|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра практической и прикладной информатики (ППИ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Разработка баз данных»

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-16-19, Мурадов Н.Н.* | (подпись) | |
| Преподаватель | *Володина А.М.* | (подпись) | |
| Отчет представлен | «\_\_\_» декабря 2021г. | |  | |

Москва 2021 г.

**Содержание**

[Введение 3](#_Toc93195921)

[Практическое задание 1 3](#_Toc93195922)

[Практическое задание 2 11](#_Toc93195923)

[Практическое задание 3 21](#_Toc93195924)

[Практическое задание 4 22](#_Toc93195925)

[Практическое задание 5 23](#_Toc93195926)

[Практическое задание 6 35](#_Toc93195927)

[Практическое задание 7 41](#_Toc93195928)

[Вывод 62](#_Toc93195929)

[Список использованных источников и литературы: 62](#_Toc93195930)

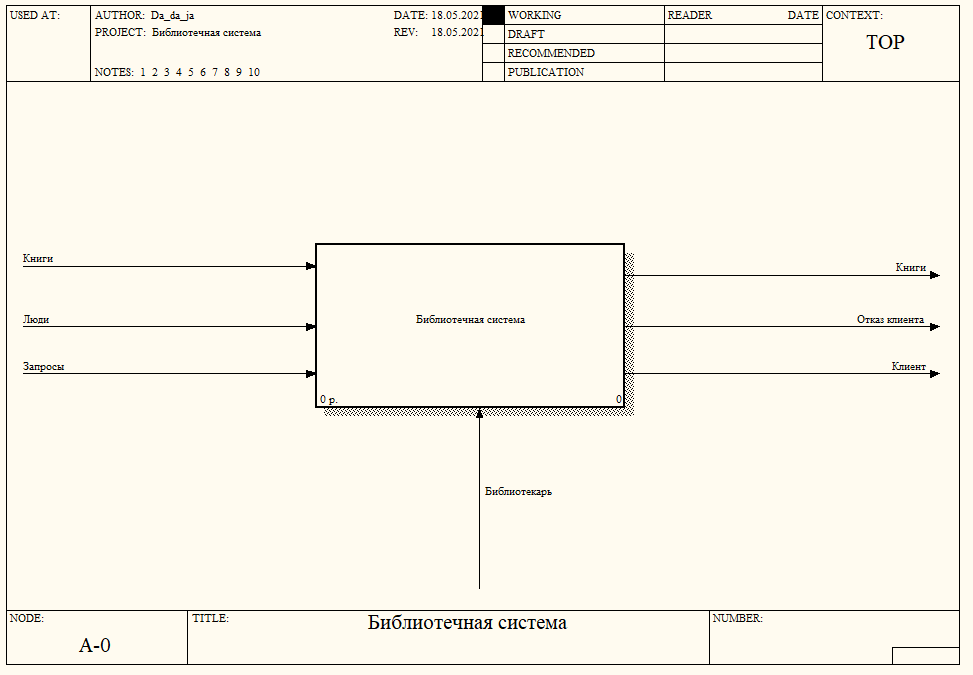
Введение

В качестве темы для создания базы данных было выбрано «Библиотека». В процессе работы данная тема приняла направленность базы данных для будущего приложения с целью упрощения работы. В виду этого была создана модель, ориентированная на десктопное приложение российской государственной библиотеки.  
В ней были описаны основные элементы, которые должны быть задействованы в подобной информационной системе.

# Практическое задание 1

*Модель IDEF0 по предметной области «Библиотека».*

**Предметная область «Библиотека»** — это учреждение культуры, организующее сбор, хранение и общественное пользование произведениями печати и другими документами.

****

*IDEF0. Контекстная диаграмма.*

IDEF0 — методология функционального моделирования, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность

Основной блок – Библиотечная система

Входной информацией системы является:

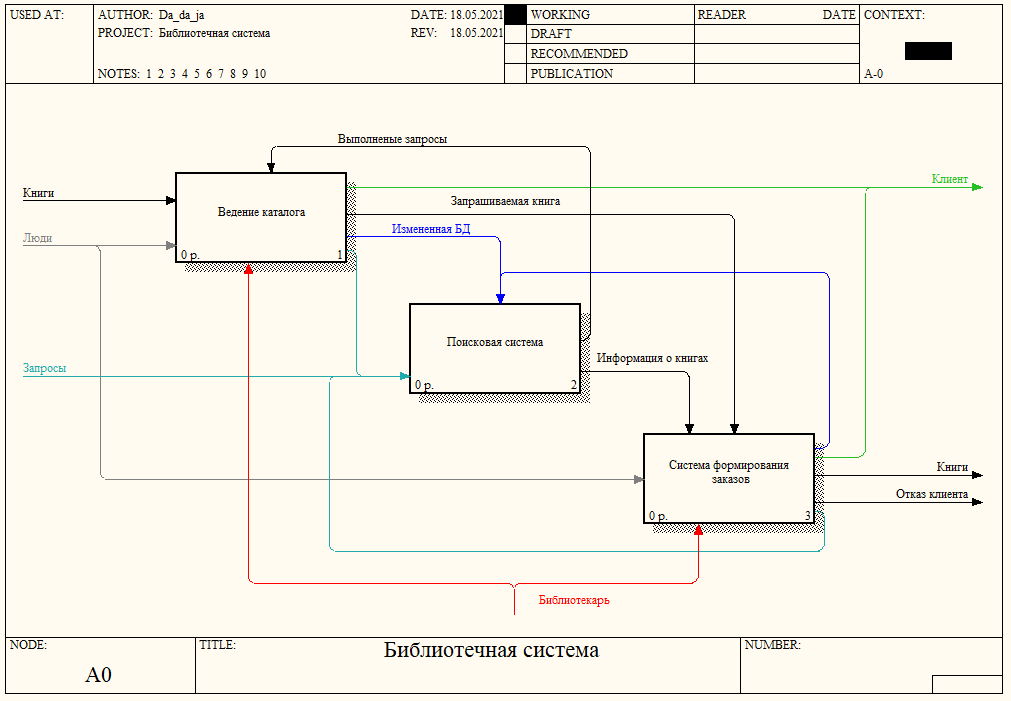
* **Книги;**
* **Люди;**
* **Запросы.**

Выходной информацией системы является:

* **Книги;**
* **Отказ клиента;**
* **Клиент.**

Механической информацией системы является:

* **Библиотекарь.**

****

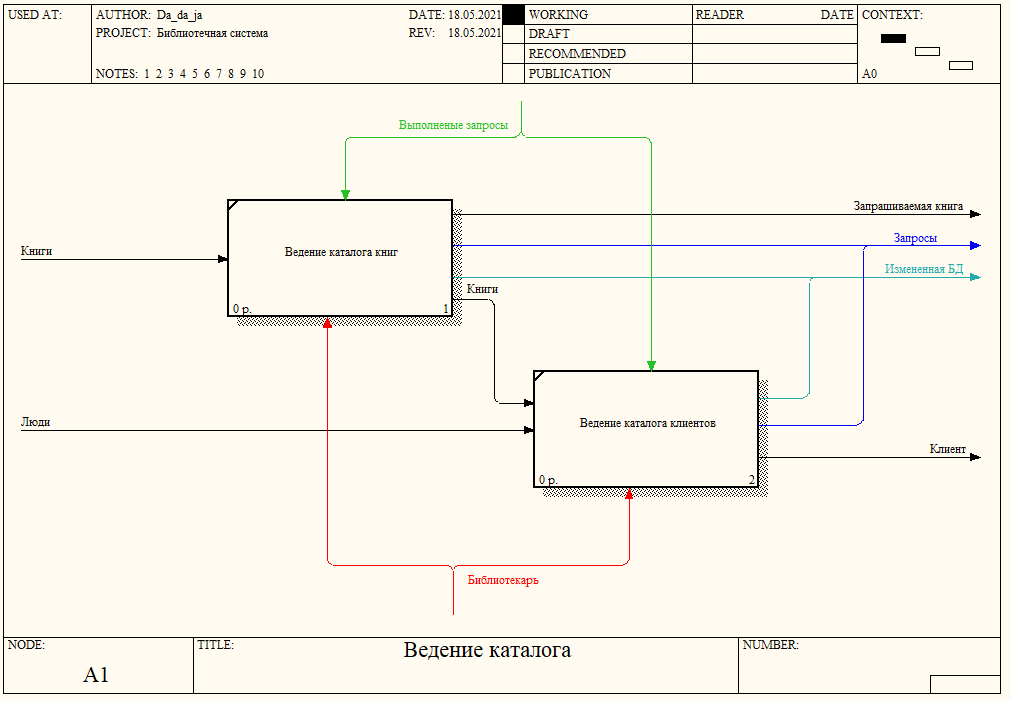
*Декомпозиция контекстной диаграммы.*

Мы декомпозировали общий блок «Библиотечная система» на связанные между собой элементы. В нашем случае делится на 3 основных этапа:

* **Ведение каталога;**
* **Поисковая система;**
* **Система формирования заказов.**

Блок «Ведение каталога» мы декомпозируем еще на 2 этапа:

* **Ведение каталога книг;**
* **Ведение каталога клиентов.**

****

Входной информацией блока "Ведение каталога книг" является "Книга";

Управляющей информацией блока "Ведение каталога книг" является "Выполненные запросы";

Механической информацией блока "Ведение каталога книг" является "Библиотекарь";

Выходной информацией блока "Ведение каталога книг" являются "Запрашиваемая книга", "Запросы", "Измененная БД", "Книги".

Входной информацией блока "Ведение каталога клиентов" являются "Люди", "Книги";

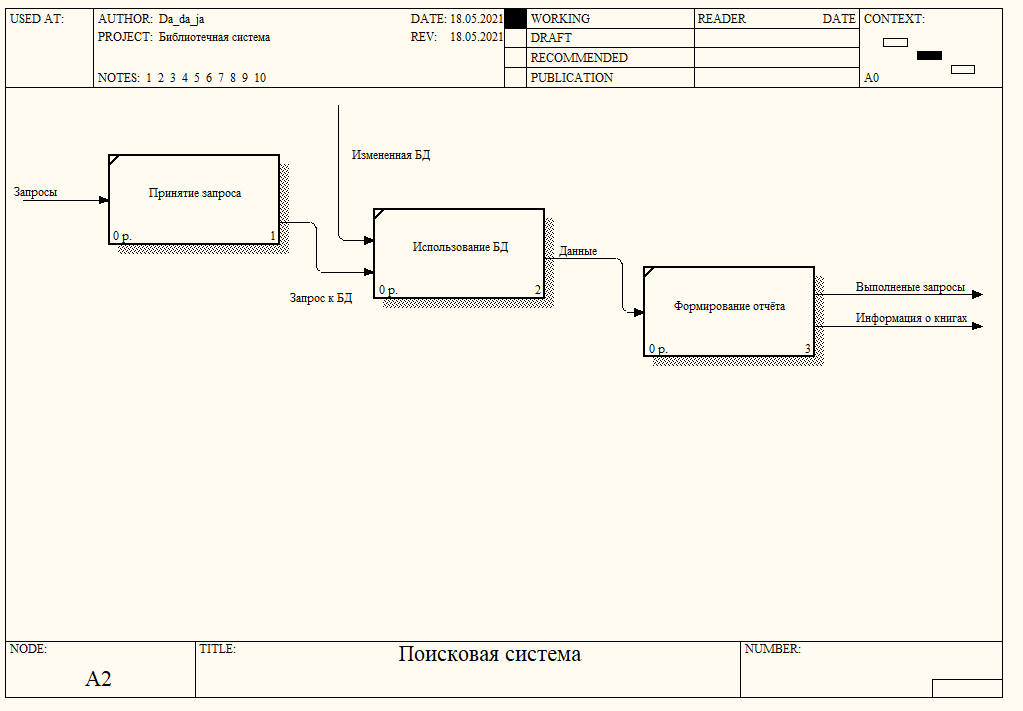
Управляющей информацией блока "Ведение каталога клиентов" является "Выполненные запросы";

Механической информацией блока "Ведение каталога клиентов" является "Библиотекарь"

Выходной информацией блока "Ведение каталога клиентов" является "Измененная БД", "Запросы", "Клиент".

Блок «Поисковая система» мы декомпозируем еще на 3 этапа:

* **Принятие запроса;**
* **Использование БД;**
* **Формирование отчёта.**

****

Входной информацией блока "Принятие запроса" является "Запросы";

Выходной информацией блока "Принятие запроса" являются "Запрос к БД", "Измененная БД".

Входной информацией блока "Использование БД" являются "Запрос к БД", "Измененная БД";

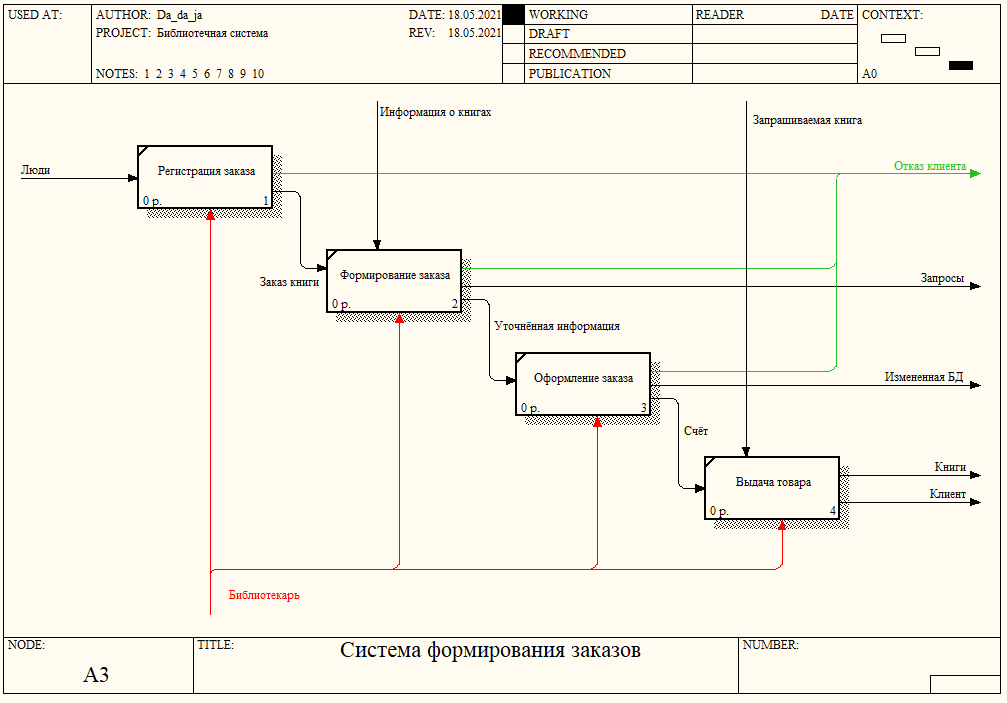
Выходной информацией блока "Использование БД" является "Данные".

Входной информацией блока "Формирование отчёта" является "Данные";

Выходной информацией блока "Формирование отчёта" являются "Выполненные запросы" и "Информация о книгах".

Блок «Система формирования заказов» мы декомпозируем еще на 4 этапа:

* **Регистрация заказа;**
* **Формирование заказа;**
* **Оформление заказа;**
* **Выдача товара.**

****

Входной информацией блока "Регистрация заказа" является "Люди";

Механической информацией блока "Регистрация заказа" является "Библиотекарь";

Выходной информацией блока "Регистрация заказа" являются "Отказ клиента", "Заказ книги".

Входной информацией блока "Формирование заказа" является "Заказ книги";

Управляющей информацией блока "Формирование заказа" является: "Информация о книгах";

Механической информацией блока "Формирование заказа" является: "Библиотекарь";

Выходной информацией блока "Формирование заказа" являются "Отказ клиента", "Запросы", "Уточнённая информация".

Входной информацией блока "Оформление заказа" является "Уточнённая информация";

Механической информацией блока "Оформление заказа" является: "Библиотекарь";

Выходной информацией блока "Оформление заказа" являются: "Отказ клиента", "Измененная БД", "Счёт".

Входной информацией блока "Выдача товара" является "Счёт";

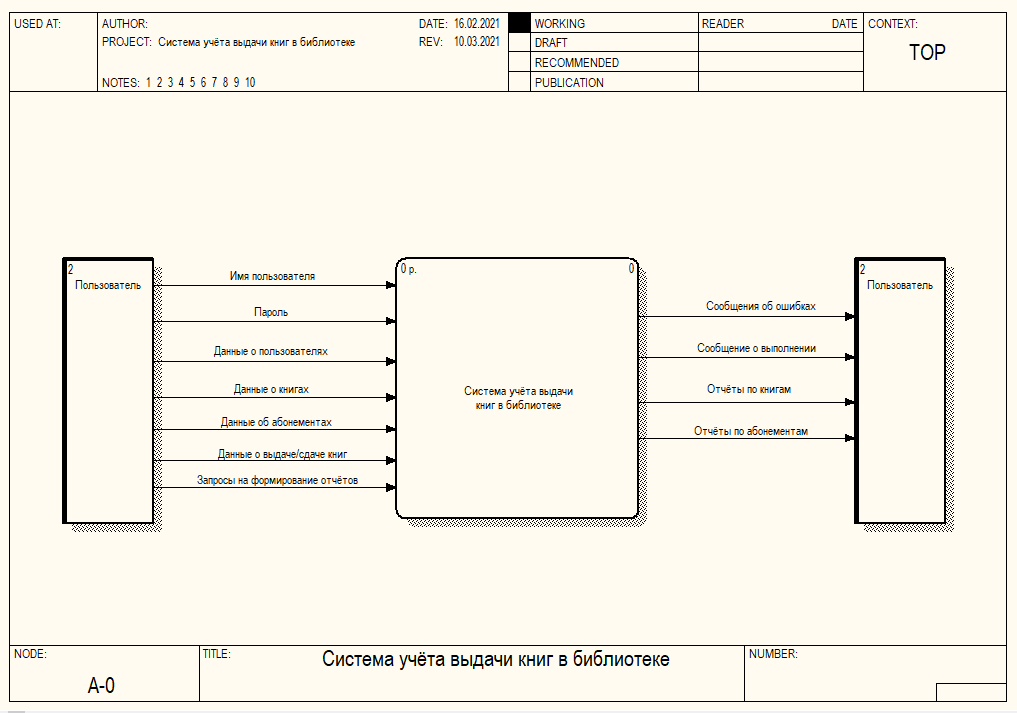
Управляющей информацией блока "Выдача товара" является: "Запрашиваемая книга";

Механической информацией блока "Выдача товара" является: "Библиотекарь";

Выходной информацией блока "Выдача товара" являются "Книги", "Клиент".

# Практическое задание 2

*Модель DFD по предметной области «Библиотека».*

**

*DFD. Контекстная диаграмма.*

На данной модели отображается основной процесс (сама система в целом) и ее связи с внешней средой (внешними сущностями). Это взаимодействие показывается через потоки данных.

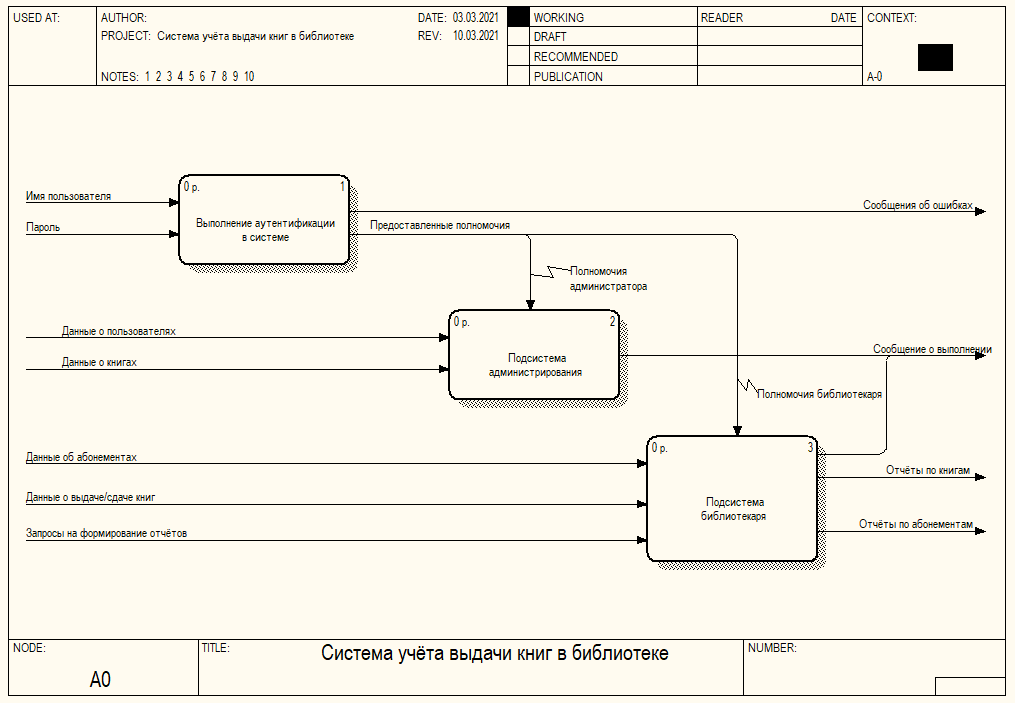
**Внешние сущности** изображают входы в систему и/или выходы из нее. У нас внешние сущности это: Клиент, Поставщик, Бюджет мастерской, Помещение для мастерской и Подмастерье.

**Стрелки (потоки данных).** Стрелки описывают движение объектов из одной части системы в другую.

**Хранилище данных.** В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое. У нас хранилища данных это: Знания, Постоянные клиенты, Интернет, Справки по изготовлению тех или иных изделий, Информация о заказах и Книга отзывов.

Блок «Система учёта выдачи книг в библиотеке» мы декомпозируем еще на 3 этапа:

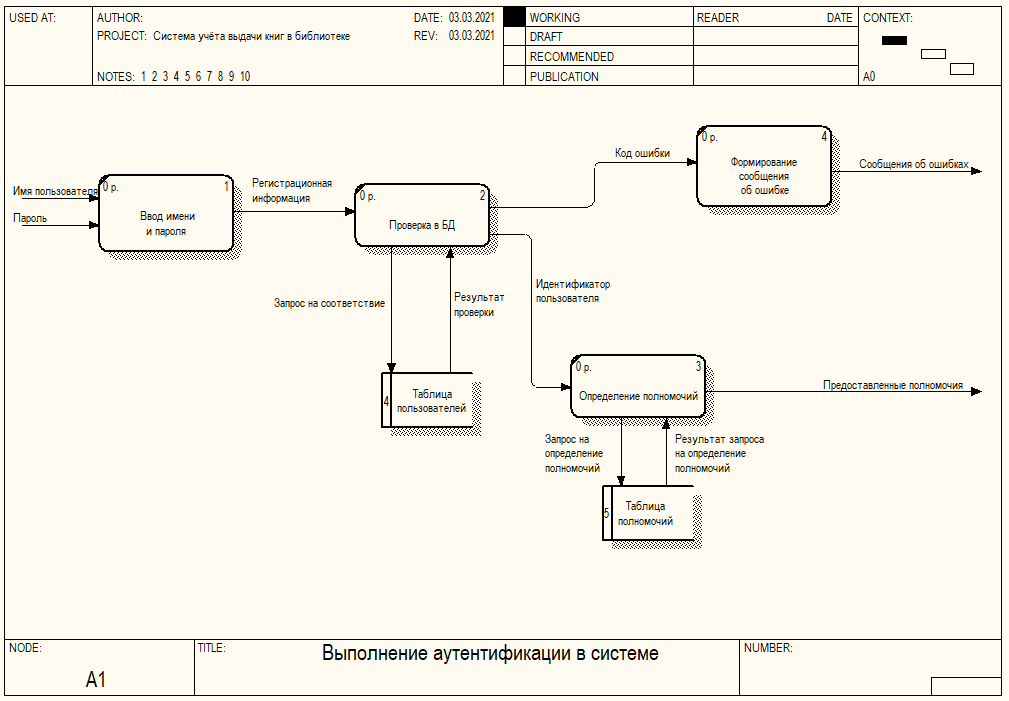
* **Выполнение аутентификации в системе;**
* **Подсистема администрирования;**
* **Подсистема библиотекаря.**

****

*Декомпозиция контекстной диаграммы*

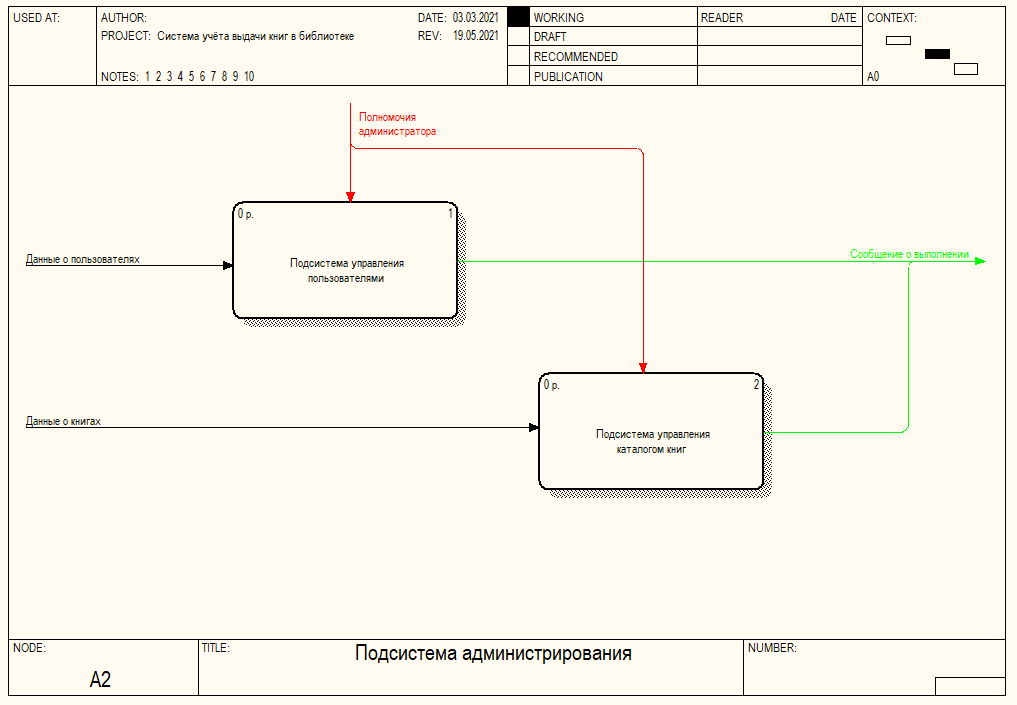
Блок «Выполнение аутентификации в системе» мы декомпозируем еще на 4 этапа:

* **Ввод имени и пароля;**
* **Проверка в БД;**
* **Определение полномочий;**
* **Формирование сообщения об ошибке.**

****

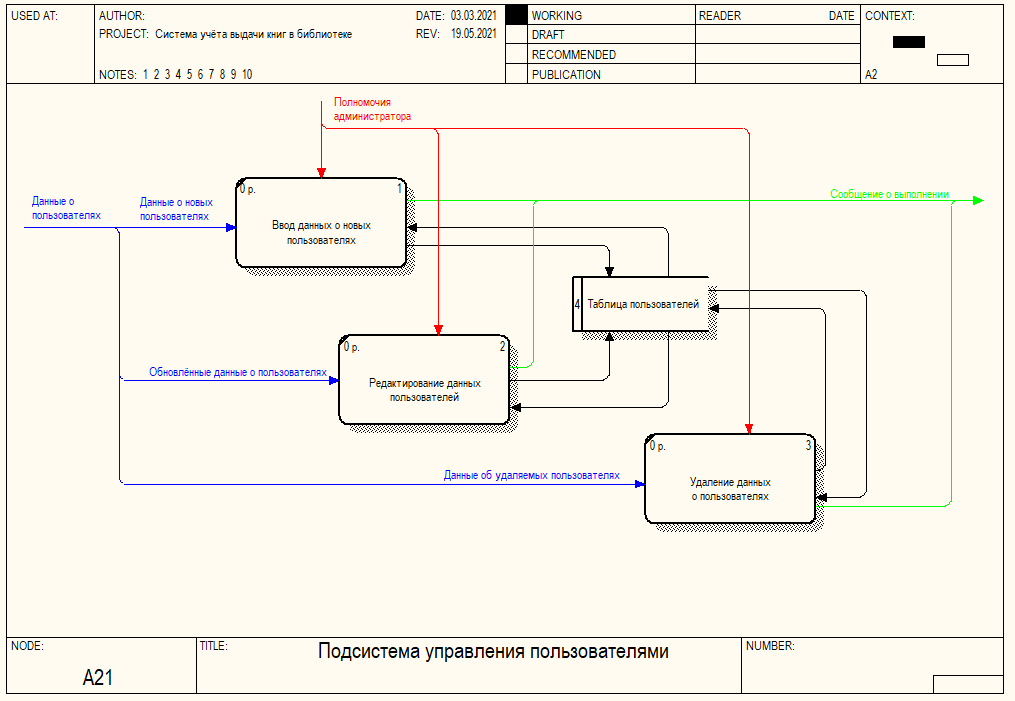
Блок «Подсистема администрирования» мы декомпозируем еще на 2 этапа:

* **Подсистема управления пользователями;**
* **Подсистема управления каталогом книг.**

****

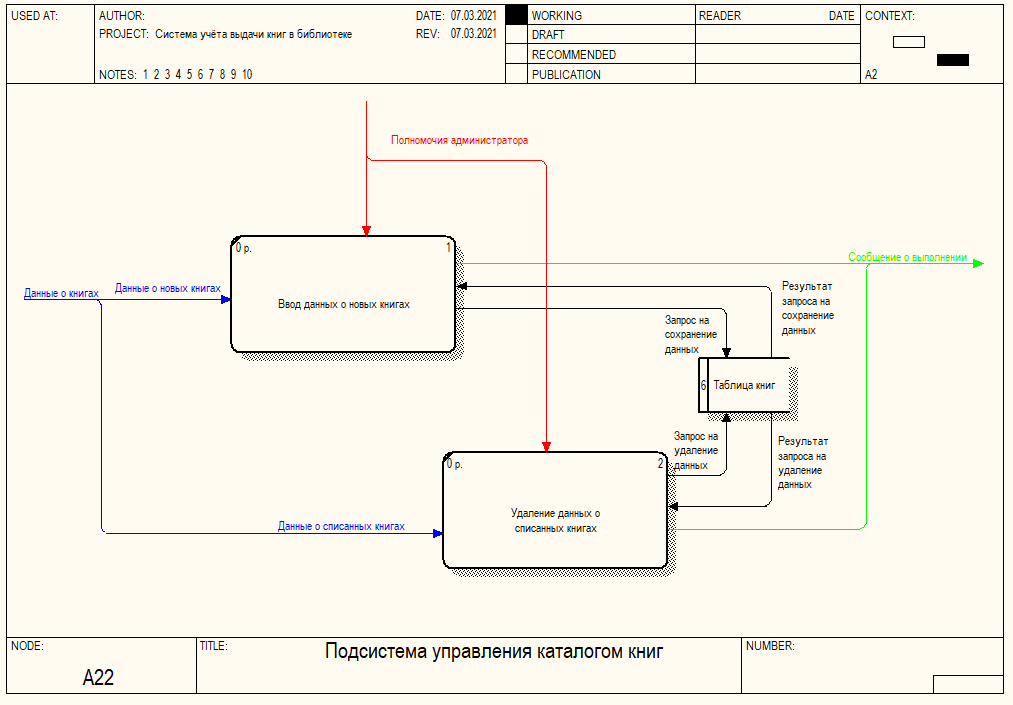
Блок «Подсистема управления пользователями» мы декомпозируем еще на 3 этапа:

* **Ввод данных о новых пользователях;**
* **Редактирование данных пользователей;**
* **Удаление данных о пользователях.**

****

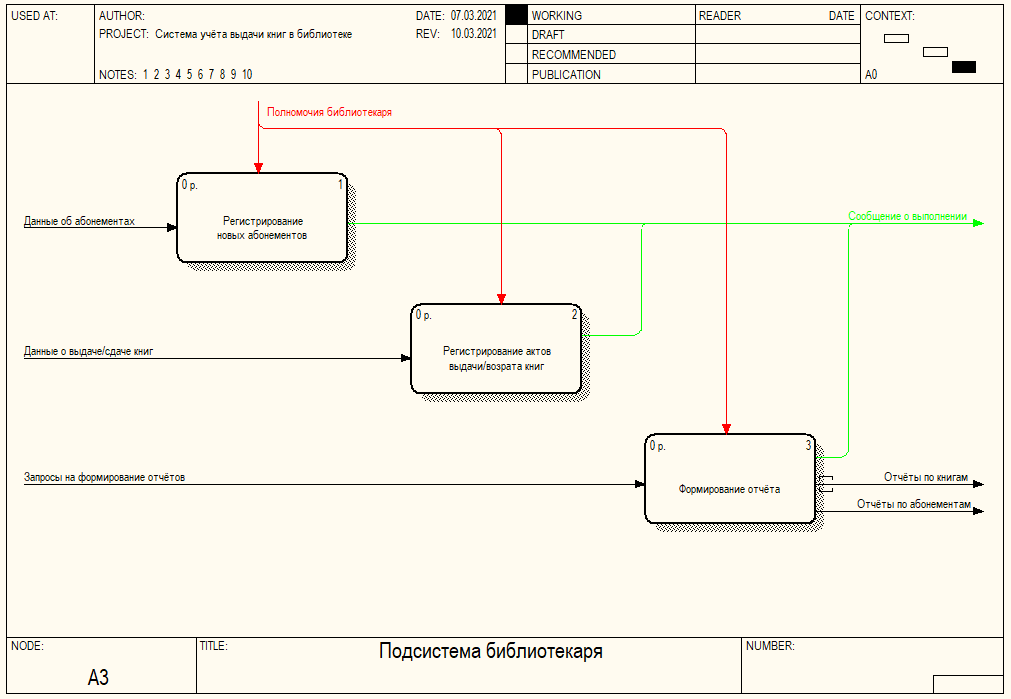
Блок «Подсистема управления каталогом книг» мы декомпозируем еще на 2 этапа:

* **Ввод данных о новых книгах;**
* **Удаление данных о списанных книгах.**

****

Блок «Подсистема библиотекаря» мы декомпозируем еще на 3 этапа:

* **Регистрирование новых абонементов;**
* **Регистрирование актов выдачи/возврата книг;**
* **Формирование отчёта.**

****

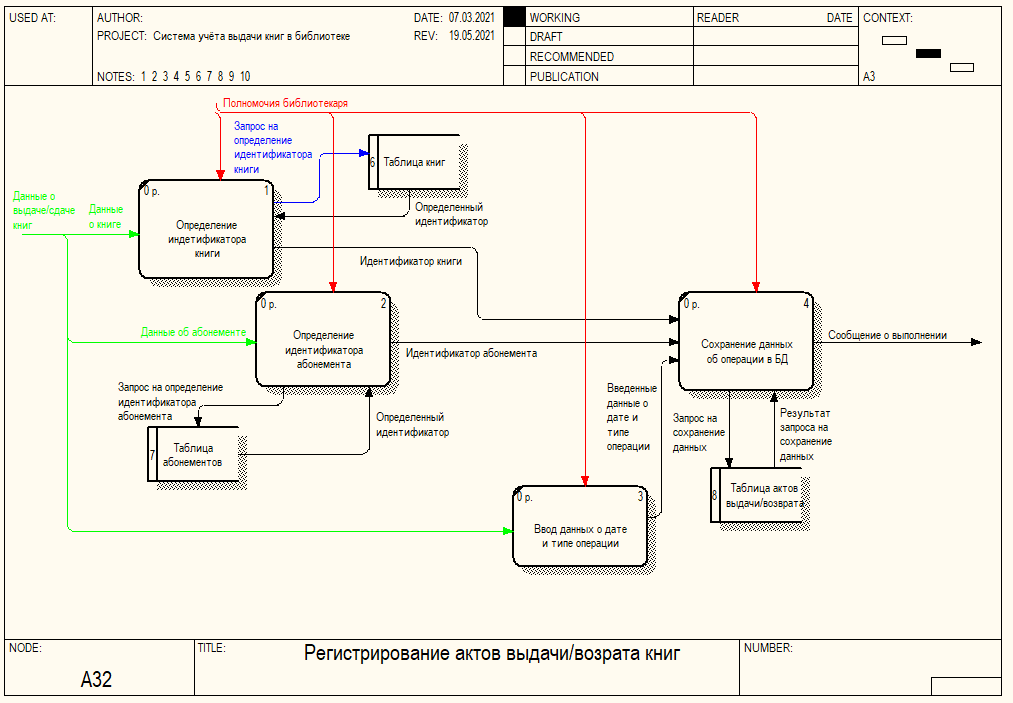
Блок «Регистрирование новых абонементов» мы декомпозируем еще на 3 этапа:

* **Заполнение полей формы ввода данных абонемента;**
* **Проверка корректности введенных данных;**
* **Сохранение данных о новом абонементе в БД.**

****

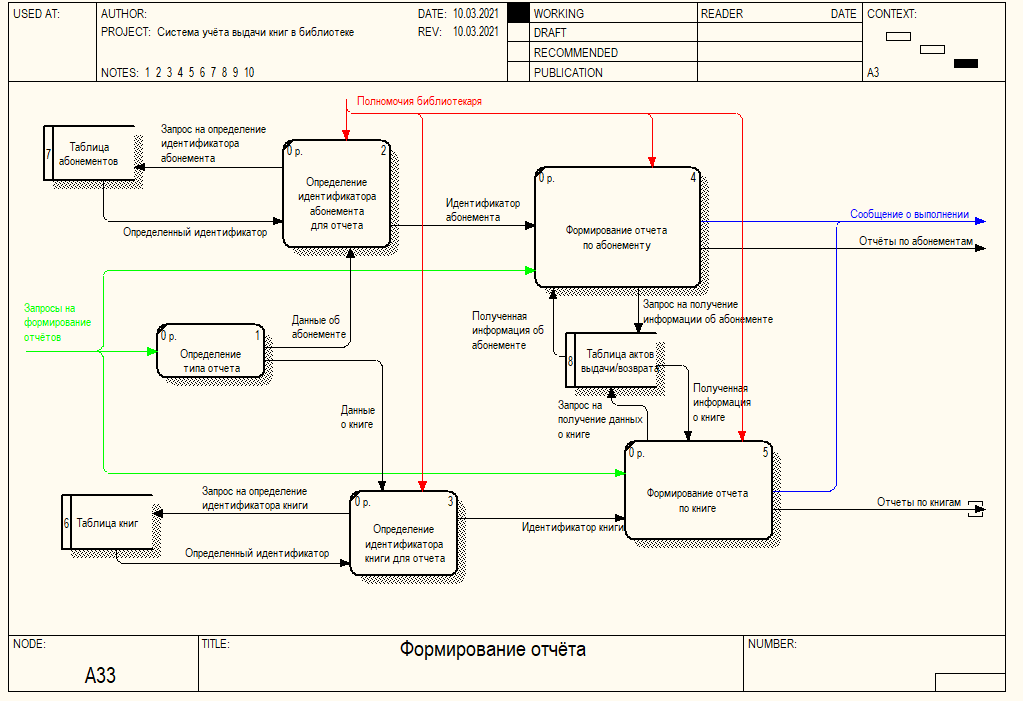
Блок «Регистрирование актов выдачи/возврата книг» мы декомпозируем еще на 4 этапа:

* **Определение идентификатора книги;**
* **Определение идентификатора абонемента;**
* **Ввод данных о дате и типе операции;**
* **Сохранение данных об операции в БД.**

****

Блок «Формирование отчёта» мы декомпозируем еще на 5 этапа:

* **Определение типа отчета;**
* **Определение идентификатора абонемента для отчета;**
* **Определение идентификатора книги для отчета;**
* **Формирование отчета по абонементу;**
* **Формирование отчета по книге.**

****

# Практическое задание 3

**Логическая модель базы данных по теме «****Библиотека»**

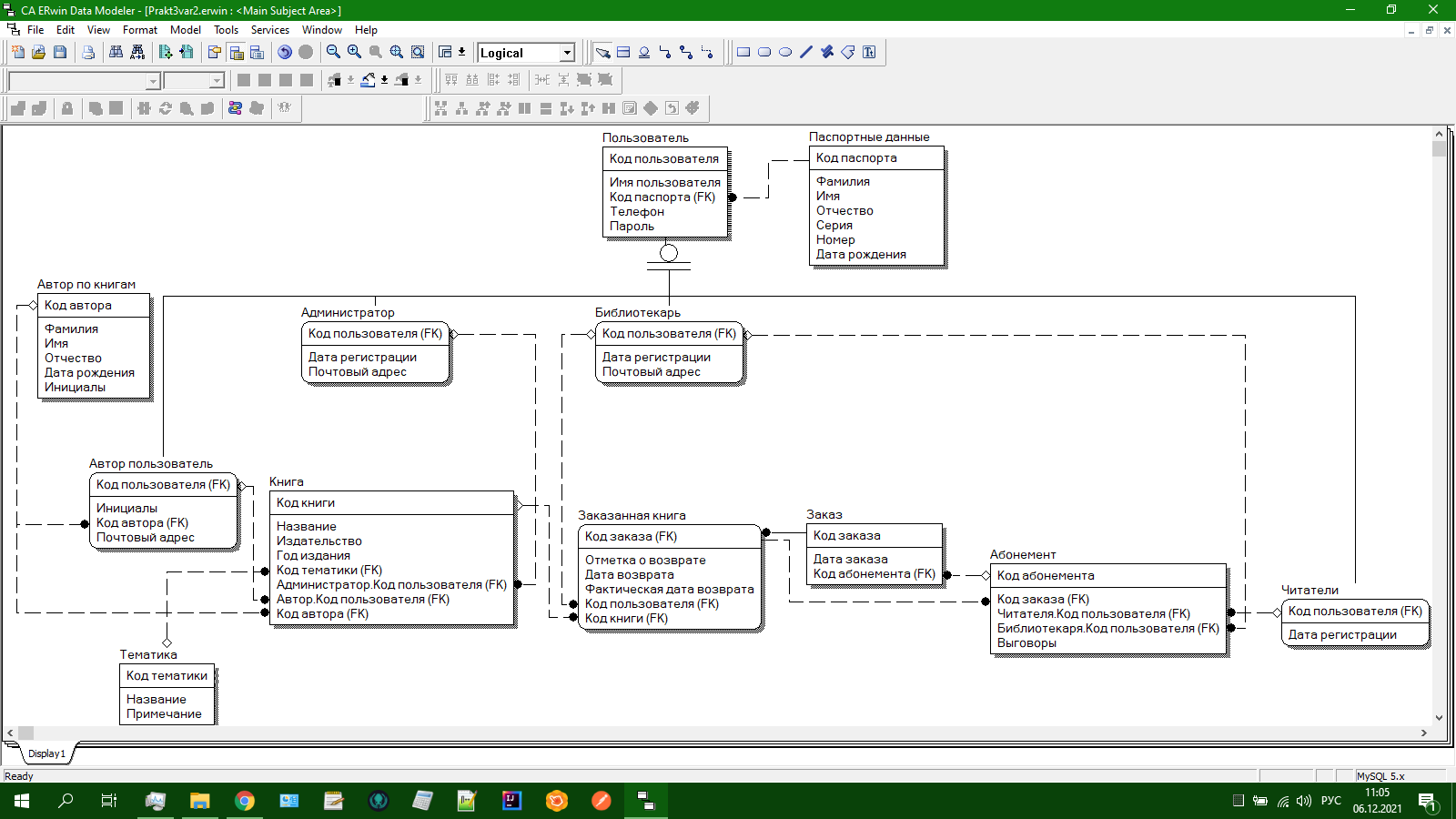


Рис. 3 ‒ Логическая модель данных на тему «Библиотека»

**Описание**

В данной модели можно увидеть модель базы данных bib, которая состоит из таблиц:

Паспортные\_данные ‒ содержит основные паспортные данные о клиенте, оформляющем заказ  
Пользователь ‒ содержит информацию о клиентах в рамках информационной системы  
Автор\_по\_книгам ‒ информация о авторах не зарегистрированных в системе  
Автор\_пользователь ‒ содержит информацию о авторах зарегистрированных в системе  
Администратор ‒ содержит информацию о администраторах  
Библиотекарь ‒ хранит информацию о библиотекарях  
Тематика ‒ содержит информацию об существующих тематиках  
Читатель ‒ информация о читателе  
Абонемент ‒ содержит информацию о читателе, а также о выговорах  
Книга ‒ вся информация о книге  
Заказанная\_книга ‒ хранит информацию о исполнении обязательств читателем и книге  
Заказ ‒ является сборником всех заказов в системе

# Практическое задание 4

**Физическая модель базы данных по теме «Библиотека»**

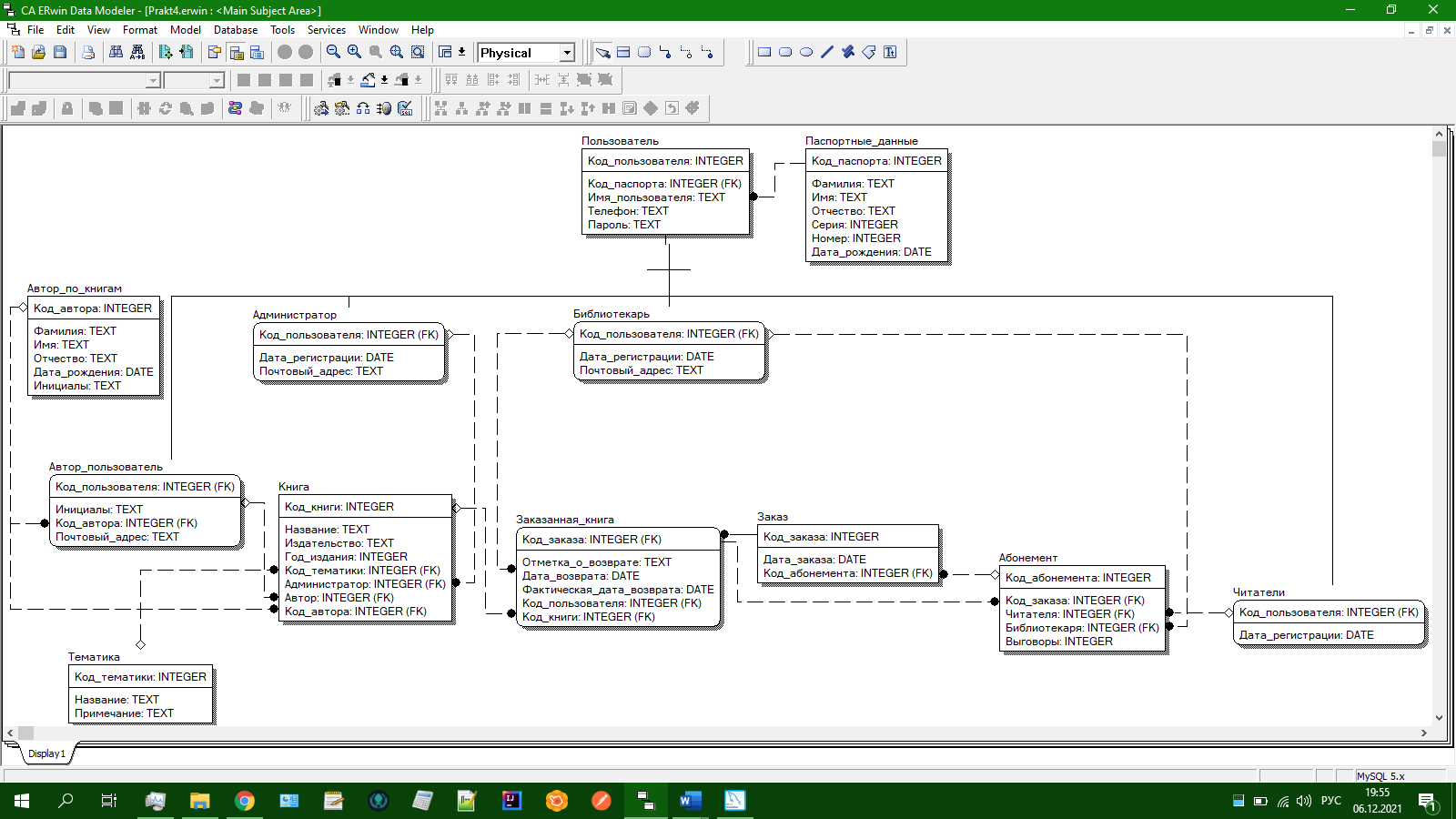


Рис. 4 ‒ Физическая модель данных на тему «Библиотека»

# Практическое задание 5

**Создание своей базы данных в MySQL CommandLine**

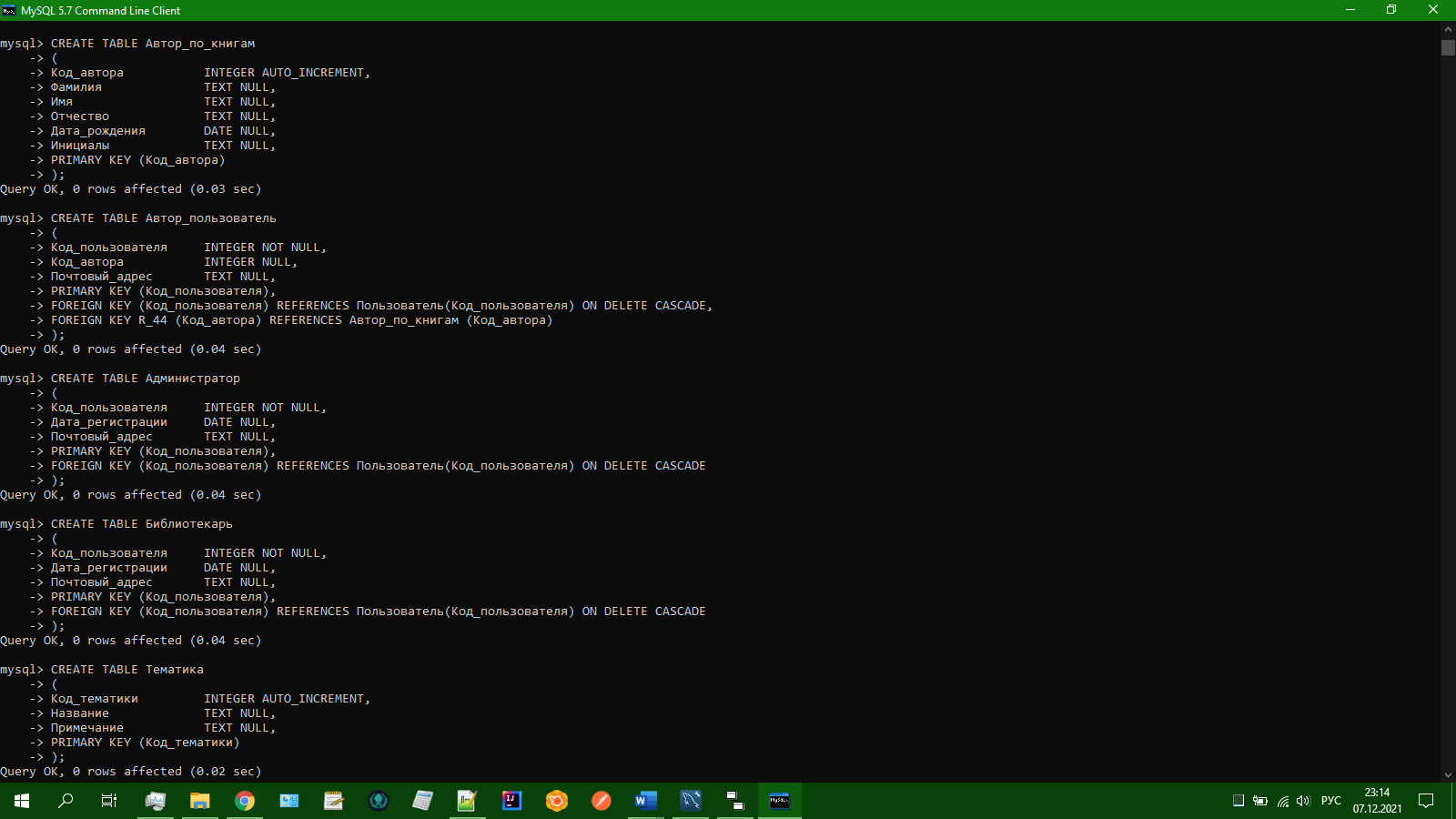
Рис. 5 Создание БД bib, проверка и выбор данной БД, создание таблиц

Рис. 6 Создание таблиц



Рис. 7 Создание таблиц

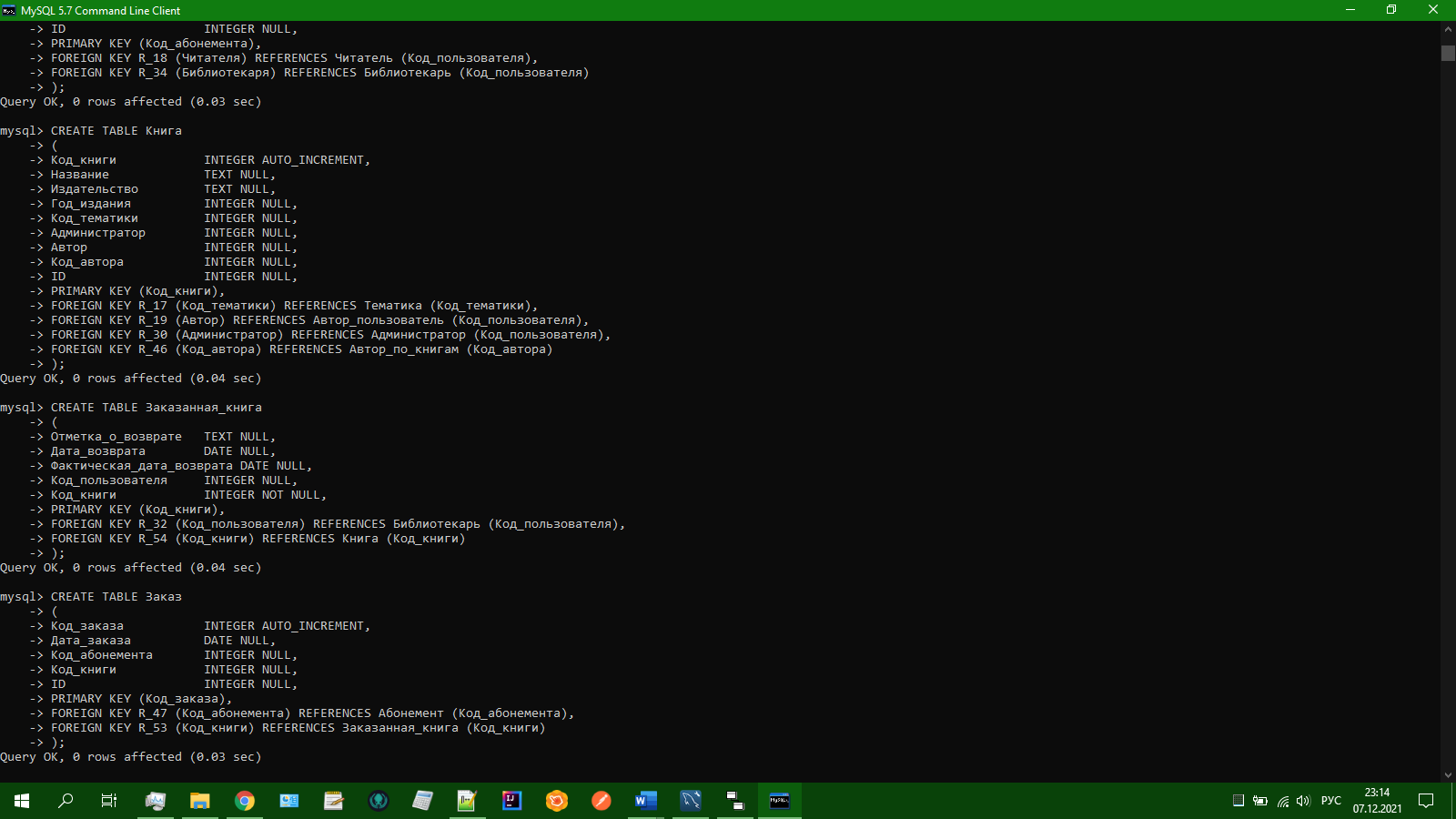


Рис. 8 Создание таблиц

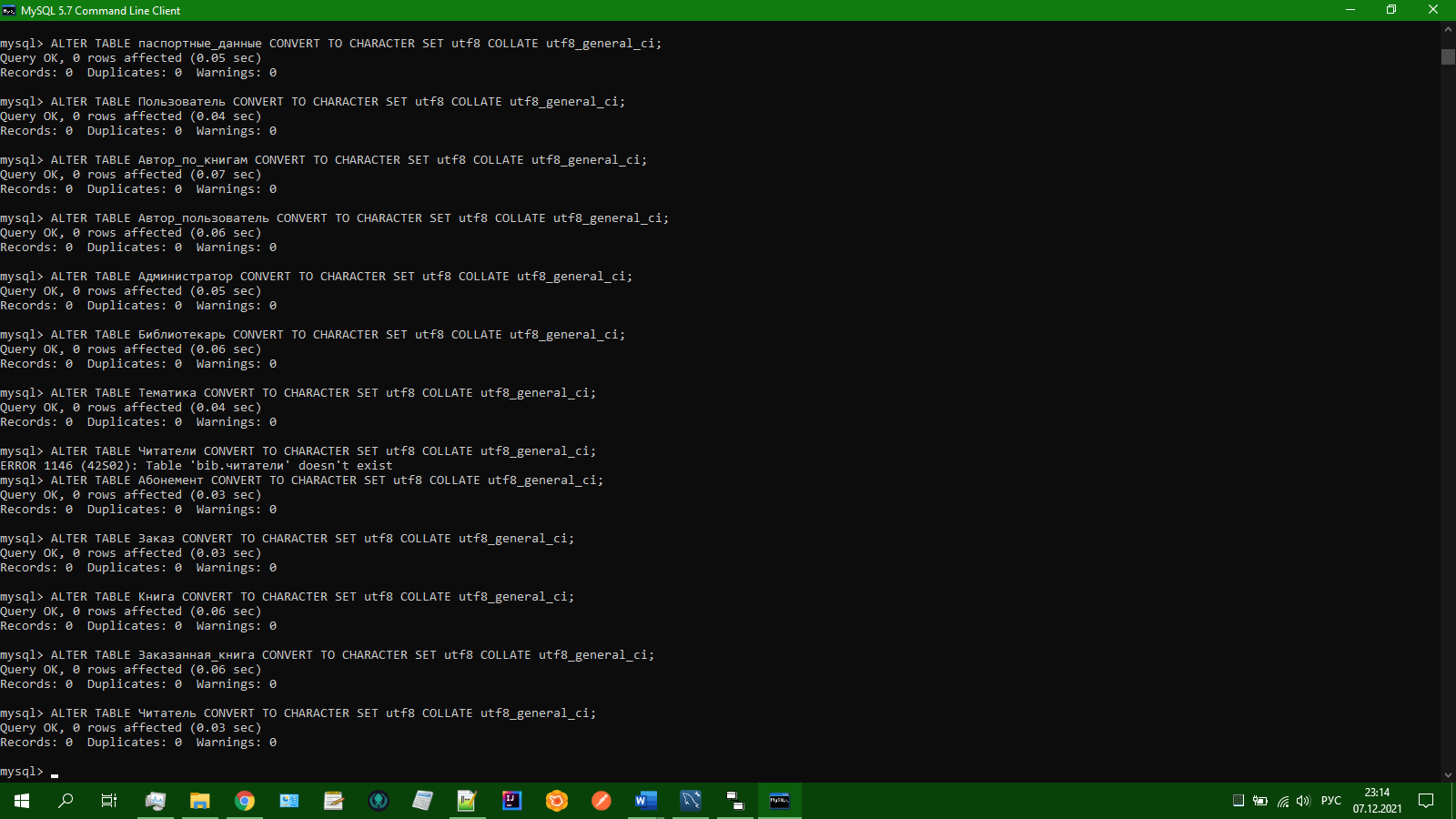


Рис. 9 Применение кодировки UTF-8

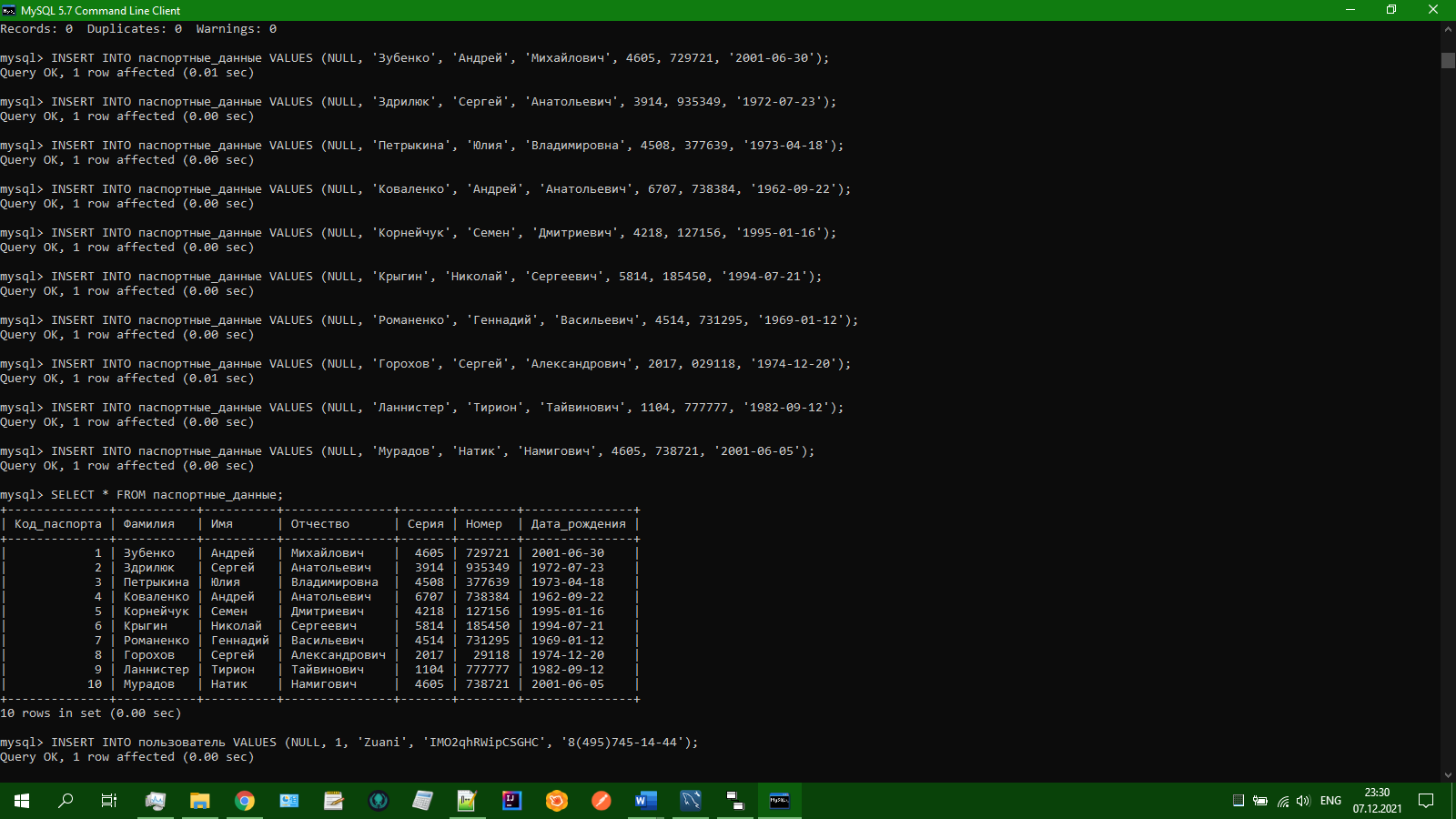


Рис. 10 Заполнение таблицы паспортные\_данные и проверка

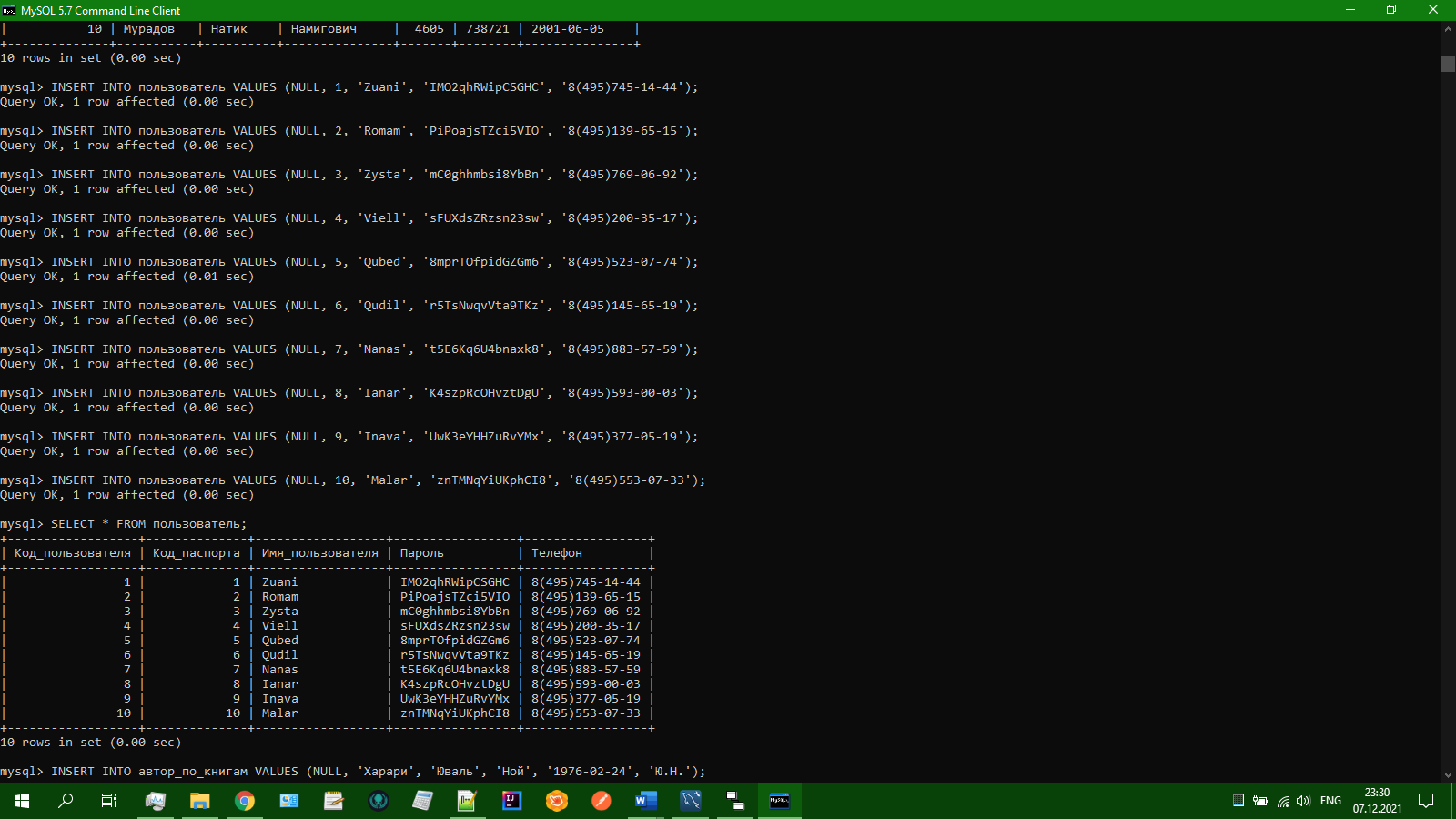


Рис. 11 Заполнение таблицы пользователь и проверка

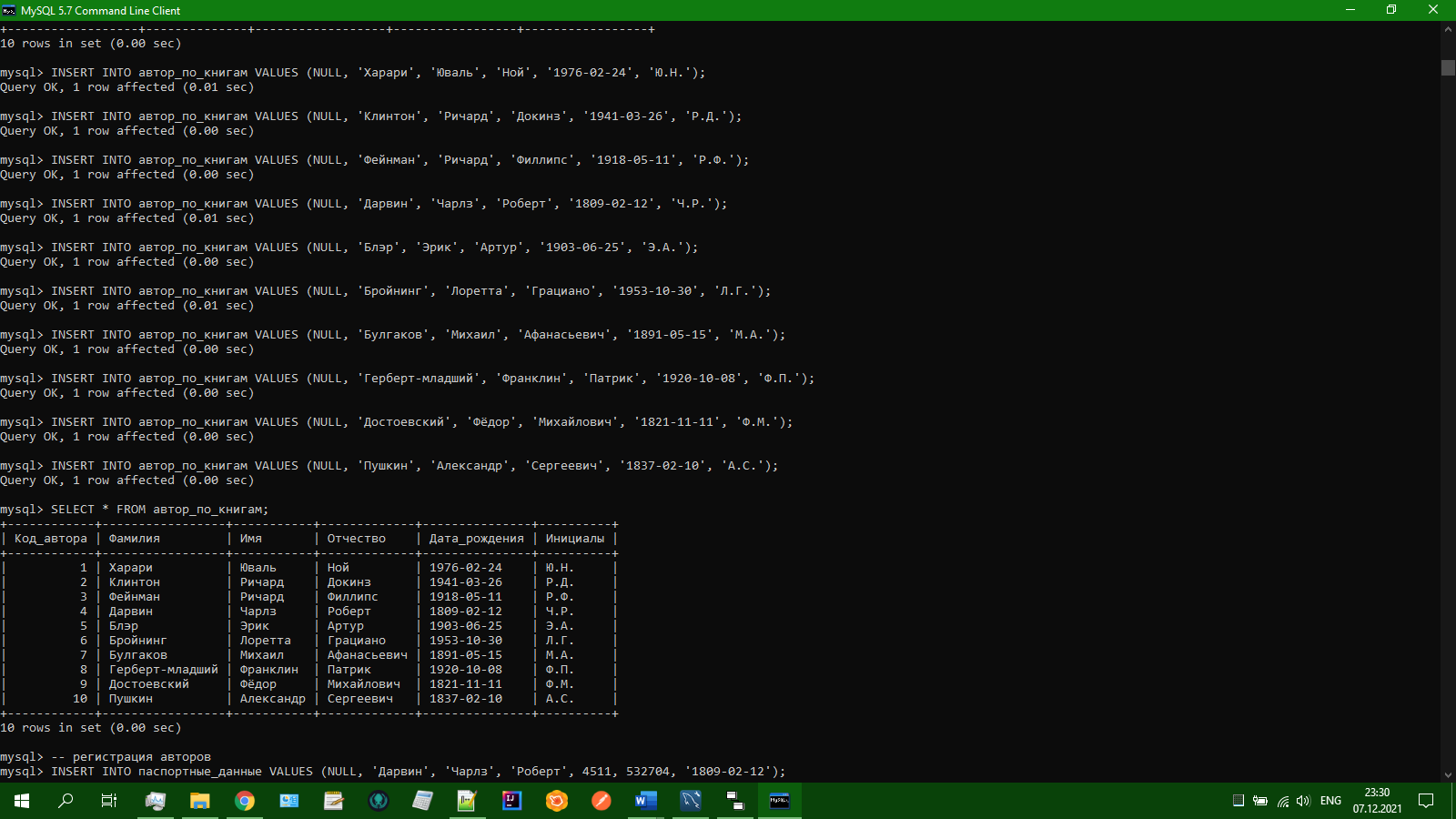


Рис. 12 Заполнение таблицы автор\_по\_книгам и проверка

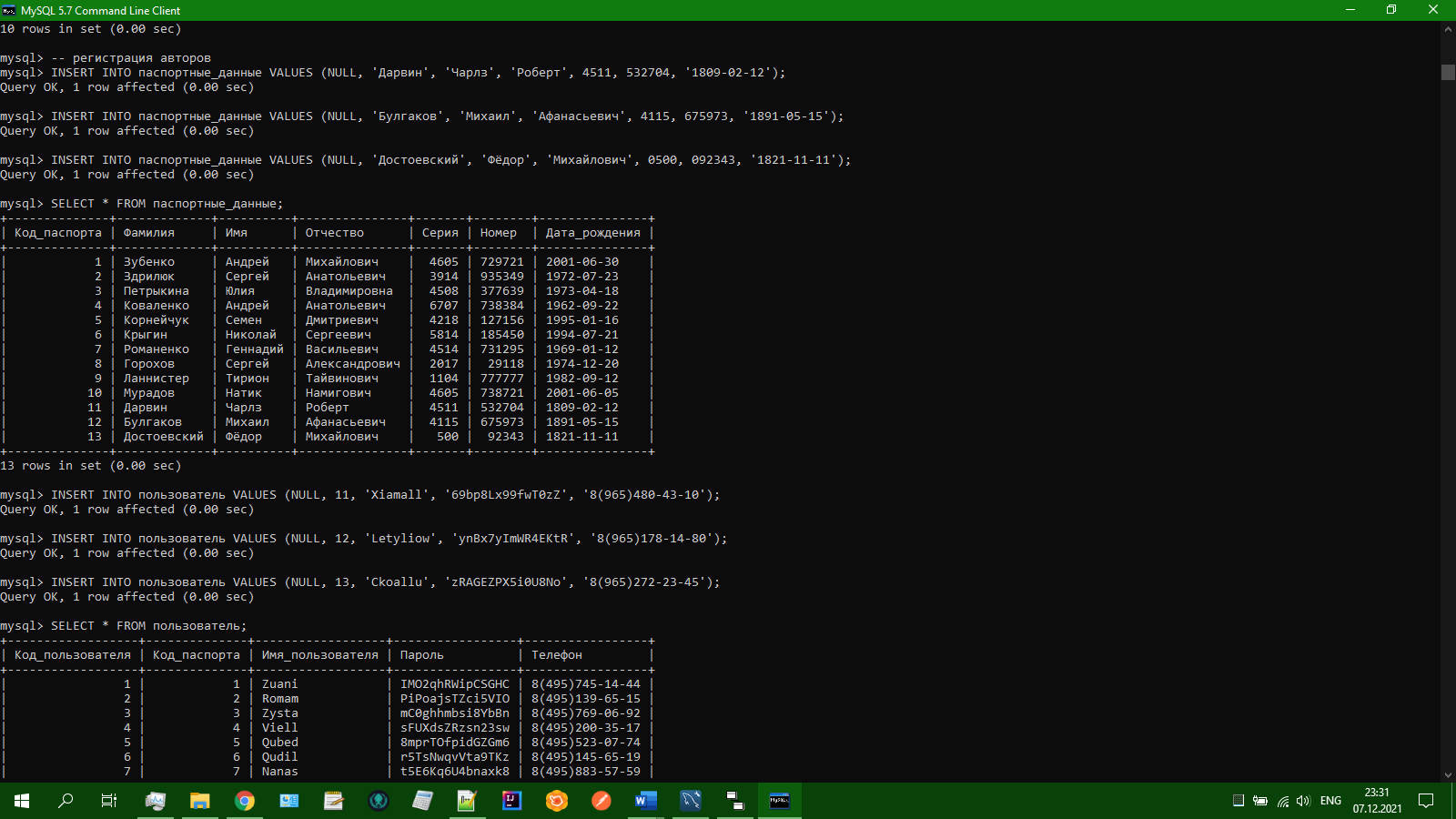


Рис. 13 Дополнительное заполнение таблиц и проверка

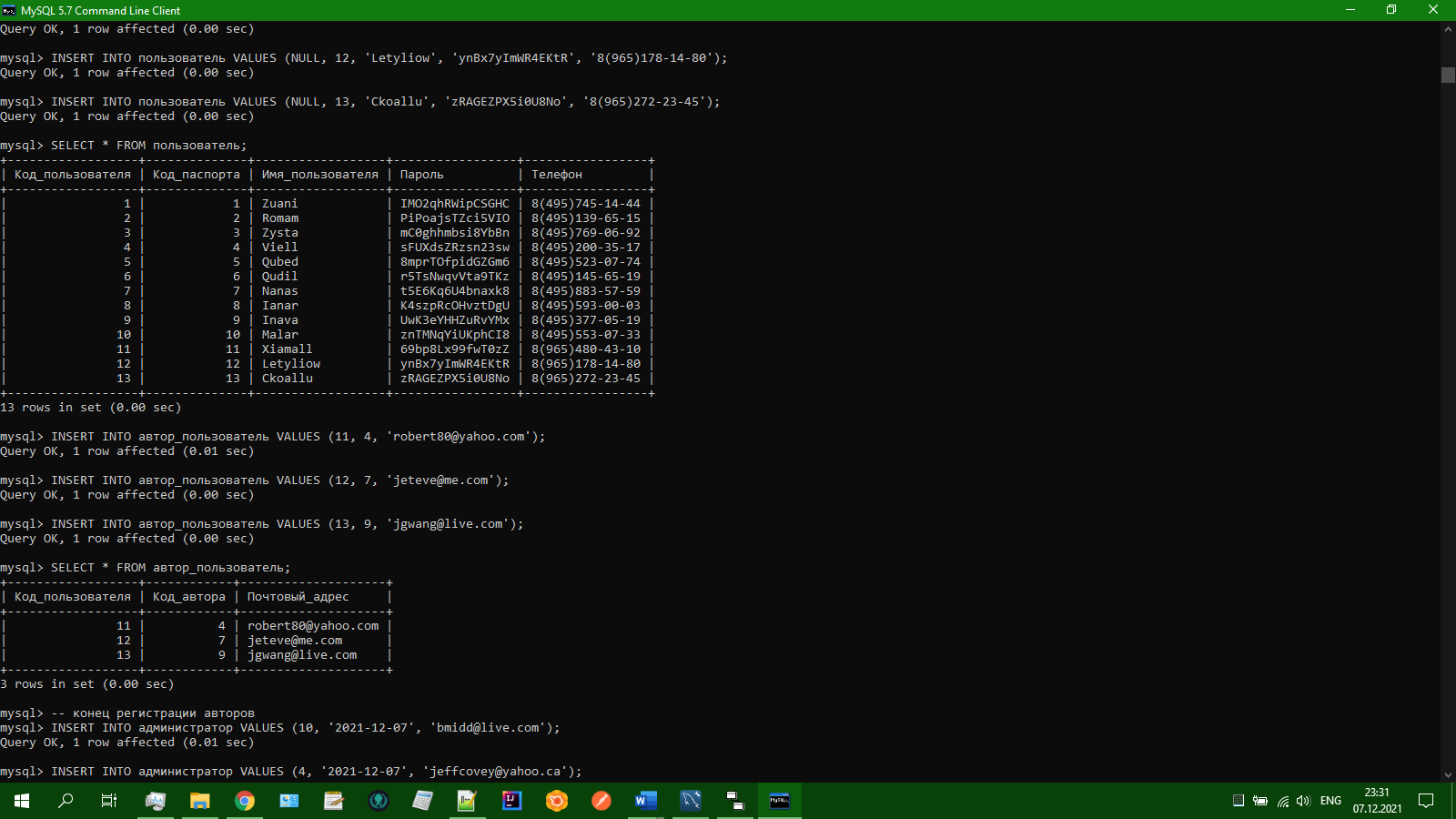


Рис. 14 Продолжение проверки, заполнение таблицы автор\_пользователь и проверка

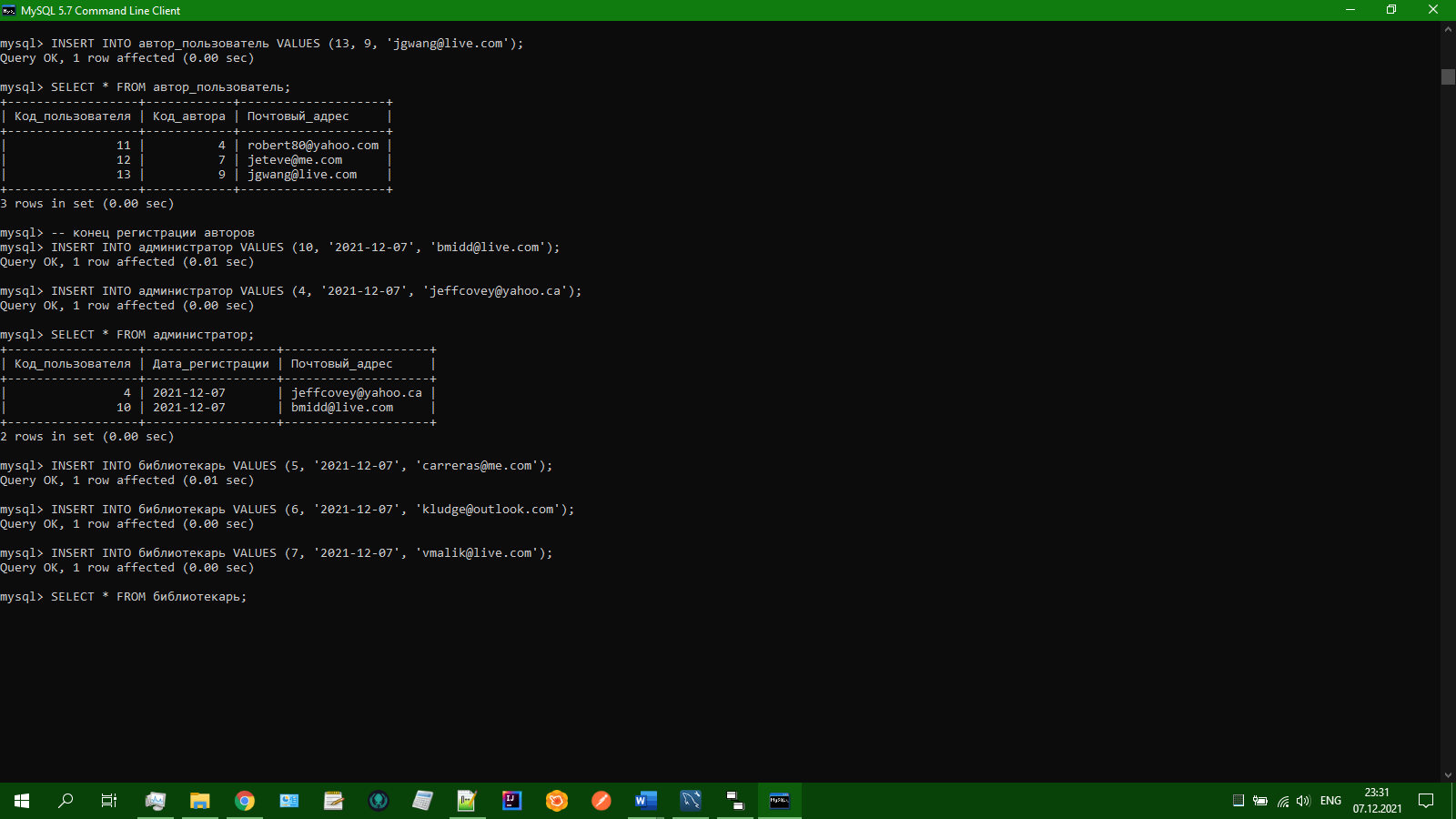


Рис. 15 Заполнение таблиц администратор и библиотекарь, проверка

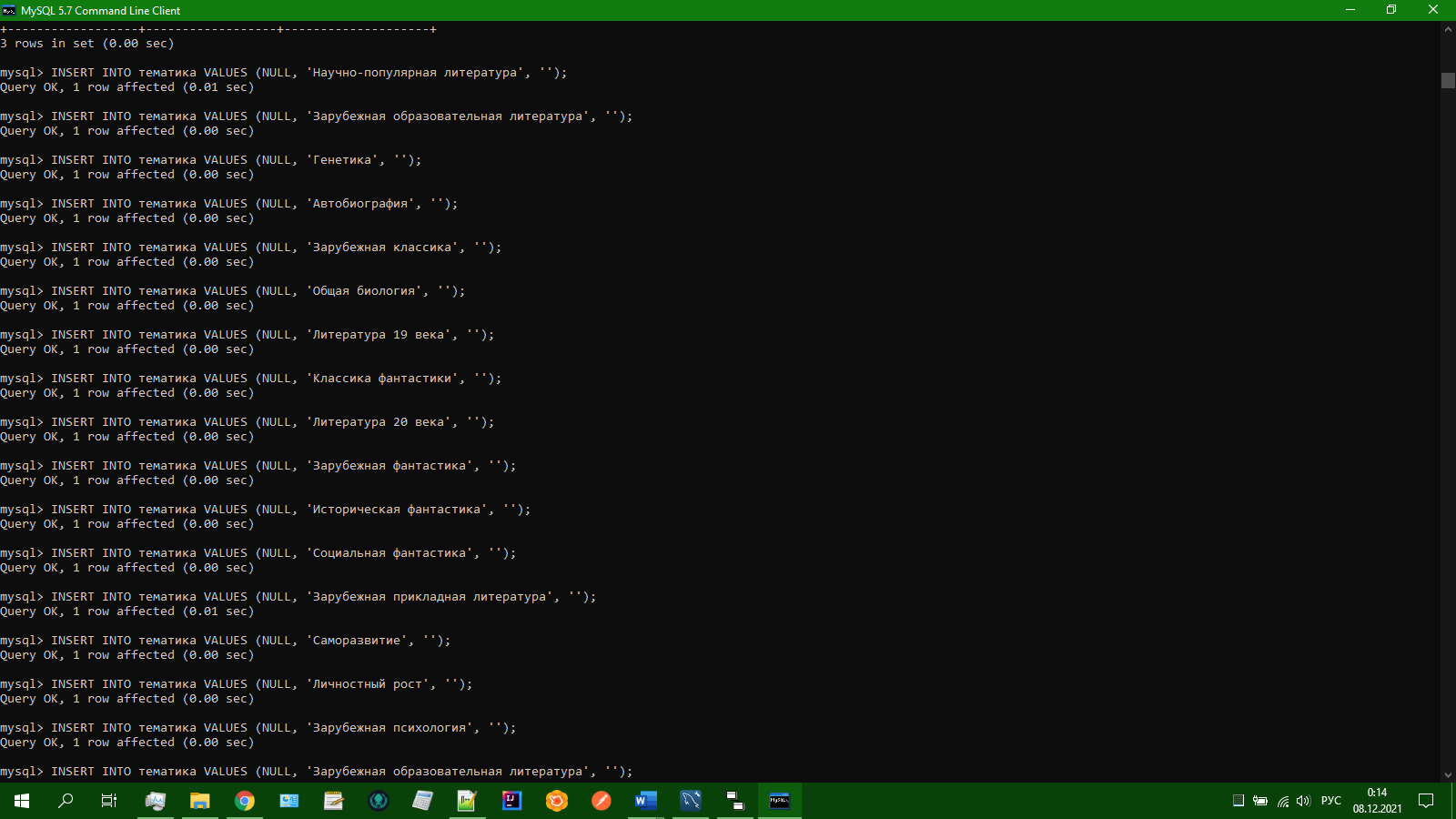


Рис. 16 Заполнение таблицы тематика

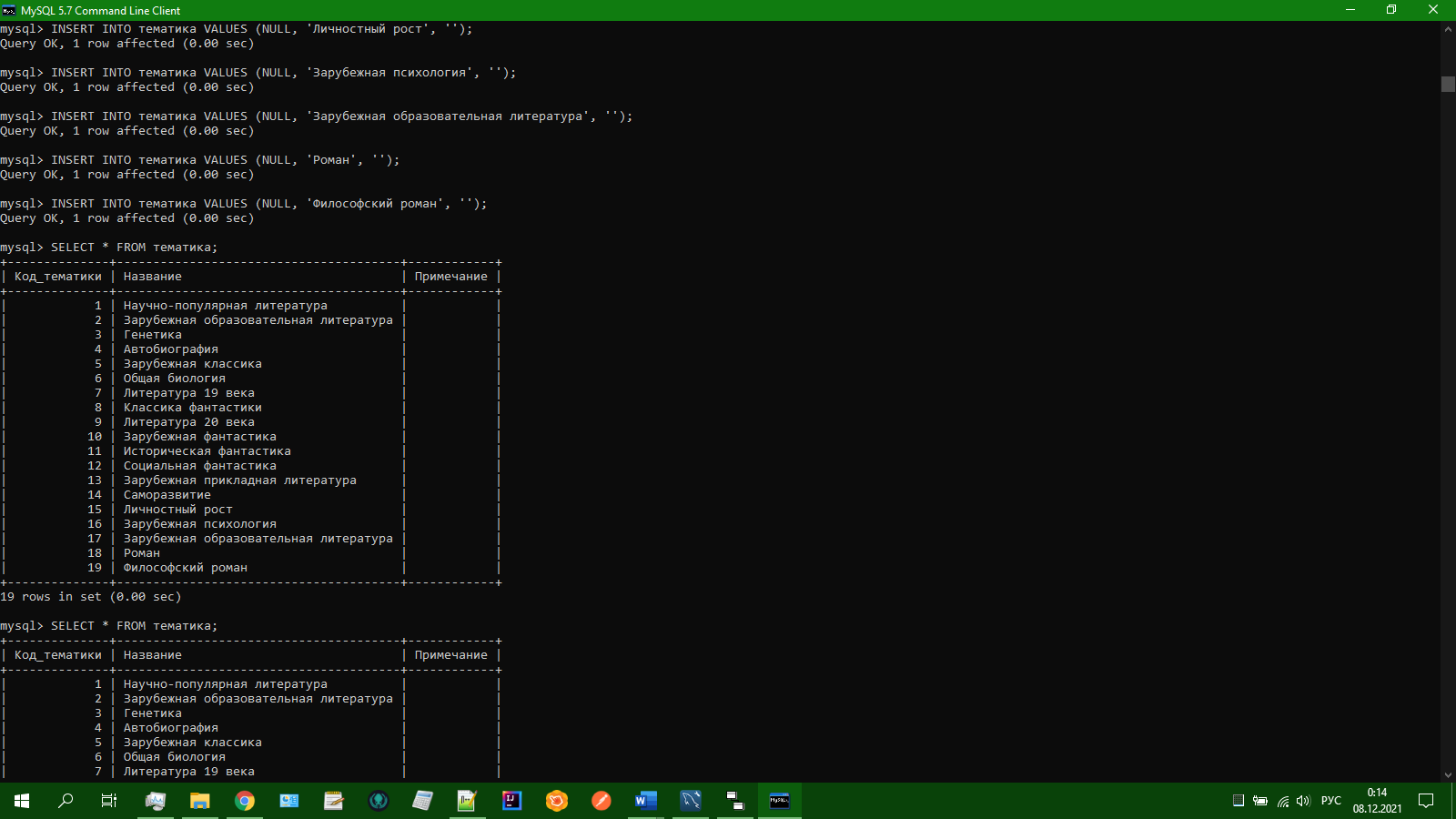


Рис. 17 Продолжение заполнение таблицы тематика и проверка

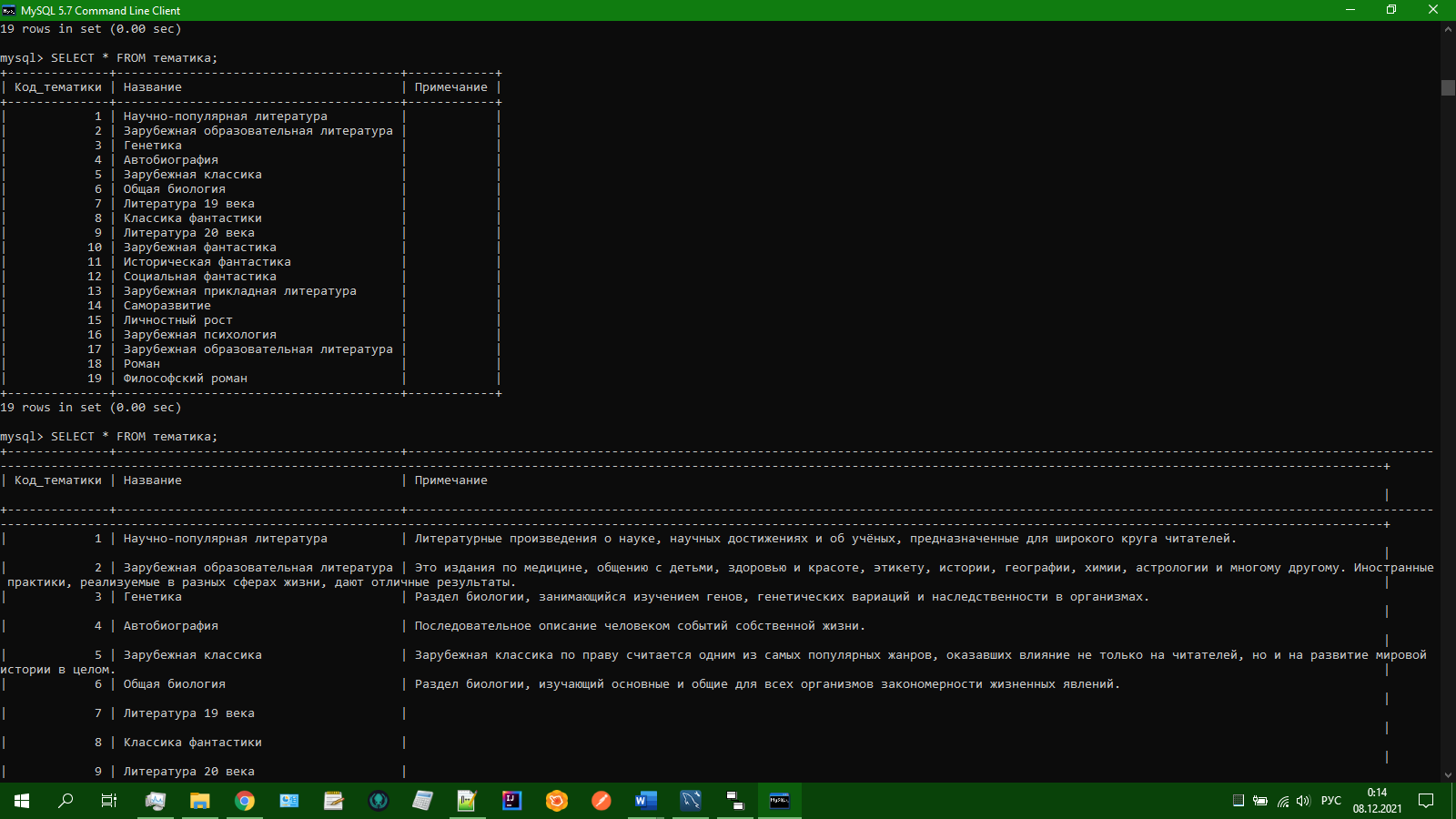


Рис. 18 Продолжение проверки таблицы тематика

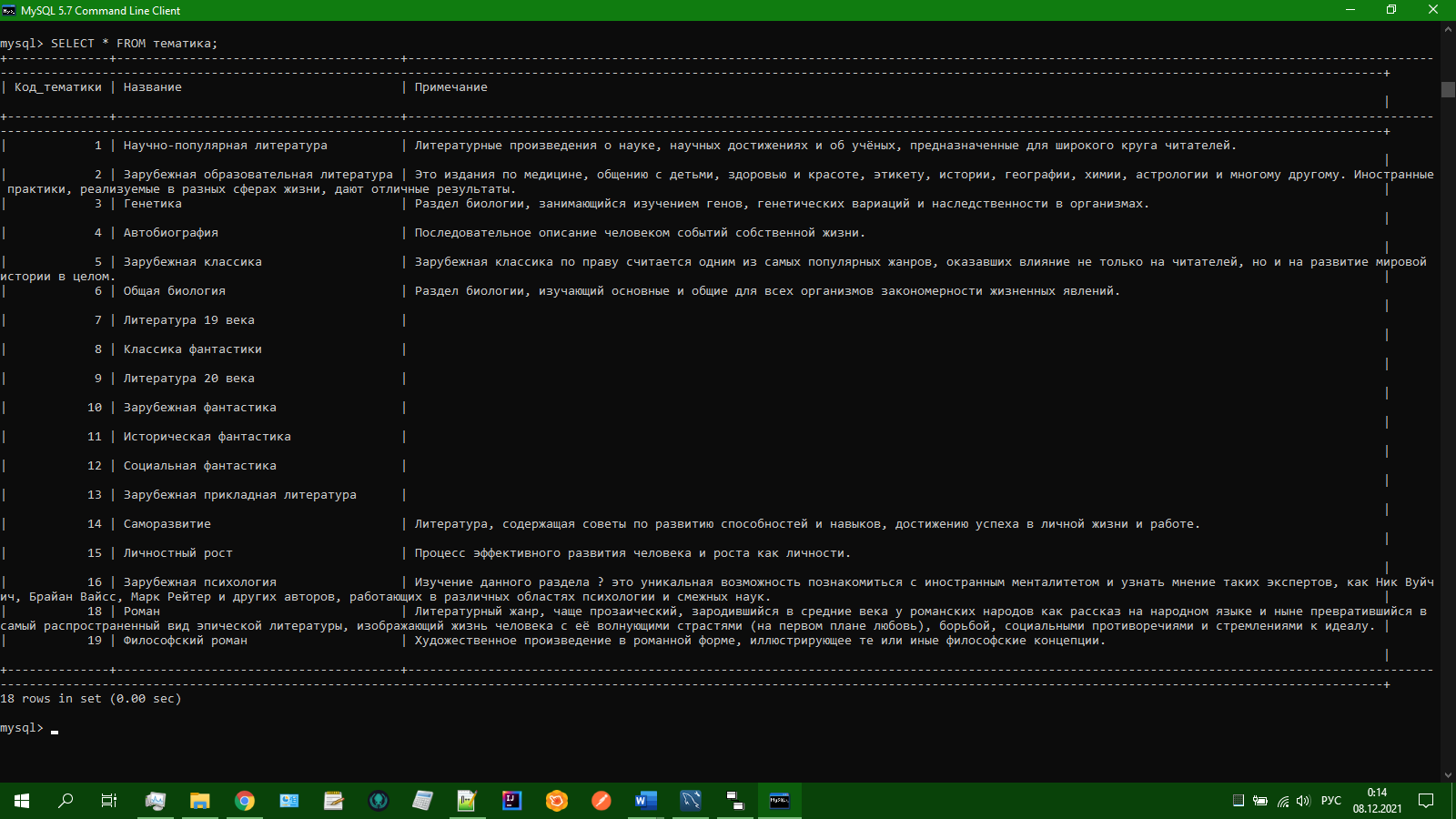


Рис. 19 Продолжение проверки таблицы тематика

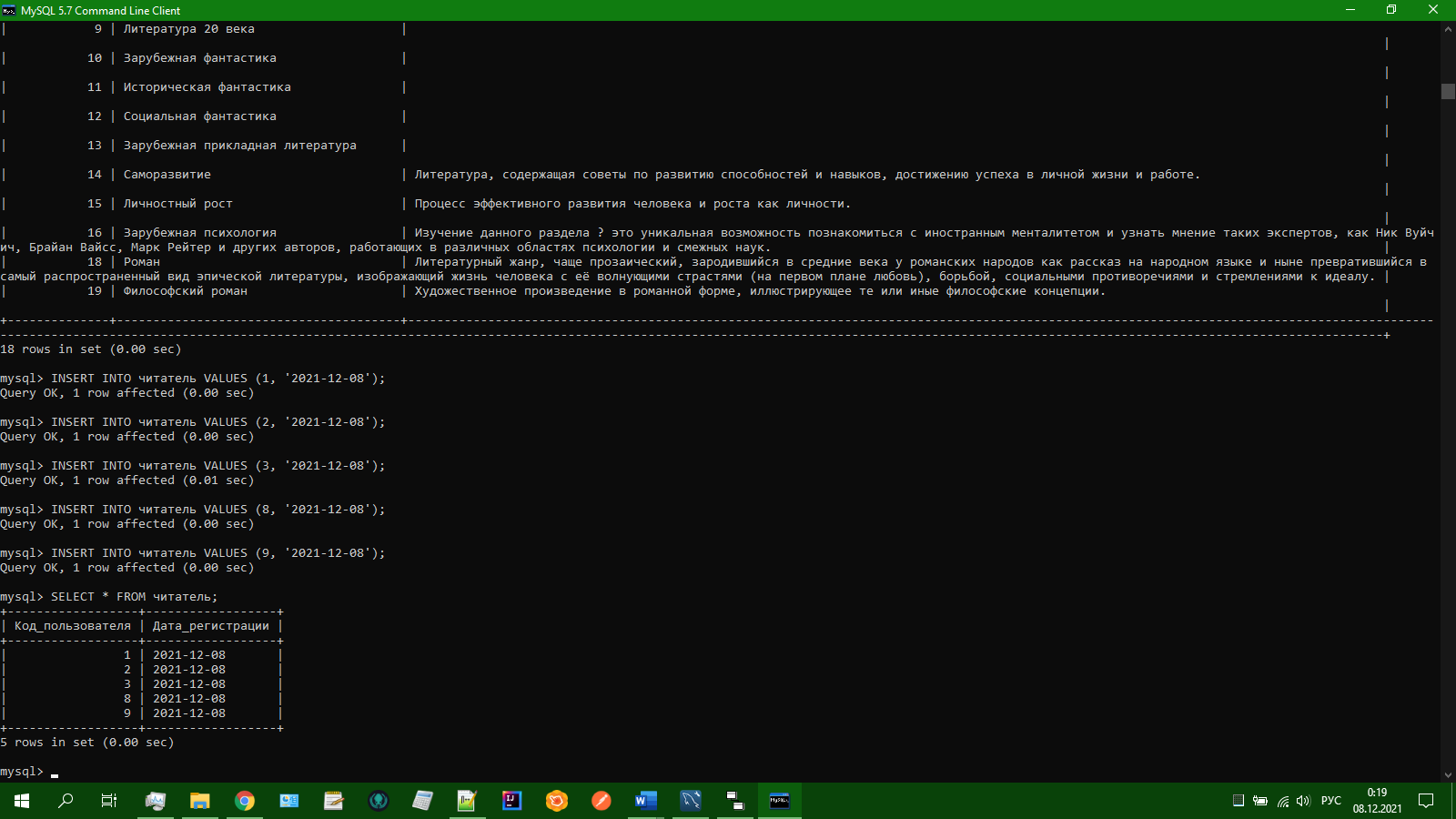


Рис. 20 Заполнение таблицы читатель и проверка

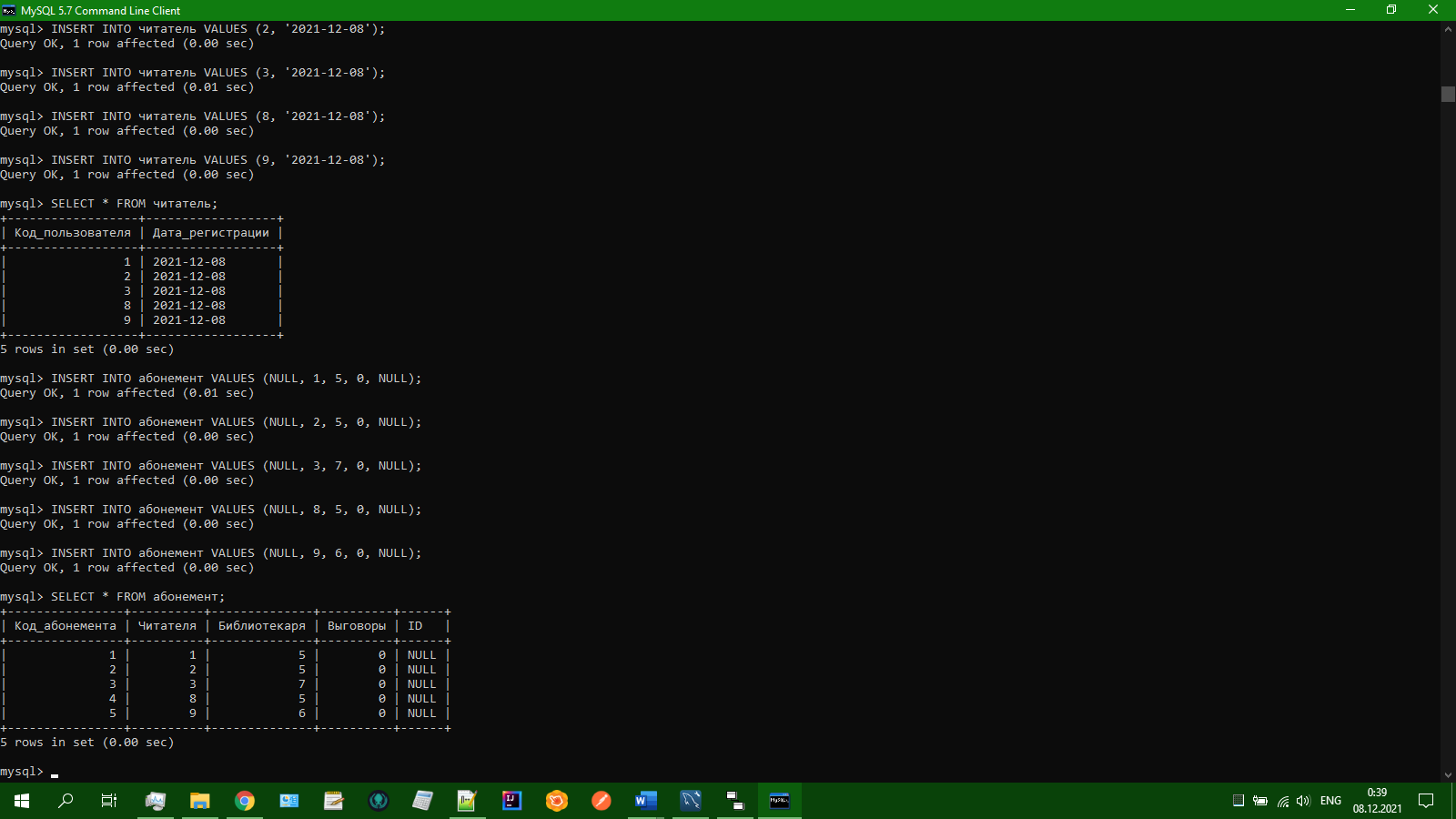


Рис. 21 Заполнение таблицы абонемент и проверка

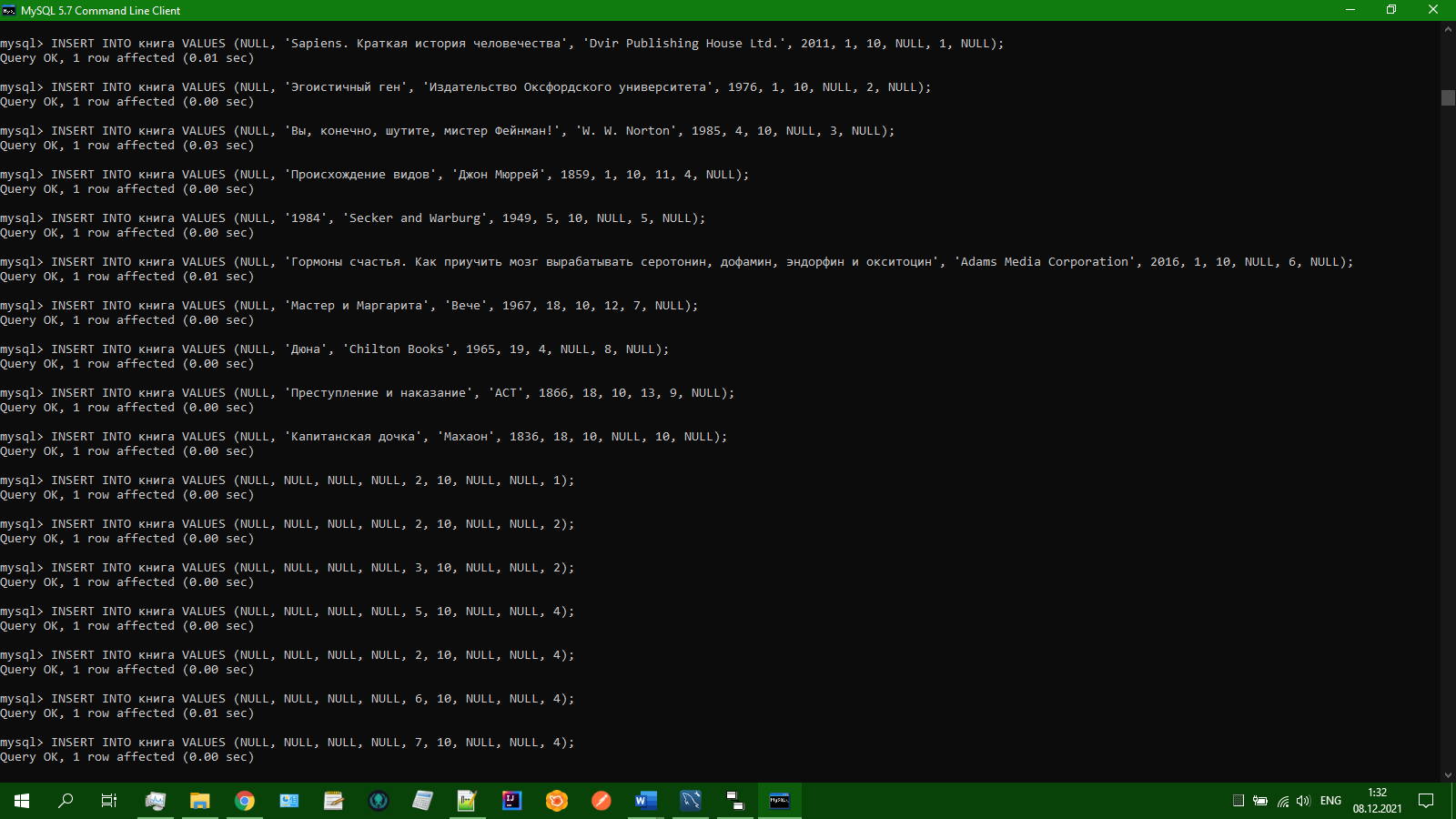


Рис. 22 Заполнение таблицы книга

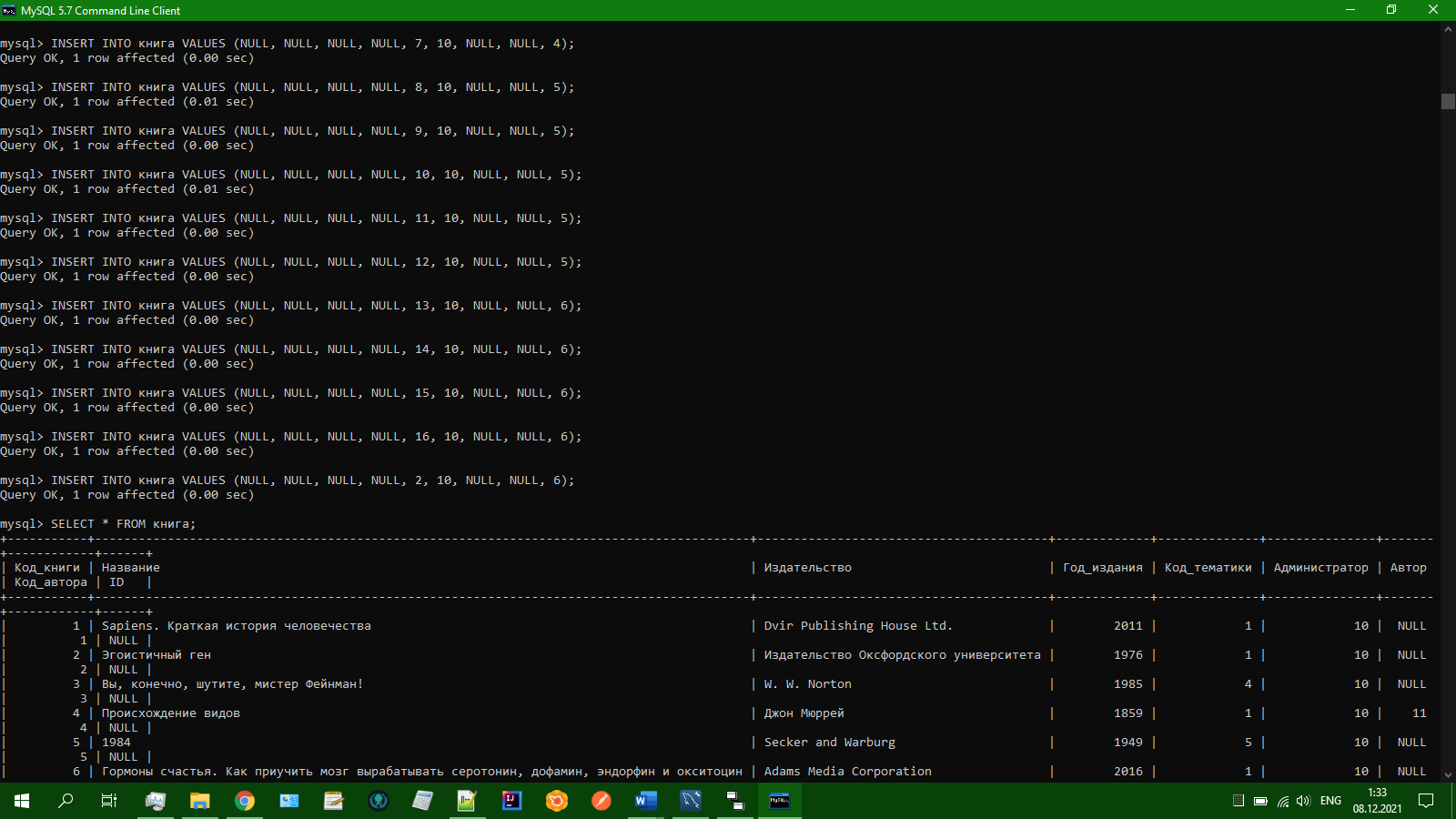


Рис. 23 Продолжение заполнения таблицы книга и проверка

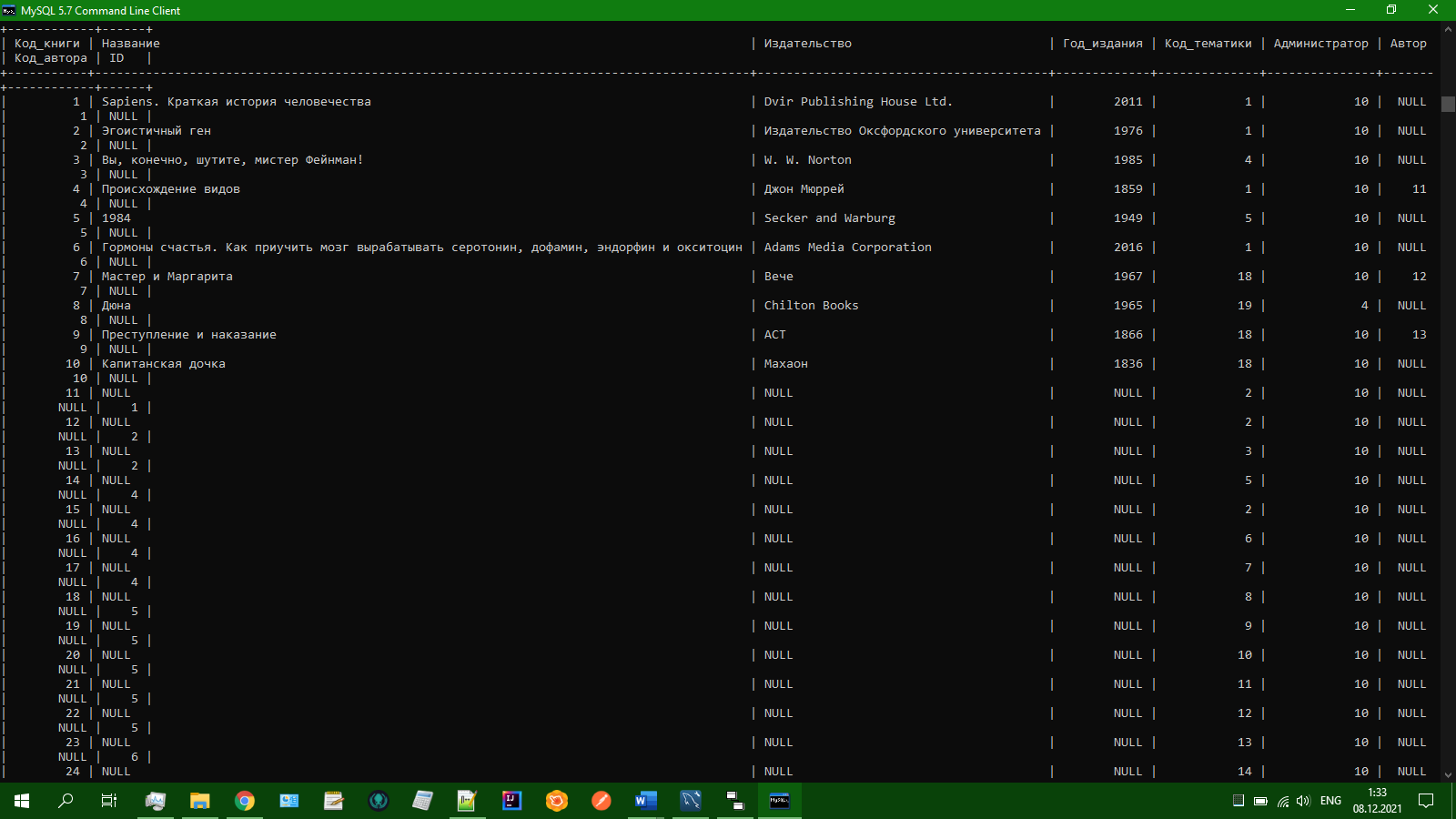


Рис. 24 Продолжение проверки таблицы книга

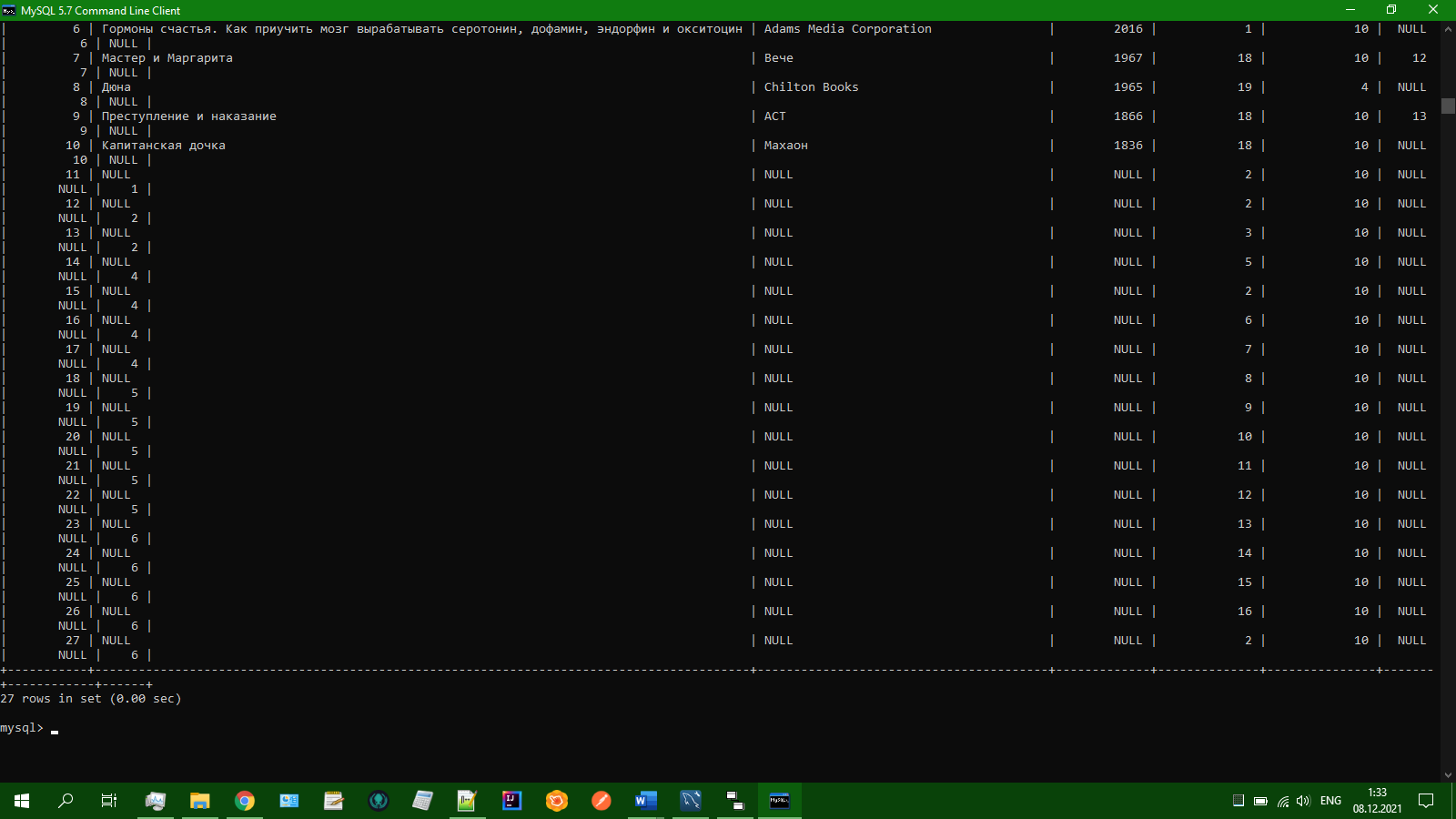


Рис. 25 Продолжение проверки таблицы книга

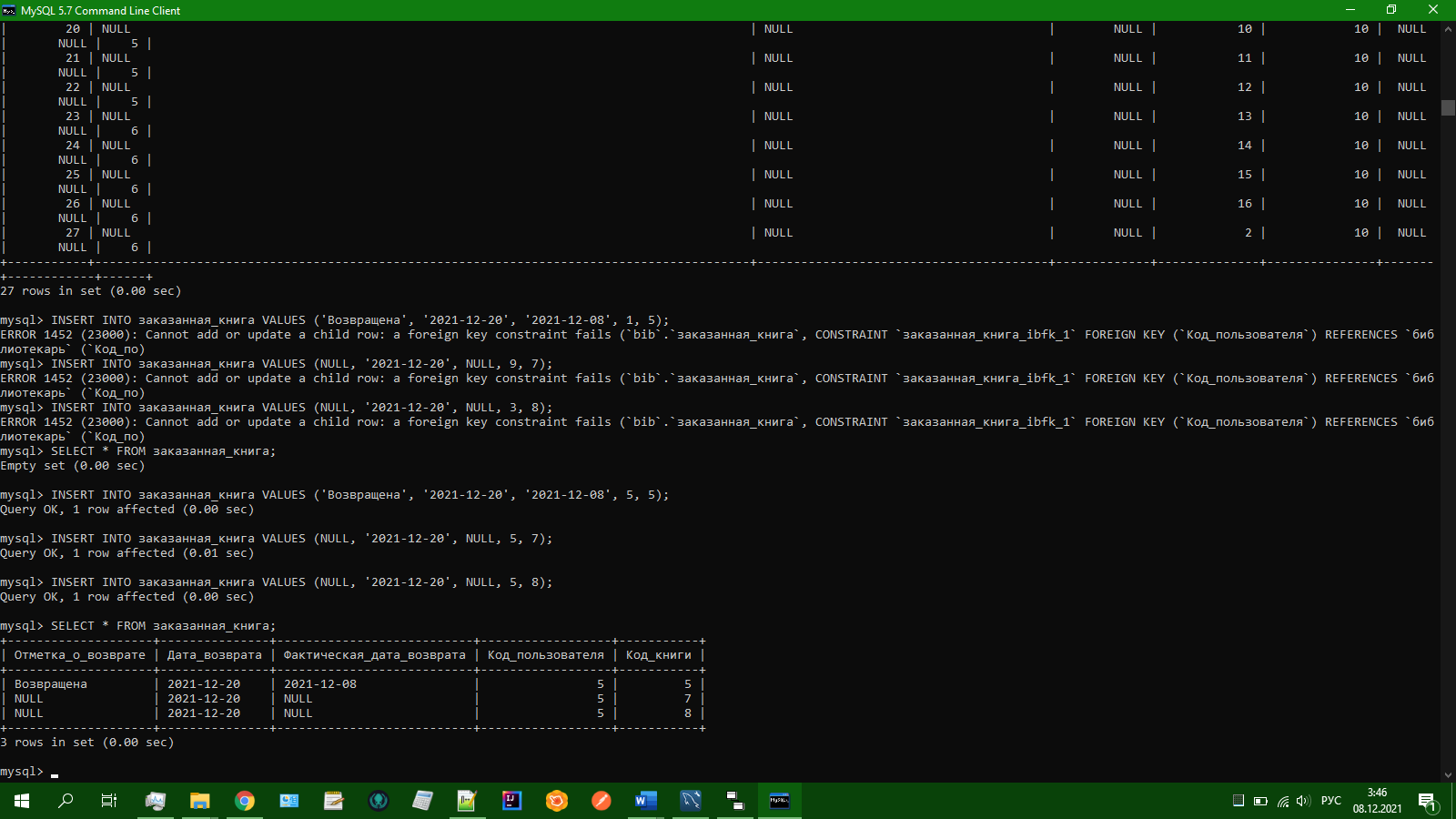


Рис. 26 Заполнение таблицы заказанная\_книга и проверка

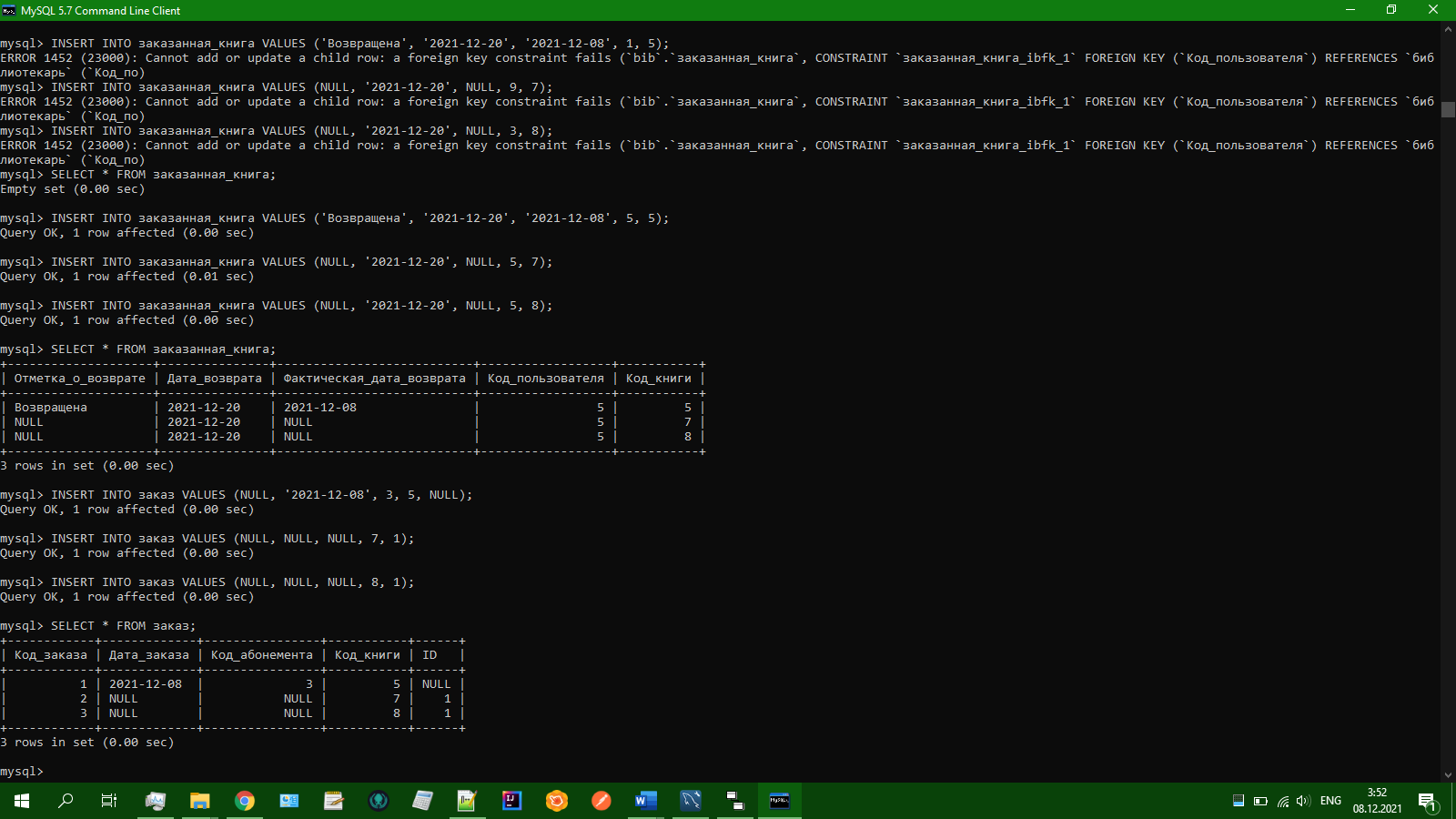
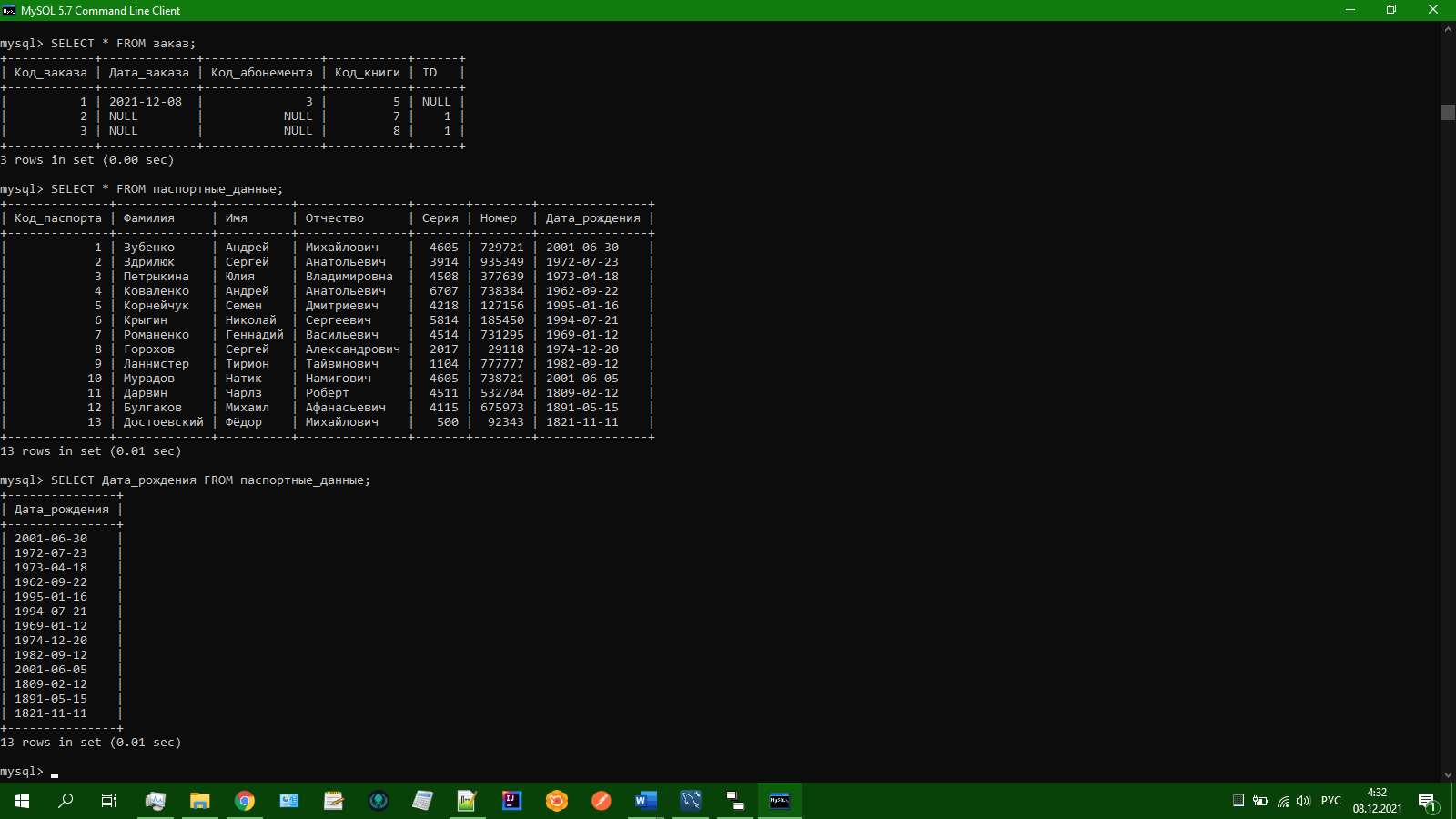


Рис. 27 Заполнение таблицы заказ и проверка

# Практическое задание 6

**Работа с запросами в базе данных bib**

  
Рис. 28 ‒ Просмотр всех столбцов из таблицы паспортные\_данные

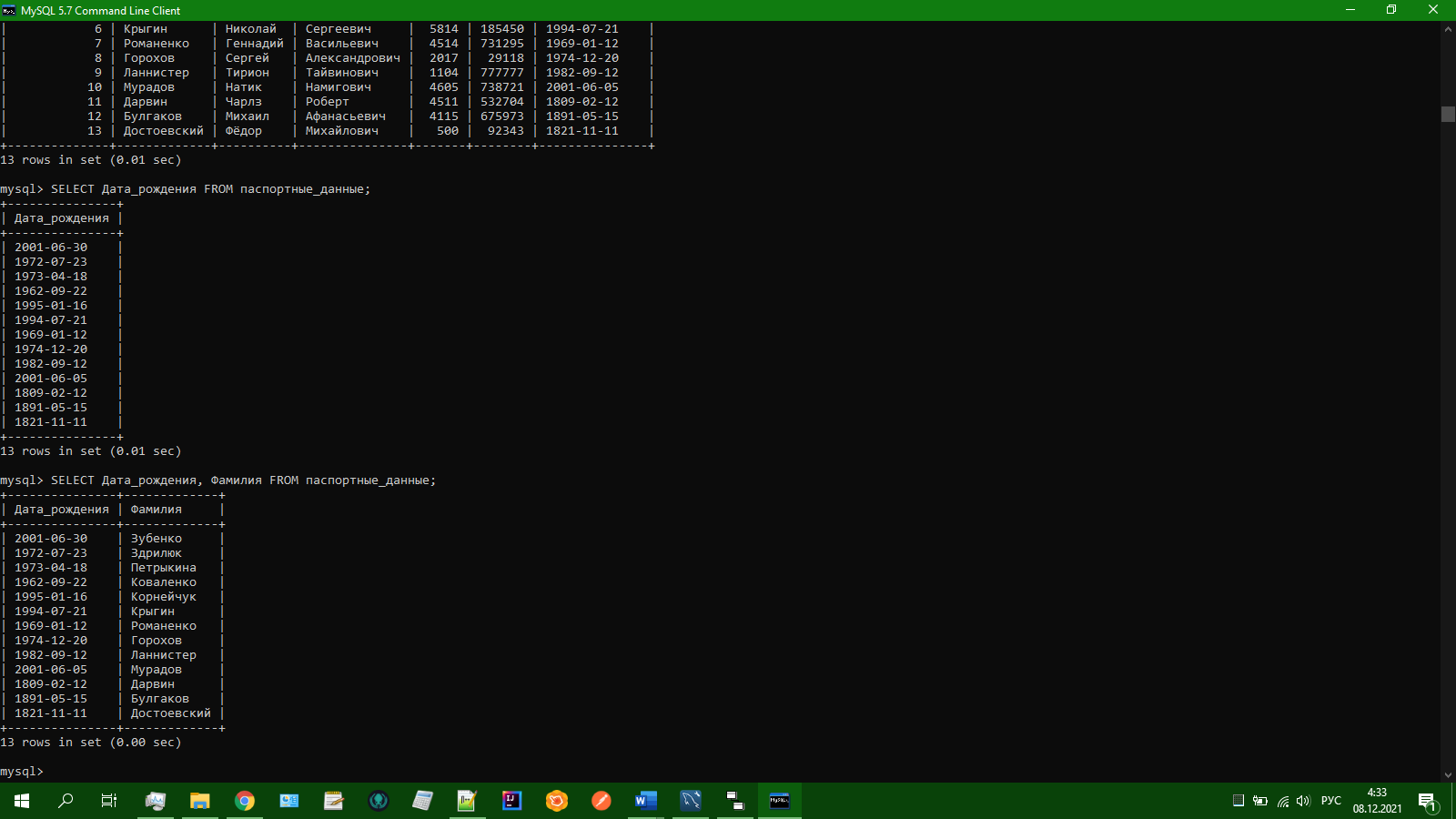


Рис. 29 ‒ Просмотр столбца Дата\_рождения из таблицы паспортные\_данные

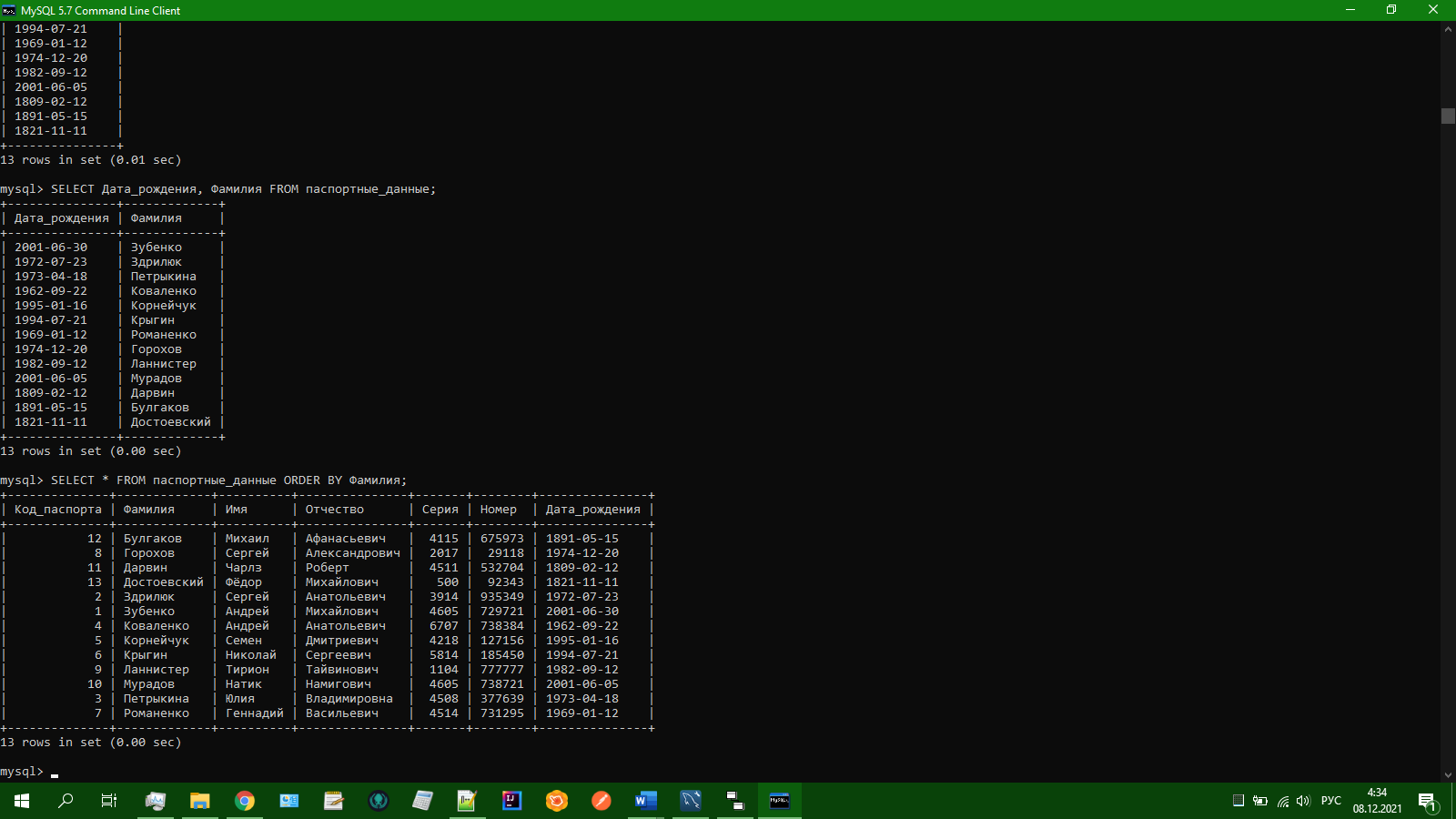
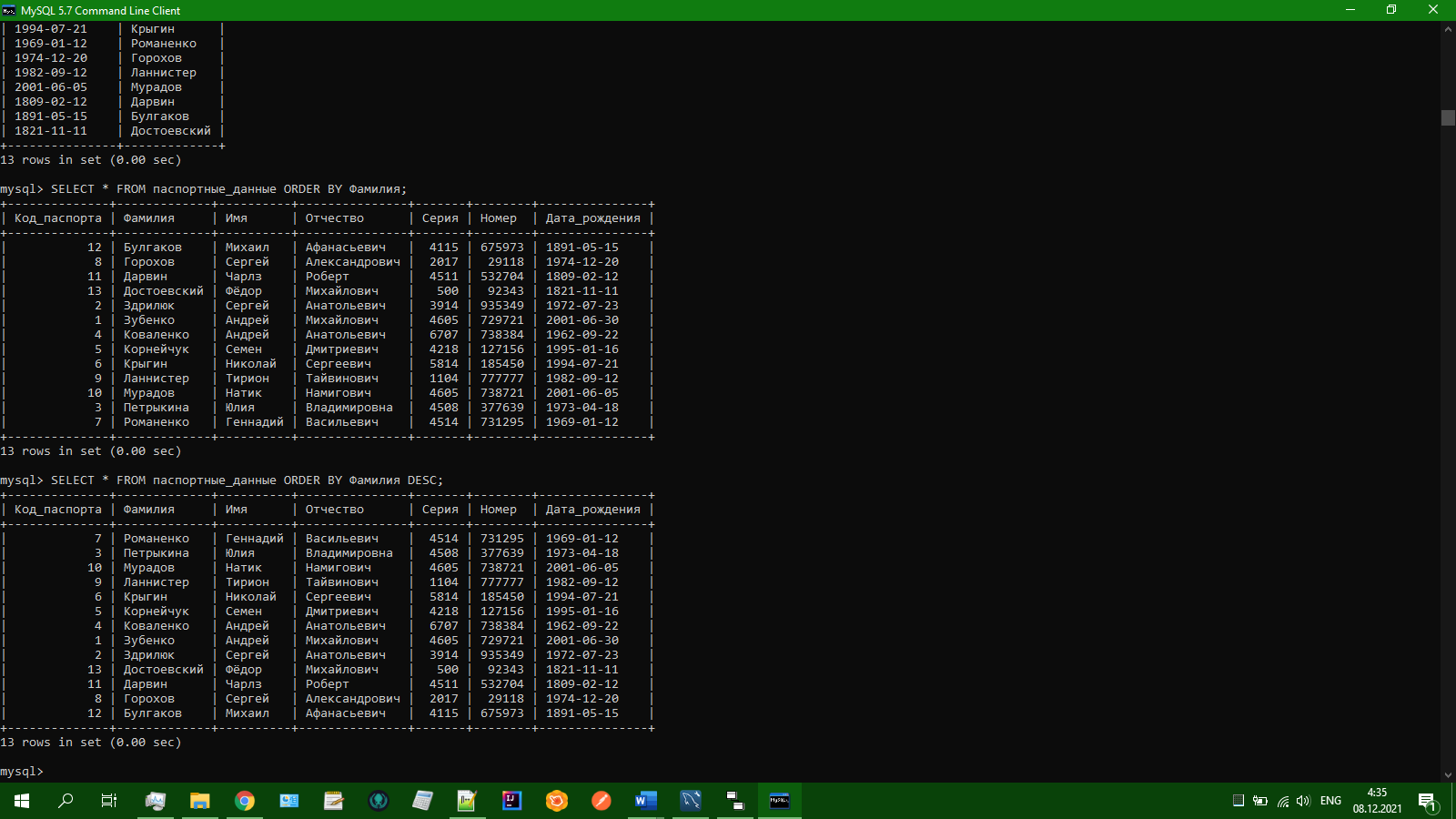
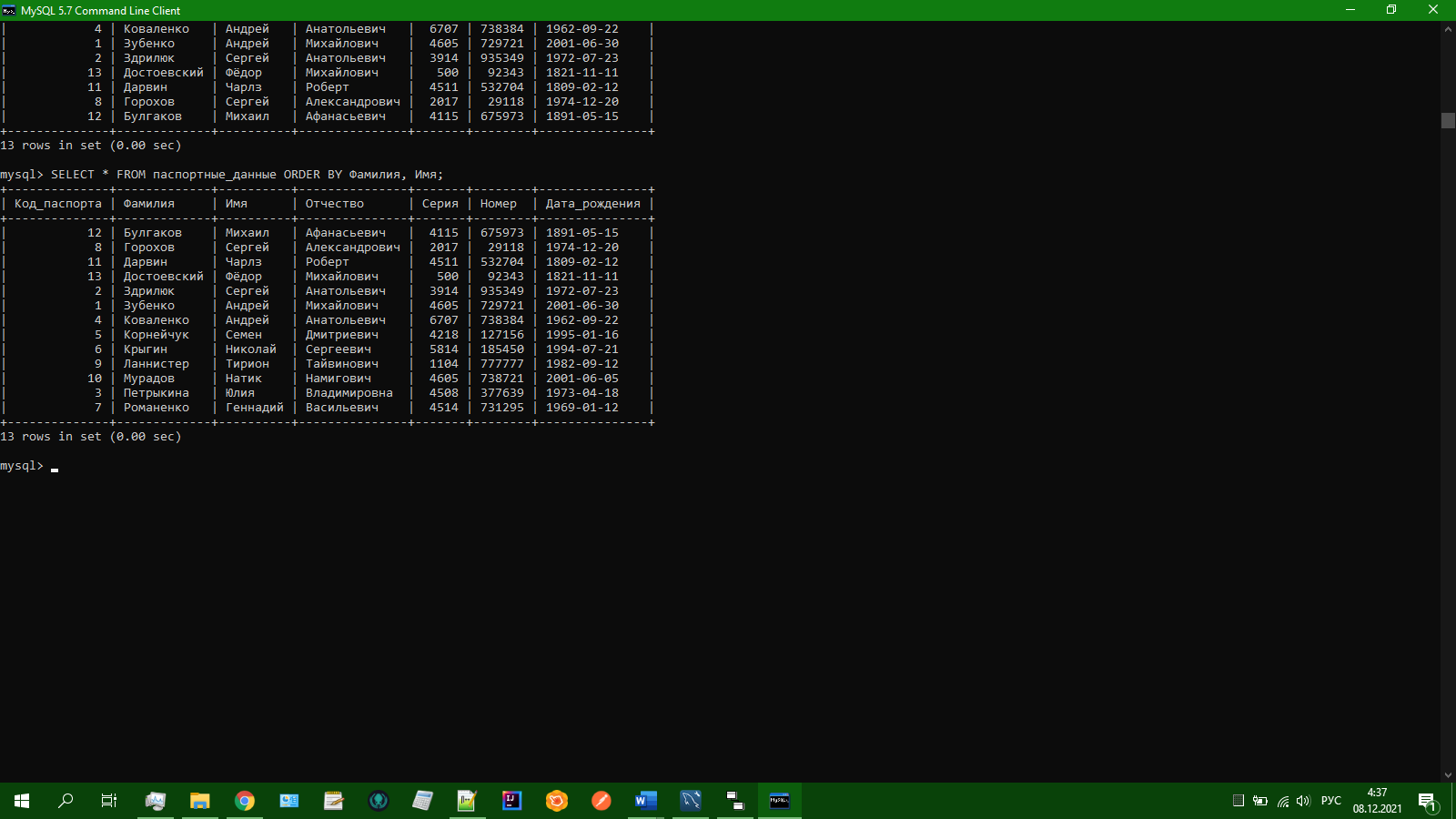
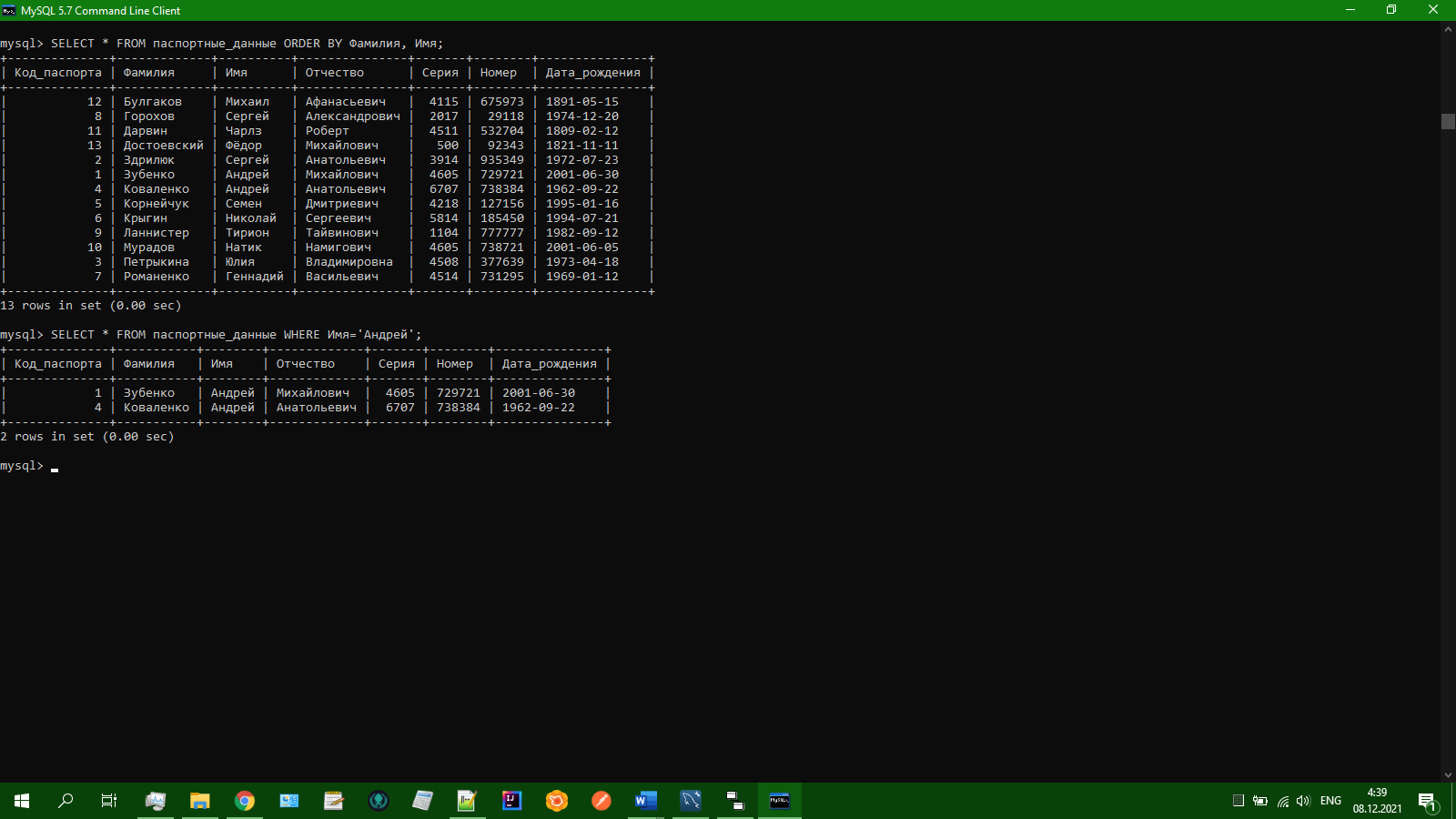
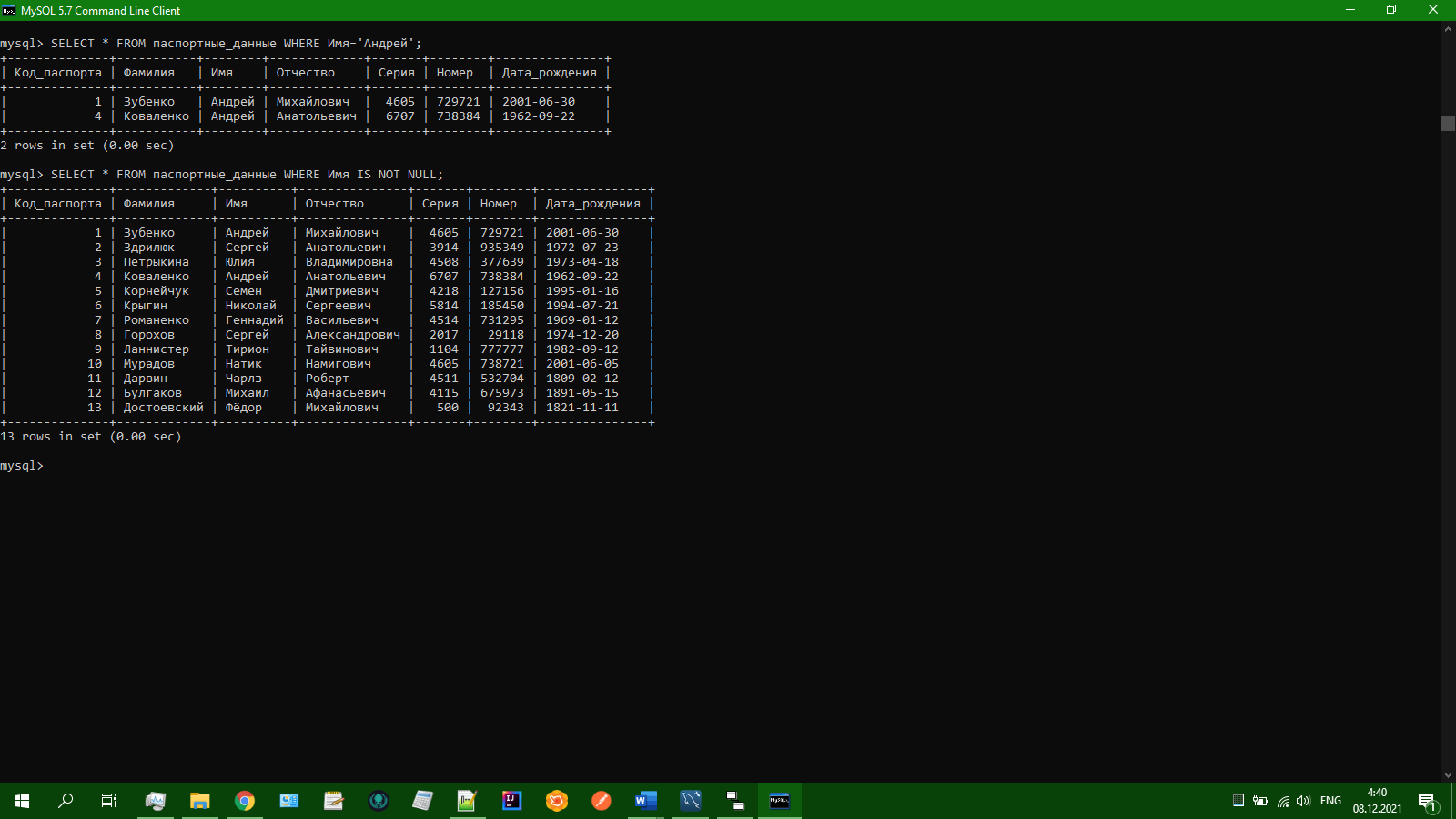


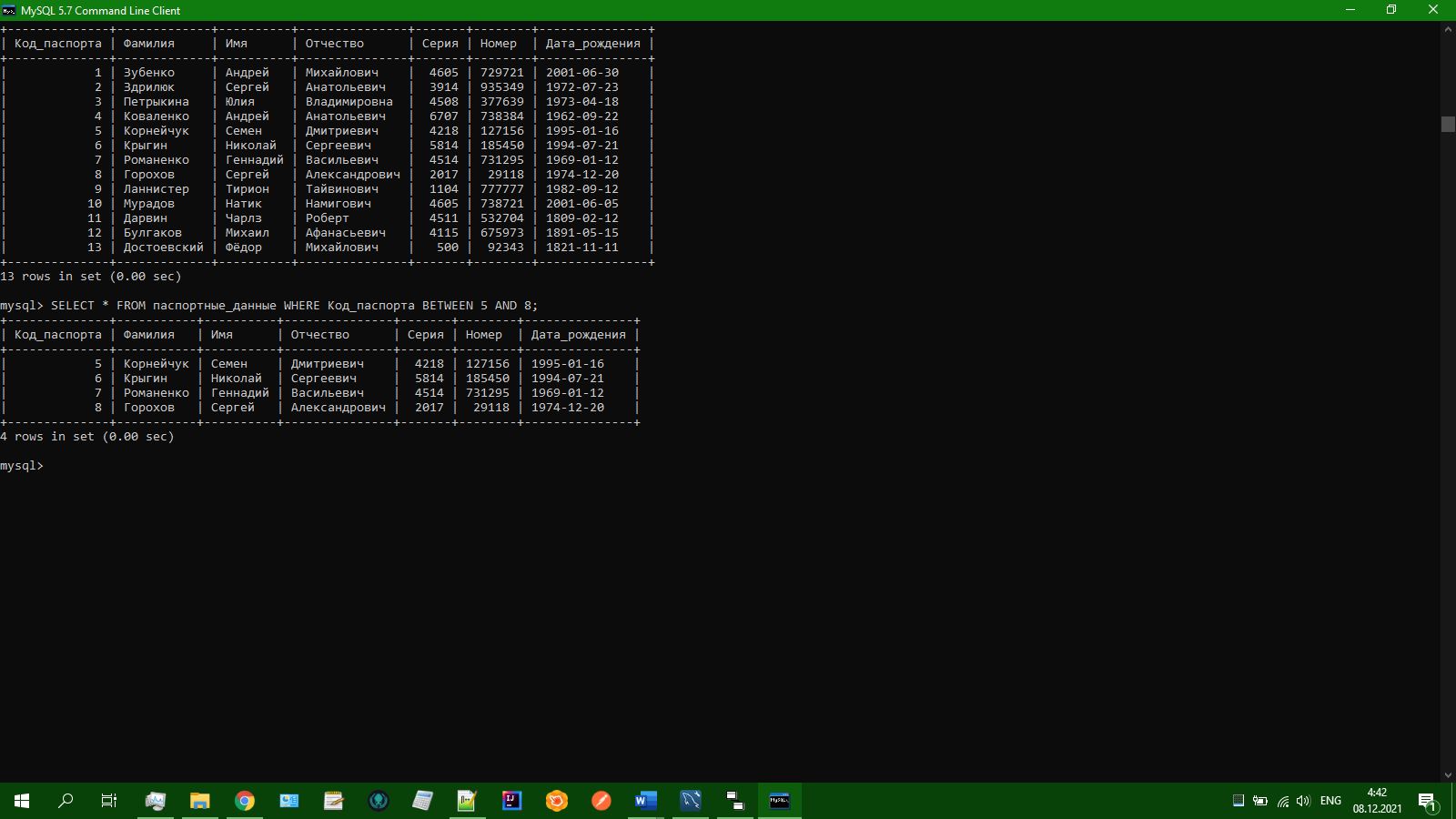
Рис. 30 ‒ Вывод таблицы паспортные\_данные, отсортированной по столбцу фамилия

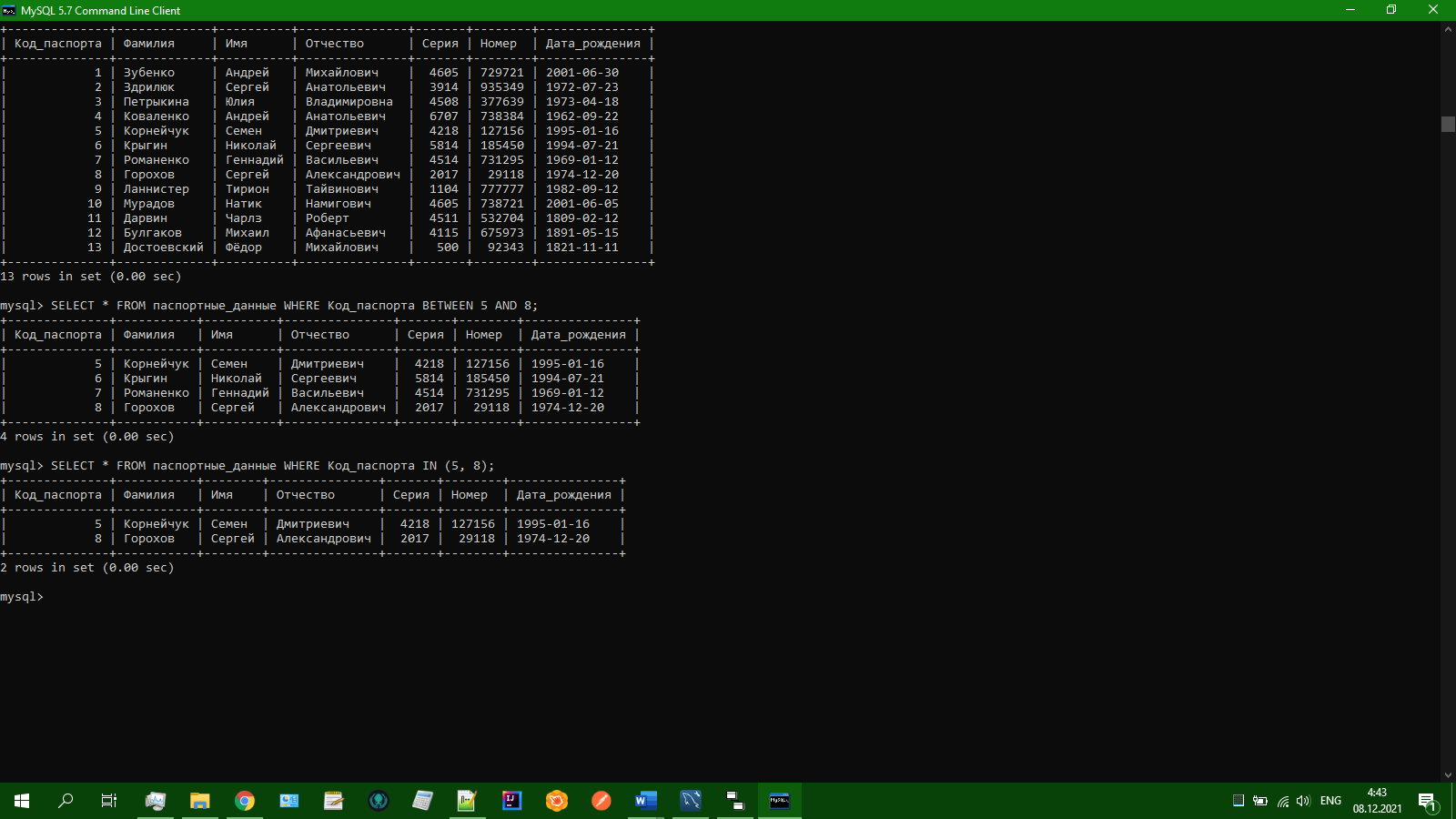
Рис. 31 Вывод таблицы паспортные\_данные, отсортированной по столбцу фамилия по убыванию

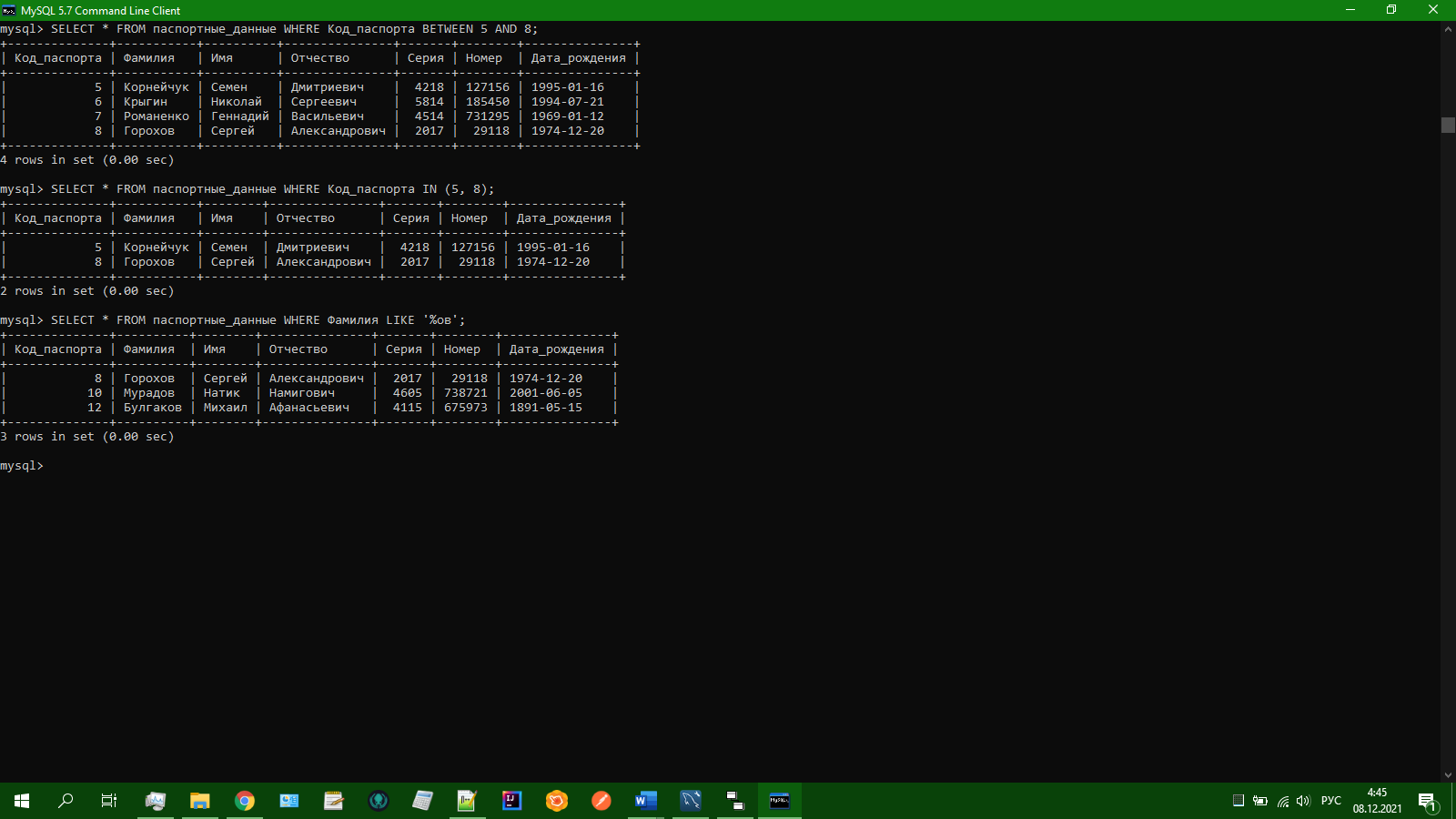
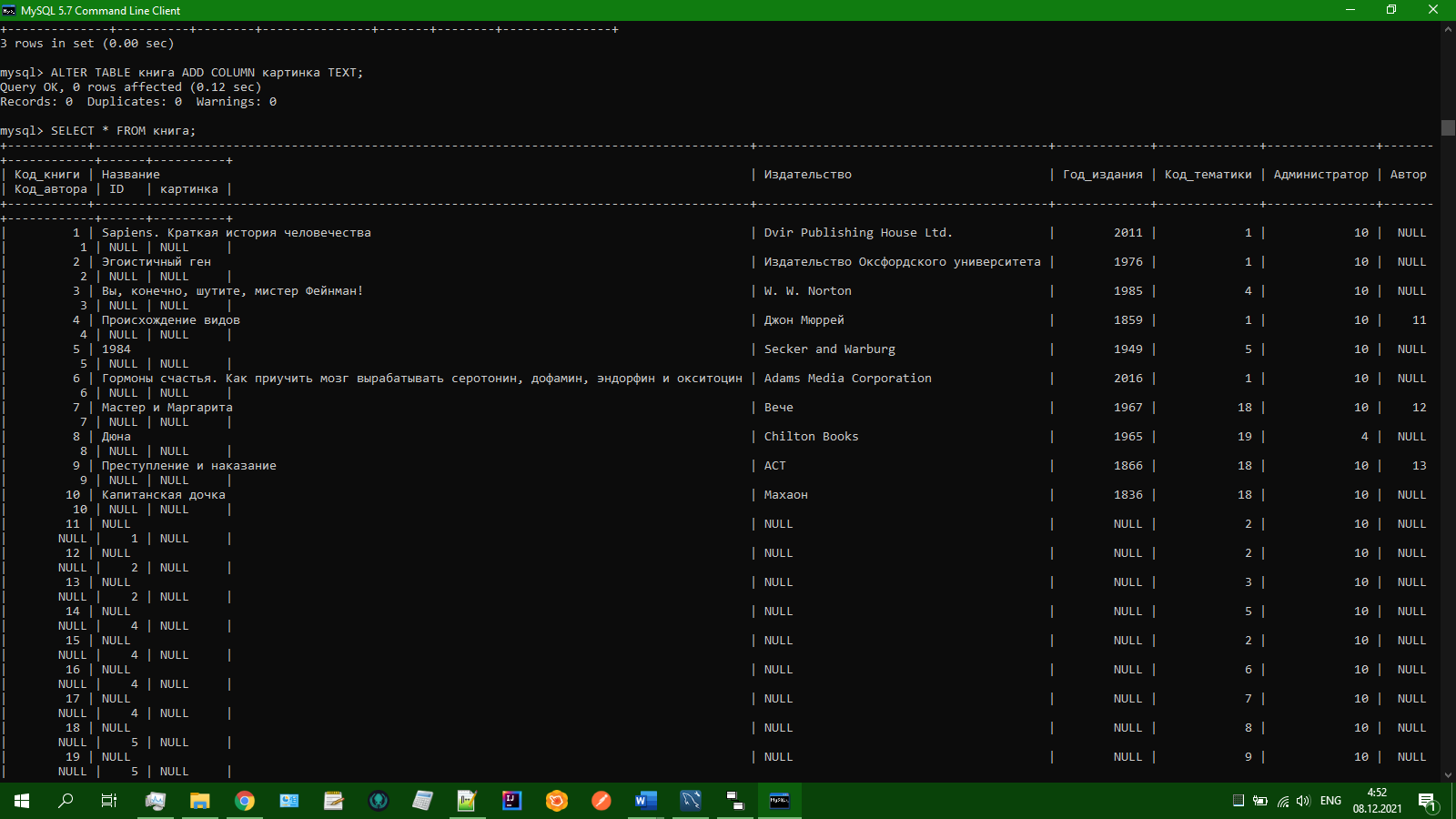
Рис. 32 Вывод таблицы паспортные\_данные, отсортированной по нескольким столбцам

Рис. 33 Вывод строки по условию(=) из таблицы паспортные\_данные

Рис. 34 Вывод строки по условию(NOT NULL) из таблицы паспортные\_данные

Рис. 35 Вывод строки по условию(BETWEEN 5 AND 8) из таблицы паспортные\_данные

Рис. 36 Вывод строки по условию(IN (3, 6)) из таблицы паспортные\_данные

Рис. 37 Вывод строки по условию(LIKE) из таблицы паспортные\_данные Рис. 38 Добавление столбца картинка в таблицу книга

# Практическое задание 7

**Построение модели с помощью оболочки MySQL Workbench**

Мною была построена модель в СУБД Workbench(Рисунок 39)

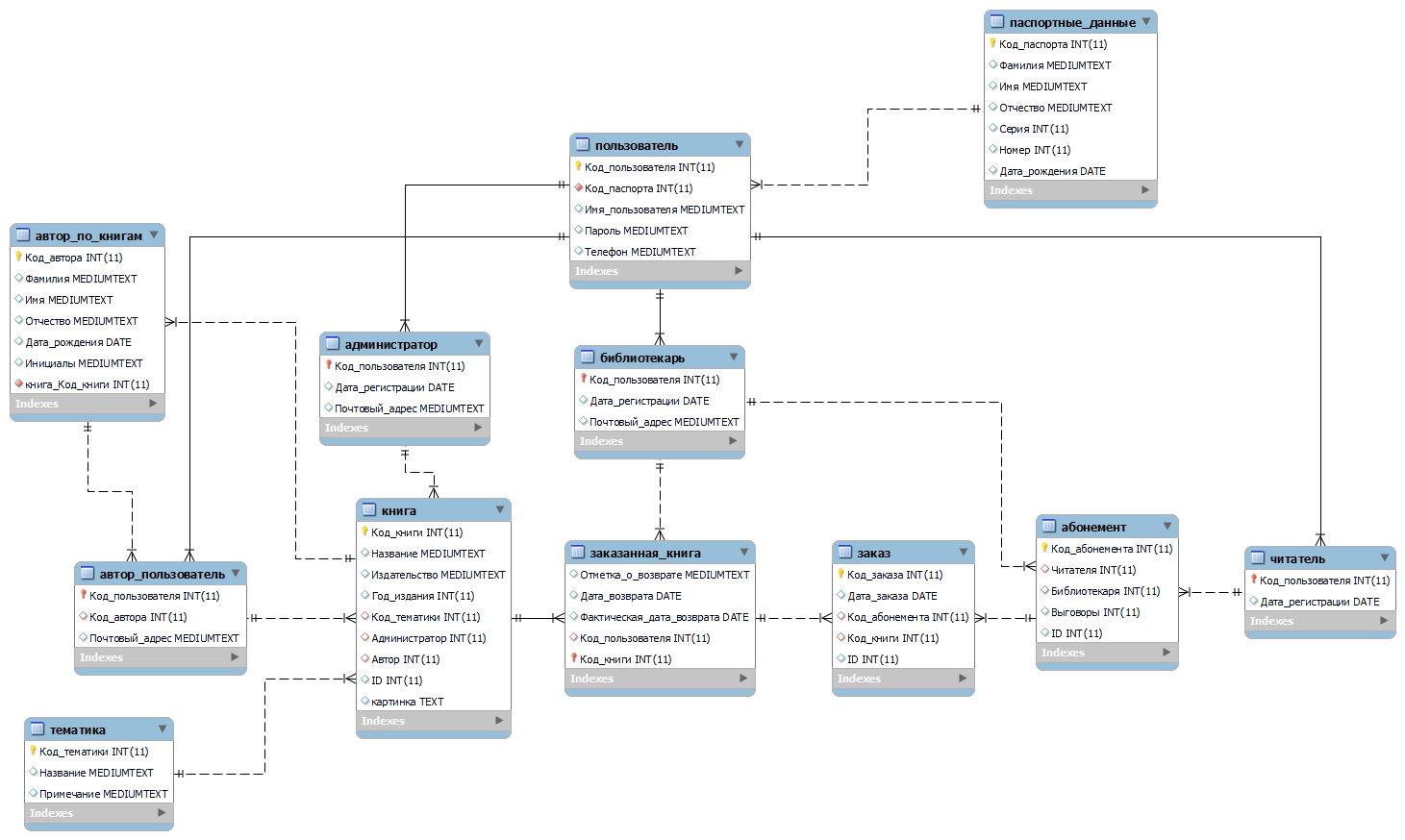


Рисунок 39 – Построенная модель

Перенос Базы Данных на другой сервер

Использовав встроенные инструменты MYSQL Workbench, был получен MySql скрипт

-- MySQL Script generated by MySQL Workbench

-- Wed Dec 8 06:46:43 2021

-- 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\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema bib

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema bib

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `bib` DEFAULT CHARACTER SET latin1 ;

USE `bib` ;

-- -----------------------------------------------------

-- Table `bib`.`паспортные\_данные`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`паспортные\_данные` (

`Код\_паспорта` INT(11) NOT NULL AUTO\_INCREMENT,

`Фамилия` MEDIUMTEXT NULL DEFAULT NULL,

`Имя` MEDIUMTEXT NULL DEFAULT NULL,

`Отчество` MEDIUMTEXT NULL DEFAULT NULL,

`Серия` INT(11) NULL DEFAULT NULL,

`Номер` INT(11) NULL DEFAULT NULL,

`Дата\_рождения` DATE NULL DEFAULT NULL,

PRIMARY KEY (`Код\_паспорта`))

ENGINE = InnoDB

AUTO\_INCREMENT = 14

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`пользователь`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`пользователь` (

`Код\_пользователя` INT(11) NOT NULL AUTO\_INCREMENT,

`Код\_паспорта` INT(11) NOT NULL,

`Имя\_пользователя` MEDIUMTEXT NULL DEFAULT NULL,

`Пароль` MEDIUMTEXT NULL DEFAULT NULL,

`Телефон` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`Код\_пользователя`),

INDEX `R\_22` (`Код\_паспорта` ASC) VISIBLE,

CONSTRAINT `пользователь\_ibfk\_1`

FOREIGN KEY (`Код\_паспорта`)

REFERENCES `bib`.`паспортные\_данные` (`Код\_паспорта`))

ENGINE = InnoDB

AUTO\_INCREMENT = 14

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`читатель`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`читатель` (

`Код\_пользователя` INT(11) NOT NULL,

`Дата\_регистрации` DATE NULL DEFAULT NULL,

PRIMARY KEY (`Код\_пользователя`),

CONSTRAINT `читатель\_ibfk\_1`

FOREIGN KEY (`Код\_пользователя`)

REFERENCES `bib`.`пользователь` (`Код\_пользователя`)

ON DELETE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`библиотекарь`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`библиотекарь` (

`Код\_пользователя` INT(11) NOT NULL,

`Дата\_регистрации` DATE NULL DEFAULT NULL,

`Почтовый\_адрес` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`Код\_пользователя`),

CONSTRAINT `библиотекарь\_ibfk\_1`

FOREIGN KEY (`Код\_пользователя`)

REFERENCES `bib`.`пользователь` (`Код\_пользователя`)

ON DELETE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`абонемент`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`абонемент` (

`Код\_абонемента` INT(11) NOT NULL AUTO\_INCREMENT,

`Читателя` INT(11) NULL DEFAULT NULL,

`Библиотекаря` INT(11) NULL DEFAULT NULL,

`Выговоры` INT(11) NULL DEFAULT NULL,

`ID` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`Код\_абонемента`),

INDEX `R\_18` (`Читателя` ASC) VISIBLE,

INDEX `R\_34` (`Библиотекаря` ASC) VISIBLE,

CONSTRAINT `абонемент\_ibfk\_1`

FOREIGN KEY (`Читателя`)

REFERENCES `bib`.`читатель` (`Код\_пользователя`),

CONSTRAINT `абонемент\_ibfk\_2`

FOREIGN KEY (`Библиотекаря`)

REFERENCES `bib`.`библиотекарь` (`Код\_пользователя`))

ENGINE = InnoDB

AUTO\_INCREMENT = 6

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`тематика`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`тематика` (

`Код\_тематики` INT(11) NOT NULL AUTO\_INCREMENT,

`Название` MEDIUMTEXT NULL DEFAULT NULL,

`Примечание` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`Код\_тематики`))

ENGINE = InnoDB

AUTO\_INCREMENT = 20

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`автор\_пользователь`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`автор\_пользователь` (

`Код\_пользователя` INT(11) NOT NULL,

`Код\_автора` INT(11) NULL DEFAULT NULL,

`Почтовый\_адрес` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`Код\_пользователя`),

INDEX `R\_44` (`Код\_автора` ASC) VISIBLE,

CONSTRAINT `автор\_пользователь\_ibfk\_1`

FOREIGN KEY (`Код\_пользователя`)

REFERENCES `bib`.`пользователь` (`Код\_пользователя`)

ON DELETE CASCADE,

CONSTRAINT `автор\_пользователь\_ibfk\_2`

FOREIGN KEY (`Код\_автора`)

REFERENCES `bib`.`автор\_по\_книгам` (`Код\_автора`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`администратор`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`администратор` (

`Код\_пользователя` INT(11) NOT NULL,

`Дата\_регистрации` DATE NULL DEFAULT NULL,

`Почтовый\_адрес` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`Код\_пользователя`),

CONSTRAINT `администратор\_ibfk\_1`

FOREIGN KEY (`Код\_пользователя`)

REFERENCES `bib`.`пользователь` (`Код\_пользователя`)

ON DELETE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`книга`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`книга` (

`Код\_книги` INT(11) NOT NULL AUTO\_INCREMENT,

`Название` MEDIUMTEXT NULL DEFAULT NULL,

`Издательство` MEDIUMTEXT NULL DEFAULT NULL,

`Год\_издания` INT(11) NULL DEFAULT NULL,

`Код\_тематики` INT(11) NULL DEFAULT NULL,

`Администратор` INT(11) NULL DEFAULT NULL,

`Автор` INT(11) NULL DEFAULT NULL,

`ID` INT(11) NULL DEFAULT NULL,

`картинка` TEXT NULL DEFAULT NULL,

PRIMARY KEY (`Код\_книги`),

INDEX `R\_17` (`Код\_тематики` ASC) VISIBLE,

INDEX `R\_19` (`Автор` ASC) VISIBLE,

INDEX `R\_30` (`Администратор` ASC) VISIBLE,

CONSTRAINT `книга\_ibfk\_1`

FOREIGN KEY (`Код\_тематики`)

REFERENCES `bib`.`тематика` (`Код\_тематики`),

CONSTRAINT `книга\_ibfk\_2`

FOREIGN KEY (`Автор`)

REFERENCES `bib`.`автор\_пользователь` (`Код\_пользователя`),

CONSTRAINT `книга\_ibfk\_3`

FOREIGN KEY (`Администратор`)

REFERENCES `bib`.`администратор` (`Код\_пользователя`))

ENGINE = InnoDB

AUTO\_INCREMENT = 28

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`автор\_по\_книгам`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`автор\_по\_книгам` (

`Код\_автора` INT(11) NOT NULL AUTO\_INCREMENT,

`Фамилия` MEDIUMTEXT NULL DEFAULT NULL,

`Имя` MEDIUMTEXT NULL DEFAULT NULL,

`Отчество` MEDIUMTEXT NULL DEFAULT NULL,

`Дата\_рождения` DATE NULL DEFAULT NULL,

`Инициалы` MEDIUMTEXT NULL DEFAULT NULL,

`книга\_Код\_книги` INT(11) NOT NULL,

PRIMARY KEY (`Код\_автора`),

INDEX `fk\_автор\_по\_книгам\_книга1\_idx` (`книга\_Код\_книги` ASC) VISIBLE,

CONSTRAINT `fk\_автор\_по\_книгам\_книга1`

FOREIGN KEY (`книга\_Код\_книги`)

REFERENCES `bib`.`книга` (`Код\_книги`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`заказанная\_книга`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`заказанная\_книга` (

`Отметка\_о\_возврате` MEDIUMTEXT NULL DEFAULT NULL,

`Дата\_возврата` DATE NULL DEFAULT NULL,

`Фактическая\_дата\_возврата` DATE NULL DEFAULT NULL,

`Код\_пользователя` INT(11) NULL DEFAULT NULL,

`Код\_книги` INT(11) NOT NULL,

PRIMARY KEY (`Код\_книги`),

INDEX `R\_32` (`Код\_пользователя` ASC) VISIBLE,

CONSTRAINT `заказанная\_книга\_ibfk\_1`

FOREIGN KEY (`Код\_пользователя`)

REFERENCES `bib`.`библиотекарь` (`Код\_пользователя`),

CONSTRAINT `заказанная\_книга\_ibfk\_2`

FOREIGN KEY (`Код\_книги`)

REFERENCES `bib`.`книга` (`Код\_книги`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `bib`.`заказ`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bib`.`заказ` (

`Код\_заказа` INT(11) NOT NULL AUTO\_INCREMENT,

`Дата\_заказа` DATE NULL DEFAULT NULL,

`Код\_абонемента` INT(11) NULL DEFAULT NULL,

`Код\_книги` INT(11) NULL DEFAULT NULL,

`ID` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`Код\_заказа`),

INDEX `R\_47` (`Код\_абонемента` ASC) VISIBLE,

INDEX `R\_53` (`Код\_книги` ASC) VISIBLE,

CONSTRAINT `заказ\_ibfk\_1`

FOREIGN KEY (`Код\_абонемента`)

REFERENCES `bib`.`абонемент` (`Код\_абонемента`),

CONSTRAINT `заказ\_ibfk\_2`

FOREIGN KEY (`Код\_книги`)

REFERENCES `bib`.`заказанная\_книга` (`Код\_книги`))

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Команды модификации данных**

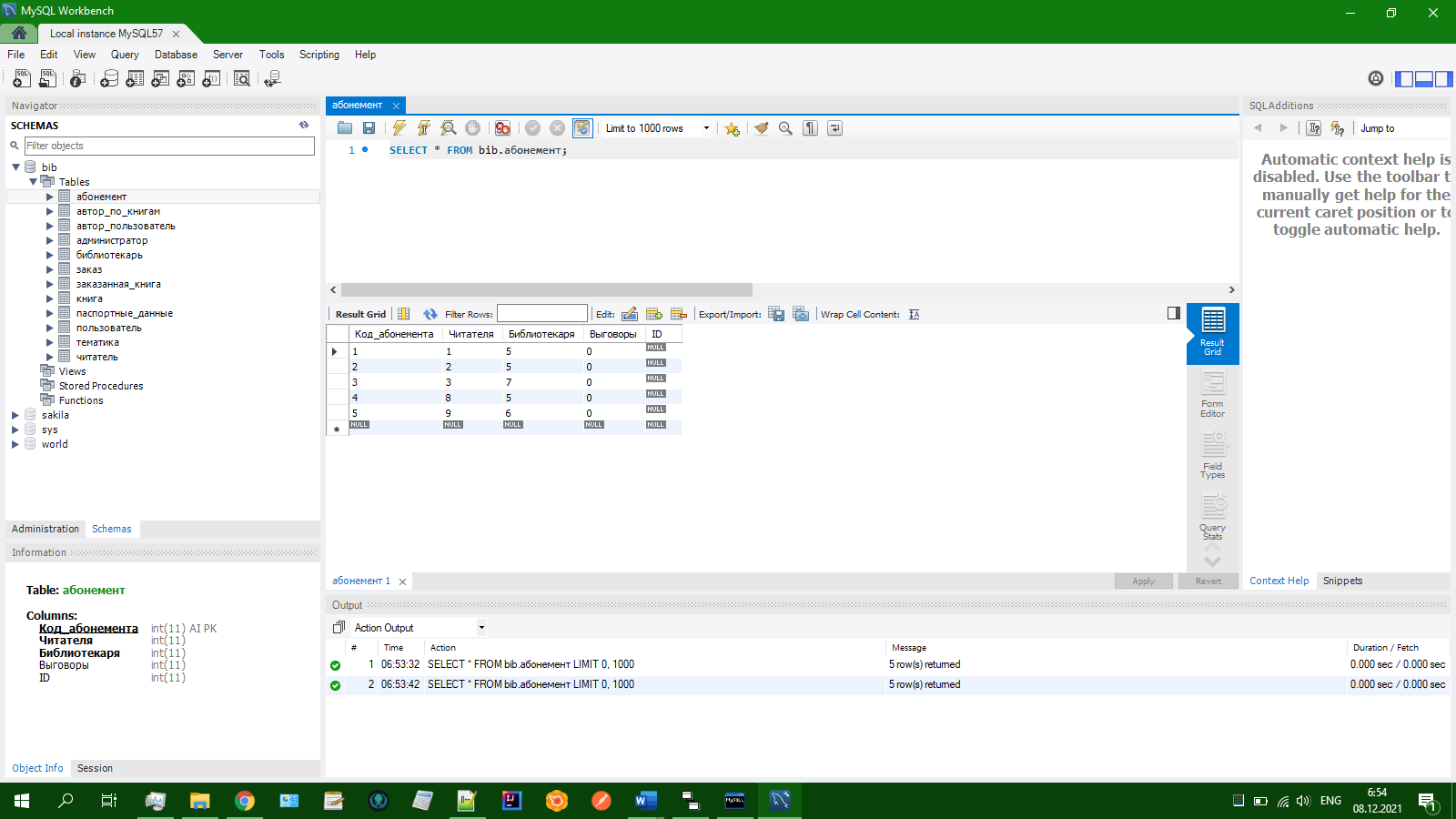


Рисунок 40 – Внесенные данные

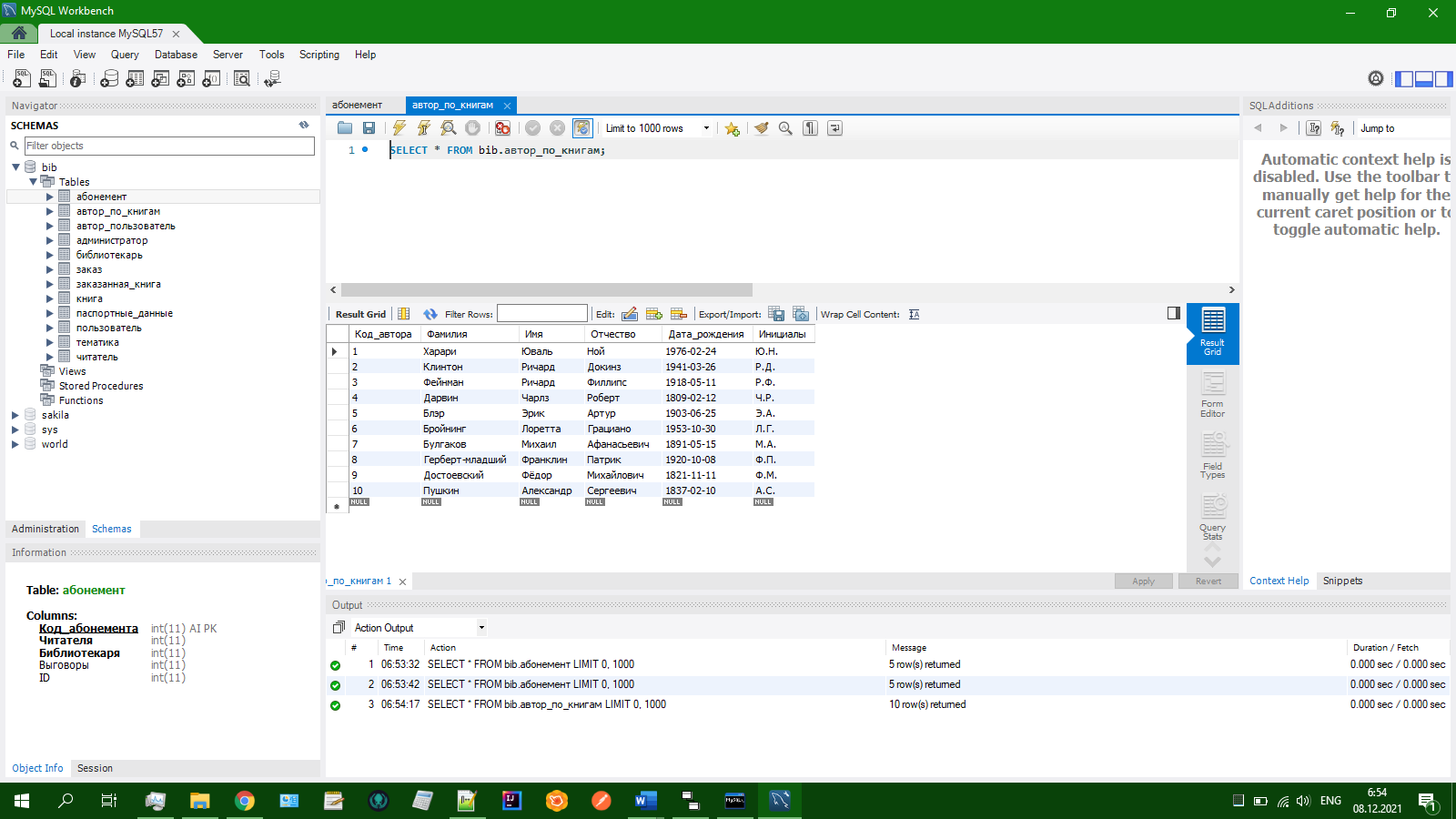


Рисунок 41 – Внесенные данные

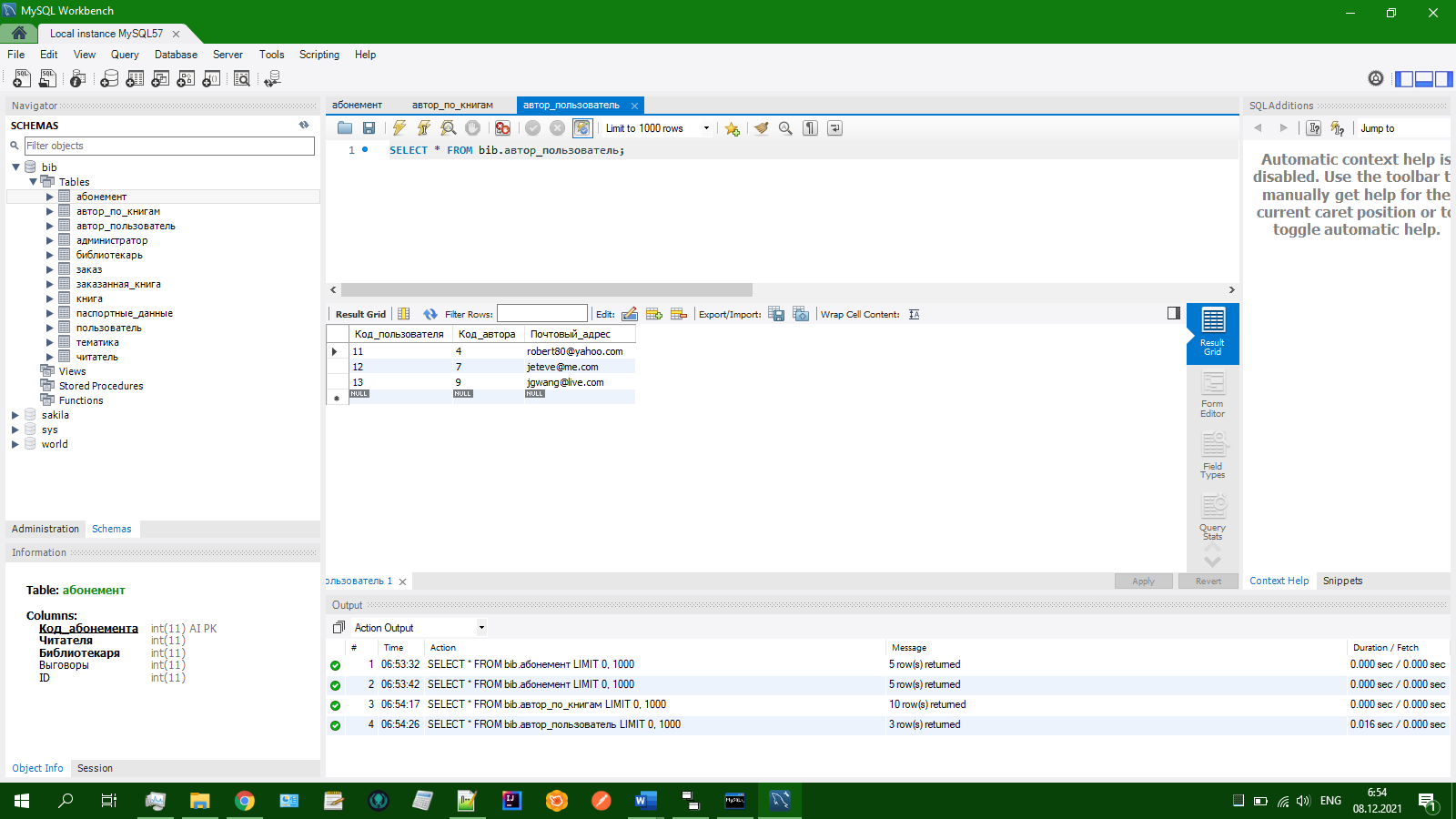


Рисунок 42 – Внесенные данные

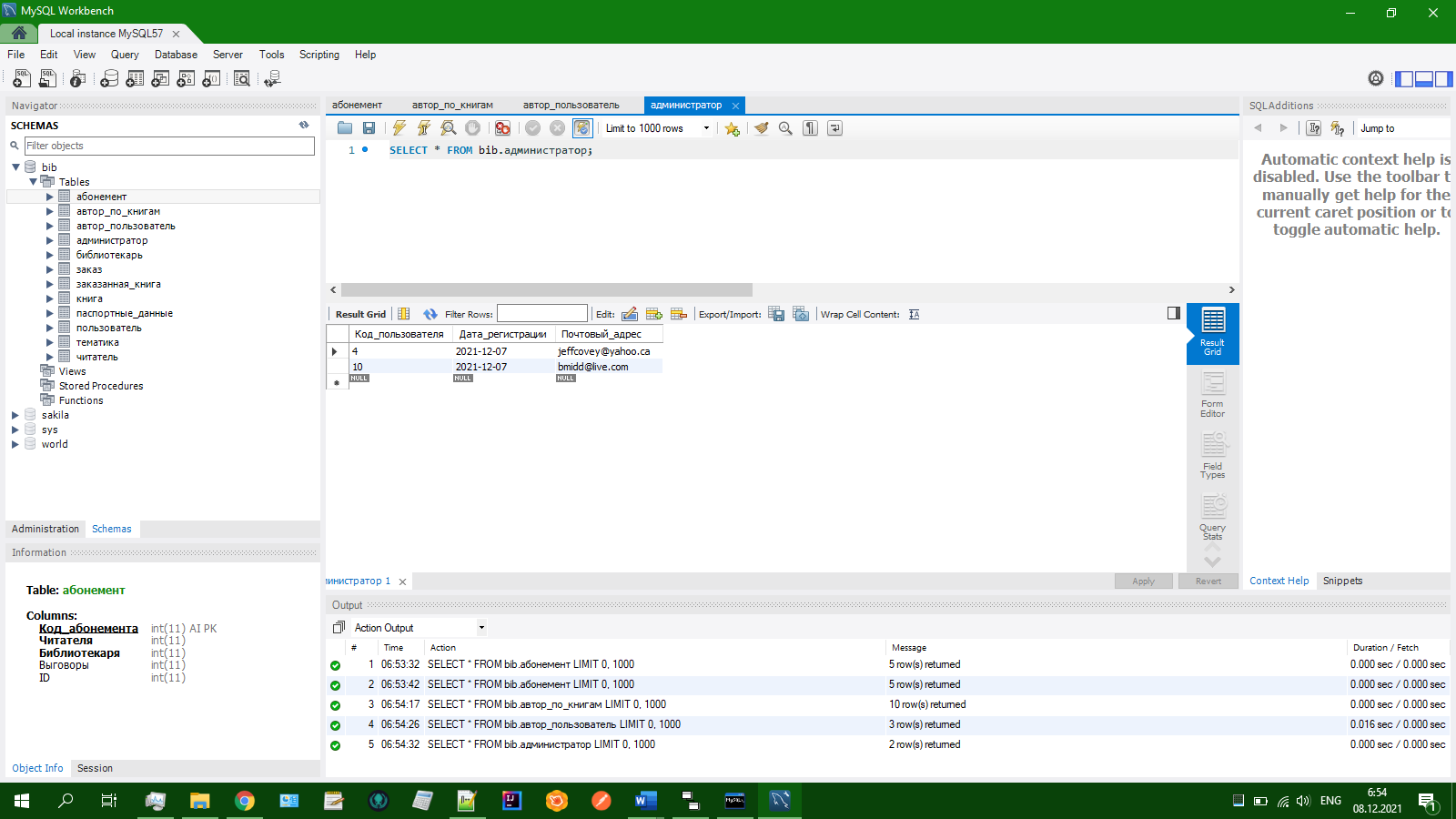


Рисунок 43 – Внесенные данные

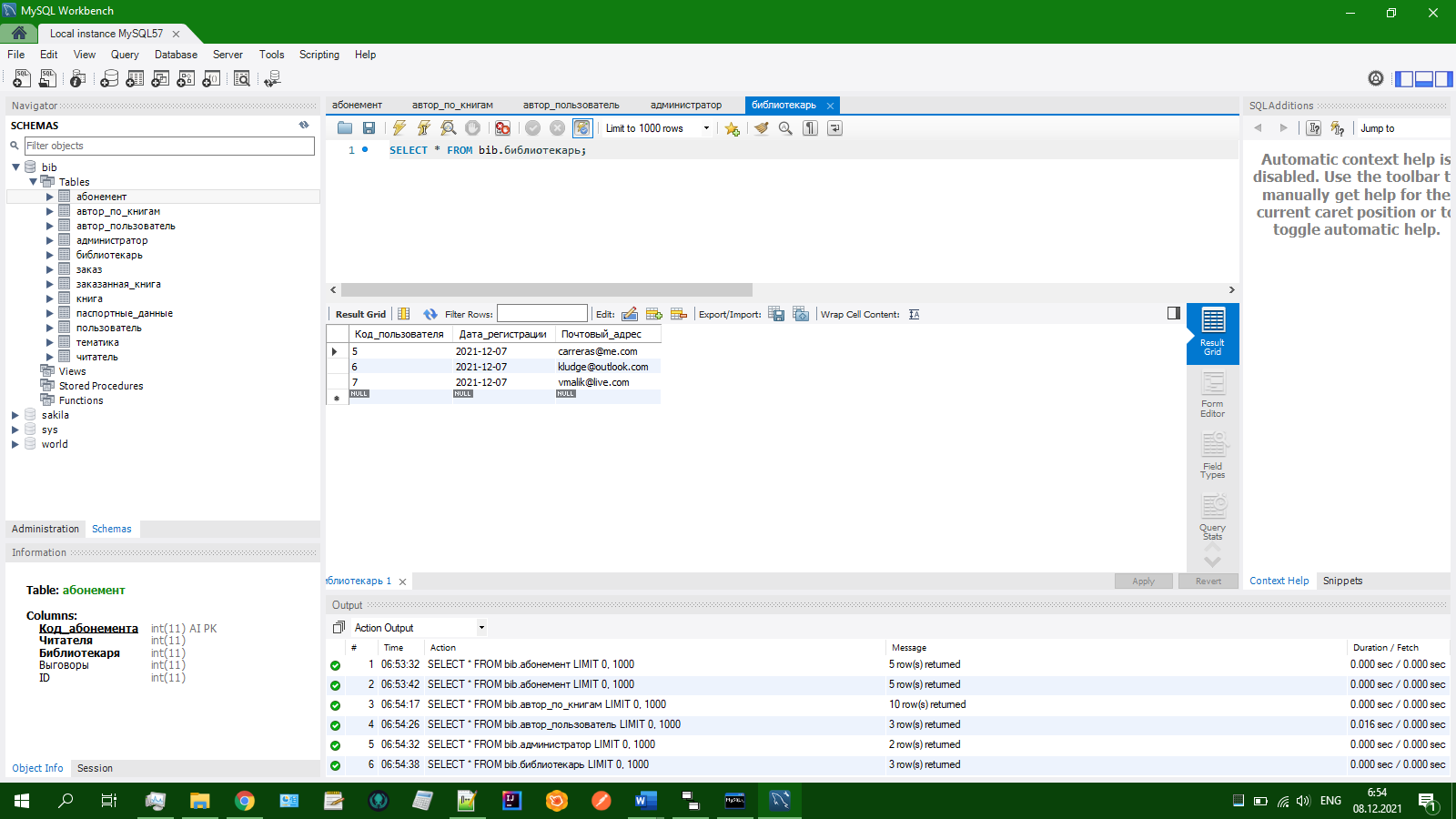


Рисунок 44 – Внесенные данные

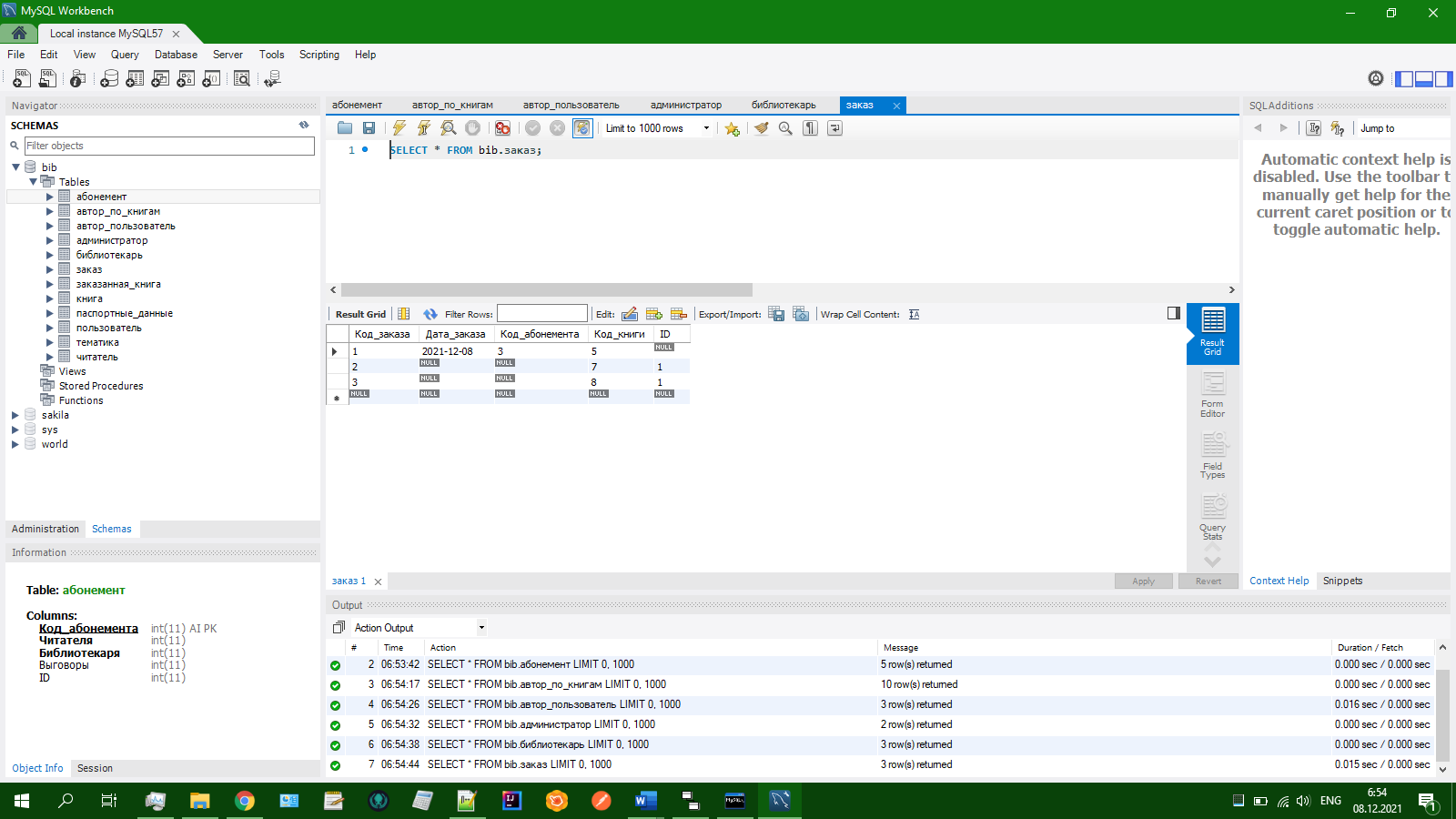


Рисунок 45 – Внесенные данные

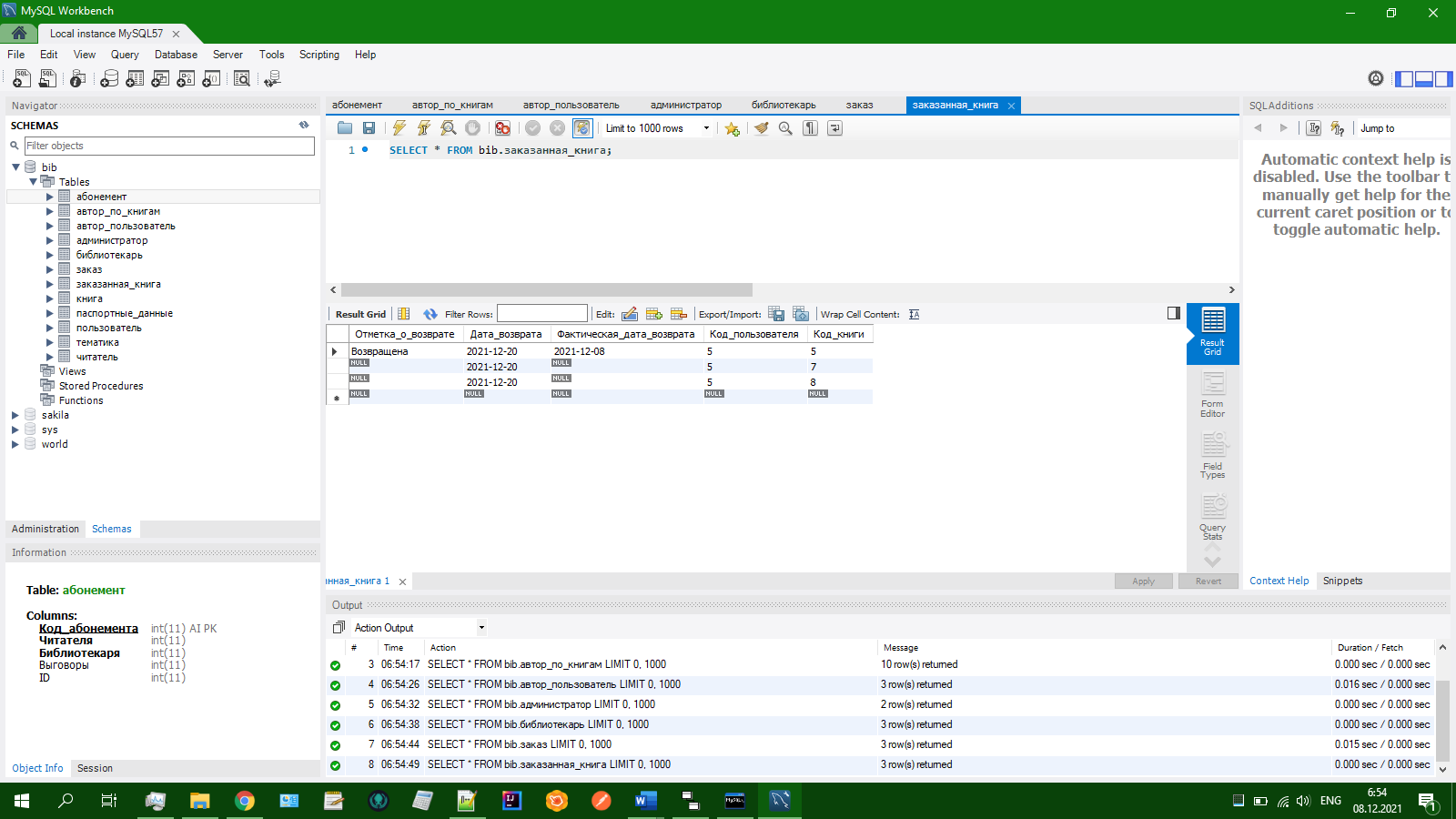


Рисунок 46 – Внесенные данные

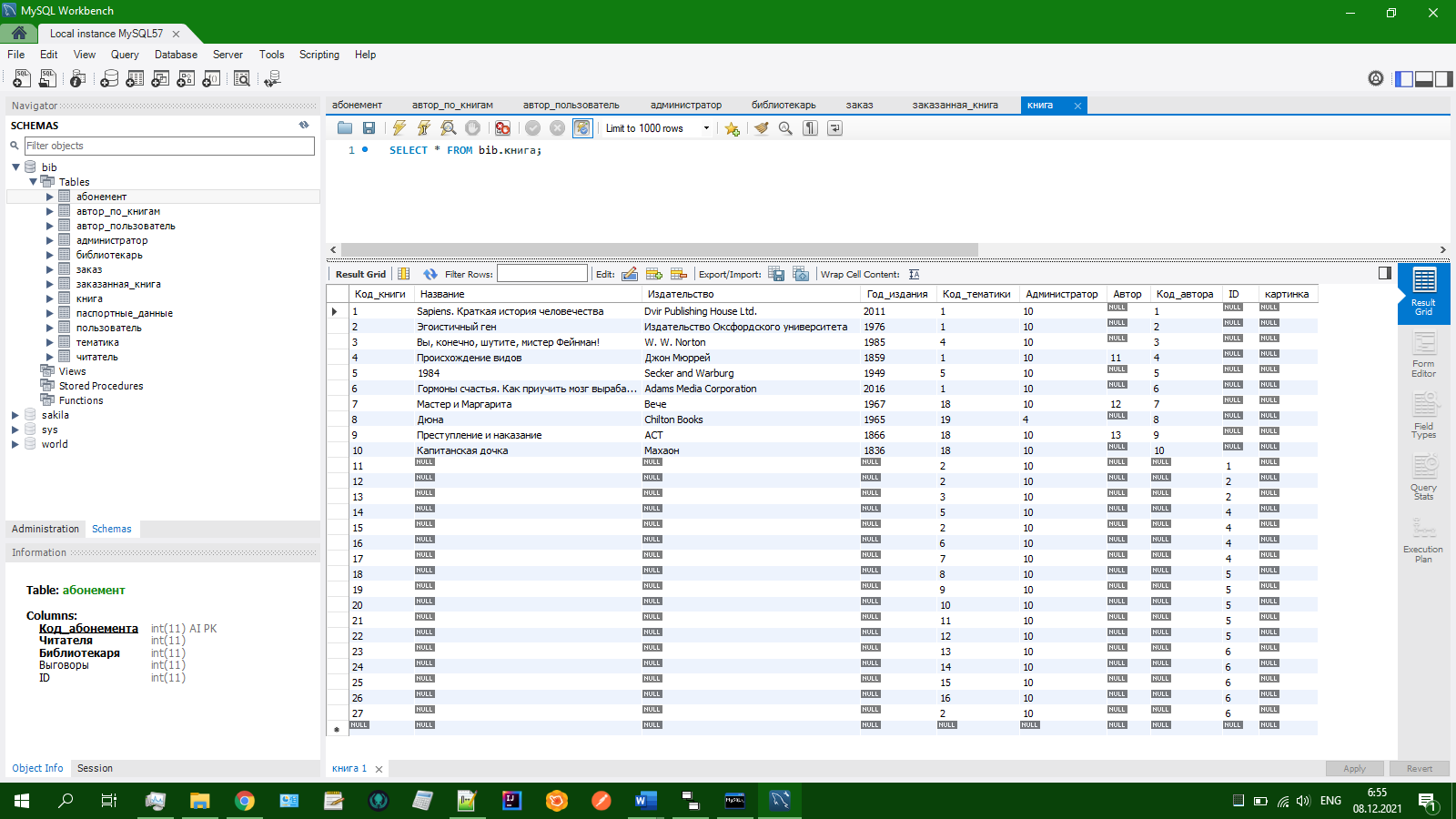


Рисунок 47 – Внесенные данные

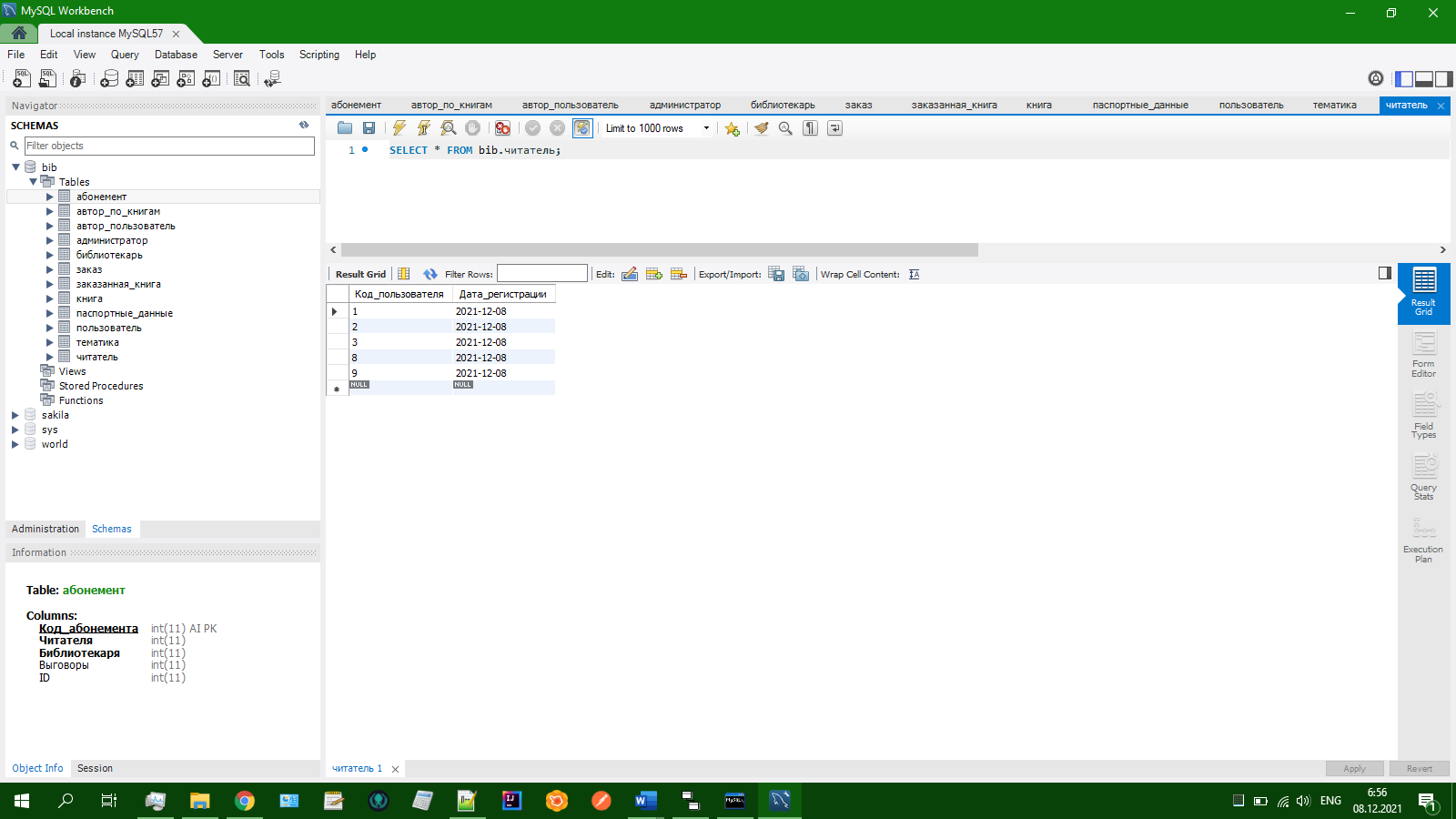


Рисунок 48 – Внесенные данные

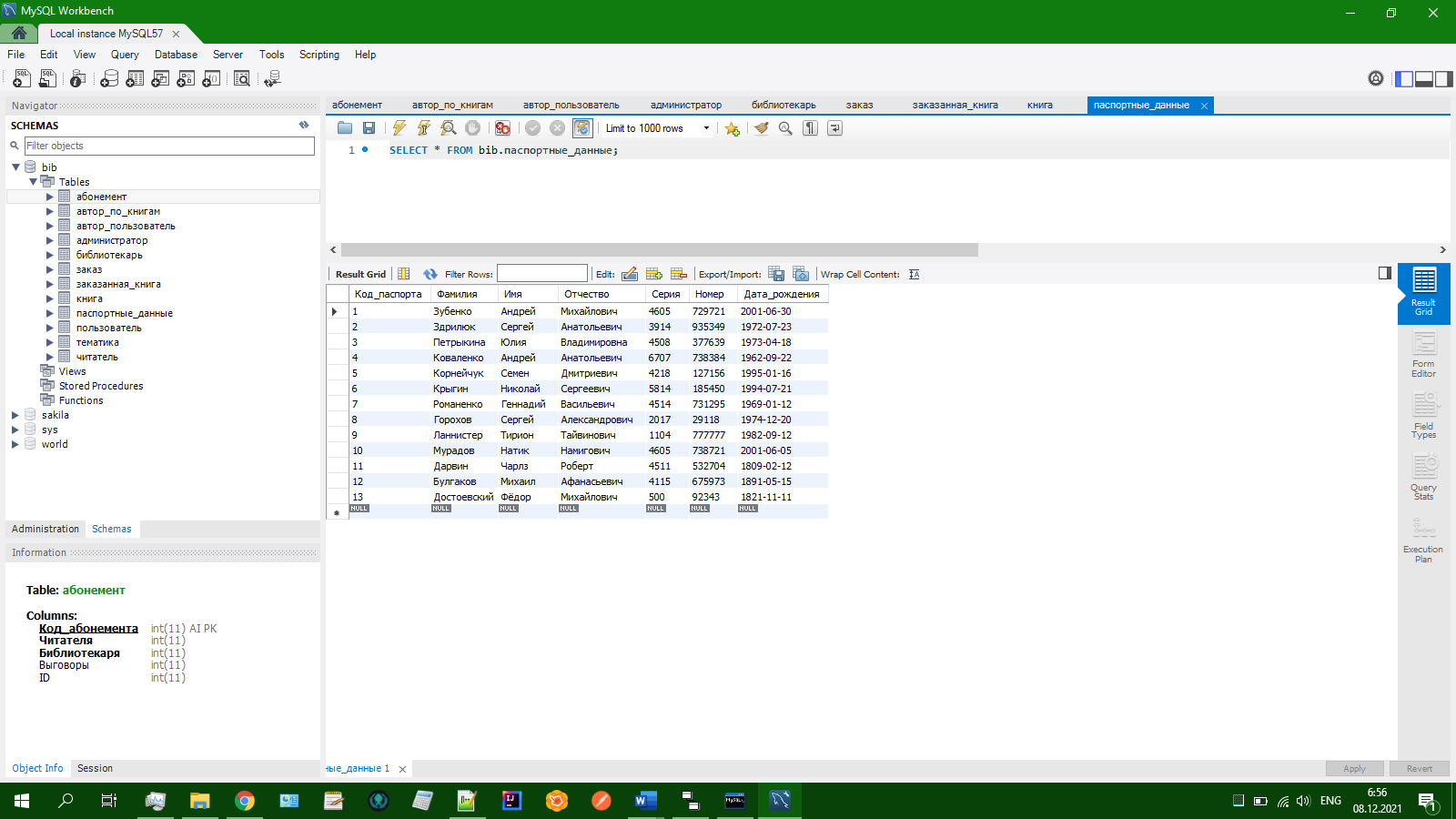


Рисунок 49 – Внесенные данные

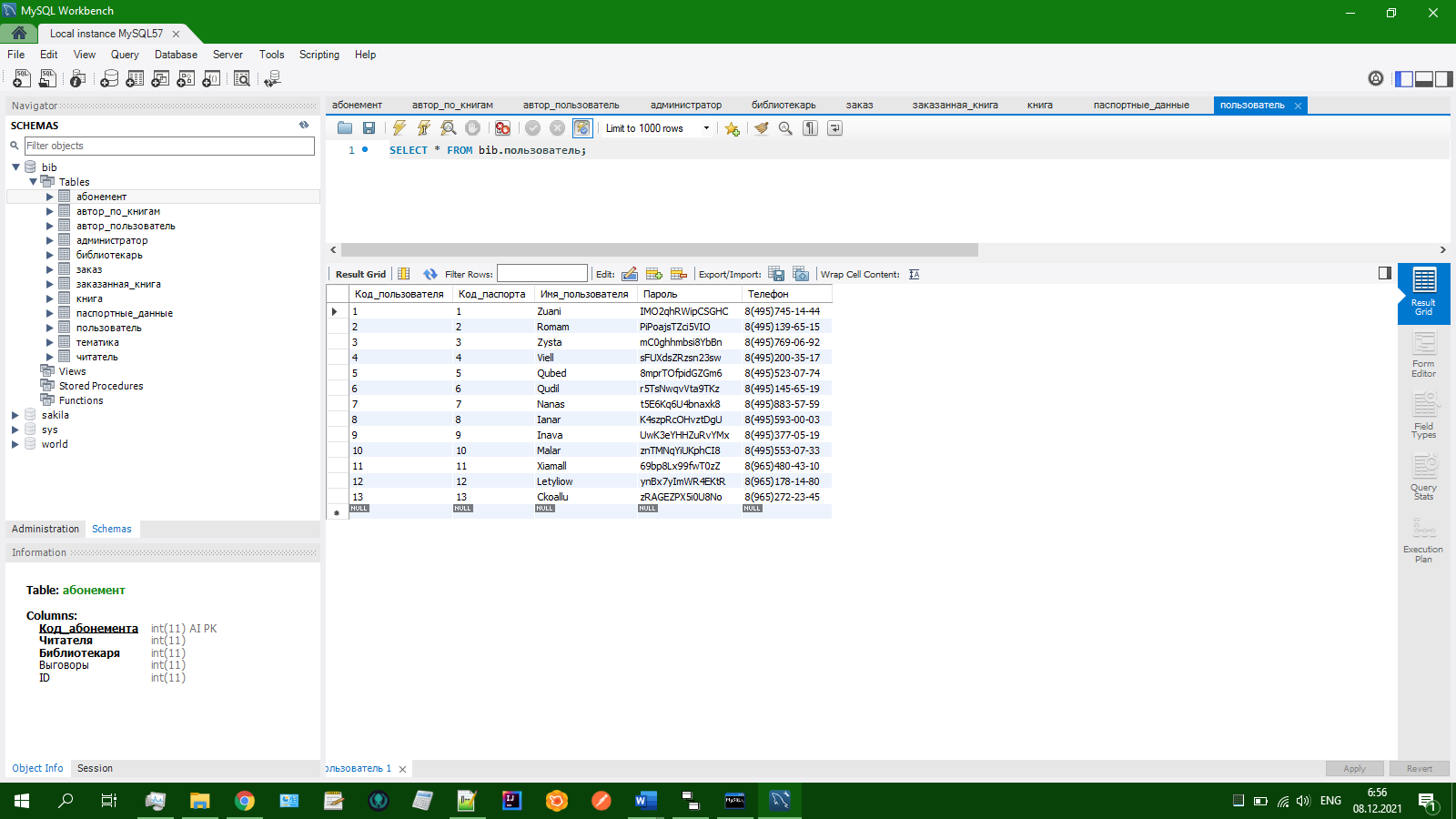


Рисунок 50 – Внесенные данные

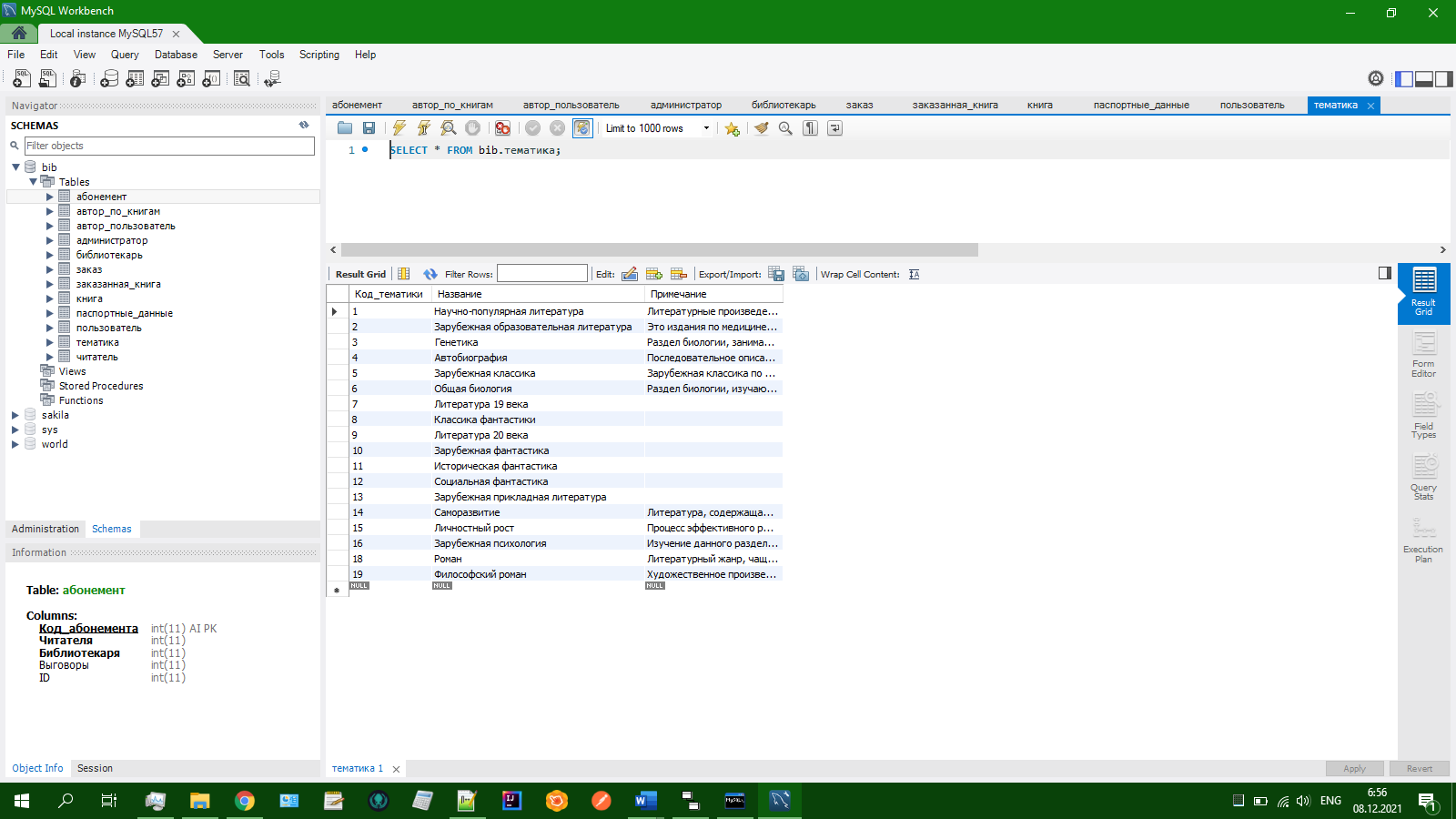


Рисунок 51 – Внесенные данные

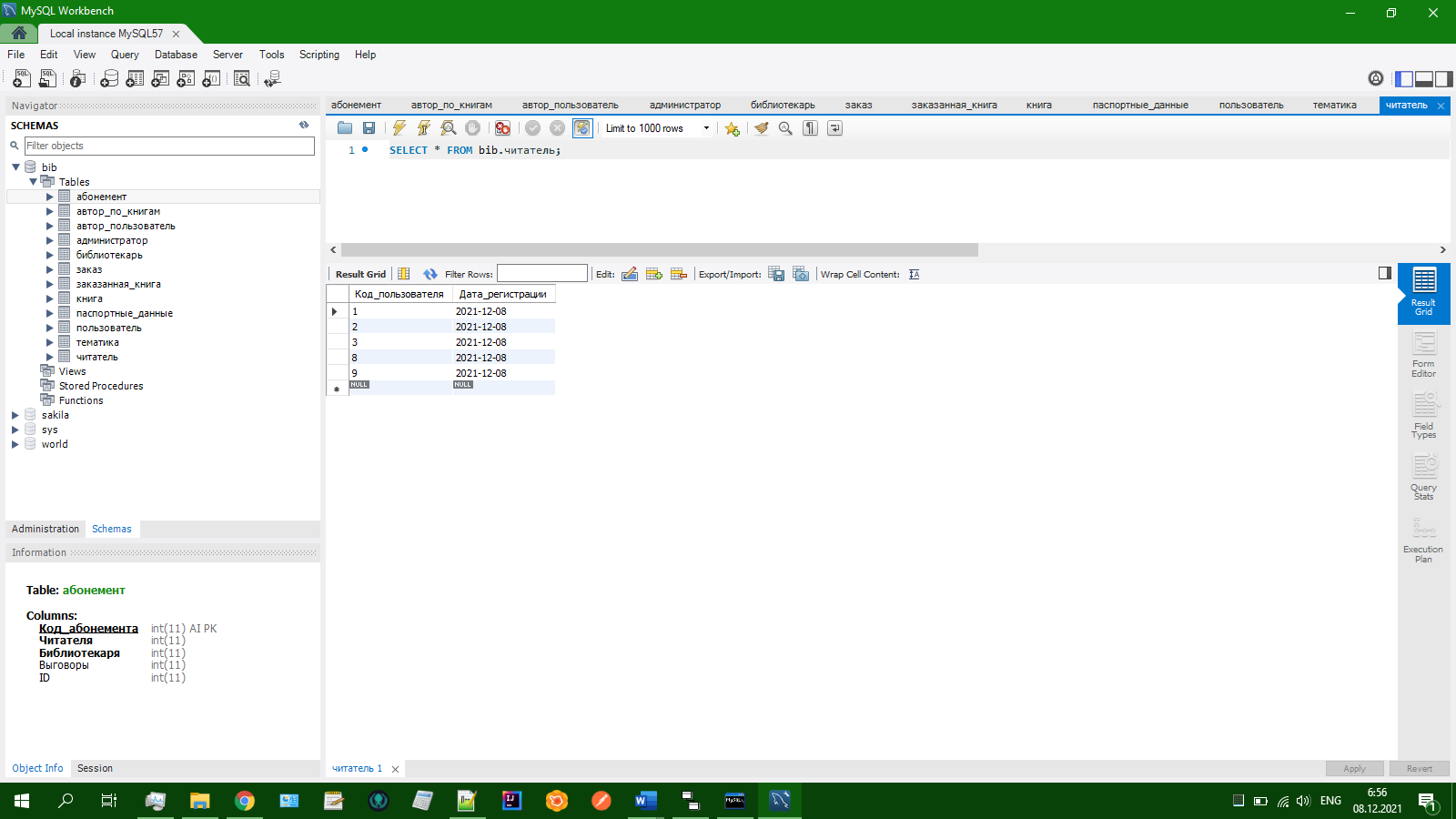


Рисунок 52 – Внесенные данные

**Выборка данных. Оператор SELECT**

Операция проекции (Рисунок 53)

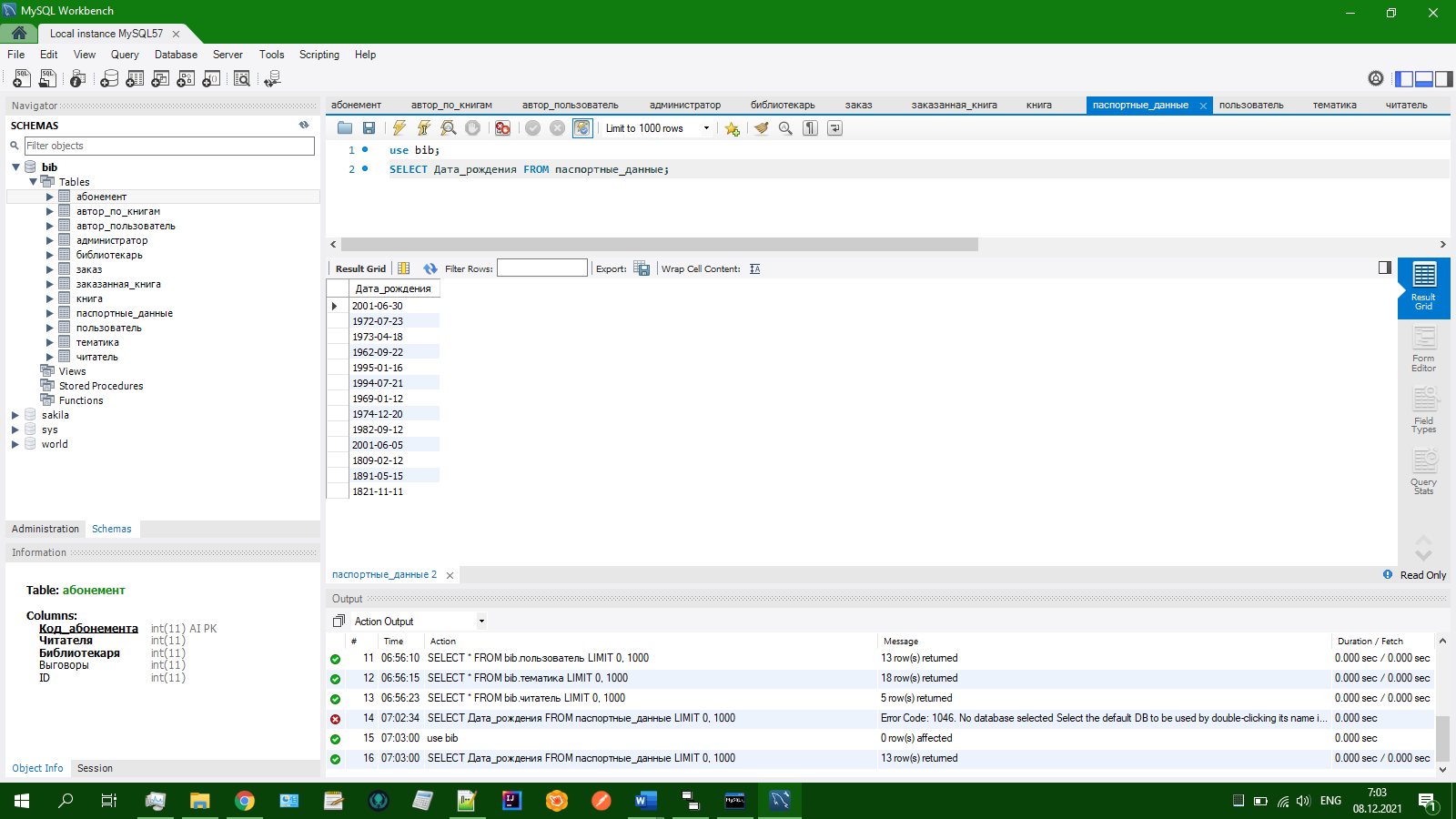


Рисунок 53 – Проекция

Операция селекции (Рисунок 54)

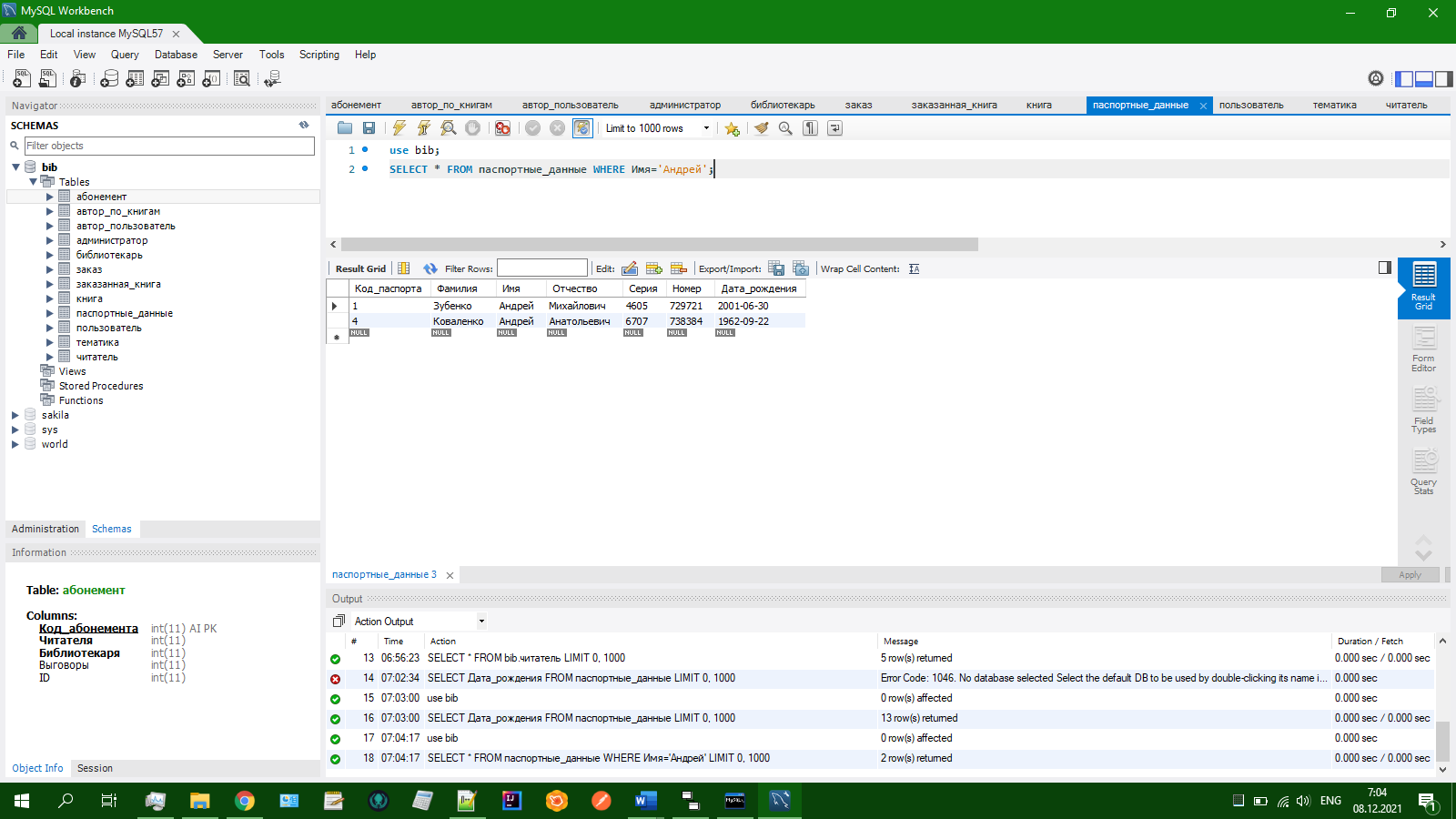


Рисунок 54 – Селекция

Операция соединения(Рисунок 55)

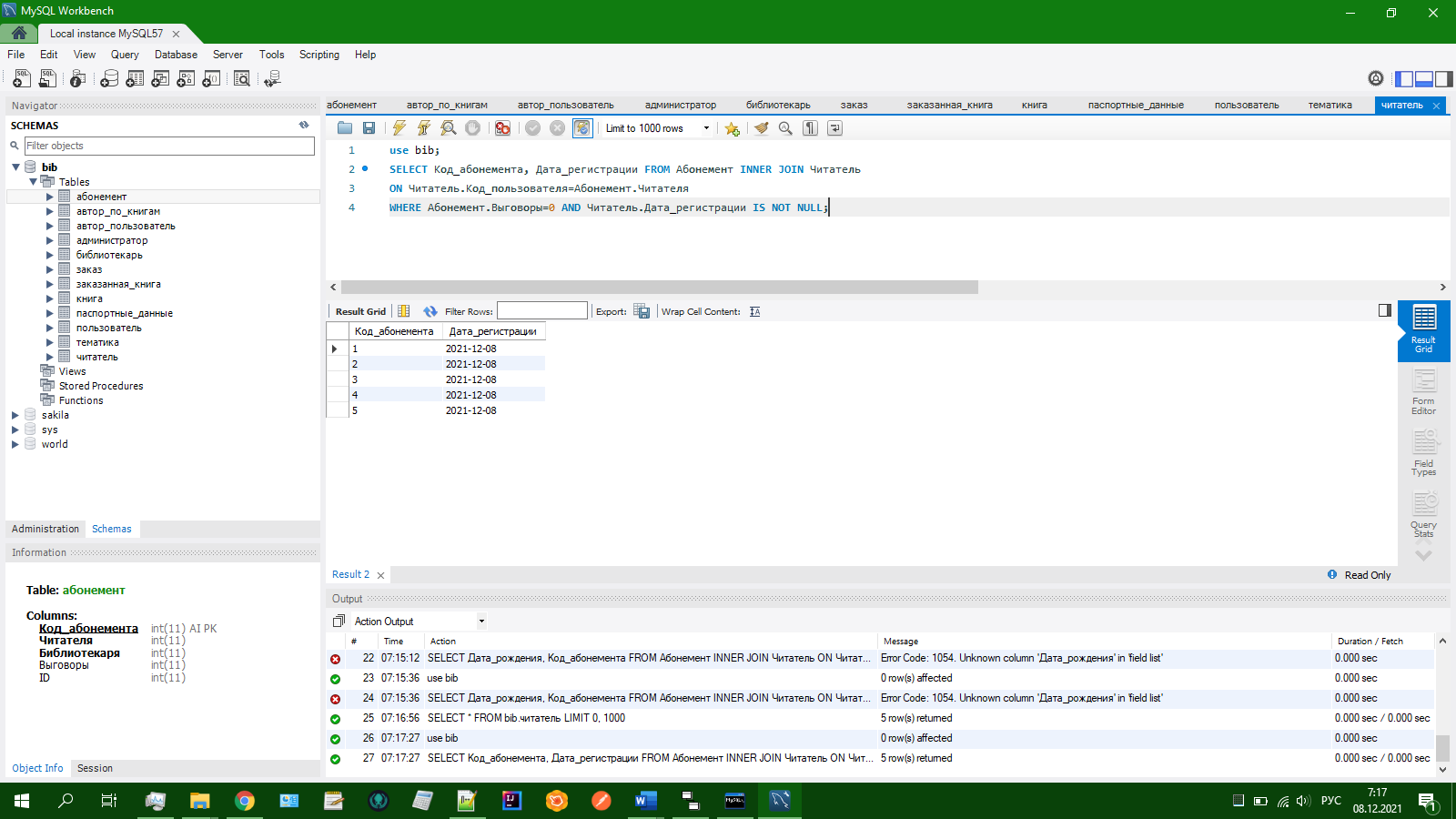


Рисунок 55 – Соединение

Операция объединения(Рисунок 56)

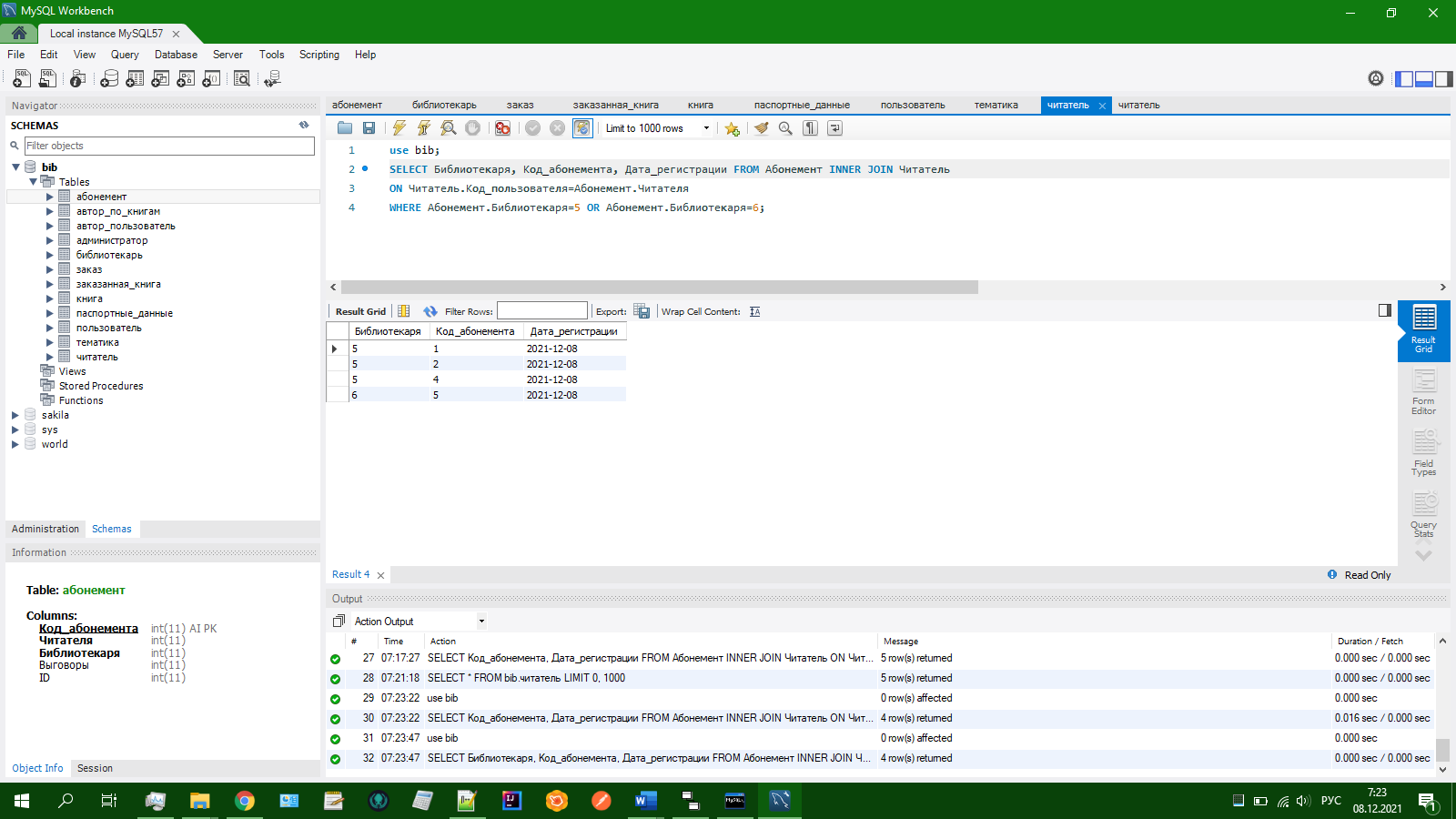


Рисунок 56 – Объединение

**Создание триггеров.**

Был создан триггер, который при пустом столбце серия, автоматически его заполняет данными.

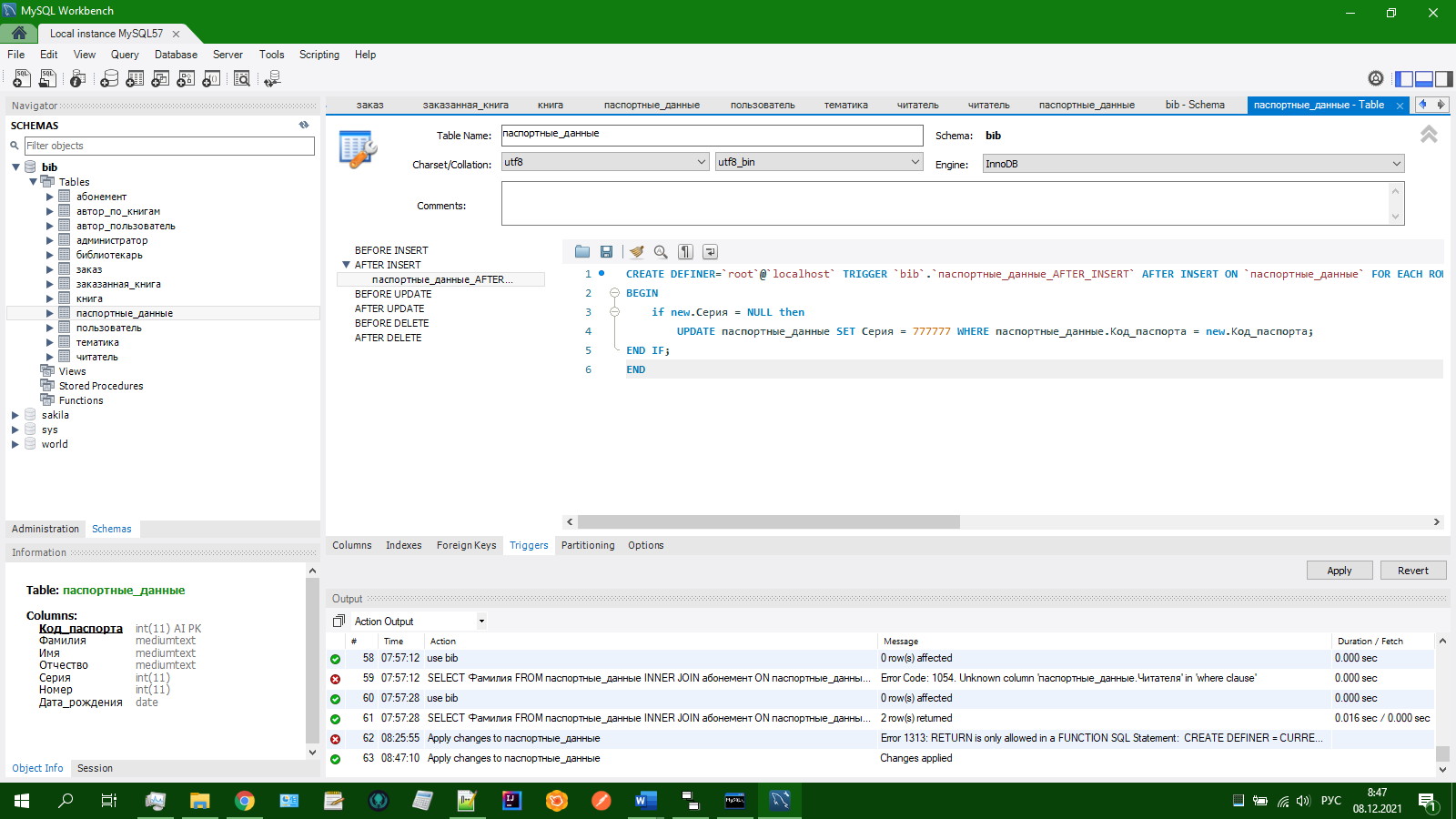


Рисунок 57 ‒ Код триггера

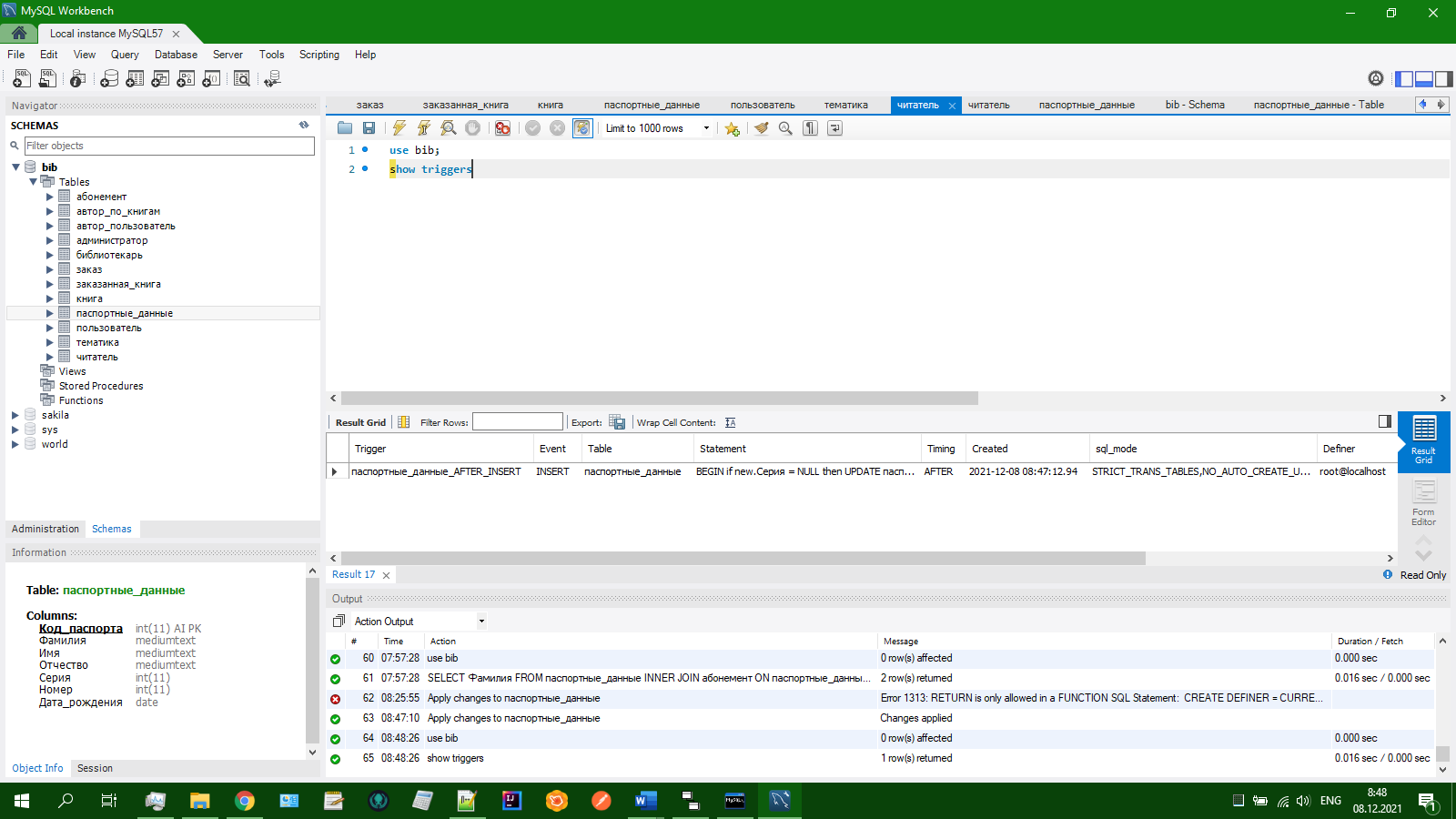


Рисунок 58 ‒ Хранение триггера

**Создание хранимых процедур**

Производится операция соединения при помощи хранимой процедуры

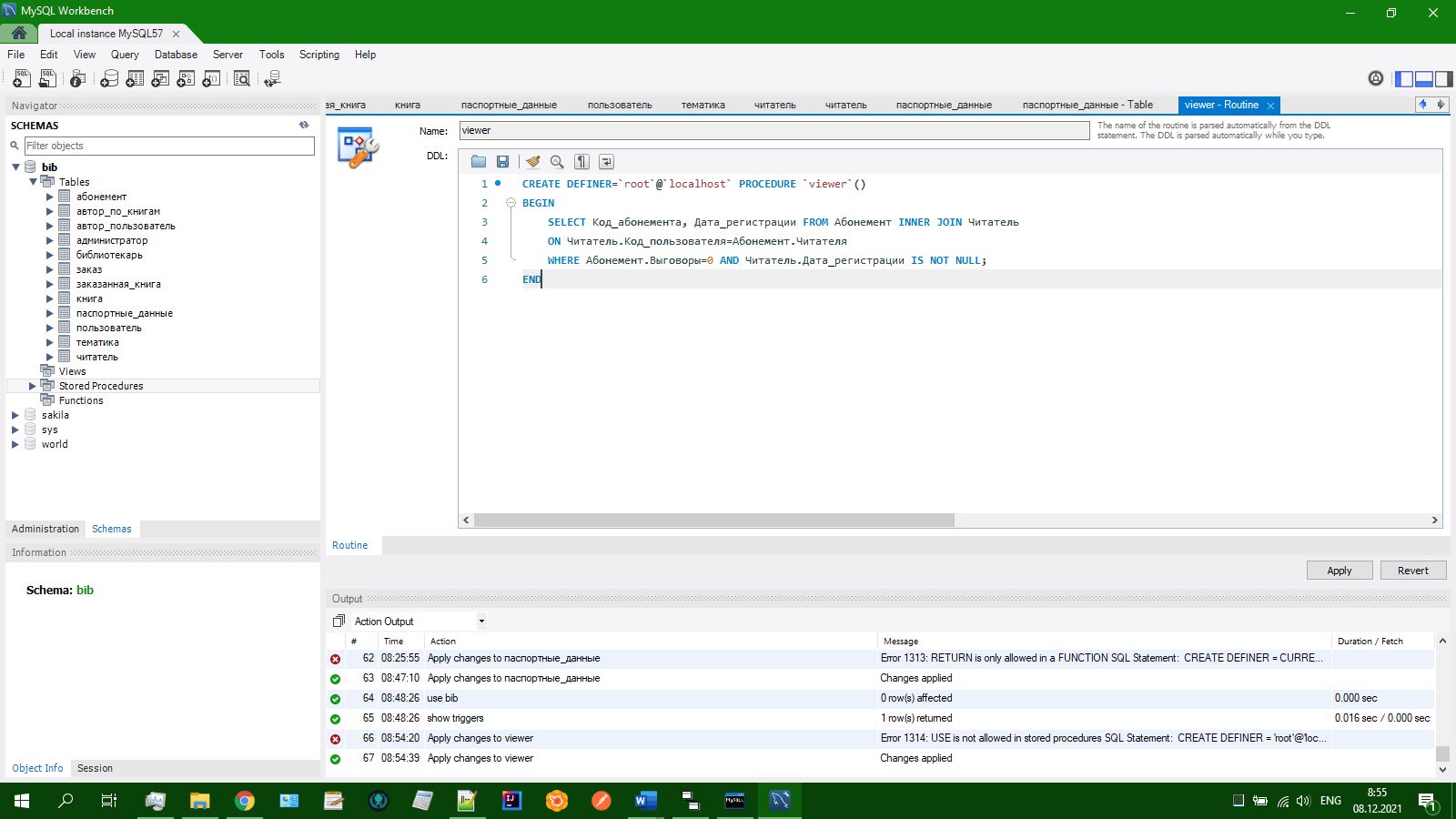


Рисунок 59 ‒ Код процедуры

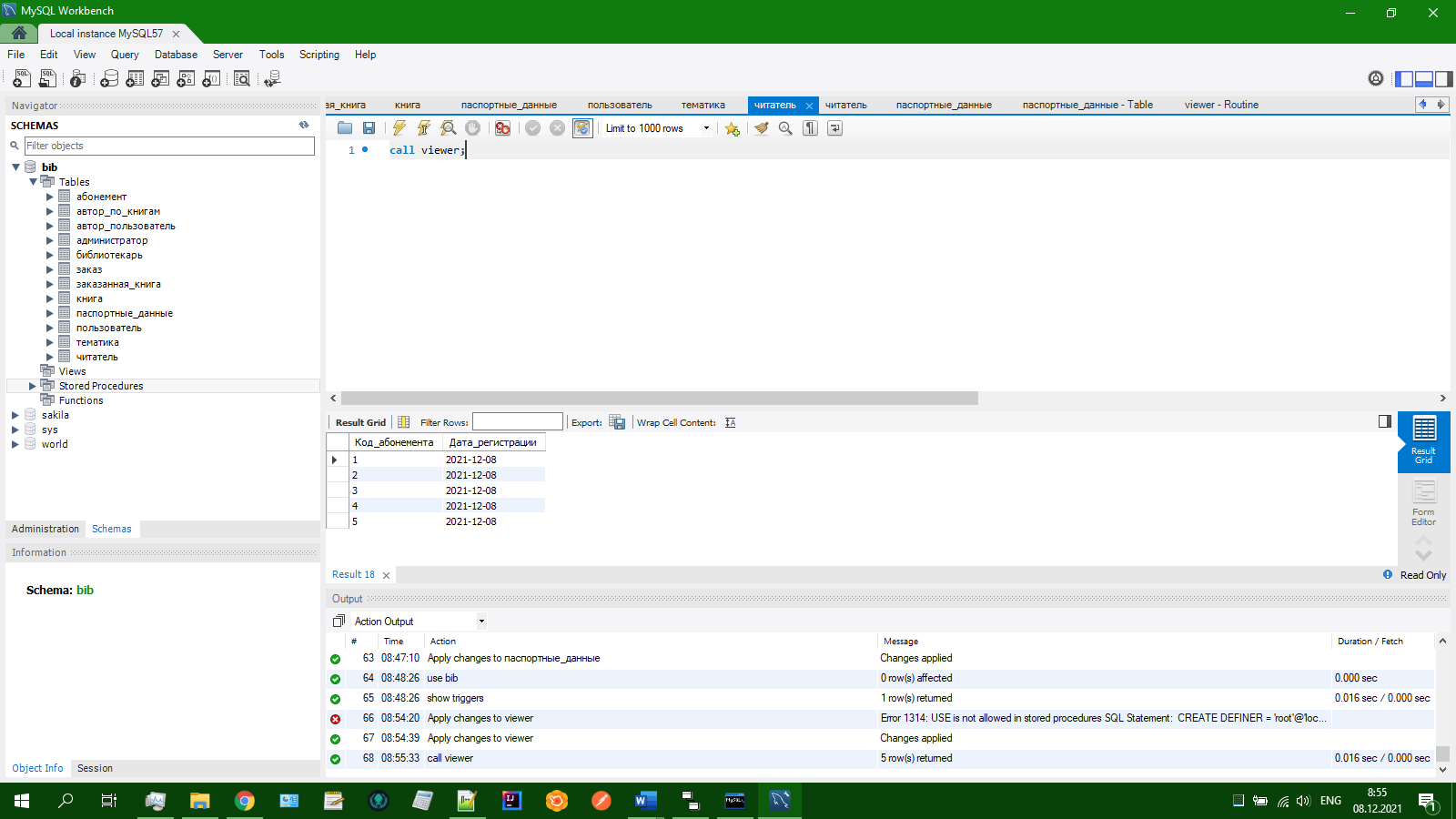


Рисунок 60 ‒ Вызов процедуры

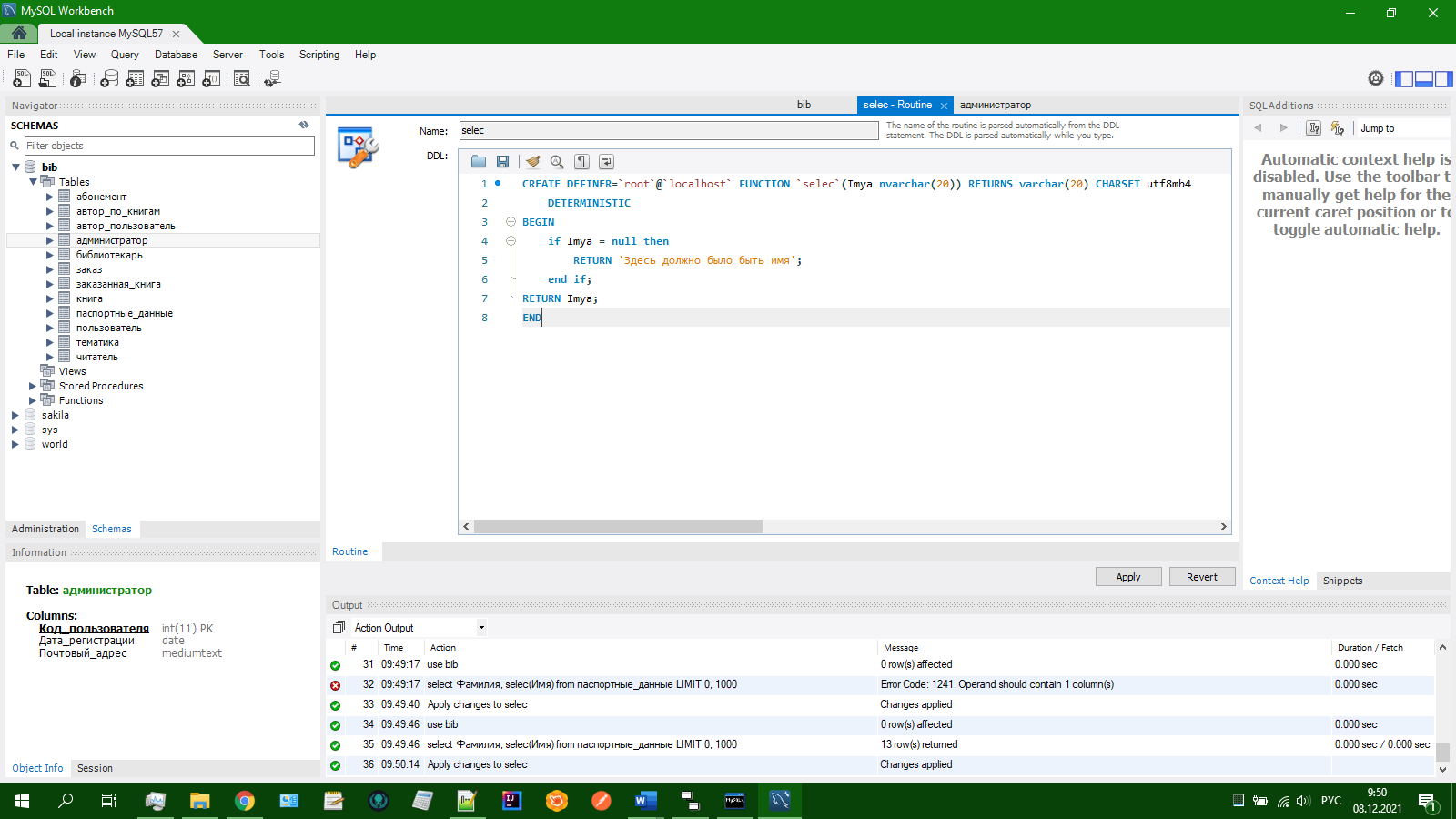


Рисунок 61 ‒ Код функции

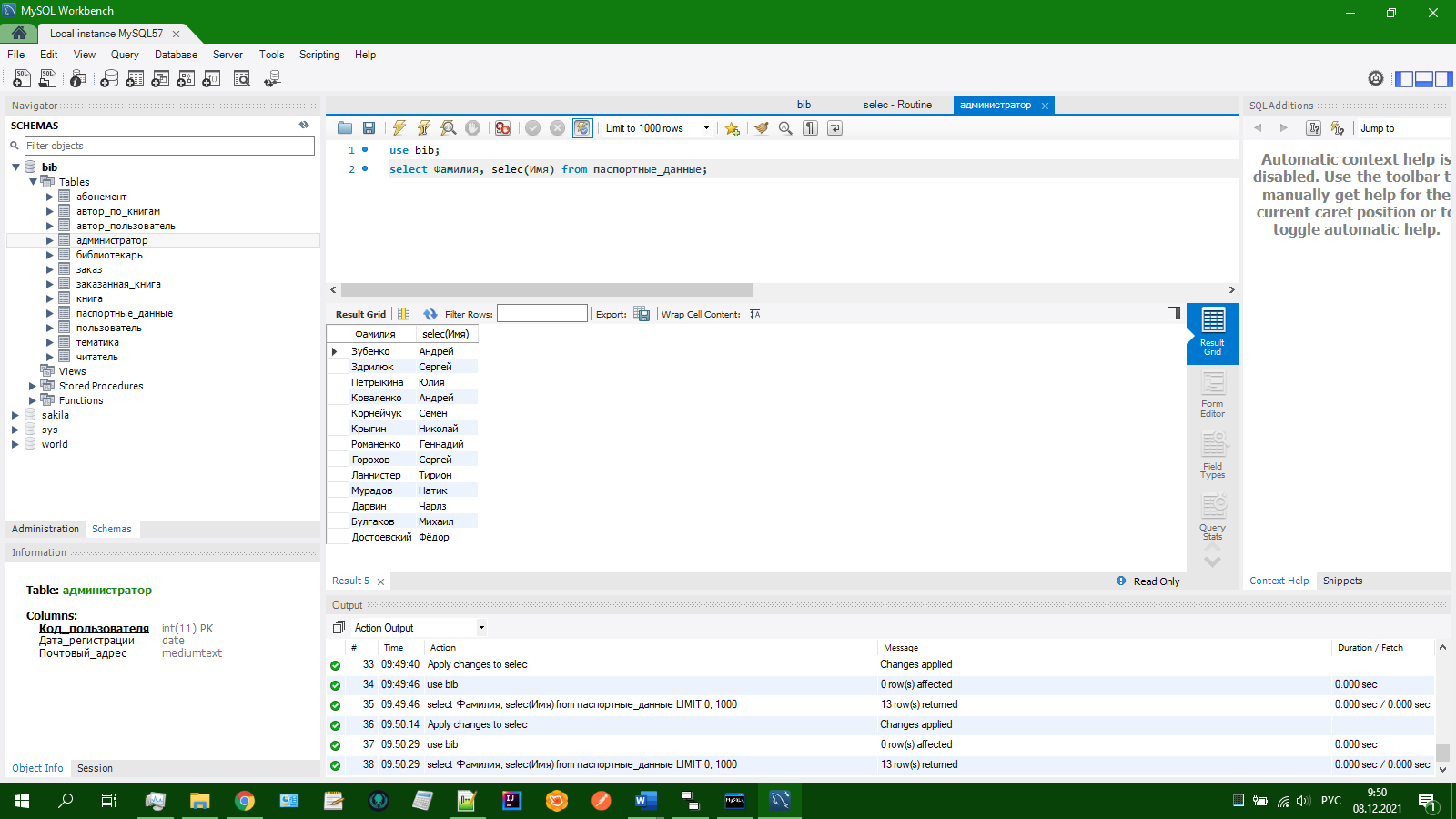


Рисунок 62 ‒ Вызов функции

**Резервное копирование**

Здесь было выполнено копирование базы данных bib:

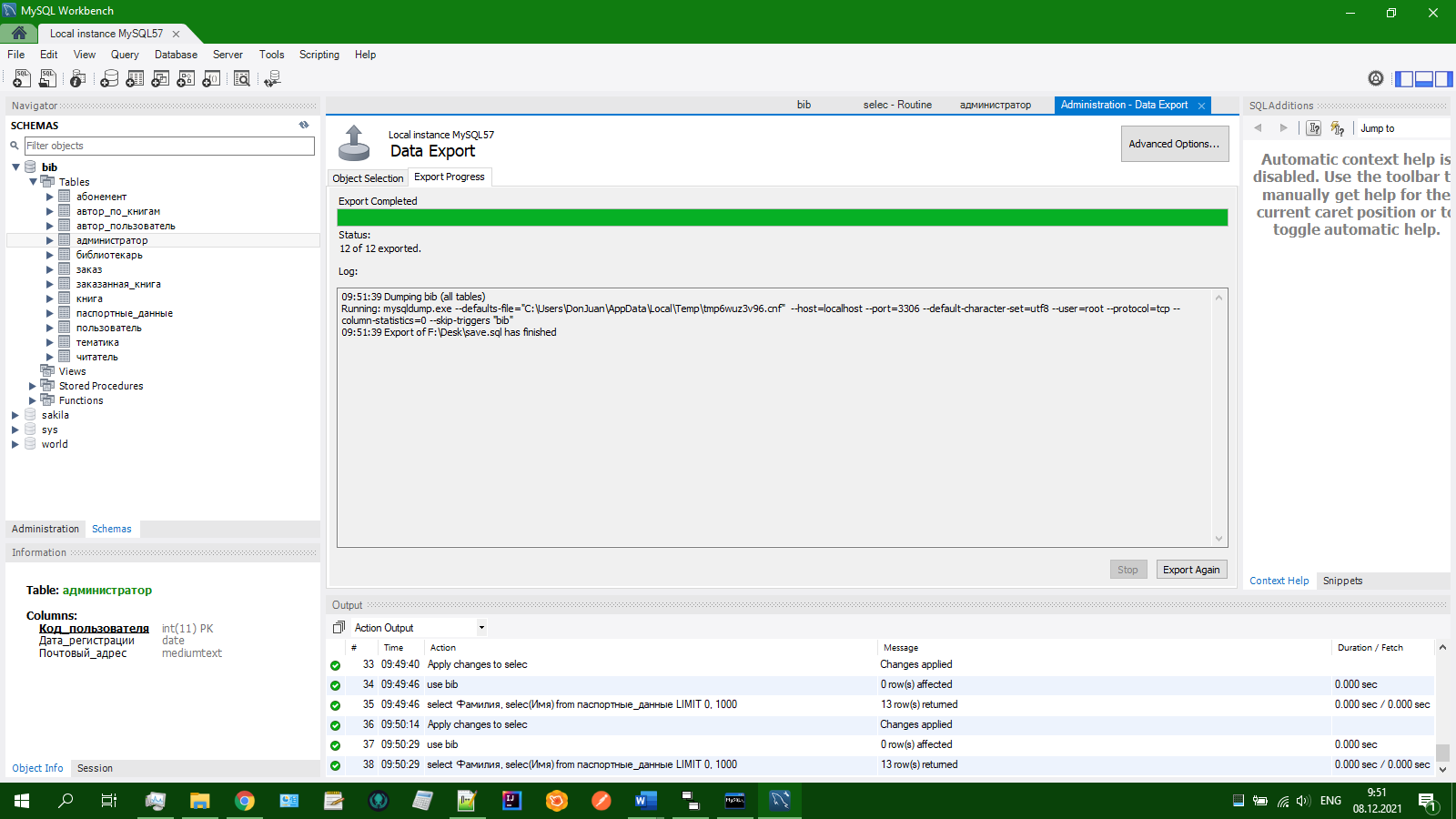


Рисунок 63 ‒ Уведомление об успешном создании резервной копии

Вывод

В ходе выполнения данной работы были получены базовые навыки:

1. По проектированию логической и физической моделей баз данных
2. По написанию базы данных в командной строке MySQL CommandLine
3. По работе с запросами внутри базы данных
4. Работы с MySQL Workbench

Список использованных источников и литературы:

1. Лекции по предмету «Разработка баз данных» Богомольной Г.В.
2. Владимир Репин. Бизнес-процессы: моделирование, внедрение, управление – Москва: Живой язык, 2020. – 470 с.
3. Ульман, Л. MySQL / Л. Ульман. — Москва : ДМК Пресс, 2008. — 352 с. — ISBN 5-94074-229-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/1241 (дата обращения: 06.12.2021). — Режим доступа: для авториз. пользователей.
4. Администрирование MySQL : учебное пособие. — 2-е изд. — Москва : ИНТУИТ, 2016. — 233 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100555 (дата обращения: 06.12.2021). — Режим доступа: для авториз. пользователей.
5. Введение в СУБД MySQL : учебное пособие. — 2-е изд. — Москва : ИНТУИТ, 2016. — 228 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100713 (дата обращения: 06.2.2021). — Режим доступа: для авториз. пользователей.