Database Programming with PL/SQL

5-5: Using Cursors FOR UPDATE

Practice Activities

Vocabulary

|  |  |
| --- | --- |
|  | Declares that each row is locked as it is being fetched so  other users cannot modify the rows while the cursor is  open |
|  | A keyword used to tell the Oracle server not to wait if the  requested rows have already been locked by another user |

Try It / Solve It

In this Practice you will INSERT and later UPDATE rows in a new table: PROPOSED\_RAISES, which

will store details of salary increases proposed for suitable employees. Create this table by executing

the following SQL statement:

**CREATE TABLE proposed\_raises**

**(date\_proposed DATE,**

**date\_approved DATE,**

**employee\_id NUMBER(6),**

**department\_id NUMBER(4),**

**original\_salary NUMBER(8,2),**

**proposed\_new\_salary NUMBER(8,2));**

1. Write a PL/SQL block that inserts a row into PROPOSED\_RAISES for each eligible employee.

The eligible employees are those whose salary is below a chosen value. The salary value is

passed as a parameter to the cursor. For each eligible employee, insert a row into

PROPOSED\_RAISES with date\_proposed = today’s date, date\_appoved null, and

proposed\_new\_salary 5% greater than the current salary. The cursor should LOCK the

employees rows so that no one can modify the employee data while the cursor is open. Test your

code using a chosen salary value of 5000.

2. SELECT from the PROPOSED\_RAISES table to see the results of your INSERT statements.

There should be 15 rows. If you run your block in question 1 more than once, make sure the

PROPOSED\_RAISES table is empty before each test.

SELECT \* FROM proposed\_raises;

DELETE FROM proposed\_raises; -- to clear all rows from the table

Before continuing, ensure there are 15 rows in PROPOSED\_RAISES.

3. Imagine these proposed salary increases have been approved by company management.

A. Write and execute a PL/SQL block to read each row from the PROPOSED\_RAISES table. For

each row, UPDATE the date\_approved column with today’s date. Use the WHERE CURRENT

OF... syntax to UPDATE each row. After running your code, SELECT from the

PROPOSED\_RAISES table to view the updated data.

B. Management has now decided that employees in department 50 cannot have a salary

increase after all. Modify your code from question 3 to DELETE employees in department 50

from PROPOSED\_RAISES. This could be done by a simple DML statement (DELETE FROM

proposed\_raises WHERE department\_id = 50;), but we want to do it using a FOR UPDATE

cursor. Test your code, and view the PROPOSED\_RAISES table again to check that the rows

have been deleted.

4. Since Oracle Academy's Application Express automatically commits changes, complete the

following activity as if you were issuing the commands in an installed/local environment with the

ability to use COMMIT and ROLLBACK. The indicated errors and pauses will not actually happen

in the Oracle Academy's online Application Express.

We are going to set up two sessions into the same schema. From one of the sessions we will

manually update an employee row NOT COMMITING. From the other session we will try to

update everyone’s salary, again NOT COMMITING. You should see the difference between

NOWAIT and WAIT when using FOR UPDATE.

In preparation, create a copy of the employees table by executing the following SQL statement. You should

use the UPD\_EMPS table for the rest of this exercise.

CREATE TABLE upd\_emps AS SELECT \* FROM employees;

A. Open a second session in a new browser window and connect to your schema.

B. In your first session, update employee\_id 200 (Jennifer Whalen) so the stored first name is Jenny.

DO NOT COMMIT. You now have a lock on row 200 that will last indefinitely.

C. In your second session, write a PL/SQL block to give every employee in UPD\_EMPS a $1 salary

raise. Your cursor should be declared FOR UPDATE NOWAIT. Execute your code. What happens?

D. Still in your second session, modify your block to remove the NOWAIT attribute from the cursor

declaration. Re-execute the block. What happens this time?

E. After waiting a minute or so, switch to your first session and COMMIT the update to Jennifer

Whalen’s row. Then switch back to your second session. What happened?