Root Cause Analysis

TABLE OF CONTENTS

INTRODUCTION.................................................................................................................................. 2

INVESTIGATIVE TEAM AND METHOD........................................................................................ 2

FINDINGS AND ROOT CAUSE......................................................................................................... 3

CORRECTIVE ACTION..................................................................................................................... 4

PREVENTING MEATURES............................................................................................................... 7

REFERENCES.……………................................................................................................................. 8

1

# INTRODUCTION

This document is mainly describing the root cause of corruption issue happened on the data pages and

index pages raised on database “WSODB10” on server M03WSO3 on 12.02.2024.It also

include the general trouble shooting activities followed to resolve the issue. There are two parts of the

troubleshooting. One to troubleshoot the corruption on index pages and the other is corruption on the data

pages. Apart from general troubleshooting activities, preventing measures also provided.

# INVESTIGATIVE TEAM AND METHOD

|  |  |
| --- | --- |
| Name Of the Technician | Action Performed. |
| HCL Team | Initial tracing of corrupted index & data pages on DB. |
| HCL Team | Fixed corrupted index pages. |
| HCL Team | Resolved corrupted data pages by restoring the healthiest backup. |
|  |  |

Team members involved in identifying the issue and helped in resolving:

FINDINGS AND ROOT CAUSE

General corruption is caused by errors and failures in storage media and\or power supply. Hard disks, like

all mechanical devices, will fail someday for some reason. Solid state storage is not mechanical, but it will

also fail eventually. The most common cause of corruption for both mechanical and solid state media is a

power failure during a write to disk. Consistency errors are also main reason for database corruption.

Error display:

The operating system returned error 1(Incorrect function.) to SQL Server during a read at

offset 0x000052d0830000 in file 'G: data\ms\adventure\_works\_data.mdf'. 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.

Warning: Fatal error 823 occurred XXXXXX . Note the error and time, and

contact your system administrator.

**The 823 error message contains the following information:**

1. The database file against which the I/O operation was performed
2. The offset within the file where the I/O operation was attempted
3. Is the I/O operation a read or write request.
4. The Operating System Error code and Error Description

Note:

The 823 error message usually indicates that there is a problem with underlying storage system or the

hardware or a driver that is in the path of the I/O request. You can encounter this error when there are

inconsistencies in the file system or if the database file is damaged.

CORRECTIVE ACTION

Action Points:

1. First make sure the User logins on that database are minimal/zero. Then start the consistency check to find

if any errors on page level/index level.

Run the Below Query first

|  |  |
| --- | --- |
| |  | | --- | | DBCC CHECKTABLE (‘Schema. Table’) with ESTIMATEONLY | |

Note: It will populate how much Temp space it will occupy to run the consistency check on that DB.

2. Then check the Consistency check on that DB.

Note: If we already have the Log file of the DBCC, we can minimize the Load on the DB and can run

|  |  |
| --- | --- |
| |  | | --- | | DBCC CHECKTABLE (‘Schema. Table’) with PHYSICAL\_ONLY.  DBCC CHECKDB(‘DBNAME’) with ALL\_ERRORMSGS | |

Note: It will take almost an hour or so to display the error pages of the tables for entire database.

3. The output of the ‘DBCC CHECKDB’ Command:

Pick the Corrupted pages on the table by considering Page ID.

4. Before Troubleshooting these pages confirm whether these pages are Data Pages (or) Index Pages.

5. To find out the page type, we need to run theDBCC PAGE Command.

Note: To get results from DBCC PAGE you must enable trace flag 3604, otherwise the results go to the

SQL server log.

|  |  |
| --- | --- |
| |  | | --- | | DBCC Traceon (3604)  DBCC page (dbid , filenum , page\_number , 0) | |

EX: Here Filenum is 1 and Pagenum is 43817920 from the above screen shot.

DBCC page(dbid, 1, 43817920, 0)

Output of the above command:

Consider m\_type=2, It is a Index page

For more info regarding page type, see http://www.sqlnotes.info/2011/10/31/page-type/

6. If it is an Index page, we need to rebuild the index. Otherwise jump to Step-8.

Pick the Object id of that index page from “DBCC PAGE “Command.

|  |
| --- |
| EXEC sp\_helpindex 'TABLE NAME’SELECTOBJECT\_NAME(object\_id) |

Take the index name from it. Then start rebuilds it.

7. Make sure the database is ideal before rebuilding the index. Because if the table is currently in use it may block and also report errors in application Portal.

|  |
| --- |
| ALTER INDEX[INDEX\_NAME] ON [TABLE NAME] REBUILD |

8. If data page is corrupted, we need to identify this page also by DBCC Page Command.

Later we must identify How many records Got Corrupted. This is a **Key Point**. Use the Best possible ways to identify the missing records.

|  |
| --- |
| Find the Primary Index Column if available This helps in identifying corrupted records.  Select \* from [Schema].[Table\_Name] This Query will Not give you the result.  **Fetch the Count of the records:**  SELECT SCHEMA\_NAME(t.schema\_id) SchemaName,  t.[name] TableName,  SUM(p.row\_count) TotalRows  FROM sys.tables t  INNER JOIN sys.dm\_db\_partition\_stats p  ON t.object\_id = p.object\_id  AND t.type\_desc = 'USER\_TABLE'  AND p.index\_id IN (0,1)  WHERE t.[name] = 'tablename' --- Provide the table name HERE  GROUP BY t.schema\_id, t.[name]  ORDER BY TotalRows DESC  **Fetch the Top count of the records:**  Select top 10000000 \* from [schema].[TableName] If this Query results, So we can still move ahead and no corruption .  Select top 15000000 \* from [schema].[TableName] If this Query results, So we can still move ahead and no corruption .  Select top 17000000 \* from [schema].[TableName] If this Query results, So we can still move ahead and no corruption .  Select top 20000000 \* from [schema].[TableName] If this Query NOT resulted, So it have corruption from after the 17000000 and before 20000000 .  What if the healthy records are available in between the Corrupted Pages.    Select top 1000000 \* from [schema].[TableName] where Column > 17000000 If this Query NOT resulted, So it have corruption from after the 17000000 and before 18000000 .  Note: Like Wise we must identify the Corrupted records.  Once the Missing\Corrupted Records are identified , we have follow the Next step to address this issue. |

9. There are two ways to resolve the issue.

A. To run DBCC Checkdb With Allow Data Loss

B. Restore the up to date existing backups.

10. To use DBCC allow data loss follow the below steps to get the corrupted database online.

A. EXEC sp\_resetstatus ‘yourDBname’

B. ALTER DATABASE yourDBname SET EMERGENCY

C. DBCC checkdb(’yourDBname’)

D. ALTER DATABASE yourDBname SET SINGLE\_USER WITH ROLLBACK IMMEDIATE

E. DBCC CheckDB (’yourDBname’, REPAIR\_ALLOW\_DATA\_LOSS)

F. ALTER DATABASE yourDBname SET MULTI\_USER

2