ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

ассистент		Ю. В. Ветрова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 6

ПРОЕКТИРОВАНИЕ И ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ В СРЕДЕ MYSQL WORKBENCH

по курсу:

УПРАВЛЕНИЕ ДАННЫМИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4326		Г. С. Томчук
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

1 Название и цель выполнения работы

Вариант № 5: «База данных домашней библиотеки».

Цель работы: изучение процесса создания MySQL реляционной БД на основе соблюдения правил ограничения целостности данных.

2 Описание предметной области

Разрабатываемая база данных предназначена для ведения учёта библиотеки, содержащей книги и компьютерные диски. Система позволяет отслеживать перемещение этих объектов между пользователями, учитывать долги и предоставлять информацию о характеристиках каждого объекта.

В библиотеке могут находиться:

- Книги (учебная и художественная литература)
- Компьютерные диски (CD/DVD, MP3/MP4, диски с программами, энциклопедии, фильмы и др.)
- Пользователи (знакомые, которые могут брать книги и диски во временное пользование)

Основные функции системы:

- Учёт доступных книг и дисков
- Отслеживание выдачи объектов пользователям
- Отслеживание собственных долгов

3 EER-диаграмма в среде MySQL Workbench

На рис. 1 изображена EER-схема разработанной базы данных.

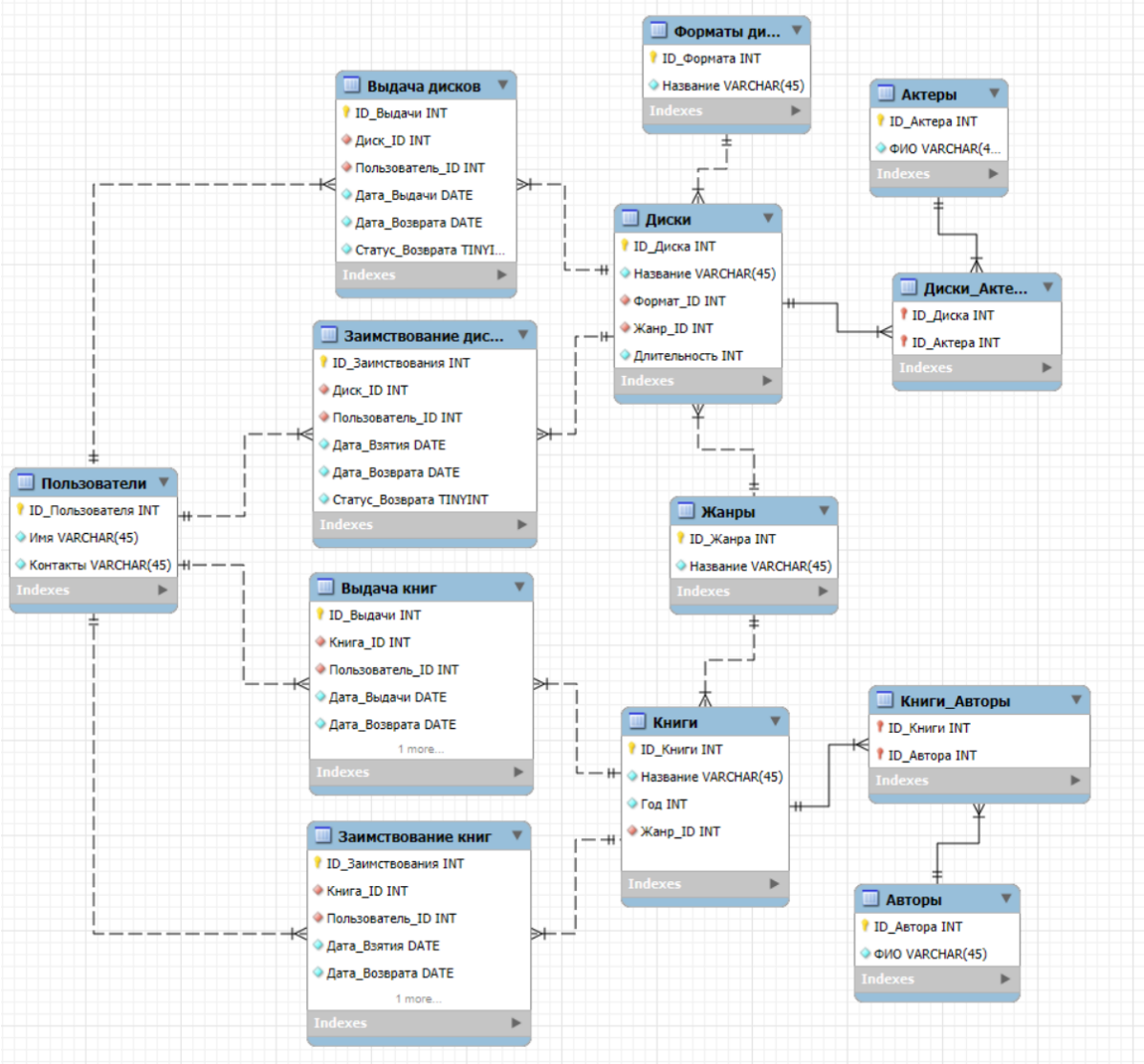

Рисунок 1 – EER-диаграмма БД

4 Физическая реализация таблиц БД на сервере

На рис. 2-4 изображена реализация таблиц БД на сервере MySQL. Программный код на создание БД приведен в Приложении А.

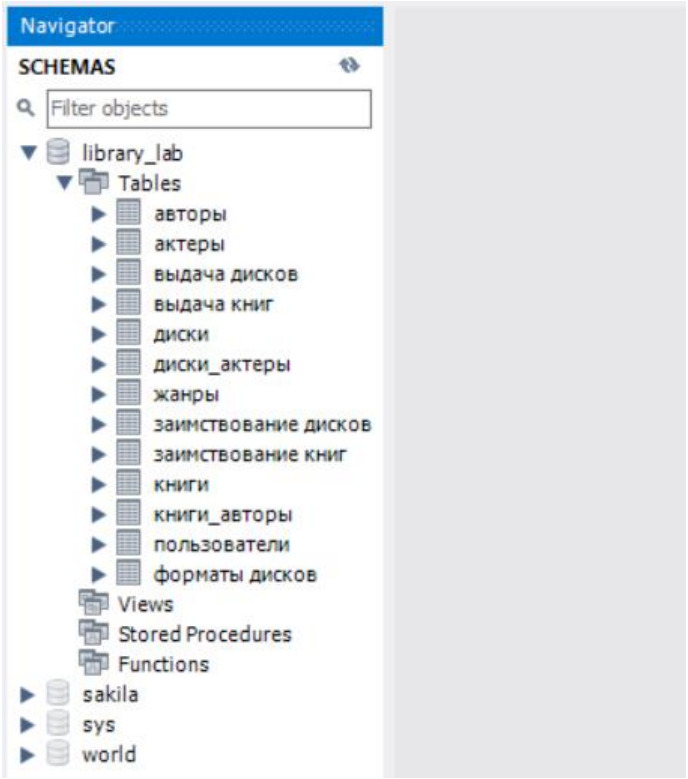

Рисунок 2 – Физическая реализация таблиц БД

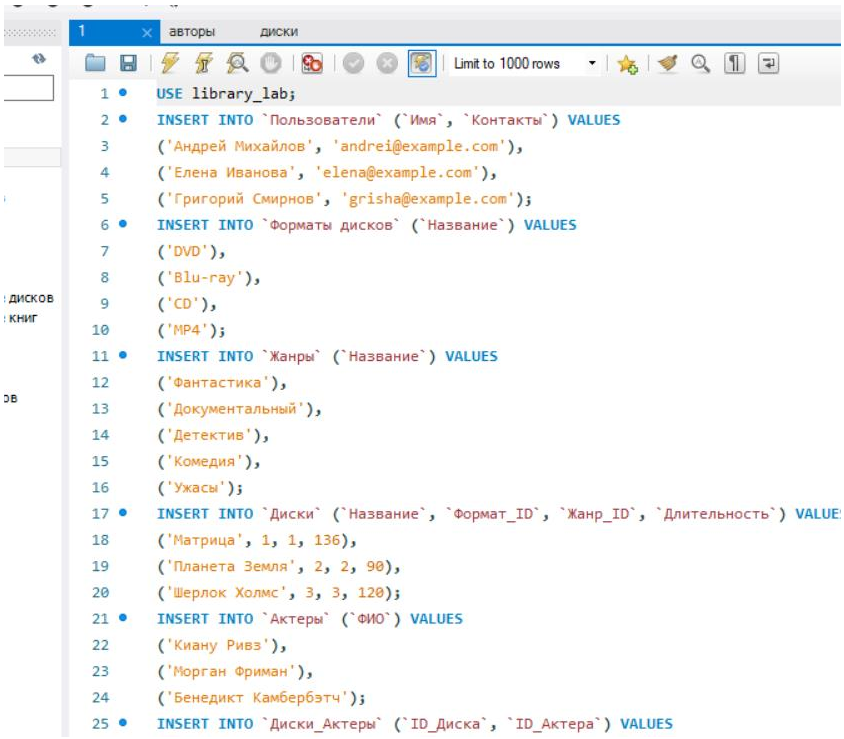

Рисунок 3 – Скрипт на заполнение БД

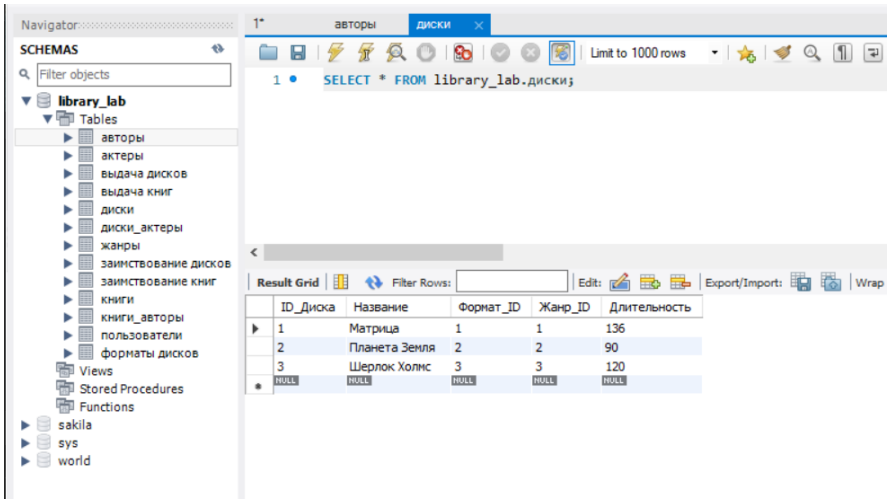

Рисунок 4 – Физическая реализация таблиц БД

5 Вывод

В ходе выполнения работы в среде MySQL Workbench была разработана база данных, предназначенная для учёта библиотеки, включающей книги и компьютерные диски. Реализована структура, обеспечивающая хранение информации об объектах, отслеживание их перемещений между пользователями, фиксацию задолженностей и доступ к характеристикам каждого экземпляра. Также была составлена EER-схема для наглядного отображения структуры базы данных и написан скрипт для её заполнения начальными данными.

ПРИЛОЖЕНИЕ А

```
-- MySQL Script generated by MySQL Workbench
-- Thu May 29 17:26:17 2025
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZE
RO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema library_lab
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema library_lab
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `library_lab` DEFAULT CHARACTER SET utf8 ;
USE `library_lab` ;

-- -----------------------------------------------------
-- Table `library_lab`.`Пользователи`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Пользователи` (
  `ID_Пользователя` INT NOT NULL AUTO_INCREMENT,
  `Имя` VARCHAR(45) NOT NULL,
  `Контакты` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_Пользователя`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Форматы дисков`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Форматы дисков` (
  `ID_Формата` INT NOT NULL AUTO_INCREMENT,
  `Название` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_Формата`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Жанры`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Жанры` (
  `ID_Жанра` INT NOT NULL AUTO_INCREMENT,
  `Название` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_Жанра`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Диски`
```

```
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Диски` (
  `ID_Диска` INT NOT NULL AUTO_INCREMENT,
  `Название` VARCHAR(45) NOT NULL,
  `Формат_ID` INT NOT NULL,
  `Жанр_ID` INT NOT NULL,
  `Длительность` INT NOT NULL,
  PRIMARY KEY (`ID_Диска`),
  INDEX `Формат_idx` (`Формат_ID` ASC) VISIBLE,
  INDEX `Жанр_idx` (`Жанр_ID` ASC) VISIBLE,
  CONSTRAINT `Формат`
    FOREIGN KEY (`Формат_ID`)
    REFERENCES `library_lab`.`Форматы дисков` (`ID_Формата`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Жанр`
    FOREIGN KEY (`Жанр_ID`)
    REFERENCES `library_lab`.`Жанры` (`ID_Жанра`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Выдача дисков`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Выдача дисков` (
  `ID_Выдачи` INT NOT NULL AUTO_INCREMENT,
  `Диск_ID` INT NOT NULL,
  `Пользователь_ID` INT NOT NULL,
  `Дата_Выдачи` DATE NOT NULL,
  `Дата_Возврата` DATE NOT NULL,
  `Статус_Возврата` TINYINT NOT NULL,
  PRIMARY KEY (`ID_Выдачи`),
  INDEX `Пользователь_idx` (`Пользователь_ID` ASC) VISIBLE,
  INDEX `Диск_idx` (`Диск_ID` ASC) VISIBLE,
  CONSTRAINT `Пользователь`
    FOREIGN KEY (`Пользователь_ID`)
    REFERENCES `library_lab`.`Пользователи` (`ID_Пользователя`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Диск`
    FOREIGN KEY (`Диск_ID`)
    REFERENCES `library_lab`.`Диски` (`ID_Диска`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Книги`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Книги` (
  `ID_Книги` INT NOT NULL AUTO_INCREMENT,
  `Название` VARCHAR(45) NOT NULL,
  `Год` INT NOT NULL,
  `Жанр_ID` INT NOT NULL,
```

```
  PRIMARY KEY (`ID_Книги`),
  INDEX `Жанр_idx` (`Жанр_ID` ASC) VISIBLE,
  CONSTRAINT `Жанр`
    FOREIGN KEY (`Жанр_ID`)
    REFERENCES `library_lab`.`Жанры` (`ID_Жанра`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Актеры`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Актеры` (
  `ID_Актера` INT NOT NULL AUTO_INCREMENT,
  `ФИО` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_Актера`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Диски_Актеры`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Диски_Актеры` (
  `ID_Диска` INT NOT NULL,
  `ID_Актера` INT NOT NULL,
  PRIMARY KEY (`ID_Диска`, `ID_Актера`),
  INDEX `Актер_idx` (`ID_Актера` ASC) VISIBLE,
  CONSTRAINT `Диск`
    FOREIGN KEY (`ID_Диска`)
    REFERENCES `library_lab`.`Диски` (`ID_Диска`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Актер`
    FOREIGN KEY (`ID_Актера`)
    REFERENCES `library_lab`.`Актеры` (`ID_Актера`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Авторы`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Авторы` (
  `ID_Автора` INT NOT NULL AUTO_INCREMENT,
  `ФИО` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`ID_Автора`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Книги_Авторы`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Книги_Авторы` (
  `ID_Книги` INT NOT NULL,
  `ID_Автора` INT NOT NULL,
```

```
  PRIMARY KEY (`ID_Книги`, `ID_Автора`),
  INDEX `Автор_idx` (`ID_Автора` ASC) VISIBLE,
  CONSTRAINT `Книга`
    FOREIGN KEY (`ID_Книги`)
    REFERENCES `library_lab`.`Книги` (`ID_Книги`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Автор`
    FOREIGN KEY (`ID_Автора`)
    REFERENCES `library_lab`.`Авторы` (`ID_Автора`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Заимствование дисков`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Заимствование дисков` (
  `ID_Заимствования` INT NOT NULL AUTO_INCREMENT,
  `Диск_ID` INT NOT NULL,
  `Пользователь_ID` INT NOT NULL,
  `Дата_Взятия` DATE NOT NULL,
  `Дата_Возврата` DATE NOT NULL,
  `Статус_Возврата` TINYINT NOT NULL,
  PRIMARY KEY (`ID_Заимствования`),
  INDEX `Пользователь_idx` (`Пользователь_ID` ASC) VISIBLE,
  INDEX `Диск_idx` (`Диск_ID` ASC) VISIBLE,
  CONSTRAINT `Пользователь`
    FOREIGN KEY (`Пользователь_ID`)
    REFERENCES `library_lab`.`Пользователи` (`ID_Пользователя`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Диск`
    FOREIGN KEY (`Диск_ID`)
    REFERENCES `library_lab`.`Диски` (`ID_Диска`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Выдача книг`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Выдача книг` (
  `ID_Выдачи` INT NOT NULL AUTO_INCREMENT,
  `Книга_ID` INT NOT NULL,
  `Пользователь_ID` INT NOT NULL,
  `Дата_Выдачи` DATE NOT NULL,
  `Дата_Возврата` DATE NOT NULL,
  `Статус_Возврата` TINYINT NOT NULL,
  PRIMARY KEY (`ID_Выдачи`),
  INDEX `Пользователь_idx` (`Пользователь_ID` ASC) VISIBLE,
  INDEX `Книга_idx` (`Книга_ID` ASC) VISIBLE,
  CONSTRAINT `Пользователь`
    FOREIGN KEY (`Пользователь_ID`)
    REFERENCES `library_lab`.`Пользователи` (`ID_Пользователя`)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Книга`
    FOREIGN KEY (`Книга_ID`)
    REFERENCES `library_lab`.`Книги` (`ID_Книги`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `library_lab`.`Заимствование книг`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `library_lab`.`Заимствование книг` (
  `ID_Заимствования` INT NOT NULL AUTO_INCREMENT,
  `Книга_ID` INT NOT NULL,
  `Пользователь_ID` INT NOT NULL,
  `Дата_Взятия` DATE NOT NULL,
  `Дата_Возврата` DATE NOT NULL,
  `Статус_Возврата` TINYINT NOT NULL,
  PRIMARY KEY (`ID_Заимствования`),
  INDEX `Пользователь_idx` (`Пользователь_ID` ASC) VISIBLE,
  INDEX `Книга_idx` (`Книга_ID` ASC) VISIBLE,
  CONSTRAINT `Пользователь`
    FOREIGN KEY (`Пользователь_ID`)
    REFERENCES `library_lab`.`Пользователи` (`ID_Пользователя`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Книга`
    FOREIGN KEY (`Книга_ID`)
    REFERENCES `library_lab`.`Книги` (`ID_Книги`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```