Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №6**

**Дисциплина**: Базы данных

**Тема**: Триггеры

Выполнила студентка гр. 43501/1 В.В. Ракова

(подпись)

Руководитель А.В. Мяснов

(подпись)

“ ” 2015 г.

Санкт-Петербург

2015

1. **Цель**

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

1. **Программа работы**
2. 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
3. Создать триггер в соответствии с **индивидуальным заданием**, полученным у преподавателя
4. Создать триггер в соответствии с **индивидуальным заданием**, вызывающий хранимую процедуру
5. Выложить скрипт с созданными сущностями в svn
6. Продемонстрировать результаты преподавателю
7. **Выполнение работы**

Был создан триггер для автоматического заполнения поля id\_genre в таблице GENRE

--create generator increment;

--set generator increment to GENRE.id\_genre;

drop trigger auto\_fill;

create trigger auto\_fill for GENRE before insert

as

begin

new.id\_genre = gen\_id(increment,1);

end;

Затем был создан триггер для контроля целостности данных в подчиненной таблице при удалении/изменении записей в главной таблице AUTHOR

create exception error1 'Error deleting author';

drop trigger check\_cel;

create trigger check\_cel for author before delete or update

as

begin

if (OLD.id\_author in (select prize.id\_author from prize)) then

exception error1;

end;

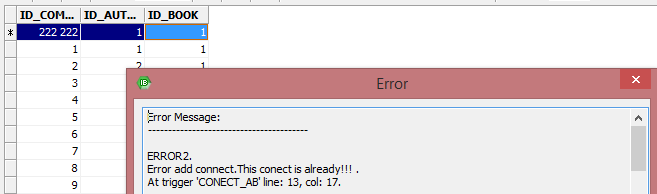
Реализовать триггеры:

1. При добавлении связи автора и книги проверять наличие такой связи. Если есть - не добавлять.
2. При выходе нового тиража добавлять запись по вознаграждении автора за тираж по некоторой формуле.

Первый триггер:

|  |
| --- |
| drop trigger CONECT\_AB;  CREATE trigger CONECT\_AB FOR AUTHOR\_BOOK BEFORE INSERT  as  declare variable id\_b int;  declare variable cal int default 0;  declare variable id\_com int;  begin  id\_com =new.id\_comab;  select author\_book.id\_comab from author\_book  where author\_book.id\_author=new.id\_author and  author\_book.id\_book = new.id\_book  into :cal;  if (cal<0) then exception error2;  else if (cal=0) then new.id\_comab=:id\_com;  end |

Добавляем уже существующею связь ID\_AUTHOR =1 , ID\_BOOK =1 , в этом случае должны получить ошибку error2. Результат:



При добавлении новой связи ошибок не возникает.

Второй триггер:

|  |
| --- |
| create trigger add\_prize1 for prize active after insert  as  declare variable num\_edit int;  declare variable num\_iss int;  declare variable id\_pr int;  declare variable addprize float;  begin  select edition.number\_of\_copy, prize.id\_prize from edition , prize  where new.id\_prize = prize.id\_prize and  new.id\_edition = prize.id\_edition and  new.id\_author = prize.id\_author and  prize.id\_edition= edition.id\_edition  into :num\_edit,:id\_pr;  select edition.number\_of\_issue from edition, prize  where new.id\_prize = prize.id\_prize and  new.id\_edition = prize.id\_edition and  new.id\_author = prize.id\_author and  prize.id\_edition= edition.id\_edition  into :num\_iss;  addprize = (num\_iss+num\_iss\*0.5)\*(num\_edit/100000\*2);  execute procedure add\_pr (addprize, id\_pr);  end |

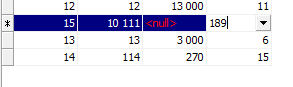
Данный тригер вызывает следующую процедуру:

|  |
| --- |
| create procedure ADD\_PR (ADDPRIZE float, ID\_PR int)  as  begin  update prize set prize\_sum = :ADDPRIZE  where :ID\_PR=prize.id\_prize;  end |

Добавляем премиальный тираж, известен номер издания и количество копий, автор получает 2 ед. за каждые 100000 копий, ценность возрастает в зависимости от номера издания.

Результаты работы:

вводим необходимые данные:



получаем результат по формуле:



**Вывод**

В данной работе были созданы триггеры. Триггеры полезно использовать для проверки корректности вносимых в БД данных и их целостности. С помощью триггеров можно выдавать сообщения (предупреждения) о том, что необходимо выполнить некоторые действия при изменении таблиц.

Также триггеры удобно использовать для оповещения об изменении данных в таблицах.

С помощью триггеров можно накладывать ограничения на вносимые данные согласно требованиям предметной области БД.