September 2022, Edition #28

ORAWORLD

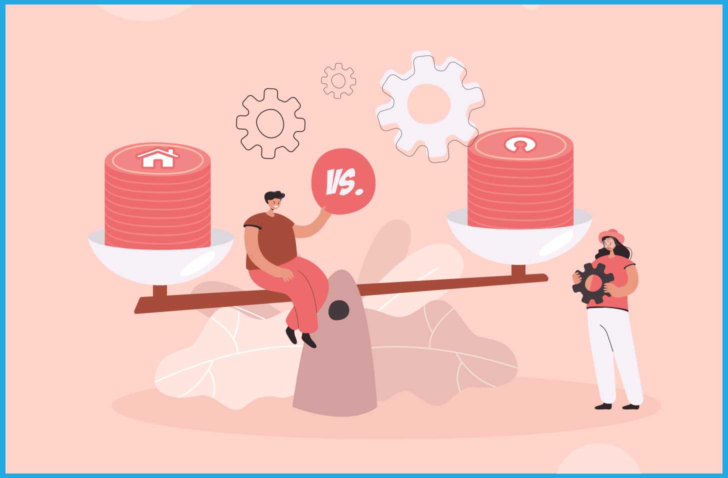
e-Magazine for Oracle Users published by the EOUC

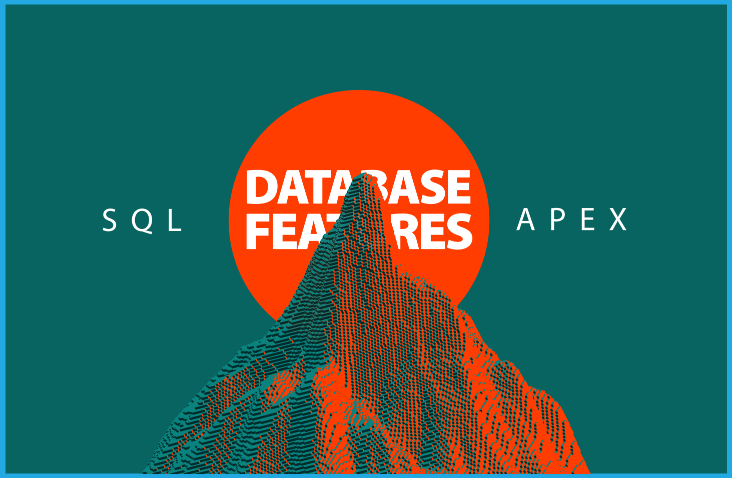
Open source database or proprietary ones? The pros and cons

APEX Series Part 11: Faceted Search, Pivot, JSON, and more

"On-Premise still has a bright future"

EOUC





I. Work & Life

II. Techs & Nerds

III. Users & Groups

IV. Past & Future

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

Editorial

Dear ORAWORLD Readers,

It was a long break for ORAWORLD. Unwanted. Unplanned.
A lot has happened since the last published edition; ORAWORLD has seen it, heard it, but only now managed to come back.

A very rich fall is in front of us, the Oracle community.
- The Oracle ACE Program has relaunched with some much appreciated improvements. The most important from all is that Oracle ACE Program is back, and it is active!
- The Oracle User Group Program, in a different form than the one we knew, is also going to be active again. We hope!
- The EMEA OUGs have scheduled their conferences one after the other: September kicked off with OUG Ireland, in conjunction with the APEX Nordic Tour. POUG followed, then DOAG conference+exhibition. The transition to October is marked by the Oracle Community EMEA Tour: 10 events hosted by CASOUG, AzerOUG, DOUG, LuxOUG, RoOUG, OUGN, HROUG and LVOUG. Some of these events are in-person, some hybrid, some exclusively online.

How else to end this chain of events than with Oracle Cloud World, in Las Vegas?
The autumn conference season will end in early December with UKOUG's 'Breakthrough'22'.
In addition to these technical conferences, there is another event worth mentioning: OUG EMEA leaders meet face-to-face in Nuremberg for the traditional Autumn Summit.
Busy times!

I hope you have a break in between sessions to browse the current issue of ORAWORLD. You will learn how to define and use Blockchain tables in Oracle.

The current issue contains an interesting comparison that will help you decide what to choose: proprietary DB or open source DB. Of course, the APEX series article is not missing. In this issue, you'll find out how APEX features make intensive use of the Oracle Database and its SQL functionality.
If you are wondering what other user groups are doing, in this issue you can read about the EBS On-Premise Days, organized by AUFO. You can also learn about the Australian Oracle User Group(AUSOUG). Yes, you read it right: Australia, the country which is on the other side of the Earth! Spoiler alert:no, things are not upside-down for AUSOUG!

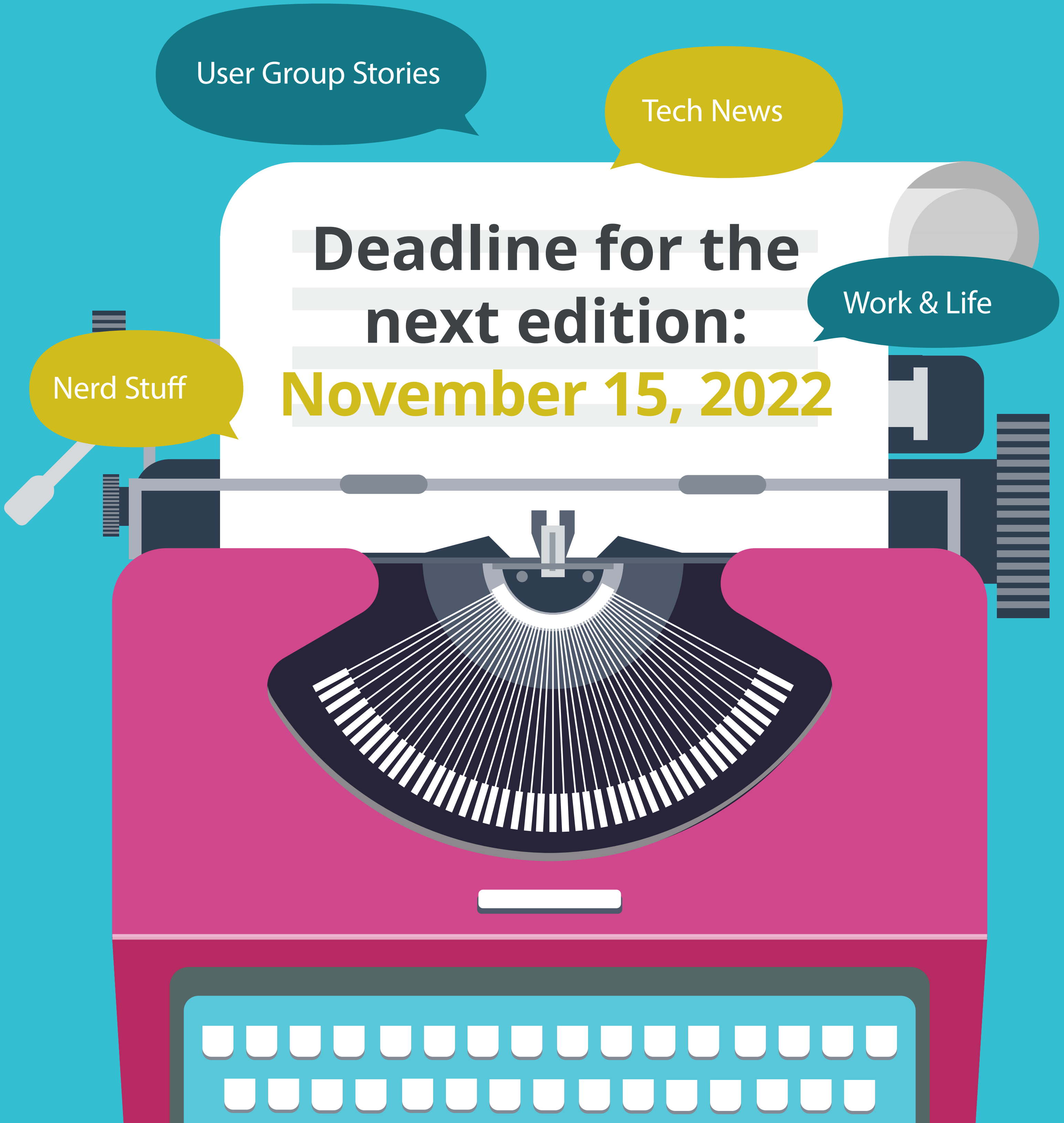
I hope you will find this edition of our eMagazine useful and enjoyable. We encourage you to send us your thoughts or any other comments you might have.
Please share ORAWORLD Magazine and our website with your co-workers and colleagues and invite them to subscribe. It's free!

I am looking forward for your submissions and feedback!

Yours,
Mirela Ardelean
Oracle ACE Pro, RoOUG Vice president, EOUC Board member

Submit Your Article!

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

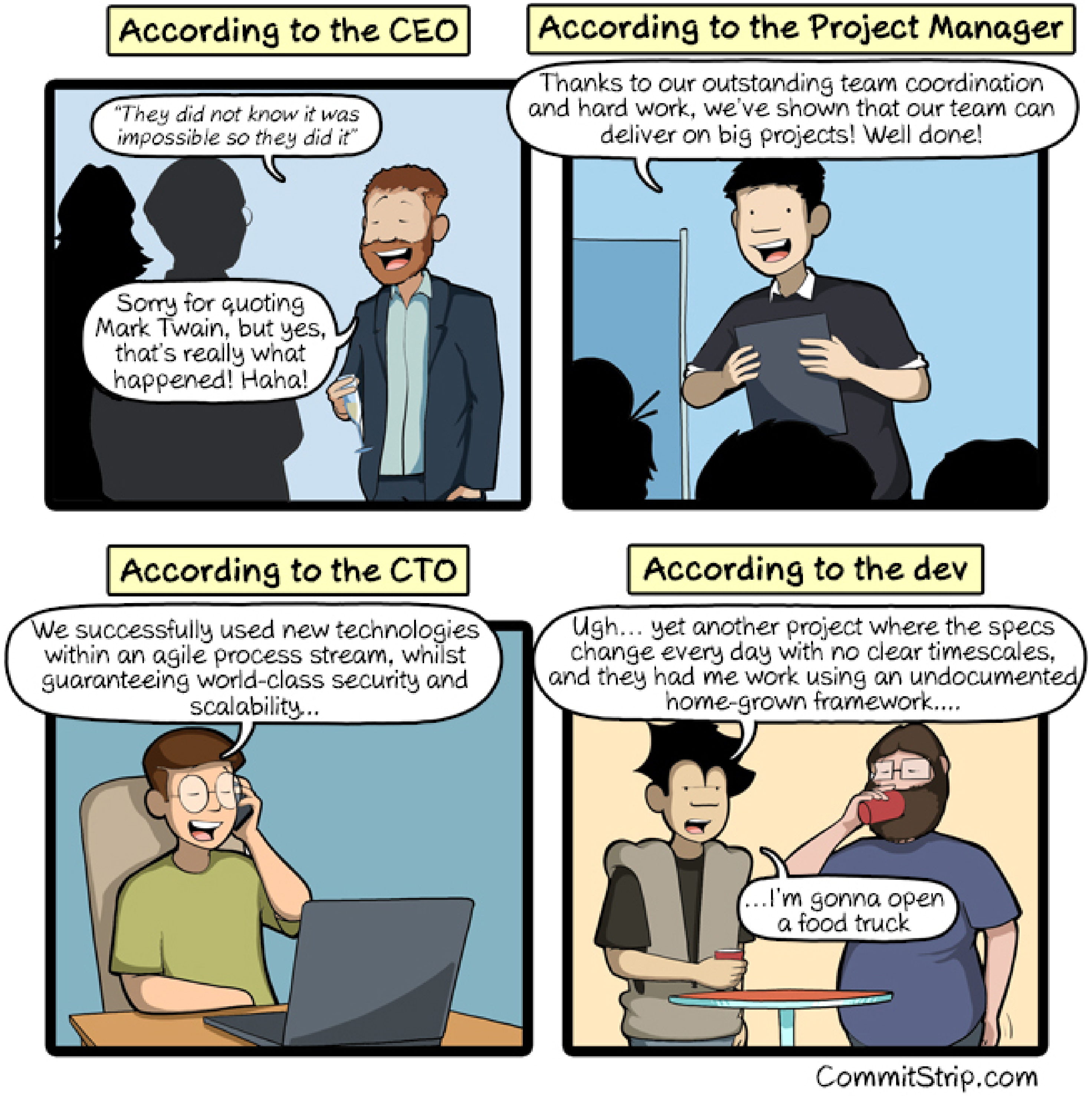


Deadline for the next edition: November 15, 2022

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

„End-of-project review“

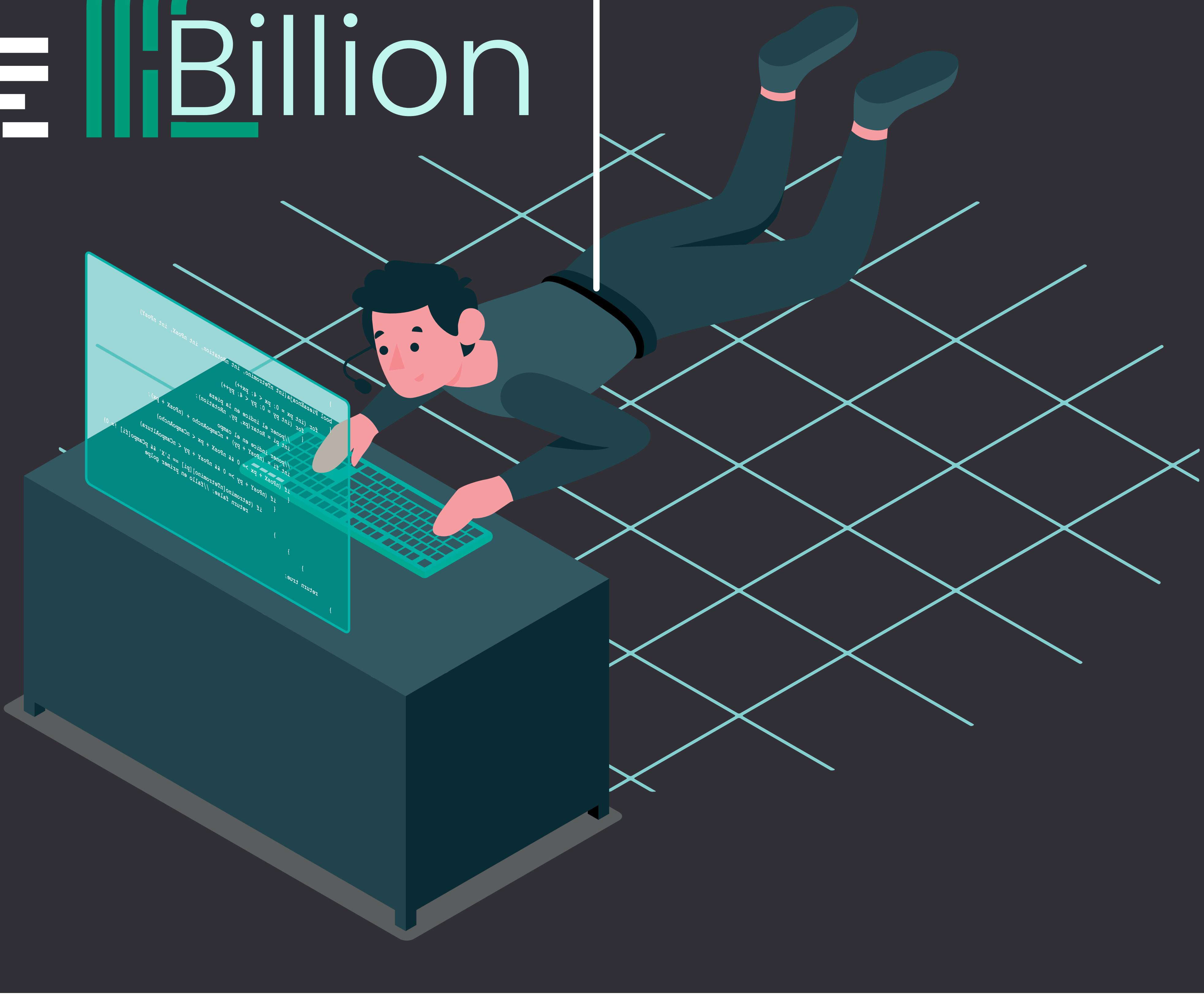
www.commitstrip.com



According to the CEO
"They did not know it was impossible so they did it"
Sorry for quoting Mark Twain, but yes, that's really what happened! Haha!
According to the Project Manager
Thanks to our outstanding team coordination and hard work, we've shown that our team can deliver on big projects! Well done!
According to the CTO
We successfully used new technologies within an agile process stream, whilst guaranteeing world-class security and scalability...
According to the dev
Ugh... yet another project where the specs change every day with no clear timescales, and they had me work using an undocumented home-grown framework....
...I'm gonna open a food truck
CommitStrip.com

CommitStrip is a daily strip recounting funny anecdotes of life as a coder mixed up with a dash of topical tech news. Find more comics here: ***www.commitstrip.com***

22 Billion



According to the statistics published by **https://techjury.net/**, in 2021 there have been 22 billion breached records.

Considering that everything connected to the internet is a potential target to hackers, no wonder that, globally, 30,000 websites are hacked daily.

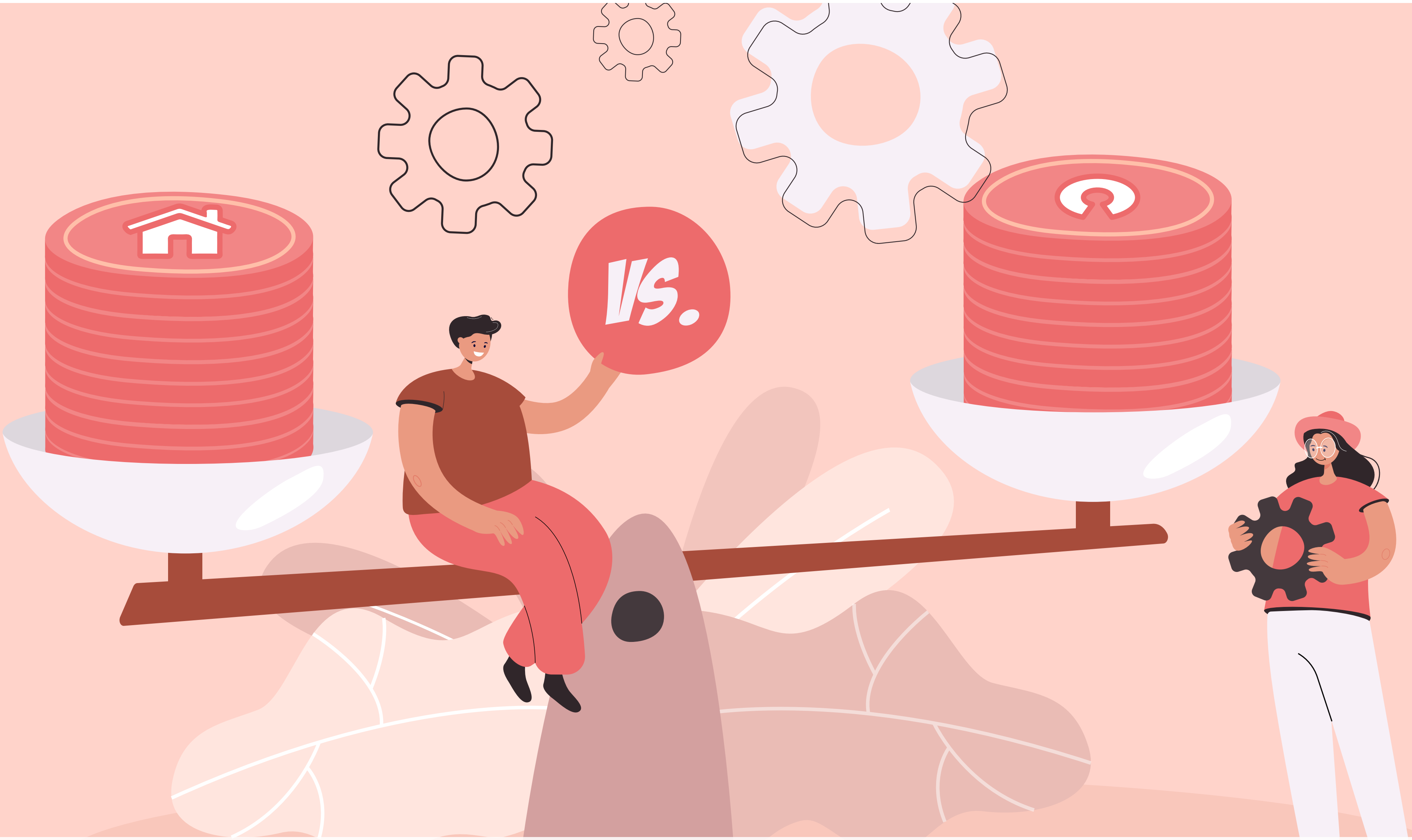
Cyber-attacks are becoming more complex and more creative each year. Years ago, stealing data for financial gain was the main motive for cyber-attacks. The motive has changed recently: the attacks have disruption of services as their main motive.

Many causes can make data breach possible: from weak or stolen credentials, to application vulnerabilities, insider threats, improper configuration and human error. Statistics revealed that insider threats caused 60% of all data breaches in one year.

Maybe you are asking yourself "How we can prevent this?". There are different solution but the most simple one is "start by educating your staff, your colleagues, your co-workers". Educate people how to recognize common social engineering tactics, how to recognize when an attack is taking place, how to code to minimize the number of security vulnerabilities a system exposes. It will cost much less than a data breach.

Franck Parchot

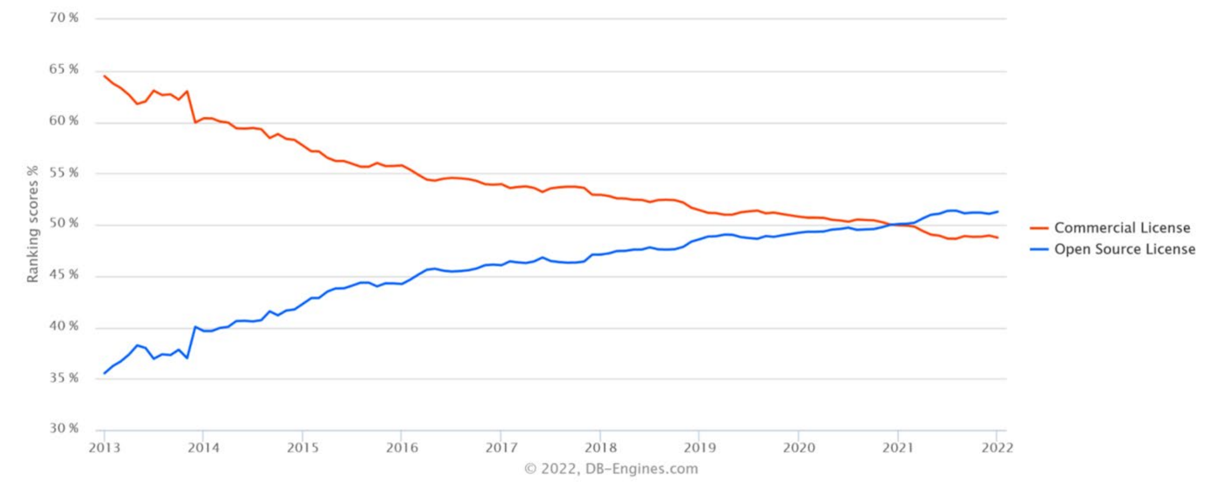
Open Source Databases versus Proprietary Databases – a comparison



When I started working with databases, more than 20 years ago, the operating systems were all commercial UNIX: IBM AIX, HP-UX, SUN Solaris, SCO Unix. Today, it is all Linux: Open Source Software has won for the enterprise operating systems. Are we going to see the same way for databases?

Open Source databases are increasingly popular, especially for new development projects. In many cases, Oracle, SQL Server and Db2 will stay for the legacy applications, and open source is considered for new applications, or when old ones are replaced by new (micro) services. However, some big companies have migrated the monoliths to PostgreSQL or MySQL. Those two are the most popular for relational databases with ACID properties, but MongoDB, Redis, Elasticsearch also initially came as open source databases for specialized use cases. The databases I've mentioned are those that are in the top popularity by the "DB ranking engine" and they sum up as the following commercial vs. open source trend:

Popularity trend



The above chart shows the historical trend of the popularity of open source and commercial database management systems.

https://db-engines.com/en/ranking_osvsc

Without taking too much out of this ranking, it shows the tendency. You can look at the DB Engines report, or other sources, for the details. In relational databases, the fastest growing are PostgreSQL and, at a lower level, MariaDB. The NoSQL ones, with exponential increase of popularity, are MongoDB, ElasticSearch, Redis, Cassandra. Among the most used multi-purpose databases, with stable popularity, already at the top for years, we find the well-known commercial ones, and one open-source player: MySQL.

Past and present

What about the future of the SQL databases? Will it be similar to the move from UNIXes to Linux? I have no crystal ball to predict the future. But in the present, we can observe a similar trend with what happened with the operating systems: more enterprise features seen in open source databases, and more critical applications using them. Enterprise level features, like security, partitioning, parallelism, are added to open source databases. Or, for operations, better instrumentation, troubleshooting tools, performance and resilience options. But the context is different. RedHat was a major player for Linux. Is there, in the DB area, a company with the same model they had when distributing the Open Source Linux with Enterprise additions, service and support? And times have changed. This happened for Linux when on-premises was the rule, with commodity hardware coming into the datacenter. Today, the cloud providers change the rule: they can reduce their cost with open-source, and provide a commercial service on top of it. So, basically, looking at what happened with Linux can happen to databases, but with no guarantee.

Rather than predicting the future, I'll look at the pros and cons of using open-source databases, versus proprietary ones. And some may be surprising. Even if I try to cover all concepts, my examples come from my personal experience. I've been working with proprietary databases for more than 20 years. Some early years with DB2, some firefighting on Sybase and Informix. And then most of the time I've been working with the Oracle Database, in development or operations teams.

I opened my area of expertise to PostgreSQL, and some AWS Database services, a few years ago, before looking at the new cloud native databases. I am now a Developer Advocate at Yugabyte. YugabyteDB is a distributed SQL database, built for OLTP workloads, the area where traditional commercial databases have grown their business in the past. YugabyteDB is a recent one (founded in 2016), which started with a mixed model of open-source core, and commercial features for enterprises, before taking the decision to be 100% Open-Source, in 2019.

It is a good example for this trend of going open-source. As an Oracle ACE Director, I am also in close contact with the Oracle community, where, besides the well-known proprietary database, there are some open-source ones like MySQL and NoSQL. I'm also an AWS Hero, the community around the Amazon Cloud, where many commercial services are built on top of open-source databases.

Free

There are several aspects of Open Source software. Let's tackle first the false idea that "open source is free ". Of course, if the source is available, you can download it, compile it and use it without paying anything. This is okay for development, the Big Tech did it to provide their cloud services, but this is not feasible for the common companies, to run production. And we are talking about enterprise databases here. What will you do when you encounter an issue or when a security leak is exposed? Will you modify the source yourself, recompile and deploy? Of course not! You need a support contract, you need consultants, or you need dedicated teams. This is not free.
Interestingly, you can do nearly the same with commercial software. All commercial databases have a free version, easy to install in a container. This is different from UNIX where it is still difficult to get a free environment. If you want to start a new database application, you will not see any "open-source vs. proprietary" difference when doing a **`docker pull mcr.microsoft.com/mssql/server`**, **`docker pull ibmcom/db2`** or **`docker pull gvenzl/oracle-xe`**, versus **`docker pull postgres`**, **`docker pull mysql`** or **`docker pull yugabytedb/yugabyte`**. And when you go to production, you will probably get a commercial license or support subscription, with a commercial company, in any case. Open Source is not always free, and Commercial is not always expensive. Of course, there are good chances that the cost is lower with open-source, but this is more about the freedom of choice to avoid vendor lock-in.

Freedom

Open Source is not free, but it gives definitely more freedom. Your core business stores and processes all its data on Oracle Database, IBM DB2 or Microsoft SQL Server. What will you do if those companies decide to discontinue their software? Although possible, it's generally unacceptable to run your company's core business with software on which you cannot fix bugs or apply new security patches.
A database vendor stopping its product is an extreme situation, and should not happen anytime soon. But, there are real cases where people suffered from vendor lock-in. Remember when Oracle 12.1.0.2 was out and we had no clue if Standard Edition was going to be discontinued. No news for more than one year, to finally get a Standard Edition 2 with more limitations than the one you bought the license for. Of course, people could upgrade to Enterprise Edition, but the cost is exponential.

Oracle changed other technical and license rules. Between 18c and 19c, RAC was suppressed from Standard Edition and a major feature for event sourcing, the continuous log miner, has also been removed with 19c. And, in one night, Oracle has doubled the price on their cloud competitors with a simple sentence ("When counting Oracle Processor license requirements in Authorized Cloud Environments, the Oracle Processor Core Factor Table is not applicable") added to the pricing document. "Authorized Cloud Environment" is where Oracle Database can be run, currently AWS and Azure. Others, like Google Cloud, are simply not authorized by licensing metrics. The document cannot be reproduced but the changes are visible in webarchive.org between **https://web.archive.org/web/20170110141657/http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf** and **https://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf**

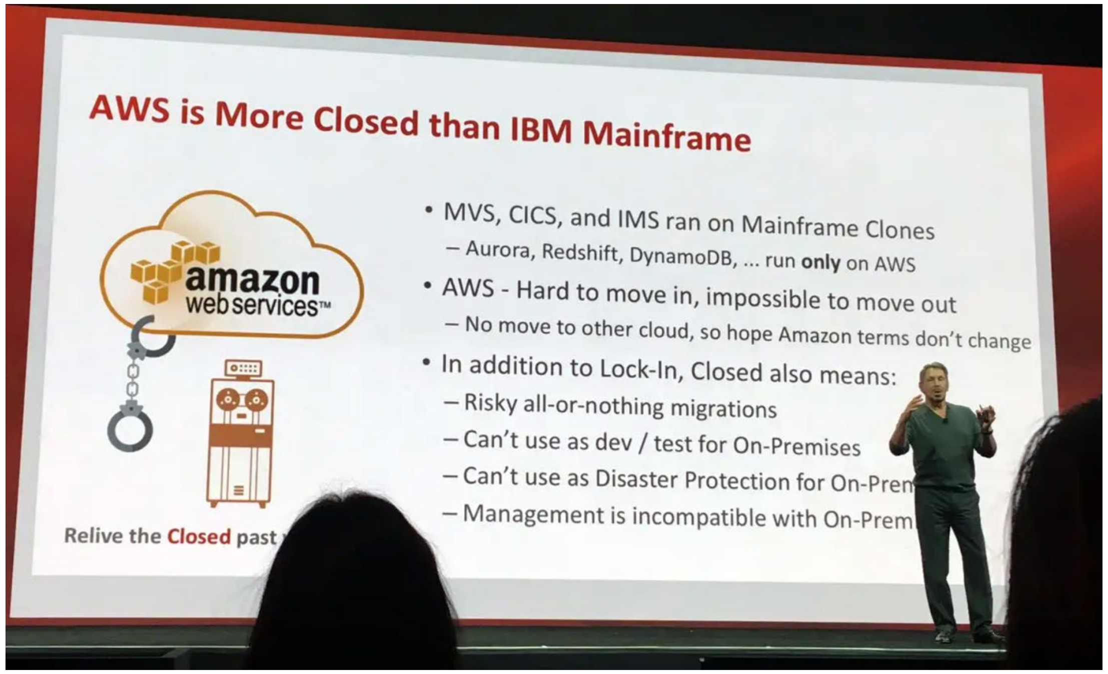
This is the real meaning of commercial software: the decisions are taken from a commercial point of view. In this example, the vendor, Oracle Corp, also sells GoldenGate for logical replication, and quite likely removed the change data capture features from the database to get their customer to buy another product. And, because it is closed source, you cannot do anything about it. You cannot add a feature yourself like you can do with open source. And to be able to announce at OpenWorld that their cloud is 2x cheaper, they changed the processor count, for the same processors on their competitors' platforms.

On the contrary, an open source database protects you from high-pricing, allows many companies to offer support, hosting or managed services, and gives you the freedom to move out. The code to access your data is open, to use or build, a migration tool, if it doesn't already exist. Even if the project closes, and nobody forks it to continue, you can still use it, patch it, until you migrate to something else. When the founder of MySQL wanted to be independent from Sun and Oracle, he forked to MariaDB. They are now two different databases, giving more choice, and more freedom.

I depicted the dark side of commercial decisions. They also have some benefits for the users. For example, Oracle has implemented complex features, like optimizer transformation, because important customers asked for it. An open source community would have probably refused the change, not willing to increase their code complexity when the problem should be solved by re-writing the queries in the application. With proprietary or open-source software, having a commercial company behind has its advantages for enterprises.

Observability

Your data is critical for your company and you want to be sure that it is processed and persisted without error. Do you trust a blackbox? If your database software missed a write, are you sure you will detect it? If you detect it years later, are you sure you will be able to fix it? I think that what AWS did with the distributed storage of Aurora or DynamoDB is very smart. They say that they never lost data. I trust them, but... how do they know it? Do they compare reads across replicas like Oracle "Lost Write detection" does?



AWS is More Closed than IBM Mainframe

- MVS, CICS, and IMS ran on Mainframe Clones
  - Aurora, Redshift, DynamoDB, ... run **only** on AWS
- AWS - Hard to move in, impossible to move out
  - No move to other cloud, so hope Amazon terms don't change
- In addition to Lock-In, Closed also means:
  - Risky all-or-nothing migrations
  - Can't use as dev / test for On-Premises
  - Can't use as Disaster Protection for On-Pren
  - Management is incompatible with On-Prem

Relive the Closed past

Funnily, even closed source database vendors can advocate for the freedom to move out. Here is Larry Ellison at Oracle Open World 2016 about platform lock-in.

The only thing I know about AWS databases is what they put in the documentation, or in re:Invent talks. And that is only a little information. With open source software, if I wonder how they write to a storage device, and if they fsync() correctly, I can look at the code, and analyze how this works in my context (my OS, my filesystem, my disks). Note that Open Source helps it, showing the code and the comments. But you can have good observability on commercial software too. I trust how Oracle processes my data because, beyond what is documented, it can be observed with Linux tools, and there's a ton of bloggers who explained how it works.

The source is not open, but the C symbols are not stripped in the oracle binaries. Some code is wrapped in PL/SQL procedures, but this is just obfuscation, easy to unwrap. Except the comments, unfortunately.

The code, the logic, can be discovered in any software by attaching a debugger. But reading the comments in the code is a real added value from open source software.

For a good understanding of how it works internally, an 'old' proprietary database is acceptable. I've never felt the need to see the Oracle Database code but found all info from the community, or my own tracing, because it has been there for decades and used all around the world. But for a new database, I would definitely have more confidence in open source. I know how the marketing can push on new feature releases and, without showing their code, database engineers may be tempted to take some shortcuts on the code quality. When you expose your source, you cannot hide anything. This is true also for potential security issues. Everybody can verify the code. I'll come back to the decision of 100% open source for YugabyteDB but, being a new database, it has to gain trust by showing everything: the architecture design, the code, the roadmap.

Community

I mentioned community and, here again, I don't see a massive difference in user community. There are a lot of free resources on both sides. I've learned a lot about Oracle by reading Tom Kyte's AskTOM years before the Developer Advocate role became popular. I've got great discussions with Oracle or AWS product managers. I've got great discussions with PostgreSQL users and contributors. There are smart and passionate people everywhere. There's no bright or dark side.

And there are also political fights everywhere as well. Even in a community that is not driven by money, the ego can show the same unpleasant aspects. And, anyway, there are often commercial companies behind the open source contributions. The power of open source is that the decisions are not solely commercial, but also not only academic or idealistic. I see that with PostgreSQL: the core team may reject some evolutions, like hints or wait events, because it doesn't fit their goal. But some companies need a more pragmatic approach and can implement them with an extension. Some share the additions with a similar open source license, like Zalando Patroni for resilience. Some others keep their additions for their commercial services, like AWS Performance Insight for observability. The PostgreSQL open source license allows that.

Open Source: free software, tiered model, copyleft

I mentioned in the introduction that there is one major difference today if we want to compare open source databases to the acceptance of open source Linux. Cloud providers build managed services to sell their infrastructure with an added value service. AWS has taken many open source databases from the community: Aurora reuses MySQL and PostgreSQL. Redshift reuses PostgreSQL, DynamoDB storage is based on MySQL's InnoDB. They take it, add interesting features, but don't give back to the community. Even worse, with MongoDB, Kafka, Elasticsearch, AWS has built database services that compete with the companies who develop the open source software.

Because of this, those companies have stepped back in their open source licenses. MongoDB relicensed their open-source core to the copyleft "SSPL" so that it is impossible to use their code for a managed service. Elastic and Confluent did similar moves. CockroachDB also re-licenced to the same license as MariaDB. MySQL, with the GPL license, also has some restrictions when it comes to embedding it with proprietary software, like for a managed service. Those are not Open Source, according to OSI's Open Source Definition. And this limits the adoption and contribution by the community. In addition to that, those databases provide only a subset of their features for free. Enterprise features, like backups, scalability, encryption come with a commercial license.

Which databases remain that are fully Open Source? There are not a lot, and PostgreSQL is the most popular one. This is important for the user community, and the ecosystem around the database. YugabyteDB is also fully open source (Apache 2.0) and this decision often raises the following question, quoting here from Kosseila Haddalene (ACE

Associate): "Is this considered a threat if cloud providers would use YugabyteDB as a service at a larger scale"? The tradeoff between the community, and this risk of cloud providers using it, is clear: building a database today needs a community. Being fully open source allows that. Then, how to mitigate the risk of a cloud provider re-using it in their commercial services?

The solution is building the best cloud offer, and partnering with the cloud providers to deploy it on their infrastructure. In the future, the cloud providers will probably focus on the infrastructure, and the managed services will be run by the software companies. The users know that the main contributor to the database software is probably the best choice to give them support.

Responsibility

Of course, you can take an open source database and run it without any support subscription. You will get help from the community, and when the open source database has a great community, you can have very good help. An example, for PostgreSQL, you have the mailing lists, slack channel, telegram, and of course Stack Overflow. For YugabyteDB we have a forum, slack channel, and smart Developer Advocates ;)
But, even if many people help, you still hold the full responsibility: perform all the changes yourself, have to own the change and the choice of change. With commercial support, you can open a Service Request and then the responsibility is transferred to the vendor. This is the old adage "Nobody ever got fired for choosing IBM", and this applies to Oracle or any well-established software vendor. This is why you still need commercial support even with a 100% free database.

A recent anecdote (**https://twitter.com/bagder/status/1484672924036616195**) was a large company asking the non-profit open source contributors to assess and solve the Log4j security issue for them, as if they were their supplier. I've said that many things are not different in commercial and open source, but the mindset is different. The code is given for free but, if you don't accept all responsibilities, you probably need commercial support to use it at enterprise level.

In summary

If your manager thinks that open source = free, you better stay on well established commercial software, or you will have all responsibility on your shoulders when something gets wrong.

However, if the management understands that the savings on a commercial license can be invested in support, training and team, you will get more freedom, trust, and control over the software that processes your data. Even when a commercial company is behind, you should feel it as a partnership more than a subscription.

In the database area, the DBAs are more conservative, like were the sysadmins when Linux came to the enterprise. However, it would be surprising today to find a company with only commercial databases. On the other hand, some large companies have all their data on open source databases, and have proven that it works. The students are taught on MySQL and PostgreSQL. I'm convinced that a DBA today should know at least one open source database in addition to the Oracle, SQL Server or Db2 they use on a daily basis. And also that commercial database vendors take a risk when underestimating the features of open-source databases. Many have undeniably proven their value in all major cloud providers, and are invading the enterprise database space.

About Franck Pachot

Franck Pachot is Developer Advocate for YugabyteDB, an open-source distributed SQL database, with PostgreSQL compatibility. He has been working, as a consultant, with commercial databases (Oracle Certified Master, AWS Database Specialty certified) and loves to learn and share in the database community (AWS Data Hero, Oracle ACE Alumni) for years.
He blogs on **https://dev.to/franckpachot** and discuss on **https://twitter.com/FranckPachot**

Blockchain tables in Oracle Database

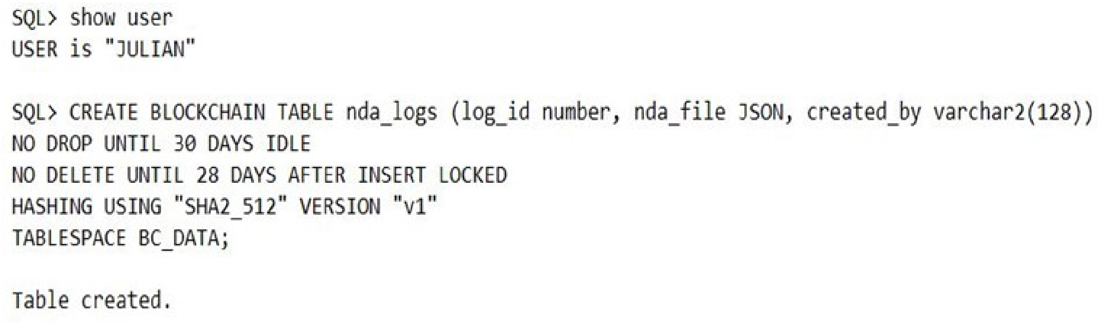
Julian Dontcheff

Blockchain tables are insert-only tables that organize rows into a number of chains and is a new concept that started with Oracle 20c (and now in 21c) but backported already to 19.10 where the database requires patch 32431413 and the COMPATIBLE parameter should be set to 19.10.0 or later. In 19.11-19.13 the additional patch is not needed.

Each row in a chain, except the first row, is chained to the previous row in the chain by using a cryptographic hash. For each Oracle RAC instance a blockchain table contains thirty-two chains, ranging from 0 through 31.

ORACLE DATABASE 20C EE HIGH PERF RELEASE 20.0.0.0.0 – PRODUCTION VERSION 20.2.0.0.0

This is an example of ***how a blockchain table is created*** *in 21c:*



```
SQL> show user
USER is "JULIAN"

SQL> CREATE BLOCKCHAIN TABLE nda_logs (log_id number, nda_file JSON, created_by varchar2(128))
NO DROP UNTIL 30 DAYS IDLE
NO DELETE UNTIL 28 DAYS AFTER INSERT LOCKED
HASHING USING "SHA2_512" VERSION "v1"
TABLESPACE BC_DATA;

Table created.
```

The "no drop" clause means that the table cannot be dropped for 30 days, and only it is was idle. Be extremely cautious as probably one day you might want to drop the table and even "drop user … cascade" will fail if you have not set the number of days or the table is never idle within the time slot you have set. If you are just testing the functionality, set the number of days to a very low number, even zero.

The "no delete" clause means that records from the table cannot be delete for 4 weeks after insertion and the number 28 is "locked" meaning you cannot change it later. This is a way to protect the rows from being deleted.

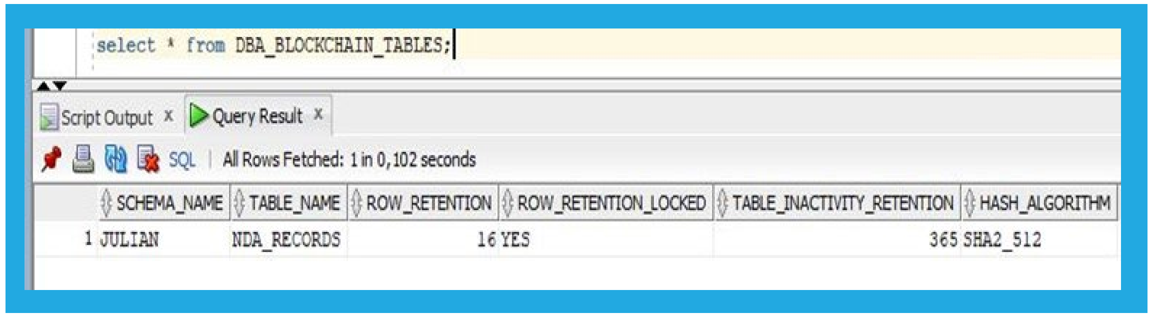
The blockchain hash algorithm, and data format clause is fixed in the current release. Most likely Oracle will allow other hashing algorithms in the future releases.

Let me next point out the main restrictions on blockchain tables:

- Blockchain tables cannot be created in the root container and in an application root container: ORA-05729: blockchain table cannot be created in root container
- You cannot update the rows: ORA-05715: operation not allowed on the blockchain table
- In general, you cannot delete rows, truncate the table, or drop the blockchain table: ORA-05723: drop blockchain table NDA_RECORDS not allowed
- Don't even try to drop the tablespace containing blockchain tables, here is what happens:

DROP TABLESPACE BC_DATA INCLUDING CONTENTS AND DATAFILES;
ORA-00604: error occurred at recursive SQL level 1
ORA-05723: drop blockchain table NDA_RECORDS not allowed

The most important new view in 20c related to blockchain tables is DBA_BLOCKCHAIN_TABLES:



```
select * from DBA_BLOCKCHAIN_TABLES;
```

All Rows Fetched: 1 in 0,102 seconds

	SCHEMA_NAME	TABLE_NAME	ROW_RETENTION	ROW_RETENTION_LOCKED	TABLE_INACTIVITY_RETENTION	HASH_ALGORITHM
1	JULIAN	NDA_RECORDS	16	YES	365	SHA2_512

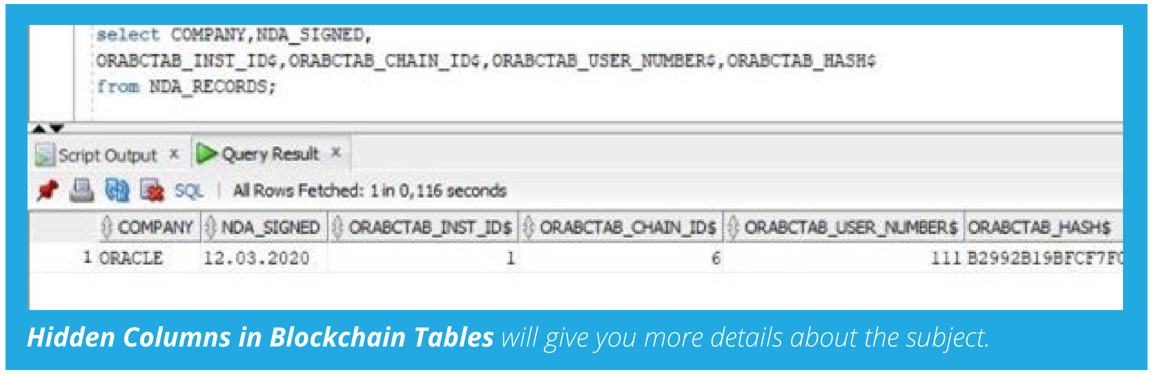
The 4 (non-trivial) columns of **DBA_BLOCKCHAIN_TABLES** contain the following information:

1. ROW_RETENTION: The minimum number of days a row must be retained after it is inserted into the table – if the value of this column is NULL, then rows can never be deleted from the table. In the example above, the row can be deleted after 16 days. Otherwise, you will get: ORA-05715: operation not allowed on the blockchain table
2. ROW_RETENTION_LOCKED: 2 possible values (YES and NO) showing if the row retention period for the blockchain table is locked.

   YES: The row retention period is locked. You cannot change the row retention period.
   NO: The row retention period is not locked. You can change the row retention period to a value **higher than the current value** with the SQL statement ALTER TABLE … NO DELETE UNTIL n DAYS AFTER INSERT.

3. TABLE_INACTIVITY_RETENTION: Number of days for which the blockchain table must be inactive before it can be dropped, that is, the number of days that must pass after the most recent row insertion before the table can be dropped. A table with no rows can be dropped at any time, regardless of this column value. In the example above, a year of inactivity must pass before the table can be dropped.
4. HASH_ALGORITHM: The algorithm used for computing the hash value for each table row.

To each row you add/insert to the blockchain table, Oracle adds values to the hidden columns of the blockchain table. Hidden columns are populated after you commit. They are used to implement sequencing of rows and verify that data is tamper resistant. You can create indexes on hidden columns. In order to view the values of the hidden columns, you should explicitly include their names in the SQL, just like this:



```
select COMPANY,NDA_SIGNED,
ORABCTAB_INST_ID$,ORABCTAB_CHAIN_ID$,ORABCTAB_USER_NUMBER$,ORABCTAB_HASH$
from NDA_RECORDS;
```

```
All Rows Fetched: 1 in 0,116 seconds

    COMPANY  NDA_SIGNED  ORABCTAB_INST_ID$  ORABCTAB_CHAIN_ID$  ORABCTAB_USER_NUMBER$ ORABCTAB_HASH$
  1 ORACLE   12.03.2020                  1                   6                    111 B2992B19BFCF7F
```

Hidden Columns in Blockchain Tables *will give you more details about the subject.*

The following additional operations are not allowed with blockchain tables:

- Adding, dropping, and renaming columns
- Dropping partitions
- Defining BEFORE ROW triggers that fire for update operations (other triggers are allowed)
- Direct-path loading
- Inserting data using parallel DML
- Converting a regular table to a blockchain table (or vice versa)

There is a new PL/SQL procedure DBMS_BLOCKCHAIN_TABLE which contains 5 procedures, one of which VERIFY_ROWS is used to **validate the data** in the blockchain table.

Use DBMS_BLOCKCHAIN_TABLE.DELETE_EXPIRED_ROWS to remove rows that are beyond the retention period of the blockchain table. There are guidelines from Oracle for managing Blockchain tables, for DBAs it is important to know that:

- For each chain in a database instance, periodically save the current hash and the corresponding sequence number outside the database.
- In an Oracle Data Guard environment, consider using the maximum protection mode or maximum availability mode to avoid loss of data.

You can use certificates to verify the signature of a blockchain table row. Check here on how to add and delete certificates to blockchain table rows.

If you use Oracle Data Pump with blockchain tables, then you can use only CONVENTIONAL access_method.

Blockchain tables can be exported only under the certain conditions:

- The VERSION parameter for the export is explicitly set to 21.0.0.0.0 or later.
- The VERSION parameter is set to (or defaults to) COMPATIBLE, and the database compatibility is set to 21.0.0.0.0 or later.
- The VERSION parameter is set to LATEST, and the database release is set to 21.0.0.0.0 or later.
- If you attempt to use Oracle Data Pump options that are not supported with blockchain tables, then you receive errors when you attempt to use those options.

The following options of Oracle Data Pump are not supported with blockchain tables. They result in errors when you attempt to use them to import data into an existing blockchain table.

ACCESS_METHOD = [DIRECT_PATH, EXTERNAL_TABLE, INSERT_AS_SELECT]
TABLE_EXISTS_ACTION = [REPLACE | APPEND | TRUNCATE]

CONTENT=DATA_ONLY results in error when you attempt to import data into a blockchain table

If you request departitioning using the option PARTITION_OPTIONS=[DEPARTITIONING | MERGE] with blockchain tables, then the blockchain tables are skipped during departitioning.

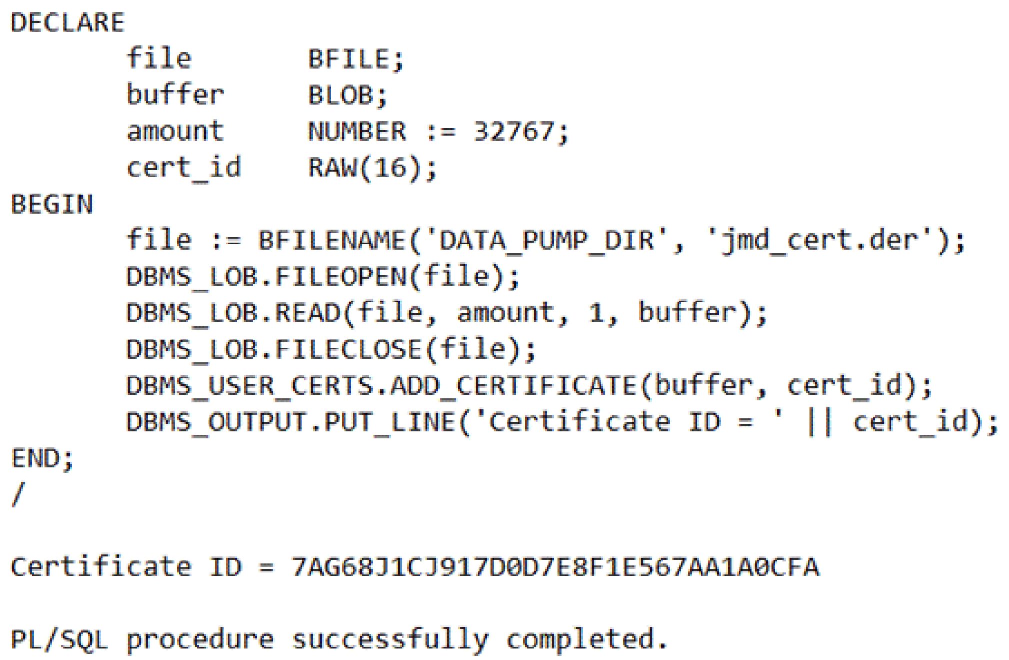
NETWORK IMPORT, TRANSPORTABLE, SAMPLE, QUERY and REMAP_DATA are not supported either.

You can use certificates to verify the signature of a blockchain table row.

You need to obtain an X.509 digital certificate from a Certificate Authority (CA). This certificate is added to the database, as a BLOB, and then used to add and verify the signature of one or more blockchain table rows. Multiple certificates can be used to sign rows in one blockchain table. Use OpenSSL APIs to manipulate digital certificates. The digital certificate to be added must be stored as a BLOB in the database. The BLOB can be within a directory object.

Use the DBMS_USER_CERTS.ADD_CERTIFICATE procedure to add a certificate.

When a certificate is added to the database, it is assigned a unique certificate ID. This ID is the output of the DBMS_USER_CERTS.ADD_CERTIFICATE procedure. The certificate ID is used when adding and verifying signatures for a blockchain table row. You must remember this certificate ID, else you cannot use the associated digital certificate.



```
DECLARE
       file       BFILE;
       buffer     BLOB;
       amount     NUMBER := 32767;
       cert_id    RAW(16);
BEGIN
       file := BFILENAME('DATA_PUMP_DIR', 'jmd_cert.der');
       DBMS_LOB.FILEOPEN(file);
       DBMS_LOB.READ(file, amount, 1, buffer);
       DBMS_LOB.FILECLOSE(file);
       DBMS_USER_CERTS.ADD_CERTIFICATE(buffer, cert_id);
       DBMS_OUTPUT.PUT_LINE('Certificate ID = ' || cert_id);
END;
/

Certificate ID = 7AG68J1CJ917D0D7E8F1E567AA1A0CFA

PL/SQL procedure successfully completed.
```

The PL/SQL procedure DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS verifies that rows in a blockchain table were not modified since they were inserted. Being tamper resistant is a key requirement for blockchain tables.

Use the DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS procedure to verify the integrity of the hash column in a blockchain table. If a row contains a signature, the signature can be verified.

You can validate all rows in the blockchain table or specify criteria to filter rows that must be validated. Rows can be filtered using the instance ID, chain ID, or row creation time.

In order to maintain the integrity of blockchain tables you need to continuously verify that the blockchain table data has not been compromised. Here is how to do it:

1. Verify the links between all the chains in the blockchain table by using the DBMS_BLOCKCHAIN_TABLE.VERIFY_ROWS procedure.

2. Generate a signature and signed digest for the blockchain table using the DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST function.

3. At another point in time, generate a signature and signed digest for the blockchain table using the DBMS_BLOCKCHAIN_TABLE.GET_SIGNED_BLOCKCHAIN_DIGEST function.

4. Verify the integrity of rows that were created between time T1 and T2 by running the DBMS_BLOCKCHAIN_TABLE.VERIFY_TABLE_BLOCKCHAIN procedure.

5. Repeat the process in Steps 2 through 4, at different time periods, to verify the integrity of rows inserted between different time periods.

It is good practice to verify of the integrity of blockchain table data on regular basis. This technique of continuous comparison and verification, between different periods of times, provides a guarantee that the rows in the blockchain table are not compromised.

Final note: you really have a good eye if you noticed the new 21c datatype I used in the table creation at the top of this blog post.

About Julian Dontcheff

Julian Dontcheff is the Global Database Lead of Accenture and has more than 30 years of database experience. He is the first Oracle Certified Master in Europe (back in 2002) and he is also an Oracle ACE Director.

Carsten Czarski

Oracle Application Express (Part 11): Faceted Search, Pivot, JSON and more: About APEX, SQL and Database Features

As APEX runs within the Oracle Database, it's obvious that components like reports, charts, or others, execute SQL queries to get and visualize data. But some APEX functionality, like the Pivot feature in Interactive Reports, or the Count Computation in Faceted Search are far beyond a simple ***"select * from table"*** *query. To provide these features, APEX makes intensive use of the Oracle Database and its SQL functionality.*

This article illustrates, how APEX uses SQL to efficiently query and compute results. Before looking into the concrete examples, the article explains how developers can use APEX Debug to find out which actual SQL Query or statement was executed for a given APEX component.

Primer: Review executed SQL with APEX Debug

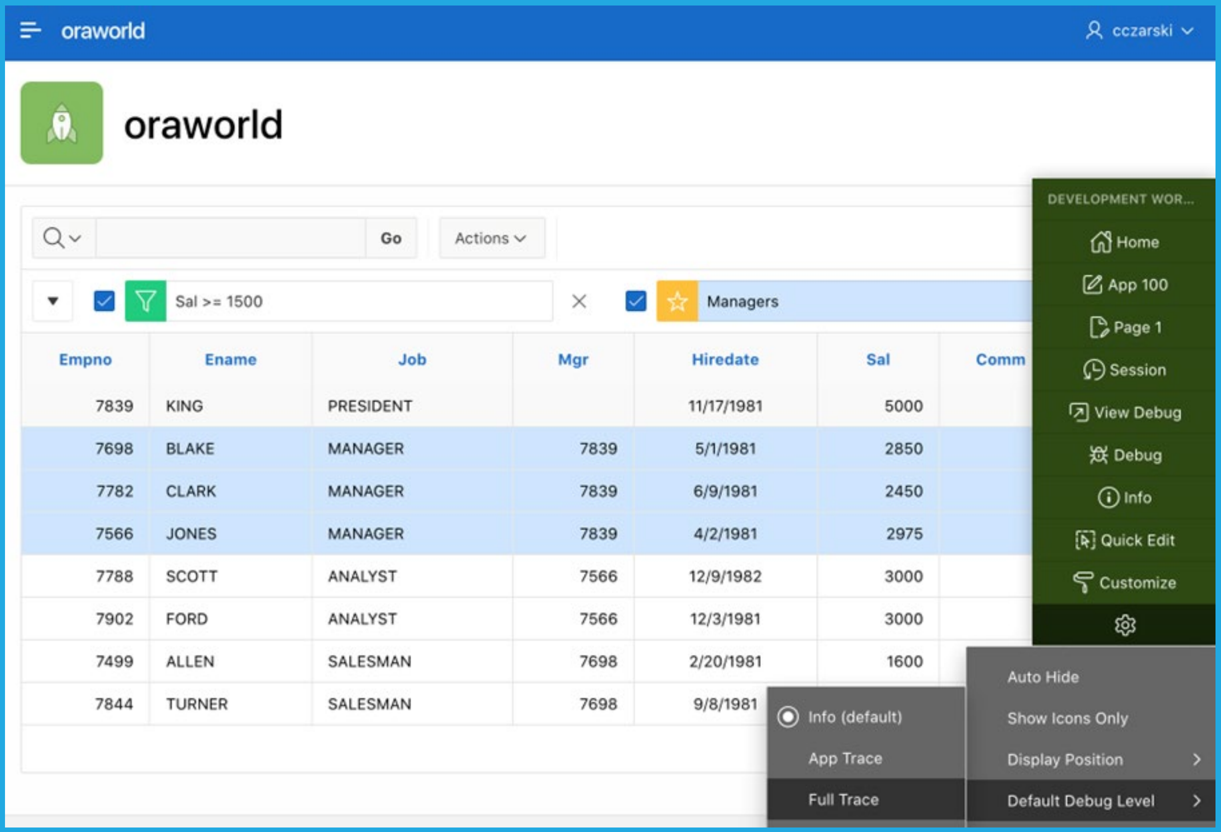
APEX Developers can always review all executed SQL or PL/SQL statements by utilizing **Debug mode**. By default, Debug mode runs in Level 4 ("*Info*"), which includes the basic SQL statements, among other information. Debug at Level 9 ("*Trace*") is far more verbose and even includes the *actual execution* plan for executed SQL queries. ***Fig. 1*** shows how to toggle the debug level using the Developer Toolbar.

Click on **Debug** or **Debug (Full Trace)** in the Developer Toolbar to enable debug mode. APEX will reload the page, and debug output will be logged while the page is rendered. The developer toolbar showing **No Debug** (to turn debug off again) is an indicator for debug mode being active.

All debug output is stored within the APEX repository. Click on **View Debug** in the developer toolbar to get an overview on available debug logs (***Fig. 2***). The first entry is typically most recent. There is also an APEX view: **APEX_DEBUG_MESSAGES**.

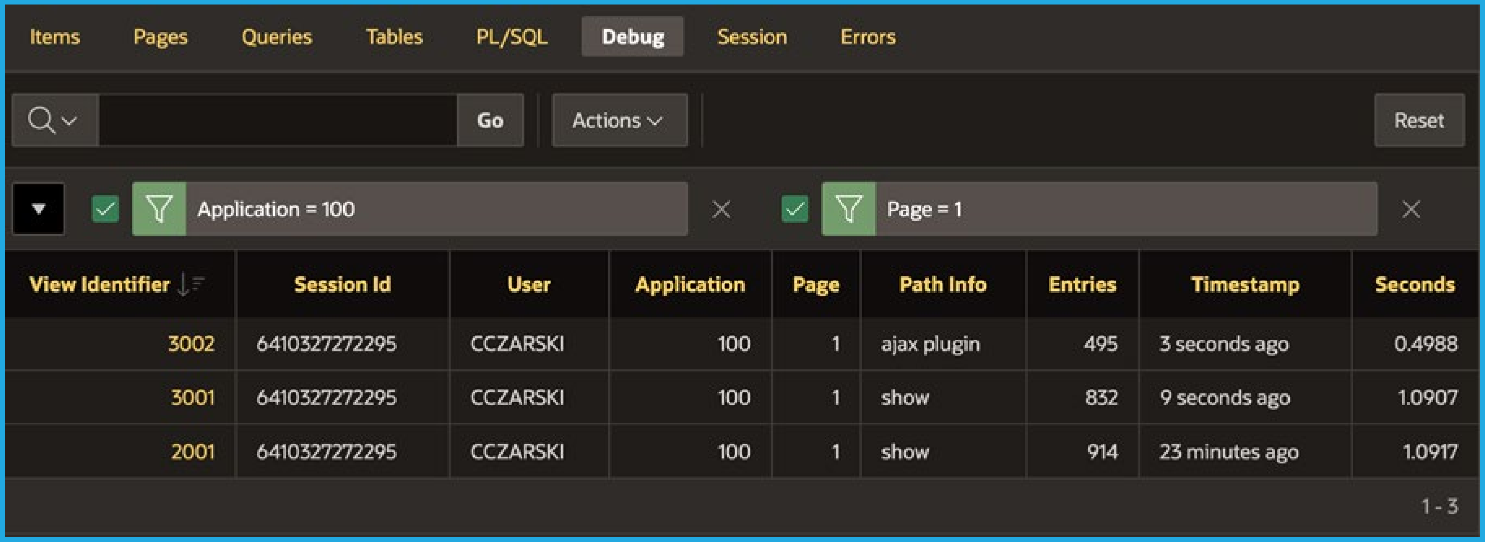
Note that AJAX requests generate their own debug output, which means that developers might need to review multiple debug logs, if page components (like Charts) execute AJAX requests.

Fig. 3 shows a snippet from a "Full Trace" debug log, which contains the actually executed SQL query (with execution plan) for an interactive report with some filters and highlights enabled. As "Trace" debug outputs contain a lot of information, developers need to scroll a bit to get to the interesting sections.



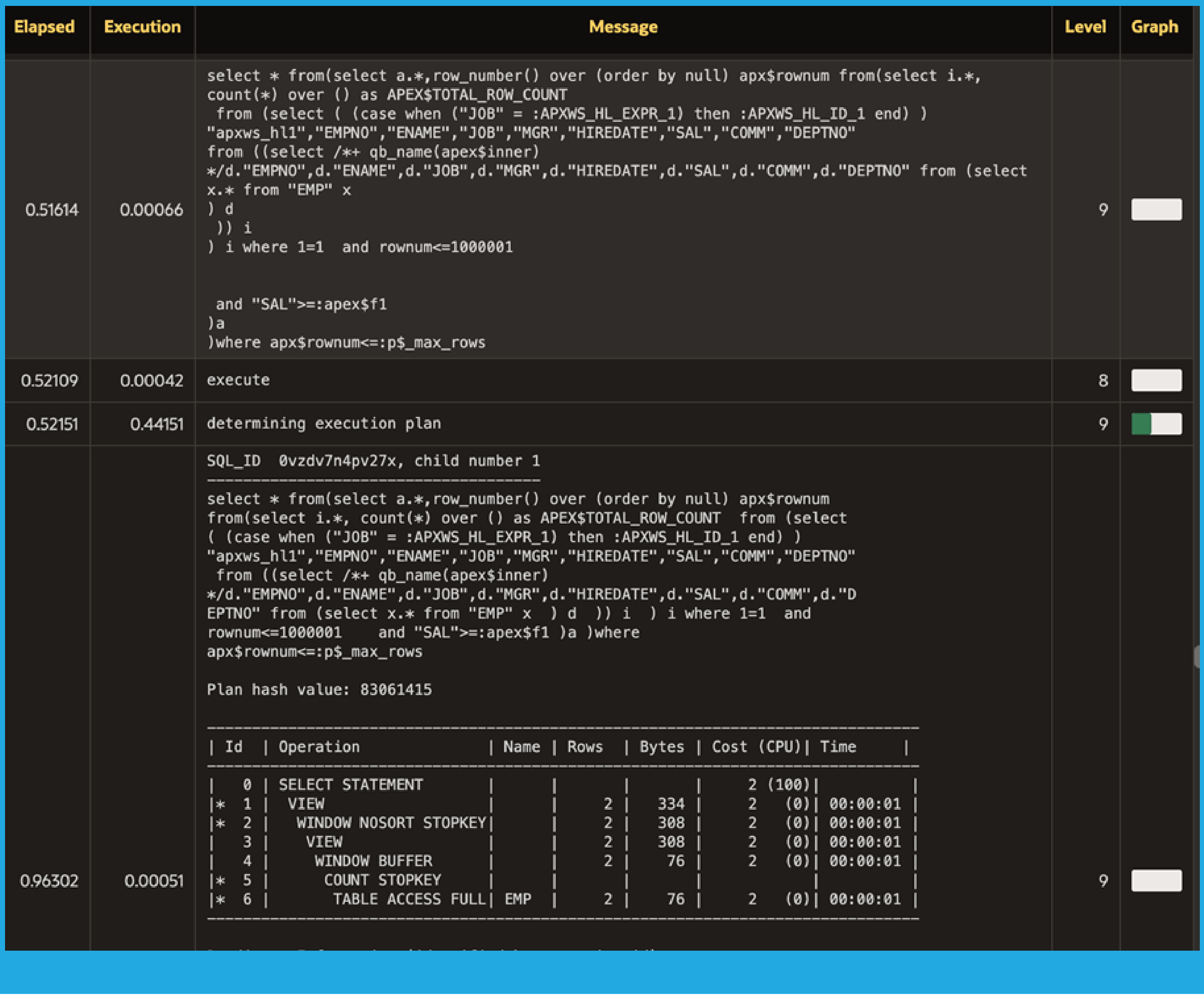
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
7839	KING	PRESIDENT		11/17/1981	5000	
7698	BLAKE	MANAGER	7839	5/1/1981	2850	
7782	CLARK	MANAGER	7839	6/9/1981	2450	
7566	JONES	MANAGER	7839	4/2/1981	2975	
7788	SCOTT	ANALYST	7566	12/9/1982	3000	
7902	FORD	ANALYST	7566	12/3/1981	3000	
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	
7844	TURNER	SALESMAN	7698	9/8/1981		

Fig. 1: Use the Developer Toolbar to toggle the debug level.



View Identifier	Session Id	User	Application	Page	Path Info	Entries	Timestamp	Seconds
3002	6410327272295	CCZARSKI	100	1	ajax plugin	495	3 seconds ago	0.4988
3001	6410327272295	CCZARSKI	100	1	show	832	9 seconds ago	1.0907
2001	6410327272295	CCZARSKI	100	1	show	914	23 minutes ago	1.0917

Fig. 2: List of available debug logs to review



```
Elapsed   Execution  Message                                                                                        Level  Graph
0.51614   0.00066    select * from(select a.*,row_number() over (order by null) apx$rownum from(select i.*,           9
                     count(*) over () as APEX$TOTAL_ROW_COUNT
                      from (select ( (case when ("JOB" = :APXWS_HL_EXPR_1) then :APXWS_HL_ID_1 end) )
                     "apxws_hl1","EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM","DEPTNO"
                     from ((select /*+ qb_name(apex$inner)
                     */d."EMPNO",d."ENAME",d."JOB",d."MGR",d."HIREDATE",d."SAL",d."COMM",d."DEPTNO" from (select
                     x.* from "EMP" x
                     ) d
                      )) i
                     ) i where 1=1  and rownum<=1000001


                      and "SAL">=:apex$f1
                     )a
                     )where apx$rownum<=:p$_max_rows
0.52109   0.00042    execute                                                                                          8
0.52151   0.44151    determining execution plan                                                                       9
0.96302   0.00051    SQL_ID  0vzdv7n4pv27x, child number 1                                                            9
                     -------------------------------------
                     select * from(select a.*,row_number() over (order by null) apx$rownum
                     from(select i.*, count(*) over () as APEX$TOTAL_ROW_COUNT  from (select
                     ( (case when ("JOB" = :APXWS_HL_EXPR_1) then :APXWS_HL_ID_1 end) )
                     "apxws_hl1","EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM","DEPTNO"
                      from ((select /*+ qb_name(apex$inner)
                     */d."EMPNO",d."ENAME",d."JOB",d."MGR",d."HIREDATE",d."SAL",d."COMM",d."D
                     EPTNO" from (select x.* from "EMP" x  ) d  )) i  ) i where 1=1  and
                     rownum<=1000001    and "SAL">=:apex$f1 )a )where
                     apx$rownum<=:p$_max_rows

                     Plan hash value: 83061415

                     -------------------------------------------------------------------------------
                     | Id  | Operation                | Name | Rows  | Bytes | Cost (CPU)| Time     |
                     -------------------------------------------------------------------------------
                     |   0 | SELECT STATEMENT         |      |       |       |     2 (100)|          |
                     |*  1 |  VIEW                    |      |     2 |   334 |     2   (0)| 00:00:01 |
                     |*  2 |   WINDOW NOSORT STOPKEY|      |     2 |   308 |     2   (0)| 00:00:01 |
                     |   3 |    VIEW                  |      |     2 |   308 |     2   (0)| 00:00:01 |
                     |   4 |     WINDOW BUFFER        |      |     2 |    76 |     2   (0)| 00:00:01 |
                     |*  5 |      COUNT STOPKEY       |      |       |       |            |          |
                     |*  6 |       TABLE ACCESS FULL| EMP  |     2 |    76 |     2   (0)| 00:00:01 |
                     -------------------------------------------------------------------------------
```

Fig. 3: Actual Interactive Report SQL Query, with execution plan.

Debug output is full of very useful information, not only get a better understanding about what an APEX component does. Besides that, it also emphasizes the fact, that *the SQL executed by APEX is different from the query entered by the developer in page designer.*

The examples in the following sections illustrate, which SQL functionality APEX leverages to provide selected features in Reports, Faceted Search, or for REST Data Sources. Of course, this article cannot cover all APEX components with all features but using Debug Mode allows developers to investigate any other APEX component or Plug-In on their own.

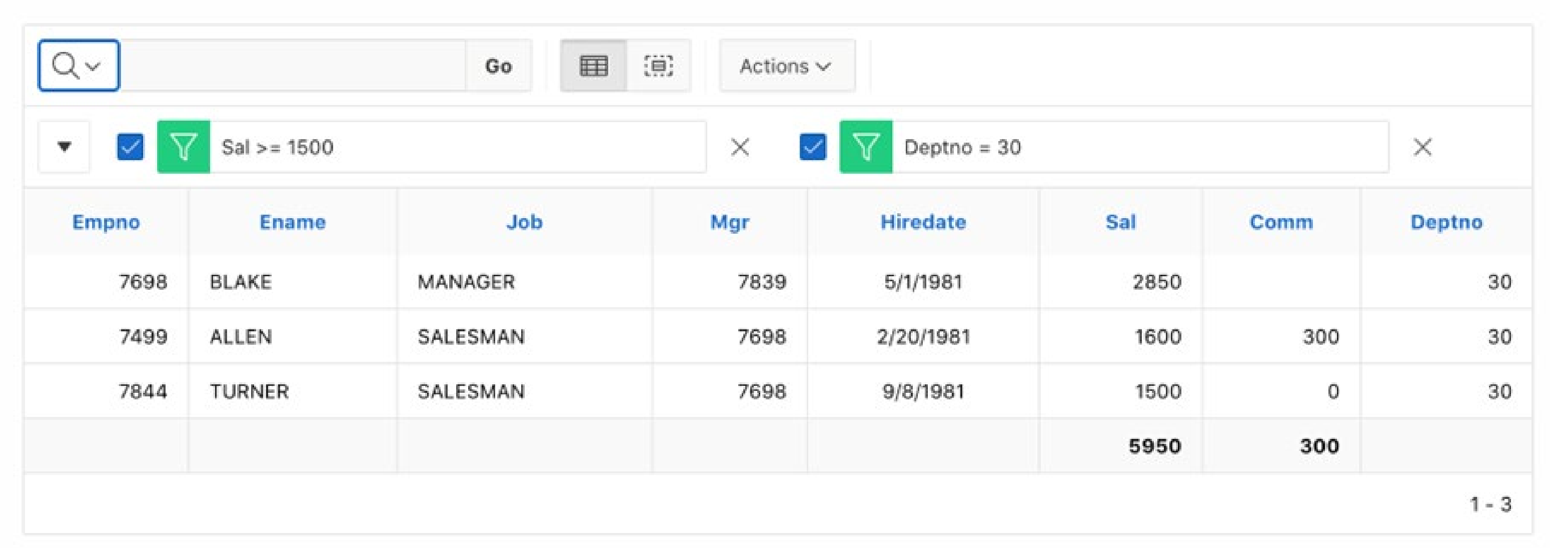
Report Aggregates

Aggregates are provided by all APEX report types: Interactive Report and Interactive Grid allows end users to configure aggregates like sum, average, or others; classic report allows the developer to display a sum under a report column. ***Fig. 4*** shows an Interactive Report with a sum aggregate, defined by the end user.

When computing aggregates, the first thought is probably about SQL aggregate functions and the GROUP BY function, for instance with a query like in ***listing 1***.

```
select sum(sal)  sum_sal,
       sum(comm) sum_comm
 from emp
where sal    >= 1500
 and deptno   = 30

             SUM_SAL             SUM_COMM
-------------------- --------------------
                5950                  300
```

Listing 1: GROUP BY SQL Query to determine aggregated values



☑ Sal >= 1500 ☑ Deptno = 30

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7698	BLAKE	MANAGER	7839	5/1/1981	2850		30
7499	ALLEN	SALESMAN	7698	2/20/1981	1600	300	30
7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30
					5950	**300**	

1 - 3

Fig. 4: Interactive Report with Aggregates on SAL and COMM columns

However, the Interactive Report in ***Fig. 4*** shows the actual rows and the aggregated values. Does that mean that APEX executes two queries? Reviewing debug output tells us, that this is not the case - APEX executes only one query. ***Listing 2*** shows what Interactive Reports do (the query is simplified to be better understandable).

```
select emp.*,
       sum(sal ) over () APEXWS_AGG_000,
       sum(comm) over () APEXWS_AGG_001
  from emp
 where sal    >= 1500
   and deptno  = 30

EMPNO ENAME  ...  SAL COMM DEPTNO APEXWS_AGG_000 APEXWS_AGG_001
----- ------ --- ---- ---- ------ -------------- --------------
 7698 BLAKE  ... 2850        30           5950            300
 7499 ALLEN  ... 1600  300   30           5950            300
 7844 TURNER ... 1500    0   30           5950            300
```

Listing 2: APEX uses analytic (window) functions to compute aggregates.

APEX uses analytic (or window) functions to compute the aggregates. The reason is that data rows and aggregates become part of the results, the latter as additional query columns. As APEX executes completely within the database (query results are processed by APEX PL/SQL code), the (tiny) additional cost of duplicating the aggregated value in each row can be neglected.

Aggregates are often used in combination with *Control Breaks*. Analytic functions can deal with these also, as ***Listing 3*** shows. The analytic function's PARTITION BY clause leads to the aggregate being computed for each DEPTNO value, and the additional ORDER BY clause makes sure that the rows are returned in the correct order. Based on the value change in the DEPTNO column, the Interactive Report component "detects" a new Control Break. Interactive Grids apply a slightly different technique to detect the control break (using the LEAD analytic function), but the aggregate calculation is similar.

```
select emp.*,
      sum(sal ) over (partition by deptno) APEXWS_AGG_000,
      sum(comm) over (partition by deptno) APEXWS_AGG_001
 from emp
 where sal >= 1500
 order by deptno
```

Listing 3: Aggregates and Control Breaks

Interactive Report PIVOT Functionality

Fig. 5 shows a *Pivot View*, which is another very interesting Interactive Report functionality: the values of the JOB column are projected as new columns; and rows are built based on values of the DEPTNO column. The actual column values in the pivot view are aggregated based on rows, columns, and the chosen aggregation function.

These pivot views are completely based on the SQL PIVOT clause, which is available since Oracle Database 11*g*. ***Listing 4*** shows its usage.

```
select deptno,
       apxws_pv1_pfc1, apxws_pv2_pfc1, -- generated column names
       apxws_pv3_pfc1, apxws_pv4_pfc1,
       apxws_pv5_pfc1
  from emp
       pivot(
          sum( sal ) pfc1 for job in (
                                      'ANALYST'   apxws_pv1,
                                      'CLERK'     apxws_pv2,
                                      'MANAGER'   apxws_pv3,
                                      'PRESIDENT' apxws_pv4,
                                      'SALESMAN'  apxws_pv5 ) )
```

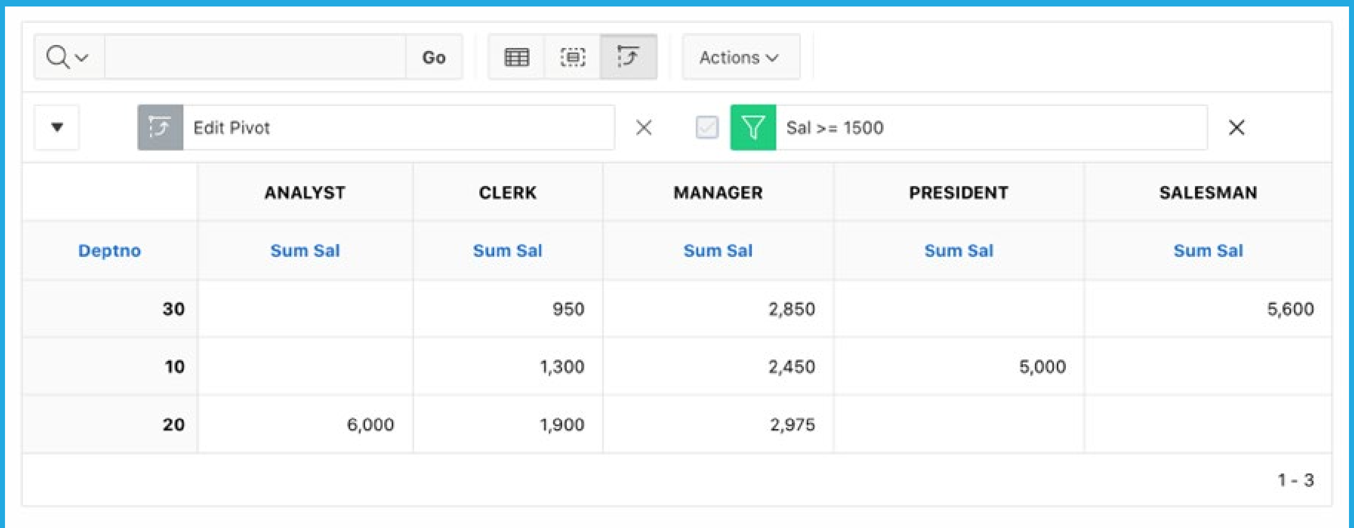
Listing 4: Interactive Report pivot view based on the SQL PIVOT clause

An interesting aspect is, that the *values of the JOB column* are part of the generated SQL PIVOT Query. To generate such a query, APEX needs to know these values. So, for a pivot view, APEX indeed executes two queries: one to get the values of the configured pivot column, the second to compute the actual pivoted values.

This can be seen in debug output as well. If turned on for an Interactive Report Pivot view, we can indeed see a **SELECT DISTINCT** query, which is executed first. Further down, debug output shows the PIVOT query, like in ***Listing 4***.

Faceted Search

Faceted Search and Smart Filters are further examples of how information can be computed in a very efficient manner, just by applying SQL functionality. ***Fig. 6*** shows a faceted search page. Note the facets on the left, which show the result count for each available facet item. These row counts are refreshed as soon as filters (and thus query results) change.



	ANALYST	CLERK	MANAGER	PRESIDENT	SALESMAN
Deptno	Sum Sal	Sum Sal	Sum Sal	Sum Sal	Sum Sal
30		950	2,850		5,600
10		1,300	2,450	5,000	
20	6,000	1,900	2,975		

1 - 3

Fig. 5: Interactive Report pivot view

But how are these values computed? Let's focus on the **Job** and **Department** facets. To get these values, an aggregation must be done for the JOB and DEPTNO columns, *independently from each other*. A simple GROUP BY or analytic function will not help, as these will always compute aggregates for the *combination of all group by columns*. ***Listing 5*** shows this.

One option would be to use one GROUP BY expression for each facet and to combine these GROUP BY queries with UNION ALL to get all the results (***Listing 6***).

However, the database still executes each part of the UNION ALL query individually, and then combines results. That does not matter when working with the tiny EMP table, but when working with larger tables, and more than just two facets, things will be different.

The good news is, that we don't need this UNION ALL construct; there is a better way: ***Listing 7*** illustrates the SQL **GROUPING SETS** clause, which allows us to get all the (independent) aggregations with a *single query*.

This clause was introduced with Oracle9*i* (together with ROLLUP and CUBE), so it's available for years and to everybody. Faceted Search and Smart Filters use GROUPING SETS to compute the result counts as efficient as possible.

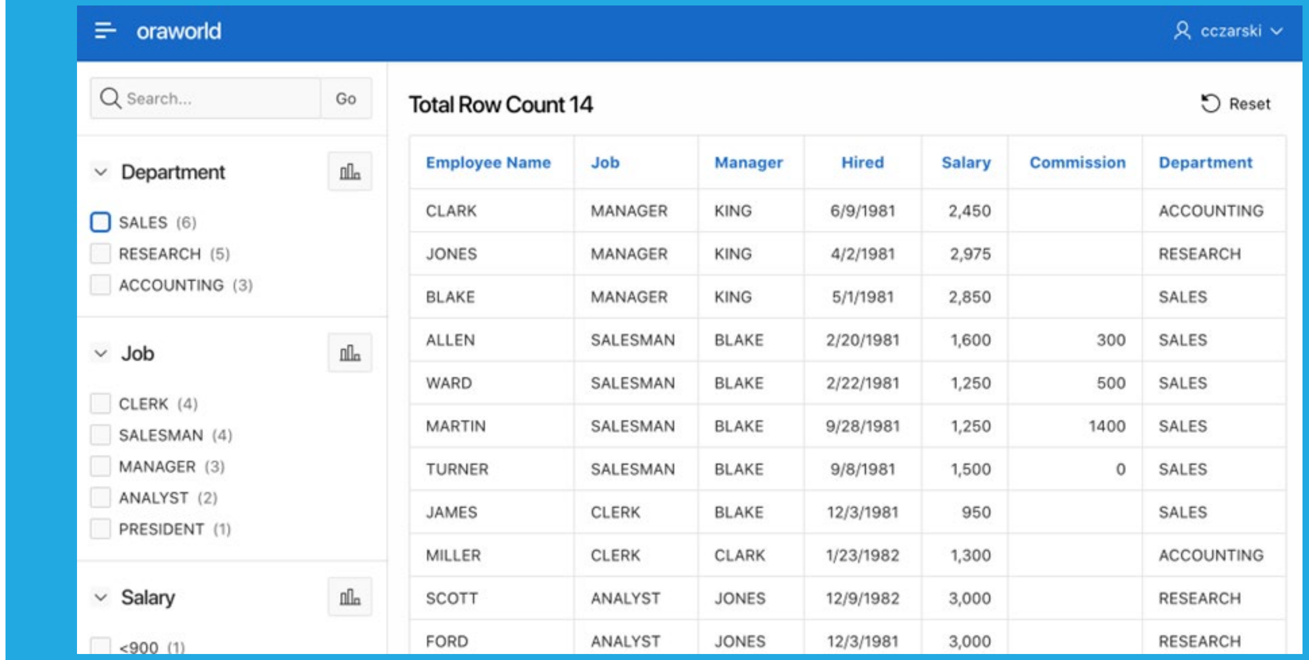
The GROUPING_ID function works in combination with GROUPING SETS and generates a unique identifier for each aggregation. APEX uses the grouping ID to assign the aggregated values (counts) to the correct facets. The overall row count is also computed with the same GROUPING SETS clause, this is what the "**()**" expression is for.

REST Data Sources

REST Data Sources allow to use external REST Services as data sources for APEX components, as they were plain tables. APEX stores meta data about the REST service, but also about its JSON response structure, as a *REST Data Source* in Shared Components. ***Fig. 7*** shows a *Data Profile* for a REST Data Source.

With the information from the data profile, APEX can generate a SQL query to parse the response JSON, and to provide the data as rows and columns, so that the APEX component is able to consume and render these (***Fig. 8***).

Reviewing debug output tells us that JSON parsing is not done with some procedural code (the APEX_JSON package is not used here), but with the native SQL/JSON function ***JSON_TABLE***, which was introduced with Oracle12c. Listing 8 illustrates this in simplified form.



Department

☐ SALES (6)
☐ RESEARCH (5)
☐ ACCOUNTING (3)

Job

☐ CLERK (4)
☐ SALESMAN (4)
☐ MANAGER (3)
☐ ANALYST (2)
☐ PRESIDENT (1)

Salary

☐ <900 (1)

Total Row Count 14

Employee Name	Job	Manager	Hired	Salary	Commission	Department
CLARK	MANAGER	KING	6/9/1981	2,450		ACCOUNTING
JONES	MANAGER	KING	4/2/1981	2,975		RESEARCH
BLAKE	MANAGER	KING	5/1/1981	2,850		SALES
ALLEN	SALESMAN	BLAKE	2/20/1981	1,600	300	SALES
WARD	SALESMAN	BLAKE	2/22/1981	1,250	500	SALES
MARTIN	SALESMAN	BLAKE	9/28/1981	1,250	1400	SALES
TURNER	SALESMAN	BLAKE	9/8/1981	1,500	0	SALES
JAMES	CLERK	BLAKE	12/3/1981	950		SALES
MILLER	CLERK	CLARK	1/23/1982	1,300		ACCOUNTING
SCOTT	ANALYST	JONES	12/9/1982	3,000		RESEARCH
FORD	ANALYST	JONES	12/3/1981	3,000		RESEARCH

Fig. 6: A Faceted Search page based on the EMP table

```
select deptno, job, count(*)
  from emp
 group by deptno, job

   DEPTNO JOB        COUNT(*)
 -------- ---------- --------
       10 CLERK             1
       10 MANAGER           1
       10 PRESIDENT         1
       20 ANALYST           2
       20 CLERK             2
        : :                 :
```

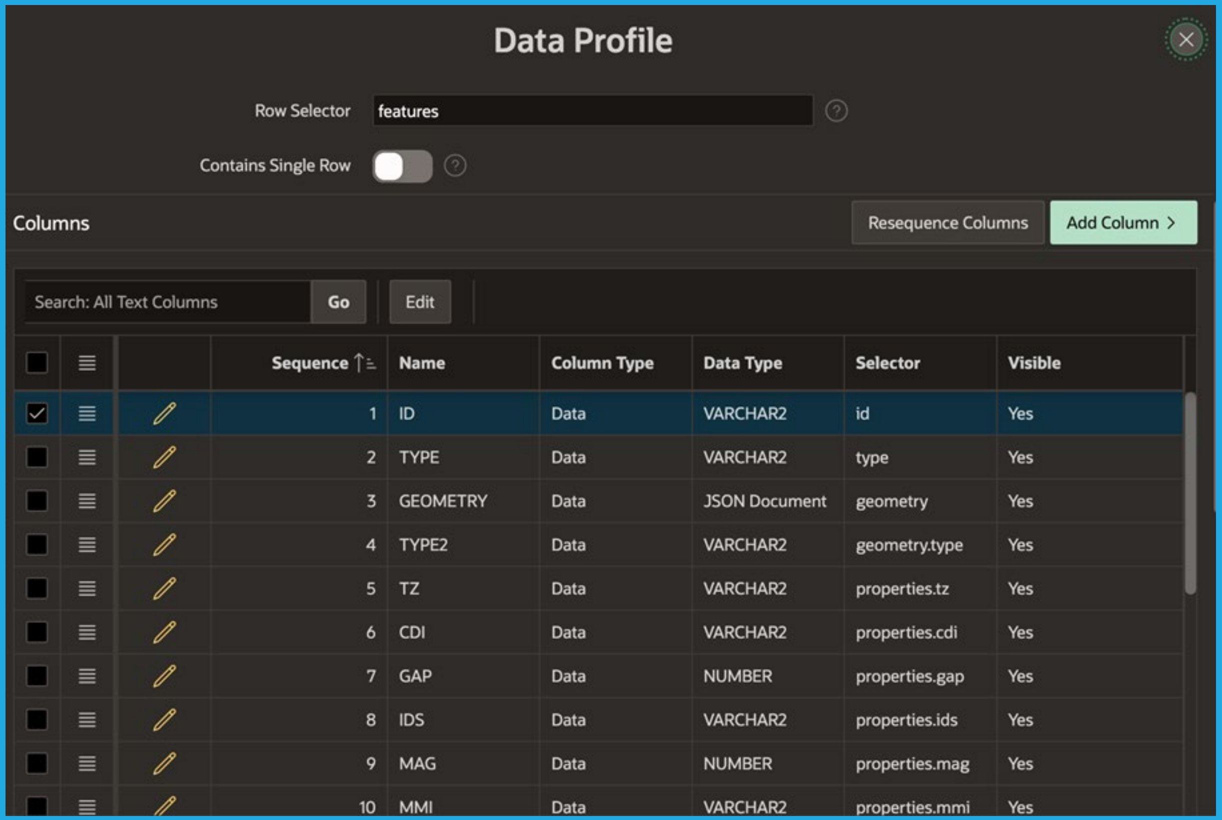
Listing 5: Simple GROUP BY clauses always combine multiple columns.

```
select 'DEPTNO' as facet,     deptno, null as job, count(*)
  from emp
 group by deptno
 union all
select 'JOB'    as facet, null deptno,        job, count(*)
  from emp
 group by job

FACET         DEPTNO JOB            COUNT(*)
---------- -------- ------------ -----------
DEPTNO           30                        6
DEPTNO           10                        3
:
JOB                       SALESMAN         4
JOB                       MANAGER          3
JOB                       PRESIDENT        1
```

Listing 6: Count computation with GROUP BY and UNION ALL

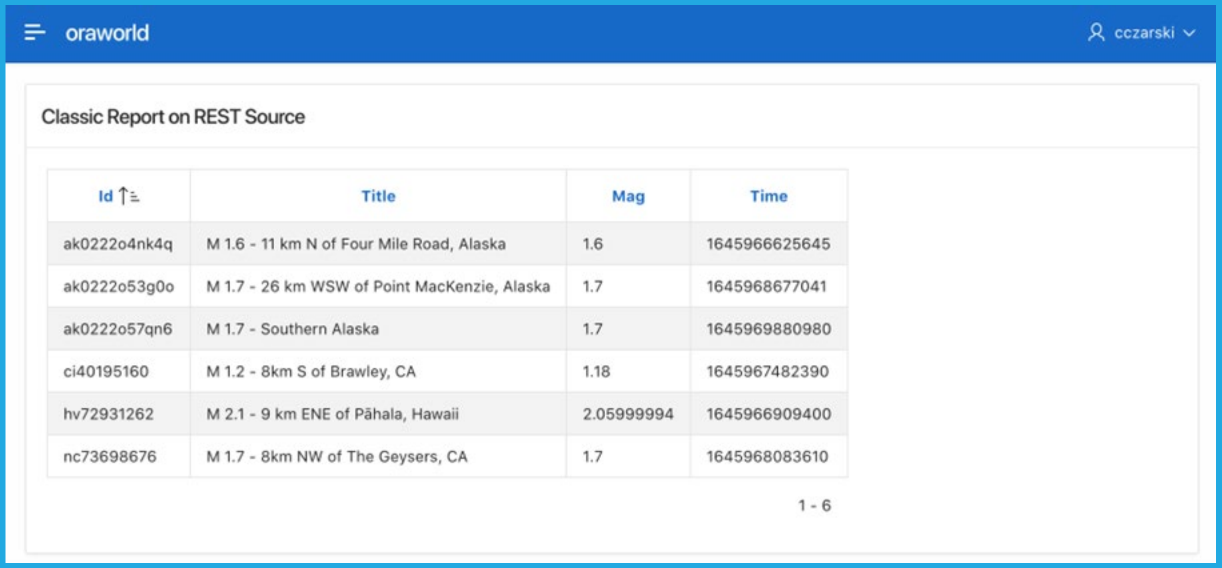
```
select grouping_id(job, deptno), deptno, job, count(*)
   from emp
 group by grouping sets (deptno, job,())

GROUPING_ID     DEPTNO JOB        COUNT(*)
------------ --------- ---------- --------
           3                            14
           2        10                   3
           2        20                   5
           :
           1               ANALYST       2
           1               SALESMAN      4
           1               CLERK         4
```

Listing 7: Count computation using the GROUPING SETS clause



☐			Sequence ↑	Name	Column Type	Data Type	Selector	Visible
☑	≣	✎	1	ID	Data	VARCHAR2	id	Yes
☐	≣	✎	2	TYPE	Data	VARCHAR2	type	Yes
☐	≣	✎	3	GEOMETRY	Data	JSON Document	geometry	Yes
☐	≣	✎	4	TYPE2	Data	VARCHAR2	geometry.type	Yes
☐	≣	✎	5	TZ	Data	VARCHAR2	properties.tz	Yes
☐	≣	✎	6	CDI	Data	VARCHAR2	properties.cdi	Yes
☐	≣	✎	7	GAP	Data	NUMBER	properties.gap	Yes
☐	≣	✎	8	IDS	Data	VARCHAR2	properties.ids	Yes
☐	≣	✎	9	MAG	Data	NUMBER	properties.mag	Yes
☐	≣	✎	10	MMI	Data	VARCHAR2	properties.mmi	Yes

Fig. 7: Data Profile for a REST Data Source



Classic Report on REST Source

Id ↑	Title	Mag	Time
ak0222o4nk4q	M 1.6 - 11 km N of Four Mile Road, Alaska	1.6	1645966625645
ak0222o53g0o	M 1.7 - 26 km WSW of Point MacKenzie, Alaska	1.7	1645968677041
ak0222o57qn6	M 1.7 - Southern Alaska	1.7	1645969880980
ci40195160	M 1.2 - 8km S of Brawley, CA	1.18	1645967482390
hv72931262	M 2.1 - 9 km ENE of Pāhala, Hawaii	2.05999994	1645966909400
nc73698676	M 1.7 - 8km NW of The Geysers, CA	1.7	1645968083610

1 - 6

Fig. 8: Classic Report based on the EMP table

```
select id, mag, title
  from json_table( :json_clob format json,
                   '$.features[*]'
                   columns(
                      id    varchar2(4000) path '$.id',
                      mag   varchar2(4000) path '$.properties.mag',
                      title varchar2(4000) path '$.properties.title'
))

ID              MAG           TITLE
--------------- ------------- -------------------------------------
--
ak0222o58elv    1.6           M 1.6 - 48 km NW of Valdez, Alaska
ak0222o580b9    1.7           M 1.7 - 34 km NW of Ninilchik, Alaska
ak0222o57qn6    1.7           M 1.7 - 44 km WNW of Ninilchik, Alaska
```

Listing 8: Parsing response JSON from a REST Service with JSON_TABLE

The beauty of JSON_TABLE is that it is a *Row Source* for query execution, so it can be treated like a table or view (in ***Listing 8*** we can see that JSON_TABLE is used in the FROM clause). Thus, we can use the JSON_TABLE expression as a subquery, or as an inline view (WITH clause). APEX can apply all other SQL functionality (Aggregates, GROUPING SETS, PIVOT clauses) just on top of the JSON_TABLE results.

Summary

As every APEX developer knows, APEX runs within the Oracle Database. That does not only provide the advantage of "zero-latency" data access, but it also allows APEX to leverage the full range of SQL capabilities: This article highlights the usage of Analytic (Window) functions, the GROUPING SETS and PIVOT clauses and the ability to parse JSON with SQL Queries.

Reviewing APEX Debug output is a great way to get an understanding about how an APEX component (or a Plug-In) works and how it leverages database functionality. Developers who are interested in other APEX components, might simply turn on debug and find out: For instance, the **Tree Region** makes use of the START WITH … CONNECT BY SQL clauses. And there are more APEX components to explore …

More Information

- Information and environment to test-drive APEX: **http://apex.oracle.com/en**
- Oracle Application Express Blog: **http://blogs.oracle.com/apex**
- Oracle Documentation: SQL Language Reference **https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/index.html**
- Oracle Documentation: JSON Developers Guide **https://docs.oracle.com/en/database/oracle/oracle-database/19/adjsn/index.html**

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 new support for REST services in APEX – beyond that Carsten looks after the Calendar component and the Data Loading facility. He is a frequent speaker at international user group conferences.

Lino Schildenfeld

Oracle User Groups around the globe:
Australian Oracle User Group (AUSOUG)

Who We Are

The Australian Oracle User Group (AUSOUG) is an independent not-for-profit professional body representing over 2,500 IT professionals that work with Oracle products and technologies across Australia, Asia and Pacific.

In an industry where the Oracle practitioner is consistently inundated with information, products and services, AUSOUG is widely recognised as a key objective provider of information to its members. In addition to face-to-face events – where you can network with other IT professionals and share experiences – we also maintain and extensive library of white papers, presentations and webinars that are available to our members.

Each year we run the Connect Conference Series that brings together domestic and international speakers that are amongst the best in their field. We also offer hands-on master classes at discounted rates to members where you can learn from experts and get up to speed quickly.

Our members enjoy a comprehensive range of exclusive information, services and networking opportunities within an independent professional body which is part of a worldwide network.

We are a group of dedicated and passionate volunteers that each bring a wealth of skills, experience and knowledge. They are responsible for both setting the long-term strategies of the AUSOUG and managing day-to-day operations.

Clay Barker National President
Mario Misso National Vice-President
Tony Scholefield National Secretary & Branch President, SA
Joseph Kattar National Treasurer
Scott Hollows Branch President, WA
Barbara Rabinwicz Branch Vice-President, VIC
Lino Schildenfeld National Committee member (Oracle APEX)
Simbarashe Ngwenya National Committee member (Big Data/ Analytics/Security/EPM & BI)

Our mission is:
“**To be the leading independent forum for Oracle professionals, partners and users in Australia**.”

History

Founded in 1980ties in New South Wales (NSW). Ever since 2001 we traditionally hosted an annual conference event called AUSOUG CONNECT as our flagship event.

Like any other user group, we went through our ups and downs, changed leaderships and affiliations to get to where we are today - healthy and well positioned OUG leader in APAC region.

Challenges of today and how is the Pandemic affecting us?

Like many other organisations AUSOUG also faces many challenges. From one side influenced by very fluctuating IT market in Australia driven by changes with introduction of Cloud technology, on the other side COVID has played its role too.

Not being able to hold these conferences in person, we like many others have been forced to look into alternatives like hybrid conference model or completely online versions.

As part of the community and knowledge sharing ethos, AUSOUG hosts a number of events with seasoned presenters from around the globe. Connect2021 was one such event that ran over four days covering a wide variety of topics from Oracle EBS, Cloud Applications, APEX, Analytics, EPM and Databases.

If it is to be judged by our partners and feedback from our speakers and attendees both virtual conferences were a big success.

This is hugely to do with having great partners supporting us even more now than ever before during these challenging times. Together with them and the speakers support from Oracle community we have been able to continuously provide top-notch events and content for our members and beyond.

From this perspective we have been very fortunate and look forward into what the future brings.

If there was one thing, we Aussies have learned, we concentrated this time on being actively present on a social media like Twitter and LinkedIn.

Things we do?

Technology workshops

With great support from Oracle APEX team, we organized multiple events like Oracle APEX APAC Tour having Shakeeb Rahman, Christina Cho, Connor McDonald and Francisco Munoz Alvarez presenting in Melbourne, Sydney and Brisbane or APEX Workshop in Melbourne presented by David Peake.

Additionally, we hosted numerous other workshops for our members by other superstars from Oracle ecosystem.

AUSOUG Masters

Each year, AUSOUG recognises the expertise, effort and support of key individuals through the AUSOUG Masters Program. These individuals are selected based on their contribution to AUSOUG, the wider IT industry and, in particular, the Oracle technology sector. Many of these individuals freely share their expertise and experience through social media, by speaking at conferences or by delivering presentations at one of our many events – and so benefitting our members and the group as a whole. While AUSOUG Masters are chosen exclusively by the National Committee, we do welcome suggestions from our members. More info www.ausoug.org.au/ausoug-masters/

Annual webinar series

In addition to our traditional annual conference, for the past 5 years we have also tried to bring community together by organizing a series of free webinars throughout most of the calendar year. This allowed us to provide our members with the latest information brought and presented by many renown international speakers. Examples would be Oracle APEX webinar series 2017 to 2021 or National webinar series.

Watch the space for 2022 – on www.ausoug.org.au

We look forward to hosting more events this year so please subscribe and become a member at **https://www.ausoug.org.au**

As the largest Oracle User Group in the APAC region, we invite you to partner with us.

G'day, mate from Down Under!

About Lino Schildenfeld

Lino is an Oracle APEX consultant with 15+ years of experience who has worked with clients in Europe, USA, Asia, Australia and New Zealand across all sectors.
He contributes to local OUG as AUSOUG committee board member, Connect conference content chair and webinar organizer.
Frequent speaker at KSCOPE, APEX Connect, APEX World, APEX ALPE Adria, UKOUG, EMEA & APAC Groundbreakers events.
APEX blogger, ORCLAPEX-NZ meetup organizer and Oracle ACE Pro.
Lino is a passionate traveller and enjoys playing sports.

Patricia Azzaro

News from Oracle French User Group: ERP On Premise Days

The return to real life and reassurance about the life of On Premise ERP

On the one hand, after a long period of webinars, users wanted to meet "in real life".

On the other hand, whether they were JD Edwards or E-Business Suite users, the same question was asked: what is the future for our On-Premises solution at Oracle?

That's why the Oracle user clubs organised the "On Premise Days".

In the framework of a close and efficient collaboration with the team of Rémy Jacquinot, Applications License Sales Manager, Oracle France, we have first organised a day dedicated to JD Edwards on November 16, 2021 and then another one dedicated to E-Business Suite on March 22, 2022.

These proposals have been very well received by users and have been met with great success. **JD Edwards Day | November 16, 2021**

The day began with a roundtable discussion during which Yohann Garcia, President of the French JD Edwards Club and Mark Herwerge, Oracle J.D. Edwards Solution Engineer interviewed Arnaud Fournier, Global ERP Service Process Lead, Glory Global Solutions and Stéphane Jourdain, CIO, MBWS about their usage strategy and the value of the solution for their organization.

CLUBS UTILISATEURS
REJOIGNEZ LA COMMUNAUTÉ DES UTILISATEURS ORACLE !
ORACLE
JDE DAY
Oracle Applications Unlimited

Mark then presented JD Edwards Product Update and Roadmap before handing over to Lionel Dubreuil, Director EMEA Oracle Cloud Infrastructure Center Of Excellence, for an overview of Oracle Cloud Infrastructure for JD Edwards.

After many questions and a convivial lunch, the afternoon was dedicated to practical workshops with feedback and sharing of best practices between users:

- Reliability of accounts through certification, Testimony of Kaufman & Broad with their partner IBM IBM Interactive
- Good practices in JD Edwards to limit supplier fraud, b.workshop
- J.D. Edwards & Oracle Field Source Cloud, Glory Global Solutions

2 hands-on sessions on related solutions were also presented by Oracle:

- JD Edwards Extensibility Framework
- JD Edwards & Oracle Supply Planning Cloud

E-Business Day Day | March 22, 2022

The day started with an extract from a cult scene from Matrix and an adapted voice dubbing : *“I'm happy to see the Oracle team, I have a lot of questions to ask them”…*

Users & Groups

Nadia Bendjedou, Vice President - Product Strategy - Oracle E-Business Suite at Oracle Corporation Nadia began the day by presenting the strategy and roadmap of the On-Premise versions

From North Africa, Mohammed Ouattassi, Business Solutions Leader – Oracle EMEA presented how to Improve User Experience through Exploration and Data Visualisation, with ECC and offered Finance & Procurement live demonstrations

Sylvain Bondil, Senior Solution Engineer – Cloud Architect then presented EBS Analytics Cloud Accelerator

The morning ended with a very nice session during which the overview of the different evolution strategies proposed by Nadia was followed by numerous user feedbacks:

- On Premise Upgrade | BANQUE DE France
- Lift & Shift Oracle Cloud Infrastructure (OCI) | ACCOR INVEST, BANDAI NAMCO Europe, SOITEC,
- Transformation to SaaS | FRANCE TELEVISIONS
- Reimplementation and evolution in a coexistence scenario (Oracle EBS + SaaS) in several African countries | GECAMINES who testified from the Republic of Congo

The exchanges between users and the questions to the speakers continued during the lunch served at Spanish time :)
Four other customers shared their experience during the afternoon :

- Implementation of the Purchasing IS | BANQUE DE FRANCE
- Oracle Cloud Supplier Portal: A private and secure space to consult P2P documents | ORANGE
- Redesign of the Oracle E Business Suite Finance, HCM, SCM 12.1.3 | AFRICA Re from Nigeria
- Presentation Modernisation and Harmonisation of IPRES & CSS IS | IPRES from Senegal

Mohammed then took the floor again to detail how to effectively manage production operations while respecting commitments made to customers, with Discrete Manufacturing Command Center

The day ended at 7.30 pm after the presentation of Advanced Customer Services by Jeanne-Aimée Gras, ACS EMEA Service Solution Architecture, ORACLE

An intense and very interesting day prepared and animated with talent by the 2 pilots of the On premise working group: Jérôme Gauvrit, CIO, Bodet and Wanda Marin, Executive director, Prerequis.

It is really necessary to keep in mind three major particularities of this E-Business Day :

- its hybrid format with on-site + remote speakers and participants
- speakers and participants from French-speaking Africa
- the record number of questions asked...and answers given

Stratégies d'évolution | Vue d'ensemble sur les différentes possibilités
Choices for EBS Customers – Any combination of these options
Modernize EBS by Upgrading to 12.2
Modernize EBS by Lift & Shift to OCI
Extend EBS with SaaS Co-existence
Transform by Migrating to SaaS

Conclusion

Clear and reassuring…the conclusion of these days is that On-Premise still has a bright future!

The ERP On Premise Days would not have happened without the hard work and dedication of of the drivers from both days: Yohann Garcia for JDE Day, Wanda Marin & Jérôme Gauvrit co-pilots for EBS Day.

We are now working on organizing PeopleSoft Day in June

Detailed programs and speakers are available on our website:
https://clubutilisateursoracle.org/
Presentations and replays are also available for our members

About Patricia Azzaro

Patricia AZZARO is the General Delegate of the French-speaking Oracle Users' Clubs since 2000.
With her current team composed of Laurine GUILLAUME and Ines POULANGES, under the governance of club boards, she manages the life of the clubs.
That means the administrative, legal and above all network animation : conception and organisation of the activities, relationships between members and with experts in the Oracle ecosystem.
Patricia enjoys diving, hiking in the mountains and has always been involved in environmental issues. She loves to spend time with her family and friends.

Call for Papers

Connect 2022 - AUSOUG

September, 30
https://www.ausoug.org.au/connect-conference-2022-call-for-papers-submission/

AUSOUG Connect 2022 will take place over 4 days from 7th November 2022 to 10th November 2022, virtually and online, with each day dedicated to a stream.

Events

DOAG 2022 Conference + Exhibition

September 20-23, 2022
Nuremberg, Germany
https://anwenderkonferenz.doag.org/en/home/

Oracle Community Tour EMEA - CASOUG Day (Kyrgyzstan)

September 26, 2022
online
www.ogbemea.com

Oracle Community Tour EMEA - CASOUG Day (Uzbekistan)

September 27, 2022
online
www.ogbemea.com

Oracle Community Tour EMEA - CASOUG Day (Tajikistan)

September 29, 2022
online
www.ogbemea.com

Oracle Community Tour EMEA - AzerOUG Day

October 1, 2022
online
www.ogbemea.com

Oracle Community Tour EMEA - DOUG Day 2022

October 3, 2022
Copenhagen, Denmark
https://www.doug.dk/doug-dag-2022/

LuxOUG - EMEA Community Tour 2022

October 5, 2022
Luxembourg
https://www.luxoug.org/luxoug-2022/

ConTech2022 + RoOUG Day, part of EMEA Community Tour 2022

October 6 - 7, 2022
Bucharest, Romania
https://contech2022.rooug.ro/

Events

Oracle Community Tour EMEA - OUGN Day 2022

October 10, 2022
Oslo, Norway and online
www.ogbemea.com

HROUG22- Autumn + HROUG day, part of Oracle Community Tour EMEA

October 11 - 14, 2022
Rovinj , Croatia
https://2022autumn.hroug.hr/eng/

Oracle Community Tour EMEA - LVOUG Day 2022

October 13, 2022
online
www.ogbemea.com

Oracle CloudWorld

October 17-20, 2022
Las Vegas, USA
https://www.oracle.com/cloudworld/

Breakthrough'22 - UKOUG

December 1-2, 2022
Birmingham, United Kingdom
https://ukoug.org/page/Breakthrough22

ORAWORLD is a publication of the EOUC — EMEA ORACLE USERGROUP COMMUNITY

The following user groups belong to EOUC:

Angola Oracle User Group, Oracle User Group Armenia, Austrian Oracle User Group, Azerbaijan Oracle User Group, Bulgarian Association of Software Developer, Bulgarian Oracle User Group, Hrvatska udruga Oracle korisnika, Czech Oracle Applications User Group, Danish Oracle User Group, Egypt Oracle Users Group, Oracle User Group Estonia, Oracle User Group Finland, Club Français des Utilisateurs JD Edwards, Association des Utilisateurs Francophones d'Oracle, Club des Utilisateurs PeopleSoft, Oracle User Group Georgia, Deutsche Oracle Anwendergruppe, PeopleSoft Germany, Hungarian Oracle User Group, Israel Oracle User Group, Taranta Valley Oracle User Group, Italian Oracle User Group, Jordan Amman Oracle User Group, Latvian Oracle Users Group, Lithuanian Oracle Users Group, Mauritius Oracle User Group, Oracle Gebruikersclub Holland, Oracle Benelux User Group, Oracle User Group Norway, Polish Oracle Users Group, Oracle Users Group Portugal, Romanian Oracle User Group, Russian Oracle User Group, EBS Finance Special Interest Group Russia, Arab Oracle User Group, Serbia and Montenegro, Slovenian Oracle User Group, South African Oracle User Group, Spanish Oracle User Group, Swedish Oracle User Group, Swiss Oracle User Group, Tajikistan Oracle User Group, Turkey Oracle Users Group, Ukraine Oracle User Group, Middle East Oracle User Group, United Kingdom Oracle User Group, Zimbabwe Oracle User Group.

Editorial board:

Registered office: DOAG Dienstleistungen GmbH
Tempelhofer Weg 64, 12347 Berlin, Germany
www.doag.org,
Director Fried Saacke,
AG Berlin Charlottenburg HRB 95694B,
VAT ID DE240700058
Contact: redaktion@doag.org
Editor-in-chief (ViSdP): Mirela Ardelean

Editorial team:

Mirela Ardelean, Jean-Jacques Camps, Heli Helskyaho,
Ann-Sofie Vikström Often, Andrejs Vorobjovs

Other editors: Marcos López, Moritz Rißmann

Other authors are credited by name with their article. They are subject to the terms and conditions for authors: www.oraworld.org/terms-and-conditions-for-authors

Graphic design:

Alexander Kermas, DOAG Dienstleistungen GmbH,
Tempelhofer Weg 64, 12347 Berlin, Germany

Article submission:

If you are interested in submitting an article, please e-mail us your article via the online form at **www.oraworld.org**. Deadline for the next edition: **November 15, 2022**.

Photo credits:

Title: © Designed by BiZkettE1 | https://freepik.com
P. 4: © floralset | https://123rf.com
P. 5: © Comic | www.commitstrip.com
P. 6 © Designed by slidesgo | https://freepik.com
P. 7: © Designed by pch.vector | https://freepik.com
P. 13: © Designed by pikisuperstar | https://freepik.com
P. 18: © Login | https://stock.adobe.com
P. 25: © Designed by freepik | https://freepik.com
P. 29: © Designed by pikisuperstar | https://freepik.com
P. 33: © Designed by stories | https://freepik.com
P. 34+35: © DOAG | www.doag.org/en/home

Disclaimer:

The information in this publication has been duly researched and is correct to the best of our knowledge and belief. The use of this information is at your own risk. No liability for the accuracy of the information is accepted and, in particular, for its practical application in individual cases. Opinions represent the views of the individual author and do not necessarily represent the view of the publisher.

The ORAWORLD e-magazine reports on events in the Oracle and IT world. The publication covers current topics in the international user group network, as well as news items on products and technologies and their use. The purpose of the magazine is to foster the sharing of knowledge and experience among readers. ORAWORLD is independent of Oracle and does not represent its commercial interests either directly or indirectly.

ORAWORLD is published by DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Germany, legally represented by director Fried Saacke, the nature and purpose of whose business is management of the group, organising events and publishing.

DOAG Deutsche Oracle Anwendergruppe e.V. holds 100 percent of the capital invested in DOAG Dienstleistungen GmbH. DOAG Deutsche Oracle Anwendergruppe e.V. is legally represented by the managing committee; Chair: Björn Bröhl.