

Database In-Memory By Example
Andy Rivenes, Product Manager, Oracle
Copyright (c) 2019, Oracle and/or its affiliates. All rights reserved.

Oracle User Group
NYOUG Fall Conference 2019
September 25, 2019

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

```
[oracle@vbgeneric scripts]$ ls
01_show_sga.sql      06_im_query_stats.sql    11_ime_usage.sql      16_use_jg.sql
02_show_parms.sql   07_buffer_query_stats.sql 12_with-ime.sql      17_vgb_im.sql
03_im_attributes.sql 08_index_comparison.sql  13_join_im.sql       18_novgb_buffer.sql
04_im_populated.sql 09_storage_index.sql     14_join_buffer.sql
05_im_usage.sql      10_no-ime.sql           15_query_jg.sql
[oracle@vbgeneric scripts]$ sqlplus /nolog
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Fri Mar 8 19:01:31 2019
Version 18.5.0.0.0
```

Copyright (c) 1982, 2018, Oracle. All rights reserved.

```
SQL> @01_show_sga.sql
Connected.
SQL> set numwidth 20
SQL>
SQL> -- This command shows the settings of all in-memory init.ora parameters
SQL>
SQL> show sga
```

```
Total System Global Area      11039407024 bytes
Fixed Size                      8907696 bytes
Variable Size                  603979776 bytes
Database Buffers               4630511616 bytes
Redo Buffers                   24645632 bytes
In-Memory Area                 5771362304 bytes
```

```
SQL>
SQL> set echo off
SQL> @02_show_parms.sql
Connected.
SQL>
SQL> -- Shows the SGA init.ora parameters
SQL>
SQL> show parameter sga_target
```

NAME	TYPE	VALUE
sga_target	big integer	10000M

```
SQL>
SQL> -- Show the KEEP pool init.ora parameters
SQL>
SQL> show parameter db_keep_cache_size
```

NAME	TYPE	VALUE
db_keep_cache_size	big integer	4000M

```
SQL>
SQL> -- Show the settings of all in-memory init.ora parameters
SQL>
SQL> show parameter inmemory
```

```

NAME                                     TYPE      VALUE
-----
_inmemory_64k_percent                   integer    10
inmemory_adg_enabled                     boolean    TRUE
inmemory_automatic_level                 string     OFF
inmemory_clause_default                  string
inmemory_expressions_usage               string     ENABLE
inmemory_force                           string     DEFAULT
inmemory_max_populate_servers            integer    4
inmemory_optimized_arithmetic            string     DISABLE
inmemory_prefer_xmem_memcompress         string
inmemory_prefer_xmem_priority             string
inmemory_query                           string     ENABLE
inmemory_size                            big integer 5504M
inmemory_trickle_repopulate_servers_    integer    1
percent
inmemory_virtual_columns                  string     ENABLE
inmemory_xmem_size                       big integer 0
optimizer_inmemory_aware                  boolean    TRUE
SQL>
SQL> set echo off
SQL> @03_im_attributes.sql
Connected.
SQL>
SQL> -- This query allows you to review the current attributes of the tables in SSB schem
SQL>
SQL> select table_name, cache, buffer_pool, compression, compress_for, inmemory,
2         inmemory_priority, inmemory_distribute, inmemory_compression
3 from    user_tables;

```

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

```

SQL>
SQL> set echo off
SQL> @04_im_populated.sql
Connected.
SQL>
SQL> -- Query the view v$IM_SEGMENTS to shows what objects are in the column store
SQL> -- and how much of the objects were populated. When the BYTES_NOT_POPULATED is 0
SQL> -- it indicates the entire table was populated.
SQL>
SQL> SELECT
2     v.owner, v.segment_name name, v.populate_status status, v.bytes bytes_in_mem,
v.bytes_not_populated
3 FROM   v$im_segments v;

```

OWNER	NAME	STATUS	BYTES_IN_MEM	BYTES_NOT_POPULATED
SSB	PART	COMPLETED	56,721,408	0
SSB	CUSTOMER	COMPLETED	24,862,720	0
SSB	LINEORDER	COMPLETED	3,693,232,128	0
SSB	DATE_DIM	COMPLETED	122,880	0

SSB SUPPLIER COMPLETED 1,761,280 0

```
SQL>
SQL> set echo off
SQL> @05_im_usage.sql
Connected.
SQL> column alloc_bytes format 999,999,999,999,999
SQL> column used_bytes format 999,999,999,999,999
SQL>
SQL> -- This query displays what objects are in the In-Memory Column Store
SQL>
SQL> SELECT *
  2 FROM v$inmemory_area;
```

POOL CON_ID	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS
1MB POOL 3	5,163,188,224	4,450,156,544	DONE
64KB POOL 3	587,202,560	9,568,256	DONE

```
SQL>
SQL> set echo off
SQL> @06_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
57346348

Elapsed: 00:00:00.13
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				14417 (100)	
1	SORT AGGREGATE		1	6		
2	TABLE ACCESS INMEMORY FULL	LINEORDER	59M	343M	14417 (10)	00:00:01

14 rows selected.
Hit enter ...

NAME	VALUE
CPU used by this session	59

```

IM scan CUs columns accessed          111
IM scan CUs memcompress for query low 111
IM scan rows                          59986052
IM scan rows projected                111
physical reads                        74
session logical reads                 461037
session logical reads - IM            450834
session pga memory                    13690600
table scans (IM)                      1

```

10 rows selected.

```

SQL> @07_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
-----
          57346348

```

```

Elapsed: 00:00:09.41
SQL>
SQL> set echo off
Hit enter ...

```

PLAN_TABLE_OUTPUT

```

-----
SQL_ID 9pnpzgghwb1g, 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				123K(100)	
1	SORT AGGREGATE		1	6		
2	TABLE ACCESS FULL	LINEORDER	59M	343M	123K (1)	00:00:05

15 rows selected.

Hit enter ...

```

NAME                                     VALUE
-----
CPU used by this session                 966
IM scan segments disk                    1
parse time cpu                           47
parse time elapsed                       70
redo size                                1400
session logical reads                    463967
session pga memory                       14083816
session pga memory max                   14083816

```

8 rows selected.

```

SQL> @08_index_comparison.sql
Connected.

```

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

Session altered.

```
SQL>
SQL> -- Execute the query again include a new comment to ensure a hard parse
SQL>
SQL> set timing on
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	48647	2456268

```
Elapsed: 00:00:00.09
SQL>
SQL> set timing off
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.

```
SQL>
SQL> -- Compare the Elapsed time of the query In-Memory and in the index access
SQL>
SQL> alter session set optimizer_use_invisible_indexes=false;
```

Session altered.

```

SQL>
SQL> set echo off
SQL> @09_storage_index.sql
Connected.
SQL>
SQL> -- Execute the In-Memory Column Store query
SQL>
SQL> select  lo_orderkey, lo_custkey, lo_revenue
           2  from    LINEORDER
           3  where   lo_orderkey = 5000000;

```

LO_ORDERKEY	LO_CUSTKEY	LO_REVENUE
5000000	48647	2456268

Elapsed: 00:00:00.13

```

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				13368 (100)	
* 1	TABLE ACCESS INMEMORY FULL	LINEORDER	4	68	13368 (2)	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
CPU used by this session	67
IM SubCU-MM CUs Examined	1
IM SubCU-MM CUs Selected	1
IM SubCU-MM SubCUs Eliminated	1064
IM SubCU-MM SubCUs in Selected CUs	1065
IM scan CUs columns accessed	3
IM scan CUs columns theoretical max	1887
IM scan CUs current	111
IM scan CUs memcompress for query low	111
IM scan CUs no cleanout	111
IM scan CUs pcode pred eevald	1
IM scan CUs predicates applied	111
IM scan CUs predicates optimized	110
IM scan CUs predicates received	111

```

IM scan CUs pruned 110
IM scan CUs readlist creation accumulated time 49
IM scan CUs readlist creation number 111
IM scan CUs split pieces 180
IM scan bytes in-memory 3223359891
IM scan bytes uncompressed 5748587884
IM scan delta - only base scan 111
IM scan rows 59986052
IM scan rows optimized 59441024
IM scan rows projected 1
IM scan rows valid 545028
IM scan segments minmax eligible 111
IM simd compare calls 3
IM simd decode symbol calls 3
IM simd decode unpack calls 3
IM simd decode unpack selective calls 3
parse time cpu 45
parse time elapsed 56
redo size 1312
session logical reads 462846
session logical reads - IM 450834
session pga memory 13887208
session pga memory max 13887208

```

37 rows selected.

```

SQL> @10_no-ime.sql
Connected.

```

Session altered.

Elapsed: 00:00:00.00

SQL>

```

SQL> -- In-Memory Column Store query

```

SQL>

```

SQL> Select lo_shipmode, sum(lo_ordtotalprice),
2          sum(lo_ordtotalprice - (lo_ordtotalprice*(lo_discount/100)) + lo_tax)
discount_price
3 From LINEORDER
4 group by
5 lo_shipmode;

```

LO_SHIPMOD	SUM(LO_ORDTOTALPRICE)	DISCOUNT_PRICE
FOB	161837922547928	153778178139883.79
SHIP	161872168268007	153814810364218.27
REG AIR	161864992173671	153805526566678.06
TRUCK	161820421065658	153761833420584.79
AIR	161811429297122	153751580704864.92
RAIL	161896622020617	153838322879643.73
MAIL	161844311638291	153787676162546.37

7 rows selected.

Elapsed: 00:00:14.68

SQL>

```

SQL> set echo off

```

Hit enter ...

PLAN_TABLE_OUTPUT

```

-----
SQL_ID g5rn8q8zp92tg, child number 0
-----

```

```

Select lo_shipmode, sum(lo_ordtotalprice),          sum(lo_ordtotalprice
- (lo_ordtotalprice*(lo_discount/100)) + lo_tax) discount_price From
LINEORDER group by lo_shipmode

```

Plan hash value: 3675673598

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				14769 (100)	
1	HASH GROUP BY		7	210	14769 (12)	00:00:01
2	TABLE ACCESS INMEMORY FULL	LINEORDER	59M	1716M	14644 (11)	00:00:01

16 rows selected.

Hit enter ...

NAME	VALUE
CPU used by this session	1493
IM scan CUs columns accessed	444
IM scan CUs columns theoretical max	1887
IM scan CUs current	111
IM scan CUs memcompress for query low	111
IM scan CUs no cleanout	111
IM scan CUs pcode aggregation pushdown	222
IM scan CUs readlist creation accumulated time	215
IM scan CUs readlist creation number	111
IM scan CUs split pieces	180
IM scan bytes in-memory	3223359891
IM scan bytes uncompressed	5748587884
IM scan delta - only base scan	111
IM scan dict engine results reused	222
IM scan rows	59986052
IM scan rows pcode aggregated	59986052
IM scan rows projected	777
IM scan rows valid	59986052
IM simd decode symbol calls	111
IM simd decode unpack calls	88647
IM simd decode unpack selective calls	444
parse time cpu	60
parse time elapsed	72
redo size	1312
session logical reads	466582
session logical reads - IM	450834
session pga memory	14083816
session pga memory max	15001320

28 rows selected.

```
SQL> @11_ime_usage.sql
Connected.
SQL>
SQL> -- This query displays what objects are in the In-Memory Column Store
SQL>
SQL> set echo off
```

Owner	Object	Partition Name	Column Name	Total IMEUs	Used Space(MB)
SSB	LINEORDER		V1	111	696

```
SQL>
SQL> --
SQL> -- Statement used to create the In-Memory Expression
SQL> --
SQL> -- alter table lineorder add v1 invisible as (lo_ordtotalprice -
(lo_ordtotalprice*(lo_discount/100)) + lo_tax);
```



```

SQL> --
SQL>
SQL> @i2_with-ime.sql
SQL> @./imlogin.sql
SQL> connect ssb/oracle12@orcl
Connected.
SQL>
SQL> set pages 9999
SQL> set lines 100
SQL>
SQL> set timing on
SQL> set echo on
SQL> set numwidth 20
SQL>
SQL> -- In-Memory Column Store query
SQL>
SQL> Select lo_shipmode, sum(lo_ordtotalprice),
      2          sum(lo_ordtotalprice - (lo_ordtotalprice*(lo_discount/100)) + lo_tax)
discount_price
      3 From    LINEORDER
      4 group by
      5    lo_shipmode;

```

LO_SHIPMOD	SUM(LO_ORDTOTALPRICE)	DISCOUNT_PRICE
FOB	161837922547928	153778178139883.79
SHIP	161872168268007	153814810364218.27
REG AIR	161864992173671	153805526566678.06
TRUCK	161820421065658	153761833420584.79
AIR	161811429297122	153751580704864.92
RAIL	161896622020617	153838322879643.73
MAIL	161844311638291	153787676162546.37

7 rows selected.

Elapsed: 00:00:07.05

```

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

```

PLAN_TABLE_OUTPUT

```

-----
SQL_ID  g5rn8q8zp92tg, child number 0
-----
Select lo_shipmode, sum(lo_ordtotalprice),          sum(lo_ordtotalprice
- (lo_ordtotalprice*(lo_discount/100)) + lo_tax) discount_price From
LINEORDER group by    lo_shipmode

```

Plan hash value: 3675673598

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				14769 (100)	
1	HASH GROUP BY		7	210	14769 (12)	00:00:01
2	TABLE ACCESS INMEMORY FULL	LINEORDER	59M	1716M	14644 (11)	00:00:01

16 rows selected.

Hit enter ...

NAME	VALUE

```

CPU used by this session                679
IM scan CUs columns accessed            444
IM scan CUs columns theoretical max     1887
IM scan CUs current                     111
IM scan CUs memcompress for query low   111
IM scan CUs no cleanout                 111
IM scan CUs pcode aggregation IME       111
IM scan CUs pcode aggregation pushdown  222
IM scan CUs readlist creation accumulated time 34628
IM scan CUs readlist creation number    111
IM scan CUs split pieces                 291
IM scan EU bytes in-memory              730204983
IM scan EU bytes uncompressed           412353699
IM scan EU rows                         59986052
IM scan EUs columns accessed            111
IM scan EUs columns theoretical max     1998
IM scan EUs memcompress for query low   111
IM scan EUs split pieces                 93
IM scan bytes in-memory                 3223359891
IM scan bytes uncompressed              5748587884
IM scan delta - only base scan          111
IM scan dict engine results reused      222
IM scan rows                           59986052
IM scan rows pcode aggregated           59986052
IM scan rows projected                   777
IM scan rows valid                      59986052
IM simd decode symbol calls             111
IM simd decode unpack calls             29919
IM simd decode unpack selective calls    444
parse time cpu                          1
parse time elapsed                       2
redo size                                684
session logical reads                   451091
session logical reads - IM              450834
session pga memory                      12642024
session pga memory max                  14149352

```

36 rows selected.

```

SQL> @i3_join_im.sql
Connected.
SQL>
SQL> -- In-Memory Column Store query
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
-----
          9710699495

```

```

Elapsed: 00:00:00.20
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				14612 (100)	
1	SORT AGGREGATE		1	43		
* 2	HASH JOIN		2085	89655	14612 (11)	00:00:01
3	JOIN FILTER CREATE	:BF0000	1	25	1 (0)	00:00:01
* 4	TABLE ACCESS INMEMORY FULL	DATE_DIM	1	25	1 (0)	00:00:01
5	JOIN FILTER USE	:BF0000	5017K	86M	14597 (11)	00:00:01
* 6	TABLE ACCESS INMEMORY FULL	LINEORDER	5017K	86M	14597 (11)	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
CPU used by this session	15
IM SubCU-MM CUs Examined	112
IM scan CUs columns accessed	445
IM scan CUs columns theoretical max	1904
IM scan CUs current	112
IM scan CUs memcompress for query low	112
IM scan CUs no cleanout	112
IM scan CUs pcode pred evaled	556
IM scan CUs predicates applied	334
IM scan CUs predicates received	334
IM scan CUs readlist creation accumulated time	202
IM scan CUs readlist creation number	112
IM scan CUs split pieces	181
IM scan bytes in-memory	3223461064
IM scan bytes uncompressed	5748842770
IM scan delta - only base scan	112
IM scan rows	59988608
IM scan rows projected	2132
IM scan rows valid	59988608
IM scan segments minmax eligible	1
IM simd bloom filter calls	111
IM simd compare calls	447
IM simd decode unpack calls	553
IM simd decode unpack selective calls	553
parse time cpu	1
parse time elapsed	3
redo size	684
session logical reads	451144
session logical reads - IM	450849
session pga memory	12379880
session pga memory max	12379880

31 rows selected.

SQL> @14_join_buffer.sql

Connected.

SQL>

SQL> -- Buffer Cache query with the column store disables via the inmemory_query parameter

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
-----
      9710699495
```

Elapsed: 00:00:10.49

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				123K(100)	
1	SORT AGGREGATE		1	43		
* 2	HASH JOIN		2085	89655	123K (1)	00:00:05
* 3	TABLE ACCESS FULL	DATE_DIM	1	25	15 (0)	00:00:01
* 4	TABLE ACCESS FULL	LINEORDER	5017K	86M	123K (1)	00:00:05

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
CPU used by this session	1009
IM scan segments disk	2
parse time elapsed	1
redo size	816

```

session logical reads          450996
session pga memory            12576488
session pga memory max        12576488

```

7 rows selected.

```

SQL> @15_query_jg.sql
Connected.
SQL>
SQL> -- This script will query Join Groups in the In-Memory Column Store
SQL>
SQL> set echo off

```

JOINGROUP_NAME	TABLE_NAME	COLUMN_NAME	GD_ADDRESS
LINEORDER_JG1	LINEORDER	LO_ORDERDATE	00000002EDE8DC50
LINEORDER_JG1	DATE_DIM	D_DATEKEY	00000002EDE8DC50
LINEORDER_JG2	LINEORDER	LO_PARTKEY	00000002EDE1DC50
LINEORDER_JG2	PART	P_PARTKEY	00000002EDE1DC50
LINEORDER_JG3	LINEORDER	LO_SUPPKEY	00000002EDE3DC50
LINEORDER_JG3	SUPPLIER	S_SUPPKEY	00000002EDE3DC50

6 rows selected.

```

SQL>
SQL> --
SQL> -- Statements used to create the In-Memory Join Groups
SQL> --
SQL> -- CREATE INMEMORY JOIN GROUP lineorder_jg1 ( lineorder(lo_orderdate),
date_dim(d_datekey) );
SQL> -- CREATE INMEMORY JOIN GROUP lineorder_jg2 ( lineorder(lo_partkey), part(p_partkey) );
SQL> -- CREATE INMEMORY JOIN GROUP lineorder_jg3 ( lineorder(lo_suppkey),
supplier(s_suppkey) );
SQL> --
SQL> @16_use_jg_xml.sql
Connected.
SQL>
SQL> -- In-Memory Column Store query
SQL>
SQL> select /*+ NO_VECTOR_TRANSFORM monitor */
2   d.d_year,
3   p.p_brand1,
4   sum(lo_revenue) rev
5 from
6   lineorder l,
7   date_dim d,
8   part p,
9   supplier s
10 where
11  l.lo_orderdate = d.d_datekey
12  and l.lo_partkey = p.p_partkey
13  and l.lo_suppkey = s.s_suppkey
14  and p.p_category = 'MFGR#12'
15  and s.s_region = 'AMERICA'
16  and d.d_year = 1997
17 group by
18  d.d_year,
19  p.p_brand1;

```

D_YEAR	P_BRAND1	REV
1997	MFGR#128	6597639363
1997	MFGR#1234	6695984533
1997	MFGR#1221	6892277556
1997	MFGR#126	6213627043
1997	MFGR#122	7101221796
1997	MFGR#121	6409702180
1997	MFGR#1224	7373166413

1997 MFGR#1236	6827320374
1997 MFGR#1222	6551372899
1997 MFGR#1217	7053041453
1997 MFGR#1225	6804217225
1997 MFGR#1231	6839363437
1997 MFGR#1213	6686343443
1997 MFGR#1220	6613283998
1997 MFGR#1210	6795372926
1997 MFGR#1215	7079477060
1997 MFGR#1223	6465041111
1997 MFGR#1239	6227318243
1997 MFGR#1227	6713851455
1997 MFGR#124	6817087386
1997 MFGR#1212	6573951551
1997 MFGR#1237	7041724061
1997 MFGR#1218	6841323272
1997 MFGR#1228	7038686432
1997 MFGR#125	6416763567
1997 MFGR#1214	6630127600
1997 MFGR#1211	6852509575
1997 MFGR#123	6280463233
1997 MFGR#1233	6914572704
1997 MFGR#1235	7075150948
1997 MFGR#1226	7185780867
1997 MFGR#1230	6596531127
1997 MFGR#1232	7404918843
1997 MFGR#1219	6651946261
1997 MFGR#129	6164317597
1997 MFGR#1216	6877592440
1997 MFGR#127	6765391794
1997 MFGR#1240	7056019394
1997 MFGR#1238	6443513085
1997 MFGR#1229	6481227038

40 rows selected.

Elapsed: 00:00:01.31

SQL>

SQL> set echo off

Hit enter ...

2 join group was leveraged on 1 processes
4 join group was leveraged on 1 processes
7 join group was leveraged on 1 processes

SQL>

SQL> @16_use_jg.sql

SQL> @../imlogin.sql

SQL> connect ssb/oracle12@orcl

Connected.

SQL>

SQL> set pages 9999

SQL> set lines 150

SQL>

SQL> set timing on

SQL> set echo on

SQL>

SQL> -- In-Memory Column Store query

SQL>

```
SQL> Select /*+ NO_VECTOR_TRANSFORM monitor */
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#128	6597639363
1997	MFGR#1234	6695984533
1997	MFGR#1221	6892277556
1997	MFGR#126	6213627043
1997	MFGR#122	7101221796
1997	MFGR#121	6409702180
1997	MFGR#1224	7373166413
1997	MFGR#1236	6827320374
1997	MFGR#1222	6551372899
1997	MFGR#1217	7053041453
1997	MFGR#1225	6804217225
1997	MFGR#1231	6839363437
1997	MFGR#1213	6686343443
1997	MFGR#1220	6613283998
1997	MFGR#1210	6795372926
1997	MFGR#1215	7079477060
1997	MFGR#1223	6465041111
1997	MFGR#1239	6227318243
1997	MFGR#1227	6713851455
1997	MFGR#124	6817087386
1997	MFGR#1212	6573951551
1997	MFGR#1237	7041724061
1997	MFGR#1218	6841323272
1997	MFGR#1228	7038686432
1997	MFGR#125	6416763567
1997	MFGR#1214	6630127600
1997	MFGR#1211	6852509575
1997	MFGR#123	6280463233
1997	MFGR#1233	6914572704
1997	MFGR#1235	7075150948
1997	MFGR#1226	7185780867
1997	MFGR#1230	6596531127
1997	MFGR#1232	7404918843
1997	MFGR#1219	6651946261
1997	MFGR#129	6164317597
1997	MFGR#1216	6877592440
1997	MFGR#127	6765391794
1997	MFGR#1240	7056019394
1997	MFGR#1238	6443513085
1997	MFGR#1229	6481227038

40 rows selected.

Elapsed: 00:00:01.18

SQL>

SQL> set echo off

Hit enter ...

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<script language="javascript" type="text/javascript">

<!--

var version = "18.0.0.0.0";

var swf_base_path = "http://download.oracle.com/otn_software/";

document.write('<script language="javascript" type="text/javascript" ' +
 'src="' + swf_base_path + 'emviewers/scripts/activeReportInit.js?' +
 Math.floor((new Date()).getTime()/(7*24*60*60*1000)) +
 "'></' + 'script>');

-->

nWNxhsGnDf2NpILOeLXMKjnqiAJqkDhaFgegfdYoYyCdG5c8EuZHGonK/FT0K+uq
001+rkhtqHs9Q8JRkiY4Lkm5wZ86yGAnKa5jhrrYAB2vL5qfahgRISddYIwYQ4
qaajx2QPEmfG6PuB8KpMycOpGQbC0+2wmGK5c48n5aqLc0m56njYX/UoJSExi4l0
J84v5TASdAIRZB1kYjg8nJM6b+WCEKHmg6eVaE0Bh+XTTxDwQB+Q+s5BZpxmofq6
k1rxItuQ2+8u21gcEnUppWQRo3yNerNpNLGZ/mpVo00EWBUg1LEqyCiLFH5x7vD1
wp84n8iV/syDioT8759B0KsoYtk4cMwAwmL5x3nI7kmIEnnUfLE0+5kYfiQnU8pw
UuCxPjn6gTQtvd4+pB+o6z8EVD1gS0oFLS4hsfbIQ7dSKQmcofQLQmV0t447ZxNk
RBKqD2YnGv1m3oLh7A/pRPTWWw+2LB60fFMcaigFiCeOIVTGSx0JAMnsSrLtaljE
BdqX5DL2HtgZpcIrvjC8tYiY+d5DwJCdHXNVbBp5jeN0SjR+Ya3f4fwy3ShMRMhV
P2BIWJs9YkFyBYO2R0wLrhVNjpcNhimYDNxvTBzoqw12nonjBujjPV3v3mYrAD0
hYPALL20PqXMQSzw4zbqtXqTeCsysWi6xLrE41N8cLcC+9wknzMLITdzdjw5toL
EgU13Dsbiz2Y5KLf0etLYTewYRTduEYKp/Nug6y+uBP4hczivCFHg4XlBxmxpFNV
jGIVtVXIh8HF5DRMNg4ScTEP6L7NIJ653IV8UsY6qQUN14YsidoQq7ktfGKv7ceZ
uwXYW2J9K0RkV94UdgmRPNGLM8/drtawzwrUetBgyIXEfffWtrzvEmxI1JXkXIt
kuvBgJSp06K/G8gmtFxoB66k6B6Gwz0jQ6I7uDpTpyjcpy3YBFxZSBkvYR2GukLhn
6E06/cuyocLYqMPnPAPkM04Z1a5xqiMxDBkHFAfLa6BxsIHmbedF0BCdr204cJs
F+F/GDwogcl/kw0tF7mReIja8IIDEiifGEsZ450nN3/eA27DiEsu4qeM6WEaSYZL
WgD/qaJ9+yH6ot1G1n8gf3AZGYKE6EJmKgGVK7B1IGEAZApqZeQc400LHouUEpK
SFJHhcbKS3dYLVXVCwWUzy/jfdrUoJAa5IGUaDsBj/+AJH0BoxT42QVJcPIJkn2I
Z0Rh10h5B66zdhCXy7d14e452mK2oib05C0WrmfBHRQYcai5cCoXP0cELk5CN9H7
ZPu+HW79NSisBbJTSn0IYdEVCyy4hHw06jUy7nEbb2KDPsxtz8IXRND0ArGiezYj
yJLPhF9GQC8W5K0BS1WnQjYRMuJgCBgsUZnh9prw7bnT98BvnV2N2IMFTELADXd7
n1hed6IWC/sVQaUz23XRMTd+vb9L4JTR1k0TYGYRHD0LBK/c9rmSeGJdwVtpvGM
WDklBxwHkrZD9ZrNFazi1QqnKYr508QewfGJjMA8qkVfAoY7tdn+a0GhZ/iDcdc
UgdEHLqumUd3T0Iuyob5dbsik30aqhzryaE8W248kWSV5a5hNI603SWszPL8aU46
FkxiXkIwvVJ09W0DFQivW8303QLBi2d9Lp4VQdYmT7BL+lasEPeMwp/shPEM3DPZ
j2qUcMXxiYnQw0L8ThvXLPYi5EKsnGuLdLws/LEbhQbfr9Xa7QmC/m8BPp3fxwA/
8rDZp9KqC1BaCjIpxCA+ER+SWP7jBNs0Duv3sHyUMvBNcD50L8yzPi0jnWeIReem
Zy/TkIqV2cLNKASRMjnp1Cq8fDE3z8XcpHW5hPGIc1uwlqgbx2KNwjSOP3XClbXJ
jN3xIKd4vQ/e64Xg/cJz50ALGc3LN0EyLjrNpZnHJYLwaD5gEztMY5g7ehPwL3eI
uClnoTBL7uYjMixiG3AGoAngfkt4ayZrBA4uoxdtRSIYL7h9QADpCFZ00ZiV4hPh
eMEHiBYbgWj7UxzbizdKYi55fZkkVjTGCKsL3tf0unJtPX4Dd3zGoAJXy/BdmC2u
kRR0cgtKp7gThvAgNfts1zPsmijAmRYdsoJJ0ieISMvi+8FjiFgUFgfHBbQ3xT6B
t17ikoM8S0jS8BD8PiHed7LCM1Ajj/etJzXSHbVIA+5dNe8Ztdb+cWksajYD13g0
xPWAG3CVBB1uecAcscG0tUqyQHqVoxPH7gsoGy0GALHmlLq7I4EAgYhiIPC1Eb44
R/4FFmgveQTfJY+A30hJamIjWBA5u6Iq0qUsxioWucL01nsHUH/F1nvnxkAe+M27
OgdTovbe+5cH6JUCwQfvB4BQAEHsewYgeAHDzwUM4yorPu0S+dXk8iw9Y++Nr0x/
9d6bX6xI6iUEKALbf/CtNwBUwixEek7U1rucx433KN1yW28ktX2Tj43uweQZn4wL
MwRcPPrIy06VGCpWniF35guIInuTyvrHyxdb81xsTUqRSzjYnX13pPX41s1hZF0G
ehtNBGry7AqQd6q7EKaoBwDe6R4ci7txZpHQ/ngzdQzuJuz+/XB3fJfrxSbQ5392
v44WZ50JT/mSmX50ZnpKyUrYhNKgmw7pEgpW4F751bas3WwWct/5BAUMQznfi0A
b9Kj3ftLeQ429+4c+inhfer2HL1SrvbgBTkQRu4286IdCF5c7nNxuZnwvoSKZbtd
HFPpdN01plcPBK2Qj1avt3C6u5/Sdqx710kKSbxpjo/cs+FxUn/LecV/0TZ+0d9Q
XGN7ubiVe30scQeKPhmYsT3//vrbqLbbrWYN960T+Pho/YXjT0pvL0B2V0ooRc6+
H8BNXBh+8cy+Moh8n/NMrmTmPE4LxB1fycw5ZpHA/PldyfXazY7AL3BWdCNYfr64
4NvC2GAWffTytSe/XCyv/UJrnZ0bXht3hQVsVJeJM24N730IOP0jv3wnpYbdjvrs
Mq+a+ellzNDu1VZxu1pt4nqvYN5w71bu6XCLmv4fHL1X/vwk0vRb

```
</report>  
<!--FXTMODEL-->  
</script>  
</body>  
</html>
```

```
SQL>  
SQL> @17_vgb_im.sql  
Connected.  
SQL> set numwidth 20  
SQL>  
SQL> -- In-Memory Column Store query with In-Memory Aggregation enabled  
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	257114232147
1992	BRAZIL	265087838628
1992	CANADA	267029429200
1992	PERU	260091746389
1992	UNITED STATES	264098681570
1993	ARGENTINA	260475234675
1993	BRAZIL	264646147899
1993	CANADA	265113570471
1993	PERU	262628508817
1993	UNITED STATES	260411002935
1994	ARGENTINA	261149015641
1994	BRAZIL	263808033983
1994	CANADA	264598150413
1994	PERU	258595600981
1994	UNITED STATES	265282504206
1995	ARGENTINA	258498976118
1995	BRAZIL	269135848643
1995	CANADA	264654265482
1995	PERU	257451709833
1995	UNITED STATES	259660457396
1996	ARGENTINA	259361903850
1996	BRAZIL	265970119048
1996	CANADA	265333193889
1996	PERU	2600916013039
1996	UNITED STATES	262339293224
1997	ARGENTINA	261099548066
1997	BRAZIL	266353055971
1997	CANADA	265036379243
1997	PERU	259114682243
1997	UNITED STATES	262208356128
1998	ARGENTINA	151054449013
1998	BRAZIL	153632348378
1998	CANADA	156899052279
1998	PERU	152297126350
1998	UNITED STATES	153937088695

35 rows selected.

Elapsed: 00:00:02.03

SQL>

SQL> set echo off

Hit enter ...

PLAN_TABLE_OUTPUT

SQL_ID czrxzt1nt4vpg, 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: 2827393718

Id	Operation	Name	Rows	Bytes
Cost (%CPU)	Time			
0	SELECT STATEMENT			
15523 (100)				
1	TEMP TABLE TRANSFORMATION			
2	LOAD AS SELECT (CURSOR DURATION MEMORY)	SYS_TEMP_0FD9D664D_645B04		
3	HASH GROUP BY		7	98
2 (50)	00:00:01			
4	KEY VECTOR CREATE BUFFERED	:KV0000	7	98
1 (0)	00:00:01			
5	TABLE ACCESS INMEMORY FULL	DATE_DIM	2556	25560
1 (0)	00:00:01			
6	LOAD AS SELECT (CURSOR DURATION MEMORY)	SYS_TEMP_0FD9D664E_645B04		
7	HASH GROUP BY		1	9
96 (27)	00:00:01			
8	KEY VECTOR CREATE BUFFERED	:KV0001	1	9
75 (6)	00:00:01			
9	TABLE ACCESS INMEMORY FULL	PART	800K	3906K
74 (5)	00:00:01			
10	LOAD AS SELECT (CURSOR DURATION MEMORY)	SYS_TEMP_0FD9D664F_645B04		
11	HASH GROUP BY		1	22
4 (25)	00:00:01			
12	KEY VECTOR CREATE BUFFERED	:KV0002	1	22
3 (0)	00:00:01			
* 13	TABLE ACCESS INMEMORY FULL	SUPPLIER	4102	73836
3 (0)	00:00:01			
14	LOAD AS SELECT (CURSOR DURATION MEMORY)	SYS_TEMP_0FD9D6650_645B04		
15	HASH GROUP BY		25	950
39 (21)	00:00:01			
16	KEY VECTOR CREATE BUFFERED	:KV0003	25	950
36 (14)	00:00:01			
* 17	TABLE ACCESS INMEMORY FULL	CUSTOMER	59761	1984K
36 (14)	00:00:01			
18	SORT GROUP BY		62	6386
15381 (15)	00:00:01			
* 19	HASH JOIN		62	6386
15380 (15)	00:00:01			
* 20	HASH JOIN		62	4588
15378 (15)	00:00:01			
21	MERGE JOIN CARTESIAN		7	315
6 (0)	00:00:01			
22	MERGE JOIN CARTESIAN		1	31
4 (0)	00:00:01			
23	TABLE ACCESS FULL	SYS_TEMP_0FD9D664E_645B04	1	9
2 (0)	00:00:01			
24	BUFFER SORT		1	22
2 (0)	00:00:01			
25	TABLE ACCESS FULL	SYS_TEMP_0FD9D664F_645B04	1	22
2 (0)	00:00:01			
26	BUFFER SORT		7	98
4 (0)	00:00:01			
27	TABLE ACCESS FULL	SYS_TEMP_0FD9D664D_645B04	7	98
2 (0)	00:00:01			
28	VIEW	VW_VT_80F21617	62	1798
15372 (15)	00:00:01			
29	VECTOR GROUP BY		62	2976
15372 (15)	00:00:01			
30	HASH GROUP BY		62	2976
15372 (15)	00:00:01			
31	KEY VECTOR USE	:KV0000	3642K	166M
15371 (15)	00:00:01			

```

| 32 |          KEY VECTOR USE          | :KV0001          | 3642K| 152M|
15370 (15)| 00:00:01 |
| 33 |          KEY VECTOR USE          | :KV0003          | 3642K| 138M|
15370 (15)| 00:00:01 |
| 34 |          KEY VECTOR USE          | :KV0002          | 12M| 419M|
15370 (15)| 00:00:01 |
|* 35 |          TABLE ACCESS INMEMORY FULL | LINEORDER          | 59M| 1830M|
15369 (15)| 00:00:01 |
| 36 |          TABLE ACCESS FULL          | SYS_TEMP_0FD9D6650_645B04 | 25 | 725 |
2 (0)| 00:00:01 |
-----

```

Predicate Information (identified by operation id):

```

-----
13 - inmemory("S"."S_REGION"='AMERICA')
      filter("S"."S_REGION"='AMERICA')
17 - inmemory("C"."C_REGION"='AMERICA')
      filter("C"."C_REGION"='AMERICA')
19 - access("ITEM_14"=INTERNAL_FUNCTION("C0"))
20 - access("ITEM_12"=INTERNAL_FUNCTION("C0") AND "ITEM_13"=INTERNAL_FUNCTION("C0") AND
      "ITEM_15"=INTERNAL_FUNCTION("C0"))
35 - 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

76 rows selected.

```

SQL> @18_novgb_buffer.sql
Connected.
SQL>
SQL> alter session set inmemory_query = disable;

```

Session altered.

Elapsed: 00:00:00.00

```

SQL>
SQL> -- Buffer Cache query with the column store disabled via the inmemory_query parameter
SQL>

```

```

SQL> SELECT /*+ NO_VECTOR_TRANSFORM */
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	257114232147
1992	BRAZIL	265087838628
1992	CANADA	267029429200
1992	PERU	260091746389

```

1992 UNITED STATES      264098681570
1993 ARGENTINA          260475234675
1993 BRAZIL             264646147899
1993 CANADA             265113570471
1993 PERU               262628508817
1993 UNITED STATES     260411002935
1994 ARGENTINA          261149015641
1994 BRAZIL             263808033983
1994 CANADA             264598150413
1994 PERU               258595600981
1994 UNITED STATES     265282504206
1995 ARGENTINA          258498976118
1995 BRAZIL             269135848643
1995 CANADA             264654265482
1995 PERU               257451709833
1995 UNITED STATES     259660457396
1996 ARGENTINA          259361903850
1996 BRAZIL             265970119048
1996 CANADA             265333193889
1996 PERU               260916013039
1996 UNITED STATES     262339293224
1997 ARGENTINA          261099548066
1997 BRAZIL             266353055971
1997 CANADA             265036379243
1997 PERU               259114682243
1997 UNITED STATES     262208356128
1998 ARGENTINA          151054449013
1998 BRAZIL             153632348378
1998 CANADA             156899052279
1998 PERU               152297126350
1998 UNITED STATES     153937088695

```

35 rows selected.

Elapsed: 00:00:22.55
SQL>
SQL> set echo off
Hit enter ...

PLAN_TABLE_OUTPUT

```

-----
SQL_ID  bgm86vz7ghjsj, child number 1
-----
SELECT /*+ NO_VECTOR_TRANSFORM */          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: 3639618634

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT					179K (100)	
1	SORT GROUP BY		124	12276		179K (1)	00:00:08
* 2	HASH JOIN		3642K	343M		179K (1)	00:00:08
3	TABLE ACCESS FULL	DATE_DIM	2556	25560		15 (0)	00:00:01
* 4	HASH JOIN		3642K	309M	12M	179K (1)	00:00:08
5	TABLE ACCESS FULL	PART	800K	3906K		1891 (1)	00:00:01
* 6	HASH JOIN		3642K	291M	2688K	160K (1)	00:00:07
* 7	TABLE ACCESS FULL	CUSTOMER	59761	1984K		838 (1)	00:00:01

* 8	HASH JOIN		12M	582M		123K	(1)	00:00:05
* 9	TABLE ACCESS FULL	SUPPLIER	4102	73836		69	(0)	00:00:01
10	TABLE ACCESS FULL	LINEORDER	59M	1830M		123K	(1)	00:00:05

Predicate Information (identified by operation id):

2 - access("L"."LO_ORDERDATE"="D"."D_DATEKEY")
4 - access("L"."LO_PARTKEY"="P"."P_PARTKEY")
6 - access("L"."LO_CUSTKEY"="C"."C_CUSTKEY")
7 - filter("C"."C_REGION"='AMERICA')
8 - access("L"."LO_SUPPKEY"="S"."S_SUPPKEY")
9 - filter("S"."S_REGION"='AMERICA')

40 rows selected.

SQL>