

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: Чернов Алексей Александрович

Группа: 227

Преподаватель: Алемасов Евгений Павлович

Дата сдачи 03.10.2024

Оценка

Подпись преподавателя _____

ЗЕЛЕНОДОЛЬСК – 2024

```
--create database PR_5_Chernov_227;

create table Manager(
Manager_id smallserial primary key,
Manager_Login varchar(30) not null unique,
Manager_Password varchar(30) not null check(length(Manager_Password)>=6 and
length(Manager_Password)<=30)
);

create table Film(
Film_id smallserial primary key,
Film_name varchar(50) not null unique,
Film_discription text
);

create table Hall(
Hall_id smallserial primary key,
Hall_name varchar(30) not null unique
);

create table Screening(
Screening_id serial primary key,
Hall_id smallint not null,
Film_id smallint not null,
Manager_id smallint not null,
Screening_date timestamp not null,
foreign key (Hall_id) references Hall(Hall_id) on delete cascade on update
cascade,
foreign key (Film_id) references Film(Film_id) on delete cascade on update
cascade,
foreign key (Manager_id) references Manager(Manager_id) on delete cascade on
update cascade
);

create table Hall_row(
Hall_row_number smallint primary key,
Hall_row_capacity smallint not null,
Hall_id smallint not null,
foreign key (Hall_id) references Hall(Hall_id) on delete cascade on update
cascade
);

create table Casies(
Casies_number smallint primary key,
Casies_login varchar(30) not null unique,
```

```

Casies_password varchar(30) not null
);

create table Ticket(
Ticket_id bigserial primary key,
Hall_row_number smallint not null,
Screening_id int not null,
Ticket_seat smallint not null unique,
Ticket_cost money not null,
foreign key (Hall_row_number) references Hall_row(Hall_row_number) on delete
cascade on update cascade,
foreign key (Screening_id) references Screening(Screening_id) on delete
cascade on update cascade
);

alter table ticket add column Casies_number bigint not null;
alter table ticket add foreign key (Casies_number) references
Casies(Casies_number);

insert into Hall values (1, '1-ый зал'), (2, '2-ой зал'), (3, '3-ий зал');
insert into Hall_row values (1, 30, 3), (2, 25, 3), (3, 30, 3);

create view hall_hallrow as
select Hall_row_capacity as "Количество мест", Hall_name as "Зал" from
Hall_row
join Hall on Hall.Hall_id = Hall_row.Hall_id where Hall.Hall_id = 3 and
Hall_row_number = 2;

insert into Film values (default, 'DeadPool', ''), (default, 'DeadPool 2',
''), (default, 'DeadPool 3', '');
insert into Manager values (default, 'Kaprutz3003@yandex.ru', '1234567');
insert into Screening values (default, 3, 1, 1, '2024-11-19 11:05'), (default,
3, 2, 1, '2024-11-19 10:05'), (default, 3, 3, 1, '2024-11-19 11:00');

create view Film_Screening as
select Film_name as "Название", Screening_date as "Дата сеанса" from Screening
join Film on Film.Film_id = Screening.Film_id where Screening_date >= '2024-
01-01 11:00'/*extract(hour from Screening_date) >= 11*/;

create view Hall_Screening as
select Screening_date as "Дата начала проката", Hall_name as "Название зала"
from Screening
join Hall on Hall.Hall_id = Screening.Hall_id where Hall.Hall_id = 3;

create view Screening_Film as

```

```

select Film_name as "Название фильма", Screening_date as "Дата начала проката"
from Screening
join Film on Film.Film_id = Screening.Film_id where Film.Film_id = 2;

/*create materialized view hall_hall_row_mat as
select hall.hall_id, hall_name, hall_row_number from hall_row
join hall on hall.hall_id = hall_row.hall_id;

create materialized view all_mat as
select hall_name as "Зал", hall_row_number as "Ряд", film_name as "Название
фильма", screening_date as "Дата начала" from screening
join hall_hall_row_mat on hall_hall_row_mat.hall_id = screening.hall_id
join film on film.film_id = screening.film_id;*/

--alter view all_ rename column "Дата начала" to "Дата_начала";
--alter view all_ rename column "Название фильма" to "Название_фильма";

--drop function getfilm(hall_name varchar(50), screening_date timestamp,
film_name varchar(50));

/*create or replace function getfilm(hall_name varchar(50), screening_date
timestamp, film_name varchar(50))
returns setof all_mat as $$
begin
    execute 'REFRESH MATERIALIZED VIEW hall_hall_row_mat';
    execute 'REFRESH MATERIALIZED VIEW all_mat';

    return query select * from all_
    where Зал = hall_name
        and Дата_начала >= screening_date
        and Название_фильма = film_name;
end;
$$ language plpgsql;

select * from getfilm('3-ий зал', '2024-11-19 11:00', 'DeadPool');*/

```