|  |
| --- |
| 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, Московка А.А.* | (подпись) | |
| Преподаватель | *Володина А.М.* | (подпись) | |
| Отчет представлен | «\_\_» декабря 2021г. | |  | |

Москва 2021 г.

**Введение**

В качестве темы для создания базы данных была выбрана «Подготовка и сдача экзаменов». В процессе работы данная тема приняла направленность базы данных для учеников для просмотра сданных экзаменов. В виду этого была создана модель, ориентированная под запросы со стороны учеников. В ней были описаны основные элементы, которые должны быть задействованы в подобной точке зрения.

**Логическая модель базы данных по теме «Подготовка и сдача экзаменов»**

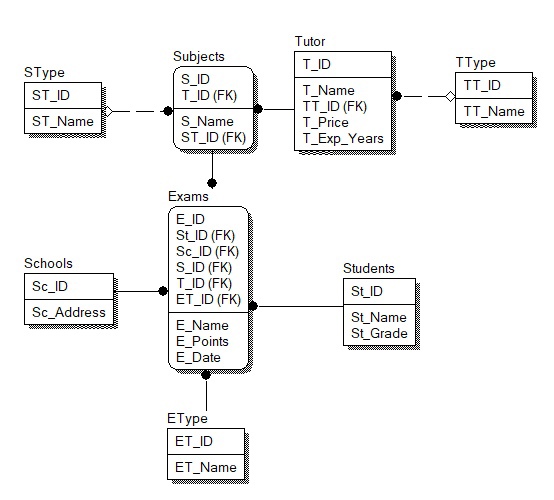


Рис. 1 ‒ Логическая модель данных на тему «Подготовка и сдача экзаменов»

**Описание**

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

Exams – содержит основные данные о названии экзамена, баллах за экзамен и дате написания экзамена.

EType – содержит типы экзаменов.

Students – содержит данные об учениках, а именно: имя и класс.

Schools – содержит информацию об адресе школы, номером школы является его уникальный идентификатор.

Subjects – содержит информацию о названии предметов, а также внешний ключ типов предметов.

SType – содержит поле, определяющее, обязательный предмет или нет.

Tutor – содержит информацию об имени преподавателя, его цене за занятие и количестве лет опыта, также внешний ключ типа репетитора.

TType – содержит сведения о том, кем преподаватель является

**Физическая модель базы данных по теме «Издание статей»**

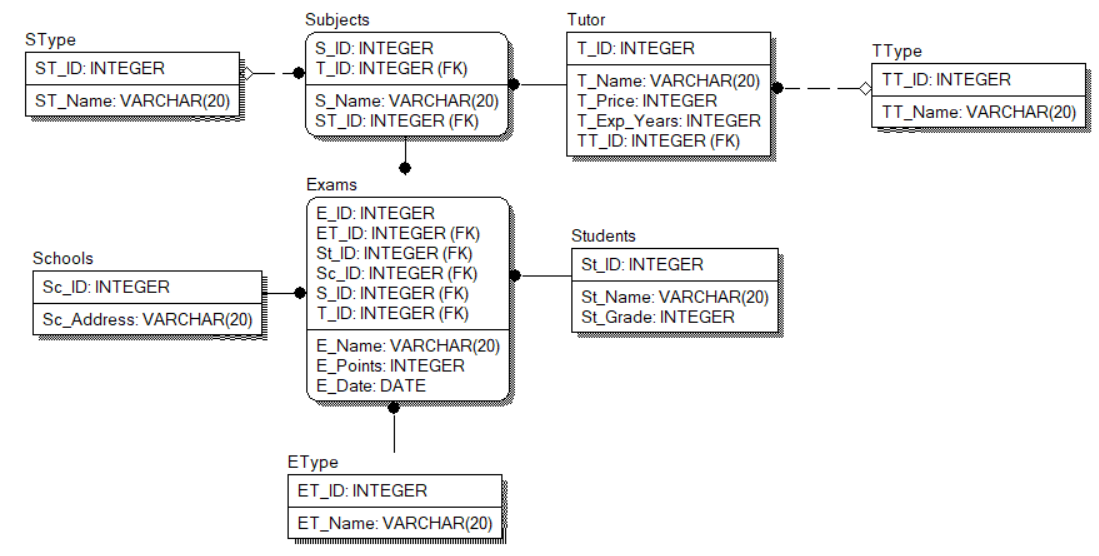
****

Рис. 2 ‒ Физическая модель данных на тему «Издание статей»

**Создание своей базы данных в MySQL CommandLine**

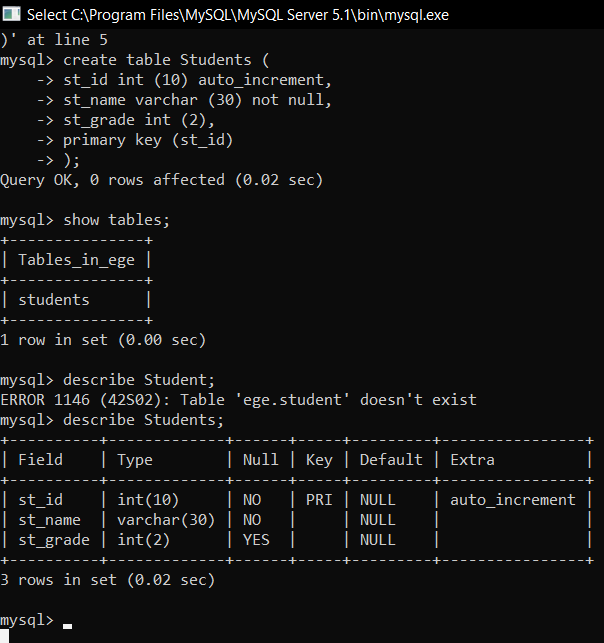
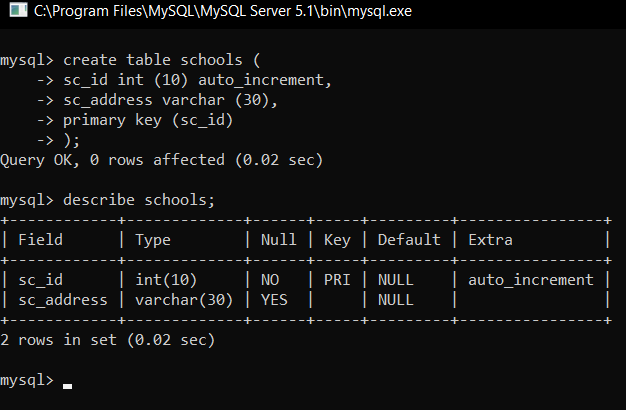
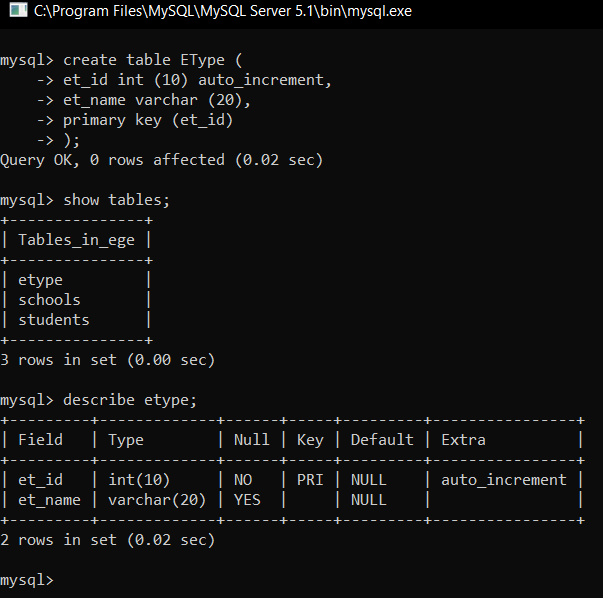


Рис. 3 – Создание БД Ege, таблицы Students

Рис. 4 – Создание таблицы Schools

 Рис. 5 – Создание таблицы EType

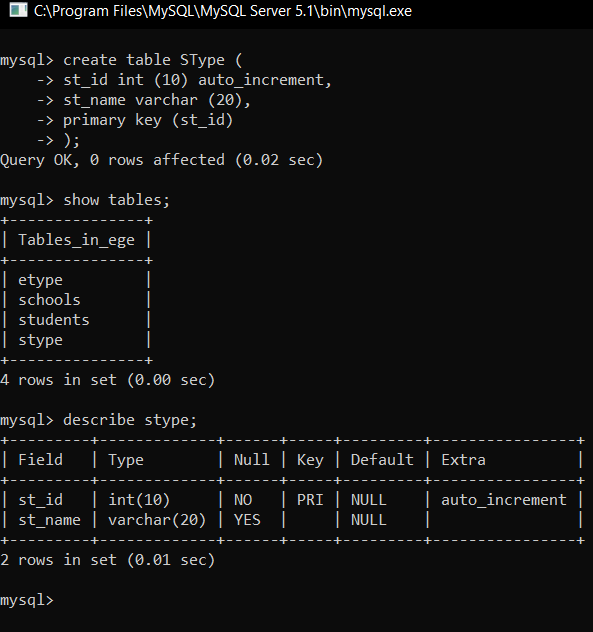


Рис. 6 – Создание таблицы SType

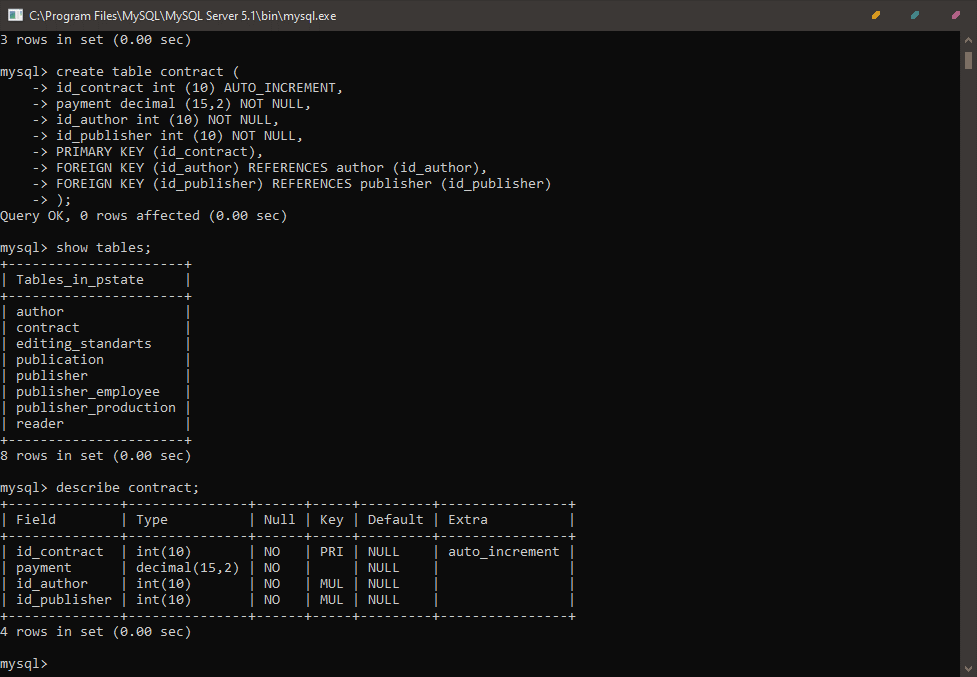


Рис. 7 Создание таблицы contract и проверка

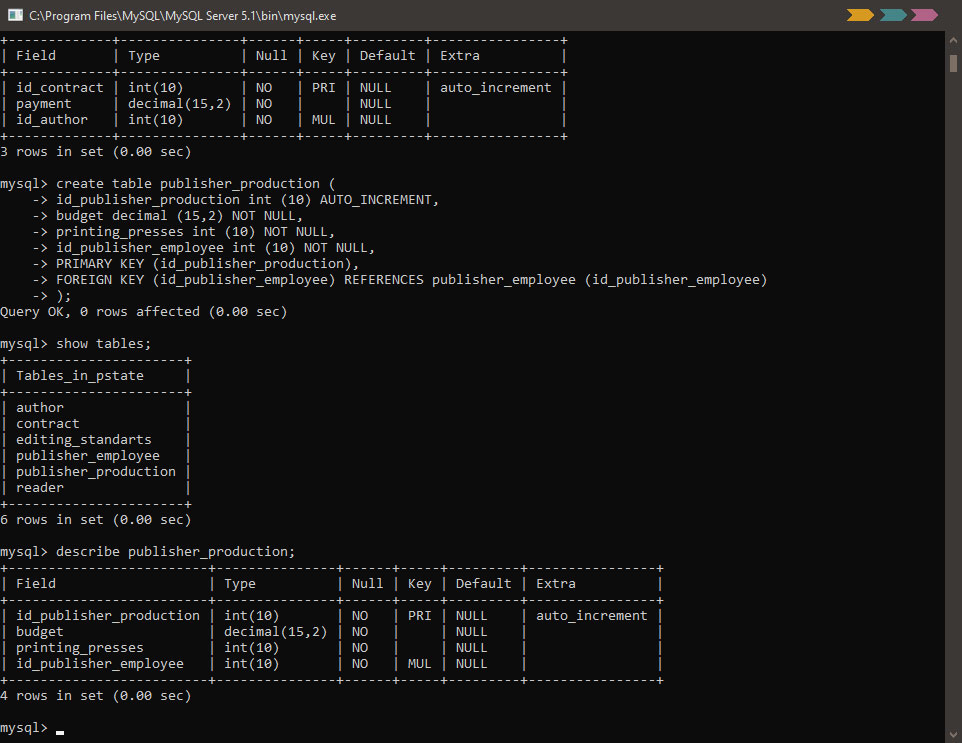


Рис. 8 Создание таблицы publisher\_production и проверка

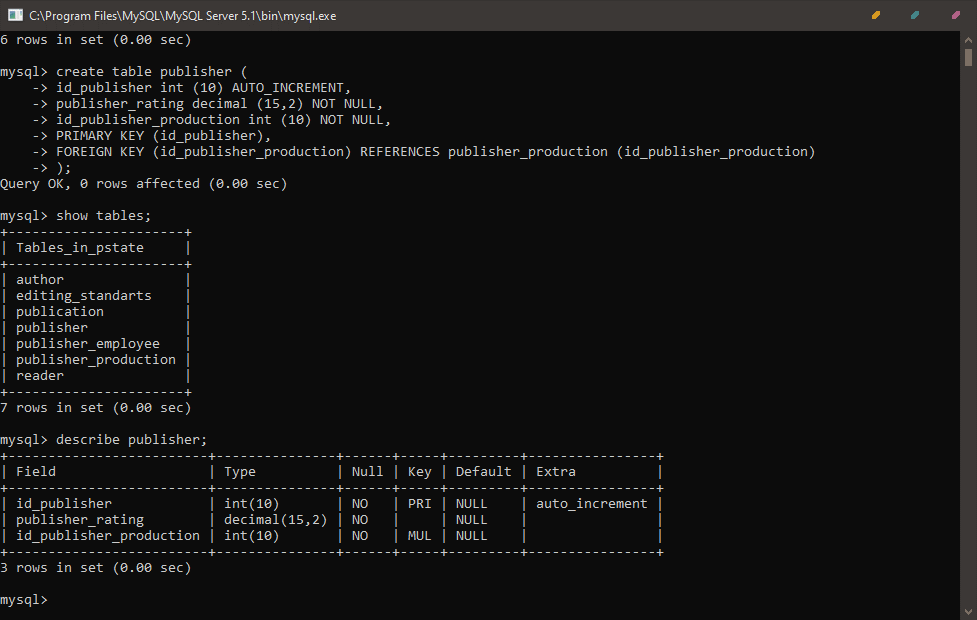


Рис. 9 Создание таблицы publisher и проверка

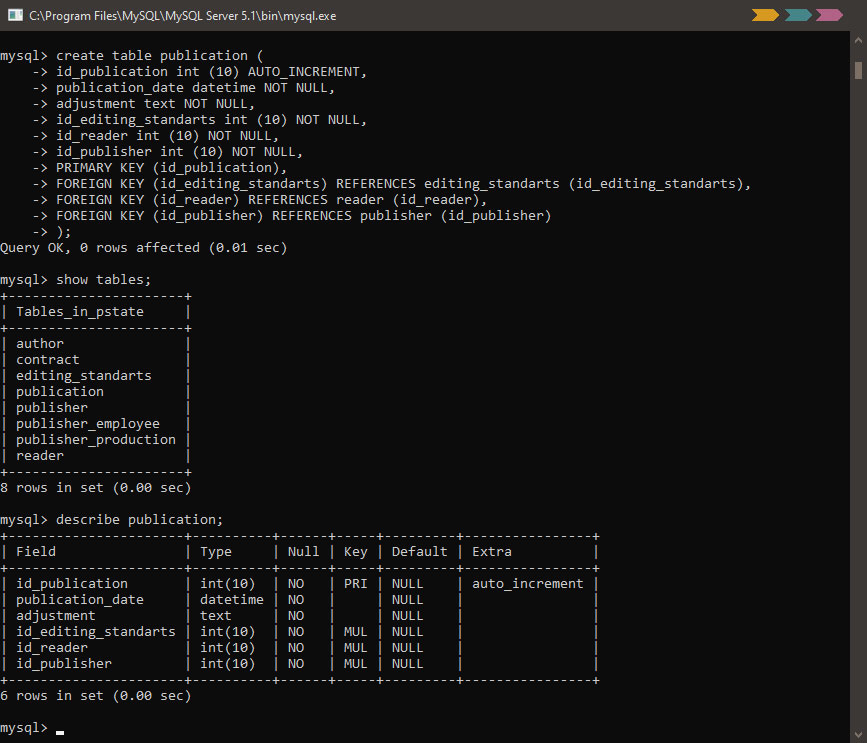


Рис. 10 Создание таблицы publication и проверка

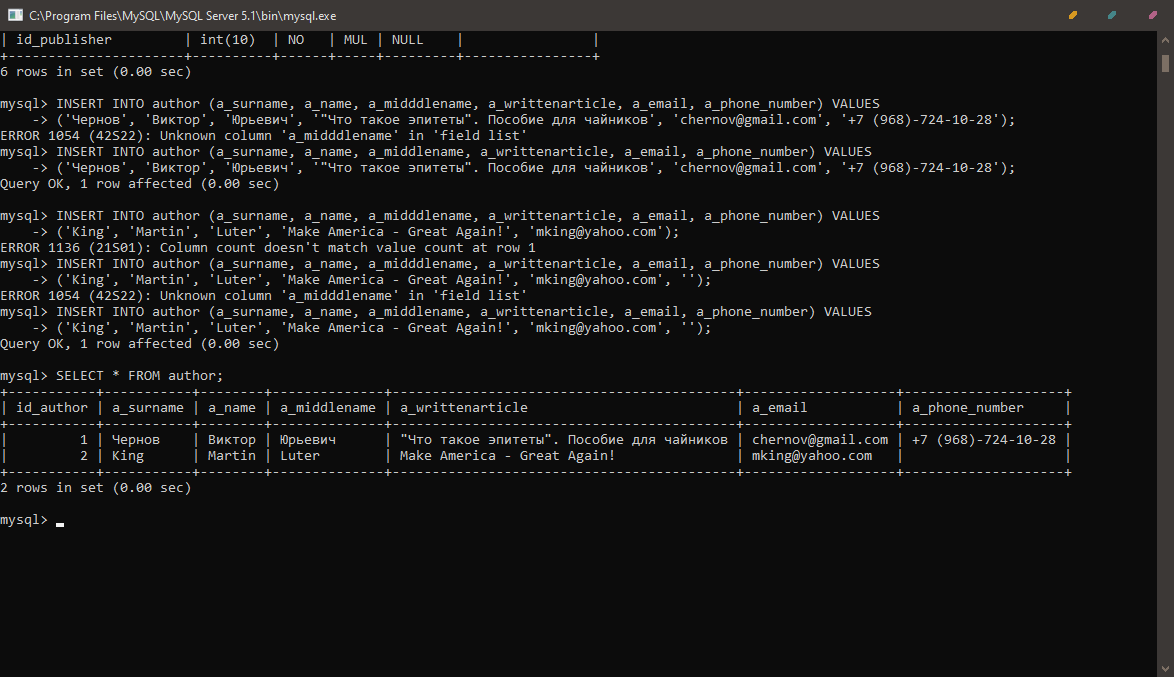


Рис. 11 Заполнение и вывод элементов таблицы author

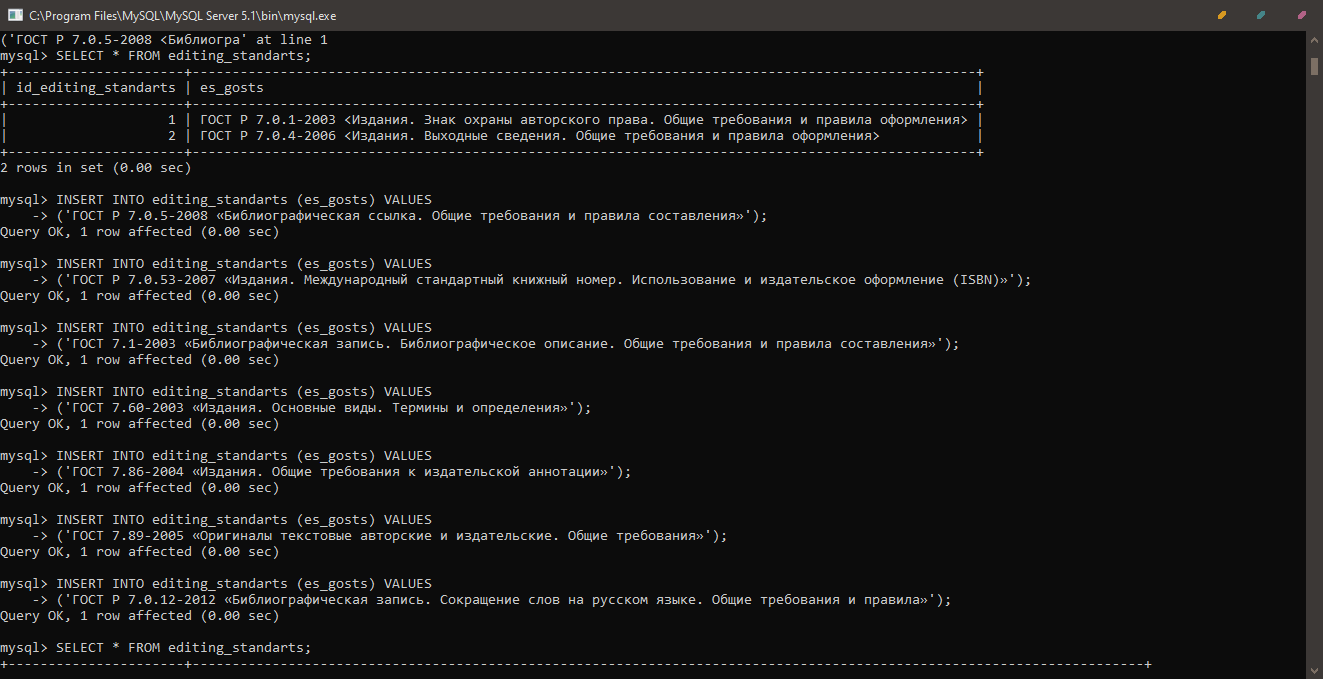


Рис. 12 Заполнение editing\_standarts

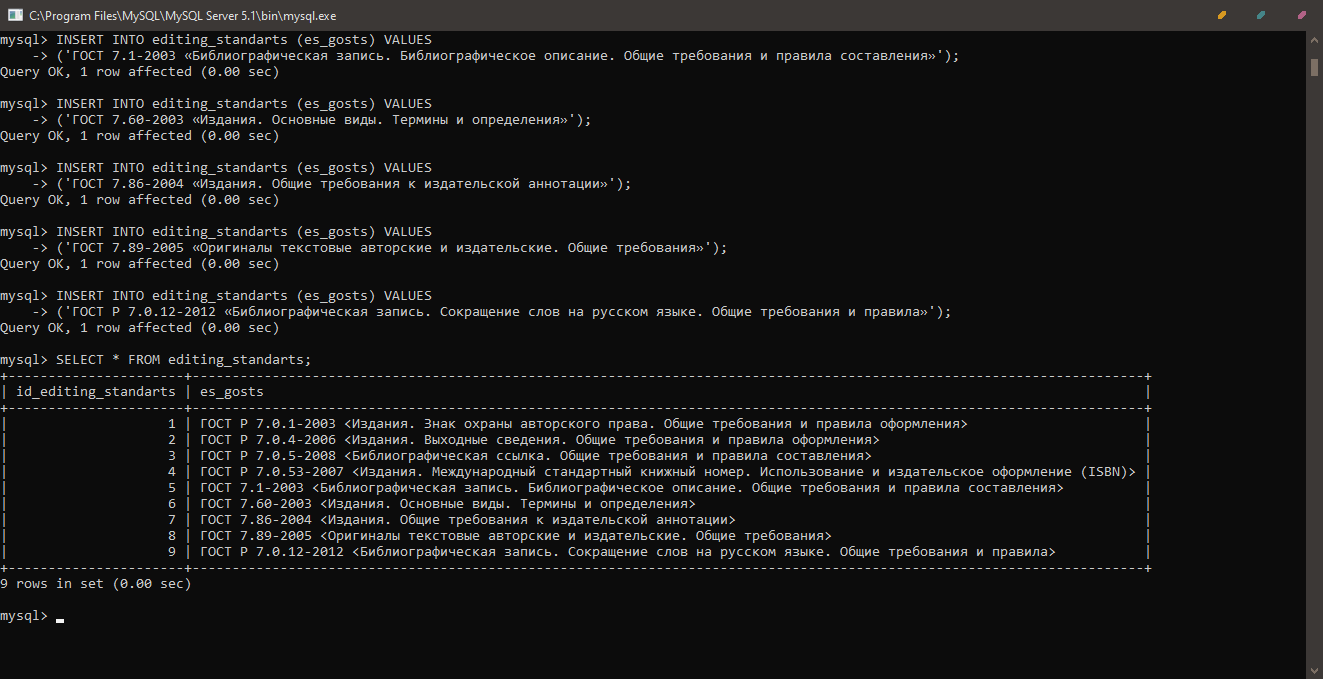


Рис. 13 Вывод элементов таблицы editing\_standarts

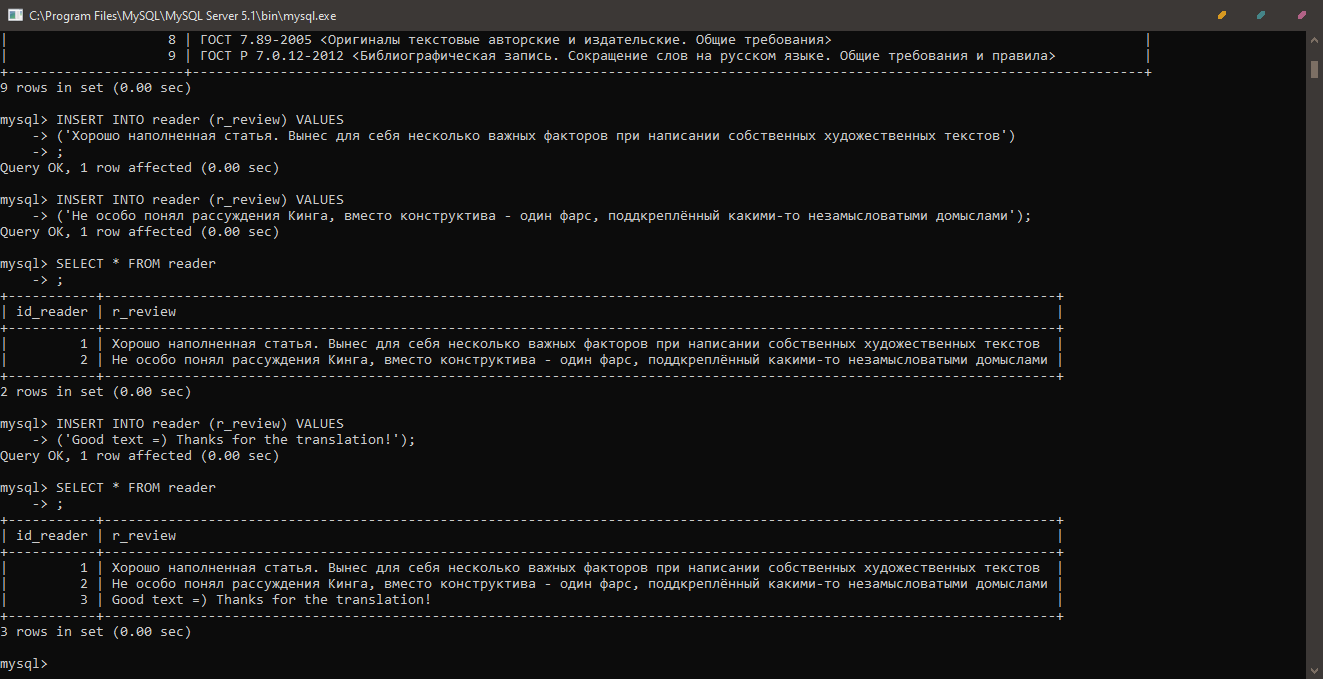


Рис. 14 Заполнение и вывод элементов таблицы reader

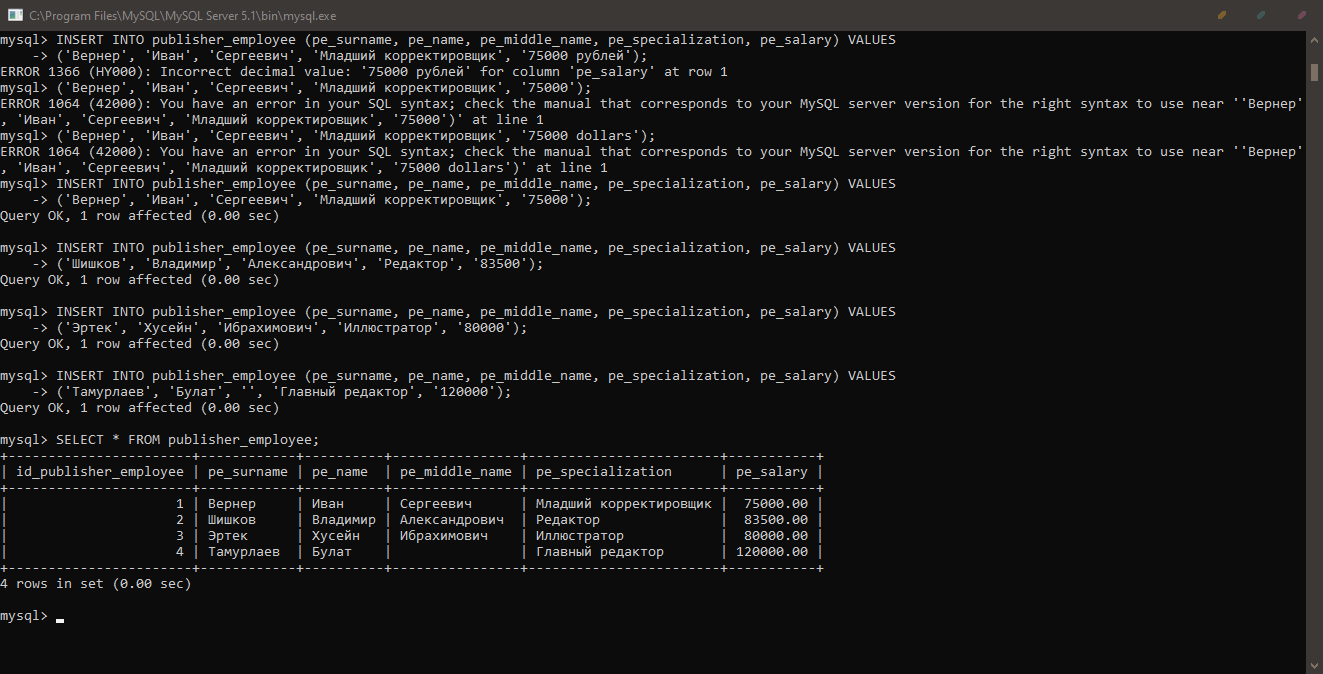


Рис. 15 - Заполнение и вывод элементов таблицы publisher\_employee

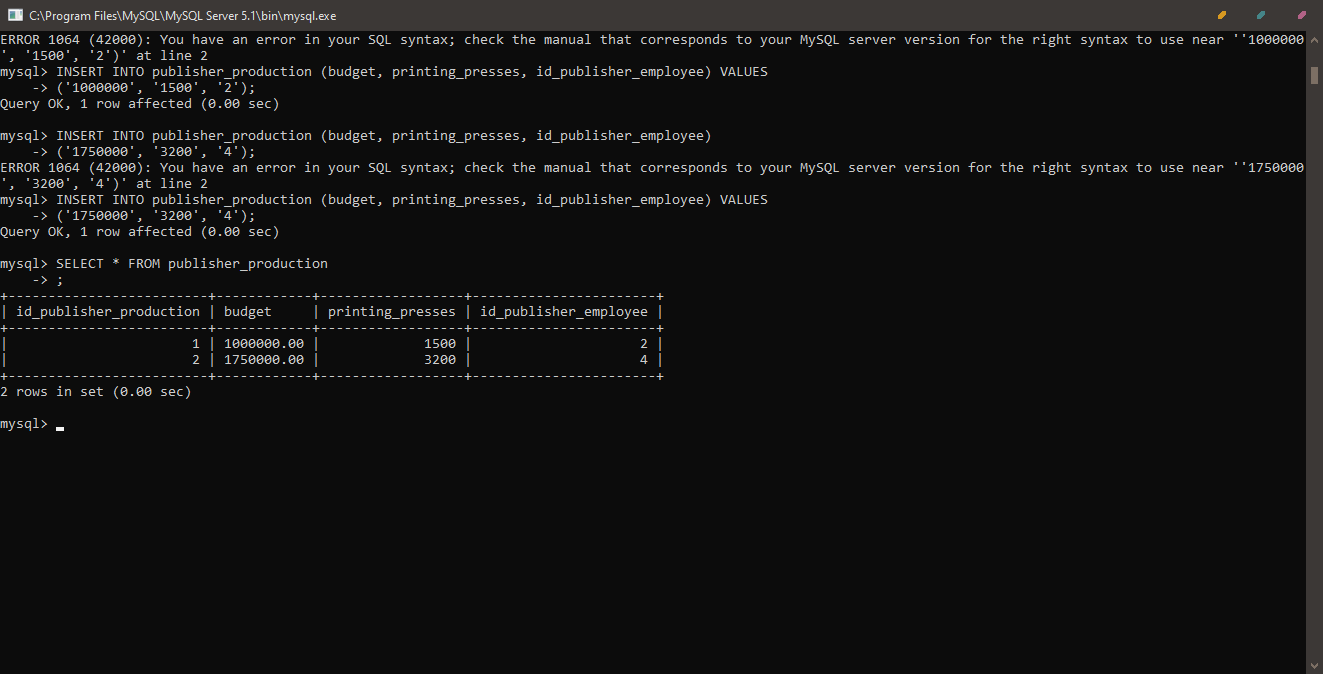


Рис. 16 - Заполнение и вывод элементов таблицы publisher\_production

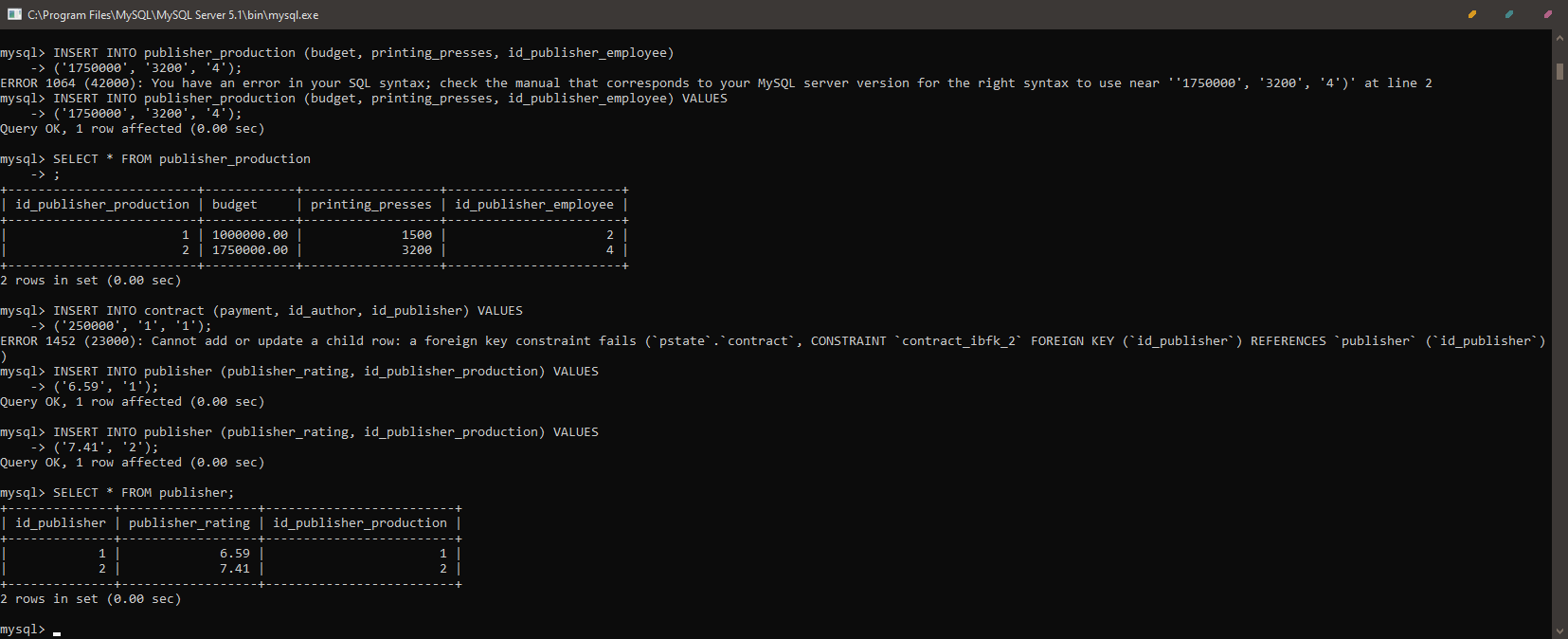


Рис. 17 - Заполнение и вывод элементов таблицы publisher



Рис. 18- Заполнение и вывод элементов таблицы contract

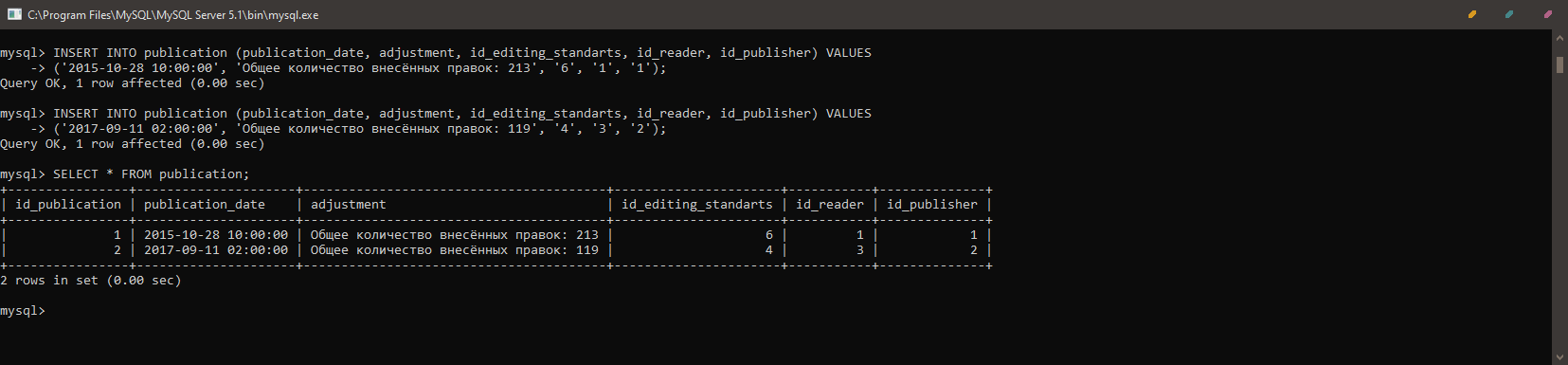


Рис. 19 - Заполнение и вывод элементов таблицы publication

**Работа с запросами в базе данных Pstate**

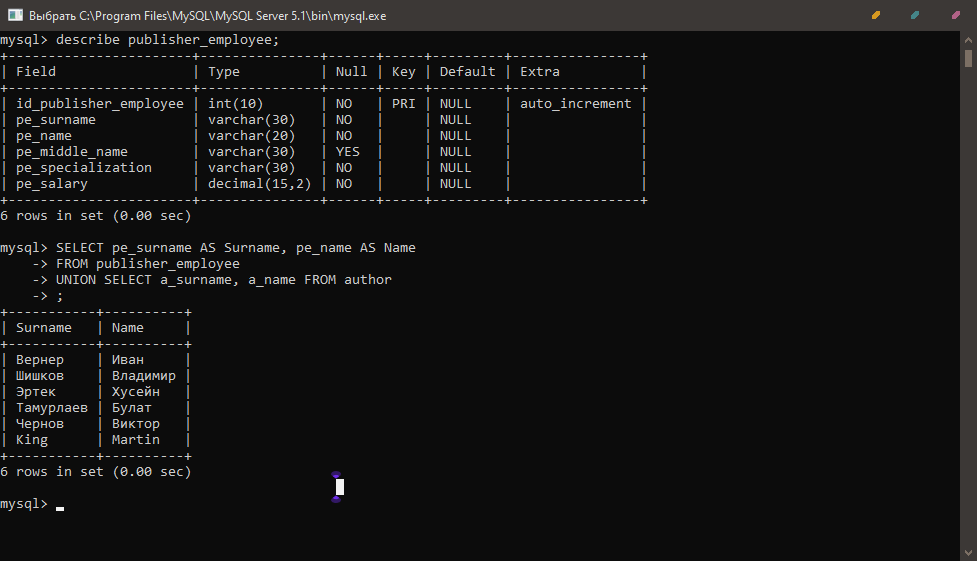
****

Рис. 20 - Объединение с использованием UNION

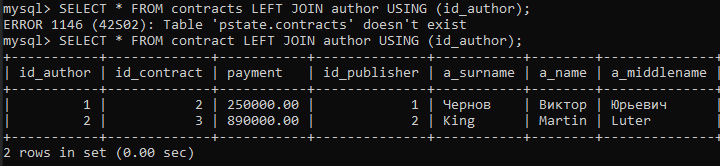


Рис. 21 - Выборка с использованием LEFT JOIN часть 1

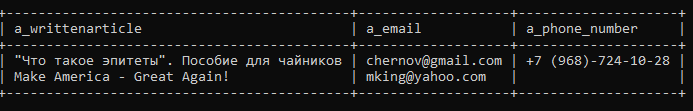


Рис. 22 - Выборка с использованием LEFT JOIN часть 2

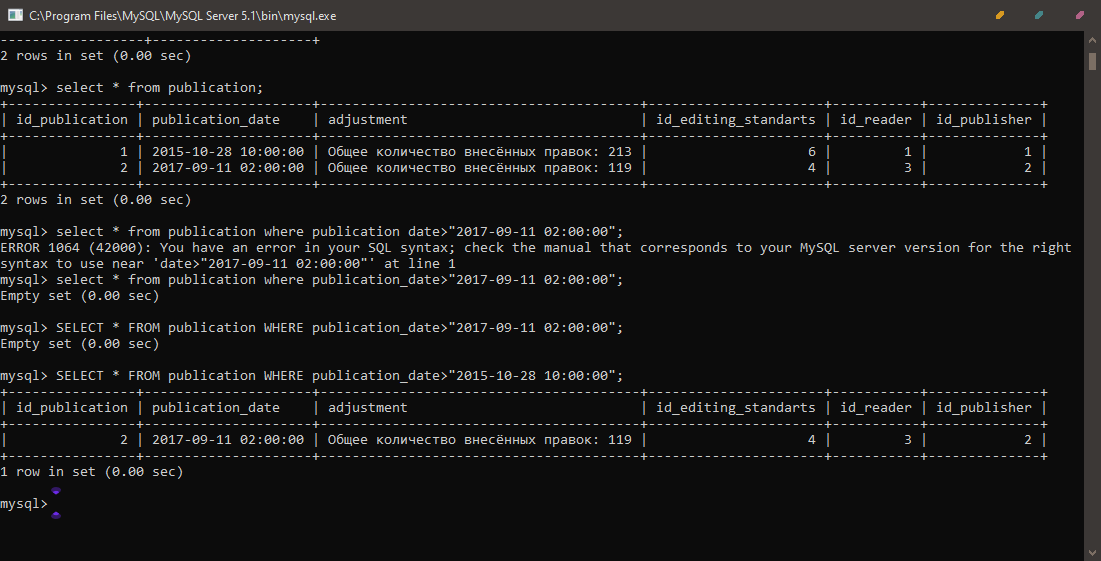


Рис. 23 – Выборка данных по параметру publication\_date

**Построение модели с помощью оболочки MySQL Workbench**

Мною была построена модель в СУБД Workbench (Рисунок 24)

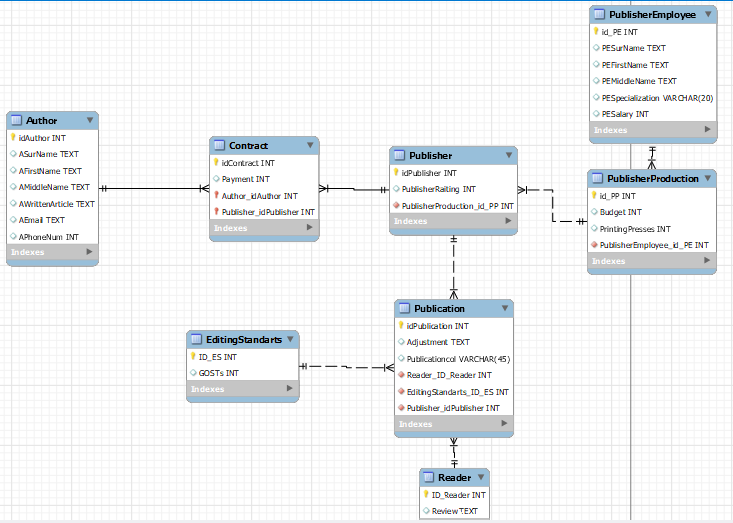


Рисунок 24 – Построенная модель

Перенос Базы Данных на другой сервер

Использовав встроенные инструменты MYSQL Workbench, был получен MySql скрипт:

-- MySQL Script generated by MySQL Workbench

-- Mon Nov 29 17:33:13 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 mydb

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

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

-- Schema mydb

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

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

USE `mydb` ;

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

-- Table `mydb`.`Author`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Author` (

`idAuthor` INT NOT NULL,

`ASurName` TEXT NULL,

`AFirstName` TEXT NULL,

`AMiddleName` TEXT NULL,

`AWrittenArticle` TEXT NULL,

`AEmail` TEXT NULL,

`APhoneNum` INT NULL,

PRIMARY KEY (`idAuthor`))

ENGINE = InnoDB;

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

-- Table `mydb`.`PublisherEmployee`

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

CREATE TABLE IF NOT EXISTS `mydb`.`PublisherEmployee` (

`id\_PE` INT NOT NULL,

`PESurName` TEXT NULL,

`PEFirstName` TEXT NULL,

`PEMiddleName` TEXT NULL,

`PESpecialization` VARCHAR(20) NULL,

`PESalary` INT NULL,

PRIMARY KEY (`id\_PE`))

ENGINE = InnoDB;

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

-- Table `mydb`.`PublisherProduction`

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

CREATE TABLE IF NOT EXISTS `mydb`.`PublisherProduction` (

`id\_PP` INT NOT NULL,

`Budget` INT NULL,

`PrintingPresses` INT NULL,

`PublisherEmployee\_id\_PE` INT NOT NULL,

PRIMARY KEY (`id\_PP`),

INDEX `fk\_PublisherProduction\_PublisherEmployee1\_idx` (`PublisherEmployee\_id\_PE` ASC) VISIBLE,

CONSTRAINT `fk\_PublisherProduction\_PublisherEmployee1`

FOREIGN KEY (`PublisherEmployee\_id\_PE`)

REFERENCES `mydb`.`PublisherEmployee` (`id\_PE`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `mydb`.`Publisher`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Publisher` (

`idPublisher` INT NOT NULL,

`PublisherRaiting` INT NULL,

`PublisherProduction\_id\_PP` INT NOT NULL,

PRIMARY KEY (`idPublisher`),

INDEX `fk\_Publisher\_PublisherProduction1\_idx` (`PublisherProduction\_id\_PP` ASC) VISIBLE,

CONSTRAINT `fk\_Publisher\_PublisherProduction1`

FOREIGN KEY (`PublisherProduction\_id\_PP`)

REFERENCES `mydb`.`PublisherProduction` (`id\_PP`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `mydb`.`Contract`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Contract` (

`idContract` INT NOT NULL,

`Payment` INT NULL,

`Author\_idAuthor` INT NOT NULL,

`Publisher\_idPublisher` INT NOT NULL,

PRIMARY KEY (`idContract`, `Author\_idAuthor`, `Publisher\_idPublisher`),

INDEX `fk\_Contract\_Author\_idx` (`Author\_idAuthor` ASC) VISIBLE,

INDEX `fk\_Contract\_Publisher1\_idx` (`Publisher\_idPublisher` ASC) VISIBLE,

CONSTRAINT `fk\_Contract\_Author`

FOREIGN KEY (`Author\_idAuthor`)

REFERENCES `mydb`.`Author` (`idAuthor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Contract\_Publisher1`

FOREIGN KEY (`Publisher\_idPublisher`)

REFERENCES `mydb`.`Publisher` (`idPublisher`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `mydb`.`Reader`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Reader` (

`ID\_Reader` INT NOT NULL,

`Review` TEXT NULL,

PRIMARY KEY (`ID\_Reader`))

ENGINE = InnoDB;

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

-- Table `mydb`.`EditingStandarts`

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

CREATE TABLE IF NOT EXISTS `mydb`.`EditingStandarts` (

`ID\_ES` INT NOT NULL,

`GOSTs` INT NULL,

PRIMARY KEY (`ID\_ES`))

ENGINE = InnoDB;

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

-- Table `mydb`.`Publication`

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

CREATE TABLE IF NOT EXISTS `mydb`.`Publication` (

`idPublication` INT NOT NULL,

`Adjustment` TEXT NULL,

`Publicationcol` VARCHAR(45) NULL,

`Reader\_ID\_Reader` INT NOT NULL,

`EditingStandarts\_ID\_ES` INT NOT NULL,

`Publisher\_idPublisher` INT NOT NULL,

PRIMARY KEY (`idPublication`),

INDEX `fk\_Publication\_Reader1\_idx` (`Reader\_ID\_Reader` ASC) VISIBLE,

INDEX `fk\_Publication\_EditingStandarts1\_idx` (`EditingStandarts\_ID\_ES` ASC) VISIBLE,

INDEX `fk\_Publication\_Publisher1\_idx` (`Publisher\_idPublisher` ASC) VISIBLE,

CONSTRAINT `fk\_Publication\_Reader1`

FOREIGN KEY (`Reader\_ID\_Reader`)

REFERENCES `mydb`.`Reader` (`ID\_Reader`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Publication\_EditingStandarts1`

FOREIGN KEY (`EditingStandarts\_ID\_ES`)

REFERENCES `mydb`.`EditingStandarts` (`ID\_ES`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Publication\_Publisher1`

FOREIGN KEY (`Publisher\_idPublisher`)

REFERENCES `mydb`.`Publisher` (`idPublisher`)

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**

Данные в таблице author соответствуют авторам, которые работают с издательством (Рисунок 25)

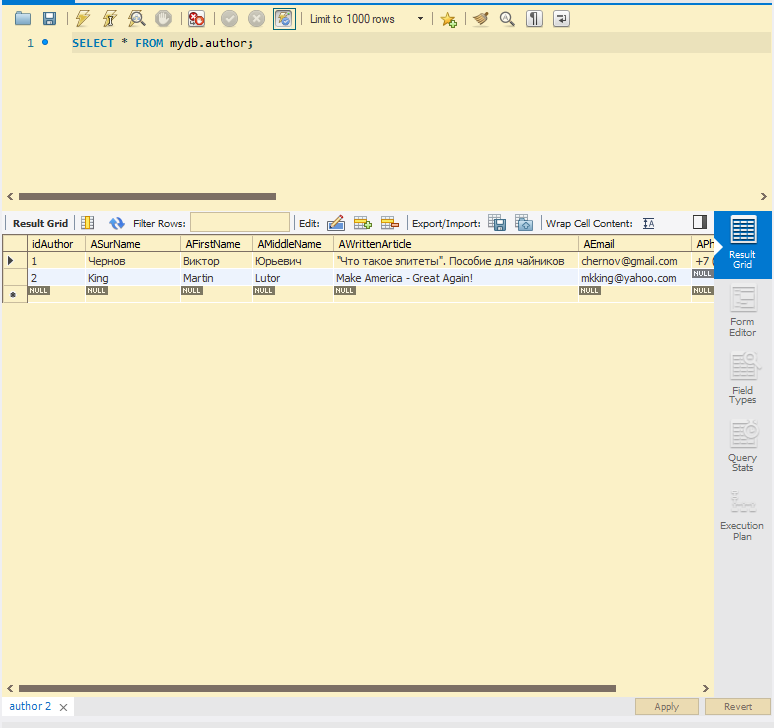
****

Рисунок 25 – Внесенные данные

Данные в таблице reader соответствуют читателям, которые оставили отзыв о статьях (Рисунок 26)

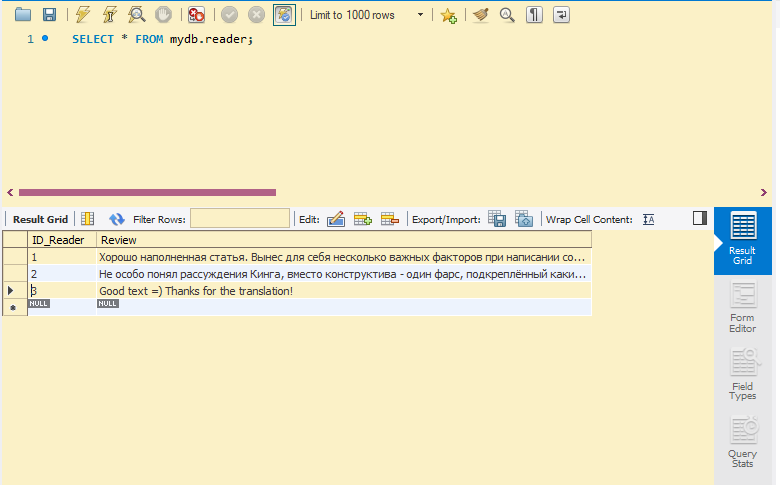


Рисунок 26 – Внесенные данные

Данные в таблице editingstandarts соответствуют ГОСТ-ам, по которым работает издательство (Рисунок 27)

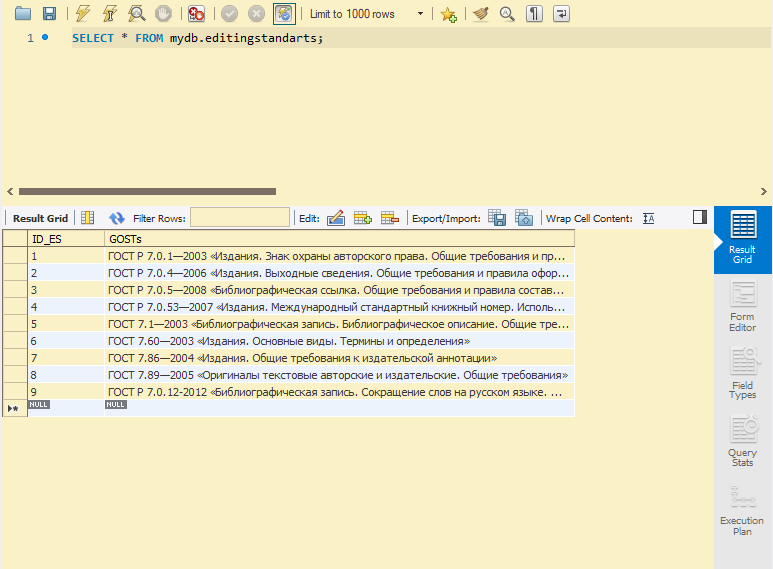


Рисунок 27 – Внесенные данные

Данные в таблице publisheremployee отражают информацию о сотрудниках (Рисунок 28)

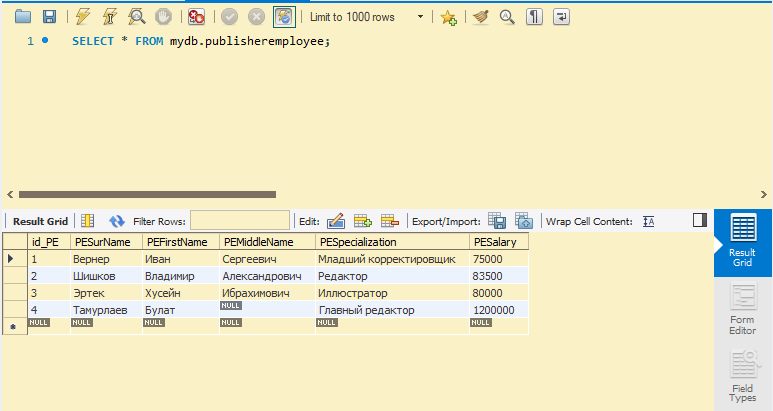


Рисунок 28 – Внесенные данные

Данные в таблице publisherproduction соответствуют информации о производстве статей (Рисунок 29)

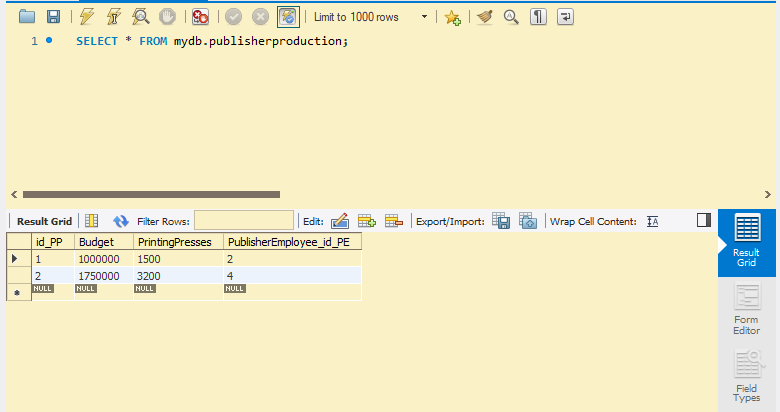


Рисунок 29 – Внесенные данные

Данные в таблице publisher отражают рейтинг статьи в издательстве (Рисунок 30)

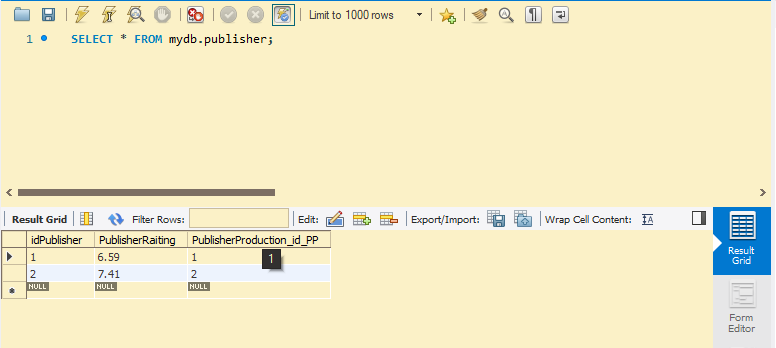


Рисунок 30 – Внесенные данные

Данные в таблице contract соответствуют деталям контрактов издательства с автором (Рисунок 31)

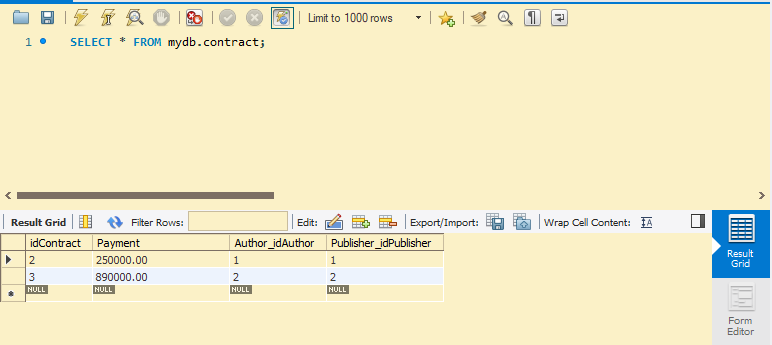


Рисунок 31 – Внесенные данные

Данные в таблице publication отражают информацию о дате публикации и о количестве общих корректировок статей (Рисунок 32)

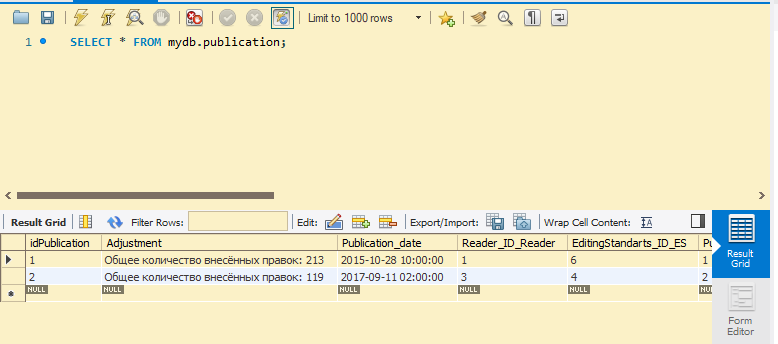


Рисунок 32 – Внесенные данные

**Выборка данных. Оператор SELECT.**

Операция проекции (Рисунок 33)

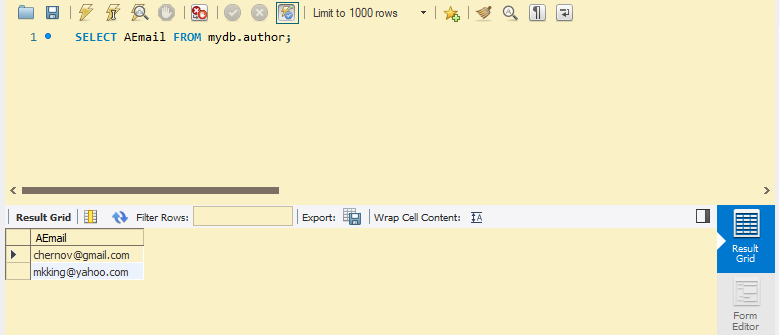


Рисунок 33 – Проекция

Операция селекции (Рисунок 34)

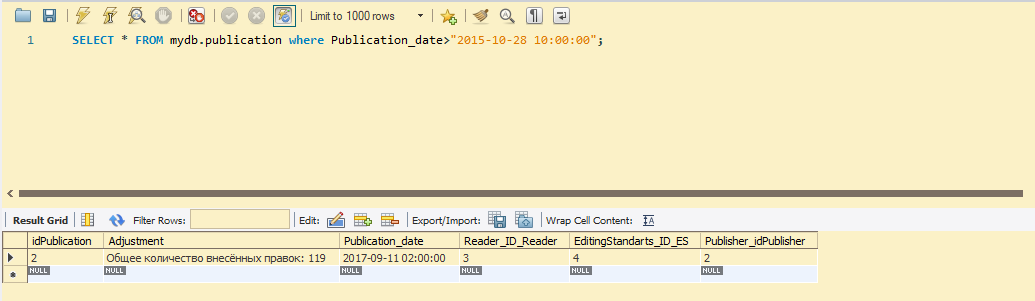


Рисунок 34 – Селекция

Операция соединения (Рисунок 35)

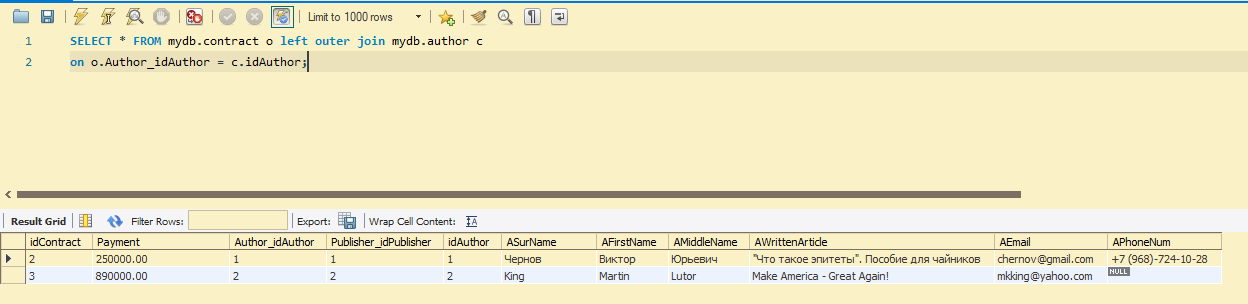


Рисунок 35 – Соединение

Операция объединения (Рисунок 35)

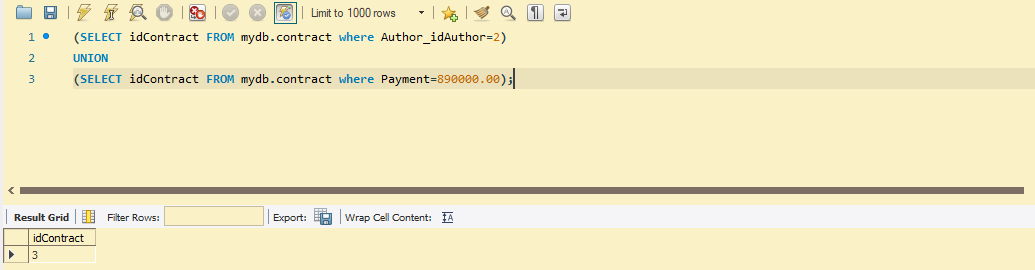


Рисунок 35 – Объединение

**Создание триггеров.**

Был создан триггер, который при определённом бюджете на производство уведомляет о необходимости увеличить процент оплаты автору (бюджет исходит от уровня известности автора).

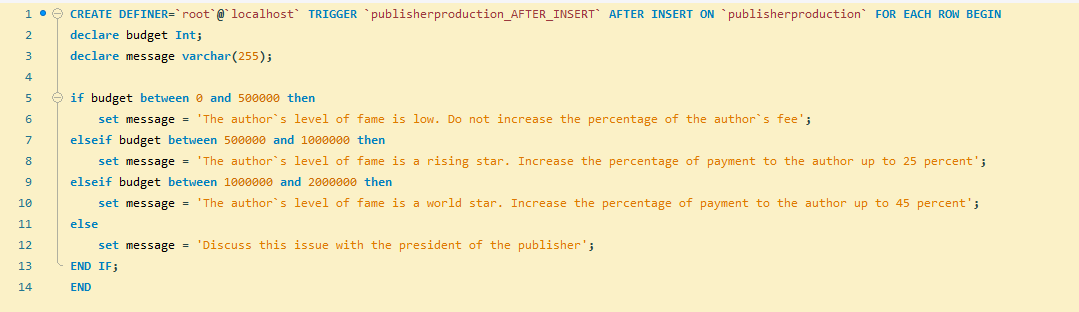


Рисунок 36 – Код триггера

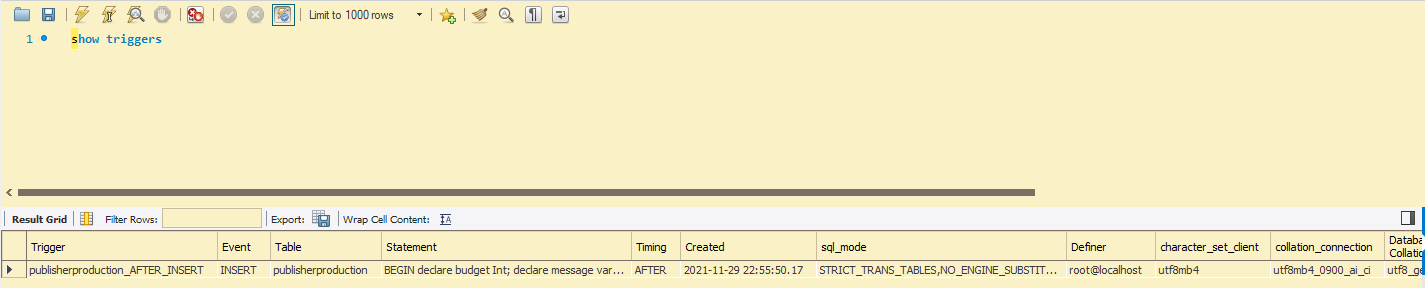


Рисунок 37 – Хранение триггера

**Создание хранимых процедур**

Вызов всех дней публикаций статей из «publication» при помощи хранимой процедуры.

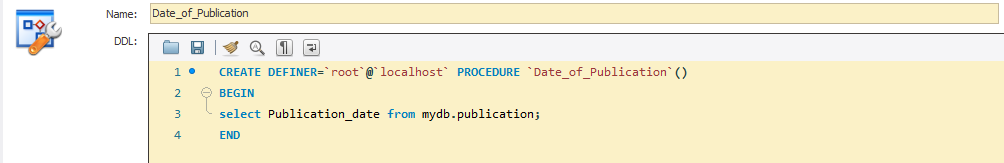


Рисунок 38 – Код процедуры

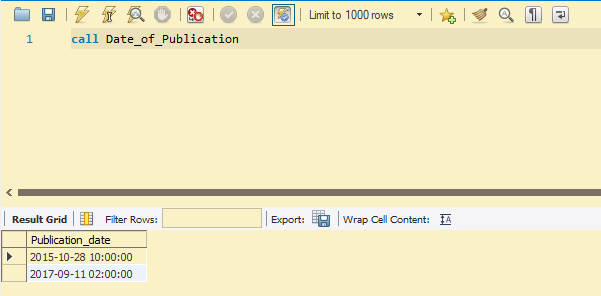


Рисунок 39 – Вызов процедуры

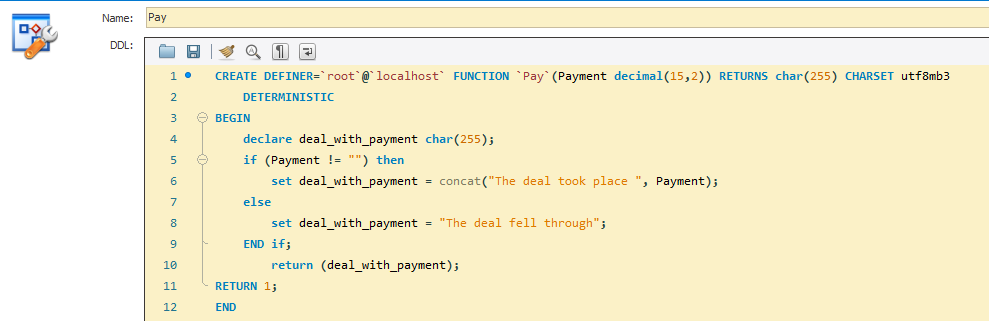


Рисунок 40 – Код функции

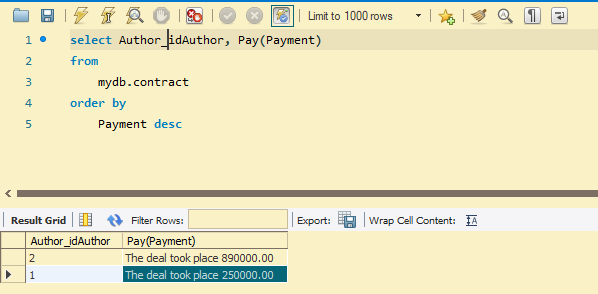


Рисунок 41 – Вызов функции

**Резервное копирование**

Здесь было выполнено копирование базы данных mydb (Pstate):

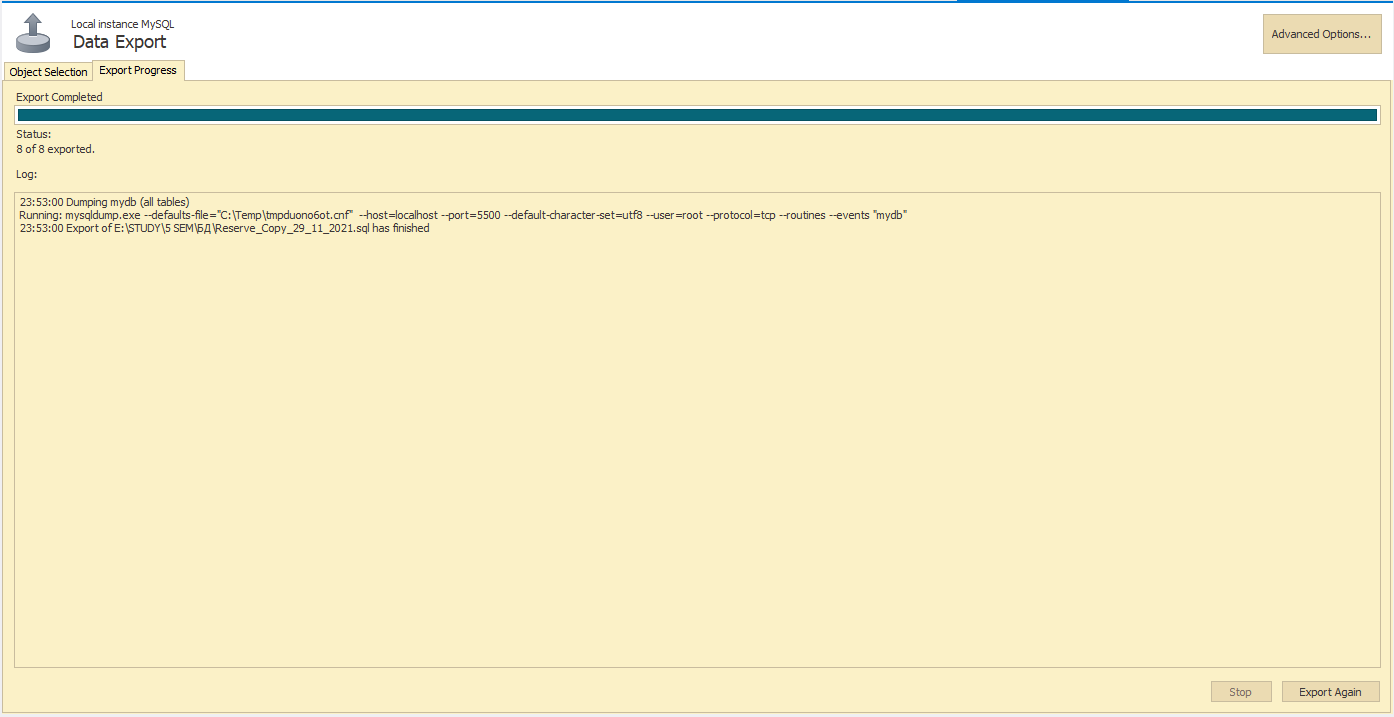


Рисунок 42 – Уведомление об успешном создании резервной копии

**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.