

3 Взаимодействие с сервером Oracle

Использование SQL в PL/SQL

- Возможность извлечения строки данных при помощи SELECT.
- Внесения изменений - DML команды.
- Управление транзакциями: COMMIT, ROLLBACK, SAVEPOINT.
- Использование неявного курсора для получения информации о результате работы DML.

Оператор SELECT в PL/SQL

- Выборка данных - оператор SELECT:

```
SELECT select_list  
INTO {variable_name[,variable_name]  
      | record_name}  
FROM table  
[WHERE condition];
```

- В PL\SQL для выборки данных фраза INTO обязательна.
- Возможно выбрать только одну запись.
- В случае получения нескольких строк – ошибка:

ORA-01422: exact fetch returns more than requested number of rows

Пример 1 (с ошибкой и без)

```
DECLARE
```

```
    v_deptno NUMBER(4);
```

```
    v_location_id NUMBER(4);
```

```
BEGIN
```

```
    SELECT DEPARTMENT_ID, LOCATION_ID
```

```
    INTO v_deptno, v_location_id
```

```
    FROM DEPARTMENTS
```

```
    -- WHERE DEPARTMENT_NAME = 'Sales';
```

```
    -- ;
```

```
    .....
```

```
END;
```

Пример 2

```
DECLARE
```

```
    v_hire_date employees.hire_date%TYPE;
```

```
    v_salary employees.salary%TYPE;
```

```
BEGIN
```

```
    SELECT HIRE_DATE, SALARY
```

```
    INTO v_hire_date, v_salary
```

```
    FROM EMPLOYEES
```

```
    WHERE EMPLOYEE_ID = 189; -- 111;
```

```
    dbms_output.put_line('DATE: ' || v_hire_date || ' ; Salary: ' || v_salary );
```

```
END;
```

```
-- ORA-01403: no data found
```

Пример 3

```
DECLARE
    v_sum_sal NUMBER(10,2);
    v_deptno NUMBER NOT NULL := 20;
BEGIN
    SELECT SUM(SALARY) --групповая функция
    INTO v_sum_sal
    FROM EMPLOYEES
    WHERE DEPARTMENT_ID = v_deptno;
    dbms_output.put_line(v_sum_sal);
END;
```

DML B PL/SQL

- INSERT
- UPDATE
- DELETE

INSERT

```
BEGIN
INSERT INTO employees
    (employee_id,first_name,last_name,
    email, hire_date, job_id, salary,
    department_id)
    VALUES
    (employees_seq.Nextval,'David','Ruth',
    'd.ruth@gmail.com', sysdate, 'SH_CLERK',
    4000, 30);
END;

-- select * from employees where last_name='Ruth'
-- rollback;
```

UPDATE

Пример:

```
DECLARE
    v_sal_increase emp.sal%TYPE := 800;
BEGIN
    UPDATE employee
    SET salary = salary + v_sal_increase
    WHERE job = 'CLERK';
END;
```

DELETE

Пример:

```
DECLARE
  v_deptno DEPARTMENTS.department_id%TYPE := 800; --220;
BEGIN
  DELETE FROM DEPARTMENTS
  WHERE department_id = v_deptno;
END;

-- нет ошибок если оператор ничего не удалит
-- Проверка:
  select * from DEPARTMENTS WHERE department_id=800;
-- До COMMIT; ROLLBACK; изменения видны - в одной сессии
  работаем SQL Developer-ом во всех закладках текущей бд
  включая все SQL Worksheet's
```

SQL курсор (анализ выполнения DML)

При выполнении команды SQL из PL/SQL РСУБД 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.

Сервер Oracle использует неявные курсоры, чтобы анализировать и выполнять пользовательские инструкции SQL.

PL/SQL предоставляет механизм доступа к этой рабочей области и содержащейся в ней информации.

Операторы DML (INSERT UPDATE DELETE) и SELECT ... INTO – это курсоры. При выполнении "глобальной системной области", (применительно к Oracle) - всегда, открывается курсор. Курсоры имеют ряд атрибутов. Курсор проще всего представить себе как указатель на виртуальную таблицу в базе данных с данными определяемыми SQL командой.

Имеются два типа курсоров:

- Неявные курсоры (неявным называется, поскольку что Oracle автоматически выполняет многие связанные с ним операции, такие как открытие, выборка данных, закрытие.)
- Явные курсоры, явно объявлены программистом (рассмотрим в след уроках)

Атрибуты SQL курсора

PL/SQL предоставляет механизм доступа к курсору - рабочей области памяти и содержащейся в ней информации.

Использование атрибутов курсора SQL. Возможна проверка результатов пользовательских инструкций SQL:

| | |
|---------------------|---|
| SQL%ROWCOUNT | Число строк, возвращенных курсором на данный момент. |
| SQL%FOUND | Булево значение. TRUE в случае выбора одной и более строк. |
| SQL%NOTFOUND | Булево значение. TRUE в случае, если не выбрана ни одна строка. |
| SQL%ISOPEN | Определяет, открыт ли соответствующий курсор. |

Атрибуты SQL курсора

```
DECLARE
```

```
  v_del VARCHAR2(30);  
  v_result VARCHAR2(30) := Null;  
  v_status VARCHAR2(30) := Null;
```

```
BEGIN
```

```
DELETE FROM departments WHERE department_id = 220;  
v_del:=SQL%ROWCOUNT;  
v_result := case SQL%FOUND  
              when TRUE then 'TRUE'  
              else 'FALSE'  
            end;  
v_status := case SQL%ISOPEN  
              when TRUE then 'TRUE'  
              else 'FALSE'  
            end;  
dbms_output.put_line(v_del);  
dbms_output.put_line(v_result);  
dbms_output.put_line(v_status );  
  
DELETE FROM departments WHERE department_id =1220;  
dbms_output.put_line('Строк удалено - '||SQL%ROWCOUNT);  
END;
```

Управление транзакциями

- Для фиксации изменений DML команд.
- Команды COMMIT или ROLLBACK для фиксации транзакции.

ИТОГИ

- Использование SQL команд в PL/SQL блоке
- Команды управления транзакцией
- Курсоры – явные и неявные
- Атрибуты курсора

Практика №3!

40 минут