SUMMIT2017

PASS

Basics of Database Corruption Repair

When Corruption Strikes, will you be ready?

Steve Stedman, Managing Technology Partner,
SQL Data Partners

Please silence cell phones

Explore everything PASS has to offer

24HOURS OF PASS

Free online webinar events

LOCAL GROUPS

Local user groups around the world

SQLSATURDAY PASS

Free 1-day local training events

VIRTUAL GROUPS

Online special interest user groups

BUSINESS ANALYTICS DAY PASS

Business analytics training

PASS VOLUNTEERS

Get involved

Free Online Resources

PASS Blog

White Papers

Session Recordings

Newsletter

PASS Connector

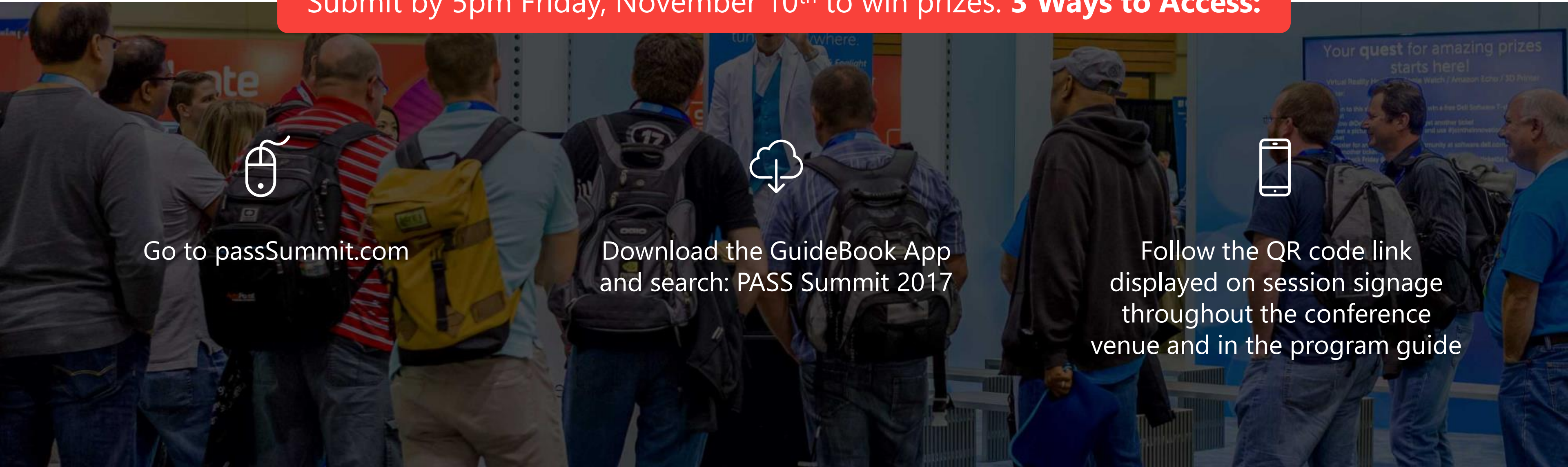
BA Insights

www.pass.org

Session evaluations

Your feedback is important and valuable.

Submit by 5pm Friday, November 10th to win prizes. **3 Ways to Access:**



Go to passSummit.com

Download the GuideBook App and search: PASS Summit 2017

Follow the QR code link displayed on session signage throughout the conference venue and in the program guide

Steve Stedman

Managing Technology Partner
at SQL Data Partners

/in/SteveStedman

@SqlEmt

SQL Data Partners

Managing Technology Partner and Podcast Co-Host

Doing SQL Server performance tuning, corruption repair and general DBA tasks.

SQL Server Experience

Using SQL Server for 27 years

Creator of the Database Health Monitor

Founder of the Database Corruption Challenge

Blog regularly at http://SteveStedman.com

Bellingham SQL Server Users Group

Chapter founder and chapter leader.

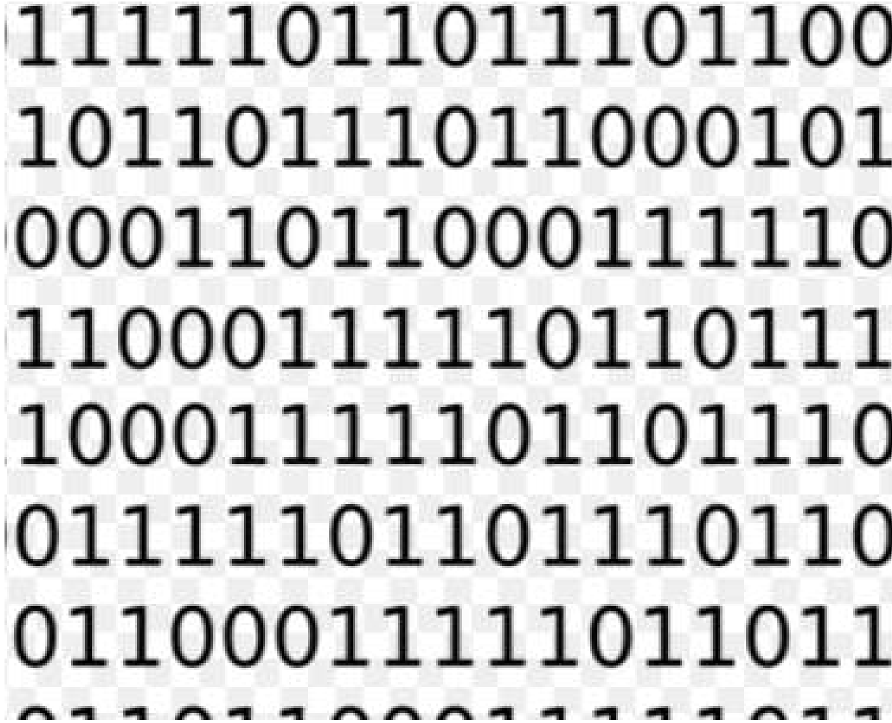
Agenda

Basics of Database Corruption Repair

- What is Database Corruption
- Causes of Corruption
- Detecting Corruption
- Tracking Corruption
- Removing Corruption
- Customer Examples
- Demo

What is Database Corruption?

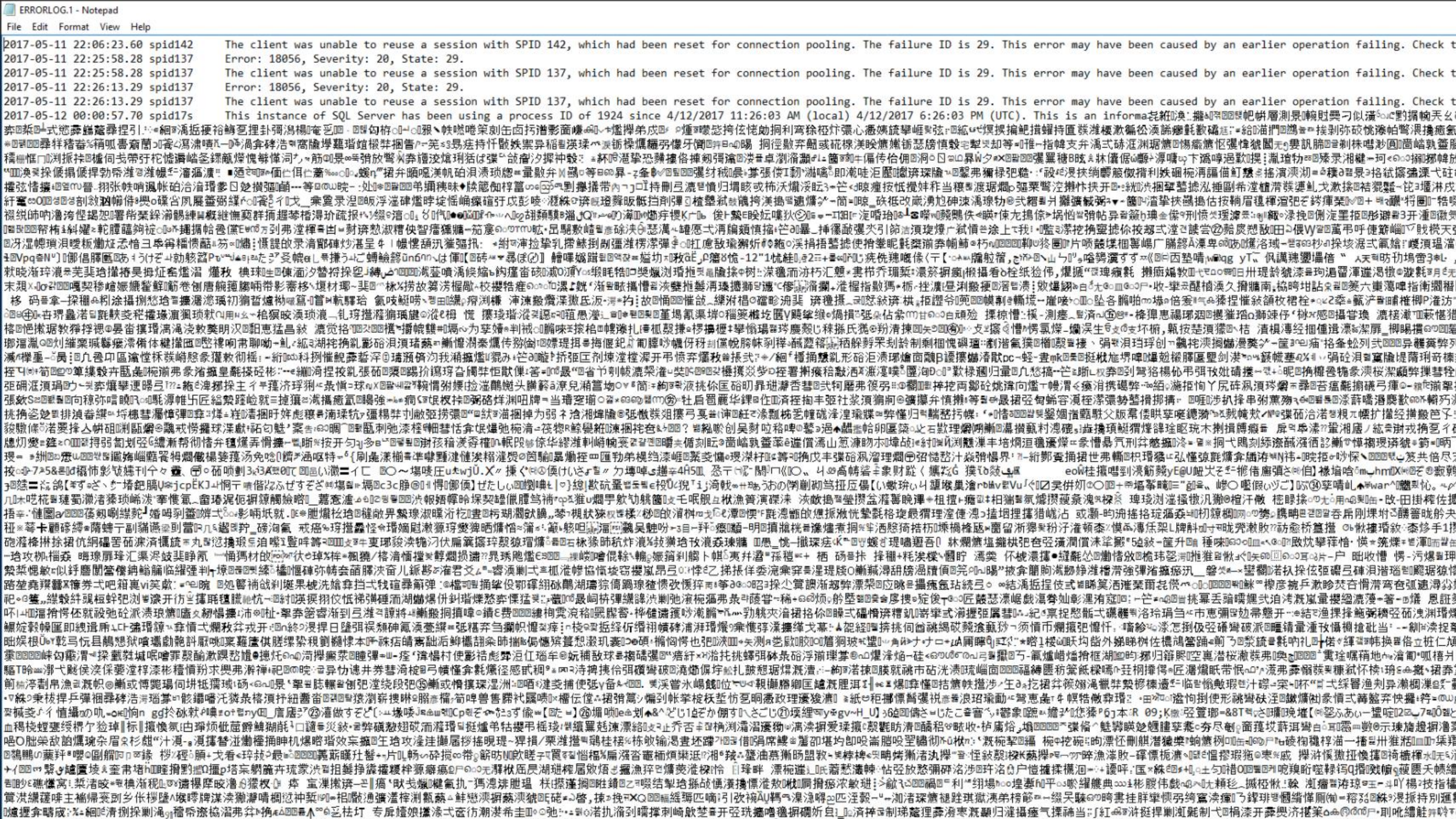
- Pages in the database that are incorrectly formatted.
  - This could be as simple as a single bit, or as huge as the entire file.
- Sometimes prevents the database from starting.
- Sometimes prevents queries from running.
- Sometimes presents as missing or incorrect data.



Causes of Database Corruption

- Drive / Storage Failure – example drive replacement in RAID array.
- Power Outage – While database pages are being written.
- Network issues for network attached storage.

- Most of the time it is problems with I/O.


ERRORLOG.1 - Notepad

File Edit Format View Help

```
2017-05-11 22:06:23.60 spid142     The client was unable to reuse a session with SPID 142, which had been reset for connection pooling. The failure ID is 29. This error may have been caused by an earlier operation failing. Check t
2017-05-11 22:25:58.28 spid137     Error: 18056, Severity: 20, State: 29.
2017-05-11 22:25:58.28 spid137     The client was unable to reuse a session with SPID 137, which had been reset for connection pooling. The failure ID is 29. This error may have been caused by an earlier operation failing. Check t
2017-05-11 22:26:13.29 spid137     Error: 18056, Severity: 20, State: 29.
2017-05-11 22:26:13.29 spid137     The client was unable to reuse a session with SPID 137, which had been reset for connection pooling. The failure ID is 29. This error may have been caused by an earlier operation failing. Check t
2017-05-12 00:00:57.70 spid17s     This instance of SQL Server has been using a process ID of 1924 since 4/12/2017 11:26:03 AM (local) 4/12/2017 6:26:03 PM (UTC). This is an informa[illegible]
[illegible]
```

Confusion With Database Corruption

A full backup and restore of a corrupt database may help fix the corruption.

- FALSE. When you do a full back up a database, the corruption is backed up also.

Rebooting the SQL Server may help with the corruption.

- FALSE. Once the file is corrupt a reboot will not help. It may make things worse.

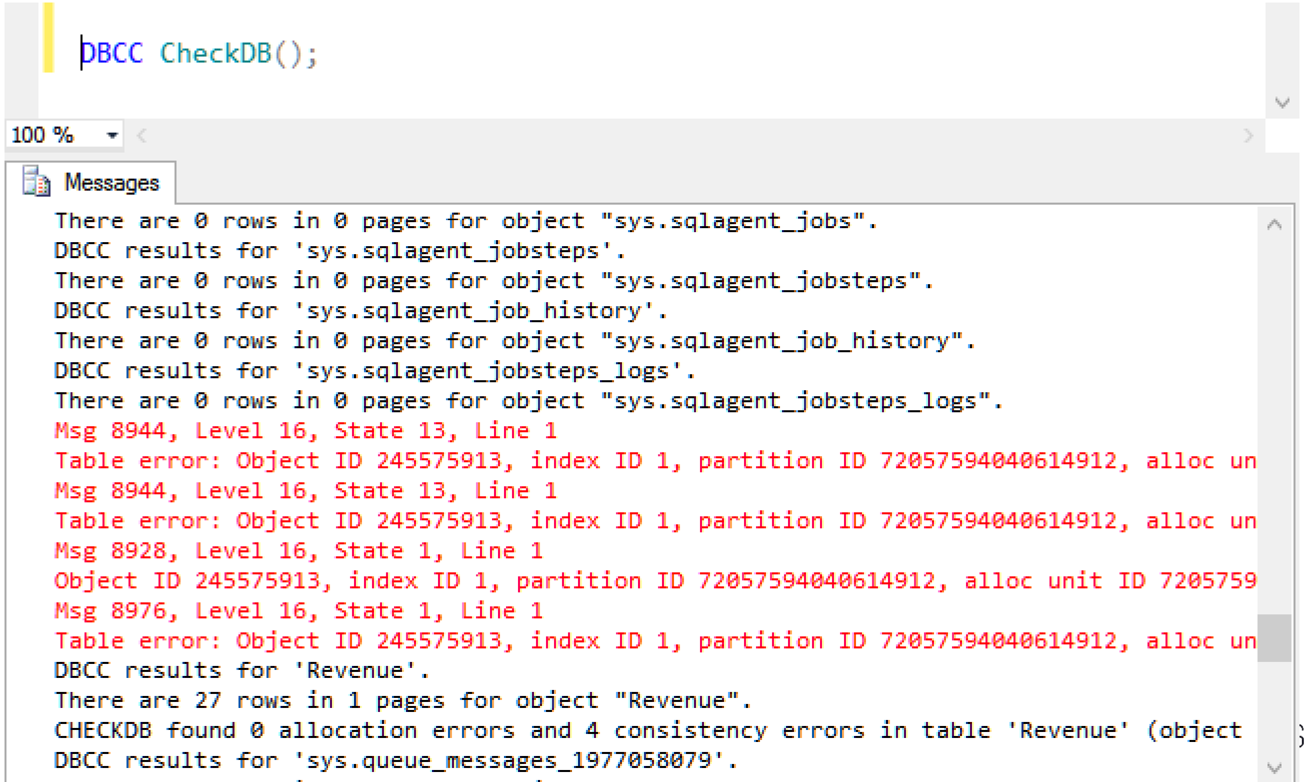
If I just ignore the corruption it may go away or fix itself.

- UNLIKELY. If your regular process truncates the table with the corruption, then it will go away… Otherwise, very unlikely.

PASS

Detecting Corruption

- DBCC CheckDB



```
DBCC CheckDB();
```

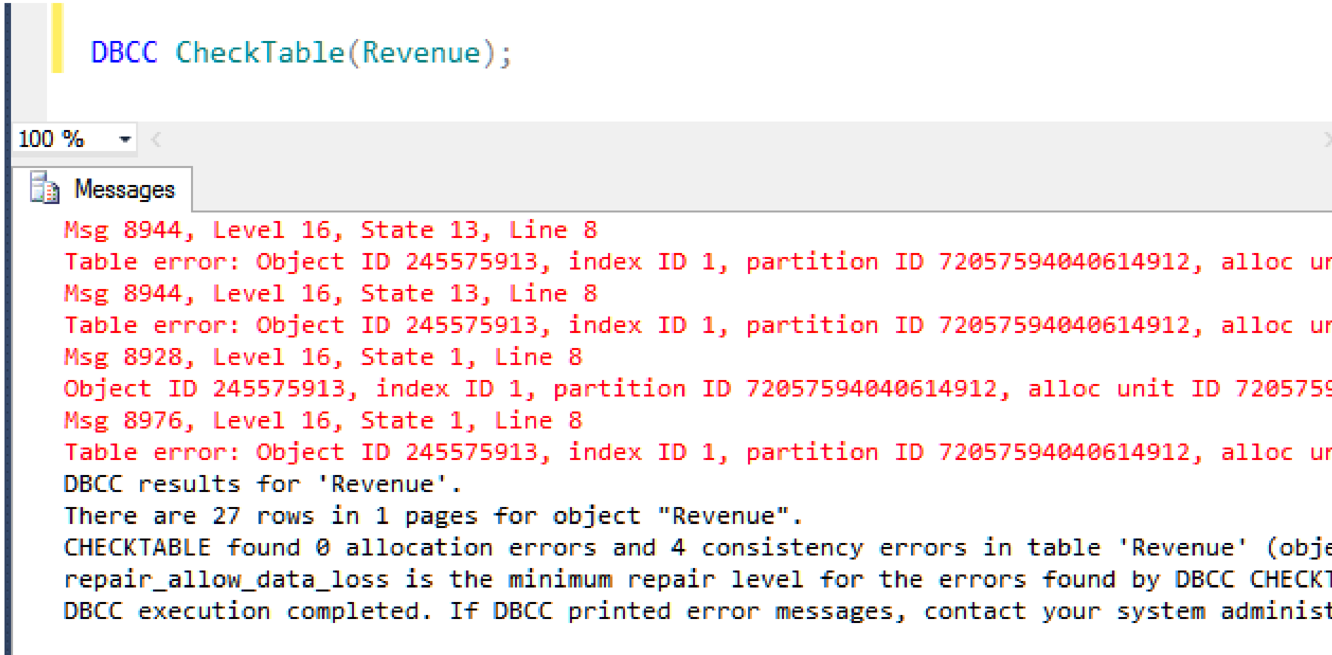
100 %

Messages

```
There are 0 rows in 0 pages for object "sys.sqlagent_jobs".
DBCC results for 'sys.sqlagent_jobsteps'.
There are 0 rows in 0 pages for object "sys.sqlagent_jobsteps".
DBCC results for 'sys.sqlagent_job_history'.
There are 0 rows in 0 pages for object "sys.sqlagent_job_history".
DBCC results for 'sys.sqlagent_jobsteps_logs'.
There are 0 rows in 0 pages for object "sys.sqlagent_jobsteps_logs".
Msg 8944, Level 16, State 13, Line 1
Table error: Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc un
Msg 8944, Level 16, State 13, Line 1
Table error: Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc un
Msg 8928, Level 16, State 1, Line 1
Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc unit ID 7205759
Msg 8976, Level 16, State 1, Line 1
Table error: Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc un
DBCC results for 'Revenue'.
There are 27 rows in 1 pages for object "Revenue".
CHECKDB found 0 allocation errors and 4 consistency errors in table 'Revenue' (object
DBCC results for 'sys.queue_messages_1977058079'.
```

Detecting Corruption

- DBCC CheckDB
- DBCC CheckTable



```
DBCC CheckTable(Revenue);
```

100 %

Messages

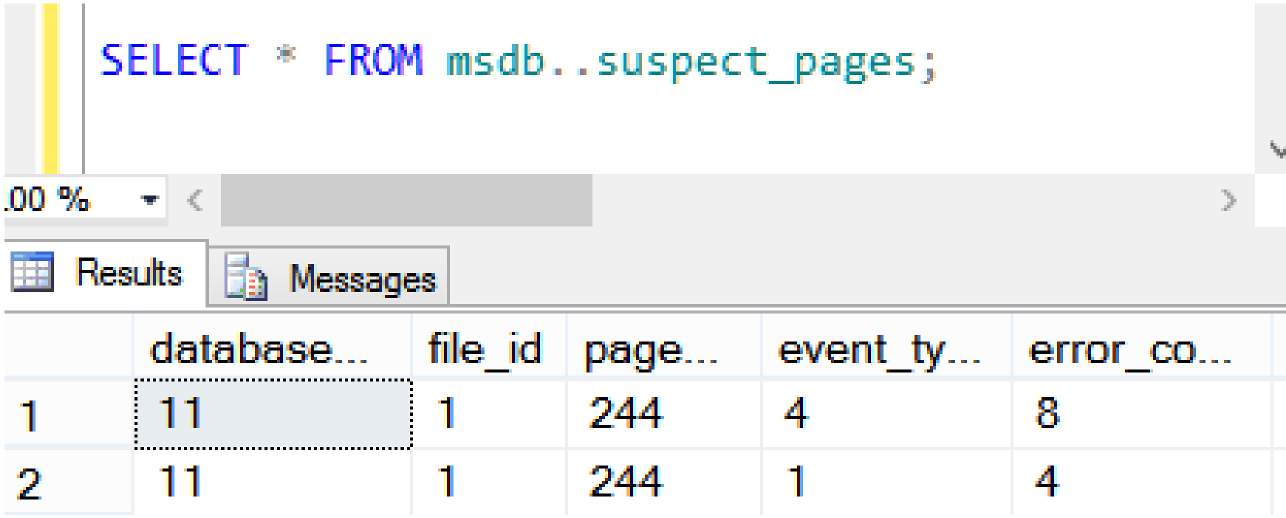
```
Msg 8944, Level 16, State 13, Line 8
Table error: Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc ur
Msg 8944, Level 16, State 13, Line 8
Table error: Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc ur
Msg 8928, Level 16, State 1, Line 8
Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc unit ID 720575
Msg 8976, Level 16, State 1, Line 8
Table error: Object ID 245575913, index ID 1, partition ID 72057594040614912, alloc ur
DBCC results for 'Revenue'.
There are 27 rows in 1 pages for object "Revenue".
CHECKTABLE found 0 allocation errors and 4 consistency errors in table 'Revenue' (obje
repair_allow_data_loss is the minimum repair level for the errors found by DBCC CHECKT
DBCC execution completed. If DBCC printed error messages, contact your system administ
```

Detecting Corruption

- DBCC CheckDB
- DBCC CheckTable
- DBCC Check_____
  (Constraints, Catalog,
  Alloc, FileGroup, Ident)

Detecting Corruption

- DBCC CheckDB
- DBCC CheckTable
- DBCC Check_____
- msdb..suspect_pages



```
SELECT * FROM msdb..suspect_pages;
```

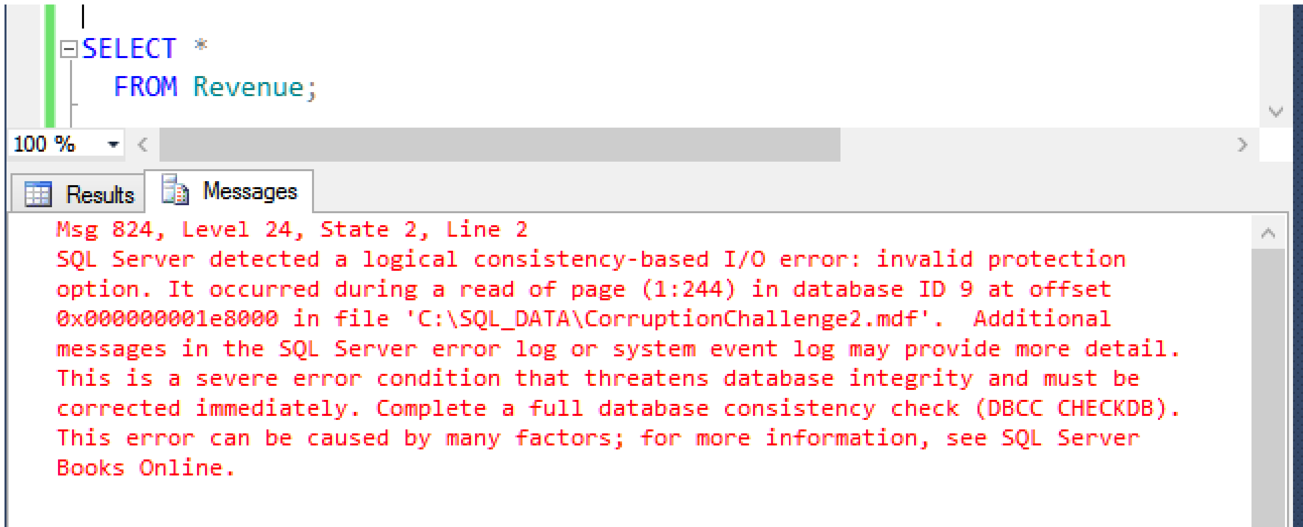
.00 %

Results | Messages

	database...	file_id	page...	event_ty...	error_co...
1	11	1	244	4	8
2	11	1	244	1	4

Detecting Corruption

- DBCC CheckDB
- DBCC CheckTable
- DBCC Check_____
- msdb..suspect_pages
- Just running a query may show corruption.

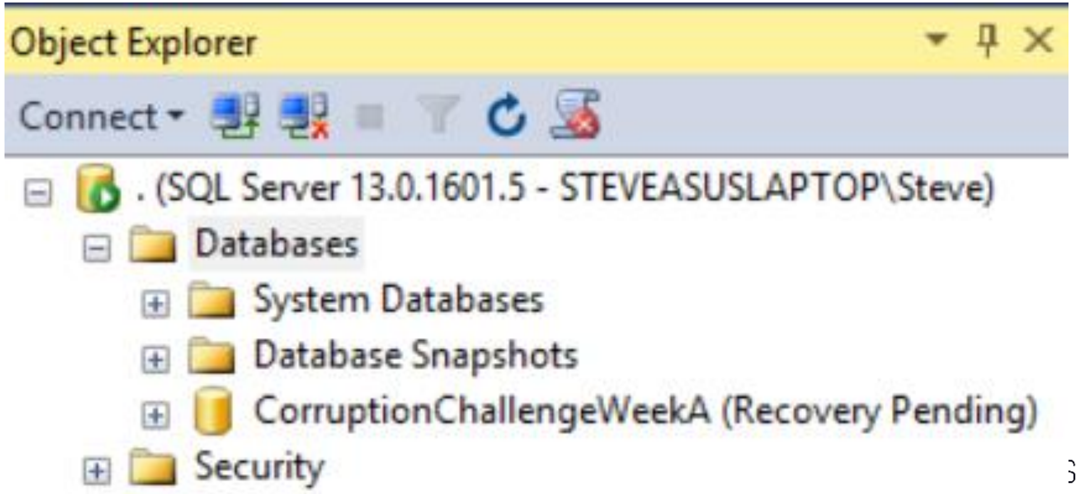


```
SELECT *
  FROM Revenue;
```

```
Msg 824, Level 24, State 2, Line 2
SQL Server detected a logical consistency-based I/O error: invalid protection
option. It occurred during a read of page (1:244) in database ID 9 at offset
0x000000001e8000 in file 'C:\SQL_DATA\CorruptionChallenge2.mdf'.  Additional
messages in the SQL Server error log or system event 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.
```

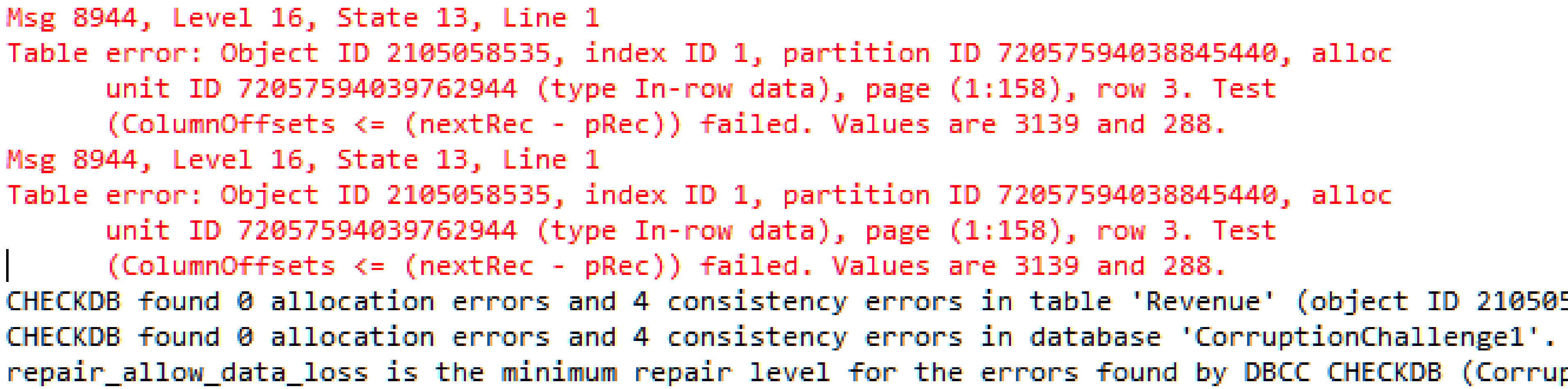
Detecting Corruption

- DBCC CheckDB
- DBCC CheckTable
- DBCC Check_____
- msdb..suspect_pages
- Just running a query may show corruption.
- Recovery Pending or Suspect


Object Explorer
Connect
. (SQL Server 13.0.1601.5 - STEVEASUSLAPTOP\Steve)
Databases
System Databases
Database Snapshots
CorruptionChallengeWeekA (Recovery Pending)
Security

Tracking Corruption
(what has gone bad?)

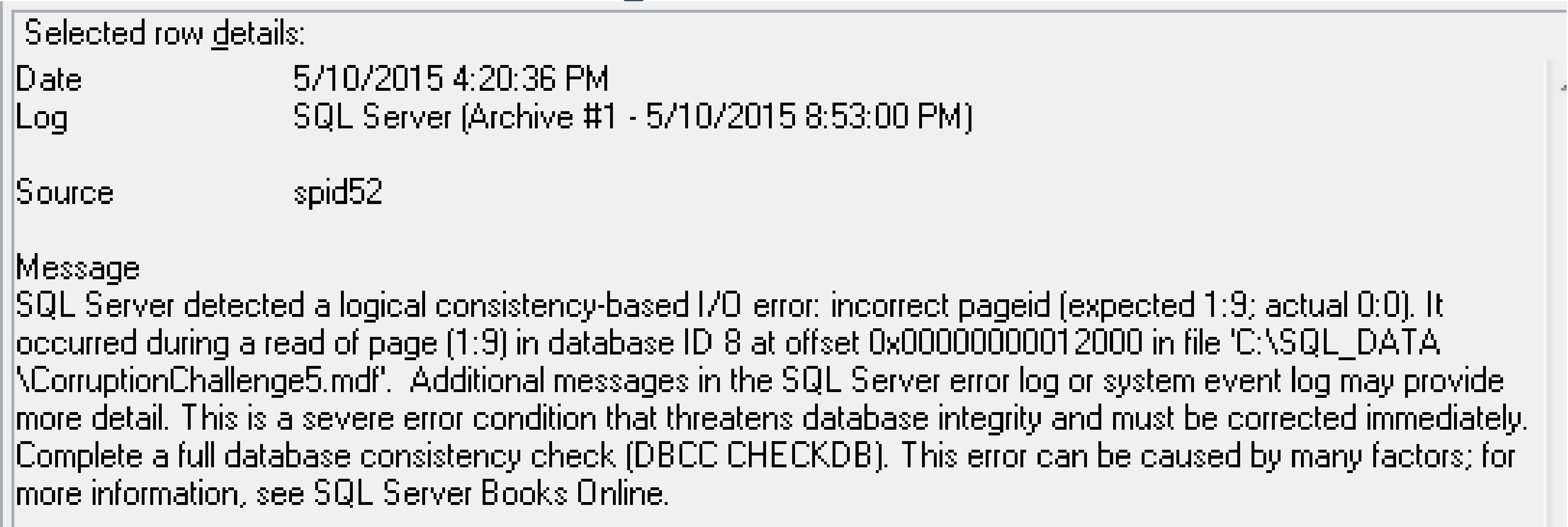
- Check error messages - focus on the red.



```
Msg 8944, Level 16, State 13, Line 1
Table error: Object ID 2105058535, index ID 1, partition ID 72057594038845440, alloc
      unit ID 72057594039762944 (type In-row data), page (1:158), row 3. Test
      (ColumnOffsets <= (nextRec - pRec)) failed. Values are 3139 and 288.
Msg 8944, Level 16, State 13, Line 1
Table error: Object ID 2105058535, index ID 1, partition ID 72057594038845440, alloc
      unit ID 72057594039762944 (type In-row data), page (1:158), row 3. Test
      (ColumnOffsets <= (nextRec - pRec)) failed. Values are 3139 and 288.
CHECKDB found 0 allocation errors and 4 consistency errors in table 'Revenue' (object ID 210505
CHECKDB found 0 allocation errors and 4 consistency errors in database 'CorruptionChallenge1'.
repair_allow_data_loss is the minimum repair level for the errors found by DBCC CHECKDB (Corrup
```

Tracking Corruption (what has gone bad?)

- Check the Error Log



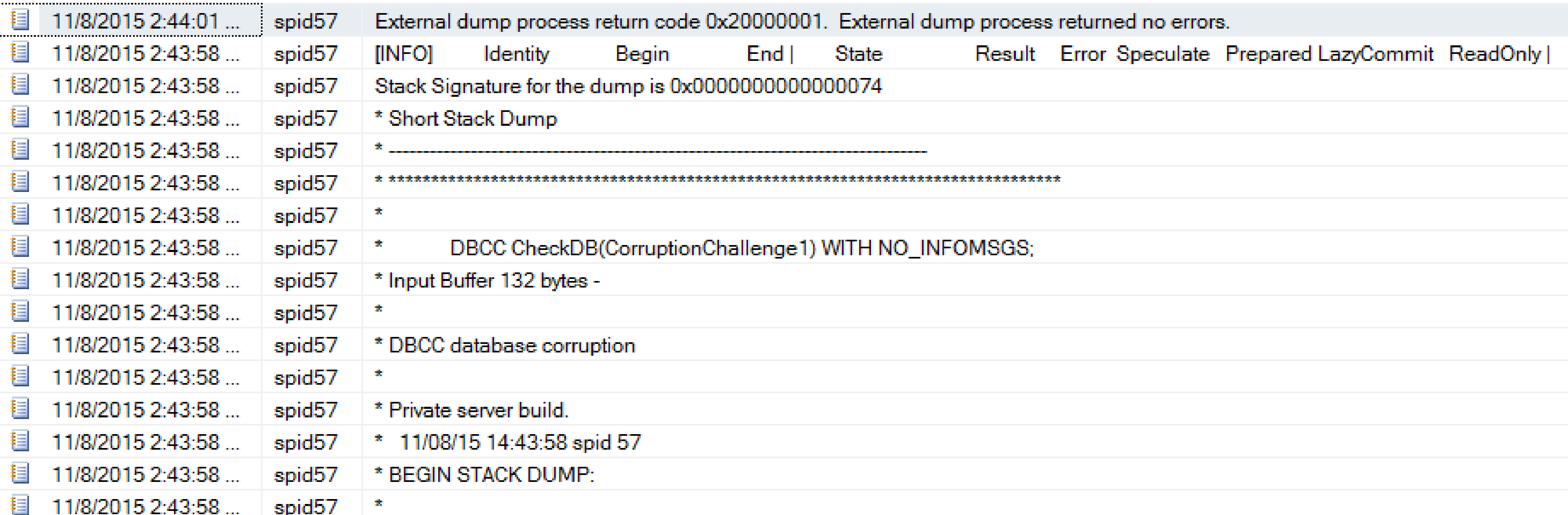
```
Selected row details:
Date                5/10/2015 4:20:36 PM
Log                 SQL Server (Archive #1 - 5/10/2015 8:53:00 PM)

Source              spid52

Message
SQL Server detected a logical consistency-based I/O error: incorrect pageid (expected 1:9; actual 0:0). It
occurred during a read of page (1:9) in database ID 8 at offset 0x00000000012000 in file 'C:\SQL_DATA
\CorruptionChallenge5.mdf'.  Additional messages in the SQL Server error log or system event 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.
```

Tracking Corruption (what has gone bad?)

- Check the Error Log



```
11/8/2015 2:44:01 ...   spid57   External dump process return code 0x20000001.  External dump process returned no errors.
11/8/2015 2:43:58 ...   spid57   [INFO]      Identity        Begin          End |      State            Result     Error  Speculate  Prepared LazyCommit  ReadOnly |
11/8/2015 2:43:58 ...   spid57   Stack Signature for the dump is 0x0000000000000074
11/8/2015 2:43:58 ...   spid57   * Short Stack Dump
11/8/2015 2:43:58 ...   spid57   * -------------------------------------------------------------------------------
11/8/2015 2:43:58 ...   spid57   * *******************************************************************************
11/8/2015 2:43:58 ...   spid57   *
11/8/2015 2:43:58 ...   spid57   *           DBCC CheckDB(CorruptionChallenge1) WITH NO_INFOMSGS;
11/8/2015 2:43:58 ...   spid57   * Input Buffer 132 bytes -
11/8/2015 2:43:58 ...   spid57   *
11/8/2015 2:43:58 ...   spid57   * DBCC database corruption
11/8/2015 2:43:58 ...   spid57   *
11/8/2015 2:43:58 ...   spid57   * Private server build.
11/8/2015 2:43:58 ...   spid57   *   11/08/15 14:43:58 spid 57
11/8/2015 2:43:58 ...   spid57   * BEGIN STACK DUMP:
11/8/2015 2:43:58 ...   spid57   *
```

Tracking Corruption (what has gone bad?)

- See what you can query

```
-- lets see what we have in the corrupt table
SELECT *
  FROM Revenue;
-- 54 rows  Is that the expected number of rows?
```

Tracking Corruption (what has gone bad?)

- Check your non-clustered indexes

Do you have the same number of rows, and same data that the clustered index has?

```
-- pull from the non-clustered index without
--   touching the clustered index
SELECT [id], [DepartmentID], [Revenue]
  FROM Revenue
  WITH (INDEX (ncDeptIdRevenue) );
```

Before Fixing or Removing Corruption

- Do you have a way to start over if something goes wrong?
- Do you have a backup of the current state?
- If your solution is going to cause data loss, can you save anything before causing that data loss?
- Do you have someone to review your ideas before proceeding?

Can I Get a "Do Over"?

What if you go through the whole process, but determine that part of your cleanup lost that could have been saved in the beginning?

PASS

Removing Corruption

Restore from backup, prior to when the corruption was encountered.

- Common solution. Lose data back to the point in time that corruption was encountered
- Not always feasible.
  - Missing Backups.
  - Corruption has been there longer than your backup retention period.
- Early detection is critical for this option to be feasible.

Removing Corruption

Drop/Recreate Index – if corruption is in a non-clustered index

- This is perhaps the easiest corruption to fix.

Updating data in a row when it is a data purity issue.

Removing Corruption

Truncate table – if you have a way to get the contents back

- Copy everything you can to another table.
- Pull what is missing from a backup or non clustered index.
- Fill in the blanks.
- Truncate the table
- Put everything back in.

PASS

Removing Corruption

```
DBCC CheckTable(Revenue, REPAIR_REBUILD);
```

- Rarely does anything

```
DBCC CheckTable(Revenue, REPAIR_ALLOW_DATA_LOSS);
```

- Will cause data loss, but won’t change anything outside of the revenue table.

```
DBCC CheckDB(database1, REPAIR_ALLOW_DATA_LOSS);
```

- Will cause data loss

Demo

Pulling data from a non-clustered index.

Customer Example 1

Customer 1

Complaint: Trouble Running Queries

- When we query a specific table we get errors about a “Table Error”
- How long has it been happening?
  - 3 weeks
- What does it impact?
  - We are not able to complete some orders when it happens
- What have you tried so far?
  - We ran CheckDB, but we haven’t tried the repair_allow_data_loss option.

Customer 1

Initial Conversation: no recent backups

- Do you have backups of that database prior to the corruption?
  - Yes we have a backup from 8 months ago.
- Do you have any more recent backups?
  - No.
- What type of data does this table contain?
  - It contains financial sales records that are needed for our year over year business forecasting and account for tax purposes.

PASS

Customer 1

Urgency: How soon do you need this repaired?

- How soon do you need this recovered?
  - Last week.
- If we can repair this over the next 24 hours would that meet your expectations?
  - The business would prefer this sooner, but if that is our option, then yes 24 hours would be fine.

Customer 1

Investigation: a single page corrupt

- In a single table with millions of rows, a single page was corrupt in the clustered index.
- SELECT * FROM table: returned all rows prior to the corrupt area.
- SELECT * with an order by on the clustering key returned all rows from the end of the table to the corrupt area.
- Examining the page header showed that the corrupt page contained 16 rows.
- Restore from a backup for 8 months prior allowed those 16 rows to be accessed.

Customer 1

Testing:

- When testing on a restored copy of the corrupt database, 2 options were found to remove the corruption.
  - CHECKDB with Repair Allow Data Loss. Removed the one page with 16 rows.
  - Truncate Table, emptied the whole table, and did succeed at removing corruption.
- Examining the contents of the corrupt page is had been overwritten with no chance of recovering the individual rows from the corrupt file.

Customer 1

Proposed Solution

- Review the 16 rows from the 8 month old backup. Confirm that they would not have changed.
- Copy everything from the table that we can.
  - All the rows before the corruption
  - All the rows after the corruption
- Run CHECKDB with Repair Allow Data Loss
- Compare all the rows that we saved before the repair to what we have left in the table
- Pull the 16 rows from the 8 month old backup and insert into the table.

Customer 1

Outcome

- After reviewing the business logic we were able to confirm that the corrupt page had not changed between the 8 month old backup and now.
- Database repair as proposed was run on a copy of the database and tested.
- Then the repair was run on the production database.
- Testing confirmed that after it was repaired everything worked as expected.

- The customer was very happy.

PASS

Customer 1

Further Tasks

- After the corruption was repaired we worked with the client to perform a full server assessment with recommendations like regular backups and regular CheckDB scripts.

Customer Example 2

Customer 2

Complaint: After a power outage, the database is not available.
We rebooted many times, and it always comes up in suspect mode.

- Was the database on a UPS, or battery backup system?
  - No, it was just plugged in under the desk.
- What are you seeing when the server starts up?
  - The database is in Suspect mode
- How long has this been occurring?
  - 2 weeks
- What have you tried so far?
  - Contacted Microsoft, but they say that SQL Server 2005 is no longer supported.

PASS

Customer 2

Initial Conversation: no recent backups

- Do you have any backups of the database?
  - We have no backups
- What is this database used for?
  - Financial data, it is our entire accounting system

PASS

Customer 2

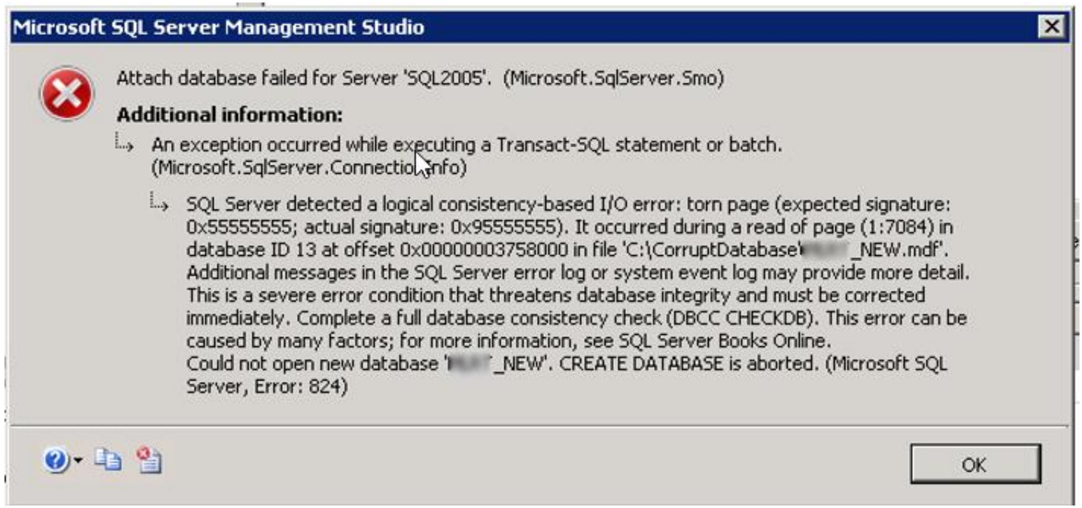
Urgency: How soon do you need this repaired?

- How soon do you need this recovered?
  - As soon as possible.

Customer 2

Investigation: Torn page detected

- In order to not make things any worse on their system, I took a copy of the MDF and LDF files and attempted to work on them in my environment.
- Attaching database threw errors.
- Tried the hack attach method with no luck.



Microsoft SQL Server Management Studio

Attach database failed for Server 'SQL2005'. (Microsoft.SqlServer.Smo)

Additional information:

An exception occurred while executing a Transact-SQL statement or batch. (Microsoft.SqlServer.ConnectionInfo)

SQL Server detected a logical consistency-based I/O error: torn page (expected signature: 0x55555555; actual signature: 0x95555555). It occurred during a read of page (1:7084) in database ID 13 at offset 0x00000003758000 in file 'C:\CorruptDatabase\[redacted]_NEW.mdf'. Additional messages in the SQL Server error log or system event 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.
Could not open new database '[redacted]_NEW'. CREATE DATABASE is aborted. (Microsoft SQL Server, Error: 824)

Customer 2

Investigation: Time for a hex editor

- I was able to determine that page 7084 was part of the table sys.allocation_units.
- Client then pointed out that they did find a backup from 18 months ago.
- We were able to copy the corrupt page from the backup prior to corruption and copy it into the test server.
- The database then came online.

Customer 2

Testing:

- We had no way to determine if everything was accurate in the sys.allocation_units table. The database did come online, but still no idea if the data was accurate.
- Working with Randolph West, he had created a program to extract all of the data out of an MDF file by directly reading the pages. He ran his script on the corrupt database and we compared the data that he had extracted to the data in the repaired database. It was a match.
- Testing confirmed that we had fixed it correctly.

Customer 2

Proposed Solution

- Detach the repaired database from the test environment.
- Copy the MDF and LDF files back to the customers environment.
- Recommended scripting the entire database with data and recreating to get it back to a safer position. They decided not to take this option.

Customer 2

Outcome

- Database was back up and running with no data loss.
- The 2 independent methods of getting it repaired confirmed that nothing was missing.

- Customer was very happy.

Customer 2

Further Tasks

- After the corruption was repaired we worked with the client to get regular backups and CheckDB scripts running.

Demo

Corruption Challenge Week 6

Summary

Basics of Database Corruption Repair

- What is Database Corruption
- Causes of Corruption
- Detecting Corruption
- Tracking Corruption
- Removing Corruption
- Customer Examples
- Demo

PASS

More Examples

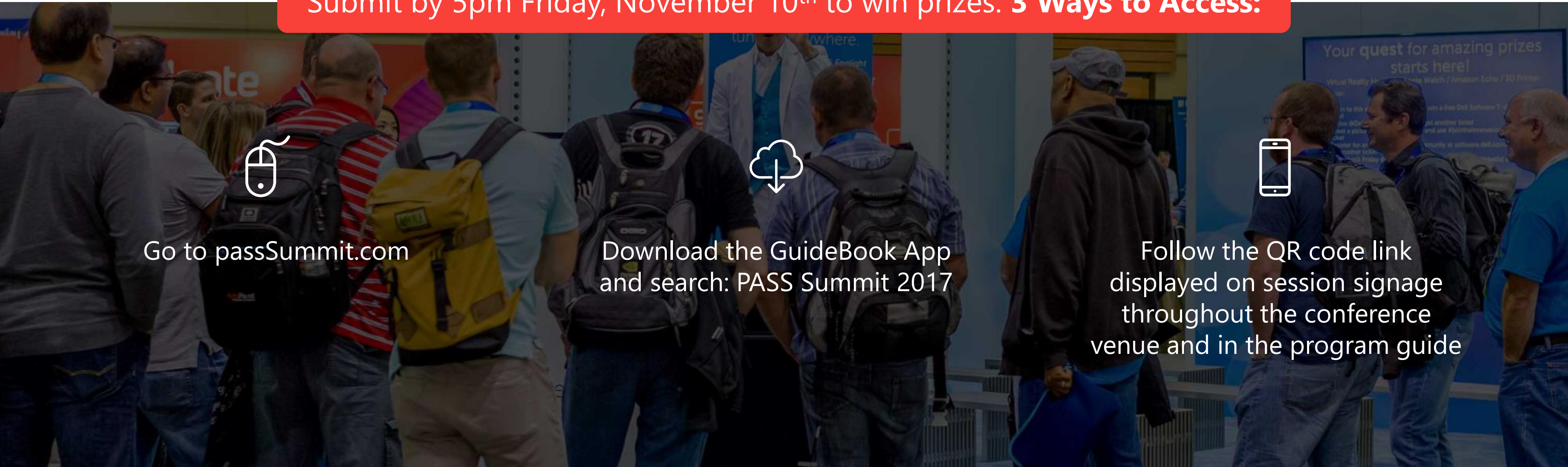
More corruption examples available on my website.

http://SteveStedman.com/Corruption

Session evaluations

Your feedback is important and valuable.

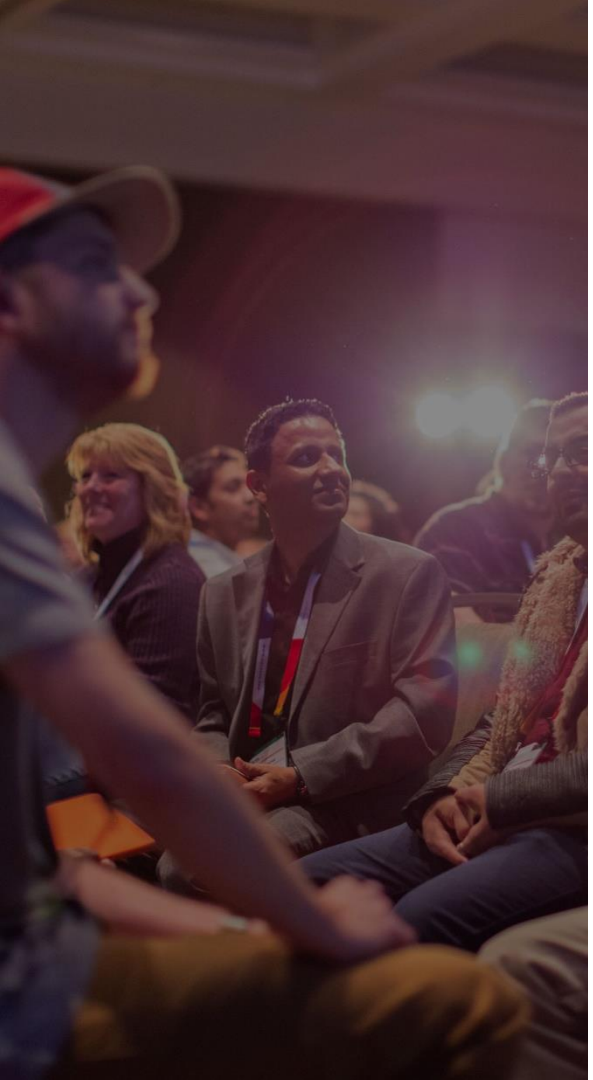
Submit by 5pm Friday, November 10th to win prizes. **3 Ways to Access:**



Go to passSummit.com

Download the GuideBook App and search: PASS Summit 2017

Follow the QR code link displayed on session signage throughout the conference venue and in the program guide



Thank You

Learn more from Steve Stedman

@sqlEmt

Steve@Stedman.us

PASS