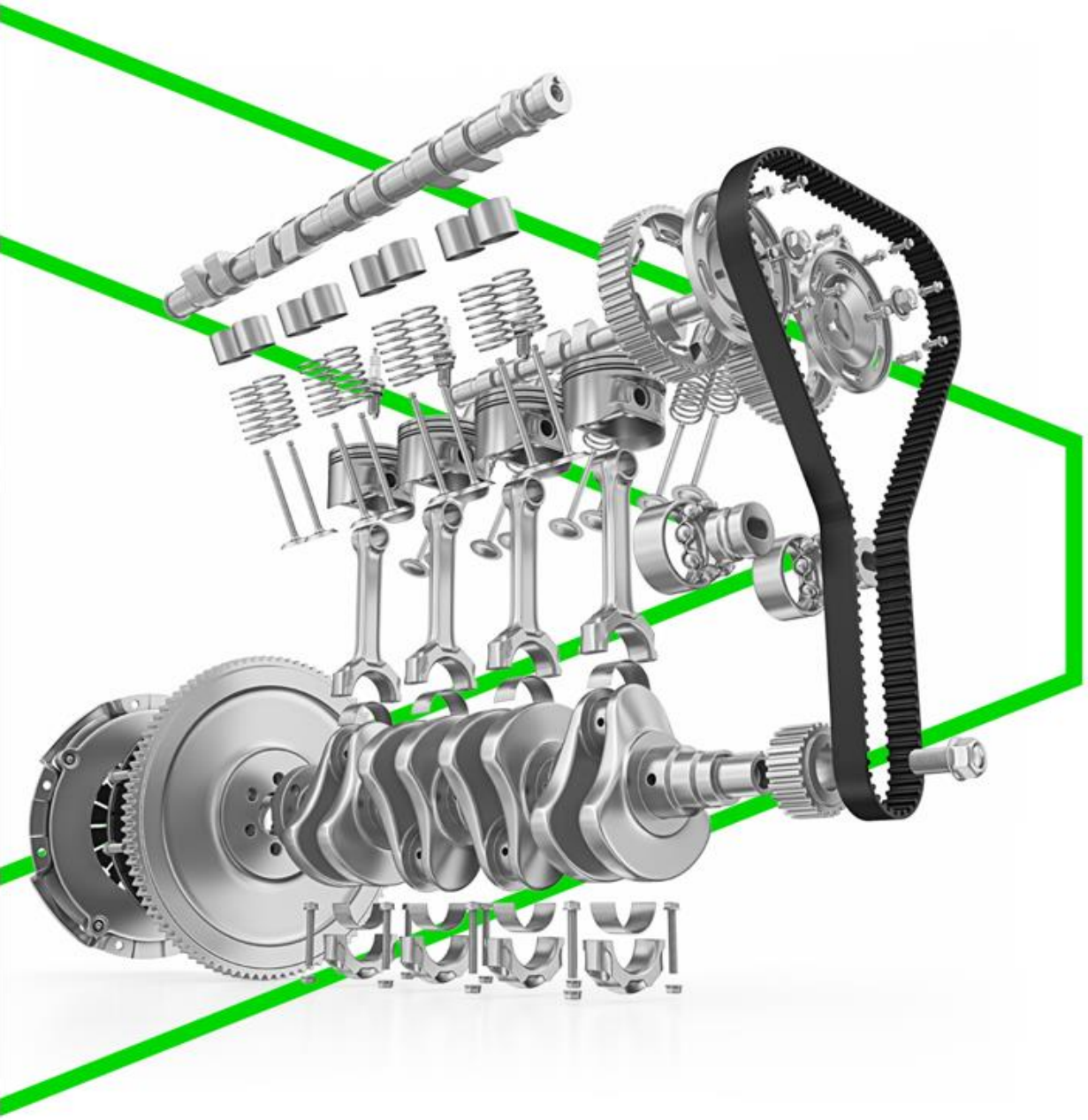


accenturetechnology

Accenture Enkitec Group

# What is new in the relational database world

**Julian Dontcheff**

**June 2nd, 2022**

ORACLE ACE Director

ORACLE Certified Master

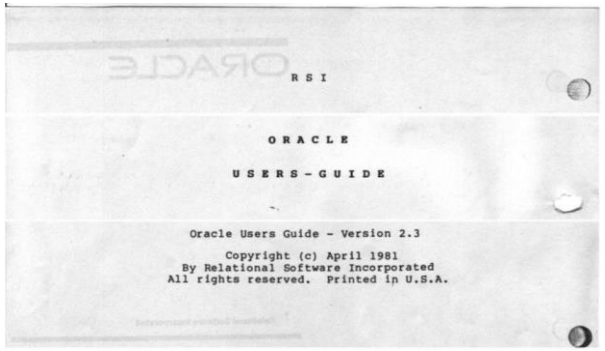
Julian Dontcheff

Managing Director

accenture

- More than 30 years of Database Experience
- Accenture Global Database Lead
- First Oracle Certified Master in Europe: 2002
- Oracle ACE Director
- More than 10.000 hours of on-call DBA duty
- Database Blog at: juliandontcheff.wordpress.com

# THE ORACLE AUTONOMOUS CLOUD
# ADW, AJD and ATP

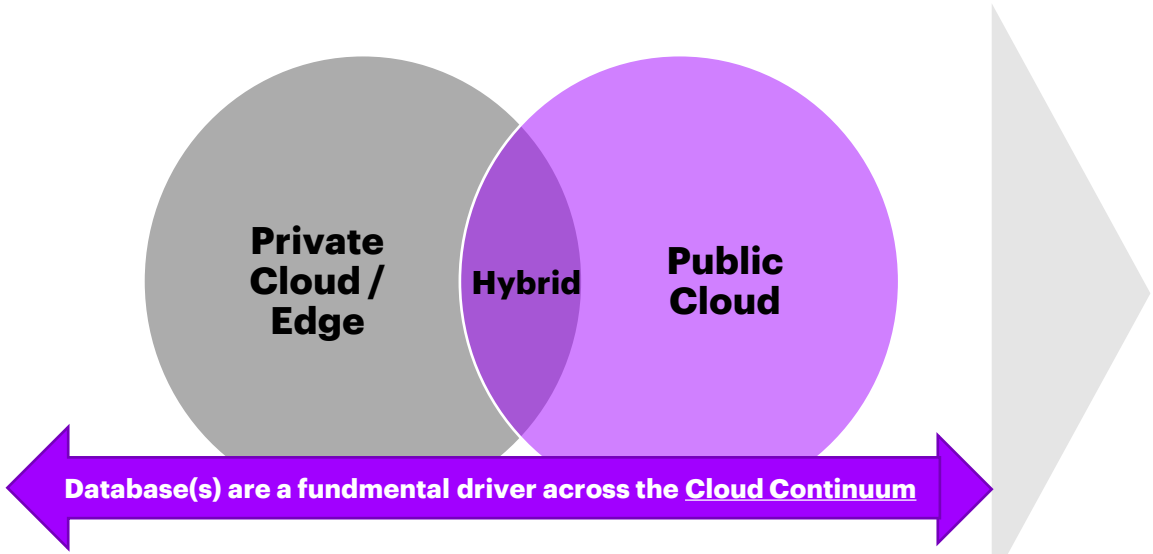




# Oracle
# Db2
# SQL Server
# PostgreSQL
# MariaDB
# Redshift
# Sybase
# Snowflake
# SAP HANA

# 2020: EXPLOSION OF DATABASE TECHNOLOGY

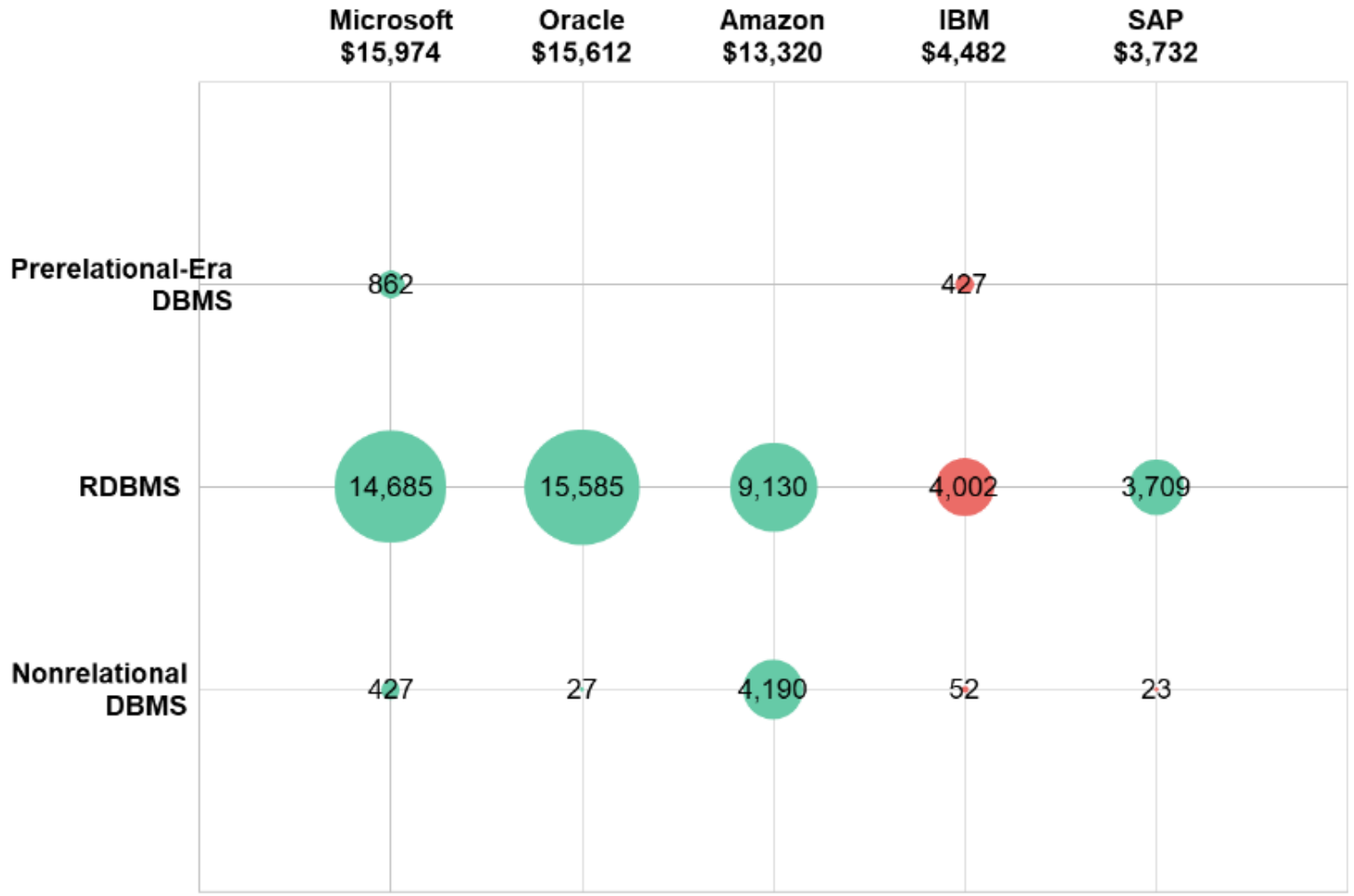
## Databases are the Foundation of all Clouds




## The Industry is in transition from legacy DB technologies to modern Cloud Native DB platforms

- 80% of organizations **will migrate entirely away from on-premises data centers** by 2025 [Gartner]
- The DBMS market saw strong growth of 17.1% to reach nearly $65 billion in 2020… 90% of the DBMS growth **came from dbPaaS** in 2020 [Gartner]
- The nonrelational (NoSQL) DBMS segment grew by 34.5% to take 15.2% share of the DBMS market, while the relational DBMS segment grew by 15.2% for a segment share of 82.0% [Gartner]
- While the nonrelational segment is the **fastest growing** segment by percentage terms, the **actual revenue growth** in relational DBMS in dollars is roughly **three times** the revenue growth in the nonrelational DBMS segment [Gartner]

# THE GROWTH OF THE DATABASE MARKET: TOP 5 DB PROVIDERS




Source: Gartner (June 2021)

# CLOUD ECOSYSTEM IS MOVING FAST

Cloud providers are rapidly developing new Cloud Native Database platforms

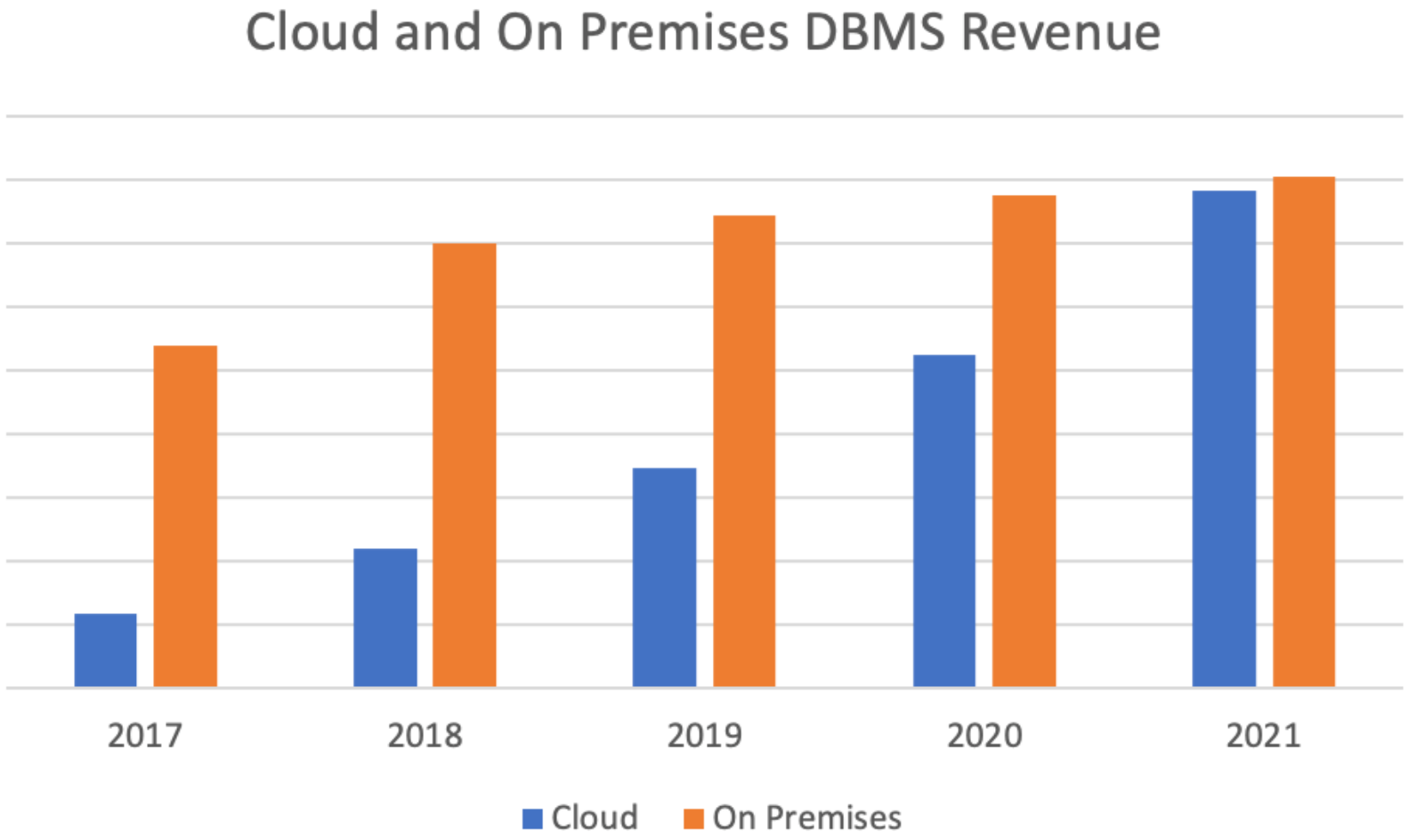
- The composition of the top five remained the same as in 2019. However, Microsoft is now the top DBMS vendor by market share, surpassing Oracle for the first time, albeit by only about $300 million. The top five vendors accounted for 82% of the DBMS revenue in 2020.
- **Oracle** continued its positive DBMS revenue growth, growing at 3.0% in 2020, driven by growth in its cloud services revenue. However, due to growing much below market growth, its share of the overall market has continued to decline, now at 24.1%, versus 27.4% in 2019.
- **Microsoft** grew close to the market growth in 2020, at 17.0%, which was still lower than the growth of some of its top competitors. Here too, the overall growth is driven by its cloud service revenue growth of 54% in 2020.
- **Amazon** saw above-market growth of 41%. Amazon remains top-ranked by dbPaaS market share.
- **Google** continued on its growth trajectory, growing at 88% in 2020. This is the second highest growth among all DBMS vendors that Gartner track for revenue, and comfortably the highest growth in the top 10 vendors by market share.
- **Alibaba** surpassed Teradata to jump one position to No. 7, due to its 70% DBMS revenue growth in 2020.

# Competitive landscape for Database technologies continue to move

## Database Market May 2022 Summary

- The string of growth for the DBMS market continued – and accelerated – as the market approached $80B, with an increase of $14,5B over 2020
- That is an unprecedented 22.3% increase from a year that itself was up 19% over the prior year
- Since its brief pause in 2015, the DBMS market has reeled off 6 consecutive years of growth, with the last 5 years all growing in the teens
- Since 2017's $38.6B year, the DBMS market has added $40B – doubling in 5 years
- 2021 also marked the second consecutive year of DBMS market position loss for Oracle, as it dropped to third place
- This is attributable to cloud results as well – Oracle's revenue in the cloud has grown well below market rates
- Others have fared better: Microsoft's cloud revenue growth came in at 39.5%, pushing its overall growth to 20.9%, just below the market's rate of 22.3%
- AWS – all cloud – rose 42.3%, nearly doubling the market rate
- New entrant Google, also an all-cloud player, with its 64% growth, nearly tripled the market rate
- Alibaba approached the $2B mark in 2021, growing 32.8% to $1.8B

# Competitive landscape for Database technologies continue to move




## Database Market - Summary

- The biggest DBMS market story continues to be the enormous impact of revenue shifting to the cloud
- In 2021, revenue for managed cloud services (dbPaaS) rose to $39.2B – it now represents over 49% of all DBMS revenue
- The growth has been stunning!

# Competitive landscape for Database technologies continue to move

- The results for the leading vendors continue to demonstrate the enormous forces wrought by the transition to cloud-based systems
- Cloud platform providers lead the way and continue to make gains based on their performance there
- Four vendors moved on the DBMS market leaderboard in 2021, an unprecedented show of volatility at the top
- Google entered the top five, displacing SAP and pushing IBM into fifth place – a long drop from its second-place standing only eight years ago in 2013
- AWS moved up for the third time in the past five years, taking over second place from Oracle and pulling to a near tie with Microsoft – the two vendors are separated by $65M, or 0.37% of Microsoft's revenue

| 2017 | | 2018 | | 2019 | | 2020 | | 2021 | |
|---|---|---|---|---|---|---|---|---|---|
| Vendor | Share | Vendor | Share | Vendor | Share | Vendor | Share | Vendor | Share |
| Oracle | 36.1% | Oracle | 31.1% | Oracle | 27.4% | Microsoft | 24.3% | Microsoft | 24.0% |
| Microsoft | 21.5% | Microsoft | 23.6% | Microsoft | 24.7% | Oracle | 23.8% | AWS | 23.9% |
| IBM | 12.7% | AWS | 13.5% | AWS | 17.1% | AWS | 20.6% | Oracle | 20.6% |
| AWS | 9.2% | IBM | 10.4% | IBM | 8.8% | IBM | 6.8% | Google | 6.5% |
| SAP | 7.4% | SAP | 6.9% | SAP | 6.5% | SAP | 5.6% | IBM | 5.6% |

# DB-Engines Ranking of all Databases – June 2022

| Rank | | | DBMS | Database Model | Score | | |
|---|---|---|---|---|---|---|---|
| Jun 2022 | May 2022 | Jun 2021 | | | Jun 2022 | May 2022 | Jun 2021 |
| 1. | 1. | 1. | Oracle | Relational, Multi-model | 1287.74 | +24.92 | +16.80 |
| 2. | 2. | 2. | MySQL | Relational, Multi-model | 1189.21 | -12.89 | -38.65 |
| 3. | 3. | 3. | Microsoft SQL Server | Relational, Multi-model | 933.83 | -7.37 | -57.25 |
| 4. | 4. | 4. | PostgreSQL | Relational, Multi-model | 620.84 | +5.55 | +52.32 |
| 5. | 5. | 5. | MongoDB | Document, Multi-model | 480.73 | +2.49 | -7.49 |
| 6. | 6. | ↑ 7. | Redis | Key-value, Multi-model | 175.31 | -3.71 | +10.06 |
| 7. | 7. | ↓ 6. | IBM Db2 | Relational, Multi-model | 159.19 | -1.14 | -7.85 |
| 8. | 8. | 8. | Elasticsearch | Search engine, Multi-model | 156.00 | -1.70 | +1.29 |
| 9. | 9. | ↑ 10. | Microsoft Access | Relational | 141.82 | -1.62 | +26.88 |
| 10. | 10. | ↓ 9. | SQLite | Relational | 135.44 | +0.70 | +4.90 |
| 11. | 11. | 11. | Cassandra | Wide column | 115.45 | -2.56 | +1.34 |
| 12. | 12. | 12. | MariaDB | Relational, Multi-model | 111.58 | +0.45 | +14.79 |
| 13. | ↑ 14. | ↑ 26. | Snowflake | Relational | 96.42 | +2.91 | +61.67 |

# DB-Engines Ranking of Relational Databases – June 2022

| Rank | | | DBMS | Database Model | Score | | |
|---|---|---|---|---|---|---|---|
| Jun 2022 | May 2022 | Jun 2021 | | | Jun 2022 | May 2022 | Jun 2021 |
| 1. | 1. | 1. | Oracle | Relational, Multi-model | 1287.74 | +24.92 | +16.80 |
| 2. | 2. | 2. | MySQL | Relational, Multi-model | 1189.21 | -12.89 | -38.65 |
| 3. | 3. | 3. | Microsoft SQL Server | Relational, Multi-model | 933.83 | -7.37 | -57.25 |
| 4. | 4. | 4. | PostgreSQL | Relational, Multi-model | 620.84 | +5.55 | +52.32 |
| 5. | 5. | 5. | IBM Db2 | Relational, Multi-model | 159.19 | -1.14 | -7.85 |
| 6. | 6. | ↑ 7. | Microsoft Access | Relational | 141.82 | -1.62 | +26.88 |
| 7. | 7. | ↓ 6. | SQLite | Relational | 135.44 | +0.70 | +4.90 |
| 8. | 8. | 8. | MariaDB | Relational, Multi-model | 111.58 | +0.45 | +14.79 |
| 9. | 9. | ↑ 16. | Snowflake | Relational | 96.42 | +2.91 | +61.67 |
| 10. | 10. | 10. | Microsoft Azure SQL Database | Relational, Multi-model | 86.01 | +0.68 | +11.22 |
| 11. | 11. | ↓ 9. | Hive | Relational | 81.58 | -0.03 | +1.89 |
| 12. | 12. | ↓ 11. | Teradata | Relational, Multi-model | 70.41 | +2.02 | +1.07 |
| 13. | 13. | ↓ 12. | SAP HANA | Relational, Multi-model | 54.53 | -0.56 | +0.42 |

# What is new in SAP HANA 2.0 SPS 06 – 03.12.2021 (2010)

- High availability for SAP HANA cockpit and fast restart option
- Simplified ability to fixing logical errors (e.g. dropped tables) with new Secondary Time Travel, without the need of a point-in-time recovery on alternative hardware
- The number of records in a table or partition in SAP HANA can't exceed 2 billion (more precisely 2.147.483.648 – 2 to the power of 31) - you can solve this problem by table partitioning (SAP Note 2044468) or deletion of no longer required records
- Enhanced autonomous statement classification
- Hybrid extensions to the cloud - tunnel between Cloud and on-prem

# What is new in Db2 z/OS 12.1 and Db2 11.5 - 27.6.2019 (1983/93)

- Catalog migration is accomplished in a single phase
- Profiles take effect immediately when Db2 is started
- Db2 12 introduces more information to aid users in diagnosing problems with real-time statistics collection
- Db2 12 supports compressing LOB data
- In Db2 12, you can assign objects to in-memory buffer pools
- Db2 12 provides the buffer pool advisory mode
- IBM i 7.5, previously know as AS/400, has new BOOLEAN data type and the maximum size of a binary radix index is extended, up to 16 TB

# What is new in SQL Server 2019 4.11.2019 (1989)

- You can also develop applications with SQL Server on Linux, Windows, Ubuntu, or Docker and deploy them on these platforms
- Resumable online index rebuild and New Polybase connectors for SQL Server, Oracle, Teradata and MongoDB
- Automatic database tuning & plan correction: ALTER DATABASE CURRENT SET AUTOMATIC_TUNING (FORCE_LAST_GOOD_PLAN = ON);
- Automatic index management (available only in Azure SQL Database)
- Static data masking - which allows for masked data to be persisted to other environments

# What is new in MariaDB 10.8.3 - 10.5.2022 (2009)

- EXECUTE IMMEDIATE statement (10.2.3)
- User password expiry and sequences and histograms collected
- Added instant DROP COLUMN and changing of the order of columns
- Oracle compatible SUBSTR() function and INVISIBLE columns
- The MariaDB SQL/PL stored procedure dialect (enabled with sql_mode=ORACLE) now supports Oracle style packages
- MariaDB ColumnStore storage engine
- Oracle Compatibility: ADD_MONTHS(), TO_CHAR(), MINUS is mapped to EXCEPT in UNION, ROWNUM, Atomic DDL

# What is new in PostgreSQL 14.3 - 12.5.2022 (1996)

- Parallel B-tree Index build and parallel index vacuuming
- Stored procedures with transaction control
- psql - the addition of the intuitive quit and exit commands to safely leave this popular command line client
- Automatic index creation
- Allow DROP DATABASE to disconnect sessions using the target database, allowing the drop to succeed
- Extended statistics can now be collected on expressions
- Stored procedures can now return data via OUT parameters

# What is new in Sybase ASE 16.0 - 1.4.2014 (1987)

- DBAs can use sp_configure and set index compression to enable or disable index compression
- Support “create or replace” command for database objects
- Partition-level Locking: large, high impact DML and DDL statements can execute on different partitions of the same table simultaneously
- Full-Database Encryption and Compression Advisor Utility
- Multiple-trigger support: a new feature in SAP ASE 16, allows up to 50 triggers to be fired for a single DML statement

# What is new in MySQL 8.0.29 - 26.4.2022 (1995)

- Multithreading is now enabled by default for replica servers
- CREATE USER and ALTER USER syntax has been extended to permit specification of multiple authentication methods
- The BLACKHOLE storage engine maximum key length has been increased to 3072 bytes to be same as InnoDB
- InnoDB: The new innodb_segment_reserve_factor system variable permits configuring the percentage of tablespace file segment pages that are reserved as empty pages (thanks to Facebook)
- Docker containers for MySQL EE no longer need root privileges

# What is new in MongoDB 5.0.8 - 25.4.2022 (2009)

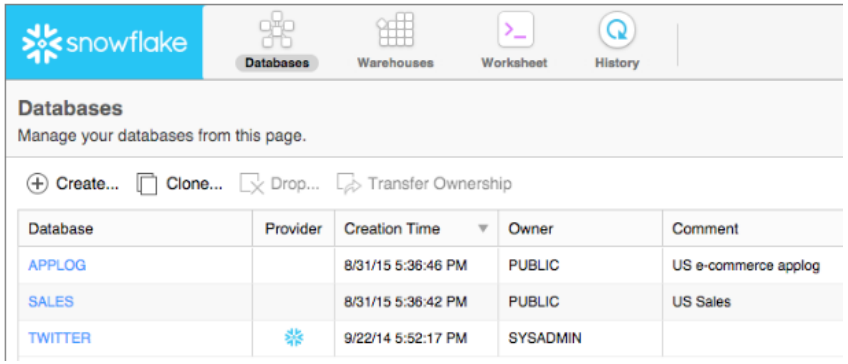
- ACID Guarantees: distributed, cross-shard transactions, global point-in-time reads, and mutable shard key values
- Full-text search service with Kubernetes and Apache Kafka integrations
- Client-side field level encryption and 3x lower auditing overhead
- Simultaneous Indexing and Streaming Replication that reduce replica lag to serve fresher data to your users
- Extend MongoDB with your own functions
- Future-proofs versioned API, client-side field level encryption, live resharding, time series support

# What is new in Amazon Redshift (2012)

- Amazon Redshift is based on an older version of PostgreSQL 8.0.2
- Amazon Redshift now updates table statistics by running ANALYZE automatically
- Amazon Redshift now supports stored procedures: since May 17, 2019
- Amazon Redshift now automatically runs the VACUUM DELETE operation to reclaim disk space occupied by rows that were marked for deletion by the previous UPDATE and DELETE operations
- AQUA (Advanced Query Accelerator) for Amazon Redshift is now generally available in Stockholm, Seoul and N. California

# What is new in Snowflake Elastic Data Warehouse 3.27 (2012)

- Automatic data encryption by Snowflake using Snowflake-managed keys
- Use Snowpipe to load data
- SnowSQL - Python-based command line




- Web-based GUI for account and general management, monitoring of resources and system usage, and querying data
- Snowflake Fail-safe for DR of historical data
- Automatic Query Rewrites Using Materialized Views (Dec 2020)

# NewSQL: NoSQL for OLTP + ACID

- Scalability of NoSQL systems for online transaction processing (OLTP) workloads while maintaining the ACID guarantees of a traditional database system (Atomicity Consistency Isolation Durability)
- The two common distinguishing features are that they support the relational data model (including ACID consistency) and use SQL as their primary interface
- Example systems in this category are Amazon Aurora, Google Spanner, TiDB, **CockroachDB**, Altibase, Apache Ignite, GridGain, Clustrix, VoltDB, MemSQL, NuoDB, **YugaByte DB** and Trafodion

# What is new in Oracle 21c

- Native Blockchain Tables
- Automatic Index Optimization
- Automatic Zone Maps
- Automatic SQL Tuning Sets
- Automatic Indexing Enhancements
- Autonomous Health Framework  and Object Activity Tracking System
- Automatic Materialized Views
- Auto-Result Cache

# Blockchain Tables

# Blockchain Tables in Oracle Database 21c

## How to create a blockchain table?

- Blockchain tables are insert-only tables that organize rows into a number of chains and is a new concept starting with Oracle 21c – backported to 19.10
- Each row in a chain, except the first row, is chained to the previous row in the chain by using a cryptographic hash - if rows are not chained then the name is immutable table
- For each Oracle RAC instance a blockchain table contains thirty-two chains, ranging from 0 through 31

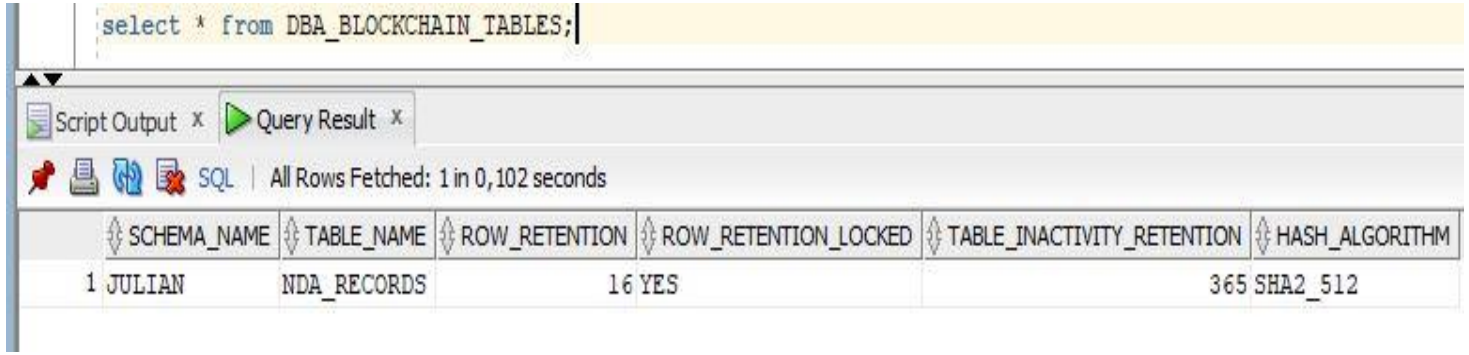
```
SQL> show user
USER is "JULIAN"

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.
```

Ever.

# Blockchain Tables in Oracle Database 21c

## Main restrictions:

- 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!




Ever.

# Blockchain Tables Oracle Database 21c

## 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.

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.

Ever.

# Blockchain Tables Oracle Database 21c

## The following 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)
- Use DBMS_BLOCKCHAIN_TABLE.DELETE_EXPIRED_ROWS to remove rows that are beyond the retention period of the blockchain table.

Ever.

# Automatic Index Optimization

# Automatic Index Optimization in Oracle 21c

## Heat Maps and Information Lifecycle Management:

- Oracle database 21c came with 214 new features and one of them is directly related to the indexes in the database (138 in 20c)
- It is called Automatic Index Optimization
- Automatic Index Optimization does not mean optimization of the Automatic Indexes in the database but rather making now the Index Optimization an automatic process.
- First, in order to implement an ILM strategy, you have to enable Heat Maps in the database to track data access and modification. You can enable and disable heat map tracking at the system or session level with the ALTER SYSTEM or ALTER SESSION statement using the HEAT_MAP init.ora parameter, for example:
- SQL> alter system set HEAT_MAP = ON;

Ever.

# Automatic Index Optimization in Oracle 21c

## Two options for ADO polices:

- Like Automatic Data Optimization (ADO) for data segments, Automatic Index Optimization works via ILM on indexes by enabling policies that automatically optimize indexes by compressing, shrinking and rebuilding them

- You have to add ADO policies for indexes in order to enable their compression and optimization using the existing Automatic Data Optimization (ADO) framework

- There are 2 options:

  – ADD POLICY TIER in order to perform the operation on a say low cost/ tier 2 tablespace when tier 1 storage is under space pressure
  – ADD POLICY OPTIMIZE in order to kick off the process after a certain number of days passes without accessing the index

Ever.

# Automatic Index Optimization in Oracle 21c

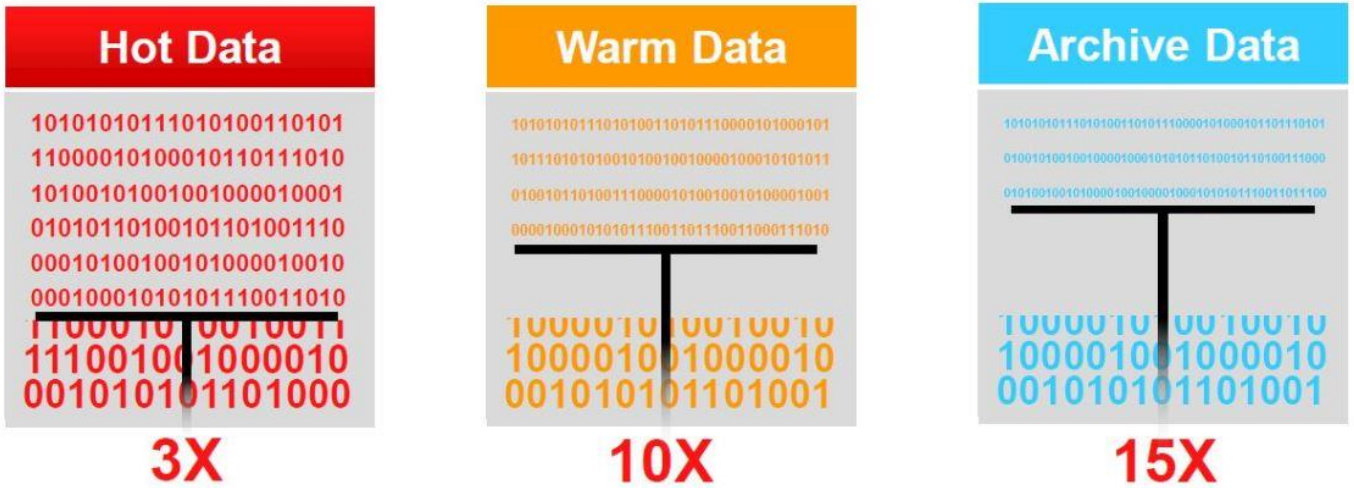
## Here are few examples:

```
SQL> create index julian.price_idx ON julian.sales(price)
ILM ADD POLICY OPTIMIZE AFTER 31 DAYS OF NO MODIFICATION;

SQL> alter index julian.price_idx ILM ADD POLICY TIER TO BC_DATA;

SQL> alter index julian.price_idx
ILM ADD POLICY OPTIMIZE AFTER 3 DAYS OF NO ACCESS;

SQL> SELECT POLICY_NAME, POLICY_TYPE, ENABLED FROM DBA_ILMPOLICIES;

POLICY_NAME             POLICY_TYPE    ENA
----------------------  -------------  ---
P1                      DATA MOVEMENT  YES
```

Ever.

# Automatic Index Optimization in Oracle 21c

- Such policies for indexes on partition level are not yet supported
- The ADO policy is cascaded to all partitions
- So, if we have hybrid tables in the Cloud, we cannot move local indexes automatically to object storage but it should work for global indexes
- Note that we can use Automatic Data Optimization (ADO) policies with hybrid partitioned tables under some conditions
- EE, ODA, and Exa: Requires the Oracle Advanced Compression option




Ever.

# Automatic Index Optimization in Oracle 21c

## The optimization process includes 3 actions

- Compress: Compresses portions of the key values in an index segment (~3 times)
- Shrink: Merges the contents of index blocks where possible to free blocks for reuse
- Rebuild: Rebuilds an index to improve space usage and access speed


Ever.

# Automatic Zone Maps

# Automatic Zone Maps in Oracle Database 21c

## What is a Zone Map?

- A zone is a set of a contiguous data blocks on disk
- A zone map is an index-like structure built on a table and stores information about the zones of that table
- There are 2 major differences between indexes and zone maps:
  - A zone map stores information per zone instead of per row which makes it much more compact than an index
  - A zone map is not actively managed the way an index is kept in sync with the DML on the table
- Zone maps are closer as a concept to Exadata's storage indexes than to B-tree indexes

Ever.

# Automatic Zone Maps in Oracle Database 21c

## What is a Zone Map? Example:

- Consider a small table containing basic information about some relational databases from db-engines.com (rank, score, initial and last release, cloud based):

```
SQL> select * from RDBMS_BRANDS;

RDBMS_NAME           DB_ENGINES_RANK DB_ENGINES_SCORE INITIAL_RELEASE CURRENT_RELEASE                C
-------------------- --------------- ---------------- --------------- ------------------------------ -
ORACLE                             1          1345.44            1979 20c, February 2020             N
MySQL                              2          1282.64            1995 8.0.20, 2020                   N
SQL Server                         3          1078.31            1989 SQL Server 2019, November 2019 N
PostgreSQL                         4           514.81            1989 12.2, February 2020            N
IBM Db2                            5           162.64            1983 12.1, October 2016             N
MariaDB                            8            90.09            2009 10.4.12, January 2020          N
Teradata                          10            73.89            1984 Vantage 1.0 MU2, January 2019  N
SAP HANA                          13            50.54            2010 2.0 SPS04, April 2019          N
Google BigQuery                   15            27.59            2010                                Y
Amazon Redshift                   18            20.27            2012                                Y
CockroachDB                       42             3.73            2015 19.2.6, April 2020             N
Snowflake                         48             2.36            2014                                Y

12 rows selected.
```

Ever.

# Automatic Zone Maps in Oracle Database 21c

## What is a Zone Map? Example:

- The RDBMS_BRANDS table has 6 data blocks with 2 rows per block:

| BLOCK | RDBMS_NAME | DB_ENGINES_RANK | DB_ENGINES_SCORE | INITIAL_RELEASE | CURRENT_RELEASE | CLOUD_ONLY |
|---|---|---|---|---|---|---|
| 1 | ORACLE | 1 | 1345,44 | 1979 | 20c, February 2020 | N |
| 1 | MySQL | 2 | 1282,64 | 1995 | 8.0.20, 2020 | N |
| 2 | SQL Server | 3 | 1078,31 | 1989 | SQL Server 2019, November 2019 | N |
| 2 | PostgreSQL | 4 | 514,81 | 1989 | 12.2, February 2020 | N |
| 3 | IBM Db2 | 5 | 162,64 | 1983 | 12.1, October 2016 | N |
| 3 | MariaDB | 8 | 90,09 | 2009 | 10.4.12, January 2020 | N |
| 4 | Teradata | 10 | 73,89 | 1984 | Vantage 1.0 MU2, January 2019 | N |
| 4 | SAP HANA | 13 | 50,54 | 2010 | 2.0 SPS04, April 2019 | N |
| 5 | Google BigQuery | 15 | 27,59 | 2010 | | Y |
| 5 | Amazon Redshift | 18 | 20,27 | 2012 | | Y |
| 6 | CockroachDB | 42 | 3,73 | 2015 | 19.2.6, April 2020 | N |
| 6 | Snowflake | 48 | 2,36 | 2014 | | Y |

Ever.

# Automatic Zone Maps in Oracle Database 21c

## What is a Zone Map? Example:

- Let us now create the zonemap on the RDBMS_BRANDS table (on 3 columns only):

```
SQL> CREATE MATERIALIZED ZONEMAP rdbms_zmap ON rdbms_brands (db_engines_rank,
db_engines_score, initial_release);
```

- We have now 3 zones and each zone contains two blocks and stores the minimum and maximum of db_engines_rank, db_engines_score and initial_release:

| BLOCK RANGE | MIN RANK | MAX RANK | MIN SCORE | MAX SCORE | MIN RELEASE | MAX RELEASE |
|---|---|---|---|---|---|---|
| 1-2 | 1 | 4 | 514,81 | 1345,44 | 1979 | 1989 |
| 3-4 | 5 | 13 | 50,54 | 162,64 | 1983 | 2010 |
| 5-6 | 15 | 48 | 2,36 | 27,59 | 2010 | 2015 |

# Automatic Zone Maps in Oracle Database 21c

## What is a Zone Map? Example:

- Let us run a query returning all RDBMS brands with ranking score more than 1000:

```
select * from RDBMS_BRANDS where DB_ENGINES_SCORE > 1000;
```

Script Output | Explain Plan | Query Result

SQL | All Rows Fetched: 3 in 0,082 seconds

| | RDBMS_NAME | DB_ENGINES_RANK | DB_ENGINES_SCORE | INITIAL_RELEASE | CURRENT_RELEASE | CLOUD_ONLY |
|---|---|---|---|---|---|---|
| 1 | ORACLE | 1 | 1345,44 | 1979 | 20c, February 2020 | N |
| 2 | MySQL | 2 | 1282,64 | 1995 | 8.0.20, 2020 | N |
| 3 | SQL Server | 3 | 1078,31 | 1989 | SQL Server 2019, November 2019 | N |

Ever.

# Automatic Zone Maps in Oracle Database 21c

## What is a Zone Map? Example:

- Looking at the execution plan below we see that Oracle is scanning only Zone 1 as the maximum score in all other zone is smaller than 1000:

```
SQL> SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------
Plan hash value: 1824380443

-----------------------------------------------------------------------------------------------
| Id  | Operation                          | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |              |     3 |   120 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL WITH ZONEMAP| RDBMS_BRANDS |     3 |   120 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(SYS_ZMAP_FILTER('/* ZM_PRUNING */ SELECT zm."ZONE_ID$", CASE WHEN
              BITAND(zm."ZONE_STATE$",1)=1 THEN 1 ELSE CASE WHEN (zm."MAX_2_DB_ENGINES_SCORE" <= :1)
              THEN 3 ELSE 2 END END FROM "SYS"."RDBMS_ZMAP" zm WHERE zm."ZONE_LEVEL$"=0 ORDER BY
              zm."ZONE_ID$"',SYS_OP_ZONE_ID(ROWID),1000)<3 AND "DB_ENGINES_SCORE">1000)

16 rows selected.
```

Ever.

# Automatic Zone Maps in Oracle Database 21c

## How to enable automatic Zone Maps?

- In 21c, you can use the new package DBMS_AUTO_ZONEMAP to enable Automatic Zone Maps in the database. Automatic zone map creation is turned off by default.

```
Last login: Tue Aug 17 13:27:26 UTC 2021
[oracle@aeg ~]$ sqlplus "/ as sysdba"

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Aug 17 13:27:46 2021
Version 21.1.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.

Connected to:
Oracle Database 21c EE High Perf Release 21.0.0.0.0 - Production
Version 21.1.0.0.0

SQL> exec DBMS_AUTO_ZONEMAP.CONFIGURE('AUTO_ZONEMAP_MODE','ON');

PL/SQL procedure successfully completed.
```

Ever.

# Automatic Zone Maps in Oracle Database 21c

## How to enable automatic Zone Maps?

- In 21c, AUTO_ZONEMAP_MODE accepts these four different values:

1. ON: Turns on auto zone map feature completely. Both for foreground and background zone map creation and maintenance

2. OFF: Turns off auto zone map feature completely. Both for foreground and background zone map creation and maintenance

3. FOREGROUND: Turns on only for foreground zone map creation and maintenance (user process accessing the table for direct path and data movement operations)

4. BACKGROUND: Turns on only for background zone map creation and maintenance (auto task)

- Automatic Zone Maps are available for now only on Exadata and requires the Oracle Partitioning option.

Ever.

“This is the most important thing we have done in a long, long time. The automation does everything. We can guarantee an availability time of 99.995%, less than 30 minutes a year of planned or unplanned downtime.”

**Larry Ellison**
Oracle Executive Chairman and CTO

ORACLE