ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

Государственное автономное образовательное учреждение

Высшего образования города Москвы

«Московский городской педагогический университет»

(ГАОУ ВО МГПУ)

Институт цифрового образования

Департамент информатики, управления и технологий

ОТЧЕТ

по дисциплине «Базы данных»

на тему:

Разработка ER-диаграммы для заданной предметной

области. Применение нормализации к созданной модели данных.

Выполнил:

студент группы ЦИБ-241

Направление подготовки/Специальность

38.03.05 - Бизнес-информатика

Профиль подготовки/Специализация

Цифровые инновации в бизнесе

Вариант 5

Горынин Леонид Анатольевич

(Ф.И.О.) руководитель:

Босенко Тимур Муртазович

Москва 2025

Цельработы**:**

1. Овладеть практическими навыками анализа бизнес-требований для проектирования баз данных;

2. Научиться строить логическую, даталогическую **(**реляционную**)** и физическую модели данных с использованием Oracle SQL Developer Data Modeler;

3. освоить принципы нормализации данных для устранения избыточности и аномалий, приводя структуру таблиц к третьей нормальной форме (3NF);

4. научиться генерировать SQL-скрипт из физической модели и выполнять его на сервере MySQL с помощью MySQL Workbench.

Описание предметной области

Разрабатываемая база данных предназначена для автоматизации учёта данных в онлайн-библиотеке.  
Система обеспечивает хранение информации об авторах, книгах, экземплярах книг и читателях. Для каждой книги фиксируются название, год издания и жанр. Каждая книга может иметь несколько экземпляров, отличающихся уникальным инвентарным номером и текущим статусом (находится в библиотеке или выдана читателю).

База данных содержит сведения о зарегистрированных читателях, а также позволяет отслеживать процесс выдачи экземпляров книг. При выдаче фиксируется, какой экземпляр, какому читателю и на какой срок предоставлен. Это обеспечивает контроль за движением книжного фонда и своевременным возвратом литературы.

Ход выполнения

1. Создание логической ER-диаграммы из Oracle SQL Developer Data Modeler. Концептуальная модель. Нотация Баркера (Рисунок 1), Концептуальная модель. Нотация Бахмана (Рисунок 2), Концептуальная Технологическая модель (Рисунок 3).



Рисунок 1 — Концептуальная модель. Нотация Баркера



Рисунок 2 — Концептуальная модель. Нотация Бахмана



Рисунок 3 — Концептуальная Технологическая модель

1. Создание реляционной модели (Рисунок 4).



Рисунок 4 — Реляционная модель. ER-модель

Общее обоснование, что структура таблиц соответствует 3NF:

Отсутствие транзитивных зависимостей: Во всех таблицах неключевые атрибуты (не PK/FK) зависят исключительно от PK. FK используются только для ссылок между таблицами, но не вводят зависимости внутри. Если бы, например, в Books был атрибут author\_name (зависящий от author\_id), это создало бы транзитивность (book\_id → author\_id → author\_name), но такого нет — модель декомпозирована правильно.

Соответствие 1NF и 2NF: Подтверждено для всех (атомарность, полная зависимость от PK).

1. Финальный SQL-код. Результат успешного выполнения кода (Рисунок 5).

CREATE TABLE Authors

(

author\_id INTEGER NOT NULL ,

first\_name VARCHAR (50) NOT NULL ,

last\_name VARCHAR (50) NOT NULL ,

birth\_date DATE ,

nationality VARCHAR (20)

)

;

ALTER TABLE Authors

ADD CONSTRAINT Authors\_PK PRIMARY KEY ( author\_id ) ;

CREATE TABLE BookCopies

(

copy\_id INTEGER NOT NULL ,

book\_id INTEGER NOT NULL ,

inventory\_number VARCHAR (20) NOT NULL ,

status VARCHAR (20) ,

acquisition\_date DATE ,

candition VARCHAR (50) ,

reader\_id INTEGER , -- NULL-able

loan\_date DATE , -- NULL-able

due\_date DATE , -- NULL-able

return\_date DATE , -- NULL-able

Books\_book\_id INTEGER NOT NULL

)

;

ALTER TABLE BookCopies

ADD CONSTRAINT BookCopies\_PK PRIMARY KEY ( copy\_id ) ;

CREATE TABLE Books

(

book\_id INTEGER NOT NULL ,

author\_id INTEGER NOT NULL ,

title VARCHAR (100) NOT NULL ,

publication\_year INTEGER NOT NULL ,

genre\_id INTEGER NOT NULL ,

publisher\_id INTEGER ,

Authors\_author\_id INTEGER NOT NULL ,

Genres\_genre\_id INTEGER NOT NULL ,

Publishers\_publisher\_id INTEGER NOT NULL

)

;

ALTER TABLE Books

ADD CONSTRAINT Books\_PK PRIMARY KEY ( book\_id ) ;

CREATE TABLE Genres

(

genre\_id INTEGER NOT NULL ,

genre\_name VARCHAR (50) NOT NULL ,

description TEXT

)

;

ALTER TABLE Genres

ADD CONSTRAINT Genres\_PK PRIMARY KEY ( genre\_id ) ;

CREATE TABLE Publishers

(

publisher\_id INTEGER NOT NULL ,

publisher\_name VARCHAR (100) ,

country VARCHAR (50) ,

founded\_year INTEGER

)

;

ALTER TABLE Publishers

ADD CONSTRAINT Publishers\_PK PRIMARY KEY ( publisher\_id ) ;

CREATE TABLE Readers

(

reader\_id INTEGER NOT NULL ,

first\_name VARCHAR (50) NOT NULL ,

last\_name VARCHAR (50) NOT NULL ,

email VARCHAR (100) NOT NULL ,

registration\_date DATE NOT NULL ,

phone VARCHAR (20) ,

BookCopies\_copy\_id INTEGER NOT NULL ,

BookCopies\_book\_id INTEGER NOT NULL

)

;

ALTER TABLE Readers

ADD CONSTRAINT Readers\_PK PRIMARY KEY ( reader\_id ) ;

ALTER TABLE BookCopies

ADD CONSTRAINT BookCopies\_Books\_FK FOREIGN KEY

(

Books\_book\_id

)

REFERENCES Books

(

book\_id

)

;

ALTER TABLE Books

ADD CONSTRAINT Books\_Authors\_FK FOREIGN KEY

(

Authors\_author\_id

)

REFERENCES Authors

(

author\_id

)

;

ALTER TABLE Books

ADD CONSTRAINT Books\_Genres\_FK FOREIGN KEY

(

Genres\_genre\_id

)

REFERENCES Genres

(

genre\_id

)

;

ALTER TABLE Books

ADD CONSTRAINT Books\_Publishers\_FK FOREIGN KEY

(

Publishers\_publisher\_id

)

REFERENCES Publishers

(

publisher\_id

)

;

ALTER TABLE Readers

ADD CONSTRAINT Readers\_BookCopies\_FK FOREIGN KEY

(

BookCopies\_copy\_id

)

REFERENCES BookCopies

(

copy\_id

)

;

Изображение выглядит как текст, Шрифт, документ, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5 — Результат успешного выполнения SQL-кода

1. Физическая модель в PhpMyAdmin (Рисунок 6).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6 — Физическая модель (PhpMyAdmin)

Вывод

В ходе выполнения лабораторной работы были закреплены практические навыки проектирования и реализации базы данных с использованием Oracle SQL Developer Data Modeler и MySQL Workbench.

Был проведён анализ предметной области и выделены ключевые сущности, их атрибуты и связи. На основе полученной информации была построена логическая модель, которая затем была преобразована в реляционную с учётом нормализации до третьей нормальной формы (3NF).

Далее была создана физическая модель с указанием типов данных, после чего автоматически сгенерирован SQL-скрипт (DDL) для создания структуры базы данных. Полученный скрипт был успешно выполнен в персональной базе данных на сервере MySQL с использованием MySQL Workbench.

Таким образом, в процессе работы были освоены все основные этапы проектирования базы данных — от анализа предметной области и логического моделирования до получения физической модели и развертывания её в СУБД.