Aggregate Navigation

using Materialized Views and Query Rewrite

John P. McKenna

Counterpoint Technologies, Inc.

Agenda

- Why Aggregate Navigation?
- What is Query Rewrite?
- Using Query Rewrite
- Design Guidelines
- Production Considerations
- Additional Assistance
- Demonstration

Why Aggregate Navigation?

- Aggregate Challenges
- Wouldn't it be nice if...
- The Aggregate Navigator
- Oracle's Aggregate Navigator

Aggregate Challenges

- What aggregates should be created?
- Are current aggregates being utilized?
- Would intermediate summarizations be more efficient?
- Are our users and developers using the best available aggregate?
- Will changing aggregates cause saved queries or applications to fail?

Wouldn’t it be nice if you could …

- utilize new aggregates without changing code
- eliminate aggregates with low usage
- change existing aggregations when needed
- eliminate the need for users and applications to understand aggregates
- insulate applications and users from the impact of changing aggregates

The Aggregate Navigator

- The aggregate navigator is a software component that intercepts SQL and transforms it to use the best available aggregate(s).
- According to Ralph Kimball
  - "An aggregate navigator is an essential component of a data warehouse because it insulates end user applications from the changing portfolio of aggregations, and allows the DBA to dynamically adjust the aggregations without having to roll over the applications base."

Oracle's Aggregate Navigator

Since the introduction of summary management in Oracle8i, the RDBMS contains an aggregate navigation capability. This feature is known as query rewrite.

What is Query Rewrite?

- The Definition
- The Process
- The Methods

The Definition

- According to the Oracle data warehousing guide, query rewrite '...transforms a SQL statement expressed in terms of tables or views into a statement accessing one or more materialized views that are defined on the detail tables.'
- In other words, your query is rewritten to take best advantage of summaries, joins or aggregations of your base tables that are found in materialized views.

The Process

- The cost based optimizer processes SQL
- Execution plans are generated
- Query is rewritten and additional execution plans are generated
- Plan costs are compared and the least cost query is processed

The Methods

- SQL text match (full & partial)
- Selection compatibility (where clause)
- Join compatibility (equivalent joins)
- Data sufficiency (columns)
- Grouping compatibility (granularity)
- Aggregate compatibility (sum, avg)

Using Query Rewrite

- Configure Parameters
- Create Materialized View(s)
- Define Constraints and Dimensions
- Write & Execute SQL Statement(s)
- Verify Rewrite Occurred

Parameter Requirements

- Compatible = '8.1.0' (or higher)
- Optimizer_mode = 'cost' or 'choose'
- Optimizer_features_enable = 'none', '8.1.6' (or higher)
- Query_rewrite_enabled = 'true'
- Query_rewrite_integrity = 'enforced', 'trusted' or 'stale_tolerated'

Query Rewrite Integrity

- Enforced
  - Only mviews known to contain fresh data
  - Only constraints that are ENABLED VALIDATED
  - Does not use hierarchy information
- Trusted
  - Trusts that mviews are fresh
  - Trusts RELY and NOVALIDATE constraints
  - Trusts hierarchy information
- Stale_tolerated
  - Maximum degree of rewrite, maximum risk

Materialized View Requirements

- Materialized view must contain the 'enable query rewrite' clause
- Query rewrite only works against materialized views, not custom built summary tables
- You may be able to register custom built summary tables as a materialized view using 'create materialized view…on prebuilt table'
  - Requires query_rewrite_integrity = 'trusted' or 'stale_tolerated'

Constraint and Dimension Requirements

Rewrite Methods	**Dimensions**	**Constraints**
Matching SQL Text	Not Required	Not Required
Join Compatibility	Not Required	Required
Data Sufficiency	Required	Required
Grouping Compatibility	Required	Required
Aggregate Computability	Not Required	Not Required

Writing SQL Statements

- All SQL should be coded to access base fact and dimension tables directly
- Do not reference any materialized views
- This removes the SQL dependency on any given aggregate

Did Query Rewrite Occur?

- Check explain plan
  - Oracle_home\rdbms\admin\utlxplan.sql
- Use dbms_mview.explain_rewrite
  - Oracle_home\rdbms\admin\utlxrw.sql

Design Guidelines

- Constraints
- Dimensions
- Aggregation
- Date Folding
- Hints

Constraints

- Define the constraints!
  - for all primary and foreign keys
  - include NOT NULL constraints
- For performance (or on views)
  - use NOVALIDATE and RELY on constraints
  - must set query_rewrite_integrity = 'stale_tolerated' or 'trusted'

Dimensions

- Define all dimensions & hierarchies!
- Ensure data in dimensions is accurate to hierarchies and levels, Oracle does not verify.
- Must set query_rewrite_integrity = ‘stale_tolerated’ or ‘trusted’

Aggregation

- Include aggregate functions in materialized views to support a greater number of rewrites.
  - Count, Sum, Avg, Etc.
- Include aggregate functions required for fast refreshes of the materialized views.
  - Count(*), count(expression), etc.

Date Folding

- Create aggregate columns that are eligible for date folding.
  - to_char(somedate, 'YYYY-MM') instead of to_char(somedate, 'MM-YYYY')

Hints

- Using hints defeats the purpose
- Make sure to implement all other techniques first (constraints, dimensions, date folding, etc.)
- If all else fails
  - /*+ norewrite */
  - /*+ rewrite (materialviewname) */

Production Considerations

- Keep statistics current!
  - They are key to evaluating SQL plans
- Keep materialized views refreshed!
  - Especially if you set query_rewrite_integrity = ‘trusted’ or ‘stale_tolerated’
  - Failed refreshes may cause data accuracy issues.

Additional Assistance

- The summary advisor (dbms_olap or enterprise manager) can provide assistance with materialized view analysis and advise.
- Materialized views (dbms_mview)
  - Explain_rewrite

Demonstration

- Detail query w/o query rewrite enabled
- Review explain plan
- Enable query rewrite
- Re-execute detail query
- Review explain plan

Questions & Answers

Feel free to contact me with additional questions:

John P. McKenna
Counterpoint Technologies, Inc.
jmckenna@counterpoint.biz
(631) 757-7264

References

- Oracle 9.2 Data Warehousing Guide
- Oracle 9.2 PL/SQL Packages Guide
- Oracle 9.2 Sample Schemas Guide
- Kimball, Ralph. The Data Warehouse Toolkit. John Wiley & Sons, Inc. 1996.