

# THE ORACLE DBA – DETERMINING THEIR ROLE AND YOUR NEEDS

*Mabesh Kalluri, Lead Consultant- Oracle DBA.*

## INTRODUCTION

With the advent of the iDatabase and the reliance on Oracle for E-Commerce, most organizations find that successfully deploying and operating their platform requires expert assistance. Oracle is a robust and complex product, and organizations that invest in this technology need to understand its support requirements. The Database Administrator (DBA) has classically provided support for just the Oracle database, but in most environments the role requires more. The DBA must sometimes be an expert in design, development, implementation, support, even internet technology, but not every organization needs all of these skills concurrently, nor even at the same location.

This paper will distinguish between the various roles, and help the IT manager determine what skills are needed during which part of the project lifecycle. We will address how to determine requirements, what a DBA does, when a DBA is needed, what type of DBA is appropriate, how to recognize the "right" DBA, and what other options are available. This paper will attempt to address an organization's specific needs and how alternate technologies and resources can be utilized to meet those needs.

## GLOSSARY OF TERMS

It is useful to first layout a glossary of terms as this paper makes use of some specific semantics:

<i>Database</i>	The collection of files and their internal structures that contain data. Does not necessarily refer to a single application's data store because applications can share a database.
<i>Database Administration</i>	Short and long-term management of a database to maximize performance and minimize downtime.
<i>DBA</i>	Database Administrator; the person that performs Database Administration (and usually other) functions. See the section on <i>Types of DBA's</i> for a description of various distinctions between DBA's.
<i>Instance</i>	The memory (cache) and processes that access a database. Can be >1 per database in an Oracle Parallel Server (OPS) configuration, a high-availability form of Oracle
<i>Oracle</i>	Besides referring to the company that makes the database software, the term Oracle is generally used to refer to the database, it's instances, and the software that is used to run and manage it.
<i>SQL</i>	Structured Query Language - the command language of most Relational Database Management Systems (RDBMS), such as Oracle
<i>PL/SQL</i>	A procedural (programming language) version of SQL specific to Oracle databases

## DO I NEED DATABASE ADMINISTRATION?

**YES!!** But do you need a DBA? Not necessarily. When asked, most IT managers respond to the question, "Why do we need a DBA?" with the answer, "We just do." When pressed, the question, "What does a DBA do?" usually comes up, and many managers will frankly admit, "I don't really know -- but I know we need one!" While this is humorous as an anecdote, it is at the same time alarming because it follows that it will be almost impossible to find the "right" DBA for your environment. There are many reasons for this lack of understanding, but many times it is simply because the

DBA is called upon to provide assistance in nearly every aspect of a database. Few beside the DBA understand the roles that need to be filled in a database's lifecycle. This paper will hopefully clarify what a DBA is.

It is important to first express the need for Database Administration before determining the need for a DBA. When Oracle installs easily, when development is going well, or when a database is running smoothly, it is easy to get lulled into a false sense of comfort. However, this is generally the point at which the most important part of database administration should take place: prevention. It may be stating the obvious, but outages, corruption, data loss, and poor performance can result in massive tangible and intangible costs to the organization. Yet each of these can be avoided or rectified with proper database administration.

Every Oracle installation requires some form of administration at some point. Many administrative tasks can be completed without outages, if properly planned, but lack of administration will definitely result in an outage eventually. The question becomes, what administration is needed for your environment? Many factors determine this, such as size, transaction type, throughput, whether it is a production system, etc. Generally the larger it is or the more it does, the more resources need to be dedicated to it. While a team of DBA's is probably not needed for a small test machine, a large-scale, millions of transactions per minute system might require that team.

Now the good news: simply because database administration is needed does not mean that you have to hire a DBA (or a team of them) right away. Even when you do need a DBA, you may not need one full-time or onsite. Every database has a lifecycle, and skills required during that lifecycle can fluctuate significantly. These options will be explored in detail later.

### **WHAT IS DATABASE ADMINISTRATION?**

As defined above, Database Administration is the short and long-term management of a database to maximize performance and minimize downtime. This is somewhat vague, but it is important to point out that things such as development are *not* a direct part of database administration. So, what is? What follows is a list of some common database administration tasks:

- Install, upgrade, and configure database software (*not* application or system software)
- Define and implement physical database layout (both on disk and within the database)
- Create and manage database objects (tables, views, indexes, etc.) per application specifications
- Monitor, tune, and troubleshoot database and environment
- Manage database security, roles, and profiles (*not* operating system or network security)
- Manage database storage and growth, purge/archive data, reorganize database as needed
- Manage database backups and define recovery plans
- Execute disaster recovery plans
- Prevent, detect, and correct problems
- Communicate with Oracle, users, and management
- Ensure license compliance with Oracle
- Contribute to hardware and software evaluations and recommendations

### **WHAT IS A DBA?**

Although it is fairly easy to define database administration, defining a database administrator is more difficult because a DBA invariably does more than just administer databases. However, in general a DBA is:

- Fireman, Policeman, Surgeon, Shepherd
  - Fireman because s/he responds to problems whenever they occur. Tends to be the first line of defense.

- Policeman because s/he is responsible for setting up and maintaining security, monitoring resource utilization, and scheduling intensive jobs or outages
- Surgeon because s/he needs to provide both preventative “medicine” as well as emergency surgery to keep the platform healthy
- Shepherd because s/he needs to work with the development and user communities, attempting to guide them toward the best “pastures” (practices) of database design and usage
- Frequently first point of contact for database problems
- Expected to provide expert advice and guidance in technology, particularly to developers
- Needs to work with management and engineering to build optimal configurations
- Emergency support during a crisis

### *CHARACTERISTICS OF A GOOD DBA*

Beyond just the definition of what a DBA is, the DBA must also have certain characteristics to be effective in your environment. The following is a list of optimal characteristics of the person that you might want as your DBA:

- Must be knowledgeable of the technology, but must also "know what they don't know" - you do *not* want the "if all I have is a hammer then every problem looks like a nail" syndrome. A good DBA will know his or her own limitations and will not attempt to feign expertise
- Is responsive 24x7 - returns pages and answers the phone
- Really enjoys the work - a disgruntled DBA can be worse than no DBA at all
- Can multitask effectively
- Can work effectively with little sleep - invariably will need to be nocturnal on occasion
- Can shoot from the hip, yet can also create and follow procedures
- Communicates and documents well
- Natural problem solver
- Detail oriented - considers all facets of a problem
- Has "the gift" (unfortunately usually comes with "the ego")

### **WHAT ISN'T A DBA?**

Now that the "core" skills of a DBA have been defined, it is important to understand what a DBA is not. The DBA is regularly called upon to assist in many areas, some of which may be more important than others for your organization. The following tasks are not usually part of a DBA's role (although skills in these areas are beneficial):

- Application development (although a background in development can be very useful)
- Project Management
- Unix/NT administration; System management
- Web development and administration
- Functional/business specialist
- Enterprise security administration
- Hardware integration and troubleshooting
- Network management
- Other Development
- Ad-hoc report-writing
- General application support
- Database design (except for Development DBA's)

- Release control (except for database-level changes)

## **TYPES OF DBA'S, OTHER DATABASE ROLES, AND NON-DATABASE ROLES**

Within the role of the DBA are sub-roles that provide different sets of knowledge or skill. Your environment may not require a production DBA, or even a very experienced DBA, to satisfy your requirements. The following list defines the different types of DBA's:

*Database Operator (DBO)* This is a person, usually of a very junior level, whose sole job is to interact with a production database in a first level of support role. He or she should have sufficient training to know where key files and logs are, have some familiarity with SQL, and the ability to recognize errors and problems with Oracle. May also be responsible for job execution and monitoring. Generally also responsible for validating regular backups and maintaining tape library.

*Junior DBA* This is a person that usually has less than about 2 years experience as a DBA, but has had training and/or hand-on experience with Oracle databases. May work in development or production environments.

*Development DBA* This DBA specializes in development environments, whose needs differ substantially from production environments. This person may prefer a "standard" workday to the 24 requirements of most production systems, may have been a developer before becoming a DBA, and should have experience handling multi-release environments. Generally requires less knowledge of Oracle, but more knowledge of programming and specific applications. Will hopefully give some thought to the production implications of changes. May also be a Database Designer.

*Production DBA* This DBA is usually a more senior DBA, generally having aspects of all other DBA's. However, it is possible for a Production DBA to have no experience with development systems or practices, because the Production DBA's core focus is on insuring uptime and optimal performance (in that order) of the production database. This person may or may not have in-depth experience with the application(s), but is usually very experienced at tuning applications and the database.

*Application DBA* A specialized form of DBA, this person usually has some of the experience of the Production and Development DBA's combined, but specializes in one or a suite of applications. This specialization allows them to make informed decisions about data, as opposed to just the tables that contain the data. For example, they might have a better understanding of what data can be purged at what time, what tables can be taken offline without impact to the platform, how poorly performing queries can be tuned based on data flow, appropriate security measures for new tables and users, etc. They are also intimately involved in releasing new versions of an application into a production environment. This person may be less knowledgeable of Oracle as a whole, or may have experience with a limited number of environments.

*Enterprise DBA* This is basically just a Production DBA that works on more than one system and application. This title is usually only encountered in large environments with teams of DBA's.

Figure 1 shows a visual representation of these types of DBA's, as well as non-DBA roles (described below), and in what areas there might be overlap with the DBA for your environment.

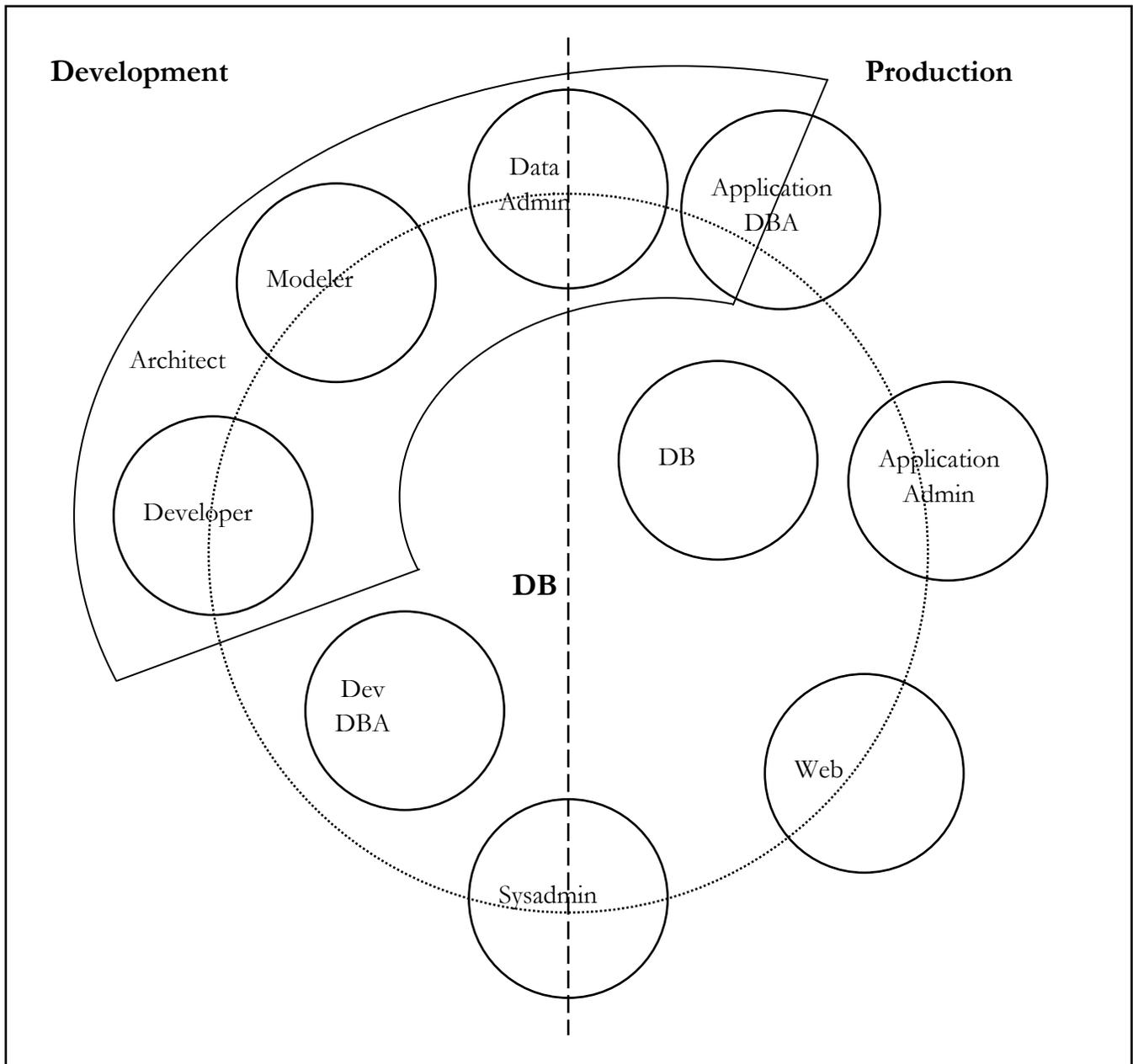


Figure 1, Roles and Relationships to a DBA

In addition to DBA's, there are other closely related roles that are not technically DBA's. However, in your organization it may be appropriate to have a DBA that can satisfy these roles as well, or vice versa:

*Data(base) Architect*

Some may consider there to be a difference between a Data Architect and a Database Architect, but the distinction can be a fine one. Generally the architect is a very senior level person that understands how to translate business needs into a working system. They are usually involved at a very early stage of system, and can even have involvement that carries over into a production system. While they technically do not need to know the inner workings of Oracle (some would argue that they should not care what the physical implementation is), many frequently do. The Architect must have excellent communications and analytical skills.

<i>Data Administrator</i>	This person is responsible for the management of data (as opposed to the database). This role can be either a development or production role, or both. On the development side, this person might be responsible for ensuring uniform data definitions throughout the database and application (for example that social security number is always 12 characters in length, with the 4 <sup>th</sup> and 7 <sup>th</sup> character being a dash and the others being numerals). In production they might be responsible for ensuring data integrity and that data are used in uniform ways (such as that area code is always entered). They are commonly the "owners" of the data model, and review any proposed change. It is an unfortunate fact that this role is often overlooked, because addressing data issues early on can save a lot of rework later. This role is sometimes satisfied through proper change control processes.
<i>Application Administrator</i>	Just as Oracle requires administration, most applications that use Oracle require administration of some sort as well. The scope of this can be large or small, and therefore it is possible that this may just be a portion of someone else's responsibilities. Generally the application administrator does not require intimate knowledge of Oracle, and may not even know SQL. However, application security is usually a key function, and therefore knowledge of Oracle security is generally mandatory.
<i>Developer</i>	This is the person that actually does the programming of an application that uses a database. Their knowledge of Oracle can range from none to very extensive, although they will generally know SQL extensively and probably PL/SQL, Pro*C, JSQL or graphical development tools as well. Some DBA's choose to become developers. Developers can sometimes provide all the necessary DBA skills needed for a development or test platform, but should almost never be the DBA for a production server (at the same time they are developing applications for it) because they tend to have an application-centric view of the database. The temptation may also be strong to "go under the hood" to make changes to production data, which can be disastrous.
<i>Database Manager</i>	This can mean two things - it can either refer to a management level person whose role it is to manage a team of DBA's and related personnel, or it can be another term for a Database Operator or Junior DBA.
<i>Database Designer</i>	Similar to a Database Architect but more limited in scope, the Database Designer's role is to define the objects that will be part of the database (referred to as a "schema"). They generally work from specifications produced by database architects and others, and understand the distinctions between logical and physical design. Furthermore, they are usually the people that actually work on the transformation from logical to physical. They are generally senior, and have a solid understanding of Oracle. More than any other single development role, designers need to understand how a database will function in a production environment, and attempt to make physical design decisions based on that. It is not uncommon for a development DBA to be a database designer as well.
<i>Data Modeler</i>	A subset of the Database Designer, the Database Modeler works specifically with the logical design of a database, from specifications provided by the Database Architect. They are generally only lightly involved with the physical design, and subsequently usually have limited knowledge of Oracle in a production environment (although they may have significant knowledge of applications that use Oracle in a production environment, a subtle but important difference).

There are usually other, non-database skills required for an Oracle shop. Some of these include:

<i>System Administrator</i>	This person is responsible for the Operating System and Hardware on which and Oracle database runs. There is usually a clear distinction in roles, and sometimes a conflicting one if the machine must share resources with non-Oracle applications. It is generally <i>not</i> a good idea to have the same person satisfy both roles, although a crossover of knowledge can be beneficial.
-----------------------------	--

*Network Administrator* This person is responsible for the network. Connections between clients and servers, or between databases, may fall into their domain, so Network Administrators sometimes have some Oracle knowledge. Generally Oracle DBA's need some amount of networking knowledge as well, particularly in complex, multi-database environments.

*Web Master/ Developer* Web skills are becoming more and more critical, and the DBA is more commonly having to have some level of knowledge about the internet and Web deployment. Your environment may not require any web connectivity, or it may do nothing other than service the web. Web skills may therefore be needed in your DBA.

*Project Management* The project manager coordinates the resources and time constraints needed to deliver a project. While a database administrator should have the ability to manage tasks and even personnel, and project managers would do well to have some understanding of Oracle, it is a mistake to have the same person fill both roles on anything but a small project. These positions both require dedicated resources, and the requirements of each can at times be at odds. If a project manager is interested in becoming more technical, they should begin with development, not database administration. Conversely, a database administrator wishing to become involved in management should not manage a project that involves a database s/he is currently administering.

**COMMON DBA CAREER PATHS**

It might be interesting for the IT manager to understand how the roles can form a career path for many individuals. Figure 2 diagrams how an individual might change roles from one job function to another. The left side of the diagram indicates people with zero or more experience, the middle section is usually achieved after 2 or 3 years of experience, and the right side is usually not reached until 5 or more years of experience. Note that many people, such as Developers, frequently stay in a single role. This should not be construed as a lower level of expertise simply because they have not chosen to take on other roles - not everyone wants to be a DBA!

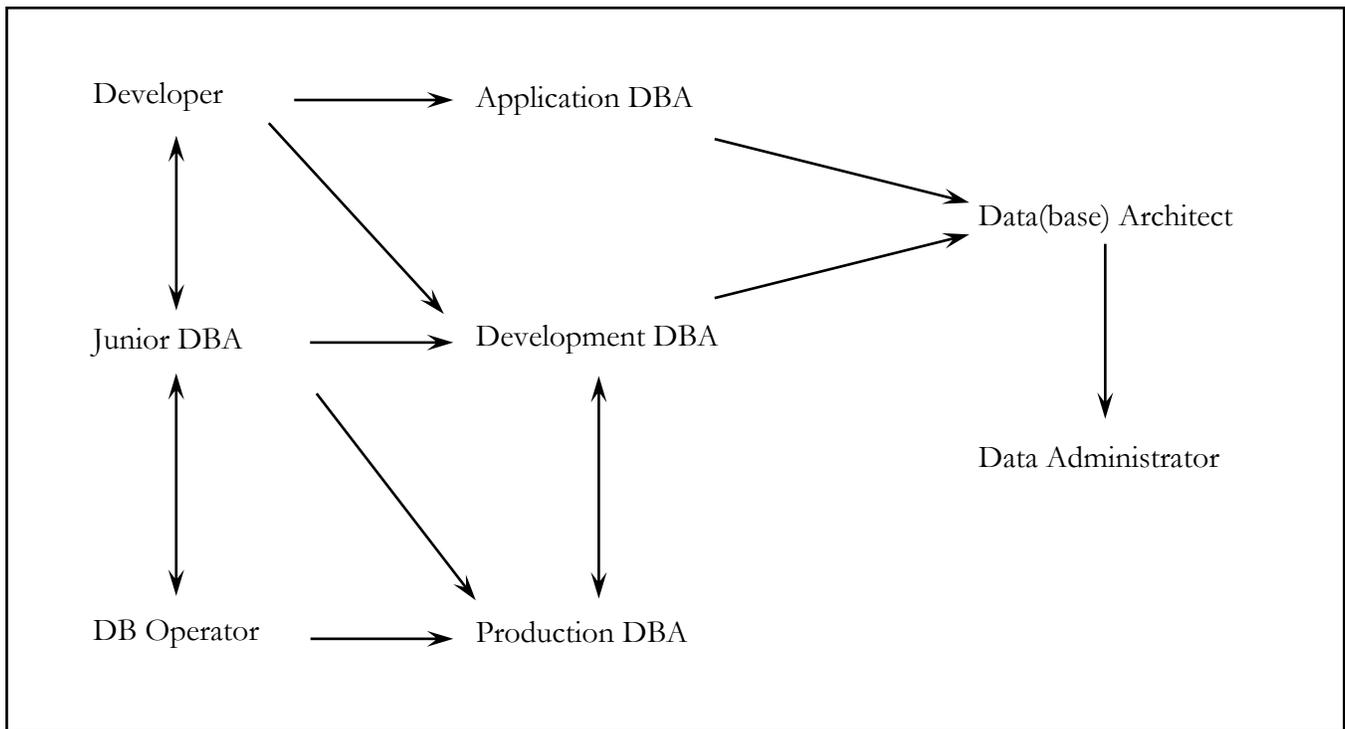


Figure 2, Common DBA Career Paths

## WHEN DO I NEED A DBA?

Not every organization needs the various roles filled at the same time. Figure 3 attempts to show at what points in a system lifecycle various roles might be needed. This figure is based on ServerCare's 7 "D's" of Database Construction:

1. Determine What are you building, and why? Strategize. Determine the business case and get buy-in
2. Define Begin analysis. Develop logical model. Define business data and processes.
3. Design Begin development of physical model. Begin definition of application. Some proofs of concept.
4. Develop Begin code development in earnest. Physical design should be mostly complete.
5. Document Begin documentation. Done concurrently with Development.
6. Deploy Define implementation strategy. Test application and rollout. Possible parallel run. Plan production
7. Do! Production. Ongoing maintenance, monitoring, tuning

There is actually an 8<sup>th</sup> "D" - Death. All applications will stop being used at some point.

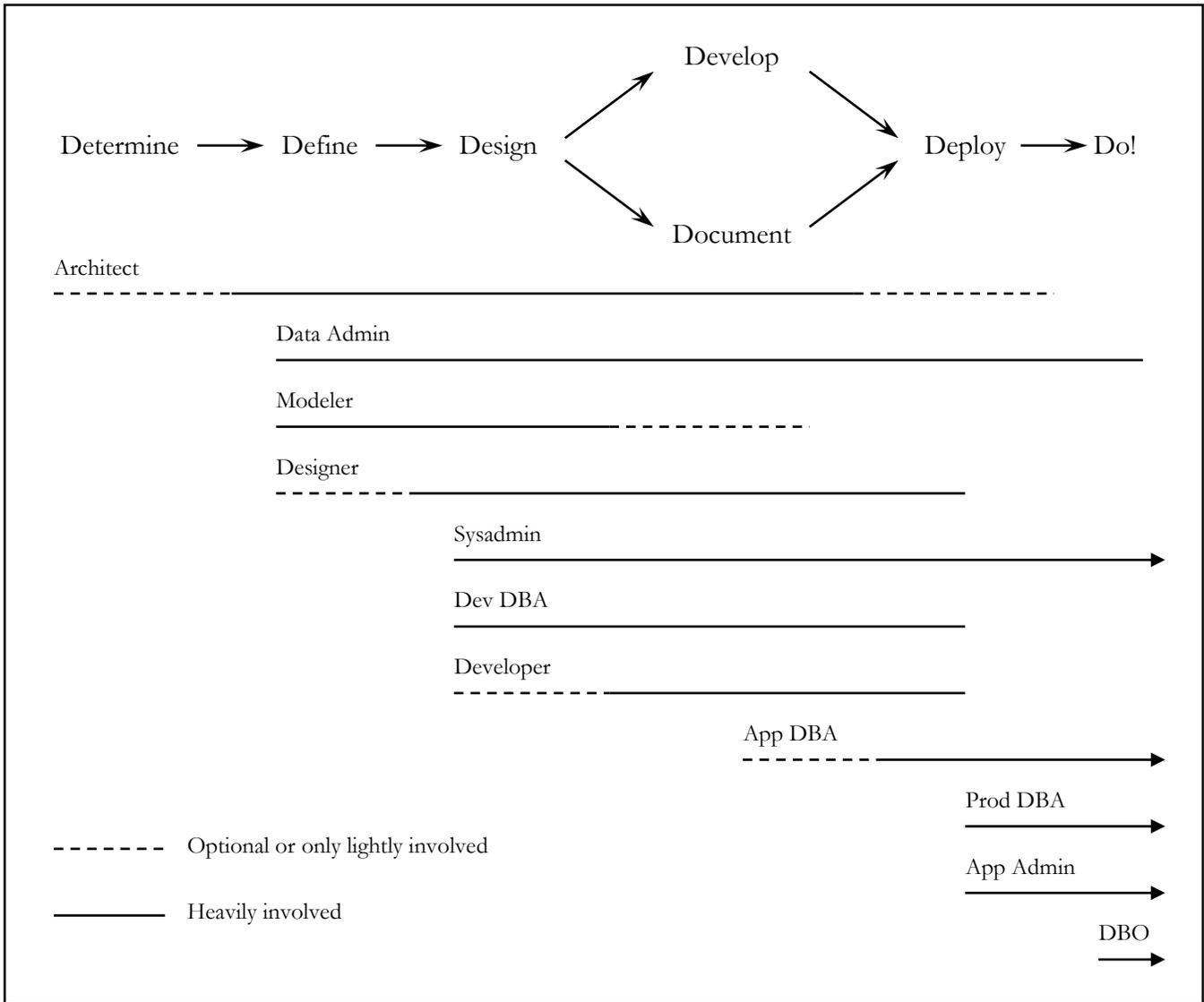


Figure 3 The 7 "D's" of Database Construction, and Roles Assigned to Each.

## **WHAT DO I NEED?**

Now that we have defined the roles, the big question is, what does my environment need? This presentation should help you understand your needs and your options, but only you can know what will work best for your situation. The good news is that by knowing the stages of your platform, and the type of usage it does or will experience, you can determine when you will need what type of database administration. You may very well find that the skills that are available to you now are sometimes enough to supplant the DBA, or allow the bulk of DBA tasks to be outsourced.

## DETERMINING YOUR NEEDS

This checklist may help you gauge your needs:

- 1 Specify your environment *Development, Test, or Production?*
- 2 Estimate complexity and size *Lots of Tables? Lots of relationships? Mb, Gb, or Tb big?*
- 3 Number of databases/instances in each server
- 4 Heterogeneous systems increase complexity
- 5 Specify downtime tolerance *99.999% = 5 minutes/year*
- 6 High availability *8x5, 24x7, 24x365?*
- 7 Redundancy or distribution *Standby databases? Clusters?*
- 8 Determine what skills already exist
- 9 Are in-house skills willing to take on additional roles?
- 10 Create responsibilities that are challenging but not overwhelming
- 11 What DBA tools do you have? *But tools can only partially make up for an inexperienced DBA*
- 12 Last, what are you willing to spend? *Critical systems need \$\$\$'s!*

## DBA OPTIONS

The classic idea is that if you need help, hire someone. However, the Oracle market has historically been very hot, and finding experienced people that are willing to take and stay on with “normal” jobs is difficult. Even junior people take what knowledge they can and jump ship. Contractors are available, but expensive. Outsourcing can be a non-traditional but excellent choice. Here are some of the available options:

- 1 Hire an employee
  - Pro: Can spend time to really learn the application(s) and business, current market is excellent for employers
  - Con: Difficult to find employees with experience, high overhead costs, turnaround is always an issue
- 2 Hire a contractor
  - Pro: Usually experienced, can "hit the ground running", can be let go easily
  - Con: Very expensive, may lose interest, may be there just to add to skill set/resume
- 3 Hire a part-time DBA
  - Pro: As with a contractor, usually experienced, costs can be controlled
  - Con: Can still be expensive, very difficult to find, may not be there when needed
- 4 Distribute workload to multiple employees
  - Pro: Great opportunity for interested employees
  - Con: Possible disgruntlement, learning curve, challenge to manage, desire and ability may not match
- 5 DBA Tools
  - Pro: Can automate many tasks, can alarm when there are problems
  - Con: Must be set up, limited, still need human intervention
- 6 Mentor or Train-up inexperienced personnel with highly skilled ones
  - Pro: Excellent long-term potential, provides a safety net during tutelage
  - Con: Can be expensive during dual-DBA term, still risk of turnover
- 7 Outsource (complete IT outsource or Remote DBA's - shared access to expert-level DBA's)
  - Pro: May be least expensive, can complement above, greatest flexibility (can pick and choose services), no learn-and-leave, accountability, no overhead
  - Con: Must have connectivity, solid communication is essential, Remote DBA must not just be a side business

## 8 Mix-n-match above

- Pro: Gets best of the above
- Con: Might get some of the worst of the above too!

## **PUTTING IT ALL TOGETHER**

Using the above, here are some suggestions based on size of the environment:

- In massive or highly critical systems, onsite DBA(s) are a must, but many tasks can be offloaded to others to minimize number of onsite DBA's and cost. For example, an onsite DBA could be responsible for strategic and long-term issues, and an outsourced DBA could deal with day-to-day issues
- In moderate servers, large single instance systems, or multiple small platforms, a mix-n-match approach may be best
- Small systems have the most options but generally the least money. Remote DBA's that mentor in-house staff can be particularly effective for systems that are still in the pre-deployment or initial growth stages.
- The current market is excellent for finding less costly employees, but that also means that outsourcing and contractors are less expensive as well. The market will again become tight at some point, so be prepared.
- A new but interesting concept is "part-time permanent," which means that personnel are onsite for only a few hours a week, and called upon remotely as needed. So far this model has only been offered for PC and Networking support, but it is only a matter of time before more specialized services such as systems and database administration will be available as well.

## **CONCLUSION**

Understanding what a DBA does and can do, for *your* specific environment, is key. To do that, you also need to know your own environment. Get outside help to determine your requirements, if you need to. Realize that the 9-to-5, onsite employee is a thing of the past - is your business still 9-to-5? Does it happen in just one place? If you are already using or thinking about using technologies such as WebDB, iDatabases, or e-commerce, then you are probably already aware that classic business functions and roles cannot operate in Internet Time. Having the proper resources in place ahead of time, and knowing when those resource needs will change, will enable your enterprise to compete in the new "online, all the time" world.

## **WHERE TO GET MORE INFORMATION**

- Other training sessions
- The Internet, of course
- Consulting companies (depending on specialization)
- Get someone in just to look over things, if nothing else. A good contractor will be willing to do a quick once-over