Oracle Database In Memory Option

Zohar Elkayam

CTO, Brillix

Zohar@Brillix.co.il

Twitter: @realmgic

Agenda

- About in memory stores
- About column stores
- Introduction to Database In Memory – Oracle In Memory Option
- How to use DBIM
- Short demo

BRILLIX
delivering brilliant DBAs

Who am I?

- Zohar Elkayam, CTO at Brillix
- Oracle DBA, team leader, instructor and senior consultant for over 16 years
- Editor (and manager) of ilDBA – Israel Database Community
- Blogger – ZoharElkayam.wordpress.com

BRILLIX
delivering brilliant DBAs

What are In Memory Databases and Column Stores?

BRILLIX
delivering brilliant DBAs

What is a In Memory Database?

In memory databases are management systems that keeps the data in a non-persistent storage (RAM) for faster access

Examples:

- MemcacheDB
- Oracle TimesTen

BRILLIX
delivering brilliant DBAs

What is a Column Store Database?

- Column Store databases are management systems that uses data managed in a columnar structure format for better analysis of single column data (i.e. aggregation). Data is saved and handled as columns instead of rows.

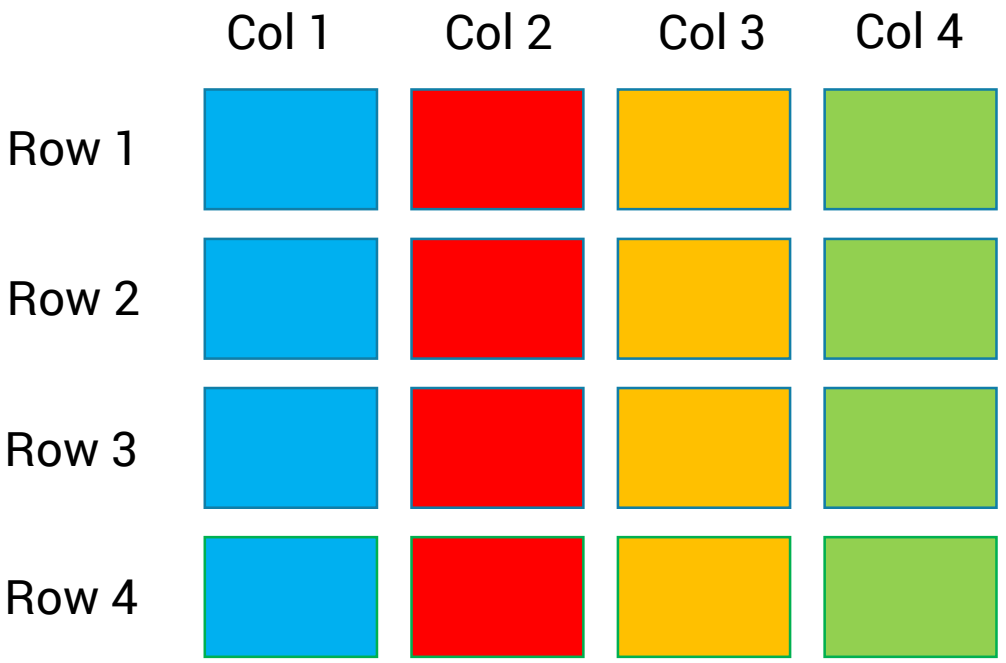
Examples:

- Vertica
- GreenPlum
- HBase

BRILLIX
delivering brilliant DBAs

How Records are Organized?

- This is a logical table in RDBMS systems
- Its physical organization is just like the logical one: column by column, row by row



BRILLIX
delivering brilliant DBAs

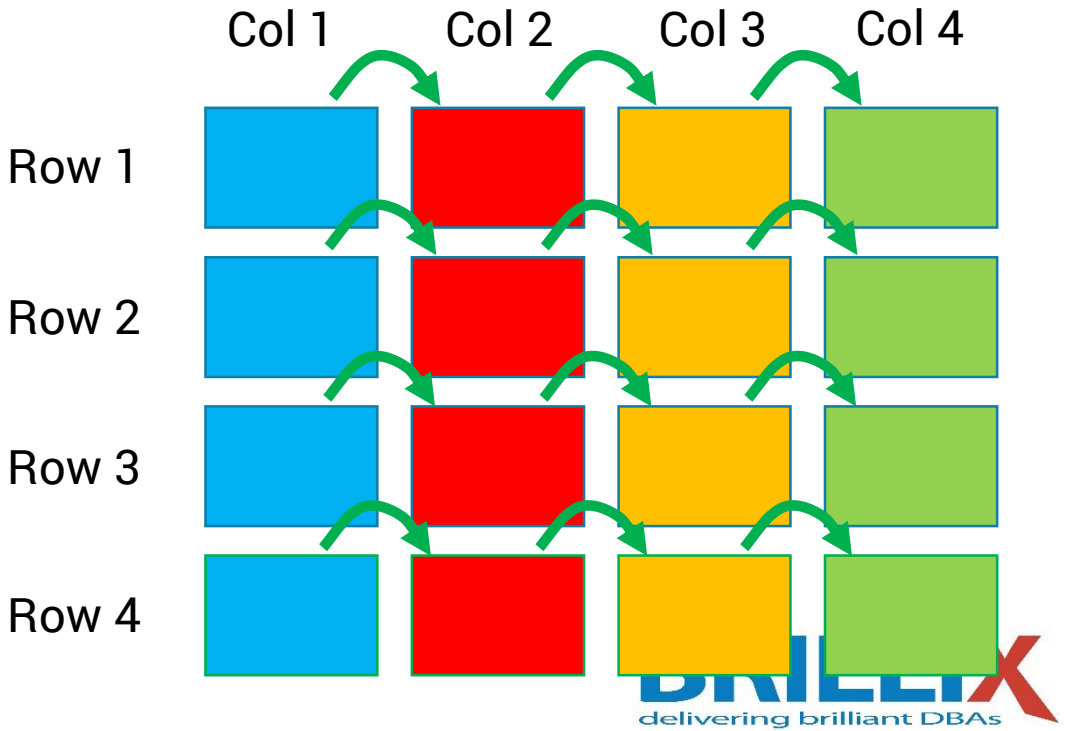
Query Data

- When we query data, records are read at the order they are organized in the physical structure

```
Select *
From MyTable
```

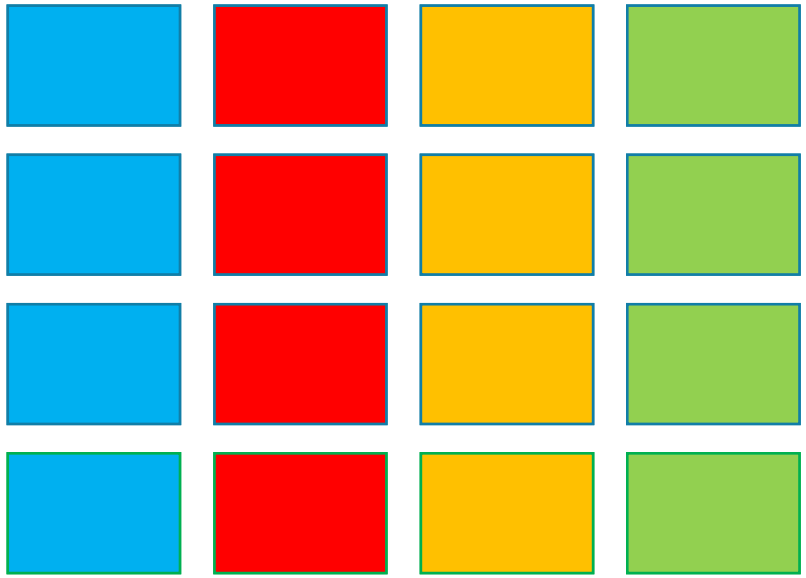
- Even when we query a single column, we still need to read the entire table and extract the column

```
Select Col2
From MyTable
```

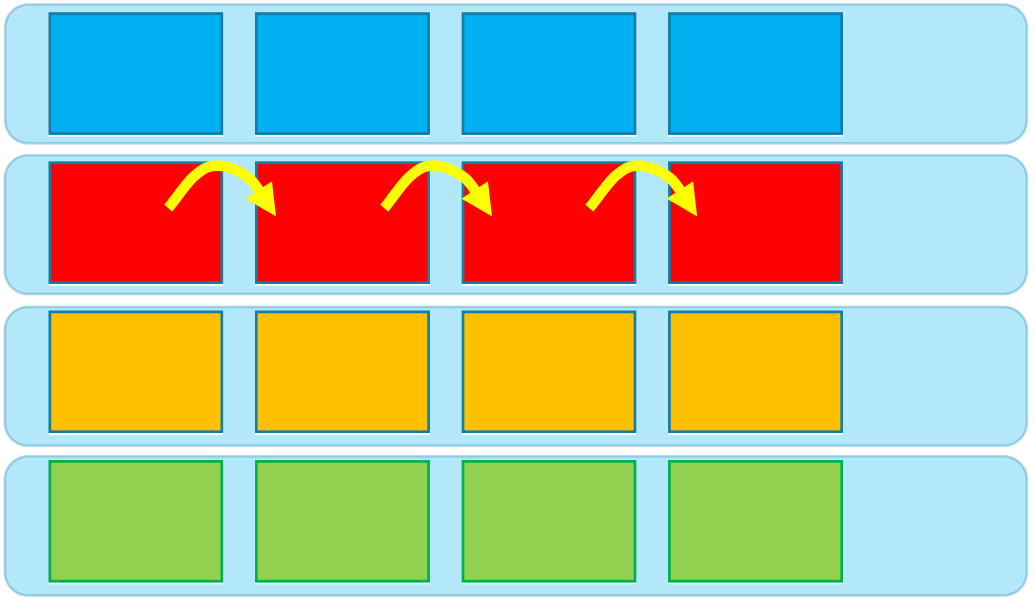


How Does Column Store Save Data

Organization in row store



Organization in column store



BRILLIX
delivering brilliant DBAs

Row Format vs. Column Format

Row

- **Transactions** run faster on row format
  - Example: Insert or query a sales order
  - Fast processing few rows, many columns

Column

- **Analytics** run faster on column format
  - Example : Report on sales totals by region
  - Fast accessing few columns, many rows

BRILLIX
delivering brilliant DBAs

Column Store Limitations

- Most Column stores avoid or limit online transactions
- Most Column stores avoid data changes (updates) and implement it as insert/delete
- Columnar organization is very good for large quantities of data but is not very efficient when storing smaller amounts of data
- Column store SQL might be somewhat different from ANSI SQL

BRILLIX
delivering brilliant DBAs

In Memory Option

BRILLIX
delivering brilliant DBAs

In Memory Option Breakthrough

- In memory option introduces a dual format database

- Tables can be accessed as row format and column format at the same time – the Optimizer is aware to the new format so:
  - OLTP continue using the old row format
  - Analytic queries start using the column format

BRILLIX
delivering brilliant DBAs

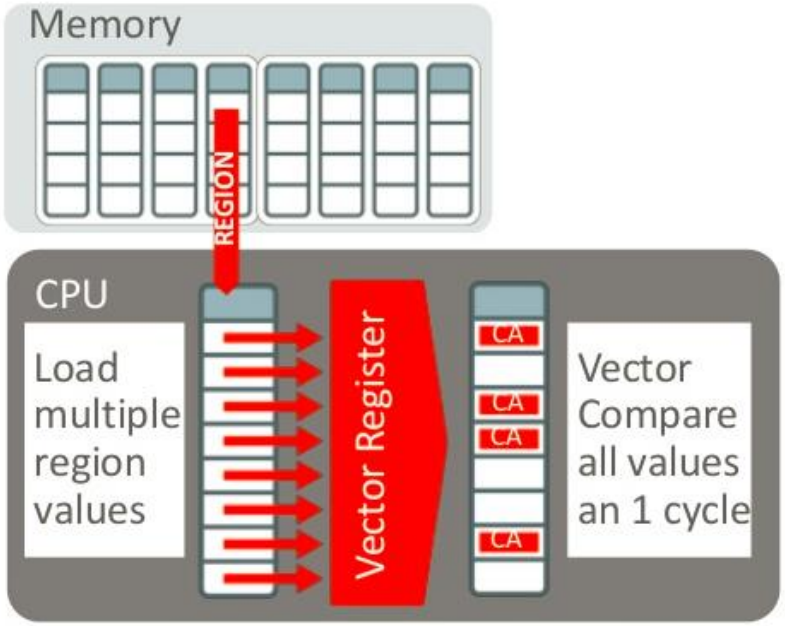
In-Memory Option

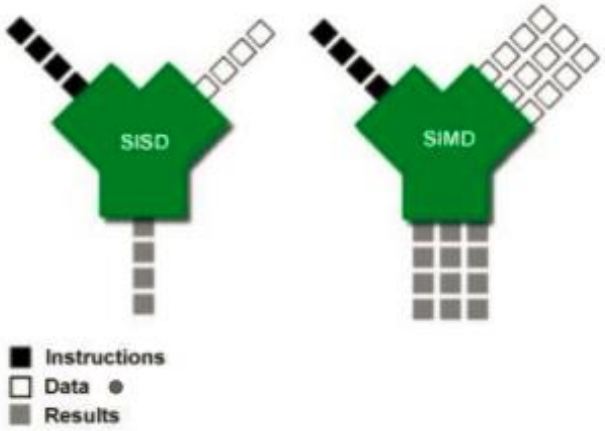
- Column data is pure in memory format and therefor is non-persistent and require no logging, archiving or backup
- Data changes are simultaneously changed in both formats so data is consistent
- Since the data is column based, it has better compression ratio and more data can be stored in memory
- There are no changes to the application required – just turn on and start using

BRILLIX
delivering brilliant DBAs

Order of Magnitude Faster

- Customer report analytic queries run 10 to 1000 times faster
- Since less analytic Indexes are needed, OLTP run faster too
- Data processing is done using SIMD Vector Instruction so billions of records can be handled by a single CPU





BRILLIX
delivering brilliant DBAs

In Memory Option – Good To Know

- Oracle 12.1.0.2 feature – additional license required
- It is **Not** In Memory Database – it's an accelerator to our current database
- It is **Not** Column Store Database – it allows keeping some of our data in column store which is non-persistent
- It has nothing to do with Times-Ten or Oracle Coherence

BRILLIX
delivering brilliant DBAs

More Good to Know

- In memory option does not work on an Active Data Guard standby database but roadmap does contains this feature in the future. It does work on logical standby database
- Optimization of the execution plan in regards of using the DBIM might (and probably will) change in future versions
- Known bugs: no functionality bugs known at this time but there is a known bug around the license auditing of this feature

BRILLIX
delivering brilliant DBAs

How To Use DBIM?

BRILLIX
delivering brilliant DBAs

How to Configure

- Configure memory capacity

```
inmemory_size = xxx GB
```

- The amount of memory allocated must be larger than the amount of data loaded (after compression). We should also allow some spare memory for maintenance.

- Configure tablespaces, tables, partitions, sub-partitions or columns to be in memory:

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

- Optional: drop unused OLTP indexes – those who used for analytics.

BRILLIX
delivering brilliant DBAs

Startup

```
SQL> startup
ORACLE instance started.

Total System Global Area 5368709120 bytes
Fixed Size                 3056960 bytes
Variable Size            620759744 bytes
Database Buffers        1509949440 bytes
Redo Buffers              13717504 bytes
In-Memory Area          3221225472 bytes
Database mounted.
Database opened.
```

BRILLIX
delivering brilliant DBAs

Parameters Related to In Memory Option

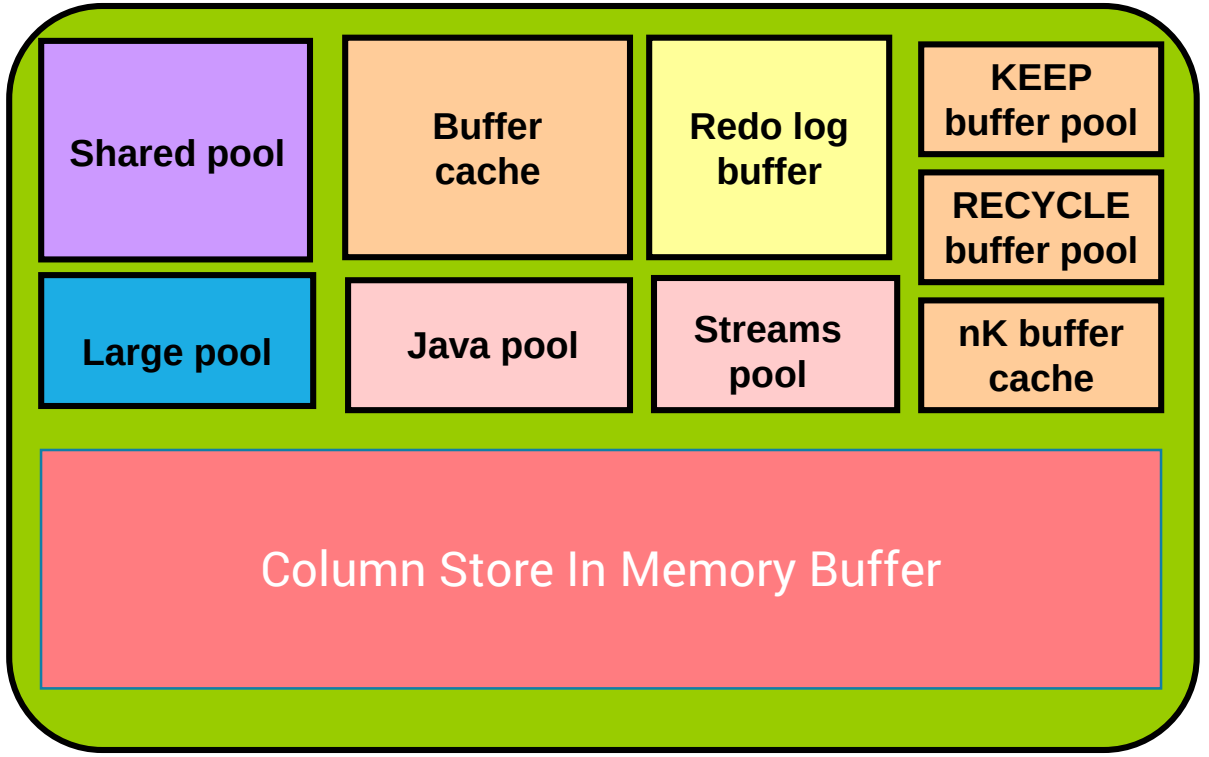
```
SQL> show parameter inmememory

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
inmemory_clause_default              string
inmemory_force                       string      DEFAULT
inmemory_max_populate_servers        integer     2
inmemory_query                       string      ENABLE
inmemory_size                        big integer 3G
inmemory_trickle_repopulate_servers_ integer     1
percent
optimizer_inmemory_aware             boolean     TRUE
```

BRILLIX
delivering brilliant DBAs

Where Does the In Memory Resides?

- In Memory is part of the SGA – adding this buffer will require changing the SGA size



System Global Area (SGA)

BRILLIX
delivering brilliant DBAs

Checking Table Settings

```
SQL> r
  1  SELECT table_name,
  2         inmemory,
  3         inmemory_priority,
  4         inmemory_distribute,
  5         inmemory_compression,
  6         inmemory_duplicate
  7  FROM   user_tables
  8* ORDER BY table_name

TABLE_NAME           INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_COMPRESS INMEMORY_DUPL
-------------------- -------- -------- --------------- ----------------- -------------
PEOPLE               ENABLED  HIGH     AUTO            FOR QUERY LOW     NO DUPLICATE
PEOPLE2              DISABLED
TOWNS                DISABLED
```

Managing In Memory at the Column Level

- Setting a table to In Memory automatically set all the column to be loaded to the memory
- If we want to load some of the columns we explicitly need to state which columns are NOT loaded to the memory:

```
CREATE TABLE im_col_tab (
  id   NUMBER,
  col1 NUMBER,
  col2 NUMBER,
  col3 NUMBER,
  col4 NUMBER
) INMEMORY
INMEMORY MEMCOMPRESS FOR QUERY HIGH (col1, col2)
INMEMORY MEMCOMPRESS FOR CAPACITY HIGH (col3)
NO INMEMORY (id, col4);
```

BRILLIX
delivering brilliant DBAs

Table Types That are Not Supported

- There are tables and column types which are not supported:
  - IOT
  - Clustered tables
  - Objects owned by SYS, SYSTEM or stored in the SYSTEM and SYSAUX tablespaces
  - LONG type columns
  - Out of line LOB

BRILLIX
delivering brilliant DBAs

How Data is Populated in the Memory

- By default, data is being populated in the in memory cache while first reading the table
- Objects could be configured to be loaded as the instance starts and prioritized according to the application needs
- There are 5 level of prioritization: None, Low, Medium, High and Critical. None means no pre-loading, critical means before all others.

BRILLIX
delivering brilliant DBAs

Checking Memory Population

```
SQL> r
  1  select segment_name,
  2         inmemory_size,
  3         bytes_not_populated,
  4         populate_status,
  5         inmemory_compression,
  6         bytes / inmemory_size comp_ratio
  7*   from v$im_segments

SEGMENT_NAME                   INMEMORY_SIZE BYTES_NOT_POPULATED POPULATE_ INMEMORY_COMPRESS COMP_RATIO
------------------------------ ------------- ------------------- --------- ----------------- ----------
PEOPLE                            853540864                   0 COMPLETED FOR QUERY LOW     2.43734644
```

How to Control In Memory Compression

User Chosen Compression Level	Optimize Mode	Details
BASIC	Populate Speed	No compression, fastest populate, slower queries
FOR QUERY	Throughput	Lightweight compression, fastest queries
FOR CAPACITY – LOW	Balanced	Fast decompress, balance between space and speed
FOR CAPACITY - HIGH	Space	Heavier weight decompressor, optimizes for space

BRILLIX
delivering brilliant DBAs

Explain Plan, No In Memory

```
SQL> select avg(salary), max(p.id_town_work), count(salary) from people p;

Elapsed: 00:00:23.56

Execution Plan
----------------------------------------------------------
Plan hash value: 470504681

-------------------------------------------------------------------------------
| Id  | Operation          | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |        |     1 |    26 | 68771   (1)| 00:00:23 |
|   1 |  SORT AGGREGATE    |        |     1 |    26 |            |          |
|   2 |   TABLE ACCESS FULL| PEOPLE |   160M|  3960M| 68771   (1)| 00:00:23 |
-------------------------------------------------------------------------------
```

BRILLIX
delivering brilliant DBAs

Query Statistics – No In Memory

```
Statistics
----------------------------------------------------------
          6  recursive calls
          0  db block gets
    2583630  consistent gets
    2501640  physical reads
          0  redo size
        726  bytes sent via SQL*Net to client
        552  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

BRILLIX
delivering brilliant DBAs

Explain Plan

```
SQL> select avg(salary), max(p.id_town_work), count(salary) from people p;
Elapsed: 00:00:01.10

Execution Plan
----------------------------------------------------------
Plan hash value: 470504681

-------------------------------------------------------------------------------------
| Id  | Operation                    | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |        |     1 |    26 |  2733   (6)| 00:00:01 |
|   1 |  SORT AGGREGATE              |        |     1 |    26 |            |          |
|   2 |   TABLE ACCESS INMEMORY FULL| PEOPLE |   160M|  3960M|  2733   (6)| 00:00:01 |
-------------------------------------------------------------------------------------
```

BRILLIX
delivering brilliant DBAs

Query Statistics – In Memory

```
Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
          9  consistent gets
          0  physical reads
          0  redo size
        726  bytes sent via SQL*Net to client
        552  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

BRILLIX
delivering brilliant DBAs

Disabling In Memory Option

- Disabling/Enabling in memory query at the session/system level at the optimizer level:

```
ALTER SESSION|SYSTEM SET INMEMORY_QUERY=DISABLE|ENABLE;
```

- Disabling object maintenance:

```
ALTER SYSTEM SET INMEMORY_FORCE=OFF|DEFAULT;
```

- Removing In Memory Option (requires restart):

```
ALTER SYSTEM RESET INMEMORY_SIZE SCOPE=SPFILE ;
SHUTDOWN IMMEDIATE ;
STARTUP ;
```

BRILLIX
delivering brilliant DBAs

Demo

BRILLIX
delivering brilliant DBAs

What Did We Not Talk About?

- Working with DBIM with RAC, Engineered systems and in multitenant environments
- What happens when there is not enough memory?
- In memory storage indexes and In memory joins
- Read consistency, IMCU staleness and trickle repopulate

BRILLIX
delivering brilliant DBAs

Conclusion

- The DBIM is a very interesting feature which can make analytic queries run much faster
- Working with in memory option requires understanding of the database model and relevant queries – memory is not infinite.
- No significant bugs found yet but it's very early
- Some behaviors may (and should) change in the future

BRILLIX
delivering brilliant DBAs

Thank You

Zohar Elkayam
Brillix
+972-54-4742963
Zohar@Brillix.co.il
ZoharElkayam.wordpress.com

BRILLIX
delivering brilliant DBAs