

The Battle For Server Dominance

(Tips on how they work, their similarities and differences)

Ejup Rustemi
Master at State University of Tetova

Mefail Tahiri
Docent at State University of Tetova

Abstract

A database is first and foremost a service provider. It waits patiently for requests that come in from clients (applications and tools), performs what it is asked to do, and responds with answers. In most cases it is deployed as a networked server and registers itself as a network service. There are many database management tools in existence, but certainly the most popular of the bunch are Oracle and Microsoft SQL Server.

Key words: *Microsoft SQL Server, Oracle, Database, Database Administrator..*

A database is not an island. Most often it is a server deployed as a network node that provides persistence and transactional services to applications. It is a networked service that waits for remote connections, authenticates connection requests, receives requests for data or operations on data, and services them. From this perspective it is similar to many other servers that exist on the corporate network (e.g., Web servers, e-mail servers, naming servers). While many other aspects make the database very different and very special servers (hence the

need for a book that is focused on database security and auditing), this commonality does mean that many things can be learned from the security realm in general—things that can assist you when implementing database security.

A database is first and foremost a service provider. It waits patiently for requests that come in from clients (applications and tools), performs what it is asked to do, and responds with answers. In most cases it is deployed as a networked server and registers

itself as a network service. Clients search for database services on the network using various network handles, and when they find what they are looking for, they initiate a connection to the database. One view of your database—and the one taken in this chapter—is therefore as a node on the network, registered to provide services.

As platforms continue to evolve in the technology industry, a central concern for IT executives is implementing the right systems to maximize the return on their investments. Since labor costs dominate most IT budgets, selecting platforms with lower implementation and on-going management costs can significantly improve overall IT efficiency, and allow for the reallocation of resources from basic maintenance and operations to more innovative functions.

There are many database management tools in existence, but certainly the most popular of the bunch are Oracle and Microsoft SQL Server.

Importance of DBA's

Database administrators (DBAs) have significant responsibilities. They must not only provide reliable access to company data, but also protect that data, monitor the

database environment, troubleshoot problems, and more. If the lives of DBAs were simple, what fun would that be? That is probably why they enjoy the job so much—because of the different opportunities and challenges that they get to face day to day.

To provide a stable and highly available database environment, along with planning backups and performing recoveries and all the other maintenance tasks, DBAs are usually exploring how to use new features, and even learning multiple database platforms. Yes, as a DBA, you probably need to know more than one database platform. The days of being just an Oracle or SQL Server DBA are probably gone. Companies are adopting more than one database platform, and applications may import or export data from one database system to another. But why we say this to ourselves? We may already have decided to add Oracle to our arsenal. The big question is, How hard will this be?

In cases when you are a SQL Server DBA, you don't need to start back at square one to support Oracle databases. Certainly, there are differences between the platforms, but many of the DBA tasks are basically the same. You can apply the skills you already

have to learning Oracle. Here we will try to translate the terms from SQL Server to Oracle and viceversa, because sometimes being able to speak the language is half of the battle.

Servers, Databases, Instances, and Schemas

As a DBA, you are certainly familiar with database terminology. The problem is that on different database platforms, the terms don't always mean the same thing. Consider that even general terms can take on various meanings. A generic definition of database is "data and information that is collected together for the ability to access and manage." The term server could refer to the actual server hardware or to the database server. A SQL Server database is not the same as an Oracle database. The SQL Server database has users allocated to it, its own system objects, and its own datafiles. In Oracle, the term schema is more closely related to the SQL Server database. Schemas can own the objects in both environments. In SQL Server, there could be several databases for the instance, so there can be several different schemas in a database. In Oracle, there are multiple schemas in a database, including the system schemas. The Oracle schema is a collection of objects, and

it could have its own tablespace allocated to it on the datafiles for the database. The user schema usually does not contain system objects, because the system objects are in their own schema. There is only one set of system objects for each database server, unlike with SQL Server, which has different layers of system objects at the server level and the database level. Also, there are no users inside the Oracle schemas, because they are only at the server level.

The Oracle database is almost like the instance level for SQL Server. The Oracle database is the overall group of datafiles and system information. The Oracle software, memory structures, and processes make up an instance. There is one Oracle database for the instance. In a clustered environment, there can be multiple instances that all point to one database on a shared disk.

Database owner is another term that doesn't really exist in Oracle. Typically, the term schema owner is used.

Hardening the database

Hardening is a process by which you make your database more secure and is sometimes referred to as locking down the database. When you harden your database

environment, you remove vulnerabilities that result from lax configuration options and can even compensate for vulnerabilities that are caused by vendor bugs. Although you cannot remediate these bugs, you can form an environment in which those bugs cannot be exploited.

Hardening is also called hack-proofing. The essence of the process involves three main principles. The first involves locking down access to important resources that can be misused—maliciously or by mistake. The second involves disabling functions that are not required for your implementation, which can be misused by their very existence. The third principle is that of least privileges (i.e., giving every user, task, and process the minimal set of privileges required to fulfill their role).

Oracle is one of the most well-documented database environments, and there are many hardening scripts on the Web (e.g., Pete Finnigan's [checklist](http://www.petefinnigan.com/orasec.htm) at www.petefinnigan.com/orasec.htm).

Hardening an Oracle environment should include at least the following tasks:

- Physically secure the server on which Oracle lives.
- In a UNIX environment:
- Don't install Oracle as root.

- Before installing, set the umask to 022.
- Don't use /tmp as the temporary install directory; use a directory with 700 permissions.
- In a Windows environment, do not install Oracle on a domain controller.
- Create an account for each DBA that will access the server; don't have all DBAs logging into the server using the the same user.
- Lock the software owner account; don't use it to administer the database.
- Verify that the Oracle user (at the operating system level) owns all of the files in \$ORACLE_HOME/bin. Check permissions in this directory and (on UNIX) check the umask value. File permissions should be 0750 or less.
- Understand what features and packages are installed on your system. Oracle is very functional and has many options. If you're installing from scratch, install only those features that you really need. If you already have an installation, review the options that are enabled and remove those that you don't need; etc.

SQL Server has suffered from a lot of bad press and from several very visible attacks. It is also one of the most functionally rich databases, which translates to “inherently insecure” in security lingo. Luckily, SQL Server is also one of the most well-documented environments. There are numerous resources available that can help you secure your SQL Server environments, many products that can be of assistance, and a very large community supporting security in this environment. Furthermore, contrary to public perception, Microsoft is actually investing a lot in making the SQL Server platform more secure.

Hardening a SQL Server environment should include at least the following tasks:

- Physically secure the server on which SQL Server lives.
- Apply all service packs and hot fixes to both the Windows operating system and SQL Server. You can execute `select @@version` to see precisely which version you are running. You can see what this version maps to in terms of patch levels at www.sqlsecurity.com/DesktopDefault.aspx?tabid=37.
- Make sure all SQL Server data files and system files are installed on an NTFS partition and that the appropriate permissions are defined for the files.
- Use a low-privilege user account for the SQL Server service. Don't use LocalSystem or Administrator.
- Delete setup files. Setup files may contain plain text and weakly encrypted credentials. They contain sensitive configuration information that has been logged during installation. These files include `sqlstp.log`, `sqlsp.log`, and `setup.iss` in the `MSSQL\Install` (or `MSSQL$<instance name>\Install`). Microsoft provides a free utility called `killpwd` that locates and removes these passwords from your system.
- Secure the sa account with a strong password.
- Remove all sample users and sample databases.
- Review all passwords. At the very least, check for null passwords using the following SQL: `select name, password from syslogins where password is null.` (strength.)

- Remove the guest user from all databases except from master and tempdb.
- Review how roles are assigned to users at a database and server level and limit assignment to the minimal set necessary; etc. (*Implementing database security and auditing; Ron Ben Natan; 2005*)

Backing up your data

As a DBA, you already know the importance of database backups. You have probably developed your own backup strategies and disaster recovery plans for the SQL Server databases you support. You also know that being able to back up a database is just part of the job. You need to manage the backups and know which backup pieces to use to be able to recover quickly and easily. Testing restore procedures validates that you have solid, usable backups, and also lets you develop restore scripts to save for a rainy day.

As a DBA, no matter which database platform you are administering, you need to plan a backup strategy that makes recovery a simple process and fits the needs of the business. There are several strategies for backing up databases, depending on

resources, the importance of the data, and if it is needed for recovery and running the business.

For SQL Server databases, creating maintenance plans sets up some backup strategies. The SQL Server database recovery model setting of either SIMPLE or FULL also determines if log backups are needed. As the database changes or grows, plans to compress, write to tape, and allocate threads to use for the backup are all taken into consideration.

The Oracle DBCA also has a step to set up maintenance plans, which include backup settings. You also have the option to create the database in NOARCHIVELOG mode, which is similar to the SQL Server SIMPLE option, or ARCHIVELOG mode, which is like the FULL option.

It is important to be able to communicate the options for backup and restore strategies to the business owners. Understanding the options, the resources needed (disks, tapes, and so on), and what data loss would mean to the business are key to being able to convey how the restore options would benefit the business. This will help you and the business owners to develop the best plan to balance the resources and protect the data assets.

What are you protecting against? You probably have been setting up backup strategies to protect against hardware failures, data loss, changes that have gone wrong, and disasters. Running backup database and transaction logs, and saving copies to disk and tape with certain retention policies are all part of this plan.

Maintaining the database

As a SQL Server DBA, you've planned database monitoring and set up maintenance jobs. With various versions of SQL Server, some tasks may be more important than others; something that was a must for SQL Server 2000 might still need to be run in SQL Server 2008, but not as frequently because it's not as crucial. Oracle versions make a difference as well, especially if your database has older features, such as dictionary-managed tablespaces.

In large database environments, it is not possible to spend all of your time logging in to every database and validating logs and jobs. Automated tasks need to be developed to perform these tasks, and you will want to generate a report or summary to let you know that all systems are looking good. (DBA'S do tend to do a manual check occasionally—not that they don't trust the

automated jobs, but a verification every now and then is reassuring.)

Generally, it's easier to develop maintenance jobs for a new database that you create, because you understand that database's setup. It may be more difficult to make sure that the maintenance jobs are running against existing systems, because jobs might be named differently or scheduled another way. However, you can use the database tools to verify that these tasks are running and if new ones need to be included.

In SQL Server, the Maintenance Plan Wizard helps you set up general maintenance tasks. These include checking for database integrity, cleaning up history, rebuilding and reorganizing indexes, shrinking the database, and updating statistics. In Oracle, you can schedule maintenance tasks in the Oracle Scheduler, and some system jobs are set up when the database is created. (*Oracle database administration for Microsoft SQL Server DBA's*; Michelle Malcer; 2001)

So, which one is better?

Again, you don't need to know everything about everything, but you do need to know

what issues are important to the databases. Also, you need to talk to other teams to be able to maintain a well-performing environment, because it is definitely a team effort. Learning from other areas is good, but learning about the databases from other DBAs is also helpful. User group members and coworkers are great sources of information. Being prepared to support a different database and even a different operating system requires learning from others.

DBAs have the opportunity to be in the middle of things, and to work with different people and teams. Being prepared to offer ideas and knowledge about database solutions makes the job interesting and makes you more valuable. The teams will start to depend on your ability to evaluate why a database solution here makes sense, while using another solution for something else is a good idea. Being able to see the big picture, including how the application is getting the information, is a skill we are constantly improving. Being able to speak the database's language is valuable when drilling down into the environments.

In becoming a DBA, you might not have initially realized all of the learning that comes with the job. Those DBAs who

challenge themselves and reach out to learn more are the ones who succeed in their careers. Learning from each other, being willing to explore new areas, and then being able to pull all of the information back to apply to the database environment are the tricks of the trade.

Learning Oracle will broaden your perspective on databases. Keeping the skills you have already developed as a DBA and learning how to apply them in other environments will continue to challenge you as you develop more experience. The fun of the job is being able to be involved in several aspects of the systems we support as we develop reliable, secure, and robust database systems. (*Microsoft SQL Server for Oracle DBA's; M. Anderson, J. Fox & C. Bolton; 2001*).

Today, the Microsoft SQL Server brand encompasses more than just a relational database engine that you install on your own servers; it now includes business intelligence features, a complex event-processing engine, highly scalable data warehousing solutions, and a version of SQL Server running in the cloud. In other words SQL server represents an outstanding alternative to Oracle, and considering that it is a Microsoft product, it means that the

working environment can be more adaptable considering the vast majority of computers run on Windows OS.

Survey

The following table shows the results of a survey that Alinean Research Group conducted by researching the actual costs and efforts required for one hundred organizations to manage these two database management systems. Alinean conducted in-depth interviews with 100 Directors of Database Administration and Senior level Database Administrators regarding their database environments, user populations, and database administration activities.

The survey results reveal that overall, Microsoft SQL Server required significantly

less effort to install and maintain, than Oracle Database. Study participants reported that on average a Database Administrator (DBA) could manage over 30 Microsoft SQL Server databases, while Oracle Database implementations required one DBA per 10 databases. Factoring in slightly higher average salaries for Oracle DBAs, the corresponding annual cost for administration for these two databases comes out to \$2,847 per year per database for Microsoft SQL Server and \$10,206 per year per database for Oracle Database; over a 350% difference in annual costs per database.

Measure	Microsoft	Oracle
Average number of databases per company	107	87
Average number of users per database	328	716
Mission critical databases	66.1%	63.8%
Transaction-based databases	55.7%	60.3%
Decision-support databases	44.3%	39.7%
Databases supported per DBA	31.2	9.9
Users supported per DBA	6784	5567
Annual TCA per database	\$2847	\$10206
Annual TCA per database user	\$13.09	\$18.15

Conclusion

Since databases touch so many environments, the learning does not stop with the new features and other database platforms but continues with operating systems, networking, and applications. The opportunities are just out there, waiting to be added to your experiences.

No matter which way we choose to follow, be the one by Oracle or that of Microsoft, we will end up learning an excellent tool that will help us overcome the many problems that exist in the world of business and productivity.

In the end as the saying goes, it's not all in the tool, but in the way we use it.

References:

- *Microsoft SQL Server for Oracle DBA's; M. Anderson, J. Fox & C. Bolton; 2011*
- *Oracle database administration for Microsoft SQL Server DBA's.; Michelle Malcer; 2011*
- *Implementing database security and auditing; Ron Ben Natan; 2005*
- *Microsoft SQL Server and Oracle Database - A Comparative Study on Total Cost of Administration; Alinean Inc.; 2006*
- *Relational Database Index Design and the Optimizers; T. Lahdenmäki & M. Leach; 2005*
- <http://www.microsoft.com/en-us/server-cloud/products/sql-server/>
- <https://www.oracle.com/database/index.html>