

ORACLE IN-MEMORY 12c

Vantagens e Cenários de Utilização
do Oracle In-Memory 12c

Willian Frasson

Apresentação

Willian Frasson – DBA Oracle

- Tecnologia em Processamento de dados pela Universidade de Maringá – Cesumar (2005)
- Especialização em Banco de Dados pela Universidade Norte do Paraná – Unopar (2007)
- Pós Graduação em Gestão de Projetos com Ênfase em TI – PUCRS (2016)
- Oracle Database 12c Administrator Certified Professional (OCP 12c)
- 11 anos de experiência trabalhando com Oracle com as versões 9i, 10g, 11g e 12c
- Experiência em ambiente de alta disponibilidade, como Core Banking, Sistema de Cartões e Internet Banking em RAC metropolitano

Conteúdo

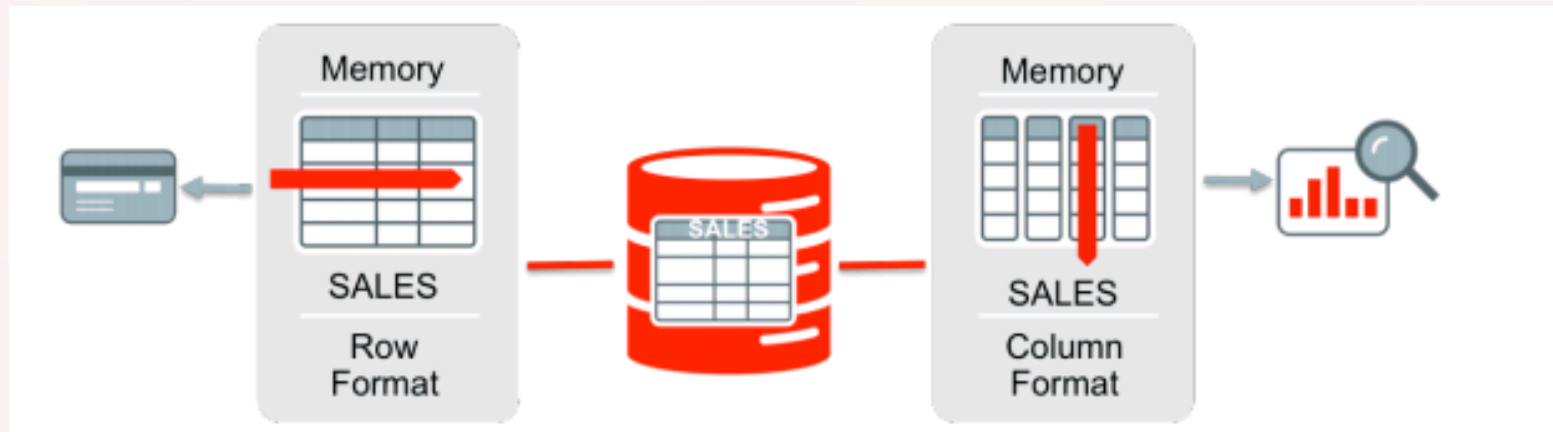
- Conceito do Oracle In-Memory 12c
- Características
- Níveis de Prioridades
- Níveis de Compressão
- Restrições
- Métodos de Agregação
- Armazenamento de Objetos em RAC
- Parâmetros
- Principais Comandos
- Views
- Cenários
- New Features 12.2

Oracle In-Memory 12c

- É uma feature da versão 12c que acelera consultas analíticas por ordens de magnitude, permitindo que as decisões de negócios sejam em tempo real.

Características do Oracle In-Memory Database 12c

- Dados colunares em memória
- Partições de tabelas em memória
- Tablespaces em memória
- Compressão de dados em memória



Níveis de Prioridades de Objetos

É possível ter os seguintes níveis de prioridades:

Priority Level	Description
CRITICAL	Object is populated immediately after the database is opened
HIGH	Object is populated after all CRITICAL objects have been populated, if space remains available in the IM column store
MEDIUM	Object is populated after all CRITICAL and HIGH objects have been populated, and space remains available in the IM column store
LOW	Object is populated after all CRITICAL , HIGH , and MEDIUM objects have been populated, if space remains available in the IM column store
NONE	Objects only populated after they are scanned for the first time (Default), if space is available in the IM column store

Figure 7. Different priority levels controlled by the **PRIORITY** sub clause of the **INMEMORY** clause

Níveis de Compressão

É possível ter os seguintes níveis de compressão de objeto:

Compression Level	Description
<code>NO MEMCOMPRESS</code>	Data is populated without any compression
<code>MEMCOMPRESS FOR DML</code>	Minimal compression optimized for DML performance
<code>MEMCOMPRESS FOR QUERY LOW</code>	Optimized for query performance (default)
<code>MEMCOMPRESS FOR QUERY HIGH</code>	Optimized for query performance as well as space saving
<code>MEMCOMPRESS FOR CAPACITY LOW</code>	Balanced with a greater bias towards space saving
<code>MEMCOMPRESS FOR CAPACITY HIGH</code>	Optimized for space saving

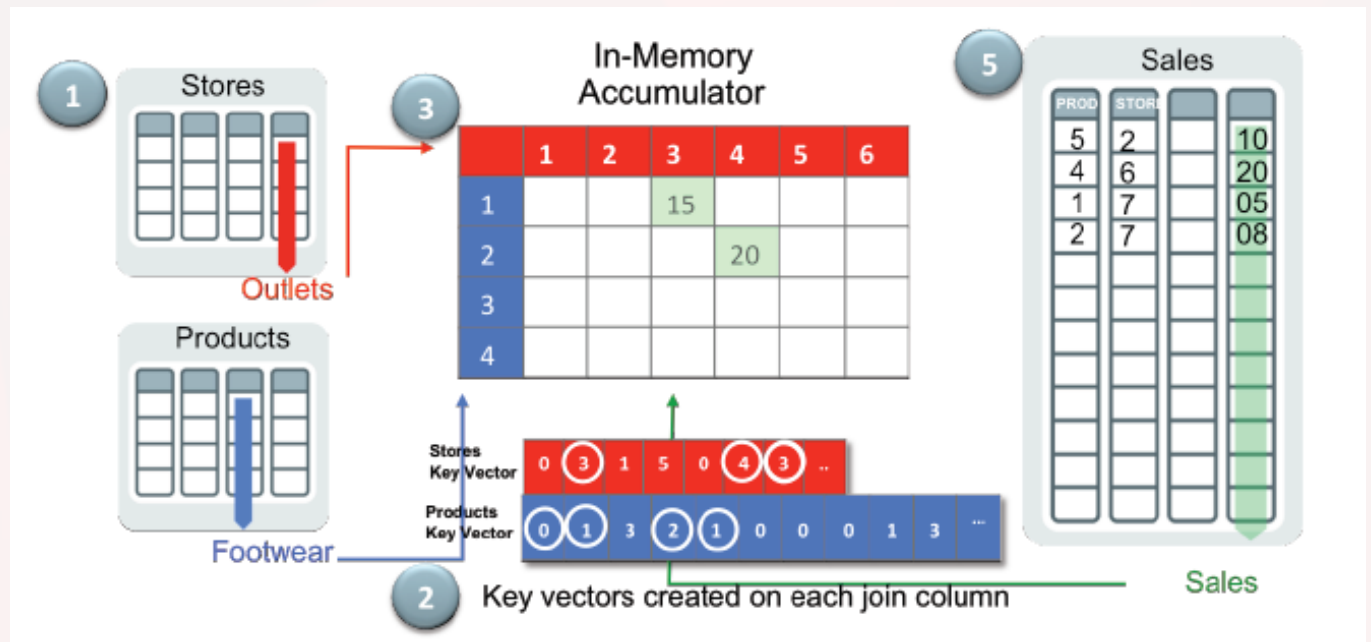
Restrições do Oracle In-Memory

Temos as seguintes limitações para a versão 12.1.0.2 do Oracle Database In-Memory:

- Qualquer objeto SYS ou SYSTEM
- Index Organized Table (IOT)
- Tabelas clusterizadas
- Colunas do tipo LONG ou BLOB
- Utilização em Active Data Guard

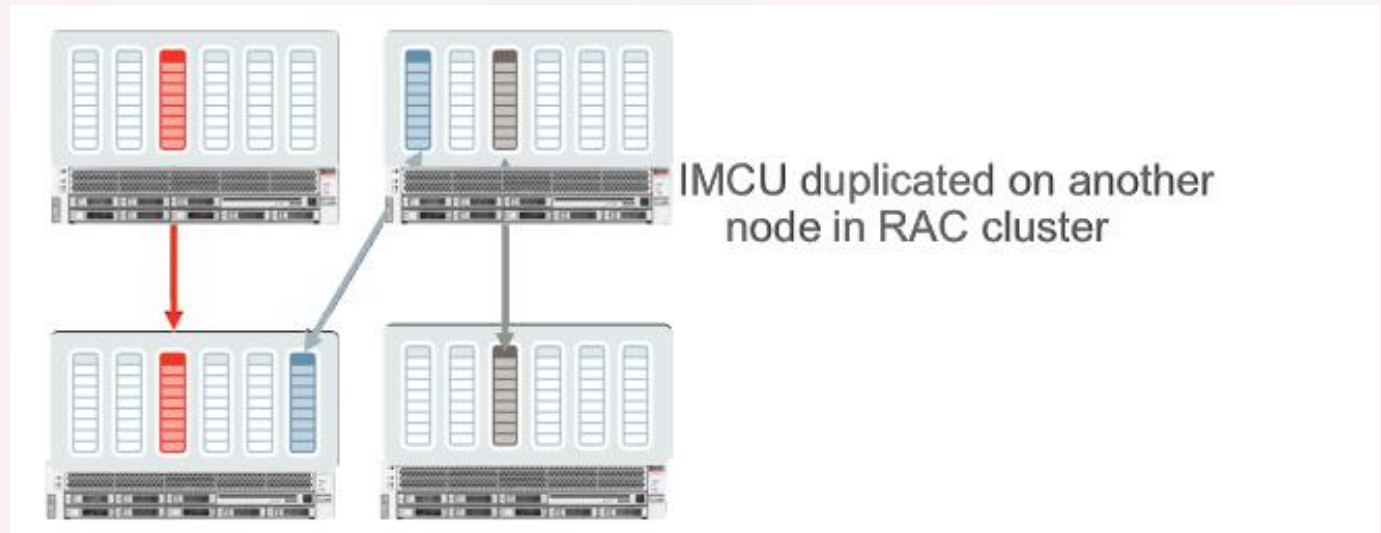
Método de Agregação de Consultas

A Oracle criou na versão 12.1.0.2 um novo método para agregação nas tabelas In-Memory chamado **Vetor Group By**. Esse método consiste em otimizar as consultas analíticas a fim de trazer os resultados em segundos.



Armazenamento de Colunas In-Memory em RAC

- Cada instância de banco de dados possui sua área separada para In-Memory, sendo possível distribuir o objeto, tabela, coluna em várias instâncias.
- Os dados podem também ser duplicados para outros nós do Cluster conforme abaixo.



Parâmetros do In-Memory 12c

- **INMEMORY_SIZE**

Parâmetro que controla a quantidade de memória alocada para o In memory

- **INMEMORY_QUERY**

Parâmetro que controla se esta ativo o In Memory a nível de sistema ou sessão

- **INMEMORY_MAX_POPULATE_SERVERS**

Número máximo de processos que irão popular os dados em memória. O default é a o parâmetro `cpu_count` x 0.5

- **OPTIMIZER_INMEMORY_AWARE**

Parâmetro para ativação ou desativação de aprimoramentos do otimizador para o In-Memory

Parâmetros do In-Memory 12c

- **INMEMORY_CLAUSE_DEFAULT**

Parâmetro que permite especificar uma cláusula padrão para novas tabelas e mviews (Ex: inmemory, no inmemory, memcompress)

- **INMEMORY_FORCE**

Parâmetro que controla se tabelas que estão com flag in-memory serão populadas na memória (default, off).

- **INMEMORY_TRICKLE_REPOPULATE_SERVERS_PERCENT**

Parâmetro que controla o percentual de repovoamento das tabelas in-memory. O percentual é o valor definido no INMEMORY_MAX_POPULATE_SERVERS.

Por exemplo se o valor de INMEMORY_MAX_POPULATE_SERVERS for definido 10 e o mesmo 1 (default), será utilizado 10% das cpus.

Hidden Parameters

`_inmemory_pct_inv_blocks_invalidate_imcu 100`

Percentual de blocos inválidos na memória para invalidar IMCU(In-Memory Compression Unity)

`_inmemory_pct_inv_rows_invalidate_imcu 50`

Percentual de linhas inválidos na memória para invalidar IMCU(In-Memory Compression Unity)

* Antes dos dados serem populados no In-Memory, ficam STALE, sendo que as entradas obsoletas serão recuperadas de disco ou buffer cache.

Principais Comandos

Adicionando tabelas:

```
alter table OWNER.TABELA inmemory priority critical memcompress for query high;
```

Adicionando partições:

```
alter table OWNER.TABELA inmemory priority critical memcompress for query high;
```

```
alter table OWNER.TABELA modify partition PART_201301 no inmemory;
```

Adicionando apenas algumas colunas:

```
alter table OWNER.TABELA inmemory priority critical no memcompress;
```

```
alter table OWNER.TABELA inmemory no inmemory(coluna);
```

Views In-Memory

Status in-memory:

```
select * from v$inmemory_area;
```

Pools:

1mb – Dados

64kb – Apenas metadados das tabelas in-memory

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS
1MB POOL	34355544064	16476274688	DONE
64KB POOL	8573157376	71368704	DONE

2 rows selected.

Elapsed: 00:00:00.01
willian_frasson@bidb12c> |

Views In-Memory

Status tabelas:

```
select table_name, inmemory, inmemory_priority,  
inmemory_distribute, inmemory_compression from dba_tables  
where inmemory = 'ENABLED' order by table_name;
```

TABLE_NAME	INMEMORY	INMEMORY	INMEMORY_DISTRI	INMEMORY_COMPRESS
AGR_CON_GRUPO_PESSOA	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
AGR_PNL_IN_PROD_PRODUTOS	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
BASE_ASSOCIADOS	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
BASE_ASSOCIADO_201409	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_CAR_CONTA_CARTAO	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_CAR_LOGO_FUNC	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_CAR_SITUACAO	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_COR_TEMPO_MES_MU	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_CPR_TEMPO	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_ENTIDADE	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_ENTIDADE_MEMORY	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_INDICADOR_AVALIACAO	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_PESSOA_CONTA	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_PNL_IN_HIE_PROD	ENABLED	CRITICAL	AUTO	FOR QUERY LOW
DIM_PNL_IN_TIPO_PESSOA	ENABLED	CRITICAL	AUTO	FOR QUERY LOW

Views In-Memory

Segmentos in-memory:

```
SELECT v.owner, v.segment_name,  
v.BYTES_NOT_POPULATED/1024/1024  
BYTES_NOT_POPULATED, v.bytes/1024/1024 size_mb,  
v.inmemory_size/1024/1024 in_mem_size_mb, v.bytes /  
v.inmemory_size comp_ratio FROM v$im_segments v  
order by bytes_not_populated desc;
```

OWNER	SEGMENT_NAME	BYTES_NOT_POPULATED	SIZE_MB	IN_MEM_SIZE_MB	COMP_RATIO
IN_OWNER	BASE_ASSOCIADOS	256,671875	38017	1913,875	19,8638887
IN_OWNER	FTO_IPP_ASSOCIADO	85,875	10176	2189,25	4,64816718
IN_OWNER	DIM_CAR_CONTA_CARTAO	40,453125	344	234,6875	1,46577896
IN_OWNER	FTO_PONTUACAO_INDICADOR	7,96875	120	49,3125	2,43346008
IN_OWNER	DIM_COR_TEMPO_MES_MU	0	,1875	1,125	,166666667
IN_OWNER	DIM_INDICADOR_AVALIACAO	0	,125	1,125	,111111111
IN_OWNER	SRV_PORTIFOLIO_CREDITO	0	29	14,25	2,03508772
WILLIAN FRASSON	MUW ENTIDADE MEMORY	0	,25	1,125	,222222222

Views In-Memory

Linhas e blocos stale:

select

OBJD,sum(TOTAL_ROWS),sum(INVALID_ROWS),sum(INVALID_BLOCKS),(sum(INVALID_ROWS)/sum(TOTAL_ROWS))*100
pct_invalid from V\$IM_SMU_HEAD group by OBJD order by 5
desc;

OBJD	SUM(TOTAL_ROWS)	SUM(INVALID_ROWS)	SUM(INVALID_BLOCKS)	PCT_INVALID
2900757	90117319	1000000	7778	1,10966461
2898544	17285789	122061	5357	,706134964
309773	46288261	0	0	0
2493593	11702845	0	0	0
2898540	17509119	0	0	0
2872463	44157417	0	0	0
2493714	47505368	0	0	0
2878058	44008022	0	0	0
2898539	13632165	0	0	0
309776	9390	0	0	0
2599951	76441605	0	0	0
2676303	54797903	0	0	0

12 rows selected.

Teste de Stress 1 (Inserts) Com In-Memory - Sem Índices

O teste de stress foi executado da seguinte forma:

- Número de sessões: 50 sessões simultâneas
- Size da tabela: 10gb
- Número de de linhas: 83.696.934
- Tempo 03 minutos
- Tabelas sem índices
- Inmemory sem compressão
- 3.258.711 transações
- Média de 65.174 transações

Teste de Stress 2 (Inserts)

Sem In-Memory - 10 Índices

O teste de stress foi executado da seguinte forma:

- Número de sessões: 50 sessões simultâneas
- Size da tabela: 10gb
- Número de de linhas: 83.696.934
- Tempo 03 minutos
- Tabela com 10 índices
- Sem Inmemory
- 2.681.061 transações
- Média de 53.621 transações

Teste de Stress 3 (Inserts) Com In-Memory - 10 Índices

O teste de stress foi executado da seguinte forma:

- Número de sessões: 50 sessões simultâneas
- Size da tabela: 10gb
- Número de de linhas: 83.696.934
- Tempo 03 minutos
- Tabela com 10 índices
- Com Inmemory
- 2.629.607 transações
- Média de 52.992 transações

Teste de Stress 4 (Inserts) Com In-Memory - 15 Índices

O teste de stress foi executado da seguinte forma:

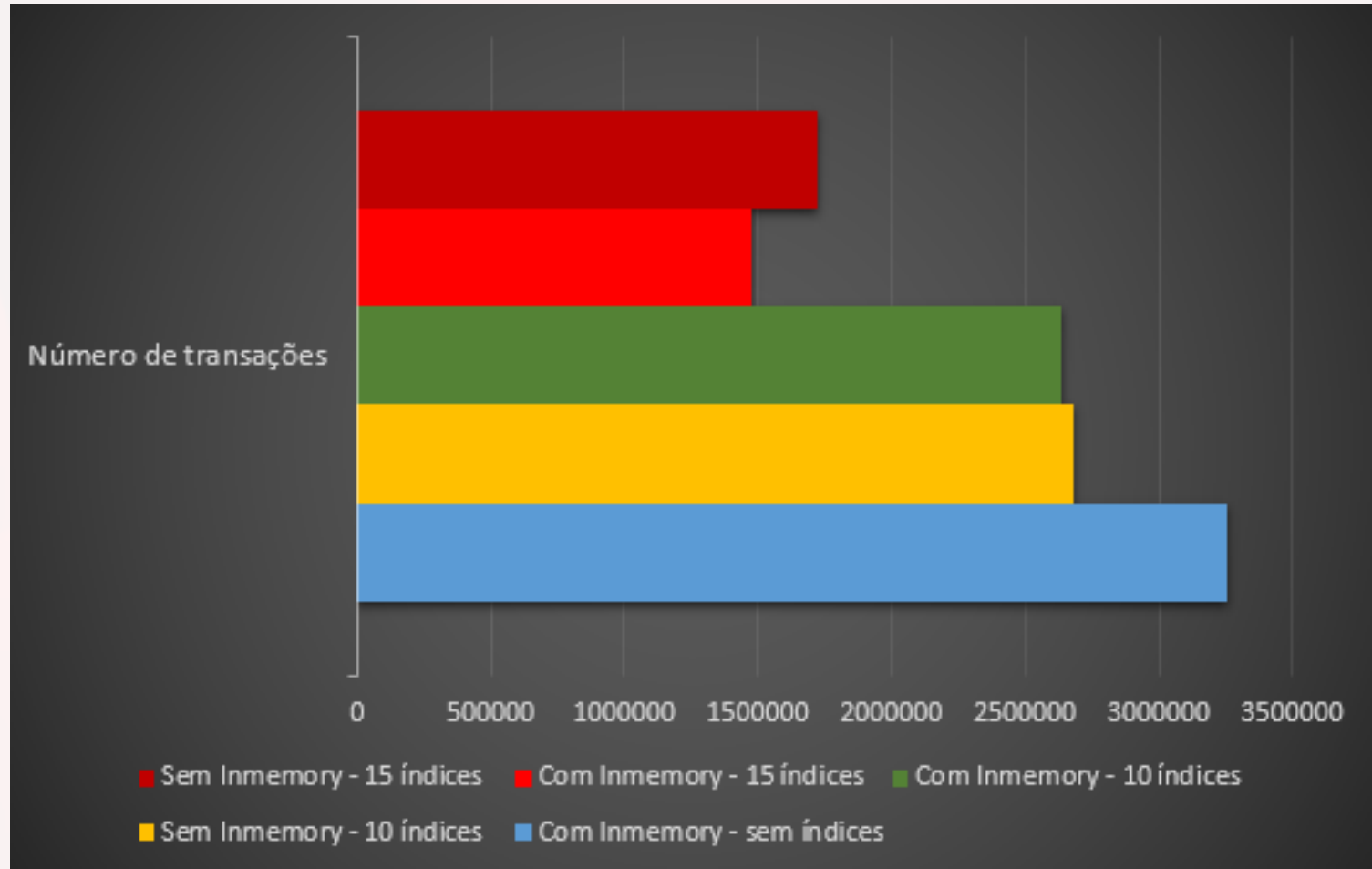
- Número de sessões: 50 sessões simultâneas
- Size da tabela: 10gb
- Número de de linhas: 83.696.934
- Tempo 03 minutos
- Tabela com 15 índices
- Com Inmemory
- 1.477.842 transações
- Média de 29.556 transações

Teste de Stress 5 (Inserts) Sem In-Memory - 15 Índices

O teste de stress foi executado da seguinte forma:

- Número de sessões: 50 sessões simultâneas
- Size da tabela: 10gb
- Número de de linhas: 83.696.934
- Tempo 03 minutos
- Tabela com 15 índices
- Sem Inmemory
- 1.723.444 transações
- Média de 34.468 transações

Número de Transações



Comandos Individuais

Planos de execução antes e depois da utilização do Oracle In-Memory:

Tempo de execução sem inmemory: **257 segundos**

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		304K	46M		265K (1)	00:01:04
1	SORT ORDER BY		304K	46M	57M	265K (1)	00:01:04
* 2	TABLE ACCESS BY INDEX ROWID	FTO_BFT_CADASTRO	304K	46M		260K (1)	00:01:03
* 3	INDEX RANGE SCAN	IDX_FTO_BFT_CADASTRO_6	4807K			7750 (1)	00:00:02

Tempo de execução com inmemory: **2 segundos**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT		3	1656	330 (22)	00:00:01		
1	SORT ORDER BY		3	1656	330 (22)	00:00:01		
2	PARTITION RANGE SINGLE		3	1656	329 (22)	00:00:01	15	15
* 3	TABLE ACCESS INMEMORY FULL	FTO_BFT_CADASTRO	3	1656	329 (22)	00:00:01	15	15

Comandos Individuais

Planos de execução antes e depois da utilização do Oracle In-Memory:

Tempo de execução sem inmemory: **1463 segundos**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1722	341K	293K (1)	00:01:11
1	SORT ORDER BY		1722	341K	293K (1)	00:01:11
2	HASH GROUP BY		1722	341K	293K (1)	00:01:11
3	NESTED LOOPS OUTER		1722	341K	293K (1)	00:01:11
* 4	TABLE ACCESS BY INDEX ROWID	FTO_BFT_CADASTRO	1559	267K	274K (1)	00:01:06
* 5	INDEX RANGE SCAN	IDX_FTO_BFT_CADASTRO_6	5066K		8168 (1)	00:00:02
* 6	TABLE ACCESS BY INDEX ROWID	FTO_BFT_CREDITO	1	27	48 (0)	00:00:01
* 7	INDEX RANGE SCAN	IDX_FTO_BFT_CREDITO_5	45		2 (0)	00:00:01

Tempo de execução com inmemory: **1,8 segundo**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT		1	613	405 (18)	00:00:01		
1	SORT ORDER BY		1	613	405 (18)	00:00:01		
2	HASH GROUP BY		1	613	405 (18)	00:00:01		
3	NESTED LOOPS OUTER		1	613	403 (18)	00:00:01		
4	PARTITION RANGE SINGLE		1	568	329 (22)	00:00:01	19	19
* 5	TABLE ACCESS INMEMORY FULL	FTO_BFT_CADASTRO	1	568	329 (22)	00:00:01	19	19
* 6	TABLE ACCESS BY GLOBAL INDEX ROWID BATCHED	FTO_BFT_CREDITO	1	45	74 (0)	00:00:01	19	19
* 7	INDEX RANGE SCAN	IDX_FTO_BFT_CREDITO_5	46		2 (0)	00:00:01		

New Features 12.2

- **Standby Database**

Com esta nova feature será possível utilizar o Oracle In-Memory em Standby

- **Resize Dinâmico**

Na versão 12.1 o valor do parâmetro Inmemorysize é estático, a partir da versão 12.2 será possível realizar o resize dinâmico

- **Fast Start**

Agora na versão 12.2 os dados são salvos de uma forma agiliza o carregamento dos dados em memória após um startup do banco de dados.

Perguntas?