ORACLE MAGAZINE

SEPTEMBER/OCTOBER 2017

BREAKTHROUGH INFRASTRUCTURE

Why next-gen Oracle Cloud Infrastructure will transform your IT and business

CLOUD COUTURE

INTERVIEW> MEET THE NEW INFRASTRUCTURE

MOBILE> SURVIVING THE MOBILE DATA OFFLINE BLUES

ORACLE

The Secret of Powerful Performance is in Fujitsu SPARC M12

FUJITSU

Fujitsu SPARC M12: POWERFUL
SECURE
SCALABLE

Fujitsu is a proud sponsor of
the Oracle OpenWorld 2017 in San Francisco.

Details

shaping tomorrow with you

CONTENTS

ORACLE MAGAZINE

26 Breakthrough Infrastructure

39
Cloud Couture

FEATURES

9 MashUp

CONTENTS

COMMUNITY

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, 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 and Audience Development Director Karin Kinnear

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.

Data Quality Made Easy. Your Data, Your Way.

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

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

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

View Video

ORACLE® PEOPLESOFT

ORACLE® JD EDWARDS

ORACLE® E-BUSINESS SUITE

Java™

ORACLE® Forms

ORACLE PL/SQL

Start Your Free Trial www.melissa.com/oracle

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

1-800-MELISSA

melissa™
GLOBAL INTELLIGENCE

Tom Haunert

Lift, Shift, and Next

Next-generation Oracle Cloud Infrastructure is about more than lifting and shifting today's IT.

Not long ago, businesses squeezed more value out of their on-premises infrastructure investments by using technologies such as virtualization to tap underutilized capacity. With virtualization, they traded away a little bit of system performance for a tremendous increase in infrastructure utilization.

But within just a few years, public cloud infrastructure services became a cost-effective *and* hardware-refresh-free way to replace data center hardware—networking, compute, and storage. Businesses lifted and shifted their IT from their on-premises virtual machines (VMs) to VM-based public cloud infrastructure services.

STRIPPED DOWN

Businesses continue to lift and shift to the cloud, but they are also looking to a future that includes cloud-native development projects and operations. And there's good news for these future-focused businesses, because the "future" cloud infrastructure that simplifies application development and deployment and improves platform performance is here *today*.

Next-generation Oracle Cloud Infrastructure bare metal services already deliver higher-performance networking than first-generation infrastructure services and support for lightweight containers for developing, deploying, and

running a range of platform services. And containers deliver those operations without the virtualization-overhead penalties of first-generation infrastructure-as-a-service offerings.

In this issue's cover story, "Breakthrough Infrastructure," three organizations discuss how they are using first- and next-generation Oracle Cloud Infrastructure services. These customers and partners share the experience of using Oracle Cloud Infrastructure for high-performance computing, to free time and investments for new projects, to overcome regional infrastructure challenges, and more.

In this issue's interview, "Meet the New Infrastructure," Marc Levy, an architect and vice president of software development at Oracle, describes how businesses are looking for cloud infrastructure services that can deliver what they are getting today from their own data centers *and* cloud-native capabilities. Levy points to Oracle Cloud Infrastructure as a kind of rethink of the public cloud and a solution for businesses looking for a full stack of cloud technology—from infrastructure to platform to applications.

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

READ more about Oracle Cloud Infrastructure.

READ more about Oracle Exadata Cloud on Oracle Cloud Infrastructure.

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

Oracle Platinum Partner

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

Oracle Platinum Partner
Specialized
Oracle Database 12*c*

Oracle Platinum Partner
Cloud Standard

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

Oracle Platinum Partner
Specialized
Oracle WebLogic Server 12*c*

Oracle Platinum Partner
Specialized
Service-Oriented Architecture

Oracle Platinum Partner
Specialized
Oracle Exadata Database Machine

Cybernoor Corporation I www.cybernoor.com I +1.925.924.0400

cybernoor™

BY PIERCE ASHWORTH

Back to School

Control time, find the lost, and trust AI.

A Smartwatch, Powered by You

Use your own body heat to electrically power this watch. The human body emits heat, and as you exercise your skin gets warmer. This thermo-electrically sustainable energy powers the watch, its activity and sleep-tracking capacity, and a multitude of supported microapplications. The Matrix PowerWatch is available for preorder at US$159; it ships in September 2017. matrixindustries.com

Find Misplaced Items with Your Smartphone

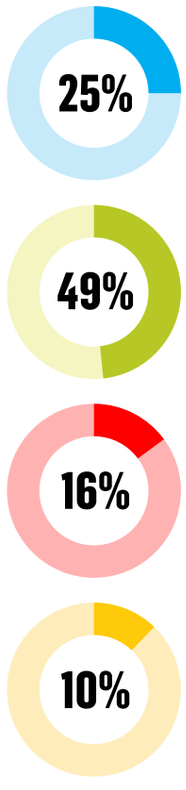
It's the solution to an age-old problem: the TrackR pixel is a Bluetooth tracker that connects to an item (wallet, keys, remote control) and lights up and rings when you need to find it. Use your smartphone to signal the tracker, or press the pixel button to ring your phone loudly—even when it's on silent. A single pixel costs US$25. thetrackr.com/pixel

PHOTOGRAPHS COURTESY OF MATRIX INDUSTRIES, TRACKR

Adoption of Artificial Intelligence

Would you consider an AI-powered tutor for your child?

For most young parents, the answer is yes. 74 percent of millennial parents surveyed said they would consider an AI-powered tutor for their child, while only 10 percent were unwilling to entertain the possibility.



Source: IEEE

DO YOU SPEAK TECH? QUIZ YOURSELF!

1. Who is credited with coining the term *artificial intelligence* in 1955 and is considered one of the early pioneers in the field?

A. L. Ron Hubbard
B. John McCarthy
C. Isaac Asimov
D. Alan Turing

2. What was the trembling name of the first general-purpose mobile robot that employed AI and was developed by the Artificial Intelligence Center of Stanford Research Institute in the late 1960s?

A. Nervous Nelly
B. Unimate
C. PUMA
D. Shakey the Robot

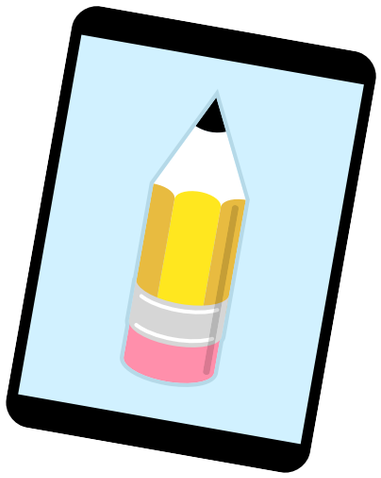
3. Which Ivy League university held the first artificial intelligence conference in 1956?

A. Dartmouth
B. Harvard
C. Princeton
D. Brown

Answers: 1. B: John McCarthy; 2. D: Shakey the Robot; 3. A: Dartmouth

APPS: THE NEW SCHOOL SUPPLIES

With these handy back-to-school apps you can organize your ideas, efficiently scan your projects and documents, and get some help with those tricky math problems.



SimpleMind

Record, arrange, and develop your ideas with mind maps—diagrams for organizing and connecting information. With customizable layouts, media-adding capacity, and cross-device syncing, SimpleMind allows you to relate and more deeply understand concepts by mind-mapping them into an accessible format. Whether you're a visual thinker or just like staying organized, mind maps offer a creative way to manage your work this fall.

Free (Android, iOS)

Office Lens

Scan and share documents efficiently. Office Lens trims, enhances, and makes pictures of whiteboards and documents easily readable so that crumpled and faded receipt you need and your group project notes are alive, well, and readable in your camera roll. Images can also be converted into editable text-based files.

Free (Android, iOS)

Photomath: Camera Calculator

Scan a handwritten math problem for an instant result—Photomath can even give you step-by-step instructions on how to solve the problem. And once scanned and input, problems can be edited with the app's smart calculator, which makes for a versatile and effective aid for taking on tough problem sets or crunching numbers.

Free (Android, iOS)

ART BY WES ROWELL

Oracle OpenWorld 2017

SAN FRANCISCO, CALIFORNIA, OCTOBER 1–5

Oracle OpenWorld 2017 wants you to participate, learn, and take it home with you.

Get a year's worth of professional development in five days at Oracle OpenWorld 2017 in San Francisco, California. With more than 2,500 sessions and completely reimagined exhibition halls designed to foster learning and community, Oracle OpenWorld 2017 is focused on a better experience at the conference and a better information experience afterward.

New session formats—including flipped sessions (two-part sessions that give you a chance to apply our content to your real-world challenges), fireside chats with specialists, brain snacks (one-on-one conversations with fellow experts), and problem-solving—encourage active participation and collaboration among attendees. A new format for exhibition halls, called "The Exchange," is designed to facilitate gathering and exchanging ideas and provide networking opportunities.

These new formats are the result of collaboration between Oracle and the Stanford Center for Design Research, which studies how people best interact and learn. Research shows that attendees retain up to 80 percent of information two weeks after receiving it if they are actively involved in the learning process, compared to retaining just 5 percent if they receive information passively.

So, join tens of thousands of customers and partners and get active. Collaborate, network, get "hands on" with technology, and learn at Oracle OpenWorld 2017.

Get more information and register now.

PHOTOGRAPH BY HARTMANN STUDIOS

Marc Levy, vice president and architect of software development at Oracle, describes Oracle Cloud Infrastructure—Oracle's IaaS offering—as a sort of rethink of the public cloud.

Meet the New Infrastructure

Next-generation infrastructure as a service enables better clouds. **BY TOM HAUNERT**

Infrastructure-as-a-service (IaaS) offerings may have started out as a way to replace data center hardware, but next-generation IaaS technology is focused on more than replacing on-premises networking, compute, and storage with cloud services. Today, cloud vendors and businesses are looking at how IaaS can better support IT and better enable the businesses that run infrastructure services. *Oracle Magazine* caught up with Marc Levy, vice president and architect of software development at Oracle, to talk about the state of IaaS technology and what businesses are looking for in IaaS today.

Oracle Magazine: Just a couple of years ago, IaaS technology was focused on networking, compute, and storage services. What is the state of IaaS today?

Levy: IaaS is evolving, but it still enables customers to build their own virtual networks—their own clouds—on a foundation of networking, compute, and storage technologies. This also describes the first generation of IaaS.

Oracle Cloud Infrastructure—Oracle's IaaS offering—is a sort of rethink of the public cloud. This next-generation technology continues to focus on the key characteristics associated with cloud, including scalability, elasticity, and low-friction provisioning, but it adds a focus on flexibility, performance, governance and control, integration, and more. Oracle Cloud Infrastructure still enables businesses to build their own clouds, but it's also enabling those businesses to migrate to the cloud in more powerful and flexible ways.

> "Businesses need an easy on-ramp to the cloud that allows them to move their existing on-premises applications and systems."

Oracle Cloud Infrastructure offers higher-performance networking, for example, which allows businesses to provision instances of Oracle Exadata on demand and on their own virtual network. So, they can migrate high-performance database-centered applications essentially as-is.

In addition, PaaS [platform-as-a-service] services leverage the unique capabilities of Oracle Cloud Infrastructure. For example, a container service running on a bare metal machine avoids

the virtualization overhead penalties associated with first-generation IaaS offerings.

Oracle Magazine: What do businesses look for or demand from IaaS services today?

Levy: Businesses have a variety of requirements for IaaS. For example, some are looking to lift and shift technology out of their data centers and into the cloud. They are also looking to build new cloud-native platforms and applications based on new infrastructure and platform technologies.

Businesses need an easy on-ramp to the cloud that allows them to move their existing on-premises applications and systems largely as-is. This includes an infrastructure capable of delivering the functionality, availability, and performance of the systems they have today in their data centers. But they also want cloud-native capabilities so they can adopt new architectures and development practices. And they want all of this in the same, integrated environment.

When businesses started looking at cloud a couple of years ago, some were looking for a single SaaS [software-as-a-service] application or offsite backup of their data center or storage and compute power for short-term projects or even a lift-and-shift "test." But they are looking for more now, including a full stack of cloud technology—from IaaS to PaaS to SaaS—from one cloud vendor. Oracle is well equipped to deliver that stack and have that stack serve as a foundation that enables businesses to build, integrate, and optimize whatever they need.

PHOTOGRAPHY BY
RON WURZER/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Cloud Infrastructure.

BY STEPHEN CHIN, DIRECTOR OF ORACLE TECHNOLOGY NETWORK COMMUNITY MANAGEMENT

Recognize

The new Oracle Developer Champion program recognizes modern expert developers who blog, write articles, and present on topics including containers, microservices, SQL, NoSQL, open source technologies, machine learning, and chatbots. Learn more about Oracle advocacy programs and follow the Oracle Developer Champions on the Oracle Developers Blog.

CHANNEL YOUR INFORMATION

Watch the Java Channel on YouTube to stay informed on hot tech topics. In this presentation, Nashorn tech lead Jim Laskey describes the trials and tribulations of integrating V8 JavaScript as a sister javax.script to Nashorn.

Any Questions?

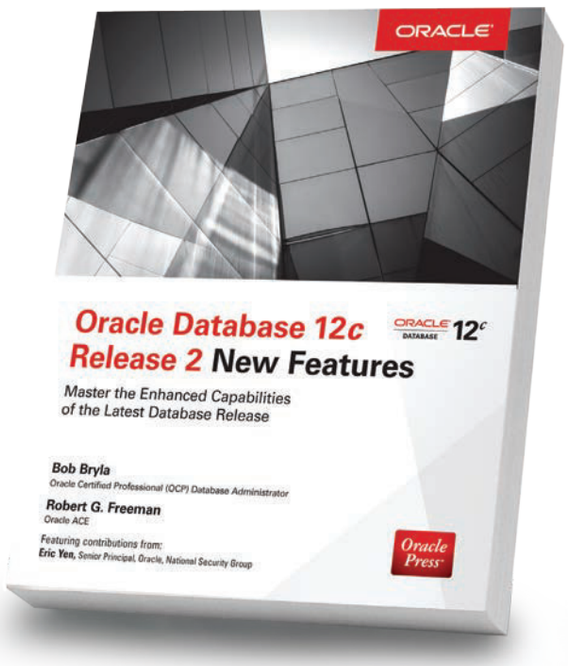
Oracle sponsors Developer Legend AMA (Ask Me Anything) sessions on Reddit with developers worldwide discussing the latest tech trends and sharing industry knowledge. Check out this session with microservices tech guru Chris Richardson, creator of the original Cloud Foundry, author of *POJOs in Action*, and keynote speaker for Oracle Code New York.

Oracle Press™

Your Destination for Oracle and Java Expertise

Visit the Oracle Store in the Moscone West lobby at Oracle OpenWorld and JavaOne 2017 to see some of our latest releases.

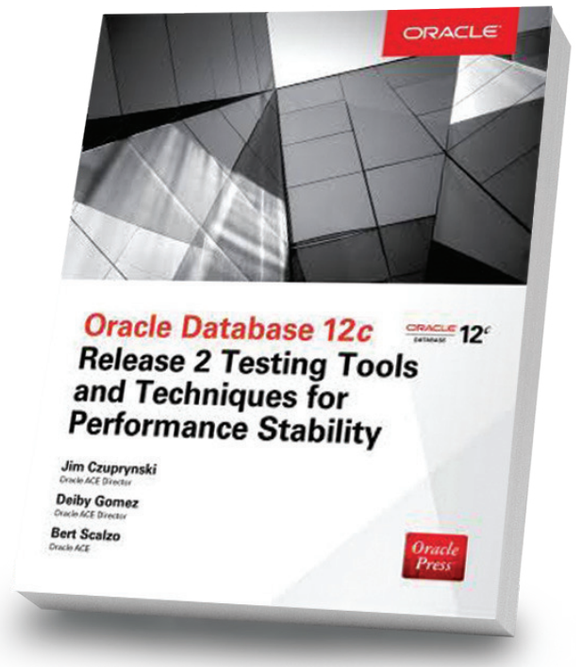
Purchase two or more Oracle Press books, to receive a free Oracle Press teddy bear!



Oracle Database 12c Release 2 New Features

Bob Bryla, Robert G. Freeman

Master the new and enhanced capabilities of the latest Oracle database release.



Oracle Database 12c Release 2 Testing Tools and Techniques for Performance and Scalability

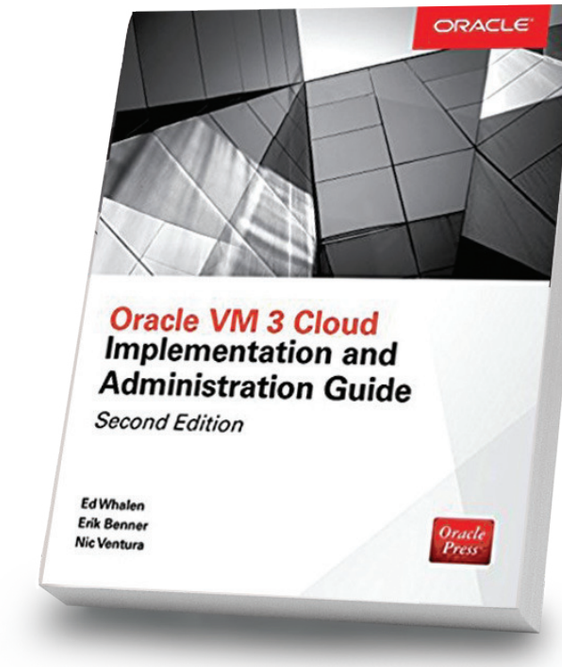
Jim Czuprynski, Deiby Gomez, Bert Scalzo

Seamlessly transition to Oracle Database 12*c* Release 2 and achieve peak performance using best practices in this step-by-step guide. Database SQL Exam.

Oracle Mobile Cloud Service Developer's Guide

John Ray Thomas

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



Oracle VM 3 Cloud Implementation and Administration Guide, Second Edition

Ed Whalen, Erik Benner, Nic Ventura

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

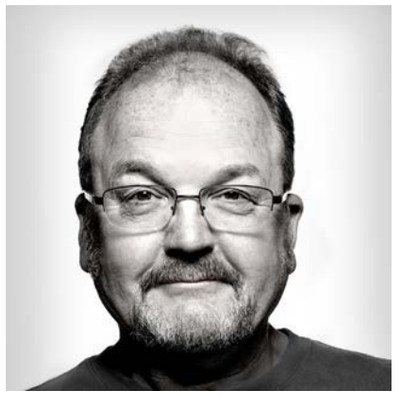
Available in print and ebook formats.

Oracle Press™

www.OraclePressBooks.com • @OraclePress

First Steps

When kids code, careers grow.



By **Bob Rhubart**

In casual conversations with the small army of software developers I've met over the last 20 years, I've learned at least two things. One is that software development is as much a lifestyle as it is a career choice. The other thing is that, probably because of the first thing, the coding bug often bites people long before they need to earn a paycheck.

But the details of those casual conversations can be a bit fuzzy. So in the interest of more-accurate and up-to-date information, I asked community members to share their stories about their earliest interest in writing code and how it developed into a career.

Oracle ACE Rolando Carrasco, now co-owner and SOA principal architect at S&P Solutions, says his interest in software has its roots in his passion for the Intellivision game console and in the 1982 film *Tron*. In elementary school he learned Logo, an educational programming language. "I had the feeling that I could do more with it, and by the time I entered junior high I was eager to have my programming classes." It was in those classes in Visual Basic that Carrasco realized he'd found what he wanted to do for a living.

Then it was off to college, where he learned Pascal, C, Prolog, and Java. "I wanted to program as much as I could," Carrasco says. "When I had the chance to look for a part-time job, I looked for one related to programming." The rest

When did your software development journey begin? Where did it lead?

is an impressive LinkedIn profile.

Oracle ACE Associate Maarten Smeets, a senior integration consultant with AMIS, was 8 years old when he began typing BASIC programs out of a book on an MSX2 machine, making small changes in the code to see what would happen. At age 13, he began experimenting with Linux and C/C++. He entered university at 18, majoring in neuroscience and ethology, and worked internships at a molecular genetics department and at a theoretical biology department, where he leveraged his coding skills on statistics and on simulating fish school behavior.

"At the time, I considered IT a tool to achieve a goal, not a goal on its own," Smeets explains. He chose not to study informatics. "Biology is quite diverse, from being in nature observing birds to researching diseases in hospitals." At the time, that held more appeal for Smeets than a desk job.

But a part-time job handling all the IT tasks at a small insurance company shifted his thinking. "I realized I could not get myself motivated by doing research alone, and I did not like the lab work," he explains. He discovered he enjoyed the more concrete IT challenges at the insurance company.

After graduation Smeets took a job at an IT consulting company, where he met many experienced people, first worked with Oracle technologies, and began accumulating experience in integration, "an area that provides a lot of diversity," he says.

At 15 years old, Martien van den Akker, an Oracle Fusion Middleware and platform-as-a-service specialist with Darwin IT Professionals, says he was drawn to computers and "the mystical idea that you can have a machine do what you want it to do in a creative way." His interest was driven by the availability of home computers. "My cousins had a ZX Spectrum, many school friends had Commodore 64s, and one uncle had a Goldstar MSX1," Akker says. He ended up with a Toshiba MSX 1, which he later

replaced with a Sony MSX2.

His first programs were written in BASIC, "mainly to learn how to do certain things, like drawing Sprite graphics on the screen," Akker says.

In college Akker majored in technical physics and also studied computer science, learning Turbo Pascal, C, and Assembly.

After a stint in the military, Akker worked as a Cobol programmer and systems analyst. "But after a few years, I got bored with COBOL and wanted to learn something new." Oracle products figured prominently in that something new, including PL/SQL, Oracle Forms, Oracle Designer, Oracle Streams AQ, Oracle Workflow, Oracle InterConnect, and BPEL. "I'm now into Oracle Fusion Middleware, including Oracle SOA Suite, Oracle BPM Suite [Oracle Business Process Management Suite], Oracle Service Bus, Oracle WebLogic, and Oracle's integration and process cloud services." But despite the expertise and experience accumulated over 30 years, "I still enjoy the mystics of having machines do what you want them to do," Akker says.

When did your software development journey begin? Where did it lead? Share your story here.

Bob Rhubart is the manager of the architect community on Oracle Technology Network, the host of the Oracle Technology Network ArchBeat *podcast series, producer of the* 2 Minute Tech Tip *video series, and a contributor to the OTN blog.*

PHOTOGRAPHY BY
MICHAEL MCELROY/THE VERBATIM AGENCY

NEXT STEPS

LISTEN to "Trajectories: Career Paths of IT Stars."

WATCH *Speed Up Your IT Career*.

WATCH *Code for Kids*.

Cloud, IoT, and the DBA

Cloud-enabled IoT is a growing force in enterprise business strategies. That's a big opportunity for DBAs.

The Internet of Things (IoT) is about a lot more than connecting toasters to the internet. For the enterprise, it's about drawing information out of business operations and equipment to uncover more-efficient ways to do work. DBAs will be in the middle of the information tsunami, making data flow from sensors and devices to Oracle Cloud services for IoT management, analytics, and more.

Oracle Magazine spoke to attendees at Kscope17, the annual meeting of ODTUG, about the intersection of the cloud, the IoT, and the DBA and database developer.

Jerry Ward

Database Developer at Viscosity

"Being able to coalesce streaming data, process code for streams, eliminate hot spots on specific databases, and federate it all out through the cloud worldwide—these tasks are all in the realm of the DBA now. I think it's more exciting than ever for the DBA."

Jon Dixon

Oracle Tech Lead at JMJ Cloud

"Oracle Exadata Cloud Service has a fantastic tool in ORDS [Oracle REST Data Services] that exposes REST services, which are what Alexa needs to execute. Because DBAs have ORDS and REST knowledge, it's a relatively easy process for them to learn the integration."

Christian Screen

Global Analytics Practice Director at Datavail

"The [IoT hands-on session at Kscope17] had Arduino boards sending info up to Oracle IoT Cloud Service and then back down to get responses from peripherals. This hands-on work relates to what we do. Analytics and data and IoT are all coming together."

BY BLAIR CAMPBELL

Learning the “Why”

Three peers advocate detailed goals, up-to-date tech, and prudent debt management.

Gary Gordhamer

Milwaukee, Wisconsin

ORACLE ACE Associate

Company: GE Digital

Job title: Principal architect

Length of time using Oracle products: 25 years

What advice do you have for those just getting into application development? Get to know the details. One of my favorite quotes is from *Star Trek II: The Wrath of Khan*. Captain Kirk says to Lieutenant Saavik, “You’ve got to learn *why* things work on a starship.” In my career, that message has time and time again allowed me to deal with increasingly complex situations, designs, and systems. Don’t just know *what* to do, but know why you’re doing it.

What’s the most common cause you see when IT projects go wrong? Lack of planning and lack of detailed goals. There’s a huge move to be more agile and iterative and deliver fast. My main comment most the time is, “Slow down to go fast.” Having a good plan and executing it quickly is way more successful than having no plan and running fast.

What’s your favorite thing to do that doesn’t involve work? I have way too many hobbies. I really enjoy brewing and tasting beer, downhill skiing, and woodworking, and I’ve recently started playing with Raspberry PI and Arduino to make a smoker controller for my BBQ needs as well as an automated temperature controller for beer fermentation. Above all that is helping my kids with Cub Scouts and Girl Scouts and running the LEGO Robotics club at their elementary school.

Tobias Arnhold

Heppenheim, Germany

ORACLE ACE Associate

Company: Tobias Arnhold–IT Consulting
Job title: Consultant
Oracle credentials: Oracle Forms Developer Certified Professional, Oracle Database SQL Certified Associate
Length of time using Oracle products: 11 years

How are you using social media in your work these days? I'm using Twitter to stay in touch with other specialists and to get the latest news about Oracle technologies. To stay connected to my local colleagues here in Germany, I'm using meetup.com, where I'm a co-organizer of the Frankfurt Oracle Application Express meetup group. To share my knowledge, I blog on APEX-at-Work, and sometimes I help at the Oracle Application Express discussion forum.

What's your go-to Oracle reference book? When I was a DBA, my life was saved several times by the book *Oracle Database 10g RMAN Backup & Recovery* [Oracle Press, 2006]. It helped me recover several database crashes by recovering the data with the Oracle Recovery Manager [Oracle RMAN] utility. At the moment when the database needs to be recovered, your pulse goes up to 180. You need a guide you can trust, and this book showed me the way to handle such challenges. These days, I mostly read blogs and Stack Overflow.

What advice do you have for those just getting into application development? Work with Oracle Application Express. It delivers the same or even better development speed than Excel or Access. On the other side, the data and the business logic are saved in a high-availability Oracle Database environment. In addition to speed, security, and high availability, you also have access to all-new web features and can include them quite easily in your applications.

Scott Spendolini

Ashburn, Virginia

ORACLE ACE Director

Company: Sumner Technologies
Job title: President and founder
Oracle credentials: Oracle Application Express Developer Certified Expert
Length of time using Oracle products: 21 years

How did you get started in IT? One day long ago, my father brought home an Atari 400 computer. I was fascinated by it—particularly with the BASIC programming language—and spent any and all free time learning how to program with it. Fast forward a few years, and I took a summer job at Aetna Health Plans, where I used Visual Basic and FoxPro to build some of my first database applications. I loved the challenge and the ability to work with both the business side and the technical side to craft solutions, and then see those solutions in action.

What's the next big thing driving change in your industry? I'd say total cost of code ownership is starting to change the way that we think about building new solutions. I've also seen this called *technical debt*. Like your personal finances, debt is a fact of life; it's how that debt is managed that makes it either good or bad. And in most cases, application costs aren't even close to realized during the design and development phase. Requirements change, platforms get swapped out, and even methods of delivery evolve. Applications that are well thought out and built on a solid foundation will weather these changes better.

How are you using cloud computing in your work these days? From my company's public website to its back-office applications, everything runs on Oracle Application Express in Oracle Database Cloud Service. This was a no-brainer for a company that provides Oracle Application Express services and training, because there's no better customer use case than your own.

ORACLE
OPEN
WORLD

Oct. 1–5, 2017 | San Francisco | #OOW17

REGISTER NOW

Save $200 by Sept. 29*

oracle.com/openworld

Explore, Learn, Connect, Inspire

ORACLE®

Global Sponsor
Deloitte.

Innovation Sponsor
intel

Premier Sponsors
pwc
TATA CONSULTANCY SERVICES

Grande Sponsors
accenture
DXC.technology
FUJITSU
IBM
wipro

*Discount based on the onsite registration price.

BREAKTHROUGH INFRASTRUCTURE

Why next-gen Oracle Cloud Infrastructure will transform your IT and business

BY DAVID BAUM

Infrastructure as a service (IaaS) got its start more than 10 years ago, as service providers made extra computing, storage, and network capacity available to customers on a subscription basis. These remote-technology offerings gave organizations an alternative to provisioning, installing, and maintaining their own IT infrastructure and paved the way for individuals, companies, and governments to tap into unlimited data center assets.

Those first-generation cloud offerings have become extremely popular—but today's customers want more. More power. More versatility. More control.

Oracle has responded with next-generation IaaS offerings that combine the elasticity and utility of the public cloud with the granular control, security, and predictability of on-premises infrastructure.

"One of the key ideas of Oracle Cloud Infrastructure—initially introduced as Oracle Bare Metal Cloud Services—is to meet customers where they are," says Marc Levy, an architect and vice president of software development at Oracle. "If they want to move their applications as they are, they can do that. And they can do that because Oracle Cloud Infrastructure supports a wide variety of compute—from virtual machines to bare metal servers to engineered systems such as Oracle Exadata—all in the same cloud infrastructure. So customers can use the familiar infrastructure patterns they use on premises—for example, a very high-performance engineered system—and they can also adopt cloud-native patterns."

Building on Raw Power

This type of flexibility and power is perfect for companies such as Zenotech, an Oracle partner that supplies cloud-based solutions for computational fluid dynamics. Zenotech helps engineers at aerospace companies, automotive companies, and civil engineering firms simulate airflow over airplane wings, airflow around buildings, and similar types of complex engineering challenges. Its software platform and cloud brokerage service, known as Elastic Private

ZENOTECH
Bristol, UK

REVENUE:
£500,000

ORACLE PRODUCTS:
Oracle Cloud Infrastructure, including Oracle Cloud Infrastructure Compute

Zenotech Director and Cofounder David Standingford (left), shown here with Director and Cofounder Jamil Appa, believes on-demand cloud services such as Oracle Cloud Infrastructure Compute are ideal for powering the kinds of resource-intensive tasks Zenotech customers require.

Interactive Cloud (EPIC), enables customers to initiate and scale these complex jobs within a high-performance computing environment. Today, many of those jobs run on Oracle Cloud Infrastructure services.

"We invested in Oracle Cloud Infrastructure because it includes some of the latest hardware available anywhere in the world," says David Standingford, director and cofounder at Zenotech. "By linking directly with Oracle infrastructure, our EPIC platform, which is designed to match elements in complex, heterogeneous

> **What we really like about the bare metal offering from Oracle is that there is very little technology between us and the hardware.**
>
> —**David Standingford**, Director and Cofounder, Zenotech

workflows with the optimal hardware, will have unmatched performance."

Zenotech powers these engineering simulation jobs using infrastructure from Amazon Web Services, the University of Cambridge, and the Bristol, UK–based Centre for Modelling and Simulation, among a growing list of providers.

But the company is always looking for additional online resources to scale up during sudden bursts of activity—particularly where new hardware offers distinctive attributes to deliver a performance advantage or where customers have special platform requirements that are difficult or impossible to deliver in a shared environment, including high-security, root access to operating systems and hypervisors and hardware-level access.

This is where Oracle Cloud Infrastructure Compute, an enterprise-grade infrastructure service that provides a rapidly provisioned virtual compute environment, really delivered, according to Zenotech.

"We were looking for solutions to deliver our engineering capabilities without incurring the overhead of buying and maintaining the latest hardware," says Standingford. "What we really like about the bare metal offering from Oracle is that there is very little technology between us and the hardware."

Zenotech is currently flexibly using up to 30 standard Oracle Cloud Infrastructure Compute instances, each with 36 cores, 256 GB of RAM, and local solid-state drive (SSD) block storage. "It's easy to use all that power because we have

so much control over the cloud environment," says Standingford.

Computational fluid dynamics and computational aerodynamics have always consumed a huge amount of computing resources. Standingford believes on-demand cloud services such as Oracle Cloud Infrastructure Compute are ideal for these resource-intensive tasks.

"Oracle's adoption-based model can be significantly less expensive than other options if you use enough of it, plus you open yourself up to all sorts of other process improvements, such as training, integration, and metering the cost of your cloud services," says Standingford. "We always have as much power as we need for any given task. Our customers aren't constrained, and we don't pay for more than we use."

And for a company vested in providing innovative cloud services to its own customers, Zenotech's partnership with Oracle is a strategic one. "When it comes to innovation in the cloud, Oracle is one of the fast movers," Standingford says. "For example, they are very responsive to emerging requirements in the areas of adaptive intelligence and machine learning."

Powering Digital Transformations

According to Oracle's Levy, the performance, versatility, and availability inherent in Oracle Cloud Infrastructure comes not only from the servers themselves, but from giving each customer its own virtual network, completely isolated from every other customer's network. Oracle virtualization differs from first-generation cloud providers, which handle virtualization within the hosts—increasing the chance for resource contention and minimizing individual control.

"Oracle's new IaaS offerings unleash the phenomenal performance that the latest-generation hardware can deliver," Levy says. "With Oracle Cloud Infrastructure, customers don't have to worry about the performance degradation that comes when multiple virtual-machine workloads access the same hypervisor at the same time, which can cause I/O bottlenecks."

It's a distinction that hasn't been lost on Zenotech—and thousands of other Oracle Cloud customers. "The Oracle model is making it easier for organizations to understand how and why they should move to the cloud," says Standingford. "More and more companies are now accepting cloud as the norm."

Now that Oracle is handling the bulk of the company's operations, EZCORP's IT team is available to address new opportunities, says EZCORP CIO Dave Hurrell (right), pictured here with EZCORP Vice President of Enterprise IT Ramanujam (Ram) Srinivasan. "By 2018, 60 percent of our IT investments will go to new products, new technology, and new projects—versus 30 percent today," Hurrell says.

Some of these companies utilize IaaS capacity to accelerate digital transformation initiatives, as well as to fast-track the adoption of new technologies. For example, two years ago, EZCORP spent 70 percent of its IT budget maintaining legacy infrastructure and just 30 percent on innovation. In the coming years, senior IT leaders are intent on reversing those numbers as they create a versatile platform for new development and offload their core infrastructure to Oracle Cloud.

"We want to get out of the hardware business and put more emphasis on our revenue-

generating applications," states Ramanujam (Ram) Srinivasan, vice president of enterprise IT at EZCORP. "Our current infrastructure supports 800 stores and cannot scale efficiently to accommodate additional mergers and acquisitions. In addition, our CIO has a vision of delivering far better customer experience, which can be delivered only through a variety of initiatives—a flexible and scalable platform coupled with agile development."

EZCORP, a 28-year-old company, runs the second-largest pawn lending chain in the United States, Canada, and Mexico. EZCORP's business is complex and comprises two closely knit verticals—finance and retail: short-term financing on collateral, and sale of items that come out of such lending as well as used merchandise purchased from customers. To realize the vision for this steadily growing business, EZCORP embarked on an IT transformation project to simplify operational processes, focus on new ideas, and leverage a variety of business data. EZCORP is migrating its on-premises apps and infrastructure to Oracle Cloud Infrastructure and conducting new development natively in Oracle Cloud.

"By 2018, 60 percent of our IT investments will go to new products, new technology, and new projects—versus 30 percent today," says EZCORP CIO Dave Hurrell. "We're reducing costs by US$2 million in 2018 and another US$2 million in 2019—US$4 million total."

EZCORP
Austin, Texas

EMPLOYEES:
7,300

REVENUE:
US$800 million

ORACLE PRODUCTS:
PeopleSoft applications
Oracle WebLogic Server
Oracle Hyperion applications
Oracle Cloud infrastructure and platform products including dedicated compute, storage, log analytics, and application performance monitoring services, Oracle Java Cloud Service, and Oracle Database Cloud Service

Going All In

One of the first applications EZCORP migrated to Oracle Cloud Infrastructure was a homegrown point-of-sale application called EZSystem, which is fundamental to operations at its stores throughout the United States, Canada, and Mexico. The

> **“Oracle’s IaaS vision was very compelling, especially their dedicated compute platforms that we don’t have to share with anybody else. It improves our performance and our security.”**
>
> —**Ramanujam (Ram) Srinivasan**, Vice President, Enterprise IT, EZCORP

company also used Oracle Cloud Infrastructure to implement a disaster-recovery solution with cloud technology company CloudBasic, to host a third-party integration platform that connects its highly distributed operation, to streamline a FreedomPay credit card application that assists with PCI compliance, and to migrate its Oracle’s PeopleSoft applications handling finance and HR to Oracle Cloud.

“We saw significant benefits with the PeopleSoft migration, in terms of the performance we could expect as well as in the amount of administration required from our side,” says Srinivasan. “Overall, we found the lift-and-shift to be easier than expected. Our finance group didn’t even realize that we had already completed a migration to the cloud.”

Srinivasan and his team are now preparing to migrate EZCORP’s Oracle Hyperion financial applications to Oracle Cloud Infrastructure as well, along with several third-party apps based on the Oracle Cloud REST API, which EZCORP depends on to manage inventory and disaster recovery. Ultimately, EZCORP plans to move the bulk of its big data center to Oracle Cloud. “Calling it a migration is a big understatement,” Srinivasan adds, “because we are actually enhancing the capabilities of everything—database upgrade to disaster recovery—that we have been doing. We are able to monitor applications through an insightful dashboard in real time.”

EZCORP examined cloud solutions from IBM, Amazon, Microsoft, and Oracle before choosing Oracle’s cloud infrastructure services. “We have been pushing the envelope with Oracle Cloud,” acknowledges Srinivasan. “We see many of the innovation benefits as well as the productivity gains. We have already identified and made performance optimizations and introduced new features very quickly, enhancing business

agility. Oracle's IaaS vision was very compelling, especially their dedicated compute platforms that we don't have to share with anybody else. It improves our performance and our security."

Srinivasan says he was drawn to Oracle Cloud Infrastructure for its performance specs, citing the outstanding speed achieved with Oracle's unique configuration of memory cores, network fabric, and storage. "The sheer computing power is definitely more compelling on the Oracle platform," he adds.

Hurrell and his team now have a good read on EZCORP's transaction metrics, thanks to the visibility they obtain through the application performance monitoring console that Oracle provides. "We can tell how long it takes users to move from one page to another, how long a shopping cart takes to write back to the database, and how long it takes to generate loan contracts," Hurrell explains. "We can see exactly what's happening across all of our stores and make quick adjustments."

Hurrell also appreciates the inherent disaster recovery capabilities that he receives by using Oracle Cloud. "We're a medium-size business so we don't have a lot of money to invest in a separate data center for disaster recovery purposes," he says. "Our previous SLA was 72 hours in the event of a critical issue. By moving to Oracle Cloud, we've been able to set up real-time database replication in a different city and are in the process of setting up full-blown disaster recovery so we can be back up and running within minutes. That's a huge step forward. We are actually saving money and getting a much better service."

From Operations to Innovation

EZCORP isn't looking to Oracle Cloud just to get out of the hardware business. "We are excited about the growing marketplace and ecosystem of integrated services that will help us concentrate on innovation and speed," Srinivasan says, "such as firewalls, CASB [cloud access security broker], mobile, big data compute, and streaming analytics."

Hurrell also plans to utilize Oracle Cloud Infrastructure Container Service for Oracle Public Cloud as well as Oracle platform-as-a-service (PaaS) products for these new development initiatives at EZCORP, including Oracle Developer Cloud, Oracle Integration Cloud Service, Oracle Big Data Cloud Service, and Oracle Analytics Cloud. Now that Oracle is han-

Elton Oil, which began by creating a disaster recovery site for its JD Edwards data, ultimately plans to move all its business data to Oracle Cloud Infrastructure. "Oracle offered an external, secure, safe, encrypted database in the cloud," says Abdoulaye Dieng, IT manager at Elton Oil.

dling the bulk of the company's operations, EZCORP's IT team is available to address new opportunities. "This is quite appealing to our IT staff," Hurrell says. "They can learn new things, and get involved with new technology rather than just struggling to administer the old."

Boosting Business Continuity

For Elton Oil Company, moving data backups from the data center to the cloud has become critically important to securing that data and making it available in the event of an outage. Elton Oil markets and distributes petroleum

products through 30 service stations throughout Senegal, in West Africa. The company offers epayment services in its Eden's convenience stores, eservices, and Oasis gas stations. Elton also exports petroleum products to neighboring Mali and to its subsidiary in Guinea-Bissau.

Although petroleum reserves have been discovered in Senegal, they have not yet been extracted. All fuel is imported through multinational companies on the open market, with the price per barrel fixed to the dollar. It's a business of sudden market fluctuations and slender margins, which requires Elton Oil to maintain an extremely efficient IT infrastructure.

In 2016, Elton Oil began looking for a cost-effective disaster recovery solution to supplement its local backup strategy. Previously, Elton had tried to back up critical data to a remote site managed by a third party through synchronization of data domains. But this approach failed because of bandwidth issues, says Abdoulaye Dieng, IT manager at Elton Oil.

"All of our business data must be available in the event of a disaster," Dieng explains. "When attending a webinar hosted by Oracle, we learned about Oracle Cloud Infrastructure Storage and decided to move our file storage and email backup to the cloud. Oracle offered an external, secure, safe, encrypted database in the cloud," he adds. "Oracle Cloud services are very affordable compared to other cloud vendors."

Each day, point-of-sale systems within Elton's retail outlets upload data to a central JD Edwards EnterpriseOne enterprise resource planning (ERP) system from Oracle at Elton's headquarters in Dakar. All sensitive data—including financials, management, transportation, distribution, and logistics—resides in a relational database that underpins these JD Edwards applications. Today, with available bandwidth of

ELTON OIL
Dakar, Senegal

EMPLOYEES:
50

REVENUE:
US$97.8 million

ORACLE PRODUCTS:
PeopleSoft and JD Edwards EnterpriseOne applications
Oracle Cloud Infrastructure, including Oracle Cloud Infrastructure Storage and Oracle Cloud Infrastructure Storage Software Appliance

> **“Oracle Cloud is not only a strategic element of Elton Oil’s IT strategy. It is an essential part of our business.”**
>
> —**Abdoulaye Dieng**, IT Manager, Elton Oil

just 2 MB, Elton is able to back up its 100 GB database, compressed to 7 GB, in fewer than 20 minutes thanks to Oracle Cloud Infrastructure Storage Software Appliance—which happens to be free and easy to implement, says Dieng.

“The data segmentation tool provided by Oracle Cloud Infrastructure Storage Software Appliance breaks the data up into small parcels, allowing us to send them to the cloud despite the low bandwidth available,” he adds.

Uploading the data to Oracle Cloud in discrete pieces mitigates Elton Oil’s previous issues with slow wide-area-network speeds. “Once the software appliance was installed and configured, all we had to do was point to the folder in the cloud, which looks like a local folder,” Dieng explains. “SQL agents on the appliance handle all the details of migrating data to Oracle Cloud.”

What’s the upshot? Elton’s widely dispersed staff gained uninterrupted access to timely distribution data, while eliminating concerns about data loss via a natural disaster or other mishap—with no capital investments or additional hardware required. Furthermore, by replicating data across multiple storage nodes within the same data center, Elton Oil gained protection from hardware failures and has minimized the chance of data corruption.

Primed for Expansion

With recent new discoveries of crude oil and gas in the region, Senegal’s petroleum industry is set to expand. As Elton Oil’s business grows, so does its database of products, clients, and suppliers. While Elton Oil began by creating a disaster recovery site for its JD Edwards data, ultimately the company plans to move all its

business data to Oracle Cloud Infrastructure—along with a complete migration of its JD Edwards applications. As Dieng evaluates Oracle's evolving family of IaaS, PaaS, and software-as-a-service (SaaS) solutions, he is confident that their new cloud assets will power the next wave of innovation.

"We like to stay up to date with what Oracle is offering," Dieng concludes. "The robustness and reliability of the Oracle technology are very important to us, and the costs are better than the competing cloud solutions. Oracle Cloud is not only a strategic element of Elton Oil's IT strategy. It is an essential part of our business."

David Baum is a freelance business writer specializing in science and technology.

PHOTOGRAPHY BY
ALI SMITH/THE VERBATIM AGENCY, PAUL S. HOWELL, AND **JANE HAHN/THE VERBATIM AGENCY**

NEXT STEPS

TRY Oracle Cloud Infrastructure Compute.

Brian Lee, engineering manager at Trunk Club, and his team are focused on delivering a better digital experience to customers.

CLOUD COUTURE

At Trunk Club, API design meets fashion design. **BY CHRIS MURPHY**

Trunk Club, an online, personalized shopping service that delivers hand-picked clothes to shoppers' homes, prides itself on helping its customers "build their best wardrobe" with top-quality apparel. Its success at providing that kind of service depends on giving customers an equally elegant connection between digital and physical experiences.

Trunk Club customers share their style preferences with the company online, talk with a stylist by phone or online who then builds a selection of clothes, and view and approve that selection via TrunkClub.com or a mobile application before the trunk is shipped to their home. Customers pay only for the items they keep, along with a US$25 stylist fee (which counts toward any clothes purchased and is waived for Nordstrom cardholders, because Trunk Club is owned by the retailer).

That vital connection between digital and physical experiences makes back-office technology choices critical, because they allow Trunk Club to continually tailor what customers can do online and then bring the experience to life in trunks filled with curated outfits.

"We're always iterating on what we can do better, making small changes and learning from the experience to make it better for the customer," says Brian Lee, engineering manager at Trunk Club.

No surprise, then, that behind its business model, Trunk Club is a technology company at its roots. The company's tech strategy revolves around an API-centric microservices architecture for its front-end web and mobile applications and its back-end systems for customer, product, billing, and other data. A critical platform Trunk Club uses to execute that API strategy is Oracle Apiary Cloud Service, part of Oracle API Platform Cloud Service, which Trunk Club uses to design, document, and manage its APIs. (See the sidebar, "Apiary Extends Oracle API Platform Cloud Service.")

TRUNK CLUB
Chicago, Illinois

INDUSTRY:
Retail

ORACLE PRODUCT:
Oracle Apiary Cloud Service

Collaboration Is Key

Trunk Club's technology strategy is built around a design-first approach—where front-end developers and back-end

Brian Lee, engineering manager at Trunk Club, is working with the data science team to get them started using the Oracle Apiary Cloud Service API design and prototype tools.

systems developers discuss the type of experience they want a Trunk Club customer to have for a given transaction before they ever write a line of code.

Collaboration at the earliest stage of design is vital. Because Oracle Apiary Cloud Service lets you build and share an API model without coding, says Lee, it is easier to get feedback from the people who will use the API. The team uses the cloud service's mock server feature to model how those APIs will work.

"We can use the mock server as if it were a

production service, which accelerates development," says Lee. "And because Apiary generates great API formatting and great documentation, it becomes a lot easier for someone else inside our tech team to look at our documentation and start using it or give feedback on it."

Microservices at the Core

Because Trunk Club is built on a microservices architecture, each element that's needed to put together a customer's trunk of clothes—data about customers, products, price, and so on—comes from its own service.

"Each service has information that it knows about, and the API pulls it together and puts it into a format that the front end can consume," Lee explains. "That's why Apiary and API design are important, because we're specifying that 'we need x, y, and z to show trunk previews to the customer.'"

That's where collaboration and learning happen, too, says Lee, "because the back-end engineer will describe how 'in order to create this trunk object, I need to talk to the order service, the customer service, and whatnot.'"

Having a microservices architecture helps Trunk Club developers quickly add new features, without fear of breaking the entire application if there's a glitch.

"The microservices and the Apiary architecture create a separation of responsibility where each small service has its own small responsibility," says Lee. "So you're able to break up the work so that you can either do it in parallel or do it in manageable pieces."

Once developers write the code, Trunk Club does a code review. "Because we're able to break things up, you don't have to have this gigantic, 10,000 lines of code to review," he adds. "But also when we're shipping it and

Apiary Extends Oracle API Platform Cloud Service

Oracle API Platform Cloud Service enables companies to comprehensively manage the full API lifecycle from design and standardization to documenting, publishing, testing, and managing APIs. Oracle Apiary Cloud Service helps companies accelerate and control the design, development, and documentation of their APIs and microservices.

For more information, visit cloud.oracle.com.

> **“We’re always iterating on what we can do better, making small changes and learning from the experience to make it better for the customer.”**
>
> —**Brian Lee**, Engineering Manager, Trunk Club

making it live, we’re confident that it’s not going to break any other things, because the components are separated and contained within their own services.”

Talk to the API

Trunk Club has created a few rules to govern the communication among its front-end and back-end systems, its APIs, and its services.

“We have microservices in the sense that different services talk to each other,” says Lee. “So one of the rules we have is that the front-end app should not talk to the service directly. It should always talk to the API. And different services can talk to each other—that’s another rule—and the API can talk to the service.” But the service should not talk back to the API, Lee adds.

Setting these guidelines for the information flow makes it easier to visualize how the system works, says Lee. “The API gathers the information and massages it together and gives it to the front end, where it can talk to different services,” he says. “But if a service needs something, the services just talk to each other without having to worry about the API.”

Next Up: Data Science APIs

True innovation, however, can’t happen inside the engineering team alone. To that end, Lee is eager to use Oracle Apiary Cloud Service as the home base for collaboration with teams outside engineering, starting with the data science team. That team builds its own applications to do research, and those applications pull data from some of the same back-end

systems—such as product and customer data—that customer-facing applications use. So Lee is working with the data science team to get them started using the API design and prototype tools.

Because Oracle Apiary Cloud Service creates such clear API formatting and documentation, Lee sees an opportunity for the data science and back-end teams to work in parallel on their projects, letting them do more-creative work, faster, with the confidence that the APIs will bring the disparate elements together. "We trust that this is exactly how the communication is going to work," Lee says. "The back-end engineers and the data science team can work in parallel without blocking each other."

Chris Murphy is director of cloud content at Oracle.

PHOTOGRAPHY BY **PAUL S. HOWELL**

NEXT STEPS

LEARN more about Oracle API Platform Cloud Service.

Database 12*c*

Built for the Cloud

- ✓ Use Less Hardware
- ✓ Lower Operating Costs
- ✓ Manage Many as One
- ✓ No Application Changes

ORACLE®

By Dan McGhan

Using the Callback Pattern and the Async Module

Here's Part 2 in a four-part series on asynchronous Node.js development.

In the previous article in this series, "Asynchronous Processing in Node.js," I covered some of the basics of asynchronous programming in Node.js. In this article, I'll dive deeper into the topic and share some of the patterns used for this development, including the Node.js callback pattern and the Async module.

The examples in the first article used `setTimeout` to simulate asynchronous work. Because `setTimeout` is a simple timer API, there's no chance of an error's occurring. But this is not the case with *real* asynchronous work. Whether you're writing to a file, interacting with a REST API, or executing a database query, errors can and will occur.

In this article, you'll use Node.js to execute a SQL query in Oracle Database. This is a three-step process that must be done serially: first obtain a connection

to the database, then use the connection to execute the query, and finally close the connection. Each step is an asynchronous operation that must include error handling logic.

To set up a test environment you can use to work through this article's examples, check out "Creating a Sandbox for Learning Node.js and Oracle Database" on the Oracle and JavaScript blog.

NODE.JS STYLE CALLBACKS

The developers of Node.js had to take potential errors into account when designing the asynchronous APIs in the core modules. To promote consistency, they developed a couple of callback handling rules:

- The callback function is always the last parameter passed to an asynchronous API.
- The first parameter in the callback function is reserved for errors. If an error occurs, the error will be an instance of the `Error` class; otherwise, the value of the first parameter will be `null`.

The resulting pattern became known as the "Node.js style callback," although some refer to it as the "error-first callback." In either case, it's a very simple pattern that requires developers to check the error parameter to see whether an error occurred and handle it accordingly.

Here's an example that demonstrates the callback pattern using the file system (fs) module:

```
const fs = require('fs');

fs.readFile('/path/to/file.csv', 'utf8', function (err, data) {
 if (err) {
```

```
  console.log('There was an error', err);
  return; // Returning here is important.
 }

 // If you get to this point, assume 'data' is valid.
 console.log(data);
});
```

As you can see, the callback function, `function (err, data){…}`, is passed in as the last parameter to `fs.readFile` and its first parameter, `err`, is for a potential error. The first part of the callback function checks to see if an error occurred. If the path is wrong, the file doesn't exist, or any other error occurs, the expression in the `if` block will evaluate to true, enabling the error to be handled.

In this example, the error is simply logged to the console and the function exits via the `return` statement. The `return` statement is important, because it prevents the success logic from executing. Newcomers to Node.js often forget to add the `return` statement, meaning that success logic executes after errors are handled.

EXECUTING A QUERY WITH NODE.JS-STYLE CALLBACKS

Because the Node.js-style callback pattern is used so heavily in the core Node.js modules, many third-party module developers have adopted the pattern as well. This includes the Oracle team members who developed the Node.js driver for Oracle Database—they used this pattern as the default for all the asynchronous APIs.

The driver, known as node-oracledb—first released to the public in 2015—is open source and hosted on GitHub. It's essentially a layer on top of the Oracle Call Interface libraries included with the Oracle Instant Client. This means that, despite

the driver's relatively young age, it's quite feature-rich and performant.

The node-oracledb driver exposes several classes for doing various operations, from executing SQL and PL/SQL to streaming large result sets and large objects (LOBs). You'll use two of those classes—Oracledb and Connection—to execute a simple query. Oracledb, the base class, will be used to obtain a connection to the database, which will return an instance of the Connection class. The connection will be used to execute the query.

Here's an example of using node-oracledb to execute a query on the employees table in the HR schema. The dbConfig code should work for the App Dev VM (described in "Creating a Sandbox for Learning Node.js and Oracle Database"), but it will need to be modified for other environments.

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

oracledb.getConnection(dbConfig, function(err, conn) {
 if (err) {
  console.log('Error getting connection', err);
  return;
 }

 console.log('Connected to database');
```

```
conn.execute(
 'select *
 from employees',
 [], // no binds
 {
  outFormat: oracledb.OBJECT
 },
 function(err, result) {
  if (err) {
   console.log('Error executing query', err);

   conn.close(function(err) {
    if (err) {
     console.log('Error closing connection', err);
    } else {
     console.log('Connection closed');
    }
   });

  return;
}

console.log('Query executed');
console.log(result.rows);

conn.close(function(err) {
```

```
    if (err) {
     console.log('Error closing connection', err);
    } else {
     console.log('Connection closed');
    }
   });
  }
 );
});
```

As you can see, three different asynchronous APIs are being used in this code:

- oracledb.getConnection
- conn.execute
- conn.close

I'll point out a few other things in the code. First, because each operation must be done serially, the call to the next operation is embedded in the callback function passed to the preceding operation. This nesting of anonymous callback functions—three levels deep in this case—is not quite "callback hell," but you could end up there if any additional steps were added to the sequence. Also, the call to `conn.close` is repeated in two places, one to handle an error and the other for successful completion of the code.

Now copy the code to a new file and name it `anon-functions.js`. To run the script with Node.js, open a terminal, change directories to where the file was created, and run `node anon-functions.js`. You should see the console log output, which includes the employees from the employees table.

Using the named function technique from the previous magazine article, you can refactor the code to eliminate some of the nesting and duplication while making the code a little easier to read and understand (although this requires a few additional lines of code).

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

function getConnection() {
 oracledb.getConnection(dbConfig, function(err, conn) {
  if (err) {
   console.log('Error getting connection', err);
   return;
  }

  console.log('Connected to database');

  executeQuery(conn);
 });
}

function executeQuery(conn) {
```

```
conn.execute(
 'select *
 from employees',
 [], // no binds
 {
  outFormat: oracledb.OBJECT
 },
 function(err, result) {
  if (err) {
   console.log('Error executing query', err);
   closeConnection(conn);
   return;
  }

  console.log('Query executed');
  console.log(result.rows);

  closeConnection(conn);
 }
 );
}

function closeConnection(conn) {
 conn.close(function(err) {
  if (err) {
   console.log('Error closing connection', err);
```

```
  } else {
   console.log('Connection closed');
  }
 });
}

getConnection();
```

Copy this code to a new file, name it `named-functions.js`, and run the script as before. You should get the same output as with the previous version of the code (anon-functions.js).

In the named-functions.js code, I've declared named functions for each operation and then kicked off the sequence of functions at the bottom of the code by invoking the first of the functions to be executed: `getConnection`. With this technique, it's relatively easy to organize sequential asynchronous operations.

More-complex operations, such as iterating an array asynchronously or doing multiple asynchronous tasks in parallel, would require something more. You could write your own library, but why reinvent the wheel when you can just use Async, one of the most popular Node.js libraries ever?

LEVERAGING THE ASYNC MODULE

You can think of Async as a module that extends the Node.js-style callback pattern. Async is not included with Node.js, so it must be installed with a command such as `npm install async`. With Async installed, you can require the library and take advantage of its 70-plus methods for various asynchronous processing situations.

In Async's documentation, its methods fall into three main categories: collections, control flow, and utils. Let's look at an example from the collections category.

```
const async = require('async');

const fakeAsyncApi = function(thing, callback) {
 setTimeout(function() {
  const error = Math.random() > .8 ? true : false;

  if (error) {
   callback(new Error('Failed to process ' + thing));
  } else {
   console.log(thing + ' processed');
   callback(null);
  }
 }, 2000);
};

const thingsToProcess = [
 'thing 1',
 'thing 2',
 'thing 3'
];

async.eachSeries(
 thingsToProcess,
```

```
 fakeAsyncApi,
 function(err) {
  if (err) {
   console.log('An error occurred!');
   console.log(err);
   return;
  }

  console.log('All done!');
 }
);
```

Here's an overview of this Async collections code:

- Line 1: The Async library is required (after having been installed).
- Lines 3–14: A fake API (`fakeAsyncApi`) that implements the Node.js callback pattern is defined. The API will occasionally simulate an error's occurring so that the resulting behavior can be observed.
- Lines 16–20: An array of "things" to process is defined. Typically, the elements would be obtained from reading a file or querying a database. Each element needs to be processed with the fake API.
- Lines 22–34: Async's `eachSeries` method processes the array. The first parameter is the array to be processed, the second parameter is a function (`fakeAsyncApi`) that will process a single element, and the third parameter is a function that should be invoked when all work is done or immediately after an error occurs. Async will pass `fakeAsyncApi` one element from the array, along with a callback

that, when invoked, will let Async know that it can move to the next element in the array.

Copy this Async collections example code to a new file, name it `async-loop.js`, and run it with Node.js (and don't forget to install Async). You should see that each element is processed serially. If an error occurs, processing will stop and the final callback will be invoked immediately.

If you change the `eachSeries` method to `each`, you'll see that all elements are processed in parallel. Pretty cool, huh?

Let's see how one of Async's control flow methods, `waterfall`, can be used to rewrite the database logic from before. I'll adapt the named function version of the code (named-functions.js) from above to make it a little clearer how Async works.

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

function getConnection(callback) {
 oracledb.getConnection(dbConfig, function(err, conn) {
  if (err) {
   console.log('Error getting connection', err);
  } else {
   console.log('Connected to database');
```

```
  }

  callback(err, conn);
 });
}

function executeQuery(conn, callback) {
 conn.execute(
  'select *
  from employees',
  [], // no binds
  {
   outFormat: oracledb.OBJECT
  },
  function(err, result) {
   if (err) {
    console.log('Error executing query', err);
   } else {
    console.log('Query executed');
    console.log(result.rows);
   }

   callback(err, conn);
  }
 );
}
```

```
function closeConnection(conn) {
 if (conn) { // If error getting conn, no need to close.
  conn.close(function(err) {
   if (err) {
    console.log('Error closing connection', err);
   } else {
    console.log('Connection closed');
   }
  });
 }
}
```

```
async.waterfall(
 [
  getConnection,
  executeQuery
 ],
 function(err, conn) {
  closeConnection(conn);
 }
);
```

Copy the code above to a new file, name it `async.js`, and run the script with Node.js. The output should match that of the previous versions of the script.

The first difference in the code you may notice is that Async is required at the top. Also, the signature of the three named functions was changed to work with Async's

conventions. Notice at the bottom that Async's `waterfall` method accepts two parameters: an array of functions to execute and a final function to execute when all the functions in the first parameter are done.

The functions in the first parameter will be executed serially. When invoked, each function is passed a callback function (hence the change in signature) to invoke to execute the next function in the array. When the callback function is invoked, the first parameter passed in should be either `null` or an instance of the `Error` class. This is how to signal to Async that an error has occurred. In the case of `waterfall`, additional parameters can be passed to subsequent functions. Many of Async's methods follow similar conventions, so once you get comfortable with one, it's often easier to use others.

I hope you now have a better understanding of Node.js-style callbacks and how Async can be used to supercharge that callback pattern. In the next article of this series, I'll look at a native way of doing asynchronous work in JavaScript, known as promises.

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

PHOTOGRAPHY BY
CAREY KIRKELLA/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about
JavaScript and Oracle.

ORACLE APPLICATION EXPRESS

From Low Code to High Control

Go no code, low code, and high control with Oracle Application Express 5.1.

By Joel Kallman

Oracle Application Express, the low-code framework of Oracle Database, enables the easy creation of modern and responsive web apps with no code. For many application requirements, however, you will need to go outside the bounds of the declarative framework and define custom logic, business rules, or an enhanced user interface. And you do this with code.

With Oracle Application Express, you can use SQL, PL/SQL, HTML, JavaScript, or CSS to extend your applications. And regardless of what you use to extend your applications, Oracle Application Express provides a gradual transition from no code to low code to high control (more code).

In this *Oracle Magazine* article, you're going to build a web application on top of a table populated with employment data from the US Bureau of Labor Statistics (total nonfarm employees from 1990 to 2017). You will initially create a web application that features a chart on this table, and you will write no code in the process.

You will then add some dynamic capability to this chart by writing a small amount of JavaScript. And then to complete the application, you will add a moderately complex piece of JavaScript to specify the legend of two separate charts.

This article's sample application is built in Oracle Application Express 5.1. If you're not already running Oracle Application Express 5.1 or later locally, you can request a free workspace at apex.oracle.com. Alternatively, you can download the Database App Development Virtual Machine from Oracle Technology Network, which includes a preconfigured Oracle Database 12*c* Release 2 Enterprise Edition instance, Oracle Application Express 5.1, Oracle REST Data Services, Oracle SQL Developer, and Oracle SQL Developer Data Modeler. You will also need to download and unzip the SQL script for this article to create the sample database objects.

CREATING THE SAMPLE DATABASE OBJECTS

Begin your exploration of low-code Oracle Application Express 5.1 development by first creating the sample database objects for your application.

1. In a web browser, log in to Oracle Application Express, click the **SQL Workshop** icon, and then click the **SQL Scripts** icon.
2. Click the **Upload** button, choose the **sample_table_lowcode_apex.sql** file (that you downloaded and unzipped in the previous section) from your local computer, and click the **Upload** button.
3. Click the **Run** icon in the same row as the sample_table_lowcode_apex.sql script, and then click the **Run Now** button.

You've thus created the EMPLOYMENT table in your schema and populated it with data.

CREATING THE APPLICATION

Now create the initial application, which features a simple chart:

1. In Oracle Application Express, click the **App Builder** tab.
2. Click the **Create** icon and then the **Desktop** icon.
3. Enter `Low Code` for **Name**, and click **Next**.
4. Click the **Create Application** button twice.
5. Click the **Create Page** button, click the **Chart** icon, and click the **Line with Area** icon.
6. Enter `Chart` for **Page Name**, and click **Next**.
7. Select **Create a new navigation menu entry**, and click **Next**.
8. For **Table / View Name**, select **EMPLOYMENT** and click **Next**.
9. Choose **MEASUREMENT_DATE** for **Label Column** and **EMPLOYEES** for **Value Column**, and click **Create**.

If you were to run your new application now, it would look a bit confusing. But with a few minor changes, it can look spectacular.

To start the changes, first ensure that you are in the Application Builder of Oracle Application Express, editing page 2.

1. In the tree view on the left side of Page Designer, select **Chart**.
2. In the list of properties on the right, in the Identification section, change **Title** to `Employment`, and in the Advanced section, enter `linechart` for **Static ID**.
3. In the tree view on the left side of Page Designer, select **Attributes** under Chart.
4. In the list of properties on the right, in the Layout section, remove any value for **Height** (if one is present).
5. In the Settings section, change **Time Axis Type** to **Mixed Frequency**.
6. In the Legend section, choose **No** for **Show**.

7. In the tree view on the left side of Page Designer, select **Series 1**.
8. In the list of properties on the right, append an ORDER BY clause to the query, changing **SQL Query** to

```
select * from "EMPLOYMENT" order by state, measurement_date asc
```

9. In the Column Mapping section, choose **STATE** for **Series Name**.
10. In the Label section, choose **No** for **Show**.

You've now created a responsive Oracle Application Express application with an area line chart, showing the nonfarm employment data for four states in the US from 1990 to 2017.

Click the **Run Application** icon in the upper right to run your application. Log in with the same credentials you used to log in to Oracle Application Express. Hover over the lines in the chart to see the data values for each series.

Your application should look similar to **Figure 1**.

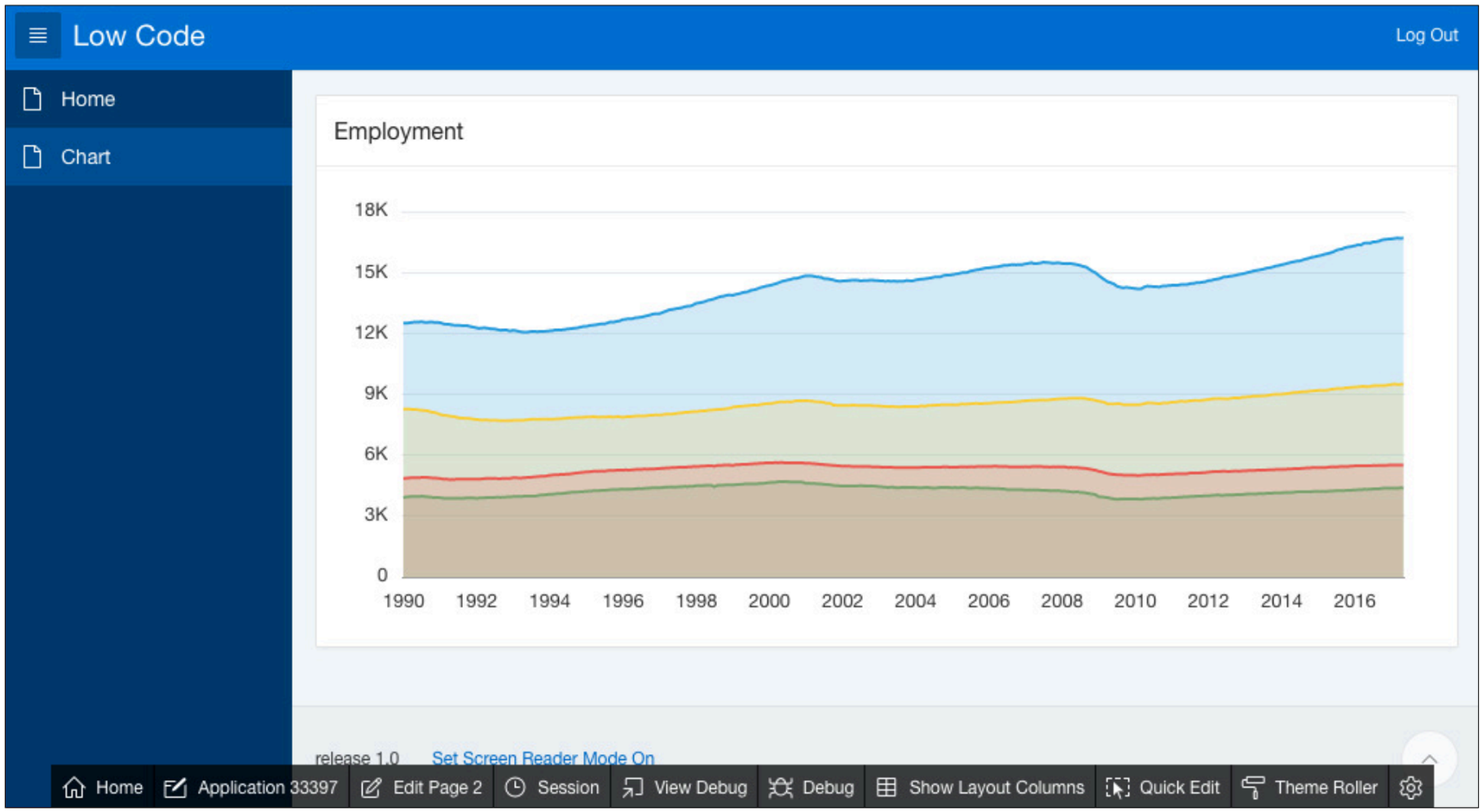
EXTENDING WITH LOW CODE

The primary data visualization engine of Oracle Application Express is based on the Oracle JavaScript Extension Toolkit (Oracle JET), which includes a set of complete and flexible JavaScript APIs.

Using just a small amount of JavaScript, now add some dynamic interactivity for the chart in your new application.

1. In the developer toolbar at the bottom of the page, click the **Edit Page 2** link.
2. In the component gallery at the center bottom of Page Designer, select **Buttons** and then drag and drop a **Text** button up to the Copy region position in the Employment region.

Figure 1: The initial chart application page



3. In the properties on the right, change **Button Name** to `Toggle`, change **Label** to `Toggle Orientation`, and in the Behavior section, change **Action** to **Defined by Dynamic Action**.
4. Move your cursor over the **Toggle** button in the right-side tree view of Page Designer, click the right mouse button, and choose **Create Dynamic Action**.
5. In the properties on the right, change **Name** to `Toggle`.
6. In the tree view at the left of Page Designer, click **Show**, to show the properties of the True action of the dynamic action. This action will be executed when the

condition is true, namely when the Toggle button is clicked.

7. In the properties on the right, change **Action** to **Execute JavaScript Code**. For the Code attribute, enter

```
if ($("#linechart_jet").ojChart("option","orientation") == "vertical") {
    $("#linechart_jet").ojChart({orientation: 'horizontal'});
}
else {
    $("#linechart_jet").ojChart({orientation: 'vertical'});
}
```

8. Click the **Run** icon in the upper right of Page Designer to run your page. Click the **Toggle** button.

Your application page now provides a button for toggling the orientation of the line chart from vertical to horizontal. The static ID attribute you entered in the Creating the Application section gives you the ability to reference the Oracle JET object in JavaScript (by appending _jet to the static ID attribute) and employ all elements of the Oracle JET API. Using a very small amount of JavaScript, you quickly and easily added interactivity to your application.

Your application should look similar to **Figure 2**.

ADDING A SECOND CHART AND A CUSTOM LEGEND

With Oracle Application Express, you can also go outside the bounds of the framework and achieve high control by including large amounts of code where necessary. This code can include CSS, JavaScript, HTML, SQL, or PL/SQL.

Figure 2: The application with the ability to toggle the chart orientation

Now add a second chart and a custom legend to control both charts via slightly more-complex JavaScript.

1. In the developer toolbar at the bottom of the page, click the **Edit Page 2** link.
2. In the component gallery at the bottom of Page Designer, ensure that **Regions** is selected. Select the **Chart** icon, drag it to the right of the Employment region in the Page Layout area, and drop it.
3. In the properties on the far right of Page Designer, for **Title** enter `Average Employment`, and in the Advanced section, enter `piechart` for **Static ID**.
4. Select **Attributes** under Average Employment in the tree view on the left side of Page Designer. Change **Type** to **Pie**, and in the Legend section, change **Show** to **No**.

5. In the tree view on the left side of Page Designer, select **New** under **Series**.
6. In the properties on the far right, change **Name** to `Average Employment`.
7. For **SQL Query**, enter

```
select avg(employees), state from employment group by state order by state
```

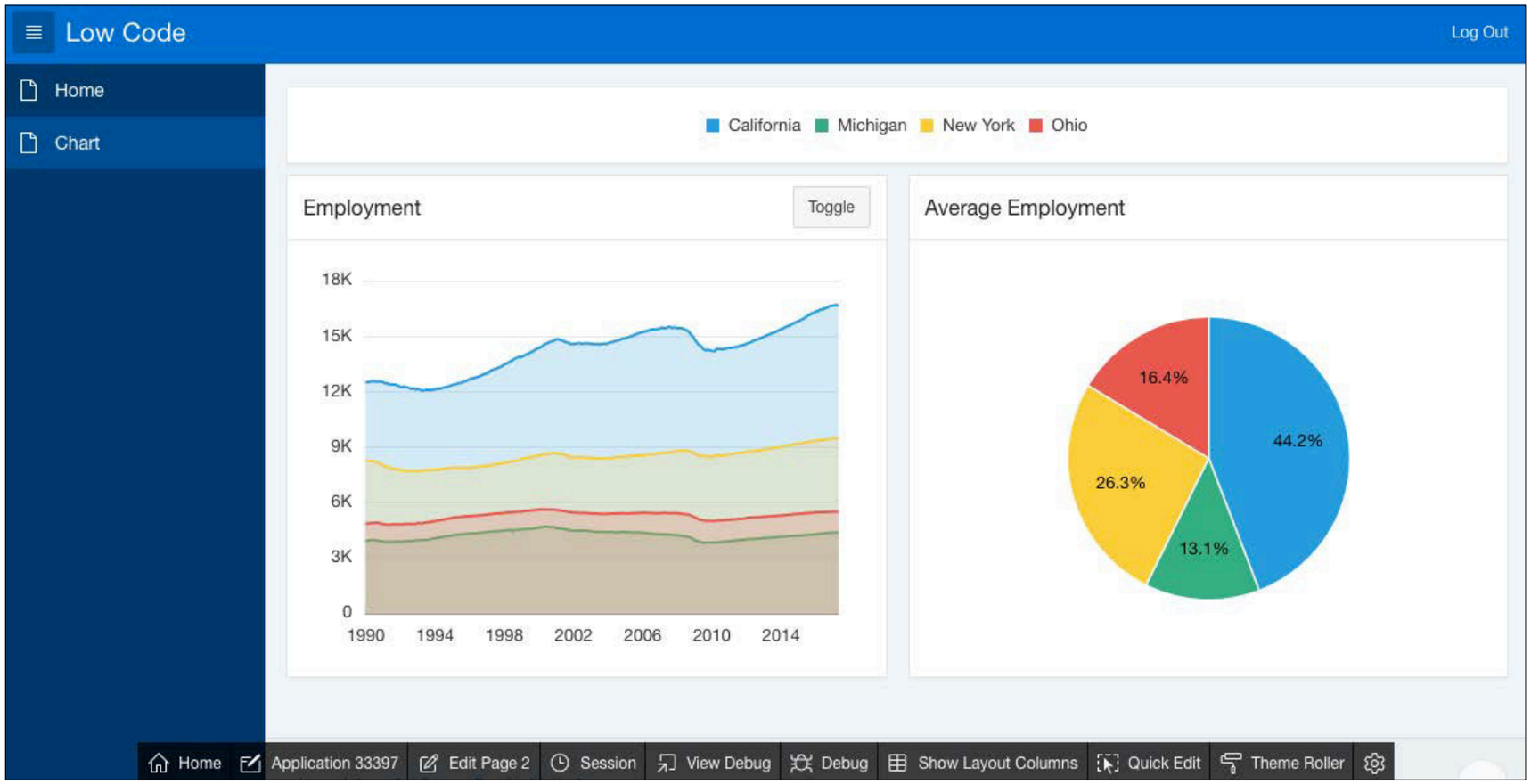
8. In the Column Mapping section, choose **STATE** for **Label** and **AVG(EMPLOYEES)** for **Value**. In the Label section, set **Show** to **Yes**.
9. In the component gallery at the bottom of Page Designer, drag a **Static Content** region above the Employment region in the Page Layout area and drop it.
10. In the properties on the right, change **Title** to **Legend**; in the Appearance section, set **Template** to **Buttons Container**; and in **Source**, enter

```
<div id="legend"
     aria-controls="piechart linechart"
     style="height:25px;">
</div>
```

11. In the left-hand tree, click **Page 2: Chart** at the top.
12. In the properties on the right side of Page Designer, in the JavaScript region, for the property **Execute when Page Loads**, include the contents of the apex_lowcode_legend.js file (which is part of the zip file for this article that you downloaded earlier).
13. Click the **Run** icon in the upper right.

Your application should look similar to **Figure 3**. You now have an application with two charts and a custom legend. Click the states in the legend to show and

Figure 3: The application with a second chart and a single legend to control the series



hide the series from both charts. This additional functionality was provided by the 39 lines of JavaScript from the apex_lowcode_legend.js file, which makes extensive use of the Oracle JET API.

SUMMARY

With no coding, Oracle Application Express enables you to create powerful and responsive web apps. When you need to step outside of the no-code framework of Oracle Application Express to extend your application, you can use code—SQL, PL/SQL, HTML, JavaScript, or CSS. The Oracle Application Express framework

enables you to customize applications with a little code or, when more control is desired, a little more code.

Oracle Senior Director of Software Development Joel Kallman is responsible for the development and product management of Oracle Application Express. He is also a contributing author of several books on Oracle technology, including Expert One-on-One Oracle*,* Beginning Oracle Programming*, and* Mastering Oracle PL/SQL: Practical Solutions*.*

NEXT STEPS

DOWNLOAD Oracle Application Express 5.1.

the SQL script for this article.

READ more about low code with Oracle Application Express.

TRY Oracle Database Cloud Services.

By Chris Muir

ORACLE MOBILE CLOUD SERVICE

Surviving the Mobile Data Offline Blues

Use Oracle Mobile Cloud Service Sync Express offline data capabilities in your JavaScript mobile applications.

Experiencing zero connectivity when you use mobile applications is exceedingly annoying, but a more frustrating experience might be connectivity that keeps dropping in and out, causing mobile apps to stutter and fail without an always-on data connection. Adding the ability for apps to cache data when connected, and rely on the cache when disconnected, allows for a much smoother user experience.

Fetching and maintaining data in the local cache is a significantly different coding challenge from just fetching remote data. Building code to store and maintain data in a cache can be a time-consuming and complex task. Ideally, mobile developers would flip a switch and this caching would happen automatically.

With its JavaScript and Apache Cordova SDKs, Oracle Mobile Cloud Service provides mobile developers with this exact switch, known as Sync Express. Sync

Express is an extension to the JavaScript and Cordova SDKs that—when turned on ("flipped") and configured for a remote REST API—will cache the data on the mobile client, without requiring extra coding.

In this article, you will use an example application to explore how to configure an existing Oracle JavaScript Extension Toolkit (Oracle JET) application to make use of the Oracle Mobile Cloud Service Sync Express feature. The example application is one you've used in previous *Oracle Magazine* articles: a simple employees' phone book application that gets its data from a remote API hosted by Oracle Mobile Cloud Service.

SETUP

For the article, you will first need to install Oracle JET. The Oracle JET documentation covers this task in detail; follow the instructions for installing both the web and mobile tooling components.

Note for this article that the steps assume you are developing on a Microsoft Windows PC and deploying to the Android Emulator on a PC.

After installing Oracle JET, you need to configure Oracle Mobile Cloud Service. This article is based on exactly the same setup instructions from the January/February 2017 *Oracle Magazine* article "Too Much of a Good Thing." Perform Step 1 through Step 8 under the "Deploying a Running Demo Application" section of that article; however, use the demo download file for this article instead of the download file provided in Step 1 of the earlier article. These steps will configure the required Oracle Mobile Cloud Service remote APIs for the mobile application you will see in a moment, supplying the employee details for the employees' phonebook app.

You must also enable cross-origin resource sharing (CORS) for the Oracle Mobile Cloud Service server:

1. From the hamburger menu, select **Administrator**.
2. In the Policies section, click the **Export** button. This will download a policies .properties file to your desktop.
3. Open this file in your favorite text editor.
4. Locate the following line:

```
*.*.Security_AllowOrigin=disallow
```

Replace the line with the following:

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

5. Save the file, and in the Oracle Mobile Cloud Service user interface, click the **Import** button to import the modified file.

With Oracle Mobile Cloud Service artifacts set up, let's create the basic Oracle JET application scaffold.

6. Via the command line, type the following:

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

```
cd OraMagDemo
rd /s src
```

This creates the application scaffold. Note that you just deleted the src directory of the application, leaving the raw scaffold for adding new code.

7. Now download the prebuilt application source code zip file and unzip it. Copy the unzipped src directory into the OraMagDemo directory using Windows Explorer.
8. Within the unzipped src directory, locate and open the src/js/mcsconfig.js file in a text editor, and from the "Too Much of a Good Thing" article's steps, copy the values from Step 5 to Step 8 into the mcsconfig.js file's annotated locations. Then save the file.

TESTING THE VANILLA APP

With the previous steps completed, you are now ready to test the vanilla application before you include Sync Express.

9. Issue the following command in the command-line window:

```
grunt serve --platform=android
```

The application runs and displays the employees' data extracted from the remote Oracle Mobile Cloud Service API.

One feature of the app is the pull-to-refresh-the-list view. Data on the screen is updated via another remote call to the Oracle Mobile Cloud Service custom API. Because the employees' details don't change frequently, refreshing the data won't often result in a visible change to the list. But in a moment, when you start to

refresh the data when the app is online and offline, you will need a way to see when fresh data is coming down from the server.

To present this fresh data, I've "hacked" the Oracle Mobile Cloud Service remote API that I provided and mapped the phone numbers to the current time in milliseconds and incremented each number by the record ID value. As such, each time you refresh the list view, the server updates all the phone numbers of all the employees, and this proves the data is getting updated via the server at each refresh, in turn proving your mobile device is currently online.

Now let's watch what happens when you switch from online to airplane mode, and the app can no longer access the remote API provided by Oracle Mobile Cloud Service.

10. In the Android Emulator, turn on airplane mode. Do this by swiping down the menu bar of the Android Emulator and selecting the airplane icon. Then swipe the menu back up and return to the app.

11. Pull to refresh the list view again, and note that now when the app attempts to refresh the list view, unsurprisingly, the app can no longer fetch fresh data. Depending on the operating system version, this attempt might eventually time out and crash or continue indefinitely. Obviously this isn't what you want. You want the application to continue to run during offline use and, preferably, default back to showing the previously fetched data rather than getting stuck.

This is where Sync Express comes in. It will provide these offline capabilities with minimal effort.

12. Before you proceed, switch the Android Emulator airplane mode off, terminate the Android Emulator, and in the command window press the Ctrl and C keys simultaneously to kill the running process.

ADDING SYNC EXPRESS

There are three steps you must complete to utilize Sync Express in this app.

First, add a script element to the HTML file to load the associated mcs.sync.js library that contains the Sync Express functionality. This file must be the first script the app fetches and loads. Typically, in an Oracle JET app, you would use RequireJS to load the other JavaScript libraries, but the Sync Express library needs to work at a lower level than that so it can capture all remote HTTP calls. That's why you need to add it to the main index.html page as the first script to run.

13. Open the src/index.html file in your favorite text editor.

14. Within the HTML head tag, where annotated, add the following script tag at the end of the <head> tag:

```
<script type="text/javascript" src="js/mcs.sync.js"></script>
```

The tag refers to the js/mcs.sync.js library, and for this article, I've taken a shortcut and provided the library for you in the src directory you copied over earlier from the downloaded zip file. Typically, this JavaScript library is downloaded from Oracle Mobile Cloud Service as part of the JavaScript or Cordova SDK.

Now add the Cordova Network Information plugin to give the sync library the ability to know if the app and device are online or not.

15. In the same command window, execute the following commands:

```
cd hybrid
cordova plugin add cordova-plugin-network-information
```

Now configure the Cordova SDK to specify which remote APIs to cache data from when fetched. You do this through the mcsconfig options you pass to the Cordova SDK when you first initialize it. In the mcsconfig options of the src/mcsconfig.js file you modified earlier, add a new element—syncExpress—at the end, including a policies element and a path element with every path you want Sync Express to cache data from.

16. For this specific use case, add the following to the end of the mcsconfig object definition (including the opening comma):

```
,"syncExpress": {
     "policies": [
       {
         "path": '/mobile/custom/hr/employees/:employeeId(\\d+)?'
       }
     ]
}
```

You've added a path from the Oracle Mobile Cloud Service custom API URL for fetching phone book data from /mobile/custom/hr/employees/:employeeId.

Note the syntax in the path example: a colon followed by "employeeId." This implies "employeeId" is a variable part of the URL path and does change. In addition, the question mark indicates the employeeId is optional, implying sometimes it will just fetch all the employees, as it is doing for this app. Finally, the odd syntax inside the parentheses—two backslashes, "d," and a plus mark—says the employeeId is a decimal rather than a string (the default). If it is a string, you can drop this notation.

Having configured Sync Express, what do you have to do with the code to make it work with Sync Express?

Nothing. As long as you include in the mcsconfig.js file the paths you want to cache, under the covers Sync Express intercepts HTTP calls and will automatically cache the data for you when it is first returned from the server.

Let's look at the outcome.

17. Execute the following in the command-line window to redeploy the app to the Android Emulator:

```
cd ..
grunt serve --platform=android
```

Once the app has been redeployed and started, it initially fetches data from the server because the device is online. Now switch to airplane mode to move the device and the app offline, and refresh the data again by performing a pull. Rather than getting stuck at the refresh screen, the app's call to fetch the remote data returns the previously cached data instead. Now switch out of airplane mode and again pull to refresh the data. The data—pulled from the server—starts updating because the device is online again.

SUMMARY

Oracle Mobile Cloud Service Sync Express allows you to quickly add offline capabilities to your mobile applications. As you've seen in this article, the name "Sync Express" fits the feature well, because adding it to your mobile apps is a relatively painless and turnkey-like process that doesn't require you to extensively modify your source code.

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

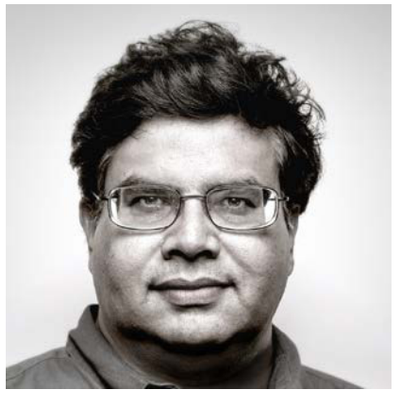
PHOTOGRAPHY BY
SABINE ALBERS/THE VERBATIM AGENCY

NEXT STEPS

READ more about Oracle Mobile Cloud Service.

WATCH Oracle Mobile Cloud Service YouTube training.

TRY Oracle Mobile Cloud Service.



By **Arup Nanda**

ORACLE DATABASE 12*c* RELEASE 2

Automatic and Easy

Oracle Database 12*c* Release 2 delivers better automatic, multicolumn, and read-only partitions.

Partitioning has been a powerful tool for efficient management of large Oracle databases since its introduction in Oracle8 Database. Oracle Database 12*c* Release 2 delivers some powerful new partitioning features. This article explores some of these new features and how to use them.

AUTOMATIC LIST PARTITIONING

Do you remember interval partitioning, which was introduced in Oracle Database 11*g*? It allowed you to create a partitioned table in which new partitions were automatically created on the fly when new data came in, freeing you from the mundane task of constantly adding new partitions. But interval partitioning applied to range partitions only. With new partition support in Oracle Database 12*c* Release 2, you can now create *list partitions* on the fly.

Consider the case of a banking database in which a table named ACCOUNTS includes customer account records. You want to partition the table on the STATUS_CODE column, which contains two-letter abbreviations for the status of customers. Because status codes are discrete values, this is a perfect candidate for list partitioning. However, you don't know all the possible status code values. The code values are created by the bank managers and assigned immediately.

Prior to Oracle Database 12*c* Release 2, if you hadn't created the partition for a new STATUS_CODE value, the insert would fail. You could create a default partition to hold the values and, in that case, the insert would not fail. However, all the new rows would go into the default partition, defeating the purpose of partitioning.

That's not the case anymore. In Oracle Database 12*c* Release 2, you can now create a list-partitioned table using automatic, on-the-fly list partitioning. All you have to do is know one STATUS_CODE column value and define one partition on it. **Listing 1** shows how to create the ACCOUNTS table.

Code Listing 1: Automatic list-partitioned table

```
create table accounts
(
      acc_no          number,
      acc_type        varchar2(1),
      status_code     varchar2(2),
      acc_balance     number
)
partition by list (status_code) automatic
(
      partition p1 values ('VG')
```

```
)
/
```

Notice the AUTOMATIC keyword in the PARTITION clause. This instructs Oracle Database to create the partitions automatically. (Note that this table can't have a default partition.)

Now, let's insert a new row into ACCOUNTS:

```
insert into accounts values (1,'S','VG',2000);
```

The insert is successful because the partition for that value has already been created. Next, insert another row with the STATUS_CODE value SG, for which no partition has been defined.

```
insert into accounts values (2,'S','SG',3000);
```

There is no partition for the SG value and there is no default partition, so how does the insert succeed?

It succeeds because the partition for that value was created instantly at runtime by Oracle Database, and the inserted row found the new partition location. Let's confirm that by checking the partitions of the ACCOUNTS table:

```
select partition_name, high_value
from user_tab_partitions
where table_name = 'ACCOUNTS'
/
```

```
PARTITION_NAME  HIGH_VALUE
______________  __________
P1              'VG'
SYS_P1095       'SG'
```

Note how Oracle Database created a new partition—SYS_P1095—for the SG value. However, you might wonder, did the new row actually go into that partition? Let's check:

```
select * from accounts partition (SYS_P1095)

    ACC_NO ACC_TYPE STATUS_CODE ACC_BALANCE
_________ ________ ___________ ____________
        2        S          SG         3000
```

Yes, the partition was created at runtime to accommodate a new row for which no partition existed, and the row was inserted into that partition.

With automatic list partitioning in Oracle Database 12*c* Release 2, you don't have to know all the possible values of the partitioning key to create the list-partitioned table. Oracle Database handles that for you automatically because of the AUTOMATIC keyword in the PARTITION BY LIST clause.

Oracle Database creates this partition automatically via an autonomous transaction, separate from the INSERT transaction. So, the partition stays, even if you roll back the operation.

The automatic list-partitioning strategy is only for partitions; it cannot be used for a subpartitioning clause. If you use list subpartitioning, you should use the DEFAULT

list as a catch-all subpartition for the values you haven't defined subpartitions for.

Now let's look at a potential irritant. The partition name—SYS_P1095—is system-generated and is not intuitive enough to identify it easily as the partition for the SG value. So, how do you know which partition name to use without constantly checking the USER_TAB_PARTITIONS view, as you did earlier?

The extended partition naming convention comes in handy here. You don't need to know the name of the partition. You can refer to the partition by the value it contains. For example, to select rows from the partition containing the SG value, use the following SQL statement:

```
select * from accounts partition for ('SG');

    ACC_NO ACC_TYPE STATUS_CODE ACC_BALANCE
__________ ________ ___________ ___________

         2        S          SG        3000
```

The result shows the rows from the partition containing the SG value, without the partition being named explicitly in the SELECT statement. You can use this extended naming technique to refer to partitions instead of specific names in any kind of partition management operation—not just in SELECT statements. For example, you can even use extended partition naming to rename partitions. If you don't want to call a partition by the extended reference syntax, you can rename it to a more appropriate name, such as P_SG. Here is how you can rename the partition:

```
alter table accounts rename partition for ('SG') to P_SG;
```

MULTICOLUMN LIST PARTITIONING

Now consider another list-partition case. Suppose the bank wants a different strategy for partitioning the ACCOUNTS table on not one but two columns: STATE_CODE, which contains the two-letter abbreviation for US states, and ACC_TYPE, which shows the type of the account, such as savings, checking, money market, and so on. Because possible values in both columns are discrete, list partitioning is the only choice. But how can you use both columns as partition keys? Remember, you have to use *both* columns as partition keys, not use one column for a partition and the other for a subpartition. The partition and subpartition combination would make a composite-partitioned table—not just a partitioned table—and eliminate the possibility of composite partitioning later.

In Oracle Database 12*c* Release 2, you can create a list-partitioned table with two or more columns as partition keys. **Listing 2** shows the SQL statements to create the table. Note the two columns in the PARTITION BY clause.

Code Listing 2: List partitioning with two columns

```
create table accounts
(
      acc_no          number,
      acc_type        varchar2(1),
      state_code      varchar2(2),
      acc_balance     number
)
partition by list (state_code,acc_type)
(
      partition ct_s values (('CT','S')),
```

```
        partition ny_s values (('NY','S')),
        partition ct_c values (('CT','C')),
        partition ny_c values (('NY','C'))
)
/
```

After the table is created, you can check the partitions in the data dictionary using the SQL statement shown in **Listing 3**. The HIGH_VALUE column shows the two partition values.

Code Listing 3: Checking partitions for two-column list-partitioned table

```
select partition_name, high_value
from user_tab_partitions
where table_name = 'ACCOUNTS'
/
PARTITION_NAME  HIGH_VALUE
______________  ________________

CT_C            ( 'CT', 'C' )
CT_S            ( 'CT', 'S' )
NY_C            ( 'NY', 'C' )
NY_S            ( 'NY', 'S' )
```

How does Oracle Database decide which partition to insert a row into in this case? Let's find out with an example. First, insert a row:

```
insert into accounts values (1,'S','CT',2000);
```

Because the values of the STATE_CODE and ACC_TYPE columns are CT and S, respectively, the inserted row should be located in the CT_S partition. Let's confirm it by selecting rows from that partition:

```
select * from accounts partition (ct_s);

    ACC_NO ACC_TYPE STATE_CODE ACC_BALANCE
__________ ________ __________ ___________
         1        S         CT        2000
```

Perfect! The row went to the right partition, as expected. This is an example with only two columns. You can define list partitioning on multiple columns to suit your specific business needs.

So far, you've learned about two very powerful features: automatic list partitioning and list partitioning on multiple columns. You might now be wondering if it's possible to combine these two features to create an automatic list-partitioned table on multiple columns.

The answer is, yes, of course. **Listing 4** shows how to create the same ACCOUNTS table with automatic list partitioning on two columns: STATE_CODE and ACC_TYPE.

Code Listing 4: Automatic list partitioning on multiple columns

```
create table accounts
(
  acc_no       number,
  acc_type     varchar2(1),
```

```
  state_code   varchar2(2),
  acc_balance  number
)
partition by list (state_code,acc_type) automatic
(
  partition ct_s values (('CT','S'))
)
/
```

Now let's insert a row into this table:

```
insert into accounts values (2,'S','NY',8000);
```

Note that NY is the value in the STATE_CODE column. There is currently no partition for this value, but as you saw earlier, Oracle Database will create new partitions for the values found in new rows and will place the rows in their appropriate partitions. You can refer to the partition by the extended partition name syntax:

```
select * from accounts partition for ('NY','S')
/

    ACC_NO ACC_TYPE STATE_CODE ACC_BALANCE
__________ ________ __________ ___________
         2        S         NY        8000
```

READ-ONLY PARTITIONS

Oracle Database 11*g* introduced a new feature called *read-only tables* that allows you to mark a specific table as read-only, preventing any data manipulation language (DML) statements from affecting that table. This is a powerful feature, but in very large databases with partitioned tables, you probably want to make only a few partitions (or subpartitions) of a table—not the entire table—read-only. For example, suppose you have tables that are range-partitioned on a date column, and you want to make the 2015 partitions read-only, not the entire table. In Oracle Database 12*c* Release 2, you can do this.

Let's consider the ACCOUNTS table in **Listing 5**. This table is range-partitioned on the LAST_TRANS_DT column and list-subpartitioned on the STATE_CODE column with one partition per year.

Code Listing 5: Range-list, composite partitioned table

```
create table accounts
(
        acc_no          number,
        acc_type        varchar2(1),
        state_code      varchar2(2),
        acc_balance     number,
        last_trans_dt   date
)
partition by range (last_trans_dt)
subpartition by list (state_code)
(
      partition p2015 values less than (to_date('01-jan-2016','dd-mon-yyyy'))
```

```
(
      subpartition p2015_ct values ('CT'),
      subpartition p2015_ny values ('NY'),
      subpartition p2015_def values (default)
),
partition p2016 values less than (to_date('01-jan-2017','dd-mon-yyyy'))
(
      subpartition p2016_ct values ('CT'),
      subpartition p2016_ny values ('NY'),
      subpartition p2016_def values (default)
),
partition p2017 values less than (to_date('01-jan-2018','dd-mon-yyyy'))
      (
            subpartition p2017_ct values ('CT'),
            subpartition p2017_ny values ('NY'),
            subpartition p2017_def values (default)
      )
)

/
```

Now, suppose you want to make sure no one can modify the data in the 2015 partition for Connecticut (CT). Here is how you can make that subpartition (of partition "p2015" with "CT") read-only:

```
alter table accounts modify subpartition p2015_ct read only;
```

After running this statement, if you attempt to update the data in that subpartition, you will get an error:

```
insert into accounts values (3,'C','CT',4000,'1-jul-2015')
            *
ERROR at line 1:
ORA-14466: Data in a read-only partition or subpartition cannot be modified.
```

Of course, you can revert the subpartition to the read/write state by using the same ALTER command—with READ WRITE instead of READ ONLY—to allow rows to be inserted into the subpartition.

This ability to make a partition (or subpartition) read-only (or read/write) is not just for the ALTER statement. You can also specify the read attribute of the partition when the table is created, as shown in **Listing 6**.

Code Listing 6: Creating read-only partitions

```
create table accounts
(
        acc_no          number,
        acc_type        varchar2(1),
        state_code      varchar2(2),
        acc_balance     number,
        last_trans_dt   date
)
partition by range (last_trans_dt)
subpartition by list (state_code)
```

```
(
      partition p2015 values less than (to_date('01-jan-2016','dd-mon-yyyy'))
      read only
      (
            subpartition p2015_ct values ('CT'),
            subpartition p2015_ny values ('NY'),
            subpartition p2015_def values (default)
      ),
      partition p2016 values less than (to_date('01-jan-2017','dd-mon-yyyy'))
      read write
      (
            subpartition p2016_ct values ('CT'),
            subpartition p2016_ny values ('NY'),
            subpartition p2016_def values (default)
      ),
      partition p2017 values less than (to_date('01-jan-2018','dd-mon-yyyy'))
      read only
      (
            subpartition p2017_ct values ('CT') read write,
            subpartition p2017_ny values ('NY'),
            subpartition p2017_def values (default)
      )
) /
```

When you define the read attribute of an object, the child objects inherit that attribute. For example, if a partition is read-only, all of its subpartitions will be read-only

as well. You can override the attribute of the subpartition by explicitly specifying it, as shown in **Listing 6** for the p2017_ct subpartition.

You can check the read attribute of partitions and subpartitions from the READ_ONLY column in the USER_TAB_SUBPARTITIONS view:

```
select partition_name, subpartition_name, read_only
from user_tab_subpartitions
where table_name = 'ACCOUNTS'
/
PARTITION_NAME  SUBPARTITION_NAME  READ_ONLY
______________  __________________  _________
P2015           P2015_CT           YES
P2015           P2015_DEF          YES
P2015           P2015_NY           YES
P2016           P2016_CT           NO
P2016           P2016_DEF          NO
P2016           P2016_NY           NO
P2017           P2017_CT           NO
P2017           P2017_DEF          YES
P2017           P2017_NY           YES
```

What if an application has a bug that might inadvertently insert a row into the wrong subpartition? Let's attempt to do exactly that with the ACCOUNTS table created in **Listing 6**:

```
insert into accounts values (5,'S','NY',3000,'1-jul-2017');
```

Because the subpartition was created as read-only, the insert will fail:

```
insert into accounts values (5,'S','NY',3000,'1-jul-2017')
            *
ERROR at line 1:
ORA-14466: Data in a read-only partition or subpartition cannot be modified.
```

You can change the attribute easily using the following SQL statement:

```
alter table accounts modify partition p2017 read write;
```

Now let's insert a row with "2017" in the value:

```
insert into accounts values (5,'S','NY',3000,'1-jul-2017');

1 row created.
```

The insert went through because the subpartition is now in read/write mode.

ONLINE CONVERSION

Now that you have learned about powerful new partitioning features, you might want to convert some unpartitioned tables to partitioned ones. And, like most organizations, yours will likely want that to be done with as little downtime as possible. In Oracle Database 12*c* Release 2, you can do that conversion online. You can convert an unpartitioned table to a partitioned one even when DML statements are

being executed against it. And, what's more, the conversion partitions the indexes of the table appropriately.

Let's see how that works. First, create an unpartitioned table named ACCOUNTS and insert some rows into it:

```
create table accounts
(
        acc_no        number,
        acc_type      varchar2(1),
        state_code    varchar2(2),
        acc_balance   number,
        last_trans_dt date
)
/

insert into accounts values (1,'S','CT',2000,'1-jul-2014')
/
insert into accounts values (2,'S','NY',8000,'01-jul-2016')
/
Commit
/
```

Now let's create four indexes on the table on different columns (note that one of the indexes is unique):

```
create unique index pk_acc on accounts (acc_no);
```

```
create index in_acc_balance on accounts(acc_balance);
create index in_last_trans_dt on accounts(last_trans_dt);
create index in_acc_type on accounts (acc_type);
```

Now, let's convert the table to a list-partitioned table with the STATE_CODE column as the partition key. This conversion operation should be done when the table is subject to DML statements. The conversion's effect on indexes will be mixed. Some indexes, but not all, will be partitioned based on the type of index and what you ask Oracle Database to do with them, as you will see in the example below.

While the DML statements are going on, convert the table to a list-partitioned table using the SQL statement in **Listing 7**:

Code Listing 7: Converting a table to a list-partitioned table

```
alter table accounts
modify
partition by list (state_code) automatic
(
  partition p1 values ('CT')
)
online
update indexes
(
      pk_acc  global,
      in_acc_type local,
      in_acc_balance global
              partition by range (acc_balance)
```

```
                (
                      partition p5000 values less than (5000),
                      partition pmax values less than (maxvalue)
                )
)
/
```

Note the ONLINE keyword in this ALTER statement, which results in the table being converted online. Now check the table for partitions:

```
select partition_name, high_value
from user_tab_partitions
where table_name = 'ACCOUNTS'
/

PARTITION_NAME  HIGH_VALUE
______________  __________
P1              'CT'
SYS_P1103       'NY'
```

That's it. The table has been converted to a partitioned one, all online. The UPDATE INDEXES clause updates the indexes as a part of the operation.

Now let's see the impact of the different index options on this newly converted partitioned table. First, let's see if the indexes are partitioned by using the following SQL statement:

```
select index_name, partitioned
from user_indexes
where table_name = 'ACCOUNTS'

INDEX_NAME           PARTIONED
____________________ _________

IN_ACC_BALANCE       YES
IN_ACC_TYPE          YES
PK_ACC               NO
IN_LAST_TRANS_DT     NO
```

All the four indexes are present in the newly partitioned table. Oracle Database maintained them automatically, which is a result of the UPDATE INDEXES clause. Because you explicitly specified the IN_ACC_TYPE index to be local, the index was partitioned as well to match the table. For the IN_ACC_BALANCE index, you explicitly asked it to be global partitioned, so it became exactly that. You asked the PK_ACC index to be global but not partitioned, so it is not partitioned. Finally, for the IN_LAST_TRANS_DT index, you didn't specify anything, so the index was not partitioned. That's the default behavior for an index of an unpartitioned table where the index didn't contain the column on which the table was partitioned. In this case, the partition column was STATE_CODE, which was not part of the IN_LAST_TRANS_DT index, so it was not partitioned. When an index contains a partition key column, that index is converted to a local partitioned index. Bitmap indexes are converted to local partitioned indexes regardless of whether they have the partition key column in their definition. Don't worry if you got dizzy while reading all that. I will summarize it neatly at the end of this section.

Next, for those indexes that became partitioned in this conversion of the table to a list-partitioned table, let's examine what kind of partitioning scheme the indexes assumed, using the following SQL statement:

```
select index_name, partitioning_type, locality
from user_part_indexes
where table_name = 'ACCOUNTS'
/

INDEX_NAME      PARTITIONING_TYPE   LOCALITY
_______________ ___________________ _________

IN_ACC_BALANCE  RANGE               GLOBAL
IN_ACC_TYPE     LIST                LOCAL
```

In **Listing 7**, because you specified LOCAL for the IN_ACC_TYPE index in the conversion operation, it was converted to LOCAL—no surprise there. Similarly, because you specified GLOBAL for the IN_ACC_BALANCE index, that's what the index was converted to. However, recall that only a local partitioned index is aligned with the partitioning scheme of a table. Global partitioned indexes need to have their own partitioning information, including partition type—range, hash, list, and so on—and the column on which they are partitioned. Because IN_ACC_BALANCE is a global partitioned index, the partitions of the index will not be aligned with the partitions of the table, and you had to explicitly state in **Listing 7** how the index should be partitioned during the conversion operation, which was range partitioning on the ACC_BALANCE column. But was it partitioned accordingly? You can check the partitions

of this index and the corresponding high values using this SQL statement:

```
select partition_name , high_value
from user_ind_partitions
where index_name = 'IN_ACC_BALANCE'

PARTITION_NAME  HIGH_VALUE
______________  __________
P5000           5000
PMAX            MAXVALUE
```

As you can see in the output, the partitions of the index were created as specified.

In summary, **Table 1** shows how the indexes are affected by the UPDATE INDEXES clause when you convert an unpartitioned table to a partitioned one:

If the index is explicitly mentioned in the UPDATE INDEXES clause this way:	The resulting index on the newly partitioned table turns into this:
Local	Local, which means the index is partitioned just like the new table
Global	Global and not partitioned
Global and partitioned	Global and partitioned
Not mentioned at all	It depends; see the following section

Table 2 shows how an index gets converted if you do not include a partitioning clause for the index:

If the index:	After the conversion operation on the table, the index becomes:
Does *not* contain the column on which the table is being partitioned	Unpartitioned
Contains the column on which the table is being partitioned	Locally partitioned
Is a bitmap index, regardless of the column it is on	Locally partitioned

It's very important to understand how the indexes on the table will change as a result of the conversion from an unpartitioned table to a partitioned one and write the UPDATE INDEXES clause appropriately. Note that while you can influence the physical attributes of an index to some extent, you cannot change its design. For example, you can't convert a nonunique index to a unique one or vice versa with the UPDATE INDEXES operation.

ONLINE SPLITTING

Conversion to a partitioned table is a onetime activity for a table, and you probably will do a lot more with a partitioned table, such as splitting partitions. Even partition splitting operations are now available online with Oracle Database 12*c* Release 2.

To see splitting in action, let's first create a partitioned table called ACCOUNTS and insert some rows, as shown in **Listing 8**.

Code Listing 8: Range-partitioned table for splitting

```
create table accounts
(
   acc_no  number,
```

```
   acc_type        varchar2(1),
   state_code      varchar2(2),
   acc_balance     number,
   last_trans_dt   date
)
partition by range (last_trans_dt)
(
   partition p2015 values less than (to_date('01-jan-2016')),
   partition p2016 values less than (to_date('01-jan-2017')),
   partition p2017 values less than (to_date('01-jan-2018'))
)
/
insert into accounts values (1,'S','CT',2000,'1-jul-2014')
/
insert into accounts values (2,'C','CT',2000,'1-jul-2015')
/
insert into accounts values (3,'S','CT',2000,'1-jul-2016')
/
insert into accounts values (4,'C','NY',2000,'1-jul-2014')
/
insert into accounts values (5,'S','NY',2000,'1-jul-2015')
/
insert into accounts values (6,'C','NY',2000,'1-jul-2016')
/
Commit
/
```

Now create two indexes to show the effect of the split on the indexes:

```
create index in_acc_balance on accounts (acc_balance)  local
/
create index in_acc_type on accounts (acc_type)  local
/
```

Note that the ACCOUNTS table has one partition for each year, with the p2015 partition holding values of 2015 and earlier. Now, suppose you want to split the p2015 partition into two partitions: the original p2015 partition containing only 2015 values and a new one called p2014 containing all 2014 and earlier values. This SQL statement accomplishes that objective:

```
alter table accounts
split partition p2015
into
(
  partition p2014 values less than
    (to_date('01-jan-2015','dd-mon-yyyy')),
  partition p2015
);
```

However, this SQL statement also creates two problems. First, this operation needs an exclusive lock on the table, so DML statements can't run against the table at the same time. This means you would have to stop your applications for the duration of the operation.

Second, let's look at the status of the index subpartitions after this partition split operation, as shown in **Listing 9**.

Code Listing 9: Status of index subpartitions

```
select index_name, subpartition_name, status
from user_ind_subpartitions
order by index_name, subpartition_name
/
INDEX_NAME            SUBPARTITION_NAME       STATUS
____________________  ______________________  ________

IN_ACC_BALANCE        P2014_CT                UNUSABLE
IN_ACC_BALANCE        P2014_DEF               UNUSABLE
IN_ACC_BALANCE        P2014_NY                UNUSABLE
IN_ACC_BALANCE        P2015_CT                UNUSABLE
IN_ACC_BALANCE        P2015_DEF               UNUSABLE
IN_ACC_BALANCE        P2015_NY                UNUSABLE
IN_ACC_BALANCE        P2016_CT                USABLE
IN_ACC_BALANCE        P2016_DEF               USABLE
IN_ACC_BALANCE        P2016_NY                USABLE
IN_ACC_BALANCE        P2017_CT                USABLE
IN_ACC_BALANCE        P2017_DEF               USABLE
IN_ACC_BALANCE        P2017_NY                USABLE
IN_ACC_TYPE           P2014_CT                UNUSABLE
IN_ACC_TYPE           P2014_DEF               UNUSABLE
IN_ACC_TYPE           P2014_NY                UNUSABLE
```

```
IN_ACC_TYPE          P2015_CT             UNUSABLE
IN_ACC_TYPE          P2015_DEF            UNUSABLE
IN_ACC_TYPE          P2015_NY             UNUSABLE
IN_ACC_TYPE          P2016_CT             USABLE
IN_ACC_TYPE          P2016_DEF            USABLE
IN_ACC_TYPE          P2016_NY             USABLE
IN_ACC_TYPE          P2017_CT             USABLE
IN_ACC_TYPE          P2017_DEF            USABLE
IN_ACC_TYPE          P2017_NY             USABLE
```

Note that the status of the split subpartitions is UNUSABLE. This is expected, and you should rebuild the index subpartitions. Alternatively, you could have used the UPDATE INDEXES clause in the ALTER TABLE…SPLIT PARTITION statement to have Oracle Database take care of this at the time of the split:

```
alter table accounts
split partition p2015
into
(
  partition p2014 values less than
    (to_date('01-jan-2015','dd-mon-yyyy')),
  partition p2015
)
update indexes
/
```

However, the SQL statement that splits the partitions would still have prevented DML operations on the table, making applications unavailable, which is not a desirable option considering today's nonstop applications.

Stopping DML operations for a partition split is no longer an issue. In Oracle Database 12*c* Release 2, there is an option to perform the SPLIT PARTITION process completely online—while DML statements are still running against the rows in that partition. Here is the online version of the earlier SPLIT PARTITION operation:

```
alter table accounts
split partition p2015
into
(
  partition p2014 values less than
    (to_date('01-jan-2015','dd-mon-yyyy')),
  partition p2015
)
online

/
```

Note that the ONLINE clause in the ALTER TABLE…SPLIT PARTITION statement does the trick. The operation also updates the indexes, keeping them usable. Also, separate index maintenance is not required, because the online operation updates the indexes as well, even without the UPDATE INDEXES clause.

EXCHANGE TABLES

The fastest way to load data into a partition is to exchange a table with it. It's an operation in the data dictionary with no actual data movement, so not only is it fast, but it also doesn't cause a data outage. Here is an example SQL statement that exchanges the temp_accounts_2014 table with the 2014 partition of the ACCOUNTS table:

```
alter table accounts
  exchange partition p2014
  with table temp_accounts_2014;
```

But you need to create the temp_accounts_2014 table with the utmost care. It has to be a faithful representation of the ACCOUNTS table, but not partitioned as such. For example, if ACCOUNTS is just a partitioned table (with no subpartitions), you have to create the temp_accounts_2014 table as unpartitioned. Similarly, if the ACCOUNTS table is composite partitioned, the temp_accounts_2014 table must be partitioned exactly as in the subpartition specification.

In Oracle Database 12*c* Release 2, you don't need to write complex data definition language (DDL) actions to account for the state of partitioning in your exchanges. You can now use a special FOR EXCHANGE clause that creates the table for that very purpose, eliminating all guesswork:

```
create table temp_accounts_2014
tablespace ts_2014
for exchange with table accounts
/
```

Created this way, the temp_accounts_2014 table will have the same columns with the same properties in the same order as the ACCOUNTS table.

SUMMARY

While it has always been a very potent tool for managing large databases, partitioning has improved significantly with Oracle Database 12*c* Release 2. In this article, you learned about a few of the new and enhanced partitioning features, including automatically created partitions, multicolumn partitions, read-only partitions, online partitioning operations, and more.

Arup Nanda has been an Oracle DBA since 1993, handling all aspects of database administration. He was Oracle Magazine*'s DBA of the Year in 2003 and received an Oracle Excellence Award for Technologist of the Year in 2012.*

PHOTOGRAPHY BY **KENNETH GABRIELSEN/ THE VERBATIM AGENCY**

NEXT STEPS

LEARN more about partitioning.

DOWNLOAD Oracle Database 12*c* Release 2.

TRY Oracle Database Cloud services.

By Melanie Caffrey

ORACLE DATABASE

Setting Parameters for Dynamic Productivity

Part 11 in a second series on the basics of the relational database and SQL

This article is the 11th 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, "Meta-Access and Repetitive Composition," introduced you to the data dictionary, from the USER_, ALL_, and DBA_ static data dictionary views to the dynamic performance views (the V$ tables). You saw database scripts combine multiple statements in a sequence into one file. You learned what a substitution variable is and how it can help reduce SQL statement writing. Last, we walked through examples of how substitution variables can be reused and reset.

In this article, you will

- Learn how to run a script with a parameter in SQL*Plus
- Discover how to change the settings of the SQL*Plus environment within a script

- See how to generate HTML reports in SQL*Plus
- Get an introduction to dynamic SQL

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

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

PARAMETERS OF CHANGE

Recall from "Meta-Access and Repetitive Composition" that you can use substitution variables to provide different input values to your scripts each time they are executed. The example in **Listing 1** shows a SQL script that uses a substitution variable. The example in **Listing 2** changes the SQL script in **Listing 1** to use a *parameter* (also referred to as an *argument*) instead of a substitution variable. The only difference between these two examples is that the substitution variable is named &v_table and the parameter is numbered &1. Both prompt users for input when they encounter the ampersand character (&).

Code Listing 1: Use a query with a substitution variable

```
SQL> set lines 10000
SQL> select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  2    from user_tables
  3   where table_name = '&v_table';
old   3:  where table_name = '&v_table'
new   3:  where table_name = 'EMPLOYEE'

TABLE_NAME                     NUM_ROWS LAST_ANALYZED
______________________________ __________ ____________________

EMPLOYEE                               16 13-MAY-2017 14:46:32
```

Code Listing 2: Change the query to use a parameter instead

```
SQL> select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  2    from user_tables
  3   where table_name = '&1';
Enter value for 1: EMPLOYEE
old   3:  where table_name = '&1'
new   3:  where table_name = 'EMPLOYEE'

TABLE_NAME                     NUM_ROWS LAST_ANALYZED
______________________________ __________ ____________________

EMPLOYEE                               16 13-MAY-2017 14:46:32
```

Now compare the result in **Listing 2** with that in **Listing 3**, which places the SQL statement from **Listing 2** into a script file—parameterize_your_script.sql—and calls the file from a SQL*Plus command prompt. Note that the substitution in **Listing 3** is done without prompting and that the substitution is done *positionally*. Positional replacement means that, in order from left to right, any parameter values passed to the parameterize_your_script.sql file at the SQL*Plus command prompt are passed as substitution values to any numbered parameters. In **Listing 3**, the parameterize_your_script.sql file has just one numbered parameter, &1. **Listing 4** expands the parameterize_your_script.sql script created in **Listing 3** to further demonstrate positional notation in parameter value replacement.

Code Listing 3: Place the parameterized statement in a script and execute it from the SQL*Plus command prompt

```
--Create a database script called
--parameterize_your_script.sql
select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  from user_tables
 where table_name = '&1';

--Execute the script with a parameter
SQL> @parameterize_your_script.sql 'EMPLOYEE'
old   3:  where table_name = '&1'
new   3:  where table_name = 'EMPLOYEE'
```

```
TABLE_NAME                     NUM_ROWS LAST_ANALYZED
------------------------------ -------- --------------------
EMPLOYEE                             16 13-MAY-2017 14:46:32
```

The expanded parameterize_your_script.sql file demonstrates how the first command-line parameter value, 'EMPLOYEE', is passed to the &1 parameter, and the second parameter value, 10, is passed to the &2 parameter. Note that this kind of notation works only if your parameterized substitution variable is a number from 1 to 9. Additionally, the script file in **Listing 4** shows how, just as with the named substitution variables, you need to remember to place single quotes around any parameters to which you expect to pass string values. The example in **Listing 4** also demonstrates how the double-line substitution variable replacement display, shown in **Listings 1** through **3**, can be suppressed with the

```
set verify off
```

SQL*Plus command.

Code Listing 4: Add a second parameter and turn off substitution variable replacement display

```
--Alter parameterize_your_script.sql to include a
--second parameter and turn off substitution variable
--replacement display
set verify off

select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
```

```
  from user_tables
 where table_name like '&1'||'%'
   and num_rows > &2;

SQL> @parameterize_your_script.sql 'EMPLOYEE' 10

TABLE_NAME                NUM_ROWS LAST_ANALYZED
_________________________ _________ ___________________________

EMPLOYEE                        16 13-MAY-2017 14:46:32
EMPLOYEE_IDENTITY               16 04-JAN-2017 01:00:32
EMPLOYEE_SUBSET                 16 11-MAY-2017 21:48:06
```

CUSTOMIZING YOUR RETURN ON INPUT

Even though you know how to write and execute individual and multiple SQL statements within script files, developers and DBAs incorporate SQL*Plus commands into their scripts to change the settings of their SQL*Plus environment and, therefore, customize how their scripts execute and return results. The script you used to create the tables for the SQL_201 schemas used for this article's examples incorporates a couple of SQL*Plus commands, including the SPOOL command. The example in **Listing 5** demonstrates the utility of the SPOOL command.

Code Listing 5: Spool the results of your script file to a results file

```
--Incorporate the SQL*Plus SPOOL command into
--your parameterize_your_script.sql file
spool user_table_last_analyzed.lst
set verify off
```

```
select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  from user_tables
 where table_name like '&1'||'%'
   and num_rows > &2;

spool off

--Run the script at the SQL*Plus command prompt
SQL> @parameterize_your_script.sql 'EMPLOYEE' 10

TABLE_NAME                NUM_ROWS LAST_ANALYZED
_________________________ __________ ____________________________

EMPLOYEE                          16 13-MAY-2017 14:46:32
EMPLOYEE_IDENTITY                 16 04-JAN-2017 01:00:32
EMPLOYEE_SUBSET                   16 11-MAY-2017 21:48:06

--View the contents of your spooled output file
TABLE_NAME                NUM_ROWS LAST_ANALYZED
_________________________ __________ ____________________________

EMPLOYEE                          16 13-MAY-2017 14:46:32
EMPLOYEE_IDENTITY                 16 04-JAN-2017 01:00:32
EMPLOYEE_SUBSET                   16 11-MAY-2017 21:48:06
```

The SQL*Plus SPOOL command, used together with a filename of your choosing, spools the results of any SQL*Plus or SQL commands issued subsequently to your

chosen file. You issue the SPOOL OFF command to stop spooling and save and close your results file. To append spooled results to a file you've already created, you can add the APPEND syntax option, such as

```
spool user_table_last_analyzed.lst append
```

If you do not add the APPEND keyword to your SPOOL command, any existing file with the same name as that used in your SPOOL command will be overwritten with each execution of your script file.

Listing 6 includes several additional SQL*Plus commands in the parameterize_your_script.sql script file. The LINESIZE (or LINES) command limits the number of characters displayed on one line before the output begins a new line. PAGESIZE (or PAGES) defaults to a value whereby it repeats the display of column headings every 14 lines. Setting the PAGESIZE value to a large number repeats the display of column headings fewer times, and setting this value to 0 suppresses the display of column headings altogether. TERMOUT (or TERM) controls the output display generated by the script commands.

Code Listing 6: Add more SQL*Plus commands to your database script file

```
--Incorporate additional SQL*Plus commands
--into your parameterize_your_script.sql file
spool user_table_last_analyzed.lst
set verify off
set linesize 60
set pagesize 100
set termout on
```

```
set feedback on

select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  from user_tables
 where table_name like '&1'||'%'
   and num_rows > &2;

spool off

--Run the script at the SQL*Plus command prompt
SQL> @parameterize_your_script.sql 'EMPLOYEE' 10

TABLE_NAME          NUM_ROWS  LAST_ANALYZED
__________________  ________  ____________________________

EMPLOYEE_SUBSET           16  11-MAY-2017 21:48:06
EMPLOYEE_IDENTITY         16  04-JAN-2017 01:00:32
EMPLOYEE                  16  13-MAY-2017 14:46:32

3 rows selected.

--View the contents of your spooled output file
TABLE_NAME            NUM_ROWS     LAST_ANALYZED
____________________  __________   ____________________________

EMPLOYEE_SUBSET             16     11-MAY-2017 21:48:06
EMPLOYEE_IDENTITY           16     04-JAN-2017 01:00:32
```

```
EMPLOYEE                       16     13-MAY-2017 14:46:32

3 rows selected.
```

The OFF setting suppresses the output to the screen if the commands are executed from a script file. Finally, FEEDBACK (or FEED) returns a count of the records returned by a query or statement. Setting this value to 0 or OFF suppresses the feedback. Additionally, if you are using Oracle Database 12*c* Release 2 (12.2.0.1.0), you can use the ONLY option to display feedback but not results. This might come in handy if you are performing any sort of troubleshooting on the performance of your SQL statement and its execution plan.

Although the default spooled output is formatted as fixed-character-length plain-text reports, you can also generate your output as HTML by inserting the following SQL* Plus command into your database script files: SET MARKUP HTML ON.

Listing 7 adds this SQL*Plus command to the parameterize_your_script.sql file and displays the resultant HTML file markup and an approximation of the file's appearance in a web browser. (Remember to change your output file extension to one that a web browser can read.) Additionally, the HTML output will look better when SQL*Plus FEEDBACK is set to OFF. To go back to spooling plain-text output, run the SET MARKUP HTML OFF command.

Code Listing 7: Change your output file from a text file to an HTML file

```
--Incorporate the SQL*Plus command into your
--parameterize_your_script.sql file to output an HTML file
spool user_table_last_analyzed.html
set markup html on spool on
```

```
set verify off
set linesize 60
set pagesize 100
set feedback off
set termout on

select table_name, num_rows, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  from user_tables
 where table_name like '&1'||'%'
   and num_rows > &2;

spool off

--Run the script at the SQL*Plus command prompt
SQL> @parameterize_your_script.sql 'EMPLOYEE' 10
<p>
<table border='1' width='90%' align='center'
summary='Script output'>
<tr>
<th scope="col">
TABLE_NAME
</th>
<th scope="col">
NUM_ROWS
</th>
```

```
<th scope="col">
LAST_ANALYZED
</th>
</tr>
<tr>
<td>
EMPLOYEE_SUBSET
</td>
<td align="right">
        16
</td>
<td>
11-MAY-2017 21:48:06
</td>
</tr>
<tr>
<td>
EMPLOYEE_IDENTITY
</td>
<td align="right">
        16
</td>
<td>
04-JAN-2017 01:00:32
</td>
</tr>
```

```
<tr>
<td>
EMPLOYEE
</td>
<td align="right">
        16
</td>
<td>
13-MAY-2017 14:46:32
</td>
</tr>
</table>
<br>

SQL&gt;

--View the contents of your spooled HTML output file
```

TABLE_NAME	NUM_ROWS	LAST_ANALYZED
EMPLOYEE_SUBSET	16	11-MAY-2017 21:48:06
EMPLOYEE_IDENTITY	16	04-JAN-2017 01:00:32
EMPLOYEE	16	13-MAY-2017 14:46:32

BUILDING ON THE MOVE

In addition to creating and running static SQL statements, you can create SQL at runtime by generating *dynamic SQL*. Dynamic SQL generates additional SQL, which

is useful in situations such as table and other database object maintenance and reporting, where generating multiple statements on the fly is less time-consuming and potentially less error-prone than writing each individual SQL statement by hand. **Listing 8** demonstrates a dynamic SQL statement that generates individual SQL statements for adding audit columns to every table in the USER_TABLES data dictionary view. The text literal

```
'alter table'
```

is concatenated with each table name retrieved from the USER_TABLES data dictionary view. The result of this concatenation is then further concatenated with the text literal

```
'add (created_on date, created_by number, updated_on date, updated_by number); '
```

Code Listing 8: Use dynamic SQL to create ALTER TABLE statements

```
SQL> set lines 60
SQL> select 'alter table '||table_name||' add (created_on
date, created_by number, updated_on date, updated_by number); '
  2    from user_tables;

'ALTERTABLE'||TABLE_NAME||'ADD(CREATED_ONDATE,…);'
_____________________________________________________________

alter table EMPLOYEE add (created_on date, created_by number,
updated_on date, updated_by number);
alter table DEPARTMENT add (created_on date, created_by number,
```

```
updated_on date, updated_by number);
alter table EMPLOYEE_IDENTITY add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_CTAS add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_EXTRA add (created_on date, created_by
number, updated_on date, updated_by number);
alter table ANNUAL_REVIEW add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_SUBSET add (created_on date, created_by
number, updated_on date, updated_by number);
```

```
7 rows selected.
```

The database script in **Listing 9** includes this dynamic SQL query and spools the SQL output to a SQL script—create_audit_columns.sql. Note that instead of spooling to a file with an .lst, .txt, or .html file extension, you spool to a file with a .sql extension to create the dynamically generated SQL command script. **Listing 10** demonstrates how you can run the create_audit_columns.sql script to execute the SQL statements.

Code Listing 9: Use a database script that creates ALTER TABLE statements with dynamic SQL

```
--Place the dynamic SQL statement in a database script file
spool create_audit_columns.sql
set linesize 60
set pagesize 100
```

```
set feedback off
set termout on

select 'alter table '||table_name||' add (created_on date,
created_by number, updated_on date, updated_by number); '
 from user_tables;

spool off

--Run the script at the SQL*Plus command prompt
SQL> @dynamic_audit_column_create.sql

'ALTERTABLE'||TABLE_NAME||'ADD(CREATED_ONDATE,…);'
_____________________________________________________________

alter table EMPLOYEE add (created_on date, created_by number,
updated_on date, updated_by number);
alter table DEPARTMENT add (created_on date, created_by number,
updated_on date, updated_by number);
alter table EMPLOYEE_IDENTITY add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_CTAS add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_EXTRA add (created_on date, created_by
number, updated_on date, updated_by number);
alter table ANNUAL_REVIEW add (created_on date, created_by
number, updated_on date, updated_by number);
```

```
alter table EMPLOYEE_SUBSET add (created_on date, created_by
number, updated_on date, updated_by number);

--View the contents of your spooled output file
'ALTERTABLE'||TABLE_NAME||'ADD(CREATED_ONDATE,…);'
```

```
alter table EMPLOYEE add (created_on date, created_by number,
updated_on date, updated_by number);
alter table DEPARTMENT add (created_on date, created_by number,
updated_on date, updated_by number);
alter table EMPLOYEE_IDENTITY add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_CTAS add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_EXTRA add (created_on date, created_by
number, updated_on date, updated_by number);
alter table ANNUAL_REVIEW add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_SUBSET add (created_on date, created_by
number, updated_on date, updated_by number);
```

Code Listing 10: Use the output file from a spooled dynamic SQL statement execution

```
--Remove the header from the script,
--create_audit_columns.sql, and run the script
SQL> @create_audit_columns.sql
SQL> desc employee
```

```
Name                                      Null? Type
_________________________________________ _____ ________________
EMPLOYEE_ID                               NOT NULL NUMBER
FIRST_NAME                                       VARCHAR2(30)
LAST_NAME                                        VARCHAR2(30)
HIRE_DATE                                        DATE
SALARY                                           NUMBER(9,2)
MANAGER                                          NUMBER
DEPARTMENT_ID                                    NUMBER
WAGE_INCREASE_WORTHINESS                         VARCHAR2(40)
EMP_FULL_NAME                                    VARCHAR2(70)
CREATED_ON                                       DATE
CREATED_BY                                       NUMBER
UPDATED_ON                                       DATE
UPDATED_BY                                       NUMBER
```

FINISHING THE HAT

Once you have a dynamically generated SQL statement file such as create_audit_columns.sql, it is a good idea to save it and add comment lines to the file similar to those illustrated in the example in **Listing 11**. The double-hyphen lines at the top of the create_audit_columns.sql file denote single-line comments. Alternatively, you can use the REMARK (or REM) command (as demonstrated in the script you used to create the tables for the SQL_201 schemas used for this article's examples) or the /* … */ single or multiline comment delimiters. Similar comment lines should also be added to your dynamic SQL scripts for reusability and documentation purposes.

Code Listing 11: Add descriptive comment lines to the new SQL statement file

```
-- Script Name:   create_audit_columns.sql
-- Created Date: <Insert creation date here>
-- Created By:   <Insert author name here>
-- Script Use:   Used to create audit columns for all of the
tables owned by the script author.

alter table EMPLOYEE add (created_on date, created_by number,
updated_on date, updated_by number);
alter table DEPARTMENT add (created_on date, created_by number,
updated_on date, updated_by number);
alter table EMPLOYEE_IDENTITY add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_CTAS add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_EXTRA add (created_on date, created_by
number, updated_on date, updated_by number);
alter table ANNUAL_REVIEW add (created_on date, created_by
number, updated_on date, updated_by number);
alter table EMPLOYEE_SUBSET add (created_on date, created_by
number, updated_on date, updated_by number);
```

CONCLUSION

This article taught you how to pass parameters to database script files. You learned how to save the results from a database script execution to an .lst or .txt file with the SQL*Plus SPOOL command. Additionally, you saw how to save database script exe-

cution results to an .HTML file by using the SQL*Plus SET MARKUP HTML command functionality. You learned how to control aspects of the SQL*Plus execution environment with various common SET commands. Last, you explored how to generate SQL statements at runtime with dynamic SQL and how to document such scripts with comments.

In the next article in this series, you'll learn how to create and alter users, privileges, roles, and synonyms.

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

PHOTOGRAPHY BY **RAY NG**

NEXT STEPS

READ SQL 101, Parts 1–12.

DOWNLOAD the sample script for this article.

LEARN more about relational database design and concepts.

READ more Beyond SQL 101.