

The White Papers

**Effective Application Source Code SQL Analysis
For Change Impact, Performance and Quality**

By Quest Software

Effective Application Source Code SQL Analysis for Change Impact, Performance and Quality

By Quest Software

Introduction

A large number of sites around the world are working harder and harder with their database applications to improve operational efficiency and to gain a competitive advantage in their industry. The rate of change at these sites has accelerated substantially since IT professionals feel more pressure to embrace new technologies and take advantage of the Internet to expand their business. There is often pressure on the IT team to get new releases into production as soon as possible.

The ultimate aim of IT management is to implement the new releases and to eliminate the risk of things going wrong. The challenge is that this must be done at the lowest possible cost, since these application releases represent a substantial investment for an organization. It is crucial that the investment is protected from all angles.

Impact Analysis Challenges

In order to stay ahead of the competition, e-business demands that applications be developed, deployed, and updated faster than ever before, regardless of the risk involved with uncontrolled changes. This rapid change environment can compound several already painful conditions:

- When database and code changes are made without a clear understanding of the effects, important business processes may be severely compromised and fail.
- When a database or application component is modified, manually identifying all affected code is difficult and time consuming.
- Applications often have single database columns referenced in many different places throughout the source code, which is often written in many different languages. Finding and editing the affected code can be time-consuming and error prone.
- It's difficult to implement change requests without adversely affecting other dependent code. Things like critical queries, reports and applications may fail due to incompatibilities between the application code and the database structures.
- Capturing source code and database information in order to make structural database changes more accurately and efficiently is challenging. It is difficult to implement auditing and documentation for source code performance and quality checks.

Performance Analysis

Poor application performance can often be traced to SQL in the source code that is not tuned to take advantage of existing indexes or may require a new index. SQL tuning often requires the creation, deletion or modification of indexes. However, an index that might speed up the performance of a specific SQL query may adversely impact another.

DBAs and developers responsible for identifying and correcting performance bottlenecks cannot easily pinpoint SQL in the source code that may benefit from a new index without executing it or performing individual explains. Application performance can significantly be improved if unused indexes can be identified in the source code.

DBAs often have to react to application performance issues by tuning SQL in production. The need to understand what other applications are dependent on a particular SQL statement is critical. Additionally, when implementing SQL tuning advice, it is often difficult to determine what other applications will be affected and to ensure that no unforeseen problems will arise.

Testing and Quality Assurance

Test and QA analysts today must deal with many diverse issues when determining what application components are affected by a change and therefore need to be examined and tested.

Many times it is a challenge just to generate an up-to-date dependency list for an application, showing what database or source code objects are affected by a proposed change to a SQL statement. The ramifications of a simple database or source code change to other dependent applications or components can be severe. In some cases, analysts may not be aware of off-line queries and other dependent shared objects that may need to be rewritten or tested.

With the rush to deliver new and enhanced functionality, testing and documentation often suffer. In order to successfully plan different test scenarios there needs to be an automated process of identifying and documenting SQL-related quality issues, and obtaining critical SQL dependency analysis information.

The issues mentioned above are extremely important to the process of fully understanding the relationship between the database objects and application code and how change affects both environments. In many environments, the answers to these critical questions involving the impacts of a change are not immediately available, if at all. Often the side effects of a database change are not known until the application undergoes system integration testing. Or, worse yet, the changes are discovered to be incompatible when migrated to a production environment.

At many sites these challenges are addressed through a great deal of manual effort which are time-consuming, and can be highly error-prone.

What's needed to address these challenges for DBAs, developers, testers and QA analysts, is a way to ...

- Analyze the contents of a release and identify what other objects are affected
- Identify and correct problem SQL in the source code before customers are affected by it
- Step through typical database change scenarios and have the impact dependencies identified
- Automatically generate documentation for different changes and source code performance and quality issues
- Collect, view, and edit the source code and database information from one environment

The Solution

Products like SQL Impact from Quest Software, Inc. offer a solution to these challenges. SQL Impact, a component of Quest Software's SQLLab Vision™, provides impact analysis, documentation and auditing for managing the dependencies between database objects and application source code. It precisely determines the components that will be affected by structural database changes (schema changes) and drills down to the lines of source code affected.

SQL Impact goes beyond impact analysis to provide comprehensive documentation on the access pattern of your application. In addition, it performs automated quality assurance on the SQL in your applications to look for problems, such as references to columns or tables that no longer exist or inefficient join conditions.

The product also provides advanced impact analysis for index changes. This helps you identify the effect that index changes will have on database performance.

SQL Impact allows you to:

- Find and scan the SQL embedded in the source code and database
- Identify specific SQL anomalies and areas for quality and performance improvements
- Generate impact analysis information by comparing the database's data dictionary with SQL Impact's catalog
- Create extensive documentation through a host of reports and queries that enhance communication and understanding of database and source code environments
- Rapidly and accurately assess the scope of a change and to determine what components may need to be re-tested.
- Proactively identify performance problems through extensive explain analysis of all SQL in an application without executing a single line of code
- Integrate with Quest Software's SQL Navigator, TOAD, Schema Manager, and SQLab Vision for comprehensive development, deployment, and production support.

Product Architecture

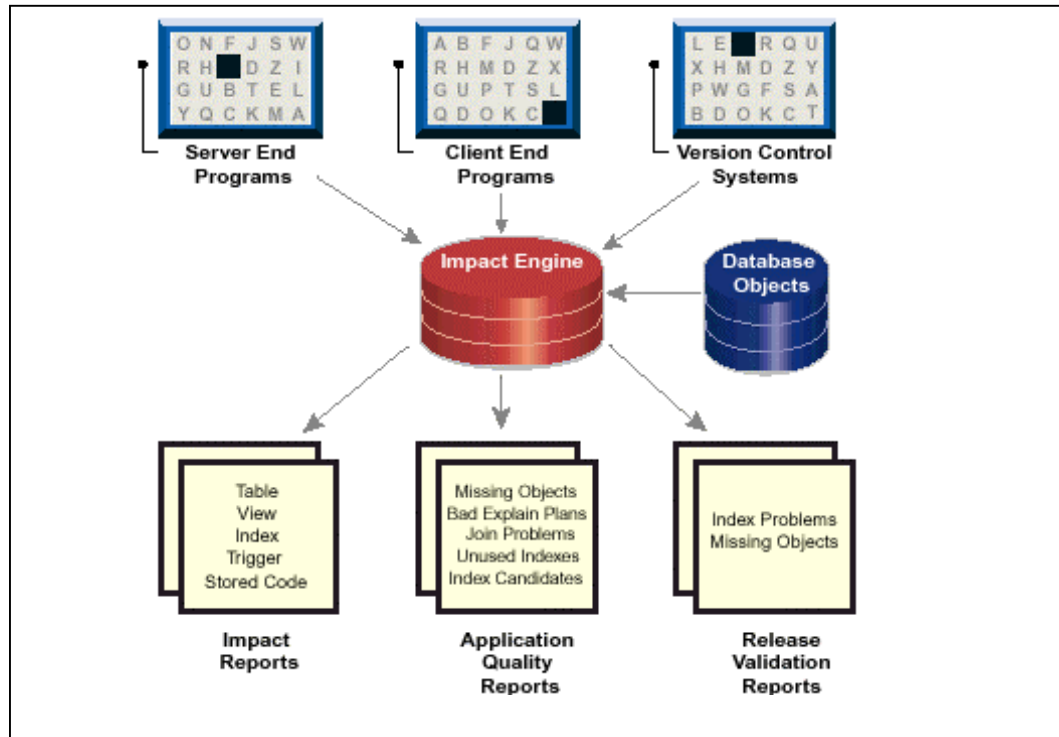
SQL Impact installs on a Windows 95/98/NT/2000 client and connects over SQL*Net or Net8 to manage Oracle (7.x – 8.i) on any server platform. Multiple Oracle databases can be accessed from a single client.

The product uses its own repository to store impact and performance analysis information and to catalog database access patterns of the application SQL. The repository is a set of Oracle tables built by SQL Impact and can be stored on any server regardless of the databases being analyzed for impact.

Source Code SQL Collection

SQL Impact's catalog provides DBAs and developers a common repository to identify change impacts, performance anomalies and to quickly determine application SQL access patterns and usage.

Source code scanning and cataloging can be run interactively from the client or scheduled to run in the background on the server for hands-off updates. SQL Impact can interface with many popular schedulers via the command line interface on both Windows and selected Unix platforms.



SQL Impact Analysis Process

SQL Impact can find SQL code in the following language formats:

- Binary executable files
- C (standard) and Pro*C
- C header files
- COBOL and Pro*COBOL
- Delphi
- Java and Oracle SQLJava
- Oracle Developer (to 6.0)
- Oracle Data Dictionary
- Perl
- PL/SQL libraries
- Pro*Fortran
- PowerBuilder applications
- SAS applications
- Shell scripts
- SQL Windows
- SQL*Forms 2.5, 3.0
- SQL*Loader
- SQL*Plus scripts
- SQL*Report
- SQL*ReportWriter
- SQR Reports
- Team Developer
- Visual Basic
- VMS DCL scripts

Comprehensive Change Impact Analysis

To help DBAs, Developers, Test and QA Analysts obtain a comprehensive understanding of a change to the database and source code components, SQL Impact provides wizards to walk through “what if” scenarios that identify specific objects affected:

- PL/SQL Stored Code impact
- Table impact
- View impact

- Identify potentially unused indexes, table join analysis and new index candidates
- Register SQL from SQLab Xpert collections and repository
- Search SQL Impact catalog for SQL captured by SQLab Xpert in the SGA

Quest Software's StealthCollect technology collects SQL statements and performance indicators from Oracle databases and host operating systems at sub-second intervals. The data is collected non-intrusively, with no added overhead on the target database and minimal overhead on the host system. StealthCollect runs 24x7, enabling you to identify and resolve problems in the past and collect performance information automatically.

Release Deployment and ERP Customizations

SQL Impact helps you to avoid the risk of release objects being missing and have more confidence that everything that interacts with a release has been analyzed.

Quest Software's integration between SQL Impact and Schema Manager provides DBAs and developers a means to quickly report affected source code contained in a database release package:

- Detect source code programs that access a specific database object in a release
- View/Edit specific SQL in programs and stored code affected
- Detect missing objects
- Provide information for sharing release with other application groups
- Create a list of all database and source code objects that are related to a trigger
- Create a report of source code object relationships with client to server side objects
- Create a list of differences between database schemas and determine the 'delta impact' on the source code

While the "Delta Impact Wizard" simplifies the discovery of affected source code and database objects for a Schema Manager release, the same overall process can also be applied in the absence of Schema Manager. By analyzing SQL scripts used for deployment, or running impact analysis reports against a pre-deployment database staging area, SQL Impact can show direct and indirect impacts to source code and database objects included in the release as well.

Local customizations applied to ERP environments can be cataloged with SQL Impact to provide more extensive patch dependency information for Oracle Forms and Reports, Peoplesoft SQR, and database stored components.

Reporting

SQL Impact provides formal documentation through a variety of reports:

- Object analysis such as hint usage by table and distinct database objects accessed
- Anomalies such as Cartesian product joins and invalid data types
- Performance analysis such as candidate columns for indexing and unused tables and indexes
- Source code quality audits
- Catalog and program where used information

These reports are directly viewable and can be output in many popular formats such as HTML, Excel, Word, Adobe, and more. Additionally, the extensive reporting capability in SQL Impact provides a way to quickly scope the time required to implement a change as well as generating comprehensive information for application test plans.

Summary

Successful impact, performance, and quality analysis when changing indexes, tables, views, triggers, PL/SQL or programs requires thorough information gathering. Data is needed from database objects, stored code, client end programs, server end programs and programs stored in configuration management tools. Effectively, you need to be able to “go where the SQL is.”

SQL Impact offers many benefits to accomplish this goal:

- Provides a comprehensive way to understand database and source code relationships
- Simplifies discovery of problem SQL
- Accurately determines scope of changes
- Improves performance by finding source code SQL problems
- Saves DBA and developer time
- Provides QA with the information to ensure better quality
- Gives management the tools they need for productivity improvements

About Quest

Quest Software, Inc. is a leading provider of performance management solutions designed to maintain the integrity of mission-critical business transactions and maximize the performance of enterprise applications. Our solutions address the needs of today's 24x7x365 businesses where demands on the information technology infrastructure are high and tolerance for downtime is low. The Internet has propagated the expectation of instant access to information, and Quest delivers the solutions necessary to meet this demand.

Founded in 1987, Quest Software now helps more than 100,000 users achieve the best possible performance from their enterprise systems so the end user experience is a positive one. Based in Irvine, California, Quest Software has offices worldwide and over 1,000 employees. For more information, visit www.quest.com.