ORACLE MAGAZINE

MAY/JUNE 2017

THE BUSINESS OF THINGS

Integrate your internet devices with your business and build better processes with Oracle Internet of Things Cloud Enterprise.

MORE AS THINGS DEVELOP

INTERVIEW > CONTAINERS AND DEVELOPER PRODUCTIVITY

MOBILE DEVELOPER > FAIL FAST, FAIL OFTEN

ORACLE

The New SPARC Servers

FUJITSU

shaping tomorrow with you

New

Fujitsu SPARC M12

World's Most Powerful Processor Core

Mainframe-class processor and system RAS

Continuous scalability for future growth

More ▶

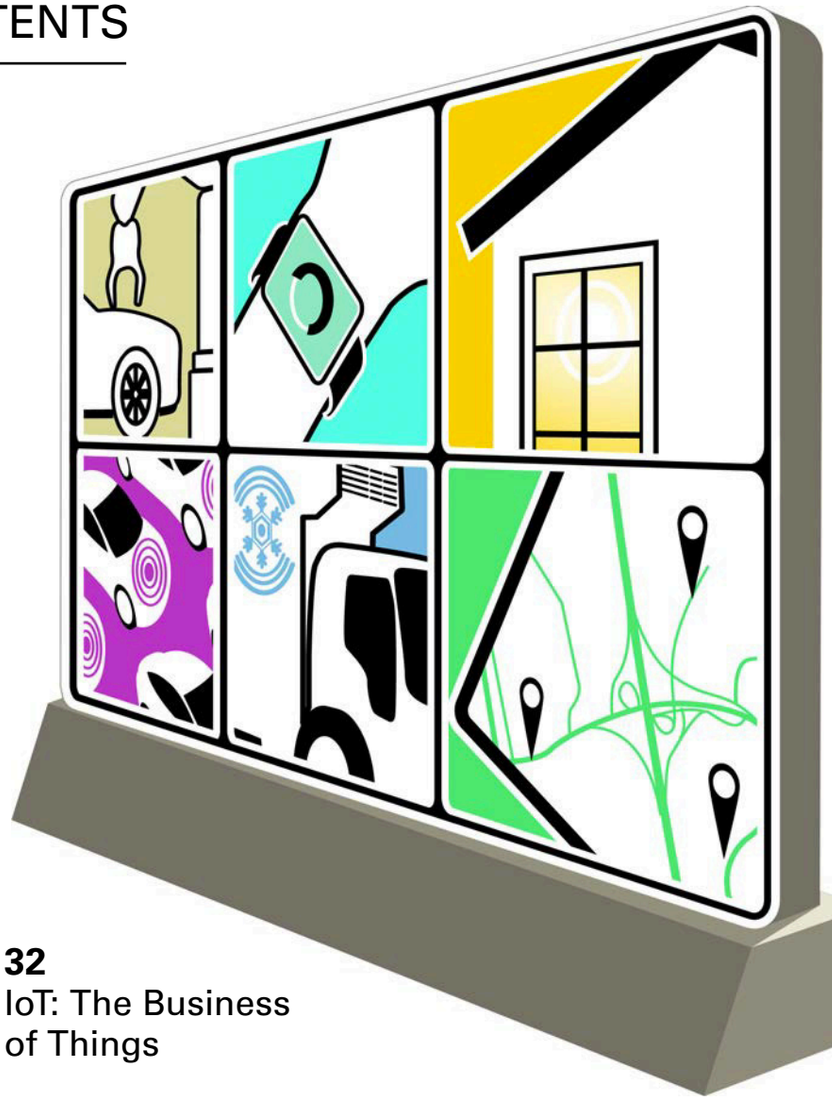
Show each model ▶

Watch

Details

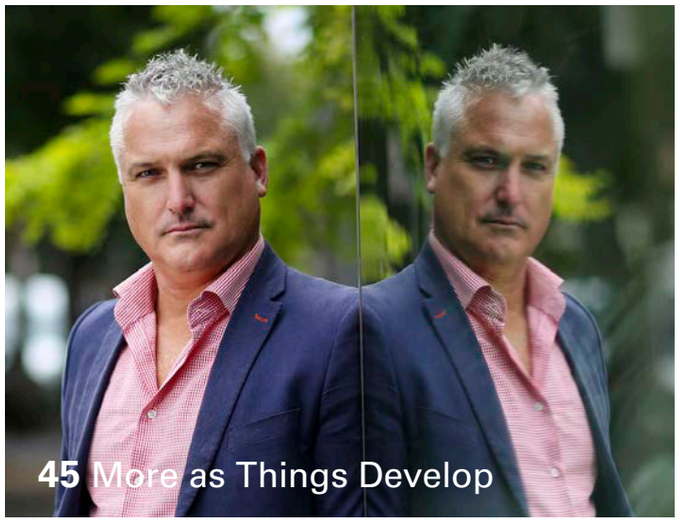
CONTENTS

ORACLE MAGAZINE



32
IoT: The Business of Things

16 Connect, Analyze, Integrate



45 More as Things Develop

FEATURES

UP FRONT

CONTENTS

28 Peer-to-Peer

COMMUNITY

23 Community Bulletin

TECHNOLOGY

ORACLE MAGAZINE

EDITORIAL

Editor in Chief Tom Haunert

Managing Editor Jan Rogers

Editorial Director Robert Preston

Contributing Editors and Writers Blair Campbell, Leslie Steere

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

DESIGN

Vice President, Brand Creative Francisco G Delgadillo

Design Director Richard Merchán

Senior Designer Arianna Pucherelli

Senior Production Manager Sheila Brennan

Designer Jaime Ferrand

Production Designer Kathy Cygnarowicz

PUBLISHING

Publisher Jennifer Hamilton +1.650.506.3794

Associate Publisher and Audience Development Director Karin Kinnear +1.650.506.1985

Audience Development Manager Jennifer Kurtz

ADVERTISING SALES

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

Eastern US and EMEA/APAC Mark Makinney +1.805.709.4745

Mailing-List Rentals Contact your sales representative

SUBSCRIPTION INFORMATION

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

MAGAZINE CUSTOMER SERVICE

Omeda Communications

PRIVACY

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

When you do just one thing,

you get very, very, very, very good at it.

Cloud-based Project Management software solutions—that's all we do.

If all you want is access to the Cloud, call our generalist competitors. But if you want the leading specialist in Cloud-based Project Management solutions, call us. This is our only business, founded on our very, very, very deep knowledge of your complex projects, applications, integrations and infrastructure.

The other guys just don't have our Project Management software expertise, dedicated technologies or unified global support. Sure, anyone can brag about providing infrastructure in minutes, but can they custom-build Your Perfect Cloud for instant productivity and ROI? Thought not.

We're also the only one with the LoadSpringPlatform—our mind-blowing Cloud portal. It not only connects you to 200+ Project Management software titles, training and administration tools, but to real-time support, real humans and real answers. So choose the best. And don't ever, ever, ever settle for less.

America's Top 20 *Engineering, Design & Construction firms all use LOADSPRING**

Technology and Expertise by

LOADSPRING

*Source: 2016 ENR's Top 100 Professional Services Firms and Top 500 Design Firms + LOADSPRING data. **loadspring.com** Follow us on Facebook.

Tom Haunert

From Devices to Decisions

The Internet of Things headlines have been about billions of devices, but the business benefits from the data will be even bigger.

Fifty billion of anything is a lot, and popular predictions put the number of physical devices making up the Internet of Things (IoT) between 20 billion and 50 billion in the next few years. That's headline-big. Those billions of devices need to be designed, built, ordered, delivered, assigned, deployed, managed, maintained, and so on, and information on those endeavors will fill a lot more headlines. But for the IoT, the 50 billion devices aren't the biggest news.

A couple of years ago, the predictions of billions of internet-connected things raised concerns about data storage, processing power, and more. Where do you store the data coming from billions more sensors in factories, vehicles, warehouses, and wearables? Where do you get the compute power to manage that data in real time? Those were good questions, but the cloud has stepped up with abundant storage and compute capacity, at ever more-affordable prices because of robust competition.

With cloud computing infrastructure ready to store and process the data from billions of devices in real time, what's the next industry challenge? Is it choosing or building an IoT platform and applications? Analyzing IoT data in real time and acting on it just as quickly? Those challenges are definitely part of what's next.

The bigger goal is for all of that sensor data—collected by IoT platforms—to inform direct actions, predictive analytics, and machine learning that improve manufacturing and distribution processes, product quality, customer experiences, personal safety, and a lot more.

The biggest IoT questions businesses must answer have little to do with the number of devices they're connecting to the internet.

TO THE DATA

In this issue's cover feature, "The Business of Things," two companies describe their IoT journeys.

Lochbridge started its IoT journey by supporting a leading auto manufacturer in its connected-car offering and is now building an IoT platform on Oracle Internet of Things Cloud Enterprise to improve the businesses of its own customers. Noble Plastics, which provides engineering, contract manufacturing, and related services, plans to use Oracle Internet of Things Asset Monitoring Cloud Service, which runs on Oracle Internet of Things Cloud Enterprise, to improve manufacturing and product quality.

In "Connect, Analyze, Integrate," Oracle Senior Director Harish Gaur discusses how the core features of Oracle Internet of Things Cloud Enterprise help to connect, analyze, and integrate devices, data, and applications. Gaur provides additional examples of how Oracle's IoT services are helping to improve businesses.

And one last thing, in the form of an announcement and an invitation: Oracle OpenWorld 2017 will run from October 1–5 in San Francisco, California. Registration is open, and the *Oracle Magazine* team would love to see you there.

Tom Haunert,
Editor in Chief

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

PHOTOGRAPH BY
BOB ADLER/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Internet of Things Cloud Enterprise.

LEARN more about Oracle Internet of Things applications.

TRY Oracle Internet of Things Cloud Enterprise.

REGISTER for Oracle OpenWorld 2017.

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

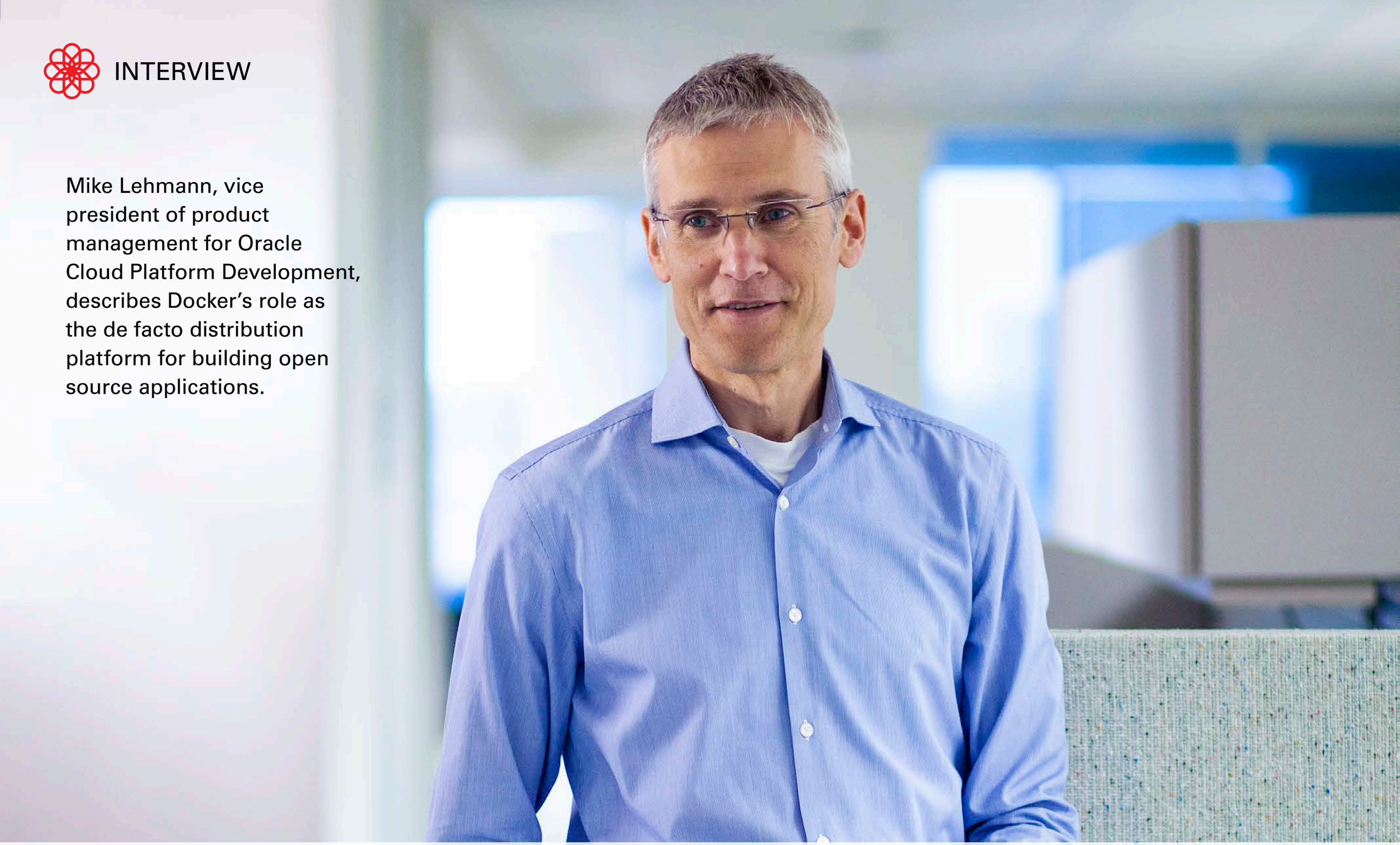
Start Your Free Trial www.melissa.com/oracle

Melissa Data is Now Melissa.
Why the change?
See for Yourself at the New www.melissa.com

melissa™
1-800-MELISSA

INTERVIEW

Mike Lehmann, vice president of product management for Oracle Cloud Platform Development, describes Docker's role as the de facto distribution platform for building open source applications.



Containers and Developer Productivity

Cloud-native development counts on containers for better application delivery and deployment. **BY TOM HAUNERT**

Operating system containers are not a new concept, but today's Linux containers have changed modern application delivery and deployment. Popular Linux containers, including Docker, and the cloud create powerful new opportunities for developers and development processes to take the idea of "write once, deploy anywhere" to a new level. *Oracle Magazine* caught up with Mike Lehmann, vice president of product management for Oracle Cloud Platform Development, to discuss containers, developer speed, infrastructure as a service [IaaS] for developers, and more.

Oracle Magazine: What are containers, and how are they part of cloud-native development?

Lehmann: Containers have been around operating system technology for many years, but modern containers have become popular in the last several years as a result of two things. The first is that Linux, with a fast-maturing Linux Container layer, emerged as a dominant cloud operating system, and the second is that the open source Docker packaging and tooling model became hugely successful in the last two to three years. Simultaneously, traditional hypervisor-based virtualization technologies were being seen as heavyweight and not naturally aligned with modern lightweight cloud-native development trends.

The difference between a modern container, such as a Docker container—the dominant container technology today—and a virtual machine is that a virtual machine generally includes a guest operating system and associated operating system drivers. And as a result, it's not only the application binaries that you're running on the virtual machine but also the guest operating system, and the combined result is fairly heavyweight to both start and move around between environments.

With the move to modern container technologies, a container uses a shared OS or shared kernel underneath, and really what you're packaging, moving around, and starting in a container are just the application and the binaries.

Cloud-native development focuses on lighter-weight technologies and more-agile methods for delivering applications, such as microservices, DevOps, and containers. And helping with cloud-native application delivery,

standardized tooling and APIs for Docker containers make it easy to create, package, and deliver them in modern DevOps pipelines.

Another benefit of containers in cloud-native development is the consistency of having the identical environment available across both what the developer builds and delivers as well as the deployment environment. That consistency across environments means you can have a much quicker release cycle, because each stage uses a standard, immutable representation of the application to validate against.

> "The containerization movement itself is an open source movement with a very broad and rich open source community contributing and building out containerization technologies."

Oracle Magazine: Speaking of the speed of release cycles, in addition to the consistency you mentioned, how—directly—do containers affect the speed of application development and deployment?

Lehmann: When developers build containers, often they are not building an entire container but simply adding image layers to a base canonical container. This supports a process for quickly building more-complex containers from base containers and moving them into the continuous integration and delivery pipeline.

Beyond deploying a single container and seeing that it starts and deploys very fast, there are also orchestration tools for managing containers at scale—enabling a DevOps team to manage large-scale farms of containers across their infrastructures. There is a rich set of emerging technologies in this space, including Kubernetes, Apache Mesos, and Docker Swarm. These tools have helped transition Docker from being a great development and test technology for cloud-native applications to being a large-scale deployment platform for modern applications.

Mike Lehmann, vice president of product management for Oracle Cloud Platform Development, sees the focus of cloud-native development on lighter-weight technologies, including containers.

Oracle Magazine: How do containers work with different infrastructure and platform technologies, frameworks, and programming languages?
Lehmann: Container technologies, such as Docker, live in the larger IaaS ecosystem. A container needs networking infrastructure to communicate with other containers. A container needs to access storage. IaaS providers, including Oracle, provide the baseline compute, networking, and storage, and containers are the next level of infrastructure abstraction leveraging those baseline layers.

The containerization movement itself is an open source movement with a very broad and rich open source community contributing and building out containerization technologies. Docker itself is a commercial enterprise, but the core technology is developed as open source.

Within this ecosystem, most popular open source application development frameworks, runtimes, and databases provide Docker containers via Docker Hub or Docker Store. Node, PHP, Ruby, Cassandra—you name it; they are all available. At the same time, Docker has become the de facto distribution platform for building open source applications.

Companies such as Oracle and other commercial vendors also make their commercial software available via Docker containers. Oracle recently worked with Docker to make database images and middleware images such as the Java Virtual Machine, Oracle WebLogic Server, Oracle Coherence, and even Oracle Tuxedo available in Docker Store as container images.

Oracle Magazine: What are Oracle's cloud services for working with containers?

Lehmann: Oracle supports three levels of deployment infrastructure where people can utilize or leverage container technologies. The first is the basic IaaS, so you can bring your own container technology and your own scheduling and orchestration framework. If you like Docker and Kubernetes or you like Docker and Apache Mesos, you can bring that technology onto Oracle IaaS and roll your own deployment infrastructure.

"Standardized tooling and APIs for Docker containers make it easy to create, package, and deliver them in modern DevOps pipelines."

The next level up is Oracle Container Cloud Service. This is a managed container service on Oracle Cloud that gives you built-in orchestration scheduling of Docker containers. The service lets you set up multiple distributed environments across your compute capacity and deploy containers. It includes powerful features such as stacks, which let you compose containers into groups so you can have a collection of containers that represent a more complex application. For example, one could create a

stack consisting of a cluster of Java microservice containers with a multicontainer Redis cluster back end.

The next level up is Oracle Application Container Cloud, and it's more of a development environment for building microservices. It lets you build out small microservices on a language of your choice—Java, PHP, Node, or another language—following a 12-Factor model of development.

In addition to this container support, Oracle offers one more service for teams building Docker-based applications, and that is Oracle Developer Cloud Service. The service supports developer collaboration, agile tools, source code management, and continuous integration and delivery tooling.

Oracle also recently announced an agreement to purchase Wercker, a Docker lifecycle automation cloud company that enables developers to completely automate the continuous integration and delivery of Docker containers for container-centric and cloud-native applications.

These are exciting times for developers working with Docker containers and Oracle Cloud.

PHOTOGRAPHY BY
BOB ADLER/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Cloud IaaS.

EXPLORE the Oracle developer portal.

TRY Oracle Cloud Platform services.

Harish Gaur, senior director, product management, Internet of Things at Oracle, focuses on the industrial—or enterprise—IoT space.

Connect, Analyze, Integrate

The Internet of Things is much greater than the sum of its devices. **BY TOM HAUNERT**

Mention the Internet of Things (IoT) in technology conversation, and chances are good that people are thinking about the devices—the things—and what they are sensing or seeing. Mention IoT at a business meeting for an organization that's heavily invested in internet-connected things and is getting overwhelmed with IoT-generated data, and chances are good that people are thinking about how that data can better help the business and generate ROI. *Oracle Magazine* caught up with Harish Gaur, senior director, product management, Internet of Things at Oracle, to discuss IoT in the enterprise; connecting devices; analyzing data; machine learning; integrating devices, data, and applications; and more.

Oracle Magazine: What is IoT in general, and what is IoT specifically in the enterprise?

Gaur: Most of the focus on IoT today is in the consumer space.

IoT starts with the idea that every physical device in the world—whether it is a home appliance, an industrial machine, a commercial trailer, or a conference room—can be connected to the internet, and every device will be able to communicate and send information about its behavior to the internet. IoT then works with these connected devices—these interconnected physical devices—to collect and exchange data through the sensors that are attached to them. So that's the textbook definition.

But if you focus on the textbook definition, IoT appears very hardware-centric, very sensor-driven, and much geekier than it really is.

At Oracle we look at the real value of IoT as much more than connecting sensor-equipped physical devices. The devices are a prerequisite, and all the devices need to be smart so that they can send and receive information. But the real value comes from making sense out of this data, from creating insights out of this data, and from taking actions based on this data. Oracle is focused less on the consumer IoT space and more on the industrial or—as I like to call it—the *enterprise* IoT space.

Oracle Magazine: How does IoT work with different types of information, and what is a typical industrial or enterprise IoT use case?

Gaur: People have different estimates for the number of physical devices connected

to the internet. Some say there will be 20 billion devices, and some say there will be 50 billion devices.

Whatever the number, it is going to be so high that the amount of data coming from these devices is just mind boggling. The question then becomes: How do you work with that data and turn it into real actionable information?

> "The breadth of data that will come from sensors . . . will require different levels of analysis—real-time as well as predictive."

So here is an example based on a customer. The company manufactures and sells industrial valves. These valves are sold and used in manufacturing plants at various other businesses, and the valves might be involved in producing high volumes of chemical products. So any malfunction of one of this customer company's valves could have a big impact on the overall quality of the products at another company's manufacturing plant.

So, first, there is a liability issue. Our customer does not want to be in a situation where its valve is functioning well but the quality of its customer's product at a manufacturing plant is suboptimal. Our customer wants to be able to prove that its valve works fine. And if a problem arises with a valve, our customer wants to fix it as soon as possible.

Our customer is remotely monitoring these industrial valves. The valves are expensive—in the €50,000 to €60,000 range—so our customer can afford to instrument these valves and monitor their health, including the rate of flow and the opening and closing of a valve.

Our customer can monitor certain parameters and use that information to detect whether the valve is going to go bad or if it has already gone bad. And based on that information, our customer can dispatch a technician to go fix the valve.

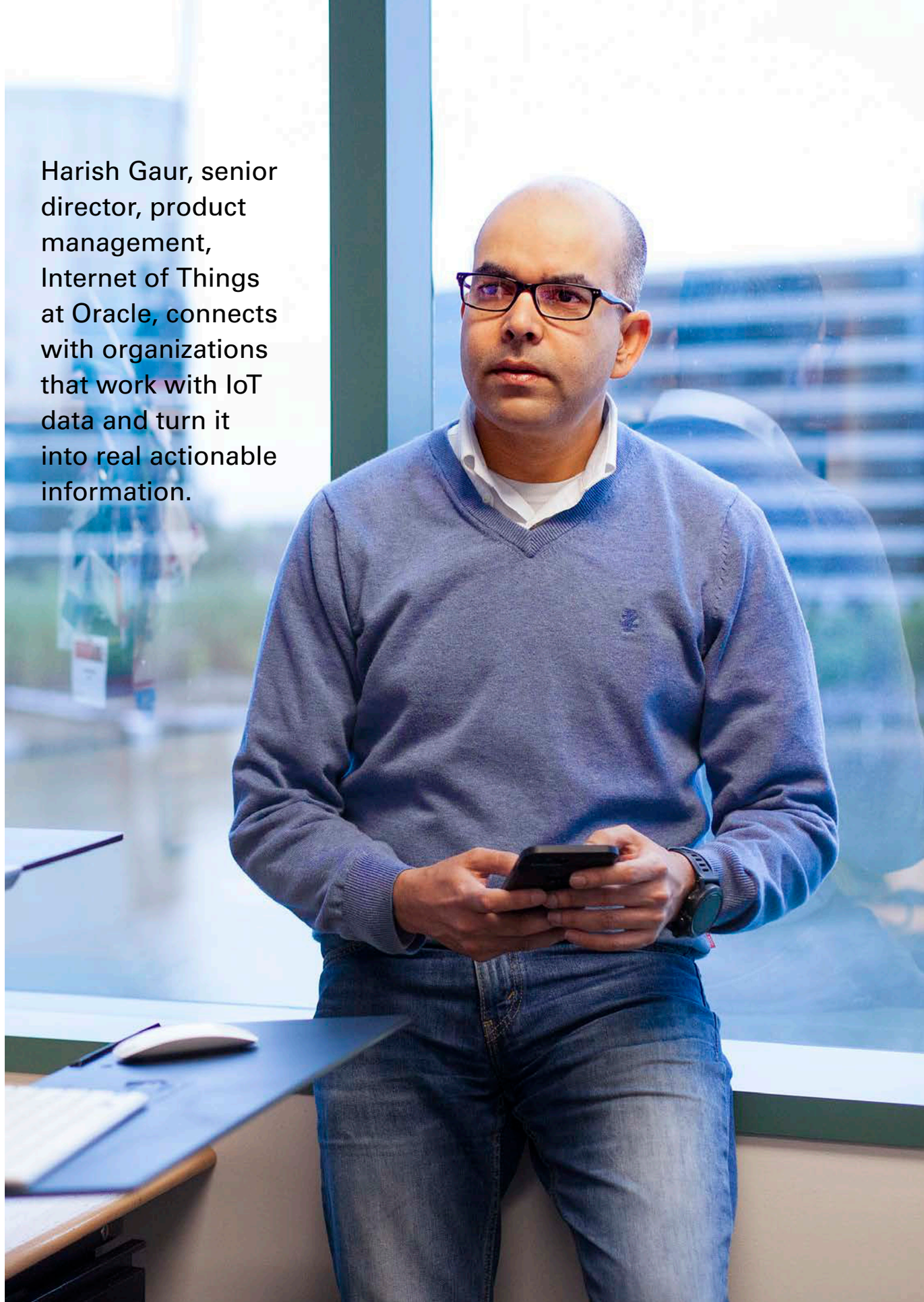
This example is about monitoring raw sensor data, but more than that, it's about using that data to understand whether the valve is currently malfunctioning, to predict future anomalies, and to use that intelligence to drive corrective action—such as creating an incident

in a CRM [customer relationship management] application that will dispatch a technician to go fix the valve so that it doesn't hinder any manufacturing processes. It's about driving ROI from industrial or enterprise IoT.

Oracle Magazine: How does IoT use machine learning today?

Gaur: Machine learning is extremely important in the context of IoT. As I mentioned, the breadth of data that will come from sensors is going to be huge, and it will require different levels of analysis—real-time as well as predictive. The first level of analysis is what's happening right now. And the challenge becomes: Can I process this data in real time and take action?

For example, if a fleet of commercial trucks is delivering perishable food to stores in a grocery chain, I want to monitor the humidity and the temperature inside the trucks so that perishable food does not



Harish Gaur, senior director, product management, Internet of Things at Oracle, connects with organizations that work with IoT data and turn it into real actionable information.

spoil. And I want that information in real time. For example, if the humidity on a truck fluctuates plus or minus 10 percent beyond its hourly average within five seconds, I know there is a problem, and I want to take action. That's real-time analysis of the data.

The next level of analysis is what we typically call predictive analytics, and that's where machine learning comes in. Based on the past performance of this fleet, traffic patterns, weather patterns, and driver behavior, I know that I can predict an estimated travel time of three hours. And using that information, I can plan my next batch of products for dispatch, who should be driving, and what routes the trucks should take. Machine learning supports the predictions and recommendations based on the IoT data.

Oracle Magazine: What are the Oracle solutions for IoT?

Gaur: Oracle offers both SaaS and PaaS [software-as-a-service and platform-as-a-service] IoT services. For the most common use cases we see in our customer base, we provide Oracle Internet of Things Asset Monitoring Cloud Service; Oracle Internet of Things Fleet Monitoring Cloud Service; Oracle Internet of Things Production Monitoring Cloud Service; and coming soon, Oracle Internet of Things Connected Worker Cloud Service. These SaaS apps are built on top of the Oracle IoT PaaS platform: Oracle Internet of Things Cloud Enterprise.

> "Integration is about using what has come from connected devices and been analyzed and telling the logistics application or an asset management module, 'Something has happened; take an action.'"

Oracle Internet of Things Cloud Enterprise is a purpose-built IoT platform that our customers use to build their own IoT applications. The platform supports core feature levels to connect, analyze, and integrate devices, data, and applications.

Connect links the physical devices—the fleets, the machines, the assets, all the devices—and

brings in the data. Oracle Internet of Things Cloud Enterprise connects the devices, pulls in data in a very secure fashion, and supports bidirectional connectivity, where the platform gets data *and* pushes data to devices.

Analyze is real-time analytics of data to see what's happening right now. Oracle Internet of Things Cloud Enterprise supports split-second analytics and machine learning via an IoT analytics module. And that module is built on Apache Spark.

The *integrate* capability is about acting based on analysis and insights. Integration is about using what has come from connected devices and been analyzed and telling the logistics application or an asset management module, "Something has happened; take an action."

PHOTOGRAPHY BY
BOB ADLER/THE VERBATIM AGENCY

NEXT STEPS

LEARN more about Oracle Internet of Things Cloud Enterprise.

LEARN more about Oracle Internet of Things Applications.

TRY Oracle Internet of Things Cloud Enterprise.

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 STEPHEN CHIN, DIRECTOR OF ORACLE TECHNOLOGY NETWORK COMMUNITY MANAGEMENT

DevOps: A Greek Tragedy at Oracle Code

Oracle Code kicked off its first two events in San Francisco, California, and Austin, Texas, with a bang, featuring keynotes from Oracle President Thomas Kurian, Douglas Crockford, and DevOps gurus Baruch Sadogursky and Leonid Igolnik.

Watch the interview with Sadogursky and Igolnik from Oracle Code Austin and their discussion of an amazing modern-day technology movement (DevOps), set in a historical Greek tragedy.

POLYGLOT DEVELOPERS: THIS BLOG'S FOR YOU

The new Oracle Developers Blog is your go-to home for the latest technical developer articles, interviews, tutorials, and demos on Oracle, non-Oracle, and open source topics. Connect with community champions and Oracle product experts as they share their how-to advice on everything from caching with Oracle Application Container Cloud Service to using the newest Apache Spark Streaming runtime infrastructures.

OGh Tech Experience

Oracle Gebruikersclub Holland (OGh) will host its first Tech Experience June 15–16, 2017. This conference merges the successful Oracle Fusion Middleware Experience and DBA & SQL Celebration Day, and it serves the complete Oracle technology stack. The conference takes place in the "Rijtuigenloods," a unique industrial location previously used for restoring and painting railcars.

THE SOUND OF TECHNOLOGY EXPERIENCE

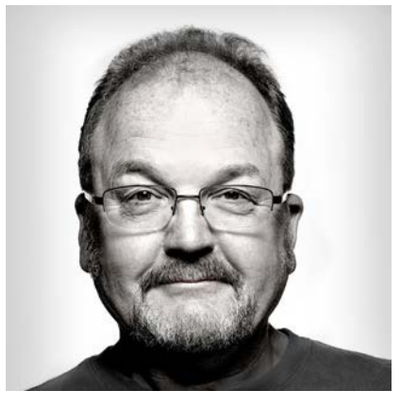
Whether they are weighing down a shelf in your office or glowing on your tablet's touch-screen, technical books are an important and evolving way to absorb insight and experience from thought leaders and technical experts. The Oracle community has its share of such experts, people who have put in the time pounding a keyboard to pour their skills onto pages for your reading pleasure. If you read tech books, or have aspirations of writing one, listen to this OTN ArchBeat Podcast featuring a panel of tech authors—and a representative from a tech publisher—in a wide-ranging conversation about what goes into writing and publishing technical books.

Gear Up for Java SE 9

Java SE 9 is feature-complete, and the ramp-down process has started. Java SE 9 introduces a module system to the Java platform, and the early draft review specification is available now. Take a look at the specification to make sure that your current code will work with JDK 9.

Relentless Tomorrows

Are you ready for the next big tech changes?



By **Bob Rhubart**

It's no small irony that while driving the future through innovation is an almost universal goal among software architects and developers, those same people must also devote a great deal of time and energy to ensuring that their skills will remain relevant in the face of tomorrow's relentless approach.

Two years ago, this column asked community members about how their work had changed in the previous 12 months and about what steps they had taken to adapt to that change. A lot can change in two years, so for this column I asked community members about the proactive steps they are taking to stay ahead of the curve, with a specific focus on any new(er) technologies they are targeting in the ongoing effort to remain future-proof.

For Oracle ACE Director Lucas Jellema, CTO at AMIS Services in the Netherlands, 2017 has been a continuation of the exploration he began in 2016 of Apache Kafka, machine learning, notebooks, Apache Spark, microservices, React, Elasticsearch (ELK stack), Service Workers, Git, Kubernetes, and Oracle Management Cloud and Oracle Database 12*c* Release 2. This year he has added NoSQL and MongoDB to the mix, as well as serverless applications, Kafka Streams, "and anything else that speaks to me," he says.

“I need to keep studying and get out of my comfort zone.”

—Oracle ACE Rolando Carrasco

For Oracle ACE Associate Maarten Smeets, his current technological focus is driven by his role as a senior integration consultant at AMIS. “I’ve been looking more at technologies like back-end JavaScript on Node.js and Apache Kafka,” Smeets says. He’s also tracking what Oracle does with products related to those open source technologies, including Oracle Application Container Cloud Service, Oracle Mobile Cloud Service, and Oracle Event Hub Cloud Service. Also on his radar: Oracle API Platform Cloud Service and Oracle’s acquisition of Apiary. “I choose my products of interest based on what is popular in the market, and also on the choices Oracle makes.” But Smeets also takes a technology-agnostic approach to other interests, including performance tuning and continuous delivery.

If you think of the future as a kind of lottery, Oracle ACE Rolando Carrasco, principal SOA architect and co-owner of S&P Solutions in Mexico, picked the right numbers. “APIs are a hot topic,” he observes, and that bodes well for him. He’s already an expert in the design, management, and use of APIs, and is the coauthor of *Oracle API Management 12c Implementation* (Packt Publishing, 2015). Even so, Carrasco still devotes time to books and articles on the topic, and to learning as much as he can about API management products on the market. “It is changing faster than ever,” he says. “I need to keep studying and get out of my comfort zone.”

Knowing where to place your bets with regard to investing time and effort in acquiring expertise in emerging technologies isn’t just a matter of chance. Every one of the respondents to my questions for this article cites the importance of community engagement as a critical resource for staying on top of the technology trends and understanding what is genuinely shaping the future.

“It is interesting to see what questions get asked on forums, and which forums are active,” says Oracle ACE Associate

Phil Wilkins, senior consultant at Capgemini. “Those regular themes that come up are a good pointer to things worth knowing about,” he explains.

Marketable expertise is all about developing an intimate familiarity with a new technology. What informs your decision to swipe left or swipe right when choosing how and where to invest your time and energy in order to remain marketable in a fickle future? Participate in the ongoing discussion.

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

PHOTOGRAPHY BY
MICHAEL MCELROY/THE VERBATIM AGENCY

NEXT STEPS

READ “Taming Change.”

LISTEN to “IoT in 2017: Things as Apps, Low Power Networks, Edge Computing, and More.”

LISTEN to “Ganging Up on Modern Enterprise Software Architecture.”

LISTEN to “Cloud Architect: Rising to the Role.”

BY BLAIR CAMPBELL

The Simple Approach

Oracle ACE Associates blog, think outside the database, and praise tech that "just works."

Emiliano Fusaglia
Trivadis A.G.
Lausanne, Switzerland

ORACLE ACE Associate

Company: Trivadis A.G, a provider of IT consultancy, system integration, solution engineering, and IT services

Job title/description: Oracle RAC DBA and data architect, working on customer projects and engineering and evolving complex architectures and database cloud solutions

Oracle credentials: Oracle Certified DBA (Oracle9*i* Database, Oracle Database 10*g*, Oracle Database 11*g*, Oracle Database 12*c*), Oracle Certified Expert (Oracle Exadata X3 and X4 Administrator), and Oracle PartnerNetwork Certified Specialist (Oracle Real Application Clusters in Oracle Database 12*c*, Exadata Database Machine X3-2 and X3-8), with more than 16 years of experience using Oracle products

What's your favorite tool on the job? Oracle Trace File Analyzer for Oracle Real Application Clusters [Oracle RAC] environments is the best log-mining tool available. In an Oracle RAC environment, sometimes with more than 10 nodes, Oracle Trace File Analyzer has the capability to highlight all important events, warnings, and errors that occurred on each of the cluster nodes during a specified time range. Such feedback at a glance speeds up any troubleshooting activities and allows me to get valuable information about the health of any known or unknown environments.

You've taken Oracle University [OU] classes in the past. What led you to do this? I took my first OU class for Oracle Advanced Backup and Recovery, and it marked the beginning of my conversion from developer to DBA. My then-manager had

asked me to start taking care of the company's Oracle databases. I took one week of reflection before answering him, and in the end I decided that it was a good move—risky but interesting. So I accepted the responsibility, but on one condition: that I could attend the backup and recovery training courses. I remember telling him, "We can afford poor performance from time to time, but we definitely cannot afford to lose data."

How are you using social media in your work these days? I'm quite active in professional social media, such as LinkedIn and Twitter. I mainly post recent analysis and troubleshooting, with the explanation of the steps performed. I personally find that describing in written form what happened gives me the opportunity to reanalyze the problem with a clear mind. And it may be useful to someone else in a similar situation.

Philip Brown

Red Stack Tech

London, England

ORACLE ACE Associate

Company: Red Stack Tech, a subsidiary of cloud services firm Data Intensity and a provider of technology service management, professional services, and Oracle license management

Job title/description: Director of cloud strategy, enabling clients to transition into the cloud and ensuring that they get the most value out of the technology and the investment

Oracle credentials: Oracle Enterprise Manager 12*c* Certified Implementation Specialist, Oracle Certified Professional (Oracle9*i* Database, Oracle Database 10*g*, Oracle Database 11*g*), Oracle Database 11*g* Security Certified Implementation Specialist, and Oracle Infrastructure as a Service Cloud 2017 Certified Implementation Specialist, with

15 years of experience using Oracle products

Which new Oracle technologies are you currently finding most valuable? At the moment, I'm looking at VPNs [virtual private networks] and networking as part of IaaS and PaaS [infrastructure as a service and platform as a service]. If anyone is thinking about Oracle Cloud, then they need to take a look at the VPN space—and this is everything from Oracle Bare Metal Cloud FastConnect Service to a standard IaaS VPN. Combined with IP networks, VPNs are going to be a cornerstone of all cloud deployments—and it's the thing you need to get right from the start.

What advice do you have about getting into database development? For core DBAs, you need to gain an understanding of the technology ecosystem outside of the core database and understand the challenges that developers face. It's no good explaining to application developers or IT staff that what they are doing is "wrong" or "not performant," or that there's a better way to do something, unless you understand both what they are trying to do and why they are doing it that way. This will help you understand the broader array of challenges faced in software and architecture and recognize the challenges and constraints everyone is working around.

What would you like to see Oracle, as a company, do more of? I'm interested in seeing the further developments in the Oracle Cloud space in IaaS and PaaS. There is clearly a lot of investment going into Oracle Cloud Platform and a lot of new capabilities coming along soon, both in the technology and the consumption models and how that can further ease cloud adoption.

Eric Erikson, CPA

PQR Company

Hickory, North Carolina

ORACLE ACE Associate

Company: PQR Company, an independent IT consulting firm
Job title/description: Consultant, functioning as a solution architect of Oracle Hyperion Financial Management and related modules
Oracle credentials: Oracle Certified Specialist (Oracle Hyperion Financial Management 11), with 24 years of experience using Oracle products

How did you get started in IT? When I was a child, my mother had a job doing what was called data processing, which at the time involved keypunch cards. I got to do things like put the cards into the feeder, carry tapes and disk packs, and make holiday wreaths from used keypunch cards. When I was in 10th grade, we learned BASIC programming on Apple II computers in our math class. Over the next two years, I made a little side money charging people to do their programming homework for them. Career-wise, after college I went to work for a major bank, and my first assignment there was a mainframe general ledger implementation.

Which new features in an Oracle application are you currently finding most valuable? In Oracle Hyperion Financial Data Quality Management, Enterprise Edition, one of the new features is called Universal Data Adapter, where you basically hook up the application to any relational table or view. Oracle Data Integrator is working underneath Oracle Hyperion Financial Data Quality Management to make things happen—and while that sounds pretty basic, there's a great value to the end users. It's simple to set up, and it just works.

What technology has most changed your life? Definitely the Oracle Hyperion consolidation products—first the now-defunct Hyperion Enterprise product; then Oracle Hyperion Financial Management; and, most recently, Oracle Financial Consolidation and Close Cloud Service, Oracle's cloud solution for financial consolidation. These technologies have helped me build a career around financial consolidation consulting. With Oracle Financial Consolidation and Close Cloud Service, I've noticed that although the technology and platform are new, the design lessons learned over the years are being applied. For example, the intercompany matching report format uses the "same row" format from the Hyperion Enterprise product. I'm glad to see the return of what worked best.

IoT: THE BUSINESS OF THINGS

Oracle Internet of Things cloud services integrate devices and help organizations build on data in new ways.

BY MIKE FADEN

Discussion of the Internet of Things (IoT) often focuses on new connected devices—whether they're cars, automation systems, or home appliances. But for enterprises looking to derive business value from IoT, connecting devices is just the first step. The real opportunities—and challenges—lie in capitalizing on the unprecedented torrents of data generated by those devices. That requires integration with other enterprise applications.

"At Oracle, we look at the real value of IoT as much more than connecting sensor-equipped physical devices," says Harish Gaur, senior director, product management, Internet of Things at Oracle. "The devices are a prerequisite, and all the devices need to be smart so that they can send and receive information. But the real value comes from making sense out of this data, from creating insights out of this data, and taking actions based on this data."

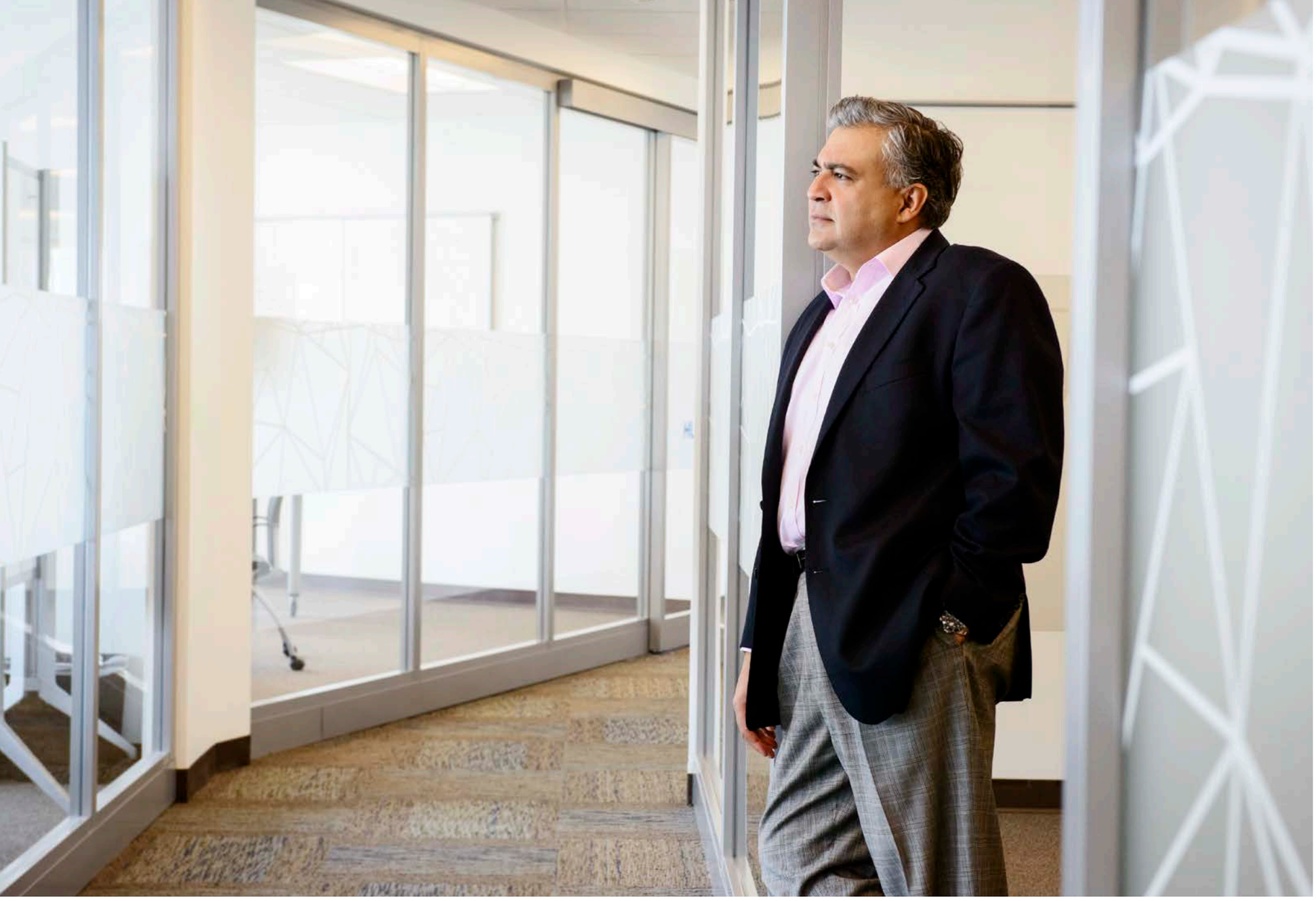
Monetizing Data

Forward-looking organizations are taking notice. "Companies tend to think about the devices. But what starts as a device play soon becomes a data play," says Romil Bahl, CEO at Detroit, Michigan–based technology consulting and services provider Lochbridge. "IoT leads to data—and there's value to be harvested from that data."

Indeed, Bahl views the rise of IoT—especially connected cars—as an opportunity to drive revenue growth and transform Lochbridge's business. Lochbridge has a long history of involvement with IoT: the company helped a leading automotive manufacturer pioneer its connected-car offering by helping to develop the product's service-delivery platform.

Now, Bahl plans to use Lochbridge's connected-car expertise to fuel an aggressive expansion plan: the goal is to double the company's revenue by 2020, while delivering a return on investment of at least 400 percent to its customers. "Our vision is to capitalize on this capability set, starting with the core automotive space but then expanding more broadly into the field of IoT," he says. "If we can connect to a car that's hurtling down a highway at 80 miles an hour, we can connect to pretty much any IoT device."

Lochbridge has used on-premises software, including Oracle Fusion Middleware, to deliver software in the connected-car space and for



Lochbridge views the rise of IoT as an opportunity to grow revenue and transform the company's business. "IoT leads to data—and there's value to be harvested from that data," says Lochbridge CEO Romil Bahl.

its nonautomotive clients as well. But for new services, the company has fully embraced the cloud. "Our internal strategy is to leverage the cloud for everything," Bahl says. "For example, we don't have an on-prem test and development environment. Everything's in the cloud."

Key advantages of the cloud include drastically reduced application provisioning times and operational costs, Bahl says. Just as important, Oracle's platform-as-a-service (PaaS) products help Lochbridge develop and deploy new services more quickly.

> **If we can connect to a car that's hurtling down a highway at 80 miles an hour, we can connect to pretty much any IoT device.**
>
> —**Romil Bahl,** CEO, Lochbridge

Notably, Lochbridge has constructed its strategic IoT intellectual property, called the IoT Acceleration Framework, using Oracle Internet of Things Cloud Enterprise, which provides built-in capabilities to connect different devices, analyze data, and integrate with other cloud-based and on-premises applications. "Out of the box, Oracle Internet of Things Cloud Enterprise provides the basic building blocks required to connect devices and then gain insights," Bahl says. "That obviously helps us onboard new customers quickly. Because the Oracle platform enables device integration and management, it enables us to focus on solutions built on top of that platform."

The ability to integrate with other enterprise applications and share intelligence is critical, adds Raj Paul, vice president of IoT and connected services at Lochbridge. "Our view of the world is that IoT brings a 'sixth sense,'" he says, "so obviously, seamless integration of that sixth sense into the enterprise is important." To build its services, Lochbridge is combining Oracle Internet of Things Cloud Enterprise with other PaaS services from Oracle, including Oracle Database Cloud Service and Oracle Java Cloud Service. The company also uses Oracle Big Data Cloud Service for applications involving complex analysis or regression modeling.

"It's very easy for us to implement all plausible use cases by having both Oracle Big Data Cloud Service and Oracle Database Cloud Service," Paul says. "That is one of the primary reasons we went with this deployment strategy, so that we could keep things open."

Using Oracle's cloud products, Lochbridge is building IoT services designed to drive specific business outcomes for its customers,

such as customer loyalty and product quality, Bahl says. An example is the company's diagnostics-as-a-service offering, which tracks a device's health, identifies potential problems, and sends alerts and service reminders. "We're able to proactively send messages to customers about the need to get to the service center for maintenance or to the manufacturer/OEM to dispatch service personnel to the customer," he says. "We can send them a health report that aggregates information from multiple sources to gather insights about the status of the device, and then tell them everything from a part's system status to 'there's an issue with this part and you need to go in or get an upgrade.'"

Lochbridge has created analogous services for companies in several industries, including a mining company that needs to monitor and diagnose the overall health of equipment deep in its mines and a utility company that monitors the devices on and utilization of its grid.

"There's tremendous value in being able to gather data on usage patterns and send customers proactive reminders when maintenance issues seem to be emerging," Bahl says. "Now you're talking differentiation. You're driving loyalty. And you're gathering a ton of data to improve your products, which over time reduces your warranty and repair costs."

Another Lochbridge service enables usage-based car insurance. The service, which Lochbridge is building on Oracle's IoT cloud platform, uses data from vehicle sensors to help insurers calculate premiums based on how people actually drive. The new service is the latest of several generations of risk-based insurance systems, Bahl says. Each generation has provided more-sophisticated risk analysis by analyzing greater amounts of data from a car's 200-plus sensors and other sources.

"The real value comes from making sense out of this data, from creating insights out of this data, and taking actions based on this data."

—**Harish Gaur,** Senior Director, Product Management, Internet of Things, Oracle

"The first generation was based simply on how many miles you drive," he says. "The second generation added driver behavior to the mix: Are you turning abruptly, or are you screeching to a halt too many times? And the third generation can score a driver based on hundreds of data elements—not just how you're driving, but also external factors. So, yes, you braked pretty hard, but there was an ice storm or other reasons for that."

Manufacturing Value

At Noble Plastics, enterprise IoT involves devices that are considerably less mobile than cars, but just as complex and challenging: robot-managed manufacturing systems.

Based in Grand Coteau, Louisiana, Noble helps companies turn their ideas into products by offering design engineering services, contract manufacturing based on injection molding, and automation systems. Noble applies its expertise to a wide variety of products, from smart munitions used by US special forces to the paddles in margarita machines.

To maximize efficiency, Noble typically runs its manufacturing cells unattended, so they can operate 24/7. A typical cell includes injection molding machines, auxiliary monitoring systems, and a Fanuc robot that supervises the process and also handles the finished parts. Vast amounts of information, including parameters such as the molding machine pressure and temperature, are collected during the manufacturing process.

Scott Rogers, technical director at Noble Plastics, wants to analyze that information to improve the process, potentially reducing the time it takes to make each part and improving product quality. "If we can reduce our manufacturing cycle time, it's a huge deal for our customers—and for us," Rogers says.

To achieve that goal, Noble plans to use Oracle Internet of Things Asset Monitoring Cloud Service, a software-as-a-service application built on Oracle Internet of Things Cloud Enterprise that monitors assets and provides real-time visibility into their health and utilization, facilitating predictive maintenance. Rogers also plans to integrate that application with Oracle Mobile Cloud Service to achieve another of Noble's goals: a more robust alerting system that informs the company's techni-

In addition to using IoT technology to monitor and improve the manufacturing processes at his company, Noble Plastics Technical Director Scott Rogers sees opportunities to offer enhanced data analysis and other capabilities to customers of Noble's automation systems.

cians when problems occur, so that they can respond quickly.

Rogers aims to use Oracle's cloud-based analytics tools, including machine learning, to automatically analyze information gathered from the robot and process-monitoring systems. These analytics could help Noble identify ways to reduce cycle time, improve the manufacturing process, enhance product quality, and cut downtime. In addition, Rogers says, "we are hoping to be able to take that information and push it back down to the robot, so that it can

> **“If we can reduce our manufacturing cycle time, it’s a huge deal for our customers—and for us.”**
>
> —**Scott Rogers,** Technical Director, Noble Plastics

make even more-intelligent decisions.”

For Noble, the fact that Oracle’s IoT services are cloud-based is crucial. With fewer than 30 employees, the company can’t justify adding specialists dedicated to managing and updating on-premises IT systems or to analyzing data. With cloud-based software, the vendor is responsible for ensuring a high level of availability and keeping applications up to date, Rogers notes.

Cloud-based software also simplifies the process of monitoring and managing Noble’s two manufacturing locations. The machine learning capabilities allow Noble to identify trends and patterns in the data without having to do time-consuming manual analysis. In addition, Rogers sees opportunities to offer enhanced data analysis and other capabilities to customers of Noble’s automation systems.

We’ve Only Just Begun

As Lochbridge and Noble Plastics demonstrate, enterprise IoT involves much more than gathering information from connected devices. The ability to analyze and integrate the data with data in other applications is critical.

“Eventually, the number [of connected devices] is going to be so high that the amount of data coming from these machines will be just mind-boggling,” Oracle’s Gaur says. “And what we do with that data—how we take that data up and work it into actionable information—is the real goal.”

Oracle Internet of Things Cloud Enterprise enables high-speed data transfer and real-time analysis, but it relies on other applications to drive further actions, he says. As a result, companies are typically taking advantage of the platform’s built-in integration with other applica-

tions and services, such as enterprise resource planning and customer relationship management software, Oracle Mobile Cloud Service, and Oracle Big Data Cloud Service.

As Lochbridge CEO Bahl sees it, the integration of IoT into the enterprise holds enormous possibilities that businesses are just beginning to explore. “What’s really exciting about all of this is that the journey on monetization of IoT has just begun,” he says. “The next 10 years will be hugely exciting in terms of the insights companies can gain into their end consumers and how they use those insights.”

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

PHOTOGRAPHY BY **BLAKE DISCHER** AND **BEN DEPP/THE VERBATIM AGENCY**

NEXT STEPS

LEARN more about Oracle Internet of Things Cloud Enterprise.

LEARN more about Oracle Internet of Things applications.

TRY Oracle Internet of Things Cloud Enterprise.

Sponsored Content

Announcing the New Fujitsu SPARC M12 Servers

World's #1 Core Performance, Extreme Flexibility, and Mainframe-Class RAS

The new Fujitsu SPARC M12 server, featuring the 12-core SPARC64TM XII processor, delivers up to 2.5 times faster core performance and saves on total cost of ownership (TCO). The success of IT in delivering mission-critical applications depends on having a platform with blazing performance, unprecedented reliability, and scalability that enables smooth growth. Better performance includes flexibility that allows customers to optimize license costs by running their current software stack on fewer cores.

The new Fujitsu SPARC M12 servers meet modern enterprise computing demands. The Fujitsu SPARC M12 delivers extreme performance, mainframe-class reliability, availability, and serviceability (RAS), and optimum scalability for mission-critical workloads. Delivering immense computing capacity in a compact size, Fujitsu uses a flexible building block approach combined with a unique core activation feature to scale the systems incrementally up to 32 processor sockets, hosting up to 32 TB of memory. Fujitsu SPARC M12 supports Oracle Solaris 10 and 11 (bare metal and virtualized), and benefits from the Oracle Solaris Binary Application Guarantee to provide compatibility with past and present SPARC/Solaris servers.

The SPARC64 XII processor with advanced Software on Chip (SWoC) acceleration for database and in-memory processing, combined with the world's most powerful processor core, makes the Fujitsu SPARC M12 an ideal system for enterprise workloads. Moreover, robust RAS features make Fujitsu SPARC M12 the ideal foundation for a highly scalable mission-critical cloud, delivering outstanding business value with low total cost of ownership.

Smooth Scalability to Grow at Your Own Pace

Scalability is critical for businesses that need to allow for future growth while achieving superior performance today. Dynamic, flexible growth also allows businesses to save money – buy only what fits organizational needs and budget. The Fujitsu SPARC M12 features unique dynamic scaling to grow in lockstep with the business through granular core activation that helps address the challenges of unpredictable data growth. A unique Building Block Architecture scales easily and economically, combining up to 16 Building Blocks for a total of 32 processor sockets and 384 cores.

With the unique core activation feature, customers can activate cores as needed and dynamically scale from 2 cores to 384 cores at their own pace, enabling rapid and cost-effective scalability. The Fujitsu SPARC M12 offers massive and flexible I/O scalability (up to 928 PCI Express Gen 3 slots in Fujitsu SPARC M12-2S) using PCI Expansion Units that enable large scale consolidation where dedicated I/O resources are required.

NEW SPARC MODELS DELIVER THE PERFORMANCE YOU NEED

Fujitsu SPARC M12 servers are available in two models that deliver the performance and scalability to meet all organizational needs.

- The Fujitsu SPARC M12-2 features up to two 3.9 GHz, 12-core processors with 8 threads per core for a total of up to 24 cores and 192 threads per server. The 4U chassis supports up to 2 terabytes of DDR4 memory.
- The Fujitsu SPARC M12-2S features two 4.25 GHz processors in a 4U chassis that can be combined with other Fujitsu SPARC M12-2S servers using Fujitsu Building Block Architecture to form a 16 Building Block configuration offering up to 32 processors, with 384 cores and 3,072 threads. Memory can scale up to 32 terabytes of DDR4 memory.

Record-Setting Performance for the Most Demanding Modern Workloads

At the heart of the Fujitsu SPARC M12, the innovative SPARC64 Xll processor has already captured 15 world records across a wide range of industry-standard benchmarks. With record-setting performance, the Fujitsu SPARC M12 helps organizations manage the explosive growth of data with cutting-edge technology for faster database and application processing.

		Per-Core Performance	Socket Performance
#1 Arithmetic Performance	SPECint_rate 2006	1	1
#1 Scientific Performance	SPECfp_rate 2006	1	1 1 1 1
Faster memory Throughput	STREAM TRIAD	Not Applicable (core count does not impact STREAM results)	1 1 1
#1 Java Performance	SPECjbb2015 max-jOPS (MultiJVM)	1	
	(Composite)	1	
Extreme Response Performance	SPECjbb2015 critical-jOPS (MultiJVM)	1	
	(Composite)	1	1

Please see the Fujitsu SPARC benchmark page for the benchmark results as of April 4, 2017.

Mainframe-Class RAS Ensures Continuous Uptime

Migrating workloads can be risky business for critical applications with zero tolerance for downtime. The Fujitsu SPARC M12 delivers mainframe-class RAS across the entire system, from the SPARC64 XII processors and memory to every subsystem. The system can perform a variety of actions to protect itself and your critical applications and data — from dynamically degrading the minimum faulty component to proactively looking for errors.

Fujitsu SPARC M12 servers use innovative processor and system design to ensure high reliability and high availability for mission-critical applications. In addition, comprehensive and exhaustive data protection and redundancy assures system uptime 24 hours a day, 365 days a year.

Innovative Cooling for Server Density and Performance

Fujitsu SPARC M12 adds innovation in cooling technologies, too. The new Vapor and Liquid Loop Cooling (VLLC) system found in the Fujitsu SPARC M12 provides twice the cooling performance over the Liquid Loop Cooling (LLC) system used in Fujitsu M10 servers. VLLC achieves significantly increased cooling performance through the phase change of liquid to vapor, allowing superior heat absorption. VLLC in the Fujitsu SPARC M12 are sealed-for-life units with no external hoses or chillers. And because VLLC alleviates the need for heatsinks, CPUs and memory can be packed more closely together, reducing latency and helping to make the Fujitsu SPARC M12 the most advanced server for mission-critical enterprise applications.

Software on Chip Delivers Performance and Security

Fujitsu SPARC M12 servers feature core-based advanced SWoC technology to accelerate performance and enhance security. The power of Fujitsu's SWoC technology helps control costs and deliver value by ensuring that businesses

don't have to choose between security and performance. With Fujitsu SPARC M12, an organization gets both.

For example, in-memory query acceleration and in-line decompression deliver dramatic improvements for in-memory databases and allow both analytics and OLTP transactions to be performed on the same data. Also, encryption acceleration enables end-to-end security with near-zero overhead. Encrypt everything, all the time, at full speed. In addition, decimal and NUMBER acceleration speed database calculations.

Mission-Critical Infrastructure for Private Cloud

Many organizations today need the security and privacy of private cloud. With Fujitsu SPARC M12 servers, companies get record-setting core performance, mainframe-class RAS, and on-demand scalability for future growth—delivered from your private datacenter. Business critical applications are the perfect fit for the Fujitsu SPARC M12 powering your private cloud, too. They can be delivered on Oracle Solaris 10 or Oracle Solaris 11, the world's most advanced enterprise operating system with security, speed, and simplicity for enterprise private cloud.

Private cloud management is readily available, too. OpenStack is integrated in the robust Oracle Solaris operating system to deliver open standards-based private cloud. And with Oracle Enterprise Manager Cloud Control and Oracle Enterprise Manager Ops Center, IT gets a robust set of software tools to build and manage a private cloud using familiar Oracle technology.

> "Oracle and Fujitsu have worked together for more than three decades to produce SPARC systems satisfying the demanding requirements of mission-critical infrastructure. Fujitsu SPARC M12 servers featuring the new SPARC64 XII processor for extreme core performance are an exciting addition to the SPARC family, allowing customers to address their most difficult computing challenges with systems that offer both high performance and enterprise reliability."
>
> —Edward Screven
> Chief Corporate Architect of Oracle Corporation

Fujitsu and Oracle – A Powerful Partnership

The SPARC M12 brings together the SPARC64 XII processor, Fujitsu's advanced system technology, and the most powerful enterprise operating system, Oracle Solaris 11. Together, Oracle and Fujitsu continue to deliver customers unmatched value with the Fujitsu SPARC M12 through record-breaking performance, reliability, and scalability for the modern enterprise.

FUJITSU

For more information, visit www.fujitsu.com/sparc

Ian Dunbar, CEO at SuiteBox, describes partnering with Oracle in the cloud as a guarantee of credibility with customers.

MORE AS THINGS DEVELOP

SuiteBox builds technology on Oracle Cloud to revamp the virtual workplace with video meetings and paperless transactions. **BY ALEXANDRA WEBER MORALES**

If you want to revolutionize the virtual workplace market, a cloud-native approach makes perfect sense—and lets you refactor your approach until you get it right. When New Zealand–based SuiteBox launched six years ago, it started out supplementing annual company meetings with online video. But a few years in, the startup had the foresight to pivot. The new target? A market that Skype, FaceTime, Google Hangouts, DocuSign, and WebEx had missed: virtual meetings in the financial world, providing broad (and legal) identity authentication via real-time digital signatures and video.

SuiteBox built a financial-grade communications and collaboration solution on Oracle Cloud technology that makes it possible for financial advisers to hold virtual meetings via connected devices, record all or part of their encounters, share and digitally sign electronic assets, and embed or white-label the interactions for any relationship management platform.

Cloud-Native and Global from the Start

SuiteBox focused on using the internet to think global from day one. The key idea: to meld video meetings with paperless transactions. The choice to go global was a natural one, but SuiteBox had other requirements as well: a cloud-native software stack and financial-caliber components.

In addition to scalability and bank-level security, the goal of deploying straight to the cloud attracted SuiteBox to Oracle technology—but it was an unusual move, according to Craig Meek, founder of SuiteBox.

"It was not an easy decision for a startup company to choose Oracle instead of open source, because there were a lot of people saying that would be a very expensive journey," says Meek. "It was kind of like

SUITEBOX

HEADQUARTERS:
Auckland, New Zealand

INDUSTRY:
High technology

ORACLE PRODUCTS:
Oracle Java Cloud Service
Oracle Developer Cloud Service
Oracle Database Cloud Service
Oracle SOA Cloud Service
Oracle Traffic Director
Oracle Storage Cloud
Oracle Cloud Machine

SuiteBox Founder Craig Meek views choosing Oracle as part of the company's journey to becoming a global player in the financial services space.

choosing Rolls-Royce first, but we realized that if we were truly going to be a global player in the financial services space, we had to make that call. When we stand in front of the CIOs and go through this technology stack, they give us their business."

Ian Dunbar, CEO at SuiteBox, agrees. "We preferred Oracle platform-as-a-service [technology] because of its robustness, scalability, and bank-grade security features," he says. "Partnering with a reputable vendor such as Oracle is also an automatic guarantee of

> **“We preferred Oracle platform-as-a-service [technology] because of its robustness, scalability, and bank-grade security features.”**
>
> **—Ian Dunbar,** CEO, SuiteBox

credibility, giving our customers absolute confidence.”

Choosing the Oracle Cloud Stack

For its logic tier, SuiteBox, whose customers range from small businesses to global banks and insurance companies, is using Oracle Java Cloud Service to provide easy implementation and integration with video and Voice over IP (VoIP) communication layers.

Oracle Cloud Database Service stores SuiteBox communication metadata in a simple and efficient database structure that links users to their meetings and their storage. The integration layer is enabled via Oracle SOA Cloud Service, which lets SuiteBox REST-ify existing and new services as JSON APIs.

That’s an additional selling point, according to Isaac Young, chief technology officer at SuiteBox. SuiteBox provides its customers with a breadth of programming interfaces for advisor platforms, enterprise resource planning, mobile apps, and customer relationship management systems. Its PDF-authoring API, for example, allows third-party developers to merge multiple PDF documents and to place images (signatures, for example) on pages at specific X and Y coordinates. Its middleware APIs let those developers work with users, authentication, meetings, documents, and more.

“We have a set of APIs that we built using the Oracle suite of tools,” says Young. “Our RESTful JSON API is predicated on Oracle technologies, such as Oracle API Gateway.”

Another Oracle solution, Oracle Traffic Director, optimizes SuiteBox’ application-to-application communication for high throughput and protects back-end applications from malicious attacks.

“Oracle Traffic Director is a really cool feature that allows us to ensure that we have business

SuiteBox chief technology officer Isaac Young sees the value of engaging with Oracle early in new cloud projects.

continuity in our service," Young says. "We also use it to whitelist and blacklist a client's IP addresses and allow only specific partners to access our API. It's fundamental throughout SuiteBox' architecture."

Key Oracle developer services enable SuiteBox to optimize development processes. SuiteBox uses Oracle Developer Cloud Service to refactor applications to meet new customer requirements and Oracle Java Cloud Service to

rapidly provision testing and production environments. The company also counts on Oracle SOA Cloud Service to ensure seamless integration with third-party applications, enabling its customers to easily integrate with SuiteBox.

Future Plans: More Oracle Cloud, AI, and BI

SuiteBox currently uses Amazon Web Services for hosting, but that will soon change, according to Meek. "We're now doing a full migration to Oracle Cloud, to clean up the whole architecture from top to bottom," he says. "We didn't really appreciate at the time we launched that Oracle Cloud would be so competitive. Now we're very comfortable in terms of stability with the way in which the platform is operating, and it's giving us a lot of credibility with our customers."

More insights into utilization and productivity are on the horizon as well. "We're looking at Oracle Business Intelligence Cloud Service and specifically the cloud analytics services to provide transparency to our customers around how they're using SuiteBox," says Young. "How many meetings are they having? How many documents are being signed? What browsers are people using?"

The most important lesson for success, according to Young, is that you don't have to go it alone. "One of the key things we learned is, engage with Oracle early when embarking on a new project or on using Oracle technology that you're not familiar with," he says. "Oracle was pivotal in helping us deliver and succeed in what we're trying to achieve."

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

PHOTOGRAPHY BY **LISA MAREE WILLIAMS/PAUL SUTHERLAND/THE VERBATIM AGENCY**

NEXT STEPS

LEARN more about Oracle Cloud Platform.

LEARN more about Oracle Cloud infrastructure.

Getting the Most Out of Your Oracle ERP Investment

A single enterprise information platform for managing your content, processes, and cases.

Bill Filion, vice president of software development for OnBase by Hyland, discusses how OnBase can help Oracle ERP customers be more agile, efficient, and effective.

Q: Why is OnBase an important solution for Oracle ERP customers?

A: As a single enterprise information platform for managing content, processes, and cases, OnBase makes Oracle ERP users more efficient and productive. OnBase works with Oracle E-Business Suite, PeopleSoft and JD Edwards to provide one-click access to needed documents. Our platform improves productivity, reduces costs, and facilitates straight-through processing from your Oracle ERP. Oracle customers make heavy investments to design, configure, install, and gain experience with their ERP systems. They need to get the most out of these investments. With OnBase, we offer many ways to quickly extend ERP systems to fully manage any process. OnBase makes it painless to add new functionalities such as reviews, legal aspects, end user feedback, and more.

Q: How does a single enterprise information platform help Oracle ERP customers be more agile, efficient, and effective?

A: OnBase is the middleware designed to fit within existing technology stacks and business systems. We compare it to the glue between business systems. This makes it easy for an Oracle ERP user to stay within the application where they are most productive, but get increased ERP functionality without writing custom code. With our platform, users can easily craft new solutions and seamlessly incorporate them into the ERP.

We connect critically-related content, such as invoices, packing slips, POs, transcripts, mail, and email into business processes. The user stays within their primary business application and this content instantly becomes presented as part of that system.

Q: How does OnBase speed up processes and increase productivity by capturing important information into one system?

A: Most medium-sized organizations utilize approximately 150 business systems; large enterprises can use up to 1,000. OnBase is designed to connect these systems to extend the ERP. When these highly configurable connections are made, OnBase responds to the related systems in fractions of a second. This eliminates manual transfers between systems and connects to all relevant information, making auditing easier and greatly increasing productivity.

Q: Why do Oracle ERP customers choose OnBase to design solutions specific to their needs?

A: OnBase makes it easy to design configurable solutions with any Oracle ERP system. OnBase can be configured at multiple places within the ERP, but also for processes that are outside it, involving any of the other systems in the business. Even legacy manual processes that developed as the business grew can be converted to electronic content and integrated as extensions of the ERP.

Q: Once deployed on Oracle ERP, how is OnBase unique in the marketplace?

A: OnBase is a platform without parallel. With each yearly release, we deliver an additional 2,000 to 5,000 enhancements – most of these new features come in response to customer needs. Also, OnBase is not a stagnant platform. Our ability to integrate with business systems is exceedingly broad and we are always looking for ways to incorporate more types of content. OnBase eliminates integration pains through codeless, point-and-click integration and rapid deployment options. The solution has hundreds of pre-defined rules and actions available from a simple drop-down menu, to readily configure complex workflows without any scripting, coding, or programming. Finally, OnBase is budget-friendly as our solutions are highly scalable.

> “OnBase makes it easy to design configurable solutions with any Oracle ERP system.”

For more information, visit **OnBase.com/Oraclemag.**

OnBase® by Hyland

By Chris Muir

ORACLE MOBILE CLOUD SERVICE

Fail Fast, Fail Often

Use Oracle Mobile Cloud Service's Express API to rapidly prototype REST APIs.

Over the last decade, REST APIs have made an amazing transition in IT, becoming a de facto standard for intersystem communications on disparate platforms. The ability of REST APIs to transmit resources and their related data over HTTP has led to all sorts of uses, including web, mobile, and the Internet of Things (IoT).

Since its launch, Oracle Mobile Cloud Service has supported REST APIs as its primary channel of communication, in both producing and consuming REST APIs. Although other channels, such as SOAP, are also supported, REST is the preferred channel for developing mobile as well as web applications via Oracle Mobile Cloud Service.

In Oracle Mobile Cloud Service's first incarnation, developers could build REST APIs by defining URL patterns for REST resources and then defining the HTTP request verbs—that is, GET/PUT/POST/PATCH/DELETE operations—for each resource

and finally defining the responses. This process allowed for a very fine-grained approach to developing REST APIs.

However, Oracle customers and teams building APIs with Oracle Mobile Cloud Service were interested in a faster approach to building REST APIs, not just for production purposes but also for fast API prototyping and testing.

Thus was born Oracle Mobile Cloud Service's Express API, a new approach to building APIs in a declarative fashion. Simply put, with the Express API, you define a resource such as an employee or a job role, you supply some data examples, and the Express API generates the rest of the API specification for you automatically.

This article describes how to create a REST API via Oracle Mobile Cloud Service's Express API, using employees and their job roles as the sample data model.

PREREQUISITES

To follow the steps in this article, you will need access to Oracle Mobile Cloud Service, which you can obtain via the **Free Trial** button on the Oracle Mobile Cloud Service home page. (You have the option to sign up for a trial account or free cloud credits.) After signing up for the trial and receiving approval, watch and follow the instructions in the video on how to set up and provision your Oracle Mobile Cloud Service instance.

You will also need to create a developer account with the **Mobile Environment Service Base Entitlement Administrator**, **Identity Domain Administrator**, and **<environment> Mobile Team Member** roles as well as all the roles for the "mobile app developer and service developer" in the Development Environment Roles column, as documented in the "Example Team Member Role Assignments" section of *Using Oracle Mobile Cloud Service*.

Once you've created the account, log in to Oracle Cloud and, via the My Services dashboard, select **Mobile Environment Service** and click **Open Service Console**, which will take you to the Oracle Mobile Cloud Service user interface.

For the purposes of testing, you will need to create one mobile back end in Oracle Mobile Cloud Service first:

1. Select the icon (hamburger) at the top left, and in the menu that opens, expand the **Applications** option and select the **Mobile Backends** option. Click **New Mobile Backend**.
2. In the dialog box, enter a **Mobile Backend** name of `OraMag` and click **Create**.
3. On the **Mobile Backend** screen, click the **API** option.

BUILDING OUR API FOR EMPLOYEES

Now you are going to quickly build an API that represents two resources: **Employees** and their **Job** roles. You'll build the API so that it returns mock data, because building an implementation in NodeJS is outside the scope of this article.

Clicking the arrow on the **New API** button gives you two options for building REST APIs via Oracle Mobile Cloud Service: the traditional API and the Express API. For this article, you will use the **Express API** option.

4. Click the **New API** button drop-down arrow, and select **Express API**.
5. In the resulting dialog box, for **API Display Name** and **API Name** enter `HR`. For **Short Description**, enter something of your choosing. Note the URL through which your new HR API will be publicly exposed. Click **Create**.
6. The resulting screen is where you define the bulk of the API, including its resources and implementation. Click **Resources**.
7. Click **New Resource**.
8. For the first step in the wizard, enter `Employee` (with a capital *E*) for **Display**

Name, enter `employee` (with a lowercase *e*) for **Name**, and enter `Employees` (with a capital *E*) for **Display Name (plural)**. Click **Next**.

9. The Express API enables you to provide a JSON payload of data that the wizard will use to work out the elements and datatypes of the resources. Click **Add Sample Data**, and then copy and paste the following JSON payload into the **Sample Data** field:

```
[{"employeeId":100,"firstName":"Steven","lastName":"King",
"hireDate":"1987-06-17","job":{"jobId":"AD_PRES",
"jobTitle":"President"}},
{"employeeid":101,"firstName":"Neena","lastName":"Kochhar",
"hireDate":"1989-09-21","job":{"jobId":"AD_VP",
"jobTitle":"Vice President"}},
{"employeeId":102,"firstName":"Lex","lastName":"De Haan",
"hireDate":"1993-01-13","job":{"jobId":"AD_VP",
"jobTitle":"Vice President"}},
{"employeeId":103,"firstName":"Alexander","lastName":"Hunold",
"hireDate":"1190-01-03","job":{"jobId":"IT_PROG",
"jobTitle":"IT Programmer"}}]
```

Note how the employee values include not only elements particular to them but also elements for the related **Jobs** resource. You will see the effect of this in a minute. With the JSON payload pasted into the **Sample Data** field, click **Next**.

10. The Fields step shows you the fields and their datatypes that the wizard has determined from the example payload, such as integers for the employee IDs and strings for the names. Note also the addition of a new field: **Id**. By default,

the wizard will create its own unique identifier for each record—what relational database experts refer to as a *primary key*. Although REST doesn't have a primary key concept, it is common in REST web services to want to return a single object identified by some arbitrary identifier. Click **Next**.

11. The final wizard screen, captured in **Figure 1**, shows all the work the Express API wizard will undertake for you, from automatically creating a GET for the employees collection to return all employees to creating GET/PATCH/DELETE/POST operations for a specific employee identified by the previously mentioned **Id** field.
12. Click **Finish**, and then click **Save**.
    The resulting screen shows the details of the Employee resource you just created. You can now investigate the separate resource options, and later you can modify the resource as you see fit. For example, click the **Sample Data** tab and see how Oracle Mobile Cloud Service has imported the sample data into the individual fields as well as generated its own unique ID value for each employee. Click the **Employee Overview** tab,

Figure 1: The generated REST API endpoints

Path	Method	Method Name	Description
/hr/employee	GET	List Employees	Returns a collection of Employees
/hr/employee	POST	Create Employee	Creates a new Employee via POST
/hr/employee/{id}	GET	Find Employee	Returns a single Employee
/hr/employee/{id}	PATCH	Edit Employee	Updates an existing instance of Employee
/hr/employee/{id}	DELETE	Delete Employee	Delete an instance of Employee
/hr/job	GET	List Job	Returns a collection of Job
/hr/job	POST	Create Job	Creates a new Job via POST
/hr/job/{id}	GET	Find Job	Returns a single Job
/hr/job/{id}	PATCH	Edit Job	Updates an existing instance of Job
/hr/job/{id}	DELETE	Delete Job	Delete an instance of Job

and note the sections **Child Resources** and **Reference Resources**, which includes **Job**. As highlighted earlier, when you entered the sample data for employees, each employee had a related job role embedded in the sample payload. Oracle Mobile Cloud Service has determined that **Job** is another resource and has created the APIs.

Child Resources are used to describe an ownership relationship, such as a department having employees. A Reference Resource represents a "type-of" relationship, such as Employee being a type of Salesman, which is a Job role. For relational database experts, this is master-detail relationship modeling.

Note that because **Job** is listed under **Employee** in the resource list, you can select **Job** to investigate what is available for **Job** in the API. For example, on the **Method** tab, you can look at the methods that have already been created for **Job**.

TESTING YOUR APIs

With your Express API set up, you can now start testing the API (even though you haven't implemented the Node.js code yet). This testing is handy to make sure you have the API signature correct, even when you don't yet have the implementation details ready.

13. Click the **Test** button.

On the test screen, down the left side, you will see all the REST API endpoints you can call. The test facility is capable of returning dummy responses for all the endpoints, and for **GET List Employees** and **GET List Job**, the text will even return the sample data you entered in the wizard. Let's try this out:

14. Click **GET List Employees**. For **Mobile Backend**, click **OraMag**. Next click **Test Endpoint**. Note the sample employee data returned, including the reference **Job** data.

15. Now click **GET List Job**; for **Mobile Backend**, again click **OraMag**; and then click **Test Endpoint** to see the sample **Job** data.

CONCLUSION

With Oracle Mobile Cloud Service's Express API, some broad ideas of what resources you need, and some quickly typed-up sample data, you can quickly and easily mock up live REST APIs. This means that you can start testing via the Oracle Mobile Cloud Service UI test pages—or even real remote mobile and web applications—to see the functions you have built. As the saying goes, "Fail fast, fail often." With the Express API, you can rapidly prototype an API from a shell of an idea to a working solution.

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.

JOIN the Oracle Mobile LinkedIn community.

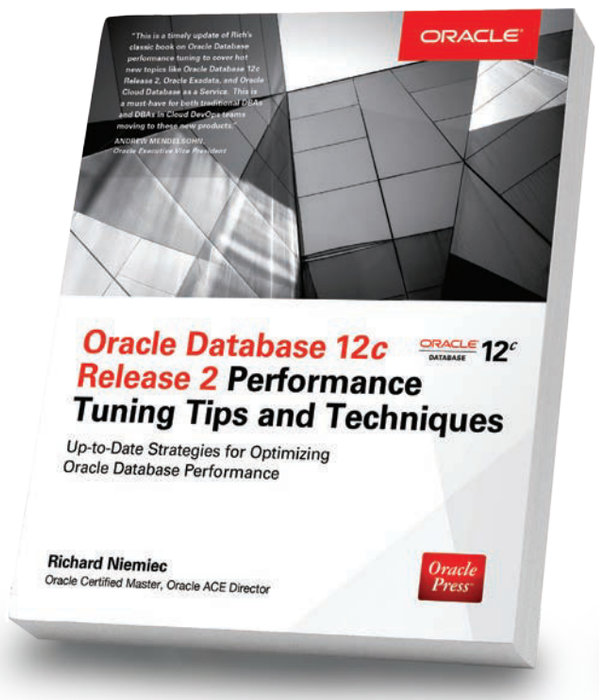
TRY Oracle Mobile Cloud Service.

WATCH Oracle Mobile Cloud Service YouTube training.

Oracle Press™

Your Destination for Oracle and Java Expertise

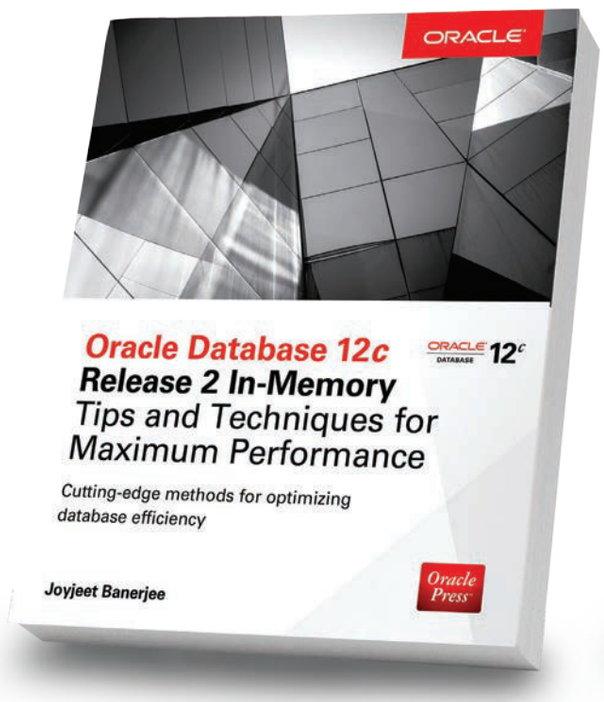
Written by leading technology professionals, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Oracle products and technologies available.



Oracle Database 12c Release 2 Performance Tuning Tips & Techniques

Richard Niemiec

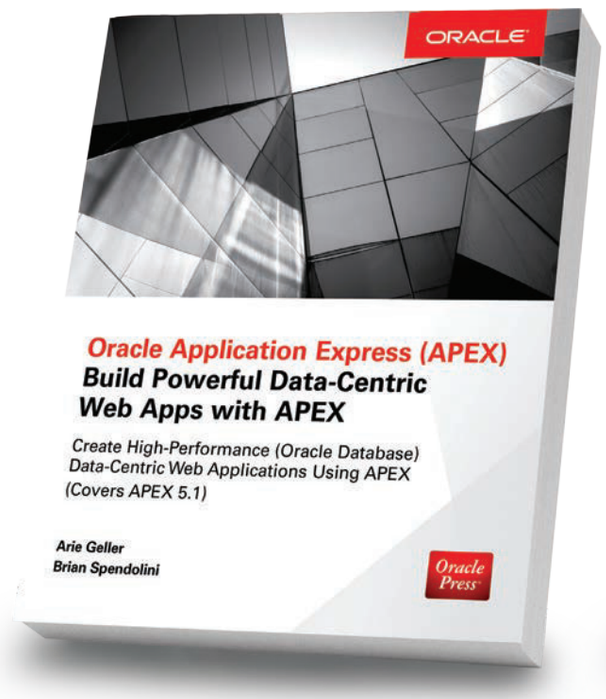
Proven database optimization solutions—fully updated for Oracle Database 12*c* Relaease 2.



Oracle Database 12c Release 2 In-Memory: Tips and Techniques for Maximum Performance

Joyjeet Banerjee

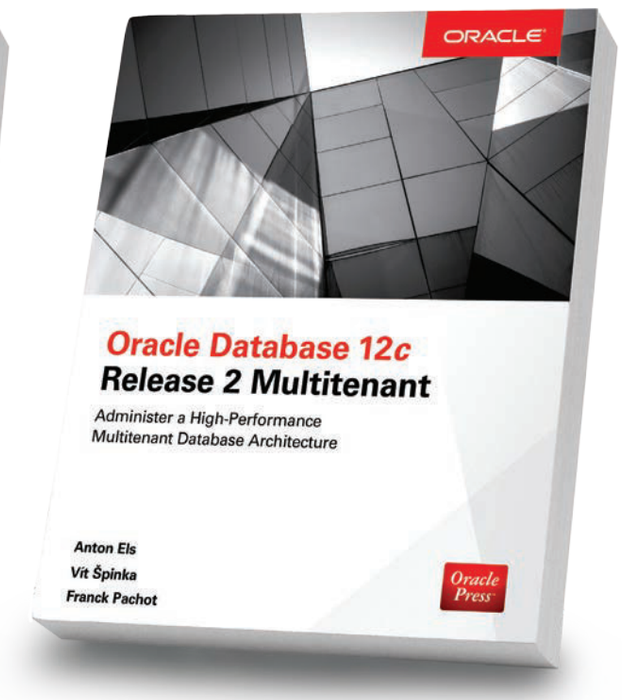
Master Oracle Database 12*c* Release 2's powerful In-Memory option.



Oracle Application Express: Build Powerful Data-Centric Web Apps with APEX

Arie Geller, Brian Spendolini

Develop robust, modern web applications with Oracle Application Express.



Oracle Database 12c Release 2 Multitenant

Anton Els, Vit Spinka, Franck Pachot

Master the powerful Multitenant features of Oracle Database 12*c*.

Available in print and eBook formats.

Oracle Press™

www.OraclePressBooks.com • @OraclePress

ORACLE APPLICATION BUILDER CLOUD SERVICE

Low-Code and On-Device

Build mobile apps with Oracle Application Builder Cloud Service.

By Shay Shmeltzer

In my previous *Oracle Magazine* article, "High on Power, Low on Code," I gave you an introduction to developing web applications with Oracle's low-code cloud development tool, Oracle Application Builder Cloud Service. I showed that Oracle Application Builder Cloud Service's responsive UI templates and the ability to customize specific items to display in specific ways on different devices makes it easy to deliver an application that runs across multiple devices with a variety of screen resolutions and sizes.

However, if you are aiming for an even better user experience for mobile users, you can create on-device mobile apps with Oracle Application Builder Cloud Service. These applications install and run on the device and provide a native look and feel, mobile-optimized gesture support, and the ability to leverage device features such as geolocation.

WHAT IS IT?

Oracle Application Builder Cloud Service provides an easy way for business users to create their own web and mobile applications and host them in a cloud environment. With its intuitive visual development interface and without prior development experience, creating engaging applications and reducing IT backlogs have never been easier.

In this article, I'll show you how Oracle Application Builder Cloud Service helps you create on-device mobile apps with ease. You'll build a simple expense reporting application that will enable you to monitor and add expenses directly from your device. In addition, this article's demonstration video shows the steps for building the application and the Oracle Application Builder Cloud Service features described in the article.

To walk through the article steps, sign up for a free trial account for Oracle Application Builder Cloud Service. Get your account at Oracle Cloud.

CREATING BUSINESS OBJECTS

The first thing you'll do is create a custom business object that will store your expenses data. To do this, create a new web application in Oracle Application Builder Cloud Service (click **New Application**), name it `Expenses`, and choose the defaults in the rest of the application creation wizard. With the application created, you'll be taken to the page designer, where you can design your web UI. Note that if you just want to create business objects, you can go directly to the data designer and define your objects there. For this article, however, you are going to create a web UI too.

Drag a collection table component into the page designer, and create a new business object for it. Name it `Expenses`. Define fields and datatypes for this object according to the following list:

- **Name** – text
- **Cost** – currency
- **When** – date

Click **Finish**, and click the run arrow (in the upper right) to run your web application.

You can now create a few expense entries, as demonstrated in the video. With the expense data entered, go back to the design environment.

To enable a mobile application—such as the one you will build next—to access the same Expenses business object, go to the main menu, select **Application Settings** -> **General Settings**, and enable resource sharing. This tells Oracle Application Builder Cloud Service that the business objects you created in this web application can also be accessed from other mobile applications that will be created. Any logic you define for the business objects will also be applicable automatically in your mobile UIs.

Next, stage and publish the application. From the main menu, select **Stage**. Choose to copy the development data you already inserted for both the staged and the published application, and click **Stage**. Again from the main menu, select **Publish**, choose to include data from Stage, and click **Publish**.

Now you are ready to create a device-resident mobile-optimized app that accesses the same data.

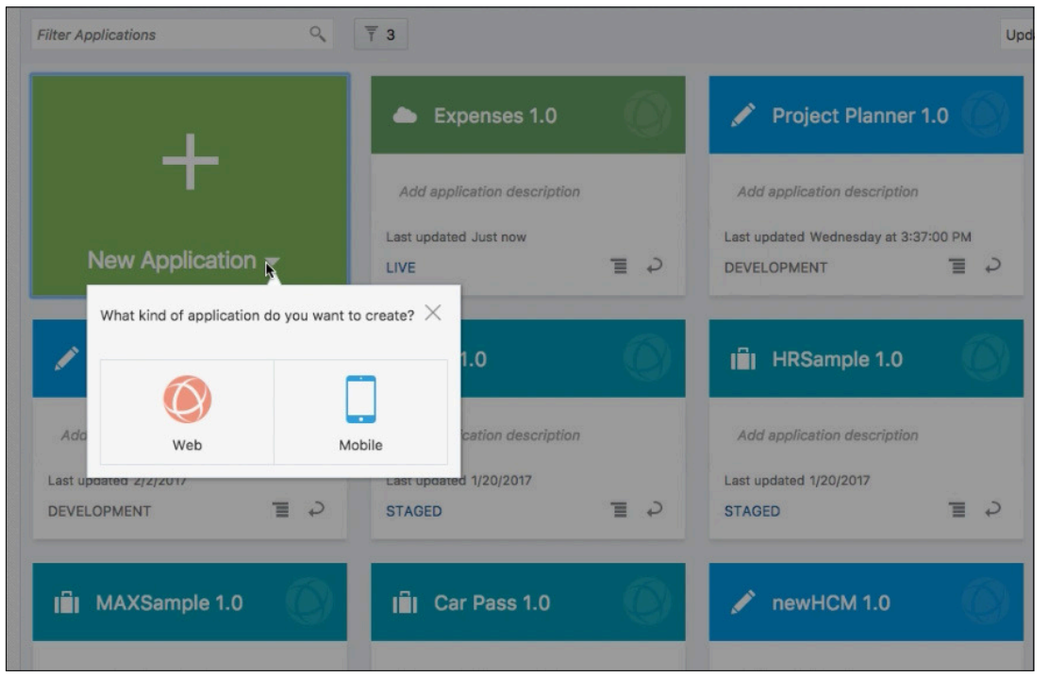


Figure 1: Creating the mobile app

CREATING A MOBILE-OPTIMIZED INTERFACE

Navigate back to the home page of Oracle Application Builder Cloud Service, and click **New Application**. This time choose the **Mobile** option, as shown in **Figure 1**.

The application creation wizard will guide you through the steps of defining your application. First choose a name for your application, such as `Mobile Expenses`.

Next choose the template you would like to use for the first screen of your application. Oracle Application Builder Cloud Service offers a set of mobile-optimized page templates for pages, based on design patterns created by

Figure 2: Choosing the Simple Screen template for the new mobile app

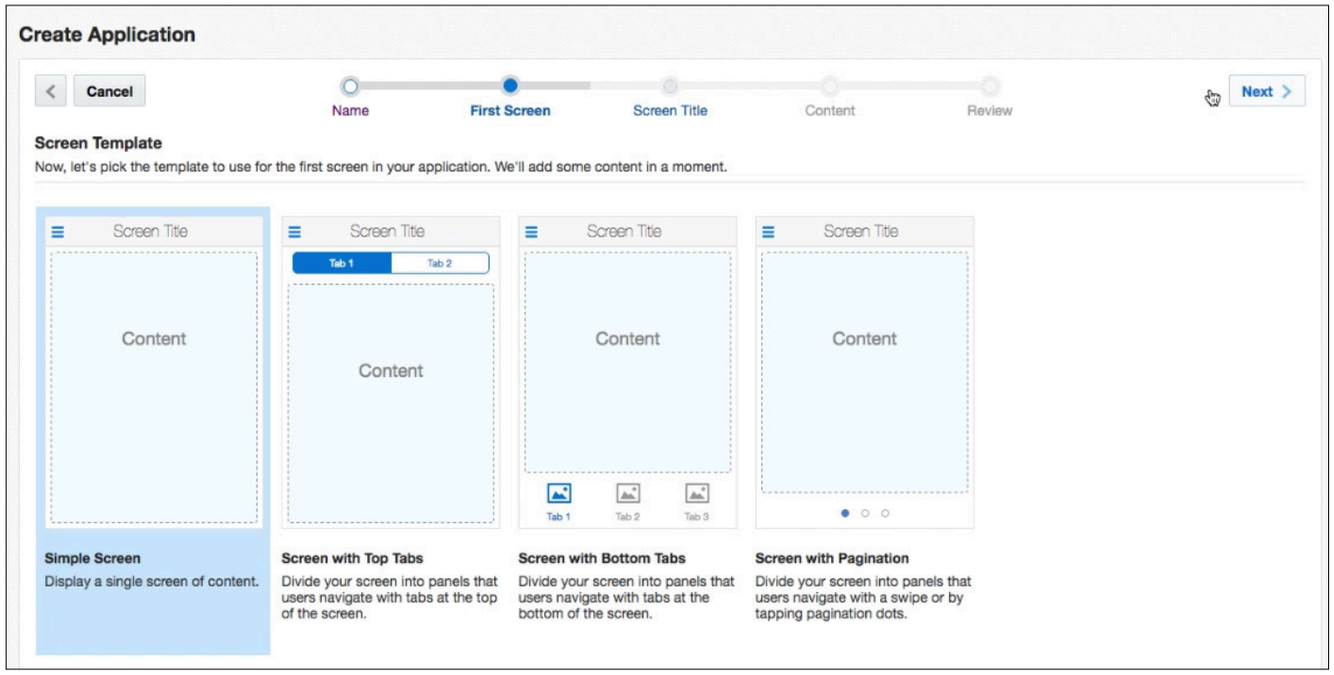
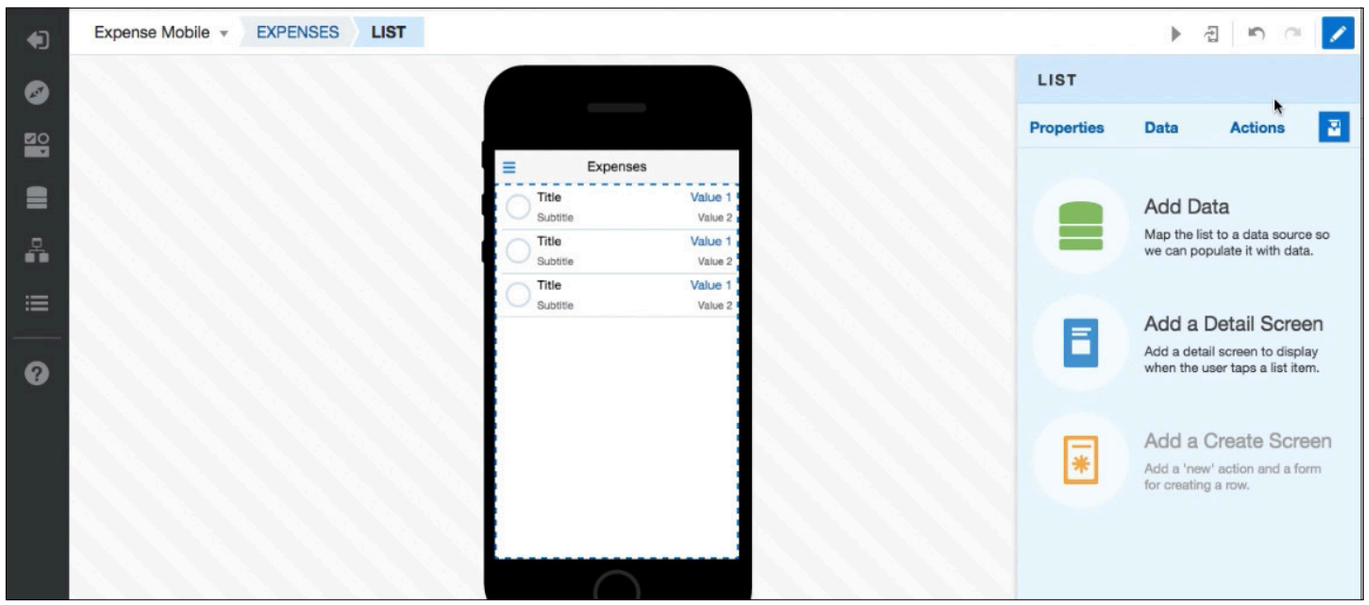


Figure 3: The newly created mobile app in the visual mobile UI designer



Oracle's usability group. Choose the **Simple Screen** template for the first page, as shown in **Figure 2**.

Enter `Expenses` as the **Screen Title** for this page. Then choose a layout for the page. You want the app to show a list of all the expenses, so choose the **List** layout.

Complete the wizard, and click **Create**. The visual mobile UI designer opens, as shown in **Figure 3**.

Just as in the web designer, the visual designer includes a "what you see is what you get" center area that shows your UI inside a mobile device. On the right, you'll see options to set properties, add data, and add related forms to your app.

Click **Add Data** on the right. You'll see various styles of the list component. Choose the simple title and subtitle list type, and click **Next**. Click **Add Application**, and then click the **Expenses** web application you created earlier, as shown in **Figure 4**. The Expenses application includes the business object you want to show in the on-device mobile app.

Figure 4: Choosing the Expenses web app

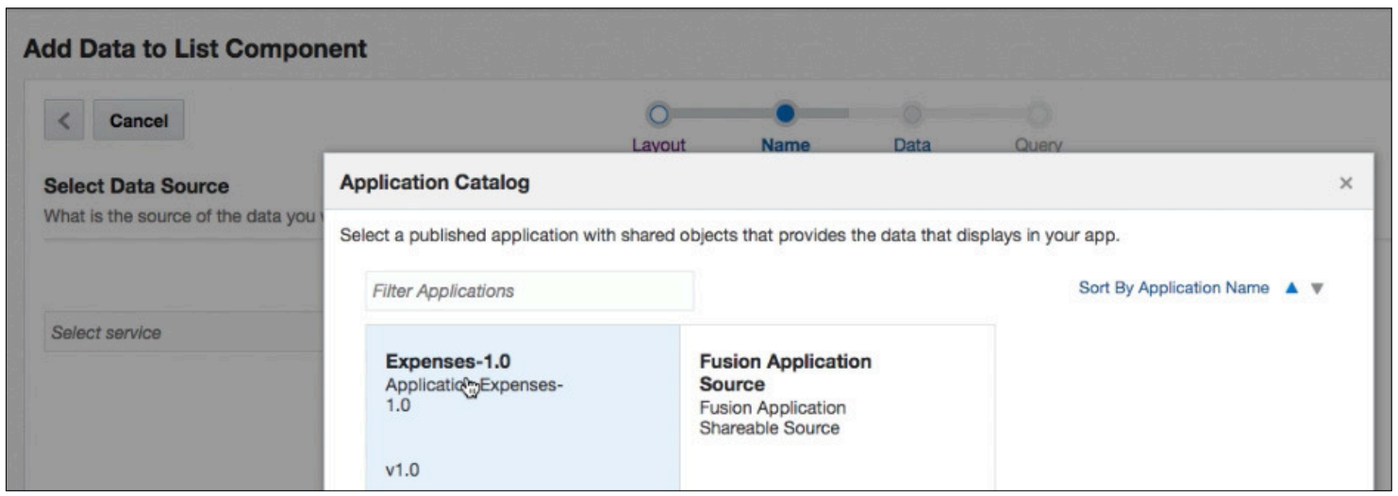
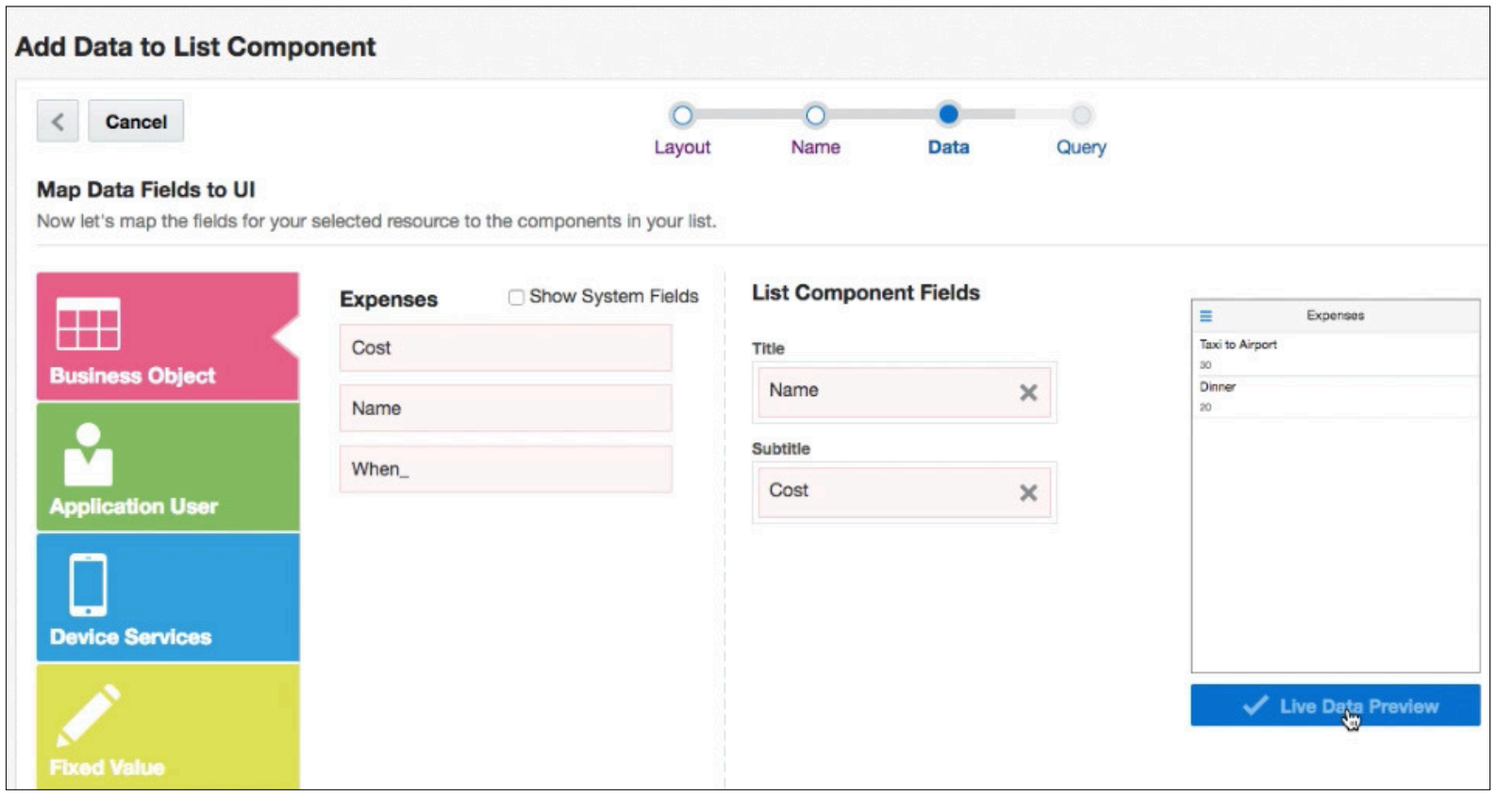


Figure 5: Selecting the title and the subtitle



Now choose the fields you want to show as the title and the subtitle—choose **Name** and **Cost**, respectively—as shown in **Figure 5**. You can click the **Live Data Preview** button on the right to see a preview of how the actual data will look in the app.

Click **Next** and **Finish**. You can now see your page with the live data in the visual editor.

Now create a detail screen where you can see more information about a specific expense report.

Click **Add a Detail Screen** on the right, and then in the following steps, choose the **Simple Screen** template, enter a title—`Expense Details`—for the page, choose the **Form** screen type, and click **Finish**. The wizard adds the detail screen and links it to the list's finger-click gesture automatically.

Select the page navigation hierarchy button on the left to see the connection between the two pages,

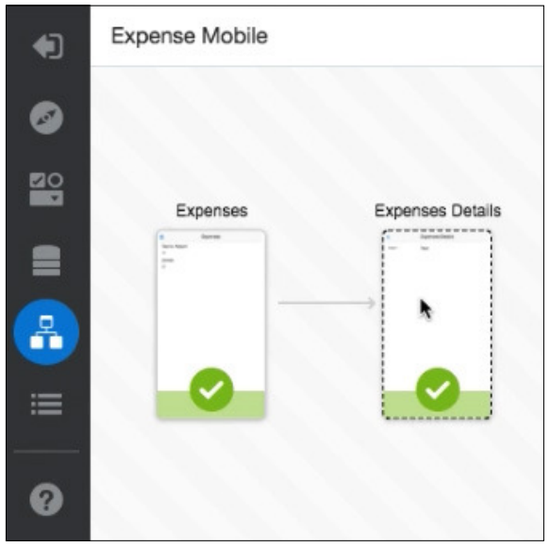
as shown in **Figure 6**. Then double-click the **Expenses Details** page to refine its content in the visual editor.

Click **Add Data** on the right, choose the **Expenses** object, and shuttle the fields you want to display in the order you prefer—as shown in **Figure 7**—and on the last page of the wizard, click **Finish**.

Click the page navigation hierarchy button on the left to see the connection between the two pages, and click the **List (Expenses)** page. Click **Add a Create Screen** on the right to add a Create screen for the application. You can learn how to further refine this page and add more customizations, such as tabs and charts, by

Figure 7: Adding data fields to the Expenses Details form

Figure 6: Viewing the page navigation and selecting the Expenses Details page



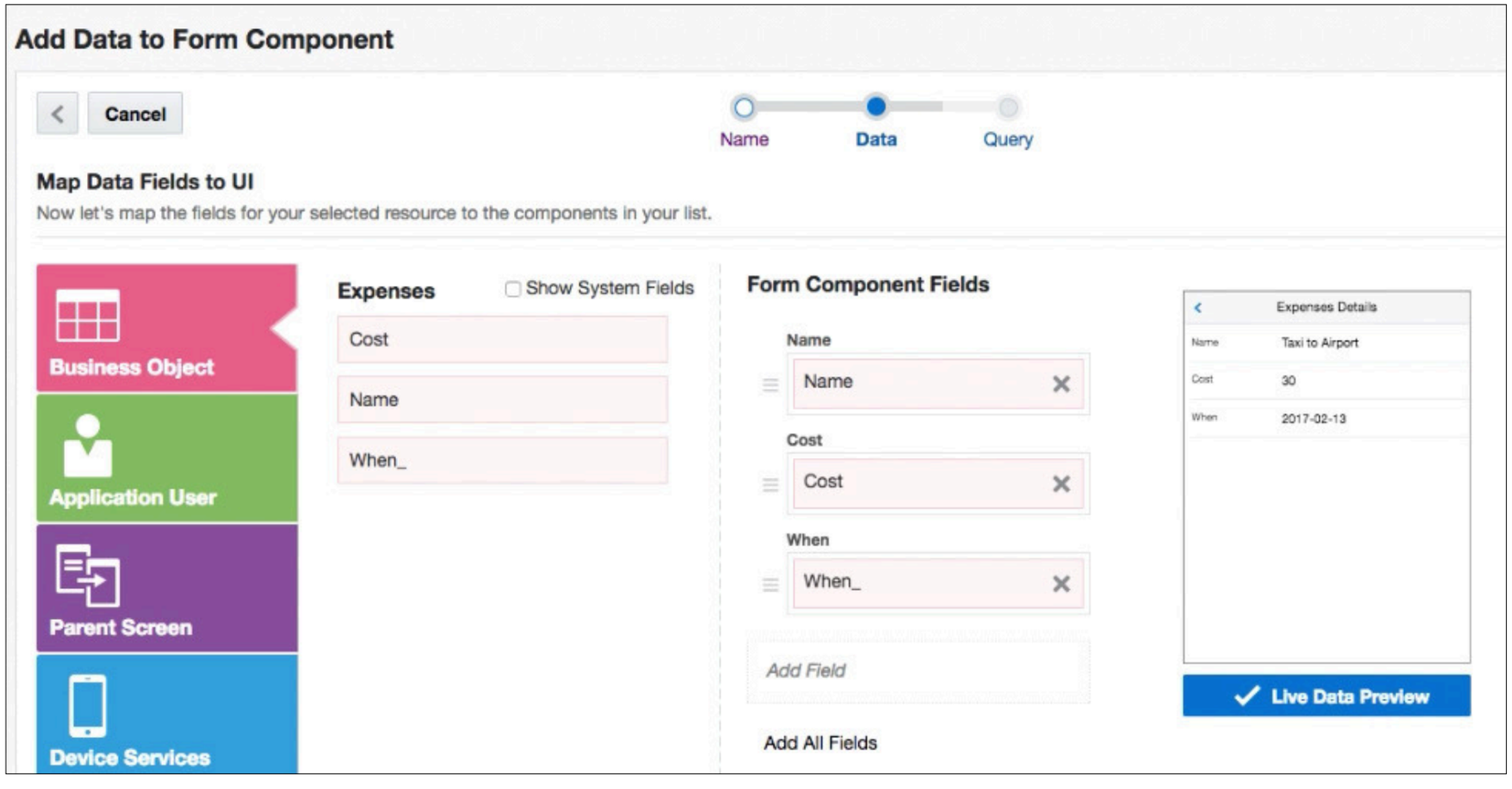


Figure 8: Running the mobile app in an emulator in your browser

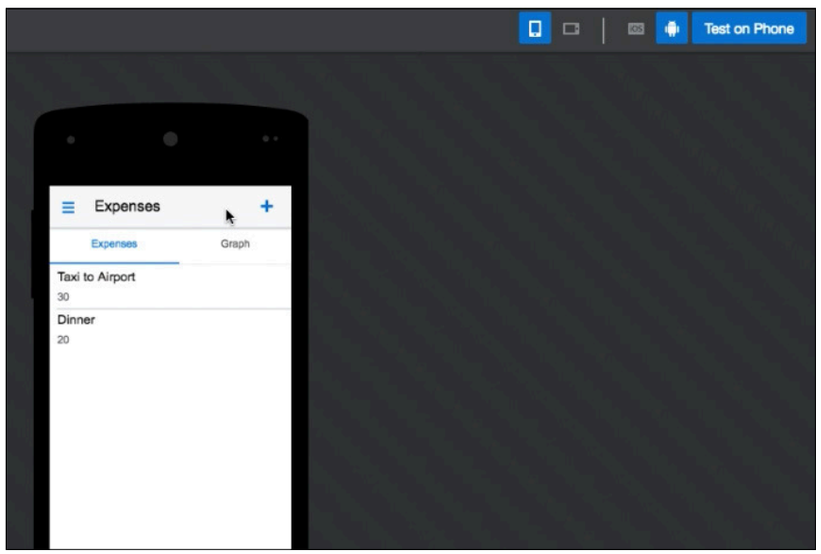
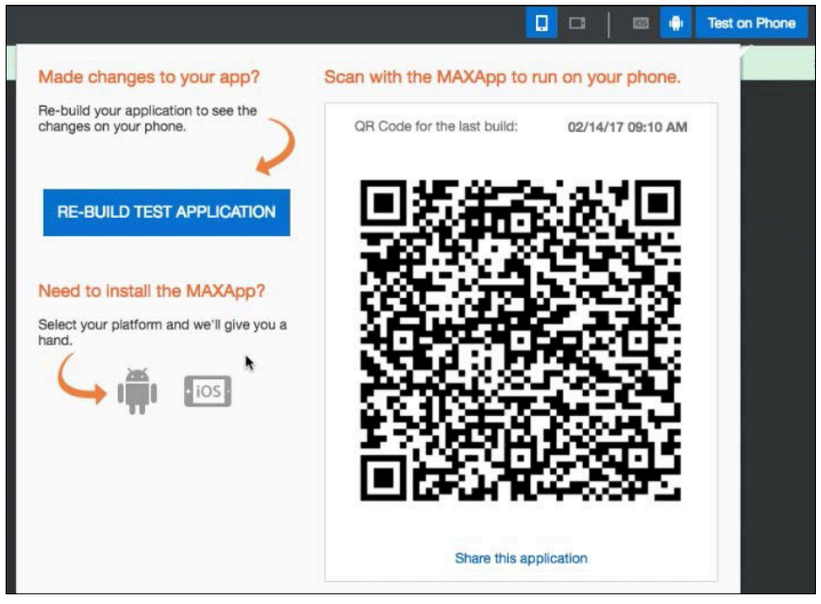


Figure 9: The packaged app, ready for download to your phone



watching the article's video. Additional customization instructions are not included in this article.

You are now ready to run the application.

TESTING AND RUNNING YOUR APPLICATION

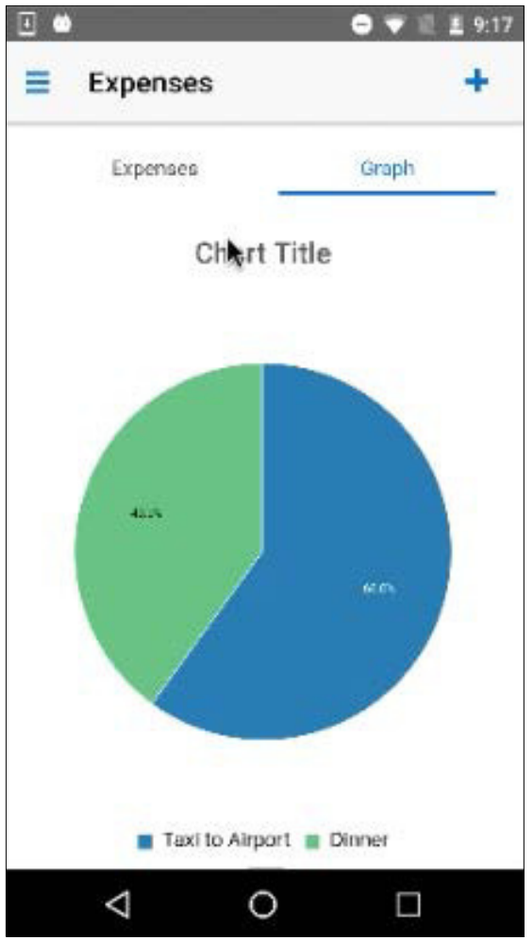
To run the application, click the run arrow at the top right. Your application shows up in the browser inside a mobile phone emulator (see **Figure 8**). You can navigate through the app and see how it is going to look on iOS as well as Android phones. Note that the application adopts a native look for each platform. This functionality is achieved thanks to the Oracle JavaScript Extension Toolkit (Oracle JET)–based user interface and its ability to render native-looking components, based on the operating system of your mobile phone.

The next step is to run the application on your device. To do that, first download Oracle Mobile Application Accelerator from the app store for your operating system. Next, in the Oracle Application Builder Cloud Service web UI, click **Test on Phone** (at the top right) and then click **Build Test Application**. This will package the application for download, and when the packaging is done, the screen displays a QR code, as shown in **Figure 9**.

Invoke Oracle Mobile Application Accelerator on your device, tap the **+** sign, and use your device's camera to scan the QR code. The application will download to your device, and you'll be able to test it on your phone, as shown in **Figure 10**.

From now on, when you invoke your Oracle Mobile Application Accelerator application, the expense application will be there.

Figure 10: Your new on-device mobile app running on your phone



SUMMARY

As you saw, you can use Oracle Application Builder Cloud Service to create and deploy an on-device mobile application without coding. This article only scratched the surface of what is possible with the new mobile development capabilities in Oracle Application Builder Cloud Service. Now it is your turn to dig deeper: sign up for a trial of Oracle Application Builder Cloud Service, and start exploring.

Shay Shmeltzer is director of product management for Oracle's cloud development tools. He is focused on helping developers simplify and streamline their work by leveraging the right tools and technologies.

NEXT STEPS

TRY Oracle Application Builder Cloud Service.

LEARN more about Oracle Application Builder Cloud Service.

EXPLORE the Oracle Application Builder Cloud Service Channel.

Change Happens

Use open source tools to handle database change management.

By Blaine Carter

Oracle Database offers a powerful foundation on which to build and manage applications. And because applications are constantly changing, the underlying databases supporting those applications must also change.

How do you manage these changing databases? The obvious answer is or should be, "Just check the SQL scripts into my version control system." That works pretty well, right up until you need to go back and alter one of the existing objects. Changing your database structure is *not* as simple as changing application code.

This article explores how you can use a popular open source tool to help you manage database changes.

ONE WAY I'VE HANDLED CHANGES MANUALLY

In the past, I've handled database changes by running a set of scripts at release time. I prefer to have each script handle a single task, such as creating a table, when possible.

For each release, I've maintained two directories of scripts:

1. A set of master/create scripts for a fresh install of the product at that version
2. A set of "alter…" scripts to modify the database from the previous release to this release

Each script set had its own master script, and each script contained validation PL/SQL to make sure the script ran only when it was supposed to and logged errors when something went wrong.

I have made this process work, but it has been a bit of a struggle. Every change needed to be performed twice, once in each directory. And this approach did not address several critical issues, such as

- What if I need to roll something back?
- Did everything run?
- Can I run certain scripts only under specific conditions, such as testing?

There are several open source tools to help manage these and other critical database update issues. And there's one open source application I've used off and on that handles these database change issues and quite a bit more: Liquibase.

LIQUIBASE: SOURCE CONTROL FOR DATABASES

Liquibase is a change management tool. It is open source, was written in Java, and works with several different databases.

Because different databases implement variations on American National Standards Institute (ANSI) SQL, Liquibase uses its own object definitions, in multiple formats such as XML, JSON, and YAML. And instead of SQL scripts, Liquibase uses change set files. Most of the change set examples on the Liquibase site use XML, but I think XML is a bit too pointy. These days the most commonly used notation is

JSON, and that's what I use most often. You can also create change sets with SQL, but Liquibase will not auto-generate rollbacks. I'll explain more on this later.

Much as with my master run script approach, Liquibase uses a master change log file that executes the individual change set files in the order they are intended to be run.

For example, the following master log file will run the changelog-1.0 and then the changelog-2.0 JSON-formatted change sets:

```
{
  "databasechangelog": [
    {
      "include": {
        "file": "changelog/db.changelog-1.0.json"
      }
    },
    {
      "include": {
        "file": "changelog/db.changelog-2.0.json"
      }
    }
  ]
}
```

The first change set—db.changelog-1.0.json, in **Listing 1**—creates a table called lb_groups with three columns: id, name, and description.

Code Listing 1: db.changelog-1.0.json change set

```
{"databaseChangeLog": [{
  "preConditions": [{
    "runningAs": {"username": "lb_demo"}
                }]
        },{
            "changeSet": {
                "id": "Two-Table-1",
                "author": "BlaineCarter",
                "comment": "Add table lb_groups",
                "tagDatabase": {"tag":"myCoolTag1"},
                "context": "context1",
"labels": "label1",
"changes": [
    {
        "createTable": {
            "tableName": "lb_groups",
            "columns": [
                {
                    "column": {
                        "name": "id",
                        "type": "int",
                        "autoIncrement": true,
                        "constraints": {
                            "primaryKey": true,
```

```
                "nullable": false
            }
        }
    },
    {
        "column": {
            "name": "name",
            "type": "varchar(50)",
            "constraints": {
                "unique": true,
                "uniqueConstraintName": "uk_lb_groups_name"
            }
        }
    },
    {
        "column": {
            "name": "description",
            "type": "varchar(200)"
        }
    }
                ]
            }
        }
    ]
}
```

```
    ]
}
```

The second and slightly more complex change set—db.changelog-2.0.json, in **Listing 2**—creates the lb_people table and includes a foreign key on lb_groups.id.

Code Listing 2: db.changelog-2.0.json change set

```
{"databasechangelog": [{
  "preConditions": [
    {"runningAs": {"username": "lb_demo"}}
  ]
        },{
    "changeSet": {
        "id": "Two-Table-2",
        "author": "BlaineCarter",
        "comment": "Add table lb_people",
        "tagDatabase": {"tag":"myCoolTag2"},
        "context": "context2",
        "labels": "label2",
        "changes": [
            {
                "createTable": {
                    "tableName": "lb_people",
                    "columns": [
                        {
                            "column": {
```

```
            "name": "id",
            "type": "int",
            "autoIncrement": true,
            "constraints": {
                "primaryKey": true,
                "nullable": false
            },
        }
    },
    {
        "column": {
            "name": "firstname",
            "type": "varchar(50)"
        }
    },
    {
        "column": {
            "name": "lastname",
            "type": "varchar(50)",
            "constraints": {
                "nullable": false
            },
        }
    },
    {
        "column": {
```

```
                                "name": "group_id",
                                "type": "int",
                                "constraints": {
                                    "foreignKeyName": "groupFK",
                                    "references": "lb_groups(id)"
                                },
                            }
                        }
                    ]
                }
            }
        ]
    }
```

The Liquibase documentation recommends that you perform one change per change set. You could do everything in a single file *if you wanted to*, but the more you pack into a single change set, the more difficult it will be to troubleshoot problems later.

As you can see in **Listings 1** and **2**, most of the change set properties are self-explanatory, but there are a couple of extra properties.

The change sets require an "id" and an "author." These values are combined with the name of the change set file and package to uniquely identify a change set.

The id is a string type and can be any value. The author value should refer to the person who authored the change set. Although an author value is required, it is not validated against anything, so you could use "Unknown" as the author value if you would rather not track authors.

The preConditions array enables you to identify checks that must pass for the changes to run. In these change sets, I'm making sure I'm logged in as lb_demo in the preConditions array. (There are many other preconditions you can use.) If any of the preConditions at the change log level fail, the current execution of the change log will fail and none of the change sets will run. You can also include preConditions inside the changeSet code, and these will apply to that specific change only.

The "comment," "tagDatabase," "context," and "labels" properties are used by Liquibase to track changes and for some more-advanced control options I won't cover in this article. There are links to the documentation for these properties and more at the end of the article.

SET UP YOUR ENVIRONMENT

To create your own master log and change set files in Liquibase and exercise Liquibase database change management features, first set up your environment:

1. In your database, create a schema named lb_demo. If you would rather use another schema, replace lb_demo with your schema in the examples. *Make sure the schema you use is safe to experiment in.*
2. Download and extract Liquibase. I extracted it to /opt/liquibase/.
3. Check your Java version with this command:

```
java -version
```

Install Java or upgrade Java as needed. (Liquibase 2.x requires Java 1.5+; Liquibase 3.x requires Java 1.6+.)

1. Locate—or download and locate—the Oracle JDBC driver. Mine is located at /usr/lib/oracle/12.1/client64/lib/ojdbc7.jar.
2. Create a directory to work in.
3. Inside that directory, create a child directory called changelog.
4. Take the ChangeLog master example above, and save it in the changelog directory to a file named db.changelog-master.json.
5. Take the first change set example—**Listing 1**—and save it in the changelog directory to a file named db.changelog-1.0.json.
6. Take the second change set example—**Listing 2**—and save it in the changelog directory to a file named db.changelog-2.0.json.
7. (Optional) Define an environment variable with any additional configurations your database connection may need. Skip this step if your database connection doesn't require anything extra.

I use Oracle Database Exadata Express Cloud Service, which includes additional security enhancements, so I use the following configuration information for my connection:

```
export JAVA_OPTS="-Doracle.net.tns_admin=/pathToMyCreds/EE_Credentials
-Doracle.net.ssl_server_dn_match=true
-Doracle.net.ssl_version=1.2
-Djavax.net.ssl.trustStore=/pathToMyCreds/EE_Credentials/truststore.jks
-Djavax.net.ssl.trustStorePassword=SuperSecurePassword
-Djavax.net.ssl.keyStore=/pathToMyCreds/EE_Credentials/keystore.jks
-Djavax.net.ssl.keyStorePassword=SuperSecurePassword"
```

RUNNING IT

Warning: There are no "Are you sure?" prompts or separate commits when you run Liquibase. Commands that alter the database *just run*. *Use a schema that is safe to experiment with.*

Liquibase is a Java application, so you can run it from the command line as follows:

```
java $JAVA_OPTS -jar /opt/liquibase/liquibase.jar
--driver=oracle.jdbc.OracleDriver
--classpath="/usr/lib/oracle/12.1/client64/lib/ojdbc7.jar"
--url=jdbc:oracle:thin:lb_demo/dd@dbaccess
--changeLogFile=changelog/db.changelog-master.json
updateSQL >> output.sql
```

Replace the values for -jar, --classpath, and --url with the values your system requires.

Here are explanations of the parameters:

- $JAVA_OPTS is optional. You can leave it out if you don't need it.
- -jar points to the liquibase.jar file.
- --driver identifies the type of database driver you're using.
- --classpath points to your JDBC driver.
- --url is a standard JDBC connection string. (Replace the url value with your connection string.)
- --changeLogFile points to the master change log file you created. Note that I'm running the command from the working directory above changelog. Refer to the Liquibase Best Practices page for the recommended directory structure.

- updateSQL is the command you're passing to Liquibase. updateSQL instructs Liquibase to generate and display the SQL that *would run* to make the changes, but it *does not actually run the SQL in the database*. The generated SQL will be returned to the command line after you run the command.
- Because just dumping the SQL to the screen can be a little messy, I prefer to pipe the output to a file, using >> output.sql.

Assuming that everything is set up correctly when you run this command, it will create a file called output.sql that you can inspect and/or run in your database manually.

Most of the Liquibase commands include an option that ends with SQL. For example, the update command will update your database and the updateSQL command will generate the same SQL and return it but not run it. If your DBA prefers to inspect any SQL before it runs in the database, use the …SQL commands and pipe the output to a file.

Listing 3 contains the output.sql file created by your Liquibase run.

Code Listing 3: output.sql

```
-- ******************************************************************
-- Update Database Script
-- ******************************************************************
-- Change Log: changelog/db.changelog-master.json
-- Ran at: 1/4/17 4:04 PM
-- Against: LB_DEMO@jdbc:oracle:thin:lb_demo/dd@dbaccess
-- Liquibase version: 3.5.1
-- ******************************************************************
```

```
SET DEFINE OFF;

-- Create Database Lock Table
CREATE TABLE LB_DEMO.DATABASECHANGELOGLOCK (ID INTEGER NOT NULL, LOCKED NUMBER(1) NOT
NULL, LOCKGRANTED TIMESTAMP, LOCKEDBY VARCHAR2(255), CONSTRAINT PK_DATABASECHANGELOGLOCK
PRIMARY KEY (ID));

-- Initialize Database Lock Table
DELETE FROM LB_DEMO.DATABASECHANGELOGLOCK;

INSERT INTO LB_DEMO.DATABASECHANGELOGLOCK (ID, LOCKED) VALUES (1, 0);

-- Lock Database
UPDATE LB_DEMO.DATABASECHANGELOGLOCK SET LOCKED = 1, LOCKEDBY = 'myPc (192.168.0.111)',
LOCKGRANTED = to_timestamp('2017-01-04 16:04:08.757', 'YYYY-MM-DD HH24:MI:SS.FF') WHERE
ID = 1 AND LOCKED = 0;

-- Create Database Change Log Table
CREATE TABLE LB_DEMO.DATABASECHANGELOG (ID VARCHAR2(255) NOT NULL, AUTHOR VARCHAR2(255)
NOT NULL, FILENAME VARCHAR2(255) NOT NULL, DATEEXECUTED TIMESTAMP NOT NULL,
ORDEREXECUTED INTEGER NOT NULL, EXECTYPE VARCHAR2(10) NOT NULL, MD5SUM VARCHAR2(35),
DESCRIPTION VARCHAR2(255), COMMENTS VARCHAR2(255), TAG VARCHAR2(255), LIQUIBASE
VARCHAR2(20), CONTEXTS VARCHAR2(255), LABELS VARCHAR2(255), DEPLOYMENT_ID VARCHAR2(10));

-- Changeset changelog/db.changelog-1.0.json::Two-Table-1::BlaineCarter
-- Add table lb_groups
```

```
CREATE TABLE LB_DEMO.lb_groups (id INTEGER GENERATED BY DEFAULT AS IDENTITY NOT NULL,
name VARCHAR2(50), description VARCHAR2(200), CONSTRAINT PK_LB_GROUPS PRIMARY KEY (id),
CONSTRAINT uk_lb_groups_name UNIQUE (name));

INSERT INTO LB_DEMO.DATABASECHANGELOG (ID, AUTHOR, FILENAME, DATEEXECUTED,
ORDEREXECUTED, MD5SUM, DESCRIPTION, COMMENTS, EXECTYPE, CONTEXTS, LABELS, LIQUIBASE,
DEPLOYMENT_ID) VALUES ('Two-Table-1', 'BlaineCarter', 'changelog/db.changelog-1.0.json',
SYSTIMESTAMP, 1, '7:a3aa285c230661c094a6e34ae7639b74', 'createTable tableName=lb_
groups', 'Add table lb_groups', 'EXECUTED', NULL, NULL, '3.5.1', '3571049605');

-- Changeset changelog/db.changelog-2.0.json::Two-Table-2::BlaineCarter
-- Add table lb_people
CREATE TABLE LB_DEMO.lb_people (id INTEGER GENERATED BY DEFAULT AS IDENTITY NOT NULL,
firstname VARCHAR2(50), lastname VARCHAR2(50) NOT NULL, group_id INTEGER, CONSTRAINT
PK_LB_PEOPLE PRIMARY KEY (id), CONSTRAINT groupFK FOREIGN KEY (group_id) REFERENCES LB_
DEMO.lb_groups(id));

INSERT INTO LB_DEMO.DATABASECHANGELOG (ID, AUTHOR, FILENAME, DATEEXECUTED,
ORDEREXECUTED, MD5SUM, DESCRIPTION, COMMENTS, EXECTYPE, CONTEXTS, LABELS, LIQUIBASE,
DEPLOYMENT_ID) VALUES ('Two-Table-2', 'BlaineCarter', 'changelog/db.changelog-2.0.json',
SYSTIMESTAMP, 2, '7:86ab24036ce9b179375b13899471ddbd', 'createTable tableName=lb_
people', 'Add table lb_people', 'EXECUTED', NULL, NULL, '3.5.1', '3571049605');

-- Release Database Lock
UPDATE LB_DEMO.DATABASECHANGELOGLOCK SET LOCKED = 0, LOCKEDBY = NULL, LOCKGRANTED = NULL
WHERE ID = 1;
```

Liquibase uses a couple of internal tables to track when changes are actively being made and what changes have already been made. The first few lines of the output.sql file set up and populate these tables. Liquibase creates these tables only the first time it runs; subsequent runs will use the existing tables to track changes.

Here are explanations of the output.sql elements:

1. Because this is the first time you're running Liquibase, your script will create and initialize the Liquibase table databasechangeloglock. This is used to make sure only one person is running Liquibase at a time.
2. The first Liquibase run also creates the databasechangelog table. This is the table Liquibase uses to track information about the change sets that have been run.
3. Based on information in the db.changelog-1.0.json file, output.sql creates the lb_groups table.
4. The output.sql file adds a row to databasechangelog with data about the execution of the change set.
5. Based on information in the db.changelog-2.0.json file, output.sql creates the lb_people table.
6. The output.sql file adds a row to databasechangelog with data about the execution of the change set.
7. The output.sql file updates databasechangeloglock to release the lock.

At this point, you or your DBA *could run* the output.sql file against your database to make the changes. Instead, let's let Liquibase make the changes.

Before Liquibase makes the changes, let's make it a little easier to run the tool. If you add the Liquibase directory to your path, you will be able to run the included shell script or .bat file that reads a properties file that stores most of the parameters described earlier (and others if needed).

Save the following property information in the working directory above changelog as liquibase.properties:

```
#Liquibase.properties
driver: oracle.jdbc.OracleDriver
classpath: /usr/lib/oracle/12.1/client64/lib/ojdbc7.jar
url: jdbc:oracle:thin:lb_demo/dd@dbaccess
changeLogFile: changelog/db.changelog-master.json
```

Replace the values for classpath: and url: with the values your system requires.

If you've already set $JAVA_OPTS, you don't need to include it here; Liquibase will include it automatically.

Now run Liquibase and have it make the changes to the database as follows, remembering that there are no "Are you sure?" prompts:

```
liquibase update
```

Connect to your database with your favorite SQL tool, and inspect the changes. You should have four new tables: databasechangeloglock, databasechangelog, lb_groups, and lb_people.

Now run the following SQL query:

```
select * from  databasechangelog;
```

and let's take a look at the data tracked by Liquibase. In this example, ID, AUTHOR, COMMENTS, TAG, CONTEXTS, and LABELS all are set in the change sets.

The rest of the columns are populated by Liquibase. Most of them are self-explanatory, except for the following:

- MD5SUM is the checksum Liquibase uses to determine if the change set file has been altered.
- LIQUIBASE is the version of Liquibase used when the change set was run.
- DEPLOYMENT_ID is an internally generated id you can ignore.

Tags can be set in the change set. Note that the last change set to run has a myCoolTag2 tag. Typically you wouldn't set the tag in the change set but would instead set it from the command line, like this:

```
liquibase tag newCoolTagCL
```

And when you rerun the SQL query

```
select * from  databasechangelog;
```

you can see that the tag on the last-run change set has changed to newCoolTagCL. When you set the tag with a command, it will *overwrite* the existing tag on the last-run change set. This is useful if you don't know what the application release number will be when you are creating change sets. I like to set up my automated build process to autogenerate the current release number and then use the tag command to tag the last-run change set.

ROLLBACK

One of the potential questions I mentioned with my pre-Liquibase script library strategy was what if I want to roll back a change. For most changes, Liquibase can

automatically generate the rollback.

There are three different command modes you can use to roll back changes:

Number of change sets. In the above example, you executed two change sets. This rollback command will roll back changes in the reverse order of execution. Running it for one count would drop the lb_people table. If you ran it a second time, it would drop the lb_groups table.

Run the command

```
liquibase rollbackCount 1
```

and look at your database to see that the lb_groups table has been dropped. In SQL, run the query

```
select * from  databasechangelog;
```

Note that the row for changelog-2.0 has been deleted. Liquibase tracks only change sets that have been applied to the database and will delete those that are rolled back.

To a date/time. This command will roll back all change sets that were applied after the given date and time. The date/time format used is “yyyy-MM-dd HH:mm:ss.”

Let’s reapply changelog-2.0 with the command

```
liquibase update
```

In SQL, run the query

```
select dateexecuted from databasechangelog order by orderexecuted;
```

Look at the output, choose a date/time between the two “dateexecuted” values, and run the following using that value:

```
liquibase rollbackToDate "2017-01-05 18:37:00"
```

Look at your database to see that the lb_groups table has been dropped. In SQL, run the query

```
select * from  databasechangelog;
```

Note that the row for changelog-2.0 has been deleted.

To a tag. This command will find the newest change set with a matching tag (tags are not unique) and roll back all the change sets that were run after that. If the tag doesn’t exist, Liquibase will throw an error.

Let’s reapply changelog-2.0 with the command

```
liquibase update
```

And let’s roll back to a specific tag:

```
liquibase rollback myCoolTag1
```

Look at your database to see that the lb_groups table has been dropped. In SQL, run the query

```
select * from  databasechangelog;
```

Note that the row for changelog-2.0 has been deleted.

Changes that cannot be automatically rolled back. Changes such as DROP TABLE and “data changes” (INSERT, UPDATE, DELETE) cannot be rolled back automatically.

I mentioned earlier that you can create change sets by using SQL instead of the other formats. The main drawback to this method is that these changes cannot be automatically rolled back. If you attempt to roll back a SQL change set and run into something that cannot be automatically rolled back, Liquibase will throw an error.

Defining a rollback. Liquibase includes a method for defining your own rollback commands for times when it cannot autogenerate them. You can also use these if you have a special case and you want more control over the rollback.

For the times when there is no automatic rollback, you need to add a rollback tag. Otherwise, Liquibase will throw an error when it hits a change that can’t be rolled back.

Let’s add a change set that inserts some data.

Create a file named db.changelog-3.0.json in the changelog directory, and add the code in **Listing 4** to the file.

Code Listing 4: db.changelog-3.0.json change set

```
{"databaseChangeLog": [{
            "preConditions": [{"runningAs": {"username": "lb_demo"}} ]
        },{
            "changeSet": {
                "id": "Two-Table-2",
                "author": "BlaineCarter",
```

```
"changes": [
  {
    "insert": {
      "columns": [
        {
          "column": {
            "name": "firstname",
            "value": "Blaine"
          }
        },
        {
          "column": {
            "name": "Lastname",
            "value": "Carter"
          }
        }],
      "dbms": "oracle",
      "schemaName": "lb_demo",
      "tableName": "lb_people"
    }
  }]
}
]
}
```

Now add the change set to the db.changelog-master.json file:

```
,
    {
      "include": {
        "file": "changelog/db.changelog-3.0.json"
      }
    }
```

Run the changes with the command

```
liquibase update
```

Use a SQL query to check that the data was inserted into the lb_people table. Now attempt a rollback:

```
liquibase rollbackCount 2
```

You should get an error message:

```
Unexpected error running Liquibase: No inverse to
liquibase.change.core.InsertDataChange created
```

To fix this problem, modify changelog/db.changelog-3.0.json to include a rollback tag:

```
...
                    "schemaName": "lb_demo",
                    "tableName": "lb_people"
                  }
                },
                {"rollback": "delete from lb_people"}
              ]
            }
...
```

Rerun the rollback:

```
liquibase rollbackCount 2
```

Look at your database to see that the lb_groups table has been dropped. Then, in SQL, run the query

```
select * from databasechangelog;
```

Note that the rows for changelog-2.0 and changelog-3.0 have been deleted.

If you're sure that nothing needs to be done for a rollback but you still want the ability to roll back multiple change sets, you can include an empty rollback:

```
{"rollback": ""}
```

HOW DOES THIS COMPARE TO MY OLD SYSTEM?

At this point, you should be able to create and roll back changes in your database by using Liquibase.

Now let's compare the database update approaches:

	Old method	Liquibase
Two sets of scripts to maintain	Two sets of scripts are not strictly required, and it is possible to maintain just the update scripts. However, the extra overhead of tracking which changes have been run with plain SQL scripts always seems to be a lot more work than maintaining two sets.	Liquibase automatically tracks what has been run, when it was run, and who ran it. This makes it easy to maintain only the update change sets, starting from an empty database. There is no need to keep a "create from scratch" set of scripts.
Master run script	Same	Same
Rollback to X	Significant effort, typically involving brand-new scripts	Autogenerated (mostly); multiple options for controlling the point to roll back to
What ran	Manual effort to add logging functionality to each script or logging script output at runtime	Run data automatically tracked in a table
Running certain scripts under specific conditions	Write specific master run scripts for each condition.	Use contexts and labels to control which scripts get run. (This is not covered in this article, but links are provided below.)

This article just scratches the surface of what's possible with Liquibase, but with this information, you should be able to get started. Even if you never go beyond what's covered here, Liquibase will save you a significant amount of effort and automatically generate some very useful data.

WANT TO KNOW MORE?

Liquibase helps you get your database changes under control and integrated into your build process, leading to smoother upgrades and fewer fires to put out. It also includes many features not covered in this article, and the Liquibase documentation is comprehensive and easy to navigate. The documentation includes advanced features such as

- Using contexts and labels for fine-grained control over which change sets are run under specific conditions
- Autotagging your change sets from your continuous integration process
- Branching and merging change sets
- Working with advanced preconditions
- Generating change logs from your existing database (to reverse-engineer your database)
- Finding the difference between two schemas
- Using and/or creating extensions to Liquibase
- Loading data from a .csv file

And if, by any chance, you come across a requirement that Liquibase cannot satisfy, well, it is an open source project. So you can submit a request for an enhancement or, even better, come up with an enhancement and submit a pull request on GitHub.

Blaine Carter is the Oracle developer advocate for open source. He applies his exploratory eye and tinkering inclinations to the intersection of open source software and Oracle Database.

NEXT STEPS

DOWNLOAD
Oracle Database 12*c* Release 2.

DOWNLOAD
Liquibase.

LEARN more about Liquibase.

ORACLE APPLICATION EXPRESS

Easy Master/Detail

Build master/detail forms for business applications with Oracle Application Express 5.1 interactive grids.

By Joel Kallman

Master/detail relationships are omnipresent in business applications. From department/employees to country/cities, data is often categorized and presented with this one-to-many type of relationship. With the interactive grid, a new component in Oracle Application Express 5.1, you can now easily create editable grids that reflect master/detail relationships. And you can easily extend your application to represent any number of relationships—parent/child/grandchild—or even a master table and multiple detail tables.

In this *Oracle Magazine* article, you're going to build a web application on top of a set of three tables: REGIONS, COUNTRIES, and LOCATIONS. COUNTRIES is a child table of REGIONS, and LOCATIONS is a child table of COUNTRIES. These tables are excerpted from the Oracle Database Sample Schemas, available on GitHub. Using Oracle Application Express 5.1, you will quickly create a single page to manage the data of all these tables, using interactive grids.

ABOUT ORACLE APPLICATION EXPRESS

Oracle Application Express is a high-productivity, low-code platform for creating modern, responsive, and accessible web applications. A no-cost feature of Oracle Database, it is a compelling application development platform available in all Oracle Database Cloud services.

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 from the Oracle Technology Network the Database App Development Virtual Machine, which includes a preconfigured Oracle Database 12*c* Release 2 Enterprise Edition database, 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 Oracle Application Express 5.1 interactive grids by first creating the sample database objects of 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_tables_master_detail.sql** file from your local computer, and click the **Upload** button.
3. Click the **Run** icon in the same row as the **sample_tables_master_detail.sql** script, and then click the **Run Now** button.

You've now created the REGIONS, COUNTRIES, and LOCATIONS tables in your schema and populated them with data.

CREATING THE APPLICATION

Begin your exploration of master/detail support in Oracle Application Express 5.1 by creating the initial application.

1. In Oracle Application Express, click the **App Builder** tab.

2. Click the **Create** icon, and then click the **Desktop** icon.
3. Enter `Data Management` for **Name**, and click **Next**.
4. Click the **x** in the Home Page row to remove the default home page.
5. Click the **Add Page** button.
6. Click the **Master Detail** icon, select **REGIONS** for **Master Table Name**, select **COUNTRIES** for **Detail Table Name**, and click **Add Page**.
7. Click the **Create Application** button, and then click the **Create Application** button again.

You've now created a responsive Oracle Application Express application with a fully functional, multiuser editable grid on the REGIONS table, along with a child editable grid on the COUNTRIES table, all accessible from a web browser.

Click the **Run Application** icon to run your application. Log in with the same credentials you used to log in to Oracle Application Express. Click each of the rows in the Region Name column of the Regions interactive grid to refresh the child interactive grid with the related rows. Enter text in the search field of the Regions and Countries interactive grids, and press the Enter key to filter the results.

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

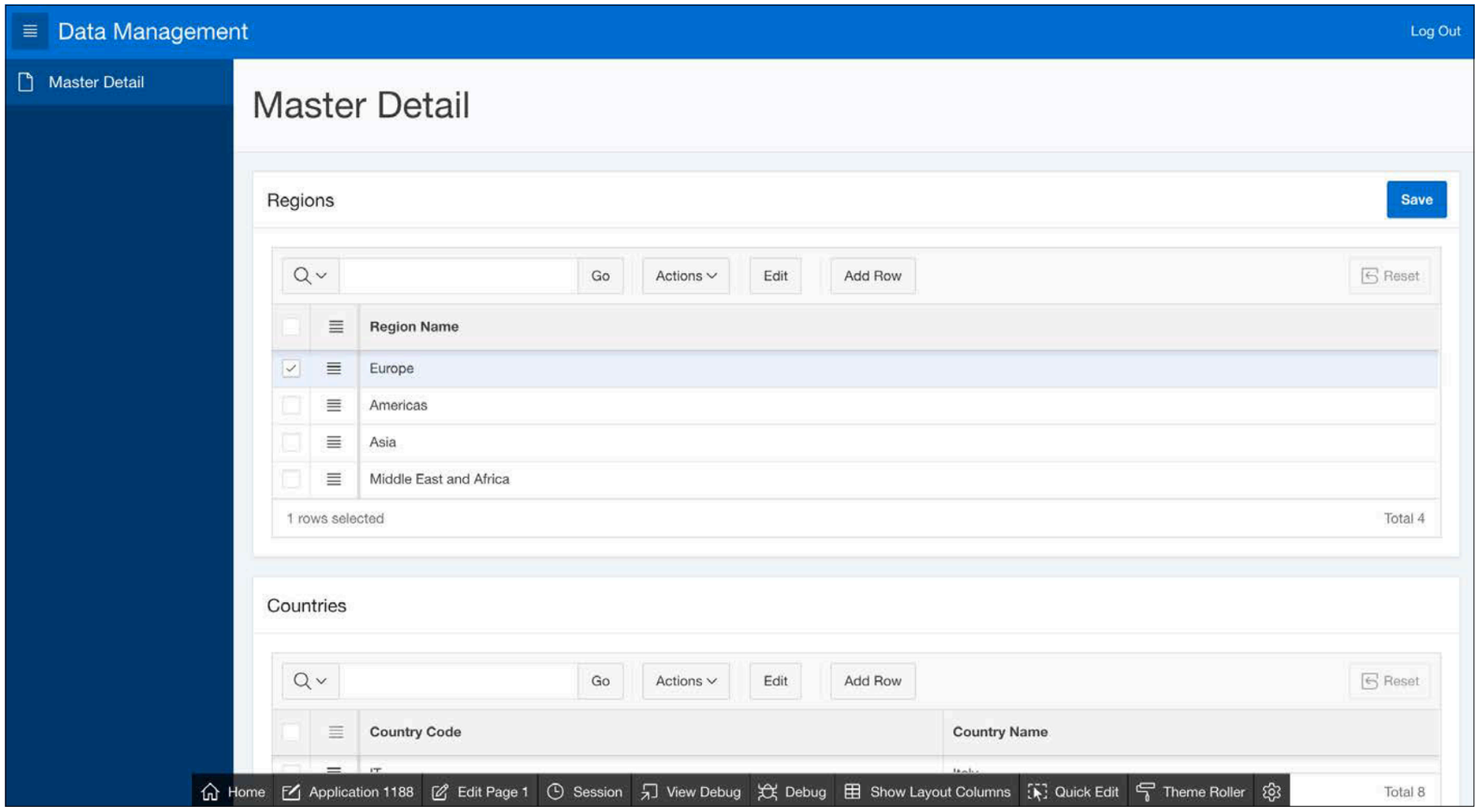
ADJUSTING THE APPEARANCE

Improve the display of the page by refining the layout and visual appearance of the interactive grid regions.

With very wide columns and only one or two columns displayed per interactive grid, a significant amount of space on the application page is wasted. You can easily adjust the properties of the grid regions to dramatically improve the display.

1. In the Developer toolbar at the bottom of the page, click the **Quick Edit** link.

Figure 1: The initial master/detail application page



2. Move your cursor over the **Save** button, and when an outline appears over the button, click the button. Do not click the wrench icon in the upper right of the outline. (The wrench icon will be used in a later step.)
3. The tab running Application Builder will be displayed, and the **Save** button in the Regions interactive grid will be selected in Page Designer. In the visual page layout section of Page Designer, drag and drop the **SAVE** button from the Regions region to the EDIT position of the Breadcrumbs region.

4. In the visual layout area of Page Designer in the middle of the page, select the Countries region. In the properties of the Countries region on the far right, change **Start New Row** to **No**.
5. Click the **Run** icon in the upper right of Page Designer to run your page.

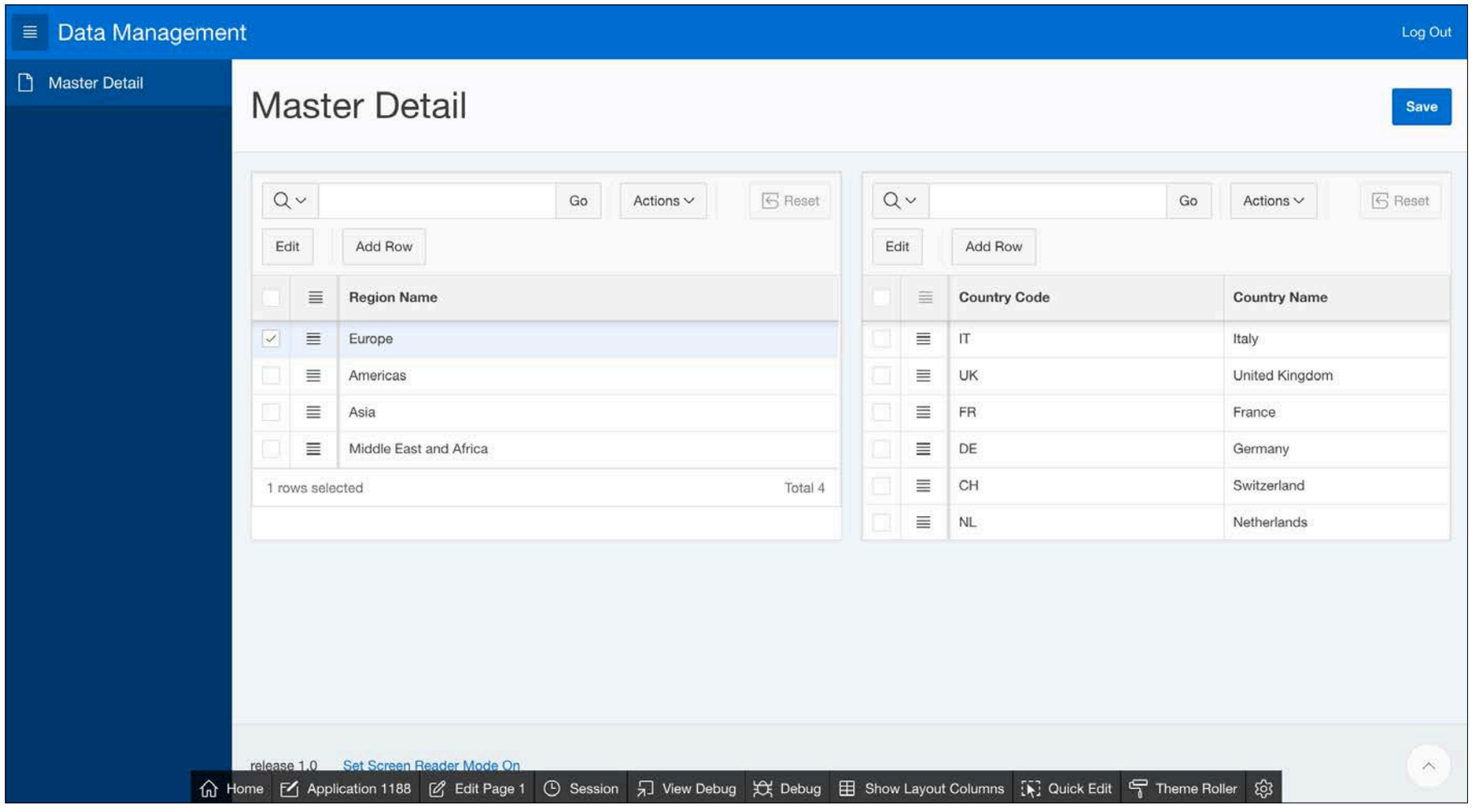
Your application page now presents the interactive grid regions side by side, with the **Save** button in the breadcrumbs bar. You're getting there, but this application can be improved further.

Template options are declarative modifiers for the display of content on a page. The new live template options feature in Oracle Application Express 5.1 enables you to modify the look and feel of your content and see the results in real time.

6. In the Developer toolbar at the bottom of the page, click the **Quick Edit** link again. Hover your cursor over the **Regions** region, but this time, click the wrench icon in the upper right of the highlighted region. This will display the live template options for your region.
7. Select the checkbox for **Remove Body Padding**. Change **Header** to **Hidden but accessible**. Change **Body Height** to **320px**, and then click the **Save** button.
8. Make the same changes to the Countries region. Click the **Quick Edit** link, hover the cursor over the **Countries** region, and click the wrench icon in the upper right. Select the checkbox for **Remove Body Padding**. Change **Header** to **Hidden but accessible**. Change **Body Height** to **320px**, and then click the **Save** button.
9. Click the **Run** icon in the upper right of Page Designer to run your page.

The functionality of the interactive grids in your application page remains as before, but the presentation of information is much more condensed on the page. Your application should look similar to **Figure 2**.

Figure 2: The master/detail application page, condensed



ADDING A THIRD INTERACTIVE GRID

Oracle Application Express supports an unlimited number of interactive grids on a single page. You will now add a third interactive grid, this time for the LOCATIONS table, and then manually adjust the properties so it is associated as a child of the COUNTRIES table.

1. In the Developer toolbar at the bottom of the page, click the **Edit Page 1** link. This will present the tab in your browser running Application Builder.

2. In the Component Gallery at the bottom of the Page Designer page, select the **Interactive Grid** component and drag it below the Regions region in the Page Layout area, and drop it. The new region will be labeled New, and it will be highlighted in red, because you must supply one or more values to make it valid.
3. In the properties on the far right of Page Designer, for **Title** enter `Locations`.
4. In the **SQL Query** property, enter

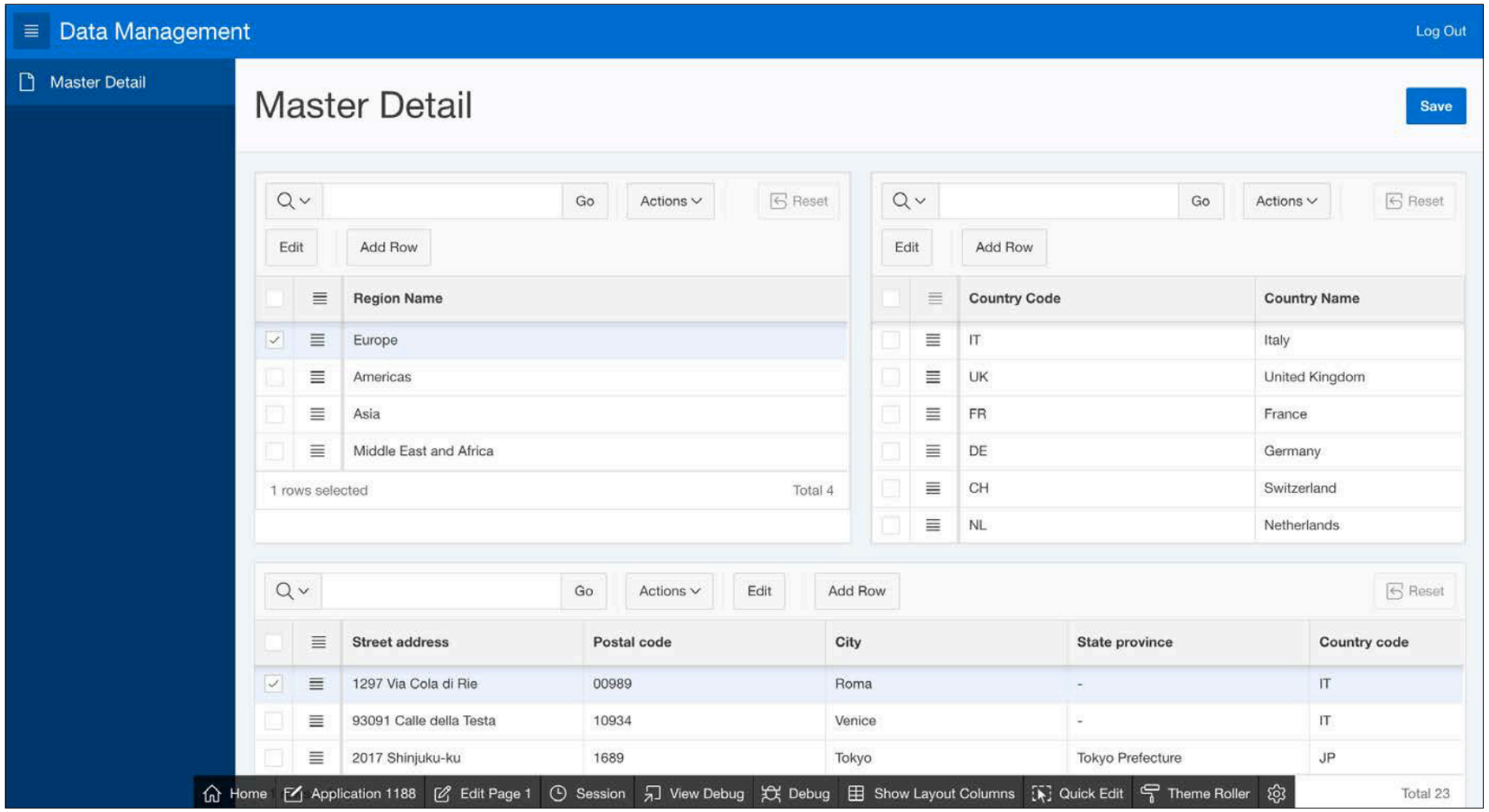
```
select location_id, street_address, postal_code,
       city, state_province, country_code
  from locations
```

5. Click **Attributes** in the Locations region in the Component Tree on the left side of Page Designer.
6. In the Property Editor on the right, in the Edit region, change **Enabled** to **Yes**. This will change the interactive grid from a read-only reporting component to a fully editable grid.
7. Scroll down the list of properties, and in the Toolbar section, for **Buttons**, uncheck **Save**. You won't need an additional **Save** button in this interactive grid, because you have one at the top of the page.
8. In the Component Tree, expand the **Columns** list below the Locations region. Select **LOCATION_ID** in the Component Tree.
9. In the Properties section on the right side of Page Designer, change **Type** to **Hidden** and change **Primary Key** to **Yes**. This instructs Oracle Application Express to *not* display this column in the interactive grid and to use the value of the column in this row for any data manipulation language (DML) operations (data modification actions).

10. Click the **Run** icon in the upper right.

Your application should look similar to **Figure 3**. You now have a page with three editable grids and can freely add, update, and delete rows in each of them. But the last remaining step for building this application is to specify the master/detail relationship between the Countries interactive grid and the Locations interactive grid.

Figure 3: The master/detail application with three editable grids



1. Click the **Quick Edit** link in the Developer toolbar. Hover your cursor over the **Locations** interactive grid and click. This will take you into Application Builder

with the Locations region selected.

2. In the Property Editor on the right side of Page Designer, in the Master Detail section, select **Countries** for **Master Region**.
3. In the Component Tree, expand the **Columns** list below the Locations region and select **COUNTRY_CODE**. In the Properties section on the right side of Page Designer, in the Master Detail section, select **COUNTRY_CODE** for **Master Column**. This creates the association between the COUNTRY_CODE column of your Locations interactive grid and the COUNTRY_CODE column of the Countries interactive grid.
4. Click the **Run** icon in the upper right.

In the updated application page, click various rows in the Regions interactive grid. Select a row in the Countries interactive grid, and use the arrow keys on your keyboard to navigate up and down the list of rows. You will see the rows in the Locations interactive grid automatically refresh.

CONCLUSION

The new interactive grid component in Oracle Application Express 5.1 makes it very easy to represent a variety of data relationships in a responsive web application. With very little code, you are able to create editable grids directly on top of your tables, preserving the parent/child/grandchild relationships. Although there are alternative ways to present tables with master/detail relationships in Oracle Application Express, multiple interactive grids on a single page will be very recognizable to users of legacy client/server applications. A future *Oracle Magazine* article will discuss other powerful interactive grid features in depth.

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.

NEXT STEPS

DOWNLOAD Oracle Application Express 5.1.

the SQL script for this article.

READ more about Oracle Application Express.

TRY Oracle Database Cloud services.

REQUEST a free Oracle Application Express workspace.

ORACLE CODE

Register Now

New One-Day, Free Event | 20 Cities Globally

Explore the Latest Developer Trends:

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

Live for the Code

Find an event near you:

developer.oracle.com/code

ORACLE®

By Melanie Caffrey

ORACLE DATABASE

The Cost of Data Retrieval

Part 9 in a second series on the basics of the relational database and SQL

This article is the ninth 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, "Rapid Retrieval of Rows in Small Data Sets," introduced B-tree indexes, both single-column and composite. You learned how they can be used to increase performance when retrieving rows. You discovered how to create them and what kind of relationship they have with constraints. You also discovered how to alter them, how to drop them, and when they *cannot* be used. Finally, you explored a few guidelines for when B-tree indexes are most useful, when they are not useful, and when a function-based index can be helpful.

In this article, you will

- Learn how to create a bitmap index
- Learn about SQL statement processing operations and parsing
- See Oracle Database's Oracle Optimizer program in action

- Discover database statistics
- Examine execution paths and explain plans

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

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

RETRIEVAL VERSUS REVISION

Recall that the previous Beyond SQL 101 article, "Rapid Retrieval of Rows in Small Data Sets," explained that one of the primary goals of index creation is to boost data retrieval performance. Additionally, you learned that B-tree indexes can be useful when your goal is to retrieve a small number of rows, using an index on column data that is highly selective. However, recall also that for each change of a value in an indexed column, any corresponding indexes must also be changed. Therefore, having indexes in place can potentially slow down data manipulation, because for every DELETE, UPDATE, and INSERT command issued that affects an indexed

column, a DELETE, UPDATE, and INSERT action must similarly take place against corresponding indexes to keep them in sync with the changed column values.

Data warehouses are types of databases in which the design and performance goals for data manipulation are different from those of a transactional database. In data warehouses, data loads take place in bulk and at certain intervals, and the primary goal of a data warehouse is to aid users in querying and analyzing data. To help users accomplish this goal, Oracle Database supports *bitmap indexes*, which are typically used on columns that have few distinct values relative to the number of rows in the table and, therefore, have *low selectivity*. For example, the EMPLOYEE table currently has only two distinct values for the virtual column WAGE_INCREASE_WORTHINESS: "No Raise Yet" and "Cost of Living Increase Eligible." If you have a data warehouse in which the users execute queries with a frequently used WHERE clause condition, including a column with few distinct values relative to the total number of rows in the table, this column may be a viable choice for a bitmap index. The structure of a bitmap index looks similar to **Figure 1**.

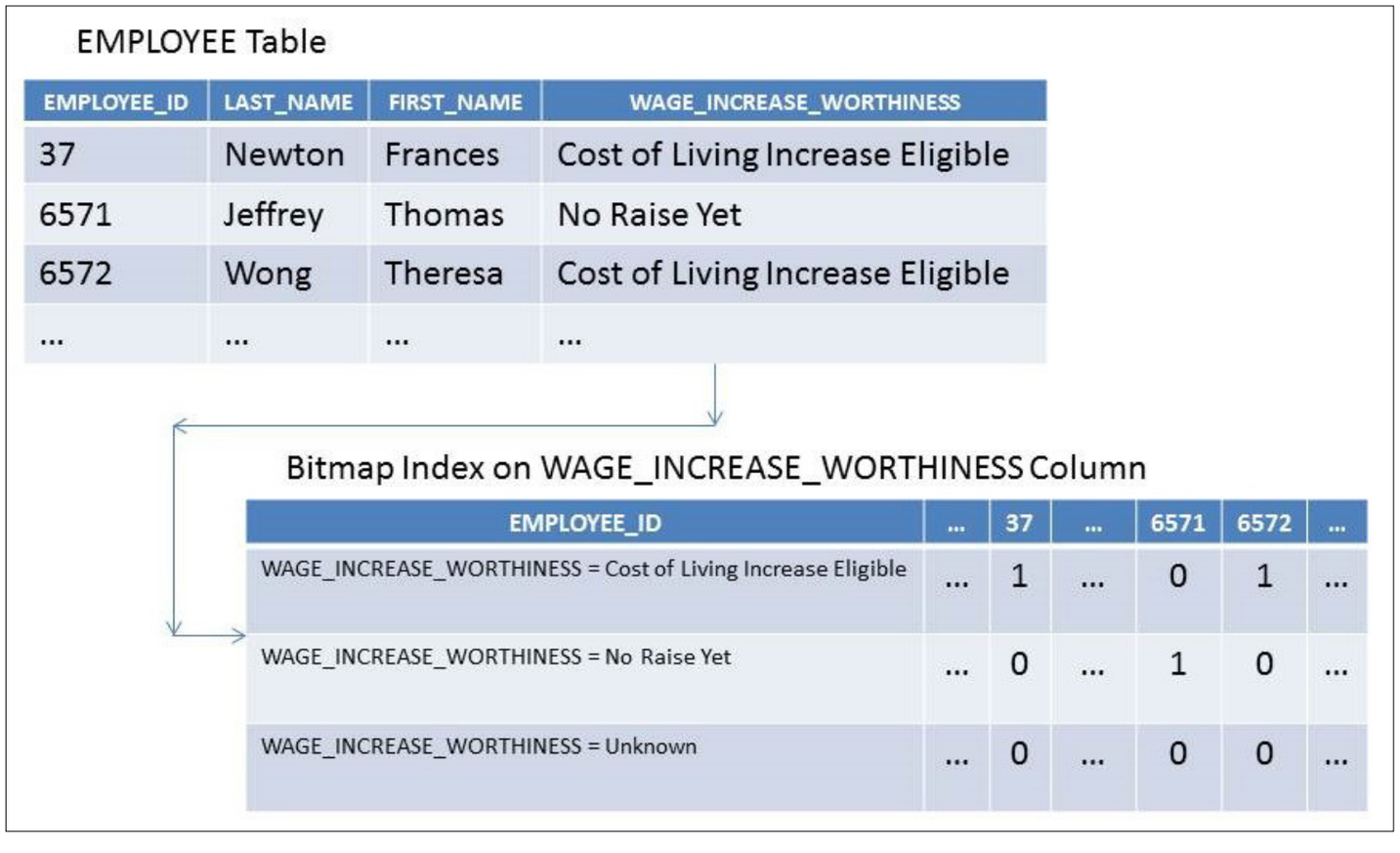


Figure 1: Structure of a bitmap index

Figure 1 shows the EMPLOYEE table with a bitmap index on the WAGE_INCREASE_WORTHINESS column. The bitmap index translates the distinct values for the WAGE_INCREASE_WORTHINESS column of individual

employees. In this simplified example, the employees Frances Newton (EMPLOYEE_ID=37) and Theresa Wong (EMPLOYEE_ID=6572) each have the value "Cost of Living Increase Eligible" for the WAGE_INCREASE_WORTHINESS column, which turns the bitmap index bit on to 1. The WAGE_INCREASE_WORTHINESS = No Raise Yet and WAGE_INCREASE_WORTHINESS = Unknown values for each of these employees have their index bits turned off to 0, indicating that these values are not true for those employee records. The employee Thomas Jeffrey (EMPLOYEE_ID=6571) has his index bit turned on to 1 for WAGE_INCREASE_WORTHINESS = No Raise Yet, and all other bits in his record are turned off to 0. The SQL in **Listing 1** demonstrates the creation of a bitmap index on the WAGE_INCREASE_WORTHINESS column of the EMPLOYEE table.

Code Listing 1: Creating a bitmap index on the WAGE_INCREASE_WORTHINESS column

```
SQL> set lines 10000
SQL> CREATE BITMAP INDEX employee_wage_inc_worth_bmi
  2  ON employee(wage_increase_worthiness);

Index created.
```

Note that whereas bitmap indexes can be useful in data warehouses, they are not suitable for transactional tables that experience heavy data manipulation activity by many users. Recall that changes to indexed column values require similar changes to the corresponding index values, so any changes to a bitmap index may significantly slow down transactions. However, if you have multiple single-column bitmap indexes in a data warehouse and you want to retrieve rows based on certain AND and/or OR WHERE clause conditions, bitmap indexes can boost performance.

Bitmap indexes are able to quickly compare and merge the bit settings that satisfy the WHERE clause conditions and return results without the need to access the large data warehouse table(s). Similarly, if your WHERE clause condition consists of a single-column predicate on a bitmap-indexed column, the bitmap index will be most useful when you are also executing a COUNT aggregate function in your SQL statement, because the index obtains and aggregates the ON bits and returns the result without having to access the underlying table. Additionally, note that although bitmap indexes typically require less storage space than B-tree indexes, they are available only with the Enterprise Edition of Oracle Database.

MAKING A STATEMENT

Indexes can improve the performance of SQL statements returning result sets, but before a SQL statement returns a result set, Oracle Database performs several steps and operations that are transparent to the user. First, Oracle Database creates a *cursor*, an area in memory where the database stores the SQL statement. Next, it *parses* the SQL statement. Parsing involves several steps:

- Check the syntax validity. This step ensures that the SQL statement has been written correctly.
- Check that the table names and column names included in the SQL statement exist.
- Check whether the user has permissions to access the requested tables and columns. (Permissions and grants will be discussed in a future article in this series.)

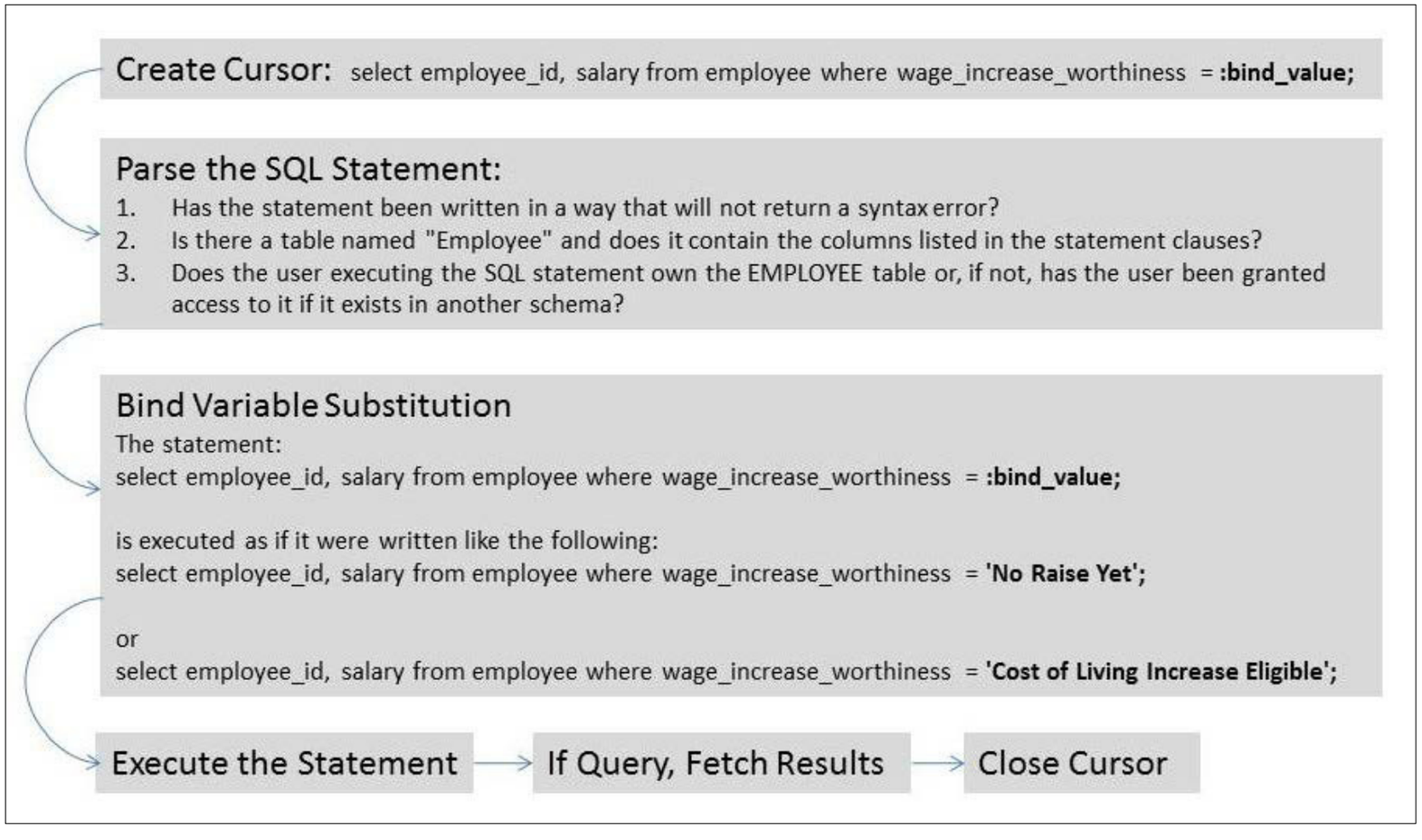
Parsing is an expensive operation that requires time and resources. Therefore, for repeatedly executing similar statements, there are ways to reduce or eliminate parsing.

Oracle Database maintains a cache of recently executed SQL statements, along with their execution plans. When a new SQL statement is executed, if an identical statement has been previously parsed, the statement doesn't need to be parsed again.

Bind variables, placeholders for values, are very useful in reducing the number of necessary parsing actions. These placeholders are substituted by different values when a SQL statement is parsed. If bind variables are used, they are associated with and substituted for the appropriate statement values. Once the bind variable substitution takes place, the SQL statement is executed.

For any SQL statement query, a cursor is created; the statement is parsed; bind variables are substituted; the statement is executed; results are fetched; and after all results are fetched, the cursor is closed. **Figure 2** illustrates an overview of SQL statement processing operations.

Figure 2: SQL statement processing operations



STATISTICALLY SELECTIVE

Part of the parsing operation is the determination of the *execution plan*, the method by which Oracle Database intends to execute the SQL statement. To determine the best way to execute a SQL statement, Oracle Database employs a program called the *Oracle Optimizer*, which examines each SQL

statement and determines the most optimal sequence of steps necessary to execute it. This sequence of steps is the execution plan.

When determining the best plan, the Oracle Optimizer considers *statistics* stored in the Oracle Database *data dictionary*. Recall from a previous Beyond SQL 101 article, "Defining, Constraining, and Manipulating Your Entities," that the data dictionary is an internally owned and controlled set of tables and views that contain information about your database's metadata.

Statistics include, among other pieces of information, values such as the number of rows in a table and the number of distinct values for each column in a table. The statistics are used by the Oracle Optimizer to calculate the estimated cost of various execution options to help determine the lowest-cost (and therefore, best) execution plan. Automatic statistics gathering may or may not be enabled. You can use Oracle Enterprise Manager to enable or disable statistics gathering, check for missing or stale statistics, and populate the data dictionary with new or updated statistics when necessary.

The query in **Listing 2** checks when tables owned by your schema last had statistics gathered on them. The query in **Listing 3** shows when indexes owned by your schema last had statistics gathered on them.

Code Listing 2: Determining when statistics were last gathered for your schema's tables

```
SQL> select table_name, num_rows, to_char(last_analyzed, 'DD-MON-YYYY HH24:MI:SS')
last_analyzed
  2    from user_tables
  3  order by table_name;
```

```
TABLE_NAME                                          NUM_ROWS LAST_ANALYZED
_____________________________________________ _____________ ____________________
ANNUAL_REVIEW                                             15 05-JUL-2016 01:00:23
DEPARTMENT                                                 4 05-JUL-2016 01:00:24
EMPLOYEE                                                  16 04-JAN-2017 01:00:32
EMPLOYEE_CTAS                                              2 11-SEP-2016 16:21:22
EMPLOYEE_EXTRA                                             2 26-OCT-2016 01:00:32
EMPLOYEE_IDENTITY                                         16 04-JAN-2017 01:00:32

6 rows selected.
```

Code Listing 3: Determining when statistics were last gathered for your schema's indexes

```
SQL> select table_name, index_name, num_rows, distinct_keys, to_char(last_analyzed,
'DD-MON-YYYY HH24:MI:SS') last_analyzed
  2    from user_indexes
  3  order by table_name, index_name;

TABLE_NAME   INDEX_NAME                      NUM_ROWS DISTINCT_KEYS LAST_ANALYZED
____________ _____________________________ _________ _____________ ____________________
DEPARTMENT   DEPARTMENT_NAME_LOCATION_UK           4             4 10-SEP-2016 21:05:26
DEPARTMENT   DEPARTMENT_PK                        4             4 05-JUL-2016 15:45:44
EMPLOYEE     DEP_WAGE_INCREASE_I                16             7 15-JAN-2017 16:39:23
EMPLOYEE     EMPLOYEE_PK                        16            16 15-JAN-2017 17:56:41
EMPLOYEE     EMPLOYEE_WAGE_INC_WORTH_BMI        2             2 06-MAR-2017 19:21:37
EMPLOYEE     EMP_DEPT_FK                        16             4 15-JAN-2017 18:39:09
EMPLOYEE     EMP_HIRE_DATE_I                    16            13 11-JAN-2017 21:53:14
```

```
EMPLOYEE_EXTRA SYS_C0010551                    2              2 26-OCT-2016 01:00:32

8 rows selected.
```

A PLAN OF EXECUTION

Using statistics, the Oracle Optimizer creates an execution plan that outlines each step Oracle Database executes to process a SQL statement. Oracle Database reads the execution plan and executes each step, starting with the most indented step and ending with the least indented step. If two or more steps have the same level of indentation, the steps are executed, in order, from top to bottom. Consider the execution plans, sometimes called *explain plans*, in **Listing 4**.

Code Listing 4: Creating and displaying execution plans

```
SQL> EXPLAIN PLAN FOR
  2     select employee_id, last_name, first_name
  3       from employee
  4      where wage_increase_worthiness = 'No Raise Yet'
  5     order by last_name, first_name;

Explained.

SQL> select *
  2    from TABLE(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
```

```
Plan hash value: 1993616227

-------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |     5 |   230 |     4  (25)| 00:00:01 |
|   1 |  SORT ORDER BY     |          |     5 |   230 |     4  (25)| 00:00:01 |
|*  2 |   TABLE ACCESS FULL| EMPLOYEE |     5 |   230 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------

   2 - filter("WAGE_INCREASE_WORTHINESS"='No Raise Yet')

14 rows selected.

SQL> EXPLAIN PLAN FOR
  2     select employee_id, last_name, first_name
  3      from employee
  4     where wage_increase_worthiness = 'Cost of Living Increase Eligible'
  5     order by last_name, first_name;

Explained.
```

```
SQL> select *
  2    from TABLE(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

Plan hash value: 1993616227


--------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |    11 |   506 |     4  (25)| 00:00:01 |
|   1 |  SORT ORDER BY     |          |    11 |   506 |     4  (25)| 00:00:01 |
|*  2 |   TABLE ACCESS FULL| EMPLOYEE |    11 |   506 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

   2 - filter("WAGE_INCREASE_WORTHINESS"='Cost of Living Increase Eligible')

15 rows selected.
```

The execution plans in **Listing 4** are created by use of the EXPLAIN FOR syntax. A query to the DISPLAY function in the built-in DBMS_XPLAN PL/SQL package obtains

each execution plan. (An explanation regarding PL/SQL or the TABLE function that encapsulates the call to DBMS_XPLAN is beyond the scope of this article. Both the TABLE function and an introduction to PL/SQL will be addressed in subsequent articles in this series.) Note that the ID column identifies only the step number, much like a step name; it does not denote the execution order of the steps. The first step performed is a full table scan of the EMPLOYEE table to retrieve all records. The next step sorts the returned records according to the ORDER BY clause, and the final step of the first execution plan in **Listing 4** returns all rows that satisfy the WHERE clause condition predicate

```
2 - filter("WAGE_INCREASE_WORTHINESS"='No Raise Yet')
```

The ROWS column of the execution plan displays approximately how many rows the Oracle Optimizer expects to process at each step. Recall from the previous Beyond SQL 101 article, "Rapid Retrieval of Rows in Small Data Sets," that if your query warrants the return of a large number of row values from your table, it might be more efficient to scan the entire table to return the row values than to look up and obtain row values from an index. Because the EMPLOYEE table is currently composed of only 16 records, the first execution plan in **Listing 4**'s approximation of 5 records returned comprises more than a quarter of the records in the entire table. For comparison purposes, the second execution plan in **Listing 4** is generated for the same SQL statement, but for the WHERE clause condition predicate

```
2 - filter("WAGE_INCREASE_WORTHINESS"='Cost of Living Increase Eligible')
```

And although the ROWS column value differs between the two execution plans, the execution plan steps are identical.

Code Listing 5: An execution plan that uses an index key lookup

```
SQL> EXPLAIN PLAN FOR
  2  select last_name, first_name
  3    from employee
  4  where hire_date > to_date('01-SEP-2016', 'DD-MON-YYYY')
  5  order by last_name, first_name;

Explained.

SQL> select *
  2    from TABLE(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

Plan hash value: 1409198325


--------------------------------------------------------------------------------
| Id  | Operation         | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |                 |     1 |    23 |     3  (34)| 00:00:01 |
|   1 |  SORT ORDER BY    |                 |     1 |    23 |     3  (34)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID BATCHED| EMPLOYEE| 1| 23| 2   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN | EMP_HIRE_DATE_I |     1 |   |     1   (0)| 00:00:01 |
```

```
_____________________________________________________________________________

PLAN_TABLE_OUTPUT
_____________________________________________________________________________

Predicate Information (identified by operation id):
___________________________________________________

   3 - access("HIRE_DATE">TO_DATE(' 2016-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))

15 rows selected.
```

Listing 5 outlines an execution plan that shows an index access included as an execution step. In the first step executed, step 3, INDEX RANGE SCAN searches the EMP_HIRE_DATE_I index for the accounting department record and retrieves the DEPARTMENT_ID value for the approximate number of rows that meet the WHERE clause condition predicate

```
3 - access("HIRE_DATE">TO_DATE(' 2016-09-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

Note that a predicate line that states that it is an “access” performs an index key lookup whereas a predicate line that states that it is a “filter,” such as both predicates in **Listing 4**, is simply a WHERE clause condition used to filter the returned results.

The next step in **Listing 5**, step 2, accesses the EMPLOYEE table with the ROWID value(s) retrieved from the EMP_HIRE_DATE_I index. Step 1 sorts the returned records according to the ORDER BY clause, and the final step, step 0, returns all rows that satisfy the WHERE clause condition.

Listing 6 demonstrates an execution plan that uses the EMPLOYEE_WAGE_INC_WORTH_BMI bitmap index. Note that because the query uses a single predicate on the bitmap-indexed column and is performing the COUNT aggregate function, the answer can be obtained from the bitmap index alone and a table access is, therefore, not necessary.

Code Listing 6: An execution plan that uses a bitmap index due to the COUNT aggregate function

```
SQL> EXPLAIN PLAN FOR
  2  select COUNT(employee_id) employees
  3    from employee
  4   where wage_increase_worthiness = 'No Raise Yet';

Explained.

SQL> select *
  2    from TABLE(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

Plan hash value: 1431199220


--------------------------------------------------------------------------------
| Id  | Operation         | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------

PLAN_TABLE_OUTPUT
```

```
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT |                          |     1 |    26 |     1   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE  |                          |     1 |    26 |            |          |
|   2 |   BITMAP CONVERSION COUNT  |                |     6 |   156 |     1   (0)| 00:00:01 |
|*  3 |    BITMAP INDEX FAST FULL SCAN| EMPLOYEE_WAGE_INC_WORTH_BMI |    |  |  |  |
-------------------------------------------------------------------------------------------------

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------


   3 - filter("WAGE_INCREASE_WORTHINESS"='No Raise Yet')

15 rows selected.
```

SUMMARY

This article introduced you to bitmap indexes. You discovered how to create them and when they might be an appropriate index choice. You also learned how bitmap indexes should *not* be used in a database with regular transaction activity but instead in a data warehouse with occasional bulk loads of data. You were introduced to the Oracle Optimizer, SQL statement processing operations, and parsing. You learned what a bind variable is and how it can help reduce parsing activity. Last, you learned about statistics and the creation and display of execution plans. In the next article in this series, you'll learn more about the Oracle data dictionary and be introduced to SQL statement scripting.

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

PHOTOGRAPHY BY **RAY NG**

NEXT STEPS

READ

SQL 101, Parts 1–12.

more Beyond SQL 101.

DOWNLOAD

the sample script for this article.

Oracle Database 12*c* Release 2.

LEARN more about relational database design and concepts.

TRY Oracle Database Cloud services.

Database 12*c*

Built for the Cloud

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

ORACLE®