

5

Создание пакетов

Рассматриваемые вопросы

- **Описание пакетов и перечень их возможных компонентов**
- **Создание пакетов для группирования взаимосвязанных переменных, курсоров, констант, исключений, процедур и функций**
- **Общедоступные и частные конструкции пакета**
- **Обращение к конструкциям пакета**
- **Описание пакета, не содержащего тела**



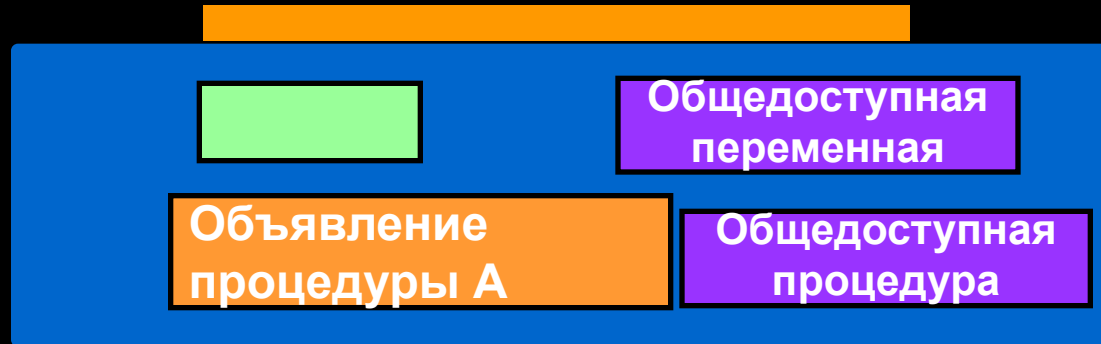
Обзор пакетов

- Объединяют логически взаимосвязанные типы PL/SQL, элементы и подпрограммы
- Состоят из двух компонентов:
 - Спецификация
 - Тело
- Не могут вызываться, принимать параметры и быть вложенными
- Позволяют Oracle одновременно считывать в память несколько объектов

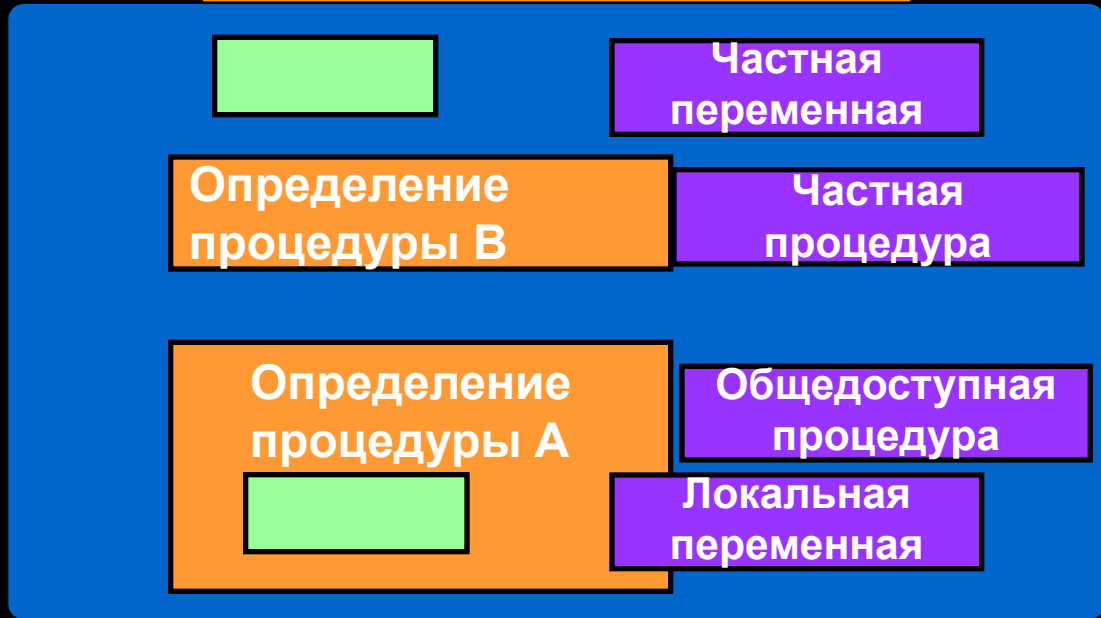


Компоненты пакета

Спецификация
пакета



Тело
пакета



Ссылки на объекты пакета

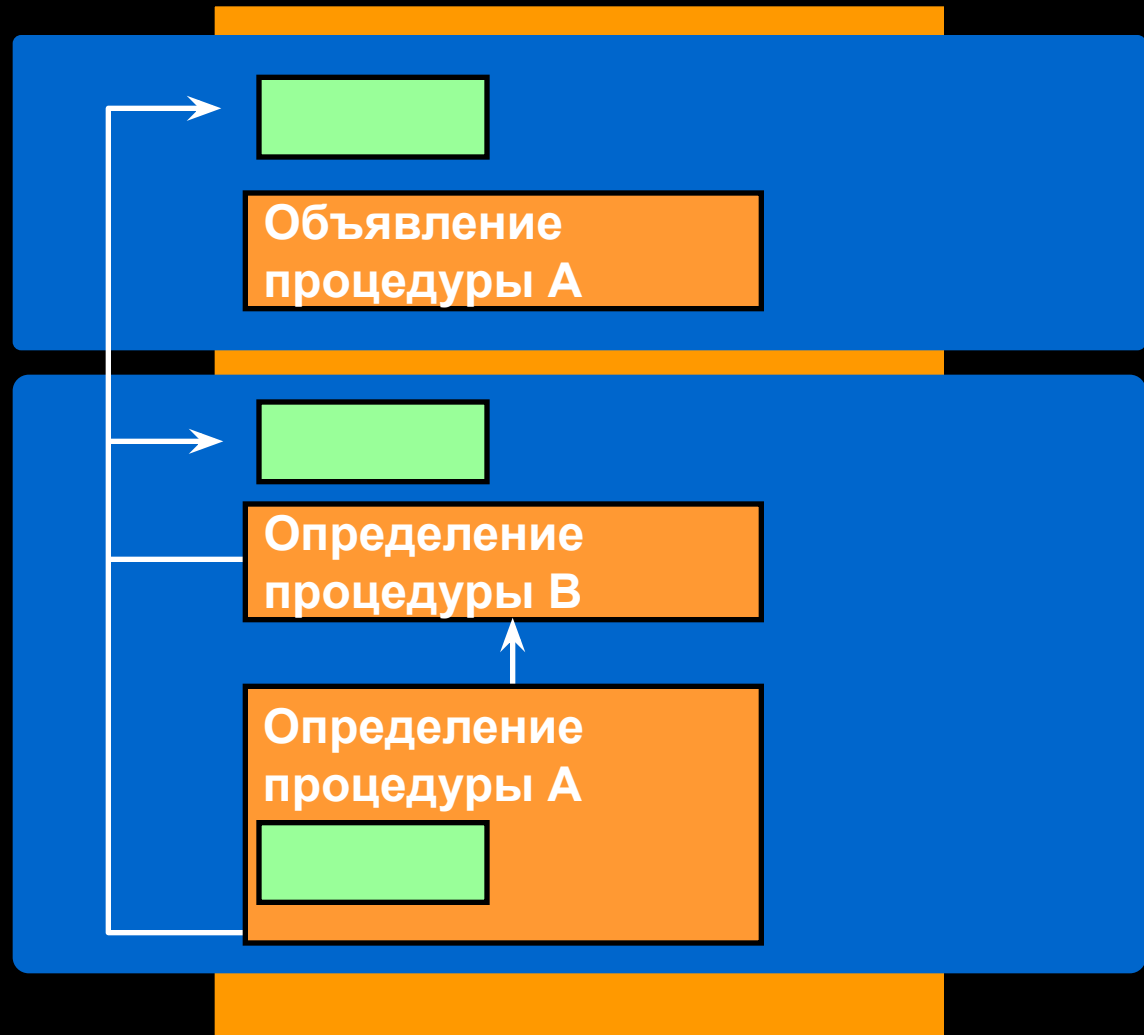
Спецификация
пакета

Объявление
процедуры А

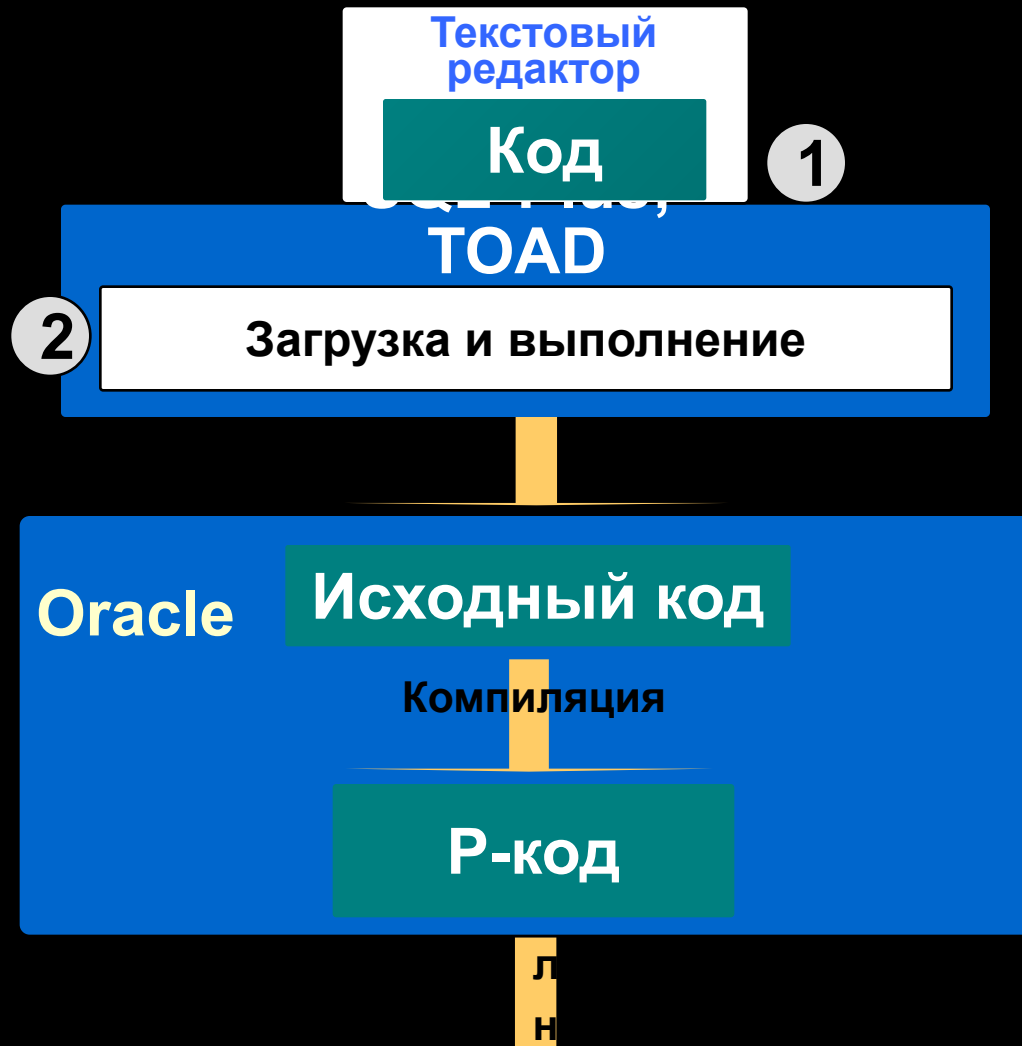
Тело
пакета

Определение
процедуры В

Определение
процедуры А



Разработка пакета



Разработка пакета

- **Сохранение текста команды CREATE PACKAGE в двух отдельных командных файлах SQL облегчает последующее изменение пакета.**
- **Спецификация пакета может существовать без тела пакета, но не наоборот.**



Создание спецификации пакета

Синтаксис:

```
CREATE [OR REPLACE] PACKAGE имя_пакета  
IS | AS  
    объявления общедоступных типов и элементов  
    спецификации подпрограмм  
END имя_пакета;
```

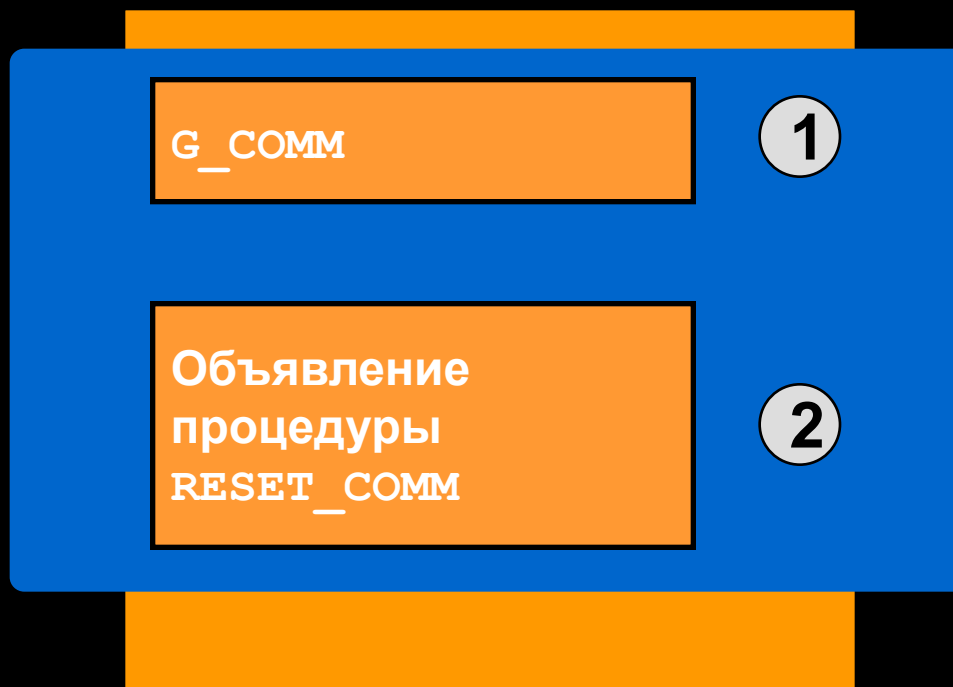
- Задание параметра **REPLACE** приводит к удалению и повторному созданию спецификации пакета.
- По умолчанию переменные, объявленные в спецификации пакета, инициализируются как неопределенные (**NULL**).
- Все конструкции, объявленные в спецификации пакета, видимы для других пользователей, которым предоставлена привилегия на этот пакет.



Объявление общедоступных конструкций

Пакет COMM_PACKAGE

Спецификация
пакета



Создание спецификации пакета: пример

```
CREATE OR REPLACE PACKAGE comm_package IS
  g_comm NUMBER := 0.10;  --initialized to
  0.10
  PROCEDURE reset_comm
    (p_comm IN NUMBER);
END comm_package;
/
```

Package created.

- **G_COMM** – глобальная переменная с начальным значением 0.10.
- **RESET_COMM** – общедоступная процедура, реализуемая в теле пакета.



Создание тела пакета

Синтаксис:

```
CREATE [OR REPLACE] PACKAGE BODY имя_пакета  
IS | AS  
    объявления частных типов и элементов  
    тела подпрограмм  
END имя_пакета;
```

- Задание параметра `REPLACE` приводит к удалению и повторному созданию тела пакета.
- Идентификаторы, объявленные только в теле пакета, являются частными конструкциями. Они невидимы вне тела пакета.
- Все частные конструкции должны быть объявлены перед (до) их использованием в общедоступных конструкциях.



Общедоступные и частные конструкции

Пакет COMM_PACKAGE

Спецификация
пакета

G_COMM

1

Объявление процедуры
RESET_COMM

2

Тело
пакета

Определение функции
VALIDATE_COMM

3

Определение процедуры
RESET_COMM

2



Создание тела пакета: пример

```
CREATE OR REPLACE PACKAGE BODY comm_package
IS
    FUNCTION  validate_comm (p_comm IN NUMBER)
        RETURN BOOLEAN
    IS
        v_max_comm      NUMBER;
    BEGIN
        SELECT      MAX(commission_pct)
            INTO      v_max_comm
            FROM      employees;
        IF    p_comm > v_max_comm THEN RETURN(FALSE);
        ELSE    RETURN(TRUE);
        END IF;
    END validate_comm;
    . . .
```



Создание тела пакета: пример

```
PROCEDURE  reset_comm (p_comm  IN  NUMBER)
IS
BEGIN
  IF  validate_comm(p_comm)
  THEN  g_comm:=p_comm;  --reset global variable
  ELSE
    RAISE_APPLICATION_ERROR(-20210,'Invalid commission');
  END IF;
END reset_comm;
END comm_package;
/
```

Package body created.

Вызов пакетных конструкций

Пример 1: вызов функции из процедуры в том же пакете.

```
CREATE OR REPLACE PACKAGE BODY comm_package IS
    .
    .
    .
    PROCEDURE reset_comm
      (p_comm IN NUMBER)
    IS
    BEGIN
      IF validate_comm(p_comm)
      THEN g_comm := p_comm;
      ELSE
        RAISE_APPLICATION_ERROR
          (-20210, 'Invalid commission');
      END IF;
    END reset_comm;
END comm_package;
```



Вызов пакетных конструкций

Пример 2: вызов пакетной процедуры

```
comm_package.reset_comm(0.15)
```

Пример 3: вызов пакетной процедуры из другой схемы.

```
scott.comm_package.reset_comm(0.15)
```

Пример 4: вызов пакетной процедуры в удаленной базе данных.

```
comm_package.reset_comm@ny_link(0.15)
```



Объявление спецификации без тела пакета

```
CREATE OR REPLACE PACKAGE global_consts IS
  mile_2_kilo      CONSTANT  NUMBER  :=  1.6093;
  kilo_2_mile     CONSTANT  NUMBER  :=  0.6214;
  yard_2_meter    CONSTANT  NUMBER  :=  0.9144;
  meter_2_yard    CONSTANT  NUMBER  :=  1.0936;
END global_consts;
/

EXECUTE DBMS_OUTPUT.PUT_LINE('20 miles = '||20*
  global_consts.mile_2_kilo||' km')
```

```
Package created.
20 miles = 32.186 km
PL/SQL procedure successfully completed.
```



Ссылка на общедоступную переменную из автономной процедуры

Пример:

```
CREATE OR REPLACE PROCEDURE meter_to_yard
(p_meter IN NUMBER, p_yard OUT NUMBER)
IS
BEGIN
  p_yard := p_meter * global_consts.meter_2_yard;
END meter_to_yard;
/
```

```
DECLALRE
  yard NUMBER(8,6);
BEGIN
  meter_to_yard (1, yard);
  dbms_out_put.put_line(yard);
END;
```

```
Procedure created.
PL/SQL procedure successfully completed.
```

YARD
1.0936

Удаление пакетов

Удаление спецификации и тела пакета:

```
DROP PACKAGE имя_пакета;
```

Удаление тела пакета :

```
DROP PACKAGE BODY имя_пакета;
```



Разработка пакетов: указания

- Пакеты должны быть как можно более общими.
- Прежде, чем определить тело пакета, следует определить его спецификацию.
- Спецификация пакета должна содержать только общедоступные конструкции.
- Располагайте элементы в декларативной части тела пакета, если они должны сохранять значение в течение сеанса или в разных транзакциях.
- После изменения спецификации пакета требуется повторная компиляция ссылающихся на него подпрограмм.
- Спецификация пакета должна включать как можно меньше конструкций.



Преимущества пакетов

- **Модульность: инкапсуляция связанных конструкций.**
- **Упрощение проектирования приложений: кодирование и компиляция спецификации и тела выполняется отдельно.**
- **Соккрытие информации:**
 - **Только объявления, сделанные в спецификации пакеты видимы и доступны для приложений.**
 - **Частные конструкции в теле пакета скрыты и недоступны.**
 - **Весь код скрыт в теле пакета.**



Преимущества пакетов

- **Дополнительная функциональность: устойчивое состояние переменных и курсоров**
- **Повышение производительности:**
 - При первом обращении к компоненте пакета весь пакет загружается в память
 - Все пользователи используют только одну копию пакета, размещенную в памяти
 - Упрощается иерархическая зависимость между компонентами приложения
- **Перезагрузка: несколько подпрограмм с одним и тем же именем**



Итоги

Использование пакетов:

- Улучшает организацию, управление и защиту. Повышает производительность.
- Позволяет группировать взаимосвязанные процедуры и функции.
- Позволяет модифицировать тело пакета, не затрагивая спецификацию.
- Позволяет предоставить право доступа ко всему пакету.



Итоги

- **Исходный код скрыт от пользователей**
- **При первом вызове в память загружается весь пакет**
- **Уменьшается количество обращений к диску при выполнении последующих вызовов**
- **Предоставляются переменные, используемые в течение сеанса пользователя**



Итоги

Команда	Описание
<code>CREATE [OR REPLACE] PACKAGE</code>	Создание (или изменение) спецификации существующего пакета
<code>CREATE [OR REPLACE] PACKAGE BODY</code>	Создание (или изменение) тела существующего пакета
<code>DROP PACKAGE</code>	Удаление как спецификации, так и тела пакета
<code>DROP PACKAGE BODY</code>	Удаление только тела пакета

Обзор практического занятия 5

- **Создание пакетов**
- **Вызов программных единиц из пакета**