ORACLE MAGAZINE

JANUARY/FEBRUARY 2018

# Open for Developers

## Explore the stack, the development services, and the technology in the open Oracle Cloud Platform

AI IS EVERYWHERE

LET'S HAVE A CHAT

USING ASYNC FUNCTIONS

ORACLE

# Vertex and Oracle — Partners in Global Tax Solutions

For accurate tax content, calculation, and reporting, nothing works better with Oracle than Vertex.

- Over 20 years of Oracle integration experience
- More than 1,000 mutual clients
- Trusted partner for Oracle ERP Cloud and Oracle ERP
- Oracle Validated Integration for transaction and payroll taxes
- Seamless integration with Oracle E-Business Suite R12 and JDE

*In the Cloud and On Premise*

**Click here to learn more about our Vertex Cloud Solution**

ORACLE Platinum Partner Cloud Standard

Available on ORACLE Cloud Marketplace

VERTEX®

CONTENTS

ORACLE MAGAZINE

**30**
## The Future Is Open
Why Oracle's cloud-native stack is fueling innovation
**BY ALEXANDRA WEBER MORALES**

**9** MashUp

## FEATURES



## UP FRONT



**12** Interview

CONTENTS

## COMMUNITY



**26** Peer-toPeer

## TECHNOLOGY

# ORACLE MAGAZINE



---


**SUBSCRIPTION INFORMATION**
Subscriptions are complimentary for qualified individuals who complete the subscription form.

**MAGAZINE CUSTOMER SERVICE**
Omeda Communications

**PRIVACY**
Oracle Publishing allows sharing of its mailing list with selected third parties. If you prefer that your mailing address or e-mail address not be included in this program, contact Customer Service at oracle@omeda.com.

# Data Quality Made Easy. Your Data, Your Way.


Melissa provides the full spectrum of data quality to ensure you have data you can trust.

We profile, standardize, verify, match and enrich global **People Data** – name, address, email, phone, and more.

Our data quality solutions are available on-premises and in the Cloud – fast, easy to use, and powerful developer tools and plugins for the **Oracle® Ecosystem**.

View Video

ORACLE PEOPLESOFT | ORACLE JD EDWARDS | ORACLE E-BUSINESS SUITE | Java | ORACLE Forms | ORACLE PL/SQL

## Start Your Free Trial www.melissa.com/oracle

Melissa Data is now Melissa.
See What's New at www.Melissa.com

1-800-MELISSA

melissa™
GLOBAL INTELLIGENCE


Tom Haunert

# DBAs: Get Ready

## Oracle Autonomous Database Cloud is coming. What are you going to do?

**Oracle announced Oracle Autonomous Database Cloud** at Oracle OpenWorld 2017. (See the links in Next Steps for information and a video on the news.) Oracle also calls the autonomous database the "self-driving database."

But DBAs—the engineers, builders, operators, and maintainers of databases—have been "driving" databases for many years. What happens to the DBA's driver role in a self-driving world?

That's way too big a question to answer in one editorial, so *Oracle Magazine* is going to answer that question in bits and pieces over the course of 2018. The first part of the answer is easy, so let's start there. Waiting until you're responsible for the data and operations of a self-provisioning, self-patching, self-tuning database to learn more about that technology platform is *not* recommended.

I spoke with Maria Colgan, master product manager for Oracle Database, about the role of the DBA as keeper of a self-driving database. She gave me a lot of great information on what's next for both administrators and Oracle's next-generation database, and I'm working on an interview with her for the next issue. Meantime, here are a few bits of advice from Colgan:

- Learn about the latest database features. Start with Oracle Database 12*c* Release 2 features.

- Understand the features and capabilities included with the latest Oracle Database Cloud service offerings.
- Consider what eliminating "generic database tasks"—such as provisioning, updating, patching, and backing up your databases—means to your processes and productivity.
- Prepare to work more closely with your company's business and developer teams to ensure the best use of your database for the company's application and business goals.

And for those DBAs sticking with current on-premises databases, Colgan also suggests focusing on security at every level—because that will continue to be a priority for DBAs *not* using Oracle Autonomous Database Cloud. (Updating and patching are not autonomous in on-premises database installations.)

Tom Haunert,
Editor in Chief

Emails and posts received by *Oracle Magazine* and its staff may be edited for length and clarity and may be published in any medium. We consider any communications we receive publishable.

PHOTOGRAPH BY
**BOB ADLER/THE VERBATIM AGENCY**

## NEXT STEPS

**LEARN** more about the self-driving database.

**WATCH** *Oracle Autonomous Database: The Role of the DBA*.

**TRY** Oracle Database Cloud services.

# We guarantee to significantly improve the performance of your Oracle environment.

ORACLE Platinum Partner

Cybernoor is the leader in **Applications Performance Optimization**. The Cybernoor System Review will optimize your Oracle environment to ensure your system performs well, scales, and is ready for tomorrow's challenges. We guarantee our solutions. Contact us today to schedule your System Review.

ORACLE Platinum Partner
**Specialized**
Oracle Database 12*c*

ORACLE Platinum Partner
Cloud Standard

ORACLE Platinum Partner
**Specialized**
Oracle Real Application Clusters 12*c*

ORACLE Platinum Partner
**Specialized**
Oracle WebLogic Server 12*c*

ORACLE Platinum Partner
**Specialized**
Service-Oriented Architecture

ORACLE Platinum Partner
**Specialized**
Oracle Exadata Database Machine

Cybernoor Corporation I www.cybernoor.com I +1.925.924.0400

cybernoor™

# Full STEAM Ahead

Smart gadgets and apps from young developers


## Watch the Wanderers

Kenneth Shinozuka was 14 when he began working on a wearable device to monitor patient activity. The resulting SafeWander—consisting of a tiny button sensor worn by the patient, a gateway, and a mobile app—sends an alert to the caregiver's mobile device when the sensor detects a change in the patient's body position. US$199. safewander.com


## See the Beat

High schooler Suman Mulumedi was curious about what physicians do when heart murmurs are too quiet to be picked up by a stethoscope. So he created Steth IO, an elegant case that turns a smartphone into a stethoscope, bringing visualization to heart and lung sounds with precise digital filters. Device users can monitor their heart at home and send recordings directly to their physician for review. Steth IO is FDA-approved for clinicians and for patients with prescriptions. US$199. stethio.com

PHOTOGRAPHS COURTESY OF SAFEWANDER AND STETH IO

## Computer Science for All?

**Does a student need to major in computer science in order to snag one of those high-paying jobs?**

Computer science experience can boost a candidate's chance of success in high-growth, high-paying careers in data analysis, engineering and manufacturing, design, marketing, and programming and IT areas, but most of those jobs do not require a computer science degree.

**62%** Jobs in the fastest-growing and highest-paying skill areas of data analysis, engineering and manufacturing, arts and design, marketing, and IT and programming that are computer science related

**18%** Jobs in those fast-growth, high-pay fields that request a computer science degree

**300%** Growth over the last decade in the number of high school students taking AP computer science

**900%** Increase from 2013 to 2016 in the number of code boot camp graduates

Source: Rebooting Jobs: "How Computer Science Skills Spread in the Job Market" (Burning Glass Technologies, November 2017)

## DO YOU SPEAK TECH? QUIZ YOURSELF!

1. **What day is celebrated as International Day of Women and Girls in Science?**
   - **A.** February 11
   - **B.** April 1
   - **C.** June 5
   - **D.** October 15

2. **Which of these four languages is the newest?**
   - **A.** Groovy
   - **B.** CoffeeScript
   - **C.** Python
   - **D.** Ring

3. **Which of the following jobs in artificial intelligence ranks highest in average salary?**
   - **A.** Machine learning engineer
   - **B.** Data scientist
   - **C.** Business intelligence developer
   - **D.** Research scientist

**Answers:** 1. A: February 11; 2. D: Ring; 3. A: Machine learning engineer

# APPS: YOUNG DEVELOPERS WITH SMART SOLUTIONS

***Turn your mobile device into a science lab, social savior, and school guide for the blind with apps from the under-20 crowd.***

### Lab4Physics

When University of Chile student Komal Dadlani became aware that many students in her country did not have the equipment to properly study science, she and two fellow students founded a company to do something about it. Lab4U develops apps such as Lab4Physics—which includes speedometer, accelerometer, and sonometer sensors—that turn smartphones and tablets into scientific tools so that any student with access to a phone can conduct research in the physical world. Free (Android, iOS, Microsoft Windows)

### Sit With Us

Sixteen-year-old Natalie Hampton ate lunch alone every day in seventh grade and wanted to spare students from the loneliness and bullying such isolation can cause. So she developed Sit With Us, a social networking app that lets kids find a place to sit with the click of a button rather than facing an awkward walk through the lunchroom. The app also allows students to act as ambassadors, inviting others to join them and hosting open lunches that anyone can join. Free (Android, iOS)

### Hello Navi

Navigating a school's traffic jam between classes can be challenging for any student, but it's especially hard for the visually impaired. A group of six grade-school girls in Los Fresnos, Texas, developed Hello Navi to address the problem. With Hello Navi, a mobility specialist can set up various paths between campus locations for a visually impaired student to use; the student then talks into the device to request help navigating to and from locations. Free (Android)

As one of several benefits, the Oracle Intelligent Bots platform addresses the challenges that customers face when they need to integrate back-end systems with their bots, explains Suhas Uliyar, vice president of mobile, bot, and AI strategy at Oracle.

# Let’s Have a Chat

Chatbots are here, and the technology behind them makes all the difference.

**BY TOM HAUNERT**

**Chatbots replace conventional application user** interfaces with human conversation. Chatbots combine and connect messaging channels, artificial intelligence (AI), and back-end integrations to help enterprises automate these conversations.

*Oracle Magazine* sat down with Suhas Uliyar, vice president of mobile, bot, and AI strategy at Oracle, to talk about chatbot and AI technologies, chatbot challenges, and chatbot solutions from Oracle.

*Oracle Magazine:* What are chatbots, and why are they important?

Uliyar: Simply defined, a *chatbot* is a computer program designed to simulate conversation with human users.

Chatbots are important because they acknowledge and address a change in engagement preferences. The last decade saw a major adoption of mobile as an engagement channel for consumers and employees within the enterprise. But now we're seeing an increase in the adoption and use of messaging channels including Facebook Messenger, WhatsApp, WeChat, Slack, and SMS, and voice personal assistants including Amazon Echo Dot, Google Home, ApplePod, and so on as preferred engagement channels.

Messaging channel adoption is happening quickly because it is instant, available 24/7, and people can use natural language and get a consistent experience across multiple devices. This is leading to innovations and use cases for chatbots powered by artificial intelligence that can help enterprises automate conversations at scale through these messaging and voice channels.

*Oracle Magazine:* How do emerging technologies—including AI, machine learning, and natural language processing—support chatbots, and what kind of challenges do they introduce to chatbot development?

Uliyar: One of the primary reasons for chatbots is to enable end users to communicate with natural language, so the first aspect of the AI and machine learning technology support is all about being able to understand the end-user conversations. A primary component of this AI technology is natural language understanding [NLU] and natural language processing [NLP].

NLP applications attempt to understand

natural human communication, either written or spoken, and communicate with us using similar, natural language. Machine learning helps machines understand the vast nuances in human language and to learn to respond in a way that an audience is likely to comprehend.

There are many reasons why AI technologies are viable and important right now. The two main ones are that, first, compute power has increased quite a lot and can handle the sophistication of the machine learning algorithms behind AI technologies. And second, the increase in access to data over the last decade means that machine learning has more to learn from, so it can be more accurate and more deterministic.

The challenges that come with AI machine learning include having enough data to learn from and understanding where and how to start developing projects that include chatbots, AI, machine learning, and other emerging technologies. At Oracle, we've been focusing on helping customers drive toward a much more effective conversational AI platform that can help them get over the hump of implementing chatbots.

*Oracle Magazine:* What are the biggest challenges for companies that want to chatbot-enable existing apps or develop new chatbot apps?

Uliyar: Our customers want to know where and how to get started with chatbots or conversational AI. And that means defining and fine-tuning the use cases for the chatbot. Organizations can look at what they've implemented in their mobile apps as a quick start toward bots. They can use their apps to look at the typical repeated questions from customers and the standard responses.

After defining use cases, the next challenge to getting started with chatbot projects is determining what channels to support. There are a plethora of different channels such as Facebook Messenger, WhatsApp, WeChat, Line, Telegram, Skype, Alexa, Amazon Echo Dot, Google Home, and more.

The next challenge is to then determine the scope of a bot as it relates to what to do when the bot is unable to answer questions, either because the bot is not capable of answering those questions or the bot hasn't been config-


ured to answer the questions that the end users are asking. If a chatbot doesn't cover *all* of the organization's use cases or something is not configured properly, how does your chatbot still provide the best possible user experience?

And finally, how does the organization integrate the bot into their enterprise? How do they securely connect the chatbot to multiple messaging channels and then connect to the different systems of record in an effective way to provide timely responses and the best possible user experience?

*Oracle Magazine:* What are Oracle's chatbot technology solutions, and how do they address enterprise and developer challenges for building chatbots?

Uliyar: We developed a mobile service—Oracle Mobile Cloud Service—in Oracle Cloud that has widespread global customer adoption across industries. We've expanded our offering and built our Oracle Intelligent Bots platform on top of the mobile service to offer Oracle Mobile Cloud Enterprise. This new platform is powered by some emerging technologies, including AI that uses machine learning algorithms, to help provide a high level of intelligent engagement with end users.

The Oracle Intelligent Bots platform addresses customer challenges in several ways. First, it's a comprehensive end-to-end-platform with everything an enterprise needs to deliver a successful chatbot solution to its customers. The bot platform provides an abstraction layer to integrate with the different channels, including Facebook Messenger, WeChat, WhatsApp, Alexa, Google, and so on.

The second benefit for the enterprise and developers is that we provide the full NLU service with our dialogue engine and a tool that helps the customer define and refine use cases. The NLU engine is implemented as a pipeline of algorithms that can help customers train the model based on the amount of data they have. Customers that have limited or no data can just provide a few sample phrases to get the model started, and as the system gets used, the algorithms auto-adjust to learn from the expanded dataset. This provides organizations and developers with a design principle and the tooling to help them understand how to model the natural language very simply, because a big challenge again for some customers is that they don't have enough data to support all of their use cases.

And at the same time, we support development that lets a bot pass on new queries with full context to a human agent in case the bot either cannot answer the question because it hasn't been programmed to or it just needs to be configured correctly. So, the human agent handoff and bot design that can clearly articulate what it can and cannot do are design principles that we also enabled as part of our dialogue engine and our NLU engine.

The third benefit for the enterprise and developers is the conversational AI capabilities that are integrated with the Oracle Intelligent Bots platform and provide all the machine learning algorithms, the cognitive services, the dialogue and context services, knowledge services, and data and insights for reporting.

Finally, the Oracle Intelligent Bots platform addresses the challenges that customers face when they need to integrate back-end systems with their bots. The bot platform includes a full integration stack that leverages current system integrations and investments in integrating with back-end systems so bots can more easily include a comprehensive end-to-end flow, enterprise security, release management capabilities, version control, and so on over the lifecycle of a bot.

PHOTOGRAPHY BY **PHIL SALTONSTALL**

## NEXT STEPS

**LEARN** more about Oracle Cloud for mobile and chatbots.

**READ** more about Oracle Mobile Cloud Enterprise.

**TRY** Oracle Cloud services.

# From Ripples to Waves

Here are some trends that made a difference in 2017.

By **Bob Rhubart**

**Software architects and developers** live in a perpetual swirl of technological trends, each with at least some potential to evolve from ripple to wave. A great many of these trends simply evaporate. But for others, powerful currents converge, mysterious forces align, and what emerges is something that changes the software development landscape in a profound and lasting way.

As we enter a new year, it seems appropriate to look to community members to get a real-world sense of the trends and technologies that had the greatest impact on their work in the year just passed. Among those who responded to my inquiries, most agree that 2017 was the year of APIs and microservices, and many offered ideas about trends for 2018.

For Oracle ACE Phil Wilkins, a Capgemini senior consultant, a highlight of 2017 was the arrival and what he describes as the "rapid maturing" of Oracle's API platform. "We have seen a lot of uptake of the platform and work to go with it," he says.

Wilkins believes that the solutions within Oracle's API platform "offer the greatest freedom and are making the greatest impression," he explains. "Cloud-born businesses are typically more technology-centric. However, those not cloud-born often need an evolutionary path. And that's what makes microservices important."

**“Chatbots are coming of age. We’re starting to see past the novelty use cases into enterprise applications.”**

***—Phil Wilkins, Oracle ACE***

“Microservices lend themselves to the need for smaller, discrete pieces that are more movable,” Wilkins explains. “We’re seeing increasing involvement in applying this kind of paradigm.”

For Oracle ACE Sven Bernhardt, the focus in 2017 was on “everything around APIs and modern, hybrid architectures, based on new architectural concepts like microservices.” A solution architect with Opitz Consulting, Bernhardt explains how the combination of microservices, container technologies, and cloud-based scalable runtime environments makes it possible to define “flexible, future-proven architectures.”

Microservices and APIs were among several trends that landed heavily on Oracle ACE Rolando Carrasco’s radar in 2017, a list that also includes DevOps and chatbots. “For microservices, it just happened that during 2017 a lot of customers wanted to invest around it,” Carrasco says. Co-owner and SOA principal architect at S&P Solutions, Carrasco says that for some customers the interest in microservices was more about following the trend than about a legitimate technical need. Either way, he was happy that microservices were in the mix. “It was fun to work with them,” because that work meshed well with his SOA expertise.

As for APIs, “API management was and will continue to be a favorite topic to work with, both for research and implementation,” Carrasco says. He presented on API management at Oracle OpenWorld 2017 and other events, and plans to do so in the future.

And speaking of the future, while Wilkins, Bernhardt, and Carrasco agree on the impact of APIs and microservices in 2017, there isn’t quite that same level of consensus about 2018.

All three agree that chatbots will be a major factor this year. “Chatbots are coming of age,” explains Wilkins. “We’re starting to see past the novelty use cases into enterprise applications.” Similarly,

Bernhardt believes that advances in artificial intelligence (AI) will drive the adoption of chatbots.

All three also agree that in 2018 serverless architectures will be a trend to watch.

Finally, both Bernhardt and Carrasco expect blockchain to gain momentum in 2018. "Blockchain has the potential to become a game changer in the area of B2B transactions," according to Bernhardt.

Of course, the opinions of these three professionals, however experienced, carry only so much weight. But Wilkins, Bernhardt, and Carrasco are in the trenches on a daily basis and are respected, active members of the community, so it is reasonable to assume that they are not alone in their opinions.

But what about you? What trends had the greatest impact on your work in 2017? And what trends will drive real change in 2018? Join the conversation.

---

*Oracle Developer Community Architect Community Manager Bob Rhubart is the host/engineer/producer of the* Archbeat *podcast series, produces the* 2 Minute Tech Tip *video series, and interviews technology experts in DevLIVE videos recorded at Oracle Code, Oracle OpenWorld, and other events.*

PHOTOGRAPHY BY
**MICHAEL MCELROY/THE VERBATIM AGENCY**

## NEXT STEPS

**LISTEN** to
"Are Microservices and APIs Becoming SOA 2.0?"

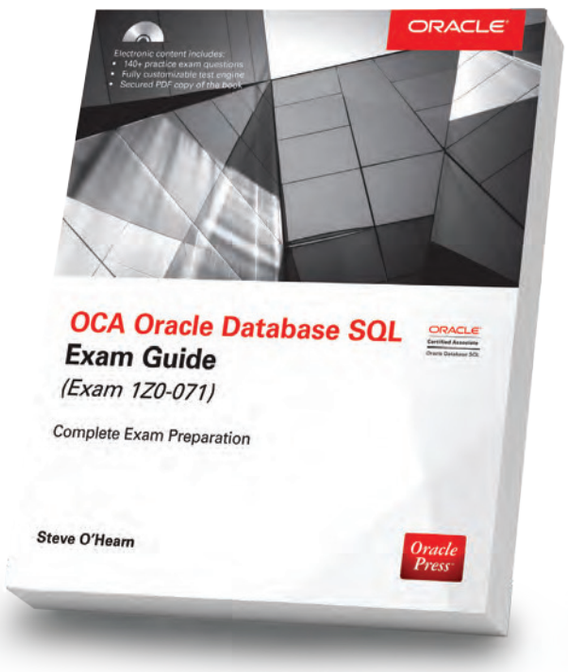
**LISTEN** to
"Microservices: Getting Real."

**WATCH**
The interview: *Microservices and API Management.*

Oracle Press™

# Your Destination for Oracle and Java Expertise

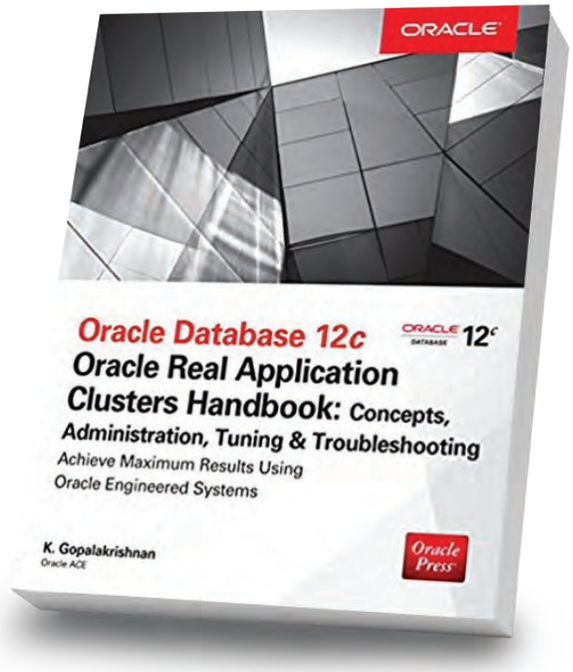
## Written by leading technology professionals, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Oracle products and technologies available.




### OCA Oracle Database SQL Certified Associate Exam Guide (Exam 1Z0-071)

*Steve O'Hearn*

A fully updated, integrated self-study system for the Oracle Database SQL Exam.




### Oracle Database 12c Oracle Real Application Clusters Handbook

*K. Gopalakrishnan*

Master Oracle Database 12*c* Release 2 Real Application Clusters with this comprehensive, fully updated guide.


### Oracle Mobile Cloud Service Developer's Guide

*John Ray Thomas*

Create modern, enterprise mobile apps with Oracle Mobile Cloud Service.


### Oracle VM 3 Cloud Implementation and Administration Guide, Second Edition

*Ed Whalen, Erik Benner, Nic Ventura*

Master cloud building with Oracle VM 3 installation, configuration, and maintenance.

**Available in print and ebook formats.**

Oracle Press™

**www.OraclePressBooks.com • @OraclePress**

# Cloud Is a Boon for Database Experts

What does cloud mean for your job? Technologists respond and point to opportunities for data wranglers.

**The uptake of cloud services is accelerating.** But how do technology strategies and operations change with the move to cloud services? Do they expand or shrink career choices for people in IT?

*Oracle Magazine* asked technologists who are actively using Oracle Cloud services how they see cloud fitting in to their technology stacks and how they see themselves fitting into the "new normal" of cloud computing. Here's a collection of their best observations and advice.

## LEARN THE CLOUD AND BECOME A LEADER

Enterprise cloud computing has evolved from its early days, when businesses were mostly using cloud to buy new software-as-a-service (SaaS) apps or build new applications. They are now looking to bring core applications to cloud. This, said Ross Smith, a chief architect at PITSS America, is a big opportunity for technologists who know how to navigate the change. "The legacy system that a company wants to modernize is often crucial, so our job as the data experts is to help a company express what its legacy systems do in a new forward-looking technology." Smith's advice is to break down the functions of the application and advise the organization on new technologies, such as NoSQL, or cloud services that pre-

serve the legacy system functionality but offer the benefits of cloud. Watch more from Smith.

One of the immediate benefits of cloud, said Tariq Farooq, Oracle ACE Director and chief architect at BrainSurface, is the instant provisioning that happens with cloud services. "In less than 30 minutes, we spun up an entire cloud service instance," he said following a workshop he helped lead at an ODTUG meeting. "It's nothing short of a revolution. It's point, click, click to provision the services, and you are ready to go. That process used to take weeks or months." Watch more from Farooq.

## CHANGE YOUR IDEA OF WHAT'S POSSIBLE

Provisioning quickly is an important and immediate benefit, but it's the ready-made integrations for building and extending applications that drew Erik Benner, an enterprise architect at Mythics, to the cloud. "Oracle platform as a service (PaaS) [Oracle Cloud Platform] is sort of like Lego bricks, where you have different components: Oracle Documents Cloud, Oracle Process Cloud Service, Oracle Mobile Cloud Service, Oracle Messaging Cloud Service, and many others. These are all prebuilt application components, so as an application developer you can combine them to make a complete application that you can rapidly deploy to your users." Watch more from Benner.

For GV Rao, a technology strategist at L&T Infotech in India, that plug-and-play ability is helping his engineering firm converge the physical and digital worlds. "We brought sensor devices to water filters [on a large water filtration project], which are connected to Oracle Internet of Things Cloud Service, which in turn connects to Oracle Java Cloud Service and Oracle Mobile Cloud Service, so our customers can see what is happening in real time on their mobile devices and respond to it." The project came together quickly, said Rao, because a group of technologists at his

company who were experts in the Oracle technology spent time to learn what Oracle Cloud had to offer. Watch more from Rao.

## BROADEN YOUR PERSPECTIVE

Learning what cloud has to offer is key for Martin D'Souza, a chief innovation officer at Insum Solutions. "Your company might think its plans will require building a huge complicated thing," he said, but if you're aware of the infrastructure and architectures that exist on cloud, "you can tell them, 'Wait—there's a cloud service that can solve the problem for five dollars a day.'" This advisory role should become part of your role as a DBA or database developer, D'Souza said. "You understand how the data is stored, how it's indexed, and where it lives—on cloud or on premises—and you can really help identify [and help execute] cloud services that make things much less complicated or expensive for the company." Watch more from D'Souza.

## EMERGING TECHNOLOGIES ARE YOUR FRIEND

A deep understanding of SQL and Oracle Database is a ticket to success for using emerging technologies, reported several New Normal participants. For example, the Internet of Things (IoT) opens opportunities for DBAs who master the art of tuning streaming data, said Jerry Ward, a database developer for Viscosity. With IoT, "we've got a new kind of data that is very high velocity and volume that's traversing over infrastructure. Being able to stand up the infrastructure that's able to coalesce that streaming data, being able to process code for streams, eliminating hot points on specific databases, and federating it all out through the cloud worldwide—those are all in the realm of the DBA now," he said. "I think it's more exciting than ever for the DBA." Watch more from Ward.

Jon Dixon agreed. An integration project helped Dixon, an Oracle tech lead at JMJ Cloud, realize that DBAs have most of the skills they need to

connect back-office information to voice assistants, such as Alexa and Google Voice. It's the type of service he thinks is going to catch on quickly with the C-suite. "Alexa and Oracle Database are a real natural marriage," he said. "Oracle Database Exadata Cloud Service has a fantastic tool in Oracle REST Data Services that exposes REST services, which is what Alexa needs to execute the engine behind a request," he said. "Because DBAs have that knowledge on the database side, it's a relatively easy process for them to learn the rest." Watch more from Dixon.

Overall, participants in New Normal agreed that the current migration to cloud is a time of opportunity for people who know Oracle Database. "It's become our job to recommend forward-looking solutions to business problems," concluded PITSS America's Smith. "It's our job as data experts to see [opportunities that cloud offers] and bring those opportunities to the floor and realize that that's a role we're going to fulfill in the future."

*Jeff Erickson is editor at large for Oracle Content Central.*

## NEXT STEPS

**LEARN** more about Oracle Cloud.

**TRY** Oracle Cloud services.

# “Just Go Do It”

When learning the ropes, try experimentation, education, and hitting the open road.

**Jim Czuprynski**
Bartlett, Illinois

ORACLE ACE Director

**Company/URL:** ViON Corporation
**Job title:** Advanced technical consultant
**Oracle credentials:** Oracle Certified Professional (Oracle Database 11*g*, Oracle Database 12*c*)
**Length of time using Oracle products:** 16 years

**How did you get started in IT?** Back in 1981, I took advantage of an offer to learn all about a fourth-generation hierarchical database tool called FOCUS to build some shadow IT for my company’s growing end-user reporting demands. I fell in love with programming and went on to get really involved with Oracle database administration. I eventually became an Oracle University instructor—training more than 2,000 DBAs during my 10-year tenure.

**What advice do you have about getting into database development?** Read through the 2-Day DBA guide to learn the basics. Then build your own Oracle VM VirtualBox environment, download the Oracle infrastructure and database software, build your own Oracle database, and learn how to back it up properly. Next you’ll want to destroy that environment and recover from its loss. In other words, *just go do it*.

**What’s the most common cause you see when IT projects go wrong?** I’ve never forgotten something I read in Gerald Weinberg’s book *The Secrets of Consulting* [Dorset House, 1986]. “No matter what your client tells you, it’s always a people problem.” Having excellent cohesion between every team member is key.

## Piet de Visser

Spa, Belgium

ORACLE ACE

**Company/URL:** PDVBV

**Job title:** Owner

**Length of time using Oracle products:** More than 20 years

**Which features in Oracle Database are you currently finding most valuable?** I'm exploring pluggable databases, notably the cloning possibilities, for use with fast-paced development teams. Using the pluggable database cloning feature for so-called *thin cloning*, or *thin provisioning*, allows me to very quickly make a copy of a database based on what storage engineers know as the "copy-on-write" principle. In the near future, I hope to have this process working on a semiautomated self-service basis, and timed down to minutes.

**What technology has most changed your life?** Every solution that has enabled mobile working. I do a lot of work remotely—it's why I work for myself—and I love having the freedom to go wherever work takes me. I prefer traveling by motorcycle, which is much more fun, and it's what I use for most of my transport these days, including my visits to customers. But the best trips are the visits to Oracle user groups in countries farther away. I do two to five longer "touring" trips every year, visiting places such as Helsinki, Paris, Vienna, Belgrade, Bucharest, and Sofia. On these trips I present at user group and SIG [special interest group] meetings, at meetups, and at OTN Developer Days.

**What are your go-to Oracle reference materials?** I really rely on, in this order, AskTom; the official documentation, which I never skip reading; support.oracle.com; and the blogosphere. Blog-wise, I particularly like ORACLE-BASE and blog.dbi-services.com. I also closely follow blogs by Richard Foote, Martin Widlake, and Frits Hoogland. Tim Hall's ORACLE-BASE has been around since the early days, and it's always informative.

**Ivica Arsov**
Skopje, Macedonia

ORACLE ACE

**Company/URL:** Pythian

**Job title:** Senior database consultant

**Oracle credentials:** Oracle Database 12*c* Administrator Certified Master; Oracle Certified Expert

**Length of time using Oracle products:** Eight years

**What technology has most changed your life?** Definitely databases, especially Oracle Database. It has been my livelihood for the past eight years, and I've also started blogging and becoming more active in the Oracle community by presenting at Oracle technology conferences. The conferences represent a unique way of learning and sharing knowledge with other experts. They are a great opportunity to meet new friends and visit different countries.

**You've taken Oracle University [OU] classes in the past. What led you to do this?** I live by the motto "Invest in your future." Educational investment is crucial for every professional. We have to upgrade our knowledge on a daily basis in order to stay in touch with fast-evolving technologies. I've taken both in-classroom and online OU classes, and I like both, for different reasons. In a classroom you can get in-person feedback from the instructor, but with online classes you can review and watch the session multiple times.

**What's the next big thing driving change in your industry?** Automation. Oracle recently announced Oracle Database 18*c* as the first autonomous database, which opens a new direction in the future of the database world. The idea is to enable the database to be self-managing; it will support fully automated patching, backups, and upgrades, and also the ability to perform routine maintenance tasks. These automation capabilities integrated in the database software will free us up to focus more on architectural and performance type of work.

COLLABORATE 18

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

# REGISTER TODAY!

EARLY BIRD REGISTRATION ENDS MARCH 8

A COLLABORATIVE EVENT
PRESENTED BY:

IOUG OAUG Quest

APRIL 22 - 26 | LAS VEGAS, NV | MANDALAY BAY RESORT & CASINO

www.attendcollaborate.com

# The Future Is Open

## Why Oracle’s cloud-native stack is fueling innovation

BY ALEXANDRA WEBER MORALES

The technologies of tomorrow have the potential to change the world for the better, whether it's via planet-scale apps scheduled with Kubernetes Federation or blockchain-based trust with Hyperledger. These two examples are both enabled by open source software (OSS)—but OSS is only one facet of Oracle's commitment to being open.

"When we say Oracle is open, we don't just mean open source. We mean that we support non-Oracle technologies as well as the ability to work in an ecosystem that is broader than Oracle. We definitely believe that," says Amit Zavery, senior vice president for Oracle Cloud Platform. Zavery describes a vision for Oracle that includes open source tools, databases, platforms, and languages; managed services to make OSS more intuitive to use; and open collaboration, innovation, and interoperability.

"We're taking important open source innovations and providing automation, management, and enterprise-grade capabilities around them," Zavery says, pointing to Oracle Cloud services for managing a plethora of OSS technologies, including Kubernetes (such as Global Multi-Cluster Management Fn, Hadoop, Spark, Kafka, the Cassandra NoSQL database (by way of Oracle Data Hub Cloud Service), TensorFlow, and Caffe (via Oracle Artificial Intelligence Platform Cloud Service), to name just a few.

**More OSS Than Ever Before**

With each year, OSS expands in importance. Developers have gained ground too: Today, their status has risen from that of hidden minions to that of visible makers in a software-bitten world. As a result, the tech industry has quickly moved from questioning the OSS business model to embracing it.

"There are a lot of areas where we need either more free software or more involvement in existing free software projects," says Berlin, Germany–based computer scientist Lydia Pintscher. A free-culture enthusiast, Pintscher is the product manager for Wikidata and the president of KDE e.V., a nonprofit supporting free and open end-user software. Her work, she says, aims to break through the tendency to lock down technology and "make it harder and harder to tinker with."

Oracle agrees with Pintscher: Innovation thrives when technology can be shared and modified. The company has delivered mightily

> **When we say Oracle is open, we don't just mean open source. We mean that we support non-Oracle technologies as well as the ability to work in an ecosystem that is broader than Oracle. We definitely believe that.**
>
> —Amit Zavery, Senior Vice President, Oracle Cloud Platform, Oracle

in the past year on the promise of an open and modern development platform, including

- Launching Oracle Container Native Application Development Platform, a container-native development stack based on Docker and Kubernetes, along with Oracle's own contributions
- Joining the Cloud Native Computing Foundation (CNCF) and putting top talent behind Kubernetes
- Contributing the Fn serverless platform to open source
- Joining Hyperledger and basing Oracle Blockchain Cloud Service on it

Today, the cloud-native application stack designed by Oracle has open elements at every layer from containers to orchestration—and one of the newest pieces, serverless development, only reinforces that commitment.

**Serverless and the Developer Experience**

One sign that developers are at the top of the heap? The degree to which technology companies such as Oracle enhance productivity with tools that flow delightfully together.

"Serverless is a foundational part of the next technology evolution, and it needs to be built on an open, integrated, and seamless stack—which leads to an integrated and seamless developer experience," says Mark Cavage, software development vice president for Oracle Container Native Application Development and Java. That seamless stack comprises Docker containers, managed Kubernetes-based orchestration, and one of Oracle's most recent open source contributions, serverless functions on the Fn platform.

*Serverless computing*—programming logical operations that run on an abstracted deployment platform in the cloud—promises to rev

up the engine of innovation. Developed by the IronFunctions team, Oracle's Fn Project is a command-line tool for coding and deploying functions in platform-independent Docker containers. These functions perform discrete operations such as image processing or semantic analysis. And the micro-operations can scale dynamically, eliminating the sticker shock of unfettered pay-as-you-go applications gobbling cloud resources.

"The way we package software is fundamentally different, thanks to containers," says Chad Arimura, Oracle vice president of software development and founding CEO of Iron.io. "But they aren't without difficulties, especially at scale. The Fn Project gives developers a 'containerless experience' by abstracting out the complexities—yet exposing their power."

**Java, a First-Class Language for the Open Cloud**

While the modern container stack is polyglot, Java is a first-class language for the cloud. Oracle takes its stewardship of Java, which is used by more than 12 million developers worldwide, seriously. To make it easier to deploy Java apps to the cloud, Oracle plans to ship OpenJDK builds under the general public license (GPL), allowing developers to freely distribute apps along with their frameworks and applications.

An exciting element of the Fn serverless platform is the Java Function Developer Kit (FDK), which simplifies the process with a language-specific library. (There are also FDKs for Go and Python.) These functions are also portable to and from other platforms such as Amazon's Lambda—an indicator that Oracle's approach to container-native stacks avoids the lock-in found in Amazon Web Services.

"Amazon is probably the most locked-in cloud provider," says Zavery. "You can't take anything you build on AWS very easily somewhere else, unless of course you build it on compute bare metal. Lambda is a good example, and so are Stack templates, Amazon Formation, and Kinesis—they're all completely locked in."

**Open Container Orchestration**

The Oracle contribution that is most likely to thrill cloud-native developers is container orchestration. In September 2017, Oracle joined the Linux Foundation's CNCF, which guides the development of the Kubernetes container orchestration tool. Bob Quillin, Oracle vice president of developer relations for the Container Native Group,

> **“Serverless is a foundational part of the next technology evolution, and it needs to be built on an open, integrated, and seamless stack—which leads to an integrated and seamless developer experience.”**
>
> —Mark Cavage, Vice President, Software Development, Oracle Container Native Application Development and Java, Oracle

notes that CNCF also boasts Prometheus for monitoring, Open Tracing for instrumenting distributed code, gRPC for remote procedure calls, and the Open Container Initiative. Oracle has released Kubernetes on Oracle Linux; committed engineers to the Kubernetes project; and open sourced tools including Smith, CrashCart, and Terraform Installer for Kubernetes on Oracle Cloud Infrastructure.

Several engineering teams at Oracle are dedicated to the Kubernetes effort, particularly in security, networking, and federation. TJ Fontaine, former Node.js project lead, is now Oracle’s lead contributor to Kubernetes. Jon Mittelhauser, who wrote the first widely used web browser and now works as an Oracle vice president of container native engineering, has joined CNCF’s governing board.

**Blockchain Could Change Everything**

If serverless is the ultimate expression of utility computing, blockchain is the ultimate expression of transactional trust in a networked world—and as such, it must be built with open standards in mind. The new Oracle Blockchain Cloud Service leverages Hyperledger Fabric, the open source effort led by the Linux Foundation. Blockchain is a cloud-based distributed ledger that can be used in business-to-business partner ecosystems for more-secure transactions and data sharing.

“Blockchain holds the promise to fundamentally transform how business is done, making business-to-business interactions more secure, transparent, and efficient,” says Zavery. “Oracle Blockchain Cloud Service provides enterprise-grade blockchain capabilities and is

able to accelerate innovation for on-premises enterprise resource planning and cloud-based SaaS and PaaS customers. Enterprises can now streamline operations across their ecosystem and expand their market reach with new revenue streams, sharing data and transacting within and outside Oracle Cloud."

As part of its commitment to helping enterprise customers realize the benefits of blockchain, Oracle recently joined Hyperledger, the open source collaborative effort hosted by the Linux Foundation and created to advance cross-industry blockchain technologies. By maintaining interoperability with open standards, Oracle enables customers to benefit from all open source innovations and avoid vendor lock-in.

> **"The Fn Project gives developers a 'containerless experience' by abstracting out the complexities—yet exposing their power."**
>
> —Chad Arimura, Vice President, Software Development, Oracle

"There are religious or scientific arguments for OSS," says Quillin, "but there's also the market. Customers are demanding it."

**Being Open Powers Innovation**

It bears repeating that Oracle's OSS credibility is strong but often ignored. "We've been doing this for years, but other companies seem to get more open source kudos than we do," says Zavery, echoing perennial comments by Quillin and others.

"We came into the Oracle Cloud group as part of the Stack Engine acquisition, and it was very much an open source–centric environment," says Quillin. "From the outside in, people scratch their heads about Oracle and OSS. From the inside out, it's a very strong commitment, with a lot of new blood and new startups combined with a rich history across open standards and open source including Java, Linux, and MySQL. Across Oracle, that's fanning out into lots of different facets of cloud computing built on an open source platform."

Beyond interoperability and market forces, there's another force driving Oracle's increasingly open technology stack: Open platforms

and OSS foster innovation. Why? Developers are frequently motivated to join something with a bigger potential reach than their own company's commercial vision. Hobby projects can hone new skills or, with luck, turn into the next Docker. Open platforms don't just avoid lock-in; they allow a best-of-breed culture to flourish—and shine a light on vulnerabilities.

"We need to demand much more to be able to tinker with our devices, change them, and make them do things they were not intended for—because a locked-down device makes it harder or even impossible to run free software on it," says Pintscher. "If you open up your software and technology, you empower people. *You* won't have all the best ideas of how a program should work and be used. Empower others to let them help themselves *and* you."

It's becoming ever more clear, as developers seek to create apps that serve diverse audiences around the world, that inclusive teams build better solutions. Giving them the tools to do so—and the ability to modify those tools as they see fit—is the way of the future.

---

*Alexandra Weber Morales is Oracle's director of developer content.*

PHOTOGRAPH BY **EASTIMAGES/ MOMENT OPEN/GETTY IMAGES**
ILLUSTRATION BY **PEDRO MURTEIRA**

## NEXT STEPS

**JOIN** the Oracle Developer Community.

**EXPLORE** Oracle application development services.

**TRY** Oracle Cloud services.

# It's Pervasive: AI Is Everywhere

Artificial intelligence is rewriting the enterprise technology landscape. Here's what you can do now, and what you will be able to do soon.

BY ALAN ZEICHICK

When the little wireless speaker in your kitchen acts on your request to add chocolate milk to your shopping list, there's artificial intelligence (AI) working, in the cloud, to understand your speech, determine what you want to do, and carry out the instruction.

When you send a text message to your HR department explaining that you woke up with a vision-blurring migraine, an AI-powered chatbot knows how to update your status to "out of the office" and notify your manager about the sick day.

When hackers attempt to systematically break into the corporate computer network over a period of weeks, AI sees the subtle patterns in historical log data, recognizes outliers in the packet traffic, raises the alarm, and recommends appropriate countermeasures.

AI is nearly everywhere in today's society. Sometimes it's fairly obvious (as with a chatbot), and sometimes AI is hidden under the covers (as with network security monitors). It's a virtuous cycle: modern cloud computing and algorithms make AI a fast, efficient, and inexpensive approach to problem-solving. Developers discover those cloud services and algorithms and imagine new ways to incorporate the latest AI functionality into their software. Businesses see the value of those advances (even if they don't know that AI is involved), and everyone benefits. And quickly, the next wave of emerging technology accelerates the cycle again.

**AI Drives Chatbots to Enhance the User Experience**

AI can improve the user experience, such as when deciphering spoken or written communications or inferring actions based on patterns of past behavior. AI techniques are excellent at pattern-matching, making it easier for machines to accurately decipher human languages using context. One characteristic of several AI algorithms is flexibility in handling imprecise data: human text. Specifically chatbots, where humans can type messages on their phones and AI-driven software can understand what they say and carry on a conversation, can provide desired information or take appropriate actions.

In many ways, chatbots are at the forefront of a revolution in creating more-natural human interfaces with machines, explains Suhas Uliyar, vice president of mobile, bot, and AI

> **“The increase in access to data over the last decade means that machine learning has more to learn from, so it can be more accurate and more deterministic.”**
>
> —Suhas Uliyar, Vice President of Mobile, Bot, and AI Strategy and Product Management, Oracle

strategy and product management at Oracle. He explains that there are three trends toward message-based conversation. First, messaging channels such as Facebook Messenger and WhatsApp are familiar to users, so users don’t need to download and learn new apps. Next, users can engage using the language of their choice. And finally, a well-programmed chatbot can respond instantly.

For enterprises to scale with the volume of interactions that comes with such messaging channels, AI is critical for processing these natural-language conversations and returning the appropriate response.

This doesn’t mean that the technology is simple, says Uliyar. It takes a lot of processing power, as well as a lot of data, to compute the sophisticated AI algorithms behind text recognition, including determining context, retrieving information, and, of course, formulating the best reply—all in a fraction of a second.

The challenge is compounded when a user sends the chatbot informal language, slang, tones (emotions), or misspellings. Still, the development of powerful cloud services with impressive computational ability, as well as huge quantities of—and improvements in—text algorithms, has brought the industry to the tipping point of making chatbots ready for pervasive deployment, says Uliyar. The compute power has increased enough to handle the sophistication of the machine learning algorithms behind AI technologies, he says, and “the increase in access to data over the last decade means that machine learning has more to learn from, so it can be more accurate and more deterministic.”

That’s why responsive chatbots are ready for prime time. “With a chatbot, you can apply things like your customer service use cases to it,

or marketing or transactional use cases. In just the same way that you enabled mobile interfaces, you can enable these chatbot interfaces," Uliyar says.

Oracle customers can develop and deploy chatbots using Oracle Mobile Cloud Enterprise. With Oracle Intelligent Bots technology, a key feature of the mobile cloud offering, chatbots can communicate with users on messaging channels such as Facebook Messenger, WeChat, Slack, Apple Siri, Amazon Echo, Google Home, or even a chat widget on a business's website—or customers can extend an existing mobile app with chat or voice capabilities.

Behind the scenes, chatbots can be trained for natural-language processing, via algorithms that use deep learning neural networks and supervised and unsupervised machine learning, to interact and respond based on an organization's specific context (such as product offerings or services). The result: the chatbots understand the user's intent and provide an intelligent response or the appropriate action in real time, 24/7.

Oracle Mobile Cloud Enterprise also uses AI to extract all the necessary information from the user's natural-language input to shape the payload for integrating with the sources of data to respond to the user. For example, when a user asks the banking bot to transfer US$5,000 from account A to account B, AI not only recognizes the user's intent to transfer funds but also extracts relevant information such as the From and To account, the currency, and the amount in order to construct the necessary transactional statement to execute in the banking application. AI algorithms continue to evolve with support for additional cognitive capabilities such as image processing and real-time video processing, further increasing the possibilities of an AI-powered conversational user interface.

**AI Thrives Buried Deep in the Code**

Chatbots are only the tip of the iceberg when it comes to AI and machine learning. Beneath the surface of Oracle Human Capital Management Cloud (Oracle HCM Cloud), for example, AI empowers many hidden functions for human resources, including making recommendations for guiding employees' careers, advising on work schedules that take childcare into account, and evaluating performance reviews.

Similarly, AI algorithms appear in everything from image recognition (using neural networks) and medical diagnostics (using expert systems)

> **“Machine learning has been in the Oracle Database, management, and security products for many years. We have a lot of expertise using AI inside products.”**
>
> —Amit Zavery, Senior Vice President, Oracle Cloud Platform, Oracle

to data analysis (using machine learning) and beyond. AI is hard at work in advanced search engines—even in so-called “recommendation engines” that learn about customers’ buying patterns and offer suggestions about other products or services they might wish to purchase.

Consider cybersecurity, where machine learning algorithms can pore through terabytes of log data from firewalls, application servers, websites, and more and spot outliers. Those outliers could be very subtle patterns indicating that a hacker has breached a system. Alternatively, those patterns might indicate that a server or mobile device is running out-of-date software, or that patches haven’t been made, or even that the hardware might be getting ready to fail.

Where’s the AI? It’s hidden. But it has a big impact nonetheless.

Retail is a practical example of applied AI, says Amit Zavery, senior vice president of Oracle Cloud Platform, citing a scenario in which a customer might upload a photo of a particular dress and then the ecommerce platform might recommend the closest available matches from that retailer. A demonstration application leverages image recognition, big data, and machine learning to present the best choices to the shopper nearly instantly.

“It’s so much less frustrating than having customers browse through a large catalog,” Zavery explains. “It doesn’t matter if the dress is from a movie or a magazine or wherever; the AI looks through the inventory and suggests matches.” The results: delighted customers who get the dress they want, and happy retailers who make the sale and possibly earn future business as well.

### The Tech Is Coming Together

AI isn’t new. It’s not even close to new. For decades, computer science researchers and industry developers have been designing and constructing software that uses AI techniques to

solve problems in medical diagnostics, factory automation, pharmaceutical research, text analysis, stock picking, and even playing games such as chess and Go. However, the results were rarely scalable to massive deployments, leaving AI out of the mainstream.

Things have changed, with several breakthroughs in microprocessors, cloud computing, databases, and algorithms coming together.

**GPUs.** In terms of microprocessors, graphics processing units (GPUs)—chips initially designed for playing computer games and doing 3D modeling—excel at the complex matrix-oriented math required for neural networks, machine learning, and image processing. In fact, GPUs are significantly more efficient, by orders of magnitude, than general-purpose central processing units (CPUs). Today, GPU-based servers form the underpinning of modern AI processing systems.

**Cloud technology.** Cloud computing has brought massive amounts of processing power, storage, and bandwidth to enterprise developers. With special cloud computing services built with GPU-based servers, high-powered AI has suddenly become accessible both for developers to experiment with and for corporations to deploy as complete applications. Helping to drive both proof-of-concept projects and line-of-business initiatives for Oracle customers are the capabilities built into the new Oracle Artificial Intelligence Platform Cloud Service (Oracle AI Platform Cloud Service) running on GPU-based servers.

**Algorithms.** Driving the intelligence behind AI are complex algorithms for processing tremendous quantities of structured and unstructured data. Popular algorithms, such as the Caffe deep learning framework, Jupyter Notebook data cleansing and modeling library, Keras neural networks API, NumPy high-performance array processing library, scikit-learn data mining and machine learning library, and TensorFlow library for machine intelligence, make AI accessible to enterprise developers.

Thanks to these libraries, coders and data scientists no longer need a PhD in applied AI to make heads or tails of advanced techniques. Nobody should pretend that AI is easy to code, but it's becoming easier for enterprises to utilize, thanks to platforms such as Oracle AI Platform Cloud Service and its support for popular algorithms and frameworks.

And then, of course, there is the data itself. The emergence of the internet sparked a huge

> **“There are so many frameworks, and users don’t know which is best or [don’t] want to maintain all of them. We give developers a choice of all the open frameworks in Oracle AI Platform Cloud Service.”**
>
> —Amit Zavery, Senior Vice President, Oracle Cloud Platform, Oracle

increase in the amount of digital information being generated, stored, and made available for analysis—in fact, 90 percent of the world’s data was created in the last two years. Data is fodder for statistical analysis, which is the key for higher accuracy in machine learning models.

**Where Is AI in Oracle Products? Everywhere.**

Oracle has long used AI in its own products, says Zavery, and thus is no newcomer to the field. In fact, AI is pervasive inside Oracle products and services. “Machine learning has been in the Oracle Database, management, and security products for many years,” Zavery says. “We have a lot of expertise using AI inside products.”

Oracle Management Cloud leverages AI to help detect anomalies in very large datasets, such as to indicate problems on a manufacturing line.

Oracle Cloud Access Security Broker Cloud Service uses machine learning to drive its threat-protection and data-protection capabilities.

Similarly, Oracle Analytics Cloud services use natural-language understanding and processing to let business users ask a computer sophisticated questions and get answers in English or the user’s native language—or displayed as charts, graphs, or other meaningful formats.

That’s only the beginning. Oracle Adaptive Intelligent Apps, built into Oracle Cloud applications, bring AI into familiar business domains. Oracle Adaptive Intelligent Apps are embedded into Oracle Enterprise Resource Planning Cloud (Oracle ERP Cloud), Oracle HCM Cloud, Oracle Supply Chain Management Cloud (Oracle SCM Cloud), and Oracle Customer Experience Cloud (Oracle CX Cloud). In addition, the recently

announced Oracle Autonomous Database Cloud uses AI to eliminate complexity, human error, and labor-intensive manual database administration.

**Build Your Own AI with Oracle AI Platform Cloud Service**

Announced in October 2017, Oracle AI Platform Cloud Service helps enterprise developers build new AI applications—and also embed AI functionality into existing applications to make those applications faster, more feature-rich, and more valuable for the business.

Before Oracle announced Oracle AI Platform Cloud Service publicly, Oracle used that service internally, explains Zavery. "We have been using many of the machine learning frameworks ourselves," he says. "We started using it first, and now we are making it available to customers."

Oracle AI Platform Cloud Service is very flexible, Zavery adds, because it includes many built-in algorithms and frameworks, such as Caffe, TensorFlow, and DL4J, and runs directly on existing Spark/Hadoop clusters for big data computation and analysis.

Because developers and data scientists often don't know which AI models or algorithms are best for a specific application and it's cumbersome or impossible to experiment with different algorithms if a vendor's AI platform supports only a single model, this multialgorithm approach is key.

"There are so many frameworks, and users don't know which is best or [don't] want to maintain all of them," Zavery says. "We give developers a choice of all the open frameworks in Oracle AI Platform Cloud Service. We give choice to customers. Oracle maintains and manages the various libraries and keeps them up to date, so developers can focus on building their application and prototyping their models. They can augment their data with other third-party data and Oracle-provided datasets."

It's much easier to use Oracle's toolkit to set up and test four open AI frameworks against the same data, he explains, "rather than setting up and testing four frameworks using services from four different vendors." Not only that, he adds, but Oracle is committed to developing and testing all popular machine learning algorithms within the cloud service, so the functionality will continue to grow over time.

**AI Is Pervasive in Business**

If you think AI is everywhere today, expect more tomorrow. Oracle's AI-enhanced software-as-a-service and platform-as-a-service products will continue to incorporate additional AI to help make cloud-delivered and on-premises services more reliable, more performant, and more secure. AI-driven chatbots will find their way into new, innovative applications, and speech-based systems will continue to get smarter. AI will handle larger and larger datasets and emerge in increasingly diverse industries.

Sometimes you'll see the AI and know that you're talking to a bot. Sometimes the AI will be totally hidden, as you marvel at the uncanny intelligence of the software, websites, and even the Internet of Things. If you don't believe me, ask a chatbot.

---

*Alan Zeichick is principal analyst at Camden Associates, a technology consultancy in Phoenix, Arizona, specializing in software development, artificial intelligence, enterprise networking, and cybersecurity. Follow him at @zeichick.*

ILLUSTRATION BY **iSTOCK.COM/INSTANTS**

## NEXT STEPS

**LEARN** more about AI at Oracle.

**EXPLORE** Oracle AI Platform Cloud Service.

**TRY** Oracle Cloud services.

# THE DATABASE FUTURE: IT'S HERE

Oracle Database 12*c* delivers tomorrow's on-premises and cloud data management—today.

BY MIKE FADEN

Data management is at the heart of fundamental trends now transforming how businesses use technology. Companies are exploiting vastly expanding volumes of data from a growing number of sources. Rapid analysis of that data has become key to maintaining a competitive edge. And cloud technologies for gathering, storing, analyzing, and using that data are enabling dramatically increased agility as well as entirely new business models.

With Oracle Database 12*c*, Oracle has engineered its industry-leading enterprise database to support these transformational trends while protecting businesses' investments. Multitenancy, a foundation of Oracle Cloud, is a powerful feature of Oracle Database 12*c*. The Oracle Database 12*c* multitenant option also facilitates greater agility on premises and zero downtime migration to Oracle Cloud, supporting instant provisioning while slashing administrative effort.

Oracle Database In-Memory (available with Oracle Database 12*c*) supports high-performance analytics simultaneously with transaction processing on live operational data. And capabilities such as native JavaScript Object Notation (JSON) support make it easier to integrate and analyze big data from many sources, including the web and Internet of Things devices.

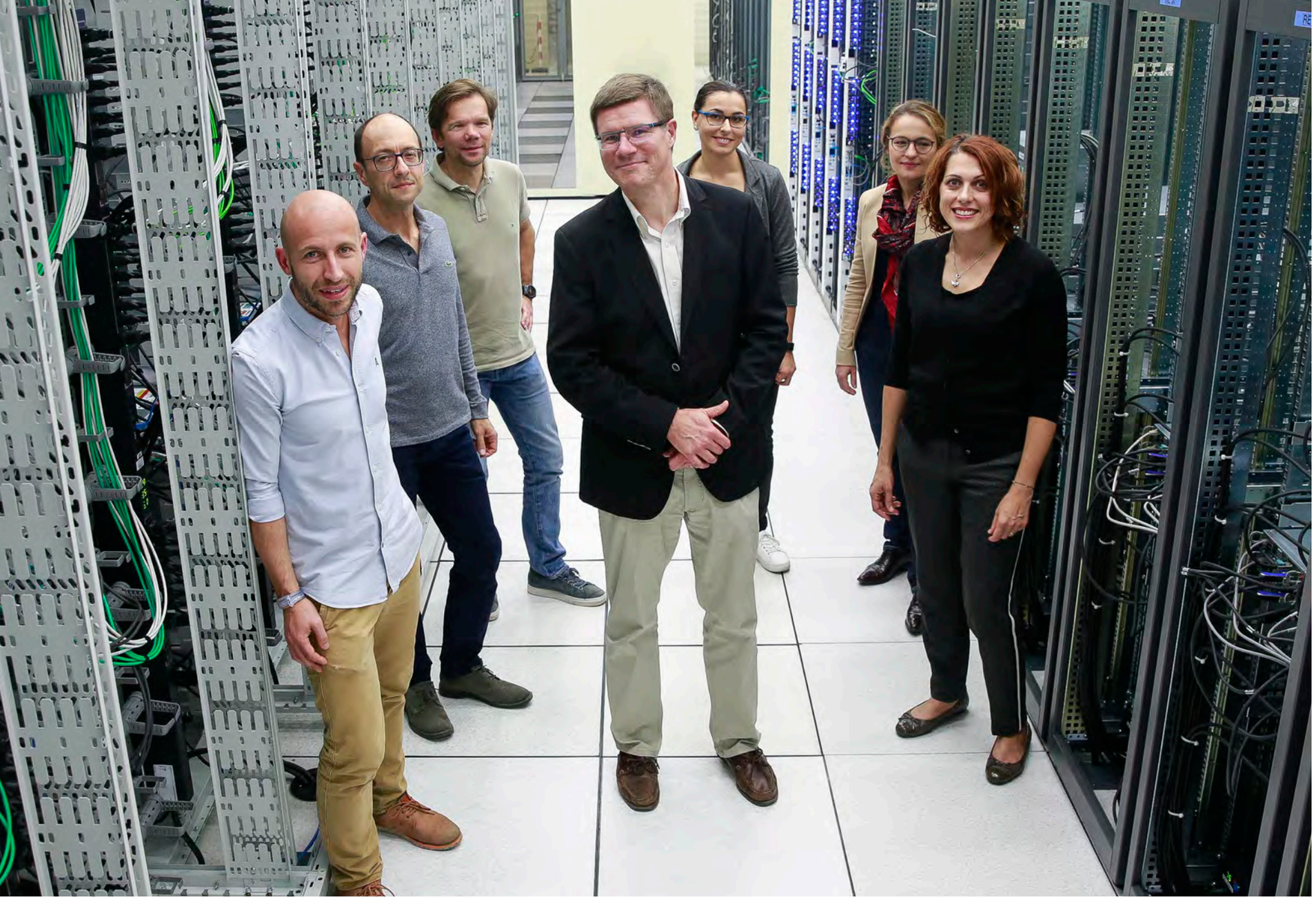
"Every ten years, the computing world changes completely," says Andy Mendelsohn, executive vice president for database server technologies at Oracle. "The huge generational change going on right now is that we're transitioning from internet computing in customer data centers to the cloud."

And with Oracle Database 12*c*, he says, Oracle continues with innovations that give customers the flexibility to choose their own path in that direction and pursue it at their own pace.

**CERN**
Geneva, Switzerland

**EMPLOYEES:**
More than 2,250

**ORACLE PRODUCTS:**
Oracle Database 11*g* and 12*c*
Oracle Database Cloud
Oracle Big Data Discovery
Oracle Database Backup Cloud Service
Oracle Java Cloud Service
Oracle Cloud Infrastructure
Oracle Active Data Guard



CERN IT department members (left to right) Manuel Martin Marquez, Luca Canali, Artur Wiecek, Eric Grancher, Ana Lameiro Fernandez, Katarzyna Dziedziniewicz-Wojcik, and Eva Dafonte Perez pose in a CERN data center where vast amounts of data are stored—data that helps to answer fundamental questions about the universe.

**Colliding Particles, Exploding Data at CERN**

Among the organizations taking advantage of Oracle Database 12*c* capabilities is CERN, the European Organization for Nuclear Research. Located near Geneva, Switzerland, the particle physics lab seeks to explore fundamental questions about the structure of the universe (becoming famous along the way for inventing the World Wide Web, in 1989). To study particles and how they interact, CERN uses the world's largest scientific instrument, the Large Hadron Collider (LHC). The LHC accelerates particles to near the speed of light, creating collisions that help scientists gain a better understanding of

> **“This can really change the way people think about computing architectures. The in-memory enhancements are one of the features we’re going to be deploying in Release 12.2 as soon as possible.”**
>
> —Eric Grancher, Group Leader, Database Services, CERN

how our universe works. Among the LHC’s key achievements was providing, in 2012, experimental evidence of the Higgs boson—the previously unverified final piece of what is known as the Standard Model of particle physics.

While creating collisions between particles, the LHC also creates extraordinary amounts of data. Even after filtering out more than 99 percent of the raw data produced by experiments, CERN still expects to generate about 50 PB in 2017, says Eric Grancher, group leader of CERN’s database services group. The data is stored and analyzed on the Worldwide LHC Computing Grid (WLCG), a cloud-based global distributed computing infrastructure, coordinated by CERN, involving more than 170 computing centers worldwide. The WLCG runs more than 2 million jobs daily.

More than 100 Oracle production databases are used to manage some of the vast amounts of control data generated by the LHC’s systems. “We are talking about a sustained level of 150,000 changes per second,” says Eva Dafonte Perez, deputy head of CERN’s database services group. CERN’s “quench protection” database alone generates 150 TB of data per month, she says, and the LHC logging database captures 1.5 million signals from a variety of sources.

In addition to providing hardware and software to CERN for 35 years, Oracle has now been a partner in CERN openlab for well over a decade. CERN openlab is a collaboration between the laboratory and leading information and communications technology (ICT) companies that works to accelerate the development of the cutting-edge ICT solutions needed

to make groundbreaking research possible. A number of Oracle solutions have been tested through CERN openlab, with CERN's input helping to guide the development of new database features.

The organization runs a combination of Oracle Database 12*c* Release 1 (12.1) and Oracle Database 11*g* and is planning to take advantage of Oracle Database 12*c* Release 2 (12.2). One feature that CERN uses is Oracle Database In-Memory, which accelerates key database operations and uses a unique dual column and row datastore to support analytics and production workloads on a single database.

"The big breakthrough with the in-memory technology in Oracle Database 12*c* is that you can do high-performance analytics against your live transactional data," Mendelsohn says.

For CERN, one key advantage of the in-memory capability is transactional integrity for specific scale-out applications, Grancher says. Rather than maintain data caches at the application level, the entire database can be stored in memory, ensuring a single source of consistent information and guaranteeing transactional integrity. Grancher says one project has already changed its plans to take advantage of the new capability. "This can really change the way people think about computing architectures," he says. "The in-memory enhancements are one of the features we're going to be deploying in Release 12.2 as soon as possible."

CERN is also looking at the possibility of using in-memory capabilities together with Oracle Active Data Guard to flow data workloads from primary LHC databases to secondary stores for analysis. With Oracle Database 12*c*, says Mendelsohn, you can now create an Oracle Active Data Guard standby for your transaction system and create only the in-memory column store technology that supports those high-performance analytics on the standby database. "You can send the transactional users to the primary database," he says, "and you can send your business analytic users to the Oracle Active Data Guard standby, where the data might only be a few seconds behind the data in the production system. Now, you have the best of both worlds, where you can completely isolate the transactional users from the analytic users, but still deliver incredible high-performance analytics against data that's almost real-time."

Big data analytics will support an expanded range of use cases across the laboratory, says

Manuel Martin Marquez, a big data specialist in CERN's database services group. "With big data technologies, we're able to perform real-time analytics; we are able to perform analytics on bigger datasets with better performance," he says.

CERN expects to start deploying Oracle Database 12*c* Release 2 when the LHC enters a scheduled shutdown phase next year. The LHC operates in carefully planned cycles, typically consisting of a three-year period producing collisions followed by a roughly two-year shutdown for upgrades and maintenance. The next upgrade will allow the LHC to run with much higher performance, resulting in more particle collisions and further opportunities for scientific analysis and discovery. The higher performance will generate even more data at faster rates, especially because CERN expects to retain a greater proportion of the raw experimental data for analysis. By 2027, the accelerator could be generating around 600 PB per year, says Grancher—requiring new, more-efficient ways to analyze data.

As the data management requirements expand, CERN, in collaboration with other research institutions, is looking to the cloud to help handle the load. CERN already manages its main data center (and an offsite extension in Hungary) through a private cloud environment based on OpenStack, with approximately 300,000 processor cores.

As part of this, CERN is exploring the integration of Oracle Cloud Infrastructure (formerly Oracle Bare Metal Cloud) into its current environment, says Katarzyna Dziedziniewicz-Wojcik, a senior database administrator at CERN, and is in the process of testing a variety of Oracle Cloud capabilities. In one very large test, CERN ran a scientific workload on 10,000 cores in Oracle Cloud Infrastructure. CERN also plans to run applications on Oracle Cloud Infrastructure Compute, to assess performance, the cloud resources required, and the effort involved.

Additional tests for which CERN is preparing include cloud-based disaster recovery using Oracle Active Data Guard. "We have quite a complex environment, with multiple networks and high-performance network links. We want to see how complex it would be to set up such a solution in the cloud and also whether a cloud solution would fulfill our performance needs," Dziedziniewicz-Wojcik says. CERN is also looking at using containers to switch work-

loads to the cloud for disaster recovery. The organization has already shown that it's possible to move applications from an internal Kubernetes cluster to a cluster in Oracle Cloud Infrastructure, for example.

**Viscosity: Consolidate and Simplify for Customers**

Like CERN, many other organizations are finding that key features introduced in Oracle Database 12*c* are valuable both in the cloud and on premises, says Charles Kim, who has extensive experience with implementations of the database both as president of the niche database and cloud consulting firm Viscosity North America in Dallas, Texas, and through his role as president of the Independent Oracle Users Group (IOUG) Cloud Computing Special Interest Group.

For example, one Viscosity healthcare client, a software-as-a-service (SaaS) company, is using multitenancy to drastically simplify its environment with server consolidation, he says. Previously, the company was supporting its users on many different servers, based largely on older hardware. "Managing all of that became a nightmare," Kim says.

With Oracle Database 12*c*, Viscosity has been able to help consolidate those databases using multitenancy by implementing them as multiple pluggable databases within database containers. Because Release 12.2 supports consolidation of pluggable databases with different character sets, the company has been able to achieve greater efficiency by consolidating databases supporting customers in different regions.

"They were able to perform extreme consolidation, which addressed scalability and stability issues—and alleviated their headaches," says Kim. Release 12.2

**VISCOSITY NORTH AMERICA**
Dallas, Texas

**EMPLOYEES:**
10–50

**REVENUE:**
US$7 million

**ORACLE PRODUCTS:**
Oracle Database 12*c*
Oracle Exadata
Oracle Private Cloud Appliance
Oracle Real Application Clusters
Oracle WebLogic
Oracle Management Cloud
Oracle Artificial Intelligence Platform Cloud Service
Oracle Cloud Infrastructure

Charles Kim, president of Viscosity North America, says multitenancy, the ability to move entire tables online while automatically updating the indexes, index usage tracking, and longer object names are some of the capabilities he sees customers benefiting from with Oracle Database 12*c*.

has also facilitated testing by enhancing support for flashback in pluggable databases.

Businesses are also taking advantage of some of the less high-profile—but extremely practical—new features in Oracle Database 12*c*, Kim says. For example, Release 12.2 introduced the ability to move entire tables online, while automatically updating the indexes. That means easier online maintenance with less or zero downtime. Another key example is the index usage tracking capability in Release 12.2, which provides additional information about how individual database indexes are used over time. “As an administrator, it’s very important

to know, because if a specific index is used only once as part of a batch job that runs every three months, why even keep it there? Eliminate the overhead," says Kim.

The longer object names allowed in Release 12.2 are proving valuable to customers as well. Before the new release, all object names were limited to 30 bytes; now, the limit has been increased to 128 bytes, making it easier to provide descriptive names. Kim says that's valuable in a variety of situations, such as when moving data into Oracle Database from other datasources that support long object names.

"We want to be at the frontier of Release 12.2 adoption to help customers migrate to Oracle Cloud with zero downtime," says Kim, whose company has authored more than 18 books focused on Oracle technology.

**AS ONE Uses JSON for Real-Time Collaboration**

Oracle Database 12*c* includes important new features for developers, including the ability to store JSON as text directly within the database. Japanese company AS ONE is taking advantage of that JSON support as it uses Release 12.2–based Oracle Database Exadata Express Cloud Service to support a key requirement: providing its 4,300 dealerships with real-time information for more than 300,000 inventory datasets.

AS ONE is a general trading company offering a broad range of equipment and scientific instruments for users in the research, industrial, and medical fields. It offers products through catalogs and the company's ecommerce website, and it provides them through a delivery system that connects users with thousands of dealerships and manufacturers. With a product range that has rapidly swelled from around 100,000 items to 1.7 million, the company's systems need to provide both high performance and high availability. To support that expansion, AS ONE created its core business systems using a combination of Oracle

**AS ONE**
Osaka, Japan

**EMPLOYEES:**
485

**REVENUE:**
US$559 million

**ORACLE PRODUCTS:**
Oracle Database 12*c*
Oracle Database Exadata Express Cloud Service
Oracle WebLogic Suite
Oracle Enterprise Manager Cloud Control 12*c*

Shinichi Hakoda, manager of the IT Promotion department at AS ONE, believes that becoming an early adopter of the Oracle Database 12*c* Release 2–based Oracle Database Exadata Express Cloud Service is giving the company an edge in being able to deliver advanced API-based services to its customers. “We can be a step ahead of our competitors,” he says.

Database, Oracle WebLogic Suite, and Oracle Enterprise Manager Cloud Control 12*c*.

AS ONE started receiving requests from dealerships for inventory information to be linked to the company’s systems in real time, instead of updated once a day. To support that need, the company began looking for a solution that would provide real-time links while minimizing the burden on the company’s core infrastructure.

That’s when Shinichi Hakoda, manager of the IT Promotion department at AS ONE, learned about Oracle Database Exadata Express Cloud Service and the fact that Oracle Database 12*c*

## Oracle Cloud at Customer: Bringing It All Home

Companies are using Oracle Database 12*c* in different ways—both in the cloud and on premises—depending on their business needs and overall technology strategy.

And those businesses that are required to meet certain regulatory or other requirements that prevent them from moving to public cloud data centers are increasingly using Oracle Database 12*c* as part of Oracle Cloud at Customer, which provides an Oracle-managed cloud behind a customer's corporate firewall.

"We took all our infrastructure-as-a-service, platform-as-a-service, and software-as-a-service technologies, and we're making them available for running in a customer data center," says Andy Mendelsohn, executive vice president for database server technologies at Oracle. As with the public cloud model, customers subscribe to the hardware and software cloud service that they need. "They can grow it on demand, shrink it on demand, and so on," says Mendelsohn. "It works just like public cloud, but it's on their premises."

Not only does this allow businesses to realize the value of the cloud while meeting specific regulatory requirements, but it also provides performance advantages, says Mendelsohn. "The database is located in the same data center and on the same network as the customer's other applications, so there is very low latency," he says.

includes RESTful interfaces that enable access to real-time information via APIs. This allows data access via JSON and other means without the need for an API server, he says. "From the development perspective, it's possible to develop REST services with only SQL knowledge and screen configurations," which allows a high level of developer productivity, he says.

"Especially the new-generation web developers are making heavy use of REST to call into the database," says Mendelsohn. "Oracle REST Data Services makes it easy to query and operate the database via RESTful interfaces."

Hakoda believes that becoming an early adopter of Oracle Database Exadata Express Cloud Service is giving the company an edge in being able to deliver advanced API-based services to its customers. "We can be a step ahead of our competitors," he says. The company's internal users are already using the information provided via the APIs, and external customers are starting to test it.

Hakoda says that choosing a cloud-

based service for the new system offers several benefits. “From the standpoint of operational costs,” he says, “the latest environment is always offered, and security is supported by Oracle’s high level of technical strength, which we feel is a major advantage.”

Additionally, JSON support in Oracle Database 12*c* will enable AS ONE to rapidly add further real-time information services. “These services make it easy to expand from our current provision of inventory information to things like price information APIs, product information APIs, and delivery period response APIs, which we also feel are extremely strong advantages,” says Hakoda.

AS ONE is also moving toward the use of chatbots as it continues to seek out new technologies to increase efficiency and help customers. “We feel that chatbots have extremely high potential,” Hakoda says, partly because they provide a simple interface that can make systems easier to use and reduce training costs. The company has already linked the inventory information API on Oracle Database Exadata Express Cloud Service with chatbots and is making bot responses to inventory inquiries from within the company. Next step? “We would also like to deploy it externally,” says Hakoda, “and boost customer satisfaction levels.”

“Oracle has led the industry by providing the fastest, most reliable, and highly secure database for organizations of all sizes,” says Mendelsohn. “With the current transition to in-memory computing and cloud, we will continue to innovate to ensure we have the best database technology for the future.”

---

*Mike Faden is a principal at Content Marketing Partners. He has covered business, technology, and science for more than 30 years as a writer, editor, consultant, and analyst. Faden is based in Portland, Oregon.*

---

ILLUSTRATION BY **iSTOCK.COM/KENTOH**;
PHOTOGRAPHY BY
**DARRIN VANSELOW/THE VERBATIM AGENCY**,
**BOB ADLER/THE VERBATIM AGENCY**,
**CHRISTOPHER JUE/THE VERBATIM AGENCY**

## NEXT STEPS

**LEARN** more about Oracle Database cloud services.

**LEARN** more about Oracle Database 12*c*.

ORACLE CODE

# Register Now

## Oracle Code is BACK! | 1-Day, Free Event

***Explore the Latest Developer Trends:***

- DevOps, Containers, Microservices, and APIs
- MySQL, NoSQL, Oracle, and Open Source Databases
- Development Tools and Low Code Platforms
- Open Source Technologies
- Machine Learning, AI, and Chatbots

**Live for the Code**

***Coming to a city near you:***

**developer.oracle.com/code**

ORACLE®

By Dan McGhan

# Using Async Functions

Here's Part 4 in a four-part series on asynchronous Node.js development.

**So far in this asynchronous Node.js series**, I've covered Node.js-style callbacks, the Async module, and promises. In this final part of the series, I'll teach you about async functions (aka async/await). To me, async functions are the most exciting thing to happen to JavaScript since Ajax, because—finally—you can read JavaScript code in a synchronous manner while it executes asynchronously as it always has.

## ASYNC FUNCTIONS OVERVIEW

Async functions are a relatively new feature of JavaScript (and not specific to Node.js). Support for the feature first landed in Node.js v7.6 via an update to the V8 JavaScript engine. Because async functions rely heavily on promises, I recommend you read the previous article if you're not familiar with them.

I like to think of async functions as two parts: async and await. Let's look at each part in turn.

**async.** For the longest time, JavaScript developers could create functions using function statements (which must be named) or function expressions (which are often anonymous).

```
function getNumber() { // Function statement
  return 42;
}

let logNumber = function() { // Function expression
  console.log(getNumber());
}

logNumber(); // 42
```

If you run this script in Node.js, you should see `42` printed to the console.

JavaScript now has asynchronous counterparts to these constructs. Placing the new `async` keyword before the function statement or expression returns an `AsyncFunction` (async function) object.

```
async function getNumber() { // Async function statement
  return 42;
}

let logNumber = async function() { // Async function expression
  console.log(getNumber());
}
```

```
logNumber(); // Promise { 42 }
```

If you run *this* script in Node.js, you should see `Promise { 42 }` printed to the console. As you can see, when async functions are invoked, they return promises rather than the actual values.

For the async version of the script to be the functional equivalent of the first example, I'd have to rewrite it as follows.

```
async function getNumber() { // Async function statement
  return 42;
}

let logNumber = async function() { // Async function expression
  getNumber() // returns a promise
    .then(function(value) {
      console.log(value);
    });
}

logNumber(); // 42
```

With this script, I'm back to logging `42` (rather than `Promise { 42 }`).

Just as you saw with promise chaining, if the async function is completed without error, the promise it returns is resolved. If the function returns a value, that becomes the promise's value. If an error is thrown and goes unhandled, the promise is rejected and the error becomes the promise's value.

Though interesting, returning promises isn't what makes async functions special. You could, after all, return promises from regular functions. What makes async functions special is `await`.

**await.** The `await` operator, which is available only inside an async function, is as close to magic as you'll get in JavaScript. It's like hitting the pause button on your code so that it can wait for a promise to be resolved or rejected before continuing. This is a concept known as a *coroutine*. Coroutines have been available in JavaScript since generator functions were introduced, but async functions make them much more approachable.

Using `await` will not block the main thread. Instead, the currently running call stack, up to the point of `await`, will be completed so that other functions in the callback queue can be executed. When the promise is resolved or rejected, the remaining portion of the code is queued for execution. If the promise was resolved, its value is returned. If the promise was rejected, the rejected value is thrown on the main thread.

Here's a demonstration of `await` that uses `setTimeout` to simulate an async API. I've added some additional console output to help illustrate what's happening.

```
function getRandomNumber() {
  return new Promise(function(resolve, reject) {
    setTimeout(function() {
      const randomValue = Math.random();
      const error = randomValue > .8 ? true : false;

      if (error) {
        reject(new Error('Ooops, something broke!'));
```

```
      } else {
        resolve(randomValue);
      }
    }, 2000);
  });
}

async function logNumber() {
  let number;

  console.log('before await', number);

  number = await getRandomNumber();

  console.log('after await', number);
}

console.log('before async call');

logNumber();

console.log('after async call');
```

When I run this script in Node.js without an error occurring, the output will appear as follows. (I've added a comment where the two-second delay happens.)

```
before async call
before await undefined
after async call
# 2 second delay
after await 0.22454453163016597
```

Note that "after async call" was logged before "after await 0.22454453163016597." Only the remaining code in the async function is paused; the remaining synchronous code in the call stack will finish executing.

If an error is thrown, you'll see `UnhandledPromiseRejectionWarning`, which I covered in the last article. You could handle the rejection with the methods mentioned in that article or by using `try/catch`.

## TRY/CATCH

In the first article in this series, I explained why `try/catch` blocks don't work with asynchronous operations—you can't catch errors that occur outside of the current call stack. But now with async functions, you *can* use `try/catch` for asynchronous operations.

Here's a slightly modified version of the previous script that catches errors that occur in the async API and uses a default value instead.

```
function getRandomNumber() {
  return new Promise(function(resolve, reject) {
    setTimeout(function() {
      const randomValue = Math.random();
      const error = randomValue > .8 ? true : false;
```

```
      if (error) {
        reject(new Error('Ooops, something broke!'));
      } else {
        resolve(randomValue);
      }
    }, 2000);
  });
}

async function logNumber() {
  let number;

  try {
    number = await getRandomNumber();
  } catch (err) {
    number = 42;
  }

  console.log(number);
}

logNumber();
```

If you run that script enough times, you'll eventually get `42` in the output. So, `try/catch` blocks finally work with async operations. Woohoo!

## ASYNC LOOPS

In addition to being able to use `try/catch` blocks again, you can use asynchronous loops, too. In the following example, I use a simple `for` loop that prints three values serially.

```
function getRandomNumber() {
  return new Promise(function(resolve, reject) {
    setTimeout(function() {
      const randomValue = Math.random();
      const error = randomValue > .8 ? true : false;

      if (error) {
        reject(new Error('Ooops, something broke!'));
      } else {
        resolve(randomValue);
      }
    }, 2000);
  });
}

async function logNumbers() {
  for (let x = 0; x < 3; x += 1) {
    console.log(await getRandomNumber());
  }
}
```

```
logNumbers();
```

Running this script in Node.js, you should see three numbers printed to the console every two seconds. There are no third-party libraries and no complicated promise chains, just a simple loop. Loops work again!

## PARALLEL EXECUTION

Clearly, async functions make it easy to do sequential flows and use standard JavaScript constructs with asynchronous operations. But what about parallel flows? This is where `Promise.all` and `Promise.race` come in handy. Because they both return promises, `await` can work with them as with any other promise-based API.

Here's an example that uses `Promise.all` to get three random numbers in parallel.

```
function getRandomNumber() {
  return new Promise(function(resolve, reject) {
    setTimeout(function() {
      const randomValue = Math.random();
      const error = randomValue > .8 ? true : false;

      if (error) {
        reject(new Error('Ooops, something broke!'));
      } else {
        resolve(randomValue);
      }
    }, 2000);
```

```
  });
}

async function logNumbers() {
  let promises = [];

  promises[0] = getRandomNumber();
  promises[1] = getRandomNumber();
  promises[2] = getRandomNumber();

  Promise.all(promises)
    .then(function(values) {
      console.log(values);
    })
    .catch(function(err) {
      console.log(err);
    });
}

logNumbers();
```

Because `Promise.all` rejects its promise if any promise passed in is rejected, you might need to run the script a few times to see the three random numbers printed out.

## EXECUTING A QUERY WITH ASYNC FUNCTIONS

I'll end this article with one last example of executing a query using the Node.js driver for Oracle Database—only this time using an async function. For brevity, I'm using the async version of an immediately invoked function expression (IIFE).

```
const oracledb = require('oracledb');
const dbConfig = {
  user: 'hr',
  password: 'oracle',
  connectString: 'localhost:1521/orcl'
};

(async function() {
  let conn; // Declared here for scoping purposes

  try {
    conn = await oracledb.getConnection(dbConfig);

    console.log('Connected to database');

    let result = await conn.execute(
      'select *
      from employees',
      [], // no binds
      {
```

```
      outFormat: oracledb.OBJECT
    }
  );

  console.log('Query executed');
  console.log(result.rows);
} catch (err) {
  console.log('Error in processing', err);
} finally {
  if (conn) { // conn assignment worked, need to close
    try {
      await conn.close();
      console.log('Connection closed');
    } catch (err) {
      console.log('Error closing connection', err);
    }
  }
}
})();
```

Because the driver's asynchronous APIs already support promises, you can use the `async` operator without any additional work. To me, this version of the query example is the simplest, and it doesn't hurt that it uses the fewest lines of code as well.

I hope you now have a better grasp of async functions and are as excited as I am about using them!

---

*Dan McGhan is the Oracle developer advocate for JavaScript and HTML5. He enjoys sharing the passion he's developed for JavaScript and HTML5 with others.*

PHOTOGRAPHY BY
**CAREY KIRKELLA/THE VERBATIM AGENCY**

## NEXT STEPS

**LEARN** more about JavaScript and Oracle.

By Steven Feuerstein

ORACLE DATABASE

# Just Say No, Unless You Can't

Dynamic SQL is powerful, flexible, and fraught with risk. Use it only when absolutely necessary.

**Whether you run Oracle Database in the cloud or in a virtual machine**, you will need to write SQL statements to both fetch from and change the contents of tables. And when you want that SQL to run as quickly as possible, be as secure as possible, and be as easy to maintain as possible, you will write PL/SQL APIs in the form of packages *around* your SQL statements.

So far, so good. But then you need to decide the following: how am I going to write that SQL inside PL/SQL? You have two basic choices:

- Static SQL
- Dynamic SQL

A SQL statement is *static* (also referred to as *embedded SQL*) when the statement is parsed at the time your PL/SQL program unit is compiled. Here is an example of a code block with static SQL:

```
DECLARE
   l_count INTEGER;
BEGIN
   SELECT COUNT(*) INTO l_count
      FROM user_objects;
  DBMS_OUTPUT.PUT_LINE ('Object count = ' || l_count);
END;
```

A SQL statement is *dynamic* when the statement is parsed at runtime. Here is the dynamic SQL version of the previous code block:

```
DECLARE
   l_count   INTEGER;
BEGIN
   EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM user_objects'
      INTO l_count;
   DBMS_OUTPUT.put_line ('Object count = ' || l_count);
END;
```

Compared to languages such as Java and JavaScript, it is very easy to write both static and dynamic SQL in a PL/SQL block. But that doesn't mean that both types of SQL are equal and can or should be used interchangeably in your code.

In fact, I suggest the following very simple rule for deciding whether to write static or dynamic SQL:

*Construct and execute SQL at runtime only when you have to.*

There are three very important reasons to follow this rule:

**Security.** Dynamic SQL opens the door to SQL injection, which can lead to data corruption and the leaking of sensitive data. It is *impossible* to inject malicious code into a static SQL statement.

**Performance.** Although the overhead of executing dynamic SQL has gone way down over the years, it is certainly still faster to use static SQL. It's also easier to optimize static SQL, because you can analyze explain plans before they execute in production and modify your SQL accordingly—right from your code editor.

**Maintainability.** The code you write to support dynamic SQL is more—literally *more* code—*and* harder to understand and maintain. Don't make your life (and the lives of developers coming after you) any more difficult than it has to be.

In this article, I explore various inappropriate or unnecessary uses of dynamic SQL that I have come across over the years. If, as you read a section, you hear alarms ringing in your head along the lines of "Gee, that looks like something I wrote last week" or "Wait a minute—that's exactly what Joe wants to put into production next week," it's time to take action and convert that dynamic SQL to static SQL.

Your users and your coworkers will thank you.

## CHANGING VALUES DO NOT DYNAMIC SQL MAKE

You need to write dynamic SQL when you don't have all the information you need to construct and parse the statement at compile time.

You do not, however, need to know the *values* of bind variables (for example, the specific department ID or the name of the customer) in order to parse a SQL statement.

Consider the following function:

```
FUNCTION name_from_id (id_in IN INTEGER)
   RETURN VARCHAR2
   AUTHID DEFINER
IS
   c_table_name   CONSTANT VARCHAR2 (100) := 'the_table';
   l_the_name              the_table.the_name%TYPE;
BEGIN
   EXECUTE IMMEDIATE
      'select the_name from ' || c_table_name || ' where id = ' || id_in
      INTO l_the_name;

   RETURN l_the_name;
END;
```

## ORACLE'S LIVE SQL

Oracle's Live SQL provides developers a free and easy online way to test and share SQL and PL/SQL application development concepts.

At first glance, it sure seems to need to be dynamic. The author of this program is concatenating the name of the table at runtime and completing the WHERE clause with the concatenation of the ID value.

On closer inspection, however, the table name is not actually dynamic. It is a constant value of "the_table." So concatenation is not needed there.

The developer apparently believed, further, that because the value of the ID can change, it needs to be concatenated to the SQL statement; therefore, it has to be done dynamically. That is *not* true. The only parts of a SQL statement that must be concatenated are pieces of syntax (for example, when you sometimes need a GROUP BY but other times you don't) or object names, such as a table name or a column name—when and only when those names change at runtime.

The values of bind variables, however, do *not* need to be concatenated. And, in fact, there's nothing dynamic about the query in this function. It can—and certainly *should*—be rewritten as follows:

```
FUNCTION name_from_id (id_in IN INTEGER)
   RETURN VARCHAR2
   AUTHID DEFINER
IS
   l_the_name   the_table.the_name%TYPE;
BEGIN
   SELECT the_name
     INTO l_the_name
     FROM the_table
    WHERE id = id_in;

   RETURN l_the_name;
END;
```

The bottom line is that every time you come across a program that contains EXECUTE IMMEDIATE or calls to DBMS_SQL, look closely at the SQL statement being constructed. Ask yourself, "What don't I know at the time of compilation?"

If the answer is, "only the values to be assigned to a column" or "the values used in a Boolean expression in the WHERE clause," and so on, there is no need for dynamic SQL.

## SAY NO TO "ONE PACKAGE FITS ALL" DYNAMIC SQL APIS

Over the years, I have occasionally received emails from developers that simultaneously delight and dismay me: delight, because I love to see developers who exult in creating generic utilities for themselves and others to use, and dismay, because the generic utilities they created should never be used.

A great example of such a utility is a package that can be used to execute "any" SQL statement needed for an application, dynamically. The body of the email usually sounds something like this:

"Rather than teach everyone on the team how to write SQL, I will give them a few packaged procedures to execute, and those procedures will construct and execute all the SQL for them. They will be more productive and can avoid really bad SQL."

Typically, the specification of such a package might look like this:

```
PACKAGE exec_any_sql AUTHID CURRENT_USER
IS
   PROCEDURE delete_from (schema_in      IN     VARCHAR2 DEFAULT USER,
                          table_in       IN     VARCHAR2,
                          where_in       IN     VARCHAR2 DEFAULT NULL,
                          rowcount_out      OUT INTEGER);

   PROCEDURE insert_into (schema_in      IN     VARCHAR2 DEFAULT USER,
                          table_in       IN     VARCHAR2,
                          columns_in     IN     DBMS_SQL.varchar2_table,
                          values_in      IN     DBMS_SQL.varchar2_table,
                          rowcount_out      OUT INTEGER);
```

```
   PROCEDURE update_in (schema_in      IN     VARCHAR2 DEFAULT USER,
                        table_in       IN     VARCHAR2,
                        where_in       IN     VARCHAR2 DEFAULT NULL,
                        columns_in     IN     DBMS_SQL.varchar2_table,
                        values_in      IN     DBMS_SQL.varchar2_table,
                        rowcount_out      OUT INTEGER);
END;
```

(The package body for exec_any_sql is located in **Listing 1**.)

And if I want to delete some row from table XYZ, I simply execute this procedure call:

```
exec_any_sql.delete_from (table_in => 'XYZ', where_in => 'xyz_id = 15');
```

How cool is that? It is not cool at all! Seriously, this is a very bad idea.

First, it is virtually impossible to support the full range of syntax available for these data manipulation language (DML) statements—and I haven't even *tried* to support SELECTs.

Second, it is all but impossible to optimize the performance of SQL statements constructed this way.

Third, no one will ever want to use it. They will hate the person who wrote it. You will hate yourself if *you* wrote it.

This sort of utility is way *too* generic to be anything but a disaster waiting to happen to your application. You might be tempted to write such utilities, but do so only as an exercise to satisfy your curiosity. Never try to use such a utility in a real application.

## SKIN-DEEP DYNAMIC SQL

Sometimes a program's requirements (or implementation) seem at first glance to be an obvious candidate for dynamic SQL. Only with further analysis—driven by that key rule to use dynamic SQL only when *necessary*—do you realize that a static SQL implementation is possible and, therefore, preferable.

Suppose you came across this function in your application's source code:

```
FUNCTION name_from_id (table_in   IN VARCHAR2,
                       id_in      IN INTEGER)
   RETURN VARCHAR2
   AUTHID DEFINER
IS
   l_the_name   VARCHAR2 (32767);
BEGIN
   EXECUTE IMMEDIATE
      'select the_name from ' ||
         table_in ||
      ' where id = ' || id_in
      INTO l_the_name;

   RETURN l_the_name;
END;
```

It looks like the developer had no choice, right? EXECUTE IMMEDIATE is required, because the name of the table wasn't known at the time of compilation—a classic requirement for a dynamic SQL implementation.

But a single function does not an application make. Either this function is being called or it should be dropped. If it is being called, you should check to see *where* and *how* the function is used and see if that usage justifies the dynamic SQL implementation.

You could do a text search through your code (via an editor such as Sublime) or an object search (in Oracle SQL Developer). You could also use the PL/Scope in Oracle Database if you gathered identifier information across your source code.

Suppose that after doing this analysis, you find that the function is called twice as follows:

```
l_name := name_from_id (table_in => 'TABLE1', id_in => l_id);
l_recent_name := name_from_id (table_in => 'TABLE2', id_in => l_most_recent_id);
```

Now, you *could* shrug and say, "Yep, there are two different table names. I need to use dynamic SQL." And then you could get to work on changing the function so that it is not quite so vulnerable to a SQL injection attack (with, among other things, the DBMS_ASSERT package).

But that would be a mistake. What you *should* think and say is this: "What? Just two different tables? I don't need dynamic SQL for that. That's just being lazy."

And then you could (and I will, in this article) pursue one of two alternatives:

- Create two different functions.
- Change to static SQL inside that single function.

**Two different functions.** Really, why not write two different functions? It's so easy and fast to write PL/SQL functions that call SQL. With a small amount of work, I can "fork" that single function into two, as follows:

```
FUNCTION name_from_table1_id (id_in IN INTEGER)
   RETURN VARCHAR2
   AUTHID DEFINER
IS
   l_the_name   table1.the_name%TYPE;
BEGIN
   SELECT the_name
     INTO l_the_name
     FROM table1
    WHERE id = id_in;

   RETURN l_the_name;
END;

FUNCTION name_from_table2_id (id_in IN INTEGER)
   RETURN VARCHAR2
   AUTHID DEFINER
IS
   l_the_name   table2.the_name%TYPE;
BEGIN
   SELECT the_name
     INTO l_the_name
     FROM table2
    WHERE id = id_in;
```

```
   RETURN l_the_name;
END;
```

There's no more EXECUTE IMMEDIATE, and no more dynamic SQL. It's true that I now have more code to debug and maintain, but this is very simple code. It should not cause any headaches.

Ah, but what if I then need the same functionality for 3, 4, or 10 tables? Doesn't that get to be too much copying and pasting? You could argue both ways on this question.

SQL statements make up the heart of any Oracle Database application. Having a separate function for each distinct query is a small price to pay for clarity and careful optimization.

If, however, you worry about the proliferation of functions, you could also get "fancy" within a *single* function and support any number of tables.

**One function, but no dynamic SQL.** If you don't want two, three, or many functions for the "same" functionality (for example, get the name for ID), fine. Keep them all in one subprogram, but switch from dynamic SQL to conditional logic.

```
FUNCTION name_from_id (table_in   IN VARCHAR2,
                       id_in      IN INTEGER)
   RETURN VARCHAR2
   AUTHID DEFINER
IS
   l_the_name   VARCHAR2 (32767);
BEGIN
   CASE table_in
```

```
      WHEN 'TABLE1'
      THEN
         SELECT the_name
           INTO l_the_name
           FROM table1
          WHERE id = id_in;
      WHEN 'TABLE2'
      THEN
         SELECT the_name
           INTO l_the_name
           FROM table2
          WHERE id = id_in;
      ELSE
         raise_application_error (
            -20000,
            'name_from_id does not support fetching from ' || table_in);
   END CASE;

   RETURN l_the_name;
END;
```

With this approach, you simply add another WHEN clause to the CASE statement whenever you want to support another table.

I do *not* recommend this; separate functions make more sense, both in the short term and in the long term. But this is still better than a function that unnecessarily introduces dynamic SQL into the mix.

## FINDING YOUR DYNAMIC SQL

OK; so I've convinced you to use dynamic SQL only when absolutely necessary. And when you do use it, you will protect yourself from SQL injection through the use of bind variables and DBMS_ASSERT. This is great advice for all the future code you write.

But what about the tens of thousands of lines of *existing* code in your application? How can you find, fix, and possibly remove the dynamic SQL in that code?

The best way to do that is to use the PL/Scope tool in Oracle Database 12*c* Release 2. PL/Scope, first introduced in Oracle Database 11*g*, is a compiler tool that gathers information about all the identifiers (things with names) in your code. It makes it easy to answer questions such as

- Where is this variable *modified* in my program unit?
- Which exceptions are declared but never used (raised or handled)?

With Oracle Database 12*c* Release 2, however, PL/Scope has been extended to also analyze the SQL statements inside your PL/SQL program units. This means you can now answer even *more* interesting questions, such as

- Which SQL statements appear more than once in my code, and where are those duplicates?
- Where are the SQL statements with hints?
- Which programs execute UPDATE statements for specific tables?

And, most relevant to this article, where is dynamic SQL executed in my source code?

Native dynamic SQL can occur through two statements: EXECUTE IMMEDIATE and OPEN (for example, OPEN cursor_variable FOR 'SELECT ...'). Even without PL/Scope, you could search the ALL_SOURCE data dictionary view for keywords, as in the following:

```
SELECT src.owner, src.name
  FROM all_source src
 WHERE UPPER (text) like '%EXECUTE%IMMEDIATE%
```

But you could get an awful lot of false positives—and miss any number of actual usages. (What if, for example, "EXECUTE" is on one line and "IMMEDIATE" is on another?)

Instead, you can and should enable PL/Scope whenever you connect to your session with the following statement:

```
ALTER SESSION SET plscope_settings='identifiers:all, statements:all'
```

Then, compile your program units with this setting active. Then, execute the following statement:

```
SELECT st.owner, st.object_name
  FROM all_statements st
 WHERE st.TYPE IN ('EXECUTE IMMEDIATE', 'OPEN')
```

And then go hunting with a checklist of questions:

- Am I concatenating values instead of using bind variables?
- Once I convert to bind variables, is there anything left that is dynamic?
- Is it a fancy-pants implementation that could be done more simply with static SQL?
- If the code must remain dynamic, is it protected from SQL injection attacks?

## SUMMARY

The PL/SQL language makes it easy to implement dynamic SQL requirements, but *easy* should not be considered an invitation to do so. Because of the practical and potential downsides of dynamic SQL, using it should be considered a technique of last resort. Fortunately, the need for dynamic SQL might seem compelling at first, but it often disappears after a little bit of analysis.

So remember: say no to dynamic SQL—unless you can't.


---

*Steven Feuerstein is a developer advocate for Oracle, specializing in PL/SQL. Feuerstein's books, including* Oracle PL/SQL Programming*, videos, and more than 1,500 quizzes at the Oracle Dev Gym (devgym.oracle.com) provide in-depth resources for Oracle Database developers.*

PHOTOGRAPHY BY **ANDREA MANDEL**

## NEXT STEPS

**LEARN** more about dynamic SQL in PL/SQL.

**EXPLORE** dynamic SQL scripts on the Live SQL website.

# Exadata Your Way


**PUBLIC**
Exadata Cloud Service

**Same Standards**
**Same Technology**
**Unified Management**

**HYBRID**
Exadata Cloud
at Customer

**PRIVATE**
Exadata On Premises

## Public, Hybrid, or Private: YOU Choose

ORACLE®

By Connor McDonald

ORACLE DATABASE 12*c*

# Old Dog, New Tricks

Take advantage of SQL extensions for hierarchy processing.

**Many blog posts and magazine articles are about** the newest releases of a database and the newest software features, focusing on new facilities and functions that have not been seen in commercial usage yet. But in this article, I'll take a look back at a feature that has been in Oracle Database since way back in Oracle 2: processing hierarchical data by using the CONNECT BY syntax in SQL. And I'll focus on some of the additional functionality that has been delivered over the course of the major releases of Oracle Database.

## FUNDAMENTALS

Most database developers are aware of the fundamentals of the hierarchical query syntax. It is one of the first things they learn after they understand rudimentary data manipulation language (DML) elements (SELECT, INSERT, UPDATE, DELETE).

**Listing 1** shows an example of the most typical introduction to the hierarchical query syntax. The two elements that identify a SQL query as a hierarchical query

are the CONNECT BY and the START WITH clauses. The syntax reads very much like the intended action of the query:

**START WITH.** The query will "start with" a nominated position (or set of rows) in the hierarchy. In this case, it's the MANAGER being null, meaning an employee who has no reporting manager, in other words, the CEO or top-level employee in the company. In this case, KING is the only person in this company who has no manager.

**CONNECT BY.** The query will then "connect" the current row to a subsequent row, "by" a nominated relationship. Once employee KING, with EMPNO = 7389, has been identified, the syntax CONNECT BY PRIOR EMPNO = MGR instructs the query processing to search for a row in the employee (EMP) table where the MGR column equals that of the employee number EMPNO of the *prior* row, namely KING. This continues recursively down through the hierarchy until it has been fully traversed for the defined conditions.

**Code Listing 1:** Introduction to hierarchical query syntax

```
SQL> select empno, ename, mgr
  2  from   emp
  3  connect by prior empno = mgr
  4  start  with mgr is null;

     EMPNO ENAME             MGR
__________ __________ __________
      7839 KING
      7566 JONES            7839
      7788 SCOTT            7566
      7876 ADAMS            7788
```

```
7902 FORD            7566
7369 SMITH           7902
7698 BLAKE           7839
7499 ALLEN           7698
7521 WARD            7698
7654 MARTIN          7698
7844 TURNER          7698
7900 JAMES           7698
7782 CLARK           7839
7934 MILLER          7782
```

An easy way to demonstrate the traversal of the hierarchy is to use the pseudo-column called LEVEL, which returns the "depth" within the hierarchy of the row that is currently being traversed. Combining LEVEL with RPAD to provide a simple indentation, **Listing 2** demonstrates how you can more clearly visualize the depth of the hierarchy starting with KING and navigating down to employees at the deepest level of the hierarchy, such as ADAMS, WARD, MARTIN, TURNER, and JAMES. The hierarchy is often described as analogous to a tree structure, with the deepest levels of the hierarchy being "leaf" elements, and the other levels—that is, those that sit logically above the leaves—being "branches."

**Code Listing 2:** Demonstrating the hierarchy depth with RPAD

```
SQL> select empno, rpad(' ',level*3)||ename ename, mgr
  2  from   emp
  3  connect by prior empno = mgr
  4  start  with mgr is null;
```

```
     EMPNO ENAME                          MGR
__________ ______________________________ __________

      7839    KING
      7566       JONES                    7839
      7788          SCOTT                 7566
      7876             ADAMS              7788
      7902          FORD                  7566
      7369             SMITH              7902
      7698       BLAKE                    7839
      7499          ALLEN                 7698
      7521          WARD                  7698
      7654          MARTIN                7698
      7844          TURNER                7698
      7900          JAMES                 7698
      7782       CLARK                    7839
      7934          MILLER                7782
```

Another way to visualize the hierarchy is with the SYS_CONNECT_BY_PATH function, which concatenates the nominated expression values within the hierarchy as it is traversed. **Listing 3** shows the same employee hierarchy using SYS_CONNECT_BY_PATH to display the relationships.

**Code Listing 3:** Demonstrating the hierarchy depth with SYS_CONNECT_BY_PATH

```
SQL> select sys_connect_by_path(ename,'-')
  2  from   emp
```

```
  3  start with mgr is null
  4  connect by prior empno = mgr;

SYS_CONNECT_BY_PATH(ENAME,'-')
------------------------------------------------------
-KING
-KING-JONES
-KING-JONES-SCOTT
-KING-JONES-SCOTT-ADAMS
-KING-JONES-FORD
-KING-JONES-FORD-SMITH
-KING-BLAKE
-KING-BLAKE-ALLEN
-KING-BLAKE-WARD
-KING-BLAKE-MARTIN
-KING-BLAKE-TURNER
-KING-BLAKE-JAMES
-KING-CLARK
-KING-CLARK-MILLER
```

## BEYOND THE BASICS

For many developers, navigating the hierarchy and displaying the depth is where their usage and understanding of hierarchical queries come to an end, but there are many more features available. One of the reasons for reticence about using hierarchical queries is that in explorations beyond the simple examples using the EMP

table, real-world data problems can cause a hierarchical query to crash with an error and return no data at all.

For example, in **Listing 4** the manager for employee KING is set to 7499. This introduces a logical corruption in the hierarchy definition for the employees in this company. Traversing through the hierarchy in one direction indicates that KING is the manager of ADAMS, but traversing in the opposite direction through the hierarchy indicates that ADAMS is the manager of KING. There is now a cyclic relationship in the hierarchy. To continuously follow the CONNECT BY syntax would infinitely traverse this hierarchy and never come to an end. To avoid an endless loop, the database will return an error when CONNECT BY encounters such a cyclic relationship.

**Code Listing 4:** Error stops endless loop in a hierarchical query

```
SQL> update emp set mgr = 7499
  2  where  ename = 'KING';

1 row updated.

SQL> select sys_connect_by_path(ename,'-')
  2  from   emp
  3  start with ename = 'KING'
  4  connect by prior empno = mgr
  5  /

ERROR:
ORA-01436: CONNECT BY loop in user data
```

To avoid this loop and error, there is a NOCYCLE extension to the CONNECT BY syntax. With NOCYCLE, CONNECT BY will traverse the hierarchy as much as possible. **Listing 5** shows NOCYCLE in action, preventing an infinite loop and not generating an error in the SQL execution.

**Code Listing 5:** NOCYCLE prevents infinite loop and error

```
SQL> select sys_connect_by_path(ename,'-')
  2  from   emp
  3  start with mgr is null
  4  connect by NOCYCLE prior empno = mgr
  5  /

SYS_CONNECT_BY_PATH(ENAME,'-')
--------------------------------------------------

-KING
-KING-JONES
-KING-JONES-SCOTT
-KING-JONES-SCOTT-ADAMS
-KING-JONES-FORD
-KING-JONES-FORD-SMITH
-KING-BLAKE
-KING-BLAKE-ALLEN
-KING-BLAKE-WARD
-KING-BLAKE-MARTIN
-KING-BLAKE-TURNER
-KING-BLAKE-JAMES
```

```
-KING-CLARK
-KING-CLARK-MILLER
```

Another extension for hierarchical SQL queries is the ORDER BY clause, so that rather than ordering the entire result set, you can nominate an ordering system within *each* branch of the tree. The ORDER SIBLINGS BY clause allows for ordering to be isolated within one branch at a time. **Listing 6** shows the five leaf entries of ALLEN, JAMES, MARTIN, TURNER, and WARD ordered alphabetically under the KING-BLAKE branch, without having any impact on the entries of other branches.

**Code Listing 6:** ORDER SIBLINGS BY isolates ordering to one branch

```
SQL> select rpad(' ',level)||sys_connect_by_path(ename,'-') tree,
  2      empno, dname
  3  from  emp e, dept d
  4  where d.deptno = e.deptno
  5  start with mgr is null
  6  connect by prior empno = mgr
  7  ORDER SIBLINGS BY ENAME;

TREE                                 EMPNO DNAME
_____________________________________ _____ ______________
 -KING                                7839 ACCOUNTING
  -KING-BLAKE                         7698 SALES
   -KING-BLAKE-ALLEN                  7499 SALES
   -KING-BLAKE-JAMES                  7900 SALES
   -KING-BLAKE-MARTIN                 7654 SALES
```

```
 -KING-BLAKE-TURNER                7844 SALES
 -KING-BLAKE-WARD                  7521 SALES
-KING-CLARK                        7782 ACCOUNTING
 -KING-CLARK-MILLER                7934 ACCOUNTING
-KING-JONES                        7566 RESEARCH
 -KING-JONES-FORD                  7902 RESEARCH
  -KING-JONES-FORD-SMITH           7369 RESEARCH
 -KING-JONES-SCOTT                 7788 RESEARCH
  -KING-JONES-SCOTT-ADAMS          7876 RESEARCH
```

Note in the previous example that even if a table contains a hierarchical relationship, it can still be joined to other tables.

## FILTERING VERSUS NAVIGATION

Navigating the hierarchy is quite distinct from the filtering that comes with WHERE clause predicates in the SQL statement. To demonstrate this, the 14 rows in the EMP table have been duplicated to be a set of rows for multiple geographic regions, as shown in **Listing 7**. There are now 42 rows in the table: the standard set of 14 rows in each of the NORTH, SOUTH, and EAST regions.

**Code Listing 7:** Duplicating EMP table rows for multiple regions

```
SQL> alter table emp add region varchar2(10);


Table altered.
```

```
SQL> update emp set region = 'NORTH';

14 rows updated.

SQL>
SQL> insert into emp
  2  select
  3    empno, ename, job, mgr, hiredate, sal, comm, deptno,
  4    decode(r,1,'SOUTH',2,'EAST') region
  5  from emp,
  6       ( select rownum r from dual connect by level <= 2 )
  7  where region = 'NORTH';

28 rows created.

SQL>
SQL> select region, count(*)
  2  from   emp
  3  group by region;

REGION       COUNT(*)
__________ __________

NORTH              14
SOUTH              14
EAST               14
```

Let's assume a requirement to obtain the hierarchy for the 14 rows for the NORTH region only. It would seem intuitive to use a query identical to **Listing 2**, but with an additional predicate to restrict the REGION to NORTH. **Listing 8** shows the result.

**Code Listing 8:** Revisiting Listing 2 with WHERE clause

```
SQL> select empno, rpad(' ',level*3)||ename ename, mgr
  2  from   emp
  3  where  region = 'NORTH'
  4  connect by prior empno = mgr
  5  start  with mgr is null;

     EMPNO ENAME                                          MGR
---------- ------------------------------------------ ----------
      7839    KING
      7566       JONES                                     7839
      7788          SCOTT                                  7566
      7876             ADAMS                               7788
      ...
      ...
      7782       CLARK                                     7839
      7934          MILLER                                 7782
      7934          MILLER                                 7782
      7934          MILLER                                 7782

136 rows selected.
```

Note that the output returned is not 14 rows but 136 rows, even though a predicate to restrict the results to the NORTH region only was applied.

To understand the cause of this problem, consider the very first NORTH region row that is being processed. The START WITH clause is MGR IS NULL, so *three* King rows, one for each region, will be located, and then the CONNECT BY operation will commence. The CONNECT BY clause is PRIOR EMPNO = MGR, so for each KING record, *three* related employees, one for each region, will be connected to, which rapidly increases the number of rows returned as the query traverses through the hierarchy. Only *after* the hierarchical relationship is established is the WHERE clause predicate applied, so even though the result set contains only rows for REGION = NORTH, a large amount of redundant work was performed doing erroneous connections to the other regions because the CONNECT BY and START WITH syntax did not do the following:

- Restrict itself to starting with only the NORTH region
- Make sure that, when connecting to the next entry in the hierarchy, the row maps from the same region to the subsequent row's region

**Listing 9** shows the corrected syntax for this query. It starts with managers that are null (MGR IS NULL) but *also* only in the NORTH region. It then ensures that when connecting from one manager to a subsequent employee, the definition for connecting also includes a common region. This is analogous to the way the ANSI join syntax is processed, namely, that there is a distinct difference between the hierarchical connecting condition and the WHERE clause predicates that will be applied subsequently as filters.

This gives rise to a common but incorrect criticism of hierarchical queries: they can run too slowly or consume too many resources. Commonly the cause is that the

CONNECT BY information is incomplete, processing many more rows than need to be processed.

**Code Listing 9:** Limiting hierarchical queries in CONNECT BY and START WITH

```
SQL> select empno, rpad(' ',level*3)||ename ename, mgr
  2  from   emp
  3  connect by prior empno = mgr
  4         and  region = 'NORTH'
  5  start  with mgr is null
  6         and  region = 'NORTH';

     EMPNO ENAME                                          MGR
---------- ------------------------------------------ ----------
      7839    KING
      7566       JONES                                     7839
      7788          SCOTT                                  7566
      7876             ADAMS                               7788
      7902          FORD                                   7566
      7369             SMITH                               7902
      7698       BLAKE                                     7839
      7499          ALLEN                                  7698
      7521          WARD                                   7698
      7654          MARTIN                                 7698
      7844          TURNER                                 7698
      7900          JAMES                                  7698
```

```
      7782       CLARK                                7839
      7934          MILLER                            7782

14 rows selected.
```

You might ask, "What, then, is the use of a WHERE clause in a hierarchical query?" The presence of a WHERE clause is still valid, not to define the hierarchy but to define *filtering* of the data. For example, once the NORTH region hierarchy has been instantiated, the WHERE clause requirement might be to include only those employees earning above a certain SALARY.

## HIERARCHY ATTRIBUTES

When dealing with hierarchical structures, there are common questions that need to be answered about the data. For the sample data in this article, questions might be along the lines of

- Is this employee a manager? That is, are there any employees who report to this employee?
- Which employee sits at the top of the hierarchy when starting a query from a given leaf?

With the initial set of sample data in this article, KING is the sole employee who sits at the top of the hierarchy, but in **Listing 10**, some rows have been added to demonstrate that this does not have to be the case for all hierarchies. Employee DENNIS has been added who also has no manager; that is, he sits at the same top level as KING. So it is possible that employees at the very bottom of the hierarchy structure might ultimately traverse up to a different highest manager.

**Code Listing 10:** Adding employees—including one who has no manager

```
SQL> insert into emp ( empno, ename, mgr, deptno)
  2  values (1000,'DENNIS',null, 10);

1 row created.

SQL> insert into emp ( empno, ename, mgr, deptno)
  2  values (1001,'CHRIS',1000, 10);

1 row created.

SQL> insert into emp ( empno, ename, mgr, deptno)
  2  values (1002,'CRAIG',1001, 10);

1 row created.
```

There are additional pseudofunctions available to answer the "is this a manager?" and "which employee is at the top of the hierarchy?" questions as the hierarchy is traversed:

- The CONNECT_BY_ROOT function, followed by a column name, returns the attribute of the root-level node up to which this current node ultimately traverses.
- The CONNECT_BY_LEAF function returns a zero if a value pertains to a branch in the hierarchy, and it returns 1 if it is a leaf—that is, if there are no subordinate entries as defined by the CONNECT BY clause.

- The CONNECT_BY_ISCYCLE function allows the identification of which entries in the hierarchy are currently involved within a cyclic relationship. It can be used when dealing with NOCYCLE in a CONNECT BY clause.
  **Listing 11** demonstrates these three pseudofunctions.

**Code Listing 11:** Three pseudofunctions that answer questions

```
SQL> select  connect_by_root ename root,
  2          connect_by_isleaf     leaf,
  3          connect_by_iscycle    cyc,
  4          sys_connect_by_path(ename,'-') full
  5  from  emp
  6  start with ename in ('KING','DENNIS')
  7  connect by NOCYCLE prior empno = mgr;

ROOT          LEAF    CYC FULL
__________ _______ _______ ____________________________

KING             0       0 -KING
KING             0       0 -KING-JONES
KING             0       0 -KING-JONES-SCOTT
KING             1       0 -KING-JONES-SCOTT-ADAMS
KING             0       0 -KING-JONES-FORD
KING             1       0 -KING-JONES-FORD-SMITH
KING             0       0 -KING-BLAKE
KING             1       1 -KING-BLAKE-ALLEN
KING             1       0 -KING-BLAKE-WARD
KING             1       0 -KING-BLAKE-MARTIN
```

```
KING            1       0 -KING-BLAKE-TURNER
KING            1       0 -KING-BLAKE-JAMES
DENNIS          0       0 -DENNIS
DENNIS          0       0 -DENNIS-CHRIS
DENNIS          1       0 -DENNIS-CHRIS-CRAIG
```

## SUMMARY

There is much more functionality for the hierarchical query syntax in SQL over and above the basic START WITH and CONNECT BY clauses. These extensions allow a far more sophisticated analysis of hierarchical data in your database systems.

My next article will explore a new mechanism to navigate hierarchical data structures without using the CONNECT BY syntax at all.

*Connor McDonald is an Oracle developer advocate for SQL. His passions are database design, SQL, and PL/SQL, and he can answer your database questions on AskTom.*

PHOTOGRAPHY BY
**SABINE ALBERS/THE VERBATIM AGENCY**

## NEXT STEPS

**LEARN** more about hierarchical query processing.

**DOWNLOAD** Oracle Database 12*c*.

ORACLE REST DATA SERVICES

# Get Your REST; POST Your SQL

Use REST-enabled SQL in Oracle REST Data Services.

By Jeff Smith

**Representational State Transfer (REST)** is today's dominant software architectural style for creating modern, scalable web services, and JavaScript Object Notation (JSON) is the most popular data-interchange format for RESTful web services that use the REST architecture.

Oracle REST Data Services accepts RESTful web service uniform resource identifiers (URIs) and directs them to the appropriate SQL statement or PL/SQL block, returning the output in either the JSON or comma-separated values (CSV) format. Oracle REST Data Services is easy to install and configure, and it empowers auto-generated REST endpoints for tables, views, and PL/SQL.

Until now, the SQL or database workload that was mapped to a URI that Oracle REST Data Services ultimately ran on an HTTPS call has always been predefined. This article introduces a new feature, REST-enabled SQL, in Oracle REST Data Services version 17.4 that enables the execution of ad hoc queries and scripts. This

article will show you how to configure Oracle REST Data Services for this feature and demonstrate a REST-enabled SQL scenario.

## INSTALLATION AND CONFIGURATION

You can download version 17.4 of Oracle REST Data Services from Oracle Technology Network. Installation of this Oracle REST Data Services version is no different from earlier versions; however, the REST-enabled SQL feature is *not* enabled by default.

The REST-enabled SQL feature can be enabled by default for all connection pools servicing Oracle Database instances, or it can be enabled for a particular database. To enable it for all databases, add the following line to the DEFAULTS.XML file:

```
<entry key="restEnabledSql.active">true</entry>
```

Otherwise, add the line above to the appropriate XYZ_PU.XML connection pool configuration file. If your environment includes multiple databases configured for Oracle REST Data Services, it is recommended that you enable REST-enabled SQL only on connection pools where the feature is required.

Restart Oracle REST Data Services, and the new REST-enabled SQL endpoint (_/sql/) will then be available on REST-enabled schemas.

## SECURITY

REST-enabled SQL supports two security schemes: web server authentication and database user authentication. Web server authentication allows SQL workloads to be executed on any REST-enabled schema for a web server user that has been granted the “SQL Developer” role.

Starting with Oracle REST Data Services 17.4, database user authentication means that HTTPS requests can also be authenticated via a database user and password. For REST-enabled SQL, the database user can authenticate SQL workloads only for that REST-enabled schema's _/sql/ endpoint.

The following scenarios will use the new database user authentication scheme. To make this available for your database, you need to REST-enable a schema first.

With Oracle REST Data Services installed and configured for your database, run this PL/SQL block as the user who is to be REST-enabled, substituting "HR" and "hr" with the name of the schema you want to REST-enable:

```
BEGIN
    ORDS.ENABLE_SCHEMA(p_enabled => TRUE,
                       p_schema => 'HR',
                       p_url_mapping_type => 'BASE_PATH',
                       p_url_mapping_pattern => 'hr',
                       p_auto_rest_auth => FALSE);
    commit;
END;
```

Note that the `p_url_mapping_pattern` value must be in lowercase.

### SCENARIO: RUNNING A SIMPLE SQL STATEMENT

As the database user for the schema you just REST-enabled, run a simple SQL statement and examine the JSON response.

To access the REST-enabled SQL endpoint, use the following URI to issue an HTTPS POST request to the web server on which Oracle REST Data Services is running:

```
https://webserver:port/ords/schema/_/sql/
```

To issue a POST request, you will need to use a command-line interface such as CURL or an application such as Postman.

Here is the CURL command to issue a POST request that contains a simple SELECT query against the HR schema:

```
curl -i -X POST --user HR:password -H "Content-Type: application/sql" -k -d
select department_id, max(salary) from employees group by department_id'
http://localhost:8888/ords/hr/_/sql
```

Oracle REST Data Services confirms that the "/schema/" portion of the URI ("hr," in this case) represents a schema that is REST-enabled. It then attempts to log in to the database as the HR user using the supplied password and, if it is successful, it runs the attached SQL statement as that user in the database.

The results from the execution of the SQL statement are then transformed to JSON and returned to the calling client, which in this case is the CURL session.

Below, the response for this request has been formatted for easier readability:

```
{
    "env": {
        "defaultTimeZone": "America/New_York"
    },
    "items": [
        {
            "statementId": 1,
```

```
                "statementType": "query",
                "statementPos": {
                    "startLine": 1,
                    "endLine": 2
                },
                "statementText": "select department_id, max(salary) from employees
group by department_id",
                "response": [],
                "result": 0,
                "resultSet": {
                    "metadata": [
                        {
                    "columnName": "DEPARTMENT_ID",
                    "jsonColumnName": "department_id",
                    "columnTypeName": "NUMBER",
                    "precision": 4,
                    "scale": 0,
                    "isNullable": 1
                },
                {
                    "columnName": "MAX(SALARY)",
                    "jsonColumnName": "max(salary)",
                    "columnTypeName": "NUMBER",
                    "precision": 0,
                    "scale": -127,
                    "isNullable": 1
```

```
        }
    ],
    "items": [
        {
            "department_id": 100,
            "max(salary)": 12000
        },
        {
            "department_id": 30,
            "max(salary)": 11000
        },
        {
            "department_id": null,
            "max(salary)": 7000
        },
        {
            "department_id": 90,
            "max(salary)": 24000
        },
        {
            "department_id": 20,
            "max(salary)": 13000
        },
        {
            "department_id": 70,
            "max(salary)": 10000
```

```
    },
    {
        "department_id": 110,
        "max(salary)": 12000
    },
    {
        "department_id": 50,
        "max(salary)": 8200
    },
    {
        "department_id": 80,
        "max(salary)": 14000
    },
    {
        "department_id": 40,
        "max(salary)": 6500
    },
    {
        "department_id": 60,
        "max(salary)": 9000
    },
    {
        "department_id": 10,
        "max(salary)": 4000
    }
],
```

```
                "hasMore": false,
                "limit": 500,
                "offset": 0,
                "count": 12
            }
        }
    ]
}
```

## INTERPRETING THE RESPONSE

The response to the request includes two components: the metadata and the results. The metadata describes the following:

- The time zone of the web server on which the Java Virtual Machine for Oracle REST Data Services is running
- A listing of the statements executed
- The items included in the results, including the column names and data types that have been transposed to JSON

The results for each statement are then included with each listing (in this example, there is only a single statement to be executed). The results also include the following:

- Any text included with the response, such as “Table TEST created”
- For data manipulation language (DML), the number of rows affected
- Whether there are more results to be paged
- The hardcoded limit for results, which is defined in your Oracle REST Data Services configuration
- The number of results included in the items for the current response

Note that any DATE or TIMESTAMP included in a result set will include the TIMEZONE for the corresponding JSON value using the Zulu time zone derived from the Oracle REST Data Services installation.

## POSSIBLE USES

REST-enabled SQL provides access to remote data—"remote" being a database where no Oracle NET Services access is provided by HTTPS. Suppose you have a web application that is set up to work with one Oracle Database instance, but you need to access data in another database. Currently, that would require one of the following:

- A DB_LINK from the first database to the second database
- The ability to open a second Oracle NET Services connection to the second database

With REST-enabled SQL, you now have an HTTPS alternative. If Oracle REST Data Services is available and configured for the second database, you can just access the data via an HTTPS POST request. You could also write a PL/SQL routine to make the data available as a VIEW that goes across HTTPS.

There is also a new Type 3 JDBC driver that uses REST-enabled SQL endpoints to make JDBC connections available over HTTPS. You can download the Oracle REST Data Services JDBC driver from the Oracle REST Data Services download page. In this blog post, I show how to use this new driver with the SQLcl command-line interface for Oracle Database to connect to a database via Oracle REST Data Services to run your queries and scripts.

## SUMMARY

Oracle REST Data Services provides a REST interface for Oracle Database. It can serve custom RESTful services that run any SQL or PL/SQL block, and it supports

REST-enabling tables and views for a full create, read, update, and delete (CRUD) API. Oracle REST Data Services recently added support for remote procedure call (RPC) for PL/SQL objects and now, with version 17.4, Oracle REST Data Services provides HTTPS access for running ad hoc database workloads.

GitHub includes a sample web page for trying many different scenarios using the new _/sql/ endpoint. Extract the files to your Oracle REST Data Services or web server HTDOCS folder, and load the HTML page. (This approach is much easier to work with than making CURL requests.)

---

*Jeff Smith, a senior principal product manager in Oracle's Database Development Tools group, is responsible for Oracle SQL Developer and Oracle SQL Developer Data Modeler.*

## NEXT STEPS

**DOWNLOAD** Oracle REST Data Services.

**TRY** Oracle Database Cloud services.

By Melanie Caffrey

ORACLE DATABASE

# Limited Profiles and Private References

Part 13 in a second series on the basics of the relational database and SQL

**This article is the thirteenth, and last, in a series that helps you** build on the fundamentals that you learned in the 12-part SQL 101 series in *Oracle Magazine*. The previous Beyond SQL 101 article, "Becoming Privileged and Creating Synonymously," taught you how to create, alter, and drop users and how users relate to schemas. You learned what system privileges and object privileges are and how privileges are granted and revoked. Additionally, you saw that roles are collections of privileges that can be granted and revoked in a similar fashion as individual privileges. Lastly, you witnessed how privileges and roles should be granted on a discretionary as-needed basis for security and ease of administration.

In this article, you will

- Learn more about object privileges
- Discover the difference between public and private synonyms
- See how to extend privileges to others
- Get an introduction to profiles

To try out the examples in this series, you need access to an Oracle Database instance. If necessary, download and install an Oracle Database edition for your operating system. I recommend installing Oracle Database, Enterprise Edition 12*c* Release 2 (12.2.0.1.0). If you install the Oracle Database software, choose the installation option that enables you to create and configure a database. A new database, including sample user accounts and their associated schemas, will be created for you. (Note that SQL_201 is the user account to use for the examples in this series; it's also the schema in which you'll create database tables and other objects.) When the installation process prompts you to specify schema passwords, enter and confirm passwords for the SYS and SYSTEM users and make a note of them.

Finally—whether you installed the database software from scratch or have access to an existing Oracle Database instance—download, unzip, and execute the SQL script to create the tables for the SQL_201 schema that are used for this article's examples. (View the script in a text editor for execution instructions.)

## THE OBJECT OF YOUR INTENTION

As you learned in the previous Beyond SQL 101 article, "Becoming Privileged and Creating Synonymously," *object privileges* allow you the right to perform actions including but not limited to

- Querying a table not owned by the current database user
- Performing data manipulation language commands on such a table

- Accessing an integer from a sequence not owned by the current database user

**Listing 1** re-creates the BEYOND_101 user and demonstrates the result received when the user attempts to query the EMPLOYEE table. Although the table exists in schema SQL_201, the query result is

```
ORA-00942: table or view does not exist
```

This error means that until the BEYOND_101 user has been granted the appropriate access for the object, Oracle Database does not know that the table or view is an object upon which the BEYOND_101 user may perform actions. This user must be granted the necessary object privilege from the table's owner.

**Code Listing 1:** A user must be granted query access to a database table the user does not own

```
SQL> set lines 60
SQL> connect / as sysdba
Connected.
SQL> alter session set container=bynd;

Session altered.

SQL> CREATE USER beyond_101
  2  IDENTIFIED BY 201
  3  DEFAULT TABLESPACE users
  4  TEMPORARY TABLESPACE temp
  5  QUOTA 100M ON USERS;
```

```
User created.

SQL> GRANT CREATE SESSION, CREATE TABLE TO beyond_101;

Grant succeeded.

SQL> connect beyond_101@bynd
Enter password:
Connected.

SQL> select first_name||' '||last_name
  2    from employee
  3  order by last_name;
  from employee
       *
ERROR at line 2:
ORA-00942: table or view does not exist
```

**Listing 2** shows how the SQL_201 user grants the necessary object privilege to the BEYOND_101 user so that the query may succeed. Note, however, that after the SQL_201 user has granted the SELECT object privilege for the EMPLOYEE table to the BEYOND_101 user, when the BEYOND_101 user reattempts the query action, the action results in the same error as that received in **Listing 1**. The same error takes place because, in the absence of a *synonym* (discussed later in this article), each Oracle Database user must *qualify* the name of any object it does not own. **Listing 3** demonstrates how to qualify the name of the EMPLOYEE table so that the

BEYOND_101 user may successfully query it. To qualify the name of an object, you provide the schema name and the object name separated by the period (.) character, for example, SQL_201.EMPLOYEE.

**Code Listing 2:** The SQL_201 user grants the SELECT object privilege for the EMPLOYEE table to the BEYOND_101 user

```
SQL> connect sql_201@bynd
Enter password:
Connected.

SQL> grant select on employee to beyond_101;
Grant succeeded.

SQL> connect beyond_101@bynd
Enter password:
Connected.
SQL> select first_name||' '||last_name
  2    from employee
  3  order by last_name;
  from employee
       *
ERROR at line 2:
ORA-00942: table or view does not exist
```

**Code Listing 3:** Qualify the name of the queried table by appending the schema name

```
SQL> select first_name||' '||last_name
  2    from sql_201.employee
  3  order by last_name;

FIRST_NAME||''||LAST_NAME
_____________________________________________________________

Lori Dovichi
Emily Eckhardt
Roger Friedli
Betsy James
Thomas Jeffrey
Sasha Meyer
Matthew Michaels
Frances Newton
Donald Newton
Don Rose
Gerald Sowell
Mary Streicher
Marcy Tamra
Theresa Wong
mark leblanc
michael peterson

16 rows selected.
```

## BY ANY OTHER NAME

If you do not wish to qualify an object name for which you've been granted access, you can create a *synonym*. A synonym is an alias for a qualified object name. **Listing 4** shows the SQL statement you use to create a synonym. The syntax, at its simplest, is

```
CREATE [OR REPLACE] SYNONYM <synonym name> FOR <qualified object name>
```

**Code Listing 4:** Create a synonym to use as an alias for a qualified object name

```
SQL> connect / as sysdba
Connected.

SQL> alter session set container=bynd;

Session altered.

SQL> grant create synonym to beyond_101;

Grant succeeded.

SQL> connect beyond_101@bynd;
Enter password:
Connected.

SQL> CREATE SYNONYM employee FOR sql_201.employee;

Synonym created.
```

```
SQL> select first_name||' '||last_name
  2    from employee
  3  order by last_name;

FIRST_NAME||''||LAST_NAME
_____________________________________________________________

Lori Dovichi
Emily Eckhardt
Roger Friedli
Betsy James
Thomas Jeffrey
Sasha Meyer
Matthew Michaels
Frances Newton
Donald Newton
Don Rose
Gerald Sowell
Mary Streicher
Marcy Tamra
Theresa Wong
mark leblanc
michael peterson

16 rows selected.
```

A synonym created using this syntax is called a *private synonym*, because it is private to the schema that created it. No other schema can use the synonym created in **Listing 4**, because it is a synonym created within the BEYOND_101 schema as another name for the SQL_201.EMPLOYEE qualified table name. All synonyms are private unless you specify the PUBLIC keyword. A *public synonym* is visible to every user in your database. For example, the SQL_201 and BEYOND_101 users are able to query the DBA_TABLES data dictionary view without having to qualify the view name with the SYS schema name, because public synonyms were created for all of the data dictionary views when the database and the data dictionary were created.

It is good practice to never create public synonyms for your application objects. Suppose you have a table named PRODUCT in your schema, but a table named PRODUCT also exists in another schema. Also suppose that the PRODUCT table in the other schema has a public synonym that points to it. When you issue a statement against the PRODUCT table without qualifying the schema, Oracle Database will first try to access the PRODUCT table within your schema. If it is unable to access the PRODUCT table within your schema, it will then attempt to refer to the public synonym for the PRODUCT table in the other schema. If the two PRODUCT tables are structurally different or contain different data, trying to access them without qualifying them could yield unexpected or erroneous results.

## AN EXTENSION OF RIGHTS

To grant an object privilege to another user, you must either

- Be the owner of the object, or
- Have received the privilege with the additional keywords WITH GRANT OPTION

The WITH GRANT OPTION addition to an object privilege grant allows the grantee the ability to pass that same privilege on to others. **Listing 5** illustrates how the

WITH GRANT OPTION privilege addition allows one user the ability to pass an object privilege on to another. The BEYOND_101 user does not have the ability to query the DBA_TABLES data dictionary view. The SQL_201 user tries to grant the access, but has access only to query the view, not to grant access to it to others.

The SYS user is the owner of the data dictionary and, therefore, is able to grant query access to the SQL_201 user with the ability to pass that object privilege on to others. After the SQL_201 user has been granted an object privilege WITH GRANT OPTION, it can pass that privilege on to others, even though it does not own the object for which the grant is being made.

**Code Listing 5:** A user granted the WITH GRANT OPTION privilege addition can pass an object privilege to another user

```
SQL> connect beyond_101@bynd
Enter password:
Connected.

SQL> select table_name
  2    from dba_tables
  3   where owner = 'SQL_201';
  from dba_tables
       *
ERROR at line 2:
ORA-00942: table or view does not exist

--The SQL_201 user is able to query from the DBA_TABLES data dictionary view
SQL> connect sql_201@bynd
```

```
Enter password:
Connected.

SQL> select table_name
  2    from dba_tables
  3   where owner = 'SQL_201'
  4  order by table_name;

TABLE_NAME
------------------------------------------------------------------------------

ANNUAL_REVIEW
DEPARTMENT
EMPLOYEE
EMPLOYEE_CTAS
EMPLOYEE_EXTRA
EMPLOYEE_IDENTITY
EMPLOYEE_SUBSET

7 rows selected.

--The SQL_201 user has been granted query access, but cannot yet pass it on
SQL> grant select on dba_tables to beyond_101;
grant select on dba_tables to beyond_101
                *
ERROR at line 1:
ORA-01031: insufficient privileges
```

```
SQL> connect / as sysdba
Connected.

SQL> alter session set container=bynd;

Session altered.

--The SYS user grants the SQL_201 user the ability to pass on the object privilege
SQL> grant select on dba_tables to sql_201 WITH GRANT OPTION;

Grant succeeded.

SQL> connect sql_201@bynd
Enter password:
Connected.

--This time, the grant succeeds
SQL> grant select on dba_tables to beyond_101;

Grant succeeded.

SQL> connect beyond_101@bynd
Enter password:
Connected.

--Similarly, now the BEYOND_101 user can query the DBA_TABLES data dictionary view
```

```
SQL> select table_name
  2    from dba_tables
  3   where owner = 'SQL_201'
  4  order by table_name;

TABLE_NAME
------------------------------------------------------------------------------

ANNUAL_REVIEW
DEPARTMENT
EMPLOYEE
EMPLOYEE_CTAS
EMPLOYEE_EXTRA
EMPLOYEE_IDENTITY
EMPLOYEE_SUBSET

7 rows selected.
```

**Listing 6** demonstrates how the WITH ADMIN OPTION privilege addition can be used to allow a user to pass system privileges on to others as well. The BEYOND_101 user unsuccessfully attempts to create a sequence for a newly created table. After the SYS user grants the SQL_201 user the ability to create sequences and pass such a privilege on to others, the SQL_201 user grants the CREATE SEQUENCE system privilege to the BEYOND_101 user, and the BEYOND_101 user is then able to successfully create a sequence.

**Code Listing 6:** A user granted the WITH ADMIN OPTION privilege addition can pass a system privilege to another user

```
SQL> create table test_me (id number);

Table created.

--The BEYOND_101 user has not been granted the ability to create a sequence
SQL> create sequence test_me_seq;
create sequence test_me_seq
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> connect / as sysdba
Connected.
SQL> alter session set container=bynd;

Session altered.

SQL> grant create sequence to sql_201 WITH ADMIN OPTION;

Grant succeeded.

SQL> connect SQL_201@bynd
Enter password:
Connected.
```

```
SQL> grant create sequence to beyond_101;

Grant succeeded.

SQL> connect beyond_101@bynd
Enter password:
Connected.

--This time, the sequence is created successfully.
SQL> create sequence test_me_seq;

Sequence created.
```

## KNOWING YOUR LIMITS

By default, every Oracle Database user account is constrained in terms of the amount of database resources it is allowed to consume and how its password features are managed. These constraints are managed through a named *profile*. A profile can enforce settings including but not limited to the following:

- How long a user account may be logged in to the database but not actively working before the account is automatically logged out
- How many times a user may attempt to log in to the database with an incorrect password before the account is locked
- How many individual sessions a single user can be logged in to for the database at any given time

The statements in **Listing 7** illustrate how to discover the profile currently associated with a given user account, as well as how to determine the limit restrictions

assigned to the profile. The SQL_201 user has been assigned the DEFAULT profile.

Every user is assigned the DEFAULT profile unless another named profile is explicitly assigned to the user during account creation or through an ALTER USER statement. Currently, the SQL_201 user can have no more than 10 failed login attempts before its user account is locked, and it must be unlocked by a privileged user such as a database administrator. Additionally, the SQL_201 user is currently allowed to be connected to the database but not actively executing any statements for an unlimited amount of time. These two limit settings are outlined in bold in the final result set for **Listing 7**.

**Code Listing 7:** Discover the name of and resource limits for the profile assigned to the SQL_201 user

```
SQL> connect / as sysdba
Connected.
SQL> alter session set container=bynd;

Session altered.

SQL> select profile
  2    from dba_users
  3   where username = 'SQL_201';

PROFILE
------------------------------------------------------------
DEFAULT
```

```
SQL> select resource_name, resource_type, limit
  2    from dba_profiles
  3   where profile = 'DEFAULT'
  4  order by resource_type, resource_name;

RESOURCE_NAME                  RESOURCE LIMIT
_____________________________ _________ ___________

COMPOSITE_LIMIT                KERNEL   UNLIMITED
CONNECT_TIME                   KERNEL   UNLIMITED
CPU_PER_CALL                   KERNEL   UNLIMITED
CPU_PER_SESSION                KERNEL   UNLIMITED
IDLE_TIME                      KERNEL   UNLIMITED
LOGICAL_READS_PER_CALL         KERNEL   UNLIMITED
LOGICAL_READS_PER_SESSION      KERNEL   UNLIMITED
PRIVATE_SGA                    KERNEL   UNLIMITED
SESSIONS_PER_USER              KERNEL   UNLIMITED
FAILED_LOGIN_ATTEMPTS          PASSWORD 10
PASSWORD_GRACE_TIME            PASSWORD 7
PASSWORD_LIFE_TIME             PASSWORD 180
PASSWORD_LOCK_TIME             PASSWORD 1
PASSWORD_REUSE_MAX             PASSWORD UNLIMITED
PASSWORD_REUSE_TIME            PASSWORD UNLIMITED
PASSWORD_VERIFY_FUNCTION       PASSWORD NULL

16 rows selected.
```

The statement in **Listing 8** demonstrates how to create a profile and assign password and kernel limit restrictions to it. The statement in **Listing 9** outlines how to explicitly assign a profile to a user. The setting for FAILED_LOGIN_ATTEMPTS has been reduced to five, and the setting for IDLE_TIME has been reduced to 20 minutes. Note that any limit not explicitly set in the SQL_201_PROFILE profile exists with the value of "DEFAULT." A profile limit that has a DEFAULT value has the same value as that specified for the same-named limit in the DEFAULT profile.

**Code Listing 8:** Create a profile with assigned password and kernel setting limits

```
SQL> connect / as sysdba
Connected.
SQL> alter session set container=bynd;

Session altered.

SQL> CREATE PROFILE sql_201_profile
  2  LIMIT
  3  FAILED_LOGIN_ATTEMPTS 5
  4  IDLE_TIME             20;

Profile created.
```

**Code Listing 9:** Assign a profile to a user and review all assigned settings

```
SQL> alter user sql_201 PROFILE sql_201_profile;

User altered.
```

```
SQL> select resource_name, resource_type, limit
  2    from dba_profiles
  3   where profile = 'SQL_201_PROFILE'
  4  order by resource_type, resource_name;

RESOURCE_NAME                  RESOURCE LIMIT
_____________________________ _________ ___________

COMPOSITE_LIMIT                KERNEL   DEFAULT
CONNECT_TIME                   KERNEL   DEFAULT
CPU_PER_CALL                   KERNEL   DEFAULT
CPU_PER_SESSION                KERNEL   DEFAULT
IDLE_TIME                      KERNEL   20
LOGICAL_READS_PER_CALL         KERNEL   DEFAULT
LOGICAL_READS_PER_SESSION      KERNEL   DEFAULT
PRIVATE_SGA                    KERNEL   DEFAULT
SESSIONS_PER_USER              KERNEL   DEFAULT
FAILED_LOGIN_ATTEMPTS          PASSWORD 5
PASSWORD_GRACE_TIME            PASSWORD DEFAULT
PASSWORD_LIFE_TIME             PASSWORD DEFAULT
PASSWORD_LOCK_TIME             PASSWORD DEFAULT
PASSWORD_REUSE_MAX             PASSWORD DEFAULT
PASSWORD_REUSE_TIME            PASSWORD DEFAULT
PASSWORD_VERIFY_FUNCTION       PASSWORD DEFAULT

16 rows selected.
```

## CONCLUSION

This article taught you more about object privileges. You learned how to grant them and how grantees might need to qualify an object name with its schema name in order to take advantage of their granted access. Additionally, you explored the difference between public and private synonyms, and why it is good practice to defer to using private synonyms in your application code. You discovered how to extend privileges to others when privileges have been granted with the WITH GRANT OPTION or WITH ADMIN OPTION options. Lastly, you saw how profiles are used to limit a user's resource and password settings.

This article concludes the Beyond SQL 101 series. You've learned basic relational database concepts and many SQL coding constructs beyond those discussed in the SQL 101 series. Be sure to continue to read the documentation and try existing and new Oracle Database features. Thank you for being readers of *Oracle Magazine* and of the Beyond SQL 101 series. As you continue writing SQL, my hope is that you enjoy it as much as I do.

*Melanie Caffrey is a senior development manager at Oracle. She is a coauthor of* Beginning Oracle SQL for Oracle Database 12*c (Apress, 2014),* Expert PL/SQL Practices for Oracle Developers and DBAs *(Apress, 2011), and* Expert Oracle Practices: Oracle Database Administration from the Oak Table *(Apress, 2010).*

PHOTOGRAPHY BY **RAY NG**

## NEXT STEPS

**READ** SQL 101, Parts 1–12.

**READ** more Beyond SQL 101.

**LEARN** more about relational database design and concepts.