

6

Явные курсоры

Понятие курсора. SQL курсор.

При выполнении команды SQL из PL/SQL PCУБД Oracle назначает ей приватную рабочую область, а некоторые данные записывает в системную глобальную область (SGA, System Global Area). В приватной рабочей области содержится информация о команде SQL и набор данных, возвращаемых или обрабатываемых этой командой. Таким образом любая инструкция SQL, выполняемая Oracle server имеет индивидуальный курсор с ней ассоциированный. Курсор проще всего представить себе как указатель на виртуальную таблицу в базе данных с данными определяемыми SQL командой.

Имеются два типа курсоров:

- **Неявные курсоры:** объявленные в PL/SQL неявно. Курсор этого типа называется неявным, поскольку Oracle автоматически выполняет многие связанные с ним операции, такие как открытие, выборка данных и даже закрытие.
- **Явные курсоры:** объявленные и названные разработчиком. Это SELECT, явно определенный в программе как курсор. Все операции с явным курсором (открытие, выборка данных, закрытие и т. д.) в программе должны выполняться явно. Как правило, явные курсоры используются для выборки из базы данных набора строк с использованием SQL.

Понятие курсора. SQL курсор.

Объявление связывает таблицу SELECT с курсором.

Объявленный курсор можно открыть.

Далее из него можно выбирать строки.

Завершив работу с курсором, его следует закрыть.

```
CURSOR emp_cur  
IS SELECT * FROM employee;  
OPEN emp_cur;  
FETCH emp_cur INTO xxx;  
CLOSE emp_cur;
```

-С курсором можно связать любую команду SELECT.

-Курсор действует как указатель на эти строки выборки

-Атрибут курсора имеет форму %имя_атрибута , добавляется к имени курсора . Это что-то вроде внутренней переменной Oracle, возвращающей информацию о состоянии курсора — например о том, открыт ли курсор, или сколько строк из курсора вернул запрос. У явных и неявных курсоров и SQL в атрибутах курсоров существуют различия

Функции явного курсора

Активный набор



7369	SMITH	CLERK
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7566	JONES	MANAGER
7654	MARTIN	SALESMAN
7698	BLAKE	MANAGER

Текущая запись

Управление явным курсором

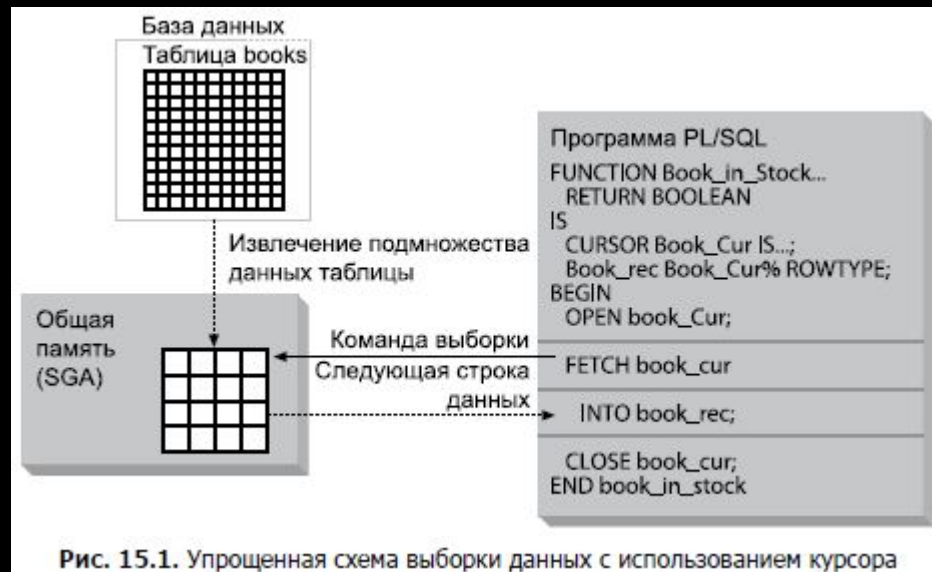
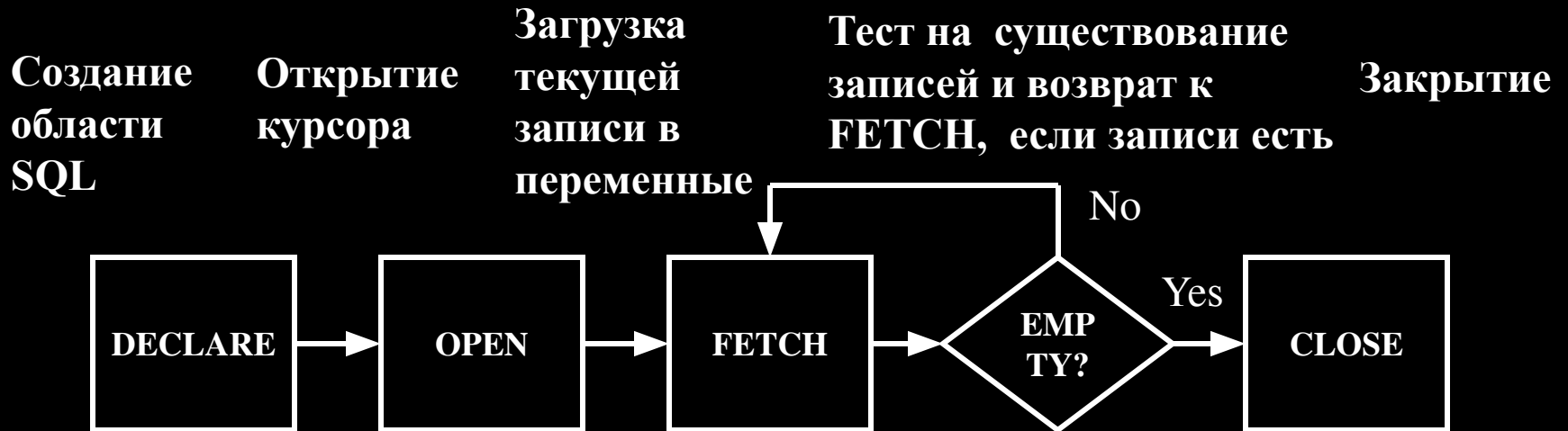


Рис. 15.1. Упрощенная схема выборки данных с использованием курсора

Объявление курсора

```
CURSOR cursor_name IS  
    select_statement;
```

- При описании курсора фраза INTO не включается.
- Для обработки строк в требуемой последовательности, используется предложение ORDER BY в запросе.

Пример

```
DECLARE
```

```
    CURSOR emp_cursor IS
```

```
        SELECT empno, ename      FROM emp;
```

```
    CURSOR dept_cursor IS
```

```
        SELECT * FROM dept WHERE deptno = 10;
```

```
BEGIN
```

```
...
```

Открытие курсора

```
OPEN cursor_name;
```

- При открытии курсора, выполняется запрос и выделяется активный набор.
- Атрибуты курсора используются для контроля выборки.

FETCH из курсора

```
FETCH cursor_name INTO [variable1,  
variable2,...] | record_name];
```

- Выборка текущей строки осуществляется в переменные.
- Должно быть то же самое число переменных, что и количество полей в запросе SELECT.
- Каждая переменная должна соответствовать каждому выбираемому столбцу.
- Должна быть обеспечена проверка на наличие записей в курсоре.

Пример

```
DECLARE
    CURSOR emp_cursor IS
        SELECT empno, ename FROM emp;
    v_empno NUMBER(6,0);
    v_ename VARCHAR2(20);
BEGIN
    OPEN emp_cursor
    LOOP
        FETCH emp_cursor INTO v_empno, v_ename;
        EXIT WHEN ... ;

        ...
    END LOOP;

    ...
END;
```

Close

```
CLOSE cursor_name;
```

- Курсор закрывается после завершения процесса выборки строк.
- Возможно повторное открытие, если требуется.
- Данные не доступны для выбора после того, как курсор закрыт.

Пример

```
...  
OPEN emp_cursor  
  FOR i IN 1..10 LOOP  
    FETCH emp_cursor INTO v_empno, v_ename;  
    ...  
  END LOOP;  
CLOSE emp_cursor;  
...
```

Атрибуты явного курсора

Таблица 15.1. Атрибуты курсоров

Имя	Что возвращает
%FOUND	TRUE, если успешно выбрана хотя бы одна строка; в противном случае возвращает FALSE
%NOTFOUND	TRUE, если команда не выбрала ни одной строки; в противном случае возвращает FALSE
%ROWCOUNT	Количество строк, выбранных из курсора на данный момент времени
%ISOPEN	TRUE, если курсор открыт; в противном случае возвращает FALSE
%BULK_ROWCOUNT	Количество измененных записей для каждого элемента исходной коллекции, заданной в команде FORALL
%BULK_EXCEPTIONS	Информация об исключении для каждого элемента исходной коллекции, заданной в команде FORALL

PL/SQL поддерживает шесть атрибутов курсоров для получения информации о состоянии курсора. Чтобы обратиться к атрибуту курсора, указывается в виде префикса имя курсора и символ %:

имя_курсора%имя_атрибута

В качестве имен неявных курсоров используется префикс SQL, например SQL%NOTFOUND.

Атрибуты явного курсора

Таблица 15.4. Значения атрибутов курсоров

Операция	%FOUND	%NOTFOUND	%ISOPEN	%ROWCOUNT
До OPEN	Исключение ORA-01001	Исключение ORA-01001	FALSE	Исключение ORA-01001
После OPEN	NULL	NULL	TRUE	0
До первой выборки FETCH	NULL	NULL	TRUE	0
После первой выборки FETCH	TRUE	FALSE	TRUE	1
Перед последующими FETCH	TRUE	FALSE	TRUE	1
После последующих FETCH	TRUE	FALSE	TRUE	Зависит от данных
Перед последней выборкой FETCH	TRUE	FALSE	TRUE	Зависит от данных
После последней выборки FETCH	TRUE	FALSE	TRUE	Зависит от данных
Перед CLOSE	FALSE	TRUE	TRUE	Зависит от данных
После CLOSE	Исключение	Исключение	FALSE	Исключение

Атрибут %ISOPEN

- Строки выбираются только в том случае, если курсор открыт.
- %ISOPEN атрибут используется для проверки состояния открытого курсора (TRUE/FALSE).

```
IF NOT emp_cursor%ISOPEN THEN
  OPEN emp_cursor;
END IF;
LOOP
  FETCH emp_cursor...
```

Управление многократной выборкой

- Обрабатывается несколько строк явного курсора, используя цикл.
- Выборка записей с каждой итерацией.
- Использование атрибута %NOTFOUND для определения невыполненной выборки.
- Использование атрибутов явного курсора для определения состояния нормальной выборки данных.

%NOTFOUND и %ROWCOUNT

- Атрибут %ROWCOUNT используется для подсчета числа строк, выбранных курсором.
- Атрибут %NOTFOUND используется для определения момента выхода из цикла (при отсутствии записей в курсоре).

```
DECLARE
  v_s employees.salary%TYPE;
  CURSOR happiness_cur IS SELECT salary FROM employees;
BEGIN
  OPEN happiness_cur;
  dbms_output.put_line(happiness_cur%ROWCOUNT);
  fetch happiness_cur into v_s;
  dbms_output.put_line(happiness_cur%ROWCOUNT);
  close happiness_cur;
  --dbms_output.put_line(happiness_cur%ROWCOUNT);
END;
```

0
1
ORA-01001: неверный курсор ORA-06512: на line 10

Пример

```
...  
OPEN emp_cursor;  
LOOP  
    FETCH emp_cursor INTO v_empno, v_ename;  
    EXIT WHEN emp_cursor%ROWCOUNT > 10  
        OR  
        emp_cursor%NOTFOUND;  
...  
END LOOP;  
CLOSE emp_cursor;
```

Курсоры и записи

Выборка записей из курсора в курсорную переменную:

```
DECLARE
    CURSOR emp_cursor IS
        SELECT empno, ename FROM emp;
    emp_record emp_cursor%ROWTYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO emp_record;
    ...
```

Курсорный цикл FOR и неявное использование курсора

```
FOR record_name IN cursor_name LOOP  
    statement1;  
    statement2;  
    ...  
END LOOP;
```

- Служит для обработки явных курсоров.
- Неявно происходит открытие, выборка и закрытие курсора.

Пример

```
DECLARE
  CURSOR emp_cursor IS
    SELECT ename, dept FROM emp;
BEGIN
  FOR emp_record IN emp_cursor
  LOOP      -- неявное открытие курсора и выборка
    IF emp_record.deptno = 30 THEN
      ...
    END LOOP; -- неявное закрытие курсора
END;
```

Пример 2

```
DECLARE
-- ничего не надо объявлять
BEGIN
  FOR emp_record IN (SELECT ename, deptno FROM emp)
  LOOP
    IF emp_record.deptno = 30 THEN
      ...
    END LOOP;
  END;
END;
```

ИТОГИ

- Курсоры
 - Явные
 - Неявные
- Управление явными курсорами
- Использование атрибутов курсора
- Цикл For для курсора

Практика №6!

40 минут