

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

должность, уч. степень, звание

подпись, дата

Н.В. Путилова

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Хранимые процедуры

по курсу: Проектирование баз данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4932

подпись, дата

Н.С. Иванов

инициалы, фамилия

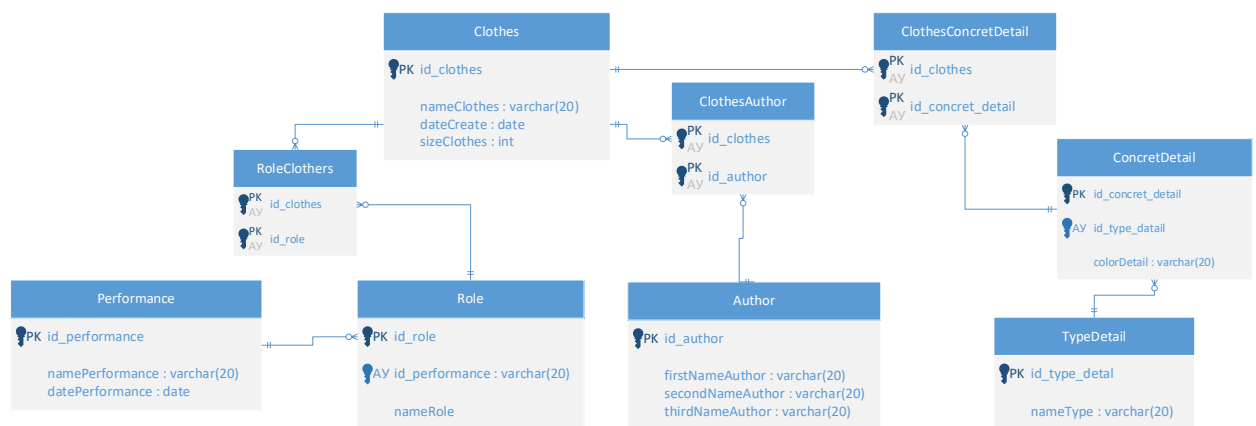
Санкт-Петербург 2021

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

создать в БД хранимые процедуры, реализующие:

#

=====

=====

1

— вставку с пополнением справочников

- получаем ссылку на внешний ключ по значению данных из родительской
таблицы

- если данных нет - добавляем в родительскую

- затем вставляем в дочернюю

delimiter //

```
create procedure insert_concrete_detail (name_type varchar(20), color_detail  
varchar(20)) begin
```

```
    declare type_id_new int;
```

```
    if exists(select * from type_detail where nameType = name_type) then  
        select td.id_type_detail into type_id_new from type_detail td where  
nameType = name_type;
```

```
    else
```

```
        begin
```

```
            insert into type_detail(nameType)
```

```
                values (name_type);
```

```
            set type_id_new = last_insert_id();
```

```
        end;
```

```
    end if;
```

```
    insert into concrete_detail (id_type_detail, colorDetail)
```

```
        values (type_id_new, color_detail);
```

```
end; //
```

delimiter ;

#

```
call insert_concrete_detail('Не существующий тип1', 'Белый1');
```

```
call insert_concrete_detail('Брюки', 'Розовый');
```

details list

```
select cd.id_concrete_detail, cd.colorDetail, td.id_type_detail, td.nameType  
from concrete_detail cd
```

```
    right join type_detail td
```

```
        on cd.id_type_detail = td.id_type_detail
```

```
group by cd.id_concrete_detail;
```

#

=====

=====

2

```

# — удаление с очисткой справочников
#     - если после удаления данных из дочерней у строки родительской
#     больше нет зависимых
#     - удаление данных из родительской таблицы
#
#     (удаляется информация о студенте,
#     если в его группе нет больше студентов,
#     запись удаляется из таблицы с перечнем групп);

# удалить деталь;
# если (нет деталей этого типа):
#     удалить этот тип;

delimiter //

create procedure delete_type_detail(id_concrete_detail_delete int) begin
    declare id_type_detail_delete int;
    select cd.id_type_detail into id_type_detail_delete from concrete_detail
cd
        where cd.id_concrete_detail = id_concrete_detail_delete;

    delete from concrete_detail cd where cd.id_concrete_detail =
id_concrete_detail_delete;
    if not exists(select * from concrete_detail cd where cd.id_type_detail =
id_type_detail_delete) then
        delete from type_detail td where td.id_type_detail =
id_type_detail_delete;
    end if;

end;

delimiter ;

# -----
# -----
call delete_type_detail(1);
call delete_type_detail(2);
call delete_type_detail(21);

# details list
select cd.id_concrete_detail, cd.colorDetail, td.id_type_detail, td.nameType
from concrete_detail cd
    right join type_detail td
    on cd.id_type_detail = td.id_type_detail
group by cd.id_concrete_detail;

#
=====
=====
# 3

# — каскадное удаление
#     (удаление всех зависимых данных)

# удалить связи "одежда" - "конкретная деталь"
# удалить все конкретные детали;
# удалить все типы деталей;

```

```

delimiter //

create procedure cascade_delete_type_detail(id_type_detail_delete int) begin

    delete from clothes_concrete_detail ccd where ccd.id_concrete_detail in
        (select cd.id_concrete_detail from concrete_detail cd where
id_type_detail = id_type_detail_delete);

    delete from concrete_detail cd where id_type_detail =
id_type_detail_delete;

    delete from type_detail td where id_type_detail = id_type_detail_delete;

end;

delimiter ;

# -----
# -----
call cascade_delete_type_detail(1);

# details list
select c.id_clothes, c.nameClothes,
        cd.id_concrete_detail, cd.colorDetail,
        td.id_type_detail, td.nameType
from concrete_detail cd
    right join type_detail td on cd.id_type_detail = td.id_type_detail
    join clothes_concrete_detail ccd on cd.id_concrete_detail =
ccd.id_concrete_detail
    right join clothes c on ccd.id_clothes = c.id_clothes
order by id_type_detail;

#
=====
=====
# 4

# — вычисление и возврат значения агрегатной функции
#     (т.к. агрегатная функция дает единственный результат)
#     (задача – вернуть данные из процедуры/функции)

delimiter //

create procedure count_clothes(out c_clothes int) begin
    select ifnull(count(id_clothes),0) into c_clothes from clothes;

end;

delimiter ;

# -----
# -----
call count_clothes(@c);
select @c;

#
=====
=====
# 5

```

```

# — формирование статистики во временной таблице
#      (задача – работа с временными таблицами)

set sql_safe_updates = 0;
drop procedure if exists authors_statistic;

delimiter //

create procedure authors_statistic() begin

    drop table if exists stat;
    create temporary table if not exists stat(
        id_stat int primary key not null auto_increment,
        id_author int,
        count_clothes_author int,
        avg_count_clothes_author double default 0,
        diff_count_clothes_author double default 0
    );

    insert into stat (id_author, count_clothes_author)
        select id_author, count(ca.id_clothes) as count_clothes
            from clothes c left join clothes_author ca
                on c.id_clothes = ca.id_clothes
            group by id_author;

    update stat set avg_count_clothes_author =
        (select avg(q.count_clothes) from
            (select count(ca.id_clothes) as count_clothes
              from clothes c left join clothes_author ca
                on c.id_clothes = ca.id_clothes
              group by id_author)q);

    update stat s set diff_count_clothes_author = s.count_clothes_author -
s.avg_count_clothes_author
    where id_stat > 0;

    select distinct
        stat.id_author,
        stat.count_clothes_author,
        stat.avg_count_clothes_author,
        stat.diff_count_clothes_author from stat;

    drop table stat;
end;//

delimiter ;

# -----
# -----
call authors_statistic();
set sql_safe_updates = 0;

```