ORACLE

25. hroug
godišnja konferencija

Oracle Database In-Memory

Best Practices for Getting Started, HrOUG 2021

Andy Rivenes
Database In-Memory Product Management
Oracle Corporation
Email: andy.rivenes@oracle.com
Twitter: @TheInMemoryGuy
Blog: blogs.oracle.com/in-memory

Just The Highlights

- This presentation will just briefly cover the highlights
- More details available on the Database In-Memory Resources page:

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

  - Database In-Memory Quick Start Guide
  - Database In-Memory Implementation Guidelines
- Don't forget the documentation:
  - Database In-Memory Guide

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?

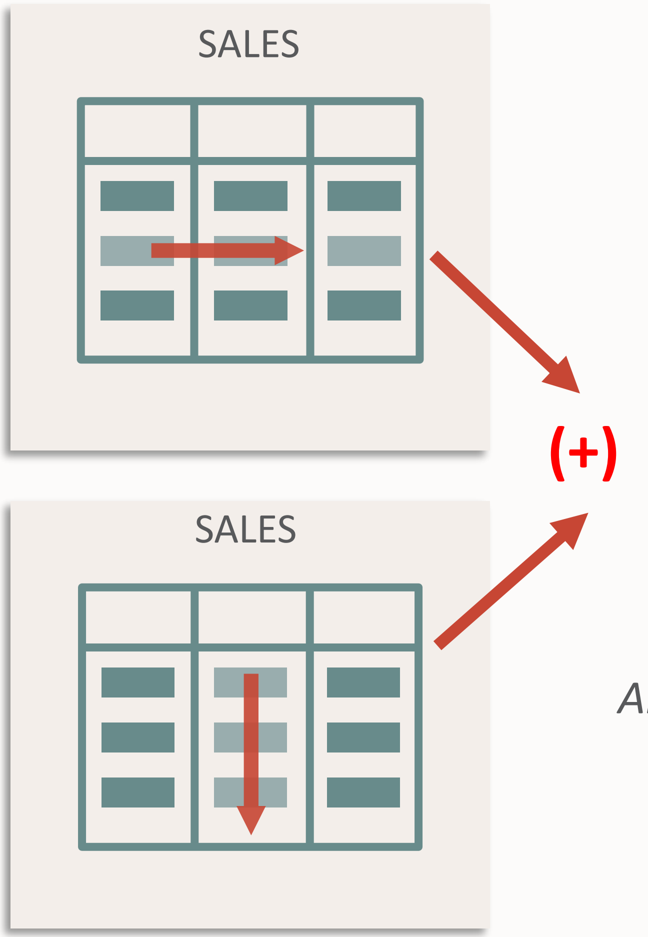
If I get a 20% discount on widget A, how much will our margins improve?

Introducing Database In-Memory

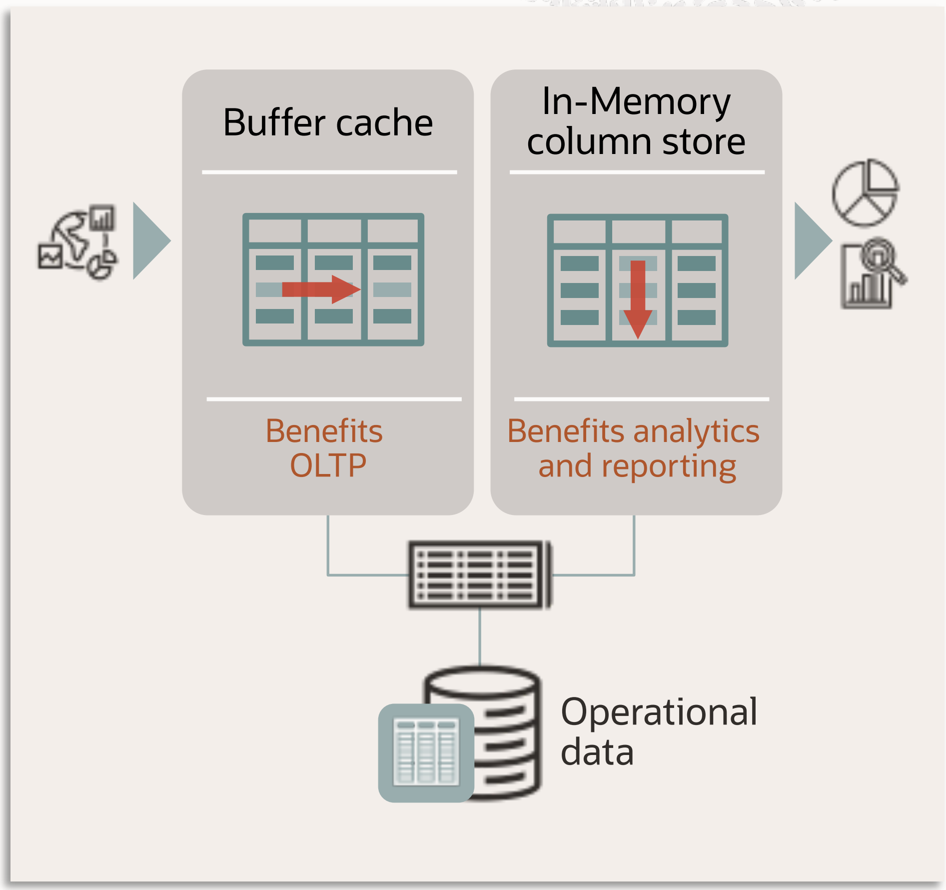
What's your favorite data format?

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

How Do You Know If You Will Benefit From Database In-Memory?

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 oracle.com 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

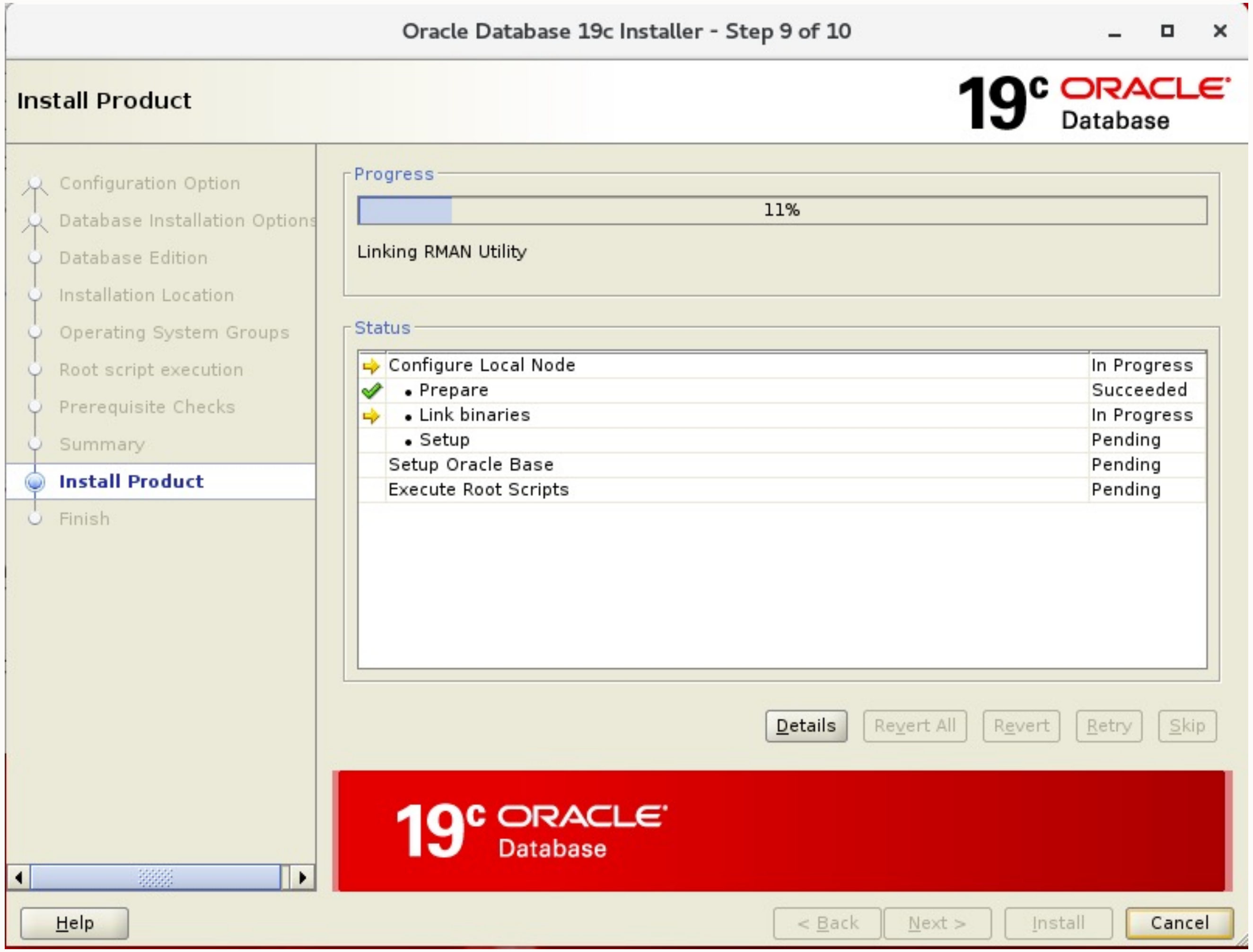
SQL Id	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Time Reduction (Seconds) With Unlimited Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 9.141GB	Estimated Analytics Processing Performance Improvement Factor With 9.141GB
fp83uwmbzt8zd	select cf.custid, sum(act.purchase_amt) sales from all_card_trans act, cust_fact cf ...	990	696	3.4X	696	3.4X
7zkhj3xhq01w8	with gold_member_aff_cust as (select custid, aff_cc_num from cust_fact w...	940	660	3.4X	660	3.4X
8p8ggufpp7699	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

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	36330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5598.SYS_SUBP5593	Memory compress for query low	1.063MB	1	0	3.0X	36330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5620.SYS_SUBP5615	Memory compress for query low	1.063MB	1	0	3.0X	28577 : 1

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

Use A Current Version of Oracle Database

Installing Oracle Database In-Memory



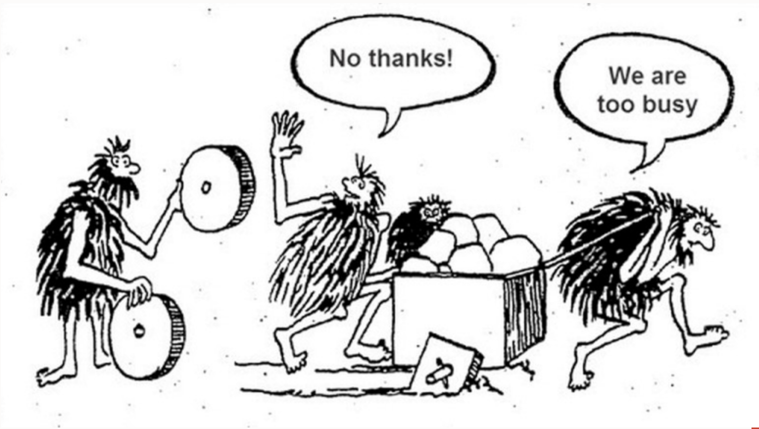
- Automatically installed as part of Oracle Database
- Strongly recommend using a current version (19c or 21c)
- No additional steps required

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

Installing: Apply the Latest Database Proactive Bundle Patch or Release Update

- Database In-Memory fixes and enhancements are only distributed through Database Proactive Bundle Patches or Release Updates
- See MOS Notes:
  - 2337415.1 – Overview of Database Patch Delivery Methods for 12.2.0.1 and greater
  - 1962125.1 – Overview of Database Patch Delivery Methods for 12.1.0.2 and older

- Starting with the latest patches avoids re-inventing the wheel discovering bugs that have already been fixed!



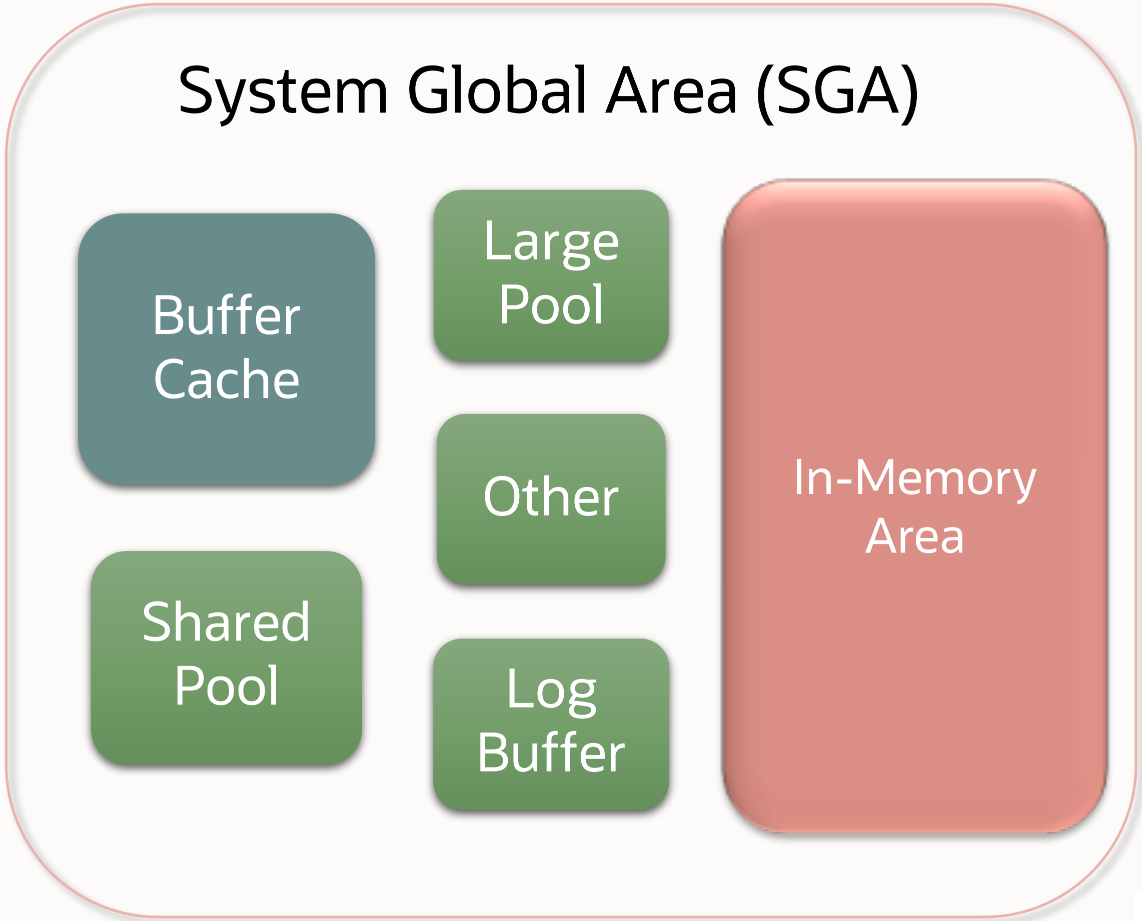
What About Upgrades?

Database Parameter Settings

- Evaluate all non-default initialization parameters
- Strongly consider unsetting any underscore parameters unless you are sure they are needed

Don't Steal Memory!

In-Memory Area: Static Area within SGA

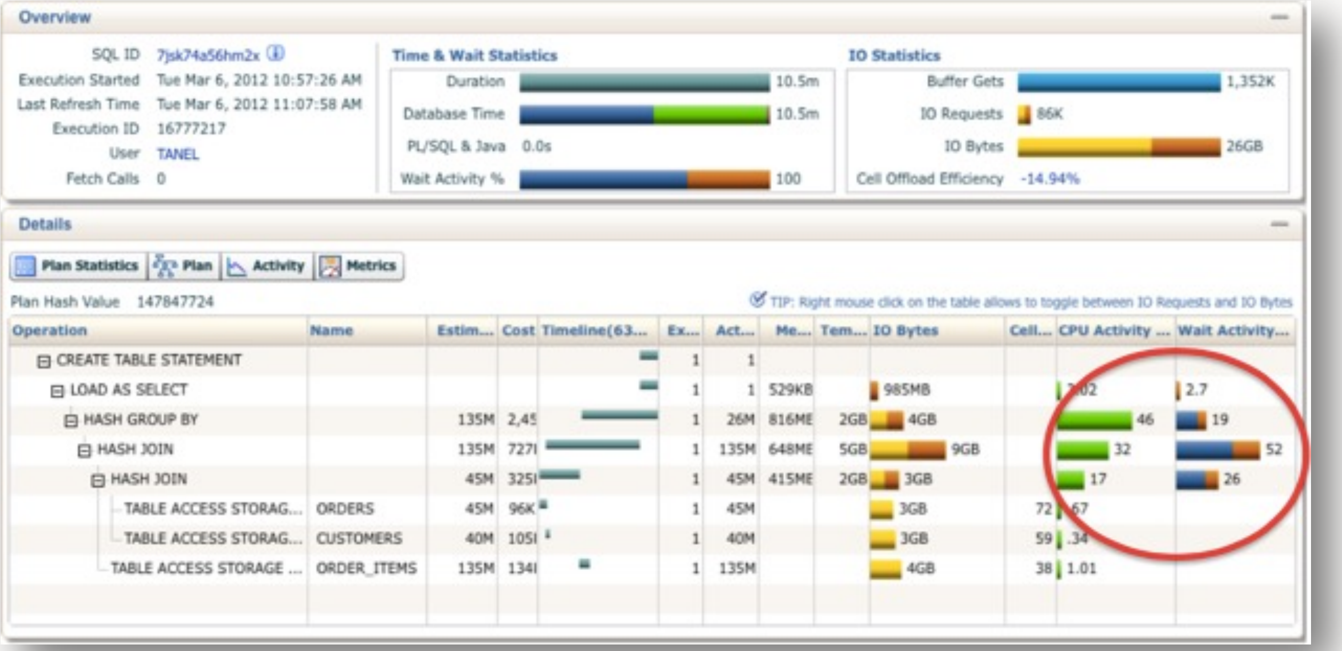


- Contains data in the new In-Memory Columnar Format
- Controlled by INMEMORY_SIZE parameter
  - Minimum size of 100MB
- Can be re-sized larger while database is running (starting in12.2)
- SGA_TARGET must be large enough to accommodate In-Memory area

Note: Don't steal Memory from other components

Configuring : In-Memory Column Store

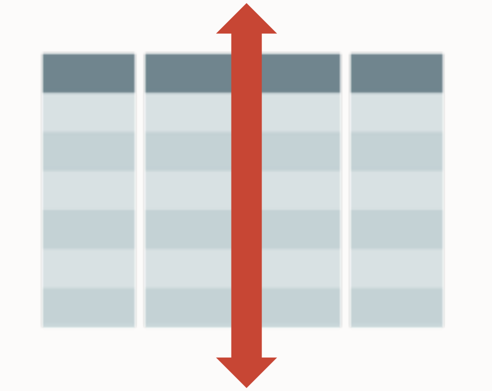
Don’t get carried away

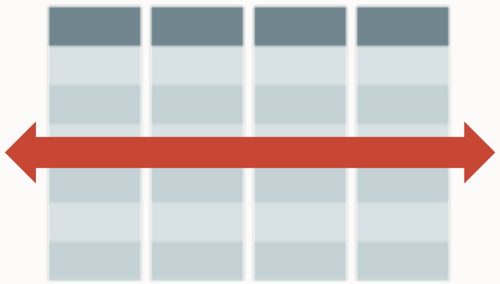


- Don’t give all memory to SGA
- Don’t want any aspect of execution plan to spill to disk
- Ensure PGA_TARGET is set large enough to keep joins & sorts in memory
- Use Parallel Execution to maximize PGA usage

Tip: Column Store Sizing

Allocate Extra Room





- The column store is fixed in size and objects are fully populated, but …
  - Inserts will cause the object to grow when the new rows are populated
  - Updates can cause existing rows to expand (i.e. column values) which can affect the size of a re-populated IMCU
- Additional space should be reserved in the column store to allow for DML activity

Fully Populate Your Data

Tip: View In-Memory Area Usage

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

```
SQL> SELECT * FROM v$inmemory_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
```

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

```
SQL> SELECT owner, segment_name, populate_status,
     inmemory_size, bytes_not_populated
     FROM v$im_segments;
                                                    Bytes Not
OWNER  NAME       STATUS          In-Memory Size  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
```

Monitoring : In-Memory Column Store Population

V$IM_SEGMENTS

- Indicates which objects are currently populated in-memory
- Shows current size of each segment in-memory
- Shows how much remains to be populated

```
SQL> select segment_name, populate_status, inmemory_priority, inmemory_size,
bytes_not_populated from v$im_segments;

SEGMENT_NAME POPULATE_STATUS INMEM_PRIORITY    INMEM_SIZE BYTES_NOT_POPULATED
------------ --------------- -------------- ------------ -------------------
ACCOUNTS     STARTED         HIGH                 196606          2434886912
SALES        COMPLETED       CRITICAL          135790592                   0
```

What If You Don't Have Enough Memory?

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

Compression

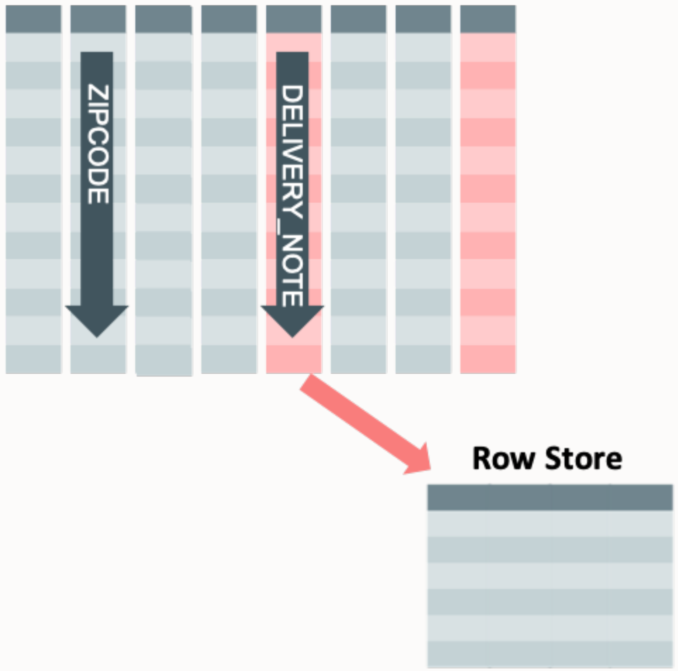
```
ALTER MATERIALIZED VIEW mv1
INMEMORY
MEMCOMPRESS FOR QUERY LOW;

CREATE TABLE trades
   (Name varchar(20),
    Desc varchar(200))
INMEMORY
MEMCOMPRESS FOR DML(desc);
```

- Objects compressed during population
- New compression techniques
  - Focused on scan performance
- 2x to 20x compression typical
- Multiple levels of compression
  - FOR DML
  - FOR QUERY LOW/HIGH
  - FOR CAPACITY LOW/HIGH
- Possible to use a different level for different partitions in a table

Columns Can Be Excluded

```
ALTER TABLE sales INMEMORY
NO INMEMORY (delivery_note);
```

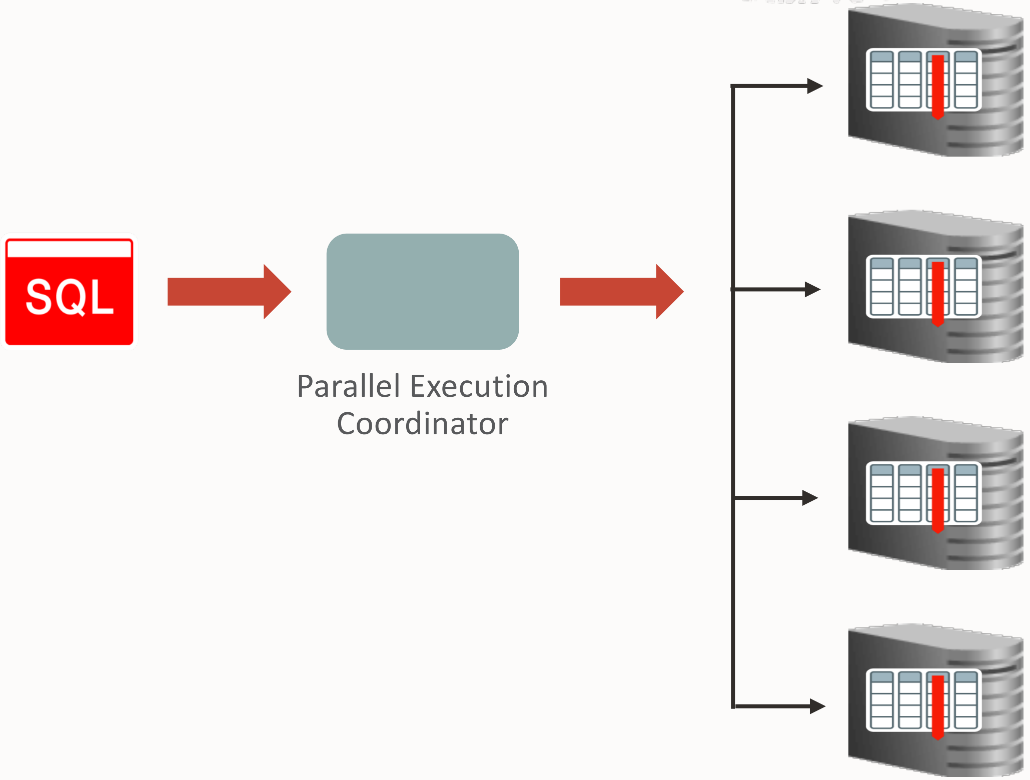


- You don't have to populate all columns
  - If excluded columns are accessed then the query will run against the row-store
  - In 21c, In-Memory Hybrid Scans supports both in-memory scans and row-store access for excluded projection columns
- Two phase approach
  1. INMEMORY attribute on table automatically inherited by columns
  2. Need to remove attribute from the columns you don't want populated

How Does Database In-Memory Work With RAC?

Parallel Query on RAC with Database In-Memory

- Scale-Out across servers to grow memory and CPUs
- Shared nothing architecture
- **IMCUs not shipped across interconnect** – cache fusion is not in play!
- In-Memory **queries are parallelized** across servers to access local columnar data



RAC : In-Memory and Distribution of Data

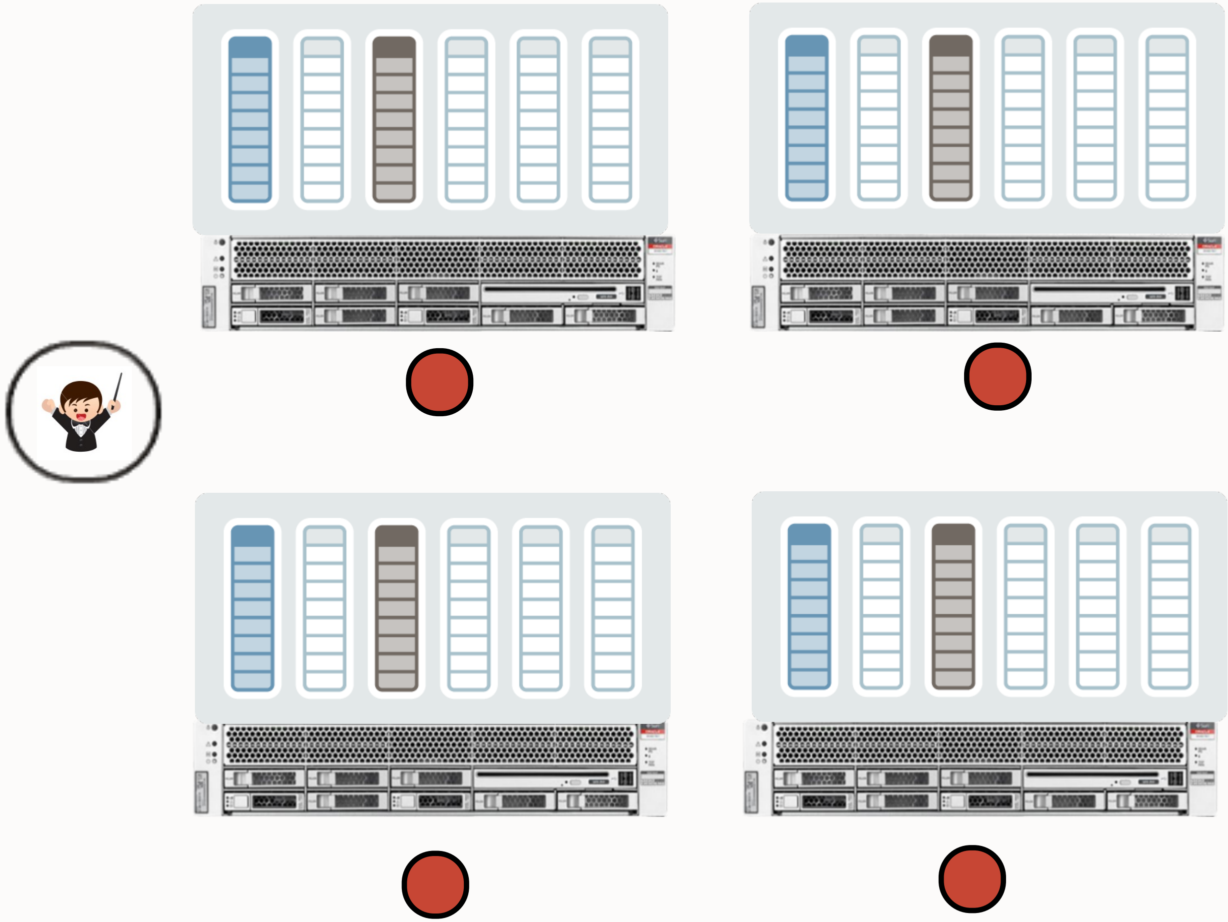
```
ALTER TABLE sales INMEMORY;

ALTER TABLE sales INMEMORY
DISTRIBUTE BY PARTITION;

ALTER TABLE sales INMEMORY
DISTRIBUTE ROWID RANGE;
```

- Distribution allows in memory segments larger than individual instance memory
- Policy is automatic (Distribute AUTO) or user-specifiable
- Controlled by DISTRIBUTE subclause
  - Distribute by rowid range
  - Distribute by partition
  - Distribute by subpartition
- Goal: Ensure Even Distribution

Querying In-Memory data in a RAC environment



- Shared nothing architecture means **Parallel Query** must be used to access data
- Must have a DOP greater than or equal to the number of column stores
- Query coordinator automatically starts parallel server processes on the correct nodes (Requires Auto DOP in 12.1.0.2)

How Does Database In-Memory Work With Other Database Features?

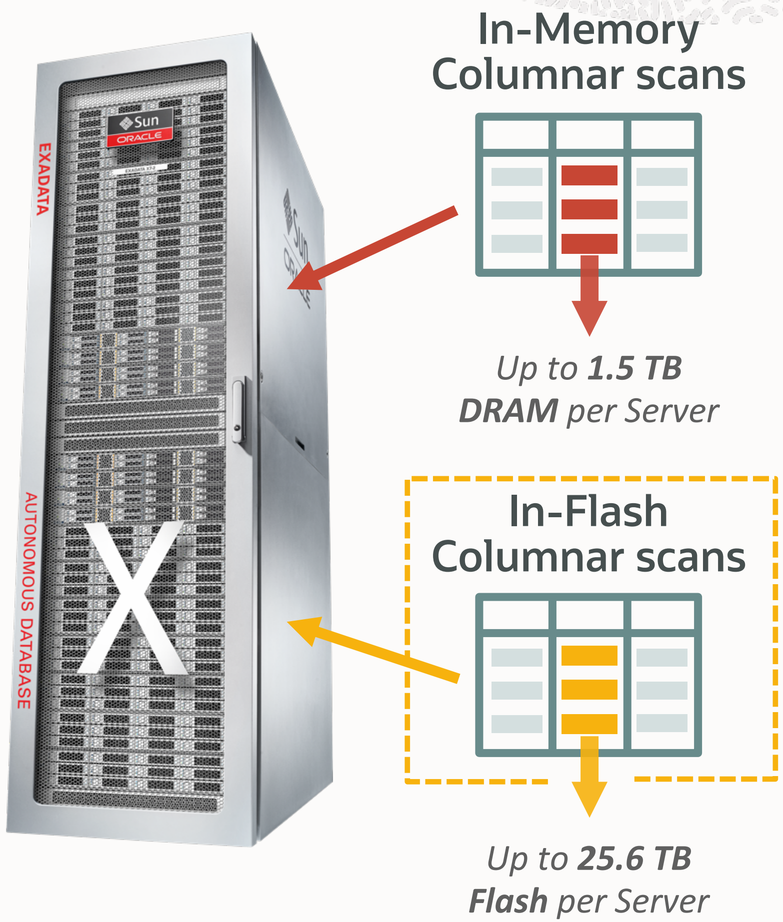
Why In-Memory on Exadata: Unique Features

Unique to Exadata

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

Available on All Flavors of Exadata

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

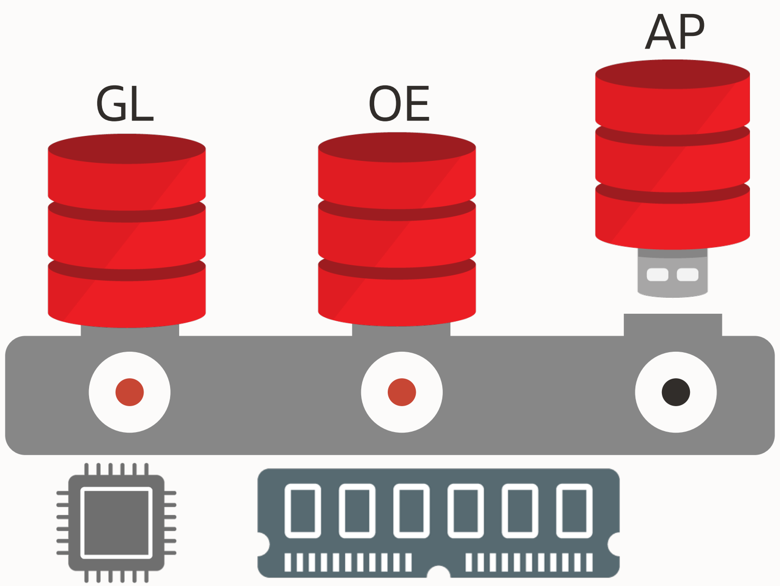


Database In-Memory Works (Better) with Multitenant

Multitenant consolidation “gives back” resources that DB In-Memory needs

Consolidation with Multitenant frees up

- Memory
- CPU Cycles



Oracle Database In-Memory wants

- Memory
- CPU Cycles

How Do I Tell If The In-Memory Column Store Is Being Used?

Target The Right Workloads

Understand Where In-Memory Helps

- In-Memory speeds up analytic data access, not:
  - DML (insert, update, delete)
  - Network round trips, logon/logoff
  - Parsing, PL/SQL, complex functions
  - Data processing (as opposed to access)
    - Complex joins or aggregations where not much data is filtered before processing
  - Load and select once – Staging tables, ETL, temp tables
- Process data in sets of rows in the Database and not one row at a time in the application

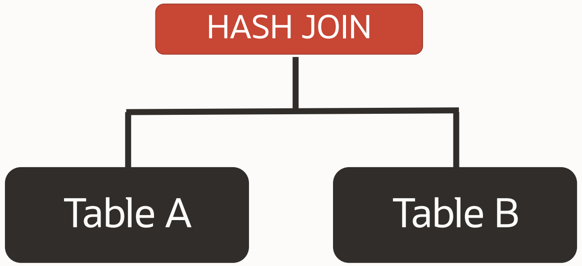
Know your bottleneck!

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



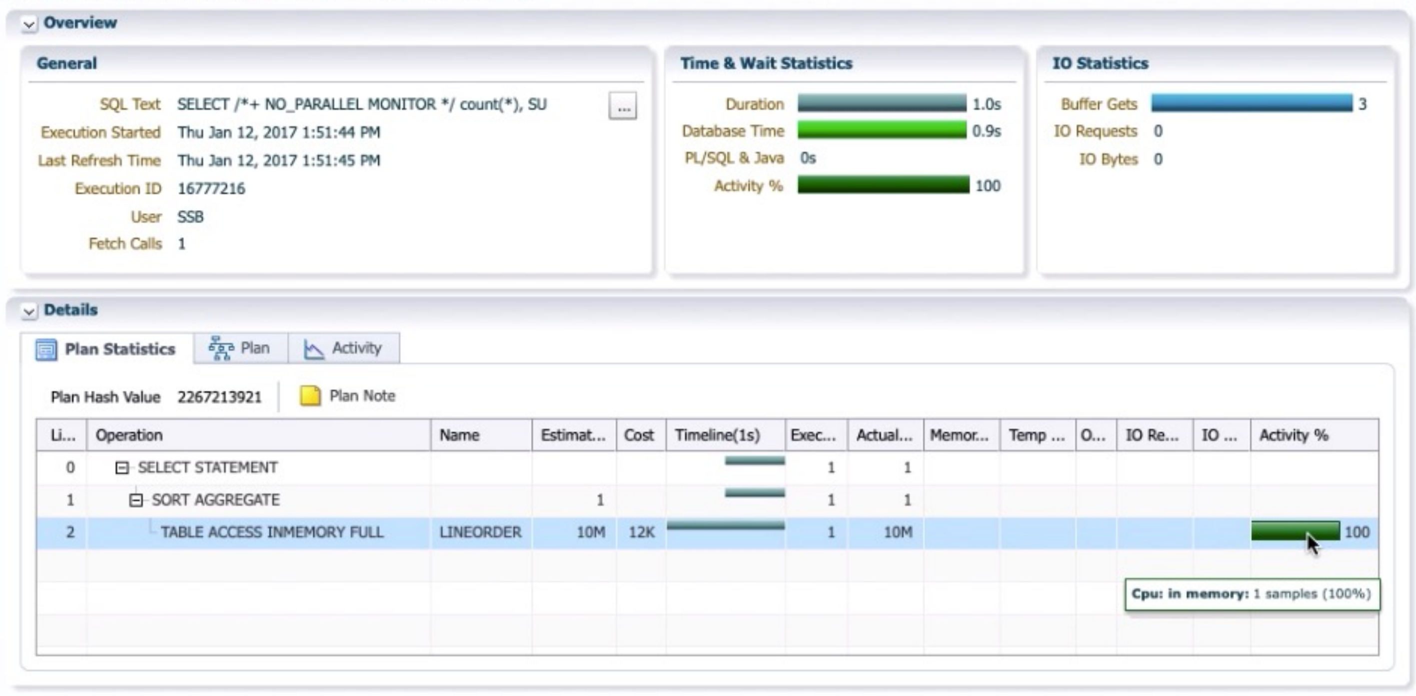
is spent aggregating data

Use Time Based Analysis Techniques To Evaluate Benefit

SQL Monitor Active Reports

- Shows how SQL was executed and where **time was spent**

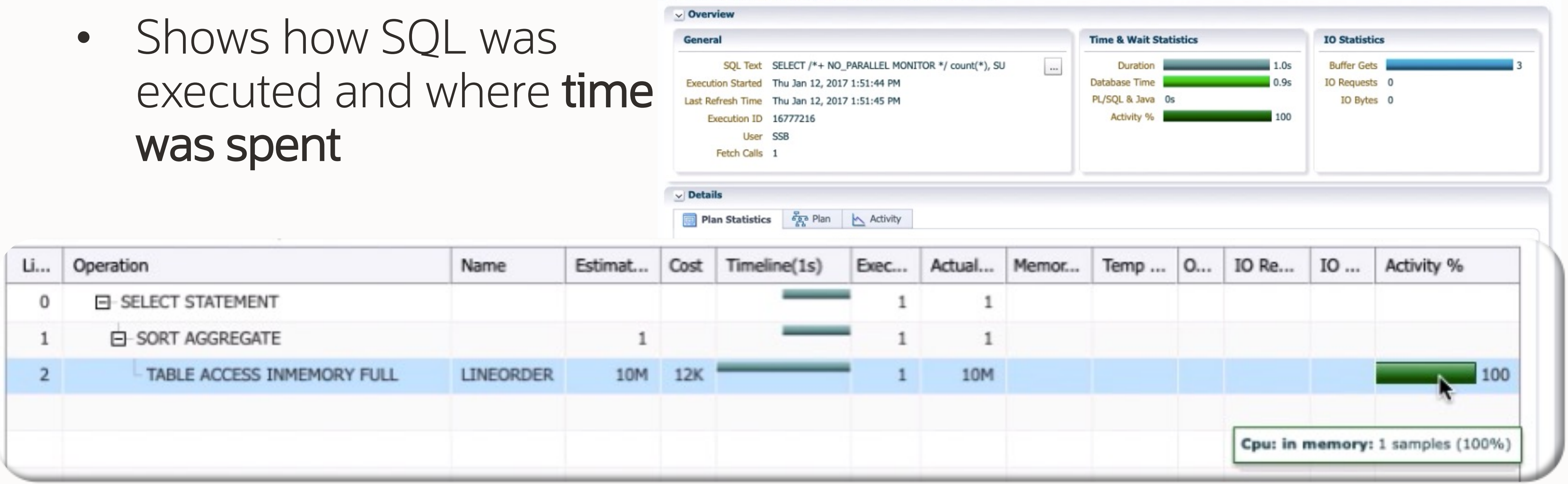
- See blogs.oracle.com/In-Memory for a technical brief on creating SQL Monitor active reports



Use Time Based Analysis Techniques To Evaluate Benefit

SQL Monitor Active Reports

- Shows how SQL was executed and where **time was spent**



Tip: Identifying In-Memory Benefits

- Session level statistics
- Best way to determine if In-Memory was used
- Best way to measure the benefits of an In-Memory scan
- See **blogs.oracle.com/in-memory** for descriptions of the key statistics

IM scan bytes in-memory
IM scan bytes uncompressed
IM scan CUs columns accessed
IM scan CUs columns decompressed
IM scan CUs columns theoretical max
IM scan rows
IM scan rows range excluded
IM scan rows excluded
IM scan rows optimized
IM scan rows projected
IM scan CUs predicates received
IM scan CUs predicates applied
IM scan CUs predicates optimized
IM scan CUs pruned
IM scan segments minmax eligible
....

What's New?

Database In-Memory Innovations

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

12.2

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

18c

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

19c

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

21c

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

12.1

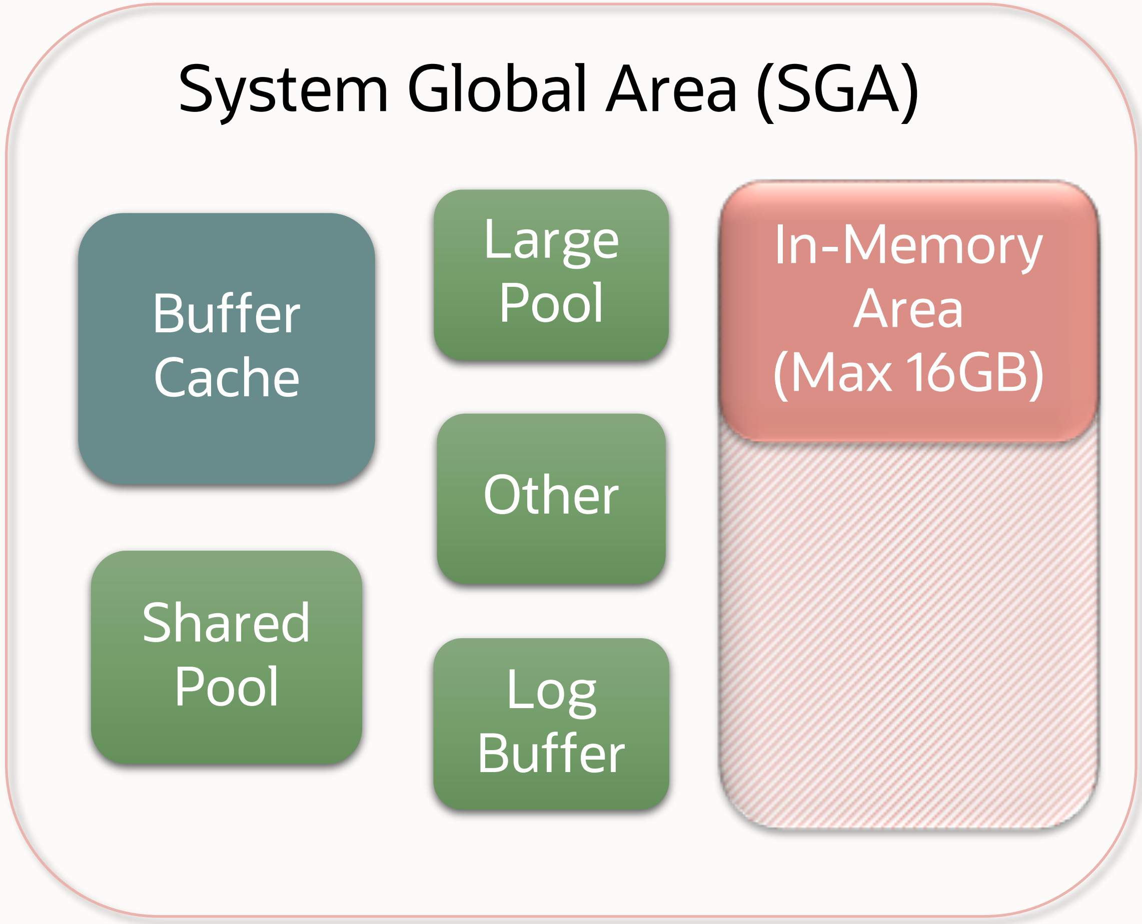
12.2 — Performance, Capacity

18c — Performance, Automation

19c

21c — Self-Managing, Convergence

IM Column Store With The Base Level Feature



- A 21c feature
- **Now available in 19c with the July 19.8 Release Update**
- Enabled with INMEMORY_FORCE parameter
  - Must be set to BASE_LEVEL
- INMEMORY_SIZE limited to a maximum of 16GB
- Feature tracking reports usage as "In-Memory Base Level"

Where Can You Get More Information?

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