

**Teach Yourself Oracle 8 In 21 Days **

**Table of Contents:**

- Introduction -

Week 1 at a Glance

Day 1 - Starting Out with Oracle

Day 2 - Exploring the Oracle Architecture

Day 3 - Installing Oracle Software

Day 4 - Properly Sizing Your Database and Planning for Growth

Day 5 - Managing the Oracle RDBMS

Day 6 - Administering Databases and Datafiles

Day 7 - Administering Tablespaces

Week 1 in Review

Week 2 at a Glance

Day 8 - Administering Redo Logs, Control Files, and Rollback Segments Day 9 - Managing Data

Day 10 - Administering User Accounts

Day 11 - Managing Processes

Day 12 - Working with Tables, Views, and Synonyms

Day 13 - Using Indexes and Sequences

Day 14 - Using Oracle Clusters, Stored Procedures, and Database Links

Week 2 in Review

Week 3 at a Glance

Day 15 - Managing Job Queues and Using Oracle Auditing

Day 16 - Understanding Effective Backup Techniques

Day 17 - Recovering the Database

Day 18 - Administering Oracle Replication

Day 19 - Advanced Oracle Options

Day 20 - Effectively Tuning and Optimizing the Database

Day 21 - Exploring the Web Publishing Assistant and Network Computing Architecture Week 3 in Review

**- Appendixes -**

Appendix A - Answers

Appendix B - Oracle Tuning Parameters

Appendix C - Oracle Roles and Privileges

Appendix D - Glossary



© Copyright, Macmillan Computer Publishing. All rights reserved.





**Teach Yourself Oracle 8 In 21 Days**

**Foreword**

"Enabling the information age" is the slogan upon which Oracle is founded, but it is more than a slogan to engineers who dedicate their knowledge and experience to the task of delivering this information with accuracy and speed. In this age of information, just getting the data is not enough. The data that is so vital to business must be dependable, accurate, and delivered as quickly as possible.

Two such engineers I have worked with in this endeavor are Edward T. Whalen and Steve Adrien DeLuca. Edward has won his mastery in the field of performance benchmarks and system tuning. A former employee of COMPAQ Computers and currently CEO of Performance Tuning Corp., Edward has held many benchmark records for the highest degree of performance from the Transaction Processing Performance Council, which is responsible for creating and maintaining industry-standard benchmarks. Edward has also published a book on Oracle tuning, which has given him well-deserved notoriety.

Steve is an Architect Engineer in the COMPAQ products division of Oracle Corporation. Steve holds the distinction of being the co-inventor of the Oracle System Sizer (patent pending), which has won him much acclaim in the field of sizing and capacity planning.

Together, Edward and Steve have delivered lectures in the field of performance tuning, sizing, and capacity planning to audiences over the world. They have also performed benchmarks for such customers as Boeing Aircraft with more than satisfactory results. It is only natural that they would collaborate on this book. I look forward to working with Edward and Steve for many years to come, breaking new ground in the field of performance.

Richard French Vice President Enterprise Platforms Division Oracle Corporation **Acknowledgments**

**Edward Whalen**

Writing the acknowledgments for a book is difficult; I owe my thanks not only to those who worked on the book itself, but to those friends and family who supported and encouraged me to move the book forward. I am always afraid I have missed someone; if I have, I apologize.

I would like to thank Rosemarie Graham, Steve Straiger, Marla Reece-Hall, Kim Hannel, and especially Kate Shoup Welsh at Sams Publishing for their help in the development of the book. The editorial staff at Sams is excellent and absolutely great to work with. Without their help, this book could not have been published. I would also like to thank Kelly Leigh for doing a great job of technical editing. Lastly I would like to thank Erika Millen for creating the index for this book.

I would especially like to thank my good friend Steve DeLuca for his participation in this book. I would also like to thank Richard French for contributing a foreword to this book.

I would like to thank not only the people who have directly influenced this book and helped make it a reality, but those who have indirectly influenced it as well. These people are Bob Nissen, Thomas Cox, Keith Yedlin, Mike Brey, and Eric Speed. I would also like to thank some of the people who have influenced me in my endeavors and have taught me over the years: Lorna Livingtree and Brent Schroeder. Some other people who have been inspired me are Marci Frohock and Bryon Georgson.

In my previous book I failed to mention someone without whom none of this would be possible: Larry Ellison. I apologize. Larry Ellison's hard work made Oracle what it is today.

Writing a book involves a lot of time and effort. I would like to thank my wife, Felicia, for putting up with the sacrifices necessary to write this book as well as for her help in finishing the book.

**Steve Adrien DeLuca**

I would like to take this opportunity to thank the great management staff at Oracle: people like Jerry Baker, Gary Bloom, Jerry Held, Richard French, Bonnie Crater, Sylvia Cummings, Charles Weiss, David Appelbaum, and of course, Larry Ellison, for letting me develop my ideas; talented database engineers such as Andrew Rist for co-inventing Oracle System Sizer (patent pending) with me; and Mike Brey, Bob Nissen, Ethan Berry, John Viguerie, Jose Sanchez, Jeff Plank, Theresa Burghgraef,

Fred Dambrosio, Bryon Georgson, Tamar Sofer-Rothenberg, Vicky Owens, and others too numerous to mention for helping me prove the theories.

What can one say about close friends such as Edward Whalen, who invited me to contribute to this writing, or family like my wife, Jean, and daughter Tina, who have always been there to support me. A special thanks to the folks at COMPAQ Computer such as Ronnie Ward, Keith Carlson, Douglas Mackbee, and Mike Nikolaiev, for letting me tinker around their labs.

**About the Authors**

**Lead Author**

**Edward Whalen** is president and CEO of Performance Tuning Corporation

(www.perftuning.com), a consulting company specializing in database performance, administration, and backup/recovery solutions. He has had extensive experience in database system design and tuning for optimal performance. Mr. Whalen has also written another Sams book, *Oracle Performance Tuning and Optimization*. He is recognized as a leader in Oracle performance tuning and optimization.

He is a representative on the Transaction Processing Performance Council, which is responsible for creating and maintaining industry-standard database benchmarks. As part of this council, he has participated in the development of several TPC benchmarks.

Mr. Whalen currently resides in Cypress, Texas, with his wife, Felicia; their Border collies, Pierce (Dash), Chip, Teller, and Ty; their Great Pyrenees, Shasta; and their cats. He is active in many dog related activities, including dog agility. He is also a certified EMT and volunteers with the local emergency ambulance service, Cypress Creek EMS, where he is a regular on Medic-53, Medic-54, and Medic-55.

**Contributing Author**

**Steve Adrien DeLuca** has been an architect engineer developing performance tools at Oracle Corporation since 1994. Mr. DeLuca co-invented the Oracle System Sizer (patent pending) and has been developing performance tools and lecturing about them around the world since 1984. Prior to working at Oracle Corporation, Mr. DeLuca served as a performance engineer specializing in sizing and capacity planning for companies such as DEC, Tandem, and Apple, and for organizations such as the U.S. Air Force.

**Tell Us What You Think!**

As a reader, you are the most important critic of and commentator on our books. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way. You can help us make strong books that meet your needs and give you the computer guidance you require.

Do you have access to the World Wide Web? Then check out our site at http://www.mcp.com.

**NOTE:** If you have a technical question about this book, call the technical- support line at 317-581-3833 or send e-mail to support@mcp.com.

As the team leader of the group that created this book, I welcome your comments. You can fax, e mail, or write me directly to let me know what you did or didn't like about this book--as well as what we can do to make our books stronger. Here's the information:

Fax: 317-581-4669

E-mail: enterprise\_mgr@sams.mcp.com

Mail: Rosemarie Graham

Comments Department

Sams Publishing

201 W. 103rd Street

Indianapolis, IN 46290

**Introduction**

I have worked with Oracle for many years. Every time a new version is released or a new problem crops up, I am rejuvenated; I become excited about working with Oracle again. I want to share this enthusiasm with you, and I hope that after you become familiar with Oracle, more and more aspects of the Oracle RDBMS will interest you. The most important thing is that you enjoy what you are doing; I hope that you enjoy working with Oracle as much as I do.

The Oracle RDBMS is an enormous environment with unlimited potential. When you start working with Oracle, you might find it overwhelming. Don't give up; you will see how the different components work together as you learn about the Oracle RDBMS. Rarely do I undertake a project without learning something new about Oracle.

**Who Should Read This Book?**

This book is designed for inexperienced Oracle users. Experienced Oracle DBAs will likely find this book too elementary. Most of the lessons are designed to step the reader through specific administrative and user tasks.

**Required Software**

To be able to work through the examples in this book, you must have the following software: ● Oracle 8.0.3 or later

● Oracle Enterprise Manager

● Microsoft Windows NT 4.0 or other OS

Although many of the examples and exercises in this book are geared toward Microsoft Windows NT, you need not run NT. If your system runs UNIX, you will be fine.

**How to Use This Book**

The best way to use this book is to read each lesson, then practice the techniques and tasks outlined in that lesson. Each lesson covers a single topic, so some lessons might be longer than others. If you complete one lesson per day, you can easily complete the full course in three weeks.

At the end of each lesson, you'll find a series of questions and answers. These questions are designed to point out some of the key concepts that were covered in the lesson. Following the Q&A section, you'll find a series of quiz questions that focus on techniques and tasks covered in the lesson. Each lesson also includes a series of exercises that are intended to familiarize you with some of the key tasks covered in that lesson.

**Conventions Used in This Book**

**New Term:** New terms appear in italic, and are accompanied by a new term icon.

**NOTE:** Notes explain interesting or important points that can help you understand concepts and techniques.

**TIP:** Tips are pieces of information that help you in real-world situations. Tips often provide shortcuts or information to make a task easier or faster.

**WARNING:** Warnings provide information about detrimental performance issues or dangerous errors. Pay careful attention to warnings.

**Working with Code and Code Conventions**

**INPUT:** The input icon indicates code that you type.

**OUTPUT:**The output icon indicates the result of running a piece of code.

All code in the listings appears in monospace. Many code-related terms within the text also appear in monospace. Placeholders in code or in text appear in italic monospace.

When a line of code is too long to fit on one line of this book, it is broken at a convenient place and

continued to the next line. A code continuation character precedes the continuation of a line of code. (You should type a line of code that has this character as one long line without breaking it.)

**ANALYSIS:** Paragraphs that begin with the analysis icon explain the preceding code example. The syntax icon identifies syntax statements.

Throughout this book there will be references to SQL statements and Oracle administration statements. These will be distinguished in the following manner:

● Any Oracle or SQL keyword that must be typed exactly will be displayed in uppercase letters, as in SELECT. This syntax does not necessarily require uppercase characters in its execution, but will be displayed that way for consistency.

● Statements that can be or need to be altered for your own particular configuration are indicated by italics, as in SELECT \* FROM table\_name. This would indicate that table\_name should be substituted with your own table's name.

● The use of braces indicates a choice of several optional variables. An example of this would be the following SQL statement: SELECT [USER\_NAME or USER\_ID] FROM table\_name;. This would indicate that the keywords USER\_NAME or USER\_ID must be used in that SQL query.

● The use of brackets indicates a choice of various optional parameters. In this case, the parameter may be used or omitted as necessary. An example of this is the SQL statement SELECT USER\_NAME, GROUP\_NAME FROM table\_name [ORDER BY GROUP\_NAME];.

**Author's Note**

Most of this book was written before the production version of Oracle8 was available. I have tried to change any example or explanation of a task that has changed since Oracle8 beta 2. I believe I have made all of these corrections, but if you see a figure or an explanation of a task that is not quite correct, it is because the original section was based on that beta release.



© Copyright, Macmillan Computer Publishing. All rights reserved.



**Teach Yourself Oracle 8 In 21 Days**

****

**- Week 1 At a Glance -**

You will spend Days 1-3 in the introductory section of the book, where the foundation for all subsequent lessons is built. This section begins with an introduction to Oracle and database concepts. Here you will learn a bit of Oracle history as well as read an explanation of terms and concepts. Subsequent lessons contain an introduction to the Oracle8 architecture. Understanding the architecture and operation of Oracle8 can offer great insight into why many DBA actions are taken. A tutorial on how to install Oracle8 will also be provided in this section. Finally, my good friend Steve DeLuca, world-renowned capacity planning expert from Oracle, will teach you how to size a system and plan for future growth.

**Managing Database Storage**

Days 4-7 cover topics concerning management of the Oracle database (including extensive use of Enterprise Manager), databases and datafiles, and tablespaces. Coverage regarding managing database storage continues into week 2.



© Copyright, Macmillan Computer Publishing. All rights reserved.



**Teach Yourself Oracle 8 In 21 Days**

****

**- Day 1 -**

**Starting Out with Oracle**

Being an Oracle database operator or administrator can be a demanding but rewarding career that carries with it a great deal of responsibility as well as authority. This book is intended to help you embark on this exciting path. I hope that within the pages of this book I can convey some of the enthusiasm and excitement I feel when working with state-of-the-art hardware and software such as Oracle8.

I think the best way to grasp a concept is to fully understand why actions are taken and the consequences of those actions. If you understand how Oracle works and its interactions with the operating system and hardware, you can more easily predict and anticipate the result of actions you take. In this book, I attempt to fully explain the workings of Oracle and the supporting software and hardware.

**A Brief History of Oracle**

In 1977, Larry Ellison, Bob Miner, and Ed Oates formed a company called Relational Software Incorporated (RSI). This company built an RDBMS called *Oracle*. Ellison, Miner, and Oates made a key decision: to develop their RDBMS using C and the SQL interface. Soon after, they came out with version 1, a prototype. In 1979, RSI delivered its first product to customers. The Oracle RDBMS version 2 worked on the Digital PDP-11 running the RSX-11 operating system and was soon ported to the DEC VAX system.

1983 heralded the release of version 3, which touted changes in the SQL language as well as performance enhancements and other improvements. Unlike earlier versions, version 3 was written almost entirely in C. At this point, RSI changed its name to Oracle Corporation.

Oracle version 4 was released in 1984. This version supported both the VAX system and the IBM VM operating system. Version 4 was the first version to incorporate read consistency. Version 5,

introduced in 1985, was a milestone because it introduced client/server computing to the market with the use of SQL\*Net. Version 5 was also the first MS-DOS product to break through the 640KB barrier.

In 1988, Oracle presented version 6, which introduced low-level locking as well as a variety of performance improvements and functionality enhancements, including sequence generation and deferred writes. I was introduced to Oracle6 back in the days when we ran the TP1, TPC-A, and TPC B benchmarks. At this point, Oracle was running on a large variety of different platforms and operating systems. In 1991, Oracle introduced the Oracle Parallel Server option on version 6.1 of the Oracle RDBMS on the DEC VAX platform. Soon the Parallel Server option was available on a variety of platforms.

Oracle7, released in 1992, included many architectural changes in the area of memory, CPU, and I/O utilization. Oracle7 is the full-featured RDBMS to which you are accustomed, the one you've been using for many years. Oracle7 introduced many advances in the area of ease of use, such as the SQL\*DBA tools and database roles.

Finally, in 1997 Oracle introduced Oracle8, which added object extensions as well as a host of new features and administrative tools.

For more information about the history of Oracle (specifically about the Oracle server), check out the two-part article by Ken Jacobs in the January/February and March/April 1995 issues of *Oracle Magazine*.

For more information about the Oracle corporation, its products, and about working with Oracle, check out www.oracle.com. This Web site contains a wealth of information about Oracle parterships and products as well as information about the Oracle Developer Program, which specifically assists developers.

**Introduction to Terms**

Many different terms and concepts will be used throughout this book. I've introduced them here to make it easier for you to grasp many of the concepts and lessons to follow. If you encounter other terms with which you are unfamiliar, check out Appendix D, "Glossary."

**Ad-Hoc Query**

This use of the Latin term means an impromptu, simple query.

**Block**

A block is the smallest unit of storage in an Oracle database. The database block contains header information concerning the block itself as well as the data or PL/SQL code. The Oracle block size is configurable with the minimum size being 2KB and the maximum size being 16KB.

**Bottleneck**

In computer terms, a bottleneck is a system component that limits the performance of the system. **Buffer**

This term refers to an amount of memory used to store data. A buffer stores data that is about to be used or that has just been used. In many cases, buffers are in-memory copies of data that is also on disk. Buffers can be used as a copy of data for quick read access, they can be modified and written to disk, or they can be created in memory as temporary storage.

In Oracle, database buffers of the SGA store the most recently used blocks of database data. The set of database block buffers is known as the *database buffer cache*. The buffers used to temporarily store redo entries until they can be written to disk are known as *redo log buffers*.

**Cache**

A cache is a storage area used to provide fast access to data. In hardware terms, the cache is a small (relative to main RAM) amount of memory that is much faster than main memory. This memory is used to reduce the time it takes to reload frequently used data or instructions into the CPU. CPU chips themselves contain small amounts of memory built in as cache.

In Oracle, the block buffers and shared pool are considered caches because they are used to store data and instructions for quick access. Caching is very effective in reducing the time it takes to retrieve frequently used data.

Caching usually works using a least recently used algorithm. Data that has not been used for a while is eventually released from the cache to make room for new data. If data is requested and is in the cache (a phenomenon called a *cache hit*), the data is retrieved from the cache, preventing it from having to be retrieved from memory or disk. After the data has been accessed again, it is marked as recently used and put on the top of the cache list.

**Checkpoint**

A checkpoint is an operation that forces all changed, in-memory data blocks to be written out to disk. This is a key factor in how long the database takes to recover in the event of a failure. This concept is discussed in depth on Day 2, "Exploring the Oracle Architecture."

**Clean Buffer**

A *clean buffer* is a buffer that has not been modified. Because this buffer has not been changed, it is not necessary for the DBWR to write this buffer to disk.

**Concurrency**

This term refers to the capability to perform many functions at the same time. Oracle provides for concurrency by allowing many users to access the database simultaneously.

**Database**

A database is a set of data, organized for easy access. The database is the actual data. It is the database that you will be accessing when you need to retrieve data.

**Data Dictionary**

The data dictionary is a set of tables Oracle uses to maintain information about the database. The data dictionary contains information about tables, indexes, clusters, and so on.

**DBA (Database Administrator)**

The DBA is the person responsible for the operation, configuration, and performance of the database. The DBA is charged with keeping the database operating smoothly, ensuring that backups are done on a regular basis (and that the backups work), and installing new software. Other responsibilities might include planning for future expansion and disk space needs, creating databases and tablespaces, adding users and maintaining security, and monitoring the database and retuning it as necessary. Large installations might have teams of DBAs to keep the system running smoothly; alternatively, the tasks might be segmented among the DBAs.

**DBMS or RDBMS**

The Database Management System is the software and collection of tools that manages the database. Oracle software is the DBMS. A Relational Database Management System is a DBMS that is relational in nature. This means that the internal workings access data in a relational manner. Oracle is an RDBMS.

**DDL (Data Definition Language) Commands**

These commands are used in the creation and modification of schema objects. These commands provide the ability to create, alter, and drop objects; grant and revoke privileges and roles; establish auditing options; and add comments to the data dictionary. These commands are related to the management and administration of the Oracle database. Before and after each DDL statement, Oracle implicitly commits the current transaction.

**Dirty Buffer**

A *dirty buffer* is a buffer that has been modified. It is the job of the DBWR to eventually write all dirty block buffers out to disk.

**DML (Data Manipulation Language) Commands**

These commands allow you to query and modify data within existing schema objects. Unlike the DDL commands, a commit is not implicit. DML statements consist of DELETE, INSERT, SELECT, and UPDATE statements; EXPLAIN PLAN statements; and LOCK TABLE statements.

**Dynamic Performance Tables**

These tables are created at instance startup and used to store information about the performance of the instance. This information includes connection information, I/Os, initialization parameter values, and so on.

**Function**

A function is a set of SQL or PL/SQL statements used together to execute a particular function. Procedures and functions are identical except that functions always return a value (procedures do not). By processing the SQL code on the database server, you can reduce the number of instructions sent across the network and returned from the SQL statements.

**IM (Information Management)**

This term is usually used to describe the department that handles your corporate data. **IS (Information Systems)**

This term is also used to describe the department that handles your corporate data. **IT (Information Technology)**

This term is used to describe the business of managing information.

**Network Computing Architecture (NCA)**

The Network Computing Architecture is a standard for computing over the network. The NCA was developed in conjunction with Oracle.

**Physical Memory**

This term refers to the actual hardware RAM (Random Access Memory) available in the computer for use by the operating system and applications.

**Procedure**

A procedure is a set of SQL or PL/SQL statements used together to execute a particular function. Procedures and functions are identical except that functions always return a value (procedures do not).

By processing the SQL code on the database server, you can reduce the number of instructions sent across the network and returned from the SQL statements.

**Program Unit**

In Oracle, program unit is used to describe a package, a stored procedure, or a sequence. **Query**

A query is a read-only transaction against a database. A query is generated using the SELECT statement. Users generally distinguish between queries and other transaction types because a query does not the change data in the database.

**Schema**

A schema is a collection of objects associated with the database.

**Schema Objects**

Schema objects are abstractions or logical structures that refer to database objects or structures. Schema objects consist of such things as clusters, indexes, packages, sequences, stored procedures, synonyms, tables, views, and so on.

**System Global Area (SGA)**

The SGA is a shared-memory region that Oracle uses to store data and control information for one Oracle instance. The SGA is allocated when the Oracle instance starts; it is deallocated when the Oracle instance shuts down. Each Oracle instance that starts has its own SGA. The information in the SGA is made up of the database buffers, the redo log buffer, and the shared pool; each has a fixed size and is created at instance startup.

**Transaction**

A transaction is a logical unit of work consisting of one or more SQL statements, ending in a commit or a rollback. Performance measurements often use the number of transactions per second or per minute as the performance metric.

**Trigger**

A trigger is a mechanism that allows you to write procedures that are automatically executed whenever an INSERT, UPDATE, or DELETE statement is executed on a table or view. Triggers can be used to enforce integrity constraints or automate some other custom function.

**Virtual Memory**

This term refers to the memory that can be used for programs in the operating system. To overcome the limitations associated with insufficient physical memory, virtual memory allows programs to run that are larger than the amount of physical memory in the system. When there is not enough physical

memory in the system, these programs are copied from RAM to a disk file called a *paging* or *swap file*. This arrangement allows small systems to run many programs. You pay a performance penalty when the computer pages or swaps.

**Storage Units**

Data is stored in the computer in a binary form. The units used to refer to this binary data are as follows:

***Term Definition Comment***

bit The smallest unit of data storage

A bit is either a 1 or a 0.

nibble 4 bits This term is not commonly used. byte 8 bits The most commonly used storage unit.

word This term is architecture dependent

On some systems, a word is 16 bits; on others, a word is 32 or 64 bits.

kilobyte (KB) Even though *kilo* usually means 1,000, a kilobyte in computer terms is actually

1,024 bytes (because we like powers of

2).

megabyte (MB) The term megabyte denotes 1,024KB or 1,048,576 bytes.

gigabyte (GB) A gigabyte is 1,024 megabytes or

1,073,741,824 bytes.

terabyte (TB) A terabyte is 1,024 gigabytes or

1,099,511,627,776 bytes.

It is not uncommon to hear large data warehousing sites talk in terms of terabytes. In the next few years, you will probably hear of systems using storage in the tens and hundreds of terabytes.

**Oracle Configurations**

There are many different types of Oracle configurations and uses. Let's look at some of these different types of systems and analyze their usage and characteristics.

**OLTP**

The Online Transaction Processing (OLTP) system is probably the most common of the RDBMS configurations. OLTP systems have online users that access the system. These systems are typically used for order-entry purposes, such as for retail sales, credit-card validation, ATM transactions, and so

on.

**Characteristics of OLTP Systems**

OLTP systems typically support large numbers of online users simultaneously accessing the RDBMS. Because users are waiting for data to be returned to them, any excessive response time is immediately noticeable. OLTP systems are characteristically read and write intensive. Depending on the specific application, this read/write ratio might vary.

**DSS**

The Decision Support System (DSS) is used to assist with the decision-making process. These decisions might be based on information such as how sales in a particular region are doing, what cross section of customers is buying a particular product, or to whom to send a mailing. The DSS system is used to help make decisions by providing good data.

**Characteristics of a DSS**

The DSS is characterized by long-running queries against a large set of data. Unlike the OLTP system, where users are waiting for data to return to them online, here users expect the queries to take minutes, hours, or days to complete. The data is typically generated from a different source and loaded onto the DSS computer in bulk. Except for during the load, the DSS system is characterized by being read

intensive (with very few writes).

**Data Warehouse**

A data warehouse is typically considered to be a large-scale system that consists of both DSS and OLTP components. These systems are typically hundreds of gigabytes in size and support many users.

**Characteristics of a Data Warehouse**

Data warehouses have some of the attributes of a DSS system, such as long-running queries and a possible online component. In many cases, this component is the source of the data used in the DSS queries.

**Data Mart**

A data mart, which is a smaller-scale version of a data warehouse, serves many of the same functions as a data warehouse.

**Characteristics of a Data Mart**

A data mart is typically 100GB or less in size. As with a data warehouse, a data mart supports many online users as well as a decision-support function.

**Video Server**

A video server can support large numbers of video data streams. These video streams can be used for purposes such as video on demand for entertainment as well as training functions.

**Characteristics of a Video Server**

The video server system must support a high network bandwidth in order to support multiple data streams. The video server must also be able to support a high I/O bandwidth. These disk accesses are typically of a very large block size and sequential in nature.

**Web Server**

The Oracle Web server is designed to support both static and dynamic Web pages. These pages can be simple Web pages or complex database-generated pages. Oracle Web server systems are also typically used in Web commerce applications. These installations can allow the customer to browse online catalogs, which might feature graphics or even video. The customer can then purchase items online.

**Characteristics of an Oracle Web Server**

The Oracle Web server typically supports many online users. There is typically a large amount of data that has been accessed frequently and other data that is less frequently accessed. A large amount of memory can help improve performance in this type of configuration.

**OLAP**

The term OLAP (Online Analytical Processing) is usually used in relation with multidimensional data. OLAP users might be financial analysts or marketing personnel looking at global data.

**Characteristics of an OLAP System**

An OLAP system typically involves a large amount of disk space with heavy I/O and memory requirements. An OLAP system might support only a few or many users. This depends on your type of configuration.

**Roles and Responsibilities of an Oracle DBA**

If you want to become an Oracle DBA, you should first understand what an Oracle DBA's job is. The basic roles of the DBA are fairly consistent among different companies, but these duties might be expanded based on the size of the company and the experience of the DBA. In fact, the DBA is considered the main resource for DBMS experience and knowledge in many companies.

Let's look at these roles and responsibilities and determine what skills are necessary to fulfill these duties. Here the roles and responsibilities are divided into two categories: basic duties and additional duties. The dividing line between these is not clear; there is significant overlap.

**Basic Duties of the DBA**

Here are some of the basic roles of the Oracle DBA. This is not an all-inclusive list. Depending on your installation and staff, your duties might not include all of these, or might include many more items. This section is simply intended as a general guide.

● Installation of new software--It is primarily the job of the DBA to install new versions of Oracle software, application software, and other software related to DBMS administration. It is important that the DBA or other IS staff members test this new software before it is moved into a production environment.

● Configuration of hardware and software with the system administrator-- In many cases the system software can only be accessed by the system administrator. In this case, the DBA must work closely with the system administrator to perform software installations, and to configure hardware and software so that it functions optimally with the DBMS.

● Security administration--One of the main duties of the DBA is to monitor and administer DBMS security. This involves adding and removing users, administering quotas, auditing, and checking for security problems.

● Performance tuning and monitoring--The DBA must continually monitor system performance and be prepared to retune the system as necessary. Even a well-tuned system must be constantly monitored and adjusted. Sometimes this involves changing tuning parameters, other times this involves rebuilding an index or restructuring a table.

● Backup and recovery--Perhaps the most important responsibility of the DBA is protecting the data in the system. To effectively do this, you must develop an effective backup and recovery strategy and make sure it is carried out. A DBA's chief responsibility is to maintain the integrity of the database. It is important that the backup and recovery process be periodically tested.

● Routine scheduled maintenance--It is the job of the DBA to schedule routine DBMS maintenance and carry out this maintenance. This maintenance is regularly carried out in the early hours of the morning or on weekends when this maintenance causes the least inconvenience to the user community.

● Troubleshooting:--In the event of a system or DBMS failure, it is the job of the DBA to troubleshoot or assist in the Troubleshooting: of the problem. The DBA might also participate in or lead the effort to find and eliminate problems or potential problems.

● Failure recovery--Because a system failure can mean that the users do not have access to their data, it can be the job of the DBA to lead efforts to recover from system failures. The well-

prepared DBA has contingency plans for system outages and can soon have the DBMS running again.

**Additional Duties of the DBA**

Some of the more advanced duties of the Oracle DBA might include the following:

● Data analysis--The DBA will frequently be called on to analyze the data stored in the database and to make recommendations relating to performance and efficiency of that data storage. This might relate to the more effective use of indexes or the use of some feature such as the Parallel Query option.

● Database design (preliminary)--The DBA is often involved at the preliminary database-design stages. Through the involvement of the DBA, many problems that might occur can be eliminated. The DBA knows the DBMS and system, can point out potential problems, and can help the development team with special performance considerations.

● Data modeling and optimization--By modeling the data, it is possible to optimize the system layout to take the most advantage of your I/O subsystem.

● Assisting developers with SQL and stored procedure development--The DBA should be prepared to be a resource for developers and users. The DBA is often called on to help with SQL problems as well as to design and write stored procedures.

● Enterprise standards and naming conventions--Because many different groups might perform different roles in developing and deploying applications, it is often the DBA who is called on to help define enterprise standards and naming conventions as well as to ensure that new applications are conforming to these standards.

● Development of production migration procedures--Because the DBA is responsible for the availability and reliability of the DBMS and applications using that DBMS, it is up to the DBA to develop and maintain procedures for rolling out new applications and DBMS software. This involves evaluating new software or patches as well as testing them. It is up to the DBA to guarantee the stability and robustness of the system.

● Environmental documentation--The DBA should document every aspect of the DBMS environment, including hardware configuration and maintenance records, software updates, changes to the applications and DBMS, and all other items related to changes made to the system. The DBA should be able to access these records and fully reproduce the current system as necessary.

● Consult with development team and end users--The DBA is often called on to act as a consultant to the development team as well as to the user community. This might involve personally assisting a single user or developing training courses for the user community as a whole.

● Evaluation of new software--The DBA might be called on to evaluate new software and make recommendations based on that evaluation. This might be related to a software purchase or a scheduled rollout of a new version of software. This evaluation must be done in the context of the stability of the system. It is your responsibility to maintain system stability and reliability.

● Evaluation of new hardware and software purchases--There is much consideration involved in purchasing new hardware and software. Much of this consideration involves the functionality and compatibility of the software or hardware as well as the cost of these components. Although the cost of the item is not usually a concern of the DBA, the functionality and compatibility is. The DBA might be asked to make recommendations based on whether these purchases make sense.

● Capacity planning and sizing--Determining whether it is necessary to purchase new hardware or software to meet increased loads is often a job for the DBA. Capacity planning and sizing is important to provide the level of service your users require. By anticipating the future needs of your users, you can provide an excellent level of service with no interruptions.

**Summary**

This lesson introduces some of the topics you will see in the rest of the book. First you saw a brief history of how Oracle got where it is today. Then you examined number of terms that you will see throughout the book. These terms are important; you will use them every day in your job as a DBA. Finally, you were presented with some of your tasks and responsibilities as a DBA.

**What's Next?**

Tomorrow's lesson examines the structure and operation of Oracle. By having an understanding of how Oracle works, you can better understand how to administer it. You will look at some of the new features in Oracle8, as well as receive an overview of Oracle performance.

**Q&A**

**Q What units of measurement are typically used in databases?**

**A** With Oracle systems you usually discuss size in terms of megabytes and gigabytes, but some systems are growing into the terabyte range.

**Q Are the duties of the DBA the same for all companies?**

**A** No, far from it. No two sites are the same. Although the basic duties and responsibilities might be similar, the extended duties are always different.

**Q Why is it important to document?**

**A** If you document the system configuration and logging changes, you will have a much easier time reproducing the system in the event of a failure. By having configuration information in a log book you can save numerous hours of trial and error in reconfiguring the system.

**Workshop**

The workshop provides quiz questions to help you solidify your understanding of the material covered. For answers to quiz questions, see Appendix A, "Answers."

**Quiz**

**1.** How is a DBMS different from a database?

**2.** What is a DDL statement?

**3.** What is a DML statement?

**4.** What are some of the characteristics of an OLTP system?

**5.** What are some of the characteristics of a DSS system?

**6.** State five duties of an Oracle DBA.

**7.** What is the most important duty of an Oracle DBA?



© Copyright, Macmillan Computer Publishing. All rights reserved.



**Teach Yourself Oracle 8 In 21 Days**

****

**- Day 2 -**

**Exploring the Oracle Architecture**

**New Term:** The Oracle Relational Database Management System, or *RDBMS*, is designed to allow simultaneous access to large amounts of stored information. The RDBMS consists of the database (the information) and the instance (the embodiment of the system). The database contains the physical files that reside on the system and the logical pieces such as the database schema. These database files take various forms, as described in the following section. The instance is the method used to access the data and consists of processes and system memory.

**NOTE:** Object extensions have been added to the RDBMS with Oracle8. The object extension to tables is covered in detail on Day 12, "Working with Tables, Views, and Synonyms." Oracle refers to Oracle8 as an O-RDBMS (Object-Relational Database Management System). In this book, I refer to Oracle as an RDBMS for clarity.

**The Database**

The Oracle database has a logical layer and a physical layer. The physical layer consists of the files that reside on the disk; the components of the logical layer map the data to these physical components.

**The Physical Layer**

The physical layer of the database consists of three types of files:

● One or more datafiles--Datafiles store the information contained in the database. You can have as few as one datafile or as many as hundreds of datafiles. The information for a single table can span many datafiles or many tables can share a set of datafiles. Spreading tablespaces over many datafiles can have a significant positive effect on performance. The number of datafiles

that can be configured is limited by the Oracle parameter MAXDATAFILES.

● Two or more redo log files--Redo log files hold information used for recovery in the event of a system failure. Redo log files, known as the redo log, store a log of all changes made to the database. This information is used in the event of a system failure to reapply changes that have been made and committed but that might not have been made to the datafiles. The redo log files must perform well and be protected against hardware failures (through software or hardware fault tolerance). If redo log information is lost, you cannot recover the system.

● One or more control files--Control files contain information used to start an instance, such as the location of datafiles and redo log files; Oracle needs this information to start the database instance. Control files must be protected. Oracle provides a mechanism for storing multiple copies of control files.

**The Logical Layer**

The logical layer of the database consists of the following elements:

● One or more tablespaces.

● The database schema, which consists of items such as tables, clusters, indexes, views, stored procedures, database triggers, sequences, and so on.

**Tablespaces and Datafiles**

**New Term:** The database is divided into one or more logical pieces known as *tablespaces*. A tablespace is used to logically group data together. For example, you can create one tablespace for accounting and a separate tablespace for purchasing. Segmenting groups into different tablespaces simplifies the administration of these groups (see Figure 2.1). Tablespaces consist of one or more datafiles. By using more than one datafile per tablespace, you can spread data over many different disks to distribute the I/O load and improve performance.

**Figure 2.1.**

*The relationship between the database, tablespaces, and datafiles.*

As part of the process of creating the database, Oracle automatically creates the SYSTEM tablespace for you. Although a small database can fit within the SYSTEM tablespace, it's recommended that you create a separate tablespace for user data. The SYSTEM tablespace is where the data dictionary is kept. The data dictionary contains information about tables, indexes, clusters, and so on.

Datafiles can be operating system files or, in the case of some operating systems, RAW devices. Datafiles and data access methods are described in detail on Day 12.

**The Database Schema**

**New Term:** The database schema is a collection of logical-structure objects, known as *schema objects*, that define how you see the database's data. These schema objects consist of structures such as tables, clusters, indexes, views, stored procedures, database triggers, and sequences.

● Table--A table, which consists of a tablename and rows and columns of data, is the basic logical storage unit in the Oracle database. Columns are defined by name and data type. A table is stored within a tablespace; often, many tables share a tablespace.

● Cluster--A cluster is a set of tables physically stored together as one table that shares a common column. If data in two or more tables is frequently retrieved together based on data in the common column, using a clustered table can be quite efficient. Tables can be accessed separately even though they are part of a clustered table. Because of the structure of the cluster, related data requires much less I/O overhead if accessed simultaneously.

● Index--An index is a structure created to help retrieve data more quickly and efficiently (just as the index in this book allows you to find a particular section more quickly). An index is declared on a column or set of columns. Access to the table based on the value of the indexed column(s) (as in a WHERE clause) will use the index to locate the table data.

**NOTE:** A new feature in Oracle8 is the index-only table. In an index-only table, the data and index are stored together. This is discussed in detail on Day 13, "Using Indexes and Sequences."

● View--A view is a window into one or more tables. A view does not store any data; it presents table data. A view can be queried, updated, and deleted as a table without restriction. Views are typically used to simplify the user's perception of data access by providing limited information from one table, or a set of information from several tables transparently. Views can also be used to prevent some data from being accessed by the user or to create a join from multiple tables.

● Stored procedure--A stored procedure is a predefined SQL query that is stored in the data dictionary. Stored procedures are designed to allow more efficient queries. Using stored procedures, you can reduce the amount of information that must be passed to the RDBMS and thus reduce network traffic and improve performance.

● Database trigger--A database trigger is a procedure that is run automatically when an event occurs. This procedure, which is defined by the administrator or developer, triggers, or is run whenever this event occurs. This procedure could be an insert, a deletion, or even a selection of data from a table.

● Sequence--The Oracle sequence generator is used to automatically generate a unique sequence

of numbers in cache. By using the sequence generator you can avoid the steps necessary to create this sequence on your own such as locking the record that has the last value of the sequence, generating a new value, and then unlocking the record.

**Segments, Extents, and Data Blocks**

Within Oracle, the space used to store data is controlled by the use of logical structures. These structures consist of the following:

● Data blocks--A block is the smallest unit of storage in an Oracle database. The database block contains header information concerning the block itself as well as the data.

● Extents--Extents consist of data blocks.

● Segments--A segment is a set of extents used to store a particular type of data, as shown in Figure 2.2.

**Figure 2.2.**

*Segments, extents, and data blocks.*

**Segments**

An Oracle database can use four types of segments:

● Data segment--Stores user data within the database.

● Index segment--Stores indexes.

● Rollback segment--Stores rollback information used when data must be rolled back.

● Temporary segment--Created when a SQL statement needs a temporary work area; these segments are destroyed when the SQL statement is finished. These segments are used during various database operations, such as sorts.

**Extents**

Extents are the building blocks of segments; in turn, they consist of data blocks. An extent is used to minimize the amount of wasted (empty) storage. As more and more data is entered into tablespaces in your database, the extents used to store that data can grow or shrink as necessary. In this manner, many tablespaces can share the same storage space without preallocating the divisions between those tablespaces.

At tablespace-creation time, you can specify the minimum number of extents to allocate as well as the

number of extents to add at a time when that allocation has been used. This arrangement gives you efficient control over the space used in your database.

**Data Blocks**

Data blocks are the smallest pieces of an Oracle database; they are physically stored on disk. Although the data block in most systems is 2KB (2,048 bytes), you can change this size for efficiency depending on your application or operating system.

**NOTE:** Oracle blocks do not need to be, and may not be the same as, operating system data blocks. In fact, in most cases they are not.

**The Oracle Instance**

The Oracle instance consists of the Oracle processes and shared memory necessary to access information in the database. The instance is made up of the user processes, the Oracle background processes, and the shared memory used by these processes (see Figure 2.3).

**The Oracle Memory Structure**

**New Term:** Oracle uses shared memory for several purposes, including caching of data and indexes as well as storing shared program code. This shared memory is broken into various pieces, or *memory structures*. The basic memory structures associated with Oracle are the System Global Area (SGA) and the Program Global Area (PGA).

**Figure 2.3.**

*The Oracle instance.*

**The System Global Area (SGA)**

The SGA is a shared memory region that Oracle uses to store data and control information for one Oracle instance. The SGA is allocated when the Oracle instance starts and deallocated when the Oracle instance shuts down. Each Oracle instance that starts has its own SGA. The information in the SGA consists of the following elements, each of which has a fixed size and is created at instance startup:

The database buffer cache--This stores the most recently used data blocks. These blocks can contain modified data that has not yet been written to disk (sometimes known as *dirty blocks*), blocks that have not been modified, or blocks that have been written to disk since modification (sometimes known as *clean blocks*). Because the buffer cache keeps blocks based on a most recently used

algorithm, the most active buffers stay in memory to reduce I/O and improve performance.

● The redo log buffer--This stores redo entries, or a log of changes made to the database. The redo log buffers are written to the redo log as quickly and efficiently as possible. Remember that the redo log is used for instance recovery in the event of a system failure.

● The shared pool--This is the area of the SGA that stores shared memory structures such as shared SQL areas in the library cache and internal information in the data dictionary. The shared pool is important because an insufficient amount of memory allocated to the shared pool can cause performance degradation. The shared pool consists of the library cache and the data-dictionary cache.

**The Library Cache**

The library cache is used to store shared SQL. Here the parse tree and the execution plan for every unique SQL statement are cached. If multiple applications issue the same SQL statement, the shared SQL area can be accessed by each to reduce the amount of memory needed and to reduce the processing time used for parsing and execution planning.

**The Data-Dictionary Cache**

The data dictionary contains a set of tables and views that Oracle uses as a reference to the database. Oracle stores information here about the logical and physical structure of the database. The data dictionary contains information such as the following:

● User information, such as user privileges

● Integrity constraints defined for tables in the database

● Names and data types of all columns in database tables

● Information on space allocated and used for schema objects

The data dictionary is frequently accessed by Oracle for the parsing of SQL statements. This access is essential to the operation of Oracle; performance bottlenecks in the data dictionary affect all Oracle users. Because of this, you should make sure that the data-dictionary cache is large enough to cache this data. If you do not have enough memory for the data-dictionary cache, you see a severe performance degredation. If you ensure that you have allocated sufficient memory to the shared pool where the data-dictionary cache resides, you should see no performance problems.

**The Program Global Area (PGA)**

The PGA is a memory area that contains data and control information for the Oracle server processes. The size and content of the PGA depends on the Oracle server options you have installed. This area

consists of the following components:

● Stack space--This is the memory that holds the session's variables, arrays, and so on.

● Session information--If you are not running the multithreaded server, the session information is stored in the PGA. If you are running the multithreaded server, the session information is stored in the SGA.

● Private SQL area--This is an area in the PGA where information such as binding variables and runtime buffers is kept.

**Processes**

**New Term:** In many operating systems, traditional processes have been replaced by *threads* or *lightweight processes*. The term *process* is used in this book to describe a thread of execution, or a mechanism that can execute a set of code; *process* refers to the mechanism of execution and can refer to a traditional process or a thread.

The Oracle RDBMS uses two types of processes: user processes and Oracle processes (also known as background processes). In some operating systems (such as Windows NT), these processes are actually threads; for the sake of consistency, I will refer to them as *processes*.

**User Processes**

User, or client, processes are the user's connections to the RDBMS system. The user process manipulates the user's input and communicates with the Oracle server process through the Oracle program interface. The user process is also used to display the information requested by the user and, if necessary, can process this information into a more useful form.

**Oracle Processes**

Oracle processes perform functions for users. Oracle processes can be split into two groups: server processes (which perform functions for the invoking process) and background processes (which perform functions on behalf of the entire RDBMS).

**Server Processes (Shadow Processes)**

Server processes, also known as shadow processes, communicate with the user and interact with Oracle to carry out the user's requests. For example, if the user process requests a piece of data not already in the SGA, the shadow process is responsible for reading the data blocks from the datafiles into the SGA. There can be a one-to-one correlation between user processes and shadow processes (as in a dedicated server configuration); although one shadow process can connect to multiple user processes (as in a multithreaded server configuration), doing so reduces the utilization of system

resources.

**Background Processes**

Background processes are used to perform various tasks within the RDBMS system. These tasks vary from communicating with other Oracle instances and performing system maintenance and cleanup to writing dirty blocks to disk. Following are brief descriptions of the nine Oracle background processes:

● DBWR (Database Writer)--DBWR is responsible for writing dirty data blocks from the database block buffers to disk. When a transaction changes data in a data block, that data block need not be immediately written to disk. Therefore, the DBWR can write this data to disk in a manner that is more efficient than writing when each transaction completes. The DBWR usually writes only when the database block buffers are needed for data to be read. Data is written in a least recently used fashion. For systems in which asynchronous I/O (AIO) is available, there should be only one DBWR process. For systems in which AIO is not available, performance can be greatly enhanced by adding more DBWR processes.

● LGWR (Log Writer)--The LGWR process is responsible for writing data from the log buffer to the redo log.

● CKPT (Checkpoint)--The CKPT process is responsible for signaling the DBWR process to perform a checkpoint and to update all the datafiles and control files for the database to indicate the most recent checkpoint. A checkpoint is an event in which all modified database buffers are written to the datafiles by the DBWR. The CKPT process is optional. If the CKPT process is not present, the LGWR assumes these responsibilities.

● PMON (Process Monitor)--PMON is responsible for keeping track of database processes and cleaning up if a process prematurely dies (PMON cleans up the cache and frees resources that might still be allocated). PMON is also responsible for restarting any dispatcher processes that might have failed.

● SMON (System Monitor)--SMON performs instance recovery at instance startup. This includes cleaning temporary segments and recovering transactions that have died because of a system crash. The SMON also defragments the database by coalescing free extents within the database.

● RECO (Recovery)--RECO is used to clean transactions that were pending in a distributed database. RECO is responsible for committing or rolling back the local portion of the disputed transactions.

● ARCH (Archiver)--ARCH is responsible for copying the online redo log files to archival storage when they become full. ARCH is active only when the RDBMS is operated in ARCHIVELOG mode. When a system is not operated in ARCHIVELOG mode, it might not be possible to recover after a system failure. It is possible to run in NOARCHIVELOG mode under certain circumstances, but typically should operate in ARCHIVELOG mode.

● LCK*n* (Parallel Server Lock)--Up to 10 LCK processes are used for interinstance locking when the Oracle Parallel Server option is used.

● D*nnn* (Dispatcher)--When the Multithreaded Server option is used, at least one Dispatcher process is used for every communications protocol in use. The Dispatcher process is responsible for routing requests from the user processes to available shared server processes and back.

**How Transactions Work**

**New Term:** To give you a better idea how Oracle operates, this section analyzes a sample transaction. Throughout this book, the term *transaction* is used to describe a logical group of work that can consist of one or many SQL statements and must end with a commit or a rollback. Because this example is of a client/server application, SQL\*Net is necessary. The following steps are executed to complete the

transaction:

**1.** The application processes the user input and creates a connection to the server via SQL\*Net. **2.** The server picks up the connection request and creates a server process on behalf of the user.

**3.** The user executes a SQL statement or statements. In this example, the user changes the value of a row in a table.

**4.** The server process checks the shared pool to see whether there is a shared SQL area that has this identical SQL statement. If it finds an identical shared SQL area, the server process checks whether the user has access privileges to the data. If so, the server process uses the shared SQL area to process the request. If a shared SQL area is not found, a new shared SQL area is

allocated, and the statement is parsed and executed.

**5.** The server process finds the data in the SGA (if it is present there) or reads the data from the datafile into the SGA.

**6.** The server process modifies the data in the SGA. Remember that the server processes can read only from the datafiles. At some later time, the DBWR process writes the modified blocks to permanent storage.

**7.** The user executes either the COMMIT or ROLLBACK statement. A COMMIT will finalize the transaction, a ROLLBACK will undo the changes. If the transaction is being committed, the LGWR process immediately records the transaction in the redo log file.

**8.** If the transaction is successful, a completion code is returned across the network to the client process. If a failure has occurred, an error message is returned.

**NOTE:** A transaction is not considered committed until the write to the redo log file is complete. This arrangement ensures that in the event of a system failure, a committed transaction can be recovered. If a transaction has been committed, it is set in stone.

While transactions occur, the Oracle background processes do their jobs, keeping the system running smoothly. While this process occurs, hundreds of other users might be performing similar tasks. Oracle's job is to keep the system in a consistent state, to manage contention and locking, and to perform at the necessary rate.

This overview is intended to give you an understanding of the complexity and amount of interaction involved in the Oracle RDBMS. As you look in detail at the tuning of the server processes and applications later in this book, you can use this overview as a reference to the basics of how the Oracle RDBMS operates. Because of the differences in operating systems, minor variances in different environments will be discussed individually.

**RDBMS Functionality**

If the RDBMS is to operate, you must provide for certain functions, including data integrity, recovery from failure, error handling, and so on. This is accomplished via events such as checkpointing, logging, and archiving. The following sections list and describe some of these functions.

**Checkpointing**

You know that Oracle uses either the CKPT background process or the LGWR process to signal a checkpoint; but what is a checkpoint and why is it necessary?

Because all modifications to data blocks are done on the block buffers, some changes to data in memory are not necessarily reflected in the blocks on disk. Because caching is done using a least recently used algorithm, a buffer that is constantly modified is always marked as recently used and is therefore unlikely to be written by the DBWR. A checkpoint is used to ensure that these buffers are written to disk by forcing all dirty buffers to be written out on a regular basis. This does not mean that all work stops during a checkpoint; the checkpoint process has two methods of operation: the normal checkpoint and the fast checkpoint.

In the normal checkpoint, the DBWR merely writes a few more buffers every time it is active. This type of checkpoint takes much longer but affects the system less than the fast checkpoint. In the fast checkpoint, the DBWR writes a large number of buffers at the request of the checkpoint each time it

is active. This type of checkpoint completes much quicker and is more efficient in terms of I/Os generated, but it has a greater effect on system performance at the time of the checkpoint.

You can use the time between checkpoints to improve instance recovery. Frequent checkpoints reduce the time required to recover in the event of a system failure. A checkpoint automatically occurs at a

log switch.

**Logging and Archiving**

The redo log records all changes made to the Oracle database. The purpose of the redo log is to ensure that in the event of the loss of a datafile as a result of some sort of system failure, the database can be recovered. By restoring the datafiles back to a known good state from backups, the redo log files (including the archive log files) can replay all the transactions to the restored datafile, thus recovering the database to the point of failure.

When a redo log file is filled in normal operation, a log switch occurs and the LGWR process starts writing to a different redo log file. When this switch occurs, the ARCH process copies the filled redo log file to an archive log file. When this archive process has finished copying the entire redo log file to the archive log file, the redo log file is marked as available. It's critical that this archive log file be safely stored because it might be needed for recovery.

**NOTE:** Remember that a transaction has not been committed until the redo log file has been written. Slow I/Os to the redo log files can slow down the entire system.

**What Affects Oracle Performance?**

Because one of the roles of the DBA is to anticipate, find, and fix performance problems, you must know what types of things affect performance. To understand why these things affect performance, you must first review the basics of how a computer system works.

**Overview of Computer Architecture**

Your computer system consists of thousands of individual components that work in harmony to process data. Each of these components has its own job to perform, and each has its own performance characteristics.

The brainpower of the system is the Central Processing Unit (CPU), which processes all the calculations and instructions that run on the computer. The job of the rest of the system is to keep the CPU busy with instructions to process. A well-tuned system runs at maximum performance if the CPU or CPUs are busy 100% of the time.

So how does the system keep the CPUs busy? In general, the system consists of different layers, or tiers, of progressively slower components. Because faster components are typically the most expensive, you must perform a balancing act between speed and cost efficiency.

**CPU and Cache**

**New Term:** The CPU and the CPU's cache are the fastest components of the system. The cache is high-speed memory used to store recently used data and instructions so that it can provide quick access if this data is used again in a short time. Most CPU hardware designs have a cache built into the CPU chip. This internal cache is known as a *Level 1* (or *L1*) *cache*. Typically, an L1 cache is quite small--8-16KB.

When a certain piece of data is wanted, the hardware looks first in the L1 cache. If the data is there, it's processed immediately. If the data is not available in the L1 cache, the hardware looks in the L2 cache, which is external to the CPU chip but located close to it. The L2 cache is connected to the CPU chip(s) on the same side of the memory bus as the CPU. To get to main memory, you must use the memory bus, which affects the speed of the memory access.

Although the L2 cache is twice as slow as the L1 cache, it's usually much larger. Its larger size means you have a better chance of getting a cache hit. Typical L2 caches range in size from 128KB to 4MB.

Slower yet is the speed of the system memory--it's probably five times slower than the L2 cache. The size of system memory can range from 4MB for a small desktop PC to 2-4GB for large server machines. Some supercomputers have even more system memory than that.

As you can see from the timeline shown in Figure 2.4, there is an enormous difference between retrieving data from the L1 cache and retrieving data from the disk. This is why you spend so much time trying to take advantage of the SGA in memory. This is also why hardware vendors spend so much time designing CPU caches and fast memory buses.

**Figure 2.4.**

*Component speed comparison.*

**CPU Design**

Most instruction processing occurs in the CPU. Although certain intelligent devices, such as disk controllers, can process some instructions, the instructions these devices can handle are limited to the control of data moving to and from the devices. The CPU works from the system clock and executes instructions based on clock signals. The clock rate and type of CPU determine how quickly these instructions are executed.

The CPU usually falls into one of two groups of processors: Complex Instruction Set Computer (CISC) or Reduced Instruction Set Computer (RISC).

**CISC Processors**

CISC processors (like the ones Intel builds) are by far the most popular processors. They are more traditional and offer a large instruction set to the program developer. Some of these instructions can be quite complicated; most instructions require several clock cycles to complete.

CISC processors are complex and difficult to build. Because these chips contain millions of internal components, the components are extremely close together. The physical closeness causes problems because there is no room for error. Each year, technology allows more complex and faster chips to be built, but eventually, physics will limit what can be done.

CISC processors carry out a wide range of tasks and can sometimes perform two or more instructions at a time in parallel. CISC processors perform most tasks, such as RDBMS processing, very well.

**RISC Processors**

RISC processors are based on the principle that if you can reduce the number of instructions processed by the CPU, the CPU can be simpler to build and can run faster. By putting fewer internal components inside the chip, the speed of the chip can be accelerated. One of the most popular RISC chips on the market is the DEC Alpha.

The system compiler determines what instructions are executed on the CPU chips. When the number of instructions was reduced, compilers were written to exploit this and to compensate for the missing instructions.

By reducing the instruction set, RISC manufacturers have been able to increase the clock speed to many times that of CISC chips. Although the faster clock speed is beneficial in some cases, it offers little improvement in others. One effect of a faster CPU is that the surrounding components such as L2 cache and memory must also run faster at an increase in cost.

One goal of some RISC manufacturers is to design the chip so that the majority of instructions complete within one clock cycle. Some RISC chips can already do this. But because some operations that require a single instruction for a CISC chip might require many instructions for a RISC chip, a speed-to-speed comparison cannot be made.

**CISC versus RISC**

Both CISC and RISC processors have their advantages and disadvantages; it's up to you to determine whether a RISC processor or a CISC processor will work best for you. When comparing the two types of processors, be sure to look at performance data and not just clock speed. Although the RISC chips have a much faster clock speed, they do less work per instruction. The performance of the system cannot be determined by clock speed alone.

**Multiprocessor Systems**

Multiprocessor systems can provide significant performance with very good value. With such a

system, you can start with one or two processors and add more as needed. Multiprocessors fall into several categories; two of the main types of multiprocessor systems are the Symmetric Multiprocessor (SMP) system and the Massively Parallel Processing (MPP) system.

**SMP Systems**

SMP systems usually consist of a standard computer architecture with two or more CPUs that share the system memory, I/O bus, and disks. The CPUs are called *symmetric* because each processor is identical to any other processor in terms of function. Because the processors share system memory, each processor looks at the same data and the same operating system. In fact, the SMP architecture is sometimes called *tightly coupled* because the CPUs can even share the operating system.

In the typical SMP system, only one copy of the operating system runs. Each processor works independently by taking the next available job. Because the Oracle architecture is based on many processes working independently, you can see great improvement by adding processors.

The SMP system has these advantages:

● It's cost effective--The addition of a CPU or CPU board is much less expensive than adding another entire system.

● It's high performing--Under most applications, additional CPUs provide an incremental performance improvement.

● It's easily upgradable--Simply add a CPU to the system to instantly and significantly increase performance.

A typical SMP system supports between four and eight CPUs. Because the SMP system shares the system bus and memory, only a certain amount of activity can occur before the bandwidth of the bus is saturated. To add more processors, you must go to an MPP architecture.

**MPP Systems**

MPP systems are based on many independent units. Each processor in an MPP system typically has its own resources (such as its own local memory and I/O system). Each processor in an MPP system runs an independent copy of the operating system and its own independent copy of Oracle. An MPP system is sometimes called *loosely coupled*.

Think of an MPP system as a large cluster of independent units that communicate through a high speed interconnect. As with SMP systems, you will eventually hit the bandwidth limitations of the interconnect as you add processors. However, the number of processors with which you hit this limit is typically much larger than with SMP systems.

If you can divide the application among the nodes in the cluster, MPP systems can achieve quite high scalability. Although MPP systems can achieve much higher performance than SMP systems, they are

less economical: MPP systems are typically much higher in cost than SMP systems. **CPU Cache**

Regardless of whether you use a single-processor system, an SMP system, or an MPP system, the basic architecture of the CPUs is similar. In fact, you can find the same Intel processors in both SMP and MPP systems.

As you learned earlier today, the system cache is important to the system. The cache allows quick access to recently used instructions or data. A cache is always used to store and retrieve data more quickly than the next level of storage (the L1 cache is faster than the L2 cache, the L2 cache is faster than main memory, and so on).

By caching frequently used instructions and data, you increase the likelihood of a cache hit. This can save precious clock cycles that would otherwise have been spent retrieving data from memory or disk.

**System Memory Architecture**

The system memory is basically a set of memory chips, either protected or not protected, that stores data and instructions used by the system. System memory can be protected by parity or by a more sophisticated advanced ECC correction method. Data parity will detect an incorrect value in memory and flag it to the system. An advanced ECC correction method will not only detect an incorrect value in memory, but in many cases can correct it. The system memory can range in size from 4MB on a small PC to 4GB on a large SMP server.

Typically, the more memory available to Oracle, the better your performance. Allocation of a large SGA allows Oracle to cache more data, thus speeding access to that data.

**New Term:** System memory is accessed by the CPUs through a high-speed bus that allows large amounts of data and instructions to be quickly moved from the CPU to L2 cache. Data and instructions are typically read from memory in large chunks and put into the cache. Because the CPU expects that memory will be read sequentially, in most cases it will read ahead the data or instruction that it thinks will be needed next. Sometimes this works, so the data that is needed next is already in cache; sometimes the CPU has guessed incorrectly and other data needs to be retrieved. This process of prereading the data is known as *prefetching*.

Depending on the specific implementation of an SMP system, the memory bus might be shared by all system processors; alternatively, each processor might have a private bus to memory.

**Virtual Memory System**

**New Term:** In a virtual memory system, the OS and hardware allow programs and users to use more memory than is actually available in the system hardware. This memory, known as *virtual memory*,

can be mapped to physical memory. Code or data that is being run by the CPU must reside in physical memory. If a program or data that is larger than physical memory is being accessed, the parts of code and data that are not immediately needed by the program can reside in virtual memory, not physical memory. As that bit of code or data is needed, it can be copied into physical memory, and parts no longer needed can be copied to disk. The process of mapping virtual memory onto physical memory by copying the memory to and from disk is called *paging* or *swapping* (depending on the OS architecture).

Both paging and swapping serve the same purpose, but each operates slightly differently from the other. In a swapping system, an entire process is swapped out (moved from memory to disk) or swapped in (moved from disk to memory). In a paging system, the movement of data to and from the secondary storage occurs on a memory page basis; when more memory is needed, one or more pages is paged out (moved from memory to disk) to make room. A memory page is the smallest unit of memory that is used in the operating system. A typical memory page size is 4KB. If data is requested from virtual memory and is not in physical memory, that data is paged in (moved from disk to memory) as needed. The rest of this section uses the term *paging* to describe both paging and swapping.

Suppose you have a computer system with 16MB of physical memory. If you have a program that needs to access 20MB of data, it obviously won't fit in physical memory. In a virtual memory system, the data is read until little memory remains (the OS reserves some for itself), then the OS copies some of the data pages to disk with the paging mechanism. This is usually done using a least recently used algorithm in which the oldest data is moved out. When some memory has been freed, the program can read more data into memory. As far as the program is concerned, all the data is still in memory; in fact, it is--in virtual memory. As the program begins to reread some of the data and manipulate it, different pieces might be paged in (from disk to physical memory) and paged out (from physical memory to disk).

As you can imagine, paging in or out can be time consuming and uses a lot of system resources. This is why I warn you several times in this book to avoid using so much memory that you cause paging or swapping. Access to disk is approximately 50 times slower than access to memory.

**Bus Design**

**New Term:** Simply put, *bus* is a connection path used by the system to move data from one place to another. Buses get complicated when you look at them from a performance perspective: Capacity, or bandwidth, becomes an issue. Over the years, the term *bandwidth*, which was originally used to describe the electronic characteristics of a circuit, has been adopted by computer designers. In this case, *bandwidth* refers to the amount of data that can be transmitted across a bus in a certain time.

Several bus designs have been introduced in the last few years, all with the same goal: increased capacity. As processors, network hardware, disk controllers, and disks become increasingly fast, buses must develop to support the load generated by these devices. Thankfully, as computers have increased in performance, computer designers have improved bus designs to accommodate these changes. The system bus should not be a bottleneck in your system.

**Oracle Resources**

The Oracle DBMS allocates different resources for various different functions, including the allocation of system memory. The memory might be allocated for database caching or for the data dictionary or library cache. The careful balance of this precious resource is very important in tuning the Oracle RDBMS.

As much data as possible must be cached to avoid the additional cost of going to disk. If you allocate a large Oracle data cache, a higher cache-hit rate can be achieved. A high cache-hit rate indicates that a large percentage of requested data is found in the Oracle cache rather than retrieved from disk.

**Application Design**

Application design can affect performance more than any other factor. In most cases, performance can be severely degraded by an application that does not have well-tuned SQL statements or does not use indexes. A good application design can also significantly improve performance. The application is typically the first place to look when you experience system performance problems.

If a database is built with indexes on a certain set of columns but those columns are not specified in the WHERE clause of the SQL statement, the index probably won't be used. It's not enough to create the correct index on tables; you must ensure that the indexes are used.

**TIP:** It's wise to create a specification identifying the tables and indexes in your database. That way, the application developers and the team that creates the database have a crystal-clear document that identifies which columns are indexed. This can help avoid confusion and allow the application code to fully exploit the indexes.

**Oracle Features**

Another way to improve Oracle performance is to enable Oracle performance features. Among the most important of these features (and my personal favorite) is the Oracle Parallel Query option. Other Oracle performance features include partitioned tables and the Oracle index-only table, both new in Oracle8.

**The Oracle Parallel Query Option**

The Oracle Parallel Query option allows parallelism of many different operations, which greatly enhances performance. The Oracle Parallel Query option consists of several different components, including

● Parallel query

● Parallel index creation

● Parallel recovery

● Parallel table creation

● Parallel index tables

**Parallel Query**

The Oracle parallel query allows a single query to be divided into components and run in parallel. Because a query spends much of its time waiting for I/O operations to complete, parallelizing queries can greatly improve performance. In a well-tuned system where I/O is not a problem, parallel queries can run many times faster than normal queries. Statements that can be parallelized include

● Table scans

● Sorts

● Joins

**NOTE:** You might be wondering why parallelizing operations would help

performance; after all, the work must still be done. In a typical Oracle operation (for example, a SELECT statement), the following steps occur:

**1.** Oracle performs some CPU processing to determine what data is needed.

**2.** Oracle submits an I/O request to disk (assuming that the data is not already in the SGA) and then waits for that I/O to complete.

**3.** This operation is repeated until all data is retrieved.

In the case of a parallel query, these steps would be adjusted like so:

**1.** Oracle performs some CPU processing to determine the query operation.

**2.** Different Oracle processes or threads receive their instructions on what data is needed.

**3.** Oracle thread 1 submits an I/O request to disk (if that data is not already in the SGA) and waits for that I/O to complete.

**4.** Oracle thread 2 submits an I/O request to disk (if that data is not already in the SGA) and

waits for that I/O to complete.

**5.** Oracle thread 3 submits an I/O request to disk (if that data is not already in the SGA) and waits for that I/O to complete.

As shown here, that the time-consuming job of retrieving data from disk is duplicated, thus improving performance. This parallelism allows the CPU(s) to be utilized while other threads are waiting for I/Os.

Retrieving data from disk is a slow process compared to the activity of the CPU, and your goal is to keep the CPUs busy. Because a significant part of any Oracle operation involves CPU processing and I/Os, it is possible and desirable to keep the CPUs busy while many I/Os are being processed simultaneously. This is the main goal of the Parallel Query option.

**Parallel Index Creation**

Index creation involves reading from data tables and then writing to the index tables. Because the parallel query allows reading of tables to be accelerated, the index-creation process is sped up. Index creations can be quite time consuming, so this can be a real advantage.

**Parallel Recovery**

Recovery from a system failure can be quite time consuming. During recovery, users must usually wait for the system to come back online, so any improvement in performance is an advantage. Parallel recovery can speed the recovery process by parallelizing the read from the redo log files, and the roll forward and rollback process.

**Parallel Table Creation**

Although the Oracle Parallel Query option does not generally allow table creations to occur, it is often the case when a table is created as a subset of other tables. Data is often reduced from several large tables into a smaller subset, and this parallelism can be beneficial. In such instances, the following statement allows for parallelism:

CREATE TABLE table\_name AS SELECT...

**Oracle Index Tables**

New to Oracle8, the index table allows indexes and tables to be stored together; this saves space and improves performance by reducing disk I/O. If you reduce the number of required disk I/Os, data can be accessed much faster.

**OS Resources**

**New Term:** In most systems, few resources can be allocated in the operating system. Most OS parameters are changed only to allocate sufficient resources to Oracle; additional resources usually do not improve performance. A lack of resources, however, can decrease performance. *OS resources* often refers to system memory or, in the case of UNIX, shared memory. Other OS resources and tunables include network buffers and disk I/O tunables.

**TIP:** Windows NT is fairly self tunable, but there are a few things, relating primarily to configuration, to look out for:

● Remove unnecessary network protocols--Depending on how the system is configured, several network protocols that you do not use might be configured into your system. These extra protocols use CPU and memory resources.

● Configure the protocols you use in order from most-often used to least-often used--This reduces some of the overhead associated with traversing the infrequently used protocols.

● Keep a close eye on paging--Windows NT treats all memory as virtual. The best way to determine whether your system is paging is to watch Pages/Sec in NT's perfmon. If paging occurs, lower the amount of memory allocated to Oracle.

**Hardware**

Several hardware factors can affect your system's performance. These factors include ● Memory capacity

● Number of CPUs

● CPU cache

● Memory-bus bandwidth

● I/O capacity

**Memory Capacity**

Earlier today you saw an overview of how the system hardware operates. Clearly, any operation that must access slower components, such as a disk or network, will slow down processing. Therefore, it is

important that you have sufficient memory in your system.

**New Term:** Most hardware architectures are limited to 4GB of physical memory, but some architectures on the market support much more. These architectures are said to support a *VLM*, or *Very Large Memory*, architecture. Soon it will be possible to support hundreds of gigabytes of physical memory in a system, allowing for very fast RDBMS operations.

System memory is allocated to Oracle and used for database caching, user memory, and the shared pool, which is used for both the data dictionary and the library cache. You must have enough memory for the shared pool because an insufficient shared pool can hurt performance. When the shared pool is satisfied, the more database buffers you can allocate to the DBMS the better. Be careful, though, to avoid starving the PGA memory needed by your processes, and avoid paging at all costs. You can never have too much memory in your system. Anything that can be cached will reduce system I/O, improving performance.

**Number of CPUs**

Oracle typically scales well with additional CPUs. By adding CPUs you can see significant performance improvement with little additional cost. Some factors that determine how much improvement you will see by adding more processors are the CPU cache and memory- bus bandwidth.

**CPU Cache**

A large CPU cache allows more data and executable code to be stored on the local processor than in memory. This reduces the number of times the CPU must access main memory. Whenever the CPU accesses memory, a slowdown occurs while the CPU waits for that data or code to be retrieved. It is especially bad when the memory bus is busy; the CPU waits even longer until the bus becomes free.

**Memory-Bus Bandwidth**

The memory-bus bandwidth determines how quickly data can be transferred between CPU to memory. If the memory bus is busy when data or code is needed, a CPU stalls waiting for the bus to free. This can severely degrade performance in a multiprocessor computer. A fast memory bus can reduce this problem. A large CPU cache can also reduce this problem by allowing more data and code to be cached.

**I/O Capacity**

I/O is typically one of the biggest factors limiting system performance. Because most DBMS operations involve retrieving data from disk, I/O can be a limiting factor if you do not have adequate capacity for your system load. Fortunately, you can usually solve this problem by carefully configuring your system for proper I/O distribution and by having sufficient I/O capacity. Simply having adequate disk space is insufficient; you must also have enough disk drives to support the number of disk I/Os that the system requires.

**Oracle8 New Features**

Oracle8 has introduced many new features, and I would like to focus on a few key features for the Oracle8 DBA:

● Partitioned objects

● Improved parallelism

● New index types

● Enhanced recovery features

**Partitioned Objects**

Partitioned objects allow Oracle objects, such as tables and indexes, to be broken into smaller, more manageable pieces. Partitioning these objects allows many operations that could normally be performed on only a table or an index to be divided into operations on a partition. By dividing these operations, you can often increase the parallelism of those operations, thus improving performance and minimizing system downtime.

Partitions are enabled via the PARTITION BY RANGE parameter of the CREATE TABLE statement. In this manner, ranges of data are assigned to each individual partition like so:

CREATE TABLE emp

(

name CHAR(30),

address CHAR(40),

region INTEGER

)

PARTITION BY RANGE ( region)

(

PARTITION VALUES LESS THAN (10) TABLESPACE tbl0,

PARTITION VALUES LESS THAN (20) TABLESPACE tbl1,

PARTITION VALUES LESS THAN (30) TABLESPACE tbl2

);

This creates a table with partitioning, as shown in Figure 2.5.

**Figure 2.5.**

*Table partitioning.*

Partitioning is recommended for large tables because it makes them much more manageable. Oracle does not currently support partitioning of clusters. By partitioning a table, you can break that large

table into several much smaller pieces. A partitioned table can take advantage of some of the following features:

● Partitioned DML

● Exporting/importing by partition

● Range partitioning

● Local and global indexing

● Parallel loading by partition

**Partitioned DML**

Parallel INSERT, DELETE, and UPDATE operations can occur on a partition basis. Using partitions allows these operations to be conducted either globally or locally within a partition.

**Exporting/Importing by Partition**

Partitioning allows operations such as exports and imports to be performed on a partition basis. This can reduce the time required by some maintenance operations, such as reorganization of data or reclustering. This also allows you to change the physical layout of your database on a partition basis. If you limit the scope of export and import operations, they can benefit from a large degree of parallelism.

**Range Partitioning**

Range partitioning is a method whereby the partitioning of data is done based on the value of the data itself. This allows for tremendous flexibility in distributing data based on ranges of data values. Range partitioning allows you to partition high-volume data separately from low-volume data or to separate current from old data.

**Local and Global Indexing**

**New Term:** A *local index* indexes data that resides in only one partition. A *global index* indexes data that resides on more than one partition. This allows for great flexibility in terms of adding new indexes, reducing index sizes, and allowing for partition independence.

An example of where local indexing might be beneficial is a table where sales records are stored. Using table and index partitioning, you can store data and indexes separately based on calendar months; doing this allows reduced index size and faster index lookups for entries of a particular month. If you partition these entries you can add new months and delete outdated entries without

reindexing the entire table. You could keep 12 months of partitions and indexes online in this manner. **Parallel Loading by Partition**

With a partitioned table, SQL\*Loader can either load an entire table in parallel by partition or simply load a single partition. Either method provides great flexibility.

If you use the conventional path load, the loader automatically distributes the data to the correct partition and updates the local and global indexes. You can also use the loader to load a partitioned table or a partition of a table. Again, indexes are built automatically. It is also possible to direct-load a partition in parallel provided that no global indexes exist, but you must rebuild the local indexes yourself.

**Improved Parallelism**

The arrival of Oracle8 has heralded tremendous improvement in the area of parallelization. In addition to the new parallel features listed previously, some existing parallel operations have been extended.

Parallel recovery has been improved by allowing rollbacks of parallel DML operations that have failed to be performed in parallel. This parallel transaction recovery is supported on transaction and process failures but not during instance recovery.

New parallel hints have been added for parallel insert operations. The APPEND hint tells the optimizer to append the insert data beyond the high water mark of the segment.

**New Index Types**

The index-only table is new in Oracle8. With traditional indexes and tables, data and indexes are stored separately. With an index-only table, the data to which the index refers is stored in the leaf block or lowest level block of the index, so the data and indexes are stored together. Depending on your application, this can be an advantage.

Applications that access data primarily via a key value can see an advantage from the use of index only tables. Because the data is stored within the index, the data is immediately available when the index has reached its lowest level. This can speed data retrieval.

Applications that do not access data primarily via a key value will see no improvement; indeed, performance will likely be degraded in these applications. Any application that involves table scans or requires multiple indexes will not benefit from the index table. The index table is covered in much more detail on Day 13.

**Enhanced Recovery Features**

Oracle has made tremendous improvements in the areas of backup and recovery. Most of these new features revolve around the Recovery Manager. Another recovery feature in Oracle8 is the image copy backup, which can improve recovery time in the event of a failure.

**Recovery Manager**

**New Term:** Recovery Manager is an online utility designed to assist the DBA with all backup and recovery operations. Not only does it perform the backup and recovery, it maintains a database called the *recovery catalog* that stores information about these operations.

**Image Copy Backup**

An image copy backup essentially allows you to copy a datafile to another place on disk or to another disk on your system. In the event of a failure, no recovery is necessary from the image copy; you must simply switch to that backup copy. You must, however, perform a recovery to make that copy current. In the event of a failure, this might be the fastest way to recover.

**NOTE:** Days 16-18 cover backup and recovery techniques in greater detail.

**Oracle Products**

As part of the overview of the Oracle system, I would like to briefly cover the optional available Oracle products. Although many of these products are covered elsewhere in this book, you should at least aware of their existence. The Oracle product line is divided into three areas:

● The Oracle server

● Development tools

● Applications

**The Oracle Server**

The Oracle server is the DBMS itself, and includes many options and features such as the Parallel Query option, network protocols, and advanced system administration options. Some of the key options available to the Oracle server include

● Enterprise Manager--This option is fairly new to Oracle, and consists of the management console and intelligent agents. The management console, which is the core element in Oracle's new graphical administrative package, runs only on Windows NT, but can manage any Oracle server. The console allows the DBA to graphically control one or more Oracle systems. The

console can be used to configure and manage Oracle instances as well as to diagnose problems and can be configured to alert the DBA in the event of a problem. The keys to Enterprise Manager are the intelligent agents, which run on the Oracle server and provide the communication layer necessary for the console to communicate with these systems. The intelligent agents use industry-standard SNMP (Simple Network Management Protocols) to communicate with the console, thus allowing for future expansion.

**NOTE:** Throughout this book, Enterprise Manager is referenced as the primary method for administering the system. Nonetheless, command-line management is also covered.

● ConText--When integrated with any text system, Oracle ConText can analyze, filter, and reduce text for speed reading and summary viewing. Oracle ConText returns detailed assessments of the text it processes, checking for grammatical errors and rating the quality and style of the writing.

● Media Server--Oracle Media Server provides high-performance, scalable, and reliable multimedia library functions on a wide variety of general-purpose systems. Media Server handles the storage, retrieval, and management of movies, music, photographs, and text articles.

● The Spatial Data option--The Oracle Spatial Data option can be used to manage a database that contains spatial data. This option allows for the storage of spatial or geographical data. If you store the spatial data within the database, the complexity of managing the storage is reduced and the performance is increased.

● The Oracle Web server--The Oracle Web server is designed to provide front-end services to allow World Wide Web access to an Oracle database. This product allows Web users to retrieve information directly from an Oracle database rather than from traditional flat files. This product can be used to enhance the performance and functionality of your Web server via the use of indexes and data caching. With the flexibility of the Oracle RDBMS, the functionality of your Web server can be enhanced via the use of language-sensitive context and other features.

● The Internet Commerce server--The Internet Commerce server is a complete set of tools designed to help you create, run, and administer an Oracle system that is used for Web commerce. Because it is based on the proven technology of the Oracle server, the system can provide these services in a robust and secure fashion.

**Development Tools**

One of Oracle's strongest points has been its development tools. Not only are these tools robust and full featured, they are flexible as well. When client/server systems became popular in the early 1990s, the Oracle tools quickly adapted. When HTML and Java applications became popular in the mid-

1990s, the Oracle development tools quickly adapted yet again. The adaptability of these tools guarantees that applications developed with them can be quickly adjusted for new uses and technologies. Oracle provides the following tools:

● Designer/2000--This set of modeling tools reduces some of the pain associated with designing systems. These tools, which help with process and data modeling, can be used to provide input into the Developer/2000 system and to develop the fundamental models that are the foundation for your business processes.

● Developer/2000--This set of tools allows you to create an application and roll it out in Windows, Macintosh, Motif, and character mode. Developer/2000 incorporates graphics and images as well as support for multimedia objects such as video and sound in a variety of standard formats.

● Discoverer/2000--This data-analysis tool supports querying, reporting, and the graphical multidimensional analysis of the data warehouse. Its key features include graphical representation and drill-down features.

● Power Objects--This lightweight, GUI development tool, which is available for Windows, Macintosh, and OS/2, allows the quick development of applications that use relatively small system resources. Power Objects is conceptually similar to Developer/2000, but lacks many of Developer/2000's features.

● Objects for OLE--This set of tools allows you to link OLE-compliant applications to an Oracle RDBMS. This tool provides a quick and easy way to exploit the power of applications such as spreadsheets. Objects for OLE also allows easy linking of database tables into word-processing documents.

● Programmer/2000--This suite of tools helps with the development of SQL, PL/SQL, and stored procedures. These tools can be helpful for application developers.

● Media Objects--Oracle's lightweight tool for developing multimedia applications, Media Objects supports client/server, CD-ROM, and interactive television processes.

● Database Designer--This lightweight version of the Oracle Designer/2000 product can assist in the design and creation of databases. Database Designer, a single-user tool, graphically designs the database tables and generates SQL that can be used to create this database.

**Applications**

Oracle's application software falls into two main categories: traditional applications and newer OLAP (Online Analytical Processing) applications.

**Traditional Oracle Applications**

Oracle's suite of traditional applications is used to perform basic and essential business tasks. These applications are used by many of the world's largest companies. The suite provides support for the following areas:

● Financial

● Human resources

● Project management

● Sales

● Manufacturing

**OLAP Applications**

The OLAP applications provide a graphical interface for DSS and data-warehousing applications. These tools lend a multidimensional model to the database, providing analysis, forecasting, and statistical operations.

**Other Products**

Oracle offers many other products that are not mentioned here. These products handle various tasks such as networking, office automation, workgrouping, and so on. Although these products and services are too numerous to cover here, rest assured that Oracle's full line can handle most (if not all) of your database and communication needs.

**Summary**

Today's lesson presents an overview of the Oracle architecture, including the physical structure (consisting of datafiles, redo log files, and control files) and the Oracle instance (consisting of processes and memory). Next you saw how a computer system works and how it depends on components such as cache memory to improve performance. Finally, you reviewed some of Oracle's products to get an idea of the different areas in which the Oracle server is used. This lesson set the foundation for many of the later lessons in this book. By having an understanding of the inner workings of Oracle, you will be better able to administer the Oracle DBMS.

**What's Next?**

You'll spend tomorrow installing the Oracle8 server. The key to the installation process is understanding what components you are installing and why you are installing them.

**Q&A**

**Q What are the main hardware components that affect performance?**

**A** The main hardware components that affect performance are the speed of the CPU(s), the amount of memory, and the I/O subsystem.

**Q What happens if a failure corrupts the redo log files?**

**A** If redo log files are lost, you cannot recover the database. All changes made since the last backup will be lost. This is why redo log files should be on protected or fault-tolerant disk drives.

**Q What happens if a failure corrupts the datafiles?**

**A** When a datafile is lost, the corrupted file can be restored from a backup. After the datafile is restored, the redo log files and archive log files can reapply any changes made before the time of the failure. No data is lost.

**Q Why does parallelizing a query make things faster?**

**A** Most of the realtime or clock-time processing a query operation is spent waiting for I/Os to complete. Parallelizing a query enables you to keep the CPUs busy while you are waiting.

**Workshop**

This workshop provides quiz questions to help you solidify your understanding of the material covered. Answers to quiz questions can be found in Appendix A, "Answers."

**Quiz**

**1.** What are the three types of files that make up an Oracle database?

**2.** What makes up the Oracle instance?

**3.** Which is faster--memory or disk?

**4.** Name two new features of Oracle8.



© Copyright, Macmillan Computer Publishing. All rights reserved.



**Teach Yourself Oracle 8 In 21 Days**

****

**- Day 3 -**

**Installing Oracle Software**

Today you will see the steps involved in installing Oracle8 software. You'll find numerous examples of installing Oracle8 on Windows NT 4.0, and other operating systems are not significantly different. Throughout this book I will be using Windows NT as the primary OS for examples and instruction, and where necessary, I will point out the differences for other operating systems.

Installing Oracle software is much less complicated than in the old days of Oracle6 and early Oracle7. Oracle has made tremendous progress in streamlining and improving the installation process. It is fairly straightforward to install Oracle8 server, client, and administrator systems. There is also an option to install individual components, which is a little bit more complex. Today you will see all four of these methods.

**What You Need to Know Before You Begin** There is very little you need to know before you begin the installation process:

● What language do you want to use to run Oracle8? A large selection of different languages is available. You must decide which language you want to use before you begin the installation.

● Where do you want the Oracle8 binaries to reside? You will be asked to provide a path to where the binaries will be copied during the installation process.

● What Oracle8 products to you want to install? A number of different products can be installed based on the type of system that you are installing. There are four different types of installations.

The installation type that you choose will be based on the function of the system that you will be installing. Types of installations include the server, client, Programmer/2000, and custom

installations. Each installs a different set of programs.

**Server Installation**

In the server installation, the components necessary for an Oracle8 server system will be installed. These components include

● Oracle8 server--The Oracle8 server is really the guts of the system. This is the Oracle8 RDBMS.

● Server networking components--These include the TNS listener program and the SQL\*Net programs as well as the administrative tools necessary to manage them.

● Oracle Enterprise Manager--This component is the key to Oracle's new administrative system. The Enterprise Manager provides a graphical interface to administer the Oracle8 server.

● Oracle intelligent agents--These are the programs that allow the Oracle Enterprise Manager to communicate with the Oracle8 server and utilities. The Oracle intelligent agents use SMTP (Simple Management Transport Protocol) to allow this communication to occur.

● Oracle8 documentation (optional)--You can install the Oracle8 documentation locally or configure it so that you can access the documentation via CD-ROM.

● Miscellaneous utilities--Other utilities include SQL\*Plus, the Oracle installer, SQL\*Loader, ODBC, and so on.

**Client Installation**

The client installation gives you a choice of the DBA client or the application user clients. The DBA installation gives you the components needed to remotely administer an Oracle system. These products include

● Oracle Enterprise Manager--As mentioned previously, this component is the key to Oracle's new administrative system. The Enterprise Manager provides a graphical interface to administer the Oracle8 server.

● SQL\*Plus--This is the primary interface into Oracle for ad-hoc access. SQL\*Plus can be used for administrative purposes if desired.

● The Oracle8 toolbar--The toolbar provides a quick way to access the Oracle Enterprise Manager applications.

● SQL\*Net--The SQL\*Net components include the client-side networking components and administrative tools.

● Oracle8 documentation (optional).

● Miscellaneous tools and utilities--Items such as OCI, Oracle Installer, and so on are installed.

The application user client installation gives you the components needed to connect to Oracle as an end user. These products include

● SQL\*Plus

● SQL\*Net

● Oracle8 documentation (optional)

**Programmer/2000 Installation**

The Programmer/2000 installation provides the necessary client components to connect to the Oracle8 RDBMS, as well as a choice of the following development packages:

● Oracle Pro\*C/C++--This is the method whereby database object types can be used within C/C++ programs. Pro\*C/C++ is essentially a preprocessor that converts these objects into C/C++ objects.

● Pro\*COBOL--This is Oracle's COBOL interface.

● ODBC--The Open Database Connectivity software and development libraries needed by the application developer.

● OCI--The Oracle Object Call Interface provides a method whereby Oracle can be accessed from a C program.

**Custom Installation**

The custom installation allows any or all products on the Oracle8 installation CD-ROM to be installed. Using the custom installation option, you simply choose which options you want to install.

**The Installation Process**

Now that you've decided where you want the Oracle8 installer to put the Oracle binaries and you've determined which language to install and which installation method to use, it's a simple matter of installing the Oracle products. In this part of the day you will see step-by-step how this installation process works.

This section is divided into four parts, each looking at a different installation type. First, the initial installation process is covered, then the Oracle8 Server, followed by the two types of client installations. Finally, the Developer/2000 and custom installation types are shown. Each is covered in detail, and each of the options is explained.

**Basic Installation**

The initial installation process begins when you place the Oracle8 CD into your CD-ROM drive. If your system has autorun configured, you will be prompted by the Oracle8 installation procedure; from here, decide whether you want to begin installing Oracle8, to browse information about Oracle8, or explore the CD-ROM (see Figure 3.1).

**Figure 3.1.**

*The Oracle8 installation autorun message.*

If you are ready to install Oracle8 at this time, simply click Begin Installation and the Oracle8 installation process will commence. If you are not ready to install, is possible to return at a later time and restart the installation process. Simply use NT Explorer and, from the CD-ROM drive, double click the program setup.exe. This will also invoke the installation program.

After the installation process has begun, you will see the initial install screen (shown in Figure 3.2). This screen asks you which language you want to use to run the Oracle products. Choose OK to select the default language as English, or select a different language.

**Figure 3.2.**

*The initial installation screen.*

Next, you will be asked for some installation settings, as shown in Figure 3.3. These questions relate to your particular installation. The first screen simply asks for the name of your company and for the OS directory where the Oracle8 binaries will be installed. Before you choose a directory, be sure the selected disk volume has adequate disk space; a full installation can require more than 100MB.

**Figure 3.3.**

*Installation settings.*

After you decide on the installation language and location, you might be informed that certain path variables have been adjusted for the Oracle installation process and the Oracle programs to work properly. This is shown in Figure 3.4.

**Figure 3.4.**

*The Oracle path changes.*

If you click the OK button, the path of the Oracle binary files will be added to your environment. Unlike earlier versions of the Oracle installation process, a reboot the system is not required at this time to continue the installation. After you have completed these steps, you will be presented with a screen asking you which type of installation you want to perform, as shown in Figure 3.5.

**Figure 3.5.**

*Select the type of installation you want to perform.*

At this point, the installation will be different based on your answer. These installations are covered separately in the following sections.

**Installing the Oracle8 Server**

The following is split into two sections. The first section describes the actual process of installing the software. The second section describes the system after the installation has occurred. Specifically, this section provides information about what products and services have been installed, and how the Registry has been modified.

**The Installation Process**

If you've chosen to install the Oracle8 server by clicking OK on the menu shown in Figure 3.5, the server installation process will begin. At this point, a few dependencies will be determined and you will be queried as to the Oracle data cartridges that you are licensed to install (see Figure 3.6). Oracle data cartridges are described on Day 21, "Exploring the Web Publishing Assistant and Network Computing Architecture."

**Figure 3.6.**

*Installing additional cartridges.*

A short time into the installation process, you will be queried as to whether you want a typical database installation or whether you want to customize your database with the Database Assistant (see Figure 3.7).

**Figure 3.7.**

*The Starter Database Installation Options dialog box.*

The three selections are

● Typical--The typical starter database is 30MB in size and includes a small set of tables and tablespaces as a starter set. In most situations, this is probably the correct selection.

● Custom--Checking this option invokes the Oracle Database Assistant, which helps you create a custom Oracle database. Choose this option if you will be using this database as a replication system, if you want to install other options, or if you want to choose the name of your database. This option provides the most flexibility. Creating a custom database with the Database Assistant is described in detail on Day 6, "Administering Databases and Datafiles."

● None--If you are planning to immediately build a new database on this system, consider not building a starter database. This will save on space and time.

The final question that you will be asked is the location of the online documentation. You can either access the documentation via CD-ROM or install it on your local hard drive, as shown in Figure 3.8.

**Figure 3.8.**

*Installing the documentation.*

Installing the documentation on the hard disk requires approximately 66MB of disk space, but it's well worth the space (if available). There is nothing worse than not having the documentation immediately available when you need it. The Oracle8 documentation set is available in HTML format with hyperlinks, which makes it very convenient.

As the installation proceeds, you can watch its progress on the status bar. As the installation gets further along, the bar on the bottom increases in length. When the installation is complete, you will be so notified.

**What Has Been Installed?**

The installation of the Oracle8 server has instated the following components: ● The Oracle8 RDBMS

● Oracle Enterprise Manager

● Oracle intelligent agents

● The Oracle Administrators toolbar

● Online documentation

● Networking components

● Oracle utilities, including SQL\*Plus

A few specifics should be mentioned here. The Oracle8 installation processes has registered Oracle as a service. Specifically, seven services have been registered (see Figure 3.9):

● OracleAgent--This service is the Oracle intelligent agent for the Oracle Enterprise Manager. This agent can communicate with multiple Oracle instances on the same system; therefore, only one agent service exists.

● OracleClientCache80--This service is manual. Once started, it stores in cache all information received from an Oracle names server.

● OracleNamesService80--Use this service, which is also manual, if you configure this system as a names server.

● OracleServiceORCL--This service is associated with the instance for Oracle SID=ORCL (default). This is not the actual Oracle instance, but a bootstrap instance. The service starting up does not start up the Oracle instance at boot time.

● If new databases are created with a different SID, a new service must be created with the new SID in the name. This is described in detail on Day 5, "Managing the Oracle RDBMS."

● OracleStartORCL--The OracleStartORCL service starts the Oracle service named ORCL (default) each time the system is rebooted.

● OracleTNSListener30--This is the TNS listener service. There is a TNS listener service for each network listener that you define. Typically, there is one listener per defined protocol.

● OracleWebAssistant. This service enables queries from the Oracle database to be published to a Web page.

**Figure 3.9.**

*These are the installed Oracle services.*

If you decide you don't want a particular service to automatically start each time the system is booted, you can change this by modifying the Oracle service properties (see Figure 3.10). By changing the service startup from automatic to manual, the service will only start manually (you must modify the service through NT).

**Figure 3.10.**

*Use this dialog to modify the Oracle service properties.*

**TIP:** This can be useful on a development machine where you do not always need the ability to start up Oracle and want to save on system resources.

Also as part of the installation process, new program groups have been created and appear as part of the taskbar. The first of these groups is the Enterprise Manager group. This group contains icons for the following Oracle products:

● Enterprise Manager

● Backup Manager

● Instance Manager

● Network Topology Generator

● Schema Manager

● Security Manager

● Storage Manager

The second of these program groups is the Oracle for NT group. This group contains icons for the following Oracle products:

● Oracle ODBC Administrator

● Oracle Instance Manager

● Oracle Installer

● Oracle Network Configuration wizard

● SQL\*Plus

The third program group created is the Oracle Replication Manager program group. One other feature of the Oracle Server installation is the Oracle Administrators toolbar. This toolbar is installed as part of the server installation as well as the administrators installation, and is shown in Figure 3.11.

**Figure 3.11.**

*The Oracle Administrators toolbar.*

This toolbar allows instant access to many of the applications that are part of the Oracle Enterprise Manager system.

**Installing the Oracle8 Client Products**

The following is split into two sections. The first section describes the actual process of installing the software. The second section provides a description of the system after the installation has occurred. Specifically, this section provides information on what products have been installed, what services have been installed, and how the Registry has been modified.

**The Installation Process**

You can choose to install the Oracle8 client products by clicking the Oracle8 Client Products button and then clicking the OK button. The client installation process immediately asks whether you want to proceed with the database administrator or application user installation. To install the database administrator client, click OK (see Figure 3.12).

**Figure 3.12.**

*Installing the database administrator client.*

Oracle immediately begins installing the database administrator client components. As with the server installation, you will be asked whether you want to install the Oracle documentation on hard disk or whether you want to access the documentation via CD-ROM.

If your installation is for an end-user system, you will choose the application user installation. To install the application user client, click the Application User button, then click OK.

Oracle immediately begins installing the application user client components. As with the server installation, you will be asked whether you want to install the Oracle documentation on hard disk or whether you want to access the documentation via CD-ROM.

**What Has Been Installed?**

The database administrator client installation instates the following components: ● Oracle Enterprise Manager

● The Oracle Administrators toolbar

● Online documentation

● Networking components

● Oracle utilities, including SQL\*Plus

● OCI

The application user client installation instates the following components:

● Networking components

● SQL\*Plus

● Online documentation

At the end of today's lesson is an in-depth description of each of the products installed and how they are used.

**Installing the Programmer/2000 System**

The following is split into two sections. The first section describes the actual process of installing the software. The second section provides a description of the system after the installation has occurred. Specifically, this section provides information on what products and services have been installed, and how the Registry has been modified.

**Installing the Oracle8 Custom Configuration**

The following is split into two sections. The first section describes the actual process of installing the software. The second section provides a description of the system after the installation has occurred. Specifically, this section provides information on what products and services have been installed, and how the Registry has been modified.

**The Installation Process**

You can choose to install the custom installation path by clicking the Custom button and then clicking OK. This begins the custom installation procedure. The custom system is for those users who do not fit into the other catagories. You will be presented with the Software Asset Manager screen, where you select any components available on the Oracle8 CD-ROM (see Figure 3.13).

**Figure 3.13.**

*The Software Asset Manager screen.*

Simply choose any components you want by either selecting an entire group (Shift+click) or clicking a number of individual components (Ctrl+click) and then clicking the Install button. These components will be installed as requested.

If you choose the Oracle Documentation option, you will be asked whether you want to install the Oracle documentation on hard disk or whether you want to access the documentation via CD-ROM.

**WARNING:** The custom installation procedure should only be used after you've used the installation procedure several times, and only if you need specific components. It is much better to run both server and Programmer/2000 if you need components of both.

**What Has Been Installed?**

With the custom installation procedure, you will have installed whatever components you requested. The installation procedure will check for dependencies and some additional components might have been installed.

I do not recommend using the custom installation procedure in most cases. The other installation methods are usually perfectly adequate. In the next section, you will see how to remove unneeded components.

**Removing Components**

To remove any unnecessary or unwanted components, simply invoke the Oracle Installer through the Oracle for Windows NT group. Select any components you want to remove, as shown in Figure 3.14, and then click the Remove button. These components will be removed from the system.

**Figure 3.14.**

*The Software Asset Manager screen with components selected for removal.*

When you finish removing components, simply click the Exit button to exit the Oracle Installer. **Product Descriptions**

The various products and components installed based on your selections were shown pre-viously. Following is a brief description of what these products and components actually are.

**Oracle8 Server**

The Oracle8 server is the RDBMS itself. Within the RDBMS package are all the standard Oracle features, including the following options:

● Distributed--This allows the database to be distributed across different systems.

● Replication--This allows for the replication or copying of data to multiple different systems. This can be used for the performance increase provided by having local copies, or for protection in the event of a system failure.

● Parallel Query--This option is my favorite, and allows for several different types of operations to be split into many individual operations, thus improving performance of long-running tasks.

● PL/SQL--The Oracle procedural language option allows for procedural SQL commands to be written.

**Enterprise Manager**

Oracle Enterprise Manager is the graphical administration tool recently introduced by Oracle. The Enterprise Manager helps the DBA by simplifying some of the necessary tasks. The Enterprise Manager comes with a suite of tools, including

● Backup Manager--This option assists the administrator in backup and recovery operations. ● Instance Manager--This handles startup, shutdown, configuration, and so on. ● Schema Manager--This is used to view and modify tables, views, clusters, and so on.

● Security Manager--This is used to manage the user community. It allows you to add users, assign roles, and so on.

● Storage Manager--This is used to view and modify tablespaces, datafiles, and so on. **Intelligent Agents**

Oracle intelligent agents are the programs that allow the Oracle Enterprise Manager to communicate with the Oracle8 server and utilities. The Oracle intelligent agents use SMTP (Simple Management Transport Protocol) to allow this communication to occur. The Oracle intelligent agents are implemented as a service under Windows NT and are started and stopped by the NT Service Manager.

**Oracle Installer**

The Oracle Installer is the application that is used to install and remove Oracle software. The Installer itself is installed on the system and can be used to install and remove additional components even after the initial installation.

**Networking Components**

The Oracle8 networking components consist of the programs and utilities necessary to connect to the Oracle8 server via a network. The networking components consist of the network server and network protocol adapters. These protocol adapters support various network protocols, such as

● TCP/IP

● SPX/IPX

● Named pipes

● DECNet

● Various other networking protocols, depending on your operating system

The networking components also include the administrative tools necessary to administer the Oracle networking components.

**SQL\*Plus**

SQL\*Plus is the ad-hoc user interface tool for the Oracle RDBMS. With SQL\*Plus, you can connect into the RDBMS and run SQL commands and PL/SQL programs. This is the primary nonapplication interface into the Oracle RDBMS.

**Utilities**

These are miscellaneous utilities that are used to administer and modify the Oracle RDBMS. These Oracle utilities include

● SQL\*Loader--This is used to load the database from input files. There are various methods of using the SQL\*Loader, which are covered on Day 9, "Managing Data."

● Export--This program stores Oracle data and table definitions externally in an Oracle-specific binary format. Export is covered in depth on Day 9.

● Import--This program loads data into the database from an Oracle export file. **Server Manager**

Server Manager is a character-based utility that can be used to perform administrative tasks on the Oracle RDBMS. Server Manager provides an interface for performing ad-hoc or scripted administrative tasks.

**Instance Manager**

The Oracle Instance Manager is part of the Enterprise Manager toolset and is used to graphically perform tasks such as starting and stopping the Oracle instance.

**Recovery Manager**

Recovery Manager is used to back up and restore the Oracle database. Recovery Manager not only performs the backup but also maintains backup and recovery information.

**Summary**

Today you learned how to install Oracle8 software. The various types of installations include ● Server installation

● Administrative client installation

● Application user client installation

● Programmer/2000 installation

● Custom installation

You were also presented with an overview of what was installed. As you can see, the new Oracle installation procedure is quite straightforward in most cases. This is quite an improvement from the Oracle6 and early Oracle7 installation procedures.

**What's Next?**

Tomorrow you will see how to properly size an Oracle system and how to plan for future growth. This is a very exciting area of Oracle administration and can be extremely important to the operation of the system. By anticipating growth and determining when your system will run out of resources, you can plan ahead, thus avoiding costly downtime and quick hardware purchases.

**Q&A**

**Q What is the Oracle Enterprise Manager?**

**A** The Oracle Enterprise Manager is the new graphical administration tool designed to help the DBA manage one or more Oracle systems.

**Q What is SQL\*Net?**

**A** SQL\*Net is Oracle's communication protocol. SQL\*Net uses various network communication protocols such as TCP/IP, DECNet, and SPX/IPX, and provides a common programming layer for the Oracle developer.

**Q What kind of user needs the Programmer/2000 installation?**

**A** A software developer would use the development tools that come with the Programmer/2000 installation.

**Q What is the Recovery Manager used for?**

**A** The Recovery Manager is used for backup and recovery. This product performs these operations and maintains a catalog of previous backups.

**Workshop**

The workshop provides quiz questions to help you solidify your understanding of the material covered. See Appendix A, "Answers," for answers to the quiz questions you see here.

**Quiz**

**1.** What is the main component of an Oracle8 server?

**2.** What do the intelligent agents do?

**3.** Name three Oracle utilities.

**4.** Name several network protocols supported by Oracle SQL\*Net.



© Copyright, Macmillan Computer Publishing. All rights reserved.



**Teach Yourself Oracle 8 In 21 Days**

****

**- Day 4 -**

**Properly Sizing Your Database and Planning for Growth**

At this time, per contractual arrangement with the co-author, we can not publish this chapter on our Web site. Please navigate to the next chapter.



© Copyright, Macmillan Computer Publishing. All rights reserved.



**Teach Yourself Oracle 8 In 21 Days**

****

**- Day 5 -**

**Managing the Oracle RDBMS**

Today you will look at the basics. Here you will see some of the objects that must be set up by the DBA (including the Oracle DBA login account, the DBA roles, and the Oracle SID) as well as some of the tasks that the DBA must perform, including

● Setting up Enterprise Manager

● Using Enterprise Manager

● Using Server Manager

● Starting up and shutting down the Oracle instance

All these are basic functions that need to be covered before you move on to topics such as creating a database. This day lays a foundation for many of the days to come. It is important that you completely understand these concepts.

**The Oracle DBA**

On Day 1, "Starting Out with Oracle," you were presented with a list of the duties and responsibilities of the Oracle DBA. Let's refresh your memory with some of the key duties and responsibilities:

● Installing and upgrading Oracle products

● Installing and maintaining your company's applications

● Creating databases, tablespaces, tables, views, and indexes to the specification of the application developers

● Creating user accounts and monitoring system security

● Monitoring space used in the database and planning for future growth

● Monitoring system performance and making changes as necessary to maintain the required performance levels

● Maintaining the integrity of the data in the database

● Planning and implementing a sound backup and recovery strategy

● All these duties are part of being an Oracle DBA.

**The DBA Account**

To accomplish these tasks, the DBA must be given special privileges. These privileges allow the DBA to run commands that other Oracle users are not allowed to perform. These privileges are maintained within Oracle itself. As part of the installation of the Oracle RDBMS, several accounts are created with these special privileges. These accounts and their privileges are described here.

**INTERNAL**

The INTERNAL account is provided mainly for backward compatibility with earlier versions of Oracle, but is still used for key functions such as starting up and shutting down the instance. The INTERNAL account appears as user SYS if you look at the connected sessions, but the INTERNAL account has additional key features: It can start up or shut down the instance. The INTERNAL account is available even when a database has not been created and when no instances are started up.

**SYS**

The SYS account is automatically created whenever a database is created. This account is used primarily to administer the data dictionary. This account is granted the DBA role, as well as CONNECT and RESOURCE roles.

**SYSTEM**

The SYSTEM account is also automatically created whenever a database is created. This account is used primarily to create tables and views important to the operation of the RDBMS. This account has been granted the DBA role.

**Administrative Users**

I recommend that you create individual user accounts and grant the DBA role to those users who will be acting as the DBA. In this way, fewer people access the same account, thus avoiding confusion. Also, if auditing is enabled, there is a record of who made these system changes.

**TIP:** Avoid using the default administrative accounts. If DBAs are authorized with the proper roles, they can perform the tasks they need to do and maintain their individual accounts. This allows you to determine which DBA or DBAs modified the system and who is currently active on it.

**The DBA Roles**

Several roles are available and are assigned to the DBAs. As you will see on Day 10, "Administering User Accounts," these roles are sets of privileges assigned to a particular Oracle role. Each role can then be assigned to a user, thus giving that user all the privileges needed for that particular task. The use of roles is covered in detail on Day 10, but it is appropriate to go over the roles and privileges assigned to the Oracle DBA here.

**DBA**

The DBA role consists of most of the other Oracle roles and privileges. By assigning the DBA role to a user, there is virtually no task that user cannot do. This role should be assigned to trusted users who are active DBAs for this system.

**OSOPER**

The OSOPER role is one of two special operating system roles. These roles are assigned to special accounts that need OS authentication. It is necessary to have OS authentication for some accounts because Oracle authentication can be done only when the database is open. If the database is shut down, Oracle cannot validate the user permissions.

The OSOPER role allows the user to perform the following operations:

● STARTUP and SHUTDOWN

● ALTER DATABASE MOUNT

● ALTER DATABASE OPEN

● ALTER DATABASE BACKUP

● ALTER DATABASE RECOVER

● ALTER DATABASE ARCHIVE LO

After the database is up and running, other users can be authenticated through Oracle security. These operations require a special authentication method because the database is not available.

**OSDBA**

The OSDBA role includes the permissions granted to the OSOPER role with some additional permissions. These additional permissions include the CREATE DATABASE command and all system privileges with the ADMIN OPTION. The ADMIN OPTION allows the user to grant these permissions to other roles or users. Without the ADMIN OPTION, you cannot propagate these permissions and roles.

**Administrator Authentication**

Authenticating the user can be done either through OS authentication using OS accounts and groups or through the use of Oracle password files. Which of these is right for your installation is up to you. These methods are covered in detail on Day 10; until then, all examples are done using Oracle password files.

**The SID**

As you learned on Day 2, "Exploring the Oracle Architecture," an Oracle instance is an Oracle database, the Oracle processes or threads, and the memory it uses. The instance is the logical term that refers to the components necessary to access the data in the database.

Each Oracle instance is identified by a SID (system identifier), which uniquely identifies this instance and is used by the Oracle utilities and networking components to connect you to the correct instance.

A SID is up to four alphanumeric characters in length and is required in order to connect to an Oracle instance. The SID is set by the ORACLE\_SID environment variable. From the NT command line, you can set the SID with the following syntax:

Set ORACLE\_SID=ORCL

The default SID value is set in the NT Registry. If you do not set the SID using the environment variable, the Registry entry is used. The TNSNAMES.ORA file also resolves the SID with a service name. As you will see tomorrow, when you create a network entry using a utility such as the Oracle Network Configuration wizard, you will be prompted for a service name that you pick, a network type and address, and an Oracle SID. This service name is used to resolve both the network and the SID. Even the Oracle service name includes the SID as an identifier.

You will be seeing more of the use of the Oracle SID in the next few days; for now, it is enough to think of the SID as a unique identifier and a way to connect to an Oracle instance.

**The Oracle Enterprise Manager**

The Oracle Enterprise Manager is a new tool from Oracle that allows the DBA to graphically administer one or more Oracle instances. By allowing many operations to be performed graphically, the presentation of data can be simplified and more meaningful.

Enterprise Manager allows the administrator to manage one or more Oracle instances either locally or via the network. Enterprise Manager consists of two main components: the graphical console and the intelligent agents.

The Enterprise Manager console is the graphical tool that allows you to graphically administer the Oracle instances. This console communicates to the various systems it administers via the intelligent agents that run on these systems. These intelligent agents allow the console to communicate with the instances. The agents use the SNMP (Simple Network Management Protocol) to take requests from the console and communicate those requests to the Oracle system running on these systems.

Enterprise Manager allows the DBA to perform the following tasks from a central location: ● Tune and administer one or more Oracle databases.

● Distribute software to both clients and servers.

● Monitor events from multiple instances.

● Perform backup and recovery operations from a single location.

● Perform standard DBA tasks such as user administration.

The Oracle Enterprise Manager is a very powerful and flexible tool that can help in many of your daily DBA duties.

**NOTE:** In many cases I prefer to graphically administer the Oracle system, but in some cases, I still prefer the character-based commands that can be run via the Oracle Server Manager. Throughout this book, both the graphical and character-based administrative methods are shown. I will point out where I feel one tool is more appropriate than another and why, but it is your preference that is important. You will have to decide which tool or set of tools is right for you.

In this section, you will learn how to configure Enterprise Manager and how to invoke its major

functions. The individual tools that comprise Enterprise Manager, such as the Instance Manager, the Storage Manager, the Schema Manager, and so on, are presented separately in the lesson where most appropriate.

**Configuring Enterprise Manager**

The Oracle Enterprise Manager is installed as part of the Oracle server installation process or can be installed as part of the administrator client installation. After Enterprise Manager is installed, you can connect to it by using the NT toolbar and selecting Enterprise Manager from Enterprise Manager program group.

If this is the first time you've invoked Enterprise Manager, you will probably see a screen indicating that the repositories for Enterprise Manager and Software Manager are not installed (see Figure 5.1).

**Figure 5.1.**

*Enterprise Manager and Software Manager repositories are not installed.*

In this case, click OK and allow Enterprise Manager to create the repositories. This operation will take some time. During this period you will see indications that the repositories are being created, as shown in Figure 5.2.

**Figure 5.2.**

*The Oracle Repository Manager screen.*

When the creation operation is complete, Enterprise Manager will start; you will see the four default panes shown in Figure 5.3.

**Figure 5.3.**

*The Enterprise Manager.*

The four panes displayed are

● The Navigator pane--This pane provides a tree-type listing of the various objects available to Enterprise Manager. Through this list, you can launch various administrative operations (described later today).

● The Map pane--Though blank at startup, the Map pane allows you to create a geographical representation of the systems to be administered. This graphical display allows you to drill down into various sites and administer systems worldwide.

● The Job pane--This pane allows you to view and administer various jobs that are running in the system. Through this pane, you can schedule jobs to run on various nodes in the system at various times. This provides you with a way of scheduling routine operations from a single console.

The Event pane--This pane is used to view system events that occur on any node that you are administering from this console. *Events* are occurrences that trigger some kind of action. This action can be a simple alert or can be some type of action.

**NOTE:** When Enterprise Manager is invoked, the Administrator toolbar is also invoked (as seen in the center of the Enterprise Manager). This toolbar allows quick access to the Enterprise Manager utilities. Some administrators like to move the toolbar or remove it altogether. This is up to you. I like to use it on occasion, but usually I remove it.

**Using the Enterprise Manager**

The Enterprise Manager is distinguished from Enterprise Manager applications in this book in that the applications are presented in the section that applies to that application's function. For example, the Schema Manager is covered in the chapter that covers the Oracle schema, the Backup Manager is covered in the chapters covering backup and recovery, and so on. Today's focus is on configuring Enterprise Manager and using the functions associated with the Navigator, Map, Job, and Event panes.

**General**

A few general setup parameters can be modified with Enterprise Manager. These pertain primarily to how Enterprise Manager looks and acts.

**The View Menu**

The View drop-down menu can be used to modify the display. Select the View menu as shown in Figure 5.4.

**Figure 5.4.**

*The View menu.*

From here you can uncheck the various panes that you do not want to view. For example, if you unselect the Show Map Pane button, the Map pane will be removed from the screen.

**The Navigator Menu**

The Navigator drop-down menu can be used to invoke the Discover New Services wizard. To access the wizard, select Navigator | Discovery, as shown in Figure 5.5.

**Figure 5.5.**

*The Navigator menu.*

Depending on whether a database is selected in the Navigator pane, you will see several available options. One of the options is for the Discover New Services wizard. This wizard finds the available services on specified systems. To use this wizard for this purpose, do the following:

**1.** Select Discover New Services Wizard, and you will see the first screen of the Discover New Services wizard, as shown in Figure 5.6.

**Figure 5.6.**

*The first screen of the Discover New Services wizard.*

**2.** Click Next to move to the next screen. This screen allows you to specify node names for the discovery process to run on. After the discovery process has run, Enterprise Manager can communicate with that node and manage the various instances available on that node.

**3.** Type the node name, then click Add. This node will then be ready to be discovered when you click the Finish button (see Figure 5.7).

**4.** After you click the Next button, you are asked for the time interval at which you want discovery to occur. You can retrieve this information immediately or on a regular basis.

**5.** Finally, you are presented with a summary of your choices. If you are satisified that everything is correct, you can proceed with the discovery by clicking Finish.

**Figure 5.7.**

*The Discover New Services wizard Add Nodes screen allows you to add nodes to be discovered.*

**6.** After you click the Finish button, the Discover New Services wizard proceeds to discover that node. When it has completed, you will see the discovered status in the Service Discovery Status screen, shown in Figure 5.8.

**Figure 5.8.**

*The Service Discovery Status screen shows you the discovery process in action.*

**NOTE:** Using the Discover New Services wizard is the best way to configure the Enterprise network topology. Enterprise Manager uses a combination of the discovery feature, the Oracle intelligent agents, and the TOPOLOGY.ORA file described later today.

**File Menu**

The File menu is important to the configuration in that it is where the user preferences setup is found. When you select the user preferences setup, you will see a screen that displays the services found in the discovery process (see Figure 5.9).

**Figure 5.9.**

*The User Preferences screen.*

The User Preferences screen allows you to configure the username, password, and role assigned to the connection that the Enterprise Manager will use to that service. This allows you to keep separate passwords and DBA accounts for each system on your network, but still administer all of them from a common console.

Setting the user preferences now will save you a lot of time and aggravation later. It will allow you to connect directly to these services without having to go through the entire login procedure.

**Using the Navigator Pane**

The Navigator pane is probably where you will do most of your work. As I mentioned, the Navigator pane provides a tree-like presentation with the following top-level branches:

● Databases--This branch shows all the databases known by Enterprise Manager (either by discovery or with the TOPOLOGY.ORA file).

● Groups--These allow you to arrange objects with similar functions together, thus allowing you to administer these objects together.

● Listeners--The known listeners to which Enterprise Manager can connect.

● Nameservers--The nameservers of which Enterprise Manager is aware.

● Nodes--The nodes known to Enterprise Manager.

● Parallel servers--The parallel-server systems known to Enterprise Manager.

An example of the Navigator pane with the first-level trees expanded is shown in Figure 5.10. Note that all the other panes except the Navigator pane are closed in this figure. Most of the objects here are the default database objects from the installation procedure.

**Figure 5.10.**

*The Navigator pane.*

**Using the Map Pane**

The Map pane is designed to facilitate the administration of systems located worldwide. The Map pane allows you to create groups of systems that can be administered by drilling down on the map. After Enterprise Manager groups are set up, you can drill down into a group by simply clicking the map location.

To create a map, perform the following steps:

**1.** Select the Map pane from the View | Map Pane menu.

**2.** Select Map | Create Map.

**3.** At this point, you will see the Create a New Map screen, shown in Figure 5.11. Fill in the name of the map, choose a bitmap file for the map, and click OK.

After the map is created, you will see a picture of the map (see Figure 5.12). By creating a group, you can then move the icon of the group to the location on the map where those systems reside.

The map can be very useful if you are administering a large number of systems that are geographically disparate. You can even draw your own graphics that depict a building or a floor in a building where these systems reside. Take some time and play around with the Map pane. Most of the features are fairly self-explanatory and easy to use.

**Figure 5.11.**

*Creating a map.*

**Figure 5.12.**

*The map.*

**TIP:** Play around with the Map pane. See what kind of configurations you can put

together. Using the Navigator pane, you can drag and drop databases into the groups you have created, and they will show up in the Map pane.

**Using the Job Pane**

The Job pane allows you to create and monitor jobs you have scheduled for one or more systems. This pane can be very useful in scheduling regular activities such as coalescing tablespaces, gathering statistics, or any other type of job that you would like to schedule.

The Job pane allows you to schedule all types of jobs, including

● SQL scripts

● SQL commands

● DBA commands, such as DDL statements

● OS commands and scripts

● Administrative tasks

● Software distribution

The Job pane is described in much more detail on Day 15, "Managing Job Queues and Using Oracle Auditing."

**Using the Event Pane**

The Event pane is used for monitoring events anywhere under the administration of Enterprise Manager. Enterprise Manager uses SNMP (Simple Network Management Protocol) to allow the intelligent agents to signal the console if an event has occurred.

Enterprise Manager allows you to configure the system to monitor whatever you want, and to alert you if anything it is monitoring has passed a threshold. Enterprise Manager can be configured to alert you via the console itself, e-mail, pager, and so on.

This allows you to set up Enterprise Manager to monitor your installation even when you are not there.

**Using Server Manager**

Server Manager provides a character-based interface into the Oracle instance. Invoke Server Manager