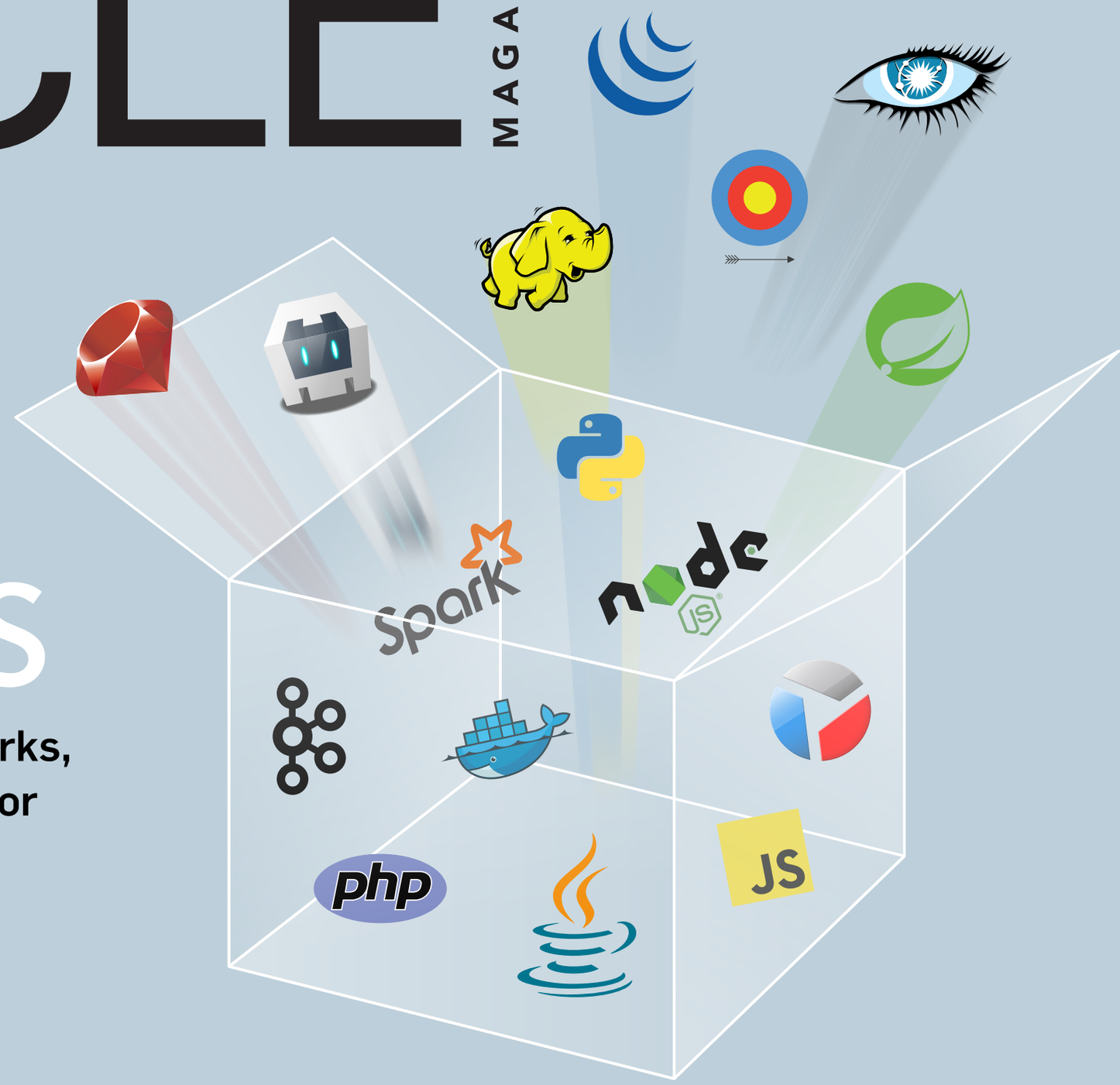
ORACLE MAGAZINE

JANUARY/FEBRUARY 2017

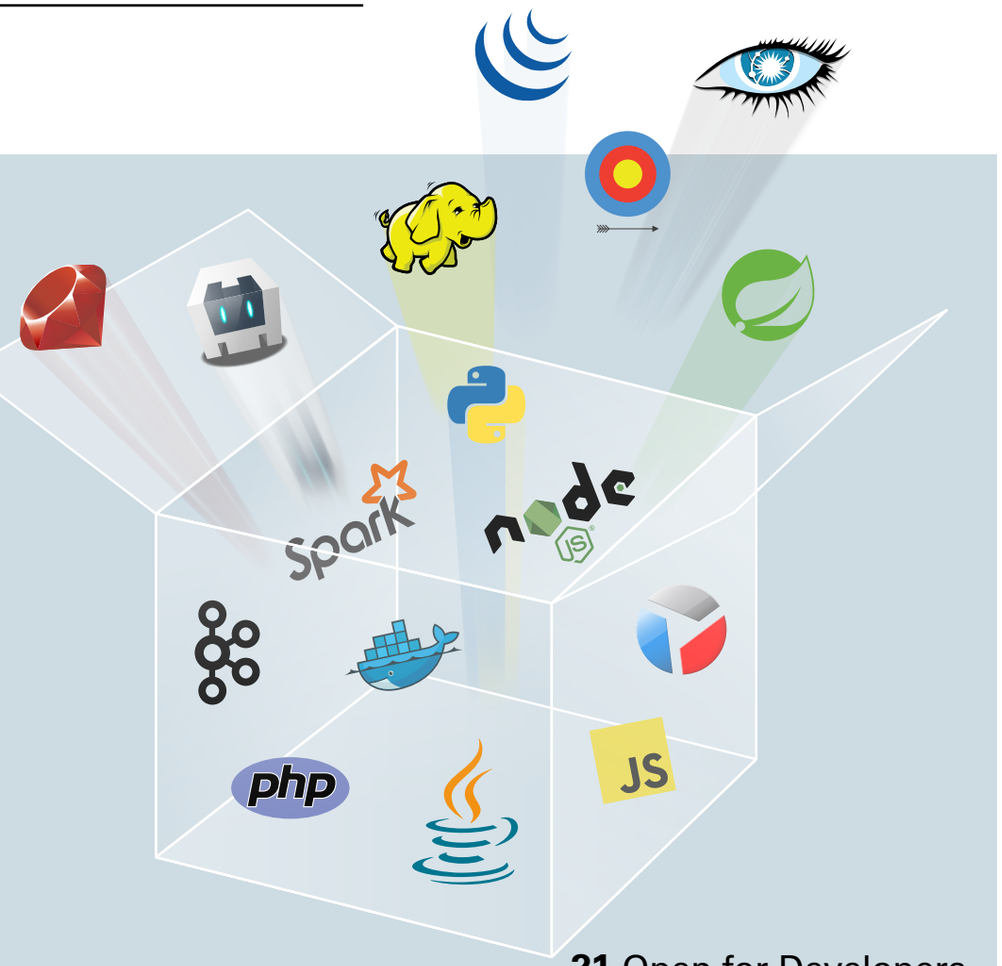


OPEN FOR DEVELOPERS

Choose the experience—the frameworks, the languages, the tools, and more—for modern application development

ORACLE

CONTENTS



21 Open for Developers

ORACLE
MAGAZINE

FEATURES

40 Engineer Assembles Developer Dream Team

35 Personal Development

UP FRONT

CONTENTS

17 Peer-To-Peer

COMMUNITY

8 Community Bulletin

TECHNOLOGY

ORACLE MAGAZINE

EDITORIAL

Editor in Chief Tom Haunert

Managing Editor Jan Rogers

Editorial Director Robert Preston

Contributing Editors and Writers Blair Campbell, Leslie Steere

Copy Editors Claire Breen, Blair Campbell, Eva Langfeldt, Karen Perkins

DESIGN

Vice President, Brand Creative Francisco G Delgadillo

Design Director Richard Merchán

Senior Designer Arianna Pucherelli

Senior Production Manager Sheila Brennan

Designer Jaime Ferrand

Production Designer Kathy Cygnarowicz

PUBLISHING

Publisher Jennifer Hamilton +1.650.506.3794

Associate Publisher and Audience Development Director Karin Kinnear +1.650.506.1985

Audience Development Manager Jennifer Kurtz

ADVERTISING SALES

Western and Central US, LAD, and Canada Tom Cometa +1.510.339.2403

Eastern US and EMEA/APAC Mark Makinney +1.805.709.4745

Mailing-List Rentals Contact your sales representative

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.

Fujitsu M10 Servers

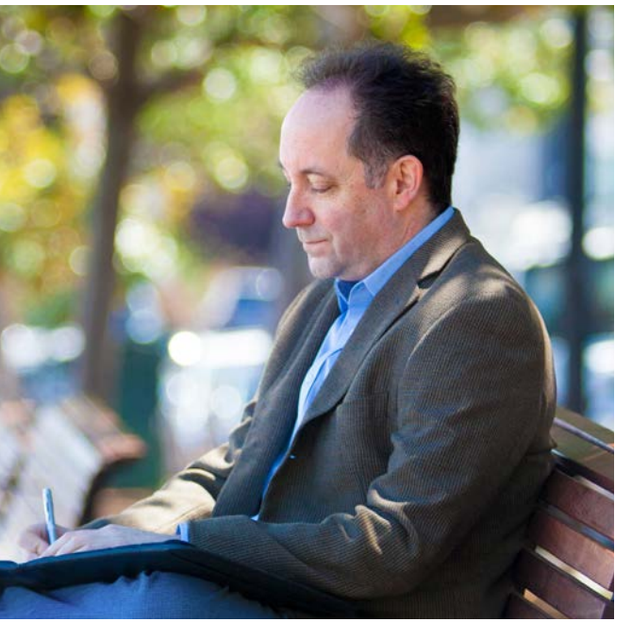
FUJITSU

shaping tomorrow with you

Fujitsu M10 Solutions for
Securing your
Digital Transformation

visit: www.fujitsu.com/sparc/

ORACLE Diamond Partner



Tom Haunert

Hello, Bigger World!

There's a lot more to modern application development than knowing how to say "Hello, World!"—and that's good news.

For developers, creating a "Hello, World!" application is a very simple start, a first step toward understanding the syntax of a programming language. But it's the next step that has gotten a lot more interesting.

Today's enterprise application developers have a "Hello, World!" fluency across multiple languages, including Java, JavaScript, PHP, Python, and Ruby, and multiple frameworks, including Spring Boot, Dropwizard, React.js, and Oracle JET.

Today's developers are also working with different types of databases, so in addition to accessing relational databases via SQL, they're connecting to NoSQL and in-memory databases, sometimes using other new frameworks and languages. And after choosing and learning languages, frameworks, and databases, today's developers are choosing and learning to use open source tools for application containers, microservices, and DevOps—tools such as Apache Spark, Kafka, Vagrant, Puppet, Chef, and Docker.

Oracle Cloud supports all of these developer choices, and that's the message in this issue of *Oracle Magazine*.

In our cover story, "Open for Developers," Alexandra Weber Morales dives into how Oracle supports open, modern, and easy-to-use languages, frameworks, and tools for today's enter-

prise development projects. In "Cloud-Native Delivers," Siddhartha Agarwal, vice president of product management and strategy at Oracle, discusses today's cloud developer challenges and Oracle's open solutions. And "Personal Development" lays out Oracle Code 2017, the 20-city developer event series scheduled to start in March 2017.

To describe the key takeaways from this issue's developer stories another way: Oracle Cloud services are open for today's development projects; cloud-native development saves time and money; and Oracle Code is coming soon to a city near you. Here's one last invitation to attend Oracle Code: The call for papers is open. Send your proposals early and often.

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

TRY Oracle Cloud.

ANSWER the Oracle Code call for papers.

EXPLORE Oracle Developer resources.

LEARN more about Oracle Code 2017.

Data Quality Made Easy. Your Data, Your Way.

Melissa Data 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.MelissaData.com/oracle

Germany
www.MelissaData.de

United Kingdom
www.MelissaData.co.uk

India
www.MelissaData.in

Australia
www.MelissaData.com.au

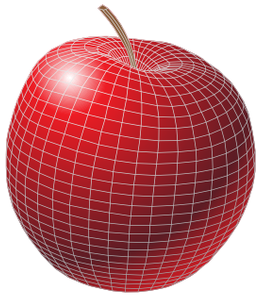
MELISSA DATA®
Your Partner in Global Data Quality

www.MelissaData.com | 1-800-MELISSA

BY ROLAND SMART, VICE PRESIDENT OF COMMUNITY AND SOCIAL MARKETING AT ORACLE

Oracle Cloud Marketplace: Update

Attention developers and technical professionals. Check out Oracle Cloud Marketplace for new applications and services. Get access to more than 450 tools, platforms, and frameworks—all integrated with Oracle Cloud Platform. Whether it's optimizing your backup and recovery mechanisms for Oracle Database or syncing new datasources with Oracle Sales Cloud, you'll find solutions for your latest project.



Teach and Empower the Next Generation

Oracle Technology Network community members are advancing computer science education around the world with the help of Oracle Academy. Seasoned instructors use Oracle Academy resources to introduce Alice—a free and introductory Java development environment—into their curricula. And volunteers who have never taught programming before are learning how to reach and engage future developers at the secondary school level.

If you're interested in empowering the next generation of science, technology, engineering, and math leaders, visit the Oracle Academy website and look into the latest free training sessions. You'll find both in-person and virtual opportunities.

ILLUSTRATIONS BY WES ROWELL

Oracle Code: Join the Developer Roadshow

Beginning in March 2017, Oracle will host Oracle Code, a global developer roadshow including hands-on labs, technical sessions with Oracle experts, and other interactive experiences. Events are scheduled for 20 cities, including San Francisco, São Paulo, Bangalore, and Prague. Sign up to receive the latest roadshow updates.

Oracle Cloud Day: 2017 Events Announced

If you're in the US, save the date for the next Oracle Cloud Day near you. 2017 event locations include Minneapolis, Denver, Los Angeles, Houston, and San Francisco. Learn more and register now.

Developers will find session tracks and networking events devoted to their own unique innovation opportunities and challenges. Presentation topics include reimagining application development and delivery; getting started with cloud-native development; and enabling digital transformation using the latest API management techniques.

GET MORE TIPS

If you missed last issue's Oracle Technology Network Oracle OpenWorld 2016 video recap, check it out as soon as possible, because there's more. Oracle Technology Network Architect Community Manager Bob Rhubart shares 16 new Oracle OpenWorld video interviews with a wide range of subject matter experts.

Tune into a discussion with 2016 Oracle Cloud Platform Innovation Award winner Helsana Group as they describe how they used Oracle Data Integrator, Oracle GoldenGate, and Oracle SOA Suite. And get hands-on advice from Oracle ACE Director Ronald van Luttihuizen on how to use the latest platform-as-a-service design patterns for Oracle Cloud Solutions.

MEETING THE DEMANDS OF TOMORROW AND DELIVERING HIGH-PERFORMANCE NOW

ORACLE Platinum Partner

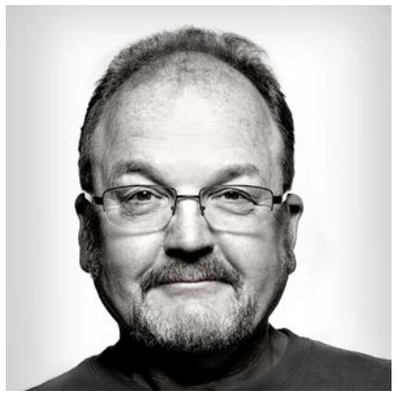
As the leader in **Applications Performance Optimization**, Cybernoor can optimize your Oracle Environment to ensure your system performs well and scales. As the data footprint continues to increase along with the complexity of business requirements, delivering a high performance system is crucial to the success of the business. Contact Cybernoor to learn about our proven performance tuning solutions, and ensure your system is prepared for tomorrow's challenges.

Cybernoor Corporation | www.cybernoor.com | +1.925.924.0400

cybernoor™

APIs: Yesterday and Today

Next-generation interfaces are demanding attention and creating change.



By **Bob Rhubart**

The term *application program interface* is as familiar to anyone involved in software development as free conference backpacks. More commonly referred to as APIs, these reusable chunks of software have existed for decades, and like everything else in IT, APIs have evolved, in both technology and terminology. Today's APIs bear only a passing resemblance to those of decades past, and the differences have made APIs a major factor in today's software solutions.

What separates old-school APIs from their newer counterparts? Oracle ACE Sven Bernhardt, a solution architect at Opitz Consulting in Germany, shares some technical insight. "In pure integration projects, we often work with legacy APIs, provided by the enterprise information systems, which are on the level of 'systems of record,'" Bernhardt explains. "There's often no comprehensive documentation available, and the interface as such is quite complex. The information exchange is usually done using specific data formats that are quite technical and therefore hard to understand. By nature, the interaction pattern is often asynchronous using protocols such as FTP, JMS [Java Message Service], or even SOAP."

The opposite is true for the new generation of APIs, including those used for mobile devices and wearable technologies. "These APIs are clearly described, easy to understand, and usually work with lightweight protocols and data

The new generation of APIs is changing how architects and developers approach what they do.

structures," Bernhardt continues. "The interaction pattern is often synchronous, which is especially important when the interaction with the API is initiated by a human actor, who expects an immediate response."

Oracle ACE Director Sten Vesterli, principal at More Than Code, offers another angle on the generation gap between today's APIs and those of the past.

"An API used to mean methods in a library that you called from your code," he explains. "For example, physical hardware came with an API that you could call from your C code." Today the term refers to web services that can be called by a third party—someone you don't know and who doesn't know you. The changing nature of APIs has brought them into much wider use. "I'm using APIs more, both as consumer and provider," Vesterli says.

Oracle ACE Director Luis Weir, principal at Capgemini UK, tells a similar story. Weir is involved in projects focused on developing mobile apps and responsive web applications, integration with software-as-a-service applications, implementing microservices architectures, and automation. The common thread? "More and more of our work goes into defining, building, managing, and consuming APIs," Weir says.

For Oracle ACE Rolando Carrasco, co-owner and principal SOA architect for S&P Solutions in Mexico, APIs have infiltrated every aspect of his professional life. He says that half the conference sessions he has recently presented are focused on APIs.

"APIs have changed the way I design for customers. They have changed my vocabulary when I try to explain something. They have changed the way I try to identify new opportunities, and the way I do business," Carrasco says.

Among the changes occurring in the wake of the increasing use and importance of APIs is the need for effective API management. "This involves proper documentation, coding examples, and a way for the caller to communicate with

the provider," explains Vesterli. "It also means that the provider needs to think about how to control access."

According to Bernhardt, "API management is a key enabler" for microservices and Internet of Things architectures and a "success factor for digital enterprises of tomorrow." Weir and Carrasco attach enough significance to API management to have coauthored a book on the topic (see the sample chapter link in Next Steps), and both have presented recent conference sessions on API management.

The message is clear. The new generation of APIs is changing how architects and developers approach what they do. How has the evolution of APIs affected your world? Join in the conversation.

Bob Rhubart is the manager of the architect community on Oracle Technology Network, the host of the Oracle Technology Network ArchBeat *podcast series, and the author of the ArchBeat blog.*

PHOTOGRAPHY BY
MICHAEL MCELROY/THE VERBATIM AGENCY

NEXT STEPS

READ a sample chapter from *Oracle API Management 12c Implementation* (Packt Publishing, 2015).

LISTEN to the podcast "API Management Implementation."

the podcast "API Management Roundtable."

COLLABORATE in the community space for API Management.

The Cloud and the Maker Movement

The cloud brings affordable, massively scalable compute power to Maker Faire projects.

Members of the international DIY culture known as the Maker Movement have always been quick to embrace new technologies on which to build their inventions. Their latest darling? The cloud.

And why not? The cloud gives tinkerers and their maker projects the same affordable compute power and array of apps and platform services that it delivers to enterprise users.

Oracle Magazine covered the Oracle Maker Faire at Oracle headquarters in November 2016 and caught up with three makers who are using cloud computing to power and connect their latest projects.

Rick Perotti

INVENTOR OF "HOMUNCULUS"

"When I speak to [Homunculus], it goes out to a cloud service to convert my voice to text and then sends the text back. It then uses RESTful services to talk to Oracle Sales Cloud and have it pull back my day's schedule."

Jasper Potts

INVENTOR OF "THE IOT CHOCOLATE FACTORY"

"Over time the plastic fatigues, and we have a whole bunch of sensors and cameras that collect that data and feed it up to the [Oracle] Internet of Things Cloud Service where we can have the compute power to do machine learning to predict the point where it's going to fail."

Mark Vilrokx

INVENTOR OF "CONNECT ALL THINGS"

"We have a Nerf gun with a chip in it that connects it to the internet. [Then] we figured out that we could add other services to it. The Nerf gun has a Twitter account, and every time it shoots a bullet it sends out a tweet. The Nerf gun is voice-activated. There's a dashboard that tells me if it's online or offline and how many bullets it's fired."

Sponsored Content

KPMG Achieves Oracle Cloud Elite Partner Status

The highest distinction in Oracle PartnerNetwork is awarded to companies that have exceptional experience in Oracle Cloud solutions.

KPMG recently earned a number of prestigious designations and awards from Oracle, including Oracle Cloud Elite Partner status in Oracle PartnerNetwork (OPN) and six Oracle Excellence Awards for Enterprise Resource Planning Cloud Partner of the Year globally and in North America and the Asia-Pacific region. In addition, KPMG is the only Oracle partner to achieve Advanced Specialization status for Cloud Finance, and one of only three Oracle partners with Advanced Specialization for Global Cloud Human Resources. Oracle also appointed its vice chairman of the board of directors, Jeff Henley, to be KPMG's executive sponsor. This is the first time KPMG has had an executive sponsor at Oracle. Linsey Ryan, principal and Oracle leader at KPMG, discusses KPMG's new distinctions, and their benefits for customers.

"With the innovation that the KPMG Powered Enterprise solution brings to the Oracle Cloud space, we have a tremendous opportunity to help clients transform their businesses."

–Linsey Ryan, KPMG

Why was KPMG bestowed with Oracle Cloud Elite Partner status in OPN?

Oracle Cloud Elite Partner status recognizes KPMG's longstanding and successful partnership with Oracle in the cloud market. It acknowledges KPMG's impact on the market, our track record of successful cloud deployments, and our significant investment in Oracle's cloud solutions.

How does achieving Oracle Cloud Elite Partner status benefit your clients?

Clients are looking to engage with a provider who has deep experience and an established track record. Oracle Cloud Elite Partner status is a statement that KPMG is one of Oracle's most strategic partners. We're now in some exclusive company, and such a designation helps clients make more timely and informed business and technology decisions.

Why has KPMG invested in the area of cloud for so many years?

No matter where our clients operate, the cloud has become the place for digital transformation. Nearly a decade ago, KPMG first recognized the cloud's impact on the market and made a significant investment in this space. And now, with the innovation that the KPMG Powered Enterprise solution brings to the Oracle Cloud space, we have a tremendous opportunity to help clients transform their businesses.

Why has KPMG devoted resources to achieving expertise in Oracle's cloud solutions?

It's about opportunity and teaming with a leader. Oracle is one of the world's most advanced enterprise cloud providers with broad and extensive experience in the market. And for a long time, Oracle has been one of KPMG's most strategic industry partners. We've built an entire alliance sales and marketing organization and have ramped up training and enablement around Oracle Cloud including training and certifications, dedicated go-to-market resources, and joint solution development.

What is the role of the cloud in your clients' digital transformation initiatives?

Clients are increasingly moving to the cloud across their businesses to transform their businesses. They realize that the cloud isn't just for sales and marketing functions, but should be leveraged for mission-critical operations to achieve real change.

KPMG

ORACLE Platinum Partner Cloud Elite

For more information, visit **www.kpmg.com**.

BY BLAIR CAMPBELL

The Deep End

Three peers learn by doing, mentoring, and competing.

Sabine Heimsath

its-people

Frankfurt, Germany

ORACLE ACE Associate

Company: its-people, a professional network of self-employed IT entrepreneurs

Job title/description: Senior consultant, working for different customers in the transportation and financial areas

Length of time using Oracle products: More than 15 years

How did you get started in IT? I always had an interest in puzzles, problem-solving, and mathematics. I enjoyed programming BASIC on my father's first computer, and I was very happy when we were offered computer classes at school. So studying computer and management science seemed like a natural choice. After university I talked to several companies, but the Oracle representative was the only one who talked about his company with sparkling eyes. I started working for Oracle Consulting, which meant jumping in at the deep end—but it was perfect for me, and that's still my favorite way of learning

What's your favorite tool on the job? Oracle SQL Developer. I love it because of the huge number of small features that make my life as a developer easier. It's like a Swiss Army Knife. I recommend that everyone invests some time in finding their own set of favorite Oracle SQL Developer features by reading Jeff Smith's blog, or for German readers, our own blog.

Which new features in Oracle Database are you currently finding most valuable? As a metadata enthusiast, I enjoy using PL/Scope for numerous

checks on our code—and I'm really excited about the analysis of SQL statements within PL/SQL, which is new in Oracle Database 12*c* Release 2.

Edward Whalen

Performance Tuning Corporation

Houston, Texas

ORACLE ACE Director

Company: Performance Tuning Corporation, a provider of application and database services

Job title/description: Chief technologist, serving as both a consultant and as a mentor to other consultants

Length of time using Oracle products: 26 years

Which technologies in Oracle Fusion Middleware are you currently finding most valuable? For the last few years I've been working with Oracle GoldenGate, which provides the ability to take data as it's being inserted into or changed within the database and use it to populate downstream data sources in real time. This offers a number of opportunities, including the ability to do stream or real-time analytics on live data. I'm very excited about using it to populate real-time data from Oracle databases into a big data system.

What advice do you have about getting into database development? Get a good book and learn as much as you can, and consider participating in the Oracle ACE Program. It's a great place to find a mentor, or even just occasional advice. The program has provided me with many contacts and the ability to share my knowledge with younger developers and DBAs. Anyone who's interested should stop by the Oracle Technology Network booth at the next

Oracle OpenWorld and meet some Oracle ACEs.

What's the most common cause you see when IT projects go wrong? Not having the right people. You shouldn't hire a Java programmer to develop a database back end for your application, just like you wouldn't hire a DBA to do the front-end Java work. I've personally been brought in to fix scalability problems caused by doing work in Java that should have been done in the database. Retrieving and processing data is best done in the database, not in the application. Every development project should have at least one database developer.

Arturo Viveros

Sysco Middleware
Oslo, Norway

ORACLE ACE

Company: Sysco Middleware, a Norwegian IT company with branches in several local cities, as well as Sweden, Denmark, and Peru
Job title/description: Principal architect, working with Sysco's development teams and customers to deliver scalable IT solutions
Oracle credentials: Oracle SOA Suite 11*g* Certified Implementation Specialist, Oracle WebLogic Server 12*c* Certified Implementation Specialist, and Oracle IT Architecture SOA 2013 Certified Architecture Specialist, with 12 years of experience using Oracle products

How are you using cloud computing these days? As an "average Joe," I use it every day in the shape of SaaS [software as a service] applications such as Dropbox, Spotify, and Netflix, which have changed the way we do regular stuff. On the other hand, as an IT professional, IaaS [infrastructure as a service] and PaaS [platform as a service] are allowing me to design and create a wide range of innovative solutions for organizations undergoing digital transformation.

How about mobile computing? Again, as a regular citizen, it feels like more and more of my life is revolving around what I can do with my mobile devices. As an IT architect and developer, this trend benefits me in the sense

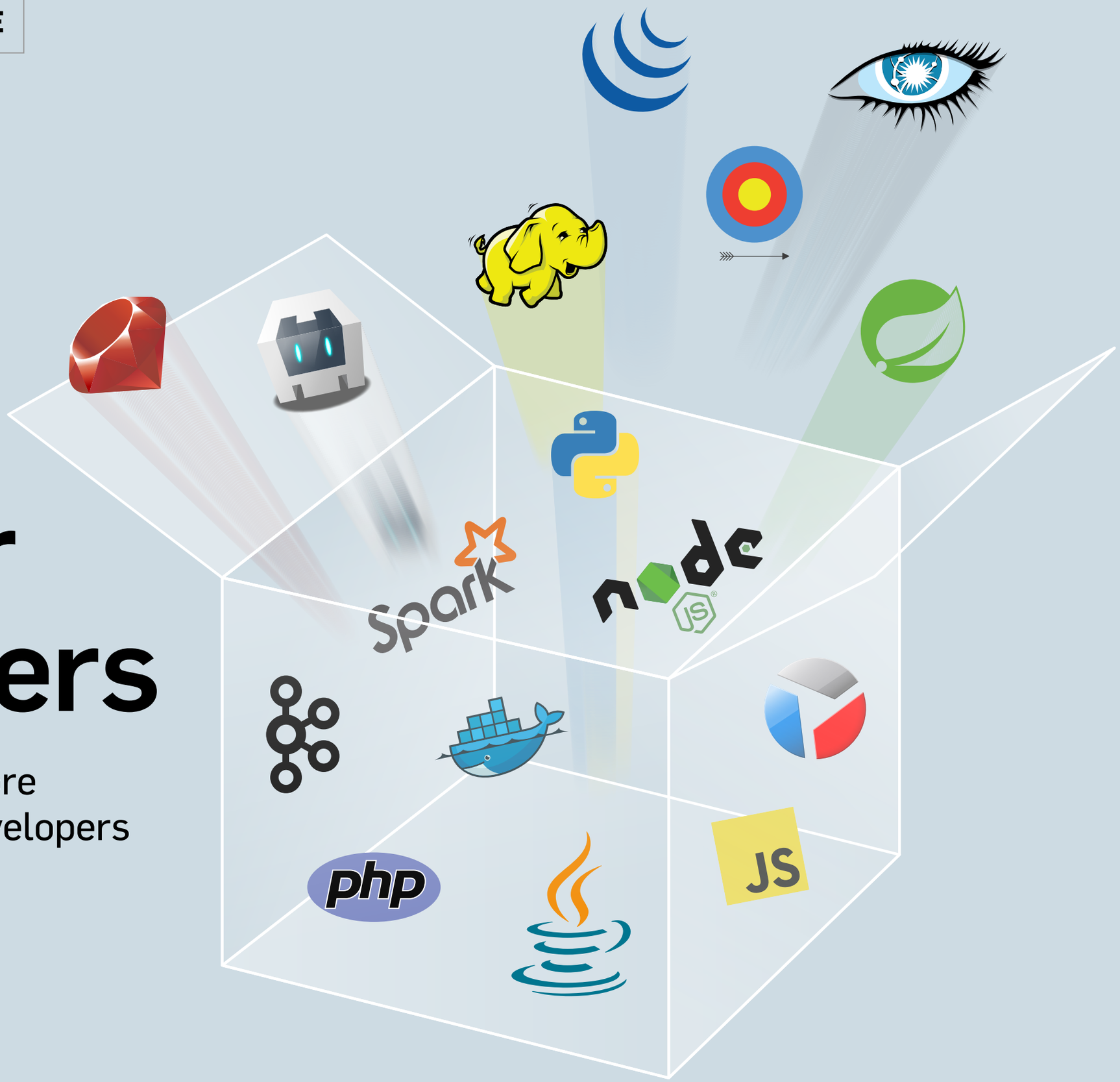
that mobile computing has become a huge driver for organizations to keep investing in technology. It has also reshaped quite significantly the way in which we design software, as more often than not we have to think mobile-first.

What green practices do you use in your development work? I'm very into sustainable software development, and I actually have an award to show for it, which several colleagues and I were proud to win at the OTN Developer Challenge hosted during the AMIS 25 conference in the Netherlands. Our project showed how the Oracle Cloud platform could provide a sustainable solution for pollution control in Mexico City and other big cities.

Open for Developers

There has never been more freedom of choice for developers building on Oracle Cloud.

BY ALEXANDRA WEBER MORALES



Oracle may be known as the source of the most-popular databases (Oracle Database and MySQL) and programming language (Java), but it hasn't always been known as the most developer-friendly tech company—until now. With Oracle Cloud Platform, the company has opened the doors wide to developers looking for an open, modern, and easy platform for building cloud-native as well as conventional enterprise applications. And it's reaching out to those developers with a variety of initiatives, including a new developer portal, a 20-city developer conference, and advocacy programs.

"If you're a developer building new cloud-native applications or modernizing existing applications, Oracle is changing the way you can leverage Oracle and non-Oracle technologies to develop these applications," says Thomas Kurian, Oracle president of product development. "We want you to be able to access and use our cloud services and reach this amazing technology that we spent so many years building."

Open, Modern, Easy

Developers have always been part of Oracle's DNA—witness Oracle Technology Network's ongoing popularity as the go-to destination for authoritative and community-driven content for Oracle developers and other technologists. And Oracle's technology has always been built on open standards. But the new developer portal and upcoming developer-focused events go a step further by helping developers find and share information about—and get hands-on access to—Oracle software innovations that make developing applications more open, modern, and easy.

Open. Oracle Cloud Platform lets developers code not only in Java but also in JavaScript/Node.js, PHP, and soon in Ruby and Python, says Bruno Borges, principal product manager for developer engagement on Oracle Cloud Platform. In addition to the ubiquitous Java Platform, Enterprise Edition (Java EE), Oracle encourages developers to explore other open source frameworks, such as Spring Framework and Dropwizard.

Oracle enables developers to use their choice of open source technology. "You want to use your best open source tools to build applications against the Oracle Cloud Platform, whether that's your DevOps tools, your testing tools, programming frameworks, or open source data

> **“We want you to be able to access and use our cloud services and reach this amazing technology that we spent so many years building.”**
>
> **—Thomas Kurian,** President, Product Development, Oracle

management products, such as Hadoop, Spark, and Kafka,” Kurian says. “We’ve done a lot of work to certify, test, and make those work really well, with great performance in our cloud.”

Developers also have a choice of databases on Oracle Cloud Platform: they can use Oracle Database, MySQL, or NoSQL databases such as Cassandra or MongoDB, Borges says.

Modern. The cloud enables different architectural styles, which themselves depend on different tools. “We enable modern applications built with containers, microservices, and APIs, using tools for DevOps and Agile methodologies,” Borges says. “And you can use open source technologies such as Apache Spark, Kafka, Vagrant, Puppet, and Docker. We also have exciting new tools and technologies coming out soon for leveraging machine learning and artificial intelligence and for building chatbots within in your applications.”

Easy. Oracle Cloud Platform makes it fast and easy to develop cloud-native and conventional enterprise applications leveraging rich platform functionality—including multitenancy, mobile, social, Internet of Things, integration, and operations management. Developers can easily build applications using containers, migrate legacy applications running on VMs to the cloud without needing to change even the network topology of those applications, or simply create applications without writing code but rather by leveraging drag-and-drop application development interfaces. And around these different types of development is a complete DevOps lifecycle for faster time to market and higher developer productivity.

“When developing, you need good debugging to release high-quality code. When your application is into production, you need to proactively identify where problems might be developing,

or if a problem has happened, to quickly identify the root cause of the problem and fix your code," says Siddhartha Agarwal, vice president of product management and strategy for Oracle Cloud Platform and Oracle Fusion Middleware. "Oracle Management Cloud capabilities leverage machine learning and big data across user experience, web/application/database tier performance, and log data to give developers rapid insight into issues without needing to install or manage any software other than dropping in an agent." (See the sidebar "Open for DevOps.")

All of these topics are on the agenda for the Oracle Code 2017 series of developer conferences to be held in 20 cities around the world. (See Developer Roundtable "Personal Development.")

Open to All Kinds of Developers

Increasingly, Oracle sees three main types of developers, says Mike Lehmann, vice president of product management for Oracle Cloud Platform development: the aforementioned cloud-native developers (a fast-growing 25 percent of Oracle Cloud customers), conventional enterprise developers (about 40 percent), and low-code developers (a surprisingly large and growing 35 percent).

Those conventional enterprise developers who already have Java EE applications running on Oracle WebLogic Server and Oracle Database "also want to get to the cloud and modernize for efficiency and agility—and they may want to build new functionality as well," Lehmann says.

The growing ranks of low-code developers—what Gartner calls "citizen developers"—are eager to use cloud services such as Oracle Application Builder Cloud Service, he says. Oracle Application Builder Cloud Service lets business users and citizen developers build and publish applications, or extend software-as-service (SaaS) applications, to keep up with rapidly changing demands. These low-code capabilities also include the ability to quickly build rich mobile applications.

"There are a lot of developers coming up in the lines of business," says Agarwal. "We're actually giving them the ability to do drag-and-drop-based application development of a rich mobile app," he adds. "They don't need to know how to write APIs—the API is exposed and now they're just dragging and dropping and making real-time data connections in their application, and they can test it right there."

> **“What we’ve done in our cloud is made all of its services API-driven and completely scriptable.”**
>
> —**Mike Lehmann,** Vice President of Product Management for Oracle Cloud Platform Development

“Our differentiation in the cloud space is that we provide solutions for all three types of developers,” Lehmann says. “We see a lot of intertwined use cases and intermingling of these three styles.”

Open for Polyglot Programming

Just as there is no single cloud use case, there is also no single language, despite Java’s dominance in enterprise application development.

Oracle Application Container Cloud is Oracle’s polyglot runtime environment, currently supporting Java Platform, Standard Edition (Java SE) as well as any Java framework, including Spring Boot and Dropwizard; Node.js; PHP; and more. “More recently, beyond language runtimes we have added more-advanced capabilities such as in-memory data caching and the ability to develop and deploy formations of services with what we call stacks,” Lehmann says.

“Oracle Application Container Cloud is the place for you to write your polyglot microservices, and then it plugs into the Oracle ecosystem to enable you to extend it into your larger enterprise infrastructure,” he adds, and connecting cloud apps to Oracle services for integration, collaboration, data, identity, and management ties development efforts to real value-creating enterprise solutions.

If you want a DevOps pipeline for producing polyglot applications, Oracle Developer Cloud Service has everything you need. Oracle Developer Cloud Service streamlines software development in any language by providing issue tracking, code versioning, wiki, Agile tools, continuous integration, and delivery automation.

If you want application performance monitoring of that polyglot runtime, Oracle Management Cloud manages that environment. And if you want to register your APIs from those polyglot applications in Oracle API Platform Cloud Service, Oracle provides that integration.

Open for Open Source

Beyond the ubiquitous Oracle Database, which

supports all major programming languages, Oracle plays an ongoing role in the Java and MySQL communities as well. MySQL is the world's most popular open source database for web, cloud, and mobile applications, with more than 100 million copies downloaded and distributed. Java itself is completely developed in open source and the OpenJDK, and Oracle continues to invest in and publish open source tools such as Oracle JavaScript Extension Toolkit (Oracle JET) and EclipseLink.

As Kurian notes, Oracle's developer initiative encourages the use of industry-standard open source tools. Oracle Cloud gives these tools first-class citizenship so that developers can stand up a MySQL database in minutes, build any open source framework on Java SE in Oracle Container Cloud, and launch PHP applications out of the box.

BONUS CONTENT

"If you're a developer, with our cloud you can access any software we have with a browser or an API core, and we've published our APIs into a catalog," Kurian says. "We know developers love open source tools and technologies. [We're integrating] these technologies with services in our cloud so you can use the best tools to automate how you build, deploy, and manage software environments."

However you define *open*—open for the easy integration of an array of popular tools and technologies without vendor lock-in, open for developers ranging from citizen to cloud-native, or open to modern development lifecycles using cloud-based DevOps tools—Oracle has clearly opened all doors to developers and invited them to step inside.

Alexandra Weber Morales, principal at World Wind Writing, is the former editor in chief of Software Development *magazine and has more than 15 years of experience as a technology content strategist and journalist.*

NEXT STEPS

EXPLORE Oracle Developer resources.

TRY Oracle Cloud.

LEARN about Oracle Code 2017.

WATCH Siddhartha Agarwal and Rex Wang discuss Oracle's developer initiative.

OPEN FOR DEVOPS

So you want to move your application to the cloud, or build a new one, or simply be able to react more quickly to changing business conditions. Modern applications are increasingly built using open source lifecycle tools for everything from enabling continuous integration to defining deployment infrastructure as code. Following a DevOps process with tools such as Puppet and Chef is no longer aspirational: It's here, in the cloud.

Oracle Developer Cloud Service integrates with popular IDEs, including Eclipse, Oracle JDeveloper, and NetBeans. IDE users can develop and commit code through Git integration tools, work with tasks and defects through Mylyn integration, monitor builds, and more.

"What we've done in our cloud is made all of its services API-driven and completely scriptable," says Mike Lehmann, vice president of product management for Oracle Cloud Platform development. "So if you need a four-node Oracle Real Application Clusters database with an elastic Node.js Docker cluster, in the past it might have taken dozens of weeks for a shop to acquire the hardware and have operations configure it. Now you can use DevOps tools in Oracle Cloud to completely automate the process, or you can reach in via our cloud service APIs and do it through DevOps tools that you have in-house."

With this ability to instantly stand up and tear down computing environments in the cloud, development teams can launch applications incredibly quickly, Lehmann says. Trends such as distributed microservice architectures deployed via containers are increasingly viable. But having this much choice can also overwhelm developers.

"Building microservices can be difficult, since you can't depend on them running on a particular server, and many chain together to deliver functionality," says

Siddhartha Agarwal, vice president of product management and strategy for Oracle Cloud Platform and Oracle Fusion Middleware. “Leveraging Docker containers is challenging, because you need an orchestration layer and a scheduling paradigm, and most of these technologies are nascent. You need tools for a continuous integration and delivery pipeline and for Agile task tracking. Debugging problems in production is hard, given the distributed nature of microservices. The big challenge is bringing together a solution for all the major cloud-native components.”

Oracle’s answer to this growing complexity is the Oracle Cloud Platform set of AppDev services, which offers and tightly integrates the six capabilities that a cloud-native platform needs: DevOps, microservices, API management, Docker, mobile, and diagnostics. The key services are Oracle Developer Cloud Service (DevOps and collaboration), Oracle Application Container Cloud (polyglot microservices), Oracle API Platform Cloud Service (API management), Oracle Container Cloud Service (Docker), Oracle Mobile Cloud Service (mobile), and Oracle Management Cloud (application performance monitoring and log analytics).

“What is so unique about Oracle’s solution is not only having a first-class cloud-native development platform, but how it plays into the surrounding ecosystem that nearly every cloud-native application development project needs: integration services, collaboration services, data services, identity services, management services, and so on,” Agarwal says. “Beyond getting an integrated cloud-native platform, being able to tie these new cloud-native applications into existing on-premises, legacy, and other cloud solutions is incredibly hard, and Oracle is unique in enabling the full story.” (See Developer Interview “Cloud-Native Delivers.”)

When technology unlocks potential, brilliant ideas come to life.

Smart. Flexible. And prepared for what's next. It's what every company needs to be. Including yours. That's why we bring the newest innovations to address your critical business challenges. We'll work with you to harness the power of emerging technologies, while getting the most out of your legacy IT. By combining our industry insights with technological expertise, we'll help you unleash the true potential of your business. That's high performance, delivered.

Strategy | Consulting | Digital | Technology | Operations

High performance. Delivered.

accenture technology

DEVELOPER | INTERVIEW

Cloud-Native Delivers

Why do cloud-native projects outperform traditional enterprise development?

BY ALEXANDRA WEBER MORALES

The biggest challenge for executives overseeing cloud-native development is cultural, says Siddhartha Agarwal, vice president of product management and strategy for Oracle Cloud Platform and Oracle Fusion Middleware.

Siddhartha Agarwal, vice president of product management and strategy for Oracle Cloud Platform and Oracle Fusion Middleware, talks with *Oracle Magazine* about the cultural shifts, technologies, and common patterns he sees in cloud-native application design and development.

Oracle Magazine: What is cloud-native development?

Agarwal: Cloud-native development refers to a modern application development paradigm that focuses on developing applications for the cloud first or only for cloud. With cloud-native development, applications can be delivered faster and in a more agile fashion than traditional applications.

Oracle Magazine: What are the challenges for executives overseeing cloud-native development projects?

Agarwal: The biggest challenge is cultural. Executives want "big bang" projects but traditionally have seen those projects take months and years to deliver. Cloud-native approaches take that development cycle down to days and weeks. But faster delivery means feedback and guidance from business executives is needed sooner rather than later.

This cultural shift goes up and down the organizational structure. Those building the applications need to understand the timeline to delivery. Those accepting the applications need to realize that what isn't quite right the first time will be easily fixed and extended in the next release. This forces a lot more communication within teams and across teams—all good outcomes—but it can be a jarring organizational change.

Oracle Magazine: What challenges do cloud-native developers and projects face, and how does Oracle address them?

Agarwal: The big challenge for cloud-native developers is bringing together a solution and process that addresses the several major components of cloud-native development. Building microservices, for example, can be difficult because developers cannot depend on them running on a particular server. Leveraging Docker containers is challenging because container orchestration requires an orchestration layer, long-running web services require scheduling, and so on.

It is often easy to find a solution that manages Docker containers for a developer team *or* to find an API management solution *or* to find a mobile development platform. What is hard is

“Oracle’s solution is more than first-class cloud-native tools—it’s an ecosystem that these projects need.”

bringing together the complete solution that enables developers to deliver applications on that platform in a fully automated DevOps-friendly environment.

Our solution is the Oracle Cloud Platform AppDev set of services. We’ve examined the five or six capabilities that a cloud-native platform needs—DevOps, microservices, API management, Docker, mobile, and diagnostics—and built a platform where these are tightly integrated to work together.

Oracle’s solution is more than first-class cloud-native tools—it’s an ecosystem that these projects need: services for integration, collaboration, data, identity, management, and so on.

Oracle Magazine: What are typical use cases for going cloud-native?

Agarwal: We see several patterns over and over again.

Customers are doing new cloud application development, often in their new business areas. Typically these projects start with a mobile front end and a back-end data service. A set of microservices implement the business rules and persistence. The new cloud-native apps connect to source software-as-a-service [SaaS] and on-premises applications via an API and integration solution. This net-new cloud application development pattern is supported by Oracle’s platform-as-a-service [PaaS] solutions, including Oracle Mobile Cloud Service, Oracle Application Container Cloud services, Oracle API Platform Cloud Service, and Oracle Integration Cloud Service. In addition, Oracle’s infrastructure-as-a-service [IaaS] platform provides high-performance compute and container-as-a-service capabilities to run any microservices/container-based applications for DevTest or production.

Then, there are Oracle customers running an enterprise Java application on Oracle WebLogic Server or Oracle SOA Suite as a key part of their on-premises portfolio. They’ll transparently migrate those applications to our Oracle Java Cloud Service, Oracle SOA Cloud Service, and Oracle Database Cloud Service. But inevitably

Oracle sees three use-case patterns for cloud-native development over and over again, says Siddhartha Agarwal, vice president of product management and strategy for Oracle Cloud Platform and Oracle Fusion Middleware.

> **Ultimately, in every application development pattern, the cloud-native experience lets developers solve business problems quickly and gain immediate value.**

they want to modernize those applications with cloud-native capabilities including a mobile front end, a new business microservice, and an API for external access.

This *modernize-and-extend* application development pattern leverages cloud services that enable cloud-native development—Oracle Mobile Cloud Service, Oracle Application Container Cloud services, and Oracle API Platform Cloud Service—alongside cloud services to modernize on-premises Oracle and non-Oracle workloads.

Finally, there is an application development pattern for extending SaaS applications via microservices and mobility to deliver a new business process or experience that spans multiple SaaS/on-premises applications. For example, suppose a business wants to integrate information from multiple SaaS applications to deliver a custom mobile experience—such as a chatbot—to drive prospect excitement and loyalty. Like the other two application development patterns, we see this one over and over again.

Ultimately, in every application development pattern, the cloud-native experience lets developers solve business problems quickly and gain immediate value.

Alexandra Weber Morales, principal at World Wind Writing, is the former editor in chief of Software Development *magazine and has more than 15 years of experience as a technology content strategist and journalist.*

PHOTOGRAPHY BY
BOB ADLER/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Code 2017.

EXPLORE Oracle Developer resources.

PERSONAL DEVELOPMENT

Oracle Code 2017 brings modern cloud development to you.

BY ALEXANDRA WEBER MORALES

There's another developer conference in town this year—in many towns, actually. And this one is dedicated exclusively to helping developers build modern web, mobile, enterprise, and cloud-native applications.

Stephen Chin, director of Oracle Technology Network (OTN); Bruno Borges, principal product manager for developer engagement on Oracle Cloud; and Gerald Venzl, senior principal product manager for Oracle Server Technologies, spoke with *Oracle Magazine* about the motivation behind the Oracle Code 2017 event series and what's in it for attendees.

Oracle Magazine: What is Oracle Code 2017?

Chin: With Oracle Code 2017, we're doing a 20-city conference across the world, with stops in every continent except Antarctica. It's an event for developers, produced by technical folks, and we have exciting content, including Oracle keynotes and luminaries. It will be a showcase of some of the latest and greatest modern software development technologies and trends—including the cloud, of course.

Folks can get hands-on experience and interactive demos, and as with any good conference, we're going to have some of the best technical content available. We're pulling in well-known third-party speakers and developer advocates to talk about topics such as containers, microservices, DevOps, and cloud development and deployment.

Oracle Magazine: Why should developers attend?

Chin: Each event will include at least 20 technical sessions with lots of live coding and live demos in addition to hands-on labs where developers can experience Oracle Cloud. And some cities' events may include free Oracle University hands-on labs that will provide discount vouchers for any Oracle certification exam.

Borges: But most importantly, developers should attend because this is an opportunity to hear from great community speakers and Oracle engineers about what is shaking the software development world these days. There is so much innovation and there are so many new things to learn that developers are sure to come away with excellent tips and guidance on ways to boost their careers and improve their workdays.

Oracle Magazine: How do developers relate to Oracle technology today?

Borges: Any developer in the world is using at least one Oracle technology, whether it's the

> **“This is an opportunity to hear from great community speakers and Oracle engineers about what is shaking the software development world these days.”**
>
> —**Bruno Borges,** Principal Product Manager for Developer Engagement on Oracle Cloud

Java platform or Oracle Database or MySQL Database. So Oracle has a huge developer base. Whether these developers realize they are using Oracle technology or not, that’s a different story.

Venzl: Also, some people think Oracle technology is great but too expensive. Well, the game has changed with the cloud, where you pay for what you use—and it’s very inexpensive.

Oracle Magazine: What technologies does Oracle bring to developers, and how might those technologies be demonstrated at Oracle Code 2017?

Borges: Oracle is an open company for any technology, with modern solutions for both cloud and on premises and technology that is easy for developers to use to build their applications.

Oracle is *open*—not just to Java development but also to Ruby, Python, PHP, JavaScript, and more. And Oracle is open not only to Java EE but also to the Spring Framework, the Dropwizard framework, Node.js, and the technologies that today’s startups are using. In addition to Oracle Cloud platform and infrastructure services, you can use Cassandra, MongoDB, Redis, or any other technologies to host data for developing applications.

Oracle is *modern*, with solutions for containers, microservices, API development, DevOps, and Agile software development. So Oracle Code 2017 sessions will address technologies such as Apache Spark, Kafka, Vagrant (Chat) and Puppet, Docker containers, and microservices. We’ll also cover machine learning, chatbot development, and artificial intelligence.

Our solutions are *easy* to use. Oracle Cloud services are easy to try—for free—and easy to sign up for. You can easily set up and use a SQL database on Oracle Cloud, whether that data-

base is Oracle Database, a MySQL database, or a NoSQL database. At Oracle Code 2017, we will demonstrate these things as well as show developers how they can use IDEs and tools of their choice to build applications for Oracle Cloud and on-premises solutions.

Venzl: With Oracle Code 2017, we are going out to developers and showing them what's in Oracle's developer toolbox. People tend to forget that great systems are built of lots of different components—which require many different toolkits.

Oracle Magazine: Does this mean Oracle is changing the way it talks to developers?

Borges: A perception in the past was that Oracle did not care about developers. We do care. If we didn't, we wouldn't be building tools for developers. Oracle recently launched a developer portal—developer.oracle.com. If we came to the portal game late, that's a fair criticism. But it is also fair to say that we want to build that connection. And that's why we created the portal, because we want to show developers how many great things they can do—and how much they have already been doing—with our technology.

Oracle Code 2017 aims to go out and show developers, "Hey, Oracle enterprise-grade technology is secure. It's open, modern, and easy, and it delivers what enterprises are looking for."

Find out more or register for the event nearest you at developer.oracle.com/code.

Alexandra Weber Morales, principal at World Wind Writing, is the former editor in chief of Software Development *magazine and has more than 15 years of experience as a technology content strategist and journalist.*

NEXT STEPS

LEARN more about Oracle Code 2017.

TRY Oracle Cloud.

EXPLORE Oracle Developer resources.

WATCH Thomas Kurian discuss Oracle's new developer programs.

Diamond Partner Cloud Premier Asia Pacific

Diamond Partner Cloud Premier EMEA

Diamond Partner Cloud Premier North America

NOW, TECHNOLOGY MEANS BUSINESS – WITH INFOSYS AND ORACLE CLOUD

Technology is enabling bold new players to change the rules of the game. To win, you must stay ahead of the pack.

Infosys' innovative Oracle Cloud solutions give you that competitive edge. With our outcome-driven innovation, improve and modernize your existing systems for greater efficiencies and customer delight.

Infosys®

- Trusted by clients as their Strategic Implementation Partner
- Recognized by Oracle with 7 Excellence Awards in 2016
- Multiple solutions listed on Oracle Cloud Marketplace

With Infosys Oracle Cloud solutions, technology is all about enabling business.
To know more, write to us at **Oracle_mktg@infosys.com**

Engineer Assembles Developer Dream Team

Industry veteran relates the fun and challenge of building the world-class Oracle Bare Metal Cloud network.

BY ALEXANDRA WEBER MORALES

Matteo Frigo, Oracle software architect, is technical lead for an expanding team of developers working on what Oracle Executive Chairman and CTO Larry Ellison has called a second-generation cloud offering.

When you're "an engineer who builds stuff," like Matteo Frigo, life's not just about perks and stock options. A lifetime of algorithmic expertise earns you access to the best people—and the hardest problems.

Over his career, Frigo has applied his expertise to a range of problems: developing medical breath analyzers, authoring a popular library for Fourier transforms, and pioneering the Cilk language and runtime system for parallel programming. He first made his name helping invent *cache-oblivious algorithms*, meaning algorithms that run optimally without hardware-specific tuning.

After architecting the Amazon Web Services (AWS) Elastic File System a year and a half ago, Frigo wanted a new challenge. When many of his former AWS colleagues began populating Oracle's cloud team in Seattle, Frigo's curiosity was piqued: "At some point, I decided Amazon wasn't for me. Then Oracle said, 'You can be the Boston branch of Oracle Bare Metal Cloud.'"

Now an Oracle software architect, Frigo is applying his deep knowledge and experience—and fielding a team of Boston-area technologists—to build a data center network that makes the Oracle cloud infrastructure service unique.

Building a virtualized network that is 4 to 10 times faster than today's highest-performing networks is technically demanding—and Frigo's team relishes that class of problem. "We're looking at 10 nanoseconds to process every packet that comes in," he says. "If you take that requirement seriously, you need a lot of engineering to get to that point. It's not something that you can get to incrementally. You have to get it right from the beginning."

Building it right means having the right team, which started with just Frigo in January 2016 and is now up to 15 people, with plans to add another 15 by June 2017. As the technical lead, Frigo defines the project and the architecture, "but don't think of me as someone who just writes design docs. I spend most of my days writing code, and the most difficult parts of the code," he says.

The main challenge? Create what Oracle Executive Chairman and CTO Larry Ellison has called a second-generation cloud offering. Oracle Bare Metal Cloud Services emulate on-premises data centers for full operational control, maximum isolation, and high performance, while still providing all the scalability and efficiency of a cloud. At the core of the

> **“Don’t think of me as someone who just writes design docs. I spend most of my days writing code.”**

offering is a virtual network that is much faster than customers can make themselves, Frigo says. “Giving customers the feeling that they have a network to themselves requires some secret sauce,” he adds.

Part of Oracle’s approach is to let technologists like Frigo apply their experience and perspective to the right problems. Here is some of that perspective.

ON SHIPPING QUICKLY

“Most people think shipping quickly means you build something that may not be complete but provides important features,” Frigo says. “I think it means you think hard to implement a simple solution to the right problem. Many will try to solve the immediate need and not think beyond that.”

For example, “When a given network packet comes in, should you forward it or not?” he says. “The quick implementation is to examine only the IP addresses where the packet comes from and where it is going, but that’s not what we did. We built a general-purpose pattern-matching engine so we can compute any property of the packet that we care about. Implementing the general solution takes roughly the same time as implementing the specific solution, but now I have a system that can cope with the inevitable requirement changes without extra development time on our part. In the end, you ship faster if you focus not on the specific use case but on a more general situation that distills the essence of the problem that you are trying to solve.”

ON STAYING MOTIVATED

“The thing that makes it fun is the interaction with brilliant people on a day-to-day basis,” Frigo says. “Half of them are PhDs from MIT. It’s a fun environment, and we’re constantly getting things done at a much faster pace than any other team I’ve worked with. There are companies that try to force things like project tracking

His development team has “a lot of freedom to do the right thing and build it right,” says Oracle software architect Matteo Frigo.

and other nonsense that is just overhead. We do something we like, and we don’t need any external motivation.”

ON AGILE METHODS

Frigo says Agile methods can be useful, depending on the situation. “If you are doing consulting work and the requirements aren’t clear, it makes sense to do a quick prototype,” he says. “That works for problems where the hard part is trying to figure out what to build. But there are other types of problems where the *thing* is hard to build. If you have to deliver something that looks like an airplane every three

> **The thing that makes it fun is the interaction with brilliant people on a day-to-day basis.**

weeks, Agile won't work. In Oracle Bare Metal Cloud, we have both kinds of work."

WHY WORK AT ORACLE? FREEDOM AND FOCUS

"The first reason is that the company is *that* serious about Oracle Bare Metal Cloud. It's clearly strategic, and there's lots of support from Larry Ellison and the board. Oracle understands that building this right is crucial," Frigo says. "Another reason is that most of the people who are architecting this cloud at Oracle have already built Amazon, Azure, or Google. We've all already made a ton of mistakes and are eager not to repeat them."

As a result of Ellison's focus, "we have a lot of freedom to do the right thing and build it right," he says. "Having a directive from the top that orients the whole company in the direction of the cloud is a major reason to be at Oracle and not somewhere else."

Alexandra Weber Morales, principal at World Wind Writing, is the former editor in chief of Software Development *magazine and has more than 15 years of experience as a technology content strategist and journalist.*

PHOTOGRAPHY BY
BOB ADLER/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Code 2017.

EXPLORE Oracle Developer resources.

By Chris Muir

ORACLE MOBILE CLOUD SERVICE AND ORACLE JAVASCRIPT EXTENSION TOOLKIT

Too Much of a Good Thing?

Use Oracle Mobile Cloud Service custom APIs in Oracle JET.

If you've followed recent *Oracle Magazine* mobile development articles, you've seen articles on mobile client technologies such as Oracle Javascript Extension Toolkit (Oracle JET), which creates web and hybrid mobile apps, and articles on mobile server technologies, particularly Oracle Mobile Cloud Service, which provides web services for mobile and web apps to consume. I'm sure it's no surprise when, as an Oracle employee, I say that each and every developer tool Oracle offers is designed to *simplify* the tasks of developers who need to deliver mobile products to market.

But can there be *too much of a good thing*? Oracle JET provides the Common Model and Collection API to make *consuming* remote REST web services easy for developers. Oracle Mobile Cloud Service, which *publishes* REST web services, also provides client SDKs to make *consuming* REST APIs easier. When combining these technologies, which "easy" option should you use? Can you have the best of both worlds?

In this article, I first explore the concepts for each development platform and then conclude with a working demo you can explore to learn more.

ORACLE JET COMMON MODEL AND COLLECTION API

A cornerstone of sharing data between modern systems is undoubtedly HTTP-based REST web services. This extremely popular approach, where a client such as a mobile app requests a URL across HTTP, using a verb (GET, PUT, POST, PATCH, or DELETE) to retrieve or act upon a resource—possibly a JSON file or image or another resource type—on a server is surprisingly easy for developers to learn and utilize, as this video describes.

To avoid large volumes of boilerplate code for apps that have many remote web services to work with, Oracle JET provides the Common Model and Collection API. This JavaScript API is designed to enable Oracle JET developers to simply write some mapping code to the remote service, after which the API takes care of working with the resource across HTTP.

The following code works with a remote HR web service to retrieve employees:

```
self.EmployeeModelDef = oj.Model.extend({
 url: "http://<myserver>/custom/hr/employees",
 parse: function(response) {
          return { id:response["id"],
                   name:response["name"] }; },
 parseSave: function(record) {
          return { id:record["id"],
                   name:record["name"] }; },
 idAttribute: "id"
```

```
});

self.EmployeesCollectionDef = oj.Collection.extend({
 url: "http://<myserver>/custom/hr/employees"
 model: new self.EmployeeModelDef(),
 comparator: "id"
});
var employees = new self.EmployeesCollectionDef();
```

In the code, I first create an oj.Model object to represent a single object in the remote REST collection. The oj.Model object's URL maps to the remote service, the parsers enable me to transform the incoming and outgoing JSON payloads for my needs, and the idAttribute helps uniquely identify each object.

Next I create a Common Model and Collection API oj.Collection object pointing to the same remote URL. I identify that the oj.Model object I just created is used to represent each individual object in the collection and that the idAttribute can be used to compare different objects in the collection.

With these objects set up, when the Oracle JET UI components act upon the collection at runtime to retrieve the list of employees, the Common Model and Collection API makes the appropriate REST call to the remote hr/employees web service—a GET/POST/PUT/PATCH/DELETE call—depending on what operations I perform on the collection.

ORACLE MOBILE CLOUD SERVICE AND THE CORDOVA SDK

The previous HR service is an overly simplistic REST service for educational purposes. Oracle Mobile Cloud Service supports multiple authentication schemes over

multiple requests, and it uses several custom HTTP headers, which can make the task of calling the service more work for the client-side developer. To make this task easier, Oracle Mobile Cloud Service provides several client SDKs, including an Apache Cordova SDK, which is relevant to Oracle JET hybrid mobile apps.

As I mentioned earlier, the availability of both the Oracle JET Common Model and Collection API and Oracle Mobile Cloud Service SDKs means two competing mechanisms for calling the remote REST web service. Which do I use? Luckily the authors of the Oracle JET Common Model and Collection API considered this, so developers can override the default behavior to plug in such SDKs. For the oj.Model object you saw earlier, I can define my own oj.sync method that can intercept and override the REST GET/POST/PUT/PATCH/DELETE behavior, as shown in the following code:

```
function employeesSync(method, model, options) {
 var url = "hr/employees";
 var verb = "UNKNOWN";
 var payload = null;

 if (method == "read") {
  url = (model instanceof oj.Model) ? url + "/" + model.id : url;
  verb = "GET";
  payload = null;
 } else if (method == "create") {
  verb = "POST";
  payload = model.toJSON();
 } else { /* similar logic for "patch", "update", "delete" */ }
```

```
 // Assumption: mbe is already authenticated against MCS
 mcsconfig.MobileBackend.
  CustomCode.invokeCustomCodeJSONRequest(url, verb, payload,
   function(statusCode, data, headers) {
     options["success"](data, null, options); },
   function(statusCode, data) {
     options["error"](data, null, options);} );
 return {};
}
```

I then plug this employeesSync function into my previous oj.Model definition as follows:

```
self.EmployeeModelDef = oj.Model.extend({
   // include previous attributes
  sync: employeesSync
});
```

Now when the collection object calls Oracle Mobile Cloud Service rather than handling the GET/POST/PUT/PATCH/DELETE calls itself, it will delegate the responsibility to the employeesSync method, which has the ability to change the logic and call the Oracle Mobile Cloud Service SDK CustomCode .invokeCustomCodeJSONRequest method to take care of the call to the Oracle Mobile Cloud Service REST API.

DEPLOYING A RUNNING DEMO APPLICATION

Let's conclude this article with a live demo app so you can look at the real Oracle JET source code and see how it interacts with Oracle Mobile Cloud Service.

To access Oracle Mobile Cloud Service, you need a trial account. You can obtain one by clicking the **Try It** button on the Oracle Cloud Mobile page and filling out the required trial request details. After your trial request has been submitted and approved, watch the following video on how to set up and provision your Oracle Mobile Cloud Service instance and follow the instructions.

With the service set up and provisioning complete, set up the Oracle Mobile Cloud Service artifacts:

1. Download this article's demo zip file, which contains the prebuilt Oracle Mobile Cloud Service REST API and other artifacts for this article.
2. Log in to Oracle Mobile Cloud Service as demonstrated in the video.
3. Select **Applications** -> **Packages** from the hamburger menu, and click the **New Import** button and the **Choose a package file** link. In the resulting open-file dialog box, select the zip file you just downloaded for this article and click **Open**. Complete the steps in the wizard by clicking **Next** and finally **Finish**.
4. From the hamburger menu, under **Applications**, select **Mobile Backends**, **OraMagDemoMBE**, and then **Open**. The mobile back end represents the package of existing code you just imported.
5. Click the **Settings** link, and on the resulting page, copy the values for **Mobile Backend Name**, **Base URL**, **mobile backend ID**, and **anonymous key** to your favorite text editor. To see **anonymous key**, click the **Show** link.
6. Click the **Clients** link and **New Client** button.
7. In the resulting dialog box, enter `Demo` for both **Client Display Name** and **Client**

Name, select **Web** as the platform, enter `1.0` as the version, and click the **Create** button.

8. Once the client is created, copy the **Application Key** value to your text editor.

Later in this article, you will test the app by running it in the Android Emulator on your desktop or via a web browser. For this to work, you must enable cross-origin resource sharing (CORS) for the Oracle Mobile Cloud Service server:

9. From the hamburger menu, select **Administrator**.
10. In the **Policies** section, click the **Export** button. This will download a **policies.properties** file to your desktop.
11. Open this file in your favorite text editor.
12. Locate the line `*.*.Security_AllowOrigin=disallow`. Replace the line with the following:

```
*.*.Security_AllowOrigin=http://10.0.2.2:8000, http://localhost:8000
```

You're now ready to work with the Oracle JET demo app. If you don't already have Oracle JET installed on your development PC, first read the previous *Oracle Magazine* article and watch the install instructions video to set up your PC to develop Android applications via Oracle JET.

With the artifacts set up and registered, set up the demo Oracle JET app:

13. In your file system, create a new directory and then, via the command line in the new empty directory, type

```
yo oraclejet:hybrid OraMagDemo --template=navbar --platform=android
cd OraMagDemo
```

14. Download this application source zip file, and unzip it. Copy the unzipped "src" directory over the identically named "src" directory in the OraMagDemo directory you just created with Yeoman. This Oracle JET app is the same app as the one in the previous *Oracle Magazine* article and is built on the Oracle JET "navbar" template, which includes several modules, such as an empty dashboard and an incidents tab. The "src" code you are adding substitutes the logic for a new employees module that replaces the generated blank customers tab.
15. With your favorite text editor, open the **/src/js/mcsconfig.js** file. Within this file, locate the "config" variable, substitute in the five values you recorded earlier for your specific Oracle Mobile Cloud Service instance, and save the file. Copy the values exactly; make sure not to include any accidental spaces.
16. If you have set up the Oracle JET tooling, including the Android SDK, type

```
grunt serve
```

You can also run the app in your browser by typing the following, which is faster than starting the Android Emulator, so it is often the preferred option:

```
grunt serve --browser
```

17. Once the app is running, select the **Employees** tab to see the employees data returned from Oracle Mobile Cloud Service.

EXPLORING THE ORACLE JET CODE

The fact that the app is running and returning data confirms that Oracle JET and Oracle Mobile Cloud Service can be connected, but how was this achieved? For the

purposes of this article, there are four key files worth exploring to help build your understanding of how Oracle JET and Oracle Mobile Cloud Service work together. Each file includes comments for further reading:

1. **src/js/mcsconfig.js.** This file includes the configuration settings for your Oracle Mobile Cloud Service instance as well as the settings for instantiating the McsConfig object used by the Oracle JET application to access the Oracle Mobile Cloud Service SDK objects. For demo purposes, there is hardcoded logic at the end of this file for logging in anonymously to Oracle Mobile Cloud Service. In a real application, you would provide a login screen to enable your mobile users to do so.
2. **src/js/mcs.js.** This is the Oracle Mobile Cloud Service Cordova SDK bundled with the app, and it is accessed via the McsConfig object at runtime.
3. **src/js/views/employees.html.** This is the HTML page that displays the employees list view. It includes a <ul id=”listview”>, which retrieves a collection of employee rows from the datasource and then, for every row retrieved via the Knockout <script> containing the <li> tag, displays the individual attributes of the employee, such as that person’s first name and last name.
4. **src/js/views/employees.js.** Here you will see the oj.Model and oj.Collection objects and the oj.sync function employeesSync described in this article.

If you’re running the app via your browser, you also have the option of discovering what is happening at the network layer between the Oracle JET app and Oracle Mobile Cloud Service. Each modern browser has a “development tools” option that enables you to inspect not just the source code of the client application but also what is happening at the network layer. With the network options open, switch back to the dashboard in the running app in your browser and then to the Employees option again. You should see, among other traffic, the GET call to Oracle

Mobile Cloud Service, and then you can dig into the actual request and response objects to watch what is passed between your Oracle JET app and the Oracle Mobile Cloud Service SDK to the service itself.

SUMMARY

By design, Oracle JET's Common Model and Collection API attempts to make working with REST web services simple. By design, the Oracle Mobile Cloud Service SDKs also make working with REST web services simple. Luckily—or, more correctly, by design—both solutions can coexist to give you, the developer, the best of both worlds. You can *never have too much of a good thing*.

Chris Muir is a senior principal product manager for mobility, cloud, and development tools at Oracle.

PHOTOGRAPHY BY
SABINE ALBERS/THE VERBATIM AGENCY

NEXT STEPS

READ more about Oracle JET.

READ more about Oracle Mobile Cloud Service.

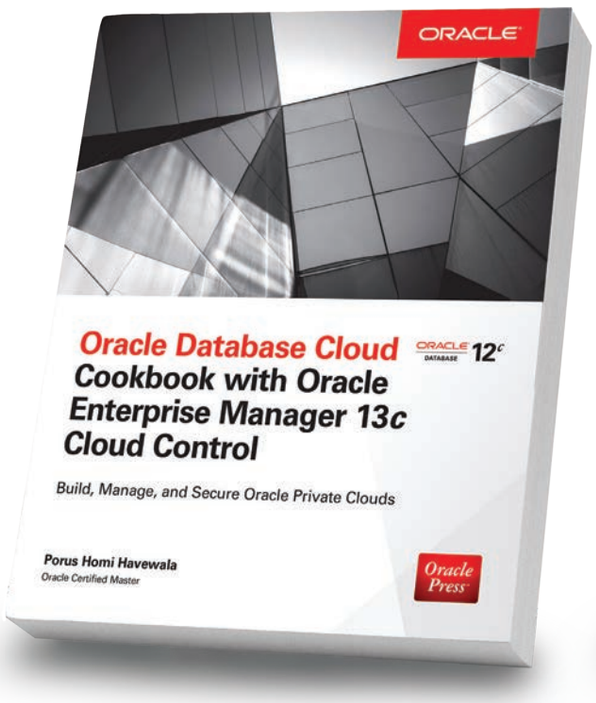
FOLLOW Oracle JET on Twitter.

FOLLOW Oracle Mobile Cloud Service on Twitter.

Oracle Press™

Your Destination for Oracle and Java Expertise

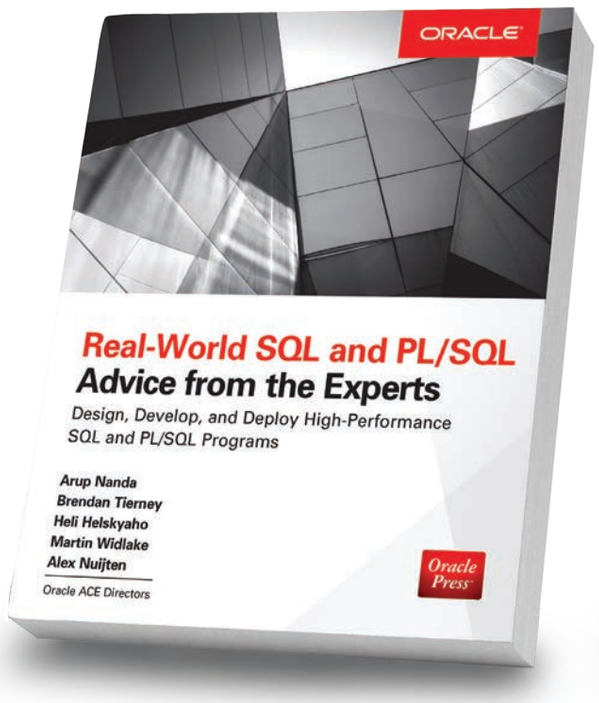
Written by leading technology professionals, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Oracle products and technologies available.



Oracle Database Cloud Cookbook with Oracle Enterprise Manager 13c Cloud Control

Porus Homi Havewala

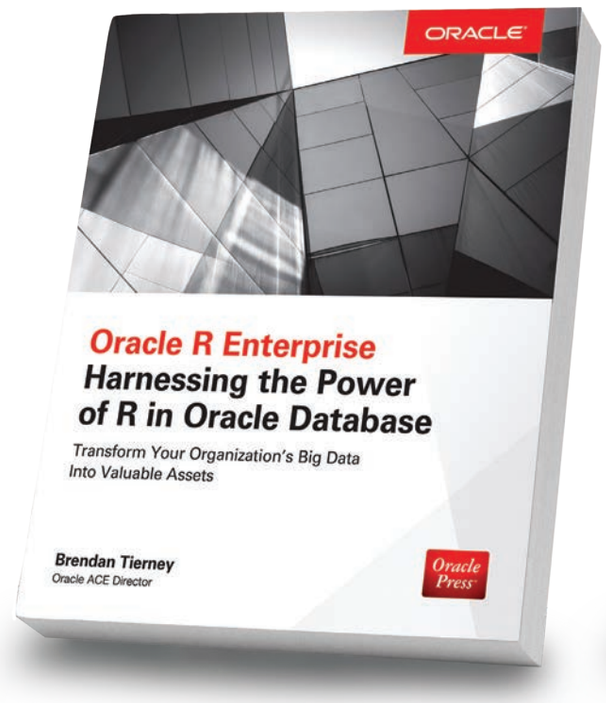
Learn how to manage a secure, private database cloud.



Real World SQL and PL/SQL: Advice from the Experts

Arup Nanda, Brendan Tierney, Heli Helskyaho, Martin Widlake, Alex Nuitjen

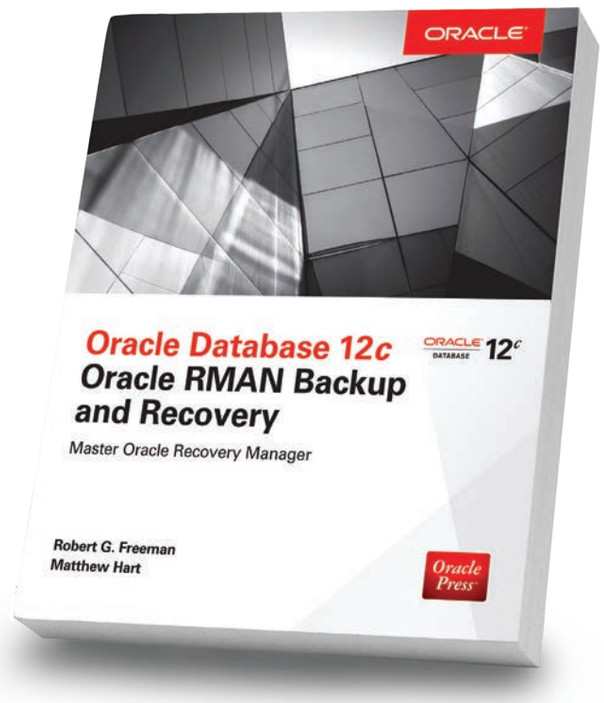
Master the underutilized advanced features of SQL and PL/SQL.



Oracle R Enterprise: Harnessing the Power of R in Oracle Database

Brendan Tierney

Effectively manage your enterprise's big data and keep complex processes running smoothly.



Oracle Database 12c Oracle RMAN Backup & Recovery

Robert G. Freeman, Matthew Hart

Fully leverage every powerful RMAN tool.

Available in print and eBook formats.

Oracle Press™

www.OraclePressBooks.com • @OraclePress

By Mark Rittman

ORACLE DATA VISUALIZATION

Tell the Story

Use Oracle Data Visualization Desktop to present information and insights.

If, like me, you still remember some of the stories you read when you were a child, it's probably the twists and turns, the cliff-hangers, the unanswered questions, and the final resolution that make those stories so memorable. As adults, we still enjoy stories, and good ones resonate both intellectually and emotionally with an audience. Stories aren't limited to fiction, of course, and using Oracle Data Visualization Desktop and some data of interest, you can visualize that data, uncover insights, and then use the tool's storytelling features to make that data more meaningful when you present it to others.

In this article, I'll use Oracle Data Visualization Desktop and its visualization and storytelling features to explore, understand, and then tell the story about my attempt to lose some weight by exercising more. I recorded my workouts and weight each day, using wearable devices and a smartphone app. You can download and unzip the sample datafiles used in this article and download Oracle Data

Visualization Desktop if you want to try the article example yourself. Or you can use your own fitness tracker or workout app data to build your own story by exporting the data to Microsoft Excel XLSX files and using Oracle Data Visualization Desktop.

UPLOADING, ANALYZING, AND DISCOVERING

Let's start by taking a spreadsheet datafile exported from a cycling workout smartphone app. It contains a record of every cycling workout I recorded over a nine-month period starting in January 2016. I'll upload the file into Oracle Data Visualization Desktop; review it; and, if necessary, change how columns within it are designated as either measures or attributes and configure the outside temperature measure to aggregate by averaging its values (rather than adding them all together).

1. Start by double-clicking the **Oracle Data Visualization Desktop 12*c*** icon on the Windows desktop to open the application, and then click the **Navigator** menu button at the top left of the application window to display a menu of options. Click the **Data Sources** menu item to open the Data Sources page, and then click **Create New Data Source**.
2. Because the data I am going to upload is contained in a spreadsheet file, click the **From a File** option in the Create New Data Source dialog box and then use the Windows Explorer file picker to locate and select the **activities.xlsx** file you downloaded and unzipped for this article, clicking **Open** to move on to the next step.
3. In the Upload a File dialog box, use the scroll bar to review the columns of data retrieved from the spreadsheet file. Oracle Data Visualization Desktop automatically categorizes any column containing alphanumeric data as containing attributes and columns with only numeric values as measures. These defaults are correct for this data set except for two columns. First, the **Activity ID** column contains numeric values but is, in fact, an attribute column used to join activities

to other data the smartphone app records. To correct this first column setting, click the **Measure** setting below the **Activity ID** column label and change that setting to **Attribute**, as shown in **Figure 1**. To correct the second column, change the aggregation type for the **Temp C** column from **Sum** to **Avg**.

4. Click **Add Data Source** to add this file and its column settings to Oracle Data Visualization Desktop's set of datasources available for projects.

ANALYZING THE DATA, RECORDING THE INSIGHTS

Now I can create my first data visualization that will show how the distance logged for the workouts I recorded with the smartphone app varied over the nine months of data. Because the workout app also recorded the outside temperature when the workout took place, I'll then overlay that information onto the visualization to see whether how hot or cold it was outside affected the distance logged for a workout.

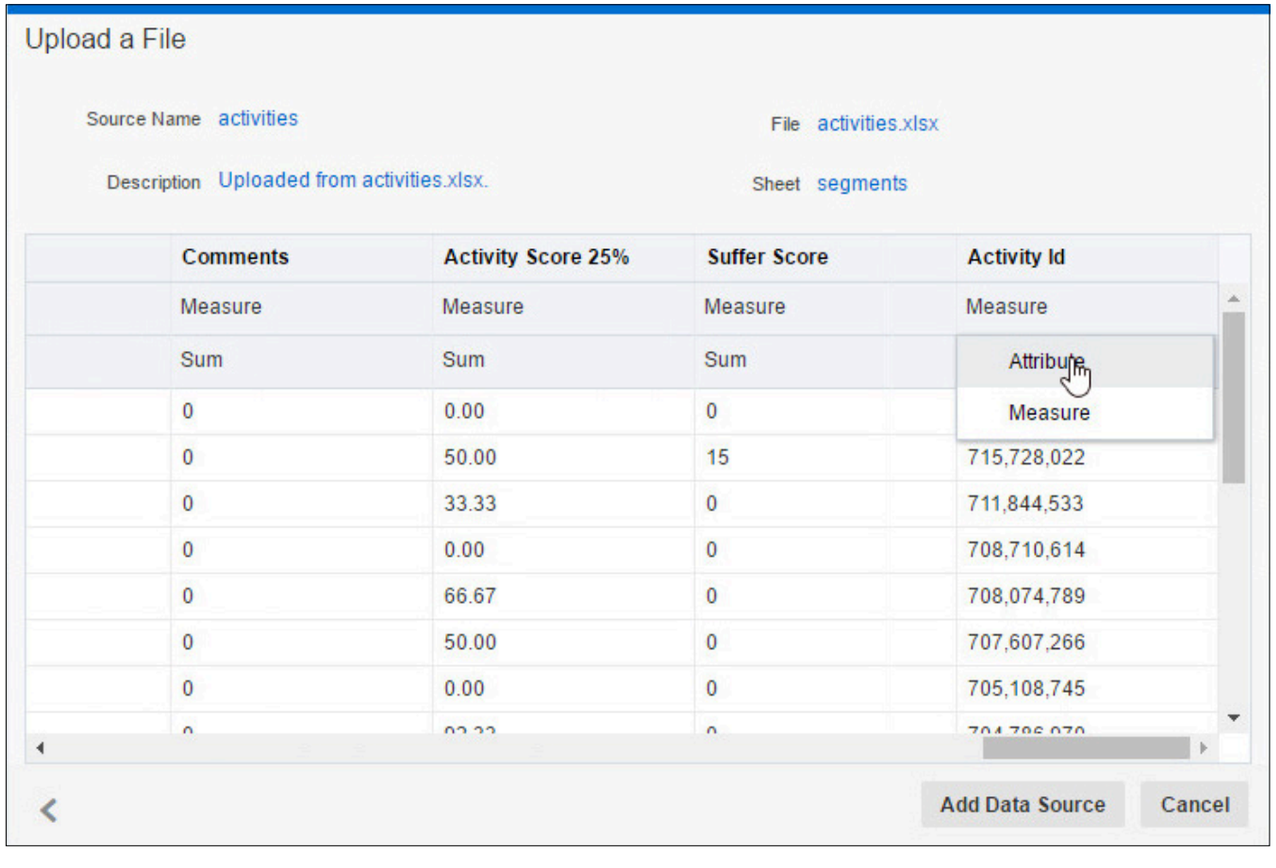


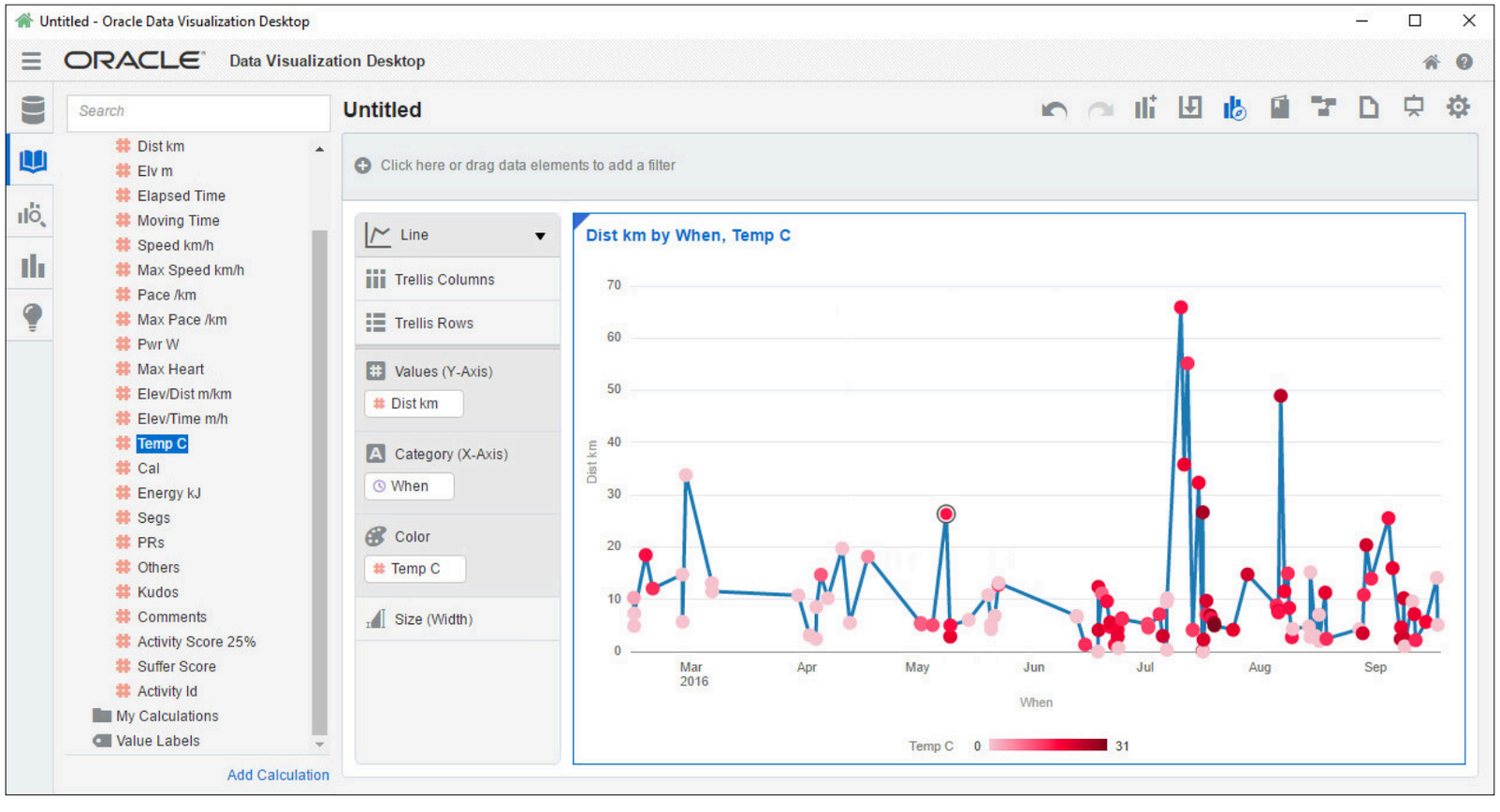
Figure 1: Changing the column type of an imported data set column

5. Click the **Data Elements** menu icon on the left-hand side of the page, and then click the **activities** datasource to display the list of columns that can be added to the canvas.
6. To display a line chart of the distance cycled over time, first double-click the **Dist km** column to add it to the canvas on the right and then double-click the **When** column. Oracle Data Visualization Desktop will automatically recognize the latter column as time-related and place it on the x-axis with

"Month" axis markers while placing the other column on the y-axis.

7. Next, double-click the **Temp C** column on the left-hand side of the page to also add it to the data visualization. In this case, the column is intelligently overlaid as colored dots over the main chart data so that each month's distance total now has an indication of how hot or cold that month was on average. To change the default color to shades of red, better indicating the heat the app recorded for that month, select **Color** -> **Series (Temp C)** and then click one of the red colors so that your final data visualization looks similar to **Figure 2.**
8. Looking at the data visualization I've created, it does seem that more workout activity happened when the weather was warmer, which was, in fact, the case. Let's save that as an "insight," a snapshot of a particular data visualization I can combine with other insights at the end to tell the story behind the data I'm analyzing. To do that, click the insights (lightbulb) icon in the left-hand menu panel, click **Add Insight,** and name the insight `Workout frequency increases as weather improves`.
9. Although the frequency of the workouts seems to have increased, they may well have been for shorter distances, perhaps because it

Figure 2: Data visualization featuring distance, frequency, and temperature



was too hot to cycle or I just exercised less as the year went on. To see this overall trend overlaid on the chart as a trend line calculated with linear regression, click the menu button at the top right side of the chart and select **Add Trend Line**. As I suspected and as shown in **Figure 3**, although some workouts were for very long distances, the overall trend in individual workout distances logged was, in fact, down in the course of the year.

10. Click the insights icon in the left-hand menu panel, click **Add Insight** to add this new insight to your project, and name it `However, the overall trend was down`.

In reality, even though the trend in the workout distance may have fallen, the overall distance logged per month actually rose, due to these additional but shorter workouts. To show this, I first need to create a new calculated column for the month each workout was logged and then use that new column to show the distance traveled per month.

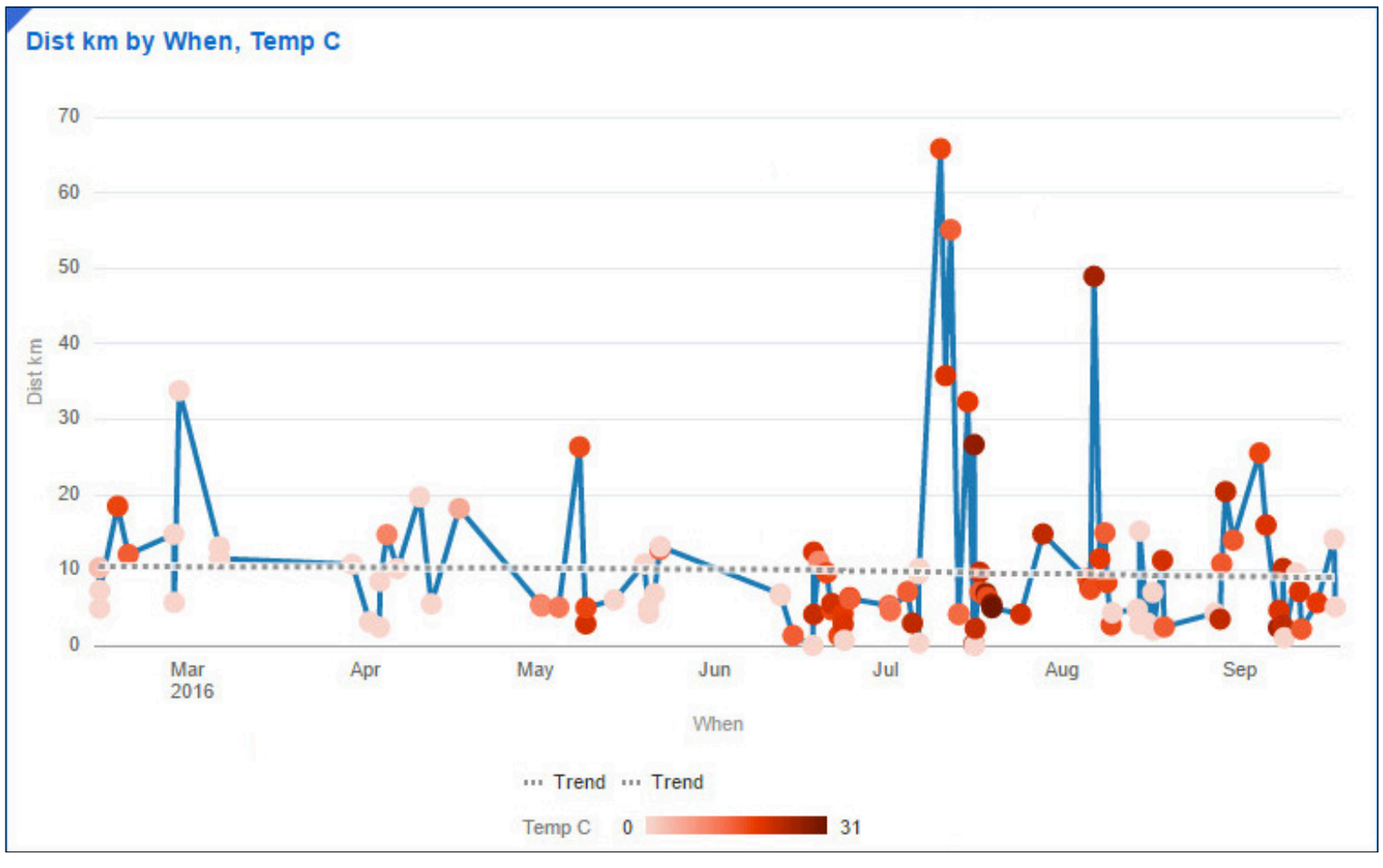


Figure 3: Visualization featuring distance, frequency, and temperature plus distance trend

11. Navigate to the bottom left of the page, and click **Add Calculation**.
12. In the New Calculation dialog box, name the calculation `Workout Month`, select **Calendar/Date** -> **Month** from the list of functions to the right, and double-click it to add that function template to the left-hand side of the dialog box. Click the

dimension expression, select **Activities** -> **When** from the list of columns, click **Validate,** and click **Save** to create the new calculation.

13. Now drag and drop that new **Workout Month** calculation from the My Calculations folder to replace the **When** column within the Category (X-Axis) visualization setting and change the visualization type to **Bar** so that it looks like **Figure 4.** As you can now see, looking at the distance logged overall each month, the trend is clearly up, with the warmest months recording the greatest overall workout distances. Click the insights icon, click **Add Insight** to save the current visualization as another insight for the project, and name the insight `Overall though, Total workout distance rose`.

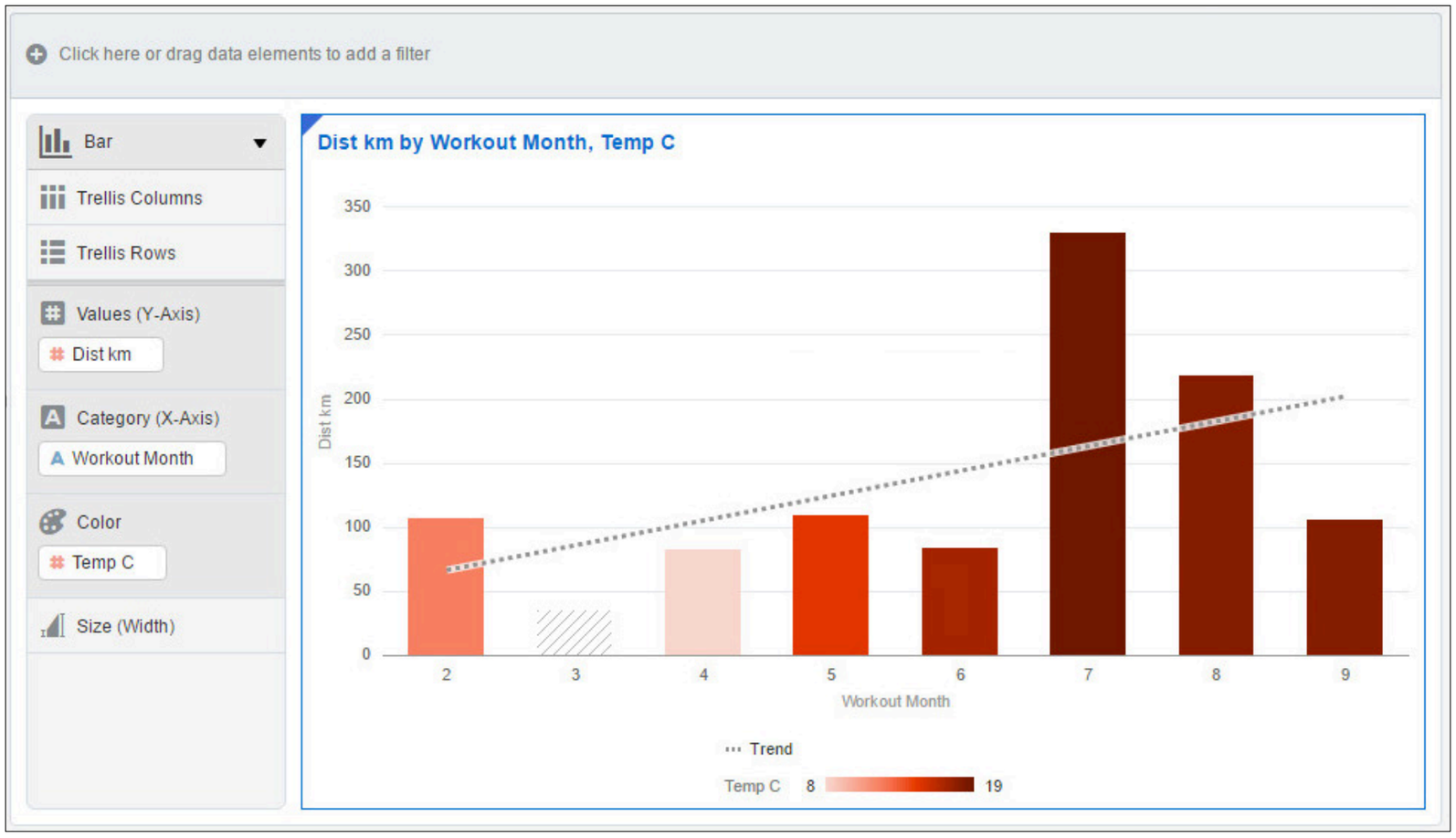


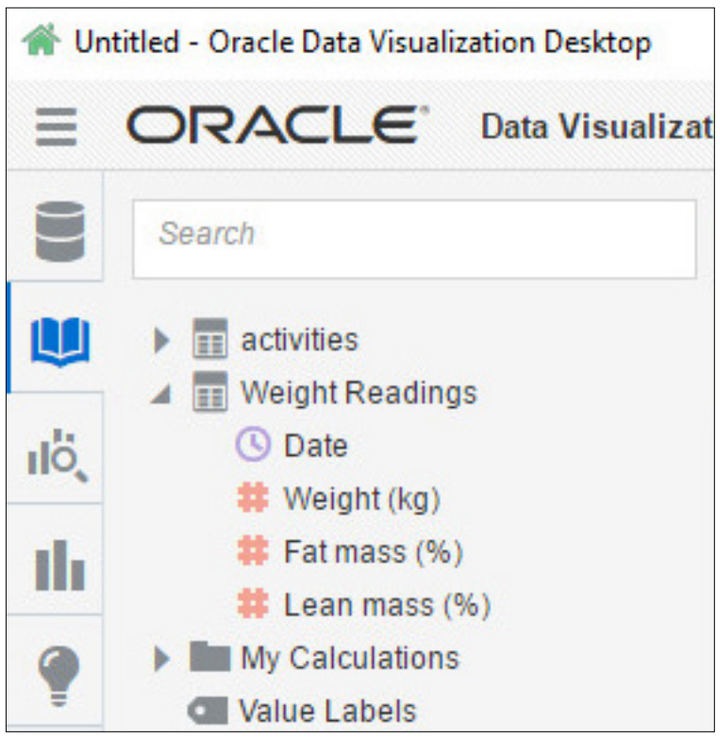
Figure 4: Bar visualization showing the monthly distance trend

ADDING A SECOND DATA SET TO COMPLETE THE PICTURE

So it looks as if my overall workout distance has increased over time, but has this extra exercise resulted in weight loss being recorded in my other smartphone health app, or was all of that working out a wasted effort on my part? To find out, I'll now add the second data-source containing weight readings over the same period and see whether this increased exercise activity has helped me shed those extra holiday pounds.

14. To add this new datasource, click the **Data Sources** icon in the left-hand menu panel, right-click anywhere in the empty space under the existing **activities** datasource, and select **Add Data Source** from the menu.
15. In the Add Data Source dialog box, click **Create New Data Source** and then click **From a File** to display a file picker so you can select and upload the second datafile, **Weight Readings.xlsx**, that you downloaded and unzipped earlier.
16. When the Upload a File dialog box appears, change the **Aggregation** setting for the **Weight (kg)** measure from **Sum** to **Average** under that column's settings, much as how you changed some of the column types after importing the first file (**activities.xlsx**) earlier in this example.
17. Now, to join this new datasource to the existing **activities** datasource, right-click anywhere underneath the two datasources and select **Source Diagram**. Then in the Connect Sources dialog box, click **Add Another Match**, select **Date** from the **Weight Readings** datasource and **When** from the **activities** datasource, click **OK** to save the connection, and click the close (**x**) button to dismiss the source diagram from the main application window.
18. Click the data elements (book) icon in the left-hand menu panel to show this second datasource listed under the one I've been using so far. Double-click it to show the three measures that are now available to add to the data visualization, as shown in **Figure 5**.
19. Now remove the existing trend line from the visualization by clicking the menu icon at the top right of the visualization canvas, selecting **Properties** -> **Analytics**, and clicking the **X** icon to delete it from this visualization.
20. Now click **Add Calculation** on the left-hand side of the page and, in the New Calculation dialog box, name the calculation `Weight Month`. Select **Calendar/Date** -> **Month** from the list of functions to the right, dou-

Figure 5: Three Weight Readings measures available for data visualization



ble-click it to add that function template to the left-hand side of the dialog box, and then click the **dimension** expression to select **Weight Readings** -> **Date**. Click **Validate** and then **Save**.

21. Click and drag the **Weight (kg)** column so that it is placed directly under the **Dist km** column in the Values (Y-Axis) visualization setting, and, from the visualization menu, select **Properties** -> **Axis** -> **Secondary Axis Values**, using 84 and 88 as the **Start** and **End** settings. To enable the secondary y-axis for the weight series, click the **Values** tab in that same dialog box and then select **Weight kg** -> **Y2 Axis** -> **On**. As you can see in **Figure 6**, the weight at the end of the period is lower than at the start and the additional workout distance logged over the summer seems to have led to a corresponding fall in weight shortly thereafter.
Save this as another insight by clicking the insights icon on the left-hand menu panel, clicking **Add Insight**, and naming this insight `And Exercise Seems to Have Worked!`

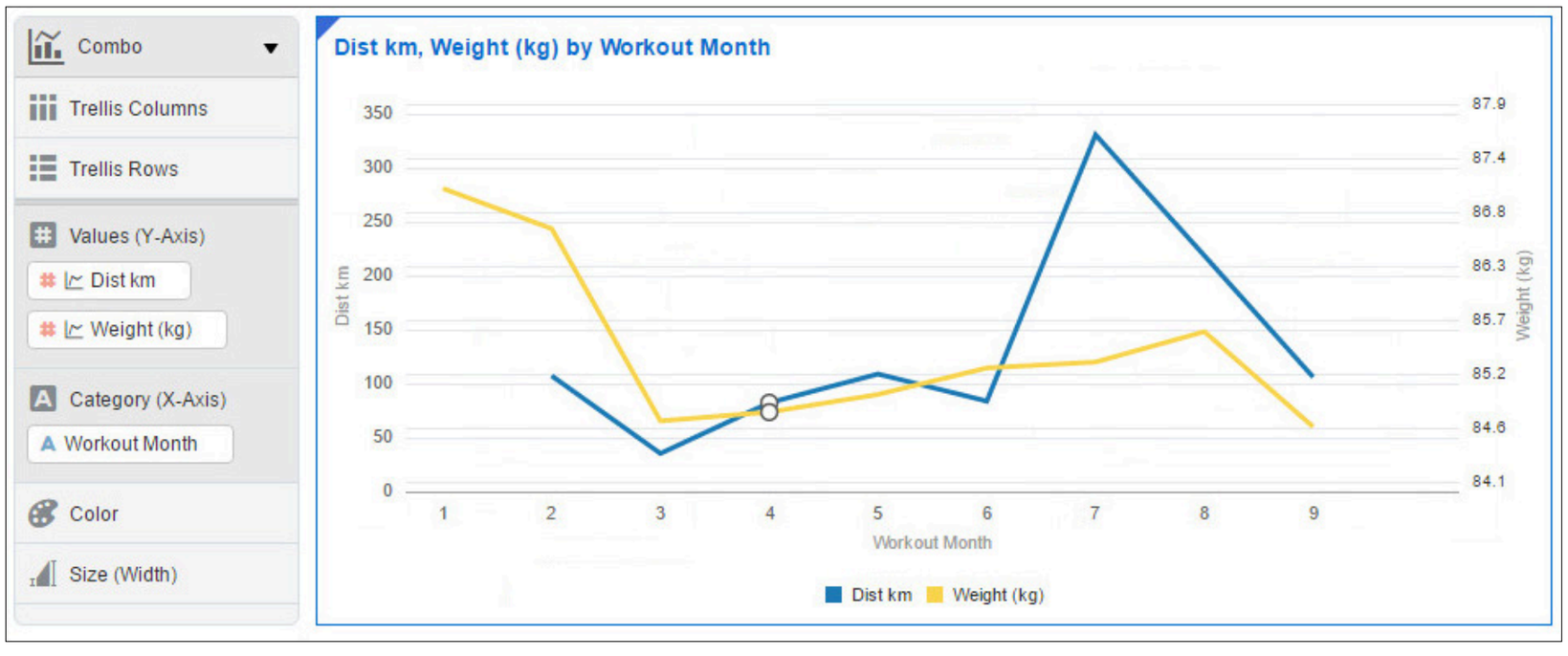


Figure 6: Data visualization showing workout distance and weight

MAKING THE DATA MEANINGFUL BY PRESENTING IT AS A STORY

This example created and explored visualizations using the data from two datasources. I saved data of interest as insights so I could come back to them later. Insights are snapshots of data visualizations at a point

in time and can be refreshed to show updated data if necessary. Insights can also be used to tell a story about the data that can be more effective than just a set of saved reports in a catalog.

To view and optionally edit the sequence of insights that make up the story for this data set, click the **Story Navigator** button in the application toolbar to show the insights in the order you created them. Use the back and forward buttons to view the story timeline, and edit or remove insights from the story timeline to create the final story presentation, as shown in **Figure 7.**

Finally, if you want to present these insights and the story they tell about the data to friends or colleagues, you can switch the application to presentation mode by clicking the **Presentation Mode** button in the same application toolbar to remove all the menus and buttons and focus just on the story and the data.

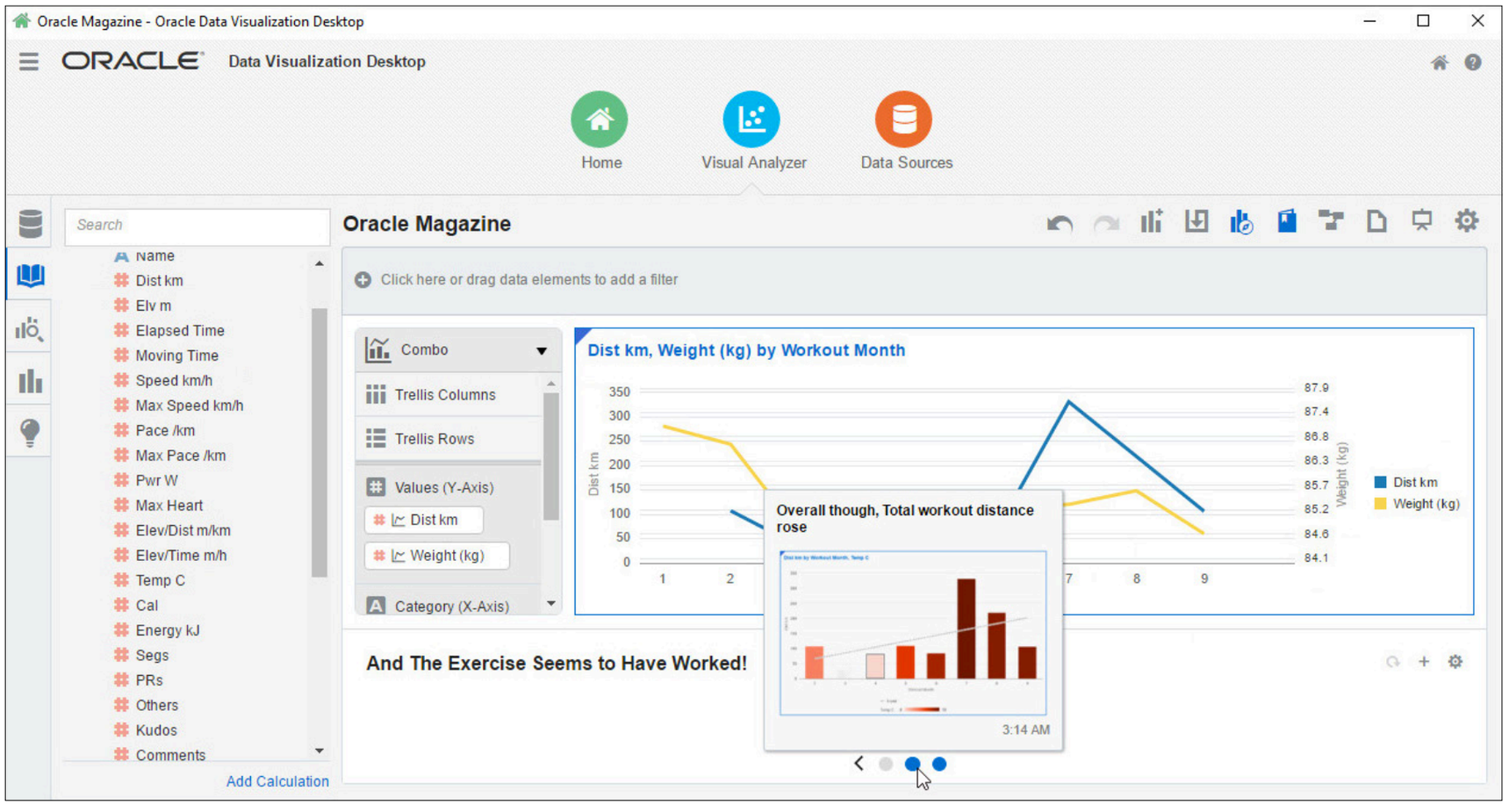


Figure 7: Data visualization with story navigator

SUMMARY

Data, and the meaning behind it, resonates better with audiences and is remembered longer when you use storytelling techniques and effective visualiza-

tions to present your insights to others. Oracle Data Visualization Desktop puts these capabilities into the hands of anyone with a recent desktop PC and some data of interest. Try this storytelling approach out yourself, and see what stories you can tell with your data.

Mark Rittman is an Oracle ACE Director and independent analyst who has been working with Oracle Business Intelligence and big data technology for more than 15 years.

PHOTOGRAPHY BY **JOHN BLYTHE**

NEXT STEPS

DOWNLOAD Oracle Data Visualization Desktop.

DOWNLOAD sample data for this article.

LEARN more about Oracle Data Visualization.

By Steven Feuerstein

ORACLE DATABASE 12*c* RELEASE 2

Powerful Impact Analysis

Easily identify program units affected by database object changes with PL/Scope in Oracle Database 12*c* Release 2.

PL/Scope is a powerful code analysis tool for PL/SQL program units, first introduced in Oracle Database 11*g*. Now with Oracle Database 12*c* Release 2 (Oracle Database 12.2), that analysis includes the SQL statements inside your PL/SQL code base. This significant enhancement enables developers to perform comprehensive impact analysis of changes to tables, such as changing a column's name.

TURNING ON PL/SCOPE

Prior to Oracle Database 12.2, you would start PL/Scope by executing this statement in your session:

```
ALTER SESSION SET plscope_settings='identifiers:all'
/
```

After that, whenever you compiled a PL/SQL program unit, the compile would gather information about all PL/SQL identifier usages and make that data available via the ALL_IDENTIFIERS view.

In Oracle Database 12*c* Release 2, to start PL/Scope *and* gather information about SQL statements, you execute this statement:

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

Then, after compilation of a program unit, you will find information about SQL statements in the ALL_STATEMENTS view, and the ALL_IDENTIFIERS view contains information about *SQL identifiers*, such as table and column names.

When you compile a program unit with PL/Scope enabled for statement gathering, the PL/SQL compiler places a row in ALL_STATEMENTS for each SQL statement in the program unit. The following information is recorded for those statements:

- **Owner.** This is the owner of the program unit.
- **Object name.** This is the name of the program unit.
- **Statement type.** This is the type of statement: SELECT, INSERT, UPDATE, DELETE, MERGE, EXECUTE IMMEDIATE, or OPEN.
- **Line number.** This is the line number in the program unit where the statement begins.
- **Column number.** This is the column position on that line.
- **Usage ID.** This is an identifier unique to this statement in this program unit.
- **Usage context ID.** Roughly speaking, this is the parent of the usage ID. It provides the context in which the usage ID is found in the program unit.

- **Signature.** This is the identifier for the statement. It is unique across all program units.

THE DATABASE OBJECTS FOR ANALYSIS

Before I go further into my exploration of PL/Scope, I need to create some database objects. For this article, I am going to keep track of endangered species, their location, and where research is done on those species.

So I create three tables, as shown in **Listing 1**.

Code Listing 1: Creating three tables

```
CREATE TABLE species_locations
(
   id          NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
   location    VARCHAR2 (100) UNIQUE
)
/

CREATE TABLE endangered_species
(
   id            NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
   name          VARCHAR2 (100) UNIQUE,
   location_id   NUMBER    REFERENCES species_locations (id)
)
/

CREATE TABLE research_locations
```

```
(
   id             NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
   name           VARCHAR2 (100) UNIQUE,
   location_id    NUMBER    REFERENCES species_locations (id)
)
/
```

In the table creation, note that GENERATED AS IDENTITY is an Oracle Database 12*c* Release 1 feature.

After enabling PL/Scope, I then create a package, species_mgr, to help manage data in these tables, as shown in **Listing 2**.

Code Listing 2: Creating the species_mgr package

```
CREATE OR REPLACE PACKAGE species_mgr
   AUTHID DEFINER
IS
   FUNCTION location_id_from_name (
      location_in   IN species_locations.location%TYPE)
      RETURN species_locations.id%TYPE;
END;
/

CREATE OR REPLACE PACKAGE BODY species_mgr
IS
   FUNCTION location_id_from_name (
      location_in   IN species_locations.location%TYPE)
```

```
      RETURN species_locations.id%TYPE
   IS
      l_return   species_locations.id%TYPE;
   BEGIN
      SELECT sl.id
        INTO l_return
        FROM species_locations sl
       WHERE sl.location = location_id_from_name.location_in;

      RETURN l_return;
   END;
END;
/
```

Next, I populate my tables, as shown in **Listing 3**.

Code Listing 3: Populating the three tables

```
BEGIN
   /* Amazon Data */

   INSERT INTO species_locations (location)
        VALUES ('Amazon');

   INSERT INTO endangered_species (name, location_id)
          VALUES (
                    'Black Spider Monkey',
```

```
                    species_mgr.location_id_from_name ('Amazon'));

INSERT INTO endangered_species (name, location_id)
     VALUES ('Sloth', species_mgr.location_id_from_name ('Amazon'));

INSERT INTO endangered_species (name, location_id)
        VALUES (
                    'Amazon River Dolphin',
                    species_mgr.location_id_from_name ('Amazon'));

INSERT INTO endangered_species (name, location_id)
        VALUES (
                    'Poison Dart Frog',
                    species_mgr.location_id_from_name ('Amazon'));

INSERT INTO endangered_species (name, location_id)
     VALUES ('Macaw', species_mgr.location_id_from_name ('Amazon'));

INSERT INTO endangered_species (name, location_id)
     VALUES ('Jaguar', species_mgr.location_id_from_name ('Amazon'));

INSERT INTO research_locations (name, location_id)
        VALUES (
                    'Smithsonian Tropical Research Institute',
                    species_mgr.location_id_from_name ('Amazon'));
```

```
INSERT INTO research_locations (name, location_id)
     VALUES ('IBAMA', species_mgr.location_id_from_name ('Amazon'));

/* Galapagos Data */

INSERT INTO species_locations (location)
     VALUES ('Galapagos');

INSERT INTO endangered_species (name, location_id)
        VALUES (
                  'Sea Turtle',
                  species_mgr.location_id_from_name ('Galapagos'));

INSERT INTO endangered_species (name, location_id)
        VALUES (
                  'Leatherback Turtle',
                  species_mgr.location_id_from_name ('Galapagos'));

INSERT INTO endangered_species (name, location_id)
     VALUES ('SEI Whale', species_mgr.location_id_from_name ('Galapagos'));

INSERT INTO endangered_species (name, location_id)
        VALUES (
                  'Green Turtle',
                  species_mgr.location_id_from_name ('Galapagos'));
```

```
INSERT INTO endangered_species (name, location_id)
        VALUES (
                  'Giant Tortoise',
                  species_mgr.location_id_from_name ('Galapagos'));

INSERT INTO endangered_species (name, location_id)
        VALUES (
                  'Galapagos Penguin',
                  species_mgr.location_id_from_name ('Galapagos'));

INSERT INTO research_locations (name, location_id)
        VALUES (
                  'Charles Darwin Foundation',
                  species_mgr.location_id_from_name ('Galapagos'));

   INSERT INTO research_locations (name, location_id)
           VALUES (
                     'Galapagos Science Center',
                     species_mgr.location_id_from_name ('Galapagos'));

   COMMIT;
END;
/
```

Finally, I create two additional PL/SQL procedures—add_species and show_location_info—that work with these three tables, as shown in **Listing 4**.

Code Listing 4: Creating the add_species and show_location_info procedures

```
CREATE OR REPLACE PROCEDURE add_species (NAME_IN       IN VARCHAR2,
                                         location_in   IN VARCHAR2)
   AUTHID DEFINER
IS
   l_info   VARCHAR2 (32767);
BEGIN
   INSERT INTO endangered_species (name, location_id)
        VALUES (NAME_IN, species_mgr.location_id_from_name (location_in));

   SELECT s.id || '-' || s.name || '-' || l.location
     INTO l_info
     FROM endangered_species s, species_locations l
    WHERE s.location_id = l.id;

   DBMS_OUTPUT.put_line ('Inserted ' || l_info);
END;
/

CREATE OR REPLACE PROCEDURE show_location_info (
   location_id_in   IN species_locations.id%TYPE)
   AUTHID DEFINER
IS
BEGIN
   FOR rec IN (  SELECT r.location_id, r.name
                   FROM research_locations r
```

```
                  WHERE r.location_id = show_location_info.location_id_in
               ORDER BY r.name)
   LOOP
      DBMS_OUTPUT.put_line (rec.location_id || '-' || rec.name);
   END LOOP;
END;
/
```

I've got my code base. Now I'd like to do some analysis of it.

PROGRAM UNIT "DUMP" OF IDENTIFIERS AND STATEMENTS

Let's take a look at what PL/Scope can tell us about the add_species procedure. And let's start with the views that will tell us what's going on.

The two views with PL/Scope-generated information, ALL_IDENTIFIERS and ALL_STATEMENTS, share many columns, such as USAGE_ID and SIGNATURE. And there are columns unique to each, such as STATEMENT_TYPE in ALL_STATEMENTS.

The USAGE_ID value is unique across the ALL_IDENTIFIERS and ALL_STATEMENTS views for a specific program unit. That makes it easier to understand how statements and identifiers relate to each other.

The query in **Listing 5** executes a union of these two views for all rows related to add_species. It uses the CONNECT BY syntax to display identifiers and statements intuitively.

Code Listing 5: Looking at ALL_IDENTIFIERS and ALL_STATEMENTS for references to add_species

```
WITH one_obj_name AS (SELECT 'ADD_SPECIES' object_name FROM DUAL)
```

```
SELECT plscope_type,
       usage_id,
       usage_context_id,
       LPAD (' ', 2 * (LEVEL - 1)) || usage || ' ' || name usages,
       line,
       col
  FROM (SELECT 'ID' plscope_type,
               ai.object_name,
               ai.usage usage,
               ai.usage_id,
               ai.usage_context_id,
               ai.TYPE || ' ' || ai.name name,
               ai.line,
               ai.col
          FROM all_identifiers ai, one_obj_name
         WHERE ai.object_name = one_obj_name.object_name
        UNION ALL
        SELECT 'ST',
               st.object_name,
               st.TYPE,
               st.usage_id,
               st.usage_context_id,
               'STATEMENT',
               st.line,
               st.col
          FROM all_statements st, one_obj_name
```

```
                WHERE st.object_name = one_obj_name.object_name)
START WITH usage_context_id = 0
CONNECT BY PRIOR usage_id = usage_context_id
```

Figure 1 (on the next page) displays the output of the **Listing 5** query from Oracle SQL Developer.

Here are some important takeaways from this report:

- A USAGE_CONTEXT_ID value in ALL_IDENTIFIERS could well reference a row in the ALL_STATEMENTS table (and vice versa).
- Column identifiers are not declared in the ALL_IDENTIFIERS view, as, say, a variable would be (see line 7 in **Figure 1** for a variable declaration). There are simply references.
- Invocations of user-defined functions in SQL statements are now tracked by PL/Scope (see line 15 in **Figure 1**).

WHICH PROGRAM UNITS USE ENDANGERED_SPECIES .LOCATION_ID?

Here's a very common developer question: where and how is a table used in my PL/SQL code? With ALL_DEPENDENCIES, I can find out if a table is *referenced*, but that reference could be from a %ROWTYPE declaration or a SELECT or a DELETE. There's no way of telling which. And I certainly can't find out where and how a particular column is being used.

In addition, I could easily enough write a query against ALL_SOURCE to find all the program units that contain the string "LOCATION_ID":

```
SELECT name, line, text
  FROM ALL_SOURCE
```

Figure 1: Result of query for references to add_species in ALL_IDENTIFIERS and ALL_STATEMENTS

	PLSCOPE_TYPE	USAGE_ID	USAGE_CONTEXT_ID	USAGES	LINE	COL
1	ID	1	0	DECLARATION PROCEDURE ADD_SPECIES	1	11
2	ID	2	1	DEFINITION PROCEDURE ADD_SPECIES	1	11
3	ID	3	2	DECLARATION FORMAL IN NAME_IN	1	24
4	ID	4	3	REFERENCE CHARACTER DATATYPE VARCHAR2	1	41
5	ID	5	2	DECLARATION FORMAL IN LOCATION_IN	2	42
6	ID	6	5	REFERENCE CHARACTER DATATYPE VARCHAR2	2	59
7	ID	7	2	DECLARATION VARIABLE L_INFO	5	4
8	ID	8	7	REFERENCE CHARACTER DATATYPE VARCHAR2	5	13
9	ST	9	2	INSERT STATEMENT	7	4
10	ID	10	9	REFERENCE TABLE ENDANGERED_SPECIES	7	16
11	ID	11	9	REFERENCE FORMAL IN NAME_IN	8	17
12	ID	12	9	REFERENCE COLUMN LOCATION_ID	7	42
13	ID	13	9	REFERENCE COLUMN NAME	7	36
14	ID	14	9	REFERENCE PACKAGE SPECIES_MGR	8	26
15	ID	15	14	CALL FUNCTION LOCATION_ID_FROM_NAME	8	38
16	ID	16	15	REFERENCE FORMAL IN LOCATION_IN	8	61
17	ST	17	2	SELECT STATEMENT	10	4
18	ID	18	17	REFERENCE TABLE SPECIES_LOCATIONS	12	33
19	ID	19	17	REFERENCE TABLE ENDANGERED_SPECIES	12	11
20	ID	20	17	REFERENCE COLUMN ID	13	29
21	ID	21	17	REFERENCE COLUMN LOCATION_ID	13	13
22	ID	22	17	REFERENCE COLUMN LOCATION	10	45
23	ID	23	17	REFERENCE COLUMN NAME	10	28
24	ID	24	17	REFERENCE COLUMN ID	10	13
25	ID	25	17	ASSIGNMENT VARIABLE L_INFO	11	11

```
WHERE upper (text) LIKE '%LOCATION_ID%'
```

Unfortunately, that query will return all kinds of false positives, including

- That phrase showing up inside a comment
- That phrase as part of a larger identifier, such as GET_LOCATION_ID
- Columns with the same name but in other tables (such as RESEARCH_LOCATIONS)

With PL/Scope in Oracle Database 12.2, you can now definitively locate all references to the LOCATION_ID column of the ENDANGERED_SPECIES table and nothing else.

Of course, it would be nice to not have to write a special-purpose query every time you want to locate references to *another* column, so I've written the procedure in **Listing 6** for just that purpose.

Code Listing 6: Package for looking for column references in ALL_IDENTIFIERS and ALL_STATEMENTS

```
CREATE OR REPLACE PROCEDURE show_column_usages (owner_in    IN VARCHAR2,
                                                table_in    IN VARCHAR2,
                                                column_in   IN VARCHAR2)
   AUTHID DEFINER
IS
BEGIN
   DBMS_OUTPUT.put_line (
      'References to ' || owner_in || '.' || table_in || '.' || column_in);
   DBMS_OUTPUT.put_line ('');

   FOR rec
```

```
IN (  SELECT idt.line,
             idt.owner || '.' || idt.object_name code_unit,
             RTRIM (src.text, CHR (10)) text
        FROM all_identifiers idt, all_source src
       WHERE     idt.usage = 'REFERENCE'
             AND idt.signature =
                    (SELECT idt_inner.signature
                       FROM all_identifiers idt_inner
                      WHERE idt_inner.object_name = show_column_usages.table_in
                            AND idt_inner.TYPE = 'COLUMN'
                            AND idt_inner.name = show_column_usages.column_in
                            AND idt_inner.usage = 'DECLARATION'
                            AND idt_inner.owner = owner_in)
                 AND idt.line = src.line
                 AND idt.object_name = src.name
                 AND idt.owner = src.owner
        ORDER BY code_unit, line)
   LOOP
      DBMS_OUTPUT.put_line (
         'In ' || rec.code_unit || ' on ' || rec.line || ': ' || rec.text);
   END LOOP;
END;
/
```

When I execute the show_column_usages procedure as follows:

```
BEGIN
   show_column_usages (owner_in    => USER,
                       table_in    => 'ENDANGERED_SPECIES',
                       column_in   => 'LOCATION_ID');
END;
/
```

I see the following results:

```
STEVEN.ENDANGERED_SPECIES.LOCATION_ID References
In STEVEN.ADD_SPECIES on 7:    INSERT INTO endangered_species (name, location_id)
In STEVEN.ADD_SPECIES on 13:     WHERE s.location_id = l.id;
In STEVEN.SHOW_SPECIES on 8:    INSERT INTO endangered_species (id, name, location_id)
In STEVEN.SHOW_SPECIES on 14:     WHERE s.location_id = l.id;
```

Note that there are no references to the RESEARCH_LOCATIONS table. Nice!

Let's take a look at how the show_column_usages procedure gets its job done.

First, interestingly, note that I am not even querying the new-to-Oracle Database 12.2 ALL_STATEMENTS view. That's because I am not interested in SQL statements, per se—simply which SQL identifiers are used, and where. So ALL_IDENTIFIERS contains all the information I need.

Next, because I am looking for all occurrences of a specific identifier, I am working with the SIGNATURE column. PL/Scope assigns a unique identifier to each distinct

identifier in my code base, and that's what is stored in the SIGNATURE column.

So the first thing I need to do is find the signature of the LOCATION_ID column. The inner query inside the show_column_usages procedure does this for me:

```
SELECT idt_inner.signature
  FROM all_identifiers idt_inner
 WHERE     idt_inner.object_name = show_column_usages.table_in
       AND idt_inner.owner = owner_in
       AND idt_inner.name = show_column_usages.column_in
       AND idt_inner.TYPE = 'COLUMN'
       AND idt_inner.usage = 'DECLARATION'
```

This query says: get the signature of the *declaration* of the *column* identifier, for the names specified by the procedure parameters. I then use that inner query for an equality match on idt.signature. In the query below, for readability purposes, I use <my_column_signature> in place of the full query. Then the outer query becomes

```
SELECT idt.line,
       idt.owner || '.' || idt.object_name code_unit,
       RTRIM (src.text, CHR (10)) text
  FROM all_identifiers idt, all_source src
 WHERE     idt.usage = 'REFERENCE'
       AND idt.signature = <my_column_signature>
       AND idt.line = src.line
       AND idt.object_name = src.name
       AND idt.owner = src.owner
```

This query finds all the *references* to my_column_signature in ALL_IDENTIFIERS. I then join those identifier rows to the ALL_SOURCE table, so that I can display the line of source code.

Hey, that's not so hard!

HOW ARE MY TABLES USED?

The primary purpose of PL/SQL is to build APIs around data structures, controlling access to tables and ensuring data integrity and consistency. This means that PL/SQL code is usually full of SQL statements, and that can make it difficult to keep track of which tables are used where and how. With PL/Scope in Oracle Database 12.2, it is now easy to get answers to questions such as "Which program units perform an INSERT into a particular table?" The show_dml_statements_on procedure in **Listing 7** provides a way to answer this and other data manipulation language (DML) use questions.

Code Listing 7: The show_dml_statements_on procedure

```
CREATE OR REPLACE PROCEDURE show_dml_statements_on (
   owner_in            IN VARCHAR2,
   table_in            IN VARCHAR2,
   statement_type_in   IN VARCHAR2 DEFAULT 'ALL')
IS
BEGIN
   DBMS_OUTPUT.put_line (
         'Locations of '
      || statement_type_in
      || ' DML Statements Against '
```

```
      || owner_in
      || '.'
      || table_in);
   DBMS_OUTPUT.put_line ('');

   FOR rec
      IN (SELECT idt.line,
                 idt.owner || '.' || idt.object_name code_unit,
                 RTRIM (src.text, CHR (10)) text
            FROM all_identifiers idt, all_source src, all_statements st
           WHERE     idt.usage = 'REFERENCE'
                 AND idt.TYPE = 'TABLE'
                 AND idt.name = table_in
                 AND idt.owner = owner_in
                 AND idt.line = src.line
                 AND idt.object_name = src.name
                 AND idt.owner = src.owner
                 AND idt.usage_context_id = st.usage_id
                 AND (   st.TYPE = statement_type_in
                      OR statement_type_in = 'ALL'))
      LOOP
         DBMS_OUTPUT.put_line (
            'In ' || rec.code_unit || ' on ' || rec.line || ': ' || rec.text);
      END LOOP;
   END;
   /
```

When I execute the show_dml_statements_on procedure with this block:

```
BEGIN
   show_dml_statements_on (USER, 'ENDANGERED_SPECIES', 'INSERT');
END;
/
```

I see the following output:

```
Locations of INSERT DML Statements Against STEVEN.ENDANGERED_SPECIES

In STEVEN.ADD_SPECIES on 7:    INSERT INTO endangered_species (name, location_id)
In STEVEN.SHOW_SPECIES on 8:    INSERT INTO endangered_species (id, name, location_id)
```

IDENTIFYING SQL INJECTION VULNERABILITIES

SQL injection presents one of the greatest vulnerabilities for database developers, and dynamic SQL is *the* path followed by those who want to inject malicious code.

So it would be very helpful to be able to easily identify all program units that execute dynamic SQL or dynamic PL/SQL statements.

Prior to Oracle Database 12.2, the best you could do was to write queries against ALL_SOURCE to look for strings such as EXECUTE IMMEDIATE and OPEN <cursor> FOR and DBMS_SQL. It is, however, very difficult to make those queries cover all possible cases without, in effect, parsing PL/SQL code. For example, I could write code like this:

```
BEGIN
   EXECUTE /* a comment */ IMMEDIATE ...
END;
```

Now with the enhanced PL/Scope in Oracle Database 12.2, it's a breeze to write a query to unambiguously locate all native dynamic SQL statements, because two possible values in the TYPE column of ALL_STATEMENTS are EXECUTE IMMEDIATE and OPEN.

In addition, if you are able to convince your DBA to recompile the SYS.DBMS_SQL package with PL/Scope turned on, you can then query the contents of ALL_IDENTIFIERS for program units that reference that built-in package in any way, as shown in **Figure 2**.

Figure 2: A better search for SQL injection risks in ALL_IDENTIFIERS

```
 1  SELECT st.owner,
 2         st.object_name,
 3         st.line,
 4         s.text
 5    FROM all_statements st, all_source s
 6   WHERE     st.TYPE IN ('EXECUTE IMMEDIATE', 'OPEN')
 7         AND st.owner = s.owner
 8         AND st.object_name = s.name
 9         AND st.line = s.line
10  UNION ALL
11  SELECT idnt.owner,
12         idnt.object_name,
13         idnt.line,
14         src.text
15    FROM all_identifiers idnt, all_source src
16   WHERE     idnt.owner <> 'SYS'
17         AND idnt.signature IN (SELECT a.signature
18                                  FROM all_identifiers a
19                                 WHERE     a.usage = 'DECLARATION'
20                                       AND a.owner = 'SYS'
21                                       AND a.object_name = 'DBMS_SQL'
22                                       AND a.object_type = 'PACKAGE')
23         AND idnt.owner = src.owner
24         AND idnt.object_name = src.name
25         AND idnt.line = src.line;
```

In **Figure 2**, lines 1 through 9 show how easy it is to find all statements in your PL/SQL code base that take advantage of native dynamic SQL. Lines 11 through 25 show how to find all references to the DBMS_SQL package. Note in particular the subquery on lines 17 through 22. I get *all* the signatures of identifiers declared in the DBMS_SQL package specification. I then look, in line 16, for any matches for those signatures across my code base, excluding any code owned by SYS.

GETTING A HANDLE ON SQL HINTS

As perhaps *too* many developers know, you can add hints to your SQL statements to direct the optimizer to alter the plan it comes up with for the execution of your statement. Tom Kyte has called hints a "path of last resort," because (1) there are usually ways to change the SQL statement or update statistics to fix the problem, (2) your hints might block the optimizer from coming up with a better plan, and (3) data changes over time and could cause a hint that works today to cause big problems tomorrow.

Unfortunately, when developers are working hard on deadline, they will take the shortest, quickest path to a solution that works *right now*. Over time, applications can end up littered with hints that cause problems that are hard to track down.

But tracking down old hints is not so hard in Oracle Database 12.2. First, compile your program units with statement gathering enabled. Then run the following query against ALL_STATEMENTS to find all statements with one or more hints:

```
SELECT owner,
       object_name,
       line,
       full_text
  FROM all_statements
 WHERE has_hint = 'YES'
```

KNOW THY CODE, IMPROVE THY CODE

Being able to slice and dice code in a variety of ways to analyze the impact of changes or understand more fully how that code is written is critical to your success. Enhancements in PL/Scope in Oracle Database 12*c* Release 2 give you pow-

erful new tools for achieving a deep and thorough understanding of your code. This improved understanding will result in the development and delivery of higher-quality applications.

Steven Feuerstein is Oracle's Developer Advocate for PL/SQL, offering guidance on the PL/SQL language through books, quizzes, videos, and more.

PHOTOGRAPHY BY **ANDREA MANDEL**

NEXT STEPS

LEARN more about Oracle Database 12*c.*

MEET the Oracle Developer Advocates team.

TEST your SQL and PL/SQL knowledge.

READ more about PL/Scope.

WATCH Feuerstein's *Practically Perfect PL/SQL* videos.

READ more about binding PL/SQL datatypes.

By Connor McDonald

ORACLE DATABASE 12*c* RELEASE 2

Long and Overflowing

LISTAGG in Oracle Database 12*c* Release 2 solves the problem of excessively long lists.

In the next few columns, I'll spend some time looking at new features in Oracle Database 12*c* Release 2. These features come from the "12 Things About Oracle Database 12*c*" presentation series that Chris Saxon and I, the Ask Tom team, gave at Oracle OpenWorld 2016 in San Francisco. (You can find the slides for these presentations on asktom.oracle.com, under the **Resources** tab). In this article, I'll take a look at improvements to the LISTAGG function in Oracle Database 12*c* Release 2.

BACKGROUND

The relational model was first described by the pioneer of relational theory, Edgar F. Codd. In 1970 he published his paper "A Relational Model of Data for Large Shared Data Banks," in what would become the catalyst for the creation of commercial relational database management systems (RDBMSs), including Oracle Database. The core principle of the relational model is that "all data is represented as mathematical n-ary

relations, an n-ary relation being a subset of the Cartesian product of n domains." Thus the origins of relational databases are based on strict mathematical theory, such databases being a mechanism to store data based on the relational model.

Of course, mention *Cartesian products* or *n-ary relations* in casual conversation, and quickly you'll find yourself standing alone. Most users of databases have a much simpler definition of *database*: "It's a spreadsheet, only bigger" with third normal form being "something the data architect did to make our lives harder." Hence, as long as databases have existed, there has been a requirement to transform the data from how it is *stored* in database tables to a format that users feel is more natural to be *displayed* by applications on their devices. One of the common transformations is to transpose and consolidate rows into a list. For example, given a set of employees identified by name and associated with a department

```
SQL> select deptno, ename
  2  from   emp
  3  order by 1,2;

    DEPTNO ENAME
__________ __________
        10 CLARK
        10 KING
        10 MILLER
        20 ADAMS
        20 FORD
        20 JONES
        20 SCOTT
```

```
20 SMITH
30 ALLEN
30 BLAKE
30 JAMES
30 MARTIN
30 TURNER
30 WARD
```

I may want to produce a concatenated list of employee names where the concatenation is segmented into the respective departments:

```
DEPTNO MEMBERS
------ ------------------------------
    10 CLARK,KING,MILLER
    20 SMITH,JONES,SCOTT,ADAMS,FORD
    30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

In older versions of Oracle Database, this was quite a difficult problem. There were various techniques that members of the Oracle developer community provided to meet the requirement. For example, in Oracle9*i* Database, user-defined aggregate functions could be used to achieve the desired result, as shown in **Listing 1**.

Code Listing 1: User-defined aggregate functions concatenate and segment results

```
SQL> create or replace type string_agg_type as object
  2  (
  3     total varchar2(4000),
```

```
 4
 5     static function
 6          ODCIAggregateInitialize(sctx IN OUT string_agg_type )
 7          return number,
 8
 9     member function
10          ODCIAggregateIterate(self IN OUT string_agg_type ,
11                               value IN varchar2 )
12          return number,
13
14     member function
15          ODCIAggregateTerminate(self IN string_agg_type,
16                                 returnValue OUT  varchar2,
17                                 flags IN number)
18          return number,
19
20      member function
21          ODCIAggregateMerge(self IN OUT string_agg_type,
22                             ctx2 IN string_agg_type)
23          return number
24  );

SQL> create or replace type body string_agg_type
  2  is
  3
  4    static function ODCIAggregateInitialize(sctx IN OUT string_agg_type)
```

```
 5    return number
 6    is
 7    begin
 8      sctx := string_agg_type( null );
 9      return ODCIConst.Success;
10    end;
11
12    member function ODCIAggregateIterate(self IN OUT string_agg_type,
13    value IN varchar2 )
14    return number
15    is
16    begin
17      self.total := self.total || ',' || value;
18      return ODCIConst.Success;
19    end;
20
21    member function ODCIAggregateTerminate(self IN string_agg_type,
22    returnValue OUT varchar2,
23    flags IN number)
24    return number
25    is
26    begin
27      returnValue := ltrim(self.total,',');
28      return ODCIConst.Success;
29    end;
30
```

```
31    member function ODCIAggregateMerge(self IN OUT string_agg_type,
32    ctx2 IN string_agg_type)
33    return number
34    is
35    begin
36      self.total := self.total || ctx2.total;
37      return ODCIConst.Success;
38    end;
39
40  end;
```

With Oracle Database 10*g*, the MODEL clause or the SYS_CONNECT_BY_PATH hierarchical query function could be used to achieve the same result with less coding, but comprehending the mechanism of precisely *how* the code worked in each case still required careful thought, as shown in **Listing 2**.

Code Listing 2: MODEL clause and SYS_CONNECT_BY_PATH concatenate and segment results

```
SQL> select deptno , rtrim(ename,',') enames
  2  from ( select deptno,ename,rn
  3         from emp
  4         model
  5         partition by (deptno)
  6         dimension by (
  7            row_number() over
  8             (partition by deptno order by ename) rn
  9               )
```

```
 10         measures (cast(ename as varchar2(40)) ename)
 11         rules
 12         ( ename[any]
 13             order by rn desc = ename[cv()]||','||ename[cv()+1])
 14         )
 15   where rn = 1
 16   order by deptno;

SQL> select deptno,
  2         substr(max(sys_connect_by_path(ename, ',')), 2) members
  3  from (select deptno, ename,
  4                row_number ()
  5                over (partition by deptno order by empno) rn
  6        from emp)
  7  start with rn = 1
  8  connect by prior rn = rn - 1
  9  and prior deptno = deptno
 10  group by deptno;
```

And with Oracle Database 11*g*, database rows could be treated as an XML structure, thereby enabling an XML stylesheet to transpose the rows into the desired result, as shown in **Listing 3**.

Code Listing 3: XML stylesheet transposes rows into desired result

```
SQL> select deptno,
  2     xmltransform
```

```
    ( sys_xmlagg
       ( sys_xmlgen(ename)
       ),
      xmltype
      (
        '<?xml version="1.0"?><xsl:stylesheet version="1.0"
           xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
           <xsl:template match="/">
             <xsl:for-each select="/ROWSET/ENAME">
               <xsl:value-of select="text()"/>;</xsl:for-each>
           </xsl:template>
         </xsl:stylesheet>'
       )
    ).getstringval() members
from emp
group by deptno;
```

All of these solutions are a credit to the ingenuity of their original coders, but in Oracle Database 11*g* Release 2, the task became simple, with the arrival of the LISTAGG function. With LISTAGG the concatenated employee list is achieved in just a few lines of easily understood SQL:

```
SQL> select deptno,
  2         listagg( ename, ',')
  3              within group (order by empno) members
  4  from   emp
```

```
  5  group  by deptno;

    DEPTNO MEMBERS
__________ ____________________________________________
        10 CLARK,KING,MILLER
        20 SMITH,JONES,SCOTT,ADAMS,FORD
        30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

I regularly conduct presentations on the topic of SQL analytic functions, the umbrella under which the LISTAGG function resides, and there is always a collective sigh of relief when LISTAGG is demonstrated and compared to the coding solutions that were required historically.

LONG LISTS

Perusing *Oracle Database Reference* in the Oracle documentation library reveals, as expected, that there is no limit on the number of rows in a database table.

However, there *is* a limit on the length of a VARCHAR2 column. And LISTAGG, by its very definition, can generate VARCHAR2 results of arbitrary length, dependent on the data contained within the source table. Having a few dozen more employees for department 10 in the EMPLOYEE table does not cause any problems:

```
SQL> select deptno,
  2         listagg( ename, ',')
  3              within group (order by empno) members
  4  from   emp
  5  group  by deptno;
```

```
  DEPTNO MEMBERS
________ ________________________________________________________________

      10 BROOKS,DOMINGUEZ,SIMON,SOLOMON,HAYS,MACIAS,HUDSON,LYNCH,CONN
         ER,CONRAD,IBARRA,MITCHELL,COMBS,WILCOX,SANTIAGO,FLOWERS,SPEN
         CER,WIGGINS,CASTRO,BENNETT,MCCARTHY,BROCK,CALHOUN,LAM,JACOBS
         ,GRIFFITH,WERNER,DUNCAN,GILMORE,MORALES,WEEKS,SIMS,OWEN,MCNE
         IL,CUMMINGS,ROY,HAYES,LUNA,MORA,ROBERSON,BURNETT,SNYDER,MAYS
         ,SHAFFER,MCDANIEL,BELTRAN,POTTER,MYERS,MONTES,SCHMIDT,CLARK,
         KING,MILLER

      20 SMITH,JONES,SCOTT,ADAMS,FORD
      30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

But if I continue to extend the example by adding hundreds more employees to department 10, LISTAGG will reach the VARCHAR2 limit and the query will no longer execute successfully.

```
SQL> select deptno,
  2           listagg( ename, ',')
  3               within group (order by empno) members
  4  from   t
  5  group  by deptno;
select deptno,
*
ERROR at line 1:
ORA-01489: result of string concatenation is too long
```

Handling this error is a "chicken and egg" problem, because I won't know if the length of the result of the LISTAGG expression exceeds the limits for a VARCHAR2 until I actually exceed it. There is no way to intercept the LISTAGG processing at the penultimate point where the logical "next" element would exceed the VARCHAR2 limit.

LISTAGG WITH ORACLE DATABASE 12*c* RELEASE 2

A typical behavior on a website that displays lengthy text is that if that text exceeds what the site deems "reasonable," the text will be truncated and suffixed in some fashion to indicate that the entirety of the text was not displayed. With Oracle Database 12*c* Release 2, this functionality has been added to the LISTAGG function.

I'll return to the previous example, but this time I'll include the new ON OVERFLOW TRUNCATE clause in the LISTAGG specification:

```
SQL> select deptno,
  2         listagg( ename, ',' ON OVERFLOW TRUNCATE)
  3              within group (order by empno) members
  4  from   emp
  5  group  by deptno;

  DEPTNO MEMBERS
________ ______________________________________________________________

      10 BROOKS,DOMINGUEZ,SIMON,SOLOMON,HAYS,MACIAS,HUDSON,LYNCH,CONN
         ER,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,C
         [lines omitted]
         TER,POTTER,POTTER,POTTER,POTTER,POTTER,POTTER,MYERS,MYERS,MY
```

```
ERS,MYERS,MYERS,MYERS,MYERS,MYERS,MONTES,MONTES,MONTES,MONTE
S,MONTES,...(1806)

20 SMITH,JONES,SCOTT,ADAMS,FORD
30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

Instead of returning an error message, the LISTAGG function now truncates the full string but appends the familiar ellipsis (...) suffix by default. Also provided in this result is "1806," the number of characters truncated to enable LISTAGG to return a valid-length VARCHAR2 string.

I have control over these parameters, using variations on the OVERFLOW clause. For example, I can specify what suffix I prefer if ellipses are not appropriate. In the example below, I've used the string "[snip]" to indicate that some of the output was "snipped."

```
SQL> select deptno,
  2         listagg( ename, ',' ON OVERFLOW TRUNCATE '[snip]')
  3              within group (order by empno) members
  4  from   emp
  5  group  by deptno;

  DEPTNO MEMBERS
________ ________________________________________________________________

      10 BROOKS,DOMINGUEZ,SIMON,SOLOMON,HAYS,MACIAS,HUDSON,LYNCH,CONN
         ER,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,C
         [lines omitted]
```

```
TER,POTTER,POTTER,POTTER,POTTER,POTTER,POTTER,MYERS,MYERS,MY
ERS,MYERS,MYERS,MYERS,MYERS,MYERS,MONTES,MONTES,MONTES,MONTE
S,MONTES,[snip](1807)

20 SMITH,JONES,SCOTT,ADAMS,FORD
30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

Similarly, if I do not want to display the count of characters that have exceeded the allowable length of the VARCHAR2 result, I can specify the additional WITHOUT COUNT clause.

```
SQL> select deptno,
  2         listagg( ename, ',' ON OVERFLOW TRUNCATE '[snip]'
WITHOUT COUNT)
  3              within group (order by empno) members
  4  from   emp
  5  group  by deptno;

  DEPTNO MEMBERS
_______ ____________________________________________________________

      10 BROOKS,DOMINGUEZ,SIMON,SOLOMON,HAYS,MACIAS,HUDSON,LYNCH,CONN
         ER,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,BROOKS,C
         [lines omitted]
         TER,POTTER,POTTER,POTTER,POTTER,POTTER,POTTER,MYERS,MYERS,MY
         ERS,MYERS,MYERS,MYERS,MYERS,MYERS,MONTES,MONTES,MONTES,MONTE
         S,MONTES,[snip]
```

```
20 SMITH,JONES,SCOTT,ADAMS,FORD
30 ALLEN,WARD,MARTIN,BLAKE,TURNER,JAMES
```

Without any of these additional clauses, LISTAGG works as it did in Oracle Database 12*c* Release 1 and earlier, although this operation is, in fact, an application of the new—and default—ON OVERFLOW ERROR clause. If I want LISTAGG to truncate the output with no indication that truncation has actually occurred—for example, if I simply want to avoid an error—I can specify the ON OVERFLOW TRUNCATE clause with a null string as the suffix.

SUMMARY

The LISTAGG function was a long-awaited addition to the suite of analytic functions available in Oracle Database. With Oracle Database 12*c* Release 2, extensions to the function make it an even more useful tool for database developers.

Connor McDonald is an Oracle Developer Advocate for SQL. His passions are database design, SQL, and PL/SQL. He can answer your database questions on Ask Tom.

PHOTOGRAPHY BY
SABINE ALBERS/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Database 12*c* Release 2.

DOWNLOAD Oracle Database 12*c*.

ISV Apps Run Better on Oracle Cloud

- ✓ Run applications without rewriting
- ✓ Advanced multitenant capabilities
- ✓ Highly secure cloud platform
- ✓ Rapid app creation and integration
- ✓ Lower cost development, deployment, and operation

Over 10,000 ISV Apps Are Ready to Run on Oracle Cloud

ORACLE®

By Melanie Caffrey

ORACLE DATABASE

Sequential Additions and Different Points of View

Part 7 in a second series on the basics of the relational database and SQL

This article is the seventh in a series that helps you build on the fundamentals you learned in the 12-part SQL 101 series in *Oracle Magazine*. The previous Beyond SQL 101 article, "Defining, Constraining, and Manipulating Your Entities," introduced you to common constraints, including primary key, foreign key, not null, check, and unique constraints. You learned how these constraints can be violated when data is inserted that doesn't conform to the constraint conditions. You discovered how to create a table with the same structure as another table by using the CREATE TABLE AS SELECT (CTAS) method. You also discovered how default values can be used for columns you want to be regularly populated. Finally, you were introduced to column definition manipulation via the ALTER TABLE command and several of its options.

In this article, you'll learn more about using Oracle data definition language (DDL):

- How to rename tables
- How to drop tables
- How to recover tables by using the *recycle bin* and how *purging* syntax affects table recovery
- The difference between dropping and truncating tables
- How a virtual column is created and how it is used

You'll also be introduced to *sequences* and IDENTITY columns and learn how they can be used to help guarantee unique values for primary keys. Last, you'll discover how you can use a *view* to simplify query writing and data hiding.

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 1 (12.1.0.2.0). 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 used for this article's examples. (View the script in a text editor for execution instructions.)

AN ENTITY BY ANY OTHER NAME

In the previous Beyond SQL 101 article, "Defining, Constraining, and Manipulating Your Entities," you learned how to rename a table column. Similarly, you can rename a table. In **Listing 1**, for example, the EMPLOYEE_EXAMPLE table is renamed EMPLOYEE_EXTRA and users can no longer query the table with its former name.

Code Listing 1: Renaming the EMPLOYEE_EXAMPLE table EMPLOYEE_EXTRA

```
SQL> set lines 10000
SQL> ALTER TABLE employee_example RENAME TO employee_extra;

Table renamed.

SQL> select *
  2    from employee_example;
  from employee_example
       *
ERROR at line 2:
ORA-00942: table or view does not exist

SQL> select *
  2    from employee_extra;

EMPLOYEE_ID FIRST_NAME LAST_NAME   SALARY     MANAGER DEPARTMENT_ID LAST_UPDA
___________ __________ _________ _________ ___________ _____________ _________

          1 Ram        Burela      100000                            11-SEP-16
          2 Seema      Patel                                         12-SEP-16
```

```
2 rows selected.
```

A table's constraint names are not changed when the table is renamed. However, a *view* (introduced later in this article) and other database objects and code—such as *synonyms* and PL/SQL (to be outlined in subsequent articles in this series)—can be affected and might require *recompilation*. (See the "Recompilation" sidebar for information on code and compiling.) Additionally, a table rename has no impact on any privileges a user may have had on the table before it was renamed. A user who could query EMPLOYEE_EXAMPLE can similarly query EMPLOYEE_EXTRA.

WHAT IS LOST CAN BE FOUND

When you no longer need a table, you can *drop* (remove) it. As **Listing 2** shows, a dropped table can no longer be queried; the table, its data and constraints, and other dependent objects are all removed.

Code Listing 2: Dropping the EMP2 table

```
SSQL> select *
  2    from emp2;

no rows selected

SQL> drop table emp2;

Table dropped.

SQL> select *
```

```
  2    from emp2;
  from emp2
       *
ERROR at line 2:
ORA-00942: table or view does not exist
```

Oracle Database puts tables that have been dropped via the syntax in **Listing 2** in a *recycle bin* similar to the one on a computer running the Windows operating system. You can restore deleted items from the recycle bin, and you can delete dropped items permanently by purging them from the recycle bin. Note, however, that dropped tables and other objects remain in the Oracle Database recycle bin only if the recycle bin has sufficient free space for them. Your dropped objects might be in the recycle bin for years or for a millisecond, depending on the amount of free space. The query in **Listing 3** shows that the dropped EMP2 table is in the recycle bin.

Code Listing 3: Locating the EMP2 table in the recycle bin

```
SQL> select object_name, original_name, type
  2    from user_recyclebin;

OBJECT_NAME                    ORIGINAL_NAME                    TYPE
______________________________ ________________________________ _______

BIN$P7tLNcHrFG3gUwEAAH/yFg==$0 EMP2                             TABLE

1 row selected.
```

RECOMPILATION

Any DDL or DML you write is also referred to as *source code*. Oracle Database takes source code and hands it to a program called a *compiler*, which transforms the source code into *object code* (also referred to as *bytecode*, much like that used by the Java programming language) that is readable by Oracle Database. This transformation action is known as *compilation*.

When you change your source code (such as by renaming a table), certain database objects that are dependent on that source code may require recompilation. For example, views, synonyms, and PL/SQL are dependent upon table names. When these objects are accessed, Oracle Database checks to ensure that their definition and the definition of the database objects upon which they are dependent have not changed. If a change has taken place, Oracle Database will recompile the objects before fulfilling an access request.

If you want to reinstate the EMP2 table, you can *flash* it back and effectively undo your drop action. Oracle Database contains *flashback* technology that enables users to

- Query data as it existed in the past
- Undo a statement or a transaction
- Restore a dropped table
- Restore a database to its pre-existing state

Code Listing 4: Restoring the EMP2 table

```
SQL> FLASHBACK TABLE emp2 TO BEFORE DROP;

Flashback complete.

SQL> select *
  2    from emp2;

no rows selected
```

To be able to restore a dropped table, the user must have either the privilege to flash back a specific table or to flash back any table. **Listing 4** shows how to flash a table back to its original state. Note that database objects and code such as *indexes* and *triggers* (to be outlined in subsequent articles in this series) and constraints do not automatically revert to their original names when they are restored to their original state. Instead, they are restored with the <object_name> from the recycle bin.

You must manually ensure that they are renamed correctly after you've executed the FLASHBACK TABLE <table_name> TO BEFORE DROP statement. Once the table is restored, it can be queried without returning the following error message:

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

To drop a table permanently, you can write a DROP TABLE command similar to the one shown in **Listing 5**. When you issue the DROP TABLE command with the PURGE option, you can't use the recycle bin to restore a table and its objects, because the table, its contents, and its dependents are permanently removed and the space they occupied is reclaimed. Alternatively, you can purge a dropped table from the recycle bin by issuing the PURGE RECYCLEBIN command as shown in **Listing 6**. You can read about additional commands and options for purging the recycle bin in the Oracle documentation.

Code Listing 5: Using the PURGE option to permanently drop a table, its contents, and its dependents

```
SQL> drop table emp2 PURGE;

Table dropped.

SQL> select object_name, original_name, type
  2    from user_recyclebin;

no rows selected
```

Code Listing 6: Purging a single table and purging the entire recycle bin

```
SQL> purge table emp2;

Table purged.

SQL> purge recyclebin;

Recyclebin purged.
```

The TRUNCATE command removes all data from a table but retains the table's structure along with permissions and certain dependent objects. Like the DROP TABLE command, TRUNCATE does not generate any UNDO information and, when invoked, implicitly issues a COMMIT statement. A TRUNCATE action cannot be flashed back. To recover from an erroneously executed TRUNCATE command, you must either

- Restore (a single table, if so desired) from a *backup* (to be outlined in subsequent articles in this series).
- Flash back the entire database to the point in time just before the TRUNCATE command was issued. Be aware that performing this action causes you to lose all data manipulation language (DML) and DDL actions that were performed after the erroneous TRUNCATE statement was issued.

VIRTUAL GENERATION

Suppose that one of your business requirements is to provide a running calculation of whether your employees are eligible for a cost-of-living wage increase. For example, if the current requirement is that anyone earning a salary value less than

100000 is eligible and anyone earning more than 100000 isn't, you could write a query similar to the one in **Listing 7**. Another way to obtain the same running calculation and simplify any queries that would require this calculation is to create a *virtual column* on the EMPLOYEE table. A virtual column is a column that is usually derived from the other columns of the table but can also be a constant expression or the result of a function. The example in **Listing 8** demonstrates how to add a virtual column to the EMPLOYEE table.

Code Listing 7: Creating a running calculation on the EMPLOYEE table

```
SQL> select first_name||' '||last_name, salary, (case when salary < 100000 then 'Cost of
Living Increase Eligible' else 'No Raise Yet' end) "Wage Increase Worthiness"
  2    from employee
  3  order by first_name||' '||last_name;

FIRST_NAME||''||LAST_NAME                  SALARY Wage Increase Worthiness
_____________________________________ ________ _____________________________________

Betsy James                             60000 Cost of Living Increase Eligible
Donald Newton                           80000 Cost of Living Increase Eligible
Emily Eckhardt                         110000 No Raise Yet
Frances Newton                          82500 Cost of Living Increase Eligible
Lori Dovichi                                  No Raise Yet
Marcy Tamra                                   No Raise Yet
Mary Streicher                         200000 No Raise Yet
Matthew Michaels                        70000 Cost of Living Increase Eligible
Roger Friedli                           60000 Cost of Living Increase Eligible
```

```
Sasha Meyer                                85000 Cost of Living Increase Eligible
Theresa Wong                               70000 Cost of Living Increase Eligible
Thomas Jeffrey                            300000 No Raise Yet
mark leblanc                               65000 Cost of Living Increase Eligible
michael peterson                           90000 Cost of Living Increase Eligible

14 rows selected.
```

Code Listing 8: Defining a VIRTUAL column for EMPLOYEE

```
SQL> alter table employee add (wage_increase_worthiness varchar2(40)
  2                            GENERATED ALWAYS AS
  3                            (case when salary < 100000 then 'Cost of Living
Increase Eligible' else 'No Raise Yet' end) VIRTUAL);

Table altered.
```

You can use the new virtual column to write simpler queries against the EMPLOYEE table that can obtain the value for WAGE_INCREASE_WORTHINESS without needing to include the code for the necessary calculation in the SELECT list each time. The results from the query in **Listing 9** are the same as the results returned in **Listing 7**. The query in **Listing 9** does not need to specify the code for the calculation. Because it specifies the column name, WAGE_INCREASE_WORTHINESS, the derived column value calculations associated with that column are executed and returned. The concatenation of the FIRST_NAME and LAST_NAME columns can also be considered a computation that could be added to

the EMPLOYEE table as another virtual column. Adding such a virtual column to the EMPLOYEE table further simplifies the query in **Listing 9**, as demonstrated in **Listing 10**. Query simplification isn't the only reason to include virtual columns in your tables, as you'll see in subsequent articles in this series.

Code Listing 9: A query selecting from a virtual column

```
SQL> select first_name||' '||last_name "Employee", salary "Salary",
wage_increase_worthiness "Wage Increase Worthiness"
  2    from employee
  3  order by first_name||' '||last_name;

Employee                                 Salary Wage Increase Worthiness
---------------------------------------- ------ ----------------------------------
Betsy James                               60000 Cost of Living Increase Eligible
Donald Newton                             80000 Cost of Living Increase Eligible
Emily Eckhardt                           110000 No Raise Yet
Frances Newton                            82500 Cost of Living Increase Eligible
Lori Dovichi                                    No Raise Yet
Marcy Tamra                                     No Raise Yet
Mary Streicher                           200000 No Raise Yet
Matthew Michaels                          70000 Cost of Living Increase Eligible
Roger Friedli                             60000 Cost of Living Increase Eligible
Sasha Meyer                               85000 Cost of Living Increase Eligible
Theresa Wong                              70000 Cost of Living Increase Eligible
Thomas Jeffrey                           300000 No Raise Yet
```

```
mark leblanc                                    65000 Cost of Living Increase Eligible
michael peterson                                90000 Cost of Living Increase Eligible

14 rows selected.
```

Code Listing 10: The concatenation of FIRST_NAME and LAST_NAME virtualized

```
SQL> alter table employee add (emp_full_name varchar2(70)
  2  GENERATED ALWAYS AS
  3  (first_name||' '||last_name) VIRTUAL);

Table altered.
```

```
SQL> select emp_full_name "Employee", salary "Salary", wage_increase_worthiness "Wage
Increase Worthiness"
  2    from employee
  3  order by emp_full_name;

Employee                                          Salary Wage Increase Worthiness
_____________________________________________ ________ ____________________________________

Betsy James                                        60000 Cost of Living Increase Eligible
Donald Newton                                      80000 Cost of Living Increase Eligible
Emily Eckhardt                                    110000 No Raise Yet
Frances Newton                                     82500 Cost of Living Increase Eligible
Lori Dovichi                                             No Raise Yet
Marcy Tamra                                              No Raise Yet
Mary Streicher                                    200000 No Raise Yet
```

```
Matthew Michaels                          70000 Cost of Living Increase Eligible
Roger Friedli                             60000 Cost of Living Increase Eligible
Sasha Meyer                               85000 Cost of Living Increase Eligible
Theresa Wong                              70000 Cost of Living Increase Eligible
Thomas Jeffrey                           300000 No Raise Yet
mark leblanc                              65000 Cost of Living Increase Eligible
michael peterson                          90000 Cost of Living Increase Eligible

14 rows selected.
```

You need to consider a few limitations and restrictions if you plan to use virtual columns. You cannot update a virtual column, and you can only create a virtual column that is derived from other nonvirtual columns in the table within which it is defined. Also, deriving and storing results physically on disk might be a better strategy if

- Your derived value is computationally complex (and therefore CPU-intensive)
- You invoke the calculation for the derived value often
- You are retrieving many rows at a time

However, in many more cases, a virtual column will give you the same flexibility and performance you obtain from a normal scalar-value column.

CREATING A SEQUENCE OF EVENTS

Recall that each record in an entity must be uniquely identified and separate from every other record in the same entity and that a primary key helps enforce such uniqueness. The most common data for a primary key value is numeric. Although it's possible to keep a running tally of every value used in a primary key column for

an entity, that process is tedious and error-prone, particularly when such a primary key value is referenced by any foreign key values. To generate the type of artificial or *surrogate* key value found in the EMPLOYEE_ID primary key column of the EMPLOYEE table, you can use a *sequence*. A sequence is an Oracle Database object used to generate unique integers.

The example in **Listing 11** outlines the simplest way to create a sequence. **Listing 12** demonstrates how this sequence is used to help create a new record and a new EMPLOYEE_ID value for the EMPLOYEE table. Instead of supplying a literal numeric value, you supply the pseudocolumn, *nextval* (employee_id_seq.nextval), of the EMPLOYEE_ID_SEQ sequence. The sequence automatically generates the next integer value it has available to it and supplies it to the nextval pseudocolumn each time you access it. If you create a sequence without a specific start value, the sequence will automatically begin generating integers, starting at 1.

Code Listing 11: Creating a sequence for the EMPLOYEE table's EMPLOYEE_ID

```
SQL> create sequence employee_id_seq;

Sequence created.
```

Note also that in the INSERT statement in **Listing 12**, the value specified for the virtual column, WAGE_INCREASE_WORTHINESS, is *default*. If you do not explicitly mention this column in your INSERT statement, your virtual column will still generate the default computation assigned to it. As with any other column with a DEFAULT value, you need not explicitly mention it in your DML statements if you want the DEFAULT value generated.

Code Listing 12: Using a sequence to create a record in the EMPLOYEE table

```
SQL> insert into employee (employee_id, first_name, last_name, hire_date, salary,
manager, department_id, wage_increase_worthiness)
  2  values (employee_id_seq.nextval, 'Don', 'Rose', sysdate, 95000, 6576, 40, default);

1 row created.

SQL> commit;

Commit complete.

SQL> select *
  2    from employee
  3   where first_name = 'Don';

EMPLOYEE_ID FIRST_NAME LAST_NAME HIRE_DATE  SALARY  MANAGER DEPARTMENT_ID WAGE_INC...
___________ __________ _________ _________ _______ ________ _____________ ___________

          1 Don        Rose      26-OCT-16   95000     6576            40 Cost of ...

1 row selected.
```

If you intend to use a sequence to generate integers for a particular column, you can simplify your DML statements that create these column values by ensuring that such column values are populated by default. The example in **Listing 13** demonstrates adding a sequence value as a DEFAULT value to the EMPLOYEE table. Alternatively, you can define a column with an IDENTITY clause for a table

column defined with a numeric type. The example in **Listing 14** illustrates one of the syntax options for using IDENTITY-generated values for the EMPLOYEE_ID data as opposed to the EMPLOYEE_ID_SEQ values. You can read more about Oracle sequences in the Oracle documentation. You can find more information about the Oracle IDENTITY clause in the Oracle documentation by searching for the "identity" keyword.

Code Listing 13: Using a sequence as a DEFAULT value for EMPLOYEE_ID

```
SQL> alter table employee modify (employee_id NUMBER DEFAULT employee_id_seq.nextval);

Table altered.

SQL> insert into employee (first_name, last_name, hire_date, salary, manager,
department_id)
  2  values ('Gerald', 'Sowell', sysdate, 100000, 6576, 40);

1 row created.

SQL> commit;

Commit complete.

SQL> select *
  2    from employee
  3   where first_name = 'Gerald';
```

```
EMPLOYEE_ID FIRST_NAME LAST_NAME HIRE_DATE SALARY  MANAGER ..._ID WAGE... EMP_FULL_NAME
___________ __________ _________ _________ _______ _______ ______ _______ _____________

         21 Gerald     Sowell    29-OCT-16  100000    6576 40     No R... Gerald Sowell

1 row selected.
```

Code Listing 14: Using an IDENTITY column for unique integer generation

```
SQL> create table employee_identity as
  2  select * from employee;

Table created.

SQL> alter table employee_identity drop column employee_id;

Table altered.

SQL> alter table employee_identity add (employee_id number
generated always as identity);

Table altered.

SQL> select employee_id, first_name, last_name
  2    from employee_identity
  3  order by employee_id;
```

```
EMPLOYEE_ID FIRST_NAME                     LAST_NAME
___________ _____________________________ _________________________
          1 Marcy                         Tamra
          2 Sasha                         Meyer
          3 Gerald                        Sowell
          4 Emily                         Eckhardt
          5 Frances                       Newton
          6 Donald                        Newton
          7 Matthew                       Michaels
          8 Roger                         Friedli
          9 Betsy                         James
         10 michael                       peterson
         11 mark                          leblanc
         12 Thomas                        Jeffrey
         13 Theresa                       Wong
         14 Lori                          Dovichi
         15 Mary                          Streicher
         16 Don                           Rose

16 rows selected.
```

ANOTHER POINT OF VIEW

Another way to simplify query writing is to create a *view*. A view is a stored SELECT statement whose structure is a virtual table with columns and rows. Its query definition accesses one or more *base tables*, which can be either actual tables or other

views. Consider the example in **Listing 15**. You'll need the appropriate system privilege granted to the SQL_201 user to create a view similar to the one shown in **Listing 15**. The necessary steps and syntax for obtaining this privilege are illustrated in **Listing 15**.

Code Listing 15: Simplifying query creation with views

```
--Need to be a privileged user to grant sql_201 permissions to create a view
SQL> connect / as sysdba
Connected.

--If you are using a pluggable database,
EXAMPLE:  alter session set container = <the container (pluggable database)
of your choosing>;
SQL> alter session set container = dbim;

Session altered.

-Grant the appropriate system privilege to sql_201
SQL> grant create view to sql_201;

Grant succeeded.

--Reconnect as the sql_201 user and proceed with your view creation
SQL> connect sql_201@dbim
Enter password:
Connected.
```

```
SQL> create or replace view emp_manager_overview as
  2  select mgr.first_name||' '||mgr.last_name manager, emp.first_name||' '||emp.last_
name employee, dep.name "Department", emp.wage_increase_worthiness
  3    from employee emp, employee mgr, department dep
  4   where emp.manager       = mgr.employee_id
  5     and emp.department_id = dep.department_id
  6  order by manager, employee;

View created.

SQL> select *
  2    from emp_manager_overview;
```

MANAGER	EMPLOYEEE	Department	WAGE_INCREASE_WORTHINESS
Donald Newton	Frances Newton	Accounting	Cost of Living Increase Eligible
Emily Eckhardt	Betsy James	Accounting	Cost of Living Increase Eligible
Emily Eckhardt	Donald Newton	Accounting	Cost of Living Increase Eligible
Emily Eckhardt	Lori Dovichi	Accounting	No Raise Yet
Emily Eckhardt	Marcy Tamra	Accounting	No Raise Yet
Emily Eckhardt	Matthew Michaels	Accounting	Cost of Living Increase Eligible
Emily Eckhardt	Roger Friedli	Accounting	Cost of Living Increase Eligible
Mary Streicher	Don Rose	Marketing	Cost of Living Increase Eligible
Mary Streicher	Sasha Meyer	Marketing	Cost of Living Increase Eligible
Thomas Jeffrey	Theresa Wong	IT	Cost of Living Increase Eligible
michael peterson	mark leblanc	Payroll	Cost of Living Increase Eligible

```
11 rows selected.
```

The CREATE OR REPLACE keywords can be used to create a view for the first time or to update the definition of an existing view. The view in **Listing 15** is composed of a query that returns a list of employees with their assigned managers and departments and their wage-increase worthiness. Once a view is created, it is not necessary to rewrite the view's query to obtain the same result. As the last query in **Listing 15** demonstrates, a view can be queried just like a regular table. The structure and complexity of the underlying query are contained only in the view and are, therefore, hidden from the user.

Because the data retrieved from a view consists of only those columns listed in the SELECT list of the underlying query, views can be a helpful tool if you want to take secure measures to hide data. Consider the example in **Listing 16**. The EMP_WAGE_INCREASE_IT_VW view is created from the EMP_MANAGER_OVERVIEW view and consists of a small subset of data from its underlying view. The example in **Listing 16** illustrates how columns in view definitions can be given names that differ from those in the base table or the underlying view.

Code Listing 16: Using views for hiding data

```
SQL> create or replace view emp_wage_increase_IT_vw as
  2  select employee "Reviewed Employee", wage_increase_worthiness
"Wage Increase Determination"
  3    from emp_manager_overview
  4   where "Department" = 'IT'
  5  order by employee;
```

```
View created.

SQL> select *
  2    from emp_wage_increase_IT_vw;

Reviewed Employee                  Wage Increase Determination
_________________________________ _____________________________________

Theresa Wong                       Cost of Living Increase Eligible

1 row selected.

SQL> select "Reviewed Employee"
  2    from emp_wage_increase_IT_vw;

Reviewed Employee
_____________________________________________________________________

Theresa Wong

1 row selected.

SQL> select "Reviewed Employee"
  2    from emp_wage_increase_IT_vw
  3   where "Wage Increase Determination" = 'No Raise Yet';

no rows selected
```

Listing 16 also demonstrates how these new column names can be used in WHERE clauses in SQL statements that access the newly created view. Anyone granted access to the new views but not to the underlying base tables has access only to the data and column names returned from the new views. Note that a view can become *invalid* after you alter one of its base tables and might require recompilation. The syntax for recompiling the EMP_MANAGER_OVERVIEW view is

```
ALTER VIEW emp_manager_overview COMPILE;
```

Recompilation of a view should almost never be necessary in practice, however. Oracle Database will automatically recompile the view for you when you attempt to access it and the database detects that changes have been made to the base tables.

Virtual columns and views are similar in terms of their query-simplification and data-hiding capabilities, but virtual columns can also be *indexed*, *constrained*, and used as *partition keys* in a partitioning option (all concepts that will be discussed in subsequent articles in this series). These capabilities might make virtual columns a choice that's superior to views. You can read more about Oracle views in the Oracle documentation.

CONCLUSION

This article has illustrated more about DDL, including how to drop and rename tables and how to recover tables by using the recycle bin. You also learned how purging syntax affects table recovery and the difference between truncating and dropping a table. You discovered how to use virtual columns to help simplify query writing. You also discovered how sequences and IDENTITY columns can be used for generating surrogate key values. Last, you were introduced to views and how

they can assist with query writing and data hiding. The next article in this series will introduce you to indexes.

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

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.

DOWNLOAD the sample script for this article.

ORACLE DATABASE

Configure and Create Resource Manager Plans

By Joel Goodman

Test your knowledge of Oracle Database's Resource Manager in Oracle Database 12*c*.

Oracle Database's Resource Manager feature is used to limit resource utilization and define minimum service levels for CPU and I/O. This column contains sample questions, answers, and explanations about Resource Manager, including functionality available in Oracle Database 12*c*. The sample questions are of the type you may encounter when taking the "Oracle Database 12*c*: Installation and Administration" (1Z0-062), "Upgrade to Oracle Database 12*c*" (1Z0-060), or "Oracle Database 12*c*: Performance Management and Tuning" (1Z0-064) exam.

CONFIGURE RESOURCE MANAGER

Resource Manager may be configured from the command line or with GUI tools. For groups of database sessions known as "resource consumer groups," Resource

Manager supports the use of limits on certain resources, and it also can define minimum resource availability for CPU and I/O bandwidth. Resource Manager allocates resources to consumer groups according to the set of resource plan directives for the currently active resource plan.

Oracle Database 12*c* supports all the features of Resource Manager from earlier database releases as well as the new Oracle Multitenant architecture introduced in Oracle Database 12*c*.

TEST YOUR NEW DBMS_SQL KNOWLEDGE

SEE ANSWERS BELOW

QUESTION 1

For which three purposes would you use Resource Manager?

A. Limiting the CPU used per consumer group session
B. Specifying the maximum number of concurrent sessions that may have a call executing
C. Specifying the amount of memory a session can allocate from the large pool
D. Limiting the degree of parallelism of operations performed by a consumer group
E. Specifying an idle-time limit that applies to idle sessions that are not blocking other sessions

---

ANSWERS

A | CORRECT

The MAX_EST_EXEC_TIME attribute specifies the maximum CPU time permitted

for a database session in the specified consumer group before an ORA-07455 error is issued.

B | CORRECT
The ACTIVE_SESS_POOL_P1 attribute limits the number of database sessions in the specified consumer group that may simultaneously have an active call.

C | INCORRECT
Resource Manager may be used to limit I/O or CPU but currently does not limit memory utilization in the Program Global Area (PGA) or the System Global Area (SGA). Oracle Database's automatic memory management or automatic SGA memory management capabilities, together with automatic PGA memory management, can control memory use adaptively, but these do not depend on Resource Manager.

D | INCORRECT
The PARALLEL_DEGREE_LIMIT_P1 attribute limits the degree of parallelism for any operation by any database session, but it does not limit the aggregate degree of parallelism for an entire consumer group.

E | CORRECT
The MAX_IDLE_TIME limits how long a session may be idle when not blocking another database session for any reason, before the session is killed by the process monitor.

QUESTION 2

Which three statements are true about Resource Manager?

A. It can take action for I/O-intensive sessions only after a specified number of physical I/Os is reached.
B. It can limit the amount of undo used by a consumer group.
C. A resource plan must always have a plan directive for a "catchall" group called OTHER_GROUPS.
D. A resource plan change can be done automatically by Oracle Database.
E. It can automate the switching of a consumer group for a session but cannot switch the session back to its original consumer group.
F. It can limit the amount of undo used by each session in a consumer group.

ANSWERS

A | INCORRECT
The SWITCH_IO_LOGICAL attribute allows an action to be taken when a database session reaches the defined limit of logical I/Os. Actions are specified in the SWITCH_GROUP attribute.

B | CORRECT
The UNDO_POOL attribute limits the undo used by a consumer group as a whole.

C | CORRECT
If a resource plan lacks a directive for the OTHER_GROUPS group, the plan will not compile successfully.

D | CORRECT
The current resource plan for a database can be changed with the Oracle Scheduler feature of Oracle Database. When a new scheduler window opens, the resource plan associated with that window is automatically activated.

E | INCORRECT
The SWITCH_FOR_CALL attribute controls whether automatic switching of consumer groups due to the number of physical I/Os, I/O megabytes, logical I/Os, or the elapsed time is done only for the current call made by a database session or for the session. If the SWITCH_FOR_CALL attribute value is TRUE, then the session is automatically switched back after the call ends.

F | INCORRECT
The UNDO_POOL attribute limits the undo used by a consumer group as a whole—not the undo used by each session in that consumer group. If there is only one session in the consumer group, then the limit applies to that database session, but if there are multiple sessions, then the limit applies to the consumer group and individual sessions may have to wait with far less undo use than the limit defined by the UNDO_POOL attribute.

CREATE RESOURCE PLANS FOR MULTITENANT DATABASES

Oracle Database 12*c* supports Resource Manager plans for pluggable databases

(PDBs) in an Oracle Multitenant environment in addition to the plans it supports when Oracle Multitenant is not used.

QUESTION 3

Which three resources might be prioritized between competing PDBs when a multitenant container database (CDB) plan is created with Resource Manager?

- **A** Maximum number of logical I/Os done by a PDB
- **B** Maximum number of physical I/Os done by a PDB
- **C** Parallel servers per PDB
- **D** CPU used by a PDB
- **E** Maximum number of PDBs for a CDB

ANSWERS

A | INCORRECT

The DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE procedure has no attributes to limit the number of logical I/Os done by a PDB. A PDB resource plan may be used to limit logical I/Os for each consumer group, but there is no way to limit the overall number of logical I/Os for the PDB as a whole.

B | CORRECT

The DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE procedure can use MAX_IOPS to limit the number of physical I/Os done by a PDB, or, beginning

in Oracle Database 12*c* Release 2, the procedure can use MAX_MBPS to limit the amount of data for which I/O is performed by a PDB.

C | CORRECT
The DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE procedure has a PARALLEL_SERVER_LIMIT attribute that can restrict the maximum number of parallel execution server processes the PDB can use.

D | CORRECT
The DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE procedure has a UTILIZATION_LIMIT attribute that can restrict the CPU utilization for a PDB.

E | INCORRECT
The DBMS_RESOURCE_MANAGER.CREATE_CDB_PLAN_DIRECTIVE procedure has no attributes to limit the number of PDBs in the CDB. There is a limit of 252 administrator-created PDBs per CDB in Oracle Database 12*c* Release 1 and a limit of 4,094 administrator-created PDBs per CDB in Oracle Database 12*c* Release 2, but those limits are not imposed by Resource Manager.

QUESTION 4

Which two statements are true about Resource Manager PDB resource plans in a multitenant CDBs?

A. If no PDB plan is enabled for a specific PDB, then all sessions for that PDB get resources prioritized by length of session.

B To enable a PDB plan, a CDB plan must be created and enabled.
C If a PDB plan is enabled for at least one PDB, then the CDB plan may not be disabled.
D A PDB plan may be disabled.
E A CDB plan may be enabled even if no PDBs have enabled PDB plans.

ANSWERS

A | INCORRECT
If no PDB resource plan is specified for a PDB, then the PDB is managed according to the default PDB plan directive, provided that a CDB resource plan is enabled.

B | INCORRECT
It is possible to have PDB resource plans with or without a CDB resource plan, but without a CDB plan, some PDB directives are disabled.

C | INCORRECT
It is possible to disable a CDB resource plan even when there are PDB resource plans, but some of the PDB plan directives will become disabled.

D | CORRECT
If a PDB resource plan is disabled, then the PDB is managed with the default PDB plan directive in the CDB resource plan, provided that the CDB plan is enabled.

E | CORRECT
If no PDB has an enabled PDB resource plan, then all PDBs are managed with the default PDB plan directive, provided that the CDB resource plan is enabled.

Joel Goodman is the global technical lead for Oracle Database Certification Exam development, a senior principal trainer for Oracle University EMEA, and a member of the OAK Table. He has been with the company since 1997.

NEXT STEPS

LEARN more about the Oracle Certification Program.

READ *Oracle Database Administrator's Guide*, "Managing Resources with Oracle Database Resource Manager."

READ *Oracle Database PL/SQL Packages and Types Reference*, "DBMS_RESOURCE_MANAGER."

EXPLORE Inside OCP columns.

ORACLE CODE

Register Now

New One-Day, Free Event | 20 Cities Globally

Explore the Latest Developer Trends:

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

Live for the Code

Find an event near you:

developer.oracle.com/code

ORACLE®