

PGConf.Russia 2023



Поддержка пакетов Oracle (packages) в СУБД Postgres Pro 15 Enterprise

Уменьшение объема работ по
миграции с Oracle

Игорь Мельников
Postgres Professional

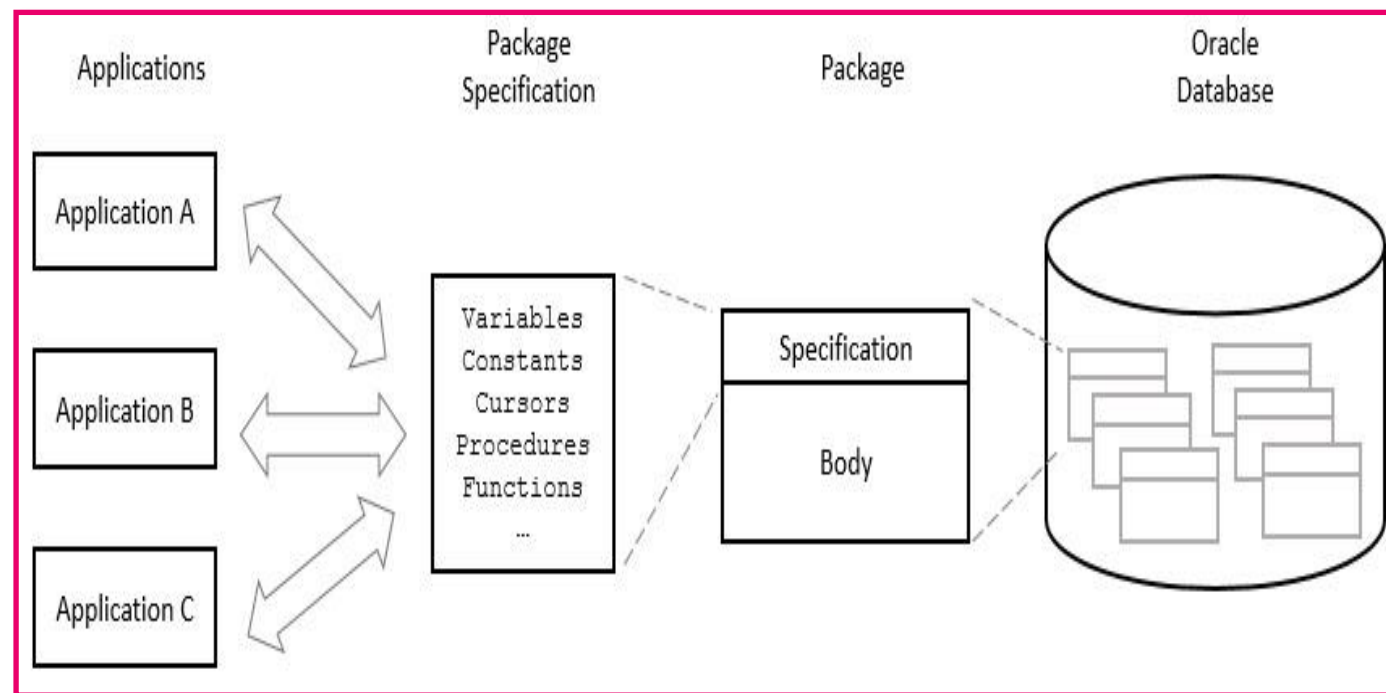
План

- Что такое пакет (package) в СУБД Oracle Database
- Поддержка пакетов в Postgres Pro 15 Enterprise
- Особенности использования пакетов в PL/pgSQL
- Ограничения в первом релизе пакетов PL/pgSQL
- Планы по развитию

Что такое пакет в СУБД Oracle Database

Обеспечивает модульность кода в СУБД Oracle:

- Состоит из спецификации (заголовка) и тела (реализации) пакета
- Элементы, объявленные в спецификации, доступны снаружи
- Элементы, объявленные в теле пакета, доступны только в самом теле пакета



Пример пакета в СУБД Oracle Database

```
CREATE OR REPLACE PACKAGE sample_pkg is
```

```
TYPE THostInfo IS RECORD
```

```
(  
    Name      VARCHAR2(64),  
    Description VARCHAR2(128)  
);
```

```
v_g_host_name VARCHAR2(50);
```

```
FUNCTION get_str_clob(p_tab_n VARCHAR2, p_col_n VARCHAR2) RETURN VARCHAR2;
```

```
END;
```

• Глобальная переменная пакета, доступна в течении “жизни” сессии, у каждой сессии – свой экземпляр!

Пример тела пакета в СУБД Oracle Database

```
CREATE OR REPLACE PACKAGE BODY sample_pkg is
  FUNCTION get_str_clob(p_tab_n VARCHAR2, p_col_n VARCHAR2) return VARCHAR2 IS
    v_xResult  VARCHAR2(32567);
  BEGIN

    ...
    RETURN v_xResult;
  END;

  FUNCTION get_default_host_name return VARCHAR2 IS
    v_xResult  VARCHAR2(32567);
  BEGIN
    ...
    RETURN v_xResult;
  END;

  BEGIN
    v_g_host_name := get_default_host_name();
  END;
```

• Приватная функция:
доступна только в теле
пакета!

• Секция инициализации:
выполняется при первом
обращении к пакету

Использование пакета в СУБД Oracle Database

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> EXEC dbms_output.put_line(sample_pkg.v_g_host_name);
```

```
localhost
```

```
PL/SQL procedure successfully completed
```


Как перенести пакет в PostgreSQL?

Для хранения переменных пакета используется расширение

- Реализует секцию инициализации “пакета”. Каждый раз нужно проверять флаг инициализации пакета

```
CREATE PROCEDURE sample_pkg__init_package AS $$  
BEGIN  
  IF NOT pgv_exists('sample_pkg') THEN --выполнить блок инициализации  
    CALL pgv_set('sample_pkg','init_package', true);  
    CALL pgv_set('sample_pkg','v_g_host_name', get_default_host_name());  
  END;  
END; $$
```

- Обертка для получения значения пакетной переменной.

```
CREATE FUNCTION sample_pkg__get_v_g_host_name RETURNS VARCHAR AS $$  
BEGIN  
  CALL sample_pkg__init_package();  
  
  RETURN pgv_get('sample_pkg','v_g_host_name',NULL::varchar);  
END; $$
```


Как перенести пакет в PostgreSQL - продолжение

Для хранения переменных пакета используется расширение *pg_variables* - продолжение

• Обертка для установки значения пакетной переменной.

```
CREATE PROCEDURE sample_pkg__set_v_g_host_name (p_value VARCHAR) AS $
BEGIN
    CALL sample_pkg__init_package();

    pgv_set('sample_pkg','v_g_host_name', p_value);
END; $
```

```
CREATE FUNCTION sample_pkg__get_str_clob( p_tab_n VARCHAR, p_col_n
                                         VARCHAR) RETURNS VARCHAR AS

    v_xResult  VARCHAR(32567);
BEGIN
    CALL sample_pkg__init_package();

    ...
    RETURN v_xResult;
END; $
```

Миграция пакетов из Oracle в PostgreSQL

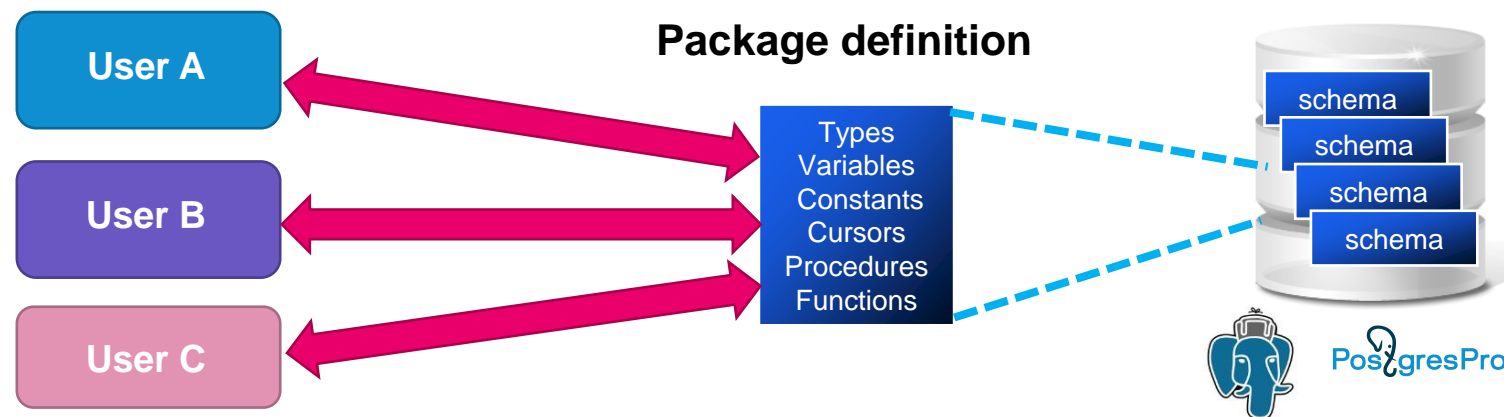
**Необходим большой
объем работ по редизайну
кода пакетов!**

Простой конвертацией синтаксических конструкций НЕ обойтись!

Поддержка пакетов в Postgres Pro 15 Enterprise

Поддержка пакетов PL/pgSQL в Postgres Pro 15 Enterprise:

- Пакет является отдельной схемой
- Поддерживаются функции инициализации – функция с фиксированным именем “***__init__***” – вызывается в момент первого обращения к пакету
- Поддерживаются глобальные переменные – все переменные объявленные в процедуре инициализации “***__init__***”
- Поддерживаются обращение к элементам других пакетов (схем), в том числе к глобальным переменным!



Пример пакета в СУБД Postgres Pro 15 Ent

```
CREATE OR REPLACE PACKAGE sample_pkg
```

```
CREATE TYPE THostInfo AS  
(  
    Name    VARCHAR(32),  
    Description VARCHAR(256)  
)
```

```
CREATE FUNCTION __init__() RETURNS VOID AS $$  
DECLARE  
    v_g_host_name VARCHAR(50);  
BEGIN  
    v_g_host_name := get_default_host_name();  
END;  
$$
```

- Функция инициализации пакета, все объявленные в ней переменные являются глобальными переменными пакета

Пример пакета PL/pgSQL - продолжение

```
CREATE FUNCTION get_str_clob(p_tab_n VARCHAR, p_col_n VARCHAR) RETURNS VARCHAR
AS $$
DECLARE
    v_xResult VARCHAR(32567);
BEGIN
    ... ...
    RETURN v_xResult || v_g_host_name;
END;
$$
```

- Прозрачно используется глобальная переменная пакета

```
CREATE FUNCTION get_default_host_name() RETURNS VARCHAR AS $$
BEGIN
    ... ...
    RETURN v_g_host_name;
;END;
$$
;
```


Использование пакета в Postgres Pro 15 Ent

```
DO $$  
#import sample_pkg  
BEGIN  
  RAISE NOTICE 'Hostname => %', sample_pkg.v_g_host_name;  
END;  
$$;
```

• Прагма для импорта пакета

```
NOTICE: Hostname => localhost  
DO
```

```
SELECT sample_pkg.get_default_host_name();  
get_default_host_name
```

• Доступ к глобальной переменной пакета

```
-----  
localhost  
(1 строка)
```

• Функция стала публичной!

Использование пакета в другой функции PL/pgSQL

```
CREATE FUNCTION qux.demo() RETURNS INT AS $$
```

```
#import foo, bar
```

```
DECLARE
```

```
  z INT := foo.x + bar.y;
```

```
  Y NUMERIC;
```

```
BEGIN
```

```
  IF z > (foo.x + bar.y) THEN
```

```
    return 1;
```

```
  END IF;
```

```
  RETURN 0;
```

```
END $$ LANGUAGE PLPGSQL;
```

• Прагма для импорта пакетов

• Доступ к глобальным переменным пакетов

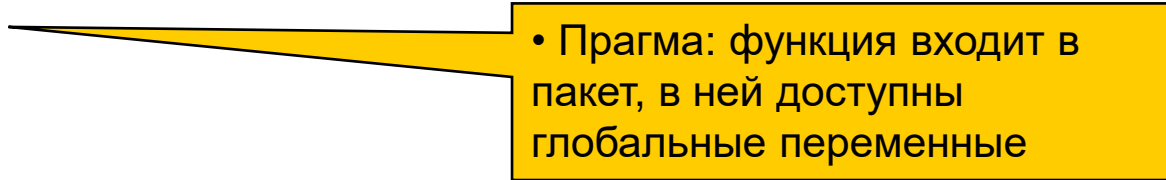


Особенности использования пакетов в Postgres Pro Enterprise

Изменение пакета по частям: добавление процедуры в пакет, изменение типа в пакете

В отличие от Oracle, где пакет может меняться только целиком

```
CREATE OR REPLACE PROCEDURE sample_pkg.new_proc(p_value VARCHAR) AS $$  
#package  
BEGIN  
...  
END;  
$$ LANGUAGE PLPGSQL;
```



- Прагма: функция входит в пакет, в ней доступны глобальные переменные

```
ALTER TYPE sample_pkg.THostInfo ADD ATTRIBUTE val_int INT;
```

Зависимости между пакетами: нет инвалидации как в Oracle, зависимые объекты нужно удалить

```
CREATE OR REPLACE PACKAGE sample_child_pkg
```

```
CREATE FUNCTION get_database_host_info() RETURNS sample_pkg.THostInfo AS $$  
DECLARE  
    v_xResult sample_pkg.THostInfo;  
BEGIN  
    RETURN v_xResult;  
END;  
$$  
;
```

- Новый пакет зависит от типа в другом пакете

```
CREATE OR REPLACE PACKAGE sample_pkg  
CREATE TYPE THostInfo AS  
(  
    Name    VARCHAR(32)  
);
```

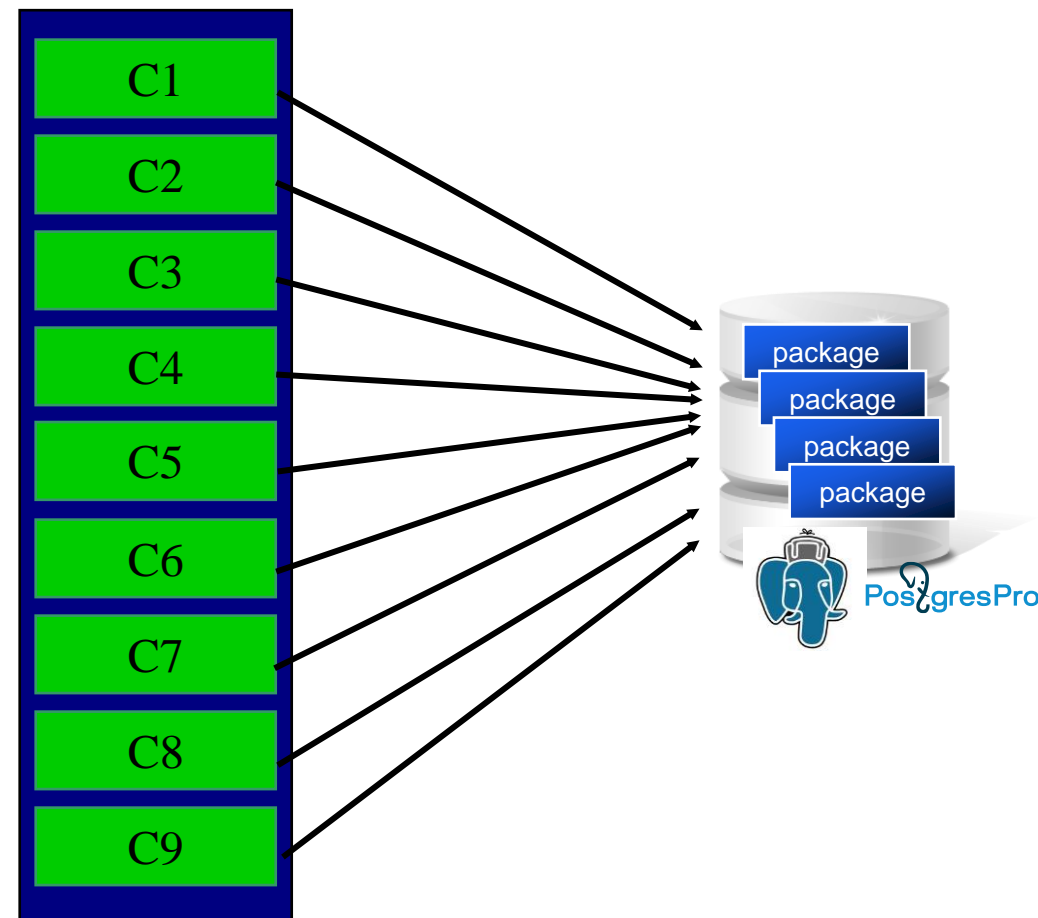
ERROR: cannot change existing type

- Деструктивное изменение типа в родительском пакете – из типа убран атрибут

Сброс состояния всех пакетов в сессии

- Встроенная процедура **plpgsql_reset_packages()**
- В сессии, все пакеты сбрасываются в НЕинициализированное состояние
- Необходимо, чтобы вернуть чистую сессию обратно в пул соединений
- Все глобальные переменные пакетов получают начальные значения
- Аналог процедуры **dbms_session.reset_package** в СУБД Oracle

Connection Pool



Ограничения пакетов в Postgres Pro Enterprise

Ограничения в пакетах PL/pgSQL

1

Пакет не имеет спецификации (публичной части) и тела (приватной части)

2

Не поддерживаются прагмы компилятора Oracle PL/SQL, например: *UDF*, *INLINE* и т.д.

3

Не поддерживаются модификаторы объявления подпрограмм пакета, например: *RESULT CACHE*, *ACCESSIBLE BY*

4

Отсутствие исключения “Package state discard” и блокировки на изменение пакета, который в данный момент выполняется

Планы по развитию: Пакеты 2.0

Что появится нового

Пакеты 2.0 – дальнейшее развитие

- Несколько пакетов в одной схеме
- Поддержка приватности элементов пакета
-

Утилита ORA2PGPRO:

- Форк от ORA2PG, с дополнительными новыми возможностями
- Транслятор кода пакетов Oracle в синтаксис Postgres Pro 15 Ent
- Генерация скриптов выдачи привилегий на пакет (grant)
- Конвертация синтаксиса автономных транзакций Oracle PL/SQL в синтаксис Postgres Pro
- И т.д.



Добавление системных пакетов Oracle

- Новое расширение PGPRO_UTL
- Реализация системных пакетов из Oracle, которые отсутствуют в ORAFCE:
 - UTL_MAIL
 - UTL_SMTP
 - UTL_HTTP
 - DBMS_LOB
 - DBMS_APPLICATION_INFO
- В дальнейшем – реализация обработки очередей в БД (Advanced Queuing)

Пакеты в Postgres Pro 15 Enterprise

**Уменьшение объема работ,
при миграции из Oracle в
Postgres Pro Enterprise!**

Поддержка основной функциональности пакетов: группировка в пакет, глобальные переменные и секция инициализации

Q & A

Спасибо!