

Раздел 5. Процедурные расширения языка SQL

Тема 5.1

Программируемые объекты в SQL

Вопросы лекции:

1. **Хранимые процедуры.**
2. Триггеры и их разновидности.

Хранимые процедуры

-это программы на SQL, хранящиеся в БД (на сервере)

Преимущества выполнения хранимых процедур

1. Может многократно выполнять действия, задаваемые пользователем.
2. Представляют собой один из видов их общих ресурсов.

Преимущества выполнения хранимых процедур

3. хранимые процедуры поддерживают модульное программирование, так как позволяют разбивать большие задачи на самостоятельные, более мелкие и удобные в управлении части;

Преимущества выполнения хранимых процедур

4. хранимые процедуры могут вызывать другие хранимые процедуры и функции;

5. хранимые процедуры могут быть вызваны из прикладных программ других ТИПОВ;

Преимущества выполнения хранимых процедур

6. хранимые процедуры выполняются быстрее, чем последовательность отдельных команд;

7. хранимые процедуры проще использовать;

Хранимые процедуры вызываются

- клиентской программой,
- другой хранимой процедурой
- триггером.

Разработчик может **управлять правами доступа** к хранимой процедуре, разрешая или запрещая ее выполнение.

Типы хранимых процедур (ХП)

1. Системные ХП:

- для выполнения административных действий;
- имеют префикс **sp_**,
- хранятся в системной БД
- могут быть вызваны в контексте любой другой БД

Типы ХП

2. Пользовательские ХП

- реализуют необходимые действия;
- располагается в конкретной БД, где и выполняется

Типы ХП

3. **Временные** хранимые процедуры существуют лишь какое-то время, после чего **автоматически** **уничтожаются** сервером.

Они делятся на ***локальные и глобальные.***

Типы ХП

3.1. **Локальные временные хранимые** процедуры могут быть вызваны **ТОЛЬКО ИЗ ТОГО соединения** (клиентского сеанса связи с БД), **в котором созданы.**

Имя начинается с символа #.

Типы ХП

Как и все временные объекты, **локальные временные** хранимые процедуры автоматически удаляются:

1. при отключении сеанса пользователя,
2. при перезапуске или остановке сервера БД.

Типы ХП

3.2. **Глобальные временные хранимые процедуры** доступны **для любых соединений сервера, на котором имеется эта процедура.**

Имя начинается с символов **##**.

Удаляются эти процедуры при перезапуске или остановке сервера, а также при закрытии соединения, в контексте которого они были созданы.

Создание ХП

CREATE PROC[EDURE]

procedure_name [;number]

[{@parameter data type}

[VARYING] [=default] [**OUTPUT**]] [...n]

[WITH {RECOMPILE|ENCRYPTION }]

[FOR REPLICATION]

AS sqlstatement [...n]

Вызов ХП

команда **EXES [UTE]**

из процедур, функций и других типов программных объектов.

При вызове процедуры **можно не указывать значения** тех параметров, для которых заданы значения **по умолчанию**.

EXEC [UTE] имя_процедуры [;номер]
[[@имя_параметра=]
{значение | @имя_переменной}
[OUTPUT]|[DEFAULT]][,...n]

Раздел 5. Процедурные расширения языка SQL

Тема 5.1

Программируемые объекты в SQL

Вопросы лекции:

1. Хранимые процедуры.
2. **Триггеры и их разновидности.**

Триггер — это специальный вид хранимой процедуры.

Триггеры обеспечивают проверку любых изменений на корректность, прежде чем эти изменения будут приняты.

Свойства :

1 . Каждый триггер привязывается к конкретной таблице.

2 . Все производимые им модификации данных рассматриваются как одна транзакция.

3. В случае обнаружения **ошибки** или нарушения целостности данных происходит **откат транзакции**.

Тем самым внесение изменений **запрещается**.

Отменяются также **все** изменения, **уже** **сделанные** триггером.

Компоненты триггера

1. *Ограничения*, для реализации которых создается триггер.
2. *Событие*, которое будет характеризовать возникновение ситуации, требующей проверки ограничений.
(Вставка, удаление и обновление строк в таблице)

Компоненты триггера

3. *Предусмотренное действие*

осуществляется за счет выполнения процедуры или последовательности процедур, с помощью которых реализуется логика, требуемая для реализации ограничений.

Триггер выполняется **неявно** в каждом случае возникновения триггерного события.

Приведение его в действие называют **запуском триггера**.

Цели применения триггеров:

1. Реализация бизнес-правил:

- Проверка корректности введенных данных
- Автоматическое выполнение необходимых расчетов
- Выдача необходимых предупреждений

Цели применения триггеров:

2. Принудительное обеспечение ограничений целостности данных

Известно два типа ограничений для обеспечения целостности данных:

- декларативные ограничения, которые определяются с помощью инструкций языка **CREATE TABLE** и **ALTER TABLE**;
- процедурные ограничения целостности, которые и реализуются посредством триггеров.

3. Создание журнала аудита действий в таблицах БД

Типы триггеров:

1. **Insert** — определяет действия до или после добавления новой записи в таблицу.
2. **Update** — определяет действия до или после изменения записи таблицы.
3. **Delete** — определяет действия до или после удаления записи из таблиц.

Основные составляющие триггера

1. **Имя;**
2. **Действие:** действием триггера может быть или инструкция DML (INSERT, UPDATE или DELETE), или инструкция DDL.;
3. **Исполнение** Исполнительная составляющая триггера обычно состоит из хранимой процедуры или пакета.

Создание триггера

```
CREATE TRIGGER [имя_триггера]
  ON имя_таблицы
  {FOR | AFTER | INSTEAD OF}
    {[INSERT] [,] [UPDATE] [,] [DELETE]}
    [WITH ENCRYPTION]

AS

  IF UPDATE (имя_столбца)
    [{AND | OR} UPDATE
    (имя_столбца)...]
    SQL_операторы
```

Предложение *FOR*

показывает, что триггер запускается **до выполнения операции по модификации данных** (и других каскадно-запускаемых действий и проверок ограничений).

Только для базовых таблиц (не для временных таблиц и представлений).

Предложение *AFTER*

показывает, что триггер запускается **только после успешного выполнения** операции по модификации данных (и других каскадно-запускаемых действий и проверок ограничений).

Только для базовых таблиц.

Триггер *INSTEAD OF*

запускается **вместо операции**, которая запустила триггер.

Этот триггер выполняется **после создания соответствующих таблиц *inserted* и *deleted***, но перед выполнением проверки ограничений целостности или каких-либо других действий.

Для таблиц и представлений.

Триггер *INSTEAD OF*

Часто применяется для **корректировки значений** в **представлениях** путем изменения значения в базовой таблице, на котором оно основано.

Преимущества использования триггеров:

1. Создаются в БД один раз и больше можно не вспоминать о них при разработке **разных приложений** для доступа к данным.

Преимущества использования триггеров:

2. При необходимости триггеры можно **изменять централизованно непосредственно** в БД.

Пользовательские программы, использующие данные из этой базы данных, не требуют модернизации.

Преимущества использования триггеров:

3. Система обработки данных, использующая триггеры, обладает **лучшей переносимостью в архитектуру клиент-сервер** за счет меньшего объема требуемых модификаций.

ОГРАНИЧЕНИЯ:

1. Нельзя использовать в теле триггера операции **создания объектов БД** (новой БД, новой таблицы, новой ХП, нового триггера, новых представлений).
2. Нельзя использовать в триггере команду **удаления объектов DROP** для всех типов базовых объектов базы данных.

ОГРАНИЧЕНИЯ:

3. Нельзя использовать в теле триггера команды изменения базовых объектов **ALTER TABLE, ALTER DATABASE.**

4. Нельзя изменять права доступа к объектам базы данных, т.е. выполнять команду **GRANT** или **REVOKE.**

ОГРАНИЧЕНИЯ:

5. Триггер **не может возвращать значений**, он запускается автоматически сервером и не может связаться самостоятельно с клиентом.

6. Внутри триггера **не допускается** выполнение **восстановления резервной копии** БД или **журнала транзакций**.