12. Процедуры и функции

12.1 Что такое процедуры и функции

Процедура или функция - это объект схемы, который логически объединяет набор команд SQL и других программных конструкций PL/SQL для выполнения определенной задачи. Процедуры и функции создаются в пользовательской схеме и хранятся в базе данных для последующего использования. Процедуры и функции выполняются интерактивно, используя средства ORACLE.

ORACLE обрабатывает как процедуры и функции, так и отдельные команды SQL. Процедура или функция - это объект схемы, которая содержит последовательность команд SQL и других конструкций PL/SQL, сгруппированных вместе. Процедуры и функции мало чем отличаются друг от друга, разница состоит в том, что функция всегда возвращает код возврата тому, кто ее вызывает, а процедура этого не делает. Кроме того, в функциях не разрешены аргументы типа INOUT и OUT. В тоже время процедуры, в отличие от функций, не могут использоваться в операторах SQL типа SELECT, INSERT, UPDATE. Предложение типа:

```
SELECT procedure(param1,param2) FROM table;
```

недопустимо, в отличии от предложения

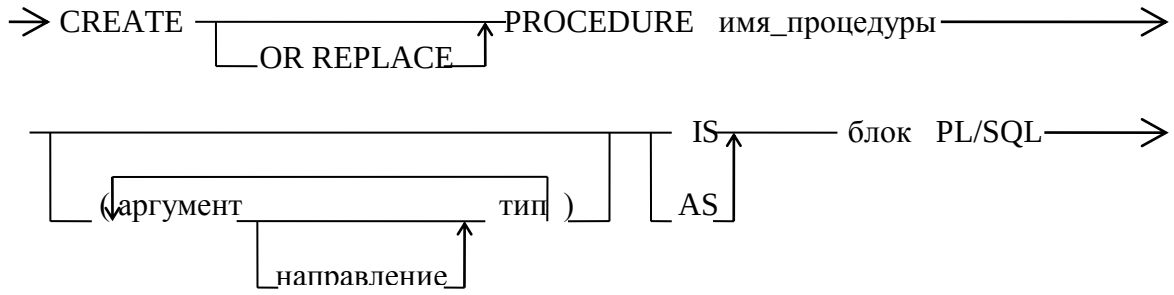
```
SELECT function(param1,param2) FROM table;
```

которое возвращает вычисленное функцией значение для каждой строки, возвращаемой запросом.

12.2 Принципы работы с процедурами и функциями

Создание и использование хранимых процедур должно удовлетворять следующим требованиям:

- процедура и функция обязательно должны быть созданы по следующему образцу:
- процедура должна выполнять отдельную законченную задачу. Она не должна быть длинной, объединяющей функции различного назначения;
- процедуры не должны дублировать возможности, встроенные в ORACLE.



12.3 Применение процедур

12.3.3 Защита данных

Процедуры могут помочь обеспечить защиту данных. Можно ограничить операции пользователей с базой данных, предоставляя им доступ к данным только через процедуры и функции.

Например, можно предоставить пользователям доступ к процедуре, которая обновляет таблицу, вместо того, чтобы выделять им привилегии для доступа к самой таблице. Поскольку пользователи имеют только привилегию на выполнение этой процедуры, а не

сами объектные привилегии на выборку, обновление или удаление данных из базовых таблиц, они не могут произвольно манипулировать данными иначе, чем это заложено в процедуре.

12.3.4 Производительность

Хранимые процедуры могут улучшить производительность базы данных, поскольку использование процедур существенно уменьшает объем информации, которая пересылается по сети по сравнению с выполнением отдельных команд SQL или пересылкой текста полного блока PL/SQL на сервер. Более того, поскольку процедура хранится в полностью откомпилированном виде в базе данных, то не требуется дополнительных шагов для компиляции, а также, если процедура уже существует в разделяемом буфере SGA, не требуется обращаться к диску - можно выполнить эту процедуру немедленно.

12.3.5 Выделение памяти

Поскольку хранение процедуры использует разделяемые области памяти, в оперативную память загружается только одна копия процедуры, которая может выполняться несколькими пользователями. Разделение кода процедуры между несколькими пользователями приводит к значительному уменьшению памяти, требуемой для приложения, и, следовательно, снижает требования к оперативной памяти сервера.

12.3.6 Продуктивность

Процедуры повышают продуктивность разработки. Создавая приложение, как общий набор процедур, можно исключить дублирующее программирование и повысить продуктивность. Также повышается структурированность программного обеспечения и облегчает его сопровождение.

12.3. 7 Целостность

Хранимые процедуры улучшают целостность и непротиворечивость приложений. При создании приложения, использующего общий набор процедур, можно уменьшить время на отладку и поиск ошибок в этих процедурах и функциях.

Например, процедуру или функцию нужно отладить только один раз для того, чтобы гарантировать, что она работает правильно и возвращает корректный результат. Сделав это один раз, можно затем использовать эту процедуру в любых приложения с гарантией ее правильной работы. Если данные, к которым обращается процедура, изменятся каким-нибудь образом, необходимо только перекомпилировать только одну процедуру, и приложения, которые ее используют, будут работать правильно.

12.4 Сравнение безымянных блоков PL/SQL с хранимыми процедурами

Безымянный блок PL/SQL создается и посылается на сервер, как блок PL/SQL без имени. ORACLE компилирует этот блок и помещает его откомпилированную версию в разделяемый буфер SGA. Однако, он не хранит исходный текст этого блока или его откомпилированную версию в базе данных для последующего использования. Разделяемая область SQL позволяет использовать откомпилированный безымянный блок PL/SQL повторно и одновременно несколькими пользователями, пока он хранится в буфере SGA.

Хранимая процедура альтернативно создается и хранится в базе данных, как отдельный объект. Будучи однажды создан и откомпилирован, этот поименованный объект может затем выполняться без повторной компиляции. Кроме этого, в словарь данных помещается информация о зависимостях между объектами базы данных, которая гарантирует правильность каждой хранимой в базе процедуры.

Итак, заменяя блоки PL/SQL на хранимые процедуры базы данных, можно избежать ненужных повторных компиляций процедур, улучшая тем самым общую производительность приложения.

12.5 Хранение процедур в базе данных ORACLE

При создании процедуры, ORACLE автоматически выполняет следующие шаги:

- компилирует процедуру;
- записывает в базу данных откомпилированный код;
- хранит процедуру в базе данных.

Компилятор PL/SQL компилирует исходный код. Этот компилятор является частью машины PL/SQL, встроенной в сервер ORACLE. Если в результате компиляции возникает ошибка, появляется сообщение.

ORACLE помещает откомпилированную процедуру в разделяемый буфер SGA. Это позволяет коду процедуры выполняться быстро и разделяться между многими пользователями. Откомпилированные версии процедур остаются в разделяемом буфере оперативной памяти в соответствии с алгоритмом LRU, даже если пользователь, который первым вызвал эту процедуру, заканчивает свой сеанс работы.

12.6 Хранение процедур в базе данных

Во время создания и компиляции, ORACLE автоматически записывает следующую информацию о процедуре базы данных;

Имя объекта	Сообщение об ошибках ORACLE использует это имя для того, чтобы идентифицировать процедуру. Это имя задается в команде CREATE PROCEDURE или CREATE FUNCTION
Исходный код и дерево разбора	Компилятор PL/SQL разбирает исходный код и строит представление исходного кода, называемое «деревом разбора»
Псевдокод (P-код)	Компилятор PL/SQL строит псевдокод, основываясь на разобранном коде. Машина PL/SQL выполняет этот код, когда вызывается процедура ORACLE может выдавать ошибки во время компиляции процедуры

Чтобы избежать ненужной перекомпиляции процедуры, дерево разбора и P-код объекта записываются в базу данных. Это позволяет машине PL/SQL прочитать откомпилированную версию процедуры в разделяемый буфер SGA, если ее разобранное представление отсутствует в SGA. Дерево разбора используется после компиляции кода вызова процедуры. Все части процедуры базы данных хранятся в табличной памяти SYSTEM (словаре данных) соответствующей базы данных.

12.7 Выполнение процедур

12.7.1 Проверка привилегий пользователя

ORACLE проверяет, является ли пользователь, вызывающий процедуру, ее владельцем или имеет ли привилегию EXECUTE для этой процедуры. От пользователя, который выполняет процедуру, не требуется, чтобы у него был доступ к другим процедурам или объектам, к которым обращается процедура. Только владелец процедуры должен иметь привилегии для доступа к объектам схемы, на которые ссылается вызываемая процедура.

12.7.2 Проверка правильности процедуры

ORACLE проверяет, используя данные словаря данных, статус процедуры. Процедура считается *недействительной*, если после ее последней компиляции возникло одно из следующих событий:

- хотя бы один из объектов (таблица, представление или другая процедура), к которым обращается процедура, были изменены или уничтожены.;
- у владельца процедуры была отобрана соответствующая системная привилегия;
- у владельца объекта, или у пользователя PUBLIC была отобрана объектная привилегия на доступ хотя бы к одному объекту, на который ссылается эта процедура.

Процедура считается *правильной*, если она не была признана неправильной при возникновении одного из вышеприведенных событий.

Когда вызывается правильная процедура, выполняется ее откомпилированный код. Если вызывается недействительная процедура, она автоматически компилируется перед выполнением и выполняется, если не изменились условия ее функционирования, в противном случае ORACLE выдает перечень ошибок, препятствующих правильной компиляции.

12.7.3 Выполнение процедур

Среда выполнения PL/SQL выполняет процедуру, используя различные шаги, зависящие от следующих условий:

- если процедура действительна и данный момент времени находится в оперативной памяти, машина PL/SQL просто выполняет ее P-код;
- если процедура действительна, но в настоящий момент времени не находится в оперативной памяти, машина PL/SQL загружает ее P-код с диска в оперативную память, а затем выполняет его.

12.7.4 Создание хранимой процедуры

Хранимая процедура создается при помощи команды CREATE PROCEDURE.
Например, команда на создание хранимой процедуры, которая принимала бы информацию о номере сотрудника и производила бы его удаление, может выглядеть так:

```
CREATE PROCEDURE remove_emp (employee_id NUMBER) AS
tot_emps NUMBER;
BEGIN
DELETE FROM employees
WHERE employees.employee_id = remove_emp.employee_id;
tot_emps := tot_emps - 1;
END;
/
```

Вызвать эту хранимую процедуру можно, например, следующим образом:

```
begin
hr.remove_emp(221);
end;
```

Лабораторная работа 12.1 Создание функции

Задание:

Создайте функцию в схеме hr в базе данных Oracle со следующими параметрами:

- функция должна называться fSalary;
- она должна принимать в качестве входящего параметра номер сотрудника;

- она должна возвращать размер заработной платы для данного сотрудника на основе информации из таблицы hr.employees.

Напишите код PL/SQL, в котором бы производился вызов этой функции и в окно приложения выводилась бы информация, возвращаемая этой функцией.

Лабораторная работа 12.2 Создание хранимой процедуры

Задание:

Напишите хранимую процедуру, которая бы:

- принимала в качестве параметров номер сотрудников и размер новой заработной платы для этого сотрудника;
- изменяла бы размер заработной платы для сотрудника, заменяя его на новый (указанный вами в качестве входящего параметра для хранимой процедуры);

Сохраните эту хранимую процедуру в схеме hr под именем hr.pSalary.

Напишите код, который бы менял при помощи этой хранимой процедуры зарплату для сотрудника с номером 100. Новая зарплата должна составлять 25000.