ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Б.К. Акопян |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4 |
| ФИЗИЧЕСКАЯ РЕАЛИЗАЦИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ НА СЕРВЕРЕ |
| по курсу: БАЗЫ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4017 |  |  |  | Г.С. Шкадов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы:**

физическая реализация реляционной БД на MySQL - сервере.

**Вариант 19:**

Агентство по продаже недвижимости. В Гуапландии любят строить новые дома. Для их продажи создано агентство по продаже недвижимости. Каждый гуапчанин, который заинтересован в новой норке (количество комнат любое), может осмотреть несколько объектов недвижимости, и каждый объект может быть осмотрен несколькими клиентами. Каждый агент по недвижимости обслуживает несколько клиентов, но каждый клиент обслуживается только одним сотрудником. Каждый клиент может заключить несколько договоров купли и аренды (вдруг у него семья большая или он разъехаться хочет со своими домочадцами). Один владелец может владеть несколькими объектами недвижимости, но у каждого объекта недвижимости может быть только один владелец. Бывают случаи, когда гуапчанин отказывается покупать осмотренную недвижимость и к нему не предъявляют никаких претензий.

**Ход работы:**

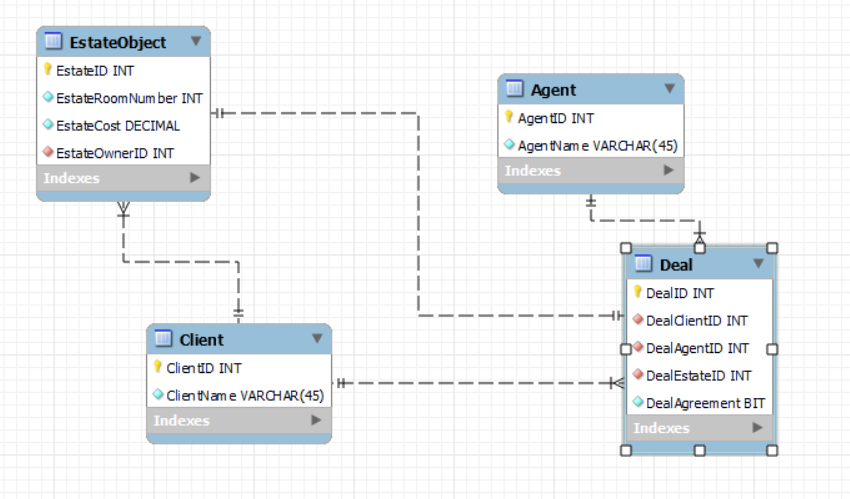


Рисунок 1 – Схема данных

Был автоматически сгенерирован программный код создания базы данных.

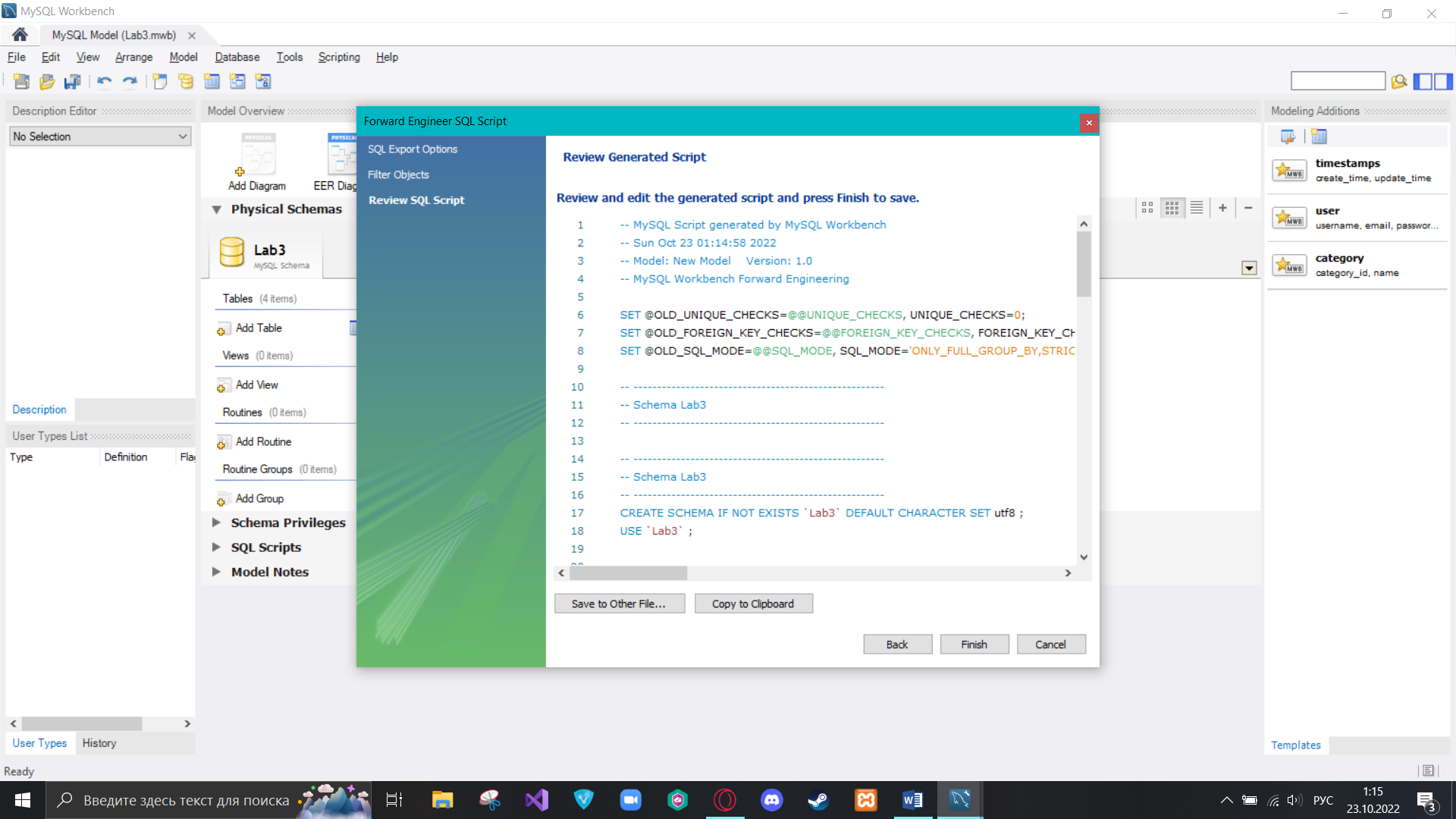


Рисунок 2 – Создание программного кода

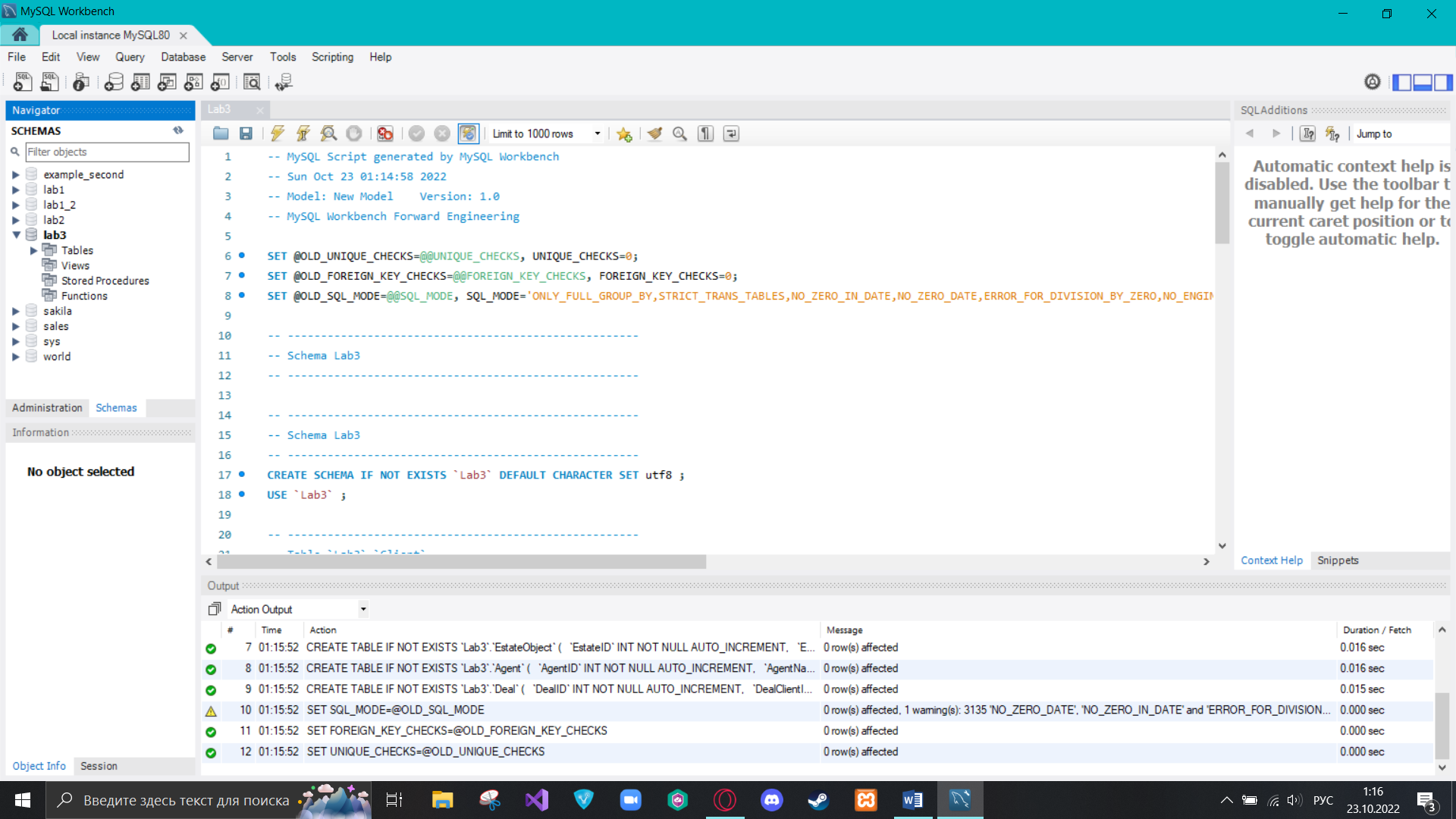


Рисунок 3 – Создание БД на основе сгенерированного программного кода

Далее была заполнена таблица ‘agent’.

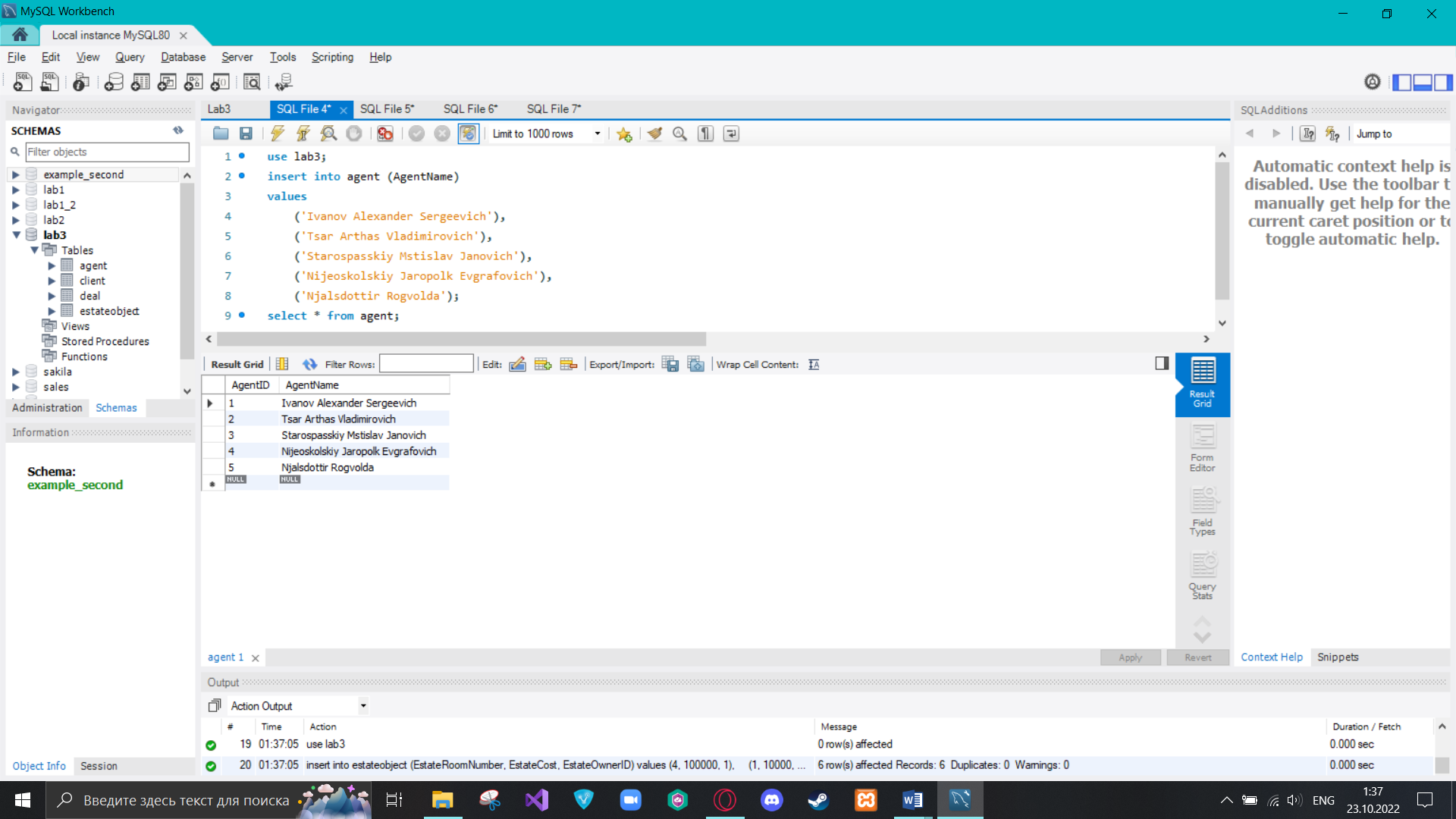


Рисунок 4 – Заполнение таблицы ‘agent’

Далее была заполнена таблица ‘client’.

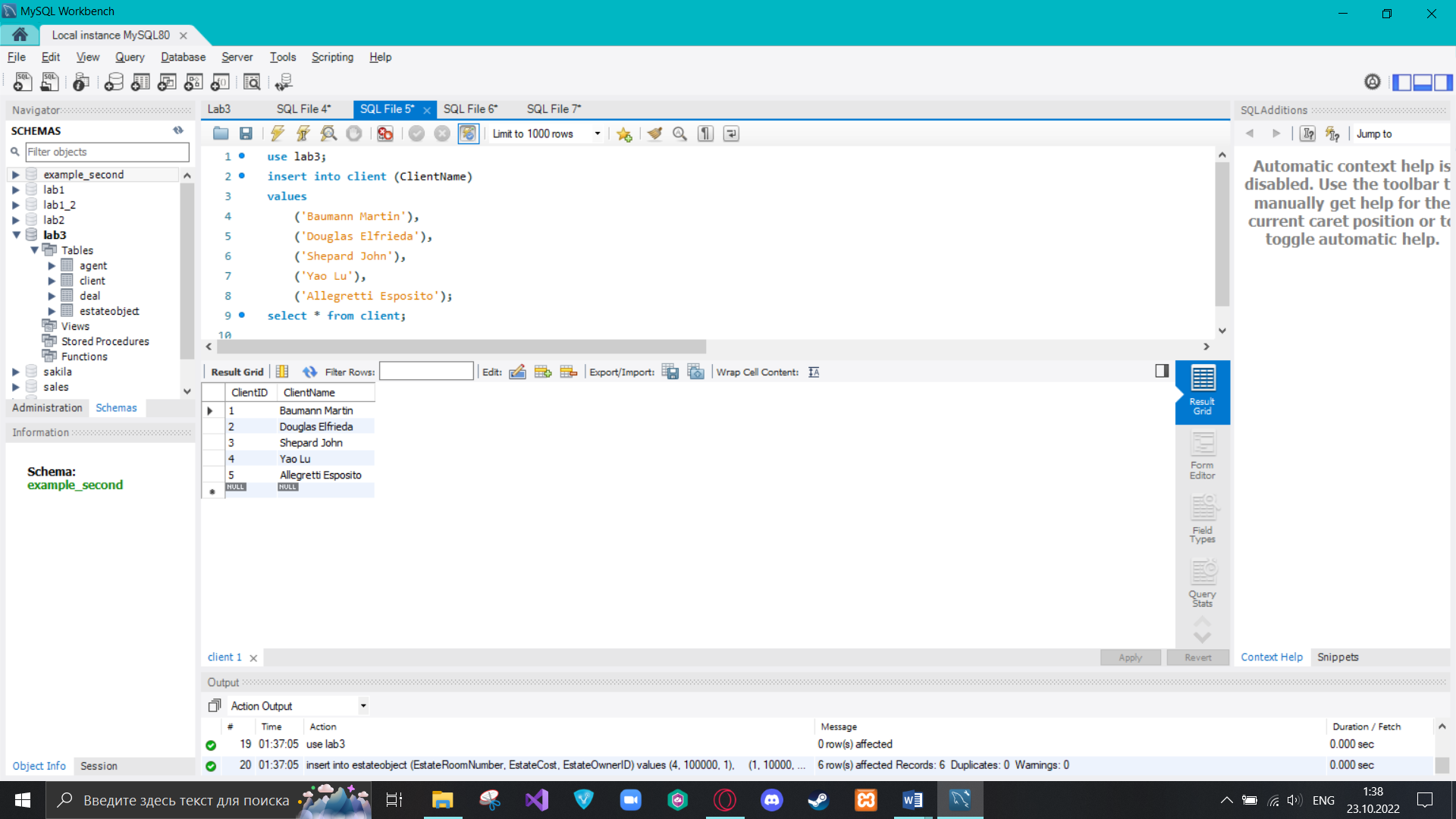


Рисунок 5 – Заполнение таблицы ‘client’

Далее была заполнена таблица ‘estateobject’.

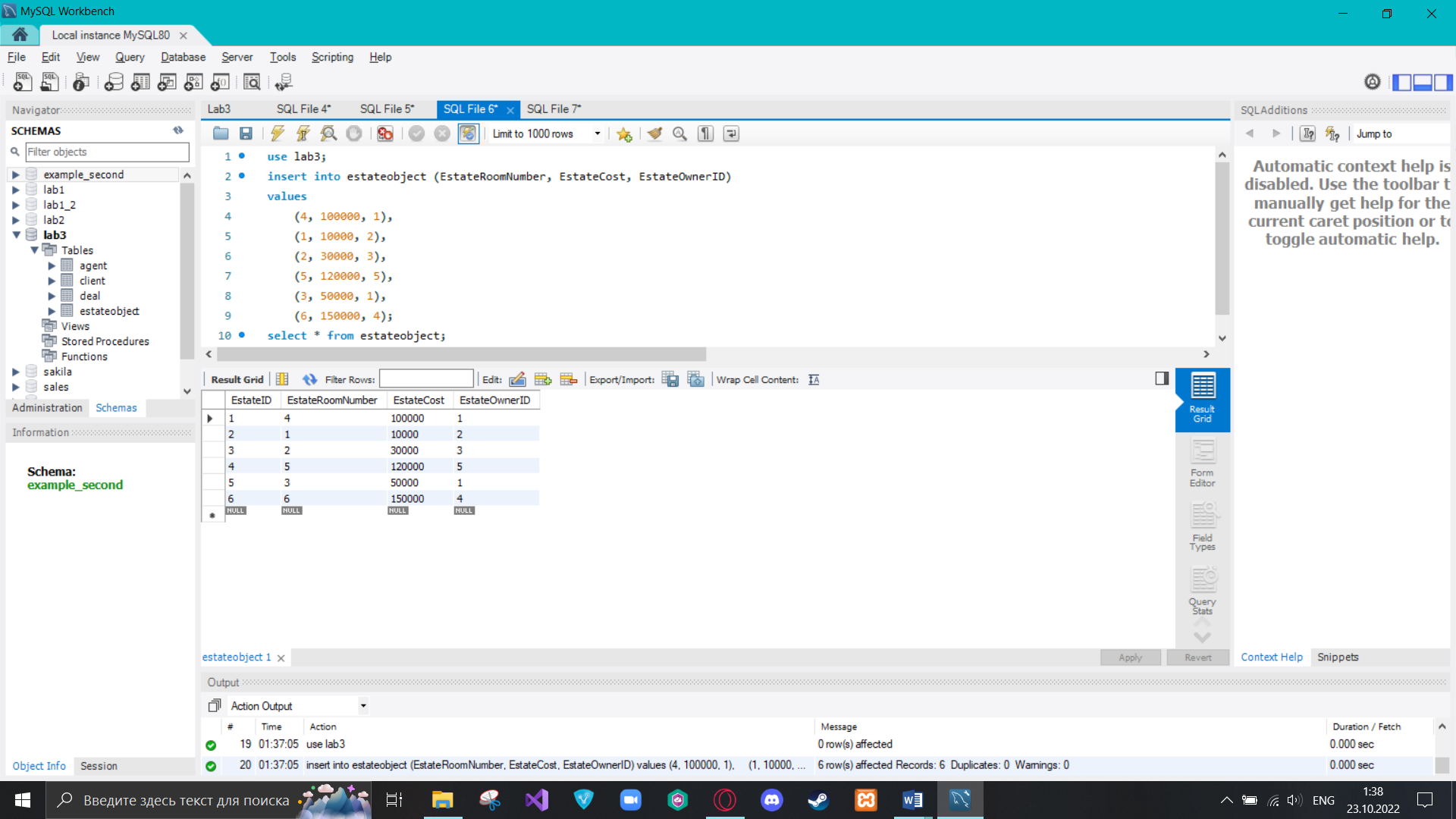


Рисунок 6 – Заполнение таблицы ‘estateobject’

Далее была заполнена таблица ‘deal’.

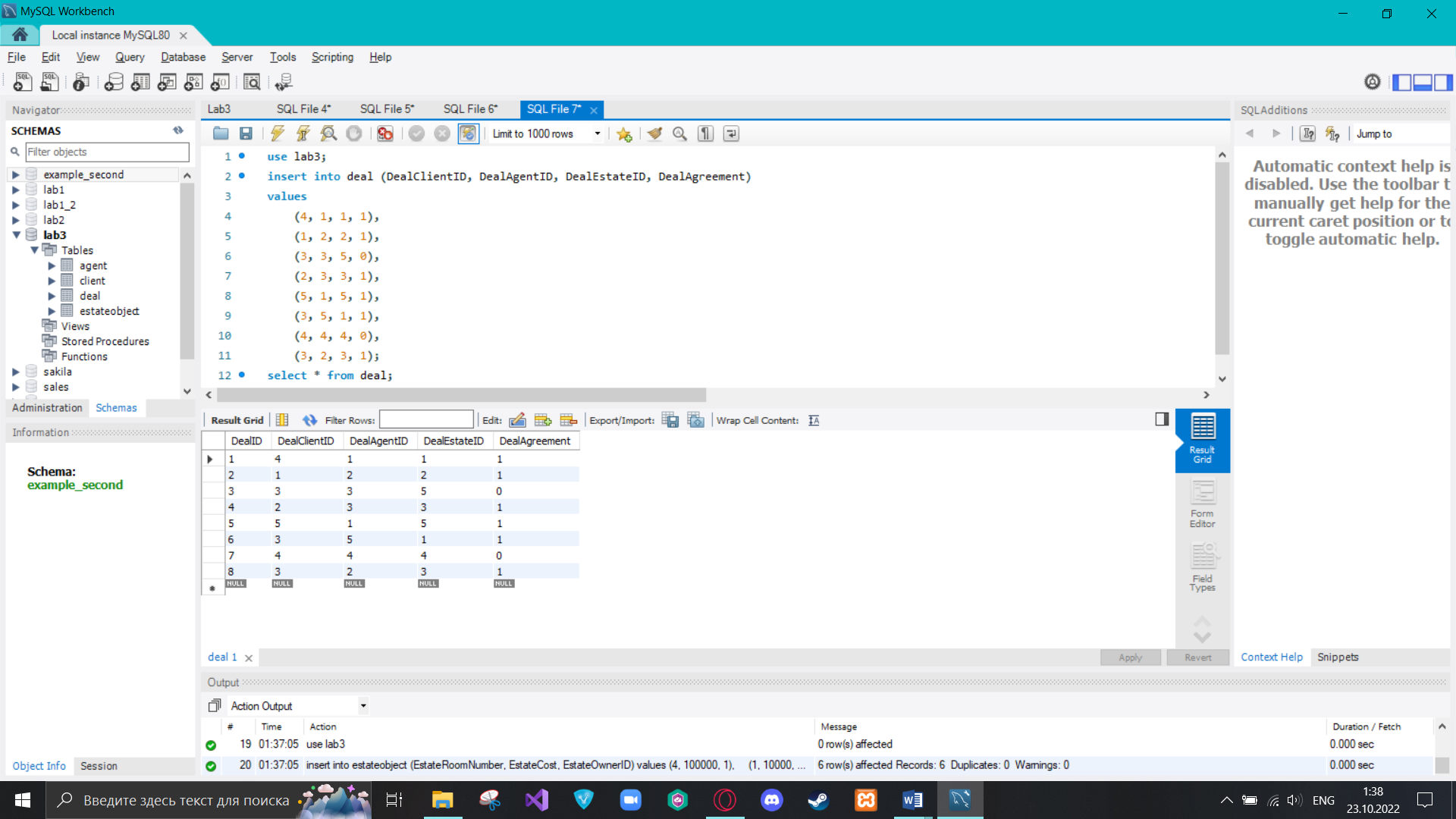


Рисунок 7 – Заполнение таблицы ‘deal’

Далее был выполнен дамп базы при помощи средств MySQL Workbench.

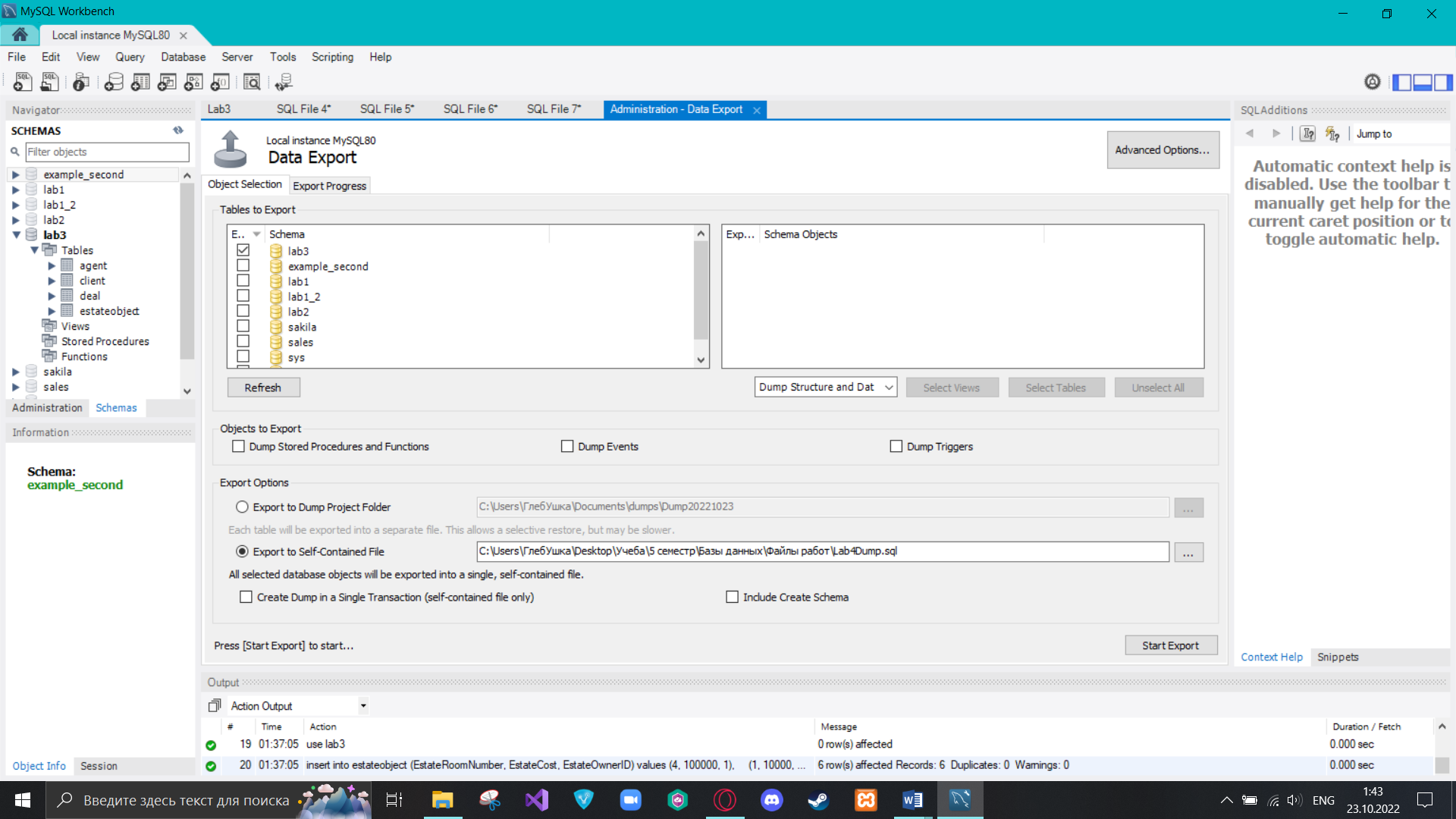


Рисунок 8 – Настройки экспорта БД

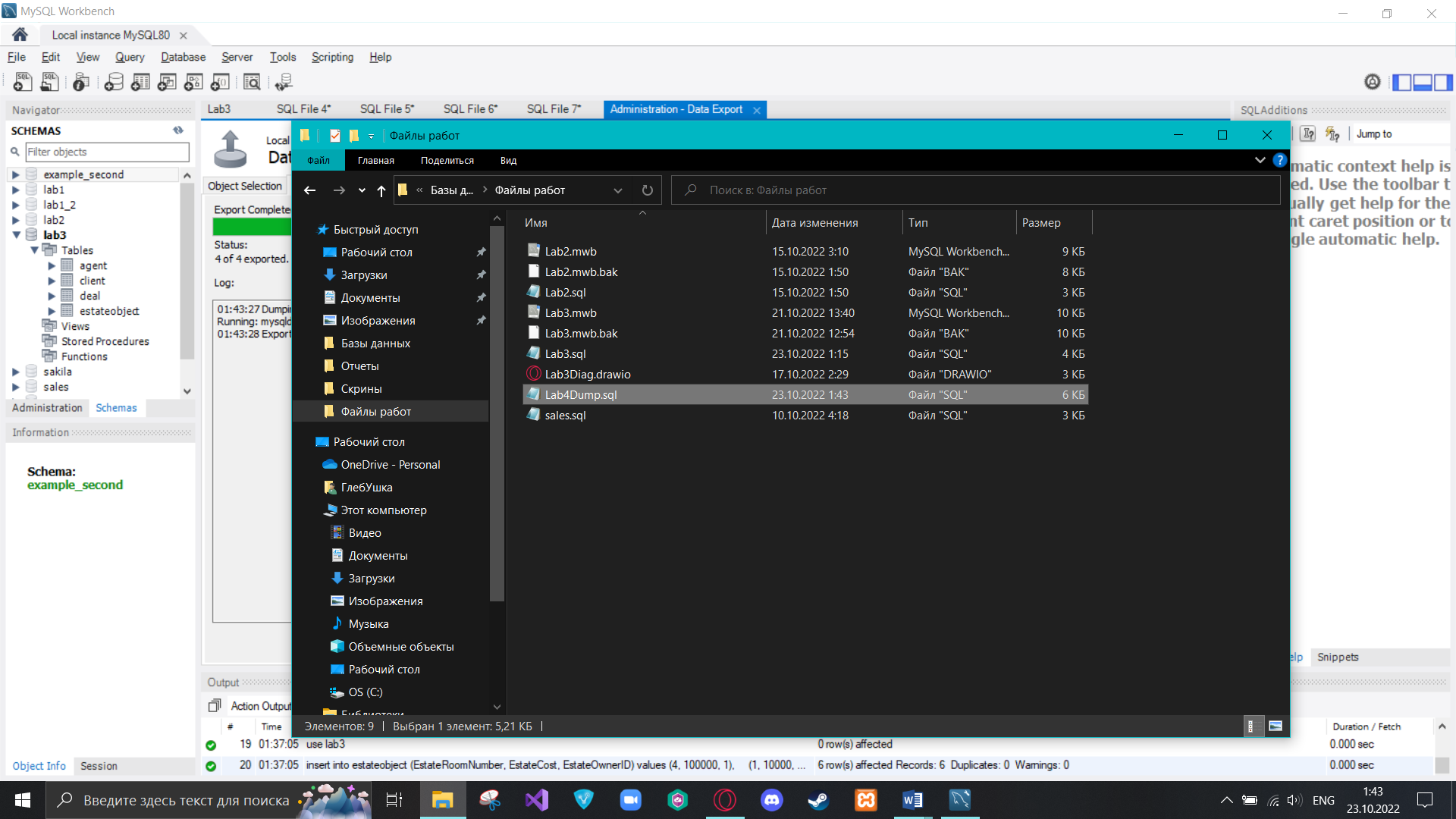


Рисунок 9 – Путь к дампу в файловой системе

**Вывод:**

Во время выполнения данной лабораторной работы я освоил физическую реализацию реляционной БД на MySQLсервере, узнал, что такое дамп БД и как его производить.

**Список использованных источников:**

1) Разработка и администрирование базы данных с открытым исходным кодом/ А.В. Аграновский, В.В. Боженко, Е.Л. Турнецкая. ‒ СПб.: ГУАП, 2022. ‒ 213 с., ил.

**Приложение A:**

**Автосгенерированный программный код на основе модели данных:**

-- MySQL Script generated by MySQL Workbench

-- Sun Oct 23 01:14:58 2022

-- 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 Lab3

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

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

-- Schema Lab3

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

CREATE SCHEMA IF NOT EXISTS `Lab3` DEFAULT CHARACTER SET utf8 ;

USE `Lab3` ;

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

-- Table `Lab3`.`Client`

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

CREATE TABLE IF NOT EXISTS `Lab3`.`Client` (

`ClientID` INT NOT NULL AUTO\_INCREMENT,

`ClientName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ClientID`))

ENGINE = InnoDB;

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

-- Table `Lab3`.`EstateObject`

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

CREATE TABLE IF NOT EXISTS `Lab3`.`EstateObject` (

`EstateID` INT NOT NULL AUTO\_INCREMENT,

`EstateRoomNumber` INT NOT NULL,

`EstateCost` DECIMAL NOT NULL,

`EstateOwnerID` INT NOT NULL,

PRIMARY KEY (`EstateID`),

INDEX `EstateOwnerID\_idx` (`EstateOwnerID` ASC) VISIBLE,

CONSTRAINT `EstateOwnerID`

FOREIGN KEY (`EstateOwnerID`)

REFERENCES `Lab3`.`Client` (`ClientID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `Lab3`.`Agent`

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

CREATE TABLE IF NOT EXISTS `Lab3`.`Agent` (

`AgentID` INT NOT NULL AUTO\_INCREMENT,

`AgentName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`AgentID`))

ENGINE = InnoDB;

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

-- Table `Lab3`.`Deal`

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

CREATE TABLE IF NOT EXISTS `Lab3`.`Deal` (

`DealID` INT NOT NULL AUTO\_INCREMENT,

`DealClientID` INT NOT NULL,

`DealAgentID` INT NOT NULL,

`DealEstateID` INT NOT NULL,

`DealAgreement` BIT NOT NULL,

PRIMARY KEY (`DealID`),

INDEX `DealClientID\_idx` (`DealClientID` ASC) VISIBLE,

INDEX `DealAgentID\_idx` (`DealAgentID` ASC) VISIBLE,

INDEX `DealClientID\_idx1` (`DealEstateID` ASC) VISIBLE,

CONSTRAINT `DealClientID`

FOREIGN KEY (`DealClientID`)

REFERENCES `Lab3`.`Client` (`ClientID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `DealAgentID`

FOREIGN KEY (`DealAgentID`)

REFERENCES `Lab3`.`Agent` (`AgentID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `DealEstateID`

FOREIGN KEY (`DealEstateID`)

REFERENCES `Lab3`.`EstateObject` (`EstateID`)

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;

**Скрипт заполнения таблицы ‘agent’:**

use lab3;

insert into agent (AgentName)

values

('Ivanov Alexander Sergeevich'),

('Tsar Arthas Vladimirovich'),

('Starospasskiy Mstislav Janovich'),

('Nijeoskolskiy Jaropolk Evgrafovich'),

('Njalsdottir Rogvolda');

select \* from agent;

**Скрипт заполнения таблицы ‘client’:**

use lab3;

insert into client (ClientName)

values

('Baumann Martin'),

('Douglas Elfrieda'),

('Shepard John'),

('Yao Lu'),

('Allegretti Esposito');

select \* from client;

**Скрипт заполнения таблицы ‘estateobject’:**

use lab3;

insert into estateobject (EstateRoomNumber, EstateCost, EstateOwnerID)

values

(4, 100000, 1),

(1, 10000, 2),

(2, 30000, 3),

(5, 120000, 5),

(3, 50000, 1),

(6, 150000, 4);

select \* from estateobject;

**Скрипт заполнения таблицы ‘deal’:**

use lab3;

insert into deal (DealClientID, DealAgentID, DealEstateID, DealAgreement)

values

(4, 1, 1, 1),

(1, 2, 2, 1),

(3, 3, 5, 0),

(2, 3, 3, 1),

(5, 1, 5, 1),

(3, 5, 1, 1),

(4, 4, 4, 0),

(3, 2, 3, 1);

select \* from deal;