

ORACLE

MAGAZINE

MAY/JUNE 2019

EMERGING TECH

THE FUTURE IS HERE

Learn how businesses are using AI, blockchain, and IoT today to reimagine their processes and delight their customers



ONLY THE BEST
BUSINESS DATA

TRACKING CONFLICT
MINERALS

THE IoT IS IN THE
APPLICATIONS

ORACLE®

Vertex + Oracle

Global Tax Solution Partners



Accurate Calculation. Seamless Integration.

A powerful solution providing
an end-to-end tax automation
solution for your business.

[Learn More](#)

 VERTEX®

 ORACLE®

Platinum
Partner
Cloud Standard



UP FRONT

6 FROM THE EDITOR

**Emerging Tech:
Data-Driven**

AI, blockchain, and IoT are more powerful than the data that feeds them.

BY TOM HAUNERT

8 MASHUP

Intelligent Trekking

Gadgets for going off-grid, apps to ease your travels, and more

BY LESLIE STEERE

11 INTERVIEW

The IoT Is in the Applications

Devices connect to multiple enterprise applications to deliver business value.

BY TOM HAUNERT



11 Interview

FEATURES

31 Only the Best Business Data

With Oracle DataFox, business data is accurate, fresh, dynamic—and AI-powered. **BY ALAN ZEICHICK**

36 Tracking Conflict Minerals

Supply chain company ensures ethical practices with Oracle Blockchain Platform. **BY MIKE FADEN**

 **COMMUNITY**


16 ORACLE GROUNDBREAKER AMBASSADOR
Lighting the Way

An educator's suggestion put Oracle Groundbreaker Ambassador Mohamed Taman on the path to success. **BY BOB RHUBART**

20 DEVELOPER PRODUCTIVITY
Use a Pencil

Java Champion Mani Sarkar's learning hacks

BY ALEXANDRA WEBER MORALES

24 DATA TYPES
Emerging Tech, Emerging Opportunities

Blockchain, IoT, and AI are here, but how do data experts use them in new business projects? **BY JEFF ERICKSON**

27 PEER-TO-PEER
Love at First Byte

The invisible line between work and play and the joys of finding the right tool for the job **BY BLAIR CAMPBELL**


27 Peer-to-Peer
 **TECHNOLOGY**

42 APPLICATION DEVELOPER
Escape Room

How to unlock users who get stuck in a chatbot conversion

BY FRANK NIMPHIUS

61 OPEN SOURCE
Getting Started with Autonomous

Connect to Oracle Autonomous Transaction Processing with multiple programming languages.

BY BLAINE CARTER

76 PL/SQL
Easy Initializing for Records and Arrays

Use the new qualified expressions to create functions when and where you need them.

BY STEVEN FEUERSTEIN

90 MACHINE LEARNING
Using Oracle Machine Learning Notebooks

Set up and run machine learning in Oracle Autonomous Data Warehouse.

BY ARUP NANDA

106 ETL
A Higher-Level Perspective on SQL Tuning, Part 2

Identifying and diagnosing poorly performing SQL

BY CONNOR McDONALD

 **COMMENT**

117 IN THE FIELD
Inspiration, Connection, and Community

UKOUG leaders pay it forward with full range of tech topics and support.

BY LESLIE STEERE

ORACLE MAGAZINE

EDITORIAL

Editor in Chief [Tom Haunert](#)

Managing Editor Jan Rogers

Editorial Director Robert Preston

Contributing Editors and Writers Blair Campbell, Leslie Steere

Copy Editors Eva Langfeldt, Karen Perkins

DESIGN

Vice President, Brand Creative Francisco G Delgadillo

Design Director Richard Merchán

Senior Designer Arianna Pucherelli

Senior Production Manager Sheila Brennan

Designer Jaime Ferrand

Production Designer Kathy Cygnarowicz

PUBLISHING

Publisher and Audience Development Director [Karin Kinnear](#)

Audience Development Manager Jennifer Kurtz

ADVERTISING SALES

[Tom Cometa](#) +1.510.339.2403

Mailing-List Rentals Contact your sales representative

EDITORIAL BOARD

Ian Abramson, Karen Cannell, Andrew Clarke, Chris Claterbos, Karthika Devi, Kimberly Floss, Kent Graziano, Taqi Hasan, Tony Jambu, Tony Jedlinski, Ari Kaplan, Val Kavi, John King, Steve Lemme, Carol McGury, Sumit Sengupta, Jonathan Vincenzo, Dan Vlamis

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@omedacom.

Copyright © 2019, Oracle and/or its affiliates. All Rights Reserved. No part of this publication may be reprinted or otherwise reproduced without permission from the editors. ORACLE MAGAZINE IS PROVIDED ON AN "AS IS" BASIS. ORACLE EXPRESSLY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED. IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING FROM YOUR USE OF OR RELIANCE ON ANY INFORMATION PROVIDED HEREIN. The information is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Address the ELEPHANT IN THE ROOM

Bad address data costs you money, customers and insight.

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

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

+ JD Edwards® EnterpriseOne

+ Oracle® E-Business Suite

+ Oracle® Peoplesoft®

**See the Elephant in Your Business -
Name it and Tame it!**



melissa

www.Melissa.com | 1-800-MELISSA

Free API Trials, Data Quality Audit & Professional Services.



Tom Haunert



Emerging Tech: Data-Driven

AI, blockchain, and IoT are more powerful than the data that feeds them.

Businesses are turning to emerging technologies—including artificial intelligence/machine learning, blockchain, the Internet of Things, and digital assistants—to put their abundance of data to better use.

For example, they're using machine learning, whose learning improves the more data you feed it, to get a more complete understanding of their customers, finance operations, and supply chain partnerships via cloud-based sales, marketing, ERP, and other applications that build in the technology. That learning turns into better, more efficient business processes.

For those who think their business isn't really creating or consuming

enough information to benefit from machine learning, their turn is coming soon. Even the smallest businesses are starting to derive value from data they acquire from *outside* their organizations and traditional enterprise boundaries.

EMERGING ARTICLES

This issue's three cover stories focus on the power of data-driving emerging technologies to reshape a range of businesses. "[Only the Best Business Data](#)" describes how organizations are using Oracle's DataFox AI technology to get up-to-date information and insights into as many as 3.7 million companies—representing many thousands of poten-

tial sales leads for customer-facing applications. In “[Tracking Conflict Minerals](#),” we show how supply chain company Circulor is building applications on top of Oracle Blockchain Platform that track the origins of gold, cobalt, and tantalum from the mine to the end user, ensuring the ethical origins of those materials. And in “[The IoT Is in the Applications](#),” Oracle Group Vice President Bhagat Nainani makes the case for taking a business-outcome-based approach—rather than a device-first approach—with your organization’s IoT projects.

In his ongoing *Oracle Magazine*

digital assistant column, this one titled “[Escape Room](#),” Oracle Master Principal Product Manager Frank Nimphius provides help for bot users who can’t find the answers they’re looking for.

For more stories on AI, blockchain, IoT, and digital assistants, check out our [emerging technologies archive](#)—and come back often for updates.



Tom Haunert,
Editor in Chief

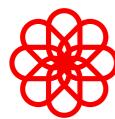
Emails and posts received by *Oracle Magazine* and its staff may be edited for length and clarity and may be published in any medium. We consider any communications we receive publishable.

PHOTOGRAPH BY BOB ADLER/GETTY IMAGES

NEXT STEPS

LEARN more about emerging technologies.

LEARN more about modern application development.



Intelligent Trekking

Gadgets for going off-grid, apps to ease your travels, and more



Travelmate S Autonomous Suitcase

How many times have you wished that your suitcase would heel like a well-trained dog? Travelmate's fully autonomous suitcase unleashes you from the bag drag by pairing with your smartphone so it follows you around—and enables you to track its location. It also weighs your contents with a built-in scale, enables you to charge almost any device via its USB port, and uses a TSA-standard Bluetooth-enabled lock. It complies with all carry-on specifications. US\$1,099. [Travelmate](#)

PHOTOGRAPHS COURTESY OF TRAVELMATE, GOTENNA



goTenna Mesh

Stay connected with your fellow trekkers the next time your adventures take you off the grid. With the goTenna Mesh, you and your companions can text each other and share your locations without cell or Wi-Fi service as long as you stay within about a four-mile point-to-point range. The device pairs with any Android or iOS smartphone and creates mobile mesh networks that enable you to stay connected in remote areas or when cell networks are overwhelmed. US\$179. [goTenna](#)

Purpose-Driven Travel—and Lots of Tech

Undersea, outer space, and earthbound experiences on the agenda

Booking.com's travel predictions for 2019, based on information from 21,500 survey respondents from around the world and insights from more than 163 million verified guest reviews, show that a majority of travelers (54%) would consider a purpose-driven volunteering trip and that an impressive 86% were willing to spend time on activities that offset the environmental impact of their stay. Other insights? Tech travel innovations incorporating AI, VR, and speech recognition continue to excite—but ease of use is key. Notable interests include



Source: ["Booking.com Reveals 8 Travel Predictions for 2019"](#)

DO YOU SPEAK TECH? QUIZ YOURSELF!

1. What is dark data?

- A. Another term for metadata—you can't see it, but it's there
- B. Operational data that is captured and stored but not analyzed or used
- C. The growing volume of data gathered during exploration of the deepest regions of outer space

2. Which of the following learning types is *not* associated with AI?

- A. Reinforcement learning
- B. Unsupervised learning
- C. Kinesthetic learning

3. Weak AI is most accurately described as

- A. Narrow AI—or AI that appears to think and understand but can do only what it is programmed to do
- B. AI intended for lightweight mobile apps
- C. AI that is easily distracted by irrelevant input that skews the outcome

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

DARING DETOURS

Travel apps for the spontaneous trekker



HotelTonight

Forget to book a room for your tech conference in Paris or the Canadian leg of your summer road trip? Find last-minute discounted hotel rooms with HotelTonight, which boasts lowest-price deals on a range of accommodations from hipster to high roller around the world. Search, tap, swipe, and you're there. [Free \(Android, iOS\)](#)



Hopper

Get the best-possible airline fares as much as a year in advance with this free app from Hopper. The company boasts advanced data science that analyzes as many as 15 billion flight prices daily and predicts future prices with 95% accuracy. Buy a cheap flight with a few taps on the app, or sign up to watch a trip and receive price alerts via push notification when prices drop. Hopper provides hotel bookings as well.

[Free \(Android, iOS\)](#)



BringFido

Anyone who travels with a dog knows the hassle of finding pet-friendly accommodations. BringFido gives you more than 200,000 places where you can enjoy your canine companion while on the road, including hotels, parks, beaches, and restaurants. (It also lists nearby dog walkers, veterinarians, pet supply stores, and other useful services.) Browse by region, city, or category, and filter by various criteria, including the number of animals permitted and price ranges. Book your stay via the app or by phone or email.

[Free \(Android, iOS\)](#)



"IoT projects that are much more likely to be successful start with a business-outcome-based approach," says Oracle Group Vice President IoT and Blockchain Applications Development Bhagat Nainani.

The IoT Is in the Applications

Devices connect to multiple enterprise applications to deliver business value. **BY TOM HAUNERT**

Smart connected devices are a good start, but businesses can get better results by using device-generated information across multiple applications to drive change. IoT applications for different types of devices and operations—from industrial devices and manufacturing to transportation and field service—can also use device data to generate and share insights in ERP, supply chain, and customer experience applications.

Oracle Magazine sat down with Oracle Group Vice President IoT and Blockchain Applications Development Bhagat Nainani to talk about today's enterprise IoT, the applications approach to IoT, and the power of IoT to drive connected business outcomes.

Oracle Magazine: What is the state of enterprise IoT?

Nainani: In the last five to seven years, IoT has gone from being sort of a buzzword or a cool new technology to being something that businesses are starting to deploy and get business value out of. Having said that, I think the rate of enterprise IoT deployment or growth has been a little slower than expected or predicted. So, yes, there are real IoT deployments, but a lot of

projects get stalled in the proof-of-concept or pilot stage.

IoT projects that are much more likely to be successful *start with* a business outcome-based approach, where there are clear KPIs [key performance indicators] that businesses are trying to improve with IoT. Next, those projects look at what kind of analytics they need or what kind of data sources they need. And *then* those IoT projects focus on where that data comes from—systems, devices, and so on.

Oracle Magazine: How are organizations looking at IoT application choices? What are Oracle's current IoT offerings?

Nainani: There are many vendors that offer a lot of IoT platform components and tool kits. They offer platform components to get your devices connected and to apply business rules. And then they offer AI and machine learning algorithms that businesses can use to create their own IoT solutions.

But businesses—industrial businesses in particular—really don't want to assemble these things themselves. They would like solutions that they can readily deploy to solve their specific business problems. They would prefer to

use an asset tracking solution, for example, which they can just configure with their assets and extend it as needed. We've seen this trend at Oracle, and over the last couple of years, we've shifted focus to more purpose-built IoT applications for addressing specific domains.

We provide five IoT applications today. Oracle Internet of Things Asset Monitoring tracks assets, their utilization, and their service needs. Oracle Internet of Things Production Monitoring looks at the factory floor, performs diagnostics, and measures performance against plans.

“We look at a business-outcome-based approach for applying machine learning and AI to IoT data.”

Oracle Internet of Things Fleet Monitoring monitors trucks, shipments, and driver behavior. Oracle Internet of Things Connected Worker tracks the health and safety of industrial workers. Oracle Internet of Things Service Monitoring for Connected Assets Cloud Service runs remote diagnostics for field service.

And these applications are built on our Oracle Internet of Things platform components, and they integrate with Oracle's ERP, supply chain, and customer experience apps. We effectively extend our business applications with insights and data from our IoT applications.

Oracle Magazine: What are best-practice strategies for handling the volumes of IoT data and getting value from all of that data?

Nainani: Less than 1% of IoT data is actually used today. Some companies start by getting basic alerts from IoT devices when something has crossed a certain threshold value. But a lot of devices have those capabilities built in.

But the real value of IoT comes from getting insights from the devices—from analyzing the data streams in real time and using machine learning to get those insights.

We look at a business-outcome-based approach for applying machine learning and AI to IoT data. First, employ KPIs to measure the outcomes; they can include equipment efficiency, asset utilization, production yield, and so on. Next, identify anomalies and use machine learning algorithms as part of the identification.



"We effectively extend our business applications with insights and data from our IoT applications," says Oracle Group Vice President IoT and Blockchain Applications Development Bhagat Nainani.

And again using machine learning, create predictions based on the anomalies reported, such as "this machine will likely fail before its next maintenance." Next, utilize AI for prescriptive actions and root-cause analysis such as "this part is the likely cause of the potential failure" or "the system recommends that this specific machine be maintained earlier than its planned

maintenance window." Finally, apply forecasting to look ahead to optimize operations, such as using production-line device information to help order the right quantity of raw material.

Because IoT information is used across applications, you need to connect IoT-based insights back to your applications to drive change. For example, when a vehicle failure means a truck

can't deliver on schedule the material needed for a field service repair, both the logistics application and the field service applications need that information. That kind of information flow

is called a *digital thread*, where IoT data gets used in the context of many other applications and provides a bidirectional workflow that connects the applications. □

PHOTOGRAPHY BY **ORACLE**

NEXT STEPS

[LEARN](#) more about Oracle Internet of Things.

[LEARN](#) more about Oracle IoT applications.



By Bob Rhubart



Lighting the Way

An educator's suggestion put Oracle Groundbreaker Ambassador Mohamed Taman on the path to success.

As a mechanical engineering student at Cairo University in Egypt in the late '90s, Mohamed Taman was completely in the dark about coding, software engineering, and computers. But his grades were good, and that caught the attention of the dean, who suggested that Taman might like to join the school's new computer engineering program. That turned out to be a moment of illumination.

Taman majored in electrical engineering, with a specialization in computer systems engineering. "Computer logic was one of the first things that caught my interest, then computer architecture and internal designs and programming," Taman recalls. "Trying to see the circuits



Oracle Groundbreaker Ambassador Mohamed Taman embraces education, relocation, and community participation.

RECOGNITION

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

and how binary systems all worked was amazing to me."

While at Zagazig University, in Egypt, Taman was also working at a major global biopharmaceutical company. "I made use of all the subjects I was studying and then applied those theories at work," he recalls. Between school, work, and occasional projects for friends, Taman gathered enough knowledge and experience to attempt Java 1.2 certification. In prepping for the certification exam, Taman relied heavily on the *Core Java* book series by Cay Horstmann. "I studied the whole book and then went to the exam and passed the first time," Taman explains. "I was very happy!"

After five years at the biopharma company, Taman worked at other companies as a software engineer and web developer. "I was always eager to study and to know what's new." That focus helped advance his career. "I had many different job offers and always chose companies that were growing and had a good reputation."

One stop along Taman's career path was a development engineering job with Oracle Egypt, where he worked on Metalink integrations for various Oracle acquisitions. He also put in a 10-year stint with an efinance startup, where he helped build an epayment system for the Egyptian government. All the while, he was accumulating more and more Java expertise.

In 2009 Taman discovered JavaOne, where he met James Gosling and several other luminaries. "I'd never been to a conference before," Taman recalls. "I was like, wow, all these authors whose books I'd read!" Inspired by the experience to get more involved with the community, he joined the Java Community Process (JCP) Program and became active in local Java user groups.

Taman places a high value on the JCP as a conduit to new people, cultures, and information. He also appreciates the impressive roster of companies represented on the JCP Executive Committee. "That's why I'm keen to keep my involve-

ment in the community. It's very good for your career."

Having become a senior manager in charge of development, IT, and I/O at the efinance startup, Taman found himself at a career crossroads, and he made the most of his skills.

"In Egypt you either become business- or technology-oriented in your career. I have chosen to be in the middle. I'm an enterprise architect, and I love R&D. I understand the business, am very close to the senior management, and can translate its vision and objectives into technical language for the technology team. I am the connection between them. It's what I love to do."

Given his particular skill set, Taman relocated from Egypt to Serbia. "Serbia is growing in terms of IT. There's a lot of demand for someone like me here."

Taman is now the senior enterprise architect for Comtrade Digital Services, based in Belgrade. Comtrade provides services for various European companies in the travel and mobility industries.

"I'm trying to put current systems into the correct architecture to make use of cloud computing," Taman explains.

In addition to working at Comtrade, Taman is founder and CEO of SiriusX Innovations, which focuses on blockchain-based applications plus training services in Java SE, Java EE, and software architecture.

A prolific writer, Taman regularly contributes articles to DZone, InfoQ, and other publications. He is the author of *JavaFX Essentials* (2015, Packt Publishing). His newest book, *Secrets of a Java Champion*, is scheduled for publication in October 2019 by Packt.

Taman often gives presentations at developer events, such as Oracle Code, DevNexus, and Devoxx. He is also active in local Java user groups (JUGs). "I love speaking; I love teaching and giving information," he says.

Through his community involvement, Taman earned Java Champion status in 2015 and became an Oracle Groundbreaker Ambassador in 2018.

Having settled into his jobs in Belgrade, Taman makes time to make the most of his new home. “I’m trying to make a lot of friends here in Serbia,” he says. “I love people. I’m exploring the country, seeing new places, and trying new foods.” □

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

PHOTOGRAPHY BY **BILL ADLER**

NEXT STEPS

FOLLOW Mohamed Taman:
his Blog
on Twitter
on GitHub

WATCH
Database Performance Tip.
Effective Design of RESTful APIs.
RESTful APIs, Blockchain, and the Need for Speed in Mobile Apps.



By Alexandra Weber
Morales

Use a Pencil

Java Champion Mani Sarkar's learning hacks

When Java Champion Mani Sarkar talks about technology, you think, “I’ll have what he’s having.” The prolific writer, speaker, and Java and JVM-based developer is a vibrant member of the London Java community as well as a passionate backer of open source projects such as Adopt OpenJDK. Although his day job focuses on strengthening teams, his interests include improving code quality and performance, implementing DevOps and machine learning, and more.

Talking to Sarkar about productivity means tapping into the mother lode. Here is just a sampling of the many ways he suggests improving your Java code and learning new technology more effectively.



Java Champion Mani Sarkar minimizes distractions, chooses more than digital tools, and learns about learning Java.

**“Read more code,
write more code,
think in code!”**

—Mani Sarkar,
Java Champion

DEEP WORK

“These days I take a lot of inspiration from the book *Deep Work* by Cal Newport,” says Sarkar. “Getting into the flow of working or coding isn’t hard at all. The idea is to shut yourself off from the outside world and distractions, timebox your tasks, take breaks, and be disciplined about not breaking this discipline. Cal Newport has a lot to say about how to do this. He calls it deep work, as opposed to the shallow work we all are used to doing. Developers must aim for and do deep work as much as possible.”

Sarkar tries to avoid time sinks, such as the kind of “unnecessary, nonpractical chatter about what’s right and wrong that’s often found in various forms of social media, ranging from Twitter and Slack to Reddit and Quora.” Instead, he recommends communicating via coding: “Read more code, write more code, think in code!” Two ways to do that, he adds, are by sharing Java code gists, which are snippets of code, and GitHub repos,

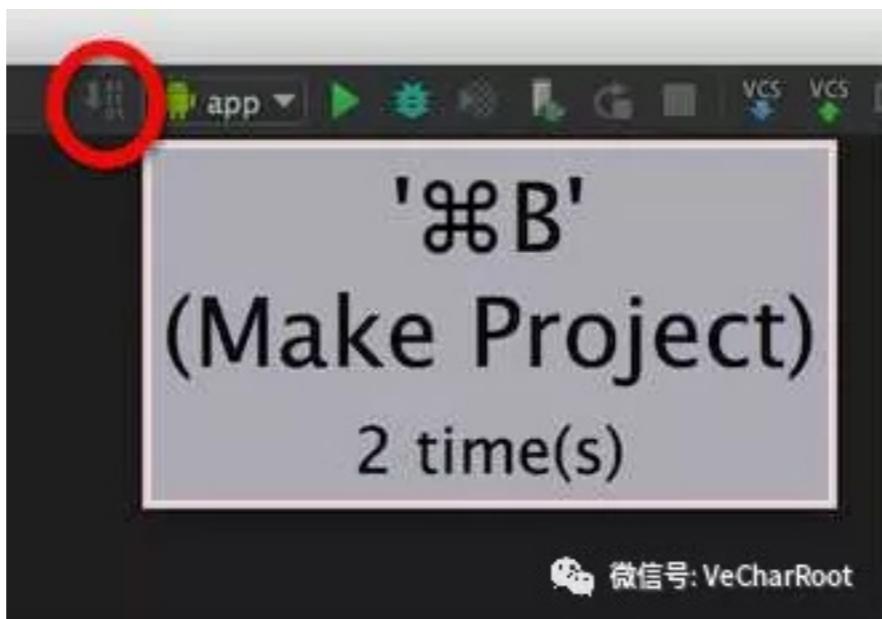
which are more-fleshed-out projects and documentation.

CHOOSE YOUR TOOLS

“No digital tool is good enough,” Sakar says. “I strongly suggest using paper and pencil (or pen) a lot more. We are able to remember a lot more and help drive our thoughts and express our creativity a lot more via our hands when we write or draw.” WET (Write Everything Twice), rhizo-mapping, and mind-mapping are three ways Sarkar tries to propel his pencil and come up with new ideas. “Such a practice helps build muscle memory,” he explains. “As the urban saying goes, ‘Your fingers are your memory.’”

As any good programmer knows, you also improve productivity—and prevent ergonomic strain—when you memorize keyboard shortcuts instead of clutching at the mouse or clawing the trackpad. Sarkar recommends using an IntelliJ plugin called Key Promoter (see [Figure 1](#)) to keep track of how often you reach for menu options instead of typing equiva-

Figure 1: Key Promoter keeps track of how often you reach for menu options instead of typing equivalent keyboard shortcuts.



lent keyboard shortcuts. “You can configure the plugin to keep track of how often you use the mouse to select the menu option, and if you cross that limit, it will nag you each time,” he says. “It also helps you create a shortcut for menu options if there isn’t one already.”

Sarkar has several favorite productivity-enhancing tools:

- [Tab Shifter](#) is for splitting the IntelliJ screen into vertical or horizontal tabs

and switching between them.

- [Spacemacs](#) combines the best of the Vim and Emacs editors.
- [Censum](#) is for finding JVM-based program memory leaks.
- [Snyk](#) is for vulnerability testing of public GitHub repositories or websites.
- Sarkar offers his own [curated resources](#) for polyglot Java running on the GraalVM virtual machine.
- [Java Flight Recorder](#) and [Java Mission Control](#) are for understanding how your app and the JVM are behaving. Both are included in the JDK, so no additional download is necessary.

LEARNING JAVA

Sarkar is passionate about the future of cloud native Java and is currently exploring serverless computing and systems for containerized deployment, including Docker, Apache Mesos, CoreOS rkt, and SmartOS containers. “The JVM language Clojure is a great example of simplicity and compactness,”

he says. “In fact, functional programming language concepts tout simplicity all the time.”

But what if you’re still learning Java or learning to be a better Java developer? Sarkar recommends attempting to solve katas, a concept that Java developers have enthusiastically borrowed from the Japanese martial arts world. The [Gilded Rose refactoring kata](#), for example, has become an extremely popular exercise that hones the skill of writing more-elegant problem-solving code.

The most contagious aspect of Sarkar’s approach is how, in a fashion befitting a programmer, Sarkar has taken a step back from the problem of learning Java to contemplate what learning means. “Your brain indexes knowledge when you are resting,” he says, “so sleep and rest let the brain index away. Don’t overload. Instead, interleave: Fill the gaps, and learn incrementally and iteratively.” □

Alexandra Weber Morales is Oracle’s director of developer content.

PHOTOGRAPHY BY JOHN BLYTHE

NEXT STEPS

FOLLOW the Oracle Developers Blog.

LEARN about the Java Champions.

SUBSCRIBE to *Java Magazine*.



Emerging Tech, Emerging Opportunities

Blockchain, IoT, and AI are here, but how do data experts use them in new business projects?



By Jeff Erickson



Jerry Ward has always been a data-base tech geek. But for him the database was never an end in itself—it was a launchpad into interesting business use cases. “Even back when I was upgrading our company to version 6 of Oracle Database, I was also working on our build of Oracle Manufacturing and Oracle Financials in Oracle Forms,” he says.

Today the regular speaker at Oracle tech user groups, including Independent Oracle Users Group (IOUG) and ODTUG, still eschews “tech for tech’s sake,” instead continually exploring use cases where his data could be valuable to the business.

One of the ways Ward does this is by staying active in the Emerging Tech special interest group (SIG) of Oracle Application Users Group (OAUG). The members of this SIG are responsible for bringing use cases to explore. For example, “We know that with blockchain most of our members are still in the proof-of-concept stage, but we’ve seen where it’s starting to make sense,” he says. One Emerging Tech SIG member is from a company that makes high-end vitamins and serums and wants to track and prove its ingredients, and “in that case, it’s worth it to spend a lot per transaction to get the firepower that is needed for blockchain,” Ward says.

Other members are from municipal governments that are interested in using blockchain to track property documents, he says. “But if you’re shipping thousands of low-cost commodity parts like calculators,” as one SIG member does, “the blockchain investment might not be worth it yet,” he adds.

IoT is further along, Ward says, citing a couple of use cases from the group. Ward’s partner in the Emerging Tech SIG, Hans Kolbe, brought a use case from German aluminum manufacturing companies he works with. “They need to track their actual costs in excruciating detail and do forecasting on the raw material cost and prices,” says Kolbe. That requires a combination of the IoT, big data, and Oracle Applications. “We’re thinking we can feed data directly to the apps, but we need to test it” and run it by SIG data experts such as Ward, Kolbe says.

Ward is also exploring IoT and Oracle Autonomous Data Warehouse. “We’re using IoT on the farm to measure animal

health” and proactively order veterinary services, he says. The project uses tiny sensors in a cow’s stomach to catch health signals early. “We get information off the device and stream it through Oracle Event Hub Cloud Service and into an Oracle Autonomous Data Warehouse instance so we can do reporting and some predictive analytics,” he says. “Then we feed that into Oracle ERP Cloud [Oracle Enterprise Resource Planning Cloud] so we can schedule veterinarian visits, create billing events, and pass everything through in an automated fashion.”

NEXT UP: ARTIFICIAL INTELLIGENCE

Ward says it’s important for people who work with data to keep up with the cloud and AI, because those technologies continue to create opportunities—with Oracle Autonomous Database being a prime example. “It’s getting easier every year to apply these technologies, because people have figured out what works and they’ve curated that and pack-

aged it up into services, so now you can do one-click provisioning," he says. "The exciting part is that you don't have to be a data scientist to leverage machine learning, because Oracle is building the algorithms into SQL for you."

A broader review of AI will be the focus of the OAUG Emerging Tech SIG this year, says Ward, including a look at how AI can piece together a coherent story in ever-shifting arenas such as trade compliance. Ward and Kolbe and their fellow SIG members want to see if AI can automate trade documents and trade compliance. "How can we combine publicly available data about current trade deals such as China/US

or Brexit with searches of government websites and translate it down to the specific company needs?" asks Kolbe. But Kolbe admits that he's coming at the problem completely from the business side. He'll need help from Ward and other data experts to understand how to feed the AI algorithms what they need. "Multinationals want to learn about this," says Kolbe. "There's an enormous amount of time and money spent on it."

And that spells opportunity for data experts who work to stay on top of the latest business use cases. □

Jeff Erickson is editor at large for Oracle Content Central.

NEXT STEPS

LEARN more about
Oracle Artificial
Intelligence Platform.

LEARN more about
Oracle Autonomous
Database.

TRY Oracle Autonomous
Database.



Love at First Byte

The invisible line between work and play and the joys of finding the right tool for the job



Marcin Przepiórowski

Dublin, Ireland



Company/URL: [Delphix](#)

Job title: Senior technical principal, solution architect

Oracle credentials: Oracle Certified Professional (Oracle Database 11g, Oracle RAC 11g)

Length of time using Oracle products: 19 years

Which new features in Oracle Database are you currently finding most valuable? I like features related to high availability, monitoring, and stability of the environments. These include the Oracle standby database solution, with or without Oracle Data Guard, the

Active Session History and Automatic Workload Repository features of Oracle Enterprise Manager 12c, and the SQL Plan Baselines feature of Oracle Database.

What advice do you have about getting into software architecture? Learn from existing source code instead of books. Reading and understanding well-designed code allows people to put particular patterns in their minds, and it also shows how important it is to have clean and commented

code. At the end of the day, we developers spend most of our time debugging code rather than writing it, so becoming skilled at reading code is very useful.

What's your favorite thing to do that doesn't involve work? As a computer geek, I have trouble saying when I stop working and when I start my hobby. When I want a break from technology, I read books or listen to music. I'm also a sports fan and like to watch volleyball and Formula One.

**Bill Dunham** 

Charlotte, North Carolina

**Company/URL:** [OATC](#)**Job title:** Principal**Length of time using Oracle products:** 34 years

How did you get started in IT? I was fortunate enough to get an IT degree in college and then started my career as an operator/developer working first on IBM mainframes and then on Prime Computer's PRIMOS in the early '80s. I first worked with Oracle on a Prime EXL316 server running Primix OS around 1986. My boss and the IT director at the time chose Oracle as the database of choice to replace Henco Software's Info database and tools, and it was love at first byte.

Looking back on your career, what early lessons you learned are still relevant today? One of the first applications I developed using Oracle was an "electronic time sheet." We did weeks of testing and user acceptance, but it was confined to a small group of people. We wrote the application based on table-level locks, and when we rolled it out organization-wide, it didn't work. Many users were trying to update their time sheets at the same time, causing locks, delays in transactions, and incomplete time sheets. Around that time, Oracle released block-level, and eventually row-level, locking—which we applied immediately—and it saved

the rollout of the application. For a young developer with a lot to learn, the lesson was around testing every application in a multiuser environment.

Which new features in an Oracle application are you currently finding most valuable? The new Alta user interface in Oracle E-Business Suite 12.2.8 is a welcome change. Its use of themes is helpful to distinguish Oracle E-Business Suite instances. I've also noticed improvements in Oracle Application Framework personalizations, mobile applications, and many other UI improvements throughout Release 12.2.8.



Nelson Calero

Montevideo, Uruguay



Company/URL: [Pythian](#)

Job title: Oracle project engineer

Oracle credentials: Oracle Certified Expert (Oracle Database 12c)

Length of time using Oracle products: 22 years

What's your favorite tool or technique on the job?

My favorite tool is always the one that solves the problem. My technique is to focus on the problem that needs solving and prepare thoroughly to make sure the chosen approach will accomplish that. Usually we have a variety of components and tools involved, and being able to repeat what we think will work and document it well is often

overlooked in favor of reaching deadlines.

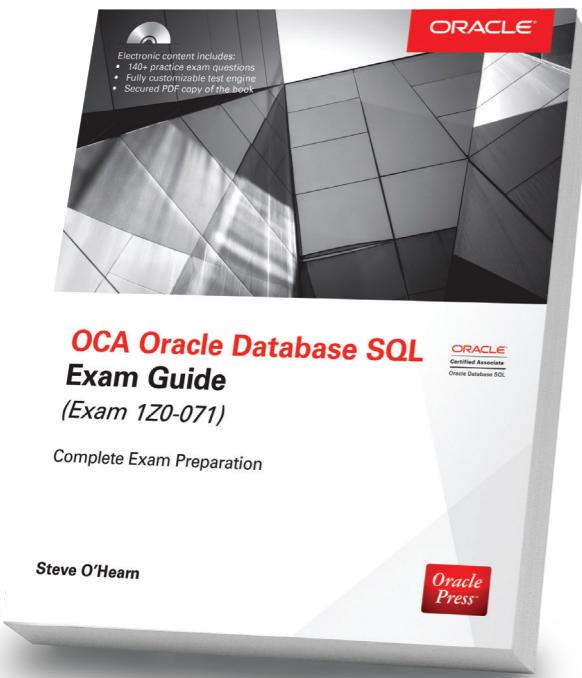
What's the most common cause you see when IT projects go wrong? Usually it's underestimating the complexity of the systems that leads to overly optimistic planning, combined with teams that are not experienced enough with the particular technologies involved. These projects get bogged down by problems and delays, creating a vicious circle that can be hard to identify and fix from the inside.

What's the next big thing driving change in your industry? Today it's artificial intelligence, just as it

was the cloud some years ago. AI is already driving change, taking over so much that once required manual and repetitive interactions. It's just a matter of time before we see it also taking over more-complex tasks, as the algorithms and the technology itself improve. It sounds a bit scary initially, but it also opens the door to so many new things. Complex technologies will be cheaper and more accessible—as we are seeing now with image recognition on mobile phones—and the possibility to apply it to every industry will lead to better products and more-efficient processes.

Your Destination for Oracle and Java Expertise

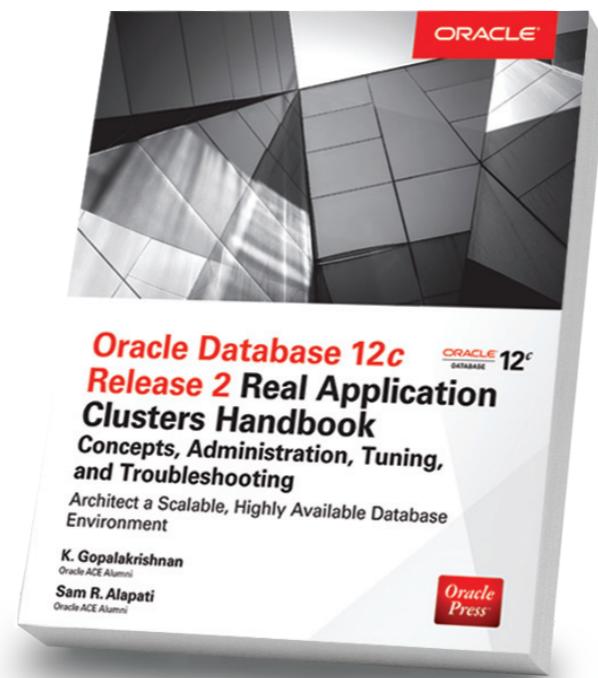
**Written by leading technology professionals,
Oracle Press books offer the most definitive, complete, and up-to-date
coverage of Oracle products and technologies available.**



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

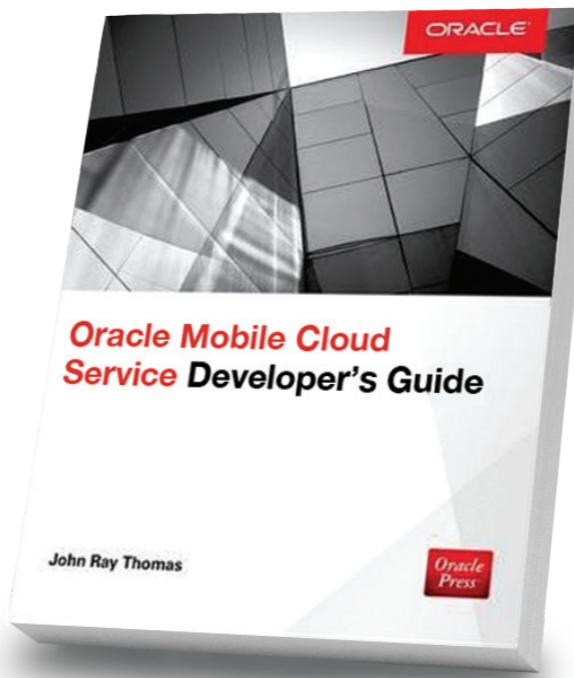
Steve O'Hearn

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



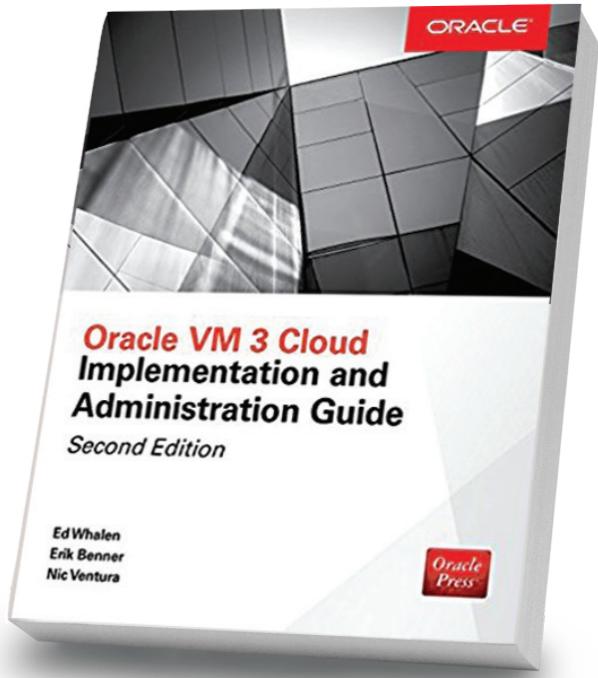
Oracle Database 12c Release 2 Real Application Clusters Handbook

K. Gopalakrishnan, Sam R. Alapati
Master Oracle Database 12c 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.

Only the Best Business Data

With Oracle DataFox, business data is accurate, fresh, dynamic—and AI-powered.

BY ALAN ZEICHICK

"Where are our best prospects?" the CEO asks the sales manager. "Two companies in our market are merging—how will that affect us?" the CEO asks a business analyst.

To answer these types of questions with enough depth to make decisions, management teams need lots of data that's relevant to the decisions and actions the team must face.

What's more, the data must be easily analyzable by traditional business intelligence tools and artificial intelligence. But simply having access to relevant data that's accurate and



fresh and interpretable by AI-based analytics is not enough. Business leaders can gain a competitive edge by incorporating dynamic signals into their applications to drive contextual action.

Dynamic signals are AI-driven insights that chew through the internal data and external feeds to provide timely, relevant alerts. By embedding these capabilities into enterprise applications, businesses can make the best decisions based on the available information and signals as fast as possible.

Moving to Europe? Let's Find the Best Customer Prospects

InnovationQuarter, based in The Hague and Rotterdam, helps foreign companies—from startups to large corporations—establish a business presence in West Holland. The organization, funded by the Dutch national, provincial, and local governments and three universities in the region, provides information to help companies learn about doing business in the Netherlands, helps them find office space and housing, assists with hiring and permits, and even offers business-to-business networking and public relations services.

AI-based data analytics and smart signals

help InnovationQuarter identify companies that might want to expand into the Netherlands, explains Chris van Voorden, director of foreign investments. A challenge: There are hundreds, even thousands, of prospective clients, and van Voorden's 40-person team needs to find the best ones. But who are they?

One tactic is trade shows. "We go to those events, but it's much better to investigate the companies that will be exhibiting or attending an event up front," van Voorden says, to determine if they're worth pursuing. That's where the data feeds and AI analytics come in. And that's where Oracle DataFox comes in.

Oracle DataFox provides access to data on millions of organizations, including most public companies worldwide. Algorithms continuously source, refresh, and analyze data, performing more than 49 million customer record updates per week to help ensure that the data is fresh and accurate. By providing data on company head count, funding, revenue, growth, and more, Oracle DataFox integrations within Oracle's ERP, sales, and marketing applications enable improved AI outcomes.

"Now we can import these exhibitor and attendee lists into Oracle DataFox, and it will

“We’re saving money, saving time on preparation, and finding the right targets.”

—Chris van Voorden, Director of Foreign Investments, InnovationQuarter

come up with indicators we are looking for in these companies, the right sorts of signals,” van Voorden says. “Did they get new funding? Do they have international expansion plans? Is something changing in their geographical footprint? Are they recruiting business development managers or directors for the EMEA region: Europe, Middle East, and Africa?

“Those are the signals we can get out of the Oracle DataFox system. This helps us to become more effective, because we know which companies to talk to. We also have the indications of what to start a conversation about. We’re saving money, saving time on preparation, and finding the right targets.”

The Best Data Is Fresh, Accurate, Complete, and Dynamic

The more of that fresh and accurate data, the better the insights. Take company information: Oracle DataFox monitors press releases, web-

based information, blogs, and other sources of competitive intelligence on more than 3.7 million public and private organizations, updating much of that information weekly. That information can then be scored and correlated to help prioritize top sales prospects, identify opportunities, and warn of risks.

The smarter that data, the better. “Smart data is precise. It’s high quality,” says Bastiaan Janmaat, former CEO of DataFox and vice president of product management at Oracle. “Smart data is continually refreshed, so it’s not stale or static. Also, smart data includes dynamic signals, reflecting data types that are constantly changing.”

And finally, Janmaat adds, “smart data, with artificial intelligence, should be embedded into business applications.” Those AI-infused applications then “nudge” users with information on what they need to know and do, says Clive Swan, senior vice president for applications

“Our vision includes the use of AI to minimize or eliminate the low-value tasks performed by our people.”

—David Pitta, Chief Marketing Officer, BrightTALK

development at Oracle.

In March 2019, [Oracle announced new Oracle DataFox integrations with Oracle Enterprise Resource Planning Cloud \(Oracle ERP Cloud\), Oracle Customer Experience Cloud \(Oracle CX Cloud\), and Oracle Marketing Cloud.](#)

“All of our solutions are underpinned by Oracle’s smart-data strategy, which enriches first-party application data with third-party, web-scale, trusted data, enabling our AI solutions to make even better decisions,” Swan says.

Presenting the Right Talk to the Right Person at the Right Time

[BrightTALK](#), which operates a library of 150,000 expert online talks on topics ranging from information technology to financial securities, reaches eight million professionals, thanks in part to its AI-based recommendation engine,

which helps connect the right content with the right person at the right time.

Oracle DataFox is part of BrightTALK’s multi-layered AI. “Oracle DataFox plays a big role in working with our customers at the business level,” says David Pitta, BrightTALK’s chief marketing officer. “It’s key to our go-to-market strategy and how we acquire, retain, and grow our business with customers.”

BrightTALK’s comprehensive, custom AI solution—Ada—presents the company’s webinars and videos to specific professionals. “Ada is constantly serving content to professional audiences and giving them the option to curate a unique experience,” Pitta says.

As professionals engage with BrightTALK, “our AI is learning their interest in specific topics and where that content fits in their journey,” he says. “This builds out the user’s profile and trains Ada”—to not only make better recommendations

but also to improve the overall experience.

Furthermore, AI is helping make BrightTALK employees more productive, enabling them to spend more time on higher-value tasks. "Our vision includes the use of AI to minimize or eliminate the low-value tasks performed by our people and get their brains focused on solving high-value, complex business problems," Pitta says. "The year 2019 is all about simplifying and automating our business."

Simply "Delicious"

Simplification and automation: Those are benefits Oracle's Janmaat sees as well. "Every customer is quick to see that AI, built on smart data, is good for them," he says.

Customers often don't need to get involved at

the most complex algorithmic level. "What they care about is the output," Janmaat says. "They don't need to know how the sausage got made, as long as the sausage is delicious."

But the *ingredients* are important. Says Janmaat: "It's the Oracle DataFox code under the hood that gets all this data from all these different places and structures it all, cleans it up, throws out the junk, keeps the stuff that's useful, deduplicates when we're grabbing data from multiple sources, and then puts it all in one place." O

Alan Zeichick is director of strategic communications for Oracle, where he provides insights into and analysis of cloud computing and other advanced technologies. Follow him [@zeichick](#).

ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about Oracle DataFox integration.

Oracle AI.



Tracking Conflict Minerals

Supply chain company ensures ethical practices with Oracle Blockchain Platform.

BY MIKE FADEN

Blockchain, the emerging distributed ledger technology, has the potential to transform the way businesses work, but it's not a panacea for every organizational challenge. The key is understanding where it is more effective than other technologies.

One of those areas is for transactions that span multiple companies and where there's a lack of visibility and trust among the entities, such as the participants in a supply chain, notes Bhagat Nainani, an Oracle group vice president whose

work includes blockchain application development. “With a blockchain, you introduce that trust, because every transaction, every document, is on a shared ledger that is visible to everybody and is endorsed by the appropriate parties,” Nainani says.

London-based Circulor is at the center of such activity. Using Oracle Blockchain Platform, it’s helping companies ensure that the raw materials in their products are sourced ethically and in compliance with regulations by tracing materials through the entire supply chain, from their origin to the finished product.

Circulor started by focusing on a particularly challenging problem: tracking valuable “conflict minerals” (minerals extracted in a conflict zone and sold to perpetuate the fighting), such as gold, tantalum, and cobalt, from the mine to their eventual use in manufactured products such as smartphones and electric cars. Regulations require manufacturers to ensure that those minerals are legally extracted and aren’t sourced in countries that use child or slave labor. Manufacturers are also keen to demonstrate to consumers that their products are ethically and sustainably sourced.

The supply chains for conflict minerals involve multiple parties in various countries worldwide, notes Doug Johnson-Poensgen, Circulor’s CEO and cofounder. Take cobalt, which is in high demand for lithium-ion batteries used in electric cars. The ore may be mined and smelted in Africa, transported to a refinery in China, incorporated into cathodes in Vietnam, sent to a battery manufacturer in South Korea, and finally built into vehicles by a carmaker in the US or Europe.

No central authority governs the entire supply chain, so it’s not possible to require that all parties use a single tracking and compliance database, Johnson-Poensgen says. So Circulor uses blockchain to store an immutable record of the material’s path through the supply chain, together with smart contracts that apply consistent rules that each party must follow before it can pass the material to the next company in the chain, he says.

“Our target customers are big enterprises, and we have built an enterprise-class solution that is delivered as software as a service to those customers,” Johnson-Poensgen says. “We need to be able to demonstrate security,

“We need to demonstrate an unbroken chain of custody on the journey from mine to manufacture and also to reliably connect the input material to the output product at each processing step.”

—**Doug Johnson-Poensgen**, CEO, Circulor

scalability, and availability out of the box—and that's what we get with Oracle blockchain." Circulor was able to develop a production system from scratch—spanning multiple companies—in a few months. "The engagement with Oracle has been amazing," he adds.

First Application

Circulor's first publicly announced application traces tantalum, a rare mineral integral to capacitors, which are used in millions of mobile devices and other electronic products. A smartphone typically contains hundreds of capacitors.

Circulor's system went live in three mines and an ore-sorting facility in Rwanda in the fall of 2018, enabling manufacturers to track regulatory compliance and ethical practices more

effectively than with stacks of paper attestations, while also reducing cost.

Rwanda is the world's largest tantalum exporter. The problem is that the mineral could be smuggled in from neighboring Congo, where it's often mined by children or slave labor. As a result, US and EU regulations require companies to trace and report the source of the tantalum they use and demonstrate that it has been responsibly sourced. But before blockchain, it wasn't possible for them to create a foolproof traceability system, which led to a black market for the mineral.

Tracing conflict minerals is particularly challenging, because the raw materials are transformed into completely different products as they move along the supply chain. The material



Circulor was able to develop a production system from scratch—spanning multiple companies—in a few months. “The engagement with Oracle has been amazing,” says Doug Johnson-Poensgen, CEO at Circulor.

starts as an anonymous-looking bag of metal ore, but it changes form multiple times as it's processed by the different companies in the chain. It may ultimately get incorporated into many different components.

This is a much more complex problem than,

say, tracking a single head of lettuce from farm to supermarket shelf, Johnson-Poensgen notes. “That complexity is why it's taken longer to deal with this challenge than with other track-and-trace applications,” he says. “We need to demonstrate an unbroken chain of custody on

the journey from mine to manufacture and also to reliably connect the input material to the output product at each processing step."

Verification at Every Step

To provide that assurance, Circulor's blockchain application applies multiple verification methods at each step in the process. Those methods ensure that each company meets the rules of the smart contract required to pass the material to the next company in the supply chain.

At the mine, a smartphone app uses facial recognition technology to verify that the tantalum originates with a registered user. The phone's GPS coordinates record the source, and other documentation methods, such as taking photos and using handheld mass spectrometers, provide additional verification. The system creates a QR code that identifies the source, which is attached to a tamper proof bag containing the ore, ready for scanning by the next company in the chain.

At each step in the supply chain, the tantalum is recorded and checked when it arrives and leaves. Companies can verify that the

materials were scanned at the designated locations during their journey through the supply chain and that they were unloaded by authorized personnel. Different verification methods, including weighing and analyzing the material, may be used, depending on the stage in the supply chain, so that companies don't need to radically change their existing procedures to improve traceability.

Now Circulor is expanding to other industries. It's working with a major car manufacturer to trace the cobalt used in its lithium-ion batteries, and it's launching into tracking other raw materials, such as rubber, wood, palm oil, and cotton.

Circulor is also integrating blockchain with other Oracle software. For example, it's moving collected data to Oracle Database so that companies can use Oracle Analytics Cloud to gain deeper insights. "When a manufacturer is tracking multiple raw materials through its supply chain, it needs to be able to make sense of all the information in a coherent way," Johnson-Poensgen says.

Circulor is on the leading edge of a wave of supply chain applications that will use block-

chain, Johnson-Poensgen maintains. "Today very few people have been able to stitch together parts of a supply chain on a blockchain," he says. "Five years from now, lots of people will be doing it." □

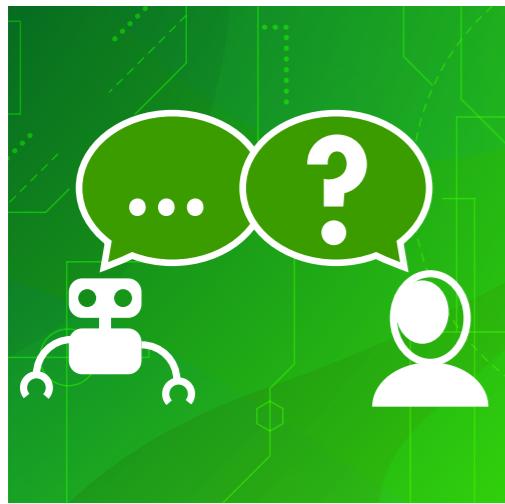
Mike Faden is a principal at Content Marketing Partners. He has covered business, technology, and science for more than 30 years as a writer, editor, consultant, and analyst.

ILLUSTRATION BY **WES ROWELL**
PHOTOGRAPHY BY **JOHN BLYTHE**

NEXT STEPS

LEARN more about Oracle Blockchain Platform.

TRY Oracle Blockchain Platform.

**ORACLE DIGITAL ASSISTANT**

Escape Room

How to unlock users who get stuck in a chatbot conversation

Building good chatbots is a mixture of art and technology. The art of developing chatbots lies in the conversation design, which is usually *not* what software developers have been trained to do.

It is human nature that conversations are not typically straightforward. Topics may change in the middle of an interaction or, if we don't understand a question, we may ask for clarification.

Though the human brain is far from being a fast supercomputer, which is able to analyze millions of records in fractions of a second, it is a real genius when it comes to understanding context and putting together 1 + 1 to become 2, even if parts of the information are missing.

Because artificial intelligence is not yet on the level of human intelligence, chatbots need good conversational design to ensure that context changes are handled and human users are safely guided through a conversation.

If your conversation design cannot prevent users from getting stuck in a bot conversation, the least it can do is to show users a way to free themselves from being locked in.

This article discusses techniques you can use in Oracle Digital Assistant to guide users in situations where they don't know how to proceed with a conversation or change the topic.

ABOUT THE SAMPLE

Oracle Digital Assistant is the next generation of the Oracle chatbot platform. It manages and coordinates multiple smaller-scoped *skills* in a composite chatbot solution that assists users in completing multitask conversations.

The use case for the hands-on exercise in this article is a simplified version of a pizza ordering bot. You don't build the complete digital assistant solution but the skill only, for which a starter bot is provided.

GETTING READY

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

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

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

1. Start Oracle Digital Assistant in a browser by typing `https://<your cloud URL>/botsui/` into the **URL** field.
2. Click the hamburger icon (≡) at the upper left.

3. Choose **Development** and then **Skills**.
4. Close the menu by clicking the hamburger icon (≡).
5. Click the **Import Skill** button at the upper right.
6. Navigate to the downloaded and extracted resources for this article and navigate to the starter folder.
7. Select the **OraclePizza2019(1.0).zip** starter bot file, and click **Open**.
8. Click the **Oracle Pizza 2019** tile in the **Skills** dashboard to open the skill bot.
Note: If you don't see the imported bot because other bots fill your screen, type Oracle into the **Filter** field above the **+ New Skill** tile.
9. Train the bot by clicking the **! Train** link in the upper right.
10. In the opened dialog box, accept the default settings and click **Submit**.
11. Click **Instant Apps** in the upper right.
Note: Instant Apps should open in a separate tab. If no tab opens, ensure that the pop-up is not blocked and try again.
12. In the opened tab, click the **Add Instant App** tile (+).
13. Click the **Import** tile (import icon).
14. Click the **Upload a Schema File** link.
15. Navigate to the downloaded and extracted resources for this article, and navigate to the starter folder.
16. Select the **OraclePizzaHelp.json** file, and click **Open**.
17. Click the **Instant App Info** button (instant app info icon) in the upper left and notice that the API ID value is **oraclepizzahelp**.
18. Close the **Instant Apps** browser tab.

What you just did: By importing the **OraclePizza2019(1.0).zip** file, you installed and then trained the pizza bot sample skill. You are going to use the imported Instant Apps web form later in the hands-on exercise.

EXPERIENCING THE PROBLEM

Before you fix the imported skill's conversation issue, first experience the problem from a user perspective.

19. Run the sample bot in the embedded conversation tester by clicking the **Skill Tester** icon (▶) in the left menu.
20. Type I like to order pizza into the **Message** field and press the Enter key on your keyboard.
21. Type salami into the **Message** field and press the Enter key again.
22. Apparently salami is not a valid answer. Type ham into the **Message** field and press the Enter key.
23. Oracle is an American company, so type bbq into the **Message** field and press the Enter key.
24. Again, this is not a valid choice. Click the **Reset** button (Reset) at the top right.
25. Close the conversation tester by clicking the **Close** icon (X), located next to the **Reset** button.

What happened: The pizza skill validates the pizza type selection against a value-list entity. The entity contains values and synonyms for the pizzas that a user can order. When you input a wrong pizza type, the entity validation fails and the prompt is displayed again. The implementation in the sample keeps you in an infinite validation loop with no exit. So the first thing you need to do to improve the user experience is to implement an exit strategy.

EXITING THE VALIDATION LOOP

Input components in Oracle Digital Assistant have two properties in common:

- `maxPrompts` expects a numeric value that defines the number of allowed failed user inputs before a cancel transition.

- variable references a dialog flow context variable that is updated with the user input. If the context variable is of an entity type (for example, PizzaType), the variable gets validated against the entity when the user provides a value.

To escape the validation loop, all you need to do is define a maxPrompts property value and configure the cancel action transition.

26. Open the dialog flow builder by clicking the **Flows** icon (Ξ) in the left menu.
27. Navigate to the askPizzaType state and set the value of the maxPrompts property to 2. The resulting line should look like this: maxPrompts: 2.
28. Put your cursor into line 42. Use the space bar on your keyboard to vertically align the cursor with the beginning of the next element.
29. Type actions: and press the Enter key.
30. Position the cursor to vertically align below the actions: element you just added. Again, use the space bar on your keyboard to do this.
31. When the cursor is vertically aligned, press the space bar two more times, so the cursor is indented by two characters.
32. Type cancel: "cancelPizzaOrder" (with the quotation marks). The transitions: element should look like it does in **Figure 1**.

Figure 1: Transitions element with cancel action

```
40  transitions:  
41    next: "askPizzaSize"  
42    actions:  
43      cancel: "cancelPizzaOrder"  
44
```

33. Repeat the testing steps 19–22.
34. Notice that you are no longer stuck in a validation loop. You are free, and you could give the pizza ordering another try.
35. Click the **Reset** button () at the top right.
36. Close the conversation tester by clicking the **Close** icon (X), located next to the **Reset** button.

What you just did: In this part of the hands-on exercise, you interrupted the validation loop by setting the `maxPrompts` parameter and defining a `cancel` transition that handles the invalid user input when the value set for `maxPrompts` is exceeded. Still, the user experience is not great.

Note: Instead of canceling the pizza order, you could have navigated the request to a state that provides detailed help. This exercise will eventually do that, so please bear with me for the moment.

PROVIDE VISUAL HELP

In this part of the hands-on tutorial, you are going to display a list of values instead of only displaying an input prompt to the user. After all, good user guidance is a great recipe for a positive user experience.

37. Go to line 39 of the dialog flow builder and edit the component definition as shown in [Figure 2](#). The highlighted parts are the new configuration information to add.
38. Run the sample bot in the embedded conversation tester by clicking the **Skill Tester** icon () in the left menu.
39. Type I like to order pizza into the **Message** field and press the Enter key.
40. A list of pizza types is displayed for the user to select from. However, you insist on ordering a salami pizza. Type salami into the **Message** field and press the Enter key.

Figure 2: Adding a list of actions to the pizza type prompt

```
28 askPizzaType:  
29   component: "System.CommonResponse"  
30   properties:  
31     processUserMessage: true  
32     keepTurn: false  
33     variable: "pizzaType"  
34     nlpResultVariable: "iResult"  
35     maxPrompts:  
36       metadata:  
37         responseItems:  
38           - type: "text"  
39             text: "What pizza do you like?"  
40           actions:  
41             - label: "${enumValue?capitalize}"  
42               type: "postback"  
43               payload:  
44                 variables:  
45                   pizzaType: "${enumValue}"  
46                   iteratorVariable: "pizzaType.type.enumValues"  
47     transitions:  
48       next: "askPizzaSize"  
49     actions:  
50       cancel: "cancelPizzaOrder"
```

41. The list of values is displayed again. Next type bbq into the **Message** field and press the Enter key.
42. Again, as before, the pizza order is canceled. Click the **Reset** button () at the top right.
43. Close the conversation tester by clicking the **Close** icon (X), located next to the **Reset** button.

What you just did: In this part of the hands-on instructions, you improved the user experience by providing good guidance on the choices of pizza. Still, if users

insist on a pizza type that is not in the menu, the behavior of the bot is not such that it provides sufficient information. For example, the System.CommonResponse component that is used in the askPizzaType state supports page ranging to display a reduced list of values when the potential list of values is too long to display without requiring users to scroll. In this case, the user assumption that salami is a valid pizza type may appear to be correct. So you need to provide a bit more guidance in the prompts, which is what you will do next.

Note: Selecting a value from the list or typing a value contained in the list would navigate the user to the next state, which is to specify a size for the pizza. But we are not at that point yet.

ASSIST USERS WITH PROMPTS

Using Apache FreeMarker expressions in Oracle Digital Assistant, you can display text messages conditionally. So for the next user experience adjustment, after the first failed user input attempt, you are going to tell the user that the provided pizza type is not valid and that only a pizza from the menu list can be chosen.

44. Edit the text property in line 39 and change it from

text: "What pizza do you like?"

to (all in a single line)

```
text: "<#if system.invalidUserInput == 'true'>Sorry, \"${system.message.messagePayload.text}\\" is not a valid pizza type. Please choose a value from the menu. </#if>What pizza do you like?"
```

Note: The value string above is provided in the starter folder of the download resources for this article. You can find it in a text file named step-44_textProperty.txt.

45. Run the embedded conversation tester by clicking the **Skill Tester** icon (▶).
46. Type I like to order pizza into the **Message** field and press the Enter key.
47. When the list of values appears, type salami into the **Message** field and press the Enter key. Notice the prompt displayed with the pizza type list, which now repeats your text input, mentioning that the input is not valid and asking you to choose a value from the menu.
48. Click **Reset** and close the tester.

What you just did: Apache FreeMarker is an open source templating engine and expression language that is used in the Oracle Digital Assistant dialog flow. The internal `invalidUserInput` system variable is set to true after the first failed attempt of a user to provide a valid data input. The Apache FreeMarker `<#if>`-directive is then used to repeat the provided user message along with a text message explaining what the user should do next.

ALLOW USER TO ACTIVELY CANCEL OR REQUEST HELP

So far, so good; but as long as there is better, just good is not enough. With your improvements, the pizza skill nicely guides users on what to do, and it still cancels the pizza order when the value in the `maxPrompts` property, which defines the maximum number of failed user input attempts, is exceeded. Instead of the chatbot automatically exiting the validation loop, let's make the user do so intentionally.

49. Go to line 35 and remove the value set for the `maxPrompts` property. This change configures the `askPizzaType` state to go into an infinite validation loop when the user does not provide a valid value for the pizza type.

50. Go to line 46 and edit the askPizzaType state with the content highlighted in **Figure 3**. Make sure the indenting looks exactly as in the figure. The newly added globalActions element aligns with the responseItems element in line 37.

Figure 3: Adding conditionally displayed buttons for canceling and getting help

```
28 askPizzaType:  
29   component: "System.CommonResponse"  
30   properties:  
31     processUserMessage: true  
32     keepTurn: false  
33     variable: "pizzaType"  
34     nlpResultVariable: "iResult"  
35     maxPrompts:  
36     metadata:  
37       responseItems:  
38         - type: "text"  
39           text: "<if system.invalidUserInput == 'true'>Sorry, \${system.invalidUserInput}</if>  
40           actions:  
41             - label: "\${enumValue?capitalize}"  
42               type: "postback"  
43               payload:  
44                 variables:  
45                   pizzaType: "\${enumValue}"  
46                   iteratorVariable: "pizzaType.type.enumValues"  
47       globalActions:  
48         - label: "Help"  
49           type: "postback"  
50           visible:  
51             onInvalidUserInput: true  
52           payload:  
53             action: "help"  
54         - label: "Cancel"  
55           type: "postback"  
56           visible:  
57             onInvalidUserInput: true  
58           payload:  
59             action: "cancel"  
60       transitions:  
61         next: "askPizzaSize"  
62       actions:  
63         cancel: "cancelPizzaOrder"  
64         help: "gettingHelp"
```

51. Go to line 63, which defines the cancel action, and create a new help action in the row below.
52. Set the value of the help action to "gettingHelp", as shown in [Figure 3](#).
53. Run the embedded conversation tester by clicking the **Skill Tester** icon (▶).
54. Type I like to order pizza into the **Message** field and press the Enter key.
55. Then, type salami into the **Message** field and press the Enter key.
56. Notice two buttons displayed at the end of the pizza type list: **Help** and **Cancel**.
57. Click the **Cancel** button to see the cancel message, as before.
58. Again, type I like to order pizza into the **Message** field and press the Enter key.
59. Type salami into the **Message** field and press the Enter key.
60. Click the **Help** button and notice the help that is provided by the bot.

What you just did: In this part of the hands-on exercise, you created two global buttons that are displayed after the first time a user fails to provide a valid data input. Instead of using Apache FreeMarker expressions to display the buttons, you used the `onInvalidUserInput` property (line 57 of [Figure 3](#)) of the `System.CommonResponse` component. Each button triggers an action that is mapped to an action transition: `cancel` (line 63) and `help` (line 64). Now the skill behavior is that the user remains in the validation loop until one of the two buttons is clicked.

IMPLEMENT HELP USING ORACLE INSTANT APPS

Often users don't need help but just a better interface. This is especially true when users get stuck in a longer conversation flow. They don't appreciate provided help that interrupts the conversation for them and makes them start over.

Oracle Instant Apps are reusable microscoped web applications you build declaratively for use in the context of a skill bot conversation. At runtime, instant apps are

executed in the mobile device's web view or a browser window outside of the conversational channel.

Oracle Instant Apps lets bot designers create instant apps to provide users with an easy way for entering structured form data into a bot conversation. When stuck in a bot conversation, users tend to be frustrated with the conversational medium. This is where Oracle Instant Apps can improve the user mood and user experience.

61. Go to line 105 and remove the complete gettingHelp state. **Figure 4** shows the content you should delete.

Figure 4: Content to delete from the dialog flow

```
105  gettingHelp:  
106    component: "System.Output"  
107    properties:  
108      text: |-  
109        Ordering pizza is no big deal. Just tell us what you like, or select from our menu.  
110  
111        You can give me your complete order e.g by typing e.g "I want a small pizza pepperoni".  
112  
113        If you type e.g. "I like to order pizza", then I will step you through the process.  
114  
115        So. Lets get started again.  
116  
117    transitions:  
118      return: "gettingHelp"  
119
```

- 62.** Go to the top of the dialog flow builder and click the green **+ Components** button.
- 63.** In the opened dialog box, click the **User Interface** category.
- 64.** In the menu to the left, select the **Interactive** entry.
- 65.** Set the **Insert After** item to **askPizzaSize**.
- 66.** Enable the **Remove Comments** toggle.
- 67.** Click the **Apply** button.
- 68.** Back in the dialog flow, change the interactive state name to **gettingHelp**.

- 69.** Edit the properties shown in the following table and apply the values shown in the table (including the quotation marks):

Property	Value
sourceVariableList	"pizzaType,pizzaSize"
variable	"instantAppReturn"
id	"oraclepizzahelp "
title	"Oracle Pizza Help"
description	"Use web form to complete your order"
linkLabel	"Click to complete order"
cancelLabel	"Cancel order"
prompt	"Complete your order in a web form"
cancel	"cancelPizzaOrder"
textReceived	"cancelPizzaOrder"

Your gettingHelp state should now look like it does in [Figure 5](#).

- 70.** Navigate to the transitions: element in the gettingHelp state and place the cursor at its end.
- 71.** Create a new line.
- 72.** Add next: "setPizzaType" and make sure the next: label vertically aligns with the actions: label below it.
- 73.** Go to the top of the dialog flow builder and click the green **+ Components** button.
- 74.** In the opened dialog box, select the **Variables** category.
- 75.** In the menu to the left, select the **Set variable** entry.
- 76.** Set the **Insert After** item to **gettingHelp**.

Figure 5: GettingHelp configuration to launch Oracle Instant Apps

```
gettingHelp:  
  component: "System.Interactive"  
  properties:  
    sourceVariableList: "pizzaType,pizzaSize"  
    variable: "instantAppReturn"  
    id: "oraclepizzahelp"  
    title: "Oracle Pizza Help"  
    description: "Use web form to complete your order"  
    imageUrl:  
    linkLabel: "Click to complete order"  
    cancelLabel: "Cancel order"  
    prompt: "Complete your order in a web form"  
    autoNumberPostbackActions:  
    textOnlyResponse:  
    footerText:  
    translate:  
    transitions:  
      next: "setPizzaType"  
      actions:  
        cancel: "cancelPizzaOrder"  
        textReceived: "cancelPizzaOrder"
```

- 77.** Enable the **Remove Comments** toggle.
- 78.** Click the **Apply** button.
- 79.** Back in the dialog flow, change the setVariable state name to setPizzaType.
- 80.** Configure the setPizzaType state as shown in **Figure 6**.

Figure 6: setPizzaType configuration

```
setPizzaType:  
  component: "System.SetVariable"  
  properties:  
    variable: "pizzaType"  
    value: "${instantAppReturn.value.pizzaType}"  
  transitions:  
    next: "setPizzaSize"
```

81. Repeat steps 73–80, using the settings in the following table:

Property	Value
state name	setPizzaSize
variable	"pizzaSize"
value	"\${instantAppReturn.value.pizzaSize} "
next	"askPizzaType"

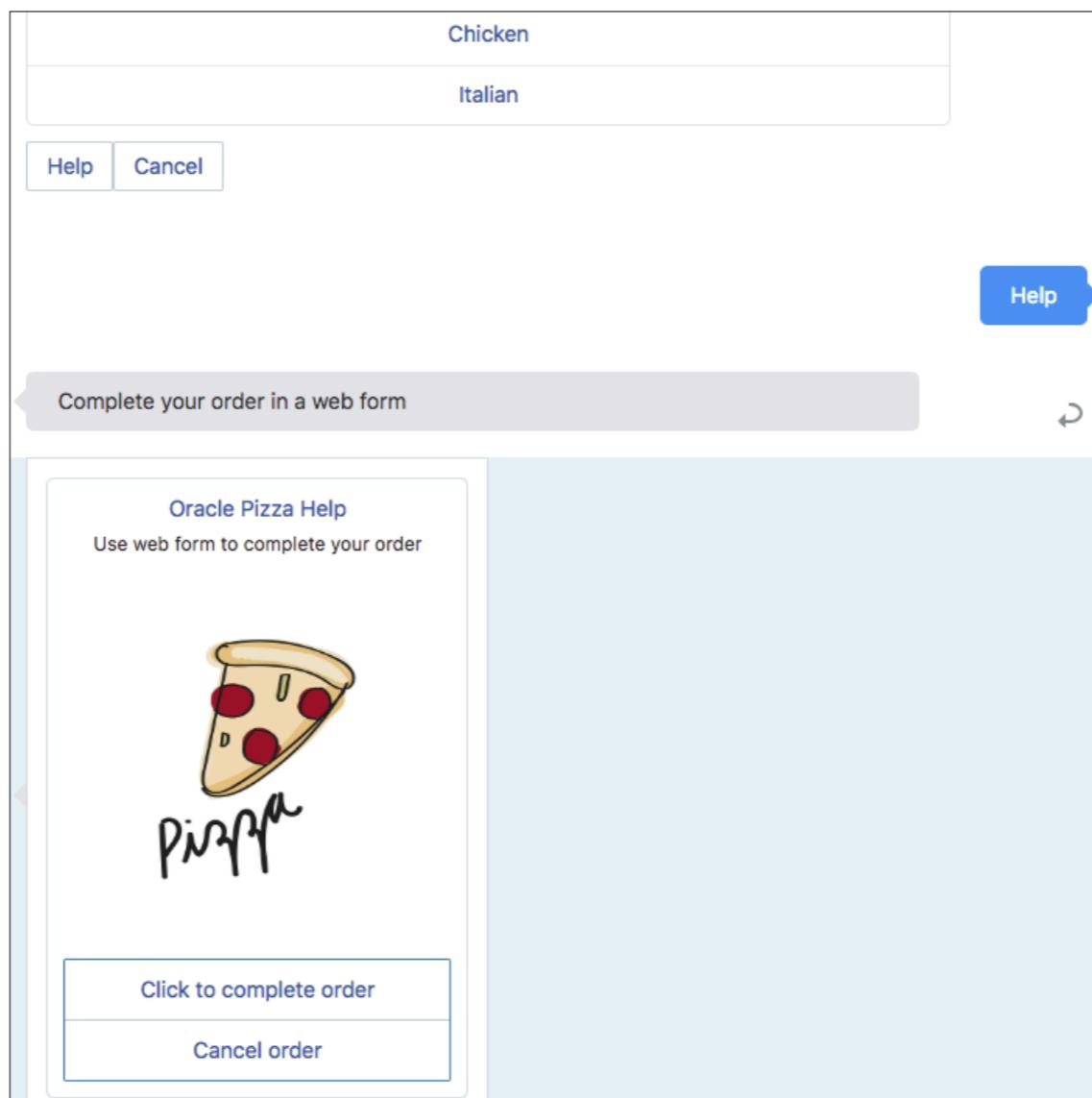
The state should look like it does in **Figure 7**.

Figure 7: setPizzaSize state

```
setPizzaSize:
  component: "System.SetVariable"
  properties:
    variable: "pizzaSize"
    value: "${instantAppReturn.value.pizzaSize}"
  transitions:
    next: "askPizzaType"
```

- 82.** You are ready to test the new help. Open the embedded conversation tester.
- 83.** Type I like to order pizza into the **Message** field and press the Enter key.
- 84.** Then, type salami into the **Message** field and press the Enter key.
- 85.** Click the **Help** button. This displays the bot response shown in **Figure 8**, so you can launch the Oracle Instant Apps web form.
- 86.** In the **Oracle Pizza Help** bot response, click the **Click to complete order** link to launch the Oracle Instant Apps form in a separate tab.
- 87.** In the opened form, select a pizza type and a pizza size.

Figure 8: Oracle Instant Apps launch UI



- 88.** Click the **Order** button.
- 89.** Close the current (Oracle Instant Apps) browser tab.
- 90.** Back in the tester, you should see the order completed and confirmed with the selected form values.
- 91.** Click the **Reset** button.

92. Type I like to order pizza into the **Message** field and press the Enter key.
93. When the pizza type list of values is displayed, select a pizza type from the list.
94. When prompted for the pizza size, type extra and press the Enter key.
95. As it did for of the pizza type list, a **Help** button is displayed. Click the **Help** button.
96. The Oracle Instant Apps form is displayed with the selected pizza type set, but with no size selected. Select a pizza size.
97. Click the **Order** button.
98. Close the current (Oracle Instant Apps) browser tab.
99. Again, the tester window should show the selected pizza type and size.
100. Click the **Reset** button.
101. Close the embedded tester window.

What you just did: In this part of the hands-on exercise, you implemented user help in the form of an instant app. Now, when a user fails to provide a valid data input when prompted for the pizza type or pizza size, the displayed **Help** button launches an instant app, which allows the user to use a web-based input form to complete the order. Any value that the user provides within the bot conversation is already set in the form. Using a web form for providing help unlocks users by providing them an environment they are familiar with. When the user closes the instant app by clicking the **Order** button, control is passed back to the bot along with the user-selected values. The `setPizzaType` and `setPizzaSize` states copy the values returned from the Oracle Instant Apps form to the dialog flow variables used within the chatbot conversation.

HUMAN AGENT INTEGRATION

Another option to assist users in escaping the validation loop and to successfully complete a task (such as pizza ordering in this hands-on exercise) is to pass the con-

versation to a human agent. Oracle Digital Assistant integrates with human agent support such that the same conversation and channel can be used for a human to interact with the user. After the human agent helps the user, control is then passed back to the bot.

With human agent integration, you now know about three levels of escalation for unlocking users in a bot conversation: written help, Oracle Instant Apps, and human agent integration.

INTELLIGENT ROUTING IN ORACLE DIGITAL ASSISTANT

Imagine that, when asked to select a pizza type, a user types Ask Trust Bank, what is my balance, just to check if he can afford a big pizza with everything on it. In this case, the user is not locked because he understands the order process for a pizza. It's just that the user has a question he needs to get resolved first. Here, the expected bot behavior is that, temporarily, the user is allowed to exit the existing conversation.

Subflows, which in Oracle Digital Assistant are referred to as *nonsequitur navigation*, are automatically handled by Oracle Digital Assistant smart request routing. If Oracle Digital Assistant has a “Trust Bank” skill defined, the user is able to exit and return to the pizza ordering conversation. So there is nothing more that bot designers need to code for.

CONCLUSION

In this article, you improved the user experience of a skill in Oracle Digital Assistant by providing options for users to continue the bot conversation when they don't know how to provide correct input for a bot response. Conversational design is an

important skill to learn for any bot developer because it helps keep users on a happy path when they are working with a bot. Understanding how to allow users to escape the validation loop and to get help is just one important design discipline.

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



ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

TRY Oracle Digital Assistant.

DOWNLOAD the bot for this article.

LEARN more about Oracle Digital Assistant.

READ an Oracle TechExchange article about escaping the validation loop.

**ORACLE AUTONOMOUS TRANSACTION PROCESSING**

Getting Started with Autonomous

Connect to Oracle Autonomous Transaction Processing with multiple programming languages.

By now you've probably heard about [Oracle Autonomous Transaction Processing](#). If not, follow the link for more information.

One major feature of Oracle Autonomous Transaction Processing is enhanced security. For example, Oracle Autonomous Transaction Processing uses machine learning and automation to improve its own security, but it also takes advantage of an Oracle wallet. There are many ways you can use an Oracle wallet to enhance the security of database connections, but with Oracle Autonomous Transaction Processing, it is already set up, packaged, and ready to go.

I am going to walk you through the process of preparing to connect securely to your Oracle Autonomous Transaction Processing database instance with an Oracle wallet, followed by some code examples in a few languages you can use to test the connection.

PREREQUISITES

Here are the prerequisites for following along with the steps in this article:

- Have an Oracle Autonomous Transaction Processing instance provisioned and running. (If you don't have a current instance, you can [sign up for a free trial](#).)
- Have a database schema and password created that you can use for testing.
- Have access to the Oracle Autonomous Transaction Processing service panel or have someone with access available to help.
- Download and install [Oracle Database Instant Client](#).
- (Optional but a good idea) Have [SQLcl](#) or [Oracle SQL Developer](#) installed to verify the connection in a neutral environment.
- Download and install the database driver for your preferred language. Make sure the driver you choose uses Oracle Database Instant Client to make its connection. I will provide links to the language-specific drivers used in my examples.

DOWNLOAD CLIENT CREDENTIALS (ORACLE WALLET)

With the prerequisites complete, download the client credentials for your Oracle Autonomous Transaction Processing database instance.

1. Go to the Oracle Autonomous Transaction Processing Database Details page for your Oracle Autonomous Transaction Processing database instance.
2. Click the **DB Connection** button, shown in [Figure 1](#).
3. Click the **Download** button, shown in [Figure 2](#).

The client credentials zip file you download will contain a `tnsnames.ora` file that includes the **TNS Name** values listed in the **Connection Strings** panel. You can refer to the [documentation](#) for an explanation of when to use each of these. For my examples, I will be using the **BlaineATP_MEDIUM TNS Name** value.

Figure 1: Making the DB connection

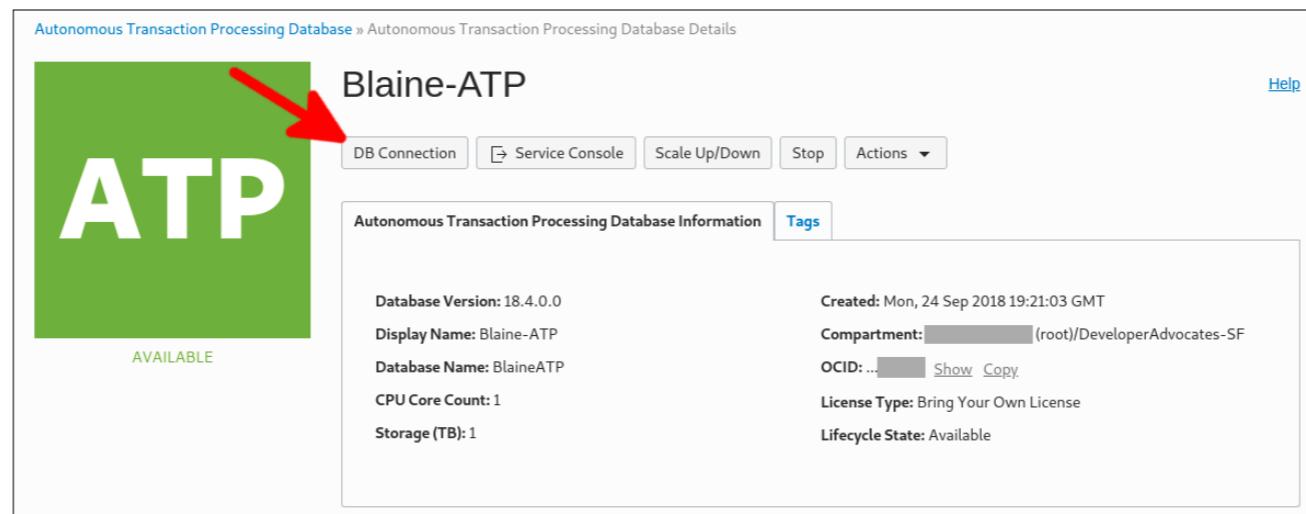


Figure 2: Downloading the client credentials

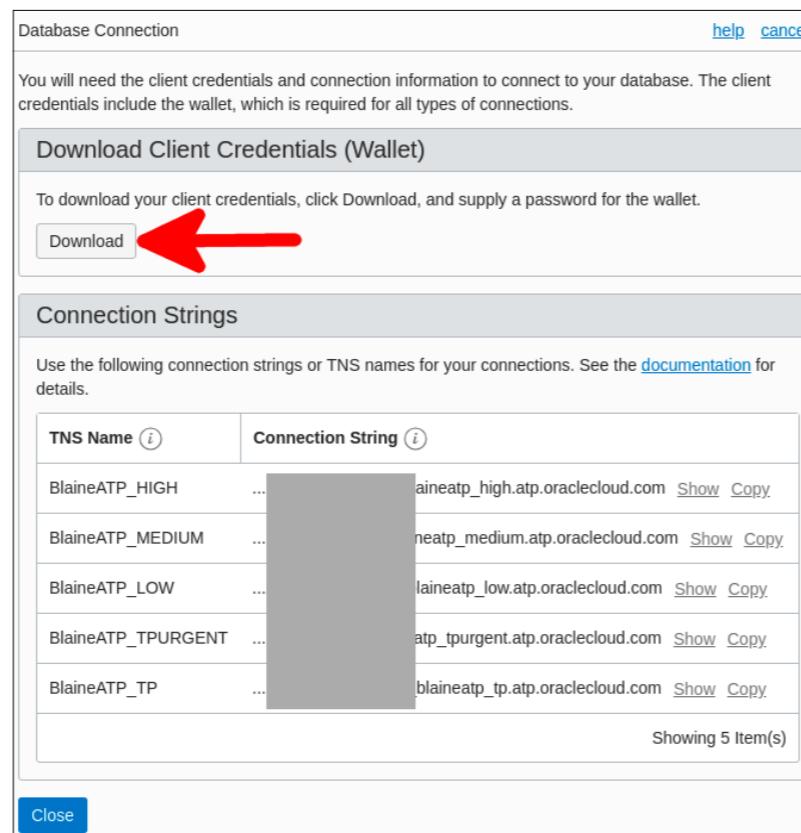


Figure 3: Creating a password for your client credentials

The screenshot shows a web-based form titled "Download Wallet". At the top right are "help" and "cancel" links. The main text area explains that database connections use a secure connection and require a wallet file for configuration. It instructs the user to create a password for this wallet, noting that some clients require both the wallet and password while others auto-login. Below this are two input fields: "PASSWORD" and "CONFIRM PASSWORD", each with a corresponding empty input field below it. At the bottom is a "Download" button.

4. Enter a password (see **Figure 3**), and click **Download**.

Remember this password. If you lose it, you will need to download a new credentials file.

5. Save the file in a secure location. Remember, this file can be used to access your database, so keep it secure.
6. Create a directory, and extract the client credentials zip file into that directory.

You should now have the following files.

- cwallet.sso
- ewallet.p12
- keystore.jks
- ojdbc.properties
- sqlnet.ora
- tnsnames.ora
- truststore.jks

7. Edit the sqlnet.ora file. Set the DIRECTORY value to the directory used in step 6. For example:

```
(DIRECTORY="/home/demouser/projects/ATP/Wallet_Creds")
```

TEST THE CONNECTION: OPTIONAL BUT RECOMMENDED

Now test the connection from your Oracle SQLcl or Oracle SQL Developer tool.

Oracle SQLcl. To test the connection from Oracle SQLcl, do the following:

1. Start Oracle SQLcl in nolog mode.

```
sql /nolog
```

2. Set the location of your credentials zip file.

```
set cloudconfig /home/demouser/projects/ATP/Wallet_Creds/client_credentials.zip
Operation is successfully completed.
Using temp directory:/tmp/oracle_cloud_config903805845690230771
```

3. Connect with a schema/password that is safe for testing.

```
connect myschema/mypassword@BlaineATP_MEDIUM
Connected.
```

4. If all goes well, you should now be connected and able to run a test query.

```
select 'Connected to Oracle Autonomous Transaction Processing from SQLcl!' "Test It"
from dual;
```

Test It

```
Connected to Oracle Autonomous Transaction Processing from SQLcl!
```

5. Exit Oracle SQLcl.

```
exit
```

```
Disconnected from Oracle Database 18c Enterprise Edition
```

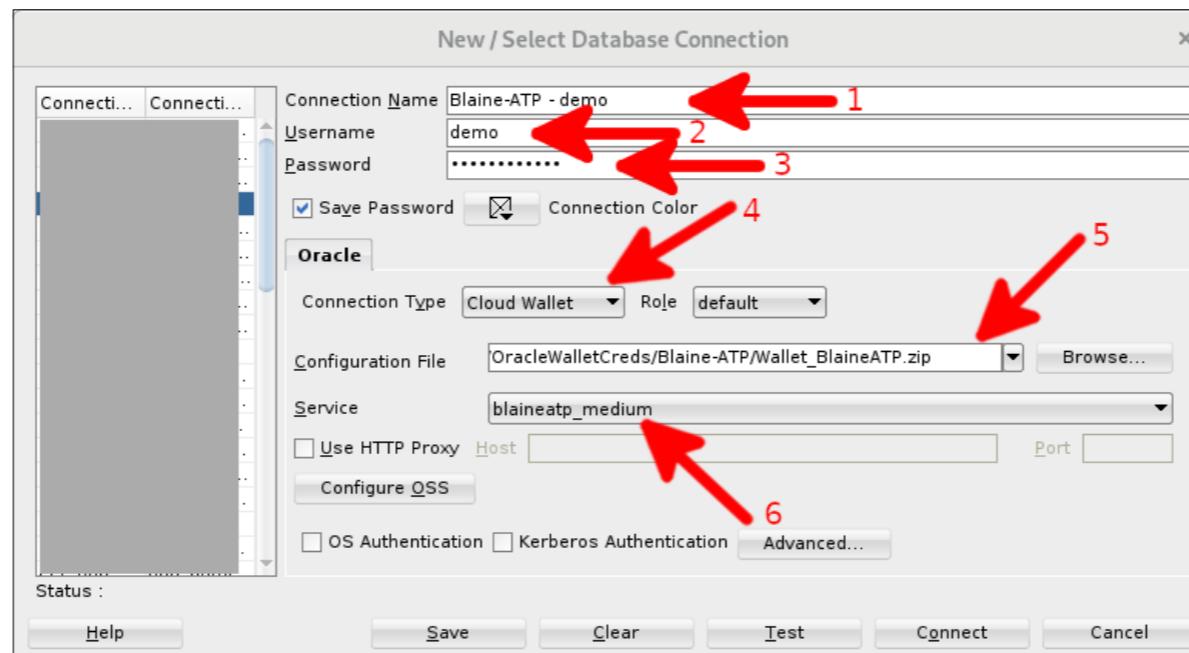
```
Release 18.0.0.0 - Production
```

```
Version 18.4.0.0.0
```

Oracle SQL Developer. To test the connection from Oracle SQL Developer, do the following:

1. Create a connection and give it a connection name, as shown in [Figure 4](#).
2. Enter the username for a schema that's safe to use for testing.
3. Enter the password.
4. Choose **Cloud Wallet** from the **Connection Type** menu, as shown in [Figure 4](#).
5. For **Configuration File**, select the zip file for the client credentials you downloaded earlier.
6. Select the service you want to use for your connection. (This list is populated from the Oracle wallet.)

Figure 4: Entering the connection name, username, password, connection type, configuration file, and service in Oracle SQL Developer



7. Click **Test**. If the test is a success, you can save the connection.
8. (Optional) Open a SQL worksheet for that connection, and run a test query.
If the Oracle SQL Developer connection or query tests were a success, you know that your client credentials file is valid and that you are able to connect to the Oracle Autonomous Transaction Processing database instance.

GENERIC INSTRUCTIONS

To connect to Oracle Autonomous Transaction Processing with most programming languages, do the following:

1. Download and install the Oracle Database driver for that language. Choose a driver that connects through Oracle Database Instant Client.

2. Set the TNS_ADMIN environment variable to point to the directory from which you extracted the client credentials files.
3. Create a database connection with the username, password, and service used in the above tests.
4. Use the connection as you would any other Oracle Database connection in your application.

LANGUAGE-SPECIFIC EXAMPLES

For the following examples, I have defined environment variables for TNS_ADMIN and the user, password, service, and connect string.

```
export TNS_ADMIN="/home/demouser/projects/ATP/Wallet_Creds"  
export OADB_USER='demo'  
export OADB_PW='demoPassword'  
export OADB_SERVICE='BlaineATP_MEDIUM'  
export OADB_CONNECTSTRING="$OADB_USER/$OADB_PW@$OADB_SERVICE"
```

Python. To connect to the Oracle Autonomous Transaction Processing database instance with Python, do the following:

1. Download and install [the cx_Oracle driver](#).
2. Create the python_demo.py file, with the following code:

```
import cx_Oracle  
import os  
  
con = cx_Oracle.connect(os.environ['OADB_USER'], os.environ['OADB_PW'],
```

```
os.environ['OADB_SERVICE'])

cur = con.cursor()

cur.execute("select 'Connected to Oracle Autonomous Transaction Processing from
Python!' from dual")
res = cur.fetchall()
for row in res:
    print(row)

cur.close()
con.close()
```

3. Run the `python_demo.py` file in Python with the following command:

```
python python_demo.py
```

The following response confirms your connection from Python to Oracle Autonomous Transaction Processing:

```
('Connected to Oracle Autonomous Transaction Processing from Python!',)
```

Node.js. To connect to the Oracle Autonomous Transaction Processing database instance with Node.js, do the following:

1. Download and install [the Node-oracledb driver](#).

2. Create the nodeDemo.js file, with the following code:

```
var oracledb = require('oracledb');

oracledb.getConnection({
    user: process.env.OADB_USER,
    password: process.env.OADB_PW,
    connectString: process.env.OADB_SERVICE
},
function(err, connection) {
    if (err) {
        console.error(err.message);
        return;
    }
    connection.execute(
        "select 'Connected to Oracle Autonomous Transaction Processing from Node.js!'"
from dual",
        function(err, result) {
            if (err) {
                console.error(err);
                return;
            }
            console.log(result.rows);
        });
});
```

3. Run the nodeDemo.js file in Node.js with the following command:

```
node nodeDemo.js
```

The following response confirms your connection from Node.js to Oracle Autonomous Transaction Processing:

```
[ [ 'Connected to Oracle Autonomous Transaction Processing from Node.js!' ] ]
```

Ruby. To connect to the Oracle Autonomous Transaction Processing database instance with Ruby, do the following:

1. Download and install [the ruby-oci8 driver](#).
2. Create the ruby-demo.rb file, with the following code:

```
require 'oci8'

con = OCI8.new(ENV['OADB_USER'],
               ENV['OADB_PW'],
               ENV['OADB_SERVICE']);

statement = "select 'Connected to Oracle Autonomous Transaction Processing from
Ruby!' from dual";
cursor = con.parse(statement)
cursor.exec
cursor.fetch() {|row|
```

```
    printf "%s\n", row[0]
}
```

3. Run the ruby-demo.rb file in Ruby with the following command:

```
ruby ruby-demo.rb
```

The following response confirms your connection from Ruby to Oracle Autonomous Transaction Processing:

```
Connected to Oracle Autonomous Transaction Processing from Ruby!
```

Go. To connect to the Oracle Autonomous Transaction Processing database instance with Go, do the following:

1. Download and install [the goracle driver](#).
2. Create the goDemo.go file, with the following code:

```
package main

import (
    "database/sql"
    _ "gopkg.in/goracle.v2"
    "fmt"
    "os"
)
```

```
func main() {
    connectString := os.Getenv("OADB_CONNECTSTRING")

    db, err := sql.Open("goracle", connectString)
    if err != nil {
        panic(err)
    }
    defer db.Close()

    row := db.QueryRow("select 'Connected to Oracle Autonomous Transaction Processing
from Go!' from dual")

    var strCol string
    err = row.Scan(&strCol)

    if err != nil {
        panic(err)
    }

    fmt.Printf("%s\n", strCol)
}
```

3. Run the goDemo.go file in Go with the following command:

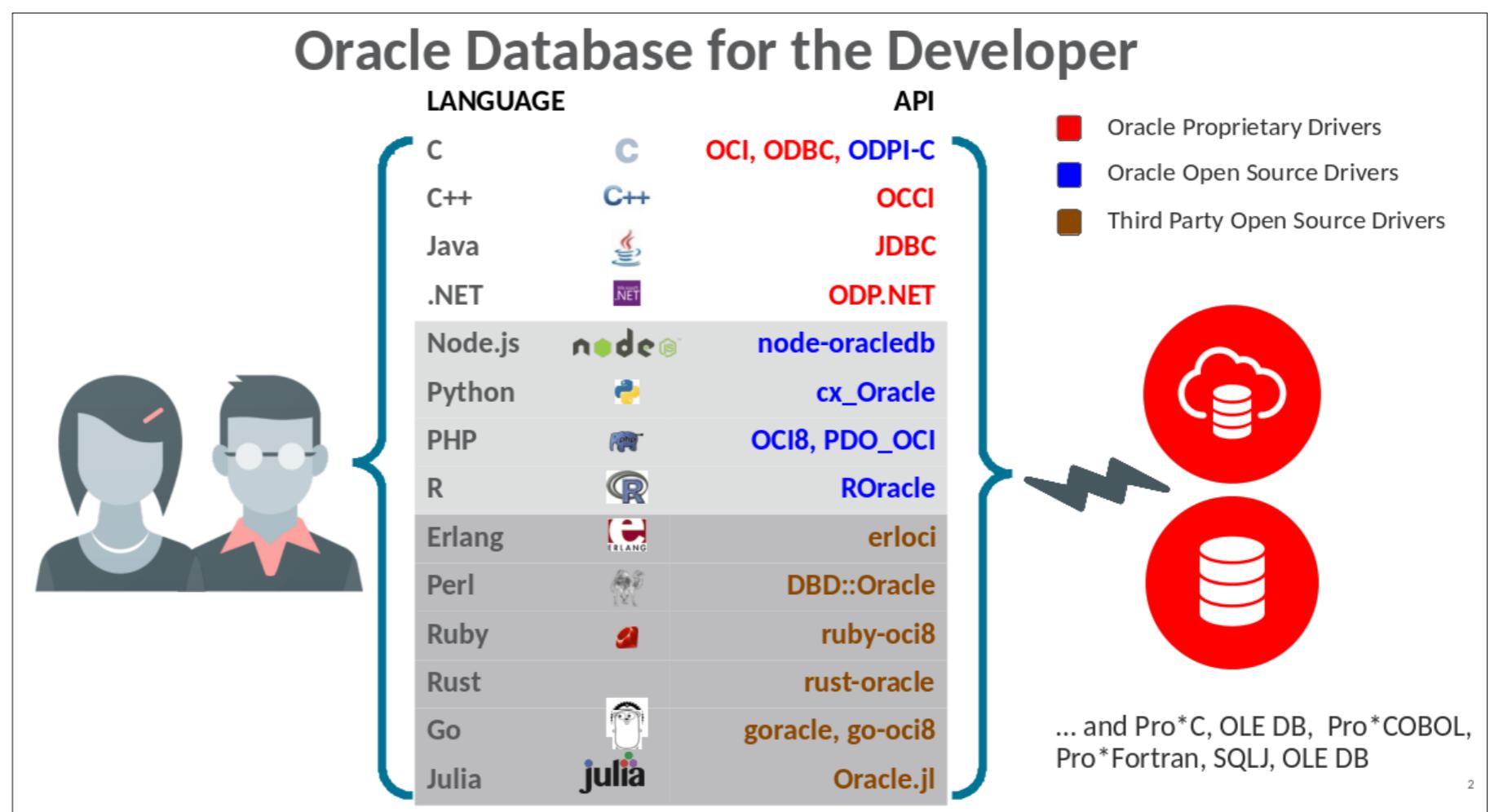
```
go build goDemo.go
./goDemo
```

The following response confirms your connection from Go to Oracle Autonomous Transaction Processing:

Connected to Oracle Autonomous Transaction Processing from Go!

Other languages. Figure 5 contains a list of some of the Oracle Database drivers available for a few other languages—there are many more.

Figure 5: Oracle Database languages and drivers supported



I have not personally tested an Oracle Autonomous Transaction Processing connection with every one of the drivers in [Figure 5](#), but the methods used in this article should work for all of them. If you have difficulty making the connection with one of these drivers, please reach out to me, and I will do my best to help make it work. 

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



ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about
Oracle Autonomous
Transaction Processing.

TRY Oracle Autonomous
Transaction Processing.



ORACLE DATABASE

Easy Initializing for Records and Arrays

Use the new qualified expressions to create functions when and where you need them.

As anyone who has followed me over the years knows, I like the Oracle PL/SQL language. Sure, it's not the newest, coolest kid on the block (and it probably *never* was). But then, neither am I. PL/SQL is, on the other hand, a delightfully straightforward, easy-to-learn and -write language that serves its purpose well: implement APIs to data (SQL) and business logic, right inside the database.

To serve that purpose, of course, PL/SQL needs to support lots of "big ticket" functionality: supersmooth and easy native dynamic SQL, canonicalization of static SQL to minimize the need for hard-parsing, invoker rights (AUTHID CURRENT_USER), and so much more.

But I must confess: The features of PL/SQL I love the most are the relatively little things that make it easy for me to be productive as I churn out packages (and, yes, I

still do write lots of PL/SQL code, most recently for the [Oracle Dev Gym](#), an active-learning website featuring quizzes, workouts, and classes).

And that's why my favorite PL/SQL enhancement in Oracle Database 18c is the qualified expression. If you are not familiar with that term, then how about "constructor function"?

Ever since Oracle Database introduced the object-relational model, in version 8, PL/SQL developers been able to use constructor functions to, well, *construct* nested tables, varrays, and object type instances. A *constructor function* is a function created implicitly by Oracle Database that has the same name as its type. The easiest way to understand constructor functions is to look at an example.

To initialize a nested table of integers with three elements, I can do this:

```
DECLARE
    TYPE numbers_t IS TABLE OF NUMBER;
    l_numbers numbers_t := numbers_t (1, 2, 3 * 3);
BEGIN
    DBMS_OUTPUT.put_line (l_numbers.COUNT);
END;
```

So `numbers_t` is the name of the type, but it is also the name of a function. For a nested table or varray type, I simply pass in a list of expressions, and the function returns a collection populated with those values. If I don't pass in any values and simply call the constructor function, then the collection will be initialized but will be empty.

But before Oracle Database 18c, if I was using an associative array (aka, an

index-by table), this was not allowed. Instead, I had to assign elements to the array, one at a time, as in

```
DECLARE
    TYPE numbers_t IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
    l_numbers numbers_t;
BEGIN
    l_numbers (1) := 100;
    l_numbers (2) := 1000;
    l_numbers (3) := 10000;
END;
```

I had the same challenge with populating values of fields in a record: Either I wrote my own function to create and return a record, or I wrote field-by-field assignments in the executable section of my block. Here's an example:

```
DECLARE
    TYPE person_rt IS
        RECORD (last_name VARCHAR2(100), hair_color VARCHAR2(100));
    l_person person_rt;
BEGIN
    l_person.last_name := 'Feuerstein';
    l_person.hair_color := 'Not Applicable';
END;
```

That's cumbersome, irritating, and...as of Oracle Database Release 18c, you and I don't have to bother with that sort of thing anymore.

Now, *any PL/SQL value* can be provided by a qualified expression, just like a constructor provides an abstract data type value.

PL/SQL uses the terms *qualified expression* and *aggregate* rather than the SQL term *type constructor*, but the functionality is the same. Qualified expressions improve program clarity and developer productivity, by providing the ability to declare and define a complex value in a compact form where the value is needed.

A qualified expression combines expression elements to create values of a RECORD type or associative array type (both integer- and string-indexed). Qualified expressions use an explicit type indication to provide the type of the qualified item. This explicit indication is known as a *typemark*.

I've put together a couple of Oracle Live SQL scripts to make it easy for you to play around with this great feature:

- [Qualified Expressions for Records \(aka, record constructors\)](#)
- [Qualified Expressions for Associative Arrays \(aka, collection constructors\)](#)

But, hey, as long as you're here, let's go exploring!

QUALIFIED EXPRESSIONS FOR RECORDS, USING POSITIONAL NOTATION

This example uses positional notation to associate values with fields. This means that each expression in the list is assigned to the field in the same position (for example, Savannah is assigned to habitat_type).

DECLARE

```
TYPE species_rt IS RECORD (
    species_name        VARCHAR2 (100),
```

```
    habitat_type          VARCHAR2 (100),
    surviving_population  INTEGER);

l_elephant  species_rt := species_rt ('Elephant', 'Savannah', '10000');

PROCEDURE display_species (
    species_in species_rt DEFAULT species_rt ('Not Set', 'Global', 0))
IS
BEGIN
    DBMS_OUTPUT.put_line ('Species: ' || species_in.species_name);
    DBMS_OUTPUT.put_line ('Habitat: ' || species_in.habitat_type);
    DBMS_OUTPUT.put_line ('# Left: ' || species_in.surviving_population);
END;
BEGIN
    display_species (species_in => l_elephant);

    /* Use the default */
    display_species ();
END;
/
```

```
Species: Elephant
Habitat: Savannah
# Left: 10000
Species: Not Set
```

```
Habitat: Global
```

```
# Left: 0
```

Note that in this example, I used the qualified expression for the record type—`species_rt`—in two places:

1. To initialize the `l_elephant` variable
2. As the default value for my `species_in` parameter

QUALIFIED EXPRESSIONS FOR RECORDS, USING NAMED NOTATION

Named notation has long been a feature of parameter lists in PL/SQL: You use the `=>` operator to associate an argument value with its formal parameter. See the first call to `display_species` in the previous code block for an example.

You can now also use named notation in a record's qualified expression. In the block below, I provide both the names of the fields and their values.

```
DECLARE
```

```
    TYPE species_rt IS RECORD (
        species_name          VARCHAR2 (100),
        habitat_type          VARCHAR2 (100),
        surviving_population  INTEGER);

    l_elephant  species_rt
        := species_rt (species_name      => 'Elephant',
                      surviving_population => '10000',
                      habitat_type       => 'Savannah');
```

```
BEGIN
    DBMS_OUTPUT.put_line ('Species: ' || l_elephant.species_name);
END;
/
```

Species: Elephant

I suggest always using named notation in your qualified expressions. This enables you to change the order of the individual assignments, which might be useful for emphasizing key values but also to improve the overall readability of your code.

QUALIFIED EXPRESSIONS FOR ASSOCIATIVE ARRAYS

When you invoke a qualified expression for a record, you can choose between named and positional notation. With associative arrays, you must *always* specify the index value (integer or string) for each expression you want to stuff into the array, as in

```
DECLARE
    TYPE ints_t IS TABLE OF INTEGER
        INDEX BY PLS_INTEGER;

    l_ints    ints_t := ints_t (1 => 55, 2 => 555, 3 => 5555);

BEGIN
    FOR indx IN 1 .. l_ints.COUNT
    LOOP
```

```
DBMS_OUTPUT.put_line (l_ints (indx));
END LOOP;
END;
/
55
555
5555
```

And just as you can change the order of a record's fields in the qualified expression when you use named notation, you don't need to specify index values in order with an array's qualified expression, as you can see here:

```
DECLARE
  TYPE ints_t IS TABLE OF INTEGER
    INDEX BY PLS_INTEGER;

  l_ints  ints_t := ints_t (2 => 55, 1 => 555, 3 => 5555);
BEGIN
  FOR indx IN 1 .. l_ints.COUNT
  LOOP
    DBMS_OUTPUT.put_line (l_ints (indx));
  END LOOP;
END;
/
```

```
55  
555  
5555
```

In case you're wondering, you *cannot* use named notation with the constructor function of a nested table or varray. If you try to do so, the PL/SQL compiler will reject it with a PLS-00306: wrong number or types of arguments in call error.

That's not too surprising, since nested tables and varrays are always populated *densely* from index value 1. This means that there are no undefined gaps between the lowest and the highest defined index values. (This is an initial requirement; you can introduce a gap between values in a nested table later, using the DELETE method).

Values in associative arrays, on the other hand, can be dense or *sparse* (with at least one undefined index value between the lowest and the highest). This is especially and obviously the case for string-indexed associative arrays (nested tables and varrays support only integer indexes).

So with qualified expressions for associative arrays, you can specify whatever index values you'd like. For example:

```
DECLARE  
    TYPE ints_t IS TABLE OF INTEGER  
        INDEX BY PLS_INTEGER;  
  
    l_ints    ints_t := ints_t (600 => 55, -5 => 555, 200000 => 5555);  
    l_index pls_integer := l_ints.first;
```

```
BEGIN
    WHILE l_index IS NOT NULL
    LOOP
        DBMS_OUTPUT.put_line (l_index || ' => ' || l_ints (l_index));
        l_index := l_ints.NEXT (l_index);
    END LOOP;
END;
/
-5 => 555
600 => 55
200000 => 5555
```

And with string-indexed arrays, “the dictionary’s the limit.” You can use any string you’d like as an index value.

```
DECLARE
    TYPE by_string_t IS TABLE OF INTEGER
        INDEX BY VARCHAR2(100);

    l_stuff    by_string_t := by_string_t ('Steven' => 55, 'Loey' => 555, 'Juna' => 5555);
    l_index varchar2(100) := l_stuff.first;
BEGIN
    DBMS_OUTPUT.put_line (l_stuff.count);

    WHILE l_index IS NOT NULL
```

```
LOOP
    DBMS_OUTPUT.put_line (l_index || ' => ' || l_stuff (l_index));
    l_index := l_stuff.NEXT (l_index);
END LOOP;
END;
/
3
Juna => 5555
Loey => 555
Steven => 55
```

Furthermore, those index values do not need to be literals. They can be expressions; they simply must resolve explicitly or implicitly to the correct type (integer or string).

```
DECLARE
    TYPE by_string_t IS TABLE OF INTEGER
        INDEX BY VARCHAR2 (100);

    l_stuff    by_string_t :=
        by_string_t (UPPER ('Grandpa Steven') => 55,
                    'Loey' || 'Juna' => 555,
                    SUBSTR ('Happy Family', 7) => 5555);

    l_index varchar2(100) := l_stuff.first;
```

```
BEGIN
    DBMS_OUTPUT.put_line (l_stuff.count);

    WHILE l_index IS NOT NULL
    LOOP
        DBMS_OUTPUT.put_line (l_index || ' => ' || l_stuff (l_index));
        l_index := l_stuff.NEXT (l_index);
    END LOOP;
END;
/
3
Family => 5555
GRANDPA STEVEN => 55
LoeyJuna => 555
```

Finally, there's nothing stopping you from using qualified expressions for arrays as well as records, which comes in really handy when you have a collection of records!

```
DECLARE
    TYPE species_rt IS RECORD (
        species_name VARCHAR2 (100),
        habitat_type VARCHAR2 (100),
        surviving_population INTEGER
    );
```

```
TYPE species_t IS TABLE OF species_rt INDEX BY PLS_INTEGER;

l_species    species_t :=
    species_t (
        2 => species_rt ('Elephant', 'Savannah', '10000'),
        1 => species_rt ('Dodos', 'Mauritius', '0'),
        3 => species_rt ('Venus Flytrap', 'North Carolina', '250'));
```

```
BEGIN
```

```
    FOR indx IN 1 .. l_species.COUNT
```

```
    LOOP
```

```
        DBMS_OUTPUT.put_line (l_species (indx).species_name);
```

```
    END LOOP;
```

```
END;
```

```
/
```

Dodos

Elephant

Venus Flytrap

LESS—AND MORE-READABLE—CODE

PL/SQL has always been an incredibly readable language. It relies heavily on common English language keywords. When you make use of all the modular features of PL/SQL, it is quite possible to produce a block of code that can almost literally be read like a book.

Certainly, prior to Oracle Database 18c, you could write and then read all those inline assignments. You could also write your own custom functions—one for each

distinct record and array type—and use them in place of inline assignments.

Now, with Oracle Database 18c, you can write even less code and end up with a code block that is even *more* readable. Let Oracle Database do even more of the heavy lifting: Use qualified expressions for your records and arrays, and focus even more of your time on implementing business logic right there in the database! ☺



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 qualified expressions documentation: “Qualified Expressions Overview.”

LEARN more about qualified expressions: “Qualified Expressions in PL/SQL in Oracle Database 18c.”

TRY Oracle Live SQL scripts: Qualified Expressions for Records (aka, record constructors).

Qualified Expressions for Associative Arrays (aka, collection constructors).

**ORACLE DATABASE**

Using Oracle Machine Learning Notebooks

Set up and run machine learning in Oracle Autonomous Data Warehouse.

Acme Corp., a leading seller of high-end merchandise on the internet, is currently in a bind. Sales are dropping, advertising costs are soaring, and the CEO is understandably disturbed. She calls a meeting of the top people from the data, products, marketing, and sales departments to brainstorm about how to increase sales. Everyone is confident that Jane, the new chief data scientist at Acme, will be able to come up with something.

Violet, the marketing manager, starts off by saying that advertising costs have gone up. To reduce them, Acme must know exactly which segments of the customer base will buy what products and then advertise to them directly instead of using the current shotgun approach of trying to reach everyone. She wants to know if there is a quick way to do that.

In the past, Acme's data scientists had to download data from the database to their local machines to analyze it with the R and Python languages. But the data was often stale, because it had to be moved from the database, and it required strong expertise in R and Python. Most data users at Acme are familiar with SQL, and that's the language they want to use. The data users want to know if that is possible without learning Python and R, but they still want the ability to use these languages.

Jane says they can use SQL without learning Python and R, but they can use those languages if they want to. Acme uses Oracle Autonomous Data Warehouse, which has built-in analytics and machine learning modules. This enables Jane to perform analytics computations inside the database without moving the data. In this article, you will learn how Jane does the data analysis.

SETUP

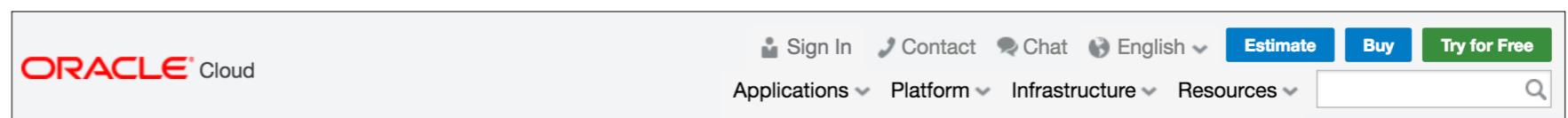
To demonstrate the concepts used by Jane, this article uses the SH sample schema that comes with Oracle Autonomous Data Warehouse. There are two tables in this schema: SALES, which shows sales fact data, and SUPPLEMENTARY_DEMOGRAPHICS, which shows the demographics data for each customer. Acme uses a loyalty card that offers discounts to returning customers, hoping they will spend more. Jane is tasked to figure out which specific demographic data items will most influence customers to spend more.

This article assumes that you know a little bit about Oracle Autonomous Data Warehouse, and it's beyond its scope to explain all Oracle Autonomous Data Warehouse concepts. If you are already familiar with Oracle Autonomous Data Warehouse and are using it now, you can skip ahead to the "Oracle Machine Learning Notebooks" section.

If you have access to Oracle Autonomous Data Warehouse, great; but if you don't

and want to follow the steps in this article, Oracle makes it easy for you. You can sign up for a trial account at cloud.oracle.com, which will enable you to spend up to US\$300 in free credits from Oracle. After you sign up, you will receive an email from Oracle Cloud with a temporary password. **Figure 1** shows the top of the Oracle Cloud homepage.

Figure 1: Oracle Cloud homepage top banner



If you want to start a cloud trial, click **Try for Free**. You will be asked to designate a name for your cloud account, which needs to be unique. In my case, I used the name arup.

After you get your password via email, click **Sign In** (see **Figure 1**), sign in with the temporary password, and then immediately change it to the password you want. Henceforth you will see the sign-in screen shown in **Figure 2**.

Always remember to choose **Cloud Account with Identity Cloud Service** and the cloud account name (**arup** in my case).

Once logged in, you will see a screen similar to the one in **Figure 3**. Click the hamburger icon (≡) to the left of the words **Oracle Cloud My Services**, which will pull up a menu on the left (also shown in **Figure 3**). Click **Oracle Autonomous Data Warehouse**, and you will see a screen similar to **Figure 4**.

Click the **Create Autonomous Data Warehouse** button. You will see a very small set of questions. Answer the questions, and click **Create**. The data warehouse will be created in a matter of minutes. After the data warehouse is created, you will see

Figure 2: Sign-in screen

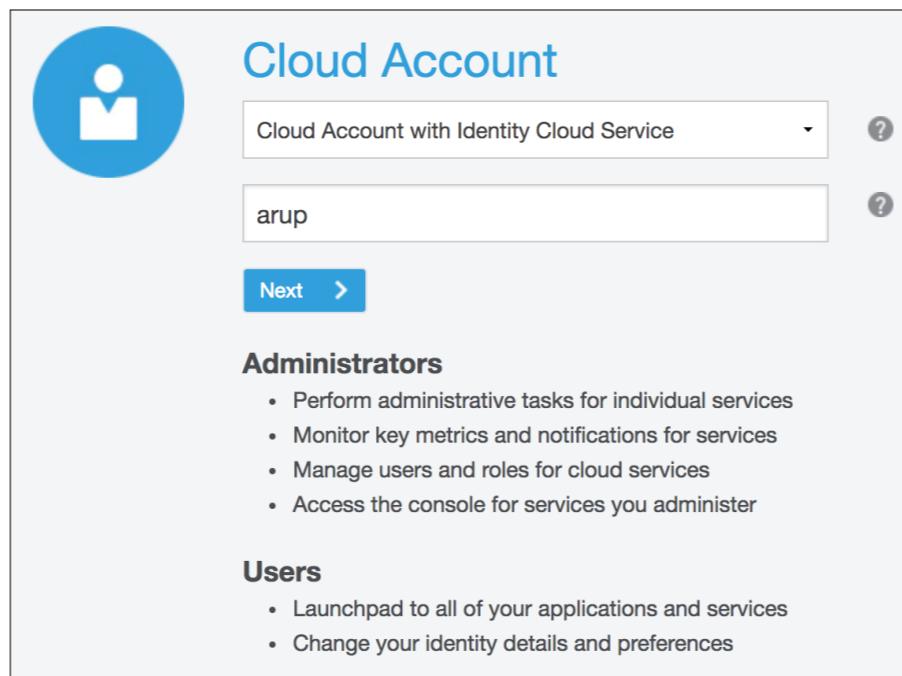


Figure 3: Oracle Cloud dashboard

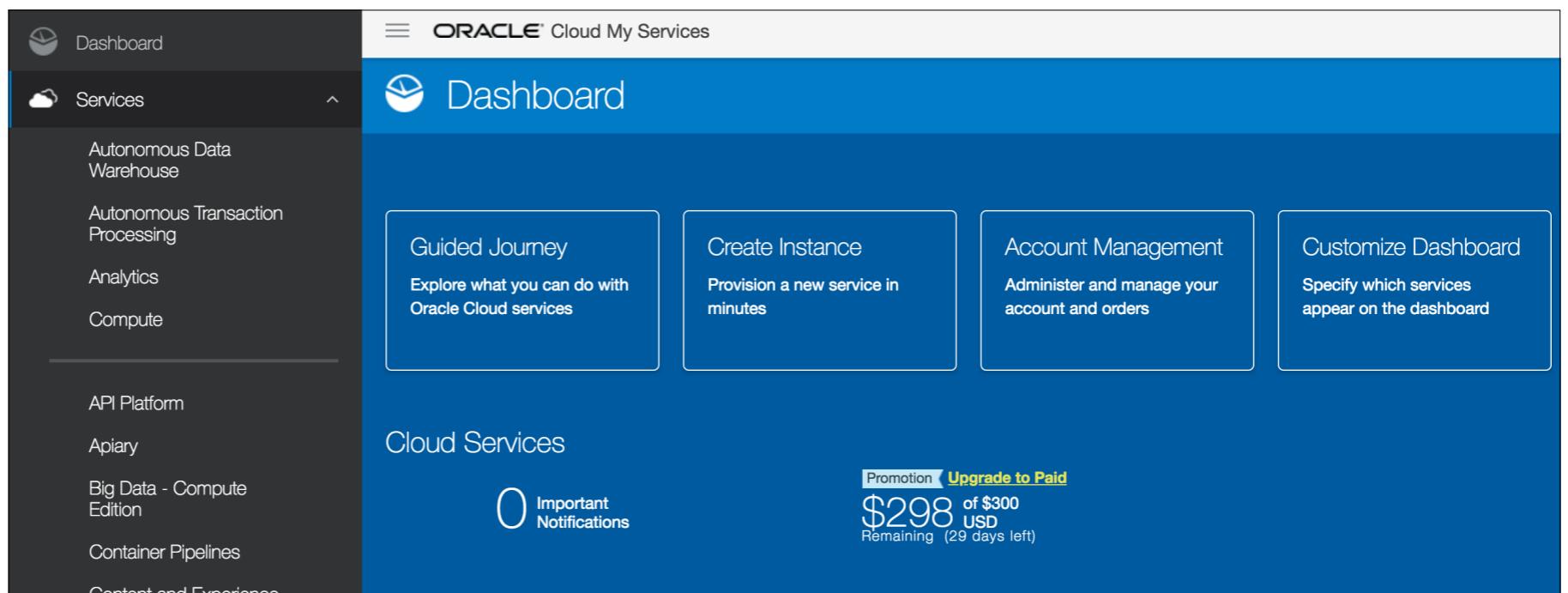
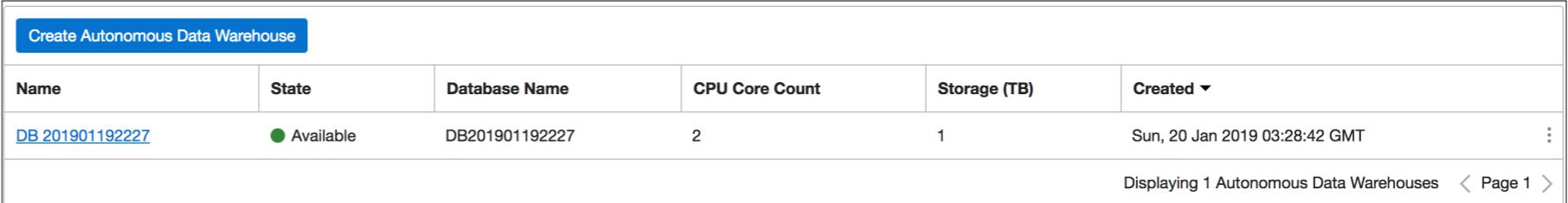
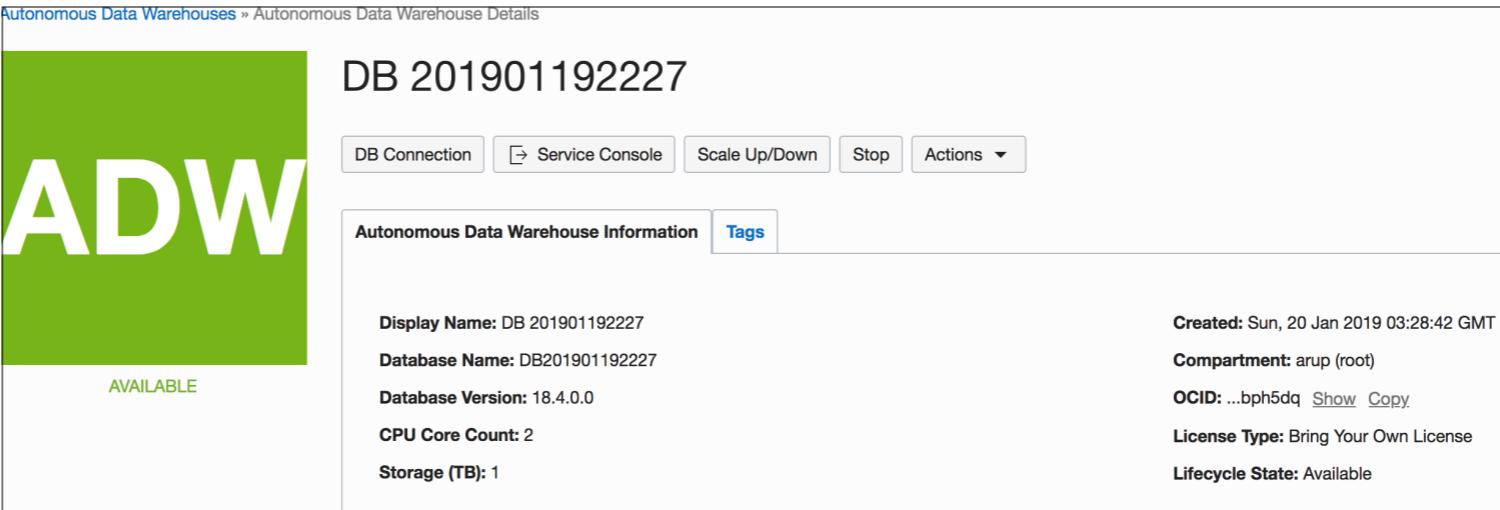


Figure 4: Oracle Autonomous Data Warehouse dashboard


Name	State	Database Name	CPU Core Count	Storage (TB)	Created	⋮
DB 201901192227	● Available	DB201901192227	2	1	Sun, 20 Jan 2019 03:28:42 GMT	
Displaying 1 Autonomous Data Warehouses						Page 1

Figure 5: Autonomous Data Warehouse management screen


The screenshot shows the details for an Autonomous Data Warehouse named "DB 201901192227". The main interface includes tabs for "DB Connection", "Service Console", "Scale Up/Down", "Stop", and "Actions". The "Autonomous Data Warehouse Information" tab is selected, displaying the following details:

Information	Value
Display Name:	DB 201901192227
Database Name:	DB201901192227
Database Version:	18.4.0.0
CPU Core Count:	2
Storage (TB):	1
Created:	Sun, 20 Jan 2019 03:28:42 GMT
Compartment:	arup (root)
OCID:	...bph5dq Show Copy
License Type:	Bring Your Own License
Lifecycle State:	Available

it listed on the same screen (**Figure 4**). Now click the name (**DB 201901192227** in my case) to bring up a screen similar to the one shown in **Figure 5**.

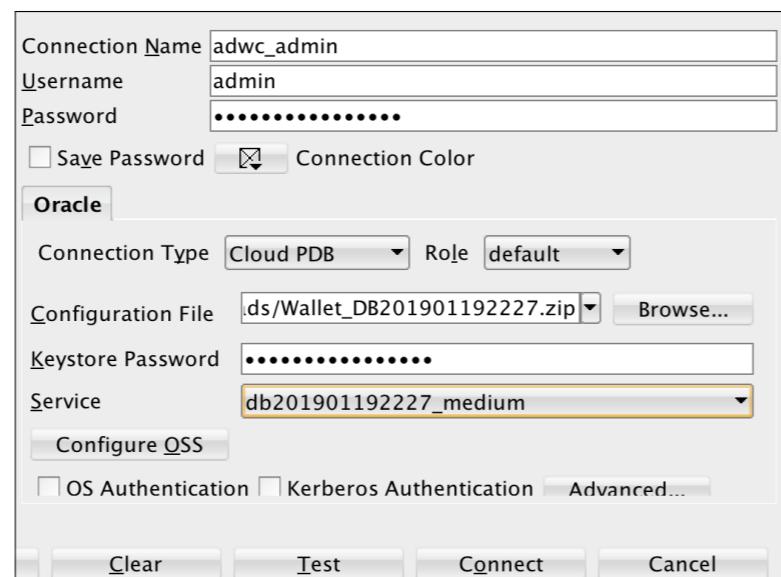
Click the **Start** button to start the data warehouse. When the data warehouse starts, the button changes to **Stop** (which is what is showing in **Figure 5**). After the data warehouse starts, the big ADW icon on the left turns green and shows **AVAILABLE**.

At this point, to connect to the data warehouse, click the **DB Connection** tab, which will bring up a window that includes the **Download Client Credentials** button.

Click that button to download the configuration file, which you will use to connect to the database from an Oracle client.

To connect from Oracle SQL Developer, for example, use the configuration shown in **Figure 6**.

Figure 6: Oracle SQL Developer connection



When you connect to the data warehouse from Oracle SQL Developer, you can see the included SH sample schema and all the tables in it. Test some simple statements, such as the following, to make sure your connection is valid:

```
select * from sh.sales;
```

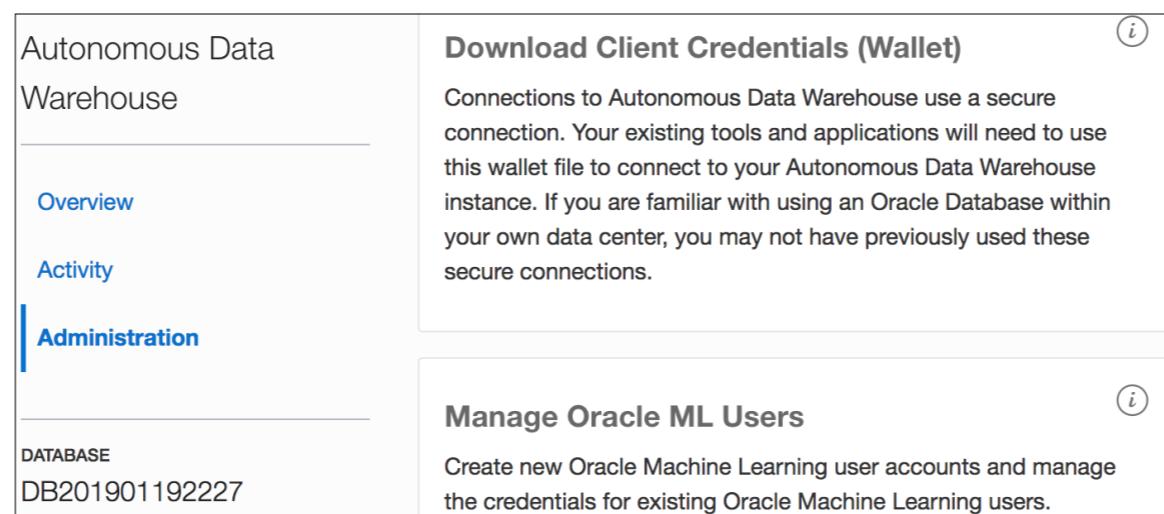
ORACLE MACHINE LEARNING NOTEBOOKS

Acme has already set up Oracle Autonomous Data Warehouse, so Jane moves on to the challenge at hand. Someone in the meeting brings up an interesting point. Most

data scientists are used to the concept of a *notebook*, which enables them to store SQL statements one after the other and store the results of the statements right afterward. (Examples of notebooks include [Jupyter](#) and [Apache Zeppelin](#)). Couldn't Acme use a notebook? Someone points out that moving the data from the cloud to a local system (where Oracle SQL Developer runs) may not be the best in terms of latency, especially when the data volume is high.

Jane explains that there is a simple solution. Oracle Autonomous Data Warehouse comes packaged with a web-based notebook server, based on Apache Zeppelin, which enables data scientists to collaborate by entering SQL in a common notebook and by sending their own notebooks to one another. To show how to use the built-in notebook server, she clicks the **Service Console** tab on the screen shown in [Figure 5](#). This opens the service console in a different window. Jane warns that if pop-ups are blocked in the browser, this will not work. The service console shows various stats on the service, such as CPU usage, the number of SQL statements, and the average SQL response time. On the left is a menu item called **Administration**, shown in [Figure 7](#).

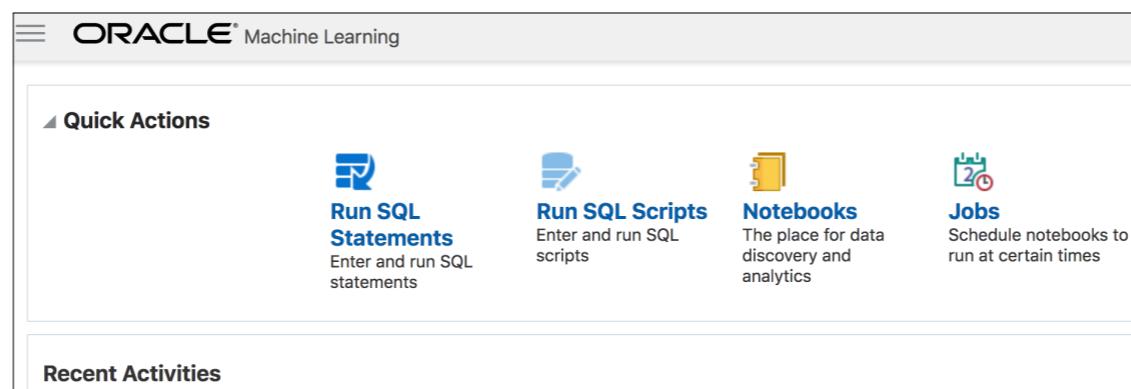
Figure 7: Autonomous Data Warehouse administration



She clicks that item and the **Manage Oracle ML Users** panel to the right of it.

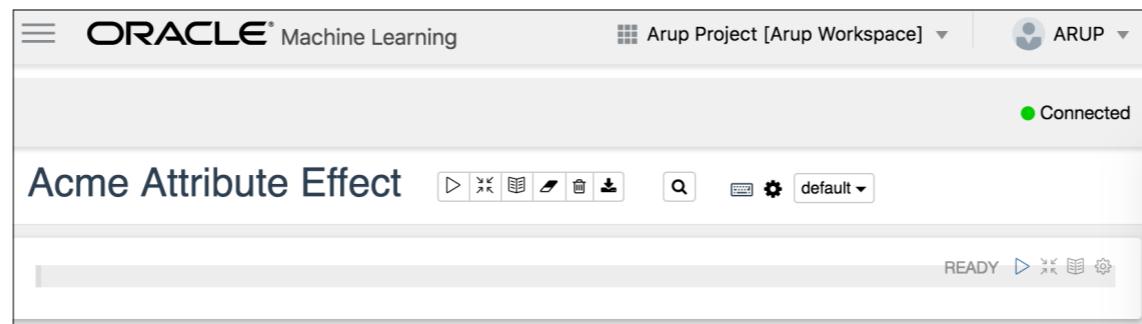
On the **Manage Oracle ML Users** screen, Jane sees all the users defined for Oracle Machine Learning. She creates a few users on this screen, including one called ARUP. Then she clicks the **Home** icon on that screen to enter Oracle Machine Learning, using the username and password she just created. This brings up a screen similar to the one shown in **Figure 8**.

Figure 8: Oracle Machine Learning home



On this screen, Jane can create notebooks and run them. She clicks the **Notebooks** icon, which brings up a screen showing all the notebooks created; clicks **Create** to create a new notebook; and names it Acme Attribute Effect. The notebook shows only a single box at this point, as shown in **Figure 9**.

In the blank entry, Jane can enter any SQL or PL/SQL code. Then she clicks the right-arrow icon, next to the word **READY**. The output of the statement will come right below the statement—that's how notebooks work. This enables a user to quickly test a statement and also record it so someone else can look at both the statement and the data.

Figure 9: New notebook

ATTRIBUTION ANALYSIS

Next Jane turns to the original question posed by the marketing department: how to better identify the customers who will buy more products, so it can advertise to them directly, offer discounts, and increase sales.

Here are the two tables (both in the SH schema):

SALES shows all the sales data for a customer, including two very important columns:

- AMOUNT_SOLD is the actual value of the goods sold.
- PRODUCT_ID is the unique product identifier.

SUPPLEMENTARY_DEMOGRAPHICS shows all demographic data on the customer, including the following columns:

- AFFINITY_CARD is the likelihood that a customer will respond to the loyalty program discount and spend more money. 1 means yes; 0 means no.
- EDUCATION is the highest education the customer has received.
- OCCUPATION is the customer's occupation.
- HOUSEHOLD_SIZE is how many other people live in the same house as the customer.
- YRS_RESIDENCE is how many years the customer has been staying in the current residence.

The important data point here is the AFFINITY_CARD value. And the question is which of the other demographic and sales data has the highest influence on the AFFINITY_CARD value. When that is known, Marketing can tailor its campaigns accordingly.

ATTRIBUTE RANKING

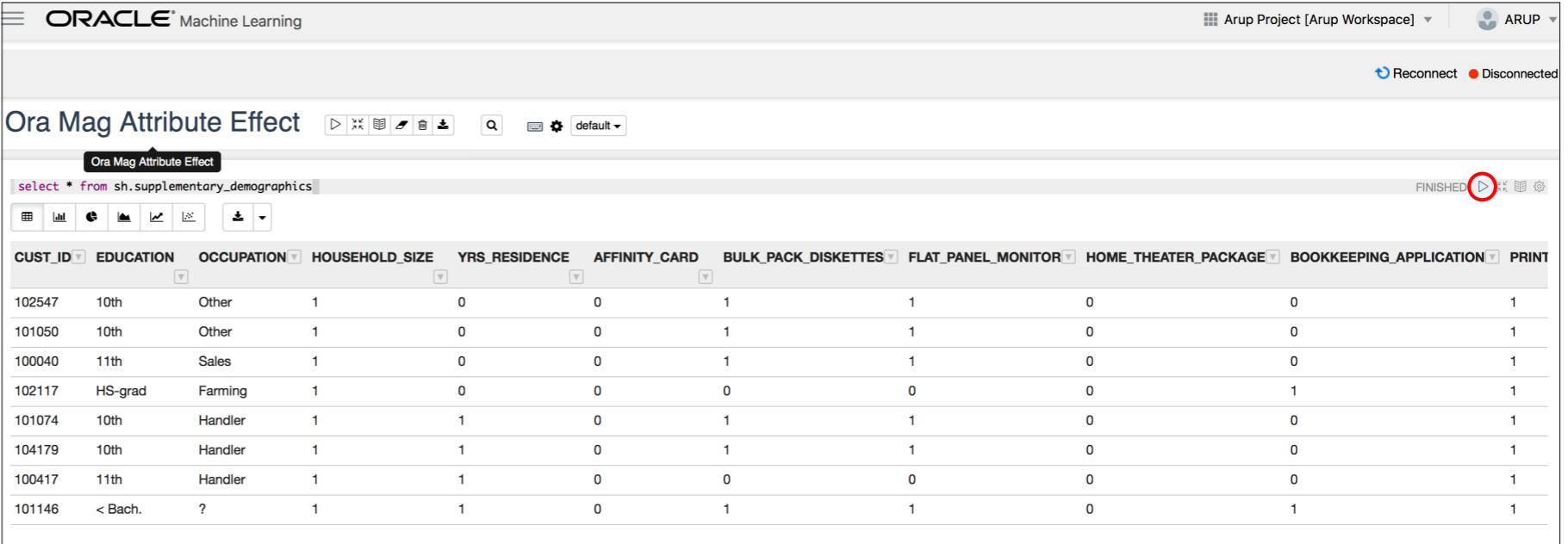
There's more than one way to solve most problems, Jane reminds everyone in the room. Oracle Autonomous Data Warehouse offers many utilities to perform machine learning on the data. Jane offers to show one of those utilities—Attribute Ranking Functionality—in which the machine learning model identifies the *strength* of any correlation between the various attributes and ranks them in order of *influence* on the dependent variable. In Oracle Autonomous Data Warehouse, the utility is available through functions of the DBMS_PREDICTIVE_ANALYTICS package. One such function is called EXPLAIN, which computes the rank of the attributes influencing the dependent variable, which is AFFINITY_CARD in this case.

Jane pulls up the notebook and enters the simple SQL statement

```
select * from sh.supplementary_demographics
```

to show all the records from SUPPLEMENTARY_DEMOGRAPHICS and then clicks the execution button (the right-arrow icon). **Figure 10** shows the SQL statement, the execution button (in the red circle), and the output.

This is how a notebook functions, Jane tells her audience. The SQL statement is entered in a field, which is called a paragraph. The output comes immediately afterward. The user can modify the paragraph but not the output. Another paragraph opens up just under the output, where the user can enter another command, which can be in SQL or PL/SQL.

Figure 10: Notebook first step


The screenshot shows a notebook interface for Oracle Machine Learning. The title bar says "ORACLE Machine Learning". The main area is titled "Ora Mag Attribute Effect". Below the title is a SQL query: "select * from sh.supplementary_demographics". The status bar at the bottom right says "FINISHED". A red circle highlights the "FINISHED" status bar.

CUST_ID	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE	AFFINITY_CARD	BULK_PACK_DISKETTES	FLAT_PANEL_MONITOR	HOME_THEATER_PACKAGE	BOOKKEEPING_APPLICATION	PRINT
102547	10th	Other	1	0	0	1	1	0	0	1
101050	10th	Other	1	0	0	1	1	0	0	1
100040	11th	Sales	1	0	0	1	1	0	0	1
102117	HS-grad	Farming	1	0	0	0	0	0	1	1
101074	10th	Handler	1	1	0	1	1	0	0	1
104179	10th	Handler	1	1	0	1	1	0	0	1
100417	11th	Handler	1	1	0	0	0	0	0	1
101146	< Bach.	?	1	1	0	1	1	0	1	1

To facilitate her analysis and make it simpler for the audience to understand, Jane decides to combine the multiple data elements into a single table, via the following SQL script:

```
create table demo_sales as
select d.cust_id, s.prod_id, s.amount_sold, d.affinity_card, d.education, d.occupation,
d.household_size, d.yrs_residence
from sh.supplementary_demographics d, sh.sales s
where s.cust_id = d.cust_id;
```

The best part of using the notebook is that she can enter this SQL right there in the notebook, in one of the paragraphs. She creates the DEMO_SALES table and selects the data from it in the next step, as shown in [Figure 11](#).

Figure 11: Creation of the DEMO_SALES table

The screenshot shows two code cells and their results in a Jupyter Notebook interface.

```
create table demo_sales as
select d.cust_id, s.prod_id, s.amount_sold, d.affinity_Card, d.education, d.occupation, d.household_Size, d.yrs_residence
from sh.supplementary_demographics d, sh.sales s
where s.cust_id = d.cust_id
Updated 2804 row(s).
```

FINISHED ▶ ✎ 📄 ⚙


```
select * from demo_sales
```

FINISHED ▶ ✎ 📄 ⚙

CUST_ID	PROD_ID	AMOUNT SOLD	AFFINITY_CARD	EDUCATION	OCCUPATION	HOUSEHOLD_SIZE	YRS_RESIDENCE
100251	26	149.99	0	9th	Crafts	2	2
100251	48	11.99	0	9th	Crafts	2	2
100199	27	44.99	0	9th	House-s	2	2

Then Jane performs the most important part of the exercise: ranking the influence of the attributes (all the columns of the DEMO_SALES table) on the AFFINITY_CARD column. She invokes the EXPLAIN function, as described above, by entering the following in the next paragraph:

```
%script
begin
    dbms_predictive_analytics.explain(
        data_table_name      => 'DEMO_SALES',
        explain_column_name => 'AFFINITY_CARD',
        result_table_name   => 'OraMag_Explain_Output');
end;
/
```

The %script reference in the first line tells the notebook that it's a PL/SQL script and not a SQL statement. The notebook executes the function and stores the results

in a table called OraMag_Explain_Output, which is created by this execution. The final act is simply to select from this table to find out the rankings, which Jane enters in the next paragraph.

```
select attribute_name, rank, to_char(explanatory_value,'9.9999') explanatory_value
from OraMag_Explain_Output
order by rank, explanatory_value desc
```

The output in the notebook is shown in **Figure 12**. Jane explains the output to her audience. The RANK column shows the ranking of the influence of that attribute on the AFFINITY_CARD column. So the output says that the HOUSEHOLD_SIZE column is the highest-ranking in influence on the AFFINITY_CARD column. The next-highest ranking one is YRS_RESIDENCE, and so on.

To make the chart more effective for presentation, Jane clicks the bar chart icon (shown in a red circle in **Figure 13**), just above the output. It shows the same data, but as a bar chart, with each bar's height representing the EXPLANATORY_VALUE

Figure 12: Rankings

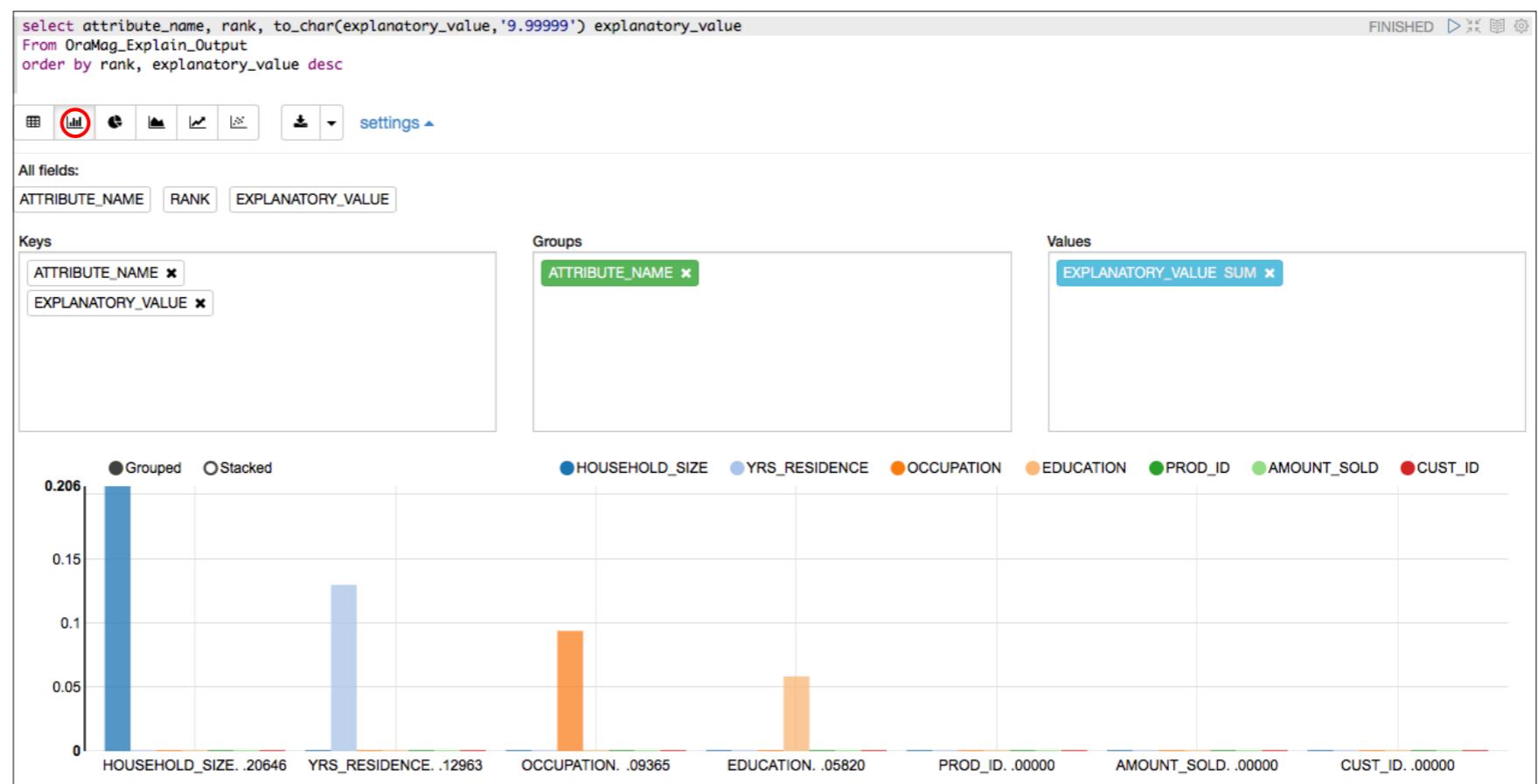
ATTRIBUTE_NAME	ATTRIBUTE_SUBNAME	EXPLANATORY_VALUE	RANK
HOUSEHOLD_SIZE		2.06461212561194818474077847391705896411E-01	1
YRS_RESIDENCE		1.29625998285064010582744182738452923476E-01	2
OCCUPATION		9.36464396837933835507199484835346081107E-02	3
EDUCATION		5.81953614310728412867375910955570967898E-02	4
PROD_ID		0	5
AMOUNT SOLD		0	5
CUST_ID		0	5

column, ordered from the highest to the lowest ranks, as shown in **Figure 13**. A picture is worth a thousand words, and this one is no exception.

The EXPLANATORY_VALUE column is very important, Jane explains. The value is a number between 0 and 1, and a value of 0 means that the columns are not correlated at all. Pointing to the output, Jane explains that the PROD_ID column is *not correlated* and should be discarded from the analysis. A value of 1 means that the columns are perfectly correlated, but that is a very rare occurrence.

The marketing folks got their answer—they need to focus on household size and years in a residence, followed by the other attributes and *not* focus on product IDs

Figure 13: Bar chart



or individual customers.

Although everyone is impressed with the initial findings, all agree that this is not the end to their quest to improve sales. A typical problem involving machine learning has to go through multiple models and iterations to find the answers. Jane explains that her demonstration was meant to showcase how easy it was to find a key answer with the built-in machine learning functionality in Oracle Autonomous Data Warehouse, however.

ADVANTAGES OF NOTEBOOKS

Someone in the room points out that all of the work was done in SQL and PL/SQL and could have been done with a simple client application such as Oracle SQL Developer. What was the point of doing it in a Notebook?

Jane concedes that this is a fair question and goes on to explain why a notebook approach may be better:

1. Large swaths of data need not be transferred to the user's client application from the cloud. Notebooks are on a notebook server in the cloud.
2. The notebooks store the statements and the results together and can be examined without re-execution of the statements. Jane clarifies that she can re-execute the statements anytime to get the most-recent data.
3. The notebooks can be shared with other team members, and multiple people can collaborate to produce a single notebook.
4. Notebooks can be turned into jobs, which can execute all the statements in the notebook in the same sequence automatically and produce the results as the last step. This is important when the data pattern changes. A job will automatically execute all the steps and get the final result.

CONCLUSION

The most important point, Jane reminds everyone, is that all these analyses were done right inside the database, with no data transfers to any outside systems. Second, the analyses were done with Oracle Database-resident packages with plain SQL—no R or Python expertise was required. And this is merely scratching the surface, she adds. Oracle Autonomous Data Warehouse provides plenty more utilities and tools for machine learning and data science, and she lets the audience know that she will send everyone there a link for more information on [Oracle Machine Learning](#). The audience leaves the meeting impressed and happy. □



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



ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about Oracle Autonomous Data Warehouse.
Oracle Machine Learning.

EXPLORE examples on GitHub.
TRY Oracle Autonomous Data Warehouse.

**ORACLE DATABASE**

A Higher-Level Perspective on SQL Tuning, Part 2

Identifying and diagnosing poorly performing SQL

In [my previous article on SQL tuning](#), I discussed the importance of stepping back from focusing on a purely technical approach to tuning SQL. Before tuning SQL statements, the practitioners charged with the task must first understand the business requirement of the application components that execute that SQL, so that they can validate the correctness of the SQL statements to meet that requirement.

Even when the SQL's correctness is validated, it is still important to have the right focus when tuning an application. As I mentioned in the previous article, no business has a "Let's make sure SQL runs fast" model; they have core functions that drive revenue. It just so happens that these core functions are often backed by IT applications that execute SQL statements.

Hence the processing cost of a SQL statement is not linked to business success *unless* that cost has an impact on the delivery of those core business functions, and

the *real* task of a database developer is to tune the delivery of the core functions. For example, if a daily report takes two hours to execute and a consequence is that critical information is not available to management for making strategic decisions about business operations, then a “tuning” exercise might be simply to automate the running of the report before standard working hours, so that the report output is available when managers come into the office. Although it is seductive to focus solely on tuning code—after all, most developers chose their IT careers to become coders—the *delivery of business functionality* must be the driving force for tuning efforts. It can also be much more rewarding for you as a developer to think this way, because when you focus on ensuring that users of the application are successful in performing their business functions, the same users will sing your praises when you improve their experience with the applications you build.

TUNING THE USER EXPERIENCE

The best way to identify the pain points for users of your application is also the simplest. *Communicate directly with the users of your application.* Nothing impresses business users more than taking the time to sit alongside them to get a firm understanding of where tuning efforts are best focused, and often this can even reduce the amount of effort required. For example, I had a client who detected huge response times on a key screen of its application, and I fully expected venting of anger about this performance when I visited. However, it turned out that the organization’s business workflow meant that the slow performance didn’t have any impact at all—the staff just launched the application each morning at the beginning of the departmental standup meeting. In fact, the most important enhancement request was simply to change the text on one of the buttons to make its function less ambiguous for the junior staff. A five-minute fix, and they were wowed by the

quick turnaround time and the “personal touch” of having an IT person liaise with them directly. The other benefit of direct contact with your users is that even if a performance problem cannot be solved immediately, or perhaps ever, they normally appreciate that you are at least reaching out to them and are aware of the things that matter *to them*.

FINDING PROBLEMATIC SQL

If it is possible to have direct interaction with application users and have them demonstrate a performance issue on a particular business function, the task becomes identifying whether the issue is being caused by slow SQL execution and, if so, what those poorly performing SQL statements are. The definitive tool for performance capture and analysis is Oracle’s SQL trace capability. Tracing is enabled just before execution of the business function and disabled immediately afterward, thus capturing only the time frame and the SQL executions of interest. A detailed look at tracing is beyond the scope of this article, but Arup Nanda covers an introduction to tracing in [*this Oracle Magazine article*](#).

As powerful as it is, there are some challenges with tracing:

- In the modern world of stateless applications running in connection pools, tracing a single activity within an application may involve tracing multiple database sessions, each of which is also performing tasks for other application requests concurrently. Extracting the SQL statements pertaining to just the application activity of interest requires extensive instrumentation throughout the application.
- The performance issue may not be easy to reproduce. For example, if a month-end task runs slowly, it may be another month before a trace can be performed to capture the issue again. Or it may be a performance issue that arises intermittently and thus cannot be reproduced on demand.

- Sometimes an impatient user will not be pleased about being asked to repeat (potentially multiple times) an activity that is causing frustration.
If tracing cannot be successfully utilized, there are still other options available for homing in on the SQL statements that are directly impacting business functions.
Automatic Workload Repository. The Automatic Workload Repository feature of Oracle Database collects and analyzes performance statistics in the database to assist with problematic SQL detection. The gathered data is stored both in memory and in the database, and the data includes SQL execution and performance statistics. An Automatic Workload Repository snapshot is a capture of the state of a database at a point in time, so when you're analyzing a performance issue encountered by a user, a delta of the snapshots before and after the time when the performance issue occurred can be used to discover problematic SQL statements that most probably were related to the issue. Much as with the trace methodology, the ideal scenario is to take snapshots immediately before and after replicating a performance problem.
Active Session History. The Active Session History feature of Oracle Database samples active database sessions each second, writing the data to memory and persistent storage. An active session is a session that is using CPU and is not sitting idle waiting for a request from an application and/or user. Slow-running SQL statements can also be detected by Active Session History, because a session running a SQL statement for, say, 10 seconds will have 10 consecutive entries in the V\$ACTIVE_SESSION_HISTORY view for the same SQL_ID/SQL_EXEC_ID columns pairing.
Automatic Workload Repository and Active Session History are powerful tools, because they allow for after-the-fact problem analysis. A business user may report a performance problem that occurred a few hours beforehand. Even if that problem cannot be replicated, the Automatic Workload Repository and Active Session History

data can often be used to detect the problematic SQL statements for that user.

However, both Active Session History and Automatic Workload Repository are additional license options of Oracle Database, Enterprise Edition, so they may not be available for your performance tuning analysis. If that's the case, you can use the dynamic performance views to find poorly performing SQL statements in a database. Marrying this information with a business user's performance issue is more difficult, but with a little luck, the most resource-intensive SQL statements will have some identifying characteristics that allow a basic level of confidence in mapping them to business functions. The V\$SQL performance view lists SQL statements in the shared SQL area and presents numerous performance measurements alongside each, as shown in [Listing 1](#).

Here are the most commonly used V\$SQL columns for providing performance measurement on SQL statements:

- DISK_READS discovers SQL statements performing large amounts of physical I/O.
- BUFFER_GETS is a measure of logical I/O, which is the most common cause of high CPU costs for SQL statements.
- EXECUTIONS determines whether a SQL statement is being run an excessive number of times.
- ROWS_PROCESSED can be useful for INSERT, UPDATE, and DELETE statements to determine large redo costs.
- ELAPSED_TIME is the cumulative elapsed time across all executions for a SQL statement, which maps most closely to user response time.

Those columns provide the performance characteristics of each SQL statement. They can be combined with other V\$SQL columns to tie SQL statements back to their root business functions:

- PARSING_USER_ID narrows the SQL statement down to the schema(s) your applica-

tion is running under. This equates to the `USER_ID` column in the `DBA_USERS` dictionary view.

- `SERVICE` is the database service your application is connecting to. Sadly, services are a dramatically underutilized feature of Oracle Database, so many applications simply use the default service name equal to the database name. See the [documentation for details on how to take advantage of database services](#).
- The `MODULE` and `ACTION` columns enable you to track back the level of granularity provided by the application developer in SQL statements—if your application is instrumented to use the `MODULE` and `ACTION` column values via the `DBMS_APPLICATION_INFO` package or via the end-to-end metrics capabilities in the various Java, Python, and other drivers.
- The `PROGRAM_ID` and `PROGRAM_LINE#` columns will include the `OBJECT_ID` function and the program line number of the PL/SQL object in the database—if the SQL statement originated from a PL/SQL stored program unit. The sidebar “[A Higher-Level Perspective on SQL Tuning, Part 2, PL/SQL](#)” includes an example that uses these `V$SQL` columns to track SQL. (Refer to the [SmartDB Resource Center](#) for details on the benefits of controlling all database access via a secure PL/SQL shell.)

Although `V$SQL` is a very powerful view for discovering information about SQL statements currently in the shared SQL area, there is a risk to using it. Querying `V$SQL` occasionally for a particular performance issue that occurs rarely is fine, but if you are diagnosing a system that is under severe duress, then hammering away at `V$SQL` can create problems of its own. A system under duress is quite possibly experiencing extreme contention in its shared memory structures, and by aggressively querying `V$SQL`, you may be adding to that workload and/or contention. This appears to be a catch-22 situation: How do you diagnose a struggling system if diag-

nosing it via queries to V\$SQL will make it struggle more? For such cases, there is an alternative performance view, V\$SQLSTATS, that contains almost as much information as V\$SQL. (Below is a summary, but see the documentation for a more detailed [description of V\\$SQLSTATS](#).)

V\$SQLSTATS displays basic performance statistics for SQL cursors and contains one row per SQL statement (that is, one row per unique value of SQL_ID). The column definitions for columns in V\$SQLSTATS are identical to those in the V\$SQL and V\$SQLAREA views. However, the V\$SQLSTATS view differs from V\$SQL and V\$SQLAREA in that it is faster, more scalable, and has a greater data retention (the statistics may still appear in this view, even after the cursor has been aged out of the shared pool). Note that V\$SQLSTATS contains a subset of columns that appear in V\$SQL and V\$SQLAREA.

My general approach is to find poorly performing SQL statements initially by querying V\$SQLSTATS and then if I need to determine more information about individual SQL statements such as the MODULE or ACTION column values, I will access V\$SQL, using the SQL_ID of just those poorly performing SQL statements. Using V\$SQLSTATS isn't an invitation to smash away continuously at the database, but by avoiding V\$SQL, you are steering clear of a critical shared-memory structure that many database sessions are typically competing for. V\$SQLSTATS has been available since Oracle Database 10g.

THE REAL SQL

To recap, the developer should always focus on the specific business functions for the user and then use the trace methodology and tools such as Automatic Workload Repository, Active Session History, and the performance views to identify the SQL

statements that are the cause of poor response times for those business functions.

Now that the SQL statements have been identified, I recommend performing one extra step before diving into tuning techniques, and that is to expand the SQL text. Expanding a SQL statement reveals what the database will truly be running when it is presented with the text of that SQL statement. For example, if I have a simple query

```
SQL> select * from EMP_LIST;
```

it is only through my prior knowledge that I created a synonym via

```
SQL> create synonym EMP_LIST as SCOTT.EMP;
```

that I know that I am actually querying the SCOTT.EMP table. Rather than taking each referenced object in the SQL statement and digging into the data dictionary to see if that object is a synonym or a view and recursively continuing that process until all objects are resolved, I can use the supplied DBMS.Utility database package to expand a SQL statement to reveal what the database will ultimately “see.” **Listing 2** uses the above query as an example.

Listing 2: Expanding the SQL text of a statement

```
SQL> variable c clob
SQL> begin
 2   dbms_utility.expand_sql_text
 3   ( 'select * from EMP_LIST',:c);
```

```
4  end;
5  /
```

PL/SQL procedure successfully completed.

```
SQL> print c
```

```
c
```

```
SELECT
```

```
  "A1"."EMPNO" "EMPNO", "A1"."ENAME" "ENAME", "A1"."JOB" "JOB",
  "A1"."MGR" "MGR", "A1"."HIREDATE" "HIREDATE", "A1"."SAL" "SAL",
  "A1"."COMM" "COMM", "A1"."DEPTNO" "DEPTNO"
FROM "SCOTT"."EMP" "A1"
```

Expanding the SQL text may seem like a waste of time and effort, but here is another example of why looks can be deceiving when you're faced with a SQL statement for tuning. **Listing 3** shows a view that joins two database dictionary objects, appropriately named LOOKS_SO_INNOCENT.

Listing 3: View definition for LOOKS_SO_INNOCENT

```
SQL> create or replace
  2  view LOOKS_SO_INNOCENT as
  3  select
  4    o.owner,
```

```
5    o.created,  
6    s.bytes,  
7    s.tablespace_name  
8  from  
9    dba_segments s,  
10   all_objects o  
11 where o.owner = s.owner  
12 and   o.object_name = s.segment_name;
```

View created.

When I expanded the `SELECT * FROM LOOKS_SO_INNOCENT` query via the `DBMS_UTILITY` package, the resulting formatted SQL statement was more than 1,000 lines, or approximately 25 pages of standard text on a printed page (and available in [Listing 4](#)). It can be puzzling to see a single-line query performing poorly—but not so much when that query is expanded to 1,000 lines.

SUMMARY

In the [previous article in this series](#), I explained the importance of validating that a SQL statement is correct in terms of its construction and is solving the intended business requirement *before* diving into the various tuning techniques at your disposal.

Now continuing in the vein of taking a holistic approach, I've explained the importance of focusing on tuning from the perspective of the business user rather than the SQL statements, because the worst-performing SQL in the database is not nec-

essarily related to the business function that is causing the most frustration for a user. Remember that the goal is to tune the user experience, and that is what drives identification of the SQL statements to be tuned.

In the next article, I'll explore how I proceed after a set of poorly performing SQL statements has been identified. □



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



ILLUSTRATION BY **WES ROWELL**

NEXT STEPS

LEARN more about SQL tuning.

TRY Oracle Autonomous Transaction Processing.

DOWNLOAD Oracle Database 18c.



Inspiration, Connection, and Community

UKOUG leaders pay it forward with full range of tech topics and support.

BY LESLIE STEERE

Brendan Tierney has what may be the most exhaustive personal library of *Oracle Magazine* issues, spanning from 1987 through the final print edition, in 2015, and he's devoted to completing the collection, which he began in 1992 when he first started working with Oracle technology. That's just one example of the commitment Tierney, a member advocate and board member of the UK Oracle User Group (UKOUG), has to Oracle, Oracle technologies, and the needs and insights of Oracle users. And it mirrors the ethos of UKOUG as an organization founded in 1983 as "a hub for connecting, learning, sharing, and shaping conversation for the benefit of the community."



A commitment to its diverse membership and a close relationship with Oracle keep the UK Oracle User Group growing and vibrant, according to member advocates Brendan Tierney (pictured) and Neil Chandler.

That user group ethos might have been simpler to follow 36 years ago, when Oracle was releasing version 3 of its single product—a relational database. Today, with Oracle providing a complete stack of cloud and on-premises technology, the challenge is a bit more expansive. “UKOUG now is a broad-based user group that covers many different sectors, from our apps community to our JD Edwards community [a separate user group that merged with UKOUG when Oracle acquired PeopleSoft, which had acquired JD

Edwards] and also quite a large tech community,” says Tierney, who is also an Oracle Groundbreaker Ambassador and Oracle ACE Director.

Across those communities, the user group supports several horizontal and vertical communities. “We have people who are very vertically focused on perhaps a database technology,” says UKOUG Member Advocate Chair and fellow Oracle ACE Director Neil Chandler. “And then we’ve got the horizontal folks as well—like our higher education or public sector community—who want to talk about how they are using the cloud.”

UKOUG’s business applications side has more of a cloud focus, because of SaaS, Chandler adds, “whereas the tech side is much more diverse: It’s got a huge on-premises focus. So we kind of package our members into three areas. We have what we call our traditionalists; and then we have the apps folks who are already fully in the cloud; and then we’ve got our transitioning members,

ON EMERGING TECHNOLOGIES

“I like the machine learning within Oracle Database,” says Brendan Tierney, a member advocate and board member of UKOUG. “I think it’s a nice product, because the data is in the database. There’s no data movement—we can do the machine learning right on the spot. We can expose the machine learning by using some very simple commands through SQL, or we can REST-enable it for HTTP access.”

who are in both camps—they may be moving totally to the cloud, or they may be moving to a hybrid solution.”

NO MEMBER LEFT BEHIND

With the Oracle Cloud push over the last couple of years, “our traditionalists feel a little ignored by Oracle,” says Chandler. “But I think Oracle is coming forward

to recognize those customers and that the complexity of the systems they’re running might make them difficult to transition to the cloud and that it may take quite a long time.” These customers may not be ready to make the move at this point, Chandler adds, “but they still need looking after. And Oracle in the UK is coming to us to see if we can help with the husbandry of those people. It’s some-

thing we’ve started to see this year that is really positive.”

The charter for UKOUG, Tierney emphasizes, is to help its members, regardless of what stage they are in, by providing the support and education that help them work out what their next step should be. “We have a range of different types of events, from a big conference with multiple tracks that brings everybody together, to smaller free events such as regular meetups, to webinars,” he says, and UKOUG also is working toward having more online content with learning tracks that help members navigate through different stages of their career.

UKOUG member advocates meet with Oracle liaisons on a monthly or bimonthly basis to discuss the different technology trends and types of events planned, Tierney adds, “and particularly in more recent years, we discuss how we can work together more closely and help target the various kinds of users and give them the right content.”

ON MOVING TO THE CLOUD

“It’s about supporting our members where they are—understanding that not everyone, especially large corporations with incredibly complex systems, can just go to the cloud,” says UKOUG Member Advocate Chair Neil Chandler. “It’s very easy for startups. I mean, no startup in its right mind would do anything but go to the cloud. But for a large corporation with systems that are 10, 20, 30, or 40 years old, transitioning to the cloud is a substantially greater challenge.”

"The JD Edwards community likes to all get into a room and talk," Chandler explains. "They don't want to see too many projectors. The technology community—they want projectors, and they want to be educated by experts and informed about real-world practices by their peers. The apps side is a more fragmented community and has some narrow industry-specific verticals. UKOUG will continue to provide special-interest days and meetups for those verticals, as well as broader conferences that span all Oracle offerings."

In addition to a growing roster of events and webinars, UKOUG publishes

a magazine three or four times a year that includes content from its various JD Edwards, business applications, and technology communities. "We get members very, very eager to contribute to #PTK [Pass the Knowledge, formally *Oracle Scene*]," says Tierney, "because people still like having their name in print and being able to share their knowledge—and like showing their mother they are published as an author. And we also get experts from around the world who are very willing to contribute." □

Leslie Steere is editor at large for Oracle Content Central.

PHOTOGRAPHY BY
ORACLE DIGITAL MEDIA PRODUCTION

NEXT STEPS

EXPLORE UKOUG's upcoming events.

CHECK OUT #PTK (Pass the Knowledge).

SEE Brendan Tierney's *Oracle Magazine* collection.

WATCH Brendan Tierney and Neil Chandler on video.