



Official Publication of the Northern California Oracle Users Group

NoCOUG

J O U R N A L

Vol. 24, No. 2 • MAY 2010

\$15



Soar with Oracle at NoCOUG

Battle Against Any Guess

An interview with the founder of the BAAG party.

See page 4.

Expert Oracle Practices

"Required reading," concludes our book reviewer.

See page 8.

Battle Against Any Guess

The opening chapter from Expert Oracle Practices.

See page 13.

Much more inside . . .



Battle Against Any Guess

With Alex Gorbachev



Alex Gorbachev

Pythian CTO Alex Gorbachev is a respected figure in the Oracle world and a sought-after leader and speaker at Oracle conferences around the globe. He also regularly publishes articles on the Pythian blog. Alex is a member of the Oak Table Network and holds an Oracle ACE director title from Oracle Corporation. He is the founder of the Battle Against Any Guess (BAAG) movement promoting scientific troubleshooting techniques.

Battle Against Any Guess

Tell us a story. Tell us two. We love stories!

It's June 2007, and I still have enough time left in my day to be active on the Oracle-L list. I'm reading the threads and once again there is one thread full of guesswork-based solutions to solve a particular performance problem. Not the first one and not the last. After entering into the discussion, I felt the conversation was the same I'd had time and time again (like a broken record), and this prompted me to create a place on the internet that I can refer to whenever I and others need to point out the fallacies of guesswork solutions. And so, the BAAG Party was born—www.BattleAgainstAnyGuess.com. The name idea came from the BAARF Party (Battle Against Any Raid Five) organized by fellow Oak Table Network members James Morle and Mogens Nørgaard.

What's wrong with making an educated guess? We have limited data, limited knowledge, limited experience, limited tools, and limited time. Can we ever really know?

"Yes we can!" At least, we should strive to know.

I'll never forget how enlightened I was the moment I saw the slide "Why Guess When You Can Know?" presented by Cary Millsap, another fellow member of the Oak Table Network. Most real life problems can be solved with the knowledge that is available in the public domain, using data that is possible to extract by applying the right experience and tools and taking enough time to do the job properly.

It is the purpose of the Battle to promote the importance of knowledge fighting ignorance, selecting the right tools for the job, popularizing the appropriate troubleshooting techniques, gaining experience, and learning to take time to diagnose the issue before applying the solution. One might think that the BAAG motto is a bit extreme but that's a political decision to emphasize the importance of the goal.

I have elaborated on the concept of the "educated guess" in the first chapter of the book *Expert Oracle Practices: Oracle Database Administration from the Oak Table*. The chapter is titled "Battle Against Any Guess." I would like to quote the following from page 11:

Oracle Database is not only a complex product, it's also proprietary software. Oracle Corporation introduced significant instrumentation and provided lots of new documentation in the last decade, but there are still many blanks about how the product works, especially when it comes to the implementation of new features and of some advanced deployments that hit the boundaries of software and hardware. Whether it's because Oracle wants to keep some of its software secrets or because documentation and instrumentation are simply lagging, we always face situations that are somewhat unique and require deeper research into the software internals.

When I established the Battle Against Any Guess Party, a number of people argued that guesswork is the cruel reality with Oracle databases because sometimes we do hit the wall of the unknown. The argument is that at such point, there is nothing else left but to employ guesswork. Several times people have thrown out the refined term "educated guess." However, I would argue that even in these cases, or especially in these cases, we should be applying scientific techniques. Two good techniques are deduction and induction.

When we have general knowledge and apply it to the particular situation, we use deductive reasoning or deductive logic. Deduction is often known as a "top-down" method. It's easy to use when we have no gaps in our understanding. Deduction is often the path we take when we know a lot about the problem domain and can formulate a hypothesis that we can confirm or deny by observation (problem symptoms).

Inductive reasoning is often considered the opposite of deductive reasoning and represents a bottom-up approach. We start with particular observations, then recognize a pattern, and based on that pattern we form a hypothesis and a new general theory.

While these techniques are quite different, we can find ourselves using both at different stages as verification that our conclusions are correct. The more unknowns we face, the more we favor inductive reasoning when we need to come up with the generic theory while explaining a particular problem. However, when we form the theory via inductive logic, we often want to

prove it with additional experiments, and that's when we enter into a deduction exercise.

When taking a deductive approach first, when applying known knowledge and principles, we often uncover some inconsistencies in the results that require us to review existing theories and formulate new hypotheses. This is when research reverts into inductive reasoning path.

Deduction and induction each have their place; they are both tools in your arsenal. The trick is to use the correct tool at the correct time.

How do we decide which competing methodology to use? Which tool is the best tool for the job? In matters of performance tuning, should we trace, sample, or summarize?

Good questions. Logic and common sense come to mind as the universal methodology for any troubleshooting. If we focus on performance then we should define what it means to improve performance. For me, performance tuning is all about reducing the response time of a business activity. When I think performance, I think response time. This is what Cary Millsap taught me through his book *Optimizing Oracle Performance*—he shifted my paradigm of performance tuning back then (by the way, you can read more about the paradigm shift concept in my chapter referenced above).

Since we identified that response time is what matters, the next step is to analyze where the time goes—build the response time profile. Adopting a top-down approach we might find that 2% of the time is spent on the application tier and 98% of the time spent in the database. Drilling down to the next level of granularity, we could identify two SQL statements that consume a 42% response time each. Focusing on those two, we drill down further into, say, wait events. We could pinpoint the reason for excessive response time at this stage or we might need to dig even deeper—somewhere where timed information isn't available. This is where the current battle lies—we could win it by introducing the right instrumentation and tools.

More than a decade ago, Oracle database performance analysts didn't have the luxury of wait interface and had to rely on various aggregations and ratios as time proxies. The same happens now on another level—when wait interface granularity is not enough, we have to rely on counters and methods such as call-stack sampling. Again, the same goes when execution exits the database, for example, to do storage I/O. Current I/O systems are not instrumented to provide a clear response time profile.

However, I want to emphasize that the vast majority of mistakes during performance diagnostic happen much earlier when we have enough knowledge and tools to avoid applying guesswork solutions, but we often don't.

I digressed in my response from the original question on what the best tools are, but, unfortunately, I will have to disappoint—there is no magic-bullet performance tool that will diagnose all problems. The most sound advice I can give is to study the performance methods and tools available, understand how they work, when they should be used, and what their limitations are and why. There are a number of books published and if you ask me to distinguish one of the recent

books, I would mention *Troubleshooting Oracle Performance* by Christian Antognini.

Should we extend the scientific method to Oracle recommendations or should we adhere to the party line: use the cost-based optimizer, don't use hints, collect statistics every night, upgrade to Oracle 11g, apply the latest patch set, CPU, and PSU, etc.? After all, nobody gets fired for following vendor recommendations. Many years ago, I lost a major political battle about Optimal Flexible Architecture (OFA) and never recovered my credibility there. Once Bitten, Twice Shy is now my motto.

I've touched on the issue of best practices in the BAAG chapter:

"Best practices" has become an extremely popular concept in recent years, and the way IT best practices are treated these days is very dangerous. Initially, the concept of best practices came around to save time and effort on a project. Best practices represent a way to reuse results from previous, similar engagements. Current application of best practices has changed radically as the term has come into vogue.

What are now called best practices used to be called "rules of thumb," or "industry standards," or "guidelines." They were valuable in the initial phase of a project to provide a reasonable starting point for analysis and design. Unfortunately, modern best practices are often treated as IT law—if your system doesn't comply, you are clearly violating that commonly accepted law.

Vendor recommendations are very valuable in the early stages of a project and even later on, as progress is made. In order to apply vendor recommendations correctly, one should understand the reasoning behind such advice, what problems it solves specifically and what else could possibly be affected. If you take an example of collecting statistics every night, then it makes sense for the majority of Oracle databases. There are plenty of exceptions, however, and at Pythian, we often modify the default collection schedule for our customers. Having a sound understanding of what a vendor recommends and why is the key to a successful implementation.

In some cases, it might be difficult to act contrary to generic vendor recommendations, and convincing management otherwise is usually very difficult. Some basic principles to keeping in mind when deciding your course of action are below:

- Vendor recommendations are generic. Consider them as the default configuration of `init.ora` parameters. Nobody runs with all default parameters.
- Instead of going against vendor recommendations, call it modifying or adapting to a particular environment.
- Find a precedent where a recommendation has failed and why. It's like being in court—nothing beats a precedent.
- Playing politics is a whole different game. Either you are a player or you stay away.

Pythian

Tell us something about Pythian. Where does the name come from?

Centuries before the Roman Empire, the Pythian Priestess,

also known as the Oracle of Delphi, was widely recognized and respected as the world's most accurate, most prolific, most trusted dispenser of wisdom and prophecy. Specially chosen, carefully trained, deeply insightful, and profoundly wise, Pythian Priestesses were thought to speak with the voice, the vision, and the very soul of Apollo, the Greek God of the sun, medicine, prophecy, and music.

“If I had to make a decision on whether to invest my money in certification versus conferences or user group fees, I would choose the latter.”

During a dynasty that would last some twelve hundred years, the temple at Delphi was the intellectual center of the world. Pythian Priestesses were credited by the world leaders of the era with guiding and inspiring many great triumphs of art, science, justice, commerce, and civilization. They were also credited with the creation of the Pythian Games, occurring every four years during much of those twelve centuries, alternating with the ancient Olympics, but emphasizing music and poetry as well as athletic contests.

This is well aligned to our vision that database administration is not only a science but an art.

The Pythian website says: “We have developed unparalleled skills, mature methodologies, best practices and tools that ensure Pythian clients receive a level of service that can't be found anywhere else.” Please do tell us some more.

Pythian has been living and breathing databases for over thirteen years. Having a dedicated global team working for our customers 24/7/365, we serve over 400 clients, and perform countless implementations, migrations, etc. To be able to provide such high quality of services, we go out of our way to hire and retain the world's best DBAs in their respective field. DBA talent is our most important asset—four of Pythian DBAs are Oracle ACEs, and many Pythian DBAs are conference presenters and authors on the Pythian blog read by tens of thousands every month seeking help for their technical challenges.

Does Pythian have any plans to support other database technologies such as MySQL and SQL Server? If so, why? If not, why not?

Pythian has been supporting MySQL and SQL Server for years in addition to Oracle databases as well as Oracle E-Business Suite and Fusion Middleware (what used to be Oracle Applications Server all those years). Many of our customers are running heterogeneous environments and our flexible, utility-based business model lets them easily take advantage of all the expertise that we have accumulated from over 13 years in business. We have separate support teams focused on MySQL and SQL Server and several top-tier cross-platform experts.

Today, I'm really excited about MySQL coming under the Oracle umbrella. I think that MySQL and Oracle technologies fit very well together. Oracle has shown strong support for the

MySQL community and it's showing already through the expansion of their Oracle ACE Program to include a MySQL domain speciality. Pythian's Sheeri Cabral has recently been named the very first MySQL Oracle ACE director, for her contributions to the community and MySQL expertise. Very exciting news for all of us as Sheeri is definitely the #1 community leader—she lives and breathes MySQL. I can't stop smiling knowing that Oracle is recognizing MySQL community contributors just like all their other technologies. Hopefully this should lay to rest whines like, “Oracle will kill MySQL.”

The Job Scene

Where have all the DBA jobs gone? I tested the waters by posting my resume on the HotJobs job board and received seven spam replies over the course of a week, none of which had anything to do with database administration. I counted the Oracle DBA positions listed nationwide in the past 30 days on HotJobs and found only 56—far fewer than I used to see.

Identifying and hiring the best candidates has always been one of our top priorities at Pythian. In the past year or two, we haven't seen much difference in the number of elite DBAs available for hire so we've just had to work a little harder to grow three times in the last three years.

Our main source of candidates is the community—through word of mouth, conferences, our blog, and social media like Twitter and LinkedIn. Perhaps job boards are not mainstream for DBA recruiting anymore? Based on what we hear and read from industry research published, database administrators continue to be one of the most in-demand IT positions.

What would you say companies are looking for in an entry-level DBA? What kind of knowledge would you say an entry-level DBA should possess before applying for a job? (Question sent in by a student from Atlanta, GA)

When we hire a DBA, we don't look at years of experience in the traditional way and when it comes to the opening of junior DBA, many candidates with years and years of experience don't qualify. We are looking for individuals with the ability to take responsibility, ability to learn quickly, who possess a broad area of interests in IT, who are logical thinkers and analytical personalities. Because you're front line in working with our clients, it's also very important to have excellent communication skills, and that's not very common among “techies,” to say the least. A number of DBAs started at Pythian as complete juniors within the strong database teams and evolved into our top-tier consultants doing impressive projects now such as Exadata implementation or architecting a distributed system to handle tens of thousands of transactions per second. On the other hand, there were quite a few that started well into their careers, and who were very experienced, but didn't cut it at Pythian.

Will certification help our careers? Which certifications do you recommend, if any? Certification is very expensive because of the requirement to attend training courses at Oracle University. Is it a good return on investment? Should we just read a few good books instead?

I do have OCP certificates for a number of database releases and even as a developer. I can't recall that being a certified DBA

has helped me in my career directly. However, in preparing for the exams I had to fill some gaps in my knowledge and that definitely helped later on. Oh, and my certificates also look good on my wall in nice black-and-gold frames.

On a serious note, some recruiters and HR departments are still looking for certifications as criteria and so I do realize that it might be a vital requirement for junior-mid-level and sometimes even senior positions. In many cases, not always known to outsiders, some vendors require their partners to employ a number of certified specialists to qualify for a certain partnership level. For example, Pythian has to have a certain number of certified DBAs to achieve the Platinum Level Oracle Partner status that we have today. Such a vendor requirement is a great way to motivate your employer to pay your certification pre-requisite courses.

In hindsight, if I had to make a decision on whether to invest my money in certification versus conferences or user group fees, I would choose the latter. It definitely gives a better career boost.

Few of us are well-rounded. We know a lot about Oracle 9i, Oracle 10g, and Oracle 11g, but not much else. Perhaps that's why our jobs are so replaceable. Do you have any recommendations in this area?

When we hire a DBA, we hire permanently. We are engaged in many short-term projects and consulting engagements but never really hire consultants for specific projects. Besides technology knowledge, it's important to consider lots of other criteria such as the ones I mentioned above. Did I emphasize enough that effective communications skills are crucial to a DBA's role?

In addition, for a DBA to be successful, it's not enough to have a good knowledge of the just the database technology. DBAs are usually required to know and understand the intricacies of a very broad set of technologies including networking, storage, operating systems, clustering, virtualization, data modelling, understanding of development frameworks, etc. Investing in learning all those areas will make you more valuable, boost your employment opportunities, and make you harder to replace.

On the other hand, Oracle database technology has become so broad that it's impossible for a mere mortal to know it all in depth these days. Specialization is another way to differentiate yourself, like digging into Security or Streams replication and other data movement features.

Is Oracle a legacy platform? Should we be worried about MySQL and PostgreSQL?

I think that any technology that has reached very high adoption levels becomes stalled and further evolution is much more difficult compared to market newcomers who can enter and revolutionize an industry. Wide adoption doesn't allow a vendor to perform radical changes to the product that will make it unusable to existing customers. However, I think Oracle has been quite successful so far in evolving its product, unlike some of its competitors, who shall stay nameless.

MySQL is definitely on the rise and is one of our fastest growing service areas. With Oracle taking over MySQL, we expect MySQL only to accelerate. This is why we developed a special MySQL Accelerator program which is specifically de-

signed to kick start your DBA team with MySQL technology, based on your own environments over the course of just few days. It's been ever popular since its introduction at Pythian. You can find out more about the program on our website or by sending an email to mysql@pythian.com.

Bonus Round

It seems that every move I make and every step I take is being watched and recorded in a database. Are databases more evil than good?

Information is power. If it's in good hands—it's good power. Information that falls into bad hands is dangerous. That's why it's so important to safeguard your data these days and why information security and privacy is so high on the radar of IT managers.

Thanks for coming out to speak at our next conference; we're looking forward to your presentations on RAC workload management. It costs us about \$14,000 per year to produce and distribute the NoCOUG Journal and considerably more than that to organize four conferences. We have about 500 members and about 200 attendees at each conference; we've stagnated at those levels for many years. Is Google making us obsolete? I begged my previous company to buy a corporate membership but, except for my manager, not a single person from that company ever showed up at our conferences. Should we close shop? What could we do better to attract more members and conference attendees?

The Internet has provided us with a new communication medium that brings completely new opportunities and communication efficiencies that shouldn't be ignored. In order to adapt to the age of the internet, we all must re-evaluate our investments and optimize their effectiveness.

Because a user group exists solely for the benefit of its members, I believe that members should decide whether the *NoCOUG Journal* is worth the investment. Perhaps it makes sense to ask them to vote whether they would rather see the printed *Journal* (which costs them \$28 of their annual fees) or a more informal online journal/newsletter edition that is five times cheaper, and spend the rest of the funds to organize an additional seminar or two for the members.

Having said that, face-to-face communication is very important and provides another dimension in networking possibilities. Online media still doesn't provide all nuances of human interaction, so gathering a big crowd of like-minded people triggers completely new discussions and ideas.

While building Pythian Australia in Sydney, Australia, I founded Sydney Oracle Meetup—an informal club sponsored by Pythian and gathering twice a month in the evening in a semiformal environment where we had a presentation followed by follow up discussions and networking—all that mixed with pizza and drinks. In just a year, we've reached almost 200 members and 20–40 people are showing up for every meeting. Perhaps you could try such informal regular gatherings? ▲

Contact Alex at gorbachev@pythian.com, follow him on Twitter [@AlexGorbachev](https://twitter.com/AlexGorbachev) or read his blog at www.pythian.com/news/author/alex.