

7 Дополнительные возможности курсоров

Курсоры с параметрами

```
CURSOR cursor_name [ (parameter_name  
datatype, ...) ]  
IS select_statement;
```

С помощью данной конструкции можно передавать параметры курсору. Курсор может выполняться с различными параметрами.

```
OPEN cursor_name (parameter_value, .....);
```

В момент открытия курсора производится разбор команды SELECT, а параметр связывается со значением. Затем определяется результирующий набор строк — и курсор готов к выборке.

Пример

Передача, в виде параметра курсору, номера отдела и должности сотрудника.

```
DECLARE
    CURSOR emp_cursor (p_deptno NUMBER, p_job VARCHAR2)
    IS
        SELECT empno, ename FROM emp
        WHERE dept = p_deptno AND job = p_job;
BEGIN
    OPEN emp_cursor (80, 'SA_REP');
    ...
    CLOSE emp_cursor;
END;
```

Параметрам курсоров могут присваиваться значения по умолчанию. Пример курсора со значением параметра по умолчанию: `CURSOR emp_cur (emp_id_in NUMBER := 0)`

Конструкция FOR UPDATE

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

При выполнении команды SELECT строки не блокируются. Блокируются строки DML командами, пока не зафиксированы COMMIT, ROLLBACK.

SELECT...FOR UPDATE - все идентифицируемые курсором строки блокируются при его открытии. Никто не сможет менять эти строки до ROLLBACK или COMMIT, другие сеансы могут читать данные (Демонстрация в сноске).

По списку OF, блокируются строки таблиц, строки которых указываются в списке. Без OF, Oracle блокирует все отобранные строки всех таблиц секции FROM.

NOWAIT. Если таблица заблокирована другим пользователем, Oracle ждет ее освобождения. Управление будет возвращено программе. WAIT с указанием максимальной продолжительности ожидания блокировки в секундах. Если ни одно из этих ключевых слов не указано, сеанс блокируется до тех пор, пока таблица не освободится.

Пример

```
DECLARE  
CURSOR emp_cursor IS  
    SELECT empno, ename, sal FROM emp  
    WHERE deptno = 30  
FOR UPDATE OF sal NOWAIT;
```

Конструкция WHERE CURRENT OF

```
UPDATE имя_таблицы  
SET предложение_set  
WHERE CURRENT OF имя_курсора;
```

```
DELETE  
FROM имя_таблицы  
WHERE CURRENT OF имя_курсора;
```

- Использование WHERE CURRENT OF, для модификации или удаления текущей выбранной из курсора строки данных.
- Использование утверждения WHERE CURRENT OF позволяет ссылаться на текущую строку курсора.

Пример

```
DECLARE
CURSOR sal_cursor IS
    SELECT sal FROM emp WHERE deptno = 30
    FOR UPDATE OF sal NOWAIT;
BEGIN
    FOR emp_record IN sal_cursor
    LOOP
        UPDATE emp SET sal = emp_record.sal*1.10
        WHERE CURRENT OF sal_cursor;
    END LOOP;
END;
```

Курсоры с подзапросами

```
CURSOR my_cursor IS
  SELECT t1.deptno, t1.dname, t2.staff
  FROM dept t1,
       (SELECT deptno, count(*) as STAFF
        FROM emp GROUP BY deptno) t2
  WHERE t1.deptno = t2.deptno
        AND t2.staff >= 5;
  ...
```

Курсорные переменные и REF CURSOR

Курсорная переменная ссылается на курсор. В отличие от явного курсора, имя которого в PL/SQL используется как идентификатор рабочей области результирующего набора строк, курсорная переменная содержит ссылку на эту рабочую область. Явные и неявные курсоры имеют статическую природу, поскольку они жестко привязаны к конкретным запросам. С помощью же курсорной переменной можно выполнить любой запрос и даже несколько разных запросов в одной программе.

```
DECLARE
    TYPE var_cur_t IS REF CURSOR;
    var_cur var_cur_t;
BEGIN
    OPEN var_cur FOR SELECT ...
    ...
    CLOSE var_cur;
END;
```

Курсорные переменные и REF CURSOR

```
DECLARE
    TYPE cur_type IS REF CURSOR;
    v_cur cur_type;
BEGIN
    OPEN v_cur FOR SELECT * FROM employees;
    ....
    OPEN v_cur FOR SELECT name FROM company;
    ....
END;
```

Курсорные переменные и REF CURSOR

Демонстрация кода

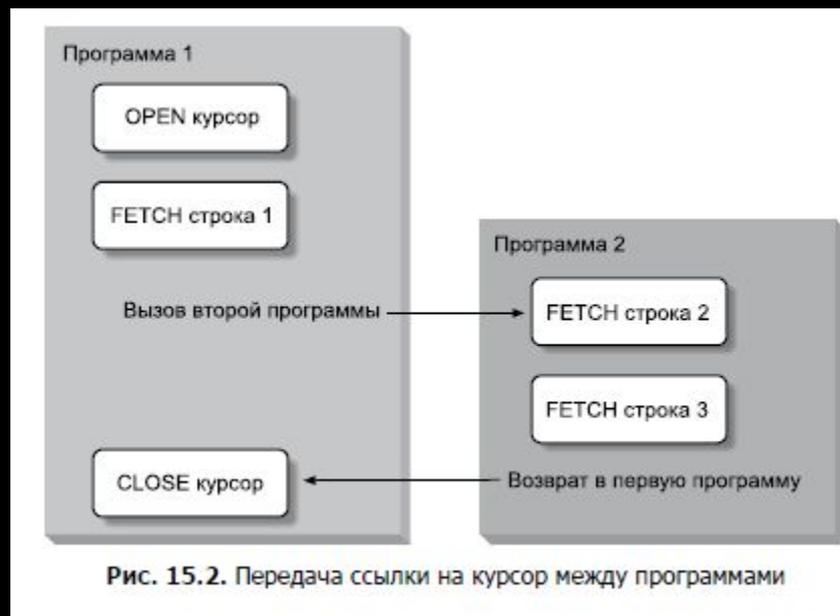
```
DECLARE
  TYPE cur_type IS REF CURSOR;
  v_cur cur_type;
  v_row_e employees%rowtype;
  v_row_d departments%rowtype;
BEGIN
  dbms_output.put_line('START employees fetch');
  OPEN v_cur FOR SELECT * FROM employees;
  LOOP
    FETCH v_cur INTO v_row_e;
    dbms_output.put_line(v_row_e.LAST_NAME || ' - ' || v_row_e.SALARY);
    EXIT WHEN v_cur%NOTFOUND;
  END LOOP;
  CLOSE v_cur;
  dbms_output.put_line('START departments fetch');

  OPEN v_cur FOR SELECT department_name FROM departments;
  LOOP
    FETCH v_cur INTO v_row_d.department_name;
    dbms_output.put_line(v_row_d.department_name);
    EXIT WHEN v_cur%NOTFOUND;
  END LOOP;
  CLOSE v_cur;
END;
```

Курсорные переменные и REF CURSOR

Преимуществом курсорных переменных является то, что они предоставляют механизм передачи результатов запроса (выбранных из строк курсора) между программами PL/SQL, в том числе между клиентскими и серверными программами. Курсорная переменная позволяет передать другой программе ссылку на объект курсора, чтобы та могла работать с его данными.

С курсорными переменными может использоваться операция присваивания



Курсорные переменные и SYS_REFCURSOR

```
DECLARE  
    my_cursor SYS_REFCURSOR;
```

С Oracle 9 появился предопределенный тип REF CURSOR с именем SYS_REFCURSOR. Не нужно определять собственный тип — достаточно использовать стандартный тип Oracle.

Нет разницы между REF CURSOR и SYS_REFCURSOR , так как по факту определение sys_refcursor такое : type sys_refcursor is ref cursor;

Курсорные переменные и SYS_REFCURSOR

```
DECLARE
    v_cur SYS_REFCURSOR; -- TYPE cur_type IS REF CURSOR;
    v_row_e employees%rowtype;
    v_row_d departments%rowtype;
BEGIN
    OPEN v_cur FOR SELECT * FROM employees;
    LOOP
        FETCH v_cur INTO v_row_e;
        dbms_output.put_line(v_row_e.LAST_NAME || ' - '
||v_row_e.SALARY);
        EXIT WHEN v_cur%NOTFOUND;
    END LOOP;
    CLOSE v_cur;
    OPEN v_cur FOR SELECT department_name FROM departments;
    LOOP
        FETCH v_cur INTO v_row_d.department_name;
        dbms_output.put_line(v_row_d.department_name);
        EXIT WHEN v_cur%NOTFOUND;
    END LOOP;
    CLOSE v_cur;
END;
```

Итоги

- Использование курсоров с параметрами
- Определение курсоров с подзапросами и коррелированными подзапросами.
- Команды для курсоров
 - FOR UPDATE
 - WHERE CURRENT OF

Практика №7!

40 минут