

Программирование на стороне сервера



Основные операторы PL/pgSQL





- Оператор IF
- Оператор CASE
- Оператор NULL
- Оператор PERFORM



Условный оператор IF

- Для перехода к операторам в разделе **THEN** условие **IF** (**ELSIF**) должно быть истинно
 - в т.ч. <u>не неизвестно</u> (не **UNKNOWN**)
- В случае отсутствия разделов ELSIF и/или ELSE исключение не генерируется



Пример

```
do language plpgsql
$block name$
DECLARE
    v_salary "Production"."Products".unitprice%type ;
    v rang char(1);
begin
    SELECT unitprice INTO STRICT v_salary
    FROM "Production". "Products"
    WHERE productid =1;
    IF v_salary > 2500::money THEN
        v_rang := 'A';
    ELSIF v_salary > 1000::money THEN
        v_rang := 'B';
    ELSE
        v rang := 'C';
    END IF;
    RAISE NOTICE 'Цена %, ранг %', v_salary, v_rang ;
end
$block_name$;
```

```
Цена $18.00, ранг С
```



Оператор CASE

• Аналогичен оператору IF, но генерирует исключение, если ни одно из условий не выполняется и отсутствует раздел ELSE

Синтаксис простого оператора

```
CASE выражение
    WHEN значение_1.1
    [, значение_1.2]... THEN
    инструкции_1;
    [WHEN значение_2.1
    [, значение_2.2] THEN
    инструкции_2;]...
    [ELSE
    инструкции_else;]
END CASE;
```

Синтаксис поискового оператора

```
CASE

WHEN выражение_1 THEN

инструкции_1;

[WHEN выражение_2 THEN

инструкции_2;]...

[ELSE

инструкции_else;]

END CASE;
```

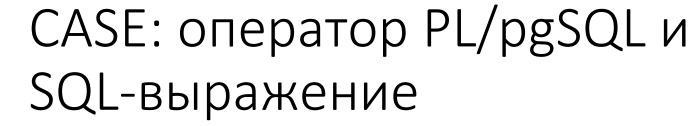


Пример

```
do language plpgsql
$block_name$
DECLARE
   v_salary numeric = 200 ;
   v_rang char(1) := 'A' ;
begin
    CASE v_rang
        WHEN 'A', 'B' THEN
                v salary := v salary + 500;
        WHEN 'C' THEN
                v salary := v salary + 1500;
        ELSE
                v salary := v salary + 1000 ;
    END CASE;
    RAISE notice 'Цена %, ранг %', v_salary, v_rang ;
end
$block_name$;
```

```
Вывод 

Цена 700, ранг А
```





ОПЕРАТОР	ВЫРАЖЕНИЕ
Выполняет действия	Возвращает результат
Каждой паре WHEN-THEN может соответствовать несколько инструкций	Каждая пара WHEN-THEN определяет только <u>одно</u> выражение
В конце – END CASE	В конце - END
Если ни одно из условий не выполняется и отсутствует раздел ELSE — генерирует <u>исключение</u>	Если ни одно из условий не выполняется и отсутствует раздел ELSE — возвращает NULL
CASE rang WHEN 'A' THEN v_salary := v_salary + 500; RAISE NOTICE 'Ok'; ELSE RAISE NOTICE 'Not ok'; END CASE;	<pre>v_salary := CASE v_rang</pre>





AND	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE
NULL	NULL	FALSE	NULL

OR	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL
NULL	TRUE	NULL	NULL

NOT	
TRUE	FALSE
FALSE	TRUE
NULL	NULL



Оператор NULL

- Оператор NULL применяется в местах, где компилятор не должен ничего делать
- B PL/pgSQL этот оператор в коде допустимо опускать

```
CASE v_rang
WHEN 'A' THEN

v_salary := v_salary + 500;
RAISE NOTICE 'Ok';
ELSE

NULL;
END CASE;
```

```
CASE v_rang
WHEN 'A' THEN

v_salary := v_salary + 500;
RAISE NOTICE 'Ok';
ELSE
END CASE;
```



Оператор PERFORM

- Равнозначен SELECT, но без возвращения результата
- После выполнения в специальную переменную **FOUND** логического типа записывается:
 - TRUE если предыдущий запрос PERFORM вернул > 0 строк
 - **FALSE** 0 строк

```
DO $$
BEGIN

PERFORM * FROM "HR"."Employees";

IF FOUND THEN

RAISE NOTICE 'Таблица не пуста!';

ELSE

RAISE NOTICE 'В таблице пока ничего нет...';

END IF;

END $$
```

```
ъ Вывод ⊠
Таблица не пуста!
```