

Simplifying Oracle Database Administration

*written by
John Pocknell, Bert Scalzo and Mike Ault
Quest Software, Inc.*



© Copyright Quest® Software, Inc. 2007. All rights reserved.

This guide contains proprietary information, which is protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software, Inc.

WARRANTY

The information contained in this document is subject to change without notice. Quest Software makes no warranty of any kind with respect to this information. QUEST SOFTWARE SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTY OF THE MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Quest Software shall not be liable for any direct, indirect, incidental, consequential, or other damage alleged in connection with the furnishing or use of this information.

TRADEMARKS

All trademarks and registered trademarks used in this guide are property of their respective owners.

World Headquarters
5 Polaris Way
Aliso Viejo, CA 92656
www.quest.com
e-mail: info@quest.com

Please refer to our Web site for regional and international office information.

Updated—September 24, 2007

CONTENTS

- INTRODUCTION 1**
- NEW APPLICATION DESIGN REVIEW 3**
- DATABASE ADMINISTRATION METHODOLOGY 7**
 - DETECT..... 8
 - DIAGNOSE 13
 - RESOLVE 15
 - INFORM..... 16
 - ADMINISTER..... 17
- PRODUCTION VALIDATION 19**
- BENCHMARKS..... 20**
- I'M A DBA, NOT A QC ENGINEER 21**
- CONCLUSION 22**
- REFERENCES 23**
- ABOUT THE AUTHORS 24**
- ABOUT QUEST SOFTWARE, INC. 25**
 - CONTACTING QUEST SOFTWARE..... 25
 - CONTACTING QUEST SUPPORT..... 25

INTRODUCTION

There's no doubt about it, Oracle DBAs have a tough job to do. They probably have a background in application development/design or system administration and quickly have established themselves as resident experts on everything Oracle as well as a host of other things, such as networking, system administration and data modeling.

There are many different types of DBAs:

Production — Must keep everything running at peak performance; best practices matter; has the need-to-work-smarter-not-harder attitude

Application — Supports a business-critical application, such as PeopleSoft or Oracle Applications; understands how the application interacts with the Oracle database

Development — Usually a developer by trade; understands PL/SQL; will only do what's absolutely necessary to keep a development instance running

In addition to their regular duties, DBAs are called upon by the application development team frequently to advise on such things as SQL performance and user access rights, which means they have less time to perform what should be essential work.

Most DBAs find themselves in reactive mode—especially if managing production instances—and have little opportunity to plan ahead or be proactive. To that end, establishing, using and maintaining proven best-practice processes is essential. This is especially important for the tasks that support the business, such as ongoing performance, security and integrity of the production systems.

Let's analyze the distribution of time expended by a typical Oracle DBA during a mere 45-hour workweek:

Task	Hours Per Week	Percent of Total
Fire fighting	10	22
Monitoring	15	33
Performance Tuning/ Change Management	5	12
Application Development Support	15	33
Planning for Growth	0	0
Backup/Recovery Planning	0	0

Source: James F Koopmann¹

Simplifying Oracle Database Administration

We can see from this table that essential tasks, such as planning for growth, and backup and recovery planning, are rarely carried out—oftentimes because they can be put off until tomorrow.

We also can see that there is about triple the time spent monitoring as is spent on performance tuning and change management. This is indicative of a need to better understand how data is being accessed in the database, both from a data structure and SQL statement execution point of view. DBAs actually spend a lot of time supporting application development, which is good, but they should work intelligently to help reduce development bugs and bottlenecks at this stage, in order to prevent performance or data integrity problems down the road.

This White Paper will help Oracle DBAs make the right choices when evaluating tools to simplify database administration. Many DBA tasks are repeatable, but if that knowledge is held only amongst a few people and none of the tasks are formally documented, then as people move departments or leave the organization, that knowledge is lost. This results in a mad search for the scripts or having to rewrite them.

Ultimately, it is this exercise that will influence the degree to which DBAs are successful in streamlining their operations. In order to become more proactive, DBAs must automate certain tasks by choosing tools that are task-oriented, to some extent, to reduce the amount of time it takes to resolve problems. DBAs also should analyze whether their tools cover the range of tasks that they are expected to perform, such as data modeling, administration, schema management, reporting, test environment creation, tuning and preproduction scalability testing.

NEW APPLICATION DESIGN REVIEW

DBAs are asked often to participate in the assessment of data model requirements for a new application to ensure that it meets business needs. The DBA also has a vested interest in this process since he/she will have to manage the database when the application goes live. If the schema is not designed correctly, SQL performance problems may ensue.

There are various stages of the data modeling process during which the DBA may be involved:

- **Initial review** — Various stakeholders, including management, support, end users and data administrators discuss the basic purpose and scope of the proposed new system
- **Logical data model review** — Enforcement of business rules, naming conventions, datatypes, default values, domain usage, constraints and more
- **Performance review** — Assessment of table structure, normalization versus denormalization, transaction design and SQL statement construction
- **Test environment set-up** — Assists the developers to understand what is required to set up and maintain a successful test environment, scalability testing and more; monitors the application during this phase to anticipate underlying problems
- **Preproduction rollout** — Migration scripts, data loading, backup and recovery
- **Postproduction rollout** — Actual and predicted performance comparison, and plan and index needs analysis

Through involvement in the modeling process, the DBA will demonstrate expertise in logical and physical data modeling techniques while assisting development. There are many modeling tools on the market, including well-known enterprise-class tools that may already be acquired by the business and in use by business process modelers, data architects and others.

In the case where DBAs do not have access to or require enterprise-level modeling solutions, they may be drawn to a straightforward, low-cost modeling tool for their desktops.

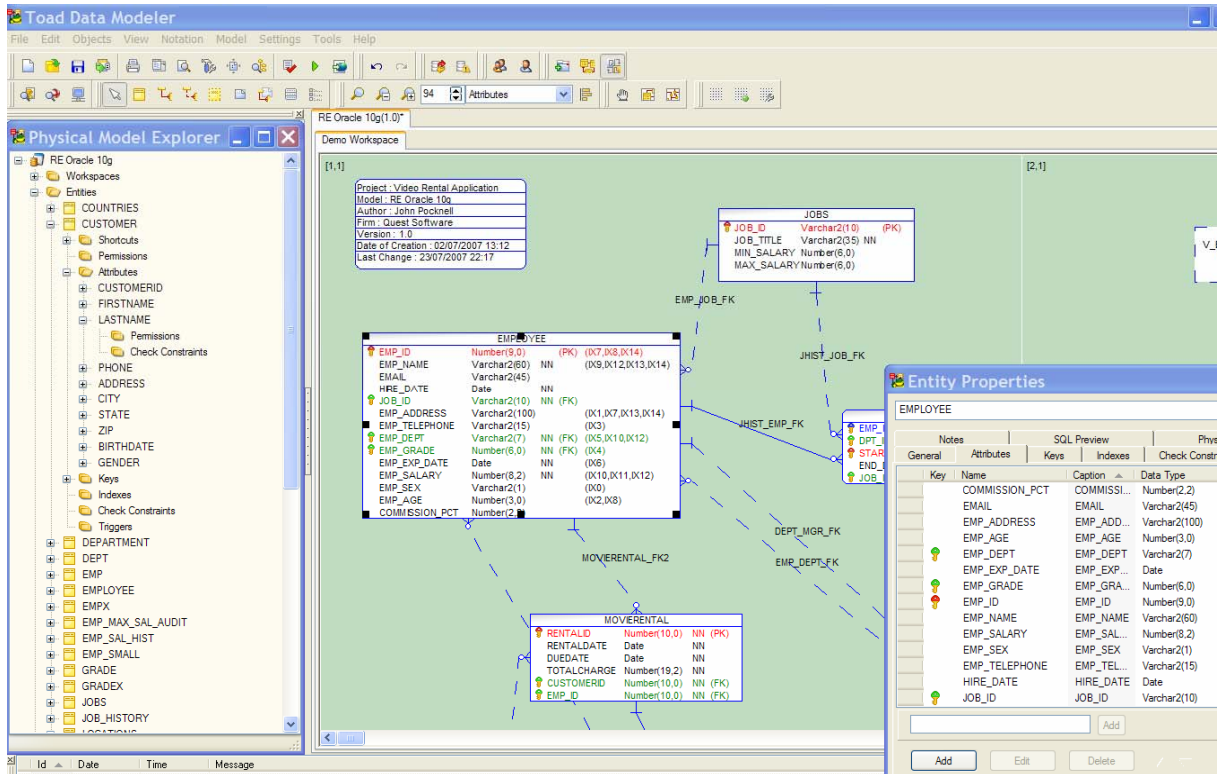


Figure 1 — Physical Data Model Complete with Full Table Properties

The DBA can produce and work with logical and physical data models and help development produce the optimum physical design.

The design should be validated to ensure that objects are created consistently and follow a standard naming convention with the appropriate use of partitioning, the creation of indexes and materialized views.

There are many ways that test environments can be created, but ensuring that all components are captured in the script in order to fully replicate the development schema environment can be challenging. To make this task easier, tools like Quest Software’s Toad for Oracle can generate a complete schema script for execution against a target environment. This includes the ability to select specific object types to migrate—filtering out anything not required—target tablespaces to use, specific scripting options and more. These scripts can be saved or checked into version control.

This same process can be used to build the initial preproduction deployment DDL script.

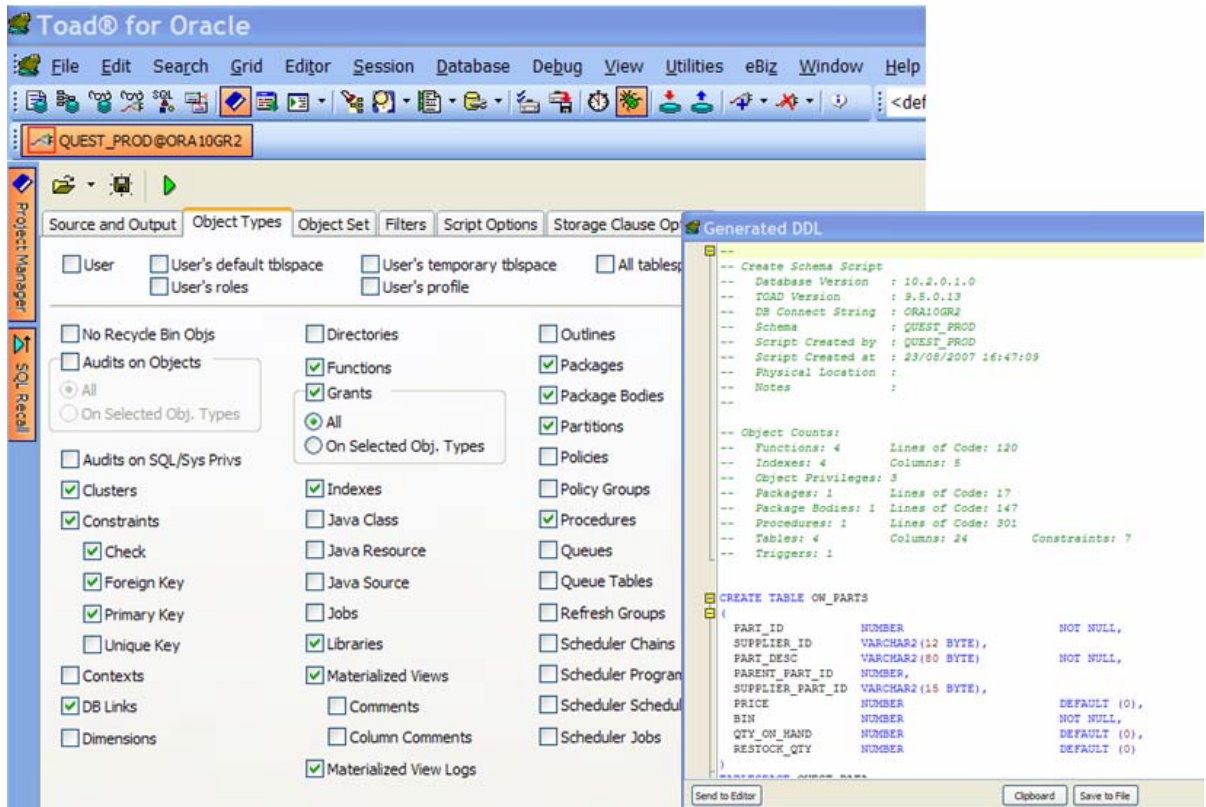


Figure 2 — Generation of Test Environment Script

Once the test environment has been created, it's important to simulate the production environment. Representative data can be generated and populated into the tables, and the instance configured as in production.

In order to anticipate application behavior in production, and prove the schema design is sound, it is extremely valuable to be able to simulate user transaction load. This will expose any weakness in design or instance configuration settings.

It then will be absolutely essential to perform a baseline test of the environment in order to make comparisons if changes such as patches, database upgrades and application changes are made.

Simplifying Oracle Database Administration

Here is a recommended approach for ensuring best practices are followed at the design stage. The DBA plays a vital role during this process.

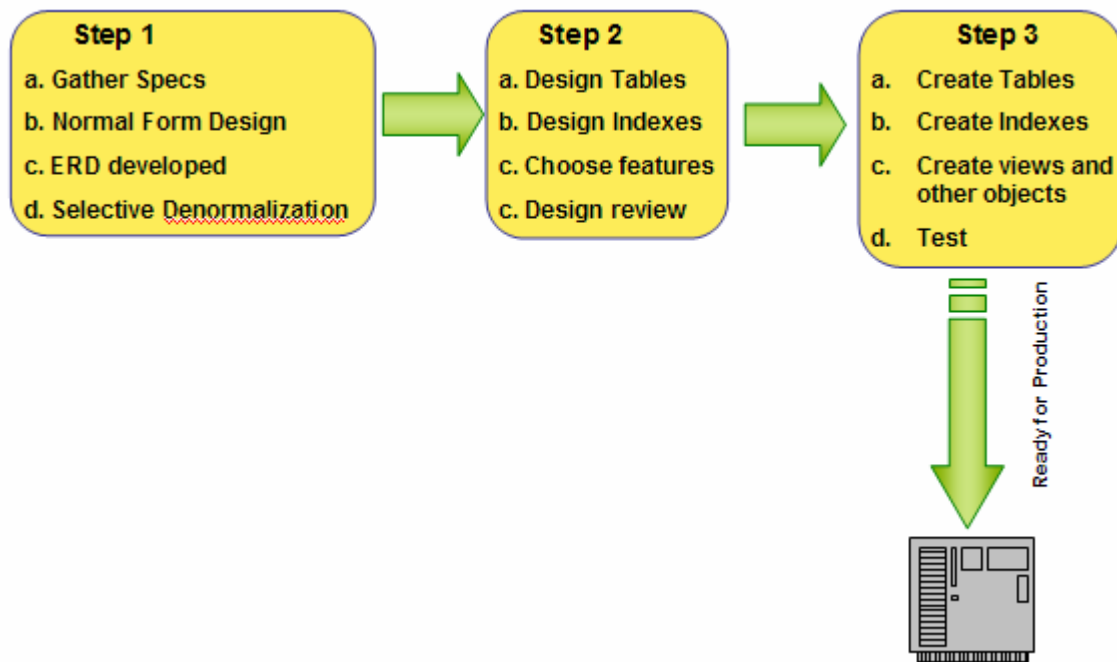


Figure 3 — Suggested Best-Practice Process for Schema Design for Production

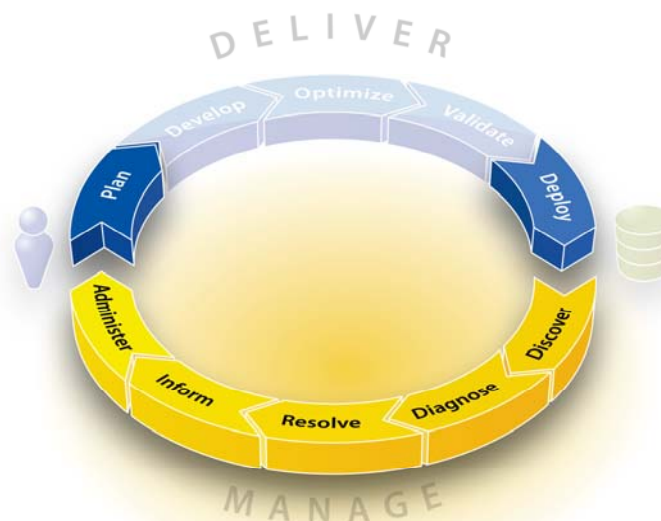
DATABASE ADMINISTRATION METHODOLOGY

It is generally recommended that DBAs perform specific tasks on a regular basis to ensure adequate controls are in place to minimize the amount of fire fighting needed.

There are excellent documents available that propose the nature of such tasks and when they should be carried out. For example, a paper written by Thomas B. Cox and Christine Choi² gives details for performing daily, weekly and monthly checks of the status of one or more Oracle databases.

There are a lot of tools on the market that can assist in making these tasks less painful. For example, Toad for Oracle helps accomplish many of these tasks with very little effort.

Quest Software has been proposing methodologies for many years. The following is the methodology we propose as the basis for simplifying database administration.



Detect

The first recommendation is that at the beginning of the day the DBA checks that end users can connect to and use the database.

Even if the telephone is already ringing upon arrival to the office, there must be a systematic approach to determining the root cause of a bottleneck or performance slowdown.

DBAs must not underestimate the usefulness of the information that end users can provide with regards to their experience. They may be the essential details needed to resolve the problem.

Investment in good monitoring tools will help DBAs operate in a much more proactive mode, because they provide the ability to anticipate problems before they happen or as they are happening.

Generally speaking, an Oracle database monitoring tool should follow these best-practices guidelines:

- Minimal impact on the system being monitored
- Proactive — DBAs shouldn't have to trawl for data
- Adequate alerting, reporting and e-mail/paging notifications
- References at hand for parameters, waits and problem resolution
- Sample rates that are high enough to catch problems as they happen, not afterward

There are many monitoring tools on the market for medium to large enterprises; however, for small to midsize shops, or for development DBAs, Toad for Oracle's DBA Module can prove quite effective.

We suggest the following methodology to determine the underlying cause of an apparent system slowdown:

1. Monitor the operating system to see if something is running external to Oracle, such as batch jobs. It is essential that this is ruled out first, before looking at the database.

Here's an example of the Unix monitor showing activity on a Linux operating system, while executing an industry-standard [TPC-C Benchmark](#).

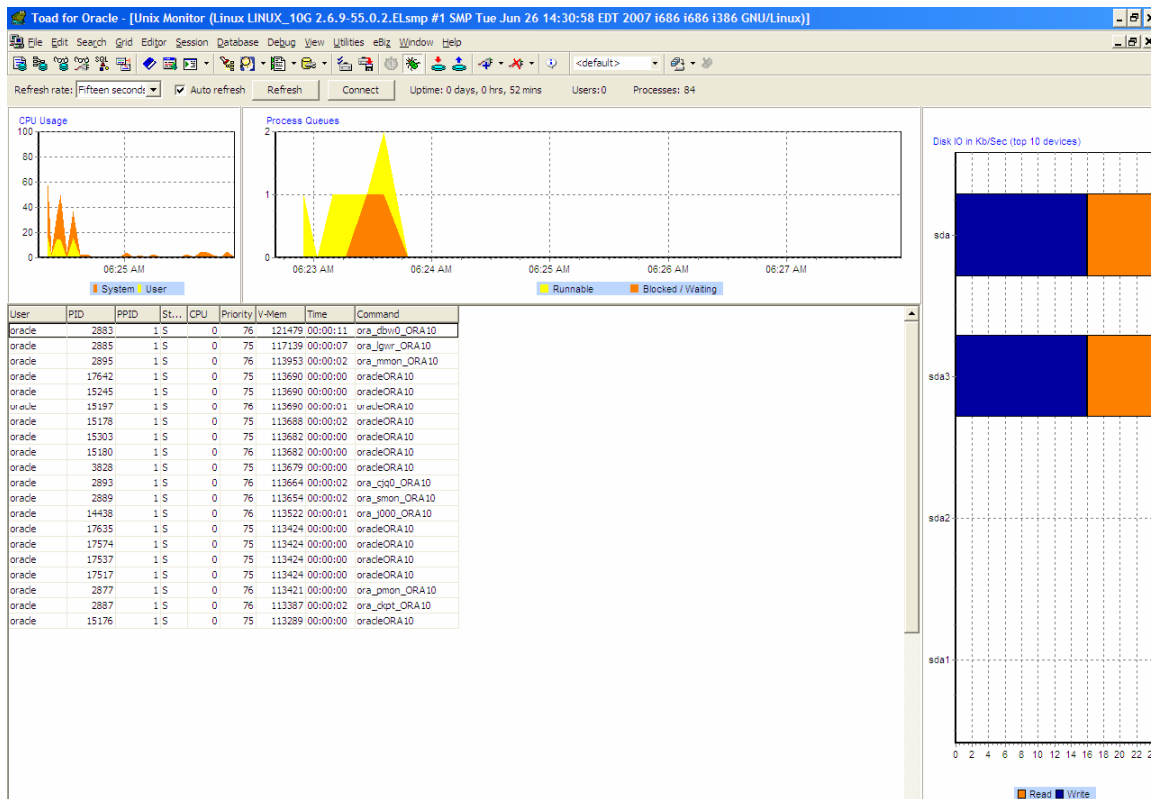


Figure 4 — Monitoring the Linux Operating System

The graph on the right shows that both I/O devices have heavy activity, and the process list shows both the Oracle database writer (DBWR) and log writer (LGWR) processes are top resource consumers. Both CPU usage and process queue are quite reasonable—and thus not an issue.

This Unix monitor uniquely found in Quest's Toad, works for Linux, Solaris, AIX, HP-UX and Tru64.

Simplifying Oracle Database Administration

2. Monitor the Oracle instances on the server to determine the one with the problem. (See Figure 5.)

A Database Browser lets users report on multiple instances to assess the problem.

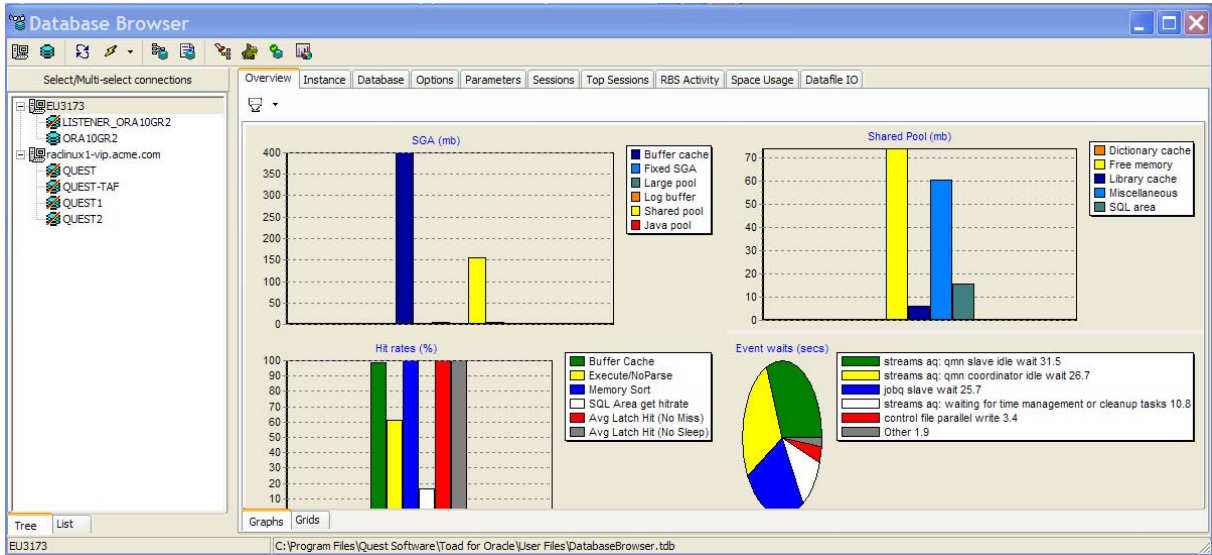


Figure 5 — Monitoring Multiple Oracle Instances

3. Focus on one instance.
 - a) Check to make sure that the server, database and TNS listener are running. (See Figure 6.)

Server	Node	Listener	Database	Version	Logon info for database test
JA817	✓	✓	✓	8.1.7.4.1	SYSTEM@JA817
JA92	✓	✓	✓	9.2.0.1.0	toad@JA92
PRV92	✓	✓	✓	9.2.0.1.0	SYSTEM@PRV92
SUNDB	✗	✓	✗		DEMO@SUNDB

Figure 6 — Monitoring the Oracle Server, Listener and Database

b) Monitor that instance for problems. (See Figure 7.)

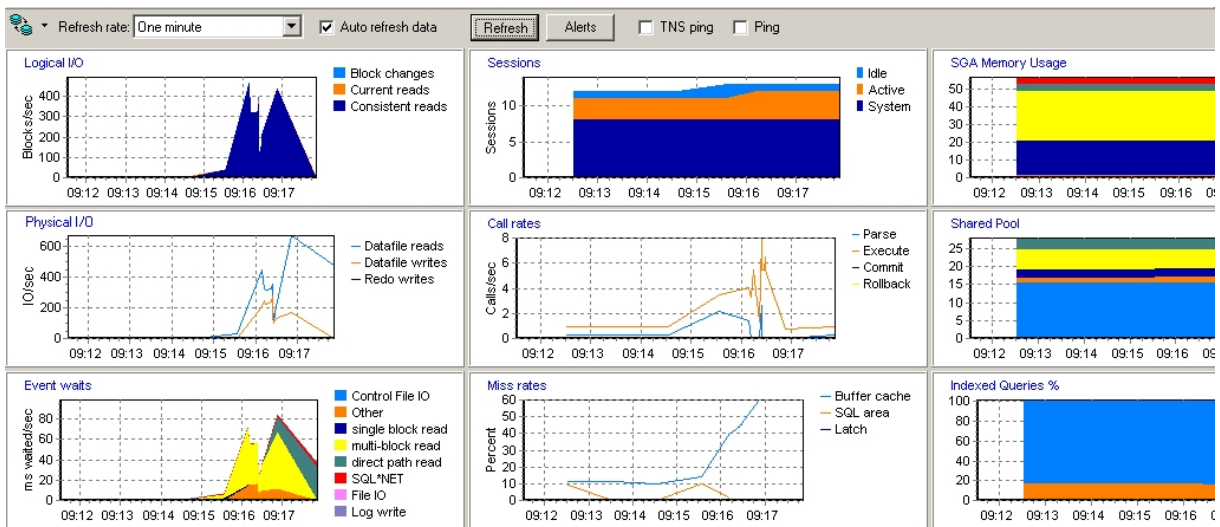


Figure 7 — Monitoring Key Oracle Database Metrics

4. Check alert logs. Via an FTP interface, you can access alert logs that are stored on a Unix or Linux server.
5. Verify rollback segments are online. A database browser will usually give a quick indication of this.
6. Check ADDM and AWR for instance performance problems. These utilities provide reports.
7. Examine top sessions for resource usage. (See Figure 8.)

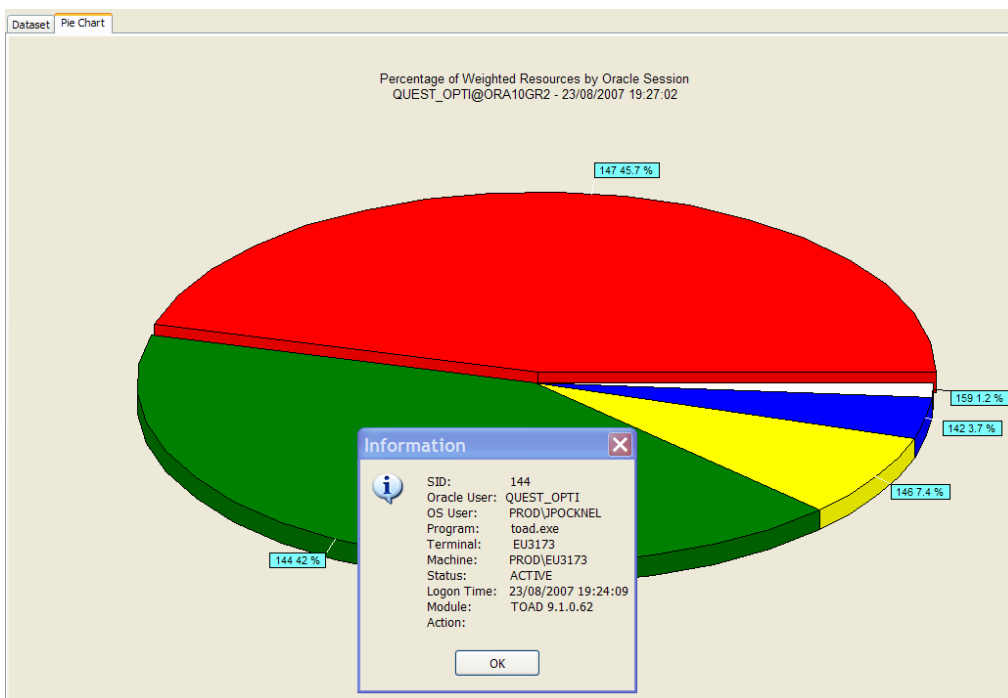


Figure 8 — Determining Top User Sessions

8. Monitor specific sessions — long operations, SQL statement execution and more. A session browser should show full session data, including the SQL statement the user is executing. (See Figure 12.)
9. Monitor tablespace usage for fragmentation and so forth. (See Figure 9.)

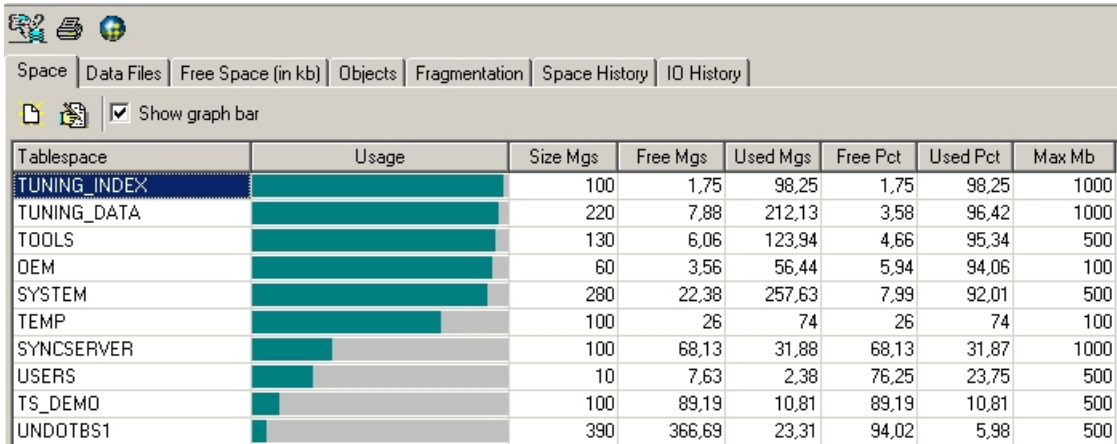


Figure 9 — Monitoring Tablespace Consumption

Diagnose

DBAs typically spend a huge proportion of their time just dealing with SQL statement execution problems, so rapid resolution is essential.

After determining the source of the issue, the DBA now needs to quickly and effectively diagnose the exact cause of the problem, and get users up and running again. Doing this as seamlessly as possible requires access from the detect window into a resolution context.

In this screenshot of Toad for Oracle's proprietary database probe window, for example, you see real-time display of instance activity with alert markers indicating potential problems. Clicking on a marker displays a description of the problem—in this case, excessive redo log buffer size.

The screenshot shows the Toad for Oracle interface with the following components:

- Sessions:** Total: 6/170, Active: 1, Lock Wait: 0.
- Processes:** Total: 23/150, Background: 14, Dispatchers: 1/-1, Dedicated: 6, Parallel: 0/20.
- PGA - 29 MB:** Total: 15/51.
- SGA - 584 MB:**
 - DB Buffer Cache:** Overall: 104/380 (98%), Default: 104/380 (98%), Recycle: 0/0 (0%), Keep: 0/0 (0%).
 - Shared Pool:** Overall: 180 (93%), Dict Cache: 0 (93%), Lib Cache: 1/4 (96%), SQL Area: 14, Misc: 59, Free: 102.
 - Large Pool:** Megs: 4, Cur Used: 7, Max Used: 23.
 - Redo Log Buffer:** Megs: 3 (highlighted with a red circle), Retries: 0, Waits: 0.
- Files:**
 - Redo Log Files:** Groups: 3, Current: 2 (33%), Megs: 150, % Active: 33.
 - Archived Log Files:** ArchiveLogMode: [unchecked], Log Switch Rates: Past Hour: 0, Past Day: 0.
 - Data Files:** Tablespaces: 10, Data Files: 10, Megs: 6239, % Used: 39.
- Alert Panel:**
 - Excessive Redo Log Buffer Size:** The redo log buffer appears to be too large. Applications the usually need to change the default log buffer size. The log b modestly sized log buffer can significantly enhance throughp reasonable first estimate for such systems is to make the log larger than 1 MB does not provide any performance benefit, negative implications on performance or recoverability. It me
 - Expression: Redo_Log_Buffer_Megs > 1
 - Do not show this warning in the future (deactivate alert)

Figure 10 — Diagnosing an Instance Configuration Problem

Another example might be slow database performance due to a user executing an inefficient SQL statement—exposed by looking at top sessions by CPU. With a view of all Oracle sessions and the ability to obtain information about each session—for example, the SQL statement the user is executing—the DBA has a quick way to resolve a problem.

In the case where the statement has already completed execution, the DBA needs to look in the SGA or access the Oracle trace file.

Resource-intensive SQL can be captured and analyzed with a tool like Toad's SQL Optimizer, SGA Inspector Module. With the SGA Inspector, DBAs can identify top

Simplifying Oracle Database Administration

SQL statements and capture SQL that is currently running, run-time execution plans and performance metrics.

The screenshot displays the Quest SQL Optimizer 7.0 for Oracle Trial interface. The main window is titled "QUEST SQL OPTIMIZER FOR ORACLE" and features a navigation bar with tabs: Home, Batch Optimizer, SQL Scanner, SGA Inspector (selected), Tuning Lab, Indexing, Impact Analyzer, and Outlines. Below the navigation bar, there is a "Job List" section with a "Group" field set to "snapshot (QUEST_OPTI@QUEST)".

The "SQL Statistics" section shows a table with the following data:

COLLEC...	ELAPSED_TIME	CPU_TIME	DISK_READS	BUFFER_GETS	EXECUTIONS	DISK_READS/...	BUFFER...	DISK_RE...	PARSE_...
1	9216589	7917685	10919	600156	2	5459.5	300078	0.0182	2
2	8100482	5893034	10084	600044	2	5042	300022	0.0168	2
3	3223744	467068	9349	11514	2	4674.5	5757	0.812	2
4	2860471	944157	8606	10868	2	4303	5434	0.7919	2
5	4515548	3942231	5459	300100	1	5459	300100	0.0182	1

The "SQL Text" section displays the following SQL statement:

```
D.LOCATION_ID,  
L.COUNTRY_ID,  
E.EMP_NAME,  
E.EMP_SALARY,  
E.COMMISSION_PCT,  
D.DPT_NAME,  
J.JOB_TITLE,  
L.CITY,  
L.STATE_PROVINCE,  
C.COUNTRY_NAME,  
R.REGION_NAME  
FROM EMPLOYEE E,  
DEPARTMENT D,
```

The "Execution Plan" section shows a hierarchical plan for a "SELECT STATEMENT ALL_ROWS" with a cost of 1,228. The plan includes several "HASH JOIN" operations and "TABLE ACCESS FULL" operations on tables like QUEST_OPTI.JOBS and QUEST_OPTI.DEPARTMENT. The bottom status bar indicates "14 Rows were returned by the SELECT statement."

Figure 11 — SGA Inspector Module in Toad's SQL Optimizer

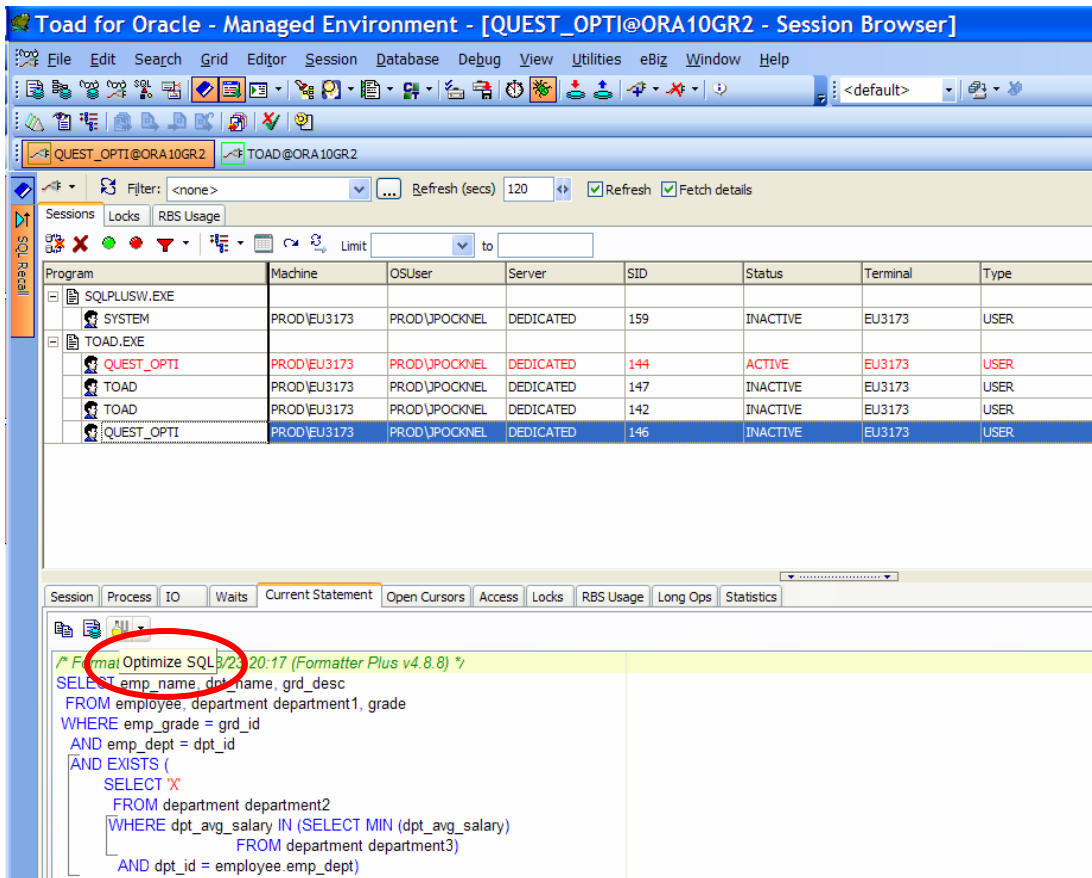


Figure 12 — Diagnosing a User SQL Execution Problem

In all cases, direct access to SQL optimization technology, through a user session window automates the process of improving execution time.

Resolve

After diagnosing the root cause of a problem, whether it has to do with database configuration, tablespace, rollback segment or SQL execution performance, it is important to resolve the problem quickly and effectively.

Integration points in the tools will help enormously. For example, the screenshot above shows that Quest's SQL Optimizer can be launched directly from the Toad for Oracle diagnosis window.

It is also important in any resolution situation to receive guidance, or tips, based on advice from industry experts. This is helpful especially in the case where DBAs may be relatively junior and require assistance in reaching the correct diagnosis.

One of the most time-consuming and problematic examples of this occurs when resolution requires improvements to the SQL execution time. There are quite a few SQL tuning solutions on the market, including Oracle's own Tuning Pack, and it is essential to provide an effective fix while maintaining visibility into the rectified SQL statement or index and its effect on the database.

SQL tuning tools such as Quest's SQL Optimizer simplify and automate the process of optimizing SQL and PLSQL, and can be configured according to the business process being used. SQL Optimization technology completely rewrites a SQL statement and provides the most efficient SQL alternative and execution plan. It also integrates index advice for multiple SQL statements, virtual index impact analysis and plan change analysis.

A batch-oriented process gives DBAs the freedom to optimize SQL according to rules they set and the manner in which the solution is deployed back into the application.

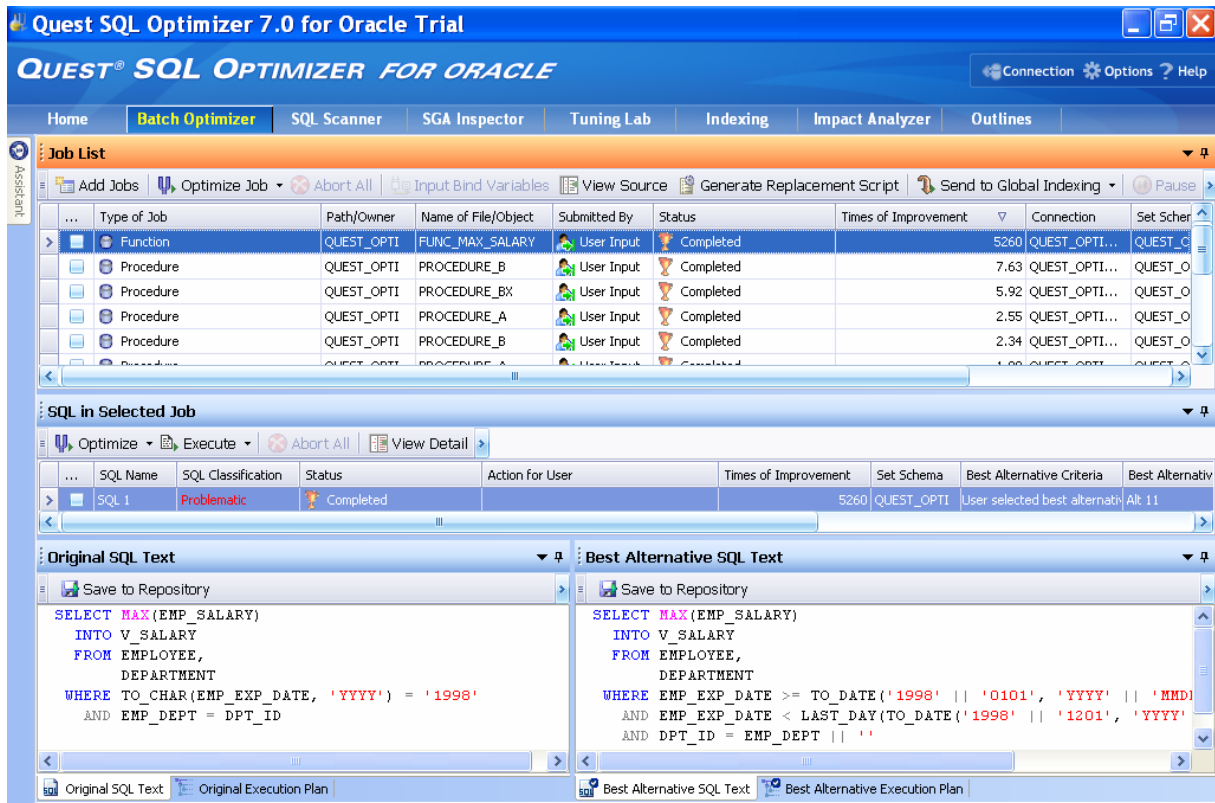


Figure 13 — Resolving a User SQL Execution Performance Problem

Inform

Information through reporting is essential for a number of reasons, from simple documentation to compliance and auditing purposes.

DBAs needs to be able to generate reports for:

- The database instance, including access to ADDM and AWR
- Schema objects
- Database health checks
- Data validation, such as for duplicate data

Administer

There are many general administration tasks that DBAs perform. It is important that the act of performing both has minimal impact on the database and does not disrupt the DBA's ability to deal with critical day-to-day end-user issues.

Database administration tools, therefore, need to simplify these tasks in a way that still keeps the DBA in touch with what is really happening. The last thing a DBA wants is a magic button that purports to fix everything automatically without giving them visibility into what's really going on.

Following is a summary list of general administration tasks that DBAs may be called upon to execute:

- Environment management — OS, disk, RAM, network
- Instance management — SGA, PGA, background processes
- Object management — Tables, indexes
- File management — Control, redo, undo
- User management — Roles, privileges, auditing
- Database maintenance — Backup and recovery
- Utilities — Data pump, import, export, LogMiner, Flashback
- Performance management — Instance, SQL execution
- Availability

Another essential characteristic that should be considered when choosing administration tools is workflow navigation. If a DBA is struggling to fix a problem or trying to get information from Oracle, the proposed solution needs to be presented in a clear, logical way that makes sense and is easily repeatable without having to make notes.

An example is shown below, where core administration functionality is clearly presented in a functional paradigm (Administer, Compare, Create, Diagnose and others), rather than by feature. This way, the tool relates more to what the DBA is trying to do, as opposed to some product-focused feature name.

Simplifying Oracle Database Administration

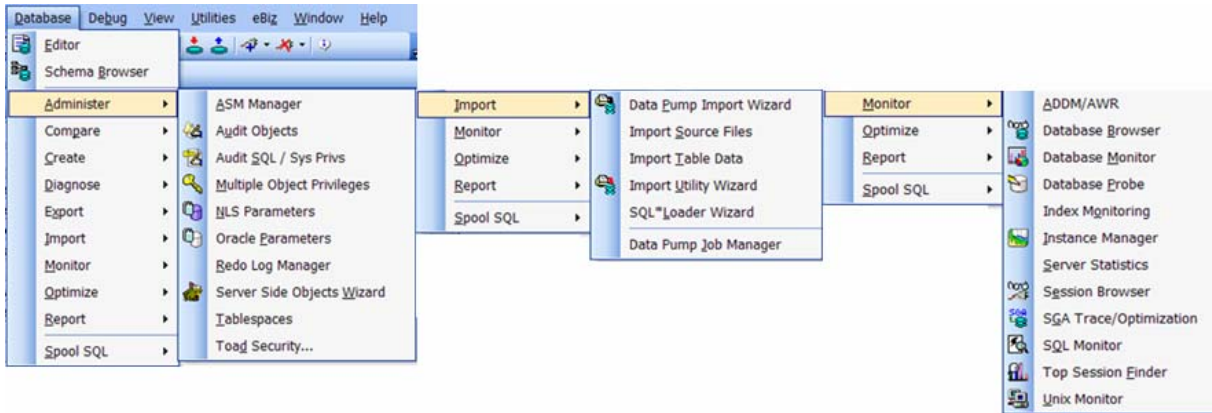


Figure 14 — Administration Task Presentation Should Be Clear and Function-Based

In terms of object management, object presentation should be clear with every feasible operation on the selected object able to be initiated directly from that object—whether single or multiple objects are selected.

Objects that don't require management or are not appropriate to the version of Oracle being managed should not be displayed, to reduce overcrowding the GUI.

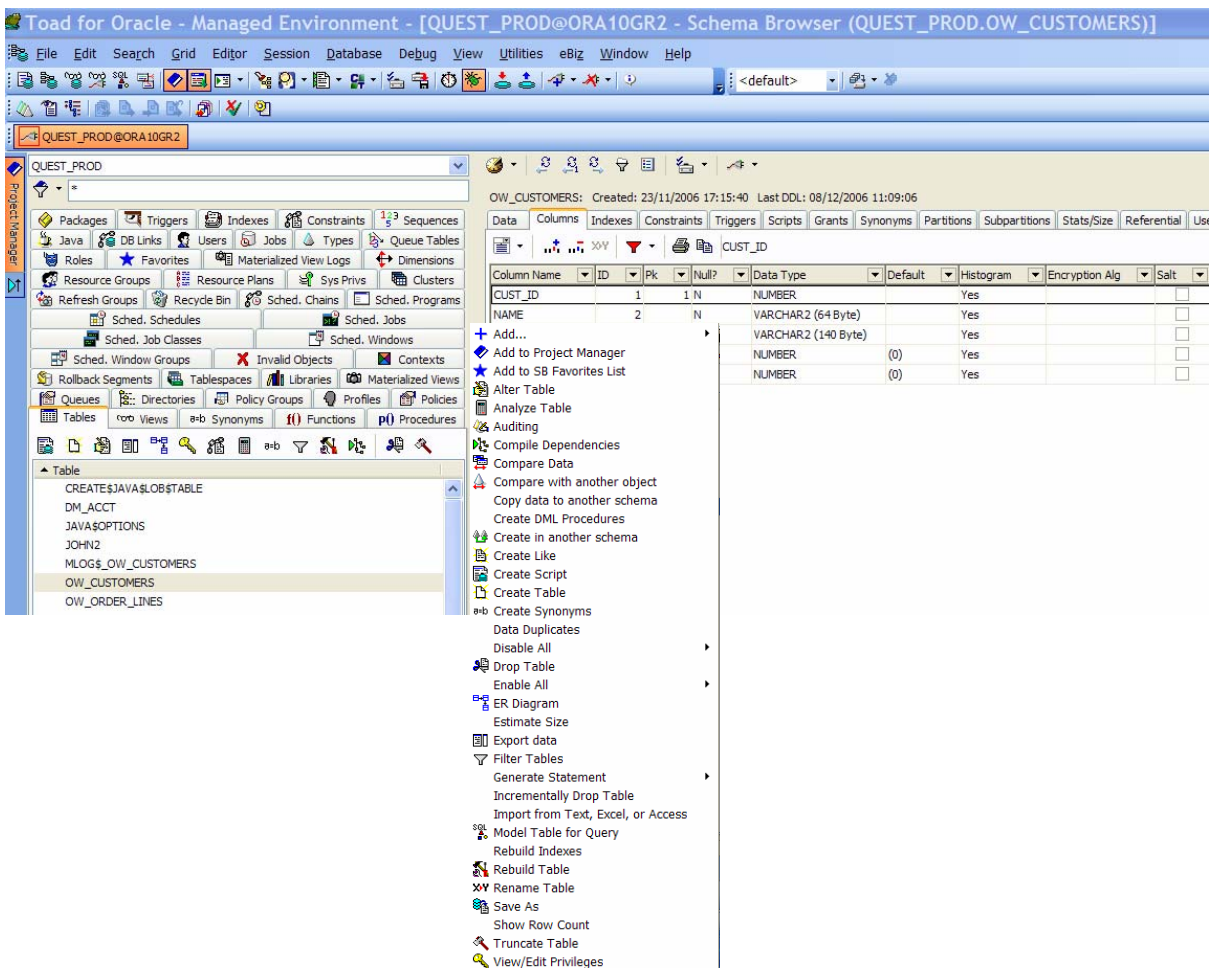


Figure 15 — A Tabbed Object Browser (Oracle 10gR2) and Object Right-Click List

PRODUCTION VALIDATION

One of the most overlooked areas when releasing applications into production is validation that the application will work at least as well in production as it did in test. Despite every effort to replicate production-like conditions in a test environment, this is sometimes difficult to achieve in practice, particularly if using more advanced capabilities of Oracle, such as Real Application Clusters (RAC).

Performance benchmarking is one of the key requirements for anticipating the impact of database changes as a result of:

- Oracle patches
- Oracle upgrades
- Application changes
- Increased user load

Tests can be run in a nonproduction environment and compared against prior benchmarks in order to mitigate the risk of problems when changes are deployed into a live environment.

BENCHMARKS

With database performance and scalability testing, there are several standard benchmarks—some of which are shown in Figure 16—each intended to test different aspects of a database environment.

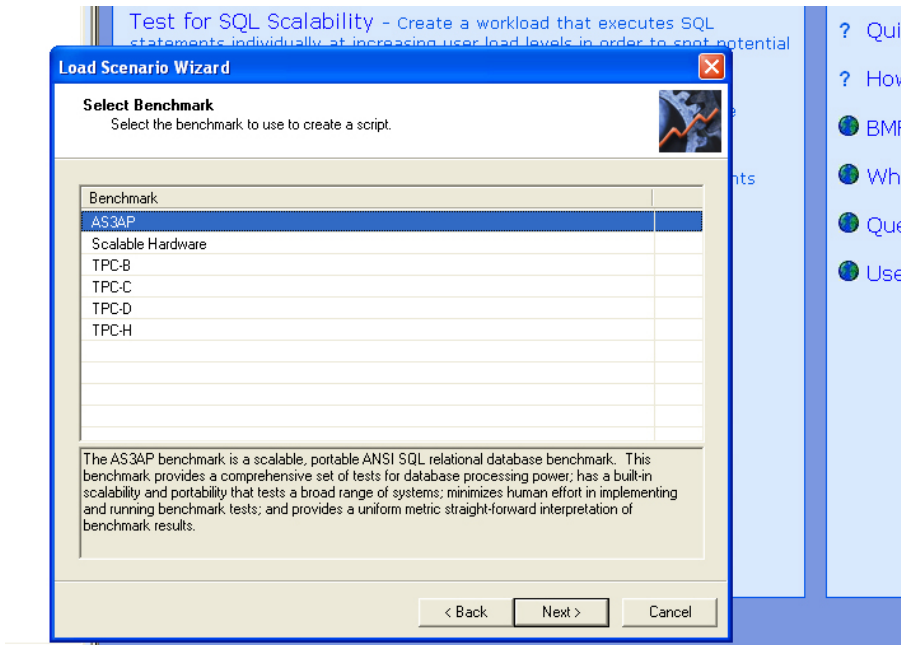


Figure 16 — Standard Benchmarks Provided in Quest's Benchmark Factory

The most common benchmarks are the TPC-C and TPC-H, which test online transaction processing (OLTP) and decision support/data warehouse-type database loads, respectively.

What's good about automated tools, such as Quest's Benchmark Factory, is that they allow the user to define timing and user loading, so multiple what-if scenarios can be validated. In addition, custom-designed scenarios, such as high-load SQL statements, can be tested at various user load levels to determine scalability.

I'M A DBA, NOT A QC ENGINEER

DBAs need to understand the use of benchmarks to document their systems' performance. The practice of setting baselines or thresholds depends on benchmarks of performance for steady-state operations. The capability to perform repeatable and scalable tests is critical to predicting future needs and documentation of current limits as well as detecting possible performance anomalies, before they impact production environments.

Snapshot testing is where a single snapshot of production load is used to test new configurations. This tactic is limited in its usability in that it is not possible to vary the timing—other than eliminating concurrency—or the user load. It just replays what exactly happened. Snapshot testing is good for the question, “Will this upgrade affect my system?” It cannot answer questions such as, “How will a 20 percent increase in user load affect my system?” This is where the DBA must understand the use of scalable benchmark testing.

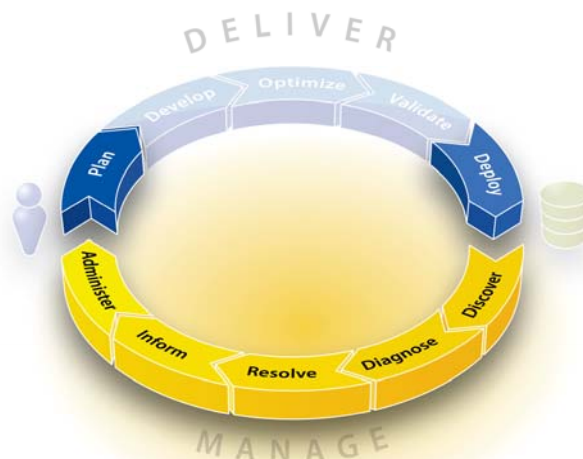
In other testing, such as limited SQL replay tests, the SQL statements being tested are usually only replayed once and in serialized order. How often does that happen in real life? For determining if a statement has changed the way it executes SQL, replay is fine. In order to determine whether a statement will scale or will work and play well with others in a concurrent environment, it is useless. To test concurrent statement interaction and statement scalability, DBAs need a scalability testing tool.

CONCLUSION

When making the right choice for a tool or suite of tools to simplify database administration, there are some key considerations that should be taken into account:

- Does the tool or suite continue to support Oracle versions currently implemented and planned for the future?
- Does the tool or suite support the key features offered by the Oracle version being used?
- Does the tool or suite provide a logical workflow, or would DBAs have to fight with it?
- Does the tool integrate well with Oracle utilities and features, and with the vendor's own products in a way that supports logical workflow?
- Can the tool or suite be used equally well by junior and senior DBAs through context-sensitive notes, tips and more?
- Can both junior and senior DBAs find value and increased productivity from the tool or suite?
- Is the tool or suite adaptable to the DBA's style of database administration, or does it make the DBA adapt to it?
- To what extent would the tool or suite actually cause resource impact on the database when being used—for example, monitoring?
- Does the tool or suite offer DBA and management reporting, and auditing information for compliance?

Choosing the right tools to simplify administration will help get DBAs out of their current reactive states and enable them to plan strategies and forecast future needs before they become critical issues to the business.



REFERENCES

¹ <http://blogs.ittoolbox.com/database/solutions/archives/a-dbas-new-year-resolutions-6043>

² <http://www.quest-pipelines.com/pipelines/dba/archives.htm#code28>

User Communities:

- Toad World: www.toadworld.com
- Data modeling: <http://modeling.inside.quest.com/index.jspa>
- Database benchmarking: <http://benchmarkfactory.inside.quest.com/index.jspa>
- SQL optimization: <http://sqloptimizeroracle.inside.quest.com/index.jspa>

Blogs:

- John Pocknell:
<http://www.toadworld.com/Community/ExpertsBlog/tabid/67/BlogID/16/Default.aspx>
- Bert Scalzo:
<http://www.toadworld.com/Community/ExpertsBlog/tabid/67/BlogID/14/Default.aspx>
- Mike Ault:
<http://www.toadworld.com/Community/ExpertsBlog/tabid/67/BlogID/11/Default.aspx>

ABOUT THE AUTHORS

John Pocknell is a product manager at Quest Software working at the company's European headquarters in Maidenhead, UK. He has been with Quest since 2000, focusing on the database design, development and testing product areas. John also spends significant time with customers at Toad and Oracle user group meetings throughout Europe and the Asia-Pacific region, promoting Toad and presenting Toad training courses. John has worked in IT for over 20 years, primarily in Oracle-based application design and development. He is also an aeronautical engineer, with more than 10 years of experience as a business development manager delivering IT consultancy services and implementing quality assurance systems to the ISO 9001 standard.

Bert Scalzo is a Product Architect for [Quest Software](#) and a member of the [TOAD](#) team. He has worked extensively with TOAD's developers and designed many of TOAD's features, including the DBA module, Code Road Map and Code Xpert. Mr. Scalzo has more than two decades of experience with Oracle databases, starting with version 4, and served at both the Oracle Education and Oracle Consulting organizations. He holds several Oracle Masters certifications as well as bachelor's, master's and Ph.D. degrees in computer science. Additionally, Mr. Scalzo holds an MBA and several insurance industry designations. He is also an accomplished speaker and has presented at numerous Oracle conferences and user groups. He is the author of four books: "[Oracle DBA Guide to Data Warehousing and Star Schemas](#)", "[TOAD Handbook](#)", "[TOAD Pocket Reference](#)" (2nd Edition), and "[Database Benchmarking: Practical Methods for Oracle 10g & SQL Server 2005](#)". You can reach Mr. Scalzo at bert.scalzo@quest.com or bert.scalzo@yahoo.com.

Mike Ault is an Oracle database specialist at Quest Software and a recognized Oracle expert with more than 16 years of experience as an Oracle DBA and consultant in a variety of industries and companies. Mike began his foray into computing in 1980, following six years of service in the Nuclear Navy riding submarines. He began programming in Basic and Fortran IV on the PDP-11 architecture, in addition to working with PDP, IBM-PC, Osborne and later VAX-VMS and HP architectures and with the Informix and Ingres databases. Following the downturn in the nuclear industry Mike began working with Oracle as the only DBA at the Iuka, Mississippi-based Advanced Solid Rocket Motor (ASRM) project for NASA in 1990. Since 1990 Mike has used his Oracle expertise in a variety of industries, working in-house and as a consultant.

ABOUT QUEST SOFTWARE, INC.

Quest Software, Inc. delivers innovative products that help organizations get more performance and productivity from their applications, databases and Windows infrastructure. Through a deep expertise in IT operations and a continued focus on what works best, Quest helps more than 50,000 customers worldwide meet higher expectations for enterprise IT. Quest Software can be found in offices around the globe and at www.quest.com.

Contacting Quest Software

Phone:	949.754.8000 (United States and Canada)
Email:	info@quest.com
Mail:	Quest Software, Inc. World Headquarters 5 Polaris Way Aliso Viejo, CA 92656 USA
Web site	www.quest.com

Please refer to our Web site for regional and international office information.

Contacting Quest Support

Quest Support is available to customers who have a trial version of a Quest product or who have purchased a commercial version and have a valid maintenance contract. Quest Support provides around the clock coverage with SupportLink, our web self-service. Visit SupportLink at <http://support.quest.com>

From SupportLink, you can do the following:

- Quickly find thousands of solutions (Knowledgebase articles/documents).
- Download patches and upgrades.
- Seek help from a Support engineer.
- Log and update your case, and check its status.

View the ***Global Support Guide*** for a detailed explanation of support programs, online services, contact information, and policy and procedures. The guide is available at: [http://support.quest.com/pdfs/Global Support Guide.pdf](http://support.quest.com/pdfs/Global%20Support%20Guide.pdf)