

# Database Management

## Module 2: Disaster Recovery

### Learning Outcomes

1. Analyze database recovery options given business requirements.
2. Analyze vulnerabilities of enterprise systems that utilize SQL and apply best practices to ensure server availability.
3. Discuss and evaluate techniques to increase SQL Server Performance.

### 1. SQL Server Always On



In the past, organizations relied upon failover clustering, peer to peer replication and database mirroring to attain their goal of creating a system with high availability. After reading the above, we can certainly understand an organization having this goal. If we take a look at figures 2-1 and 2-2 in the text, we will see an example of how organizations developed strategies for protecting their databases using previous versions of SQL server. Keep in mind that these implementations were highly successful in protecting their data, but there were restrictions. For example, in the case of database mirroring, the number of instances a database could be mirrored was limited to roughly 10. The mirroring features were also left off of many versions of SQL server. Here are some of the more common complaints in relation to the previous attempts made inside SQL Server to address high availability:

- Users wanted an option for more than a single mirror
- Data replication needs to be extended to ensure scalability
- The ability to relate multiple databases within a single mirror
- An urgent need to access data in real time on the mirror
- The option to back up data from a mirror is required.

## **Enter SQL Server Always On.**

The goal Microsoft had for addressing high availability and recovery practices for their products came down to creating a family of tools that help administrators tackle their IT needs with less maintenance and overhead. AlwaysOn was the answer they came up with. AlwaysOn is the name of the built-in collection of comprehensive technologies that provides impressive disaster recovery/high availability results using SQL Server 2012. With AlwaysOn you receive the following:

- Multi database failover with up to four mirrors
- Multiple secondaries in tune with synchronous data
- Active secondaries in tune with asynchronous data
- Integrated High Availability Management
- The addition of secondary subnet support
- An innovative system health check for SQL Server which also reaches out to any shared disks in use.

## **AlwaysOn Availability Groups**

Let us start by examining the diagram on page 23 of your text (figure 2-3) which shows an AlwaysOn availability group. As we saw earlier, AlwaysOn is an alternative to database mirroring. You can clearly see the single primary and three secondary objects in place. Many early reviews of SQL Server 2012 tout this as the best disaster recovery and high availability feature they've ever used. It appears Microsoft may have "hit one out of the park" on this development. The cost is somewhat pricey when you consider the multiple licensing that is required; however, the product does work and enables database system administrators to bring a solution to their organization that raises the bar of scalability, disaster recovery and data uptime. Here are some of the benefits of AlwaysOn Availability Groups:

- SQL Server 2012 provides the built-in tools necessary to deploy Availability Groups
- Built in dashboard that continuously observes AlwaysOn Groups, databases and replicas for errors
- PowerShell cmdlets
- Support for customized failover polices to support each group
- Trim IT spending and improve efficiency in utilizing existing hardware for secondary groups.

## **2. Windows Server Core**



When you install the Standard, Enterprise or even Datacenter editions of Microsoft Windows Server 2008 R2, there is the option to choose a Server Core mode of installation. By choosing this method you are eliminating much of the overhead that is involved in administering a Windows server and actually increasing the security of that server by keeping the number of services installed to a minimum. You can tailor the install of a Core Server to match the application that will run on that server. SQL Server 2012 runs quite well on a Server Core installation. However, you must match the needs of your organization with the advantages and disadvantages of doing so. Let's take a look at some of these on both sides:

#### Advantages

- Fewer patching duties on a core server
- Only specific applications can be installed on a core server, but those that can will be much more secure and have a greater availability in terms of uptime
- No GUI, which streamlines the command interface for many administrators

#### Disadvantages

- It may be hard to convince your Database admins to move your SQL database to a Core Server
- No GUI; while this can be an advantage, it can also become a disadvantage for those DBA's that have not brushed up on their PowerShell commands
- Lack of features such as Windows Explorer or Internet Explorer to quickly download an article.

### **3. Recovery**



As you may know, disaster recovery is a critical function of every IT department throughout the world. Data loss could result in the closing of a company. Managers want to know their data is safe. Many things affect a good recovery process, such as how much time it takes to recover that data and how intact that data will be. As you can see, SQL Server 2012 provides built-in recovery options in case of data loss. It builds upon previous versions and adds many enhancements. Which one you choose determines how much data you can restore. There is no clear-cut choice, and you will need to choose the option that best meets your needs. You can choose the recovery option you like when creating a database and can also change it whenever you wish using one of several methods built into SQL Server 2012.

## 4. SQL Server 2012 Performance



This latest version of SQL Server brings with it a new index type known as **columnstore**. One of the main goals for columnstore was to increase database query performance. It accomplishes this by storing its data in a column, hence the new name. In older designs, data is stored in a type of row. The new design allows SQL Server to reduce disk I/O, which in turn speeds the search process. Despite faster hardware specs coming out each year, the one area not much improved over the years has been disk I/O. The SQL team understood this bottleneck and found a way around it that greatly improves overall performance. Let's refer to the diagrams in the first edition of your textbook.

- Figure 3-1 shows the traditional table for a group of employees.
- Figure 3-2 shows that data in the old format of a “row store.”
- Figure 3-3 shows that same data now in the new “columnstore” format.

After looking at these diagrams, the difference in data storage becomes clear. Now that we have an idea of how that portion works, let's examine the rest of the process the developers created. There are two chief explanations for the storage improvements:

- Only data requested is taken from disk and put into memory, which greatly reduces the I/O workload.
- Compression allows all of your data to be saved on a smaller number of pages, which again reduces the I/O workload.

The new columnstore indexes may not be right for all organizations. Careful examination of your needs is required to determine which method you should choose. Here are a few reasons why you might not want to choose the newer indexing:

- Your data involves regular updating
- Your organization often asks for small inquiries into the database
- Testing proves no benefit over the older method of indexing.

## 5. Hints and Tips for Running SQL Server 2012



Finally, let's take a look at some tips for getting the most from your SQL Server 2012 product. Remember: the performance you receive from your database is directly linked to where your objects are stored. Design is another area that is key when we think of performance. Hardware also plays a role in how much speed is allocated for tasks. To get the most from SQL Server 2012, develop a plan that addresses the locations of your data files, log files, indexes, temp databases and large objects. By streamlining the placement of these objects to work with the indexing you choose, you can ensure your SQL Server gives you the finest rate possible on data acquisition that it can deliver.

### References

Mistry, R., & Misner, S. (2012). *Introducing Microsoft SQL Server 2012*. Redmond, WA: Microsoft Press.