ch01_1-48.qxd 5/22/02 1:24 PM Page 1

1

What Is a DBA?

The need for a database administrator is greater today than ever before.

Every organization using a database management system (DBMS) to manage data requires a database administration group to ensure the effective use and deployment of the company's databases. Since most modern organizations of any size use a DBMS, the need for a database administrator (DBA) is greater today than ever before. However, the discipline of database administration is neither well understood nor universally practiced in a coherent and easily replicated manner.

The DBA: Revered or Reviled?

An oft-repeated story about database administration underscores both the necessity for database administration and the lack of understanding of a DBA's function. It goes something like this:

The CIO of Acme Corporation hires a management consulting company to streamline their information technology (IT) operations. The consultant, determined to understand the way Acme works, begins by interviewing the CIO. One of his first questions is: "So, I see that you have a DBA on staff. What does he do?"

ch01_1-48.qxd 5/22/02 1:24 PM Page 2

> The CIO replies, "Well, I'm told that we need the DBA to make sure our Oracle databases stay online. I know that some of our critical business processes like order entry and inventory use Oracle, but I really don't know what the DBA does. But please don't tell me I need another one, because we can barely afford to pay the one we have!"

This is a sad but too often true commentary on the state of database administration in many organizations. DBMS software is so complex these days that very few people understand more than just the basics (like SQL). However, DBAs understand the complexities of the DBMS, making them a valuable resource. Indeed, sometimes the only source of database management and development knowledge within the organization is the DBA.

The DBA, often respected as a database guru, is just as frequently criticized as a curmudgeon with vast technical knowledge but limited people skills. Just about every database programmer has his or her favorite DBA story. You know, those anecdotes that begin with "I had a problem..." and end with "and then he told me to stop bothering him and read the manual." DBAs simply do not have a "warm and fuzzy" image. However, this perception probably has more to do with the nature and scope of the job than with anything else. The DBMS spans the enterprise, effectively placing the DBA on call for the applications of the entire organization.

The truth is, many database problems require periods of quiet reflection and analysis for the DBA to resolve. Therefore, DBAs generally do not like to be disturbed. However, due to the vast knowledge most DBAs possess (the guru, again), their quiet time is usually less than quiet; constant interruptions to answer questions and solve problems is a daily fact of life.

DBAs need to acquire exceptional communication skills.

DBAs, more than most, need to acquire exceptional communication skills. Data is the lifeblood of computerized applications. Application programs are developed to read and write data, analyze data, move data, perform calculations using data, modify data, and so on. Without data, there would be nothing for the programs to do. The DBA is at the center of the development life cycle—ensuring that application programs have efficient, accurate access to the corporation's data. As such, DBAs frequently interface with many different types of people: technicians, programmers, end users, customers, and executives. However, many DBAs are so caught up in the minutiae of the inner workings of the DBMS that they never develop the skills required to relate appropriately to their coworkers and customers.

ch01_1-48.qxd 5/22/02 1:24 PM Page 3

A DBA ensures the ongoing operational functionality and efficiency of an organization's databases and applications.

However, we have not yet answered the question: What is a DBA? The short answer is simple: A DBA is the information technician responsible for ensuring the ongoing operational functionality and efficiency of an organization's databases and the applications that access those databases.

The long answer to that question requires a book to answer—this book. This text will define the management discipline of database administration and provide practical guidelines for the proper implementation of the DBA function.

Why Learn Database Administration?

Data is at the center of today's applications; today's organizations simply cannot operate without data. In many ways, business today *is* data. Without data, businesses would not have the ability to manage finances, conduct transactions, or contact their customers. Databases are created to store and organize this data. The better the design and utility of the database, the better the organization will be positioned to compete for business.

Indeed, one of the largest problems faced by IT organizations is ensuring quality database administration. A survey of IT managers conducted by *Information Week* in December 2000 showed that the top two database management execution issues faced by companies are ease of administration and availability of qualified administrators.

Both of these issues were cited by 58% of survey respondents. Additionally, the 1999 Market Compensation Survey conducted by people[3], a Gartner Company, shows that DBA positions take longer to fill than any other position. Clearly, there is no lack of demand for DBA skills in today's job market.

A Unique Vantage Point

A good DBA needs to enjoy challenges and be a good problem solver.

The DBA is responsible for designing and maintaining an enterprise's databases, placing the DBA squarely at the center of the business. The DBA has the opportunity to learn about many facets of business and how they interrelate. The DBA can explore groundbreaking technologies as they are adopted by the organization. Exposure to new technology keeps the job stimulating—but frustrating if you are trying to figure out how a new technology works for the first time. The DBA is often working alone in these endeavors; he does not have access to additional expertise to assist when troubles arise. Therefore, a good DBA needs to enjoy challenges and be a good problem solver.

ch01_1-48.qxd 5/22/02 1:24 PM Page 4

DBA Salaries

You can find no more challenging job in IT than database administration. Fortunately, DBAs are well paid. DICE.com, a career planning and research Web site, provides valuable statistics on DBA compensation. For example, database administration is one of the top ten contract jobs when ranked by salary, as well as one of the top ten jobs for full-time employment. The mean compensation for DBA consultants is $81 per hour; the mean level of experience just 4.98 years. For full-time employees with four or more years of experience, the mean salary ranges from the low $60,000s to over $80,000. Figure 1-1 shows the mean salary for full-time DBAs broken down by years of experience.

Another Web site, searchdatabase.com, a portal of database information for IT professionals, conducted a salary survey of database professionals. As of late January 2001, the average annual salary for all database professionals was more than $62,000. As might be expected, as the years of experience and the number of people managed increases, so does the salary. Of course, DBA salaries, as with all salaries, vary by region of the country. In the United States, DBA salaries are usually higher in the Northeast and on the West Coast than in other regions.

Database administration is a nonstop job.

So, DBAs are well paid, have challenging jobs, and are likely to be engaged in the most visible and important projects. What's not to like? Well, DBAs are expected to know everything, not just about database technologies, but about any-

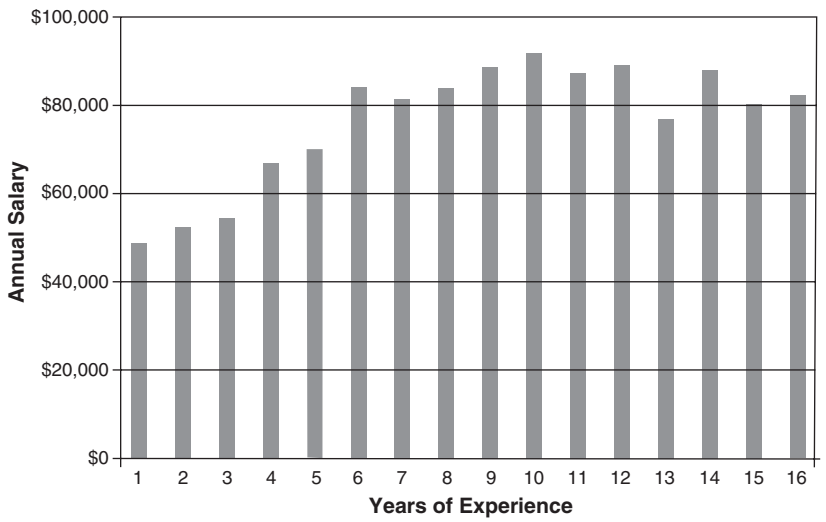

Figure 1-1 *Mean salary for full-time DBAs (Dice.com 2000 IT Rate Survey)*

ch01_1-48.qxd 5/22/02 1:24 PM Page 5

thing remotely connected to them. Database administration is a nonstop job, and DBAs work long days with lots of overtime, especially when performance is suffering or development projects are behind schedule. DBAs frequently have to work on weekends and holidays to maintain databases during off-peak hours. A DBA must be constantly available to deal with problems, because database applications run around the clock. Most DBAs carry pagers or cell phones so they can be reached at any time. If there is a database problem at 2:00 A.M., the DBA must get out of bed, clear his head, and solve the problem to get the applications back up and running. Failure to do so can result in database downtime, and that can completely shut down business processes. DBAs frequently spend weekends in front of the computer performing database maintenance and reorganizations during off peak hours. You can't bring mission-critical databases down during the nine-to-five day to maintain them. According to industry analysts at the META Group, the average DBA works more than fifty hours per week, including an average of six hours on weekends.

So, database administration is technically challenging and rewarding; it can also be exhausting and frustrating. But don't let that scare you. The positive aspects of the job far outweigh the negative.

Database Technology

This book assumes a basic knowledge of relational database technology and DBMS fundamentals. For readers needing to review these concepts, please refer to Appendix 1. This is not a trivial matter; people sometimes think they know more than they actually do. For example, what is a database? I'll bet most readers believe they know the answer to that question. However, some (perhaps many) of you would be wrong. Oracle is not a database; it is a database management system. You can use Oracle to create a database, but Oracle, in and of itself, is not a database.

A database is an organized store of data wherein the data is accessible by named data elements.

So, what is a database? A *database* is an organized store of data wherein the data is accessible by named data elements (for example, fields, records, and files). And what is a database management system? A *DBMS* is software that enables end users or application programmers to share and manage data. It provides a systematic method of creating, updating, retrieving, and storing information in a database. A DBMS is also generally responsible for data integrity, data security, data access control and optimization, automated rollback, restart, and recovery. Figure 1-2 shows the relationship between a DBMS and a database.

ch01_1-48.qxd 5/22/02 1:24 PM Page 6

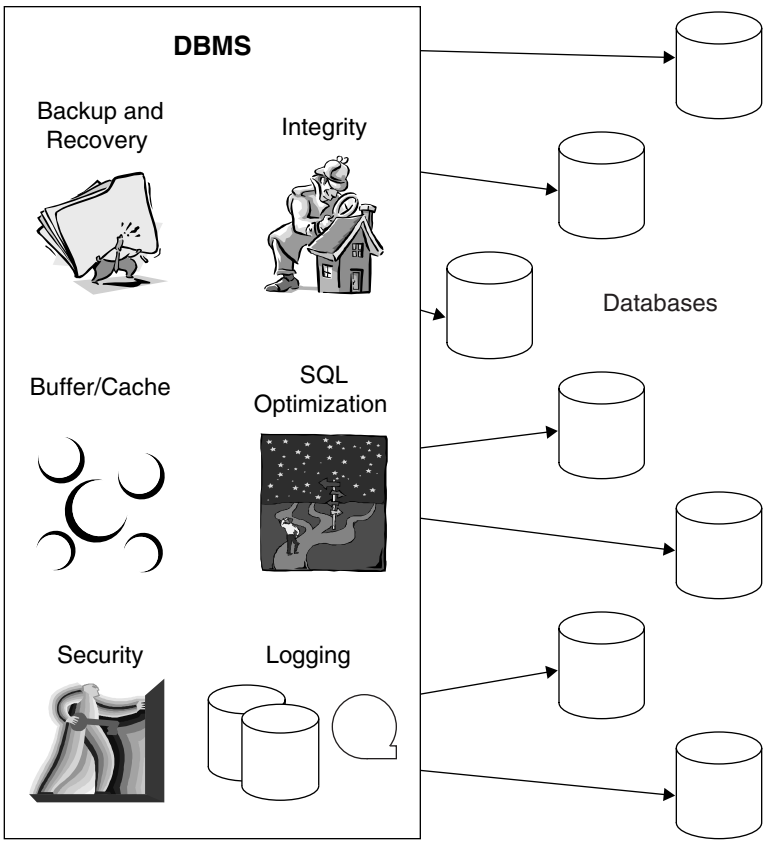

Figure 1-2 *Relationship of DBMS to database*

You might think of a database as a file folder, and a DBMS as the file cabinet holding the labeled folders. You implement and access database instances using the capabilities of the DBMS. Your payroll application uses the payroll database, which may be implemented using a DBMS such as DB2, Oracle9i, or SQL Server.

Why is this important? If we don't use precise terms in the workplace, confusion can result. And confusion leads to over-budget projects, improperly developed systems, and lost productivity.

In addition to database management fundamentals, DBAs must be experts in the specific DBMS in use, and there may be many in the organization. For example, a large organization might use DB2 on the mainframe, Oracle and Informix on several different UNIX platforms, and SQL Server on Windows

ch01_1-48.qxd 5/22/02 1:24 PM Page 7

2000. Older legacy systems might use IMS databases, and then there is that one crazy application out there that uses a fringe DBMS like Adabas or Ingres.[1]

DBAs must implement decisions based on the best fit of application, DBMS, and platform.

The DBA group, therefore, must have expertise in each of these different management systems and platforms. Furthermore, the DBA must be capable of determining which DBMS and platform is best suited to the needs of each application. This can be a difficult job fraught with politics and conflicting opinions. The DBA group must be able to act impartially and implement decisions based on the best fit of application, DBMS, and platform.

Once again, for a short introduction to DBMS concepts, refer to Appendix 1.

The Management Discipline of Database Administration

Database administration is rarely approached as a management discipline. The term *discipline* implies a plan, and implementation according to that plan. When database administration is treated as a management discipline, the treatment of data within your organization will improve. It is the difference between being reactive and proactive.

All too frequently, the DBA group is overwhelmed by requests and problems. This ensues for many reasons, such as understaffing, overcommitment to supporting new (and even legacy) application development projects, lack of repeatable processes, or lack of budget. The reactive DBA functions more like a firefighter than an administrator; he attempts to resolve problems only after problems occur. The reactive DBA is focused on resolving the biggest problem confronting him.

A proactive DBA develops and implements a strategic blueprint for deploying databases within the organization.

In contrast, the proactive DBA implements practices and procedures to avoid problems before they occur. A proactive database administrator develops and implements a strategic blueprint for deploying databases within the organization. This plan should address all phases of the application development life cycle. A data specialist, usually the DBA, should be involved during each phase of the cycle, as shown in Figure 1-3. During the initiation and requirements gathering phase, the DBA must be available to identify the data components of the project. He can help to determine if the required data already exists elsewhere

1. I refer to Adabas and Ingres as "fringe," not because of any functional or technical deficiencies, but only because of their minimal marketshare.

ch01_1-48.qxd 5/22/02 1:24 PM Page 8

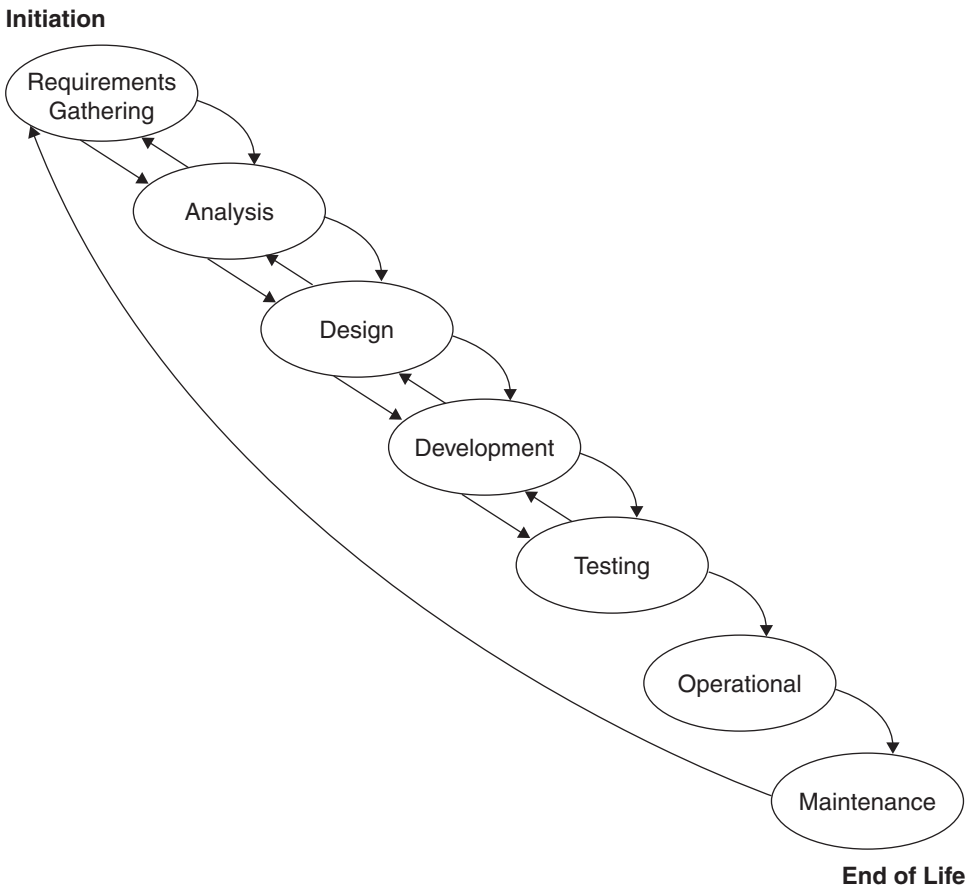

Figure 1-3 *The application development life cycle*

in the organization or if the data is brand new. During the analysis and design phases, the rudimentary data requirements must be transformed into a conceptual and logical data model.

Before development can begin, the logical data model must be translated to a physical database design that can be implemented using a DBMS such as Oracle or DB2. Sample data must be populated into the physical database to allow application testing. Furthermore, the DBA must develop and implement a process to refresh test data to enable repeatable test runs.

When the application moves from development to operational status, the DBA ensures that the DBMS is prepared for the new workload. This preparation includes implementing appropriate security measures, measuring and

ch01_1-48.qxd 5/22/02 1:24 PM Page 9

modifying the storage and memory requirements for the new application, and anticipating the impact of the new workload on existing databases and applications. The DBA is also responsible for migrating the new database from the test environment to the production environment.

While the application is operational, the DBA performs a host of duties including assuring availability, performance monitoring, tuning, backup and recovery, and authorization management. However, no application or database remains static for long. Because business needs change over time, the IT systems that support the business will also change. When maintenance is requested, the DBA becomes engaged in the entire process once again, from requirements gathering to implementation.

Finally, when the application reaches the end of its useful life, the DBA must help to determine the final status of the data used by the application. Is the data no longer required, or do other applications and processes use the data, too? Are there regulations that require the data to be stored longer than the life of the application? Does the business have any privacy policies that impose special rules for handling the data? See the sidebar "Privacy Policies and Data" for further information.

Privacy Policies and Data

The recent bankruptcy of e-business toy seller Toysmart.com provides a good lesson in the impact of privacy policies on corporate data. In May 2000, Toysmart filed for bankruptcy and announced its intention to sell its database of customer information. Toysmart's customer list was estimated to contain information on 250,000 former customers, including their names, phone numbers, street and e-mail addresses, and product preferences. However, Toysmart's own privacy policy, previously posted on its Web site, promised that it would not disclose the personal information of its customers to third parties.

The FTC and a group of state attorneys general blocked the sale on the grounds that the sale would violate Toysmart's privacy policy. They argued that Toysmart's customers conducted business with Toysmart.com under the conditions of the privacy policy. The court ruling further stipulated that the company had to provide an affidavit describing how the list was destroyed.

This is just one example of how privacy policies can impact database administrators and corporate data experts. Of course, you may never work for a company that goes bankrupt, but your company may decide to retire applications and data due to legal regulations, business conditions, or mergers.

ch01_1-48.qxd 5/22/02 1:24 PM Page 10

The DBA is responsible for managing the overall database environment. Often this includes installing the DBMS and setting up the IT infrastructure to allow applications to access databases. These tasks need to be completed before any application programs can be implemented. Furthermore, ad hoc database access is a requirement for many organizations.

Additionally, the DBA is in charge of setting up an ad hoc query environment that includes evaluating and implementing query and reporting tools, establishing policies and procedures to ensure efficient ad hoc queries, and monitoring and tuning ad hoc SQL.

A good DBA is integral to the entire application development life cycle.

As you can see, a good DBA is integral to the entire application development life cycle. The DBA is "in demand" for his knowledge of data and the way in which data is managed by modern applications.

A Day in the Life of a DBA

A day in the life of a DBA is usually quite hectic. The DBA maintains production and test environments, monitors active application development projects, attends strategy and design meetings, selects and evaluates new products, and connects legacy systems to the Web. And, of course: *Joe in Accounting, he just resubmitted that query from hell that's bringing the system to a halt. Can you do something about that?* All of this can occur within a single workday.

To add to the chaos, DBAs are expected to know everything about everything. From technical and business jargon to the latest management and technology fads, the DBA is expected to be "in the know." And do not expect any private time: A DBA must be prepared for interruptions at any time to answer any type of question—and not just about databases, either.

When application problems occur, the database is "guilty until proven innocent."

When application problems occur, the database environment is frequently the first thing blamed. The database is "guilty until proven innocent." A DBA is rarely approached with a question like "I've got some really bad SQL here. Can you help me fix it?" Instead, the DBA is forced to investigate problems where the underlying assumption is that the DBMS or perhaps the DBA is at fault, when the most common cause of relational performance problems is inefficiently coded applications.

Oftentimes the DBA is forced to prove that the database is not the source of the problem. The DBA must know enough about all aspects of IT to track down errors and exonerate the DBMS and database structures he has designed. So he must be an expert in database technology, but also have semi-expert knowledge of the IT components with which the DBMS interacts: application programming

ch01_1-48.qxd 5/22/02 1:24 PM Page 11

languages, operating systems, network protocols and products, transaction processors, every type of computer hardware imaginable, and more. The need to understand such diverse elements makes the DBA a very valuable resource. It also makes the job interesting and challenging. If database administration still sounds intriguing to you, read on. Actually, the job isn't as bad as it sounds. The work is interesting, there is always something new to learn, and, as previously mentioned, the pay can be good. The only question is: Can anyone do this type of job for twenty or more years without needing a rest? And, oh, by the way, I think I hear your pager going off, so you might want to pause here to see what is wrong.

Evaluating a DBA Job Offer

As a DBA, it is almost inevitable that you will change jobs several times during your career. When making a job change, you will obviously consider requirements such as salary, bonus, benefits, frequency of reviews, and amount of vacation time. However, you also should consider how the company treats their DBAs. Different organizations place different value on the DBA job. It is imperative to your career development that you scout for progressive organizations that understand the complexity and ongoing learning requirements for the position. Here are some useful questions to ask:

- Does the company offer regular training for its DBAs to learn new DBMS features and functionality? What about training for related technologies such as programming, networking, e-business, transaction management, message queueing, and the like?
- Does the company allow DBAs to regularly attend local user groups? What about annual user groups at remote locations?
- Are there backup DBAs, or will you be the only one on call 24/7?
- Are there data administration and system administration organizations, or are the DBAs expected to perform all of these duties, too?
- Does the DBA group view its relationship with application development groups as a partnership? Or is the relationship more antagonistic?
- Are DBAs included in design reviews, budgeting discussions, and other high-level IT committees and functions?

The more "yes" answers you get to these questions, the more progressive the DBA environment is.

ch01_1-48.qxd 5/22/02 1:24 PM Page 12

Database, Data, and System Administration

Many organizations combine data administration into the database administration role.

Some organizations define separate roles for the business aspects and the technical aspects of data. The business aspects of data are aligned with data administration, whereas the more technical aspects are handled by database administration. Not every organization has a data administration function. Indeed, many organizations combine data administration into the database administration role.

Sometimes organizations also split up the technical aspects of data management, with the DBA responsible for using the DBMS and a system administrator or systems programmer responsible for installing and upgrading the DBMS.

Data Administration

Data administration separates the business aspects of data resource management from the technology used to manage data; it is more closely aligned with the actual business users of data. The data administrator (DA) is responsible for understanding the business lexicon and translating it into a logical data model. Referring back to the ADLC, the DA would be involved more in the requirements gathering, analysis, and design phase, the DBA in the design, development, testing, and operational phases.

Another difference between a DA and a DBA is the focus of effort. The DA is responsible for the following tasks:

- Identifying and cataloging the data required by business users
- Producing conceptual and logical data models to accurately depict the relationship among data elements for business processes
- Creating an enterprise data model that incorporates all of the data used by all of the organization's business processes
- Setting data policies for the organization
- Identifying data owners and stewards
- Setting standards for control and usage of data

The data administrator can be thought of as the Chief Data Officer of the corporation.

In short, the DA can be thought of as the Chief Data Officer of the corporation. However, in my experience, the DA is never given an executive position, which is unfortunate. Many IT organizations state that they treat data as a corporate asset, a statement that is belied by their actions. Responsibility for data policy is often relegated to technicians who fail to concentrate on the non-

ch01_1-48.qxd 5/22/02 1:24 PM Page 13

technical business aspects of data management. Technicians do a good job of ensuring availability, performance, and recoverability, but are not usually capable of ensuring data quality and setting corporate policies.

In fact, data is rarely treated as a true corporate asset. Think about the assets that every company has in common: capital, human resources, facilities, and materials. Each of these assets is modeled: charts of account, organization charts, reporting hierarchies, building blueprints, office layouts, and bills of material. Each is tracked and protected. Professional auditors are employed to ensure that no discrepancies exist in a company's accounting of its assets. Can we say the same thing about data?

A mature DA organization is responsible for planning and guiding the data usage requirements throughout the organization. This role encompasses how data is documented, shared, and implemented companywide. A large responsibility of the DA staff is to ensure that data elements are documented properly, usually in a data dictionary or repository. This is another key differentiation between a DA and a DBA. The DA focuses on the repository, whereas the DBA focuses on the physical databases and DBMS.

Furthermore, the DA deals with metadata, as opposed to the DBA, who deals with data. *Metadata* is often described as data about data; more accurately, metadata is the description of the data and data interfaces required by the business. Data administration is responsible for the business's metadata strategy. Examples of metadata include the definition of a data element, business names for a data element, any abbreviations used for that element, and the data type and length of the element. Data without metadata is difficult to use. For example, the number 12 is data, but what kind of data? In other words, what does that 12 mean? Without metadata, we have no idea. Consider this: Is the number 12

- A date representing December, the twelfth month of the year?
- A date representing the twelfth day of some month?
- An age?
- A shoe size?
- Or, heaven forbid, an IQ?

And so on. However, there are other, more technical aspects of metadata, too. Think about the number 12 again.

- Is 12 a large number or a small one?

ch01_1-48.qxd 5/22/02 1:24 PM Page 14

- What is its domain (that is, what is the universe of possible values of which 12 is but a single value)?
- What is its data type? Is it an integer or a decimal number with a 0 scale?

Metadata provides the context by which data can be understood and therefore become information.

Metadata provides the context by which data can be understood and therefore become information. In many organizations, metadata is not methodically captured and cataloged; instead, it exists mostly in the minds of the business users. Where it has been captured in systems, it is spread throughout multiple programs in file definitions, documentation in various states of accuracy, or in long lost program specifications. Some of it, of course, is in the system catalog of the DBMS.

A comprehensive metadata strategy enables an organization to understand the information assets under its control and to measure the value of those assets. Additional coverage of metadata is provided in Chapter 21.

One of the biggest contributions of data administration to the corporate data asset is the creation of data models. A *conceptual data model* outlines data requirements at a very high level. A *logical data model* provides in-depth details of data types, lengths, relationships, and cardinality. The DA uses normalization techniques to deliver sound data models that accurately depict the data requirements of an organization.

Many DBAs dismiss data administration as mere data modeling, required only because someone needs to talk to the end users to get the database requirements. However, a true DA function is much more than mere data modeling. It is a business-oriented management discipline responsible for the data asset of the organization.

Why spend so much time talking about data administration in a book about database administration? Well, very few organizations have implemented and staffed a DA role. The larger the organization is, the more likely that a DA function exists. However, when the DA role is undefined in an organization, the DBA must assume the mantle of data planner and modeler. Unfortunately, the DBA will usually not be able to assume all of the functions and responsibility of a DA as summarized in this section for a number of reasons:

- The DBA has many other technical duties to perform that will consume most of his time.
- The manager of the DBA group typically does not have an executive position enabling him to dictate policy.
- The DBA generally does not have the skills to communicate effectively with business users and build consensus.

ch01_1-48.qxd 5/22/02 1:24 PM Page 15

- Frankly, most DBAs are happier dealing with technical issues and technicians than with business issues and nontechnicians.

When DA and DBA functions coexist within the organization, the two groups must work very closely with one another. It is not necessary that both have the same manager, though it would facilitate cooperation. At any rate, it is imperative that some skills cross-pollinate the two groups. The DA will never understand the physical database like a DBA, and the DBA will never understand the business issues of data like a DA, but each job function is more effective with some knowledge about the other.

Organizations truly concerned about data quality, integrity, and reuse will invariably implement and staff the DA function.

In short, organizations that are truly concerned about data quality, integrity, and reuse will invariably implement and staff the DA function.

Database Administration

Database administration is the focus of this entire book, so I will not spend a lot of time defining it in this short section. The rest of the book will accomplish that nicely. This section will quickly outline the functions performed by the DBA group when the DA function exists. The first duty of the DBA is to understand the data models built by the DA and to communicate the model to the application developers and other appropriate technicians. The logical data model is the map the DBA will use to create physical databases. The DBA will transform the logical data model into an efficient physical database design. It is essential that the DBA incorporate his knowledge of the DBMS to create an efficient and appropriate physical database design from the logical model. The DBA should not rely on the DA for the final physical model any more than a DA should rely on a DBA for the conceptual and logical data models. Figure 1-4 depicts this relationship.

The DBA is the conduit for communication between the DA team and the technicians and application programming staff.

The DBA is the conduit for communication between the DA team and the technicians and application programming staff. Of course, the bulk of the DBA's job is ongoing support of the databases created from the physical design and management of the applications that access those databases. An overview of these duties is provided in the DBA Tasks section of this chapter.

System Administration

Some organizations, usually the larger ones, also have a system administrator (SA) or systems programming role that impacts DBMS implementation and operations. The SA is responsible for the installation and setup of the DBMS.

ch01_1-48.qxd 5/22/02 1:24 PM Page 16

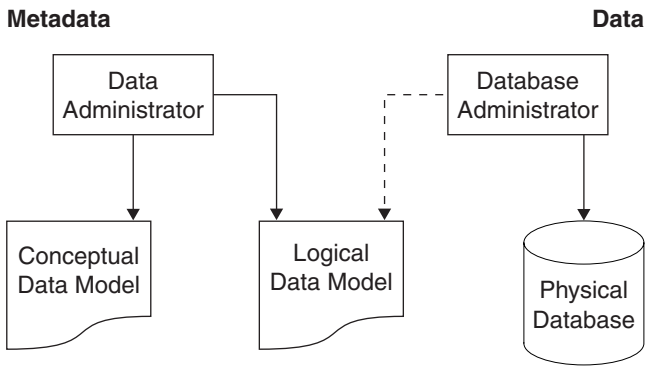

Figure 1-4 *DBA vs. DA*

The system administrator ensures that the IT infrastructure is operational for database development by setting up the DBMS appropriately, applying ongoing maintenance from the DBMS vendor, and coordinating migration to new DBMS releases and versions.

The SA typically has no responsibility for database design and support. Instead, the DBA is responsible for the databases and the SA is responsible for DBMS installation, modification, and support. (If this distinction is not clear to you, please refer to Appendix 1.)

Furthermore, the SA ensures that the IT infrastructure is implemented such that the DBMS is configured to work with other enabling system software. The SA may need to work with other technicians to configure transaction processors, message queueing software, networking protocols, and operating system parameters to enable the DBMS to operate effectively. The SA ensures that the IT infrastructure is operational for database development by setting up the DBMS appropriately, applying ongoing maintenance from the DBMS vendor, and coordinating migration to new DBMS releases and versions.

As with data administration, there must be cross-training of skills between the SA and DBA. The SA will never understand the physical database like the DBA, but the DBA is unlikely to understand the installation and in-depth technical relationships of system software like the SA. Each job function will be more effective with some knowledge of the other.

If no system administration group exists, or if its focus is not on the DBMS, the DBA assumes responsibility for DBMS-related system administration and programming. Figure 1-5 delineates the responsibilities of the DA, the DBA, and the SA.

DBA Tasks

Ensuring that an organization's data and databases are useful, usable, available, and correct requires the DBA to perform a variety of tasks in a variety of areas. These

ch01_1-48.qxd 5/22/02 1:24 PM Page 17

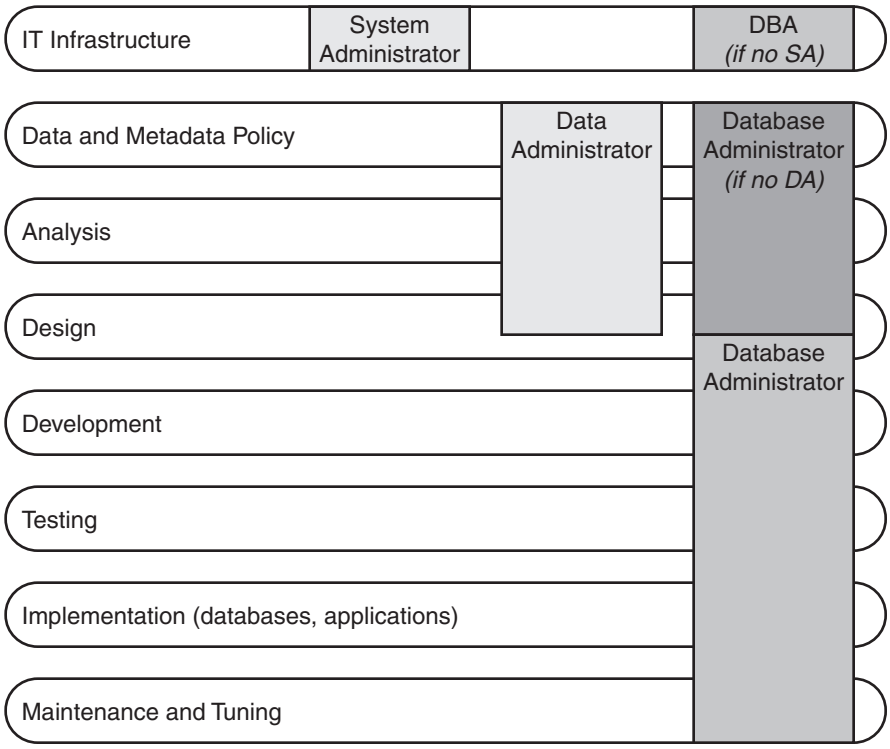

Figure 1-5 *DA, DBA, and SA responsibilities*

areas include database design, performance monitoring and tuning, database availability, security, backup and recovery, data integrity, release migration—really, anything that involves the company's databases. Let's examine each of these topics.

Database Design

To properly design and create relational databases, the DBA must understand and adhere to sound relational design practices. The DBA must understand both relational theory and the specific implementation of the relational database management system (RDBMS) he's using to create the database. Database design requires a sound understanding of conceptual and logical data modeling techniques. The ability to create and interpret entity-relationship diagrams is essential to designing a relational database.

The DBA must be able to transform a logical data model into a physical database implementation. The DBA must ensure that the database design and

ch01_1-48.qxd 5/22/02 1:24 PM Page 18

implementation will enable a useful database for the applications and clients that will use it.

Although database design is a significant skill for the DBA to possess, the job of the DBA is often disproportionately associated with database design. Although designing optimal databases is important, it is a relatively small portion of the DBA's job. A DBA will most likely spend more time administering and tuning databases than in designing and building databases.

A poor relational design can result in poor performance.

By no means, though, should you interpret this to mean that database design is not important. A poor relational design can result in poor performance, a database that does not meet the needs of the organization, and potentially inaccurate data.

Performance Monitoring and Tuning

What is meant by database performance? Let's use the familiar concept of supply and demand. Users demand information from the database, and the DBMS supplies this demand for information. The rate at which the DBMS supplies the information can be termed *database performance*. However, it is not really that simple. Five factors influence database performance: workload, throughput, resources, optimization, and contention.

The *workload* that is requested of the DBMS defines the demand. It is a combination of online transactions, batch jobs, ad hoc queries, data warehousing, analytical queries, and commands directed through the system at any given time. Workload can fluctuate drastically from day to day, hour to hour, minute to minute, and even second to second. Sometimes workload can be predicted (such as heavy month-end processing of payroll, or very light access after 7:30 P.M., when most users have left for the day), but at other times it is unpredictable. The overall workload has a major impact on database performance.

Throughput defines the overall capability of the computer hardware and software to process data. It is a composite of I/O speed, CPU speed, parallel capabilities of the machine, and the efficiency of the operating system and system software. The hardware and software tools at the disposal of the system are known as the *resources* of the system. Examples include the database kernel, disk space, cache controllers, and microcode.

Optimization refers to the analysis of database requests with query cost formulas to generate efficient access paths to data. All types of systems can be optimized, but relational queries are unique in that optimization is primarily accomplished internal to the DBMS. However, many other factors need to be op-

ch01_1-48.qxd 5/22/02 1:24 PM Page 19

timized (SQL formulation, database parameters, programming efficiently, and so on) to enable the database optimizer to create the most efficient access paths.

When the demand (workload) for a particular resource is high, *contention* can result. Contention is the condition in which two or more components of the workload are attempting to use a single resource in a conflicting way (for example, dual updates to the same piece of data). As contention increases, throughput decreases.

Database performance can be defined as the optimization of resource usage to increase throughput and minimize contention.

Therefore, database performance can be defined as the optimization of resource usage to increase throughput and minimize contention, enabling the largest possible workload to be processed.

Whenever performance problems are encountered by an application that uses a database, the DBA is usually the first one called to resolve the problem. Of course, the DBA cannot manage database performance in a vacuum. Applications regularly communicate with other applications, systems, and components of the IT infrastructure. An effective performance monitoring and tuning strategy requires not just DBMS expertise but knowledge outside the scope of database administration. Many performance management tasks must be shared between the DBA and other technicians. In other words, handling performance problems is truly an enterprisewide endeavor.

Many tasks and abilities are required of DBAs to ensure efficient access to databases. Some of these abilities include building appropriate indexes, specifying large enough buffers and caches, aligning the database implementation with the IT infrastructure, monitoring databases and applications, reorganizing databases, and adapting to business changes—more users, more data, additional processing, and changing requirements and regulations.

Availability

Ensuring database availability is a multifaceted process.

The *availability* of data and databases is often closely aligned with performance, but it is actually a separate concern. Of course, if the DBMS is offline, performance will be nonexistent because no data can be accessed. However, ensuring database availability is a multifaceted process.

The first component of availability is keeping the DBMS up and running. Vigilant monitoring and automated alerts can be used to warn of DBMS outages and the need for corrective action.

Individual databases also must be maintained so that data is available whenever applications and clients require it. The DBA needs to design the database so that it can be maintained with minimal disruptions, but he also helps

ch01_1-48.qxd 5/22/02 1:24 PM Page 20

developers design applications to minimize conflicts when concurrent access is required.

An additional component of availability is minimizing the amount of downtime required to perform administrative tasks. The faster the DBA can perform administrative tasks that require databases to be offline, the more available the data becomes. Increasingly, the DBMS vendors and ISVs are providing nondisruptive utilities that can be performed on databases while applications read and write from the databases. Additionally, database clustering technologies provide failover techniques that help to reduce downtime. Nevertheless, such technology usually requires more skill and up-front planning to implement.

The DBA must understand all of these aspects of availability and ensure that each application is receiving the correct level of availability for its needs.

Database Security and Authorization

Once the database is designed and implemented, programmers and users will need to access and modify the data. However, to prevent security breaches and improper data modification, only authorized programmers and users should have access. It is the responsibility of the DBA to ensure that data is available only to authorized users.

Typically, the DBA works with the internal security features of the DBMS in the form of SQL GRANT and REVOKE statements, as well as with any group-authorization features of the DBMS. Security must be administered for many actions required by the database environment:

- Creating database objects, including databases, tables, views, and program structures
- Altering the structure of database objects
- Accessing the system catalog
- Reading and modifying data in tables
- Creating and accessing user-defined functions and data types
- Running stored procedures
- Starting and stopping databases and associated database objects
- Setting and modifying DBMS parameters and specifications
- Running database utilities such as LOAD, RECOVER, and REORG

ch01_1-48.qxd 5/22/02 1:24 PM Page 21

Database security can be enforced in other ways as well. For example, views can be created that block access to sensitive data by end users and programmers. In addition, the DBA interfaces frequently with external security methods when they impact database security. In short, the DBA must understand and be capable of implementing any aspect of security that impacts access to databases.

Backup and Recovery

The majority of recoveries today occur as a result of application software error and human error.

The DBA must be prepared to recover data in the event of a problem. "Problem" can mean anything from a system glitch or program error to a natural disaster that shuts down an organization. The majority of recoveries today occur as a result of application software error and human error. Hardware failures are not as prevalent as they used to be. In fact, analyst estimates indicate that 80% of application errors are due to software failures and human error. The DBA must be prepared to recover data to a usable point, no matter what the cause, and to do so as quickly as possible.

The first type of data recovery that usually comes to mind is a *recover to current*, usually in the face of a major shutdown. The end result of the recovery is that the database is brought back to its current state at the time of the failure. Applications are completely unavailable until the recovery is complete.

Another type of traditional recovery is a *point-in-time recovery*. Point-in-time recovery usually deals with an application-level problem. Conventional techniques to perform a point-in-time recovery remove the effects of *all* transactions since a specified point in time. This can cause problems if valid transactions occurred during that timeframe that still need to be applied.

Transaction recovery is a third type of recovery; it addresses the shortcomings of the traditional types of recovery: downtime and loss of good data. Thus, transaction recovery is an application recovery whereby the effects of specific transactions during a specified timeframe are removed from the database. Therefore, transaction recovery is sometimes referred to as application recovery.

To be prepared for any type of recovery, the DBA needs to develop a backup strategy to ensure that data is not lost in the event of an error in software, hardware, or a manual process. The strategy must be applicable to database processing, so it must include image copies of database files as well as a backup/recovery plan for database logs. It needs to account for any nondatabase file activity that can impact database applications, as well.

ch01_1-48.qxd 5/22/02 1:24 PM Page 22

Data Integrity

Three aspects of integrity are relevant to our discussion of databases: physical, semantic, and internal.

A database must be designed to store the correct data in the correct way without that data becoming damaged or corrupted. To ensure this process, the DBA implements integrity rules using features of the DBMS. Three aspects of integrity are relevant to our discussion of databases: physical, semantic, and internal.

Physical issues can be handled using DBMS features such as domains and data types. The DBA chooses the appropriate data type for each column of each table. This action ensures that only data of that type is stored in the database. That is, the DBMS enforces the integrity of the data with respect to its type. A column defined as "integer" can only contain integers. Attempts to store non-numeric or non-integer values in a column defined as integer will fail. DBAs can also utilize constraints to further delineate the type of data that can be stored in database columns. Most relational DBMS products provide the following types of constraints:

- *Referential constraints* are used to specify the columns that define any relationships between tables. Referential constraints are used to implement referential integrity, which ensures that all intended references from data in one column (or set of columns) of a table are valid with respect to data in another column of the same or a different table.
- *Unique constraints* ensure that the values for a column or a set of columns occur only once in a table.
- *Check constraints* are used to place more complex integrity rules on a column or set of columns in a table. Check constraints are typically defined using SQL and can be used to define the data values that are permissible for a column or set of columns.

Semantic integrity is more difficult to control and less easily defined. An example of semantic integrity is the quality of the data in the database. Simply storing any data that meets the physical integrity definitions specified to the database is not enough. Procedures and practices need to be in place to ensure data quality. For example, a customer database that contains a wrong address or phone number in 25% of the customer records is an example of a database with poor quality. There is no systematic, physical method of ensuring data accuracy. Data quality is encouraged through proper application code, sound business practices, and specific data policies. Redundancy is another semantic issue. If data elements are stored redundantly throughout the database, the

ch01_1-48.qxd 5/22/02 1:24 PM Page 23

DBA should document this fact and work to ensure that procedures are in place to keep redundant data synchronized and accurate.

The final aspect of integrity comprises internal DBMS issues. The DBMS relies on internal structures and code to maintain links, pointers, and identifiers. In most cases, the DBMS will do a good job of maintaining these structures, but the DBA needs to be aware of their existence and how to cope when the DBMS fails. Internal DBMS integrity is essential in the following areas:

- *Index consistency.* An index is really nothing but an ordered list of pointers to data in database tables. If for some reason the index gets out of sync with the data, indexed access can fail to return the proper data. The DBA has tools at his disposal to check for and remedy these types of errors.
- *Pointer consistency*. Sometimes large multimedia objects are not stored in the same physical files as other data. Therefore, the DBMS requires pointer structures to keep the multimedia data synchronized to the base table data. Once again, these pointers may get out of sync if proper administration procedures are not followed.
- *Backup consistency*. Some DBMS products occasionally take improper backup copies that effectively cannot be used for recovery. It is essential to identify these scenarios and take corrective actions.

Overall, ensuring integrity is an essential DBA skill.

DBMS Release Migration

The task of keeping the DBMS running and up-to-date is an ongoing effort that will consume many DBA cycles.

The DBA is also responsible for managing the migration from release to release of the DBMS. DBMS products change quite frequently—new versions are usually released every year or so. The task of keeping the DBMS running and up-to-date is an ongoing effort that will consume many DBA cycles. Whatever approach is taken must conform to the needs of the organization, while reducing outages and minimizing the need to change applications.

Jack-of-All-Trades

Databases are at the center of modern applications. If the DBMS fails, applications fail, and if applications fail, business can come to a halt. And if business comes to a halt often enough, the entire business can fail. Database administration is therefore critical to the ongoing success of modern business.

ch01_1-48.qxd 5/22/02 1:24 PM Page 24

Databases interact with almost every component of the IT infrastructure. The IT infrastructure of today comprises many tools:

- Programming languages and environments such as COBOL, Microsoft Visual Studio, C/C++, and Java
- Database and process design tools such as ERwin and Rational Rose
- Transaction processing systems such as CICS and Tuxedo
- Message queueing software such as MQSeries and MSMQ
- Networking software and protocols such as SNA, VTAM, TCP/IP, and Novell
- Networking hardware such as bridges, routers, hubs, and cabling
- Multiple operating systems such as Windows, OS/390 and MVS, UNIX, Linux, and perhaps others
- Data storage hardware and software such as enterprise storage servers, Microsoft SMS, IBM DFHSM, storage area networks (SANs), and NAS
- Operating system security packages such as RACF, ACF2, and Kerberos
- Other types of storage hardware such as tape machines, silos, and solid state (memory-based) storage
- Non-DBMS data set and file storage techniques such as VSAM and B-tree
- Database administration tools
- Systems management tools and frameworks such as BMC PATROL and CA Unicenter
- Operational control software such as batch scheduling software and job-entry subsystems
- Software distribution solutions for implementing new versions of system software across the network
- Internet and Web-enabled databases and applications
- Client/server development techniques such as multitier, fat server/thin client, thin server/fat client
- Object-oriented and component-based development technologies and techniques such as CORBA, COM, OLE DB, ADO, and EJB
- PDAs such as Palm Pilots and PocketPCs

ch01_1-48.qxd 5/22/02 1:24 PM Page 25

Although it is impossible to become an expert in all of these technologies, the DBA should have some knowledge of each of these areas and how they interrelate. Even more importantly, the DBA should have the phone numbers of experts to contact in case any of the associated software and hardware causes database access or performance problems.

Types of DBAs

There are DBAs who focus on logical design and DBAs who focus on physical design; DBAs who specialize in building systems and DBAs who specialize in maintaining and tuning systems; specialty DBAs and general-purpose DBAs. Truly, the job of DBA encompasses many roles.

Some organizations choose to split DBA responsibilities into separate jobs. Of course, this occurs most frequently in larger organizations, because smaller organizations often cannot afford the luxury of having multiple, specialty DBAs.

Still other companies simply hire DBAs to perform all of the tasks required to design, create, document, tune, and maintain the organization's data, databases, and database management systems. Let's look at some of the more common types of DBA.

System DBA

A system DBA focuses on technical rather than business issues.

A *system DBA* focuses on technical rather than business issues, primarily in the system administration area. Typical tasks center on the physical installation and performance of the DBMS software and can include the following:

- Installing new DBMS versions and applying maintenance fixes supplied by the DBMS vendor
- Setting and tuning system parameters
- Tuning the operating system, network, and transaction processors to work with the DBMS
- Ensuring appropriate storage for the DBMS
- Enabling the DBMS to work with storage devices and storage management software
- Interfacing with any other technologies required by database applications
- Installing third-party DBA tools

ch01_1-48.qxd 5/22/02 1:24 PM Page 26

System DBAs are rarely involved with actual implementation of databases and applications. They might get involved in application tuning when operating system parameters or complex DBMS parameters need to be altered.

Indeed, the job of system DBA usually exists only if the organization does not have an official system administration or systems programming department.

Database Architect

The database architect is involved in new design and development work only.

Some organizations create a separate position, *database architect*, for design and implementation of new databases. The database architect is involved in new design and development work only; he is not involved in maintenance, administration, or tuning of established databases and applications. The database architect designs new databases for new or existing applications.

The rationale for creating a separate position is that the skills required for designing new databases are different from the skills required to keep an existing database implementation up and running. A database architect is more likely than a general-purpose DBA to have data administration and modeling expertise.

Typical tasks performed by the database architect include:

- Creating a logical data model (if no DA or data modeler position exists)
- Translating logical data models into physical database designs
- Implementing efficient databases, including specifying physical characteristics, designing efficient indexes, and mapping database objects to physical storage devices
- Analyzing data access and modification requirements to ensure efficient SQL and optimal database design
- Creating backup and recovery strategies for new databases

Most organizations do not staff a separate database architect position, instead requiring DBAs to work on both new and established database projects.

Database Analyst

Another common staff position is the *database analyst*. There is really no set definition for this position. Sometimes junior DBAs are referred to as database analysts. Sometimes a database analyst performs a role similar to that of the

ch01_1-48.qxd 5/22/02 1:24 PM Page 27

database architect. Sometimes the data administrator is referred to as the database analyst or perhaps as the data analyst. And sometimes a database analyst is just another term used by some companies instead of database administrator.

Data Modeler

A *data modeler* is usually responsible for a subset of the DA's responsibilities. Data modeling tasks include the following:

- Collecting data requirements for development projects
- Analyzing the data requirements
- Designing project-based conceptual and logical data models
- Creating and updating a corporate data model
- Ensuring that the DBAs have a sound understanding of the data models

Application DBA

The application DBA focuses on database design and the ongoing support and administration of databases for a specific application or applications.

In direct contrast to the system DBA is the *application DBA*. The application DBA focuses on database design and the ongoing support and administration of databases for a specific application or applications. The application DBA is likely to be an expert at writing and debugging complex SQL and understands the best ways to incorporate database requests into application programs. The application DBA must also be capable of performing database change management, performance tuning, and most of the other roles of the DBA. The difference is the focus of the application DBA—it is on a specific subset of applications rather than the overall DBMS implementation and database environment. (See Figure 1-6.)

Not every organization staffs application DBAs. However, when application DBAs exist, general-purpose DBAs are still required to support the overall database environment and infrastructure. When application DBAs do not exist within an organization, general-purpose DBAs are likely to be assigned to support specific applications while also maintaining the organization's database environment.

There are pros and cons to staffing application DBAs. The arguments in favor of application DBAs include the following:

- An application DBA can better focus on an individual application, which can result in better service to the developers of that application.

ch01_1-48.qxd 5/22/02 1:24 PM Page 28

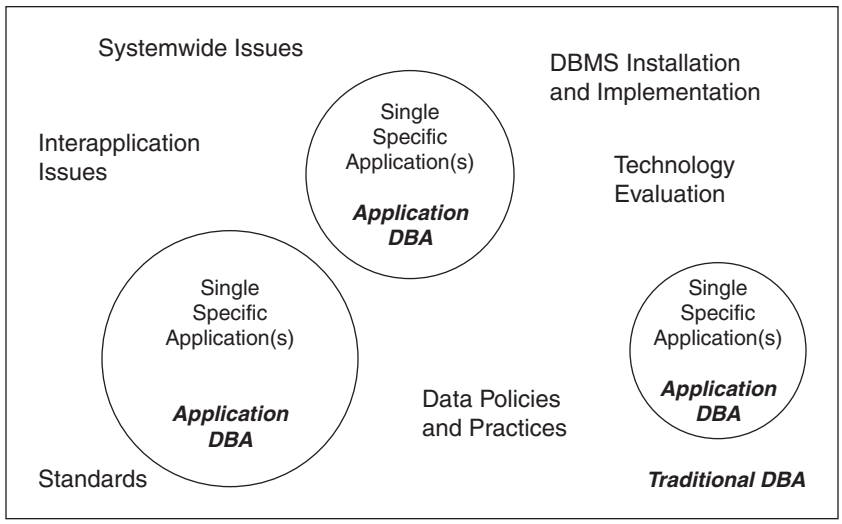

Figure 1-6 *Focus of the application DBA*

- The application DBA is more often viewed as an integral component of the development team and therefore is better informed about new development plans and changes.
- Because the application DBA works consistently on a specific set of applications, he can acquire a better overall understanding of how each application works, enabling him to better support the needs of the application developers.
- With a more comprehensive understanding of the application, an application DBA will have a better understanding of how the application impacts the overall business. This knowledge will likely result in the execution of DBA tasks to better support the organization.

But all is not favorable for application DBAs. There are cons to implementing an application DBA role:

- An application DBA can lose sight of the overall data needs of the organization because of his narrow focus on a single application.
- The application DBA can become isolated. Lack of communication with a centralized DBA group (if one exists) can result in diminished sharing of skills.

ch01_1-48.qxd 5/22/02 1:24 PM Page 29

- When an application DBA implements useful procedures, it takes more effort to share these procedures with the other DBAs.
- Due to the application-centric nature of the position, an application DBA can lose sight of new features and functionality delivered by the DBMS group.

When staffing application DBAs, be sure to also staff a centralized DBA group.

In general, when staffing application DBAs, be sure to also staff a centralized DBA group. The application DBAs should have primary responsibility for specific applications, but should also be viewed as part of the centralized DBA group.

Task-Oriented DBA

Larger organizations sometimes create very specialized DBAs that focus on a specific DBA task. However, task-oriented DBAs are quite rare outside of very large IT shops. One example of a task-oriented DBA is a backup-and-recovery DBA who devotes his entire day to ensuring the recoverability of the organization's databases.

Most organizations cannot afford this level of specialization, but when possible, task-oriented DBAs can ensure that very knowledgeable specialists tackle very important DBA tasks.

Performance Analyst

Performance analysts are a specific type of task-oriented DBA. The performance analyst, more common than other task-oriented DBAs, focuses solely on the performance of database applications.

A performance analyst must understand the details and nuances of SQL coding for performance and be able to design databases for performance. A performance analyst will have very detailed technical knowledge of the DBMS so that he can make appropriate changes to DBMS and system parameters when required.

However, the performance analyst should not be a system DBA. The performance analyst must be able to speak to application developers in their language in order to help them facilitate appropriate program changes for performance.

The performance analyst is usually the most skilled, senior member of the DBA staff.

The performance analyst is usually the most skilled, senior member of the DBA staff, a role that he has grown into due to his experience and the respect he has gained in past tuning endeavors.

ch01_1-48.qxd 5/22/02 1:24 PM Page 30

Data Warehouse Administrator

Organizations that implement data warehouses for performing in-depth data analysis often staff DBAs specifically to monitor and support the data warehouse environment. *Data warehouse administrators* must be capable DBAs, but with a thorough understanding of the differences between a database that supports OLTP and a data warehouse. Data warehouse administration requires experience with the following:

- Business intelligence, query, and reporting tools
- Database design for read-only access
- Data warehousing design issues such as star schema
- Data warehousing technologies such as OLAP (including ROLAP, MOLAP, and HOLAP)
- Data transformation and conversion
- Data quality issues
- Data formats for loading and unloading of data
- Middleware

Staffing Considerations

Staffing the DBA organization is not a simple matter. Several nontrivial considerations must be addressed, including the size of the DBA staff and the reporting structure for the DBAs.

How Many DBAs?

One of the most difficult things to determine is the optimal number of DBAs required to keep an organization's databases online and operating efficiently. Many organizations try to operate with the minimal number of DBAs on staff; the idea being that fewer staff members lowers cost. However, that assumption may not be true. An overworked DBA staff can make mistakes that cause downtime and operational problems far in excess of the salary requirements of an additional DBA.

Determining how many DBAs is optimal is not a precise science. It depends on many factors:

ch01_1-48.qxd 5/22/02 1:24 PM Page 31

- *Number of databases*. The more databases that need to be supported, the more complex the job of database administration becomes. Each database needs to be designed, implemented, monitored for availability and performance, backed up, and administered. There is a limit to the number of databases that an individual DBA can control.
- *Size of the databases***.** The larger the databases that need to be supported, the more difficult the job of database administration. A larger database takes longer to create, maintain, and tune. In addition, more potential for confusion arises when SQL takes longer to execute—causing the DBA to spend more time working with developers to tune SQL.
- *Number of users.* As additional users are brought online, optimal database performance becomes more difficult to ensure. Additionally, as the number of users increases, the potential for increase in the volume of problems and calls increases, further complicating the DBA's job.
- *Number of applications.* A single database can be utilized by numerous applications. Indeed, one of the primary benefits of the DBMS is that it enables the sharing of data across an organization. As more applications are brought online, additional pressure is exerted on the database in terms of performance, availability, and resources. As more applications are brought online, more DBAs may be required to support the same number of databases.
- *Service-level agreements (SLAs)*. The more restrictive the SLA, the more difficult it becomes for the DBA to deliver the service. For example, a service-level agreement requiring subsecond response time for transactions is more difficult to support than an agreement requiring three-second response time.
- *Availability requirements*. Database administration becomes easier if databases have an allowable period of scheduled downtime. Some DBA tasks either require an outage, or are easier when an outage can be taken. Considerations such as supporting e-business transactions and the Web drive the need for 24/7 database availability. 24/7 availability is often incompatible with certain DBA tasks.
- *Impact of downtime*. The greater the financial impact of an unavailable database, the greater the pressure on the DBA to assure greater database availability.

ch01_1-48.qxd 5/22/02 1:24 PM Page 32

- *Performance requirements.* As the requirements for database access become more performance oriented, database administration becomes more complicated.
- *Type of Applications*. The type of applications supported has a direct bearing on the number of DBAs required. The DBMS and database needs of a mission-critical application differ from those of a non-mission-critical application. Mission-critical applications are more likely to require constant monitoring to ensure availability. Likewise, an OLTP application has different characteristics and administration requirements than an OLAP application. OLTP transactions are likely to be of shorter duration than OLAP queries; OLTP applications perform both read and write operations whereas OLAP applications are predominantly read-only. Each has administration challenges that require different DBA procedures.
- *Volatility.* The frequency of database change requests is an important factor in the need for additional DBAs. A static database environment requiring few changes will not require the same level of DBA effort as a volatile, frequently changing database environment. Unfortunately, the level of volatility for most databases and applications tends to change dramatically over time. It's usually very difficult to ascertain how volatile an overall database environment will be over its lifetime.
- *DBA staff experience*. The skill of the existing DBA staff affects the need for additional DBAs. A highly skilled DBA staff will accomplish more than a novice team. Skills, more than experience, dictate DBA staffing requirements. A highly skilled DBA with two years of experience might easily outperform a ten-year veteran who is burned out and unmotivated.
- *Programming staff experience***.** If the application developers are not highly skilled in database and SQL programming, the DBAs will need to be more involved in the development process. DBAs will be needed for tasks such as composing complex SQL, analyzing SQL and application code, debugging, tuning, and ensuring connectivity. As the experience of the programming staff increases, the complexity of DBA requirements decreases.
- *End user experience.* When end users access databases directly with ad hoc SQL, their skill level has a direct impact on the complexity of DBA. If the end user has few SQL skills, the DBA will need to be initiate more performance monitoring and tuning.

ch01_1-48.qxd 5/22/02 1:24 PM Page 33

- *Variety of DBMSs.* The more heterogeneous the environment, the more difficult it becomes to administer. For example, acquiring and maintaining expertise in both Oracle and DB2 is more difficult than gaining expertise in only one of them. Moreover, as multiple DBMSs of different types are installed, database administration becomes even more difficult. For example, a shop with DB2, IMS, and IDMS will have to possess relational (DB2), hierarchical (IMS), and network/CODASYL (IDMS) expertise.
- *DBA tools*. DBMS vendors and a number of ISVs offer tools that automate DBA tasks and make database administration easier. DBA tasks become less complex with the more tools available and the degree to which they are integrated. Lou Agosta, an industry analyst with Giga Group, states that "without [DBA] tools up to twice the number of DBAs might [be] required."

Creating a formula that will dictate the optimal number of DBAs to employ is difficult.

This list of issues notwithstanding, creating a formula that will dictate the optimal number of DBAs to employ is difficult. Industry analysts at the META Group have established a loose formula for calculating DBA level of effort. The formula arrives at a level of effort by applying weights to six factors: system complexity, application immaturity, end-user sophistication, software functionality, system availability, and staff sophistication. After measuring each of these items, you plug in values to the formula to arrive at an estimate for the number of DBAs required. If you are interested in pursuing this metric further, I refer you to the META Group research (META Group, Open Computing & Server Strategies, File: 656, Date: 20-Mar-1998). META Group can be contacted at http://www.metagroup.com or by phone at 1-203-973-6700.

DBA Reporting Structures

To whom should the DBA group report? Different companies have taken different approaches to the DBA reporting structure, but a few reporting hierarchies are quite common. Some reporting structures work better than others, so let's review some of the possibilities.

One of the best structures is a data resource management group that consists of all the data and information specialist of the organization.

One of the best structures is a data resource management (DRM) group that consists of all the data and information specialist of the organization—DA, DBA, data analysts, performance analysts, and so on. This group usually reports directly to the CIO, but might report through a systems programming unit, the data center, or technical support. Figure 1-7 depicts a typical reporting structure.

ch01_1-48.qxd 5/22/02 1:24 PM Page 34

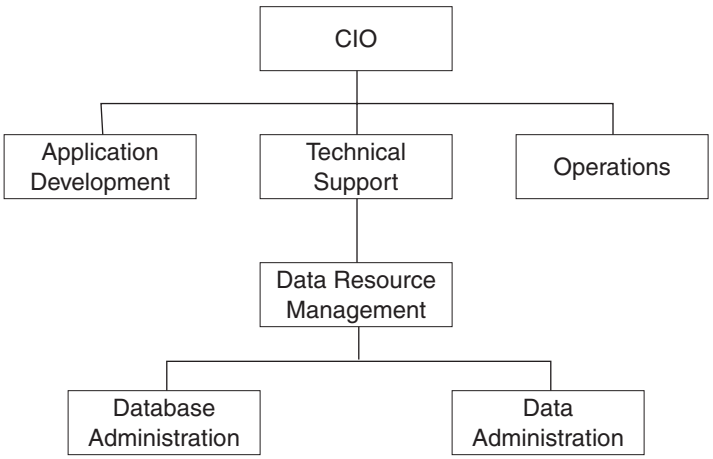

Figure 1-7 *Typical DBA reporting structure*

When an organization staffs application DBAs, they will be spread out in application groups, typically with a direct line of report to the business programming managers. Each application development team has a dedicated application DBA resource as shown in Figure 1-8.

There are problems with both of these reporting structures, though. First, the DRM needs to be placed higher in the IT reporting hierarchy. It's a good idea to have the DRM group report directly to the CIO. When an organization understands the importance of data to the health of the organization, placing the DRM group at this level is encouraged.

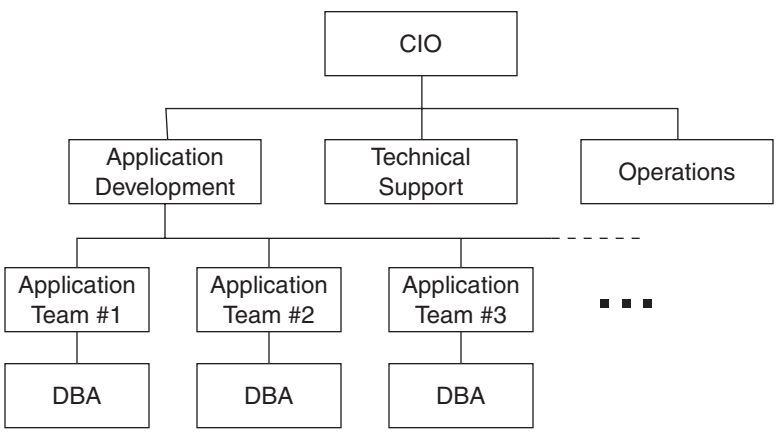

Figure 1-8 *Application DBA reporting structure*

ch01_1-48.qxd 5/22/02 1:24 PM Page 35

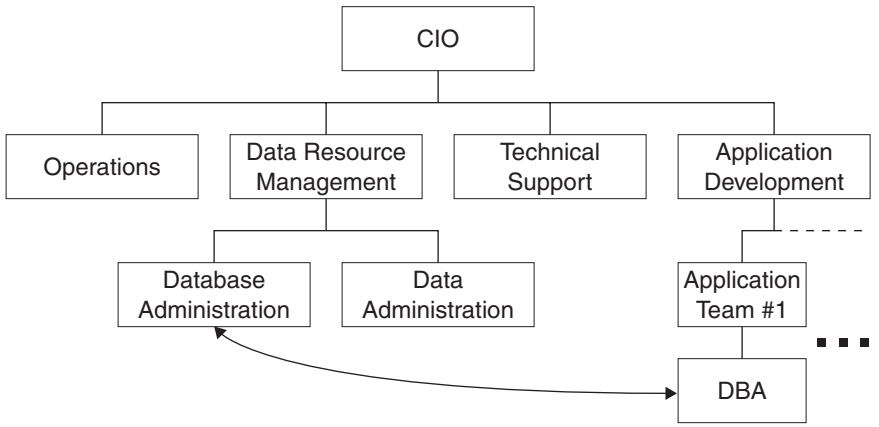

Figure 1-9 *Recommended DRM reporting structure*

Furthermore, when application DBAs exist, they should not report to the application programming manager only. A secondary line of report to the DRM group will ensure that DBA skills are shared and communicated throughout the organization. Figure 1-9 delineates the recommended reporting structure for the DRM group.

Multiplatform DBA Issues

Managing a multiplatform environment complicates the job of database administration.

Managing a multiplatform environment complicates the job of database administration. A whole batch of different problems and issues arise that need to be addressed. The first task is to define the scope of each DBA's job. Does a single DBA administer all of the different DBMSs or does each DBA focus on supporting only one DBMS?

This is a particularly thorny issue. On the one hand, the functionality of a DBMS is strikingly similar regardless of platform and vendor. A DBMS is designed to store, retrieve, and protect data. Programmers, programs, and end users all interact with the DBMS to access and modify data. Administration issues are similar—design, creation, optimization, and so on—though each DBMS implements these items differently. So, the case can be made that a DBA should support multiple DBMSs and databases, regardless of platform or vendor.

On the other hand, each DBMS offers different features, functionality, and technology. Keeping all of the differences and nuances straight is a monumental task. Wouldn't it be better to develop platform-expert DBAs? That way, your Oracle DBAs can focus on learning all there is to know about Oracle, your DB2 DBAs can focus on DB2, and so on.

ch01_1-48.qxd 5/22/02 1:24 PM Page 36

Every organization will have to make this determination based on their particular mix of DBMSs, features, and DBA talent. If your organization uses one DBMS predominantly, with limited use of others, it may make sense for each DBA to support all of them, regardless of platform or vendor. Sparse usage of a DBMS usually means fewer problems and potentially less usage of its more sophisticated features. By tasking your DBAs to be multi-DBMS and multiplatform, you can ensure that the most skilled DBAs in your shop are available for all database administration issues. If your organization uses many different DBMSs, it is probably wise to create specialist DBAs for the heavily used platforms and perhaps share administration duties for the less frequently used platforms among other DBAs.

When DBA duties are shared, be sure to carefully document the skills and knowledge level of each DBA for each DBMS being supported. Take care to set up an effective and fair on-call rotation that does not unduly burden any particular DBA or group of DBAs. Furthermore, use the organizational structure to promote sharing of database standards and procedures across all supported DBMS environments.

Keep in mind, too, that when multiple DBMSs and platforms are supported, you should consider implementing DBA tools, performance monitors, and scripts that can address multiple platforms. For this reason, DBA tools from third-party vendors are usually better for heterogeneous environments than similar tools offered by the DBMS vendors.

When your organization supports multiple DBMSs, the DBA group should develop guidelines for which DBMS should be used in which situations. These guidelines should not be hard-and-fast rules, but instead should provide guidance for the types of applications and databases best supported by each DBMS. Forcing applications to a given DBMS environment is not a good practice. The guidelines should be used simply to assure best fit of application to DBMS. These guidelines should take into account:

- Features of each DBMS
- Features and characteristics of the operating system
- Networking capabilities of the DBMS and operating system combination
- DBMS skills of the application developers
- Programming language support
- Any other organizational issues and requirements

ch01_1-48.qxd 5/22/02 1:24 PM Page 37

Test and Production

Separating the test and production environments ensures the integrity and performance of operational work.

At least two separate environments must be created and supported for a quality database implementation: *test* and *production*. Completely separating the test and production environments ensures the integrity and performance of operational work. New development and maintenance work can be performed in the test environment while operational applications are run in the production environment. Failure to separate test and production will cause development activities to impair the day-to-day business of your organization. Errant program code in the early stages of development could access or modify production data and cause production performance problems or invalid data.

The test and production environments need not be identical. While the production environment contains all of the data required to support the operational applications, the test environment needs only a subset of data required for acceptable application testing. Furthermore, the test DBMS implementation will usually not command the same amount of resources as the production environment. For example, less memory will be allocated to buffering and caches, data set allocations will be smaller and on fewer devices, and the DBMS software may be a more recent version in test than in production (to shake out any bugs in the DBMS code itself before it is trusted to run in production).

The test and production environments should be structured similarly though. Both environments should have access to the same system software because the programming staff needs to create applications in the same type of environment in which they will eventually run.

DBAs may need to create multiple copies of databases in the test environment to support concurrent development by multiple programmers. Furthermore, the programming staff must be able to control the contents of the test databases. Because programmers may need to run data modification programs multiple times during the development process, they must be able to ensure that the data at the beginning of each test run is the same. Failure to do so can render the results of the tests invalid. Therefore, the DBA must assist the programming staff in the creation of database load and unload jobs to set up test databases. Prior to a test run, the database must be loaded with the test data. After the test run, the programmer can examine the output from the program and the contents of the database to determine if the program logic is correct. If not, he can repeat the process, loading to reset the data in the database and retesting. Automated procedures can be put in place to unload the databases impacted by the program and compare the results to the load files.

ch01_1-48.qxd 5/22/02 1:24 PM Page 38

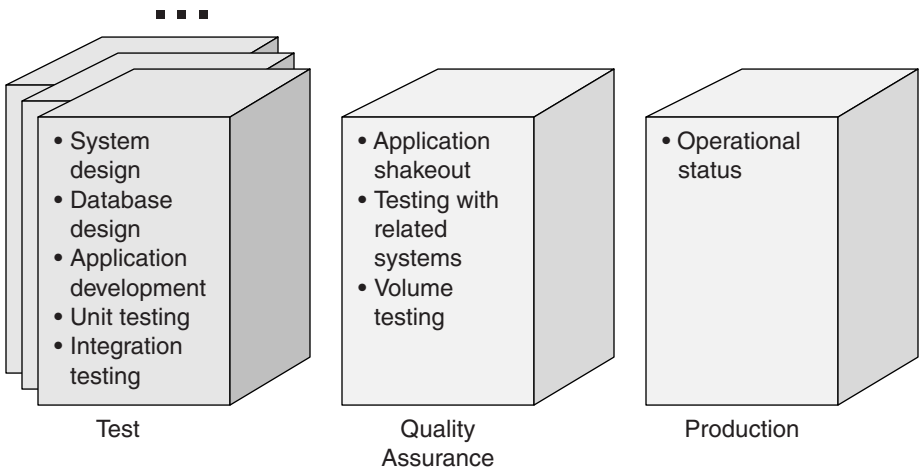

Figure 1-10 *Establishing multiple database environments*

Predicting how test applications will perform once they are moved to production is difficult, but the DBA can assist here as well. A relational DBMS typically provides a method to gather statistical information about the contents of its databases. These statistics are then used by the relational optimizer to determine how SQL will retrieve data. (This topic is covered in more depth in Chapter 12.) But remember, there will be much less data in test databases than in production. In some cases, though, the DBA can set up scripts to read the production statistics and copy them to the test environment, thereby enabling developers to gauge more accurately how test applications will perform in production.

A QA environment may be needed to perform rigorous testing against new and modified programs before they are migrated.

Some organizations implement more than two environments, as shown in Figure 1-10. If special care is needed for complex application development projects, additional levels of isolated testing may need to occur. For example, a unit test environment may exist for individual program development, while an integration testing environment ensures that new programs work together or with existing programs. A quality assurance environment may be needed to perform rigorous testing against new and modified programs before they are migrated to the production environment.

New Technology and the DBA

The DBA is at the center of the action whenever new ways of doing business and new technologies are introduced to the organization. Data is the lifeblood

ch01_1-48.qxd 5/22/02 1:24 PM Page 39

of modern business, data is housed by the database, and the DBA is the expert who understands database technology—and in particular, how databases can be integrated with other new technologies.

Let's examine three specific newer technologies that rely on database administration—at least somewhat—to be effectively implemented: database-coupled application logic, Internet-enabled e-business development, and handheld computing.

Procedural DBAs: Managing Database Logic

Triggers, user-defined functions, and stored procedures provide the ability to define business rules to the DBMS.

Until recently, the purpose of a database management system was, appropriately enough, to store, manage, and access data. Although these core capabilities are still required of modern DBMS products, additional procedural functionality is slowly becoming not just a nice feature to have, but a necessity. Features such as triggers, user-defined functions, and stored procedures provide the ability to define business rules to the DBMS instead of in separate application programs. These features couple application logic tightly to the database server.

Since all of the most popular RDBMS products provide sometimes-complex features to facilitate database-coupled procedural logic, additional management discipline is required to ensure the optimal use of these features. Typically, as new features are added, their administration, design, and management are assigned to the DBA by default. However, without proper planning and preparation, chaos can ensue. First let's examine how database logic is stored in a DBMS.

Stored Procedures

Stored procedures can be thought of as programs that live in a database. The procedural logic of a stored procedure is maintained, administered, and executed through the database commands. The primary reason for using stored procedures is to move application code from a client workstation to the database server. Stored procedures typically consume less overhead in a client/server environment because one client can invoke a stored procedure that causes multiple SQL statements to be run. The alternative, the client executing multiple SQL statements directly, increases network traffic and can degrade overall application performance.

A stored procedure is a freestanding database object; it is not "physically" associated with any other object in the database. A stored procedure can access and/or modify data in many tables.

ch01_1-48.qxd 5/22/02 1:24 PM Page 40

Triggers

Triggers are event-driven specialized procedures that are attached to database tables. The trigger code is automatically executed by the RDBMS as data changes in the database. Each trigger is attached to a single, specified table. Triggers can be thought of as an advanced form of rule or constraint that uses procedural logic. A trigger cannot be directly called or executed; it is automatically executed (or "fired") by the RDBMS as the result of a SQL INSERT, UPDATE, or DELETE statement issued on its associated table. Once a trigger is created, it is always executed when its firing event occurs.

User-Defined Functions

A *user-defined function* (UDF) provides a result based on a set of input values. UDFs are programs that can be executed in place of standard, built-in SQL scalar or column functions. A scalar function transforms data for each row of a result set; a column function evaluates each value for a particular column in each row of the results set and returns a single value. Once written, and defined to the RDBMS, a UDF becomes available just like any other built-in database function.

Table 1-1 summarizes the differences between stored procedures, triggers, and UDFs.

Administering Stored Procedures, Triggers, and UDFs

Once developers begin to rely on stored procedures, triggers, and UDFs, DBAs need to take steps to manage them properly. DBAs must grapple with the issues of quality, maintainability, efficiency, and availability. How and when will these procedural objects be tested? The impact of a failure is enterprisewide,

Table 1-1 *Procedural Database Objects*

Object Type	Definition	Executed	How
Stored Procedure	Program logic executed on the database server	By request	Explicit
Triggers	Event-driven procedures attached to database tables	Automatically	Implicit
UDFs	Program logic extending SQL functionality	By request in SQL	Explicit

ch01_1-48.qxd 5/22/02 1:24 PM Page 41

increasing the visibility and criticality of these objects. Who is responsible if they fail? The answer must be—the DBA.

The role of administering procedural database logic should fall upon someone skilled in that discipline. A new type of DBA is required to accommodate the administration of database procedural logic. This new role can be defined as a *procedural DBA*.

The procedural DBA is responsible for those database management activities that require procedural logic support.

The procedural DBA is responsible for those database management activities that require procedural logic support. He ensures that stored procedures, triggers, and user-defined functions are effectively planned, implemented, shared, and reused. The procedural DBA also takes primary responsibility for coding and testing all triggers. Stored procedures and user-defined functions, although likely to be coded by application programmers, should be reviewed for accuracy and performance by procedural DBAs. (See Figure 1-11.)

The procedural DBA leads the review and administration of all procedural database objects: triggers, stored procedures, and UDFs. Although the procedural DBA is unlikely to be as skilled at programming as an applications programmer

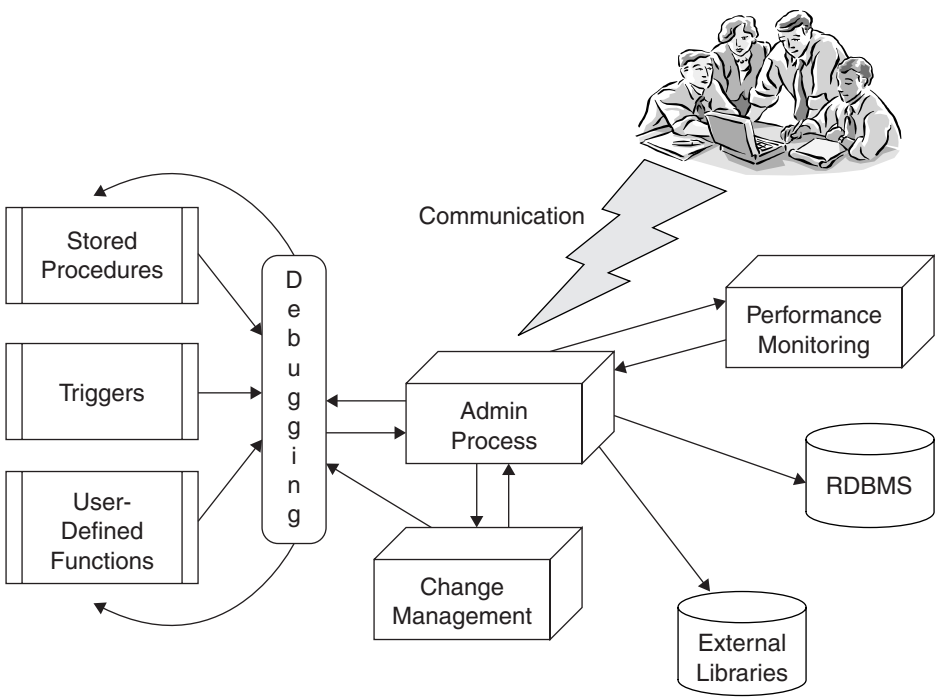

Figure 1-11 *Procedural DBA duties*

ch01_1-48.qxd 5/22/02 1:24 PM Page 42

The procedural DBA leads the review and administration of all procedural database objects: triggers, stored procedures, and UDFs.

or systems analyst, he must be able to write and review program code reasonably well. The skill level required depends on what languages are supported by the DBMS for creating procedural objects, the rate and level of adoption within the organization, and whether an internal organization exists for creating common, reusable programs. Table 1-2 provides a reasonable level of procedural DBA involvement for each type of procedural object. Additionally, the procedural DBA should be on call for any problems that occur with database procedural objects in production.

As shown in Figure 1-11, the procedural DBA requires communication skills as much as he requires technological acumen. In addition to managing and optimizing database procedural objects, the procedural DBA must inform the development community of new triggers, stored procedures, and UDFs. Furthermore, the DBA must promote reuse. If the programmers do not know that these objects exist, they will never be used.

Other procedural administrative functions can be allocated to the procedural DBA. Depending on the number of DBAs and the amount of application development needed, the procedural DBA can be assigned to additional functions such as the following:

- Participating in application code design reviews
- Reviewing and analyzing SQL access paths (from "EXPLAIN" or "SHOW PLAN")
- Debugging SQL
- Writing and analyzing complex SQL statements
- Rewriting queries for optimal execution

Table 1-2 *Procedural DBA Involvement by Object*

Object Type	Level of Procedural DBA Involvement
Stored Procedure	Not likely to write stored procedures; must review all code before migration to production; communicates availability and promotes reuse.
Triggers	Likely to write, test, and debug triggers; communicates deployment of triggers to ensure application awareness.
UDFs	Not likely to write user-defined functions; works closely with the development team to ensure UDF functionality and performance; reviews all code before migration to production; communicates availability and promotes reuse.

ch01_1-48.qxd 5/22/02 1:24 PM Page 43

Offloading coding-related tasks to the procedural DBA can help the other staff DBAs concentrate on the actual physical design and implementation of databases, resulting in much better designed databases. Procedural DBAs should have the same line of report as traditional DBAs to enable better sharing of skills between the groups. Of course, there will need to be a greater synergy between procedural DBAs and the application programmers. The procedural DBA should typically come from the application programming ranks because this is where the coding skill exists.

The Internet: From DBA to e-DBA

Companies of every size are using Internet technologies to speed up business processes. Indeed, *e-business* has evolved as a new term to describe the transformation of key business processes using Internet technologies. Modern organizations use the Web to communicate with their partners and customers, to connect with their back-end databases, and to conduct transactions (e-commerce). E-business is the integration of traditional information technology with the Internet. This integration creates a more nimble business, prepared for the trials and tribulations of conducting business in the 21st century.

E-businesses must be able to adapt and react to constant change.

E-businesses must be able to adapt and react to constant change. When a business is online, it never closes. People expect full functionality on Web sites they visit regardless of the time. And the Web *is* worldwide. It may be two o'clock in the morning in New York City, but it is always prime time somewhere in the world. An e-business must be available and prepared to engage with customers 24 hours a day, 365 days a year (366 during leap years). Failure to do so risks losing business. When a Web site is down, the customer will go elsewhere to do business because the competition is just a simple mouse-click away. Therefore, those who manage an e-business must be adept, proactive, and ever vigilant.

The frantic pace of an e-business makes extreme demands on those that keep it operational, and DBAs are much affected. The need to integrate the Web with traditional IT services, such as the DBMS, places high expectations on database administrators.

An e-DBA is a DBA who is capable of managing Web-based applications because he understands the special issues that arise because of the Internet.

An e-DBA is a DBA who is capable of managing Web-based applications and their Internet-related issues. With all of the knowledge and training of a traditional DBA, the e-DBA adapts these skills to suit applications and databases that are Internet enabled. When the Web is coupled with traditional applications and databases, a complex infrastructure is the result. (See Figure 1-12.) The e-DBA must be capable of navigating this complex, heterogeneous infrastructure and providing expertise wherever databases interact within this infrastructure.

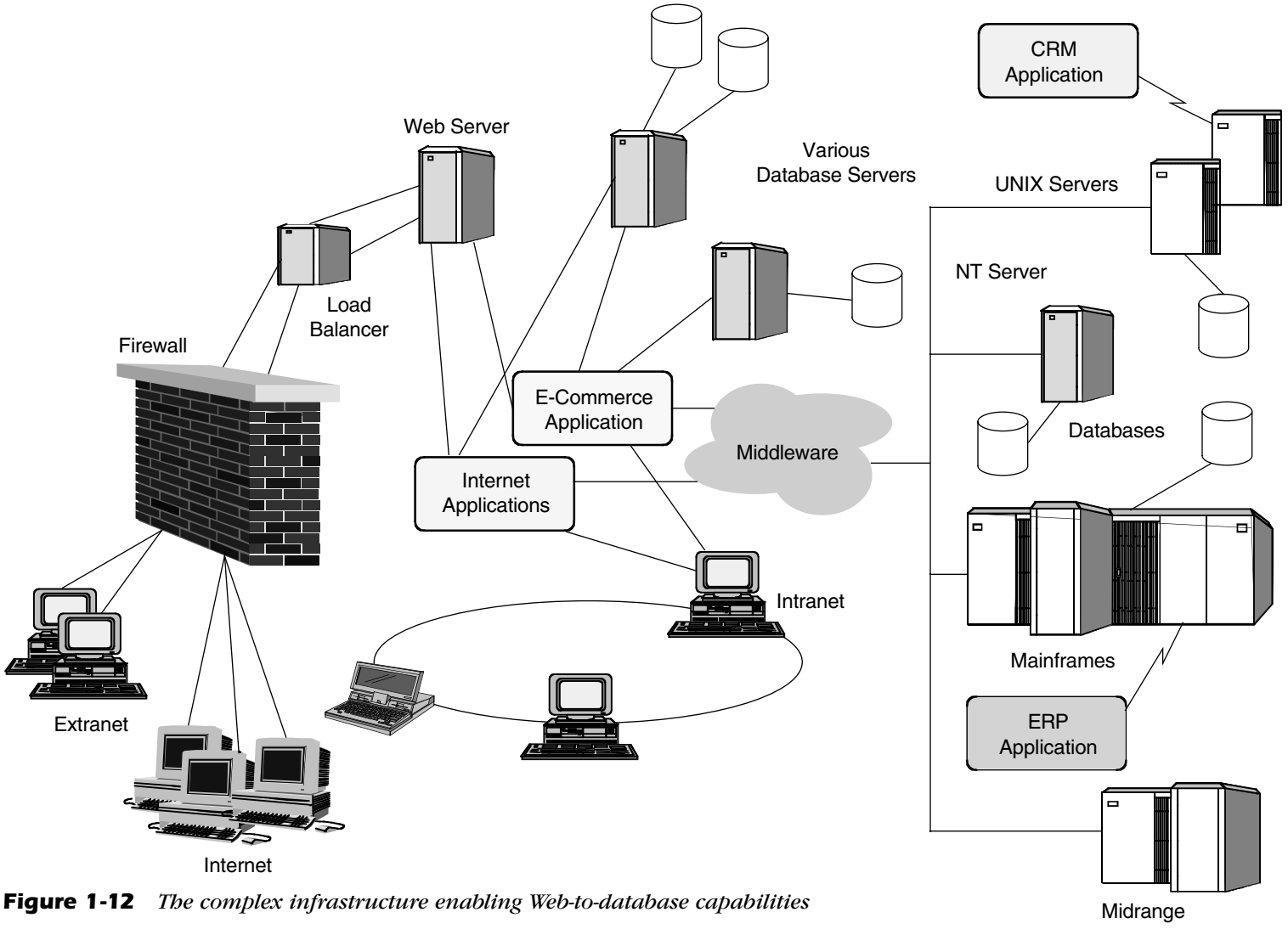

Figure 1-12 *The complex infrastructure enabling Web-to-database capabilities*

ch01_1-48.qxd 5/22/02 1:24 PM Page 44

ch01_1-48.qxd 5/22/02 1:24 PM Page 45

Many factors impact database administration when you couple the Internet with database technology. Some of these issues include

- 24/7 data availability
- New technologies such as Java and XML
- Web connectivity
- Integration of legacy data with Web-based applications
- Database and application architecture
- Web-based administration
- Performance engineering for the Internet
- Unpredictable workload

The PDA DBA

Personal digital assistant devices, better known as PDAs, are fast becoming a necessity for modern executives and businessmen. A PDA is a handheld computing device. Whether your PDA of choice is a Palm Pilot or a PocketPC, your PDA may soon have a DBMS running on it. Why is that interesting? Does it change the way you will use your PDA? What will that mean to your IT department?

PDAs offer many benefits. The devices are small and therefore easily transportable. They enhance a mobile worker's ability to be mobile. However, challenges must be faced as organizations incorporate PDAs into their infrastructure. Companies with remote workers such as a distributed sales force or delivery tracking services will most likely be the first impacted. The data on the PDAs must be managed professionally to ensure integrity and reliability. Because the device is remote, sharing of data can be difficult. The data on the PDAs must be reliably synchronized with existing enterprise systems and databases.

All major DBMS vendors provide small-footprint versions of their flagship products to run on PDAs.

All major DBMS vendors provide small-footprint versions of their flagship products to run on PDAs. For example, IBM markets DB2 Everyplace, Oracle sells Oracle8i Lite, and Sybase offers Adaptive Server Anywhere. The general idea is to store a small amount of critical data on the PDA in a database; the local PDA database is later synchronized to long-term data stores on enterprise database servers. Each PDA DBMS provides technology to synchronize data back and forth from the PDA to the enterprise server platforms.

A database the size of those stored on PDAs should not require the in-depth tuning and administration required of enterprise database implementations.

ch01_1-48.qxd 5/22/02 1:24 PM Page 46

However, DBAs will be called upon to help design appropriately implemented databases for small-form-factor devices like PDAs. However, the biggest impact on the DBA will be the necessity for managing data synchronization from hundreds or thousands of PDAs. When should synchronization be scheduled? How will it impact applications that use large production databases that are involved in the synchronization? How can you ensure that a mobile user will synchronize his data reliably and on schedule?

These are not trivial issues. The DBA staff must be ready to support the organization's inevitable request for this technology by understanding data synchronization technology and the need for remote database users at their organization. Pervasive computing and the mobile workplace are here to stay. The DBA staff must be ready to support these mobile workers with a valid, shared data infrastructure.

As new technology is introduced to the organization, the DBA group is typically the first to examine and use it. The preceding three technologies are merely examples of new trends and technologies that require database administration for efficient and effective implementation.

DBA Certification

Professional certification is a recent trend in IT and is available for many different IT jobs. The availability and levels of certification for database administration have been growing at an alarming rate. Certification programs are available for most of the popular DBMS platforms including IBM DB2, Microsoft SQL Server, and Oracle. The idea behind DBA certification is to ensure that an individual is capable of performing database administration tasks and duties.

This is a noble goal, but the problem is that passing a test is not a viable indicator of success with the complexities of database administration. Some things you just have to learn by doing. I'm not saying that certification is useless: Taking the test and focusing on the questions you miss can help to point out areas of weakness. But does anyone *really* believe that someone passing a formalized test is necessarily as capable as someone with several years of experience as a DBA? Organizations should hire DBAs based on experience that indicates a level of capability. Of course, a DBA with both experience and certification is even better.

Certification will make you more employable.

That said, I do recommend that professional DBAs take the time to study and pass the certification exams. Not because certification will make you a better DBA, but because it will make you more employable. Some companies hire

ch01_1-48.qxd 5/22/02 1:24 PM Page 47

only certified professionals. The trend toward using certification to guide hiring practices will increase because of increasing IT complexity. If you think you might change jobs at some point in your career (and who among us will not?), then certification is a worthwhile pursuit.

Keep in mind that DBA certification tests sometimes ask arcane syntax questions that are not good indicators of a DBA's skills. Getting the syntax 100% accurate is what manuals and design tools are for. Memorizing every detail about SQL syntax and structure is a waste of time because it is complex and changes all the time. It is better to know where to find the syntax, parameters, and answers to your questions when you need them—that is, which manuals and textbooks contain the needed information. DBAs should possess a broad, overarching knowledge of DBMS concepts, IT fundamentals, and the working of their organization's database systems. In other words, it is better to know off the top of your head that something can (or cannot) be done than to know the exact syntax for how to accomplish it.

If you decide to pursue certification, take the time to prepare for the tests. Books and self-learning software titles are available that can be quite useful. These books and programs cover the most common test topics and provide sample questions to help you prepare. In many ways, it is like preparing for a college entrance exam like the SAT.

Finally, once you earn your certification, make sure you display it proudly on your resume and your business card (if your company allows it).

Table 1-3 lists Web sites that contain information about professional certification for the most popular DBMS products.

Table 1-3 *Sources of DBA Certification Information*

DBMS	Web site
Oracle	http://www.oracle.com/education/certification/
Microsoft SQL Server	http://www.microsoft.com/trainingandservices/default.asp?PageID=training
IBM DB2	http://www.ibm.com/certify
Sybase Adaptive Server	http://www.sybase.com/education/profcert/
Informix	http://www.informix.com/informix/training/courses/certific/welcome.htm

ch01_1-48.qxd 5/22/02 1:24 PM Page 48

The Rest of the Book

This first chapter has introduced you to the world of the DBA. I hope that you have gained respect for the complexity of the position and the qualities required of a good DBA. The remainder of the book will examine the details of the tasks, roles, and responsibilities required of the DBA.

Review

1. What are the primary high-level job responsibilities of a DBA?
2. What is the single biggest problem faced by organizations using relational databases?
3. What is the difference between a data administrator and a database administrator?
4. What factors determine the number of DBAs needed to support an organization's database environment properly?
5. How does new technology impact the job of the DBA?
6. What are the technologies that mandate the need for procedural DBAs?
7. What is the difference between a database architect and a system administrator?
8. Which staff member is most likely to be responsible for installing a new DBMS release?
9. What are the three types of integrity that DBAs must understand?
10. Is a certified DBA necessarily a qualified DBA? Why or why not?

Bonus Question

Why must the DBA be a jack-of-all-trades?