

Курс «Базы данных»

Тема. Программирование на языке PL/SQL. Часть 5

Барабанщиков
Игорь Витальевич

План лекции

1. Обработка исключений
2. Предопределенные ИС
3. Непредопределенные ИС
4. Пользовательские ИС

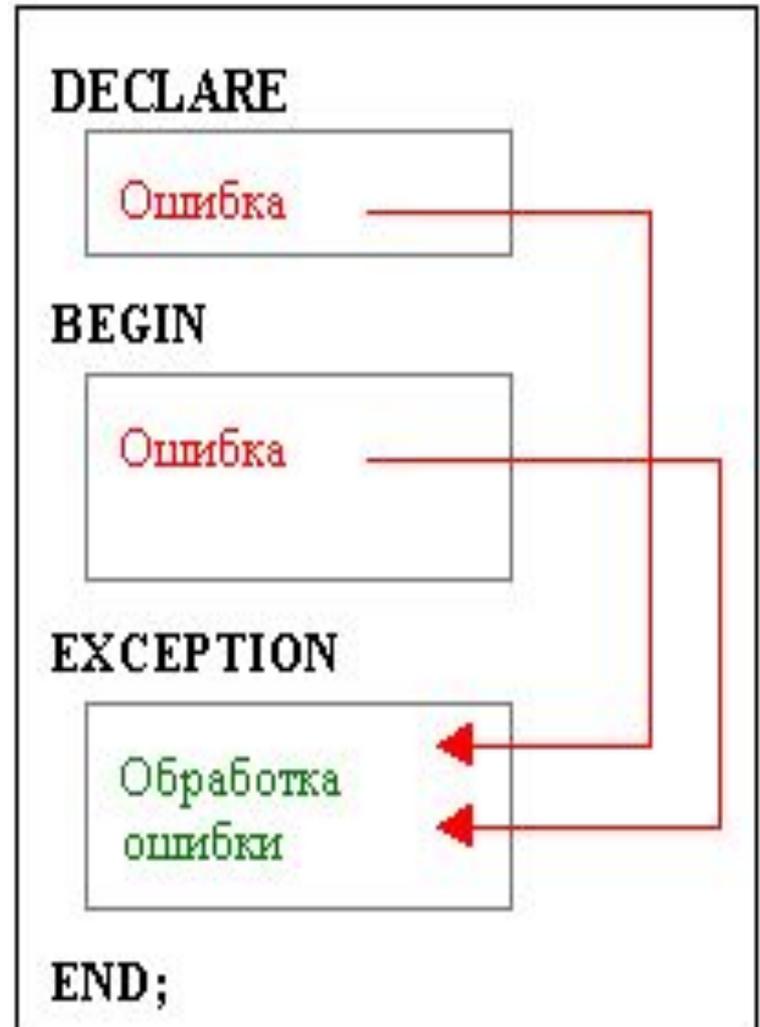
Исключительные ситуации

- При выполнении блока PL/SQL могут возникнуть разные ошибки.
- Эти **ошибки должны быть обработаны.**
- Для обработки ИС в блоке PL/SQL предназначен раздел **EXCEPTION**.



Обработка исключительных ситуаций

- В случае возбуждения исключительной ситуации блок всегда прекращает работу.
- Можно задать обработчик исключения для выполнения заключительных действий.
- Обработчики ИС помещают в раздел **EXCEPTION** блока PL/SQL.



Синтаксис ИС

Каждый обработчик исключения состоит из ключевого слова **WHEN**, задающего исключение, за которым следует последовательность команд, выполняемых в случае, если исключение возбуждено.

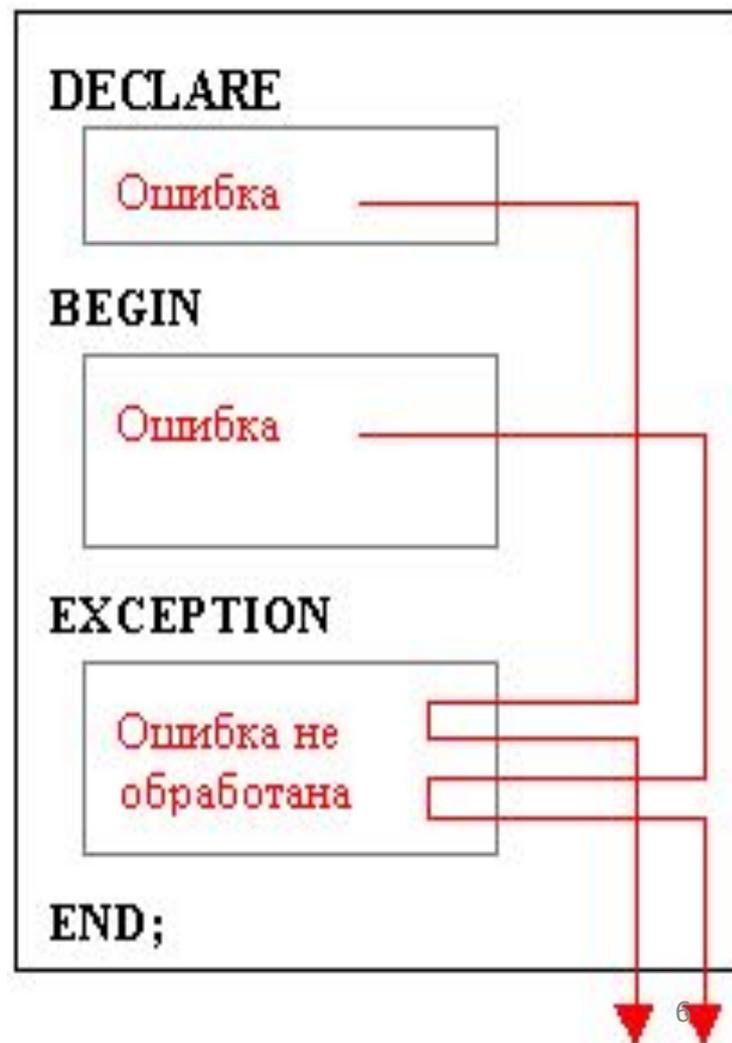
Если возникает ИС, то перед выходом из блока PL/SQL выполняется **только один**

EXCEPTION

```
WHEN исключение1
[OR исключение2 ...]
THEN
    выражение1;
    выражение2;
    ...
[WHEN исключение3
[OR исключение4 ...]
THEN
    выражение1;
    выражение2;
...]
[WHEN OTHERS THEN
    выражение1;
    выражение2;
```

Необработанная в блоке ИС

- Необработанные в блоке ИС приводят к его аварийному завершению.
- В этом случае управление передается в вызывающую среду.



Способы возбуждения ИС

- **Явный** – имеет место **ошибка СУБД Oracle**, исключение возбуждается автоматически. Примеры:
 - деление на ноль,
 - нарушение уникальности.
- **Неявный** – исключение возбуждается **в блоке искусственно с помощью команды RAISE.**

Типы ИС

Исключени е	Описание	Указание по обработке
Предопреде- ленное исключение	Одна из примерно 20- ти типичных ошибок.	Описание не требуется. Явного возбуждения не требуется.
Непредопред е-ленное исключение	Любая стандартная ошибка сервера (невходящая в число предопределенных)	Исключение д. б. объявлено в декларативной части. Явного возбуждения не требуется.
Пользовател ь-ское исключение	Условие, которое разработчик считает ошибочным.	Исключение д. б. объявлено в декларативной части. Требует явного возбуждения.

Предопределенные исключения

Имя исключения	Ошибка Oracle	Описание
NO_DATA_FOUND	ORA-01403	Однострочная команда SELECT не возвращает ни одной строки.
TOO_MANY_ROWS	ORA-01422	Однострочная команда SELECT возвращает много строк.
INVALID_NUMBER	ORA-01722	Неудачная попытка преобразования строки в число.
NOT_LOGGED_ON	ORA-01012	Программа PL/SQL обращается к БД, не подключившись к Oracle.
ZERO_DIVIDE	ORA-01476	Попытка деления на ноль.

Пример предопределенных ИС

```
DECLARE
```

```
  v_id emp.emp_id%TYPE;
```

```
BEGIN
```

```
  SELECT emp_id INTO v_id -- должна возвращать 1 строку  
    FROM emp WHERE ename = 'Иванов';
```

```
  -- другие команды;
```

```
EXCEPTION
```

```
  WHEN no_data_found THEN -- обработчик для 1-й ИС  
    dbms_output.put_line('Сотрудник не найден');
```

```
  WHEN too_many_rows THEN -- обработчик для 2-й ИС  
    dbms_output.put_line('Найдено много сотрудников');
```

```
END;
```

Непредопределенные ИС

Для перехвата непредопределенной ИС, возбуждаемой сервером, необходимо:

1. в декларативной секции блока:

- объявить исключительную ситуацию

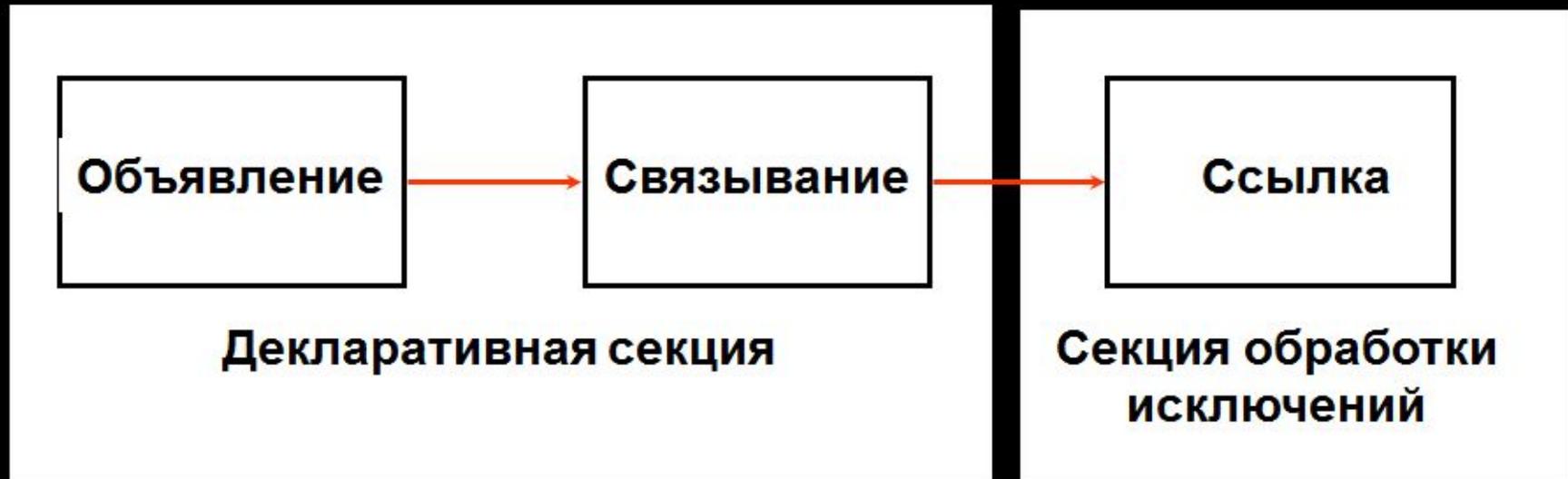
- связать ИС со стандартной ошибкой

Oracle.

2. в секции обработки исключений:

- на ИС можно ссылаться по объявленному имени.

Перехват непредопределенных исключений сервера Oracle



Присвоение имени
исключению

Кодирование указания
компилятору `PRAGMA
EXCEPTION_INIT`

Обработка
исключения

Объявление непредопределенной ИС

- Для связывания ошибки Oracle с именем исключения, используется директива компилятора **PRAGMA EXCEPTION_INIT**.

Синтаксис:

```
DECLARE
    имя_ИС EXCEPTION;
    PRAGMA EXCEPTION_INIT(имя_ИС,
ошибка);
BEGIN
    ...
```

Пример непредопределенной ИС

```
CREATE PROCEDURE del_dept(p_deptno number) IS
  e_emp_exist EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_emp_exist, -2292);
BEGIN
  DELETE FROM dept WHERE deptno=p_deptno;
  COMMIT;
EXCEPTION
  WHEN e_emp_exist THEN
    dbms_output.put_line('Нельзя удалить
отдел!');
END;
```

Функции для перехвата ИС

- **SQLCODE** – возвращает **числовой код** ошибки.
- **SQLERRM** – возвращает **сообщение**, связанное с кодом ошибки

Эти функции используют совместно с обработчиком **OTHERS**.

Обработчик OTHERS д.б. последним в разделе EXCEPTION.

Пример обработчика OTHERS

```
DECLARE
    v_err_code NUMBER;
    v_err_msg  VARCHAR2(200);
BEGIN
    ...
EXCEPTION
    WHEN others THEN
        v_err_code := SQLCODE;
        v_err_msg  := SQLERRM;
        INSERT INTO t_error VALUES(v_err_code, v_err_msg);
END;
```

PL/SQL позволяет определять и возбуждать собственные исключения. Для этого они должны быть объявлены в декларативной секции и возбуждены командой RAISE

Перехват пользовательских исключений



Пример пользовательской ИС

```
DECLARE
    e_error_dept EXCEPTION; -- объявление ИС
BEGIN
    UPDATE dept SET dname='IT' WHERE id=777;
    IF sql%notfound THEN
        RAISE e_error_dept; -- возбуждение ИС
    END IF;
EXCEPTION
    WHEN e_error_dept THEN -- обработка ИС
        dbms_output.put_line('Нет такого отдела!');
END;
```

Процедура RAISE_APPLICATION_ERRO R

- Позволяет посылать собственные пользовательские сообщения об исключениях из хранимых программ.
- Возвращает пользователю информацию об исключении тем же способом, которым пользуется сервер Oracle для возвращения информации для прочих исключений.
- **Вызывается только из хранимых подпрограмм.**

Синтаксис

RAISE_APPLICATION_ERROR

`raise_application_error(ошибка, сообщение, {TRUE | FALSE})`

Ошибка – номер исключения в диапазоне -20000...-20999

Сообщение – текст сообщения об исключении,
заданный

пользователем (до 2048 байт).

TRUE | FALSE – необязательный логический параметр.

TRUE – исключение поместить в стек предыдущих ошибок.

FALSE – исключение должно заместить все предыдущие

Пример

RAISE_APPLICATION_ERROR

```
BEGIN
```

```
...
```

```
DELETE FROM emp
```

```
WHERE empno = v_id;
```

```
IF sql%notfound THEN
```

```
raise_application_error(-20025, 'Не  
найден');
```

```
END IF;
```

```
...
```

Итоги

- PL/SQL мощный язык для программирования серверной логики.
- Он значительно расширяет возможности SQL.

