# **Microsoft SQL Server Database Security, Disaster Recovery & High Availability**

[Microsoft SQL Server Database Security, Disaster Recovery & High Availability - Simple Talk (red-gate.com)](https://www.red-gate.com/simple-talk/blogs/microsoft-sql-server-database-security-disaster-recovery-high-availability/)

****Microsoft SQL Server**** is a relational database management system (RDBMS) and it is one of the most popular and powerful Database software used worldwide. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

## **Database Security**

Securing your Microsoft SQL server is incredibly important if you are a business that handles sensitive data. If improperly configured, the server may become the entry point for an attacker to eventually compromise your network and dump all private information on the internet. While Microsoft secures its SQL servers to an extent, there are a few things that you should keep in mind when custom configuring one for your business.

## **Ways to secure your Microsoft SQL Server**

### **1. Isolating the Database Server**

The Microsoft SQL server should be as isolated as possible from the rest of your company network. Additionally, the installed OS should be lean and only running the processes needed to keep the server running smoothly. If possible, the SQL server should be connected to a restricted network segment and only allow authorized access. This prevents illicit connections and lets the admin enforce restrictive policies for additional security.

### **2. Keeping the Server Updated**

MSSQL receives regular security patches from Microsoft, which fixes bugs and vulnerabilities as they crop up. That’s why it is vital to have a predetermined update schedule, as delays can cause attackers to exploit known security holes in backdated installations. A regular schedule is also necessary to have all code tested on the latest environment possible before anything is pushed to production.

### **3. Common practices when assigning permissions**

It would be best if you took a few measures when assigning permissions to users so that one bad apple does not compromise your entire system. Whenever possible, read access should be granted to views instead of the original tables.

* Users accessing the SQL server should be given only as many privileges as they require.
* Refrain from assigning “sysadmin” membership in MSSQL unless absolutely necessary.
* Procedures and automated tasks should also be run like users and assigned appropriate permissions.

### **4. Securing against SQL Injections**

An SQL injection is a database maneuver where an attacker can slip malicious code into the server disguised as verified user input. The hacker can then perform any number of unauthorized tasks, which include getting full shell access. The standard practice to protect your database against an SQLi attack is by not letting web apps use direct SQL queries for interaction, utilizing stored procedures instead. Stored procedures allow only certain specific operations to be performed, thereby disallowing any data injection into a raw SQL query.

### **5. Secure Database Backups**

Securing database backups and backup storage is also vital. Backup can protect your SQL Server data, storing it in a safe and protected location in the cloud. Backup can tighten security while also simplifying the backup process.

### **6. Surface area configuration in SQL Server**

Surface area configuration in SQL Server allows enabling or disabling components and features that are not used. Major of these options should be disabled, unless there is solid reason to enable them are,

* xp\_cmdshell
* Ad Hoc Distributed Queries
* Ole Automation Procedures
* cross db ownership chaining
* clr enabled
* scan for startup procs

If there is a need to enable any of these features, then you must make sure that you follow the recommended security precautions prior to enabling them.

### **7. Use AD groups or roles for SQL Database Permission**

Stop creating individual logins and granting permission for them. Best practice is to create AD groups or roles, and assign those containers permissions to access resources. This will simply permission assignment for any new hires, modify permissions as a user’s role changes and remove a user’s permissions when they leave the company.  Grant least Privilege when Assigning SQL Server Roles.

### **8. Enforce Password Policies**

Make sure that Password Policy and Password Expiration options are set for all logins. ****Note:**** Logins which are used as service accounts, it is not recommended to set their passwords to expire as such thing could possibly break the connection of the applications to the databases if the password expired.

### **9. Implementing an Authentication Procedure**

There are fundamentally two ways by which you can implement an authentication mechanism for your SQL server.

1. On-server Authentication:- This authentication method requires users to connect to the SQL server before entering their login/password combo to get in. Such a procedure mandates that everyone’s login credentials be stored on the server itself. On-server authentication is typically less secure and should only be used when an Active Directory is not available.
2. Windows Authentication: -Windows Authentication verifies users by using an Active Directory. This procedure takes place without needing to connect to the server.

A baked-in feature in most Windows Server operating systems, Active Directory allows for effortless management of password policies and user access to company applications. It also eliminates the need to store access information on the server.

There is a Whitepaper published by author **[“Pinal Dave”](https://www.idera.com/resourcecentral/whitepapers/sql-server-security-practices)** on ****“**[SQL Server Security](https://www.idera.com/~/media/Corporate/Files/WhitePapers/IDERA-WP-sql-server-security)**“**** and it talks about how security is a core area and non-negotiable when it comes to mission critical applications.

## 

## **Database Disaster Recovery & High Availability**

According to a report by FEMA, 40 – 60 percent of small businesses are unable to reopen in the aftermath of a disaster. These disasters could mean everything from a natural calamity to IT incidents that cause massive data loss. Among the businesses that manage to recuperate somehow, a startling 90% of them eventually fail if their recovery isn’t fast enough.

Safeguarding data and thereby the database is a practice that is often overlooked by most SMEs that get busy expanding their services without covering their bases. An Avast survey says that 43% of companies that have no disaster recovery plan go out of business in the event of a major data loss. On the flip side, a separate study by PhoenixNap shows 96% of companies that do invest in recovery facilities successfully do so.

Database high availability refers to the uptime that a database system can maintain throughout its tenure. If a database is highly available, external users can run critical workloads using data stored on the system without worrying about the server going down or losing any data. Even in the event of a failure, these systems must be able to be back online and working in typically less than a minute.

## **History**

Highly Available databases are a must in today’s data-driven society. However, if a notable server suffers more than a couple of seconds of downtime, it can have disastrous consequences for businesses and even the military and government.

In this article, we shall explore the methods to implement a robust recovery policy for your company.  Microsoft offers variety of options for Database Disaster Recovery & High Availability as per your organization need to meet their Service Level Agreements (SLA) for the Recovery Time Objective (RTO) and Recovery Point Objective (RPO).

A ****Recovery Time Objective**** is the maximum amount of time that a system can be down before it is recovered to an operational state.

A ****Recovery Point Objective**** is the amount of data loss, measured in time, that is tolerable in a disaster.

* Windows Fail-over Clustering
* Always-On Availability Groups
* Log Shipping
* Database Mirroring
* Replication

****Windows Fail-over Clustering**** method is used for downtime management and disaster recovery. Here, multiple clusters are created out of the group of servers. In case a node crashes or seizes to function, another cluster takes over the operation and automatically restarts the application. Here, it does not require any manual intervention. This method uses SAN and NAS for shared data storage and various network connections and thus removes the single failure point. Through quorum-based approach, node-level fault tolerance is increased.

****Always-On Availability Groups**** First introduced with Server 2012 Enterprise Edition, Always On Availability Groups (AGs) provide database-level high availability (HA) with automatic failover for multiple SQL Server user databases. Always On AGs are SQL Server’s premier HA and disaster recovery (DR) technology and essentially replace the older Database Mirroring technology, which had several significant limitations. Always On AGs overcome these limitations and extend SQL Server’s HA and DR capabilities. An AG protects a set of user databases, supporting up to eight secondary replicas of the primary database. In the event of a failure on the primary database, all of the databases within an AG will fail over together and become active on the designated failover partner. Additionally, you can configure secondary replicas to allow read-only activity and most backup operations, potentially taking some of the load away from the primary.

****Log Shipping**** introduced in the Developer and Enterprise editions of SQL Server 2000. In the latest SQL 2017, this is supported by both the Standard and Enterprise editions. The term log shipping means exporting log file of the primary database to another database. In this process, minimum two or more SQL Server instances are involved. Apart from the primary server, other servers are the standby ones. This can be used at the database level only and not at the instance level. Also, automated failover is not available here.

****Database Mirroring****  solution is commonly used for SQL Server Disaster Recovery as well as High Availability . This technique can be implemented of databases with full recovery model. Users can choose from synchronous or asynchronous copy for a single database only. Since no future SQL Server versions (SQL Server 2017 is the last one) will have this feature, Microsoft asked users to go for AlwaysOn Availability Groups or Basic Availability Groups.

****Database Transactional Replication****where data from the primary Server (the publisher) is copied to the secondary Server (the subscriber). This approach can be applied to table level and other object levels. It works best when the database is small and the recovery process requires to be fast. In this way, users can use the subscriber Server if the publisher Server fails after disaster. It is mostly used in server to server environment.

## **Why do we need High Availability?**

Contrary to what some believe, high availability is not just for reducing downtime. While it is a crucial concern, high availability also reduces the strain of multiple transactions on a single principal server. By implementing a solution such as Microsoft’s Always-On Availability, database clusters can share the load of reads and writes among the replica machines. As all these server instances are synchronized, any changes made to one copy of the data are immediately reflected everywhere.

## **The relation between High Availability and Disaster Recovery**

Disaster Recovery and High Availability go hand in hand when deciding what redundancy mechanism you have to choose to implement among your servers. In other words, techniques such as Replication, Log Shipping, Clustering, and Always-On Availability can serve both the purposes mentioned above. However, Always-On Availability is considered the industry standard.

There is an excellent article written by author ****“**[Hugo Shebbeare](https://www.red-gate.com/simple-talk/sql/database-administration/disaster-recovery-for-sql-server-databases/)**“**** on “****Disaster Recovery for SQL Server Databases****” which provides lot more details for SQL Server Database Disaster and High Availability options and methodology and how you can implement one and which method is right for your organization.

## **Conclusion**

With our growing reliance on databases, it is vital to have databases hosted on fault-tolerant systems that can be online 24×7. High Availability is the term used to describe such systems. Implementing safeguards to ensure high availability will ensure that your company is ready for any disaster or outage – planned or unplanned.

## **References**

1. “SQL Server High Availability and Disaster Recovery” Article from ****IDERA**** – an industry leading enterprise database software product. – [View Article](https://www.idera.com/resourcecentral/infographics/sql-server-high-availability-and-disaster-recovery)
2. “****Pinal Dave****” a Developer Evangelist. An author of eleven SQL Server database books, 14 Pluralsight courses, and over 2,900 articles on the database technology on his blog at http://blog.sqlauthority.com. – [View Article](https://www.idera.com/resourcecentral/whitepapers/sql-server-security-practices)
3. “****Hugo Shebbeare****” – Microsoft Database Leader , Speaker & Writer of ****Redgate**** – industry leader in database solutions since 20 years used by 91% of the Fortune 100 companies – [View Article](https://www.red-gate.com/simple-talk/sql/database-administration/disaster-recovery-for-sql-server-databases/)
4. “****Brent Ozar****” –  SQL Server Master, Trainer, Consultant – [View Article](https://www.brentozar.com/?s=disaster+recovery)