Welcome to the world of fast Oracle SQL tuning with SQLTXPLAIN, or SQLT as it is typically called. Never heard of SQLT? You’re not alone. I’d never heard of it before I joined ORACLE, and I had been a DBA for more years than I care to mention. That’s why I’m writing this book. SQLT is a fantastic tool because it helps you diagnose tuning problems quickly. What do I mean by that? I mean that in half a day, maximum, you can go from a slow SQL to having an understanding of why SQL is malfunctioning, and finally, to knowing how to fix the SQL.

Will SQLT fix your SQL? No. Fixing the SQL takes longer. Some tables are so large that it can take days to gather statistics. It may take a long time to set up the test environment and roll the fix to production. The important point is that in half a day working with SQLT will give you an explanation. You’ll know why the SQL was slow, or you’ll be able to explain why it can’t go any faster.

You need to know about SQLT because it will make your life easier. But let me back up a little and tell you more about what SQLT is, how it came into existence, why you probably haven’t heard of it, and why you should use it for your Oracle SQL tuning.

What Is SQLT?

SQLT is a set of packages and scripts that produces HTML-formatted reports, some SQL scripts and some text files. The entire collection of information is packaged in a zip file and often sent to Oracle Support, but you can look at these files yourself. There are just over a dozen packages and procedures (called “methods”) in SQLT. These packages and procedures collect different information based on your circumstances. We’ll talk about the packages suitable for a number of situations later.

What’s the Story of SQLT?

They say that necessity is the mother of invention, and that was certainly the case with SQLT. Oracle support engineers handle a huge number of tuning problems on a daily basis; problem is, the old methods of linear analysis are just too slow. You need to see the big picture fast so you can zoom in on the detail and tell the customer what’s wrong. As a result, Carlos Sierra, a support engineer at the time (now a member of the Oracle Center of Expertise—a team of experts within Oracle) created SQLT. The routines evolved over many visits to customer sites to a point where they can gather all the information required quickly and effectively. He then provided easy-to-use procedures for reporting on those problems.

Carlos Sierra, the genius of SQLT, now spends much of his time improving SQLT code and adapting the SQLT code to new versions of the RDBMS. He also assists Oracle Tuning Performance engineers with SQL tuning through the medium of SQLT.
Why Haven’t You Heard of SQLT?

If it’s so useful, why haven’t you heard about SQLT? Oracle has tried to publicize SQLT to the DBA community, but still I get support calls and talk to DBAs who have never heard of SQLT—or if they have, they’ve never used it. This amazing tool is free to supported customers, so there’s no cost involved. DBAs need to look at problematic SQL often, and SQLT is hands down the fastest way to fix a problem. The learning curve may be high, but it’s nowhere near as high as the alternatives: interpreting raw 10046 trace files or 10053 trace files. Looking through tables of statistics to find the needle in the haystack, guessing about what might fix the problem and trying it out? No thanks. SQLT is like a cruise missile that travels across the world right to its target.

Perhaps DBAs are too busy to learn a tool, which is not even mentioned in the release notes for Oracle. It’s not in the documentation set, it’s not officially part of the product set either. It’s just a tool, written by a talented support engineer, and it happens to be better than any other tool out there. Let me repeat. It’s free.

It’s also possible that some DBAs are so busy focusing on the obscure minutiae of tuning that they forget the real world of fixing SQL. Why talk about a package that’s easy to use when you could be talking about esoteric hidden parameters for situations you’ll never come across? SQLT is a very practical tool.

Whatever the reason, if you haven’t used SQLT before, my mission in this book is to get you up and running as fast and with as little effort from you as possible. I promise you installing and using SQLT is easy. Just a few simple concepts, and you’ll be ready to go in 30 minutes.

How Did I Learn About SQLT?

Like the rest of the DBA world (I’ve been a DBA for many years), I hadn’t heard of SQLT until I joined Oracle. It was a revelation to me. Here was this tool that’s existed for years, which was exactly what I needed many times in the past, although I’d never used it. Of course I had read many books on tuning in years past: for example, Cary Millsaps’s classic *Optimizing Oracle Performance*, and of course *Cost-Based Oracle Fundamentals* by Jonathan Lewis.

The training course (which was two weeks in total) was so intense that it was described by at least two engineers as trying to drink water from a fire hydrant. Fear not! This book will make the job of learning to use SQLT much easier.

Now that I’ve used SQLT extensively in day-to-day tuning problems, I can’t imagine managing without it. I want you to have the same ability. It won’t take long. Stick with me until the end of the book, understand the examples, and then try and relate them to your own situation. You’ll need a few basic concepts (which I’ll cover later), and then you’ll be ready to tackle your own tuning problems. Remember to use SQLT regularly even when you don’t have a problem; this way you can learn to move around the main HTML file quickly to find what you need. Run a SQLT report against SQL that isn’t a problem. You’ll learn a lot. Stick with me on this amazing journey.

Getting Started with SQLT

Getting started with SQLT couldn’t be easier. I’ve broken the process down into three easy steps.

1. Downloading SQLT
2. Installing SQLT
3. Running your first SQLT report

SQLT will work on many different platforms. Many of my examples will be based on Microsoft Windows, but Linux or Unix is just as easy to use, and there are almost no differences in the use of SQLT between the platforms. If there are, I’ll make a note in the text.