

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

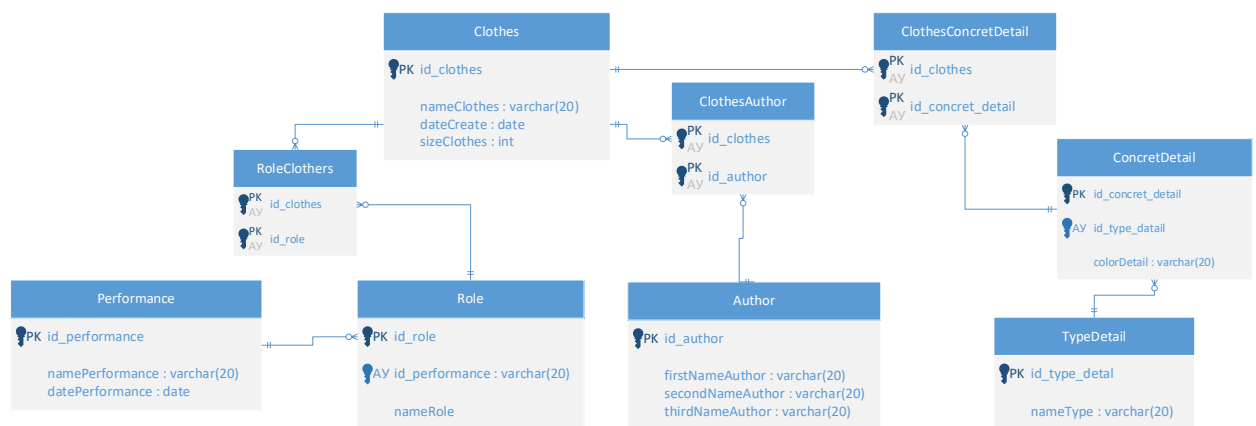
Санкт-Петербург 2021

### 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;

```