|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ | | | | | | | | |
|  | | |  | | |  | | |
| ФГБОУ ВО «Пермский государственный  национальный исследовательский университет» | | | | | | | | |
|  | | |  | | |  | | |
|  | | ОТЧЕТ  по лабораторной работе № 4 «Метаданные базы данных» по дисциплине «Базы данных и СУБД» | | | | |  | |
|  | | |  | | |  | | |
|  | Работу выполнил  студент гр. \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Зимин И. В.\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 | | |  | Проверил  \_\_\_\_\_\_\_\_\_\_\_ кафедры МОВС  (доц., ст. преп., асс.)  Постаногов И. С.\_\_\_\_\_\_\_  (подпись)  «\_\_\_» \_\_\_\_\_\_\_\_ 2018 | | |  |
|  |  | | |  |  | | |  |
| Пермь 2018 | | | | | | | | |

1 Постановка задачи

Решить проблему неконтролируемых ошибок разработанной ранее хранимой процедуры (ХП) в случае, если в качестве параметров передана несуществующая таблица, несуществующий столбец существующей таблицы или столбец, который не имеет целочисленный тип, посредством проверки корректности параметров на основе метаданных базы данных. В случае некорректных параметров самостоятельно порождать ошибку.

Внести такие изменения в код ХП, чтобы имя триггера формировалось по шаблону "{имя таблицы}\_{имя столбца}\_{номер триггера}", где номер триггера выбирается как количество существовавших ранее триггеров для этой таблицы + 1. Перед созданием требуется проверить существование другого объекта в базе данных с предлагаемым именем. В случае, если такой объект уже есть, требуется сформировать имя любым другим образом.

Провести тестирование корректности работы программы.

Написать отчёт о проделанной работе.

2 Описание алгоритма работы программы

Метаданные — это информация об используемых данных. Структурированные данные, представляющие собой характеристики описываемых сущностей для целей их идентификации, поиска, оценки, управления ими.

Напомню, задача в которой требовалось создавать триггер для каждой новой пары имя таблицы и имя столбца решила проблемы вставки и изменения.

Но мы по-прежнему можем подать в качестве параметров в нашу ХП какие-либо имена несуществующих таблиц или столбцов, или в качестве имени столбца передать имя существующего столбца, но не целого типа. При этом мы получим неосмысленную ошибку. Поэтому этот случай надо обработать.

* В метаданных о столбцах БД найдем такую запись, где имя столбца совпадает с текущим, имя таблицы, которой принадлежит столбец равен текущему, и тип текущего столбца равен 'int'.
* Если такая запись не найдена выдаём ошибку следующего вида "Недопустимое имя или тип столбца или имя таблицы".
* Иначе продолжается работа ХП.

Так же следует сделать нововведения в генерацию имени триггера, теперь так называемая "осмысленная часть" будет иметь следующий вид {имя таблицы}\_{имя столбца}\_{номер триггера}. И вместо использования GUID мы будем добавлять после символа '\_' номер триггера, это число, равное количеству триггеров для текущей таблицы + 1. Но и GUID так же нам пригодится, если вдруг такое имя объекта уже существует в БД то ко всему выше описанному имени триггера мы добавим GUID.

* Найдём количество всех триггеров, относящихся к данной таблице, воспользовавшись метаданными о всех триггерах БД.
* Прибавим к найденному количеству единицу тем самым получив номер триггера и составим имя триггера следующим образом: {имя таблицы}\_{имя столбца}\_{номер триггера}
* Проверим с помощью метаданных о всех объектах БД, существует ли составленное выше имя для триггера среди имён объектов.
* Если существует, то добавляем GUID
* Если не существует продолжается работа ХП без изменения имени

3 Особенности реализации на SQL

Метаданные в SQL — особые таблицы, отличающиеся от «обычных» тем, что, во-первых, за их состоянием следит сама СУБД, а во-вторых тем, что эти таблицы описывают структуру всех объектов в составе БД.

Воспользоваться пришлось двумя каталогами метаданных. А именно каталогом INFORMATION\_SCHEMA для получения метаданных, эта схема поддерживается во многих СУБД и содержится в любой БД и sys, специфичный для MS SQL server. При помощи первого каталога будем решать проблему корректности ввода параметров.

Для выборки данных о столбцах БД воспользуемся представлением INFORMATION\_SCHEMA.COLUMNS. Следующим образом будет выглядеть запрос на проверку существования записи с поданными параметрами:

NOT EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = @tableName AND COLUMN\_NAME = @columnName AND DATA\_TYPE = 'int')

С помощью EXISTS проверяем нашлась ли соответствующая запись

Если запрос вернет истину то мы выдаем ошибку при помощи RAISERROR

RAISERROR('Недопустимое имя или тип столбца или имя таблицы', 16, 1)

При помощи каталога sys будем генерировать имя триггера, именно этот каталог я использую потому что не нашёл нужной информации о триггерах в INFORMATION\_SCHEMA. Для подсчёта количества триггеров используются метаданные о триггерах sys.triggers. Производим подсчёт всех триггеров, относящихся к текущей таблице, для этого из sys.triggers сравниваем parent\_id что является идентификатором таблицы к которой он относится с идентификатором текущей таблицы, который был получен при помощи функции OBJECT\_ID

DECLARE @cntTriggers int

SELECT @cntTriggers = count(\*)+1 FROM sys.triggers

WHERE parent\_id = OBJECT\_ID(@tableName)

Для проверки существования такого имени воспользуемся представлением sys.objects в котором содержатся данные о всех объектах БД. Мы воспользуемся полем name которое будем сравнивать с текущим именем таблицы. И с помощью EXISTS определим нашёлся ли объект с таким же именем, которое мы заготовили для триггера. Запрос выглядит следующим образом:

EXISTS (SELECT \* FROM sys.objects WHERE name = @nameTrigger)

Если мы получили истину то добавляем GUID при помощи функции NEWID

Тестирование программы

Рассмотрим следующие варианты входных параметров:

1. Вызов ХП от несуществующей таблицы
2. Вызов ХП от несуществующего столбца
3. Вызов ХП от существующей таблицы и существующего столбца, столбец не целочисленный
4. Вызов ХП от корректных данных для проверки правильности создания имени триггера, при этом объектов с составленным именем нет.
5. Создадим таблицу, подобрав имя для нее таким образом, чтобы в последствии при создания триггера для какой-либо таблицы совпало с данным.
6. Создание таблицы и вызов ХП от параметров содержащих спец символы и drop table для проверки на SQL-инъекцию.

В случаях 1 - 3 была выведена ошибка

Тест № 4

Создадим таблицу с именем «test» с тремя параметрами целого типа

|  |  |  |
| --- | --- | --- |
| num1 | num2 | num3 |
| int | int | int |

Трижды вызвав ХП получим следующий список триггеро

|  |
| --- |
| triggerName |
| test\_num1\_1 |
| test\_num2\_2 |
| test\_num3\_3 |

Тест №5

Создадим две таблицы одну с именем test1\_num\_1 с любым столбцом.

И таблицу test1 со столбцом num

|  |
| --- |
| num |
| int |

Вызвав ХП от параметров test1 и num будет создан триггер со следующим именем:

|  |
| --- |
| triggerName |
| test1\_num\_1BA457FE52-2C93-49BE-BF21-A521440452F6 |

Тест №6

Создадим таблицу с именем 'drop table spec \n!@$%&\*(')}?{<>`-+'' и столбцом test после вызова ХП создастся триггер с именем.

|  |
| --- |
| triggerName |
| drop tale spebc \n!@$%&\*(')}?{<>`-+'\_test\_ |

На данном наборе тестов программа отработала как нужно.