

ORACLE



Oracle Database In-Memory és Exadata

Gyorsított adatbázisok, nagyobb teljesítmény, konszolidációs lehetőség

Fekete Zoltán

Principal Solution Engineer

2022. május 18.

Oracle Database World, 2022. visszánézése: előadások, gyakorlatok

<https://www.oracle.com/database/database-world/2022>



How to Build a Cloud Data Mart in 20 Minutes (19:04)

Patrick Wheeler, Vice President, Product Management



Deep Dive: Data Loading (20:32)

William (Bud) Endress, Director, Product Management



Tips and Tricks: Data Warehousing with Data from Oracle E-Business Suite and SaaS (19:53)

Jayant Mahto, Senior Principal Product Manager



Predict the Future (or Don't) with Machine Learning in Applications (21:49)



Graph Analytics: From Zero to Hero (17:38)

Melliyal (Melli) Annamalai, Distinguished Product



Store and Analyze JSON Data Using the Oracle API for MongoDB and SQL/JSON (22:40)

[Keynote](#)

[AppDev Made Simple](#)

[Mission Critical Made Simple](#)

[Data Analytics Made Simple](#)

[Hands-On Labs](#)



Oracle LiveLabs: közel 500 hands-on workshop

<https://bit.ly/golivelabs> a Performance, Security-től a GoldenGate-ig és Verrazzanoig...

EM - Database Performance Management: Find, Fix, Validate

Manage on-premises and Oracle Cloud Databases (Virtual Machine/Bare Metal/Exadata Cloud) using EM (..)

Launch

9024 Views

🕒 55 mins

Boost Analytics Performance with Oracle Database In-Memory

Accelerate column-oriented data access by analytics operations using Oracle's In-Memory column (..)

Launch

7624 Views

🕒 1 hr, 30 mins

Provision HPC Cluster from Oracle Marketplace Image

Provision your High Performance Compute platform in Oracle Cloud Infrastructure.

Launch

3531 Views

🕒 4 hrs

Scaling and Performance in the Autonomous Database

Apply scaling and performance to your Autonomous Database instance.

Launch

3496 Views

🕒 30 mins

Real Application Testing : SQL Performance Analyzer-Database Replay

Real Application Testing : SQL Performance Analyzer and Database Replay

Launch

2733 Views

🕒 55 mins

Performance through Database Design

Increase the performance of your database by focusing on good database design.

Launch

2155 Views

🕒 1 hr, 10 mins



Database 21c - Performance & High Availability

Explore the performance and HA new features in Database 21c.

Launch

792 Views

🕒 1 hr

New

Monitor Traces for WebLogic on Kubernetes utilizing Oracle Application Performance Monitoring

Enable OpenTracing for a WebLogic app on Kubernetes using Oracle Application Performance Monitoring

Launch

747 Views

🕒 1 hr, 30 mins

Use OpenTracing for Microservices with Helidon Utilizing Oracle Application Performance Monitoring

Monitor an application built on microservices that is implemented on Helidon using Oracle (..)

Launch

706 Views

🕒 1 hr

Return to top of page

<https://bit.ly/golivelabs>



Enhancements to Database Features & Licensing App

<https://apex.oracle.com/database-features>

The screenshot displays the Oracle Database Features and Licensing App interface. At the top, a dark header contains the 'DATABASE' logo and navigation icons. Below this, a section titled 'Features and Licensing' includes a sub-header and a link to a 'new learning guide'. The main content area is divided into a left sidebar and a right main panel. The sidebar has tabs for 'Features' and 'Licensing', a search bar with the text 'blockchain table', a 'Focus Area' dropdown set to 'All Focus Areas', and a 'Version' section with checkboxes for 11.2, 12.1, 12.2, 18c, 19c (checked), and 21c. A 'New features only' checkbox is also checked. The main panel shows the 'Feature Details' for 'Oracle Blockchain Table', which is categorized under 'Database Overall + Core Database'. It includes a description of blockchain tables as append-only tables with insert-only operations, a 'Business Benefit' section explaining their use in blockchain applications, and a 'Release Availability' section with a visual indicator showing availability from 19.10 onwards. Other details include 'Available On: ALL OFFERINGS', 'Initial Release: 21c', and 'Backported To: 19.10 More Information'.

Oracle Blockchain Table
Database Overall + Core Database

Blockchain tables are append-only tables in which only insert operations are allowed. Deleting rows is either prohibited or restricted based on time. Rows in a blockchain table are made tamper-resistant by special sequencing and chaining algorithms. Users can verify that rows have not been tampered. A hash value that is part of the row metadata is used to chain and validate rows.

Business Benefit: Blockchain tables can be used to implement blockchain applications where the participants trust the Oracle Database provider, but want means to verify that their data hasn't been tampered with. The participants are different database users who trust the Oracle Database provider to maintain a verifiable, tamper-resistant blockchain of transactions. All participants must have privileges to insert data into the blockchain table. The contents of the blockchain table are defined and managed by the application, with a few added metadata fields maintained by Oracle Database. By leveraging a trusted provider with verifiable crypto-secure data management practices, such applications can avoid the distributed consensus requirements. This provides most of the protection of the distributed peer-to-peer blockchains, but with much higher throughput and lower transaction latency compared to peer-to-peer blockchains using distributed consensus.

Release Availability: 11.2, 12.1, 12.2, 18c, 19.10, 21c

Available On: ALL OFFERINGS

Initial Release: 21c

Backported To: 19.10 [More Information](#)



Oracle Dev Gym - <https://devgym.oracle.com>

DEV GYM Search the Dev Gym... ? Sign In

Expertise Through Exercise!

Join the Oracle Dev Gym (FREE!) and build your Oracle technology muscles by taking workouts, quizzes and even entire classes on SQL, PL/SQL, database design, logic and more.

[Learn More](#) [Get Started →](#)

Classes

Classes at the Dev Gym are 100% free and open to everyone. Start whenever you are ready, and work on them at your own pace. Reinforce your new knowledge with quizzes at the end of each module.

Quizzes

We've got over 2,500 quizzes available to help you test and reinforce your knowledge of key Oracle technologies and more. Most quizzes are multiple choice and packed full of code, so you get to improve your code reading skills as well! And did we mention it's all free?

Workouts

Workouts are 100% free and perfect when you don't have time for a whole class, but want to get up to speed (or get a refresher) on specific topics. Each workout consists of a video or article, followed by a set of quizzes. Choose from our workouts or create your own in a few seconds.

Oracle Database Tools State of the Union 2022 - február 9. 17h

DBA-knak és fejlesztőknek tudásbővítő összefoglaló előadás

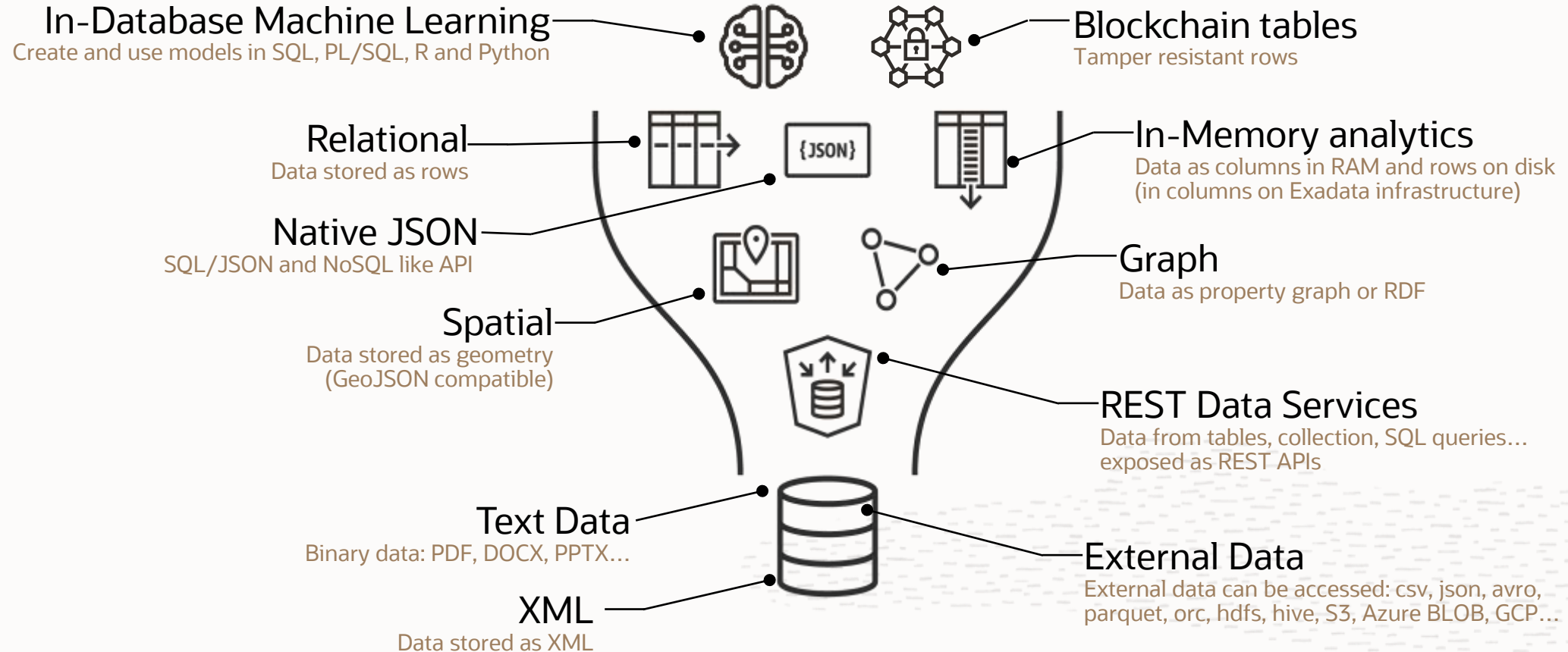
<https://asktom.oracle.com/pls/apex/asktom.search?oh=16721>



**SQL Developer Web, SQLcl, Oracle REST Data Services,
our new OCI Database Tools Service, etc.**

Adatvezérelt szervezetek, alkalmazások, konvergens Database: **multi-tenant, -model, -workload**

Közös rendelkezésre állás, teljesítmény, biztonság, párhuzamosság, üzemeltetés...



Multi-model and **multi-workload**

Organizations have many business questions

Yet critical reporting and analytics don't perform at business speeds

Which products give us our highest margins?

Who are the top 10 sales reps in the north west region this month?

If I get a 20% discount on widget A, how much will our margins improve?



Introducing Database In-Memory

What's your favorite data format?

Row Format

*Fast for OLTP!
Slower for Analytics*



Column Format

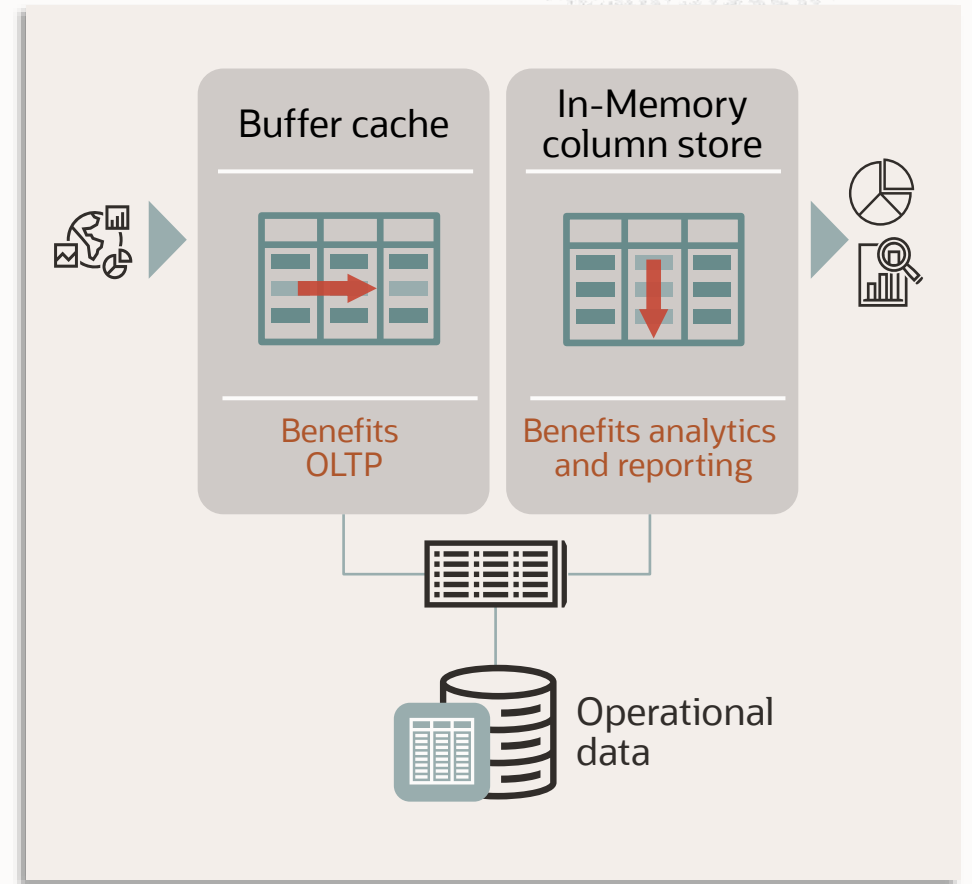
*Fast for Analytics!
REALLY slow for OLTP!*



(+)

Dual Format

*Best for both
Fast Analytics
and fast OLTP
(No need for
Analytic indexes)*



Oracle In-Memory: Simple to Implement

1. Configure Memory Capacity

```
inmemory_size = XXX GB
```

2. Configure tables or partitions to be in memory

```
alter table | partition ... inmemory;
```

3. Later drop analytic indexes to speed up OLTP

Where Is Database In-Memory Available?

Database In-Memory is an **option** for Oracle Database Enterprise Edition

Database In-Memory was included in the first patchset (12.1.0.2) for 12.1 and all subsequent Oracle Database releases

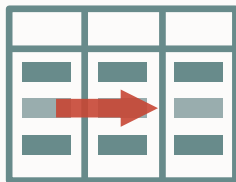
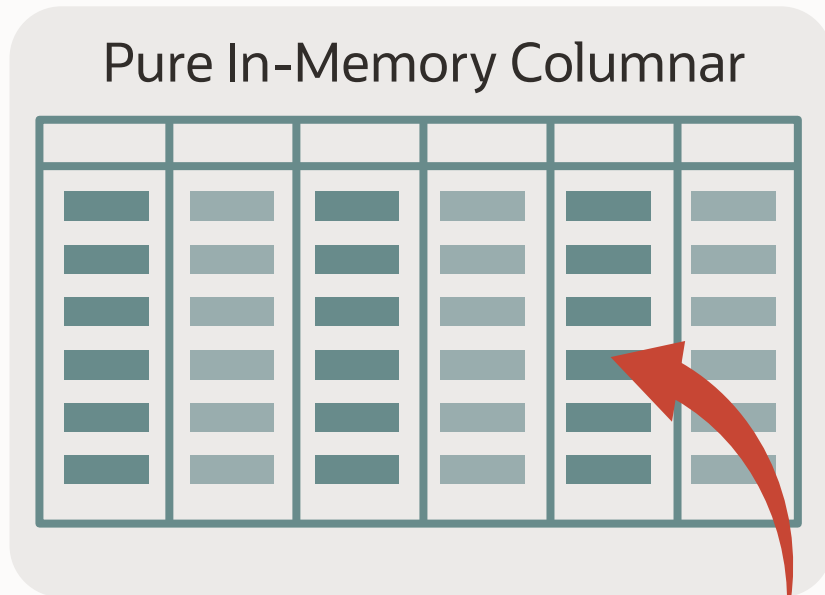
Available:

- Database Cloud Service – Virtual Machines: **Extreme Performance**
- Database Cloud Service – Bare Metal: **Extreme Performance**
- Exadata Cloud Service
- Exadata Cloud at Customer
- Autonomous Data Warehouse (Flash only)
- On-premises
- Oracle Database XE



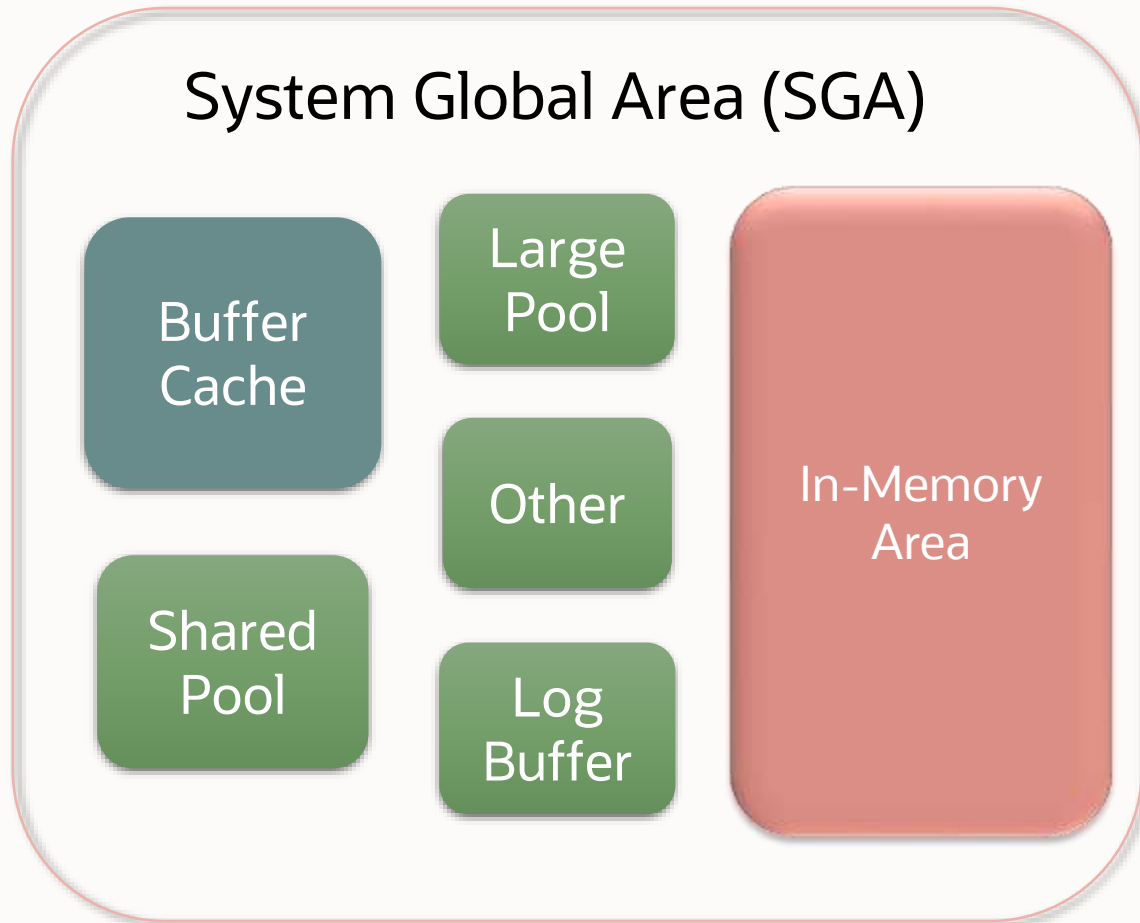
Note: Database In-Memory is **not** enabled by default

Oracle In-Memory Columnar Technology



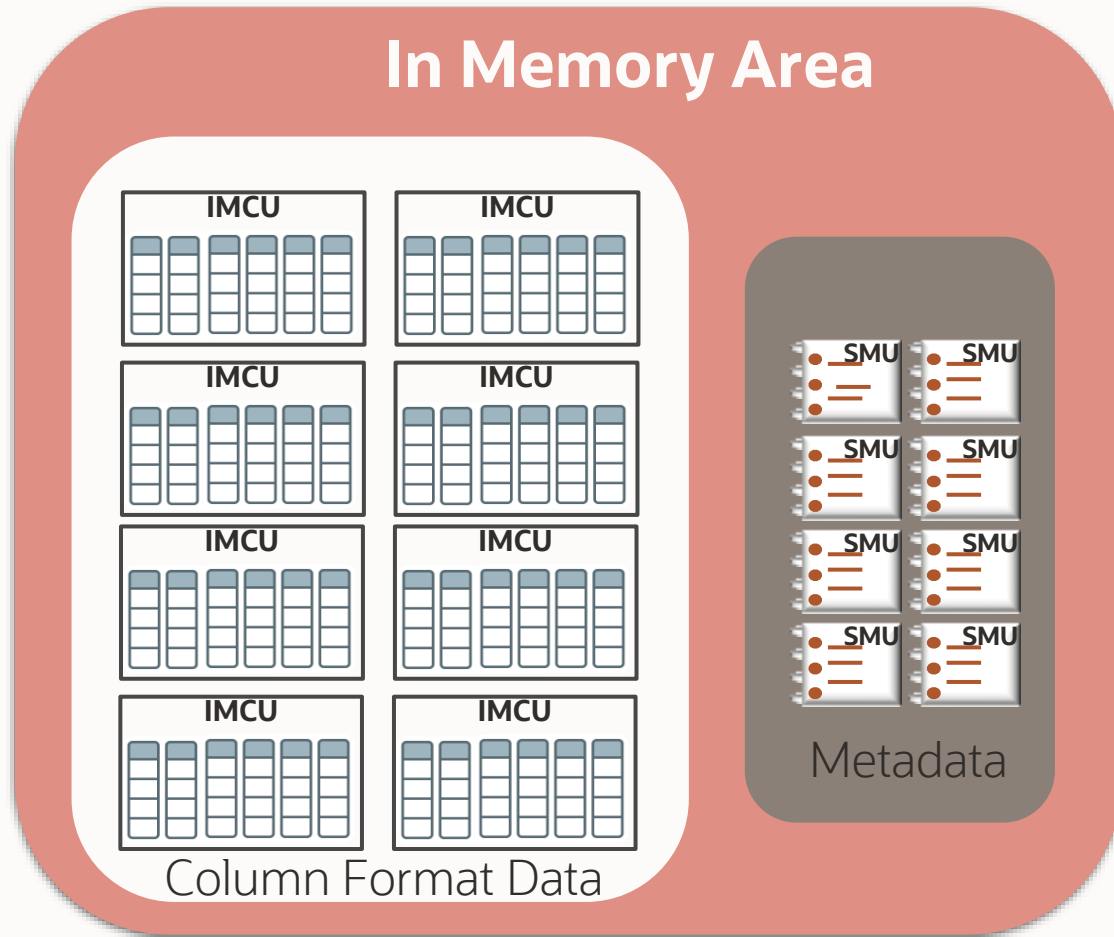
- Pure in-memory columnar format
 - Not persistent, so no undo or redo is generated
- **Table, partition, subpartition or materialized view**
- **2x to 20x compression typical**
- Available on all hardware platforms
- Does not require the whole database to be in-memory
- Can be enabled for just active data (table, partition, sub-partition, materialized view)

In-Memory Area: **Static Area within SGA**



- Contains data in the new In-Memory Columnar Format
- Controlled by **INMEMORY_SIZE** parameter
 - Minimum size of 100MB
- Can be re-sized larger while database is running (12.2)
- **SGA_TARGET** must be large enough to accommodate In-Memory area

Composition of In-Memory Area



- Contains two subpools:
 - IMCU pool: Stores In Memory Compression Units (IMCUs)
 - SMU pool: Stores Snapshot Metadata Units (SMUs)
- IMCUs contain column formatted data
- SMUs contain metadata and transactional information

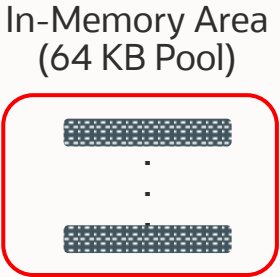
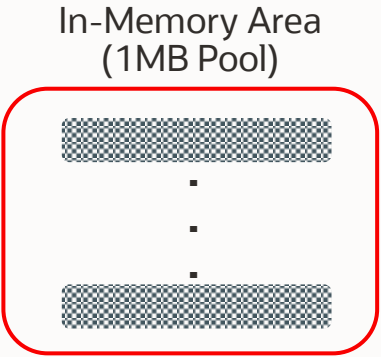
Composition of In-Memory Object

Employee Table		



An IMCU/SMU is composed of one or more 1MB/64KB memory segments

An object in memory is composed of one or more IMCUs/SMUs



View In-Memory Area Usage

```
SQL> SELECT * FROM v$inmemory_area;
```

V\$INMEMORY_AREA
Current size of pools in the In-Memory area

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS
1MB POOL	5,179,965,440	3,241,148,416	DONE
64KB POOL	570,425,344	9,568,256	DONE

```
FROM v$im_segments;
```

V\$IM_SEGMENTS
List of segments currently populated in the In-Memory column store

OWNER	NAME	STATUS	In-Memory Size	Bytes Not Populated
SSB	LINEORDER	COMPLETED	3,206,086,656	0
SSB	DATE_DIM	COMPLETED	1,179,648	0
SSB	SUPPLIER	COMPLETED	2,228,224	0
SSB	PART	COMPLETED	18,022,400	0
SSB	CUSTOMER	COMPLETED	23,199,744	0



Population

- Order in which objects are populated controlled by PRIORITY subclause:
 - **ALTER TABLE sales**
INMEMORY PRIORITY HIGH;
- Levels:
 - CRITICAL > HIGH > MEDIUM > LOW
 - Controls order (not speed) of populate
- Default PRIORITY is NONE
 - Populate only on first access

```
oracle@srv80101:~/In_Memory_Beta/lesson4
top - 15:32:09 up 7 days, 23:45, 7 users, load average: 14.72, 4.14, 1.55
Tasks: 622 total, 36 running, 586 sleeping, 0 stopped, 0 zombie
Cpu(s): 96.7%us, 1.9%sy, 0.0%ni, 0.0%id, 1.2%wa, 0.0%hi, 0.1%si, 0.0%st
Mem: 148834648k total, 146686500k used, 2148148k free, 187748k buffers
Swap: 2096440k total, 92k used, 2096348k free, 131648316k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 24673 oracle    20   0 120g  1.7g  1.6g  R  79.0   1.2   6:13.27 ora_w014_orcl
 24569 oracle    20   0 120g  2.5g  2.4g  R  76.7   1.7   9:15.96 ora_w003_orcl
 24663 oracle    20   0 120g  1.7g  1.7g  R  74.4   1.2   6:32.96 ora_w00z_orcl
 24627 oracle    20   0 120g  2.0g  1.9g  R  73.1   1.4   7:57.44 ora_w00o_orcl
 24625 oracle    20   0 120g  2.2g  2.1g  R  72.4   1.5   8:42.75 ora_w00n_orcl
 24667 oracle    20   0 120g  2.0g  1.9g  R  72.1   1.4   7:31.26 ora_w011_orcl
 24571 oracle    20   0 120g  2.5g  2.3g  R  71.8   1.8   9:32.78 ora_w004_orcl
 24657 oracle    20   0 120g  1.8g  1.7g  R  71.1   1.3   6:41.06 ora_w00w_orcl
 24669 oracle    20   0 120g  2.2g  2.1g  R  70.8   1.6   8:56.33 ora_w012_orcl
 24683 oracle    20   0 120g  1.7g  1.7g  R  70.5   1.2   6:46.75 ora_w018_orcl
 24621 oracle    20   0 120g  2.0g  1.9g  R  70.1   1.4   8:12.00 ora_w001_orcl
 24687 oracle    20   0 120g  1.9g  1.8g  R  70.1   1.4   7:58.64 ora_w019_orcl
 24611 oracle    20   0 120g  2.3g  2.0g  R  69.8   1.6   8:13.25 ora_w00g_orcl
 24619 oracle    20   0 120g  1.9g  1.8g  R  68.5   1.3   6:36.25 ora_w00k_orcl
 24671 oracle    20   0 120g  1.9g  1.8g  R  68.2   1.3   7:13.82 ora_w013_orcl
 24675 oracle    20   0 120g  1.7g  1.6g  R  67.5   1.2   6:42.16 ora_w015_orcl
 24659 oracle    20   0 120g  1.9g  1.8g  R  67.2   1.3   6:53.53 ora_w00x_orcl
 24631 oracle    20   0 120g  2.3g  2.3g  R  66.9   1.6   9:28.46 ora_w00p_orcl
 24654 oracle    20   0 120g  1.8g  1.7g  R  66.9   1.3   6:57.75 ora_w00v_orcl
```

- Population completed by background processes
ora_w00x_orcl
- Number of processes controlled by parameter:

INMEMORY_MAX_POPULATE_SERVERS



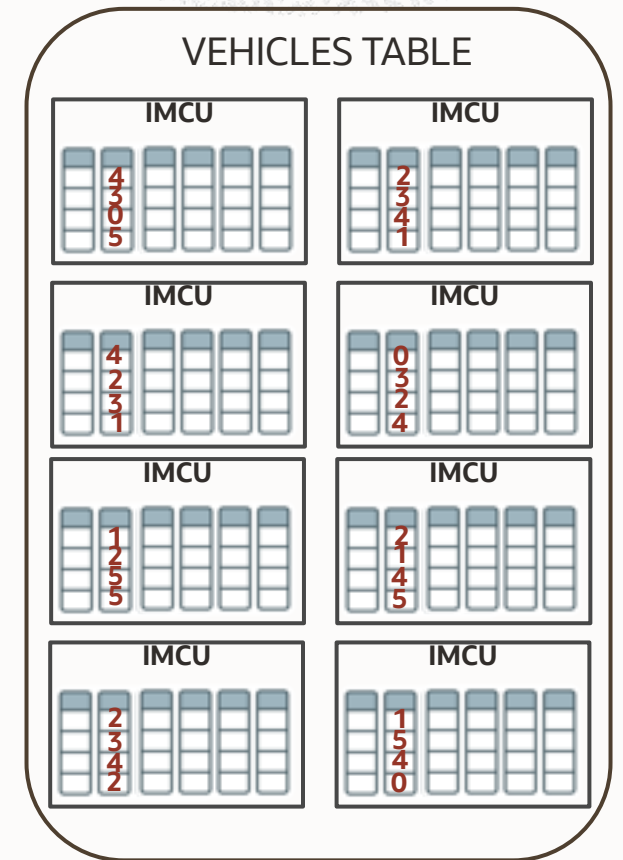
Database In-Memory Technology

Compression

- Multiple levels of compression
 - FOR DML
 - FOR QUERY LOW/HIGH
 - FOR CAPACITY LOW/HIGH
- Query Low and High use dictionary and run length encoding – evaluated directly against compressed data
- Capacity Low and High add additional “zip-like” compression

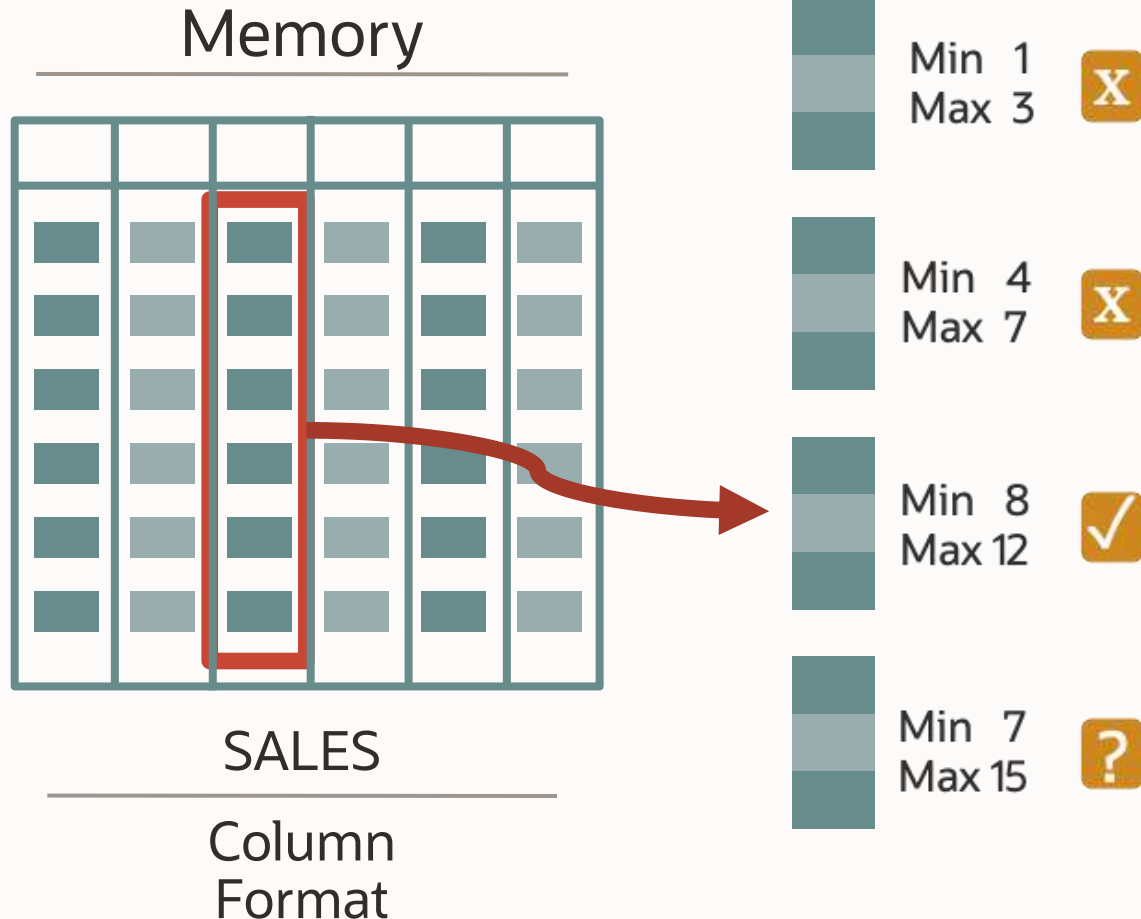
Common Dictionary

NAME	ID
AUDI	0
BMW	1
CADILLAC	2
PORSCHE	3
TESLA	4
VW	5



In-Memory Storage Indexes

Only look at the data you need!

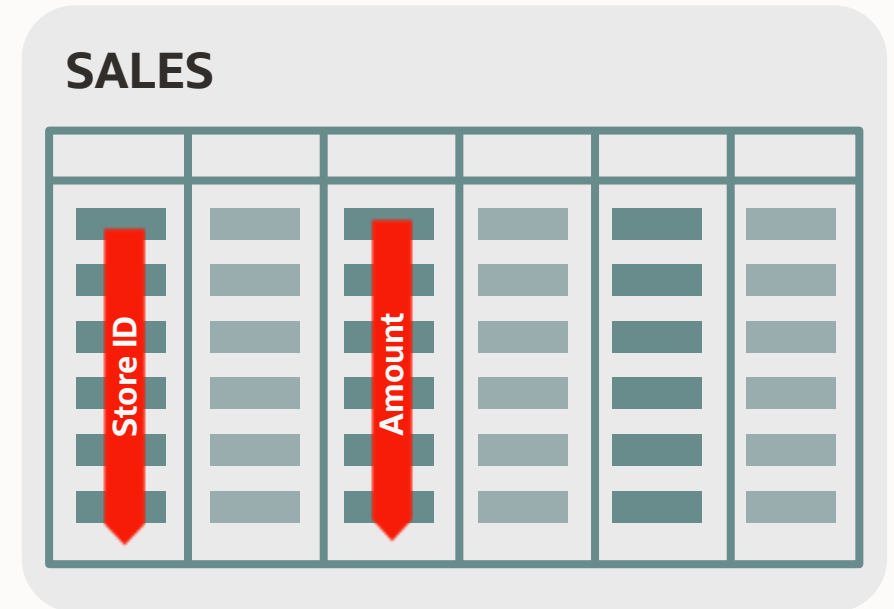


- Example: Find all sales from stores with a store_id of 8
 - Each column is made up of multiple column units
 - Min / max value is recorded for each column unit in a storage index
 - Storage index provides partition pruning like performance for **ALL** queries

In-Memory Scans

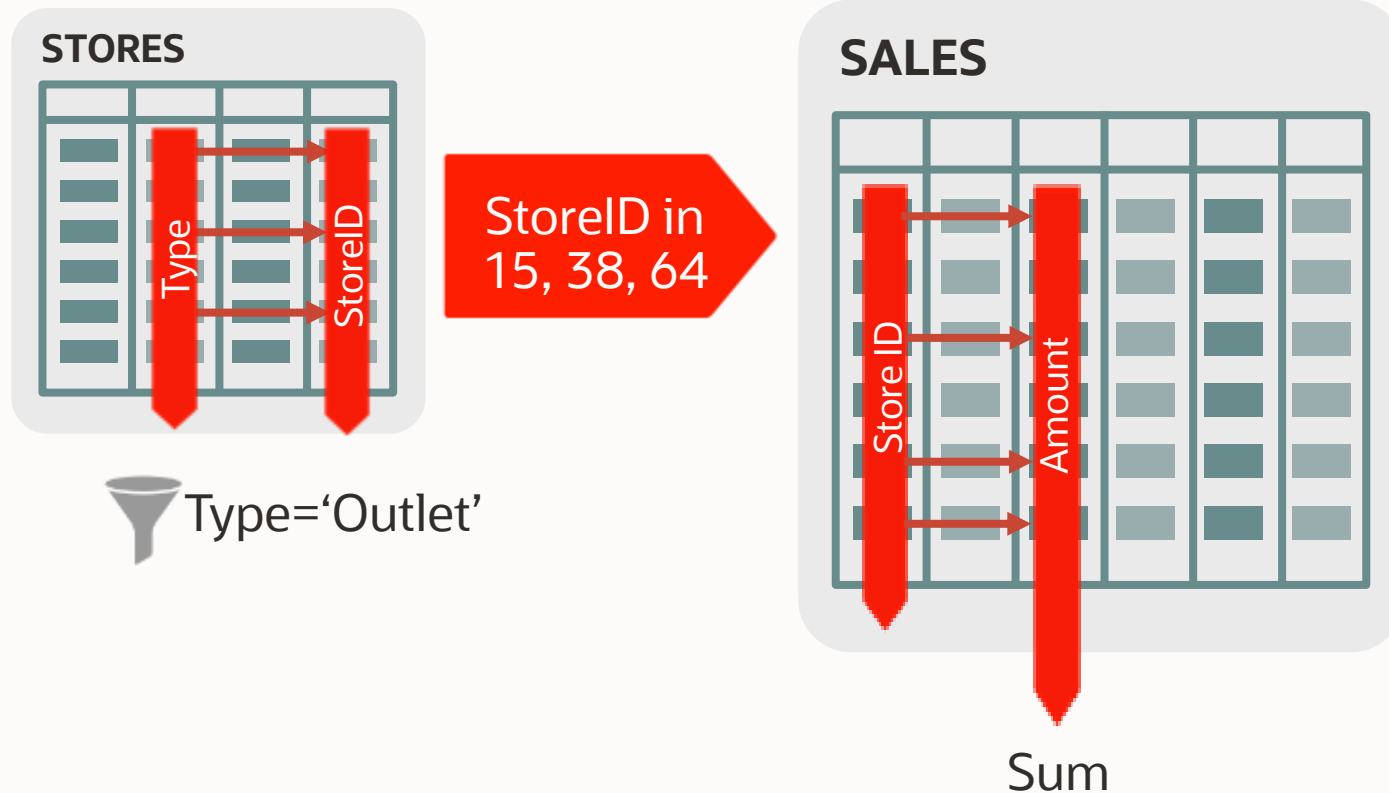
Many types of filter predicates can be more efficiently evaluated **during** the In-Memory scan rather than after

- Only scan the columns needed for the query
- Prune IMCUs using storage indexes and dictionary-based compression metadata
- Evaluate predicates directly against compressed columnar data
- Use SIMD to evaluate predicates on multiple column values concurrently
- Aggregate data during the scan



Joining and Combining Data Also Dramatically Faster

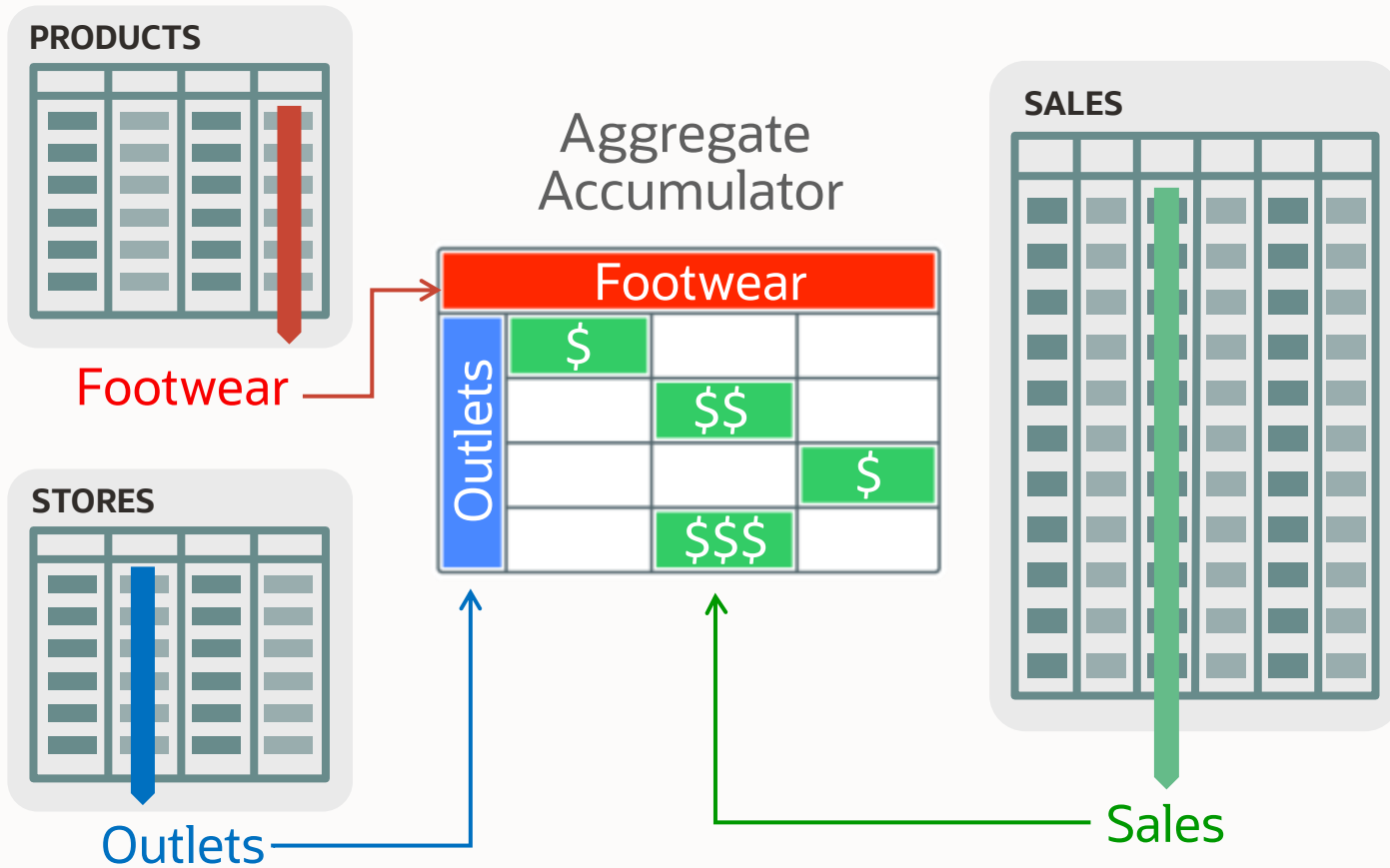
Example: Find total sales in outlet stores



- Converts joins of data between two or more tables into fast column scans using Bloom filters
- Joins tables **10x** faster

In-Memory Aggregation

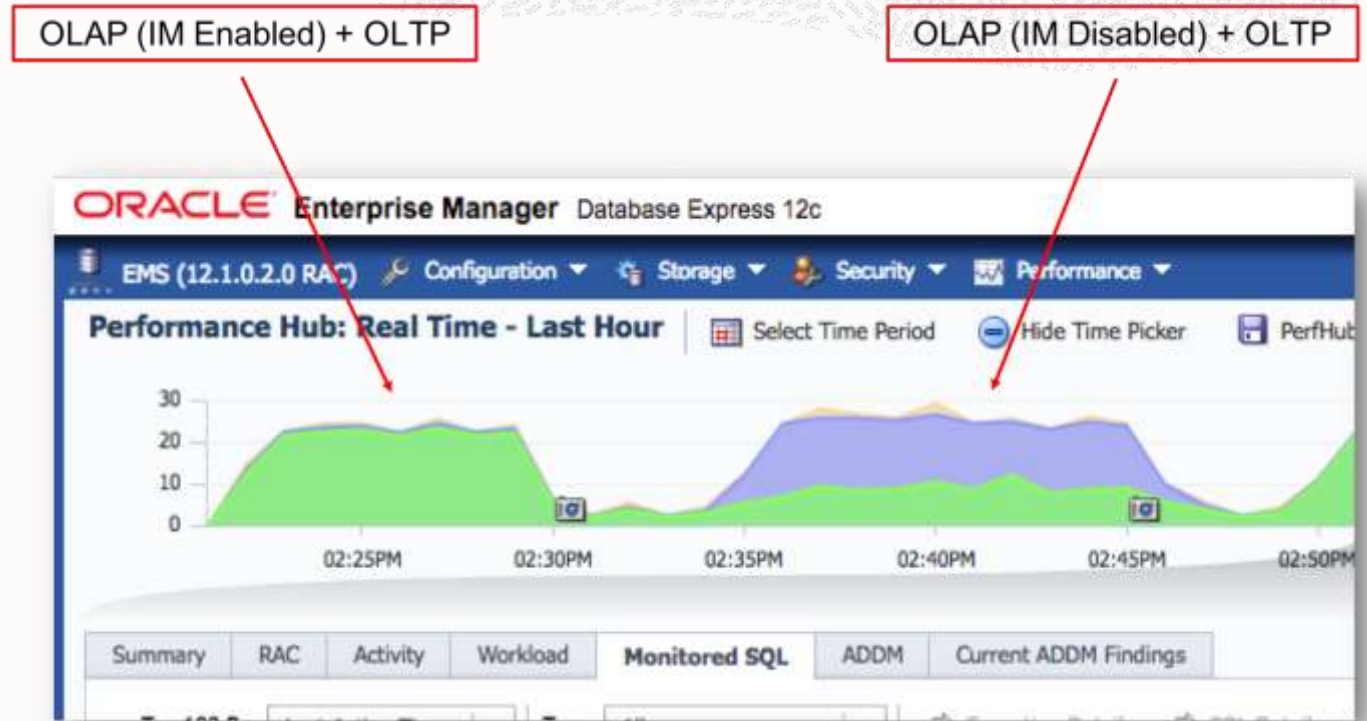
Example: Report sales of footwear in outlet stores



- Execution plan shows **Vector Group By**
- Dynamically creates in-memory report outline (aggregate accumulator)
- Aggregation performed in-memory during fast fact scan
- Key vectors are used instead of Bloom filters
- Key vectors use dense grouping keys to map all key combinations

Real World Application Demo – [DBIM YouTube Channel](#)

- Database In-Memory enables real time analytics
- OLTP transaction workload with Database In-Memory
- Workload is not allowed to exceed the capacity of the machine
 - OLTP workload (simulates 6400 users each running a transaction every 10 seconds)
 - OLAP workload (20 connections running a stream of 100 randomly selected reports with no think time)



DBIM enabled – Maximum throughput, no OLTP degradation

DBIM disabled – I/O required, fewer reports, OLTP impacted

Compare Column-store to Row-store

```
SQL> -- In-Memory Column Store query
SQL>
SQL> select max(lo_ordtotalprice) most_expensive_order From LINEORDER;

MOST_EXPENSIVE_ORDER
-----
57346348
Elapsed: 00:00:00.01

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT | | 1 | | 5401 (100) | |
| 1 | SORT AGGREGATE | | 1 | 6 | | |
| 2 | TABLE ACCESS INMEMORY FULL | LINEORDER | 59M | 343M | 5401 (16) | 00:00:01 |

SQL> -- Buffer Cache query with the column store disabled via NO_INMEMORY hint
SQL>
SQL> select /*+ NO_INMEMORY */ max(lo_ordtotalprice) most_expensive_order From LINEORDER;

MOST_EXPENSIVE_ORDER
-----
57346348
Elapsed: 00:00:08.38

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT | | 1 | | 123K(100) | |
| 1 | SORT AGGREGATE | | 1 | 6 | | |
| 2 | TABLE ACCESS FULL | LINEORDER | 59M | 343M | 123K (1) | 00:00:05 |
```

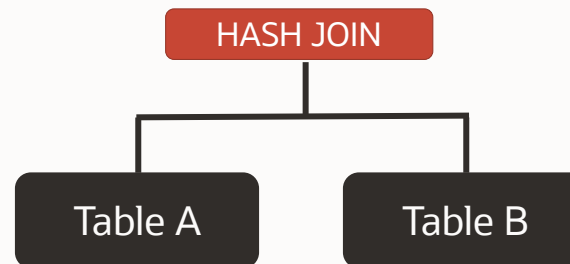
Which Queries Benefit From Database In-Memory?

For a non-trivial amount of rows and execution time, when a significant amount of time ...

is spent accessing data



is spent joining data



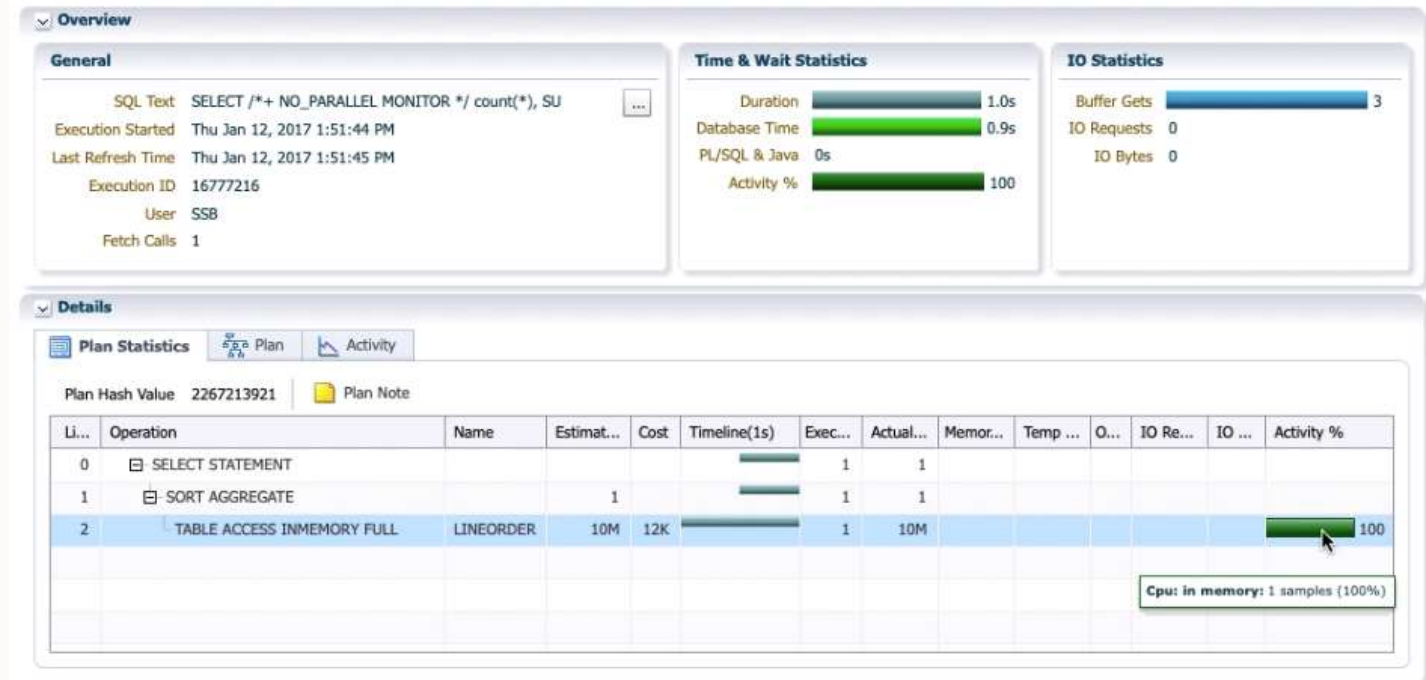
is spent aggregating data



Use Time Based Analysis Techniques To Evaluate Benefit

SQL Monitor Active Reports

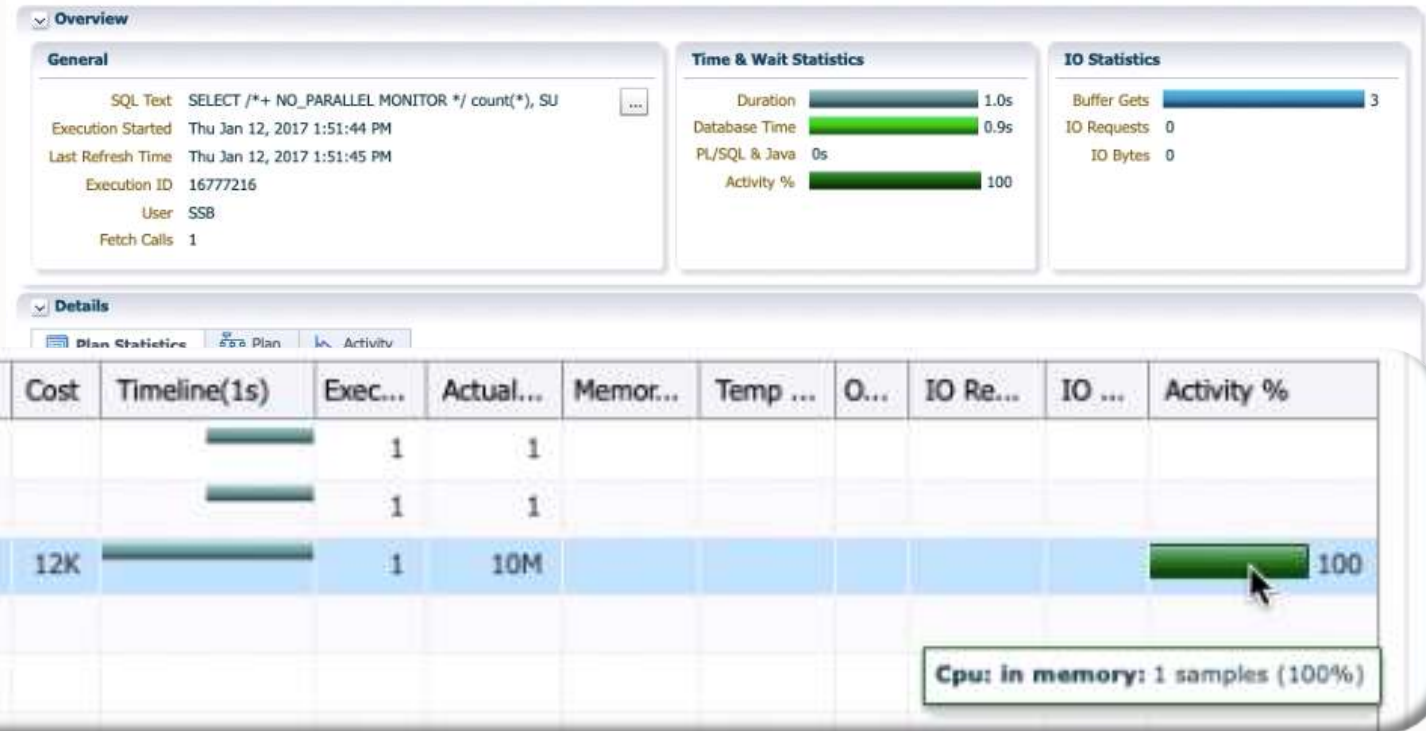
- Shows how SQL was executed and where time was spent
- See blogs.oracle.com/In-Memory for a technical brief on creating SQL Monitor active reports



Use Time Based Analysis Techniques To Evaluate Benefit

SQL Monitor Active Reports

- Shows how SQL was executed and where time was spent



Oracle In-Memory Advisor

Workload Database Usage

Total Database Time (Seconds)	Analytics Processing Time (Seconds)	Analytics Processing Percentage
2990	2640	88%

In-Memory Size	Percentage of Maximum SGA Size (100.0GB)	Estimated Analytics Processing Time Reduction (Seconds)	Estimated Analytics Processing Performance Improvement Factor
9.141GB	9%	2102	4.9X
8.684GB	9%	2101	4.9X
8.226GB	8%	2101	4.9X
7.769GB	8%	2100	4.9X

- In-Memory Advisor – free download available on OTN for 11.2.0.3+ DBs
- Analyzes existing DB workload via AWR & ASH repositories
- Provides list of objects that would benefit most from being populated into IM column store



Note: Database Tuning Pack license required

Oracle In-Memory Advisor



SQL Id	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Time Reduction (Seconds) With Unlimited Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 8.14TGB	Estimated Analytics Processing Performance Improvement Factor With 8.14TGB
fp8jwvthaknd	select cf.custid, sum(act.purchase_amt) sales from all_card_trans act, cust_fact cf ...	990	696	3.4X	696	3.4X
7akhj3xhq01w8	with gold_member_all_cust as (select custid, aff_cc_num from cust_fact w...	940	660	3.4X	660	3.4X
8p8ggu/p97699	with act as (select act.card_no, act.purchase_amt from all_card_trans act ,mcc m, zipcodes z...	710	450	2.7X	450	2.7X

- Multiple sections available
 - In-Memory Size
 - SQL Statements with Analytic Benefit
 - Top object recommendations
 - All object based on memory size
 - Recommendation Rationale
 - Implementation SQL

Object Type	Object	Compression Type	Estimated In-Memory Size	Analytics Processing Seconds	Estimated Reduced Analytics Processing Seconds	Estimated Analytics Processing Performance Improvement Factor	Benefit / Cost Ratio (Reduced Analytics Processing / In-Memory Size)
TABLE	TEST_UNCOMP.ZIPCODES	Memory compress for query low	1.063MB	50	33	3.0X	507741 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5598.SYS_SUBP5592	Memory compress for query low	1.063MB	1	0	3.0X	56330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5598.SYS_SUBP5593	Memory compress for query low	1.063MB	1	0	3.0X	56330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5620.SYS_SUBP5615	Memory compress for query low	1.063MB	1	0	3.0X	28577 : 1



Oracle Compression Advisor And In-Memory

```
DECLARE
v_blkcnt_cmp      BINARY_INTEGER;
v_blkcnt_uncmp    BINARY_INTEGER;
v_row_cmp         BINARY_INTEGER;
v_row_uncmp       BINARY_INTEGER;
v_cmp_ratio       NUMBER := -1;
v_comptype_str    VARCHAR2(60);
BEGIN
DBMS_COMPRESSION.GET_COMPRESSION_RATIO(
  scratchtbsname => 'TS_DATA',
  ownname        => 'SSB',
  objname        => 'LINEORDER',
  subobjname     => NULL,
  comptype       => DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW,
  blkcnt_cmp     => v_blkcnt_cmp,
  blkcnt_uncmp   => v_blkcnt_uncmp,
  row_cmp        => v_row_cmp,
  row_uncmp     => v_row_uncmp,
  cmp_ratio      => v_cmp_ratio,
  comptype_str   => v_comptype_str,
  subset_numrows => DBMS_COMPRESSION.COMP_RATIO_ALLROWS);
DBMS_OUTPUT.PUT_LINE('Compression Type: '||TO_CHAR(v_comptype_str));
DBMS_OUTPUT.PUT_LINE('Estimated Compression Ratio: '||TO_CHAR(v_cmp_ratio));
END;
/
```

- Easy way to determine memory requirements
- Use DBMS_COMPRESSION
- Applies MEMCOMPRESS to sample set of data from a table
- Returns estimated compression ratio

Exadata Is The Best Platform for Database In-Memory

Why In-Memory on Exadata: 4 Unique Features

Unique to Exadata

- In-Memory formats on Exadata Flash
- In-Memory Duplication
- In-Memory on Active Data Guard
- Automatic In-Memory

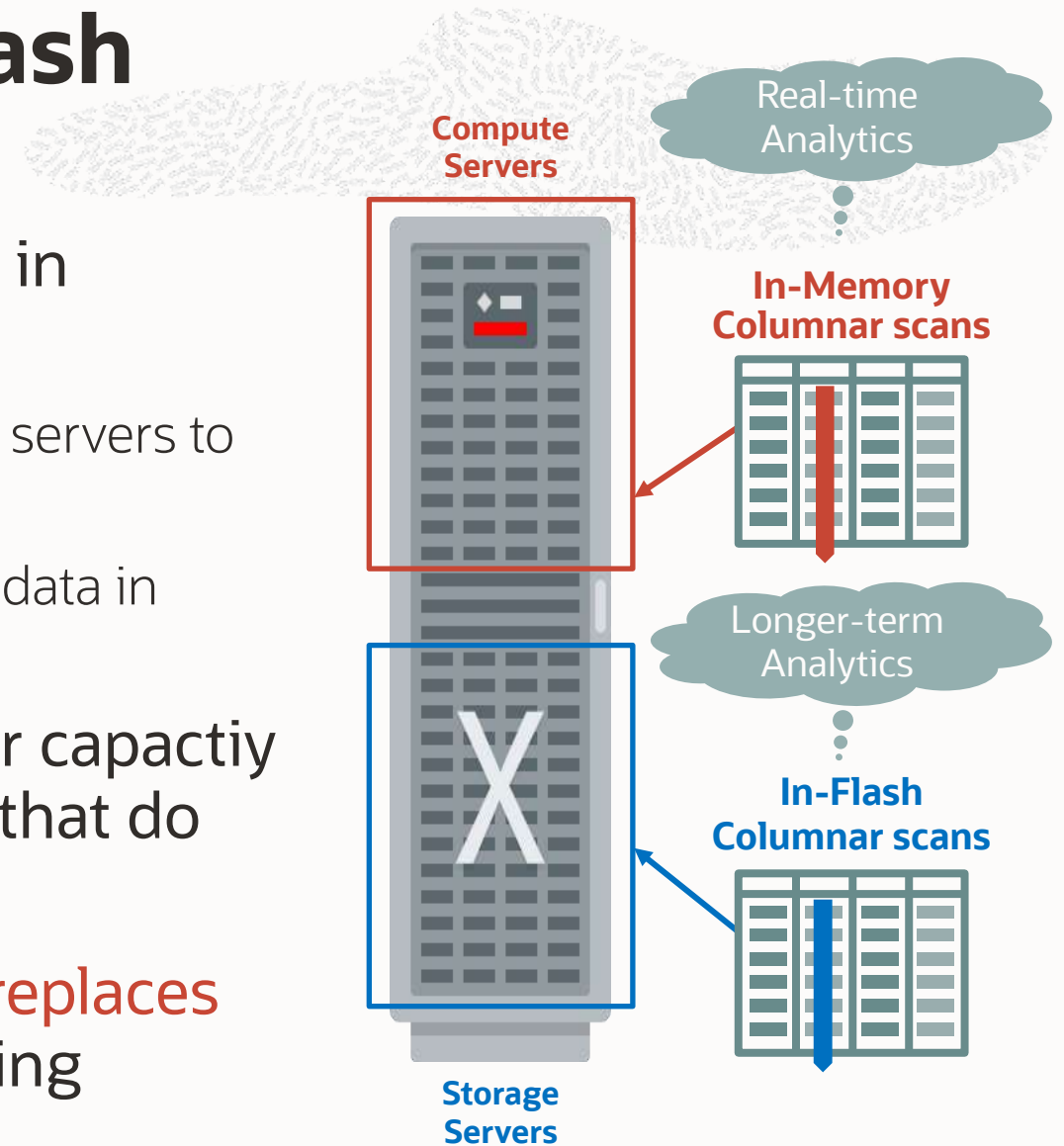
Available on All Flavors of Exadata

- On-Premises
- Exadata Cloud Service
- Exadata Cloud at Customer



In-Memory **Extended** To Flash

- In-Memory columnar format also available in *Exadata Smart Flash Cache*
 - Extends in-memory from DRAM in DB compute servers to Flash in storage servers
 - Enables the **SAME** in-memory optimizations on data in storage servers as on DB compute servers
- Massive increase in In-Memory columnar capacity (~500TB on full rack X8) for large tables that do not fit in DRAM
- Exadata smart query offload to storage **replaces** the need to offload workload to a reporting database



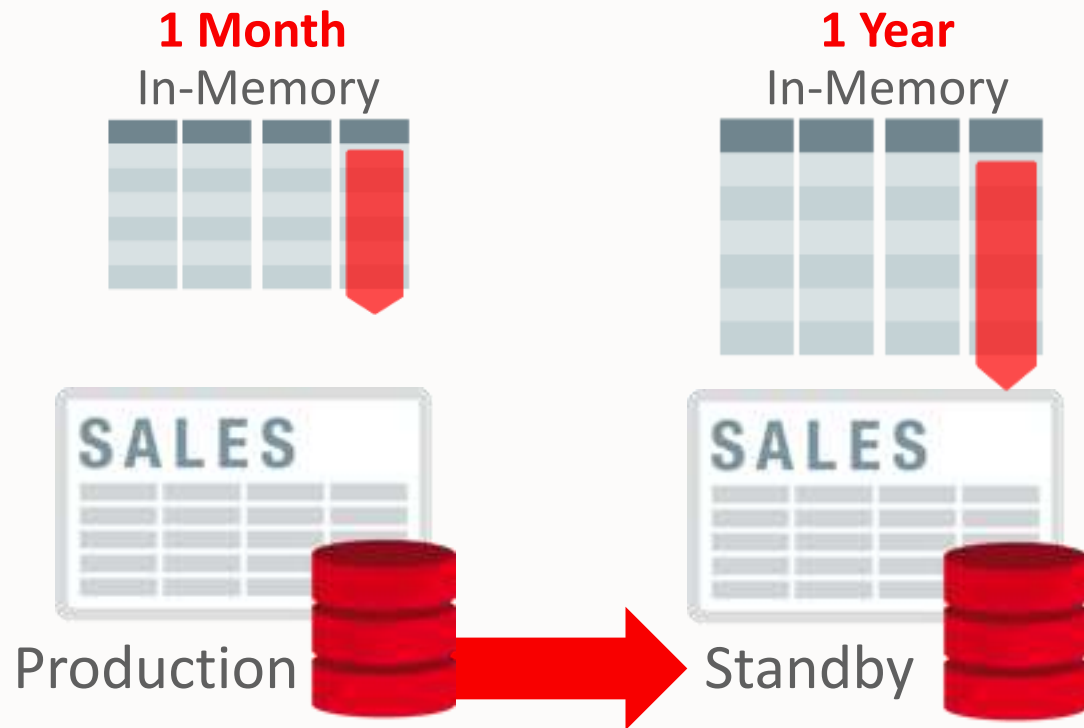
Engineered Systems: Unique Fault Tolerance



Only Available on Engineered Systems

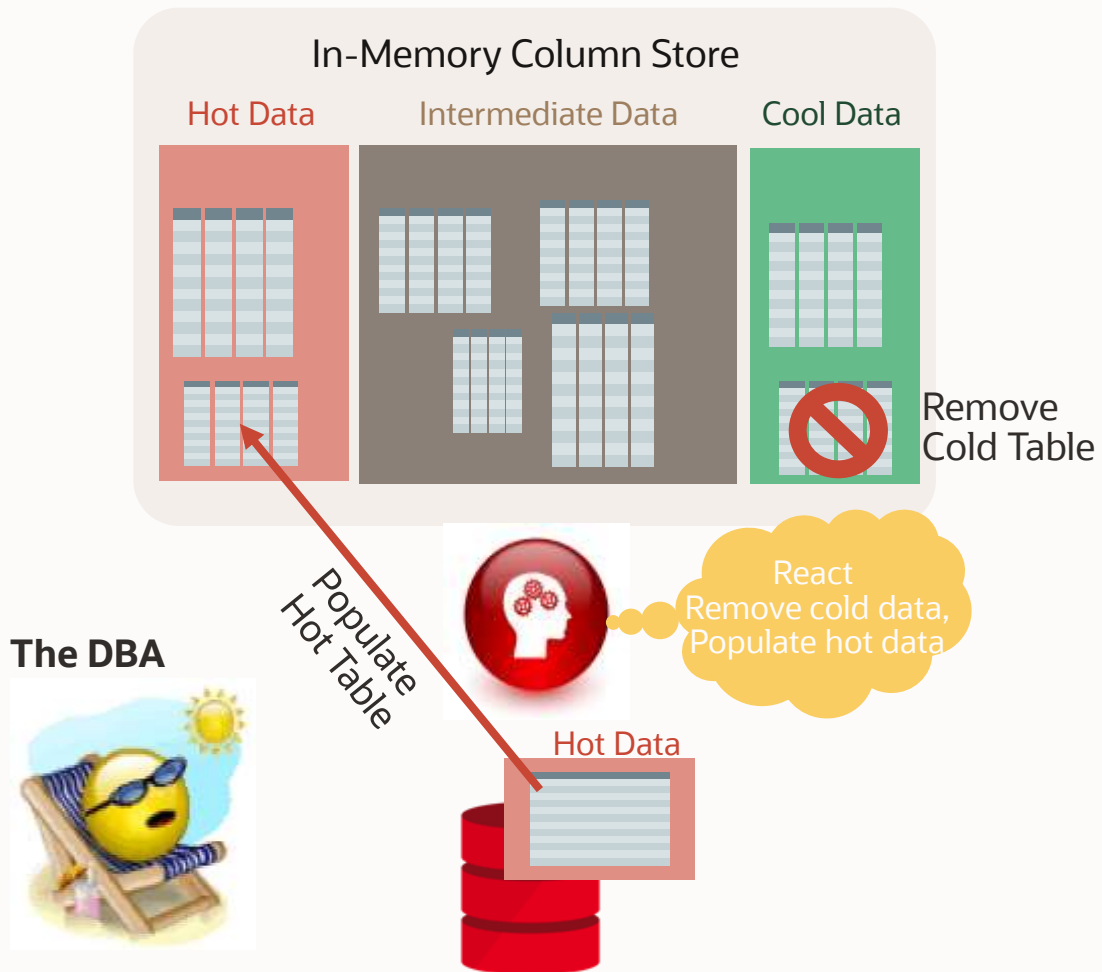
- Similar to storage mirroring
- Duplicate in-memory columns on another node
 - Enabled per table/partition
 - Application transparent
- Performance preserved by using duplicate during a node failure
- Performance can be improved by performing joins within each node (partial partition wise joins)

Mixed Workload: In-Memory on Active Data Guard



- Real-time analytics with no impact on primary database
- Makes full use of memory on standby system
- Standby can populate different data than production database
- Available on Exadata and PaaS Cloud Services

Automatic In-Memory



- Eliminates trial and error regarding in-memory area contents
- Constant background action:
 - Classifies data as hot, intermediate or cold
 - Hotter in-memory tables automatically populated
 - Colder in-memory tables automatically removed
 - Intelligent algorithm takes into account space-benefit tradeoffs
- Controlled by new parameter `inmemory_automatic_level`
- Useful for autonomous cloud services since no user intervention required

Database In-Memory Innovations

21c

- **Self Managing In-Memory**
- In-Memory Spatial Analytics
- In-Memory Full Text Columns
- External Table Enhancements
- Hybrid Scans
- JSON Data Type
- Vector Joins
- Base Level Feature

19c

- Performance
- External Tables: Hive & HDFS
- Memoptimized Rowstore – Fast Ingest

21c

Self-Managing, Convergence

18c

- **Automatic In-Memory**
- In-Memory Dynamic Scans
- In-Memory External tables
- In-Memory Optimized Arithmetic
- Memoptimized Rowstore – Fast Lookup

19c

Performance, Automation

18c

Performance, Capacity

12.2

- Join Groups
- In-Memory Expressions
- JSON/OSON support
- **Massive capacity - In-Memory on Exadata flash**
- Auto population policies
- Fast-Start
- Active Data Guard

12.2

12.1

- Pure In-Memory column format
- Scan & Filter on compressed data
- Fast joins
- Data pruning via storage indexes
- SIMD vector processing
- In-Memory aggregation

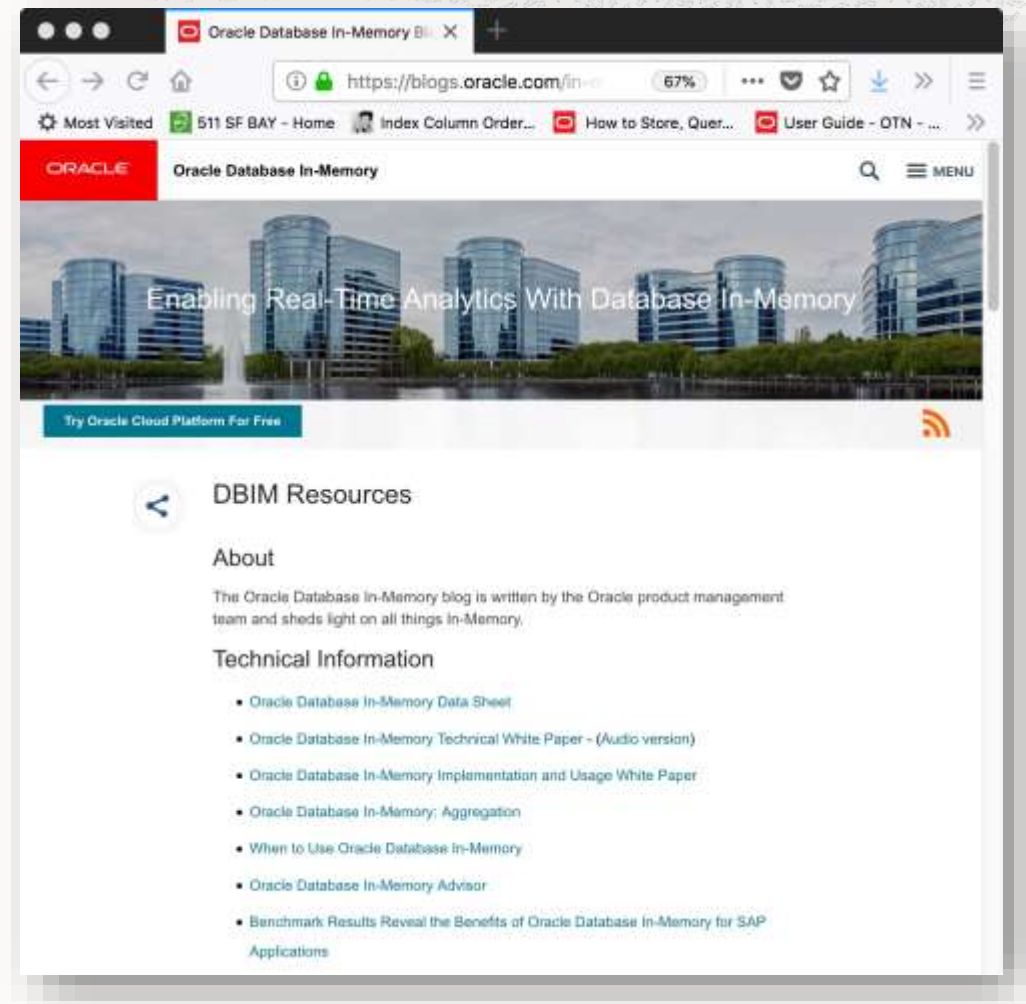
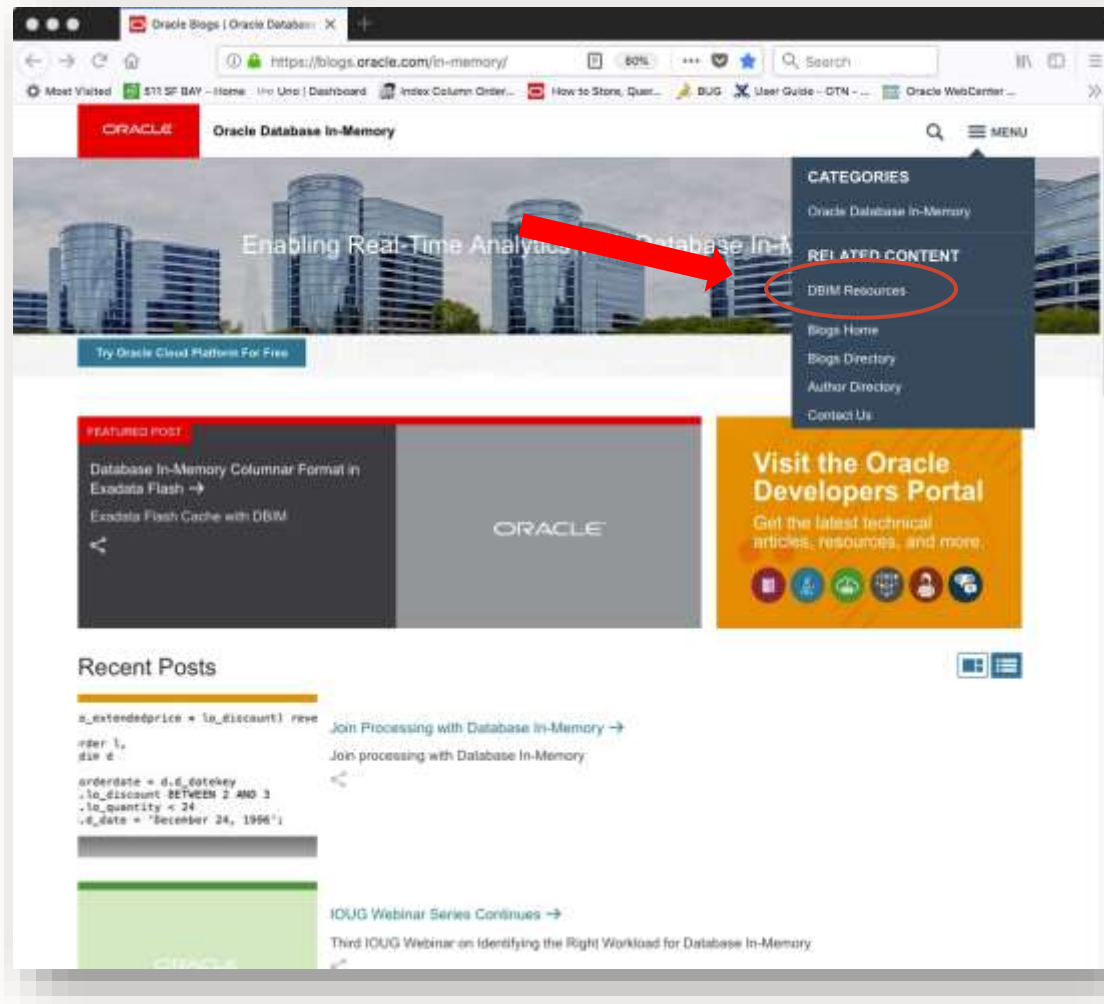
12.1



Database In-Memory Base Level Feature

- Customers can now use up to a **16GB** IM column store without having to license the Database In-Memory option
- The purpose of this **Base Level** feature is to allow customers to see the value of Database In-Memory
- Not all Database In-Memory features are available with Base Level
- Available in 21c and starting with the 19.8 RU
- No backports to previous versions are planned
- See Database In-Memory AskTOM Office Hours sessions and blog posts

https://blogs.oracle.com/in-memory/dbim-resources



ORACLE