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 *ИКБО-20-19, Сибгатулов М.Д.* _____
(подпись)

Преподаватель *Володина А.М.* _____
(подпись)

Отчет представлен «___» октября 2021г.

Москва 2021 г.

Введение

В качестве темы для создания базы данных было выбрано «Кузнечное дело». В процессе работы данная тема приняла направленность базы данных для кузнечной мастерской для упрощения работы. В виду этого была создана модель, ориентированная на полноценный онлайн магазин кузнечных изделий. В ней были описаны основные элементы, которые должны быть задействованы в подобном предприятии.

Логическая модель базы данных по теме «Кузнечное дело»

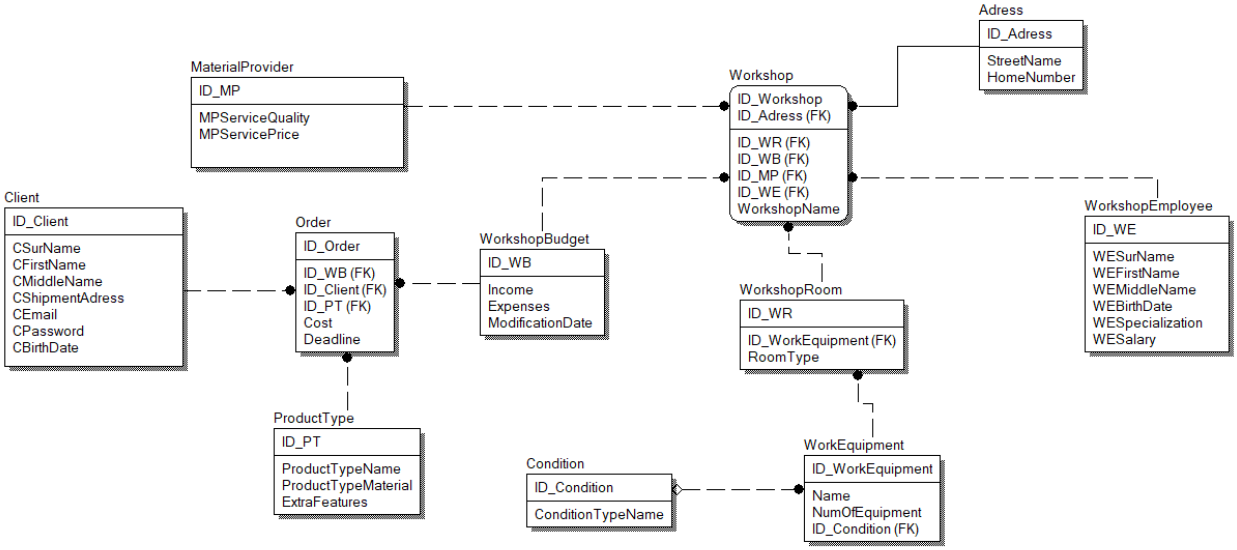


Рис. 1 – Логическая модель данных на тему «Кузнечное дело»

Описание

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

Client – содержит основные данные о клиенте, оформляющем заказ
MaterialProvider – содержит информацию о поставщиках
ProductType – содержит информацию об основных характеристиках заказа
Address – содержит адреса различных филиалов мастерской
WorkshopEmployee – содержит информацию о работниках мастерской
Condition – хранит информацию о состояния инструментария
WorkEquipment – содержит информацию об инструментарии
WorkshopRoom – характеризует рабочие области(помещения)
Order – содержит информацию о заказах
WorkshopBudget – используется для учёта бюджета
Workshop – является сборником основной информации о мастерской в целом

Физическая модель базы данных по теме «Кузнечное дело»

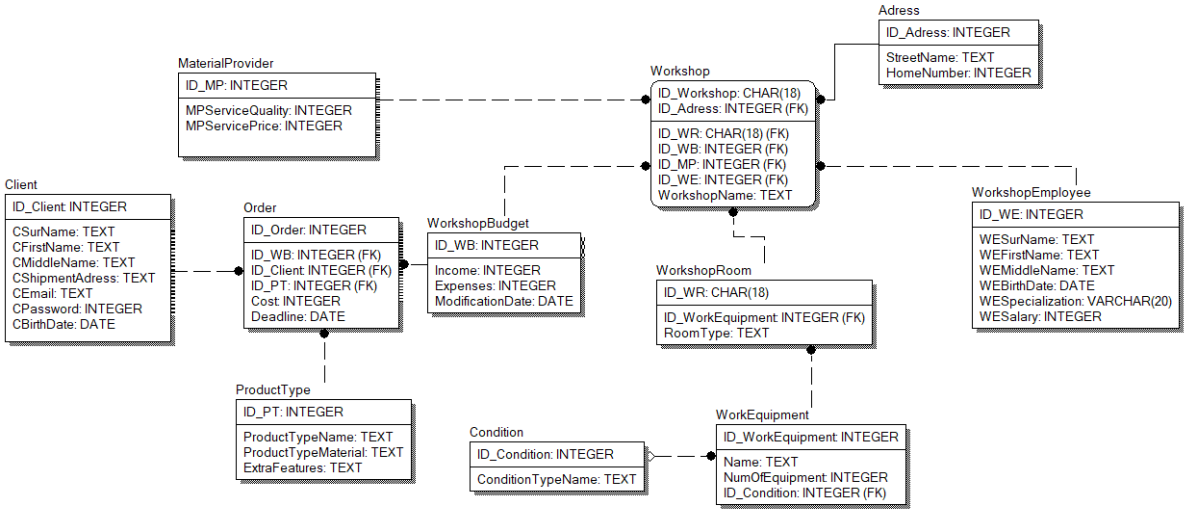


Рис. 2 – Физическая модель данных на тему «Кузнечное дело»

Создание своей базы данных в MySQL CommandLine

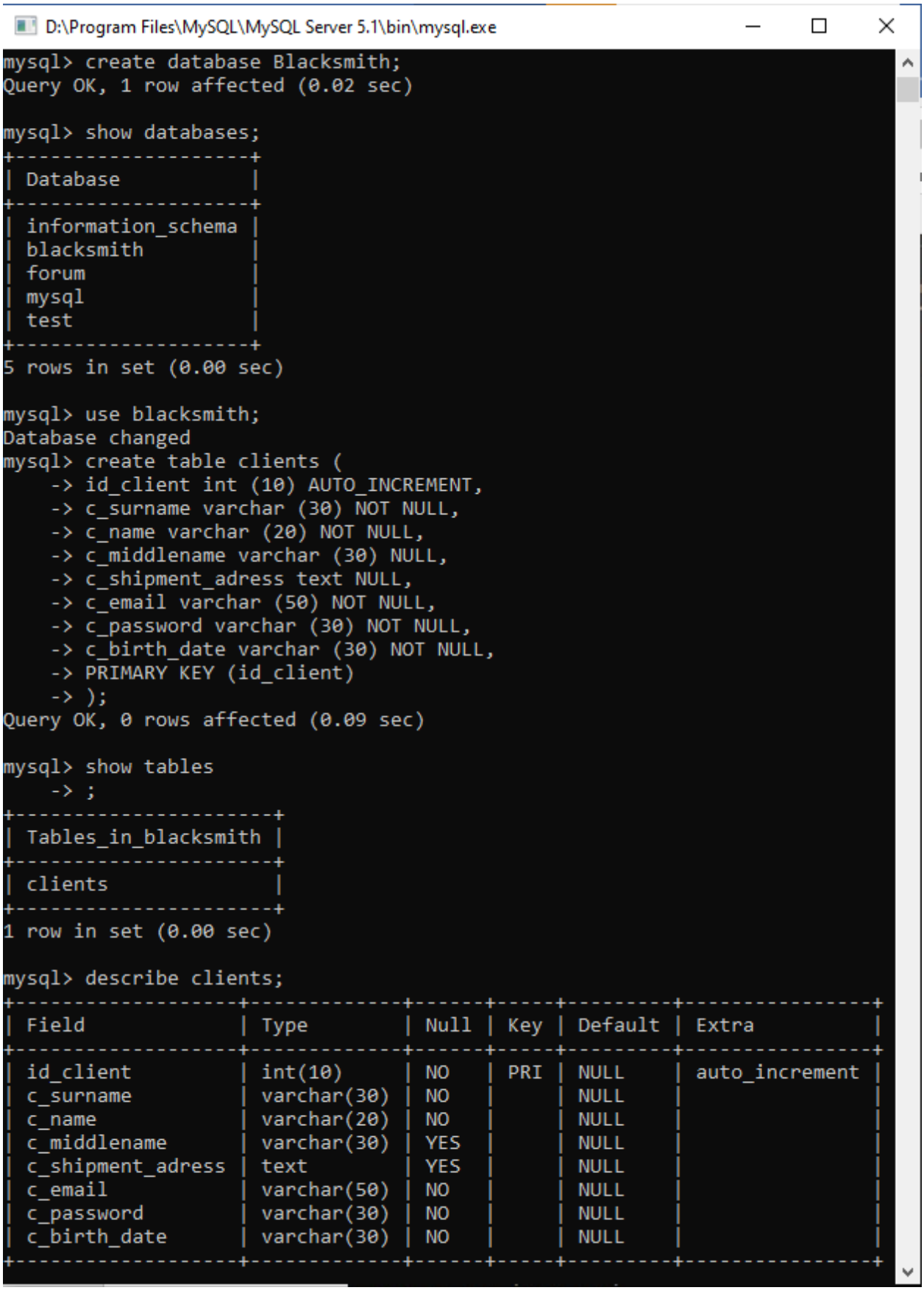


Рис. 3 Создание БД Blacksmith, таблицы clients и проверка

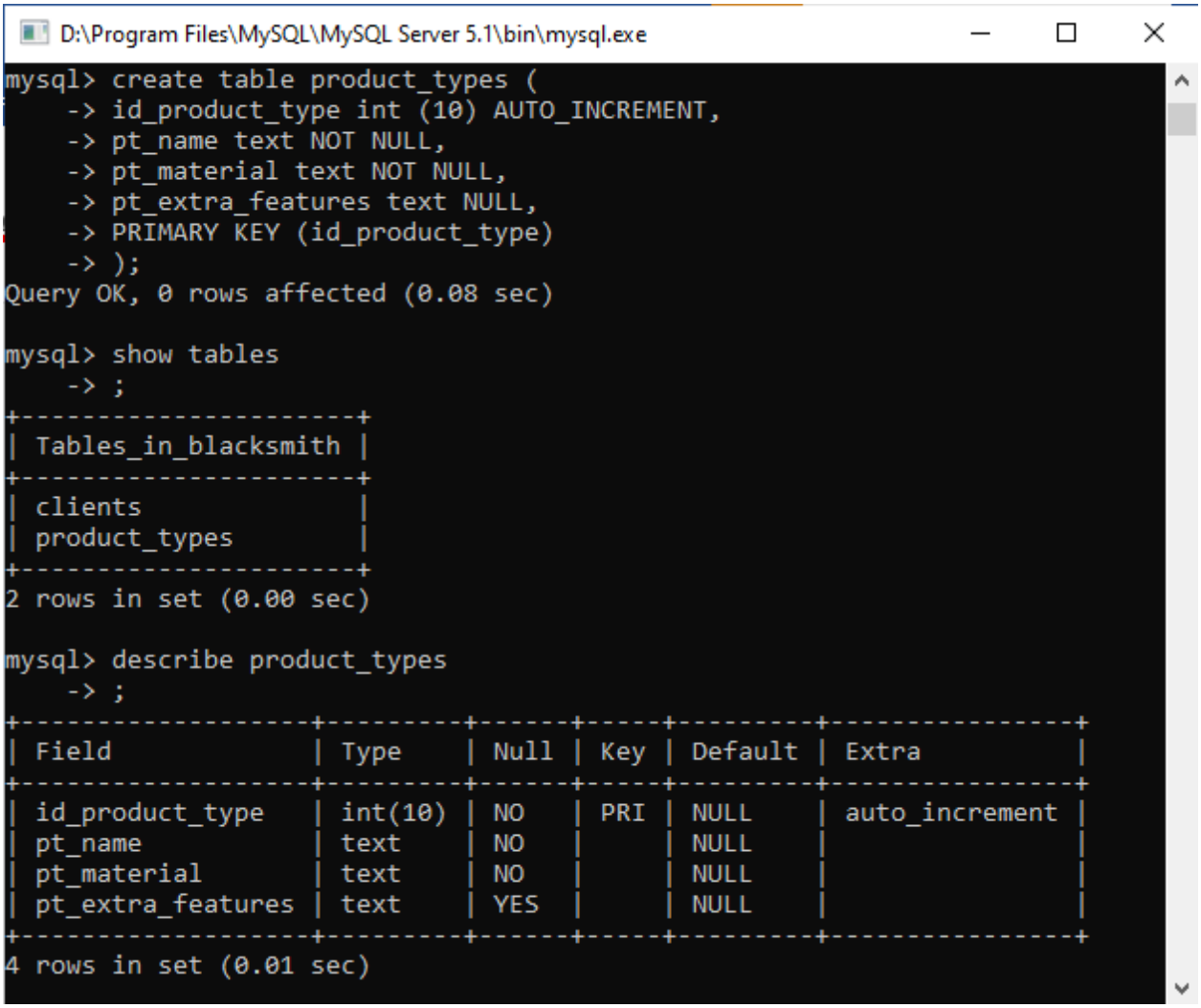


Рис. 4 Создание таблицы product_types и проверка

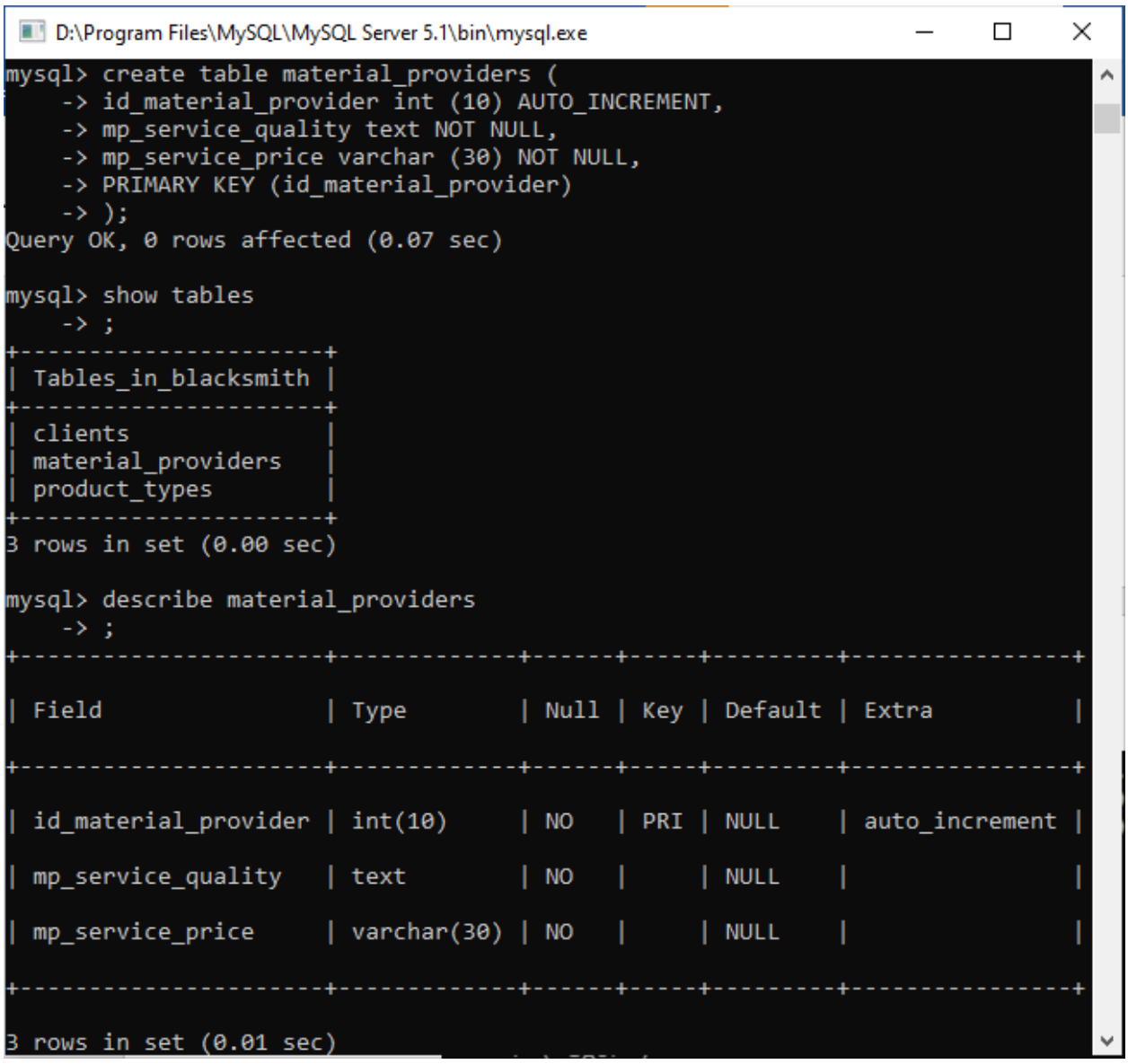


Рис. 5 Создание таблицы material_providers и проверка

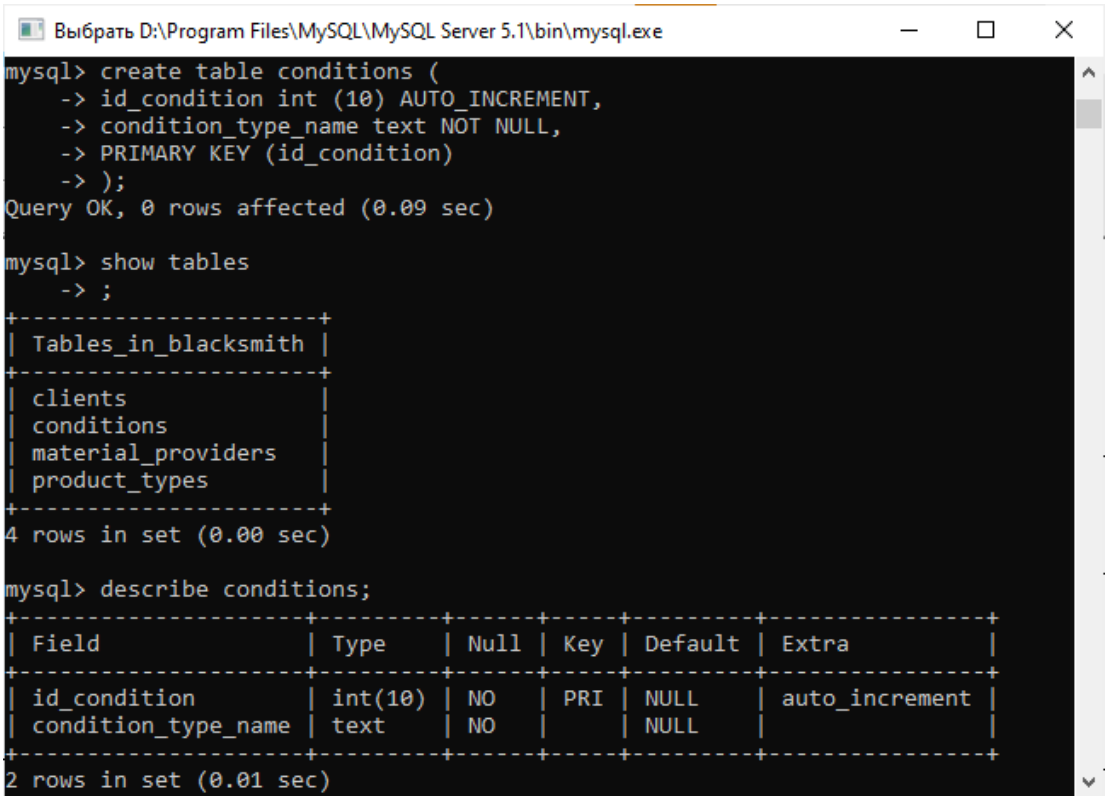


Рис. 6 Создание таблицы conditions и проверка

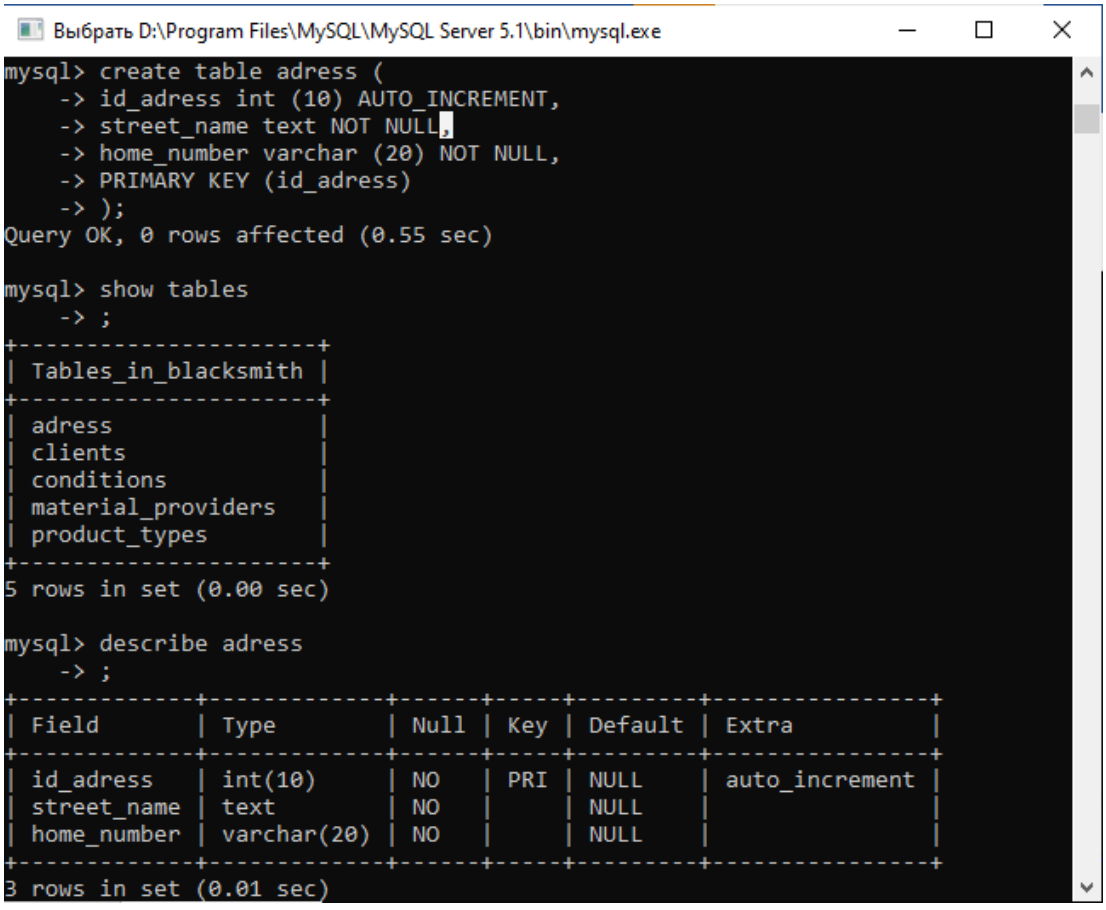


Рис. 7 Создание таблицы adress и проверка

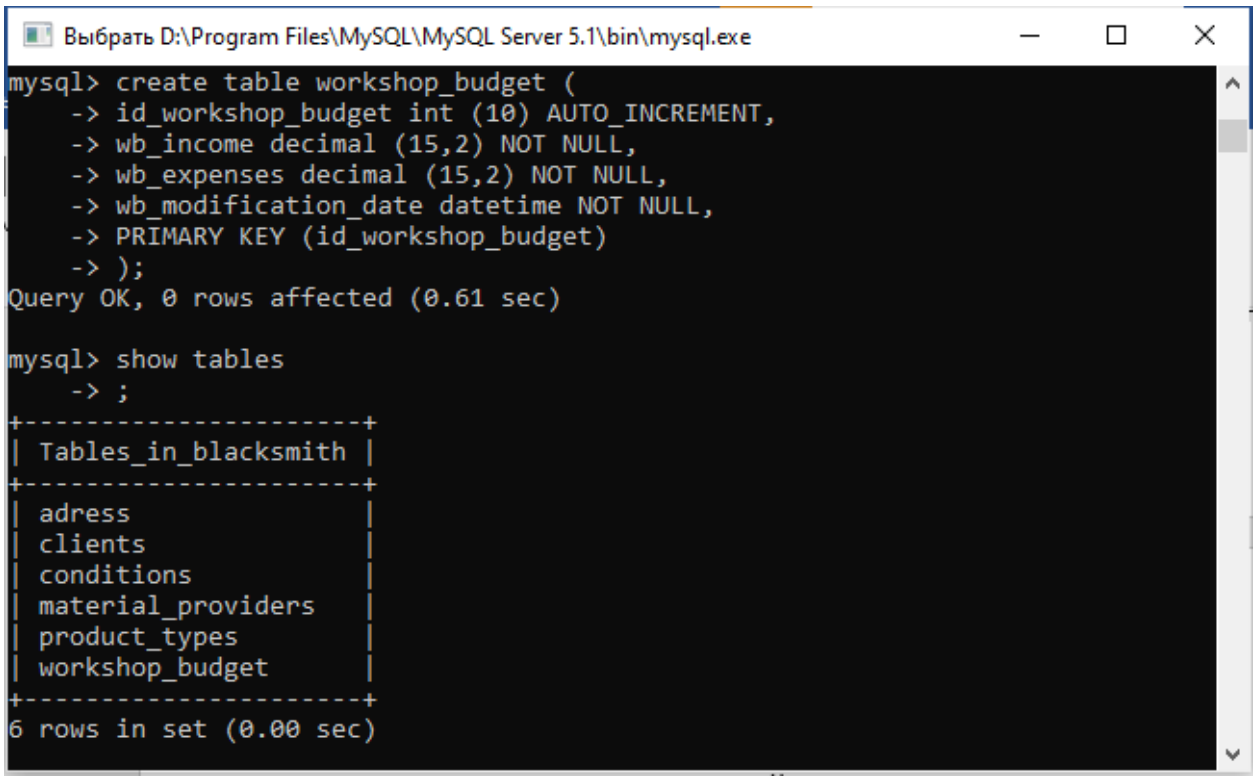


Рис. 8 Создание таблицы workshop_budget

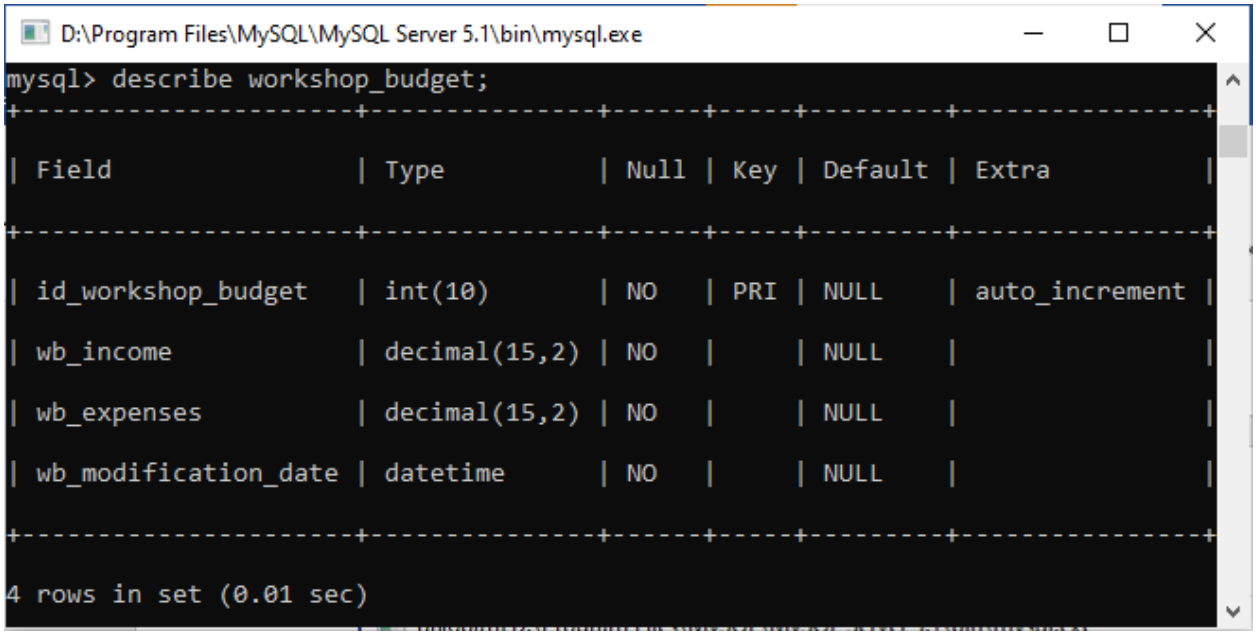


Рис. 9 Проверка правильности таблицы workshop_budget

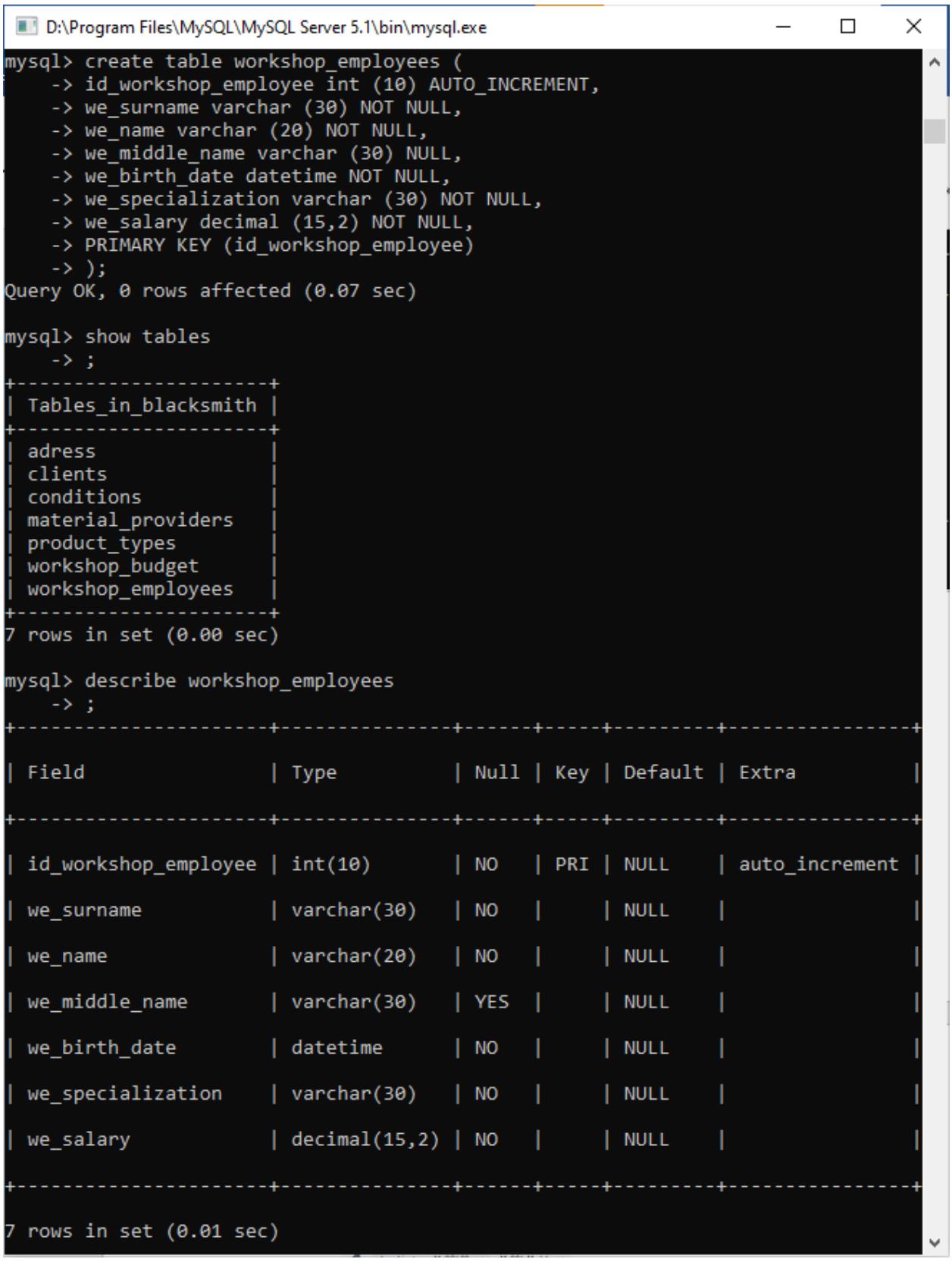


Рис. 10 Создание таблицы workshop_employees и проверка

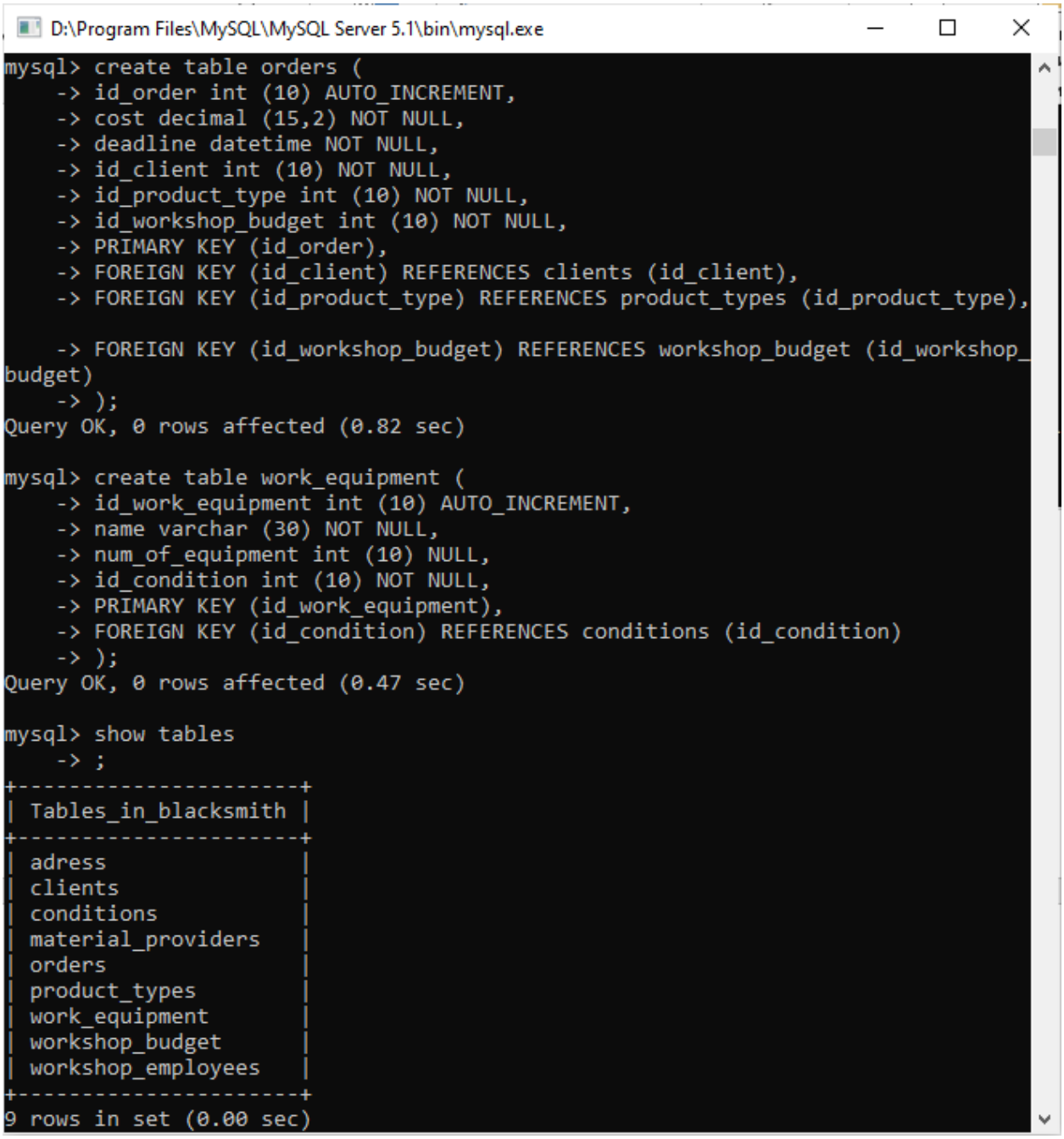


Рис. 11 Создание таблиц orders и work_equipment

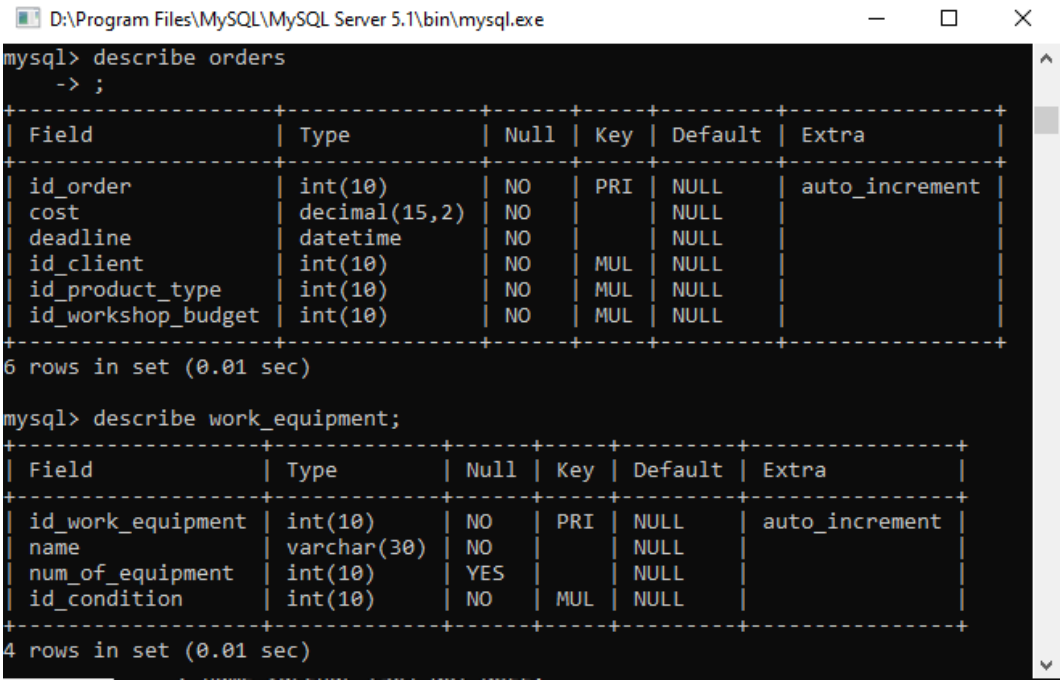


Рис. 12 Проверка правильности таблиц orders и work_equipment

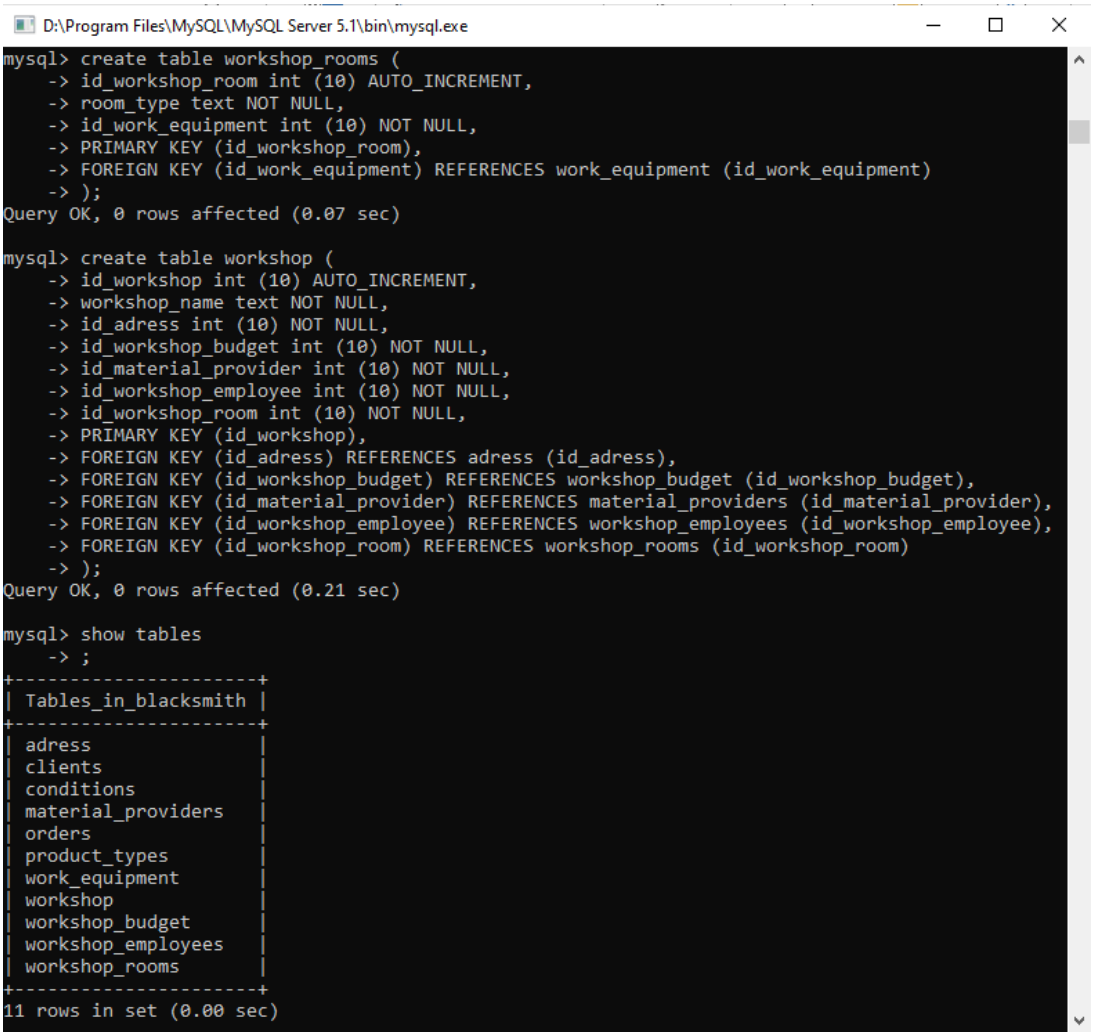


Рис. 13 Создание таблиц workshop_rooms и workshop

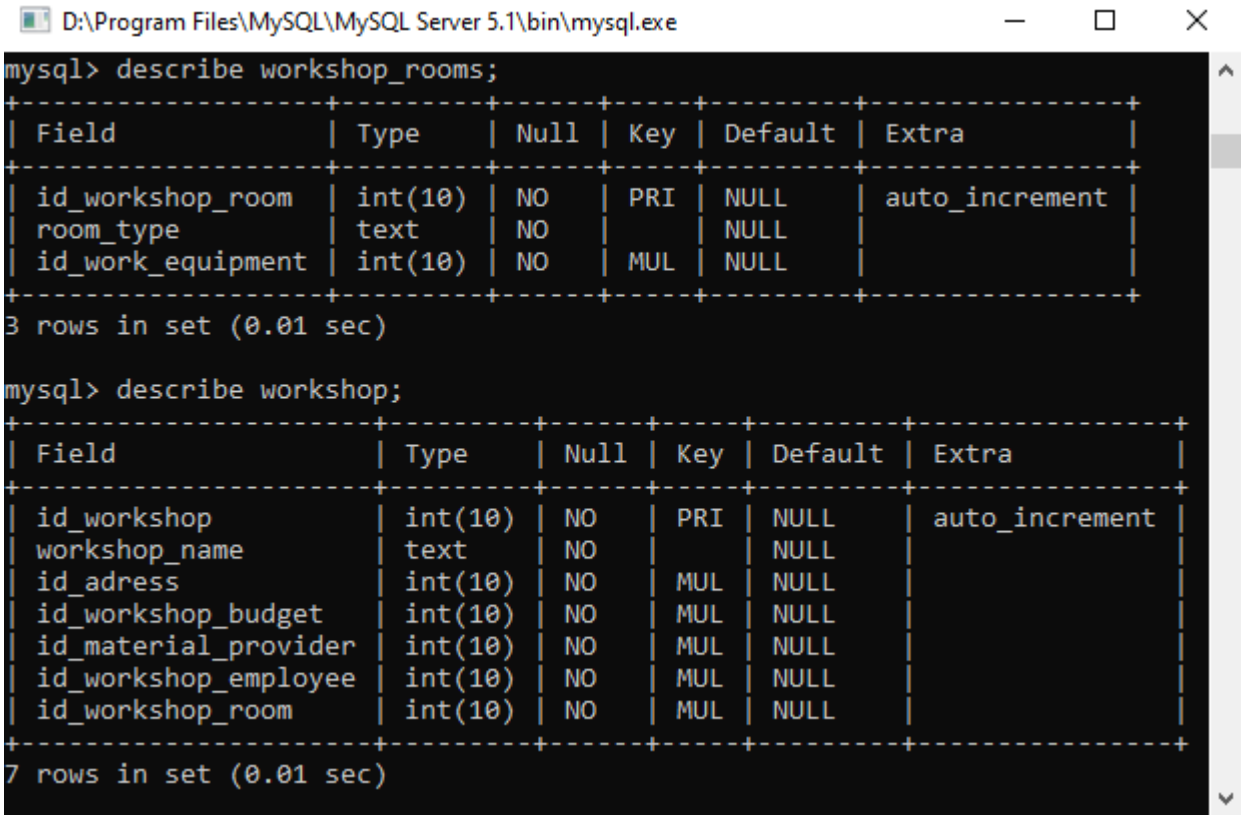


Рис. 14 Проверка правильности таблиц workshop_rooms и workshop

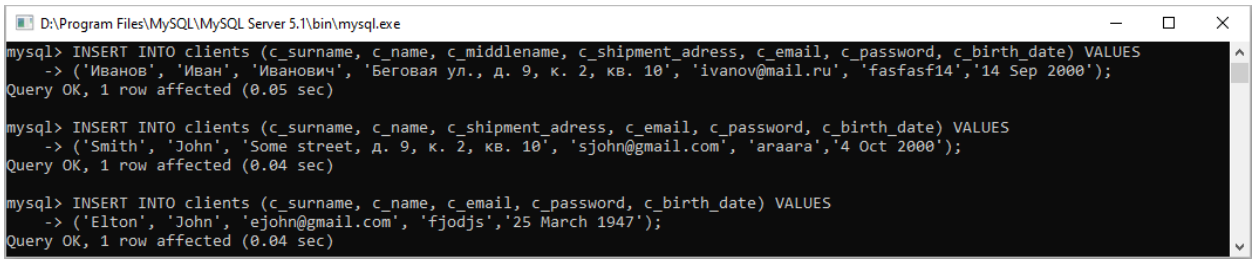


Рис. 15 Заполнение таблицы clients

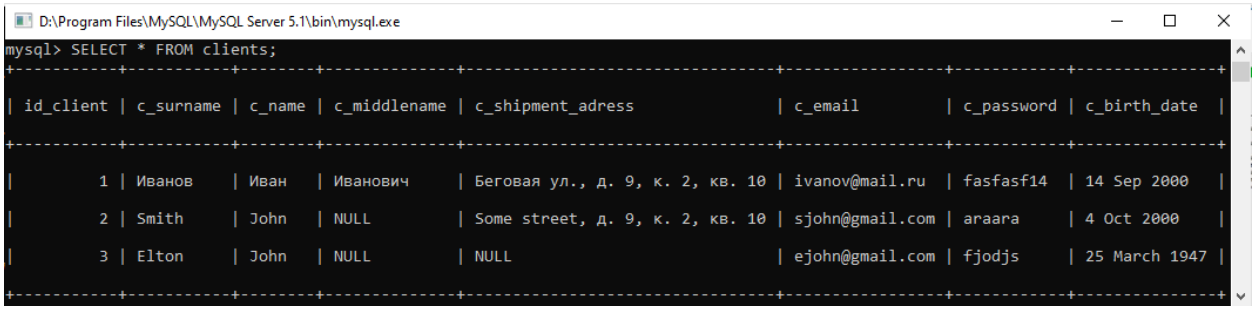


Рис. 16 Вывод элементов таблицы clients

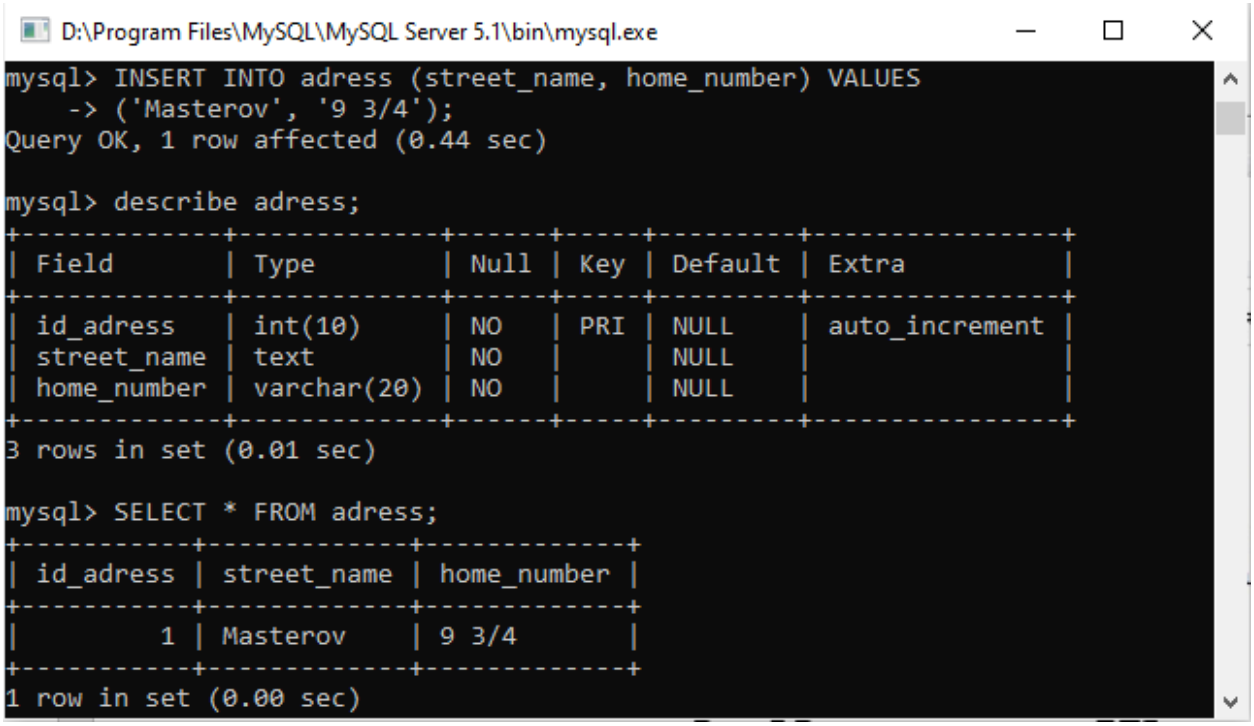


Рис. 17 Заполнение и вывод элементов таблицы adress

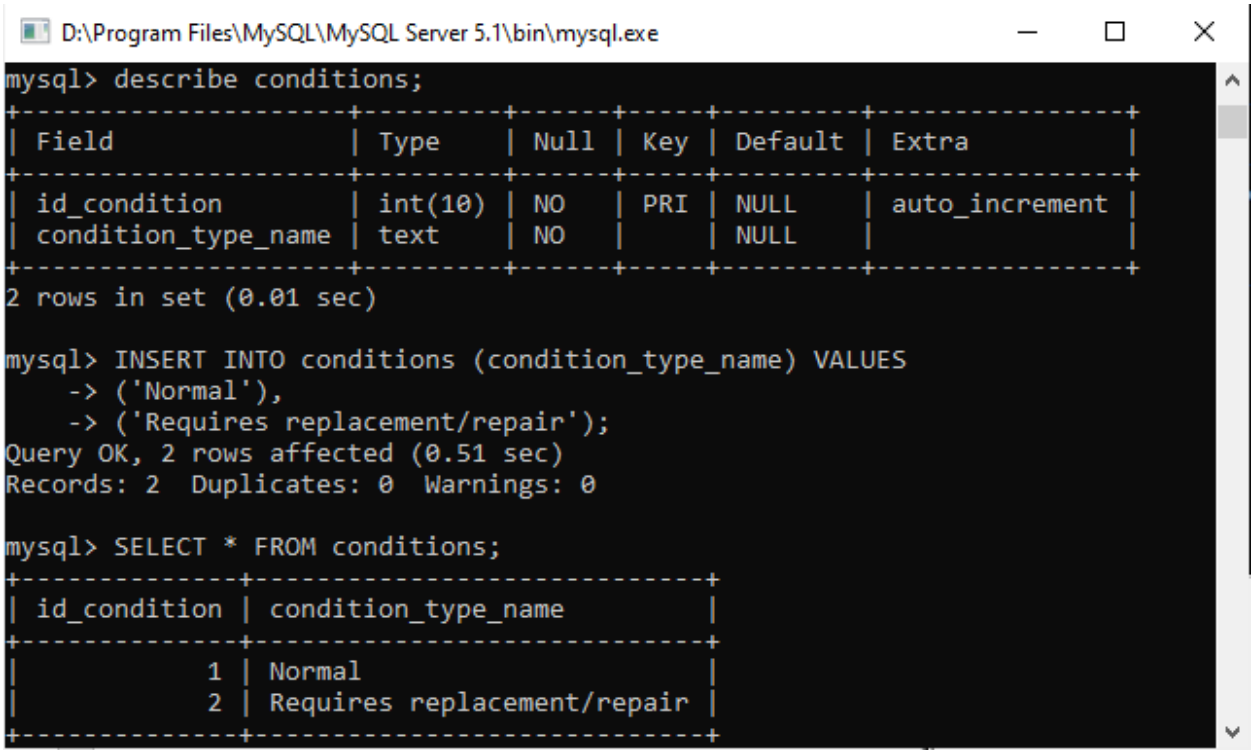


Рис. 18 Заполнение и вывод элементов таблицы conditions

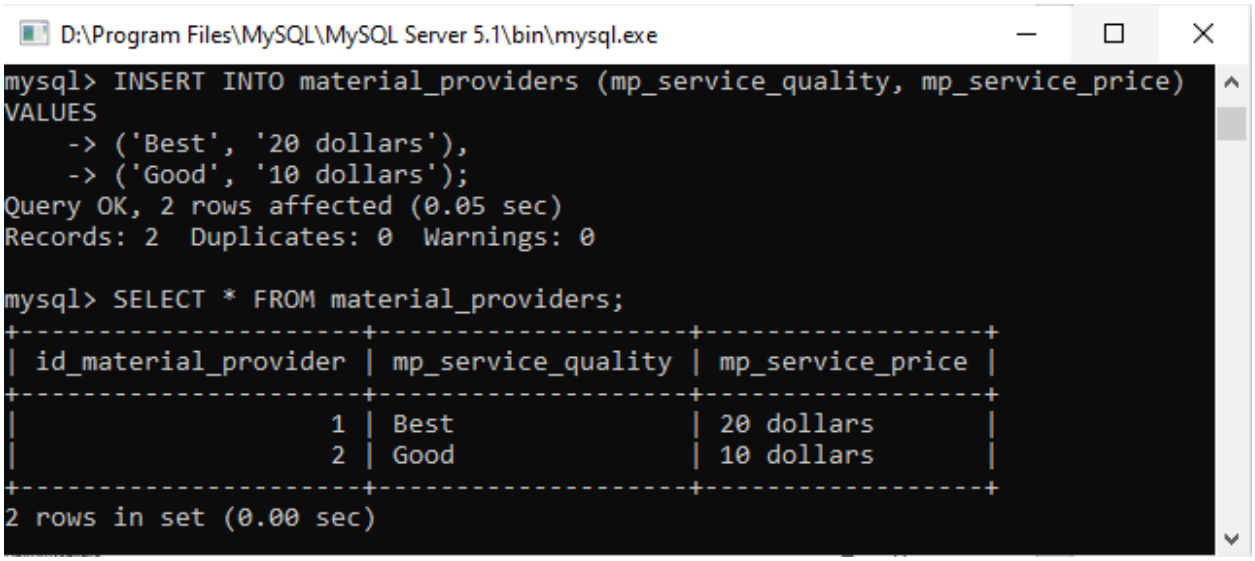


Рис. 19 Заполнение и вывод элементов таблицы material_providers

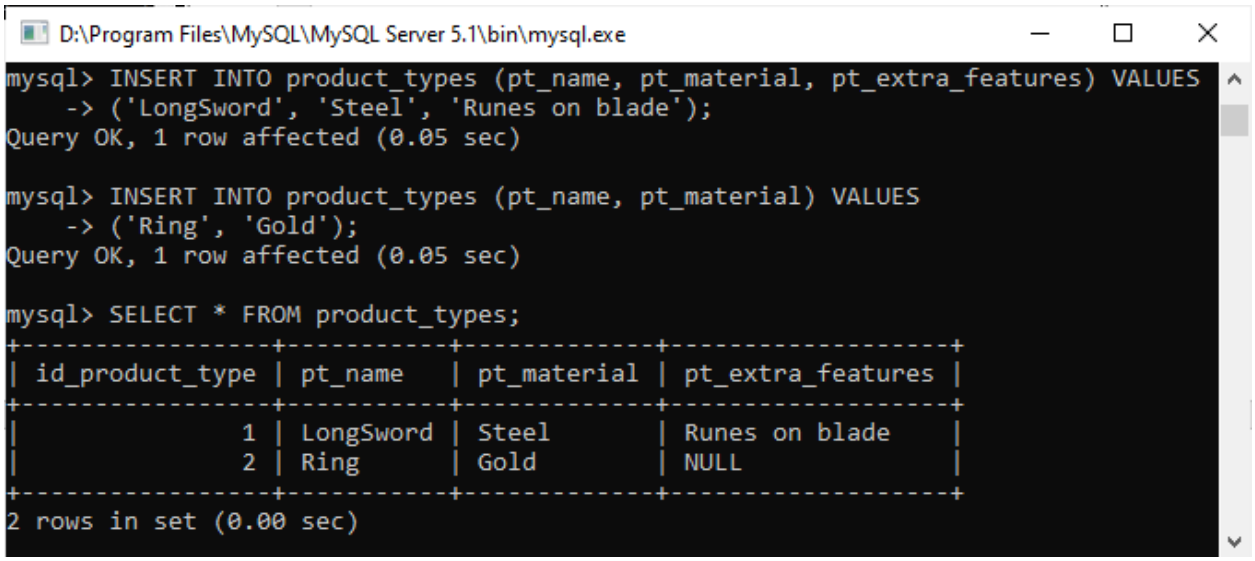


Рис. 20 Заполнение и вывод элементов таблицы product_types

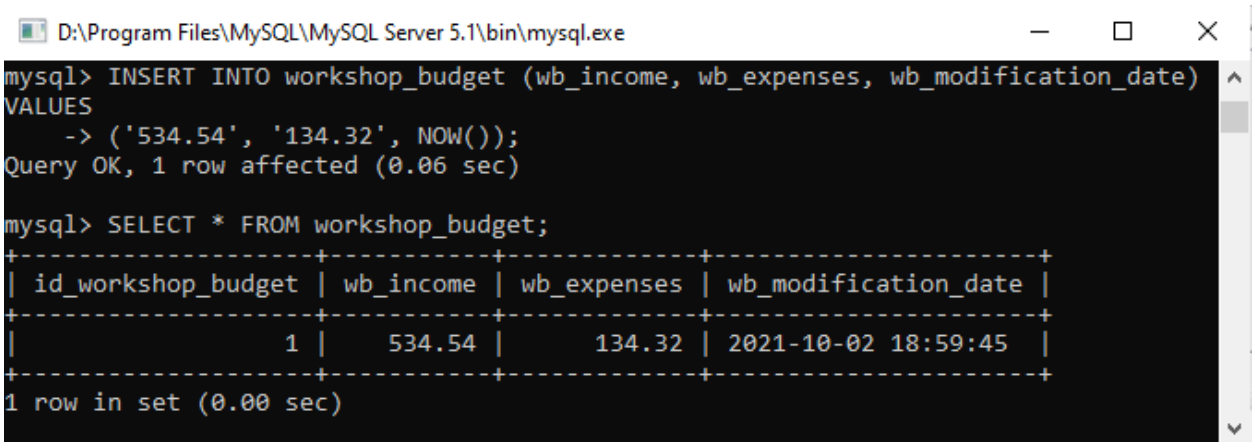


Рис. 21 Заполнение и вывод элементов таблицы workshop_budget

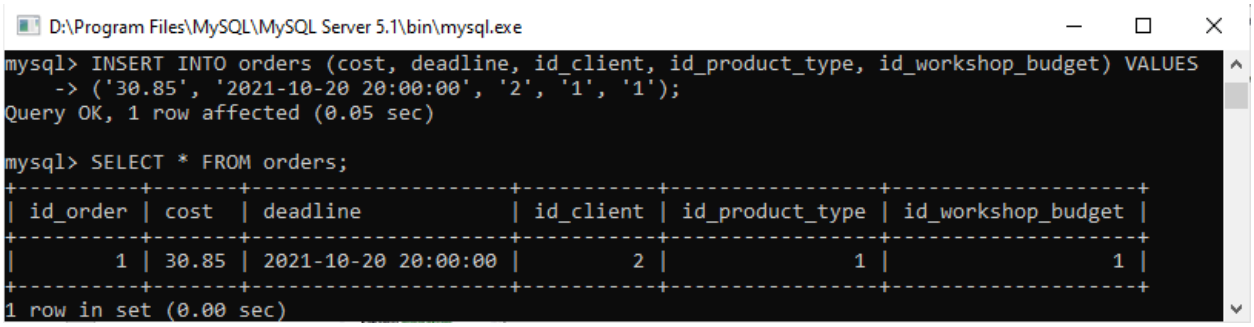


Рис. 22 Заполнение и вывод элементов таблицы orders

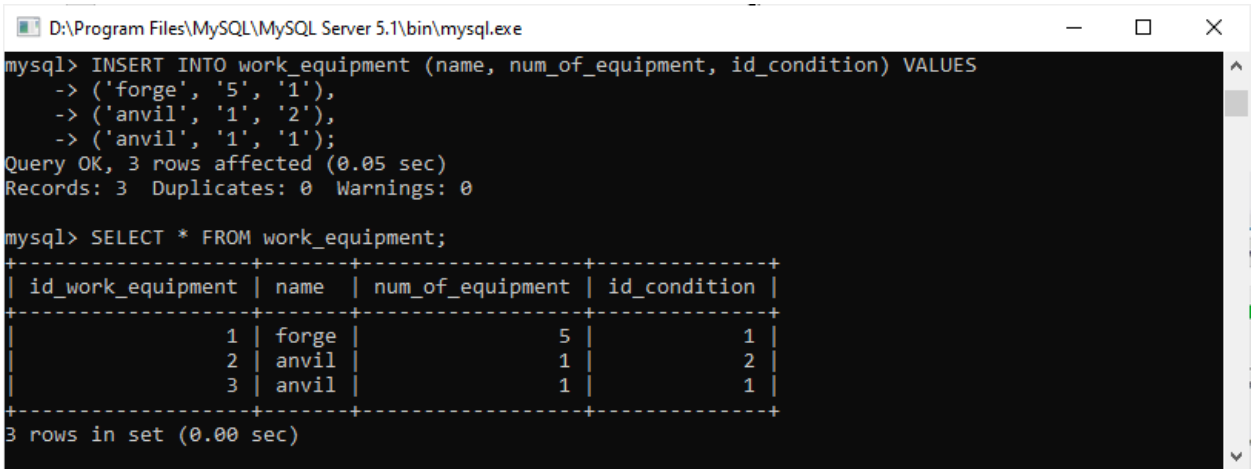


Рис. 23 Заполнение и вывод элементов таблицы work_equipment

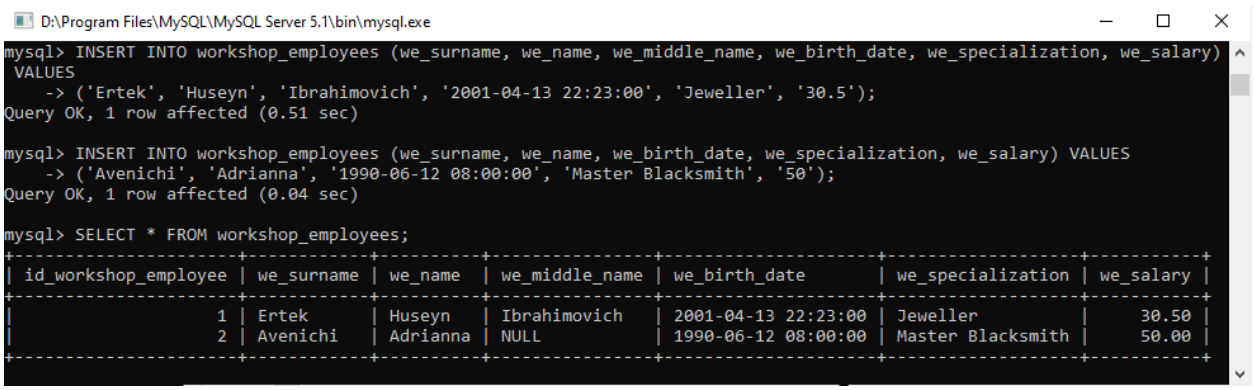


Рис. 24 Заполнение и вывод элементов таблицы workshop_employees

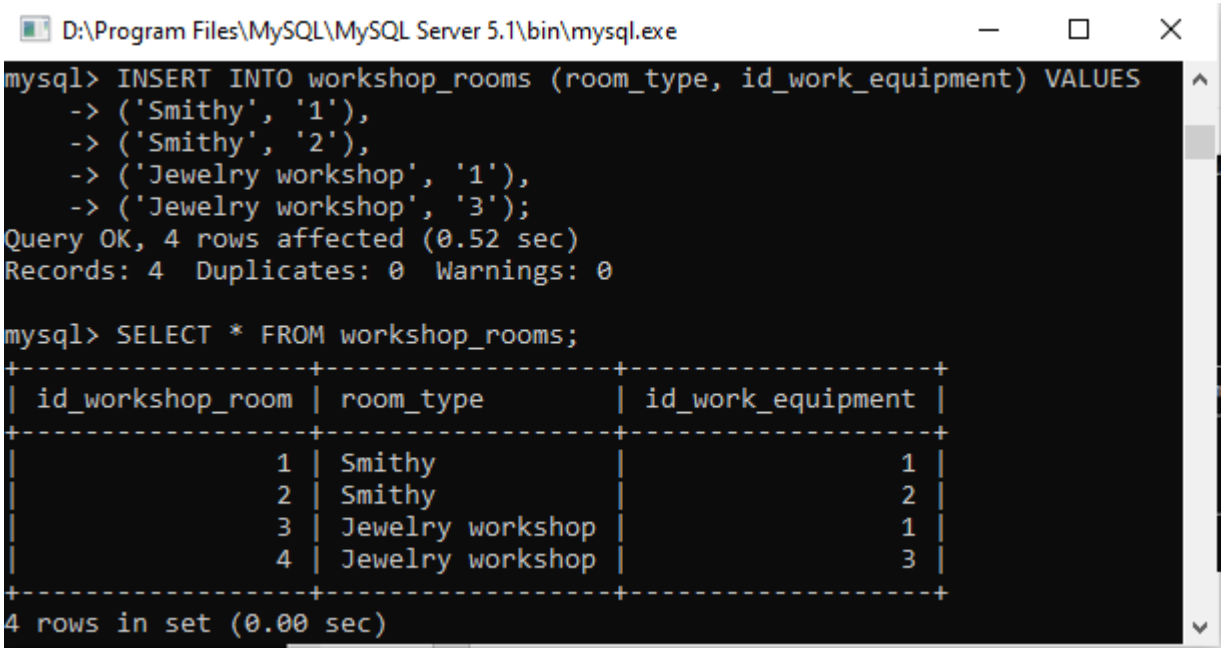


Рис. 25 Заполнение и вывод элементов таблицы workshop_rooms

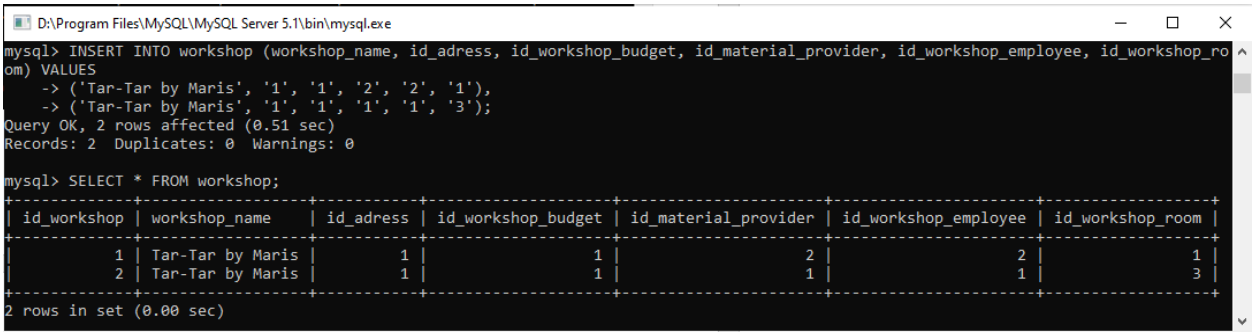


Рис. 26 Заполнение и вывод элементов таблицы workshop

Работа с запросами в базе данных Blacksmith

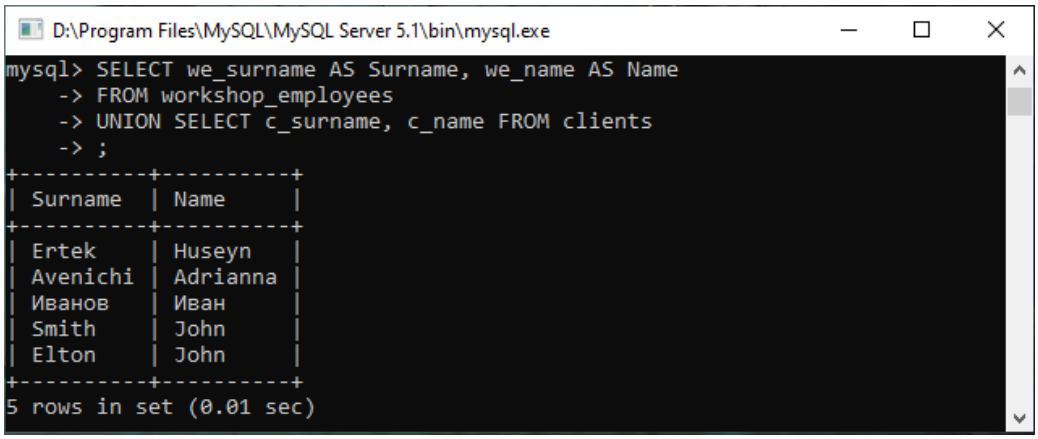


Рис. 27 – Объединение с использованием UNION

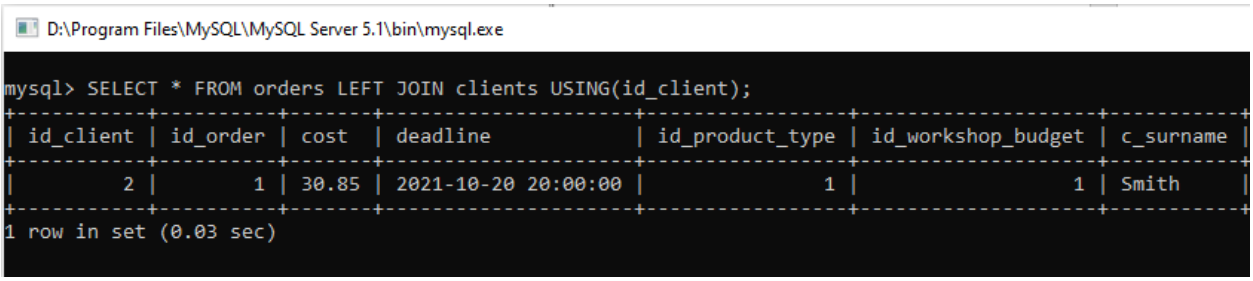


Рис. 28 – Выборка с использованием LEFT JOIN часть 1

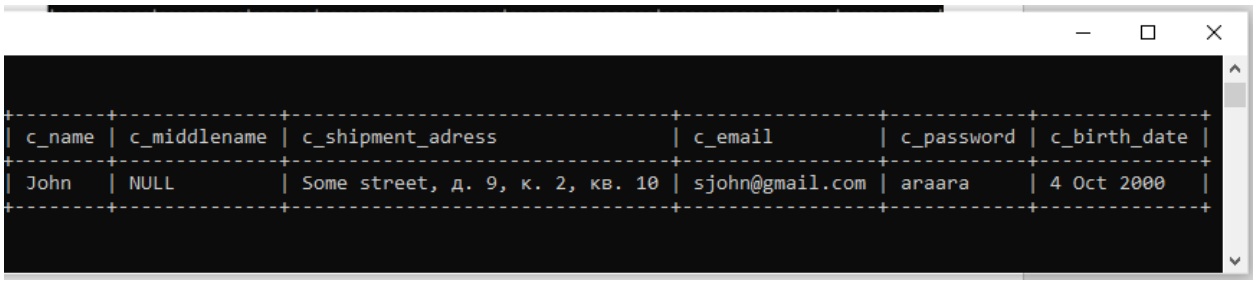


Рис. 29 – Выборка с использованием LEFT JOIN часть 2

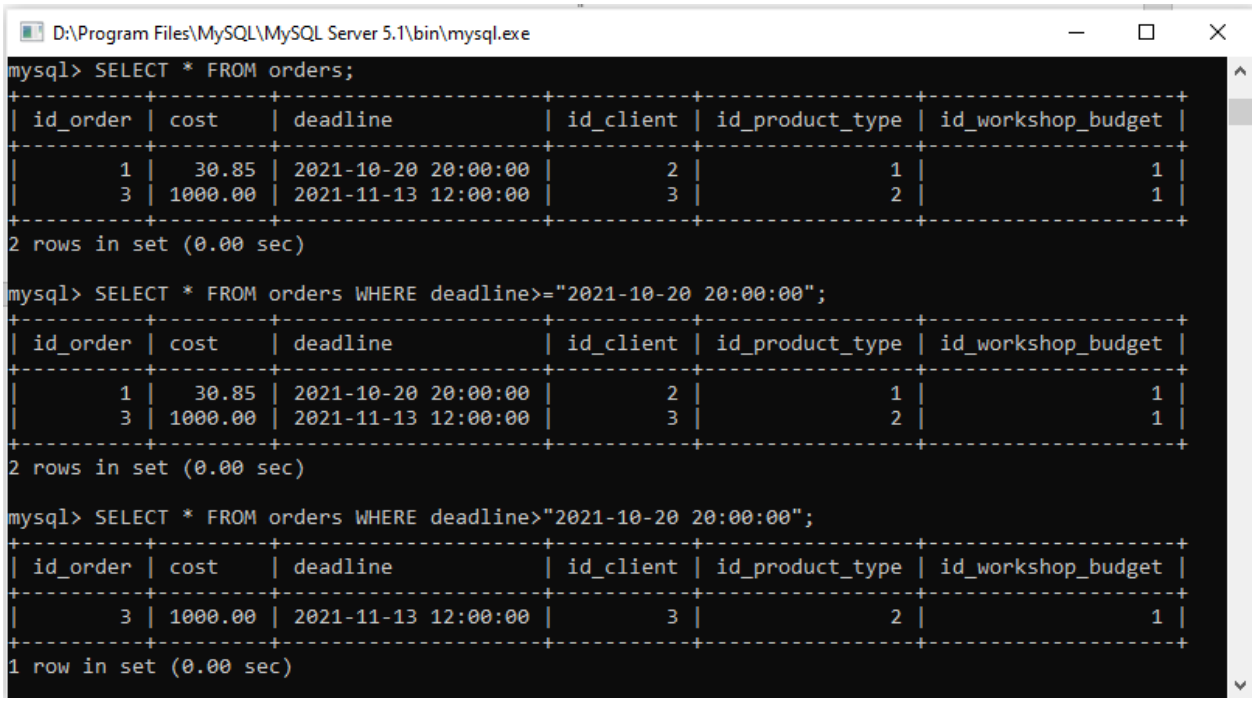


Рис. 30 Выборка данных по параметру deadline

Построение модели с помощью оболочки MySQL Workbench

Мною была построена модель в СУБД Workbench(Рисунок 31)

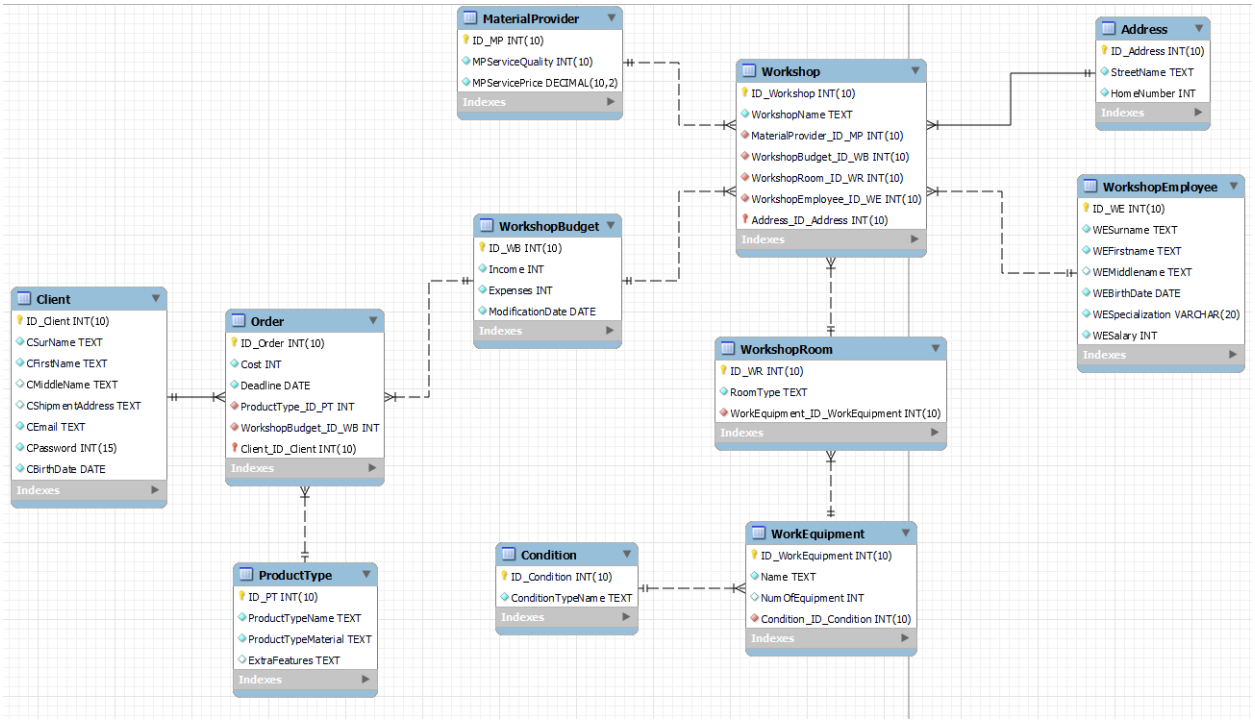


Рисунок 31 – Построенная модель

Перенос Базы Данных на другой сервер

Использовав встроенные инструменты MYSQL Workbench, был получен MySql скрипт

-- MySQL Script generated by MySQL Workbench

-- Tue Nov 9 17:20:15 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 blacksmith
-- -----------------------------------------------------


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


-- -----------------------------------------------------
-- Table `blacksmith`.`Client`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`Client` (
  `ID_Client` INT(10) AUTO_INCREMENT,
  `CSurName` TEXT NOT NULL,
  `CFirstName` TEXT NOT NULL,
  `CMiddleName` TEXT NULL,
  `CShipmentAddress` TEXT NULL,
  `CEmail` TEXT NOT NULL,
  `CPassword` INT(15) NOT NULL,
  `CBirthDate` DATE NOT NULL,
  PRIMARY KEY (`ID_Client`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`MaterialProvider`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`MaterialProvider` (
  `ID_MP` INT(10) AUTO_INCREMENT,
  `MPServiceQuality` INT(10) NOT NULL,
  `MPServicePrice` DECIMAL(10,2) NOT NULL,
```

```
  PRIMARY KEY (`ID_MP`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`ProductType`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`ProductType` (
  `ID_PT` INT(10) AUTO_INCREMENT,
  `ProductTypeName` TEXT NOT NULL,
  `ProductTypeMaterial` TEXT NOT NULL,
  `ExtraFeatures` TEXT NULL,
  PRIMARY KEY (`ID_PT`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`WorkshopBudget`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`WorkshopBudget` (
  `ID_WB` INT(10) AUTO_INCREMENT,
  `Income` INT NOT NULL,
  `Expenses` INT NOT NULL,
  `ModificationDate` DATE NOT NULL,
  PRIMARY KEY (`ID_WB`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`Order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`Order` (
```

```
  `ID_Order` INT(10) AUTO_INCREMENT,
  `Cost` INT NOT NULL,
  `Deadline` DATE NOT NULL,
  `Client_ID_Client` INT NOT NULL,
  `ProductType_ID_PT` INT NOT NULL,
  `WorkshopBudget_ID_WB` INT NOT NULL,
  PRIMARY KEY (`ID_Order, Client_ID_Client`),
  INDEX `fk_Order_Client_idx` (`Client_ID_Client` ASC) VISIBLE,
  INDEX `fk_Order_ProductType1_idx` (`ProductType_ID_PT` ASC) VISIBLE,
  INDEX `fk_Order_WorkshopBudget1_idx` (`WorkshopBudget_ID_WB` ASC) VISIBLE,
  CONSTRAINT `fk_Order_Client`
    FOREIGN KEY (`Client_ID_Client`)
    REFERENCES `blacksmith`.`Client` (`ID_Client`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Order_ProductType1`
    FOREIGN KEY (`ProductType_ID_PT`)
    REFERENCES `blacksmith`.`ProductType` (`ID_PT`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Order_WorkshopBudget1`
    FOREIGN KEY (`WorkshopBudget_ID_WB`)
    REFERENCES `blacksmith`.`WorkshopBudget` (`ID_WB`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`Condition`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`Condition` (
```

```
  `ID_Condition` INT(10) AUTO_INCREMENT,
  `ConditionTypeName` TEXT NOT NULL,
  PRIMARY KEY (`ID_Condition`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`WorkEquipment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`WorkEquipment` (
  `ID_WorkEquipment` INT(10) AUTO_INCREMENT,
  `Name` TEXT NOT NULL,
  `NumOfEquipment` INT NULL,
  `Condition_ID_Condition` INT(10) NOT NULL,
  PRIMARY KEY (`ID_WorkEquipment`),
  INDEX `fk_WorkEquipment_Condition1_idx` (`Condition_ID_Condition` ASC) VISIBLE,
  CONSTRAINT `fk_WorkEquipment_Condition1`
    FOREIGN KEY (`Condition_ID_Condition`)
    REFERENCES `blacksmith`.`Condition` (`ID_Condition`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`WorkshopRoom`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`WorkshopRoom` (
  `ID_WR` INT(10) AUTO_INCREMENT,
  `RoomType` TEXT NOT NULL,
  `WorkEquipment_ID_WorkEquipment` INT(10) NOT NULL,
  PRIMARY KEY (`ID_WR`),
```

```
  INDEX `fk_WorkshopRoom_WorkEquipment1_idx` (`WorkEquipment_ID_WorkEquipment` ASC) VISIBLE,
  CONSTRAINT `fk_WorkshopRoom_WorkEquipment1`
    FOREIGN KEY (`WorkEquipment_ID_WorkEquipment`)
    REFERENCES `blacksmith`.`WorkEquipment` (`ID_WorkEquipment`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`WorkshopEmployee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`WorkshopEmployee` (
  `ID_WE` INT(10) AUTO_INCREMENT,
  `WESurname` TEXT NOT NULL,
  `WEFirstname` TEXT NOT NULL,
  `WEMiddlename` TEXT NULL,
  `WEBirthDate` DATE NOT NULL,
  `WESpecialization` VARCHAR(20) NOT NULL,
  `WESalary` INT NOT NULL,
  PRIMARY KEY (`ID_WE`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `blacksmith`.`Address`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`Address` (
  `ID_Address` INT(10) AUTO_INCREMENT,
  `StreetName` TEXT NOT NULL,
  `HomeNumber` INT NOT NULL,
```

```
  PRIMARY KEY (`ID_Address`))
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `blacksmith`.`Workshop`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `blacksmith`.`Workshop` (
  `ID_Workshop` INT(10) AUTO_INCREMENT,
  `WorkshopName` TEXT NOT NULL,
  `MaterialProvider_ID_MP` INT(10) NOT NULL,
  `WorkshopBudget_ID_WB` INT(10) NOT NULL,
  `WorkshopRoom_ID_WR` INT(10) NOT NULL,
  `WorkshopEmployee_ID_WE` INT(10) NOT NULL,
  `Address_ID_Address` INT(10) NOT NULL,
  PRIMARY KEY (`ID_Workshop`, `Address_ID_Address`),
  INDEX `fk_Workshop_MaterialProvider1_idx` (`MaterialProvider_ID_MP` ASC) VISIBLE,
  INDEX `fk_Workshop_WorkshopBudget1_idx` (`WorkshopBudget_ID_WB` ASC) VISIBLE,
  INDEX `fk_Workshop_WorkshopRoom1_idx` (`WorkshopRoom_ID_WR` ASC) VISIBLE,
  INDEX `fk_Workshop_WorkshopEmployee1_idx` (`WorkshopEmployee_ID_WE` ASC)
VISIBLE,
  INDEX `fk_Workshop_Address1_idx` (`Address_ID_Address` ASC) VISIBLE,
  CONSTRAINT `fk_Workshop_MaterialProvider1`
    FOREIGN KEY (`MaterialProvider_ID_MP`)
    REFERENCES `blacksmith`.`MaterialProvider` (`ID_MP`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Workshop_WorkshopBudget1`
    FOREIGN KEY (`WorkshopBudget_ID_WB`)
    REFERENCES `blacksmith`.`WorkshopBudget` (`ID_WB`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

```
  CONSTRAINT `fk_Workshop_WorkshopRoom1`
    FOREIGN KEY (`WorkshopRoom_ID_WR`)
    REFERENCES `blacksmith`.`WorkshopRoom` (`ID_WR`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Workshop_WorkshopEmployee1`
    FOREIGN KEY (`WorkshopEmployee_ID_WE`)
    REFERENCES `blacksmith`.`WorkshopEmployee` (`ID_WE`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Workshop_Address1`
    FOREIGN KEY (`Address_ID_Address`)
    REFERENCES `blacksmith`.`Address` (`ID_Address`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


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 32)

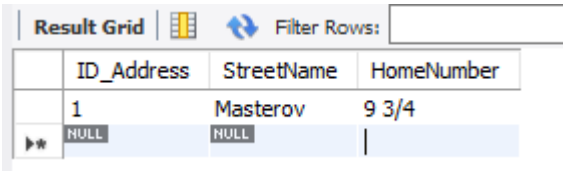


Рисунок 32 – Внесенные данные

Данные в таблице client соответствуют клиентам, которые собираются воспользоваться услугами(Рисунок 33)

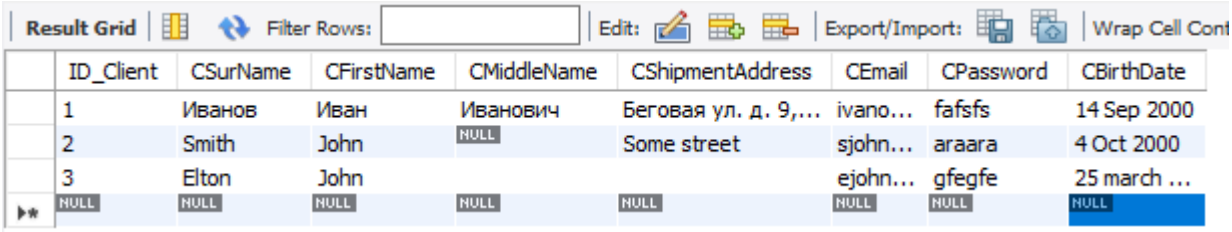


Рисунок 33 – Внесенные данные

Данные в таблице condition соответствуют состояниям инструментов(Рисунок 34)

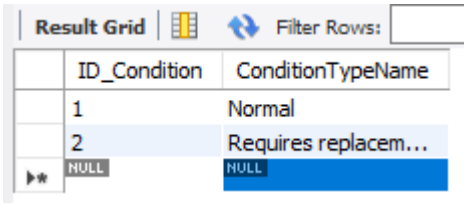


Рисунок 34 – Внесенные данные

Данные в таблице materialprovider соответствуют информации о поставщиках материалов (Рисунок 35)

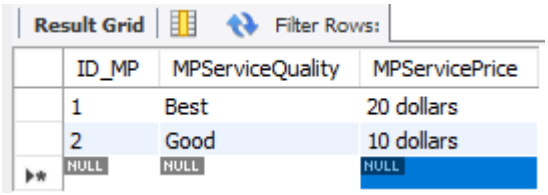


Рисунок 35 – Внесенные данные

Данные в таблице order соответствуют данным о заказах(Рисунок 36)

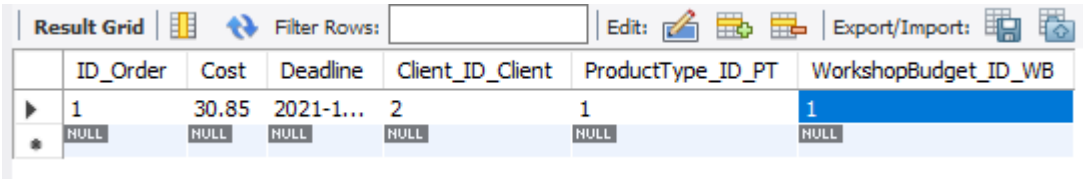


Рисунок 36 – Внесенные данные

Данные в таблице productype соответствуют предлагаемым вариантам товаров(Рисунок 37)

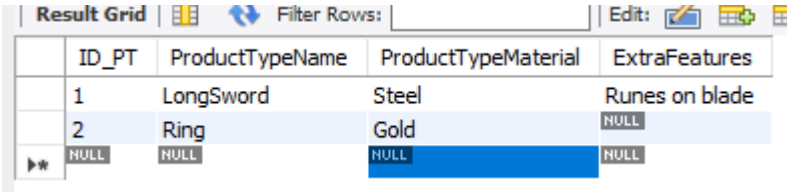


Рисунок 37 – Внесенные данные

Данные в таблице workequipment соответствуют информации об экипировке(Рисунок 38)

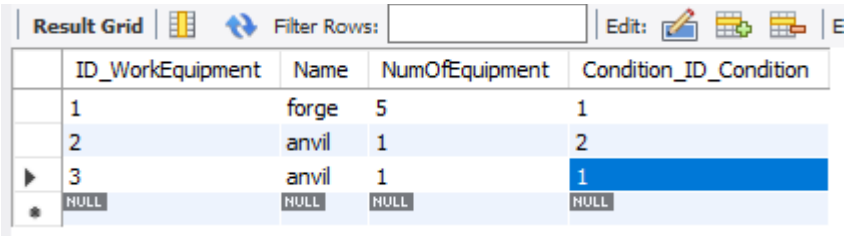


Рисунок 38 – Внесенные данные

Данные в таблице workshopbudget соответствуют состоянию бюджета (Рисунок 39)

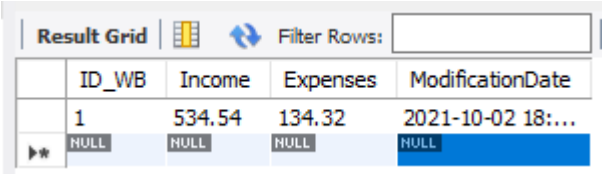
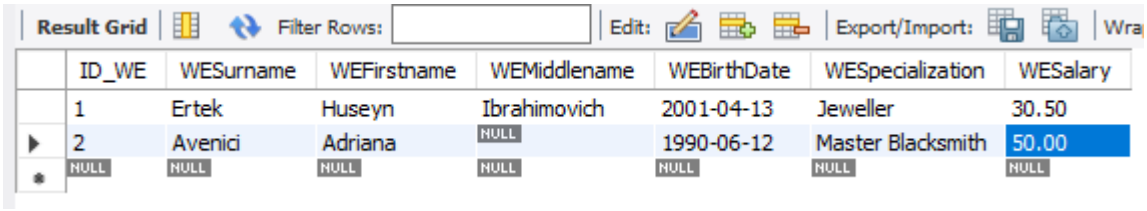


Рисунок 39 – Внесенные данные

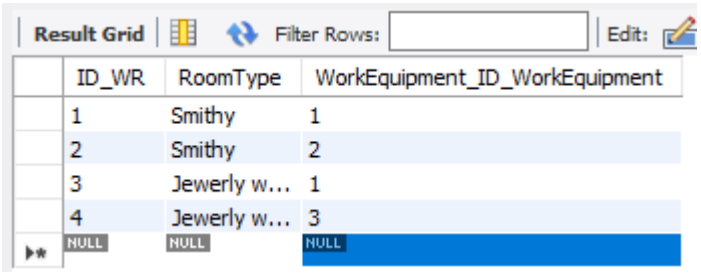
Данные в таблице workshopemployee отражают информацию о сотрудниках(Рисунок 40)



ID_WE	WESurname	WEFirstname	WEMiddlename	WEBirthDate	WESpecialization	WESalary
1	Ertek	Huseyn	Ibrahimovich	2001-04-13	Jeweller	30.50
2	Avenici	Adriana	NULL	1990-06-12	Master Blacksmith	50.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 40 – Внесенные данные

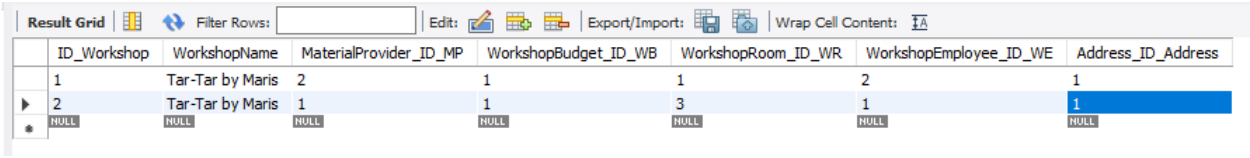
Данные в таблице workshoproom отражают информацию о помещениях(Рисунок 41)



ID_WR	RoomType	WorkEquipment_ID_WorkEquipment
1	Smithy	1
2	Smithy	2
3	Jewerly w...	1
4	Jewerly w...	3
NULL	NULL	NULL

Рисунок 41 – Внесенные данные

Данные в таблице workshop отражают информацию о мастерской в целом(Рисунок 42)



ID_Workshop	WorkshopName	MaterialProvider_ID_MP	WorkshopBudget_ID_WB	WorkshopRoom_ID_WR	WorkshopEmployee_ID_WE	Address_ID_Address
1	Tar-Tar by Maris	2	1	1	2	1
2	Tar-Tar by Maris	1	1	3	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 42 – Внесенные данные

Выборка данных. Оператор SELECT

Операция проекции (Рисунок 43)

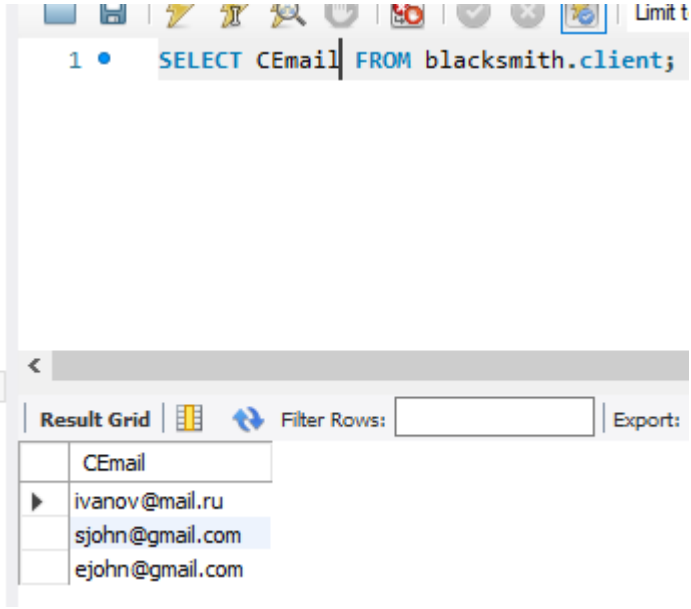


Рисунок 43 – Проекция

Операция селекции (Рисунок 44)

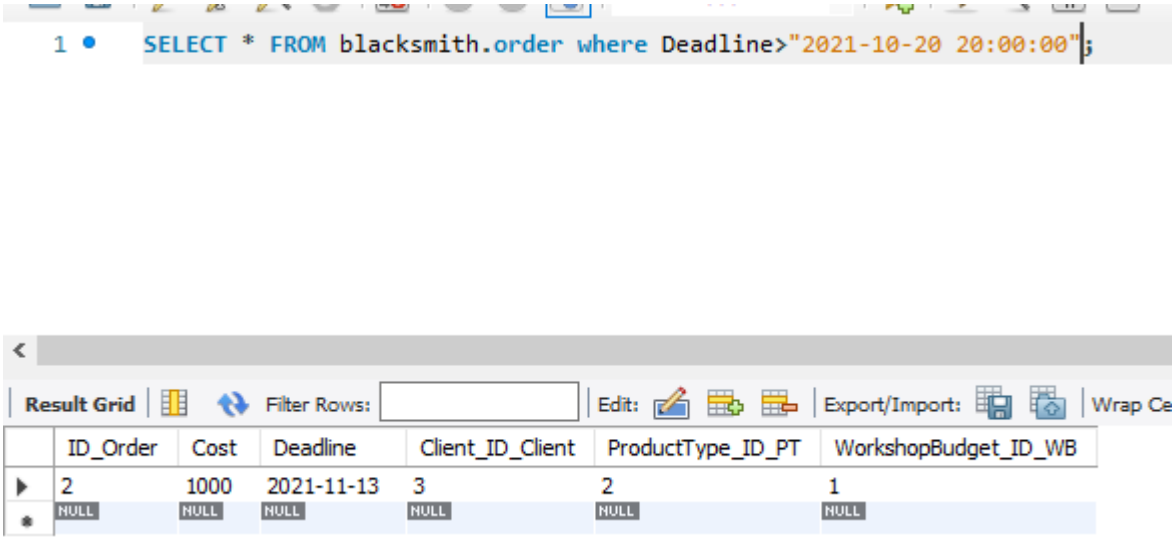


Рисунок 44 – Селекция

Операция соединения(Рисунок 45)

```
SELECT * FROM blacksmith.order o left outer join blacksmith.client c
on o.Client_ID_Client = c.ID_Client;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ID_Order	Cost	Deadline	Client_ID_Client	ProductType_ID_PT	WorkshopBudget_ID_WB	ID_Client	CSurName	CFirstName	CMiddleName	CShipmentAddress	CEmail	CPassword	CBirthDate
1	31	2021-10-20	2	1	1	2	Smith	John	NULL	Some street	sjohn@gmail.com	76445	2000-10-04
2	1000	2021-11-13	3	2	1	3	Elton	John			ejohn@gmail.com	3244234	1947-03-25

Рисунок 45 – Соединение

Операция объединения(Рисунок 46)

Limit to 1000 rows

```
(SELECT ID_Order FROM blacksmith.order where Client_ID_Client=3)
UNION
(SELECT ID_Order FROM blacksmith.order where cost=1000);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ID_Order
2

Рисунок 46 – Объединение

Создание триггеров.

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

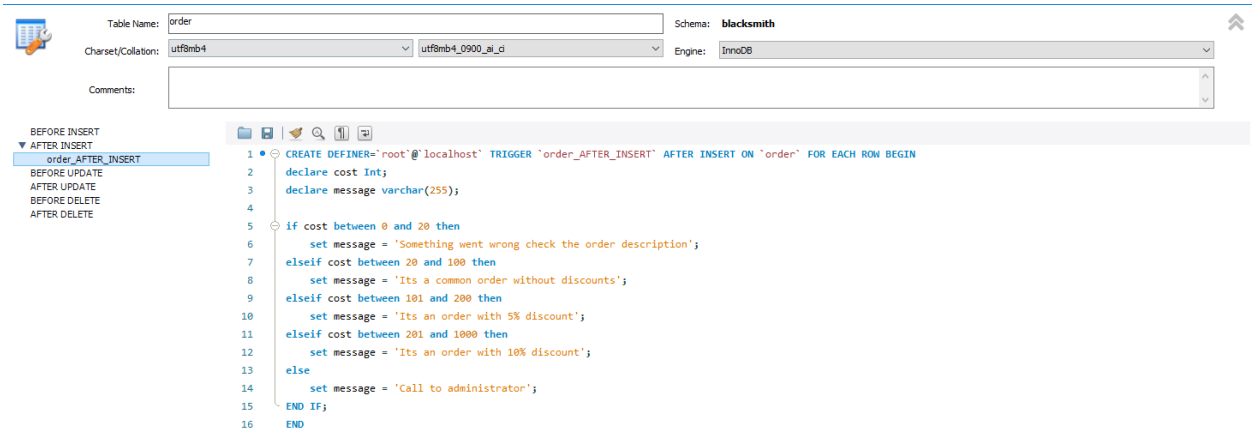


Рисунок 47 – Код триггера

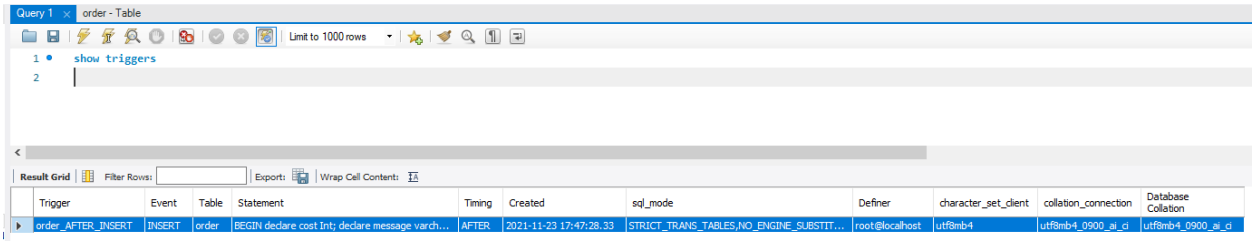


Рисунок 48 – Хранение триггера

Создание хранимых процедур

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

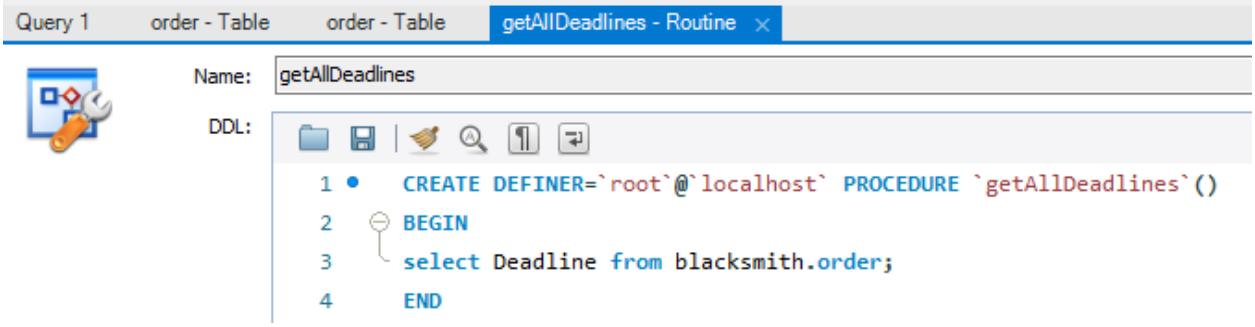


Рисунок 49 – Код процедуры

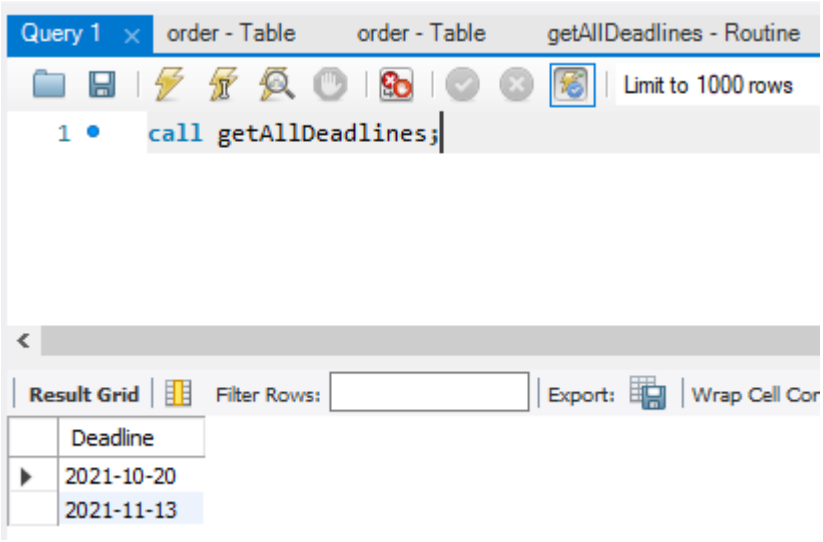


Рисунок 50 – Вызов процедуры

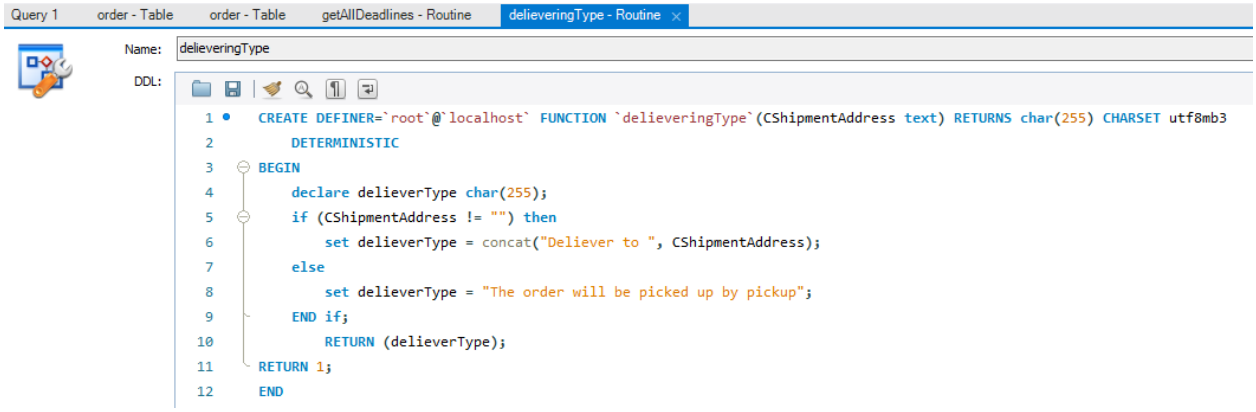


Рисунок 51 – Код функции

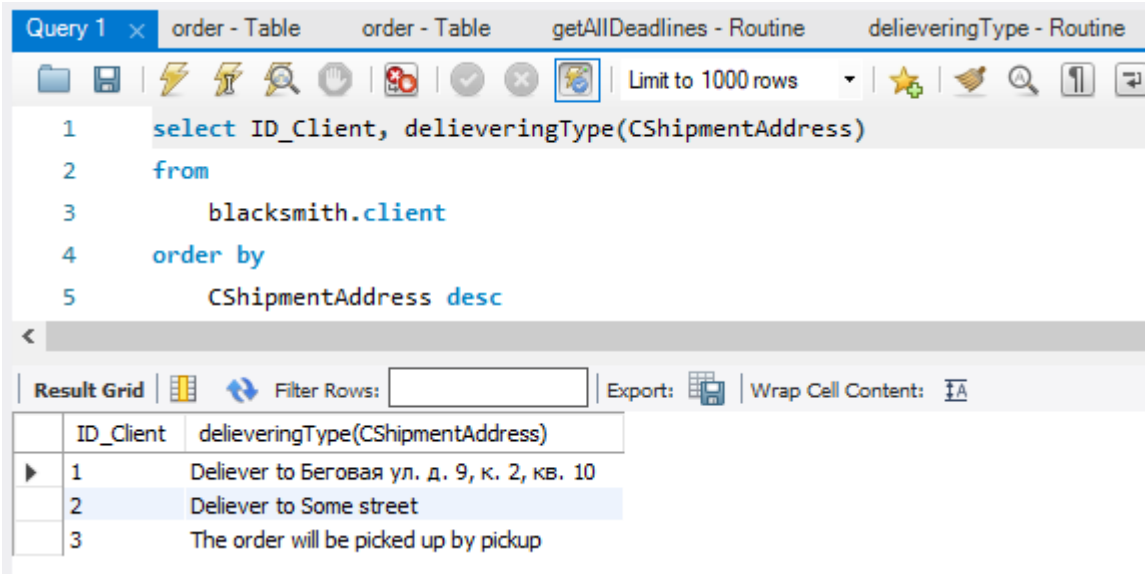


Рисунок 52 – Вызов функции

Резервное копирование

Здесь было выполнено копирование базы данных blacksmith:

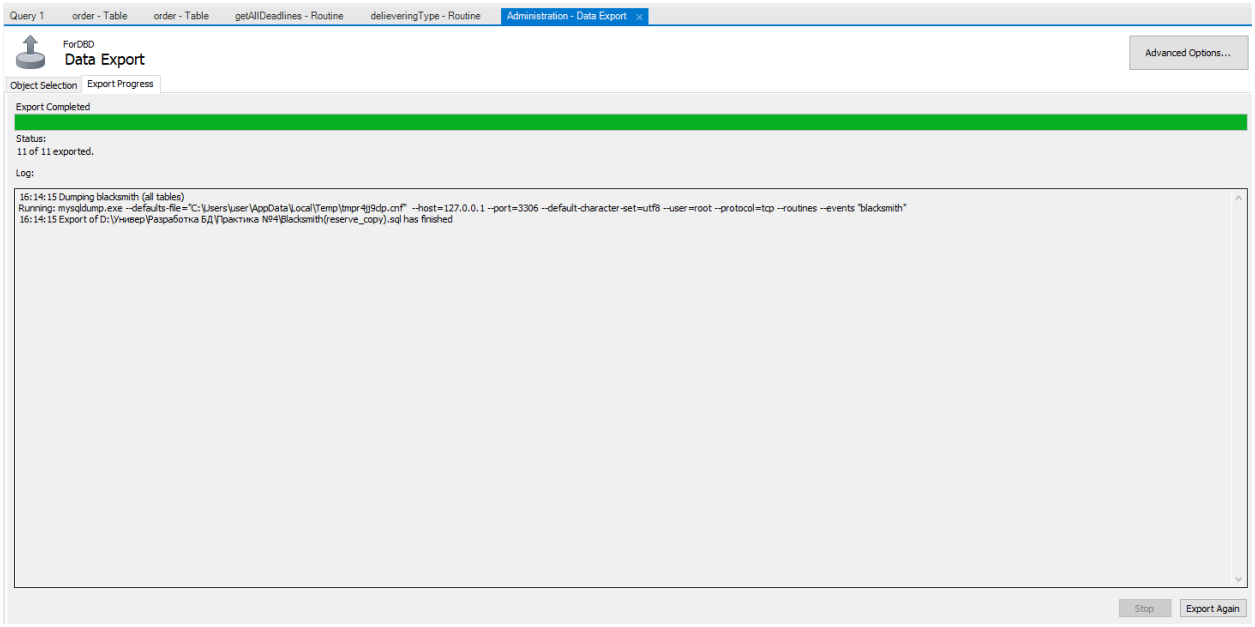


Рисунок 53 – Уведомление об успешном создании резервной копии

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.