МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ТАТАРСТАН

**Государственное автономное профессиональное образовательное учреждение**

**«ЗЕЛЕНОДОЛЬСКИЙ МЕХАНИЧЕСКИЙ КОЛЛЕДЖ»**(ГАПОУ «ЗМК»)

**09.02.07 «Информационные системы и программирование»**

# МДК 07.01. УПРАВЛЕНИЕ И АВТОМАТИЗАЦИЯ БАЗ ДАННЫХ

**Отчет о практических работах**

**Исполнитель**: Камдина Полина Евгеньевна

**Группа**: 227

**Преподаватель:** Алемасов Евгений Павлович

**Дата сдачи** 30.10.2024 **Оценка** \_\_\_\_\_\_\_\_\_\_\_\_

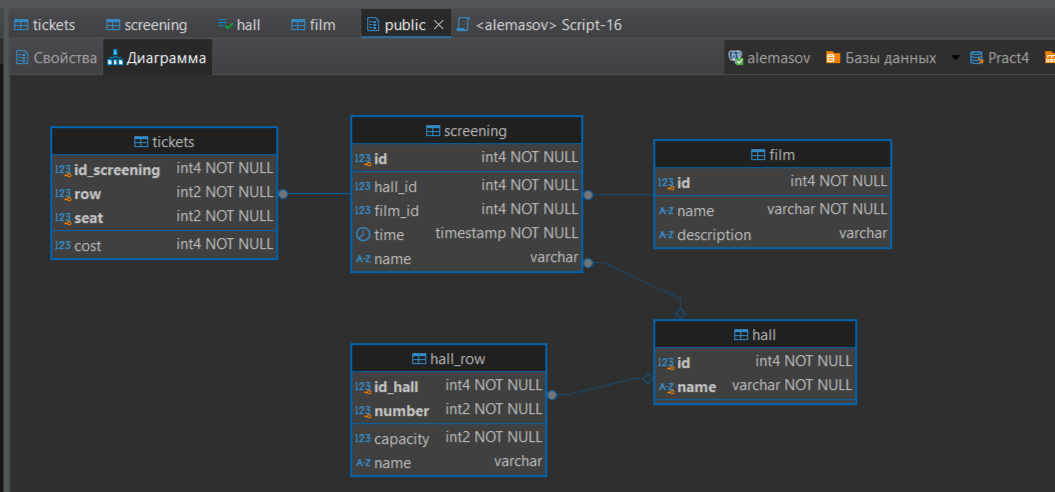
**Подпись преподавателя**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

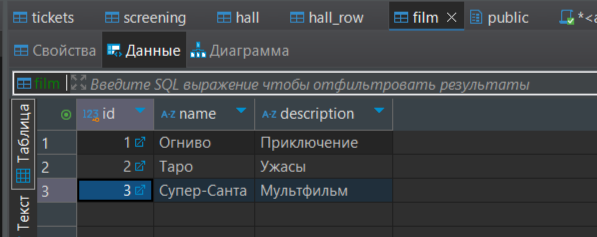
**ЗЕЛЕНОДОЛЬСК – 2023**

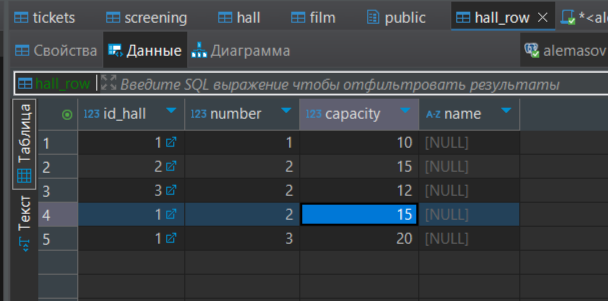
**Работа №5**

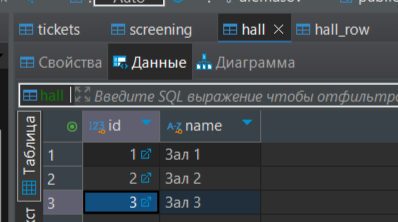
1. Создайте представление, чтобы постоянно знать количество мест во втором ряду третьего зала.
2. Создайте представление, чтобы автоматически получать названия фильмов, показ которых начинается после 11 часов 01.01.2021.
3. Создайте представление для получения расписания проката в конкретном зале кинотеатра.
4. Создайте представление для получения расписания проката конкретного фильма.

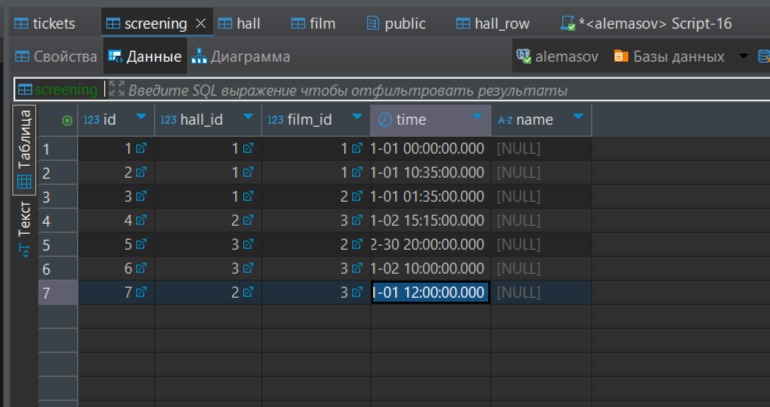
Решение

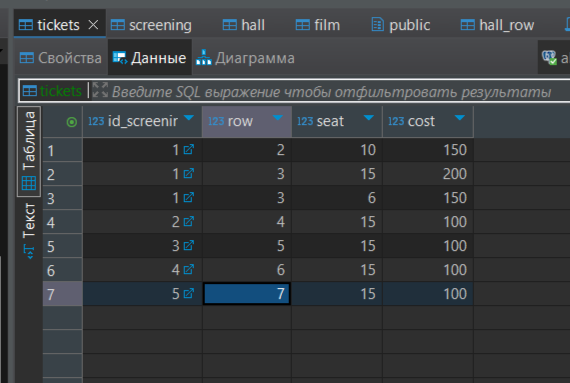




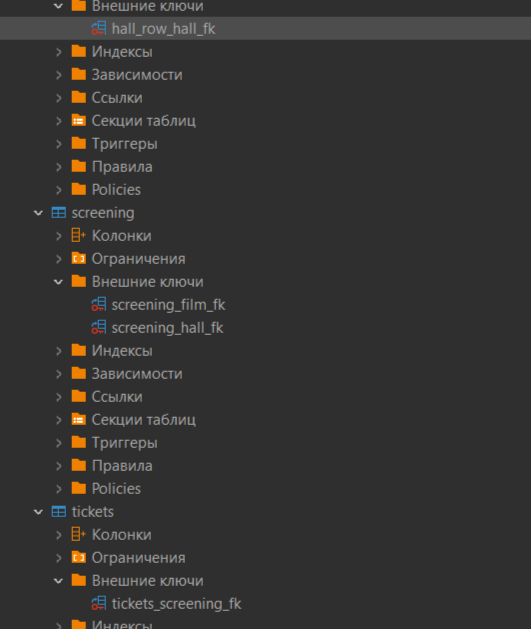




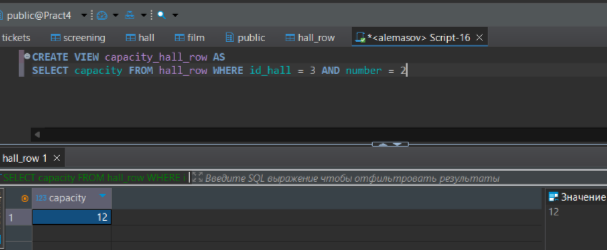




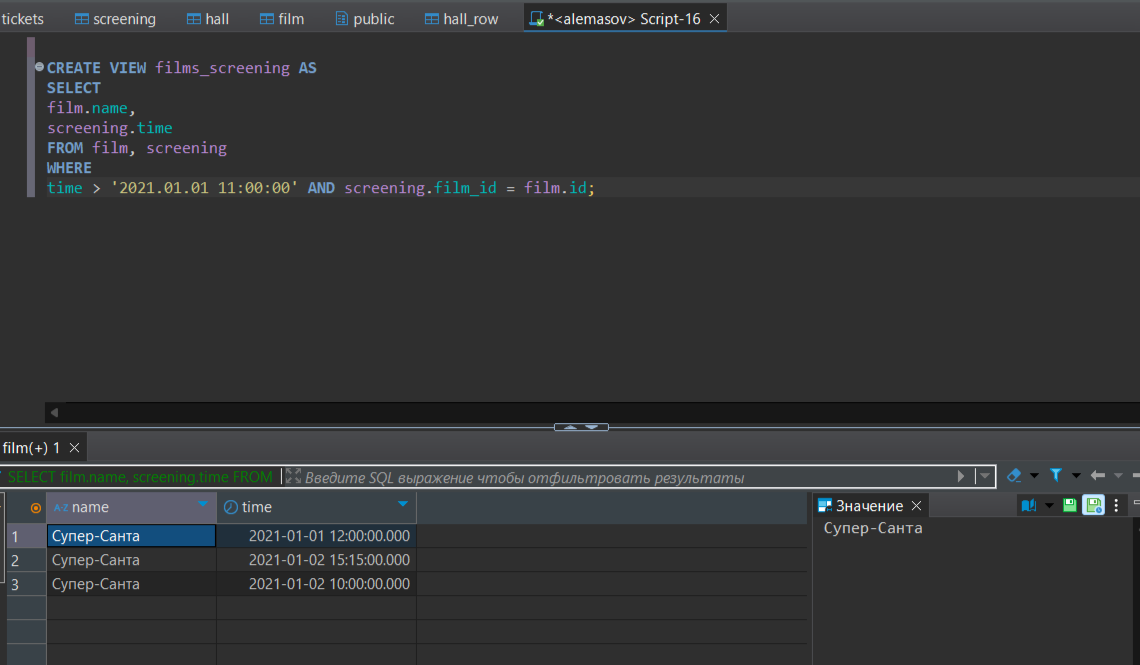
Внешние ключи



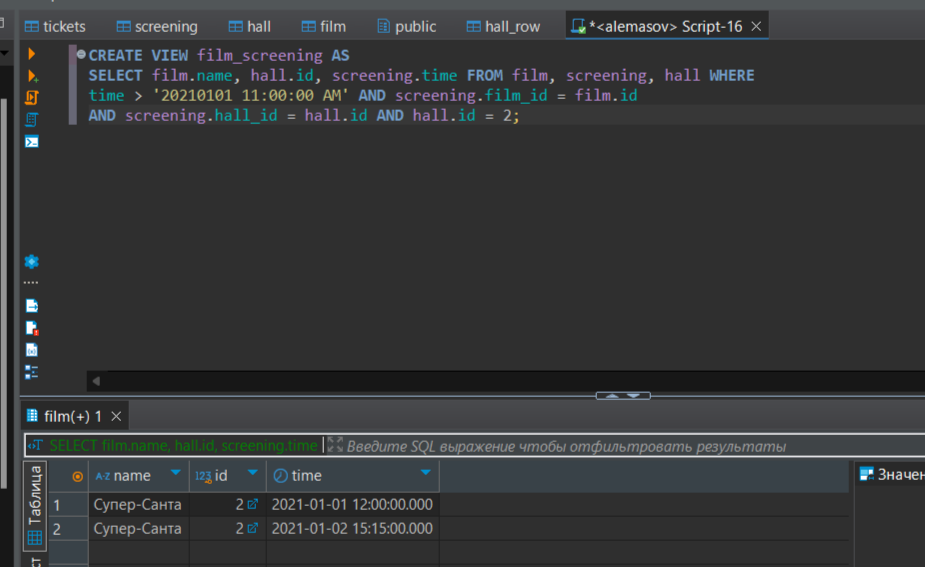
1) Создайте представление, чтобы постоянно знать количество мест во втором ряду третьего зала.



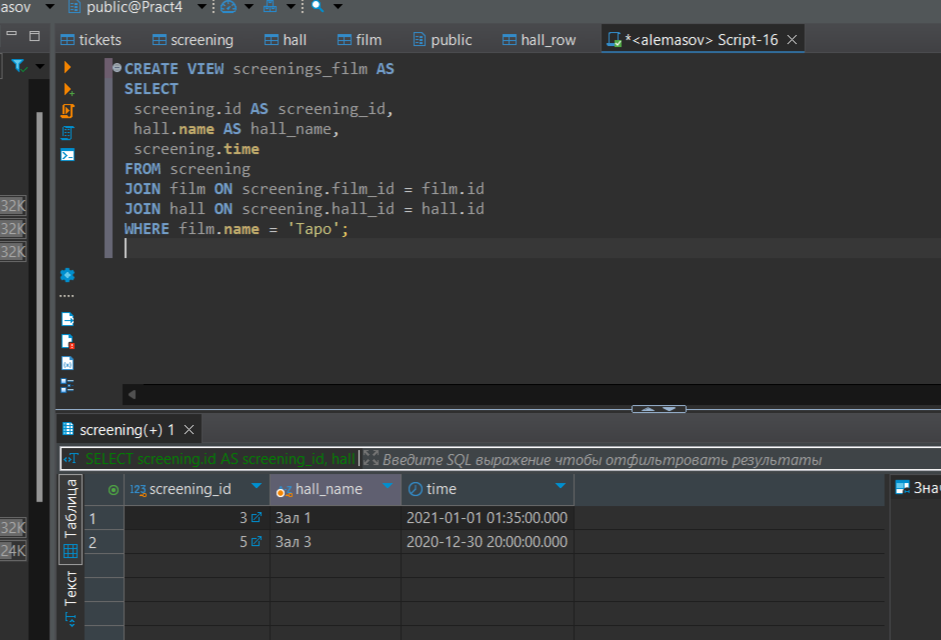
2) Создайте представление, чтобы автоматически получать названия фильмов, показ которых начинается после 11 часов 01.01.2021.



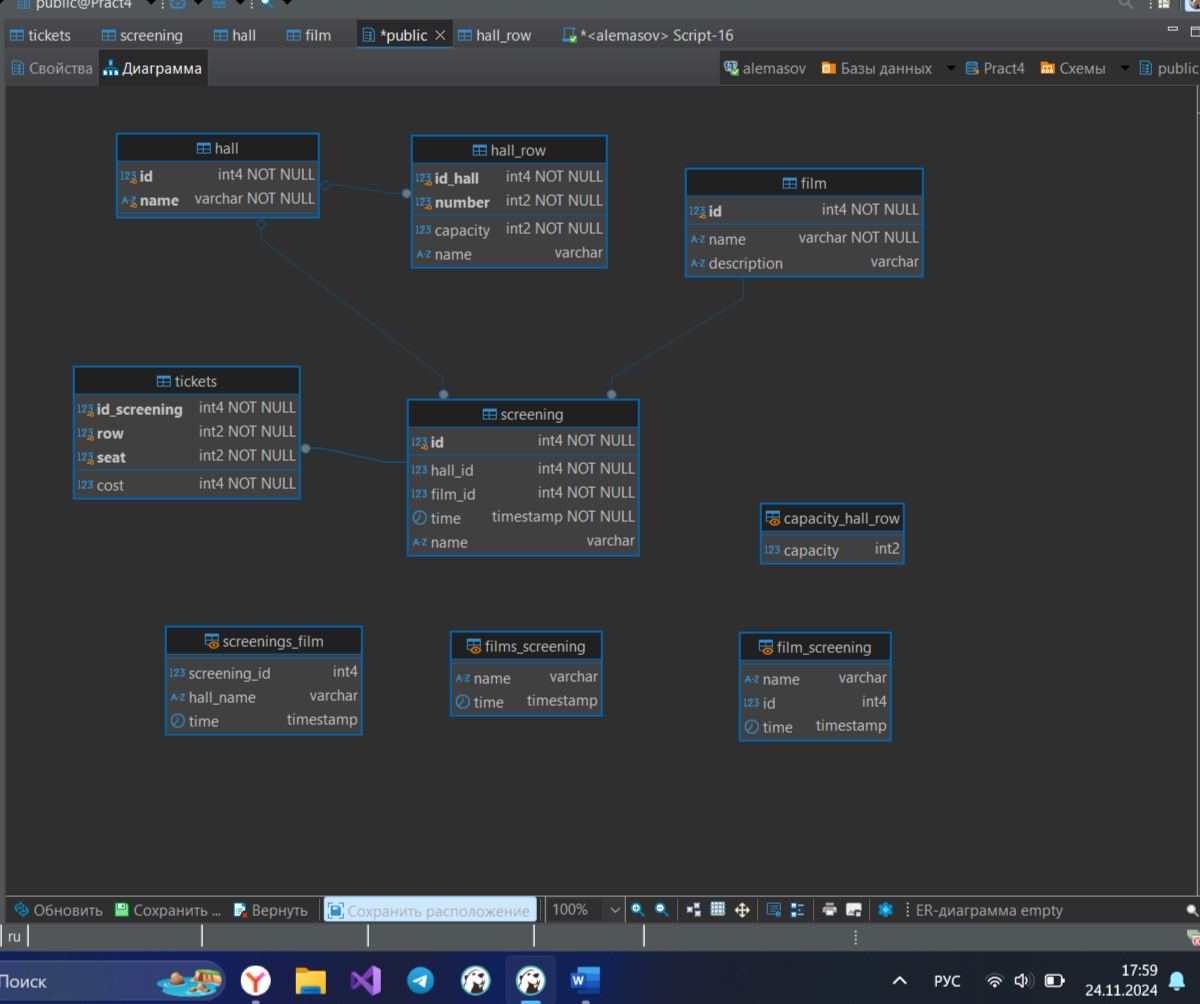
3)Создайте представление для получения расписания проката в конкретном зале кинотеатра.

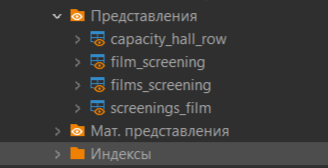


4)Создайте представление для получения расписания проката конкретного фильма.



Представления:





-- public.film определение

-- Drop table

-- DROP TABLE film;

**CREATE** **TABLE** film (

id **int4** **NOT** **NULL**,

"name" **varchar** **NOT** **NULL**,

description **varchar** **NULL**,

**CONSTRAINT** film\_pk **PRIMARY** **KEY** (id)

);

-- Permissions

**ALTER** **TABLE** film **OWNER** **TO** postgres;

**GRANT** **ALL** **ON** **TABLE** film **TO** postgres;

-- public.hall определение

-- Drop table

-- DROP TABLE hall;

**CREATE** **TABLE** hall (

id **int4** **NOT** **NULL**,

"name" **varchar** **NOT** **NULL**,

**CONSTRAINT** hall\_pk\_1 **PRIMARY** **KEY** (name, id)

);

-- Permissions

**ALTER** **TABLE** hall **OWNER** **TO** postgres;

**GRANT** **ALL** **ON** **TABLE** hall **TO** postgres;

-- public.hall\_row определение

-- Drop table

-- DROP TABLE hall\_row;

**CREATE** **TABLE** hall\_row (

id\_hall **int4** **NOT** **NULL**,

"number" **int2** **NOT** **NULL**,

capacity **int2** **NOT** **NULL**,

"name" **varchar** **NULL**,

**CONSTRAINT** hall\_row\_pk **PRIMARY** **KEY** (id\_hall, number),

**CONSTRAINT** hall\_row\_hall\_fk **FOREIGN** **KEY** ("name",id\_hall) **REFERENCES** hall("name",id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

-- Permissions

**ALTER** **TABLE** hall\_row **OWNER** **TO** postgres;

**GRANT** **ALL** **ON** **TABLE** hall\_row **TO** postgres;

-- public.screening определение

-- Drop table

-- DROP TABLE screening;

**CREATE** **TABLE** screening (

id **int4** **NOT** **NULL**,

hall\_id **int4** **NOT** **NULL**,

film\_id **int4** **NOT** **NULL**,

"time" **timestamp** **NOT** **NULL**,

"name" **varchar** **NULL**,

**CONSTRAINT** screening\_pk **PRIMARY** **KEY** (id),

**CONSTRAINT** screening\_film\_fk **FOREIGN** **KEY** (film\_id) **REFERENCES** film(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**CONSTRAINT** screening\_hall\_fk **FOREIGN** **KEY** ("name",hall\_id) **REFERENCES** hall("name",id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

-- Permissions

**ALTER** **TABLE** screening **OWNER** **TO** postgres;

**GRANT** **ALL** **ON** **TABLE** screening **TO** postgres;

-- public.tickets определение

-- Drop table

-- DROP TABLE tickets;

**CREATE** **TABLE** tickets (

id\_screening **int4** **NOT** **NULL**,

"row" **int2** **NOT** **NULL**,

seat **int2** **NOT** **NULL**,

"cost" **int4** **NOT** **NULL**,

**CONSTRAINT** tickets\_pk **PRIMARY** **KEY** (id\_screening, "row", seat),

**CONSTRAINT** tickets\_screening\_fk **FOREIGN** **KEY** (id\_screening) **REFERENCES** screening(id) **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

-- Permissions

**ALTER** **TABLE** tickets **OWNER** **TO** postgres;

**GRANT** **ALL** **ON** **TABLE** tickets **TO** postgres;

**INSERT** **INTO** public.film (id,"name",description) **VALUES**

(1,'Огниво','Приключение'),

(2,'Таро','Ужасы'),

(3,'Супер-Санта','Мультфильм');

**INSERT** **INTO** public.hall (id,"name") **VALUES**

(1,'Зал 1'),

(2,'Зал 2'),

(3,'Зал 3');

**INSERT** **INTO** public.hall\_row (id\_hall,"number",capacity,"name") **VALUES**

(1,1,10,**NULL**),

(1,2,15,**NULL**),

(1,3,20,**NULL**),

(2,2,15,**NULL**),

(3,2,12,**NULL**);

**INSERT** **INTO** public.screening (id,hall\_id,film\_id,"time","name") **VALUES**

(1,1,1,'2021-01-01 00:00:00',**NULL**),

(2,1,1,'2021-01-01 10:35:00',**NULL**),

(3,1,2,'2021-01-01 01:35:00',**NULL**),

(5,3,2,'2020-12-30 20:00:00',**NULL**),

(7,2,3,'2021-01-01 12:00:00',**NULL**),

(4,2,3,'2021-01-02 15:15:00',**NULL**),

(6,3,3,'2021-01-02 10:00:00',**NULL**);

**INSERT** **INTO** public.tickets (id\_screening,"row",seat,"cost") **VALUES**

(2,4,15,100),

(3,5,15,100),

(4,6,15,100),

(5,7,15,100),

(1,2,10,150),

(1,3,15,200),

(1,3,6,150);

**CREATE** **OR** **REPLACE** VIEW public.capacity\_hall\_row

**AS** **SELECT** capacity

**FROM** hall\_row

**WHERE** id\_hall = 3 **AND** **number** = 2;

**CREATE** **OR** **REPLACE** VIEW public.film\_screening

**AS** **SELECT** film.**name**,

hall.id,

screening.**"time"**

**FROM** film,

screening,

hall

**WHERE** screening.**"time"** > '2021-01-01 11:00:00'::**timestamp** **without** **time** **zone** **AND** screening.film\_id = film.id **AND** screening.hall\_id = hall.id **AND** hall.id = 2;

**CREATE** **OR** **REPLACE** VIEW public.films\_screening

**AS** **SELECT** film.**name**,

screening.**"time"**

**FROM** film,

screening

**WHERE** screening.**"time"** > '2021-01-01 11:00:00'::**timestamp** **without** **time** **zone** **AND** screening.film\_id = film.id;

**CREATE** **OR** **REPLACE** VIEW public.screenings\_film

**AS** **SELECT** screening.id **AS** screening\_id,

hall.**name** **AS** hall\_name,

screening.**"time"**

**FROM** screening

**JOIN** film **ON** screening.film\_id = film.id

**JOIN** hall **ON** screening.hall\_id = hall.id

**WHERE** film.**name**::**text** = 'Таро'::**text**;