

Использование явных курсоров

Рассматриваемые вопросы

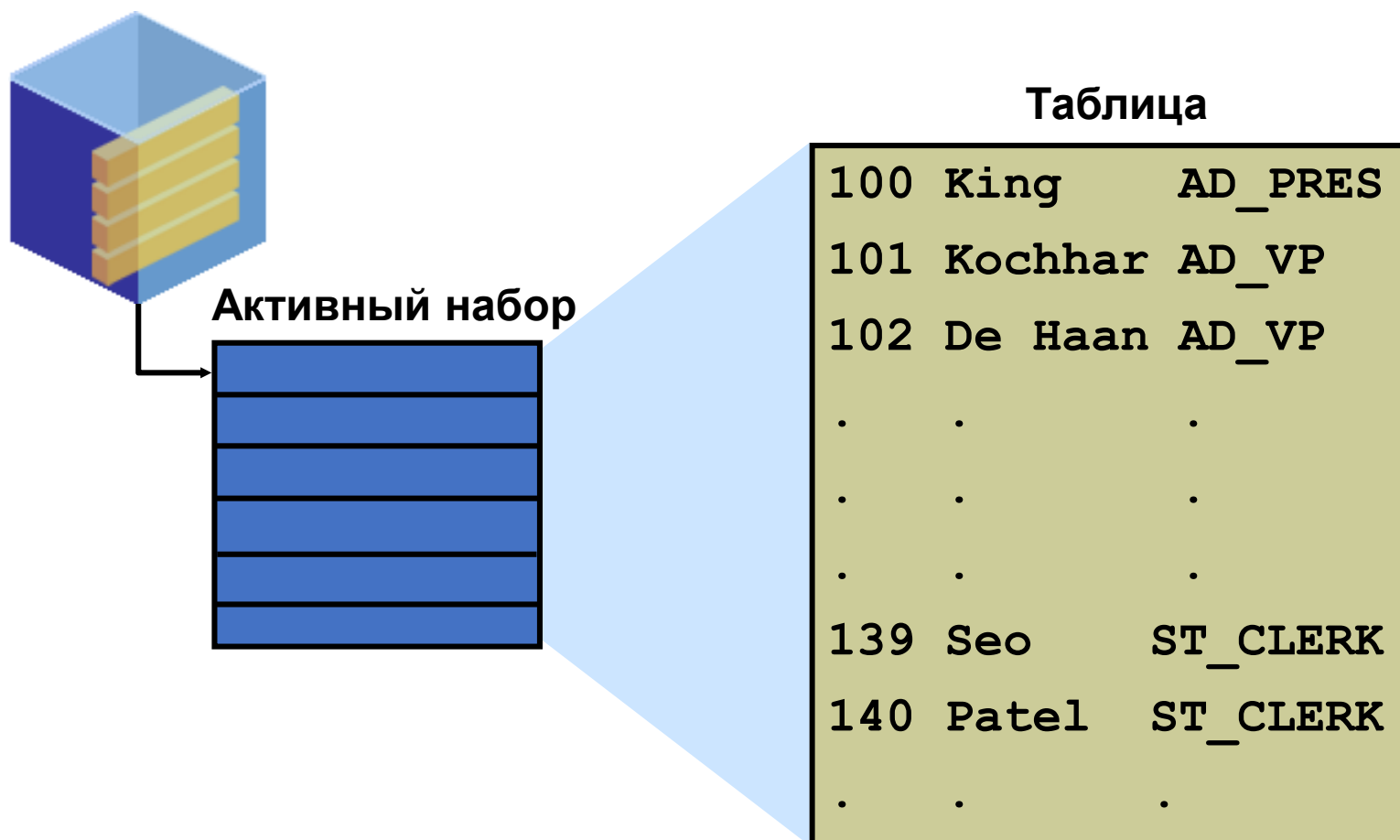
- Разница между неявными и явными курсорами
- Когда и для чего используются явные курсоры
- Объявление и управление явными курсорами
- Использование для выбора данных простого цикла и цикла `FOR` с курсором
- Объявление и использование курсоров с параметрами
- Блокировка строк с помощью предложения `FOR UPDATE`
- Ссылка на текущую строку с помощью предложения `WHERE CURRENT`

О курсорах

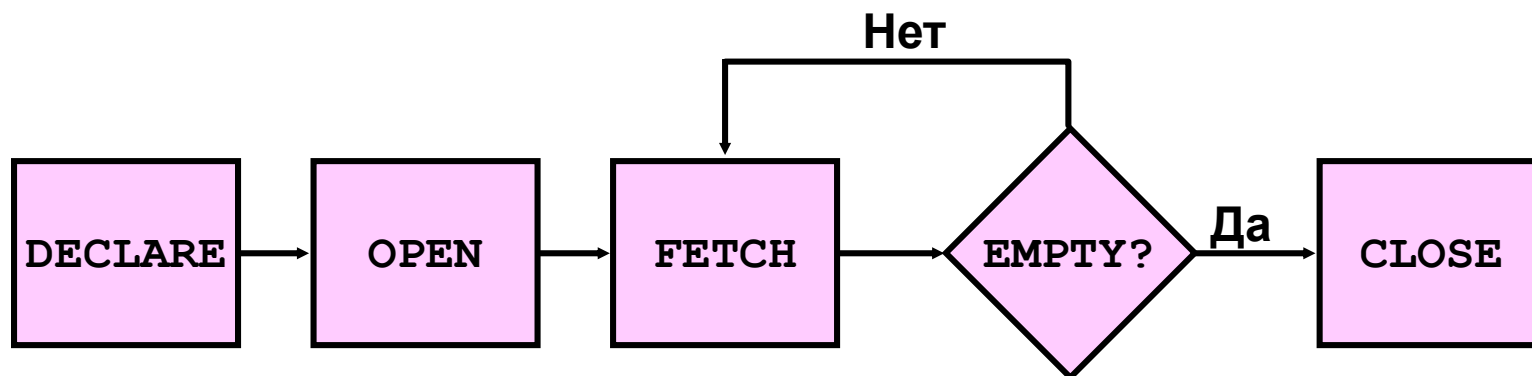
- Каждая команда SQL, выполняемая сервером Oracle, имеет отдельный, связанный с ней курсор.
 - Неявные курсоры: PL/SQL объявляет и использует такие курсоры для управления всеми командами DML и `SELECT`.
 - Явные курсоры: объявляются и контролируются программистом.



Операции с явными курсорами



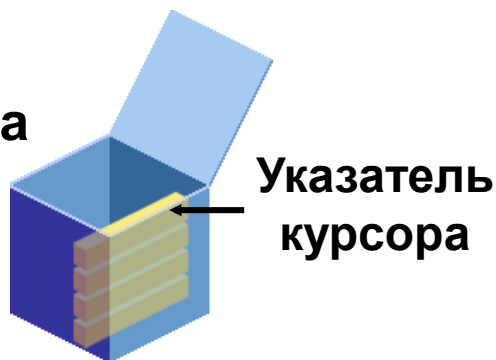
Управление явными курсорами



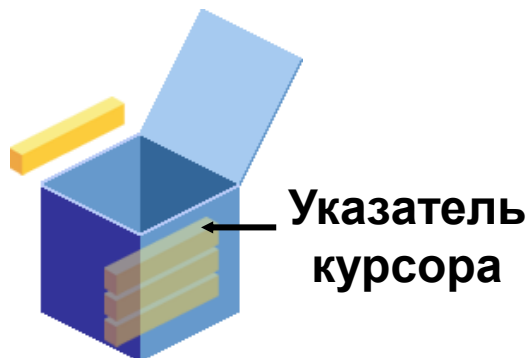
- Создание именованной рабочей области SQL
- Выявление активного набора
- Загрузка текущей строки в переменные
- Проверка на наличие строк
- Освобождение активного набора
- Возврат к FETCH, если строки обнаружены

Управление явными курсорами

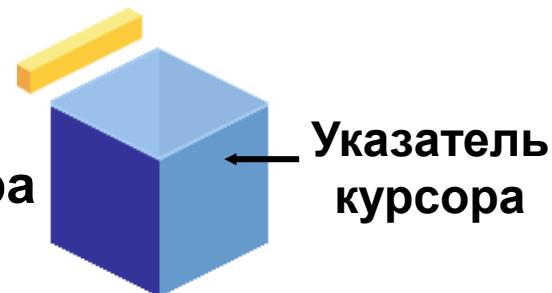
1 Открытие курсора



2 Выборка строки



3 Заккрытие курсора



Объявление курсора

- Синтаксис:

```
CURSOR имя_курсора IS  
    команда_select;
```

Примеры:

```
DECLARE  
    CURSOR emp_cursor IS  
        SELECT employee_id, last_name FROM employees  
        WHERE department_id = 30;
```

```
DECLARE  
    locid NUMBER:= 1700;  
    CURSOR dept_cursor IS  
        SELECT * FROM departments  
        WHERE location_id = locid;  
    ...
```

Открытие курсора

```
DECLARE
  CURSOR emp_cursor IS
    SELECT employee_id, last_name FROM employees
    WHERE department_id =30;
  ...
BEGIN
  OPEN emp_cursor;
```


Выборка данных из курсора

```
SET SERVEROUTPUT ON
DECLARE
  CURSOR emp_cursor IS
    SELECT employee_id, last_name FROM employees
    WHERE department_id = 30;
  empno employees.employee_id%TYPE;
  lname employees.last_name%TYPE;
BEGIN
  OPEN emp_cursor;
  FETCH emp_cursor INTO empno, lname;
  DBMS_OUTPUT.PUT_LINE( empno || ' ' || lname);
  ...
END;
/
```

Выборка данных из курсора

```
SET SERVEROUTPUT ON
DECLARE
    CURSOR emp_cursor IS
        SELECT employee_id, last_name FROM employees
        WHERE department_id = 30;
    empno employees.employee_id%TYPE;
    lname employees.last_name%TYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO empno, lname;
        EXIT WHEN emp_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE( empno || ' ' || lname);
    END LOOP;
    ...
END;
/
```

Заккрытие курсора

```
...  
  LOOP  
    FETCH emp_cursor INTO empno, lname;  
    EXIT WHEN emp_cursor%NOTFOUND;  
    DBMS_OUTPUT.PUT_LINE( empno || ' ' || lname);  
  END LOOP;  
  CLOSE emp_cursor;  
END;  
/
```

Курсоры и записи

- Строки из активного набора удобно обрабатывать, выбирая значения в переменную PL/SQL типа RECORD.

```
DECLARE
  CURSOR emp_cursor IS
    SELECT employee_id, last_name FROM employees
    WHERE department_id =30;
  emp_record emp_cursor%ROWTYPE;
BEGIN
  OPEN emp_cursor;
  LOOP
    FETCH emp_cursor INTO emp_record;
    ...
```

Циклы FOR с курсором

- Синтаксис:

```
FOR имя_записи IN имя_курсора LOOP  
    команда1;  
    команда2;  
    . . .  
END LOOP;
```

- Сокращенная форма кодирования операций с явными курсорами.
- Неявное открытие, выборка и закрытие.
- Запись объявляется неявно.

Циклы FOR с курсором

```
SET SERVEROUTPUT ON
DECLARE
    CURSOR emp_cursor IS
        SELECT employee_id, last_name FROM employees
        WHERE department_id = 30;
BEGIN
    FOR emp_record IN emp_cursor
    LOOP
        DBMS_OUTPUT.PUT_LINE( emp_record.employee_id
            || ' ' || emp_record.last_name );
    END LOOP;
END;
/
```

Атрибуты явного курсора

- Информацию о состоянии курсора можно получить с помощью атрибутов явного курсора.

| Атрибут | Тип | Описание |
|------------------------|---------|---|
| <code>%ISOPEN</code> | Boolean | Истинно (TRUE), если курсор открыт |
| <code>%NOTFOUND</code> | Boolean | Истинно (TRUE), если последняя команда <code>FETCH</code> не возвращает строку |
| <code>%FOUND</code> | Boolean | Истинно (TRUE), пока последняя команда <code>FETCH</code> возвращает строку; дополняет <code>%NOTFOUND</code> |
| <code>%ROWCOUNT</code> | Number | Общее количество строк, выбранных на данный момент |

Атрибут %ISOPEN

- Выборка строк возможна только при открытом курсоре.
 - Прежде, чем выполнять операцию `FETCH`, проверьте с помощью атрибута `%ISOPEN`, открыт ли курсор.
- Пример:

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


Пример использования атрибутов %ROWCOUNT и %NOTFOUND

```
SET SERVEROUTPUT ON
DECLARE
    empno    employees.employee_id%TYPE;
    ename    employees.last_name%TYPE;
    CURSOR emp_cursor IS SELECT employee_id,
    last_name FROM employees;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO empno, ename;
        EXIT WHEN emp_cursor%ROWCOUNT > 10 OR
                emp_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (TO_CHAR(empno)
                                || ' ' || ename);
    END LOOP;
    CLOSE emp_cursor;
END ;
/
```

Циклы FOR с курсорами, в которых используется подзапрос

- Объявлять курсор не требуется.
- Пример:

```
SET SERVEROUTPUT ON
BEGIN
  FOR emp_record IN (SELECT employee_id, last_name
                     FROM employees WHERE department_id =30)
  LOOP
    DBMS_OUTPUT.PUT_LINE( emp_record.employee_id || '
    ' || emp_record.last_name);
  END LOOP;
END;
/
```

Курсоры с параметрами

- Синтаксис:

```
CURSOR имя_курсора  
  [ (имя_параметра тип_данных, ...) ]  
IS  
  команда_select;
```

- Значения параметров передаются в курсор, когда курсор открывается и выполняется запрос.
- Явный курсор можно открывать многократно, получая каждый раз новый активный набор.

```
OPEN   имя_курсора (значение_параметра, . . . . .) ;
```

Курсоры с параметрами

```
SET SERVEROUTPUT ON
DECLARE
    CURSOR    emp_cursor (deptno NUMBER) IS
        SELECT employee_id, last_name
        FROM    employees
        WHERE   department_id = deptno;
    dept_id NUMBER;
    lname    VARCHAR2(15);
BEGIN
    OPEN emp_cursor (10);
    ...
    CLOSE emp_cursor;
    OPEN emp_cursor (20);
    ...
```

Предложение FOR UPDATE

- Синтаксис:

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

- Явная блокировка позволяет запретить доступ к данным в течение всей транзакции.
- Строки должны блокироваться до операции обновления или удаления.

Предложение WHERE CURRENT OF

- Синтаксис:

```
WHERE CURRENT OF курсор ;
```

- Курсоры используются для обновления или удаления текущей строки.
- Предложение FOR UPDATE включается в запрос курсора для предварительной блокировки строк.
- Предложение WHERE CURRENT OF используется для ссылки на текущую строку явного курсора.

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

Курсоры с подзапросами

Пример:

```
DECLARE
  CURSOR my_cursor IS
    SELECT t1.department_id, t1.department_name,
           t2.staff
    FROM   departments t1, (SELECT department_id,
                                   COUNT(*) AS STAFF
                           FROM employees
                           GROUP BY department_id) t2
    WHERE  t1.department_id = t2.department_id
    AND    t2.staff >= 3;
...

```

ИТОГИ

- Различные типы курсоров:
 - Неявные курсоры: используются для выполнения всех команд DML и однострочных запросов
 - Явные курсоры: используются для запросов, не возвращающих ничего, возвращающих одну или несколько строк
- Создание и обработка явных курсоров
- Использование простого цикла и цикла FOR с курсором для обработки множества строк, определяемых курсором.
- Определение состояния курсора с помощью его атрибутов
- Использование предложений FOR UPDATE и WHERE CURRENT OF для обновления и удаления текущей выбираемой строки