



Carlos Sierra

YesSQL(T)

with Carlos Sierra and Mauro Pagano



Mauro Pagano

What is the history of SQLT? How did it evolve? Who contributed to its development? Why do I need it? How do I get it?

Carlos: SQLT, also known as SQLTXPLAIN, has been around for almost 16 years. On December 2, 2015, it celebrates its Sweet 16! At the beginning, its name was `coe_xplain.sql`, and it was a simple script to produce an execution plan for a given SQL text. Since its creation, it included metadata for tables and indexes related to the SQL of interest, in addition to the execution plan. Over time, `coe_xplain.sql` evolved into a more complex script. Then on October 21, 2002, it became SQLTXPLAIN. So I may say SQLTXPLAIN was born 13 years ago, but its predecessor `coe_xplain.sql` was already 3 years old.

SQLT evolved from a simple script with 350 lines of code to a complete tool with over 150K lines of code! Then there is SQLd360, which is the “new SQLT,” but I will let Mauro expand on that.

I was the original developer of SQLT, and when I left Oracle and went to work for the Accenture Enkitec Group (AEG) in August 2013, Mauro took over. But years before that, Mauro contributed many ideas, like the one about SQLT XPLORE. Abel Macias also contributed a lot early on, and now he maintains SQLT at Oracle. Mauro is currently also working for AEG.

Mauro: When Carlos left Oracle, I worked on including most of the 12c new features known at the time (not many clients were using 12c back then). I always joke, saying I spent most of my time fixing Carlos’s bugs, but truth be told, I probably added more than I fixed!

SQLd360 is a natural evolution of SQLT. This brand-new SQLd360 free tool encapsulates our current view on SQL tuning, and it has been built based on the lessons learned with SQLT, on the limitations we wanted to address, and on the approach we now take to SQL tuning.

Carlos: Mauro is very humble. I recognize SQLd360 as “The Next Generation,” while SQLT remains like the original “Star Trek.” Personally, I like this “next generation” better. Walt Disney said once, “We keep moving forward, opening new doors, and doing new things, because we’re curious and curiosity keeps leading us down new paths.” Mauro is now keeping the effort I started with SQLT moving forward with SQLd360. First he enhanced SQLT, now he works on a new and fresh remake.

I’m sure you have some special success stories of how you solved complex SQL tuning problems using a SQLT report.

Mauro: There are many! I remember one where a DBA walked into his office and found out everything was horribly slow because the execution plan of one of his massively long SQL statements had changed. It was one of those typical cases of “it’s so complex nobody wants to touch it, but it works so we are safe,” and it was taking 30 seconds instead of a few milliseconds! After a few hours trying to figure out what happened, the DBA decided to create a Service Request for Oracle, and I took it. We asked for SQLT, but the DBA was reluctant to install a tool in production. Fortunately, the issue was so urgent that he decided to go ahead and do it anyway. Once we got the SQLT files it took us 3 minutes to solve the issue! The automatic statistics-gathering job introduced a histogram on a `varchar2` column that had very repetitive leading characters (in 11.2, only the first 32 characters are considered when building a histogram), so the CBO moved from seeing millions of distinct values for that column to just a few thousand, big change in selectivity and thus a big change in the execution plan! The DBA was so surprised that he proactively installed SQLT anywhere he could, “just in case I’ll need it tomorrow.”

Anyway I think what makes SQLT a great tool is not how we were able to solve one specific case, either simple or complex, but the fact that in almost every success story, SQLT was there, playing either a small or big role, but it was there helping the engineer do his/her job in a more efficient way.

Carlos: One I dearly remember is a request to help a client on a recent 10g upgrade that caused havoc at this company, where a twin database kept producing different execution plans. Both databases were supposedly identical, yet one produced inconsistent execution plans, changing often and with erratic performance. The other “twin” database was more consistent. The problem required me to create the SQLT COMPARE method, which basically allowed us to identify the root cause. It had to do with column usage and the automatic generation of histograms using a very small sample size and producing an incorrect number of distinct values and missing buckets. There are many complex issues we solved with the help of SQLT, but this one was special since the original hypothesis for all of us was that both systems were identical—and they were indeed in terms of structure and data—but back on 10g we were not fully aware of the consequences of “auto” functions of the ever-changing DBMS_STATS package. SQLT provided the irrefutable proof that we needed at that time!

I am in awe. But, if SQLT is the godsend you say it is, why isn't it widely known and used? Is it just a matter of education and evangelization? (I hope this interview helps a little.) Or are there other barriers to adoption?

Carlos: I would have to disagree here. SQLT is widely used around the world. If you ever log a ticket with Oracle Support about a SQL performing poorly, there is a big chance you will be asked to show your issue with SQLT. Keep in mind that according to many engineers in Oracle Support, this SQLT tool saves over 50% of their time when they need to diagnose a SQL statement. Some engineers in Oracle Support even make SQLT a hard requirement in order to progress an issue! The alternative to not using a tool like SQLT is usually a lengthy and painful investigation.

Mauro: SQLT isn't magic; it doesn't do anything anybody with lots of time on his/her hands could not do. What I loved about SQLT during my Support days is that it defines kind of a communication standard between customers and Support. The advantage is twofold: on one hand, clients really simplify and speed up diagnostics collection, plus they make sure not to forget any piece the engineer might need. On the other hand, the Oracle Support engineers receive information in a consistent format so they can focus on the real problems instead of just trying to put pieces together. I remember one customer that was sending me info in Word docs, Excel docs, screenshots of execution plans, and extracts from internal emails at his site; it took me way more time to put the pieces together than to solve the real issue.

I agree with Carlos on SQLT being widely used, especially when dealing with Oracle Support. Outside Oracle I think there are two main barriers to wide adoption of SQLT.

The main one is that it requires an installation. Imagine you are in the middle of a production crisis with your boss over your shoulder waiting for you to fix an issue, and you turn around, look at him or her, and say, "Hey we need a change request [that usually takes days to get processed] to create two new schemas before we can even start using a tool that can help us solve the problem." How do you picture your boss's face? Yeah, me too . . .

The second one, and Carlos may disagree here, is that SQLT is a tool designed by Support to be used by Support; it covers the needs of a Support engineer. It provides a very large amount of information (which translates to long execution times) that, at best, doesn't interest the average DBA/developer. Still, the major drawback of using a tool designed for somebody who looks at SQL tuning issues all day long is, IMHO, that there is very little help in making data analysis easier. It's expected that whoever reviews a SQLT knows what to look for and where. Unfortunately, not everybody outside Oracle Support tunes SQL statements all day long. The first thing that comes to my mind is the tabular format used in the main report. It works well for some types of info (e.g., DBA_TABLES details), but it's painful to review some others (e.g., DBA_HIST_SQLSTAT).

Carlos: Mauro has some good points here, which I need to expand a bit. Mauro and I delivered many sessions of an internal SQL tuning workshop at Oracle while training over 700 Oracle engineers around the world. During those sessions we listened to the needs of many potential users of SQLT, so I would say the content of SQLT was not coming from thin air, and it was not just a product of its authors' prolific imaginations. SQLT actually reflects many needs of many people dealing with SQL tuning issues

often. It is just a big tool that collects and displays diagnostic details about pretty much everything that can affect the performance of a SQL statement.

Is SQLT completely perfected or is there still room for improvement? Are you continuing to expand its feature set?

Carlos: The same way SQLTXPLAIN improved over its ancestor coe_xplain.sql, these days a new and fresh tool, SQLd360, is slowly but surely replacing SQLT. Why is that? Some core reasons follow: 1. SQLd360 does not install anything on the database (SQLT does), and this is so important for a very large number of clients. 2. SQLd360 is 100% free software, while SQLT requires a My Oracle Support (MOS) account, and some consultants and third parties may struggle to obtain a valid login for MOS or would have to use their client's credentials. 3. SQLd360 makes use of newer technologies including Google Charts, presenting the metadata in a very "Wow!" way, while SQLT focuses on HTML tables. 4. SQLd360 is 13X smaller in terms of lines of code, and it pretty much does the same job as SQLT. 5. SQLd360 is much faster to execute than SQLT.

Mauro: Nothing is perfect, especially in IT!

About SQLd360, Carlos said pretty much everything. What I would like to add is an invitation to try out the tool and provide feedback to us so that we can make the tool as useful as possible to the community. The more we hear from the community, the more we can understand what people use the tool for, what is missing, and what might need to be changed or improved. We'd like to think our tools make other people's jobs easier, and it's our way to say "thank you" to a community that every day teaches us so much! (Editor's Note: Please send feedback to mauro.pagano@gmail.com.)

Carlos: Again, Mauro is humble here. Remember, SQLd360 is The Next Generation: younger, faster, thinner, sexier, etc. Honestly, I no longer require my clients to install and run SQLT. I simply tell them about SQLd360 and ask for its output. I am proud of SQLT, and I am also proud of SQLd360. The former served its purpose well, and Oracle will continue to maintain and use it for the foreseeable future, but the latter is the future!

Do SQLT, SQLd360, and eDB360 draw upon Statspack data. Or are these tools only geared toward the extremely fortunate 1% who have a license for Diagnostics Pack and Tuning Pack?

Mauro: Where is the 1% coming from?

The goal of all these tools is to make full use of what is available in the database, so depending on the licensed options the output files will include more or less information.

AWR/ASH are a goldmine of information; they are the foundation of basically every serious performance-monitoring tool, and they are, IMHO, the best way to approach the performance challenges that companies face nowadays. Especially from 11g, where ASH is able to capture information that is almost impossible to extract outside the kernel (e.g., the execution plan line ID that the SQL was on at the time of the sample).

I have a hard time picturing a production database without Diagnostic Pack.

SQLd360 relies a lot on ASH and there is no SP equivalent, so there would be little benefit in using Statspack. Having said that, even without any additional option, SQLd360 would provide quite a lot of info, focusing on those APIs that are part of the base software.

There are some free alternatives to ASH and I would be very happy to consider them, but the reality is that the Diagnostic Pack is likely to get more and more complex with info that will be harder and harder to extract from “free” views.

Carlos: In my humble opinion, having at least the Diagnostics Pack license should be a must on a production database. This pack, which includes AWR and ASH, as Mauro said, is crucial when it comes to fast and accurate diagnostics of performance issues on a mission-critical Oracle database. Yes we could use Statspack as a second best, and yes SQLd360 (and SQLT) could be enhanced to mine data from Statspack, but on the big scheme of cost and benefit, these requirements fall short. Maybe we need to revisit this topic if more clients drift out of the Diagnostics Pack, but for now we benefit from what AWR and ASH kindly provide to all of us. Just be sure you have the Diagnostics or the Tuning Pack licenses before enabling those features on SQLT and SQLd360.

We're excited about your “Practical SQL Tuning” workshop at the YesSQL Summit in January. What topics are covered?

Carlos: We are also excited. This one-day class focuses on what is really used on real-life cases where we have to do some SQL Tuning. Of course it has to start with some foundations, but it moves rapidly into controlling and manipulating Execution Plans. I'll let Mauro expand on its content. I hope to see many of you there, bringing lots of questions.

Mauro: The workshop covers the 4 “T’s” of SQL Tuning: Techniques, Tips, Tricks, and Tools. It takes a practical approach to the topics, keeping the focus on how to actually tackle issues and leverage Oracle functionalities rather than exploring the theory behind each aspect. The session is interactive, with several examples to keep the conversation close to real life.

The first “T,” on the Techniques chapter, covers the different approaches to SQL performance, from the fundamental steps to identify bottlenecks in SQL processing to the multiple options available in Oracle to improve SQL performance.

The second “T,” on the Tips chapter, is a collection of best practices and lessons learned throughout the years on how to avoid or address SQL performance issues, especially when it comes to the first reason for poor SQL performance: CBO statistics.

The third “T,” on the Tricks chapter, groups together all those “mini-tools” you have in your Swiss knife when it comes to tackling and solving SQL problems, covering both custom scripts and standard Oracle APIs.

The last “T,” on the Tools chapter, is an overview of what is freely available out there in terms of tools for SQL tuning, with special focus on what we worked on for years: SQLT and SQLd360.

I cannot adequately express my appreciation for your contributions to the Oracle Database community; you guys are just the greatest! Does SQL tuning continue to be your exclusive focus, or are you moving on to greener pastures at Accenture Enkitec Group?

Mauro: While at AEG I still do a lot of SQL tuning, even though it's not my exclusive focus (it wasn't the exclusive focus at Oracle either). We are part of a team led by Carlos that focuses on database health checks, where we take a holistic approach to the overall database as a component of a larger (and more complex) architecture. Basically we focus not only on the database

health on its own but also on how applications use (or misuse) the database. I really enjoy the type of work we do because by using a top-down approach, you are able to identify real application bottlenecks and problems that when addressed make customers achieve a level of performance that is almost impossible when focusing on the database alone. So we are now doing SQL Tuning and a lot more. Have you heard of “Tuning as a Service (TaaS)”?

Well, we are helping on this new and exciting offering! With TaaS, our clients get some hours of our time, which they can use to tune some challenging SQL or have us look at some global database bottlenecks; it is exciting!

Carlos: I see what I do regarding performance tools as my little contribution to the Oracle community, from which I have received so much. I have learned pretty much all I know about Oracle from reading material others have put together: books, blog posts, articles in magazines, distribution lists, conferences, etc. I just do my share, trying to help our Oracle community to become a better one for the benefit of all of us.

These days Mauro and I are expanding our scope, and we are providing another free software: eDB360. This tool is, for an entire Oracle database, what SQLd360 is for one SQL statement. In other words, it presents a 360-degree view of an Oracle database. We use it every day to perform unbiased health-checks of Oracle databases, as Mauro mentioned. By reviewing the output of this eDB360 tool, we can build a story behind a database's current state and elaborate on sound recommendations to improve the database in terms of scalability, performance, availability, etc. As always, we use our own tools, which helps us to make them better from version to version. Many have asked, “Why do you guys share for free what you use to do your job?” And my answer is, “Why shouldn't we?” I dream of a world where more people freely share knowledge and tools for the benefit of their communities! By the way, have you heard of the new and free Apache Log Analyzer (apalyzer) developed by Dimas Chbane? Maybe a topic for another day. ▲

Carlos Sierra is a regular speaker at Oracle User Group conferences on topics related to Performance and SQL Tuning. He is the author of tools like SQLTXPLAIN, SQLHC, eDB360, and eSP (Enkitec's Sizing and Provisioning). Carlos has almost 20 years of experience in Oracle databases, and many more on legacy UNISYS and IBM mainframes. Carlos currently works as a consultant for Accenture Enkitec Group, where he leads the Oracle database Health-Check team. His home is Seattle now, after 20 years in sunny Florida. He and his wife have raised four responsible young adults, and now they enjoy the peace of an empty nest. Carlos enjoys helping and mentoring others, and contributes to the Oracle community as much as his free time permits.

Mauro Pagano is a database performance engineer with special interest in SQL Tuning and Wrong Results. He is an active member of the Oracle community, always willing to help or mentor other peers, and he enjoys giving back by developing tools and presenting at Oracle User Group conferences. He is the author of SQLd360 and the previous maintainer of SQLTXPLAIN and SQLHC. Mauro has over a decade of experience in the Oracle world, from database tuning to applications development. Mauro currently works as a consultant for Accenture Enkitec Group, as a member of the Oracle database Health-Check team.