МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

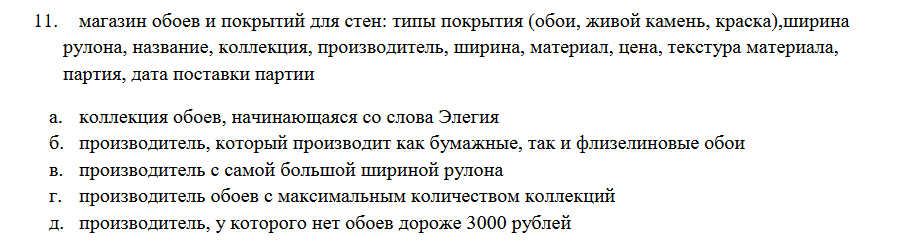
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ст. преподаватель | |  |  | | | |  | | Н. В. Путилова |
| должность, уч. степень, звание | |  | подпись, дата | | | |  | | инициалы, фамилия |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №10 | | | | | | | | | | |
| Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы. | | | | | | | | | | |
| по дисциплине: Проектирование баз данных | | | | | | | | | | |
|  | | | | | | | | | | |
| РАБОТУ ВЫПОЛНИЛА | | | | | | | | | | |
| СТУДЕНТКА ГР. | 4931 | | |  | 22.03.2022 |  | | Е.Ю. Ильченко | | |
|  |  | | |  | подпись, дата |  | | инициалы, фамилия | | |
|  |  | | |  |  |  | |  | | |

Санкт-Петербург 2022

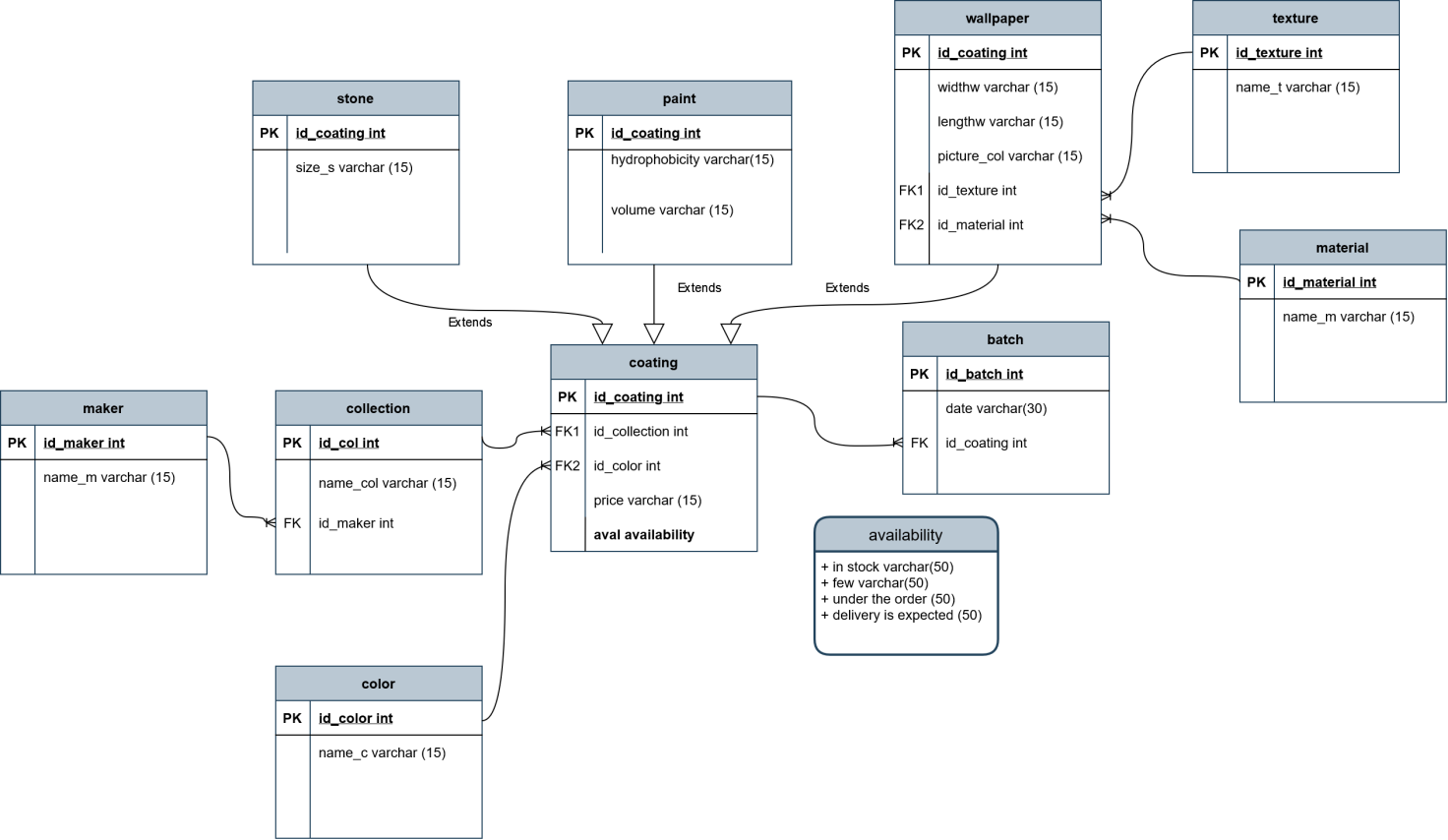
Вариант № 11

1. **Задание**

Выполнить вставку тестовых данных в таблицы, созданные в ходе выполнения лабораторной работы 9. Сделать запрос выборки с условием к таблицам предку и потомку. Придумать и создать пользовательский оператор для своей предметной области. Придумать и создать пользовательскую агрегатную функцию для своей предметной область.



1. **Физическая модель БД**

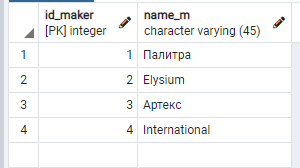
****

1. **Наборы данных, содержащихся в таблицах**

INSERT INTO maker (id\_maker, name\_m) VALUES (1, 'Палитра'), (2, 'Elysium'), (3, 'Артекс'), (4,'International');

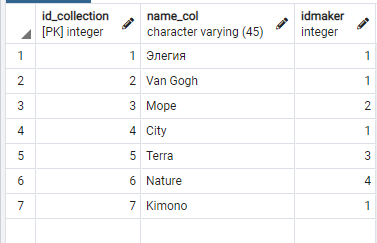
SELECT id\_maker, name\_m

FROM maker;



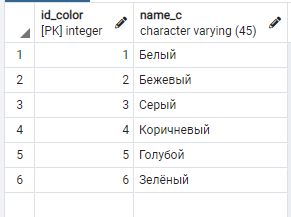
INSERT INTO collection (id\_collection, name\_col, idmaker) VALUES (1, 'Элегия',1), (2, 'Van Gogh',1), (3, 'Море',2), (4, 'City',1), (5, 'Terra',3), (6, 'Nature',4), (7, 'Kimono',1);

SELECT id\_collection, name\_col, idmaker FROM collection;



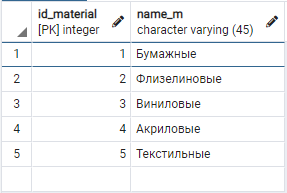
INSERT INTO color (id\_color, name\_c) VALUES (1, 'Белый'), (2, 'Бежевый'), (3, 'Серый'), (4, 'Коричневый'), (5, 'Голубой'), (6, 'Зелёный');

SELECT id\_color, name\_c FROM color;



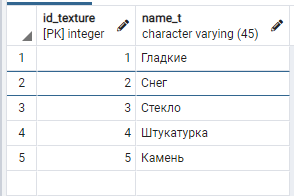
INSERT INTO material (id\_material, name\_m) VALUES (1, 'Бумажные'), (2, 'Флизелиновые'), (3, 'Виниловые'), (4, 'Акриловые'), (5, 'Текстильные');

SELECT id\_material, name\_m FROM material;



INSERT INTO texture (id\_texture, name\_t) VALUES (1, 'Гладкие'), (2, 'Снег'), (3, 'Стекло'), (4, 'Штукатурка'), (5, 'Камень');

SELECT id\_texture, name\_t FROM texture;



INSERT INTO wallpaper (id\_coating, price, aval, idcollection, idcolor, lengthw, widthw, picture\_col, idmaterial, idtexture)

VALUES (1, 2550, 'в наличии', 1, 1, 0.53, 10, 'мраморный узор', 5, 1),

(2, 995, 'мало', 2, 5, 1.06, 10, 'картины Ван Гога', 1, 2),

(3, 3200, 'под заказ', 3, 5, 1.00, 15, 'волны', 2, 3),

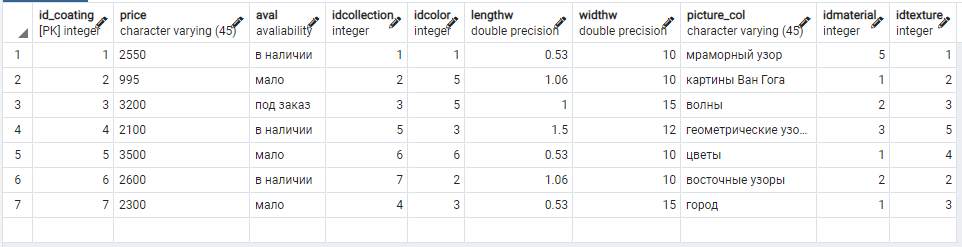
(4, 2100, 'в наличии', 5, 3, 1.50, 12, 'геометрические узоры', 3, 5),

(5, 3500, 'мало', 6, 6, 0.53, 10, 'цветы', 1, 4),

(6, 2600, 'в наличии', 7, 2, 1.06, 10, 'восточные узоры', 2, 2),

(7, 2300, 'мало', 4, 3, 0.53, 15, 'город', 1, 3);

SELECT id\_coating, price, aval, idcollection, idcolor, lengthw, widthw, picture\_col, idmaterial, idtexture FROM wallpaper;



INSERT INTO coating (id\_coating, price, aval, idcollection, idcolor)

VALUES

(8, 1300, 'под заказ', 3, 5),

(9, 1400, 'под заказ', 5, 2),

(10, 1600, 'в наличии', 6, 3);

SELECT id\_coating, price, aval, idcollection, idcolor FROM coating;



**Триггеры для таблицы batch**

**для добавления данных**

DROP TRIGGER insert\_batch ON batch;

DROP FUNCTION insert\_batch ();

CREATE FUNCTION insert\_batch()

RETURNS trigger AS $insert\_batch$

BEGIN

if not exists (select\*from coating where id\_coating=new.id\_coating )

then

RAISE EXCEPTION 'Покрытия с таким id не существует!';

end if;

RETURN NEW;

END;

$insert\_batch$ LANGUAGE plpgsql;

CREATE TRIGGER insert\_batch BEFORE INSERT ON batch FOR EACH ROW

EXECUTE FUNCTION insert\_batch();

INSERT INTO batch (id\_batch, date, id\_coating)

VALUES (1, '2022-03-05',1),

(2, '2022-03-10',2),

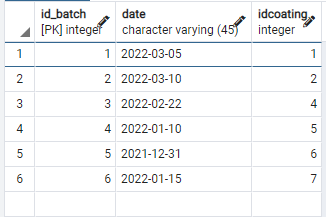
(3, '2022-02-22',4),

(4, '2022-01-10',5),

(5, '2021-12-31',6),

(6, '2022-01-15',7);

SELECT id\_batch, date, id\_coating FROM batch;

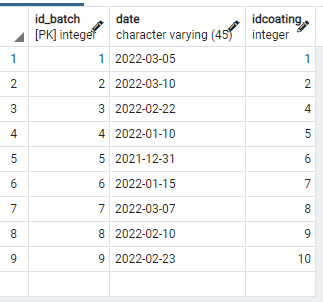


INSERT INTO batch (id\_batch, date, id\_coating)

VALUES (7, '2022-03-07',8),

(8, '2022-02-10',9),

(9, '2022-02-23',10);



**для обновления данных**

DROP TRIGGER update\_coating ON coating;

DROP FUNCTION update\_coating();

CREATE FUNCTION update\_batch()

RETURNS trigger AS $update\_batch$

BEGIN

update batch

set id\_coating=new.id\_coating

where id\_coating=old.id\_coating;

RETURN NEW;

END;

$update\_batch$ LANGUAGE plpgsql;

CREATE TRIGGER update\_batch

AFTER UPDATE ON coating FOR EACH ROW

EXECUTE FUNCTION update\_batch();

**для удаления данных**

DROP TRIGGER delete\_batch ON coating;

DROP FUNCTION delete\_batch();

CREATE FUNCTION delete\_batch()

RETURNS trigger AS $delete\_batch$

BEGIN

delete from batch where

id\_coating=old.id\_coating;

RETURN NEW;

END;

$delete\_batch$ LANGUAGE plpgsql;

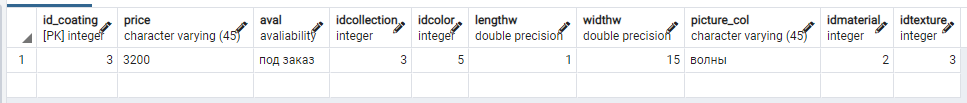
CREATE TRIGGER delete\_batch

AFTER DELETE ON coating FOR EACH ROW

EXECUTE FUNCTION delete\_batch();

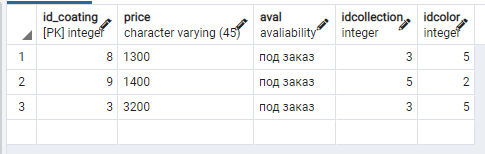
1. **Тексты запросов на SQL**

**Запрос к потомку**

SELECT\*FROM wallpaper where aval='под заказ';

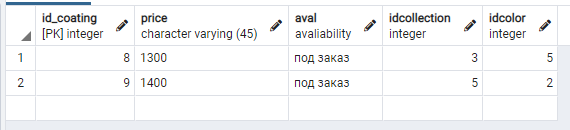
**Запрос к предку**

SELECT\*FROM coating where aval='под заказ';



**Запрос только к предку**

SELECT\*FROM ONLY coating where aval='под заказ';



1. **Код операторов**

**Пользовательский оператор**

CREATE FUNCTION for\_square (lengthw double precision, widthw double precision)

RETURNS double precision LANGUAGE SQL as

$$

select lengthw\*widthw;

$$;

CREATE OPERATOR %!(

leftarg = double precision,

rightarg = double precision,

function = for\_square

);

**Пользовательская агрегатная функция**

CREATE or replace FUNCTION cnt\_w ( s integer, aval avaliability)

RETURNS integer LANGUAGE plpgsql AS

$$

BEGIN

IF aval!='закончился'

then

return s+1;

else return s;

end if;

END;$$;

DROP AGGREGATE COUNT (avaliability)

CREATE AGGREGATE COUNT (avaliability)

(

STYPE = integer,

SFUNC = cnt\_w,

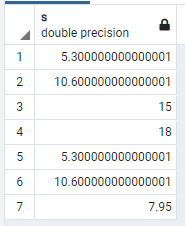
initcond = 0

);

1. **Пример выполнения операторов**

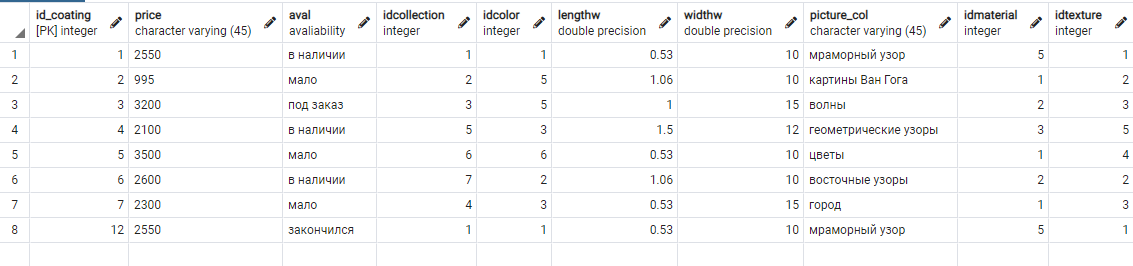
**Для пользовательского оператора**

SELECT (lengthw %! widthw) as S from wallpaper;



**для пользовательской агрегатной функции**

select \*from wallpaper;



select \*from wallpaper;

