

## SELECT による FOR UPDATE 句とオプション (PL/SQL)

(例 1)

```
DECLARE
  p_dept emp_table.empno%TYPE ;           -- WHERE 文中の値指定用変数定義
emp_record emp_table%ROWTYPE ;       -- レコード型変数の定義
/* カーソルの定義 */
CURSOR c_emp IS
  SELECT * FROM emp_table WHERE empno = p_dept FOR
    UPDATE ;                             -- 更新ロックが掛るまで待機する
    FOR UPDATE NOWAIT ;                 -- 他者がロックの時は例外が発生
    FOR UPDATE n ;                       -- ロック獲得のため n 秒間だけ待機

BEGIN
  p_dept := 4 ;
OPEN e_emp ;                         -- この時点で排他的ロックが実施
  FOR r_emp IN c_emp LOOP
    /* r_emp がレコード型変数として自動定義され、レコードの値がループごとに
       セットされる */
    DBMS_OUTPUT.PUT_LINE('名前: ' || r_emp.ename);
LOOP
  FETCH e_emp INTO emp_record ;     -- フェッチ
  EXIT WHEN e_emp%NOTFOUND ;       -- フェッチ失敗判断
    /* データが存在した時の処理 */
  END LOOP ;
CLOSE e_emp ;                       -- カーソル・クローズ
END ;
```

---- 実行結果 ----

名前: PENTA

(例 2) 複数テーブルの select に対し、**特定テーブルだけロック**したい場合  
**FOR UPDATE OF 列名**で、指定した列名を持ったテーブルだけがロック対象となる  
【注意】ロックは、列に対してではなく 対象テーブルのレコード全体に対し行  
われる

```
DECLARE
  /* カーソルの定義 */
  CURSOR c_emp IS
    SELECT emp_table.* , kubun_name FROM emp_table, kubun_table
    WHERE emp_no = kubun_no FOR UPDATE OF 列名 WAIT n ;

  /* 例えば、emp_no を指定すると、更新ロックの対象テーブルは、emp_table
  だけになる */
  /* 【注意】ロックは、列に対してではなく レコードに対し行われる */

BEGIN
  p_dept := 10 ;
  OPEN c_emp ; -- この時点で排他的ロックが実施
  FOR r_emp IN c_emp LOOP
    /* r_emp がレコード型変数として自動定義され、レコードの値がループごとに
    セットされる */
    /* r_emp の中には、kubun_name 列も含まれてくる */

  END LOOP ;
END ;
```

実行例)

```
DECLARE
  p_dept emp_table.empno%TYPE ;
  /* カーソルの定義 */
  CURSOR c_emp IS
  SELECT * FROM emp_table WHERE empno = p_dept
  FOR UPDATE OF empno ;

BEGIN
  p_dept := 4 ;
  FOR r_emp IN c_emp LOOP
    /* r_emp がレコード型変数として自動定義され、レコードの値がループごとに
    セットされる */

    DBMS_OUTPUT.PUT_LINE('名前: ' || r_emp.ename) ;
  END LOOP;
END ;
/
```

---- 実行結果 ----

名前: PENTA