МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Н.В. Путилова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7 |
| Триггеры. Обеспечение активной целостности данных базы данных |
| по курсу: Проектирование баз данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4932 |  |  |  | Н.С. Иванов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

#### Текст задания

8. костюмерная театра: роль, спектакль, название костюма, деталь костюма, размер, автор модели, дата разработки

а. спектакли, в которых используются костюмы, имеющие в названии слово «принц»

б. костюм, в котором есть и плащ и штаны

в. спектакль, на который пока нет костюмов

г. роль, к которой разрабатывался самый старый из костюмов

д. автор, разработавший наибольшее число костюмов

е. костюм, в котором есть все типы деталей

ж. автор, не разрабатывавший костюмы к «Золушке», но разрабатывавший к «Мастеру и Маргарите»

#### Физическая модель



#### Текст запросов на SQL

##### table author\_log

create table if not exists author\_log(  
 id int not null auto\_increment primary key,  
 id\_author int not null,  
 firstNameAuthor varchar(40) default null,  
 secondNameAuthor varchar(40) default null,  
 thirdNameAuthor varchar(40) default null  
);

##### # before insert

*# Если (такого типа нет):  
# добавляет "неизвестный тип";  
# связывает с "неизвестный тип";  
# Либо:  
# связывает с "неизвестный тип";*create trigger tr31  
 before insert on concrete\_detail  
 for each row begin  
 if not *exists*(select *\** from type\_detail  
 where type\_detail.id\_type\_detail = NEW.**id\_type\_detail**) then  
 if not *exists*(select *\** from type\_detail  
 where nameType like 'unknown type') then  
 insert into type\_detail (nameType)  
 values ('unknown type');  
 end if;  
  
 set NEW.id\_type\_detail =  
 (select id\_type\_detail from type\_detail  
 where nameType like 'unknown type');  
 end if;  
 end;

##### # after insert

*# logging*create trigger tr32  
 after insert on author  
 for each row begin  
 insert into author\_log  
 (id\_author, firstNameAuthor,  
 secondNameAuthor, thirdNameAuthor)  
 values  
 (NEW.id\_author, *concat*('<insert trg> ', NEW.firstNameAuthor),  
 NEW.secondNameAuthor, NEW.thirdNameAuthor);  
 end;

##### # before delete

*# При удалении автора удаляет все его связи с одеждой*create trigger tr11  
 before delete on author  
 for each row begin  
 delete from clothes\_author  
 where clothes\_author.id\_author = OLD.id\_author;  
 end;

##### # after delete

*# Если при удалении конкретной детали  
# тип, которому она принадлежала стал избыточным  
# (никому не принадлежащим, ни с чем не связанным)  
#  
# удаляет этот тип*create trigger tr12  
 after delete on concrete\_detail  
 for each row begin  
 if not *exists*(select *\** from concrete\_detail cd  
 where cd.id\_type\_detail = OLD.**id\_type\_detail**) then  
  
 delete from type\_detail  
 where type\_detail.id\_type\_detail = OLD.id\_type\_detail;  
  
 end if;  
 end;

##### # before update

*# Каскадное обновление первичного ключа*create trigger tr21  
 before update on type\_detail  
 for each row begin  
 declare fkc int;  
 set fkc=@@foreign\_key\_checks;  
 set *@@foreign\_key\_checks*=0;  
  
 if (OLD.id\_type\_detail != NEW.id\_type\_detail) then  
 update concrete\_detail  
 set id\_type\_detail = NEW.id\_type\_detail  
 where id\_type\_detail = OLD.id\_type\_detail;  
 end if;  
  
 set @@foreign\_key\_checks=fkc;  
 end;

##### # after update

*# logging*create trigger tr22  
 after update on author  
 for each row begin  
 insert into author\_log  
 (id\_author, firstNameAuthor,  
 secondNameAuthor, thirdNameAuthor)  
 values  
 (OLD.id\_author, *concat*('<before upd> ', OLD.firstNameAuthor),  
 OLD.secondNameAuthor, OLD.thirdNameAuthor);  
 end;

##### # test before insert

insert into concrete\_detail (id\_type\_detail, colorDetail)  
 values (1000, 'test trigger insert before');  
  
select td.id\_type\_detail, td.nameType, cd.colorDetail  
 from concrete\_detail cd  
 left outer join type\_detail td  
 on td.id\_type\_detail = cd.id\_type\_detail;

##### # test after insert

insert into author (firstNameAuthor)  
 values ('test trigger insert');  
select *\** from author\_log;

##### # test before delete

delete from author where id\_author = 1;  
  
select a.id\_author, a.firstNameAuthor, c.nameClothes  
from clothes\_author  
 right outer join clothes c  
 on c.id\_clothes = clothes\_author.id\_clothes  
 right outer join author a  
 on a.id\_author = clothes\_author.id\_author;

##### # test after delete

delete from concrete\_detail cd where id\_concrete\_detail = 1;  
delete from concrete\_detail cd where id\_concrete\_detail = 2;  
delete from concrete\_detail cd where id\_concrete\_detail = 3;  
  
select td.id\_type\_detail, td.nameType, cd.colorDetail, cd.id\_concrete\_detail  
 from concrete\_detail cd  
 right outer join type\_detail td  
 on td.id\_type\_detail = cd.id\_type\_detail;

##### # test before update

update type\_detail  
 set id\_type\_detail = 300  
 where id\_type\_detail = 3;  
  
select td.id\_type\_detail, td.nameType, cd.colorDetail  
 from concrete\_detail cd  
 join type\_detail td  
 on td.id\_type\_detail = cd.id\_type\_detail;

##### # test after update

update author  
 set firstNameAuthor = 'test trigger update'  
 where author.id\_author = 2;  
select *\** from author\_log;