МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ТЮМЕНСКИЙ ГОСУДАРСВЕННЫЙ УНИВЕРСТИТЕТ»**

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Кафедра информационных систем

ОТЧЕТ ОБ ИНДИВИДУАЛЬНОЙ РАБОТЕ

ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

«АВТОМАТИЗАЦИЯ УЧЕТА УСПЕВАЕМОСТИ СТУДЕНТОВ»

Выполнили:

студенты А.В. Зимин,

В.А. Спицын

группы БД 23 05–01

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель:

к. ф.- м. н., доцент

Моор Павел Климентьевич

Тюмень

2023

# 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Описание предметной области

Требуется вести учёт успеваемости студентов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Зачётная книга** | | | | |
| Номер зачетной книги: | | | | 23123332 |
| ФИО студента: | | | | Иванов Иван Иванович |
| Код направления: | | | | 000323417 |
| Направление: | | | | МОиАИС |
|  | | | | |
| **Результаты промежуточной аттестации:** | | | | |
| № | Код Дисциплина | Код  Оценка | Дата | Код  Преподаватель |
| 1 | Алгебра | Отлично | 21.12.2000 | Степанов Степан Степанович |
| 2 | Философия | Зачёт | 22.12.2000 | Петров Пётр Петрович |
| 3 | … | … | … | … |

В зачётной книге указано одно направление. Одно направление указано во многих зачётных книгах.

В одной зачётной книге может содержаться несколько строк. Строка относится к одной зачётной книге.

В каждой строке указана одна дисциплина. Одна дисциплина указана во многих строках.

За дисциплиной закреплён один преподаватель. Один преподаватель закреплён за многими дисциплинами.

В одной строке указана одна оценка. Одна оценка указана во многих строках.

В системе должна быть возможность добавления, изменения, удаления данных о студентах, их оценках, преподавателях и дисциплинах.

В системе должна быть возможность просмотра списка студентов, с отбором по ФИО, и (или) направлению, и (или) оценкам.

В системе должна быть возможность просмотра общей статистики оценок, с отбором по ФИО.

**2. НОРМАЛИЗАЦИЯ**

Представим первичный документ в виде плоской таблицы.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код зач. книги | ФИО студента | Код направления | Направление | № стр | Код дисц. | Дисц. | Код оценки | Оценка | Дата экзамена | Код преподавателя | Преподаватель |
| **KodZ** | Stud | KodSp | Specialty | **NS** | KodD | Discip | KodM | Mark | Date | KodP | Professor |
| 0000001 | Иванов | 0000001 | МОиАИС | 1  2  3 | 00001  00002  00003 | Алгебра  Ин. Яз.  КС | 4  3  1 | Отл.  Хор.  Зач. | 28.12.22  27.12.22  28.12.22 | 000001  000002  000003 | Петров  Кузнецов  Платонов |
| 0000002 | Сидоров | 0000002 | КБ | 1  2  3 | 00001  00002  00004 | Алгебра  Ин. Яз.  Логика | 2  4  1 | Удв.  Отл.  Зач. | 28.12.22  27.12.22  29.12.22 | 000001  000002  000004 | Петров  Кузнецов  Груздев |

## Первая нормальная форма

Ключевые атрибуты: KodZ

Повторяющаяся группа: (NS, KodD, Discip, KodM, Mark, Date, KodP, Professor).

Из каждой повторяющейся группы образуем новый кортеж, для которого дублируются значения атрибутов, не входящих в повторяющуюся группу.

## Вторая нормальная форма

C(KodZ, Stud, KodSp, Specialty, NS, KodD, Discip, KodM, Mark, Date, KodP, Professor)

Потенциальные ключи:

KodZ, NS

KodZ, KodD

Назначим ключ KodZ, NS первичным.

Частичные зависимости:

* KodZ -> Stud, KodSp, Specialty  
  Получаем Z(**KodZ**, Stud, KodSp, Specialty)
* Остаётся C(**KodZ, NS**, KodD, Discip, KodM, Mark, Date, KodP, Professor)

## Третья нормальная форма

Транзитивные зависимости:

* KodZ -> KodSp -> Specialty  
  Получаем S(**KodSp**, Specialty)
* KodZ, NS -> KodD -> Discip, KodP, Professor  
  Получаем D(**KodD,** Discip, KodP, Professor)

В полученном отношении находим транзитивную зависимость от первичного ключа KodD

* KodD -> KodP -> Professor  
  Получаем P(**KodP**, Professor)
* KodZ, NS -> KodM -> Mark  
  Получаем M(**KodM**, Mark)

Остаётся:

* Z(**KodZ**, Stud, KodSp)
* C(**KodZ, NS**, KodD, KodM, Date)

Третья нормальная форма также будет являться нормальной формой Бойса-Кодда потому что каждый детерминант во всех отношениях является потенциальным ключом.

## Итого после нормализации

Отношения:

* Z(**KodZ**, Stud, KodSp)
* C(**KodZ, NS**, KodD, KodM, Date)
* D(**KodD,** Discip, KodP)
* P(**KodP**, Professor)
* S(**KodSp**, Specialty)
* M(**KodM**, Mark)

# 3. ER-МОДЕЛИРОВАНИЕ

В зачётной книге указано одно направление. Одно направление указано во многих зачётных книгах.

Создаём сущности зачётная книга (RecordBook) и направление (Specialty

) и связываем их.

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рисунок 1.1 ER-моделирование.

В одной зачётной книге может содержаться несколько строк. Строка содержится в одной зачётной книге.

Создаём сущность строка (Line) и её связываем с зачётной книгой.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 1.2 ER-моделирование.

В каждой строке указана одна дисциплина. Одна дисциплина указана во многих строках.

Создаём сущность дисциплина (Discipline) и связываем её со строкой.

Изображение выглядит как текст, снимок экрана, диаграмма, число

Автоматически созданное описание

Рисунок 1.3 ER-моделирование.

За дисциплиной закреплён один преподаватель. Один преподаватель закреплён за многими дисциплинами.

Создаём сущность преподаватель (Professor) и связываем его с дисциплиной.

Изображение выглядит как текст, снимок экрана, диаграмма, число

Автоматически созданное описание

Рисунок 1.4 ER-моделирование.

В одной строке указана одна оценка. Одна оценка указана во многих строках. Создаём сущность оценка (Mark) и связываем её со строкой.

Изображение выглядит как текст, диаграмма, Параллельный, снимок экрана

Автоматически созданное описание

Рисунок 1.5 ER-моделирование.

Согласно первичному документу в каждой строке указана дата. Добавляем атрибут дата (date) сущности строка.

Изображение выглядит как текст, диаграмма, снимок экрана, Параллельный

Автоматически созданное описание

Рисунок 1.6 ER-моделирование.

# 4. ОПИСАНИЕ БАЗЫ ДАННЫХ

Для создания БД была использована СУБД SQL Server от Microsoft. Создаём базу данных Study\_results:

CREATE DATABASE Study\_results

Создаём таблицу Professor:

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Атрибут | Тип данных | Примечание |
| **P\_Code** | **Код преподавателя** | **INT** | **PK, IDENTITY** |
| Name | Преподаватель | NVARCHAR(50) | NOT NULL |

Рисунок 2.1 Таблица Professor.

CREATE TABLE dbo.Professor(

P\_Code int IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name nvarchar(50) NOT NULL,

);

INSERT INTO dbo.Professor (Name)

VALUES ('Иванов Иван Иванович'),

('Игорев Игорь Игоревич'),

('Сергеев Сергей Сергеевич')

Создаём таблицу Discipline:

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Атрибут | Тип данных | Примечание |
| **D\_Code** | **Код дисциплины** | **INT** | **PK, IDENTITY** |
| Name | Дисциплина | NVARCHAR(50) | NOT NULL |
| FK\_Professor | Код преподавателя | INT | FK, NOT NULL |

Рисунок 2.2 Таблица Discipline.

CREATE TABLE dbo.Discipline(

D\_Code int IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name nvarchar(50) NOT NULL,

FK\_Professor int NOT NULL,

FOREIGN KEY (FK\_Professor) REFERENCES dbo.Professor (P\_Code)

);

INSERT INTO dbo.Discipline (Name, FK\_Professor)

VALUES ('Алгебра', 1),

('Математический анализ', 2),

('Английский язык', 3)

Создаём таблицу Specialty:

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Атрибут | Тип данных | Примечание |
| **S\_Code** | **Код направления** | **INT** | **PK, IDENTITY** |
| Name | Направление | NVARCHAR(50) | FK, NOT NULL |

Рисунок 2.3 Таблица Specialty.

CREATE TABLE dbo.Specialty(

S\_Code int IDENTITY(1,1) PRIMARY KEY NOT NULL

Name nvarchar(50) NOT NULL

);

INSERT INTO dbo.Specialty(Name)

VALUES ('МОиАИС'),

('КБ'),

('ИСиТ')

Создаём таблицу Mark:

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Атрибут | Тип данных | Примечание |
| **M\_Code** | **Код оценки** | **INT** | **PK, IDENTITY** |
| Name | Оценка | NVARCHAR(20) | NOT NULL |

Рисунок 2.4 Таблица Mark.

CREATE TABLE dbo.Mark(

M\_Code int IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name nvarchar(20) NOT NULL,

);

INSERT INTO dbo.Mark(Name)

VALUES ('Отлично'),

('Хорошо'),

('Удовлетворительно'),

('Зачтено')

Создаём таблицу RecordBook:

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Атрибут | Тип данных | Примечание |
| **RB\_Code** | **Код зачётной книги** | **INT** | **PK, IDENTITY** |
| Name | ФИО студента | NVARCHAR(50) | NOT NULL |
| FK\_Specialty | Код направления | INT | FK, NOT NULL |

Рисунок 2.5 Таблица RecordBook.

CREATE TABLE dbo.RecordBook(

RB\_Code int IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name nvarchar(50) NOT NULL,

FK\_Specialty int NOT NULL,

FOREIGN KEY (FK\_Specialty) REFERENCES dbo.Specialty (S\_Code)

);

INSERT INTO dbo.RecordBook(Name, FK\_Specialty)

VALUES ('Сидоров Иван Иванович', 1),

('Иванов Сергей Сергеевич', 2),

('Михалков Федор Андреевич', 3)

Создаём таблицу Line:

В ER-моделировании было указано, что у таблицы Line составной первичный ключ (код зачётной книги (FK\_RecordBook) + номер строки (Number)), но использование составного первичного ключа неудобно. Поэтому создаём столбец L\_Code, который будет являться первичным ключом для таблицы Line.

|  |  |  |  |
| --- | --- | --- | --- |
| Столбец | Атрибут | Тип данных | Примечание |
| **L\_Code** | **Код строки** | **INT** | **PK, IDENTITY** |
| Number | Номер строки | INT | NOT NULL |
| FK\_RecordBook | Код зачётной книги | INT | FK, NOT NULL |
| FK\_Discipline | Код дисциплины | INT | FK, NOT NULL |
| FK\_Mark | Код оценки | INT | FK, NOT NULL |
| Date | Дата экзамена | DATE | NOT NULL |

Рисунок 2.6 Таблица Line.

CREATE TABLE dbo.Line(

L\_Code int IDENTITY(1,1) PRIMARY KEY NOT NULL,

Number int NOT NULL,

FK\_RecordBook int NOT NULL,

FOREIGN KEY (FK\_RecordBook) REFERENCES dbo.RecordBook (RB\_Code),

FK\_Discipline int NOT NULL,

FOREIGN KEY (FK\_Discipline) REFERENCES dbo.Discipline (D\_Code),

FK\_Mark int NOT NULL,

FOREIGN KEY (FK\_Mark) REFERENCES dbo.Mark (M\_Code),

Date date NOT NULL

);

INSERT INTO dbo.Line(Number, FK\_RecordBook, FK\_Discipline, FK\_Mark, Date)

VALUES (1, 1, 1, 1, '21/07/2022'),

(2, 1, 2, 2, '21/07/2022'),

(3, 1, 3, 1, '22/07/2022'),

(1, 2, 1, 1, '21/07/2022'),

(2, 2, 3, 3, '22/07/2022'),

(3, 2, 2, 2, '23/07/2022')

# 5. ОПИСАНИЕ ПРИЛОЖЕНИЯ

На рисунках 3.1–3.6 отображена главная форма приложения с панелью вкладок, позволяющая работать с зачётными книгами, направлениями подготовки, дисциплинами, преподавателями, видами оценок. Двойным нажатием на ячейку таблицы можно редактировать параметры сущности. На каждой вкладке присутствует возможность фильтрации и сортировки данных.

Изображение выглядит как текст, число, программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 3.1 Главная форма приложения, вкладка зачётных книг.

Изображение выглядит как текст, число, снимок экрана

Автоматически созданное описание

Рисунок 3.2 Главная форма приложения, вкладка направлений.

Изображение выглядит как текст, число, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 3.3 Главная форма приложения, вкладка дисциплин.

Изображение выглядит как текст, число, программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 3.4 Главная форма приложения, вкладка преподавателей.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, снимок экрана

Автоматически созданное описание

Рисунок 3.5 Главная форма приложения, вкладка видов оценок.

В приложении есть возможность просматривать обобщённую информацию об успеваемости студентов. Количество экзаменов, которые сдал студент и средний бал за экзамены; количество дисциплин, за которыми закреплён преподаватель, количество студентов на направлении.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 3.6 Главная форма приложения, вкладка отчётов.

На рисунке 4 отображена форма редактирования зачётной книги, позволяющая работать с зачётными книгами и строками, входящие в зачётные книги. Для редактирования строки требуется дважды кликнуть по соответствующей ячейке - тогда откроется форма выбора.

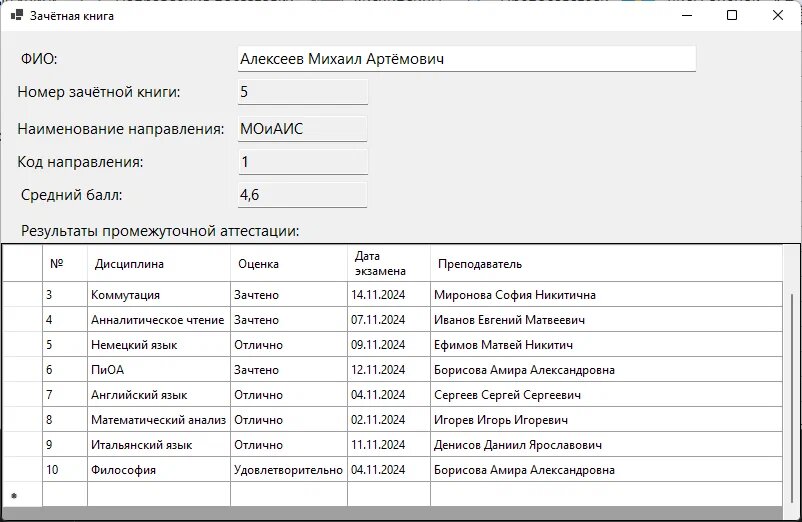


Рисунок 4 Форма зачётной книги.

# Заключение

Мы изучили предметную область, определили первичный документ и провели процесс нормализации до нормальной формы Бойса-Кодда. Выполнили ER-моделирование, разработали физическую модель для СУБД Microsoft SQL Server и реализовали ее. Мы создали приложение на языке C# с использованием Windows Forms для удобного взаимодействия с базой данных, наполнили его тестовыми данными и провели тестирование.

# ЛИТЕРАТУРА

Metanit: cайт о программировании: раздел, посвящённый sql. – URL : <https://metanit.com/sql/sqlserver/> (дата обращения: 26.11.2023).

Microsoft: сайт компании Microsoft: раздел, посвящённый обучению SQL. – URL : [https://learn.microsoft.com/ru-ru/sql/](https://learn.microsoft.com/ru-ru/sql/relational-databases/databases/) (дата обращения: 26.11.2023).

SQL-academy: сайт: онлайн-тренажер с упражнениями по SQL. – URL : <https://sql-academy.org/ru/> (дата обращения: 26.11.2023).

Yandex: сайт компании Yandex: статья о ER-моделировании. – URL : <https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/> (дата обращения: 26.11.2023).