|  |
| --- |
| 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-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»**

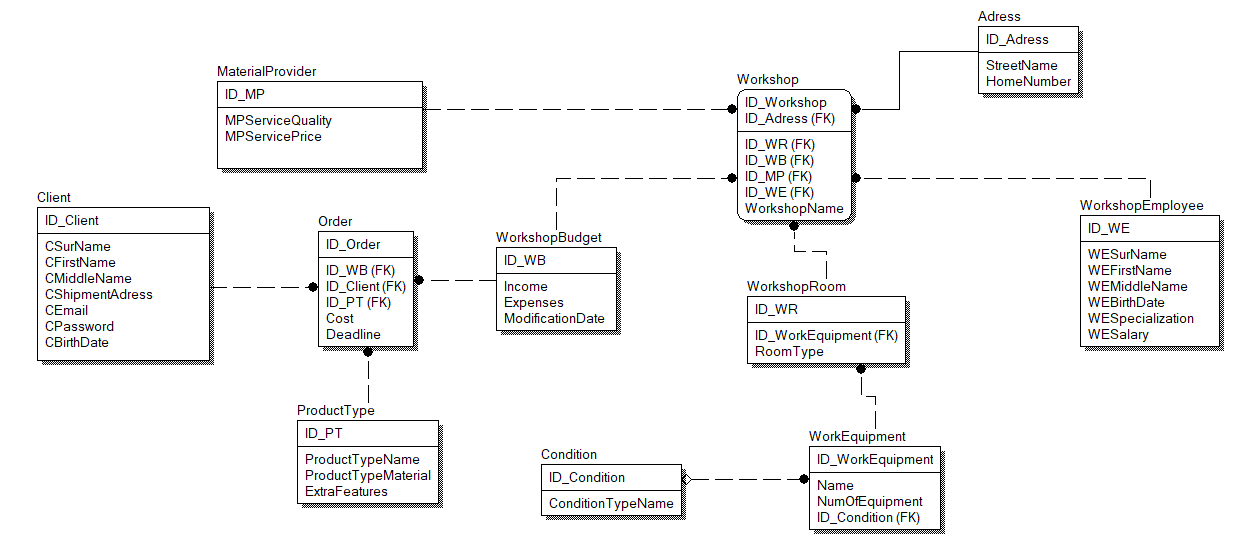


Рис. 1 ‒ Логическая модель данных на тему «Кузнечное дело»

**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»**

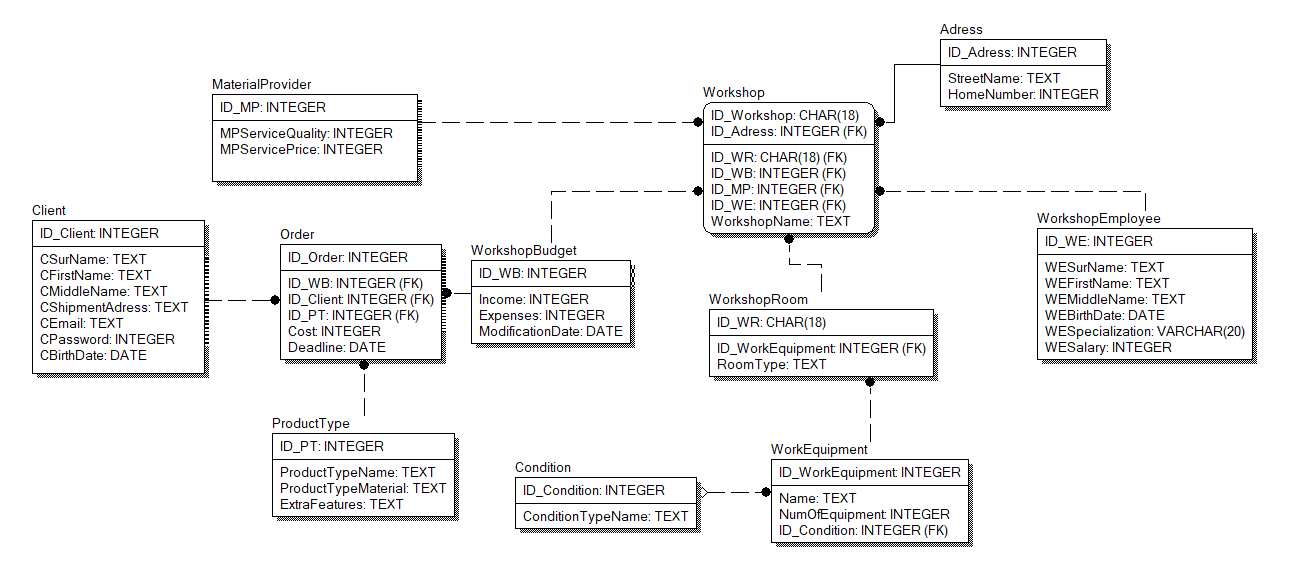
****

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

**Создание своей базы данных в MySQL CommandLine**

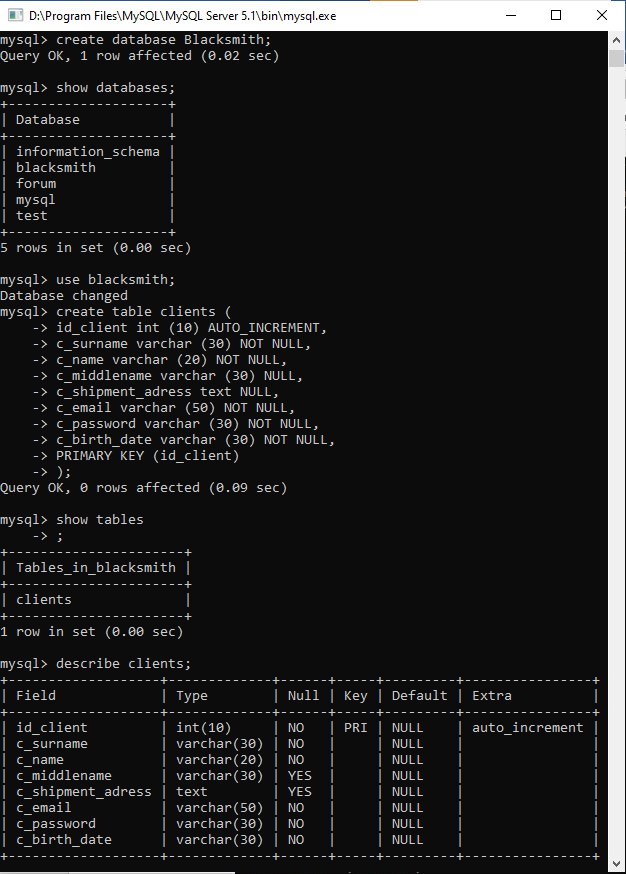
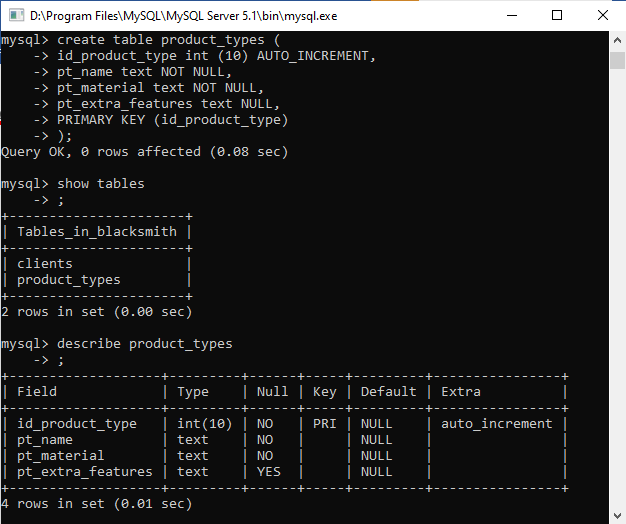
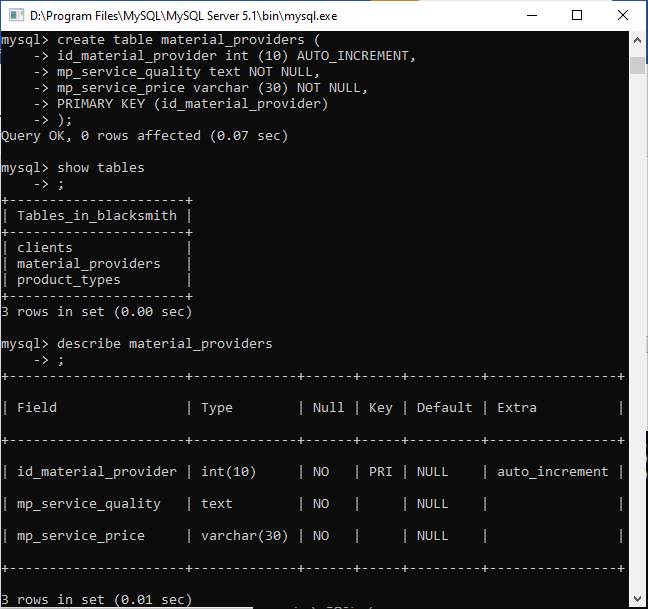


Рис. 3 Создание БД Blacksmith, таблицы clients и проверка

Рис. 4 Создание таблицы product\_types и проверка

Рис. 5 Создание таблицы material\_providers и проверка

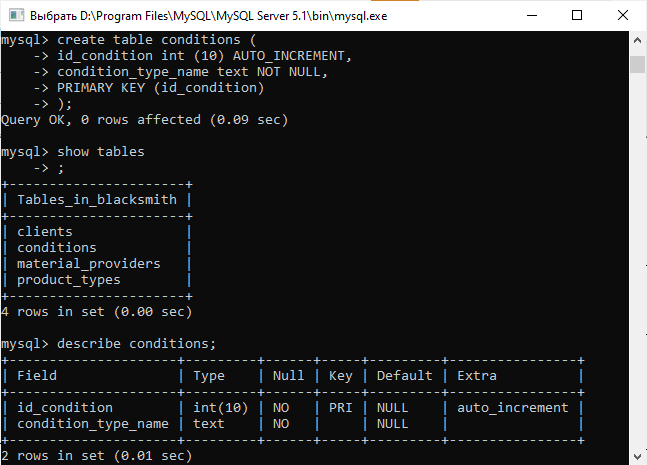


Рис. 6 Создание таблицы conditions и проверка

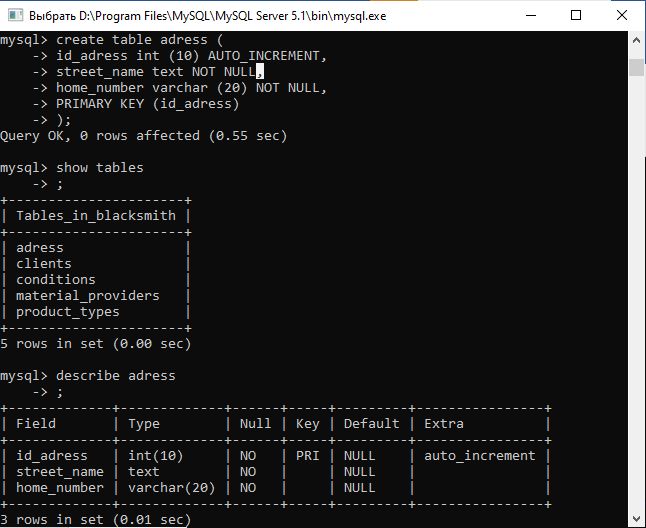


Рис. 7 Создание таблицы adress и проверка

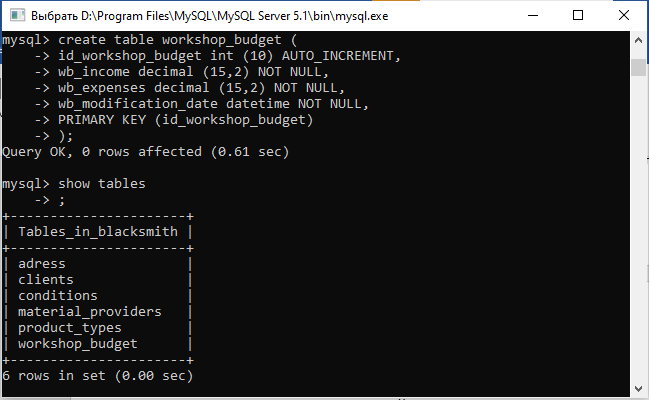


Рис. 8 Создание таблицы workshop\_budget

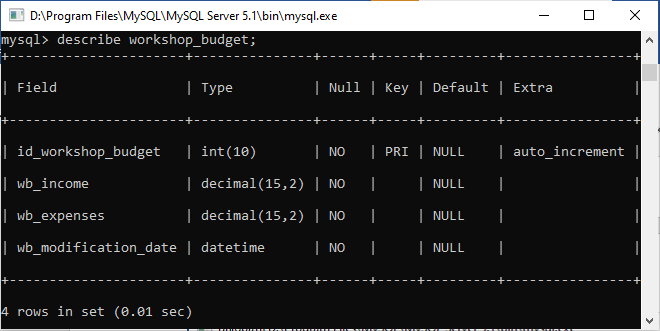


Рис. 9 Проверка правильности таблицы workshop\_budget

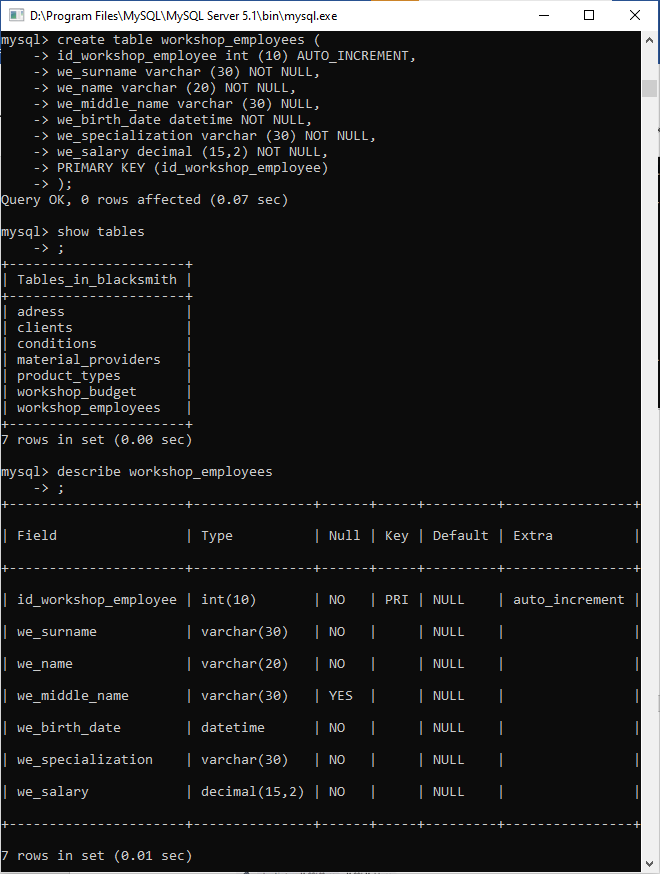


Рис. 10 Создание таблицы workshop\_employees и проверка

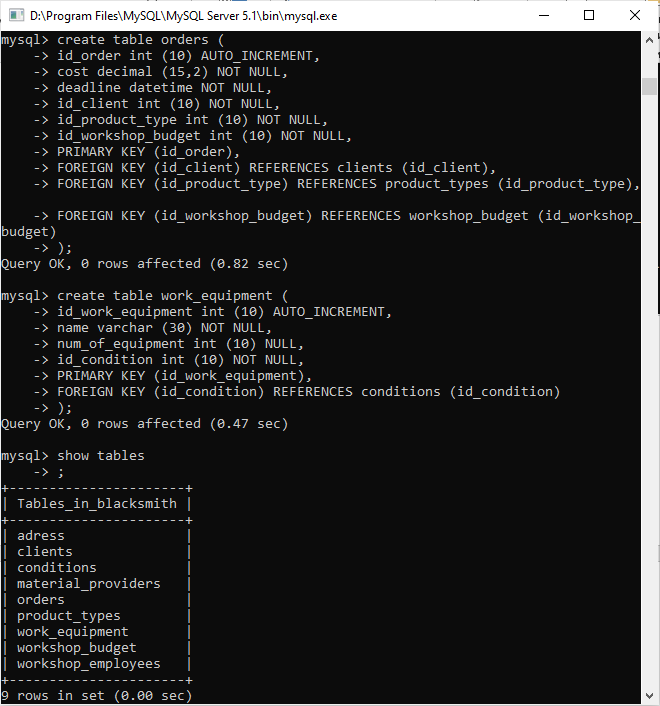


Рис. 11 Создание таблиц orders и work\_equipment

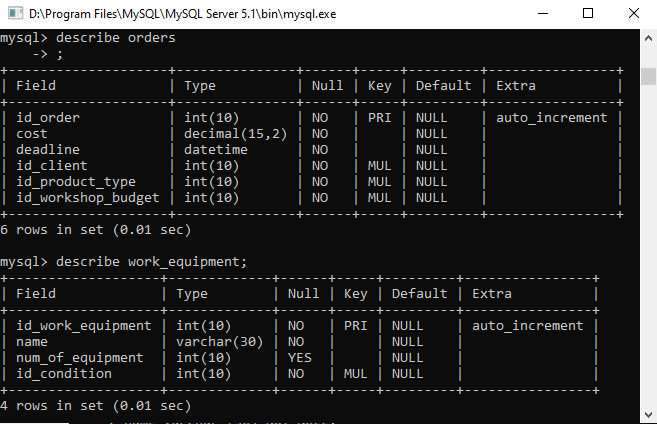


Рис. 12 Проверка правильности таблиц orders и work\_equipment

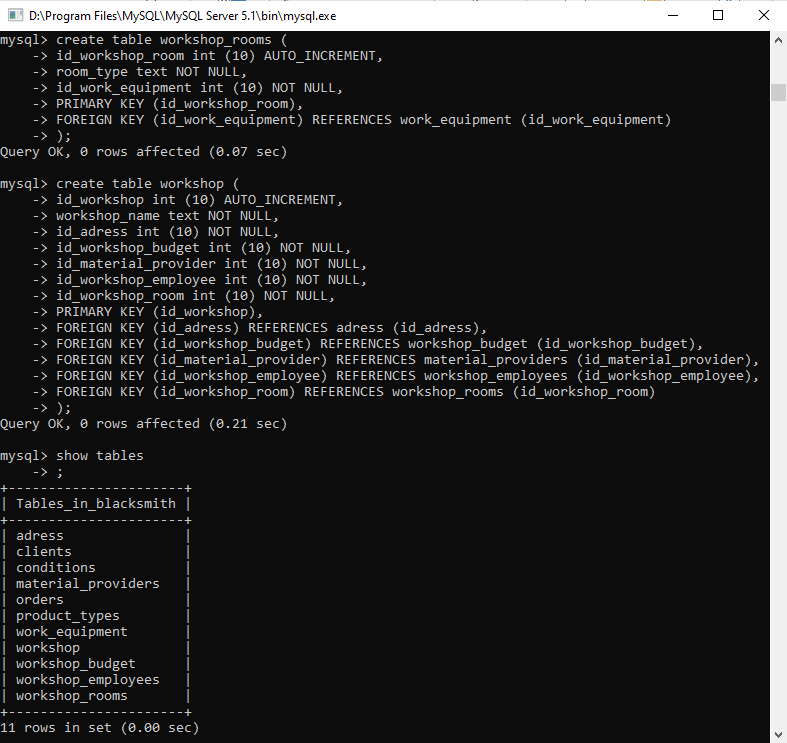


Рис. 13 Создание таблиц workshop\_rooms и workshop

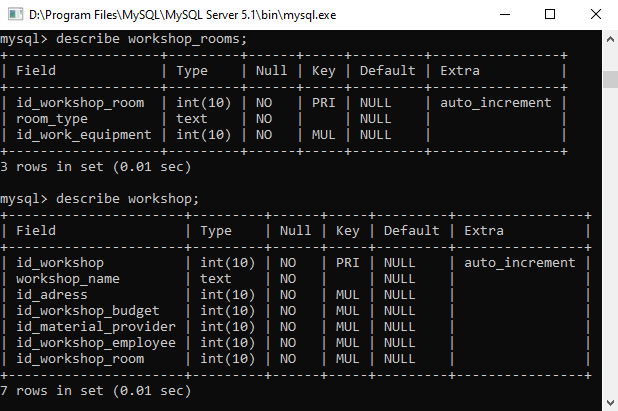


Рис. 14 Проверка правильности таблиц workshop\_rooms и workshop

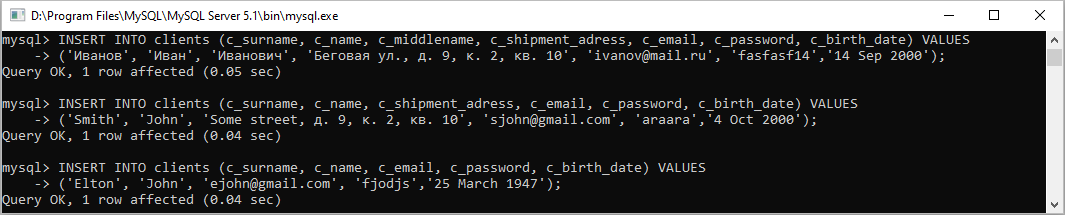


Рис. 15 Заполнение таблицы clients

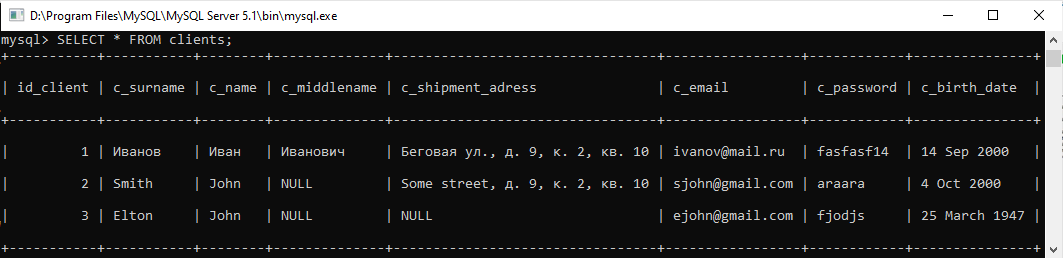


Рис. 16 Вывод элементов таблицы clients

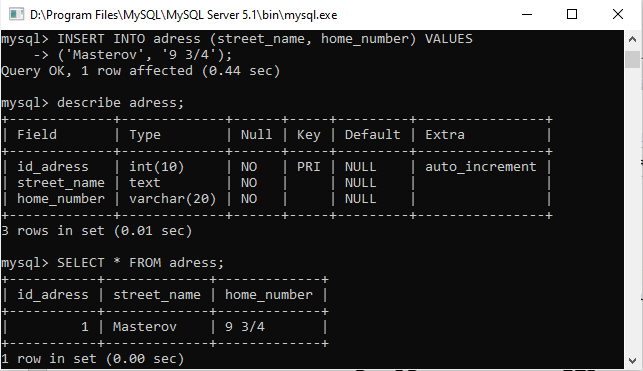


Рис. 17 Заполнение и вывод элементов таблицы adress

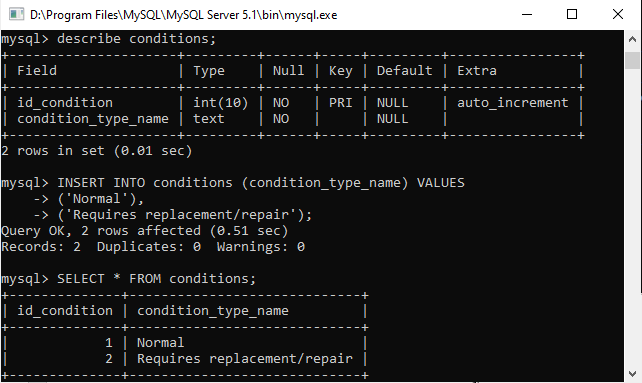


Рис. 18 Заполнение и вывод элементов таблицы conditions

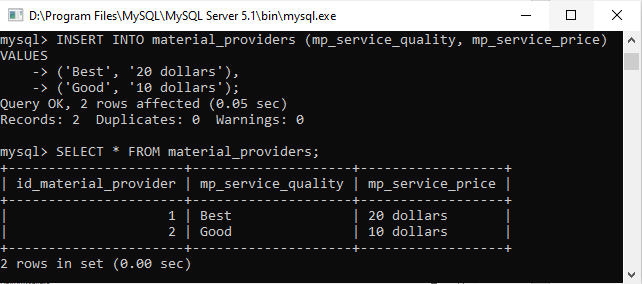


Рис. 19 Заполнение и вывод элементов таблицы material\_providers

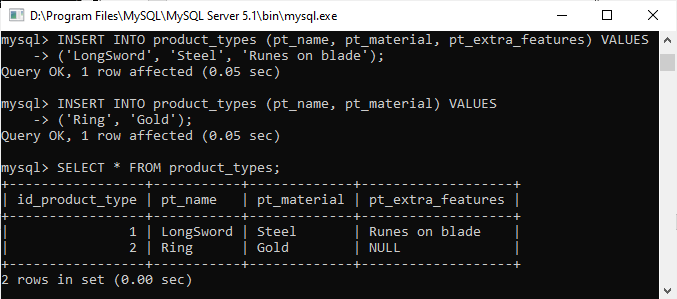


Рис. 20 Заполнение и вывод элементов таблицы product\_types

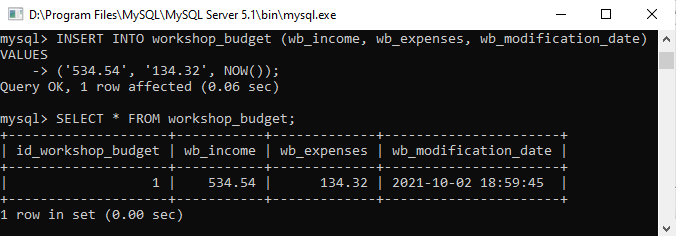


Рис. 21 Заполнение и вывод элементов таблицы workshop\_budget

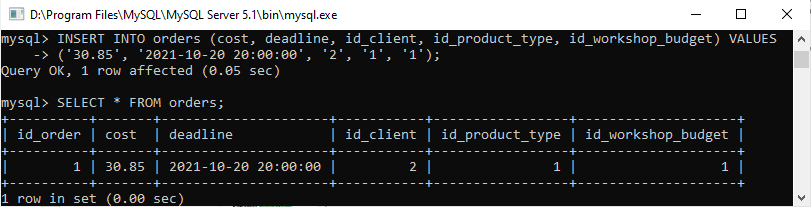


Рис. 22 Заполнение и вывод элементов таблицы orders

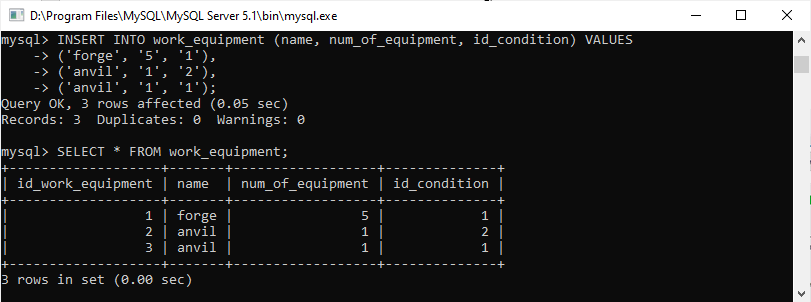


Рис. 23 Заполнение и вывод элементов таблицы work\_equipment

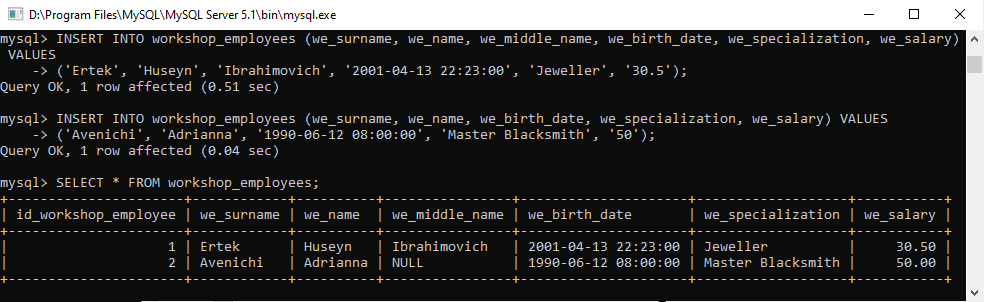


Рис. 24 Заполнение и вывод элементов таблицы workshop\_employees

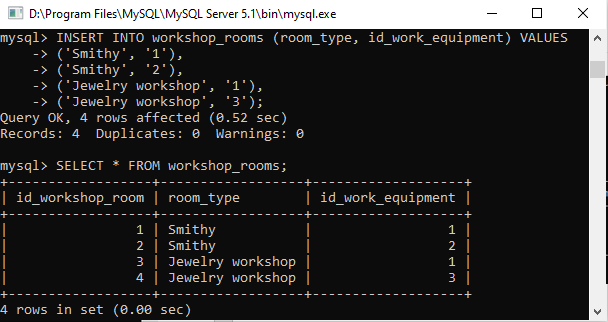


Рис. 25 Заполнение и вывод элементов таблицы workshop\_rooms

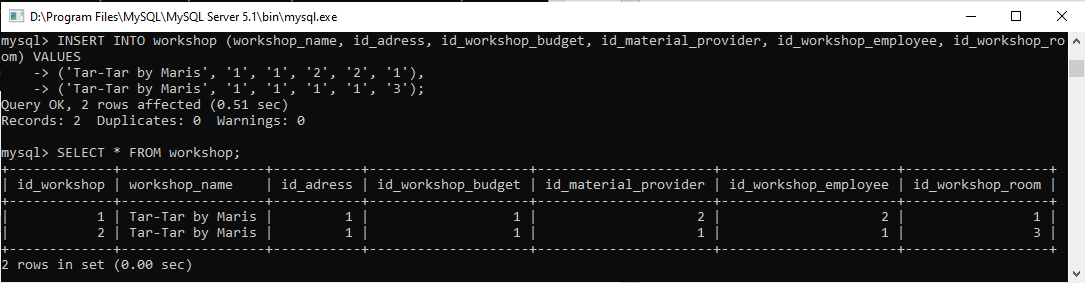
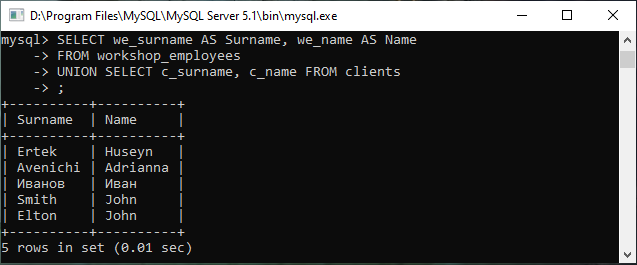


Рис. 26 Заполнение и вывод элементов таблицы workshop

**Работа с запросами в базе данных Blacksmith**

  
Рис. 27 ‒ Объединение с использованием UNION

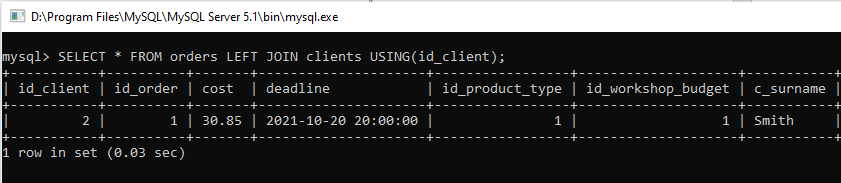


Рис. 28 ‒ Выборка с использованием LEFT JOIN часть 1

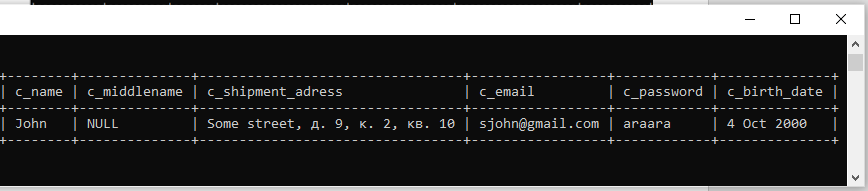
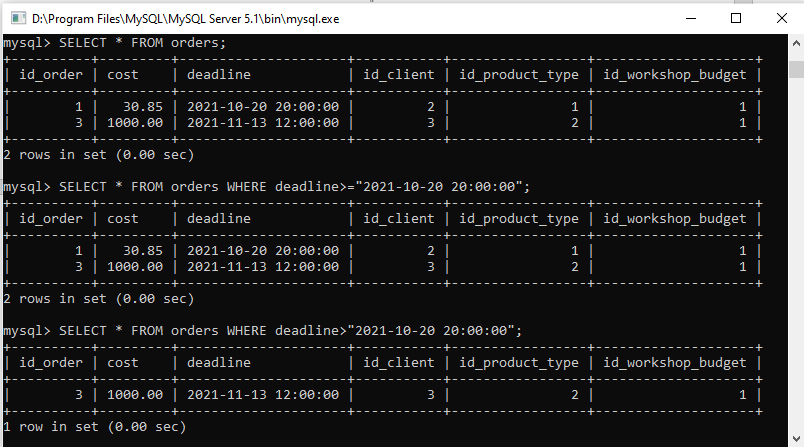


Рис. 29 ‒ Выборка с использованием LEFT JOIN часть 2

Рис. 30 Выборка данных по параметру deadline

**Построение модели с помощью оболочки MySQL Workbench**

Мною была построена модель в СУБД Workbench(Рисунок 31)

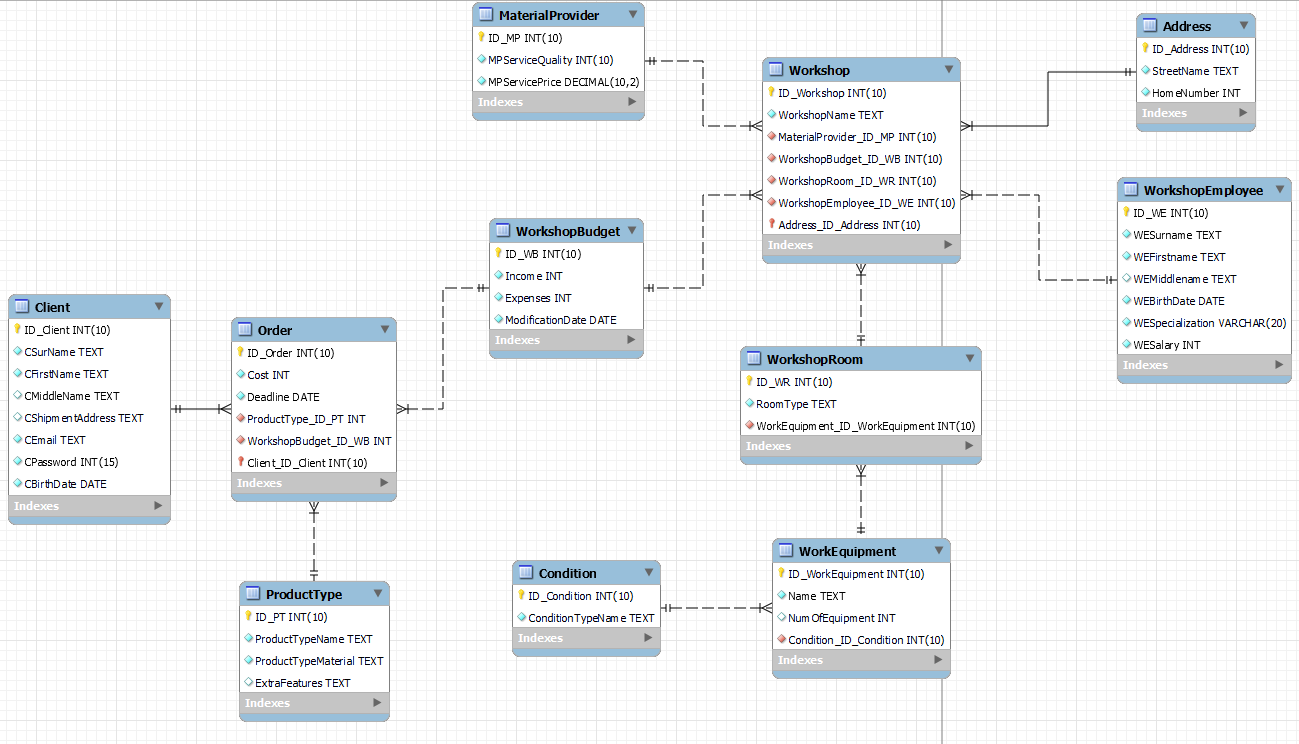


Рисунок 31 – Построенная модель

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)

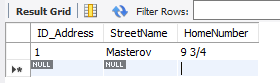


Рисунок 32 – Внесенные данные

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

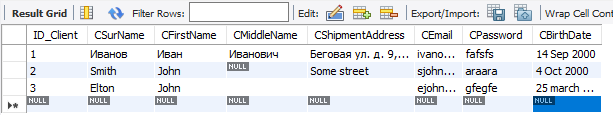


Рисунок 33 – Внесенные данные

Данные в таблице condition соответствуют состояниям инструментов(Рисунок 34)

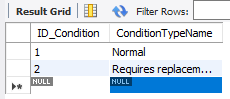


Рисунок 34 – Внесенные данные

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

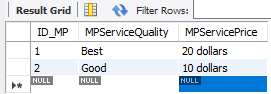


Рисунок 35 – Внесенные данные

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

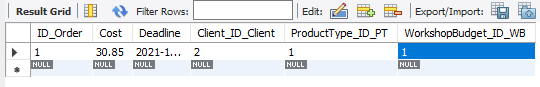


Рисунок 36 – Внесенные данные

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

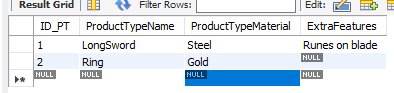


Рисунок 37 – Внесенные данные

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

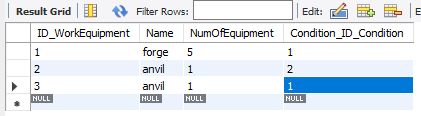


Рисунок 38 – Внесенные данные

Данные в таблице workshopbudget соответствуют состоянию бюджета (Рисунок 39)

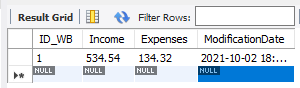


Рисунок 39 – Внесенные данные

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

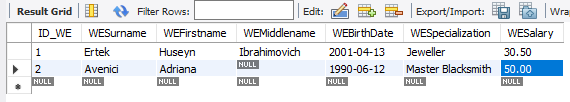


Рисунок 40 – Внесенные данные

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

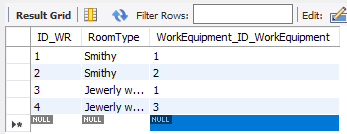


Рисунок 41 – Внесенные данные

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

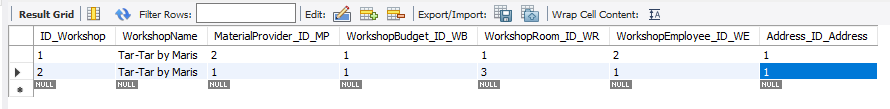


Рисунок 42 – Внесенные данные

**Выборка данных. Оператор SELECT**

Операция проекции (Рисунок 43)

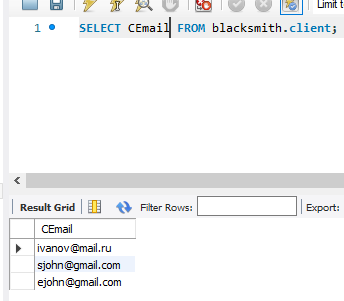
****

Рисунок 43 – Проекция

Операция селекции (Рисунок 44)

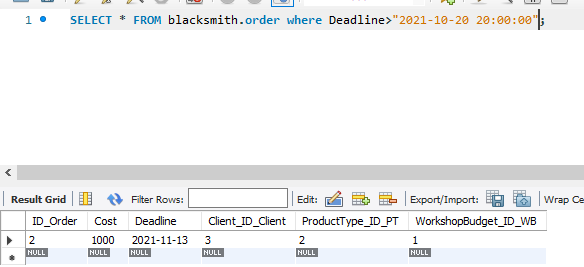


Рисунок 44 – Селекция

Операция соединения(Рисунок 45)

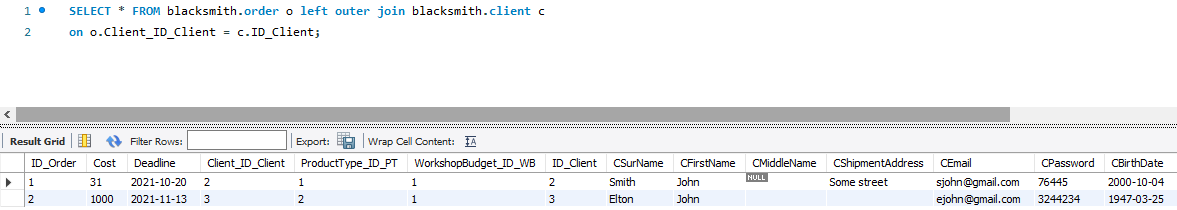


Рисунок 45 – Соединение

Операция объединения(Рисунок 46)

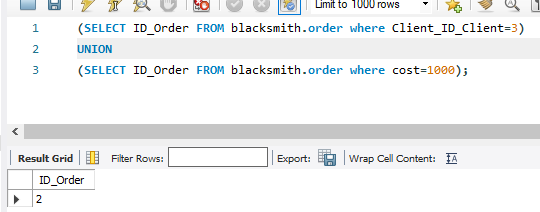


Рисунок 46 – Объединение

**Создание триггеров.**

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

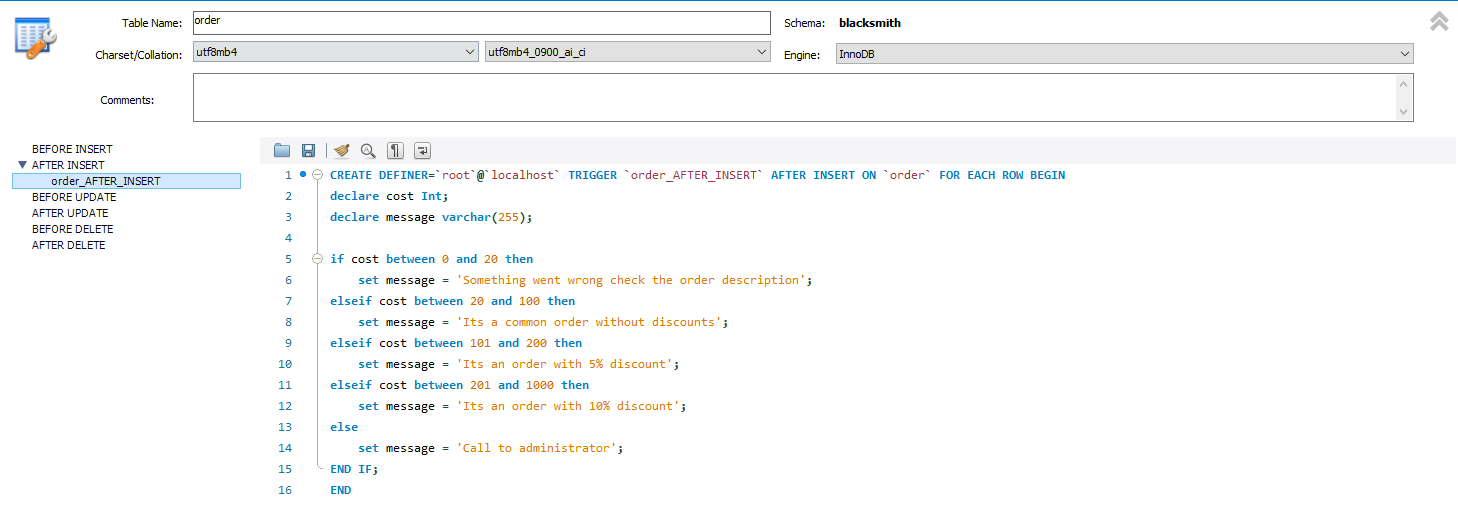


Рисунок 47 ‒ Код триггера

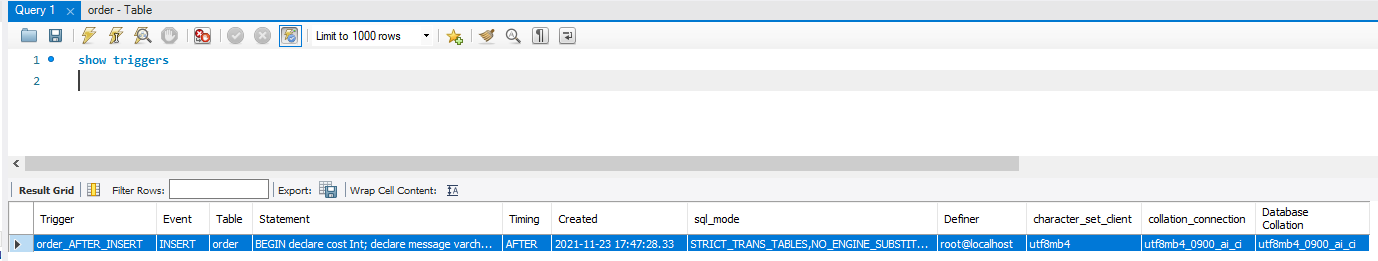


Рисунок 48 ‒ Хранение триггера

**Создание хранимых процедур**

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

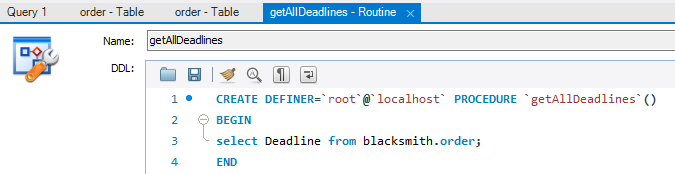


Рисунок 49 ‒ Код процедуры

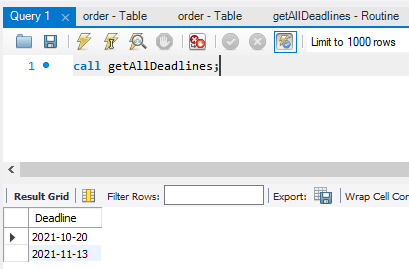


Рисунок 50 ‒ Вызов процедуры

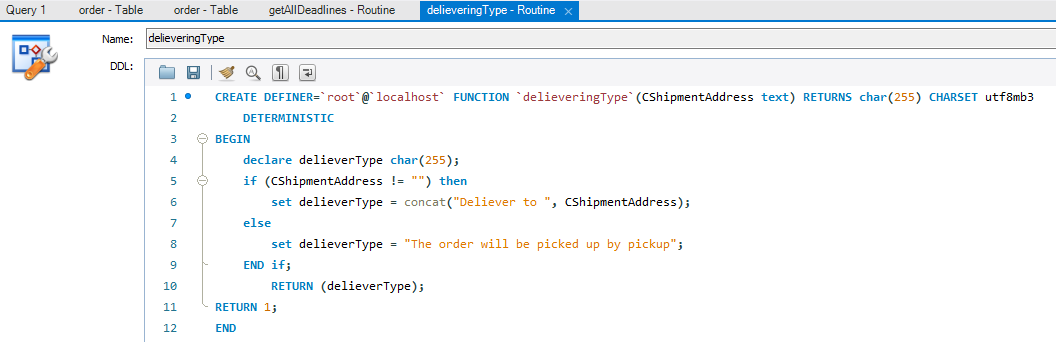


Рисунок 51 ‒ Код функции

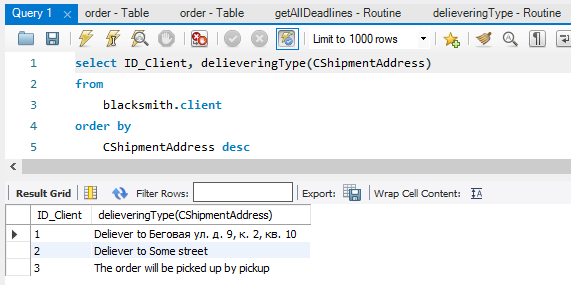


Рисунок 52 ‒ Вызов функции

**Резервное копирование**

Здесь было выполнено копирование базы данных blacksmith:

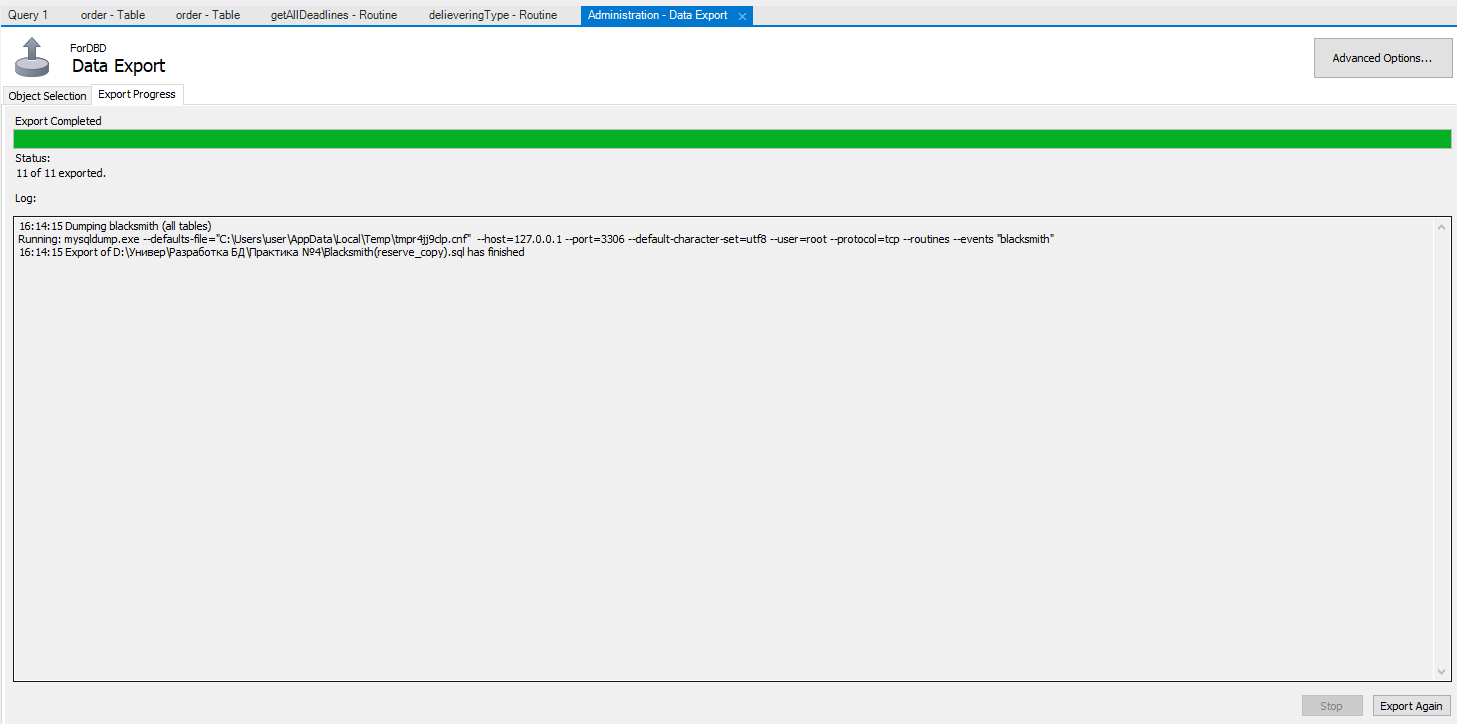


Рисунок 53 ‒ Уведомление об успешном создании резервной копии

**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.