

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Базы данных»
Тема: Проектирование ER модели и структуры БД по текстовому
описанию предметной области.

Студент гр. 3384

Копасова К. А.

Преподаватель

Михайлова С. А.

Санкт-Петербург

2025

Цель работы: спроектировать ER модель и структуру базы данных по текстовому описанию предметной области.

Задание:

Пусть требуется создать программную систему, предназначенную для работников справочной службы кинотеатров города. Такая система должна обеспечивать хранение сведений о кинотеатрах города, о фильмах, которые в них демонстрируются, о сеансах и билетах на эти сеансы. Сведения о кинотеатре — это его название, район города, где расположен кинотеатр, категория, вместимость. Сведения о фильме — это название фильма, режиссёр, оператор, актёры, сыгравшие главные роли, жанр; производство, наличие призов кинофестивалей, продолжительность сеанса, кадр из фильма для рекламы. Кроме того, должна храниться информация о репертуаре кинотеатров на месяц, то есть о том какие фильмы, когда и где демонстрируются, о ценах на билеты и о количестве свободных мест на тот или иной сеанс. На разных сеансах в одном кинотеатре могут идти разные фильмы, а если в кинотеатре несколько залов, то и на одном. Кинотеатр может ввести новый фильм в репертуар или убрать фильм из репертуара. Работник справочной службы может корректировать перечень фильмов, находящихся в прокате – добавлять новые фильмы и снимать с проката, а также перечень кинотеатров, поскольку кинотеатры могут открываться или закрываться, причём иногда временно, например, на ремонт. Цена билета определяется прокатной стоимостью копии фильма, сеансом и категорией кинотеатра. Справочной службе могут потребоваться следующие сведения о текущем состоянии проката фильмов в городе:

1. Репертуар кинотеатра?
2. Адрес и район кинотеатра ?
3. Число свободных мест на данный сеанс в указанном кинотеатре?
4. Цена билетов на данный сеанс в указанном кинотеатре?

5. Жанр, производство и режиссёр данного фильма ?
6. Какие фильмы имеют награды, когда и в каких кинотеатрах они демонстрируются?
7. В каких кинотеатрах в указанный день на указанных сеансах демонстрируется комедия?

Выполнение работы

Проведём анализ технического задания и выделим важные сущности для ER модели:

Кинотеатр - сущность для хранения информации о кинотеатрах города, хранит следующие атрибуты: название, район города, где расположен кинотеатр, категория и вместимость. Позволяет отвечать на запросы о расположении кинотеатров и их характеристиках.

Фильм - сущность для хранения информации о показываемых фильмах, хранит следующие атрибуты: название, режиссёр, оператор, главные актёры, жанр, производство, продолжительность сеанса и кадр для рекламы.

Зал - сущность для хранения залов у кинотеатров, хранит следующие атрибуты: номер/название и количество свободных мест. Позволяет учитывать, что в одном кинотеатре могут идти одновременно разные фильмы, и управлять вместимостью.

Сеанс - сущность для формирования репертуара и расписания показов, содержит следующие атрибуты: дата, время начала и время окончания.

Билет - сущность для учёта мест на сеансах и контроля продаж, содержит следующие атрибуты: номер ряда, номер места, состояние (продан или нет) и цена.

Призы - сущность для связки фильмов и фестивалей. Содержит атрибут названия.

Фестиваль - сущность для указания различных фестивалей для фильмов. Содержит атрибут названия.

Опишем связи между всеми сущностями:

Кинотеатр - зал: связь «один ко многим» - один кинотеатр может иметь несколько залов, но каждый зал принадлежит только одному кинотеатру.

Зал - сеанс: связь «один ко многим» - в одном зале может проходить множество сеансов в разное время, но каждый сеанс привязан к конкретному залу.

Фильм - сеанс: связь «один ко многим» - один фильм может демонстрировать на многих сеансах в разных кинотеатрах, но каждый сеанс показывает только один фильм.

Сеанс - билет: связь «один ко многим» - на одном сеансе может продаваться множество билетов, но каждый билет привязан к конкретному сеансу.

Фильм - призы: связь «один ко многим» - один приз может принадлежать разным фильмам, но один фильм может иметь конкретный приз только один раз.

Фестиваль - призы: связь «один ко многим» - один фестиваль имеет множество призов, но каждый конкретный приз принадлежит определённому фестивалю.

Получаем следующую ER модель, изображённую на рис. 1.

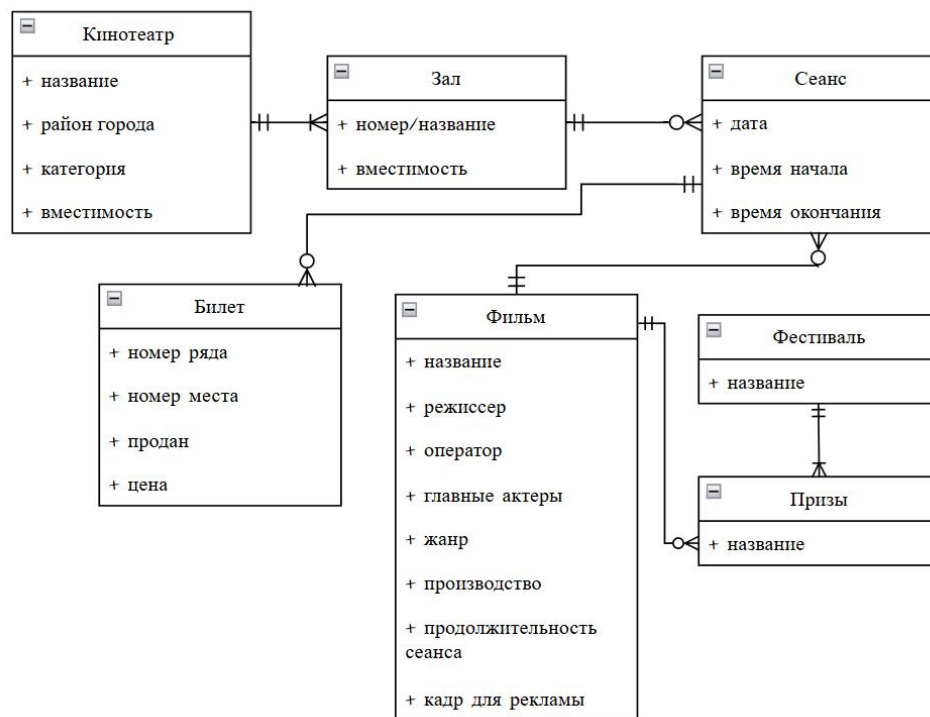


Рисунок 1 - ER модель.

Теперь на основе ER модели создадим реляционную модель:

1) Cinema (Кинотеатры):

- a) Первичный ключ - cinema_id: INT;
- b) Атрибуты - title:VARCHAR (название), city_region:VARCHAR (регион города), category:VARCHAR (категория), total_capacity:INT (общая вместимость).

2) Films (Фильм):

- a) Первичный ключ - film_id: INT;
- b) Атрибуты - title:VARCHAR (название), director:VARCHAR (режиссер), operator:VARCHAR (оператор), main_actors:VARCHAR (главные актеры), genre:VARCHAR (жанр), production:VARCHAR (производство), session_duration:INT (продолжительность сеанса), shot_advertising:VARCHAR (кадр для рекламы).

3) Cinema_halls (Залы):

- a) Первичный ключ - hall_id:INT;
- b) Внешний ключ - cinema_id: INT;
- c) Атрибуты - title:VARCHAR (название), capacity:INT (вместимость).

4) Sessions (Сеансы):

- a) Первичный ключ - session_id:INT;
- b) Внешний ключ - hall_id:INT, film_id: INT;
- c) Атрибуты - date:DATE (дата), start_time:TIME (время начала), end_time:TIME (время окончания).

5) Tickets (Билеты):

- a) Первичный ключ - ticket_id:INT;

b) Внешний ключ - session_id:INT;

c) Атрибуты - row_number:INT (номер ряда), place_number:INT (номер места), sold_out:BIT (продан - 1, не продан - 0, изначально 0), cost:INT (цена).

6) Prizes (Призы):

a) Первичный ключ - prizes_id:INT;

b) Внешний ключ - festival_id:INT, film_id:INT;

c) Атрибуты - title:VARCHAR (название).

7) Festival (Фестивали):

a) Первичный ключ - festival_id:INT;

b) Атрибуты - title:VARCHAR (название).

Получаем следующую реляционную модель, изображённую на рис. 2.

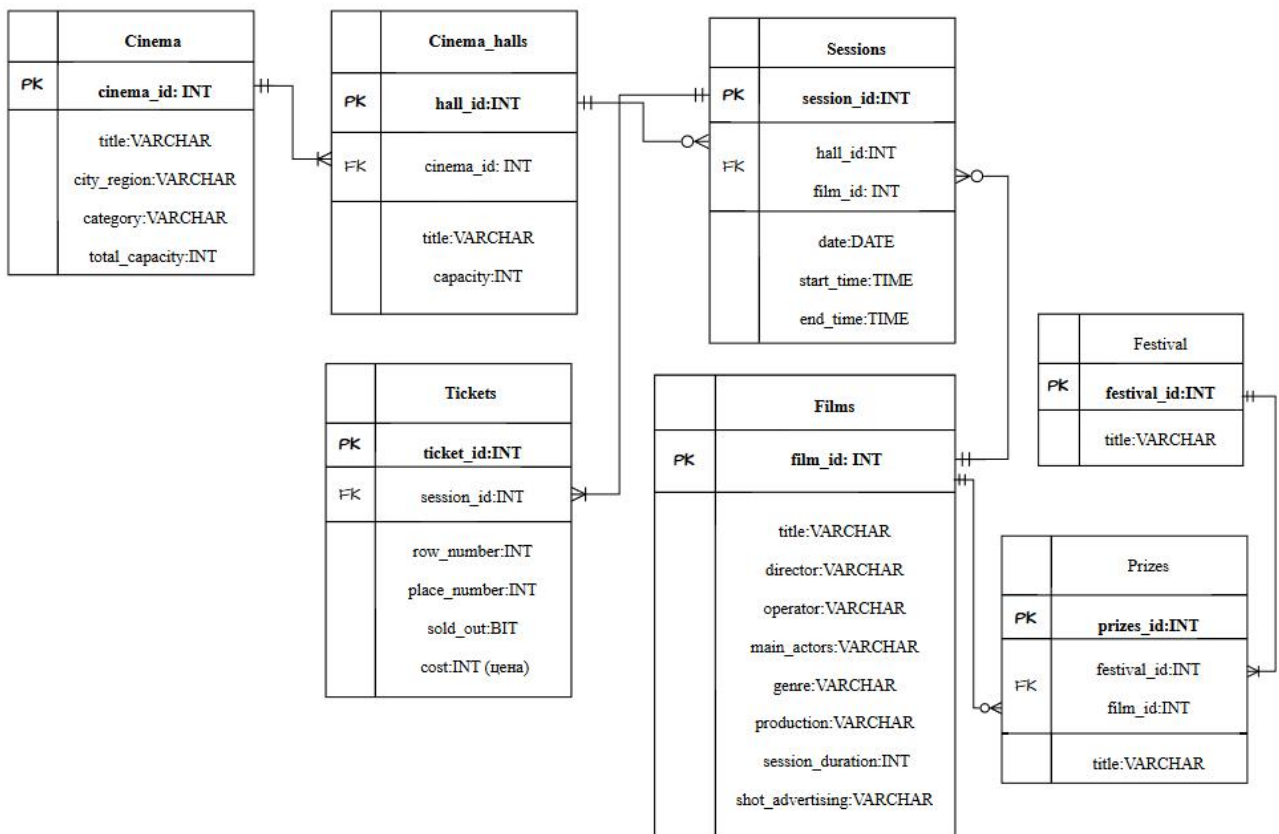


Рисунок 2 - Реляционная модель.

Далее проверим и обоснуем, что реляционная модель соответствует НФБК.

1. Проверка на соответствие первой нормальной формы (1НФ): все атрибуты реляционной модели содержит только неделимые значения. Поля `main_actors` и `festival_prizes` хранятся как текстовые описания, а не списки, что обеспечивает соответствие 1НФ.

2. Проверка на соответствие второй нормальной формы (2НФ): выполняется 1НФ и во всех таблицах атрибуты полнофункционально зависят от своих первичных ключей. Для таблицы `Cinema` все атрибуты полнофункционально зависят от первичного ключа `cinema_id`. Значение `cinema_id` однозначно определяет название кинотеатра, район города, категорию и общую вместимость. Для таблицы `Films` все атрибуты полнофункционально зависят от первичного ключа `film_id`. Идентификатор фильма однозначно определяет его название, режиссёра, оператора, главных актёров, жанр, производство, призы кинофестивалей, продолжительность сеанса и рекламный кадр. Для таблицы `Cinema_halls` все атрибуты полнофункционально зависят от первичного ключа `hall_id`. Идентификатор зала однозначно определяет кинотеатр, к которому он относится, название зала и его вместимость. Для таблицы `Sessions` все атрибуты полнофункционально зависят от первичного ключа `session_id`. Идентификатор сеанса однозначно определяет зал, фильм, дату, время начала и время окончания. Для таблицы `Tickets` все атрибуты полнофункционально зависят от первичного ключа `ticket_id`. Идентификатор билета однозначно определяет сеанс, номер ряда, номер места, статус продажи и стоимость. Во всех таблицах отсутствуют частичные зависимости, что подтверждает соответствие второй нормальной форме.

3. Проверка на соответствие третьей нормальной формы (3НФ): выполняется 2НФ и отсутствуют транзитивные зависимости между неключевыми атрибутами. В таблице `Cinema` отсутствуют транзитивные зависимости между атрибутами названия, района города, категории и общей

вместимости, они зависят только от первичного ключа. В таблице Films отсутствуют транзитивные зависимости между атрибутами режиссёра, оператора, жанра, производства и другими характеристиками фильма, они зависят исключительно от первичного ключа. В таблице Cinema_halls отсутствуют транзитивные зависимости между атрибутами кинотеатра, названия зала и вместимости, они независимы и определяются только идентификатором зала. В таблице Sessions отсутствуют транзитивные зависимости между атрибутами зала, фильма, даты, времени начала и окончания сеанса. В таблице Tickets отсутствуют транзитивные зависимости между атрибутами сеанса, номера ряда, номера места, статуса продажи и стоимости билета. Транзитивные зависимости между неключевыми атрибутами отсутствуют во всех таблицах базы данных, что подтверждает соответствие третьей нормальной форме.

4. Проверка на соответствие нормальной формы Бойса-Кодда (НФБК): каждая нетривиальная функциональная зависимость в таблице должна быть зависимостью от потенциального ключа. Для проверки соответствия необходимо выявить все потенциальные ключи в каждой таблице и убедиться, что все функциональные зависимости являются зависимостями от этих ключей. Таблица Cinema: первичный ключ - cinema_id. Единственным потенциальным ключом является cinema_id. Все атрибуты таблицы, включая название, район города, категорию и общую вместимость, функционально зависят от данного ключа. В таблице Films первичный ключ - film_id. Единственным потенциальным ключом является film_id. Все атрибуты таблицы, включая название фильма, режиссёра, оператора, главных актёров, жанр, производство, призы кинофестивалей, продолжительность сеанса и рекламный кадр, функционально зависят от данного ключа. В таблице Cinema_halls первичный ключ - hall_id. Единственным потенциальным ключом является hall_id. Все атрибуты таблицы, включая идентификатор кинотеатра, название зала и его вместимость, функционально зависят от данного ключа. В таблице Sessions первичный ключ - session_id. Потенциальными ключами являются session_id и

составной ключ (hall_id, date, start_time). Все атрибуты таблицы функционально зависят от обоих потенциальных ключей. Составной ключ (hall_id, date, start_time) обеспечивает уникальность сеансов в пределах одного зала и времени. В таблице Tickets первичный ключ - ticket_id. Потенциальными ключами являются ticket_id и составной ключ (session_id, row_number, place_number). Все атрибуты таблицы функционально зависят от обоих потенциальных ключей. Составной ключ (session_id, row_number, place_number) гарантирует уникальность каждого места на сеансе. Реляционная модель полностью соответствует нормальной форме Бойса-Кодда. Во всех таблицах каждый детерминант является потенциальным ключом, что исключает наличие нетривиальных функциональных зависимостей от неключевых атрибутов.

Вывод

В ходе работы была успешно спроектирована база данных для справочной службы кинотеатров. Разработана ER модель, отражающая сущности предметной области и связи между ними, а также детализированная реляционная модель с указанием типов данных и ключей. Проведённый анализ нормальных форм подтвердил, что спроектированная структура полностью соответствует НФБК, что обеспечивает отсутствие аномалий хранения и обработки данных.

ПРИЛОЖЕНИЕ 1

Ссылка на pull request: <https://github.com/moevm/sql-2025-3384/pull/9>