Database Disaster Manual

Table of Contents

[Summary 2](#_Toc29981541)

[Scenarios 3](#_Toc29981542)

[Power Outage 4](#_Toc29981543)

[Hard Drive Failure 5](#_Toc29981544)

[Complete Server Failure 6](#_Toc29981545)

[Loss of Known Data 7](#_Toc29981546)

[Loss of Unknown Data 8](#_Toc29981547)

[Loss of Entire Database 9](#_Toc29981548)

[Unauthorized Access 10](#_Toc29981549)

[Contacts 11](#_Toc29981550)

[File Locations 12](#_Toc29981551)

[User Accounts 13](#_Toc29981552)

[Maintenance Outages 14](#_Toc29981553)

[Approvals 15](#_Toc29981554)

[Acknowledgement 16](#_Toc29981555)

# Summary

The purpose of this manual is to be a guide in case of a disaster affecting the database including down-time, data-loss, or security breech.

In the event of a disaster described in this manual, the steps in this manual should be followed as documented, whereas they have been tested and approved. Your first step will be to identify your scenario in the table of contents and then to follow the directions in that section of this document. If the disaster is not described in this manual, you may find the closest scenario available and adapt its steps as needed.

This manual is a sample template provided by The SQL Guy. As you go through this sample, you will find several formatting hints to help you adapt it to your purposes. Red fonts, such as this one, are used for my comments and instructions on how to customize this manual. Once you are done with customizing your template, you may remove all these comments if you wish. The entirety of this manual is a sample. Pieces of it may not apply perfectly to your situation or at all. You should read through each section completely and edit or remove anything which does not apply. Any text with square brackets, such as [sampleemail@sampledomain.com], is text used as a placeholder, and you should definitely edit it. All text should be reviewed for applicability however.

# Scenarios

The following scenarios have been identified, tested, and approved by the organization as being properly documented in this manual. You should identify the scenario you are experiencing and follow the steps outlined in it. If your scenario is not found here, you may adapt the closest scenario to give you some level of guidance.

* Power Outage
* Hard Drive Failure
* Complete Server Failure
* Loss of Known Data
* Loss of Unknown Data
* Loss of Entire Database
* Unauthorized Access

This list of scenarios is a list of common scenarios to start with. You should adapt these scenarios to reflect the risks you see for your organization. This should also be a living document where you regularly review it and add additional scenarios, especially if you encounter one not found here. The steps listed in this template are a starting point, but you should add as much detail as possible. Consider if a junior administrator is on-call when the emergency occurs, how detailed your instructions should be for that person.

## Power Outage

**Definition**

This scenario is for when you have lost power and there are no other known issues.

This guide is making the assumptions that your server has a battery backup.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. Submit an outage report with the power company at [555-555-5555].
3. Shut down the following applications and components to save power:
   1. If network access is still available, turn off the server’s monitor and perform the other operations remotely from a computer not on the same battery.
   2. Microsoft Updates
   3. Identify any software which might use the CPU and is considered non-essential. This might include scheduled tasks, reporting replicas, connections by other users, etc. Do not shut down anything important like anti-virus. It is good to know when your backups run and plan accordingly. Transactional backups may be your primary strategy in recovering from this outage. Full backups are unlikely to finish before the end of the outage, so you will most likely want to stop them. Remember that any operation on the server uses CPU time and battery power, so don’t bother shutting down inconsequential operations, because the act of turning them off might use more power than they would have to begin with.
4. Check estimated battery time remaining and set a reminder for 15 minutes before it runs out.

**Shutdown Steps**

1. Monitor the remaining battery time until 10 minutes of estimated time are remaining.
2. Start a maintenance outage.
3. Perform a transactional backup of the database.
4. Shut down the server.
5. Wait for power to be restored. After power is restored, wait 5 minutes before beginning recovery.

**Recovery**

1. If the server was shut down, restart the server.
2. If any applications were stopped, confirm they are now running.
3. End the Maintenance Outage.
4. Test applications for proper operation.
5. Notify the following people that systems have been restored - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com)]
6. Monitor the server to confirm the battery returns to full power.

## Hard Drive Failure

**Definition**

A single hard drive on the SQL Server has failed, but the RAID array and the database are still operational.

This guide is making the assumptions that your server uses a RAID array with redundancy.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. Configure the RAID controller to replace the failed drive with the hot spare.
3. Run repair on the RAID array.
4. Pause any unnecessary server operations in order to limit I/O usage.
5. Verify integrity of the RAID array and operability of the database.
6. Notify the people in step 1 of the completed repair.
7. Resume any paused operations.
8. Purchase a new hard drive and add it as a hot spare.

## Complete Server Failure

**Definition**

The database server is completely unoperational or is not responding for unknown reasons.

This guide assumes that you have an AlwaysOn availability group.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. In Object Explorer, connect to a server instance that hosts a secondary replica of the availability group that needs to be failed over. Expand the server tree.
3. Expand the AlwaysOn High Availability node and the Availability Groups node.
4. Right-click the availability group to be failed over, and select Failover.
5. Follow the steps of the wizard to complete the failover.
6. Confirm that the database and any connected applications are operational.
7. Notify the people in step 1 of the resolution.

**After the server is repaired or replaced**

1. Add the server to the availability group as a secondary replica.
2. If the server has superior performance to the current active replica, prepare to failover to this server.
3. Notify the people in Immediate Steps Step 1 of your intend to failover, and schedule a maintenance outage.
4. During the maintenance outage, perform the Immediate Steps 2-7 for the new server.

## Loss of Known Data

**Definition**

Specific records have been deleted or modified. The records can be identified, and the remainder of the database is operational and intact.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. Determine whether a maintenance outage should be started for the repair, or if it is safe to allow users to continue to operate during the fix. Start the outage if needed.
3. Determine the cause of the modification and whether it is likely to occur again. Shut down the process or disable the account if needed.

**Repair**

1. Determine the time of the modification.
2. Restore backups to a point-in-time just before the modification to a new test database.
3. Identify the modified data and confirm an acceptable copy has been found. It may be necessary to restore to an older version if the data can’t be found. It may be beneficial to also restore a newer version if not all modifications to the affected data occurred at the same time.
4. If the modifications are small, it may be sufficient to send the affected data to an end user to enter manually. If not, a programmatic repair may be more efficient.
5. If a programmatic repair is selected, restore a 2nd test copy of the database, and apply all fixes to that copy using scripts. Verify that the test copy is an acceptable fix, manually run a transactional backup of production, and then run those same fix scripts against production.
6. Verify all data is correct.
7. If a maintenance outage was started, end the outage and restore user access now.
8. Notify the people in Immediate Steps step 1 that the repair is completed.

## Loss of Unknown Data

**Definition**

Data has been lost, and the scope of the loss is unknown. This may be the result of a modification to unknown records or at an unknown time or with unknown related records or modifications made since the loss or because of it.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. Determine whether a maintenance outage should be started for the repair, or if it is safe to allow users to continue to operate during the fix. Start the outage if needed.
3. Determine the cause of the modification and whether it is likely to occur again. Shut down the process or disable the account if needed.

**Repair**

1. Determine the time of the last known good state of the data. This may require several iterations of restoring and testing backups to a new test database.
2. Restore the most recent available good backup to a new test database.
3. Compare the current production database to the restored database. Determine what changes if any have been made since that backup and should be applied to it. If any need to be made, a 2nd test copy of the database should be created and all changes applied to it using scripts. It may be helpful to restore other backups to identify additional changes to apply. It may be necessary to consider outside data, such as replicas, reports, email receipts, and paper forms. Once all available data has been identified, take a transactional backup of production, and then run the fix scripts against it.
4. Verify all data is correct.
5. If a maintenance outage was started, end the outage and restore user access now.
6. Notify the people in Immediate Steps step 1 that the repair is completed.

## Loss of Entire Database

**Definition**

The database is completely unoperational, and data has been lost or corrupted.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. Start a maintenance outage.

**Repair**

1. Attempt to perform a transactional backup of the database.
2. Restore the most recent full backup WITH NORECOVERY. Be sure to select backing up the tail of the log.
3. Right-click the database, point to Tasks, point to Restore, and then click Transaction Log, which opens the Restore Transaction Log dialog box.
4. On the General page, in the Database list box, select the name of a database.
5. Select all transaction files and restore to the most recent available point in time WITH RECOVERY.
6. Verify all data is correct.
7. End the maintenance outage.
8. Notify the people in Immediate Steps step 1 that the repair is completed.

## Unauthorized Access

**Definition**

It is suspected that someone gained access to data outside their intended permissions level or for unethical purposes.

This guide assumes that your database keeps detailed logs of logins and data access. It also presumes that you have already created scripts to query these data structures.

**Immediate Steps**

1. Notify the following people - [[jdoe@abc.com](mailto:jdoe@abc.com); [fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]
2. Disable any user accounts associated with the access if they are known. This step should be repeated at any point an account is identified.
3. Determine whether a maintenance outage should be started. If the unauthorized access is in-progress or sensitive data has been breeched and the cause has not been identified, this will likely be necessary.
4. Run anti-virus and anti-malware software.
5. Run the script [\\corporate-server\security\unauthorized-access\logins.sql] to determine the users or the suspected user logged in. Save the results in Excel.
6. Run the script [\\corporate-server\security\unauthorized-access\data-access.sql] to determine which data was accessed or modified by the suspected user. Save the results in Excel.
7. Analyze the patterns of usage of this data and attempt to identify users with similar patterns. Rerun the scripts for any identified users and disable their accounts if appropriate.
8. Submit the results of all analysis to the following people - [[fdoe@abc.com](mailto:fdoe@abc.com); [mdoe@abc.com](mailto:mdoe@abc.com)]

**Follow-up Steps**

1. Create automated tasks to monitor for similar behavior in the future and to notify administrators.
2. Attempt to identify how the unauthorized access was possible and implement changes to protect against this in the future.

# Contacts

In the event of an emergency, these are the contacts which may be useful or required to inform or consult.

At a minimum, you should list contact information for anyone who is mandatory to inform of issues, who has responsibility for your databases, and who has expertise which may be useful. If there is an official on-call schedule or day-time/night-time numbers or emails, you may want to include those as well.

**Mandatory Notifications**

You must notify the following people in the event of an outage

[John Doe

Manager of Operations

555-555-5555

[jdoe@abc.com](mailto:jdoe@abc.com)]

**Other Contacts**

[John Doe

Database Administrator

555-555-5555

[jdoe@abc.com](mailto:jdoe@abc.com)]

[John Doe

Webmaster

555-555-5555

[jdoe@abc.com](mailto:jdoe@abc.com)]

[John Doe

Server Administrator

555-555-5555

[jdoe@abc.com](mailto:jdoe@abc.com)]

# File Locations

Files which may be required for various operations are found at the following locations.

**Full Backups**

Full backups are .bak files found at [D:\Program Files\Microsoft SQL Server\MSSQL\Backups\]

**Transaction Log Backups**

Transaction Log backups are .trn files found at [D:\Program Files\Microsoft SQL Server\MSSQL\Backups\]

# User Accounts

Below is a list of some key accounts.

Not knowing a login can halt a process, so it is important to plan for this. In the event of malicious changes, it can be helpful to know who other administrators are, so you know which dangerous accounts to monitor or disable. Storing a variety of use types can be helpful for testing, so you can verify correct operation for other types of users. It is dangerous to store passwords in unprotected documents, so consider implementing a password keeper and referencing it in this document rather than storing the actual password in plain text.

**Administrators**

[Username: jdoe

Owner: John Doe]

# Maintenance Outages

A maintenance outage is a thoughtful way to take systems offline while work is being performed. By disabling access to systems during work, you protect against lost data, a poor user experience, and unpredictable results. Users will be notified of the outage to reassure them.

**Starting a Maintenance Outage**

1. Send a notification to all users there will be a maintenance outage.
2. Disconnect the following applications and show a maintenance screen
   1. Identify any applications connecting to the database. If they can be disconnected, especially if you can display a maintenance screen, do that here. The idea is to provide users a graceful way to be notified and not be kicked out in the middle of something, but you also need to make sure that they don’t make changes which will not be preserved after maintenance is completed.
3. Put the database in single-user mode.

**Ending a Maintenance Outage**

1. Put the database into multi-user mode.
2. Reconnect all disconnected applications and remove the maintenance screen.
3. Send a notification to all users that the maintenance is completed.

# Approvals

This manual has been tested and approved by the following people.

The purpose of this section is to show agreement of key personnel on the plans outlined in this manual. It provides management a chance to review these policies and know they have been properly tested. It provides administrators assurance that these policies have approval. It is recommended that the person testing these procedures is not the author of this manual. It also may be helpful to add signatures of management at multiple levels. Whenever a revision is made to this manual, and revisions should happen often to ensure completeness and relevance, signatures should be updated. It is a judgement call whether there is value in getting hand-written signatures every time. It may be sufficient to use the honor system and have approvers type their name and date, so that they only need to update the date on their signatures when needed.

**Testing Approval**

Testing has been performed and passed by the following people.

Person: [John Doe]

Title: [Sr Database Administrator]

Scope: [All Scenarios]

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Management Approval**

Approval by management of the organization has been authorized by the following people.

Person: [John Doe]

Title: [Manager of Operations]

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Acknowledgement

This manual has been created using a template written by Steve Henderson, The SQL Guy. This template has been provided free of charge for general use by anyone. Please do not remove my branding or this acknowledgement when editing this document.

I have worked with Microsoft SQL Server for 20+ years, have multiple certifications from Microsoft, have worked for multiple governments and Fortune500 companies, have published articles, have taught courses, and have built two corporations specializing in data. This template is based on some of the most common scenarios I have seen during my career. Its advice is based on generic needs and resources. Your exact situation is likely to vary in some or many ways, so this template should be modified as is appropriate for you.

If you need any assistance with this template, repair of a disaster, an external opinion of your systems, training of your staff, or any other SQL needs, please consider me, The SQL Guy. You may find many free articles on my blog, <https://thesqlguy.com/blog>, or you may contact me directly for personalized consulting, training, or speaking.

Sincerely,

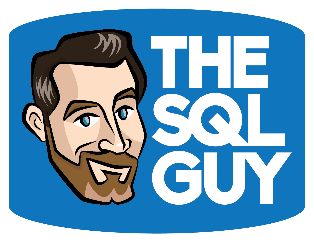
Steve Henderson

The SQL Guy

404-410-6449

[steve@thesqlguy.com](mailto:steve@thesqlguy.com)

<https://thesqlguy.com>

[](https://thesqlguy.com)

[](https://www.facebook.com/SQLGuy)[](https://www.linkedin.com/company/the-sql-guy)[](https://twitter.com/The_SQLGuy)