Санкт-Петербургский политехнический университет

Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе

Дисциплина: «Базы данных»

Тема: «SQL-программирование: Триггеры, вызов процедур»

Выполнил студент гр. 43501/3 М.Н. Козлов

(подпись)

Преподаватель А.В. Мяснов

(подпись)

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г.

Санкт-Петербург

2016

Оглавление

[Цель работы: 3](#_Toc470428713)

[Программа работы: 3](#_Toc470428714)

[Выполнение работы: 3](#_Toc470428715)

[Триггер для автоматического заполнения ключевого поля 3](#_Toc470428716)

[Триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице 4](#_Toc470428717)

[Индивидуальные задание: триггер 1 4](#_Toc470428718)

[Индивидуальные задание: триггер 2 4](#_Toc470428719)

[Выводы: 5](#_Toc470428720)

# Цель работы:

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

# 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.

# Выполнение работы:

Триггер для автоматического заполнения ключевого поля

В начале работы был создан генератор CAR\_GEN:

CREATE sequence BOOK\_GEN ;

ALTER SEQUENCE BOOK\_GEN RESTART WITH 100001;

Затем создадим триггер, использующий данный генератор. Если id был введен вручную, то автоматически изменяется значения генератора для последующего использования.

create trigger book\_id\_autoinc for table\_book

active before insert position 0

as

declare variable tmp DECIMAL (18 ,0) ;

begin

if (new.id is null ) then

new.id = gen\_id ( BOOK\_GEN ,1) ;

else

begin

tmp = gen\_id ( BOOK\_GEN ,0) ;

if ( tmp < new.id ) then

tmp = gen\_id ( BOOK\_GEN , new.id - tmp ) ;

end

end

Триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице

Создадим исключение и триггер для контроля целостности данных в подчиненных таблицах при удалении/изменении записей в главной таблице CAR:

create exception ex\_modify\_book 'This book in other tables !';

create trigger book\_delete\_update for table\_book

before delete or update

as

begin

if (old.id in( select book\_id from table\_order\_list))

then exception ex\_modify\_book ;

end

Индивидуальные задание: триггер 1

Задание:

1. При добавлении книги в заказ вызывать процедуру 1 из [#104](http://tiger.ftk.spbstu.ru/trac/edu-db-2016/ticket/104).
2. Не давать повторно добавлять книгу в заказ выбрасывая исключение.

connect 'D:\Database\booksshop\BooksShop.fdb' user 'SYSDBA' password 'masterkey';

set term ^;

drop trigger new\_book\_for\_order;

drop exception ex\_isexist;

create exception ex\_isexist 'The duplicating record';

create trigger new\_book\_for\_order for table\_order\_list

active before insert

declare variable ex integer;

as

begin

execute procedure name\_procedure\_104(new.order\_id, new.book\_id, new.price, new.number)

RETURNING\_VALUES :ex;

if( ex > 0 ) then

exeption ex\_isexist;

end

^

set term ; ^

Индивидуальные задание: триггер 2

Задание:

При превышении количества продаж книги выше заданного выдавать ей специальную премию однократно.

connect 'D:\Database\booksshop\BooksShop.fdb' user 'SYSDBA' password 'masterkey';

set term ^;

drop trigger special\_reward;

drop exception ex\_isexist;

create exception ex\_no 'There is no necessary to give a special reward';

create trigger special\_reward for table\_order\_list

active before insert /\* при insert нового заказа \*/

declare variable number\_in\_stock integer;

as

begin

select count(table\_order\_list.number) into :number\_in\_stock from table\_order\_list where table\_order\_list.book\_id = :new.book\_id;

if(number\_in\_stock > 100) then

insert into table\_book\_rewards values (1, :new.book\_id, 2016); /\* 1 - id специальной награды \*/

else

exeption ex\_no;

end

^

set term ; ^

# Выводы:

В результате выполнения данной лабораторной работы ознакомился со сложной обработкой данных на стороне сервера при помощи хранимых процедур и триггеров.

В ходе работы были созданы триггера в соответствии с общими и индивидуальными заданиями.

Можно заметить, что триггер очень похож на хранимую процедуру, но запуск его, в отличии от ХП, производится автоматически сервером при модификациях данных таблицы. Триггер очень полезен, так как он ограждает БД от пользователей, пользователь не имеет доступа к таблицам, а значит не может их повредить. Триггер не может навредить БД так как выполняется в виде транзакции и при обнаружении ошибки, порчи таблицы произойдет отказ этой транзакции.

Триггер запускается с помощью ключей BEFORE или ALTER, что определяет момент его запуска – до выполнения, связанного с ним события или после, соответственно.