10 Ways To Lose Your DBA Job Cheat Sheet

by [deleted] via cheatography.com/2754/cs/8463/

Introduction

list of things that can cost you a job quickly if you can't manage to avoid them. I think these are fairly critical things for any DBA that has to work on a production system, regardless of whether you do development as well. Development is a different beast, with so many diverse skills needed for different projects, and it's hard to give a list of which skills are critical.

http://www.sqlservercentral.com/articles/Miscellaneous/2744/

1) Backups

Backups are one of those areas that you really need to to know well. While most of these will never be used, it's critical that you can run them on a regular basis and on demand. Having regular backups enable you to recover from issues, which you'll never see coming. Hardware failures, disasters, or the most likely of all, user corruption. In my decade and a half of managing databases, I've only had one or two hardware failures, and no disasters, that forced a recovery from a backup.

In addition to recovering from unexpected issues, you should be able to recover from bad application upgrades or patch problems. It's relatively rare, but I definitely have had cases where we had to roll back from a patch or upgrade and required a restore. And that recovery required a backup to be taken before the upgrade occurred.

One last note on backups: this includes log backups. The most common problem with many databases is the log fills up and the database stops accepting updates. That requires a log backup

2. Restores

Being able to perform a backup is only half of what you need. Restores are even more critical since the backups are useless if you cannot get that data back. This is a very visible part of your job to management since restores are required for production systems only when things are down or in bad shape. Being unable to perform a restore, flipping through Books Online, or making mistakes is not good for your career. It undermines confidence in your abilities and could get you fired on the spot.

For the most esoteric restores, differentials, logs, point in time, etc., you should build some practice scenarios with smaller databases and practice the syntax and options. Keep in mind that many times a restore will be done under pressure, possibly with a President/CEO/Senior executive looking over your shoulder, so the more you've practiced things, the better prepared you'll be. Especially if you use a third party product; those require extra practice.

3. Forgetting about Media

With all the new regulations that are becoming mandatory for data retention, disclosure, and protection. Think Sarbannes-Oxley and similar legislation around the world.

It is rare that a restore takes place from more than a week or two prior to the event, but I've been involved twice with restores that were done from backups more than a year old. In this case you often need to know the service pack/patch level as well. This is where documentation comes into play and even though it's not in this list, it's important.

But this is where a scheme by where no media are used, as advocated by some vendors, could get you into trouble. It's no often it will be required, but if it is and you cannot produce a restore, it could be a problem.

The last thing about media is it gives you a second chance to recover a database. I've always advocated, as have many others, to run your backups to disk, keep a day or two's worth of backups on disk, and also copy these to tape. That gives you a quick restore from disk, but in case your disks fail, you have the chance to get things from tape. As reliable and fault tolerant many of the new SAN and other disk technologies are, they are far from foolproof and suffer from failures. Having your backup disk system fail when you learn that the end of quarter reports aren't complete could put you in a bad situation..

4. Blank Passwords

Losing your passwords to someone because of a brute force cracker is one thing. Giving them easy access to everything is pretty bad and not even using a password, like "sa/blank", should get you fired. I'd fire you if you worked for me.

There's no defense for a blank password at all, but these days there's really no defense for default passwords or simple ones being used on production systems. They don't have to be "kd5#kdj9s2m", but they shouldn't be "password" or "1234". Your boss may not be able to figure things like this out, but at some point your use of easy passwords will come out and your only real excuse is you were lazy.

Do yourself a favor, get a copy of Password Safe or some other secure utility and store complex passwords for your servers and services in there. If you want easier to remember passwords than "-k%s92kD5", then use phrases "BlueCornTorT!llas" and add these to your store.

And change them! Even secure passwords can be cracked. Changing them on a regular basis is the best way for you to be sure that your passwords will secure your data. Nothing is for certain, but applying a few layers (strong passwords changed often) will help increase your security.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

4. Blank Passwords (cont)

One last thing. I mentioned production servers, but really you should perform the same actions on development servers. This is for two reasons. First we often restore production data to development servers, so not securing the development servers is a bad idea. The second thing is this helps build the habit of being secure. Security is hard, so having good habits help ensure you will consider it a regular part of your job rather than an annoyance.

5. Patches

When Slammer hit my company many years ago, we were caught with many servers that weren't patched. Most of them were servers controlled by other IT groups or departments, but we had a few that we hadn't patched. We had no good excuse and while some were un-patched because clients would not give us a maintenance window, others were just not patched. We also had some servers that were patched with code that undid security from earlier patches, but those were not our fault.

All software has bugs and that includes databases. Vendors are regularly releasing patches and you should have a process to test and apply those patches. Microsoft has gone to a monthly patch delivery cycle, though these are for all products, not just SQL Server. It's pretty rare that we see a SQL Server patch, but sometimes the OS patches are just as critical and can affect SQL Server, so be sure that you look at all patches.

You should expect patches on the second Tuesday of every month from Microsoft and be prepared. Test systems should be ready for patches that first week with people available to perform a specific set of tests. You should have a list of basic functionality that you need to work and run this workload against your systems. If you can automate this, that's great, but it's not the always available for many companies. You should also have a schedule to deploy these patches sometime within a week to a month after release. You might stagger the deployment if you have a large number of servers.. By building a regular patch cycle into your process, everyone in your organization, from the CEO on down, will get used to the idea of security being a regular and important part of IT. And you won't be left explaining to anyone why critical servers were compromised in some way because of missing patches.

6. Securing Data

I once started a new job and went to look at their SQL Server. As I looked over the system as "sa", I could see and work with all objects and didn't think about it. A few weeks later I was deploying a change to the schema and realized that there were no roles to assign security against. So I then went to check the server roles and realized that every user, from internal employees to the web applications, were only a member of the public group. Digging further in, I realized that every object in the database had every permission assigned to the public role, including the auditing tables.. I brought this up as a concern to one of the executives and he didn't understand why anyone cared about the objects in the database. After all, the data was important, not any objects. It took some explanations for him to understand the data lives in the objects and if the objects aren't secured, neither is the data.

Often security inside a database is lax since many applications require access to most objects. If the application cannot access the object, why does it exist? There are a few cases, however, where security is more important. Auditing tables often should allow only inserts and selects, not updates or deletes. You may have other types of data that is read only. Or you may have other specific reasons for limiting access to objects, like you use stored procedures and allow no access directly to tables or views..

Whatever your needs are, be sure that you apply security properly to meet those needs. You want to ensure that people get access the data they need, but not purposely or inadvertently, change data that should not be changed. The last thing you want to explain to an executive is why last year's sales figures were "updated" accidentally by an analyst.

7. Maintain Your Server

Different than just patching your server since SQL Server doesn't have too many patches. It does require some level of maintenance on the databases in addition to backups and restores and this is one place a DBA proves their value.

Basic maintenance on a SQL Server involves running regular DBCC commands and checking to ensure that there are no integrity problems inside your database. I don't necessarily expect you to be able to fix them without some help from Microsoft, but you should know they are there. If a database gets corrupt and it's found out DBCCs would have been failing for the last month, you might be out of a job.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

8. NULL Math

It's not often that you'll get into situations where NULLs affect your queries, but when they do, the last thing you want to explain to someone is that the database's math is wrong. As an example, look at this table:

Region Sales
--------- -----
East CO 3.0
North CO NULL
West CO 9.0
North NM 8.0

What's the average for sales per region? It could be construed as 5 or 6.67, depending on whether you count the NULL as 0 or ignore it as a missing value. By design, SQL Server aggregates ignore NULL values and so the result is 6.67, but if users or applications use NULLs when they mean to use a zero then you might produce results that are incorrect.

8. Loading Data

If there's one thing that I've done the most often as a SQL Server DBA, it's writing a report for someone. The thing I probably do second most is load or export data for one thing or another. Often the data movements are reports to or from Excel.

Any DBA should be able to load and export data on demand to or from a table. Heck, the wizards can do most of the work, but you should know how to get data loaded and then get it into the tables where an application can use it.

9. Tuning

I've been truly amazed at how often people that interview for a DBA position don't know how to read a query plan, can't define a bookmark lookup or clustered index, and perhaps are unable to even describe how to go about tuning a query. This is a core skill for anyone that is a DBA. If you're a system administrator stuck with SQL Server or a developer building .NET applications you might not be familiar with performance tuning inside SQL Server. A DBA, however, needs to be able to do some tuning of queries.. Every DBA should know the basics of indexing, reading query plans and determining if indexes are being used, and how to run Profiler

10. Monitoring

You need to know how to look at a server, through Profiler or Performance Monitor, and determine what's wrong with the performance. Or at least how to tell if it's hardware or the application. You should be able to monitor and explain what the server is doing.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com