

13.9 Триггеры

13.9.1 Что такое триггеры

ORACLE позволяет определять процедуры, которые неявно вызываются, когда над связанными с ними таблицами выполняются команды INSERT, UPDATE и DELETE. Эти процедуры называются *триггерами базы данных*.

Триггеры похожи на хранимые процедуры. Триггер может содержать команды SQL и PL/SQL и выполняется как отдельная программная единица; он также может вызывать другие хранимые процедуры. Однако процедуры и триггеры отличаются способом их вызова. Процедуры вызываются явно пользователем или приложением, а триггер выполняется неявно, когда ORACLE выполняет команду INSERT, UPDATE или DELETE для таблицы, на которую этот триггер настроен, вне зависимости от того, кто эту команду выполняет.

Триггеры хранятся в базе данных отдельно от таблиц, с которыми они связаны. Триггеры могут определяться только для таблиц, но не для представлений. Однако триггеры выполняются для базовых таблиц представлений при их обновлении командами DML.

13.9.2 Использование триггеров

Как правило, триггеры применяются для автоматической поддержки сложных правил целостности и дополнительной защиты базы данных.

Стандартные возможности защиты данных - роли и привилегии, задают, какие именно команды языка DML может выполнять конкретный пользователь над конкретной таблицей. В дополнение к этому, триггер может ограничить операции DML, например, в определенные дни недели.

Кроме этого триггеры часто используются в следующих случаях:

- автоматическая генерация выводимых значений столбцов;
- защита от неправильных транзакций;
- обеспечение сложных проверок прав доступа;
- обеспечение ссылочной целостности в распределенной базе данных;
- задание сложных правил целостности;
- фиксирование и обработка различных событий в прозрачном для пользователя режиме;
- обеспечение сложного протоколирования;
- обеспечение синхронной репликации таблиц;
- сбор статистики о доступе к таблицам.

13.9.3 Компоненты триггера

Событие или команда триггера вызывают его выполнение. Таким событием могут быть команды DML, выполняющиеся для конкретной таблицы. При задании нескольких типов команд DML для одного триггера нужно использовать дополнительные условия, которые будут идентифицировать конкретный тип команды. Таким образом, можно создать один триггер, который будет выполнять различные команды в зависимости от типа команды, которая вызвала этот триггер.

```
create or replace trigger arh_reviz  
AFTER DELETE OR UPDATE ON reviz
```

```

.....
IF UPDATING THEN
    INSERT                                INTO                                nal_arhu
VALUES (:old.npor, :old.datzap, :old.kod, :old.dokum,
        :old.srok, :old.summa, :old.ispol, :old.kol_rab,
        :old.day_pr, :old.type, to_char(sysdate, 'YYMMDD'), USER);
IF DELETING THEN
    INSERT                                INTO                                nal_arhd
VALUES (:old.npor, :old.datzap, :old.kod, :old.dokum,
        :old.srok, :old.summa, :old.ispol, :old.kol_rab,
        :old.day_pr, :old.type, to_char(sysdate, 'YYMMDD'), USER);
END IF;

```

После изменения таблицы REVIZ или удаления из нее строк, все старые значения заносятся в таблицу изменений и удалений.

Триггеры используются для отслеживания несанкционированного изменения и удаления данных из таблицы, где корректировки допустимы, но необходимо их отслеживать, во избежание потери важных данных по халатности или с личным умыслом. Для этого используются триггеры типа аудит. Данный триггер записывает все старые данные в архивную таблицу и еще он пишет дату изменения и имя пользователя, который произвел эти изменения.

13.9.4 Ограничения или условия выполнения триггера

Условие триггера задается булевым выражением, которое должно быть истинно, чтобы сработал этот триггер. Если это условие принимает значение «ложь», действия триггера не выполняются.

Условие триггера - это необязательная возможность, которая может быть включена в триггер при его создании. Эта возможность задается только для триггера, который выполняется для каждой строки таблицы, чтобы ограничить применение триггера только к отдельным строкам таблицы. Условие триггера задается с помощью фразы WHEN.

```
when (:old.summa>:new.summa)
```

Процедура выполняется только в том случае, если начисления штрафных санкций уменьшаются.

13.9.5 Действие триггера

Действие триггера - это процедура или блок PL/SQL, который содержит команды SQL и PL/SQL, выполняющиеся всякий раз при возникновении событий триггера и выполнении условия триггера.

По аналогии с хранимой процедурой, действие триггера может содержать команды SQL и PL/SQL. Триггер может определять языковые конструкции PL/SQL - переменные, константы, курсоры, исключения и т.д., а также может вызывать хранимые процедуры. Кроме этого, для триггеров строк команды действия триггера могут обращаться к новым и старым значениям столбцов текущей строки. Для доступа к новым и старым значениям столбца существуют специальные имена корреляции.

```
when (:old.summa>:new.summa)
```

В вызове условия хранимой процедуры используются новые и старые значения поля summa .

13.9.6 Типы триггеров

При задании триггера можно определить, сколько раз его выполнять: для каждой обрабатываемой строки (это может происходить, когда команда, на которую заряжен триггер, обновляет много строк), либо всего один раз для целой команды, на которую заряжен триггер, вне зависимости от того, сколько строк (пусть даже ни одной) она обрабатывает.

13.9.7 Триггеры строки

Триггеры строки срабатывают каждый раз, когда команда триггера действует на таблицу. Например, если команда UPDATE действует на несколько строк таблицы, триггер строки выполняется для каждой строки, которую обрабатывает эта команда. Если команда триггера не обрабатывает ни одной строки, триггер не выполняется вообще. Триггеры строки полезны, если действие UPDATE зависит от данных, которые содержатся в команде триггера или данных строк, на которые эти команды действуют.

13.9.8 Триггеры команды

Триггеры команды срабатывают только один раз для всей команды триггера независимо от того, сколько строк таблицы она обрабатывает. Например, если команда DELETE удаляет несколько строк таблицы, триггер команды срабатывает только один раз вне зависимости от того сколько строк удаляется из таблицы.

Триггеры команды полезны, если действия триггеров не зависят от данных, содержащихся в команде триггера или строк, на которые она действует. Например, если триггер производит проверку на текущее время или на конкретного пользователя, или если триггер строит запись в журнальной таблице, основываясь на типе команды триггера.

13.9.9 Сравнение триггеров BEFORE и AFTER

Задавая триггер можно определить момент срабатывания триггера, т.е. можно задать, будет ли действие триггера выполняться до команды триггера или после нее. Триггеры BEFORE и AFTER применяются как к триггерам команды , так и к триггерам строки.

13.9.10 Триггеры BEFORE

Триггеры BEFORE выполняют действия триггеров перед тем, как выполнить саму команду, вызвавшую триггер. Этот тип триггеров чаще всего используется в следующих ситуациях:

когда действие триггера должно определить, стоит ли выполняться команде триггера. Используя для этой цели триггер BEFORE, можно избежать ненужной обработки команды триггера и возможного отката в случае ненужности ее выполнения;

для построения выводимых значений столбцов перед тем, как выполнить над этими столбцами команду триггера INSERT или UPDATE.

13.9.11 Триггеры AFTER

Триггеры AFTER выполняются после того, как выполняется команда триггера. Триггеры AFTER используются в следующих случаях:

- когда необходимо, чтобы команда триггера завершилась полностью перед тем, как выполнять действие триггера;
- если существует триггер BEFORE, триггер AFTER может также присутствовать и выполнять другие действия для той же команды триггера.

Используя параметры, описанные выше, можно создать четыре типа триггеров:

- триггер BEFORE на уровне команды

Перед тем, как выполнять команду триггера, будет выполняться действие триггера;

- триггер BEFORE на уровне строки

Перед тем, как изменять каждую строку, которую обрабатывает команда триггера и перед тем, как проверять соответствующее правило целостности, будет проверяться действие триггера, если условие триггера для этой строки принимает значение «истина»;

- триггер AFTER на уровне команды

После выполнения команды триггера и применения соответствующих объявленных правил целостности, выполняется действие триггера;

- триггер AFTER на уровне строки

После изменения каждой строки и, возможно, применения правил целостности, выполняется действие триггера для текущей строки, если условие триггера принимает значение «истина». В отличие от триггеров BEFORE, триггеры AFTER на уровне строки оставляют строки заблокированными.

13.9.13 Триггеры INSTEAD OF

Триггеры INSTEAD OF выполняются вместо той команды, на выполнение которой они реагируют. Например, представим, что у нас есть такое представление:

```
CREATE VIEW order_info AS
SELECT c.customer_id, c.cust_last_name, c.cust_first_name,
o.order_id, o.order_date, o.order_status
FROM customers c, orders o
WHERE c.customer_id = o.customer_id;
```

Чтобы пользователи могли производить через него вставку, можно создать следующий триггер INSTEAD OF:

```
CREATE OR REPLACE TRIGGER order_info_insert
INSTEAD OF INSERT ON order_info
DECLARE
duplicate_info EXCEPTION;
PRAGMA EXCEPTION_INIT (duplicate_info, -00001);
BEGIN
INSERT INTO customers
(customer_id, cust_last_name, cust_first_name)
VALUES (
:new.customer_id,
:new.cust_last_name,
:new.cust_first_name);
INSERT INTO orders (order_id, order_date, customer_id)
VALUES (
:new.order_id,
:new.order_date,
:new.customer_id);
EXCEPTION
WHEN duplicate_info THEN
RAISE_APPLICATION_ERROR (
num=> -20107,
msg=> 'Duplicate customer or order ID');
END order_info_insert;
```

Теперь можно обращаться к данному представлению, выполняя вставку данных при помощи обычной команды INSERT:

```
INSERT INTO order_info VALUES
(999, 'Smith', 'John', 2500, '13-MAR-2001', 0);
```

Конечно же, на самом деле вместо этой команды отработает триггер.

13.9.13 Выполнение триггера

Триггер может существовать в одном из двух различных состояний:

Включенный	Включенный триггер выполняется, когда выполняется команда триггера и ограничение триггера принимает значение «истина»
Выключенный	Выключенный триггер при выполнении команд триггера не

	выполняется
--	-------------

Для включенных триггеров ORACLE автоматически делает следующее:

- выполняет триггеры в соответствующей последовательности, когда от одной команды SQL срабатывает несколько триггеров;
- выполняет проверку правил целостности в зависимости от типов существующих триггеров и обеспечивает соответствие работы триггеров и объявленных правил целостности;
- обеспечивает целостность чтения для запросов и правил целостности;
- управляет зависимостями между триггерами и объектами, к которым обращаются действия триггеров.

13.9.13 Модель выполнения триггеров и проверка правил

Одна команда SQL может вызывать до четырех триггеров: триггер BEFORE на уровне строки, триггер BEFORE на уровне команды, триггер AFTER на уровне строки, триггер AFTER на уровне команды. Команды триггера или действия внутри самого триггера могут вызывать проверку правил целостности, кроме того, триггер может содержать команды, которые вызывают другие триггеры, так называемые, каскадные триггеры. ORACLE использует следующую модель выполнения, чтобы обеспечить выполнение нескольких триггеров и проверку правил целостности в правильном порядке.

1. Выполняется триггер BEFORE на уровне команды.
2. Выполняется цикл для каждой строки, на которую действует команда SQL.
3. Выполняется триггер BEFORE для строки.
4. Строка блокируется и изменяется, затем для нее выполняется проверка правил целостности. Эта блокировка не освобождается до тех пор, пока не завершится вся транзакция.
5. Выполняется триггер AFTER для строки.
6. Завершаются все проверки правил целостности.
7. Выполняется триггер AFTER на уровне команды.

Определение этой модели рекурсивно. Например, данная команда SQL может вызвать триггер BEFORE для строки и проверку правил целостности. Этот BEFORE триггер может выполнить обновления, которые вызовут проверку правил целостности и выполнение триггера AFTER на уровне команды. Этот триггер AFTER вызывает также проверку правил целостности. В этом случае эти шаги выполняются рекурсивно следующим образом:

1. Выдается оригинальная команда SQL.
2. Выполняется триггер BEFORE на уровне строки.
3. От команды UPDATE внутри триггера BEFORE для строки выполняется триггер AFTER на уровне команды.
4. Выполняются действия триггера AFTER на уровне команды.
5. Выполняется проверка целостности для таблиц, измененных триггером AFTER на уровне команды.
6. Выполняется команда триггера BEFORE для строки.
7. Выполняются проверки правил целостности для таблиц, измененных триггером BEFORE на уровне строки.
8. Выполняется оригинальная команда SQL.

9. Выполняется проверка правил целостности для таблиц, измененной командой SQL.

Важным свойством этой модели выполнения является то, что все действия и проверки, производимые в результате оригинальной команды SQL, должны выполняться успешно. Если произойдет ошибка внутри триггера, и эта ошибка не будет явно обработана в триггере, все действия, выполненные в результате этой оригинальной команды триггера, включая все действия, выполненные ранее другими триггерами, будут отменены. Таким образом, правила целостности не нарушаются с помощью триггеров. Эта модель выполнения принимает во внимание правила целостности и отменяет триггеры, которые могут нарушить декларативные правила целостности.

13.9.14 Доступ к данным для триггеров

Когда выполняется триггер, таблицы, к которым обращаются, могут иметь текущие изменения, сделанные командами SQL из других пользовательских транзакций. В этом случае команды SQL, выполняемые внутри триггеров, следуют общему правилу, заданному для отдельных команд SQL. В частности, если незавершенная транзакция изменила значение, которое триггер должен либо прочитать, либо изменить, команда SQL в теле триггера использует следующие правила:

запросы видят целостную на уровне чтения моментальную копию таблицы, к которой они обращаются, а также все изменения, сделанные внутри той же самой транзакции;

обновления ожидают, пока данные, которые им нужны, будут разблокированы.

13.9.15 Хранение триггеров

В отличие от хранимых процедур, в базе данных хранится только исходный код триггера. Во время первого выполнения исходный код компилируется и записывается в разделяемый буфер оперативной памяти. Если триггер будет удален из этого разделяемого буфера, исходный текст триггера должен быть откомпилирован заново при его следующем вызове.

ORACLE выполняет триггеры, используя те же самые шаги для выполнения процедуры. Одно небольшое отличие состоит в том, что привилегией пользователя на запуск триггера считается наличие у пользователя привилегий для выполнения команды, от которой срабатывает триггер. Во всем остальном триггеры проверяются и выполняются таким же образом, как и хранимые процедуры.

ORACLE автоматически управляет зависимостью триггеров от других объектов схемы, на которые ссылается действие триггера. Однако, такие зависимости управляются только тогда, когда триггеры находятся в разделяемом буфере, они не поддерживаются на том же уровне, что и зависимости для хранимых процедур.

Лабораторная работа 13.4 Создание триггеров

Задание:

Создайте для таблицы HR.EMPLOYEES триггер MySalaryCheck, который бы при добавлении новых записей (и внесении изменений в существующие) проверял бы новое значение в столбце SALARY. При этом:

- не должен допускаться ввод значений зарплаты менее 1000 и более 50000;
- при попытке нарушить эти условия клиентскому приложению должно передаваться сообщение 'Зарплата ниже допустимого минимума' и информация об ошибке -20300 или сообщение 'Зарплата выше допустимого максимума' и сообщение об ошибке 20301 соответственно.