

In The Dbms_ddl Package, The ____ Procedure Creates Statistics For Database Objects To Be Utilized To

In The Dbms_ddl Package, The ____ Procedure Creates Statistics For Database Objects To Be Utilized To

Understanding the intricacies of database management is essential for optimizing performance and ensuring efficient data retrieval. One crucial aspect of this is the maintenance and creation of accurate statistics for database objects. The Oracle `DBMS\_DDL` package plays a vital role in managing Data Definition Language (DDL) operations, and among its various functionalities, the procedure responsible for creating statistics is fundamental for database administrators and developers alike. This article explores the purpose and functionality of the `CREATE\_STATISTICS` procedure within the `DBMS\_DDL` package, illustrating how it contributes to database optimization.

Overview of the DBMS_DDL Package

The `DBMS\_DDL` package is a built-in Oracle PL/SQL package designed to facilitate Data Definition Language operations such as creating, altering, and dropping database objects. It provides a programmatic way to manage database objects and execute DDL commands dynamically within PL/SQL code.

Main Features of DBMS_DDL

- Executing DDL statements dynamically: Allows for the automation of schema management tasks.
- Managing object dependencies: Ensures that operations are executed without violating dependencies.
- Supporting audit and logging: Tracks changes made to database objects.
- Providing advanced features: Such as enabling or disabling constraints, triggers, and gathering statistics.

The package is particularly useful in scripts and applications that require dynamic SQL execution or complex database object management.

Understanding Statistics in Oracle Databases

Before delving into the specifics of the procedure, it's essential to understand the importance of statistics in Oracle databases.

What Are Database Statistics?

Database statistics are metadata that describe the characteristics of database objects like tables, indexes, columns, and partitions. These statistics include data such as:

- Number of rows
- Number of distinct values in columns
- Data distribution
- Column histograms
- Index selectivity

Why Are Statistics Important?

- Query Optimization: The Oracle optimizer relies heavily on statistics to determine the most efficient execution plans.
- Performance Tuning: Accurate statistics lead to faster query execution and overall system performance.
- Resource Management: Helps in managing system resources effectively by predicting query behavior.
- Adaptive Behavior: Facilitates the optimizer's ability to adapt to data changes over time.

Inaccurate or outdated statistics can lead to suboptimal execution plans, resulting in slow queries and increased resource consumption.

The Role of the `CREATE_STATISTICS` Procedure in `DBMS_DDL`

Within the `DBMS_DDL` package, the procedure typically responsible for creating or managing statistics is called `CREATE_STATISTICS`. This procedure allows database administrators to programmatically generate or refresh statistics for various database objects.

Purpose of `CREATE_STATISTICS`

The primary goal of the `CREATE_STATISTICS` procedure is to automate the process of gathering statistics, especially in scenarios such as:

- Custom statistics creation for specific columns or objects
- Rebuilding or updating statistics after significant data changes
- Ensuring consistency and accuracy in optimizer decisions
- Automating maintenance tasks in scripts or scheduled jobs

How CREATE_STATISTICS Works

The procedure accepts parameters specifying the target object, the type of statistics to generate, and optional settings for the process.

In general, the syntax resembles:

```
```sql
BEGIN
DBMS_DDL.CREATE_STATISTICS (
own_name => 'schema_name',
object_name => 'table_name',
method_opt => 'for all columns size auto',
stattab => 'statistics_table_name',
statid => 'statistic_id'
);
END;
```
```

- own_name: Owner/schema name of the object
- object_name: Name of the object (table, index, etc.)
- method_opt: Options specifying which columns or objects to gather statistics for
- stattab: Optional parameter to specify a table for storing statistics
- statid: Identifier for the set of statistics

Parameters and Their Significance

| Parameter | Description | Optional/Required |
|-------------|--|-------------------|
| own_name | Schema owner of the object | Required |
| object_name | Name of the database object | Required |
| method_opt | Specifies which columns or objects to gather stats for | Optional |
| stattab | Name of the statistics table to store data | Optional |
| statid | Identifier for the statistics set | Optional |

Use Cases for the CREATE_STATISTICS Procedure

The flexibility of the `CREATE\_STATISTICS` procedure makes it suitable for a variety of scenarios.

1. Creating Custom Column Statistics

When default statistics are insufficient, administrators can create custom statistics for specific columns to guide the optimizer better.

Example:

```
```sql
BEGIN
DBMS_DDL.CREATE_STATISTICS(
own_name => 'HR',
object_name => 'EMPLOYEES',
method_opt => 'for columns salary size 1'
);
END;
```
```

This gathers detailed statistics for the `salary` column in the `EMPLOYEES` table.

2. Rebuilding Statistics After Data Changes

Significant insertions, updates, or deletions can render existing statistics outdated. Using `CREATE\_STATISTICS`, administrators can regenerate accurate statistics.

Example:

```
```sql
BEGIN
DBMS_DDL.CREATE_STATISTICS(
own_name => 'HR',
object_name => 'DEPARTMENTS',
method_opt => 'for all columns size auto'
);
END;
```
```

3. Automating Statistics Maintenance

In large environments, manual updates are inefficient. Automating via scripts or scheduled jobs ensures statistics stay current.

Best Practices for Using CREATE_STATISTICS

Proper use of the `CREATE\_STATISTICS` procedure enhances database performance and reliability.

1. Understand Your Data

Before creating or updating statistics, analyze data characteristics and distribution to determine which statistics are necessary.

2. Use Method Options Wisely

- Use `'for all columns size auto'` for comprehensive statistics.
- Specify particular columns if only certain attributes change frequently.

3. Schedule Regular Updates

Establish routines to refresh statistics periodically, especially after bulk data loads or significant changes.

4. Monitor and Validate

After creating statistics, monitor query performance and validate that execution plans have improved.

5. Leverage Oracle's Auto-Statistics Features

Oracle provides automatic statistics gathering; however, for specific needs, manual procedures like `'CREATE_STATISTICS'` are invaluable.

Advantages of Programmatic Statistics Creation with DBMS_DDL

Using `'DBMS_DDL'` for creating statistics offers several benefits:

- Automation: Integrate into maintenance scripts.
- Precision: Target specific objects and columns.
- Consistency: Maintain uniformity across environments.
- Control: Customize statistics creation parameters.

Limitations and Considerations

While powerful, there are some caveats:

- Permissions: Requires appropriate privileges.

- Impact on Performance: Gathering statistics can be resource-intensive; schedule during off-peak hours.
- Compatibility: Ensure version support for the specific procedures and parameters.
- Data Sensitivity: Be cautious when creating statistics on sensitive data.

Conclusion

The `CREATE_STATISTICS` procedure within the `DBMS_DDL` package is a vital tool for Oracle database administrators aiming to optimize query performance through accurate and timely statistics. By enabling programmatic, precise, and automated creation of statistics for database objects, it plays a crucial role in maintaining an efficient, responsive database environment. Proper understanding and implementation of this procedure can lead to significant improvements in query execution plans, resource utilization, and overall database health.

Further Resources

- Oracle Database SQL Language Reference
- Oracle Database Performance Tuning Guide
- Official Oracle Documentation on `DBMS_DDL` Package
- Best practices for database statistics management

Implementing effective statistics management is an ongoing process that requires understanding your data, monitoring system performance, and leveraging tools like `CREATE_STATISTICS` to maintain optimal database operation.

Frequently Asked Questions

What is the purpose of the procedure in the `DBMS_DDL` package that creates statistics for database objects?

The procedure in the `DBMS_DDL` package that creates statistics is used to gather and update statistical information about database objects, which helps the optimizer make informed decisions to improve query performance.

Which procedure in the `DBMS_DDL` package is responsible for creating statistics for database objects?

The procedure responsible for creating statistics in the `DBMS_DDL` package is the `'CREATE_STATISTICS'` procedure.

How does the CREATE_STATISTICS procedure in DBMS_DDL improve database performance?

It enhances performance by collecting detailed statistical information about database objects, enabling the optimizer to choose more efficient execution plans.

Can the CREATE_STATISTICS procedure in DBMS_DDL be used to gather statistics on both tables and indexes?

Yes, the CREATE_STATISTICS procedure can be used to gather statistics on tables, indexes, and other database objects to optimize query execution.

Is it necessary to run the CREATE_STATISTICS procedure regularly, and why?

Yes, regularly running CREATE_STATISTICS ensures that the optimizer has up-to-date information, which is essential for maintaining optimal query performance as data changes over time.

Additional Resources

DBMS_DDL Package: The CREATE_STATISTICS Procedure – Enhancing Database Optimization

In the realm of modern database management systems (DBMS), performance optimization is paramount. As databases grow in complexity and volume, the need for accurate, up-to-date statistics becomes critical for the query optimizer to generate efficient execution plans. Among the tools provided by Oracle Database to facilitate this, the DBMS_DDL package stands out for its versatility, particularly through its CREATE_STATISTICS procedure. This procedure plays a vital role in creating and managing statistics for database objects, directly impacting query performance and resource utilization.

In this article, we delve deeply into the CREATE_STATISTICS procedure within the DBMS_DDL package. We explore its purpose, functionality, usage scenarios, and best practices, providing a comprehensive understanding of how it empowers database administrators and developers to optimize their Oracle databases effectively.

Understanding the DBMS_DDL Package

Overview of DBMS_DDL

The DBMS_DDL package in Oracle provides APIs for managing Data Definition Language

(DDL) operations dynamically. It allows for programmatic creation, modification, and removal of database objects such as tables, indexes, and views. Beyond these basic capabilities, it also facilitates advanced operations like enabling or disabling constraints, renaming objects, and managing statistics.

The package's primary advantage lies in its ability to execute DDL statements within PL/SQL blocks or procedures, offering automation and scripting flexibility that manual commands may lack. This makes it a powerful tool for database administrators aiming for efficient, repeatable, and controlled schema modifications.

The Role of CREATE_STATISTICS in Database Optimization

Why Statistics Matter

Statistics are vital for the Oracle optimizer to determine the most efficient way to execute queries. They include data such as:

- Row counts
- Data distribution
- Column cardinality
- Histograms

Accurate and current statistics help the optimizer choose optimal execution plans, which translate into faster query responses and reduced resource consumption. Conversely, stale or missing statistics can lead to suboptimal plans, causing performance degradation.

Purpose of CREATE_STATISTICS

The CREATE_STATISTICS procedure is designed to create or update optimizer statistics for various database objects, such as tables, columns, or entire schemas. Its primary objectives include:

- Collecting detailed data distribution information
- Enhancing the accuracy of optimizer decisions
- Replacing or supplementing existing statistics
- Facilitating manual or automated statistics management

By leveraging this procedure, DBAs can ensure their database maintains a high level of statistical integrity, especially after significant data modifications or schema changes.

Functionality of CREATE_STATISTICS Procedure

Core Capabilities

The CREATE_STATISTICS procedure offers a range of functionalities, including:

- Creating statistics for individual columns, groups of columns, or entire tables
- Overriding existing statistics if desired
- Generating global or object-specific statistics
- Automating statistics creation within scripts or scheduled jobs

This flexibility allows for precise control over the scope and nature of statistics being generated.

Syntax Overview

While the full syntax can be complex, a simplified version is as follows:

```
```sql
BEGIN
DBMS_STATS.CREATE_STATISTICS (
ownname => 'schema_name',
tablename => 'table_name',
pname => 'column_name',
stattype => 'CONSTANT',
cascade => TRUE,
statid => 'my_stat_id',
statown => 'owner_name',
stattab => 'statistics_table_name'
);
END;
```
```

This procedure allows detailed specification of:

- The object for which statistics are created
- The specific columns or groups
- The type of statistics
- Additional options like cascading and custom identifiers

Key Parameters and Their Significance

Understanding the parameters is crucial for effective use:

- ownname: The schema owning the object (table or index)
- tabname: The name of the object (table, index, or column)
- pname: The specific column or columns for which statistics are created
- stattype: Type of statistics, e.g., 'SAMPLE', 'CONSTANT'
- cascade: Whether to gather statistics on dependent objects like indexes
- statid: An optional identifier for the statistics, useful for versioning
- stattab: The name of a custom table to store the statistics, if needed

By tuning these parameters, users can tailor the statistics creation process to suit specific optimization needs.

Use Cases and Practical Applications

1. Manual Statistics Refreshing

While Oracle automatically maintains statistics through tools like `DBMS_STATS.GATHER_TABLE_STATS`, there are scenarios where manual intervention is necessary, such as:

- Post-large data loads or bulk updates
- Schema changes impacting data distribution
- Performance troubleshooting

Using `CREATE_STATISTICS`, DBAs can precisely target specific objects or columns to refresh statistics, ensuring the optimizer operates on accurate data.

2. Custom Statistics Generation

In cases where default statistics are insufficient, `CREATE_STATISTICS` enables the creation of custom or extended statistics, including:

- Extended statistics for columns with correlated data
- Histograms for skewed data distributions
- Grouped column statistics for multi-column selectivity

This customization can significantly improve query plan quality for complex queries.

3. Managing Historical or Versioned Statistics

By assigning statid parameters, users can maintain different versions of statistics, facilitating:

- A/B testing of different statistical scenarios
- Rollbacks to previous statistics if new ones cause regressions
- Tracking statistical changes over time

This capability enhances control and transparency in database tuning efforts.

Best Practices for Using CREATE_STATISTICS

1. Use with Caution in Production Environments

While manual statistics creation is powerful, it should be used judiciously, especially in production, to avoid unintended performance impacts. Always test changes in staging environments before deploying to production.

2. Combine with Automated Tools

Leverage Oracle's DBMS_STATS package for routine statistics gathering and reserve CREATE_STATISTICS for targeted, specialized tasks. This integrated approach ensures comprehensive and accurate data for the optimizer.

3. Maintain Consistency

Use statid identifiers to maintain consistency and facilitate comparisons across different statistics snapshots. Document your statistical strategies for clarity and future reference.

4. Monitor and Validate

After creating or updating statistics, monitor query performance and execution plans. Use EXPLAIN PLAN and SQL Monitor tools to validate improvements or identify issues.

5. Keep Statistics Up-to-Date

Regularly review and refresh statistics, especially after significant data loads, schema modifications, or performance regressions. The CREATE_STATISTICS procedure is an

essential tool in this ongoing maintenance process.

Limitations and Considerations

While the `CREATE_STATISTICS` procedure is a powerful feature, users should be aware of its limitations:

- It requires appropriate privileges (``EXECUTE`` on ``DBMS_STATS``)
- Overuse or improper application can cause performance overhead
- It doesn't replace automated statistics gathering but complements it
- Careful planning is necessary when creating custom or versioned statistics to avoid conflicts

Conclusion: Empowering Optimization with `CREATE_STATISTICS`

The `CREATE_STATISTICS` procedure within Oracle's `DBMS_DDL` package is a sophisticated and flexible tool that significantly enhances the database administrator's ability to maintain optimal query performance. By enabling precise, manual creation and management of optimizer statistics, it addresses the complex needs of large, dynamic, and performance-critical databases.

When used judiciously and in conjunction with Oracle's broader statistics management ecosystem, `CREATE_STATISTICS` empowers DBAs to fine-tune their databases, adapt to changing data landscapes, and ensure that the query optimizer always has the most relevant information to generate efficient execution plans.

In the ever-evolving landscape of database performance tuning, mastering this procedure is a valuable step toward achieving robust, high-performing Oracle environments.

[In The Dbms Ddl Package The Procedure Creates Statistics For Database Objects To Be Utilized To](#)

In The Dbms Ddl Package The Procedure Creates Statistics For Database Objects To Be Utilized To

Related Articles

- [In Going Home, Jacob Lawrence Uses Abstract Figures And Objects And Color Themes To](#)

_____. Group

- [If Amite And The Author Of Fish Cheeks Wrote A Text Terrell My First Taste Of Fish Cheek What Was Her](#)
- [How Lionbridge Plans And Creates Optimized Articles For A Global Multilingual Website Seo Copywriting](#)

[Back to Home](#)