

Oracle Core

Тема 8

**Коллекции и курсоры**

(часть 2)

# Содержание

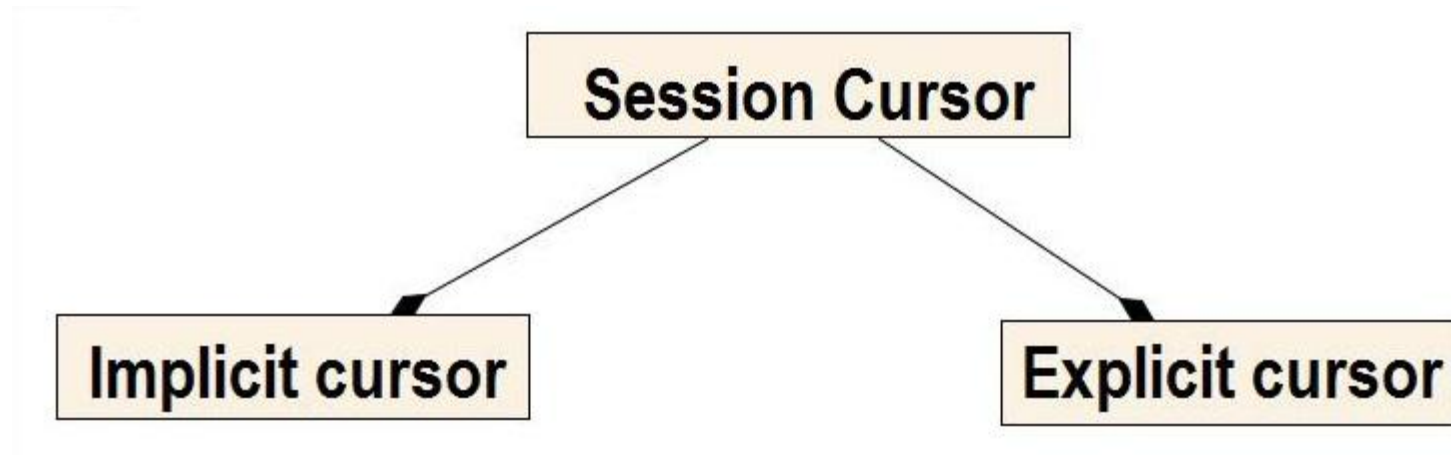
- ✓ Implicit cursor
- ✓ Explicit cursor
- ✓ Операции с cursor
- ✓ Атрибуты cursor
- ✓ Ref cursor
- ✓ Параметры БД, словари данных

# Cursors



# Cursor

**Курсор** – указатель на приватную SQL область, в которой содержится информация о выполнении конкретного select-оператора (или любого другого dml оператора)



# SQL Cursor

- ✓ PL/SQL открывает implicit cursor каждый раз, когда выполняет select или dml-оператор. Сразу после выполнения закрывает его.

- ✓ Мы не

```
CREATE OR REPLACE PROCEDURE p(dept_no  
NUMBER) AUTHID DEFINER AS  
BEGIN
```

- ✓ К неко

```
    DELETE FROM dept_temp  
    WHERE department_id = dept_no;
```

- ✓ SQL%IS

```
    IF SQL%FOUND  
    THEN
```

- ✓ SQL%F

```
        DBMS_OUTPUT.PUT_LINE('Delete  
succeeded for department number ' ||  
dept_no);
```

- ✓ SQL%M

```
    ELSE  
        DBMS_OUTPUT.PUT_LINE('No department  
number ' || dept_no);
```

- ✓ SQL%R

```
    END IF;  
END;
```

# Explicit cursor

✓ Именованный курсор (named cursor)

✓ Объявление курсора

`CURSOR cursor_name [ parameter_list ] [ RETURN return_type ] IS select_statement;`

**DECLARE**

**CURSOR c1 IS**

**SELECT** employee\_id,  
job\_id,  
salary

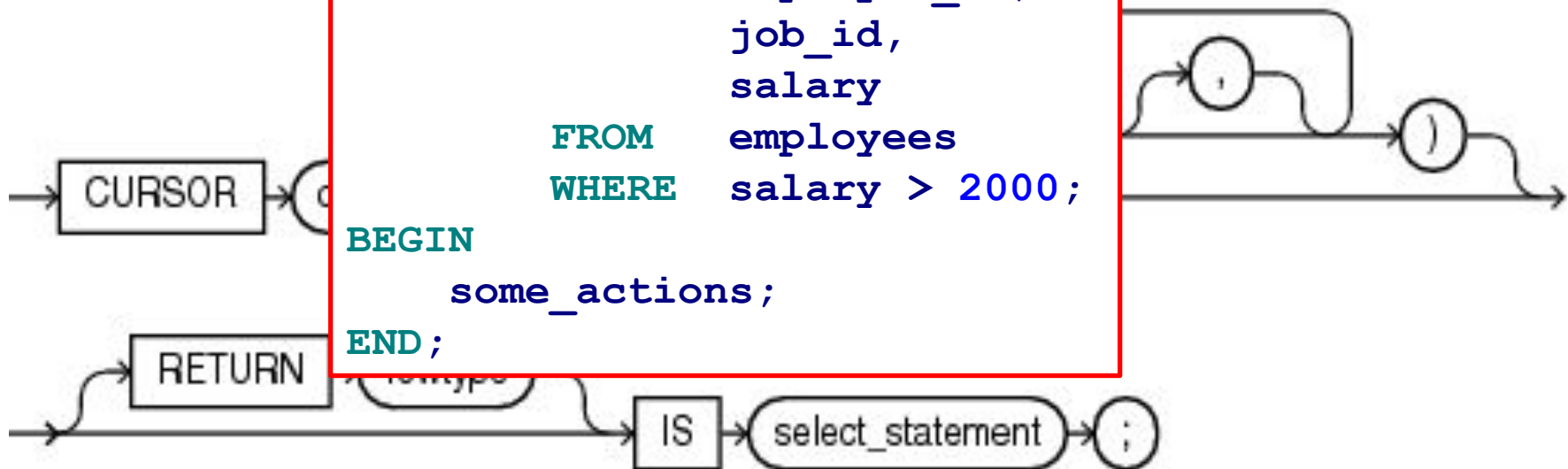
**FROM** employees

**WHERE** salary > 2000;

**BEGIN**

some\_actions;

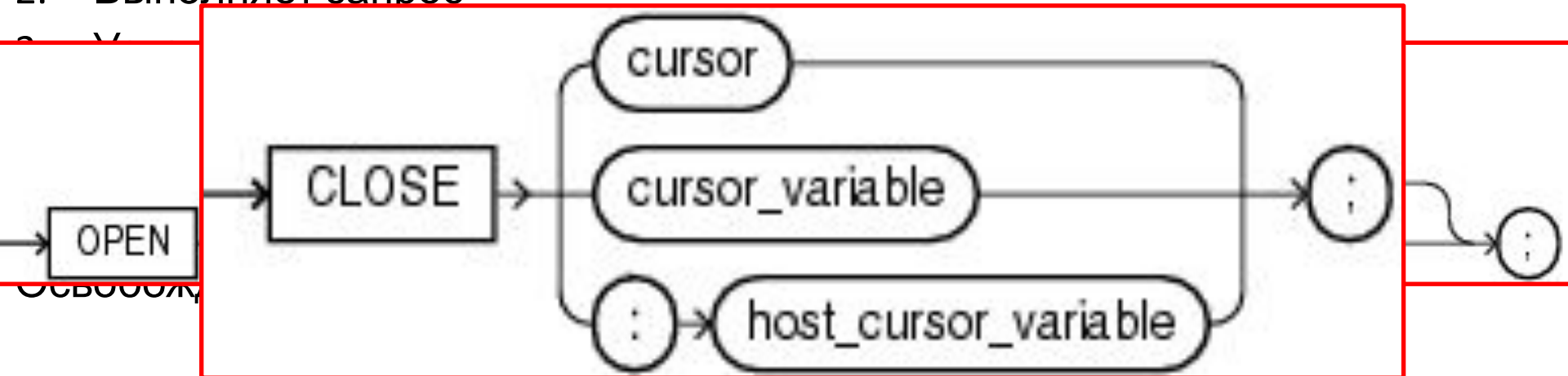
**END;**



# Open cursor

**open cursor;**

1. Выделяет ресурсы базы данных для выполнения запроса
2. Выполняет запрос



## Exceptions

2. INVALID\_CURSOR
3. CURSOR\_ALREADY\_OPEN

# Fetching data

```
DECLARE
    sal            employees.salary%TYPE;
    sal_multiple   employees.salary%TYPE;
    factor          INTEGER := 2;

    CURSOR c1 IS
        SELECT salary,
               salary * factor
        FROM   employees
        WHERE  job_id LIKE 'AD_%';

BEGIN
    OPEN c1;

    LOOP
        FETCH c1
            INTO sal, sal_multiple;
        EXIT WHEN c1%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('factor = ' || factor);
        DBMS_OUTPUT.PUT_LINE('sal      = ' || sal);
        DBMS_OUTPUT.PUT_LINE('sal_multiple = ' || sal_multiple);
        factor := factor + 1;
    END LOOP;

    CLOSE c1;
END;
```



# Параметры в курсоре

DECLARE

```
CURSOR c(job          VARCHAR2,
          max_sal      NUMBER,
          hired        DATE DEFAULT '31-DEC-99') IS
  SELECT last_name,
         first_name,
         (salary - max_sal) overpayment
  FROM   employees
 WHERE  job_id = job
 AND    salary > max_sal
 AND    hire_date > hired
 ORDER BY salary;
```

BEGIN

```
OPEN c('SA_REP', 10000, '31-DEC-04');
do_something;
CLOSE c;
```

END;

# Атрибуты explicit cursor

```
DECLARE
    CURSOR c1 IS
        SELECT last_name,
               salary
        FROM   employees
        WHERE  ROWNUM < 11
        ORDER  BY last_name;

    my_ename  employees.last_name%TYPE;
    my_salary employees.salary%TYPE;
BEGIN
    OPEN c1;
    LOOP
        FETCH c1
            INTO my_ename,
                my_salary;
        IF c1%NOTFOUND
        THEN
            EXIT;
        ELSE
            DBMS_OUTPUT.PUT_LINE('Name = ' || my_ename || ', salary = ' ||
my_salary);
        END IF;
    END LOOP;
END;
```

# Select into

## ✓ Получение одной записи:

```
SELECT select_item [, select_item ]...  
  INTO variable_name [, variable_name ]...  
FROM table_name;
```

## ✓ Получение нескольких записей:

```
SELECT select_item [, select_item ]...  
  BULK COLLECT INTO variable_name [, variable_name ]...  
FROM table_name;
```

## ✓ Исключения:

```
NO_DATA_FOUND  
TOO_MANY_ROWS
```

# Cursor For Loop Statement

```
DECLARE
    CURSOR c1 IS
        SELECT last_name,
               job_id
        FROM   employees
        WHERE  job_id LIKE '%CLERK%'
        AND    manager_id > 120
        ORDER BY last_name;

BEGIN
    FOR item IN (SELECT last_name,
                        job_id
                  FROM   employees
                  WHERE  job_id LIKE '%CLERK%'
                  AND    manager_id > 120
                  ORDER BY last_name)
    LOOP
        DBMS_OUTPUT.PUT_LINE('Name = ' || item.last_name || ', Job = ' ||
item.job_id);
    END LOOP;

    FOR item IN c1
    LOOP
        DBMS_OUTPUT.PUT_LINE('Name = ' || item.last_name || ', Job = ' ||
item.job_id);
    END LOOP;
END;
```

# Cursor variable (REF CURSOR)

- ✓ Может использоваться для выполнения разных select-запросов
- ✓ Может участвовать в выражениях
- ✓ Может быть входным параметром
- ✓ Может быть параметром, передающимся от БД клиенту
- ✓ Не может содержать параметров
- ✓ **Сильный курсор**
- ✓ **Слабый курсор (SYS\_REFCURSOR)**

# Операции с cursor variable

```
DECLARE
    TYPE empcurtyp IS REF CURSOR RETURN employees%ROWTYPE; -- strong type
    TYPE genericcurtyp IS REF CURSOR; -- weak type

    cursor1    empcurtyp; -- strong cursor variable
    cursor2    genericcurtyp; -- weak cursor variable
    my_cursor  SYS_REFCURSOR; -- weak cursor variable

    v_employees employees%ROWTYPE;

    TYPE emplist IS TABLE OF employees%ROWTYPE;
    emp_list emplist;

    v_sql varchar2(200) := 'SELECT * FROM    employees WHERE    REGEXP_LIKE(job_id, ''S[HT]_CLERK'')
ORDER BY last_name';
BEGIN
    OPEN cursor1 FOR v_sql;
    LOOP
        FETCH cursor1
            INTO v_employees;
        EXIT WHEN cursor1%NOTFOUND;
        some_actions;
    END LOOP;
    CLOSE cursor1;

    OPEN cursor1 FOR v_sql;
    FETCH cursor1 BULK COLLECT
        INTO emp_list;
    CLOSE cursor1;
    some_actions_with_emp_list;
END;
```

# Cursor expressions

```
DECLARE
    TYPE emp_cur_typ IS REF CURSOR;

emp_cur    emp_cur_typ;
dept_name  departments.department_name%TYPE;
emp_name    employees.last_name%TYPE;

CURSOR c1 IS
    SELECT department_name,
           CURSOR ( SELECT e.last_name
                     FROM employees e
                     WHERE e.department_id = d.department_id
                     ORDER BY e.last_name
                   ) employees
    FROM departments d
    WHERE department_name LIKE 'A%'
    ORDER BY department_name;

BEGIN
    OPEN c1;
    LOOP -- Process each row of query result set
        FETCH c1 INTO dept_name, emp_cur;
        EXIT WHEN c1%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Department: ' || dept_name);

        LOOP -- Process each row of subquery result set
            FETCH emp_cur INTO emp_name;
            EXIT WHEN emp_cur%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('-- Employee: ' || emp_name);
        END LOOP;
    END LOOP;
    CLOSE c1;
END;
```

# Парсинг курсора

## Память сессии (UGA)

Открытые курсоры  
(open cursors)

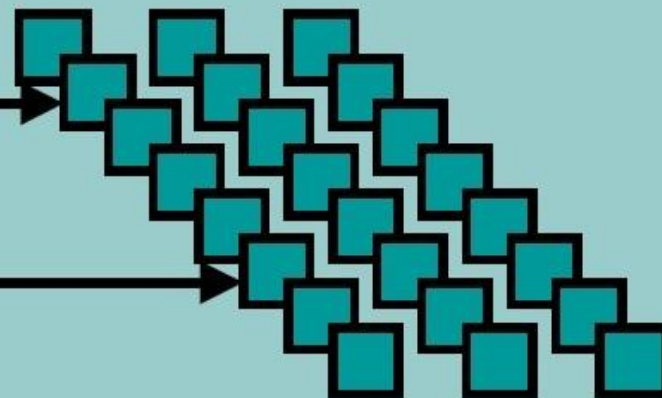


Закрытые курсоры в  
кэше (closed cursors)



## Разделяемый пул (Shared Pool в SGA)

Идентификаторы курсоров  
(cursor handles)





# Параметры БД

✓ `select * from v$parameter where name like '%cursor%'`

✓ `select * from v$system_parameter where name like '%cursor%'`

Параметр	Описание
<b>open_cursors</b>	Максимальное количество открытых курсоров в рамках одной сессии (по умолчанию 50). Максимально возможное значение 65 535.
<b>session_cached_cursors</b>	Максимальное количество session cursors в кэше (по умолчанию 50).
<b>cursor_space_for_time</b>	Параметр устаревший. Пользоваться не следует.

# Словари данных

- ✓ Представление V\$OPEN\_CURSOR возвращает список кэшированных сессией курсоров
- ✓ Чтобы получить количество открытых курсоров, используйте запрос:

```
SELECT a.value,  
       s.username,  
       s.sid,  
       s.serial#  
FROM   v$sesstat a,  
       v$statname b,  
       v$session s  
WHERE  a.statistic# = b.statistic#  
AND    s.sid = a.sid  
AND    b.name = 'opened cursors current'
```

# Summarizing

- ✓ Implicit cursor
- ✓ Explicit cursor
- ✓ Операции с cursor
- ✓ Атрибуты cursor
- ✓ Cursor с параметрами
- ✓ Select into
- ✓ Cursor for loop
- ✓ Ref cursor
- ✓ Cursor expression
- ✓ Парсинг cursor
- ✓ Параметры БД
- ✓ Словари данных

# Список использованных материалов

1. [PL/SQL Static SQL](#) (oracle documentation)
2. [Implicit Cursor Attribute](#) (oracle documentation)
3. [Explicit Cursor Declaration and Definition](#) (oracle documentation)
4. [Open Statement](#) (oracle documentation)
5. [Close Statement](#) (oracle documentation)
6. [Fetch Statement](#) (oracle documentation)
7. [OPEN CURSORS](#) (oracle documentation)
8. [SESSION CACHED CURSORS](#) (oracle documentation)
9. [CURSOR SPACE FOR TIME](#) (oracle documentation)