

Oracle Database In-Memory By Example

Part 1:

```
SQL> @01_show_sga.sql
Connected.
SQL>
SQL> -- This command shows the SGA allocations
SQL>
SQL> show sga
```

```
Total System Global Area 4294967296 bytes
Fixed Size                 2932632 bytes
Variable Size              587202664 bytes
Database Buffers          2113929216 bytes
Redo Buffers              13844480 bytes
In-Memory Area            1577058304 bytes
```

```
SQL>
SQL> set echo off
SQL> @02_show_inmemory.sql
Connected.
SQL>
SQL> -- This command shows the settings of all in-memory init.ora parameters
SQL>
SQL> show parameter inmemory
```

| NAME | TYPE | VALUE |
|--------------------------------------|-------------|---------|
| inmemory_clause_default | string | |
| inmemory_force | string | DEFAULT |
| inmemory_max_populate_servers | integer | 1 |
| inmemory_query | string | ENABLE |
| inmemory_size | big integer | 1504M |
| inmemory_trickle_repopulate_servers_ | integer | 1 |
| percent | | |
| optimizer_inmemory_aware | boolean | TRUE |

```
SQL>
SQL> set echo off
SQL> @03_im_usage.sql
Connected.
SQL>
SQL> -- This query displays In-Memory Column Store memory allocations
SQL>
SQL> SELECT *
  2 FROM v$inmemory_area;
```

| POOL | ALLOC_BYTES | USED_BYTES | POPULATE_STATUS | CON_ID |
|-----------|-------------|------------|-----------------|--------|
| 1MB POOL | 1257242624 | 0 | DONE | 3 |
| 64KB POOL | 301989888 | 0 | DONE | 3 |

```
SQL>
SQL> set echo off
SQL> @04_desc.sql
Connected.
SQL>
SQL> -- This command displays the columns in the view v$IM_SEGMENTS
SQL>
SQL> desc v$IM_SEGMENTS
```

| Name | Null? | Type |
|----------------------|-------|---------------|
| OWNER | | VARCHAR2(128) |
| SEGMENT_NAME | | VARCHAR2(128) |
| PARTITION_NAME | | VARCHAR2(128) |
| SEGMENT_TYPE | | VARCHAR2(18) |
| TABLESPACE_NAME | | VARCHAR2(30) |
| INMEMORY_SIZE | | NUMBER |
| BYTES | | NUMBER |
| BYTES_NOT_POPULATED | | NUMBER |
| POPULATE_STATUS | | VARCHAR2(9) |
| INMEMORY_PRIORITY | | VARCHAR2(8) |
| INMEMORY_DISTRIBUTE | | VARCHAR2(15) |
| INMEMORY_DUPLICATE | | VARCHAR2(13) |
| INMEMORY_COMPRESSION | | VARCHAR2(17) |
| CON_ID | | NUMBER |

```
SQL>
SQL> set echo off
SQL> @05_im_segments.sql
Connected.
```

```
SQL>
SQL> -- This query displays what objects are in the In-Memory column store
SQL>
SQL> Select v.owner, v.segment_name name, v.populate_status status From v$im_segments v;
```

no rows selected

```
SQL>
SQL> set echo off
SQL> @06_im_alter_table.sql
Connected.
SQL>
SQL> -- This script alters the demo objects to be in-memory eligible
SQL>
SQL> alter table LINEORDER inmemory priority high;
```

Table altered.

```
SQL> alter table PART inmemory;
```

Table altered.

```
SQL> alter table CUSTOMER inmemory;
```

Table altered.

```
SQL> alter table SUPPLIER inmemory;
```

Table altered.

```
SQL> alter table DATE_DIM inmemory;
```

Table altered.

```
SQL>
SQL> set echo off
SQL> @07_im_attributes.sql
Connected.
SQL>
```

```
SQL> -- This query allows you to review the current attributes of the tables in SSB schemeam
```

```
SQL>
SQL> select table_name, cache, compression, compress_for, inmemory, inmemory_priority,
2         inmemory_distribute, inmemory_compression
3 from user_tables;
```

| TABLE_NAME | CACHE | DISK COMPRESSION | COMPRESS_FOR | INMEMORY | INMEMORY PRIORITY | INMEMORY DISTRIBUTE | INMEMORY COMPRESSION |
|------------|-------|---------------------|--------------|----------|----------------------|------------------------|-------------------------|
| PART | Y | ENABLED | BASIC | ENABLED | NONE | AUTO | FOR QUERY LOW |
| CUSTOMER | Y | ENABLED | BASIC | ENABLED | NONE | AUTO | FOR QUERY LOW |
| SUPPLIER | Y | ENABLED | BASIC | ENABLED | NONE | AUTO | FOR QUERY LOW |
| LINEORDER | Y | ENABLED | BASIC | ENABLED | HIGH | AUTO | FOR QUERY LOW |
| DATE_DIM | Y | ENABLED | BASIC | ENABLED | NONE | AUTO | FOR QUERY LOW |

```
SQL>
SQL> set echo off
SQL> @08_im_start_pop.sql
Connected.
SQL>
SQL> -- This script queries the demo objects to initiate population
SQL>
SQL> select /*+ full(LINEORDER) noparallel(LINEORDER) */ count(*) from LINEORDER;
```

```
COUNT(*)
-----
23996604
```

```
SQL> select /*+ full(PART) noparallel(PART) */ count(*) from PART;
```

```
COUNT(*)
-----
600000
```

```
SQL> select /*+ full(CUSTOMER) noparallel(CUSTOMER) */ count(*) from CUSTOMER;
```

```
COUNT(*)
-----
120000
```

```
SQL> select /*+ full(SUPPLIER) noparallel(SUPPLIER) */ count(*) from SUPPLIER;
```

COUNT(*)

8000

SQL> select /*+ full(DATE_DIM) noparallel(DATE_DIM) */ count(*) from DATE_DIM;

COUNT(*)

2556

SQL>
 SQL> set echo off
 SQL> @09_im_populated.sql
 Connected.
 SQL>
 SQL> -- Query the view v\$IM_SEGMENTS to shows what objects are in the column store
 SQL> -- and how much of the objects were populated. When the BYTES_NOT_POPULATED is 0
 SQL> -- it indicates the entire table was populated.
 SQL>
 SQL> SELECT v.owner, v.segment_name name, v.populate_status status, v.bytes_not_populated
 2 FROM v\$im_segments v;

| OWNER | NAME | STATUS | BYTES_NOT_POPULATED |
|-------|-----------|---------|---------------------|
| SSB | LINEORDER | STARTED | 1370537984 |

SQL>
 SQL> set echo off
 SQL> !top

top - 04:13:08 up 58 min, 2 users, load average: 1.05, 0.65, 0.43
 Tasks: 198 total, 2 running, 196 sleeping, 0 stopped, 0 zombie
 Cpu(s): 48.1%us, 3.0%sy, 0.0%ni, 46.7%id, 1.7%wa, 0.0%hi, 0.5%si, 0.0%st
 Mem: 5081108k total, 5044700k used, 36408k free, 1364k buffers
 Swap: 5095420k total, 94612k used, 5000808k free, 4498292k cached

| PID | USER | PR | NI | VRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|------|----------|----|-----|-------|------|------|---|------|------|---------|----------------|
| 4029 | oracle | 20 | 0 | 4662m | 1.2g | 1.0g | R | 96.5 | 24.0 | 2:07.28 | ora_w006_cdb1 |
| 2562 | oracle | -2 | 0 | 4435m | 8868 | 8716 | S | 4.7 | 0.2 | 4:47.60 | ora_vktm_cdb1 |
| 29 | root | 20 | 0 | 0 | 0 | 0 | S | 0.3 | 0.0 | 0:06.53 | kswapd0 |
| 1917 | haldaemo | 20 | 0 | 36764 | 1692 | 1172 | S | 0.3 | 0.0 | 0:00.22 | hald |
| 2578 | oracle | 20 | 0 | 4437m | 16m | 14m | S | 0.3 | 0.3 | 0:02.07 | ora_dia0_cdb1 |
| 2586 | oracle | 20 | 0 | 4435m | 15m | 15m | S | 0.3 | 0.3 | 0:00.91 | ora_lg00_cdb1 |
| 3362 | oracle | 20 | 0 | 289m | 6720 | 3216 | S | 0.3 | 0.1 | 0:03.08 | gnome-terminal |
| 4131 | root | 20 | 0 | 0 | 0 | 0 | S | 0.3 | 0.0 | 0:00.04 | kworker/0:0 |
| 1 | root | 20 | 0 | 19416 | 948 | 768 | S | 0.0 | 0.0 | 0:00.74 | init |
| 2 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kthreadd |
| 3 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:02.59 | ksoftirqd/0 |
| 5 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/0:0H |
| 6 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/u:0 |
| 7 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/u:0H |
| 8 | root | RT | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.08 | migration/0 |
| 9 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | rcu_bh |
| 10 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:01.56 | rcu_sched |
| 11 | root | RT | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.15 | watchdog/0 |
| 12 | root | RT | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.16 | watchdog/1 |
| 13 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:01.07 | ksoftirqd/1 |
| 14 | root | RT | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.02 | migration/1 |
| 16 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kworker/1:0H |
| 17 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | cpuset |
| 18 | root | 0 | -20 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | khelper |
| 19 | root | 20 | 0 | 0 | 0 | 0 | S | 0.0 | 0.0 | 0:00.00 | kdevtmpfs |

SQL> /

| OWNER | NAME | STATUS | BYTES_NOT_POPULATED |
|-------|-----------|---------|---------------------|
| SSB | LINEORDER | STARTED | 1203568640 |

SQL> /

| OWNER | NAME | STATUS | BYTES_NOT_POPULATED |
|-------|-----------|---------|---------------------|
| SSB | LINEORDER | STARTED | 936181760 |

SQL> /

| OWNER | NAME | STATUS | BYTES_NOT_POPULATED |
|-------|------|--------|---------------------|
|-------|------|--------|---------------------|

```
SSB LINEORDER STARTED 534118400
```

```
SQL> /
```

| OWNER | NAME | STATUS | BYTES_NOT_POPULATED |
|-------|-----------|---------|---------------------|
| SSB | LINEORDER | STARTED | 232783872 |

```
SQL> /
```

| OWNER | NAME | STATUS | BYTES_NOT_POPULATED |
|-------|-----------|-----------|---------------------|
| SSB | DATE_DIM | COMPLETED | 0 |
| SSB | PART | COMPLETED | 0 |
| SSB | SUPPLIER | COMPLETED | 0 |
| SSB | LINEORDER | COMPLETED | 0 |
| SSB | CUSTOMER | COMPLETED | 0 |

```
SQL> @i0_im_usage.sql
```

```
Connected.
```

```
SQL>
SQL> -- This query displays what objects are populated in the In-Memory Column Store
SQL>
```

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

| POOL | ALLOC_BYTES | USED_BYTES | POPULATE_STATUS | CON_ID |
|-----------|-------------|------------|-----------------|--------|
| 1MB POOL | 1257242624 | 1251999744 | DONE | 3 |
| 64KB POOL | 301989888 | 3604480 | DONE | 3 |

```
SQL>
SQL> set echo off
SQL> @i1_comp_ratio.sql
Connected.
SQL>
SQL> -- This query compares the actual size of the tables on disk to the size in the
SQL> -- In-Memory Column Store and calculates the resulting compression ratio
SQL>
```

```
SQL> SELECT v.owner,
2         v.segment_name name,
3         v.bytes orig_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 | ORIG_SIZE | IN_MEM_SIZE | COMP_RATIO |
|-------|-----------|------------|-------------|------------|
| SSB | DATE_DIM | 67108864 | 1179648 | 56.89 |
| SSB | SUPPLIER | 67108864 | 1179648 | 56.89 |
| SSB | CUSTOMER | 67108864 | 9568256 | 7.01 |
| SSB | PART | 67108864 | 13828096 | 4.85 |
| SSB | LINEORDER | 1476395008 | 1229848576 | 1.2 |

```
SQL>
SQL> set echo off
```

```
Part2:
```

```
SQL> @o1_im_query_stats.sql
```

```
Connected.
```

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

```
MOST_EXPENSIVE_ORDER
-----
55279127
```

```
Elapsed: 00:00:00.06
```

```
SQL>
SQL> set echo off
Hit enter ...
```

```
PLAN_TABLE_OUTPUT
```

```
-----
SQL_ID 7htp8zu9x0vq7, child number 0
```

```
-----  
select max(lo_ordtotalprice) most_expensive_order From LINEORDER
```

Plan hash value: 2267213921

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|----|----------------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 2211 (100) | |
| 1 | SORT AGGREGATE | | 1 | 6 | | |
| 2 | TABLE ACCESS INMEMORY FULL | LINEORDER | 23M | 137M | 2211 (12) | 00:00:01 |

14 rows selected.

Hit enter ...

| NAME | VALUE |
|---------------------------------------|------------|
| IM scan CUs columns accessed | 45 |
| IM scan CUs columns theoretical max | 765 |
| IM scan CUs memcompress for query low | 45 |
| IM scan CUs no cleanout | 45 |
| IM scan CUs split pieces | 63 |
| IM scan bytes in-memory | 1197266153 |
| IM scan bytes uncompressed | 2285455994 |
| IM scan rows | 23996604 |
| IM scan rows projected | 45 |
| IM scan rows valid | 23996604 |
| parse time cpu | 13 |
| parse time elapsed | 25 |
| physical reads cache | 107 |
| redo size | 736 |
| session logical reads | 185599 |
| session logical reads - IM | 179686 |
| session pga memory | 10907080 |
| session pga memory max | 11824584 |

18 rows selected.

```
SQL> @02_buffer_query_stats.sql
```

Connected.

```
SQL>
```

```
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

55279127

Elapsed: 00:00:02.05

```
SQL>
```

```
SQL> set echo off
```

Hit enter ...

PLAN_TABLE_OUTPUT

SQL_ID 9pnpqzggghwb1g, child number 0

```
select /*+ NO_INMEMORY */ max(lo_ordtotalprice) most_expensive_order  
From LINEORDER
```

Plan hash value: 2267213921

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|----|-------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 49056 (100) | |
| 1 | SORT AGGREGATE | | 1 | 6 | | |
| 2 | TABLE ACCESS FULL | LINEORDER | 23M | 137M | 49056 (1) | 00:00:02 |

15 rows selected.

Hit enter ...

| NAME | VALUE |
|------------------------|----------|
| IM scan segments disk | 1 |
| parse time cpu | 3 |
| parse time elapsed | 2 |
| physical reads cache | 1 |
| redo size | 736 |
| session logical reads | 178899 |
| session pga memory | 10513864 |
| session pga memory max | 10513864 |

8 rows selected.

SQL> @03_list_im_stats.sql

Connected.

SQL>

SQL> -- Display the list of IM session statistics

SQL>

```
SQL> SELECT display_name
2 FROM v$statname
3 WHERE display_name LIKE 'IM %';
```

DISPLAY_NAME

```
IM populate blocks invalid
IM populate transactions check
IM populate undo segheader rollback
IM populate undo records applied
IM populate transactions active
IM repopulate blocks invalid
IM repopulate transactions check
IM repopulate undo segheader rollback
IM repopulate undo records applied
IM repopulate transactions active
IM scan CUs rollback
IM scan CUs no rollback
IM scan CUs undo records applied
IM scan CUs cleanout
IM scan CUs no cleanout
IM scan journal cleanout
IM scan journal no cleanout
IM scan journal
IM scan rows journal total
IM scan found invalid smu
IM transactions
IM transactions rows journaled
IM transactions rows invalidated
IM transactions downgrade mode
IM transactions blocks invalidated
IM transactions CU cleanout
IM transactions journal cleanout
IM transactions found invalid CU
IM transactions CUs invalid
IM rac blocks invalid
IM rac CUs invalid
IM space CU extents allocated
IM space SMU extents allocated
IM space CU bytes allocated
IM space SMU bytes allocated
IM space CU creations initiated
IM space SMU creations initiated
IM space CU creations committed
IM space SMU creations committed
IM space private journal extents allocated
IM space private journal bytes allocated
IM space shared journal extents allocated
IM space shared journal bytes allocated
IM space CU extents freed
IM space SMU extents freed
IM space CU bytes freed
IM space SMU bytes freed
IM space private journal extents freed
IM space private journal bytes freed
IM space shared journal extents freed
IM space shared journal bytes freed
IM space segments allocated
IM space private journal segments allocated
IM space shared journal segments allocated
```

IM space segments freed
IM space private journal segments freed
IM space shared journal segments freed
IM populate segments requested
IM repopulate segments requested
IM repopulate segments requested
IM populate segments
IM repopulate segments
IM repopulate segments
IM populate CUs requested
IM repopulate CUs requested
IM repopulate CUs requested
IM repopulate (trickle) CUs requested
IM repopulate (scan) CUs requested
IM populate CUs resubmitted
IM repopulate CUs resubmitted
IM repopulate CUs resubmitted
IM repopulate (trickle) CUs resubmitted
IM repopulate (scan) CUs resubmitted
IM populate CUs
IM repopulate CUs
IM repopulate CUs
IM repopulate (trickle) CUs
IM repopulate (scan) CUs
IM populate bytes from storage
IM repopulate bytes from storage
IM repopulate bytes from storage
IM repopulate (trickle) bytes from storage
IM populate accumulated time (ms)
IM repopulate accumulated time (ms)
IM repopulate accumulated time (ms)
IM repopulate (trickle) accumulated time (ms)
IM populate CUs empty
IM repopulate CUs empty
IM repopulate CUs empty
IM repopulate (trickle) CUs empty
IM populate CUs columns
IM repopulate CUs columns
IM repopulate CUs columns
IM repopulate (trickle) CUs columns
IM populate bytes in-memory data
IM repopulate bytes in-memory data
IM repopulate bytes in-memory data
IM repopulate (trickle) bytes in-memory data
IM populate bytes uncompressed data
IM repopulate bytes uncompressed data
IM repopulate bytes uncompressed data
IM repopulate (trickle) bytes uncompressed data
IM populate rows
IM repopulate rows
IM repopulate rows
IM repopulate (trickle) rows
IM populate CUs chain pieces
IM repopulate CUs chain pieces
IM repopulate CUs chain pieces
IM repopulate (trickle) CUs chain pieces
IM populate CUs no memcompress
IM repopulate CUs no memcompress
IM repopulate CUs no memcompress
IM repopulate (trickle) CUs no memcompress
IM populate CUs memcompress for dml
IM repopulate CUs memcompress for dml
IM repopulate CUs memcompress for dml
IM repopulate (trickle) CUs memcompress for dml
IM populate CUs memcompress for query low
IM repopulate CUs memcompress for query low
IM repopulate CUs memcompress for query low
IM repopulate (trickle) CUs memcompress for query low
IM populate CUs memcompress for query high
IM repopulate CUs memcompress for query high
IM repopulate CUs memcompress for query high
IM repopulate (trickle) CUs memcompress for query high
IM populate CUs memcompress for capacity low
IM repopulate CUs memcompress for capacity low
IM repopulate CUs memcompress for capacity low
IM repopulate (trickle) CUs memcompress for capacity low
IM populate CUs memcompress for capacity high
IM repopulate CUs memcompress for capacity high
IM repopulate CUs memcompress for capacity high
IM repopulate (trickle) CUs memcompress for capacity high

```

IM populate (faststart) CUs read
IM populate (faststart) CUs verified
IM populate (faststart) CUs read attempts
IM populate (faststart) CUs bytes read
IM populate (faststart) accumulated time (ms)
IM populate (faststart) number of invalid savepoints
IM populate (faststart) number of incompatible savepoints
IM populate (faststart) number of savepoints updated
IM populate (faststart) CUs write requests
IM populate (faststart) CUs writes
IM populate (faststart) CUs bytes written
IM populate (faststart) CUs accumulated write time (ms)
IM populate (faststart) CUs wall clock write time (ms)
IM scan CUs no memcompress
IM scan CUs memcompress for dml
IM scan CUs memcompress for query low
IM scan CUs memcompress for query high
IM scan CUs memcompress for capacity low
IM scan CUs memcompress for capacity high
IM scan segments disk
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 valid
IM scan rows range excluded
IM scan rows discontinuous
IM scan rows excluded
IM scan rows optimized
IM scan rows projected
IM scan rows cache
IM scan blocks cache
IM scan fetches journal
IM scan rows journal
IM fetches by rowid from journal
IM scan CUs split pieces
IM scan CUs predicates received
IM scan CUs predicates applied
IM scan CUs predicates optimized
IM scan CUs optimized read
IM scan CUs pruned
IM scan segments minmax eligible
IM fetches by rowid from IMCU
IM fetches by rowid from disk
IM fetches by rowid from fetch list
IM fetches by rowid row invalid in IMCU
IM scan CUs column not in memory
IM scan CUs invalid
IM scan invalid all blocks
IM scan CUs invalid or missing revert to on disk extent
IM scan CUs failed to reget pin
IM scan CUs invalid (all rows are invalid)
IM repopulate (incremental) CUs
IM repopulate blocks now valid (ATC)
IM repopulate optimized
IM repopulate (doublebuffering) CUs requested
IM repopulate (doublebuffering) CUs
IM scan CUs doublebuffering
IM zzzz spare7
IM zzzz spare8
IM zzzz spare9
IM zzzz spare10

```

198 rows selected.

Elapsed: 00:00:00.01

SQL>

SQL> set echo off;

SQL> @04_single_key_im.sql

Connected.

SQL>

SQL> -- In-Memory Column Store query with a single key lookup

SQL>

```

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

```

LO_ORDERKEY LO_CUSTKEY LO_REVENUE

5000000 19459 2044455

Elapsed: 00:00:00.02

SQL>

SQL> set echo off

Hit enter ...

PLAN_TABLE_OUTPUT

SQL_ID 513g163sj3cv2, child number 0

select lo_orderkey, lo_custkey, lo_revenue from LINEORDER where
lo_orderkey = 5000000

Plan hash value: 4017770458

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|----------------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 2057 (100) | |
| * 1 | TABLE ACCESS INMEMORY FULL | LINEORDER | 4 | 68 | 2057 (5) | 00:00:01 |

Predicate Information (identified by operation id):

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

20 rows selected.

SQL> @05_single_key_buffer.sql

Connected.

SQL>

SQL> -- Buffer cache query with a single key lookup

SQL> -- IM column store access disabled via NO_INMEMORY hint

SQL>

SQL> select /*+ NO_INMEMORY */

2 lo_orderkey, lo_custkey, lo_revenue

3 from LINEORDER

4 where lo_orderkey = 5000000;

LO_ORDERKEY LO_CUSTKEY LO_REVENUE

5000000 19459 2044455

Elapsed: 00:00:01.66

SQL>

SQL> set echo off

Hit enter ...

PLAN_TABLE_OUTPUT

SQL_ID 4rcm8pc7xktun, child number 0

select /*+ NO_INMEMORY */ lo_orderkey, lo_custkey, lo_revenue
from LINEORDER where lo_orderkey = 5000000

Plan hash value: 4017770458

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 48965 (100) | |
| * 1 | TABLE ACCESS FULL | LINEORDER | 4 | 68 | 48965 (1) | 00:00:02 |

Predicate Information (identified by operation id):

1 - filter("LO_ORDERKEY"=5000000)

19 rows selected.

SQL> @06_index_comparison.sql

```

Connected.
SQL>
SQL> -- Execute the In-Memory query as baseline
SQL>
SQL> Select /* Without index */ lo_orderkey, lo_custkey, lo_revenue
  2 From   LINEORDER
  3 Where  lo_orderkey = 5000000;

```

```

LO_ORDERKEY LO_CUSTKEY LO_REVENUE
-----
5000000      19459      2044455

```

Elapsed: 00:00:00.01

```

SQL>
SQL> pause Hit enter ...
Hit enter ...

```

```

SQL>
SQL> select * from table(dbms_xplan.display_cursor());

```

PLAN_TABLE_OUTPUT

```

SQL_ID 7adpvn2jz419b, child number 0
-----

```

```

Select /* Without index */ lo_orderkey, lo_custkey, lo_revenue From
LINEORDER Where lo_orderkey = 5000000

```

Plan hash value: 4017770458

```

-----
| Id | Operation                               | Name          | Rows | Bytes | Cost (%CPU)| Time          |
-----
|  0 | SELECT STATEMENT                       |               |      |      |  2057 (100)|               |
|*  1 |  TABLE ACCESS INMEMORY FULL           | LINEORDER    |     4 |     68 |  2057 (5)  | 00:00:01    |
-----

```

Predicate Information (identified by operation id):

```

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

```

20 rows selected.

Elapsed: 00:00:00.02

```

SQL>
SQL> pause Hit enter ...
Hit enter ...

```

```

SQL>
SQL> -- Enable the use of invisible indexes
SQL>
SQL> alter session set optimizer_use_invisible_indexes=true;

```

Session altered.

Elapsed: 00:00:00.01

```

SQL>
SQL> pause Hit enter ...
Hit enter ...

```

```

SQL>
SQL> -- Execute the query again and include a new comment to ensure a hard parse
SQL>

```

```

SQL> Select /* With index */ lo_orderkey, lo_custkey, lo_revenue
  2 From   LINEORDER
  3 Where  lo_orderkey = 5000000;

```

```

LO_ORDERKEY LO_CUSTKEY LO_REVENUE
-----
5000000      19459      2044455

```

Elapsed: 00:00:00.04

```

SQL>
SQL> -- Compare the elapsed time of the query In-Memory and with index access
SQL>
SQL> pause Hit enter ...
Hit enter ...

```

```
SQL>
SQL> select * from table(dbms_xplan.display_cursor());
```

PLAN_TABLE_OUTPUT

```
SQL_ID 8kgqurq43dgq1, child number 0
```

```
Select /* With index */ lo_orderkey, lo_custkey, lo_revenue From
LINEORDER Where lo_orderkey = 5000000
```

Plan hash value: 747895665

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 4 (100) | |
| 1 | TABLE ACCESS BY INDEX ROWID BATCHED | LINEORDER | 4 | 68 | 4 (0) | 00:00:01 |
| * 2 | INDEX RANGE SCAN | STEP3_3 | 4 | | 3 (0) | 00:00:01 |

Predicate Information (identified by operation id):

2 - access("LO_ORDERKEY"=5000000)

20 rows selected.

Elapsed: 00:00:00.09

```
SQL>
SQL> -- Disable the use of invisible indexes
SQL>
SQL> alter session set optimizer_use_invisible_indexes=false;
```

Session altered.

Elapsed: 00:00:00.00

```
SQL>
SQL> set echo off
SQL> @07_storage_index.sql
Connected.
SQL>
SQL> -- Show the use of In-Memory storage indexes
SQL>
SQL> select lo_orderkey, lo_custkey, lo_revenue
  2 from LINEORDER
  3 where lo_orderkey = 5000000;
```

| LO_ORDERKEY | LO_CUSTKEY | LO_REVENUE |
|-------------|------------|------------|
| 5000000 | 19459 | 2044455 |

Elapsed: 00:00:00.00

```
SQL>
SQL> set echo off
Hit enter ...
```

PLAN_TABLE_OUTPUT

```
SQL_ID 513g163sj3cv2, child number 0
```

```
select lo_orderkey, lo_custkey, lo_revenue from LINEORDER where
lo_orderkey = 5000000
```

Plan hash value: 4017770458

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|----------------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 2057 (100) | |
| * 1 | TABLE ACCESS INMEMORY FULL | LINEORDER | 4 | 68 | 2057 (5) | 00:00:01 |

Predicate Information (identified by operation id):

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

20 rows selected.

Hit enter ...

| NAME | VALUE |
|---------------------------------------|------------|
| IM scan CUs columns accessed | 4 |
| IM scan CUs columns theoretical max | 765 |
| IM scan CUs memcompress for query low | 45 |
| IM scan CUs no cleanout | 45 |
| IM scan CUs predicates applied | 45 |
| IM scan CUs predicates optimized | 43 |
| IM scan CUs predicates received | 45 |
| IM scan CUs pruned | 43 |
| IM scan CUs split pieces | 63 |
| IM scan bytes in-memory | 1197266153 |
| IM scan bytes uncompressed | 2285455994 |
| IM scan rows | 23996604 |
| IM scan rows optimized | 22898323 |
| IM scan rows projected | 1 |
| IM scan rows valid | 1098281 |
| IM scan segments minmax eligible | 45 |
| redo size | 736 |
| session logical reads | 179743 |
| session logical reads - IM | 179686 |
| session pga memory | 9924040 |
| session pga memory max | 9924040 |

21 rows selected.

```
SQL> @08_min_max.sql
Connected.
SQL>
SQL> -- Show the use of In-Memory storage indexes and min/max pruning
SQL>
SQL> select max(lo_supplycost) most_expensive_bulk_order
2   from LINEORDER
3   where lo_quantity > 52;
```

MOST_EXPENSIVE_BULK_ORDER

125939

Elapsed: 00:00:00.01

```
SQL>
SQL> set echo off
Hit enter ...
```

PLAN_TABLE_OUTPUT

```
SQL_ID 1500su9tjy2hd, child number 0
-----
select max(lo_supplycost) most_expensive_bulk_order from LINEORDER
where lo_quantity > 52
```

Plan hash value: 2267213921

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|----------------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 2054 (100) | |
| 1 | SORT AGGREGATE | | 1 | 8 | | |
| * 2 | TABLE ACCESS INMEMORY FULL | LINEORDER | 1 | 8 | 2054 (5) | 00:00:01 |

Predicate Information (identified by operation id):

```
2 - inmemory("LO_QUANTITY">52)
   filter("LO_QUANTITY">52)
```

21 rows selected.

Hit enter ...

| NAME | VALUE |
|---------------------------------------|------------|
| IM scan CUs columns accessed | 4 |
| IM scan CUs columns theoretical max | 765 |
| IM scan CUs memcompress for query low | 45 |
| IM scan CUs no cleanout | 45 |
| IM scan CUs predicates applied | 45 |
| IM scan CUs predicates optimized | 43 |
| IM scan CUs predicates received | 45 |
| IM scan CUs pruned | 43 |
| IM scan CUs split pieces | 63 |
| IM scan bytes in-memory | 1197266153 |
| IM scan bytes uncompressed | 2285455994 |
| IM scan rows | 23996604 |
| IM scan rows optimized | 22897896 |
| IM scan rows projected | 2 |
| IM scan rows valid | 1098708 |
| IM scan segments minmax eligible | 45 |
| parse time cpu | 1 |
| parse time elapsed | 1 |
| physical reads cache | 1 |
| redo size | 692 |
| session logical reads | 179768 |
| session logical reads - IM | 179686 |
| session pga memory | 9924040 |
| session pga memory max | 9924040 |

24 rows selected.

```

SQL>
SQL> @09_min_max_buffer.sql
Connected.
SQL>
SQL> -- Buffer cache query cannot use In-Memory storage indexes
SQL> -- IM column store access disabled via NO_INMEMORY hint
SQL>
SQL> SELECT /*+ NO_INMEMORY */
  2     max(lo_supplycost) most_expensive_bulk_order
  3     from LINEORDER
  4     where lo_quantity > 52;

```

MOST_EXPENSIVE_BULK_ORDER

125939

Elapsed: 00:00:01.81

```

SQL>
SQL> set echo off
Hit enter ...

```

PLAN_TABLE_OUTPUT

SQL_ID 67xx7kfjy85mr, child number 0

```

SELECT /*+ NO_INMEMORY */      max(lo_supplycost)
most_expensive_bulk_order from LINEORDER where lo_quantity > 52

```

Plan hash value: 2267213921

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 49062 (100) | |
| 1 | SORT AGGREGATE | | 1 | 8 | | |
| * 2 | TABLE ACCESS FULL | LINEORDER | 1 | 8 | 49062 (1) | 00:00:02 |

Predicate Information (identified by operation id):

2 - filter("LO_QUANTITY">52)

20 rows selected.

Hit enter ...

| NAME | VALUE |
|------|-------|
|------|-------|

```

-----
IM scan segments disk          1
parse time cpu                 1
parse time elapsed             1
redo size                      736
session logical reads         178759
session pga memory            9924040
session pga memory max        9924040

```

7 rows selected.

SQL> @l0_multi_preds.sql

Connected.

SQL>

SQL> -- In-Memory column store query with multiple filter predicates

SQL>

```

SQL> Select  lo_orderkey, lo_custkey, lo_revenue
           2 From    LINEORDER
           3 Where   lo_custkey = 5641
           4 And    lo_shipmode = 'XXX AIR'
           5 And    lo_orderpriority = '5-LOW';

```

```

LO_ORDERKEY LO_CUSTKEY LO_REVENUE
-----
2183172      5641      2750802

```

Elapsed: 00:00:00.02

SQL>

SQL> set echo off

Hit enter ...

PLAN_TABLE_OUTPUT

SQL_ID chyfdb9dv938r, child number 0

```

Select  lo_orderkey, lo_custkey, lo_revenue From    LINEORDER Where
lo_custkey = 5641 And    lo_shipmode = 'XXX AIR' And
lo_orderpriority = '5-LOW'

```

Plan hash value: 4017770458

```

-----
| Id | Operation                               | Name          | Rows | Bytes | Cost (%CPU)| Time          |
-----
|  0 | SELECT STATEMENT                         |               |      |      |  2056 (100)|               |
|*  1 | TABLE ACCESS INMEMORY FULL             | LINEORDER    |     1 |    44 |  2056 (5)  | 00:00:01    |
-----

```

Predicate Information (identified by operation id):

```

-----
1 - inmemory(("LO_CUSTKEY"=5641 AND "LO_SHIPMODE"='XXX AIR' AND
"LO_ORDERPRIORITY"='5-LOW'))
   filter(("LO_CUSTKEY"=5641 AND "LO_SHIPMODE"='XXX AIR' AND
"LO_ORDERPRIORITY"='5-LOW'))
-----

```

23 rows selected.

Hit enter ...

```

NAME                                     VALUE
-----
IM scan CUs columns accessed             5
IM scan CUs columns theoretical max      765
IM scan CUs memcompress for query low    45
IM scan CUs no cleanout                  45
IM scan CUs predicates applied            91
IM scan CUs predicates optimized          44
IM scan CUs predicates received           91
IM scan CUs pruned                        44
IM scan CUs split pieces                  63
IM scan bytes in-memory                  1197266153
IM scan bytes uncompressed                2285455994
IM scan rows                              23996604
IM scan rows optimized                    23979398
IM scan rows projected                     1
IM scan rows valid                        17206

```

```

IM scan segments minmax eligible          45
parse time cpu                            1
parse time elapsed                         1
physical reads cache                       3
redo size                                  736
session logical reads                     179765
session logical reads - IM                 179686
session pga memory                         9989576
session pga memory max                     9989576

```

24 rows selected.

SQL> @i1_multi_preds_buffer.sql

Connected.

SQL>

SQL> -- Execute the same multi-predicate query against the buffer cache

SQL> -- IM column store access disabled via NO_INMEMORY hint

SQL>

```

SQL> select /*+ NO_INMEMORY */ lo_orderkey, lo_custkey, lo_revenue
  2 from LINEORDER
  3 where lo_custkey = 5641
  4 and lo_shipmode = 'XXX AIR'
  5 and lo_orderpriority = '5-LOW';

```

```

LO_ORDERKEY LO_CUSTKEY LO_REVENUE
-----
  2183172      5641    2750802

```

Elapsed: 00:00:01.66

SQL>

SQL> set echo off

Hit enter ...

PLAN_TABLE_OUTPUT

SQL_ID ayk17sfxn6u9n, child number 0

```

select /*+ NO_INMEMORY */ lo_orderkey, lo_custkey, lo_revenue from
LINEORDER where lo_custkey = 5641 and lo_shipmode = 'XXX AIR'
and lo_orderpriority = '5-LOW'

```

Plan hash value: 4017770458

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 49159 (100) | |
| * 1 | TABLE ACCESS FULL | LINEORDER | 1 | 44 | 49159 (1) | 00:00:02 |

Predicate Information (identified by operation id):

```

1 - filter(("LO_CUSTKEY"=5641 AND "LO_SHIPMODE"='XXX AIR' AND
"LO_ORDERPRIORITY"='5-LOW'))

```

21 rows selected.

Hit enter ...

```

NAME                                     VALUE
-----
IM scan segments disk                    1
parse time cpu                            1
parse time elapsed                         1
physical reads cache                       1
redo size                                  764
session logical reads                     178763
session pga memory                         9858504
session pga memory max                     9858504

```

8 rows selected.

SQL> @i2_multi_col.sql

Connected.

SQL>

SQL> -- In-Memory column store query accessing multiple columns

```

SQL>
SQL> Select lo_orderkey, lo_revenue
2 From LINEORDER
3 Where lo_revenue = (Select min(lo_revenue)
4 From LINEORDER
5 Where lo_supplycost = (Select max(lo_supplycost)
6 From LINEORDER
7 Where lo_quantity > 10)
8 And lo_shipmode LIKE 'TRUCK%'
9 And lo_discount between 2 and 5
10 );

```

```

LO_ORDERKEY LO_REVENUE
-----
11710945    1645608
13060610    1645608
14466627    1645608
15212229    1645608
15871937    1645608
17024903    1645608
18797413    1645608
19726659    1645608
21329476    1645608
3770919     1645608
5574821     1645608
8646727     1645608
8872933     1645608

```

13 rows selected.

Elapsed: 00:00:00.65

```

SQL>
SQL> set echo off
Hit enter ...

```

PLAN_TABLE_OUTPUT

```

SQL_ID bnysz8dqbamnd, child number 0
-----
Select lo_orderkey, lo_revenue From LINEORDER Where lo_revenue =
(Select min(lo_revenue) From LINEORDER
Where lo_supplycost = (Select max(lo_supplycost)
From LINEORDER
Where lo_quantity > 10)
And
lo_shipmode LIKE 'TRUCK%' And lo_discount between
2 and 5
)

```

Plan hash value: 482539359

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|----------------------------|-----------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | | | 6713 (100) | |
| * 1 | TABLE ACCESS INMEMORY FULL | LINEORDER | 4 | 48 | 2175 (10) | 00:00:01 |
| 2 | SORT AGGREGATE | | 1 | 25 | | |
| * 3 | TABLE ACCESS INMEMORY FULL | LINEORDER | 91 | 2275 | 2314 (16) | 00:00:01 |
| 4 | SORT AGGREGATE | | 1 | 8 | | |
| * 5 | TABLE ACCESS INMEMORY FULL | LINEORDER | 19M | 146M | 2224 (12) | 00:00:01 |

Predicate Information (identified by operation id):

- ```

1 - inmemory("LO_REVENUE"=)
 filter("LO_REVENUE"=)
3 - inmemory(("LO_DISCOUNT"<=5 AND "LO_SHIPMODE" LIKE 'TRUCK%' AND
 "LO_DISCOUNT">=2 AND "LO_SUPPLYCOST"=))
 filter(("LO_DISCOUNT"<=5 AND "LO_SHIPMODE" LIKE 'TRUCK%' AND
 "LO_DISCOUNT">=2 AND "LO_SUPPLYCOST"=))
5 - inmemory("LO_QUANTITY">10)
 filter("LO_QUANTITY">10)

```

35 rows selected.

Hit enter ...

NAME VALUE



```

IM scan CUs columns accessed 259
IM scan CUs columns theoretical max 2295
IM scan CUs memcompress for query low 135
IM scan CUs no cleanout 135
IM scan CUs predicates applied 135
IM scan CUs predicates received 135
IM scan CUs split pieces 189
IM scan bytes in-memory 3591798459
IM scan bytes uncompressed 6856367982
IM scan rows 71989812
IM scan rows optimized 793030
IM scan rows projected 61
IM scan rows valid 71989812
IM scan segments minmax eligible 45
parse time cpu 1
parse time elapsed 2
physical reads cache 2
redo size 736
session logical reads 539152
session logical reads - IM 539058
session pga memory 9924040
session pga memory max 9924040

```

22 rows selected.

SQL> @i3\_multi\_col\_buffer.sql

Connected.

SQL>

SQL> -- Execute same multi-column query against the buffer cache

SQL> -- IM column store access disabled via NO\_INMEMORY hint

SQL>

```

SQL> SELECT /*+ NO_INMEMORY */ lo_orderkey, lo_revenue
 2 From LINEORDER
 3 Where lo_revenue = (Select /*+ NO_INMEMORY */ min(lo_revenue)
 4 From LINEORDER
 5 Where lo_supplycost = (Select /*+ NO_INMEMORY */
 6 max(lo_supplycost)
 7 From LINEORDER
 8 Where lo_quantity > 10)
 9 And lo_shipmode LIKE 'TRUCK%'
 10 And lo_discount between 2 and 5
 11);

```

LO\_ORDERKEY LO\_REVENUE

```

3770919 1645608
5574821 1645608
8646727 1645608
8872933 1645608
11710945 1645608
13060610 1645608
14466627 1645608
15212229 1645608
15871937 1645608
17024903 1645608
18797413 1645608
19726659 1645608
21329476 1645608

```

13 rows selected.

Elapsed: 00:00:06.50

SQL>

SQL> set echo off

Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID 3m82kk7q6as80, child number 0

```

SELECT /*+ NO_INMEMORY */ lo_orderkey, lo_revenue From LINEORDER Where
lo_revenue = (Select /*+ NO_INMEMORY */ min(lo_revenue)
 From LINEORDER
 Where lo_supplycost = (Select
/*+ NO_INMEMORY */
max(lo_supplycost)
 From
LINEORDER
 Where
lo_quantity > 10)
 And lo_shipmode LIKE 'TRUCK%'
 And lo_discount between 2 and 5

```

)

Plan hash value: 482539359

| Id  | Operation         | Name      | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|-----------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |           |      |       | 147K(100)   |          |
| * 1 | TABLE ACCESS FULL | LINEORDER | 4    | 48    | 49111 (1)   | 00:00:02 |
| 2   | SORT AGGREGATE    |           | 1    | 25    |             |          |
| * 3 | TABLE ACCESS FULL | LINEORDER | 91   | 2275  | 49159 (1)   | 00:00:02 |
| 4   | SORT AGGREGATE    |           | 1    | 8     |             |          |
| * 5 | TABLE ACCESS FULL | LINEORDER | 19M  | 146M  | 49111 (1)   | 00:00:02 |

Predicate Information (identified by operation id):

- 1 - filter("LO\_REVENUE"=)
- 3 - filter(("LO\_DISCOUNT"<=5 AND "LO\_SHIPMODE" LIKE 'TRUCK%' AND "LO\_DISCOUNT">=2 AND "LO\_SUPPLYCOST"=))
- 5 - filter("LO\_QUANTITY">10)

33 rows selected.

Hit enter ...

| NAME                   | VALUE   |
|------------------------|---------|
| IM scan segments disk  | 3       |
| parse time cpu         | 1       |
| parse time elapsed     | 2       |
| physical reads cache   | 1       |
| redo size              | 736     |
| session logical reads  | 536159  |
| session pga memory     | 9989576 |
| session pga memory max | 9989576 |

8 rows selected.

SQL>

Part 3:

SQL> @@1\_join\_im.sql

Connected.

SQL>

SQL> -- In-Memory column store query with a two table join

SQL>

SQL> Select sum(lo\_extendedprice \* lo\_discount) revenue

- 2 From LINEORDER l, DATE\_DIM d
- 3 Where l.lo\_orderdate = d.d\_datekey
- 4 And l.lo\_discount between 2 and 3
- 5 And l.lo\_quantity < 24
- 6 And d.d\_date='December 24, 1996';

REVENUE

4062078072

Elapsed: 00:00:00.27

SQL>

SQL> set echo off

Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID b2jysvyzbss5p, child number 0

```
Select sum(lo_extendedprice * lo_discount) revenue From LINEORDER l,
DATE_DIM d Where l.lo_orderdate = d.d_datekey And l.lo_discount
between 2 and 3 And l.lo_quantity < 24 And d.d_date='December 24,
1996'
```

Plan hash value: 2403472142

| Id  | Operation                  | Name      | Rows  | Bytes | Cost (%CPU) | Time     |
|-----|----------------------------|-----------|-------|-------|-------------|----------|
| 0   | SELECT STATEMENT           |           |       |       | 2341 (100)  |          |
| 1   | SORT AGGREGATE             |           | 1     | 43    |             |          |
| * 2 | HASH JOIN                  |           | 834   | 35862 | 2341 (16)   | 00:00:01 |
| 3   | JOIN FILTER CREATE         | :BF0000   | 1     | 25    | 12 (0)      | 00:00:01 |
| * 4 | TABLE ACCESS INMEMORY FULL | DATE_DIM  | 1     | 25    | 12 (0)      | 00:00:01 |
| 5   | JOIN FILTER USE            | :BF0000   | 2005K | 34M   | 2324 (16)   | 00:00:01 |
| * 6 | TABLE ACCESS INMEMORY FULL | LINEORDER | 2005K | 34M   | 2324 (16)   | 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")))

```

32 rows selected.

Hit enter ...

| NAME                                  | VALUE      |
|---------------------------------------|------------|
| IM scan CUs columns accessed          | 181        |
| IM scan CUs columns theoretical max   | 782        |
| IM scan CUs memcompress for query low | 46         |
| IM scan CUs no cleanout               | 46         |
| IM scan CUs predicates applied        | 136        |
| IM scan CUs predicates received       | 136        |
| IM scan CUs split pieces              | 64         |
| IM scan bytes in-memory               | 1197355302 |
| IM scan bytes uncompressed            | 2285710880 |
| IM scan rows                          | 23999160   |
| IM scan rows projected                | 878        |
| IM scan rows valid                    | 23999160   |
| IM scan segments minmax eligible      | 1          |
| parse time cpu                        | 2          |
| parse time elapsed                    | 1          |
| physical reads cache                  | 1          |
| redo size                             | 736        |
| session logical reads                 | 179789     |
| session logical reads - IM            | 179701     |
| session pga memory                    | 9989576    |
| session pga memory max                | 9989576    |

21 rows selected.

SQL> @@02\_join\_buffer.sql

Connected.

SQL>

SQL> -- Buffer cache query with the same two table join

SQL>

SQL> SELECT /\*+ NO\_INMEMORY \*/

2 SUM(lo\_extendedprice \* lo\_discount) revenue

3 FROM lineorder l,

4 date\_dim d

5 WHERE l.lo\_orderdate = d.d\_datekey

6 AND l.lo\_discount BETWEEN 2 AND 3

7 AND l.lo\_quantity < 24

8 AND d.d\_date='December 24, 1996';

REVENUE

4062078072

Elapsed: 00:00:02.58

SQL>

SQL> set echo off

Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID 5fn4vtw9a8a7u, child number 0

```

SELECT /*+ NO_INMEMORY */ SUM(lo_extendedprice * lo_discount)
revenue FROM lineorder l, date_dim d WHERE l.lo_orderdate =
d.d_datekey AND l.lo_discount BETWEEN 2 AND 3 AND l.lo_quantity <
24 AND d.d_date='December 24, 1996'
```

Plan hash value: 2963256899

| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|-----------|-------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |           |       |       | 49135 (100) |          |
| 1   | SORT AGGREGATE    |           | 1     | 43    |             |          |
| * 2 | HASH JOIN         |           | 834   | 35862 | 49135 (1)   | 00:00:02 |
| * 3 | TABLE ACCESS FULL | DATE_DIM  | 1     | 25    | 15 (0)      | 00:00:01 |
| * 4 | TABLE ACCESS FULL | LINEORDER | 2005K | 34M   | 49114 (1)   | 00:00:02 |

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))
```

27 rows selected.

Hit enter ...

| NAME                   | VALUE   |
|------------------------|---------|
| IM scan segments disk  | 2       |
| parse time elapsed     | 1       |
| redo size              | 736     |
| session logical reads  | 178786  |
| session pga memory     | 9989576 |
| session pga memory max | 9989576 |

6 rows selected.

SQL> @03\_2join\_im.sql

Connected.

SQL> -- In-Memory column store query with a three table join (two dimension tables)

```
SQL> Select /*+ parallel(2) */ p.p_name, sum(l.lo_revenue)
2 From LINEORDER l, DATE_DIM d, PART p
3 Where l.lo_orderdate = d.d_datekey
4 And l.lo_partkey = p.p_partkey
5 And p.p_name = 'hot lavender'
6 And d.d_year = 1996
7 And d.d_month = 'December'
8 Group by p.p_name;
```

| P_NAME       | SUM(L.LO_REVENUE) |
|--------------|-------------------|
| hot lavender | 141260131         |

Elapsed: 00:00:00.75

SQL>  
SQL> set echo off  
Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID br1bv694n1mu, child number 0

```

Select /*+ parallel(2) */ p.p_name, sum(l.lo_revenue) From
LINEORDER l, DATE_DIM d, PART p Where l.lo_orderdate = d.d_datekey
And l.lo_partkey = p.p_partkey And p.p_name = 'hot
lavender' And d.d_year = 1996 And d.d_month = 'December'
Group by p.p_name
```

Plan hash value: 59878400

| Id   | Operation                  | Name      | Rows | Bytes | Cost (%CPU) | Time     | TQ    | IN-OUT | PQ Distrib |
|------|----------------------------|-----------|------|-------|-------------|----------|-------|--------|------------|
| 0    | SELECT STATEMENT           |           |      |       | 1384 (100)  |          |       |        |            |
| 1    | PX COORDINATOR             |           |      |       |             |          |       |        |            |
| 2    | PX SEND QC (RANDOM)        | :TQ10001  | 1    | 56    | 1384 (19)   | 00:00:01 | Q1,01 | P->S   | QC (RAND)  |
| 3    | HASH GROUP BY              |           | 1    | 56    | 1384 (19)   | 00:00:01 | Q1,01 | PCWP   |            |
| 4    | PX RECEIVE                 |           | 1    | 56    | 1384 (19)   | 00:00:01 | Q1,01 | PCWP   |            |
| 5    | PX SEND HASH               | :TQ10000  | 1    | 56    | 1384 (19)   | 00:00:01 | Q1,00 | P->P   | HASH       |
| 6    | HASH GROUP BY              |           | 1    | 56    | 1384 (19)   | 00:00:01 | Q1,00 | PCWP   |            |
| * 7  | HASH JOIN                  |           | 23   | 1288  | 1384 (19)   | 00:00:01 | Q1,00 | PCWP   |            |
| 8    | JOIN FILTER CREATE         | :BF0000   | 31   | 620   | 6 (0)       | 00:00:01 | Q1,00 | PCWP   |            |
| * 9  | TABLE ACCESS INMEMORY FULL | DATE_DIM  | 31   | 620   | 6 (0)       | 00:00:01 | Q1,00 | PCWP   |            |
| * 10 | HASH JOIN                  |           | 1802 | 64872 | 1377 (19)   | 00:00:01 | Q1,00 | PCWP   |            |
| 11   | JOIN FILTER CREATE         | :BF0001   | 52   | 988   | 38 (6)      | 00:00:01 | Q1,00 | PCWP   |            |
| * 12 | TABLE ACCESS INMEMORY FULL | PART      | 52   | 988   | 38 (6)      | 00:00:01 | Q1,00 | PCWP   |            |
| 13   | JOIN FILTER USE            | :BF0000   | 23M  | 389M  | 1309 (17)   | 00:00:01 | Q1,00 | PCWP   |            |
| 14   | JOIN FILTER USE            | :BF0001   | 23M  | 389M  | 1309 (17)   | 00:00:01 | Q1,00 | PCWP   |            |
| 15   | PX BLOCK ITERATOR          |           | 23M  | 389M  | 1309 (17)   | 00:00:01 | Q1,00 | PCWC   |            |
| * 16 | TABLE ACCESS INMEMORY FULL | LINEORDER | 23M  | 389M  | 1309 (17)   | 00:00:01 | Q1,00 | PCWP   |            |

Predicate Information (identified by operation id):

```

7 - access("L"."LO_ORDERDATE"="D"."D_DATEKEY")
9 - inmemory(("D"."D_YEAR"=1996 AND "D"."D_MONTH"='December'))
 filter(("D"."D_YEAR"=1996 AND "D"."D_MONTH"='December'))
10 - access("L"."LO_PARTKEY"="P"."P_PARTKEY")
12 - inmemory("P"."P_NAME"='hot lavender')
 filter("P"."P_NAME"='hot lavender')
16 - inmemory(:Z>=:Z AND :Z<=:Z AND SYS_OP_BLOOM_FILTER_LIST(SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_PARTKEY"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE")))
 filter(SYS_OP_BLOOM_FILTER_LIST(SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_PARTKEY"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE")))

```

#### Note

- dynamic statistics used: dynamic sampling (level=AUTO)
- Degree of Parallelism is 2 because of hint

51 rows selected.

SQL> @04\_2join\_buffer.sql

Connected.

SQL>

SQL> -- Buffer Cache query with the same three table join (two dimension tables)

SQL> -- Parallel hint used to enable bloom filters for comparison

SQL>

```

SQL> SELECT /*+ NO_INMEMORY parallel(2) */ p.p_name, sum(l.lo_revenue)
2 From LINEORDER l, DATE_DIM d, PART p
3 Where l.lo_orderdate = d.d_datekey
4 AND l.lo_partkey = p.p_partkey
5 And p.p_name = 'hot lavender'
6 And d.d_year = 1996
7 And d.d_month = 'December'
8 Group by p.p_name;

```

```

P_NAME SUM(L.LO_REVENUE)

hot lavender 141260131

```

Elapsed: 00:00:01.69

SQL>

SQL> set echo off

Hit enter ...

#### PLAN\_TABLE\_OUTPUT

SQL\_ID 73dfz5z6gbaqb, child number 0

```

SELECT /*+ NO_INMEMORY parallel(2) */ p.p_name, sum(l.lo_revenue)
From LINEORDER l, DATE_DIM d, PART p Where l.lo_orderdate =
d.d_datekey AND l.lo_partkey = p.p_partkey And p.p_name
= 'hot lavender' And d.d_year = 1996 And d.d_month =
'December' Group by p.p_name

```

Plan hash value: 59878400

| Id   | Operation           | Name      | Rows | Bytes | Cost (%CPU) | Time     | TQ    | IN-OUT | PQ Distrib |
|------|---------------------|-----------|------|-------|-------------|----------|-------|--------|------------|
| 0    | SELECT STATEMENT    |           |      |       | 28077 (100) |          |       |        |            |
| 1    | PX COORDINATOR      |           |      |       |             |          |       |        |            |
| 2    | PX SEND QC (RANDOM) | :TQ10001  | 1    | 56    | 28077 (1)   | 00:00:02 | Q1,01 | P->S   | QC (RAND)  |
| 3    | HASH GROUP BY       |           | 1    | 56    | 28077 (1)   | 00:00:02 | Q1,01 | PCWP   |            |
| 4    | PX RECEIVE          |           | 1    | 56    | 28077 (1)   | 00:00:02 | Q1,01 | PCWP   |            |
| 5    | PX SEND HASH        | :TQ10000  | 1    | 56    | 28077 (1)   | 00:00:02 | Q1,00 | P->P   | HASH       |
| 6    | HASH GROUP BY       |           | 1    | 56    | 28077 (1)   | 00:00:02 | Q1,00 | PCWP   |            |
| * 7  | HASH JOIN           |           | 23   | 1288  | 28077 (1)   | 00:00:02 | Q1,00 | PCWP   |            |
| 8    | JOIN FILTER CREATE  | :BF0000   | 31   | 620   | 8 (0)       | 00:00:01 | Q1,00 | PCWP   |            |
| * 9  | TABLE ACCESS FULL   | DATE_DIM  | 31   | 620   | 8 (0)       | 00:00:01 | Q1,00 | PCWP   |            |
| * 10 | HASH JOIN           |           | 1802 | 64872 | 28068 (1)   | 00:00:02 | Q1,00 | PCWP   |            |
| 11   | JOIN FILTER CREATE  | :BF0001   | 52   | 988   | 789 (1)     | 00:00:01 | Q1,00 | PCWP   |            |
| * 12 | TABLE ACCESS FULL   | PART      | 52   | 988   | 789 (1)     | 00:00:01 | Q1,00 | PCWP   |            |
| 13   | JOIN FILTER USE     | :BF0000   | 23M  | 389M  | 27249 (1)   | 00:00:02 | Q1,00 | PCWP   |            |
| 14   | JOIN FILTER USE     | :BF0001   | 23M  | 389M  | 27249 (1)   | 00:00:02 | Q1,00 | PCWP   |            |
| 15   | PX BLOCK ITERATOR   |           | 23M  | 389M  | 27249 (1)   | 00:00:02 | Q1,00 | PCWC   |            |
| * 16 | TABLE ACCESS FULL   | LINEORDER | 23M  | 389M  | 27249 (1)   | 00:00:02 | Q1,00 | PCWP   |            |

Predicate Information (identified by operation id):

```

7 - access("L"."LO_ORDERDATE"="D"."D_DATEKEY")
9 - filter(("D"."D_YEAR"=1996 AND "D"."D_MONTH"='December'))
10 - access("L"."LO_PARTKEY"="P"."P_PARTKEY")
12 - filter("P"."P_NAME"='hot lavender')
16 - access(:Z>=:Z AND :Z<=:Z)
 filter(SYS_OP_BLOOM_FILTER_LIST(SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_PARTKEY"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE")))

```

Note

- dynamic statistics used: dynamic sampling (level=AUTO)
- Degree of Parallelism is 2 because of hint

48 rows selected.

SQL> @05\_3join\_im.sql

Connected.

SQL>

SQL> -- In-Memory column store query with a four table join (three dimension tables used)

SQL>

SQL> Select /\*+ parallel(2)\*/ d.d\_year, p.p\_brand1, sum(lo\_revenue) rev

```

2 From lineorder l,
3 date_dim d,
4 part p,
5 supplier s
6 Where l.lo_orderdate = d.d_datekey
7 And l.lo_partkey = p.p_partkey
8 And l.lo_suppkey = s.s_suppkey
9 And p.p_category = 'MFGR#12'
10 And s.s_region = 'AMERICA'
11 AND d.d_year = 1997
12 Group by d.d_year, p.p_brand1;

```

| D_YEAR | P_BRAND1  | REV        |
|--------|-----------|------------|
| 1997   | MFGR#1225 | 2663448296 |
| 1997   | MFGR#121  | 2454237315 |
| 1997   | MFGR#1211 | 2979768752 |
| 1997   | MFGR#1231 | 2773065946 |
| 1997   | MFGR#1224 | 2933451091 |
| 1997   | MFGR#1229 | 2608144275 |
| 1997   | MFGR#125  | 2350731825 |
| 1997   | MFGR#1233 | 2688814245 |
| 1997   | MFGR#1237 | 2766844305 |
| 1997   | MFGR#1214 | 2580617186 |
| 1997   | MFGR#1227 | 2565626008 |
| 1997   | MFGR#1234 | 2742044147 |
| 1997   | MFGR#1226 | 2846266929 |
| 1997   | MFGR#1220 | 2532940235 |
| 1997   | MFGR#1216 | 2996329702 |
| 1997   | MFGR#124  | 2716893742 |
| 1997   | MFGR#1215 | 2634452840 |
| 1997   | MFGR#1213 | 2543587507 |

```

1997 MFGR#1228 3041905638
1997 MFGR#1235 2604792147
1997 MFGR#1230 2630001512
1997 MFGR#123 2815766932
1997 MFGR#1232 2901995021
1997 MFGR#122 2891165737
1997 MFGR#1240 2684084049
1997 MFGR#128 2813002419
1997 MFGR#126 2298455483
1997 MFGR#1236 2945067090
1997 MFGR#1218 2521082527
1997 MFGR#1238 2434206026
1997 MFGR#1210 2635829698
1997 MFGR#1219 2554568422
1997 MFGR#1222 2573595720
1997 MFGR#1239 2676667346
1997 MFGR#1212 2576636519
1997 MFGR#129 2572628884
1997 MFGR#127 2929389059
1997 MFGR#1223 2493139888
1997 MFGR#1221 2819768177
1997 MFGR#1217 2635742286

```

40 rows selected.

Elapsed: 00:00:01.65

SQL>

SQL> set echo off

Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID 3pr674wvcma89, child number 0

```

Select /*+ parallel(2)*/ d.d_year, p.p_brand1, sum(lo_revenue) rev
From lineorder l, date_dim d, part p,
supplier s Where l.lo_orderdate = d.d_datekey And l.lo_partkey
= p.p_partkey And l.lo_suppkey = s.s_suppkey And
p.p_category = 'MFGR#12' And s.s_region = 'AMERICA' AND
d.d_year = 1997 Group by d.d_year, p.p_brand1

```

Plan hash value: 1077726526

| Id   | Operation                  | Name      | Rows  | Bytes | Cost (%CPU) | Time     | TQ    | IN-OUT | PQ Distrib |
|------|----------------------------|-----------|-------|-------|-------------|----------|-------|--------|------------|
| 0    | SELECT STATEMENT           |           |       |       | 1430 (100)  |          |       |        |            |
| 1    | PX COORDINATOR             |           |       |       |             |          |       |        |            |
| 2    | PX SEND QC (RANDOM)        | :TQ10002  | 994   | 70574 | 1430 (21)   | 00:00:01 | Q1,02 | P->S   | QC (RAND)  |
| 3    | HASH GROUP BY              |           | 994   | 70574 | 1430 (21)   | 00:00:01 | Q1,02 | PCWP   |            |
| 4    | PX RECEIVE                 |           | 994   | 70574 | 1430 (21)   | 00:00:01 | Q1,02 | PCWP   |            |
| 5    | PX SEND HASH               | :TQ10001  | 994   | 70574 | 1430 (21)   | 00:00:01 | Q1,01 | P->P   | HASH       |
| 6    | HASH GROUP BY              |           | 994   | 70574 | 1430 (21)   | 00:00:01 | Q1,01 | PCWP   |            |
| * 7  | HASH JOIN                  |           | 43853 | 3040K | 1429 (21)   | 00:00:01 | Q1,01 | PCWP   |            |
| 8    | JOIN FILTER CREATE         | :BF0000   | 1636  | 27812 | 7 (0)       | 00:00:01 | Q1,01 | PCWP   |            |
| 9    | PX RECEIVE                 |           | 1636  | 27812 | 7 (0)       | 00:00:01 | Q1,01 | PCWP   |            |
| 10   | PX SEND BROADCAST          | :TQ10000  | 1636  | 27812 | 7 (0)       | 00:00:01 | Q1,00 | P->P   | BROADCAST  |
| 11   | PX BLOCK ITERATOR          |           | 1636  | 27812 | 7 (0)       | 00:00:01 | Q1,00 | PCWC   |            |
| * 12 | TABLE ACCESS INMEMORY FULL | SUPPLIER  | 1636  | 27812 | 7 (0)       | 00:00:01 | Q1,00 | PCWP   |            |
| * 13 | HASH JOIN                  |           | 214K  | 11M   | 1422 (21)   | 00:00:01 | Q1,01 | PCWP   |            |
| 14   | JOIN FILTER CREATE         | :BF0001   | 365   | 3650  | 6 (0)       | 00:00:01 | Q1,01 | PCWP   |            |
| * 15 | TABLE ACCESS INMEMORY FULL | DATE_DIM  | 365   | 3650  | 6 (0)       | 00:00:01 | Q1,01 | PCWP   |            |
| * 16 | HASH JOIN                  |           | 1413K | 59M   | 1413 (21)   | 00:00:01 | Q1,01 | PCWP   |            |
| 17   | JOIN FILTER CREATE         | :BF0002   | 23943 | 537K  | 41 (13)     | 00:00:01 | Q1,01 | PCWP   |            |
| * 18 | TABLE ACCESS INMEMORY FULL | PART      | 23943 | 537K  | 41 (13)     | 00:00:01 | Q1,01 | PCWP   |            |
| 19   | JOIN FILTER USE            | :BF0000   | 23M   | 480M  | 1342 (19)   | 00:00:01 | Q1,01 | PCWP   |            |
| 20   | JOIN FILTER USE            | :BF0001   | 23M   | 480M  | 1342 (19)   | 00:00:01 | Q1,01 | PCWP   |            |
| 21   | JOIN FILTER USE            | :BF0002   | 23M   | 480M  | 1342 (19)   | 00:00:01 | Q1,01 | PCWP   |            |
| 22   | PX BLOCK ITERATOR          |           | 23M   | 480M  | 1342 (19)   | 00:00:01 | Q1,01 | PCWC   |            |
| * 23 | TABLE ACCESS INMEMORY FULL | LINEORDER | 23M   | 480M  | 1342 (19)   | 00:00:01 | Q1,01 | PCWP   |            |

Predicate Information (identified by operation id):

- 7 - access("L"."LO\_SUPPKEY"="S"."S\_SUPPKEY")
- 12 - inmemory(:Z>=:Z AND :Z<=:Z AND "S"."S\_REGION"='AMERICA')  
filter("S"."S\_REGION"='AMERICA')
- 13 - access("L"."LO\_ORDERDATE"="D"."D\_DATEKEY")

```

15 - inmemory("D"."D_YEAR"=1997)
 filter("D"."D_YEAR"=1997)
16 - access("L"."LO_PARTKEY"="P"."P_PARTKEY")
18 - inmemory("P"."P_CATEGORY"='MFGR#12')
 filter("P"."P_CATEGORY"='MFGR#12')
23 - inmemory(:Z>=:Z AND :Z<=:Z AND SYS_OP_BLOOM_FILTER_LIST(SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_PARTKEY"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_SUPPKEY")))
 filter(SYS_OP_BLOOM_FILTER_LIST(SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_PARTKEY"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_ORDERDATE"),SYS_OP_BLOOM_FILTER(:BF0000,"L"."LO_SUPPKEY")))

```

Note

- dynamic statistics used: dynamic sampling (level=AUTO)
- Degree of Parallelism is 2 because of hint

62 rows selected.

```

SQL> @06_3join_buffer.sql
Connected.
SQL>
SQL> -- Buffer Cache query with the same four table join (three dimension tables used)
SQL>
SQL> SELECT /*+ opt_param('inmemory_query','disable') NO_VECTOR_TRANSFORM */
2 d.d_year, p.p_brand1, sum(lo_revenue) rev
3 From lineorder l,
4 date_dim d,
5 part p,
6 supplier s
7 Where l.lo_orderdate = d.d_datekey
8 And l.lo_partkey = p.p_partkey
9 And l.lo_suppkey = s.s_suppkey
10 And p.p_category = 'MFGR#12'
11 And s.s_region = 'AMERICA'
12 AND d.d_year = 1997
13 Group by d.d_year, p.p_brand1;

```

| D_YEAR | P_BRAND1  | REV        |
|--------|-----------|------------|
| 1997   | MFGR#1211 | 2979768752 |
| 1997   | MFGR#1225 | 2663448296 |
| 1997   | MFGR#1213 | 2543587507 |
| 1997   | MFGR#121  | 2454237315 |
| 1997   | MFGR#1215 | 2634452840 |
| 1997   | MFGR#1228 | 3041905638 |
| 1997   | MFGR#1235 | 2604792147 |
| 1997   | MFGR#123  | 2815766932 |
| 1997   | MFGR#1232 | 2901995021 |
| 1997   | MFGR#1224 | 2933451091 |
| 1997   | MFGR#1230 | 2630001512 |
| 1997   | MFGR#1231 | 2773065946 |
| 1997   | MFGR#1233 | 2688814245 |
| 1997   | MFGR#128  | 2813002419 |
| 1997   | MFGR#125  | 2350731825 |
| 1997   | MFGR#1229 | 2608144275 |
| 1997   | MFGR#1240 | 2684084049 |
| 1997   | MFGR#122  | 2891165737 |
| 1997   | MFGR#1218 | 2521082527 |
| 1997   | MFGR#1237 | 2766844305 |
| 1997   | MFGR#126  | 2298455483 |
| 1997   | MFGR#1210 | 2635829698 |
| 1997   | MFGR#1236 | 2945067090 |
| 1997   | MFGR#1238 | 2434206026 |
| 1997   | MFGR#1219 | 2554568422 |
| 1997   | MFGR#1222 | 2573595720 |
| 1997   | MFGR#1214 | 2580617186 |
| 1997   | MFGR#1234 | 2742044147 |
| 1997   | MFGR#1239 | 2676667346 |
| 1997   | MFGR#1227 | 2565626008 |
| 1997   | MFGR#127  | 2929389059 |
| 1997   | MFGR#129  | 2572628884 |
| 1997   | MFGR#1212 | 2576636519 |
| 1997   | MFGR#1226 | 2846266929 |
| 1997   | MFGR#1220 | 2532940235 |
| 1997   | MFGR#1223 | 2493139888 |
| 1997   | MFGR#1221 | 2819768177 |
| 1997   | MFGR#1217 | 2635742286 |
| 1997   | MFGR#124  | 2716893742 |
| 1997   | MFGR#1216 | 2996329702 |



40 rows selected.

Elapsed: 00:00:04.00

SQL>  
SQL> set echo off  
Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID 20m81zd0jahtx, child number 0

```
SELECT /*+ opt_param('inmemory_query','disable') NO_VECTOR_TRANSFORM
*/ d.d_year, p.p_brand1, sum(lo_revenue) rev From
lineorder l, date_dim d, part p, supplier s
Where l.lo_orderdate = d.d_datekey And l.lo_partkey =
p.p_partkey And l.lo_suppkey = s.s_suppkey And p.p_category
= 'MFGR#12' And s.s_region = 'AMERICA' AND d.d_year
= 1997 Group by d.d_year, p.p_brand1
```

Plan hash value: 42622955

| Id  | Operation         | Name      | Rows  | Bytes | Cost (%CPU) | Time     |
|-----|-------------------|-----------|-------|-------|-------------|----------|
| 0   | SELECT STATEMENT  |           |       |       | 50621 (100) |          |
| 1   | HASH GROUP BY     |           | 39    | 2834  | 50621 (1)   | 00:00:02 |
| * 2 | HASH JOIN         |           | 43853 | 3040K | 50619 (1)   | 00:00:02 |
| * 3 | TABLE ACCESS FULL | SUPPLIER  | 1636  | 27812 | 34 (0)      | 00:00:01 |
| * 4 | HASH JOIN         |           | 214K  | 11M   | 50584 (1)   | 00:00:02 |
| * 5 | TABLE ACCESS FULL | DATE_DIM  | 365   | 3650  | 15 (0)      | 00:00:01 |
| * 6 | HASH JOIN         |           | 1413K | 59M   | 50566 (1)   | 00:00:02 |
| * 7 | TABLE ACCESS FULL | PART      | 23943 | 537K  | 1425 (1)    | 00:00:01 |
| * 8 | TABLE ACCESS FULL | LINEORDER | 23M   | 480M  | 49080 (1)   | 00:00:02 |

Predicate Information (identified by operation id):

- 2 - access("L"."LO\_SUPPKEY"="S"."S\_SUPPKEY")
- 3 - filter("S"."S\_REGION"='AMERICA')
- 4 - access("L"."LO\_ORDERDATE"="D"."D\_DATEKEY")
- 5 - filter("D"."D\_YEAR"=1997)
- 6 - access("L"."LO\_PARTKEY"="P"."P\_PARTKEY")
- 7 - filter("P"."P\_CATEGORY"='MFGR#12')

Note

- dynamic statistics used: dynamic sampling (level=2)
- 1 Sql Plan Directive used for this statement

41 rows selected.

SQL> @07\_vgb\_im.sql  
Connected.

SQL>  
SQL> -- In-Memory Column Store query using In-Memory Aggregation (vector group by)  
SQL>

```
SQL> select d.d_year, c.c_nation, sum(lo_revenue - lo_supplycost) profit
2 From LINEORDER l, DATE_DIM d, PART p, SUPPLIER s, CUSTOMER C
3 Where l.lo_orderdate = d.d_datekey
4 And l.lo_partkey = p.p_partkey
5 And l.lo_suppkey = s.s_suppkey
6 And l.lo_custkey = c.c_custkey
7 And s.s_region = 'AMERICA'
8 And c.c_region = 'AMERICA'
9 Group by d.d_year, c.c_nation
10 Order by d.d_year, c.c_nation;
```

| D_YEAR | C_NATION      | PROFIT       |
|--------|---------------|--------------|
| 1992   | ARGENTINA     | 106691630942 |
| 1992   | BRAZIL        | 104171666577 |
| 1992   | CANADA        | 107159201705 |
| 1992   | PERU          | 101923998880 |
| 1992   | UNITED STATES | 106393397709 |
| 1993   | ARGENTINA     | 106191074801 |
| 1993   | BRAZIL        | 102356854362 |
| 1993   | CANADA        | 107824198188 |

```

1993 PERU 103049782330
1993 UNITED STATES 106759303012
1994 ARGENTINA 106502469295
1994 BRAZIL 103859787886
1994 CANADA 106181312862
1994 PERU 103634838303
1994 UNITED STATES 106568434053
1995 ARGENTINA 105717204642
1995 BRAZIL 104471664538
1995 CANADA 107840765138
1995 PERU 104480478600
1995 UNITED STATES 105696989785
1996 ARGENTINA 106073220381
1996 BRAZIL 104862977843
1996 CANADA 108498604007
1996 PERU 105096173692
1996 UNITED STATES 106825318105
1997 ARGENTINA 106012641521
1997 BRAZIL 103377388474
1997 CANADA 108844514171
1997 PERU 103182763745
1997 UNITED STATES 106984669382
1998 ARGENTINA 62087063983
1998 BRAZIL 62381889968
1998 CANADA 63275449761
1998 PERU 61096753808
1998 UNITED STATES 62192673019

```

35 rows selected.

Elapsed: 00:00:01.06

SQL>

SQL> set echo off

Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID czxxt1nt4vpg, child number 0

```

select d.d_year, c.c_nation, sum(lo_revenue - lo_supplycost) profit
 From LINEORDER l, DATE_DIM d, PART p, SUPPLIER s, CUSTOMER
 C Where l.lo_orderdate = d.d_datekey And
 l.lo_partkey = p.p_partkey And l.lo_suppkey
 = s.s_suppkey And l.lo_custkey = c.c_custkey
 And s.s_region = 'AMERICA' And
 c.c_region = 'AMERICA' Group by d.d_year,
 c.c_nation Order by d.d_year, c.c_nation

```

Plan hash value: 2079491949

| Id   | Operation                  | Name                      | Rows  | Bytes | TempSpc | Cost (%CPU) | Time     |
|------|----------------------------|---------------------------|-------|-------|---------|-------------|----------|
| 0    | SELECT STATEMENT           |                           |       |       |         | 4748 (100)  |          |
| 1    | TEMP TABLE TRANSFORMATION  |                           |       |       |         |             |          |
| 2    | LOAD AS SELECT             |                           |       |       |         |             |          |
| 3    | VECTOR GROUP BY            |                           | 8     | 80    |         | 13 (8)      | 00:00:01 |
| 4    | KEY VECTOR CREATE BUFFERED | :KV0000                   |       |       |         |             |          |
| 5    | TABLE ACCESS INMEMORY FULL | DATE_DIM                  | 2556  | 25560 |         | 12 (0)      | 00:00:01 |
| 6    | LOAD AS SELECT             |                           |       |       |         |             |          |
| 7    | VECTOR GROUP BY            |                           | 1     | 5     | 7072K   | 1897 (2)    | 00:00:01 |
| 8    | HASH GROUP BY              |                           | 1     | 5     |         | 1897 (2)    | 00:00:01 |
| 9    | KEY VECTOR CREATE BUFFERED | :KV0001                   |       |       |         |             |          |
| 10   | TABLE ACCESS INMEMORY FULL | PART                      | 600K  | 2929K |         | 66 (4)      | 00:00:01 |
| 11   | LOAD AS SELECT             |                           |       |       |         |             |          |
| 12   | VECTOR GROUP BY            |                           | 1     | 17    |         | 13 (8)      | 00:00:01 |
| 13   | HASH GROUP BY              |                           | 1     | 17    |         | 13 (8)      | 00:00:01 |
| 14   | KEY VECTOR CREATE BUFFERED | :KV0002                   |       |       |         |             |          |
| * 15 | TABLE ACCESS INMEMORY FULL | SUPPLIER                  | 1636  | 27812 |         | 12 (0)      | 00:00:01 |
| 16   | LOAD AS SELECT             |                           |       |       |         |             |          |
| 17   | VECTOR GROUP BY            |                           | 25    | 850   | 1048K   | 247 (2)     | 00:00:01 |
| 18   | KEY VECTOR CREATE BUFFERED | :KV0003                   |       |       |         |             |          |
| * 19 | TABLE ACCESS INMEMORY FULL | CUSTOMER                  | 24000 | 796K  |         | 25 (4)      | 00:00:01 |
| 20   | SORT GROUP BY              |                           | 71    | 10437 |         | 2578 (24)   | 00:00:01 |
| * 21 | HASH JOIN                  |                           | 71    | 10437 |         | 2577 (24)   | 00:00:01 |
| 22   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D660E_44D02D | 25    | 850   |         | 2 (0)       | 00:00:01 |
| * 23 | HASH JOIN                  |                           | 71    | 8023  |         | 2575 (24)   | 00:00:01 |
| 24   | MERGE JOIN CARTESIAN       |                           | 8     | 256   |         | 6 (0)       | 00:00:01 |
| 25   | MERGE JOIN CARTESIAN       |                           | 1     | 22    |         | 4 (0)       | 00:00:01 |

|      |                            |                           |     |      |      |      |          |
|------|----------------------------|---------------------------|-----|------|------|------|----------|
| 26   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D660C_44D02D | 1   | 5    | 2    | (0)  | 00:00:01 |
| 27   | BUFFER SORT                |                           | 1   | 17   | 2    | (0)  | 00:00:01 |
| 28   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D660D_44D02D | 1   | 17   | 2    | (0)  | 00:00:01 |
| 29   | BUFFER SORT                |                           | 8   | 80   | 4    | (0)  | 00:00:01 |
| 30   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D660B_44D02D | 8   | 80   | 2    | (0)  | 00:00:01 |
| 31   | VIEW                       | VW_VT_80F21617            | 71  | 5751 | 2569 | (24) | 00:00:01 |
| 32   | VECTOR GROUP BY            |                           | 71  | 2201 | 2569 | (24) | 00:00:01 |
| 33   | HASH GROUP BY              |                           | 71  | 2201 | 2569 | (24) | 00:00:01 |
| 34   | KEY VECTOR USE             | :KV0000                   |     |      |      |      |          |
| 35   | KEY VECTOR USE             | :KV0001                   |     |      |      |      |          |
| 36   | KEY VECTOR USE             | :KV0003                   |     |      |      |      |          |
| 37   | KEY VECTOR USE             | :KV0002                   |     |      |      |      |          |
| * 38 | TABLE ACCESS INMEMORY FULL | LINEORDER                 | 23M | 709M | 2549 | (23) | 00:00:01 |

Predicate Information (identified by operation id):

```

15 - inmemory("S"."S_REGION"='AMERICA')
 filter("S"."S_REGION"='AMERICA')
19 - inmemory("C"."C_REGION"='AMERICA')
 filter("C"."C_REGION"='AMERICA')
21 - access("ITEM_16"=INTERNAL_FUNCTION("C0") AND "ITEM_17"="C2")
23 - access("ITEM_12"=INTERNAL_FUNCTION("C0") AND "ITEM_13"="C2" AND "ITEM_14"=INTERNAL_FUNCTION("C0") AND
 "ITEM_15"="C2" AND "ITEM_18"=INTERNAL_FUNCTION("C0") AND "ITEM_19"="C2")
38 - inmemory((SYS_OP_KEY_VECTOR_FILTER("L"."LO_SUPPKEY",:KV0002) AND
 SYS_OP_KEY_VECTOR_FILTER("L"."LO_CUSTKEY",:KV0003) AND SYS_OP_KEY_VECTOR_FILTER("L"."LO_PARTKEY",:KV0001) AND
 SYS_OP_KEY_VECTOR_FILTER("L"."LO_ORDERDATE",:KV0000)))
 filter((SYS_OP_KEY_VECTOR_FILTER("L"."LO_SUPPKEY",:KV0002) AND
 SYS_OP_KEY_VECTOR_FILTER("L"."LO_CUSTKEY",:KV0003) AND SYS_OP_KEY_VECTOR_FILTER("L"."LO_PARTKEY",:KV0001) AND
 SYS_OP_KEY_VECTOR_FILTER("L"."LO_ORDERDATE",:KV0000)))

```

Note

- vector transformation used for this statement

78 rows selected.

Hit enter ...

| NAME                                  | VALUE      |
|---------------------------------------|------------|
| IM scan CUs columns accessed          | 279        |
| IM scan CUs columns theoretical max   | 815        |
| IM scan CUs memcompress for query low | 50         |
| IM scan CUs no cleanout               | 50         |
| IM scan CUs split pieces              | 68         |
| IM scan bytes in-memory               | 1218953217 |
| IM scan bytes uncompressed            | 2350992308 |
| IM scan rows                          | 24727160   |
| IM scan rows projected                | 1609733    |
| IM scan rows valid                    | 24727160   |
| parse time cpu                        | 6          |
| parse time elapsed                    | 6          |
| physical reads cache                  | 4          |
| redo size                             | 4592       |
| session logical reads                 | 186425     |
| session logical reads - IM            | 186193     |
| session pga memory                    | 10513864   |
| session pga memory max                | 49049032   |

18 rows selected.

SQL> @08\_vgb\_buffer.sql

Connected.

SQL>

SQL> -- Buffer cache version of the same In-Memory Aggregation (vector group by) query

SQL>

```

SQL> SELECT /*+ opt_param('inmemory_query','disable') */
2 d.d_year, c.c_nation, sum(lo_revenue - lo_supplycost) profit
3 From LINEORDER l, DATE_DIM d, PART p, SUPPLIER s, CUSTOMER C
4 Where l.lo_orderdate = d.d_datekey
5 And l.lo_partkey = p.p_partkey
6 And l.lo_suppkey = s.s_suppkey
7 And l.lo_custkey = c.c_custkey
8 And s.s_region = 'AMERICA'
9 And c.c_region = 'AMERICA'
10 Group by d.d_year, c.c_nation

```

11 Order by d.d\_year, c.c\_nation;

| D_YEAR | C_NATION      | PROFIT       |
|--------|---------------|--------------|
| 1992   | ARGENTINA     | 106691630942 |
| 1992   | BRAZIL        | 104171666577 |
| 1992   | CANADA        | 107159201705 |
| 1992   | PERU          | 101923998800 |
| 1992   | UNITED STATES | 106393397709 |
| 1993   | ARGENTINA     | 106191074801 |
| 1993   | BRAZIL        | 102356854362 |
| 1993   | CANADA        | 107824198188 |
| 1993   | PERU          | 103049782330 |
| 1993   | UNITED STATES | 106759303012 |
| 1994   | ARGENTINA     | 106502469295 |
| 1994   | BRAZIL        | 103859787886 |
| 1994   | CANADA        | 106181312862 |
| 1994   | PERU          | 103634838303 |
| 1994   | UNITED STATES | 106568434053 |
| 1995   | ARGENTINA     | 105717204642 |
| 1995   | BRAZIL        | 104471664538 |
| 1995   | CANADA        | 107840765138 |
| 1995   | PERU          | 104480478600 |
| 1995   | UNITED STATES | 105696989785 |
| 1996   | ARGENTINA     | 106073220381 |
| 1996   | BRAZIL        | 104862977843 |
| 1996   | CANADA        | 108498604007 |
| 1996   | PERU          | 105096173692 |
| 1996   | UNITED STATES | 106825318105 |
| 1997   | ARGENTINA     | 106012641521 |
| 1997   | BRAZIL        | 103377388474 |
| 1997   | CANADA        | 108844514171 |
| 1997   | PERU          | 103182763745 |
| 1997   | UNITED STATES | 106984669382 |
| 1998   | ARGENTINA     | 62087063983  |
| 1998   | BRAZIL        | 62381889968  |
| 1998   | CANADA        | 63275449761  |
| 1998   | PERU          | 61096753808  |
| 1998   | UNITED STATES | 62192673019  |

35 rows selected.

Elapsed: 00:00:03.17

SQL>

SQL> set echo off

Hit enter ...

PLAN\_TABLE\_OUTPUT

SQL\_ID c25778unnnc8j, child number 0

```

SELECT /*+ opt_param('inmemory_query','disable') */ d.d_year,
c.c_nation, sum(lo_revenue - lo_supplycost) profit From
LINEORDER l, DATE_DIM d, PART p, SUPPLIER s, CUSTOMER C Where
 l.lo_orderdate = d.d_datekey And l.lo_partkey
= p.p_partkey And l.lo_suppkey = s.s_suppkey
 And l.lo_custkey = c.c_custkey And
s.s_region = 'AMERICA' And c.c_region
= 'AMERICA' Group by d.d_year, c.c_nation
Order by d.d_year, c.c_nation

```

Plan hash value: 2079491949

| Id | Operation                  | Name     | Rows | Bytes | TempSpC | Cost (%CPU) | Time     |
|----|----------------------------|----------|------|-------|---------|-------------|----------|
| 0  | SELECT STATEMENT           |          |      |       |         | 52986 (100) |          |
| 1  | TEMP TABLE TRANSFORMATION  |          |      |       |         |             |          |
| 2  | LOAD AS SELECT             |          |      |       |         |             |          |
| 3  | VECTOR GROUP BY            |          | 8    | 80    |         | 16 (7)      | 00:00:01 |
| 4  | KEY VECTOR CREATE BUFFERED | :KV0000  |      |       |         |             |          |
| 5  | TABLE ACCESS FULL          | DATE_DIM | 2556 | 25560 |         | 15 (0)      | 00:00:01 |
| 6  | LOAD AS SELECT             |          |      |       |         |             |          |
| 7  | VECTOR GROUP BY            |          | 1    | 5     | 7072K   | 3251 (1)    | 00:00:01 |
| 8  | HASH GROUP BY              |          | 1    | 5     |         | 3251 (1)    | 00:00:01 |
| 9  | KEY VECTOR CREATE BUFFERED | :KV0001  |      |       |         |             |          |
| 10 | TABLE ACCESS FULL          | PART     | 600K | 2929K |         | 1420 (1)    | 00:00:01 |
| 11 | LOAD AS SELECT             |          |      |       |         |             |          |
| 12 | VECTOR GROUP BY            |          | 1    | 17    |         | 35 (3)      | 00:00:01 |

|      |                            |                           |       |       |       |       |     |          |
|------|----------------------------|---------------------------|-------|-------|-------|-------|-----|----------|
| 13   | HASH GROUP BY              |                           | 1     | 17    |       | 35    | (3) | 00:00:01 |
| 14   | KEY VECTOR CREATE BUFFERED | :KV0002                   |       |       |       |       |     |          |
| * 15 | TABLE ACCESS FULL          | SUPPLIER                  | 1636  | 27812 |       | 34    | (0) | 00:00:01 |
| 16   | LOAD AS SELECT             |                           |       |       |       |       |     |          |
| 17   | VECTOR GROUP BY            |                           | 25    | 850   | 1048K | 562   | (1) | 00:00:01 |
| 18   | KEY VECTOR CREATE BUFFERED | :KV0003                   |       |       |       |       |     |          |
| * 19 | TABLE ACCESS FULL          | CUSTOMER                  | 24000 | 796K  |       | 341   | (1) | 00:00:01 |
| 20   | SORT GROUP BY              |                           | 71    | 10437 |       | 49121 | (1) | 00:00:02 |
| * 21 | HASH JOIN                  |                           | 71    | 10437 |       | 49120 | (1) | 00:00:02 |
| 22   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D6612_44D02D | 25    | 850   |       | 2     | (0) | 00:00:01 |
| * 23 | HASH JOIN                  |                           | 71    | 8023  |       | 49118 | (1) | 00:00:02 |
| 24   | MERGE JOIN CARTESIAN       |                           | 8     | 256   |       | 6     | (0) | 00:00:01 |
| 25   | MERGE JOIN CARTESIAN       |                           | 1     | 22    |       | 4     | (0) | 00:00:01 |
| 26   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D6610_44D02D | 1     | 5     |       | 2     | (0) | 00:00:01 |
| 27   | BUFFER SORT                |                           | 1     | 17    |       | 2     | (0) | 00:00:01 |
| 28   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D6611_44D02D | 1     | 17    |       | 2     | (0) | 00:00:01 |
| 29   | BUFFER SORT                |                           | 8     | 80    |       | 4     | (0) | 00:00:01 |
| 30   | TABLE ACCESS FULL          | SYS_TEMP_0FD9D660F_44D02D | 8     | 80    |       | 2     | (0) | 00:00:01 |
| 31   | VIEW                       | VW_VT_80F21617            | 71    | 5751  |       | 49112 | (1) | 00:00:02 |
| 32   | VECTOR GROUP BY            |                           | 71    | 2201  |       | 49112 | (1) | 00:00:02 |
| 33   | HASH GROUP BY              |                           | 71    | 2201  |       | 49112 | (1) | 00:00:02 |
| 34   | KEY VECTOR USE             | :KV0000                   |       |       |       |       |     |          |
| 35   | KEY VECTOR USE             | :KV0001                   |       |       |       |       |     |          |
| 36   | KEY VECTOR USE             | :KV0003                   |       |       |       |       |     |          |
| 37   | KEY VECTOR USE             | :KV0002                   |       |       |       |       |     |          |
| * 38 | TABLE ACCESS FULL          | LINEORDER                 | 23M   | 709M  |       | 49092 | (1) | 00:00:02 |

Predicate Information (identified by operation id):

```

15 - filter("S"."S_REGION"='AMERICA')
19 - filter("C"."C_REGION"='AMERICA')
21 - access("ITEM_16"=INTERNAL_FUNCTION("C0") AND "ITEM_17"="C2")
23 - access("ITEM_12"=INTERNAL_FUNCTION("C0") AND "ITEM_13"="C2" AND "ITEM_14"=INTERNAL_FUNCTION("C0")
AND "ITEM_15"="C2" AND "ITEM_18"=INTERNAL_FUNCTION("C0") AND "ITEM_19"="C2")
38 - filter((SYS_OP_KEY_VECTOR_FILTER("L"."LO_SUPPKEY",:KV0002) AND
SYS_OP_KEY_VECTOR_FILTER("L"."LO_CUSTKEY",:KV0003)))

```

Note

- vector transformation used for this statement

73 rows selected.

Hit enter ...

| NAME                   | VALUE    |
|------------------------|----------|
| IM scan segments disk  | 5        |
| parse time cpu         | 5        |
| parse time elapsed     | 6        |
| physical reads cache   | 4        |
| redo size              | 4592     |
| session logical reads  | 185420   |
| session pga memory     | 10382792 |
| session pga memory max | 48131528 |

8 rows selected.

SQL>