

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Базы данных»**  
**Тема: Проектирование ER модели и структуры БД**

Студент гр. 3341

Шаповаленко Е.В.

Преподаватель

Заславский М.М.

Санкт-Петербург

2025

### **Цель работы.**

Целью данной лабораторной работы является проектирование ER-модели, а затем структуры базы данных по текстовому описанию задачи из предметной области.

### **Задание.**

#### **Вариант 21.**

Пусть требуется создать программную систему, предназначенную для продавца журналов/комиксов. Такая система должна обеспечивать хранение сведений об имеющихся в магазине журналах, о читателях журналов и списке магазинов. Для каждого журнала в БД должны храниться следующие сведения: название журнала, серия, автор (ы), издательство, год издания, число экземпляров в каждом магазине, а также ISBN и дата продажи журнала. Сведения о читателях библиотеки должны включать почту (email), ФИО покупателя, дату рождения, адрес, номер телефон. Нужно учесть, что покупатели могут делать заказы с разных магазинов, но нужно сохранять информацию, о предпочтительных магазинах (в которых чаще делают заказы). Магазин имеет несколько отделов, которые характеризуются номером, названием и кол-во журналов (вместимость). Магазин может получать новые журналы и списывать старые. ISBN может измениться в результате переклассификации, а почта в результате перерегистрации. Продавцу могут потребоваться следующие сведения о текущем состоянии библиотеки:

- Какие журналы были куплены определенным покупателем?
- Как называется журнал с заданным ISBN?
- Какой ISBN у журнала с заданным названием?
- Когда журнал был куплен?
- Кто из покупателей купил журнал более месяца тому назад?
- Найти покупателя самых редких журналов (по наличию в магазине)?
- Какое число покупателей пользуется определенным магазином?
- Сколько покупателей младше 20 лет?

## Проектирование ER-модели.

В начале работы на основе текстового описания задания была спроектирована ER-модель, проиллюстрированная на рис. 1. Данная модель содержит основные сущности, требуемые для выполнения поставленной задачи, а также «ассоциативные сущности», создающие и расширяющие связи между выделенными сущностями.

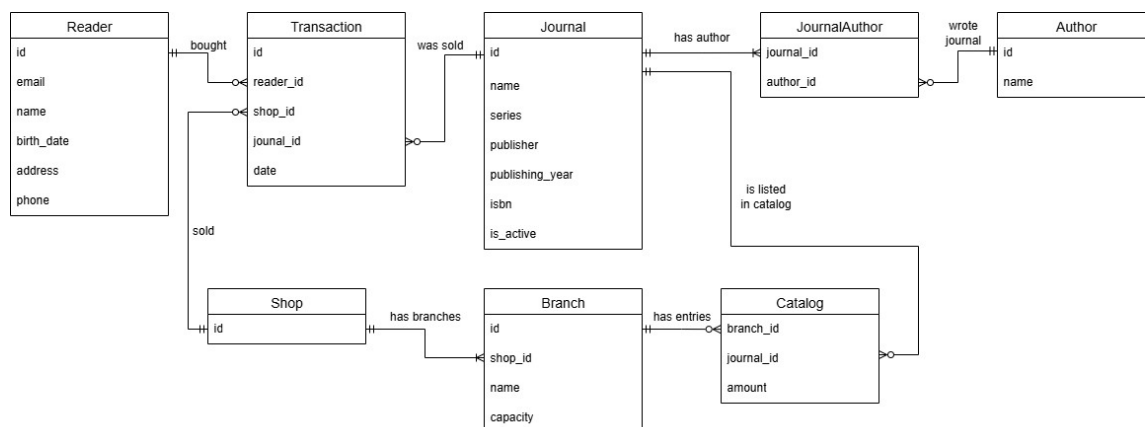


Рисунок 1 – спроектированная ER-модель

Основными сущностями в данной ER модели являются:

- *Journal* – сущность, содержащая основную информацию о журнале: название *name*, серия *series*, издательство *publisher*, год издания *publishing\_year*, шифр ISBN *isbn*. Также сущность содержит атрибут *is\_active*, который показывает, не списан ли журнал. В качестве ключа сущности используется суррогатный ключ *id*, т.к. ISBN может меняться, а книг с одинаковым названием, авторами и т.д. потенциально может быть несколько.
- *Author* – сущность, отражающая автора журнала. Содержит атрибут *name* в качестве имени автора, а в качестве ключа – суррогатный *id*, т.к. авторов с одинаковыми именами может быть несколько.
- *Reader* – сущность, отражающая покупателя. В качестве атрибутов содержит основные характеристики покупателя: почту *email*, имя *name*, дату рождения *birth\_date*, адрес *address*, номер телефона *phone\_number*. В качестве ключа используется суррогатный *id*, так как адрес электронной почты и номер телефона могут меняться, а другие атрибуты могут быть неуникальными.

- *Shop* – сущность отражающая магазин. В качестве ключа используется суррогатный *id*, так как другие атрибуты не подразумеваются.

- *Branch* – сущность содержащая основную информацию об отделе в магазине: номер *id*, название *name* и вместительность *capacity*. Также имеется атрибут *shop\_id* для связи с магазином. В качестве ключа используется номер отдела *id*, так как он не меняется и является уникальным. Магазин имеет несколько отделов, но по крайней мере один, а отдел обязательно принадлежит одному магазину. Поэтому связь обязательное «1 ко многим».

В качестве «ассоциативных сущностей», реализующих связи между описанными выше сущностями и хранящих важную информацию об этих связях, используются:

- *Transaction* – отражение связи между журналом и читателем, когда читатель покупает журнал из магазина. Журнал может быть куплен несколькими читателями, а читатель может покупать несколько журналов, причем по несколько раз, поэтому данная сущность важна, а её введение в том числе позволяет сохранять историю покупок журналов читателем. Кроме того, сохраняется информация о том, в каком магазине был куплен журнал. Атрибуты данной связи: идентификаторы журнала *journal\_id*, читателя *reader\_id* и магазина *shop\_id* и дата покупки книги *date*. В качестве ключа сущности используется суррогатный *id*. Сущность *Transaction* реализует связь «многие ко многим» между *Reader*, *Journal* и *Shop*, потому связана с *Reader*, *Journal* и *Shop* как необязательное «1 ко многим».

- *Catalog* – сущность, отражающая связь между журналом и отделом – сколько экземпляров журнала имеется в наличии в отделе. Атрибуты сущности – номер отдела зала *branch\_id* и идентификатор журнала *journal\_id*, которые вместе составляют ключ сущности, а также *amount* – количество экземпляров данного журнала в данном отделе. Экземпляры журнала могут находиться в разных отделах, да и не во всех, а в отделе может быть несколько журналов, а может и не быть, поэтому связь сущности *Catalog* с *Branch* и *Journal* – необязательное «1 ко многим».

- *JournalAuthor* – сущность, отражающая связь между журналом и автором(ами). Атрибуты сущности – идентификаторы журнала *journal\_id* и автора *author\_id*, которые вместе составляют ключ сущности. У журнала может быть несколько авторов, а у автора может быть несколько журналов или ни одного, поэтому связь сущности *JournalAuthor* с *Journal* – обязательное «1 ко многим», а *JournalAuthor* с *Author* – необязательное «1 ко многим».

## Проектирование структуры БД.

На основе спроектированной ER-модели была создана структура базы данных, проиллюстрированная рис. 2. Поскольку основные сущности и их связи в ER-модели были прописаны достаточно подробно, реляционная модель структурно похожа на ER-модель, а основные изменения – типизация атрибутов.

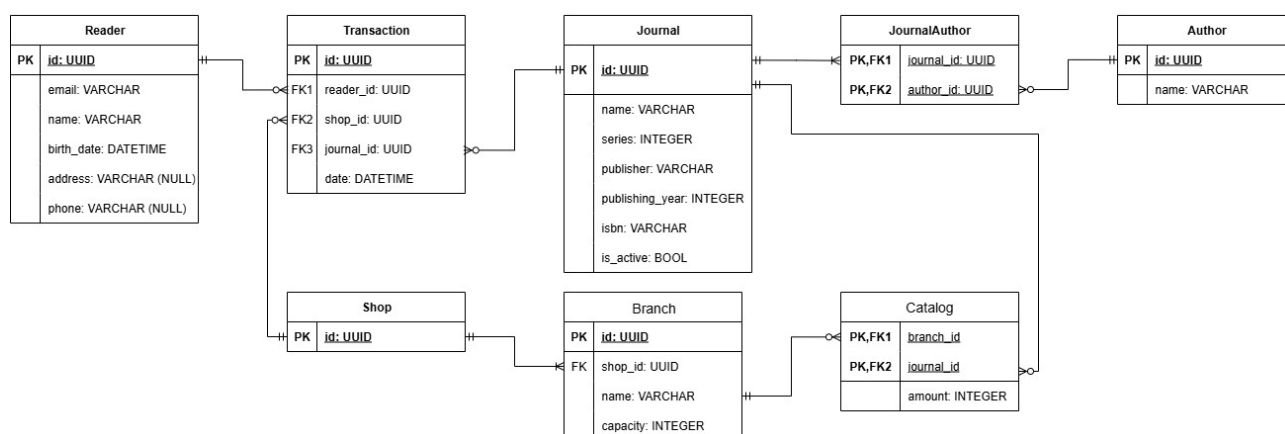


Рисунок 2 – структура БД

В качестве типа идентификаторов всех отношений был использован тип UUID, так как он хорошо справляется с созданием уникальных идентификаторов и по нему невозможно определить внутренний порядок в отношении, что улучшает безопасность системы. Поля с текстовой информацией и некоторые поля с числовой информацией, которую тяжело хранить в целочисленном типе или использовать её с таким типом (как, например, номер телефона), представлены типом *varchar*. Атрибуты, данных о которых может не иметься (например, номер телефона, адрес.), могут быть NULL.

## Обоснование НФБК.

После составления моделей была проведена проверка на их соответствие нормальной форме Бойса-Кодда (НФБК). Ниже перечислены нетривиальные функциональные зависимости отношений.

Для *Reader* функциональной зависимостью является *id* -> прочие атрибуты, что очевидно соответствует НФБК, поскольку *id* – суррогатный ключ. Если *email* в предметной области считать уникальными, то, объявив их уникальными в отношении, получим функциональные зависимости *email* -> прочие атрибуты, причем они будут соответствовать НФБК, т.к. тогда детерминант будет являться потенциальным ключом.

Для *Journal* очевидной функциональной зависимостью будет *id* -> прочие атрибуты, что, очевидно, соответствует НФБК. Также при уникальности шифра ISBN возможна функциональная зависимость *isbn* -> прочие атрибуты, что будет соответствовать НФБК, если считать атрибут *isbn* уникальным. Определить прочие детерминанты функциональной зависимости, как, например, название, издательство, год издания нельзя.

Соответствие отношений *Author*, *JournalAuthor*, *Shop*, *Catalog* НФБК тривиально или очевидно.

В отношении *Transaction* соответствие функциональной зависимости *id* -> прочие атрибуты очевидно. Если тройка *reader\_id*, *journal\_id*, *shop\_id* уникальна, то возможна функциональная зависимость с этой тройкой в качестве детерминанта, однако тогда эта же тройка будет потенциальным ключом, потому *Transaction* не нарушает НФБК.

## Выводы.

В результате выполнения лабораторной работы были изучены ER-модель и реляционная модель данных, были изучены принципы проектирования структуры БД, была создана структура БД, а также было проведено

исследование на соответствие спроектированных отношений нормальной форме Бойса-Кодда.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Ссылка на Pull-request с результатами выполненной лабораторной работы:

<https://github.com/moevm/sql-2025-3341/pull/8>