|  |
| --- |
| 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»**

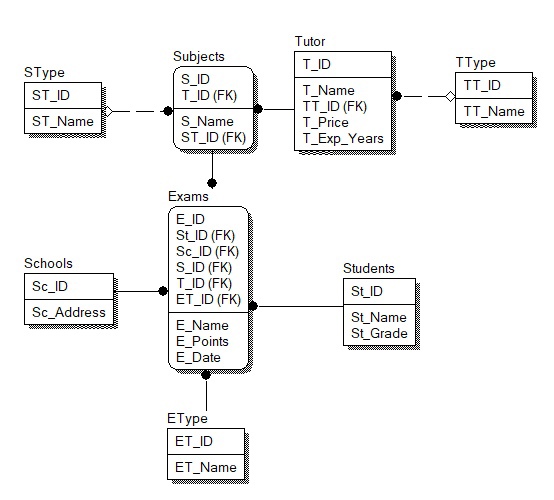


Рис. 1 ‒ Логическая модель данных на тему «Подготовка и сдача экзаменов»

**Описание**

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

Examinations – 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 «Подготовка и сдача экзаменов»**

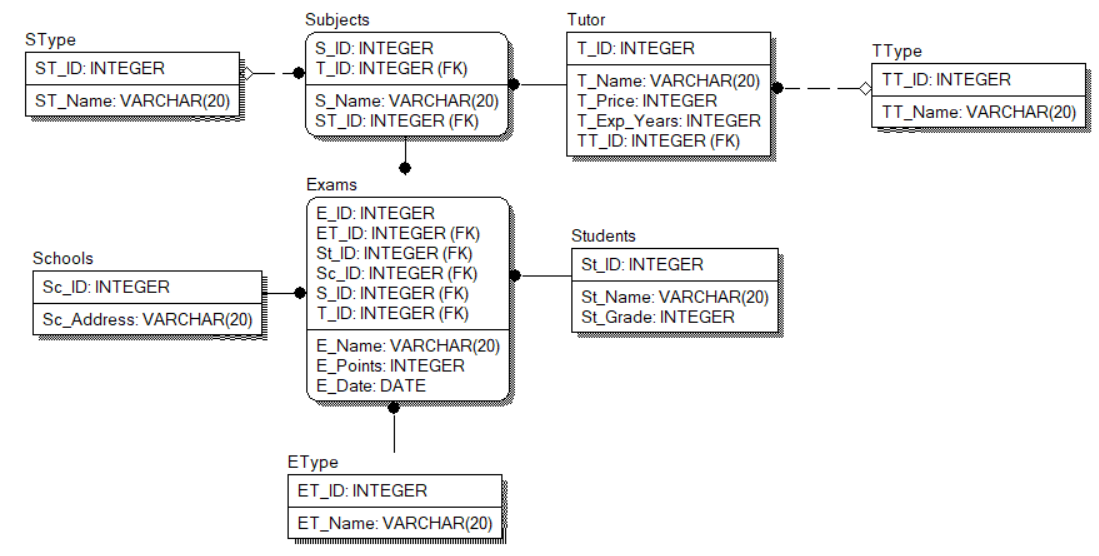
****

Рис. 2 ‒ Физическая модель данных на тему «Подготовка и сдача экзаменов»

**Создание своей базы данных в MySQL CommandLine**

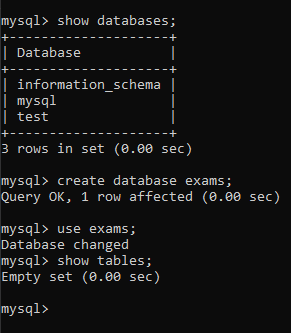


Рис. 3 – Создание базы данных exams

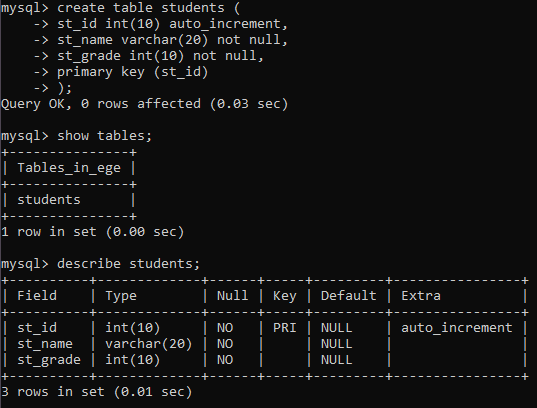


Рис. 4 – Создание таблицы students

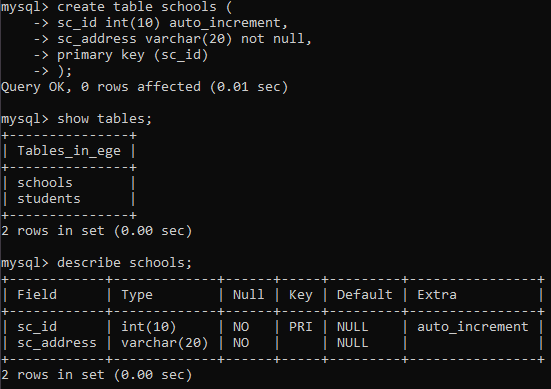


Рис. 5 – Создание таблицы schools

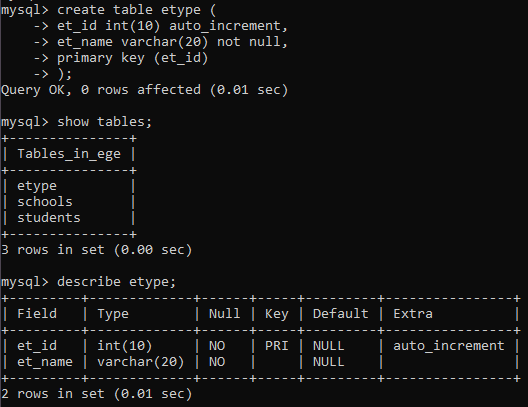


Рис. 6 – Создание таблицы etype

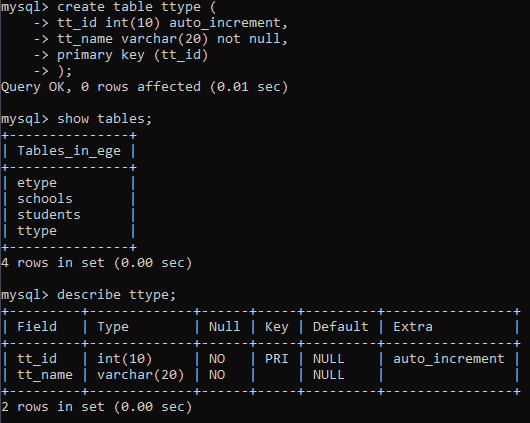


Рис. 7 – Создание таблицы ttype

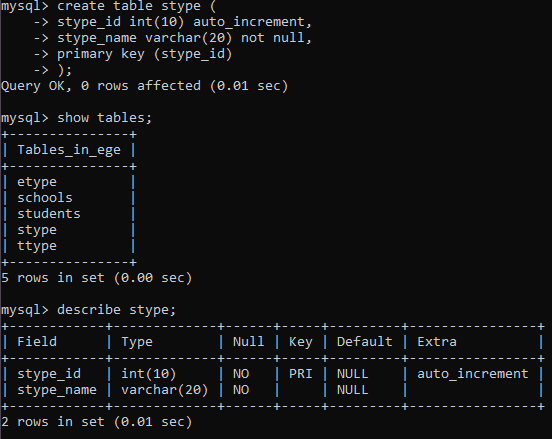


Рис. 8 – Создание таблицы stype

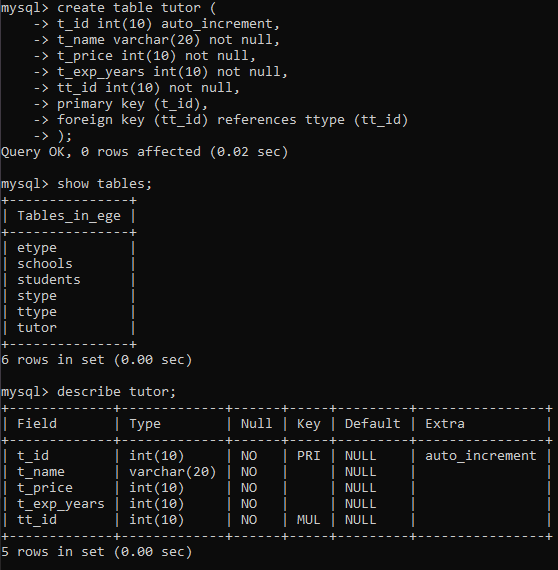


Рис. 9 – Создание таблицы tutor

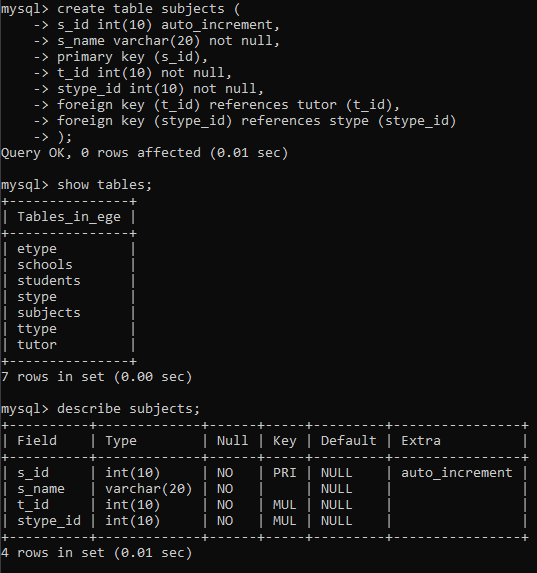


Рис. 10 – Создание таблицы subjects

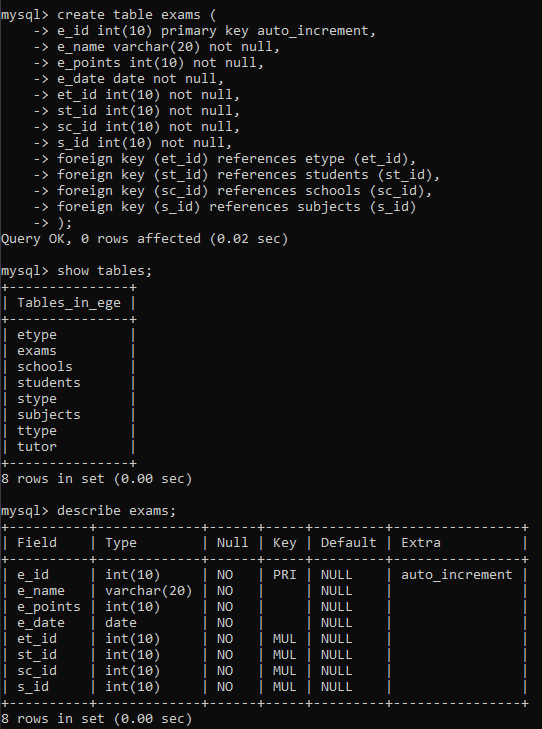


Рис. 11 – Создание ключевой таблицы exams

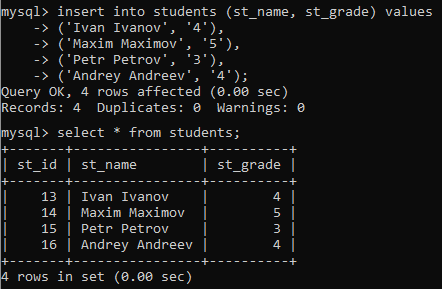


Рис. 12 – Заполнение и вывод таблицы students

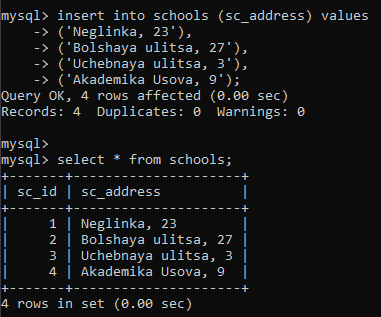


Рис. 13 – Заполнение и вывод таблицы schools

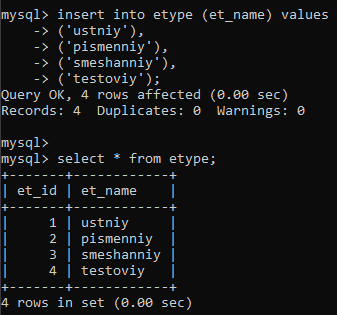


Рис. 14 – Заполнение и вывод таблицы etype

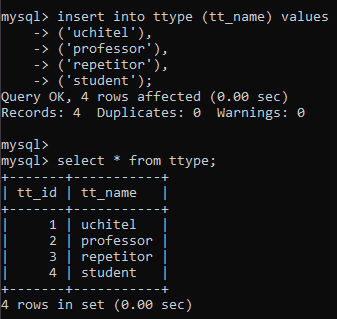


Рис. 15 – Заполнение и вывод таблицы ttype

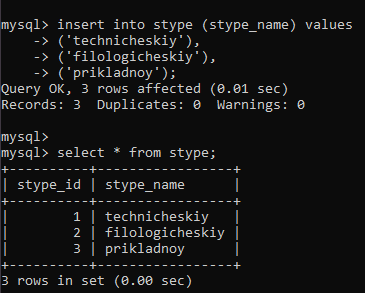


Рис. 16 – Заполнение и вывод таблицы stype

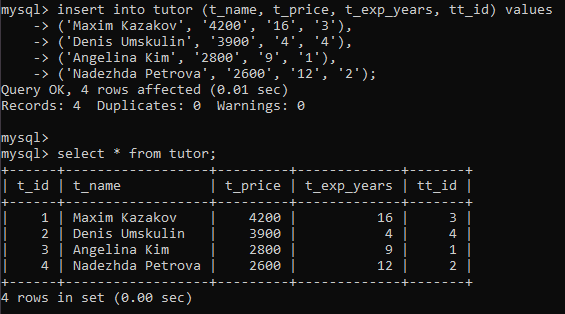


Рис. 17 – Заполнение и вывод таблицы tutor

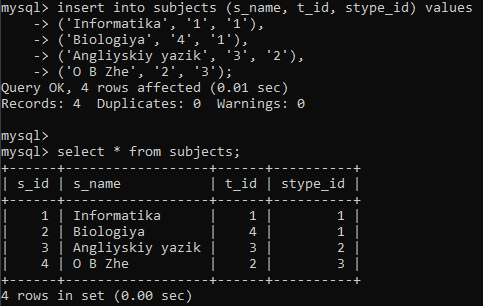


Рис. 18 – Заполнение и вывод таблицы subjects

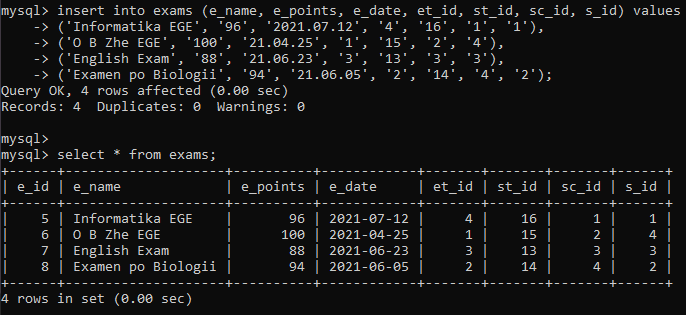


Рис. 19 – Заполнение и вывод таблицы exams

**Работа с запросами в базе данных EGE**

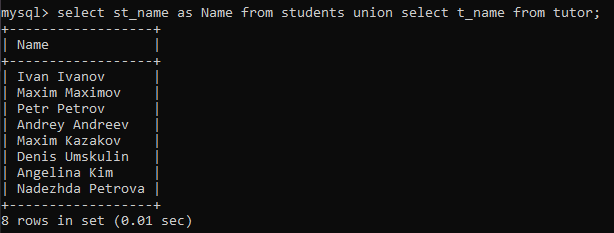
****

Рис. 20 – Объединение с использованием union

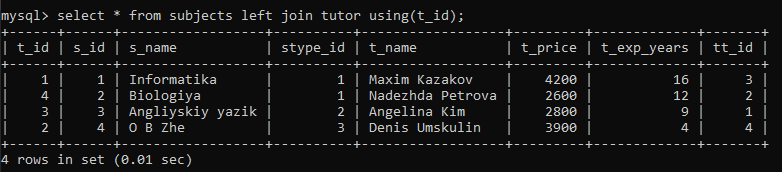


Рис. 21 – Выборка с использованием left join

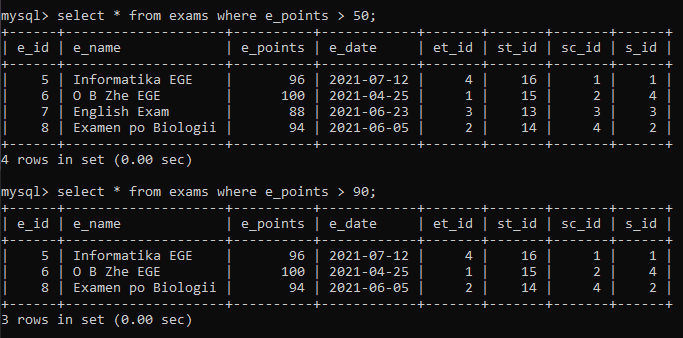


Рис. 22 – Выборка данных по параметру e\_points

**Построение модели с помощью оболочки MySQL Workbench**

Мною была построена модель в СУБД Workbench (Рисунок 24):

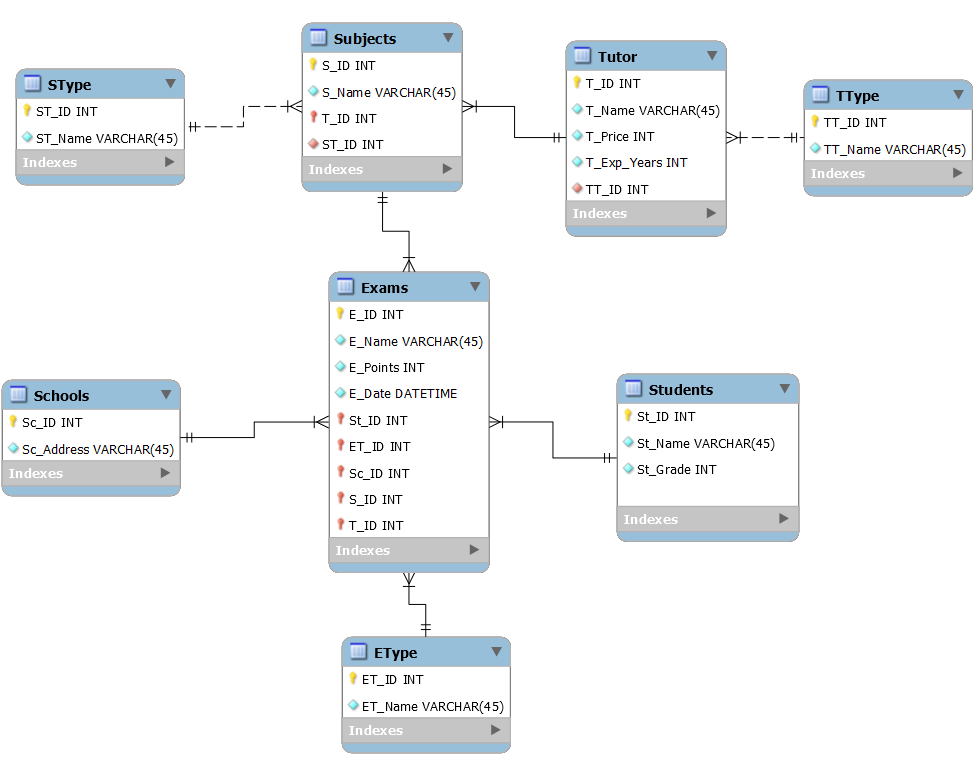


Рисунок 23 – Построенная модель в СУБД MySQL Workbench

Перенос Базы Данных на другой сервер

Использовав встроенные инструменты MYSQL Workbench, был получен MySql скрипт:

-- MySQL Script generated by MySQL Workbench

-- Sun Dec 12 18:56:56 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 ege

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

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

-- Schema ege

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

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

USE `ege` ;

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

-- Table `ege`.`Students`

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

CREATE TABLE IF NOT EXISTS `ege`.`Students` (

`St\_ID` INT NOT NULL,

`St\_Name` VARCHAR(45) NOT NULL,

`St\_Grade` INT NOT NULL,

PRIMARY KEY (`St\_ID`),

UNIQUE INDEX `St\_ID\_UNIQUE` (`St\_ID` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `ege`.`EType`

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

CREATE TABLE IF NOT EXISTS `ege`.`EType` (

`ET\_ID` INT NOT NULL,

`ET\_Name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ET\_ID`),

UNIQUE INDEX `ET\_ID\_UNIQUE` (`ET\_ID` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `ege`.`Schools`

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

CREATE TABLE IF NOT EXISTS `ege`.`Schools` (

`Sc\_ID` INT NOT NULL,

`Sc\_Address` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Sc\_ID`),

UNIQUE INDEX `Sc\_ID\_UNIQUE` (`Sc\_ID` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `ege`.`TType`

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

CREATE TABLE IF NOT EXISTS `ege`.`TType` (

`TT\_ID` INT NOT NULL,

`TT\_Name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`TT\_ID`),

UNIQUE INDEX `TT\_ID\_UNIQUE` (`TT\_ID` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `ege`.`Tutor`

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

CREATE TABLE IF NOT EXISTS `ege`.`Tutor` (

`T\_ID` INT NOT NULL,

`T\_Name` VARCHAR(45) NOT NULL,

`T\_Price` INT NOT NULL,

`T\_Exp\_Years` INT NOT NULL,

`TT\_ID` INT NOT NULL,

PRIMARY KEY (`T\_ID`),

UNIQUE INDEX `T\_Name\_UNIQUE` (`T\_ID` ASC) VISIBLE,

INDEX `fk\_Tutor\_TType\_idx` (`TT\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_Tutor\_TType`

FOREIGN KEY (`TT\_ID`)

REFERENCES `ege`.`TType` (`TT\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ege`.`SType`

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

CREATE TABLE IF NOT EXISTS `ege`.`SType` (

`ST\_ID` INT NOT NULL,

`ST\_Name` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ST\_ID`),

UNIQUE INDEX `ST\_ID\_UNIQUE` (`ST\_ID` ASC) VISIBLE)

ENGINE = InnoDB;

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

-- Table `ege`.`Subjects`

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

CREATE TABLE IF NOT EXISTS `ege`.`Subjects` (

`S\_ID` INT NOT NULL,

`S\_Name` VARCHAR(45) NOT NULL,

`T\_ID` INT NOT NULL,

`ST\_ID` INT NOT NULL,

PRIMARY KEY (`S\_ID`, `T\_ID`),

UNIQUE INDEX `S\_ID\_UNIQUE` (`S\_ID` ASC) VISIBLE,

INDEX `fk\_Subjects\_Tutor1\_idx` (`T\_ID` ASC) VISIBLE,

INDEX `fk\_Subjects\_SType1\_idx` (`ST\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_Subjects\_Tutor1`

FOREIGN KEY (`T\_ID`)

REFERENCES `ege`.`Tutor` (`T\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Subjects\_SType1`

FOREIGN KEY (`ST\_ID`)

REFERENCES `ege`.`SType` (`ST\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `ege`.`Exams`

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

CREATE TABLE IF NOT EXISTS `ege`.`Exams` (

`E\_ID` INT NOT NULL,

`E\_Name` VARCHAR(45) NOT NULL,

`E\_Points` INT NOT NULL,

`E\_Date` DATETIME NOT NULL,

`St\_ID` INT NOT NULL,

`ET\_ID` INT NOT NULL,

`Sc\_ID` INT NOT NULL,

`S\_ID` INT NOT NULL,

`T\_ID` INT NOT NULL,

PRIMARY KEY (`E\_ID`, `St\_ID`, `ET\_ID`, `Sc\_ID`, `S\_ID`, `T\_ID`),

UNIQUE INDEX `E\_ID\_UNIQUE` (`E\_ID` ASC) VISIBLE,

INDEX `fk\_Exams\_Students1\_idx` (`St\_ID` ASC) VISIBLE,

INDEX `fk\_Exams\_EType1\_idx` (`ET\_ID` ASC) VISIBLE,

INDEX `fk\_Exams\_Schools1\_idx` (`Sc\_ID` ASC) VISIBLE,

INDEX `fk\_Exams\_Subjects1\_idx` (`S\_ID` ASC, `T\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_Exams\_Students1`

FOREIGN KEY (`St\_ID`)

REFERENCES `ege`.`Students` (`St\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Exams\_EType1`

FOREIGN KEY (`ET\_ID`)

REFERENCES `ege`.`EType` (`ET\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Exams\_Schools1`

FOREIGN KEY (`Sc\_ID`)

REFERENCES `ege`.`Schools` (`Sc\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Exams\_Subjects1`

FOREIGN KEY (`S\_ID` , `T\_ID`)

REFERENCES `ege`.`Subjects` (`S\_ID` , `T\_ID`)

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 students соответствуют ученикам, которые сдают экзамены (Рисунок 24):

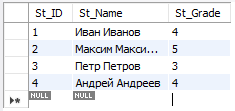
****

Рисунок 24 – Внесенные данные

Данные в таблице schools соответствуют школам, в которых происходят экзамены (Рисунок 25):

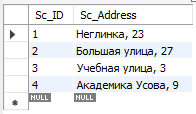


Рисунок 25 – Внесенные данные

Данные в таблице etypes соответствуют типам экзаменов, в каком формате они будут проходить (Рисунок 26):

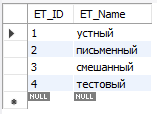


Рисунок 26 – Внесенные данные

Данные в таблице ttypes соответствуют типам преподавателей (Рисунок 27):

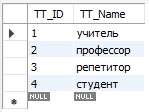


Рисунок 27 – Внесенные данные

Данные в таблице stype отражают информацию о типах предметов (Рисунок 28):

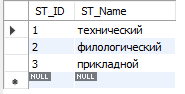


Рисунок 28 – Внесенные данные

Данные в таблице tutor соответствуют информации о преподавателях (Рисунок 29):

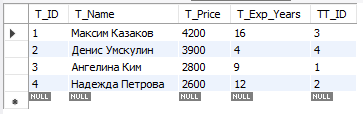


Рисунок 29 – Внесенные данные

Данные в таблице subjects отражают сдаваемые предметы на экзамене (Рисунок 30):

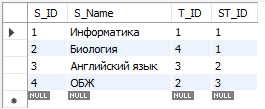


Рисунок 30 – Внесенные данные

Данные в таблице exams соответствуют экзаменам, которые проводились (Рисунок 31):

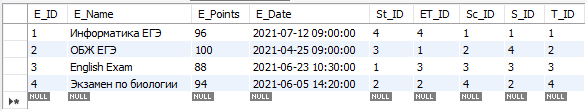


Рисунок 31 – Внесенные данные

**Выборка данных. Оператор SELECT.**

Операция проекции (Рисунок 32):

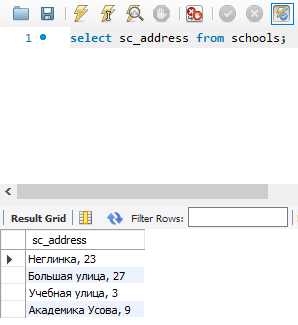


Рисунок 32 – Проекция

Операция селекции (Рисунок 33):

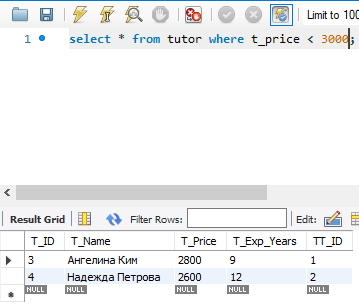


Рисунок 33 – Селекция

Операция соединения (Рисунок 34):

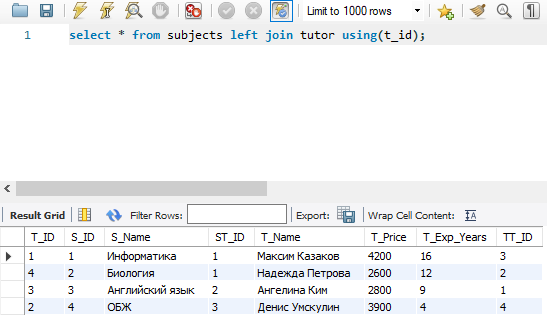


Рисунок 34 – Соединение

Операция объединения (Рисунок 35):

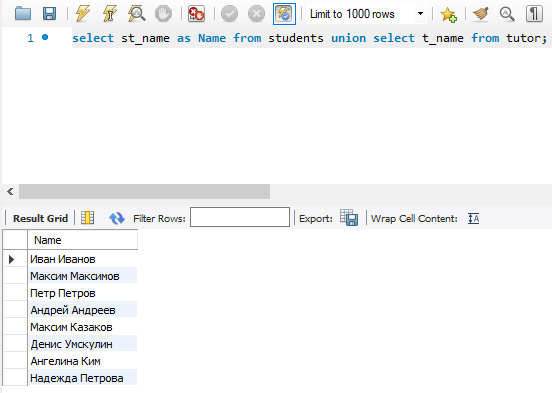


Рисунок 35 – Объединение

**Создание триггеров.**

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

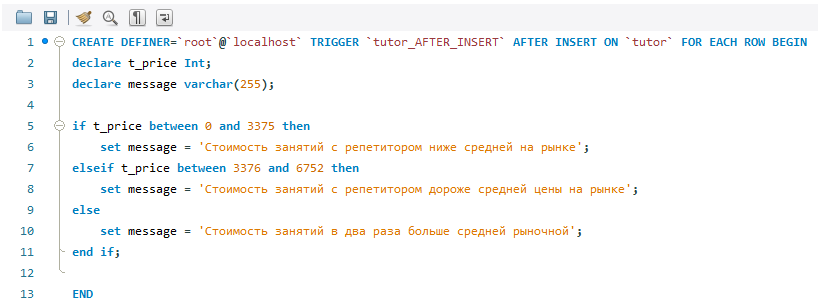


Рисунок 36 – Код триггера

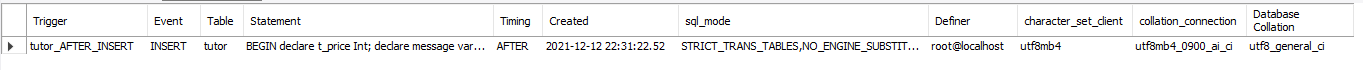


Рисунок 37 – Хранение триггера

**Создание хранимых процедур**

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

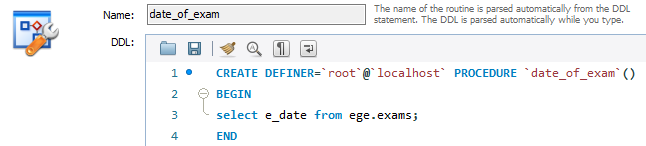


Рисунок 38 – Код процедуры

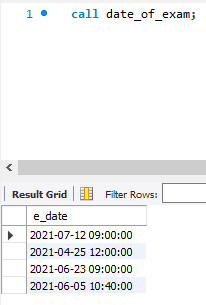


Рисунок 39 – Вызов процедуры

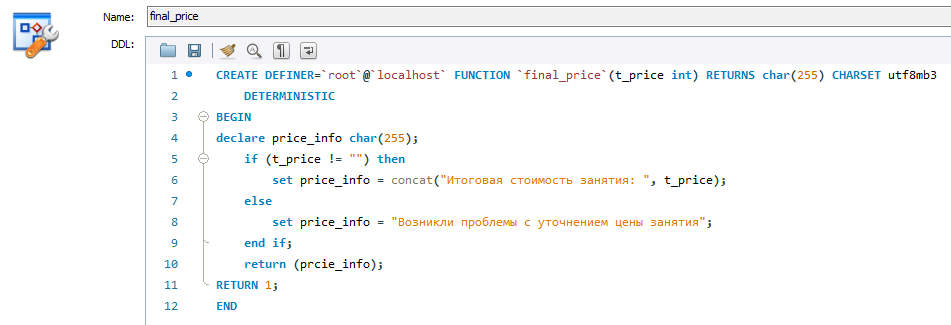


Рисунок 40 – Код функции

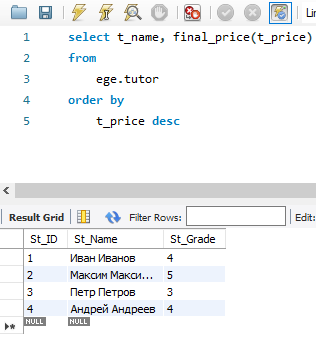


Рисунок 41 – Вызов функции

**Резервное копирование**

Здесь было выполнено копирование базы данных mysql (EGE):

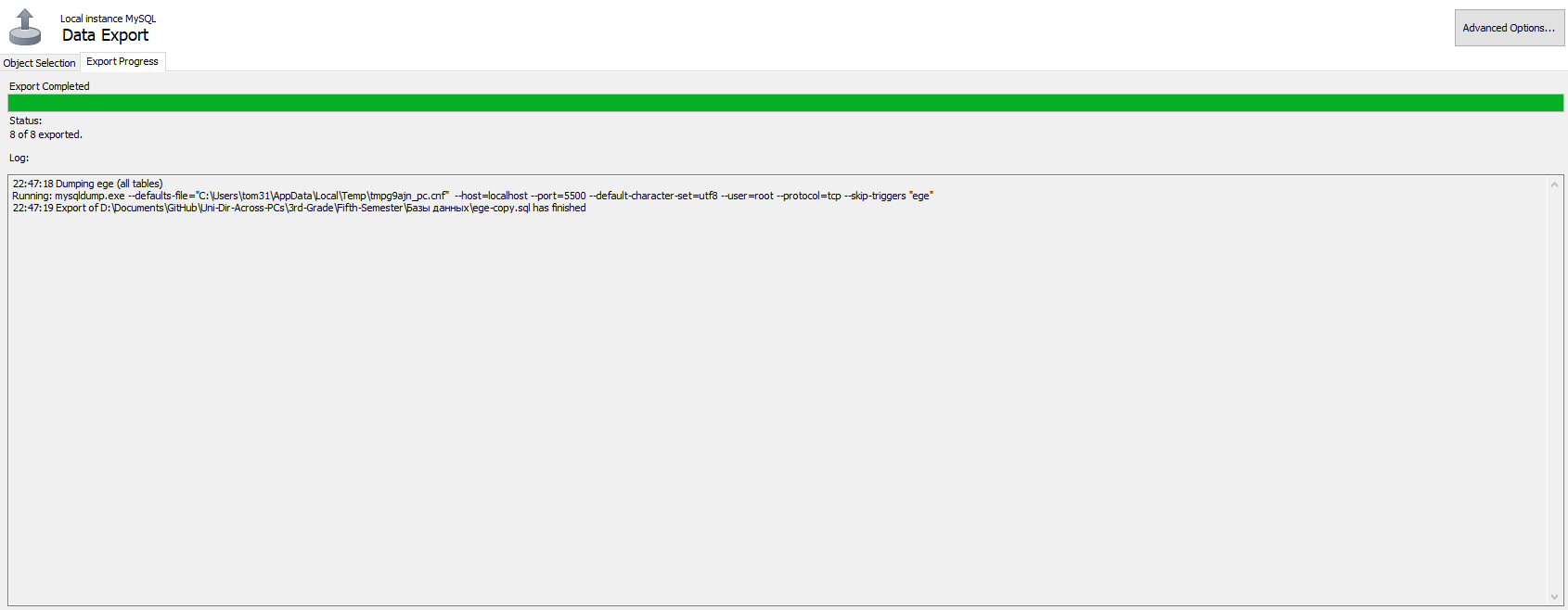


Рисунок 42 – Уведомление об успешном создании резервной копии

**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.