It's Time to Get Agile

The complete story behind accelerating database DevOps

Software

Introduction

Shorter time to value, lower risk and greater flexibility? With benefits like these, it's definitely time to get agile.

The agile methodology is an alternative to traditional project management, like waterfall. It helps development teams respond to unpredictability through incremental, iterative work cycles and collaboration between self-organizing, cross-functional teams.

While most application developers now consider agile to be mainstream, database developers have been slower to embrace it because of the need to respect the state of a database when deploying changes. Instead, database professionals have had to rely on more manual techniques that do not scale up to the faster development cycles that businesses today require.

It's time for database development to keep pace with the agility seen in application development environments.

Agile methodologies can help you and your database development team speed development cycles with automation, accelerate project delivery and implement consistent repeatable development processes.

This e-book offers guiding principles for creating agile development environments and suggests tools to help you get there. We'll also explain how the right DevOps solution can help your team quickly and easily get agile around these key development tasks:

- Code changes and version control
- Unit testing
- Static code analysis
- Deployment staging
- Automation of everything

Get agile to speed development cycles with automation, accelerate project delivery and implement consistent, repeatable development processes.

Guiding principles for agile development

The role of database development has become more important than ever as companies launch proprietary applications to create new revenue streams, solve problems, collect and analyze data, and help businesses stay competitive. Historically, database developers have not been included in the fast-moving application planning process, yet they are still responsible for creating high-quality, high-performance database code to run mission-critical business applications.

By following a few basic guiding principles, you can help set your team up for getting agile — and delivering more efficient, reliable, high-quality database code:

- Improve productivity
- Ensure correct code function
- Increase code quality, performance and scalability

One way to implement practices in line with these principles — and save hours of development time in the process — is to automate day-to-day tasks from SQL optimization to testing and code reviews using the right tools.

Improve productivity

Agile methodology enables developers to implement changes individually and continuously rather than making batch updates over long release cycles. Certain technology solutions can help speed the review and implementation of code changes during initial development, ongoing maintenance and application improvement:

- **Build tools** enable developers to work more quickly and efficiently with a user-friendly software interface that provides access to all the tasks needed during development. Some tools offer predefined project coding standards, which saves considerable time by embedding and automating the code review process.
- **Editing tools** allow developers to quickly identify faulty lines of code or edit specific lines rather than execute a full program overhaul. The best tools combine editing, testing, debugging, standards review and tuning to help boost productivity.
- **Collaboration tools** enable team members to work concurrently using shared coding standards without risking version control — and your code integrity.

When looking at database management tools, do not sacrifice quality for speed. Faster development without proper testing is not what agile is about.

Get agile to implement changes individually and continuously rather than making batch updates over long release cycles.

Ensure correct code function

To truly get agile, database organizations need effective integration with business strategists and users to comprehensively define requirements. Visualizing complex code interdependencies during initial code development can help improve understanding and minimize errors. Ongoing testing throughout the development cycle that involves input from and reporting to business units is critical to successful deployments. Consider the following solutions:

- **Review and refactoring tools** help expedite analysis of code functionality. Look for technology solutions that graphically show code dependencies and provide opportunities to refactor code. Determining the full impact of what appears to be a simple change saves time and reduces the risk and cost of future rework.
- **Testing tools** save time during debugging, testing and code reviews. Software solutions that generate test data (randomized or real world) and perform automated unit testing provide the insights developers need to understand how an application will run in production so they can make changes accordingly.
- **Troubleshooting tools** provide visibility down to the line of code to help identify faulty code and streamline ongoing maintenance.

Get agile to boost application performance while reducing the resources and time needed to execute.

Increase code quality, performance and scalability

Database development tools help developers build top-quality applications that will perform well in production. Tuning and testing code gives developers an opportunity to boost application performance while potentially reducing the resources and time needed to execute. Consider the following solutions:

- **Code review tools** automate the process by flagging coding violations as developers write code. Be sure to look for solutions that are comprehensive and customizable as well as those that provide summary management dashboards.
- **Change management tools** help developers understand how one change can affect the entire development environment. The best tools provide code mapping to help visualize dependencies within a database.
- **Optimization tools** simplify the task of tuning SQL to improve code performance. These solutions enable quick scans for problematic code, and some recommend alternate SQL implementations to boost efficiency with a single click through a management interface.
- **Collective intelligence** helps database developers stay in touch with emerging development methods to address changing business initiatives. The best solutions are updated frequently and use industry-leading experts and recommendations from developer customers.
- **Scalability tools** ensure code and SQL coming from development will scale up to the number of transactions expected in production and still perform within acceptable limits.

DevOps solutions for database development

Agile development ultimately requires a fundamental shift in the way teams work. Instead of having people manually walk through each step of the process, teams must look for ways to create fully automated pipelines that submit changes to regression testing, review and staging for deployment without further interaction from the team. One way to start is by implementing a DevOps methodology.

DevOps is a software development method that stresses communication, collaboration, integration, automation and measurement of cooperation between software developers and other IT professionals. And it's critical if you want to get agile.

Having a DevOps methodology in place and the tools to support it are key to keeping pace with mobility, cloud technologies, big data analytics, social business and other technologies as they emerge. DevOps can help you create an agile development model that:

- Speeds deployment processes
- Enables fast response to changing business requirements

Get agile to keep pace with mobility, cloud technologies, big data analytics, social business and other technologies as they emerge.

- Reduces the risk of production changes causing outages
- Breaks down silos

Get agile with Toad for Oracle

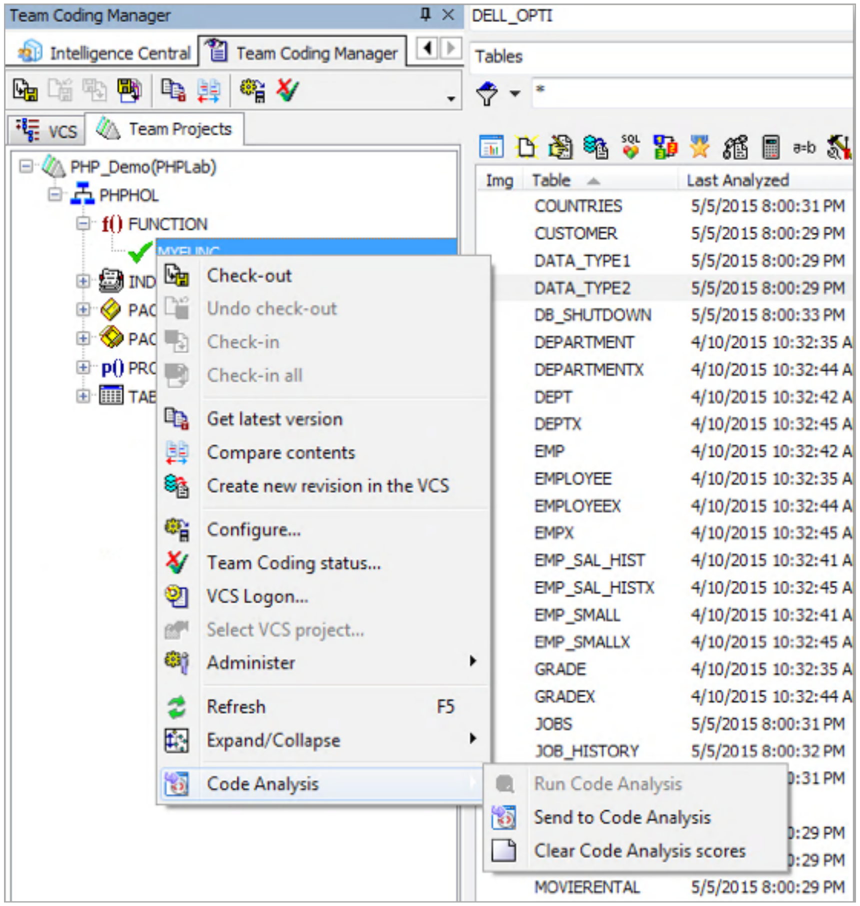
Automating database development cycles as part of an agile DevOps strategy can help remove process bottlenecks to meet tight project deadlines and improve code quality. Contrary to the myth of higher risk, automation can actually help reduce risk. In fact, automation is key to synchronizing the development cycles of application software teams and database teams.

Toad for Oracle provides database tools that help your organization implement a DevOps strategy to not only automate manual tasks but ensure code quality, speed time to market, lower costs, reduce risk and improve productivity. More specifically, Toad can help with these five key tasks:

- Code changes and version control
- Unit testing
- Static code analysis
- Deployment staging
- Automation of everything

Manage code changes

While application developers can create updates and overwrite complete versions of code, database developers typically maintain a



Img	Table	Last Analyzed
	COUNTRIES	5/5/2015 8:00:31 PM
	CUSTOMER	5/5/2015 8:00:29 PM
	DATA_TYPE1	5/5/2015 8:00:29 PM
	DATA_TYPE2	5/5/2015 8:00:29 PM
	DB_SHUTDOWN	5/5/2015 8:00:33 PM
	DEPARTMENT	4/10/2015 10:32:35 A
	DEPARTMENTX	4/10/2015 10:32:44 A
	DEPT	4/10/2015 10:32:42 A
	DEPTX	4/10/2015 10:32:45 A
	EMP	4/10/2015 10:32:42 A
	EMPLOYEE	4/10/2015 10:32:35 A
	EMPLOYEEX	4/10/2015 10:32:44 A
	EMPX	4/10/2015 10:32:45 A
	EMP_SAL_HIST	4/10/2015 10:32:41 A
	EMP_SAL_HISTX	4/10/2015 10:32:45 A
	EMP_SMALL	4/10/2015 10:32:41 A
	EMP_SMALLX	4/10/2015 10:32:45 A
	GRADE	4/10/2015 10:32:35 A
	GRADEX	4/10/2015 10:32:44 A
	JOBS	5/5/2015 8:00:31 PM
	JOB_HISTORY	5/5/2015 8:00:32 PM
		:31 PM
		:29 PM
		:29 PM
	MOVIERENTAL	5/5/2015 8:00:29 PM

Figure 1. Seamlessly check code in or out, directly from within Toad instead of juggling multiple tools.

current version of the database at all times. Changes must be made through detailed scripting to avoid data loss.

Version control in database development projects is actually a valuable way of tracking data definition language (DDL) changes every step of the way. Tracking revisions to stored functions and procedures over time makes it easier to access them, compare them side by side and diagnose any problems that arise on the path toward production.

Toad for Oracle includes a powerful source control feature, Team Coding, that helps development teams manage code changes in the database, including preventing developers from overwriting the changes of other team members. Figure 1 shows some of the options available to you. With Toad, you can:

- **Use your existing version control system.** Toad works with all the major version control systems, including Git, Microsoft® Team Foundation Server (TFS), Concurrent Versions System (CVS), IBM® ClearCase®, Perforce and Apache Subversion® (SVN).
- **Improve productivity.** Developers can seamlessly check code in and out, directly from within Toad instead of juggling multiple tools.

> Get agile by using version control to track DDL changes every step of the way.

Test early and often

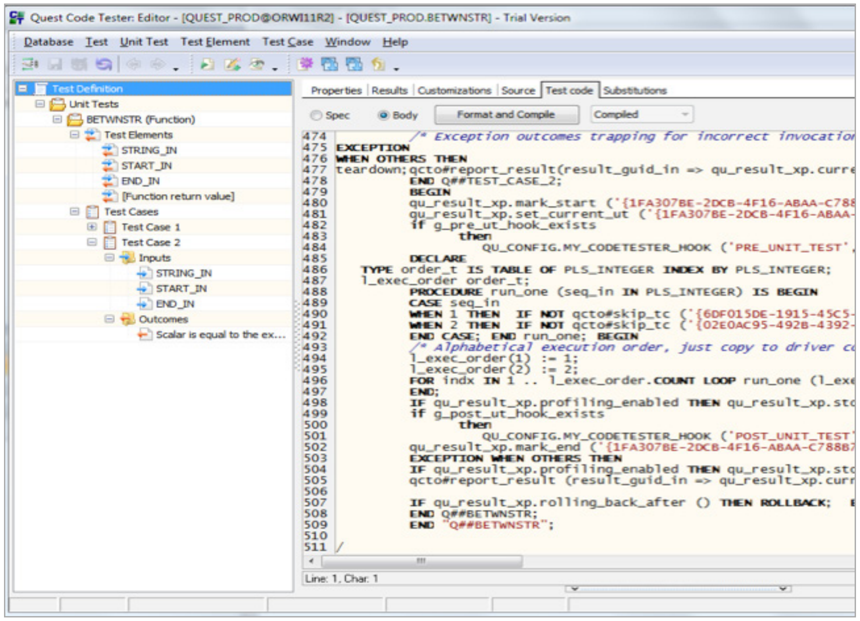
The only way to be certain that a proposed change does not break something else is through automated unit and regression testing. Even if developers and QA teams are comprehensive and remember everything they need to test, manual testing is still vastly slower than automated tests that run against the code change at the point of check-in.

Automated testing provides a safety net for accelerating database deployments because it offers near-immediate assurance that new changes do not break earlier work. Furthermore, by automating and executing this process at the point of check-in, developers will receive immediate feedback about a break and be in the best position to fix it quickly, when the relative cost to fix a bug is at its lowest.

Toad's Code Tester for Oracle automates the process of testing PL/SQL programs by identifying bugs and verifying program correctness in a fraction of time that would otherwise be required (see Figure 2). Code Tester helps development teams:

- **Guarantee code quality.** Because developers can create representative test cases automatically, they can test more often and more thoroughly to ensure the correct function of their code.
- **Reuse test cases.** Test cases are stored in a repository so they can be reused for later regression testing.
- **Work more productively.** Code Tester for Oracle is integrated with Toad, so developers can easily create new unit tests as a part of their debugging activity.
- **Keep test definitions in sync as code changes.** Code Tester for Oracle includes a code evolution feature to ensure that unit test definitions are kept in sync with code changes.

Get agile by using automated testing to provide a safety net for accelerating database deployments.



```
            /* Exception outcomes trapping for incorrect invocatior
EXCEPTION
WHEN OTHERS THEN
teardown;qcto#report_result(result_guid_in => qu_result_xp.curre
            END Q##TEST_CASE_2;
            BEGIN
            qu_result_xp.mark_start ('{1FA307BE-2DCB-4F16-ABAA-C788
            qu_result_xp.set_current_ut ('{1FA307BE-2DCB-4F16-ABAA-
            if g_pre_ut_hook_exists
                  then
                      QU_CONFIG.MY_CODETESTER_HOOK ('PRE_UNIT_TEST',
            DECLARE
    TYPE order_t IS TABLE OF PLS_INTEGER INDEX BY PLS_INTEGER;
    l_exec_order order_t;
            PROCEDURE run_one (seq_in IN PLS_INTEGER) IS BEGIN
            CASE seq_in
            WHEN 1 THEN  IF NOT qcto#skip_tc ('{6DF015DE-1915-45C5-
            WHEN 2 THEN  IF NOT qcto#skip_tc ('{02E0AC95-492B-4392-
            END CASE; END run_one; BEGIN
            /* Alphabetical execution order, just copy to driver cc
            l_exec_order(1) := 1;
            l_exec_order(2) := 2;
            FOR indx IN 1 .. l_exec_order.COUNT LOOP run_one (l_exe
            END;
            IF qu_result_xp.profiling_enabled THEN qu_result_xp.stc
            if g_post_ut_hook_exists
                  then
                      QU_CONFIG.MY_CODETESTER_HOOK ('POST_UNIT_TEST'
            qu_result_xp.mark_end ('{1FA307BE-2DCB-4F16-ABAA-C788B7
            EXCEPTION WHEN OTHERS THEN
            IF qu_result_xp.profiling_enabled THEN qu_result_xp.stc
            qcto#report_result (result_guid_in => qu_result_xp.curr

            IF qu_result_xp.rolling_back_after () THEN ROLLBACK;  E
            END Q##BETWNSTR;
            END "Q##BETWNSTR";

/
```

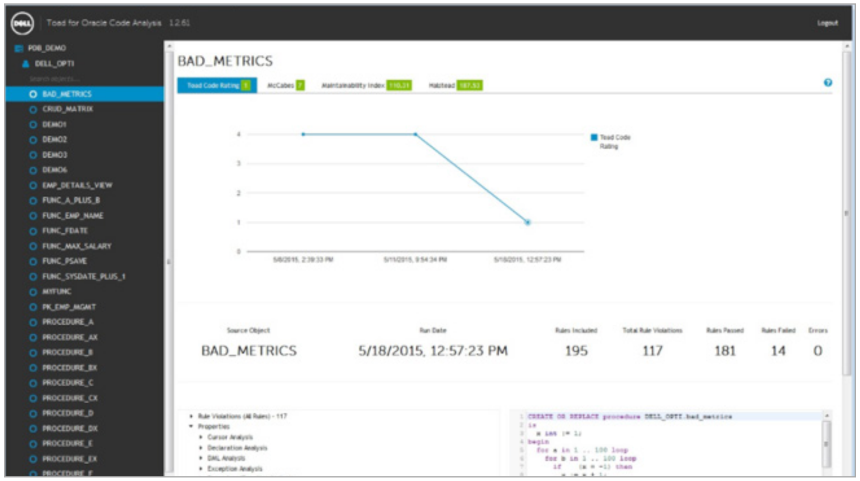
Figure 2. Toad's Code Tester for Oracle automates the testing of PL/SQL programs.

Ensure quality with standards

Database developers stand to gain from applying static code analysis after unit testing, perhaps even more than their application development peers. Given the extreme pressure and tight deadlines of the database development cycle, many teams rush the code review process or skip it entirely. Automating the process at the database development level can reduce the time the team spends and increase overall consistency of code reviews. Automated code review can check against rules written to help enforce company standards and improve quality, performance, maintainability, security and flow.

Toad's Code Analysis provides automated code review based on a rules engine to determine whether code meets company standards and follows industry best practices (see Figure 3). Consolidated, historical reporting of Code Analysis reviews is available with Toad Intelligence Central. Code Analysis helps development teams:

- **Consistently follow best practices for code quality.** Code Analysis ensures that developers working in a team environment are consistent in making code changes. It also ensures that they use standard processes for making changes, eliminating backdoor methods that could admit sub-standard code.
- **Ensure that code is production-ready.** Automated code review enables developers and managers to make more accurate decisions about code readiness, thereby reducing disruption and minimizing unplanned development cycles.



Source Object	Run Date	Rules Included	Total Rule Violations	Rules Passed	Rules Failed	Errors
BAD_METRICS	5/18/2015, 12:57:23 PM	195	117	181	14	0

```
CREATE OR REPLACE procedure DELL_OPTI.bad_metrics
is
  a int := 1;
begin
  for a in 1 .. 100 loop
   for b in 1 .. 100 loop
     if   (a = -1) then
```

Figure 3. Toad's automated code review provides insight into code quality.

- **Improve agility.** Automation speeds the code review process, making each development cycle better and faster.
- **Gain better visibility.** A centralized dashboard and analytics provide managers with the insight they need into code quality.

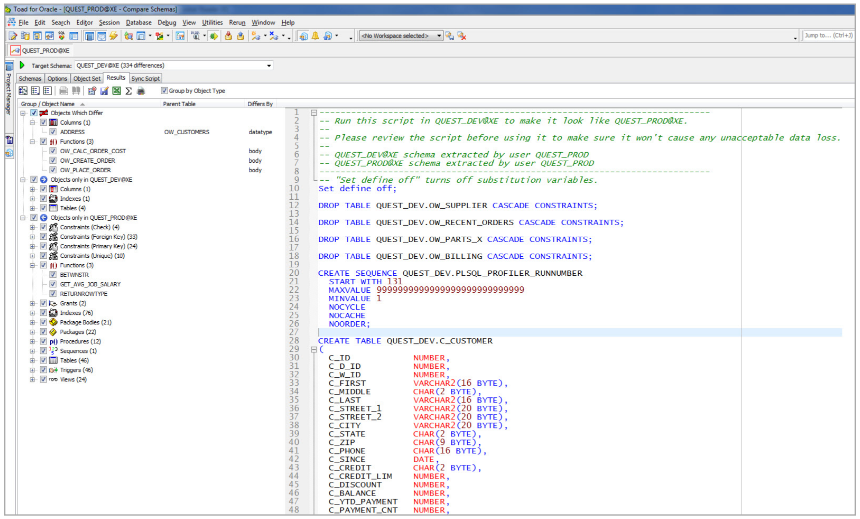
> Get agile by implementing automated code analysis to reduce time and increase overall consistency of code reviews.

Deploy faster

To reduce the risk of data loss and other mishaps, the database development path usually includes a DBA stop in the deployment stage for a review of code changes before they go into production.

DBAs and developers managing the path to deployment can use automated tools to collect all the queued changes that have passed regression tests and static code analysis, compare them with the production environment, and generate the scripts to commit them. Not only can this improve DBA efficiency and shorten the development cycle, but it can also ensure that all of a project's changes make it into production. These tools can help:

- **Compare and sync schemas.** The task of collecting changes and writing ALTER scripts is tedious and highly error-prone. Automation solves both of those problems. With push-button ease, Toad's


Toad for Oracle - [QUEST_PROD@XE - Compare Schemas]
File Edit Search Editor Session Database Debug View Utilities Rerun Window Help
<No Workspace selected>
Jump to... (Ctrl+J)
QUEST_PROD@XE
Project Manager
Schemas
Options
Object Set
Results
Sync Script

Target Schema: QUEST_DEV@XE (334 differences)

☑ Group by Object Type

Group / Object Name	Parent Table	Differs By
☑ Objects Which Differ		
☑ Columns (1)		
☑ ADDRESS	OW_CUSTOMERS	datatype
☑ Functions (3)		
☑ OW_CALC_ORDER_COST		body
☑ OW_CREATE_ORDER		body
☑ OW_PLACE_ORDER		body
☑ Objects only in QUEST_DEV@XE		
☑ Columns (1)		
☑ Indexes (1)		
☑ Tables (4)		
☑ Objects only in QUEST_PROD@XE		
☑ Constraints (Check) (4)		
☑ Constraints (Foreign Key) (33)		
☑ Constraints (Primary Key) (24)		
☑ Constraints (Unique) (10)		
☑ Functions (3)		
☑ BETWNSTR		
☑ GET_AVG_JOB_SALARY		
☑ RETURNROWTYPE		
☑ Grants (2)		
☑ Indexes (76)		
☑ Package Bodies (21)		
☑ Packages (22)		
☑ Procedures (12)		
☑ Sequences (1)		
☑ Tables (46)		
☑ Triggers (46)		
☑ Views (24)		

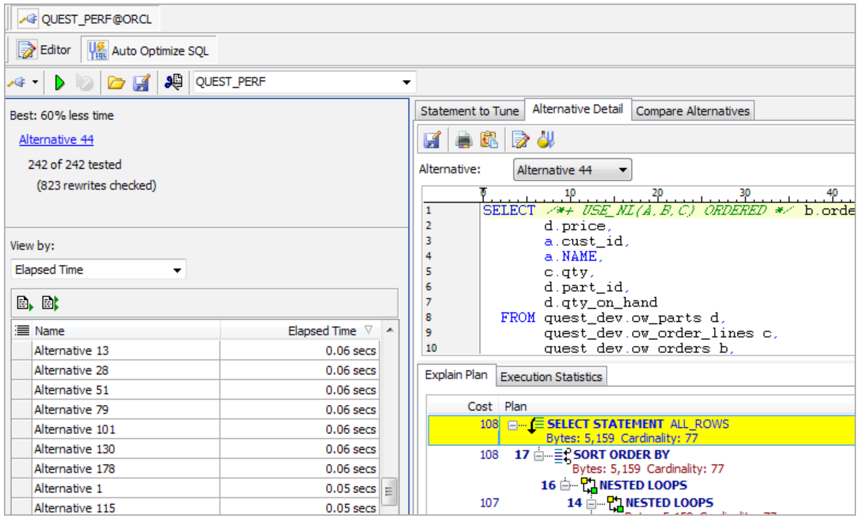
```
-------------------------------------------------------------------------
-- Run this script in QUEST_DEV@XE to make it look like QUEST_PROD@XE.
--
-- Please review the script before using it to make sure it won't cause any unacceptable data loss.
--
-- QUEST_DEV@XE schema extracted by user QUEST_PROD
-- QUEST_PROD@XE schema extracted by user QUEST_PROD
-------------------------------------------------------------------------
-- "Set define off" turns off substitution variables.
Set define off;

DROP TABLE QUEST_DEV.OW_SUPPLIER CASCADE CONSTRAINTS;

DROP TABLE QUEST_DEV.OW_RECENT_ORDERS CASCADE CONSTRAINTS;

DROP TABLE QUEST_DEV.OW_PARTS_X CASCADE CONSTRAINTS;

DROP TABLE QUEST_DEV.OW_BILLING CASCADE CONSTRAINTS;

CREATE SEQUENCE QUEST_DEV.PLSQL_PROFILER_RUNNUMBER
  START WITH 131
  MAXVALUE 9999999999999999999999999999
  MINVALUE 1
  NOCYCLE
  NOCACHE
  NOORDER;

CREATE TABLE QUEST_DEV.C_CUSTOMER
(
  C_ID             NUMBER,
  C_D_ID           NUMBER,
  C_W_ID           NUMBER,
  C_FIRST          VARCHAR2(16 BYTE),
  C_MIDDLE         CHAR(2 BYTE),
  C_LAST           VARCHAR2(16 BYTE),
  C_STREET_1       VARCHAR2(20 BYTE),
  C_STREET_2       VARCHAR2(20 BYTE),
  C_CITY           VARCHAR2(20 BYTE),
  C_STATE          CHAR(2 BYTE),
  C_ZIP            CHAR(9 BYTE),
  C_PHONE          CHAR(16 BYTE),
  C_SINCE          DATE,
  C_CREDIT         CHAR(2 BYTE),
  C_CREDIT_LIM     NUMBER,
  C_DISCOUNT       NUMBER,
  C_BALANCE        NUMBER,
  C_YTD_PAYMENT    NUMBER,
  C_PAYMENT_CNT    NUMBER,
```

Figure 4. Toad simplifies and automates schema comparisons.

Compare and Sync utility simplifies the process of generating the schema change script, saving time and reducing the chances of human error due to improper sequencing or forgotten changes (see Figure 4).

- **Proactively tune SQL.** Even high-quality code may run into performance issues when deployed to production. Toad's SQL tuning capability proactively and automatically identifies the best way for SQL to function at peak performance prior to deployment. This minimizes risks in production, when a DBA may have to tune SQL on the fly, and post-production, after inefficient performance has already impacted the business (see Figure 5).

  Dell SQL Optimizer for Oracle offers even more advanced tuning options. Rather than tuning a single piece of code, you can optimize batches of code in a single operation. In addition, you can determine which index combinations would work best across the entire batch. Because of variations in the optimizer, SQL Optimizer's Impact Analysis feature can quickly identify code that may regress in the production environment.



Best: 60% less time

Alternative 44

242 of 242 tested

(823 rewrites checked)

View by: Elapsed Time

Name	Elapsed Time
Alternative 13	0.06 secs
Alternative 28	0.06 secs
Alternative 51	0.06 secs
Alternative 79	0.06 secs
Alternative 101	0.06 secs
Alternative 130	0.06 secs
Alternative 178	0.06 secs
Alternative 1	0.05 secs
Alternative 115	0.05 secs

Alternative: Alternative 44

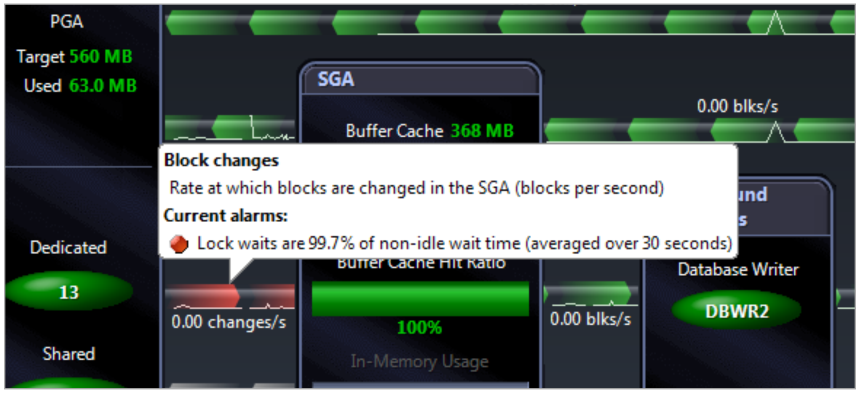
```
SELECT /*+ USE_NL(A,B,C) ORDERED */ b.orde
         d.price,
         a.cust_id,
         a.NAME,
         c.qty,
         d.part_id,
         d.qty_on_hand
  FROM   quest_dev.ow_parts d,
         quest_dev.ow_order_lines c,
         quest_dev.ow_orders b,
```

Cost	Plan
108	SELECT STATEMENT ALL_ROWS Bytes: 5,159 Cardinality: 77
108	17 SORT ORDER BY Bytes: 5,159 Cardinality: 77
	16 NESTED LOOPS
107	14 NESTED LOOPS

Figure 5. Toad automatically identifies problematic SQL so developers can proactively tune it, preventing costly problems in production.

- **Ensure scalability.** Even code that has gone through quality and tuning checks can sometimes bog down when user load is placed upon the environment. Benchmark Factory integrates with Toad so developers can easily check whether their code will scale properly for thousands of concurrent users.

- **Quickly identify, diagnose and resolve issues.** Before deploying code to production, you can analyze the workload in test using Spotlight on Oracle for diagnostics (see Figure 6). Spotlight provides a graphical representation of the Oracle database architecture, so



Block changes
Rate at which blocks are changed in the SGA (blocks per second)
Current alarms:
● Lock waits are 99.7% of non-idle wait time (averaged over 30 seconds)

Figure 6. Spotlight alerts you to problems in the test environment and in production.

you can quickly spot emerging issues. Moreover, Spotlight provides context-sensitive guidance that gives your DevOps teams the ability to quickly investigate and resolve performance inefficiencies. And should there be any issues after the deployment, Spotlight can help you quickly tackle them as well.

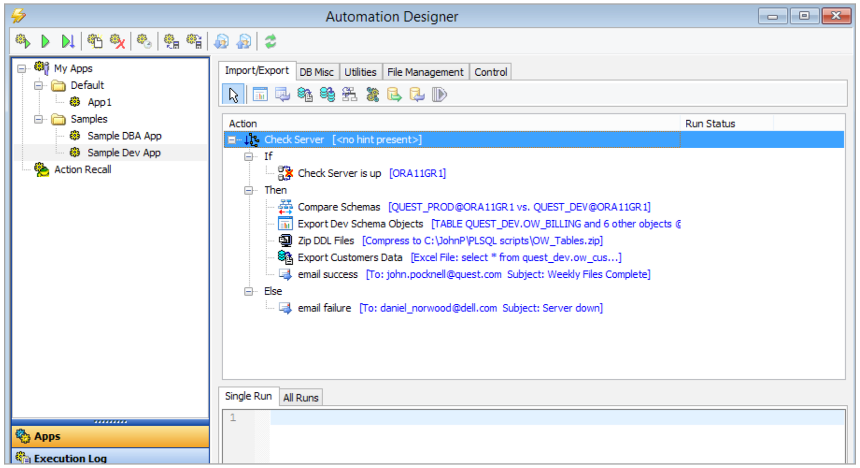
Automate everything

As your organization moves away from manual processes and toward automated tools, database development cycles will start shrinking and your teams will begin to realize the promise of agile. Using software tools in a piecemeal fashion each step along the way is faster than purely manual processes, but it provides only incremental improvements.

> Get agile with automated staging to improve DBA efficiency, shorten the development cycle and ensure that all changes make it into production.

> Get agile by creating fully automated workflows that apply conditional logic to groups of actions.

Automating frequent and repetitive tasks using Toad further reduces the risks associated with human error, while dramatically improving user productivity. Toad gives you the power to automate discrete tasks, in addition to creating fully automated workflows by applying conditional logic to groups of actions. What's more, all of this can be controlled remotely from an external process, making it easy to integrate your automated workflows into other applications and services (see Figure 7).



```
Check Server   [<no hint present>]
  If
    Check Server is up   [ORA11GR1]
  Then
    Compare Schemas   [QUEST_PROD@ORA11GR1 vs. QUEST_DEV@ORA11GR1]
    Export Dev Schema Objects   [TABLE QUEST_DEV.OW_BILLING and 6 other objects @
    Zip DDL Files   [Compress to C:\JohnP\PLSQL scripts\OW_Tables.zip]
    Export Customers Data   [Excel File: select * from quest_dev.ow_cus...]
    email success   [To: john.pocknell@quest.com  Subject: Weekly Files Complete]
  Else
    email failure   [To: daniel_norwood@dell.com  Subject: Server down]
```

Figure 7. Toad enables you to easily automate a variety of tasks.

It's time to learn more about Toad

Dell Software's Toad Development Suite for Oracle offers a set of guiding principles based on a sound methodology of best practices. When coupled with our Toad Intelligence Central server technology, Toad Development Suite gives you the power to implement consistent and repeatable database development processes. This helps your organization become more agile and deliver higher-functioning, easy-to-maintain code up to 40 percent faster than comparative tool sets. Extensive automation and collaboration functionality makes it easy to blaze through development cycles and minimize risks using agile methodology for database development.

The Toad Development Suite for Oracle consists of three integrated products:

Toad for Oracle Xpert Edition

- Use Toad for Oracle, the world's leading development tool, to build, edit and maintain your code.
- Pinpoint inefficient SQL and validate database code with the SQL Optimizer component to ensure the best possible performance.
- Use the Auto Optimize feature to easily improve SQL code directly within the Toad Editor.

Code Tester for Oracle

- Perform practical and thorough functional code testing for PL/SQL without lengthy test scripts.
- Run tests as you develop code or as regression tests before the code enters production, reducing the risk of costly bugs and functional errors over years of use and maintenance.

Benchmark Factory for Databases

- Conduct repeatable and measurable testing for performance and scalability.
- Replay database workload to simulate production activity levels prior to deployment.

With the Toad Development Suite for Oracle, you can ensure that the code you deploy is tested, functions properly and is optimized. It allows you to:

- Automate frequent and repetitive tasks and make the development organization more productive.
- Successfully deploy high-quality applications that perform reliably in production, meet user requirements and ensure that service levels are met.
- Lower the total cost of ownership for new applications by reducing the impact of inefficient code on productivity, future development cycles, performance and availability.

To learn more, and download a free 30-day trial, visit the Toad Development Suite for Oracle page.

About Dell Software

Dell Software helps customers unlock greater potential through the power of technology — delivering scalable, affordable and simple-to-use solutions that simplify IT and mitigate risk. The Dell Software portfolio addresses five key areas of customer needs: data center and cloud management, information management, mobile workforce management, security and data protection. This software, when combined with Dell hardware and services, drives unmatched efficiency and productivity to accelerate business results.

If you have any questions regarding your potential use of this material, contact:

Dell Software
4 Polaris Way
Aliso Viejo, CA 92656
www.dellsoftware.com

Refer to our Web site for regional and international office information.

The information in this document is provided in connection with Dell products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Dell products. EXCEPT AS SET FORTH IN DELL'S TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, DELL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DELL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF DELL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Dell makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Dell does not make any commitment to update the information contained in this document.