
Optimising Your Database Performance... The Easy Way

Tanya Hardy

Database Consultant

Synergy International Limited

Introduction

Customers—and managers of the companies who serve them—demand instant, fast access to information. This means that the database administrators (DBAs) who are responsible for this information must optimise the performance and availability of databases and anticipate and head off bottlenecks before they happen. These DBAs must be familiar with multiple database platforms, operating systems, and hardware platforms—with little to no training.

Maintaining database performance and availability is a challenging task. There are many events that can impact database performance. Once these events occur, it is important to diagnose and resolve these issues quickly and easily. Most importantly, it is important to be proactive and notify the DBAs *before* these events actually occur and impact the user experience.

How do DBAs normally handle performance problems? Let's take a look.

Manual Optimisation of Database Performance

One of the most common reasons for poor database performance is un-tuned SQL.

There are many rules and guidelines that need to be followed when queries are being written. If these guidelines are not followed, performance issues may be impacted. Some of these guidelines are as follows:

- Avoid a full-table scan if it is more efficient to get the required rows through an index.
- Avoid using an index that fetches 10,000 rows from the driving table if instead you could instead use another index that fetches 100 rows and choose selective indexes.
- Choose the correct join order so you will join fewer rows to tables later in the join order.
- Set up the driving table to be the one containing the filter condition that eliminates the highest percentage of the table.
- Use untransformed column values.
- Do not use SQL functions in predicate clauses or WHERE clauses. Use function-based indexes where possible.
- Avoid mixed-type expressions.
- When using IN and NOT IN with a sub-query, try to rewrite it with WHERE (NOT) EXISTS as an alternative.
- Minimise the use of DISTINCT.
- Use equi-joins.
- Avoid using inequality operators such as '<>', '!='.
- Be careful when using 'IS NULL' and 'IS NOT NULL'.

When a SQL performance issue occurs in production, the DBA or developer attempting to address the issue does the following:

- Checks the SQL to make sure the rules listed above are being followed.
- Possibly rewrites the query.
- Re-runs EXPLAIN PLAN to see if a better access path is selected.

Typical Manual Tuning Steps

Here are the typical steps a DBA follows to tune SQL manually once he knows there are performance problems:

1. Locate the problem queries.
2. Run EXPLAIN PLAN.
3. Interpret the output.
 - What is the operation order?
 - Look at the full table scans.
 - Look at join operations.
4. Look at the indexes.
 - Are there too many?
 - Are there any not being used?
 - Is there no index?
5. Rewrite the query.
6. Run EXPLAIN PLAN.
7. Interpret the output.
 - What is the operation order?
 - Look at the full table scans.
 - Look at join operations.
8. Compare with the previous EXPLAIN PLAN(s).
9. Is it better? If not, repeat all of the steps again.

This process can take a while, depending on the tuning experience of the DBA or developer. To get the best performance, he would have to manually rewrite a query several different ways, run the query, and then write down statistics on how fast it ran. Did it improve? Did it improve enough? Was it fast enough? The DBA can spend many hours in trying to tune the SQL, and in the end, discover that it was an application design issue, not a SQL problem.

There are issues that go beyond rewriting a query. EXPLAIN PLAN can show that the query is using the indexes that it is supposed to, but the performance is still bad. The potential cause could be unbalanced indexes, which can greatly impact query performance. If there are a large number of deletes on their associated tables, there will be unbalanced indexes. These indexes need to be rebuilt on a regular basis.

Using Proactive and Diagnostic Tools to Optimise Database Performance

Some of the events that can impact database performance and availability are database configuration parameters, migrated rows, resource contention and, most important, un-tuned queries. Two types of tools are needed in order to ensure acceptable levels of performance and availability—proactive and diagnostic. The proactive tools alert DBAs to problems that are getting close to unacceptable levels, and the diagnostic tools analyse and pinpoint problem areas and provide advice as to how to resolve issues.

With the right tools, DBAs can substantially increase the performance and availability of their databases. BMC Software has several expert DBA tools that have built-in intelligence and automation. These tools can advise the DBA and make it easier for him/her to be effective and efficient.

Customer Case

A customer was experiencing performance issues at specific times during the day. The DBA was running scripts to attempt to capture the problematic SQL. After several days of trying to do this manually, the DBA was not able to find the issue. Then DBXray, SQL-Explorer, and Space Expert were brought in to the picture.

First he monitored user sessions with **DBXray** and noticed that there was a connection that was performing a great deal of physical I/O. He selected that session to view the SQL that was being executed, and selected the TUNE button. This took the SQL statement that was being executed by the session and pushed it to **SQL-Explorer**. The query was using NOT IN with a sub-query. The sub-query was not making use of available indexes.

SQL-Explorer created a rewrite of the query. To see how the rewrites improved performance, he then selected the SQL statistics option from SQL-Explorer. When looking at the number of rows in the tables, he should have seen a greater improvement. Improvement was good, but more analysis was needed.

Next a “Find Problems” job from **Space Expert** was executed. Much to his surprise, several indexes had experienced many deletes. The two indexes that were listed were ones that were needed to satisfy the query. Space Expert rebuilt the index and he ran the trials again. The query response time was *dramatically improved by 85% - 100%*.

This is an interesting case where one layer of problems was peeled back and then another layer was exposed. An entry-level DBA with no tuning experience might struggle manually with the above problem for several days. An entry-level DBA could use DBXray, SQL-Explorer, and Space Expert and solve the problem *in less than an hour*.

To solve the problem in such a short time, the DBA followed these steps:

1. He noticed the problem in **DBXray** with the User Sessions Drilldown information. A lot of physical I/O indicates that there are some un-tuned—not optimised—SQL statements and this indicates to use **SQL-Explorer**. SQL-Explorer automatically considers all of the rules in optimising a query and gives the DBA several different rewrites of the SQL statement.

2. The DBA used SQL-Explorer and ran the Get Actual SQL Stats. It showed all of the possible rewrites, the elapsed time it took to run, the number of buffer gets, CPU time, physical reads, and number of rows fetched. It put a green check mark next to the statement that had the best statistics of the rewrites. There were very large tables in this example. The DBA saw some improvement with the rewritten statement, but should have seen a great improvement by the rewrite that SQL-Explorer generated. It added a WHERE clause to the sub-query, thereby making use of available indexes. The NOT IN was changed to a NOT EXIST (this is one of the above rules for creating a query) and this helped improve performance even further. NOT IN can suppress indexes even when they are there so now he has ensured that that is not a problem.

At this point, some DBAs might give up and might say that they tuned the SQL statements to be as fast as possible. There could still be object-related issues affecting performance such as fragmented tablespaces, migrated rows, and indexes that needed rebuilding.

3. With the click of a mouse button, the DBA moved over to **Space Expert** and ran Find Problems and discovered the index problem. Space Expert rebuilt the index, he ran the now optimised query and realised an *85% - 100% improvement* in performance over the original query!

Tools for Performance Optimisation

DBXray for Oracle

DBXray is the personal productivity tool that allows database professionals of all experience levels to manage and master the everyday challenges of database administration from any PC with a common Internet browser - simply, quickly and effectively.



DBXray:

- Improves the performance and prevents outages of a database by identifying and eliminating database bottlenecks
- Delivers ready access from anywhere, anytime through a Web browser
- Provides detailed drilldown information on user session details, 24-hour database view, 10 data files with highest I/O hits, 4 top memory consumers, space management, delays caused by redo log space requests, locks blocking other sessions, transactions that must wait on a rollback segment for completion, waits and latches, and many other critical metrics
- Includes SmartDBA Cockpit for access—even remotely via the Web—to Oracle, DB2 UDB, and Microsoft SQL Server databases from one single console. DBXray, Space Expert, and SQL-Explorer plug in to SmartDBA Cockpit and the user can access them from this one single interface.

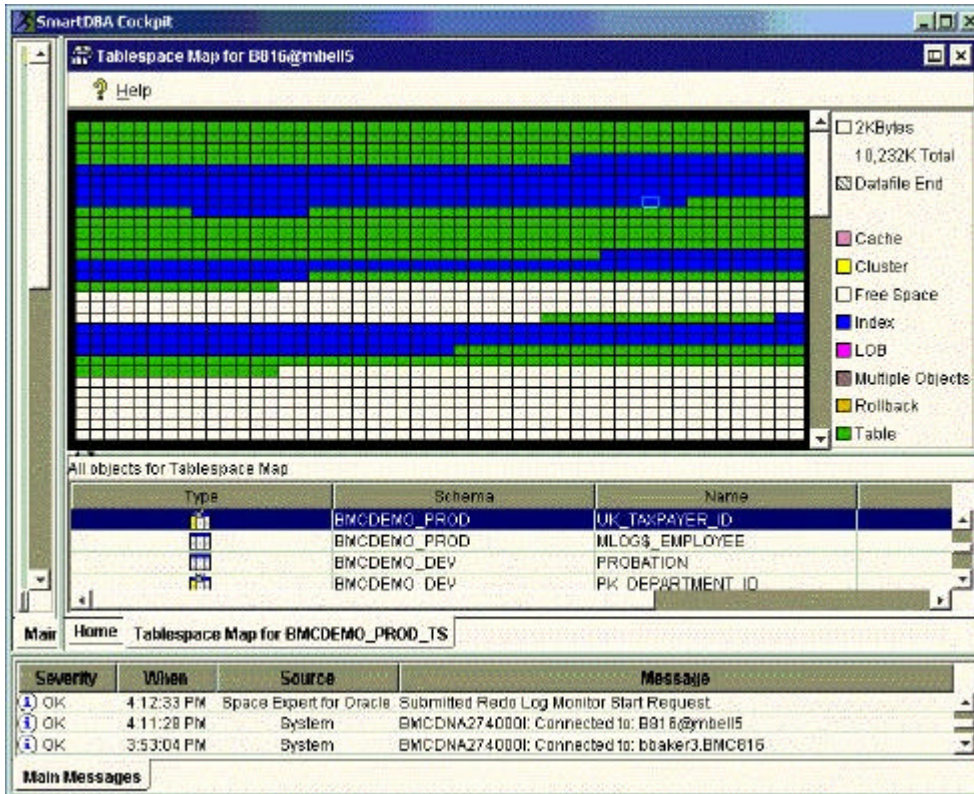
Space Expert for Oracle

Space Expert for Oracle can analyse a database, predict problems, and help a DBA fix them before they even happen. It provides information on when reorganisation is needed and can even do the reorganisation online.

Space Expert:

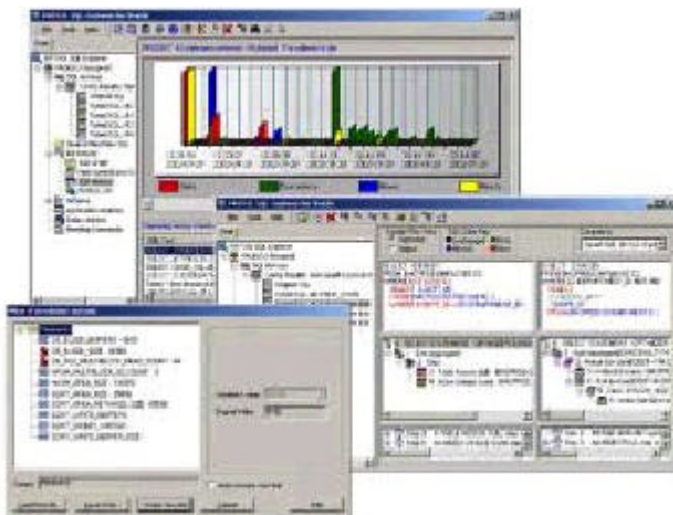
- Reorganises 24/7 databases online, without downtime, providing complete read and write access to database applications
- Avoids unplanned outages by identifying objects that have high severity problems and provides recommended actions
- Reduces the need for future reorganisations by performing powerful analysis of object condition, producing ideal settings that restore the object to peak performance

- Performs automated sophisticated scheduling, object selection, multi-processing, and full checkpoint restart capability
- Includes SmartDBA Cockpit and DBXray



SQL-Explorer for Oracle

SQL-Explorer offers the most advanced SQL tuning technology in the industry, allowing DBAs to improve business application performance by proactively analysing, diagnosing, managing and tuning databases and applications offline while preserving application and business availability. It lets DBAs collect data non-intrusively, analyse SQL statements, optimise a database, and do “what-if” analysis with exclusive simulation technology without impacting your production environment.



SQL-Explorer:

- Prevents production interruption by using a non-intrusive high-speed data collector for capturing and saving SQL for later analysis

- Extends DBA resources and improves application performance by automatically providing expert tuning recommendations
- Reduces tuning time by displaying all relevant tuning information in an intuitive format
- Tunes SQL before releasing application
- Optimises SQL that is tuned
- Provides indexing suggestions based on SQL execution.
- Includes SmartDBA Cockpit and DBXray

Typical Tuning Steps Using DBXray, Space Expert, and SQL-Explorer

When the DBA uses no tool, he typically does not know there are performance problems until after they have already occurred. With DBXray, he can instantly spot a performance problem on the database-at-a-glance screen. he can also be proactive and head off performance problems with the information provided in the monitors on the DBXray screen. Once he figures out that he has performance problems—or that he will have performance problems if he doesn't take action now—he can use SQL-Explorer and Space Expert to optimise the performance of his database.

Here are the steps a DBA follows to tune SQL using SQL-Explorer and Space Expert.

1. Use SQL-Explorer which automatically and quickly:
 - Identifies queries that are affecting the performance of the database
 - Gathers history via the High Speed Data Collector with very low overhead
 - Runs EXPLAIN PLAN
 - Displays several suggested query rewrites and a plain English explanation of execution steps
 - Provides table definitions and defines indexes
 - Uses simulation technology to run trials on a Virtual Instance to compare execution statistics
 - a. No impact to production
 - b. Great for long running queries
 - Identifies the best query to use
 - Compares EXPLAIN PLAN of the re-written SQL with the original SQL and color codes the differences
2. Use Space Expert to determine if indexes are selective or not and whether the indexes need to be rebuilt.

Vendor Applications

When using vendor applications like SAP, Siebel, and Oracle Financials, the SQL in the application cannot be modified. However, major performance gains can be achieved by using SQL-Explorer to analyse index usage and create indexes based on specific usage patterns in your database. SQL-Explorer can also be used to identify poorly written SQL in these applications. Even though you can't change the SQL from SQL-Explorer due to the restrictions in those applications, you can use SQL-Explorer to identify problem SQL and then the application vendors can fix it.

Conclusion

DBAs can ease their workloads and ensure that their databases are available and tweaked for top performance by using DBXray, Space Expert, and SQL-Explorer - intelligent, automated, integrated DBA tools. These tools make DBAs' jobs easier and they also dramatically cut the time to solve and prevent performance problems, and they ensure that their databases are available and operating at peak performance.

About the Author

Tanya Hardy

Tanya Hardy is a Database Consultant with more than eight years of experience in the IT industry, mostly in the fields of Oracle database administration and development. She has worked in a wide range of environments, taking complete responsibility for the Oracle database component in architectural design, modelling, development, implementation, performance tuning and support.

Tanya particularly enjoys working in the support environment and is always interested in ways to improve her tuning and diagnostic skills. Tanya's experience with Oracle products encompasses the Oracle databases' versions from 7 to 9i, Oracle 9i Application Server, Oracle Forms and Reports, High Availability Oracle and many others.

Tanya has a BS degree in Computer Science and is an Oracle 8i Certified DBA.