

## Программные единицы в составе БД Oracle - хранимые процедуры, функции и пакеты.

Хранимая процедура (stored procedure) — это программа, которая вы-полняет некоторые действия с информацией в базе данных и при этом сама хранится в базе данных. В Oracle хранимые процедуры можно мож-но писать на языках PL/SQL и Java.

Хранимые процедуры могут входные параметры и возвращать резуль-таты. В отличие от триггеров, которые принадлежат определенной табли-це или представлению, хранимые процедуры принадлежат базе данных в целом. Они могут вызываться любым процессом, использующим базу дан-ных, при условии, что у этого процесса есть достаточные права доступа.

Хранимые процедуры используются для многих целей. Хотя админи-страторы баз данных используют их для выполнения рутинных задач ад-министрирования, главной областью их применения являются все же при-ложения баз данных. Эти процедуры могут вызываться из прикладных программ, написанных на таких языках, как Java, C#, C++ или VB.Net, а также из веб-сценариев, написанных на VBScript или JavaScript. Кроме того, эти процедуры можно вызывать в интерактивном режиме из команд-ной оболочки SQL\*Plus.

Можно выделить следующие преимущества хранимых процедур:

В отличие от кода приложений, хранимые процедуры никогда не пере-даются на клиентские компьютеры. Она всегда находятся в базе данных и выполняются СУБД на том компьютере, где располагается сервер базы данных. Таким образом, они более безопасны, чем распространяемый код приложения, а кроме того, снижают сетевой трафик. Хранимые процеду-ры постепенно становятся предпочтительным режимом реализации логи-ки приложения в сети Интернет и корпоративных интрасетях. Еще одно преимущество хранимых процедур заключается в том, что SQL-операторы в них могут быть оптимизированы компилятором СУБД.

```
CREATE OR REPLACE PROCEDURE TESTPRM(NUM IN NUMBER)
```

```
IS
```

```
in_COMP VARCHAR2(50);
```

```
BEGIN
```

```
    SELECT COMPANY INTO in_COMP FROM customers
```

```
    WHERE customers.CUST_NUM = NUM;
```

```
    DBMS_OUTPUT.enable;
```

```
    DBMS_OUTPUT.put_line(in_COMP);
```

```
END TESTPRM;
```

```
/
```

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
BEGIN
```

```
TESTPRM(2112);
```

```
END;
```

```
/
```

Функция – это подпрограмма, которая вычисляет значение.

```
CREATE OR REPLACE Function FindCourse
```

```
( name_in IN varchar2 )
```

```
RETURN number
```

```
IS
```

```
number number;
```

```
cursor c1 is
```

```
SELECT course_number
```

```
FROM courses_tbl
```

```
WHERE course_name = name_in;
```

```
BEGIN
```

```
open c1;
```

```
fetch c1 into cnumber;
```

```
if c1%notfound then
```

```
    cnumber := 9999;
```

```
end if;
```

```
close c1;
```

```
RETURN cnumber;
```

```
EXCEPTION
WHEN OTHERS THEN
  raise_application_error(-20001,'An error was encountered - '||SQLCODE||' -ERROR-
' ||SQLERRM);
END;
```

```
CREATE OR REPLACE PACKAGE имя_модуля {IS AS}
  описание_процедуры |
  описание_функции |
  объявление_переменной |
  определение_типа |
  объявление_исключительной_ситуации |
  объявление_курсора |
END [имя_модуля];
Тело модуля определяется так:
```

```
CREATE OR REPLACE PACKAGE BODY имя_модуля {IS AS}
  код_инициализации_процедуры |
  код_инициализации_функции |
END [имя_модуля];
```

По большому счету тело модуля не является обязательной частью. Если заголовок модуля содержит описание, скажем нескольких переменных, типов и курсоров, то создавать тело модуля нет необходимости. Такой способ целесообразен при объявлении глобальных переменных, поскольку все объекты модуля видимы вне его пределов.

процедуры  
функции  
типы  
курсоры

### Синонимы в БД Oracle

Синоним (Synonym) – это альтернативное имя (псевдоним) для объекта схемы. Если для какого либо объекта базы данных Oracle существует синоним, то к объекту из SQL запроса можно обращаться либо по его настоящему имени, либо по синониму. Так же они обеспечивают некоторый уровень безопасности, поскольку скрывают имя объекта и его владельца, а так же делают прозрачным местоположение удаленных объектов распределенных баз данных. Синонимы позволяют переименовывать и перемещать базовые объекты. При том переопределяется только синоним, а приложение не требует никаких модификаций.

Различают два типа синонимов:

Частный (PRIVATE)- синонимы содержатся в схеме конкретного пользователя и доступны только самому пользователю, и тем, кому он предоставил соответствующие права доступа.

Общий (PUBLIC)- этими синонимами владеет специальная группа пользователей – PUBLIC, в результате чего эти синонимы доступны всем пользователям базы данных.

```
CREATE SYNONYM [имя_синонима] FOR[имя_объекта]
CREATE PUBLIC SYNONYM [имя_синонима] FOR[имя_объекта]
```

### Словарь данных - назначение, основные представления

Oracle поддерживает исчерпывающий словарь метаданных. Этот словарь описывает структуру таблиц, последовательностей, представлений, индексов, ограничений, хранимых процедур и многое другое. Он также содержит исходные тексты процедур, функций и триггеров.

В таблице DICT словаря метаданных содержатся данные, описывающие сам словарь. Можно запрашивать данные из этой таблицы, чтобы узнать больше о содержимом словаря данных, но нужно иметь в виду, что она имеет большие размеры. Например, если запросить имена всех таблиц словаря данных, будет возвращено более 800 строк.

Предположим, нужно узнать, какие таблицы с информацией о пользовательских и системных таблицах имеются в словаре данных. В этом может помочь следующий запрос:

```
SELECT Table_Name, Contents
FROM DICT
WHERE Table_Name LIKE (' %TABLES%');
```

Будет возвращено около двадцати пяти строк. Одна из таблиц будет называться USER\_TABLES. Чтобы увидеть столбцы этой таблицы, нужно ввести:

```
DESC USER_TABLES;
```

Можно использовать эту стратегию для получения из словаря мета-данных информации об интересных объектах и структурах. В табл. 4.1 перечислены многие из представлений и указано их назначение. Таблицы USER\_SOURCE и USER\_TRIGGERS полезны, когда требуется узнать, исходные тексты каких процедур и триггеров хранятся в настоящий момент в базе данных.

#### Таблица 4.1. Метаданные в СУБД Oracle

Имя таблицы	Содержимое
-------------	------------

DICT	Метаданные, описывающие словарь данных
------	--

USER_CATALOG	Список таблиц, представлений, последовательно-стей и других структур, принадлежащих пользователю
--------------	--

USER_TABLES	Структуры таблиц пользователя
-------------	-------------------------------

USER_TAB_COLUMNS	Потомок таблицы USER TABLES. Содержит данные о столбцах таблиц. Синонимом является COLS
------------------	---

USER_VIEW	Пользовательские представления
-----------	--------------------------------

USER_CONSTRAINTS	Пользовательские ограничения
------------------	------------------------------

USER_CONS_COLUMNS	Потомок таблицы USER_CONSTRAINTS. Содержит столбцы, на которые наложены ограничения
-------------------	---

USER_TRIGGERS	Метаданные, описывающие триггеры. Есть смысл запрашивать столбцы Trigger Name, Trigger Type и Trigger Event. Предупреждение: Trigger Body в действительности не содержит исходного кода триггера
---------------	--

USER_SOURCE	Исходные тексты. Например, для получения текста процедуры MYTRIGGER: SELECT Text FROM USER_SOURCE WHERE Name = 'MYTRIGGER' AND Type = 'PROCEDURE'
-------------	---

**Dynamic performance views в словаре данных Oracle - назначение и основные представления**

**V\$LOCK**

Locks currently held/requested on the instance

**V\$LATCHHOLDER**

Sessions/processes holding a latch

**V\$OPEN\_CURSOR**

Cursors opened by sessions on the instance

**V\$SESSION**

Sessions currently connected to the instance

**V\$SESSION\_WAIT**

Different resources sessions are currently waiting for