJECT_ID('Production.Product')
                AND name = 'AK_Product_Name');
DBCC CHECKTABLE ('Production.Product', @indid);

-- output:
DBCC results for 'Production.Product'.
There are 504 rows in 13 pages for object "Production.Product".
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

Completion time: 2022-10-14T08:02:05.2513046+00:00
```

When it has finished, you should look for output like this:

```
-- no issues:
CHECKDB found 0 allocation errors and 0 consistency errors in database 'XXXXXXX'.

-- corruption errors:
CHECKTABLE found 0 allocation errors and 7 consistency errors in table 'xxxxxxx' (object ID 1221579390).
```

Further considerations on CHECKDB:

- is an online operation
- should not cause blocking
- will consume a lot of CPU, data I/O, memory
- REPAIR_REBUILD and REPAIR_ALLOW_DATA_LOSS does NOT work on Azure SQL Database because you cannot set it to single user mode. This can be done by the PG on the backend platform though.

Mitigation

General guidance

The mitigation steps depend on the error message and the outcome of the DBCC CHECKDB/CHECKTABLE. The guideline is the following:

- If a heap table (indid=0) or a clustered index (indid=1) is affected: open an ICM immediately.
- If a nonclustered index or indexed view is affected: rebuild the index. An ICM is not absolutely necessary, unless several different tables are affected or if you are unsure about the steps.
- If the corruption is with the transaction log (error 9003), open an ICM. It might be possible to resolve it by a node failover; if it persists it might require a database restore; if it is related to transactional replication or CDC, the customer might have to reset either replication or CDC.

The indid=0 (heap) means that the data pages itself are involved in the corruption. Any wrong step will lead to data loss. The clustered index might be affected in two ways: either the leaf-level pages = data pages are affected, or the index structure (non-leaf-level pages) is affected. The first has again the risk of data loss, the

second could be fixed by an index rebuild. But the difference is not easy to see, so rather go for an ICM if the clustered index is affected.

Example

Let's assume that the following output was returned by DBCC CHECKDB/CHECKTABLE:

```
Msg 8909, Level 16, State 1, Line 10
Table error: Object ID 0, index ID -1, partition ID 0, alloc unit ID -3031890307258974208 (type Unknown), page
The PageId in the page header = (30239:809547096).
CHECKTABLE found 0 allocation errors and 1 consistency errors not associated with any single object.
Msg 8928, Level 16, State 1, Line 10
Object ID 1221579390, index ID 0, partition ID 72057594083868672, alloc unit ID 72057594124107776 (type In-row
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42713252), slot 27 points to page (1:49780352), slot 17. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42359585), slot 71 points to page (1:49780352), slot 30. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42339549), slot 39 points to page (1:49780352), slot 36. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42087587), slot 23 points to page (1:49780352), slot 38. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42532466), slot 29 points to page (1:49780352), slot 39. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42470149), slot 53 points to page (1:49780352), slot 43. Did not
DBCC results for 'order_sale'.
There are 36735628 rows in 874103 pages for object "order_sale".
CHECKTABLE found 0 allocation errors and 7 consistency errors in table 'order_sale' (object ID 1221579390).
repair_allow_data_loss is the minimum repair level for the errors found by DBCC CHECKTABLE (59ef4966-f250-43f2
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

Error message 1

```
Msg 8909, Level 16, State 1, Line 10
Table error: Object ID 0, index ID -1, partition ID 0, alloc unit ID -3031890307258974208 (type Unknown),
page ID (1:49780352) contains an incorrect page ID in its page header. The PageId in the page header =
(30239:809547096).
CHECKTABLE found 0 allocation errors and 1 consistency errors not associated with any single object.
```

This indicates that page ID (1:49780352) contains the corruption. It has an invalid page ID and empty object ID in its header, hinting at that (1:49780352) was incorrectly written, or overwritten by something else, or messed up due to disk corruption. This error symptom is "harmless" in itself, but is usually related to additional errors.

Error message 2

```
Msg 8928, Level 16, State 1, Line 10
Object ID 1221579390, index ID 0, partition ID 72057594083868672, alloc unit ID 72057594124107776
(type In-row data): Page (1:49780352) could not be processed. See other errors for details.
```

The keyword here is "index ID 0", indicating a heap. As soon as you see this, or "index ID 1", you do not look further and go straight for an ICM.

Error message 3

Msg 8993, Level 16, State 1, Line 10

Object ID 1221579390, forwarding row page (1:42713252), slot 27 points to page (1:49780352), slot 17. Did not encounter forwarded row. Possible allocation error.

Msg 8993, Level 16, State 1, Line 10

Object ID 1221579390, forwarding row page (1:42359585), slot 71 points to page (1:49780352), slot 30. Did not encounter forwarded row. Possible allocation error.

These errors confirm that something is wrong with page (1:49780352), affecting several rows and confirming the object ID. The relevant information is in error message 2 though.

Confirm the affected object

You can confirm the affected object by asking the customer to run a `select object_name(1221579390)` (insert the object ID found on the error). The name is also shown in clear text in the summary at the bottom of the DBCC output (it is "order_sale" in this case).

Or you can see for yourself in Kusto:

```
let srv = "servername";
let db = "databasename";
MonDatabaseMetadata
| where LogicalServerName =~ srv
| where logical_db_name =~ db
| project TIMESTAMP, NodeName, AppName, LogicalServerName, logical_db_name, table_name, name, id
| where table_name == 'sysschobjs'
| where id == 50099219 // object_id
| limit 100
| distinct LogicalServerName, logical_db_name, id, name

let srv = "servername";
let db = "databasename";
MonWiSysIndexes
| where LogicalServerName =~ srv
| where logical_database_name =~ db
| where object_id == 50099219 // object_id
//| where index_id == long(0)
| project TIMESTAMP, NodeName, AppName, LogicalServerName, logical_database_name, object_id, index_name, index_id, type_desc;
| distinct LogicalServerName, logical_database_name, object_id, index_name, index_id, type_desc;
```

Sample output:

| LogicalServerName | logical_db_name | id | name |
|-------------------|-----------------|----------|----------------|
| weholgerl | AdventureWorks | 50099219 | BusinessEntity |

| LogicalServerName | logical_database_name | object_id | index_name | index_id | type_desc |
|-------------------|-----------------------|-----------|------------------------------------|----------|--------------|
| weholgerl | AdventureWorks | 50099219 | PK_BusinessEntity_BusinessEntityID | 1 | CLUSTERED |
| weholgerl | AdventureWorks | 50099219 | AK_BusinessEntity_rowguid | 2 | NONCLUSTERED |

Heap indid=0 or Clustered Index indid=1 - Open ICM

Let the PG decide on the best next steps. Depending on the type of corruption, they might be able to repair the corruption e.g. by restoring the corrupted pages from a backup or a restored database (see investigation step

#2 from above!!). In some cases it might be sufficient to failover to another, healthy node.

Nonclustered index indid=2 or larger - Consider an index rebuild

Take the object ID and index ID from the DBCC error message. Use the following SQL query to create the `ALTER INDEX ... REBUILD` command. Execute the appropriate command(s) to rebuild the index. See [ALTER INDEX \(Transact-SQL\)](#) for the full syntax.

```
-- set the correct object_id in the Where clause
SELECT
    'schema_name' = s.name,
    t.object_id,
    'table_name' = t.name,
    i.index_id,
    'index_name' = i.name,
    'rebuild_index_sql' =
        CASE WHEN i.index_id < 2 THEN CONCAT('-- ', QUOTENAME(I.name), ' is heap or clustered index, do not rebuild')
              WHEN i.index_id > 1 and i.index_id < 255 THEN CONCAT('ALTER INDEX ', QUOTENAME(I.name), ' ON ', QUOTENAME(t.name), ' REBUILD;')
              ELSE '--invalid index_id'
        END
    END
FROM sys.tables t
INNER JOIN sys.indexes i ON t.object_id = i.object_id
INNER JOIN sys.schemas s ON t.schema_id = s.schema_id
WHERE index_id > 1          -- do not rebuild heap or clustered index
and t.object_id = 50099219 -- set object_id from DBCC CHECKDB
```

Sample output:

| | schema_name | object_id | table_name | index_id | index_name | rebuild_index_sql |
|---|-------------|-----------|----------------|----------|------------------------------------|--|
| 1 | Person | 50099219 | BusinessEntity | 1 | PK_BusinessEntity_BusinessEntityID | -- [PK_BusinessEntity_BusinessEntityID] is heap or clustered index, do not rebuild |
| 2 | Person | 50099219 | BusinessEntity | 2 | AK_BusinessEntity_rowguid | ALTER INDEX [AK_BusinessEntity_rowguid] ON [Person].[BusinessEntity] REBUILD; |

Indexed Views - drop and create the view's clustered index

It is also possible that an indexed view is affected by the corruption. DBCC CHECKDB/CHECKTABLE would then return either of the following messages:

Error: 8907

Indexed view '%.*Is' (object ID %d) contains rows not produced by the view definition. Refer to Books Online for more information on this error. This does not necessarily represent an integrity issue with the data in this database.

Error: 8908

Indexed view '%.*Is' (object ID %d) does not contain all rows that the view definition produces. Refer to Books Online for more information on this error. This does not necessarily represent an integrity issue with the data in this database.

The mitigation for this is straightforward: simply drop the clustered index on the indexed view and recreate it.

More Information

Sample output of a DBCC CHECKDB with corruption messages

(note this was captured by a DBCC CHECKTABLE, but the output would be similar/same for DBCC CHECKDB)

```
Msg 8909, Level 16, State 1, Line 10
Table error: Object ID 0, index ID -1, partition ID 0, alloc unit ID -3031890307258974208 (type Unknown), page
CHECKTABLE found 0 allocation errors and 1 consistency errors not associated with any single object.
Msg 8928, Level 16, State 1, Line 10
Object ID 1221579390, index ID 0, partition ID 72057594083868672, alloc unit ID 72057594124107776 (type In-row
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42713252), slot 27 points to page (1:49780352), slot 17. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42359585), slot 71 points to page (1:49780352), slot 30. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42339549), slot 39 points to page (1:49780352), slot 36. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42087587), slot 23 points to page (1:49780352), slot 38. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42532466), slot 29 points to page (1:49780352), slot 39. Did not
Msg 8993, Level 16, State 1, Line 10
Object ID 1221579390, forwarding row page (1:42470149), slot 53 points to page (1:49780352), slot 43. Did not
DBCC results for 'order_sale'.
There are 36735628 rows in 874103 pages for object "order_sale".
CHECKTABLE found 0 allocation errors and 7 consistency errors in table 'order_sale' (object ID 1221579390).
repair_allow_data_loss is the minimum repair level for the errors found by DBCC CHECKTABLE (59ef4966-f250-43f2
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
```

Sample output of a good DBCC CHECKDB

```

DBCC results for 'AdventureWorks'.
Service Broker Msg 9675, State 1: Message Types analyzed: 14.
Service Broker Msg 9676, State 1: Service Contracts analyzed: 6.
Service Broker Msg 9667, State 1: Services analyzed: 3.
Service Broker Msg 9668, State 1: Service Queues analyzed: 3.
Service Broker Msg 9669, State 1: Conversation Endpoints analyzed: 0.
Service Broker Msg 9674, State 1: Conversation Groups analyzed: 0.
Service Broker Msg 9670, State 1: Remote Service Bindings analyzed: 0.
Service Broker Msg 9605, State 1: Conversation Priorities analyzed: 0.
DBCC results for 'sys.syrscolls'.
There are 2394 rows in 33 pages for object "sys.syrscolls".
DBCC results for 'sys.sysrowsets'.
There are 390 rows in 6 pages for object "sys.sysrowsets".
(...)
DBCC results for 'Person.AddressType'.
There are 6 rows in 1 pages for object "Person.AddressType".
DBCC results for 'Person.BusinessEntity'.
There are 20777 rows in 96 pages for object "Person.BusinessEntity".
DBCC results for 'Person.BusinessEntityAddress'.
There are 19614 rows in 110 pages for object "Person.BusinessEntityAddress".
DBCC results for 'sys.sqlagent_jobs'.
There are 0 rows in 0 pages for object "sys.sqlagent_jobs".
DBCC results for 'Person.BusinessEntityContact'.
There are 909 rows in 6 pages for object "Person.BusinessEntityContact".
DBCC results for 'sys.sqlagent_jobsteps'.
There are 0 rows in 0 pages for object "sys.sqlagent_jobsteps".
(...)
DBCC results for 'Sales.CreditCard'.
There are 19118 rows in 187 pages for object "Sales.CreditCard".
DBCC results for 'Sales.Currency'.
There are 105 rows in 1 pages for object "Sales.Currency".
DBCC results for 'Sales.CurrencyRate'.
There are 13532 rows in 96 pages for object "Sales.CurrencyRate".
DBCC results for 'Sales.PersonCreditCard'.
There are 19118 rows in 60 pages for object "Sales.PersonCreditCard".
DBCC results for 'Sales.SalesOrderDetail'.
There are 121317 rows in 1233 pages for object "Sales.SalesOrderDetail".
(...)
DBCC results for 'HumanResources.JobCandidate'.
There are 13 rows in 11 pages for object "HumanResources.JobCandidate".
DBCC results for 'sys.filestream_tombstone_2073058421'.
There are 0 rows in 0 pages for object "sys.filestream_tombstone_2073058421".
DBCC results for 'sys.syscommittab'.
There are 0 rows in 0 pages for object "sys.syscommittab".
DBCC results for 'HumanResources.Shift'.
There are 3 rows in 1 pages for object "HumanResources.Shift".
DBCC results for 'sys.filetable_updates_2105058535'.
There are 0 rows in 0 pages for object "sys.filetable_updates_2105058535".
DBCC results for 'Person.Address'.
There are 19614 rows in 339 pages for object "Person.Address".
CHECKDB found 0 allocation errors and 0 consistency errors in database 'AdventureWorks'.
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

Completion time: 2022-10-14T07:43:41.0487877+00:00




```

Internal Doc Reference

- [Data Integrity in Azure SQL Database](#)

Public Doc Reference

- [DBCC CHECKDB](#) 

- [DBCC CHECKTABLE \(Transact-SQL\)](#), 
- [Error 823](#) 
- [Error 824](#) 
- [Error 825](#) 
- [Error 9003](#) 
- <https://www.sqlskills.com/blogs/paul/category/checkdb-from-every-angle/> 

How good have you found this content?

