

CFQUERYPARAM Tag and Oracle Databases

ISSUE:

When ColdFusion's CFQUERY tag is used to issue an SQL statement (e.g. select, insert, update or delete) against an Oracle database, bind variables are not used. It is considered good coding practice to use bind variables when writing Oracle SQL statements since their usage may result in both application and Oracle server response time improvements. Bind variables are placeholders in SQL statements for query input values. They point to a memory location where the data values will be placed. The CFQUERYPARAM tag, introduced in ColdFusion 4.5.1, enables the use of bind variables with Oracle.

RECOMMENDATION:

The CFQUERYPARAM should be used to force the use of bind variables with Oracle.

The following ColdFusion code example shows a CFQUERY statement that does not force the use of bind variables:

```
<cfquery datasource=#Application.cDataSource#
  username=#Application.cUserName#
  password=#Application.cPassWord#  name="qUser">
  select * from ext_account
  where web_pass_fl = 'Y'
  and account_name=
    '#ucase(rtrim(ltrim(cUserid)))#'
  and deleted_fl = 'N'
  and instance = '#cInstance#'
</cfquery>
```

The following ColdFusion code example shows the same CFQUERY statement that has been modified to use bind variables:

```
<cfquery datasource=#Application.cDataSource#
  username=#Application.cUserName#
  password=#Application.cPassWord#  name="qUser">
  select * from ext_account
  where web_pass_fl = 'Y'
  and account_name= <CFQUERYPARAM
    VALUE="#ucase(rtrim(ltrim(cUserid)))#"
    CFSQLTYPE="CF_SQL_VARCHAR">
  and deleted_fl = 'N'
  and instance = <CFQUERYPARAM value="#cInstance#"
    CFSQLTYPE="CF_SQL_VARCHAR">
</cfquery>
```

The second ColdFusion code example replaces all references to variables (cUserId and cInstance) with the variables enclosed in the CFQUERYPARAM tag and forces the use of bind variables. By using bind variables, the resulting SQL statement does not have to be completely parsed each time it is executed and a separate copy of the resulting SQL statement is not stored in the shared pool for each set of values assigned to the variables. The shared pool is a sharable area of memory used by Oracle to hold SQL statements, packages, object information and other items. From the ColdFusion application perspective, the use of bind variables can improve the application response time (i.e. reduce the time it takes to load a page). From the Oracle server perspective, the use of bind variables can improve overall server response time and reduce the number of SQL statements kept in the shared pool.

DISCUSSION:

To further explain the performance implications of the CFQUERYPARAM tag, the following explains how Oracle processes SQL statements and shows the SQL statements stored in the shared pool for the two ColdFusion code examples.

Oracle Processing of SQL Statements

When an application sends an SQL statement to the Oracle server, the Oracle server checks the statement for syntax correctness, verifies privileges, and checks if the same SQL statement has already been run and an exact copy of the SQL statement is in the shared pool. It is important to note that from Oracle's perspective, two SQL statements are the same if they are identical character for character (e.g. same capitalization and spacing) and the object names in both statements resolve to the same objects (i.e. same tables owned by the same schema). If the same SQL statement is found in the shared pool, then the SQL statement is executed using the execution plan already established for the statement. Execution plan is Oracle's term for the steps followed in evaluating and executing the statement. A SQL statement found in the shared pool results in the least work by the Oracle server and the least amount of time to process the statement. If the SQL statement is not found in the shared pool, then additional work is required by the Oracle server to completely parse the statement, set up the most efficient execution plan and place the statement in the shared pool. Thus, a SQL statement not found in the shared pool results in more work being done by the Oracle server and a longer response time.

SQL Statements Stored in the Shared Pool

All SQL statements, in addition to copies of packages, object information and other items, are stored in the shared pool. When an Oracle instance is first set up, memory is allocated for the shared pool. When the Oracle database is up, items are loaded into the shared pool as needed. If the Oracle server finds that there is not enough room in the shared pool for an item, then it will swap out the least-recently used item from the shared pool to make room for the new item. Thus, it is important to minimize the number of copies of similar items in the shared pool to avoid swapping.

In the first ColdFusion code example, each time the page containing the CFQUERY tag was loaded, a different set of values was assigned to the variables (cUserId and cInstance) in the CFQUERY tag. Thus, each CFQUERY tag evaluation resulted in a different SQL statement being sent to the Oracle server. Each statement had to be completely parsed by the Oracle server before being stored as a new statement in the shared pool. The following are a few of the SQL statements found in the shared pool after loading the page containing the CFQUERY tag several times:

```
select * from ext_account where web_pass_fl = 'Y'  
and account_name= 'TEST_ACCT'  
and deleted_fl = 'N' and instance= 'D815A'
```

```
select * from ext_account where web_pass_fl = 'Y'  
and account_name= 'TEST_ACCT'  
and deleted_fl = 'N' and instance= 'P815A'
```

```
select * from ext_account where web_pass_fl = 'Y'  
and account_name= 'TEST_ACCT'  
and deleted_fl = 'N' and instance= 'T815A'
```

```
select * from ext_account where web_pass_fl = 'Y'  
and account_name= 'JONES'  
and deleted_fl = 'N' and instance= 'P815A'
```

```
select * from ext_account where web_pass_fl = 'Y'  
and account_name= 'JONES'  
and deleted_fl = 'N' and instance= 'T815A'
```

In the second ColdFusion code example, each time the page containing the CFQUERY tag using CFQUERYPARAM was loaded, bind variables (:1 and :2) were inserted into the resulting SQL statement. Thus, each CFQUERY tag evaluation resulted in the same SQL statement being sent to the Oracle server and the Oracle server was able to take advantage of the processing done the first time the SQL statement was received. In addition, it resulted in the fewest number of SQL statements being stored in the shared pool. The following is the only SQL statement found in the shared pool after loading the page containing the CFQUERY tag several times:

```
select * from ext_account where web_pass_fl = 'Y'  
and account_name= :1  
and deleted_fl = 'N' and instance = :2
```

As illustrated by the ColdFusion code examples, the number of copies of the SQL statement in the shared pool is reduced due to the use of bind variables. This has a number of benefits including faster loading of ColdFusion pages, improved Oracle server response time, and minimize duplicate SQL statements in the shared pool.

ADDITIONAL INFORMATION:

CFQUERYPARAM and Oracle Databases article by Justin Fidler
[CFQUERYPARAM and Oracle Databases](#)

On-line documentation at Macromedia's website
[On-line documentation](#)