ORACLE
AUTONOMOUS DATABASE **LEARNING LOUNGE**

Developer's nirvana with Autonomous Database: JSON-Relational Duality in Oracle Database 23ai

Autonomous Database Learning Lounge

Hosted by Marcos Arancibia
Autonomous Database Product Management

Agenda

Julian Dontcheff

Topics

- **Overview** of the **JSON-Relational** capability and features
- **JSON-Relational Duality View** restrictions and limitations
- New **Flex** columns
- **Outliers and minFrequency** special calculations
- **JSON Collections Views** and the **MongoDB API**
- **LIVE Demo**

Q&A

- **Product Managers will answer any questions**

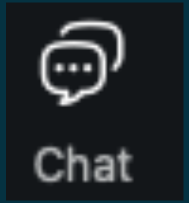
Before we begin…

This session is for you !!!

Ask your questions using **Q&A**

Product Managers are monitoring your questions

We will share links in **Chat**



The recording will be made available in a few days at oracle.com/goto/adb-learning-lounge

Important links to bookmark

Links to get you started and to keep up to date with Autonomous Database

1 New Get Started page:
oracle.com/autonomous-database/get-started/

2 Join us:

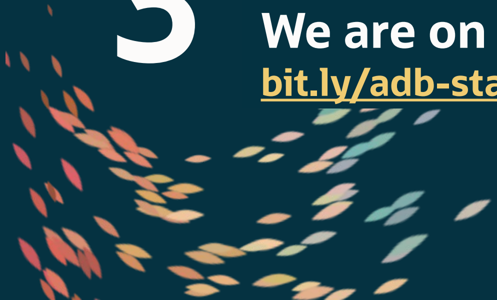
LinkedIn
bit.ly/adb-linkedin-grp

X
@AutonomousDW

Bluesky
autonomousdb.bsky.social

3 Got a question?
We are on stackoverflow
bit.ly/adb-stackoverflow

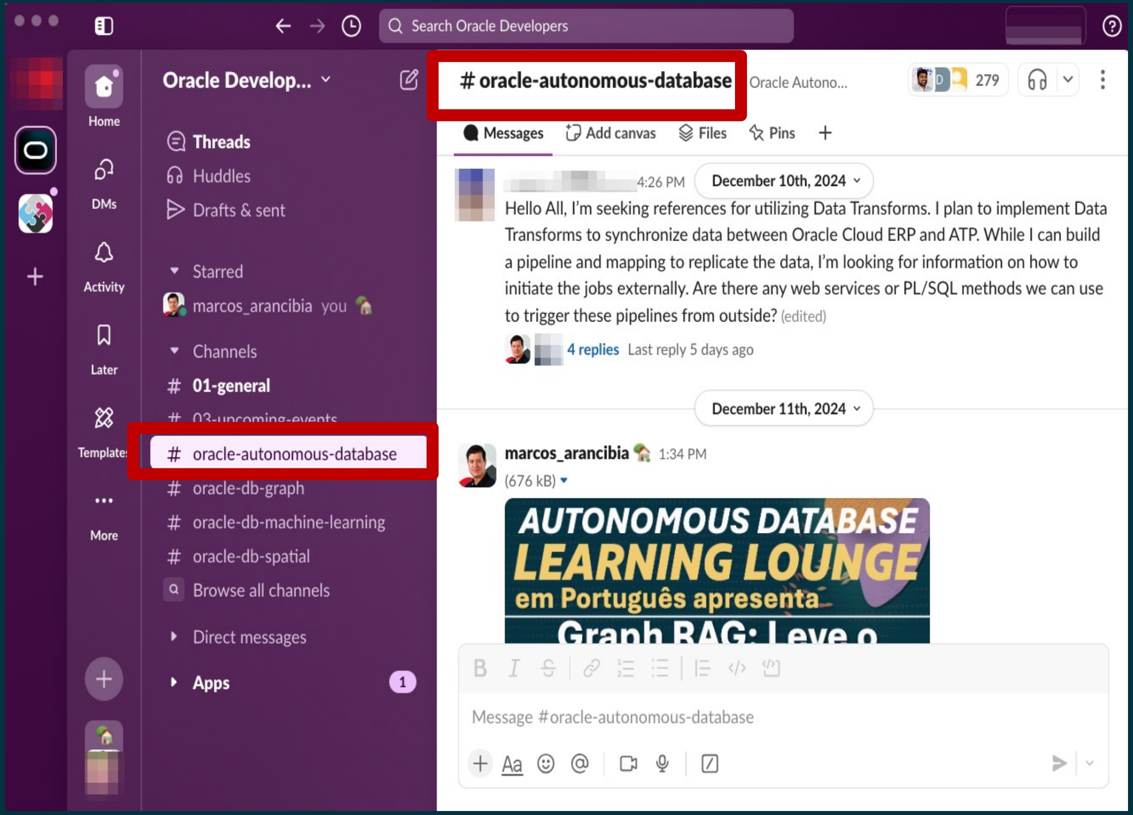
Join us on Developers Slack
(search #oracle-autonomous-database)
(odevrel_slack)



Join our External Slack

STEP 1: Join our Slack workspace at:
https://join.slack.com/t/oracledevs/shared_invite/zt-327lxqzeo-7cfyrWzWAY7curl7MCVF1w

STEP 2: search for #oracle-autonomous-database at the top and click on the Channel



[redacted] 4:26 PM

Hello All, I'm seeking references for utilizing Data Transforms. I plan to implement Data Transforms to synchronize data between Oracle Cloud ERP and ATP. While I can build a pipeline and mapping to replicate the data, I'm looking for information on how to initiate the jobs externally. Are there any web services or PL/SQL methods we can use to trigger these pipelines from outside? (edited)

4 replies Last reply 5 days ago

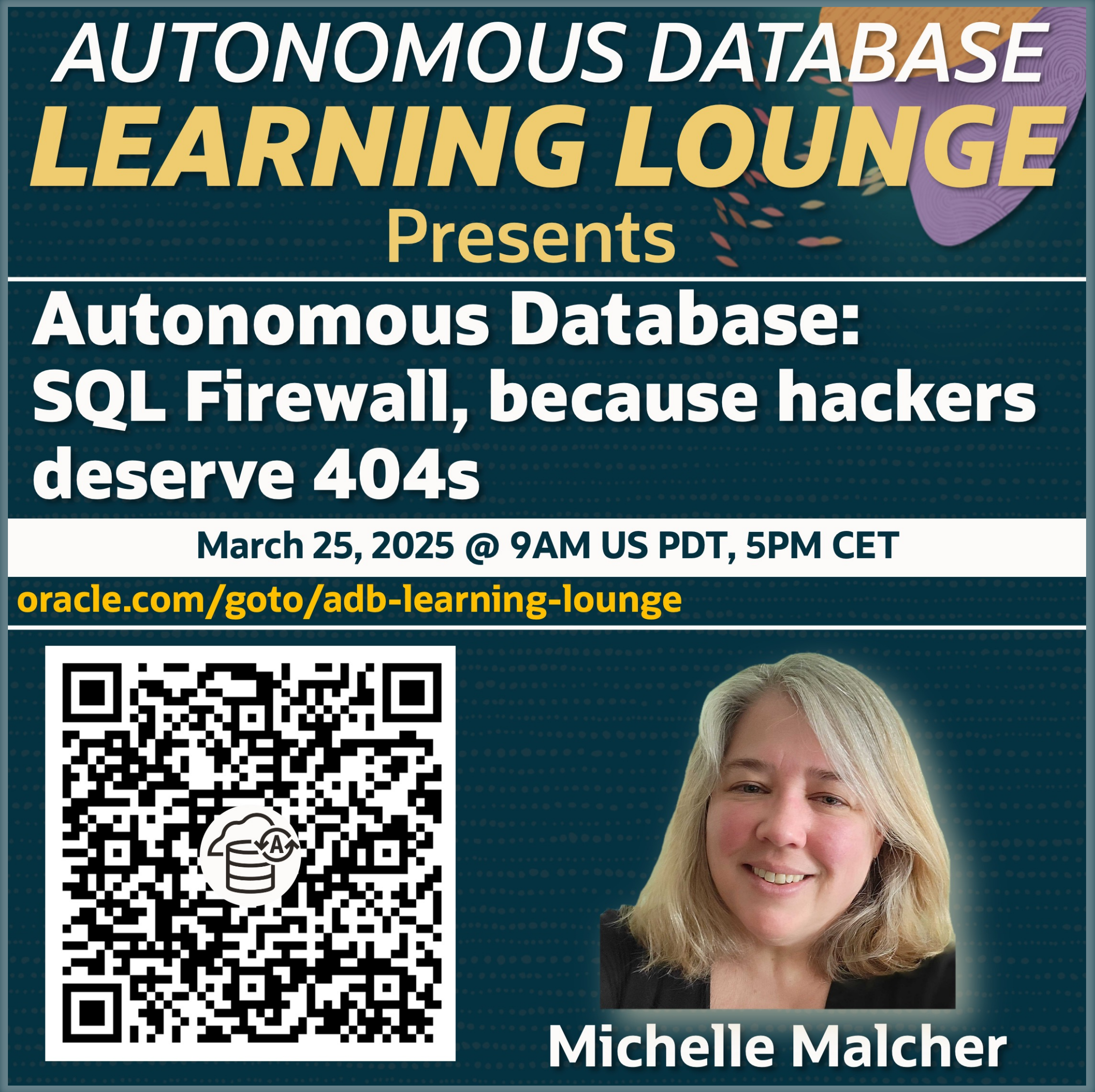
marcos_arancibia 1:34 PM

(676 kB)

AUTONOMOUS DATABASE LEARNING LOUNGE em Português apresenta Graph RAG: Leve o

Message #oracle-autonomous-database

Upcoming Sessions



AUTONOMOUS DATABASE

LEARNING LOUNGE

Presents

**Autonomous Database:
SQL Firewall, because hackers deserve 404s**

March 25, 2025 @ 9AM US PDT, 5PM CET

oracle.com/goto/adb-learning-lounge

Michelle Malcher

Speaker

Julian Dontcheff

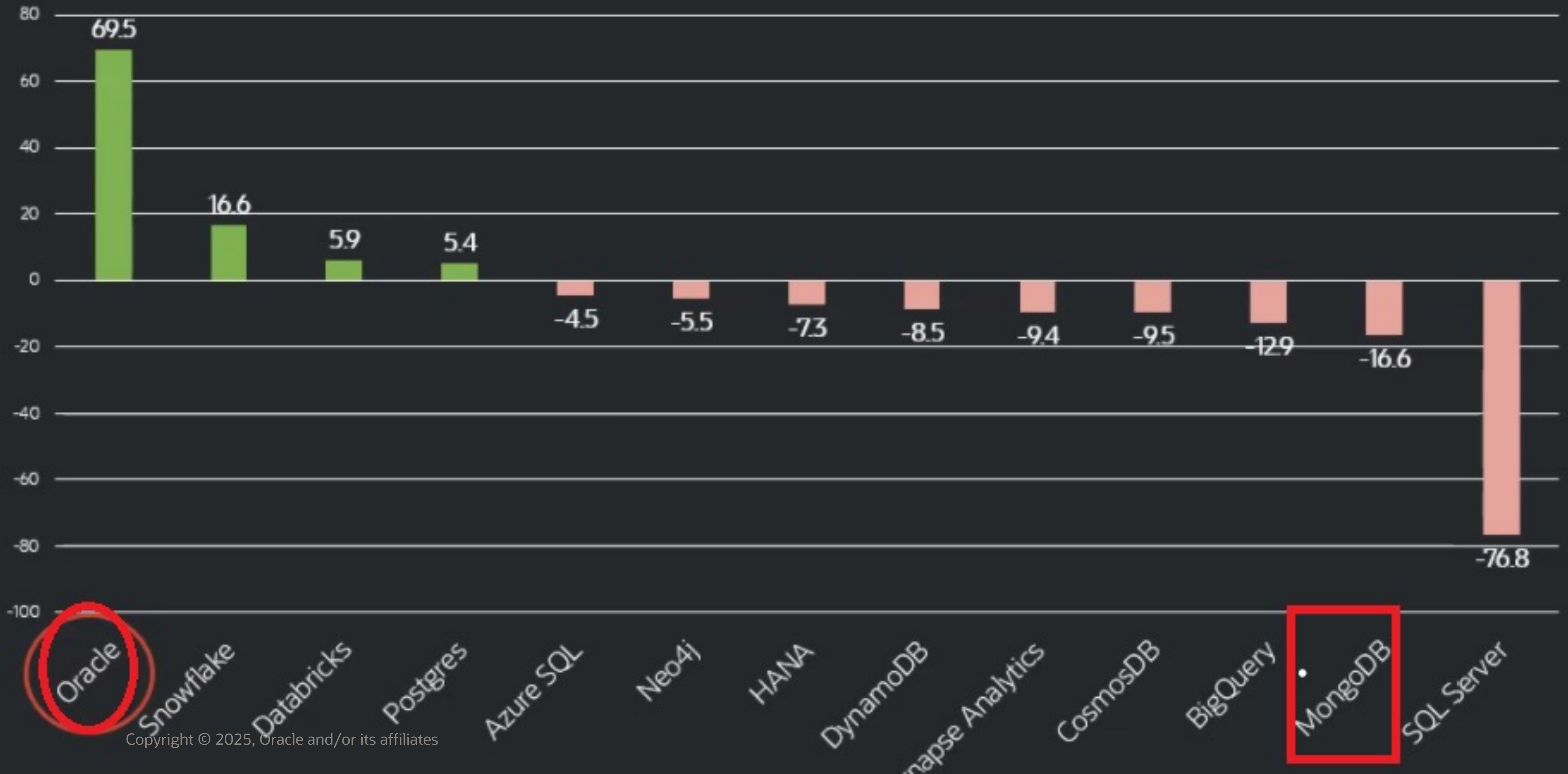
ORACLE

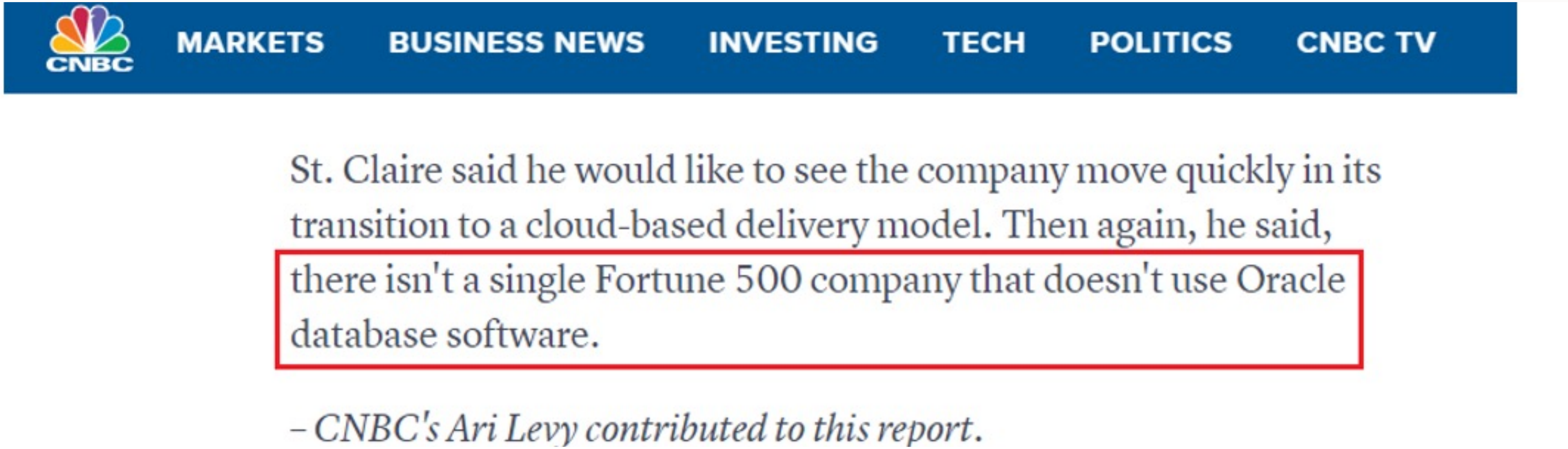
New features of JSON (Duality) for DBAs and Developers

Julian Dontcheff and Hermann Bär

March 18th, 2025

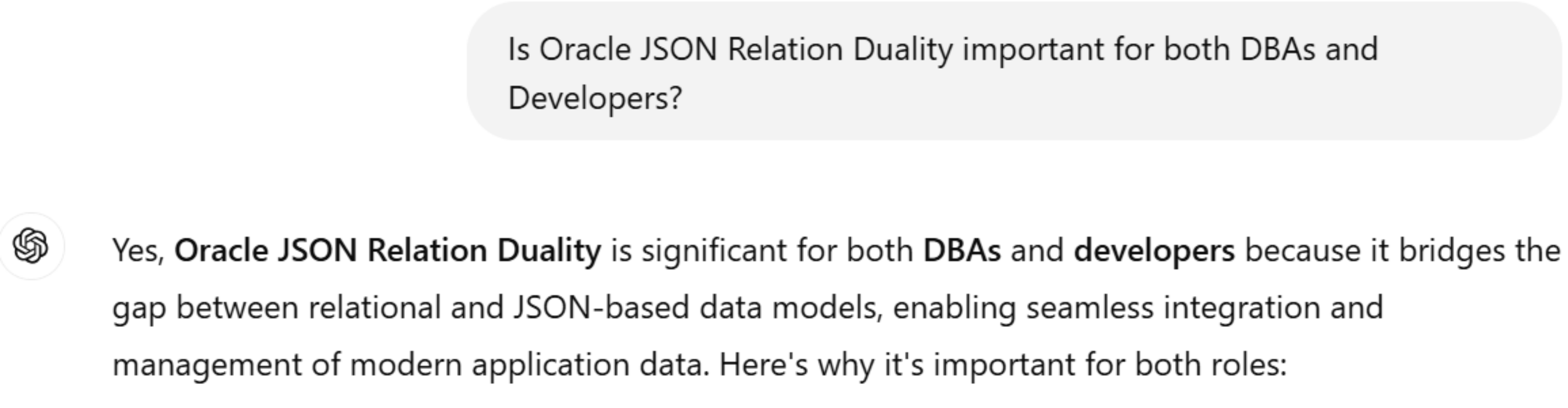
DB-Engines Points Change from Jan 2024 to Nov 2024





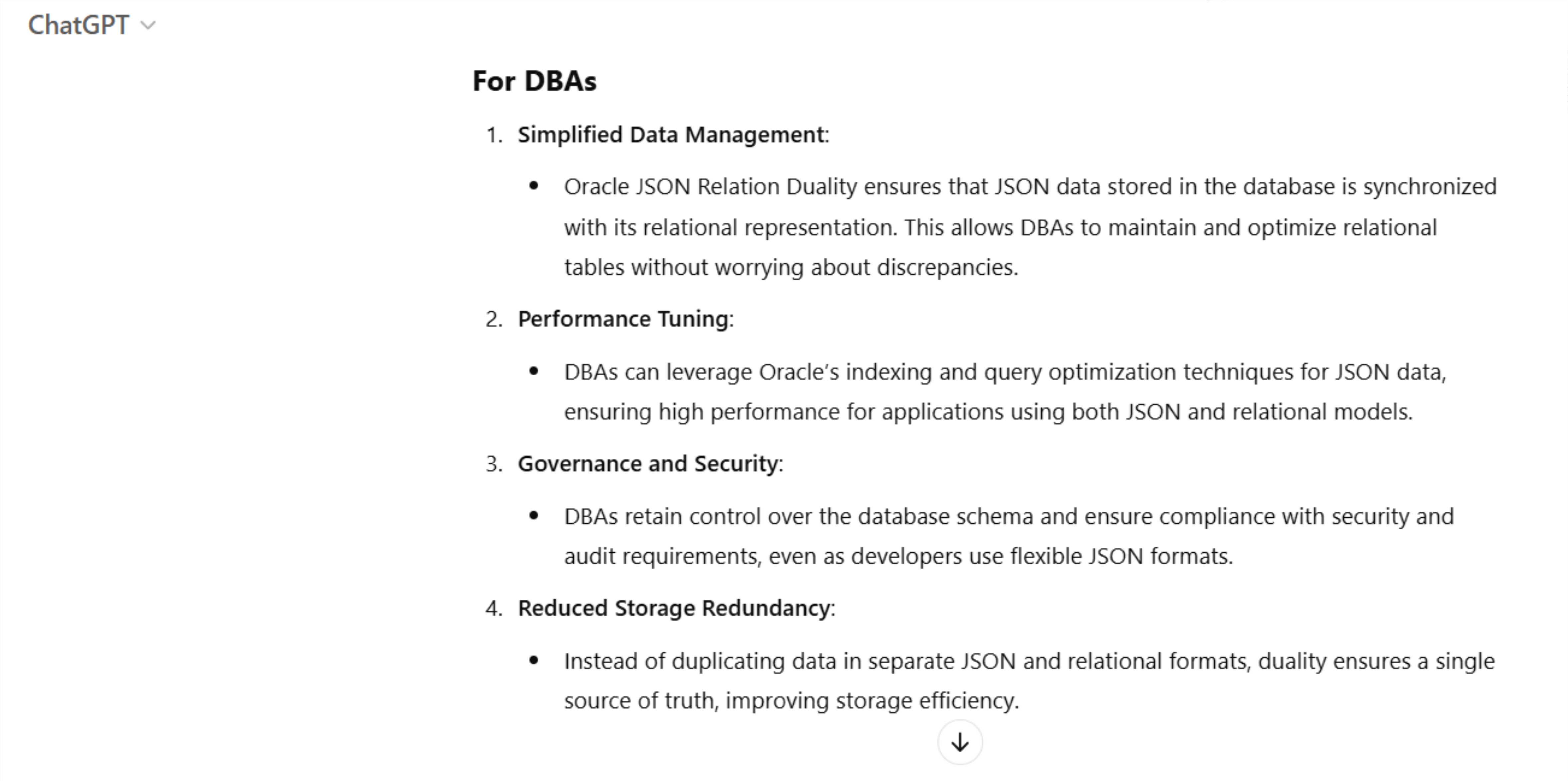
St. Claire said he would like to see the company move quickly in its transition to a cloud-based delivery model. Then again, he said, there isn't a single Fortune 500 company that doesn't use Oracle database software.

– CNBC's Ari Levy contributed to this report.



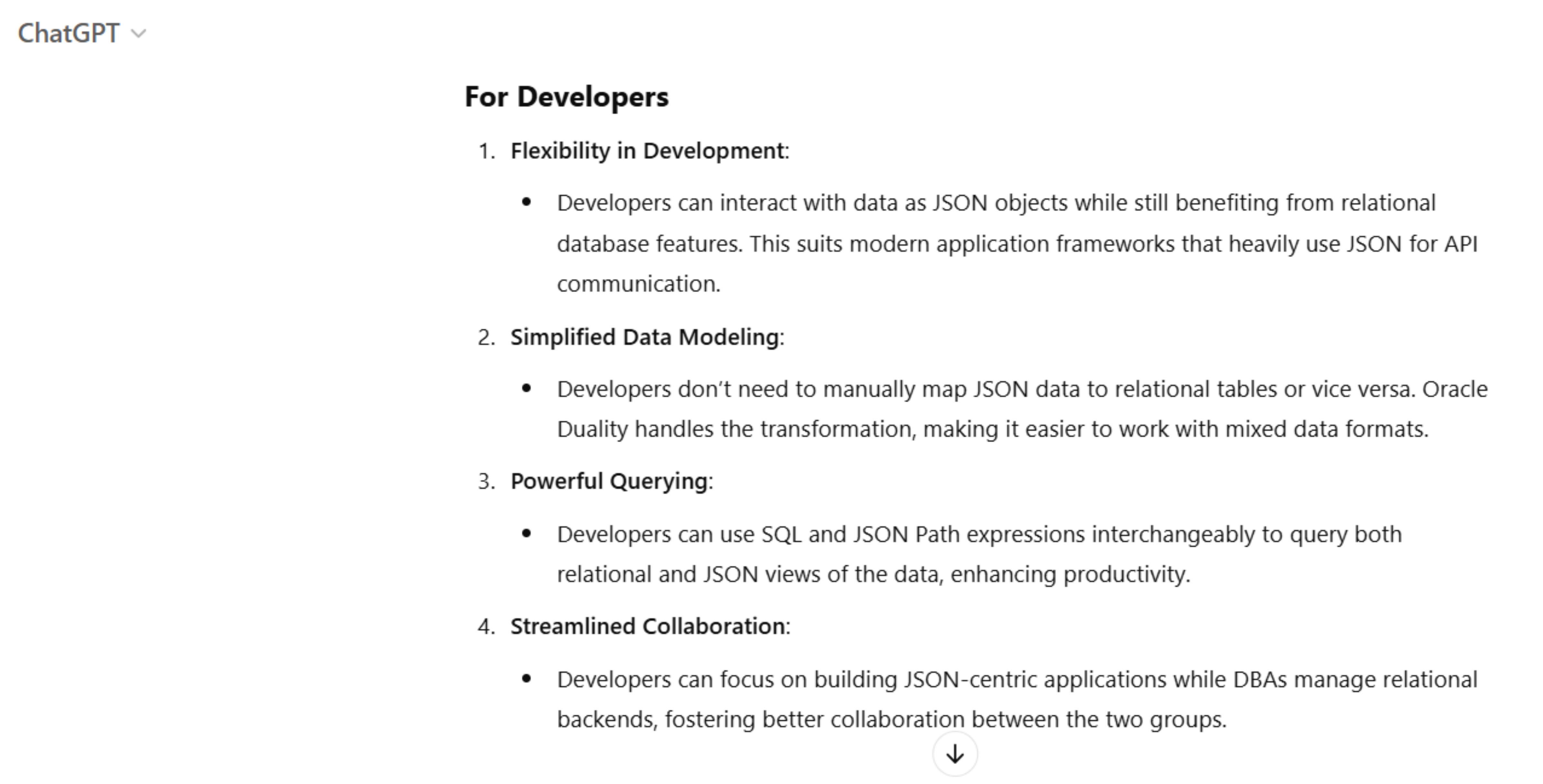
Is Oracle JSON Relation Duality important for both DBAs and Developers?

Yes, **Oracle JSON Relation Duality** is significant for both **DBAs** and **developers** because it bridges the gap between relational and JSON-based data models, enabling seamless integration and management of modern application data. Here's why it's important for both roles:



For DBAs

1. **Simplified Data Management**:
   - Oracle JSON Relation Duality ensures that JSON data stored in the database is synchronized with its relational representation. This allows DBAs to maintain and optimize relational tables without worrying about discrepancies.
2. **Performance Tuning**:
   - DBAs can leverage Oracle's indexing and query optimization techniques for JSON data, ensuring high performance for applications using both JSON and relational models.
3. **Governance and Security**:
   - DBAs retain control over the database schema and ensure compliance with security and audit requirements, even as developers use flexible JSON formats.
4. **Reduced Storage Redundancy**:
   - Instead of duplicating data in separate JSON and relational formats, duality ensures a single source of truth, improving storage efficiency.



For Developers

1. **Flexibility in Development**:
   - Developers can interact with data as JSON objects while still benefiting from relational database features. This suits modern application frameworks that heavily use JSON for API communication.
2. **Simplified Data Modeling**:
   - Developers don’t need to manually map JSON data to relational tables or vice versa. Oracle Duality handles the transformation, making it easier to work with mixed data formats.
3. **Powerful Querying**:
   - Developers can use SQL and JSON Path expressions interchangeably to query both relational and JSON views of the data, enhancing productivity.
4. **Streamlined Collaboration**:
   - Developers can focus on building JSON-centric applications while DBAs manage relational backends, fostering better collaboration between the two groups.

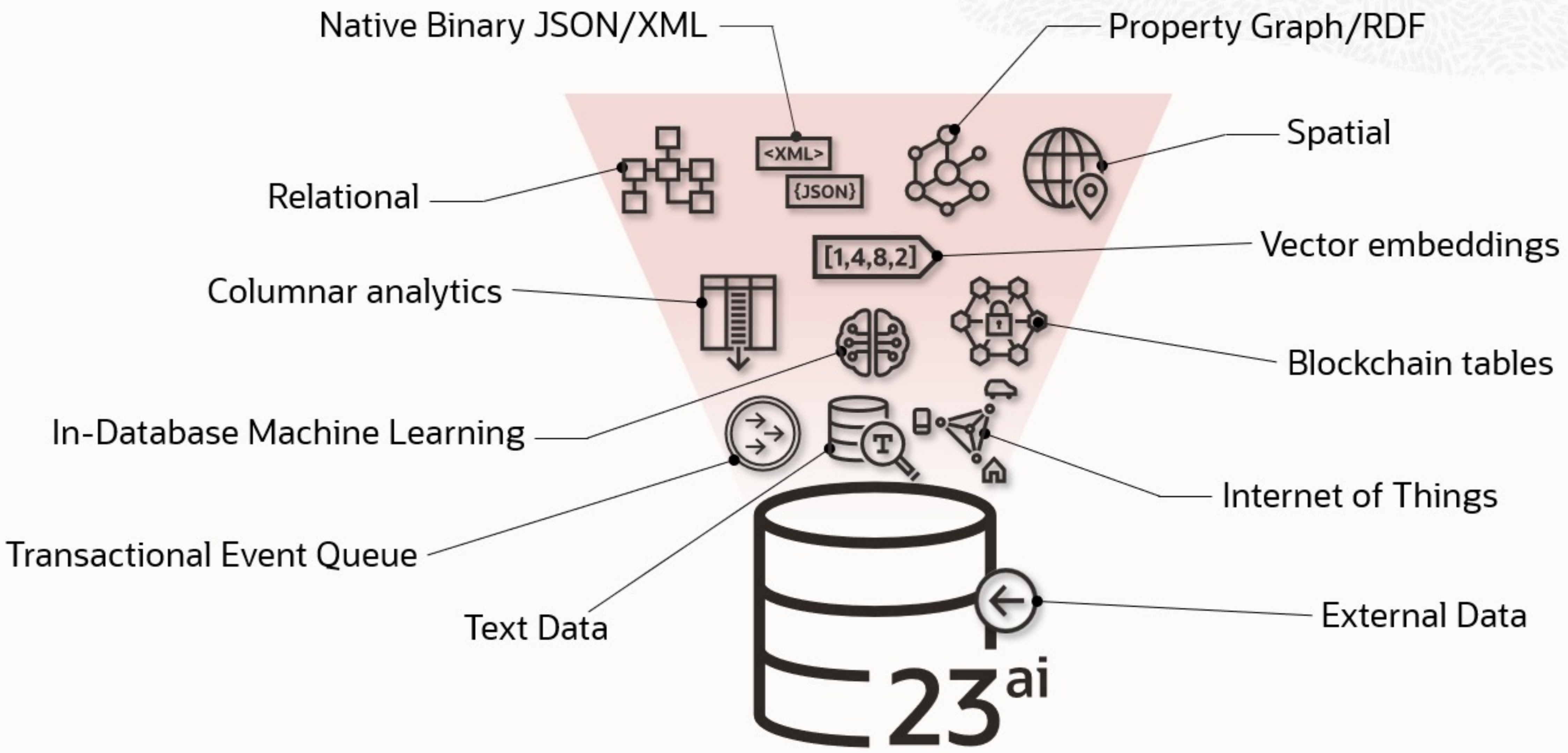
Agenda

- What’s new and enhanced in Oracle Database 23.6 and 23.7 for JSON?
- Mongo-to-Duality Migrator: Flex columns, Outliers and minFrequency
- LIVE demo
- JSON Collections and the MongoDB API
- LIVE demo
- Wrap up

What's new and enhanced in Oracle Database 23.6 and 23.7 for JSON?

23.6 and 23.7 new features for JSON

Oracle Converged Database



JSON in the Oracle Database before 2025

Quick historical update

- 12cR1 (2014)
  - JSON-text storage and query processing (BLOB, CLOB, varchar2), SODA APIs
- 12cR2 (2017)
  - Dataguide, Columnar processing for JSON, Search indexes, JSON generation from relational data
- 18c (2018)
  - GeoJSON, SQL improvements, On statement Materialized Views, SODA for PL/SQL
- 19c (2019)
  - Binary JSON storage (OSON) and MongoDB API added for ADB (BLOB columns), Partial updates
- 21c (2020)
  - Native JSON datatype and collections backed by OSON (all database types), Multi-value indexes
- 23c (2023)
  - Duality Views, JSON Schema, SQL improvements, JS stored procedures, Graph on JSON, perf.
- 23ai (2025)
  - JSON-to-Duality Migrator, JSON collection tables, Multi-value JSON indexes, External JSON Tables, Aggregation Pipelines

New 23.6 and 23.7 features

JSON

- JSON Collection Views
- JSON Replication
- The Limit Clause for JSON
- JSON Search Index Path Subsetting
- Replication Support for JSON Collection Tables
- GoldenGate Replication of JSON-Relational Duality Views

New 23.6 and 23.7 features

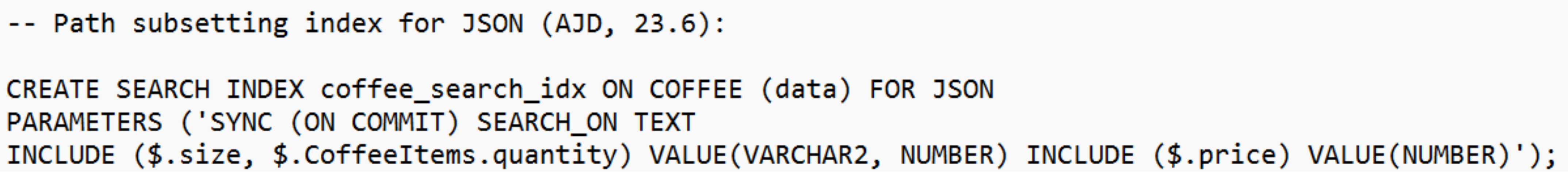
JSON Duality Views

- In JSON Relational Duality View DDL users can do the following:
  - Use field name "_id" in subobjects, to identify the column that selects the row for updates, (even if it has no UK or PK constraint)
  - Use an identity column as an identifying column
- The additional DDL possibilities allow for more use cases supported by JSON Relational Duality views

New 23.6 and 23.7 features

JSON Search Index Path Subsetting

- When creating a JSON search index you can specify the fields to include or exclude from indexing
- Path subsetting can reduce the size of a search index and improve its performance - LIVE Demo:



```
-- Path subsetting index for JSON (AJD, 23.6):

CREATE SEARCH INDEX coffee_search_idx ON COFFEE (data) FOR JSON
PARAMETERS ('SYNC (ON COMMIT) SEARCH_ON TEXT
INCLUDE ($.size, $.CoffeeItems.quantity) VALUE(VARCHAR2, NUMBER) INCLUDE ($.price) VALUE(NUMBER)');
```

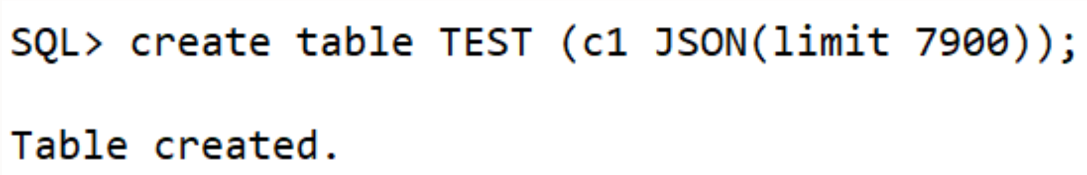
- Hybrid Vector Indexes can now be built on JSON columns
- Hybrid Vector Indexes allow users to easily index and query their documents using a combination of full-text search and semantic vector search to achieve higher-quality search results

New 23.6 and 23.7 features

JSON Data Type and JSON Replication

JSON Type Modifier Enhancements: Limit Clause

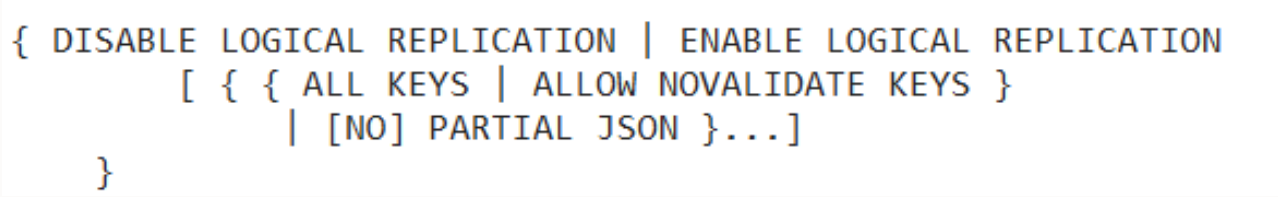
- The limit clause for JSON type allows to restrict the size of the JSON data that can be inserted into the JSON type column (similar idea to that of specifying varchar2 bytes storage size)
- If it is known that the JSON data does not exceed a certain amount then the execution can be optimized to be more efficient, Oracle 23ai will improve performance without code change as for small JSON (encoded as OSON < 8000 bytes), those will be inlined vs (4000 bytes in 19c)



```
SQL> create table TEST (c1 JSON(limit 7900));

Table created.
```

- The logical_replication_clause of the CREATE/ALTER TABLE statement is extended to allow disabling and enabling of partial JSON updating under supplemental logging
- Partial JSON updating makes replication more efficient because less data needs to be replicated or modified

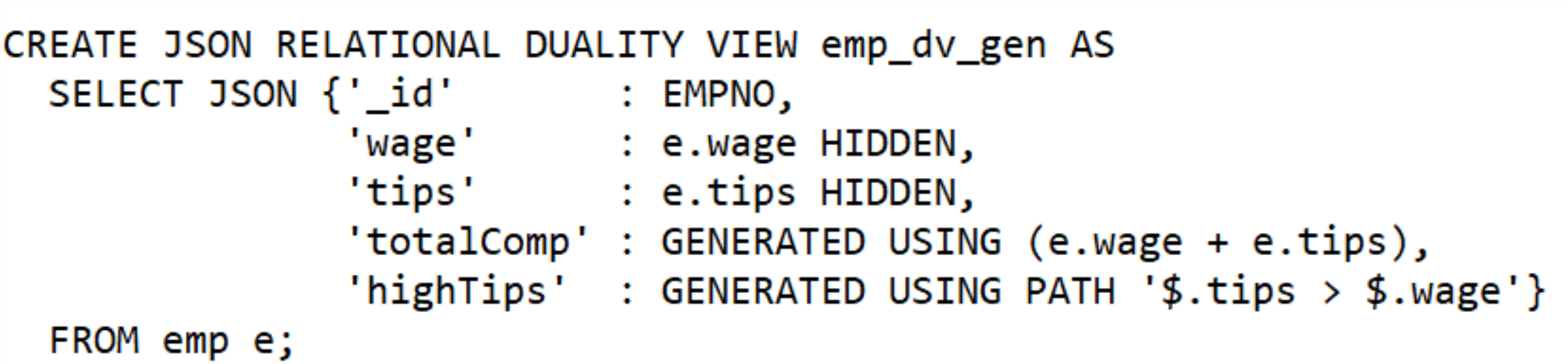


```
{ DISABLE LOGICAL REPLICATION | ENABLE LOGICAL REPLICATION
        [ { { ALL KEYS | ALLOW NOVALIDATE KEYS }
             | [NO] PARTIAL JSON }...]
     }
```

New 23.6 and 23.7 features

Hidden and Generated Fields in JSON-Relational Duality Views

- A column in a table underlying a JSON-relational duality view can be mapped to a hidden field; that is, a field that's not present in the documents supported by the view
- A generated field in a JSON document supported by a JSON-relational duality view can use the value of hidden fields
- A JSON document supported by a duality view can be simpler if it need not have a field for every underlying column, in particular for columns needed only for calculating



```
CREATE JSON RELATIONAL DUALITY VIEW emp_dv_gen AS
  SELECT JSON {'_id'         : EMPNO,
               'wage'        : e.wage HIDDEN,
               'tips'        : e.tips HIDDEN,
               'totalComp'   : GENERATED USING (e.wage + e.tips),
               'highTips'    : GENERATED USING PATH '$.tips > $.wage'}
  FROM emp e;
```

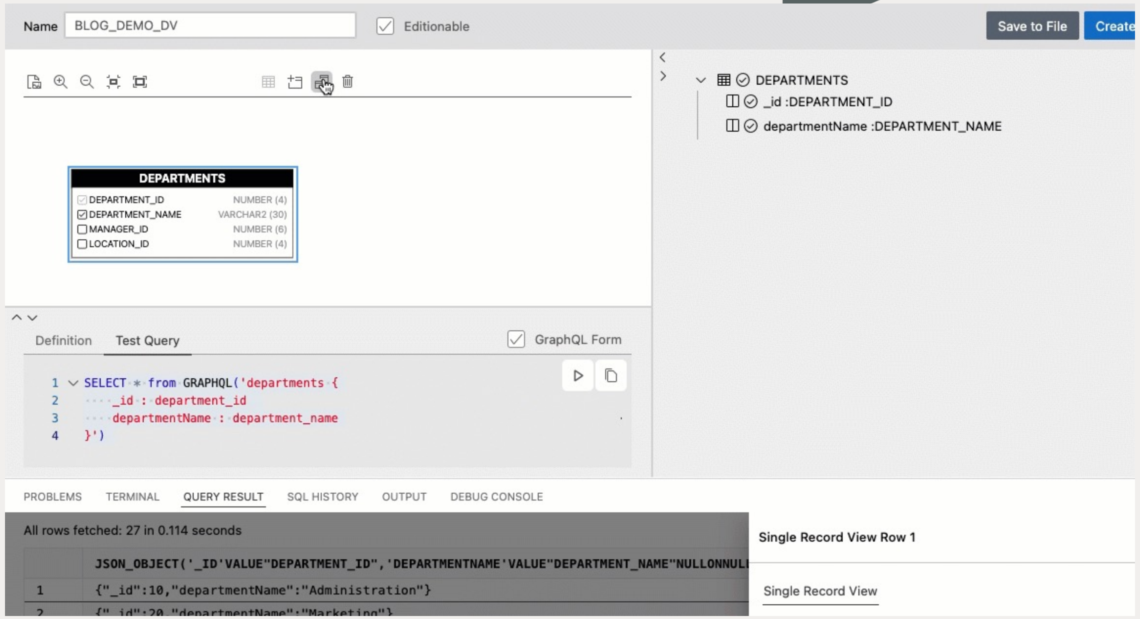
New 23.6 and 23.7 features

GoldenGate Replication of JSON-Relational Duality Views

- JSON Collection Tables can be enabled for logical replication using GoldenGate
- Replication is supported to and from JSON Relational Duality Views as well to and from third-party products, such as MongoDB
- This feature allows developers to use Oracle GoldenGate technology to replicate JSON-relational duality view data **as JSON documents**, instead of relational tables, from an Oracle Database to a target Oracle or non-Oracle database
- Replication of JSON data is an important feature for high availability, fail-over, and real-time migration from a non-Oracle database to Oracle Database (such as from MongoDB)
- Oracle GoldenGate Replication to non-Oracle databases such as MongoDB (a document database) or Redis (a NoSQL key/value store) is simple and performant with the ability to replicate JSON documents from JSON-relational duality views

Duality View support in SQL Developer Extension for VS Code

- Visually construct your JSON objects
- Generate and test your GraphQL / DDL
- Preview your changes by inspecting the resulting JSON responses
- Generate JSON Data Guides
- Accompanied with our Automatic REST API support via ORDS!



Name: BLOG_DEMO_DV

☑ Editionable

DEPARTMENTS	
☑ DEPARTMENT_ID	NUMBER (4)
☑ DEPARTMENT_NAME	VARCHAR2 (30)
□ MANAGER_ID	NUMBER (6)
□ LOCATION_ID	NUMBER (4)

DEPARTMENTS
- _id :DEPARTMENT_ID
- departmentName :DEPARTMENT_NAME

☑ GraphQL Form

```
SELECT * from GRAPHQL('departments {
    _id : department_id
    departmentName : department_name
}')
```

All rows fetched: 27 in 0.114 seconds

	JSON_OBJECT('_ID'VALUE"DEPARTMENT_ID",'DEPARTMENTNAME'VALUE"DEPARTMENT_NAME"NULLONNULL
1	{"_id":10,"departmentName":"Administration"}
2	{"_id":20,"departmentName":"Marketing"}

Mongo-to-Duality Migrator: flex columns, outliers and minFrequency

A duality-view flex column stores (in an underlying table) JSON objects whose fields are not predefined

A field is an outlier for a given document set if it occurs, or if any of its values occurs with a given type, in less than minFrequency percent of the documents

Flex columns

Flex columns and JSON-Relational Duality Views

- A duality-view flex column stores (in an underlying table) JSON objects whose fields aren't predefined: they're not mapped individually to specific underlying columns
- Unrecognized fields of an object in a document you insert or update are automatically added to the flex column for that object's underlying table
- You can thus add fields to the document object produced by a duality view with a flex column underlying that object, without redefining the duality view which provides another kind of schema flexibility to a duality view, and to the documents it supports
- If a given underlying table has no column identified in the view as flex, then new fields are not automatically added to the object produced by that table – thus add flex columns where you want this particular kind of flexibility
- Note that it's technically incorrect to speak of a flex column of a table: a flex column is a duality-view column that's designated as flex — flex for the view

Flex columns

Flex columns and JSON-Relational Duality Views

- Any tables underlying a duality view can have any number of JSON-type columns. At most one JSON column per table can be designated as a flex column at each position where that table is used in the view definition
- If a given table is used only at one place in a view definition (a typical case) then only one flex column for the table can be used
- If the same table is used in N different places in a view definition, then up to N different flex columns for the table can be designated at those places
- You can designate the same flex column to provide the fields for different places of the same document
- Updates to any of the places must concord, by not providing different new fields or different values for the same field

Flex columns

Flex columns and JSON-Relational Duality Views

- In a given duality-view definition, you cannot use the same JSON column as a flex column in one document place and as a non-flex column in another place. An error is raised if you try to do this
- In any table, a JSON column generally provides for flexible data: by default, its typing and structure are not constrained/specified in any way
- The particularity of a JSON column that's designated as a flex column for a duality view is this:
  - The column value must be a JSON object or SQL NULL
  - This means that it must be declared as type JSON (OBJECT), not just JSON - otherwise, an error is raised when you try to use that column in a duality-view definition
- The above rule does not apply to a non-flex JSON-type column; its value can be any JSON value: scalar, array, or object

Outliers and minFrequency

Oracle Database 23ai

- A field is a field-occurrence outlier, or a field outlier, for a given document set if it occurs in less than minFieldFrequency percent of the documents
- A field is a type-occurrence outlier, or type outlier, for a given document set if any of its values occurs with a given type in less than minTypeFrequency percent of the documents
- **minFrequency** (up to 23.6) is the minimum frequency for a field or a type not to be considered an outlier (high-entropy)
- **minFieldFrequency** (23.7) is the minimum frequency for a field not to be considered an outlier (high-entropy)
- **minTypeFrequency** (23.7) is the minimum frequency for the type of a field's value not to be considered an outlier (high-entropy)
- A value of zero (0) percent means that no fields are considered as outliers
- Unrecognized fields of an object in a document you insert or update are automatically added to the flex column for that object’s underlying table

Outliers and minFrequency

Oracle Database 23ai

- A duality-view flex column stores (in an underlying table) JSON objects whose fields are not predefined: they’re not mapped individually to specific underlying columns
- PL/SQL procedure DBMS_JSON_DUALITY.IMPORT_ALL is provided to import multiple document collections into a JSON-relational duality view
- Importing multiple collections in a single PL/SQL call makes the import process simpler and avoids constraint-violation errors
- PL/SQL functions DBMS_JSON_DUALITY.VALIDATE_SCHEMA_REPORT and VALIDATE_IMPORT_REPORT are provided to validate the relational schema and data that are created and imported by the JSON-To-Duality Migrator
- The validation APIs help users verify that the recommended relational schema is correct and there is no data loss, when they migrate their document collections to duality views

Demo

Oracle JSON Autonomous Database

Cloud document database service that makes it simple to develop JSON-centric applications

Oracle Database for JSON storage

Converged Database:

- NoSQL-style document storage
- High-concurrency, low-latency, interactive applications
- Analytics and reporting

Access options

- SQL
- REST
- MongoDB API

Autonomous JSON Database:

Converged Database PLUS

- *Managed* cloud service
- Automatic scaling, backups, and management
- **Faster** and **cheaper** than **MongoDB Atlas**
- Always-free service
- Oracle Autonomous JSON is FedRAMP high (see here), whereas MongoDB is only FedRAMP Moderate (see here) (Federal Risk and Authorization Management Program)

Autonomous JSON Database

Elastic compute and storage

Single-digit latency reads and writes

Highly available

Low-price, always-free tier

Autonomous JSON Database

More than a NoSQL document store

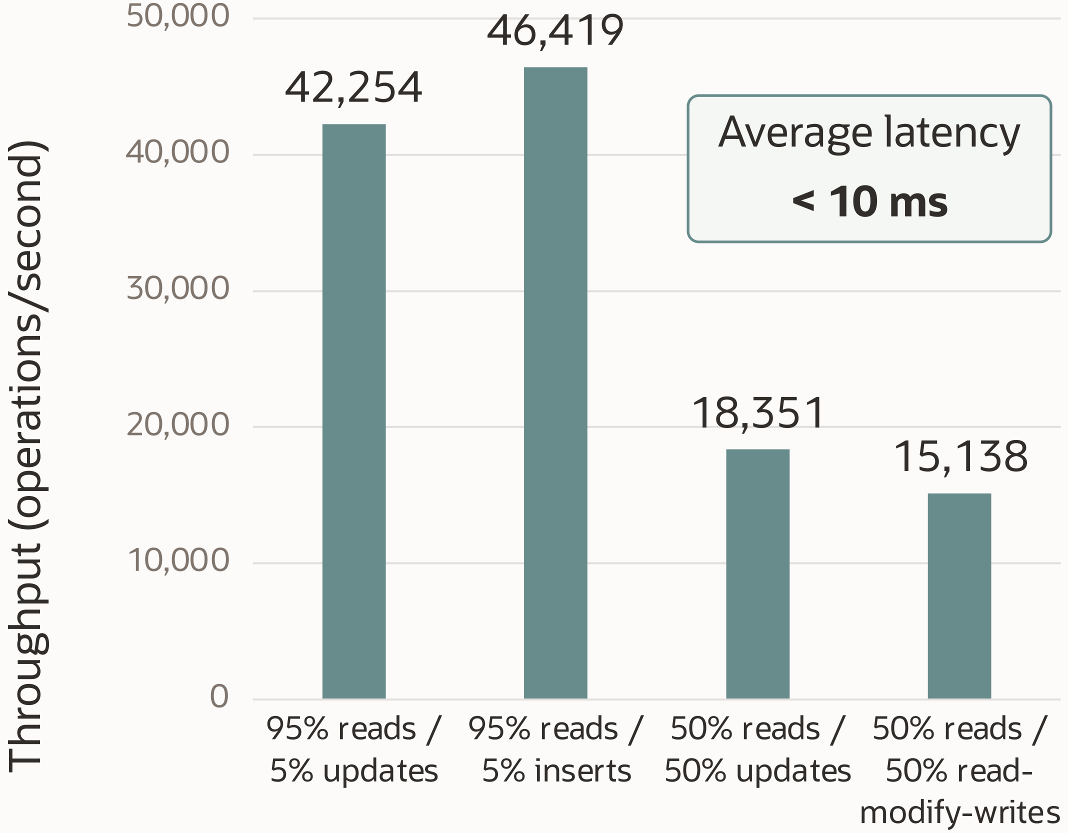
Native and comprehensive SQL support

- ANSI SQL/JSON
- Joins
- Advanced analytics
- Full-text search

Powered by the Oracle Database

- ACID compliant transactions
- Secure by default
- Mission-critical

YCSB NoSQL Benchmark



** Autonomous JSON Database with 8 OCPU running in San Jose region, Mongo API

Autonomous JSON Database: Disaster Recovery

More than a NoSQL document store

As Autonomous Data Guard is currently not supported for Autonomous JSON Database workloads, the natural questions is what are the alternatives:

Option1: The backup-based Disaster Recovery solution is one alternative. Backup-based DR uses database backups to instantiate a peer database at the time of switchover or failover. This enables you to have a lower cost and higher Recovery Time Objective (RTO) disaster recovery option for your Autonomous Database, as compared with Autonomous Data Guard.

Option 2: For having a copy of the Autonomous JSON Database in a different region (and not just in another AD), an option to consider is Refreshable Clones.

Option 3: Oracle GoldenGate is another way to replicate your data do another region. You can add a replicat for Autonomous JSON Database. It is even possible to use Oracle GoldenGate to replicate MongoDB to AJD, good for use case of migrating out of MongoDB to Oracle.

Autonomous JSON Database: the 20GB relational data limit

More than a NoSQL document store

- As Autonomous JSON Database is the same as an Autonomous Transaction Processing database, except that in the Autonomous JSON Database you can store only up to 20 GB of data other than JSON document collections.
- Collections cannot be **heterogeneous**. That is, they can only contain JSON documents. For example, you cannot have a collection of image documents or a collection that contains both JSON documents and image documents.
- You can subscribe to information event **AJDNonJsonStorageExceeded**, to be informed when the 20 GB limit is exceeded.
- This event is generated when an Autonomous JSON Database has exceeded the maximum storage limit of 20GB of data stored outside of SODA collections. This limit does not apply to data stored in SODA collections or objects associated with SODA collections, such as indexes or materialized views.
- In addition to this event, an email is sent to the account owner. You must either reduce your usage of non-SODA-related data to below the 20GB limit or promote the Autonomous JSON Database to Autonomous Transaction Processing.

JSON Collections and the MongoDB API

A native MongoDB API compatible Document Store

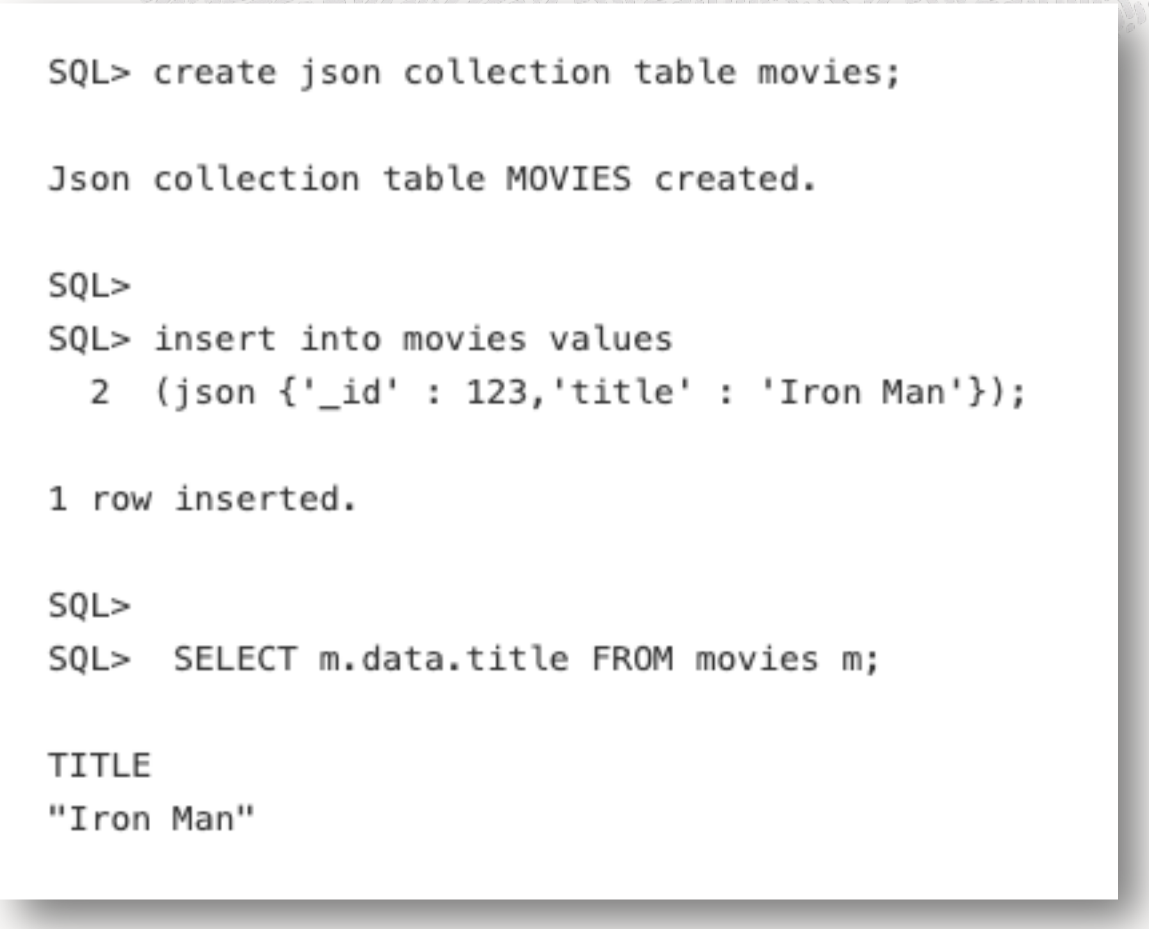
JSON Collections

A **collection** is a special type of **table** or **view** that only stores **JSON**

Each document has an **"_id"** field

- _id is unique, immutable, indexed
- Automatically generated by the database if not set explicitly

Accessible from document APIs including **SODA** (Simple Oracle Document API) and **MongoDB API**



```
SQL> create json collection table movies;

Json collection table MOVIES created.

SQL>
SQL> insert into movies values
  2  (json {'_id' : 123,'title' : 'Iron Man'});

1 row inserted.

SQL>
SQL>  SELECT m.data.title FROM movies m;

TITLE
"Iron Man"
```

Three different types of JSON Collections

	Example	Description
Collection Table	create json collection table movies;	**Native table format.** Backed by OSON storage

Collection View	create json collection view movies as
	  select json
	    {'_id' : t.movieid, 'title' : t.title }
	  from movies_rel t;
		Defined by SQL.
		Read-only.

Duality View	create json duality view movies as
	  select json
	    {'_id' : t.movieid, 'title' : t.title }
	  from movies_rel t with update insert delete;
		Defined by SQL.
		Inserts, updates, and deleted propagated to base tables.

JSON Collections Views and the MongoDB API

Oracle Database 23ai and JSON Collections Views

JSON Collection Views

- Always generated by SQL
  - Maps JSON documents to underlying relational data
- Automatically visible in MongoDB API and as SODA collection

Two flavors

- JSON-relational duality view
  - Fully updatable, both sql (view, underlying tables) and through document store API (collection)
  - `CREATE JSON RELATIONAL DUALITY VIEW`
- Non-duality JSON collection view
  - Read only through document store API (collection) and sql (view), updateable through SQL (table)
  - `CREATE JSON COLLECTION VIEW`

JSON Collections Views and the MongoDB API

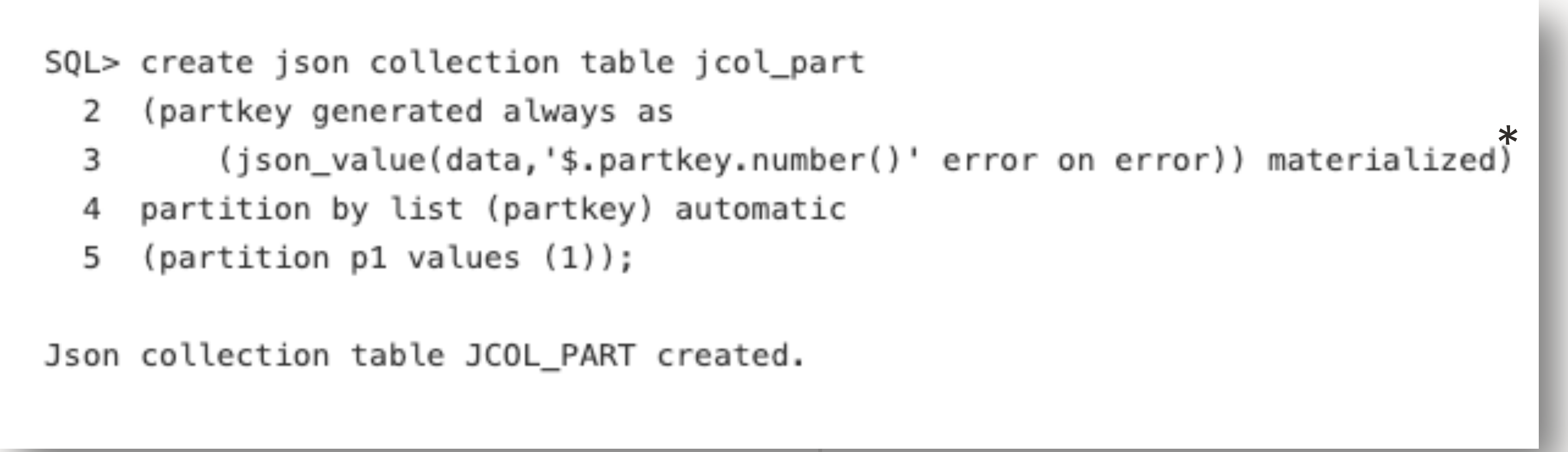
Oracle Database 23ai and JSON Collections Views

DDL Support for JSON Collection tables

- ALTER TABLE MOVE, RENAME
- ADD/DROP CONSTRAINTS
- ADD/DROP VIRTUAL COLUMNS
- Partitioning, Indexing

Data Dictionary views

- *_JSON_COLLECTIONS
- *_JSON_COLLECTION_TABLES
- *_JSON_COLLECTION_VIEWS

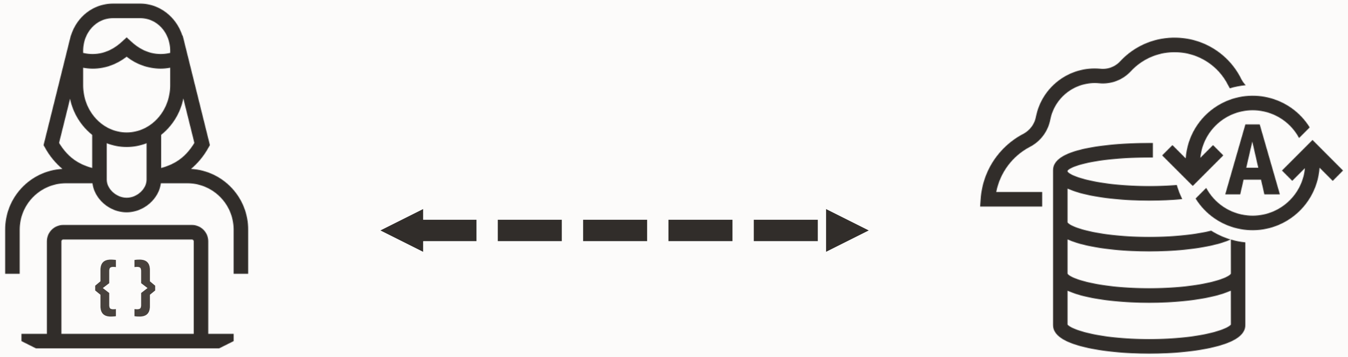


```
SQL> create json collection table jcol_part
     (partkey generated always as
         (json_value(data,'$.partkey.number()' error on error)) materialized)*
     partition by list (partkey) automatic
     (partition p1 values (1));

Json collection table JCOL_PART created.
```

* Materialized expression columns supported in 23.7

Oracle API for MongoDB



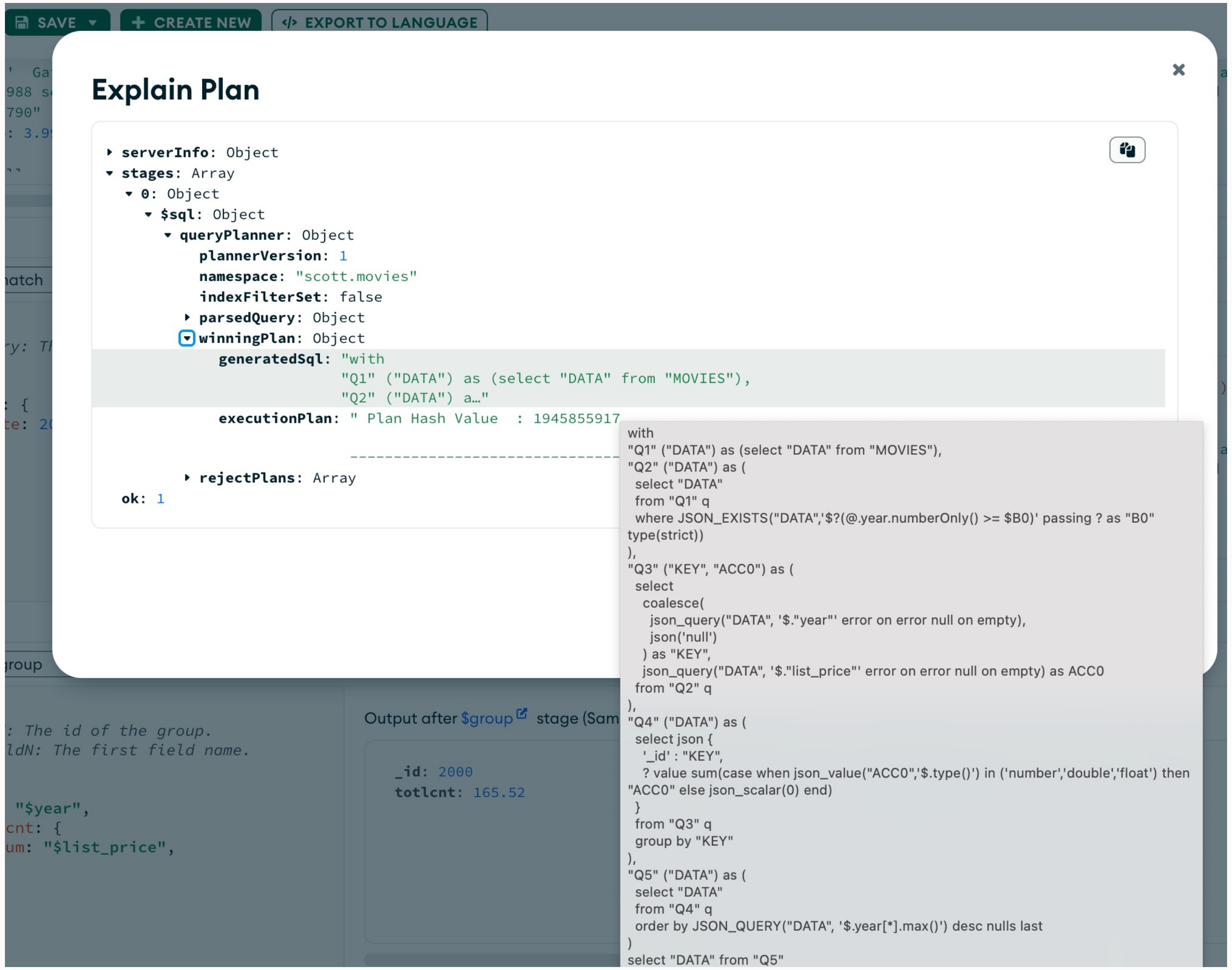
JSON Collection Tables
JSON Collection Views
JSON Relational Duality Views

MongoDB Application

Oracle Database

- Easy migrations MongoDB to Oracle (just change the connection string!)
- MongoDB developers keep using same skills, tools, and frameworks
- No SQL required

MongoDB API Aggregation Pipeline in Oracle Database 23ai



Explain Plan

```
▸ serverInfo: Object
▾ stages: Array
  ▾ 0: Object
    ▾ $sql: Object
      ▾ queryPlanner: Object
          plannerVersion: 1
          namespace: "scott.movies"
          indexFilterSet: false
        ▸ parsedQuery: Object
        ▾ winningPlan: Object
            generatedSql: "with
                          "Q1" ("DATA") as (select "DATA" from "MOVIES"),
                          "Q2" ("DATA") a…"
            executionPlan: " Plan Hash Value  : 1945855917

                           ----------------------------------
        ▸ rejectPlans: Array
  ok: 1
```

```
with
"Q1" ("DATA") as (select "DATA" from "MOVIES"),
"Q2" ("DATA") as (
 select "DATA"
 from "Q1" q
 where JSON_EXISTS("DATA",'$?(@.year.numberOnly() >= $B0)' passing ? as "B0"
type(strict))
),
"Q3" ("KEY", "ACC0") as (
 select
  coalesce(
   json_query("DATA", '$."year"' error on error null on empty),
   json('null')
  ) as "KEY",
  json_query("DATA", '$."list_price"' error on error null on empty) as ACC0
 from "Q2" q
),
"Q4" ("DATA") as (
 select json {
  '_id' : "KEY",
  ? value sum(case when json_value("ACC0",'$.type()') in ('number','double','float') then
"ACC0" else json_scalar(0) end)
 }
 from "Q3" q
 group by "KEY"
),
"Q5" ("DATA") as (
 select "DATA"
 from "Q4" q
 order by JSON_QUERY("DATA", '$.year[*].max()') desc nulls last
)
select "DATA" from "Q5"
```

Output after $group stage (Sam

```
_id: 2000
totlcnt: 165.52
```

Translated to Oracle analytical SQL

- Mongo pipelines operations equivalent to rudimentary SQL operations
- **Translation to Oracle SQL streamlines processing and improves performance**

More than 85% of stages, aggregators, and accumulators already implemented and working

- E.g. $count, $kip, $out, $group, $match, $avg, $max, $min, $sum, …

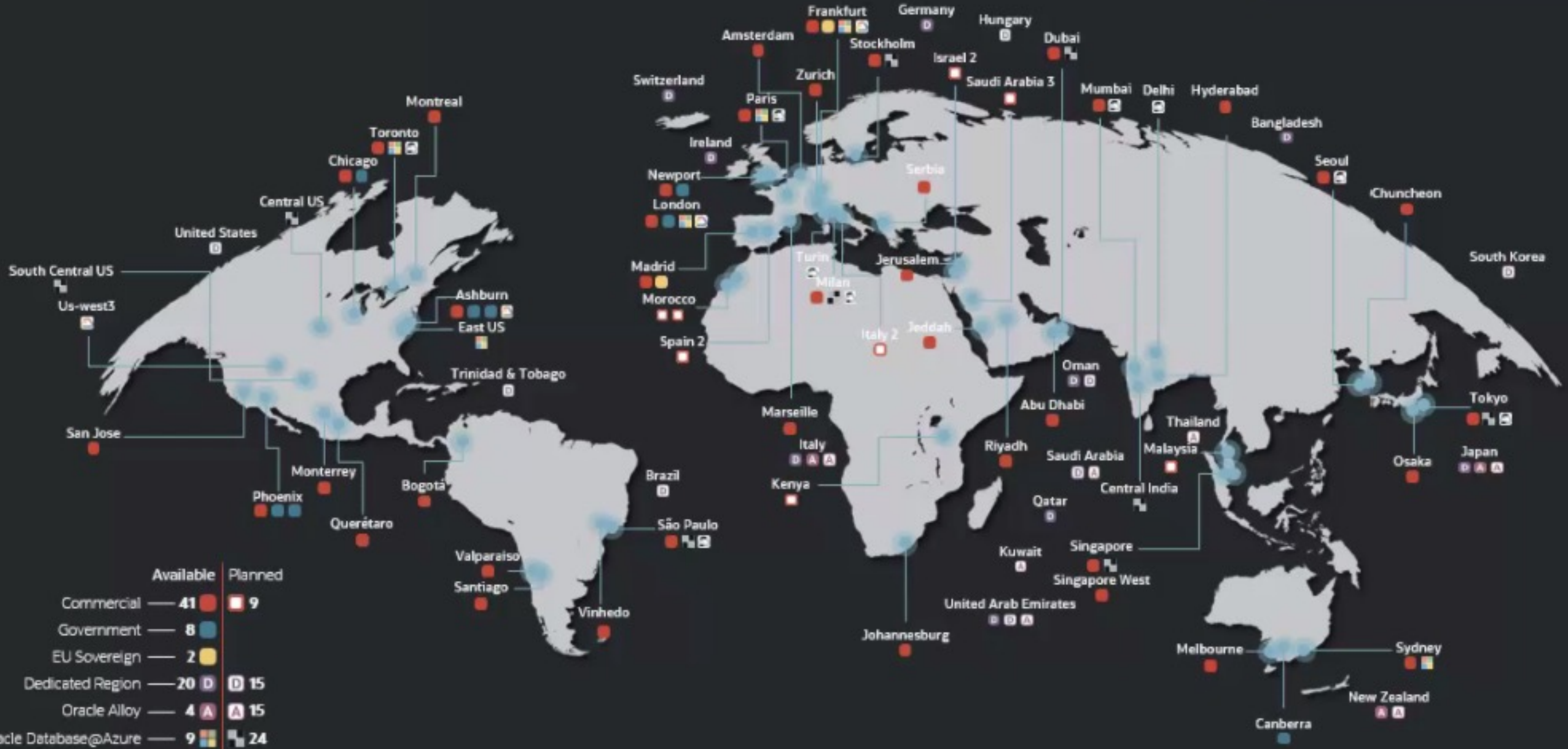
Oracle-proprietary Mongo-compatible aggregation stages

- **$sql, $external**

Demo

November 2024 – 168 Live or Planned OCI Regions

100% renewable energy for commercial and EU Sovereign regions by 2025



	Available	Planned
Commercial	41 🟥	□ 9
Government	8 🟦	
EU Sovereign	2 🟨	
Dedicated Region	20 D	D 15
Oracle Alloy	4 A	A 15
Oracle Database@Azure	9	24
le Database@Google Cloud	4	14
Oracle Database@AWS	1	

[1]Not all regions listed on map

Summary

1 Leverage the full power of standard SQL on JSON documents and relational data with a single execution engine

2 Store, use, and manage relational data and JSON documents in a single converged database. Unified management, security, consistency model

3 Flexible access with relational and document-store APIs and languages, like SQL, JDBC, MongoDB API, Python, and Oracle SODA

Where To Get More Information

- Live Lab: Developing with JSON and SODA
- Live Lab: Using the Database API for MongoDB
- LiveSQL: SQL/JSON features
- O.com: JSON-based Development in Oracle Database
- O.com: Autonomous JSON Database
- Documentation: JSON Developer's Guide
- Documentation: Overview of Oracle Database API for MongoDB
- @bch_t @sdjh2000 @juliandontcheff @OracleDatabase
- DW-PM_us@oracle.com

Try it For Free

free-oracle.github.io

cloud.oracle.com/free

oracle.com/xe

developer.oracle.com/livelabs

It’s now time for Q&A

Got any questions?

Q&A Open

Important links to bookmark

Links to get you started and to keep up to date with Autonomous Database

1 New Get Started page:
oracle.com/autonomous-database/get-started/

2 Join us:

LinkedIn
bit.ly/adb-linkedin-grp

X
@AutonomousDW

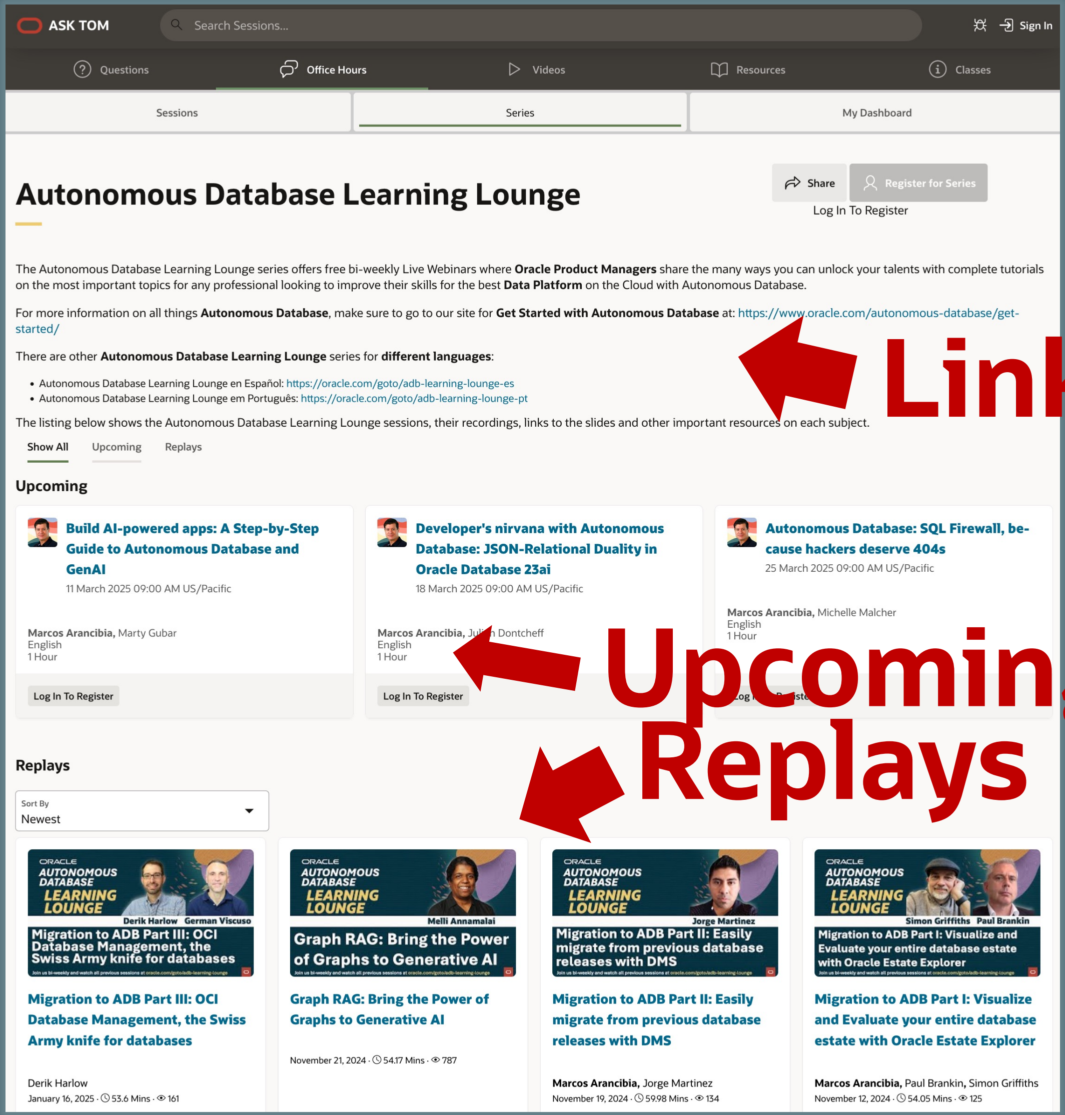
Bluesky
autonomousdb.bsky.social

3 Got a question?
We are on stackoverflow
bit.ly/adb-stackoverflow

Join us on Developers Slack
(search #oracle-autonomous-database)
bit.ly/odevrel_slack (odevrel_slack)

Final Thoughts

oracle.com/goto/adb-learning-lounge



Autonomous Database Learning Lounge

Log In To Register

The Autonomous Database Learning Lounge series offers free bi-weekly Live Webinars where **Oracle Product Managers** share the many ways you can unlock your talents with complete tutorials on the most important topics for any professional looking to improve their skills for the best **Data Platform** on the Cloud with Autonomous Database.

For more information on all things **Autonomous Database**, make sure to go to our site for **Get Started with Autonomous Database** at: https://www.oracle.com/autonomous-database/get-started/

There are other **Autonomous Database Learning Lounge** series for **different languages**:

- Autonomous Database Learning Lounge en Español: https://oracle.com/goto/adb-learning-lounge-es
- Autonomous Database Learning Lounge em Português: https://oracle.com/goto/adb-learning-lounge-pt

The listing below shows the Autonomous Database Learning Lounge sessions, their recordings, links to the slides and other important resources on each subject.

Show All Upcoming Replays

Upcoming

Build AI-powered apps: A Step-by-Step Guide to Autonomous Database and GenAI
11 March 2025 09:00 AM US/Pacific
Marcos Arancibia, Marty Gubar
English
1 Hour
Log In To Register

Developer's nirvana with Autonomous Database: JSON-Relational Duality in Oracle Database 23ai
18 March 2025 09:00 AM US/Pacific
Marcos Arancibia, Julian Dontcheff
English
1 Hour
Log In To Register

Autonomous Database: SQL Firewall, because hackers deserve 404s
25 March 2025 09:00 AM US/Pacific
Marcos Arancibia, Michelle Malcher
English
1 Hour
Log In To Register

Replays

Sort By
Newest

ORACLE AUTONOMOUS DATABASE LEARNING LOUNGE
Derik Harlow German Viscuso
Migration to ADB Part III: OCI Database Management, the Swiss Army knife for databases

Migration to ADB Part III: OCI Database Management, the Swiss Army knife for databases
Derik Harlow
January 16, 2025 · 53.6 Mins · 161

ORACLE AUTONOMOUS DATABASE LEARNING LOUNGE
Melli Annamalai
Graph RAG: Bring the Power of Graphs to Generative AI

Graph RAG: Bring the Power of Graphs to Generative AI
November 21, 2024 · 54.17 Mins · 787

ORACLE AUTONOMOUS DATABASE LEARNING LOUNGE
Jorge Martinez
Migration to ADB Part II: Easily migrate from previous database releases with DMS

Migration to ADB Part II: Easily migrate from previous database releases with DMS
Marcos Arancibia, Jorge Martinez
November 19, 2024 · 59.98 Mins · 134

ORACLE AUTONOMOUS DATABASE LEARNING LOUNGE
Simon Griffiths Paul Brankin
Migration to ADB Part I: Visualize and Evaluate your entire database estate with Oracle Estate Explorer

Migration to ADB Part I: Visualize and Evaluate your entire database estate with Oracle Estate Explorer
Marcos Arancibia, Paul Brankin, Simon Griffiths
November 12, 2024 · 54.05 Mins · 125

Links

Upcoming

Replays

Thank you for joining !!!

AUTONOMOUS DATABASE

LEARNING LOUNGE