13

**Хранимая процедура** – это поименованный код на языке Transact-SQL. Хранимая процедура может быть создана с помощью CREATE, изменена с помощью ALTER и удалена с помощью оператора DROP. Процедура может принимать входные и формировать выходные параметры. Результатом ее выполнения может быть целочисленное значение, которое возвращается к точке вызова оператором RETURN, либо один или более результирующих наборов, сформированных операторами SELECT, либо содержимое стандартного выходного потока, полученного при выполнении операторов PRINT.

Вызов процедуры осуществляется оператором EXECUTE (EXEC). В хранимых процедурах допускается применение основных DDL, DML и TCL-операторов, конструкций TRY/CATCH, курсоров, временных таблиц.

14

**Функция** – это объект БД, представляющий собой поименованный код T-SQL. Для создания, удаления и изменения функций надо использовать операторы CREATE, DROP и ALTER соответственно.

Отличие функций от хранимых процедур в ограничениях, накладываемых на код функции, в форме представления результата работы, а также в способе вызова. В функции не допускается применение DDL-операторов, DML-операторов, изменяющих БД (INSERT, DELETE, UPDATE), конструкций TRY/CATCH, а также использование транзакций.

Результатом выполнения функции является возвращаемое к точке вызова значение. Если функция возвращает единственное значение (число, строка, дата, время и пр.), то она называется *скалярной*. Функция, возвращающая таблицу, называется *табличной*. В зависимости от структуры кода, различают *встроенные* функции и *многооператорные* табличные функции.

При вызове функции надо указывать ее имя с точностью до схемы БД. Если при создании функции имя схемы не указано, то она размещается по умолчанию в схеме **DBO**.

Скалярная функция возвращает единственное значение и принимает любое количество параметров. Допускается возвращать и принимать все типы данных, за исключением TIMESTAMP и устаревших IMAGE, TEXT и NTEXT.

Встроенные табличные функции возвращают к точке вызова результирующий набор SELECT-запроса и по своим возможностям очень напоминают представления.

Встроенные функции допускается применять в операциях внутреннего (INNER JOIN) и внешнего (OUTER JOIN) соединения таблиц

Встроенные функции не дают никакого выигрыша в производительности ни в сравнении с явными SELECT-запросами, ни в сравнении с представлениями. Главная ценность встроенных функций в их повторной применимости, а также в возможности существенно сократить размер SELECT-запроса.

Многооператорные табличные функции возвращают к точке вызова результирующий набор и допускают более сложную логику работы.

15

**Триггер** – это особый вид хранимой процедуры, предназначенной для обработки событий в БД. Поддерживается два типа триггеров: *DDL-триггеры* и *DML-триггеры*.

**DML-**триггеры бывают двух типов: AFTER-триггеры и INSTEAD OF-триггеры. Триггеры типа AFTER исполняются *после* выполнения оператора, вызвавшего соответствующее событие. При этом создаются автоматически две псевдотаблицы INSERTED и DELETED.

Т`риггер типа INSTEAD OF выполняется *вместо* оператора, вызвавшего соответствующее событие. Выполнение INSTEAD OF триггера предшествует проверке установленных для таблицы ограничений целостности.

Событие INSERT при выполнении AFTER-триггера приводит к тому, что в псевдотаблицу INSERTED помещаются строки, добавленные оператором INSERT, вызвавшим это событие. Псевдотаблица DELETED остается пустой.

При возникновении события DELETE в таблицу DELETED копируются удаленные строки, а таблица INSERTED остается пустой.

При изменении строк таблицы с помощью оператора UPDATE заполняются обе псевдотаблицы, при этом таблица INSERTED содержит обновленные версии строк, а таблица DELETED − версию строк до их изменения.

Важной особенностью AFTER-триггера является то, что он вызывается после выполнения активизирующего его оператора. Поэтому, если оператор нарушает ограничение целостности, то возникшая ошибка не допускает выполнения этого оператора и соответствующих триггеров.

Различают два вида DDL-триггеров: триггеры уровня сервера (ALLSERVER), триггеры уровня БД (DATABASE). Триггеры уровня сервера могут обрабатывать события сервера СУБД (создание, изменение или удаление объектов сервера, подключение к серверу и пр.). Триггеры уровня базы данных предназначены для обработки событий, происходящих в рамках БД.