|  | | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | |  | | |  | | |
|  | «Пермский государственный национальный  исследовательский университет» | | | | | | | |  |
|  | | | |  | | |  | | |
|  | | | ОТЧЕТ  О ВЫПОЛНЕНИИ ЗАДАНИЯ №3  ПО ДИСЦИПЛИНЕ БД И СУБД  по теме:  «Хранимые процедуры» | | | | |  | |
|  | | | |  | | |  | | |
|  | |  | | |  | Работу  Выполнил  Студент гр.ПМИ-4-20,  3 курс  Пуховкин В.И. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_\_»\_\_\_\_\_\_\_2023г | | |  |
|  | | | | Пермь 2023 | | |  | | |

**1. Задание**

Написать хранимую процедуру (ХП), которая возвращает следующее целое число в столбце. Для этого используется отдельная спец. таблица, в которой есть столбцы `id`, `имя таблицы`, `имя столбца` и `текущее максимальное значение`. Пользователь (программист) передаёт в функцию параметром имя таблицы и имя столбца. ХП ищет есть ли такая запись в спец. таблице. Если запись есть, то значение инкрементируется, после чего возвращается пользователю. Если такой записи нет - ХП сперва ищет максимальное число в столбце в запрашиваемой таблице, записывает новую строку, содержащую следующее за найденным число, в спец. таблицу и возвращает это значение пользователю. При отсутствии значений в запрашиваемой таблице, пользователю возвращается 1, и этот же результат записывается в спец. таблицу. Следующий идентификатор для новой строки в спец. таблице формируется рекурсивным вызовом разработанной ХП.

Провести тестирование корректности работы программы.

**3. Решение.**

**Описание алгоритма:**

1. Создание спец. таблицы, в которой есть столбцы `id`, `имя таблицы`, `имя столбца` и `текущее максимальное значение`.
2. Создание хранимой процедуры (ХП), принимающей на вход имя таблицы и имя столбца, которая возвращает следующее целое число в столбце.
3. Внутри ХП выполняем проверку, на наличие в специальной таблице записи с полученными на вход данными об имени таблицы и имени столбца. Если ХП находит данную запись, то инкрементируем значение, находящееся в столбце “CurrentMaxValue” и переходим к шагу 5.

В случае, если такой записи нет, переходим к шагу 4.

1. ХП ищет максимальное число в запрашиваемой таблице и затем инкрементирует данное число и переходим к шагу 5. Если в запрашиваемой таблице отсутствуют значения, то берём число 1 число и переходим к шагу 5.
2. Возвращаем полученное значение.

**Описание реализации:**

1. **Создание функции:**

CREATE [ OR REPLACE ] FUNCTION

***имя*** ( [ [ ***режим\_аргумента*** ] [ ***имя\_аргумента*** ] ***тип\_аргумента*** [ { DEFAULT | = } ***выражение\_по\_умолчанию*** ] [, ...] ] )

[ RETURNS ***тип\_результата***

| RETURNS TABLE ( ***имя\_столбца*** ***тип\_столбца*** [, ...] ) ]

LANGUAGE ***имя\_языка***

CREATE OR REPLACE FUNCTION создаёт новую функцию, либо заменяет определение уже существующей.

Если указано имя схемы, функция создаётся в заданной схеме, в противном случае — в текущей. Имя новой функции должно отличаться от имён существующих функций с такими же типами аргументов в этой схеме. Однако функции с аргументами разных типов могут иметь одно имя (это называется *перегрузкой*).

Чтобы заменить текущее определение существующей функции, используйте команду CREATE OR REPLACE FUNCTION. Но учтите, что она не позволяет изменить имя или аргументы функции (если попытаться сделать это, на самом деле будет создана новая, независимая функция). Кроме того, CREATE OR REPLACE FUNCTION не позволит изменить тип результата существующей функции. Чтобы сделать это, придётся удалить функцию и создать её заново. (Это означает, что если функция имеет выходные параметры (OUT), то изменить типы параметров OUT можно, только удалив функцию.)

Когда команда CREATE OR REPLACE FUNCTION заменяет существующую функцию, владелец и права доступа к этой функции не меняются. Все другие свойства функции получают значения, задаваемые командой явно или по умолчанию. Чтобы заменить функцию, необходимо быть её владельцем (или быть членом роли-владельца).

Если вы удалите и затем вновь создадите функцию, новая функция станет другой сущностью, отличной от старой; вам потребуется так же удалить существующие правила, представления, триггеры и т. п., ссылающиеся на старую функцию. Поэтому, чтобы изменить определение функции, сохраняя ссылающиеся на неё объекты, следует использовать CREATE OR REPLACE FUNCTION.

BEGIN и END показывают начало и конец блока и обрамляются символами “$$”.

В качестве имени языка выбираем plpgsql, это нам необходимо для использования его функционала.

1. **Выполнение динамически формируемых команд:**

EXECUTE ***строка-команды*** [ INTO [STRICT] ***цель*** ] [ USING ***выражение*** [, ... ] ];

Оператор EXECUTE предусмотрен для исполнения динамических команд, где ***строка-команды*** это выражение, формирующее строку (типа text) с текстом команды, которую нужно выполнить. Необязательная ***цель*** — это переменная-запись, переменная-кортеж или разделённый запятыми список простых переменных и полей записи/кортежа, куда будут помещены результаты команды. Необязательные выражения в USING формируют значения, которые будут вставлены в команду.

Все необходимые значения переменных должны быть вставлены в командную строку при её построении, либо нужно использовать параметры, как описано ниже.

Также, нет никакого плана кеширования для команд, выполняемых с помощью EXECUTE. Вместо этого план создаётся каждый раз при выполнении. Таким образом, строка команды может динамически создаваться внутри функции для выполнения действий с различными таблицами и столбцами.

Предложение INTO указывает, куда должны быть помещены результаты SQL-команды, возвращающей строки. Если используется переменная строкового типа или список переменных, то они должны в точности соответствовать структуре результата запроса (когда используется переменная типа record, она автоматически приводится к строковому типу результата запроса). Если возвращается несколько строк, то только первая будет присвоена переменной(ым) в INTO. Если не возвращается ни одной строки, то присваивается NULL. Без предложения INTO результаты запроса отбрасываются.

С указанием STRICT запрос должен вернуть ровно одну строку, иначе выдаётся сообщение об ошибке.

В тексте команды можно использовать значения параметров, ссылки на параметры обозначаются как $1, $2 и т. д. Эти символы указывают на значения, находящиеся в предложении USING. Такой метод зачастую предпочтительнее, чем вставка значений в команду в виде текста: он позволяет исключить во время исполнения дополнительные расходы на преобразования значений в текст и обратно, и не открывает возможности для SQL-инъекций, не требуя применять экранирование или кавычки для спецсимволов.

FORMAT определяет возвращаемое значение.

QUOTE\_IDENT ( text ) → text

Преобразует аргумент в строку, подходящую для использования в качестве идентификатора в SQL-операторе. При необходимости идентификатор заключается в кавычки (например, если он содержит символы, недопустимые в открытом виде, или буквы в разных регистрах). Если переданная строка содержит кавычки, они дублируются.

1. **Получение максимального значения в столбце:**

\*SELECT COALESCE(description, short\_description, '(none)') ...

Функция COALESCE возвращает первый попавшийся аргумент, отличный от NULL. Если же все аргументы равны NULL, результатом тоже будет NULL. Это часто используется при отображении данных для подстановки некоторого значения по умолчанию вместо значений NULL.

Этот \* запрос вернёт значение description, если оно не равно NULL, либо short\_description, если оно не NULL, и строку (none), если оба эти значения равны NULL.

COALESCE вычисляет только те аргументы, которые необходимы для получения результата; то есть, аргументы правее первого отличного от NULL аргумента не вычисляются.

В нашей задаче в поле description мы используем параметр поиска максимального (max(%I)+1) и значение 1 для short\_description.

**Реализация:**

--1. Создание спец таблицы

CREATE TABLE SpecTable

(

id INT PRIMARY KEY,

TableName VARCHAR NOT NULL,

ColumnName VARCHAR NOT NULL,

CurrentMaxValue INT NOT NULL

);

--2. Добавление в спец. таблицу записи (1, spec, id, 1).

INSERT INTO SpecTable VALUES(1,'spec','id',1);

--3. Создание хранимой процедуры (ХП).

CREATE OR REPLACE FUNCTION Search\_Inc(IN TableNameIN varchar,IN ColumnNameIN varchar, OUT Res int) AS

$$

BEGIN

IF

EXISTS(SELECT \* FROM SpecTable WHERE TableName = TableNameIN AND ColumnName = ColumnNameIN)

THEN

UPDATE SpecTable SET CurrentMaxValue = CurrentMaxValue+1

WHERE TableName = TableNameIN AND ColumnName = ColumnNameIN

RETURNING CurrentMaxValue INTO Res;

ELSE

EXECUTE format('SELECT COALESCE(max(%I)+1,1) FROM %I ', ColumnNameIN,TableNameIN) INTO Res;

INSERT INTO SpecTable(id, TableName, ColumnName, CurrentMaxValue)

VALUES(Search\_Inc('spec', 'id'), (TableNameIN), (ColumnNameIN), Res);

END IF;

END;

$$ LANGUAGE plpgsql;

**4. Тестирование.**

--4. вызов ХП с параметрами 'spec' и 'id'

SELECT Search\_Inc('spec','id');

--5. распечатка спец. таблицы

SELECT \* FROM spectable;

--6. вызов ХП с параметрами 'spec' и 'id'

SELECT Search\_Inc('spec','id');

--7. распечатка спец. таблицы

SELECT \* FROM spectable;

--8. создаем новую таблицу с одним столбцом 'id'

CREATE TABLE test

(

id INT NOT NULL

);

--9. добавляем в таблице test новой записи (10)

INSERT INTO test VALUES(10);

--10. вызов ХП с параметрами 'test' и 'id'

SELECT Search\_Inc('test','id');

--11. распечатка спец. таблицы

SELECT \* FROM spectable;

--12. вызов ХП с параметрами 'test' и 'id'

SELECT Search\_Inc('test','id');

--13. распечатка спец. таблицы

SELECT \* FROM spectable;

--14. создаем новую таблицу с столбцами 'num\_value1', 'num\_value2'

CREATE TABLE test2

(

num\_value1 INT NOT NULL,

num\_value2 INT NOT NULL

);

--15. вызов ХП с параметрами 'test2' и 'num\_value1'

SELECT Search\_Inc('test2','num\_value1');

--16. распечатка спец. таблицы

SELECT \* FROM spectable;

--17. вызов ХП с параметрами 'test2' и 'num\_value1'

SELECT Search\_Inc('test2','num\_value1');

--18. распечатка спец. таблицы

SELECT \* FROM spectable;

--19. добавляем в таблице test2 новой записи (2, 13)

INSERT INTO test2 VALUES(2, 13);

--20. вызов ХП с параметрами 'test2' и 'num\_value2'

SELECT Search\_Inc('test2','num\_value2');

--21. распечатка спец. таблицы

SELECT \* FROM spectable;

--22. Удаление ХП

DROP FUNCTION Search\_Inc(TableNameIN varchar, ColumnNameIN varchar);

--23. Удаление таблиц

DROP TABLE SpecTable;

DROP TABLE test;

DROP TABLE test2;