ORACLE MAGAZINE

SEPTEMBER/OCTOBER 2018

Autonomous in Action

Choose self-managing, self-repairing, and self-securing Oracle Cloud autonomous services to break down barriers between people, places, data, and systems

AUTONOMOUS IS EASY

DEVELOPMENT OPENS UP

MOLECULAR DYNAMICS FOR THE MASSES

VISIT US AT ORACLE OPENWORLD
Oct. 22-25 • Booth #2619

Cloud | Hosted | On-Premise

Vertex and Oracle® Partnering for Global Tax Solutions

- Oracle® ERP Cloud
- Oracle® E-Business Suite
- Oracle® + NetSuite
- JD Edwards EnterpriseOne
- PeopleSoft Enterprise

Learn More

ORACLE Platinum Partner Cloud Standard

ORACLE Validated Integration Oracle E-Business Suite

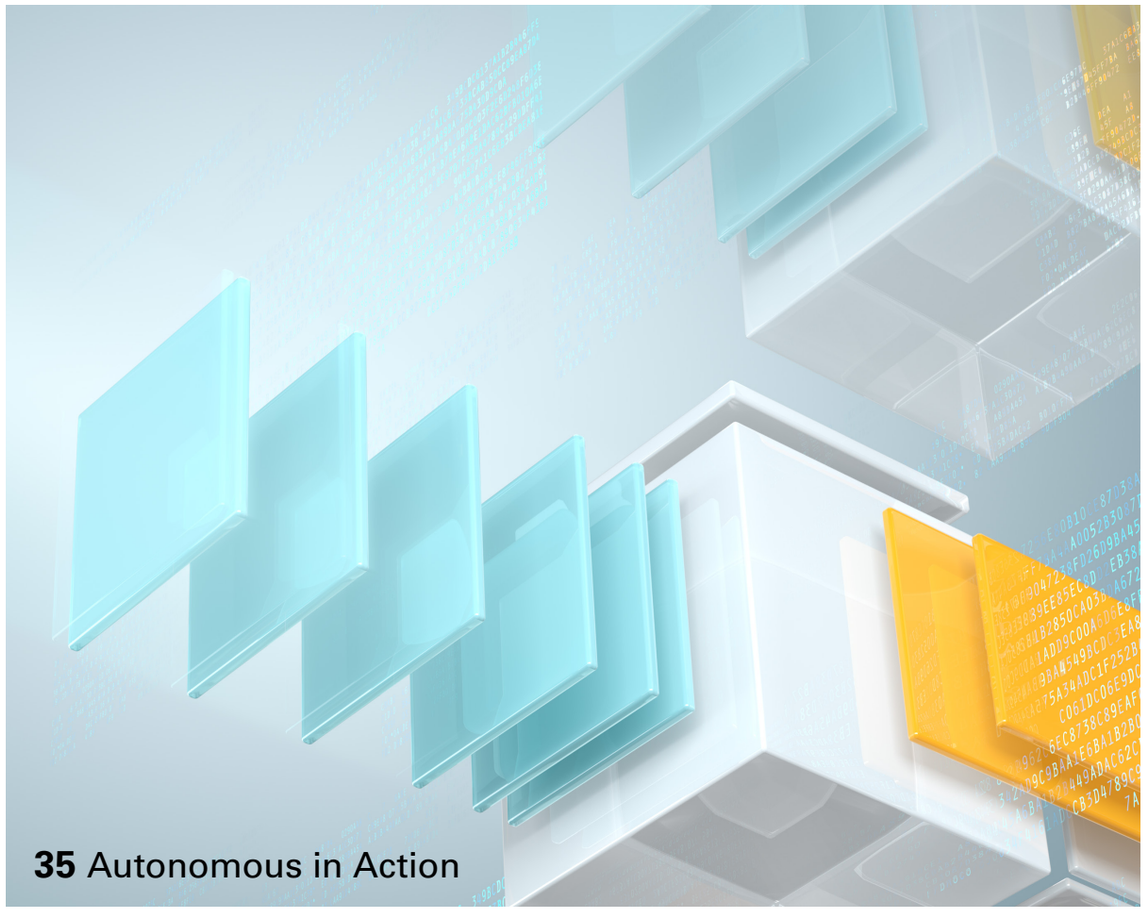
Available on ORACLE Cloud Marketplace

ORACLE NETSUITE

VERTEX®

CONTENTS

ORACLE MAGAZINE



35 Autonomous in Action

FEATURES

UP FRONT

15 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.

Is bad data threatening your business?

Call in the Fabulous 4

It's Clobberin' Time...with Data Verify Tools!

Visit Melissa Developer Portal to quickly combine our APIs (address, phone, email and name verification) and enhance ecommerce and mobile apps to prevent bad data from entering your systems.

With our toolsets, you'll be a data hero – preventing fraud, reducing costs, improving data for analytics, and increasing business efficiency.

- **Single Record & Batch Processing**
- **Scalable Pricing**
- **Flexible, Easy to Integrate Web APIs: REST, JSON & XML**
- **Other APIs available: Identity, IP, Property & Business**

Let's Team Up to Fight Bad Data Today!

melissadeveloper.com
1-800-MELISSA

melissa®

Tom Haunert

DBAs: No Early Retirement

It's time for database administrators to start thinking like data experts, not caretakers.

On August 7, 2018, Oracle Executive Chairman and CTO Larry Ellison announced the availability of Oracle Autonomous Transaction Processing Cloud, part of Oracle Autonomous Database Cloud. And for Oracle DBAs, he had some straightforward career advice. For any Oracle DBAs who might have been thinking that autonomous technology meant their careers are done, Ellison was direct: "I've got some bad news for you. No early retirement."

Ellison went on to point out that, yes, some tedious tasks would no longer be required of DBAs with autonomous databases, including allocating new storage and upgrading the operating system. With those things out of the way, DBAs will be able to focus on the data, as well as working more closely with their developer colleagues.

"DBAs should be . . . trying to maximize the value of the data, create more applications, improve the analytics, improve the insights," Ellison said. "It's nice to automate the tedious stuff away so you can focus on the mission, which is to get the most out of your data and build new applications for your customers."

WANTED: DATA EXPERTS

For more on the many time-saving features and benefits of Oracle Autonomous

Transaction Processing Cloud, available now, check out this issue's cover story, "Autonomous in Action." Here's a sneak peak: the new service delivers 99.995% uptime, automatic patching, lower operating and labor costs, and "nothing-to-learn" administration.

In "Autonomous Is Easy," Oracle Executive Vice President Amit Zavery describes the common features and strategies for all Oracle Cloud Platform autonomous services and calls out some specific AI and machine-learning-powered features in the different services. In "Development Opens Up," Mike Hichwa, vice president of software development at Oracle, talks about database development, Oracle Autonomous Database Cloud, open source, and more.

For examples of new types of tasks DBAs may take on in the autonomous database world, read Jeff Erickson's latest column, "Hey, Data Experts: Voice Assistants Are Calling Your Name."

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/STUDIO AT GETTY IMAGES FOR ORACLE**

NEXT STEPS

WATCH *Think Autonomous*.

LEARN more about Oracle Autonomous Database Cloud.

LEARN more about Oracle Cloud Platform autonomous services.

TRY Oracle Autonomous Transaction Processing Cloud.

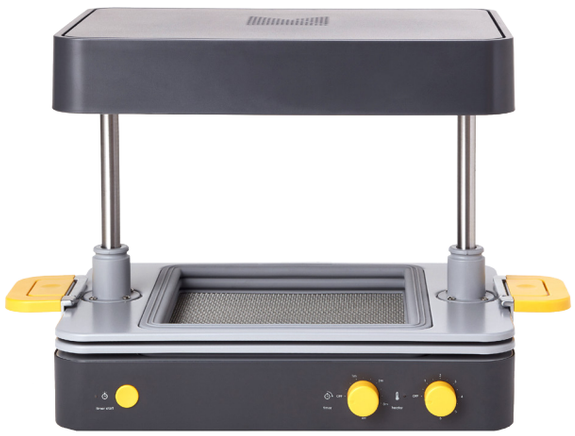
REGISTER for Oracle OpenWorld 2018.

3D, AI, Blockchain, and More

Gadgets, apps, and attitudes for the technophile

Mayku FormBox

Have vacuum cleaner, start a business. Use almost any material or item to create your master 3D form. Place the master on the Mayku FormBox bed and cover with a thermoplastic sheet, hook up your vacuum cleaner, and you've got a mold in minutes, with no additional software or digital model manipulation. Fill your molds, choosing from an endless choice of materials, to start your production line from home. Custom chocolates, anyone? US$699. Mayku



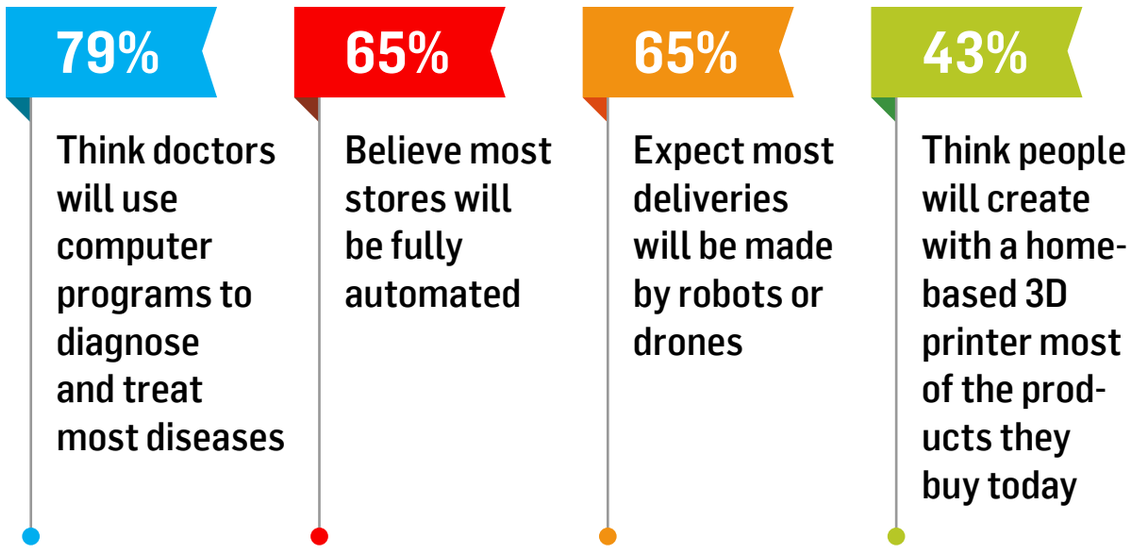
Digitsole Smartshoe

Digitsole already sells "connected insoles" for cyclers and runners that use advanced technology—including AI—to track activity and technique, monitor fatigue level, and keep your feet warm. Now the company is developing the Digitsole Smartshoe, which it calls "the first autotightening, interactive, heating, and shock-absorbing sneaker." The Smartshoe, scheduled to be commercially available soon, integrates street style with wearable technology and offers individual coaching based on your movements. Smartshoe price currently listed at US$599. Digitsole

PHOTOGRAPHS COURTESY OF MAYKU; DIGITSOLE

Everyday Automation—Ready or Not?

Emerging automation technologies such as driverless cars got mixed reviews in a recent Pew Research Center survey of 4,135 "everyday" American adults. Although a majority expressed concerns about a future in which robots and computers can perform many jobs humans do now, only 30% of workers think it's at least somewhat likely their own jobs will be done mostly by those technologies during their lifetimes, and a majority of those with college-level education said technology had increased opportunities and made their jobs more interesting. Wary or not, when asked about four specific scenarios over the next 20 years, here's how the respondents weighed in:



Source: "Automation in Everyday Life," a Pew Research Center survey

DO YOU SPEAK TECH? QUIZ YOURSELF!

1. A self-driving database is

- **A.** The collection of data embedded in an autonomous car that allows it to make driverless decisions
- **B.** A database that performs all routine database maintenance tasks without human intervention
- **C.** A database engineered to propel itself toward continual self-improvement—also referred to as a Kaizenbase

2. Strong AI can be defined as

- **A.** Artificial intelligence embedded in athletic equipment
- **B.** An artificial intelligence system immune to hackers
- **C.** The ability of a machine to perform all tasks as well as or better than humans can

3. In addition to her movie career, Hedy Lamarr is famous for

- **A.** Developing the first-known instance of video-enabled phone conversations, created so she could avoid in-person interviews
- **B.** Inventing spread-spectrum technology, a method of manipulating radio frequencies that formed the backbone for today's wireless communications technologies
- **C.** Funding Austria's first STEM-focused school for girls in 1997

Answers: 1. B; 2. C; 3. B

APPS FOR THE TWENTY-FIRST CENTURY

Get intelligent transcription, blockchain your promises, and find a chatbot friend.

Otter Voice Notes

Put down the steno pad, and turn on this smartphone app. Otter Voice Notes creates notes that combine audio, transcription, speaker identification, inline photos, and key phrases. Simply hit the record button during an English-language conversation or meeting, and Otter will produce a usable transcript a few minutes later. (You can also import existing recordings.) The app integrates features such as AI-enabled cross-conversation speaker identification, search tools, and in-app transcript editing. Free (Android, iOS)

Handshake on the Blockchain

Blockchain that promise! The Handshake app transcribes commitment into code, creating an unchangeable time-stamped blockchain record of who promised what to whom. You can choose from a selection of common templates or create your own agreement, and the app transcribes your handshakes into a cryptographically signed snapshot, viewable in-app. So whether it's an officemate's promise to buy the next pizza or a bet on the World Cup, the deal is on record, private, and secure. Free (Android, iOS)

Replika

Replika is an AI chatbot with a personal twist. Think of it as an interactive journal—or a nonjudgmental friend you can carry in your pocket. Replika asks questions about you and learns based on your interactions, taking on aspects of your personality over time. As it becomes more "like" you, it can even give you some insights into how you come across to others. Vent, explore your feelings, record your thoughts, and more with an always-on digital companion. Free (Android, iOS)

Oracle Executive Vice President Amit Zavery describes autonomous capabilities as “core to the new Oracle Cloud Platform services released on the modern Oracle Cloud Infrastructure.”

Autonomous Is Easy

Self-driving, self-securing, and self-repairing services are engineered to simplify.

BY TOM HAUNERT

As autonomous car technologies, projects, and testing evolve, the idea of autonomous technology gets a bit easier to understand. That's not to say the tech that's making autonomous cars possible is easy to develop and deploy, but one clear goal for autonomous tech is about making things easier for people.

Oracle Magazine sat down with Oracle Executive Vice President Amit Zavery to talk about autonomous technology in the enterprise, how Oracle Cloud Platform autonomous services are making enterprise technology easier to work with, and more.

Oracle Magazine: Many consumers may understand the concept of self-driving or autonomous cars, but what are autonomous cloud services? What makes Oracle Cloud Platform services autonomous?

Zavery: The way we look at autonomous services first is, no doubt, the idea of self-driving. But beyond that, we broaden the autonomous concept from driving to securing and repairing. All Oracle Cloud Platform autonomous services are self-driving, self-securing, and self-repairing.

All Oracle Cloud Platform autonomous services also use machine learning to continuously gather and learn from the development process experience and apply this learning across services. This ensures that all Oracle Cloud Platform autonomous services run efficiently, accelerate customer innovation, and reduce risk and cost for our customers.

Oracle Magazine: The first autonomous service Oracle announced was Oracle Autonomous Data Warehouse Cloud, during Oracle OpenWorld 2017. That service and several other autonomous PaaS services are now live. What is common across these services?

Zavery: Autonomous capabilities are core to the new Oracle Cloud Platform services released on the modern Oracle Cloud Infrastructure. Since Oracle OpenWorld 2017, we have released several new services, including Oracle Autonomous Data Warehouse Cloud, Oracle Autonomous Transaction Processing Cloud, Oracle Autonomous Blockchain Cloud Service, Oracle Autonomous Visual Builder Cloud Service, Oracle Autonomous Mobile Cloud, Oracle Autonomous Data Integration Platform Cloud, and Oracle Autonomous API Platform Cloud Service. What is common across all these services is that we have eliminated the need

for enterprises to assign valuable resources to operational functions such as tuning, patching, doing backups, and reducing the risk from security vulnerabilities across all services.

In addition, each service includes unique autonomous features that add value for customers. Oracle Autonomous Analytics Cloud, for example, reveals hidden patterns and performance drivers through predictive insights and automatic natural-language explanations powered by machine learning. Oracle Autonomous Integration Cloud speeds up integrations in the complex process of mapping attributes of objects across two different applications by using crowd-sourced data of all executed integrations and machine learning to deliver visual recommendations of how to connect those objects. Oracle Autonomous Mobile Cloud continuously increases the accu-

"All Oracle Cloud Platform autonomous services also use machine learning to continuously gather and learn from the development process experience and apply this learning across services," says Oracle Executive Vice President Amit Zavery.

racy of chatbot conversations with end users by updating the NLP [natural language processing] AI models used in the chatbots with Oracle Digital Assistant Cloud.

> "All Oracle Cloud Platform autonomous services are self-driving, self-securing, and self-repairing."

We are committed to supporting the common autonomous principles with new and updated autonomous features in our services.

Oracle Magazine: What are users and customers looking for in Oracle Cloud Platform autonomous services? What are the common themes?

Zavery: Customers want to reduce their operations costs, get the best-possible value from the cloud, and focus on innovation. IT budgets are flat or shrinking, and enterprises are dealing with the dilemma of creating innovation versus managing current operations. Oracle Cloud Platform autonomous services are delivering what our customers want: self-driving cloud services that give them relief from managing operations such as provisioning, backup, patching, and recovery, which gives them the bandwidth to focus on their own innovation.

Oracle autonomous cloud services also give customers the assurance that any vulnerabilities will be detected and action will be taken faster than it would be for their staff to detect and repair them. Customers will begin to expect this level of intelligence with cloud services, and as Oracle adds more self-driving features that learn from each customer usage and across customers, they will benefit beyond operational automation and start discovering, creating, and automating new business functions.

PHOTOGRAPHY BY **BOB ADLER/STUDIO AT GETTY IMAGES FOR ORACLE**

NEXT STEPS

LEARN more about Oracle Cloud Platform autonomous services.

TRY Oracle Cloud Platform autonomous services.

Mike Hichwa, vice president of software development at Oracle says, “cloud makes full-stack development easier than before. So do autonomous database services.”

Development Opens Up

Think full-stack, cloud native, open source, and Oracle. **BY TOM HAUNERT**

Do you remember two-tier development? Three-tier development? N-tier development? Dividing development into tiers formalizes code and expert separation on development projects. But what if that application code separation did not require different experts for each tier, each datasource, or each new deployment target?

Oracle Magazine sat down with Mike Hichwa, vice president of software development at Oracle, to talk about the concept of full-stack development, the skills of a full-stack developer, open source development for Oracle Database, the effect of autonomous databases on development, and more.

Oracle Magazine: Traditional application development often splits dev responsibilities between an app team and a database team, but what is full-stack development? And what is a full-stack developer?

Hichwa: Historically there has been a divide between front-end and back-end application development. You might refer to that divide as client/server, where one team might have specific skills for client development and build out the client application while another team has server skills and is responsible for the server side. And that split made a lot of sense, because, for example, you might have five different applications that connect to the same database or back-end system of record, and you might have one team that would maintain all of the application data on the server, make sure that all the server access from all the client applications was secure, and so on.

But as is always the case, whenever you have split responsibility you can have some inefficiencies.

For example, an application developer could ask a DBA to add an extra column to a database table, and that process could involve logging a ticket, getting approvals, and still more time-consuming steps. Split responsibilities don't always create the fastest solutions.

And today, people want everything on the web, they want everything to be responsive, and they want mobile interfaces. And of course, the people who want these new systems want them yesterday. There is a lot of pressure on application developers today to get the user experience and user interface right and do it quickly. Frequently, when today's developers are building out new systems, it's just easier

to control both the front end and the back end. When the same developer is responsible for the application front end and back end, that's a full-stack developer—a developer jack-of-all-trades.

This kind of general expertise across the front end and back end would have been difficult in the past, but now in the cloud, a lot of things are much easier than they used to be. For example, getting, installing, configuring, and provisioning a new on-premises server used to require a purchase order, approvals, delivery, installation, and a lot more. But today, I can simply go to cloud.oracle.com or my favorite cloud infrastructure provider and within minutes get access to infrastructure components and an operating system on a server.

What the cloud has done for the traditional back-end responsibilities means that today's full-stack cloud developer can do in minutes the things that used to take days, weeks, and months. That opportunity has also contributed to the rise of the full-stack developer.

Oracle Magazine: Organizations are successfully using open source, such as JavaScript, Node.js, and other dev frameworks. How do open source frameworks work with Oracle Database?

Hichwa: It's important for Oracle Database to support any and all clients, and we make sure that we support everything that's popular. Oracle Database doesn't have a favorite client. We want to support every client as best we can, so we have database drivers for every popular language, including Node.js.

You can also get drivers for PHP, Python, C, C++, PHP, Argo—you name it. You can go to the database application development page on oracle.com, and you'll find pointers to all the drivers and all the links to gain access to each driver.

The popularity of open source frameworks and languages is also great for developer productivity. Members of the open source developer community who are from different companies and regions and who are working on different types of projects contribute lots of code examples on places such as GitHub, their blogs, and other locations—they share. And then as those developers gain experience, they often publish their own best practices or frameworks that allow other developers to do things more easily. Developers don't need to build stuff from scratch; they can pick up and start

with something that someone else has already done.

Oracle Magazine: Oracle is now offering autonomous database services. How do autonomous database services change the developer strategy or development operations for full-stack development?

Hichwa: I mentioned earlier that cloud makes full-stack development easier than before. So do autonomous database services.

With the autonomous database comes the idea of near-instant access to the database technology—within seconds. A few clicks, and you can create an autonomous database that's fully managed and includes all of the high-availability, disaster recovery, online patching, and elasticity properties of Oracle Database cloud services.

And if your organization's full-stack developers are not database experts or are simply tight on time,

"Oracle Database doesn't have a favorite client. We want to support every client as best we can, so we have database drivers for every popular language, including Node.js," says Mike Hichwa, vice president of software development at Oracle.

an autonomous database can automatically tune the database—thus taking a highly specialized and labor-intensive task off the table. But over time, production databases grow and may be used in unexpected ways. An autonomous database, however, tunes itself as it grows based on actual usage patterns.

Oracle Magazine: Oracle offers different drivers, services, and frameworks for developing Oracle Database apps. What are the key technologies, and whom are they for?

Hichwa: For core data access, there are drivers, which I mentioned earlier. There's a JDBC driver for Java, a Node.js driver for Node.js, a PHP driver for PHP, and so on.

RESTful [Representational State Transfer] data access is growing in popularity and is available in every mainstream language. Oracle provides Oracle REST Data Services, which lets a developer write SQL and publish that SQL as a REST API. It will automatically map input values in REST GET or POST operations, bind them to the SQL statement, and return the results in JSON [JavaScript Object Notation]. Oracle REST Data Services is a no-cost feature of Oracle Database and can dramatically simplify the task of creating RESTful APIs.

And then there are higher-level tools. One of the most popular tools in the Oracle development community today is Oracle APEX [Oracle Application Express]. Oracle APEX enables developers who are comfortable with SQL to create modern, responsive web applications quickly and with great fidelity.

Editor's note: Refer to the Oracle Database Application Development page for information about and links to database development languages, drivers, and tools.

PHOTOGRAPHY BY **JOSHUA ROBERTS/ STUDIO AT GETTY IMAGES FOR ORACLE**

NEXT STEPS

DOWNLOAD Oracle Database Application Development drivers and tools.

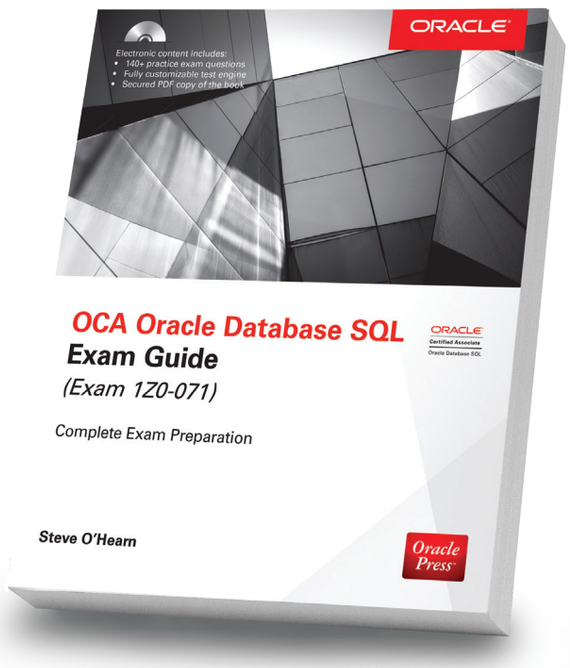
LEARN more about Oracle Cloud Platform autonomous services.

TRY Oracle Cloud Platform.

Oracle Press™

Your Destination for Oracle and Java Expertise

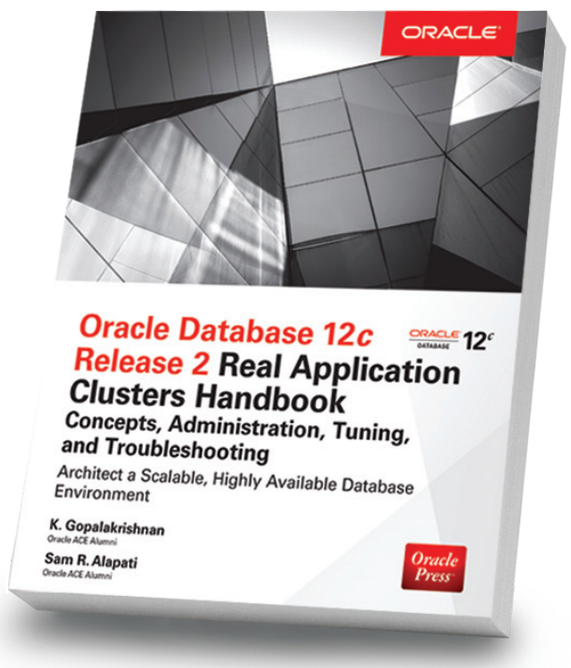
Written by leading technology professionals, Oracle Press books offer the most definitive, complete, and up-to-date coverage of Oracle products and technologies available.



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

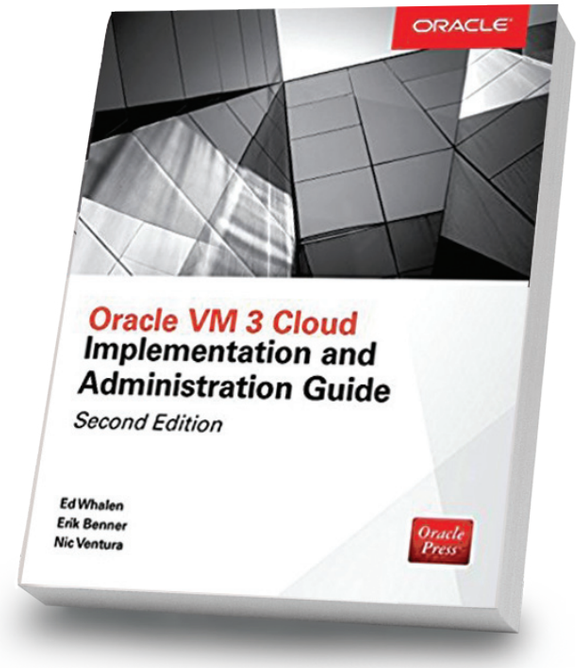
K. Gopalakrishnan, Sam R. Alapati

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

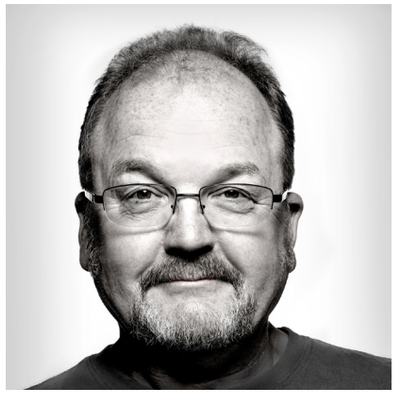
Ed Whalen, Erik Benner, Nic Ventura

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

Available in print and ebook formats.

Oracle Press™

www.OraclePressBooks.com • @OraclePress

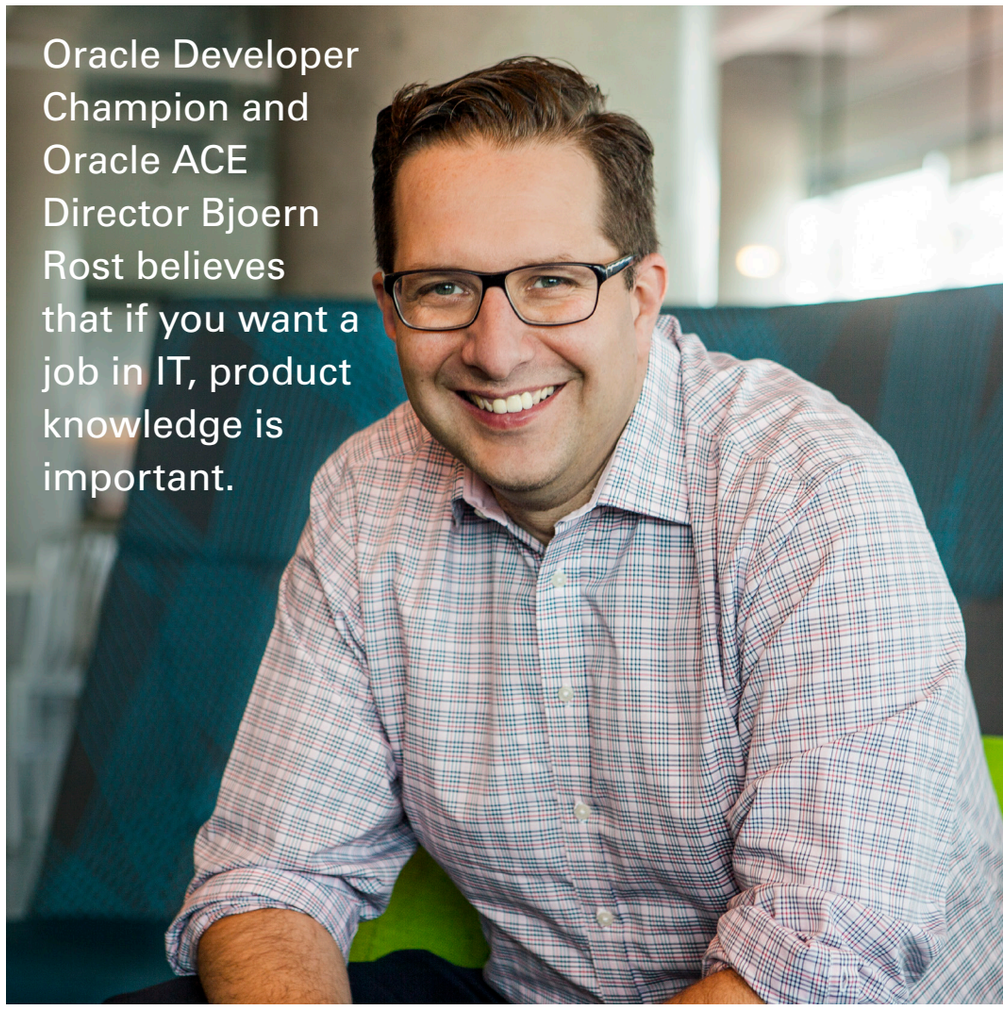


Bring Your Own Piñata

Oracle Developer Champion Bjoern Rost describes his IT journey so far and his interest in what's next.

By **Bob Rhubart**

Don't be alarmed if you notice a man wandering around Oracle OpenWorld with a piñata under his arm. The piñata's name is Lumpy, and one assumes that Lumpy is filled with sugary treats, as is often the case with piñatas. The man carrying Lumpy is Oracle Developer Champion and Oracle ACE Director Bjoern Rost, a data and analytics specialist with Pythian. If you've seen one of Rost's presentations at Oracle OpenWorld, Oracle Code, ODTUG Kscope, or any of an astonishing number of other tech conferences around the globe, you know that *he* is filled not with candy but with an impressive level of expertise and insight on data modeling, performance tuning, high-availability



Oracle Developer Champion and Oracle ACE Director Bjoern Rost believes that if you want a job in IT, product knowledge is important.

RECOGNIZE

The Oracle Developer Champion program recognizes modern expert developers who blog; write articles; and present on topics such as containers, microservices, SQL, NoSQL, open source technologies, machine learning, and chatbots. Learn more and follow the Oracle Developer Champions.

design, and just about anything else that relates to Oracle Database.

The accumulation of that expertise began when Rost was in grade school, when his father brought home a Commodore 64. Rost first used the machine for gaming, but that soon gave way to an interest in writing BASIC. Thanks to a failed floppy drive on the Commodore, "if I wanted to do anything, I had to program it," says Rost.

As a teenager in the late '90s, Rost began working with a group of friends to network computers together to play games and organize LAN parties. Groups of 20 players soon grew to groups of 2,000 or more. "That's when I got into infrastructure and Linux," Rost says.

Rost applied his skills to build a website on which people attending those LAN gaming parties could chat or post messages in a guest book. That experience led to his first job. "They needed somebody to do Linux, because nobody knew Linux," Rost explains.

Despite his interest in computers and programming, Rost entered a general engineering program in college, but after a couple of semesters he specialized in electrical engineering and computer engineering. "Much more suited to my interests," he explains.

For his bachelor's thesis, Rost built and programmed an inexpensive but effective signal processing device for tracing electrical circuits in aircraft. He built the circuit board and wrote the code for the device's field programmable gate array.

Two decades and several DBA, system administrator, and consultant jobs later, Rost continues to evolve. "I'm kind of moving from being an Oracle Database specialist to more of a generalist in building data platforms," he says. His work with Pythian is focused on Oracle Database, "but I'm also moving on to do cloud-native data platforms, which means working with cloud-native data warehouses, building data pipelines, interfacing with many different source

data systems, transforming them, and doing all of that in as serverless a way as possible."

As the IT ecosystem evolves, so do Rost's interests. "I find it hard to do something that I'm not passionate about, and I've been passionate about Oracle Database for a long time," he explains. "What I find fascinating now is real-time streaming analytics—things like Kafka, KSQL, Spark Streaming, and Apache Beam. You have huge volumes of data coming, and you want to make sense of them as fast as possible."

As his expertise grows within the ever-expanding IT universe, Rost offers this career advice: "If you want to have a job in IT, I think it's important to have product knowledge. It's hard to get hired if you say, 'I know a lot about relational databases.' That might be true, and you might have studied relational databases, and you might have studied a lot of theory, and you might even know SQL." That knowledge and experience has great value, Rost says, but knowledge and experience with specific products is key.

"For getting a job it is still important to know products," Rost explains. "Get exposure to them, and do as many internships and co-op things and projects as you can, because that's really valuable, real-life, real-world experience." But you'll have to get your own piñata.

Oracle Architect Community Manager Bob Rhubart is the host-engineer/producer of the Oracle Developer Podcast *series and produces the* 2 Minute Tech Tip *video series.*

PHOTOGRAPHY BY **JORGE CAMAROTTI/ STUDIO AT GETTY IMAGES FOR ORACLE**

NEXT STEPS

LEARN more about Oracle Developer Champions.

By **Alexandra Weber Morales**

Fit, Fresh, Flow

Oracle Developer Champion Lonneke Dikmans handles code and coders with poise.

The key to productivity? Discover what works for you. That's why perusing the routines of successful developers can inspire new routines and motivation. Lonneke Dikmans, for example, is an Oracle Developer Champion and head of the Center of Excellence at eProseed, an Oracle Platinum Partner headquartered in Luxembourg. Bitten by the programming bug at age 12, Dikmans now spends hours managing technical teams every day—but she still makes a point of finding flow when she programs.

Oracle Developer Champion Lonneke Dikmans uses the outdoors to help manage distractions.

CHOOSE LIGHTWEIGHT TOOLS

Dikmans measures her productivity with a to-do list that includes both coding and

noncoding activities: "Each day, I pick a number of those items for that week. If I finish all of them, I feel productive." To help tick those items off her list, Dikmans gravitates toward lightweight solutions, be they editors, test frameworks, or even analog devices.

For shared coding project management tasks, Dikmans finds that Jira fits well. For her personal to-dos, she has moved from her beloved Wunderlist to Notepad++, which is also quite convenient, Dikmans notes, for occasional JavaScript edits.

"I don't think I have a favorite development tool, but what I do like is if it's really lightweight, responsive, and fast," she says. She recently moved from Oracle JDeveloper to NetBeans IDE for Java. Postman is her JavaScript go-to for testing REST APIs, and SoapUI is her choice for testing web services. Her company was one of the first references for Oracle JavaScript Extension Toolkit (Oracle JET), which bundles JavaScript libraries and offers productive, low-code interfaces.

"I have to admit, usually the things I'm into at the moment will get done. It has to be some new concept for me as a person, such as new solutions for integrations. Getting into JavaScript and Node.js is something new for me," she says, noting a recent project for Transdev, a French transportation company. Oracle Mobile Cloud Service and Oracle API Platform Cloud Service were the foundations for a JavaScript application that integrates data from train schedules, car sharing, bikes, buses, and more. "We were the first partner to go live with Oracle API Platform Cloud Service," Dikmans says.

LIMIT DISTRACTIONS

"I realized recently that I could not concentrate for more than 10 minutes anymore," Dikmans says. "I have to make a very conscious decision that I'm going to sit here for an hour and

"I've learned to accept that sometimes you won't find it, but sometimes you need a couple of failed attempts to get into that flow."

—Lonneke Dikmans, Oracle Developer Champion

not check my phone." Of course, being easily distracted is an affliction of our time. As a technical manager with multiple concurrent projects and teams in the Netherlands, the UK, Portugal, Luxembourg, and Romania, among others, Dikmans communicates via Slack, Skype, Teams, email, phone, text, and WhatsApp. "I've switched all the notifications off, because at one point, my messages were going off on my phone 24 hours a day," she says. "Currently I am a member in at least five active Slack workspaces—and this is still growing. My biggest distraction is all the communication."

That's why it's also important to get outside, she believes. "I try to have everyone in the office take a proper break for lunch," she says. "After half an hour outside, you're much more productive." Walking meetings also go toward achieving one of her daily exercise goals. The minimalist Nokia smartwatch with analog dials she wears makes it easy to track daily progress toward 10,000 steps.

When she isn't outside, she loves whiteboards. "Whenever I'm explaining something, I want a whiteboard or flip board," Dikmans says. "But I hate it when those pens run out of ink. When you store it, the tip should be pointing down."

SEEK FLOW EVERY DAY

Despite frequent travel for speaking engagements and meetings, Dikmans is a firm believer in flow. Here's how she finds it: first, she chooses times of day where she won't be interrupted, and then she puts on headphones and listens to music. "I like Queens of the Stone Age a lot. I have a Diet Coke and completely block out everything else. When I really get into it, I forget about everything and don't even feel the time pass."

No matter how you do it, Dikmans believes it's critical to try to find flow every day. "Sometimes I'll try to start something for three days and it won't

work, then the fourth day I find my flow. It has to do with interruptions, but also with my own ability to focus," she says. "I've learned to accept that sometimes you won't find it, but sometimes you need a couple of failed attempts to get into that flow."

Alexandra Weber Morales is Oracle director of developer content.

PHOTOGRAPHY BY **MICHEL PORRO/STUDIO AT GETTY IMAGES FOR ORACLE**

NEXT STEPS

LIVE for the Code (Oracle Code).

LEARN more about Oracle Developer Champions.

Hey, Data Experts: Voice Assistants Are Calling Your Name

Data professionals connected back-office applications to voice assistants. Here's what they learned.

By **Jeff Erickson**

If you can ask Apple Siri or Amazon Alexa to search databases for restaurants and obscure bands, why can't you ask them to search or update the databases of your enterprise resource planning (ERP) or human resources (HR) applications?

This question, or something like it, crossed the minds of Jorge Rimblas and Christoph Ruepprich—both of whom are database developers and Oracle ACEs. But as data professionals, the question wasn't academic. They each took the initiative to answer the question and learned lessons along the way.

"It's a great time to be a database developer," says Rimblas, who has worked with Oracle Database since 1995. "But you have to keep learning," he says. Rimblas spent his early years as a database developer learning to write efficient SQL queries and wrapping his mind around large data models. He's now polishing his abilities with REST services, plumbing JavaScript libraries, and sharing projects on GitHub.

Rimblas wanted to know if he could make interacting with back-office applications easier, or even fun, by using a

BOT WAIT

Along with the growing interest in voice interfaces, there's a demand in the enterprise for both voice- and text-based bots. Oracle Intelligent Bots in Oracle Autonomous Mobile Cloud, for example, embeds intelligent bots in voice assistants such as Alexa and Siri and surfaces bots in messaging platforms such as Facebook Messenger and WeChat.

voice assistant. He learned that he can. Rimblas created a proof of concept for his company by connecting Alexa to its timekeeping and billing application, which had been built internally using Oracle Application Express (Oracle APEX).

"I can just ask Alexa what client jobs I'm working on and she'll tell me, or I can ask her to add hours to a job and she'll do it," he reports. "If I tell her I'm done for the day, she'll give a happy reply, or if I tell her to add hours to my vacation time she'll update the app and say, 'Yay; enjoy your vacation!'"

"You're not touching a keyboard, but you're finding information and updating information in your work application," he says. Now he can focus on client work and dispense with timekeeping and billing tasks by speaking a few words to a voice assistant. "It's a really nice way to work."

For Rimblas, the process of connecting Alexa to his application meant using Oracle APEX and Oracle REST Data Services to retrieve and present data to Alexa's developer framework. "I did it all through the browser with Oracle APEX on one side and Alexa's testing framework on the other," he says. "I didn't even need an Alexa device," he adds.

Rimblas noted that a developer can also use AWS Lambda along with Node.js or JavaScript to get to the database, "but Oracle REST Data Services is directly connected to the database, so I don't have to make another hop to get to the data." Knowing SQL and Oracle Database made the Alexa proof of concept work go quickly because Rimblas could efficiently manage the database side.

A GAZILLION THINGS YOU CAN DO WITH VOICE ASSISTANTS

Christoph Ruepprich feels that working with voice assistants is a natural extension of the DBA and database development work he's done most of his career and shares on his popular blog. "When Alexa came along, I just naturally saw

that this is yet another way to access the data," he recalls.

To work with voice assistants, Ruepprich simply combined his knowledge of Node.js (which he had learned at an ODTUG Kscope meeting and has been using ever since) with the SQL, Oracle APEX, and REST services he uses for other development projects. "It's a perfect fit," he says of using a voice assistant. "I can simply access my database and ask it questions."

His prototypes have been a success, he says, and now he's looking for a bigger project to apply his knowledge to. "There are a gazillion things you can do with a voice assistant," he says. "It can call up a screen, read back some results, or trigger actions in the application." It is, he says, a perfect approach for situations where people have their hands busy. "If someone is working for an airline and they're working on an airplane and they need a part, they can say, 'Hey Alexa, do we have this part in stock? Can I have it brought out to me on the shop floor?'" he says. "Or show me the schematics for this or that."

"You always want to find the best way of getting the data to the user," Ruepprich concludes. He credits his knowledge of SQL and PL/SQL with helping him quickly build effective voice apps, "because if you don't know how to work with the database, that's where the bottlenecks happen," he says. "If you don't write efficient SQL, your great app isn't going to work as well as you want."

Jeff Erickson is editor at large for Oracle Content Central.

NEXT STEPS

LEARN more about Oracle Application Express.

LEARN more about Oracle Intelligent Bots.

TRY Oracle Intelligent Bots.

BY BLAIR CAMPBELL

Follow Your Bliss

Three peers on tools as best friends, career paths to happiness, and the joys of specializing

Carina Mendes

São Paulo, Brazil

ORACLE ACE

Company/URL: GAVB IT Solutions

Job title: Data scientist

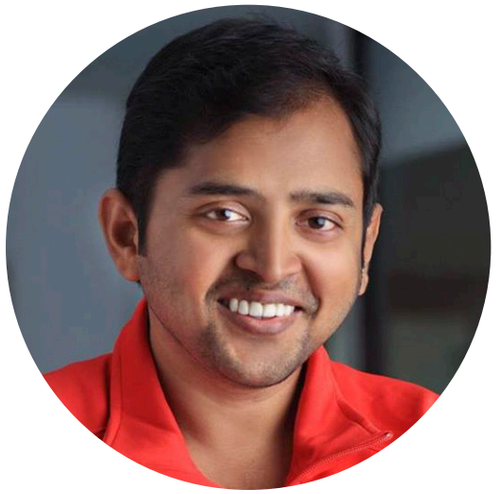
Oracle credentials: Oracle Business Intelligence 11g Certified Implementation Specialist

Length of time using Oracle products: Eight years

What are your favorite tools on the job? I love working with the data, and I often say that SQL and Oracle Data Visualization are my best friends. With SQL, I can select, aggregate, and work with the data exactly the way I need to. With Oracle Data Visualization, I make the information intelligently available so our customers have new insights about their data.

Which new features in Oracle Cloud Platform services are you currently finding most valuable? Traditional BI techniques are no longer sufficient. So at my company, we're adapting our methodologies through the new modeling and machine learning functionalities in Oracle Analytics Cloud. Now we can see the future, because Oracle Analytics Cloud allows you to create predictive data models where, from the trend of standard behavior in the data, it's possible to estimate what event will occur and when.

How about new features in other Oracle business analytics services? The latest data preparation features in Oracle Data Visualization Cloud Service make it much easier to work with different datasources. And in the mobile version of Oracle Day by Day, the ability to do a voice search for indicators is a great differentiator.



Nassyam Basha
Riyadh, Kingdom of Saudi Arabia

ORACLE ACE Director

Company/URL: eProseed
Job title: Database expert
Oracle credentials: Oracle Certified Master (Oracle Database 11g), Oracle Exadata Database Machine Certified Implementation Specialist
Length of time using Oracle products: 11 years

How did you get started in IT? By the time I finished university, there was already a boom in the Indian IT sector. I decided to pursue a career in computer science, and by then I had a sense of which branch of IT had the best job prospects. This led to an initial focus on software development, but I later realized that planning, coordinating, and analyzing data is an area of specialization that makes me happy. So, I decided to become a database administrator.

What's your go-to Oracle reference book? I always turn to docs.oracle.com first, but in the course of my career the two best books I've read are *Expert Oracle Database Architecture* by Thomas Kyte [Apress, 2014] and *Troubleshooting Oracle Performance* by Christian Antognini [Apress, 2008]. Apart from these, I often consult the My Oracle Support knowledgebase and various blogs.

What green practices do you use in your DBA work? I think that my company is taking good steps toward reducing our environmental impact, including our use of virtual servers, utilizing cloud infrastructure instead of having on-premises physical machines, consolidating database and application servers, consolidating networks, and ensuring proper disposal of electronic waste.

Mónica Godoy Millán
Santiago de Cali, Colombia

ORACLE ACE

Company/URL: Self-employed

Job title: Oracle Applications consultant

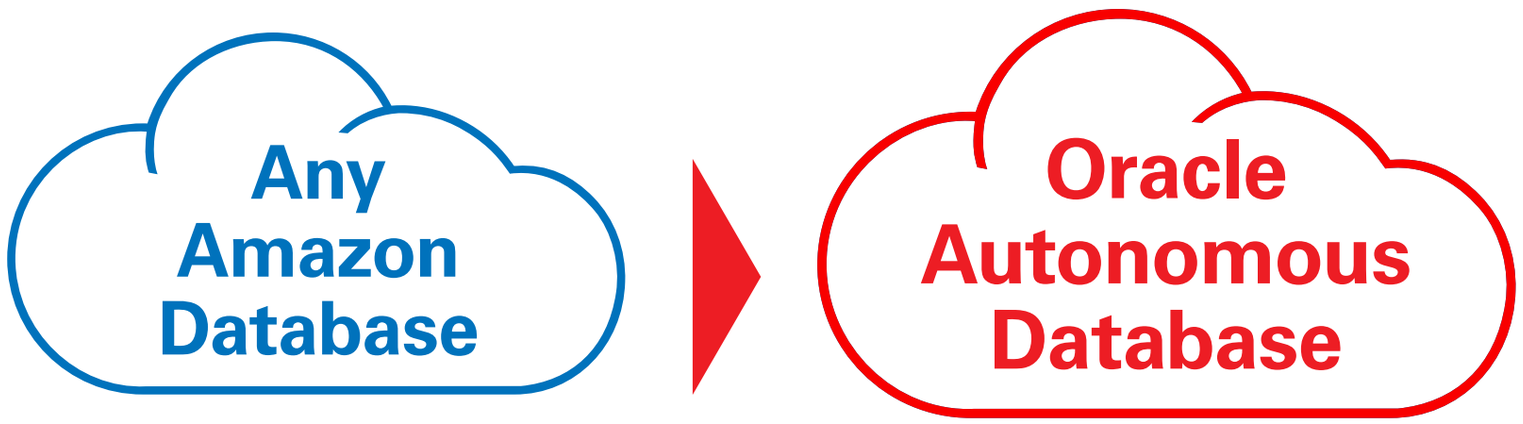
Length of time using Oracle products: 12 years

What advice do you have about getting into application development? When you're first starting out, it's OK to try out several different tools, but later on—say, five years down the road—it's best to focus on one or two. When you don't specialize, your knowledge can be pretty limited, and you're not going to feel confident about the decisions you have to make. Your knowledge should be deep enough so that you can answer a customer's questions, define a development architecture, and find the best solution to a problem. Those things are very important in gaining the customer's trust. Of course in my view, the best tool on the market right now is Oracle Application Express—so one good option is to begin specializing in that tool right away.

What's the most common cause you see when IT projects go wrong? I believe that it's when IT just does what the customer asks for, without further inquiry. Oftentimes if you ask in-depth questions about their needs, you'll discover that they don't need what they're asking for or that there are better solutions to implement than the one they may have selected.

What would you like to see Oracle, as a company, do more of? I would like for Oracle to get more involved with communities in Latin America, in order to encourage new generations to pursue IT careers. If you become acquainted with Oracle technologies at an early age, you're far more likely to develop a love of technology and follow a path to success.

Cut Your Amazon Bill in Half



Easy to Move—Guaranteed Savings

There's the cloud… and there's the Oracle Autonomous Cloud.

#thinkautonomous

ORACLE®

Autonomous in Action

Choose self-managing, self-repairing, and self-securing Oracle Cloud Platform autonomous services to focus on innovation—not on administration.

BY CHRIS MURPHY AND JEFF ERICKSON

Your job is about to get better—autonomously. Since March 2018, Oracle has announced the availability of multiple Oracle Cloud Platform autonomous services, including those for database, development, mobile, business analytics, integration, blockchain, and more. With this flood of new autonomous services, how will organizations prepare to administer them? To answer that question, consider the administrator requirements for Oracle Autonomous Database Cloud described by Oracle Executive Chairman and CTO Larry Ellison.

"You know what you have to learn? Nothing. It's the same thing with a self-driving car," Ellison said. "When you get in that self-driving car, how hard is it to drive? What do you need to do to drive that self-driving car? Well, you have to tell it where you want to go."

Like a self-driving car, Oracle Autonomous Database Cloud now automatically takes care of a maze of complicated things that a person used to do—and still must do for other databases. That includes setting up the database; optimizing its performance; monitoring for security breaches; patching it; and deciding how much compute, storage, and network capacity is needed.

Ellison took the stage on August 7 at an event at Oracle headquarters to announce the availability of Oracle Autonomous Transaction Processing Cloud, the second of Oracle's cloud-based autonomous databases. The first, Oracle Autonomous Data Warehouse Cloud, is optimized for analytics. Oracle Autonomous Transaction Processing Cloud is optimized for a complex set of high-performance transactions as well as for mixed workloads, so it can handle batch processing, reporting, IoT data, and much more.

"Between these two systems…Oracle Autonomous Database Cloud now handles all of your workloads," Ellison said. "All of them."

That autonomous capability enables lower operating costs, improved security and reliability, and lower IT labor costs, and it gives database administrators more time to focus on new ideas instead of tuning databases and allocating storage.

Developers, Ellison said, now can get all the features and perfor-

START
your free trial of Oracle Autonomous Transaction Processing Cloud.

mance of Oracle Database in a service that's easier to set up and use than even lesser-featured databases.

"Oracle Database was very complicated to learn," Ellison said. "That's all gone now. Now we're as simple to use as the simplest databases on the planet."

Autonomous for Innovation

Perhaps the member of the IT team who benefits the most from autonomous databases is the DBA, who can turn attention from database maintenance toward improving application performance and leading data-driven innovations.

"There are more in-house developers at our enterprise customers than ever before, and they all need access to data and database services," says Maria Colgan, master product manager for Oracle Database. If DBAs are spending less time provisioning, patching, and tuning databases, they can engage developers and help them understand what the database can do. "If developers can do something their application needs inside the database and just get results back, they can save themselves a whole lot of effort and make their application more efficient," Colgan says. Likewise, a DBA's knowledge of datasources, formats, and policies is in high demand by data scientists and business analysts. With less time spent managing the database and more time helping the company use data to innovate, "DBAs can become even more valuable partners for developers and business leaders," Colgan adds.

Automatic Patching Means Better Data Protection

"Most people who've suffered data theft suffer data theft long after the vulnerability was known about and a patch was available to fix the vulnerability," Ellison said at the August 7 event. "That makes it all the more embarrassing and costly." With Oracle Autonomous Database Cloud, "there is no time delay," he said.

With no humans maintaining the database, machine learning takes over, applying security patches in the background while the database remains online. IT ops and security pros no longer must weigh how big a risk a vulnerability poses against the cost and inconvenience of taking a system down. Even more important, there's no risk of just overlooking a database that needs patching.

> **“Oracle Database was very complicated to learn. That’s all gone now. Now we’re as simple to use as the simplest databases on the planet.”**
>
> —**Larry Ellison**, Oracle Executive Chairman and CTO

“People have thousands or tens of thousands of databases, believe it or not, at large companies,” Ellison said. “It’s very difficult just to keep track of all of them, take them down, and make sure they’re patched. When this is a manual process, human beings make errors. When it’s a manual process, it’s a riskier process. Now it’s completely automated, and it’s immediate.”

Automatic patching is also part of why Oracle can promise 99.995% uptime—with no exclusions for scheduled downtime for tasks such as patching and upgrading—which translates to maximum downtime of less than two and a half minutes a month.

Autonomous in Common

The technology in Oracle Autonomous Transaction Processing Cloud, Oracle Autonomous Data Warehouse Cloud, and all Oracle Cloud Platform autonomous services runs on the modern Oracle Cloud Infrastructure and supports the core autonomous ideas of self-driving, self-securing, and self-repairing. That technology includes artificial intelligence and machine learning in different services and features, including the decision engine in Oracle Autonomous Transaction Processing Cloud and Oracle Autonomous Data Warehouse Cloud.

“Oracle has released several autonomous services, including Oracle Autonomous Transaction Processing Cloud, Oracle Autonomous Data Warehouse Cloud, Oracle Autonomous Blockchain Cloud Service, Oracle Autonomous Visual Builder Cloud Service, Oracle Autonomous Mobile Cloud, Oracle Autonomous Data Integration Platform Cloud, and Oracle Autonomous API Platform Cloud

Service," says Oracle Executive Vice President Amit Zavery. "What is common across all these services is that we have eliminated the need for enterprises to assign valuable resources to operational functions such as tuning, patching, and backups as well as reducing the risk from security vulnerabilities across all services."

Chris Murphy is director of cloud content at Oracle. He was previously editor of InformationWeek*. Jeff Erickson is editor-at-large for Oracle Content Central.*

ILLUSTRATION BY **PEDRO MURTEIRA**

NEXT STEPS

WATCH
Think Autonomous.

LEARN more about Oracle Autonomous Database Cloud.

TRY Oracle Autonomous Transaction Processing Cloud.

LEARN more about Oracle Cloud Platform autonomous services.

Interactive Scientific was founded in 2013 with the goal of “taking all the complexity of science and making it simple and accessible using digital tools,” says Becky Sage, CEO at Interactive Scientific.

MOLECULAR DYNAMICS FOR THE MASSES

Interactive Scientific uses Oracle Cloud to bring 3D science to everyone, on everyday devices. **BY MIKE FADEN**

Interactive Scientific Ltd. is on a mission to change the way people experience, research, and learn about science. The company's simulation platform is designed to bring science to life for everyone from K–12 students to researchers. Students can understand molecules by interacting with 3D models instead of relying on abstract chemical formulas. But it has other uses as well—for example, making it possible for pharmaceutical researchers to build new drug molecules, transform them into different shapes to explore how they work, and investigate ways to design and deliver them.

Because such simulations are highly compute-intensive, they've traditionally required substantial onsite computing power delivered by powerful servers or even supercomputers. But by using Oracle Cloud to do the heavy lifting, Interactive Scientific is making its technology available almost anywhere—on everyday devices including smartphones, tablets, and PCs as well as on virtual reality (VR) headsets.

"We're on a mission to bring to life the invisible scientific world that exists all around us," says Becky Sage, CEO at Interactive Scientific. "Oracle opens up new opportunities for us. With compute power and accessibility via the cloud, we are able to scale up this really complex science technology and help solve problems in education and beyond."

Bringing Science to Life

Based in Bristol, England, Interactive Scientific was founded in 2013 with the goal of "taking all the complexity of science and making it simple and accessible using digital tools," Sage says.

The company initially focused on education, aiming to make science education fun for kids who are bored by the way it's currently taught. "We felt that science education wasn't suited to the world that we live in right now—a lot of people are getting disengaged and switched off," Sage says. "But science can be extremely exciting, and we can use digital tools to bring science to life."

Interactive Scientific's Nano Simbox software turns complex molecular data into interactive models, using a simulation technique known as molecular dynamics. The goal is to make science fun and playful—to make the invisible world of molecules visible and engaging for students.

Designed to support school curricula, the software lets students explore and manipulate

> **“With compute power and accessibility via the cloud, we are able to scale up this really complex science technology and help solve problems in education and beyond.”**
>
> —**Becky Sage,** CEO, Interactive Scientific

molecules while being prompted to discover additional facts for themselves. Although the first products using the platform originally targeted K–12 students, Interactive Scientific is now collaborating with universities to bring it to higher education as well.

The company is also starting to expand into pharmaceutical research. In the past, researchers painstakingly constructed physical models of molecules to visualize how they work. Today, simulations let researchers assemble and explore new molecules many times as fast. For example, using Nano Simbox VR technology, researchers can play a kind of 3D molecular Tetris—arranging different molecules together, ultimately to discover new drugs, Sage says.

“For researchers, the platform is used more as a toolkit,” she explains. “They want to upload their own simulations and use the platform to manipulate and understand their molecular systems in the way that’s most useful to them.”

The platform could also accelerate research, as demonstrated by a recent VR project involving Interactive Scientific, Oracle, and scientists at Bristol University. In the project, users performed tasks applicable to molecular research—such as retwisting helical molecules and even tying them in knots—significantly faster and more successfully with VR headsets and controls than with a traditional touchscreen or mouse. “As simulations become faster, we can now do this in real time, which will change how drugs are designed and how chemical structures are taught,” says Adrian Mulholland, a Bristol University chemistry professor.

“We want to give people access to the same rigorous science that universities and pharma companies run on their supercomputers—and to consume these simulations on a smartphone, regular laptop, or VR headset,” says Phill Tew, CTO at Interactive Scientific.

Powering Real-Time Simulations

Making real-time molecular dynamics available on consumer devices such as smartphones and VR headsets is key to Interactive Scientific’s plans to bring the technology to a much bigger audience. “We want to give people access to the same rigorous science that universities and pharma companies run on their supercomputers—and to consume these simulations on a smartphone, regular laptop, or VR headset,” says Phill Tew, Interactive Scientific CTO.

However, this approach also requires a huge amount of back-end processing power to execute the simulations, which are then displayed over the network to users’ devices. Interactive Scientific wanted to make sim-

ulations available to a wide range of users, including institutions that didn't have the wherewithal to invest in supercomputers. So two years ago, the company started to look into how it could use the cloud to provide that back-end processing power. Another big potential advantage of the cloud was that it could support collaborative research, allowing multiple researchers in different locations to work on the same model at the same time.

One big obstacle: at the time, the company had no experience with cloud computing. To help make the leap to the cloud, Interactive Scientific applied to join the Oracle Startup Cloud Accelerator Program, which was just getting off the ground in the UK. Run by Oracle's research and development staff, the worldwide program aims to help startups by providing technical and business mentoring; coworking space; state-of-the-art technology; free Oracle Cloud credits; and access to Oracle customers, partners, and investors.

Interactive Scientific was one of just 5 companies selected from more than 100 that applied to join the first UK cohort of startups. Tew says that Oracle's support and expertise proved invaluable. "I had a crash course in cloud computing, and most of the lessons I've learned have been through talking to Oracle engineers," he says. "They really primed us for the problems that we couldn't foresee at the outset. How do you make something scalable in the cloud? How do you slice up a problem in the correct way, so that when you scale up it doesn't explode on you?"

The challenge was particularly daunting because the company was trying to achieve something that had never been done. "We had to prove it was even possible to run the simulations in the cloud and stream them to your phone or another device," Tew says. "It really wasn't a solved problem at all."

Oracle's support helped the company meet the challenges, one step at a time. "We didn't want to go from zero to autoscaling infrastructure immediately. There are a bunch of steps in between—from the initial prototype implementation to creating a scalable architecture," says Tew.

Interactive Scientific found that running the simulations on Oracle Cloud Infrastructure Compute bare metal instances provided the performance the company needs to support multiple users in real time. "We get the full power

> **We had to make a product infrastructure that would allow deployment both on premises and in the cloud, to allow seamless access for these simulations. We've containerized every part of our stack.**
>
> —**Phill Tew,** CTO, Interactive Scientific

of the compute resources that we need: direct access to the bare metal without the abstraction of virtual machines," Tew says. "The algorithms used in real-time molecular dynamics are really fast, so we don't need a GPU. We can just run the simulations on a regular server with a bunch of cores. And we can run them in real time."

To ensure that the simulation software was both scalable and portable, Interactive Scientific divided it into container-based microservices and migrated it to Linux, Tew adds. Besides ensuring the required performance in the cloud, this approach also meant that universities and pharmaceutical companies that wish to use their existing on-premises infrastructure to run the simulations can continue to do so. "We had to make a product infrastructure that would allow deployment both on premises and in the cloud, to allow seamless access for these simulations," he says. "We've containerized every part of our stack."

To support real-time simulations, Interactive Scientific aims to refresh the model displayed on users' devices at 30 frames per second, Tew says. That refresh rate ensures that users don't experience lags or jerky movements as they manipulate molecular models. To achieve that goal, it's critical to consistently move data quickly between the cloud and the client device. The broad geographic presence of Oracle Cloud has helped to make that possible; by running the back-end software in a local Oracle Cloud data center, the company is currently able to deliver the required 30-frames-per-second refresh rate when simulating molecules as large as 8,000 atoms. In addition, Interactive Scientific has developed data-compression and other techniques for reducing bandwidth requirements, making it possible to display models even on a smartphone using a 4G connection, Tew says.

From Zero to Cloud in Six Months

With help from Oracle Cloud, Interactive Scientific's molecular simulations are making science more accessible—and fun—while helping to transform the way researchers discover new medicines. "The best practices shared by the Oracle team have helped us create a scalable solution," Tew says. "Thanks to our collaboration with Oracle, we were able to go from zero cloud experience to full implementation in six months."

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

PHOTOGRAPHY BY **JOHN BLYTHE**

NEXT STEPS

LEARN about Oracle Cloud Infrastructure Compute.

TRY Oracle Cloud Platform.

WATCH *Oracle Cloud Powers Interactive Scientific's Simulation Engine*.

READ about the Oracle Startup Cloud Accelerator program.

ORACLE
OPEN
WORLD

Oct. 22–25, 2018 | San Francisco | #OOW18

Don't Miss Out

Sessions, Networking, and Product Training Designed for You

Explore, Learn, Connect, Inspire

ORACLE®

// Global Sponsor

Deloitte.

// Premier Sponsors

accenture

pwc

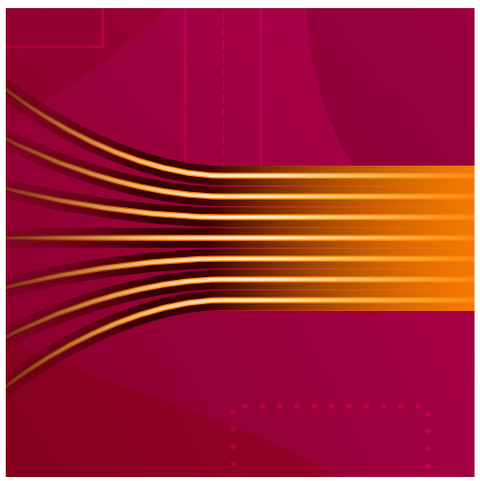
TATA CONSULTANCY SERVICES

// Grande Sponsors

intel

KPMG

wipro



ORACLE DATABASE

Streaming Table Functions

Transform your data on the way to your data warehouse.

A table function is a function you can use like a table in the FROM clause of a SELECT statement. A common usage of table functions is to stream data directly from one process or transformation to the next process without intermediate staging, and a table function used in this way is called a *streaming table function*. As you might be able to tell from the reference to *transformation* above, this technique is most often used in data warehouses as part of extract, transform, and load (ETL) operations.

My previous article, "When Is a Function Like a Table? When It's a Table Function!," includes an overview of table functions. In this article, I will show you the basic steps needed to create a streaming table function, but before diving into the details, let's look at a streaming table function example:

```
INSERT INTO tickers
   SELECT *
```

```
FROM TABLE (doubled (CURSOR (SELECT * FROM stocks)))
```

What's going on here? Let's take it step by step from the inside out:

Code	Description
`SELECT * FROM stocks`	Gets all the rows from the stocks table
`CURSOR ()`	Creates a cursor variable with the CURSOR expression that points to the result set
`()`	Passes that cursor variable to the doubled table function
`doubled ()`	Performs its transformation and returns a nested table of object type instances
`SELECT * FROM TABLE(...)`	Converts the collection into a relational set of rows and columns
`INSERT INTO tickers`	Inserts those rows into the tickers table

Sometimes (often?) you will need to perform more than one transformation as part of the streaming process. No problem. You can certainly string together multiple invocations of table functions:

```
INSERT INTO tickers
SELECT *
  FROM TABLE (transform2 (
               CURSOR (SELECT *
                         FROM TABLE (transform1 (
                                       CURSOR (SELECT * FROM stocks
      ))))))
```

SETTING UP TABLES FOR TRANSFORMATION

To transform data from one table to another, you need tables and the data in those tables. In this article, I will start with the stocks table, each row of which contains the opening and closing trading prices for each stock ticker symbol:

```
CREATE TABLE stocks (
ticker VARCHAR2 (20),
trade_date DATE,
opening_price NUMBER,
closing_price NUMBER)
/
```

My transformation is simple: for each row in the stocks table, generate two rows for the tickers table (one row each for the opening and closing prices):

```
CREATE TABLE tickers
(
   ticker      VARCHAR2 (20),
   pricedate   DATE,
   pricetype   VARCHAR2 (1),
   price       NUMBER
)
/
```

Before continuing, I feel obligated to point out that for this particular transformation (one row in stocks to two rows in tickers), you don't *need* a table function to get

the job done. For example, you can use INSERT ALL to insert into tickers twice:

```
INSERT ALL
  INTO tickers (ticker,
                pricedate,
                pricetype,
                price)
VALUES (ticker,
        trade_date,
        ‹O›,
        opening_price)
  INTO tickers (ticker,
                pricedate,
                pricetype,
                price)
VALUES (ticker,
        trade_date,
        ‹C›,
        closing_price)
   SELECT * FROM stocks
/
```

You could also use UNPIVOT (thanks, Chris Saxon @chrisrsaxon, for making me aware of this technique!):

```
INSERT INTO tickers (ticker,
```

```
                    pricedate,
                    pricetype,
                    price)
   SELECT *
     FROM stocks UNPIVOT (price
                 FOR price_type
                 IN (opening_price AS 'O', closing_price AS 'C'))
/
```

SQL is an extraordinarily powerful language. It is quite likely that the transformation you would like to perform *is* doable in pure SQL. And if you *can* avoid the use of a table function, implementing your requirement in SQL instead, then you should by all means do so. To strengthen your SQL skills, take advantage of workouts, classes, and quizzes at the Oracle Dev Gym, especially the SQL analytics class by Connor McDonald.

If, however, the transformation requires the use of procedural logic (hence, PL/SQL) or if you can't sort through the SQL syntax, table functions offer a powerful, straightforward way to get the job done.

For the purposes of this article, assume that the transformation is much more complex and requires use of a table function.

TYPES AND PACKAGE FOR TABLE FUNCTIONS

As shown in my previous article on table functions, when you need a table function to return more than one piece of data in each collection element (for example, more than just a list of strings or numbers), you need to create an object type and a collection of those object types.

In this article's example, I want to move stock data from the stocks table to the tickers table, so I need an object type that "looks like" the tickers table.

Ideally, I would create a collection type like this:

```
CREATE TYPE tickers_nt AS TABLE OF tickers%ROWTYPE;
/
```

But %ROWTYPE is a PL/SQL declaration attribute and is not known to the SQL engine, so this statement fails with

```
PLS-00329: schema-level type has illegal reference to TICKERS
```

Instead, I create an object type that mimics the structure of my table, as follows:

```
CREATE TYPE ticker_ot
   AUTHID DEFINER IS OBJECT
(
   ticker VARCHAR2 (20),
   pricedate DATE,
   pricetype VARCHAR2 (1),
   price NUMBER
);
/
```

I then create a nested table of those object types:

```
CREATE TYPE tickers_nt AS TABLE OF ticker_ot;
/
```

I plan to use this table function in a streaming process. This means that I will be passing in a set of data (rows and columns) from SQL. To do this, I will also need to define a strong REF CURSOR type that will be used as the data type of the parameter accepting the dataset inside the SQL statement. In the package specification below, I create *two* REF CURSOR types, one for rows from the stocks table and another for the tickers table.

```
CREATE OR REPLACE PACKAGE stock_mgr AUTHID DEFINER
IS
   TYPE stocks_rc IS REF CURSOR
      RETURN stocks%ROWTYPE;

   TYPE tickers_rc IS REF CURSOR
      RETURN tickers%ROWTYPE;
END stock_mgr;
/
```

The variable you declare based on a REF CURSOR type is a cursor variable. Within PL/SQL, you might write code like this:

```
DECLARE
   l_cursor   stock_mgr.stocks_rc;
   l_stock    stocks%ROWTYPE;
```

```
BEGIN
   /* With a static SQL statement */
   OPEN l_cursor FOR SELECT * FROM stocks;

   LOOP
      FETCH l_cursor INTO l_stock;

      EXIT WHEN l_cursor%NOTFOUND;
   END LOOP;

   CLOSE l_cursor;

   /* Or with a dynamic SQL statement */
   OPEN l_cursor FOR 'select * from stocks';

   LOOP
      FETCH l_cursor INTO l_stock;

      EXIT WHEN l_cursor%NOTFOUND;
   END LOOP;

   CLOSE l_cursor;
END;
/
```

Note that you can use all the usual cursor attributes and operations on cursor variables: FETCH, %FOUND, CLOSE, and so on.

As you will see later, the way you use this REF CURSOR type will be a bit different for streaming table functions in the SQL context.

DEFINE THE TABLE FUNCTION

The main distinction between streaming table functions and "normal" table functions such as those addressed in the previous article is that *at least* one parameter to that streaming table function is a cursor variable. The table function *could* have more than one cursor variable input and other parameters of other types such as a string or date. In this article, I will stick with the minimum: a single cursor variable parameter.

Generally, the flow within a streaming table function is

1. Fetch a row from the cursor variable.
2. Apply the transformation to each row.
3. Put the transformed data into the collection.
4. Return the collection when done.

Now let's see how this pattern unfolds in my doubled function—one stocks row doubled to two ticker rows—and the line descriptions that follow.

```
1 CREATE OR REPLACE FUNCTION doubled (rows_in stock_mgr.stocks_rc)
2    RETURN tickers_nt
3    AUTHID DEFINER
4 IS
5    TYPE stocks_aat IS TABLE OF stocks%ROWTYPE INDEX BY PLS_INTEGER;
```

```
   l_stocks    stocks_aat;

   l_doubled   tickers_nt := tickers_nt ();
BEGIN
   LOOP
      FETCH rows_in BULK COLLECT INTO l_stocks LIMIT 100;
      EXIT WHEN l_stocks.COUNT = 0;

      FOR l_row IN 1 .. l_stocks.COUNT
      LOOP
         l_doubled.EXTEND;
         l_doubled (l_doubled.LAST) :=
            ticker_ot (l_stocks (l_row).ticker,
                       l_stocks (l_row).trade_date,
                       'O',
                       l_stocks (l_row).opening_price);

      l_doubled.EXTEND;
      l_doubled (l_doubled.LAST) :=
         ticker_ot (l_stocks (l_row).ticker,
                    l_stocks (l_row).trade_date,
                    'C',
                    l_stocks (l_row).closing_price);
   END LOOP;
END LOOP;
```

```
31    CLOSE rows_in;
32
33    RETURN l_doubled;
34 END;
```

Line(s)	Description
1	Use the REF CURSOR type defined in the package for the rows passed in. Because I am selecting from the stocks table, I use the stocks_rc type.
2	Return an array, each of whose elements looks *just like* a row in the tickers table.
5, 6	Declare an associative array to hold rows fetched from the rows_in cursor variable.
8	Declare the local variable to be filled and then returned back to the SELECT statement.
10	Start up a simple loop to fetch rows from the cursor variable. It's already open—the CURSOR expression takes care of that.
11, 12	Use the BULK COLLECT feature to retrieve as many as 100 rows with each fetch. I do this to avoid row-by-row processing, which is not efficient enough. Exit the loop when the associative array is empty.
14	For each element in the array (row from the cursor variable)...
16–21	Use EXTEND to add another element at the end of the nested table, then call the object type constructor to create the first ("opening") of the two rows for tickers table, and put the element into the new last index value of the collection.
23–28	Do the same for the second ("closing") row of the tickers table.
31	Close the cursor variable, now that all the rows have been fetched. Note: this step is optional. When you use a CURSOR expression to pass in the result set, the cursor will be closed automatically when the function terminates.
33	Send the nested table back to the SELECT statement for streaming.

REGARDING FETCH-BULK COLLECT-LIMIT

I used a value of 100 for the LIMIT clause. That's a decent default value—it's the number of rows retrieved by cursor FOR loops with each fetch. But if you are processing an extremely large number of rows and want to squeeze better performance out of your function, you might try a larger LIMIT value. Note, however, that this will consume more Process Global Area (PGA) memory and that, at some point, your code will slow down due to excessive memory consumption.

You should also pass the LIMIT value as a parameter to give you the ability to modify the performance profile without recompiling your function, as in

```
CREATE OR REPLACE FUNCTION doubled (
   rows_in stock_mgr.stocks_rc, limit_in IN INTEGER DEFAULT 100)
...
BEGIN
   LOOP
      FETCH rows_in BULK COLLECT INTO l_stocks LIMIT limit_in;
```

OK, let's do some streaming. Assume that I previously loaded the stocks table with 1,000 rows.

```
SELECT COUNT (*) c FROM tickers
/

C
____
0
```

```
INSERT INTO tickers
   SELECT * FROM TABLE (doubled (CURSOR (SELECT * FROM stocks)))
/

2000 row(s) inserted.

SELECT COUNT (*) FROM tickers
/

C
____
2000

     SELECT *
       FROM tickers
FETCH FIRST 10 ROWS ONLY
/

TICKER  PRICEDATE  PRICETYPE  PRICE
STK214  06-JUL-18  C          229
STK215  06-JUL-18  O          215
STK215  06-JUL-18  C          230
STK216  06-JUL-18  O          216
STK216  06-JUL-18  C          231
STK217  06-JUL-18  O          217
STK217  06-JUL-18  C          232
```

```
STK218  06-JUL-18  O          218
STK218  06-JUL-18  C          233
STK219  06-JUL-18  O          219
```

A TWO-STEP TRANSFORMATION

In my two-step transformation, I will create a function that returns a nested table of elements that matches the stocks table. So I will need an object type and a nested table type.

```
CREATE OR REPLACE TYPE stock_ot
   AUTHID DEFINER IS OBJECT
(
   ticker VARCHAR2 (20),
   trade_date DATE,
   opening_price NUMBER,
   closing_price NUMBER
)
/

CREATE OR REPLACE TYPE stocks_nt AS TABLE OF stock_ot;
/
```

Now I create a table function that accepts a cursor variable, each of whose rows contains ticker data and returns a nested table, each of whose elements looks like a row in the stocks table. Because it follows precisely the same pattern as the doubled function, I will not describe the individual lines.

```
CREATE OR REPLACE FUNCTION singled (tickers_in IN stock_mgr.tickers_rc)
   RETURN stocks_nt
   AUTHID DEFINER
IS
   TYPE tickers_aat IS TABLE OF tickers%ROWTYPE
      INDEX BY PLS_INTEGER;

   l_tickers   tickers_aat;
   l_singles   stocks_nt := stocks_nt ();
BEGIN
   LOOP
      FETCH tickers_in BULK COLLECT INTO l_tickers LIMIT 100;

   EXIT WHEN l_tickers.COUNT = 0;

   FOR indx IN 1 .. l_tickers.COUNT
   LOOP
      l_singles.EXTEND;
      l_singles (l_singles.LAST) :=
         stock_ot (l_tickers (indx).ticker,
                   l_tickers (indx).pricedate,
                   l_tickers (indx).price,
                   l_tickers (indx).price * .5);
   END LOOP;
END LOOP;
```

```
   RETURN l_singles;
END;
/
```

Now let's do a two-step transformation: stocks -> tickers -> more_stocks!

```
CREATE TABLE more_stocks

AS
   SELECT *
     FROM TABLE (
             singled (
                CURSOR (
                   SELECT *
                     FROM TABLE (doubled (CURSOR (SELECT * FROM stocks))))))
/

SELECT COUNT (*) FROM more_stocks
/
```

SUMMARY

Streaming table functions play a crucial role in data warehouse ETL operations. Oracle Database makes building such functions easy, through its implementation of PL/SQL cursor variables and the CURSOR expression.

Remember that the collection constructed and returned by a streaming table func-

tion will consume PGA memory, so very large datasets passed into the function via the cursor variable may result in errors.

What can you do about that? Make that streaming table function a *pipelined* streaming table function, which I'll cover in the next article in this series.

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 more about table functions.

LEARN more about PL/SQL table functions (a Dev Gym class).

EXPLORE the code used in this article at Oracle Live SQL.

ORACLE DATABASE

Build REST APIs for Node.js, Part 3

Add routing, controller, and database logic.

After our brief detour to create a generic database module, it's time to continue building out the high-level REST API components for Node.js discussed in the first article of this series. In this article, you will add routing, controller, and database logic to handle an HTTP GET request on an "employees" API endpoint.

Note: the instructions and steps in this article assume that you have completed the steps in Part 1 of this article series.

ADDING ROUTING LOGIC

Express, which creates the web server for this project, ships with a Router class that makes it easy to route HTTP requests to the appropriate controller logic. Route paths define the URL endpoints of the API and can contain route parameters that capture values in the URL (query strings are not part of route paths).

There are many ways to define the routes for your application. For example, when the app starts, you could read all the files in the controllers directory and autogene-

rate routing logic based on some predefined rules, such as the filenames and properties they expose. Alternatively, you could add a file to the config directory and read that at start time. Consider such automation when your API matures and its patterns are well known.

In this application, you will take a slightly lower-level approach by defining routes programmatically via a new router module. Create a new file named router.js in the services directory. Add the following code to the file, and save your changes:

```
const express = require('express');
const router = new express.Router();
const employees = require('../controllers/employees.js');

router.route('/employees/:id?')
  .get(employees.get);

module.exports = router;
```

This router module starts by bringing in Express and then creates a new instance of Express' Router class. The router's `route` method is used to define a route based on the route path passed in. The path includes a parameter named `id`, which is made optional by the question mark that follows it. The route that's returned from `route` includes methods that correspond to HTTP methods and enable handlers to be defined. In this case, the `get` method is used to map an incoming GET request to the `get` function defined in the employees controller (which will be created in the next part of this article).

At this point, you have a router but it's not currently used in the application. To use it, open the services/web-server.js file and remove the line at the top that requires the database module (it was used only for testing in the previous article). Add the following line of code in its place:

```
// *** line that requires ../config/web-server.js is here ***
const router = require('./router.js');
```

Next, replace the entire app.get handler that responds to GET requests using the database module (all seven lines) with the following code:

```
// *** line that adds morgan to app here ***

// Mount the router at /api so all routes start with /api
app.use('/api', router);
```

Now the router is required in the web service module and mounted at /api. This means that the full URL for the employees endpoint will be http://server:port/api/employees/:id.

ADDING CONTROLLER LOGIC

The controller logic will take over from the point that the URL endpoint and the HTTP method are known. Because the web server is built with Express, the controller logic will be defined with custom middleware or functions that have access to the request and response objects as well as the `next` function.

The middleware function will use incoming data from the request object to generate a response that is sent using the response object. The `next` function is typically used to invoke the next middleware function in the pipeline. However, in this API, the controller logic will be the last step in the pipeline and will end the HTTP response. The next function will be invoked only if an error occurs, in which case control will be passed to Express' default error handler.

I usually create one module in the controllers directory for each endpoint in the API. Here are some examples:

URL Endpoint	Controller File
/api/employees/:id	controllers/employees.js
/api/departments/:id	controllers/departments.js
/api/departments/:dept_id/employees/:emp_id	controllers/departments_employees.js

A middleware function that handles a particular HTTP method will be defined and exposed within each module. I usually name each function based on the HTTP method it handles, which makes it easy to wire things up in the router module.

Go to the controllers directory, and create a new file named `employees.js`. Copy and paste the following code into the file, and save your changes:

```
1 const employees = require('../db_apis/employees.js');
2
3 async function get(req, res, next) {
4   try {
```

```
  const context = {};

  context.id = parseInt(req.params.id, 10);

  const rows = await employees.find(context);

  if (req.params.id) {
    if (rows.length === 1) {
      res.status(200).json(rows[0]);
    } else {
      res.status(404).end();
    }
  } else {
    res.status(200).json(rows);
  }
} catch (err) {
  next(err);
}
}

module.exports.get = get;
```

Here’s a breakdown of the controller module so far:

Line(s)	Description
1	The employees database API (created in the next part of this article) is required.
3–23	An async function named `get` is declared. A try-catch block is used in the body of the function to catch exceptions thrown on the main thread and pass them to the `next` function.
5–7	A constant named `context` is declared. This is a generic object that will contain properties that are relevant to the database API's `find` method. An id property is added to `context`, based on the value that comes in via `req.params.id`.
9	The database API's `find` method is used to fetch the appropriate employee records in the database.
11–19	Conditional logic is used to determine the correct HTTP status code and body for the response. If one employee was requested but not found, a "404 Not Found" error code will be sent as a response. Otherwise, a "200 OK" code, along with a JSON-based response body, will be sent.
25	The `get` function is exported from the module so it can be used in the router module.

The `req.params` object is just one of several properties used to get data from the incoming request object. Other common properties include `req.query`, for the query string values in the URL; `req.body`, for the request body; and `req.cookies`. HTTP headers can be fetched with the `req.get` method.

If you don't like the magic numbers used for the status codes, consider using a module such as http-status instead. That module provides constants, such as `OK` and `NOT_FOUND`, that can add clarity to the code.

ADDING DATABASE LOGIC

As I mentioned in Part 1 of this article series, I'll be using the Node.js database driver/API for Oracle Database, node-oracledb, instead of a higher-level object-relational mapping (ORM). To start the employees database module, go to the db_apis directory and create a new file named employees.js. Add the following code to the file:

```
const database = require('../services/database.js');

const baseQuery =
 `select employee_id "id",
    first_name "first_name",
    last_name "last_name",
    email "email",
    phone_number "phone_number",
    hire_date "hire_date",
    job_id "job_id",
    salary "salary",
    commission_pct "commission_pct",
    manager_id "manager_id",
    department_id "department_id"
  from employees`;

async function find(context) {
  let query = baseQuery;
  const binds = {};
```

```
  if (context.id) {
    binds.employee_id = context.id;

    query += `\nwhere employee_id = :employee_id`;
  }

  const result = await database.simpleExecute(query, binds);

  return result.rows;
}

module.exports.find = find;
```

The employees database module brings in the generic database module and then initializes a constant named `baseQuery` to a SQL query on the employees table. Double-quoted column aliases are used in the query to control the case of the keys returned.

Next, a function named `find` is declared and used to execute the query and return the rows fetched. If the `context` parameter passed in has a "truthy" value for `id`, then a WHERE clause is appended to the query so that only a single employee is returned.

Note that the value of `context.id` was not appended to the query directly. Instead, a placeholder named `:employee_id` was used—this is known as a *bind variable*. Using bind variables with Oracle Database is *very* important, for security and performance reasons. The value of the bind variable is assigned to the `binds` object, which

is passed with the query to `database.simpleExecute`. Finally, the rows returned from the database are returned to the caller.

Once the database module is in place, you'll be ready to test everything. Start the app, and then navigate Firefox to http://localhost:3000/api/employees. You should see a list of employees (I've collapsed a couple), as shown in **Figure 1**.

You can fetch a single employee by adding an ID to the end of the URL, such as http://localhost:3000/api/employees/100, as shown in **Figure 2**.

Figure 1: Getting the list of employees

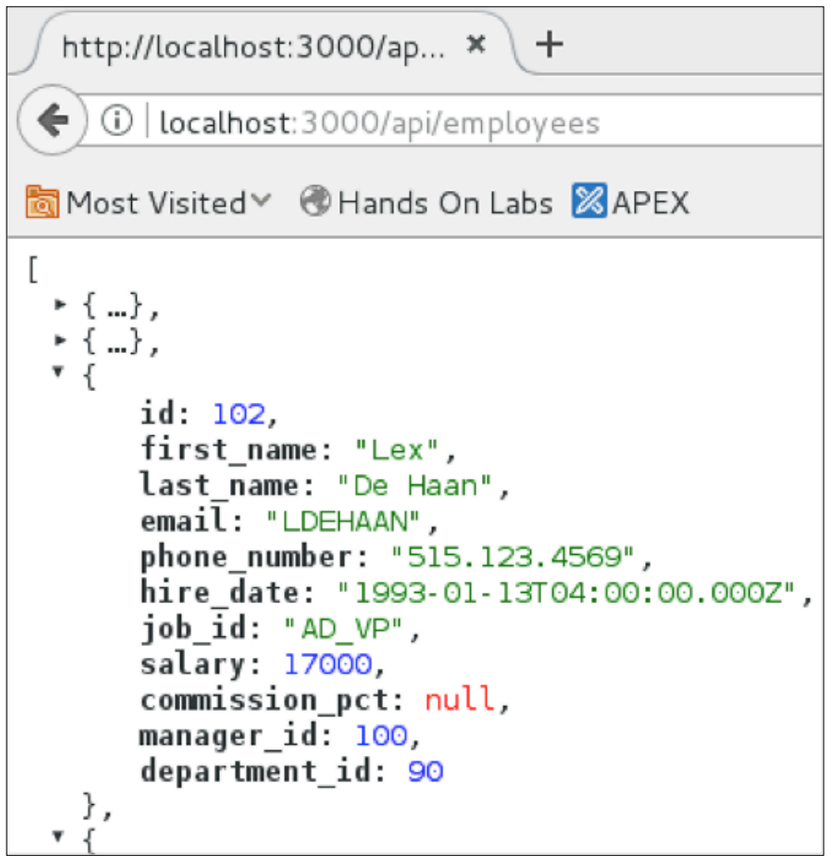
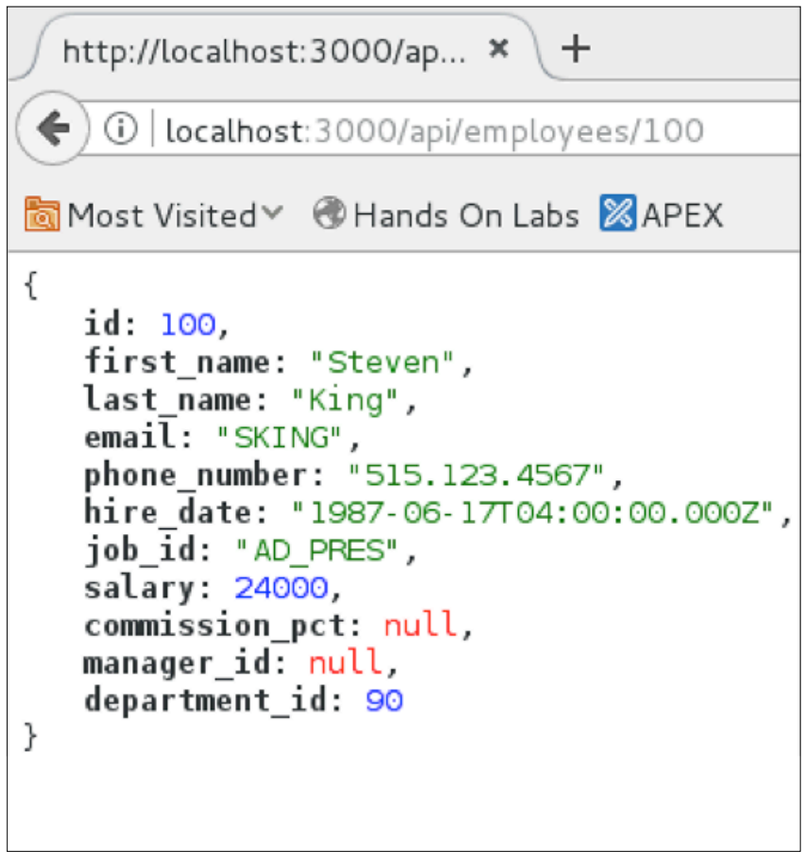


Figure 2: Getting a single employee



At this point, your API can handle GET requests on the employees endpoint. In the next article, you will round out the create, replace, update, delete (CRUD) functionality by adding logic that handles POST, PUT, and DELETE requests.

Dan McGhan is the Oracle developer advocate for JavaScript and Oracle Database. He enjoys sharing what he's learned about these technologies and helping others be successful with them.

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

READ Part 1 of this article series.

READ Part 2 of this article series.

LEARN more about JavaScript and Oracle.

TRY Oracle Cloud Platform.

GET more about this article's code from GitHub.

ORACLE APPLICATION EXPRESS

Beyond the Database

Integrate application data from anywhere with Oracle Application Express 18.1.

Oracle Application Express (Oracle APEX) is the low-code application development framework of Oracle Database, enabling the easy creation of modern and responsive web apps. Oracle APEX runs in Oracle Database and provides zero-latency access to your application data. But increasingly in enterprises and in the cloud, the datasource for your application may not be resident in your local Oracle Database instance. Using web and RESTful APIs over HTTP has become the industry-standard method to provide programmatic access to APIs and data in heterogeneous, distributed systems. Oracle Application Express 18.1, with guided step-by-step wizards, now makes it easy to build powerful, feature-rich applications against remote, web, and REST datasources.

In this *Oracle Magazine* article, you're going to build a web application on top of the Apple iTunes Search API, which is a simple API over HTTP that takes input arguments via parameters in the URL. You will first create a report against the iTunes

music video catalog. Next, you will simplify the report to include a preview of the music video and a link to watch a clip of the music video. Finally, you will add a form field to the Oracle APEX application to permit dynamic searching of the iTunes music video catalog.

This article's sample application is built in Oracle Application Express 18.1. If you're not already running Oracle Application Express 18.1 or later locally, you can request a free workspace at https://apex.oracle.com. Alternatively, you can sign up for a free trial of Oracle Database Cloud Service. Oracle APEX can be used wherever you have an Oracle Database instance, including every Oracle Database cloud service.

CREATING THE APPLICATION

Begin your exploration of RESTful web services with Oracle Application Express 18.1 by creating the initial application.

1. In Oracle Application Express, click the **App Builder** tab.
2. Click the **Create** icon and then the **New Application** icon.
3. Enter `APEX and REST` for **Name**, and click the **Create Application** button.

If you were to run the application now, it would look a bit empty, with nothing but a home page. But now is where the fun begins. You first need to create a Web Source Module for the iTunes Search API. A Web Source Module is a shared component in Oracle APEX, and it is used to maintain the definition of a remote data source—the URL of the REST endpoint, data type mapping, credentials, and more. APIs returning JSON are the lingua franca of the internet, and using the native JSON capabilities of Oracle Database, the Oracle APEX engine can efficiently parse and process JSON results.

1. From within Application Builder, click the **Shared Components** icon.
2. In the Data Sources region, click **Web Source Modules** and then **Create**. You will

want to create this "from scratch," so click **Next**.

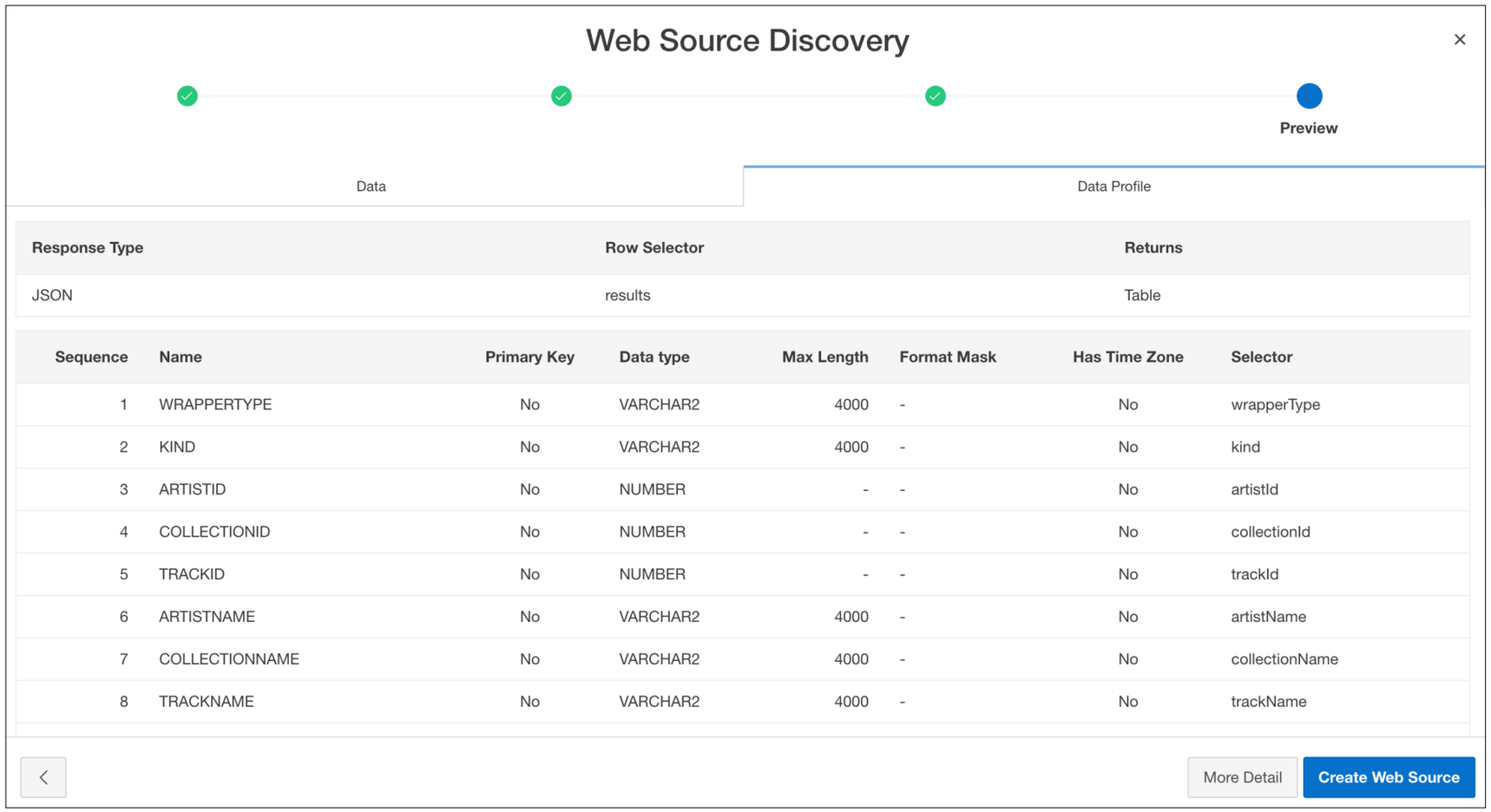
3. For **Web Source Type**, choose **Simple HTTP** (the default). Enter `iTunes Search` for **Name**, and for **URL Endpoint**, copy and paste the URL https://itunes.apple.com/search. Click **Next**.
4. Oracle APEX will parse the URL into its Base URL and Service URL path. Click **Next** again.
5. Because authentication will not be required for this API, leave the value for **Authentication Required** as the default No and click **Advanced**.
6. Choose a **Parameter Type** of **Query String variable**; for **Parameter Name**, enter `term`; and for **Value**, enter `beyonce`.
7. On the second row, choose a **Parameter Type** of **Query String variable**; enter a **Parameter Name** of `entity`; and for **Value**, enter `musicVideo`. Change **Is Static** to **Yes**, and click **Discover**.
8. Oracle APEX will invoke the REST API and attempt to parse the results and create a data profile. Click the **Data Profile** subtab to view the entities parsed from the JSON result and see how they will be mapped to specific column names and data types. Click **Create Web Source**.

You have now created a shareable component, a Web Source Module, which will return a list of music videos that include Beyoncé in any of the searchable attributes. **Figure 1** shows a preview of the data profile for this module.

Now you will expose this in your Oracle APEX application by creating a classic report on the Web Source Module.

1. Click the Application breadcrumb in the upper left to return to the top-level definition of your application in Application Builder.
2. Click **Create Page**. Click **Report** and then **Classic Report**.
3. Enter `iTunes` for **Page Name**, and then click **Next**.

Figure 1: The data profile of the JSON response, during Web Source Module creation



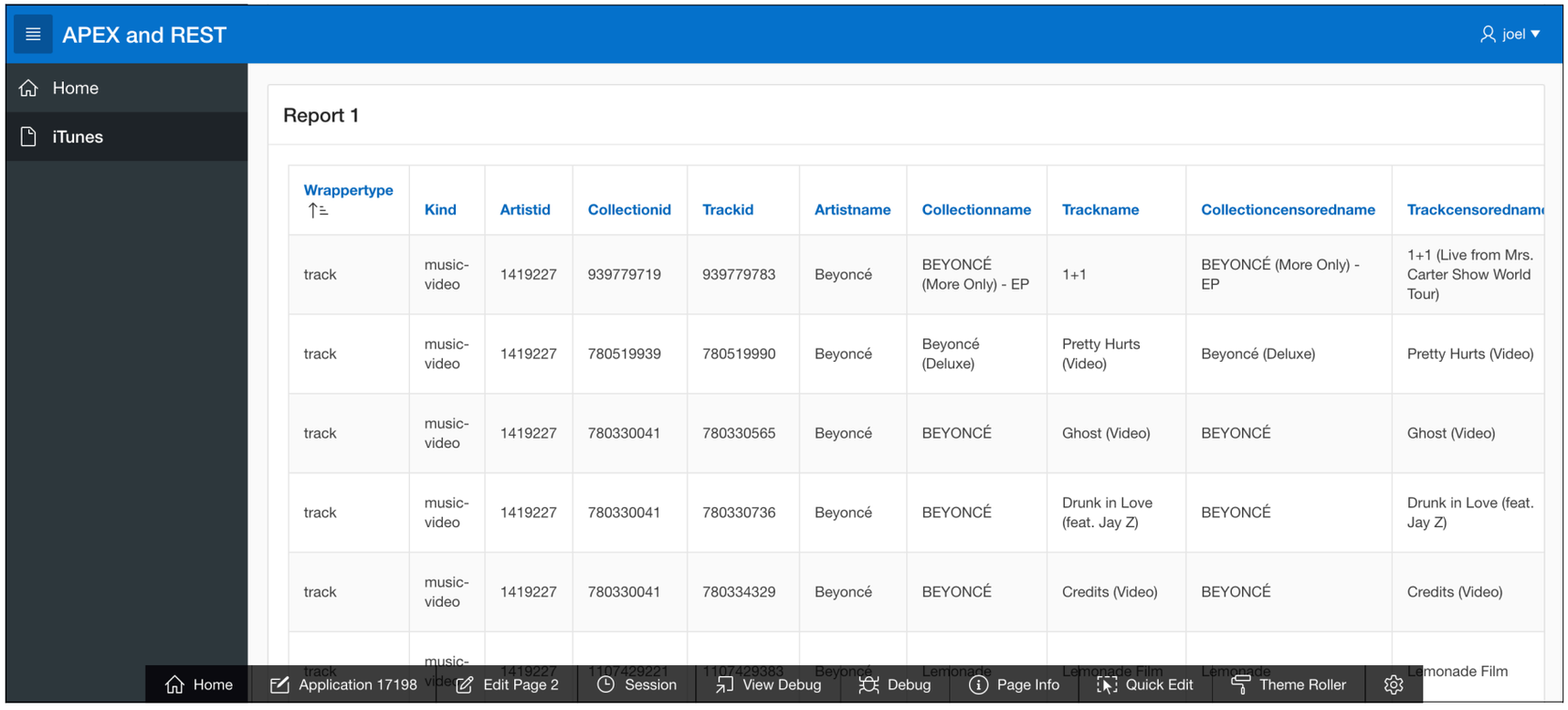
4. Select **Create a new navigation menu entry**, and click **Next**.
5. For **Data Source**, choose **Web Source**. Then select **iTunes Search** for **Web Source Module**. Click **Create** twice.

Voilà! You just created a page in your Oracle APEX application that will display the results of the iTunes Search API, searching for Beyoncé. Click the **Run Application** icon at the upper right to run your newly created Oracle APEX application and report. Using the same credentials you used to log in to your Oracle APEX workspace, authenticate yourself to the application and click the iTunes navigation

menu entry on the left. Scroll to the far right and paginate forward to see the second set of results.

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

Figure 2: The initial report application page



REFINING THE REPORT

Improve the display of the report page by reducing the number of columns and making use of graphics.

Once the results of a REST service are fetched by Oracle APEX, you can perform postprocessing on the results, using ordinary SQL. With postprocessing you can reduce the number of columns in the SELECT clause, limit the results further by adding a WHERE clause, or even join results to local tables. In this article, you will

use a SQL postprocessing mechanism to reduce the number of columns displayed in the report.

1. In the developer toolbar at the bottom of the page, click the **Edit Page 2** link.
2. In the tree view on the left of Page Designer, click **Report 1** to select the report region.
3. In the properties on the right, in the Local Post Processing section, choose **SQL Query** for **Type**.
4. In the SQL Query region, overwrite the existing query with a copy-and-paste of the following query:

```
select ARTISTNAME,
       TRACKNAME,
       PREVIEWURL,
       ARTWORKURL100,
       COLLECTIONPRICE,
       RELEASEDATE,
       COUNTRY,
       CURRENCY
  from #APEX$SOURCE_DATA#
```

5. In the Appearance section of the properties on the right, choose **Blank with Attributes** for **Template**. This will simplify the region display and remove the unnecessary region title.
6. In the tree view on the left of Page Designer, under the report region, expand the column list by clicking **>** to the left of **Columns**.

ABOUT ORACLE APPLICATION EXPRESS

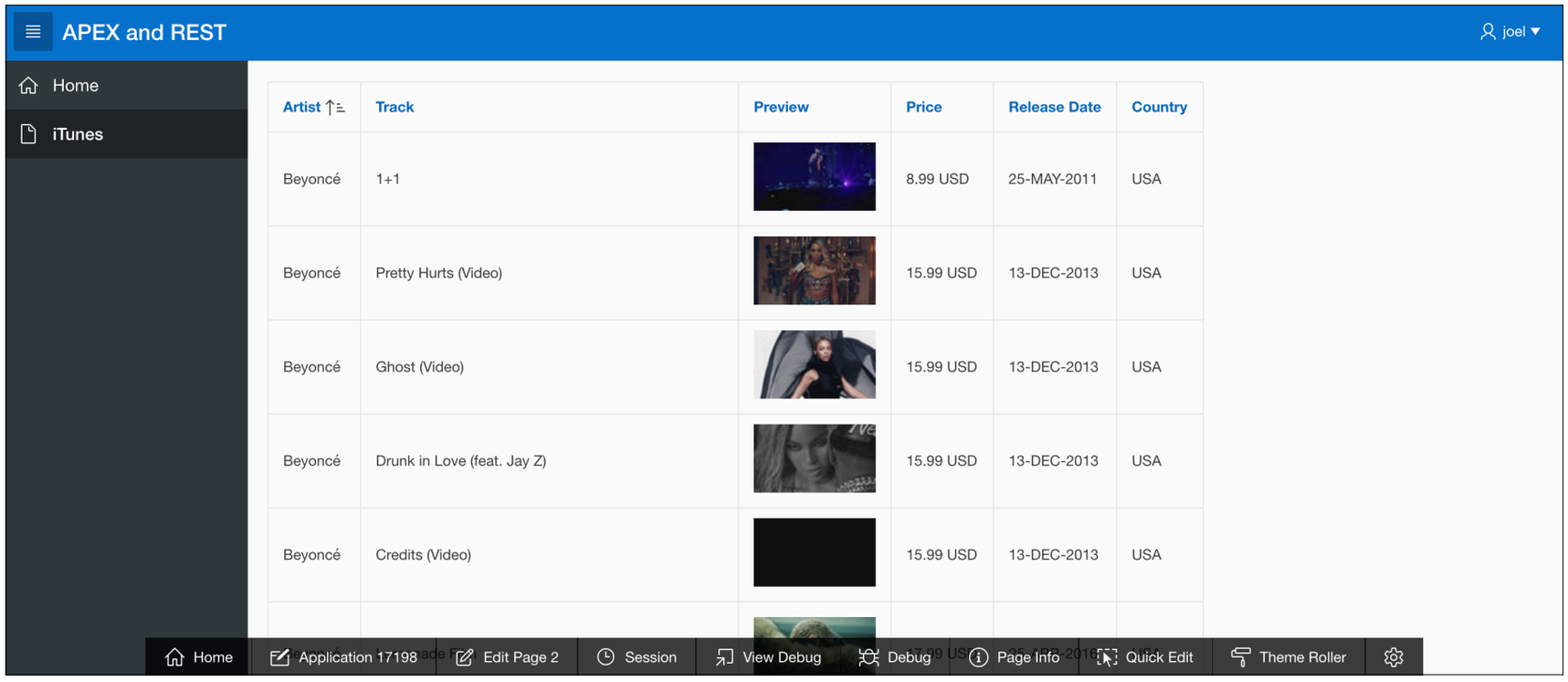
Oracle Application Express (Oracle APEX) is a high-productivity, low-code platform for creating modern, responsive, and accessible web applications. A no-cost feature of Oracle Database, it is a compelling application development platform available in all Oracle Database Cloud services.

7. Select **ARTISTNAME**, and change the heading to `Artist`. Change the other headings: **TRACKNAME** to `Track`, **COLLECTIONPRICE** to `Price`, **RELEASEDATE** to `Release Date`, and **COUNTRY** to `Country`.
8. Select the **ARTWORKURL100** column, and in the properties on the right, change **Type** to **Hidden Column**.
9. In the tree view on the left, select the **CURRENCY** column. In the properties on the right, change **Type** to **Hidden Column**.
10. In the tree view on the left, select the **PREVIEWURL** column. In the properties on the right, change **Heading** to `Preview`. Change **Type** to **Link**.
11. In the **Link** region of the **PREVIEWURL** column, click **No Link Defined** for the **Target** attribute. In the dialog box for the **Link** attribute, change **Type** to **URL**. In the **URL** attribute, enter `#PREVIEWURL#`. This will create an HTML anchor to the value of the Preview URL returned in the REST results. Click **OK**.
12. For **Link Text**, enter `<img src="#ARTWORKURL100#"/>`. This will display the image returned in the value of the **Artwork** column in the REST results.
13. For **Link Attributes**, enter `target="_new"`. This will cause the video preview to open in a new browser tab when clicked.
14. In the tree view on the left, select the **COLLECTIONPRICE** column. In the properties on the right, enter the HTML expression `#COLLECTIONPRICE# #CURRENCY#`. This will include the currency code after the price value.
15. Click the **Run Application** icon at the upper right to save all your changes and run your Oracle APEX application.

Your application now includes a much-better-looking report, with an image of the video preview. Additionally, if you click the image, it should open a new tab in your browser and start playing the video.

Your application should look similar to **Figure 3**.

Figure 3: The application with a reduced number of columns in the report and a clickable preview of each associated video



MAKING THE REPORT SEARCHABLE

You have produced a nice report that shows all of Beyoncé's videos on iTunes, and it fetches these results dynamically via the iTunes Search API on the internet. The last remaining step is to make this report searchable, so that the end user can search for any video, not just Beyoncé's.

1. In the developer toolbar at the bottom of the page, click the **Edit Page 2** link.
2. In the Component Gallery at the bottom of the Page Designer page, ensure that **Items** is selected. Select the **Text Field** icon, drag it up to the ITEMS section of the **Report 1** region in the Page Layout area, and drop it.

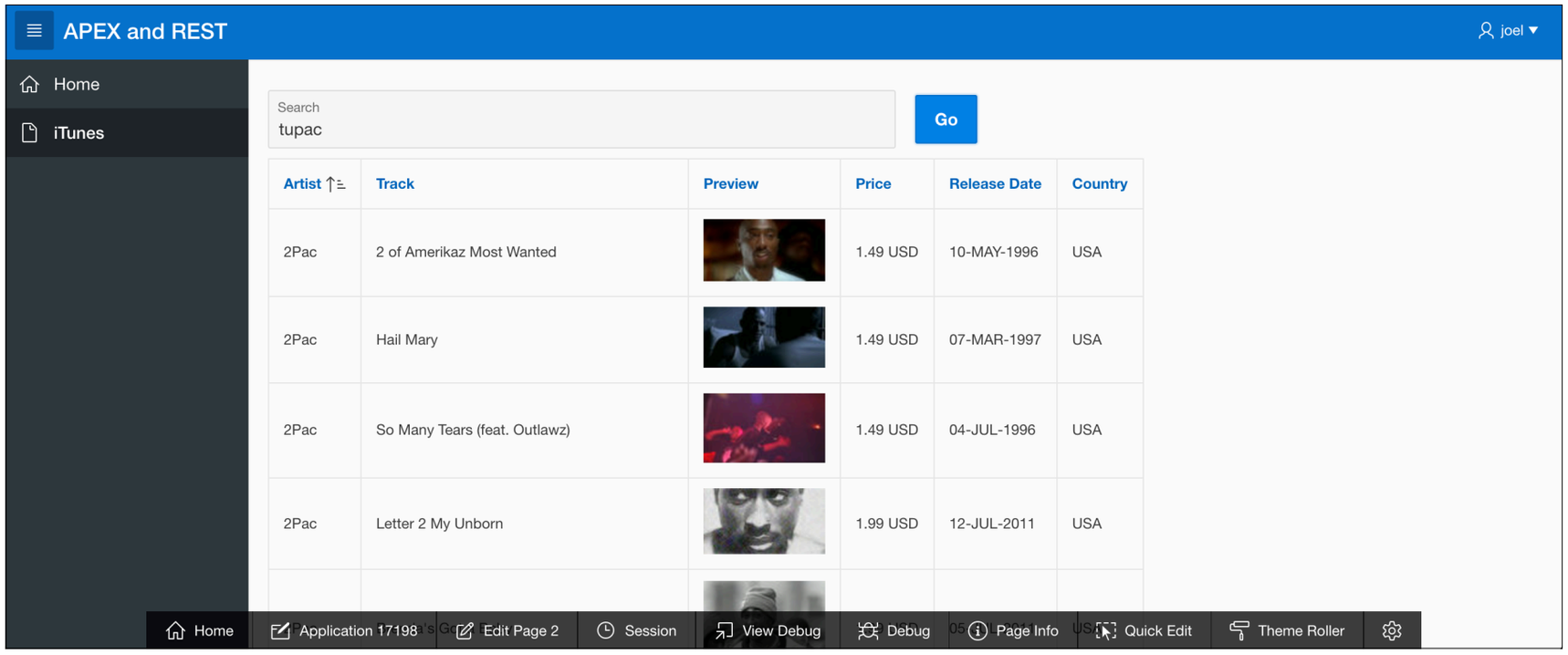
3. In the properties on the right of Page Designer, for **Name**, enter `P2_SEARCH`, and for **Label**, enter `Search`.
4. In the Component Gallery at the bottom of the Page Designer, ensure that **Buttons** is selected. Select the **Text** icon, drag it to the right of **P2_SEARCH** in the Report 1 region in the Page Layout area, and drop it.
5. In the properties on the right of Page Designer, enter `GO` for **Button Name** and **Label**. In the Appearance section, change **Hot** to **Yes**. Click **Template Options**, change **Size** to **Large** and **Spacing Top** to **Large**, and then click **OK**. In the Behavior section, change **Action** to **Defined by Dynamic Action**.
6. In the tree view on the left of Page Designer, select **P2_SEARCH**, hover over it, and right-click the mouse to display the context-sensitive menu. Choose **Create Dynamic Action**.
7. In the attributes on the right of Page Designer, change the dynamic action **Name** to `Refresh Report with Search Term`. In the When section, change **Event** to **Click**; change **Selection Type** to **Button**; and for **Button**, select the newly created **Go** button.
8. In the tree view on the left of Page Designer, select **True** for the dynamic action (it will be **Show**). Change **Action** to **Refresh**; change **Selection Type** to **Region**; and for **Region**, select **Report 1**.
9. In the tree view on the left of Page Designer, under the **Report 1** report region, expand the parameter list by clicking the **>** to the left of **Parameters**.
10. Select the **term** parameter. In the properties on the right of Page Designer, change **Type** to **Item**. For **Item**, enter `P2_SEARCH`, which is the name of your search term item.
11. Select the **Report 1** report region in the tree view on the left of Page Designer. In the properties on the right, for **Page Items to Submit**, enter `P2_SEARCH`. This will

ensure that the value of the P2_SEARCH text item will be submitted to the Oracle APEX engine when the region is refreshed.

12. Click the **Run Application** icon on the upper right to save all your changes and run your Oracle APEX application.

Your application should look similar to **Figure 4**. Enter any search term you wish, such as `tupac` or your favorite music artist, and click **Go**. You now have an application that uses the iTunes Search API, which makes a request via HTTP, returns the results in JSON, and parses the results before further manipulating the results via SQL postprocessing.

Figure 4: The application with a search field for dynamic searching of the iTunes music video catalog



CONCLUSION

Oracle Application Express enables you to create applications that integrate data from local Oracle Database tables and remote datasources. These remote datasources can be simple HTTP data feeds, as in this article, but they can also be OAuth2 or Basic Authentication–protected resources, REST services provided by Oracle REST Data Services, or even Oracle cloud applications (SaaS) REST services. Oracle APEX also provides, uniquely, the ability to manipulate the data from remote datasources with industry-standard SQL. The full breadth of Oracle Database SQL functionality is at your disposal to properly present the results of remote data. All of this is accomplished with no code other than a simple SQL statement. Now *that's* low code!

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

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

DOWNLOAD Oracle Application Express 18.1.

REQUEST a free Oracle Application Express workspace.

READ more about Oracle Application Express.

what's new in Oracle Application Express 18.1.

TRY Oracle Database Cloud Service.

BY CONNOR McDONALD

ORACLE DATABASE 12c RELEASE 2

Old Dog, New Tricks, Part 2

Here's a new SQL syntax for hierarchy processing.

In my previous article, I described ways database professionals have historically been working with hierarchy structures inside Oracle Database. The CONNECT BY and START WITH clauses have been around since Oracle Database version 2, and using them is the most common way of coding hierarchical SQL processing. However, ANSI-standard SQL does not recognize CONNECT BY and START WITH—these are exclusively Oracle-specific functions. The ANSI standard employs an alternative mechanism for hierarchical data relationships, but before I tackle that, I'll set the foundation by returning to some basic principles of navigating a hierarchy.

FIRST PRINCIPLES

To consider hierarchy processing from the standpoint of first principles, I'll take a simple hierarchical query against the EMP table with the output left-padded to demonstrate the depth of employees in the hierarchy, as shown in **Listing 1**.

Listing 1: A simple hierarchical query

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

ENAME
_____________________________

 KING
   BLAKE
     JAMES
     ALLEN
     WARD
  CLARK
     MILLER
```

In this query, I start with employees who have no manager (MGR is NULL) and then apply a CONNECT BY clause to traverse the hierarchy. But what if no explicit hierarchical syntax was possible? How would a hierarchy be traversed with standard SQL commands? To achieve this, I can break the query into its constituent parts.

First I need to extract the rows where the manager is not specified, as shown in **Listing 2**.

Listing 2: First level of the hierarchy

```
SQL> select empno, ename, mgr
  2  from   emp
```

```
  3  where  mgr is null;

     EMPNO ENAME                      MGR
__________ _____________________ __________

      7839 KING
```

I have now discovered the first, or top, level of the hierarchy. Now I must find all employees whose manager is KING. I could construct a new query that explicitly references WHERE MGR = ‘ KING’, but in general what I need is the *output* from the query in **Listing 2** to become an *input* into a query to derive the next level of the hierarchy. I can do that with a nonhierarchical query by using a standard join. In **Listing 3**, I nest the initial query locating those rows where the manager is null as an inline view into a wrapping query that joins the MGR column to the EMPNO column to get the employees in the next level in the hierarchy. This yields the three employees—JONES, BLAKE, and CLARK—who report to KING.

Listing 3: Second level of the hierarchy via a join

```
SQL> select e2.empno, e2.ename, e2.mgr
  2  from   emp e2,
  3         ( select empno, mgr
  4           from   emp
  5           where  mgr is null ) inner
  6  where  e2.mgr = inner.empno;
```

```
     EMPNO ENAME                      MGR
__________ _____________________ __________
      7566 JONES                      7839
      7698 BLAKE                      7839
      7782 CLARK                      7839
```

Now the second level in the hierarchy has been established, and it returns a set of rows that is the input for deriving the *third* level of the hierarchy. I take the query from **Listing 3**, nest it again as an inline view into a wrapper query, and join MGR to the EMPNO column, as shown in **Listing 4**.

Listing 4: Third level of the hierarchy via a join

```
SQL> select e3.empno, e3.ename, e3.mgr
  2  from   emp e3,
  3         ( select e2.empno, e2.ename, e2.mgr
  4           from   emp e2,
  5                  ( select empno, mgr
  6                    from   emp
  7                    where  mgr is null ) inner
  8           where  e2.mgr = inner.empno ) inner
  9  where  e3.mgr = inner.empno;

     EMPNO ENAME                      MGR
__________ _____________________ __________
      7902 FORD                       7566
      7788 SCOTT                      7566
```

```
7900 JAMES                7698
7844 TURNER               7698
7654 MARTIN               7698
7521 WARD                 7698
7934 MILLER               7782
```

Finally, because there are four levels in the employee hierarchy, I can nest the query from **Listing 4** to reveal the people who sit at the very bottom layer of the hierarchy, as shown in **Listing 5**.

Listing 5: Fourth (and last) level of the hierarchy via a join

```
SQL> select e4.empno, e4.ename, e4.mgr
 2  from   emp e4,
 3         (  select e3.empno, e3.ename, e3.mgr
 4            from   emp e3,
 5                   ( select e2.empno, e2.ename, e2.mgr
 6                     from   emp e2,
 7                            ( select empno, mgr
 8                              from   emp
 9                              where  mgr is null ) inner
10                     where  e2.mgr = inner.empno ) inner
11           where  e3.mgr = inner.empno ) inner
12  where  e4.mgr = inner.empno;
```

```
     EMPNO ENAME                      MGR
__________ _____________________ __________

      7876 ADAMS                      7788
```

Obviously, this approach is untenable in general. The number of levels in the hierarchy is never known in advance, so it is never known how many nested inline views are required to entirely traverse the hierarchy. However, this approach does reveal that processing a hierarchy via SQL is a *recursive* operation, in that a set of output data from one query becomes the input into a similar wrapping query. This leads naturally to the new form of hierarchy processing that is available from Oracle Database 11g Release 2 onward—the ANSI-standard SQL approach of using recursive common table expressions (CTEs).

SIMPLER SQL WITH WITH

Before I describe recursive CTEs, it is worth covering the simpler case of *nonrecursive* CTEs.

First a *virtual table* consisting of a query expression can be defined with the WITH clause, enabling that table to be referenced throughout the remainder of the query. **Listing 6** shows an example of the WITH clause that refers to a virtual table named LAST_HIRE, which is the query expression for deriving the most recent hiring date per department in the EMP table.

Listing 6: Basic usage of the WITH clause

```
SQL> WITH last_hire AS
  2  ( select deptno, max(hiredate)
  3    from   emp
```

```
  4    group by deptno
  5  )
  6  select * from last_hire;

    DEPTNO MAX(HIRED
__________ _________

        30 03-DEC-81
        20 12-JAN-83
        10 23-JAN-82
```

At first glance, the WITH clause may seem like merely a waste of keystrokes, but its true benefit can be seen when the functional requirements of a query become more complex. Consider this business requirement given to a developer—

"Find the total salary paid by each department,
get the average of these totals, and
list those departments above that average."

—which needs to be expressed with a SQL statement. Rather than tackling the entire task holistically, developers using the WITH clause can tackle the task in smaller, more manageable pieces, much as they would in their 3GL application code. I can start with the first part of the requirement—

"Find the total salary paid by each department"

—and construct an appropriate query for just that component, as shown in **Listing 7**.

Listing 7: Part 1, finding the total

```
SQL> SELECT dname, SUM(sal) dept_sal
  2  FROM   emp e, dept d
  3  WHERE  e.deptno = d.deptno
  4  GROUP  BY dname;
```

Using a WITH statement, I can now assume the existence of the **Listing 7** query as if it were a table in the database. Given a virtual table containing that data (dept_salaries), I can then construct a query to handle the second part of the requirement—

"get the average of these totals"

—as shown in **Listing 8**.

Listing 8: Part 2, getting the average

```
SQL>  WITH dept_salaries AS (
  2         SELECT dname, SUM(sal) dept_sal
  3         FROM emp e, dept d
  4         WHERE e.deptno = d.deptno
  5         GROUP BY dname),
  6   SELECT AVG(dept_sal) avsal
  7   FROM  dept_salaries;
```

The final part of the requirement—

“list those departments above that average”

—implies that the **Listing 8** result containing average departmental salaries needs to be referenced in a subsequent SQL statement, so it makes sense to also assign that query expression a virtual table name (avg_sal), as shown in **Listing 9**.

Listing 9: Creating another virtual table name

```
SQL>  WITH dept_salaries AS (
  2         SELECT dname, SUM(sal) dept_sal
  3         FROM emp e, dept d
  4         WHERE e.deptno = d.deptno
  5         GROUP BY dname),
  6     avg_sal AS ( SELECT AVG(dept_sal) avsal
  7                  FROM dept_salaries);
```

With the two query expressions aliased, I can complete the three-step task by joining the two virtual tables to find the appropriate departments, as shown in **Listing 10**.

Listing 10: Part 3, listing the departments

```
SQL>  WITH dept_salaries AS (
  2         SELECT dname, SUM(sal) dept_sal
  3         FROM emp e, dept d
  4         WHERE e.deptno = d.deptno
  5         GROUP BY dname),
  6     avg_sal AS ( SELECT AVG(dept_sal) avsal
```

```
 7                   FROM dept_salaries)
 8   SELECT * FROM dept_salaries d, avg_sal a
 9   WHERE d.dept_sal > a.avsal
10   ORDER BY d.dname;
```

Building SQL in piecemeal fashion with CTEs is a great way of solving complicated query problems. It also assists in the maintenance of the SQL, because it reveals the thought process by which the original author of the SQL statement tackled the task.

RECURSIVE CTEs

Oracle Database 11g Release 2 and later releases support an extension to CTEs. The WITH syntax can also be recursive—that is, a table you are defining with a query expression using the WITH clause can contain references to the table currently being defined. Earlier I described how navigating hierarchical data can be tackled as a series of self-referencing queries—recursive CTEs are a natural solution for this task. To demonstrate the structure of the new hierarchical syntax, I'll return to the first principles I used for navigating the employer hierarchy via its component parts.

Listing 11 shows the definition of a table called EACH_LEVEL, the name indicating that it will navigate each level of the hierarchy. The definition is a UNION ALL of two queries. The first part of the UNION ALL is the entry point into the hierarchy, equivalent to START WITH in the conventional hierarchical SQL syntax style, retrieving all employees who have no manager. The second query following the UNION ALL is where the recursive nature of the WITH statement becomes apparent. I am querying the EMP table and then joining back to the EACH_LEVEL table, even though I am currently defining the EACH_LEVEL table. The recursive nature demonstrates a concept similar to when I was repeatedly creating and running inline queries. In this case,

EACH_LEVEL is referring back to itself, so each time I encounter the recursive part of the definition, I navigate recursively through the structure, drilling deeper into the hierarchy.

Listing 11: The recursive CTE

```
SQL> with EACH_LEVEL (empno, name, mgr) as
  2  ( --
  3    -- start with
  4    --
  5    select empno, ename, mgr
  6    from   emp
  7    where  mgr is null
  8    --
  9    -- connect by
 10    --
 11    union all
 12    select emp.empno, emp.ename, emp.mgr
 13    from   emp, EACH_LEVEL
 14    where  emp.mgr = each_level.empno
 15  )
 16  select *
 17  from   each_level;

     EMPNO NAME                 MGR
__________ _______________ __________
      7839 KING
```

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

The two sides of the UNION ALL are analogous to two clauses in the conventional hierarchical syntax, in that an entry point into the hierarchy is defined, followed by the definition of how to use CONNECT BY or recursively traverse the hierarchy. It is the developer who now takes control of the conventional pseudofunctions available in the conventional hierarchical syntax. For example, if I need to print out the depth of the hierarchy as I would typically do with the LEVEL pseudofunction, I can add a column to increment the level each time I recursively pass through EACH_LEVEL. **Listing 12** shows this in action. The upper part of the UNION ALL must by definition be level = 1, because it is the entry point, and as the recursive statement under the UNION ALL is executed, the level increments by 1 each time I recursively navigate through the hierarchy.

Listing 12: Incrementing the level recursively

```
SQL> with each_level (empno, name, mgr, rlevel) as
  2  ( select empno, ename, mgr, 1 rlevel
  3    from   emp
  4    where  mgr is null
  5    union all
  6    select emp.empno, emp.ename, emp.mgr, rlevel+1
  7    from   emp, each_level
  8    where  emp.mgr = each_level.empno
  9  )
 10  select * from each_level;
```

```
    EMPNO NAME                 MGR     RLEVEL
_________ ________________ _________ __________
     7839 KING                              1
     7566 JONES                7839          2
     7698 BLAKE                7839          2
     7782 CLARK                7839          2
     7499 ALLEN                7698          3
     ...
```

Unfortunately, I cannot use the name LEVEL for my virtual table column, because it is reserved for its built-in predecessor. Similarly, to access the SYS_CONNECT_BY_PATH function to concatenate children in their hierarchy with their respective parent rows, I can build it myself by using the standard concatenation operators in the recursive definition. At the lowest level of the hierarchy, an employee name concate-

nation is just the employee name itself, but within the recursive definition under the UNION ALL, I take the existing concatenation of employee names so far and append the employee name I'm currently navigating. This builds the concatenated list of employee names just as the SYS_CONNECT_BY_PATH function does, as demonstrated in **Listing 13**.

Listing 13: Building the concatenated list of employee names

```
SQL> with each_level (empno, name) as
  2  ( select empno, ename from  emp
  3    where  mgr is null
  4    union all
  5    select e.empno,
  6           each_level.name||'-'||e.ename
  7    from   emp e, each_level
  8    where  e.mgr = each_level.empno
  9  )
 10  select empno, name from each_level;

    EMPNO NAME
_________ ____________________________________

     7839 KING
     7566 KING-JONES
     7698 KING-BLAKE
     7782 KING-CLARK
     7499 KING-BLAKE-ALLEN
```

```
7521 KING-BLAKE-WARD
 ...
```

EXTENSIONS TO THE SYNTAX

In the same way that, by default, a conventional syntax hierarchical query will fail if a cycle is detected, a recursive WITH definition will also fail if a cycle is detected, albeit with a different error code. **Listing 14** shows the recursive WITH definition attempt and error.

Listing 14: Recursive WITH definition detecting cycle and returning error message

```
SQL> with each_level (empno, name, mgr) as
  2  ( select empno, ename, mgr
  3    from   emp
  4    where  ename = 'KING'
  5    union all
  6    select emp.empno, emp.ename, emp.mgr
  7    from   emp, each_level
  8    where  emp.mgr = each_level.empno
  9  )
 10  select *
 11  from   each_level;

ERROR:
ORA-32044: cycle detected while executing recursive WITH query
```

You can detect cyclic relationships and avoid errors by extending the definition with the CYCLE clause, as shown in **Listing 15**. The behavior is slightly different from the way cycles are handled in the conventional hierarchical syntax, in that cyclic rows are *still* presented in the result set but a new column is added with a single character flag indicating whether this row is a member of a cyclic relationship.

Listing 15: Recursive WITH, cycle, CYCLE clause, and new column

```
SQL> with each_level (empno, name, mgr) as
  2  ( select empno, ename, mgr from emp
  3    where  ename = 'KING'
  4    union all
  5    select emp.empno, emp.ename, emp.mgr
  6    from   emp, each_level
  7    where  emp.mgr = each_level.empno )
  8  CYCLE mgr SET is_cycle TO 'Y' DEFAULT 'N'
  9  select * from each_level;

    EMPNO NAME              MGR IS_CYCLE
_________ _________ __________ __________
     7839 KING            7499 N
     7566 JONES           7839 N
     7521 WARD            7698 N
     7839 KING            7499 Y
     7876 ADAMS           7788 N
     ...
```

The recursive CTE syntax allows for some additional facilities in terms of how the hierarchy should be traversed. You can navigate any hierarchy either by taking "slices" in a top-down approach or by simply working down along a branch until all the leaves are exhausted and then working back up through the branches in the same way someone might navigate through a maze. Each mechanism is available to developers using the SEARCH clause in the recursive CTE definition. I can search breadth-first or depth-first, as shown in **Listing 16**, with the ordering of the results achieved by the SET clause, which nominates a new column that yields a sequence number defining the order in which the rows should be presented.

Listing 16: Recursive CTE, SEARCH, and SET clause

```
SQL> with each_level (empno, name, hiredate, mgr) as
  2  ( select empno, ename, hiredate, mgr from emp
  3    where  ename = 'KING'
  4    union all
  5    select e.empno,
  6      each_level.name||'-'||e.ename, e.hiredate, e.mgr
  7    from   emp e, each_level
  8    where  e.mgr = each_level.empno )
  9  SEARCH BREADTH FIRST BY HIREDATE SET IDX
 10  select name, hiredate, idx  from each_level;

NAME                                HIREDATE          IDX
___________________________________ _________ ___________

KING                                17-NOV-81           1
KING-JONES                          02-APR-81           2
```

```
KING-BLAKE                      01-MAY-81          3
KING-CLARK                      09-JUN-81          4
KING-BLAKE-ALLEN                20-FEB-81          5
KING-BLAKE-WARD                 22-FEB-81          6
...
KING-JONES-FORD-SMITH           17-DEC-80         13
KING-JONES-SCOTT-ADAMS          23-MAY-87         14

SQL> with each_level (empno, name, hiredate, mgr) as
  2  ( select empno, ename, hiredate, mgr from emp
  3    where  ename = 'KING'
  4    union all
  5    select e.empno,
  6      each_level.name||'-'||e.ename, e.hiredate, e.mgr
  7    from   emp e, each_level
  8    where  e.mgr = each_level.empno )
  9  SEARCH DEPTH FIRST BY HIREDATE SET IDX
 10  select name, hiredate, idx  from each_level;

NAME                            HIREDATE         IDX
_______________________________ _________ __________

KING                            17-NOV-81          1
KING-JONES                      02-APR-81          2
KING-JONES-FORD                 03-DEC-81          3
KING-JONES-FORD-SMITH           17-DEC-80          4
```

```
KING-JONES-SCOTT                19-APR-87          5
KING-JONES-SCOTT-ADAMS          23-MAY-87          6
KING-BLAKE                      01-MAY-81          7
KING-BLAKE-ALLEN                20-FEB-81          8
...
```

ALTERNATIVE USE CASES

The ability to run operations recursively via a single SQL statement gives rise to the ability to do recursive problem solving even when the data is not necessarily hierarchical in nature. If I need to iterate through some data an unknown number of times, recursion is a very effective tool for achieving this. Here is an example of using recursive CTEs when it does not immediately appear to be a hierarchical problem. I have a table called MESSAGES that includes some sentences containing names of my fellow developer advocates and myself. I would like to replace those names with the other people's respective Twitter handles. To do that, I have a lookup table called TWITTER_HANDLES with the terms I will replace and the appropriate Twitter handles to replace them, as shown in **Listing 17**.

Listing 17: Looking up Twitter handles

```
SQL> select * from messages;

TXT
--------------------------------------------------------------------------------
I caught up with Connor and Maria Colgan today. They have taken over AskTOM
for Oracle Developers
```

```
SQL> select * from twitter_handles;

  ID  TERM                          HANDLE
____  ____________________________  _________________

   1  Connor McDonald               @connor_mc_d
   2  Connor                        @connor_mc_d
   3  Maria Colgan                  @sqlmaria
   4  Oracle Developers             @otndev
   5  Oracle                        @oracle
   6  AskTOM                        @oracleasktom
```

This means that for each of the rows in TWITTER_HANDLES, I need to continually iterate through all the rows of data in my MESSAGES table, each time replacing a found term with its appropriate Twitter handle until all the terms have been exhausted, as shown in **Listing 18**.

Listing 18: Selecting tweets and replacing names with Twitter handles

```
SQL> with
  2    tweetised(ind,tweet_txt)  as
  3  (
  4    select 1 ind, txt tweet_txt
  5    from   messages
  6    union all
  7    select ind+1, replace(tweet_txt,term,handle)
  8    from   tweetised, twitter_handles
  9    where  ind = id
```

```
10  )
11  select * from tweetised;

     IND  TWEET_TXT
________  ____________________________________________________________________________

       1  I caught up with Connor and Maria Colgan today. They have taken over
          AskTOM for...

       2  I caught up with Connor and Maria Colgan today. They have taken over
          AskTOM for...

       3  I caught up with @connor_mc_d and Maria Colgan today. They have taken over
          AskTOM for...

       4  I caught up with @connor_mc_d and @sqlmaria today. They have taken over
          AskTOM for ...

       5  I caught up with @connor_mc_d and @sqlmaria today. They have taken over
          AskTOM for @otndev

       6  I caught up with @connor_mc_d and @sqlmaria today. They have taken over
          AskTOM for @otndev

       7  I caught up with @connor_mc_d and @sqlmaria today. They have taken over
          @oracleasktom ...
```

I am generating multiple rows of output for each single row of data in my messages table, with each new row containing an additional term replaced with its associated Twitter handle. Hence, simply by picking up the very last row, using a FETCH FIRST clause, I will display the result I need. I recursively run a REPLACE operation, using the recursive nature of the WITH statement, as shown in **Listing 19**.

Listing 19: Picking the last row with FETCH FIRST

```
SQL> with
  2  tweetised(ind,tweet_txt)  as
  3  (
  4    select 1 ind, txt tweet_txt
  5    from   messages
  6    union all
  7    select ind+1, replace(tweet_txt,term,handle)
  8    from   tweetised, twitter_handles
  9    where  ind = id
 10  )
 11  select * from tweetised
 12  order by ind desc
 13  fetch first 1 row only;

       IND  TWEET_TXT
__________  ____________________________________________________________

         7  I caught up with @connor_mc_d and @sqlmaria today. They...
```

SUMMARY

Oracle Database continues to support the conventional START WITH and CONNECT BY syntax, but don't be afraid to expand your SQL toolkit by utilizing recursive common table expressions to achieve the same results. As shown in this article, you might also find uses for these expressions beyond standard hierarchical relationships.

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

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about hierarchical SQL processing.

DOWNLOAD Oracle Database 12c.

TRY Oracle Database cloud services.

User Groups Prepare for Oracle OpenWorld 2018

Here's what four user groups expect from and bring to Oracle OpenWorld San Francisco this year. **BY LESLIE STEERE**

From its very beginnings as Relational Software Inc.'s "first international user group meeting" in 1982, attended by a few dozen customers representing fewer than 20 companies, Oracle OpenWorld has grown to become the industry's most important business and technology conference for Oracle users and partners, serving thousands of educational sessions to tens of thousands of attendees every year. Although Oracle has managed the event since 1996, user group participation remains a critical part of the conference. *Oracle Magazine* asked

Oracle OpenWorld comes to San Francisco October 22–25 this year, and as always, user groups are a critical element to the conference, bringing expert presentations, interested attendees, and successful customers to share experiences, best practices, and tips and tricks.

leaders of some key user groups what their members most wanted to gain from Oracle OpenWorld in San Francisco this year (October 22–25) and what their user groups were bringing to the conference. Here's a selection from their responses.

Gary Gordhamer, Director of Education Independent Oracle Users Group

Because Oracle technology is the core of hundreds of thousands of critical business systems, it's essential to integrate and upgrade in a proven and reliable manner. With that in mind, for Oracle OpenWorld 2018, I believe Independent Oracle Users Group (IOUG) members will be looking at three hot-topic areas: how to adopt new features in Oracle Database 12c Release 2 into production running systems, what to look forward to in Oracle Database 18c and how to start preparing to use those features in their next projects, and where the cloud can continue to enhance existing business systems as well as enable new business outcomes.

One of IOUG's goals for its membership is to help develop not just technical skills but also presentation and leadership skills. So it's great to see that more than 20 of our members are scheduled to speak at Oracle OpenWorld 2018. Anuj Mohan will discuss Oracle Database 18c new features and how they build on what Oracle Database 12c Releases 1 and 2 already delivered, Arup Nanda will help prepare DBAs for their future role in cloud-based organizations with a discussion of DBA best practices for the cloud, and Michael Gangler will present real-world experiences in his session on upgrading to Oracle Database 12.2 in the cloud. That is just a taste of the great content we're bringing to the conference this year.

Christine Hipp, President Oracle Applications Users Group

The Oracle Applications Users Group (OAUG) originated and grew for many years largely to support customers

running Oracle E-Business Suite. In many cases, this contingent of customers is continuing to use Oracle E-Business Suite while actively planning for and beginning to adopt cloud solutions. We also have members with BI/analytics solutions that are in similar positions. As a result, our members are very interested in seeing hybrid cloud sessions at Oracle OpenWorld.

"To do their jobs better, members need to know how to balance the foundations they have in place with the effort to embrace promising new solutions."

—*Christine Hipp*, *President, Oracle Applications Users Group*

While some OAUG members maintain a portion of their Oracle apps on premises, a lot of them are venturing into cloud solutions. To do their jobs better, members need to know how to balance the foundations they have in place with the effort to embrace promising new solutions. They want to hear from other customers who've "been there, done that." Highlighting "Real Customers, Real Stories" right up front in the Session Catalog makes it clear that Oracle OpenWorld is providing a platform where customers can learn from each other how to plan and streamline implementations.

We're pleased to be part of "Real Customers, Real Stories," with our members sharing how they evaluated and planned for adopting cloud solutions and overcame challenges to get there. One of our member-presented sessions, for example, looks at a migration from Oracle E-Business Suite on premises to the cloud. Another session presents how an OAUG member organization fully adopted out-of-the-box best practices with Oracle Enterprise Planning and Budgeting Cloud Service. You can view the full list of OAUG's member-presented sessions on the OAUG website.

Ann Hansen, President
Carrie Hollack, Board of Directors
Oracle HCM Users Group

Members of Oracle HCM Users Group (OHUG) are primarily interested in all human capital management (HCM) and analytics products. We also want to understand the integration points between HCM and enterprise resource planning (ERP) products. Lastly, members using Oracle's PeopleSoft and Oracle E-Business Suite on premises are interested in cloud options for hosting.

Oracle's focus at this year's Oracle OpenWorld on the changing workforce is also particularly relevant to OHUG members, who view their HR systems as business enablers. Our members rely on Oracle's HR applications to meet rapidly changing business needs, including a more collaborative and dynamic environment.

The customer success stories at the conference are also a big draw. Many of our members are focused on differentiating capabilities such as attracting, retaining, engaging, and growing talent, whereas others are constantly adapting to international compliance requirements. And all OHUG members are keenly interested in the new functionality that Oracle continues to roll out in its HCM cloud products as well as its on-premises HR applications.

Of course, OHUG member organizations will be at Oracle OpenWorld to share their own cloud experiences as well. Whether you're interested in an executive overview of Oracle Human Capital Management Cloud (Oracle HCM Cloud) transformation, a tactical approach to configuring Oracle HCM Cloud security, a how-to on quickly standing up real-time talent insight, or tips on reducing customizations within Oracle's PeopleSoft applications, you'll find OHUG members there to share advice and best practices.

Jon Vaughn, CEO
Quest Oracle Community

Overwhelmingly, Quest members are interested in hearing real stories from customers about product implementation and use. They want to hear about what went well as well as what pain points customers are experiencing with their products.

"Quest will be presenting multiple stories from Oracle customers about their journey to Oracle Cloud, with special focus on Oracle HCM Cloud, Oracle EPM Cloud, and Oracle ERP Cloud."

—Jon Vaughn, *CEO, Quest Oracle Community*

Quest members who are using Oracle Cloud products are looking for best practices and postimplementation resources from those who are on Oracle Cloud. Members evaluating Oracle Cloud want to hear from current Oracle Cloud users about what life looks like once Oracle Cloud is implemented and what benefits can be quickly realized.

At this year's conference, Quest will be presenting multiple stories from Oracle customers about their journey to Oracle Cloud, with special focus on Oracle HCM Cloud, Oracle Enterprise Performance Management Cloud (Oracle EPM Cloud), and Oracle Enterprise Management Cloud (Oracle ERP Cloud). Companies including Sinclair Broadcast Group, Yum! Brands, Ohio National Insurance Company, Harbison Walker International, Wagstaff, and Intermountain Health Systems bring with them a great depth of experience that comes from their Oracle Cloud journeys, and we think other customers will benefit greatly from that experience, helping them benchmark their own companies' progress in their adoption of Oracle Cloud.

We are also presenting sessions focused on the various options Oracle customers have in their utilization of

Oracle Cloud, whether that be with IaaS, PaaS, or SaaS, and we will be sharing findings from our annual community surveys for Oracle Cloud and Oracle's JD Edwards and PeopleSoft products as well.

Of course, Oracle OpenWorld wouldn't be Oracle OpenWorld without some great parties, and we are hosting several receptions where customers can connect with us and with one another.

Leslie Steere is editor at large for Oracle Content Central. She has more than 30 years of journalism and marketing content experience.

PHOTOGRAPHY BY **TUE NAM TON**

NEXT STEPS

REGISTER for Oracle OpenWorld.

REGISTER for Oracle Code One.