

FOR UPDATE Clause

Syntax:

```
SELECT ...  
FROM ...  
FOR UPDATE [OF column_reference] [NOWAIT | WAIT n];
```

- Use explicit locking to deny access to other sessions for the duration of a transaction.
- Lock the rows *before* the update or delete.

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

FORUPDATE Clause

If there are multiple sessions for a single database, there is the possibility that the rows of a particular table were updated after you opened your cursor. You see the updated data only when you reopen the cursor. Therefore, it is better to have locks on the rows before you update or delete rows. You can lock the rows with the FORUPDATE clause in the cursor query. In the syntax:

column_reference Is a column in the table against which the query is performed (A list of columns may also be used.)

NOWAIT Returns an Oracle Server error if the rows are locked by another session

The FORUPDATE clause is the last clause in a SELECT statement, even after ORDERBY (if it exists). When you want to query multiple tables, you can use

the FORUPDATE clause to confine row locking to particular tables. FOR UPDATE OF col_name(s) locks rows only in tables that contain col_name(s).

FORUPDATE Clause (continued)

The SELECT...FORUPDATE statement identifies the rows that are to be updated or deleted, and then locks each row in the result set. This is useful when you want to base an update on the existing values in a row. In that case, you must make sure that the row is not changed by another session before the update.

The optional NOWAIT keyword tells the Oracle Server not to wait if the requested rows have been locked by another user. Control is immediately returned to your program so that it can do other work before trying again to acquire the lock. If you omit the NOWAIT keyword, the Oracle Server waits until the rows are available.

Example:

```
DECLARE
  CURSOR c_emp_cursor IS
    SELECT employee_id, last_name, FROM employees
    WHERE department_id = 80 FOR UPDATE OF salary
    NOWAIT;
```

...

If the Oracle Server cannot acquire the locks on the rows it needs in a SELECTFOR UPDATE operation, it waits indefinitely. Use NOWAIT to handle such situations. If the rows are locked by another session and you have specified NOWAIT, opening the cursor results in an error. You can try to open the cursor later. You can use WAIT instead of NOWAIT, specify the number of seconds to wait, and then determine whether the rows are unlocked. If the rows are still locked after n seconds, an error is returned. It is not mandatory for the FORUPDATEOF clause to refer to a column, but it is recommended for better readability and maintenance.

WHERE CURRENT OF Clause

Syntax:

```
WHERE CURRENT OF cursor ;
```

- Use cursors to update or delete the current row.
- Include the FOR UPDATE clause in the cursor query to first lock the rows.
- Use the WHERE CURRENT OF clause to reference the current row from an explicit cursor.

```
UPDATE employees  
SET    salary = ...  
WHERE CURRENT OF c_emp_cursor;
```

ORACLECopyright © 2010, Oracle and/or its affiliates. All rights reserved.

WHERECURRENTOF Clause

The WHERECURRENTOF clause is used in conjunction with the FOR UPDATE clause to refer to the current row in an explicit cursor. The WHERE CURRENTOF clause is used in the UPDATE or DELETE statement, whereas the FORUPDATE clause is specified in the cursor declaration. You can use the combination for updating and deleting the current row from the corresponding database table. This enables you to apply updates and deletes to the row currently being addressed, without the need to explicitly reference the row ID. You must include the FORUPDATE clause in the cursor query so that the rows are locked on OPEN.

In the syntax:

cursor Is the name of a declared cursor (The cursor must have been declared with the FOR UPDATE clause.)

