Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Системы автоматизированного проектирования»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Информационное обеспечение САПР»

на тему: «Экспорт базы данных в средство по администрированию БД «phpMyAdmin»

с помощью SQL-запросов»

Выполнили студенты

группы 16ВВ3:

Страхов А.В.

Самушкин А.Д.

Борисов Д.А.

Приняли:

Глотова Т.В.

Евсеева Ю.И.

Пенза 2018

**Цель работы:** импортировать созданную в «DBDesigner» инфологическую модель БД в «phpMyAdmin» с помощью SQL-запросов.

**Этапы выполнения лабораторной работы:**

1. Описание БД;
2. Экспорт БД из «DBDesigner»;
3. Редактирование полученного SQL-кода;
4. Экспорт в БД «phpMyAdmin»;
5. Проверка на целостность экспортированной БД.

**Этапы:**

1. **Описание БД:**

БД создана для поиска, хранения, добавления, редактирования, удаления информации о практических работах студентов.

Сущности:

* Universities- содержит наименования университетов, ректора, ID;
* Faculty- содержит наименования факультетов, деканов, ID;
* Speciality- содержит наименования специальностей, ID;
* Department- содержит наименования кафедр, ID;
* Groups- содержит наименования групп, ID;
* Student- содержит поля: имена, фамилии студентов, стипендия, ID;
* Professor- содержит имена и фамилии профессоров, ID;
* CourseWork- содержит название работы, оценку, ID;

Суммарно наша БД имеет 11 связей (описание которых не играет роли в данном учете, так как вся информация о связях при экспорте будет находится в объявление таблиц) и 9 таблиц, см. рис. 1, данная информация потребуется для проверки корректности экспорта в «phpMyAdmin».

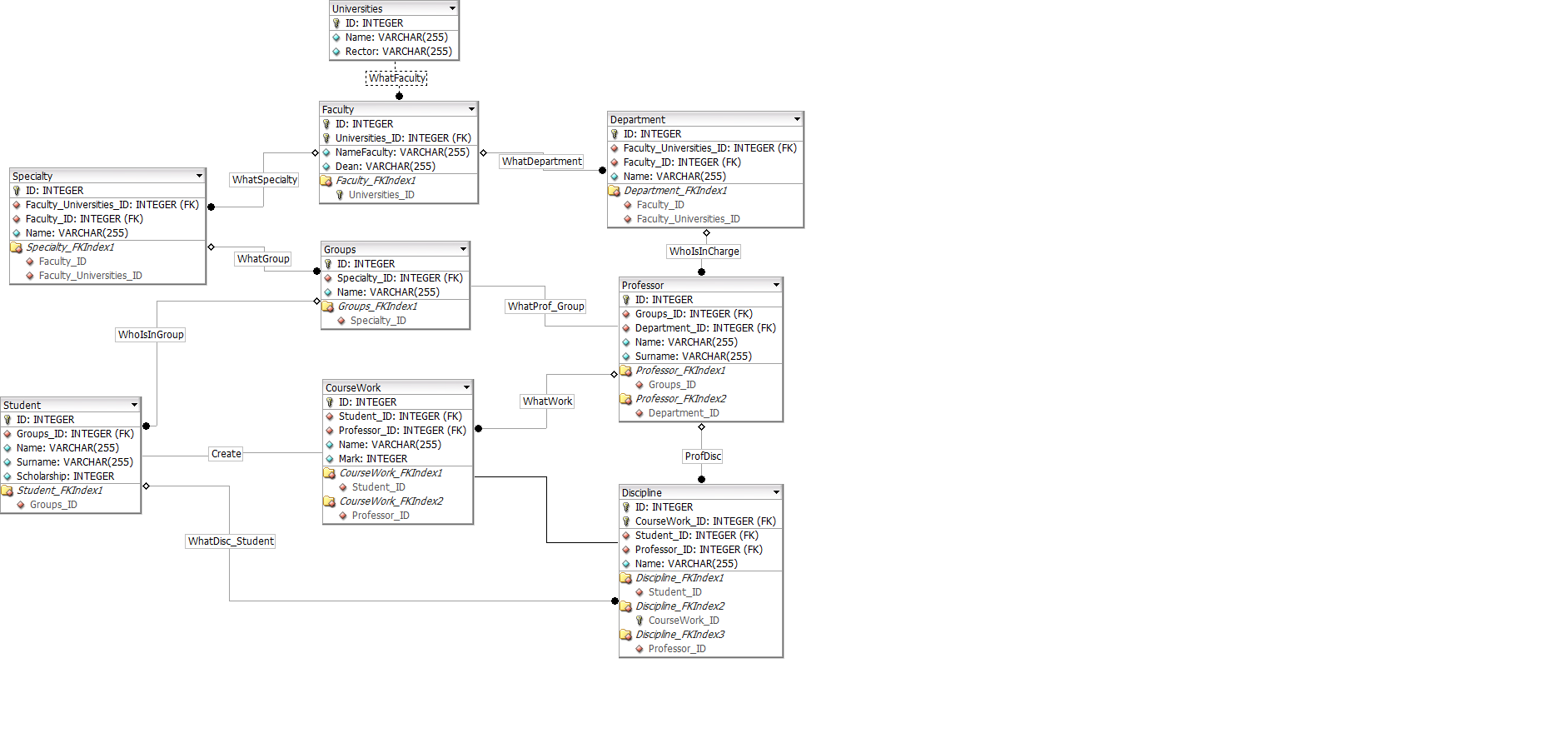


Рисунок 1 – Графическое представление БД «Курсовые работы»

1. **Экспорт БД из «DBDesigner»:**

Изначально наш проект имеет расширение «.xml», для экспорта в «phpMyAdmin» мы создаем на основе нашей БД SQL-скрипт, см. рис. 2.

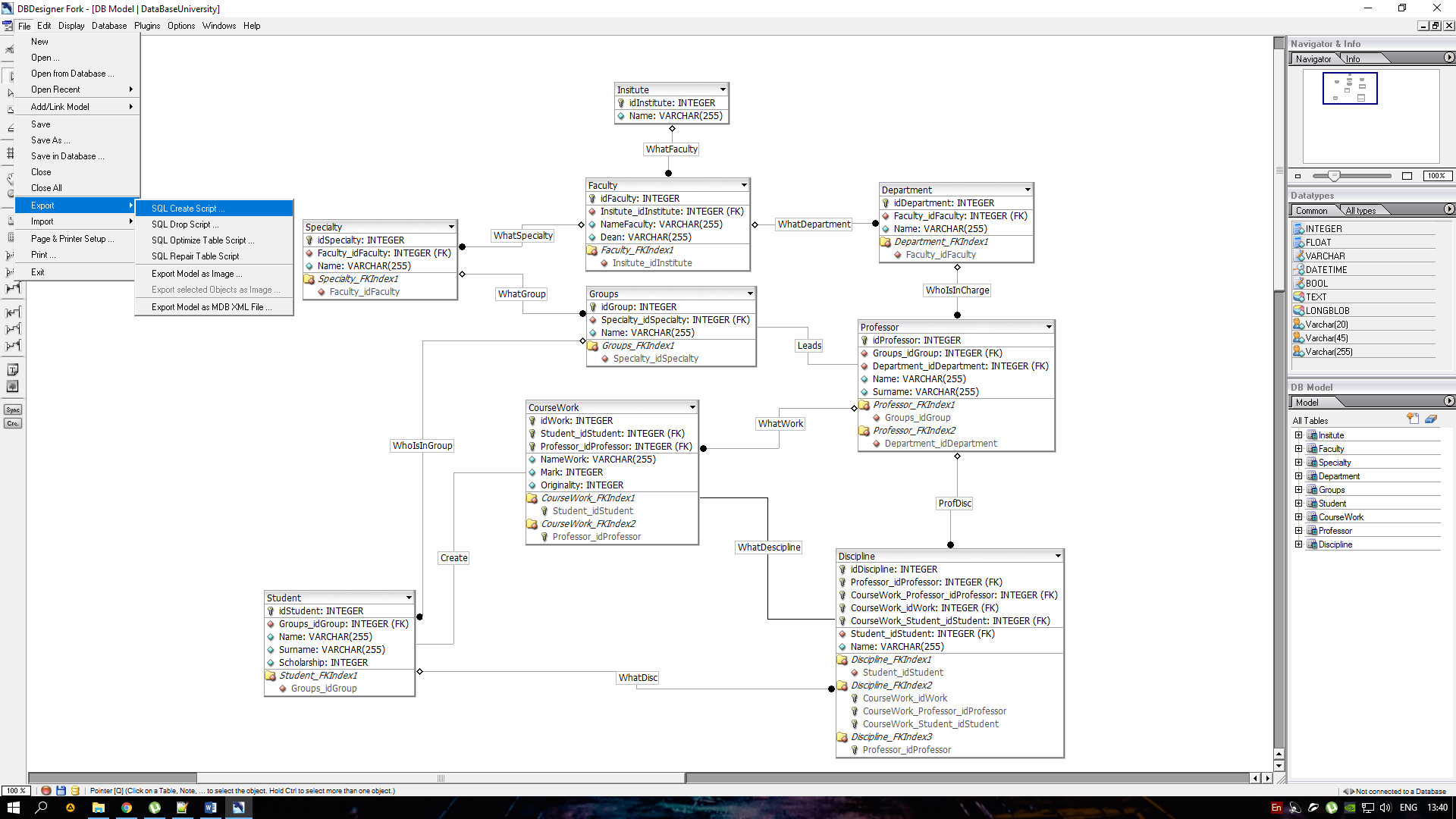


Рисунок 2 – Структура рассматриваемой БД

В параметрах экспорта выбираем конечного «получателя» нашего скрипта, в данном случае это SQL-Server, см. рис. 3.

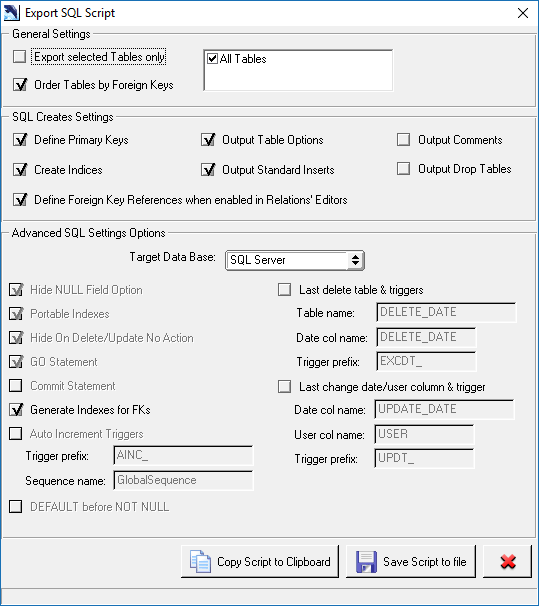


Рисунок 3 – Структура рассматриваемой БД

Полученный SQL-скрипт:

CREATE TABLE Universities (

ID INTEGER NOT NULL IDENTITY ,

Name VARCHAR(255) ,

Rector VARCHAR(255) ,

PRIMARY KEY(ID));

GO

CREATE TABLE Faculty (

ID INTEGER NOT NULL IDENTITY ,

Universities\_ID INTEGER NOT NULL ,

NameFaculty VARCHAR(255) ,

Dean VARCHAR(255) ,

PRIMARY KEY(ID, Universities\_ID) ,

FOREIGN KEY(Universities\_ID)

REFERENCES Universities(ID));

GO

CREATE INDEX Faculty\_FKIndex1 ON Faculty (Universities\_ID);

GO

CREATE INDEX IFK\_Rel\_13 ON Faculty (Universities\_ID);

GO

CREATE TABLE Specialty (

ID INTEGER NOT NULL IDENTITY ,

Faculty\_Universities\_ID INTEGER NOT NULL ,

Faculty\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Faculty\_ID, Faculty\_Universities\_ID)

REFERENCES Faculty(ID, Universities\_ID));

GO

CREATE INDEX Specialty\_FKIndex1 ON Specialty (Faculty\_ID, Faculty\_Universities\_ID);

GO

CREATE INDEX IFK\_WhatSpecialty ON Specialty (Faculty\_ID, Faculty\_Universities\_ID);

GO

CREATE TABLE Department (

ID INTEGER NOT NULL IDENTITY ,

Faculty\_Universities\_ID INTEGER NOT NULL ,

Faculty\_ID INTEGER NOT NULL ,

Name VARCHAR(255) NOT NULL ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Faculty\_ID, Faculty\_Universities\_ID)

REFERENCES Faculty(ID, Universities\_ID));

GO

CREATE INDEX Department\_FKIndex1 ON Department (Faculty\_ID, Faculty\_Universities\_ID);

GO

CREATE INDEX IFK\_WhatDepartment ON Department (Faculty\_ID, Faculty\_Universities\_ID);

GO

CREATE TABLE Groups (

ID INTEGER NOT NULL IDENTITY ,

Specialty\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Specialty\_ID)

REFERENCES Specialty(ID));

GO

CREATE INDEX Groups\_FKIndex1 ON Groups (Specialty\_ID);

GO

CREATE INDEX IFK\_WhatGroup ON Groups (Specialty\_ID);

GO

CREATE TABLE Student (

ID INTEGER NOT NULL IDENTITY ,

Groups\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

Surname VARCHAR(255) ,

Scholarship INTEGER ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Groups\_ID)

REFERENCES Groups(ID));

GO

CREATE INDEX Student\_FKIndex1 ON Student (Groups\_ID);

GO

CREATE INDEX IFK\_WhoIsInGroup ON Student (Groups\_ID);

GO

CREATE TABLE Professor (

ID INTEGER NOT NULL IDENTITY ,

Groups\_ID INTEGER NOT NULL ,

Department\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

Surname VARCHAR(255) ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Department\_ID)

REFERENCES Department(ID),

FOREIGN KEY(Groups\_ID)

REFERENCES Groups(ID));

GO

CREATE INDEX Professor\_FKIndex1 ON Professor (Groups\_ID);

GO

CREATE INDEX Professor\_FKIndex2 ON Professor (Department\_ID);

GO

CREATE INDEX IFK\_WhoIsInCharge ON Professor (Department\_ID);

GO

CREATE INDEX IFK\_WhatProf\_Group ON Professor (Groups\_ID);

GO

CREATE TABLE CourseWork (

ID INTEGER NOT NULL IDENTITY ,

Student\_ID INTEGER NOT NULL ,

Professor\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

Mark INTEGER ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Student\_ID)

REFERENCES Student(ID),

FOREIGN KEY(Professor\_ID)

REFERENCES Professor(ID));

GO

CREATE INDEX CourseWork\_FKIndex1 ON CourseWork (Student\_ID);

GO

CREATE INDEX CourseWork\_FKIndex2 ON CourseWork (Professor\_ID);

GO

CREATE INDEX IFK\_Create ON CourseWork (Student\_ID);

GO

CREATE INDEX IFK\_WhatWork ON CourseWork (Professor\_ID);

GO

CREATE TABLE Discipline (

ID INTEGER NOT NULL IDENTITY ,

CourseWork\_ID INTEGER NOT NULL ,

Student\_ID INTEGER NOT NULL ,

Professor\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

PRIMARY KEY(ID, CourseWork\_ID) ,

FOREIGN KEY(Student\_ID)

REFERENCES Student(ID),

FOREIGN KEY(CourseWork\_ID)

REFERENCES CourseWork(ID),

FOREIGN KEY(Professor\_ID)

REFERENCES Professor(ID));

GO

CREATE INDEX Discipline\_FKIndex1 ON Discipline (Student\_ID);

GO

CREATE INDEX Discipline\_FKIndex2 ON Discipline (CourseWork\_ID);

GO

CREATE INDEX Discipline\_FKIndex3 ON Discipline (Professor\_ID);

GO

CREATE INDEX IFK\_WhatDisc\_Student ON Discipline (Student\_ID);

GO

CREATE INDEX IFK\_WhatDisc\_CW ON Discipline (CourseWork\_ID);

GO

CREATE INDEX IFK\_ProfDisc ON Discipline (Professor\_ID);

GO

1. **Редактирование полученного SQL-кода:**

Если мы попытаемся создать БД на основе неотредактированного скрипта, то при отправке SQL-запроса мы получем множество ошибко при импорте. По этому мы выполняем синтактическое редактирование скрипта и доводим его до состояния принятия средой «phpMyAdmin».

Этапы отладки скрипта:

* Убираем «GO» из скрипта;
* Параметр поля «IDENTITY» заменяем на auto\_increment;
* Для более качественной отладки при создании таблиц указываем «IF NOT EXISTS»;
* Также важно указывать размерность текстовых полей типа «VARCHAR»

Обработанный SQL-скрипт:

CREATE TABLE IF NOT EXISTS Universities (

ID INTEGER NOT NULL auto\_increment ,

Name VARCHAR(255) ,

Rector VARCHAR(255) ,

PRIMARY KEY(ID));

CREATE TABLE IF NOT EXISTS Faculty (

ID INTEGER NOT NULL auto\_increment ,

Universities\_ID INTEGER NOT NULL ,

NameFaculty VARCHAR(255) ,

Dean VARCHAR(255) ,

PRIMARY KEY(ID, Universities\_ID) ,

FOREIGN KEY(Universities\_ID)

REFERENCES Universities(ID));

CREATE INDEX Faculty\_FKIndex1 ON Faculty (Universities\_ID);

CREATE INDEX IFK\_WhatFaculty ON Faculty (Universities\_ID);

CREATE TABLE IF NOT EXISTS Specialty (

ID INTEGER NOT NULL auto\_increment ,

Faculty\_Universities\_ID INTEGER NOT NULL ,

Faculty\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Faculty\_ID, Faculty\_Universities\_ID)

REFERENCES Faculty(ID, Universities\_ID))

CREATE INDEX Specialty\_FKIndex1 ON Specialty (Faculty\_ID, Faculty\_Universities\_ID);

CREATE INDEX IFK\_WhatSpecialty ON Specialty (Faculty\_ID, Faculty\_Universities\_ID);

CREATE TABLE IF NOT EXISTS Department (

ID INTEGER NOT NULL auto\_increment ,

Faculty\_Universities\_ID INTEGER NOT NULL ,

Faculty\_ID INTEGER NOT NULL ,

Name VARCHAR(255) NOT NULL ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Faculty\_ID, Faculty\_Universities\_ID)

REFERENCES Faculty(ID, Universities\_ID));

CREATE INDEX Department\_FKIndex1 ON Department (Faculty\_ID, Faculty\_Universities\_ID);

CREATE INDEX IFK\_WhatDepartment ON Department (Faculty\_ID, Faculty\_Universities\_ID);

CREATE TABLE IF NOT EXISTS Groups (

ID INTEGER NOT NULL auto\_increment ,

Specialty\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Specialty\_ID)

REFERENCES Specialty(ID));

CREATE INDEX Groups\_FKIndex1 ON Groups (Specialty\_ID);

CREATE INDEX IFK\_WhatGroup ON Groups (Specialty\_ID);

CREATE TABLE IF NOT EXISTS Student (

ID INTEGER NOT NULL auto\_increment ,

Groups\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

Surname VARCHAR(255) ,

Scholarship INTEGER ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Groups\_ID)

REFERENCES Groups(ID));

CREATE INDEX Student\_FKIndex1 ON Student (Groups\_ID);

CREATE INDEX IFK\_WhoIsInGroup ON Student (Groups\_ID);

CREATE TABLE IF NOT EXISTS Professor (

ID INTEGER NOT NULL auto\_increment ,

Groups\_ID INTEGER NOT NULL ,

Department\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

Surname VARCHAR(255) ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Department\_ID)

REFERENCES Department(ID),

FOREIGN KEY(Groups\_ID)

REFERENCES Groups(ID));

CREATE INDEX Professor\_FKIndex1 ON Professor (Groups\_ID);

CREATE INDEX Professor\_FKIndex2 ON Professor (Department\_ID);

CREATE INDEX IFK\_WhoIsInCharge ON Professor (Department\_ID);

CREATE INDEX IFK\_WhatProf\_Group ON Professor (Groups\_ID);

CREATE TABLE IF NOT EXISTS CourseWork (

ID INTEGER NOT NULL auto\_increment ,

Student\_ID INTEGER NOT NULL ,

Professor\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

Mark INTEGER ,

PRIMARY KEY(ID) ,

FOREIGN KEY(Student\_ID)

REFERENCES Student(ID),

FOREIGN KEY(Professor\_ID)

REFERENCES Professor(ID));

CREATE INDEX CourseWork\_FKIndex1 ON CourseWork (Student\_ID);

CREATE INDEX CourseWork\_FKIndex2 ON CourseWork (Professor\_ID);

CREATE INDEX IFK\_Create ON CourseWork (Student\_ID);

CREATE INDEX IFK\_WhatWork ON CourseWork (Professor\_ID);

CREATE TABLE IF NOT EXISTS Discipline (

ID INTEGER NOT NULL auto\_increment ,

CourseWork\_ID INTEGER NOT NULL ,

Student\_ID INTEGER NOT NULL ,

Professor\_ID INTEGER NOT NULL ,

Name VARCHAR(255) ,

PRIMARY KEY(ID, CourseWork\_ID) ,

FOREIGN KEY(Student\_ID)

REFERENCES Student(ID),

FOREIGN KEY(CourseWork\_ID)

REFERENCES CourseWork(ID),

FOREIGN KEY(Professor\_ID)

REFERENCES Professor(ID));

CREATE INDEX Discipline\_FKIndex1 ON Discipline (Student\_ID);

CREATE INDEX Discipline\_FKIndex2 ON Discipline (CourseWork\_ID);

CREATE INDEX Discipline\_FKIndex3 ON Discipline (Professor\_ID);

CREATE INDEX IFK\_WhatDisc\_Student ON Discipline (Student\_ID);

CREATE INDEX IFK\_WhatDisc\_CW ON Discipline (CourseWork\_ID);

CREATE INDEX IFK\_ProfDisc ON Discipline (Professor\_ID);

1. **Экспорт в БД «phpMyAdmin»:**

Установив локальный сервер phpMyAdmin мы идет в раздел создания новой БД, см. рис. 4.

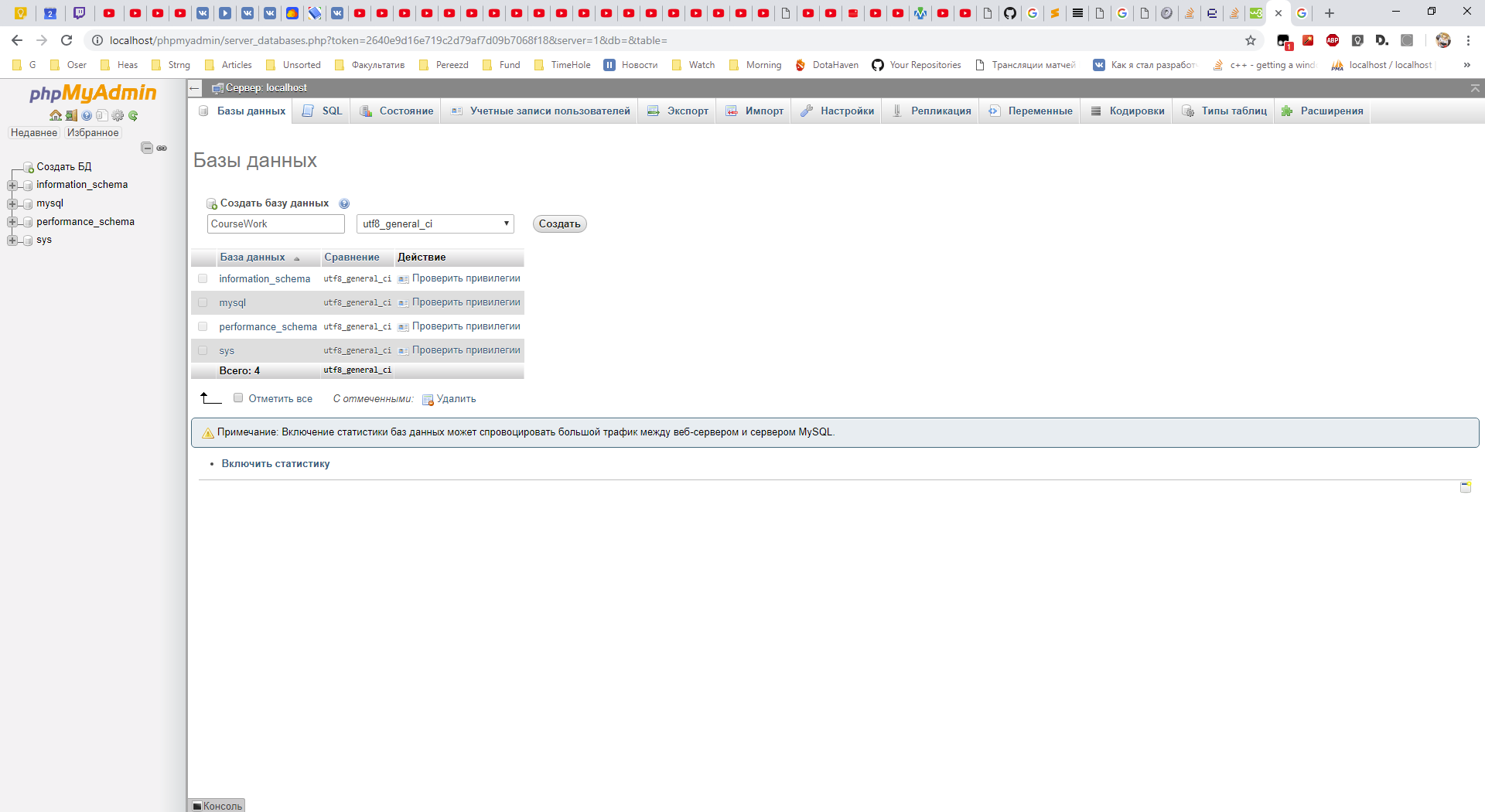


Рисунок 4 – Создание БД в phpMyAdmin

После создания БД, идем в раздел SQL, в котором с помощью скрипта будем наполнять нашу пока что пустую БД, см. рис. 5.

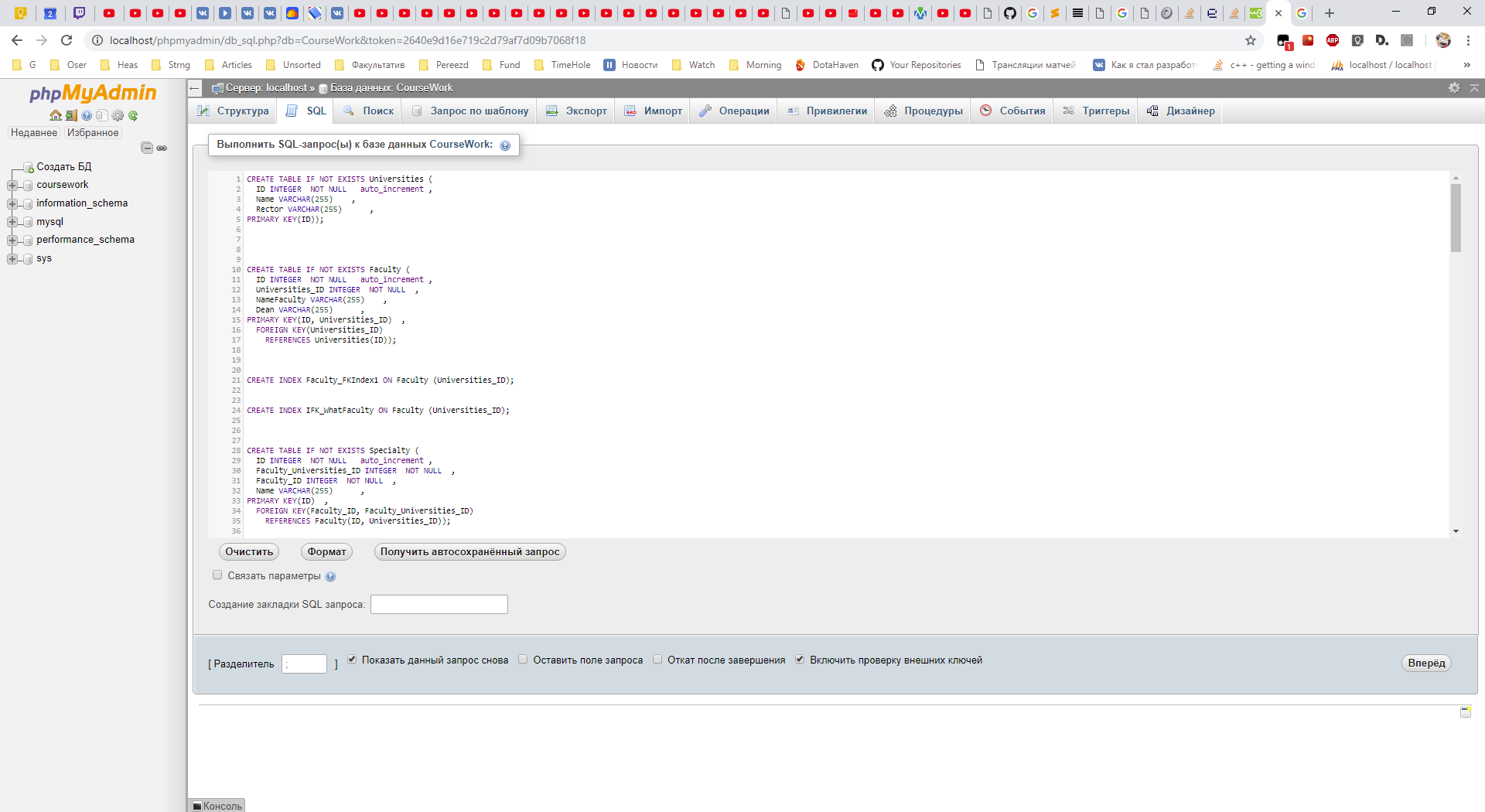


Рисунок 5 – Запуск SQL-скрипта

При успешном запросе к серверу мы получаем сообщения о успошном выполнения скрипта, см. рис. 6.

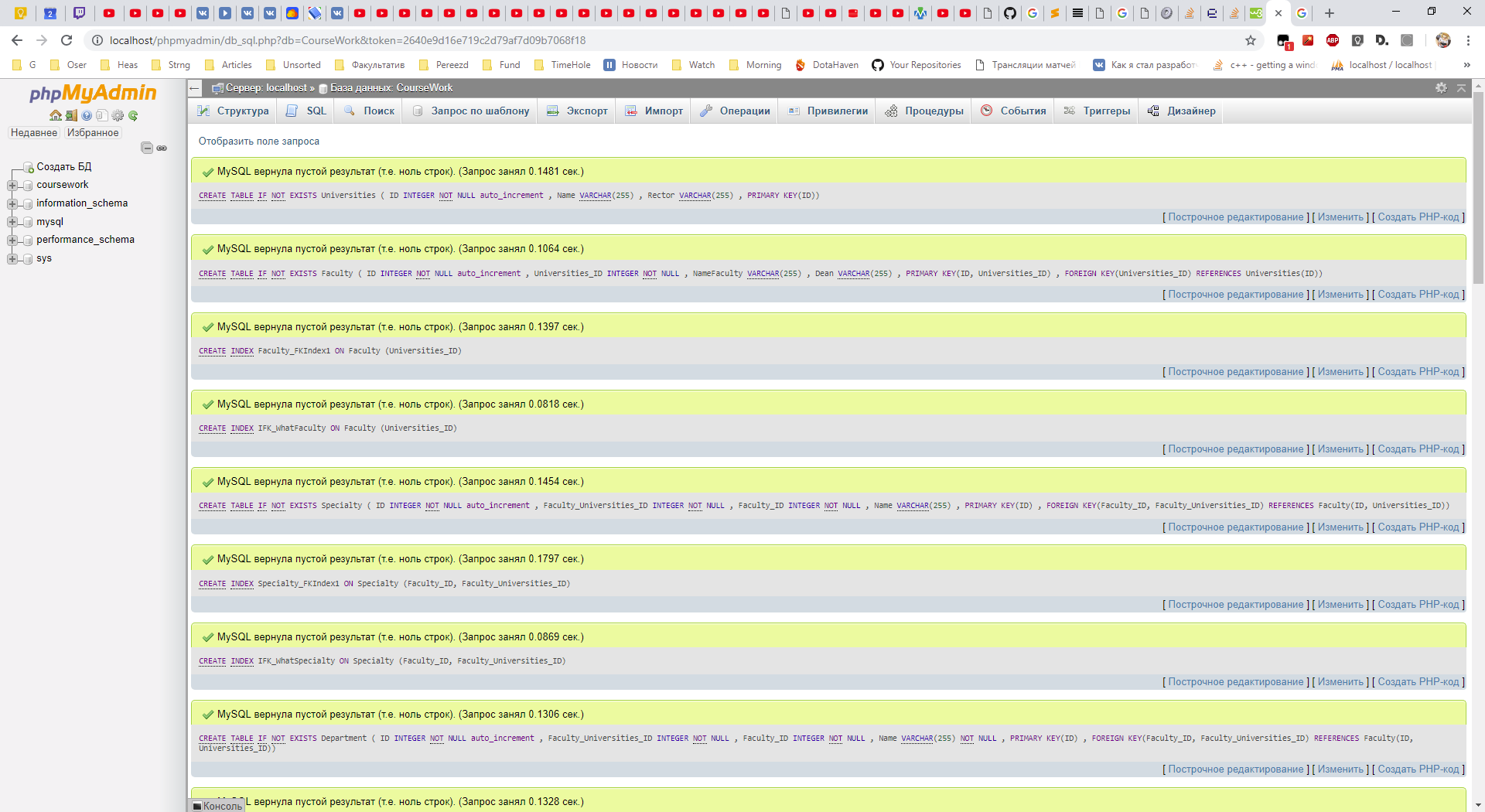


Рисунок 6 – Сообщения о успешном завершении SQL-запроса

Во складке «Дизайн» мы можем смоделировать инфологическую модель БД, см. рис 7.

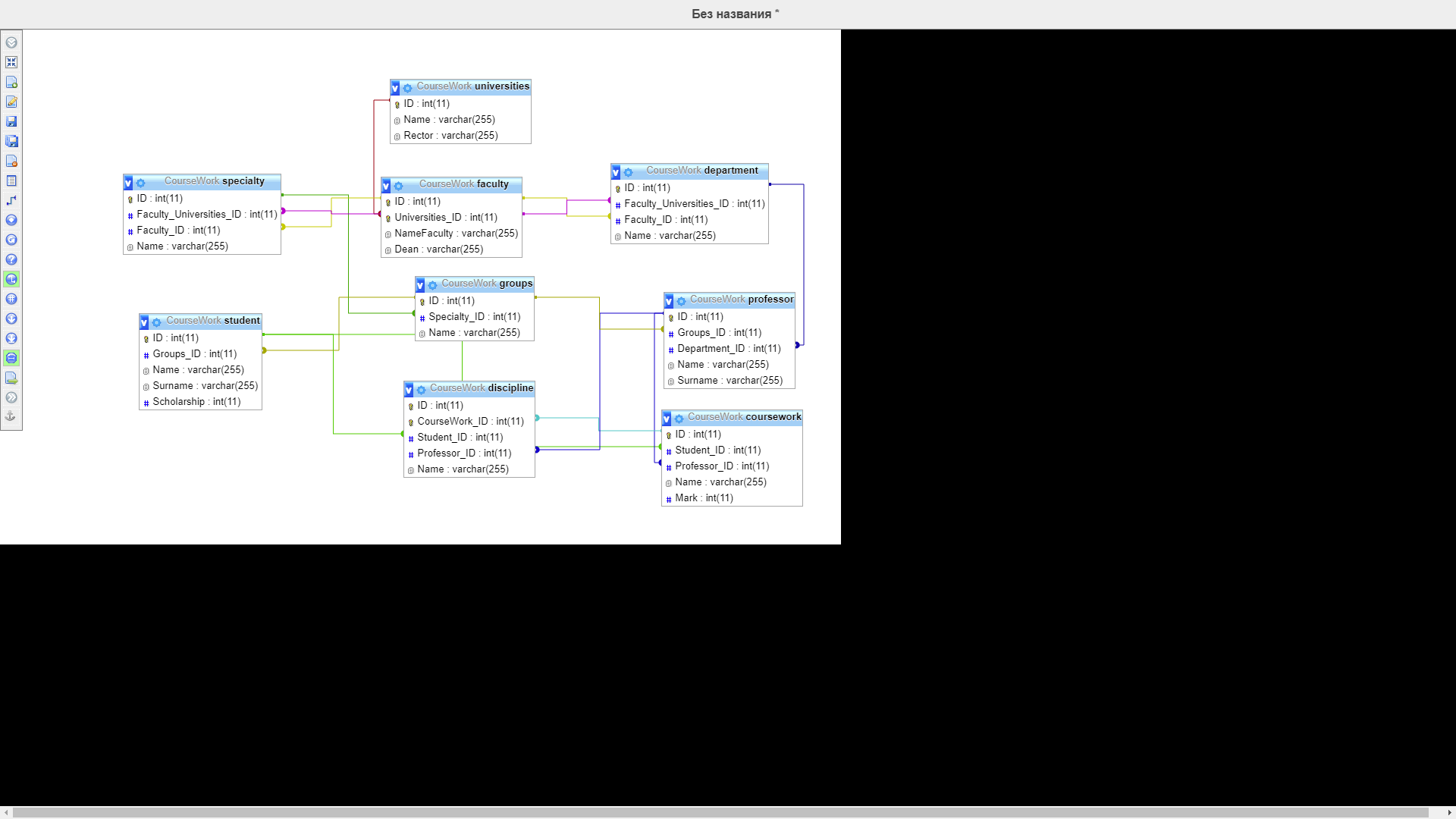


Рисунок 7 – Графическое представление БД

**5) Проверка на целостность экспортированной БД:**

Сформированная в среде phpMyAdmin БД полностью соответствует той, что и исходная из среды «DBDesigner», которая была описана в п. 1.

**Вывод:** в этой работе мы получили навыки экспортирования и импортирования SQL-скрипта из разных сред разработки БД.