Week 4 Challenge Details

The database name is [CorruptionChallenge4], it was created and backed up on SQL Server 2008R2. Sorry, no SQL Server 2005 this week due to features in the challenge that didn’t exist in SQL Server 2005 (perhaps that a hint).

Here is what we know. Everything was just fine, and all of a sudden users started reporting the following error when selecting from the database:

or this message.

As usual, you aren’t given much info, but you are able to reproduce this error on the [CorruptionChallenge4] database.

When you run DBCC CheckDB you get the following:

```
Msg 2534, Level 16, State 2, Line 1
Table error: page (3:22), whose header indicates that it is allocated to object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data), is allocated by another object.
Msg 2534, Level 16, State 2, Line 1
Table error: page (3:25), whose header indicates that it is allocated to object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data), is allocated by another object.
Msg 2511, Level 16, State 2, Line 1
Table error: Object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data). Keys out of order on page (3:25), slots 8 and 9.
Msg 2534, Level 16, State 2, Line 1
Table error: page (3:26), whose header indicates that it is allocated to object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data), is allocated by another object.
Msg 2511, Level 16, State 2, Line 1
Table error: Object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data). Keys out of order on page (3:26), slots 18 and 19.
Msg 2534, Level 16, State 2, Line 1
Table error: page (3:27), whose header indicates that it is allocated to object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data), is allocated by another object.
Msg 2511, Level 16, State 2, Line 1
Table error: Object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data). Keys out of order on page (3:27), slots 27 and 28.
Msg 2534, Level 16, State 2, Line 1
Table error: page (3:28), whose header indicates that it is allocated to object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data), is allocated by another object.
Msg 2511, Level 16, State 2, Line 1
Table error: Object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data). Keys out of order on page (3:28), slots 41 and 42.
Msg 2534, Level 16, State 2, Line 1
Table error: page (3:29), whose header indicates that it is allocated to object ID 29, index ID 1, partition ID 281474978611200, alloc unit ID 1900544 (type In-row data), is allocated by another object.
Msg 2511, Level 16, State 2, Line 1
```

Followed by around 17,000 lines of additional errors.

Your job should you accept it is to eliminate the corruption in the database without losing any data.

How will you know when you are successful? That’s part of the challenge, figuring out how to know if you succeeded or not.

Is it possible?

I would like to state at this point that it is entirely possible to recover 100% of the data. Also 100% of the data can be recovered without the dirty tricks of DBCC WritePage (that’s a clue, you don’t need it). When I tested the recovery I was able to do it on SQL Server 2008R2, and SQL Server 2014 to confirm that it works, it should work on any SQL Server 2008R2 or newer, however I didn’t test it on Azure.

What you are given:

The full backup file after the corruption occurred. That’s it this week, let’s just say this database didn’t have the best backup and restore schedule in place. There is one backup file, a full backup containing the corruption. I wish I had some transaction logs, and I wish I had a backup from before the corruption, but I don’t. That’s part of making this more diabolical than the last challenge.

Download the corrupt database files here: CorruptionChallenge4.zip

When restoring, you may need to use the **KEEP_CDC** flag on your restore.

Requirements:

- SQL Server 2008R2 or newer
- SQL Server Enterprise Edition, Evaluation Edition, or Developer Edition

If you don’t have one of these versions, you can download the evaluation version of SQL Server 2014 from Microsoft.

How to know if you succeed

When you think you have success, there are four additional items that I need beyond the typical steps, and proof. These are:

- What version of SQL Server did you use to fix the problem?
- What is at the row in the corrupt database table with an id of 510900 and 510901?
- How many rows are in the table that was corrupt?
- How many user objects are in the [CorruptionChallenge4] database.

The reason I have added these requirement is to make it possible for me to judge your solution.

In order to win you must also provide the usual:

- Detailed steps with what you did to complete the challenge.
- Proof that you removed the corruption, and that you didn’t lose any data.
- Provide a link to the corruption challenge url on Twitter, Facebook or Linked in (http://SteveStedman.com/corruption)

The first person to submit a winning solution with all this info will receive 2 points in the challenge; all those who submit correct answers who are not first will receive 1 point.

Mail your results to

before Monday 4th, 2015 at 8:00am pacific time. Results will be posted soon after that.

Extra Clue

For the first time ever in the Database Corruption Challenge, several people have asked for a clue. So I have created a clue that is intended to help, but not give away the solution.

SPOILER ALERT:::: To view the clue click here.

Related Links

- Current Scores

- Newsletter Sign up
- Database Corruption Challenge T-Shirt
- Week 1, Week 2, Week 3
- Database Corruption Worksheet

Database Corruption Challenge Sponsored by