|  |
| --- |
| 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»

**Промежуточная проверка**

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

Москва 2021 г.

**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 3

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

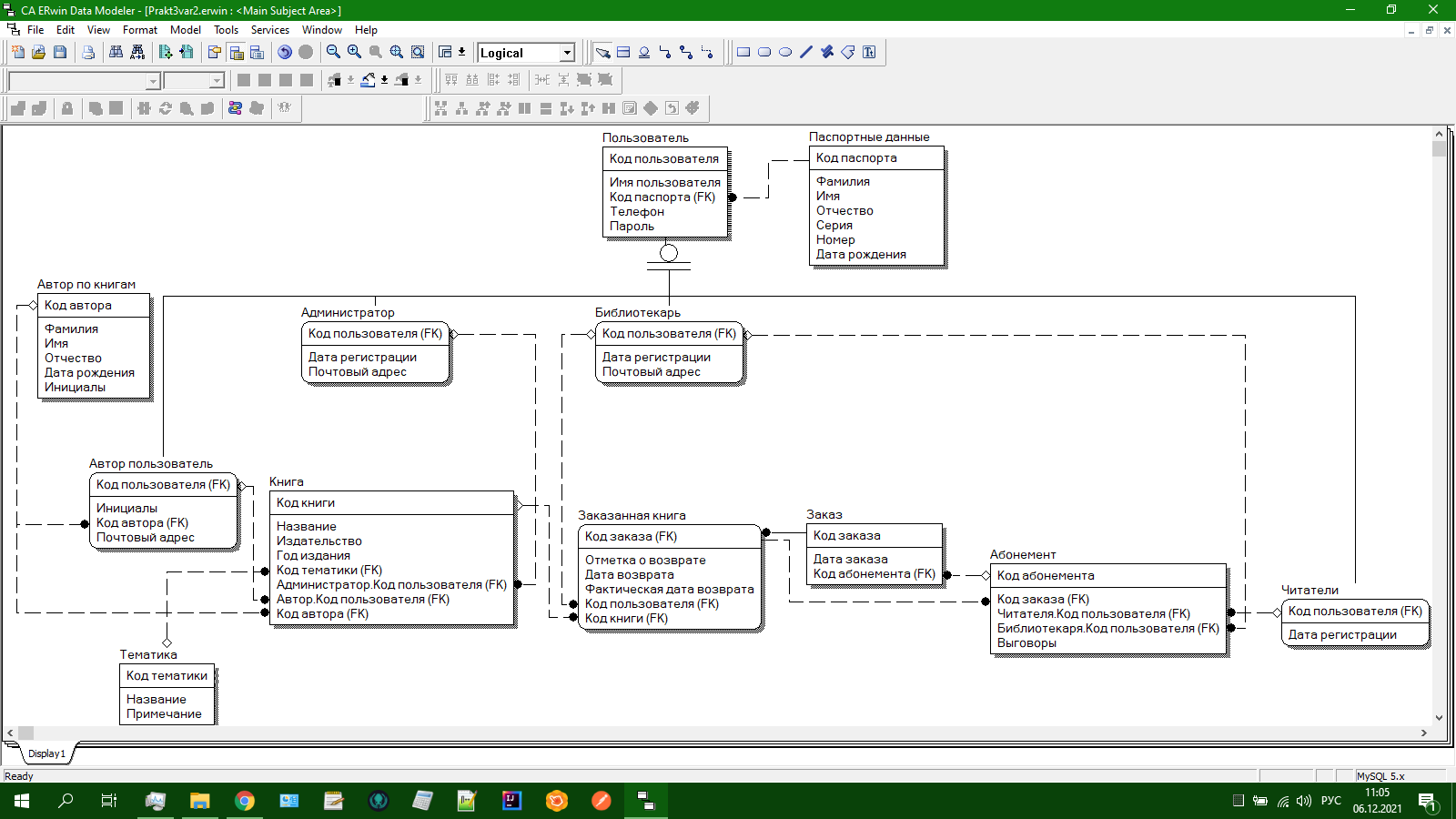


Рис. 1 ‒ Логическая модель данных на тему «Библиотека»

**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»**

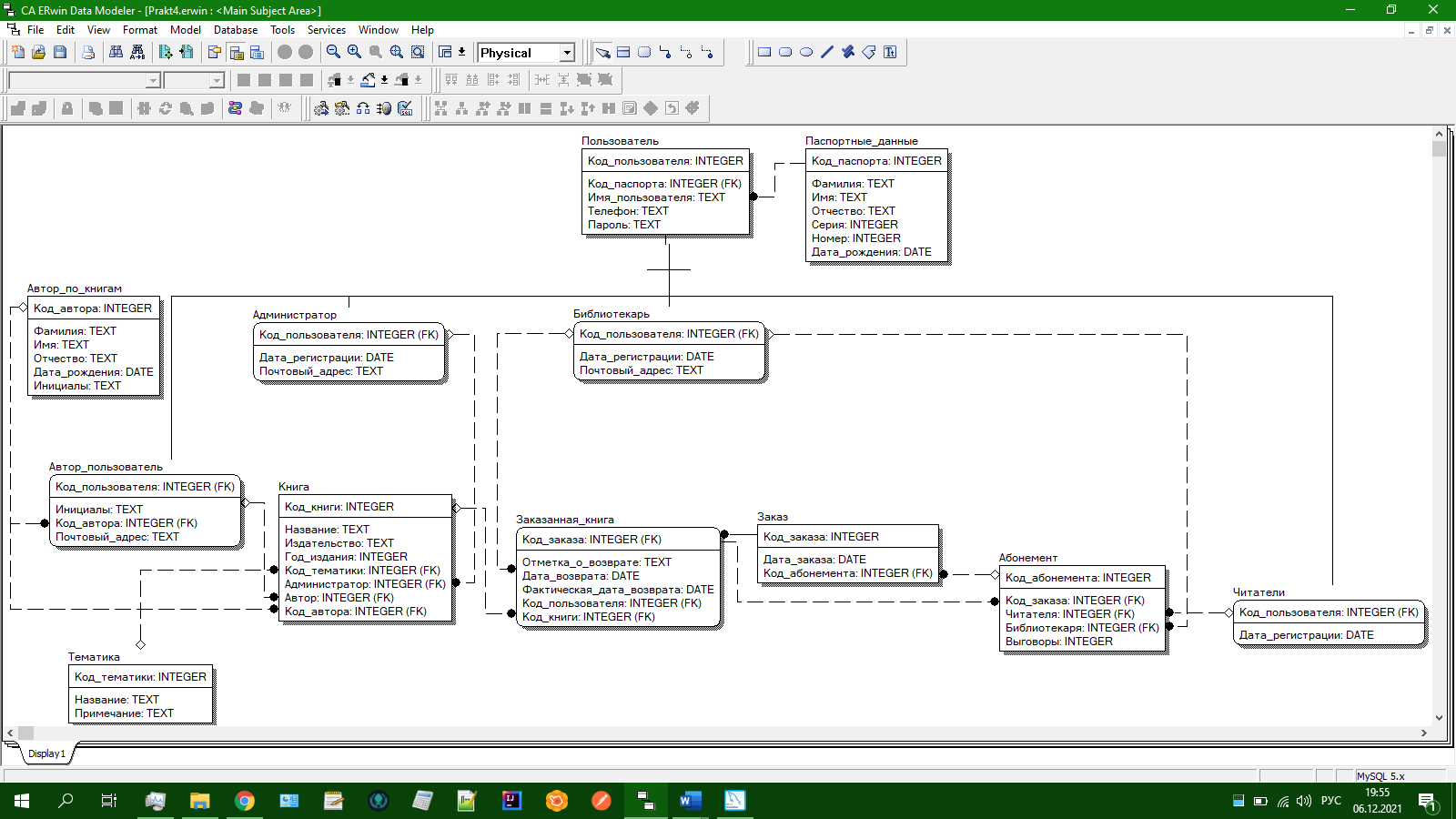


Рис. 2 ‒ Физическая модель данных на тему «Библиотека»

# Практическое задание 5

**Создание своей базы данных в MySQL CommandLine**

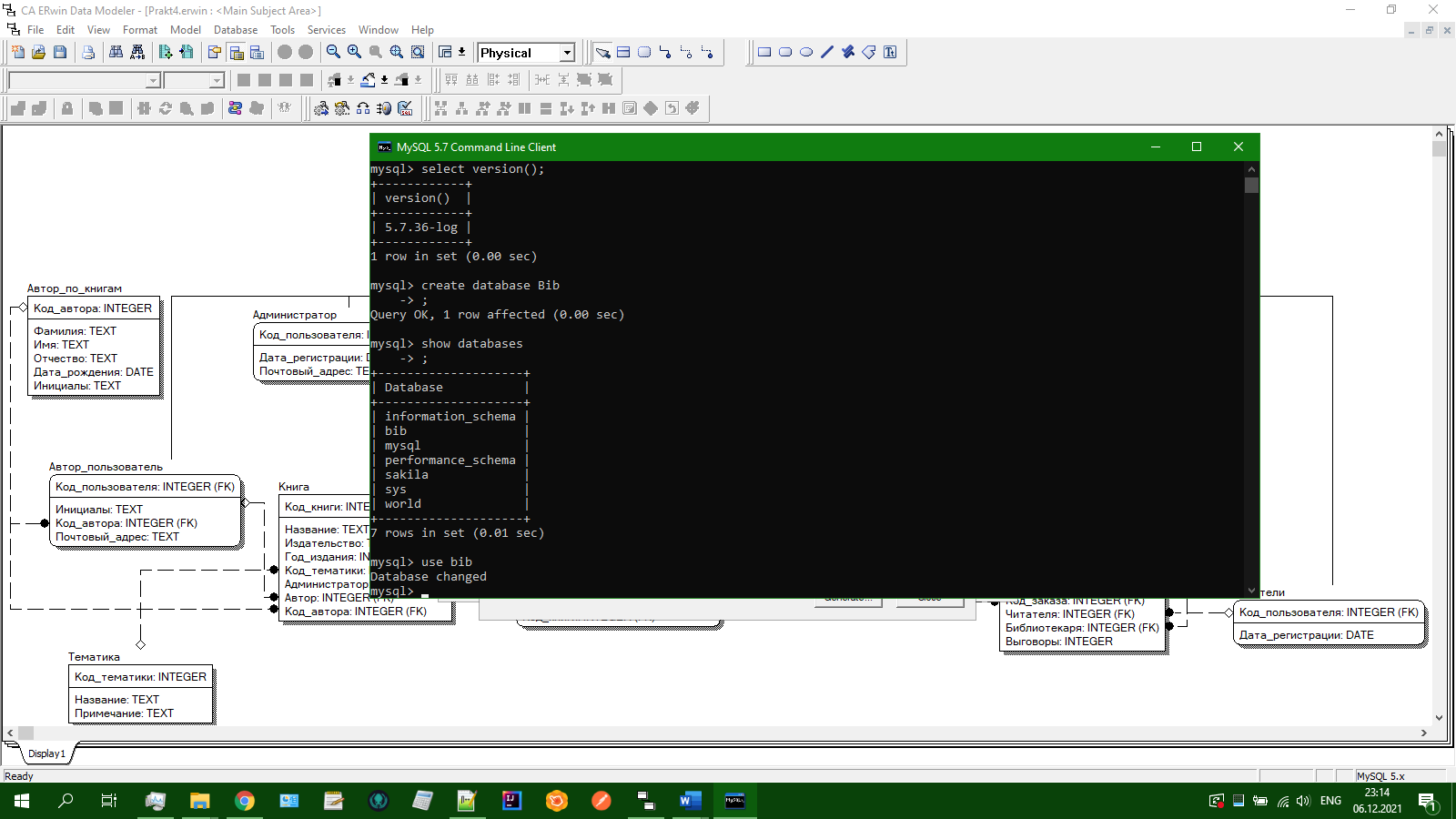


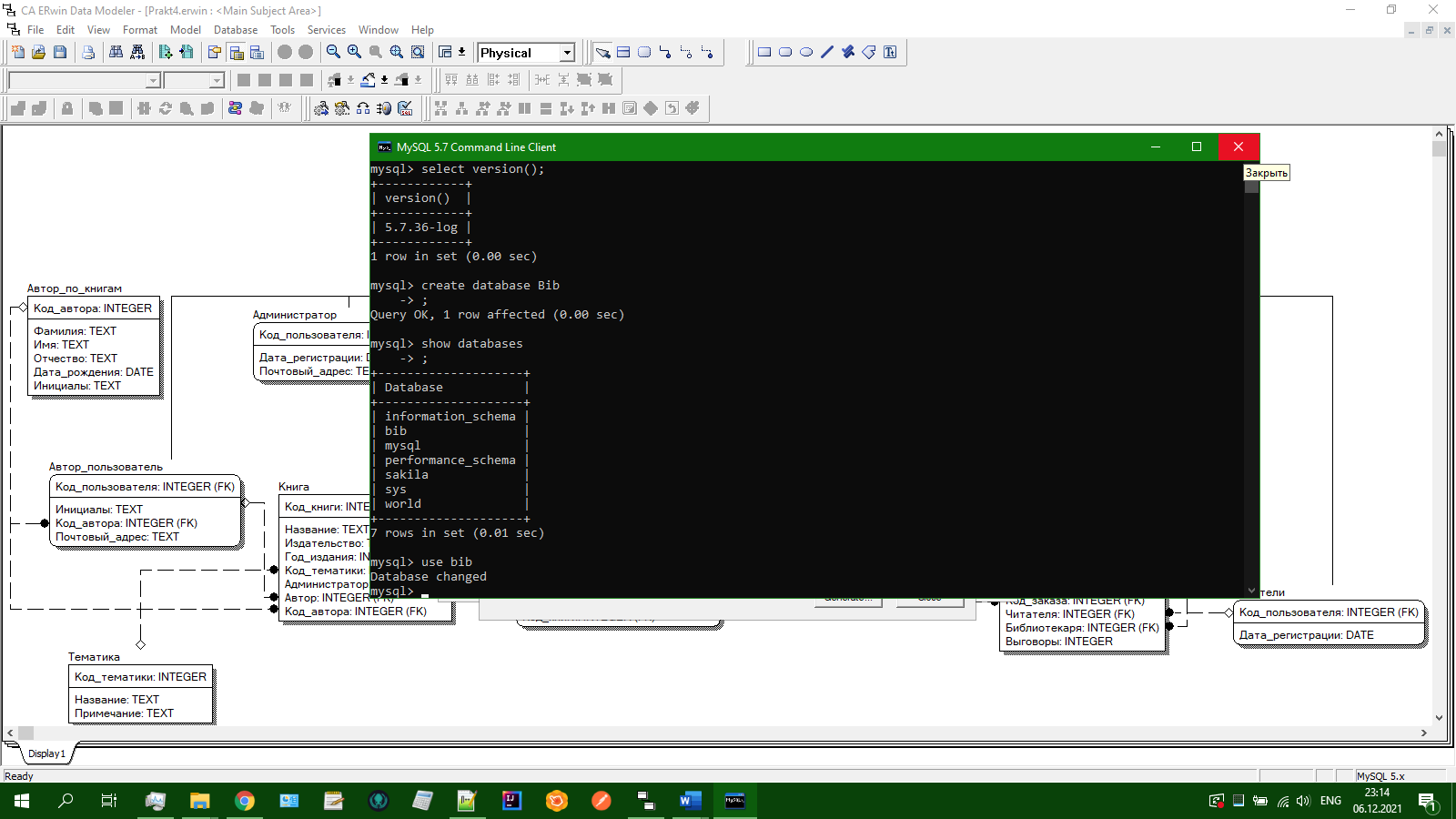
Рис. 3 Создание БД Blacksmith, таблицы clients и проверка

Рис. 4 Создание таблицы product\_types и проверка

Рис. 5 Создание таблицы material\_providers и проверка

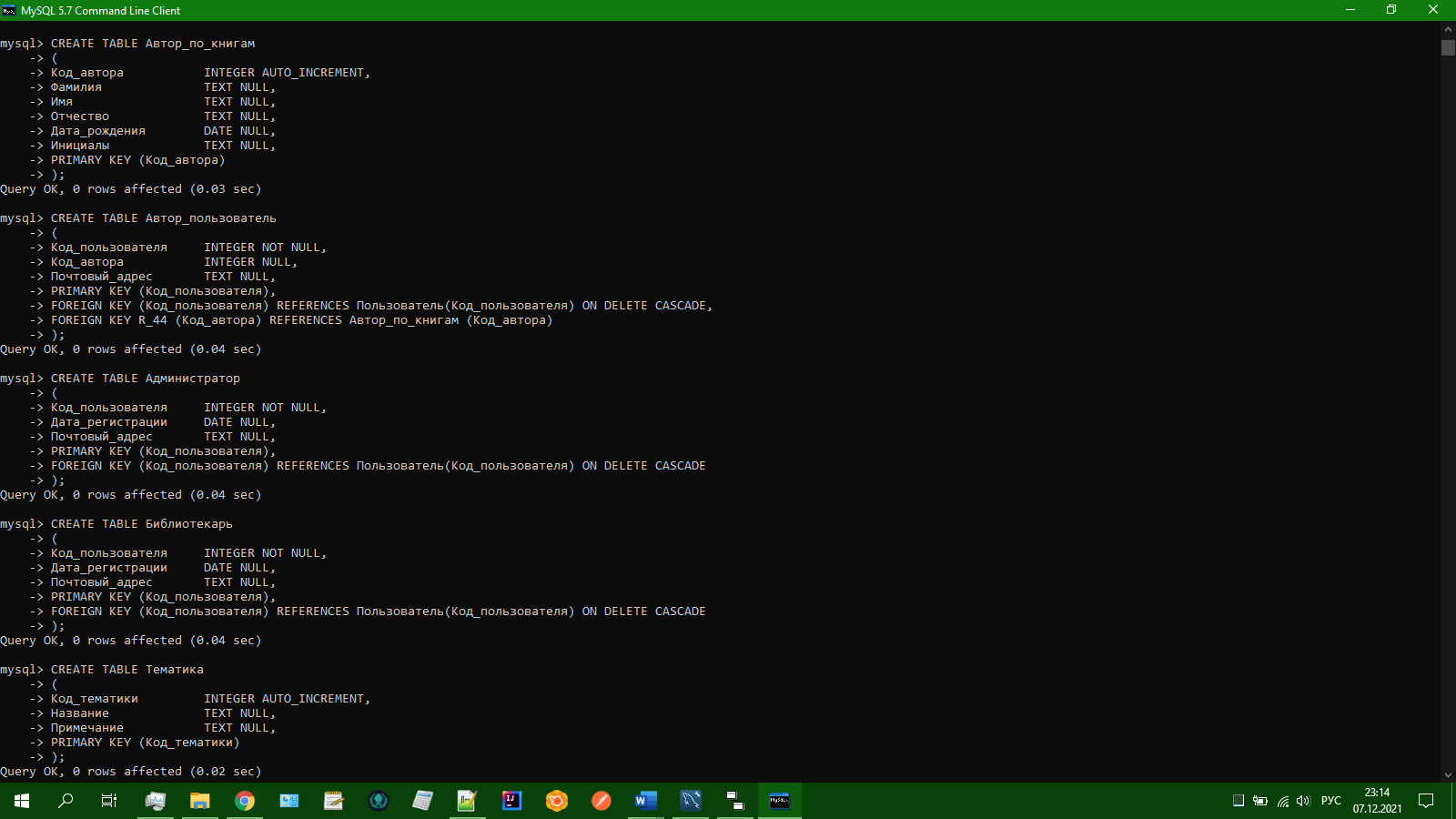


Рис. 6 Создание таблицы conditions и проверка



Рис. 7 Создание таблицы adress и проверка

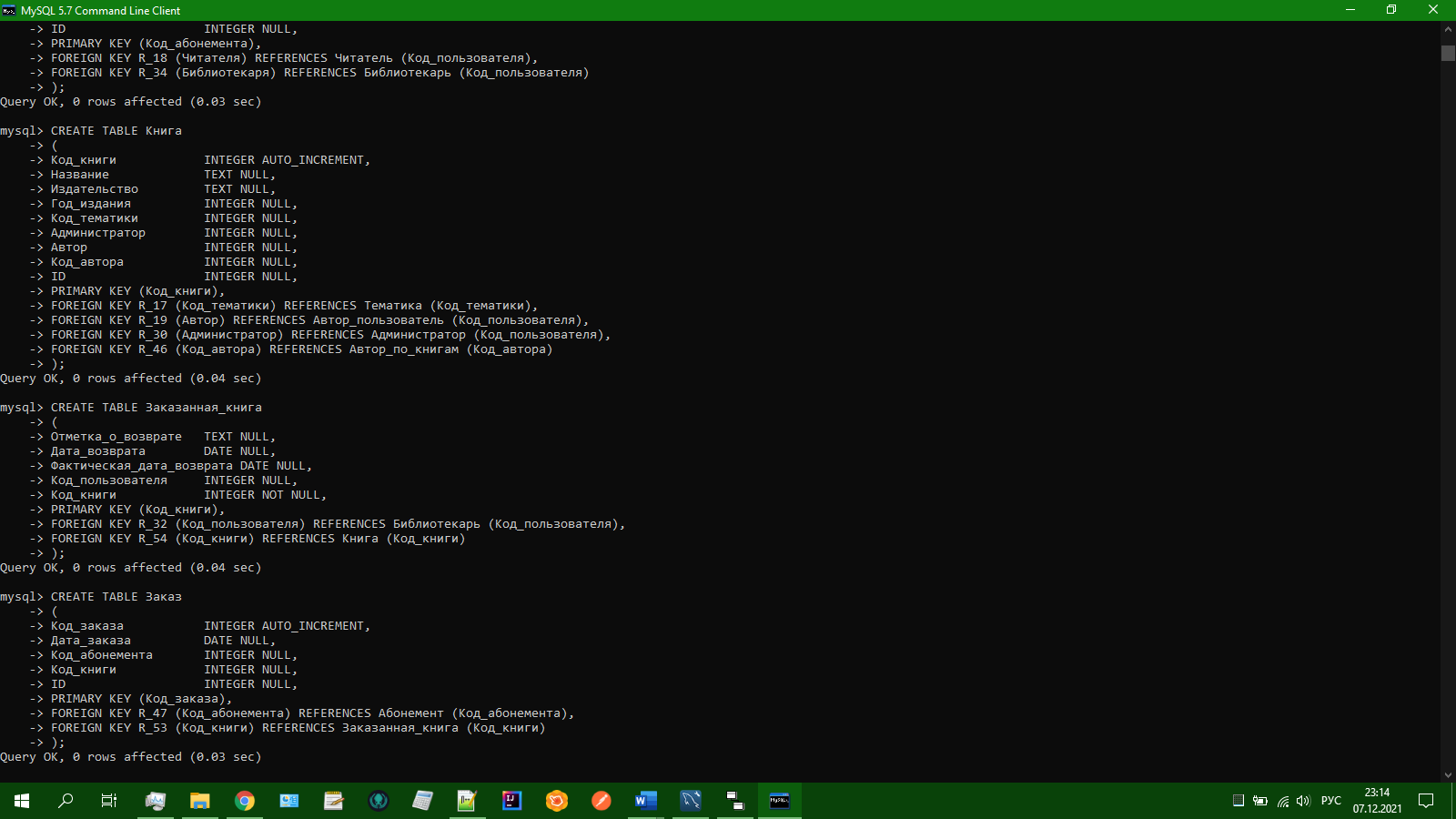


Рис. 8 Создание таблицы workshop\_budget

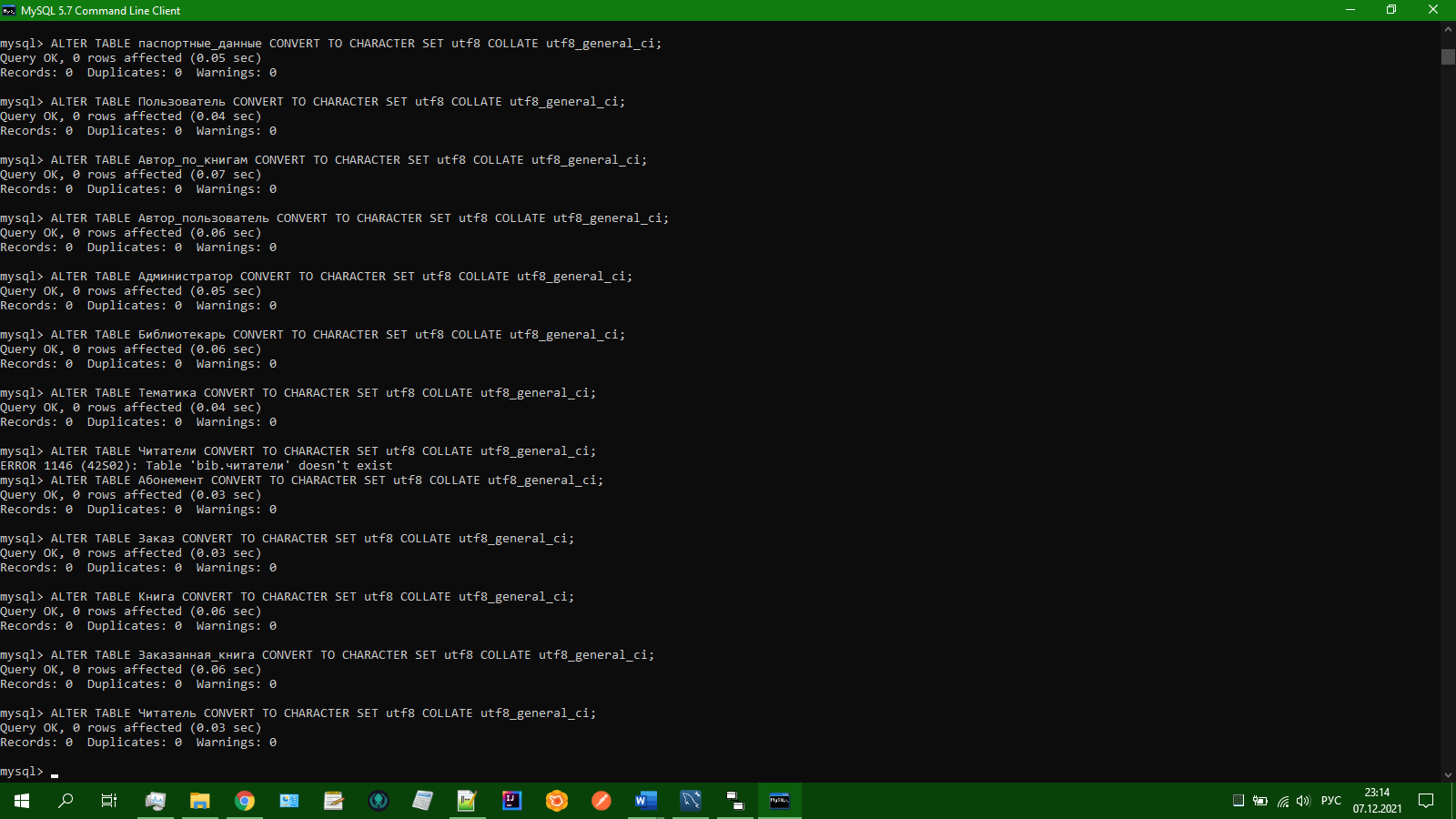


Рис. 9 Проверка правильности таблицы workshop\_budget

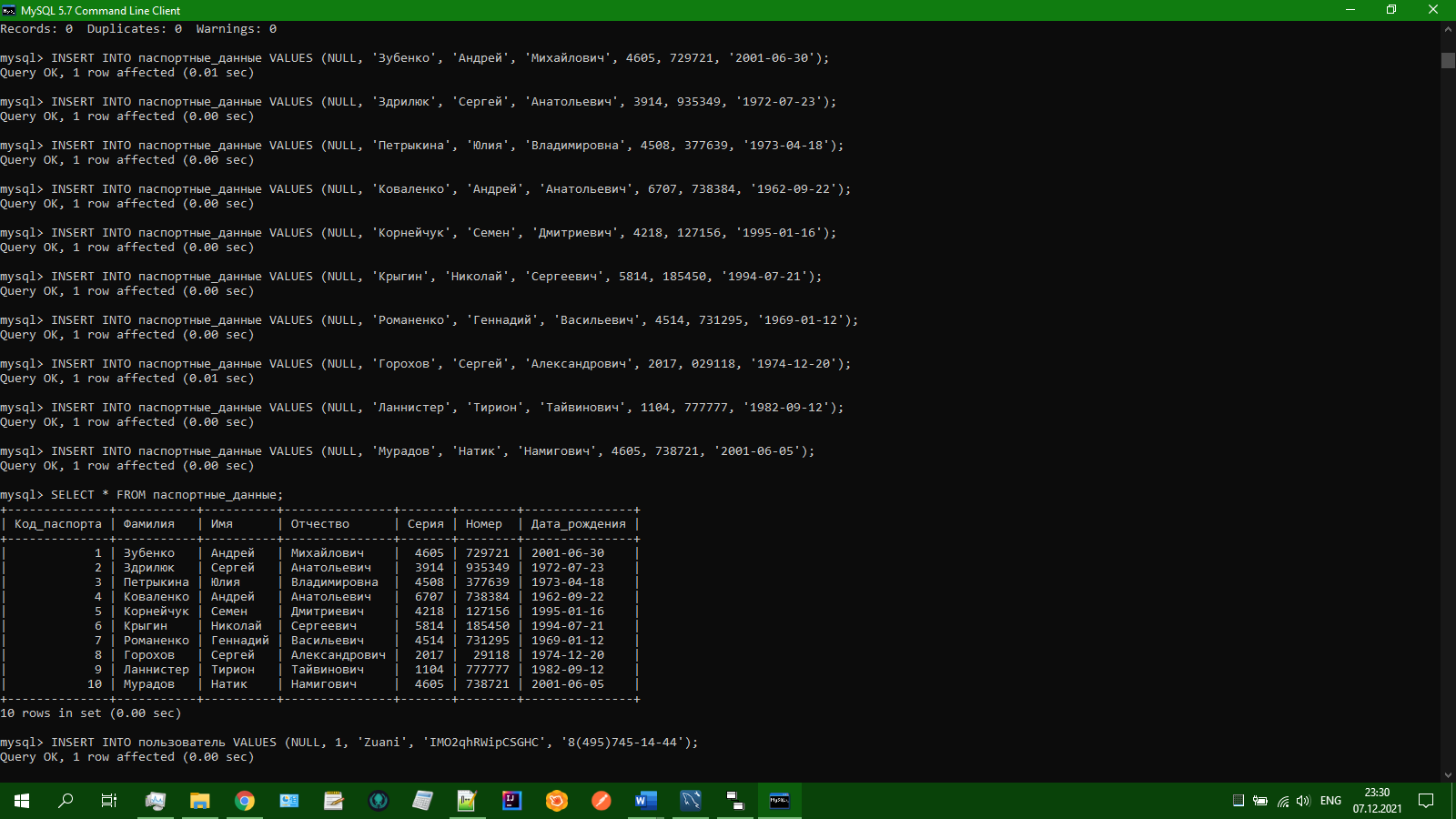


Рис. 10 Создание таблицы workshop\_employees и проверка

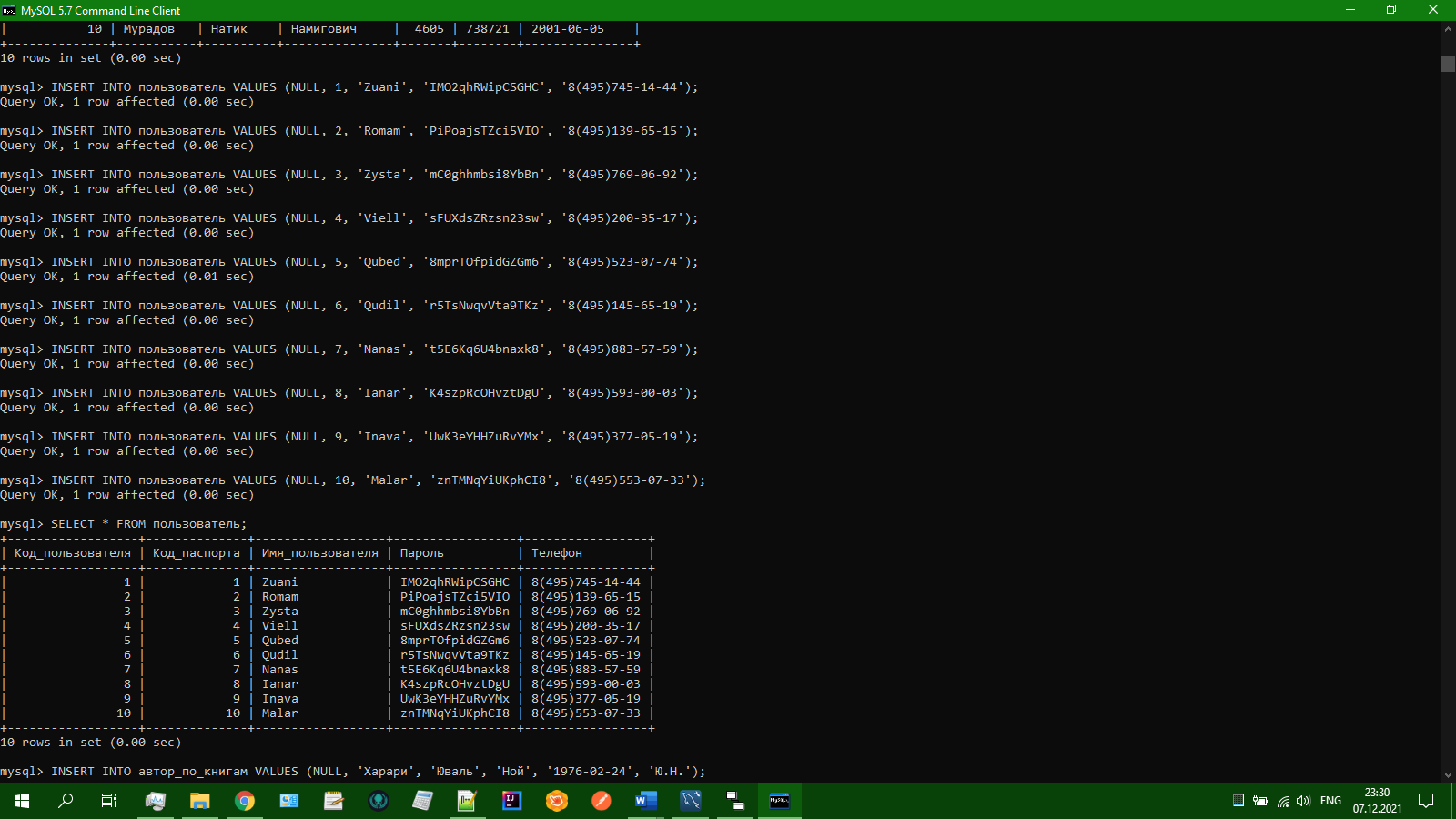


Рис. 11 Создание таблиц orders и work\_equipment

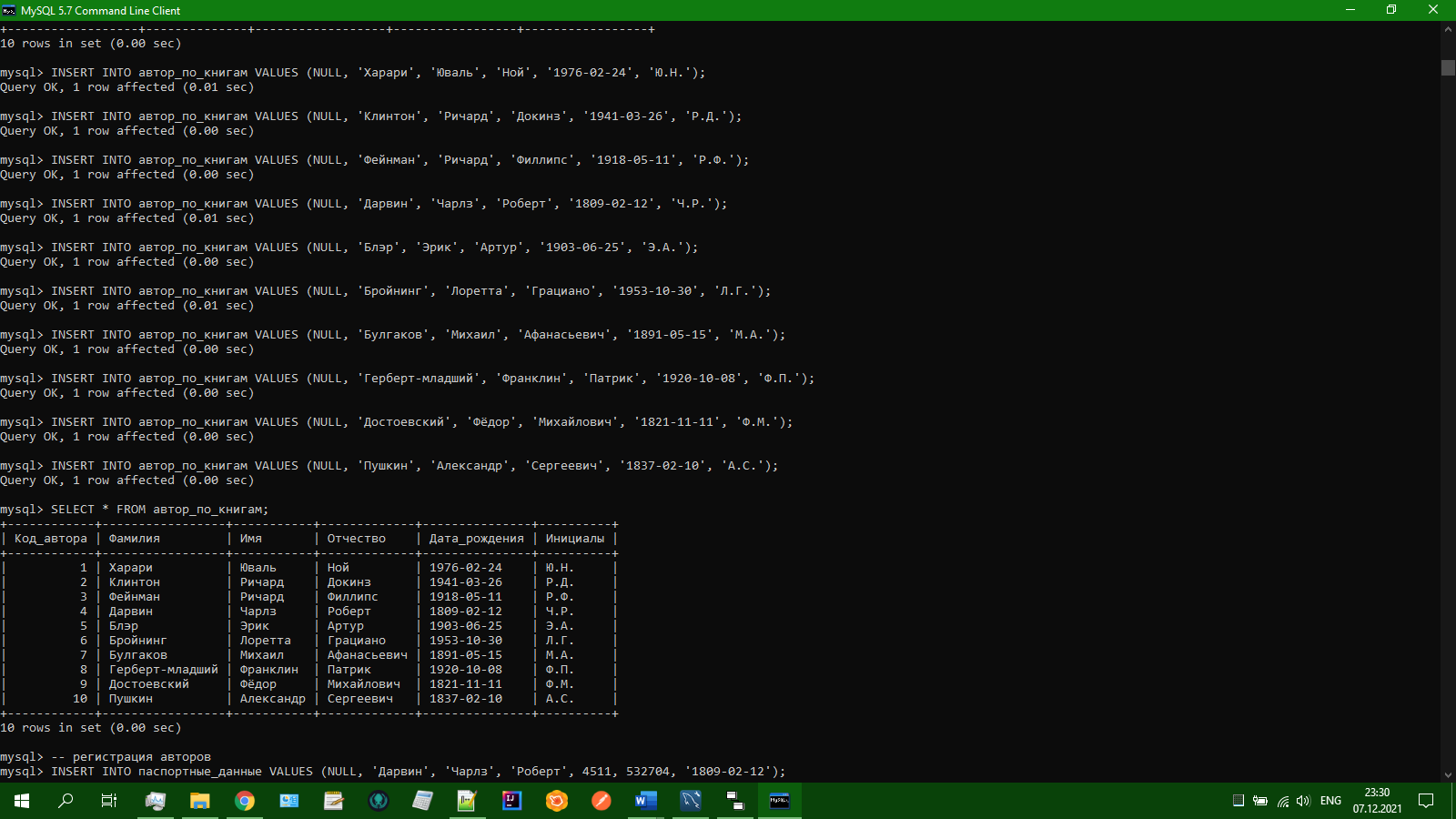


Рис. 12 Проверка правильности таблиц orders и work\_equipment

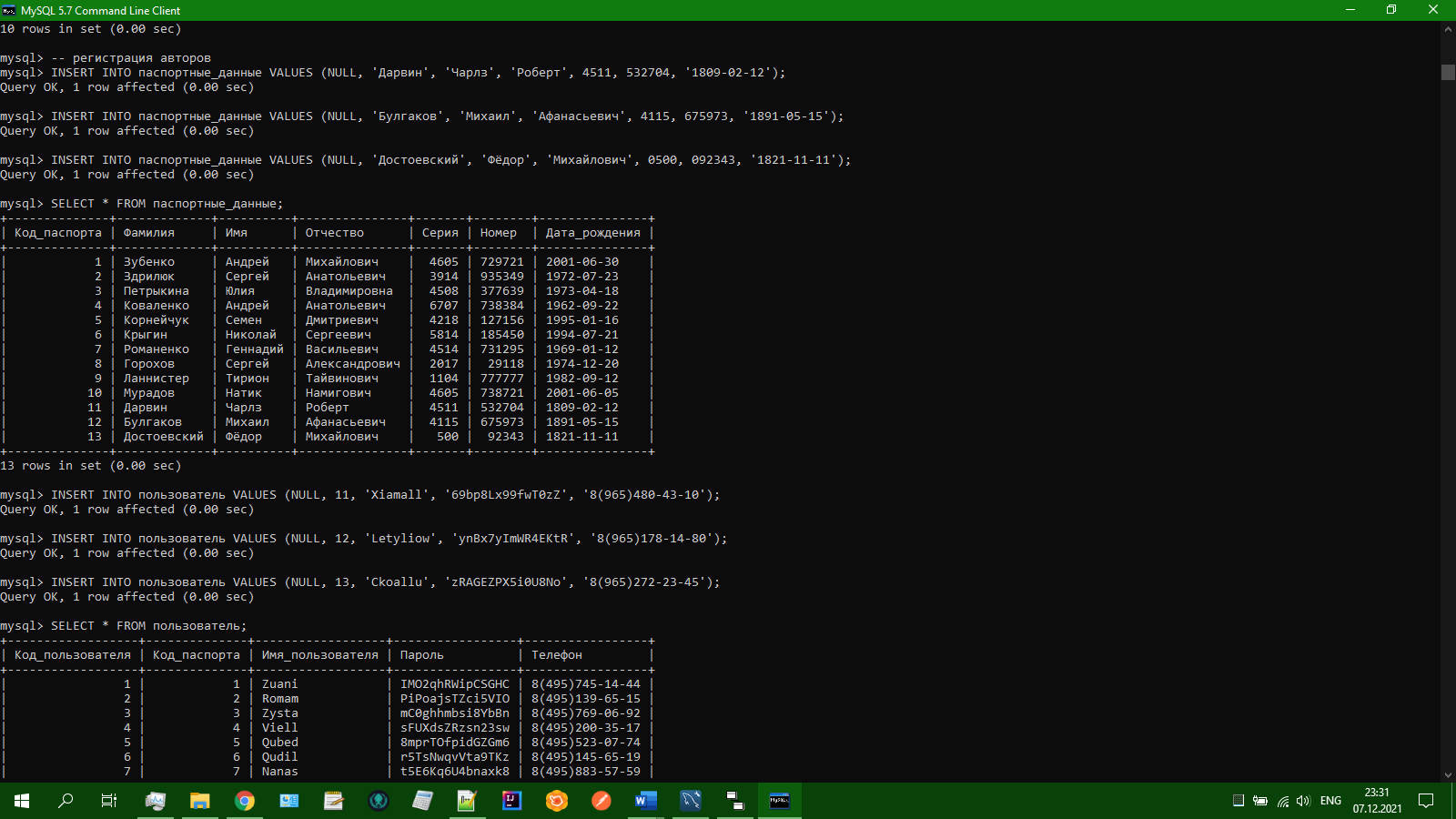


Рис. 13 Создание таблиц workshop\_rooms и workshop

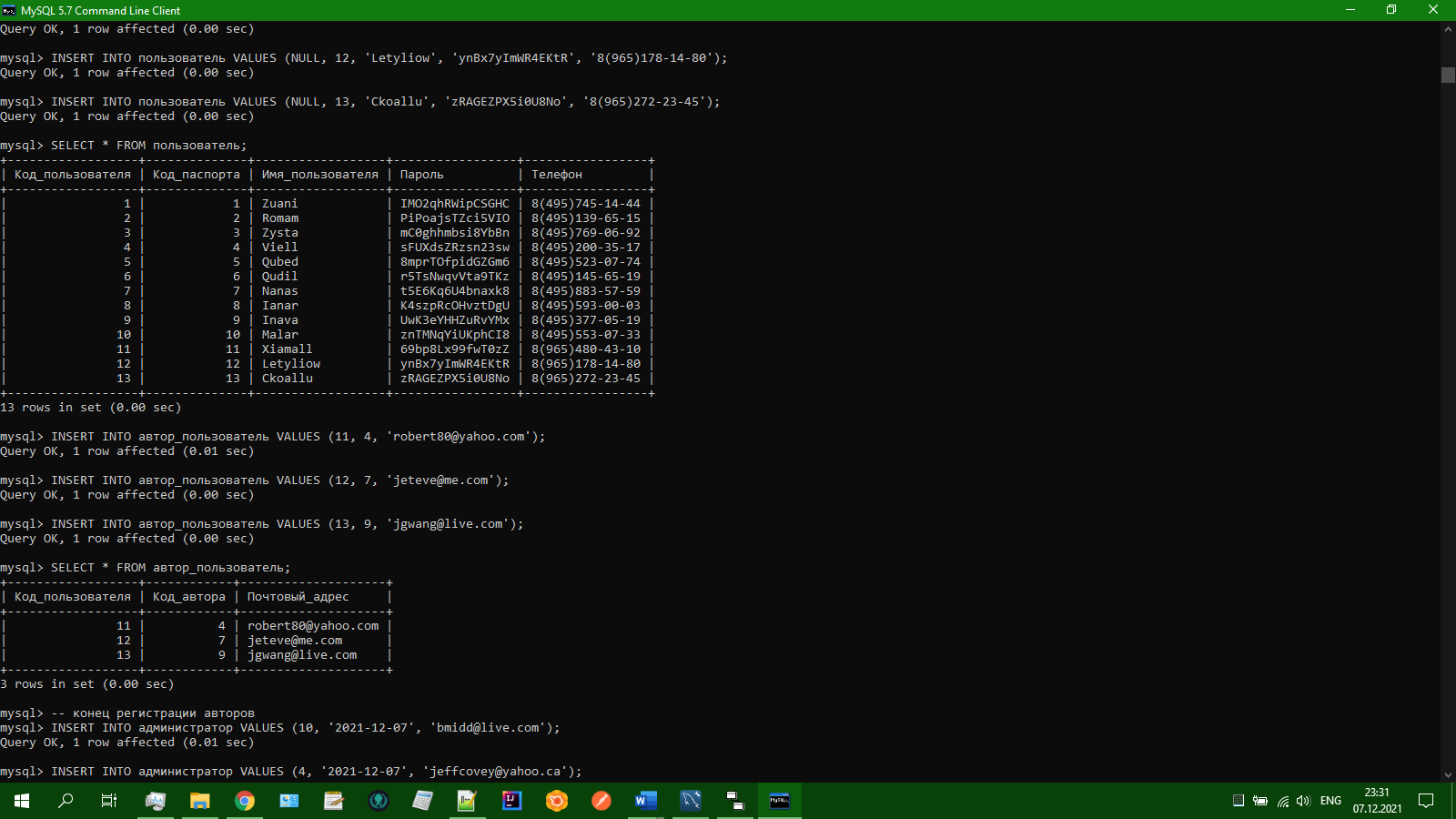


Рис. 14 Проверка правильности таблиц workshop\_rooms и workshop

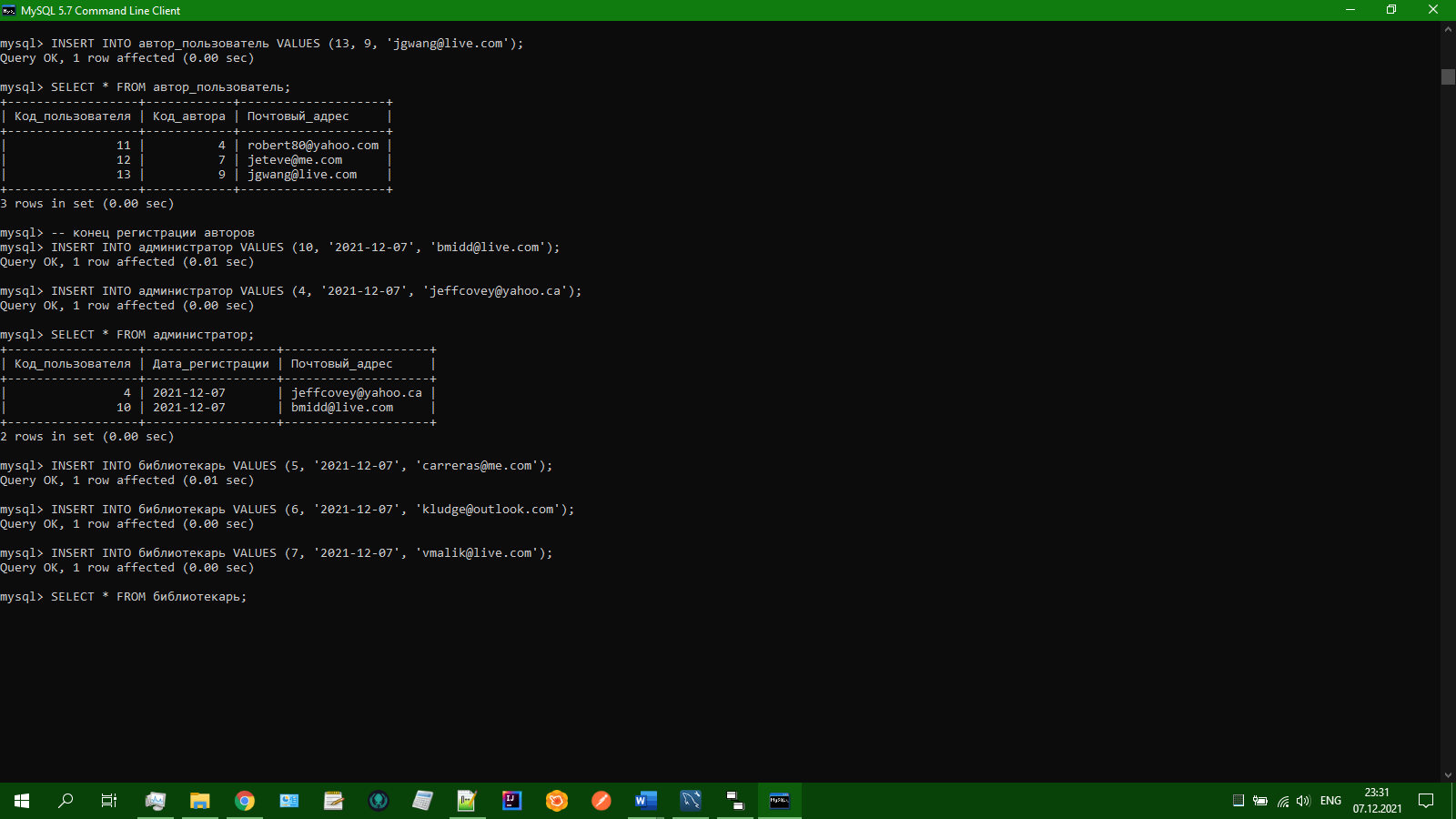


Рис. 15 Заполнение таблицы clients

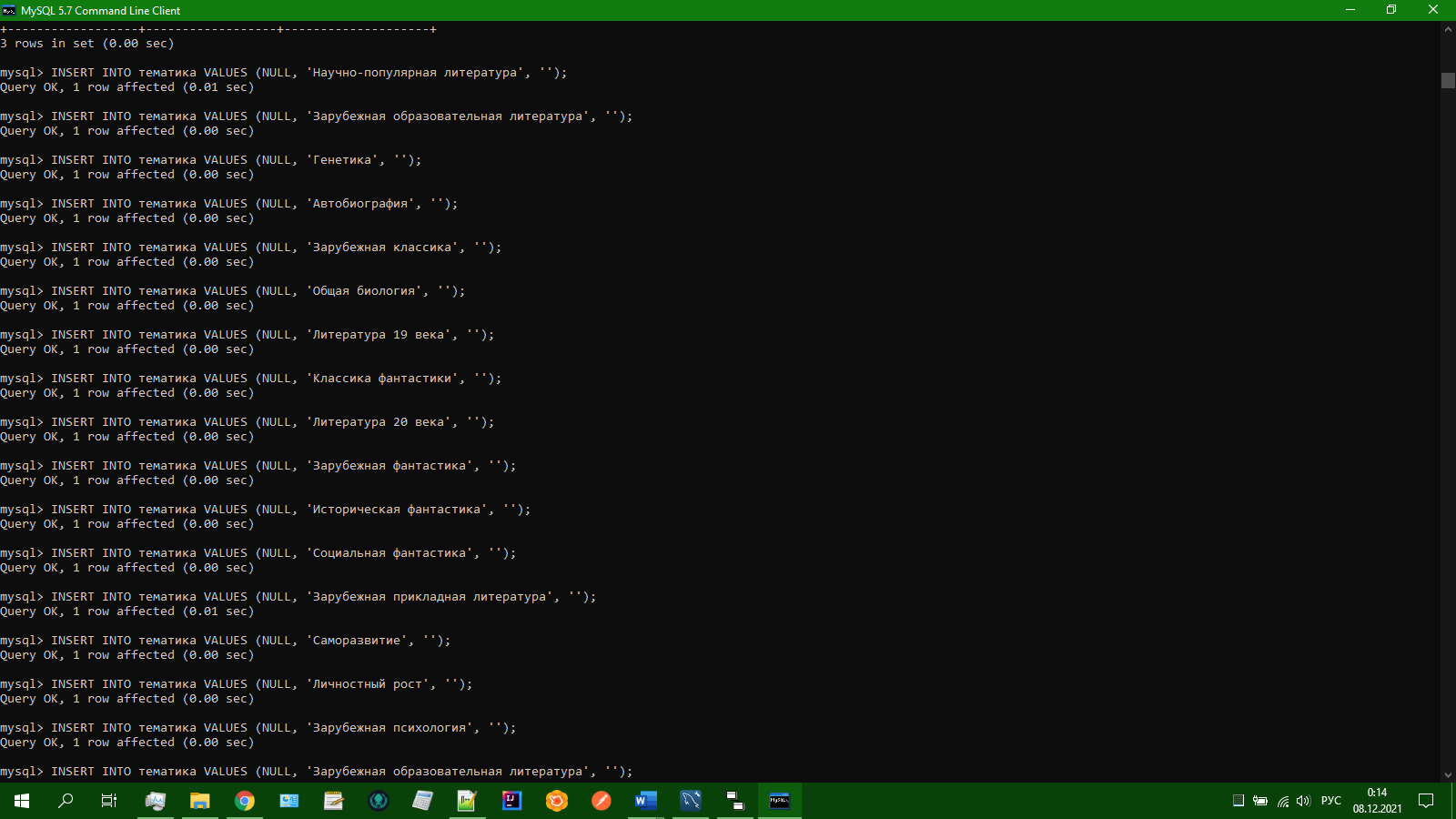


Рис. 16 Вывод элементов таблицы clients

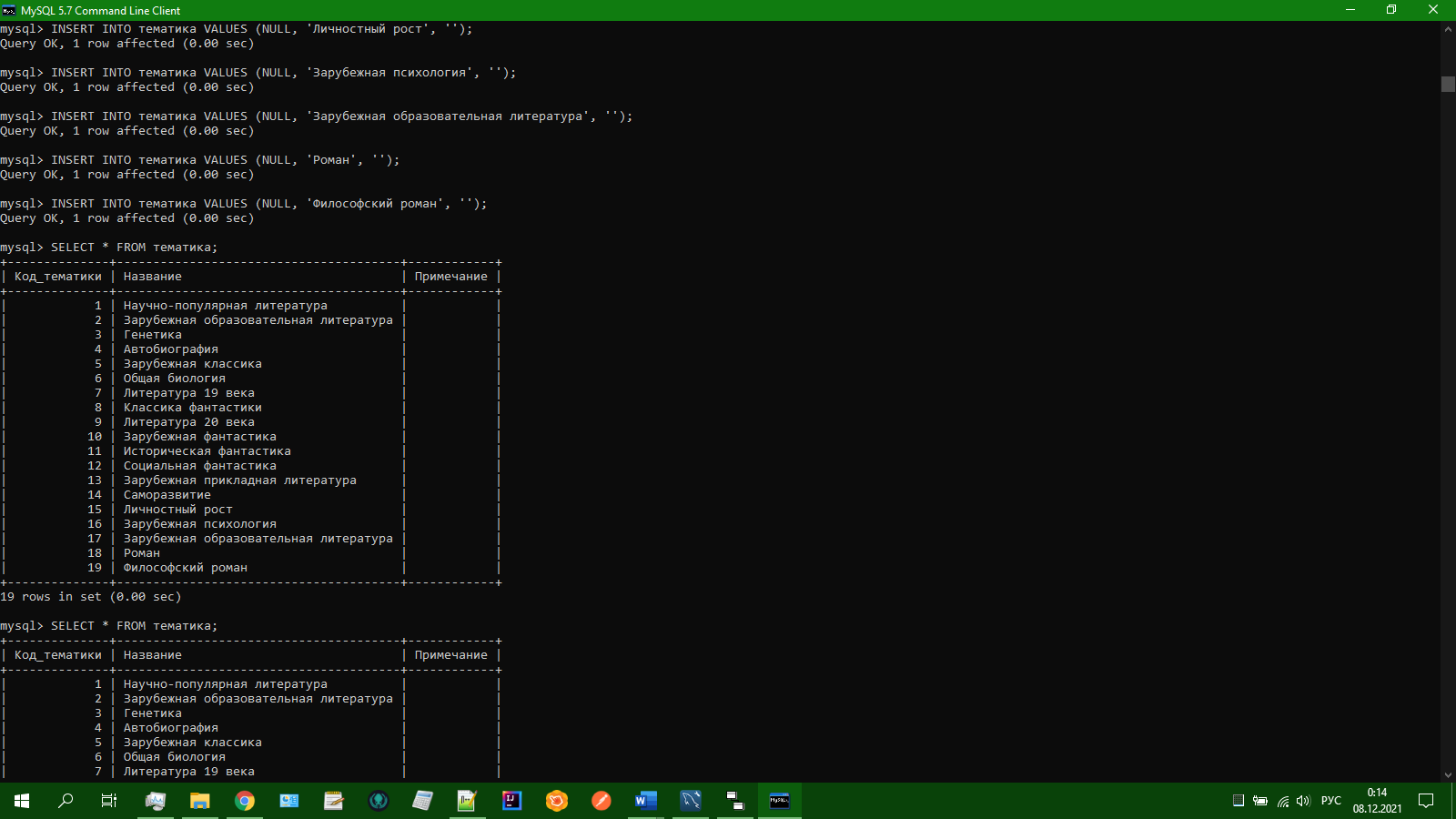


Рис. 17 Заполнение и вывод элементов таблицы adress

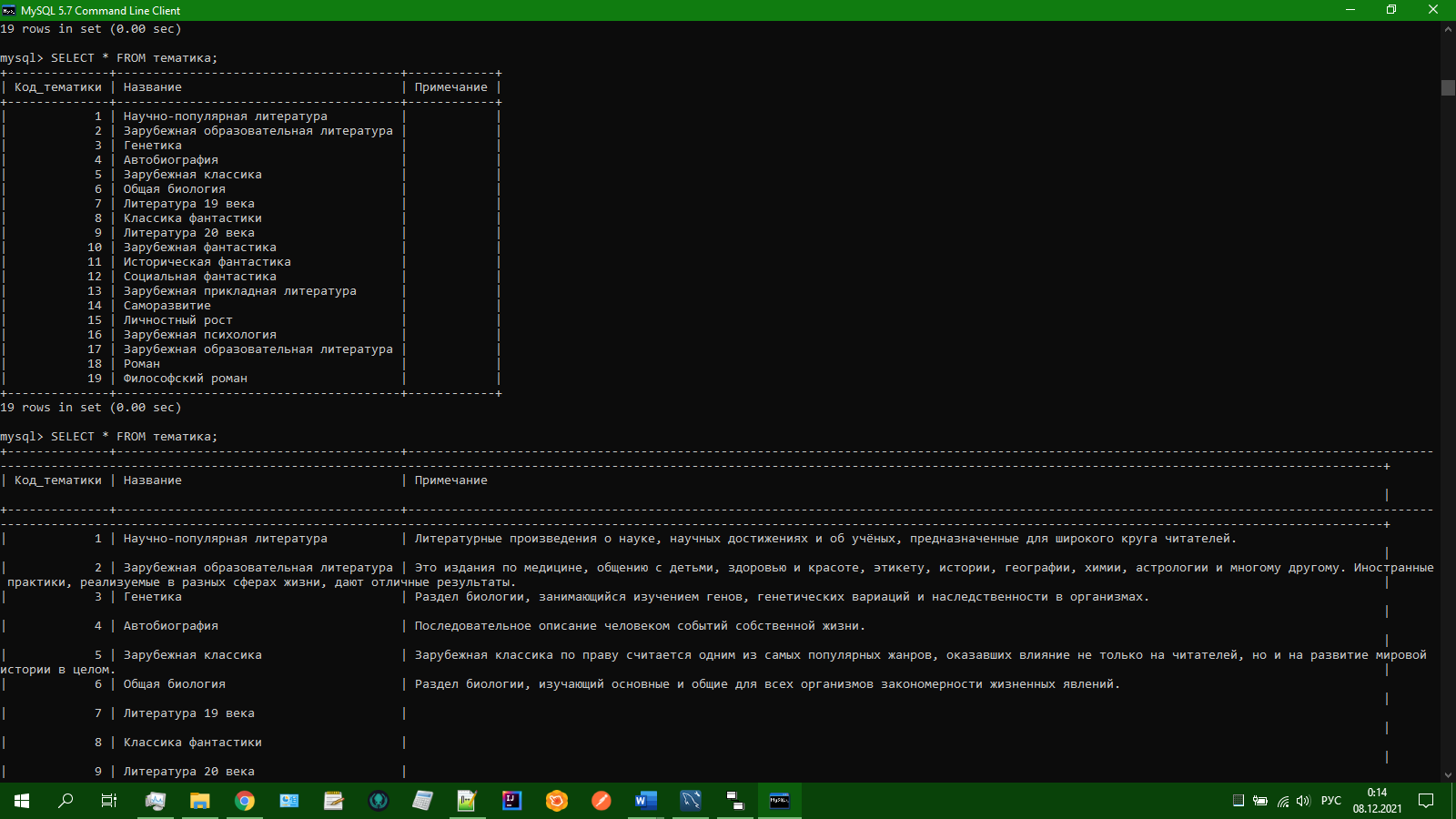


Рис. 18 Заполнение и вывод элементов таблицы conditions

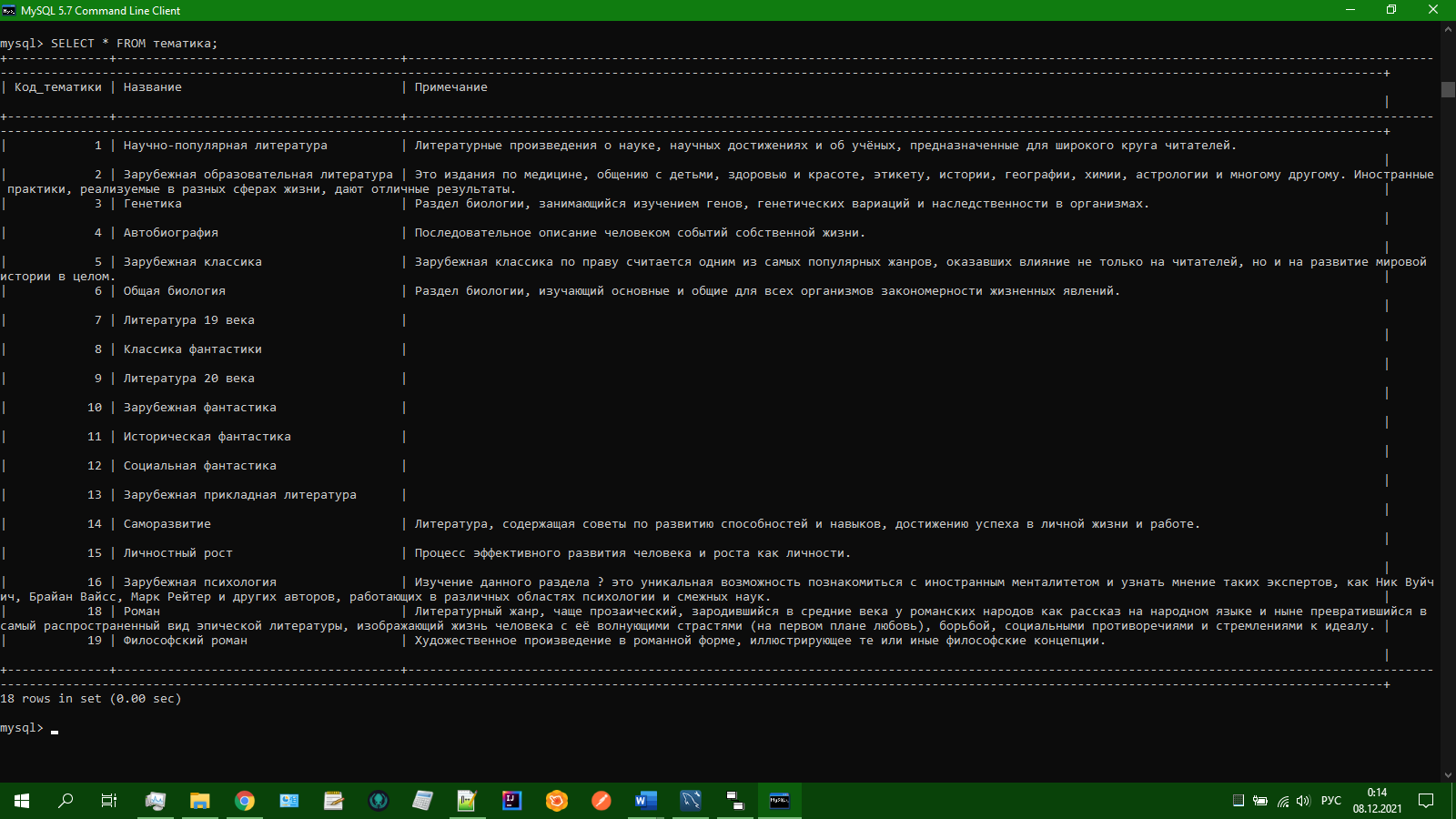


Рис. 19 Заполнение и вывод элементов таблицы material\_providers

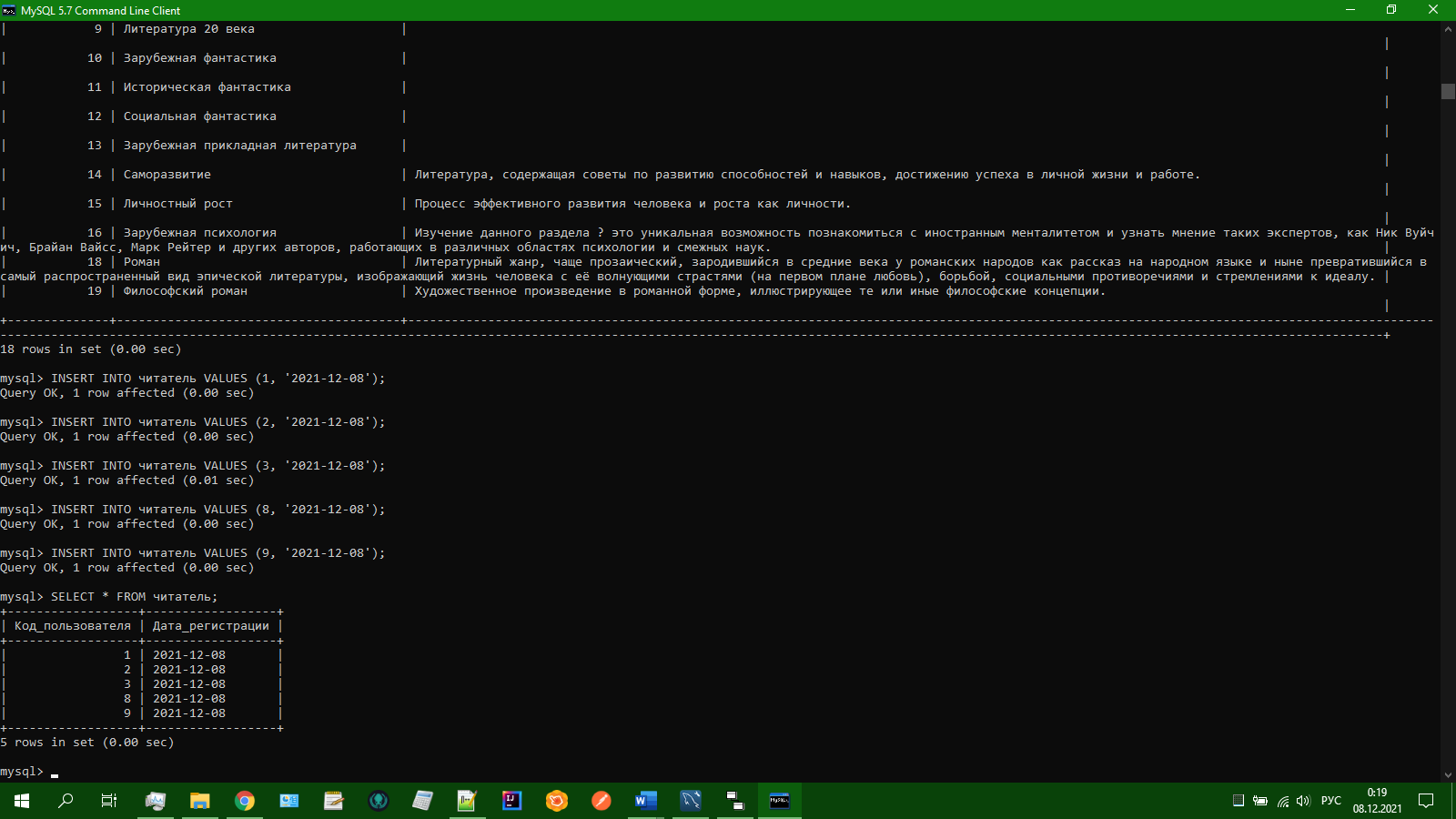


Рис. 20 Заполнение и вывод элементов таблицы product\_types

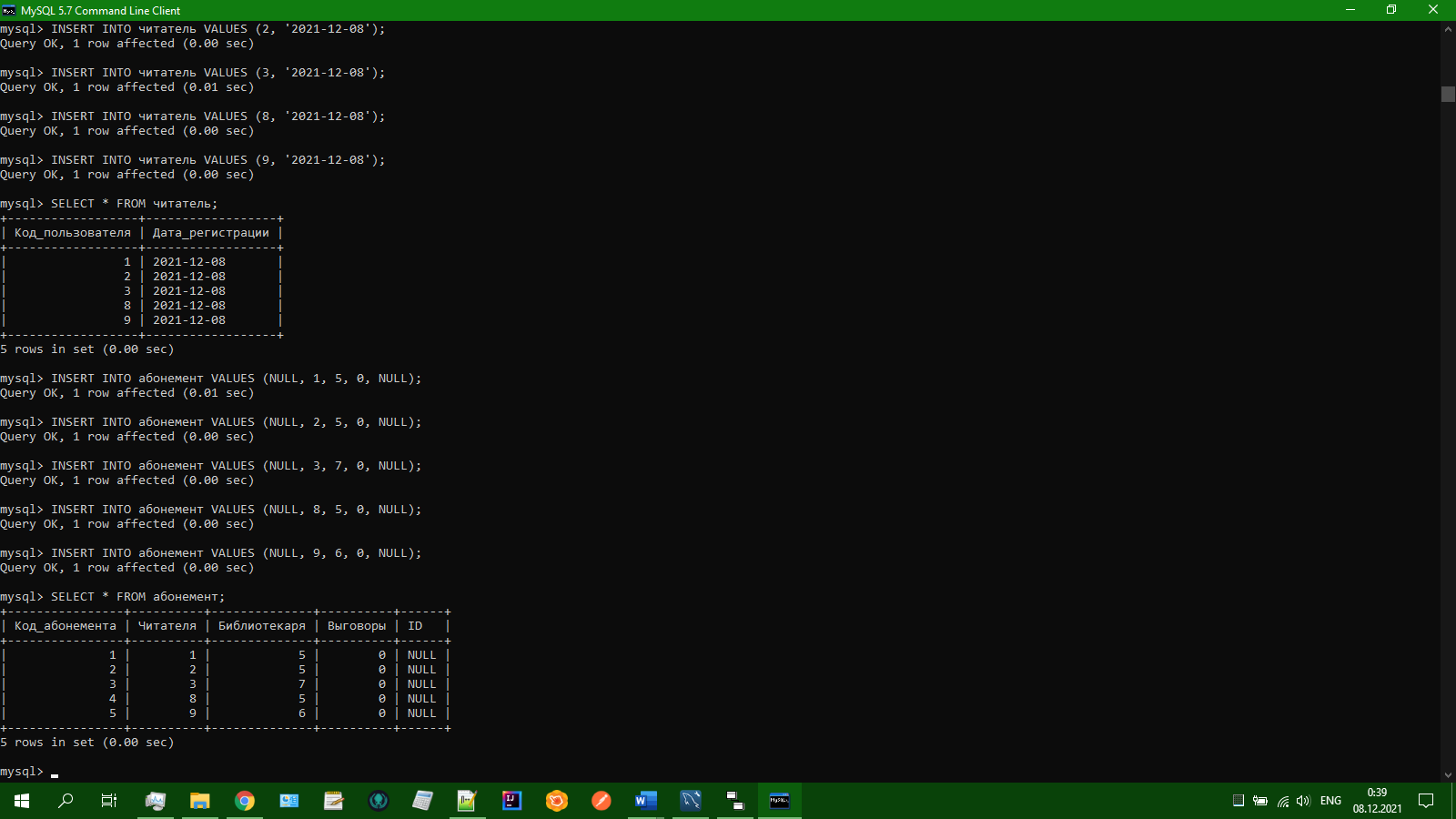


Рис. 21 Заполнение и вывод элементов таблицы workshop\_budget

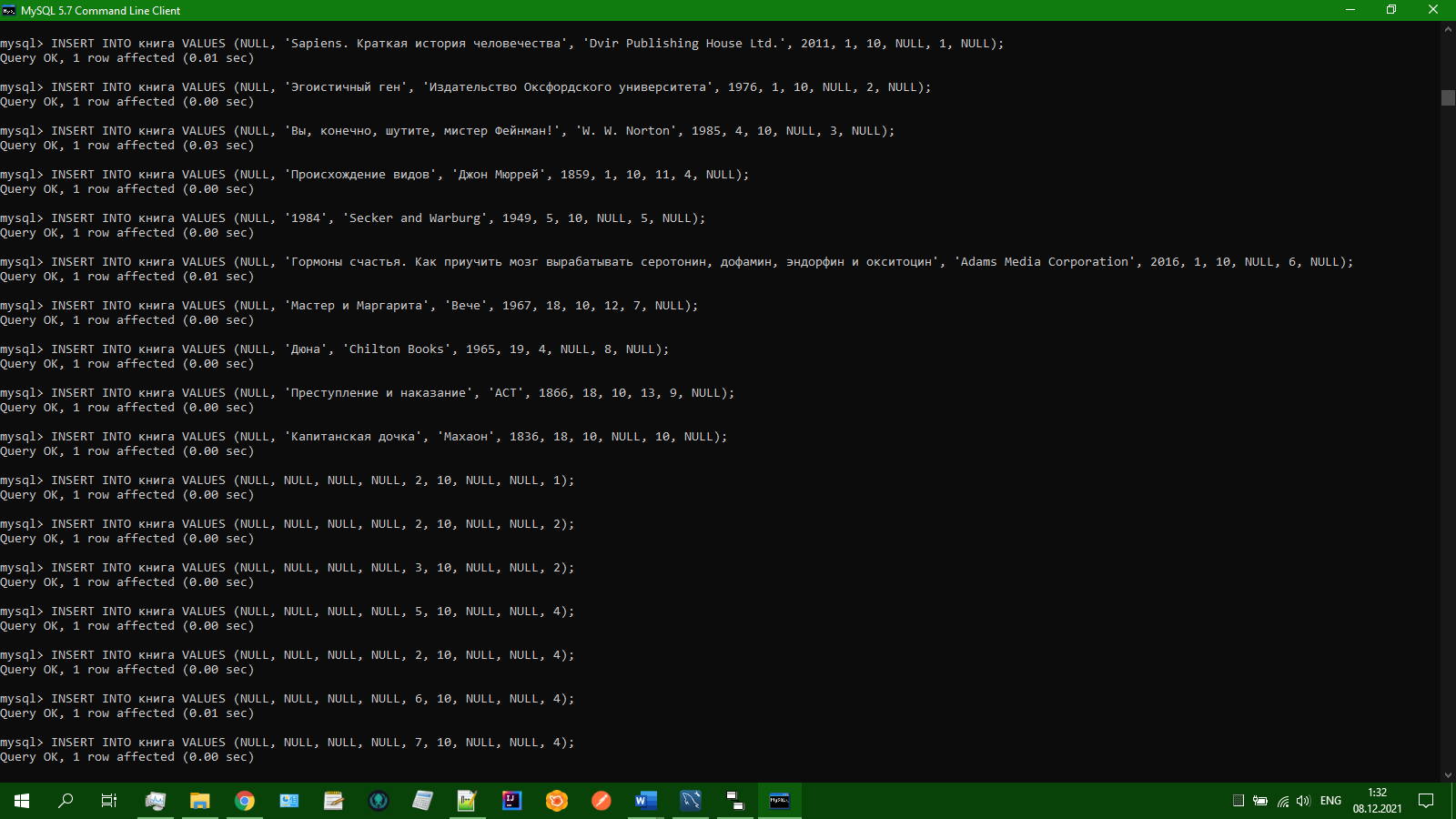


Рис. 22 Заполнение и вывод элементов таблицы orders

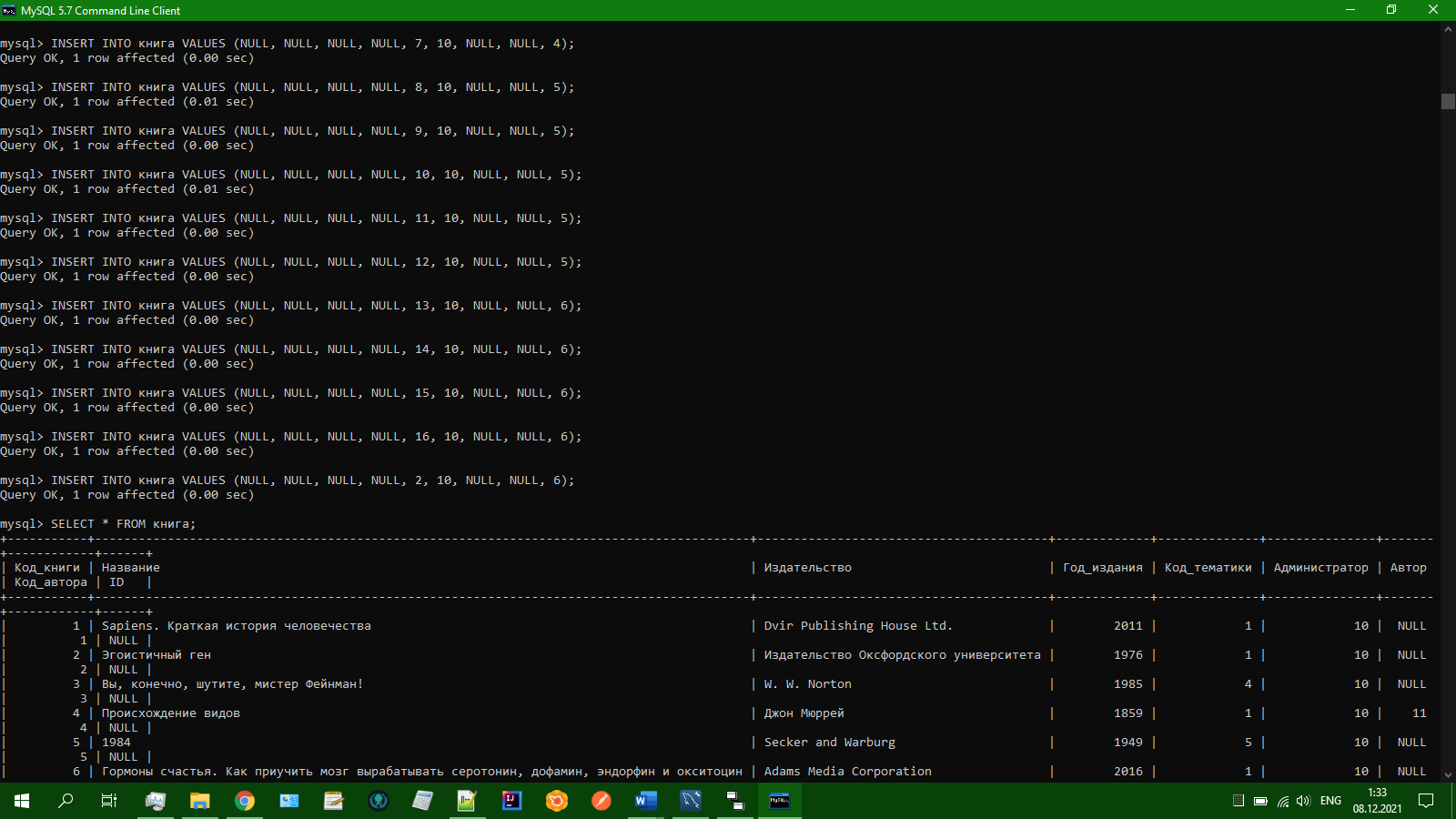


Рис. 23 Заполнение и вывод элементов таблицы work\_equipment

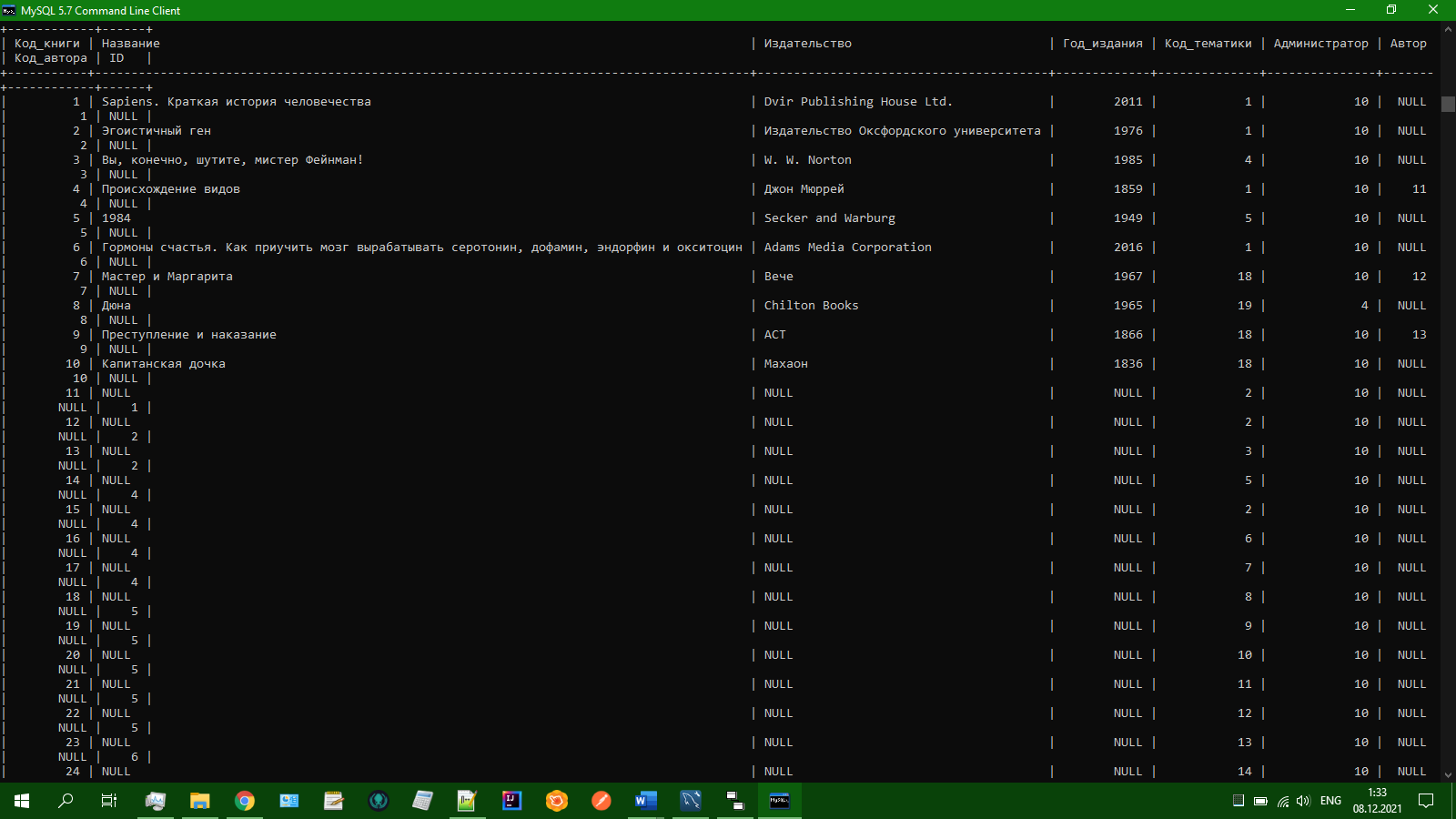


Рис. 24 Заполнение и вывод элементов таблицы workshop\_employees

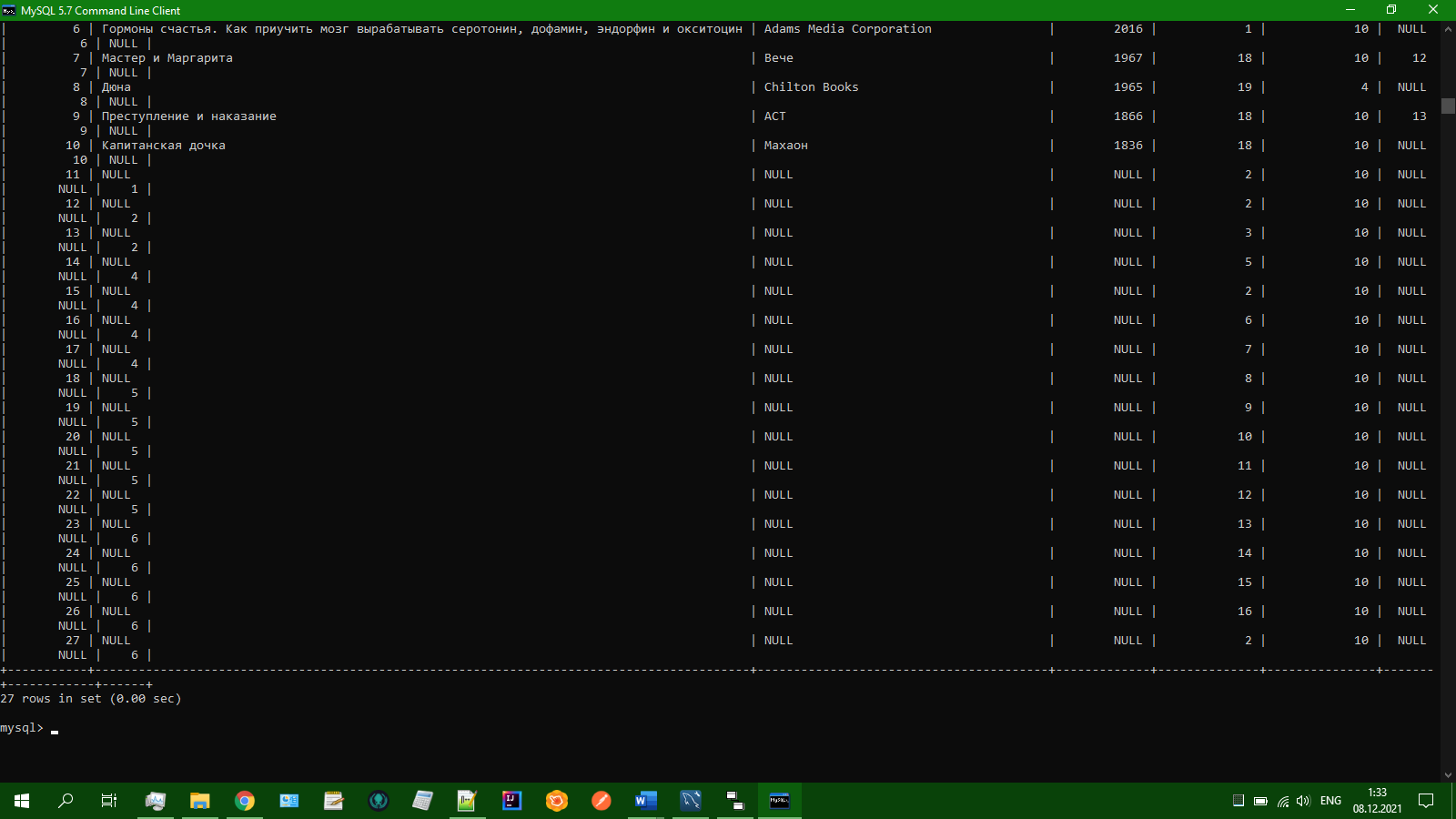


Рис. 25 Заполнение и вывод элементов таблицы workshop\_rooms

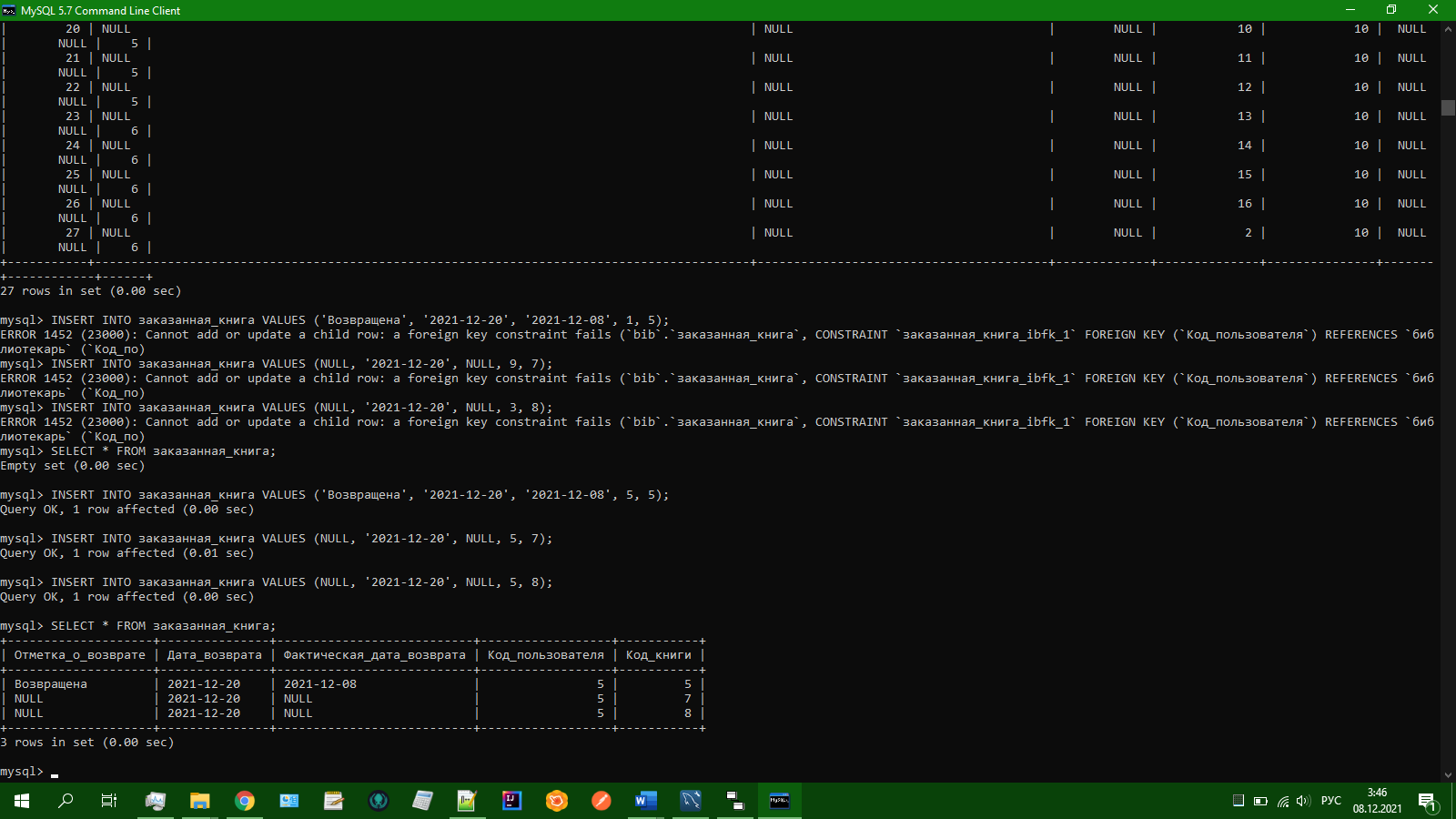


Рис. 26 Заполнение и вывод элементов таблицы workshop

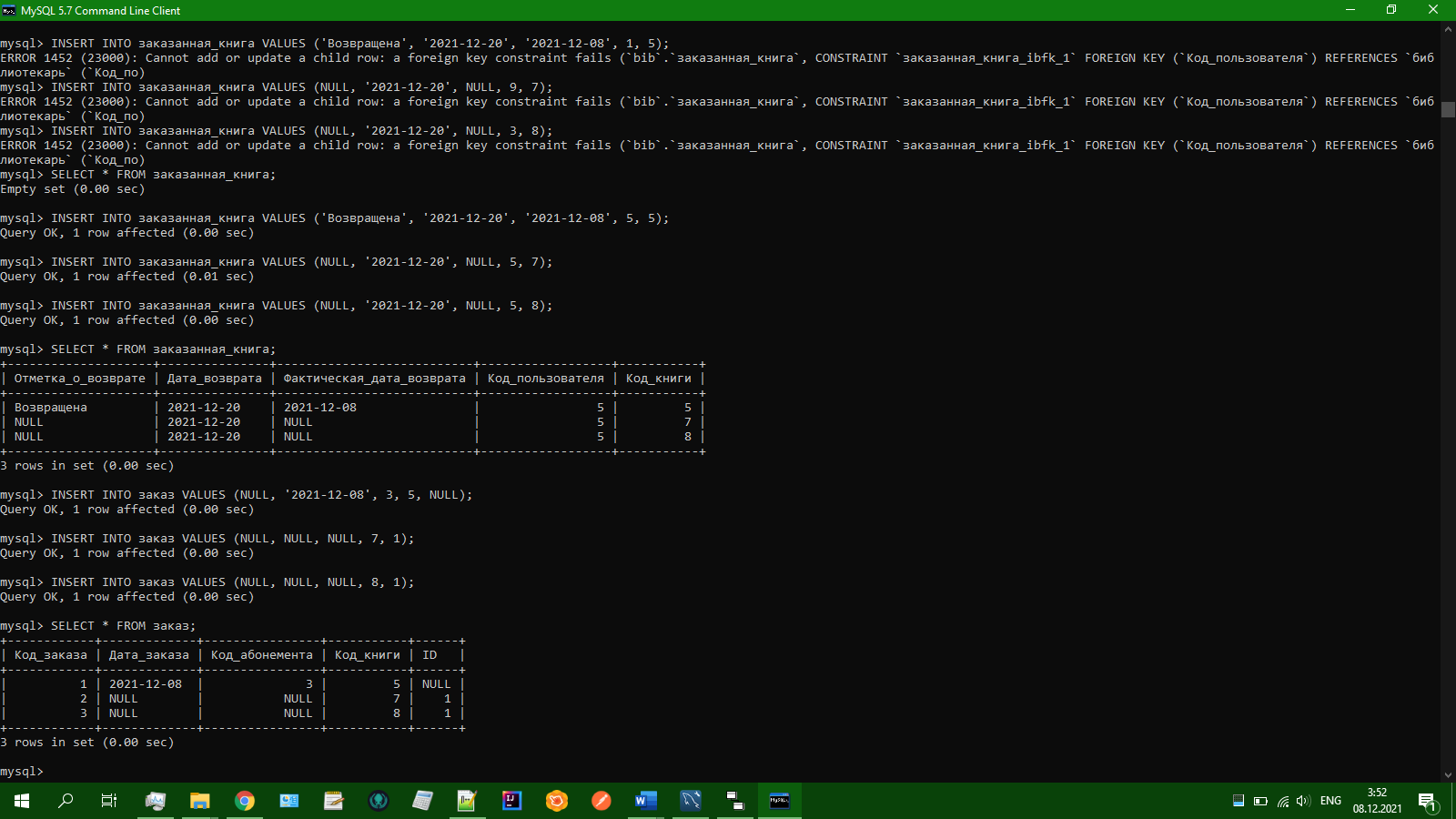
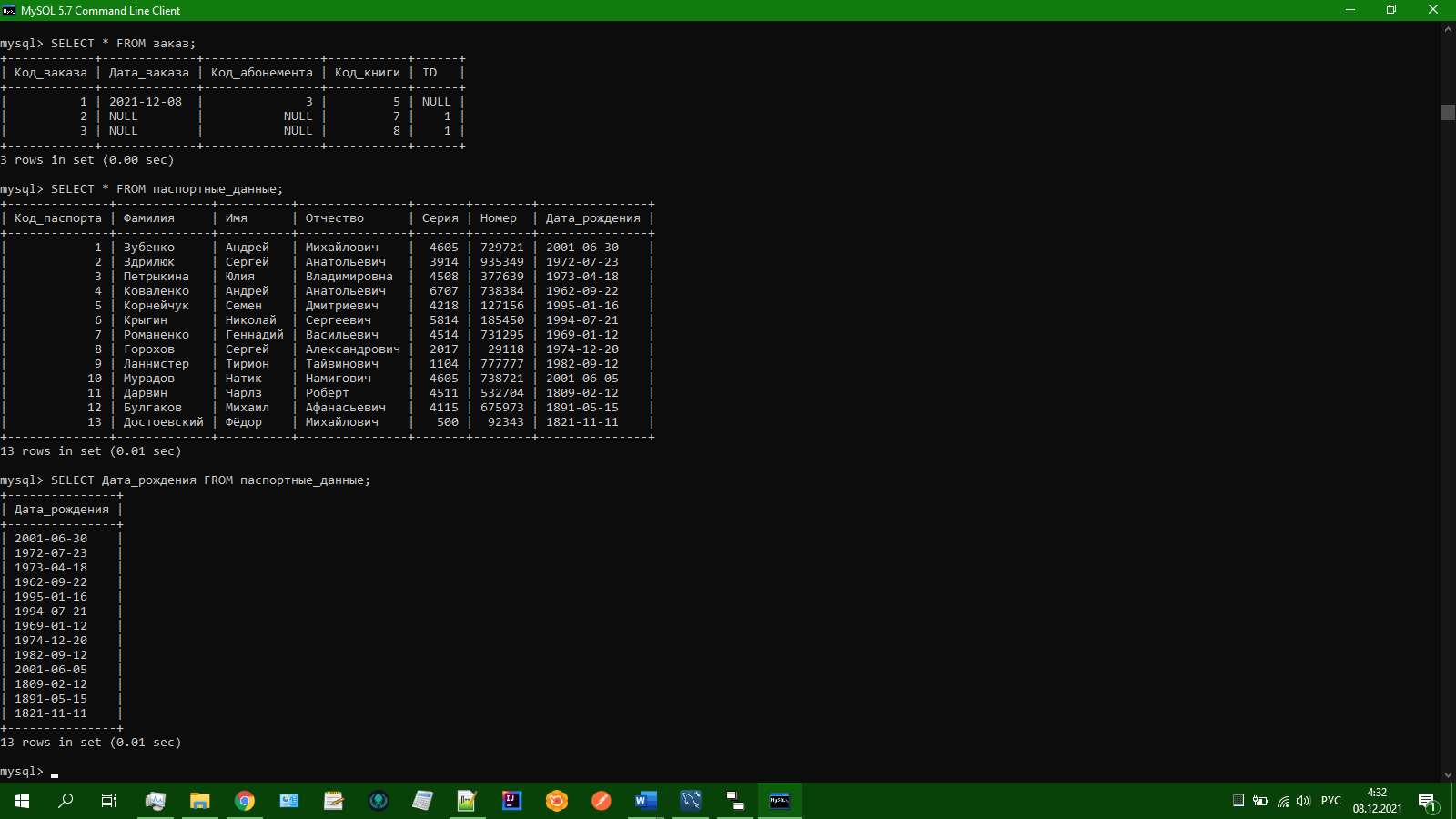


Рис. 27 Заполнение и вывод элементов таблицы workshop

# Практическое задание 6

**Работа с запросами в базе данных bib**

  
Рис. 28 ‒ Объединение с использованием UNION

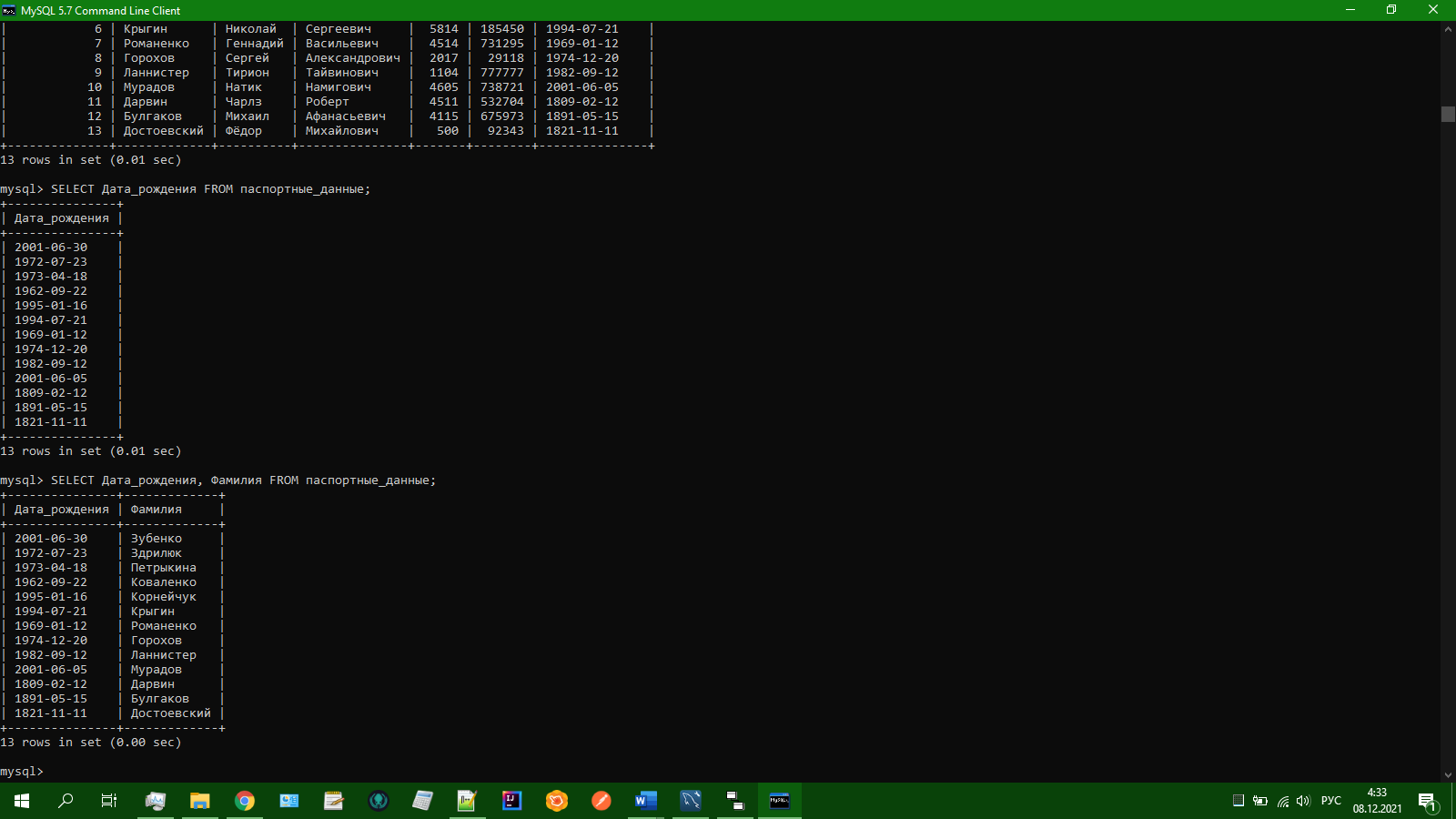


Рис. 29 ‒ Выборка с использованием LEFT JOIN часть 1

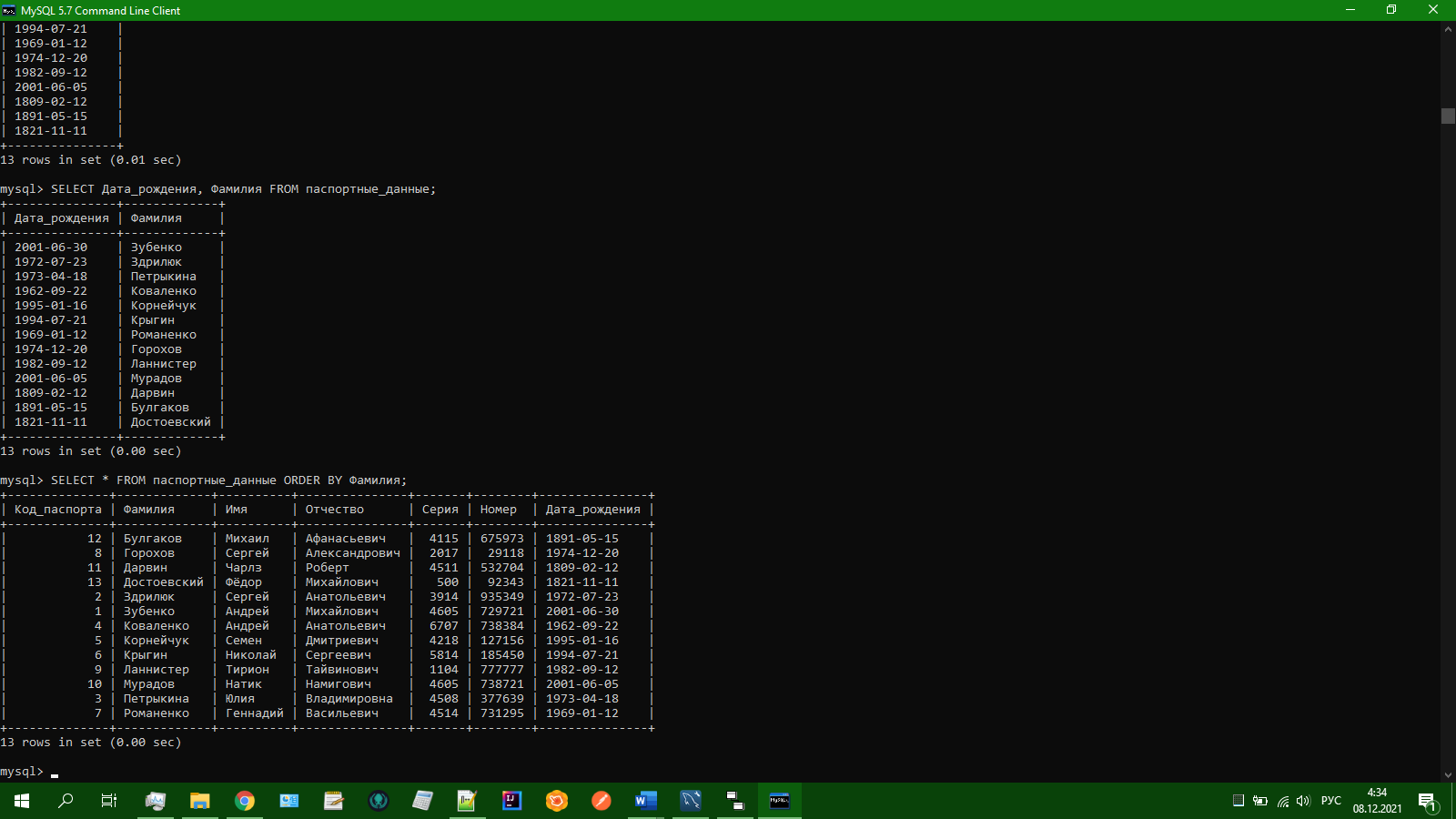
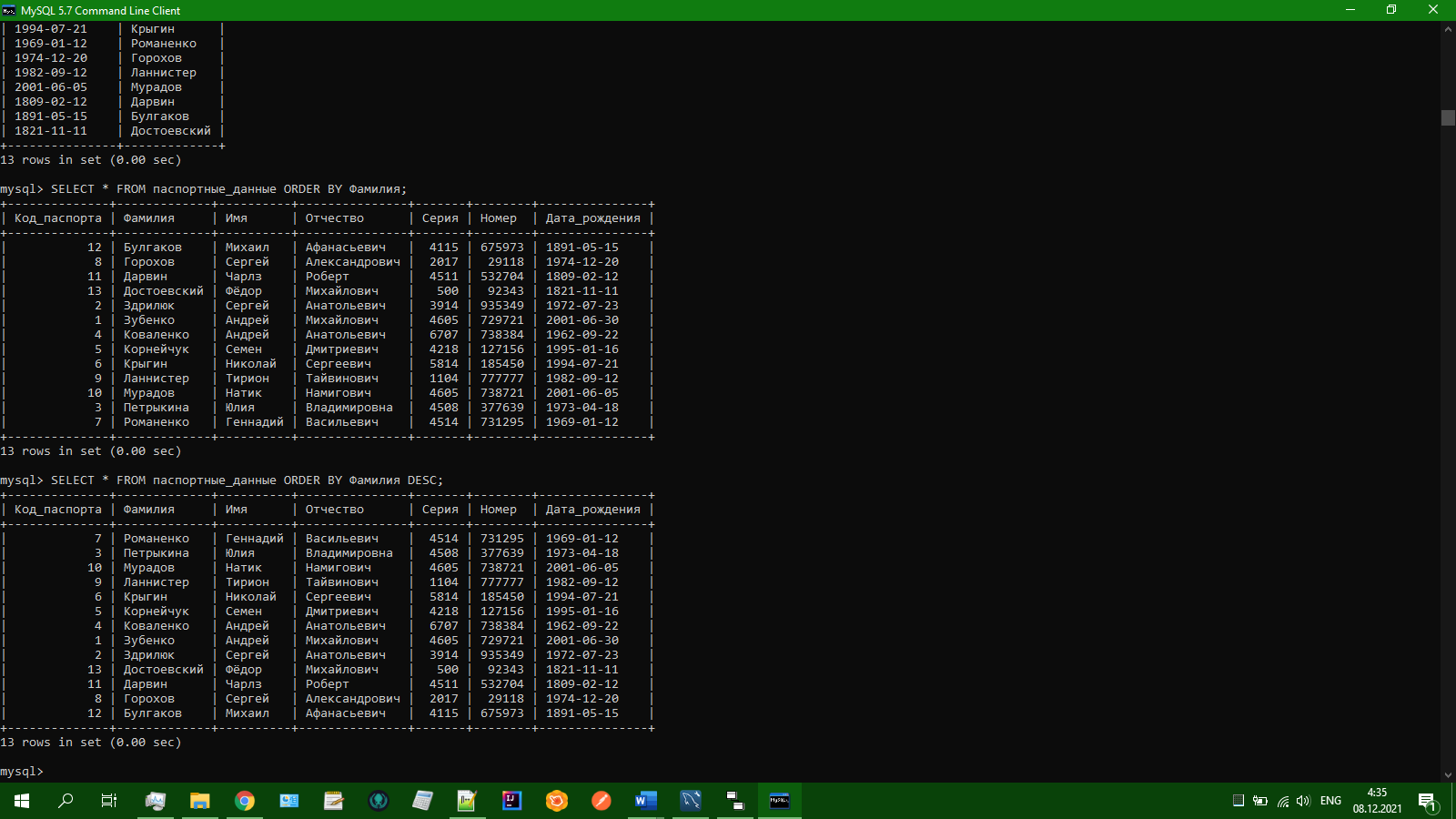
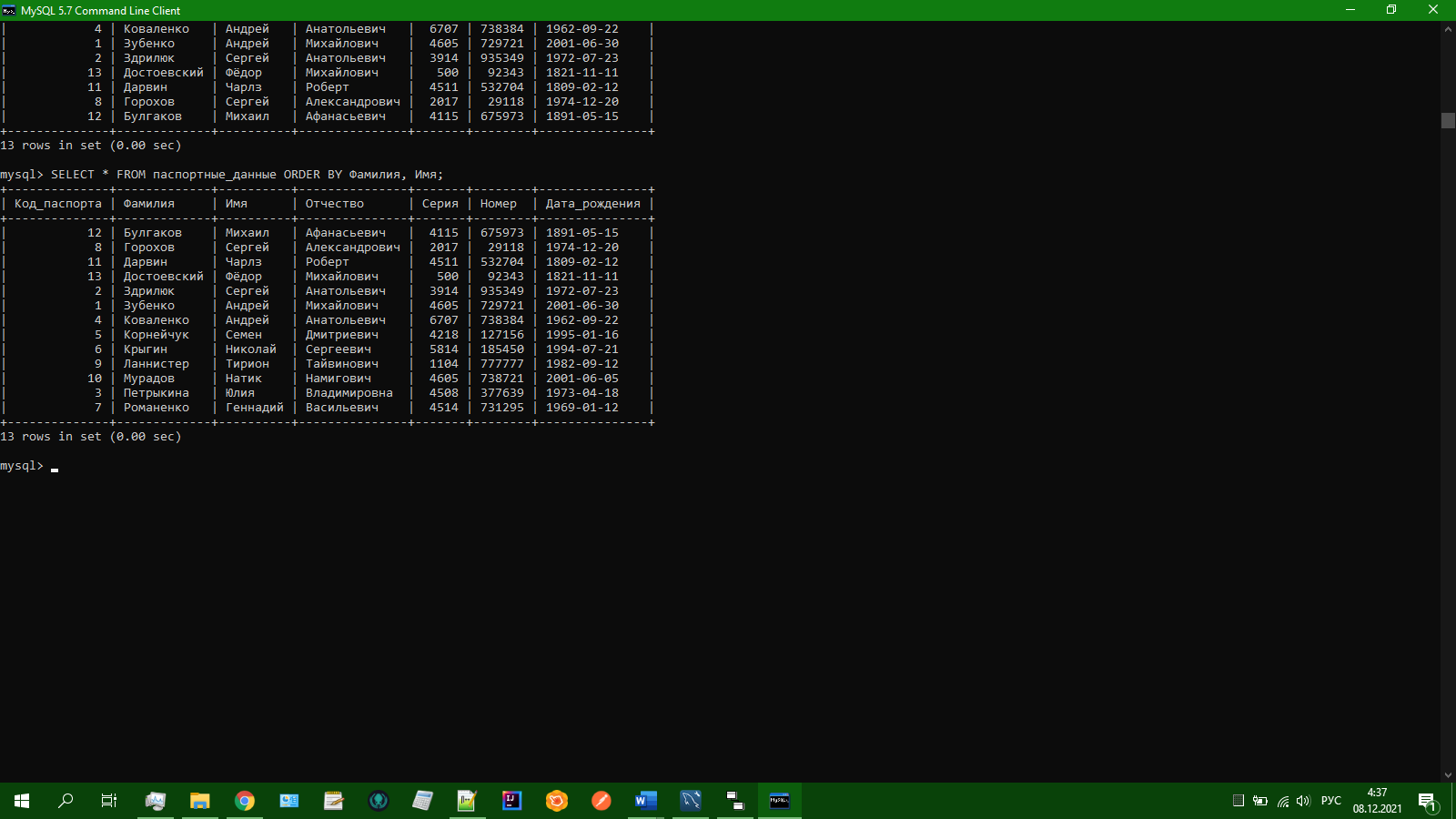
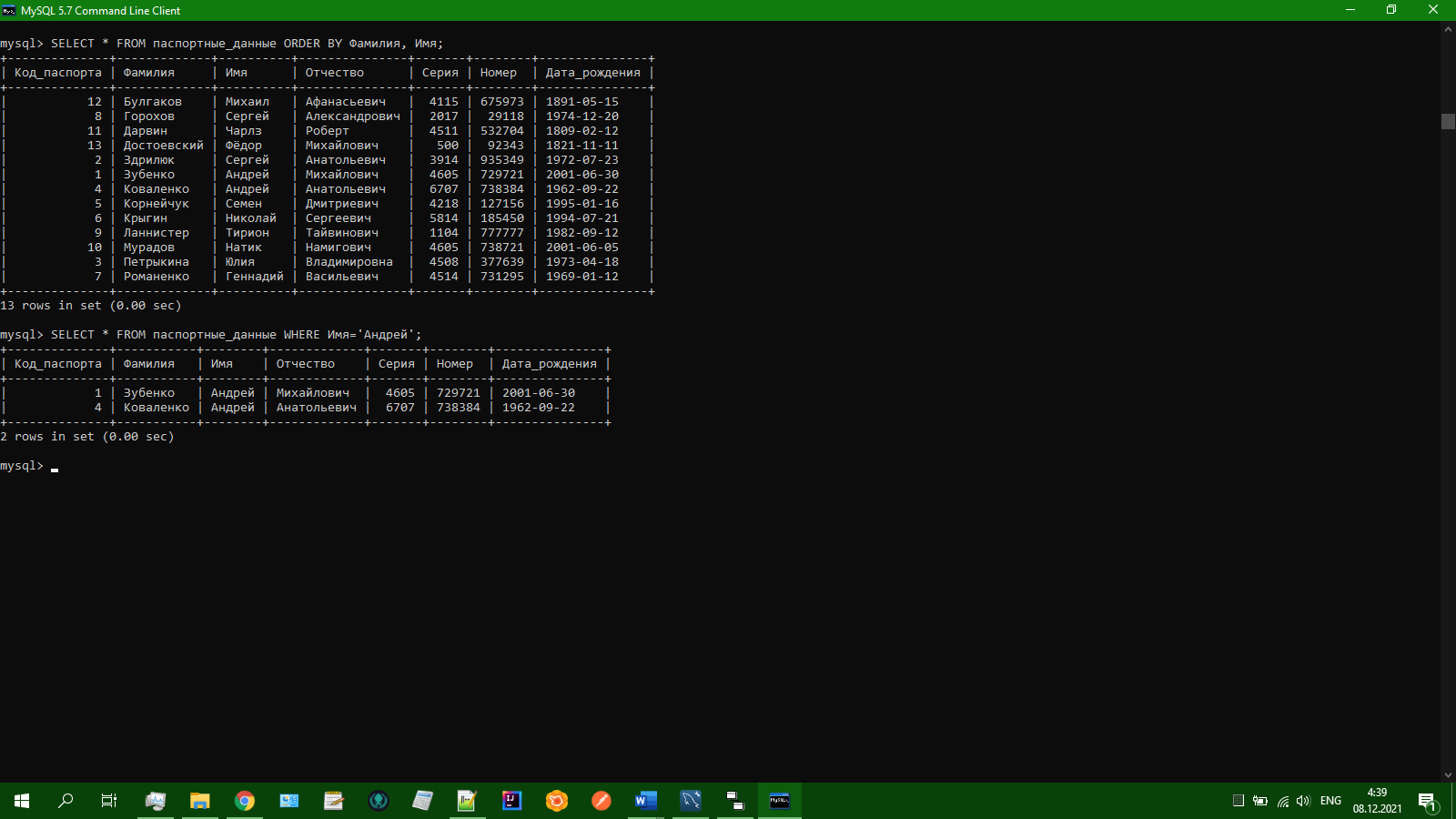
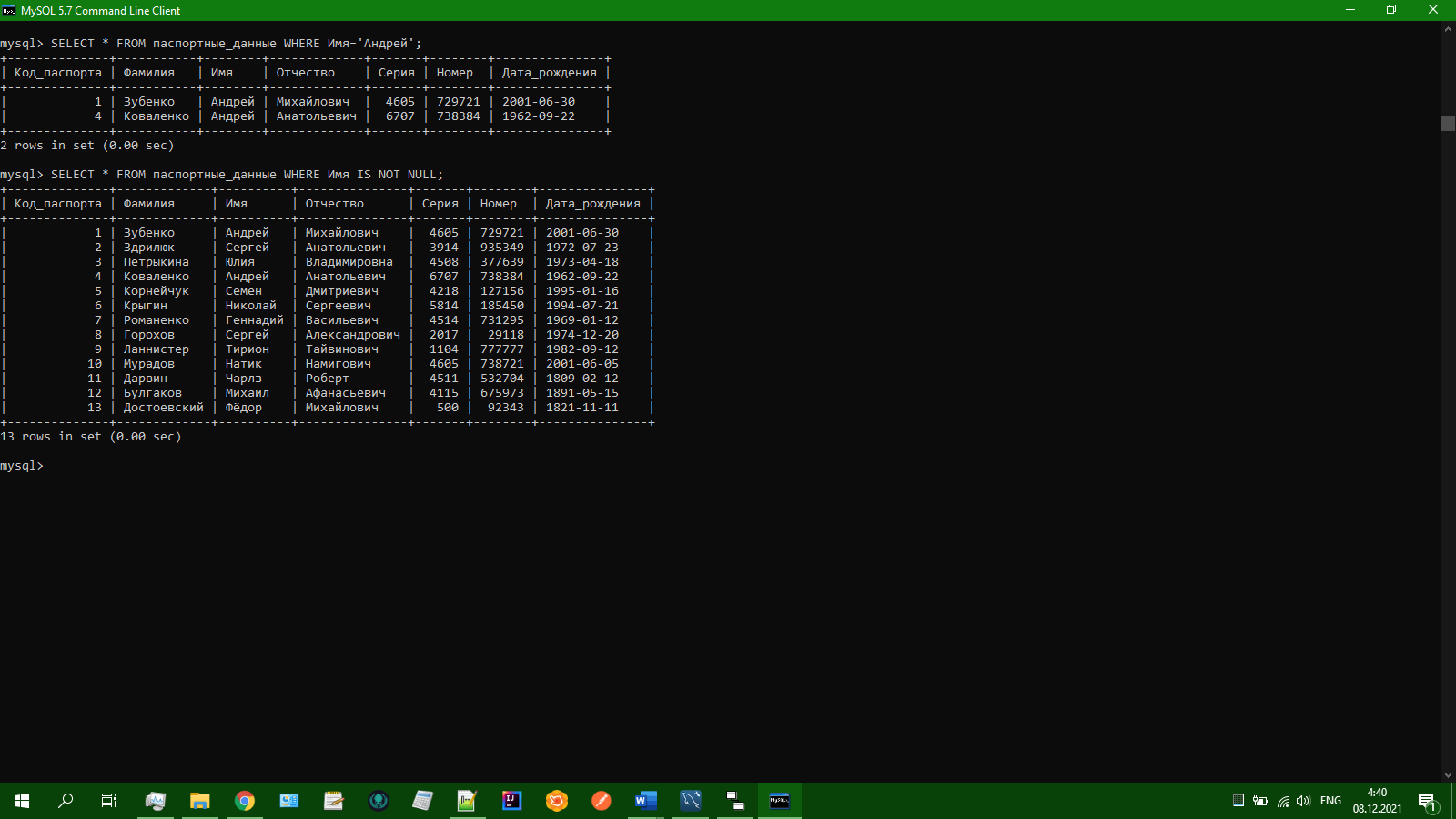


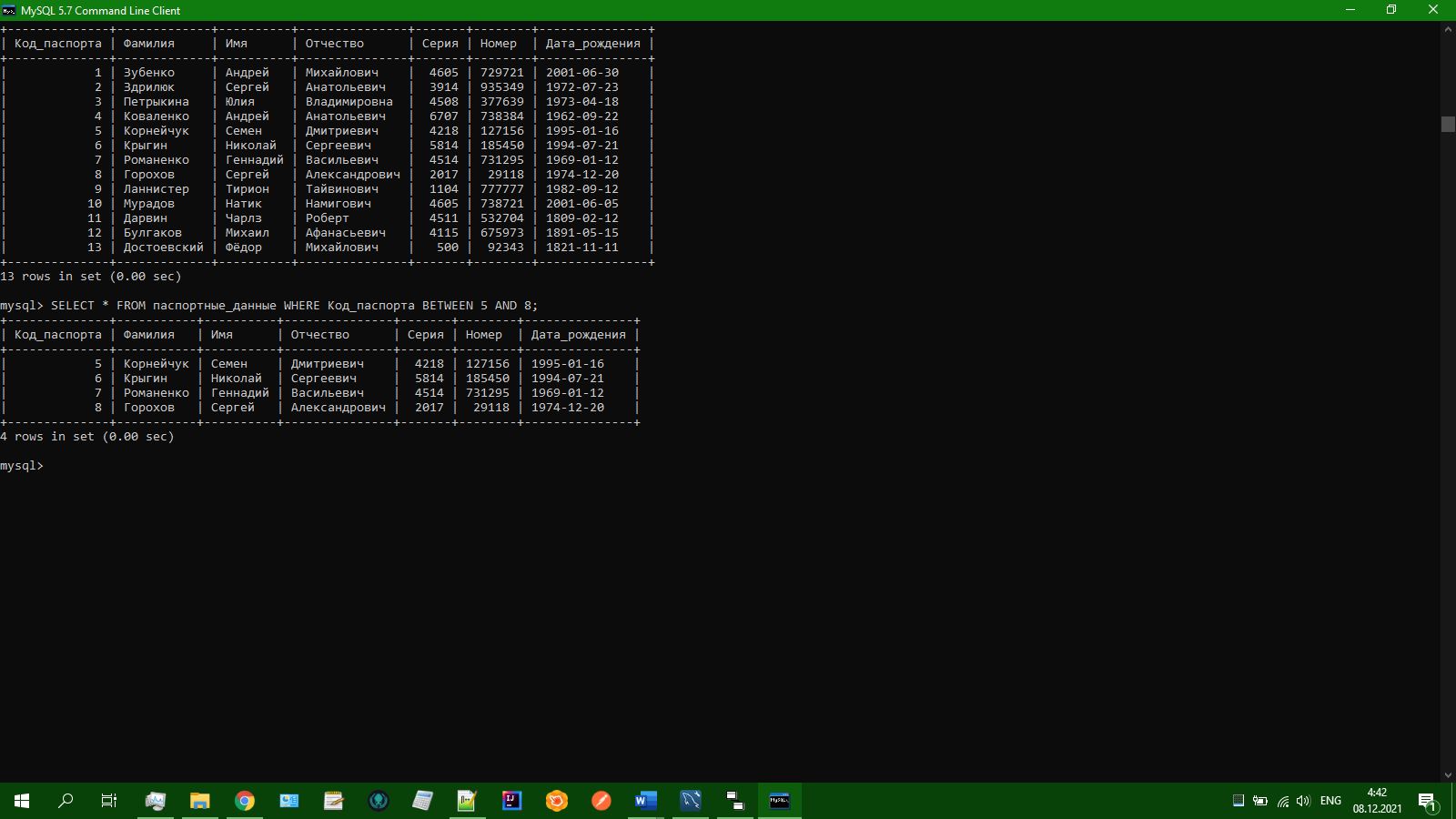
Рис. 30 ‒ Выборка с использованием LEFT JOIN часть 2

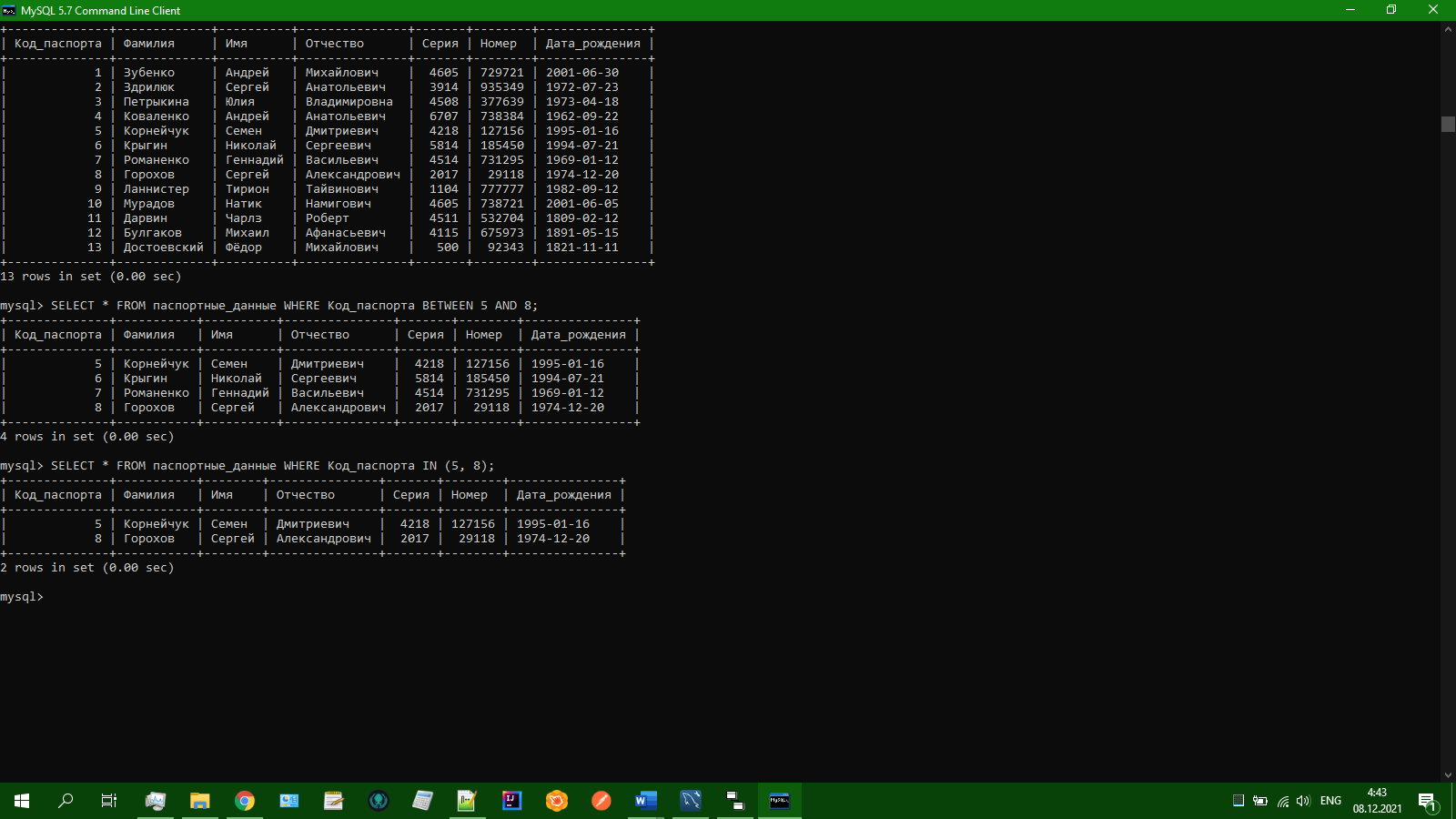
Рис. 31 Выборка данных по параметру deadline

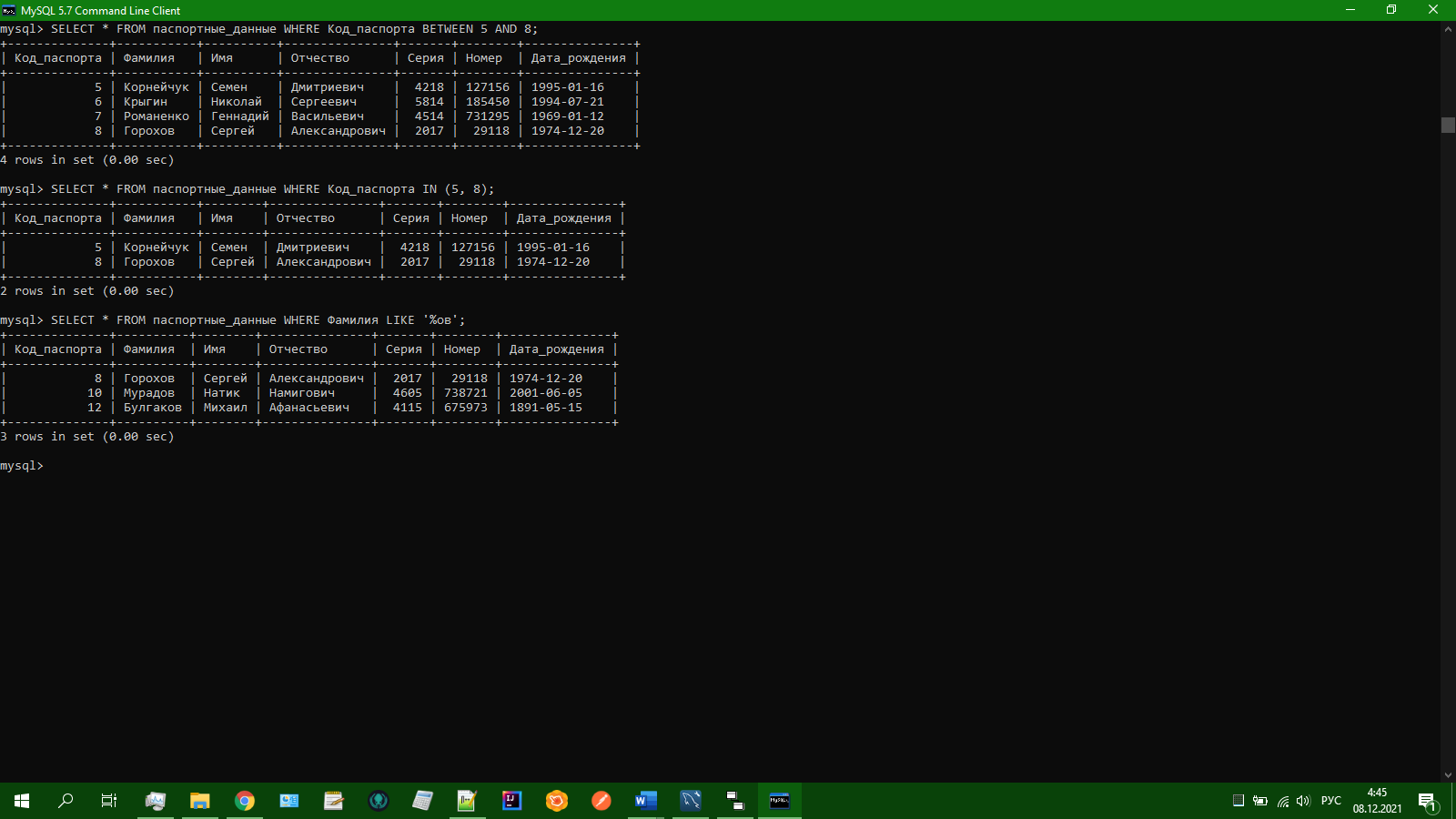
Рис. 32 Выборка данных по параметру deadline

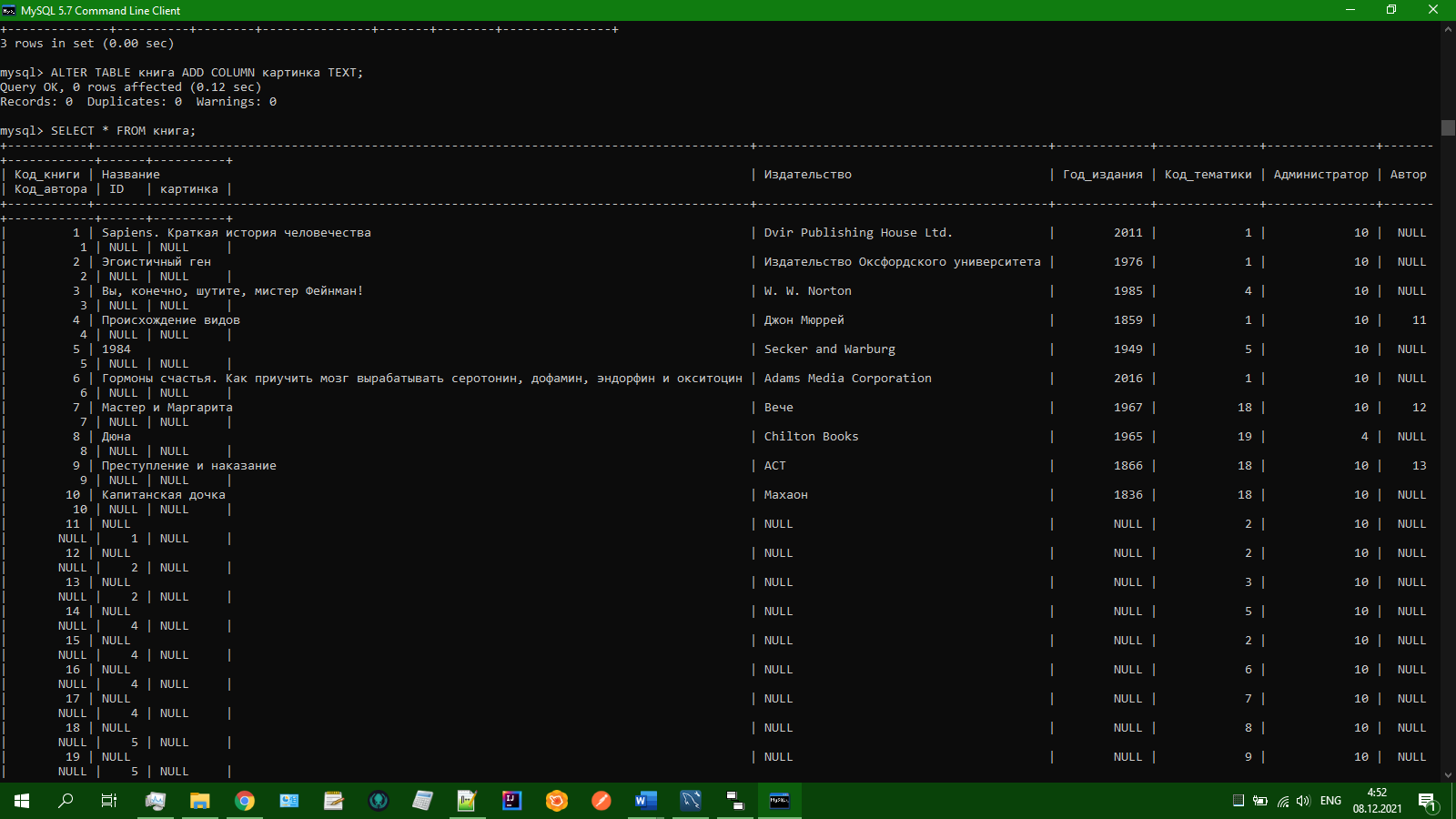
Рис. 33 Выборка данных по параметру deadline

Рис. 34 Выборка данных по параметру deadline

Рис. 35 Выборка данных по параметру deadline

Рис. 36 Выборка данных по параметру deadline

Рис. 37 Выборка данных по параметру deadline

Рис. 38 Выборка данных по параметру deadline

# Практическое задание 7

**Построение модели с помощью оболочки MySQL Workbench**

Мною была построена модель в СУБД Workbench(Рисунок 39)

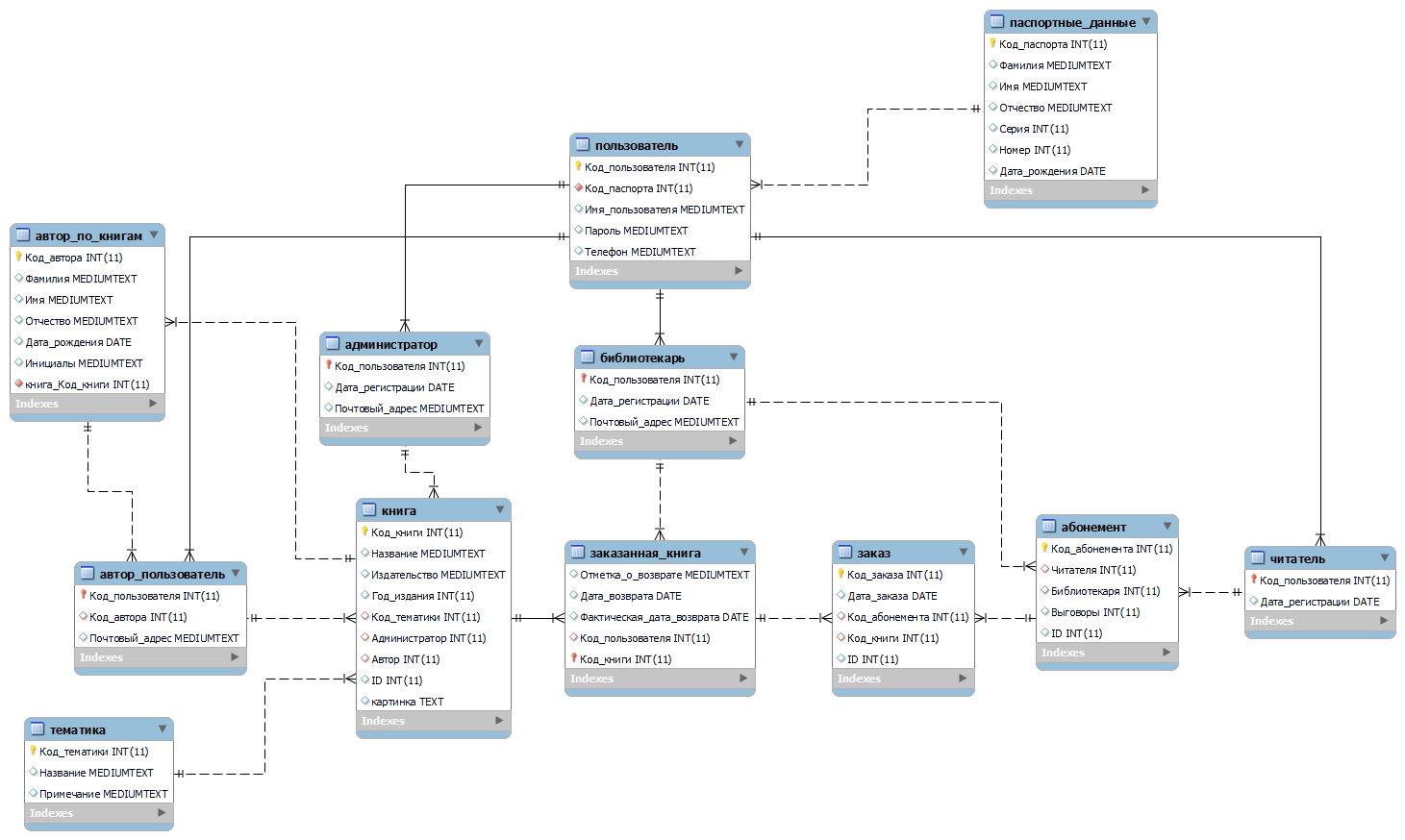


Рисунок 39 – Построенная модель

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**

В таблицу address были внесены данные, соответствующие местам нахождения мастерских(Рисунок 40)

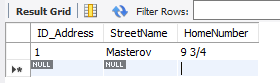


Рисунок 40 – Внесенные данные

Данные в таблице client соответствуют клиентам, которые собираются воспользоваться услугами(Рисунок 41)

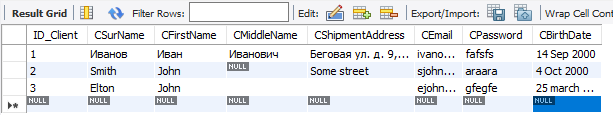


Рисунок 41 – Внесенные данные

Данные в таблице condition соответствуют состояниям инструментов(Рисунок 42)

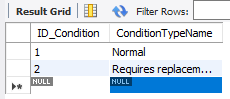


Рисунок 42 – Внесенные данные

Данные в таблице materialprovider соответствуют информации о поставщиках материалов (Рисунок 43)

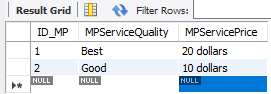


Рисунок 43 – Внесенные данные

Данные в таблице order соответствуют данным о заказах(Рисунок 44)

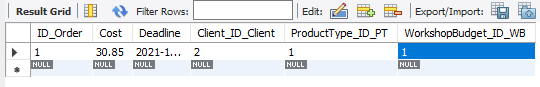


Рисунок 44 – Внесенные данные

Данные в таблице productype соответствуют предлагаемым вариантам товаров(Рисунок 45)

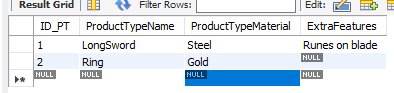


Рисунок 45 – Внесенные данные

Данные в таблице workequipment соответствуют информации об экипировке(Рисунок 46)

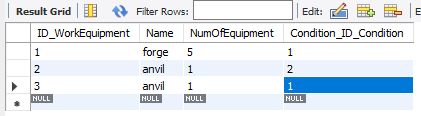


Рисунок 46 – Внесенные данные

Данные в таблице workshopbudget соответствуют состоянию бюджета (Рисунок 47)

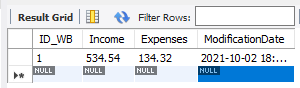


Рисунок 47 – Внесенные данные

Данные в таблице workshopemployee отражают информацию о сотрудниках(Рисунок 48)

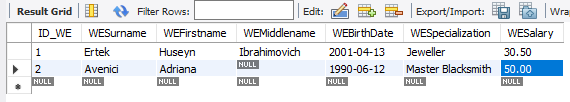


Рисунок 48 – Внесенные данные

Данные в таблице workshoproom отражают информацию о помещениях(Рисунок 49)

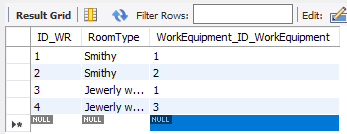


Рисунок 49 – Внесенные данные

Данные в таблице workshop отражают информацию о мастерской в целом(Рисунок 50)

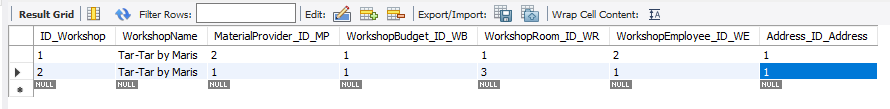


Рисунок 50 – Внесенные данные

**Выборка данных. Оператор SELECT**

Операция проекции (Рисунок 51)

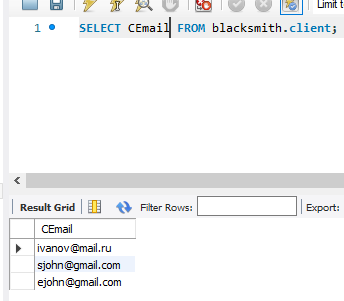
****

Рисунок 51 – Проекция

Операция селекции (Рисунок 52)

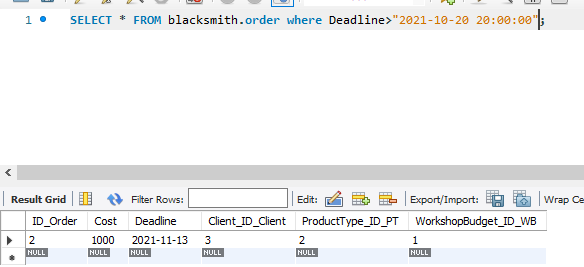


Рисунок 52 – Селекция

Операция соединения(Рисунок 53)

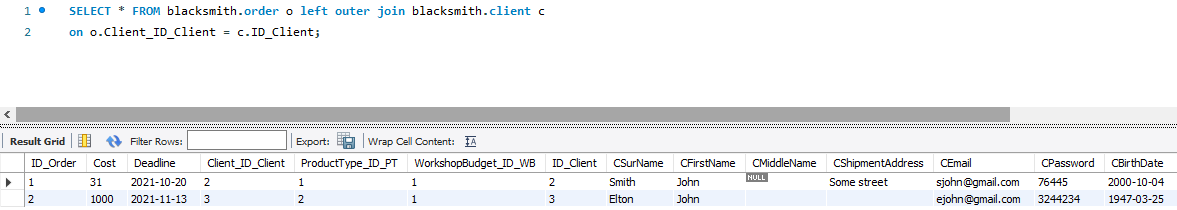


Рисунок 53 – Соединение

Операция объединения(Рисунок 54)

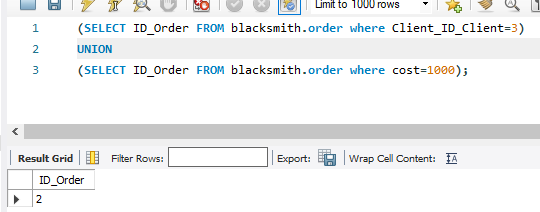


Рисунок 54 – Объединение

**Создание триггеров.**

Был создан триггер, который при определенной цене заказа уведомляет о необходимости сделать скидку

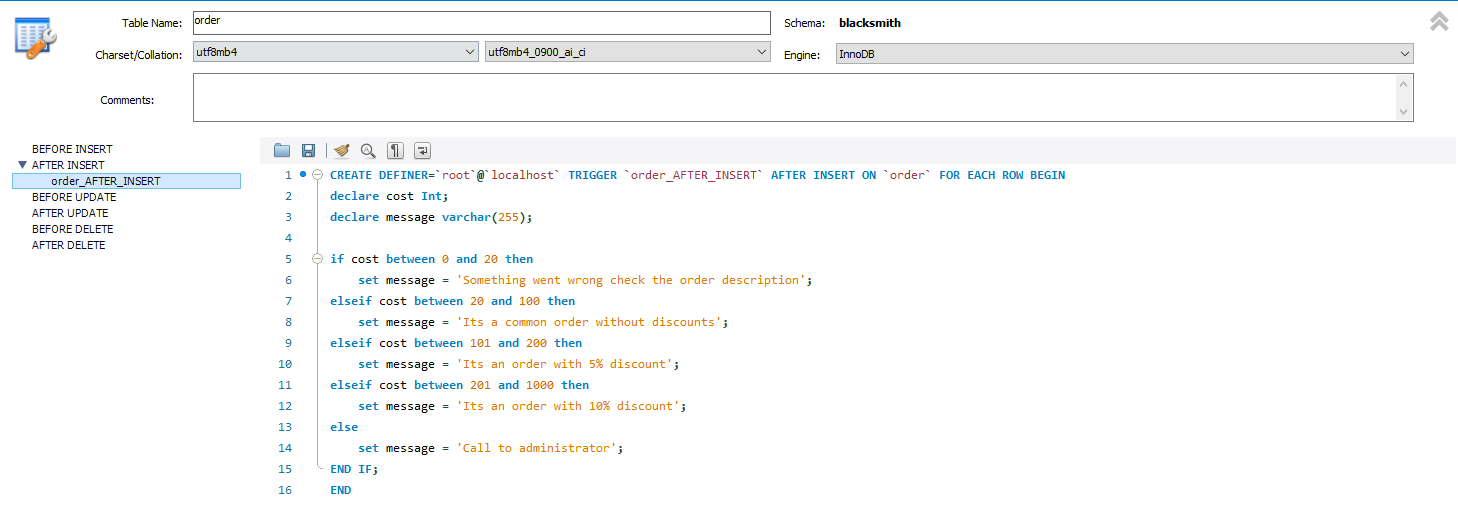


Рисунок 55 ‒ Код триггера

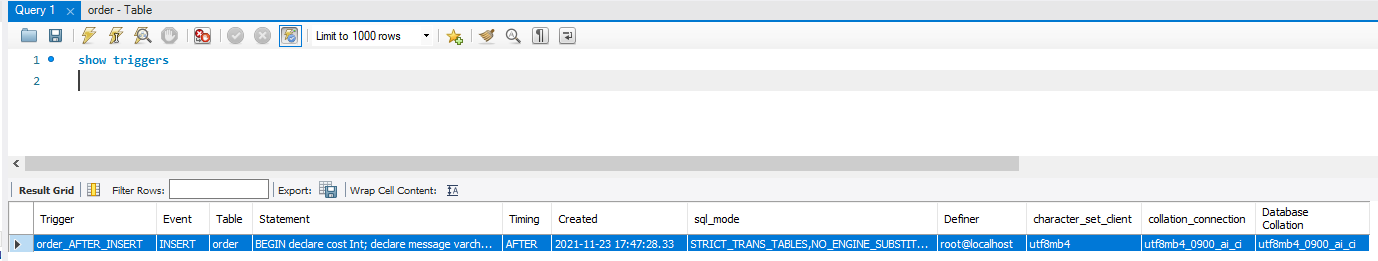


Рисунок 56 ‒ Хранение триггера

**Создание хранимых процедур**

Вызов всех крайних сроков из “order” при помощи хранимой процедуры

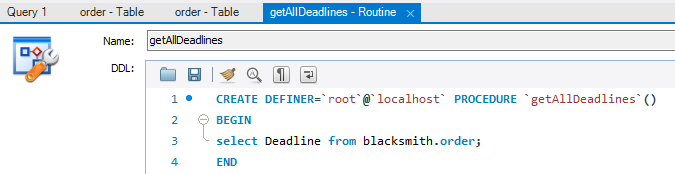


Рисунок 57 ‒ Код процедуры

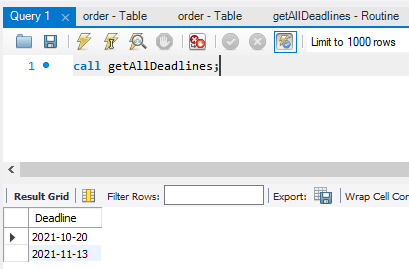


Рисунок 58 ‒ Вызов процедуры

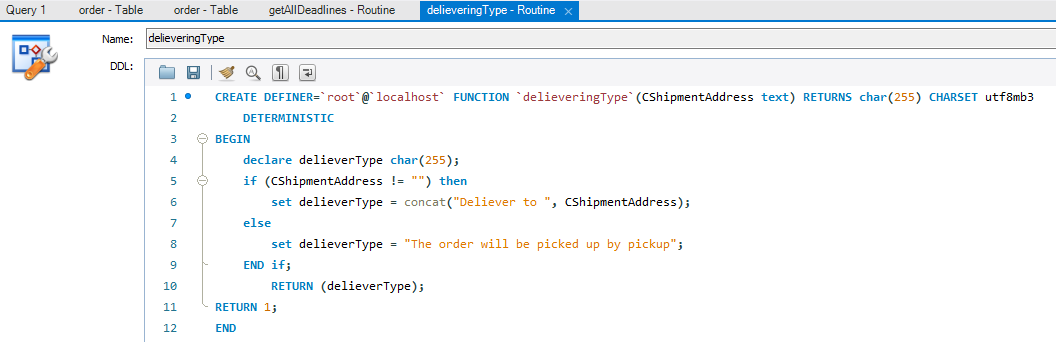


Рисунок 59 ‒ Код функции

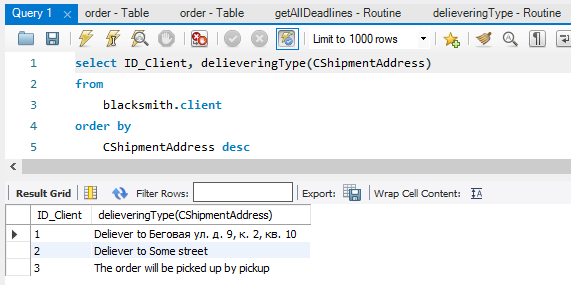


Рисунок 60 ‒ Вызов функции

**Резервное копирование**

Здесь было выполнено копирование базы данных bib:

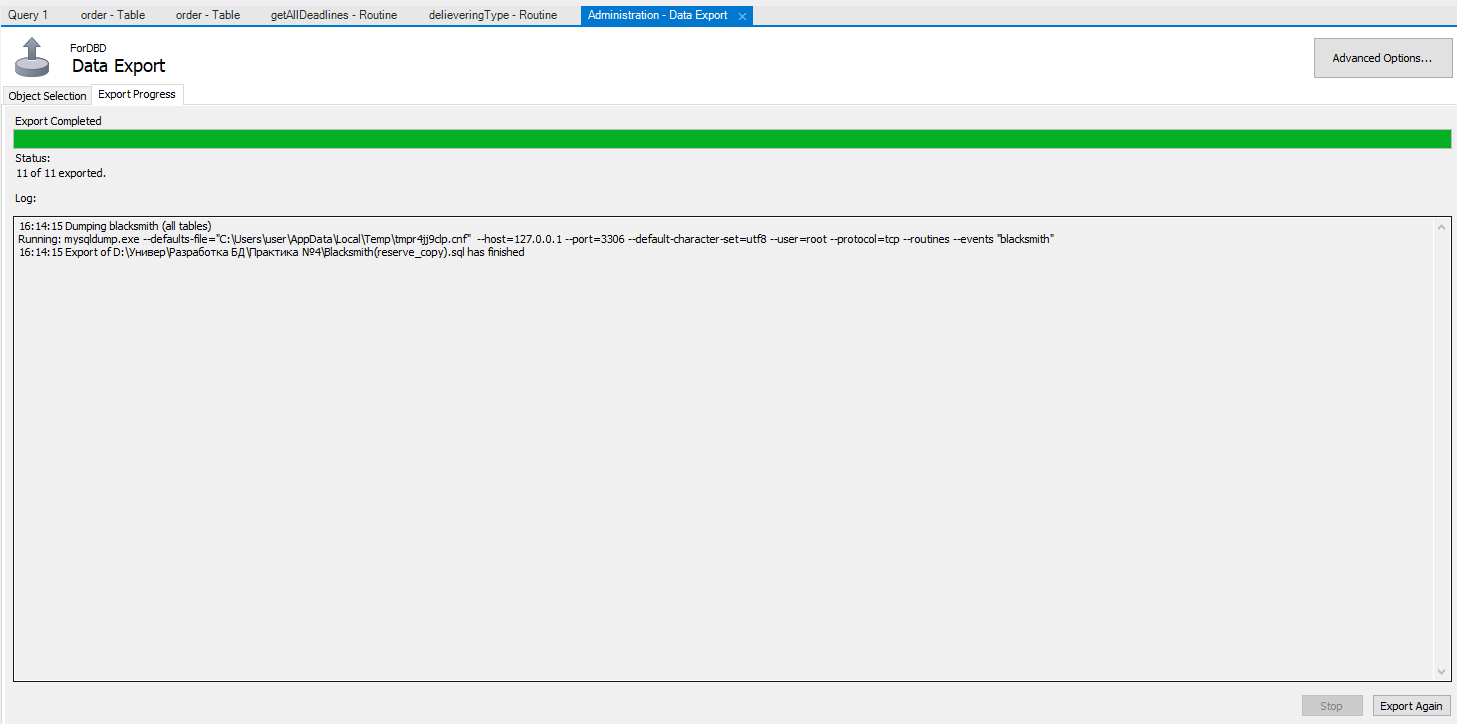


Рисунок 61 ‒ Уведомление об успешном создании резервной копии

**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.