EOUC

Oracle 23c

Oracle 23c: What to expect

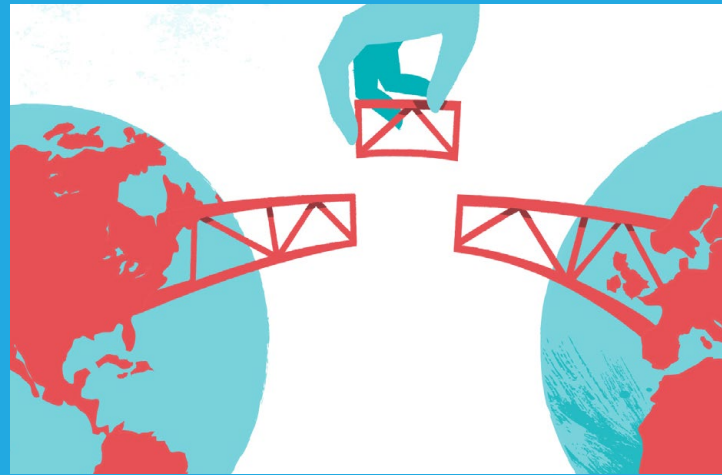**Bridging Worlds:** Unleashing the power of JSON

Observability and Databases: Oracle DBAs have an advantage

APEX 23.2: Workflows, Working Copies and much more
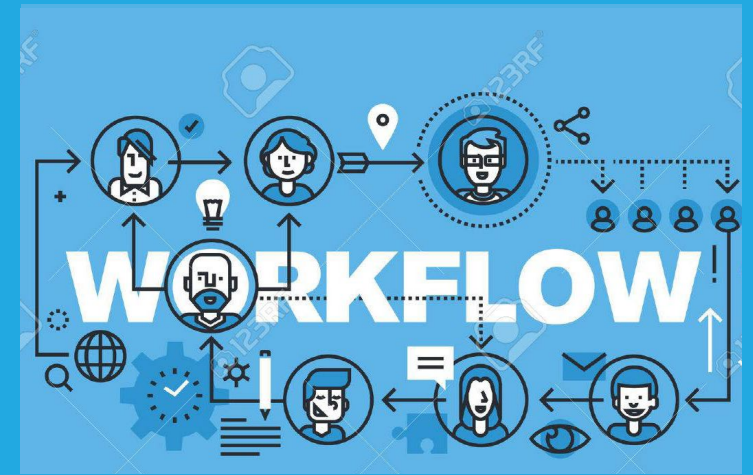
Observability and Databases: Oracle DBAs have an advantage

page 11

Bridging Worlds: Unleashing the power of JSON

page 15

APEX 23.2: Workflows, Working Copies and much more

page 36

Have a comment? Suggestion? Want to submit an article proposal? **Here's how to do it.**

Editorial

Dear ORAWORLD Readers,

The Oracle 23c database is just around the corner. Accordingly, many articles deal with this database version, which represents the next long-term release and is also the first Oracle database version that only contains the multitenant architecture. This will require some migration effort in the coming period.

But Oracle APEX and the Oracle Cloud also play a role in this issue. For example, Jim Czuprynski and Carsten Czarski present interesting aspects of Oracle APEX in their respective articles.

To ensure that the ORAWORLD magazine can continue to appear regularly, we are looking for editors to help organize the articles for our magazine. If you have any questions and / or are interested, please contact me.

As always, we welcome your feedback and contributions to our magazine. If you have a story to share or an idea for an article, please don't hesitate to get in touch.

From my point of view this issue has an interesting mix of different articles.

I hope you enjoy this issue of our magazine.

Yours,

Christian Trieb
EOUC Board Member, DOAG Board Member,
Oracle ACE Direktor, Paragon Data GmbH

Submit Your Article!

You have an interesting topic to publish in ORAWORLD Magazine? Then submit your article and be part of it!

Please e-mail us your article via the online form at **www.oraworld.org**.

User Group Stories

Tech News

Nerd Stuff

Work & Life

Deadline for the next edition:
June 1, 2024
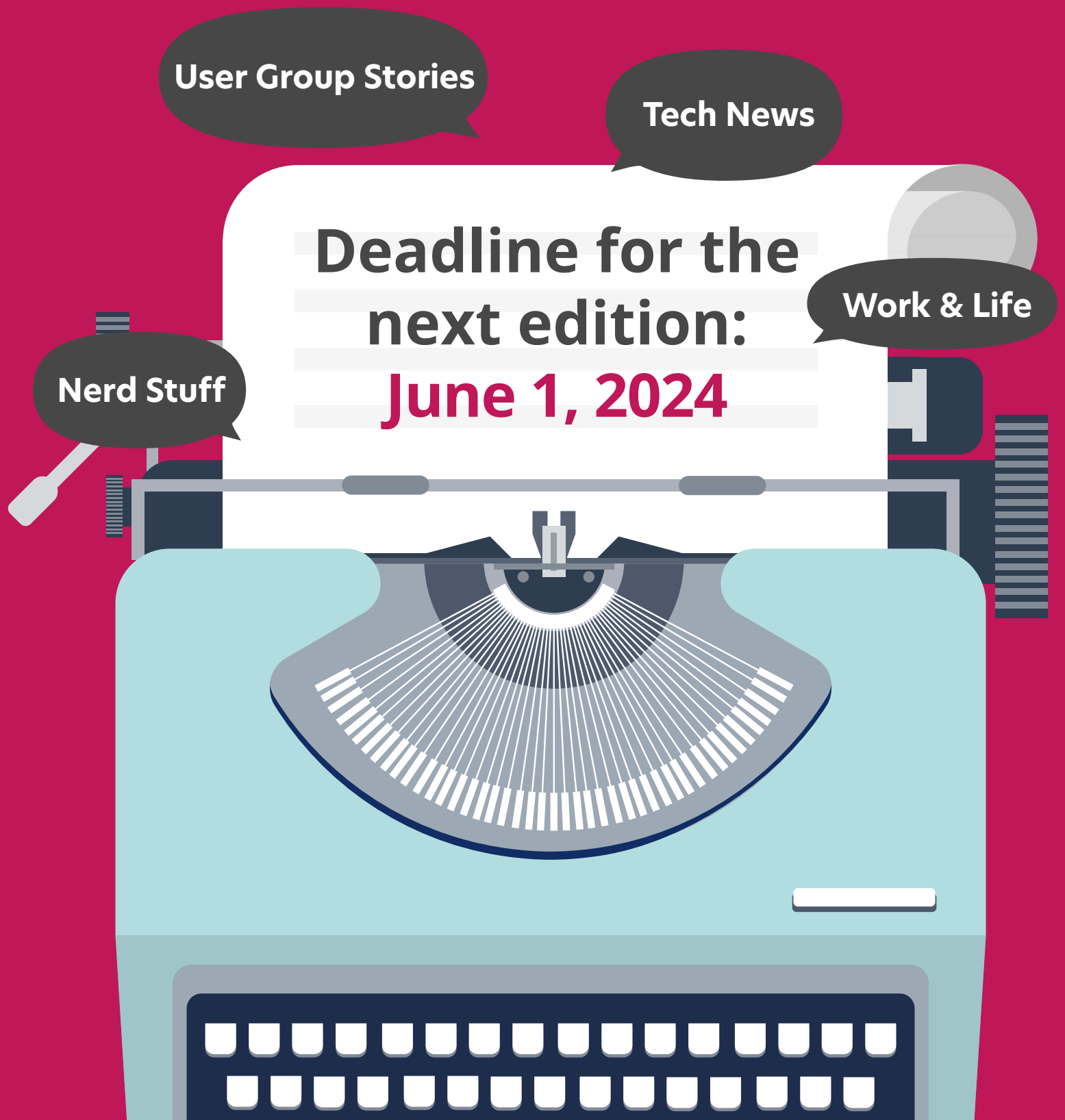
"The right tool for the right job at the right time?"

www.commitstrip.com

Oracle Database 23c: What to Expect

Christian Pfundtner

23c

During Oracle CloudWorld 2023, Oracle announced the general availability of Oracle Database 23c. Currently it is only available in the Oracle Cloud as OCI Oracle Base Database Service. The exact date for the release of the on-premises version is not yet communicated but expected within the next few months, what do we have to expect?

Starting with Oracle 12c the company follows a rather confusing way of releasing new versions. In the older times there would be a major version in 2 minor releases like 9.1 and 9.2, 10.1 and 10.2 and so on, also only those versions brought elementary new features. There would be 10.2.0.1 and 10.2.0.2 up to 10.2.0.4 but new features would be only available in 10.1.0.1 and 10.2.0.1.

12c suddenly saw the birth of InMemory database in 12.1.0.2 and after 12.2 Oracle decided to have new yearly versions, numbered by the year. We all know how well that worked for another big IT company in the 90s.

So, the current situation is: There are major versions like the current 19c and the new 23c that are for production with long time support. But there are also non-production versions like 21c that only exist for testing purposes.

Since April 2023 there is an Oracle Database 23c Free Developer Edition, but it is strongly limited: 2 CPU cores, 2GB RAM, 12GB data. Also, some of the enterprise Edition Features like Partitioning are missing. It is intended for developers who want to get acquainted with some of the new features. There is no support neither there are patches – so if you encounter a bug, you are on your own.

Though availability of 23c full version now has been announced on the Oracle Cloud, it is still limited: Only Base Database Services are available, even some cloud options like Autonomous Database und Cloud@Customer are not yet available (as of February 2024). Oracle promised them to arrive within the next months. Same applies to the on-premises database (that version you may install on your own machines). Usually that one will arrive roughly 6 months after the Cloud version, so we expect it to be available in a few months.

Let's look at the new features.

AI and Machine Learning

Everybody talks about AI and since ChatGPT there seems to be no limit. The situation is a little similar to the mid-90s when fuzzy logic suddenly appeared in every electric device. Yes, Oracle too has a bit called FUZZY in the View V$DATAFILE_HEADER, and guess what: It has nothing to do with fuzzy logic. It is even defunct since Oracle 10g.

This time they did much better: Data Mining with Machine Learning (ML) has been in the database for a longer time than the AI hype is around. Data scientists, developers and analysts, searching for information in big *Datalakes*, may make usage of ML. So, in the meantime it is possible to find the best ML algorithm including its parameters automatically, but unfortunately that's only accessible in the cloud. Oracle 23c brings a lot of improvements in that area, but the discrepancy between on-premises and cloud will pertain.

One of the improvements are XGBoost and Generalized Linear Model (GLM) algorithm to obtain more exact predictions. There is even the possibility to implement a chat function by using Large Language Models (LLMs). Oracle talks about implementing an interface where you may formulate your query in direct speech – maybe we finally reach the original goal of SQL: To have a tool that uses simple, natural language. We all know how well that went. But this time, maybe …

There are already demos you can watch, and it looks at least quite interesting. So, in the future, a data analyst just tells the database what he/she needs, and the database comes up with a query. I would call my database HAL then.

Back to earth: There is a completely new functionality in 23c: AI vectors. They represent content of unstructured data like documents, images, videos and so on. A vector is an array of numbers (dimensions) representing the critical characteristics of that data. This kind of vectors is generated and used by neuronal networks and represents the semantic content – without using the basic information. These kinds of vectors could already be stored and queried, but only 23c brings full integration into the database. There is now a special datatype to store that vectors, along with new SQL functions. Since Oracle emphasizes the database to be a Converged Database (meaning there are several ways to store and access things), these vectors may be handled by the means of different data models like JSON, Spatial, and so on. An example for usage of AI vectors was given by Oracle Vice President Juan Loaiza in his keynote during CloudWorld this year. He presented the search for a new home. A customer using the app from Realtor taking a picture of a house appealing to him/her. Entering additional parameter like area, number of rooms, price and so on triggers creation of

AI vectors describing the input. The AI in the database will then create appropriate suggestions. The search may be continued by entering additional pictures, parameter etc.

SQL Features

Oracle 23c brings some interesting SQL novelties and improvements. Some of them have been on the wish list of developers and DBAs for years. We will concentrate on the most important ones, as there are many of them. Some of them have been already introduced in newer releases of 19c, like blockchain tables or immutable tables.

One very specific new feature, that nobody understood why it was missing from the database for a long time: The datatype BOOLEAN. Oracle is really the last of the big database suppliers to implement it, others have since decades. Falling into the same category is – finally! – that the FROM clause is now optional. In your face, DUAL!

SQL Domains make it possible to define properties of tables and columns. They might contain even rules for sorting, exact content and representing that data. An example would be the rule that an email address must contain exact one "@" and when printing it out, some of its content may be anonymized (by using "*" instead of the real data).

A similar concept is represented by annotations: They store additional (meta-)information about tables and columns in a much more flexible way than mere comments could do. You might add pairs of (name, value) to any column, table or view. Thus, it is possible to create categories of objects that even may overlap. You might add e.g. the property (SECURITY, HIGH) to any column or table, and you may handle objects of that category differently than others. There is a plethora of possibilities.

One very painful issue (at least for DBAs) has always been the impossibility of granting privileges in a more common way, like "put all read privileges from that one role into another". Now, this is possible: Schema level privileges will ease the pain of administrators.

Another tiny but relevant improvement that Orace23c brings is the new role DB_DEVELOPER_ROLE, that consists of really all those privileges a developer might need. This is definitely an improvement, compared to the awful and incomplete

role RESOURCE. The new role will help with development, but unfortunately has shortcomings when it comes to performance tuning. Maybe a later patch will fix that.

And here is the next tiny feature that may ease your pain – at least when it come to writing deployment scripts: Any DDL Statement may now have the clause IF [NOT] EXISTS. That will eliminate a huge amount of unnecessary error messages: DROP TABLE xxx IF EXISTS or CREATE TABLE xxx … IF NOT EXISTS is really something long awaited. You may ask yourself: "Why did that take so long?" And you will get no good answer.

There is now a table value constructor: creating table data on the fly, without the need to use SELECT … **UNION ALL** SELECT … and so on. Instead, you may write it like this:

SELECT * FROM (VALUES (1,'SCOTT'), (2,'SMITH'), (3,'JOHN')) t1 (empid, fname);

Converged Database

This term means: Several types of data structures may not only co-exist but also interact. It started with Oracle 8i, when object relational data structures and Java took hold within the database. Also, the support for spatial data, multimedia and XML goes a long way back.

With the upcoming of JSON that started with Oracle 12c and really took off with 21c/23c, the database earned the attribute "converged". Oracle's commitment to JSON is much stronger than all the other stuff before (XML, spatial, …). Oracle database now allows to use all MongoDB tools and APIs towards 23c – they also claim that their database is faster in doing that than a MongoDB. In the same direction goes the inclusion of JavaScript. You may now write procedures/functions in JavaScript. That step seems to be necessary when talking about JSON integration. At last, you may run Kafka applications directly in the database (with minimal changes), because of High Performance Transaction Event Queues that also got support now.

Miscellaneous features

- Oracle did some work on error messages. So instead of "Computer says no" you'll get a more detailed message, telling you what object is concerned (some error messages in the older versions really omit even that information).

- SQL Firewall is a security feature going back to Oracle 11g, but as an external service. With Oracle 23c it became part of database, thus reducing latency.
- To reduce the probability of an important process to be blocked by a lesser one, Oracle 23c introduces priorities for transactions.
- Once again Oracle tackles the problem of rolling patches: Patching the system without downtime. It works now for single instance and cluster systems.
- Speaking about high availability: Data Guard, Real Application Cluster and und Global Distributed Databases (aka Database Sharding) experience many improvements. Some functionalities like Domain Cluster and automatic instance management policies came with 12c and are gone now. Some of them have already been deprecated in 19c.
- True Cache is a memory SQL cache and key/value cache to keep data as near as possible at the consumer (application server).
- Real time SQL Plan Management has been introduced (partially) in Oracle 19c, new is Auto SQL Repair: If performance of a statement degrades or an internal error occurs, an alternative plan is created to circumvent the problem. Oracle says, data will arrive with a short delay instead of not being delivered at all (in case of an internal error).
- Flashback Data Archives are now open for Blockchain Tables (introduced in 19c).
- Tablespaces may now shrink automatically.

Miscellaneous features for the Autonomous Database
- Materialized views may now be created automatically, if it improves performance. Currently (19c) only indexes get created automatically.
- Enhanced Real Time Object Statistics are now also available for partitioned tables and for prediction of data in newly created partitions.

Currently the Oracle documentation only shows new features that are also covered by the Oracle 23c Free Edition. The documentation will get completed when the on-premises version gets released.
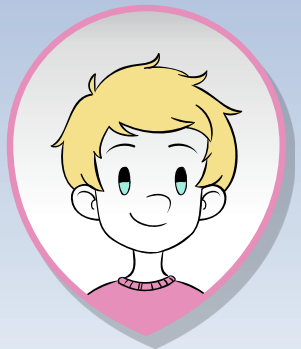
Summary

Oracle 23c has a lot to offer, this applies also to Standard and Enterprise Edition (without Cloud). A lot of long wished for features have been implemented making the life of administrators and developers a little bit easier. Especially the new Machine Learning seems to be an interesting asset, and we look forward to some interesting applications in the future.

About Christian Pfundtner

Christian has been working in IT since 1990 – straight after graduating from high school. At the beginning of 1992, his path led him to Oracle, where he worked in support and got to know Oracle 6. Since then, he has always been connected with the Oracle database and has worked for various Oracle partners in the consulting area. Since 1998, he has also been working as an Oracle trainer in the DBA area. Through his work as a trainer, he automatically had his finger on the pulse of the Oracle database and was also able to apply his knowledge in high-end consulting. He is an Oracle ACE Director, one of the first Oracle Certified Masters in Europe and the founder of DB Masters GmbH.

"What is your favorite new feature in Oracle 23c?"

JSON Relational Duality Views

One of my favorite new features of Oracle Database 23c is JSON Relational Duality Views. This gives me as a relational database developer to work with the exact same data as my Document database-oriented developers. They can work with JSON documents the way they are used to, and I can use my SQL skills. We both use the same data in the format we are most comfortable with.

The Document database-oriented developers can request just about any layout for their JSON documents, and I only have to create a (simple) SQL query to retrieve the data, without having to worry about the Object Relational Mapping because Oracle Database takes care of that. I think that, with this feature, Oracle has a major advantage over their competitors.

Patrick Barel, Oracle ACE Director,
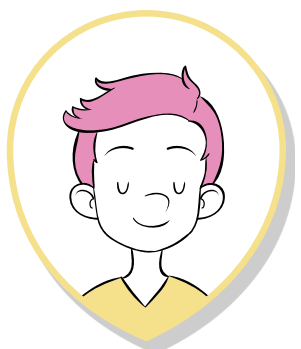*Sr. Oracle consultant*

Table Value Constructor (TVC)

Instead of reading rows from a table/view, you can produce rows on the fly using the TVC. You can use the TVC instead of a table/view in any SQL statement to provide read-only data. This makes it possible to produce rows without querying the DUAL table for each row and to use UNION ALL as a kind of row separator.

I will use a TVC in my utPLSQL tests to define the expected result set. It will make my tests easier to write and easier to read. And I like that. Also, it's an elegant way to get rid of several single INSERT statements in SQL scripts and the like. This will make the scripts smaller and faster to run. Why faster? Because we get the same result with fewer network round trips. And that I like very much. The TVC is an ANSI SQL-92 feature and is available in many database systems. I am glad that it is finally available in the Oracle database as well.
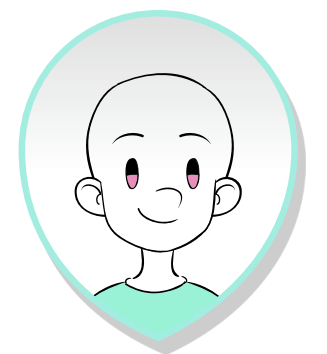
Philipp Salvisberg, Oracle ACE Pro,
Data Engineering Principal

Schema Privileges

In the past we either had to grant object privileges on individual objects, or on all objects in the database. For example, we can grant the select privilege on a single table, or on all tables in the database.

grant select on scott.emp to testuser;
grant select any table to testuser;

Granting database wide privileges such as SELECT ANY TABLE is considered a security risk, but in Oracle 23c we can limit the scope of the grant to an individual schema. For example, we can grant SELECT ANY TABLE on all tables owned by the EMP schema using a single command.

grant select any table on schema emp to testuser;

In addition to granting access to existing tables, this will also grant the same access to any future tables added to the schema. This makes object grant maintenance much simpler compared to previous versions.

Tim Hall, Oracle ACE Director,
DBA, Developer, Author, Trainer

Observability and Databases – Oracle DBAs have an advantage

Steven Lemme
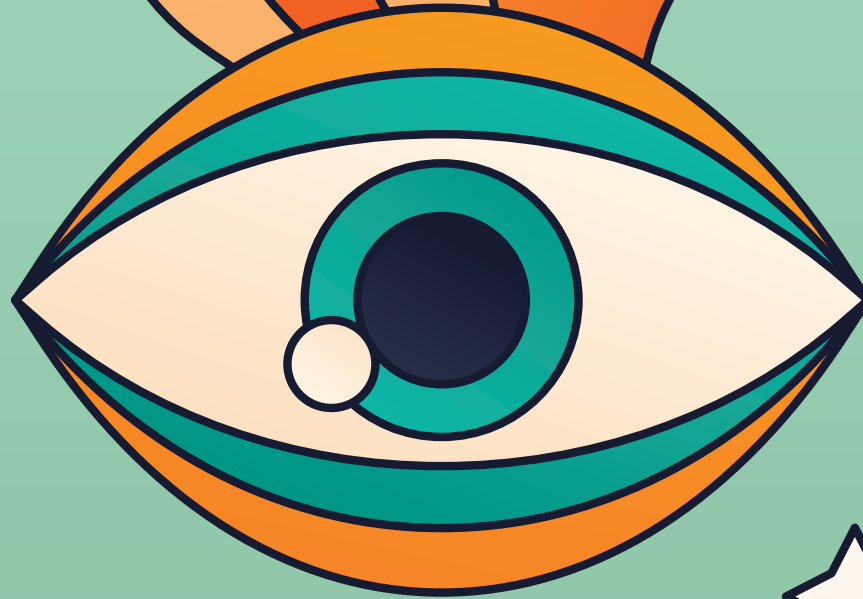
If you're a database administrator and unfamiliar with the term Observability, this article is for you. Perhaps you're one of those DBAs like me who spent a lot of time architecting, acquiring tools, and building an IT Operations Management database monitoring system for the requirements of your data center. Or perhaps you inherited a monitoring system someone else built that you have been tasked with keeping alive. No matter who built it, it's something that helps you keep watch over all your databases to see if there is a problem.

I often meet with businesses and have DBAs and IT Operations personnel tell me they aren't using the cloud, yet in other parts of their company, their Developers and DevOps tell me they do use the cloud. Developers are deploying databases on compute, VMs, containers, or using cloud-native database services in support of line of business application development initiatives. So, what about databases that are not in the data center, deployed in a cloud somewhere? What is watching over those databases, making sure they are available and secure?

Companies continue to adopt agile and modern development practices: DevOps, CI/CD, cloud-native technologies such as microservices, container-based architectures (Docker, Kubernetes), serverless functions, cloud-based database services, and hybrid cloud configurations. Developers are deploying new application components in so many places and faster than ever before so that traditional IT monitoring can't keep pace. Developers also need more abundant and better-quality telemetry to create high-fidelity, context-rich, fully correlated multi-level metrics for their applications, events, logs, traces, SQL transactions, and infrastructure changes. This is where Observability comes in.

In the Development community, data management services or databases are a component of "the stack" – part of the Observability equation referred to as Metrics Events Logs Traces (M.E.L.T.). Cloud-native development includes the instrumentation of apps to emit telemetry data as things occur. This telemetry data can then be observed in real-time as the app runs and is used to adjust configurations to keep the app operating smoothly. This is similar to concepts of process control used by the manufacturing industry, or culture concepts pioneered by Kanban and Deming. The cloud has accelerated the use of Observability because businesses can access the latest technology, bring all the different data together, correlate, and derive insights from it using AI and machine learning.

In contrast, a majority of enterprise monitoring tools in use by IT had been designed as Network or Systems Management tools for data centers supporting applications and databases. Within the IT Operations community, Simple Network Management Protocol (SNMP) has been a source of telemetry – often used to monitor if something in the infrastructure was up, down or a condition changed. When something goes wrong, this information is used to notify of an issue that has occurred so that someone can be dispatched to diagnose and fix it. However, databases and applications are not instrumented with SNMP, so businesses

acquired more tools trying to further monitor and pinpoint in applications and databases where a problem was rooted, troubleshoot it, and fix that specific component of the stack.

DBAs who manage Oracle Databases have found it's the most telemetry-instrumented database in the industry. Each release has even more instrumentation added – just compare the Oracle Database 23c Data Dictionary **documentation** (2785 pages) to that of Oracle Database v7 (355 pages). Some of its telemetry is observable by writing a script, more using Oracle Enterprise Manager (designed for managing the Oracle stack), and even more using Oracle Cloud Infrastructure **Observability and Management**.

Earlier generation tools have focused on the historic performance of standard and tightly controlled onsite infrastructures. However, those tools and methods were not designed for the elastic nature of public, private, hybrid, or multicloud environments.

I advise DBAs to be inquisitive and ask for an example when someone says that monitoring and observability are the same thing. Why? Because there is a difference:

Monitoring: something bad has already happened
○ Monitoring is based on gathering predefined sets of metrics, typically collected using scripts or tools that alert about the state of a change (it was working/it is not working now) or a predefined threshold was reached (condition exceeded). Something has happened, it is having an impact, and it needs to be corrected for the impact to stop.

Observability: something bad is getting ready to happen
○ Observability is based on having insight and the ability to watch for and explore properties and patterns not defined in advance. It's comprised of metric, log, tracing, SQL, and user experience data that enables actively watching and debugging an app and its stack while it's operating, making corrections before something bad happens.

Industry Analysts also have advised to be on the watch for tool vendors who now call themselves an observability tool. These are tools that are not integrated into the infrastructure platform. They aren't integrated at the control plane level, or bolt-on and lack integrated full-stack observability across the application and its infrastructure (on-premises, cloud, hybrid, and multicloud).

What does this mean for DBAs?
Many DBAs have already discovered that when a Developer

uses a standalone Application Performance Monitoring (APM) tool (one that doesn't have deep database or infrastructure insight), problem-solving becomes more of a guessing game. Too often a Developer cites their APM tool identified the issue as SQL and it's the DBA's job to tune and fix it. And some DBAs themselves using a tool that lacks application insight, are telling the Developers they need to recode the app. Continuing to add more monitoring tools is not the answer.

For DBAs, similar to other job roles, new technologies provide a way to do something better, faster, or accomplish something unachievable before. DBAs should request the support of their VPs and Directors of Database Management for the inclusion of Observability as part of their new skills learning. Oracle DBAs should seek out opportunities to apply their deep Oracle Database instrumentation and SQL knowledge in application modernization and cloud-native projects to provide their businesses with an Observability competitive advantage using Metrics Events Logs Traces and SQL (M.E.L.T.S).

It's a misconception that you need to replace what you already have to get started with Observability. Oracle's Observability and Management services are subscription-based so they can be used in parallel with other tools, you can augment or complement them, or use them to fill in gaps. Many businesses have amassed so many different monitoring tools (upwards of 50+ as reported by Industry Analysts) that DBAs, ITOps, and Developers often inherited, so your Observability journey should begin with:

- An inventory of the "technical debt" of monitoring tools the business has accumulated over the years (database monitoring, IT infrastructure monitoring, system monitoring, log monitoring, APM tools, Observability tools, and so on).
- Research vendors' documentation/websites to confirm what versions of Oracle technologies they support today (ensure they aren't just listed as "compatible"). In other words, assess the depth of their Oracle product or database insight (basic Oracle Database 7 single architecture vs. the current **Oracle Database 23c multitenant converged architecture** / **Oracle 23c data dictionary**). While I often find DBAs are aware of the 10x increase in Oracle Database instrumentation and telemetry, most 3rd party tools and Developers are not.
- Reviewing the vendors' documentation/website/roadmaps to confirm what they plan to support modern application deployment architectures (on-premises, cloud, hybrid, multicloud).

Database Observability

Application troubleshooting can be particularly difficult and time-consuming when issues occur at the database level. Database observability goes beyond traditional database monitoring introducing a multi-step iterative process that incorporates relevant telemetry collection, performance analysis, and tuning as first-class citizens to improve application performance and user satisfaction.

For cloud-native apps and application modernization projects, DBAs must move beyond traditional database monitoring and incorporate agile and iterative Database Observability. This is especially true when applications are not operating or performing as they should, and their Developers and DevOps teams lack insight at the database level or the ability to discern whether the root of the issue is in a database, the SQL, or the infrastructure.

DBAs themselves should leverage the rich observability instrumentation and telemetry in Oracle Databases to observe and track database performance, health, and capacity with the end goal of being able to create and maintain a high-performing and available database infrastructure. In addition, they should be advocating and advising their Developer and DevOps peers to incorporate these capabilities to provide end-to-end full-stack observability down to the SQL (M.E.L.T. + S.).

There are multiple advantages to having a Database Observability framework:

- Knowing overall database health at a glance
- Ability to adjust resources up or down depending on the workload
- Tracking of key database performance indicators and analysis using AI and machine learning
- Pinpointing bottlenecks based on database load or the system load, by monitoring CPU load, memory utilization, database wait classes, and more

Conclusion

The reality is that all organizations should be working together to ensure the optimal experience for their customers and the best business success. Observability helps shorten the time and effort between applications, infrastructure, and operations teams to optimize performance and perform better capacity planning that enables business stakeholders and organizations to improve their cost reduction and annual budget planning decisions. Modern manageability is more agile and enables the ability to bring together and manage

different environments without having to rip and replace what you already have.

When Lines of Business work to modernize or migrate applications, it is important to include IT in project planning to prevent overall negative company impact that can impede modernization efforts in several ways.

Monitoring of databases in on-premises data centers may seem to be fine, but as more databases are deployed outside of those well-known environments in the cloud, DBAs struggle and become overwhelmed handling hybrid, multicloud complexities and new challenges that their older tools are ill-equipped or not designed to handle.

In parallel and often in different organizations or lines of business, Development and DevOps teams think they have it figured out and buy another tool or code custom Observability that is not inclusive of existing on-premises infrastructure. Some Development team members don't even engage IT or DBAs in modernization projects thinking their role is to just manage legacy infrastructure and applications.

Modern businesses require Observability that goes beyond traditional IT Operations monitoring and detection methods and extends beyond M.E.L.T. down into the database layer. So be sure your DBAs and Developers collaboratively work together, using the latest technology to ensure the optimal experience for your customers and the best business success. To aid these efforts, Oracle was one of the first in the industry to provide an integrated Observability and Management platform. If your business leaders are not familiar with it, they can learn more about it at **oracle.com/manageability**. For DBAs, Developers, App Admins, and other team members there are **Blogs on using it to solve problems** and free virtual **LiveLabs Hands-on workshops** where they can try it.

About Steven Lemme

As Senior Principal Product Manager at Oracle, **Steven** helps customers and partners be successful in managing their Oracle products on-premises and in the cloud. He creates training, volunteers, and contributes to Oracle communities, helps support Oracle's ACE program, and speaks at events worldwide. He's an Oracle Master DBA, Oracle book and Data Dictionary poster author, past Oracle Partner, IOUG Director, and recipient of the IOUG Ken Jacobs award for service to the Oracle User Group community. He helped create the first Oracle Community Cloud Experience hands-on lab workshop and Cloud SIG and routinely presents on the impact of business transformation, new technology, and the evolving role of the DBA.

Bridging Worlds: Unleashing the power of JSON within the Oracle Database over pure document stores

Julian Dontcheff

Oracle Database presents a compelling alternative to traditional document store databases. The converged Oracle approach combines the flexibility of JSON document storage with the robustness, security, high availability, and scalability of a relational database system. By seamlessly integrating JSON capabilities, Oracle Database empowers developers to model and query complex, semi-structured data alongside structured data in a single, unified environment. This approach of using a converged database eliminates the need for separate document store databases (MongoDB, DynamoDB, Databricks, Cosmos DB, etc.), reducing complexity, streamlining data management workflows and the enabling the use of enterprise features like partitioning, compression, data redaction, etc. which are already part of the Oracle database. Additionally, the Oracle Database's optimized JSON processing capabilities ensure efficient storage, retrieval, and manipulation of JSON documents, providing superior performance and much lower cost compared to standalone document stores. This article explores the key benefits of adopting JSON and JSON Relational Duality within the Oracle Database, emphasizing the synergy achieved between flexibility and reliability in modern twenty-first century data-centric applications.

In the database community, there is a rather strong perception that simple and pure JSON is better for development which is simply false. The reality is that duality is by far the better choice. This means that applications can access (create, query, modify, delete) the same data as a set of JSON documents or as a set of related tables, and both approaches can be employed at the same time! This is a **huge** thing! Here is what Carl Olofson, research vice president, Data Management Software, IDC stated (**https://www.idc.com/getdoc.jsp?containerId=US49859922**):

"Oracle has created a feature that allows the user to create a JSON document template (called JSON Duality View) that specifies the layout of the document, but with data that is mapped from the relational database. This is a revolutionary capability that is likely to be emulated throughout the industry in the coming years."

With JSON duality views, data is stored in relational tables in a normalized way and that data can be exposed to applications

as JSON documents. Storing the data in relational tables delivers all the benefits of the relational model, whereas by exposing the same data as JSON documents, apps are getting the schema flexibility and several other benefits of the document store model.

Let us consider if it make sense to use a dedicated single-purpose document database or the Oracle converged database for straining JSON data. Now, as there are nowadays more than 50 document store databases (as of early 2023), let us start first with the top arguments highlighting why using JSON within the **Oracle Database** is advantageous compared to using **MongoDB**. Here are the top eight reasons:

1. Unified Data Management:
Oracle Database allows for the seamless integration of JSON with traditional relational data, providing a unified platform for managing structured and semi-structured data. In contrast, MongoDB is a NoSQL document store designed primarily for JSON-like documents, which might lead to a fragmented data management approach when dealing with diverse data types. Note that moderns applications use several data types in the underlying database!

2. ACID Compliance:
Oracle Database adheres to ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring data integrity and reliability even in the face of system failures. MongoDB, while offering high availability and scalability, sacrifices some ACID properties for eventual consistency, making it potentially less suitable for use cases with strict transactional requirements.

3. Transaction Support:
Oracle Database provides robust support for transactions, allowing developers to execute multiple operations as a single, atomic unit. MongoDB's transaction support historically lagged behind, and though it has improved, it may not offer the same level of transactional capabilities as Oracle in certain scenarios. In fact, few databases can!

4. Mature Query Language:
Oracle Database offers SQL as a powerful and mature query language for both relational and JSON data. This enables developers to leverage the expressive capabilities of SQL for analytical purposes and reporting, while MongoDB's query language might be considered less feature-rich and expressive in comparison.

5. Security and Authentication:
Oracle Database has a long-established reputation

for security features, including robust authentication mechanisms, access controls, and encryption. MongoDB also provides security features but might be perceived by some experts as playing catch-up in terms of maturity and depth of security offerings.

6. Enterprise Support:
Oracle provides comprehensive enterprise-level support for its database products, ensuring timely updates, patches (including security patches), bug-fixing, and a dedicated support team. MongoDB offers a support subscription model, but the depth and breadth of support services might not match the long-standing enterprise support ecosystem of Oracle.

7. Integration with Existing Systems:
Many enterprises already rely on Oracle Database for their core business and mission critical applications. Leveraging JSON within the Oracle Database allows for streamlined integration with existing systems, leveraging established infrastructure, security, and management practices.

From operational point of view, it does not make sense to have a separate/additional DBA team for MongoDB, that is one of the main reasons why customers are now moving out of MongoDB. IT organizations prefer to manage less database brands by the same operational DBA team. And it is not just the DBA team being involved. There is also usually a separate storage, backup/recovery and networking teams which also prefer less database brands to manage. Ideally, managing more than three database brands is when operation administration starts to get tedious and complex.

8. Price/cost:
Both the on-premises version of **MongoDB** and the cloud one **MongoDB Atlas** are about double the price of an **Oracle** database (on-prem EE or Autonomous Database in OCI). And here we are not talking about the free/community/developer editions of **MongoDB** and **Oracle**, but the ones used by enterprises for business and mission critical applications.

Just as an example looking at the price list we see that Oracle Autonomous JSON database costs $0.3226 per hour for 1 OCPU (= $240 per month) and $118.40 for 1TB of storage per month (together $358.40 per month) (**https://www.oracle.com/autonomous-database/pricing/**) while the M140 tier of MongoDB Atlas which comes also with 1TB of storage has a monthly cost of $7914.65 per month ($10.99 per hour) (**https://www.mongodb.com/mongodb-on-aws/pricing**). You can see from this example that even with 10 OCPUs, the Oracle JSON database is 2 times cheaper than MongoDB Atlas.

In order to be fair, taking into account the reasons listed above, it is important to note that the choice between Oracle Database with JSON support and **MongoDB** depends on specific project requirements, preferences, and the overall ecosystem within which the database is deployed. Each solution has its strengths and weaknesses, and the decision should be based on a thorough analysis of the particular use case. But of course, taking into consideration all 8 points listed above!

Besides **MongoDB**, other popular document databases are **Databricks**, **Cosmos DB**, and **DynamoDB**.

Databricks, while powerful, can be complex to set up and manage. Organizations may face a learning curve, particularly if they are new to the Apache Spark ecosystem. Databricks costs can be significant, and users should carefully consider the pricing model, especially for large-scale deployments.

Databricks is complex and difficult to learn and understand. On one hand it is due to the many tools, integrations, and features available. Here the Oracle Autonomous Database is the better alternative – especially for new developers and DBAs. Having Scala as a primary language is another issue for many people as it is considered hard and difficult to learn as it is more complex than other programming languages. Though Databricks supports SQL, R, Python, and Scala, commands issued in non-JVM languages require additional transformations to be executed on top of a JVM process.

Databricks charges on consumption and thus unless you can find an expert you should learn yourself on how to optimize the platform from the start, it will save you a lot of money. But granted the small community, the latter options might be easier that the former one. The Oracle Autonomous Database in that aspect is self-tuning and fully automated.

Cosmos DB can be expensive, especially for resource-intensive workloads. Users should carefully manage and monitor their resource consumption to control costs. While Cosmos DB supports SQL-like queries, some developers might find certain limitations in the query language compared to traditional SQL (for example no ad-hoc queries are supported). Not to mention the long list of limitations the database has! Azure Cosmos DB can be slow when it comes to large amounts of data and complex queries, as it is optimized for high availability and consistency over raw performance. Cosmos DB does not support distributed transactions across multiple partitions, making it difficult

to implement complex business logic that requires atomic operations.

DynamoDB's query language is more limited compared to SQL. DynamoDB uses a simple query language, and while it is efficient for key-value access patterns, it lacks the expressive power of SQL. Complex queries or aggregations may require additional processing on the client side. DynamoDB uses a provisioned throughput model, which may lead to over-provisioning and increased costs if not carefully managed. Designing the table schema in DynamoDB requires careful consideration of access patterns. Changes to the schema may be challenging, and denormalization might be necessary, leading to potential data redundancy. DynamoDB uses a provisioned throughput model where users need to specify the read and write capacity units. While this provides control, it can lead to over-provisioning or under-provisioning, impacting costs and performance.

Creating secondary indexes in DynamoDB can incur additional costs. Users need to be mindful of the number and types of indexes they create to avoid unnecessary expenses. DynamoDB does not provide built-in transparent caching mechanisms. Users needing caching for read-heavy workloads might need to implement caching solutions independently. DynamoDB transactions are limited to a single table, and transactions across multiple tables are not supported. This can be a limitation for scenarios requiring complex multi-table transactions. DynamoDB allows you to write (Put, Update or Delete) multiple items across multiple tables within a single atomic transaction using the TransactWriteItems API operation. However, the problem is that the TransactWriteItems operation has an upper limit of 25 items that can be updated in a single transaction. A transaction cannot contain more than 4 MB of data and no two actions in a transaction can work against the same item in the same table. For any AWS account, there is an initial quota of 2,500 tables per AWS Region. For a table, we can define a maximum of 5 local secondary indexes and 20 global ones.

While Document Stores provide backup and restore functionality, there are limitations on the frequency and granularity of backups. Users should carefully plan their backup and recovery strategies.

It's crucial to note that the choice between these databases depends on specific project requirements, scalability needs, and the overall architecture of the application. Each database solution has its strengths and weaknesses, and the decision should be based on a thorough understanding of

the particular use case and desired trade-offs. It is important to note that the concept of "JSON Relational Duality" in Oracle Database 23c refers to the ability to seamlessly integrate JSON data with traditional relational data, offering a unified approach to data modelling and query execution. This approach provides the following major benefits over pure document databases, where JSON is the primary or sole data model.

1. Relational and JSON in one go:
Oracle Database's JSON Relational Duality allows developers to work with both structured (relational) and semi-structured (JSON) data in the same database system. This eliminates the need for maintaining separate systems for different data models, simplifying data management and reducing architectural complexity.

2. Data Consistency and Integrity:
By leveraging a relational database system with ACID compliance, Oracle ensures data consistency and integrity. Transactions can span both relational and JSON data, maintaining the reliability and integrity of the entire dataset.

3. SQL for JSON Querying:
Oracle Database supports SQL for querying both relational and JSON data. This enables developers to use the familiar and powerful SQL language to query and analyze JSON documents alongside traditional structured data, providing a unified querying experience.

4. Efficient Joins and Relationships:
With JSON Relational Duality, developers can establish relationships between JSON and relational data using foreign keys and leverage efficient joins. This is particularly beneficial when dealing with complex data models that involve relationships between different types of data.

5. Flexible Schema Evolution:
Oracle Database allows for flexible schema evolution, enabling developers to modify and evolve the structure of JSON documents and relational tables without requiring significant changes to the application logic. This flexibility is crucial in dynamic and evolving application scenarios.

Security features inherent to Oracle Database, including fine-grained access controls, encryption, compression, and authentication mechanisms, apply uniformly to both relational and JSON data. This integrated security approach simplifies the implementation of consistent security policies across all data types.

6. Enterprise features for business and mission critical applications:

Oracle provides comprehensive enterprise-level support for its database products. This includes regular updates, patches, and a dedicated support team, ensuring the reliability and performance of the entire database system, regardless of the data model.

7. Ecosystem Integration:

Oracle Database with JSON Relational Duality seamlessly integrates into the broader Oracle ecosystem, allowing users to leverage additional Oracle technologies and tools for analytics, reporting, and business intelligence.

8. Lingua Franca:

Any types of applications can make use of the same data directly using languages such as SQL, PL/SQL, Python, C, Java, and JavaScript. You need not adapt an existing database features or code that make use of the tables data in order to use JSON documents. As already explained above, in Oracle Database 23c, JSON Duality Views allow you to treat JSON data as if it were relational data, enabling you to use SQL to query and join JSON documents alongside traditional relational tables.

In summary, Oracle's approach to JSON Relational Duality provides the benefits of a unified data management system, enabling developers to work with both relational and JSON data in a cohesive manner. This approach combines the flexibility of JSON with the reliability and transactional capabilities of a mature relational database system. Using pure document databases requires that you actually know and understand to a certain extent the concepts of relational databases. By just following the hype, you might find yourself at some point in a dead-end as often document stores do not deliver on the promises you read about before adopting them. Document-oriented databases for storing JSON are part of the past. **Reality needs Duality; Pure JSON is for Fiction**.

About Julian Dontcheff

Julian is the Database Product Manager for PL/SQL and JSON at Oracle Corporation. Julian was the first Oracle Certified Master in Europe (back in June 2002) and he was also an Oracle ACE (Director) since 2009 (as an Oracle employee now an alumni). In the past, Julian was the Global Database Lead of Accenture. His primary responsibility was managing and leading the Global Database Platforms which included Autonomous Cloud, IaaS, PaaS, Database Services, Engineered Systems, Security and all other areas falling under Database Engineering. He was also the Global Managing Director of the Accenture-Enkitec Group. Before Accenture, Julian headed Nokia's Database Technology team and the Continual Service Improvement unit in Nokia Computing. The DBA team took care of about 2,300 databases worldwide, mostly Oracle. Julian has experience and knowledge in the retail, banking, telecom, and high-tech sector. He has been a consultant and database architect for several Fortune 500 companies and global IT projects.

Oracle Autonomous Database all the way

Divit Gupta

In the quest to modernize enterprise computing, IT leaders are compelled to shift towards a cloud-based model, aiming for increased efficiency while minimizing costs. This transition necessitates enhanced cybersecurity measures without compromising the reliability of crucial workloads. A comprehensive transformation involves both existing and new applications, with the public cloud playing a pivotal role. However, a full-scale migration to the public cloud may not be practical. Oracle's Autonomous Database stands out as a solution that facilitates this transformation by introducing a new era of data management, available in both public cloud and on-premises options, ensuring immediate cost savings and agility enhancements.

Similar to an autonomous car, the Oracle Autonomous Database offers a performance and reliability level beyond what manually managed databases can achieve. In comparison to a database under manual management, the Autonomous Database delivers cost savings, superior performance, increased availability, and eradicates the potential for human error.

Key Capabilities

Let's assess some of the key capabilities of Oracle's Autonomous Database.

Self-Securing

The Autonomous Database ensures enhanced security compared to manually operated databases, as it autonomously safeguards itself without relying on the availability of an administrator. This defense mechanism is effective against both external and internal threats. Security patches are automatically applied every quarter, minimizing the window of vulnerability compared to traditional Oracle databases. In the event of a zero-day exploit, off-cycle patching can swiftly occur. The Autonomous Database employs a rolling patching approach across cluster nodes, ensuring self-securing without causing application downtime. Additionally, the database features always-on encryption, and future enhancements will allow customers to manage their encryption keys for added security. Oracle's Data Masking and Redaction technologies further contribute to security by concealing sensitive data for specific users or workloads and masking it in test databases.

Self-Repairing

The Autonomous Database surpasses the reliability of manually operated databases through various automated mechanisms. Upon startup, it autonomously configures a triple-mirrored scale-out setup in a regional cloud datacenter, with the option for a full standby copy in another region. The database adeptly recovers from physical failures at both server and datacenter levels and can revert data to a specific point in time to rectify user errors. Rolling updates across cluster nodes maintain application availability during database, cluster ware, OS, VM, hypervisor, or firmware updates. In the event of an impending error, the database employs AI diagnostics to identify the root cause, with nightly backups serving as an additional safety net. Future updates can be validated through replaying the full production workload in a parallel testing environment. Oracle guarantees a 99.995% uptime, covering planned maintenance and other common sources of downtime.

Tailored for Diverse Workloads

Modern vehicles are designed for specific purposes such as family transportation, vans, pickup trucks, or sports cars. Similarly, the Autonomous Database encompasses a unified set of technologies offered through various products, each tailored to address distinct workloads (also see **Fig. 1**):

1. **Data Warehousing:** The Oracle Autonomous Data Warehouse is optimized for simplicity and efficiency, making it ideal for data marts, reporting databases, and data warehousing. It has been available since March 2018.
2. **Transaction Processing and Mixed Workloads:** The Oracle Autonomous Transaction Processing is crafted to support mission-critical enterprise applications, including mixed workloads and real-time analytics, without compromising application performance. It has been available since July 2018.

As Oracle continues to evolve, the autonomous principles of self-driving, self-securing, and self-repairing will be extended to other types of databases. Furthermore, the Autonomous Database serves as a cloud-native enterprise-class foundation, providing IT leaders with a robust platform for new application and data science development.
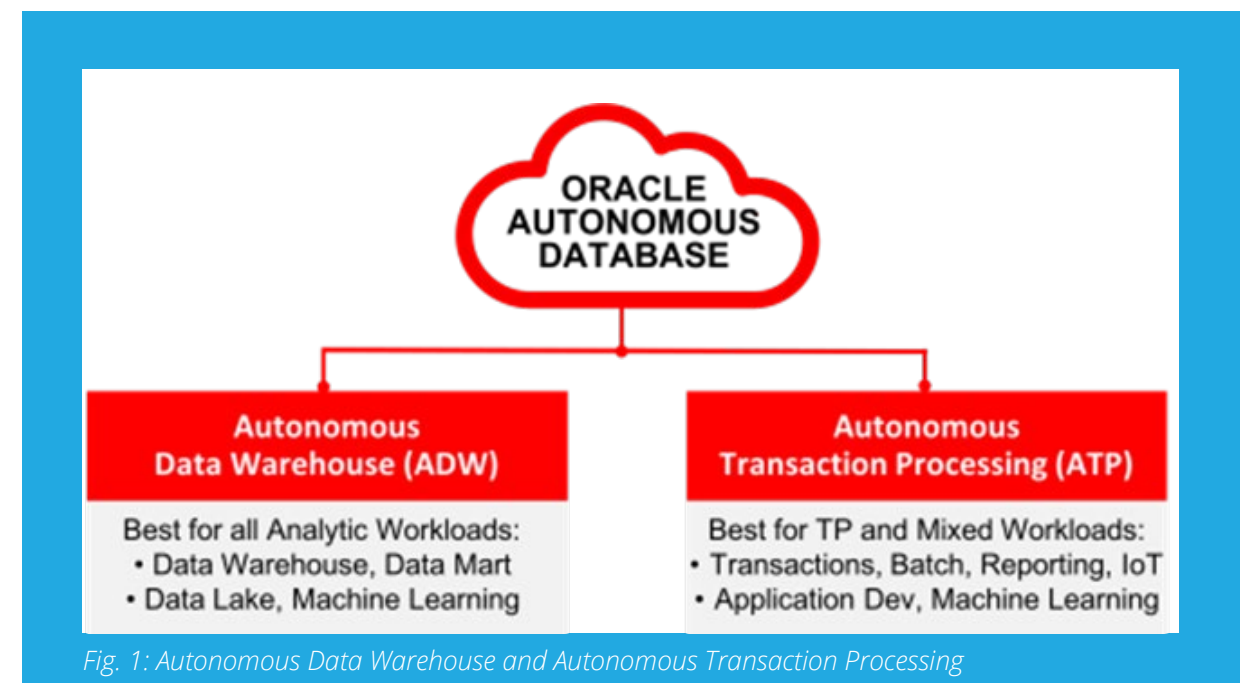

Fig. 1: Autonomous Data Warehouse and Autonomous Transaction Processing

Oracle Application Express (APEX), a low-code development framework, is pre-configured and ready for use immediately upon provisioning an Autonomous Database. This user-friendly development environment allows developers to efficiently handle tasks such as data loading, database object management, REST interface development, and the creation of applications that function seamlessly on both desktop and mobile devices. Much of the development process is automated, covering aspects like security, authentication, database interactions, input validation, session state management, and various other dependencies, all of which work effortlessly out of the box.

The Autonomous Database delivers an instant platform for application developers, offering a range of data management methods while maintaining the simplicity of a self-managing database. With a simple button push, app developers can provision a database capable of handling mission-critical tasks. Additionally, the Autonomous Database provides advanced SQL and PL/SQL support, enhancing developer productivity by minimizing the amount of application code needed to implement complex business logic.

The Enterprise Edition

The basis of the Autonomous Database is rooted in Oracle Database Enterprise Edition, the very database that customers have relied on for years to handle a diverse range of workloads, encompassing Data Warehouse, Analytics, and Transaction Processing. The Autonomous Database streamlines management complexity by utilizing automated configuration settings, as elaborated later in this document. Furthermore, the autonomous database is constructed on and automates various advanced database technologies unique to Oracle, including:

- Real Application Clusters for scale-out, failover, and online patching
- Online operations for seamless schema changes
- Active Data Guard for database-aware Disaster Recovery
- Parallel SQL for optimal throughput
- Database Containers for enhanced agility

- Database In-Memory for superior performance
- Transparent Database Encryption for robust data protection
- Database Vault for role segregation

Provisioning Databases

In a conventional data center setting, administrators undertake various tasks to finalize the database provisioning process, a process that typically spans days or even weeks. These tasks encompass identifying and allocating compute and storage resources, installing and configuring system software, installing and configuring database software, configuring Oracle Grid Infrastructure, configuring Oracle Real Application Clusters, and installing and configuring monitoring software. The Autonomous Database streamlines these tasks, executing them automatically, and promptly addressing any arising issues either automatically or through Oracle's Cloud Operations team. The provisioning of Autonomous Databases is notably straightforward, involving users providing essential information for database provisioning along with optional advanced settings.

Provisioning – Basic Settings

Customers can rapidly provision Highly Available Autonomous Databases, which run RAC databases on Exadata Infrastructure, within a matter of minutes. This can be effortlessly accomplished through either the service console or the REST API, requiring only a few straightforward pieces of information to be provided.

Autonomous Databases are swiftly provisioned within minutes to accommodate the selected workload type. Oracle automatically conducts nightly backups for all Autonomous Databases, retaining these backups for a duration of up to 60 days.

Autonomous Database Administration

Autonomous Database streamlines nearly all administrative tasks associated with Oracle databases, tasks that traditionally demand substantial time and effort from Database Administrators, System Administrators, and other IT Professionals. The core automation capabilities of

Autonomous Database are listed below.

- Provisioning
- Scaling
- Management
- Security
- Data Protection
- Optimization

Customers maintain the required level of control to provide database services aligned with business requirements, all while leveraging this unparalleled degree of automation. Oracle offers a suite of powerful tools that enable customers to manage Autonomous Database seamlessly within the broader infrastructure of any enterprise, facilitating the adoption of Autonomous Database without significant disruption to Information Technology teams.

Summary

Oracle Autonomous Database, a revolutionary platform, brings unparalleled automation to data management, ensuring optimal performance, security, and reliability. With features like self-driving, self-securing, and self-repairing capabilities, it eliminates the need for extensive manual interventions, providing a robust and efficient solution. Available in two editions, Autonomous Data Warehouse and Autonomous Transaction Processing, it caters to different workloads, ensuring simplicity, cost-effectiveness, and advanced support for SQL and PL/SQL. Additionally, it offers tools like Oracle Application Express for low-code development and integration with existing infrastructure, empowering businesses with flexibility and control. Provisioning is streamlined, taking just minutes, and daily backups are automatically managed, enhancing data protection. The automation extends to administrative tasks, saving time for Database Administrators and IT Professionals. The solution strikes a balance between automation and customer control, ensuring seamless integration into diverse enterprise infrastructures.

Learn more about Oracle's Autonomous and database consolidation thru the references and links below.
https://www.oracle.com/autonomous-database/
https://www.oracle.com/a/ocom/docs/database/oracle-autonomous-database-strategy-wp.pdf
https://blogs.oracle.com/database/post/a-guide-to-why-and-how-to-consolidate-databases

About Divit Gupta

Divit, a seasoned IT professional with over 20 years of industry expertise, he spearheads technical partnerships, defines team vision, and champions new strategic endeavors. As the host of popular podcasts like "Tech Talk with Divit", "Live Labs with Divit", and "Cloud Bites with Divit", he showcases technological initiatives and leadership. In 2022 and 2023, he served as Oracle TV's correspondent for CloudWorld. His passion for knowledge sharing extends to international conference talks, technical blogs, and multiple books on emerging technologies.

Stefan Oehrli

Enhanced Security Scans using the DBSat

Learn more about the groundbreaking advances in Oracle database security made with the introduction of DBSAT 3.0 and the recent update to DBSAT 3.1. This article explores the innovative features and enhancements introduced in these releases and provides valuable insights for technical professionals looking to strengthen their database systems.

In November 2023, Oracle introduced version 3.0.0, a comprehensive upgrade of its Database Security Assessment Tool (DBSAT), taking a major step forward in the optimization of database security and compliance. This was followed in January 2024 by the release of DBSAT 3.1.0, which focused on the latest improvements. This article looks at the key features and enhancements of these releases and how they improve the security and efficiency of Oracle database environments.

Fig. 1 DBSat Icon (© Stefan Oehrli)

Key Features and Enhancements in DBSAT 3.0.0

DBSAT 3.0.0 marked a significant update with several key improvements:

- **STIG V2R8 Compliance:** The update introduced 30 new STIG findings and revised group IDs for compliance with the latest Security Technical Implementation Guide standards.
- **Enhanced Auditing and Security:** New auditing results were added, including five specific additions, with updates to all existing ones. Noteworthy is the focus on Sensitive Data and Transparent Sensitive Data Protection (TSDP), along with the integration of the Oracle Database 23c SQL Firewall.
- **Sensitive Data Discovery:** Enhanced capability to identify Indian PAN and Aadhaar numbers, widening its scope in sensitive information search.

- **Improved Clarity and Quality:** Introduction of one-line summaries for each check for better understanding, labeling according to Oracle best practices, and notes on unsupported features in Oracle Database 23c.
- **Operational Enhancements:** New options to exclude certain users from reports, removal of Python dependency, performance optimizations for faster data collection, compatibility with Linux 64-bit Arm, and support for Oracle Database 23c.

Enhancements in Release 3.1.0

The DBSAT 3.1.0 release in January 2024 focused on further enhancements:

- **Alignment with CIS Benchmark v1.2:** Included 10 new findings based on CIS recommendations for Oracle Database 19c, with updated references.
- **New Finding for Autonomous Database Serverless:** Introduction of a finding related to pre-authenticated URL requests.
- **Comprehensive Security Checks:** New checks for user profile limits, EXECUTE permissions on various packages to PUBLIC, and database security and administration-related permissions.
- **Auditing and Operating System User Configurations:** Addition of checks for auditing actions on synonyms and operating system user configurations in pluggable databases.
- **Enhanced Existing Findings:** Improved logic in user expiry checks, optimizations in application owner assessments, and updated TDE recommendations for Oracle Database 23c.

These collective updates in DBSAT 3.0.0 and 3.1.0 significantly enhance the tool's capabilities, offering a comprehensive and current assessment of Oracle Database security in line with evolving standards and practices.

DBSAT in Action

Let's walk through a simple example of using *DBSAT* 3.1 to evaluate a database running Oracle 19c. For this purpose, I am using a database container on my MacBook Pro running Oracle Database 19c for ARM Linux. It's worth noting that *DBSAT* now supports this configuration as well.
To find DBSAT, the easiest way is to visit the *Oracle Database Security Assessment Tool (DBSAT)* product page. From this page you will be redirected to Oracle support document 2138254.1 to download the tool. Please note that an Oracle account is required to access the download.
As mentioned above, Python is no longer required to run DBSAT. All you need is an Oracle database, the appropriate

credentials and a JDK. The JDK located in your Oracle home directory will suffice.

We can proceed by unpacking the package into the $ORACLE_BASE/product directory.

```
unzip dbsat.zip -d $ORACLE_BASE/product/dbsat_3.1.0
```

If not yet set define the *JAVA_HOME* environment variable

```
export JAVA_HOME=$ORACLE_HOME/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

Verify if we can run dbsat and display its new usage.

```
$ORACLE_BASE/product/dbsat_3.1.0/dbsat -h
```

If successfully you should see something like this.

```
oracle@cdbua190:~/ [CDBUA190] $ORACLE_BASE/product/dbsat_3.0.0/dbsat -h

Database Security Assessment Tool version 3.1 (Jan 2024)

   Usage: dbsat collect  [ -n ] <database_connect_string> <output_file>
          dbsat report   [ -a ] [ -n ] [ -g ] [ -x <section> ] [ -u <user> ]
                              <input_file>
          dbsat discover [ -n ] -c <config_file> <output_file>

   Options:
      -a  Report with all user accounts, including locked and schema-only,
           Oracle-supplied users
      -n  No encryption for output
      -g  Show all grants including Common Grants in a Pluggable Database
      -x  Specify sections to exclude from report (may be repeated for
           multiple sections)
      -u  Specify users to exclude from report
      -c  Configuration file for discoverer
```

Run the Assessment
Like its predecessors, DBSAT can be operated in three different execution modes: collect, report, and discover:

1. **dbsat collect:** This mode collects data from the specified database.
2. **dbsat report:** A comprehensive security assessment report is generated.
3. **dbsat discover:** This function identifies sensitive data located in the database.

Let us now proceed to collect data from the container database CDBUA190 using the user account *system*.

```
$ORACLE_BASE/product/dbsat_3.1.0/dbsat collect -n system@CDBUA190
$HOME/CDBUA190_v3.1.0
```

Here is an excerpt from the output of the DBsat call without the license and version information:

```
...
Setup complete.
SQL queries complete.
Warning: Exit status 256 from OS rule: dbcs_status
/bin/cat: /u00/app/oracle/product/19.0.0.0/ldap/admin/fips.ora: No such file
or directory
Warning: Exit status 256 from OS rule: fips1.ora
/bin/cat: /fips.ora: No such file or directory
Warning: Exit status 256 from OS rule: fips2.ora
/bin/ls: cannot access '/u00/app/oracle/product/19.0.0.0/rdbms/log/diag': No
such file or directory
Warning: Exit status 512 from OS rule: diag_dest_home
OS commands complete.
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.19.0.0.0
DBSAT Collector completed successfully.
```

As you can see, there are a few warnings for certain OS checks, particularly regarding the FIPS configuration and a Diag directory located below $ORACLE_HOME. These warnings can typically be ignored, especially if FIPS is not being utilized in your environment.

If you wish to do so, processing the generated JSON document is an option. However, creating a report from this data is usually more straightforward. DBSAT offers the flexibility to create reports in different formats such as JSON, text, HTML or Excel. It is important that you are in the DBSat directory when creating the report so that the tool can find the xlsxwriter. After executing the "dbsat collect" command, the report can be created as shown below.

```
cp $HOME/CDBUA190_v3.1.0.json  $ORACLE_BASE/product/dbsat_3.1.0/
cd $ORACLE_BASE/product/dbsat_3.1.0/
./dbsat report -n -a -g CDBUA190_v3.1.0
```

Additional variations on utilizing DBSat are detailed in the User Guide available in the online documentation. This includes features such as output encryption, including common grants in a pluggable database, excluding specific sections or users, and more.

Analyze the Report
When examining the HTML report, the additional details for each finding become immediately evident. For instance, consider the example below, which shows the rule checking for the latest security patch.
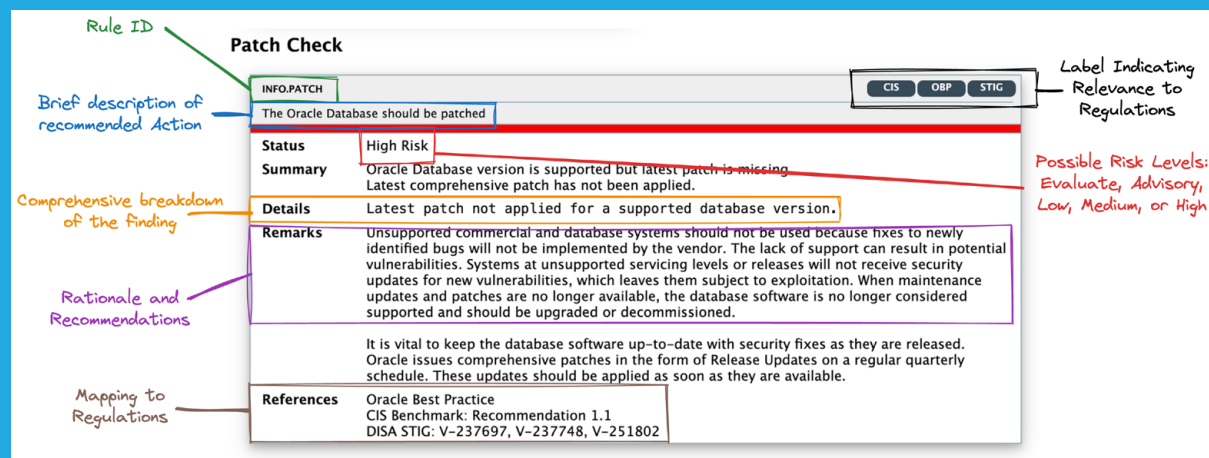
Fig. 2: Description of the fields for a security finding (©Stefan Oehrli)

DBSat has significantly broadened its coverage to include more than 120 deliverables. These encompass not only STIG and CIS recommendations but also Oracle Best Practice (OBP) topics. This expansion is especially advantageous given that Oracle's feature and release cycles don't always align with updates to existing standards and frameworks. It's crucial to note that each new Oracle release introduces fresh features. These features demand proper configuration and can have a considerable impact on database security.

DBSAT stands out for those managing databases with a long history, including multiple migrations. It offers crucial desupport information pertaining to the latest Oracle database version, 23c. This aspect is invaluable, ensuring that even databases with a complex, evolved history maintain compatibility and security with the newest Oracle technologies.

Further Information and Tips on Additional Tools

Collect and *Report* are the most frequently used functions of DBSat, but it is important to know that these are not the only options. The *Discover* mode, for example, offers a valuable feature: it scans the data dictionary or table definitions to identify columns with sensitive data. Oracle provides a number of pattern files in different languages to help detect tables and columns with sensitive data. These pattern files are customizable and can be adapted to specific requirements. The DBSAT documentation contains some examples and use cases for the Discover mode.

In addition to the well-known features, two lesser known but equally useful tools are available in Oracle Support Note **2138254.1:**
- **dbsat_extract:** This Python script is designed to extract specific results based on their identifiers. It is particularly useful for focusing on specific areas of interest or concern within the DBSat results.
- **dbsat_diff:** Another Python script, dbsat_diff, is used to compare two DBSat reports to identify differences. This is particularly useful when tracking the evolution of database security over time or evaluating the impact of changes to the database environment.

These tools are particularly useful when processing DBSAT JSON files on a regular basis. They provide a more detailed and customized approach to assessing database security, enabling in-depth analysis and more informed decisions to maintain robust security standards in Oracle databases.

Conclusion
DBSAT 3.0.0 and the update to 3.1.0 represent a significant leap forward and offer numerous enhancements that improve not only functionality but also usability. One outstanding feature is the newly gained independence from Python, which simplifies deployment directly on the DB server or remotely from a DBA workstation.

The security checks have been carefully enhanced and prepare the tool for the upcoming Oracle Database 23c. Updates such as STIG-V2R6 compliance and Oracle Best Practice tagging help to interpret and prioritize results. The revised format of the report, which now includes clear explanations, risk levels and best practice guidance for

each finding, greatly aids understanding and remediation of security issues.

In addition, DBSAT provides flexibility in handling large-scale findings. Users can streamline assessments and reports by excluding specific users or areas and focusing on critical areas.

Additionally, DBSAT's integration with Oracle Data Safe, Oracle Audit Vault and Database Firewall underscores its importance. The enhancements in DBSAT 3.0.0 and 3.1.0 not only increase the standalone capabilities, but also enrich the security capabilities of these integrated Oracle products.

Additional Resources
Some links and references related to this topic.

- Oracle Database Security **a technical primer**
- Oracle Database Security Assessment Tool **DBSat**
- Oracle Database Security Assessment Tool **Documentation**
- LiveLabs - **Database Security Assessment Tool (DBSAT)**
- Oracle Support Document **2138254.1** Oracle Database Security Assessment Tool (DBSAT)
- Oracle Support Document **2484219.1** Common Questions and Issues For Oracle Database Security Assessment Tool (DBSAT)
- Oracle Support Document **2651827.1** Does DBSAT Scan for all of the STIG And CIS Benchmark Controls?

About Stefan Oehrli

Stefan, an Oracle ACE Pro and Tech Architecture Manager at Accenture, brings over two decades of expertise in Oracle database systems. With a strong focus on physical design, implementation, and operation, Stefan is renowned for his deep knowledge in Oracle database security, including Oracle Advanced Security and Oracle Unified Directory. He excels in implementing robust security measures and is a sought-after speaker at conferences and user group meetings. Notably, Trivadis, a renowned Oracle company, is now part of Accenture Technology, further strengthening their capabilities in delivering comprehensive Oracle database security solutions. Stefan is co-author of the book Der Oracle DBA (Hanser, 2016). He lives with his family in Muri, Switzerland.

Sybase to Oracle Migration – Seamless and Smooth

Valli Ramanathan
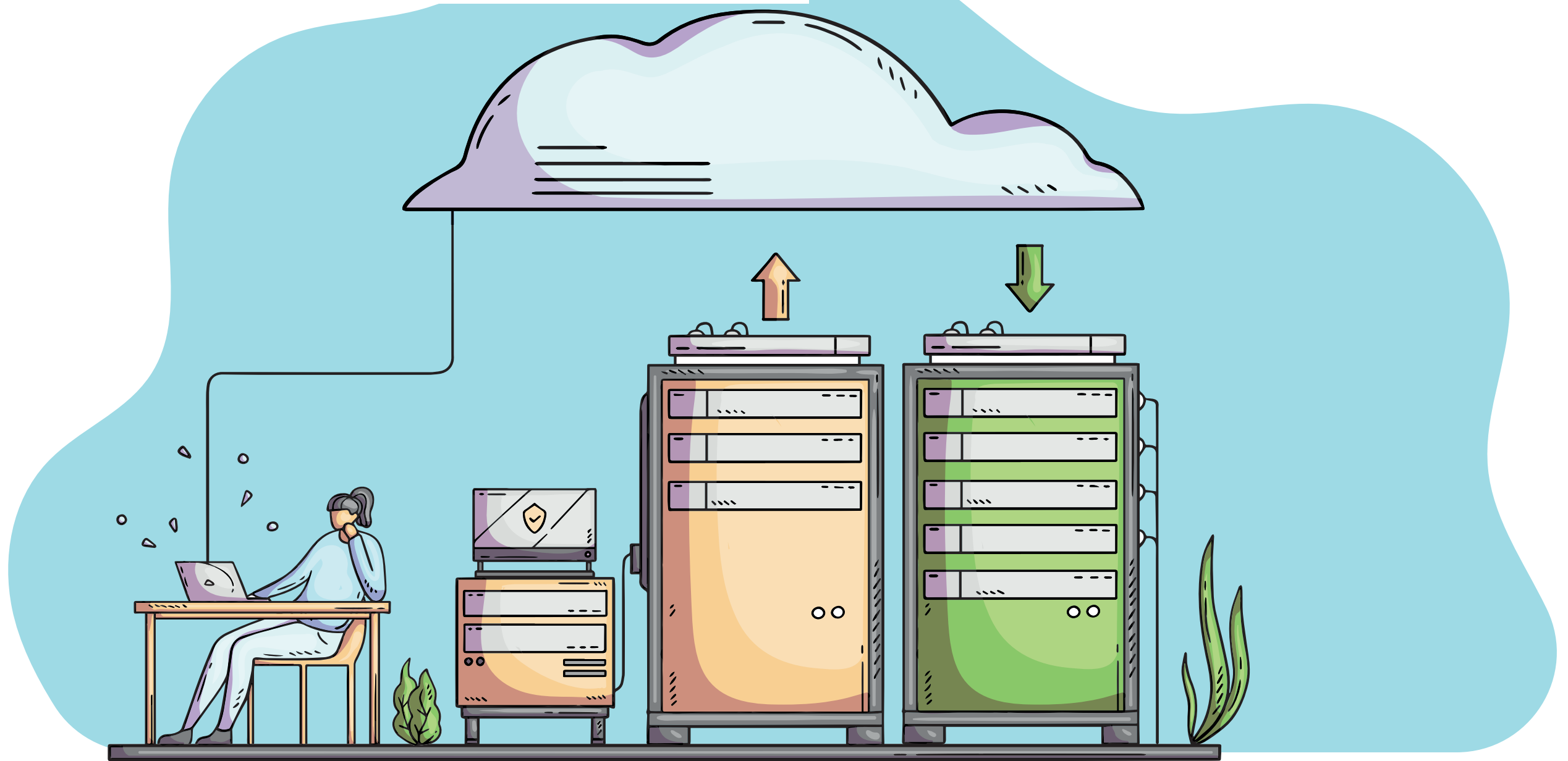
Migrating data between different database environments is something all of us as Database Administrators have encountered in our career path as IT professionals. Whether our involvement was from design to implementation or just participating as an end user is immaterial – we all have participated and appreciate the planning and attention to details that go into a migration. Ensuring a database migration is smooth and seamless as opposed to becoming a nightmare is only possible by paying extra attention to an enormous number of details. As the old saying goes, the devil is in the details.

Database Migrations: An Overview

Migrations are often the backbone of advancing our database and application environments to a more powerful and secure digital state. Here are just a few of the more common ones and a short list of some of the tools and techniques we can deploy:

- Migrating database across physical data centers
  - Lift and shift
  - SAN-based replication methods
- Migrating database between different operating systems
  - RMAN Cloning
  - GoldenGate replication
  - Cross-Platform Transportable Tablespaces
- Migrating to a higher database release
  - In-place or out-of-place upgrades
- Migrating across technologies – GoldenGate replication / RMAN Cloning
  - Sybase to Oracle
  - SQL Server to Oracle
- Migrating from on-premises datacenter to a public cloud like Oracle Cloud Infrastructure (OCI) or Microsoft Azure

The last decade or two has seen a huge amount of digital transformation; all of us have worked tirelessly to move from a variety of OS's – for example, from AIX/Solaris to Linux. We have been planning and designing data center migrations to ensure better resiliency and disaster recovery. RMAN cloning and transportable tablespaces and in some cases GoldenGate have been the go-to methodologies for OS migration.

The most common of all and least complicated are of course the plethora of upgrades we do for the enterprise system's OS and the database itself. This is at least an annual if not quarterly task for most of us working on upgrades, whether those upgrades are in-place or out-of-place. Between the different OS upgrades and periodic patching and the database version upgrades, we as technology subject matter experts are always planning and designing them and implementing them to perform as flawlessly as possible.

Cross-Technology Migrations: Migrating between technologies becomes a more evolved process, as there are several nuances that we need to take care of to ensure data integrity and optimal performance on the target technology at least to match the source technology if not surpass it.

Cloud Migration: The last few years have been dominated by different types of cloud migrations that have catapulted us into a whole new era of digital transformation. With the advent of cloud technology there has been a surge of migrations to the latest technology available in the various public clouds to help companies give themselves a technological facelift.

Is there a one-size-fits-all formula for successful migration?

At this point, it should be painfully obvious that every migration effort is unique! Irrespective of what is being migrated, for it to be successful we need a detailed plan that walks us through all the steps from start to finish, but that is not necessarily enough: We also need to consider how to ensure optimal performance and efficiency on the target technology/destination. Wouldn't it be great to have a framework we can rely on regardless of the underlying technologies, including some basic steps that we can go through to achieve a smooth transition? Let us review a few basics and build this framework by answering these questions. We will use a Sybase to Oracle (S2O) migration as our example:

- What encompasses a Sybase to Oracle (S2O) migration?
- What key elements/issues do we envision we will likely encounter *during* the migration?
- What *nightmare possibilities* are we able to identify *during* and *after* the migration?
- How do we guarantee *enhanced performance* at the destination *after* the migration?

The Basics of Any Migration – S2O included
- Migration of Data Objects
- Data Migration and Validation
- Maintaining Data Relationship Integrity

Migrating databases is all about the data objects, data integrity and the relationships between the objects that keeps the database coherent. To migrate them from any one technology to another we need to have a good grasp on the integrity relationships and especially the datatypes and how they map between the two technologies.

Sybase to Oracle Migration: A Case Study
Let us use a case study to go through the specific issues of Sybase to Oracle migration and data mapping to ensure a smooth migration and resulting in increased performance.

The project encompassed the migration of 15 applications running on 35 databases. The design of the databases was intertwined and convoluted at best. Legacy design and prior migrations had created lot of duplicates in the design. The

same data was being collected by multiple applications in different contexts and then ETL jobs were run nightly to sync all the different databases to ensure data integrity was maintained across the different systems.

Issues at a Glance: Multiple Systems / Applications
- A common issue with legacy apps: Multiple databases which really needed to be aggregated
- Duplicated functionality in application
- Data issues arising from duplicated objects
- Triggers used for data updates
- Convoluted entity relationships
- Complex data aggregation code for reporting across applications
- Multiple scripted jobs to satisfy data validation and nighty sync

Redesigning the database for clarity and increased functionality is one of the main complex tasks that needs to be done at the outset. Creating an ERD (if you don't already have one!) will help with normalizing the data and working on consolidation to ensure we have a good database design going into Oracle. This is vital to improve query efficiency and performance. It's also a good idea to evaluate and if necessary to redesign triggers and data flow to enable updates efficiently via procedures.

Establishing a single data flow through the application is essential to remove duplicated functionality. It also helps establish data integrity. Once the database design is finalized, we can create the target and start planning the data migration.

Here are the key goals identified for the migration process:

- Database consolidation
- Data integrity
- Ease of maintenance of database and application
- Ease of operations
- Application consolidation
- Ease of enhancements
- Security and hardening
- Audit requirements

I also identified these steps to ensure the above goals are successful:

- Migration assessment
  - Data analysis
  - Reverse-engineering data structures (building an ERD is immensely helpful here!)

- Schema code conversion
- Data conversion from Sybase to Oracle
- Extensive testing
- Go-live checklists
- Monitoring for post-migration success, including application connectivity and performance

Finally, we performed extensive application analyses alongside our developers on these crucial activities to ensure overall success:

- Data flow testing
- Replacing Sybase-specific features with Oracle corollaries
- Dynamic SQL
- JDBC connect logic

Data Migration Options
Fortunately, we have several excellent tools that make data migration less painful. Choosing which tool to use will depend on your individual circumstances.

SQL Developer
SQL Developer is an extremely reliable tool for all types of migrations. Setting up SQL Developer and running the built-in migration toolset helps to understand the mapping of objects between the source and target (Sybase and Oracle). See **Fig. 1** and **Fig. 2** for examples of how simple it is to set up this toolset.
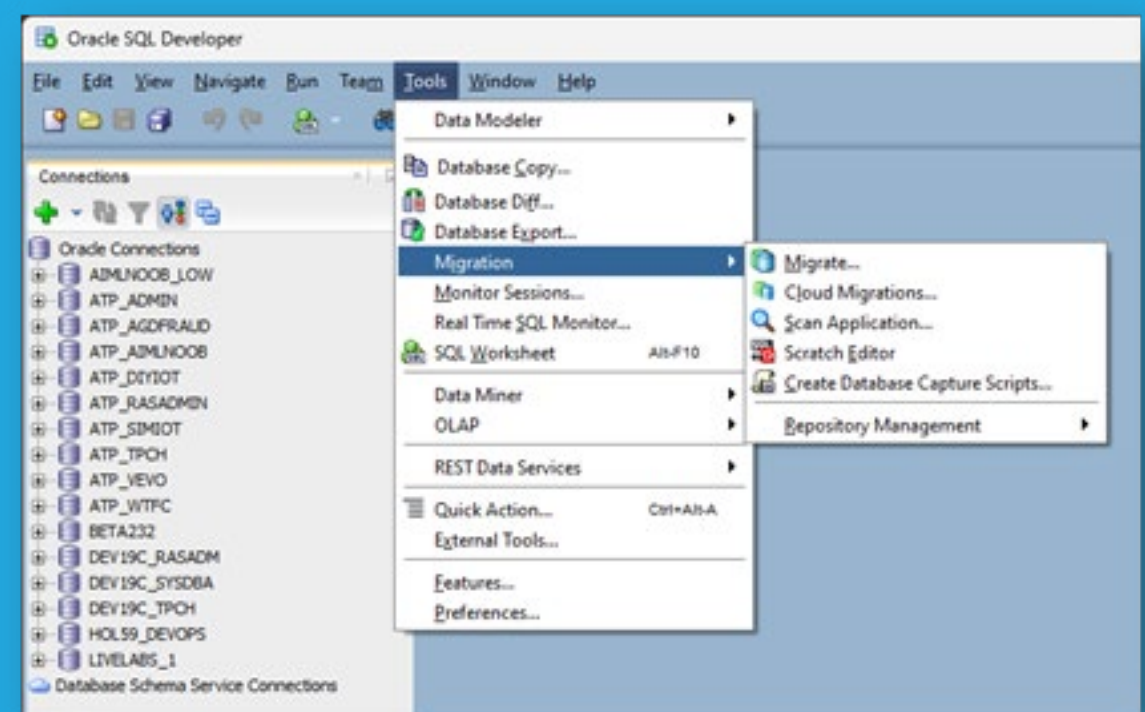
Fig. 1: SQL Developer– Finding the Migration Toolset

Fig. 2: SQL Developer – Setting up Migration Toolset

Often, SQL Developer does a good job of migrating the data; however, there are a few datatypes that may have issues and we have to look at other tool options like GoldenGate. LOBs and BLOBs can cause issues and these may need to be migrated piecemeal using other tools.

GoldenGate

One of my favorite tools for migration is Oracle GoldenGate. It is extremely versatile and robust due to the numerous functionalities it offers. GoldenGate replication can be set up between any two environments; differing OS and database technologies between source and target environments therefore become less relevant, which is what makes its ability to map multiple databases into a single database versatile for all cross-technology migration.

We can use GoldenGate for migration from Sybase, SQL Server, or even Postgres to Oracle. When we set up the initial sync from source to target, we should enable change synchronization as this feature helps for continuous flow of replication between source and target and helps with cut-over during the final migration.

Sybase to Oracle Data Migration: Some Notable Pain Points and Resolutions

As you might guess, my suggestions above derive from my own personal experiences. Here are just a few examples my team and I recently encountered during a Sybase and Oracle

conversion project and how we finally resolved them.

Trailing Blanks

Data type: Sybase CHAR/VARCHAR conversion.

Issue: Queries in Sybase return data while same queries returned no data in Oracle.

Cause: Sybase CHAR/VARCHAR allows leading and trailing blanks in the field.
Sybase ignores leading and trailing blanks, so results sets are matched and returned. Oracle does not ignore these blanks, so resulting data is a mismatch and queries return no data.

Resolution: Oracle enforces blanks, so creating a special script to remove leading & trailing blanks helps retain data validity.

NOT NULL Constraints

Issue: Tables with NULL values having NOT NULL definition.

Cause: Sybase NOT NULL definition can be added to a table with NULL values; the rule is not enforced to previously housed data.
SQL Developer converts these as NOT NULL constraints resulting in data load failures.

Resolution: Redefine table as needed (NULL/NOT NULL), cleanse the source data, and reload it in Oracle.

Data Mapping Issues

Any migration across database technologies will encounter datatype mapping issues among others, but being prepared by doing adequate research on the type of data we have in the source and how it will be accommodated by the target helps us be prepared to resolve the issues we may encounter. When migrating data from Sybase to Oracle, here are just a few of the most common issues I've encountered. I've broken them down between those related to data transfer vs. database objects. Going through all of these issues and their resolution might seem boring, but they're absolutely crucial to identify at the outset to ensure a successful migration.

Datatype Issues

It's not surprising that the largest number issues revolve around how Sybase and Oracle store and handle data. Here are some of the most glaring problems our team discovered during our migration.

Datatype conversions are the most crucial part of migration. The variety of data types we have and the comparable counterparts in Oracle like Bit to Number, Int to Number, Text to CLOB, Image to BLOB, datetime. The reason we run into issues is predominantly due to the way the data is stored by Sybase Versus Oracle.

For instance, consider the TIMESTAMP datatype. Oracle stores at the nanosecond level, so it may not match what how Sybase has stored it within its tables. Another datatype issue that can be puzzling to resolve are Sybase's image datatypes versus Oracle's various LOB (CLOB, BLOB, NCLOB) datatypes.

The best way to resolve these conversion issues is by doing at least a few due diligence activities like understanding the differences between Sybase and Oracle datatypes. Conversions to Oracle's DATE and TIMESTAMP datatypes are not necessarily trivial, and it's important the conversions have occurred properly, especially when those columns are used in selection criteria by applications and queries. Likewise, loading of image data types may even need to be performed via individual files to ensure data integrity is preserved.

Object Issues
"Scratch" Tables vs. Temporary Tables: Some older Sybase-based applications made frequent use of "scratch" tables. Be aware that these are definitely not the same as Oracle's temporary tables! In fact, they're more like Oracle PL/SQL collections. However, there's no easy way to simply translate Sybase logic around scratch table usage to PL/SQL collections, so it will likely consume a lot of time you may not have anticipated.

Most of all, don't just blindly shift the usage of Sybase scratch tables to Oracle temporary tables without understanding exactly how Oracle handles redo log entries, or you may see enormous use of redo logs when you least expect it.

Summary: Yes, The Devil Is Indeed in the Details.
Hopefully my experiences have convinced you that there's incredible value in meticulously planning your migration process using a framework so that you achieve a smooth and seamless cross-technology migration. If this article proves anything, it's that the old saying is indeed true: *the devil is in the details.*

About Valli Ramanathan

Valli is an Oracle ACE Associate and an Oracle Certified Cloud Architect Professional who has worked on several types of database migrations. Over the years she has played a key role in various enterprise organizations to help with data center migrations and cross database technology migrations. Having successfully performed different types of migrations, Valli likes to share her knowledge with the community by speaking about them at conferences and by writing articles. As a consultant, bringing the most efficient and cost-effective techniques to the table is a key skill that she values more than anything else. Valli has 25+ years of Oracle database administration expertise and is a well-known speaker at Oracle conferences in the USA.

Rethink your Data Management: Oracle Open Approach to Data Lakes

Alexey Filanovskiy

Dive into the world of modern data management with Oracle's Autonomous Data Warehouse (ADW). This article explores the open and innovative Data Lake architecture of ADW, giving a purely business-driven choice of a database-centric and an object store-centric model without the need of making financial-driven sacrifices with Oracle's eCPU model.

In the rapidly evolving landscape of data management, Oracle's Autonomous Data Warehouse (ADW) stands at the forefront, offering innovative solutions for data lakes. This article dives into the open architecture approach of Oracle ADW, exploring the unique strengths and applications of both database-centric and object store-centric models jointly without any sacrifices. We will also highlight the transformative eCPU model, showcasing how Oracle is redefining the economics of data storage and management, making it a game-changer for organizations seeking cost-effective, high-performance data solutions.
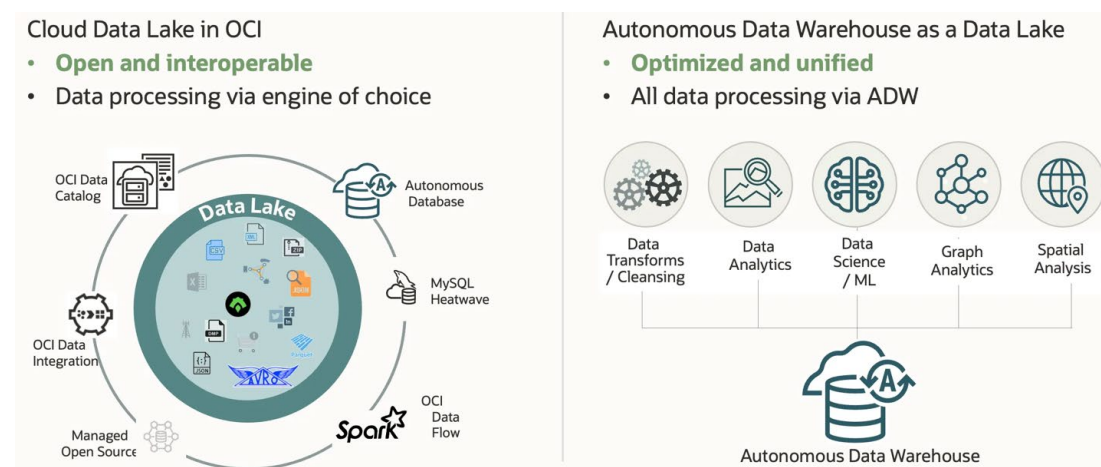
Fig.: Different approaches to Data Lakes on Autonomous Database.

Understanding Oracle's Open Architecture, offering both Database-Centric vs. Object Store-Centric models

In the realm of data management, two prominent architectures emerge: the database-centric and the object store-centric. The database-centric model, known for its superior performance and robust security features, is designed for organizations prioritizing fast, efficient data processing and solid on data governance. On the other hand, the object store-centric approach offers flexibility. It facilitates instant data access, bypassing traditional ETL processes and supports diverse analytical engines, making it ideal for environments demanding adaptability and multi-cloud integration.

Oracle's Game-Changing eCPU Model: A New Cost Paradigm

Oracle's introduction of the eCPU model marks a turning point in data storage economics. Historically, the cost has been an important factor in choosing data architectures, with object stores being the more budget-friendly option. The eCPU model disrupts this status quo by aligning the cost of Oracle's database-centric solutions with those of object stores. This innovation empowers organizations to leverage the high performance and security of database-centric architectures without the traditional cost compromise, catalyzing a shift in the data management landscape.

Embrace the Best of Both Worlds: Oracle's Flexible Data Lake Strategy

Oracle ADW's flexible, open data lake strategy starts a new era in data management. By allowing organizations to blend the database-centric and object store-centric models, Oracle ADW provides a customized solution that embraces the strengths of both. This strategy includes supporting a wide range of file formats, ensuring compatibility with multiple cloud platforms, and planning for future integrations like the Iceberg table format and Delta Sharing Protocol. All these formats are available in multiple clouds, such as AWS, Azure and GCP. Such flexibility ensures that Oracle ADW adapts to varied organizational needs, offering a comprehensive, secure, and efficient data management solution.

About Alexey Filanovskiy

Alexey has been working in the Data Warehouse and Data Lake areas for 15 years, 11 years of those at Oracle. Alexey started his career as a Data Warehouse developer. He has hands-on experience with technologies such as Oracle Exadata, Teradata, Greenplum, and Apache Hadoop. In 2012, Alexey joined Oracle as a Sales Consultant and evangelized Big Data technologies such as Hadoop and NoSQL. After he joined the development team in 2015, his primary focus was outbound product management (bringing best practices from HQ to the field and bringing back customer feedback to incorporate it into Oracle's product roadmap). Starting in 2019, Alexey focuses on inbound product management for several Cloud Services, including Big Data Service and Lakehouse features of Autonomous Database.

Carsten Czarski

Oracle Application Express (Part 13)
2023, the second: APEX 23.2

Workflows, Working Copies and much more ...

WORKFLOW

Introduction

APEX 23.2 is available since November 2023. As always, the new release introduces some marquee features, like *Working Copies* and *declarative Workflows*, as well as several other new functionality and enhancements. These include new page items, improved components, new REST connector types, application builder improvements and much more.

Working Copies

Many APEX developers work on small applications in rather small environments – without separate APEX instances for development, test, and production. When changes are done to an application, they're immediately visible to end users. This medal has two sides: APEX developers can be blazingly fast in applying quick fixes and improvements to an application. The backside is that it's as easy to break the application with just a "small change".

The new **Working Copies** feature in APEX 23.2 is targeted to those people. APEX allows to create a *copy of an application* in which such changes can be applied – the original application remains unchanged. Developers can run the working copy, test the changes, verify that only intended changes were done and finally merge the changes back into the original application.

Working Copy is not intended as a replacement for version control like Git or SVN, or a replacement for dedicated development and testing systems. It's rather intended as a solution for those APEX developers who don't have access to such systems, who only work within their APEX workspace. Such developers can now get a great deal of additional safety before applying changes to their applications.

Using the new Working Copy feature is super-simple. To create a new working copy for an application, simply click the **Create Working Copy** link within the **Working Copy Tasks** section to the right. A dialog opens where a name and a description for the working copy can be provided **(Fig. 1)**.

The working copy is technically just a copy of the application – **Fig. 2** shows that the working copy has its own application ID. However, APEX maintains the relationship between the original application and the working copy, and provides compare and merge options. Working Copies can only be created on top of applications, a working copy "from a working copy" is not allowed.

The working copy can be changed and tested like a normal application. Developers can now change pages and shared components. The working copy can be started as always, using the **Run Application** button.

With the **Compare Changes** option within the working copy menu, developers can get an overview of all differences between the original application and the working copy. **Fig. 3** shows the results of a **Compare Working Copy** operation. The dialog shows a summary of changes per component (page, shared component); a detailed diff is brought up by clicking the **Diff** icon.
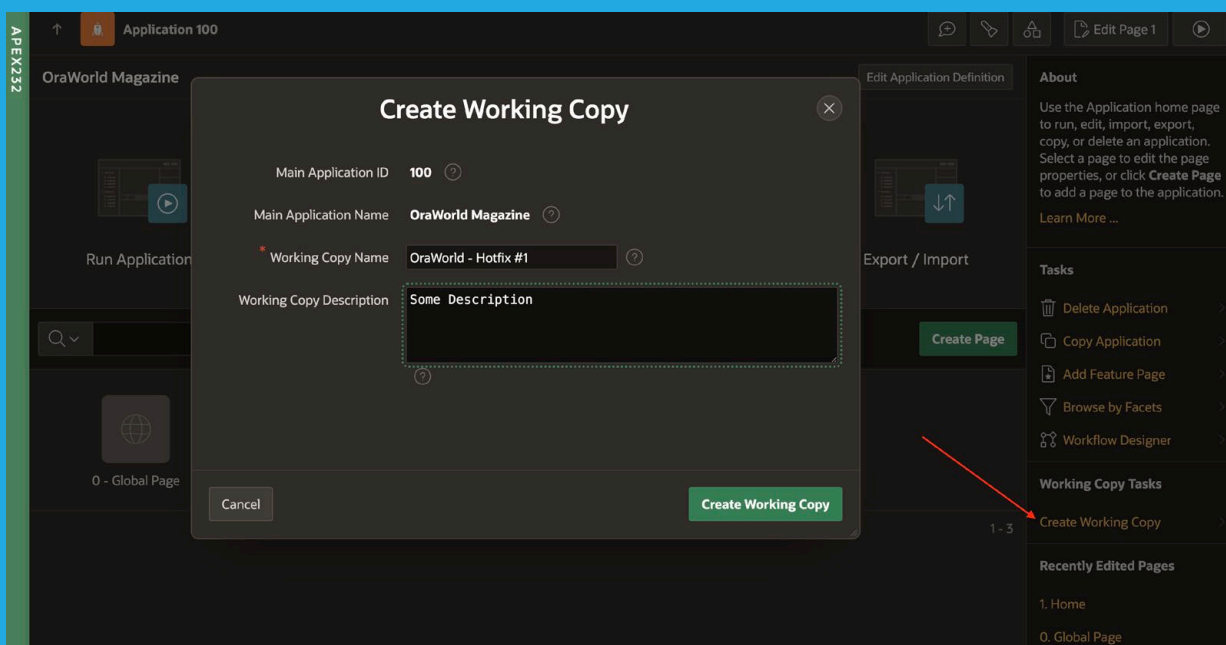
Fig. 1: Create a new Working Copy for an existing application.
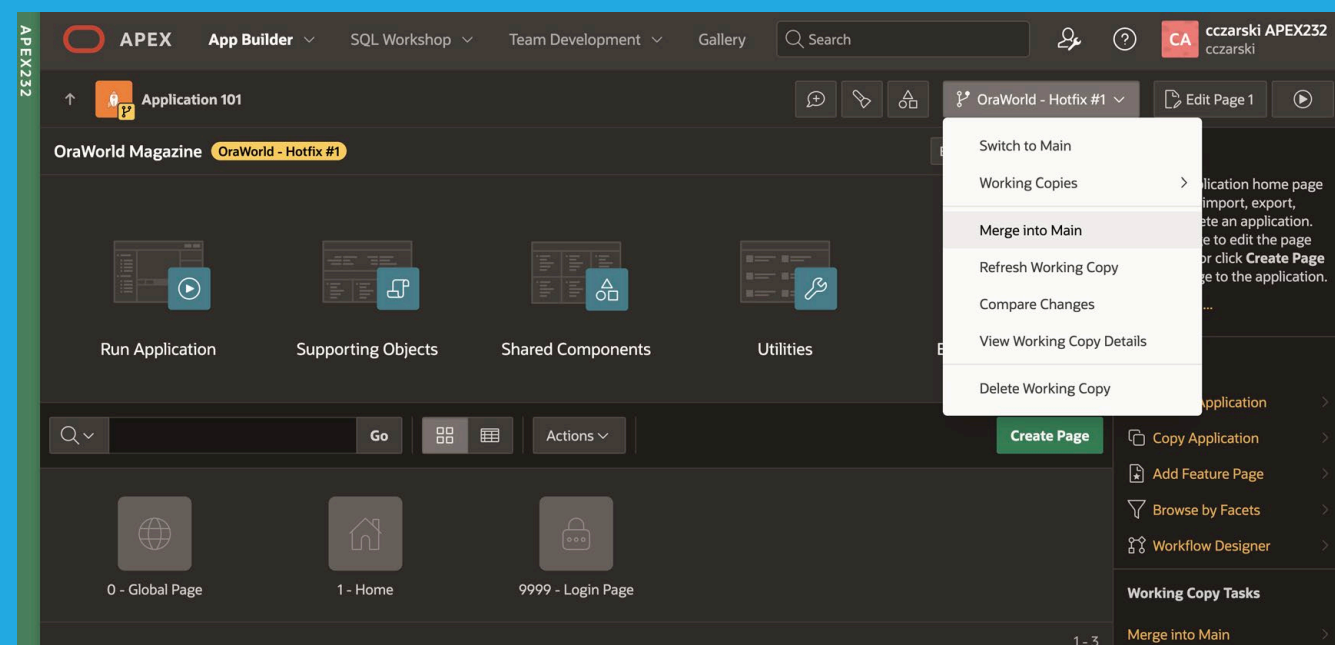
Fig. 2: A working copy of application 100 has been created.

Once changes are verified, developers can merge these into the original application using the **Merge Working Copy** function. APEX allows to choose whether the original application is to be backed up before and whether the working copy should be kept or removed. After merge, the cycle is completed: the developer was able to apply a change in a protected environment (the working copy), to verify it and to finally merge it.

A few limitations apply to the Working Copies feature; an overview is accessible from the **Compare Working Copy** dialog (**Fig. 4**).

Workflows

APEX 23.2 introduces the new declarative **Workflow** functionality, which enables developers to automate business processes using a visual workflow designer. The workflow engine is deeply integrated with APEX: Workflow activities are like page processes, and known page processes like **Execute Code**, **Send Email**, or **Invoke API** are natively available as **Workflow Activities**. Process Type Plug-Ins can be enabled for Workflow activities so that these are as extensible as APEX page processes are.

A Workflow is created and maintained in **Shared Components**, **Workflows** (**Fig. 5**). The workflow designer is very similar to page designer, it should be immediately familiar to APEX developers.

Workflows maintain their own state using **Workflow Variables** and are thus independent from APEX sessions, application, or page items. **Parameters** are used to pass information into a workflow when it's started.

A workflow can be started from an APEX page using the **Workflow** page process (**Fig. 6**), or from anywhere else, with the **APEX_WORKFLOW** package. The Workflow page process type also allows to *terminate*, *suspend*, *resume* or to *retry* a workflow instance.

When the **Workflow** page process, with the **Start** type, is executed, a new instance of the workflow is initialized and started. The ID of that workflow instance can be returned into a page item using the **Workflow ID Item** attribute. This workflow instance ID can later be used to get status information or to perform additional actions on that very workflow instance.

To provide an overview of existing workflow instances, and their states, to application users, a **Workflow Console** is
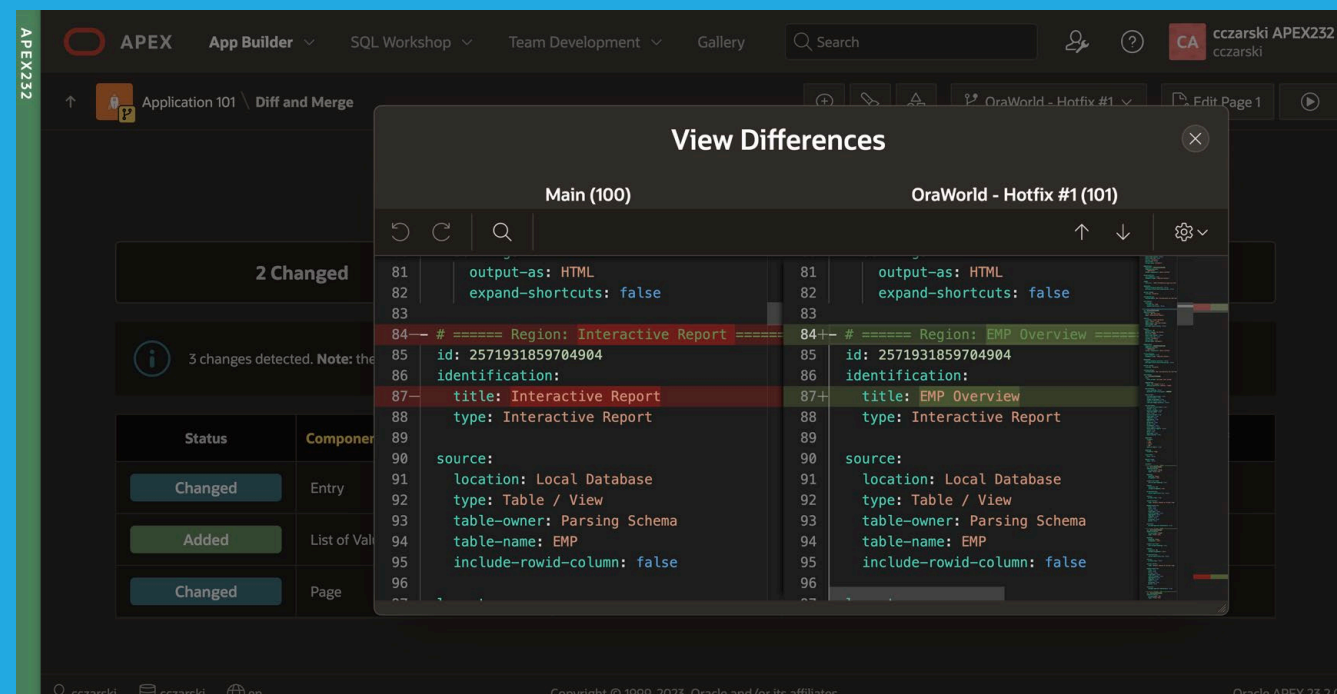
Fig. 3: Compare Working Copy shows detailed differences.

About Working Copies

Tips
- Exporting an application with working copies does *not* include them
  - Export working copies separately as needed
  - Use *Original IDs* to maintain their link with main

- *Unlink* a working copy from main by using the *Copy Application* task
- Delete any working copies *before* deleting an application
- Developers can use distinct working copies. To avoid overwrites or errors on merge:
  - Reserve page ranges for new pages each developer will create
  - Agree on shared component names each developer will add

Unsupported Component Types
These component types are not yet supported for *automatic* merging: Translations, Themes, Theme Templates, Supporting Objects, Application Properties, and Workspace-level components (for example, App Groups, Credentials, Remote Servers). Developers can merge the working copy without deleting it, then manually apply the additions or changes to these component types.

Limitations
Working copies have the following limitations:

- You cannot create a working copy of a working copy
- Deleted working copy pages or shared components must be *manually* deleted in main
- Components listed as *Missing* may have been removed in the working copy or may only exist in main

Fig. 4: Working Copy Limitations
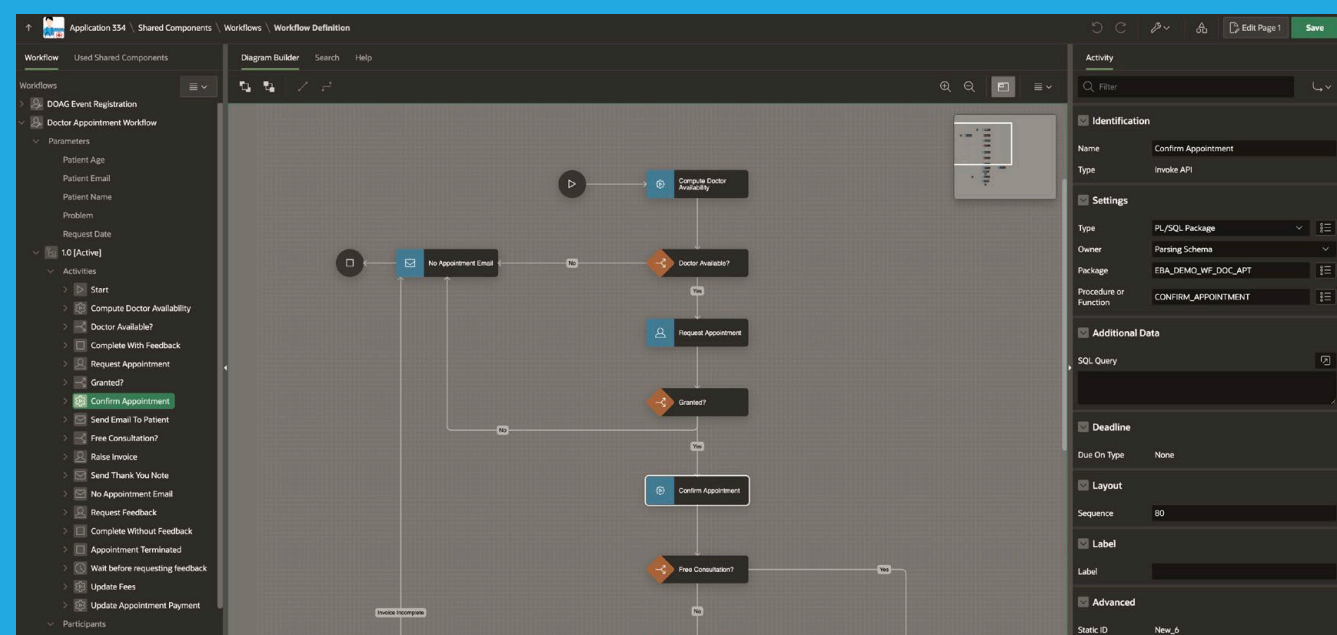
Fig. 5: Visual Workflow Designer in APEX 23.2
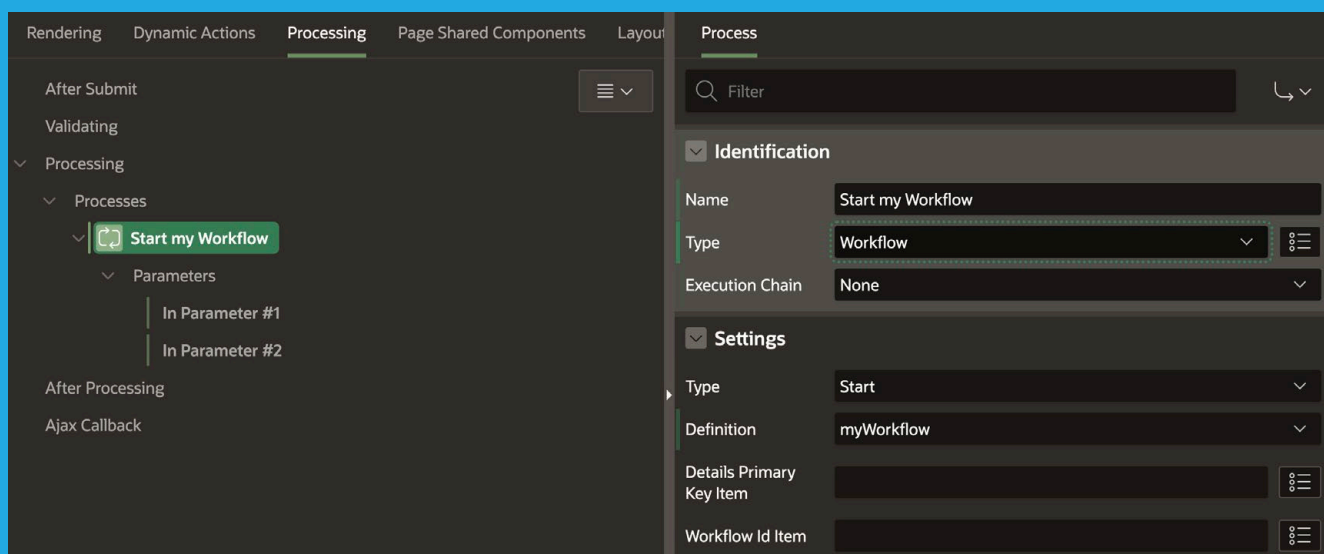
Fig. 6: Adding a page process to start a workflow instance in Page Designer.
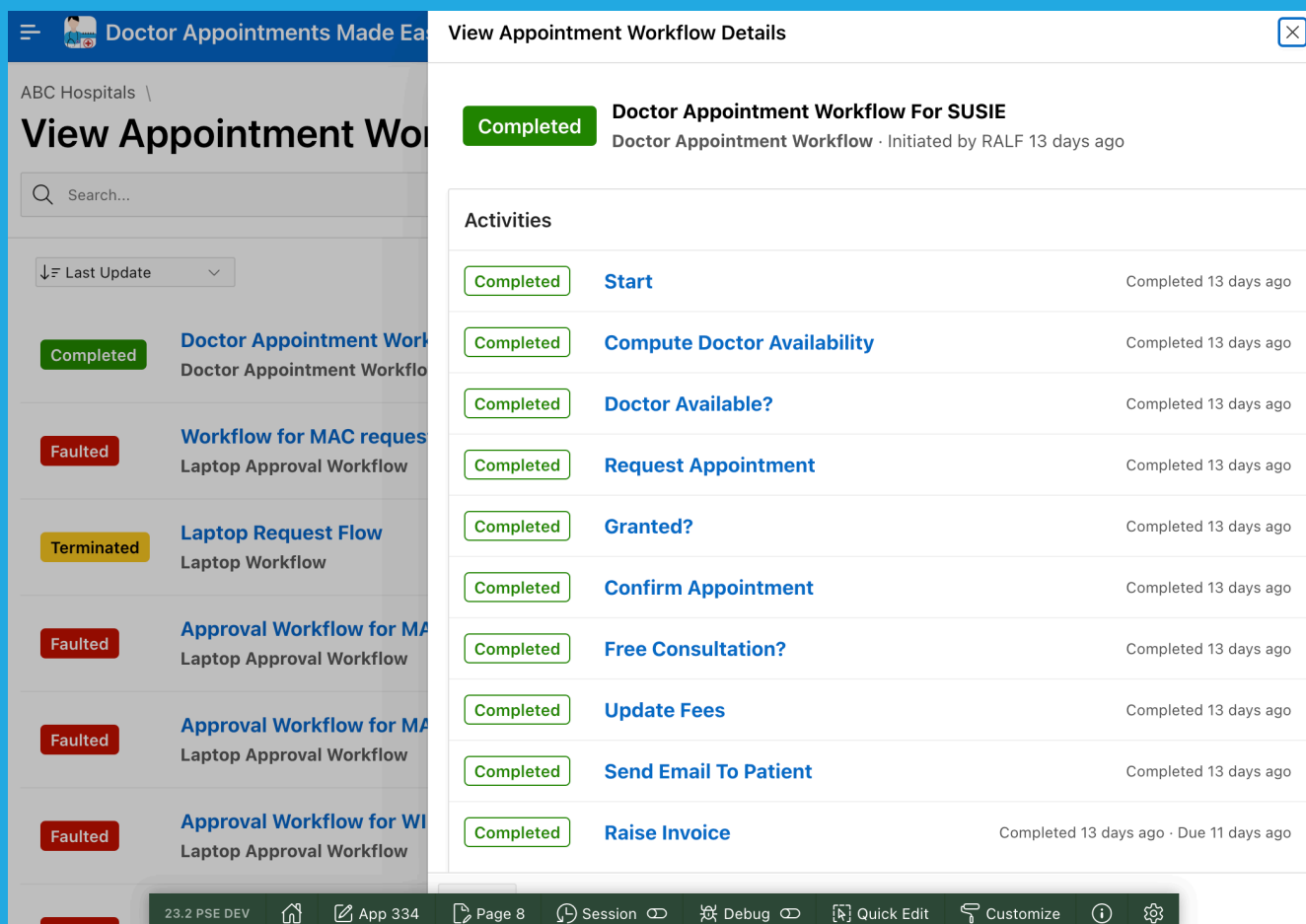
Fig. 7: The workflow console pages provide an overview on Workflow Instances and state
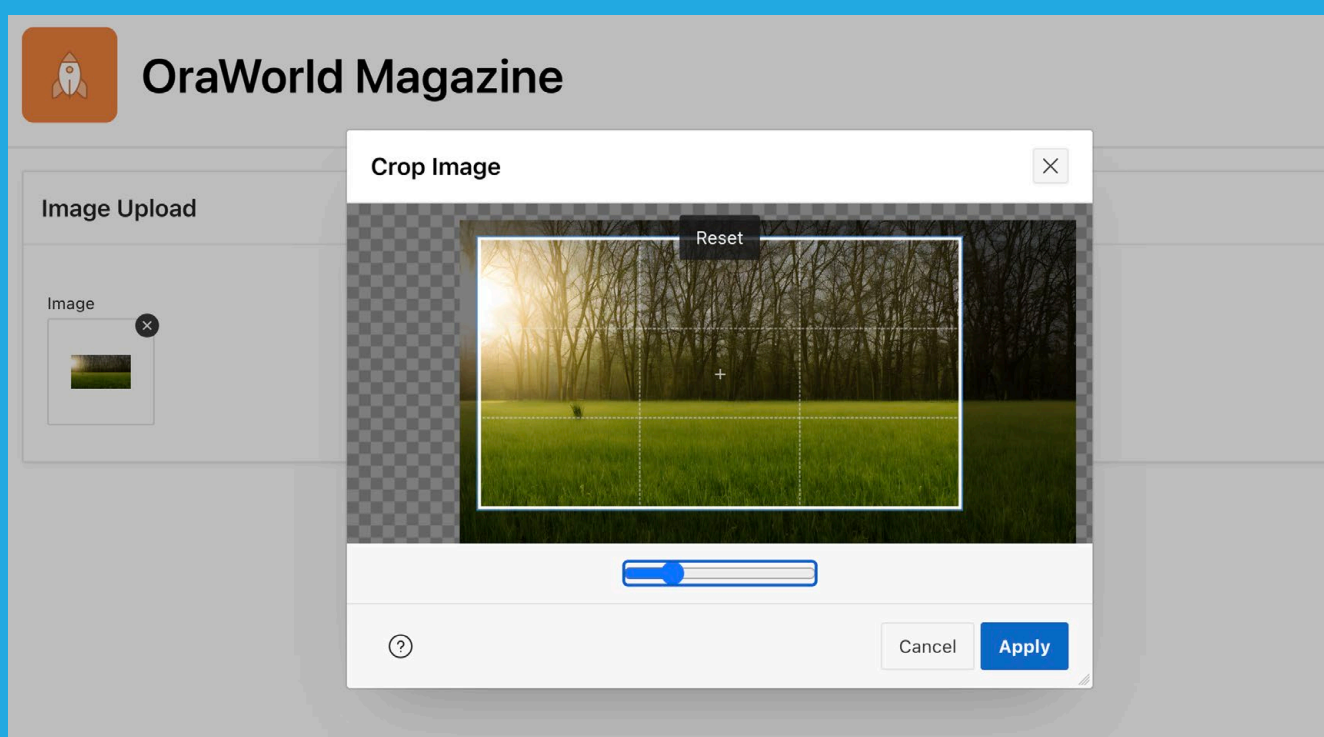
Fig. 8: The new Image Upload item has built-in processing capabilities.

typically being used. The APEX create page wizard allows adding such a **Workflow Console** to an application, which consists of two pages: the *overview page* is based on Smart Filters and shows a list of all workflow instances the current user can access. The *detail page* allows to drill into a given workflow instance. **Fig. 7** shows how the workflow console looks like.

New Page Items and Enhancements

APEX 23.2 adds quite some enhancements to Page Items, Faceted Search and Smart Filters. First, three new page items are introduced: **Image Upload**, **Combobox** and **QR Code Generator**.

The new **Image Upload** item makes it easy to crop, resize, and upload images. Resize and crop operations are done in the browser on the client-side (**Fig. 8**). On mobile devices, an image can also be captured from the camera.

The QR code item allows to generate QR codes directly from session state values. Supported types are **URL**, **phone number**, **geolocation**, **email**, **SMS**, or **plain text**. In addition to the page item, APEX 23.2 also provides the new **APEX_BARCODE** PL/SQL package which allows to generate QR codes and barcodes from within PL/SQL – as PNG or SVG files. In addition to the page item capabilities, APEX_BARCODE also supports EAN8 and CODE128 barcodes.

The new Combobox item provides a modern and state-of-the art UI for selecting one or many values from an existing list of values. A special characteristic is the handling of *manually entered values*, i.e., values which do not exist in the provided List of Values.

Old page item types like the Popup LOV "mixed" these manually entered values with the values from the LOV, which made it hard for the developer to distinguish whether a value was chosen from the list or manually entered.

The new Combobox allows to specify a "sidecar" item, which takes up all manually entered values – developers can clearly distinguish these from other values in later processing. **Fig. 9** shows an example of the Combobox item, highlighting the manual entries item.

Besides that, the Combobox item supports template directives for versatile display of list of value entries, client-side, and server-side search, as well as caching.

APEX 23.2 also contains enhancements to Faceted Search and
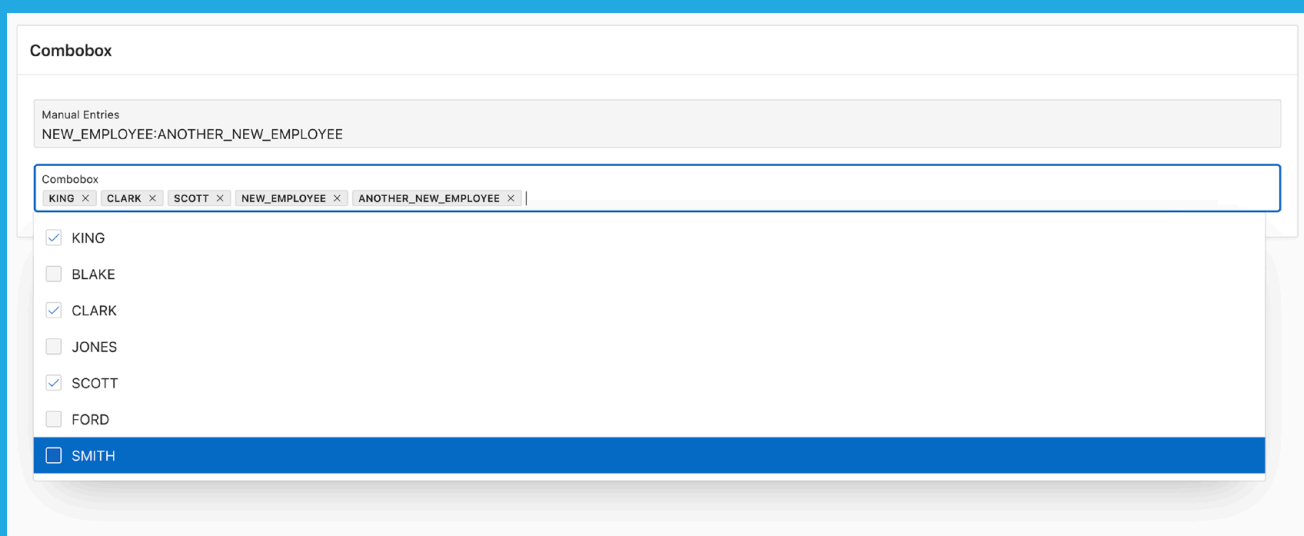
Fig. 9: APEX 23.2 Combobox: manually entered values are maintained in an extra item.
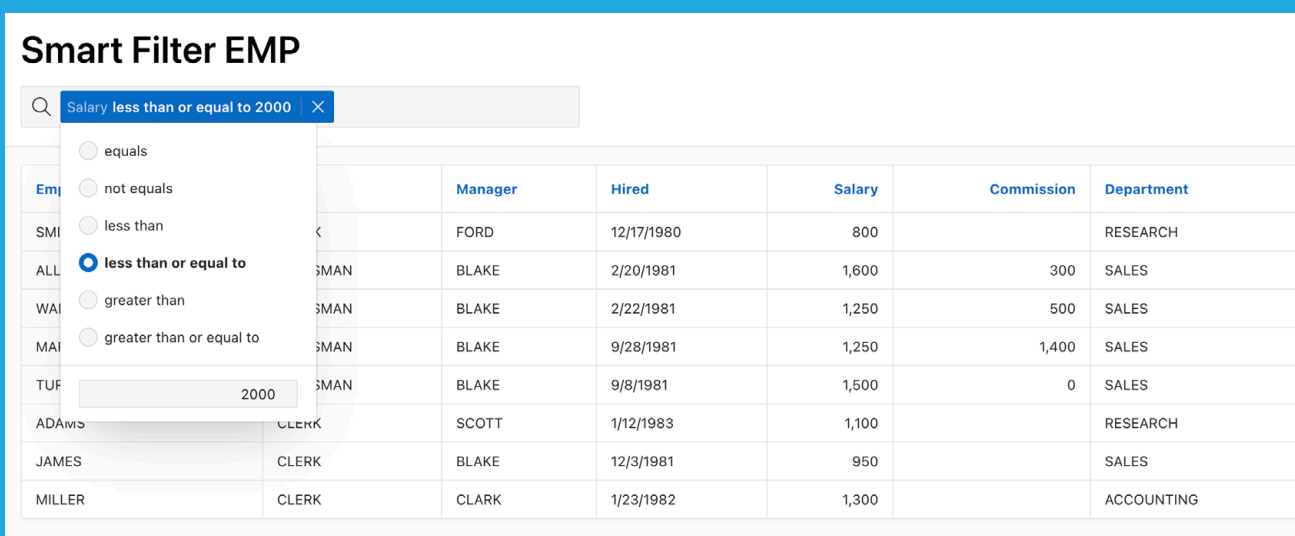
Fig. 10: End Users can choose the filter operator for the Input Field Filter or Facet.
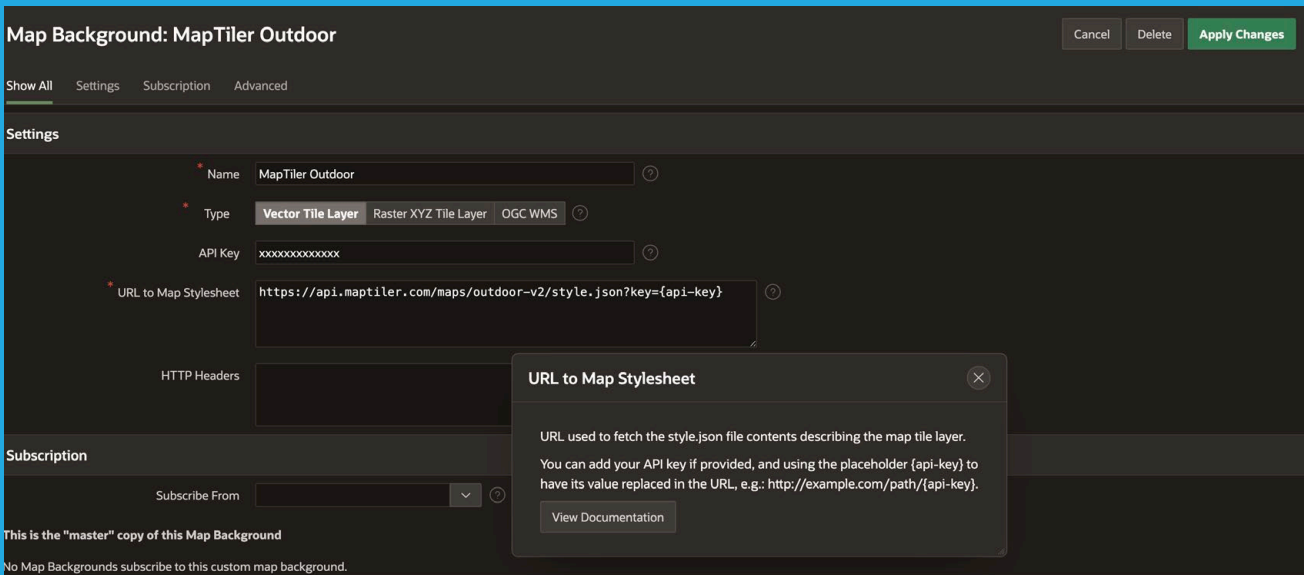
Fig. 11: Configuring a custom Map Background.
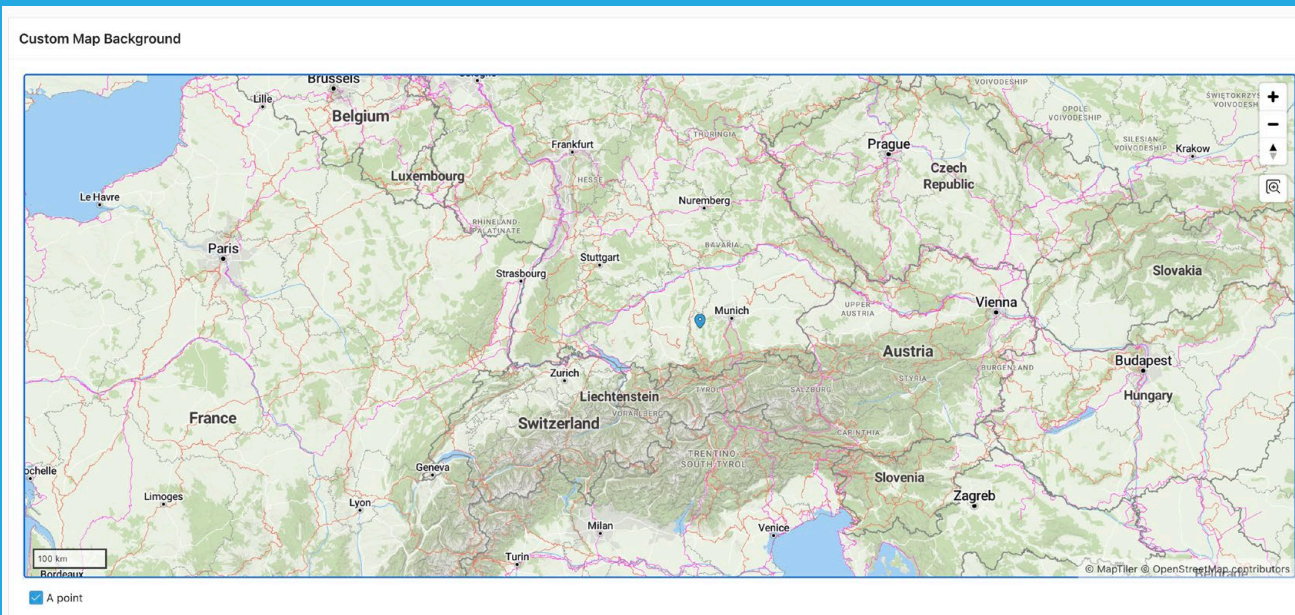
Fig. 12: Custom Map Backgrounds in Action.

Smart Filters. The **Input Field** Facet or Filter type can now be configured to allow end users to determine the filter operator themselves (**Fig. 10**).

Map Backgrounds
The Map Region was introduced with APEX 21.1. So far, only the built-in background maps from the Oracle Map Server (maps.oracle.com) were supported. APEX 23.2 finally allows to use a different background map from a Raster or Vector Tile Server, or from an OGC WMS (Web Map Service) endpoint.

Map Backgrounds are configured in Shared Components. **Fig. 11** shows an example for using an external Vector Tile Layer (from maptiler.org) as a custom map background.

The custom map background can then be chosen in Map Region attributes in Page Designer, instead of the default map background. For the example in Fig. 11, the map would look as in **Fig. 12**.

Enhancement to REST Sources
REST Data Sources also experienced enhancements in APEX 23.2: Parameters for REST Sources are now datatype-aware (**Fig. 13**) and are seamlessly converted between page items and the expected parameter format. You can also specify a

Validation Regular Expression which is evaluated by APEX before reaching out to the REST API.

First support to create REST Data Sources from OpenAPI descriptions was already added in APEX 23.1. This release extends OpenAPI support to *REST Catalogs*: A new REST Catalog (in Workspace Utilities) can now be created from an OpenAPI file, in JSON format (**Fig. 14**).

Once the REST Catalog is in place, new REST Data Sources can be created with a "shopping cart" experience – the typical four-step wizard is no longer required (**Fig. 15**).

APEX 23.2 also introduces new and improved *REST Data Source Connector Types* (**Fig. 16**). The new **OData REST Source** connector allows to seamlessly integrate APEX applications with *OData* REST APIs, and supports pagination as well as pushing filters and sorts, e.g., from an Interactive Report, down to the REST API. The OData adapter currently only supports read operations; it cannot write data back to an OData REST API yet.

The existing **Oracle Cloud Applications (SaaS) REST Service** REST Source connector was improved to support server-side filtering, ordering and to write data from APEX (from a form or an Interactive Grid) back to the Fusion SaaS REST API.
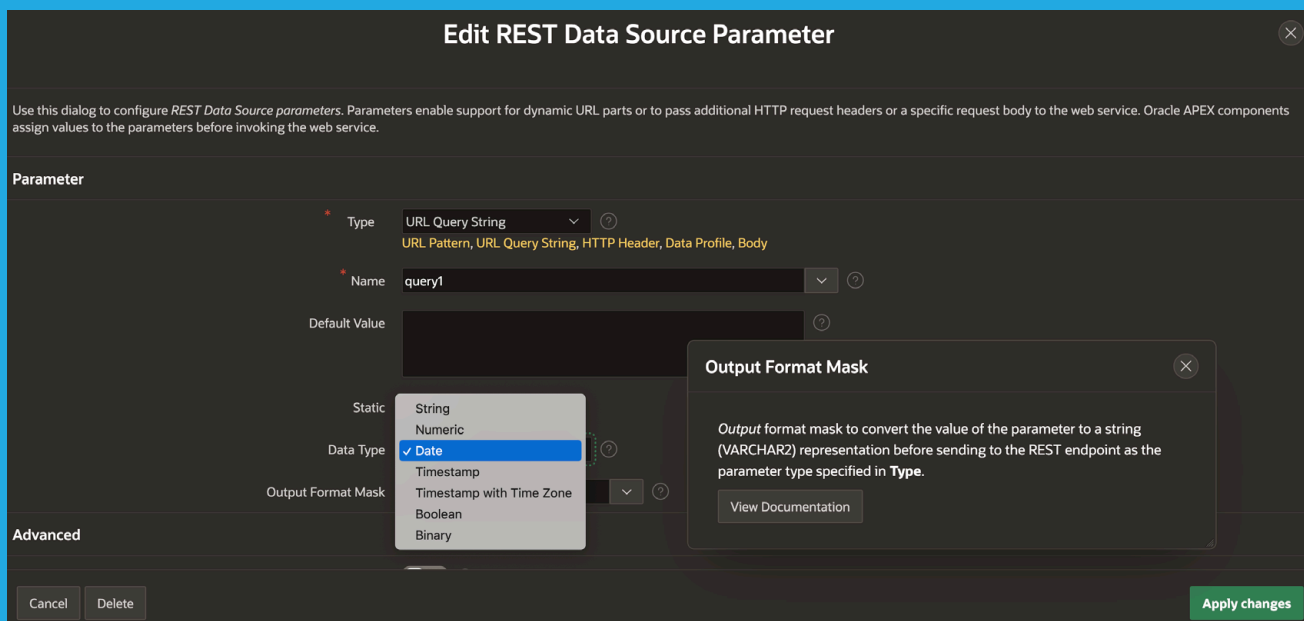
Fig. 13: REST Data Source Parameters are now datatype-aware.
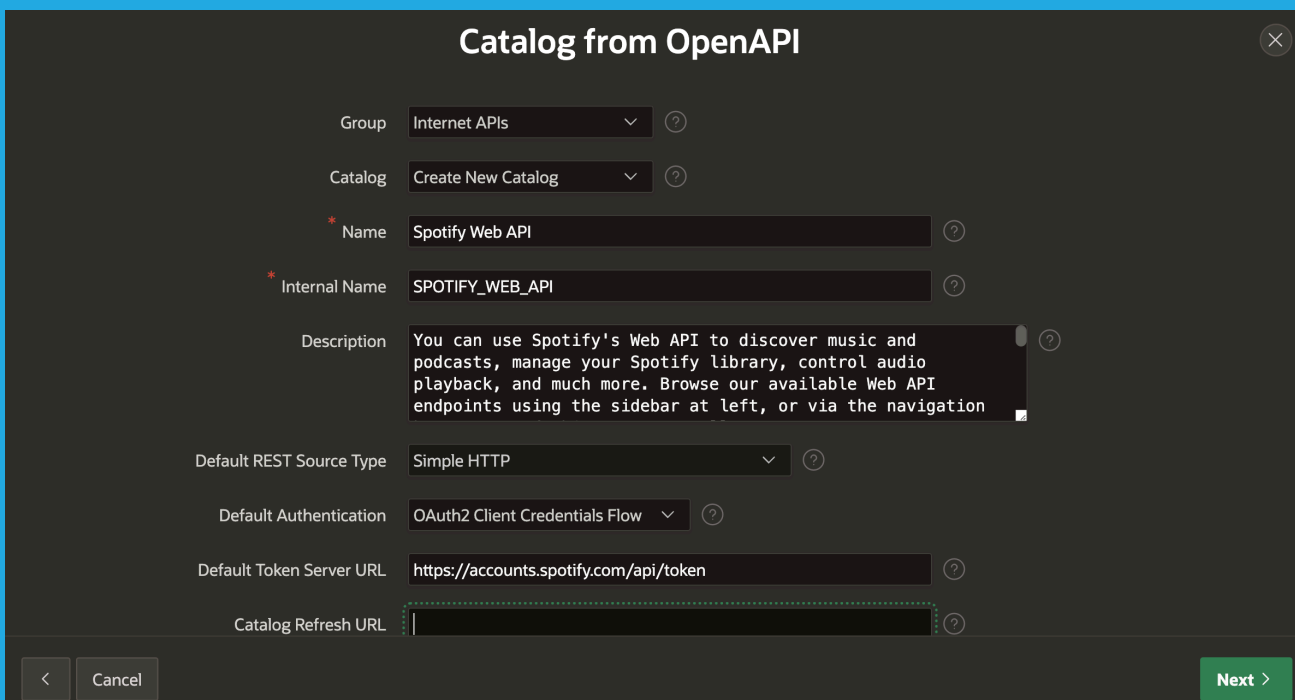
Fig. 14: Creating a REST Source Catalog from an OpenAPI JSON file.
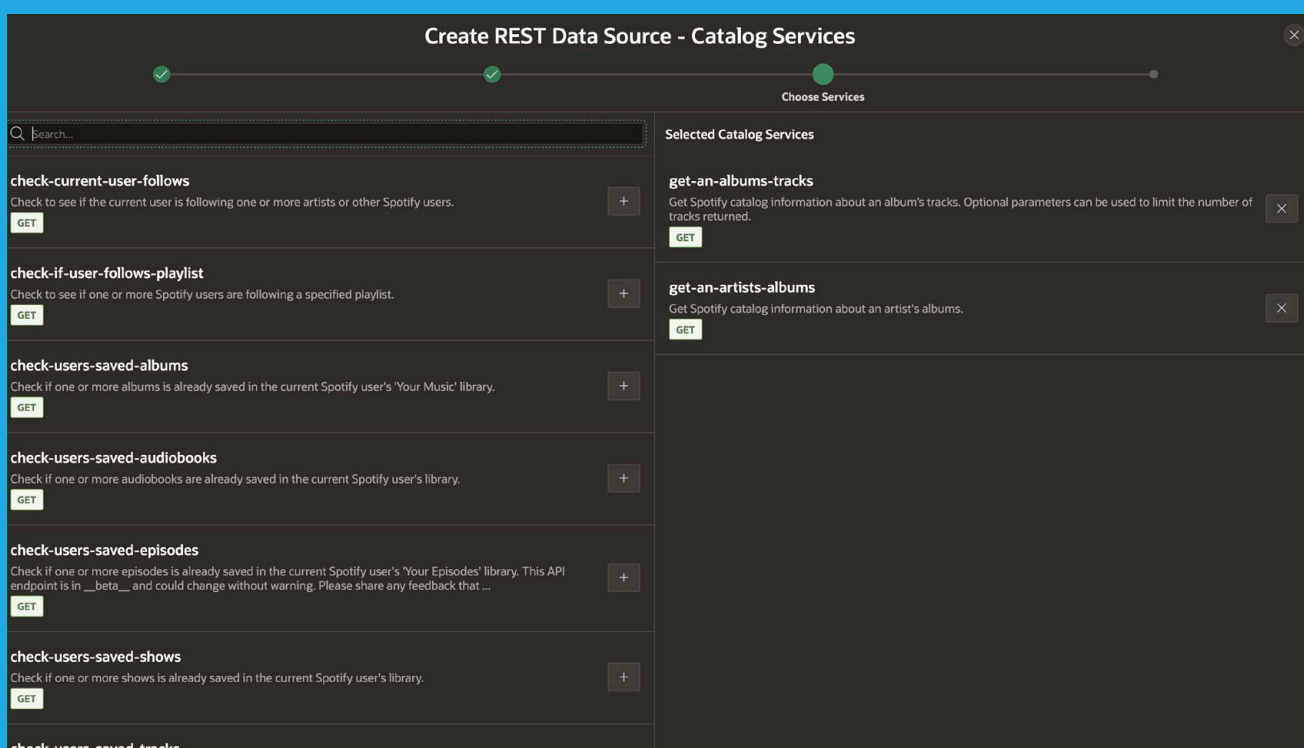
Fig. 15: Create a REST Data Source from a Catalog
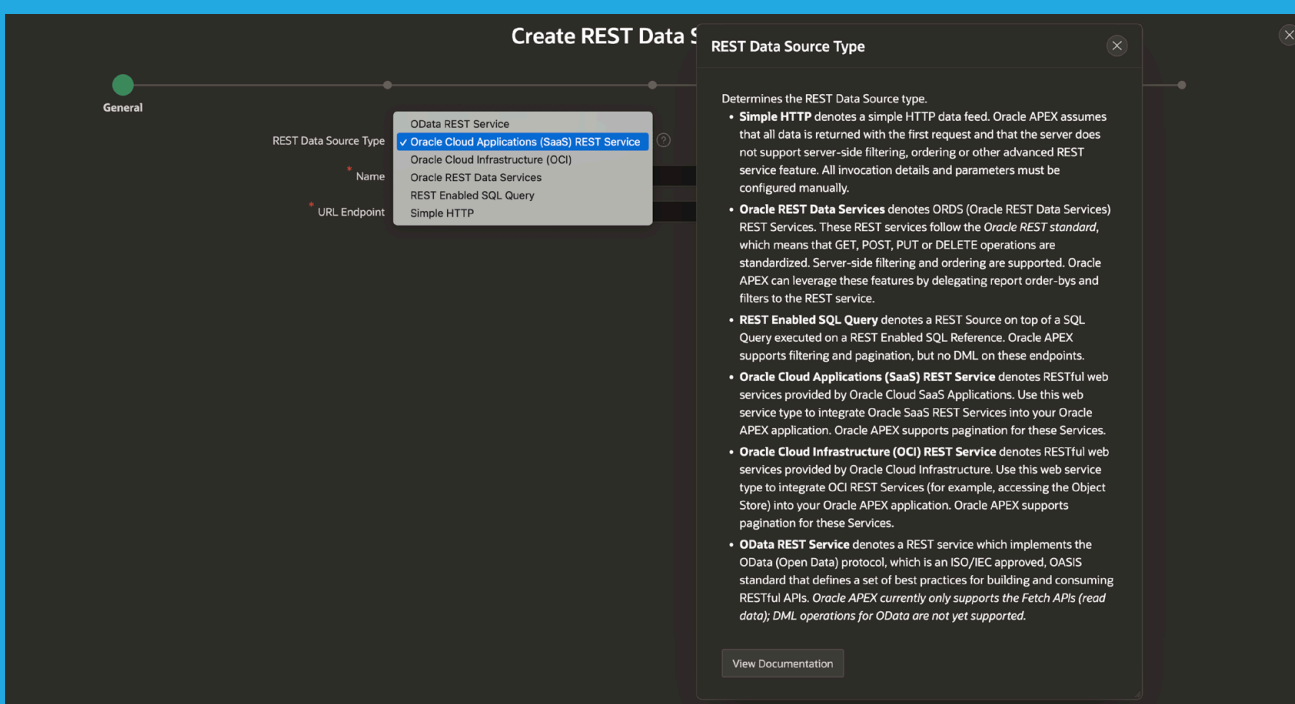
Fig. 16: APEX 23.2 introduces new and improved REST Source adapters.

Other new Features

The *subscription* feature of Shared Components has been improved across the board and for almost all Shared Component types. Starting with APEX 23.2, the edit page for a shared component is always *Read Only*, to prevent confusion when developers make changes, and these are being overwritten once the subscription is refreshed.

APEX now also shows the current subscription status, which means that developers can quickly see not only whether a component is subscribed, but also whether the subscribing component is up to date or needs refresh (**Fig. 17**).

Improvements to the APEX application builder include a more streamlined Create Application wizard, which allows to create a simple application with only one mouse click, an improved UI for application export and import, and various accessibility improvements.

A long-awaited new feature is the automatic display of a Success Message after a modal dialog closed. In the past that required some custom coding to work correctly, in APEX 23.2 it's just toggling the **Show Success Messages** switch in the **Close Dialog** process.

Application Static Files can now be moved to *Oracle Cloud Object Store*, directly from the APEX application builder. Once that is done, the application will automatically pull these files from Object Store at runtime, and not from the APEX database anymore.

Search Configurations in APEX 23.2 provide support for the new **Ubiquitous Search** feature *in Oracle Database 23c*. If APEX is running on a 23c database (the Oracle23c Free database is already available), a new Search type **Ubiquitous Search** is available when creating a search configuration.

Ubiquitous Search in Oracle Database 23c introduces the **DBMS_SEARCH** package, which makes it super-simple to create an Oracle Text index on multiple tables, to make all table columns searchable. Once such a DBMS_SEARCH index is in place, an APEX Search Configuration can be created with just a few clicks (**Fig. 18**).

Additional features include new Sample Datasets in Arabic and Italian, minor improvements to Background Page Processing, a refreshed Redwood Light theme style, several JavaScript library upgrades and many more features.

Summary

APEX 23.2 contains several new features and enhancements. Working Copies make it easy for small development teams to apply a change or a hotfix in a safe environment before making it available to end users. The new APEX workflow engine was long awaited and finally brings declarative business process modelling to APEX. The new custom Map Backgrounds were requested since the Map Region was introduced and are now finally there. And as always, it's *the combination* of all the new features and improvements – big ones and small ones – which make a new APEX release so interesting.

The best way to find out about everything is to sign up for a workspace on **apex.oracle.com**, or to get a subscription to Oracle Cloud "Always Free" Services, configure the APEX workspace and get rolling.

More Information

- Information and environment to test-drive APEX:
  http://apex.oracle.com/en

- Overview: What's new in APEX 23.2
  https://apex.oracle.com/en/platform/features/whats-new-232/

- Download APEX 23.2
  https://www.oracle.com/tools/downloads/apex-downloads/
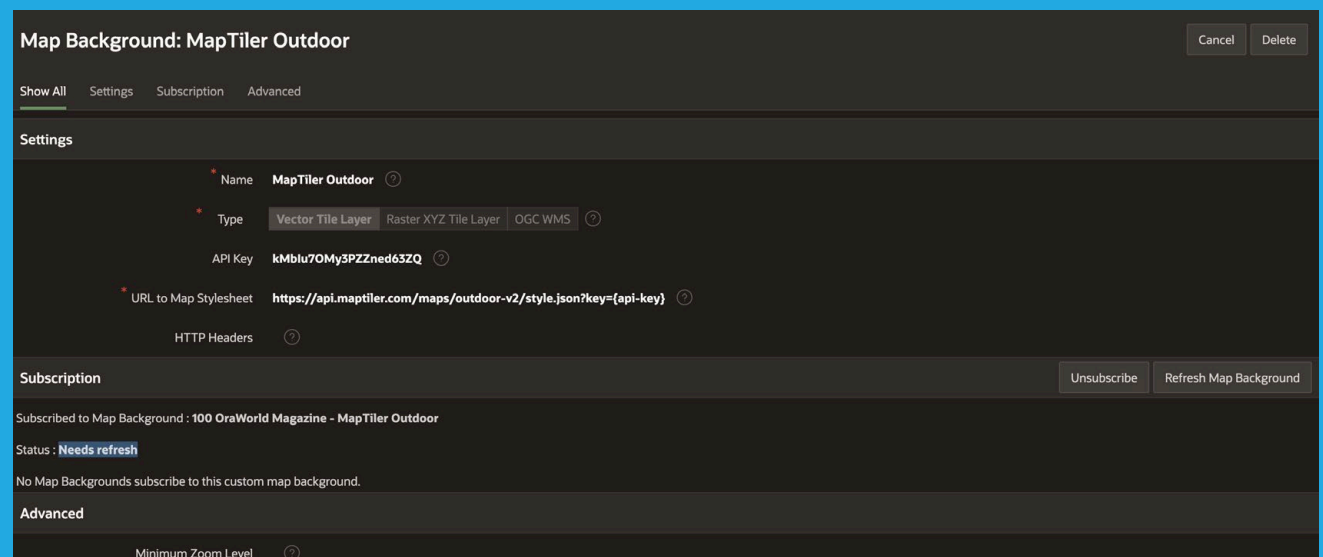
Fig. 17: A subscribed map background is stale and needs refresh.
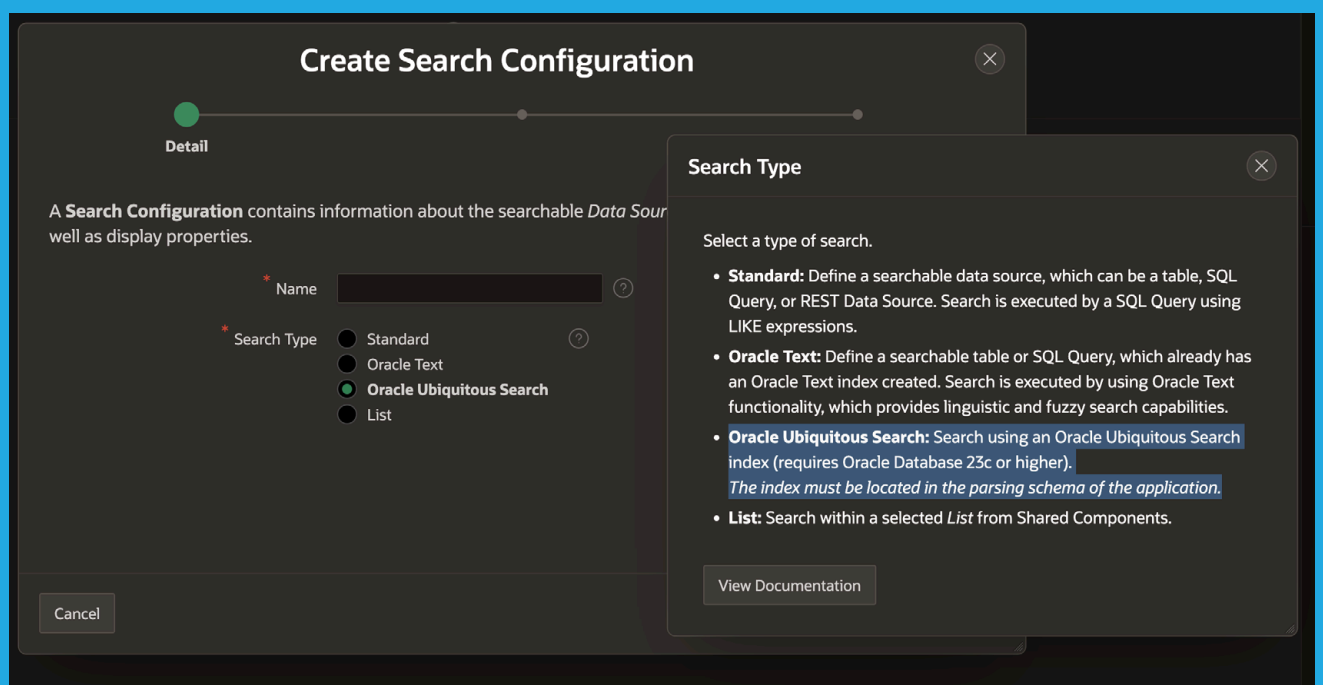
Fig. 18: APEX 23.2 provides built-in support for Oracle Database 23c Ubiquitous Search

About Carsten Czarski

Carsten works for Oracle in Germany since 2001. He started in the Presales organization helping customers and partners regarding database-centric application development. Since March 2016, Carsten is a member of the Application Express development team. Focus of his work is on the support for REST services in APEX, Data Loading, SQL Execution and a lot of other areas. He is a frequent speaker at international user group conferences.

Jim Czuprynski

Building an Equitable EV Charging Network Using APEX Native Map Regions

Modern business solutions often require robust geospatial feature sets to translate information about precise locations of physical assets, the logistics of routing product efficiently to market, the impact of traffic flows within cities and across boundaries, and even proof of regulatory compliance. Available since APEX 21.1, Native Map Regions provide developers access to powerful visual mapping technology that seamlessly incorporates Oracle Spatial Data Object (SDO) tools and techniques already available within Oracle Database.

I've always enjoyed working with maps. Indeed, it became a necessity in my childhood: My father was a careful and safe driver, but he would invariably get lost whenever we travelled just a few miles outside our native Chicago neighborhood. And as a somewhat precocious ten-year-old, it also let me give my dad sudden orders like "No, turn *right, now!*" while my mother muttered things like "We are going to be so late to my sister's party!" under her breath.

So, when the APEX Native Map Region debuted a few years back, I immediately began experimenting with its feature set. I discovered it's actually possible to solve a complex geospatial business case with just a few lines of SQL code while simultaneously viewing results in real time on an interactive map within an APEX application.

Business Case: US 2022 Inflation Reduction Act's Justice40 Initiative

First, some background on the business case we're trying to solve:

- In 2022, the United States Congress passed the Inflation Reduction Act (IRA). Among its many sweeping mandates, it's intended to bring US electrical infrastructure into the 21st century by emphasizing the need to decarbonize as quickly as possible, including a rapid expansion of the electrical vehicle (EV) charging infrastructure throughout the country.
- Even though many battery electric vehicles (BEVs) still cost more than normal internal combustion engine (ICE) powered vehicles, prices have continued to fall as US and foreign automakers shift towards purely EV production in the coming decades. And it's not just automakers who are shifting their orientation; logistics organizations such as Amazon and the US Postal Service (USPS) are rapidly shifting their last-mile delivery vehicles to BEVs.
- The IRA also recognized that EV charging is currently

centralized in metropolitan areas with higher incomes, so it also included an initiative aptly named Justice40, which encourages each US state to place EV chargers equitably so that their 40% most underserved populations will have access to EV charging resources.
- Without sufficient charging stations, many poorer areas wouldn't be able to provide charging resources for commercial and personal BEVs, and many last-mile delivery vehicles would have limited incentive to offer delivery services. For example, rural Americans – some of the poorest areas of the country – disproportionately depend on the USPS for crucial deliveries of medicine and other supplies.
- Building out this new energy infrastructure will offer excellent employment opportunities for folks in poorer communities as well. Manufacturing, installing, and maintaining EV components and their sprawling infrastructure over the next decade is even more ambitious than the 20th century mandate known as the Tennessee Valley Authority (TVA) to electrify most of the rural southern USA.

Locating the Disadvantaged

To prototype a possible solution, I decided to focus on the State of Wisconsin, just north of my home state of Illinois. I admit a vested interest in my research, since I own an EV and often drive through much of southern and central Wisconsin every few weeks.

First, I needed to plot the boundaries of the most disadvantaged communities within the state. Thankfully, the **US Department of Transportation (USDOT) web site** shown in **Fig. 1** hosts a cornucopia of data that state governments can use to determine which census tracts contain *disadvantaged communities* (DACs) within each state. I downloaded the data for the State of Wisconsin from the DOT downloads page (**Fig. 2**).
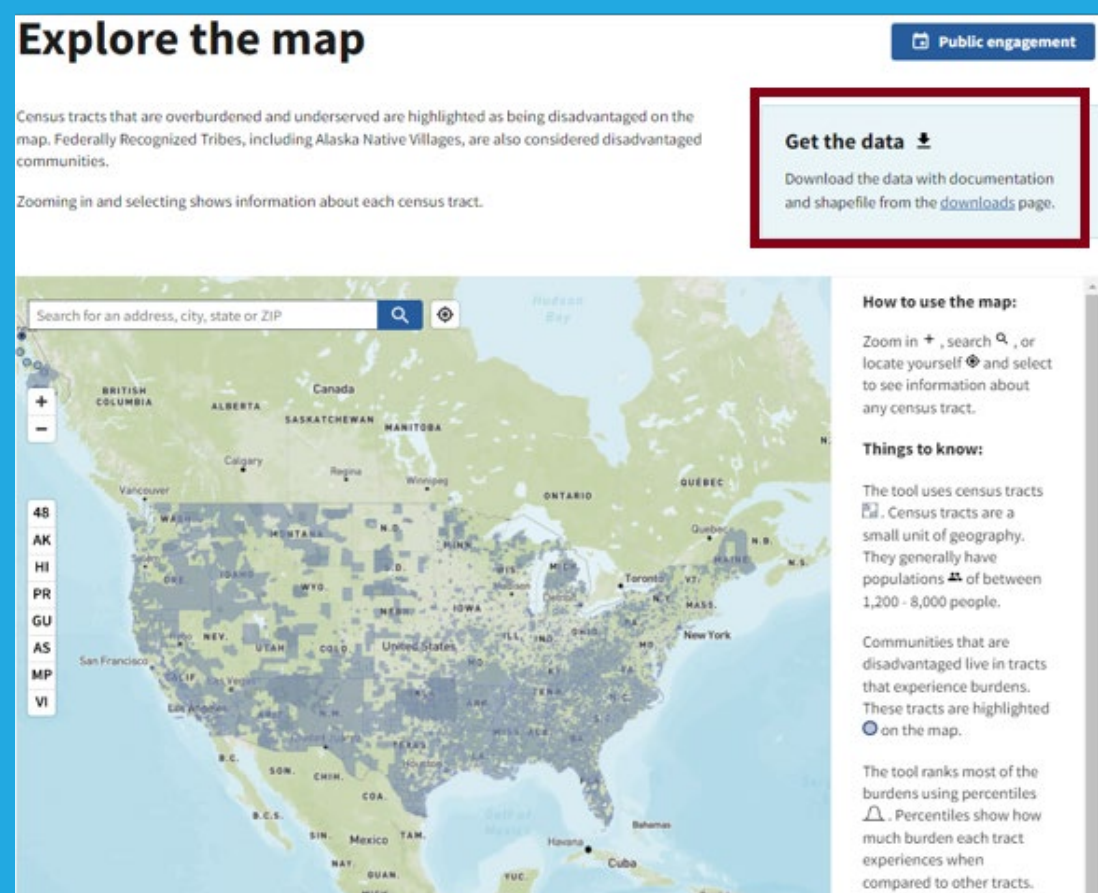
Fig. 1: US Department of Transportation (DOT) Home Page for Disadvantaged Communities
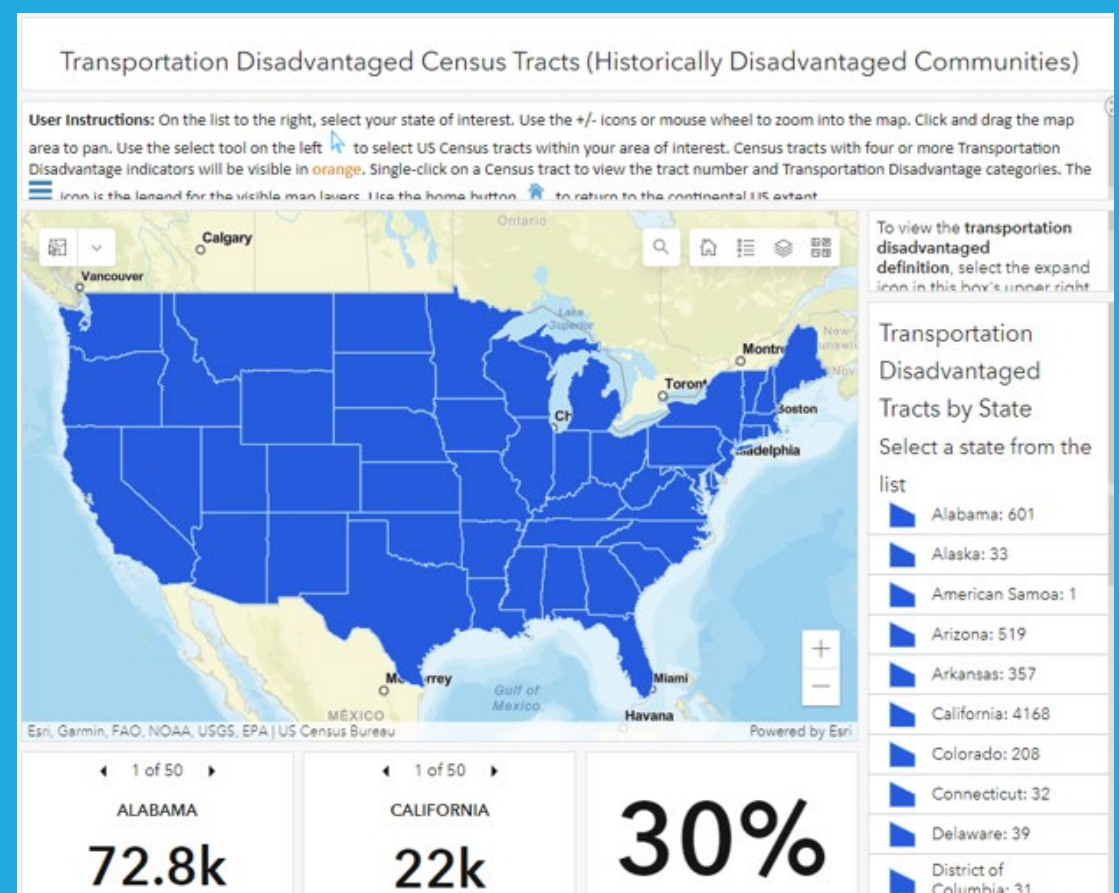
Fig. 2: Gathering data from US Department of Transportation (DOT)

The US DOT data uses several dozen individual factors – things like annual median income, population age ranges, employment availability, and even environment health metrics – to calculate an overall *disadvantage* score for each community at the census tract level. A single census tract in the USA typically contains between 1,000 and 7,000 people, so larger metropolitan areas will have many more than exurban or rural areas.

Gleaning and Gathering Data

Each US DOT dataset is stored as an ArcGIS shapefile, so I used Oracle Spatial Studio to convert its contents into a database table named **DOT_DISADVANTAGED_LAYERS**. As **Fig. 3** shows, Spatial Studio created all necessary columns and even automatically created a column named **GEOM** with a datatype of **SDO_GEOMETRY** that incorporated the longitude and latitude boundaries of the polygons representing each census tract.

Obtaining information about existing and potential EV chargers proved a bit more problematic. I had already procured a list of *existing* available chargers from an earlier project stored as an Excel spreadsheet; however, I had limited insight into which local governments and businesses were actually *planning* to implement new charging stations. I therefore decided to make some common-sense projections based on which businesses were most likely to invest in EV charging in coming years. I settled on several types – gas stations, hotels, grocery stores, pharmacies, and other retail establishments – and spent a weekend scraping locations from their corporate websites for probable locations. I captured the results of that labor in a single table named **CHARGING_POINTS** (**Listing 1**).

Unfortunately, none of these actual or potential charger locations included the needed geospatial reference points, so I also used Spatial Studio to enrich the data in that table with longitude and latitude values based on address, locality, state, and postal codes (**Fig. 4**).

Of course, I could have used other mechanisms to capture those data, but Spatial Studio completed the task in a matter of minutes, including the creation of a new column named **GC_GEOMETRY** of datatype **SDO_GEOMETRY** (**Fig. 5**). Spatial Studio also created the spatial index on that column I'd need to cross-reference the universe of chargers versus the census tracts of the disadvantaged communities (**Fig. 6**).
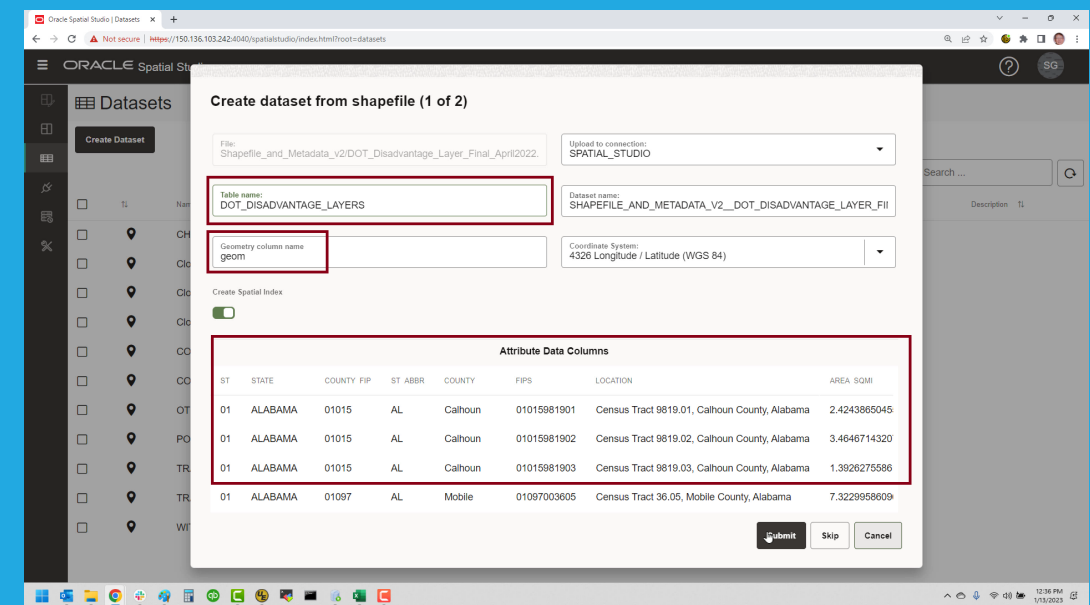
Fig. 3: Converting an ArcGIS ShapeFile

```
SQL> CREATE TABLE wtfc.charging_points (
     cp_id              NUMBER(8,0)    NOT NULL
    ,cp_name            VARCHAR2(60)   NOT NULL
    ,cp_address         VARCHAR2(40)   NOT NULL
    ,cp_city            VARCHAR2(40)   NOT NULL
    ,cp_state_abbr      VARCHAR2(02)   NOT NULL
    ,cp_zip_code        CHAR(05)       NOT NULL
    ,cp_status          CHAR(03)       NOT NULL
    ,cp_bus_type        VARCHAR2(03)   NOT NULL
    ,cp_chg_type        VARCHAR2(04)   NOT NULL
    ,cp_chg_avail       VARCHAR2(04)   NOT NULL
    ,cp_potentiality    NUMBER(8,6) );
```

Listing 1: Creating the CHARGING_POINTS table
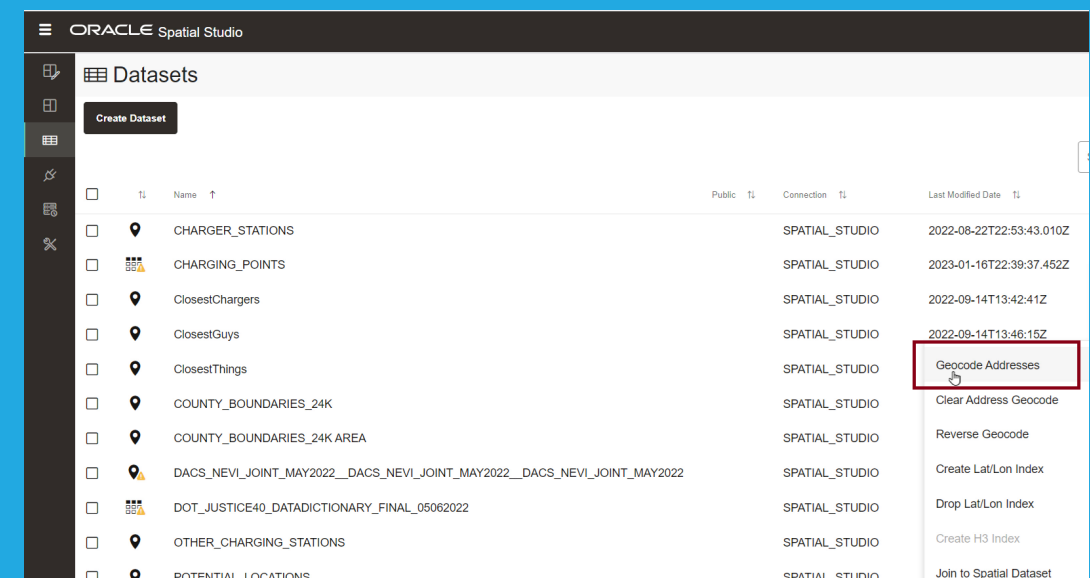
Fig. 4: Geocoding – Picking the Table
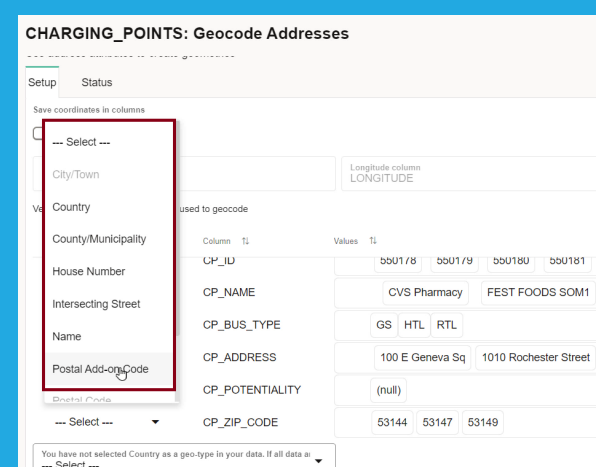
Fig. 5: Geocoding – Choosing Best Columns
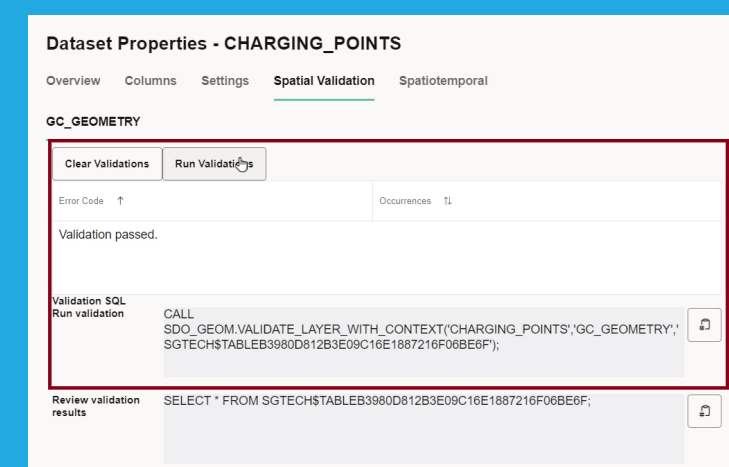
Fig. 6: Validating New GC_GEOMETRY Contents
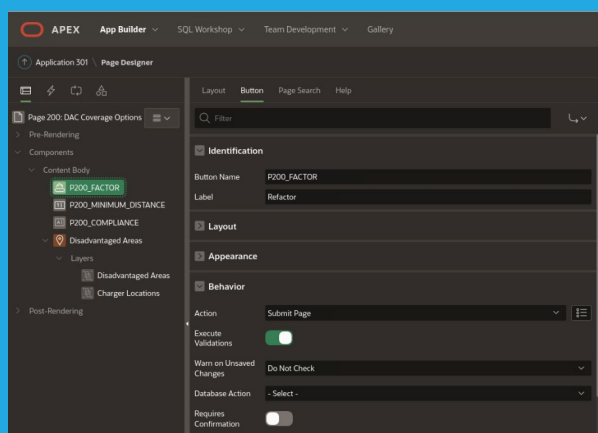
Fig. 7: Button
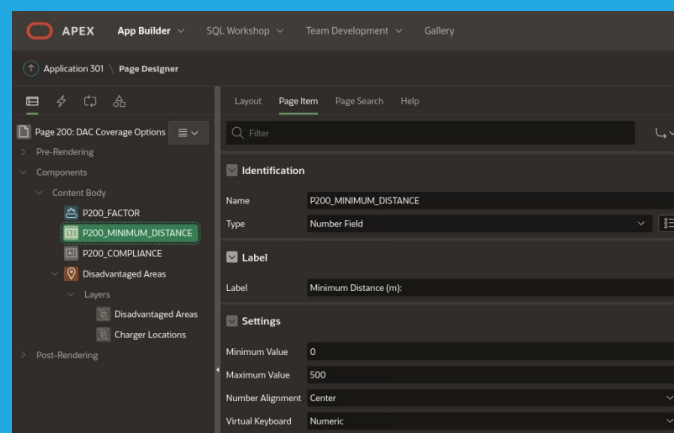
Fig. 8: Distance Selector
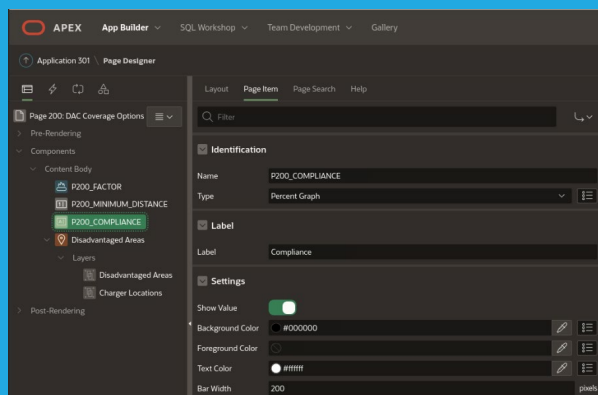
Fig. 9: Percentage Graph
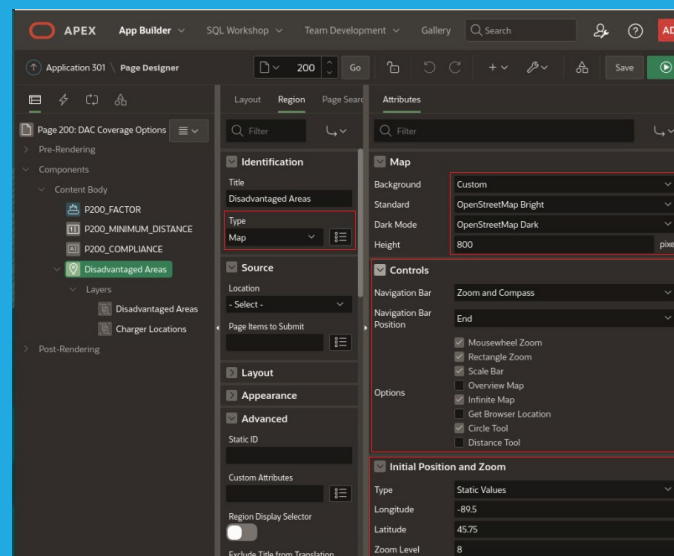
Fig. 10: Native Map Region
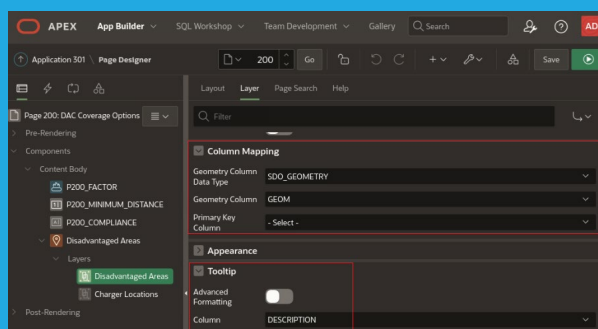
Fig. 11: Bottom Layer – DACs


```
Code Editor - SQL Query

1  SELECT
2      geom
3      ,(e_totpop * sumdis) AS dac_factor
4      ,'#' || fips || ': ' || e_totpop || ' residents: $' || avg_annual AS description
5    FROM hol23c.dot_disadvantage_layers
6   WHERE st_abbr = 'WI'
7   ORDER BY sumdis DESC, e_totpop ASC
8   FETCH FIRST 450 ROWS ONLY;
9
```

Fig. 12: Bottom Layer – Query to Retrieve DAC Information


```
Code Editor - SQL Query

1  SELECT
2      geom
3      ,(e_totpop * sumdis) AS dac_factor
4      ,'#' || fips || ': ' || e_totpop || ' residents: $' || avg_annual AS description
5    FROM hol23c.dot_disadvantage_layers
6   WHERE st_abbr = 'WI'
7   ORDER BY sumdis DESC, e_totpop ASC
8   FETCH FIRST 450 ROWS ONLY;
9
```

Fig. 13: Top Layer – Potential Charger Locations

So ... Where's the Fastest Charger?

To put this all together, I created a new APEX application named **WTFC** to represent the question that most EV owners mutter under their breath when their battery percentage drops below 20%. (I typically find myself muttering something like *"Where's the fastest charger?"* but I will admit to having occasionally used slightly different words with the same initials.)

Then I created a new page named DAC Coverage Options that contains four objects:

• A *button* (**Fig. 7**) that submits the page when clicked
• A *number field* (**Fig. 8**) that will specify minimum acceptable distance (in meters) between DACs and potentially close-by EV chargers
• A percentage graph (**Fig. 9**) that shows how well the Justice40 coverage provision is satisfied, based on the number of EV chargers closest to DACs
• An *APEX Native Map Region* (**Fig. 10**) with two layers: one to display the DACs and their boundaries, and one that displays EV chargers based on the minimum acceptable distance specified

Native Map Regions and Layers

One of the neatest features of Native Map Regions is the detailed control I have over how the map will be displayed. As **Fig. 10** shows, I can specify the map's initial zoom level, precisely where it will be centered, and even which types of map backgrounds to use.

Each of the two layers implement the business logic that determines how close I can approach the target of 40% DCA coverage across the state of Wisconsin. The *bottom* layer (**Fig. 11**) displays all relevant information about the 450 most disadvantaged DACs in the state, including their census tract boundaries.

I've also applied some visual cues into this layer map, with smallest communities represented by darker greens and more populated communities ranging toward yellows and reds. That's implemented through the query in **Fig. 12**, which returns the 450 most disadvantaged DACs in Wisconsin.

Meanwhile, the *top* layer (**Fig. 13**) shows just how well the Justice40 coverage provision is satisfied based on the number of EV chargers closest to DACs. (We'll delve into the business logic implemented within this layer in just a bit, but it's actually handled by a single SQL statement.) I've employed some visual cues in this layer as well:

- *Brown* pegs represent *proposed* chargers, while *blue* pegs represent ones that *currently* exist.
- A *round* peg shape means the charger is a lower-powered *Level 2* charger – usually placed nearer hotels or public spaces for longer-term charging – but a *square* peg represents a *Level 3* "fast" charger that can fully charge a typical EV battery in less than 30 minutes.

The Secret Sauce: Calculating Effective Coverage

The real magic within this APEX page is a single SQL statement. As **Fig. 14** shows, it's not as complicated as you might think.

Let's break down what this SQL statement implements within the top layer of our APEX Native Map Region:

- That *first* subquery labeled **APC** captures values from the **CHARGING_POINTS** table – for example, each charging point's actual location on the map, its charging capacity, and ancillary information about each location like its physical address – as well as the boundaries of each polygon in the **DOT_DISADVANTAGE_LAYERS** table.
- The *last* subquery labeled **DF** isolates the 450 most disadvantaged communities in the state of Wisconsin. Since there are approximately 1100 DACs in Wisconsin, these 450 DACs will comprise 40.9% or just enough of that 40% threshold the Justice40 directives require to be satisfied to receive any government funding,

as described earlier. Note that the **ORDER BY** clause sorts the DACS from most to least disadvantaged, so the **FETCH FIRST** clause returns only the desired 450 census tracts.
- That *middle* clause in the APC subquery is the secret sauce: Its **SDO_WITHIN_DISTANCE** function tells the Native Map Region to display on the map any planned or actual charger that falls within each of those 450 most disadvantaged communities, but *only if they are within the specified distance of any of the qualified DACs*. By default, we'll start with a minimum distance of zero (0) meters that we'll specify in the editable page item named **P200_MINIMUM_DISTANCE**.

How Close Is Close Enough?

Our initial trial run of the Native Map Region (**Fig. 15**) with the minimum distance set to zero meters shows a less than satisfactory outcome: Only about **53%** percent of the coverage goal has been attained. (The percent graph at the top of the page is refreshed automatically through an almost-identical query to the one shown in **Fig. 14**; whenever I change the minimum distance value and click the *Refactor* button, the page refreshes both the Native Map Region and the graph.)

A closer look at the EV chargers near the city of Appleton (**Fig. 16**) shows there are only four chargers within any of the three DACs nearby.


```
Code Editor - SQL Query

1   SELECT * FROM
2   (SELECT /*+ INDEX(DA hol23c.dot_disadvantage_layers_geom_spidx, CP hol23c.charging_points_gc_geometry_spidx */
3       CP.cp_id
4       ,CP.cp_name AS info_title
5       ,CP.cp_address || ', ' || CP.cp_city || ' ' || CP.cp_zip_code AS info_body
6       ,DECODE(CP.cp_status,'PLN','brown','ACT','blue','purple') AS color
7       ,DECODE(CP.cp_chg_type,'L2','Pin Circle','L3','Pin Square','point') AS icon
8       ,CP.cp_bus_type
9       ,CP.gc_geometry
10      ,DA.fips
11      ,DA.location
12   FROM
13        hol23c.charging_points CP
14       ,hol23c.dot_disadvantage_layers DA
15   WHERE SDO_WITHIN_DISTANCE(DA.geom, CP.gc_geometry, 'distance=&P200_MINIMUM_DISTANCE. unit=meter') = 'TRUE'
16   ORDER BY CP.cp_id) APC,
17   (SELECT /*+ NOINDEXES */
18       fips AS ltd_fips
19       FROM hol23c.dot_disadvantage_layers
20      WHERE st_abbr = 'WI'
21      ORDER BY sumdis DESC, e_totpop ASC
22      FETCH FIRST 450 ROWS ONLY) DF
23   WHERE DF.ltd_fips = APC.fips;
```

Fig. 14: Top Layer – Query to Retrieve Potential Chargers Based on Distance to DAC Boundaries
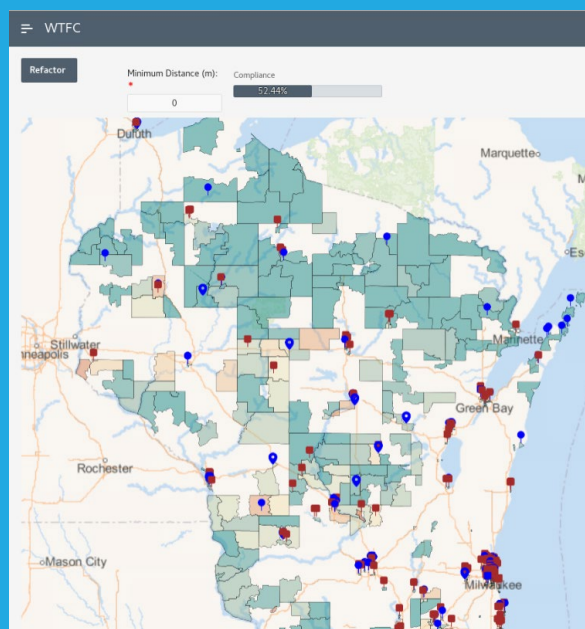
Fig. 15: Big Picture – Zero Meters Distance
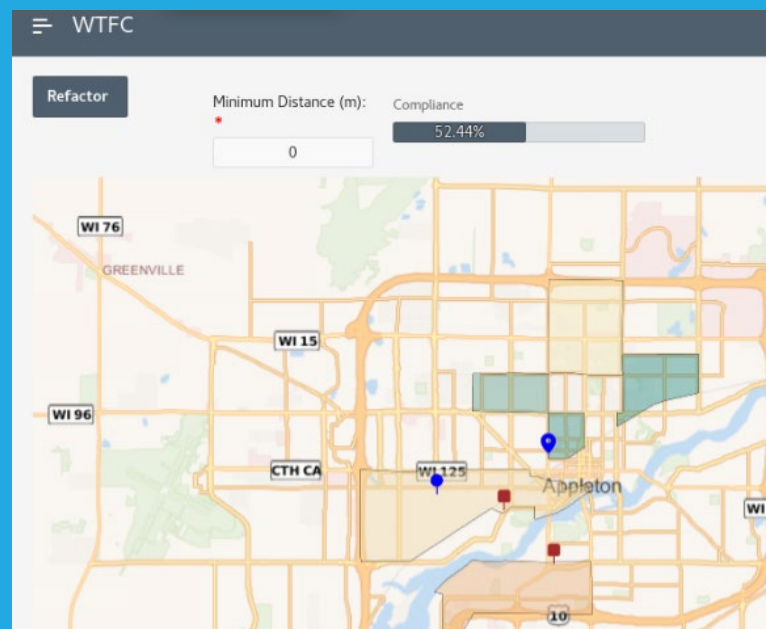
Fig. 16: Looking Closer – Zero Meters Distance

However, what if it were possible to satisfy the Justice40 regulations if there were some chargers *just barely outside* the boundaries of these 450 DACs – maybe, say, 100 meters (about 330 feet)? I can explore that by changing the *Minimum Distance* value to 150 and refreshing the page. **Fig. 17** shows the result at large scale – an improvement to **80%** of coverage.

Looking a bit more closely again at the vicinity of Appleton (**Fig. 18**), there are now eight chargers within range of its three DACs.

What if I bump **Minimum Distance** to 250 meters (just over 800 feet)? Success at last: the mandated Justice40 coverage is now 100%. Again, the big picture (**Fig. 19**) shows quite a few more EV chargers close to DACs; looking closely again at the vicinity of Appleton (**Fig. 20**), there are now 11 chargers within range of its three DACs.

Wrapping Up: Deeper Exploration Available

This brief introduction into APEX Native Map Regions demonstrates robust geospatial problem-solving is available at any APEX developer's fingertips with low code and little technical debt. The good news is if you'd like to explore this scenario in much more detail, you can run through these exact scenarios in **this Oracle Live Lab session** I published in conjunction with my theatre session on Native Map Regions at Oracle CloudWorld 2023. This Live Labs session takes only 45 minutes to complete; as an added bonus, it also demonstrates how to use *Extruded Polygons* within Native Map Regions to dive deeper into myriad information contained within the US DOT DAC data sources.
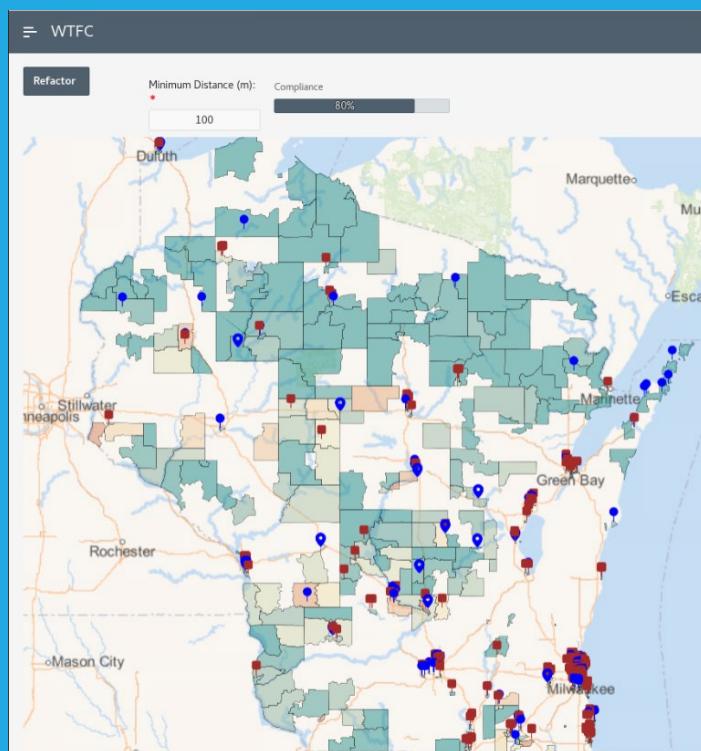
Fig. 17 – Big Picture: 100 Meters Distance
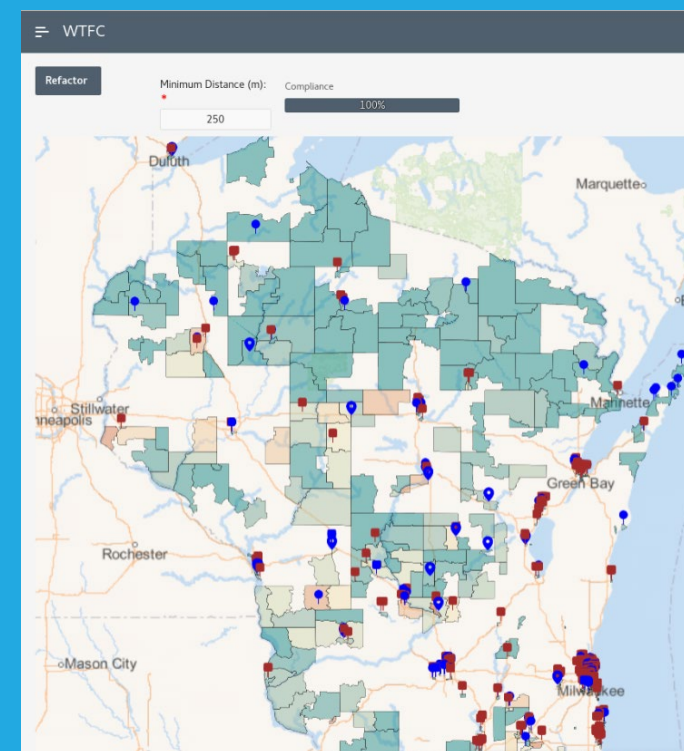
Fig. 19: Big Picture – 100 Meters Distance
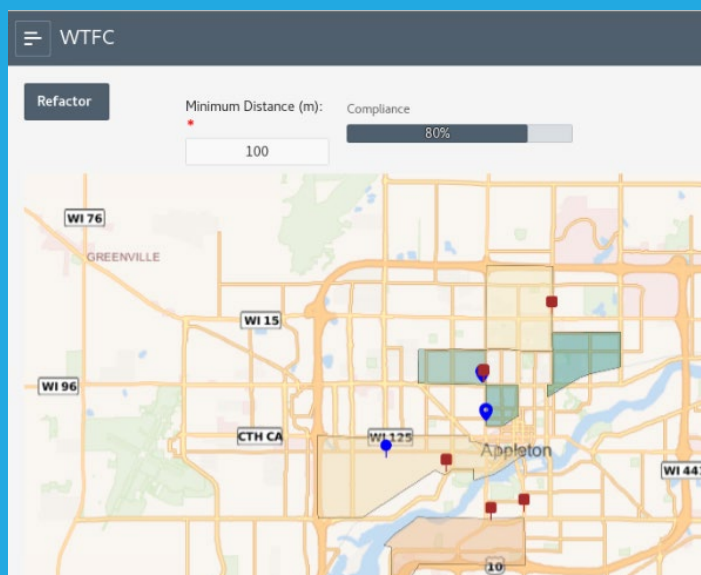
Fig. 18: Looking Closer – 100 Meters Distance
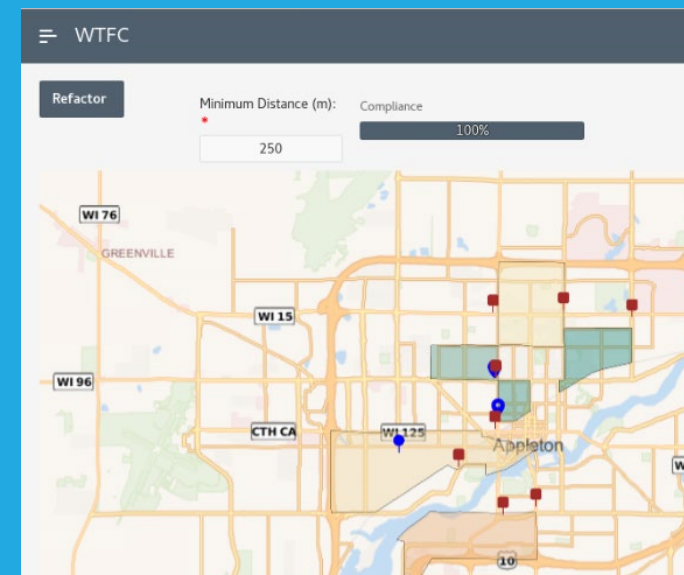
Fig. 20: Looking Closer – 100 Meters Distance

References

Spatial Studio Guide:
https://docs.oracle.com/en/database/oracle/spatial-studio/index.html
Spatial Studio Application in Oracle Cloud Marketplace:
https://cloud.oracle.com/marketplace/application/71472162/overview
Oracle Spatial Developer's Guide:
https://docs.oracle.com/en/database/oracle/oracle-database/19/spatl/index.html
APEX 21.1 Native Map Regions:
https://docs.oracle.com/en/database/oracle/application-express/21.1/htmdb/creating-maps.html
Carsten Czarski's Article on APEX Native Map Region:
http://www.oraworld.org/fileadmin/documents/26-ORAWORLD.pdf
SDO_GEOMETRY Datatype:
https://docs.oracle.com/en/database/oracle/oracle-database/23/spatl/spatial-datatypes-metadata.html
SDO_NN_DISTANCE Function:
https://docs.oracle.com/en/database/oracle/oracle-database/23/spatl/spatial-operators-reference.html
US Department of Transportation Justice40 Portal:
https://www.transportation.gov/equity-Justice40
National Electric Vehicle Infrastructure (NEVI) Competitive Program:
https://driveelectric.gov/

About Jim Czuprynski

Jim has 40+ of professional experience in information technology throughout his career – mainly focused on application development – before becoming an Oracle DBA in 2001. He was named an Oracle ACE Director in 2014 and is a sought-after public speaker on Oracle Database technology features, presenting often at Oracle OpenWorld, IOUG COLLABORATE, ODTUG KSCOPE, Oracle Development Community tours, and Oracle User Group conferences around the world.

Jim has authored well over 100 articles focused on facets of Oracle Database administration since 2003 published on databasejournal.com, IOUG SELECT, ODTUG TechCeleration, and OracleWorld. He has also co-authored four books on Oracle database technology. Jim's blog, **Generally...It Depends**, contains his regular observations on all things Oracle and the state of the IT industry.

Need some support in achieving your Sustainability goals for 2024?

Claudia Philipps

Oracle Consulting can help! We offer a variety of packages supporting your Sustainability needs in terms of Modernization, Integration, Reporting, and many other.

Consumers, Investors, Regulators and Employees have all been focusing on Sustainability in the past months when developing their corporate strategy. Recent world events including the global COVID 19 pandemic, climate change, and political tensions have all played a role in driving this situation. Nielsen studies show that 66% of consumers would spend more for a product if it came from a sustainable brand, and 81% of global consumers feel strongly that companies should help improve the environment.

Sustainable IT using Oracle Cloud Services

Oracle Cloud Services can support your Sustainability goals from a variety of perspectives. Once and foremost, Oracle Cloud Infrastructure (OCI) is a high-performance green cloud solution. A high-density computing environment powered by renewable resources, OCI aligns with the circular economy to support a clean cloud solution. Furthermore, Oracle Cloud reduces its

Fig 1: Technology as an Enabler for Sustainability within your organization (© Oracle).

environmental footprint by leveraging state-of-the-art cooling and energy efficient technologies at our green data centers. For example, in Europe, Oracle Cloud data centers are powered using 100% renewable energy. Our goal is to achieve 100% renewable energy use in all our OCI data centers by 2025.

Leverage the expertise of Oracle Consulting

Of course, Oracle Consulting is here to address your needs and obstacles when it comes to achieving your sustainability goals using Oracle Cloud Services. No matter, if you'd like to migrate your IT infrastructure to the Cloud, develop innovative solutions using advanced technology tools, or comply with local regulations that are mandatory in the field of sustainability. Oracle Consulting is able to provide migration support, targeted expertise, and analytic outcomes taking advantage of our experienced workforce.
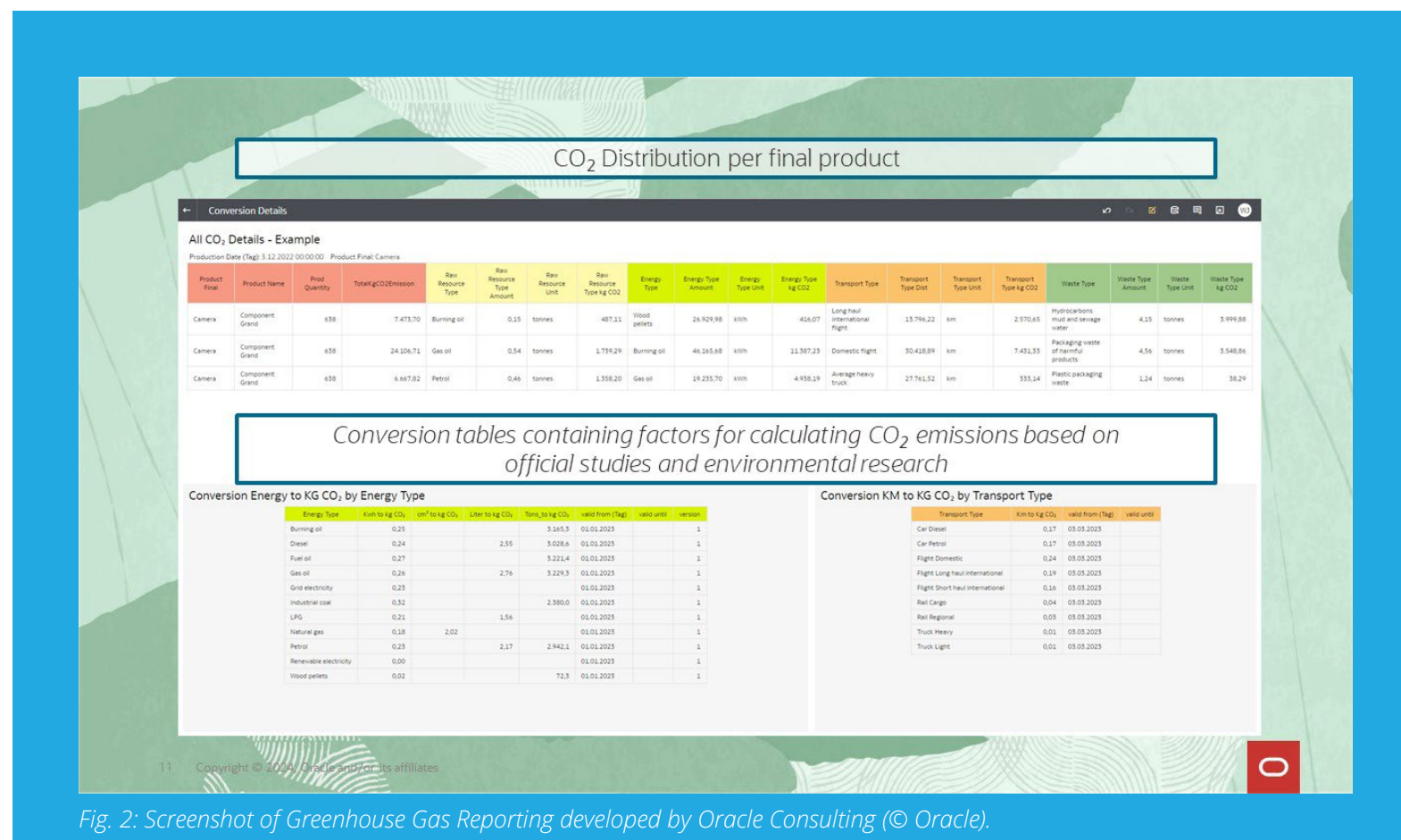
Fig. 2: Screenshot of Greenhouse Gas Reporting developed by Oracle Consulting (© Oracle).

Crucial System Integration to enable mandatory Greenhouse Gas (GHG) Reporting

Let me highlight the Oracle Consulting Analytics IP (Intellectual Property) which has been developed together with a set of customers from different industries. This highly flexible reporting package allows you to set up a Greenhouse Gas (GHG) Reporting which complies with the regulations required for Scope 1, 2, and 3 of Carbon Reporting. The biggest challenge when it comes to GHG Reporting is the aspect of integration. And here, you can rely on the expertise of Oracle Consulting to help you identify the right sources and integrate them into your reporting dashboard.

Achieving your Sustainability goals

Oracle Consulting can support you in achieving your sustainability goals and ultimately to save the earth by managing your data from every source in a precise and fast way, set goals, and achieve the necessary changes to keep the earth a good place to live in. If you have any questions or would like additional information, please, reach out to your local Oracle contact or get in touch with the author directly.

About Claudia Philipps

As we only have one world that we are living in, **Claudia**, Business Development Manager at Oracle, is passionate about the topic of Sustainability in order to save the environment we live in. In this context, Claudia strongly believes in the power of technology to achieve sustainability goals. Claudia not only believes in the power of technology to achieve sustainability goals but simultaneously, leveraging technology to build success for the company. Claudia takes a lot of pleasure in working with people. To achieve success, Claudia takes on business challenges and develops personnel to facilitate their success and the success of the organization.

Call for Papers

AOUG Conference 2024

March 17, 2024
https://www.aoug.at/Event/633

ConTech2024 by RoOUG

March 24, 2024
https://contech2024.rooug.ro/

Events

Oracle CloudWorld Tour London

March 14 2024
London, UK
https://www.oracle.com/uk/cloudworld-tour

APEX World

March 21 + 22, 2024
Amersfoort, Netherlands
https://nloug.nl/en/events/apex-world-2024-en

UTOUG Training Days 2024

March 21 + 22, 2024
Salt Lake City, US
https://www.utoug.org/training-days.html

JavaLand

April 9 - 11, 2024
Nürburgring, Germany
https://www.javaland.eu/en/home/

APEX Alpe Adria 2024

April 12, 2024
Maribor, Slovenia
https://www.aaapeks.info/home

SOUG Day Spring

April 17, 2024
Basel, Switzerland
https://soug.ch/event/soug-day-spring-17-april-2024-inklusive-generalversammlung/

APEX connect

April 22 - 24, 2024
Düsseldorf, Germany
https://apex.doag.org/en/home/

eInvoicing update for Europe and Benelux

April 24, 2024
Oracle Office, Utrecht, Netherlands
https://theoraclecommunity.eu/

OUGN 2024

April 24 + 25, 2024
Oslo, Norway
https://2024.ougn.no/

DOAG 2024 Database with Exaday

May 15 + 16, 2024
Düsseldorf, Germany
https://datenbank.doag.org/en/home/

ConTech2024 by RoOUG

May 23 + 24, 2024
Bucharest, Romania
https://contech2024.rooug.ro/

MakeIT 2024 & JCON Slovenia 2024

May 29 - 31, 2024
Portorož, Slovenia
https://makeit.si/

AppsConnected24

June 6, 2024
Mauritskazerne in Ede, Netherlands
https://theoraclecommunity.eu/en/event/appsconnected24/

AOUG Conference 2024

June 17 + 18, 2024
Vienna, Austria
https://www.aoug.at/

CloudLand

June 18 - 21, 2024
Phantasialand in Brühl, Germany
https://www.cloudland.org/en/home/

ODTUG KSCOPE24

July 14 - 18, 2024
Nashville, US
https://kscope24.odtug.com/