

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе №6
По дисциплине «Базы данных»
«Триггеры»

Работу выполнил студент группы №43501/1

Пузанов А.В. _____

Работу принял преподаватель _____

Мяснов А.В. _____

Санкт-Петербург

2016

1. Цели работы

Познакомить студентов с возможностями реализации более сложной обработки данных на стороне сервера с помощью триггеров.

2. Программа работы

1. Создать два триггера: один триггер для автоматического заполнения ключевого поля, второй триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице
2. Создать триггер в соответствии с индивидуальным заданием, полученным у преподавателя
3. Создать триггер в соответствии с индивидуальным заданием, вызывающий хранимую процедуру
4. Выложить скрипт с созданными сущностями в svn
5. Продемонстрировать результаты преподавателю

3. Ход работы

3.1. Триггер для автоматического заполнения поля ListDoc_id в таблице ListDoctor

Создадим триггер, для автоматического заполнения ключевого поля ListDoc_ID в таблице ListDoctor, если при добавлении новой записи, ListDoc_ID не был указан (NULL). Создадим автоматический генератор ListDoc_ID:

```
CREATE SEQUENCE LISTDOC_GEN;  
ALTER SEQUENCE LISTDOC_GEN RESTART WITH 12;  
SET TERM ^ ;  
CREATE OR ALTER TRIGGER LISTDOC_GEN FOR LISTDOCTOR  
ACTIVE BEFORE INSERT POSITION 0  
AS BEGIN  
IF (NEW.LISTDOC_ID IS NULL) THEN  
NEW.LISTDOC_ID = GEN_ID(LISTDOC_GEN, 1);  
END
```

^

```
SQL> select * from listdoctor;
```

LISTDOC_ID	FULLNAME
1	Uyatkin Igor Vladimirovich
2	Jmurkil Ivan Petrovich
3	Smirnov Aleksandr Kirrilovich
4	Igoraev Nikita Vladimirovich
5	Lukin Vladimir Aleksandrovich
6	Ivanov Andrey Georgievich
7	Jukov Aleksandr Stepanovich
8	Petrov Igor Nikolaevich
9	Ivanov Ivan Ivanovich
10	Petrov Petr Petrovich
11	Berkova Natalia Nikolaevna
12	Marishko Lydmila Petrovna

Выполним команду:

```
INSERT INTO listdoctor(fullname) VALUES ('Petrov Petr Petrovich');
```

```
SQL> INSERT INTO listdoctor(fullname) VALUES ('Petrov Petr Petrovich');
SQL> select * from listdoctor;
```

```
LISTDOC_ID FULLNAME
=====
1 Uyatkin Igor Vladimirovich
2 Jmurkil Ivan Petrovich
3 Smirnov Aleksandr Kirrilovich
4 Igoraev Nikita Vladimirovich
5 Lukin Vladimir Aleksandrovich
6 Ivanov Andrey Georgievich
7 Jukov Aleksandr Stepanovich
8 Petrov Igor Nikolaevich
9 Ivanov Ivan Ivanovich
10 Petrov Petr Petrovich
11 Berkova Natalia Nikolaevna
12 Marishko Lydmila Petrovna
13 Petrov Petr Petrovich
```

3.2. Триггер для контроля целостности данных в подчиненной таблице sale при удалении/изменении записей в главной таблице optionlist

```
CREATE EXCEPTION ERROR_REC 'ERROR: cannot delete!'^
SET TERM ^ ;
CREATE OR ALTER TRIGGER check_rec FOR PRIZYVNIK before delete or update
AS

BEGIN
    if (OLD.privatedata_id IN (select pd_id from privatedata)) then
        EXCEPTION ERROR_REC;
END

^
```

Попробуем удалить данные из главной таблицы:

```
SQL> delete from prizyvnik where pr_id = 7;
Statement failed, SQLSTATE = HY000
exception 2
-ERROR_REC
-ERROR: cannot delete!
-At trigger 'PRIZYUNIK_BUD0' line: 6, col: 5
```

4. Индивидуальное задание

При создании записи о прохождении службы проверять результаты последнего медосвидетельствования. Если группа ниже заданной – не давать создавать запись, выбрасывать исключение.

```
CREATE EXCEPTION Warning1 'groupa nije zadannoy';
SET TERM ^;
CREATE TRIGGER MILLITARYDATA_B10 BEFORE INSERT ON MillitaryData
AS
declare variable group1 int;
begin
```

```

group1 = 4;
if (new.pr_id in (
select prizyvnik.pr_id from prizyvnik,medicaldata
where medicaldata.pr_id=prizyvnik.pr_id and group_id>=4 and md_id not in (
SELECT m2.md_id
from medicaldata as m1, medicaldata as m2
where (m1.pr_id = m2.pr_id
AND m1.medical_date > m2.medical_date)) )
)
then exception warning1;
end
SET TERM;^

```

Попробуем добавить данные о службе к призывнику, у которого группа ниже заданной (B2)

```

SQL> INSERT INTO MillitaryData(military_id, pr_id, action, ranks, mil_division,
Date_From, Date_To) VALUES (7, 2, 1, 1, 1, '01.03.2009', '01.03.2010');
Statement failed, SQLSTATE = HY000
exception 1
-WARNING1
-groupa nije zadannoy
-At trigger 'MILLITARYDATA_BI0' line: 13, col: 10

```

5. Выводы:

Выполнив лабораторную работу №5, мы ознакомились с возможностями реализации более сложной обработки данных на стороне сервера с помощью триггеров. Были созданы различные триггеры в соответствии с поставленным заданием, которые были успешно протестированы на работоспособность.

Так или иначе, было выяснено, что триггера являются мощным инструментом при разработке БД, которые обеспечивают удобство и стабильность при работе БД в дальнейшем. Ведь это та же самая хранимая процедура, только которую пользователь не вызывает непосредственно. Ее исполнение обусловлено действием по модификации данных: добавлением INSERT, удалением DELETE строки в заданной таблице, или изменением UPDATE данных в определенном столбце заданной таблицы реляционной базы данных. Однако стоит помнить, злоупотребление данным инструментом ведет к падению производительности СУБД.

Цель данной работы была достигнута. Полученные знания при необходимости могут быть применены на практике.