Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1 по дисциплине «База данных»

Реализация SQL-запросов для создания схемы базы данных

Студента-заочника 4 курса

группы №250541

Власова Р.Е.

Проверила: магистр технических наук

Куприянова Д. В.

МИНСК 2024

Реализация SQL-запросов для создания схемы базы данных

Цель работы: выполнить реализацию схемы базы данных по ранее

построенной реляционной схеме данных, сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД.

Выполнение работы

1. Создаем в СУБД (PostgreSql) новую схему данных для хранения пользовательских объектов.
2. В этой новой схеме данных с помощью запросов на языке DDL SQL реализуем таблицы, соответствующие реляционным отношениям схемы данных полученной в контрольной работе с помощью оператора CREATE TABLE.
3. Заполняем созданные таблицы строками данных для проверки правильного выбора первичных ключей и работоспособности ссылок между таблицами.

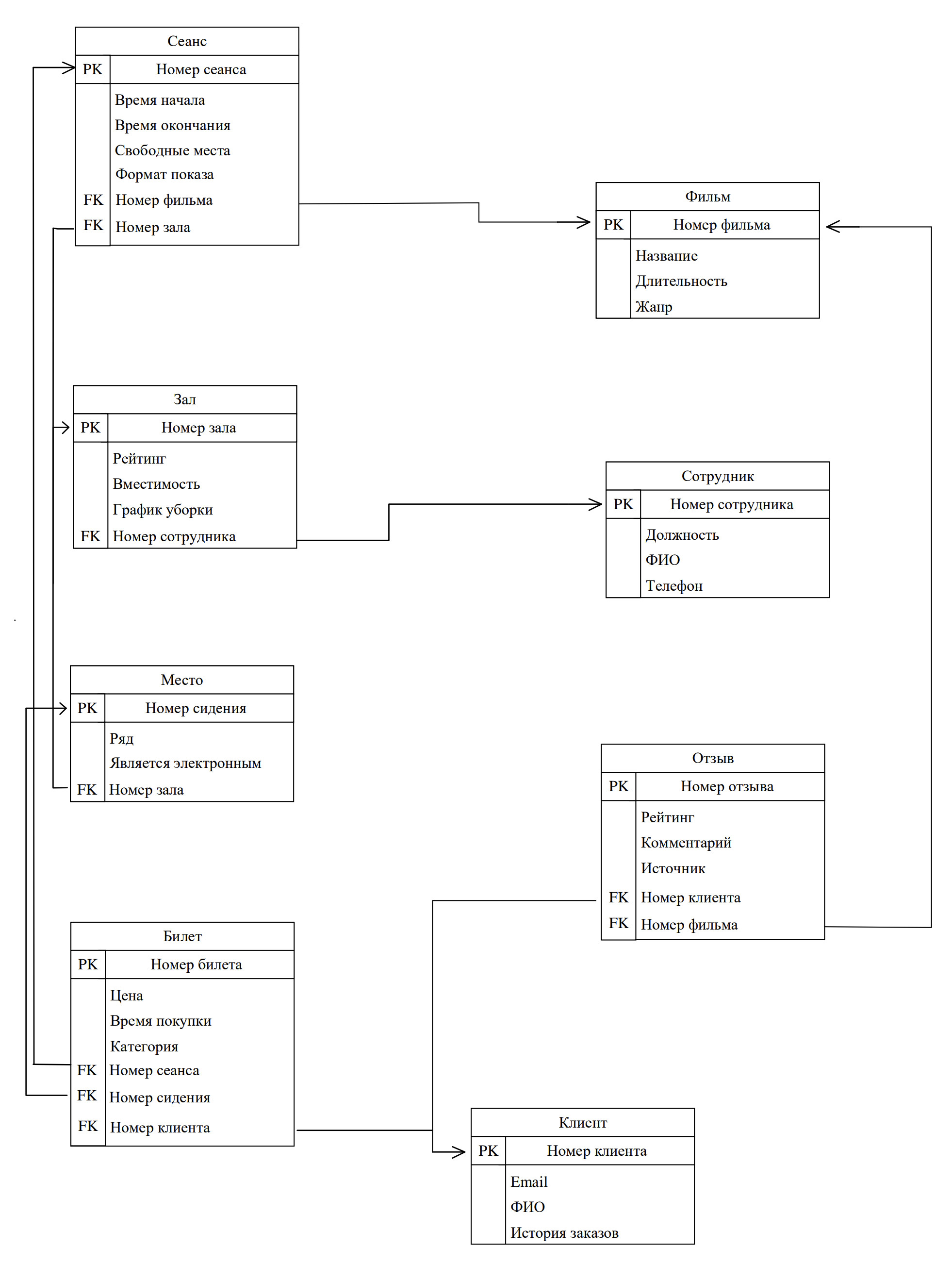


Рисунок 1. UML-диаграмма « Кинотеатр»

1. Создание схемы данных «Кинотеатр»:

CREATE SCHEMA IF NOT EXISTS Кинотеатр;

SHOW search\_path;

SET search\_path TO Кинотеатр;

1. Создание таблицы «Сотрудник»:

CREATE TABLE Сотрудник (  
 Номер\_сотрудника INT PRIMARY KEY,

Должность VARCHAR(50) NOT NULL,

ФИО VARCHAR(255) NOT NULL,

Телефон VARCHAR(50) NOT NULL  
 );

1. Создание таблицы «Фильм»:

CREATE TABLE Фильм (  
 Номер\_фильма INT PRIMARY KEY,

Название VARCHAR(255) NOT NULL,

Длительность TIME NOT NULL,

Жанр VARCHAR(50) NOT NULL  
 );

CREATE INDEX idx\_фильм\_название ON Фильм (Название);

1. Создание таблицы «Зал»:

CREATE TABLE Зал (

Номер\_зала INT PRIMARY KEY,

Рейтинг INT NOT NULL,

Вместимость INT NOT NULL,

График\_уборки VARCHAR(255) NOT NULL,

Номер\_сотрудника INT,

FOREIGN KEY (Номер\_сотрудника) REFERENCES Сотрудник(Номер\_сотрудника) ON DELETE SET NULL

);

CREATE INDEX idx\_зал\_сотрудника ON Зал (Номер\_сотрудника);

1. Создание таблицы «Клиент»:

CREATE TABLE Клиент (  
 Номер\_клиента INT PRIMARY KEY,

Email VARCHAR(255) NOT NULL UNIQUE,

ФИО VARCHAR(255) NOT NULL,

История\_заказов TEXT NOT NULL  
);

CREATE INDEX idx\_клиент\_email ON Клиент (Email);

1. Создание таблицы «Сеанс»:

CREATE TABLE Сеанс (  
 Номер\_сеанса INT PRIMARY KEY,

Номер\_фильма INT NOT NULL,

Номер\_зала INT NOT NULL,

Время\_начала TIME NOT NULL,

Время\_окончания TIME NOT NULL,

Формат\_показа VARCHAR(50) NOT NULL,

Свободные\_места INT NOT NULL,

FOREIGN KEY (Номер\_фильма) REFERENCES Фильм(Номер\_фильма) ON DELETE CASCADE,

FOREIGN KEY (Номер\_зала) REFERENCES Зал(Номер\_зала) ON DELETE CASCADE  
 );

CREATE INDEX idx\_сеанс\_фильм ON Сеанс (Номер\_фильма);

CREATE INDEX idx\_сеанс\_зал ON Сеанс (Номер\_зала);

1. Создание таблицы «Место»:

CREATE TABLE Место (  
 Номер\_сидения INT PRIMARY KEY,

Номер\_зала INT NOT NULL,

Ряд VARCHAR(10),

Номер INT,

Является\_электронным BOOLEAN NOT NULL,

FOREIGN KEY (Номер\_зала) REFERENCES Зал(Номер\_зала) ON DELETE CASCADE);

CREATE INDEX карта\_id ON Запись\_к\_врачу (Номер\_мед\_карты);

8) Создание таблицы «Билет»:

CREATE TABLE Билет (

Номер\_билета INT PRIMARY KEY,

Номер\_сеанса INT NOT NULL,

Номер\_сидения INT NOT NULL,

Номер\_клиента INT NOT NULL,

Цена DECIMAL(10,2) NOT NULL,

Время\_покупки TIMESTAMP NOT NULL,

Категория VARCHAR(50) NOT NULL,

FOREIGN KEY (Номер\_сеанса) REFERENCES Сеанс(Номер\_сеанса) ON DELETE CASCADE,

FOREIGN KEY (Номер\_сидения) REFERENCES Место(Номер\_сидения) ON DELETE CASCADE,

FOREIGN KEY (Номер\_клиента) REFERENCES Клиент(Номер\_клиента) ON DELETE CASCADE

);

CREATE INDEX idx\_билет\_сеанс ON Билет (Номер\_сеанса);

CREATE INDEX idx\_билет\_сидение ON Билет (Номер\_сидения);

CREATE INDEX idx\_билет\_клиент ON Билет (Номер\_клиента);

9) Создание таблицы «Отзыв»:

CREATE TABLE Отзыв (

Номер\_отзыва INT PRIMARY KEY,

Номер\_клиента INT NOT NULL,

Номер\_фильма INT NOT NULL,

Рейтинг FLOAT NOT NULL,

Комментарий VARCHAR(255),

Источник VARCHAR(50),

FOREIGN KEY (Номер\_клиента) REFERENCES Клиент(Номер\_клиента) ON DELETE CASCADE,

FOREIGN KEY (Номер\_фильма) REFERENCES Фильм(Номер\_фильма) ON DELETE CASCADE

);

CREATE INDEX idx\_отзыв\_клиент ON Отзыв (Номер\_клиента);

CREATE INDEX idx\_отзыв\_фильм ON Отзыв (Номер\_фильма);

Заполнение данными таблицы «Сотрудник»:  
  
INSERT INTO Сотрудник (Номер\_сотрудника, Должность, ФИО, Телефон)

VALUES (1, 'Менеджер', 'Иванов Иван Иванович', '+375-29-345-67-89'),

(2, 'Администратор', 'Петрова Мария Сергеевна', '+7-913-456-78-90'),

(3, 'Техник', 'Сидоров Алексей Петрович', '+375-29-567-89-01'),

(4, 'Уборщик', 'Кузнецова Елена Владимировна', '+375-29-678-90-12'),

(5, 'Бухгалтер', 'Морозова Ольга Николаевна', '+375-29-789-01-23'),

(6, 'Менеджер', 'Федоров Дмитрий Александрович', '+375-29-890-12-34'),

(7, 'Администратор', 'Васильев Сергей Михайлович', '+375-29-901-23-45'),

(8, 'Техник', 'Николаева Анна Викторовна', '+375-29-012-34-56'),

(9, 'Уборщик', 'Лебедев Павел Юрьевич', '+375-29-123-45-67'),

(10, 'Бухгалтер', 'Григорьева Светлана Ивановна', '+375-29-234-56-78'),

(11, 'Менеджер', 'Новиков Сергей Иванович', '+375-29-345-67-89'),

(12, 'Администратор', 'Соколова Ольга Петровна', '+375-29-456-78-90'),

(13, 'Техник', 'Кузнецов Николай Александрович', '+375-29-567-89-01'),

(14, 'Уборщик', 'Иванова Наталья Сергеевна', '+375-29-678-90-12'),

(15, 'Бухгалтер', 'Смирнов Павел Викторович', '+375-29-789-01-23'),

(16, 'Менеджер', 'Козлова Марина Дмитриевна', '+375-29-890-12-34'),

(17, 'Администратор', 'Васильев Андрей Николаевич', '+375-29-901-23-45'),

(18, 'Техник', 'Лебедева Елизавета Петровна', '+375-29-012-34-56'),

(19, 'Уборщик', 'Федоров Максим Иванович', '+375-29-123-45-67'),

(20, 'Бухгалтер', 'Григорьев Алексей Сергеевич', '+375-29-234-56-78');

Заполнение данными таблицы «Фильм»:

INSERT INTO Фильм (Номер\_фильма, Название, Длительность, Жанр) VALUES

(1, 'Титаник', '03:15:00', 'Драма'),

(2, 'Властелин колец: Братство кольца', '02:58:00', 'Фэнтези'),

(3, 'Интерстеллар', '02:49:00', 'Научная фантастика'),

(4, 'Начало', '02:28:00', 'Экшн'),

(5, 'Матрица', '02:16:00', 'Боевик'),

(6, 'Побег из Шоушенка', '02:22:00', 'Драма'),

(7, 'Темный рыцарь', '02:32:00', 'Боевик'),

(8, 'Форрест Гамп', '02:22:00', 'Драма'),

(9, 'Парк Юрского периода', '02:05:00', 'Приключения'),

(10, 'Криминальное чтиво', '02:34:00', 'Криминал'),

(11, 'Гарри Поттер и философский камень', '02:32:00', 'Фэнтези'),

(12, 'Аватар', '02:42:00', 'Научная фантастика'),

(13, 'Бойцовский клуб', '02:19:00', 'Драма'),

(14, 'Зеленая миля', '03:09:00', 'Фэнтези'),

(15, 'Список Шиндлера', '03:15:00', 'Историческая драма'),

(16, 'Король Лев', '01:58:00', 'Мультфильм'),

(17, 'Железный человек', '02:06:00', 'Боевик'),

(18, 'Форсаж', '02:10:00', 'Экшн'),

(19, 'Пираты Карибского моря: Проклятие Черной жемчужины', '02:23:00', 'Приключения'),

(20, 'Индиана Джонс: В поисках утраченного ковчега', '01:58:00', 'Приключения');

Заполнение данными таблицы «Зал»:  
  
INSERT INTO Зал (Номер\_зала, Рейтинг, Вместимость, График\_уборки, Номер\_сотрудника) VALUES

(1, 4, 100, 'Понедельник, Среда, Пятница', 4),

(2, 6, 80, 'Вторник, Четверг, Суббота', 9),

(3, 10, 150, 'Ежедневно', 4),

(4, 8, 120, 'Понедельник, Вторник, Среда', 10),

(5, 9, 90, 'Четверг, Пятница, Воскресенье', 7);

(6, 110, 'Вторник, Четверг, Суббота', 9),

(7, 95, 'Понедельник, Среда, Пятница', 4),

(8, 85, 'Четверг, Пятница, Воскресенье', 7),

(9, 130, 'Понедельник, Вторник, Среда', 10),

(10, 75, 'Вторник, Четверг, Суббота', 12),

(11, 140, 'Ежедневно', 13),

(12, 100, 'Понедельник, Среда, Пятница', 14),

(13, 120, 'Четверг, Пятница, Воскресенье', 15),

(14, 90, 'Вторник, Четверг, Суббота', 11),

(15, 105, 'Понедельник, Вторник, Среда', 12);

Заполнение данными таблицы «Клиент»:  
INSERT INTO Клиент (Номер\_клиента, Email, ФИО, История\_заказов) VALUES

(1, 'alex@gmail.com', 'Алексей Смирнов', 'Властелин колец: Братство кольца, Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода'),

(2, 'maria@gmail.com', 'Мария Иванова', 'Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво'),

(3, 'sergey@gmail.com', 'Сергей Ковалев', 'Титаник, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво'),

(4, 'olga@gmail.com', 'Ольга Петрова', 'Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Форрест Гамп, Парк Юрского периода, Криминальное чтиво'),

(5, 'dmitry@gmail.com', 'Дмитрий Васильев', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво'),

(6, 'anna@gmail.com', 'Анна Николаева', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода'),

(7, 'pavel@gmail.com', 'Павел Лебедев', 'Титаник, Интерстеллар, Начало, Матрица, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво'),

(8, 'svetlana@gmail.com', 'Светлана Григорьева', 'Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Форрест Гамп, Парк Юрского периода, Криминальное чтиво'),

(9, 'elena@gmail.com', 'Елена Морозова', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво'),

(10, 'ivan@gmail.com', 'Иван Федоров', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво'),

(11, 'sergey\_novikov@gmail.com', 'Сергей Новиков', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво'),

(12, 'olga\_sokolova@gmail.com', 'Ольга Соколова', 'Властелин колец: Братство кольца, Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода'),

(13, 'nikolai\_kuznetsov@gmail.com', 'Николай Кузнецов', 'Титаник, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво, Интерстеллар'),

(14, 'natalia\_ivanova@gmail.com', 'Наталья Иванова', 'Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, IMAX'),

(15, 'pavel\_smirnov@gmail.com', 'Павел Смирнов', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво'),

(16, 'marina\_kozlova@gmail.com', 'Марина Козлова', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода'),

(17, 'andrey\_vasiliev@gmail.com', 'Андрей Васильев', 'Титаник, Интерстеллар, Начало, Матрица, Темный рыцарь, Форрест Гамп, Криминальное чтиво, Властелин колец: Братство кольца'),

(18, 'elizaveta\_lebedeva@gmail.com', 'Елизавета Лебедева', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, IMAX'),

(19, 'maksim\_fedorov@gmail.com', 'Максим Федоров', 'Титаник, Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво'),

(20, 'aleksey\_grigorev@gmail.com', 'Алексей Григорьев', 'Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода');

Заполнение данными таблицы «Сеанс»:

INSERT INTO Сеанс (Номер\_сеанса, Номер\_фильма, Номер\_зала, Время\_начала, Время\_окончания, Формат\_показа, Свободные\_места) VALUES

1. (1, 1, 1, '10:00:00', '13:15:00', '2D', 100),
2. (2, 2, 2, '12:00:00', '14:58:00', '3D', 80),
3. (3, 3, 3, '14:00:00', '16:49:00', 'IMAX', 150),
4. (4, 4, 4, '16:00:00', '18:28:00', '2D', 120),
5. (5, 5, 5, '18:00:00', '20:16:00', '3D', 90),
6. (6, 6, 1, '20:00:00', '22:22:00', '2D', 100),
7. (7, 7, 2, '22:30:00', '00:02:00', '3D', 80),
8. (8, 8, 3, '09:30:00', '11:52:00', '2D', 150),
9. (9, 9, 4, '11:00:00', '13:05:00', 'IMAX', 120),
10. (10, 10, 5, '13:00:00', '15:34:00', '2D', 90),
11. (11, 1, 2, '15:00:00', '18:15:00', '3D', 80),
12. (12, 3, 1, '17:00:00', '19:49:00', 'IMAX', 150),
13. (13, 5, 3, '19:00:00', '21:16:00', '3D', 90),
14. (14, 7, 4, '21:00:00', '23:32:00', '2D', 120),
15. (15, 9, 5, '23:00:00', '01:05:00', 'IMAX', 90);

Заполнение данными таблицы «Место»:  
  
INSERT INTO Место (Номер\_сидения, Номер\_зала, Ряд, Номер, Является\_электронным) VALUES

-- Зал 1

(1, 1, 'A', 1, TRUE),

(2, 1, 'A', 2, TRUE),

(3, 1, 'A', 3, TRUE),

(4, 1, 'A', 4, TRUE),

(5, 1, 'A', 5, TRUE),

(6, 1, 'A', 6, TRUE),

(7, 1, 'A', 7, TRUE),

(8, 1, 'A', 8, TRUE),

(9, 1, 'A', 9, TRUE),

(10, 1, 'A', 10, TRUE),

-- Зал 2

(11, 2, 'B', 1, TRUE),

(12, 2, 'B', 2, TRUE),

(13, 2, 'B', 3, TRUE),

(14, 2, 'B', 4, TRUE),

(15, 2, 'B', 5, TRUE),

(16, 2, 'B', 6, TRUE),

(17, 2, 'B', 7, TRUE),

(18, 2, 'B', 8, TRUE),

(19, 2, 'B', 9, TRUE),

(20, 2, 'B', 10, TRUE),

-- Зал 3

(21, 3, 'C', 1, TRUE),

(22, 3, 'C', 2, TRUE),

(23, 3, 'C', 3, TRUE),

(24, 3, 'C', 4, TRUE),

(25, 3, 'C', 5, TRUE),

(26, 3, 'C', 6, TRUE),

(27, 3, 'C', 7, TRUE),

(28, 3, 'C', 8, TRUE),

(29, 3, 'C', 9, TRUE),

(30, 3, 'C', 10, TRUE),

-- Зал 4

(31, 4, 'D', 1, TRUE),

(32, 4, 'D', 2, TRUE),

(33, 4, 'D', 3, TRUE),

(34, 4, 'D', 4, TRUE),

(35, 4, 'D', 5, TRUE),

(36, 4, 'D', 6, TRUE),

(37, 4, 'D', 7, TRUE),

(38, 4, 'D', 8, TRUE),

(39, 4, 'D', 9, TRUE),

(40, 4, 'D', 10, TRUE),

-- Зал 5

(41, 5, 'E', 1, TRUE),

(42, 5, 'E', 2, TRUE),

(43, 5, 'E', 3, TRUE),

(44, 5, 'E', 4, TRUE),

(45, 5, 'E', 5, TRUE),

(46, 5, 'E', 6, TRUE),

(47, 5, 'E', 7, TRUE),

(48, 5, 'E', 8, TRUE),

(49, 5, 'E', 9, TRUE),

(50, 5, 'E', 10, TRUE);

Заполнение данными таблицы «Билет»:

INSERT INTO Билет (Номер\_билета, Номер\_сеанса, Номер\_сидения, Номер\_клиента, Цена, Время\_покупки, Категория) VALUES

(1, 1, 1, 1, 500.00, '2024-04-01 10:05:00', 'Стандарт'),

(2, 1, 2, 2, 500.00, '2024-04-01 10:10:00', 'Стандарт'),

(3, 2, 11, 3, 600.00, '2024-04-01 12:15:00', '3D'),

(4, 3, 21, 4, 700.00, '2024-04-01 14:20:00', 'IMAX'),

(5, 4, 31, 5, 550.00, '2024-04-01 16:25:00', 'Стандарт'),

(6, 5, 41, 6, 600.00, '2024-04-01 18:30:00', '3D'),

(7, 6, 3, 7, 500.00, '2024-04-01 20:35:00', 'Стандарт'),

(8, 7, 12, 8, 600.00, '2024-04-01 22:40:00', '3D'),

(9, 8, 22, 9, 700.00, '2024-04-02 09:35:00', 'IMAX'),

(10, 9, 32, 10, 550.00, '2024-04-02 11:40:00', 'Стандарт'),

(11, 10, 42, 1, 600.00, '2024-04-02 13:45:00', '3D'),

(12, 11, 4, 2, 500.00, '2024-04-02 15:50:00', 'Стандарт'),

(13, 12, 23, 3, 700.00, '2024-04-02 17:55:00', 'IMAX'),

(14, 13, 33, 4, 600.00, '2024-04-02 19:00:00', '3D'),

(15, 14, 43, 5, 550.00, '2024-04-02 21:05:00', 'Стандарт'),

(16, 15, 50, 6, 600.00, '2024-04-02 23:10:00', 'IMAX'),

(17, 2, 13, 7, 600.00, '2024-04-03 12:20:00', '3D'),

(18, 3, 24, 8, 700.00, '2024-04-03 14:25:00', 'IMAX'),

(19, 4, 34, 9, 550.00, '2024-04-03 16:30:00', 'Стандарт'),

(20, 5, 44, 10, 600.00, '2024-04-03 18:35:00', '3D');

Заполнение данными таблицы «Отзыв»:

INSERT INTO Отзыв (Номер\_отзыва, Номер\_клиента, Номер\_фильма, Рейтинг, Комментарий, Источник) VALUES

(1, 1, 1, 4.5, 'Отличный фильм, трогательная история.', 'Google'),

(2, 2, 2, 5.0, 'Великолепное фэнтези с потрясающими спецэффектами.', 'Yandex'),

(3, 3, 3, 4.8, 'Увлекательная научная фантастика.', 'IMDb'),

(4, 4, 4, 4.2, 'Интересный сюжет и отличная игра актеров.', 'Google'),

(5, 5, 5, 4.7, 'Классический боевик, рекомендую.', 'Facebook'),

(6, 6, 6, 5.0, 'Вдохновляющая драма с глубоким смыслом.', 'Yandex'),

(7, 7, 7, 4.9, 'Отличный фильм о супергероях.', 'IMDb'),

(8, 8, 8, 4.6, 'Замечательная история с трогательными моментами.', 'Google'),

(9, 9, 9, 4.3, 'Увлекательные приключения и отличная анимация.', 'Facebook'),

(10, 10, 10, 4.8, 'Интересный криминальный сюжет с неожиданными поворотами.', 'Yandex'),

(11, 1, 3, 4.7, 'Очень понравился фильм, отличный режиссер.', 'IMDb'),

(12, 2, 4, 4.5, 'Интересная идея и реализация.', 'Google'),

(13, 3, 5, 4.4, 'Классический фильм, стоит пересмотреть.', 'Facebook'),

(14, 4, 6, 5.0, 'Вдохновляющий сюжет, рекомендую всем.', 'Yandex'),

(15, 5, 7, 4.9, 'Отличный фильм, захватывающий до конца.', 'IMDb'),

(16, 6, 8, 4.6, 'Прекрасная история, хорошо снято.', 'Google'),

(17, 7, 9, 4.3, 'Увлекательный фильм для всей семьи.', 'Facebook'),

(18, 8, 10, 4.8, 'Интересный и захватывающий сюжет.', 'Yandex'),

(19, 9, 2, 5.0, 'Лучший фильм в жанре фэнтези.', 'IMDb'),

(20, 10, 1, 4.5, 'Классика кинематографа, рекомендую посмотреть.', 'Kinopoisk');

Ниже приведен скрипт базы данных для ее восстановления:

-- PostgreSQL database dump

--

-- Dumped from database version 17.2

-- Dumped by pg\_dump version 17.2

-- Started on 2024-12-10 01:53:44 +03

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET transaction\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

--

-- TOC entry 6 (class 2615 OID 24878)

-- Name: Кинотеатр; Type: SCHEMA; Schema: -; Owner: postgres

--

CREATE SCHEMA "Кинотеатр";

ALTER SCHEMA "Кинотеатр" OWNER TO postgres;

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

--

-- TOC entry 224 (class 1259 OID 24933)

-- Name: Билет; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Билет" (

"Номер\_билета" integer NOT NULL,

"Номер\_сеанса" integer NOT NULL,

"Номер\_сидения" integer NOT NULL,

"Номер\_клиента" integer NOT NULL,

"Цена" numeric(10,2) NOT NULL,

"Время\_покупки" timestamp without time zone NOT NULL,

"Категория" character varying(50) NOT NULL

);

ALTER TABLE "Кинотеатр"."Билет" OWNER TO postgres;

--

-- TOC entry 220 (class 1259 OID 24889)

-- Name: Зал; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Зал" (

"Номер\_зала" integer NOT NULL,

"Вместимость" integer NOT NULL,

"График\_уборки" character varying(255) NOT NULL,

"Номер\_сотрудника" integer

);

ALTER TABLE "Кинотеатр"."Зал" OWNER TO postgres;

--

-- TOC entry 221 (class 1259 OID 24899)

-- Name: Клиент; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Клиент" (

"Номер\_клиента" integer NOT NULL,

email character varying(255) NOT NULL,

"ФИО" character varying(255) NOT NULL,

"История\_заказов" text NOT NULL

);

ALTER TABLE "Кинотеатр"."Клиент" OWNER TO postgres;

--

-- TOC entry 223 (class 1259 OID 24923)

-- Name: Место; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Место" (

"Номер\_сидения" integer NOT NULL,

"Номер\_зала" integer NOT NULL,

"Ряд" character varying(10),

"Номер" integer,

"Является\_электронным" boolean NOT NULL

);

ALTER TABLE "Кинотеатр"."Место" OWNER TO postgres;

--

-- TOC entry 225 (class 1259 OID 24953)

-- Name: Отзыв; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Отзыв" (

"Номер\_отзыва" integer NOT NULL,

"Номер\_клиента" integer NOT NULL,

"Номер\_фильма" integer NOT NULL,

"Рейтинг" double precision NOT NULL,

"Комментарий" character varying(255),

"Источник" character varying(50)

);

ALTER TABLE "Кинотеатр"."Отзыв" OWNER TO postgres;

--

-- TOC entry 222 (class 1259 OID 24908)

-- Name: Сеанс; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Сеанс" (

"Номер\_сеанса" integer NOT NULL,

"Номер\_фильма" integer NOT NULL,

"Номер\_зала" integer NOT NULL,

"Время\_начала" time without time zone NOT NULL,

"Время\_окончания" time without time zone NOT NULL,

"Формат\_показа" character varying(50) NOT NULL,

"Свободные\_места" integer NOT NULL

);

ALTER TABLE "Кинотеатр"."Сеанс" OWNER TO postgres;

--

-- TOC entry 218 (class 1259 OID 24879)

-- Name: Сотрудник; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Сотрудник" (

"Номер\_сотрудника" integer NOT NULL,

"Должность" character varying(50) NOT NULL,

"ФИО" character varying(255) NOT NULL,

"Телефон" character varying(50) NOT NULL

);

ALTER TABLE "Кинотеатр"."Сотрудник" OWNER TO postgres;

--

-- TOC entry 219 (class 1259 OID 24884)

-- Name: Фильм; Type: TABLE; Schema: Кинотеатр; Owner: postgres

--

CREATE TABLE "Кинотеатр"."Фильм" (

"Номер\_фильма" integer NOT NULL,

"Название" character varying(255) NOT NULL,

"Длительность" time without time zone NOT NULL,

"Жанр" character varying(50) NOT NULL

);

ALTER TABLE "Кинотеатр"."Фильм" OWNER TO postgres;

--

-- TOC entry 3667 (class 0 OID 24933)

-- Dependencies: 224

-- Data for Name: Билет; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Билет" ("Номер\_билета", "Номер\_сеанса", "Номер\_сидения", "Номер\_клиента", "Цена", "Время\_покупки", "Категория") FROM stdin;

1 1 1 1 500.00 2024-04-01 10:05:00 Стандарт

2 1 2 2 500.00 2024-04-01 10:10:00 Стандарт

3 2 11 3 600.00 2024-04-01 12:15:00 3D

4 3 21 4 700.00 2024-04-01 14:20:00 IMAX

5 4 31 5 550.00 2024-04-01 16:25:00 Стандарт

6 5 41 6 600.00 2024-04-01 18:30:00 3D

7 6 3 7 500.00 2024-04-01 20:35:00 Стандарт

8 7 12 8 600.00 2024-04-01 22:40:00 3D

9 8 22 9 700.00 2024-04-02 09:35:00 IMAX

10 9 32 10 550.00 2024-04-02 11:40:00 Стандарт

11 10 42 1 600.00 2024-04-02 13:45:00 3D

12 11 4 2 500.00 2024-04-02 15:50:00 Стандарт

13 12 23 3 700.00 2024-04-02 17:55:00 IMAX

14 13 33 4 600.00 2024-04-02 19:00:00 3D

15 14 43 5 550.00 2024-04-02 21:05:00 Стандарт

16 15 50 6 600.00 2024-04-02 23:10:00 IMAX

17 2 13 7 600.00 2024-04-03 12:20:00 3D

18 3 24 8 700.00 2024-04-03 14:25:00 IMAX

19 4 34 9 550.00 2024-04-03 16:30:00 Стандарт

20 5 44 10 600.00 2024-04-03 18:35:00 3D

\.

--

-- TOC entry 3663 (class 0 OID 24889)

-- Dependencies: 220

-- Data for Name: Зал; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Зал" ("Номер\_зала", "Вместимость", "График\_уборки", "Номер\_сотрудника") FROM stdin;

1 100 Понедельник, Среда, Пятница 4

2 80 Вторник, Четверг, Суббота 9

3 150 Ежедневно 4

4 120 Понедельник, Вторник, Среда 10

5 90 Четверг, Пятница, Воскресенье 7

6 110 Вторник, Четверг, Суббота 9

7 95 Понедельник, Среда, Пятница 4

8 85 Четверг, Пятница, Воскресенье 7

9 130 Понедельник, Вторник, Среда 10

10 75 Вторник, Четверг, Суббота 12

11 140 Ежедневно 13

12 100 Понедельник, Среда, Пятница 14

13 120 Четверг, Пятница, Воскресенье 15

14 90 Вторник, Четверг, Суббота 11

15 105 Понедельник, Вторник, Среда 12

\.

--

-- TOC entry 3664 (class 0 OID 24899)

-- Dependencies: 221

-- Data for Name: Клиент; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Клиент" ("Номер\_клиента", email, "ФИО", "История\_заказов") FROM stdin;

1 alex@example.com Алексей Смирнов Властелин колец: Братство кольца, Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода

2 maria@example.com Мария Иванова Интерстеллар, Начало, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво

3 sergey@example.com Сергей Ковалев Титаник, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво

4 olga@example.com Ольга Петрова Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Форрест Гамп, Парк Юрского периода, Криминальное чтиво

5 dmitry@example.com Дмитрий Васильев Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво

6 anna@example.com Анна Николаева Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Парк Юрского периода

7 pavel@example.com Павел Лебедев Титаник, Интерстеллар, Начало, Матрица, Темный рыцарь, Форрест Гамп, Парк Юрского периода, Криминальное чтиво

8 svetlana@example.com Светлана Григорьева Титаник, Властелин колец: Братство кольца, Начало, Матрица, Побег из Шоушенка, Форрест Гамп, Парк Юрского периода, Криминальное чтиво

9 elena@example.com Елена Морозова Титаник, Властелин колец: Братство кольца, Интерстеллар, Матрица, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво

10 ivan@example.com Иван Федоров Титаник, Властелин колец: Братство кольца, Интерстеллар, Начало, Побег из Шоушенка, Темный рыцарь, Форрест Гамп, Криминальное чтиво

\.

--

-- TOC entry 3666 (class 0 OID 24923)

-- Dependencies: 223

-- Data for Name: Место; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Место" ("Номер\_сидения", "Номер\_зала", "Ряд", "Номер", "Является\_электронным") FROM stdin;

1 1 A 1 t

2 1 A 2 t

3 1 A 3 t

4 1 A 4 t

5 1 A 5 t

6 1 A 6 t

7 1 A 7 t

8 1 A 8 t

9 1 A 9 t

10 1 A 10 t

11 2 B 1 t

12 2 B 2 t

13 2 B 3 t

14 2 B 4 t

15 2 B 5 t

16 2 B 6 t

17 2 B 7 t

18 2 B 8 t

19 2 B 9 t

20 2 B 10 t

21 3 C 1 t

22 3 C 2 t

23 3 C 3 t

24 3 C 4 t

25 3 C 5 t

26 3 C 6 t

27 3 C 7 t

28 3 C 8 t

29 3 C 9 t

30 3 C 10 t

31 4 D 1 t

32 4 D 2 t

33 4 D 3 t

34 4 D 4 t

35 4 D 5 t

36 4 D 6 t

37 4 D 7 t

38 4 D 8 t

39 4 D 9 t

40 4 D 10 t

41 5 E 1 t

42 5 E 2 t

43 5 E 3 t

44 5 E 4 t

45 5 E 5 t

46 5 E 6 t

47 5 E 7 t

48 5 E 8 t

49 5 E 9 t

50 5 E 10 t

\.

--

-- TOC entry 3668 (class 0 OID 24953)

-- Dependencies: 225

-- Data for Name: Отзыв; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Отзыв" ("Номер\_отзыва", "Номер\_клиента", "Номер\_фильма", "Рейтинг", "Комментарий", "Источник") FROM stdin;

1 1 1 4.5 Отличный фильм, трогательная история. Google

2 2 2 5 Великолепное фэнтези с потрясающими спецэффектами. Yandex

3 3 3 4.8 Увлекательная научная фантастика. IMDb

4 4 4 4.2 Интересный сюжет и отличная игра актеров. Google

5 5 5 4.7 Классический боевик, рекомендую. Facebook

6 6 6 5 Вдохновляющая драма с глубоким смыслом. Yandex

7 7 7 4.9 Отличный фильм о супергероях. IMDb

8 8 8 4.6 Замечательная история с трогательными моментами. Google

9 9 9 4.3 Увлекательные приключения и отличная анимация. Facebook

10 10 10 4.8 Интересный криминальный сюжет с неожиданными поворотами. Yandex

11 1 3 4.7 Очень понравился фильм, отличный режиссер. IMDb

12 2 4 4.5 Интересная идея и реализация. Google

13 3 5 4.4 Классический фильм, стоит пересмотреть. Facebook

14 4 6 5 Вдохновляющий сюжет, рекомендую всем. Yandex

15 5 7 4.9 Отличный фильм, захватывающий до конца. IMDb

16 6 8 4.6 Прекрасная история, хорошо снято. Google

17 7 9 4.3 Увлекательный фильм для всей семьи. Facebook

18 8 10 4.8 Интересный и захватывающий сюжет. Yandex

19 9 2 5 Лучший фильм в жанре фэнтези. IMDb

20 10 1 4.5 Классика кинематографа, рекомендую посмотреть. Kinopoisk

\.

--

-- TOC entry 3665 (class 0 OID 24908)

-- Dependencies: 222

-- Data for Name: Сеанс; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Сеанс" ("Номер\_сеанса", "Номер\_фильма", "Номер\_зала", "Время\_начала", "Время\_окончания", "Формат\_показа", "Свободные\_места") FROM stdin;

1 1 1 10:00:00 13:15:00 2D 100

2 2 2 12:00:00 14:58:00 3D 80

3 3 3 14:00:00 16:49:00 IMAX 150

4 4 4 16:00:00 18:28:00 2D 120

5 5 5 18:00:00 20:16:00 3D 90

6 6 1 20:00:00 22:22:00 2D 100

7 7 2 22:30:00 00:02:00 3D 80

8 8 3 09:30:00 11:52:00 2D 150

9 9 4 11:00:00 13:05:00 IMAX 120

10 10 5 13:00:00 15:34:00 2D 90

11 1 2 15:00:00 18:15:00 3D 80

12 3 1 17:00:00 19:49:00 IMAX 150

13 5 3 19:00:00 21:16:00 3D 90

14 7 4 21:00:00 23:32:00 2D 120

15 9 5 23:00:00 01:05:00 IMAX 90

\.

--

-- TOC entry 3661 (class 0 OID 24879)

-- Dependencies: 218

-- Data for Name: Сотрудник; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Сотрудник" ("Номер\_сотрудника", "Должность", "ФИО", "Телефон") FROM stdin;

1 Менеджер Иванов Иван Иванович +7-912-345-67-89

2 Администратор Петрова Мария Сергеевна +7-913-456-78-90

3 Техник Сидоров Алексей Петрович +7-914-567-89-01

4 Уборщик Кузнецова Елена Владимировна +7-915-678-90-12

5 Бухгалтер Морозова Ольга Николаевна +7-916-789-01-23

6 Менеджер Федоров Дмитрий Александрович +7-917-890-12-34

7 Администратор Васильев Сергей Михайлович +7-918-901-23-45

8 Техник Николаева Анна Викторовна +7-919-012-34-56

9 Уборщик Лебедев Павел Юрьевич +7-920-123-45-67

10 Бухгалтер Григорьева Светлана Ивановна +7-921-234-56-78

11 Менеджер Новиков Сергей Иванович +375-29-345-67-89

12 Администратор Соколова Ольга Петровна +375-29-456-78-90

13 Техник Кузнецов Николай Александрович +375-29-567-89-01

14 Уборщик Иванова Наталья Сергеевна +375-29-678-90-12

15 Бухгалтер Смирнов Павел Викторович +375-29-789-01-23

16 Менеджер Козлова Марина Дмитриевна +375-29-890-12-34

17 Администратор Васильев Андрей Николаевич +375-29-901-23-45

18 Техник Лебедева Елизавета Петровна +375-29-012-34-56

19 Уборщик Федоров Максим Иванович +375-29-123-45-67

20 Бухгалтер Григорьев Алексей Сергеевич +375-29-234-56-78

\.

--

-- TOC entry 3662 (class 0 OID 24884)

-- Dependencies: 219

-- Data for Name: Фильм; Type: TABLE DATA; Schema: Кинотеатр; Owner: postgres

--

COPY "Кинотеатр"."Фильм" ("Номер\_фильма", "Название", "Длительность", "Жанр") FROM stdin;

1 Титаник 03:15:00 Драма

2 Властелин колец: Братство кольца 02:58:00 Фэнтези

3 Интерстеллар 02:49:00 Научная фантастика

4 Начало 02:28:00 Экшн

5 Матрица 02:16:00 Боевик

6 Побег из Шоушенка 02:22:00 Драма

7 Темный рыцарь 02:32:00 Боевик

8 Форрест Гамп 02:22:00 Драма

9 Парк Юрского периода 02:05:00 Приключения

10 Криминальное чтиво 02:34:00 Криминал

11 Гарри Поттер и философский камень 02:32:00 Фэнтези

12 Аватар 02:42:00 Научная фантастика

13 Бойцовский клуб 02:19:00 Драма

14 Зеленая миля 03:09:00 Фэнтези

15 Список Шиндлера 03:15:00 Историческая драма

16 Король Лев 01:58:00 Мультфильм

17 Железный человек 02:06:00 Боевик

18 Форсаж 02:10:00 Экшн

19 Пираты Карибского моря: Проклятие Черной жемчужины 02:23:00 Приключения

20 Индиана Джонс: В поисках утраченного ковчега 01:58:00 Приключения

\.

--

-- TOC entry 3502 (class 2606 OID 24937)

-- Name: Билет Билет\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Билет"

ADD CONSTRAINT "Билет\_pkey" PRIMARY KEY ("Номер\_билета");

--

-- TOC entry 3485 (class 2606 OID 24893)

-- Name: Зал Зал\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Зал"

ADD CONSTRAINT "Зал\_pkey" PRIMARY KEY ("Номер\_зала");

--

-- TOC entry 3488 (class 2606 OID 24907)

-- Name: Клиент Клиент\_email\_key; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Клиент"

ADD CONSTRAINT "Клиент\_email\_key" UNIQUE (email);

--

-- TOC entry 3490 (class 2606 OID 24905)

-- Name: Клиент Клиент\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Клиент"

ADD CONSTRAINT "Клиент\_pkey" PRIMARY KEY ("Номер\_клиента");

--

-- TOC entry 3497 (class 2606 OID 24927)

-- Name: Место Место\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Место"

ADD CONSTRAINT "Место\_pkey" PRIMARY KEY ("Номер\_сидения");

--

-- TOC entry 3506 (class 2606 OID 24957)

-- Name: Отзыв Отзыв\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Отзыв"

ADD CONSTRAINT "Отзыв\_pkey" PRIMARY KEY ("Номер\_отзыва");

--

-- TOC entry 3494 (class 2606 OID 24912)

-- Name: Сеанс Сеанс\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Сеанс"

ADD CONSTRAINT "Сеанс\_pkey" PRIMARY KEY ("Номер\_сеанса");

--

-- TOC entry 3479 (class 2606 OID 24883)

-- Name: Сотрудник Сотрудник\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Сотрудник"

ADD CONSTRAINT "Сотрудник\_pkey" PRIMARY KEY ("Номер\_сотрудника");

--

-- TOC entry 3482 (class 2606 OID 24888)

-- Name: Фильм Фильм\_pkey; Type: CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Фильм"

ADD CONSTRAINT "Фильм\_pkey" PRIMARY KEY ("Номер\_фильма");

--

-- TOC entry 3498 (class 1259 OID 24976)

-- Name: idx\_билет\_клиент; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_билет\_клиент" ON "Кинотеатр"."Билет" USING btree ("Номер\_клиента");

--

-- TOC entry 3499 (class 1259 OID 24974)

-- Name: idx\_билет\_сеанс; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_билет\_сеанс" ON "Кинотеатр"."Билет" USING btree ("Номер\_сеанса");

--

-- TOC entry 3500 (class 1259 OID 24975)

-- Name: idx\_билет\_сидение; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_билет\_сидение" ON "Кинотеатр"."Билет" USING btree ("Номер\_сидения");

--

-- TOC entry 3483 (class 1259 OID 24969)

-- Name: idx\_зал\_сотрудника; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_зал\_сотрудника" ON "Кинотеатр"."Зал" USING btree ("Номер\_сотрудника");

--

-- TOC entry 3486 (class 1259 OID 24970)

-- Name: idx\_клиент\_email; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_клиент\_email" ON "Кинотеатр"."Клиент" USING btree (email);

--

-- TOC entry 3495 (class 1259 OID 24973)

-- Name: idx\_место\_зал; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_место\_зал" ON "Кинотеатр"."Место" USING btree ("Номер\_зала");

--

-- TOC entry 3503 (class 1259 OID 24977)

-- Name: idx\_отзыв\_клиент; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_отзыв\_клиент" ON "Кинотеатр"."Отзыв" USING btree ("Номер\_клиента");

--

-- TOC entry 3504 (class 1259 OID 24978)

-- Name: idx\_отзыв\_фильм; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_отзыв\_фильм" ON "Кинотеатр"."Отзыв" USING btree ("Номер\_фильма");

--

-- TOC entry 3491 (class 1259 OID 24972)

-- Name: idx\_сеанс\_зал; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_сеанс\_зал" ON "Кинотеатр"."Сеанс" USING btree ("Номер\_зала");

--

-- TOC entry 3492 (class 1259 OID 24971)

-- Name: idx\_сеанс\_фильм; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_сеанс\_фильм" ON "Кинотеатр"."Сеанс" USING btree ("Номер\_фильма");

--

-- TOC entry 3480 (class 1259 OID 24968)

-- Name: idx\_фильм\_название; Type: INDEX; Schema: Кинотеатр; Owner: postgres

--

CREATE INDEX "idx\_фильм\_название" ON "Кинотеатр"."Фильм" USING btree ("Название");

--

-- TOC entry 3511 (class 2606 OID 24948)

-- Name: Билет Билет\_Номер\_клиента\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Билет"

ADD CONSTRAINT "Билет\_Номер\_клиента\_fkey" FOREIGN KEY ("Номер\_клиента") REFERENCES "Кинотеатр"."Клиент"("Номер\_клиента") ON DELETE CASCADE;

--

-- TOC entry 3512 (class 2606 OID 24938)

-- Name: Билет Билет\_Номер\_сеанса\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Билет"

ADD CONSTRAINT "Билет\_Номер\_сеанса\_fkey" FOREIGN KEY ("Номер\_сеанса") REFERENCES "Кинотеатр"."Сеанс"("Номер\_сеанса") ON DELETE CASCADE;

--

-- TOC entry 3513 (class 2606 OID 24943)

-- Name: Билет Билет\_Номер\_сидения\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Билет"

ADD CONSTRAINT "Билет\_Номер\_сидения\_fkey" FOREIGN KEY ("Номер\_сидения") REFERENCES "Кинотеатр"."Место"("Номер\_сидения") ON DELETE CASCADE;

--

-- TOC entry 3507 (class 2606 OID 24894)

-- Name: Зал Зал\_Номер\_сотрудника\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Зал"

ADD CONSTRAINT "Зал\_Номер\_сотрудника\_fkey" FOREIGN KEY ("Номер\_сотрудника") REFERENCES "Кинотеатр"."Сотрудник"("Номер\_сотрудника") ON DELETE SET NULL;

--

-- TOC entry 3510 (class 2606 OID 24928)

-- Name: Место Место\_Номер\_зала\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Место"

ADD CONSTRAINT "Место\_Номер\_зала\_fkey" FOREIGN KEY ("Номер\_зала") REFERENCES "Кинотеатр"."Зал"("Номер\_зала") ON DELETE CASCADE;

--

-- TOC entry 3514 (class 2606 OID 24958)

-- Name: Отзыв Отзыв\_Номер\_клиента\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Отзыв"

ADD CONSTRAINT "Отзыв\_Номер\_клиента\_fkey" FOREIGN KEY ("Номер\_клиента") REFERENCES "Кинотеатр"."Клиент"("Номер\_клиента") ON DELETE CASCADE;

--

-- TOC entry 3515 (class 2606 OID 24963)

-- Name: Отзыв Отзыв\_Номер\_фильма\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Отзыв"

ADD CONSTRAINT "Отзыв\_Номер\_фильма\_fkey" FOREIGN KEY ("Номер\_фильма") REFERENCES "Кинотеатр"."Фильм"("Номер\_фильма") ON DELETE CASCADE;

--

-- TOC entry 3508 (class 2606 OID 24918)

-- Name: Сеанс Сеанс\_Номер\_зала\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Сеанс"

ADD CONSTRAINT "Сеанс\_Номер\_зала\_fkey" FOREIGN KEY ("Номер\_зала") REFERENCES "Кинотеатр"."Зал"("Номер\_зала") ON DELETE CASCADE;

--

-- TOC entry 3509 (class 2606 OID 24913)

-- Name: Сеанс Сеанс\_Номер\_фильма\_fkey; Type: FK CONSTRAINT; Schema: Кинотеатр; Owner: postgres

--

ALTER TABLE ONLY "Кинотеатр"."Сеанс"

ADD CONSTRAINT "Сеанс\_Номер\_фильма\_fkey" FOREIGN KEY ("Номер\_фильма") REFERENCES "Кинотеатр"."Фильм"("Номер\_фильма") ON DELETE CASCADE;

-- Completed on 2024-12-10 01:53:44 +03

--

-- PostgreSQL database dump complete

--

В результате работы были созданы и занесены данные в таблицы:

1. Сотрудник (рисунок 2),
2. Фильм (рисунок 3),
3. Зал (рисунок 4),
4. Клиент (рисунок 5),
5. Сеанс (рисунок 6),
6. Место (рисунок 7),
7. Билет (рисунок 8),
8. Отзыв (рисунок 9).

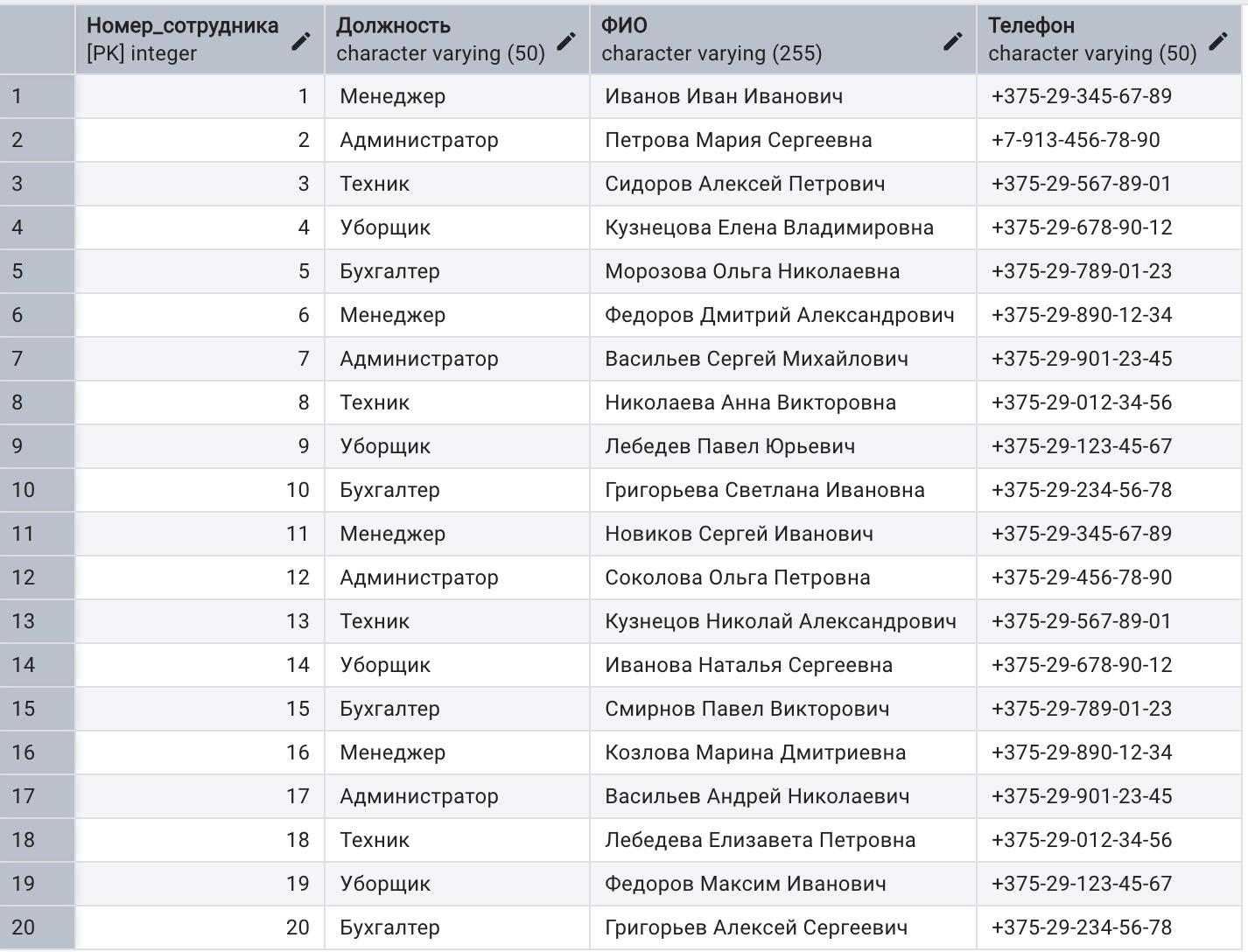


Рисунок 2. Сотрудник

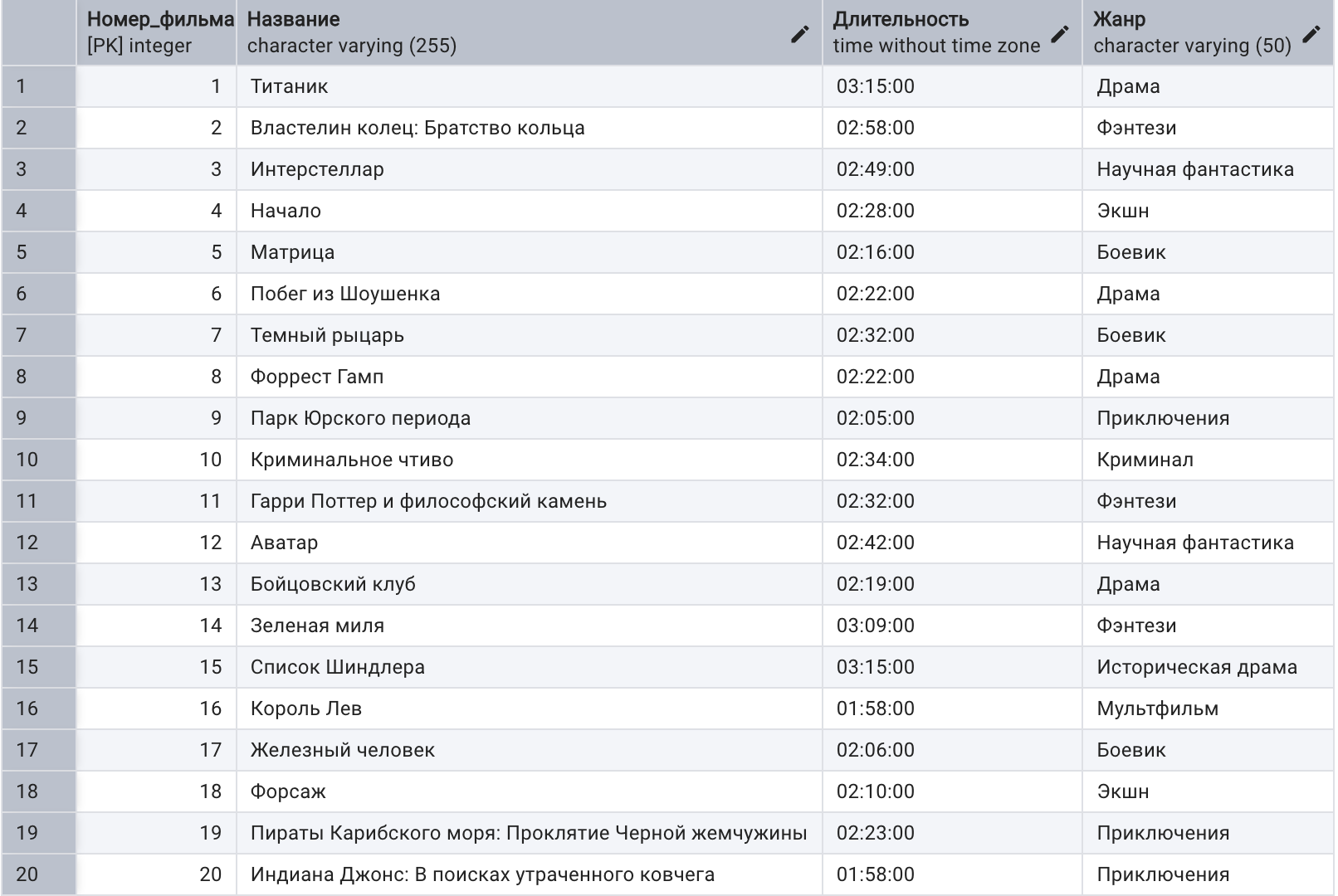


Рисунок 3. Фильм

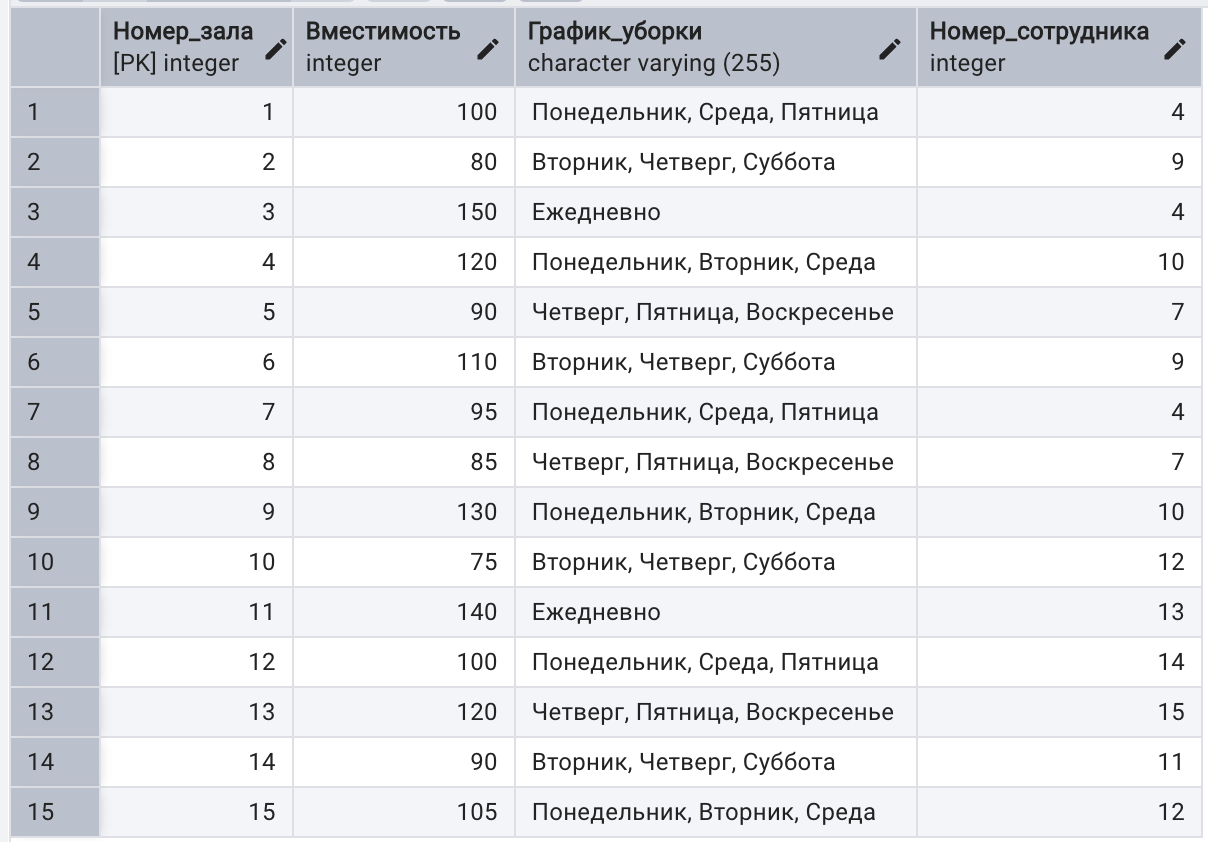


Рисунок 4. Зал

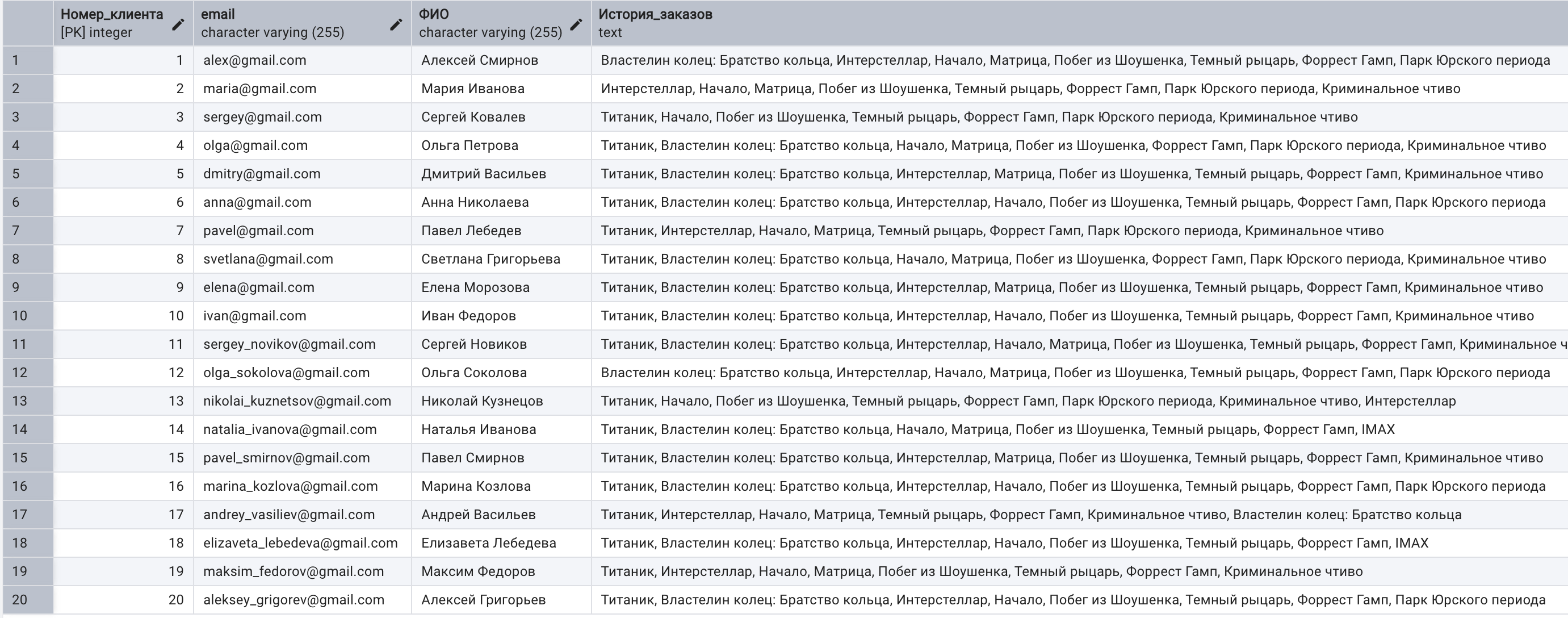


Рисунок 5. Клиент

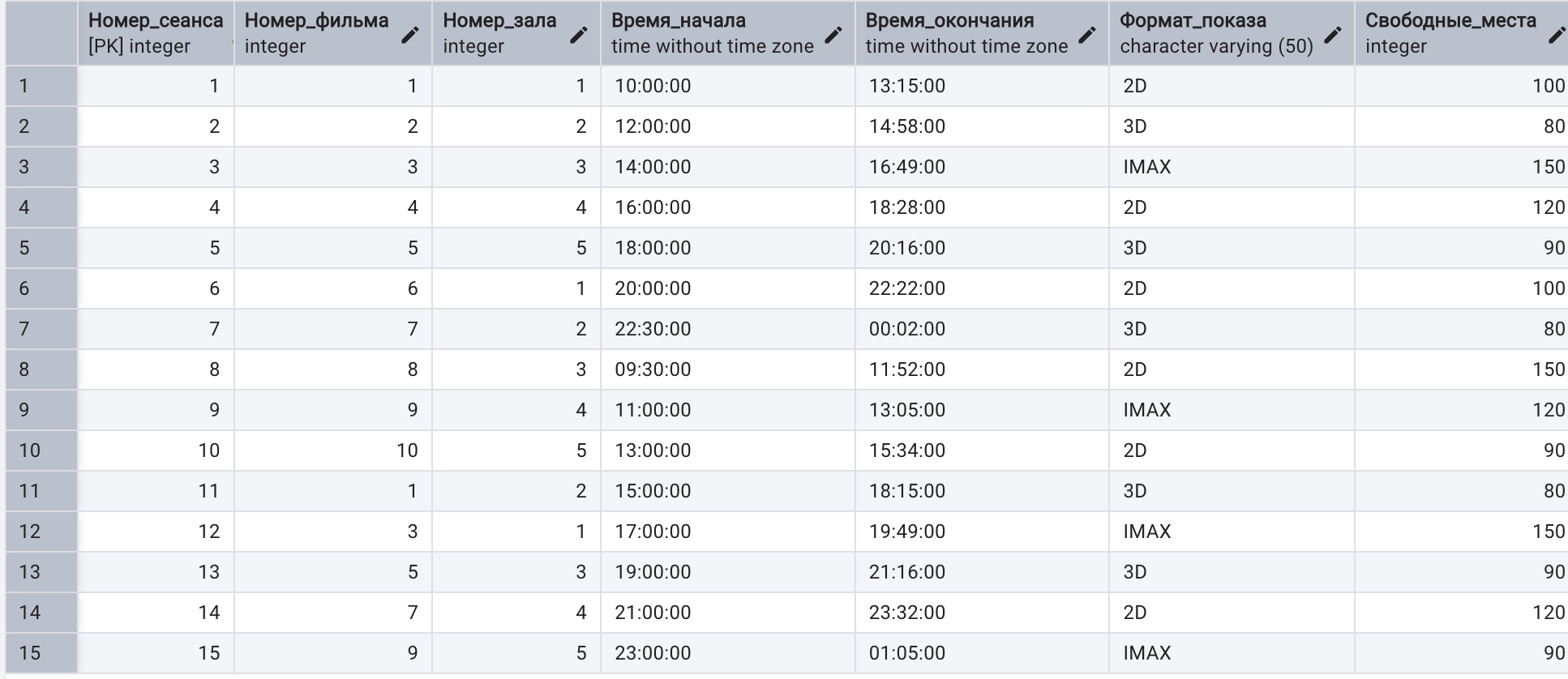


Рисунок 6. Сеанс

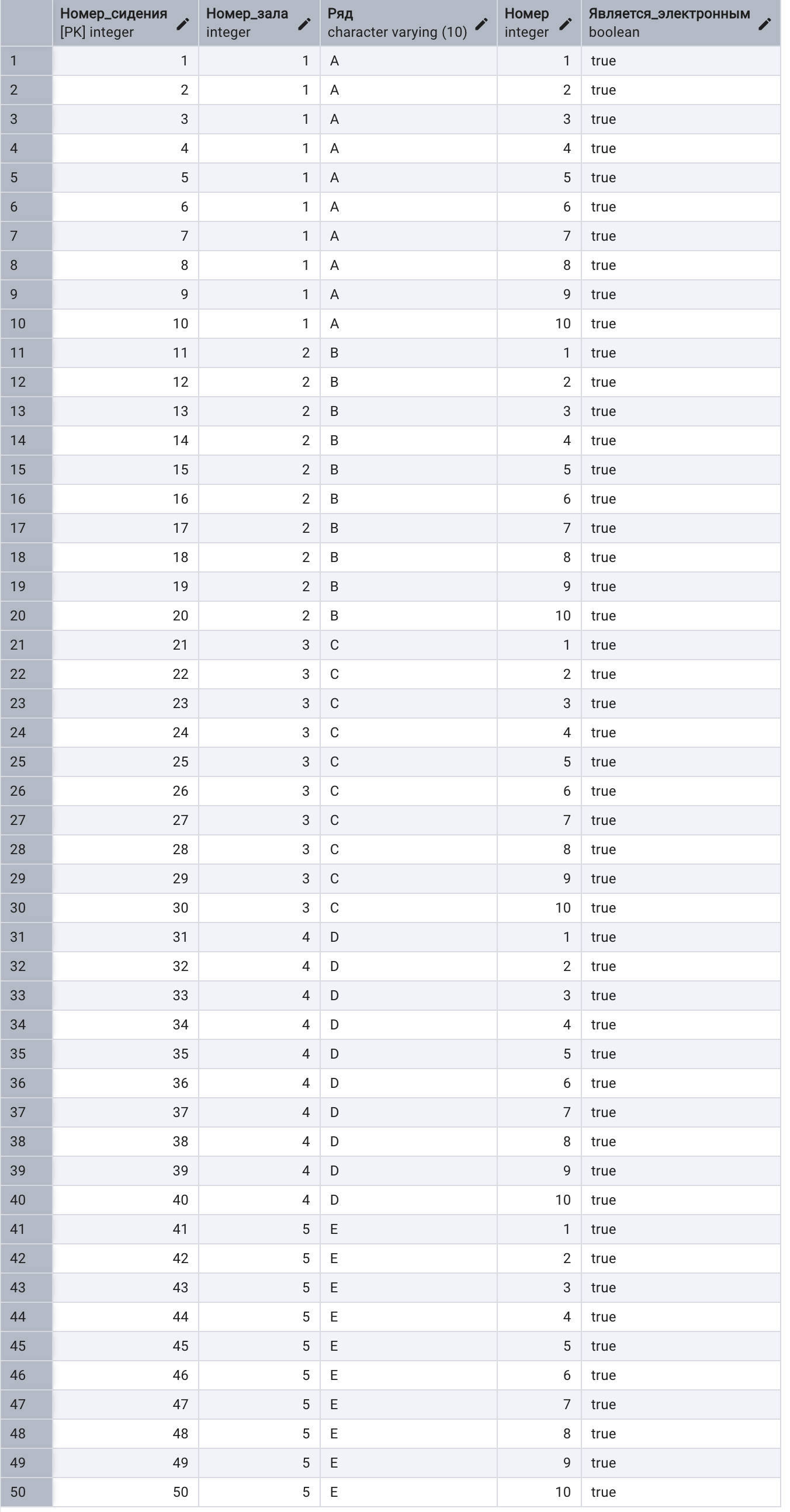


Рисунок 7. Место

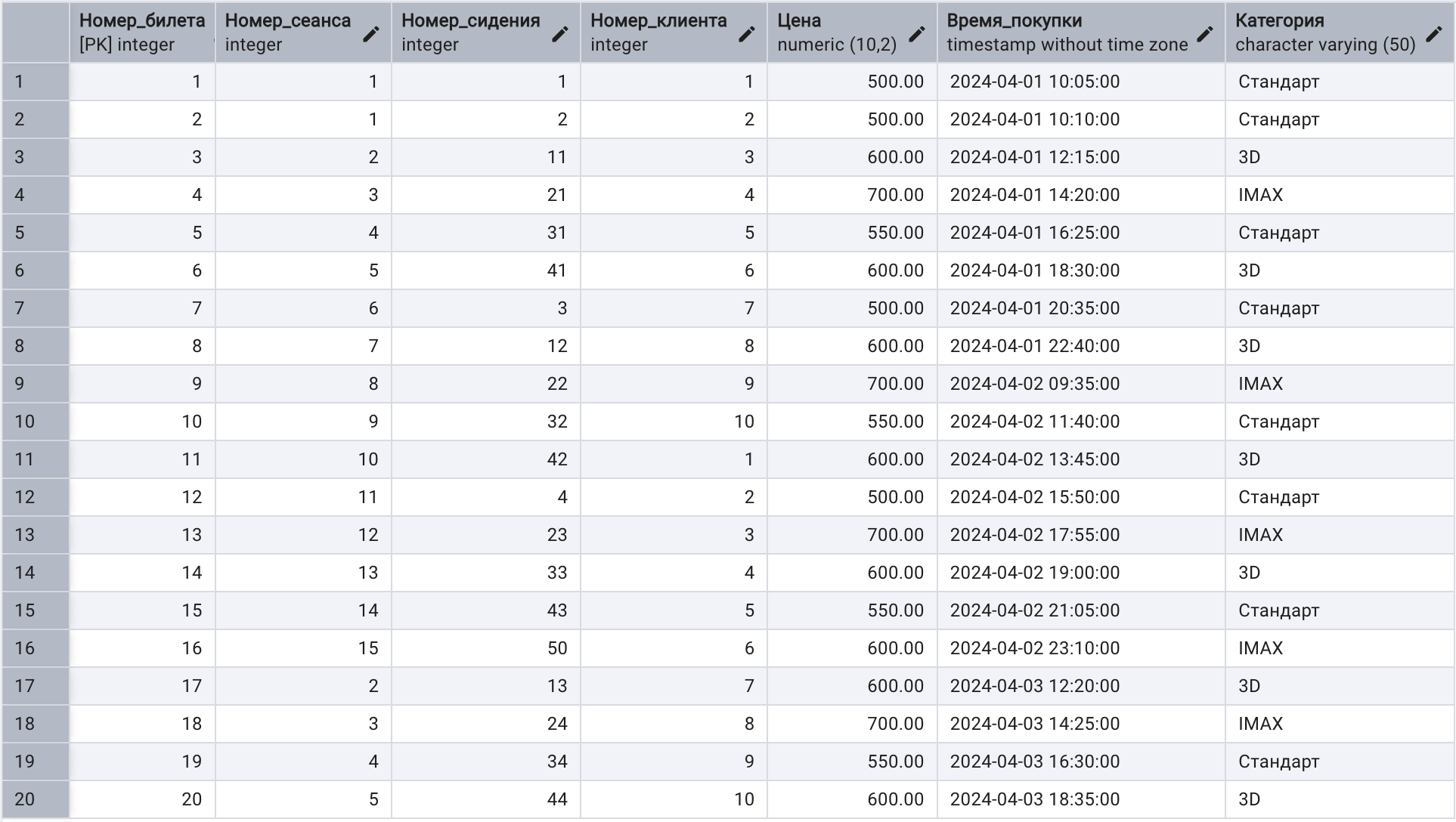


Рисунок 8. Билет

Вывод: в ходе лабораторной работы были сформированы и выполнены QL-запросы для создания таблиц базы данных и заполнения данными

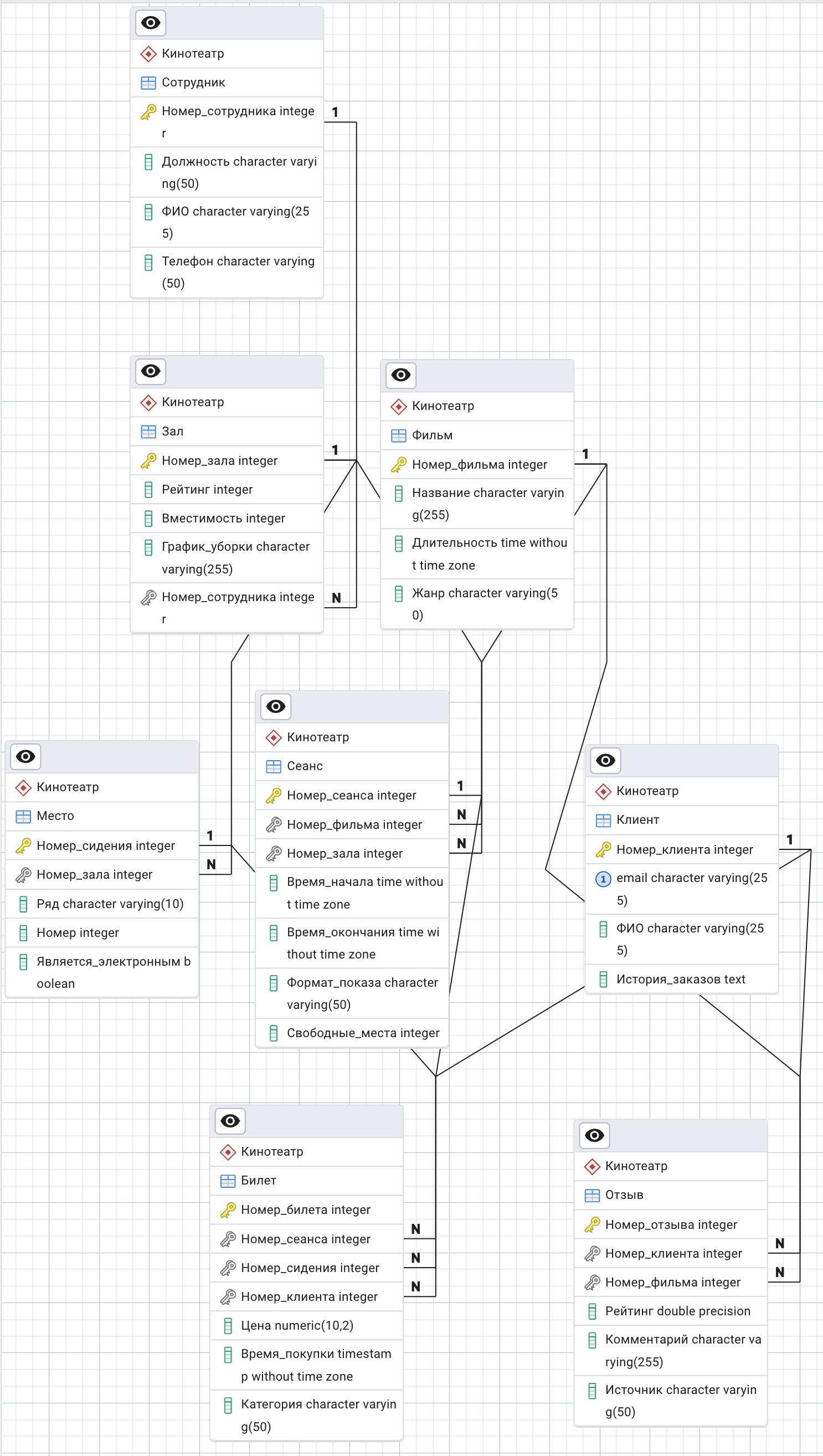


Рисунок 10. Реляционная диаграмма схемы «Кинотеатр»