What Can You Accomplish with Oracle In-Memory?

datAvail

Anuj Pandey

Datavail
Oracle Database
Administrator

Over 16+ years of experience

Experienced

- 16+ years of Oracle DBA
- Multiple roles & industries

Accredited

- OCP certificated
- Master in Computer Application

Connect or follow me on LinkedIn:

linkedin.com/in/anujpandeydba

Fill Out the Session Evaluation Form to Enter to Win Free Beer For a Year!

Datavail Company Overview

We are database, analytics and application data integration specialists.

1	16+ years delivering data services
2	700+ customers, average client retention of 7 years
3	Managed services, projects, and staffing

Database Operations

- All major platforms: MS SQL, Oracle, DB2, MySQL, MongoDB, Postgres
- Comprehensive development & operational services
- 24x7 in-office delivery
- US & Global models
- Reactive & proactive services

Analytics, Visualization & Integration

- Microsoft SSRS, SSIS, SSAS
- PowerBI
- Oracle OBIEE, BICS, Golden Gate
- Informatica
- Oracle Data Integrator
- ETL development & operational run

Application Development, Integration & Maintenance

- Oracle EBS
- Hyperion
- Microsoft Dynamics CRM
- SharePoint
- Custom development (Microsoft .NET)

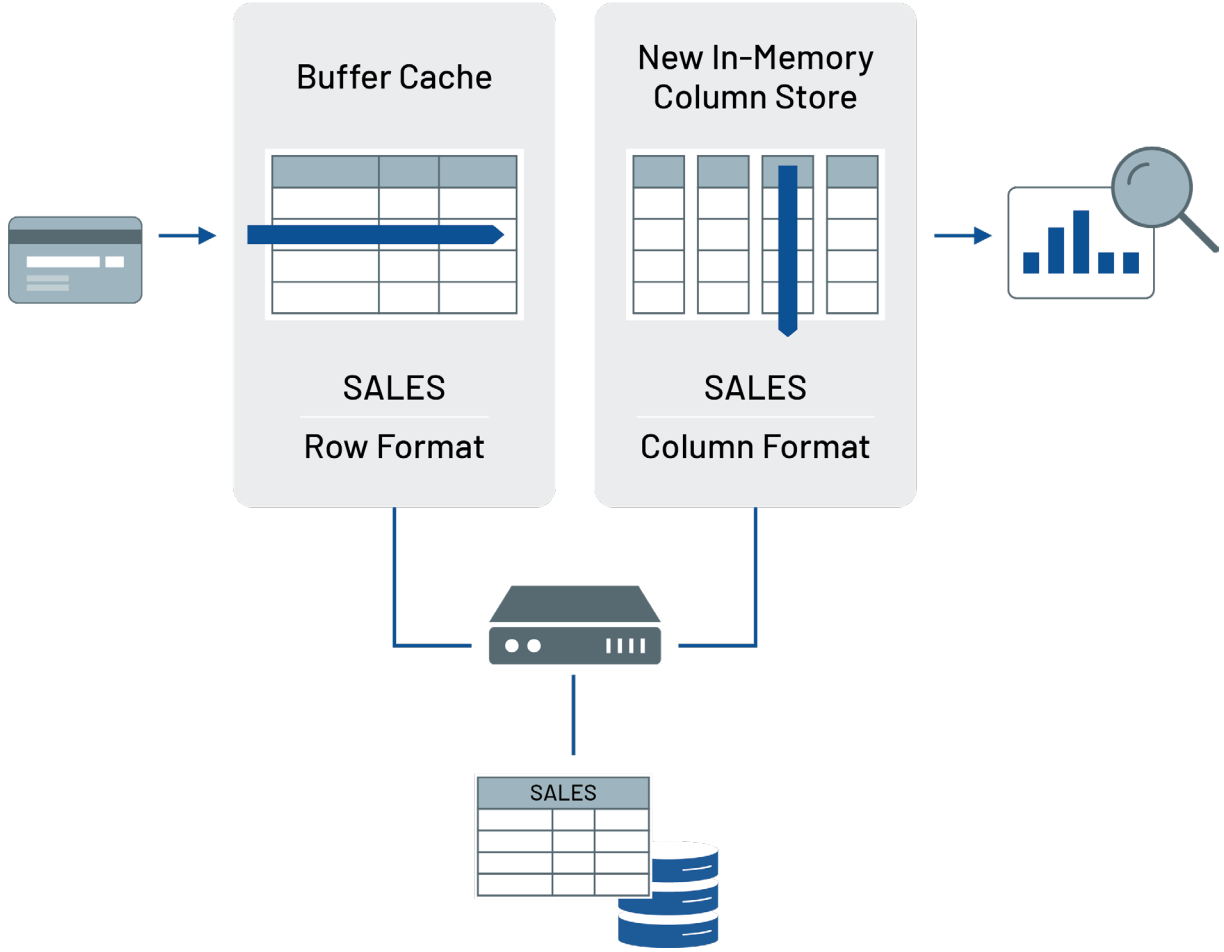
Agenda

- What is Oracle In-Memory?
- Release History & Licensing
- Configuration, Loading Tables In-Memory
- In-Memory Advisor
- Example Queries and Explain Plans
- Final Thoughts

datAvail

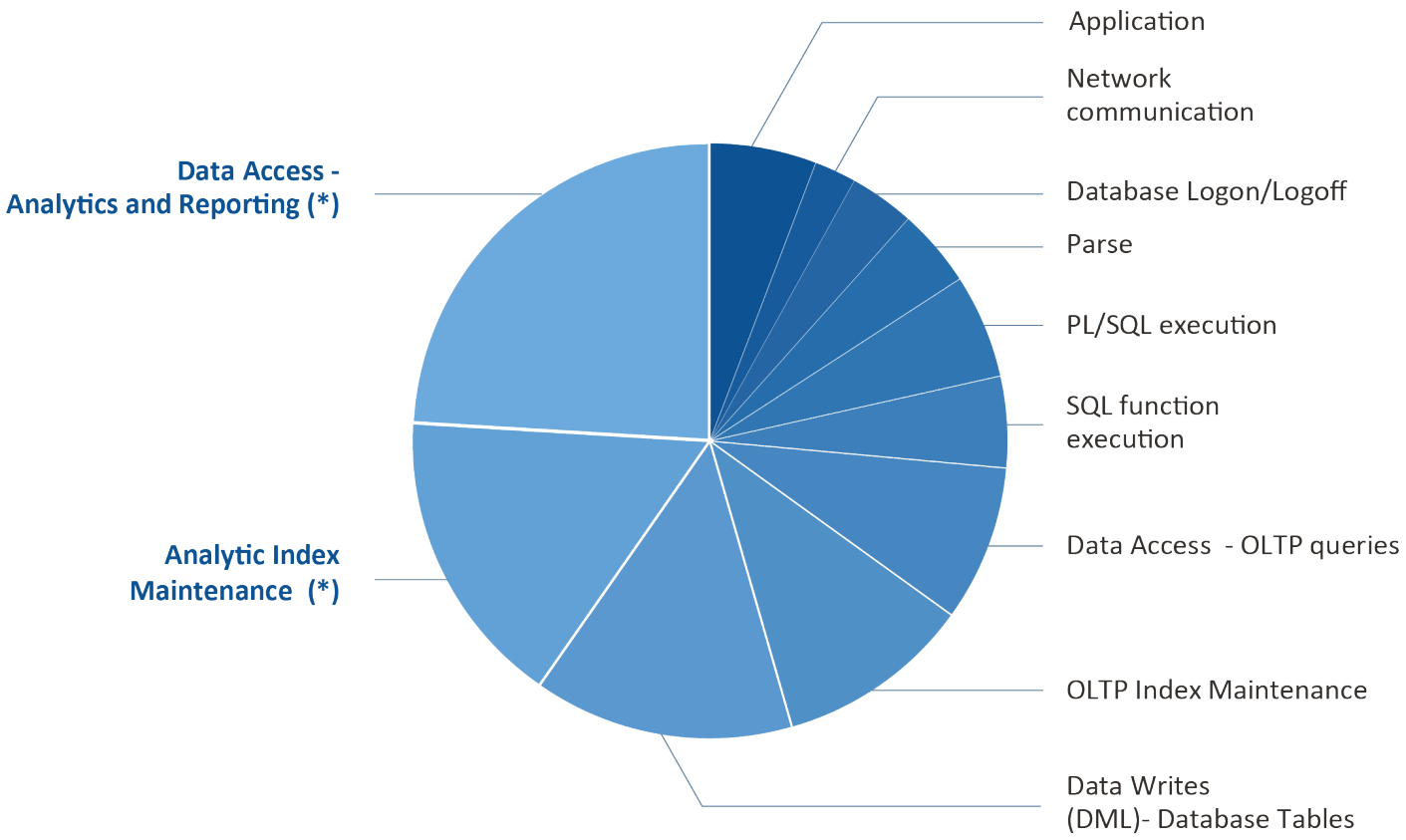
What is Oracle In-Memory?

- Uses part of your SGA memory for a compressed column store
- Primarily for Analytical Queries
- Simple to use, no application changes required
- Oracle Optimizer will choose the best path for your SQL query



What Query Type Benefits Best?

- Analytics and Reporting
- Analytic Index Maintenance



https://www.oracle.com/technetwork/database/in-memory/overview/twp-dbim-usage-2441076.html

In-Memory Options by Release

datAvail

12.1

- Columnar format in memory
- Vector SIMD instructions, compression, In-Memory Storage Indexing

18c

- Automatic In-Memory

12.2

- In Memory for Active Data Guard
- Heat Maps, controlled by configurable policies
- Bloom Filters, Join groups
- IME – In Memory Expressions
- Exadata Flash

19c

- External Tables
- Resource Manager enabled for Databases In-Memory

Licensing Requirements

Please confirm with your Oracle Sales Representative

Documented Here

https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dblic/Licensing-Information.html

- Additional cost for on-prem EE or EE-EX
- Not Available for SE2
- Included OCI DBCS EE-EP/ExaCS
- Trial setup allowed

Useful In-Memory Views

datAvail

Initialization Parameter	Description
DBA_FEATURE_USAGE_STATISTICS	Displays information about database feature usage statistics. When the IM column store is accessed, the NAME column shows In-Memory Column Store.
DBA_SEGMENTS	Describes the storage allocated for all segments in the database. Several columns, including INMEMORY and INMEMORY_PRIORITY, describe In-Memory attributes of the segment.
DBA_TABLES	Indicates which tables have the INMEMORY attribute set (the INMEMORY column is ENABLED) or not set (DISABLED).
V$ACTIVE_SESSION_HISTORY	Displays sampled session activity. Several columns, including INMEMORY_QUERY and INMEMORY_POPULATE, describe session activity relating to the In-Memory Column Store at the time of sampling.
V$IM_SEGMENTS	Presents information about all In-Memory segments in the database. Only segments that have an In-Memory representation are displayed. If a segment is marked for the IM column store but is not populated, the view does not contain a corresponding row for this segment.
V$INMEMORY_AREA	Displays information about the space allocation inside the In-Memory Area.
V$SGA	Displays the size of the In-Memory Area.

In-Memory Configuration

- Simply set inmemory_size
- Check system memory
- Adjust SGA_TARGET or MEMORY_TARGET as needed

```
SQL> alter system set sga_target=4g scope=spfile;

System altered.

SQL> alter system set inmemory_size=2g scope=spfile;

System altered.
```

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 4294963264 bytes
Fixed Size                  8904768 bytes
Variable Size             486539264 bytes
Database Buffers         1644167168 bytes
Redo Buffers                7868416 bytes
In-Memory Area           2147483648 bytes
Database mounted.
Database opened.
SQL> select name,value from v$sga where NAME='In-Memory Area';

NAME                         VALUE
-------------------- ----------
In-Memory Area       2147483648
```

OEM In-Memory View

datAvail

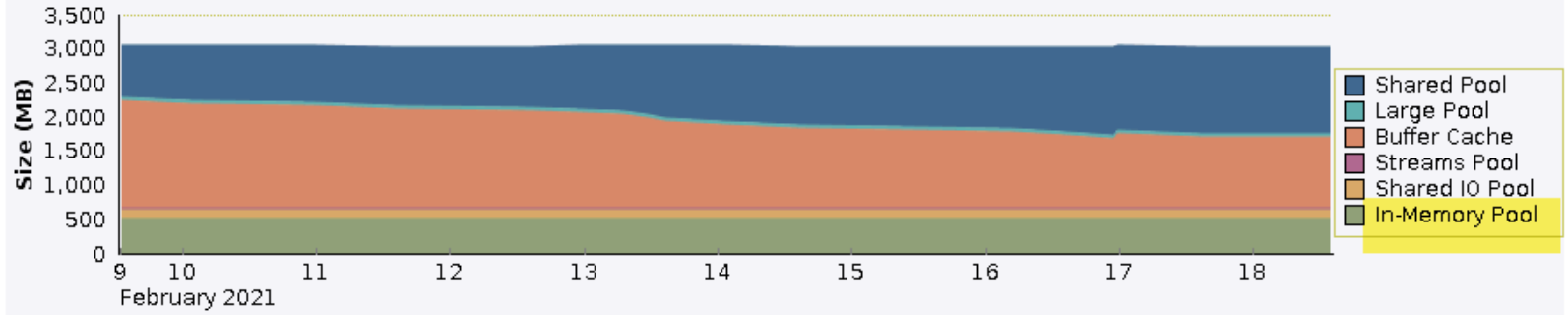
Memory Advisors

SGA | PGA

The System Global Area (SGA) is a group of shared memory structures that contains data and control information for one Oracle database. The SGA is allocated in memory when an Oracle da

Allocation History

This chart shows the history of the components of the SGA.



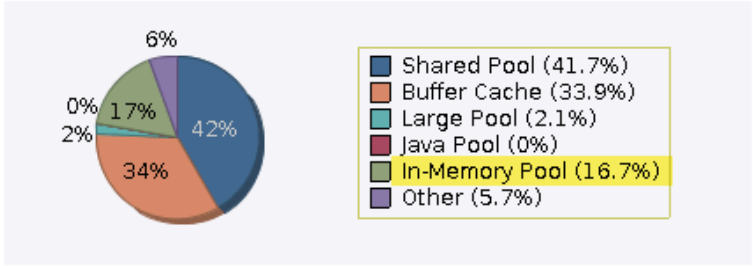
Current Allocation

! Automatic Shared Memory Management does not apply to the In-Memory pool.

Automatic Shared Memory Management Enabled Disable

Total SGA Size (MB) 3072 Advice

SGA Component	Current Allocation (MB)
Shared Pool	1280
Buffer Cache	1040
Large Pool	64
Java Pool	0
In-Memory Pool	512
Other	176



Checking the In-Memory Pool

datAvail

```
SQL>  select OWNER, SEGMENT_NAME, POPULATE_STATUS, bytes_not_populated from v$im_segments;

no rows selected

SQL>  select POOL, ALLOC_BYTES/1024/1024 ALLOC_MB, USED_BYTES/1024/1024 USED_MB, POPULATE_STATUS from V$INMEMORY_AREA;

POOL                        ALLOC_MB    USED_MB POPULATE_STATUS
--------------------------- ---------- ---------- -------------------------
1MB POOL                          1423          0 DONE
64KB POOL                          608          0 DONE

2 rows selected.
```

Loading Tables into Memory

Data set and query examples are from OOW Database In-Memory Hands-On Lab

```
SQL> alter table ssb.date_dim inmemory;

Table altered.

SQL> alter table SSB.SUPPLIER inmemory;

Table altered.

SQL> alter table SSB.CUSTOMER inmemory;

Table altered.

SQL> alter table SSB.PART inmemory;

Table altered.


SQL>  select count(*) from SSB.CUSTOMER inmemory;

  COUNT(*)
----------
     60000

SQL>  select count(*) from SSB.LINEORDER;

  COUNT(*)
----------
  11997996
```

Loading Tables into Memory

```
SQL>  select POOL, ALLOC_BYTES/1024/1024 ALLOC_MB, USED_BYTES/1024/1024 USED_MB, POPULATE_STATUS from V$INMEMORY_AREA;

POOL                        ALLOC_MB    USED_MB POPULATE_STATUS
--------------------------- ---------- ---------- -------------------------
1MB POOL                          1423        569 DONE
64KB POOL                          608       5.25 DONE

SQL> select OWNER, SEGMENT_NAME, POPULATE_STATUS, bytes_not_populated from v$im_segments;

OWNER      SEGMENT_NAME               POPULATE_STAT BYTES_NOT_POPULATED
---------- -------------------------- ------------- -------------------
SSB        DATE_DIM                   COMPLETED                       0
SSB        CUSTOMER                   COMPLETED                       0
SSB        LINEORDER                  COMPLETED                       0
```

Or simply load all objects in a tablespace

```
SQL>  ALTER TABLESPACE  TS_DATA  default INMEMORY PRIORITY CRITICAL DISTRIBUTE AUTO FOR SERVICE DEFAULT DUPLICATE ALL;

Tablespace altered.
```

Compression Example

datAvail

```
SQL> SELECT v.owner,
  2         v.segment_name            name,
  3         v.bytes                   on_disk_size,
  4         v.inmemory_size           in_mem_size,
  5         ROUND(v.bytes / v.inmemory_size, 2) comp_ratio
  6  FROM   v$im_segments v
  7  ORDER BY 4;

OWNER                NAME                           ON_DISK_SIZE          IN_MEM_SIZE COMP_RATIO
-------------------- ------------------------------ -------------------- -------------------- ----------
SSB                  SUPPLIER                              393,216            1,310,720         .3
SSB                  DATE_DIM                              131,072            1,310,720         .1
SSB                  CUSTOMER                            5,586,944            5,505,024       1.01
SSB                  PART                               32,292,864            8,650,752       3.73
SSB                  LINEORDER                         817,586,176          595,329,024       1.37
```

Other Interesting In-Memory Options

datAvail

Active Data Guard In-Memory

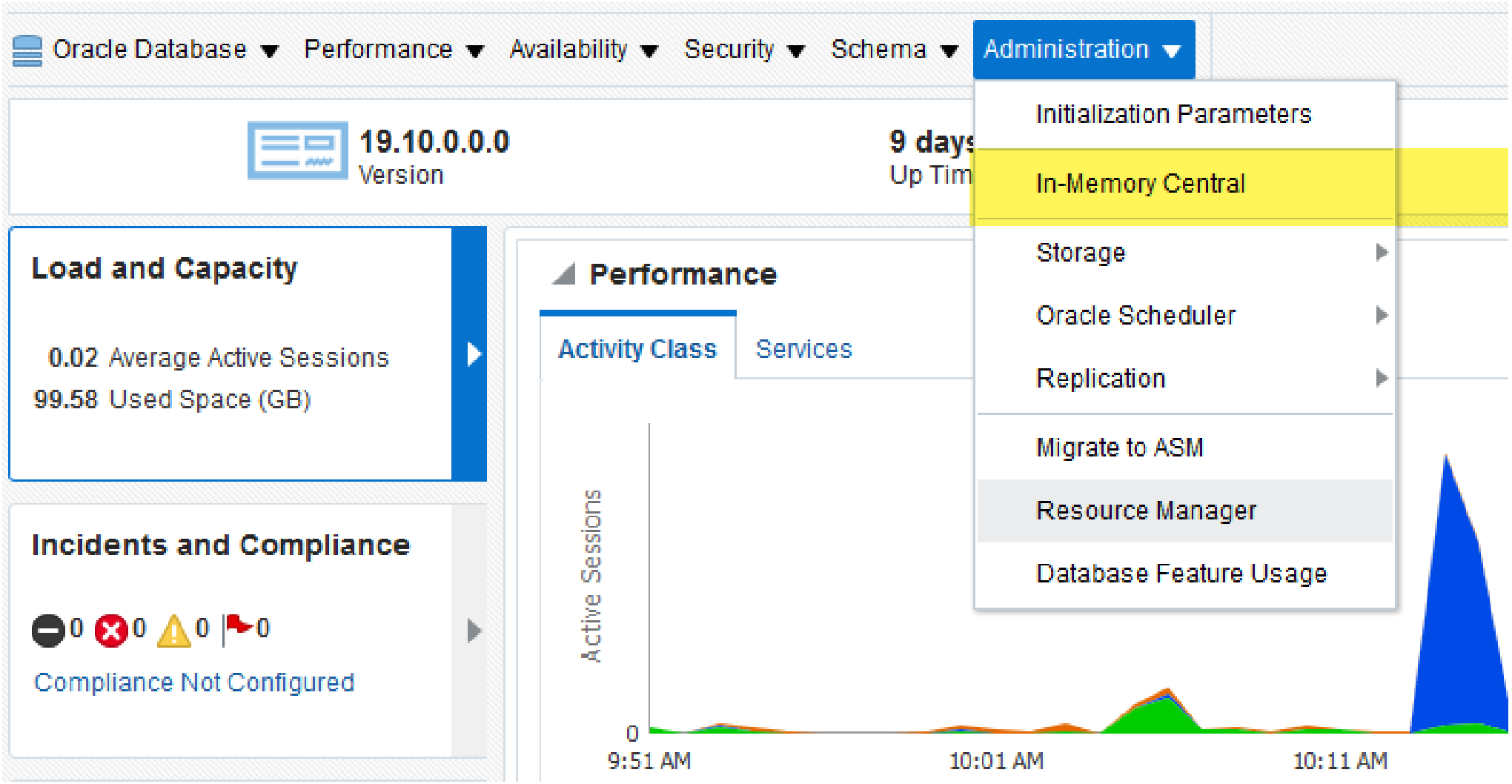
- In-Memory can be configured separately
- ALTER TABLE … INMEMORY FOR SERVICE …

RAC In-Memory

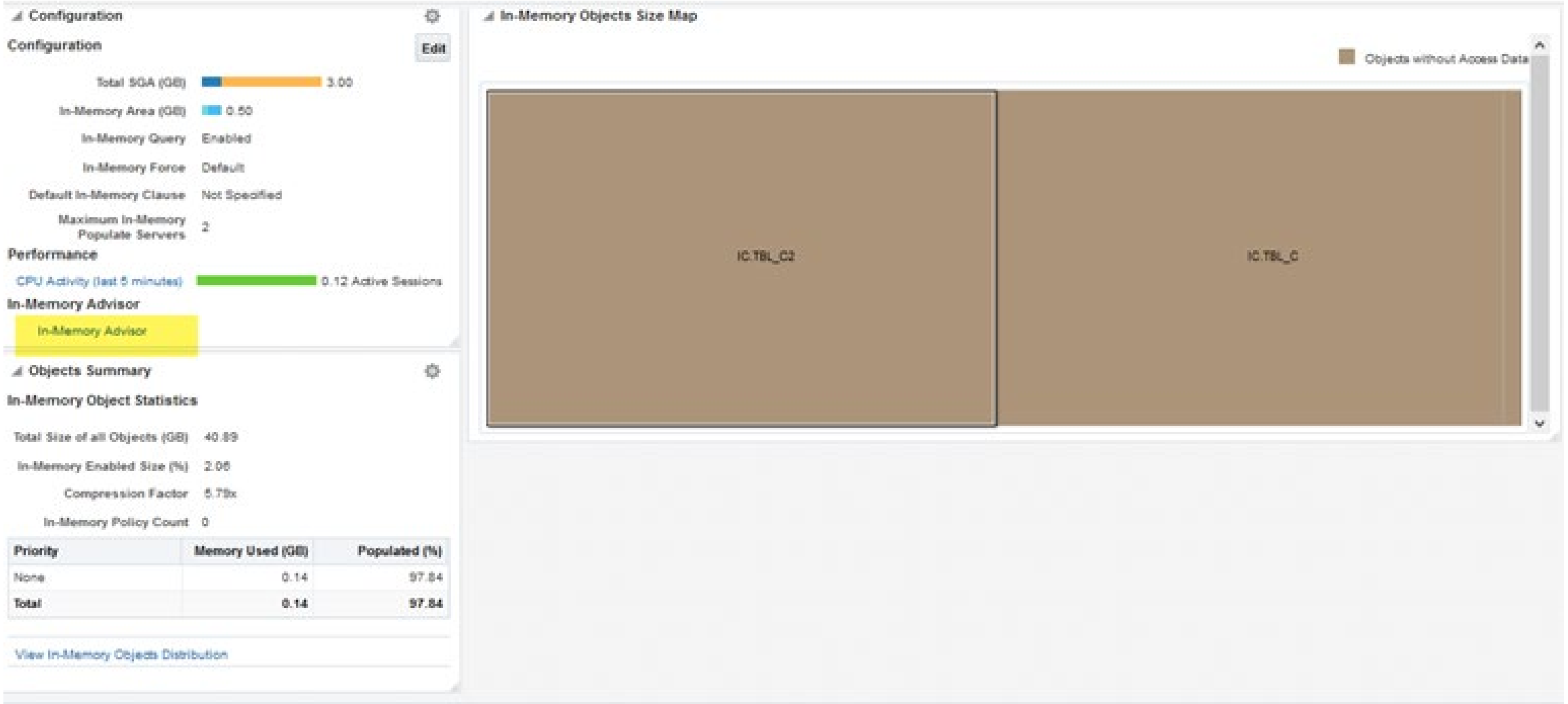
- Each Node has own column store
- INMEMORY_DISTRIBUTE column of in-memory views

In-Memory Advisor - 1965343.1

datAvail



In-Memory Advisor - 1965343.1



In-Memory Advisor - 1965343.1

datAvail

In-Memory Objects Search

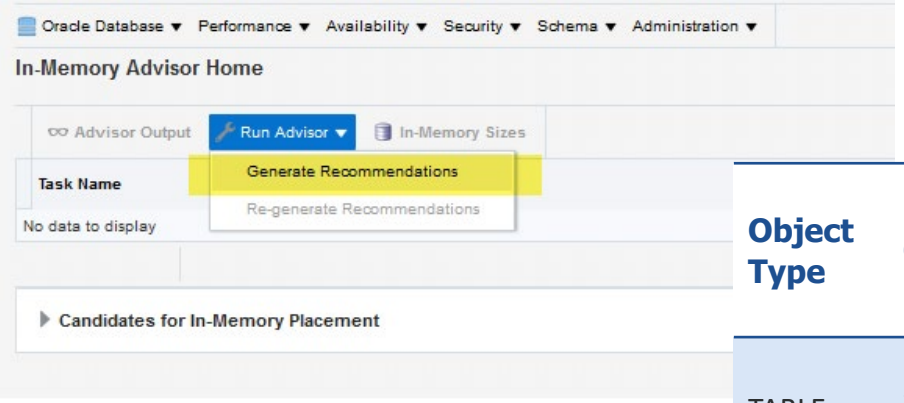
Search In-Memory Objects Display segments if not all partitions are in-memory enabled

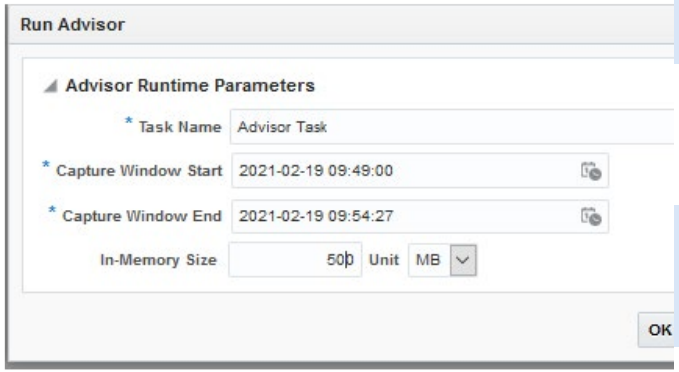
Filters

Edit In-Memory Options View Segment Details Repopulate

Name	Type	Partitioned	Last Scan Date	In-Memory Size (GB)	Populated (%)	Compression Factor	In-Memory Parameters				Size on Storage (GB)	Storage Compression
							Priority	Compression	Distribution	Duplicate		
IC.TBL_C	Table	No	NA	0.07	100	5.78	NONE	FOR QUERY LOW	AUTO	NO DUPLICATE	0.4041	None
IC.TBL_C2	Table	No	NA	0.07	100	5.78	NONE	FOR QUERY LOW	AUTO	NO DUPLICATE	0.4041	None
IC.TAB_P	Table	Yes	NA	0.002	100	6.29					0.0154	None
Total				0.142							0.8236	

In-Memory Advisor - 1965343.1





Object Type	Object Name	Compression Type	Estimated In-Memory Size	Analytics processing seconds	Estimated reduced processing	Estimated Performance Factor	Benefit: Cost Ratio
TABLE	schema.table_name1	No memory compression	1.471MB	22	20	10.0X	128370 : 1
TABLE	schema.table_name2	No memory compression	1.063MB	10	9	10.0X	80782 : 1
TABLE	schema.table_name3	No memory compression	1.063MB	10	9	10.0X	80782 : 1
TABLE	schema.table_name4	No memory compression	3.109MB	21	19	10.0X	57974 : 1
TABLE	schema.table_name4	No memory compression	3.431MB	20	18	10.0X	50025 : 1

Query Examples

datAvail

```
SQL> select
  2    max(lo_ordtotalprice) most_expensive_order,
  3    sum(lo_quantity) total_items
  4  from
  5    LINEORDER;

MOST_EXPENSIVE_ORDER TOTAL_ITEMS
-------------------- -----------
            55528516   305976330
```

```
-------------------------------------------------------------------------------------
| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |           |       |       | 24378 (100)|          |
|   1 |  SORT AGGREGATE    |           |     1 |     9 |            |          |
|   2 |   TABLE ACCESS FULL| LINEORDER |    11M|   102M| 24378   (1)| 00:00:01 |
-------------------------------------------------------------------------------------
```

In-Memory Explain Plan

```
-----------------------------------------------------------------------------------------
| Id  | Operation                   | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |           |       |       |   993 (100)|          |
|   1 |  SORT AGGREGATE             |           |     1 |     9 |            |          |
|   2 |   TABLE ACCESS INMEMORY FULL| LINEORDER |    11M|   102M|   993   (9)| 00:00:01 |
-----------------------------------------------------------------------------------------
```

- 1.02 sec vs .03 sec In-Memory

Query 2 – Normal Plan

datAvail

```
SQL> select     lo_orderkey, lo_custkey, lo_revenue
  2  from       LINEORDER
  3  where      lo_orderkey = 5000000;

         LO_ORDERKEY          LO_CUSTKEY          LO_REVENUE
-------------------- -------------------- --------------------
             5000000                 9730              1634013
```

```
-------------------------------------------------------------------------------
| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |           |       |       | 24333 (100)|          |
|*  1 |  TABLE ACCESS FULL| LINEORDER |     4 |    68 | 24333   (1)| 00:00:01 |
-------------------------------------------------------------------------------

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

   1 - filter("LO_ORDERKEY"=5000000)
```

Query 2 – In-Memory Plan

datAvail

```
-------------------------------------------------------------------------------------
| Id  | Operation                   | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |           |       |       |   959 (100)|          |
|*  1 |  TABLE ACCESS INMEMORY FULL| LINEORDER |     4 |    68 |   959   (6)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   1 - inmemory("LO_ORDERKEY"=5000000)
       filter("LO_ORDERKEY"=5000000)
```

- .61 vs .02
- HOL has example with Index

Query 3 – Normal Plan

datAvail

```
SQL> select
  2    sum(lo_extendedprice * lo_discount) revenue
  3  from
  4    LINEORDER l,
  5    DATE_DIM d
  6  where
  7    l.lo_orderdate = d.d_datekey
  8    and l.lo_discount between 2 and 3
  9    and l.lo_quantity < 24
 10    and d.d_date='December 24, 1996';

   REVENUE
----------
2049020339
```

```
-------------------------------------------------------------------------------
| Id  | Operation           | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |           |       |       | 24425 (100)|          |
|   1 |  SORT AGGREGATE     |           |     1 |    43 |            |          |
|*  2 |   HASH JOIN         |           |   417 | 17931 | 24425   (1)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL| DATE_DIM  |     1 |    25 |    15   (0)| 00:00:01 |
|*  4 |    TABLE ACCESS FULL| LINEORDER |  1003K|    17M| 24408   (1)| 00:00:01 |
-------------------------------------------------------------------------------

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

   2 - access("L"."LO_ORDERDATE"="D"."D_DATEKEY")
   3 - filter("D"."D_DATE"='December 24, 1996')
   4 - filter(("L"."LO_DISCOUNT"<=3 AND "L"."LO_QUANTITY"<24 AND
              "L"."LO_DISCOUNT">=2))
```

Query 3 – In-Memory Plan

datAvail

```
-------------------------------------------------------------------------------------------
| Id  | Operation                     | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |           |       |       |  1089 (100)|          |
|   1 |  SORT AGGREGATE               |           |     1 |    43 |            |          |
|*  2 |   HASH JOIN                   |           |   417 | 17931 |  1089  (17)| 00:00:01 |
|   3 |    JOIN FILTER CREATE         | :BF0000   |     1 |    25 |     1   (0)| 00:00:01 |
|*  4 |     TABLE ACCESS INMEMORY FULL| DATE_DIM  |     1 |    25 |     1   (0)| 00:00:01 |
|   5 |    JOIN FILTER USE            | :BF0000   |  1003K|    17M|  1085  (17)| 00:00:01 |
|*  6 |     TABLE ACCESS INMEMORY FULL| LINEORDER |  1003K|    17M|  1085  (17)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   2 - access("L"."LO_ORDERDATE"="D"."D_DATEKEY")
   4 - inmemory("D"."D_DATE"='December 24, 1996')
       filter("D"."D_DATE"='December 24, 1996')
   6 - inmemory(("L"."LO_DISCOUNT"<=3 AND "L"."LO_QUANTITY"<24 AND
              "L"."LO_DISCOUNT">=2 AND SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE")))
       filter(("L"."LO_DISCOUNT"<=3 AND "L"."LO_QUANTITY"<24 AND
              "L"."LO_DISCOUNT">=2 AND SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE")))
```

- .88 vs .02 sec
- Bloom filter example

Final Thoughts

- Easy setup
- No application changes
- Most useful for analytical queries, but will also help other operations
- Allows removal of some indexing
- Single database can now support mixed workload

datAvail

Q&A

Fill Out the Session Evaluation Form to Enter to Win Free Beer For a Year!

Thank You

Anuj Pandey

Oracle Database Administrator

anuj.pandey@datavail.com

517-512-5097

datAvail