Oracle Database In-Memory

How To Identify The Right Workload

May 15-16, 2019

Andy Rivenes
Database In-Memory Product Management
Oracle Corporation

Email: andy.rivenes@oracle.com
Twitter: @TheInMemoryGuy
Blog: blogs.oracle.com/in-memory

ORACLE

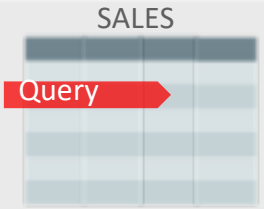
Safe Harbor Statement

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.

Database In-Memory: A Quick Recap

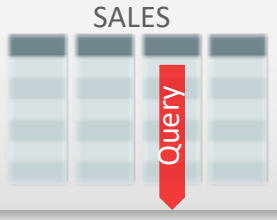
Row Format Databases vs. Column Format Databases

Rows Stored Contiguously



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

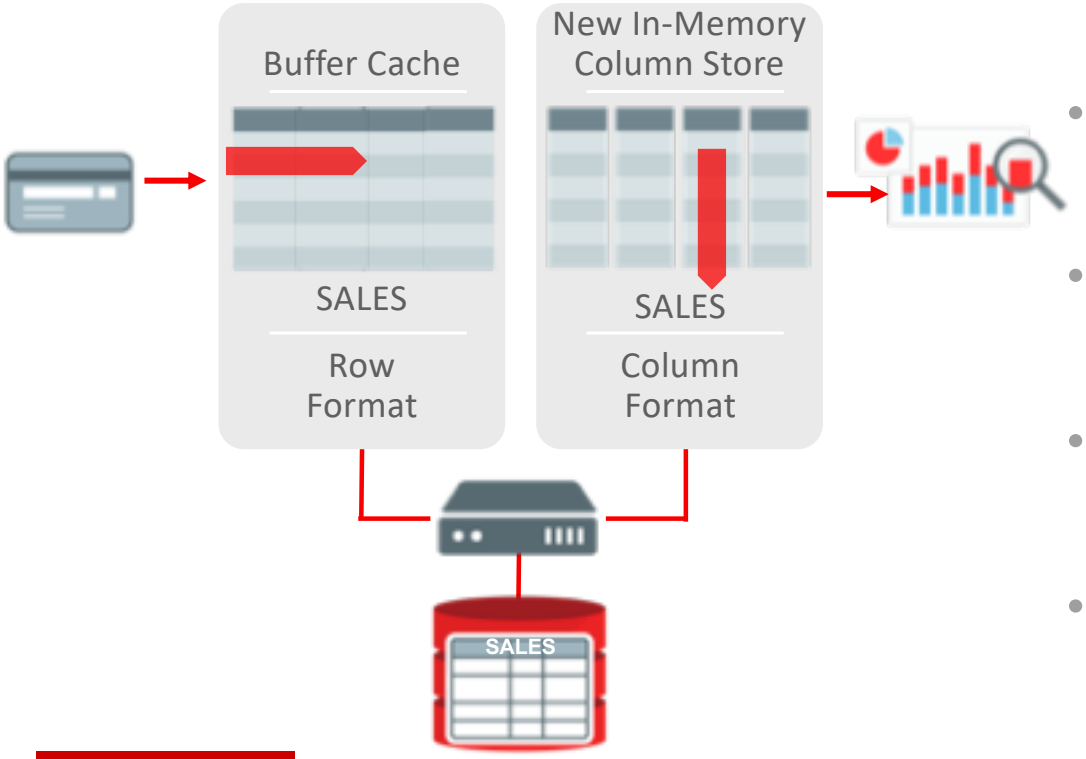
Columns Stored Contiguously



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

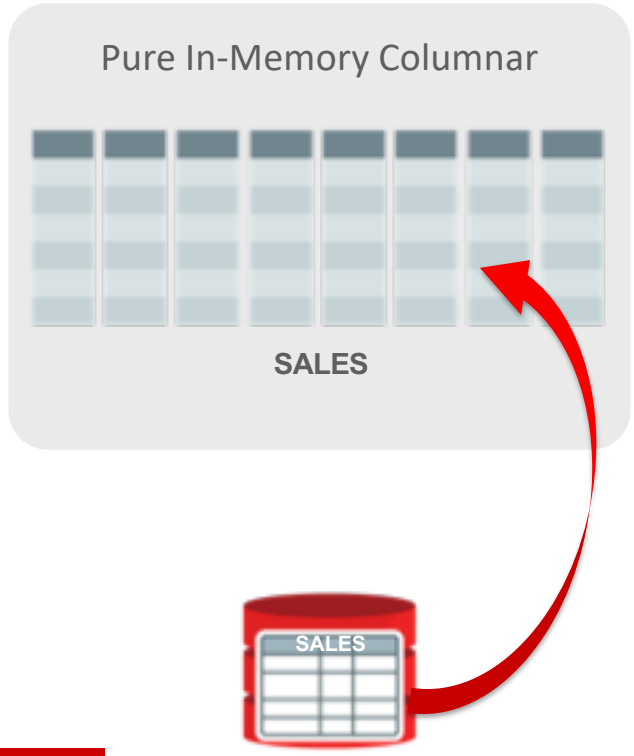
Until Now Must Choose One Format and Suffer Tradeoffs

Breakthrough: Dual Format Database



- **BOTH** row and column formats for same table
- Simultaneously active and transactionally consistent
- Analytics & reporting use new in-memory Column format
- OLTP uses proven row format

Oracle In-Memory Columnar Technology



- Pure in-memory column format
  - Enable for subset of database
  - Cheap to maintain – no logging or IO
  - Allows efficient OLTP
  - No change to disk format
- Built **seamlessly** into Oracle Database
  - Appears as a new storage type
  - **Transparent** to Applications
  - All Enterprise Features work ..
    - Availability – RAC, Flashback, DataGuard, etc.
    - Security – Encryption, Auditing, etc.

In-Memory Columnar Technology

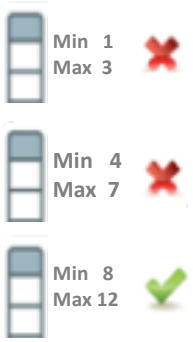
Columnar Format

Access only the columns you need

Compression

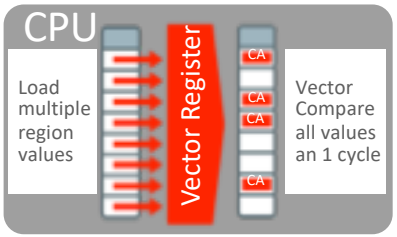
Scan & filter data in compressed format

Storage Indexes



Prune out any unnecessary data from the column

SIMD Vector Processing



Process multiple column values in a single CPU instruction

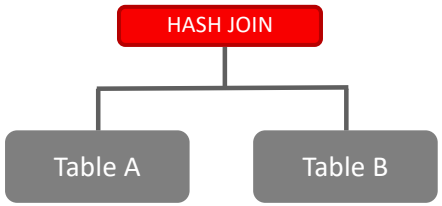
Optimizer Enhancements

Improves key aspects of analytic queries

Data Scans

- Speed of memory
- Scan and filter only the needed columns
- Vector Instructions

Joins



- Converts star joins into 10X faster column scans
- Scans large table for values that match the smaller table(s)

In-Memory Aggregation

- Creates an In-Memory aggregate accumulator
- Aggregates data during the fact table scan
- Runs Group By aggregations 3-8x faster than non-IMA scans

Database In-Memory New Features

Performance

- In-Memory Expressions
- Join Groups
- In-Memory Dynamic Scans
- In-Memory Optimized Arithmetic

Managability

- Automatic Data Optimization
- Automatic In-Memory

Expanded Capacity

- Exadata Flash
- Active Data Guard
- External Tables

Why Use Database In-Memory

Improved Reporting Performance

Faster Reports – No Application Changes

- Organizations can use Oracle reporting/analytical applications or existing 3rd party reporting tools
  - No application or data format changes required
- Improves performance (**10x typical**) of reporting applications with existing data warehouse and/or data marts
- Improves performance to ensure SLA's continue to be maintained
- Increases capacity of mixed workload environments to enable additional growth and performance

General Mills

- Using Database In-Memory resulted in:
  - **Triple the volume of Data**
  - **No changes required to Business Objects reports**
  - **50X performance improvement on reports**
    - Reports that took days now return in less than 1 hour

ORACLE

Real-Time Analytics

Use Operational Data for Real-Time Analytics

- Real-time analytics on operational data directly -- without the time delay of moving data for reporting
- Enables real-time business intelligence at the point of contact
  - Delivers real-time insight, visibility and agility for critical business operations and decisions
- Enables real-time ad-hoc reporting /analysis and iterative drill-downs on operational data
- No application or data format changes required

Pricerite實惠

- Using Database In-Memory resulted in:
  - **Analytic queries up to 5x faster**
  - **Real-time analytics dashboard**

ORACLE

Reduced Overhead

Faster Analytics -- Less Storage Overhead

- Analytic indexes can slow down the performance of transactional applications
  - Requires significantly more database storage (on costly tier 1 storage)
  - Increases overhead due to index maintenance
- Database In-Memory allows users to eliminate analytic reporting indexes – without impacting performance
- Removing the need for analytic reporting indexes greatly simplifies tuning and reduces ongoing administration

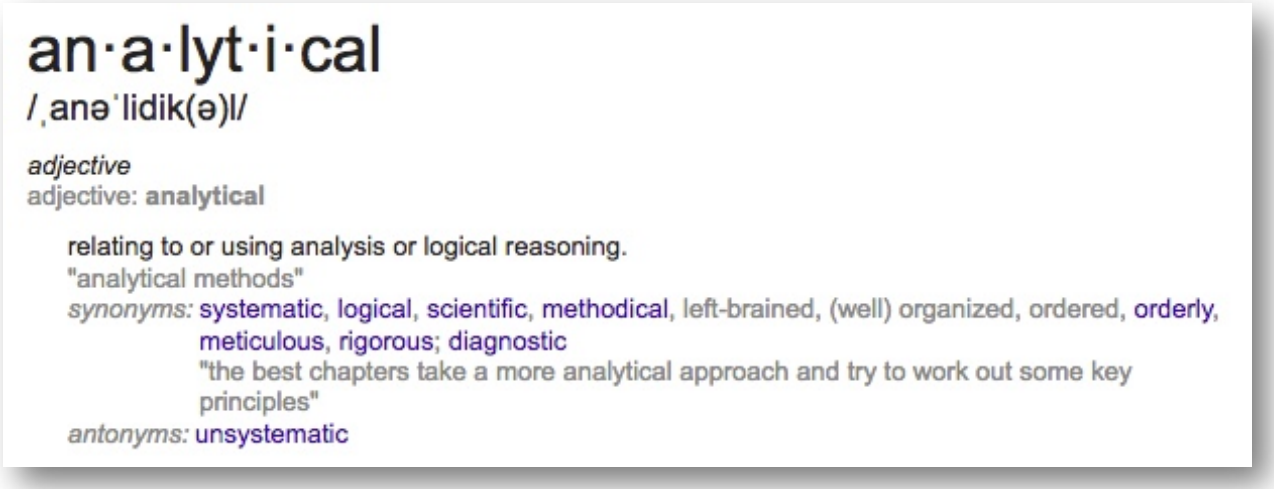
Walgreens

- Using Database In-Memory resulted in:
  - **Performance Gains: 1.8X to 12X**
  - **Space savings and reduced contention on DML by dropping analytic indexes**

ORACLE

How to Identify Analytic Workloads

What are Analytics?



an·a·lyt·i·cal

/ˌanəˈlidik(ə)l/

adjective

adjective: **analytical**

relating to or using analysis or logical reasoning.

"analytical methods"

synonyms: systematic, logical, scientific, methodical, left-brained, (well) organized, ordered, orderly, meticulous, rigorous; diagnostic

"the best chapters take a more analytical approach and try to work out some key principles"

antonyms: unsystematic

Source: Google Search

- Our definition: Using aggregation to find patterns and trends in the data

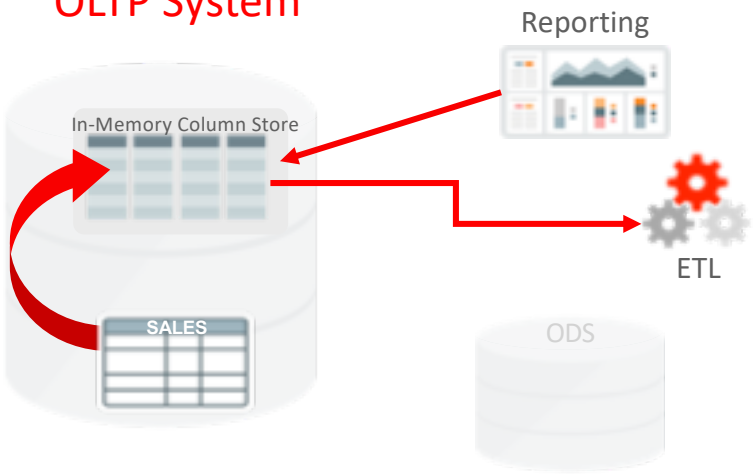
What is an analytic query?

Only **analytic queries** benefit from accessing the IM column store

When and Where Should I Use Database In-Memory

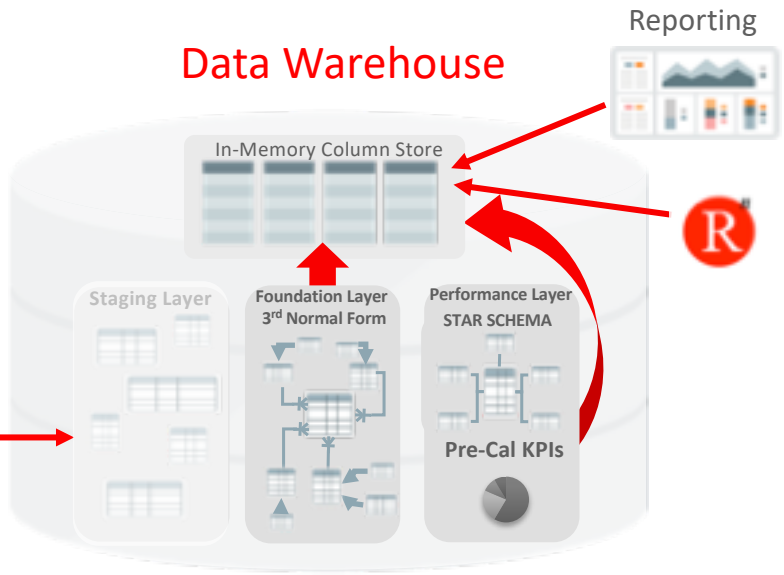
Where to use In-Memory

OLTP System



- Enables real-time reporting directly on OLTP data
- Speeds data extraction part of ETL process
- Removes need for separate ODS
- Speeds up mixed workload

Data Warehouse



- Star-schema and pre-calculated KPIs
  - Improves performance of dash-boards
- All or a subset of Foundation Layer
  - For time-sensitive analytics on 3rd normal form
- Staging/ETL/Temp not good candidates
  - Write once, read once

Database In-Memory - Use Cases

Mixed Workload

Real-Time Analytics

Reporting/BI

Which Objects Should I Populate

Oracle In-Memory Advisor

Workload Database Usage

Total Database Time (Seconds)	Analytics Processing Time (Seconds)	Analytics Processing Percentage
2990	2640	88%

In-Memory Size	Percentage of Maximum SGA Size (100.0GB)	Estimated Analytics Processing Time Reduction (Seconds)	Estimated Analytics Processing Performance Improvement Factor
9.141GB	**9%**	**2102**	**4.9X**
8.684GB	9%	2101	4.9X
8.226GB	8%	2101	4.9X
7.769GB	8%	2100	4.9X

- In-Memory Advisor – free download available on OTN for 11.2.0.3+ DBs
- Analyzes existing DB workload via AWR & ASH repositories
- Provides list of objects that would benefit most from being populated into IM column store

Note: Database Tuning Pack license required

Oracle In-Memory Advisor

SQL Id	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Time Reduction (Seconds) With Unlimited Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 9.141GB	Estimated Analytics Processing Performance Improvement Factor With 9.141GB
fp83uwmbzt8zd	select cf.custid, sum(act.purchase_amt) sales from all_card_trans act, cust_fact cf ...	990	696	3.4X	696	3.4X
7zkhj3xhq01w8	with gold_member_aff_cust as (select custid, aff_cc_num from cust_fact w...	940	660	3.4X	660	3.4X
8p8ggufpp7699	with act as (select act.card_no, act.purchase_amt from all_card_trans act ,mcc m, zipcodes z...	710	450	2.7X	450	2.7X

Object Type	Object	Compression Type	Estimated In-Memory Size	Analytics Processing Seconds	Estimated Reduced Analytics Processing Seconds	Estimated Analytics Processing Performance Improvement Factor	Benefit / Cost Ratio (Reduced Analytics Processing / In-Memory Size)
TABLE	TEST_UNCOMP.ZIPCODES	Memory compress for query low	1.063MB	50	33	3.0X	507741 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5598.SYS_SUBP5592	Memory compress for query low	1.063MB	1	0	3.0X	36330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5598.SYS_SUBP5593	Memory compress for query low	1.063MB	1	0	3.0X	36330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_MERCHANT_SALES.SYS_P5620.SYS_SUBP5615	Memory compress for query low	1.063MB	1	0	3.0X	28577 : 1

- Multiple sections available
  - In-Memory Size
  - SQL Statements with Analytic Benefit
  - Top object recommendations
  - All object based on memory size
  - Recommendation Rationale
  - Implementation SQL

How Much Memory Do I Need

Oracle Compression Advisor And In-Memory

```
DECLARE
    l_blkcnt_cmp        PLS_INTEGER;
    l_blkcnt_uncmp      PLS_INTEGER;
    l_row_cmp           PLS_INTEGER;
    l_row_uncmp         PLS_INTEGER;
    cmp_ratio           PLS_INTEGER;
    l_comptype_str      VARCHAR2(100);
    comp_ratio_allrows  NUMBER := -1;
BEGIN
    dbms_compression.Get_compression_ratio (
    scratchtbsname => 'TS_DATA',
    ownname        => 'SSB',
    objname        => 'LINEORDER',
    subobjname     => NULL,
    comptype       => dbms_compression.comp_inmemory_query_low,
    blkcnt_cmp     => l_blkcnt_cmp,
    blkcnt_uncmp   => l_blkcnt_uncmp,
    row_cmp        => l_row_cmp,
    row_uncmp      => l_row_uncmp,
    cmp_ratio      => cmp_ratio,
    comptype_str   => l_comptype_str,
    subset_numrows => dbms_compression.comp_ratio_allrows);
    dbms_output.Put_line('The IM compression ratio is '|| cmp_ratio);
END;
/
```

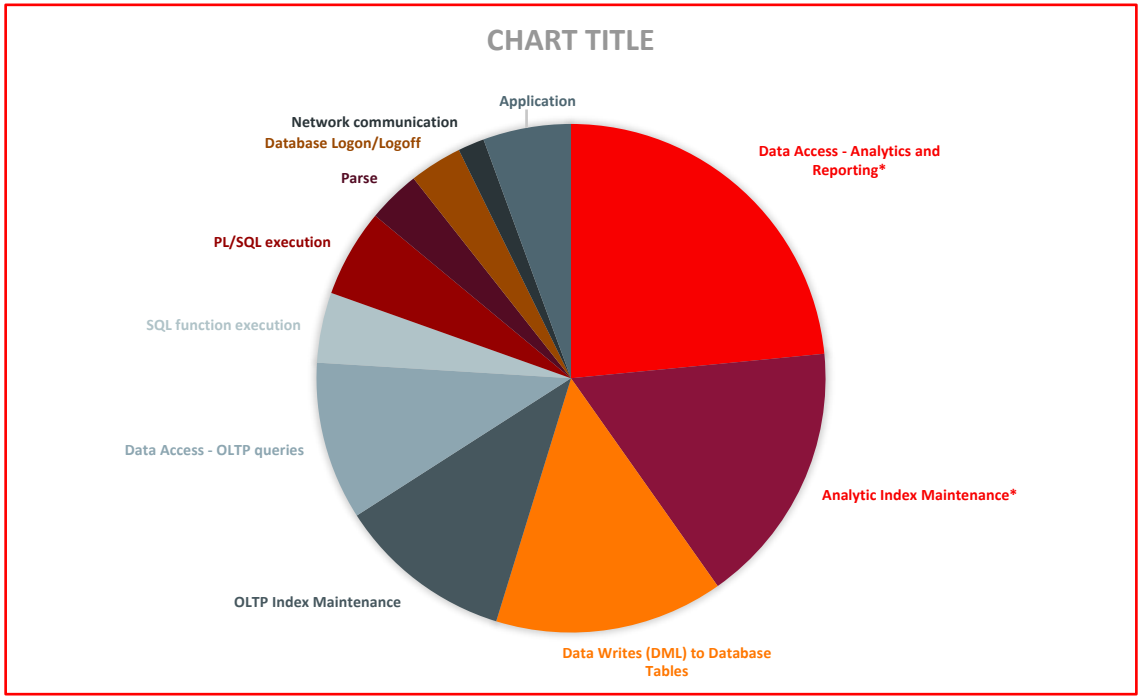
- Easy way to determine memory requirements
- Uses DBMS_COMPRESSION
- Applies MEMCOMPRESS to sample set of data from a table
- Returns estimated compression ratio
- Requires 12.1.0.2 or higher

What Queries Benefit From Database In-Memory?

Areas that Benefit from Database In-Memory

- Data Access for Analytics and Reporting
- Analytic Index Maintenance
- More Information:

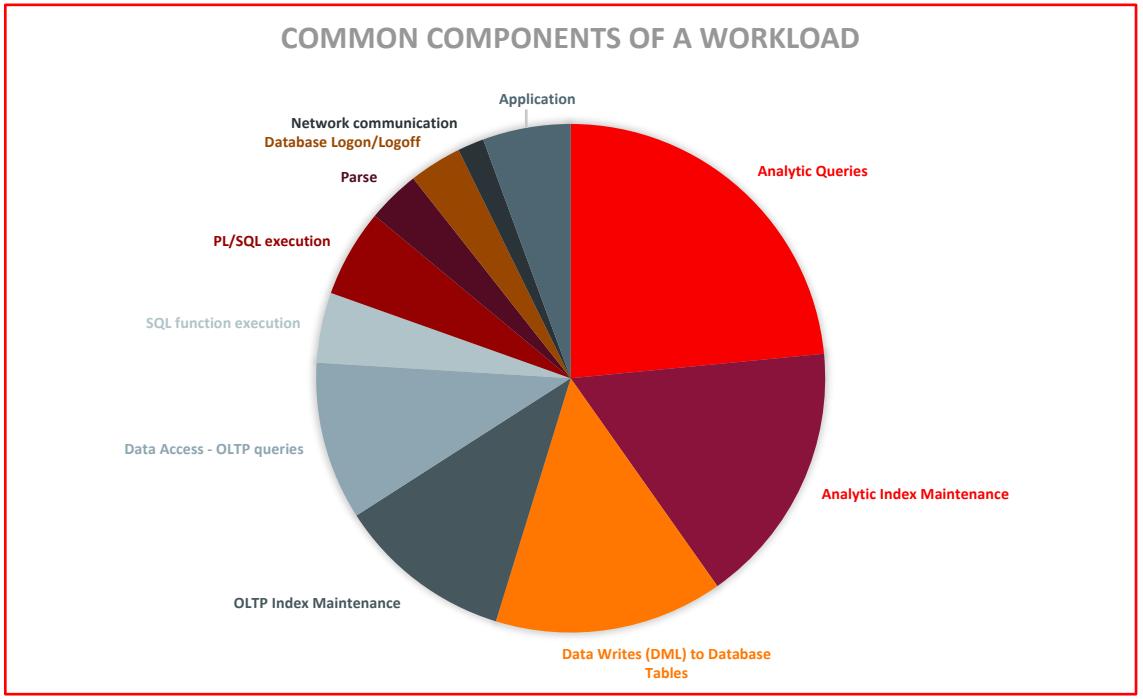
  When to Use Oracle Database In-Memory



Abstract Time Profile for a Typical Application (* - time potentially reduced by Database In-Memory)

Areas that DON'T Benefit from Database In-Memory

- Application Time
- Network communication
- Logon and Logoff
- Parse Time
- PL/SQL functions
- OLTP Queries
- OLTP index maintenance
- Writes to Database Tables



What SQL Techniques Can Increase The Benefit of Database In-Memory?

ORACLE

SQL Techniques That Can Improve Benefit

- Return as few rows as possible
- Limit the number of columns accessed
- Use selective column predicates
- Use selective join conditions
- Limit the number of tables being joined
- Avoid complex SQL functions

What SQL Techniques Can Reduce the Benefit of Database In-Memory?

ORACLE

Hints can prevent the Optimizer from Choosing In-Memory

- Hints that force the Optimizer to choose a join method
- Hints that force the Optimizer to use an Index
- Hints that result in the creation of temporary tables (do not benefit from Database In-Memory)
- Hints that disable In-Memory access
- Hints that disable features (i.e. NO_PARALLEL, NO_REWRITE)

SQL Techniques That Can Reduce Benefit

- Sub-query factoring (WITH) – Can prevent optimial scan filtering – typically used as a generic row source (MVs with query rewrite may be a better choice)
- Use of Function-based Indexes – prevents predicate push down, has to be evaluated after the scan
- Common views – often accesses more data than needed for the query
- Nested Views – can be difficult for the Optimizer to efficiently unnest
- Correlation issues – unless using extended statistics, the Optimizer may calculate the wrong cardinality

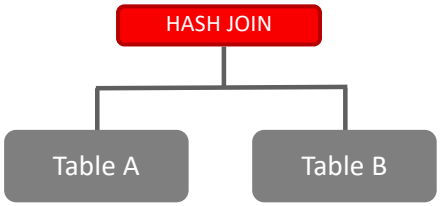
What Query Characteristics Benefit From Database In-Memory?

Which Queries Benefit From Database In-Memory?

For a non-trivial amount of rows and execution time, when a significant amount of time …

is spent accessing data

is spent joining data

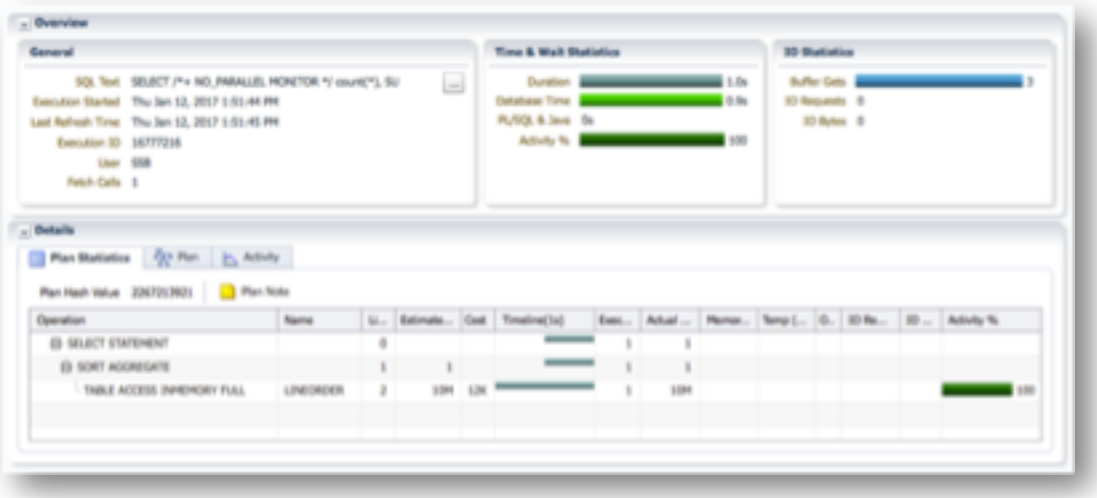


is spent aggregating data

Use Time Based Analysis Techniques To Evaluate Benefit

SQL Monitor Active Reports

- Shows how SQL was executed and where **time was spent**
- See blogs.oracle.com/In-Memory for a technical brief on creating SQL Monitor active reports



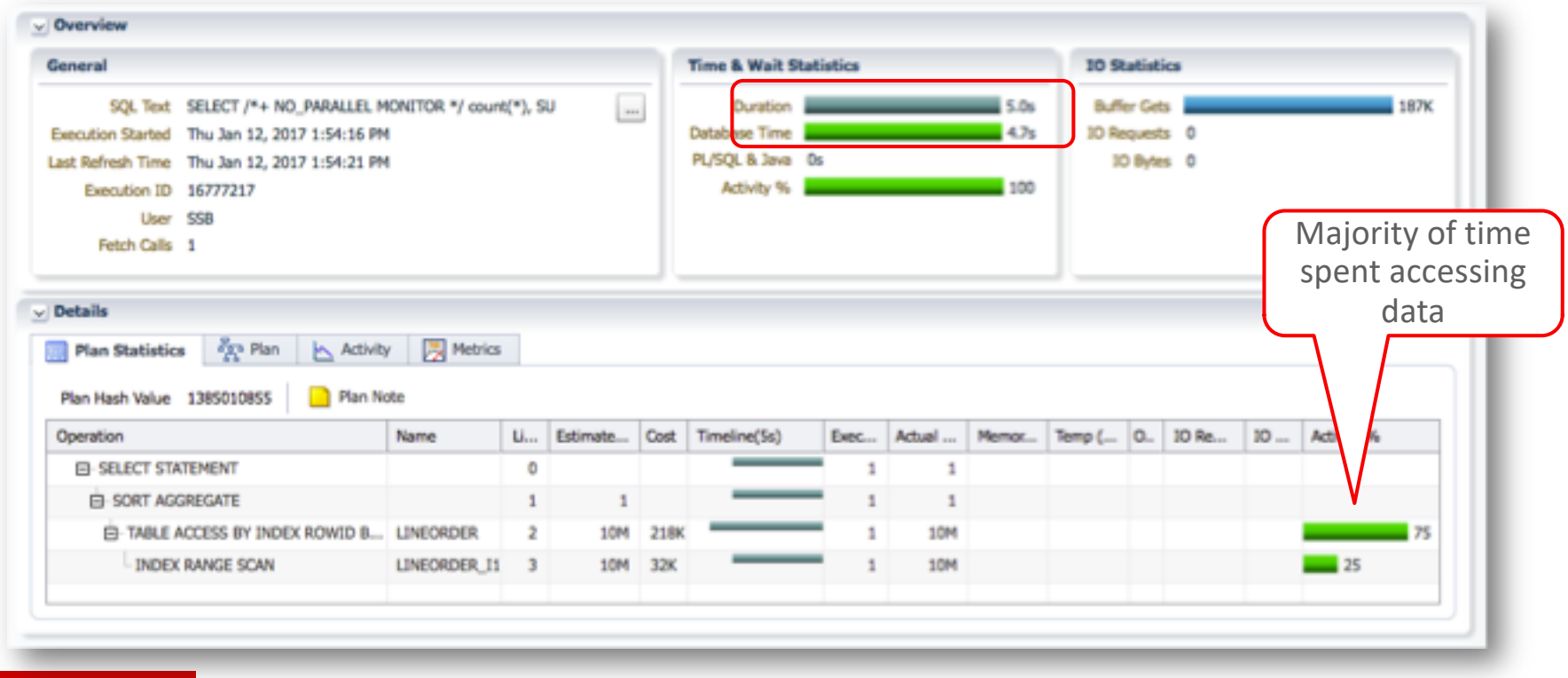
Data Scans

Scanning & Filtering Query

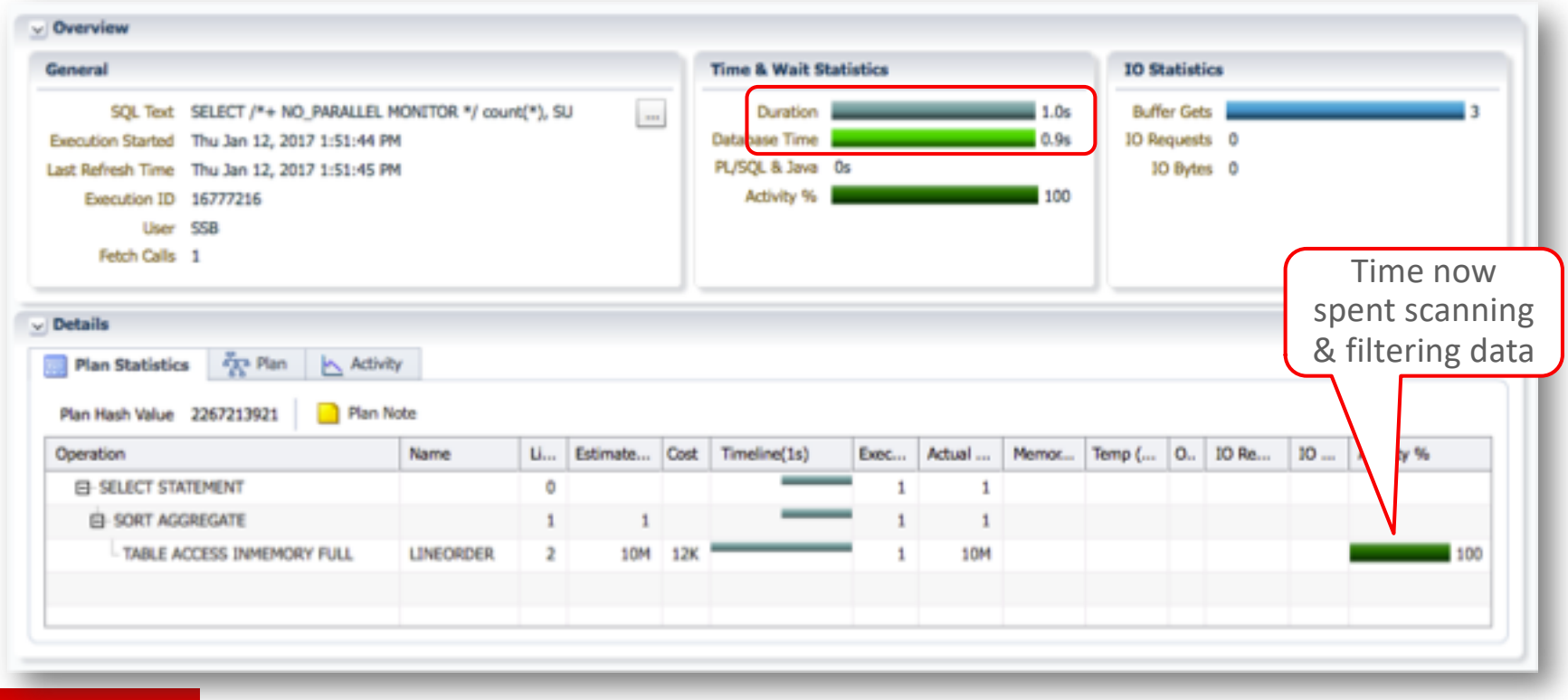
- Query to list total number of orders and the total value of merchandise shipped by air.

```
SELECT
   count(*),
   SUM(l.lo_ordtotalprice)
FROM  lineorder l
WHERE l.lo_shipmode = 'AIR';
```

Scanning & Filtering Query: Traditional data access



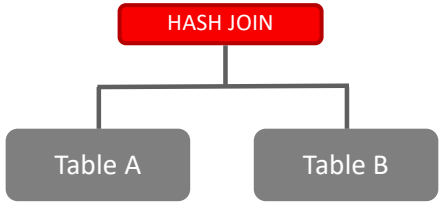
Scanning & Filtering Query: Scan & filter data in-memory



Things To Remember With Scans

- Queries that spend a significant amount of time accessing data are good candidates for Database In-Memory
- What Queries benefit most?
  - Queries that select just a subset of the columns from a table
  - Queries with selective where clause filter predicates
  - Queries with equality predicates, in-lists or range predicates work best
  - Queries with like or not exists predicates don't see as much benefit
- 12.2 Adds
  - In-Memory Expressions – Avoids repeated expression evaluation
  - More predicate push down operations – improves scan performance

Joins

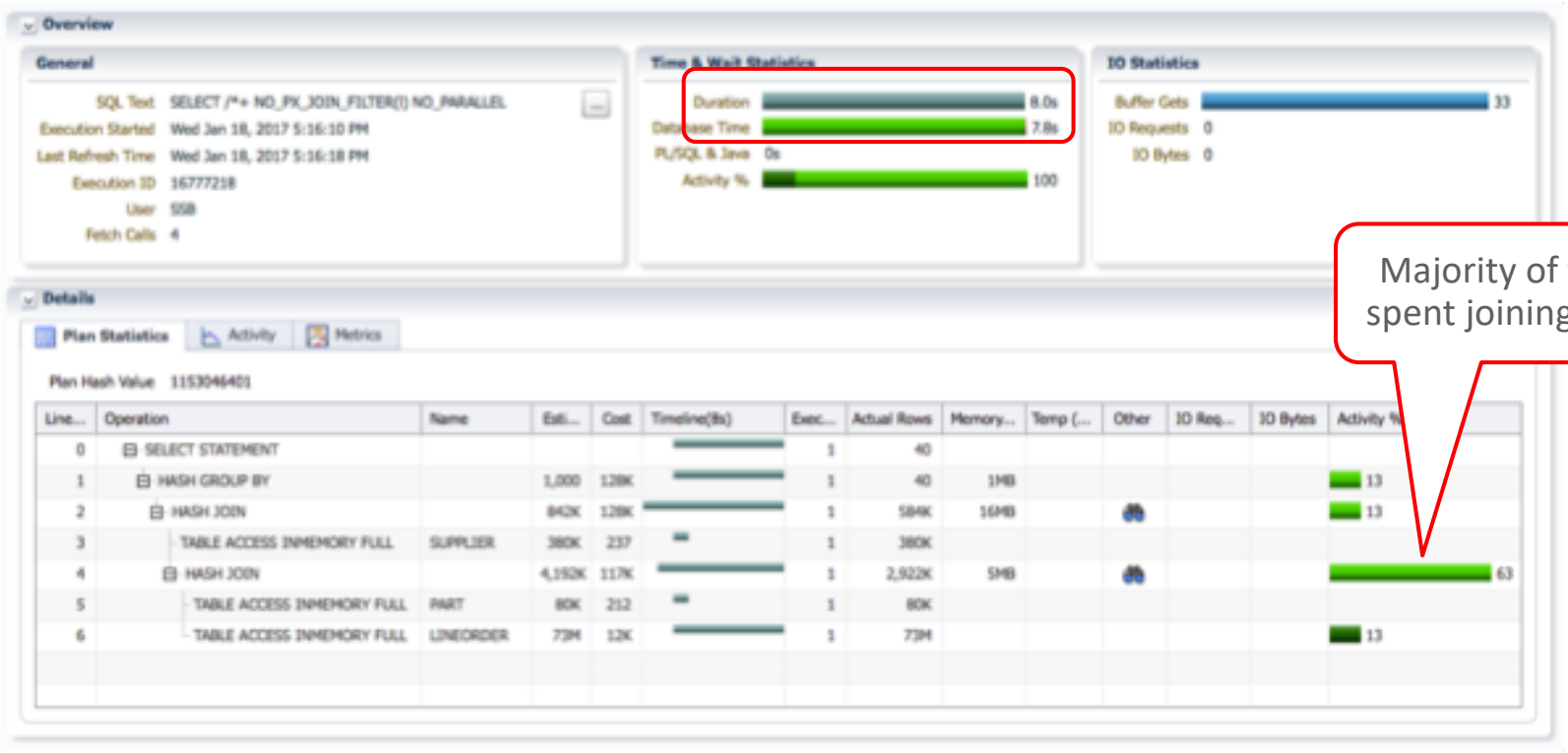


Join Query

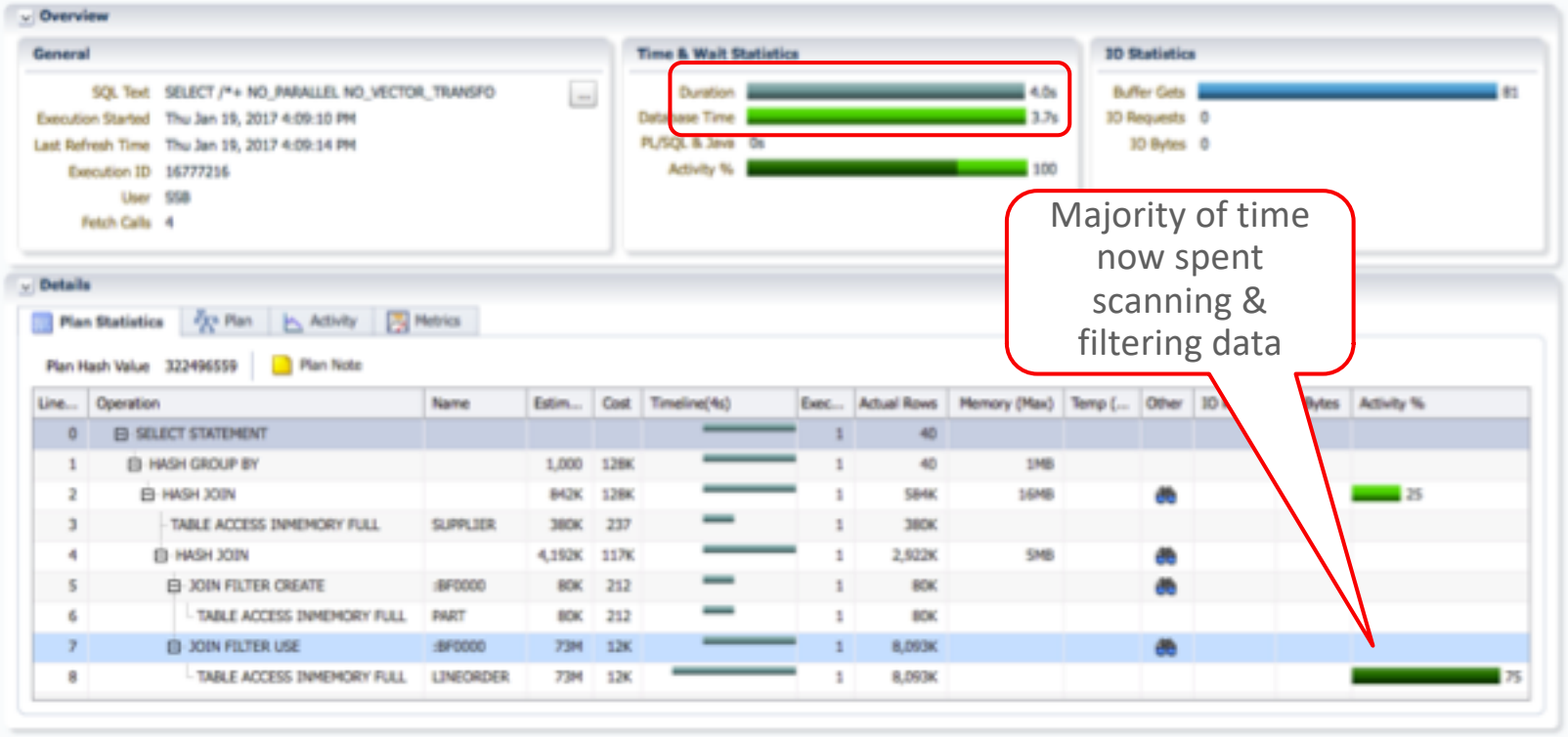
- Query to show total revenue by brand

```
SELECT  p.p_brand1,
        SUM(lo_revenue) rev
 FROM   lineorder l,
        part p,
        supplier s
 WHERE  l.lo_partkey = p.p_partkey
 AND    l.lo_suppkey = s.s_suppkey
 AND    p.p_category = 'MFGR#12'
 AND    s.s_region   = 'AMERICA'
 GROUP  BY p.p_brand1
```

Join Query – Traditional hash join



Join Query – In-Memory hash join with Bloom filters



Things To Remember With Joins

- Queries that spend a significant amount of time joining data are good candidates for Database In-Memory
- Work best with selective equality joins
  - Generates Bloom filter(s)
- Non equality joins won't benefit as much
  - **Consider using materialized views**
  - Make sure you have enough PGA allocated
  - Consider using parallel execution
  - **12.2 Adds Join Groups** – can improve join performance by using common dictionary encodings

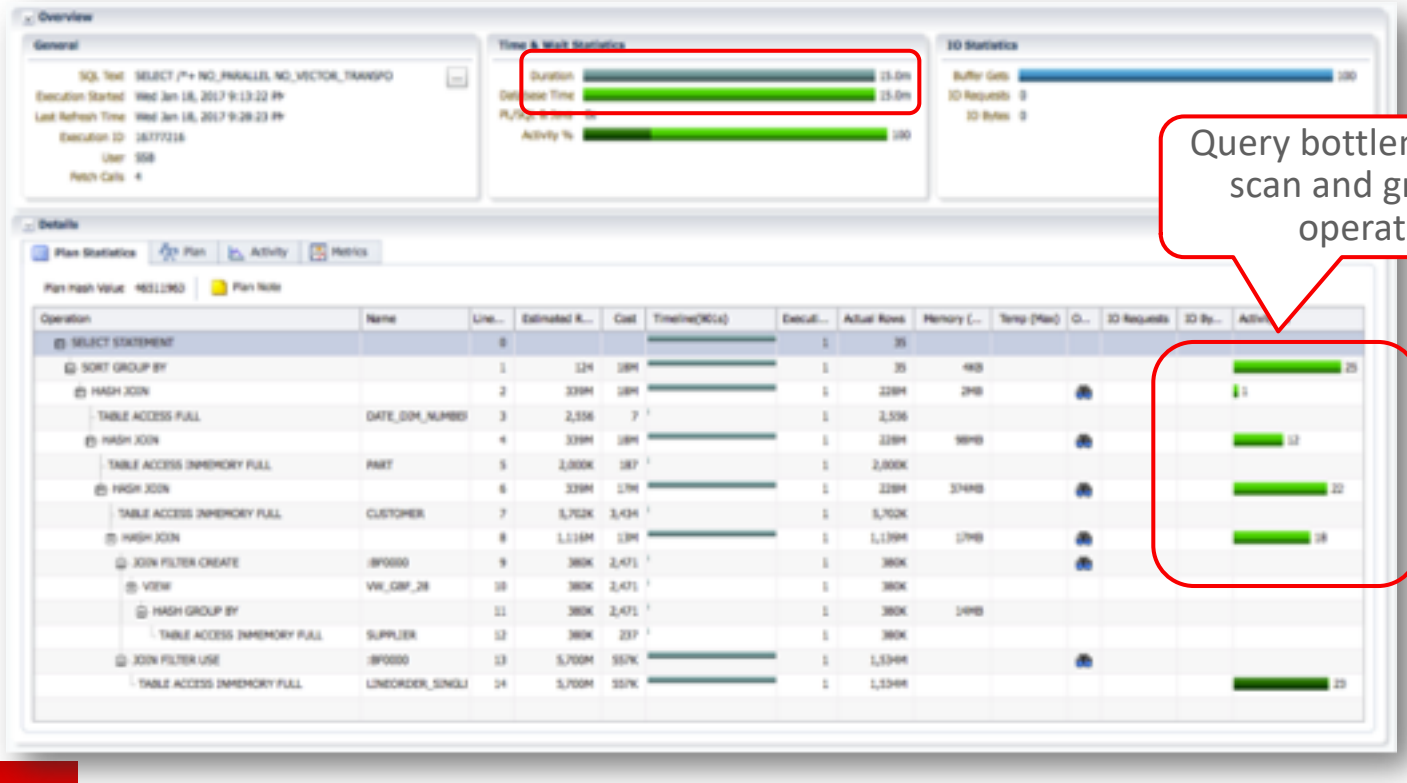
Aggregation

Aggregation Query

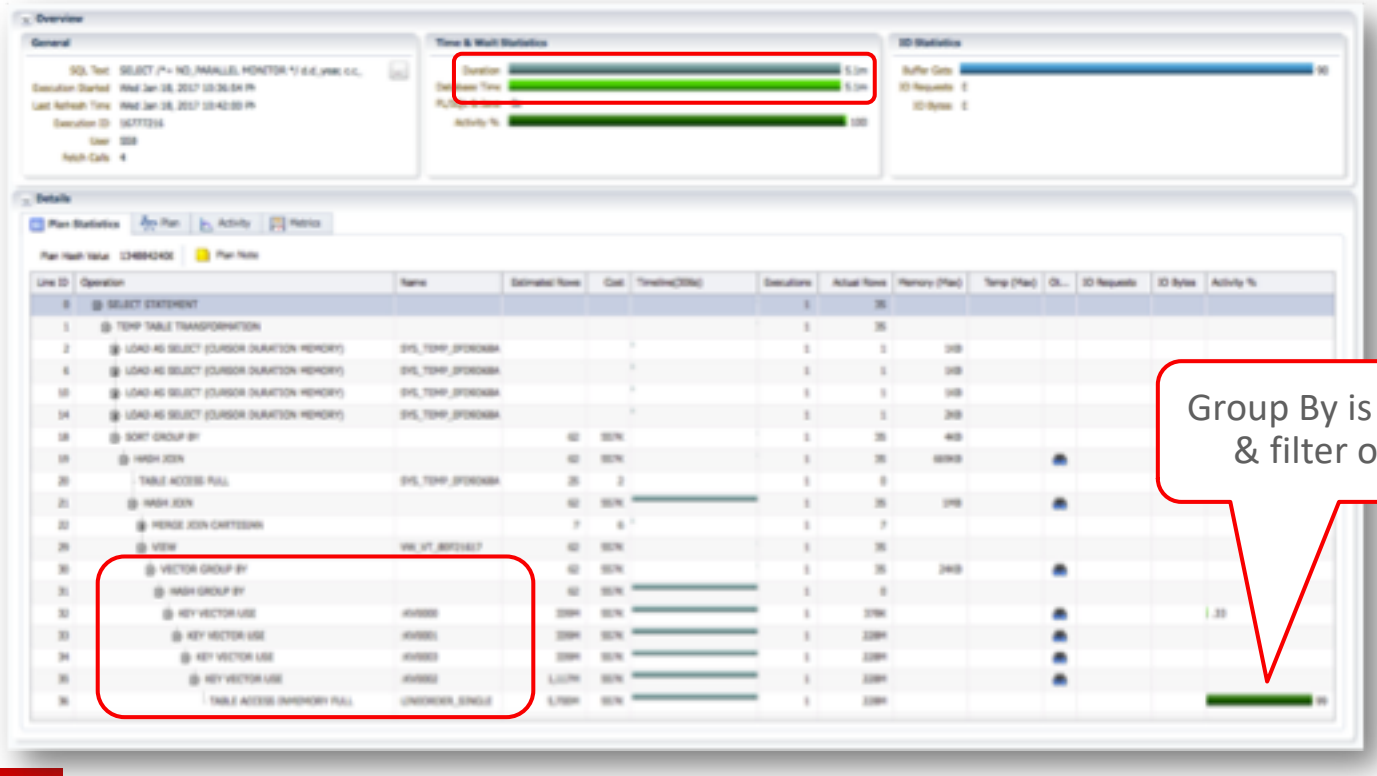
- Query to show total profit by year and nation

```
SELECT  d.d_year, c.c_nation, sum(lo_revenue - lo_supplycost)
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;
```

Aggregation – Traditional Group By



Aggregation – Vector Group By with In-Memory



Things To Remember With Aggregation

- Queries that spend a significant amount of time aggregating data are good candidates for Database In-Memory
- Vector Group By works best when
  - INMEMORY_SIZE parameter must be set to a non-zero value
  - The join columns between the tables contain "mostly" unique keys or numeric keys
  - The fact table (largest table in the query) is at least 10X larger than the other tables
  - The tables are populated into the IM column store
- **NOTE:** Tables don't have to be in the IM column store - vector group by can also be offloaded to Exadata storage

How Have Customers Benefited From In-Memory

How Customers Use Database In-Memory

AT&T WiFi – Data Warehouse

- Business Objects reports 100X faster
- ETL processes improved by 50% faster
- No changes to SAP Business Objects reports

Villeroy & Boch – SAP BW

- SAP BW COPA queries 30 – 33X faster
- SAP Transaction list queries 4 – 4,800X faster
- Avoided expensive & risky upgrade to S4/Hana

BOSCH – SAP CRM

- Dropped all custom indexes
- Analytic queries 2-20X faster, DML 2-3X faster
- No changes to application required

Die Mobiliar – Mixed Workload

- Analytic queries 50-200X faster
- Database size reduced considerably
- Phase out of Netezza and mainframe systems

How Customers Use Database In-Memory

Mankind Pharma – Mixed Workload

- Analytical reports 11x faster
- Dropping indexes improved OLTP
- 90% reduction in database size

Shanghai Customs – Mixed Workload

- Processes Clearance 43x Faster
- Improves Declaration-Services Efficiency
- Reduced Costs

- **LION** – SAP ERP

- Analytic queries 4X faster
- Transactions 2X faster
- Analytic queries now possible on 100s of Millions of Point-of-Sales Transactions

Lufthansa – Reporting Application

- Analytic queries up to 100x faster
- Improved data ingest performance
- Reduction in database size

Database In-Memory - Use Case Summary

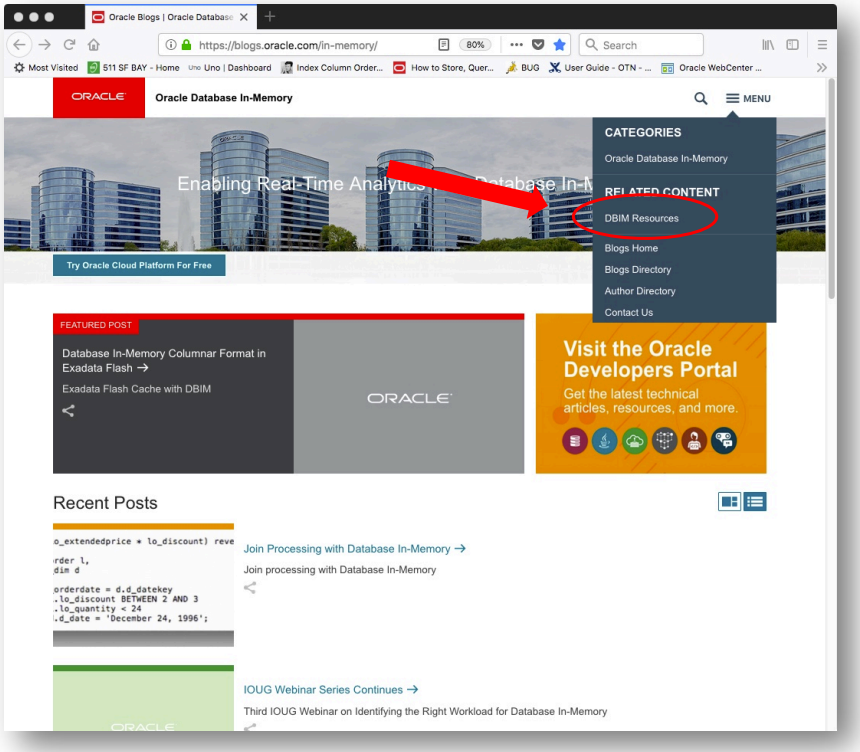
Mixed Workload

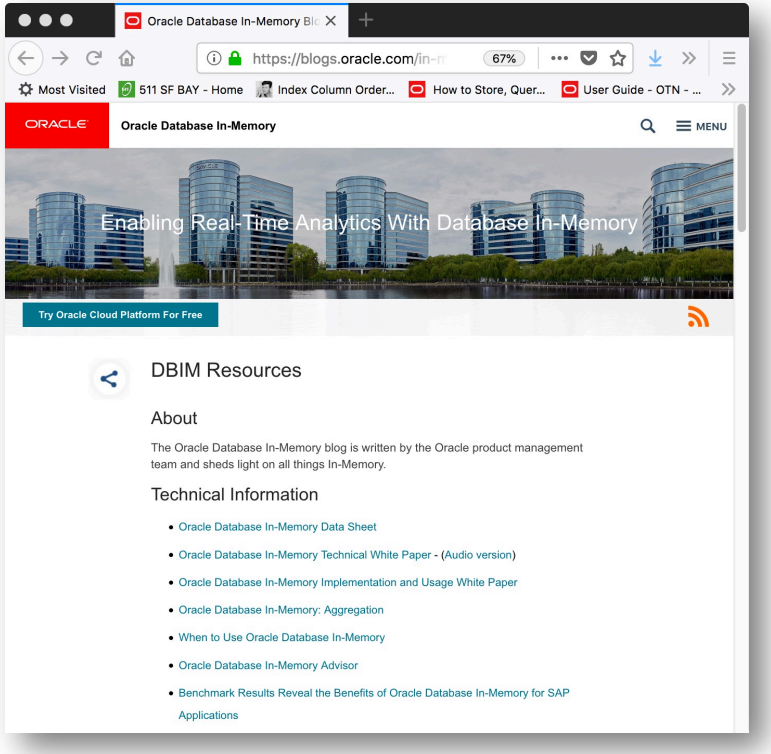
Real-Time Analytics

Reporting/BI

How Do I Get Started

https://blogs.oracle.com/in-memory/dbim-resources





Additional Resources

Join the Conversation

- https://twitter.com/db_inmemory
- https://twitter.com/TheInMemoryGuy
- http://www.oracle.com/goto/dbim.html

Database In-Memory Information

- Database In-Memory Blog
- oracle.com – Database In-Memory
- Database In-Memory YouTube Channel
- Ask TOM Database In-Memory Office Hours
- Database In-Memory Guide (Documentation)

ORACLE