Министерство науки и образования Российской Федерации

Пензенский государственный университет

Кафедра «Системы автоматизированного проектирования»

**Отчет**

По лабораторной работе №2

По курсу «Информационное обеспечение САПР»

На тему «РАЗРАБОТКА ЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ И ГЕНЕРАЦИЯ SQL -

СКРИПТА ДЛЯ СОЗДАНИЯ БАЗЫ ДАННЫХ»

Выполнил студент гр.20ВВ2.1

Горбунов Н.А.

Проверили:

Глотова Т.В.  
Евсеева Ю.И.

Пенза, 2022

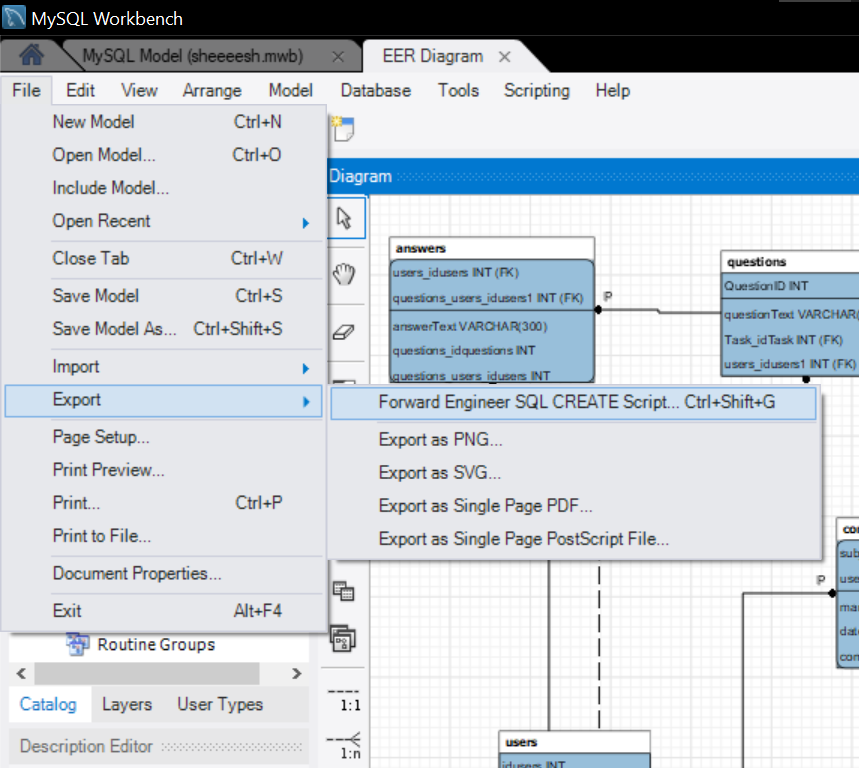
**Цель работы:** изучение возможностей языка SQL для создания баз

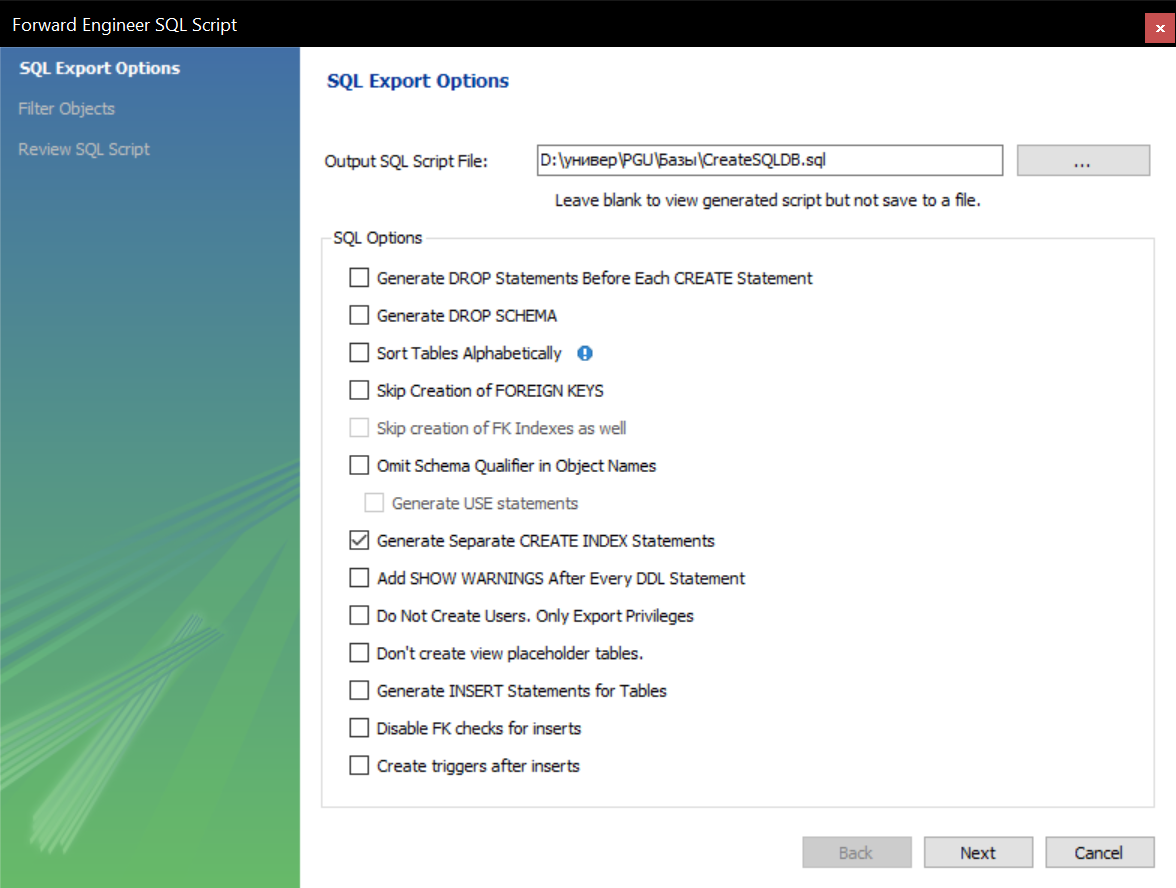
Данных.

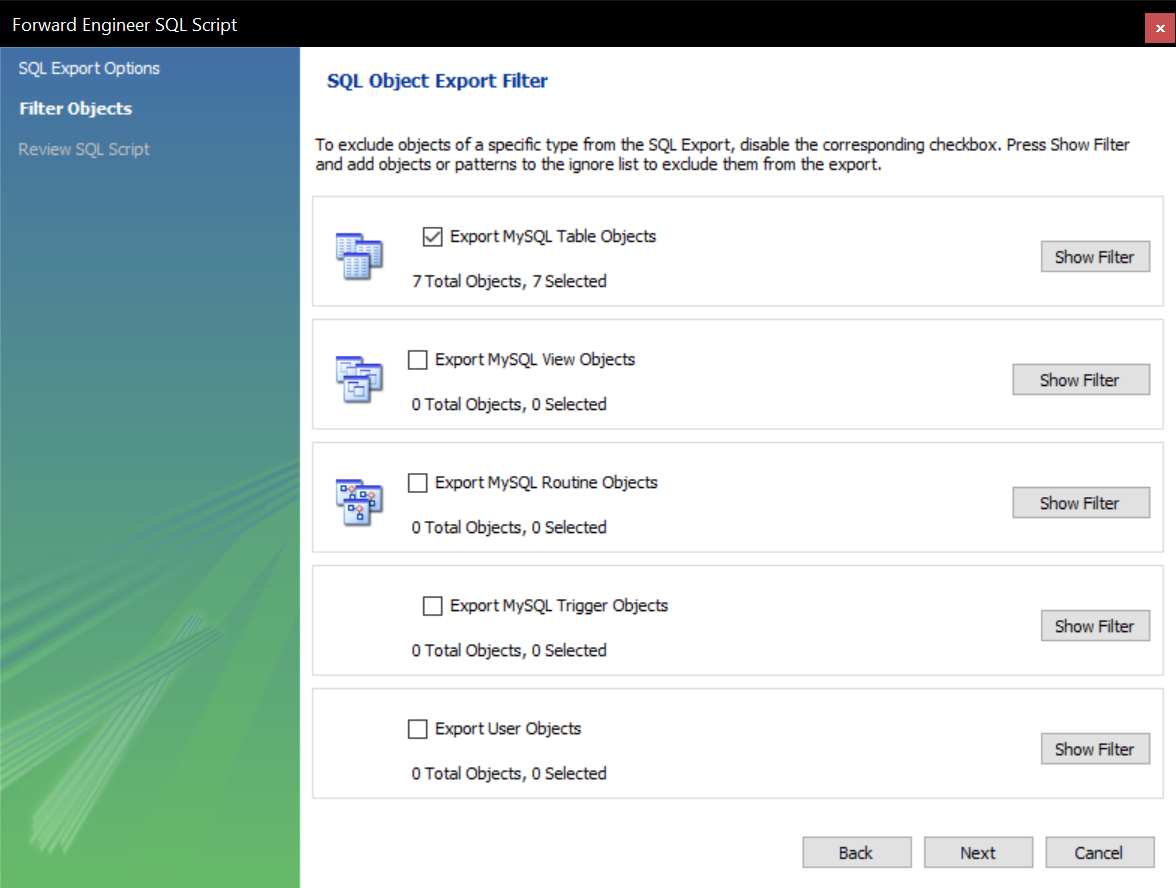
**Описание заданной предметной области (назначение системы):**

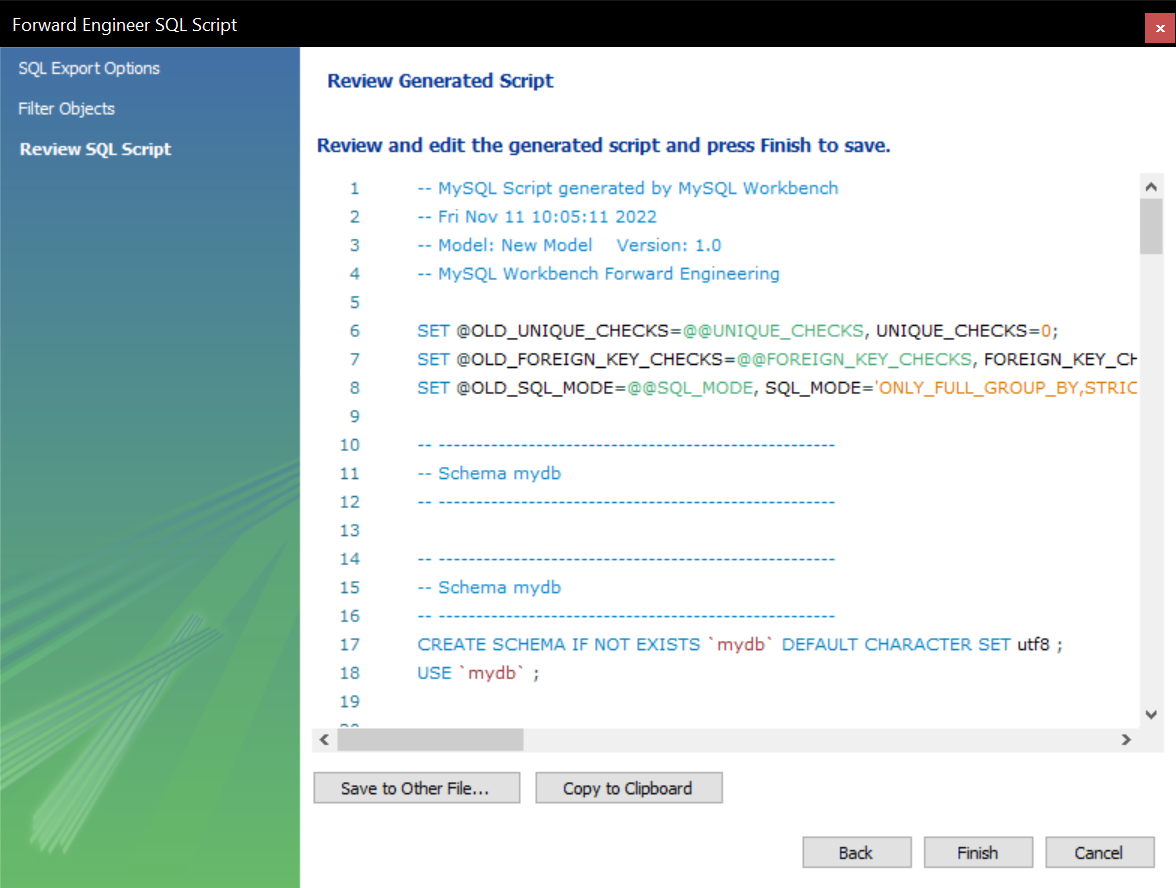
Проектирование базы данных задач студента и разработки приложения их учета.

1. Экспортируем БД, как новый скрипт создания базы данных.









**Листинг:**

-- MySQL Script generated by MySQL Workbench

-- Fri Nov 11 10:05:11 2022

-- 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`.`users`

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

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

`idusers` INT NULL AUTO\_INCREMENT,

`username` VARCHAR(45) NULL,

`password` VARCHAR(255) NULL,

`raiting` INT NULL,

PRIMARY KEY (`idusers`))

ENGINE = InnoDB;

CREATE UNIQUE INDEX `idusers\_UNIQUE` ON `mydb`.`users` (`idusers` ASC) VISIBLE;

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

-- Table `mydb`.`Subject`

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

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

`idSubject` INT NOT NULL AUTO\_INCREMENT,

`ThemeName` VARCHAR(255) NULL,

PRIMARY KEY (`idSubject`))

ENGINE = InnoDB;

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

-- Table `mydb`.`Task`

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

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

`idTask` INT NULL AUTO\_INCREMENT,

`TaskName` VARCHAR(255) NULL,

`Charactiristic` JSON NULL,

`Subject\_idSubject` INT NOT NULL,

PRIMARY KEY (`idTask`),

CONSTRAINT `fk\_subject\_Subject1`

FOREIGN KEY (`Subject\_idSubject`)

REFERENCES `mydb`.`Subject` (`idSubject`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE INDEX `fk\_subject\_Subject1\_idx` ON `mydb`.`Task` (`Subject\_idSubject` ASC) VISIBLE;

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

-- Table `mydb`.`files`

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

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

`file` VARCHAR(255) NULL,

`date` DATETIME NULL,

`comment` VARCHAR(300) NULL,

`filescol` VARCHAR(45) NULL,

`users\_idusers` INT NOT NULL,

`Task\_idTask` INT NOT NULL,

`id\_file` INT NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`id\_file`),

CONSTRAINT `fk\_files\_users1`

FOREIGN KEY (`users\_idusers`)

REFERENCES `mydb`.`users` (`idusers`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_files\_Task1`

FOREIGN KEY (`Task\_idTask`)

REFERENCES `mydb`.`Task` (`idTask`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE INDEX `fk\_files\_users1\_idx` ON `mydb`.`files` (`users\_idusers` ASC) VISIBLE;

CREATE INDEX `fk\_files\_Task1\_idx` ON `mydb`.`files` (`Task\_idTask` ASC) VISIBLE;

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

-- Table `mydb`.`questions`

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

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

`questionText` VARCHAR(255) NULL,

`QuestionID` INT NOT NULL AUTO\_INCREMENT,

`Task\_idTask` INT NOT NULL,

`users\_idusers1` INT NOT NULL,

PRIMARY KEY (`QuestionID`),

CONSTRAINT `fk\_questions\_Task1`

FOREIGN KEY (`Task\_idTask`)

REFERENCES `mydb`.`Task` (`idTask`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_questions\_users1`

FOREIGN KEY (`users\_idusers1`)

REFERENCES `mydb`.`users` (`idusers`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE INDEX `fk\_questions\_Task1\_idx` ON `mydb`.`questions` (`Task\_idTask` ASC) VISIBLE;

CREATE INDEX `fk\_questions\_users1\_idx` ON `mydb`.`questions` (`users\_idusers1` ASC) VISIBLE;

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

-- Table `mydb`.`answers`

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

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

`answerText` VARCHAR(300) NULL,

`questions\_idquestions` INT NOT NULL,

`questions\_users\_idusers` INT NOT NULL,

`users\_idusers` INT NOT NULL,

`questions\_users\_idusers1` INT NOT NULL,

PRIMARY KEY (`users\_idusers`, `questions\_users\_idusers1`),

CONSTRAINT `fk\_answers\_users1`

FOREIGN KEY (`users\_idusers`)

REFERENCES `mydb`.`users` (`idusers`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_answers\_questions1`

FOREIGN KEY (`questions\_users\_idusers1`)

REFERENCES `mydb`.`questions` (`QuestionID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE INDEX `fk\_answers\_users1\_idx` ON `mydb`.`answers` (`users\_idusers` ASC) VISIBLE;

CREATE INDEX `fk\_answers\_questions1\_idx` ON `mydb`.`answers` (`questions\_users\_idusers1` ASC) VISIBLE;

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

-- Table `mydb`.`completeData`

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

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

`subject\_idsubject` INT NOT NULL,

`users\_idusers` INT NOT NULL,

`mark` INT NULL,

`date` DATETIME NULL,

`completeData` DATETIME NULL,

PRIMARY KEY (`subject\_idsubject`, `users\_idusers`),

CONSTRAINT `fk\_completeData\_subject1`

FOREIGN KEY (`subject\_idsubject`)

REFERENCES `mydb`.`Task` (`idTask`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_completeData\_users1`

FOREIGN KEY (`users\_idusers`)

REFERENCES `mydb`.`users` (`idusers`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

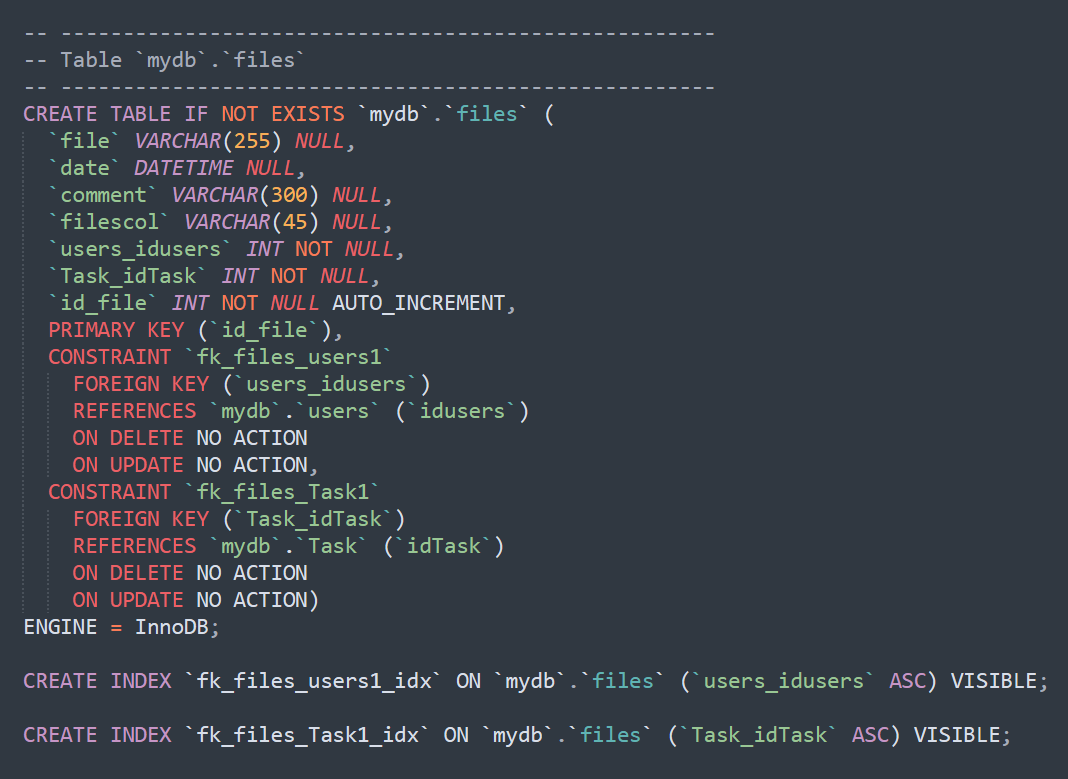
CREATE INDEX `fk\_completeData\_users1\_idx` ON `mydb`.`completeData` (`users\_idusers` ASC) VISIBLE;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

Проведем анализ скрипта на примере таблицы: files



**CREATE TABLE** - устанавливает новую таблицу, ее столбцы и ограничения целостности в существующей базе данных.

**IF NOT EXIST** - не создаст таблицу, если таблица с таким же именем уже существует в базе данных.

“**`file` VARCHAR(255) NULL,**” – “file” – Название атрибута

**VARCHAR(255) –** Тип значения атрибута “file”

**NULL –** Означает, что значения может быть null

**PRIMARY KAY –** указывает, какое поле таблицы является первичным ключом.

**CREATE** **INDEX –** создаёт индекс. В данном случае индекс создается из-за того что у таблицы есть внешний ключ idTask

**CONSTRAINT –** Задаёт ограничение (поддержку целостности), так как таблица имеет внешний ключ и в ней могут быть только данные для которых определен этот ключ. В данном случае для ограничения используется внешний ключ idUser.

**REFERENCES –** Указывает имя таблицы с которой связана текущая таблица и её поле, использующееся, как внешний ключ.

Таким образом база данных InnoDB поддерживает целостность данных, далее написаны предложения **ON DELETE NO ACTION, ON UPDATE NO ACTION.** Это указывает, что при изменении и удалении кортежа со связанными таблицами не будет производится никаких действий. Для данной БД это не является проблемой

**Вывод:** научился использовать MYSSQLWORKBENCH для генерации скриптов создания БД. Проанализировал синтаксис сгенерированного скрипта создания таблицы «CREATE TABLE» и синтаксис формирования связей и поддержки ссылочной ценности.