ORACLE MAGAZINE

MARCH/APRIL 2019

GOING AUTONOMOUS

Learn what automatic indexing and other advanced capabilities of Oracle Database 19c will do for your business

READY-TO-GO AI

THREE KEYS TO CLOUD SECURITY

THE NEW AGNOSTIC APPLICATIONS

ORACLE®

Vertex + Oracle

Global Tax Solution Partners

+ JDEdwards EnterpriseOne
+ Oracle® ERP Cloud
+ Oracle® E-Business Suite
+ Oracle® NetSuite
+ Oracle® Peoplesoft

Come visit us at Collaborate19

April 7-11 | San Antonio, TX

VERTEX® | ORACLE Platinum Partner Cloud Standard

CONTENTS

ORACLE MAGAZINE

31 Database Innovation

FEATURES

UP FRONT

11 Interview

CONTENTS

COMMUNITY

TECHNOLOGY

COMMENT

ORACLE MAGAZINE

SUBSCRIPTION INFORMATION

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

MAGAZINE CUSTOMER SERVICE

Omeda Communications

PRIVACY

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

Tom Haunert

The Next Database Story

The headline for each Oracle Database release starts a conversation, but there's always more to say.

Database articles have always filled the pages of *Oracle Magazine,* but how we look at new database releases has changed in the last couple of years. Cloud and autonomous have been the focus of recent Oracle Database articles, and rightly so. In this issue, however, we're reminding readers that there's much more to learn in every release, particularly as it pertains to the future of their business.

In our "Database Innovation" cover feature, Oracle Content Central Editor-at-Large Jeff Erickson describes the top new features of the latest release, Oracle Database 19c. Among them: automatic indexing, which improves database performance by making sure queries use the best and most up-to-date indexes; Active Data Guard DML Redirect, which makes your standby database *more active* and a better value for your business; Hybrid Partitioned Tables, a feature that enables partitions to reside in Oracle Database segments as well as in external files; partial updates for JSON documents in the database; and updated Simple Oracle Document Access (SODA) APIs.

CONTINUING INNOVATION AND SECURING THE FUTURE

Beyond the catalog of new features, Oracle Database 19c offers customers a commitment to more-regular, continuous

innovation, now that Oracle has moved from major database releases every two to three years to annual releases (thus the *19*, for *2019*, in the product name). For those planning to upgrade from on-premises Oracle Database 11g and 12c to Oracle Database 19c, the most important news may be this: 19c is a long-term-support release, with premier support planned through March 2023 and extended support through March 2026.

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/GETTY IMAGES**

NEXT STEPS

READ "Database Innovation."

EXPLORE Oracle Database 19c new features.

TRY Oracle Live SQL (Oracle Database 19c).

Address the ELEPHANT IN THE ROOM

Bad address data costs you money, customers and insight.

Melissa's 30+ years of domain experience in address management, patented fuzzy matching and multi-sourced reference datasets power the global data quality tools you need to keep addresses clean, correct and current. The result? Trusted information that improves customer communication, fraud prevention, predictive analytics, and the bottom line.

- Global Address Verification
- Digital Identity Verification
- Email & Phone Verification
- Location Intelligence
- Single Customer View

See the Elephant in Your Business -

Name it and Tame it!

melissa

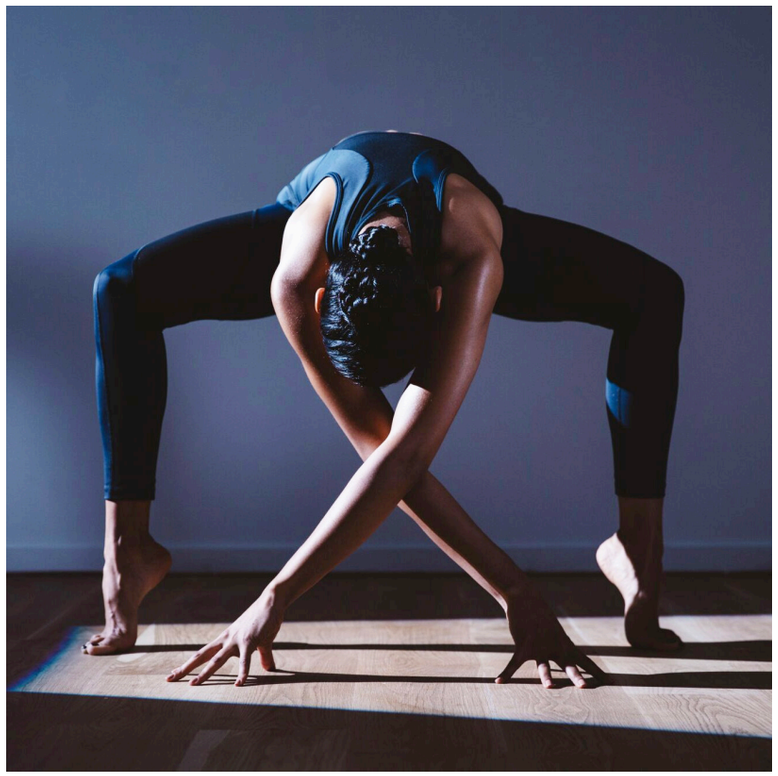
www.Melissa.com | 1-800-MELISSA

Free API Trials, Data Quality Audit & Professional Services.

Put It On!

Gadgets, apps, opinions, and terms from the wearables world

Nadi X



Pull on a pair of these ultrawearable yoga pants, and connect them via Bluetooth to the free Nadi X iOS app to get sensor-based feedback and help on your yoga poses. Nadi X guides your yoga practice through state-of-the-art technology, based on your body's alignment. Listen to the audio instructor on your phone, and feel the guidance on your skin via the pants' integrated sensors and haptic feedback (vibration). Detailed instructions break down each yoga pose step by step. The washable pants come in men's and women's styles and sizes. US$249. Wearable X

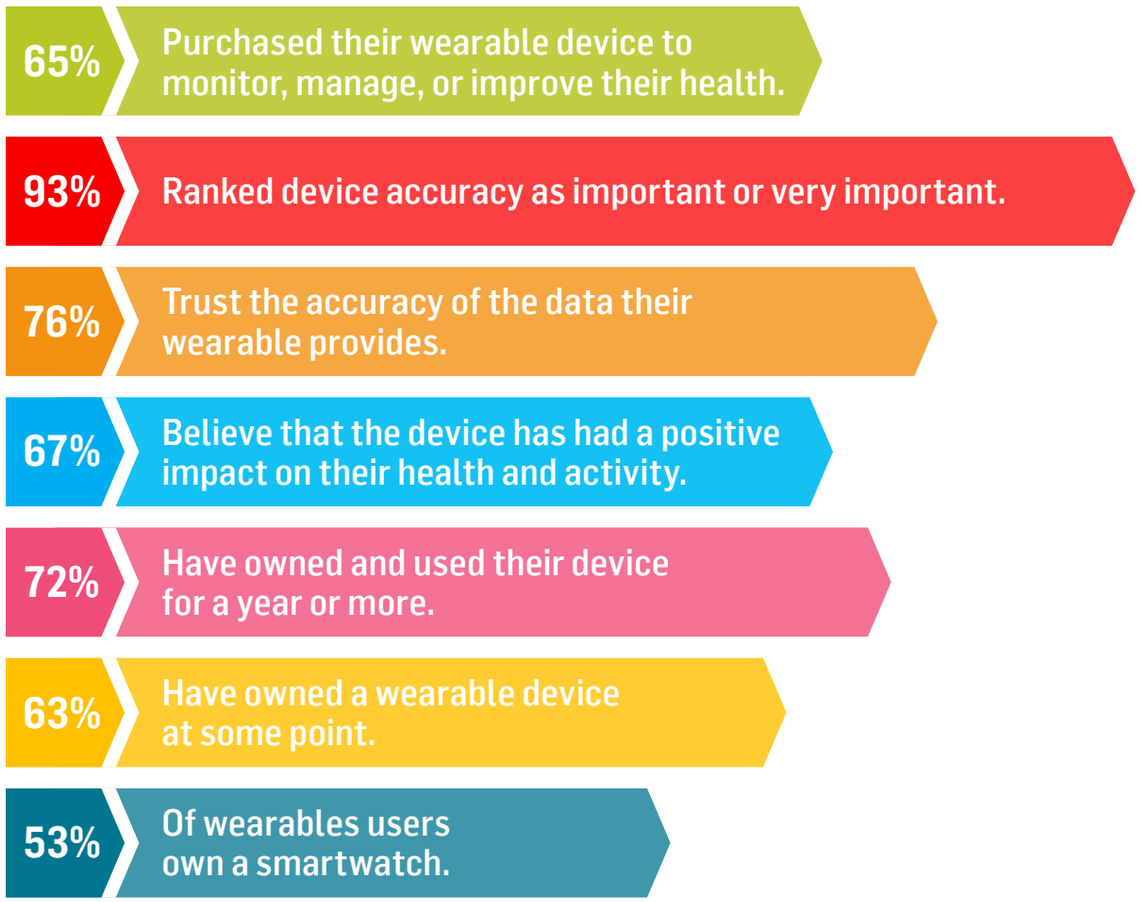
RunScribe Plus

For athletes who want to know everything about their runs, here's a footpod that provides 3D insight. RunScribe Plus captures every footstep and offers powerful analysis tools for analyzing metrics, comparing runs, and tracking the impact of shoes and terrain on stride mechanics. The system provides real-time metrics via iOS, Apple Watch, and compatible Garmin watches, and you can upload run data onto your iOS or Android device. Bonus: RunScribe offers custom development tools and full source code for developers. US$249 (includes footpods, lace cradles, charger, USB cable, iOS or Android app, and RunScribe dashboard account). RunScribe

PHOTOGRAPHS COURTESY OF WEARABLE X, RUNSCRIBE

Wearables: A Healthy Habit?

Wearable device adoption has jumped by 21% since 2016, according to a recent survey, and a majority of wearables owners purchased their device for health-related purposes such as disease management and fitness. The 2018 survey, which polled 826 US consumers between the ages of 18 and 65, also found that although users are keeping their devices longer and deriving more value from them, data accuracy remains a top priority.



Source: The 2018 "The State of Wearables Today" survey

DO YOU SPEAK TECH? QUIZ YOURSELF!

1. Qi ("chee") is

A. The universal standard for wireless charging of battery-operated devices

B. The Chinese term for life force or energy flow

C. The sound of a DBA holding back a sneeze in a dusty server room

D. All of the above

E. None of the above

2. Measuring body composition and water content by passing an electrical current through the body is called

A. Qi Finder

B. Bioimpedance

C. Torture, as defined by the United Nations Convention Against Torture

3. When you employ smart speakers, smart thermostats, wearable devices, and the like, you are

A. Joining the industrial IoT revolution

B. Really cool

C. Using ambient computing

Answers: 1. D, 2. B, 3. C

WATCH YOURSELF!

Smartwatch apps for tracking health and fitness

Lifesum

Boost your fitness routine with a personalized diet plan, food tracker, and calorie counter plus healthy recipes with this highly rated smartphone/smartwatch app. Scan barcodes to add calorie and nutrition information to your macro tracker, experiment with vegan and keto plans, monitor your water intake, track your activity via smartphone or smartwatch to get accurate feedback and advice, and more. Note: Although the app offers premium options, the free "classic" app works well without them. Free (iOS, watchOS; Android, Wear OS)

Strava

Billed as "the #1 app for runners and cyclists," Strava has added wearables to the list of devices it supports. Track and measure every activity; record countless performance metrics; share your activity along with photos and captions; broadcast your location in real time to friends, emergency contacts, and coaches; tap into a road and trail network mapped by millions of athletes around the world; and sync data to a long list of compatible devices. Note: The latest release offers additional stats for wheelchair and handcycle activities. Free (iOS, watchOS; Android, Wear OS)

Seven's Seven-Minute Workout

Workouts getting a bit stale? Add fun to your fitness routine with Seven's personalized workout plans, based on scientific studies to provide the maximum benefit in the shortest time possible. Simply set your goals and fitness level, and the app provides tailored, seven-minute workout routines that you can launch from your smartphone or smartwatch and do anywhere, with no equipment required. The app includes interactive 3D guides, progress tracking, and achievement awards to motivate your advancements. Free (iOS, watchOS; Android, Wear OS)

INTERVIEW

"For AI solutions to adapt and learn, they have to assess the efficacy of their recommendations very rapidly and frequently," says Clive Swan, senior vice president of Oracle Adaptive Intelligent Apps at Oracle.

Ready-to-Go AI

Make AI part of your core ERP and HCM applications. **BY TOM HAUNERT**

From web store recommendation engines to autonomous cars and databases, artificial intelligence (AI) is already part of your personal life and business processes. But AI is also part of today's business applications, and it's assuming an ever-larger role in time-saving, money-saving, and money-generating business processes.

Oracle Magazine sat down with Clive Swan, senior vice president of Oracle Adaptive Intelligent Apps at Oracle, to talk about evaluating when to use AI, measuring the value of AI in the enterprise, Oracle's AI for apps strategy, and more.

Oracle Magazine: How do businesses evaluate or consider various emerging technology application and platform choices?

Swan: The first thing is, as with any other technology, they've got to make choices based on business needs and not just choose a new technology for technology's sake. Businesses need to identify target use cases with a target end state in mind. They need to identify where that emerging technology can apply, and then they need to plan to adopt that technology, based on the skill sets they have available, their attitude about risk, and how quickly they want to achieve a win.

Businesses also often need to make a decision between build and buy. Do they have the resources in-house and access to the specialist skill sets required to build it themselves? For example, data scientists—an expensive and rare commodity—are needed to build AI solutions. So even if a business does have ready access to data scientists, the company still needs to decide whether a given use case is sufficiently differentiating to the business to warrant using those data scientists to develop a solution in-house instead of buying one off the shelf.

Whether businesses build or buy, there are three components that are core to how AI solutions work. There's the data from the applications, there's the real-time context in the applications at that point in time, and then there's measuring the outcome in the applications. For AI solutions to adapt and learn, they have to assess the efficacy of their recommendations very rapidly and frequently.

Businesses can buy third-party AI point solutions, and those solutions can be loosely

integrated with ERP and human capital management (HCM) suites, for example. But in many cases, those point solutions will be suboptimal, because they can't be integrated deeply enough within the base applications to have full and timely access to the three core AI data components I mentioned. Alternatively, AI solutions can be a part of the core ERP and HCM suites, as they are with Oracle's offerings, with tightly connected AI component processes, leading to optimal AI performance and requiring no additional integration work.

"We're shifting the user experience from something user-initiated and user-driven to AI push-driven activity."

There are cases where it makes sense to build a new custom AI application based on the type of data and how it will be used. For example, car insurers can train a neural net with photographs of thousands of cars that have been in crashes to determine which cars are most likely a write-off. In such cases, the data and context are lightly coupled with the base application, so a custom standalone AI application makes sense.

Oracle Magazine: How do companies look at or measure the value of AI in back-office applications?

Swan: The preliminary benefits can often be measured quite readily, but the related benefits, which are often more substantial, are much harder to measure and probably aren't even recognized in many cases.

First, looking at AI today, the majority of solutions are not doing something that a human being couldn't do. But AI solutions do a better job, because they do those tasks very quickly; never take a tea break; and execute predictably, the same way, time after time.

So I would argue that in many cases, when people start measuring the benefit of an AI solution, that benefit is measured in terms of productivity gains. But there are additional positive side effects—employees' spending less time on mundane tasks results in improved employee retention and frees the employees up to make more-strategic contributions to the business.

For example, consider a recruiter who needs to find 10 candidates for further phone

"In many cases, when people start measuring the benefit of an AI solution, that benefit is measured in terms of productivity gains," says Clive Swan, senior vice president of Oracle Adaptive Intelligent Apps at Oracle.

screening from 100 résumés submitted. That recruiter might take two hours to review 50 résumés and then pick the best 10 from that 50. An AI solution doing just as good a job as the recruiter in ranking those résumés will scan all 100 résumés and statistically find a better set of 10 candidates. And it will do it in a few minutes, plus the recruiter will have gained two hours to spend on more-valuable tasks, such as selling the company to potential candidates.

Oracle Magazine: What is Oracle's apps AI strategy?

Swan: We term Oracle's apps AI strategy "pervasive AI," and there are three components to it.

First, we've got Oracle Adaptive Intelligent Apps, which are ready-to-go, out-of-the-box,

fully integrated, functional application extensions that are typically involved in making recommendations and decisions for professional users, consumers, and end users in our applications. We're delivering these solutions across the enterprise application suite, including B2C, customer experience, B2B CX, ERP, HCM, and supply chain.

To give a few examples: In Accounts Payable, the app recommends to the accounts payable professional which suppliers will generate the most savings if paid early. In Sales, the app can predict the likelihood of a sales rep closing a given deal and recommend the next actions to take to increase the likelihood of winning that deal. Commerce apps recommend products for consumers, and marketing apps recommend the time of day for sending a communication to a consumer and over what channel.

Then, we've got intelligent UX [user experience]. We're shifting the user experience from something user-initiated and user-driven to an AI push-driven activity in which AI nudges tell users what they need to know and what they need to do.

And third, we've got virtual assistants. We're bringing the conversational interaction models from consumer apps into enterprise apps. These solutions use AI, of course, not only to map voice to text but also to interpret what the request is and then serve up the best response, taking into account the context of the requester.

And all of these solutions are underpinned by Oracle's smart-data strategy, which enriches application data with third-party, web-scale, trusted data, enabling our AI solutions to make even better decisions.

PHOTOGRAPHY BY **JOHN BLYTHE**

NEXT STEPS

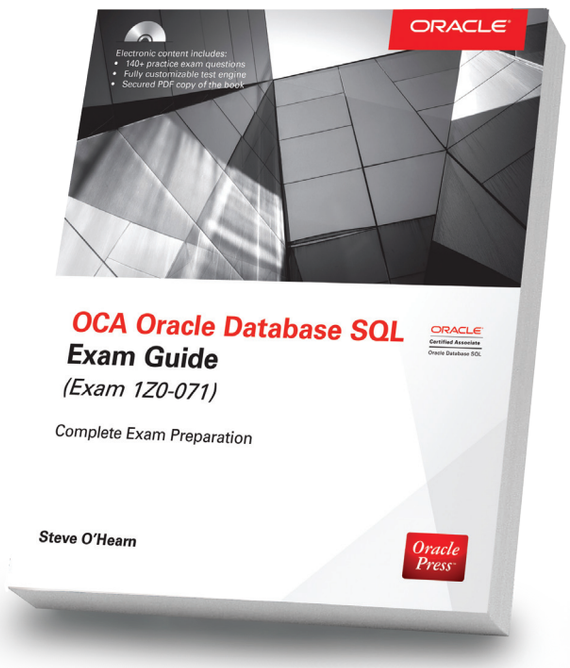
LEARN more about AI and machine learning.

LEARN more about Oracle Adaptive Intelligent Apps.

Oracle Press™

Your Destination for Oracle and Java Expertise

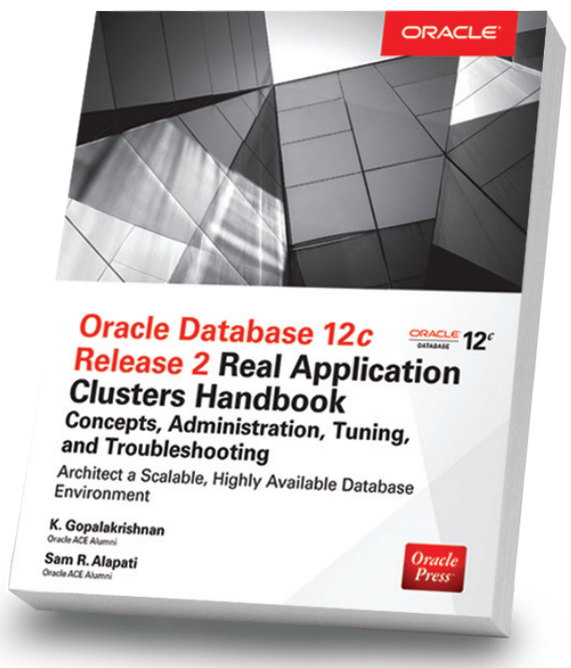
Written by leading technology professionals, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Oracle products and technologies available.



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

Steve O'Hearn

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



Oracle Database 12c Release 2 Real Application Clusters Handbook

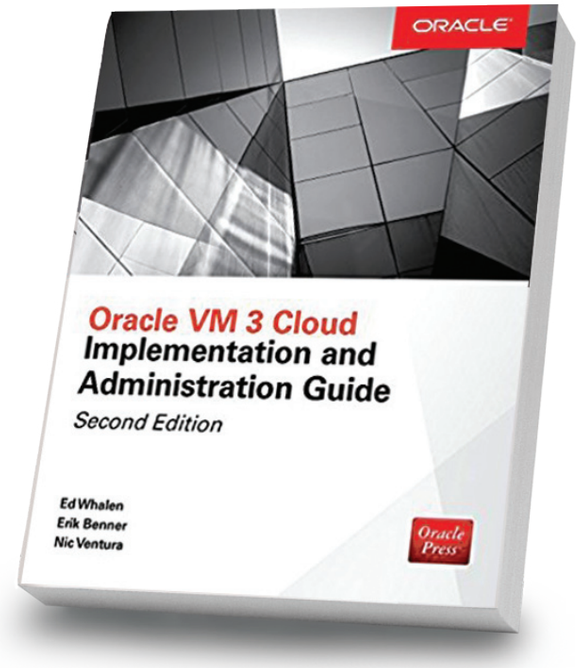
K. Gopalakrishnan, Sam R. Alapati

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

Oracle Mobile Cloud Service Developer's Guide

John Ray Thomas

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



Oracle VM 3 Cloud Implementation and Administration Guide, Second Edition

Ed Whalen, Erik Benner, Nic Ventura

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

Available in print and ebook formats.

Oracle Press™

www.OraclePressBooks.com • @OraclePress

By **Bob Rhubart**

Speaking the Language

Oracle Groundbreaker Ambassador Gail Anderson translates expertise into opportunity.

How has a degree in linguistics from University of California San Diego helped Gail Anderson in her successful career as a specialist in Java, C++, C, Python, Perl, and various other technologies?

"You would think that it would be more helpful than it really was," Anderson says. "Linguistics is the study of human languages, but certainly some of the things apply to artificial languages as well. But I don't see a really strong influence."

Although any direct influence on her work may be fuzzy, it was linguistics that brought Anderson to University of California San Diego, where she signed up for a programming class to explore interests beyond her selected major.

Oracle Groundbreaker Ambassador Gail Anderson trains professionals and encourages a new generation of programmers.

RECOGNITION

The Oracle Groundbreaker Ambassador program recognizes modern experts who blog; write articles; and give presentations on topics such as containers, microservices, SQL, NoSQL, open source technologies, machine learning, and chatbots. Learn more, and follow the Oracle Groundbreaker Ambassadors.

The class Anderson took focused on ALGOL, and this was at a time when programming was still very new, as Anderson explains, before gaming technology and smart devices. "Programming was mostly done in scientific and math circles. I enjoyed my math classes, but I was a linguistics major and I was more interested in human learning and in languages. I found computer programming interesting to do, but I didn't have a good feel for how I would apply it. I learned more of that later when I used it more."

That opportunity came with her first job, running the on-campus UNIX machine at the computer center at UC Santa Barbara. "The thing that has stayed with me from that experience is how pervasive that technology remains today. They came up with the idea of pipes. You have commands that would do one thing, but they were written in a way that they could get their input from the output of preceding commands. And you could string these commands together in very unique ways without knowing ahead of time *how* they would be strung together, just that they could be strung together."

Anderson sees similarities in today's Java streams. "Core streams work with language, and pipes were an OS feature. But I marvel today when people talk about 'open your terminal window.' Whether you're on a Mac or a Red Hat or Bluetooth system, you're using a flavor of UNIX with the command-line shell. That's old technology, but it still persists."

These days Anderson and her husband, Paul, a fellow Groundbreaker Ambassador and Java Champion, stay busy sharing their expertise through training classes under the auspices of their company, Anderson Software Group; through copresenting at Oracle Code One, Devoxx, and similar conferences around the world; and through a series of books on which they have collaborated, including *JavaFX Rich Client Programming on the NetBeans Platform* (Addison-Wesley, 2015).

Anderson attributes her success to the many people she has encountered along the way. “When Paul and I started talking at conferences, we started meeting other people, and that was very helpful to us.” She singles out Geertjan Wielenga, senior principal product manager for Oracle Developer Tools, and JClarity CTO, Kodewerk LTD principal consultant, and fellow Groundbreaker Ambassador and Java Champion Kirk Pepperdine as having been instrumental in introducing the Andersons to the Java community. She is passionate about paying it forward through efforts such as the 2018 JCrete4Kids event.

“That was one of the most rewarding experiences, working with 9-, 10-, 11-year-old girls and seeing them excitedly complete a programming assignment. It made me realize that girls are just as interested in computers as boys.” The experience was a reminder that women are underrepresented in computer science. “Working with these girls made me happy to encourage girls to enter computer science.”

Anderson’s passion for translating her skills and knowledge into opportunities for others is a defining characteristic of her career. That’s language we can all understand.

Oracle Architect Community Manager Bob Rhubart is the host-engineer/producer of the Oracle Groundbreakers Podcast series; produces the 2 Minute Tech Tip video series; and interviews technology experts in DevLIVE videos recorded at Oracle events.

PHOTOGRAPHY BY **RAFFI ALEXANDER**

NEXT STEPS

LEARN more about Anderson Software Group.

WATCH *Are You Listening? JavaFX Binding Techniques for Rich Client UIs*.

WATCH *Swing Away! Move to JavaFX 8 and the NetBeans Platform*.

By **Alexandra Weber Morales**

How a CTO Finds Flow

Groundbreaker Ambassador Luis Weir's productivity tips

As a globe-traveling software executive, London-based Luis Augusto Weir has seen service-oriented architectures (SOAs) evolve from complexity to ubiquity. The old SOA model has been reborn via several approaches to middleware and APIs—and he's happy to be part of the booming API economy.

Weir has penned several books on middleware, including the forthcoming *Enterprise API Management* (Packt Publishing, July 2019), thanks to his experience with building SOA, middleware, and API solutions for Fortune 500 companies. And he started young, building one of the country's first social media websites in his native Venezuela as well as starting a

Oracle Groundbreaker Ambassador Luis Weir sets directions, understands digital natives, networks, and avoids exhaustion.

small development firm before he finished university. How does he stay so productive while keeping current with changing technology?

SET THE DIRECTION

"I'm the CTO, so I manage everyone but I don't manage anyone," laughs Weir. As chief technology officer for Capgemini's Oracle Practice Cloud Solutions team, "I have four projects going on right now, so I'm indirectly managing several people." His role is to set the technology direction, defining where his practice should invest, based on market understanding and then engaging the technology team leads to ensure that staff is trained and enabled. He does this by getting his hands dirty, so to speak.

"I think the era of software architects who just do PowerPoint is gone. I use it to sell a concept or an idea and to draw diagrams," Weir says. But the main way to inspire is by example, he believes. "More and more I see CTOs who are like me. A good example is Lucas Jellema @lucasjellema, CTO of Amis, who is even more hands-on than I am."

UNDERSTAND DIGITAL NATIVES

Getting hands-on is also key to managing the next generation of developers, according to Weir. "What we get from university is young talent who want to get inspired and play with cool tech. They are digital-born; they've never known anything else." However, he feels it's also important to guide developers toward not trying to reinvent the wheel. "For many, especially if you are inexperienced, you try to solve a problem from scratch without investigating whether others have solved the problem already," he says.

Although "managing conflict is a whole other topic," he says, "one thing I am always battling with is design decisions. I hate accidental decisions." He misses the focus on software architecture that he says used to be more prevalent.

"Developers today tend to forget about architecture a little bit. I love

“Developers today tend to forget about architecture a little bit. I love developing, but making the right decisions can save a lot of time and effort down the line.”

—Luis Weir, Oracle ACE Director and Groundbreaker Ambassador

developing, but making the right decisions can save a lot of time and effort down the line,” Weir says. That’s why he has what he calls a “red-tape process for design decisions,” where he asks questions such as “Why are you using JavaScript as opposed to Java?” or “Why did you choose REST APIs over GraphQL APIs?” The key, he says, is to keep an open mind and make decisions that are justified for the goal you are trying to achieve.

USE YOUR PROFESSIONAL NETWORK

Staying abreast of the changing technology landscape and how other executives are managing change is critical for a CTO. “You need to go out there and meet people at conferences such as Oracle OpenWorld Europe,” says Weir. As a long-standing Oracle ACE Director and now an Oracle Groundbreaker Ambassador, he appreciates the opportunities that having a close relationship with a software vendor such as Oracle brings and the chances to interact with peer Oracle ACEs and Groundbreakers.

“Groundbreakers is a brilliant network for learning and collaborating,” Weir says. He also reads magazines online and is a Slack and Twitter enthusiast—but like most of us, he admits to struggling to shut off those information sources at times.

DON’T EXHAUST YOURSELF

Indeed, for people such as Weir, knowing when he is able to concentrate and when to back off is key. “Mornings are best. I’m more creative, I’m rested, and I don’t have 100 things in my head,” Weir says. That focus has a limit, however: “Sometimes at the end of the day, my wife complains that I don’t want to talk. I find it difficult to think, because I’m mentally exhausted. I couldn’t do something creative at night.”

His favorite way to find flow is to isolate himself from office distractions. “If, for example, I have to deliver a hardcore demo, like a sample pattern to show my developers how to implement some-

thing the right way, I stay home, isolate myself in my home studio, shut off email, and just focus," he says.

When you put your head down, time flies, Weir finds—but that, too, has risks. He's found it critical to take breaks for meals and "to disconnect a little and come back fresh. Your creativity can be affected if you don't look after yourself a little. That's a common problem. You can exhaust yourself without knowing it."

Alexandra Weber Morales is Oracle director of developer content.

PHOTOGRAPHY BY
BRINGINTOBEING/MATT LEVER

NEXT STEPS

FOLLOW the Oracle Developers Blog.

LEARN about the Oracle Groundbreaker Ambassadors program.

Hidden Opportunities in Autonomous Databases

A longtime database architect says there's always opportunity in times of change. Right now, the time is ripe for those who can help AI do its job.

By **Jeff Erickson**

If you've been in the database game as long as Kerry Osborne, you've gotten used to upheavals due to leaps in the technology. Osborne has built complex and ambitious database architectures for banks and telecoms in the midst of tech trends that have swung "from integrated solutions to best-of-breed and back again." Meanwhile, says the Oracle ACE and Oak Table member, his database architectures have had to adjust to the rise of the internet and open source software; exponential data growth; and, now, the growth of cloud-based infrastructure and databases.

At every turn, there have been new opportunities. "I just had to be willing to change my focus a little," he says. When Oracle launched Oracle Exadata Database Machine, it effectively undercut his business—building complex multinode Oracle Real Application Clusters solutions—with a preintegrated package that "did what we did, but better and cheaper." In response, Osborne had to move his work up the value chain a little and stop worrying about "the low-level integrated components, getting the network setup correctly, and all that stuff."

The arrival of Oracle Autonomous Database, which uses artificial intelligence in Oracle's second-generation cloud infrastructure and in Oracle

Database 18c to deploy, tune, patch, and secure the database with no human intervention, is causing a similar disruption for many of the DBAs Osborne talks to. "I've had lots of people asking me, 'What's going on with what I'm doing as a DBA?'" His advice? Roll with it.

"If you're building databases, if you're patching databases, or if you're adding storage to databases, those jobs over time are going to go away," advises Osborne. But not right away, he says, because there will be a long tail of traditional DBA work as on-premises databases continue to play a role for some companies. For younger DBAs who are doing this kind of work, however, he says "those infrastructure jobs will dwindle."

But not all traditional DBA work is going away. "If you're doing performance optimization, architecture, and design work, and if you're solving hard problems, those skills, as well as communication skills, are things where humans have the advantage over AI," Osborne observes.

One opportunity he sees emerging is to help AI do its job better—a tactic he learned when Oracle first came out with the cost-based optimizer, he says. The optimizer uses query history and other information to choose the best way to retrieve the data. "It was a way to let the database decide for itself how to do some of the things that database administrators were already doing," he says.

"Skills then shifted to focus on how the cost-based optimizer works. The folks who really had a good understanding of the way the optimizer did its calculations and better understood why it was doing what it was doing could build the systems or tune systems to help the optimizer do the right things," he says. "Over time the optimizer got better and better, and DBAs had to do that less than less."

Osborne sees the same dynamic at play with the newer machine learning capabilities in Oracle Autonomous Database. In your database design and modeling work, "Be the one to under-

stand how the current optimizer and other, newer machine learning processes work," he says—whether the AI is identifying and remediating threats to the data, patching databases in the background, or predicting user behavior. Figure out how to "get the most out of those processes now and adjust as they get better and better at what they do," he advises.

Do that, and "If you're not doing public speaking, go sign up for a Toastmasters class or something where you can start practicing your public speaking. It doesn't help to have the best idea in the world if you can't communicate," Osborne concludes. Plus, "that's a human thing that a computer can't ever do."

Jeff Erickson is editor at large for Oracle Content Central.

NEXT STEPS

LEARN more about Oracle Autonomous Database.

TRY Oracle Autonomous Database.

BY BLAIR CAMPBELL

Here's Your Help

Atypical solutions for forest maintenance, father/son bonding, and master data management

Alex Nuijten
Oosterhout, The Netherlands

ORACLE ACE Director

Company/URL: allAPEX

Job title: Principal

Length of time using Oracle products: 19 years

Which new features in Oracle Database are you currently finding most valuable? I'm mostly interested in features that make my life as a database developer easier. With every release of the database, there are things added that make SQL and PL/SQL more elegant than they already are. One particular example: the qualified expressions that were added in Oracle Database 18c. They make the code that you write a lot easier to read and interpret. Another thing are the polymorphic table functions, although I'm still studying the details of this functionality.

What's your favorite tool on the job? Oracle Application Express [Oracle APEX] by far. It makes it easy to create beautiful applications that leverage everything Oracle Database has to offer. The community that embraces Oracle APEX is simply awesome—very enthusiastic about the tool and always willing to help out when needed.

What's your favorite thing to do that doesn't involve work? Barbecuing, smoking, grilling, and working in the small forest near my home—where I cut down trees with my chainsaw, split wood, and then can savor the joys of having a nice fire when it gets cold outside. Although I must admit that I created a small Oracle APEX application to help with the maintenance of it all.

Brian Bream

Minneapolis/St. Paul, Minnesota

ORACLE ACE Director

Company/URL: Collier IT

Job title: CTO

Oracle credentials: Oracle Real Application Clusters 12c Certified Implementation Specialist and Oracle Cloud Infrastructure Classic 2018 Certified Associate Architect

Length of time using Oracle products: More than 20 years

How did you get started in IT? My father was a mainframe programmer working for the Department of Defense on space projects, and his idea of father/son time was my changing magnetic tapes and loading keypunch cards. It rubbed off, and eventually I took a position in the Department of Defense working on transistor-level maintenance and repair of control systems that operated ship propulsion and electrical generation equipment.

What's the next big thing driving change in your industry? Movement away from platforms and a focus on the data itself. For years companies have failed to monetize data. It was important to have the infrastructure to handle the data, but not much was being done with it. This meant missed opportunities for increasing revenue that were hidden in the datasets. With the renewed focus on data, technologies such as big data, machine learning, artificial intelligence, predictive analytics, and the Internet of Things are improving the lives of humans globally.

What advice do you have about how to get into application development? Be a technologist. It's fine to focus on a particular vertical, but it's imperative that the supporting solution sets are understood. Understanding the entire ecostructure of a solution allows for better decision-making, with fewer mistakes, in less time. Consider entry-level positions with a vendor or one of its partners. It's a fantastic way to grow rapidly, be exposed to cutting-edge technology, and network.

Gary Crisci
Norwalk, Connecticut

ORACLE ACE Director

Company/URL: General Electric
Job title: Principal architect
Length of time using Oracle products: 16 years

Which new features in an Oracle application are you currently finding most valuable? Zero-footprint Oracle Essbase is something I'm excited about. It appears to be a perfect blend of relational database technology and OLAP. It can provide users the Oracle Essbase experience they love, but it also has the depth and breadth of a relational database. It will be an exciting leap forward for Oracle Essbase when users can query data from an Oracle Essbase cube as soon as the record is created in a transactional system.

Name one unique use of applications at your company. We're currently exploring how to use EPM [enterprise performance management] and BI [business intelligence] tools such as Oracle Data Visualization to analyze master data. Typical use cases for data visualization focus on numeric measures and key performance indicators. We have a very extensive enterprise chart of accounts with thousands of members across various dimensional segments. New advances and ease of use with visualization tools such as Oracle Analytics Cloud are providing ways to interpret relationships between master data elements that were not easily seen before.

What technology has most changed your life? Oracle Essbase, by far. I would not have the career I have today had I never begun working with this interesting and powerful tool. The second tool I have to credit is internet user forums. Early on in my technical career, I was learning on the fly. Now, having an easily accessible network of experts who can help me figure out how to resolve an issue is priceless. And, over time I've become one of those experts giving others advice and solutions.

COLLABORATE19

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Register Today!

attendcollaborate.com

APRIL 7–11, 2019

SAN ANTONIO

Database Innovation

New features in Oracle Database 19c: A few favorites explained

BY JEFF ERICKSON

Oracle Database 19c has arrived, with new features that make Oracle Database even more useful for the daily work of running a digital business. Released on Oracle Live SQL in January 2019, Oracle Database 19c delivers long-term stability and an impressive set of innovations. There's something here for developers, DBAs, data analysts, and security experts alike.

Oracle Magazine asked Dominic Giles, a master product manager for Oracle Database at Oracle, to share a few of his top features in Oracle Database 19c.

Automatic Indexing

The Automatic Indexing feature uses machine learning algorithms to create and constantly adjust indexes to improve performance and cost savings. "For the first time ever, a database can determine for itself the optimal set of indexes for your dataset," Giles says. That means "you can start the database without any, or with very few, indexes, and over a short period of time, the database will look at the way the data is queried and build indexes to provide efficient access plans for that dataset."

The feature works equally well with an existing database, where Automatic Indexing can help fine-tune the collection of indexes in the database. The database can implement indexes and then continually validate them. It can also remove unnecessary indexes, Giles says. This is important, because over time indexes pile up, "often for reports or batch jobs that are no longer needed," he says. That can be expensive, because those indexes can increase the compute and I/O resources needed for the database, adds Giles, who notes that large commercial applications running on top of Oracle Database can build up thousands of indexes over years of use.

Active Data Guard DML Redirect

This new capability in Oracle Database 19c helps you get more value from a standby database—which is a mirror copy of a production database meant for disaster recovery. "That's an expensive piece of infrastructure for our enterprise customers," he says. "It's just sitting there consuming space and electricity."

In Oracle Database 11g, Oracle launched Oracle Active Data Guard to help customers get more use from that standby database by running reports and backups against it.

In Oracle Database 19c, Oracle adds an important twist on this feature called Active Data Guard DML Redirect, which enables you to do transactions against the standby database. One reason for this, says Giles, is that "a lot of reporting applications don't just report or flag or retrieve information from the database; they also write lightweight transactions. With Active Data Guard DML Redirect, those transactions are immediately and transparently redirected back to the primary database, and once committed on the primary, they are made available on the standby. From the user's perspective, it's as if they were writing to a standard Oracle Database."

> **“Stability is a core aim for Oracle Database 19c—it’s a long-term-support release.”**
>
> —**Dominic Giles,** Master Product Manager, Oracle

Active Data Guard DML Redirect works equally well, he says, whether the backup is running on premises or in the cloud or whether both primary and standby databases are running in the cloud. “This will give customers more flexibility to creatively use that additional asset,” says Giles.

Hybrid Partitioned Tables

Oracle knows that its customers are often caught between exploding data volumes and regulations that require them to keep data on hand for years, says Giles. Hybrid Partitioned Tables in Oracle Database 19c will help address this situation.

Hybrid Partitioned Tables enables database administrators to manage a table between partitions inside the database and partitions held on low-cost read-only datastores outside the database.

“This means you can use all of the core analytics features of Oracle Database, even when you place data in low-cost read-only storage,” says Giles.

These datastores can reside on premises or in the cloud.

“The beauty of this model is that a table running at the customer’s site could effectively be stretched to the cloud,” he says.

And read-only data outside of Oracle Database doesn’t require regular database backups but is still accessible from Oracle Database. “It’s a very attractive solution for data lifecycle management, and the customers I’ve spoken to are very excited about this technology,” says Giles.

JSON Support

JSON support in Oracle Database began back in Oracle Database 12c, with native JSON document storage and SQL access, and continued in 18c, with high-performance analytics on JSON

> **“We’ve improved and simplified the syntax for our JSON functions and introduced the capability to do a partial JSON update.”**
>
> —**Dominic Giles,** Master Product Manager, Oracle

documents—just as if the JSON data had been ingested into database table rows and columns, says Giles. “That’s a very fast means of doing analytics on JSON documents,” he adds.

In Oracle Database 19c, Oracle improved its support for JSON—making things even easier for traditional developers.

“We’ve improved and simplified the syntax for our JSON functions and introduced the capability to do a partial JSON update, so you can go in and update one attribute of a large JSON document, instead of updating the whole thing,” Giles says.

In addition, Oracle Database 19c includes new Simple Oracle Document Access (SODA) APIs for Java, Python, C, and Node.js.

“You can work with a broad range of lightweight NoSQL APIs that get rows from JSON documents in the database,” says Giles.

Query Quarantine

The overall performance of a data mart or a data warehouse can suffer when a user runs a query that consumes an excessive amount of I/O and compute resources, explains Giles. Oracle Database 19c “can automatically quarantine those queries and ensure that they don’t run again.” This, he says, results in consistent performance for all database users.

Stability and Availability

New features are important in every Oracle Database release. Stability for on-premises database installations and applications is important too, and Oracle Database 19c has that as well.

“Stability is a core aim for Oracle Database 19c—it’s a long-term-support release,” says Giles. “Our on-premises customers go through lengthy upgrade cycles, and this release,

Oracle Database 19c, is the one that a lot of our customers have been waiting to upgrade to from Oracle Database 11g or Oracle Database 12c."

For more information on the features described here and other new features in Oracle Database 19c, check out the full new features list in the Oracle Database documentation. And for more from Dominic Giles on more features, check out his Oracle Database 19c features post.

You can try Oracle Database 19c today at livesql.oracle.com. On-premises database customers can get more information on the availability of Oracle Database 19c for their platforms at My Oracle Support, Document ID 742060.1.

Jeff Erickson is editor at large for Oracle Content Central.

ILLUSTRATION BY **PEDRO MURTEIRA**

NEXT STEPS

LEARN more about Oracle Database 19c new features.

TRY Oracle Database 19c.

READ "Oracle Database 19c Introduction and Overview."

Three Keys to Cloud Security

"Oracle and KPMG Cloud Threat Report 2019" demonstrates the importance of visibility, shared responsibility, and a CISO seat at the table.

BY ALAN ZEICHICK

Want to protect your assets in the cloud? You need to know what those assets are and who is using them. Your security teams must be able to see everything going on in the cloud infrastructure, from the cloud's core to its edge. They need to be certain about which parts of your cloud applications are the business's responsibility to secure—and which fall under the domain of the cloud service provider. And at the C level, the chief information security officer (CISO) must have a seat at the table during each and every

discussion that involves acquiring or using new cloud applications or resources, in order to make sure those services are safe and compliant with enterprise policies.

Those are three of the top takeaways from the "Oracle and KPMG Cloud Threat Report 2019." Attention to cloud security is essential for modern-day enterprises—as a glance at any newspaper instantly communicates, with headlines reporting downloads of unsecured customer files from retailers, theft of intellectual property from tech firms, and complete business disruption.

Cloud security is a big challenge for another reason: Enterprise use of the cloud has reached surprising levels of adoption and is continuing to increase. In the Oracle/KPMG study, 7 out of 10 organizations reported an increase in the use of business-critical cloud services—and there's a huge increase in the number of enterprises storing their data in the cloud.

At the same time that cloud usage is accelerating, security considerations are being left behind.

Fully 93% of the participating organizations reported that users have adopted rogue cloud applications. That's a prime example of "shadow IT"—that is, technology decisions being made by employees without the knowledge or approval of the IT department. These decisions are rooted in the BYOD movement and the consumerization of IT.

Individual employees, for example, may be running consumer-grade cloud services (think Evernote or Dropbox) to improve personal productivity—and, in the process, might store or even share confidential business information such as customer data or financial documents in those services. Departments may be signing up for hosted SaaS applications (such as WordPress or Adobe Creative Suite). Developers could be using popular cloud-based software development code repositories (GitHub, say, or SourceForge). And staffers might be sharing cloud-based collaboration platforms such as Slack or SharePoint with partners, suppliers, or customers.

Are those cloud applications bad? In most cases, the products are fine from a software-quality perspective. But having a solid reputation doesn't clear those specific apps for use in your business without the IT department's knowledge and approval. And even after an application is approved for use, the CISO must

> **Organizations don't know what their employees are doing with cloud services and where their corporate data is being placed.**
>
> —**Greg Jensen,** Senior Director, Oracle

ensure that it is implemented in accordance with your company's security policies; otherwise, the organization is at risk of having critical data lost or stolen or of letting outsiders gain access to confidential internal information and processes. There are too many risks to organizations for leaders to be complacent about security. Here are three key ways to address those threats—and tackle the challenges head-on.

Key #1: See Everything You Need to Protect

Visibility is essential to every aspect of security. Consider the office building: Cameras are watching over exterior doorways, for example, and logging software is recording when employees and vendors badge in to secure work areas.

The same must be true of critical information about network traffic, successful and unsuccessful attempts to log in to the network, and use of enterprise applications. It's not enough to know that the CFO logged in to the accounting system at 1 a.m. It's also important to know the device type, device location, and telemetry involved. The transaction might be completely valid, or it might come from a place halfway around the world when the CFO is actually at home. Or it might come from the CFO's own smartphone, after a click on a link in a phishing email.

Without visibility, AI-based security software can't detect anomalies or piece together patterns of behavior that might indicate fraud or illegal activity. Without visibility, security investigators can't find root causes of unusual situations quickly and accurately.

That's particularly true with cloud services, says Greg Jensen, senior director of cloud security at Oracle and coauthor of the "Oracle and KPMG Cloud Threat Report 2019." "There are so many examples throughout this report about challenges with visibility," he

says. “Organizations don’t know what their employees are doing with cloud services and where their corporate data is being placed. Is it going on Google? Or Amazon? Is it going on Bill and Ted’s excellent cloud service? They don’t have that visibility.”

One way to get more visibility is to implement CASB-compliance technology for the cloud ecosystem, says report coauthor Brian Jensen (no relation), a risk-management consultant at KPMG.

A CASB, or Cloud Access Service Broker, provides visibility into the entire cloud stack while providing security automation for enforcing corporate policies. A full-featured CASB platform provides threat detection, automated incident response, predictive analytics, and security configuration management.

“A CASB shows what employees are doing with cloud-sanctioned and unsanctioned cloud services,” says KPMG’s Jensen.

“The average organization is running in excess of 1,900 applications—including cloud applications. By and large, security professionals need to use a CASB to monitor business-critical cloud transactions” and then enforce policies regarding those apps.

Key #2: Understand the Shared Security Model

In a classic data center application, the enterprise has complete ownership of security: everything from the physical installation to network access, from patching vulnerabilities to checking users’ digital credentials. In a cloud service—any cloud service—security responsibility is shared between the enterprise and the cloud services provider.

Problems occur when the enterprise fails to realize its security responsibilities, says Oracle’s Jensen. This can happen because of shadow IT or because of misunderstandings about the shared security model for cloud services.

For example, take penetration testing, which measures how easy it is to attack a cloud service with known hacking techniques. Many enterprises don’t see that as any part of their responsibility, so they don’t do it. “A lot of businesses believe they aren’t responsible for testing the security of a cloud service,” Oracle’s Jensen says. “The reality is that whether you are using IaaS, PaaS, or SaaS, your business is responsible for doing penetration testing. The business is responsible for ensuring that the cloud cannot be penetrated—either the service or the application itself.”

> **If there are suspicious user activities associated with your portion of the shared responsibility model, you have to be aware of those events, monitor them, and react to them.**
>
> —**Brian Jensen,** Risk-Management Consultant, KPMG

KPMG's Jensen points to user authentication as an area of common misunderstanding. "While SaaS providers include a single-sign-on authentication solution, passwords simply aren't good enough," he says. "You need balanced user enablement with the requirement to protect sensitive data and transactions, so organizations should consider the use of multifactor authentication with biometrics."

Event monitoring touches both the visibility issue and responsibility sharing, he adds. "Security event monitoring in SaaS is still your responsibility," he says. "If there are suspicious user activities associated with your portion of the shared responsibility model, you have to be aware of those events, monitor them, and react to them." (This shouldn't be confused with the foundational event monitoring that the cloud services provider uses to defend against a variety of network-level events.)

Key #3: Seat the CISO at the Table

A line-of-business department is considering adoption of a cloud-based application—perhaps a turnkey SaaS application. Is the CISO invited to the meetings where that product is discussed, evaluated, and approved? Maybe. But then again, maybe not. And it's quite likely that the CISO's team is not involved in the implementation and integration of that cloud application. In fact, the security team members may not even know about that app until security incidents begin showing up on their dashboard.

"There's a lack of communication, lack of collaboration, and lack of visibility across the C-suite," says Oracle's Jensen. "The C-suite is facing challenges in terms of how to collaborate on security, risk, compliance, and privacy."

Teams won't work together if their managers don't work together. "We have to address these C-suite problems head-on," Jensen says. "We

have to try to make sure that this is a collaborative conversation where everyone understands their unique role in making cloud security successful for the organization. When executives aren't doing their part, the company as a whole is at risk."

A Bright Future for the Cloud

Increasingly, organizations trust the cloud for critical applications and for storing essential data. Security technology is doing a good job of keeping up, but more still needs to be done, as is documented in the "Oracle and KPMG Cloud Threat Report 2019," says Oracle's Jensen.

"The cloud capabilities and the solutions available today are far superior to what we had just a couple of years ago," he says. "There is much more security awareness now than what we had in years past—and more acceptance about the need to have conversations with the security teams and the risk teams."

Alan Zeichick is director of strategic communications for Oracle, where he provides insights and analysis on cloud computing and other advanced technologies. Follow him @zeichick.

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

READ the "Oracle and KPMG Cloud Threat Report 2019."

LEARN more about Oracle CASB.

25,000+ Companies Run Their Business in the Oracle Cloud

More Enterprise Cloud Applications Than Anyone Else

ORACLE®

BY FRANK NIMPHIUS

ORACLE DIGITAL ASSISTANT

Building Dialogs Without a Flow

An introduction to the new composite bag entities in Oracle Digital Assistant

Oracle Digital Assistant is the next generation of the Oracle chatbot platform. It manages and coordinates multiple smaller-scope *skill bots* in a composite chatbot solution that helps users complete multitask conversations.

Dialog flows in Oracle Digital Assistant skill bots define the user/bot conversations needed for completing a specific business task. A dialog flow is like a script whose goal is to understand what a chatbot user wants and collect the information needed to perform a particular task. Like human agents in customer service, chatbots can skip portions of a scripted conversation if the information to be collected has previously been provided by the user.

Before Oracle Digital Assistant, chatbots built with the Oracle Cloud platform required a dialog flow state to be defined for each information entity. For example, to create a chatbot for ordering pizza, you needed to create separate dialog flow

states to ask for the pizza type, to ask for the pizza size, to ask for the type of crust, and to ask for additional toppings.

With Oracle Digital Assistant, you are no longer required to write all of these separate dialog flow states. Instead, using the new composite bag entity feature, you can group related information into a composite data object. The composite data objects are then automagically resolved at runtime.

This article provides an overview of composite bag entities. Following the hands-on instructions, you will build a composite bag entity for a pizza ordering bot that you then use to reduce the number of dialog flow states in the provided sample bot.

ABOUT COMPOSITE BAG ENTITIES

In general, you use entities to extract important information from user messages through natural-language processing (NLP). Entities in Oracle Digital Assistant can be of various types, including value list, entity list, regular expression, entities derived from other entities, and composite bag.

Entities are associated with context variables in the bot dialog flow. If the value of an entity gets extracted from the user message through NLP, then the UI component that references the context variable associated with the entity is not rendered, meaning that users will *not* be prompted for information they have already provided.

Composite bag entities in Oracle Digital Assistant enable you to define entities and variables of type `string`, `location`, and `attachment` as attributes—bag items—of a single object. When a composite bag entity is resolved at runtime, a prompt gets generated for each entity type attribute that did *not* get its value from the user message through entity extraction in NLP.

In this article's pizza ordering bot sample, the composite bag entity describes the pizza the user attempts to order. The pizza composite bag entity object contains

attributes referencing the **PizzaType**, **PizzaSize**, **PizzaCrust**, and **CheeseType** entities. In addition, a pizza object may have a customer name attribute associated with it, so the waiter can identify the customer.

At runtime, composite bag entities are referenced through a context variable that has the custom composite bag entity defined as its type. To resolve a context variable of a composite bag entity type in a dialog flow, you create a dialog flow state that uses the `System.ResolveEntities` or the `System.CommonResponse` component for rendering the bot response. Both components are able to render user prompts as input text and lists.

The difference between the two system components, when used with composite bag entities, is that the `System.CommonResponse` component enables you to customize the rendered bot UI.

Composite bag entities, however, can do more than just group entities and variables into a single object. For each item in a composite bag entity, you can define validation rules, prompts, support for multiple values, fuzzy matching, and other features. For entity references to value list entities, you can specify or inherit a range size value that determines when pagination controls should automatically be displayed.

Both the `System.ResolveEntities` and the `System.CommonResponse` components enable you to visit other dialog flow states.

GETTING READY

In Oracle Digital Assistant, child bots are referred to as skills, or skill bots. In the hands-on instructions for this article, you are working with a skill bot only. Skill bots in Oracle Digital Assistant can be tested standalone with the embedded tester.

A starter skill bot is provided, so you can focus on building and implementing the composite bag entity, following the instructions in this article.

These are the prerequisites for following along with the hands-on steps in this article:

- You need a trial or paid instance of Oracle Digital Assistant. You can sign up for a free trial at cloud.oracle.com.
- You need to download and extract the resources for this hands-on exercise to your computer.

Follow these initial hands-on steps to start the service and import, train, and test the bot.

1. Start Oracle Digital Assistant in a browser, by typing `https://<your cloud URL>/botsui/` into the **URL** field.
2. Click the hamburger icon (☰) at the upper left.
3. Choose **Development** and then **Skills**.
4. Close the menu, by clicking the hamburger icon (☰).
5. Click the **Import Skill** button at the upper right.
6. Navigate to the downloaded and extracted resources for this article and then to the `starter` folder.
7. Select the `OracleMagazinePizza(1.0).zip` starter bot file, and click **Open**.
8. Click the **OracleMagazinePizza** tile in the **Bot** dashboard to open the bot.

Note: If you don't see the imported bot because other bots fill your screen, type `OracleMagazine` into the **Filter** field above the **+ New Bot** tile.

9. Train the bot, by clicking the Train link in the upper right.
10. In the opened dialog box, accept the default settings and click **Submit**.
11. Run the embedded bot tester, by clicking the test icon (▶) in the left menu.

Note: In Oracle Digital Assistant, the tester opens in full window view. To switch back to the bot, click the **Close** button.

12. Type `I like to order a pizza` into the **Message** field, and press the Enter key.
13. Select values from the menus for pizza crust type and pizza size.

14. When asked for the pizza type, type `supreme` and press the Enter key.
15. Click the **Reset** link at the top of the tester window.
16. Click the **Close** button in the top right of the tester window.

What you just did: You just explored the functionality of the simple pizza order skill. Note that for this article and its hands-on instructions, the order in which the bot responses are displayed and how they are getting displayed are relatively unimportant. In a real bot implementation, user experience matters and the UI should be more pleasing to the eye. For this article, it is important for you to understand how this little bot conversation is built, which is what you will do next.

UNDERSTANDING THE DIALOG FLOW

All bot conversations are defined as dialog flows in the dialog flow builder. Take a look at some dialog flow content to see how it is used in a bot conversation.

17. Open the dialog flow builder, by clicking the dialog flow builder icon ( ) in the left menu.
18. In the dialog flow builder, explore lines 16 to 52, which define the conversation you just tested.
19. Note that each step in the conversation is defined as a state in the dialog flow builder (lines 15, 25, 35, and 45).

Note: All of these conversation states are required to collect the information necessary for ordering a pizza. The variables that are populated within the states are of an entity type and are defined in lines 9–11.

Line 12 defines the `iResult` variable, which is of type `nlpresult`. This variable is used to extract information from the user message.

20. Again, run the embedded bot tester by clicking the test icon ( ) in the left menu.

21. Type `I like to order a thick pizza salami` into the **Message** field, and press the Enter key.
22. In the displayed menu, select a size for the pizza.
23. Note that the pizza order was completed within a single bot/user interaction.
24. Click the **Reset** link at the top of the tester window.
25. Click the **Close** button at the top right of the tester window.
26. Open the dialog flow builder, by clicking the dialog flow builder icon (☰) in the left menu.
27. Note the use of the `nlpResultVariable` property in lines 31, 41, and 50. The component property checks whether the requested information is contained in the initial user message. If it is, the component will *not* render and navigation will continue to the next state.

CLEANING UP THE DIALOG FLOW

Composite bag entities help reduce the overall number of dialog flow states needed for a task such as ordering pizza. To prepare to use a composite bag entity in the pizza order skill bot, you first need to clean up the current dialog flow.

28. In the dialog flow builder, select and *delete* lines 25–52. Ensure that the `intent` and `done` states are kept as they are.
29. Change the value of the `OrderPizza` transition action (line 21) to `"getOrder"`. **Figure 1** shows the `intent` state with this change.

What you just did: You prepared the dialog flow for adding a composite bag entity variable state. The `OrderPizza` transition action is followed when the NLP engine understands the user message as a pizza order. The value of this action needed to change, because you removed the `getCrust` state. You will create the `getOrder` state later in the hands-on instructions.

Figure 1: OrderPizza action transition value changed to "getOrder"

```
states:
  intent:
    component: "System.Intent"
    properties:
      variable: "iResult"
    transitions:
      actions:
        OrderPizza: "getOrder"
        CancelPizza: "cancelorder"
        unresolvedIntent: "unresolved"
```

BUILDING THE PIZZAORDER COMPOSITE BAG ENTITY

Composite bag entities are composed of existing entities as well as variables of type `string`, `attachment`, and `location`.

The starter skill bot comes with the following entities defined: **CheeseType**, **PizzaCrust**, **PizzaSize**, and **PizzaType**. Following the instructions in this section you are going to create a new composite bag entity with the name **PizzaOrder**.

30. Navigate to the entity editor, by selecting the entities icon ( ) in the left menu.

31. Click the **PizzaType** entity. **PizzaType** is a value list entity that holds a set of pizza names and synonyms that are valid to order. (It's not covered in these hands-on instructions, but a list of values can be dynamically populated from a remote service.)

Note: The only other property of interest for this article is the `Prompts` property. Here you can define one or more user prompts that get randomly displayed for an entity if the entity is resolved by the `System.ResolveEntities` or `System.CommonResponse` component. If an entity is referenced in a composite bag entity, then the defined prompts are used if they are not overridden in the composite bag entity.

32. Click the **+ Entity** button.
33. Type `PizzaOrder` in the **Name** field.
34. In the **Configuration** section, from the **Type** menu, select **Composite Bag**.
35. Click the **+ Bag Item** button.
36. Type `PizzaType` in the **Name** field.

Note: The **Name** field value doesn't need to be set to the name of the entity it references. You can define any name you like for a bag item. The name of the bag item will be referenced in the dialog flow to read the value saved for an entity.

37. From the **Type** menu, select **Entity**.

Note: At this time, a dialog box appears with the following message: "Entity CheeseType has been set to the bag item PizzaType. Do you want to copy over its properties also here? Any edits previously made to this bag item will be lost. Are you sure you want to continue?" Ignore this message, and close the dialog box, by clicking the **X** icon in the header.

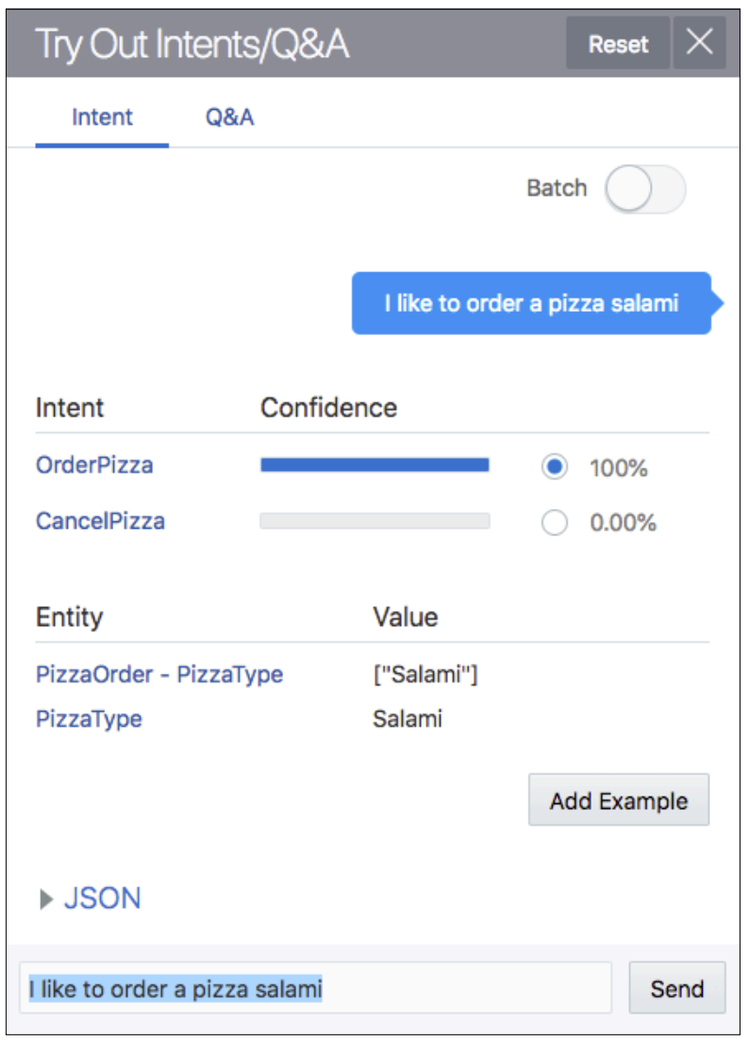
38. From the **Entity Name** menu, select **PizzaType**.
39. Click **Overwrite** in the opened confirmation dialog box.
40. Keep the default settings for all other properties, and click **Close**.
41. Repeat steps 35–40 with the following information to create the corresponding bag items:

Name	Type	Entity name
PizzaSize	Entity	PizzaSize
PizzaCrust	Entity	PizzaCrust

42. Next associate the **PizzaOrder** entity with the **OrderPizza** intent. To do this, open the intent editor, by clicking the intents icon ( ) in the left menu.

43. Select the **OrderPizza** intent.
44. Click the **+ Entity** button at the top right.
45. In the **New Entity** list, enter `PizzaOrder` to search for the **PizzaOrder** entity, and select it.
46. Train the bot, by clicking the Train link at the top right.
47. After training the skill bot, click the Try It Out! icon (on the same line as the **Description** label) to test the composite bag entity extraction in the embedded intent tester.
48. In the intent tester, type `I like to order a pizza salami` into the **Message** field and click **Send**.
49. You should see the **PizzaOrder** entity getting resolved from the user message, as shown in **Figure 2**.
50. Click the **X** icon in the upper right to close the intent tester.

Figure 2: Intent tester with PizzaOrder entity resolved from the user message



What you just did: In this part of the hands-on instructions, you created a composite bag entity to hold the pizza order information. The order in which entities are added as bag items to a composite bag entity is the default order in which the entities are resolved at runtime. (You can also change the order by changing the configuration.) Finally, you associated the entity with the **OrderPizza** intent and tested the entity extraction through NLP in the embedded tester.

Next, add the use of the composite bag entity to the dialog flow.

REFERENCING THE COMPOSITE BAG ENTITY IN THE DIALOG FLOW

To use the **PizzaOrder** composite bag entity in the pizza order skill bot, you need to associate it first with a context variable. The variable is then referenced from a dialog flow state with the `System.ReferenceEntities`

component. In the end, when testing the bot, the behavior is the same as in the starter skill bot, although with fewer lines of code to write.

51. Open the dialog flow builder, by clicking the dialog flow builder icon (☰) in the left menu.
52. On line 13, create a new context variable with the name `pizzaOrder` of type `"PizzaOrder"`:

```
pizzaOrder: "PizzaOrder"
```

53. Click the **+ Components** button.
54. In the opened dialog box, select the user interface (User Interface) category.
55. Select the **Resolve entities** template (the third template option from the bottom of the list).
56. Switch the **Remove Comments** toggle to on.
57. From the **Insert After** list, select **intent**.
58. Click the **Apply** button.
59. Change the `resolveEntities` state name to `getOrder`.
60. Delete the following properties: `maxPrompts`, `cancelPolicy`, `transitionAfterMatch`, `autoNumberPostbackActions`, `headerText`, `footerText`, `showMoreLabel`, and `translate`.
61. Remove the following elements: `actions`, `match`, and `cancel`.
62. In a line under the `transitions` element, add

```
next: "done"
```

Note: The `next` element must be indented two spaces, but the `transitions` element does not.

63. Edit the remaining `System.ResolveEntities` component properties as follows:

Property	Value
`Variable`	`"pizzaOrder"`
`nlpResultVariable`	`"iResult"`

Note: The `getOrder` state should now look like it does in **Figure 3**.

Figure 3: getOrder state after the changes

```
25   getOrder:
26     component: "System.ResolveEntities"
27     properties:
28       variable: "pizzaOrder"
29       nlpResultVariable: "iResult"
30     transitions:
31       next: "done"
32
```

64. Next, edit the `done` state: Change the value of the `text` property from

```
"Your ${size.value} ${type.value} Pizza is on its way."
```

to (in a single line)

```
"Your ${pizzaOrder.value.PizzaSize}, ${pizzaOrder.value.PizzaCrust}
${pizzaOrder.value.PizzaType} is on its way."
```

65. To ensure that your dialog flow is well formatted, click the **Validate** link in the upper right corner. If an error is found, correct the error and validate again.
66. If the dialog flow was validated successfully, run the embedded bot tester, by clicking the test icon () from the left menu.
67. Type `I like to order a pizza` into the **Message** field, and press the Enter key.
68. You should see a menu listing the pizza types. Select a pizza.
69. In the next menu, select the pizza size.
70. Finally, select a pizza crust type.
71. With the order confirmation displayed, click the **Reset** link at the top of the tester.
72. Next, type `I like to order a thick pizza salami` into the **Message** field to test NLP entity extraction, and press the Enter key. You provided information about the pizza type and the pizza crust in the input message, so the bot asks only for the pizza size.
73. Click the **Reset** link at the top of the tester window.
74. Click the **Close** button at the top right of the tester window.

What you just did: In this section, you created a new context variable for the **PizzaOrder** entity. When testing the bot conversation, you could see that the user is prompted to provide information that was not contained in the initial user message.

ADDING THE CHEESETYPE ENTITY TO THE COMPOSITE BAG

Besides reducing the lines of code that need to be written for a user/bot conversation, composite bag entities make it really easy to add or remove information:

75. Open the entity editor, by clicking the entities icon () in the left menu.
76. Select the **PizzaOrder** entity.
77. Click the **+ Bag Item** button.
78. Type `Cheese` in the **Name** field.

79. From the **Type** field menu, select **Entity**. The **Entity Name** field should automatically be set to **CheeseType**.
80. Click **Overwrite** in the opened dialog box.
81. Select the dialog flow builder icon ( ).
82. Change the value of the `text` property in the `done` state from

```
"Your ${pizzaOrder.value.PizzaSize}, ${pizzaOrder.value.PizzaCrust}
${pizzaOrder.value.PizzaType} is on its way.”
```

to

```
"Your ${pizzaOrder.value.PizzaSize}, ${pizzaOrder.value.PizzaCrust}
${pizzaOrder.value.PizzaType} with ${pizzaOrder.value.Cheese} is on its way."
```

83. Click the **Close** button on the upper right.
84. Click the test icon ( ) in the menu on the left.
85. Type `I like to order a small thick pizza salami` into the **Message** field, and press the Enter key.
86. Because you did not specify a cheese type, this information is now requested in the bot response.
87. Select a cheese type.
88. You should see the changed confirmation message with the type of cheese added.

What you just did: With minimal effort, you added information to the **PizzaOrder** composite bag entity. A change like this requires more work if the order process uses dialog flow states to collect the individual pieces of required information. This certainly is another argument for using composite bag entities.

Note: Before adding any entity as a bag item, you always set the name of the bag item to the name of the entity it references. The bag item name is what is getting referenced when composite bag entity values are being accessed. This is why `${pizzaOrder.value.Cheese}` is used in the confirmation message instead of `${pizzaOrder.value.CheeseType}`.

CONCLUSION

Composite bag entities in Oracle Digital Assistant enable you to group related information in a single object, an object that is comparable to a business domain object. Changes applied to composite bag entities have an immediate impact on the bot/user conversation.

In this article and its hands-on instructions, I introduced the concepts and basic use of composite bag entities. Advanced functionality such as value disambiguation, fuzzy matches, validation, error messages, and multiple entity values are also worth mentioning, although they fall outside the focus of this introductory article.

Frank Nimphius is a master principal product manager in the Oracle Digital Assistant Product Management group.

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

READ "Oracle Digital Assistant Version 18.4.3 Introduces Skill Chatbot Capability."

TRY Oracle Digital Assistant.

DOWNLOAD the bot for this article.

BY JT THOMAS

ORACLE VISUAL BUILDER

The New Agnostic Applications

The future is progressive web apps.

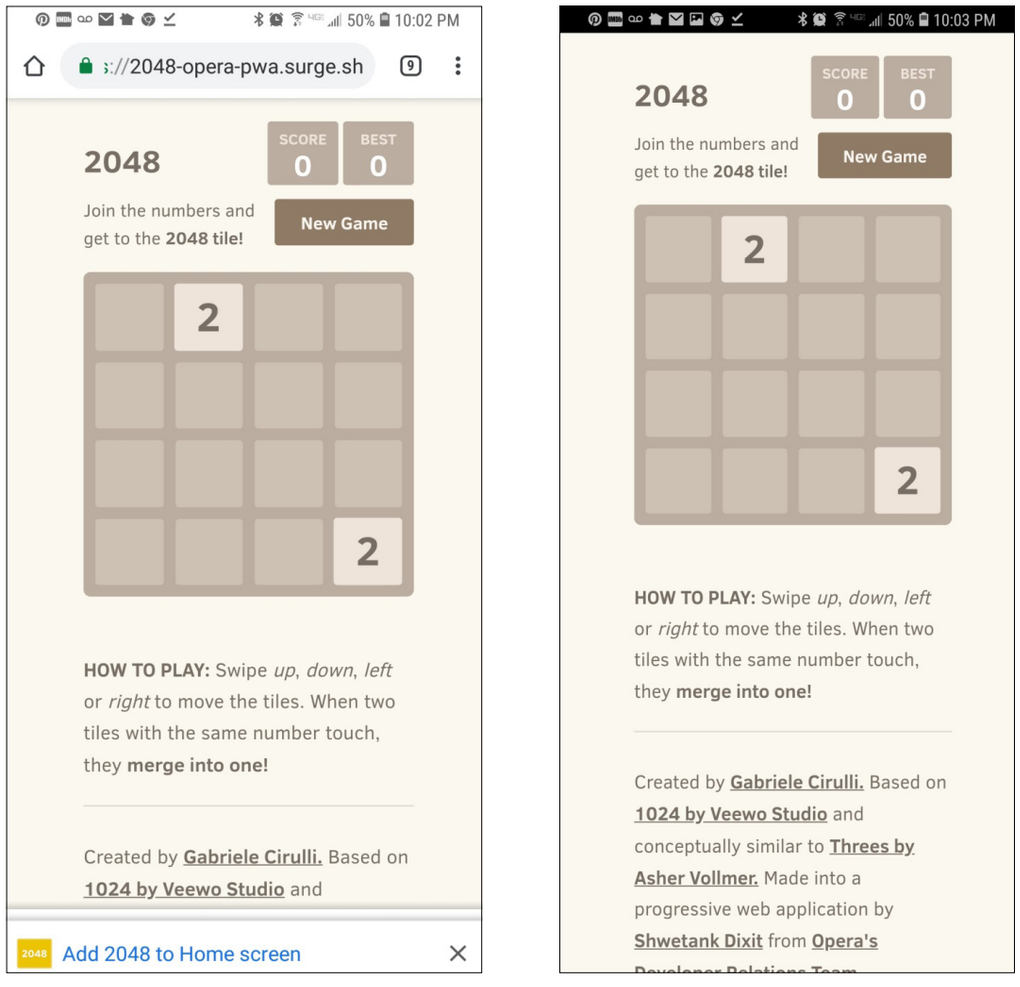
Progressive web apps (PWAs) deliver feature-rich web applications that can take full advantage of native browser *and* mobile device features. Whereas responsive web apps can change their *layout*, depending on the device resolution and orientation, PWAs enable and disable *features*, based on the browser and device running the apps. PWAs give a development team the option of managing a single, common, standards-based codebase not just across device platforms but also across desktop and mobile while still delivering a top-notch experience for each target platform.

This article focuses on the linkable and installable capabilities that enable seamless distribution and installation of a PWA onto a mobile device, using Oracle Visual Builder. You can find a large catalog of web apps at https://pwa.rocks that deliver their content to mobile devices as PWAs. View the apps on your mobile device, where you will also be presented with the option to add the apps to the device home screen. Although a PWA is initially delivered to a device through a browser, you can

add it to your mobile home screen, which will install the app onto the device. See **Figure 1** for an example of this experience.

As you can see in **Figure 1**, the app is initially loaded through a browser via a URL and users are given the option to add the PWA to their home screen. When added to the home screen, the PWA delivers a set of icons, resources, and cached files for the app and the appropriate home screen icon for your device. From this point on, users

Figure 1: A PWA accessed through the browser with the Add to Home Screen option, followed by the PWA loading full-screen from the home screen



can launch the PWA from the home screen and it will look and act just like a native full-screen app—for example, there is no browser chrome, as shown in the address bar in **Figure 1**. Furthermore, most, if not all HTML, JavaScript, and CSS files will be local, so no HTTP fetch is needed at startup, which improves performance as well.

CREATING A PWA WITH ORACLE VISUAL BUILDER

Oracle Visual Builder is a cloud development platform for delivering modern, connected, web and mobile applications, using standard web technologies such as HTML, JavaScript, and CSS. Oracle Visual Builder has been used to build SaaS extensions for Oracle Sales Cloud, Oracle Human Capital Management Cloud, and Oracle Enterprise Resource Planning Cloud; for building updated user interfaces for on-premises systems; and for delivering business- and customer-facing mobile applications to the Apple and Google app stores. Oracle Visual Builder is available as a standalone Oracle Cloud service, but it also ships with Oracle Integration Cloud as well as several SaaS services, including Oracle Engagement Cloud Digital Customer Service.

Within Oracle Visual Builder, you can build responsive web apps or mobile-specific apps. For mobile apps, you can deploy each app as a native package or as a PWA. You can connect those apps to REST services and define server-side business objects that are shared among all apps within a project.

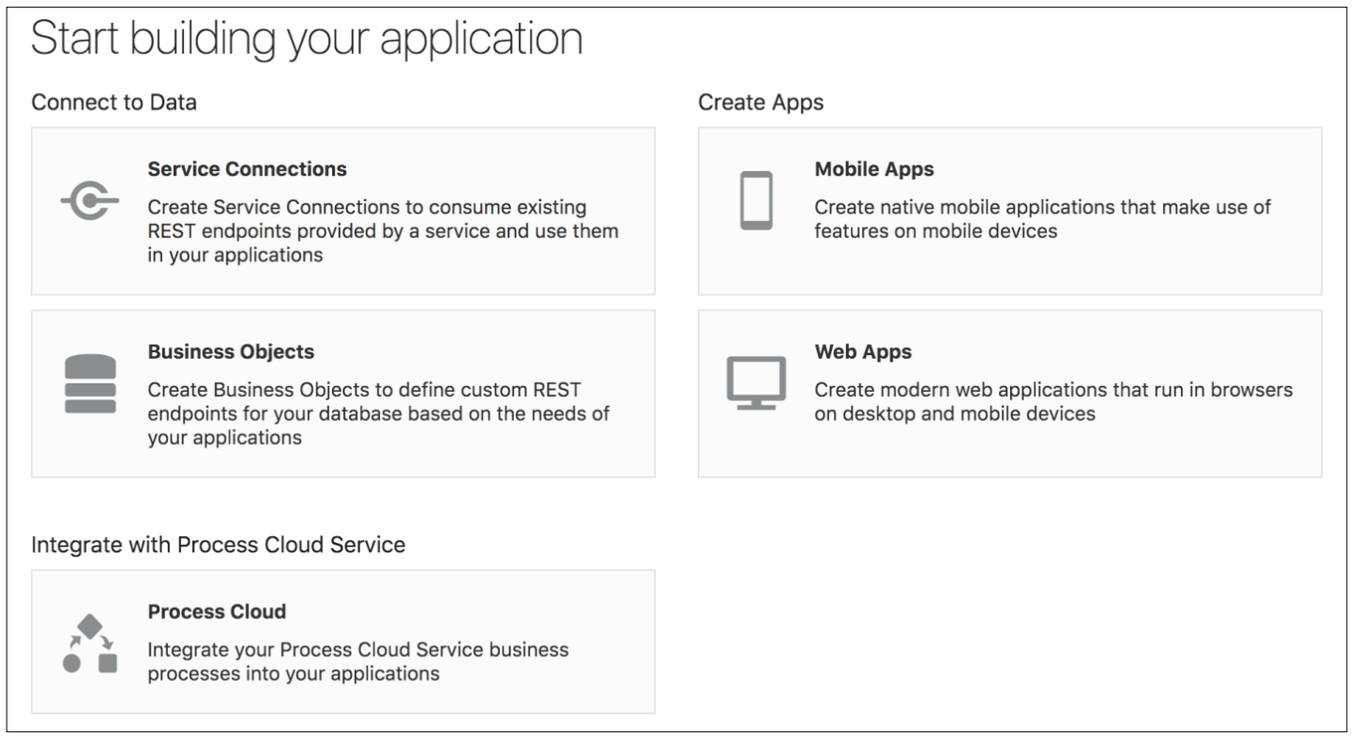
For the purposes of this article, I will create a simple mobile app that connects to a REST service and is deployed as a PWA. To follow the hands-on instructions in this article, you need to have access to Oracle Visual Builder, which is available as a free trial.

Within Oracle Visual Builder, click the **New** button at the top right of the screen. Enter a name for **Application Name** (the application ID will be generated for you),

and click **Finish**. **Figure 2** shows the Welcome screen that greets you after you create your project, and it highlights the primary functions of Oracle Visual Builder.

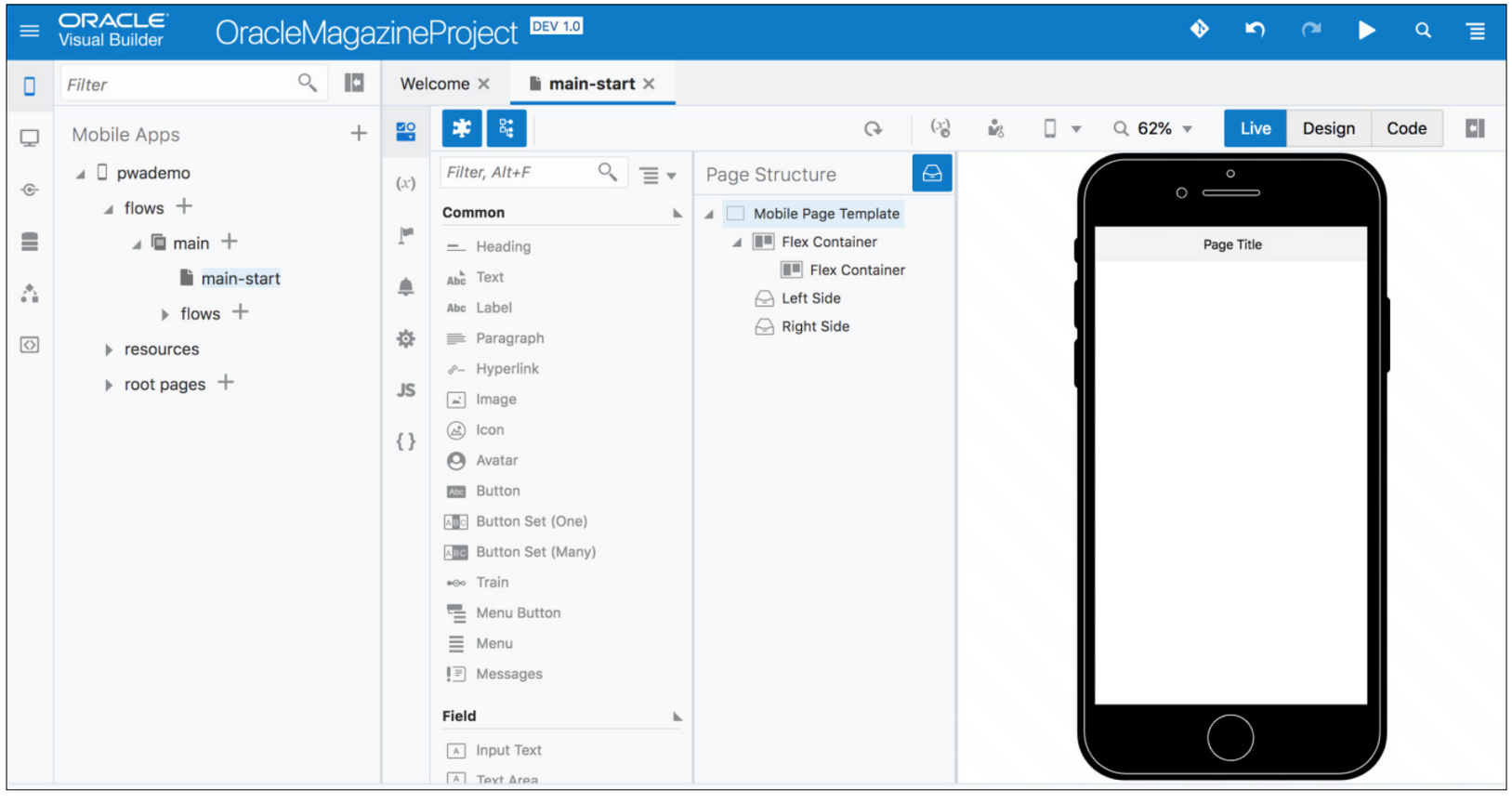
Oracle Visual Builder web and mobile apps both use standard HTML, JavaScript, and CSS for the applications. The apps share the same runtime and access to service connections, business objects, and business process integrations within the same project. A web app is fully responsive from a large wide-screen desktop format down to a mobile portrait-mode-oriented device, whereas a mobile app is typically designed for a similar set of device resolutions and orientations. See the "What's the Difference?" sidebar for details on the differences between Oracle Visual Builder web and mobile apps.

Figure 2: Quickly creating web and mobile applications that integrate any REST service, server-side business objects, and business processes



To continue building the PWA, select **Mobile Apps** from the **Create Apps** section and then click the **+ Mobile Application** button in the left pane. This will take you to a mobile app wizard in which you can provide your mobile app name and select a layout. Enter a name for **Application Name** (mine is `pwademo`), and select **None** for **Navigation**, because this is a simple one-screen demonstration. Click the **>** button at the top right to move to the next screen of the wizard, and select **No Content**. Click **Finish** to create the app and load the designer. **Figure 3** shows the designer screen with (from left to right) the project manager, component palette, page structure, and designer surface. Each of these windows can be collapsed. Farther to the right is a

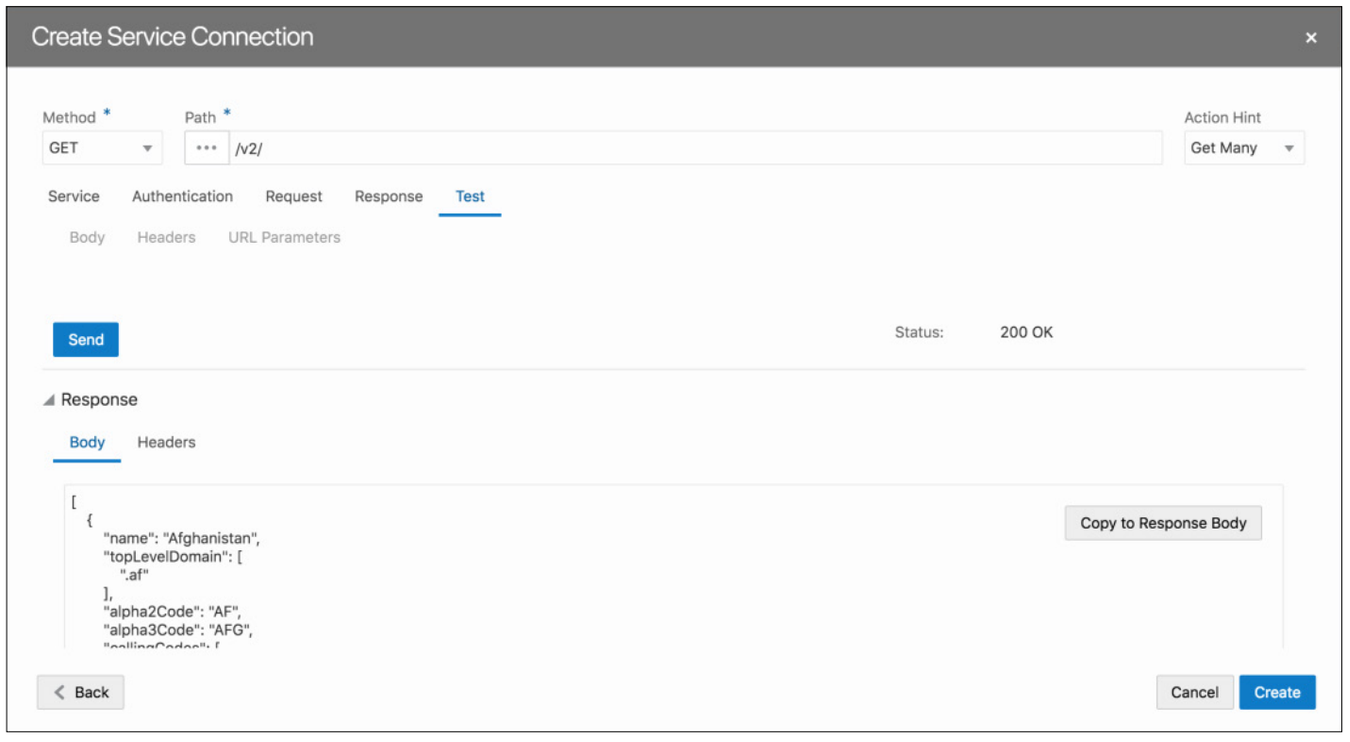
Figure 3: The Oracle Visual Builder design-time experience



property inspector, which you can make visible by selecting the top right "drawer" button next to the **Code** button.

For this project, I want to connect to a REST service for the data. I'll use a public REST service that requires no authentication and returns a list of countries with some data on each of them. To add the REST service, create a service connection by selecting the button on the far left of the designer screen. Click the **+** button to the right of the **Service** text, and select **Define by Endpoint** from the **Create Service Connection** screen. In the next screen, enter `https://restcountries.eu/rest/v2/` into the empty URL text box and change **Action Hint**, if needed, to **Get Many**. With this REST service added, with the default name Countries, you can test the service within Oracle Visual Builder. Click **Test** to test the connection. **Figure 4** shows

Figure 4: Testing the REST endpoint



the service connection, the testing interface, and the test response. Click **Copy to Response Body** in the **Response** section.

Now let's add some UI elements. For this application, I want to show a list of countries, so in your **Component Palette**, select the **List View** component under **Collection** and drag it onto your design surface. You should see a rendering of a list view with three sections showing generic field names. With that **List View** component selected on the screen, open the **Property Inspector** (top right drawer button)

WHAT'S THE DIFFERENCE?

Within Oracle Visual Builder, a mobile app differs from a web app in the following ways:

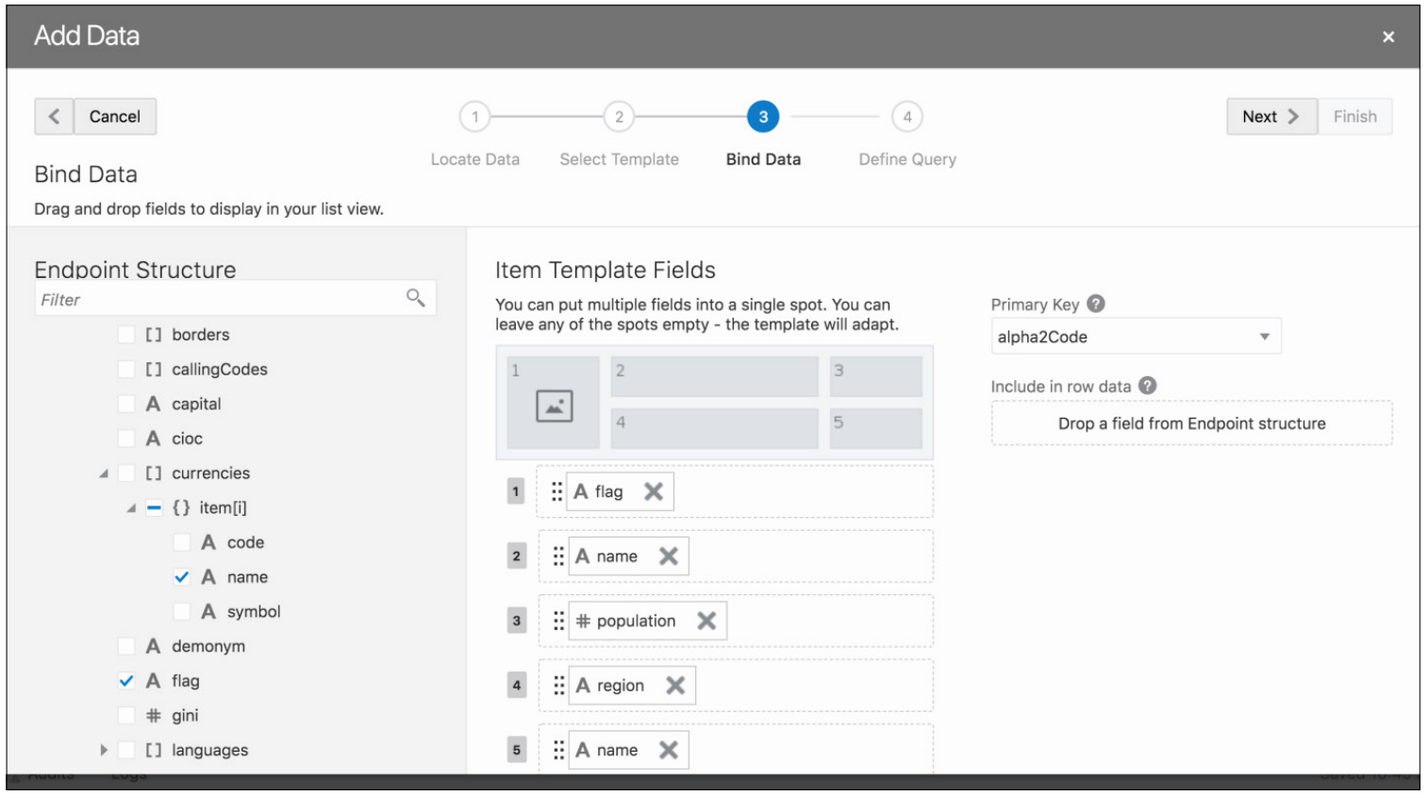
- The mobile app creation wizard includes an optional tabbed navigation view (a typical mobile UX pattern) and can autogenerate flows and pages for each tab.
- The mobile app is designed within a mobile device frame that can be selected for common popular formats (iPhone X/8 and Samsung Android) and can be rotated back and forth between portrait and landscape mode.
- The mobile app can generate a native package for Android and iOS (requires uploading developer certificates and provisioning profiles as needed).
- The mobile app runtime can display UI controls and actions (for example, slide to delete a row) to match the target device (Apple or Android) UX.
- The mobile app can include Cordova plugins for specific device features (requires building a native package for Android and/or iOS).
- A mobile app can be deployed as a progressive web app.

and click **Add Data**. In the **Add Data** wizard, select the **Countries** (or whatever you called your service connection) GET endpoint, and select the default **List Item Template** with the image. In the **Bind Data** screen, first select **flag** for field 1, then **name** for field 2, and then whatever additional fields you would like. **Figure 5** shows an example of selected fields.

TESTING AND DEPLOYMENT

With the mobile app created, you can start testing it. You can run the app live in the Oracle Visual Builder designer window, you can run and test it in the browser in a device preview window, and you can deploy it to a device as a native package and/

Figure 5: Visually binding specific fields from the response to the UI control



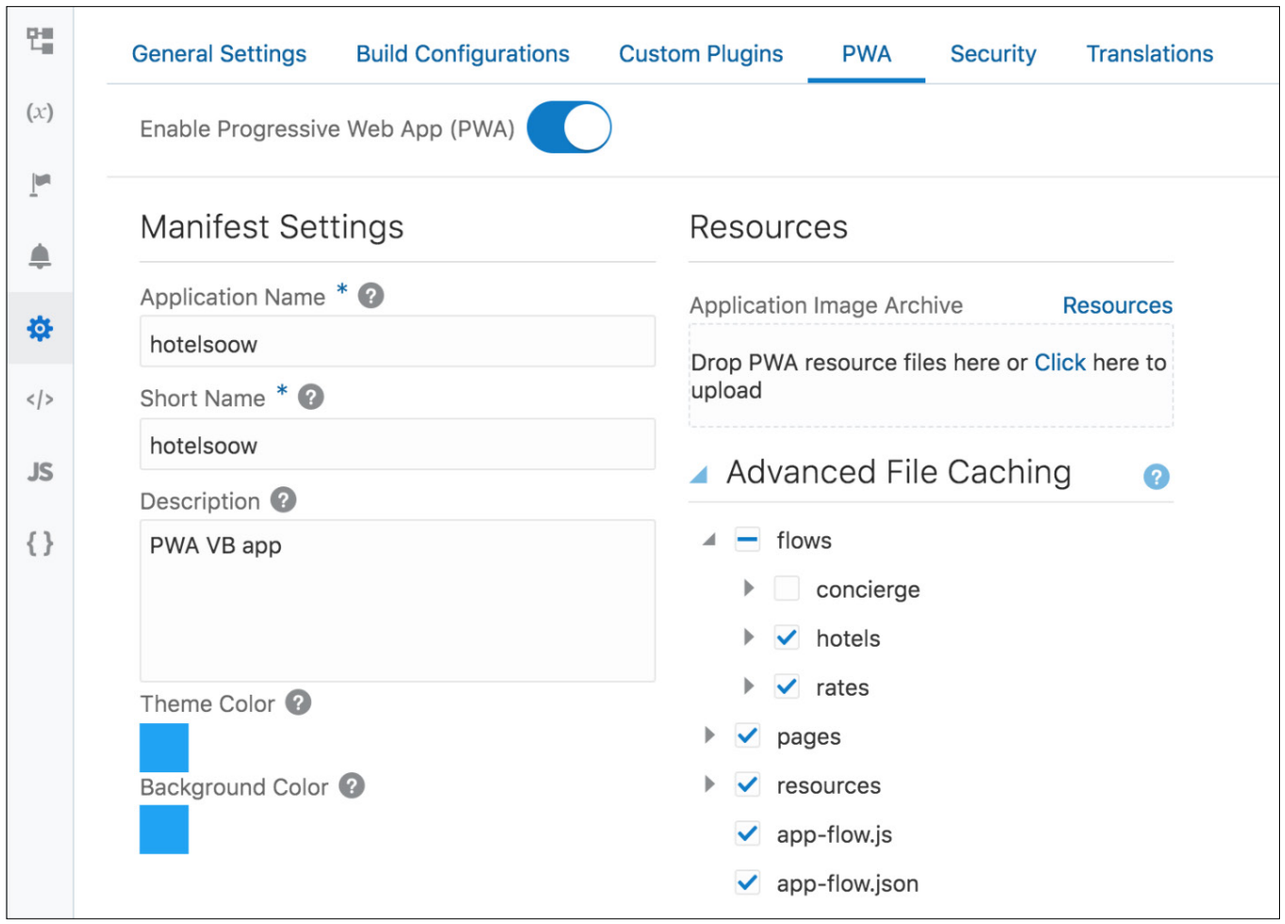
or a PWA for testing and final distribution. To test the app in the browser, select the button toward the top right. A new browser window will open in which you can interact with the application and connect Google or Safari developer tools.

To generate the QR codes for app deployment, you need to set up the build for either a native package or a PWA. For a native package, you must upload Apple developer certificates and a provisioning profile for iOS and/or the Google keystore for Android. Oracle Visual Builder will generate the native package for you each time you stage your application or explicitly rebuild your app. You can also download the package and deliver the IPA (iOS App Store Package) or APK (Android Package) files directly to your end user or through the public app stores or an enterprise app store. To deliver your app to a device as a PWA, select the PWA option in your mobile app settings and configure a few optional settings. **Figure 6** shows PWA settings available within Oracle Visual Builder for PWA deployment.

The PWA settings for deployment offer a few options, including **Application Name**, **Short Name**, and **Description**. The short name is what is displayed on the device home screen. You can also use the PWA settings to upload a set of images for icons and splash screens. The Oracle Visual Builder default resource package includes example sets of icons and splash screens in the different required sizes to support several device types. You can replace those images with your own in that same size and upload them for this article's PWA.

You can also specify which files you want installed on the device when you add it to the home screen. This option helps a PWA feel like a native app. As opposed to a web app, which must fetch all of its files, including the initial HTML page, the PWA caches the initial page and other selected files onto the device. The benefit, of course, is that it loads immediately and that, coupled with the removal of the

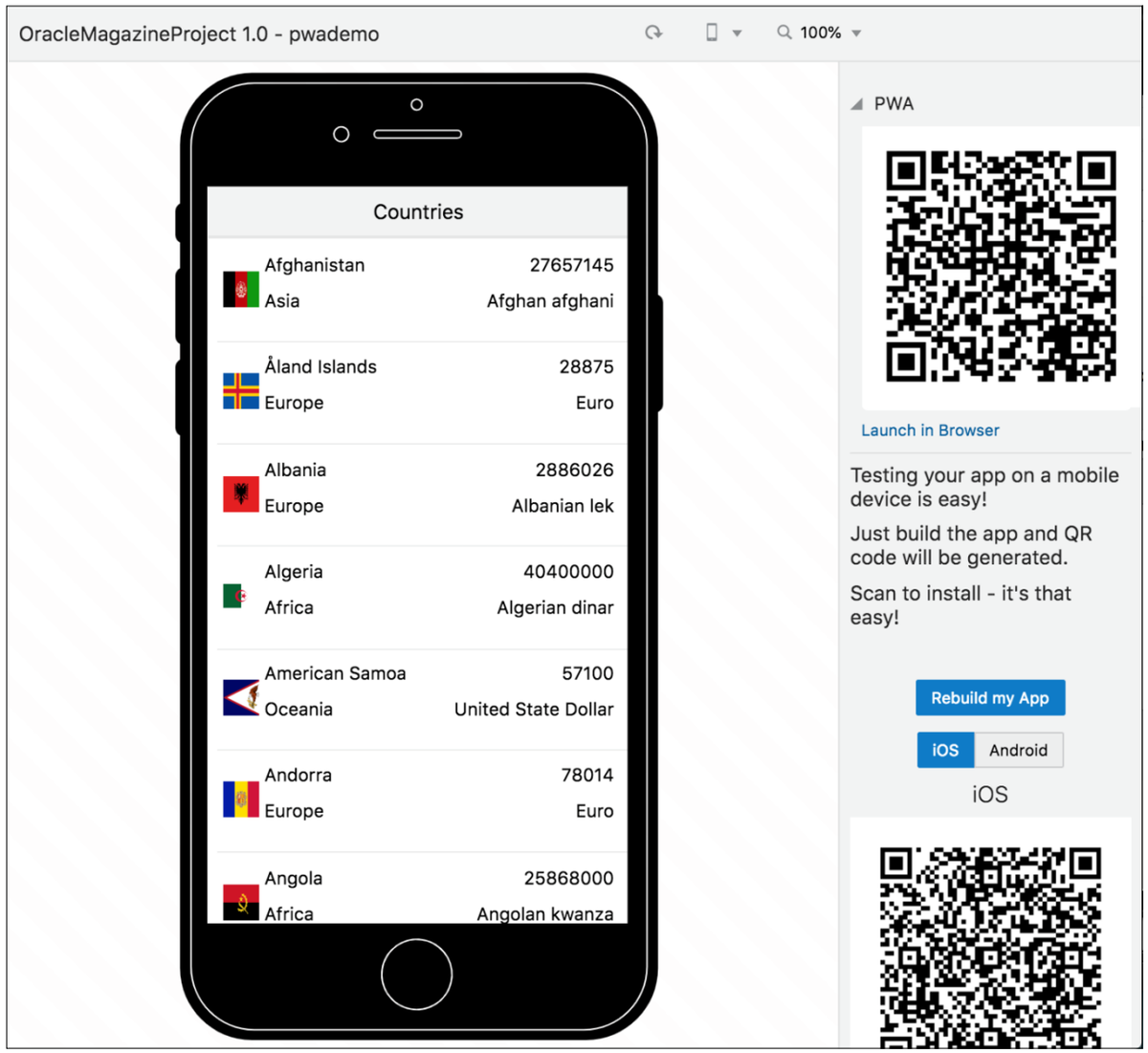
Figure 6: Easily generating a PWA that can be deployed to the device with a QR code or from a URL



browser chrome (including the address bar), the app looks and feels like a natively installed app.

With the PWA settings complete—accepting the defaults is fine—you can deploy the application to a device by scanning the QR code or by downloading and distributing the native package APK or IPA files. **Figure 7** demonstrates the mobile app running in the browser with a QR code displayed to access the PWA and the iOS and Google packages for installation.

Figure 7: Running and testing the app with QR codes for device deployment



CONCLUSION

Thanks to advances in web technologies and standards, PWAs homogenize the various target platforms with a common technology and a progressive feature set that offers the ability to deal with—or take advantage of—some of the differences between them. PWAs remove the stigma often associated with web apps running

on mobile devices, and the momentum of PWAs is clear, with large companies and app innovators alike taking advantage of this new approach.

JT Thomas is a product manager at Oracle but a developer at heart. He works on Oracle PaaS with a focus on the high-productivity JavaScript development environment Oracle Visual Builder Cloud Service.

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about progressive web apps.

TRY Oracle Visual Builder.

World's First "Self-Driving" Database

Oracle Autonomous Database

Protects your Data from Cyber Attacks!
Automatically Updates Security while Running

ORACLE®

ORACLE APPLICATION EXPRESS

Rapid Data Model Development

Quickly create data models for any application with Quick SQL.

Oracle Application Express (Oracle APEX) is the low-code development framework of Oracle Database. Oracle APEX enables the easy creation of modern and responsive web apps with no code. It is ideal for creating new user interfaces on top of existing data models or application logic in the database, but it is also widely used for creating opportunistic applications: brand-new applications that have loosely defined requirements and need to be up and running as soon as possible.

Every application (regardless of technology) is only as good as its data model, and a poor data model will result in a poor application. Quick SQL, a new feature in Oracle APEX Release 18, is the ideal tool for modeling data and creating a physical data model in Oracle Database. This article focuses on how to use the Quick SQL feature embedded in Oracle APEX 18, but even if you're not an Oracle APEX user, you can still use Quick SQL online for free, via Oracle Live SQL, the free site where you can learn, practice, and share examples of SQL and PL/SQL.

Now suppose you've been tasked to quickly build an application to manage the employees of your organization. In this *Oracle Magazine* article, you're going to build a new data model to manage these employees and then build an Oracle APEX application on top of this new data model. Employees have jobs, and they also belong to a department, so you will need to model these entities as well.

This article's sample is built in Oracle APEX 18.1. If you're not already running Oracle APEX 18.1 or later, you can request a free workspace at apex.oracle.com. Alternatively, you can go to the Oracle Technology Network to download the Database App Development Virtual Machine, which includes a preconfigured Oracle Database 18c Enterprise Edition database, Oracle APEX 18.2, Oracle REST Data Services, Oracle SQL Developer, and Oracle SQL Developer Data Modeler.

DESIGNING THE INITIAL TABLE

Begin your exploration of Quick SQL by modeling the `departments` table.

1. In Oracle APEX, click the **SQL Workshop** tab.
2. Click the **Utilities** icon, and then click the **Quick SQL** icon. (At livesql.oracle.com, click **Quick SQL** on the menu.)
3. In the left text area (**Quick SQL Shorthand**), type `departments` and then press the Enter key on your keyboard. This is your table name.
4. Type two spaces, type `name`, and then press Enter.
5. Type two spaces, type `cost center`, and then press Enter.
   The Quick SQL shorthand you entered should result in this:

```
departments
  name
```

```
  cost center
```

By indenting two spaces under `departments`, you are using the shorthand syntax of Quick SQL to specify the column names of the `departments` table. As you press Enter after each new line, Quick SQL will dynamically update the database definition language (DDL) of your data model in the right-hand text area (**Oracle SQL Output**). With three simple lines in Quick SQL, you now have DDL for the `departments` table, a primary key for the `departments` table, and a trigger to populate the primary key column with a globally unique ID (GUID). (At livesql.oracle.com, the default settings are different, so your DDL may not match this description exactly.)

Now update the data types and lengths for the `name` and `cost center` columns:

1. Type `vc100` after `name`, to change the length of the VARCHAR2 data type of the column to 100 characters.
2. Type `/nn` after `name vc100`, which is a modifier in Quick SQL to mark a column as not null (always requiring a value).
3. Type `num` after `cost center` to change the data type of the column to NUMBER.
4. Click the **Generate SQL** button.
   The Quick SQL shorthand you entered should now result in this:

```
departments
  name vc100 /nn
  cost center num
```

To complete the `departments` table, you will want to normalize the values in the name column to all uppercase, apply a unique constraint (to ensure that the name

values of all rows in the table are unique), and supply the domain of values for the name column.

1. Type `/upper /unique` at the end of the `name` column line, followed by `/check hr, development, legal, sales, operations`.
2. Click **Generate SQL**.
   The Quick SQL shorthand you entered should now result in this:

```
departments
  name vc100 /nn /upper /unique /check hr, development, legal, sales, operations
  cost center num
```

With this little bit of Quick SQL shorthand, you have a `departments` table with a named primary key, unique and check constraints, and a database trigger to generate the primary key value and normalize the `name` column to uppercase. Pretty easy. But it gets better.

ADDING A CHILD TABLE

One of the hallmarks of relational databases is the ability to declaratively express relationships between tables. These are typically expressed via referential integrity constraints and are used to ensure that the data represented in associated tables is consistent. To put this into practical terms, you will now define a table for employees, and the `employees` table will be a child of the parent `departments` table. You will not be able to insert a row into the `employees` table and reference a department that does not exist.

Defining a child table in Quick SQL is easy. All you need to do is indent the lines of the child table directly underneath the parent table.

1. Start a new line after the last line of the `departments` table definition, indent two spaces, and type `employees.`
2. On the next line, indent four spaces and type `first_name`. Continue adding new lines, indenting each line four spaces, and type the column names `last_name`, `email`, `hire_date`, `city`, and `country`.
3. Convert the `email` column to lowercase, by adding `/lower` after the email column.
4. Click **Generate SQL**.
   The Quick SQL shorthand you entered should now result in this:

```
departments
  name vc100 /nn /upper /unique /check hr, development, legal, sales, operations
  cost_center_num
employees
  first_name
  last_name
  email /lower
  hire_date
  city
  country
```

Review the generated DDL in the **Oracle SQL Output** text area. You now have a second table for employees; a `department_id` column with a named foreign key constraint to the `employees` table; and, following best practices, an index on the column of the foreign key constraint (`department_id`).

ABOUT ORACLE APPLICATION EXPRESS

Oracle Application Express (Oracle APEX) 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.

ADDING A SECOND RELATED TABLE, GENERATING DATA, AND EXPLORING OPTIONS

As you just saw, you can represent the relationship between a parent table and a child table with simple indentation in Quick SQL. But what if there are multiple relationships to the same table? You can easily represent those relationships via the `/references` column modifier, which is used to explicitly define a relationship between tables.

Each employee has a job. Now define the jobs in a separate table.

1. At the beginning of the Quick SQL shorthand, enter the definition of the `jobs` table as

```
jobs
  name /upper /unique /check developer, analyst, manager, assistant
```

2. At the end of the `employees` table, add the following line as a new column of the `employees` table:

```
job id / references jobs.
```

The Quick SQL shorthand you entered should result in this:

```
jobs
  name /upper /unique /check developer, analyst, manager, assistant
departments
  name vc100 /nn /upper /unique /check hr, development, legal, sales, operations
```

```
cost center num
employees
  first_name
  last_name
  email /lower
  hire_date
  city
  country
  job id /references jobs
```

The `employees` table now has an additional named foreign key constraint referencing the new `jobs` table.

There are several settings you can use to control the generation of the DDL and also enable additional features. You can, for example, specify a prefix to use for all object names, change the primary key value generation to use identity data types, and even generate a PL/SQL API.

Now enable the Audit Columns feature:

1. Click the **Settings** link.
2. In the **Additional Columns** section, select **Audit Columns**.
3. In the **Options** section, change **APEX Enabled** to **Yes** .
4. Click the **Save Changes** button.

Selection of the **Audit Columns** option means that every table now includes four additional columns to track who created or updated the row and when. The database triggers for the tables have also been updated to automatically populate the audit columns. Setting **APEX Enabled** to **Yes** affects the **Audit Columns** logic so that

the authenticated username of an Oracle APEX application is correctly referenced in the database trigger code.

It's far easier to prototype an application when you can see it with data, even if it's dummy data. Fortunately, Quick SQL has an easy facility for generating sample data with the `/insert` table directive. After every table, you can append `/insert NN`, where `NN` is the number of rows of data to generate.

Now use Quick SQL to generate some sample data for the tables:

1. After the `jobs` table, add `/insert 4`.
2. After the `departments` table, add `/insert 5`.
3. After the `employees` table, add `/insert 100`.
4. Click **Generate SQL**.

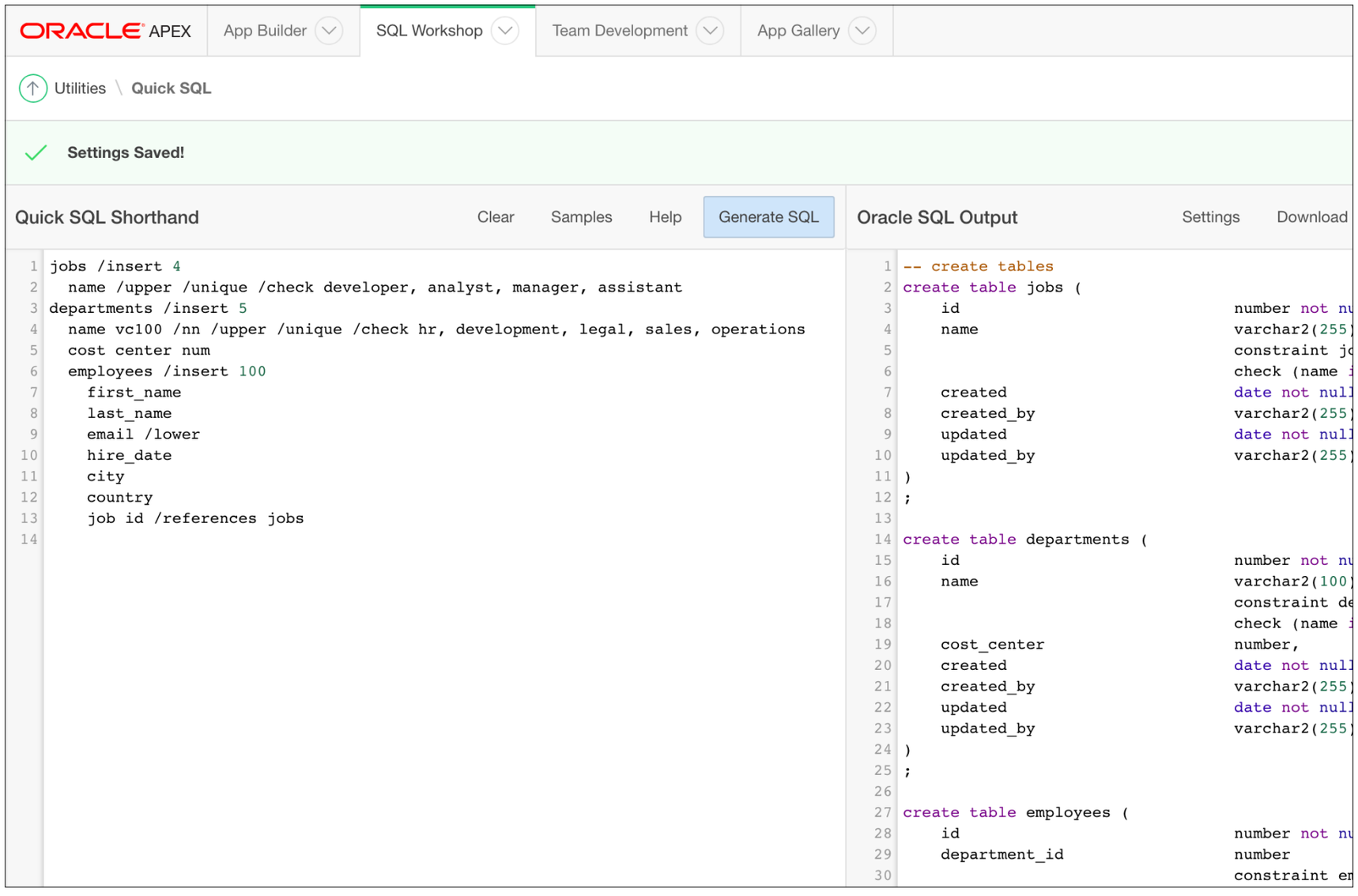
The Quick SQL shorthand you entered should now result in this:

```
jobs /insert 4
  name /upper /unique /check developer, analyst, manager, assistant
departments /insert 5
  name vc100 /nn /upper /unique /check hr, development, legal, sales, operations
  cost center num
  employees /insert 100
    first_name
    last_name
    email /lower
    hire_date
    city
    country
```

```
job id /references jobs
```

And there you have it. With 14 lines of Quick SQL shorthand, you have now generated a well-formed and properly named data model—shown in **Figure 1**—with

Figure 1: The completed data model and SQL script



three tables, indexed foreign-key constraints, named check constraints and unique constraints, database triggers to audit changes made to the tables, and sample data. More than 2,000 lines of SQL script. Amazing!

CREATING THE ORACLE APEX APPLICATION

If you're using Quick SQL within Oracle APEX, you can save and execute the statements of your generated SQL script to quickly create an Oracle APEX application on this data model.

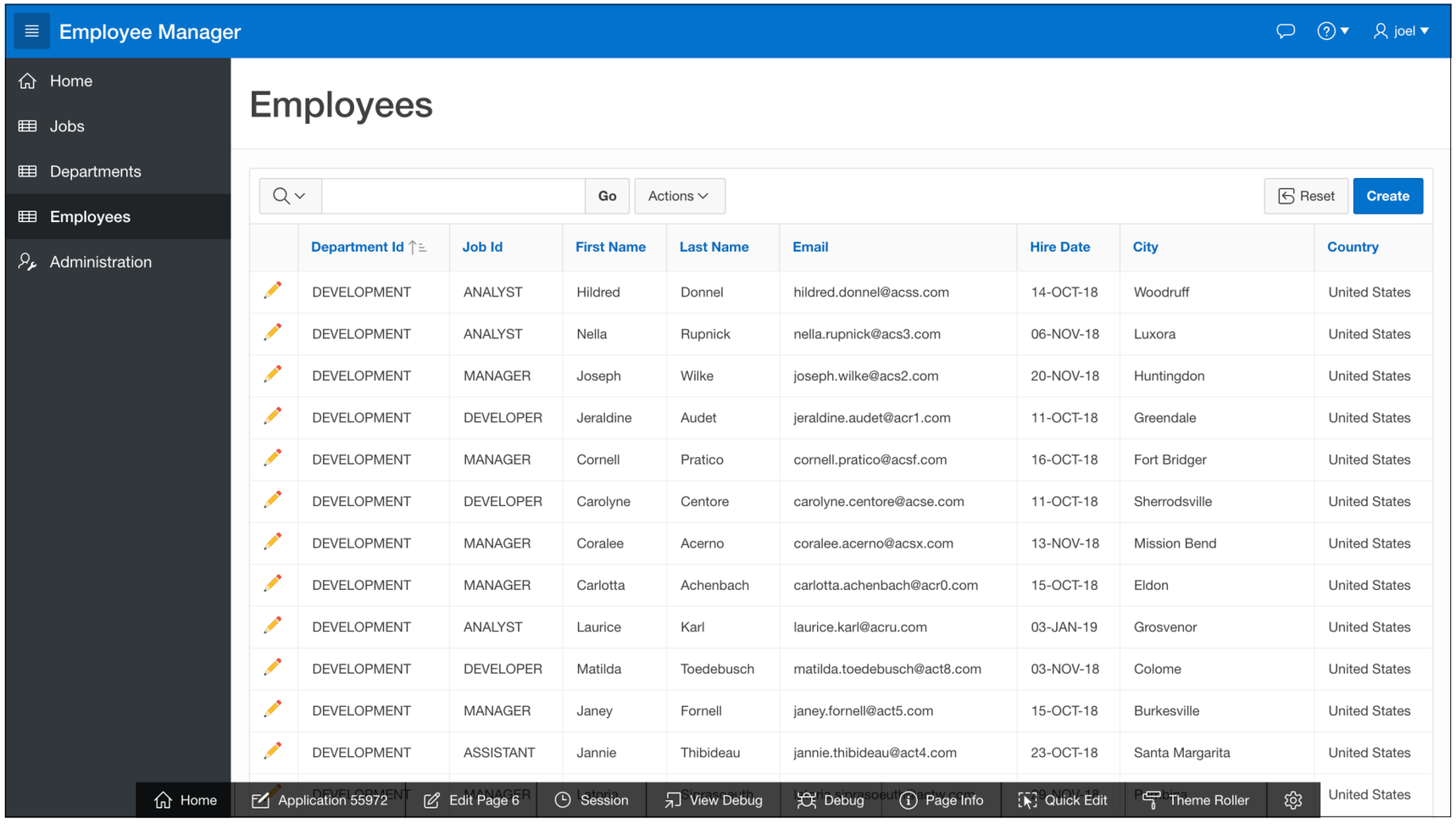
1. Click the **Save SQL Script** button.
2. For **Script Name**, type `QuickSQLDemo` and click **Save Script**.
3. Click **Review and Run**.
4. Click **Run**, and then click **Run Now**.

After the tables and triggers have been created and the sample data inserted, it's time to create your application.

1. Click **Create App from Script**.
2. For **Name**, type `Employee Manager`.
3. In the **Features** region, click **Check All**.
4. Click **Create Application**.
5. Click the **Run Application** icon, and log in to your application.

You can navigate among the different pages of your application, with reports and forms on your new tables. Note that on the form for editing employees, the job and department fields are controlled via a select list, constraining the values to those defined in the related tables. With 14 lines of Quick SQL shorthand and no code, you now have a 28-page responsive web application, shown in **Figure 2**, complete with user feedback, activity monitoring, and access control!

Figure 2: The Oracle APEX application on the new data model



CONCLUSION

An application is only as good as its data model. Although Quick SQL is not a replacement for the sophisticated data modeling capabilities of Oracle SQL Developer Data Modeler, rapid and concise data model development with Quick SQL is ideal for many opportunistic applications. Available as a service for all (via livesql.oracle.com) and tightly integrated with Oracle APEX, Quick SQL enables

you to quickly generate complete and accurate data models, along with sample data. Using the combination of the low-code rapid application development of Oracle APEX and Quick SQL is an excellent way to design and develop your next opportunistic application.

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

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

DOWNLOAD Oracle Application Express 18.2.

LEARN more about Oracle APEX.

TRY Oracle Database Cloud services.

Oracle Live SQL.

REQUEST a free Oracle APEX workspace.

ORACLE DATABASE

SODA and PL/SQL

Use the SODA API for PL/SQL to work with JSON—and without SQL—in Oracle Database.

SQL—Structured Query Language—is the most powerful and widely used language for manipulating data, primarily in relational databases such as Oracle Database. Its declarative and set-oriented approach handles a wide variety of use cases and produces concise, elegant ways to both query and change data.

For some scenarios and some developers, however, relational data and SQL are not seen as a good fit. The flexibility of a JSON document can, for example, better meet the demands of a rapidly changing schema or one that is poorly defined at the outset of a project. For documents in such cases and others, Oracle Database offers Simple Oracle Document Access (SODA) APIs. This set of NoSQL ("not only SQL") –style APIs enables you to create and store collections of documents in Oracle Database. You can retrieve and query them without needing to know SQL or how the documents are stored in the database.

SODA APIs are available for Java, C, REST, Python, Node.js, and PL/SQL. This article explores the basic concepts behind SODA, as exposed through the SODA API for PL/SQL. Links at the end of the article provide more details on SODA generally and on JSON.

OVERVIEW

All of the SODA APIs share the same concepts and flow. First, because the point of SODA is to relieve a developer of the need to know SQL, the APIs are not table-focused but are instead document-centric. Use a SODA API to manage (create, read, update, delete) documents of just about any kind, including videos, images, and—most commonly—JSON.

Documents are organized into collections. You can have one collection for *all* your documents, you can create a collection for each *type* of document (my video collection, my song collection, and so on), or you can create collections for *different components* of your application.

You can query the contents of documents by using pattern matching (query-by-example) or document keys.

You might be wondering why anyone who writes PL/SQL would be interested in avoiding SQL and instead opt for a SODA API?

Most back-end database developers will, of course, stick to the normal way of using PL/SQL: as a way to enhance the SQL language and provide additional security and as a means of implementing business logic.

In large enterprises that have Oracle Database installed, however, more and more front-end (and/or full-stack) developers want to work with document databases. With the wide array of SODA APIs now available for Oracle Database, they can have the best of both worlds: the power and security of the world's best rela-

tional database combined with the ease of use and flexibility of JSON-based document management with easy-to-use NoSQL-style SODA drivers for various programming languages.

In addition, the PL/SQL SODA API makes it possible for database developers to access collections and documents created through other SODA APIs. Thus, a JavaScript developer could use the Node.js SODA API to load JSON documents into the database. A SQL-savvy back-end developer could then bring the full power of SQL to that data: indexing access to the documents and building efficient analytic queries against them.

All PL/SQL SODA operations are made available through the new-to-Oracle Database 18c `DBMS_SODA` package and several object types, including `SODA_collection_t` and `SODA_document_t`. To use the package and manage SODA collections and documents in your schema of choice, you'll need to grant the `SODA_APP` role to that schema.

That's all you need to get going with SODA in PL/SQL! Let's start exploring.

WORKING WITH COLLECTIONS

Before you can put documents into a collection, you have to create it. In the following block, I create a collection to hold information about my friends. I then use the `get_name` and `get_metadata` methods of the `SODA_collection_t` type to display that information.

```
DECLARE
   l_collection   soda_collection_t;
BEGIN
   l_collection := dbms_soda.create_collection ('MyFriends');
```

```
   DBMS_OUTPUT.put_line (
      'MyFriends ID = ' || l_collection.get_name ());
   DBMS_OUTPUT.put_line (
      'MyFriends JSON metadata = ' || l_collection.get_metadata ());
END;
/

MyFriends name = MyPhotos
MyFriends JSON metadata =
{
      "schemaName": "HR",
      "tableName": "MyFriends",
"keyColumn": {
      "name": "ID",
      "sqlType": "VARCHAR2",
      "maxLength": 255,
      "assignmentMethod": "UUID"
},
"contentColumn": {
      "name": "JSON_DOCUMENT",
      "sqlType": "BLOB",
      "compress": "NONE",
      "cache": true,
      "encrypt": "NONE",
      "validation": "STANDARD"
},
```

```
                "lastModifiedColumn": {
                      "name": "LAST_MODIFIED"
                },
                "versionColumn": {
                      "name": "VERSION",
                      "method": "SHA256"
                },
                "creationTimeColumn": {
                      "name": "CREATED_ON"
                },
                "readOnly": false
          }
```

Since I passed only a name to `create_collection`, all the defaults were used for the metadata, which means in part that only JSON documents can be stored in the collection, audit information about the collection is tracked in predefined column names, and the SQL type used for documents is BLOB.

You can override the metadata defaults by supplying a second argument in the call to `create_collection`, in the form of a JSON document with the values you want to change.

Once a collection is created, I can perform these operations:

- Open a collection
- Drop a collection
- Get a list of all the collections in the user's schema

You must open a collection before you perform read or write operations. The `open_collection` function accepts a collection name and returns an instance of

type `SODA_collection_t`. If the function returns NULL, then no collection exists with that name.

The `list_collection_names` function returns the names of all available collections in the current schema as a PL/SQL nested table of type `SODA_collname_list_t`.

The following code demonstrates all of these subprograms.

```
CREATE OR REPLACE PROCEDURE show_soda_collections (title_in IN VARCHAR2)
IS
   l_collections   soda_collname_list_t := dbms_soda.list_collection_names;
BEGIN
   DBMS_OUTPUT.put_line (title_in);

   FOR indx IN 1 .. l_collections.COUNT
   LOOP
      DBMS_OUTPUT.put_line (
         'Collection ' || indx || ' is named ' || l_collections (indx));
   END LOOP;
END;
/

DECLARE
   l_collection   soda_collection_t;
BEGIN
   l_collection := dbms_soda.create_collection ('Coll1');

   l_collection := dbms_soda.create_collection ('Coll2');
```

```
l_collection := dbms_soda.open_collection ('Coll3');

IF l_collection IS NULL
THEN
   DBMS_OUTPUT.put_line ('No collection named "Coll3"');
END IF;

show_soda_collections (title_in => 'Before Drop');

IF dbms_soda.drop_collection ('Coll1') = 1
THEN
   show_soda_collections (title_in => 'After Drop');
   END IF;
END;
/

No collection named "Coll3"

Before Drop
Collection 1 is named Coll1
Collection 2 is named Coll2

After Drop
Collection 1 is named Coll2
```

WORKING WITH DOCUMENTS

Well, a collection all by itself doesn't really get you anywhere. What you need is some documents!

Once the collection is created, you use methods of the `SODA_collection_t` type to manage documents in that collection as follows:

Insert a document into a collection	`insert_one` and `insert_one_and_get`
Find a document in a collection	`find_one`
Remove a document from a collection	`remove_one`
Replace a document in a collection	`replace_one` and `replace_one_and_get`

Note that the two `and_get` methods return all the information about the new or replaced document *except* for the content. You can get that data by calling the `find_one` method to retrieve all the document's information.

In the block below, I insert two new documents, using each of the two insert methods. I then demonstrate how `find_one` works.

```
DECLARE
   l_collection     soda_collection_t;
   l_document       soda_document_t;
   l_new_document   soda_document_t;
BEGIN
   l_collection := dbms_soda.create_collection ('WithDocuments');
```

```
IF l_collection.insert_one (
      soda_document_t (
         b_content   => UTL_RAW.cast_to_raw (
                           '{"friend_type":1,"friend_name":"Lakshmi"}'))) = 1
THEN
   DBMS_OUTPUT.put_line ('BLOB document inserted');
END IF;

l_new_document :=
   l_collection.insert_one_and_get (
      soda_document_t (
         b_content   => UTL_RAW.cast_to_raw (
                           '{"friend_type":2,"friend_name":"Samuel"}')));

   DBMS_OUTPUT.put_line ('Samuel''s key: ' || l_new_document.get_key);
   DBMS_OUTPUT.put_line (
      'Samuel''s media_type: ' || l_new_document.get_media_type);
END;
/

BLOB document inserted
Samuel's key: 1697CFFB902A4FC2BFAD61DA31CF3B07
Samuel's media_type: application/json
```

Now let’s try to insert a CLOB document. To do that, I can specify the `C_CONTENT`

parameter name in my named notation invocation of `insert_one` (in the previous block, I referenced the `B_CONTENT` parameter).

```
DECLARE
   l_collection   soda_collection_t;
   l_document     soda_document_t;
BEGIN
   l_collection := dbms_soda.create_collection ('WithDocuments');

   IF l_collection.insert_one (
         soda_document_t (
            c_content   => '{"friend_type":1,"friend_name":"Lakshmi"}')) = 1
   THEN
      DBMS_OUTPUT.put_line ('VARCAHR2 document inserted');
   END IF;
END;
/

ORA-40659: Data type does not match the specification in the collection metadata.
```

The call to `insert_one` failed, precisely for the reason referenced in the error message. When you create a collection using all default metadata, the document type defaults to BLOB. So my first insertion worked but the CLOB payload resulted in an error. There are two ways to fix this: (1) cast the CLOB to BLOB with `DBMS_LOB.CONVERTTOBLOB` or (2) create the collection specifying CLOB as the data type for the collection, as follows:

```
DECLARE
  c_use_clob constant   VARCHAR2(1000) :=
       '{"keyColumn":{"assignmentMethod": "CLIENT"},
        "contentColumn":{"sqlType": "CLOB"}}';
  l_collection  soda_collection_t;
BEGIN
  l_collection := DBMS_SODA.create_collection('ClobCollection', c_use_clob);
END;
```

I've shown you how to insert a document. What if you need to remove one? Call the `remove_one` method. This method accepts the key for your document, which you can obtain from the document's `get_key` method.

In the block below, I

- Create a collection
- Insert a document and get back the metadata
- Use the `find_one` method to verify that it is there
- Invoke the `remove_one` method to remove the document
- Call `find_one` again to confirm that it is gone

```
DECLARE
   l_collection       soda_collection_t;
   l_new_document     soda_document_t;
   l_found_document   soda_document_t;
BEGIN
   l_collection := dbms_soda.create_collection ('MyFriends');
```

```
l_new_document :=
   l_collection.insert_one_and_get (
      soda_document_t (
         b_content   => UTL_RAW.cast_to_raw (
                           '{"friend_type":2,"friend_name":"Samuel"}')));

/* Verify it's there using the find_one method and the key. */
l_found_document := l_collection.find_one (l_new_document.get_key);

IF l_found_document IS NOT NULL
THEN
   DBMS_OUTPUT.put_line (
      'Found document with key: ' || l_new_document.get_key);
END IF;

IF l_collection.remove_one (l_new_document.get_key) = 1
THEN
   DBMS_OUTPUT.put_line ('Document is removed.');
END IF;

/* Verify it's there using the find_one method and the key. */
l_found_document := l_collection.find_one (l_new_document.get_key);

IF l_found_document IS NULL
THEN
   DBMS_OUTPUT.put_line ('No document for key.');
```

```
   END IF;
END;
/

Found document with key: 42B13A837F964FD5BFC78238985448D2
Document is removed.
No document for key.
```

What if you need to change a document already in the collection? Remember, you are not using SQL, so you cannot use the handy `UPDATE` statement to apply a change. Instead, you must replace the existing document (found by key) with another document that contains the modified data.

In the block below, I do the following:

- Insert a document with a friend named “Helen”
- Call the `find_one` method to retrieve the document from the collection, with its content
- Display that JSON—verifying the name as “Helen”
- Create a new document with the name “Helena”
- Call the `replace_one_and_get` method, specifying the key of the original document and passing it the new document for replacement
- Verify that the name has been changed to “Helena”

```
DECLARE
   l_collection          soda_collection_t;
   l_original_document   soda_document_t;
   l_new_document        soda_document_t;
```

```
   l_replaced_document   soda_document_t;
BEGIN
   l_collection := dbms_soda.create_collection ('MyFriends2');

   l_original_document :=
      l_collection.insert_one_and_get (
         soda_document_t (
            b_content   => UTL_RAW.cast_to_raw (
                              '{"friend_type":3,"friend_name":"Helen"}')));

   /* Show the content */
   l_original_document := l_collection.find_one (l_original_document.get_key);

DBMS_OUTPUT.put_line (
      'Original content: '
   || UTL_RAW.cast_to_varchar2 (l_original_document.get_blob));

/* Create a document (not inserted into any collection) with
   the corrected name "Helena". */
l_new_document :=
   soda_document_t (
      b_content   => UTL_RAW.cast_to_raw (
                        '{"friend_type":3,"friend_name":"Helena"}'));

/* Replace the original document and verify the contents */
```

```
   l_replaced_document :=
      l_collection.replace_one_and_get (l_original_document.get_key,
                                        l_new_document);
   l_replaced_document := l_collection.find_one (l_replaced_document.get_key);

   DBMS_OUTPUT.put_line (
         'Replaced content: '
      || UTL_RAW.cast_to_varchar2 (l_replaced_document.get_blob));
END;
/

Original content: {"friend_type":3,"friend_name":"Helen"}
Replaced content: {"friend_type":3,"friend_name":"Helena"}
```

BEHIND THE SCENES

The SODA API is implemented with tables and rows. When you create a collection, a table of the same name is created (with the case preserved):

```
DECLARE
   l_collection          soda_collection_t;
BEGIN
   l_collection := dbms_soda.create_collection ('SODACollection');
END;
/
SELECT table_name FROM USER_TABLES
 WHERE table_name like 'SODA%'
```

```
/

TABLE_NAME
__________________

SODACollection
```

Each document is stored as a row in the collection's table:

```
DECLARE
   l_collection soda_collection_t;
   l_document   soda_document_t;
BEGIN
   l_collection := dbms_soda.create_collection ('MyFriends');
   l_document :=
      l_collection.insert_one_and_get (
         soda_document_t (
            b_content   => UTL_RAW.cast_to_raw (
                              '{"friend_type":3,"friend_name":"Helen"}')));
END;
/
SELECT id "UTF8-Encoded JSON String" FROM "MyFriends"
/
UTF8-Encoded JSON String
____________________________________

50D58CF995704F70BFED14673D482A95
```

These tables are managed for you automatically through the SODA API.

SUMMARY

Most back-end database developers will, of course, stick to the normal way of using PL/SQL: to enhance the SQL language, provide additional security, and implement business logic.

Oracle Database is used more and more widely in distributed applications involving multiple kinds of data. In this context, strong support for both JSON and schema-less development via the SODA APIs will ensure that front-end developers can make the most of the features and power of Oracle Database.

In my next article on SODA and PL/SQL, I will explore the use of the builder pattern to specify read/write patterns in SODA and more.

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

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

READ
an introduction to SODA.

SODA for PL/SQL Developer's Guide.

WORK with the LiveSQL script for this article.

BY CONNOR McDONALD

ORACLE DATABASE

A Higher-Level Perspective on SQL Tuning

The commonly missed first steps of tuning a SQL statement

Fire up your favorite search engine, enter "SQL tuning" as the search term, and you are likely to detect a common theme in the results. They typically relate to modification of the database structure, such as adding or removing indexes; modification of the execution environment, such as gathering or modifying optimizer statistics; or more-substantial modifications such as rewriting the SQL or changing the physical design of the database. What is often missing in those same search engine results is a warning that any such modifications carry an inherent risk to the entire database environment in which the SQL statement is running. The larger the modification, the higher the risk.

As Oracle Database has evolved, the number of tuning tools has grown and the level of sophistication of each of these tools has also increased, giving developers and DBAs a plethora of options to explore. One consequence is that it is easy to dive

straight into utilizing these tuning tools without stepping back and asking a few key questions about the SQL statement first:

- What is the business functional requirement being served by the SQL?
- Is the SQL correct?
- Can this SQL *ever* run fast enough to meet the requirements?

Even without diving into low-level tools, it is easy to forget these questions. I frequently visit clients to assist with performance tuning issues, and upon my arrival, often the first thing presented to me is a single SQL statement, with no context or explanation surrounding it, and this plea: *"This is the problem! Please solve it."* It may seem counterintuitive, but the first step of SQL tuning is to forget about the SQL.

THE BUSINESS REQUIREMENT

No organization I'm aware of has ever had a business model of "Let's make sure SQL runs fast," unless that business was a SQL tuning consultancy! SQL statements, the applications that run them, and the IT departments that build and support those applications exist to meet a suite of *business* functional requirements. Those requirements may be created by the business as part of its desire to thrive commercially, or they may be imposed on the business by regulatory bodies. In either case, satisfying business requirements must be kept uppermost in a developer's mindset when it comes to tuning SQL, because it drives the very decisions made by developers in creating the database design and the SQL that runs on it. Ignoring the underlying business requirements is a catalyst for poorly performing SQL.

I'll demonstrate that with a real example of my experience with a client that requested some SQL tuning assistance. For the sake of anonymity, the descriptions and the SQL code are obfuscated, but other than that, this is a reasonably accurate

depiction of the events that transpired. The client had a typical online retail presence, where customers could create an account with the business to get access to discounts, promotional offers, and the like. Each customer (and hence each customer account) would buy goods from the business, and each purchase would be termed a customer "transaction." The business had a report that showed the last time each customer had completed a transaction, similar to what you see in **Figure 1**. As customer transaction volume grew over time, the report had started to run more slowly, which is what prompted the business to request my assistance.

Figure 1: Customer transaction report

Report: REP-020C **Page 1 of 17**

Customer Recent Transactions

Customer #	Last Transaction	Current Balance
3000114	16/07/2018 12:47:27	118.11
3000125	29/06/2018 14:04:00	209.79
3000136	10/08/2018 19:52:38	176.54
3000147	09/08/2018 17:32:09	144.60
3000156	09/06/2018 19:40:05	137.19
3000161	26/09/2018 04:10:45	207.86
3000176	11/07/2018 12:59:24	189.92
3000186	12/07/2018 18:45:48	149.37
3000192	07/06/2018 03:05:54	201.59
3000205	05/06/2018 14:42:58	128.49
3000213	27/09/2018 19:10:48	127.93
3000224	05/08/2018 08:27:28	116.97

Requested by: Connor McDonald
Data as at: November 14th, 2018

The SQL statement to produce the data for the report was a simple aggregation query, as shown in **Listing 1**.

Listing 1: Customer transaction query

```
SQL> select CUSTOMER_NUM,
  2         max(TRANS_TIMESTAMP) LAST_TS
  3  from   CUSTOMER_TRANSACTIONS
  4  group by CUSTOMER_NUM
  5  order by 1;
```

This returns one row per customer, which is then joined back to the `CUSTOMERS` table to get the current account balance in dollars.

I made the mistake (which in part motivated this article) of focusing solely on how to improve the performance of the SQL statement, rather than stepping back and investigating the business requirement.

First I tackled the problem by using the query shown in **Listing 2**, colloquially referred to as the KIWI (kill it with iron) approach, where more server resources are thrown at the query via parallelism.

Listing 2: More hands making less work

```
SQL> select /*+ PARALLEL(t) */ CUSTOMER_NUM,
  2         max(TRANS_TIMESTAMP) LAST_TS
  3  from   CUSTOMER_TRANSACTIONS t
  4  group by CUSTOMER_NUM
```

This improved the performance of the report, but to the detriment of other functions of the application, which subsequently struggled to obtain sufficient I/O resources from the server, because it was being hammered by parallel I/O slaves. Also, the response time of the report became less predictable, because it was dependent on the number of concurrent executions of the report and how many parallel slaves were available to a given report request. *Variability* of response time is often more frustrating to application users than *slow but consistent* performance; hence, parallelism was dismissed as a permanent solution.

The next alternative was to make the transaction table "thinner." This is a common technique in which the fields contained in either the `SELECT` clause or the predicates are added to an index so that only the index, rather than the full table, needs to be scanned. The index becomes a "thinner" version of the table. **Listing 3** shows this strategy and the resultant query execution plan.

Listing 3: Using an index as a thin table

```
SQL> create index CUSTOMER_TRANS_IX
  2     on CUSTOMER_TRANSACTIONS( CUSTOMER_NUM, TRANS_TIMESTAMP )

SQL> select CUSTOMER_NUM,
  2         max(TRANS_TIMESTAMP) LAST_TS
  3  from   CUSTOMER_TRANSACTIONS
  4  group by CUSTOMER_NUM
  5  order by 1;
```

```
-----------------------------------------------------
| Id | Operation             | Name                |
-----------------------------------------------------
|  0 | SELECT STATEMENT      |                     |
|  1 |  INDEX FAST FULL SCAN | CUSTOMER_TRANS_IX   |
-----------------------------------------------------
```

This decreased the report response time, but I stressed to the client that this was only a temporary solution, because as the transaction volume continued to increase, even an index being used as a thin version of the transaction table would ultimately mean that the performance problems would recur.

Other solutions (partitioning, compression, and materialized views) were considered as well, the details of which I'll omit for brevity's sake, but in each case, the performance benefits also came with side effects that were undesirable for this client. This is not to dismiss any of these or the above techniques out of hand. All of the solutions are potentially valid in other use cases, but were just not for this particular client.

I convinced the development team to ask the business users what the motivation for this report was. After all, a report that shows the "most recent" of *any* high-volume activity is always out of date the moment it has been run. So I was curious to discover what value the report gave the business. The response was that the report was used to identify those customers who had not visited the website recently, so that they could be offered incentives via promotional offers to return to the site and become active again.

Understanding the *business* requirement made the solution to tuning the SQL trivial. All it took was a look at the columns in the `CUSTOMERS` table, as shown in **Listing 4**.

Listing 4: CUSTOMERS table

```
SQL> desc CUSTOMERS

 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 CUSTOMER_NU                               NOT NULL NUMBER(8)
 CURRENT_BALANCE                           NOT NULL NUMBER(14,2)
 ...
 ...
 ...
 LAST_LOGIN_TIME                           NOT NULL TIMESTAMP(6)
```

There was no need to query the large `CUSTOMER_TRANSACTIONS` table at all! The last visit to the website for each customer was already being captured in metadata on the `CUSTOMERS` table. The `LAST_LOGIN_TIME` had been implemented as part of a password expiry mechanism but could now also be used for recent-activity checks.

This is a key point of "tuning" SQL. Without consulting with the business users and getting their requirements, there is no way the original SQL could have been recast to avoid querying the `CUSTOMER_TRANSACTIONS` table. Before you tune a SQL statement, you *must* understand the business requirement that led to it.

SQL CORRECTNESS

Once the business functional requirement has been confirmed, you can turn your attention to the SQL statement. Just because a SQL statement returns without error, or even if the statement *returns the correct results*, that does *not* constitute a

guarantee that the SQL statement is actually correct. Often the cause of poorly performing SQL is malformed construction, which can easily slip through testing cycles if the query results are still plausible (especially with small test datasets). Hence, when I'm asked to tune a SQL statement, I will spend a few minutes before tackling any performance-related avenues making sure that the SQL statement does not have any obvious syntax errors. Here are some of the common errors I see that typically cause SQL statements to be misdiagnosed as performance problems.

Order of operations. My children are just completing primary school, and in their mathematics classes, they use the acronym BIMDAS. Many (many!) years ago, when I was a similar age, it was called BODMAS, but the premise was the same. The acronym is a simple way of remembering the order of mathematical operations (BIMDAS = brackets first, then indices, then multiplication/division, and finally addition/subtraction), which is why "2 times 3 plus 5" evaluates to 11, not 16.

The same rules apply to the processing order of logical operations within the predicates of a SQL statement, and failing to observe the correct ordering can lead to performance issues. Business requirements are often given in language within which there is a natural or implicit ordering of operations, which can lead to errors when transposed to SQL code. For example, the requirement

> "For regions in California, find the highest transaction amount for sales consultants, where the tax levy is more than 10% or the government subsidy is nonzero."

could yield the SQL query shown in **Listing 5**.

Listing 5: Business requirement translated to SQL query

```
SQL> select REGION, max(SALE_AMOUNT)
  2  from   EMP e,
  3         SALES_TRANSACTIONS s
  4  where  e.JOB     = 'SALESMAN'
  5  and    s.EMPNO   = e.EMPNO
  6  and    s.LOCATION  = 'CA'
  7  and    s.TAX_LEVY > 10
  8  or     s.GOVT_SUBSIDY > 0
  9  group by REGION;
```

But this is probably an incorrect translation, due to the ordering of operations implied, but not explicitly stated by, the business requirement. It is most likely (and worth confirming with the business stakeholders) that the requirement when phrased with more precision was

- "First, identify the set of transactions for sales consultants from regions in California.
- Then, with that set of data, filter where the tax levy is more than 10% or the government subsidy is nonzero.
- Then, with that reduced set of data, find the highest transaction amount per region."

The SQL in **Listing 5** does not fulfill this requirement and, moreover, is likely to have performance issues, because the final predicate of `OR GOVT_SUBSIDY > 0` becomes a standalone predicate that is not associated with the other conditions or even the joins. Because the query performs an aggregation on the `REGION` column,

such errors can easily slip through testing phases unnoticed, since even though vastly more rows will be processed by the query, the aggregation will reduce the set down to a small list of regions. Recalling the BIMDAS lessons of my children, the query should be corrected as shown in **Listing 6**.

Listing 6: Corrected SQL query

```
SQL> select REGION, max(SALE_AMOUNT)
  2  from   EMP e,
  3         SALES_TRANSACTIONS s
  4  where  e.JOB     = 'SALESMAN'
  5  and    s.EMPNO   = e.EMPNO
  6  and    s.LOCATION  = 'CA'
  7  and    ( s.TAX_LEVY > 10
  8           or  s.GOVT_SUBSIDY > 0 )
  9  group by REGION;
```

Missing join predicates. Developers who are still coming to grips with relational databases can make the mistake of thinking that joins are operations used solely to collect additional attributes for a result set, rather than being data *filters*. For example, when the requirement is to show a department name along with employee details, a join is required to collect the additional detail, as shown in **Listing 7**.

Listing 7: Simple join

```
SQL> select e.*, d.DNAME
  2  from   EMP e,
```

```
  3         DEPT d
  4  where  e.DEPTNO = d.DEPTNO;
```

A common coding mistake happens when each table in the SQL query has its own set of predicates. Because developers assume that all of the required filtering is handled by these predicates, the crucial join condition is forgotten, as in the example in **Listing 8**.

Listing 8: Missing join predicate due to presence of other clauses

```
SQL> select d.DNAME, max(e.SAL)
  2  from   EMP e,
  3         DEPT d
  4  where  e.JOB = 'SALESMAN'
  5  and    d.LOC = 'NORTH'
  6  group by d.DNAME;
```

As already mentioned, aggregations can mask the incorrectness of the SQL statement. It is only when the execution plan is examined that a Cartesian join, which can be a predictor of a future performance problem for large volumes of data, is detected (see **Listing 9**).

Listing 9: Missing join predicate leading to Cartesian cross-product

```
-------------------------------------------
| Id | Operation                | Name |
-------------------------------------------
|  0 | SELECT STATEMENT         |      |
```

```
|  1 |  HASH GROUP BY          |      |
|  2 |   MERGE JOIN CARTESIAN  |      |
|* 3 |    TABLE ACCESS FULL    | DEPT |
|  4 |    BUFFER SORT          |      |
|* 5 |     TABLE ACCESS FULL   | EMP  |
-------------------------------------
```

Even without `GROUP BY` aggregations, a common but erroneous approach I see to "fixing" SQL statements that have missing join conditions or an incorrect order of operations is adding the `DISTINCT` keyword. **Listing 10** starts with the query in **Listing 8** but just lists employee details without the `GROUP BY`.

Listing 10: Employee details still with the missing join predicates

```
SQL> select e.*
  2  from   EMP e,
  3         DEPT d
  4  where  e.JOB = 'SALESMAN'
  5  and    d.LOC = 'NORTH';

     EMPNO ENAME      JOB              MGR HIREDATE
---------- ---------- --------- ---------- ---------
      7499 ALLEN      SALESMAN        7698 20-FEB-81
      7499 ALLEN      SALESMAN        7698 20-FEB-81
      7499 ALLEN      SALESMAN        7698 20-FEB-81
      7499 ALLEN      SALESMAN        7698 20-FEB-81
```

```
7521 WARD       SALESMAN        7698 22-FEB-81
7521 WARD       SALESMAN        7698 22-FEB-81
7521 WARD       SALESMAN        7698 22-FEB-81
7521 WARD       SALESMAN        7698 22-FEB-81
```

The duplicated results are typically a clear enough indication to developers that the SQL is not correct, but rather than add the missing join predicates, they might add a `DISTINCT` keyword to "fix" the SQL, as in **Listing 11**.

Listing 11: Incorrect SQL with a false appearance of correctness

```
SQL>  select DISTINCT e.*
  2      from   EMP e,
  3             DEPT d
  4      where  e.JOB = 'SALESMAN'
  5      and    d.LOC = 'BOSTON';

     EMPNO ENAME      JOB              MGR HIREDATE
_________ _________  _________  __________ _________
      7654 MARTIN     SALESMAN         7698 28-SEP-81
      7521 WARD       SALESMAN         7698 22-FEB-81
      7499 ALLEN      SALESMAN         7698 20-FEB-81
      7844 TURNER     SALESMAN         7698 08-SEP-81
```

Using `DISTINCT` does not correct the SQL query, because the missing join predicate means that the results are still incorrect. Moreover, `DISTINCT` can be the cause of a perceived performance issue, because reducing a set of rows down to a distinct

set of values is a resource-intensive operation. I remember that in the very first SQL tuning course I attended, in the early 1990s, the instructor gave us a simple maxim: "Yes, the DISTINCT keyword has valid uses, but in general, Mr. Distinct is not our friend." Whenever I see the `DISTINCT` keyword in a SQL statement, I take some time to double-check that it is not masking some other error in the SQL query text.

Lack of aliasing. Aliasing tables in SQL statements and fully qualifying the columns referenced with those aliases represent more than just a maintenance convenience for the next developer who comes along to alter the SQL code. These steps also protect against silent errors in the SQL text that can cause performance problems. Consider the example in **Listing 12**, which identifies all sales conducted in regions that offered a "Black Friday" promotion.

Listing 12: Promotions on Black Friday

```
SQL> select *
  2  from   sales
  3  where  region in
  4    ( select region
  5      from   promoted_locations
  6      where  campaign = 'BLACK FRIDAY');
```

The SQL query looks plausible given the stated requirement and runs without error. However, it will most probably run poorly, because it will potentially return every single row in the `SALES` table. This is not discernible from the query text until I look at the column definitions for the `PROMOTED_LOCATIONS` table, as shown in **Listing 13**:

Listing 13: Columns for PROMOTED_LOCATIONS

```
SQL> desc PROMOTED_LOCATIONS
 Name                          Null?    Type
 ----------------------------- -------- ----------------
 ID                            NOT NULL NUMBER
 CAMPAIGN                      NOT NULL VARCHAR2(128)
 PROMOTED_REGION               NOT NULL VARCHAR2(24)
```

Note that there is no column called `REGION` and yet the SQL statement in **Listing 12**—with its references to the `REGION` column—runs *without* error. This is because the columns were not fully qualified in the SQL statement, so it becomes logically equivalent to the text in **Listing 14**:

Listing 14: Promotions on Black Friday

```
SQL> select sales.*
  2  from   sales
  3  where  sales.region in
  4    ( select sales.region
  5      from   promoted_locations
  6      where  promoted_locations.campaign = 'BLACK FRIDAY');
```

In the absence of a `REGION` column in the `PROMOTED_LOCATIONS` table in the subquery, the reference to the column will then resolve to that of the outer `SALES` table. On the assumption that there is at least one row in `PROMOTED_LOCATIONS`, the subquery becomes an "always true" result and thus all rows from the `SALES` table are

returned, most likely with performance problems associated with processing a large volume of rows. For this reason, I advocate to developers that they always fully qualify all expressions in a SQL statement with appropriate aliasing. In the example above, full aliasing would have immediately informed the developer of the wrong-column-name error in the SQL text, as shown in **Listing 15**.

Listing 15: Aliased SQL query

```
SQL> select s.*
  2  from   sales s
  3  where  s.region in
  4    ( select p.region
  5      from   promoted_locations p
  6      where  p campaign = 'BLACK FRIDAY');

ERROR at line 4:
ORA-00904: "REGION": invalid identifier
```

THE LAWS OF PHYSICS

Once business requirements are validated and SQL statements have been confirmed as being syntactically and functionally correct, there's one more step to take before you start digging into the lower-level performance tuning steps: Analyze the amount of work the SQL statement must do with the current database design. For example, if the business requirement is

> "Find the largest single transaction amount the company has ever had."

—unless that artifact is stored in some way by the existing database design or there are appropriate database structures (such as indexing) in place to satisfy that query—a SQL statement to handle that requirement will need to scan every transaction in the company's history. Business stakeholders can be unaware of the resource cost of requirements, especially when the requirement can be very simply phrased, as in the example above.

Sometimes a SQL "tuning" exercise is simply the task of explaining to business stakeholders what work the database will be undertaking to satisfy the business requirement. That can then lead to a decision about whether to modify the requirement or take action to modify the database structures to better align with that requirement.

SUMMARY

For IT professionals, there is always the temptation of "Let's just jump into the code," no matter what language the code is written in, and SQL code is no exception. And when it comes to the *tuning* of SQL, that temptation is even greater, considering that Oracle Database offers such a plethora of SQL tuning facilities. Throughout the various versions of Oracle Database, technologies such as SQL profiles, stored outlines, SQL plan baselines, tuning advisors, and optimizer hints all provide enormous opportunities for developers to improve SQL code performance.

But these technologies work on the underlying assumption that the SQL statement is correct in terms of its construction and that it is meeting the intended business requirement. It is important for developers and DBAs to validate this correctness before diving deep into the various tuning techniques at their disposal.

In my next article, I'll explore how to proceed with SQL tuning once the prerequisite steps of requirements validation and correctness testing have been completed.

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

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about SQL tuning.

DOWNLOAD Oracle Database 18c.

Can DBAs Relax About Security?

AIOUG president outlines new challenges and opportunities for India's database professionals. **BY LESLIE STEERE**

Want to talk about a big job? How about launching and managing a single Oracle user group to serve the seventh-largest country in the world (at about 1,269,219 square miles), in which information technology accounts for 7% to 8% of the GDP, with much of that work outsourced from other countries? Then you have some sense of the large-scale challenges facing Sai Penumuru, president of the All India Oracle Users Group (AIOUG). *Oracle Magazine* caught up with Penumuru recently to see just how those challenges are evolving in the new world of autonomous services and other emerging technologies.

Begun in 2007, AIOUG is now one of the largest Oracle user groups in the

DBAs in India manage volumes of infrastructure "not seen anywhere else in the world," says AIOUG President Sai Penumuru, so Oracle Autonomous Database will allow them to focus on other areas such as big data and data modeling.

Asia-Pacific region, with eight chapters across the country and about 10,000 members. “We started as one group,” says Penumuru, who is also chief technologist at DXC Technology, an Oracle ACE Director, and an Oracle Groundbreaker Ambassador and somehow manages to wear all four hats with élan. “But at a certain stage, we realized that one person or one board of directors could not manage covering the whole of India.” So AIOUG set out to expand to different areas, establishing common standards and bylaws for each new chapter to follow. “There are a lot of benefits to having one single user group,” says Penumuru about this hierarchical structure. “We can approach Oracle as one voice, and we get really good support.”

INDIAN DBAS IN AN AUTONOMOUS WORLD

Its large population of skilled workers and cost-effective tax and wage structures, among other advantages, have made India a primary destination for IT outsourcing since the early 1990s—meaning that today database professionals in India “manage hundreds or thousands of computing resources for large multinational customers,” says Penumuru. “They have enormous workloads.”

That makes technology advances such as Oracle Autonomous Database both frightening and attractive to AIOUG members. “There is always an initial fear from anyone in IT when discussions about automation come up,” says Penumuru. “That is natural. People think about the security of their career. But for many AIOUG members, autonomous should help them, not hinder them. Indian consultancies and data centers manage volumes of infrastructure that are not seen anywhere else in the world, so tools and mechanisms to reduce the workload of our members will be a good thing.”

With an autonomous database taking care of most of the administration and management tasks, he says, “we can focus on other areas like database mod-

eling, security, and big data. So it is not really cutting your jobs. You are basically shifting your career to different areas." And that is the message AIOUG members need to hear more often from the experts at Oracle, Penumuru adds. "Oracle Autonomous Database is really good news for us," he says. "It is really a vacation time for DBAs, because [today] we are spending most of our time doing the admin activities."

AUTONOMOUS: VACATION TIME FOR DBAS?

In a country where database professionals are managing computing resources for huge multinationals from around the world, security has to be a top priority and DBAs spend an inordinate amount of time and energy ensuring that their clients' data is safe.

But with Oracle Autonomous Database, says Sai Penumuru, president of the All India Oracle Users Group, "I can say it's not such a concern anymore. Because initially, as DBAs, we need to apply CPU patches. And I think most organizations don't follow the schedule. So we always have a threat, right? Now we don't. We can say it's like a holiday time for DBAs, because most of [the threats] are covered with Oracle Autonomous Database. I think even if there is some security attack in Oracle Autonomous Database—even if we don't know about it—the database is going to fix that. So I think it's a good time for DBAs to take a vacation."

SECURING DATABASES—AND JOBS

Many of AIOUG's members are involved in ensuring high-availability solutions for customers, says Penumuru, "and perhaps the biggest challenge to high availability nowadays is not infrastructural but the risk of getting hacked."

For Indian IT workers, he says, "customer trust is perhaps the new definition of high availability. It is not just having your systems available that matters but that the data in them is secure and that you can be trusted with people's data." Most organizations "don't apply security patches—CPU patches—on time," he adds. Besides, "just when one security incident hits the newspapers and is fixed, new and innovative ways for people to breach your security mechanisms quickly come along." Now,

seeing that Oracle is doing things such as automated security patching, “I feel as a DBA, I’m actually relaxed, because Oracle is taking care of it.”

Database professionals today, he says, need to have more-general skills rather than focusing tightly on specific roles such as backup DBA or performance DBA. There will be a shift from specialist options such as DBA or Java developer to customers wanting IT people with a broader skill spectrum, Penumuru adds. “For example, DBAs will look after cloud networks, and developers will have operations-style experience for DevOps.”

And user groups need to help their members prepare for these new realities. “We need to support new technologies for our members such as microservices, DevOps, and blockchain,” he says.

AIOUG addresses these needs with a multipronged approach. Its chapters each run monthly Tech Days focused on specific technologies. The group’s annual two-day event, Sangam, is “something like a mini Oracle OpenWorld,” says Penumuru, and covers a range of technologies. And its Oracle Groundbreakers Yatra is a seven-city tour where developers can meet with Oracle experts and learn about the latest technologies.

Because AIOUG supports members across such a large region, “webinars and remote education are essential,” says Penumuru; for example, members take advantage of the publicly available Ask TOM Office Hours to learn from live Q&A sessions with experts on various aspects of Oracle technologies. The user group also publishes a monthly magazine, *Connect*, to keep members informed.

Leslie Steere is editor at large for Oracle Content Central.

PHOTOGRAPHY BY
ORACLE DIGITAL MEDIA PRODUCTION

NEXT STEPS

WATCH a Sai Penumuru video. **JOIN** AIOUG. **LEARN** more about Sangam.