

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»

**ЭЛЕКТРОННЫЙ ТРЕНАЖЕР**  
**«ДЕКОМПОЗИЦИЯ ПЛОСКОЙ ТАБЛИЦЫ»**

Выпускная квалификационная работа  
по направлению подготовки 44.03.04 Профессиональное обучение  
(по отраслям)  
профилю подготовки «Информатика и вычислительная техника»  
специализации «Компьютерные технологии»

Идентификационный номер ВКР: 021

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический университет»  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ  
Заведующий кафедрой ИС  
\_\_\_\_\_ И. А. Сулова  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**ЭЛЕКТРОННЫЙ ТРЕНАЖЕР**  
**«ДЕКОМПОЗИЦИЯ ПЛОСКОЙ ТАБЛИЦЫ»**

Исполнитель:

обучающийся группы КТ-403

О. А. Глушков

Руководитель:

ст. преподаватель каф. ИС

С. Ю. Ярина

Нормоконтролер:

ст. преподаватель каф. ИС

Н. В. Хохлова

Екатеринбург 2019

## АННОТАЦИЯ

Выпускная квалификационная работа состоит из электронного тренажера «Декомпозиция плоской таблицы» и пояснительной записки на 54 страницах, содержащей 23 рисунка, 11 таблиц, 30 источников литературы, а также 1 приложение на 2 страницах.

Ключевые слова: ДЕКОМПОЗИЦИЯ, БАЗА ДАННЫХ, ЭЛЕКТРОННЫЙ ТРЕНАЖЕР, ПРОЕКТИРОВАНИЕ, РЕЛЯЦИОННАЯ БАЗА ДАННЫХ, ПЛОСКАЯ ТАБЛИЦА.

**Глушов О. А.**, Электронный тренажер «Декомпозиция плоской таблицы»: выпускная квалификационная работа / О. А. Глушков; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2019. — 54 с.

В работе рассмотрены вопросы разработки электронного тренажера по теме «Декомпозиция плоской таблицы» для обучающихся по дисциплине «Базы данных».

Целью работы является разработка электронного тренажера «Декомпозиция плоской таблицы» предназначенный для студентов третьего курса направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля «Информационные технологии». Для достижения цели были рассмотрены основные техно-технологические требования, предъявляемые при разработке электронных средств обучения, проанализирована учебно-методическая документация дисциплины «Базы данных», литературные и интернет-источники по теме «Декомпозиция плоской таблицы», выбраны средства реализации электронного тренажера. Разработан электронный тренажер «Декомпозиция плоской таблицы», включающий в себя три режима работы: свободный, обучающий и контроля знаний, справочную документацию по работе с тренажером и теоретические материалы, а также 30 заданий разной уровни сложности.

# СОДЕРЖАНИЕ

Введение.....	5
1 Декомпозиция, как метод проектирования реляционных баз данных. Нормальные формы .....	7
1.1 Основные понятия и определения.....	7
1.2 Нормальные формы .....	9
1.2.1 Первая нормальная форма.....	9
1.2.2 Вторая нормальная форма.....	10
1.2.3 Третья нормальная форма .....	11
1.3 Анализ литературы и интернет-источников по теме «Декомпозиция плоской таблицы» .....	13
1.3.1 Анализ литературы .....	13
1.3.2 Анализ интернет-источников .....	14
1.3.3 Анализ рабочей программы .....	16
2 Описание электронного тренажера «Декомпозиция плоской таблицы» .....	22
2.1 Педагогический адрес.....	22
2.2 Общие требования по созданию электронных средств обучения .....	22
2.3 Требования к тренажеру «Декомпозиция плоской таблицы» .....	26
2.4 Описание средств разработки.....	28
2.5 Структура электронного тренажера.....	30
2.6 Режимы работы тренажера.....	34
2.6.1 Свободный режим.....	34
2.6.2 Режим обучения .....	35
2.6.3 Режим контроля знаний.....	36
2.6.4 Режим редактирования .....	37
2.7 Сопроводительная документация.....	40
2.8 Описание работы с тренажером .....	42
2.9 Описание основных алгоритмов программы .....	44

Заключение .....	49
Список использованных источников .....	50
Приложение .....	53

## **ВВЕДЕНИЕ**

В настоящее время технология баз данных (БД) занимает центральное место в любом информационном процессе. Эффективность разрабатываемых информационных систем во многом зависит от качества спроектированной базы данных. Поэтому потребность в квалифицированных специалистах, имеющих достаточно знаний по проектированию реляционных баз данных, является важной и актуальной задачей. В свою очередь процесс проектирования является достаточно сложной задачей и дается далеко не всем обучающимся с первого раза. Это связано со сложностью в определении состава таблиц и связей между ними в следствии малого количества выделяемых учебных часов на изучение данного раздела.

При изучении раздела «Основы проектирования реляционных баз данных» остро стоит необходимость в использовании информационных технологий для значительного увеличения эффективности процесса обучения и сокращения времени на изучения нового материала, что в свою очередь позволит обучающимся работать в наилучшем темпе для оптимального усвоения материала, то есть позволяет осуществлять личностно-ориентированный подход в обучении, а также позволит отработать навыки проектирования структур баз данных.

Электронный тренажер является примером информационных технологий в образовании, использование которого дает возможность отработать навыки решения типовых задач, позволяющих наглядно связать теоретические знания с конкретными проблемами, на решение которых они могут быть направлены.

Для эффективного изучения обучающимися основ проектирования структур реляционных баз данных необходимо наличие педагогического программного средства, объединяющего в себе простоту использования гра-

фического интерфейса и возможности самостоятельного составления SQL-запросов.

Кроме того, автоматизация контроля выполнения заданий по теме «Декомпозиция плоской таблицы» в этом случае является уместной и оправданной, так как позволит провести объективную и эффективную проверку знаний, позволит студентам отследить свои ошибки непосредственно после выполнения задания, что может служить дополнительным стимулом в процессе обучения.

**Объект выпускной квалификационной работы** — процесс обучения студентов направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля «Информационные технологии» дисциплине «Базы данных».

**Предмет выпускной квалификационной работы** — учебные материалы по теме «Декомпозиция плоской таблицы» дисциплины «Базы данных».

**Цель выпускной квалификационной работы** — разработать электронный тренажер «Декомпозиция плоской таблицы».

Для достижения поставленной цели необходимо решить следующие задачи:

1. Проанализировать литературу и интернет-источники по теме «Декомпозиция плоской таблицы» с целью выявления основных методов проектирования реляционной базы данных, являющихся основными и критическими для обучения данному разделу.

2. Проанализировать литературу и интернет-источники с целью выделения требований, предъявляемых к программам-тренажерам и электронным учебным пособиям на современном этапе развития образования.

3. Спроектировать структуру и реализовать функционал тренажера «Декомпозиция плоской таблицы» в выбранной среде программирования.

4. Разработать набор практических заданий с целью проверки функционала разработанного электронного тренажера и создания базы для дальнейшего развития разработанного продукта.

# 1 ДЕКОМПОЗИЦИЯ, КАК МЕТОД ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ. НОРМАЛЬНЫЕ ФОРМЫ

## 1.1 Основные понятия и определения

Реляционная база данных — это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа [5].

Благодаря простоте и гибкости структуры реляционной модели данных, удобству реализации с помощью компьютерных технологий, реляционные базы данных в настоящее время применяются в подавляющем большинстве информационных систем.

Основной информационной единицей в реляционной базе данных является таблица, в которой хранятся сведения об однотипных объектах информационной системы.

Строки таблицы называются записями, а столбцы — полями. Каждая запись хранит информацию о некотором объекте, а в полях размещаются, описывающие объект, свойства. Пример реляционной базы данных представлен на рисунке 1.

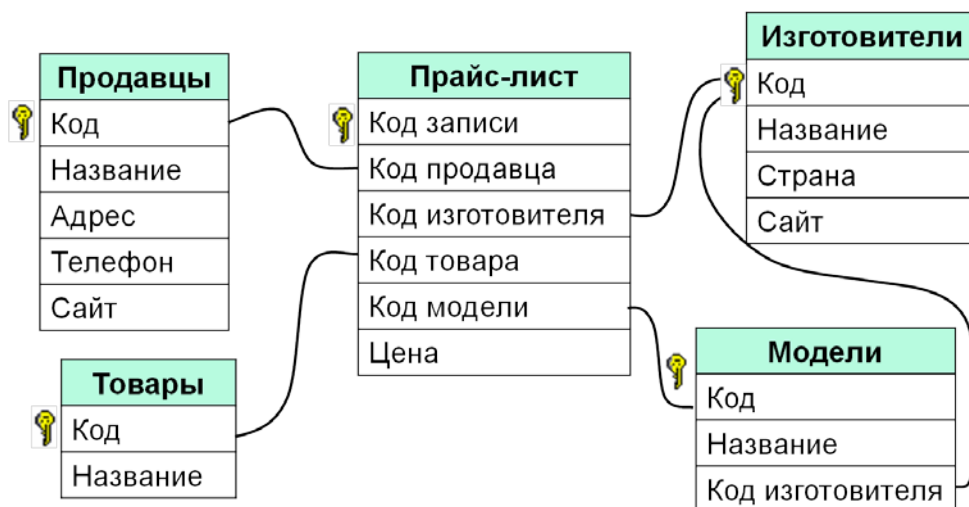


Рисунок 1 — Реляционная база данных



Реляционная таблица обладает следующими свойствами:

- любая таблица имеет уникальное имя и состоит из однотипных строк;
- записи имеют фиксированное число полей и значений (не допускается использование повторяющихся групп);
- в таблице не допускается хранение одинаковых записей;
- полям присваиваются уникальные имена и в каждом поле хранятся элементы имеющие один и тот же тип данных;
- при выполнении операций с таблицей записи и поля можно обрабатывать в произвольном порядке [2].

Одним из методов проектирования реляционных баз данных является метод декомпозиции.

Декомпозиция — исходное множество отношений, входящих в схему базы данных заменяется другим множеством отношений, являющихся проекциями исходных отношений! При этом число отношений возрастает.

Классическое проектирование БД связано с теорией нормализации, которая основана на анализе функциональных зависимостей между атрибутами отношений [1].

Метод декомпозиции представляет собой последовательный процесс нормализации схем отношений: каждая новая итерация соответствует нормальной форме более высокого порядка и обладает лучшими свойствами по сравнению с предыдущей формой, таким образом, изначально предполагается существование универсального отношения, содержащего все атрибуты БД, на основе которых осуществляется или делается попытка осуществить декомпозицию универсальных отношений, т.е. переход к отношениям меньшей размерности с помощью операций естественного соединения.

Нормализация — разбиение исходного отношения на два или несколько, обладающих лучшими свойствами при включении, изменении и удалении данных.

Целью нормализации является получение БД, где каждое значение данных появляется только один раз, т.е. исключена избыточность, причем к этому следует стремиться не столько ради экономии памяти, сколько для исключения возможности противоречивости хранимых данных.

Нормализация выражается в приведении отношений в соответствии с некоторыми требованиями, называемыми нормальными формами. На сегодняшний день известны шесть нормальных форм:

- 1 Первая нормальная форма (1 НФ).
- 2 Вторая нормальная форма (2 НФ).
- 3 Третья нормальная форма (3 НФ).
- 4 Нормальная форма Бойса-Кодда (НФБК).
- 5 Четвертая нормальная форма (4 НФ).
- 6 Пятая нормальная форма (5 НФ).

Основные свойства нормальных форм:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

Ниже рассмотрим примеры нормализации и первые три нормальные формы, являющиеся основными при создании реляционных баз данных.

## **1.2 Нормальные формы**

### **1.2.1 Первая нормальная форма**

Отношение называется нормализованным или приведенным к первой нормальной форме (1 НФ), если все его атрибуты являются простыми, то есть не могут быть далее разделены [12].

Пример нарушения требования к первой нормальной форме представлена в таблице 1.

Таблица 1 — Пример нарушения первой нормальной формы

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Нарушение первой нормальной формы происходит в моделях BMW, т.к. в одной ячейке содержится список из трех элементов: M5, X5M, M1, т.е. он не является атомарным. Ниже в таблице 2 представлен преобразованный вид таблицы 1 к правильному представлению первой нормальной формы:

Таблица 2 — Преобразование отношения

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

Путем дублирования данных фирмы BMW было достигнуто отношение первой нормальной формы, в каждой ячейки содержится только одно логическое значение, т.е. соблюдается правило атомарности. Избавиться от избыточности в базе данных поможет вторая нормальная форма.

### 1.2.2 Вторая нормальная форма

Отношение находится во второй нормальной форме (2 НФ), если оно находится в первой нормальной форме и каждый не ключевой атрибут неприводимо зависит от первичного ключа [12].

Вторая нормальная форма требует, чтобы у каждой сущности обязательно был ключевые атрибут, а также отсутствовала избыточность вывода данных в справочники.

Пример, представленный в таблице 3, находится в первой нормальной форме, но не во второй. Скидка зависит от фирмы, но не зависит от модели и цены, то есть зависимость от первичного ключа неполная.

Таблица 3 — Пример нарушения второй нормальной формы

<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%

Исправляется это путем декомпозиции на два отношения (таблица 4, таблица 5), в которых не ключевые атрибуты зависят от первичного ключа.

Таблица 4 — Пример правильного представления второй нормальной формы

<u>Модель</u>	<u>Фирма</u>	Цена
M5	BMW	5500000
X5M	BMW	6000000
M1	BMW	2500000
GT-R	Nissan	5000000

Таблица 5 — Справочник к таблице 4

<u>Фирма</u>	Скидка
BMW	5%
Nissan	10%

Выполнив такое преобразование и избавившись от некоторой избыточности данных, получается отношение, находящееся во второй нормальной форме.

### 1.2.3 Третья нормальная форма

Отношение находится в третьей нормальной форме (3 НФ), когда находится во второй нормальной форме, и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа [12].

Третья нормальная форма избавляет нас от транзитивных зависимостей: любой столбец таблицы должен зависеть только от ключевого столбца.

Пример, представленный в таблице 6, находится во второй нормальной форме, но не в третьей. В отношении атрибут «Модель» является первичным ключом. Личных телефонов у автомобилей нет, и телефон зависит исключительно от магазина.

Таблица 6 — Пример нарушения третьей нормальной формы

<u>Модель</u>	Магазин	Телефон
BMW	Автобан-запад	87-33-98
Audi	Автобан-запад	87-33-98
Nissan	Автовек	94-54-12

Таким образом, в отношении существуют следующие функциональные зависимости: Модель → Магазин, Магазин → Телефон, Модель → Телефон.

Зависимость Модель → Телефон является транзитивной, следовательно, отношение не находится в третьей нормальной форме. Полученные отношения представлены в таблице 7 и таблице 8.

Таблица 7 — Отношение «Магазин → Телефон»

Магазин	Телефон
Автобан-запад	87-33-98
Автовек	94-54-12

Таблица 8 — Отношение «Модель → Магазин»

<u>Модель</u>	Магазин
BMW	Автобан-запад
Audi	Автобан-запад
Nissan	Автовек

Выполнив декомпозицию, получаем два отношения, находящиеся в третьей нормальной форме.

Моделирование структуры БД при помощи алгоритма нормализации, имеет серьёзно недостатки:

1. Первоначальное размещение всех атрибутов в одном отношении является очень неестественной операцией. Интуитивно разработчик сразу проектирует несколько отношений в соответствии с обнаруженными сущностями. Даже если создать одно или несколько отношений, включив в них все предполагаемые атрибуты, то совершенно неясен смысл полученного отношения.

2. Невозможно сразу определить полный список атрибутов. Пользователи имеют привычку называть разными именами одни и те же вещи, или наоборот, называть одними именами разными вещи.

3. Для проведения процедуры нормализации необходимо выделить зависимости атрибутов, что тоже очень нелегко, так как необходимо явно выписать все зависимости, даже те, которые являются очевидным [3].

### **1.3 Анализ литературы и интернет-источников по теме «Декомпозиция плоской таблицы»**

#### **1.3.1 Анализ литературы**

Для разработки электронного тренажера необходимо проанализировать литературу по теме «Декомпозиция плоской таблицы», что позволит понять теоретическую сторону работы и систематизировать материал.

В учебнике Илющечкина В. М. «Основы использования и проектирования баз данных» [7] содержатся теоретические и практические сведения о современных системах управления базами данных (СУБД), об использовании и проектировании баз данных. Рассматриваются языковые и программные средства СУБД и систем автоматизации проектирования баз данных. Кроме известного структурированного языка запросов SQL, в данном учебнике также уделяется внимание языкам определения данных (DDL), манипулирования данными (DML), а также языку запросов по образцу QBE. Приведены примеры создания инфологических и даталогических моделей, позволяющие студентам научиться проектировать базы данных. Описаны принципы нормализации и алгоритмы проектирования реляционных баз данных на основе этих принципов. Отдельная глава учебника посвящена реляционной модели данных.

В учебнике Избачкова Ю. С., Петровой В. Н., Васильева А. А., Телиной И. С. «Информационные системы» [6] основное внимание уделяется вопросам разработки клиентской части информационных систем с использованием приложений Delphi. Несмотря на это, книга содержит достаточно большое количество практического материала, посвященного проектирова-

нию баз данных. В учебном пособии подробно излагаются теоретические сведения о реляционной модели данных.

В учебнике Кузнецова С. Д. «Базы данных» [9] обсуждаются потребности разработчиков информационных систем в технологии баз данных, рассматриваются основные функции и типовая архитектура СУБД, а также приводится краткая характеристика нескольких популярных моделей данных. Подробно описывается реляционная модель данных, проектирования реляционных баз данных с использованием принципов нормализации и на основе семантических диаграммных моделей данных. В учебнике представлены также основные методы и алгоритмы, используемые в SQL-ориентированных СУБД. Выделены наиболее важные черты языка SQL как отдельной модели данных.

В учебнике Нагела К. и Ивьена Б. «С# 4.0 и платформа.NET 4 для профессионалов» [11] содержится большое количество примеров и рекомендаций для написания высококачественных программ. Легкий и доступный стиль изложения поможет новичкам как можно быстрее приступить к созданию собственных программ. Данное пособие отличается большой охватом тем: от общего анализа архитектуры.NET до руководства по использованию технологии LINQ. Дополнительно, книга рассказывает о самом языке С# и о способах его применения в различных областях.

### **1.3.2 Анализ интернет-источников**

Несмотря на то, что теоретическая информация по теме «Декомпозиция плоской таблицы» подобно основам проектирования реляционных баз данных обновляется довольно редко, интернет-источники предоставляют обширное количество наглядных примеров операций в открытом доступе, а также научно-популярные статьи на данную тему. С другой стороны, для разработки электронного тренажера необходимо знание средств разработки и языка программирования (в данном случае, С#). Для данной цели интернет-

источники подходят намного лучше литературных источников в силу своей актуальности.

Сайт «Хабр» [21] представляет собой крупнейший в Европе ресурс для специалистов в сфере информационных технологий (ИТ), посвященный всевозможным областям данной индустрии, включая программирование и работу с реляционными базами данных. В частности, на данном сайте опубликована подробная статья [16], состоящая из 15 частей, посвященная основам проектирования реляционных баз данных. Данная статья включает в себя вводную информацию об основах проектирования реляционных баз данных, описание основных правил при создании реляционной БД, сопровождаемых их примерами.

Сайт «Microsoft Developer Network» [27] является официальным сайтом Microsoft, посвященным разработке и поддержке продуктов компании Microsoft (в том числе и платформы .NET Framework и языка программирования C# в частности). Для каждого раздела существует отдельная библиотека функций, методов и различных параметров с их подробным описанием и примерами применения. Кроме того, раздел «Сообщество» позволяет задать вопрос другим пользователям ресурса.

Сайт «Stackoverflow» [30] содержит огромное количество публичных решений задач и вопросов, связанных с программированием. Следует отметить, что данный ресурс не является «Решебником для задач», а больше нацелен на обмен опытом среди программистов и детальным разбором работы тех или иных операторов и функций.

Сайт «ProfessorWeb» [28] содержит подробное пошаговое руководство по языку C#. Данное руководство разделено на две части, каждая из которых в свою очередь разделена на модули в рамках которых рассматривается определенный набор тем. В список таких модулей входят знакомство с платформой .NET в целом, создание приложений на данной платформе, работа со сложными типами данных, перегрузка функций-членов классов, а также основные принципы объектно-ориентированного программирования и 17 неко-



торые другие тонкости работы на платформе .NET. Кроме того, ресурс содержит подробную информацию по работе с Microsoft SQL Server 2012 и языком SQL в принципе.

На обучающем портале «Интуит» [10] расположено множество курсов, в том числе бесплатных, направленных на получение знаний в сфере работы с базами данных. В частности, в рамках курса «Основы SQL» обсуждаются типы данных, допускаемые в SQL, средства определения объектов базы данных, манипулирование данными, управление соединениями, сессиями и транзакциями, обеспечение безопасности, динамический и встроенный SQL. Отдельный раздел данного курса посвящен проектированию реляционных баз данных. Также за каждой темой следует тест, проверяющий изученный материал. Несмотря на это, следует отметить, что форма тестирования не предполагает решения обучающимся практических задач, а лишь набор вопросов с несколькими заранее определенными вариантами ответов на выбор, что не позволяет закрепить практические навыки составления SQL-запросов.

Таким образом, анализ печатной литературы и интернет-источников позволил выявить два основных метода проектирования схем реляционных БД:

1. Метод декомпозиции (разбиения).
2. Метод синтеза.

Кроме того, альтернативы в качестве онлайн-курсов также не позволяют сформировать практических навыков, несмотря на наличие систем контроля знаний.

### **1.3.3 Анализ рабочей программы**

Учебная дисциплина «Базы данных» включена в учебный план по подготовке бакалавров по направлению 44.03.04 Профессиональное обучение (по отраслям). Дисциплина «Базы данных» входит в вариативную часть

учебного плана направления подготовки 44.03.04 Профессиональное обучение (по отраслям) [14].

Целью освоения дисциплины является знакомство с различными типами баз данных, их историей, а также спецификой их использования в различных видах профессиональной деятельности.

Предлагаемый курс обучения предназначен для формирования у студентов представления о назначении и возможностях баз данных различных типов и умений их эффективного применения в профессиональной деятельности.

В дальнейшем полученные знания позволяют студентам проектировать, разрабатывать и администрировать базы данных в различных сферах профессиональной деятельности.

В учебном плане подготовки бакалавров на изучение дисциплины «Базы данных» отводится 180 часов, из них 68 аудиторных часов. Дисциплина изучается на третьем курсе, в шестом семестре. Предусмотрены лекционные и лабораторные занятия в компьютерном классе. По окончании изучения дисциплины предполагается экзамен. План изучения дисциплины приведен в таблице 9.

Таблица 9 — План изучения дисциплины «Базы данных»

Виды учебной работы	Объем учебной работы в часах
1. Общая трудоемкость дисциплины	180
2. Аудиторные занятия	68
2.1 Лекции	34
2.2 Лабораторные занятия	34
3. Самостоятельная работа	112
3.1 Изучение теоретического курса	50
3.2 Подготовка к защите лабораторных работ	32
3.3 Подготовка к экзамену	30

Раздел «Основы проектирования реляционных баз данных» включает в себя следующие темы:

- типы данных, записи, база данных как целостный объект, принципы проектирования базы данных;

- системный анализ предметной области;
- параметры описания предметной области;
- этапы жизненного цикла баз данных;
- этапы проектирования базы данных;
- плоская таблица;
- понятие декомпозиции таблицы;
- понятие нормализация отношений;
- нормальные формы.

План изучения раздела «Основы проектирования реляционных баз данных» представлен в таблице 10.

Таблица 10 — План изучения раздела «Основы проектирования реляционных баз данных»

Виды учебной работы	Объем учебной работы в часах
Лекции	6
Лабораторные занятия	6
Самостоятельная работа	12

В преподавании дисциплины «Базы данных» используются следующие формы:

- лекции; лабораторные занятия, в рамках которых решаются задачи, обсуждаются вопросы лекций и домашних заданий;
- экспресс-диагностика и тестирование по отдельным темам дисциплины;
- экспресс-диагностика и тестирование по отдельным темам дисциплины;
- рейтинговая технология контроля учебной деятельности студентов для обеспечения их ритмичной работы в течение семестра;
- консультирование студентов по вопросам учебного материала.

Основной и самой результативной формой обучения дисциплине являются лабораторные занятия. Для преподавания дисциплины «Базы данных» создан учебно-методический комплекс, обеспечивающий предоставление

информации (электронные пособия и лабораторный практикум) и управление деятельностью обучаемого, включая контроль.

Разработанные лабораторные работы включают в себя обучающие тексты, набор пошаговых инструкций, учебных задач и заданий, демонстрационный материал и контрольные вопросы.

Такой подход к организации проведения лабораторного практикума стимулирует и организует самостоятельную работу студента. Достигаются не только учебные, но и развивающие цели, т.к. формируется умение учиться. Роль преподавателя смещается от транслятора учебной информации к консультанту и организатору учебной деятельности студента при формировании различных компетенций.

Задачи изучения дисциплины:

- ознакомление с историей и тенденциями развития баз данных;
- ознакомление с основными понятиями и терминологией баз данных вообще и реляционных баз данных, в частности;
- ознакомление с математическим аппаратом, используемым в указанной области человеческой деятельности, в частности, с операциями над отношениями;
- ознакомление с основными способами защиты данных в СУБД, ознакомление с проблемами и возможностями администрирования в СУБД;
- выработку умения использования основных команд SQL;
- формирование навыков практического использования баз данных для решения различных прикладных задач, а также их администрирования.

Дисциплина формирует культуру и ответственное отношение к профессиональной деятельности, поскольку она является базовой для многих форм деятельности и без нее невозможно современное производство в любой сфере жизни. Участие в сложных совместных разработках требует выработки внимательного отношения исполнителей друг к другу, умения поступиться собственными удобствами ради успешного функционирования совместно созданного продукта.

Дисциплина «Базы данных» способствует формированию компетенций, предусмотренных федеральным государственным образовательным стандартом (ФГОС).

Общепрофессиональные компетенции (ОПК):

- способность к когнитивной деятельности (ОПК-6);
- способен к самостоятельной работе на компьютере (элементарные навыки) (ОПК-5);
- способен к обоснованию профессионально-педагогических действий (ОПК-7).

Профессиональные компетенции (ПК):

- готовность к конструированию содержания учебного материала по общепрофессиональной и специальной подготовке рабочих, служащих и специалистов среднего звена (ПК-20);
- готов к применению технологий формирования креативных способностей при подготовке рабочих (специалистов) (ПК-14);
- способен проектировать и оснащать образовательно пространственную среду для теоретического и практического обучения рабочих (специалистов) (ПК-16).

Профильно-специализированные компетенции (ПСК):

- готовность к работе в специализированных программных пакетах прикладного назначения (ПСК-4);
- способность анализировать рынок программно-технических средств, информационных продуктов и услуг для решения прикладных задач (ПСК-9);
- способность проектировать базы данных и осуществлять доступ к информации (ПСК-13).

В результате освоения дисциплины студент должен знать:

- сущность моделирования, используемого при построении баз данных, в первую очередь, реляционной модели;

- основы проектирования баз данных и принципы нормализации реляционных баз данных;

- идеологию клиент-серверных технологий использования баз данных, существо распределенных баз данных и распределенной обработки данных; принципы организации транзакций;

- основы управления доступом к данным, основы оптимизации функционирования СУБД;

- основные проблемы безопасности, которые встречаются в приложениях, работающих с базами данных;

- основные принципы обеспечения безопасности баз данных.

По итогам освоения дисциплины студент должен уметь:

- пользоваться операциями реляционной алгебры;
- выполнять декомпозицию больших таблиц, проектировать базы данных;

- составлять запросы на языке SQL.

В результате освоения дисциплины обучающийся должен владеть:

- основами сопровождения систем управления базами данных и их администрирования, экспорта и импорта данных;

- способами восстановления баз данных и их отдельных объектов;

- теоретическими знаниями о способах обеспечения безопасности баз данных в современных серверных СУБД;

- навыками применения методов защиты информации в базах данных.

В результате анализа нормативной документации можно сделать вывод, что 24 часа на изучения раздела «Основы проектирования реляционных баз данных» не достаточно для полноценной отработки у обучающихся навыков проектирования реляционных баз данных.

## **2 ОПИСАНИЕ ЭЛЕКТРОННОГО ТРЕНАЖЕРА «ДЕКОМПОЗИЦИЯ ПЛОСКОЙ ТАБЛИЦЫ»**

### **2.1 Педагогический адрес**

Электронный тренажер «Декомпозиция плоской таблицы» предназначен для студентов направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля «Информационные технологии».

Разработанное педагогическое программное средство (ППС) может быть использовано при изучении раздела «Основы проектирования реляционных баз данных» дисциплины «Базы данных» в рамках аудиторных лабораторных занятий, либо в виде инструмента самостоятельного ознакомления, работы и самоконтроля по теме «Декомпозиция плоской таблицы».

### **2.2 Общие требования по созданию электронных средств обучения**

Электронный тренажер относится к электронным средствам обучения (ЭСО) и предназначен для отработки и закрепления практических умений и навыков учебной деятельности, а также позволяющий самостоятельно проводить закрепление полученных знаний [23].

Разработка электронных средств обучения, невозможна без выработки и соблюдения комплекса требований к качеству ЭСО. Более того, соблюдение таких требований является основополагающим элементом общей технологии создания ЭСО.

При создании электронных тренажеров предъявляются следующие технико-технологические требования:

1. Аппаратно-программной платформе. Создаваемое ЭСО должно быть рассчитано на стандартные аппаратно-программные платформы и информационные системы наиболее распространенные в среде использования

данных средств обучения. Решения, связанные с применением нестандартных аппаратных и программных средств, выходящие за рамки типовой конфигурации, должны быть обоснованы.

Рекомендуется обеспечить адаптируемость ЭСО к разным конфигурациям в пределах одной аппаратно-программной платформы. Разные конфигурации, например, могут определяться:

- версией операционной системы;
- типом и версией браузера;
- производительностью компьютера, видеосистемы, канала связи;
- наличием или отсутствием периферийного оборудования;
- наличием или отсутствием системных компонентов (кодеков, драйверов и т.п.);
- средствами расширения функций браузера (подключаемыми программными объектами, Java-апплетами) и др.

Адаптируемость обеспечивается путем реализации нескольких версий средства обучения для разных конфигураций либо функций настройки на текущую конфигурацию. Второй вариант является предпочтительным.

2. Телекоммуникационные технологии. Телекоммуникационные возможности ЭСО должны быть реализованы на основе стандартных средств (сетевых функций операционной системы, браузера, электронной почты, телекоммуникационных функций системы управления учебным процессом и т.д.).

Рекомендуется обеспечить универсальность ЭСО с точки зрения работоспособности в разных условиях, определяемых использованием или неиспользованием различных телекоммуникационных технологий:

- работа на локальном компьютере без выхода в сеть;
- работа в локальной вычислительной сети (работа пользователей с ЭСО независимо друг от друга; режим, поддерживающий взаимодействие пользователей);
- удаленный доступ к ЭСО через Интернет;



- удаленный доступ к ЭСО через Интернет с использованием локального мультимедийного содержания для минимизации объема передаваемой информации (технология гибридных средств обучения);

- удаленный доступ к ЭСО через Интернет с возможностью копирования части содержания на компьютер пользователя для работы в локальном режиме (технология загружаемых электронных средств обучения, исключающая необходимость наличия постоянного соединения с Интернет и позволяющая минимизировать время соединения).

Рекомендуется обеспечить адаптируемость сетевого ЭСО к скорости передачи данных по каналу связи.

3. Надежность. ЭСО должно быть завершённым средством обучения, учебным материалом, готовым к практическому применению при этом обеспечивая защиту от некорректных действий пользователя, способных вызвать нарушение его допустимого режима функционирования, повреждение данных, сбой в работе вычислительной системы и т.п.

В ЭСО должны быть предусмотрены средства обработки нештатных ситуаций, вызванных некорректным функционированием:

- системного программного обеспечения;
- сопряженных приложений;
- аппаратного обеспечения (в том числе периферийных устройств);
- телекоммуникационного оборудования;
- системы управления учебным процессом;
- средств взаимодействия между клиентскими и серверными компонентами.

Рекомендуется реализовать в ЭСО функции контроля наличия и корректной работы критически важных системных компонентов (драйверов, кодеков и т.д.). Такой контроль следует выполнять либо непосредственно после запуска ЭСО, либо перед вызовом функций, использующих эти компоненты.

4. Ресурсоемкость и производительность. Требования ЭСО к ресурсам вычислительной системы (типу и тактовой частоте процессора, объему оперативной и дисковой памяти, объему видеопамяти и др.) должны быть обоснованными и соответствовать обеспечиваемым им функциям и характеристикам. По завершении работы ЭСО должен полностью освободить использовавшиеся им ресурсы вычислительной среды.

5. Информационной безопасности. В ЭСО должны быть предусмотрены средства защиты информации от несанкционированного доступа (чтения и (или) изменения неуполномоченными лицами или неустановленными способами). К защищаемым информационным компонентам относятся:

- информация о ходе и результатах работы обучаемого с ЭСО (протоколы, оценки);
- индивидуальные задания обучаемых;
- ответы на учебно-тренировочные задачи, подсказки, другие виды информационной помощи, относящиеся к изучаемому курсу (дисциплине) и используемые в контрольно-аттестационных мероприятиях;
- персональная информация об обучаемом;
- профиль обучаемого (данные персональной настройки ЭСО, закладки, «история» работы и т.д.).

Программные компоненты ЭСО должны соответствовать единому уровню информационной безопасности и использовать согласованные программно-технические решения, связанные с ее обеспечением.

6. Сопровождаемость. Рекомендуется обеспечить возможности изменения ЭСО разработчиками (исправления ошибок, адаптации к изменениям, добавления и обновления учебного материала и т.д.). Возможности изменения обуславливаются как технологией реализации, так и документацией. Обеспечение изменяемости ЭСО (широты и удобства соответствующих возможностей) способствует увеличению срока их эффективного использования в обучении, повышению качества и ценности, что должно находить адекватное отражение в экспертизе ЭСО.

Рекомендуется, чтобы программные компоненты электронных средств обучения обеспечивали возможности для их модернизации, связанной с переходом к новым или усовершенствованным аппаратно-программным платформам, реализацией новых информационных технологий и т.д.

7. Дистрибутив и установка. Электронные средства обучения, создаваемые педагогами, могут распространяться в бумажном виде, на информационных носителях или через телекоммуникационные сети.

Дистрибутив ЭСО, записанный на информационных носителях, должен включать:

- упакованную версию ЭСО для установки и программу установки;
- метаданные;
- краткое руководство по установке ЭСО;
- лицензионное соглашение (в том числе для свободно распространяемых средств обучения);
- эксплуатационную документацию;
- средства автозапуска.

Рекомендуется обеспечить возможность работы с ЭСО непосредственно с дистрибутивного носителя, т.е. без установки средства обучения на компьютер пользователя [18].

### **2.3 Требования к тренажеру «Декомпозиция плоской таблицы»**

В соответствии с выделенными требованиями к электронным тренажерам и проведенным анализом литературы и интернет-источников к разрабатываемому ППС были выставлены следующие требования:

1. Система должна позволять создавать новые задания, а также загружать ранее сохраненные задания на основе кроссплатформенная библиотеки SQLite [29].

2. В системе должна быть реализована возможность сохранения результата работы в тренажере для дальнейшего продолжения.

3. Система должна поддерживать следующие режимы функционирования:

- свободный режим — режим работы, при котором отсутствуют задания и какие-либо подсказки по работе с тренажером.

- режим обучения — режим работы, при котором выводится пример задания с полным разбором посредством подсказок всех действий, выполняемых пользователем во время работы с тренажером.

- режим контроля знаний — режим работы, при котором обучающийся закрепляет полученные на семинарских занятиях теоретический материал, выполняя заранее подготовленное задание с дальнейшим оцениванием результата выполненной работы с тренажером.

- режим редактирования — режим работы, при котором доступно редактирование системных таблиц для создания новых или редактированию ранее созданных заданий.

4. Взаимодействие пользователей с системой должно осуществляться посредством визуального графического интерфейса (GUI) с возможностью масштабирования окна тренажера.

5. Значимые элементы управления GUI должны содержать клавиатурные сокращения (иметь «горячие клавиши»).

6. Система должна поддерживать возможность отправки результатов выполненной работы на электронную почту.

7. При работе в редакторе SQL-запросов должны быть доступны следующие команды:

- SELECT — возвращает строки из базы данных и позволяет делать выборку одной или нескольких строк или столбцов из одной или нескольких таблиц;

- INSERT — добавляет одну или несколько строк в таблицу или представление;

- DELETE — удаляет одну или несколько строк из таблицы или представления;
- UPDATE — изменяет существующие данные в таблице или представлении;
- DROP — удаляет одно или больше определений таблиц и все данные, индексы, триггеры, ограничения и разрешения для этих таблиц;
- CREATE — создает новую таблицу.

8. В системе должна быть реализована структурированная справка по работе с тренажером, а также наличие теоретического материала описывающего основы создания реляционных баз данных.

9. Система должна поддерживать полноценный редактор SQL-запросов с возможностью подсвечивания системных тегов и комментирования ранее написанных запросов.

10. Система должна позволять просматривать ранее создание таблицы и быстро выводить содержание этих таблиц без дополнительных SELECT запросов.

## 2.4 Описание средств разработки

В качестве платформы разработки тренажера была выбрана платформа Microsoft.NET Framework 4 [19]. Это обусловлено следующими факторами:

1. Простотой интеграции в используемые в процессе обучения системы.
2. Поддержкой данной версии платформы на операционных системах (ОС) Microsoft Windows 7 и выше.

В качестве языка программирования для разработки пособия был выбран объектно-ориентированный язык C# 4.0 [17]. Это обусловлено следующими факторами:

1. Объектной ориентированностью данного языка программирования.
2. Простотой разработки приложений с графическим интерфейсом.

3. Привычностью графического интерфейса разрабатываемых программных средств для обучающихся.

4. Поддержка всеми популярными настольными операционными системами.

5. Отсутствие необходимости устанавливать полученный электронный тренажер на компьютер. Достаточно будет скопировать папку с исполняемым файлом.

В качестве среды разработки электронного тренажера была выбрана среда Microsoft Visual Studio 2017. Это обусловлено следующими факторами:

1. Официальной поддержкой выбранной платформы разработки в данной среде разработки.

2. Удобством интерфейса среды разработки.

3. Широким функционалом для разработки графических приложений.

В ходе разработки пособия используются следующие технологии разработки:

1. Коллекции типов Dictionary и List [8]. Используются для удобства хранения однотипных данных.

2. LINQ-запросы [20]. Используются для оптимизации процессов выборки данных из коллекций.

3. Регулярные выражения (RegEx) [15]. Используются в процессе анализа составленных обучающимся сущностей.

4. Обработчики событий. Используются для обеспечения работы графического интерфейса.

5. Алгоритм считывания и записи данных в файлы базы данных. Используется для открытия ранее созданных файлов заданий или создания новых.

6. Библиотека FastColoredTextBox [26] выполняет роль редактора SQL-запросов. Используется для подсветки синтаксиса и дополнительных возможностей форматирования SQL-запросов.

В качестве среды для создания файла со справочным материалом была выбрана программа Dr.Explain версии 5.7.

Это обусловлено следующими факторами:

1. Простотой наполнения материалом.
2. Возможностью экспорта созданной справки в стандартном для Windows формате справки [25].

## **2.5 Структура электронного тренажера**

Структура тренажера изображена на рисунке 2 и представляет собой две основных графических формы и трех дополнительных сопутствующих окон ввода или показа информации:

1. Форма «Работа с таблицами».
2. Форма «Справка».
3. Окно «Авторизации».
4. Окно «Идентификации».
5. Окно «О программе».

Основная работа с тренажером ведется в форме «Работа с таблицами». Данная форма содержит элементы управления для составления и выполнения SQL-запросов, а также для создания, загрузки и сохранения заданий, также на форме присутствует вкладка для переключения режимов работы с тренажером. Список доступных режимов работы:

- свободный режим;
- режим обучения;
- режим контроля знаний.

Форма «Справка» представлена в виде отдельного документа с расширением \*.chm и содержит основную информацию о тренажере и работе в нем, а также теоретический материал, содержащий вводную информацию о проектировании реляционной базы данных.



Рисунок 2 — Структура электронного тренажера

Вследствие использования при разработке объектно-ориентированного языка программирования, код программы распределен по классам, в зависимости от выполняемых функций. Разрабатываемое пособие содержит следующие классы:

1. Program — класс инициализации программы. Отвечает за открытие формы приветствия (класс IntroForm) при запуске программы.

2. IntroForm — класс формы приветствия. Содержит набор обработчиков событий для действий с элементами управления на данной форме. Содержит операции для вызова формы «Работа с таблицами» (класс MainForm) и завершения работы с пособием. Графический интерфейс формы приветствия представлен на рисунке 3.



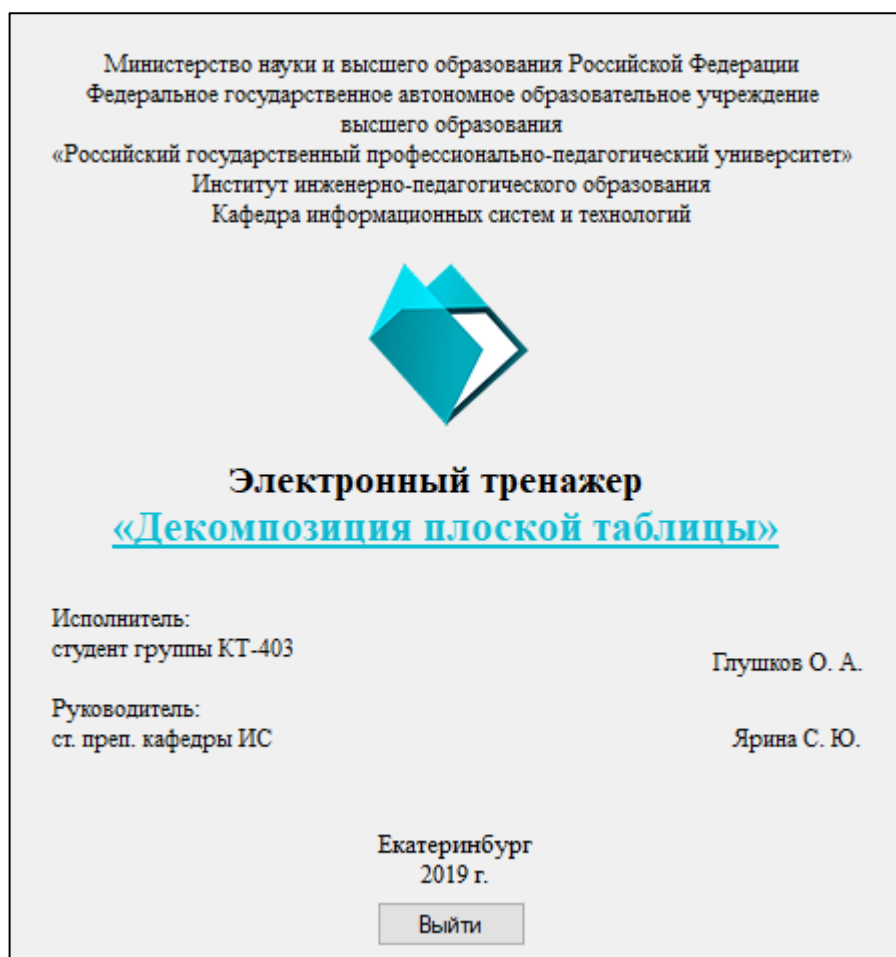


Рисунок 3 — Графический интерфейс формы приветствия

3. MainForm — класс формы «Работа с таблицами». Содержит набор обработчиков событий для действий с элементами управления на данной форме. Содержит элемент управления режимами работы тренажеры, панель инструментов обеспечивающая быстрый доступ к ключевым функциям программ, редактор SQL-запросов с подсветки синтаксиса, поля отображения задания, результатов запросов и статуса выполненного задания, а также наглядный список созданных таблиц с возможностью просмотра содержания таблицы без дополнительных SQL-запросов.

Также содержит операции для вызова формы «Авторизации» (класс AuthDeveloperForm), формы «О программе» (класс AboutForm) и формы «Идентификации» (класс AuthUserForm).

Графический интерфейс формы «Работа с таблицами» представлен на рисунке 4.

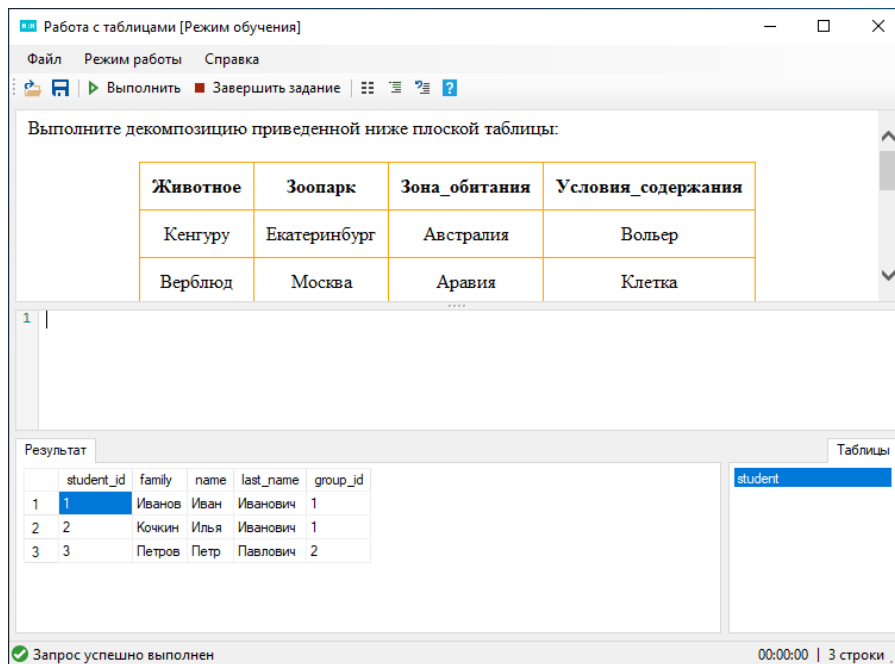


Рисунок 4 — Графический интерфейс формы «Работа с таблицами»

4. AuthDeveloperForm — класс формы «Авторизация». Предназначен для аутентификации пользователя в режиме редактирования задания (рисунок 5).

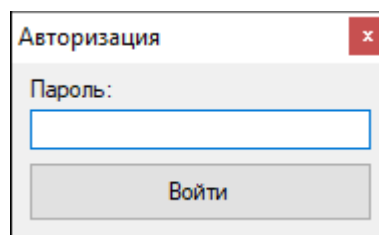


Рисунок 5 — Графический интерфейс формы «Авторизации»

5. AuthUserFrom — класс формы «Идентификация». Предназначен для установления фамилии и имени, а также группы обучающегося выполняющего задание с отправкой результатов на почту преподавателя (рисунок 6).

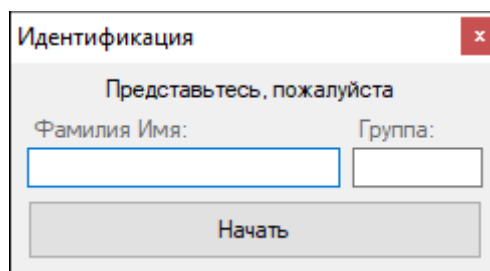


Рисунок 6 — Графический интерфейс формы «Идентификация»

6. AboutForm — класс формы «О программе». Содержащий основную информацию об авторах (рисунок 7).

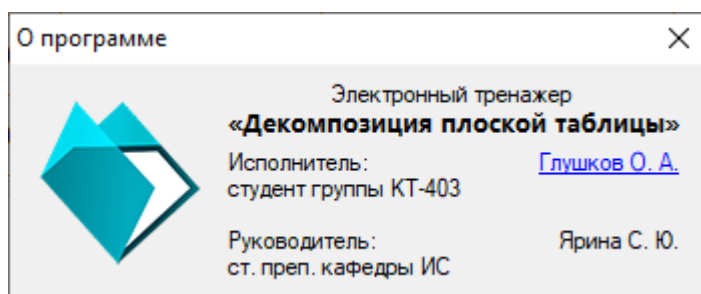


Рисунок 7 — Графический интерфейс формы «О программе»

7. Table — класс объекта таблицы. Данные объекты добавляются и удаляются в коллекцию в ходе работы программы.

8. Task — класс объекта «Задание». Используется для хранения данных, считанных из HTML-документа, содержащего практические задания.

9. Task — класс объекта «Задание». Используется для хранения данных, считанных из файла задания при его загрузке. Хранит такие данные, как вариант задания, текст задания, контрольные значения для проверки выполненного задания.

## 2.6 Режимы работы тренажера

### 2.6.1 Свободный режим

Данный режим предназначен для свободной работы с тренажером, он максимально приближен к работе в среде управления базами данных SQL Server Management Studio (SSMS) используемый студентами при изучении дисциплины «Базы данных», что позволяет обучающимся не тратить время на изучения нового интерфейса. В нем нет никаких заданий и статистики, нет подсказок. Основная работа в данном режиме видется с полем ввода SQL-запросов и вывода результатов.

На рисунке 8 представлен интерфейс формы в свободном режиме.

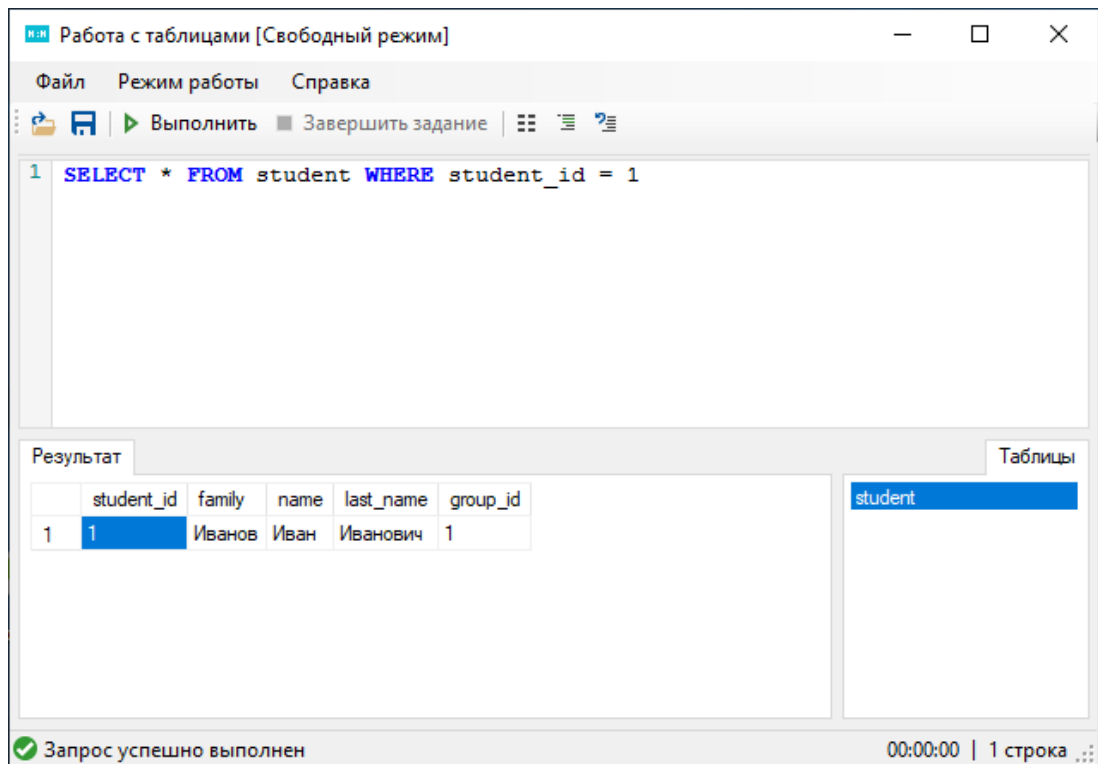


Рисунок 8 — Интерфейс формы в «Свободном режиме»

В свободном режиме обучающихся может отрабатывать навыки написания SQL-запросов, а также изучать синтаксис команд при их написании.

### 2.6.2 Режим обучения

Предназначен для получения начальных навыков работы с тренажером, содержит пример задания с полным разбором действий выполняемых обучающимся во время работы с тренажером.

В случае затруднения обучающихся может воспользоваться подсказкой посредством нажатия соответствующей кнопки на панели быстрого доступа. В зависимости от этапа содержание подсказки может меняться.

Например, обучающихся дошел до этапа написания SQL-запроса, при первом запросе помощи обучающему будет выведена подсказка с указанием действия, которое должен совершить обучаемый на данном этапе.

Пример подсказки представлен на рисунке 9.

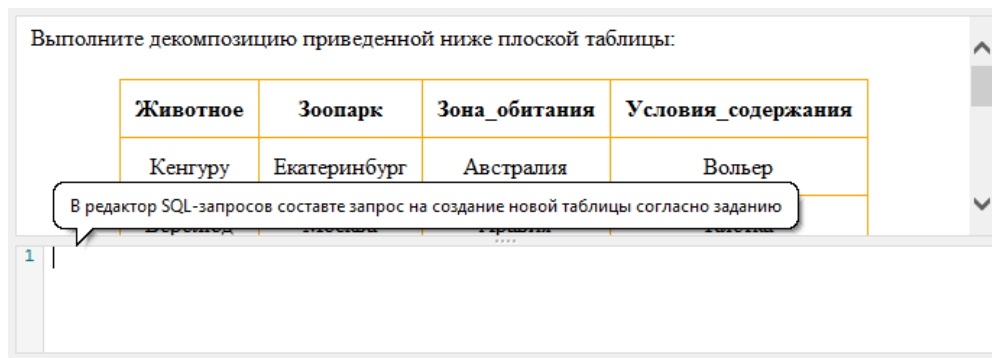


Рисунок 9 — Краткая подсказка с указанием действия

В случае дальнейших затруднений тренажер выведет более подробный текст подсказки с указанием команды для ввода. Пример такой подсказки показан на рисунке 10.

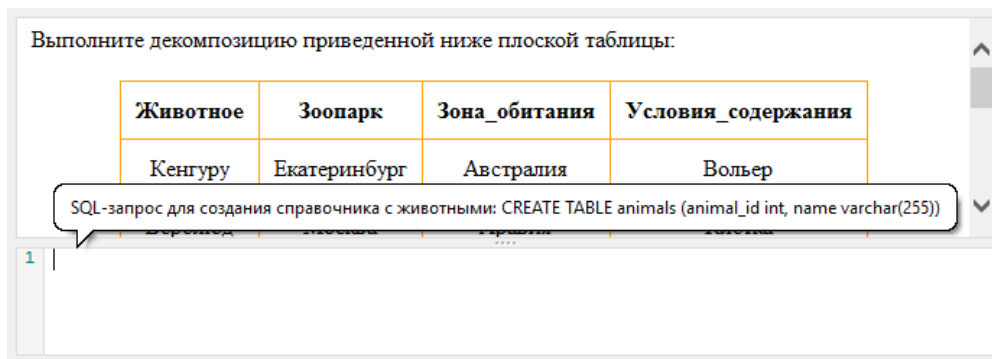


Рисунок 10 — Подробная подсказка с указанием команды

Данный режим предполагает параллельную работу со справкой для упрощения процесса изучения.

### 2.6.3 Режим контроля знаний

Предназначен для проверки полученных обучающимся навыков по декомпозиции плоской таблицы. Данный режим приближен к работе в реальной системе, т.е. в нем отсутствуют подсказки при выполнении задания. Это позволяет более справедливо оценивать работу обучающегося, ему предоставляется заранее подготовленное задание.

Итоговый результат выполнения задания обучающимся может быть отправлен на почту преподавателя.

Интерфейс тренажера в режиме контроля знаний представлен на рисунке 11.

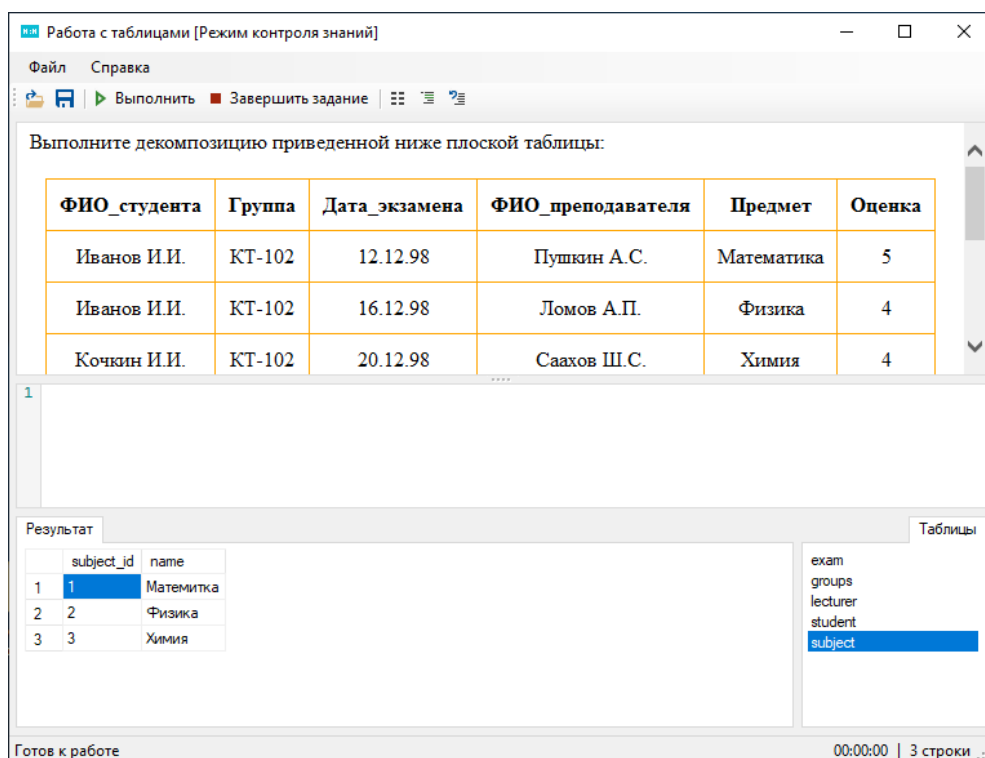


Рисунок 11 — Интерфейс формы в «Режиме контроля знаний»

Выполнив задание, пользователь получает итоговую оценку и результат разбора ошибок при их наличии.

#### 2.6.4 Режим редактирования

Предназначен для создания новых заданий или редактированию ранее созданных. В режиме редактирования открывается доступ к изменению четырех системных таблиц, три из которых создаются автоматически при открытии окна тренажера, а четвертая создается преподавателем при составлении нового задания.

Во всех остальных режимах редактирование системных таблиц запрещено. Интерфейс тренажера в режиме редактирования представлен на рисунке 12.

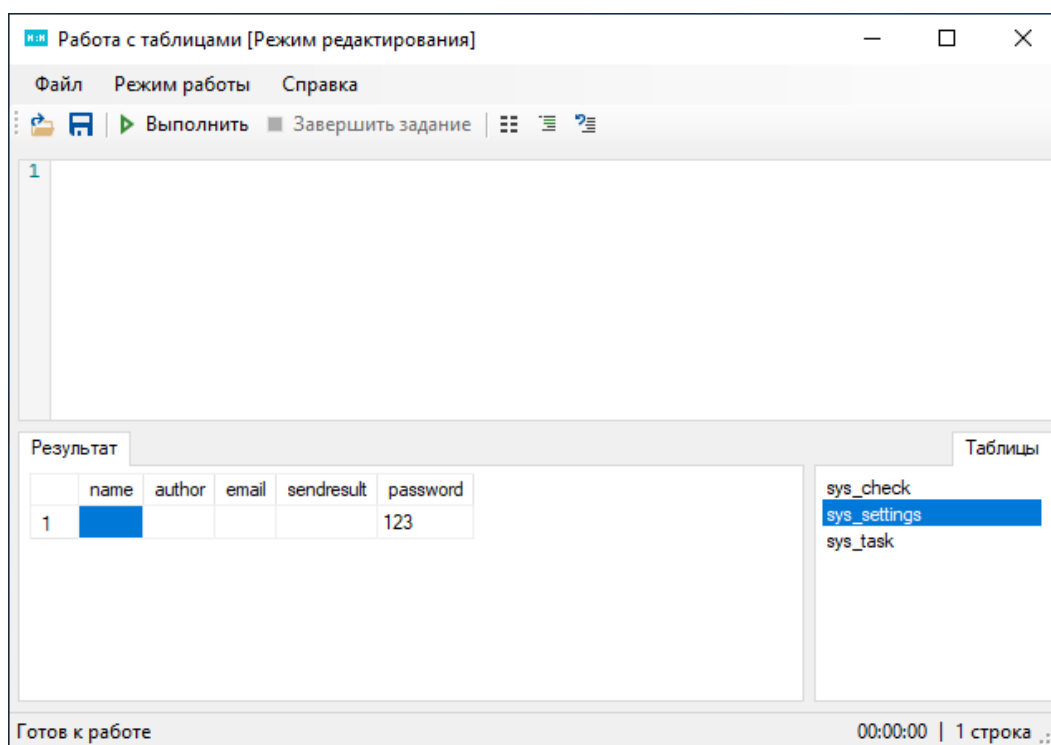


Рисунок 12 — Интерфейс формы в «Режим редактирования»

Список выводимых системных таблиц в режиме редактирования:

1. Таблица настроек задания — `sys_setting`. Данная таблица включает в себя несколько полей при изменении которых меняются настройки задания.

Список доступных полей для редактирования:

- `name` — наименования задания, используется при отправке результатов на почту;
- `author` — автор созданного задания;
- `email` — адрес электронной почты, на который будут приходить результаты выполненного задания;
- `sendresult` — параметр отправки результатов выполненного задания, принимает значения: 0 — не отправлять результата, 1 — отправлять;
- `password` — пароль доступа к редактированию задания, при первом создании задания устанавливается по умолчанию — 123.

2. Таблица с контрольными значениями — `sys_check`. Предназначена для заполнения эталонных таблиц с контрольными значениями используемые для проверки выполненного задания.

Эталонная таблица представляет из себя одну таблицу из разобранного варианта плоской таблицы с указанием характеристик данной таблицы.

Список доступных полей для редактирования:

- `table_name` — наименование таблицы, служит для разграничения созданных эталонных таблиц;
- `columns_count` — количество колонок в эталонной таблицы;
- `directories_count` — количество справочников, используемых в эталонной таблице;
- `primarykey_count` — количество ключевых полей, используемых в эталонной таблице;
- `records_count` — количество записей в эталонной таблице;
- `indicator` — параметр в котором указывается некоторое слово используемое для поиска сравниваемой таблицы в списке созданных обучающимся при выполнении задания;

3. Таблица с формулировкой задания и примечания — `sys_task`. Используется при форматировании задания.

Все задания имеет общую структуру в виде HTML-документа с изменяемыми значениями полей формулировки задания, примечания и обязательных требований (рисунок 13).

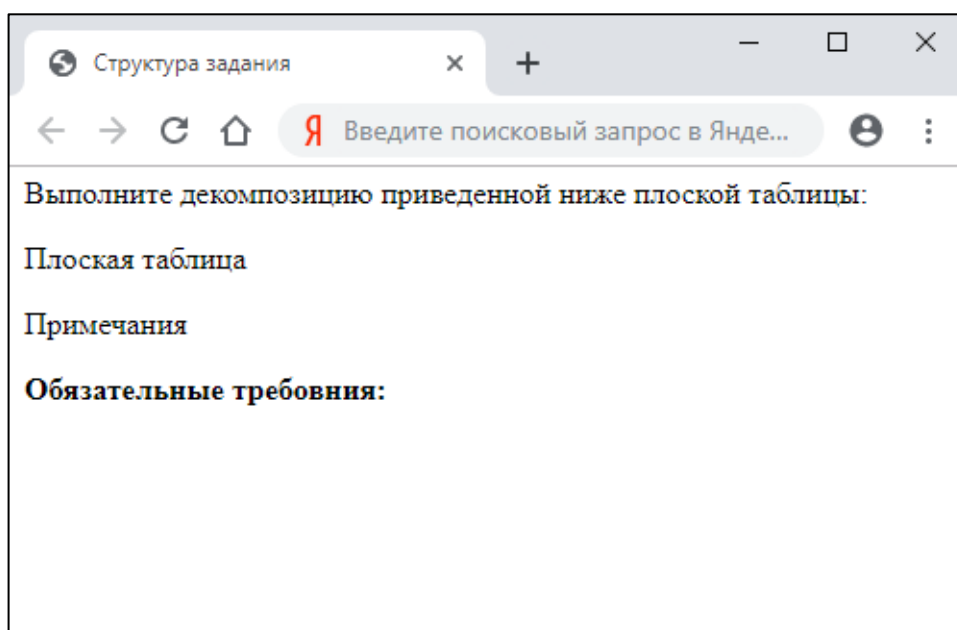


Рисунок 13 — Вид структуры задания



4. Таблица — sys\_table создается преподавателем при составлении задания. В ней указывается плоская таблица, выводимая в формулировке задания вместо слов «Плоская таблица» в структуре задания на рисунке 13.

Режим редактирования становится доступным только при правильно введенном пароле доступа в окне «Авторизации», изображённом на рисунке 5.

## 2.7 Сопроводительная документация

Файл справка представляет собой единый документ с расширением \*.chm. Данный формат разработан компанией Microsoft для создания справок программ, работающих в любых версиях системы Windows, начиная с Windows 98 без установки какого-либо дополнительного программного обеспечения.

В данном файле содержатся:

1. Теоретический материал по теме «Проектирования реляционных баз данных» с описанием основных терминов, а также представлен небольшой пример по проектированию реляционной базы данных на основе метода декомпозиции (рисунок 14).

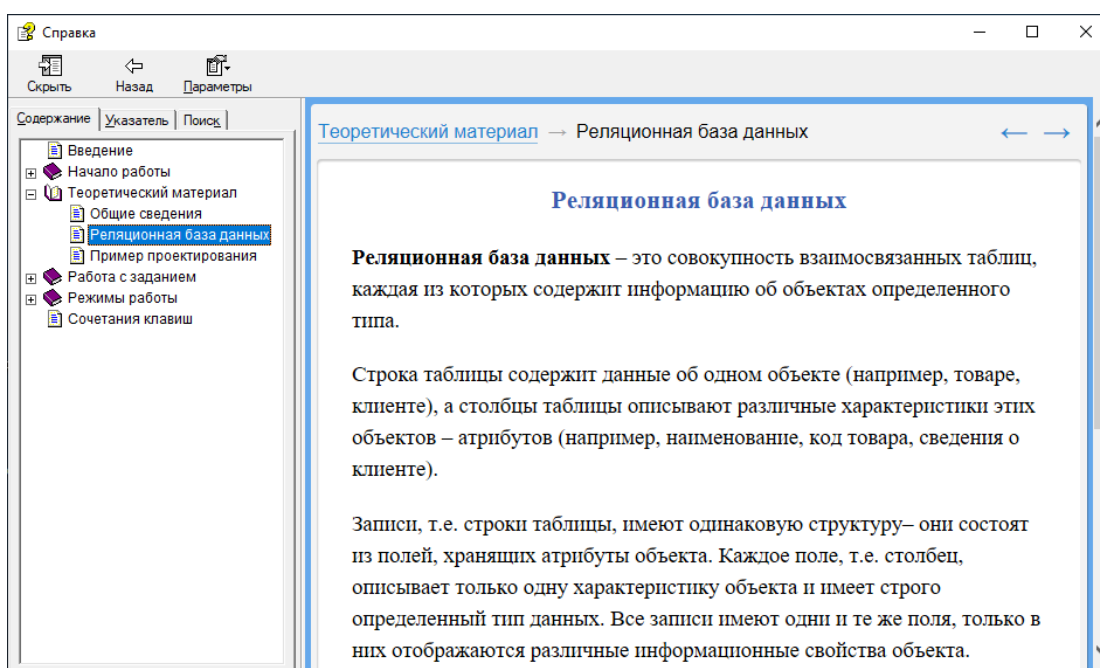


Рисунок 14 — Теоретический материал в документе «Справка»

2. Подробное описание интерфейса тренажера с указанием всех элементов управления, их назначения и дополнительных сведений, при необходимости. На страницах описания работают гиперссылки, упрощающие процесс изучения интерфейса (рисунок 15).

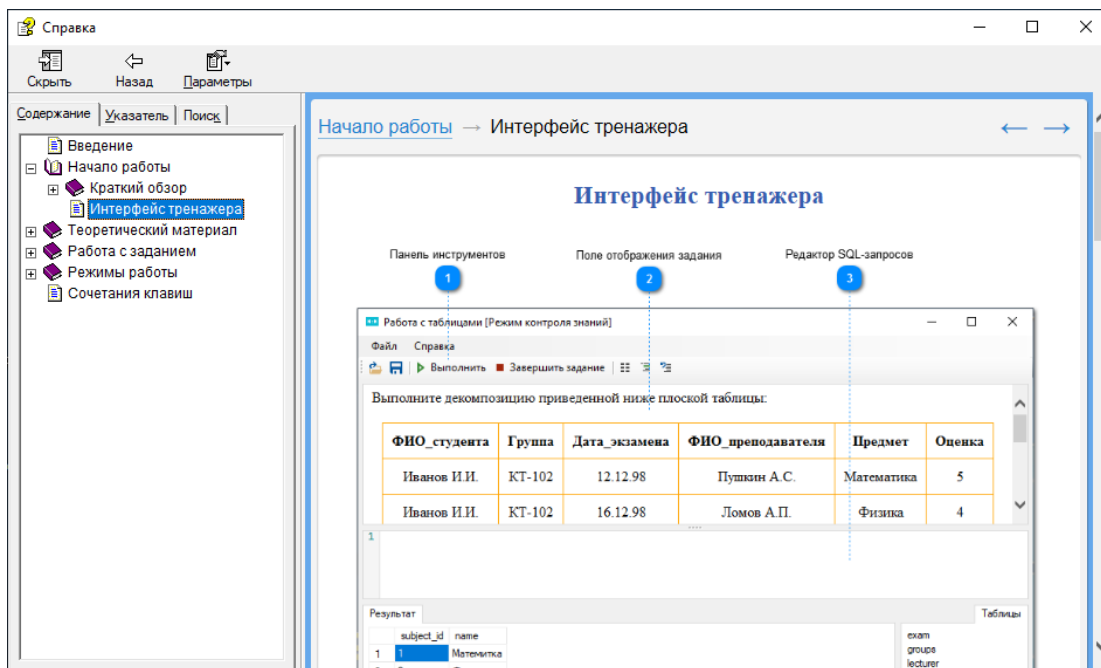


Рисунок 15 — Описание интерфейса тренажера в документе «Справка»

3. Описаны основные действия по работе с заданиями и самим тренажером (рисунок 16).

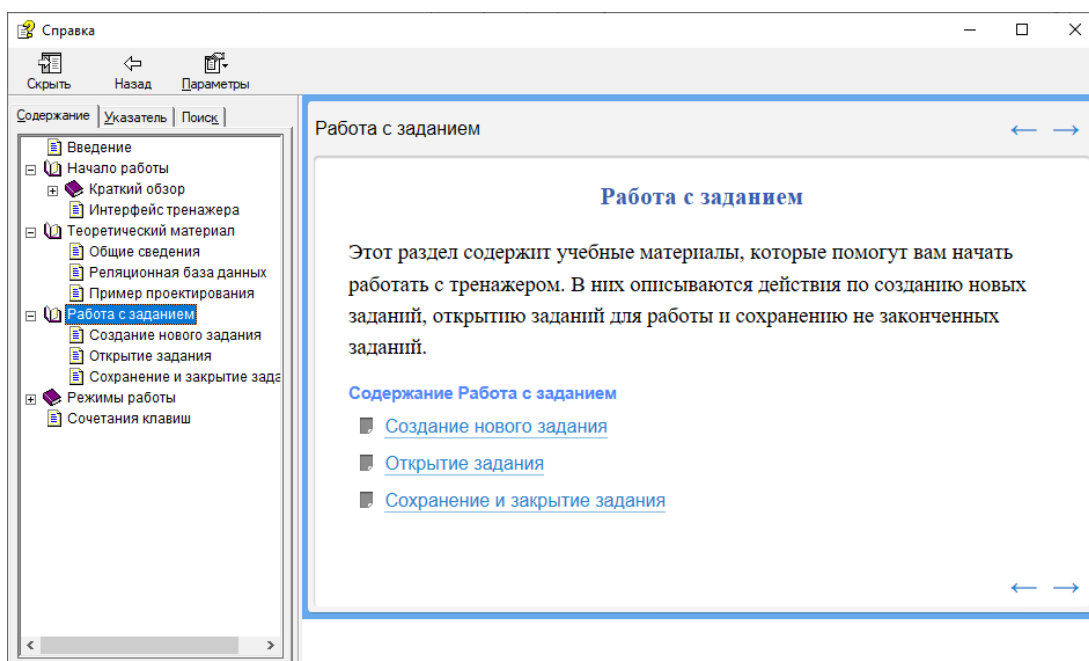


Рисунок 16 — Учебный материал по работе с заданиями и тренажером

При отсутствии файла справки в директории тренажера, использование справки будет недоступно.

## 2.8 Описание работы с тренажером

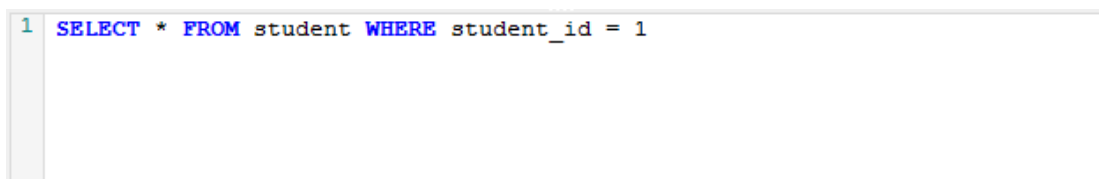
Запуск тренажера осуществляется при помощи исполняемого файла программы — `DecompilingFlatTable.exe`. При запуске происходит создание нового файла задания во временной папке текущего пользователя. После успешного запуска тренажера на экране появится основная форма программы в свободном режиме.

Для начала обучения пользователю рекомендуется открыть файл справки, нажав соответствующую кнопку на основной форме или воспользоваться клавишей F1.

В справочной документации подробно описана основная информация по работе с тренажером, а также содержится теоретический материал и указан рекомендуемый порядок работы с тренажером.

При выборе одного из режимов, произойдет настройка главного окна в соответствии выбранному режиму.

Основная работа с тренажером осуществляется при помощи поля редактора SQL-запросов, изображенного на рисунке 17.



```
1 SELECT * FROM student WHERE student_id = 1
```

Рисунок 17 — Поле редактора SQL-запросов

Формулировка загруженного задания выводится в поле отображения задания (рисунок 18) с изменением структурных переменных на соответствующие в системных таблицах данных. Если в системной таблице не указана формулировка задания или примечание используются значения по умолчанию. В случае отсутствия у ранее созданного задания обязательных условий при выполнении, то поле обязательные условия не выводиться.

Выполните декомпозицию приведенной ниже плоской таблицы:

ФИО_студента	Группа	Дата_экзамена	ФИО_преподавателя	Предмет	Оценка
Иванов И.И.	Кт-102	12.12.98	Пушкин А.С.	Математика	5
Иванов И.И.	Кт-102	16.12.98	Ломов А.П.	Физика	4
Кочкин И.И.	Кт-102	20.12.98	Саахов Ш.С.	Химия	4
Петров П.П.	Ию-101	12.12.98	Пушкин А.С.	Математика	3

Примечания

**Обязательные требования:**

1. Все ключевые поля должны иметь структуру \* \_id.
2. Ни одно из полей во всех отношениях не должно содержать групп значений.  
Например, недопустимо существование поля ФИО, если в нем указаны фамилия, имя, отчество (или инициалы)

Рисунок 18 — Поля отображения формулировки задания

Для выполнения составленного SQL-запроса можно воспользоваться панелью быстрого доступа, изображенной на рисунке 19, или использовать клавишу F9.



Рисунок 19 — Панель быстрого доступа

При успешном выполнении запроса в строке статуса будет выведено соответствующее сообщение (рисунок 20), иначе будет выведено сообщение о неуспешном выполнении с указанием ошибки (рисунок 21) на английском языке ввиду отсутствия у используемой библиотеки SQLite поддержки русского языка.



Рисунок 20 — Успешное выполнение запроса

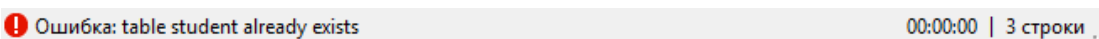


Рисунок 21 — Неуспешное выполнение запроса

Для завершения работы с заданием и проверки правильности выполнения необходимо нажать кнопку «Завершить задание» на панели быстрого доступа (рисунок 19). В окне вывода результата будет подведен итог выполнения задания с оценкой и подробным описанием ошибок при их наличии.

По завершению выполненное задание можно сохранить нажатием соответствующей кнопки на панели инструментов или используя вкладку «Файл — Сохранить как» или воспользоваться комбинацией клавишами Ctrl + S.

## 2.9 Описание основных алгоритмов программы

В основе работы разработанного тренажера лежат процессы создания и считывания базы данных, анализа результатов работы обучающегося, а также разбора и выполнения составленного обучающимся SQL-запроса.

1. Алгоритм создания нового задания. Данный алгоритм инициализируется при запуске тренажера. В результате работы алгоритма создается новый файл базы данных с расширением \*.db во временной папке текущего пользователя и указывается как источник данных при инициализации библиотеки SQLite. После успешного соединения библиотеки с базой данных в ней создаются системные таблицы:

- sys\_setting — таблица настроек задания;
- sys\_check — таблица с контрольными значениями;
- sys\_task — таблица с формулировкой задания и примечания.

Системные таблицы позволяют интерпретировать текущую базу данных как задание для электронного тренажера «Декомпозиция плоской таблицы». В конце данного алгоритма открывается главное рабочее окно тренажера «Работа с таблицами» в свободном режиме.

2. Алгоритм считывания задания. При загрузке задания алгоритм анализирует и считывает структуру системных таблиц. В результате считывания тренажер настраивает основные режимы работы согласно значениям в системных таблицах. Форма «Работы с таблицами» переводится в режим контроля знаний и происходит считывание системных таблиц «sys\_table» и «sys\_task» в которых указаны данные задания для вывода в поле отображения задания. При считывании системных таблиц пользователю не отобража-

ется информация о вариантах ответов или решении. Эти данные используются только при сравнении полученных обучающимся результатов работы с заданием.

3. Алгоритм анализа выполнения задания. После загрузки задания для контроля знаний и выполнения представленного задания, инициализируется алгоритм оценки работы. В данный алгоритм входят следующие этапы:

3.1. Проверка созданных таблиц. Создание пользователям таблицы сравнивается с каждой эталонной таблице в поисках совпадений в значении поля «indicator», который служит признаком определения пользовательской таблице, если таблица не найдена в эталонных таблица, то выводится соответствующее сообщение об это. После это все найденная таблица сравнивается со списком контрольных значений из эталонных таблиц «sys\_check». Если созданная таблица соответствует контрольным значениям, то она считается корректной. Иначе, на основе различий с контрольными значениями в эталонной таблице выполняется анализ возможных ошибок. Например, отсутствие справочников/таблиц/полей, недостаток записей по сравнению с исходными значениями и т.д.

3.2. Подсчет ошибок выполнения задания. В результате анализа созданных обучающимся таблиц, тренажер высчитывает в пятибалльной системе оценку выполнения задания по средствам вычитывания из максимальной оценки «5» ошибок совершенных обучающимся во время выполнения задания. Каждое нарушение имеет определенное значение для результирующей таблицы. Например, при отсутствии одной из обязательных таблиц задание является не решенным, так как декомпозиция была выполнена с нарушением целостности базы данных, что противоречит правилам декомпозиции. Однако при правильно выполненной декомпозиции отсутствии записей в полученной таблице не является грубым нарушением, так как не нарушает структуру полученной таблице.

Выводимые ошибки и количество баллов, вычитаемое при их нарушении, представлены в таблице 11.

Таблица 11 — Разбор выводимых ошибок

Текст ошибки	Описание	Количество вычитаемых баллов
Наличие лишних записей	Количество записей в составленной таблице превышает исходное значение	0.5
Присутствуют не все записи	Количество записей в составленной таблице не соответствует исходным значениям	0.6 за каждую отсутствующую запись
Избыточное количество справочников	В составленной таблице присутствует излишние справочники	1.0
Присутствуют не все справочники или некорректно заданы значения	В составленной таблице отсевают указатели на необходимые справочники или они указаны неверно	1.0 за каждый отсевающий справочник
Наличие избыточного количество полей	Количество полей в составленной таблице превышает исходные значения	0.6
Присутствуют не все поля	Количество полей в составленной таблице меньше исходных значений	0.6 за каждое отсеваемое поле
Не задано ключевое поле	В составленной таблице отсутствует обязательное ключевое поле	2.0

По завершению данных этапов, пользователю отображаются итоговый результат, сообщающий его результат, оценка, полученная в результате выполнения задания и краткий разбор ошибок в случае, если в решении присутствуют ошибки. Пример итогового результата изображен на рисунке 22.

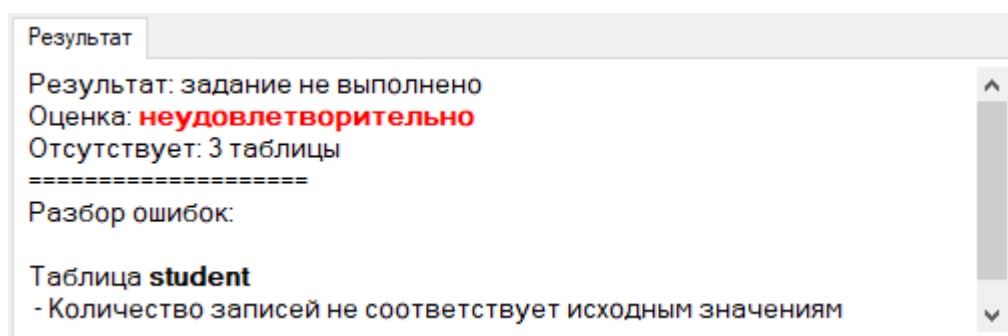


Рисунок 22 — Итоговый результат выполнения задания

4. Алгоритм отправки результатов на почту. Данный алгоритм отправляет полученный результат обучаемого на почту при условии указания почты преподавателя и равенности значения поля «sendresult» = 1 в системной таблице «sys\_settings». Алгоритм отправляет почту по протоколу Simple Mail Transfer Protocol (SMTP) через почтовый сервис Mail.ru в режиме шифрова-

ния данных SSL. Ниже представлен пример сообщения отправленного тренажера о результатах обучающегося изображен на рисунке 23.

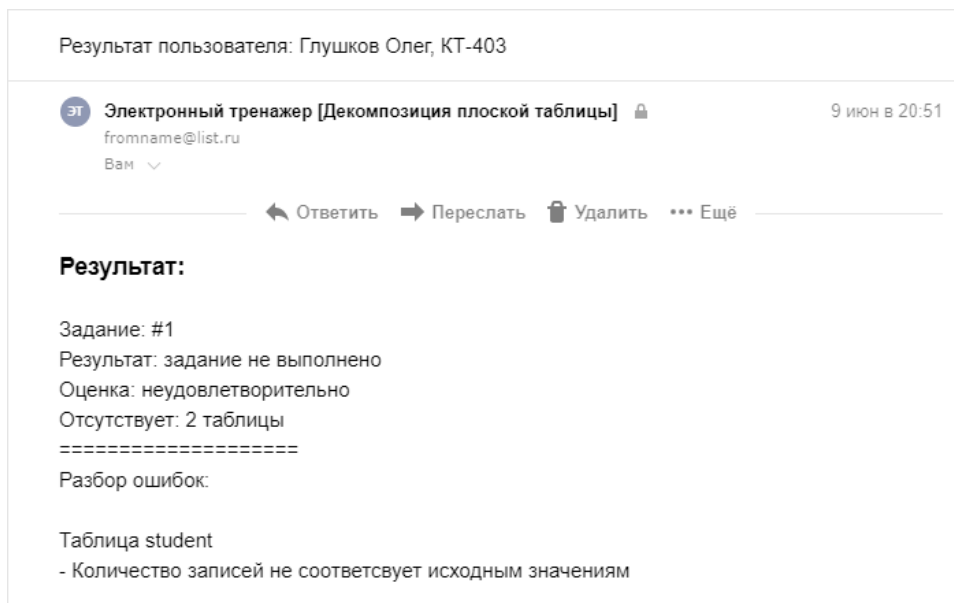


Рисунок 23 — Сообщение с результатами обучающегося на почте

## 5. Алгоритм разбора и выполнения составленного SQL-запроса.

Данный алгоритм инициализируется при нажатии пользователем кнопки «Выполнить» на форме «Работа с таблицами». Алгоритм разбивает многострочный запрос на однострочные и выполняет их средствами библиотеки SQLite. При этом производятся следующие этапы:

5.1 Анализ SQL-запроса. В запросе определяется начальный оператор в зависимости от которого будет выбрана процедура для выполнения данного запроса. Пропускаются все закомментированные строки.

5.2 Поиск имени к обращаемой таблицы. В зависимости от начального оператора выполняется разбор запроса и поиск имени таблицы для дальнейшей работы с ней.

5.3 Выполнение запроса и вывод результата. Выполнение запроса происходит средствами библиотеки SQLite, результат выполнения обрабатывается и выводится в поле вывода результатов.

5.4 Выделение имени таблицы в списке созданных таблиц, как индикатор работы с ней. Происходит это для облегчения понимания с какой таблицей сейчас идет работа.



При выполнении оператора CREATE происходит добавление имени таблицы в список созданных таблиц и автоматически выполняется SELECT запрос на вывод данной таблицы в поле вывода результатов SQL-запросов.

При выполнении оператора DROP происходит удаление таблицы из списка созданных таблиц с последующим перемещением индикатора выбранной таблиц на одну таблицу ниже.

Любой запрос выполняемый в тренажере автоматически обновляет содержание поля результата SQL-запроса путем выполнения SELECT запроса на вывод всего содержания.

б. Алгоритм переключения между таблицами. Инициализации алгоритма происходит при щелчке мыши по списку созданных таблиц. В результате данный алгоритм проверяет выбранную таблицу на наличие ее в базе данных и в случае успешного нахождения выполняет SELECT-запрос на вывод всего содержания таблицы.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был разработан электронный тренажер «Декомпозиция плоской таблицы» предназначенный для студентов третьего курса направления подготовки 44.03.04 Профессиональное обучение (по отраслям) профиля «Информационные технологии», а также набор из 30 практических заданий разной степени сложности, формирующих базу для дальнейшего развития возможностей внедрения, разработанного тренажера в процесс обучения.

В рамках работы были решены следующие задачи:

1. Проанализирована литература и интернет-источники по теме «Декомпозиция плоской таблицы» с целью формирования круга печатных и электронных изданий, необходимых для создания тренажера.

2. Рассмотрены требования, предъявляемые к электронному учебному пособию и программам-тренажерам. Критерии учитывались для того, чтобы студент мог эффективно воспринимать информацию.

3. Продумана структура классов и форм, а также реализован графический интерфейс тренажера «Декомпозиция плоской таблицы».

4. Создан набор базовых заданий для режима самостоятельной работы в разработанном тренажере. Данный набор включает 30 заданий различной степени сложности.

В качестве дальнейших перспектив исследования по данной проблеме можно расширить функционал тренажера добавив функции анализа указанных ключевых полей и дополнительных проверок правильности выполненного задания, а также пополнить набор практических заданий для работы обучающихся в режиме контроля.

Таким образом, задачи решены, цель достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архитектура СУБД. Реляционная модель данных [Электронный ресурс]. — Режим доступа: <https://u.to/ze2iFQ> (дата обращения: 25.04.2019).
2. Базы данных [Электронный ресурс]. — Режим доступа: <http://informatics-lesson.ru/bd/index.php> (дата обращения: 01.06.2019).
3. Базы данных, нормальные формы отношений [Электронный ресурс]. — Режим доступа: [http://isu.nntu.ru/doc/bd\\_1.pdf](http://isu.nntu.ru/doc/bd_1.pdf) (дата обращения: 30.04.2019).
4. Балыкина Е. Н. Сущностные характеристики электронных учебных изданий [Электронный ресурс]. — Режим доступа: <http://goo.gl/BhdF63> (дата обращения: 08.04.2019).
5. Евсеева О. Н. Работа с базами данных на языке C#. Технология ADO.NET [Текст]: учебное пособие / О. Н. Евсеева, А. Б. Шамшев. — Ульяновск: УлГТУ, 2009. — 170 с.
6. Избачков Ю.С. Информационные системы [Текст]: учебное пособие для вузов / Ю. С. Избачков, В. Н. Петров, А. А. Васильев и др. — 3-е издание. — Санкт-Петербург: Питер, 2011. — 544 с.
7. Илюшечкин В. М. Основы использования и проектирования баз данных [Текст]: учебник [Гриф УМО] / В. М. Илюшечкин. — 1-е издание. — Москва: Юрайт, 2014. — 213 с.
8. Коллекция Dictionary [Электронный ресурс]. — Режим доступа: <https://metanit.com/sharp/tutorial/4.9.php> (дата обращения: 20.04.2019).
9. Кузнецов С. Д. Базы данных [Текст]: учебник / С. Д. Кузнецов. — Москва: Academia, 2015. — 496 с.
10. Курс обучения «Основы SQL» [Электронный ресурс]. — Режим доступа: <http://www.intuit.ru/studies/courses/5/5/info> (дата обращения: 21.03.2019).

11. Нагел К. C# 4.0 и платформа .NET 4 для профессионалов [Текст] / К. Нагел, Б. Ивсен. — пер. с англ. Волкова Я., Мухина Н. — Москва: Вильямс, 2015. — 1440 с.
12. Нормализация отношений. Шесть нормальных форм [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/post/254773/> (дата обращения: 28.05.2019).
13. НОУ Интуит — курс «Основы Visual Studio 2015» [Электронный ресурс]. — Режим доступа: <http://www.intuit.ru/studies/courses/78/78> (дата обращения: 28.04.2019).
14. Рабочая программа дисциплины «Базы данных». Для студентов всех форм обучения направления подготовки 44.03.04 Профессиональное обучение (по отраслям) [Текст] / В. В. Вьюхин, С. Ю. Ярина. — Екатеринбург: ФГАОУ ВО «Российский государственный профессионально-педагогический университет», 2017. — 22 с.
15. Регулярные выражения в C# [Электронный ресурс]. — Режим доступа: [http://professorweb.ru/my/csharp/charp\\_theory/level4/4\\_10.php](http://professorweb.ru/my/csharp/charp_theory/level4/4_10.php) (дата обращения: 03.04.2019).
16. Руководство по проектированию реляционных баз данных [Электронный ресурс]. — Режим доступа: <https://habrahabr.ru/post/145381/> (дата обращения: 28.03.2019).
17. Скит Д. C#. Программирование для профессионалов [Текст] / Д. Скит. — пер. с англ. Коваленко В. — Москва: Вильямс, 2011. — 544 с.
18. Технология создания электронных средств обучения [Электронный ресурс]. — Режим доступа: <https://u.to/xWSkFQ> (дата обращения: 01.05.2019).
19. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 [Текст] / Э. Троелсен. — Москва: Вильямс, 2015. — 1312 с.
20. Фримен А. LINQ. Язык интегрированных запросов в C# для профессионалов [Текст] / А. Фримен, Д. Раттц. — пер. с англ. Мухина Н. — Москва: Вильямс, 2016. — 656 с.

21. Хабр [Электронный ресурс]. — Режим доступа: <https://habr.com/> (дата обращения: 30.03.2019).
22. Шилдт Г. С# 4.0. Полное руководство [Текст] / Герберт Шилдт. — пер. с англ. Перестюк Е. П. — Москва: Вильямс, 2011. — 1056 с.
23. Электронный учебник: за и против [Электронный ресурс]. — Режим доступа: <http://festival.1september.ru/articles/505639/> (дата обращения: 09.04.2019).
24. С# для начинающих [Электронный ресурс]. — Режим доступа: <http://www.cyberforum.ru/csharp-beginners> (дата обращения: 03.05.2019).
25. DrExplain [Электронный ресурс]. — Режим доступа: <http://www.drexplain.ru> (дата обращения: 03.06.2019).
26. Fast Colored TextBox for Syntax Highlighting [Электронный ресурс]. — Режим доступа: <https://www.codeproject.com/Articles/161871/Fast-Colored-TextBox-for-syntax-highlighting> (дата обращения: 21.02.2016).
27. Microsoft Developer Network [Электронный ресурс]. — Режим доступа: <https://goo.gl/TIElsg> (дата обращения: 28.03.2019)
28. ProfessorWeb [Электронный ресурс]. — Режим доступа: <http://professorweb.ru/> (дата обращения: 29.03.2019).
29. SQLite [Электронный ресурс]. — Режим доступа: <https://www.sqlite.org/index.html> (дата обращения: 27.05.2019).
30. Stackoverflow [Электронный ресурс]. — Режим доступа: <http://stackoverflow.com/> (дата обращения: 28.03.2019).

# ПРИЛОЖЕНИЕ

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий  
Направление подготовки 44.03.04 Профессиональное обучение (по отраслям)  
Профиль «Информатика и вычислительная техника»  
Профилизация «Компьютерные технологии»

УТВЕРЖДАЮ

Заведующий кафедрой

И. А. Сулова

подпись

и.о. фамилия

« \_\_\_\_ » \_\_\_\_\_ 2019 г.

## ЗАДАНИЕ

### на выполнение выпускной квалификационной работы бакалавра

студента (ки) 4 курса группы КТ-403  
Глушков Олег Александрович  
фамилия, имя, отчество полностью

1. Тема Электронный тренажер «Декомпозиция плоской таблицы»

утверждена распоряжением по институту от « \_\_\_\_ » \_\_\_\_\_ 20 г. № \_\_\_\_

2. Руководитель Ярина Светлана Юрьевна  
фамилия, имя, отчество полностью

ст.преподаватель РГППУ  
ученая степень ученое звание должность место работы

3. Место преддипломной практики Кафедра информационных систем и технологий  
института инженерно-педагогического образования ФГАОУ ВО РГППУ

4. Исходные данные к ВКР Рабочая программа по дисциплине «Базы данных»;  
Беляев М.И. Технология создания электронных средств обучения;

Евсеева О.Н., Шамшиев А.Б. Работа с базами данных на языке С#. Технология ADO.NET;  
Руководство по проектированию реляционных баз данных [Электронный ресурс]. — Ре-  
жим доступа: <https://habrahabr.ru/post/145381/>

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)

