

Select Magazine - January 2002

Volume 8, No. 2

Enhancements to Oracle9i SQL

By Biju Thomas

Although most of the major features of Oracle9i are on the database and application server side, many enhancements to SQL also are available. While reviewing them, two of the new features specifically caught my attention - the CASE expression and the MERGE command. This article will provide a brief review of these new SQL structures.

With the introduction of the CASE expression, we can now use the IF THEN ELSE construct within Oracle. Two forms of CASE expression exist- simple case and searched case. Both cases can be manipulated using a DECODE function, but CASE is more readable.

The syntax of a simple CASE expression is illustrated in the following code box:

```
CASE expression
WHEN comparison THEN return_value ... ..
ELSE return_value;
```

The following is an example:

```
SELECT COUNTRY_ID, COUNTRY_NAME,
       CASE COUNTRY_ID WHEN 'US' THEN 'USA'
                        WHEN 'CA' THEN 'N AMERICA'
                        WHEN 'MX' THEN 'C AMERICA'
                        ELSE 'S AMERICA' END REGION
FROM   COUNTRIES
WHERE  REGION_ID = 2
/
```

CO	COUNTRY_NAME	REGION
AR	Argentina	S AMERICA
BR	Brazil	S AMERICA
CA	Canada	N AMERICA
MX	Mexico	C AMERICA
US	United States of America	USA

5 rows selected.

The same statement can be written using the familiar DECODE function. The SQL is as follows:

```
SELECT COUNTRY_ID, COUNTRY_NAME,
       DECODE(COUNTRY_ID, 'US', 'USA', 'CA', 'N AMERICA',
              'MX', 'C AMERICA', 'S AMERICA') REGION
FROM   COUNTRIES
WHERE  REGION_ID = 2
/
```

In searched CASE, several conditions can be specified and the value is returned when the first condition is met. The syntax of searched CASE is illustrated in the following code box:

```
[begin code]
CASE
WHEN condition THEN return_value ... ..
ELSE return_value;
```

Here is an example:

```
SELECT LAST_NAME, SALARY,
       CASE WHEN SALARY < 5000 THEN 'LOW'
            WHEN SALARY < 8000 THEN 'MEDIUM'
            ELSE 'HIGH' END CATEGORY
FROM   EMPLOYEES
WHERE  DEPARTMENT_ID = 60
/
```

```
LAST_NAME                SALARY CATEGO
-----
Hunold                    9000 HIGH
Ernst                     6000 MEDIUM
Austin                    4800 LOW
Pataballa                 4800 LOW
Lorentz                   4200 LOW
```

5 rows selected.

The above SQL can be written using PL/SQL or by using multiple DECODE and SIGN functions in a pre-9i release.

The MERGE Command

MERGE is a new command introduced in 9i. This is very powerful and will be widely used by programmers. The MERGE statement combines two operations in one statement - an INSERT and an UPDATE. The decision to INSERT or UPDATE an operation is based on the condition specified in the ON clause. The syntax of the MERGE command is illustrated in the following code sample.

```
MERGE INTO table_name
USING [table|view|subquery]
ON (condition)
WHEN MATCHED THEN update_clause
WHEN NOT MATCHED THEN insert_clause
```

A simple example would be an employee_email table which has the employee_id and email address. This table is periodically updated with current information from the Employee table. If the employee ID is already in the table, the email information will be updated. Otherwise a new record will be created with the employee ID and email. The following statement will do this for all employees belonging to Department 60:

```
MERGE INTO EMPLOYEE_EMAIL EE
USING (SELECT EMPLOYEE_ID, LAST_NAME, FIRST_NAME, EMAIL
       FROM   EMPLOYEES
       WHERE  DEPARTMENT_ID = 60) E
ON (E.EMPLOYEE_ID = EE.EMPLOYEE_ID)
WHEN MATCHED THEN UPDATE SET EE.EMAIL = E.EMAIL
WHEN NOT MATCHED THEN INSERT (EMPLOYEE_ID, EMAIL)
VALUES (E.EMPLOYEE_ID, SUBSTR(E.LAST_NAME,1,1)
       || FIRST_NAME)
/
```

5 rows merged.

Notice that, since we needed to perform a WHERE condition to specify operation only on department 60, we created an inline view (subquery) with that condition. Using this statement in a pre-9i release would take several lines of PL/SQL code.

About the Author

Biju Thomas is Oracle Certified DBA based in Fort Worth, TX. He co-authored the Oracle8i OCP SQL study guide and authored the Oracle8i OCP Architecture and Administration study guide published by Sybex. He has also written articles for Oracle Internals and Oracle Magazine. He can be reached at biju@bijoos.com. Visit his website for Oracle DBAs @ <http://www.bijoos.com/oracle>.

[Download Acrobat Reader](#)

Copyright 2003 by the International Oracle Users Group