ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ |
| по курсу: БАЗЫ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2022

**Цель работы:**

Разработка инфологической и даталогической моделей реляционной БД.

**Вариант 19:**

Агентство по продаже недвижимости.

**Ход работы:**

Был проведен системный анализ БД:

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.

Далее были выделены сущности, их атрибуты и связи между ними:

Сущности:

1. Клиент (атрибуты: код клиента, ФИО клиента)
2. Агент (атрибуты: код агента, ФИО агента)
3. Объект недвижимости (атрибуты: код объекта недвижимости, количество комнат, стоимость покупки, владелец)
4. Договор (атрибуты: код договора, номер клиента, номер агента, номер объекта недвижимости, заключенность)

Связи:

Один к одному:

Объект недвижимости – Договор

Один ко многим:

Клиент – Объект недвижимости

Агент – Договор

Клиент – Договор

Была создана инфологическая модель БД:

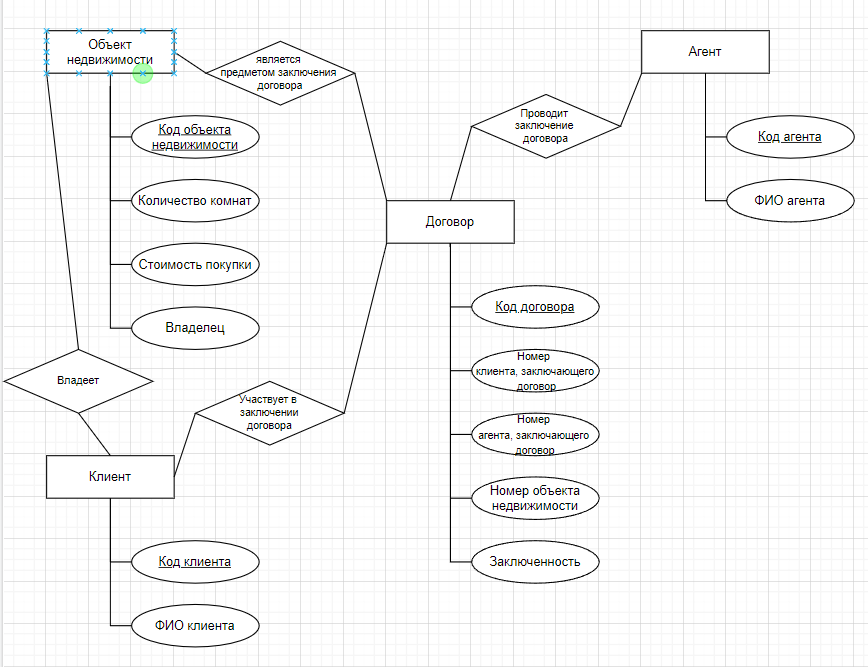


Рисунок 1 – Инфологическая модель

Была создана даталогическая модель БД

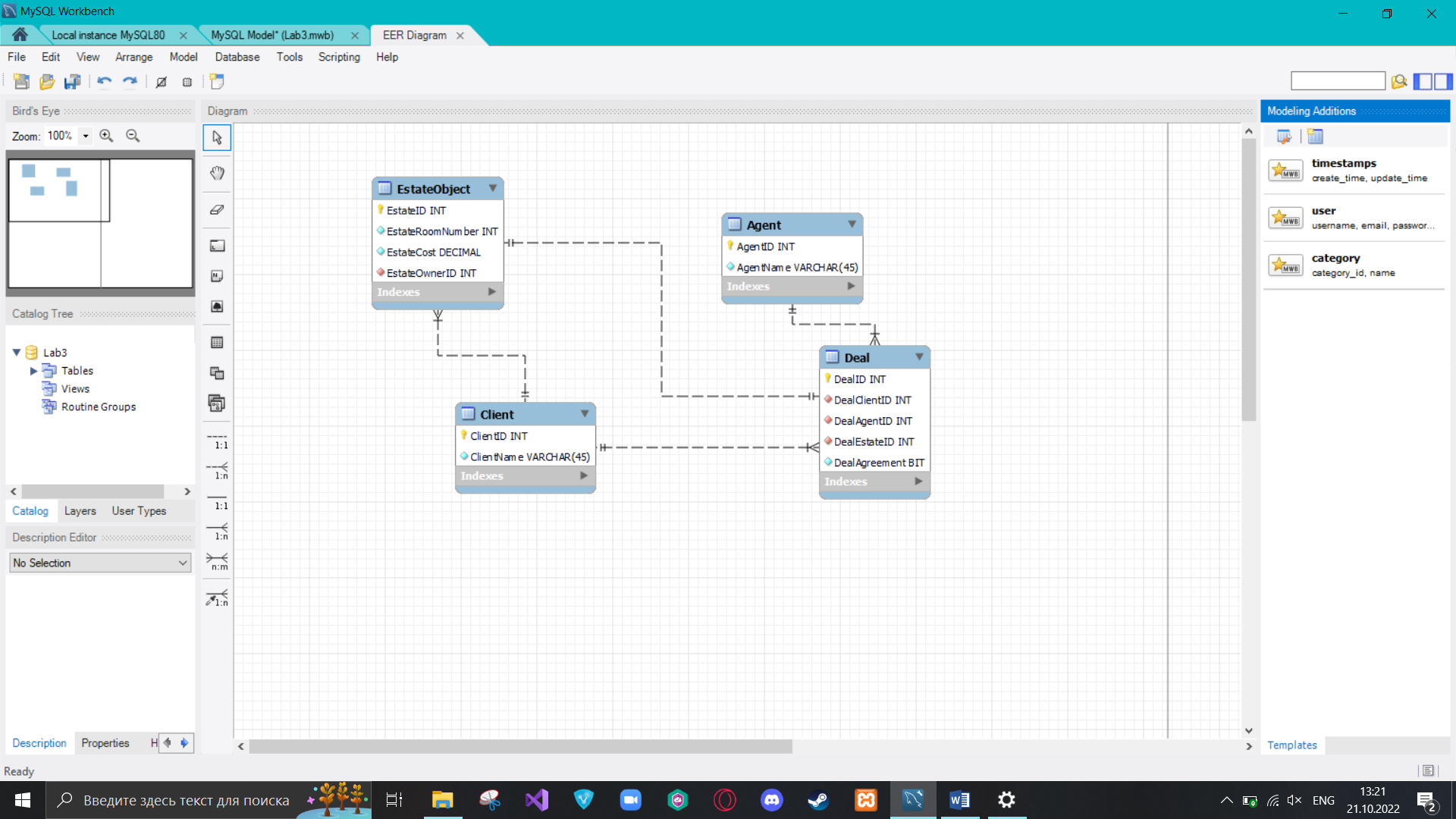


Рисунок 2 – Даталогическая модель

Было проведено описание атрибутов сущностей

Сущность Client:

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Размер (симв.) | Комментарии |
| ClientID | int | 32 | ПК, NN, AI |
| ClientName | varchar | 30 | ФИО клиента |

Таблица 1 – Описание сущности Client

Сущность Agent:

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Размер (симв.) | Комментарии |
| AgentID | int | 32 | ПК, NN, AI |
| AgentName | varchar | 30 | ФИО агента |

Таблица 2 – Описание сущности Agent

Сущность EstateObject:

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Размер (симв.) | Комментарии |
| EstateID | int | 32 | ПК, NN, AI |
| EstateRoomNumber | int | 32 | NN |
| EstateCost | decimal | 32 | NN |
| EstateOwnerID | int | 32 | NN |

Таблица 3 – Описание сущности EstateObject

Сущность Deal:

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Размер (симв.) | Комментарии |
| DealID | int | 32 | ПК, NN, AI |
| DealClientID | int | 32 | NN |
| DealAgentID | int | 32 | NN |
| DealEstateID | int | 32 | NN |
| DealAgreement | bit | 1 | Статус договора |

Таблица 4 – Описание сущности Deal

Далее эта модель была сохранена в формате .mwb

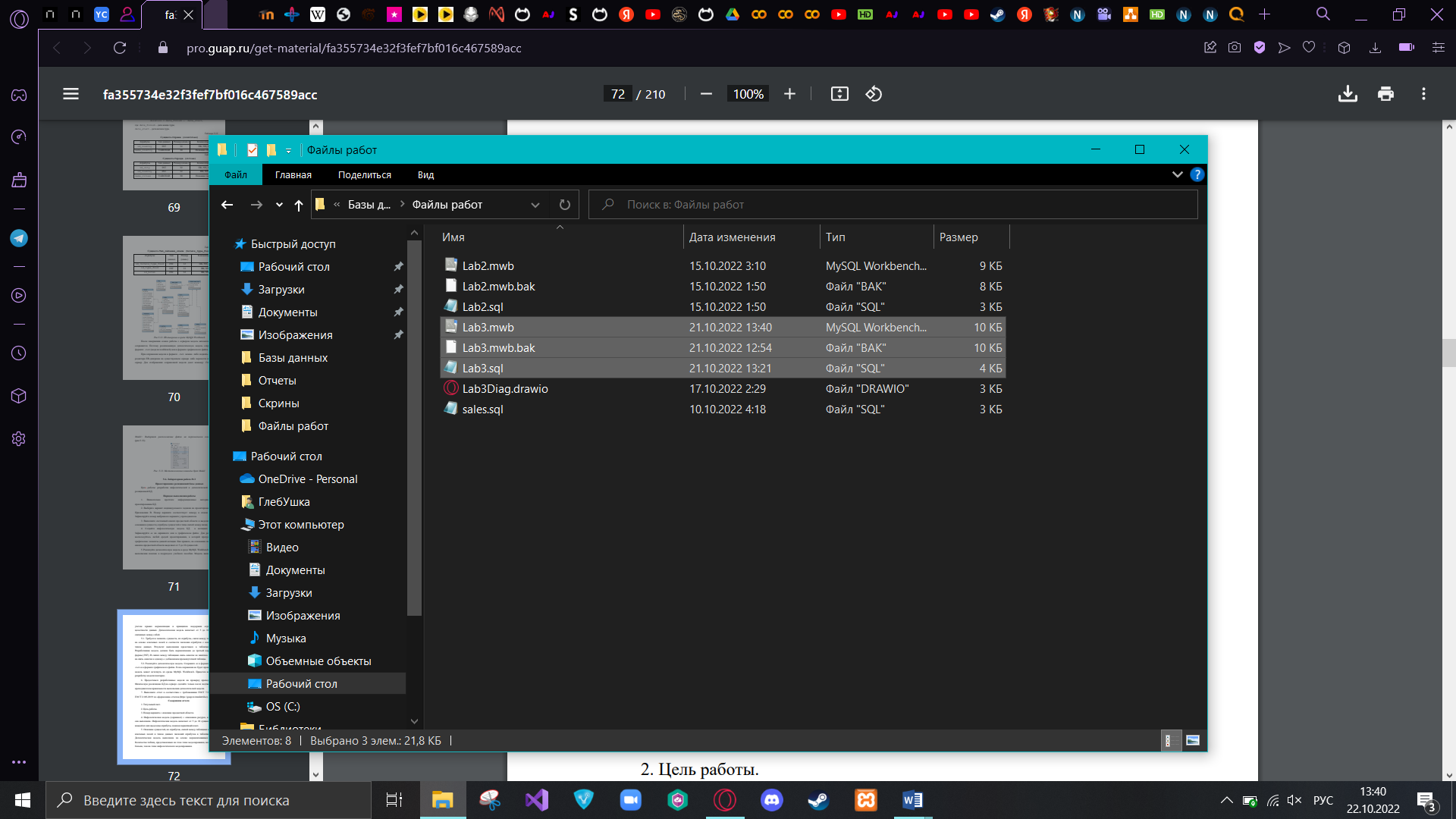


Рисунок 3 – Сохраненная даталогическая модель

Далее модель была импортирована в виде sql-скрипта в MySQL Workbench.

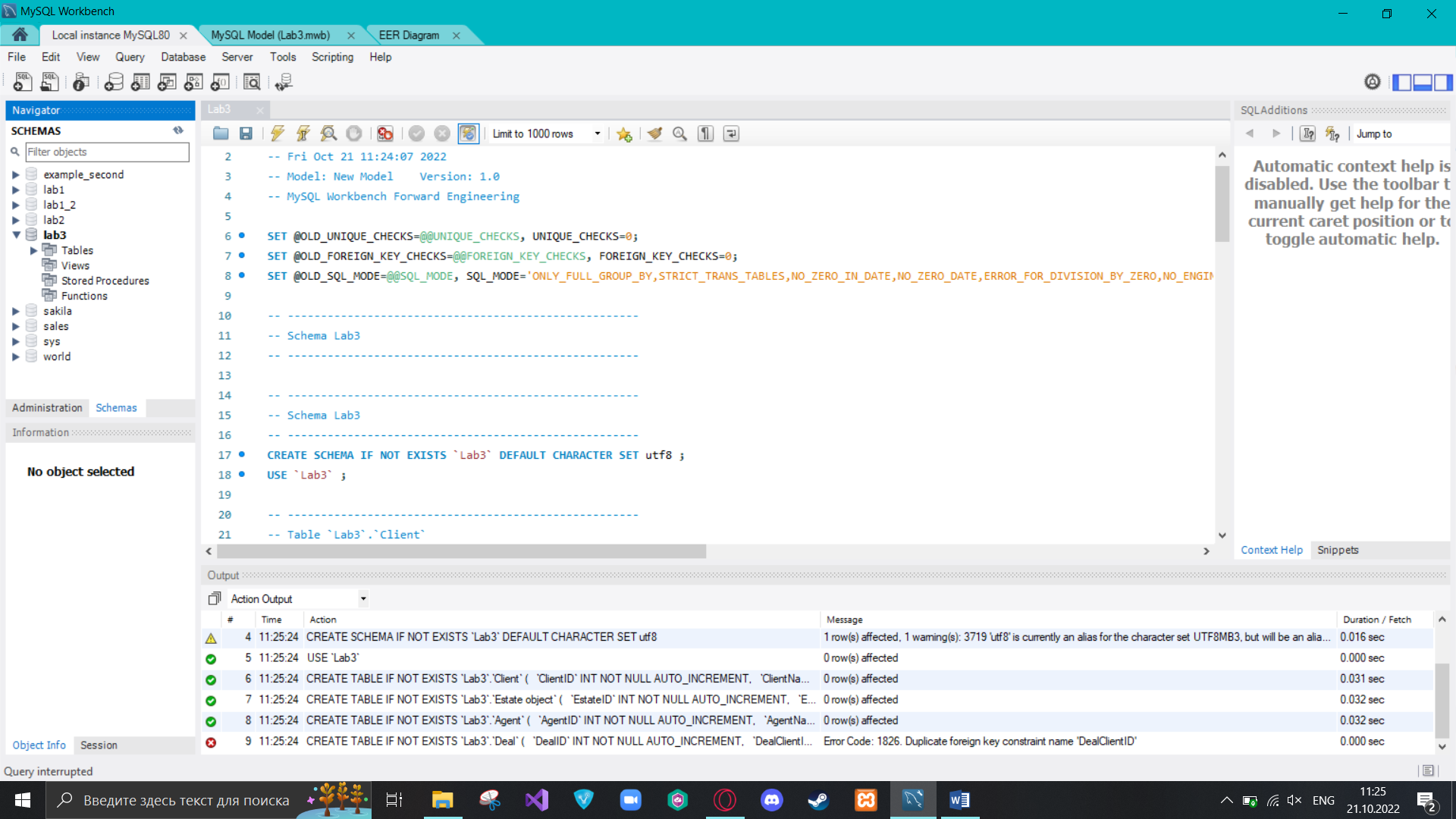


Рисунок 4 – Создание БД в среде MySQL Workbench

Далее таблицы были заполнены данными.

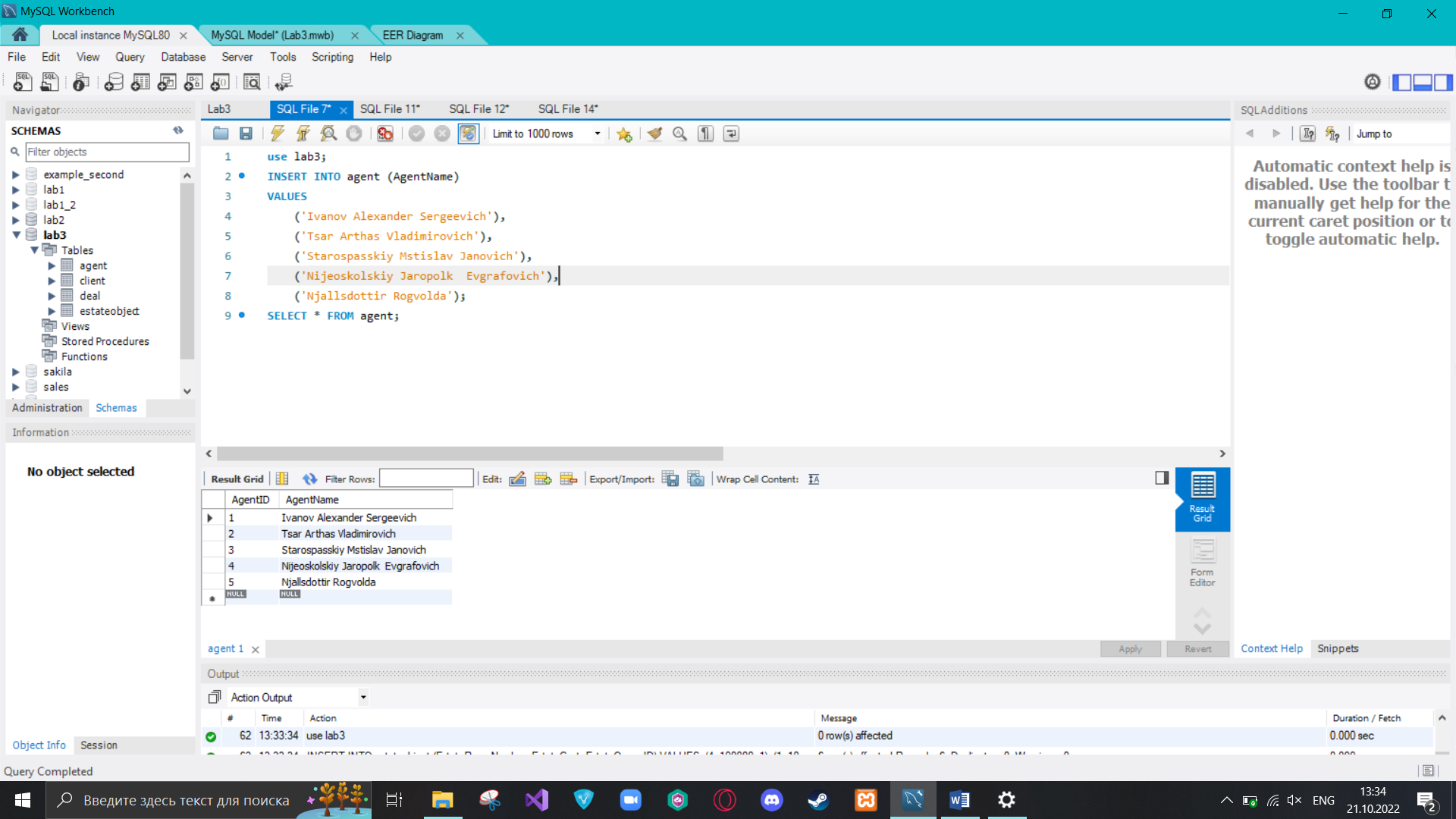


Рисунок 5 – Заполнение таблицы ‘agent’

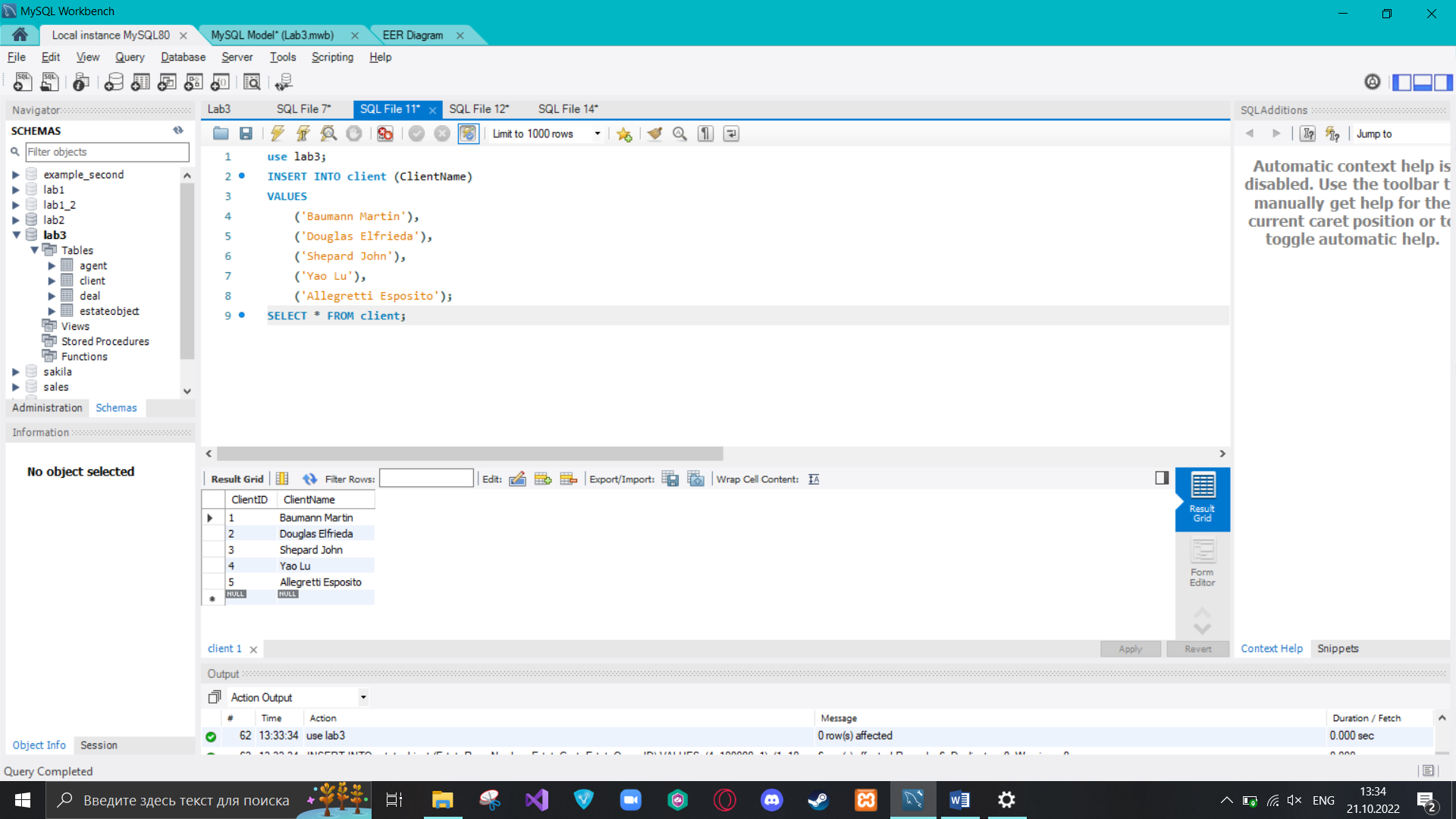


Рисунок 6 – Заполнение таблицы ‘client’

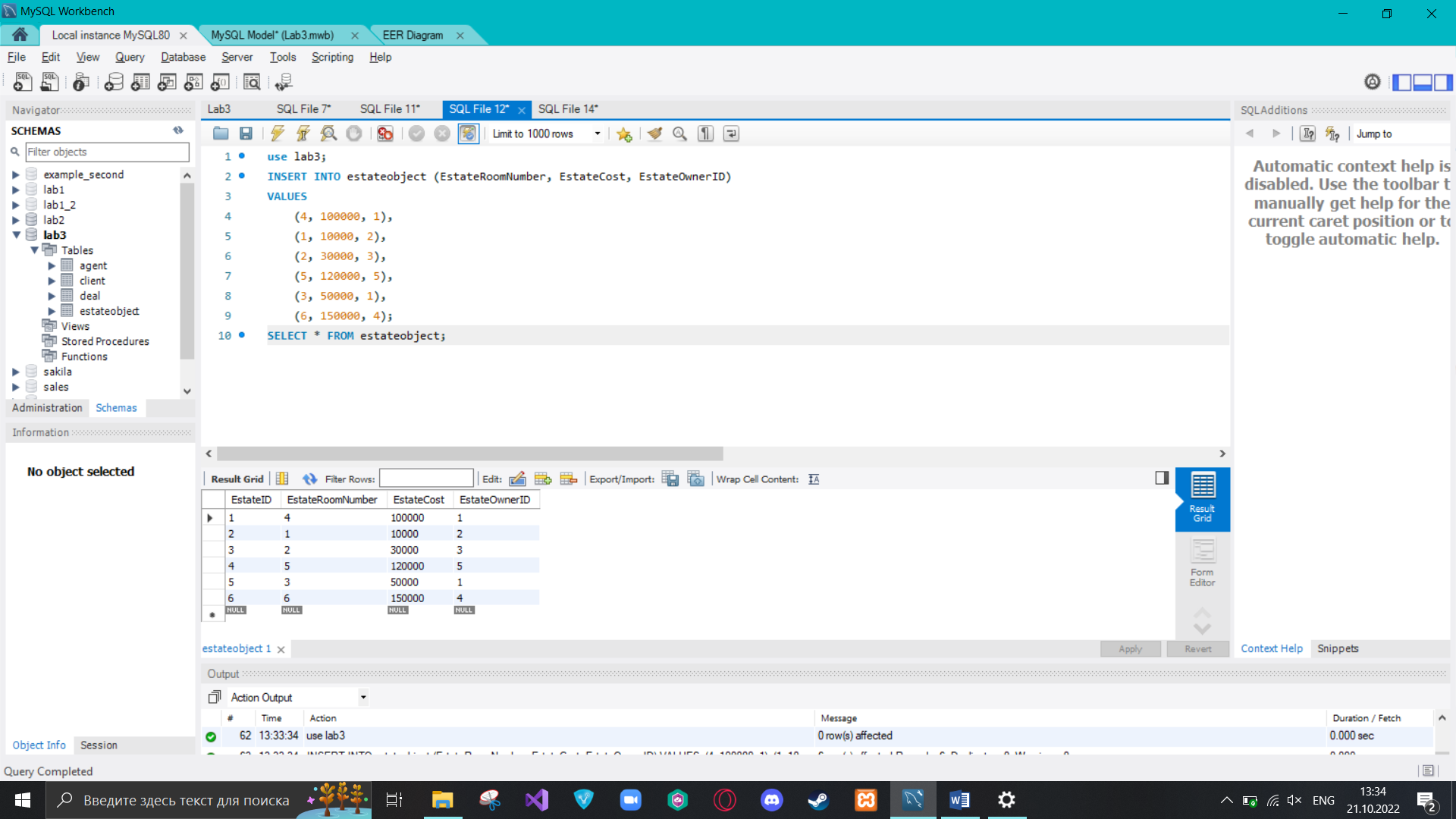


Рисунок 7 – Заполнение таблицы ‘estateobject’

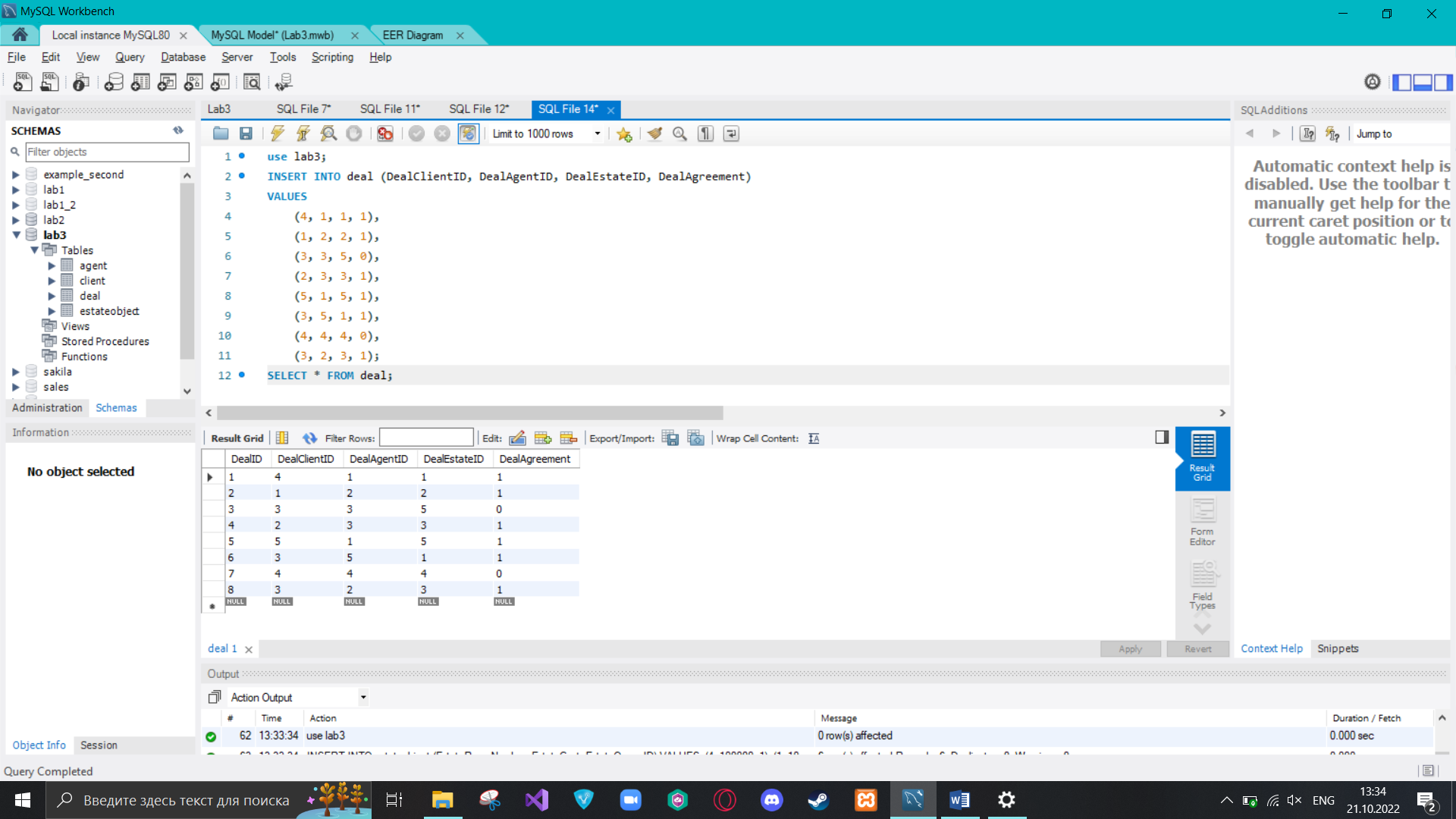


Рисунок 8 – Заполнение таблицы ‘deal’

**Вывод:**

Во время выполнения данной лабораторной работы я изучил разработку инфологической и даталогической моделей реляционной БД. Назначение инфологической модели состоит в том, чтобы формализовать объекты предметной области при помощи, например, ER-модели. Назначение даталогической модели состоит в том, чтобы спроектировать данные на основе особенностей используемой СУБД.

**Список используемых 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

-- Fri Oct 21 13:21:20 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;