**Министерство транспорта Российской Федерации**

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«РУТ (МИИТ)»**

**Институт транспортной техники и систем управления**

**Кафедра «Управление и защита информации»**

КУРСОВОЙ ПРОЕКТ

по дисциплине

**«Основы построения защищенных баз данных»**

**на тему  
«Школа»**

Выполнил: ст. гр. ТКИ-411  
Митин Г.А.  
Проверил: Васильева М.А.

Москва 2022

Оглавление

[ЦЕЛЬ КУРСОВОГО ПРОЕКТА 3](#_Toc103618427)

[ЗАДАНИЕ НА КУРСОВОВОЙ ПРОЕКТ 3](#_Toc103618428)

[ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ 4](#_Toc103618429)

[1 Инфологическое проектирование 4](#_Toc103618430)

[1.1 Анализ предметной области 4](#_Toc103618431)

[1.2 Анализ информационных задач и круга пользователей системы 5](#_Toc103618432)

[2 Логическое проектирование реляционной БД 6](#_Toc103618433)

[2.1 Составление реляционных отношений 6](#_Toc103618434)

[2.2 Нормализация полученных отношений 11](#_Toc103618435)

[3 Физическое проектирование БД 12](#_Toc103618436)

[3.1 Разработка скриптов на создание базы данных и таблиц 12](#_Toc103618437)

[3.2 Разработка скриптов на добавление данных в таблицы 14](#_Toc103618438)

[3.3 Разработка необходимых запросов, представлений (view), процедур, функций и триггеров 18](#_Toc103618439)

[Список литературы 21](#_Toc103618440)

# ЦЕЛЬ КУРСОВОГО ПРОЕКТА

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) в системе управления базами данных (СУБД) Microsoft SQL Server.

# ЗАДАНИЕ НА КУРСОВОВОЙ ПРОЕКТ

В данном курсовом проекте ставится задача разработать РБД в СУБД Microsoft SQL Server по заданной теме «Школа». Проектирование РБД проводится с помощью метода «Сущность-связь». Проверка построенной модели РБД осуществляется с помощью метода нормализации отношений.

Описание предметной области.

БД создается для информационного обслуживания администрации школы. В школе учатся дети, разбитые на классы. У каждого класса свой классный руководитель. Каждый учитель имеет свое направление, вести он может шесть уроков день максимум, по данному направлению - один учитель.

Готовые запросы:

1. Показывать список учащихся данного класса.

2. Показывать классного руководителя данного класса.

3. Показывать занятость данного учителя.

4. Показывать успеваемость данного ученика.

5. Показывать список учеников, учащихся без троек.

# ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

## Инфологическое проектирование

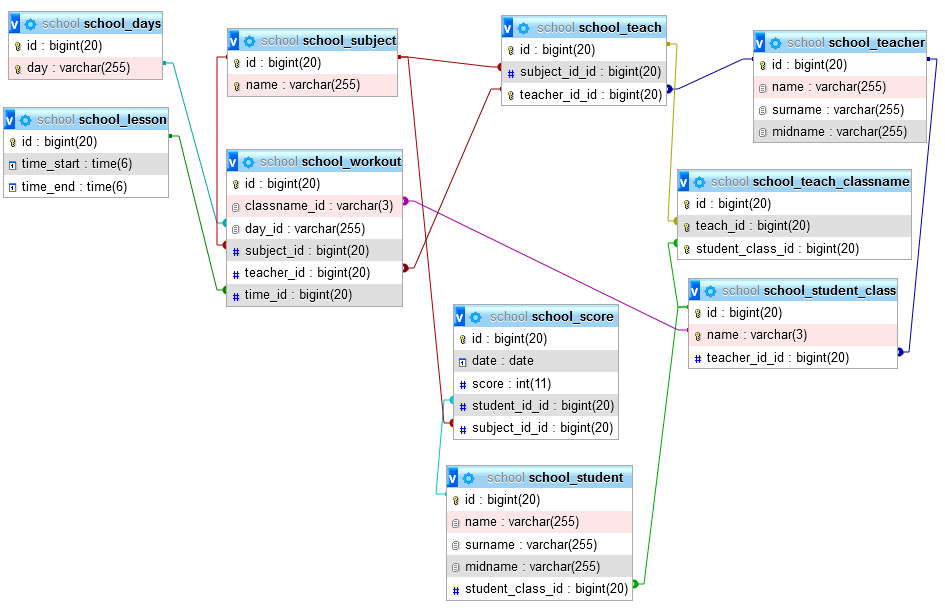
### Анализ предметной области

База данных создаётся для информационного обслуживания администрации школы

Выделим базовые сущности этой предметной области:

* school\_days
* school\_lesson
* school\_subject
* school\_workout
* school\_teach
* school\_score
* school\_student
* school\_teacher
* school\_teach\_classname
* school\_student\_class

ER-диаграмма БД, выполненная в среде визуального построения диаграмм PHPMyAdmin приведена на Рисунке 1. На данной ER-диаграмме присутствуют связи многие - ко - многим.



*Рисунок 1.*

Рисунок 1 – Уточненная ER-диаграмма

В полученной ER-диаграмме присутствуют только связи один-ко-многим. Такие связи легко реализуются средствами СУБД.

### Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

* Преподаватели
* Ученики

Определим границы информационной поддержки пользователей:

**1) Функциональные возможности:**

* ведение БД (запись, чтение, модификация, удаление в архив);
* обеспечение логической непротиворечивости БД;
* обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
* реализация наиболее часто встречающихся запросов в готовом виде.

**2) Готовые запросы:**

* Показывать список учащихся данного класса.
* Показывать классного руководителя данного класса.
* Показывать занятость данного учителя.
* Показывать успеваемость данного ученика.
* Показывать список учеников, учащихся без троек.

## Логическое проектирование реляционной БД

### Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО), и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть). В том случае, если базовое отношение не имеет потенциальных ключей, вводится *суррогатный первичный ключ*, который не несёт смысловой нагрузки и служит только для идентификации записей.

1. Для сущности **Успеваемость** (school\_workout) потенциальным ключом является атрибут «id», т.к. его значение является уникальным. Но значениями этого атрибута будут, числа. Введем для этой сущности суррогатный ключ ID.
2. Для сущности **Направление преподавателя** (school\_teach) атрибут teacher\_id\_id является потенциальным ключом. Для единообразия именования для этой сущности в качестве ключевого атрибута введем суррогатный ключ ID.
3. Для сущности **Ученики** (school\_student) атрибут student\_class\_id является потенциальным ключом. Для единообразия именования для этой сущности в качестве ключевого атрибута введем суррогатный ключ ID.
4. Для сущности **Оценки** (school\_score) совокупность атрибутов subject\_id\_id и student\_id\_id является уникальной и является потенциальным ключом.
5. Для сущности **Учебный день** (school\_days) атрибут id является уникальным и является потенциальным ключом.
6. Для сущности **Учебное время** (school\_lesson) атрибут id является уникальным и является потенциальным ключом.
7. Для сущности **Предмет** (school\_subject) атрибут id является уникальным и является потенциальным ключом.
8. Для сущности **Классный руководитель** (school\_teach\_classname) совокупность атрибутов teach\_id и student\_class\_id является уникальной и является потенциальным ключом.
9. Для сущности **Класс** (school\_student\_class) атрибут teacher\_id\_id является потенциальным ключом. Для единообразия именования для этой сущности в качестве ключевого атрибута введем суррогатный ключ ID.
10. Для сущности **Преподаватель** (school\_teacher) атрибут id является уникальным и является потенциальным ключом.

**Таблица 1 – Схема отношения** **Успеваемость** (school\_workout)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | первичный ключ |
| ID Учебного класса | classname\_id | Строка (3) | уникальное, обязательное поле |
| ID Учебного дня | day\_id | Строка (255) | обязательное поле |
| Дата добавления | add\_date | Дата, время | обязательное поле |
| ID Предмета | subject\_id | Целое число | обязательное поле |
| ID Преподавателя | teacher\_id | Целое число | обязательное поле |
| ID Расписания | time\_id | Целое число | обязательное поле |

Таблица 2 – Схема отношения Направление Преподавателя (school\_teach)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | id | Целое число | первичный ключ |
| ID Предмета | subject\_id\_id | Целое число | обязательное поле |
| Вторичный ключ | teacher\_id\_id | Целое число | обязательное поле |

Таблица 3 – Схема отношения Ученики (school\_student)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Ученика | id | Целое число | первичный ключ |
| Имя | name | Строка(255) | обязательное поле |
| Фамилия | surname | Строка(255) | обязательное поле |
| Отчество | midlname | Строка(255) | обязательное поле |
| ID класса ученика | student\_class\_id | Целое число | обязательное поле |

Таблица 4 – Схема отношения Связь Оценки (school\_score)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Оценки | id | Целое число | первичный ключ |
| Дата добавления | date | Дата, время | обязательное поле |
| Стоимость | score | Целое число | обязательное поле |
| ID Ученика | student\_id\_id | Целое число | обязательное поле |
| ID Предмета | subject\_id\_id | Целое число | обязательное поле |

Таблица 5 – Схема отношения Связь Учебные дни (school\_days)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Учебного дня | id | Целое число | первичный ключ |
| День | day | Строка(255) | обязательное поле |

Таблица 6 – Схема отношения Расписание (school\_lesson)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Расписания | id | Целое число | первичный ключ |
| Время начала урока | time\_start | Время | обязательное поле |
| Время конца урока | time\_end | Время | обязательное поле |

Таблица 7 – Схема отношения Предмет (school\_subject)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Предмета | id | Целое число | первичный ключ |
| Название предмета | name | Строка(255) | обязательное поле |

Таблица 8 – Схема отношения Классный руководитель (school\_teach\_classname)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Классного руководителя | id | Целое число | первичный ключ |
| ID Преподавателя | teach\_id | Целое число | обязательное поле |
| ID Класса | student\_class\_id | Целое число | обязательное поле |

Таблица 9 – Схема отношения Класс (school\_student\_class)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Класса | id | Целое число | первичный ключ |
| Название класса | name | Строка(3) | обязательное поле |
| ID Преподавателя | teacher\_id\_id | Целое число | обязательное поле |

Таблица 10 – Схема отношения Преподаватель (school\_teacher)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| ID Учителя | id | Целое число | первичный ключ |
| Имя | name | Строка(255) | обязательное поле |
| Фамилия | surname | Строка(255) | обязательное поле |
| Отчество | midlname | Строка(255) | обязательное поле |

### Нормализация полученных отношений

**1НФ.** Для приведения таблиц к 1НФ требуется составить таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.  
В наших таблицах все атрибуты простые.

**2НФ.** Все отношения находятся во 2НФ, так как для каждого из них введен уникальный ключ.

**3НФ**. Все неключевые атрибуты зависят только от первичного ключа и не зависят друг от друга. Поэтому спроектированная **база данных находится в третьей нормальной форме**.

## Физическое проектирование БД

### Разработка скриптов на создание базы данных и таблиц

БД «Школа» проектируется в среде Microsoft SQL Server Management Studio.

*Создание базы данных «*Школа*»:*

USE school;

*Создание таблицы Учебные дни*

CREATE TABLE [Days] (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

[day] varchar(255) NOT NULL UNIQUE ([day])

);

*Создание таблицы Расписание*

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

time\_start time(6) NOT NULL,

time\_end time(6) NOT NULL

);

*Создание таблицы связь Оценки*

CREATE TABLE Score (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

[date] date NOT NULL,

score int NOT NULL CHECK (score>0 and score<6),

student\_id bigint NOT NULL,

subject\_id bigint NOT NULL

);  
 *Создание таблицы связь Ученики*

CREATE TABLE Student (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

[name] varchar(255) NOT NULL,

surname varchar(255) NOT NULL,

midname varchar(255) NOT NULL,

student\_class\_id bigint NOT NULL

);

*Создание таблицы связь Класс*

CREATE TABLE Student\_class (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

[name] varchar(3) NOT NULL UNIQUE ([name]),

teacher\_id bigint NOT NULL

);

*Создание таблицы связь Предметы*

CREATE TABLE [Subject] (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

[name] varchar(255) NOT NULL

);

*Создание таблицы связь Направление учителя*

CREATE TABLE Teach (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

subject\_id bigint NOT NULL,

teacher\_id bigint NOT NULL UNIQUE (teacher\_id)

);

*Создание таблицы связь Преподаватель*

CREATE TABLE Teacher (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

[name] varchar(255) NOT NULL,

surname varchar(255) NOT NULL,

midname varchar(255) NOT NULL

);

*Создание таблицы связь Классный руководитель*

CREATE TABLE Teach\_classname (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

teach\_id bigint NOT NULL,

student\_class\_id bigint NOT NULL

);

*Создание таблицы связь Успеваемость*

CREATE TABLE Workout (

id bigint NOT NULL PRIMARY KEY IDENTITY(1,1),

classname\_id varchar(3) NOT NULL,

day\_id varchar(255) NOT NULL,

subject\_id bigint NOT NULL,

teacher\_id bigint NOT NULL,

time\_id bigint NOT NULL

);

### Разработка скриптов на добавление данных в таблицы

*Заполнение таблицы* *Учебные дни:*

INSERT INTO [Days] ([day]) VALUES

('Вторник'),

('Понедельник'),

('Пятница'),

('Среда'),

('Четверг');

*Заполнение таблицы* *Преподаватель:*

INSERT INTO [Teacher] ([name], [surname], [midname]) VALUES

('Светлана', 'Лоцманова', 'Николаевна'),

('Любовь', 'Ноздрина', 'Дмитриевна'),

('Ольга', 'Михалева', 'Федоровна'),

('Ирина', 'Кириллова', 'Алексеевна'),

('Ольга', 'Косьяненко', 'Николаевна'),

('Нина', 'Касьянова', 'Валентиновна'),

('Наталья', 'Лебедева', 'Борисовна'),

('Виктория', 'Алексеева', 'Олеговна'),

('Дарья', 'Савельева', 'Александровна'),

('Сергей', 'Зябрев', 'Иванович'),

('Светлана', 'Сидичкина', 'Валерьевна'),

('Александра', 'Зотова', 'Александровна'),

('Лариса', 'Якименко', 'Николаевна'),

('Елена', 'Пудова', 'Владимировна');

*Заполнение таблицы* *Расписание:*

INSERT INTO Lesson ([time\_start], [time\_end]) VALUES

('08:30:00.000000', '09:15:00.000000'),

('09:25:00.000000', '10:10:00.000000'),

('10:30:00.000000', '11:15:00.000000'),

('11:30:00.000000', '12:15:00.000000'),

('12:25:00.000000', '13:10:00.000000'),

('13:20:00.000000', '14:05:00.000000'),

('14:10:00.000000', '14:55:00.000000');

*Заполнение таблицы* *Названия класса:*

INSERT INTO [Student\_class] ([name], [teacher\_id]) VALUES

('10А', 1),

('10Б', 2),

('11А', 3),

('11Б', 4);

*Заполнение таблицы* *Связь Предметы:*

INSERT INTO [Subject] ([name]) VALUES

('Русский язык'),

('Алгебра'),

('Биология'),

('История'),

('Химия'),

('География'),

('Английский язык'),

('Физика'),

('Материальные технологии'),

('Физическая культура'),

('Информатика и ИКТ'),

('Французский язык'),

('Ритмика'),

('МХК');

*Заполнение таблицы* *Связь Меню Ученики:*

INSERT INTO [Student] ([name], [surname], [midname], [student\_class\_id]) VALUES

('Денис', 'Антонов', 'Николаевич', 1),

('Ольга', 'Арсентьева', 'Святославовна', 1),

('Анна', 'Афанасьева', 'Леонидовна', 1),

('Алексей', 'Васильев', 'Геннадьевич', 1),

('Ольга', 'Васильева', 'Витальевна', 1),

('Вера', 'Владимирова', 'Владимировна', 1),

('Галина', 'Гурьева', 'Игоревна', 1),

('Алексей', 'Егоров', 'Валерьевич', 1),

('Андрей', 'Егоров', 'Витальевич', 1),

('Антон', 'Ермаков', 'Эдуардович', 1),

('Елена', 'Александрова', 'Александровна', 2),

('Марина', 'Алексеева', 'Юрьевна', 2),

('Станислав', 'Васильев', 'Фёдорович', 2),

('Валерий', 'Викторов', 'Михайлович', 2),

('Наталия', 'Гордеева', 'Михайловна', 2),

('Алиса', 'Григорьева', 'Игоревна', 2),

('Надежда', 'Григорьева', 'Александровна', 2),

('Анатолий', 'Егоров', 'Валерьевич', 2),

('Анастасия', 'Егорова', 'Владимировна', 2),

('Ольга', 'Егорова', 'Николаевна', 2),

('Инесса', 'Васильева', 'Николаевна', 3),

('Алексей', 'Дементьев', 'Анатольевич', 3),

('Виктор', 'Иванов', 'Вячеславович', 3),

('Константин', 'Иванов', 'Георгиевич', 3),

('Дмитрий', 'Ильин', 'Юрьевич', 3),

('Диана', 'Ильина', 'Ильинична', 3),

('Екатерина', 'Ильина', 'Александровна', 3),

('Снежана', 'Маркова', 'Анатольевна', 3),

('Николай', 'Михайлов', 'Валерьянович', 3),

('Екатерина', 'Михайлова', 'Анатольевна', 3),

('Людмила', 'Андреева', 'Алексеевна', 4),

('Алена', 'Андреева', 'Витальевна', 4),

('Анастасия', 'Василькова', 'Валентиновна', 4),

('Владимир', 'Викторов', 'Михайлович', 4),

('Алексей', 'Григорьев', 'Львович', 4),

('Анна', 'Ефимова', 'Васильевна', 4),

('Николай', 'Илларионов', 'Александрович', 4),

('Сергей', 'Козлов', 'Леонидович', 4),

('Алина', 'Маркова', 'Юрьевна', 4),

('Денис', 'Кузнецов', 'Леонидович', 4);

*Заполнение таблицы* *Направление учителя:*

INSERT INTO [Teach] ([subject\_id], [teacher\_id]) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10),

(11, 11),

(12, 12),

(13, 13),

*Заполнение таблицы* *Классный руководитель:*

INSERT INTO [Teach\_classname] ([teach\_id], [student\_class\_id]) VALUES

(14, 1),

(14, 2),

(14, 3),

(14, 4),

(13, 1),

(13, 2),

(12, 1),

(12, 3),

(11, 3),

(11, 4),

(10, 1),

(10, 2),

(9, 2),

(9, 3),

(9, 4),

(8, 1),

(8, 2),

(8, 3),

(8, 4),

(7, 2),

(7, 4),

(6, 1),

(6, 2),

(5, 3),

(5, 4),

(4, 2),

(3, 1),

(3, 2),

(3, 3),

(3, 4),

(2, 1),

(1, 1),

(1, 3);

*Заполнение таблицы* *Успеваемость:*

INSERT INTO [Workout] ([classname\_id], [day\_id], [subject\_id], [teacher\_id], [time\_id]) VALUES

('10А', 'Понедельник', 1, 1, 1),

('10Б', 'Понедельник', 6, 6, 1),

('11А', 'Понедельник', 8, 8, 1),

('11Б', 'Понедельник', 2, 2, 1),

('10А', 'Понедельник', 6, 6, 2),

('10Б', 'Понедельник', 1, 1, 2),

('11А', 'Понедельник', 2, 2, 2),

('11Б', 'Понедельник', 9, 9, 2),

('10А', 'Понедельник', 2, 2, 3),

('10Б', 'Понедельник', 5, 5, 3),

('11А', 'Понедельник', 8, 8, 3),

('11Б', 'Понедельник', 7, 7, 3),

('10А', 'Понедельник', 4, 4, 4),

('10Б', 'Понедельник', 2, 2, 4),

('11А', 'Понедельник', 3, 3, 4),

('11Б', 'Понедельник', 8, 8, 4),

('10А', 'Понедельник', 8, 8, 5),

('10Б', 'Понедельник', 4, 4, 5),

('11А', 'Понедельник', 11, 11, 5),

('11Б', 'Понедельник', 12, 12, 5),

('10А', 'Вторник', 14, 14, 1),

('10Б', 'Вторник', 5, 5, 1),

('11А', 'Вторник', 9, 9, 1),

('11Б', 'Вторник', 2, 2, 1),

('10А', 'Вторник', 5, 5, 2),

('10Б', 'Вторник', 14, 14, 2),

('11А', 'Вторник', 2, 2, 2),

('11Б', 'Вторник', 12, 12, 2),

('10А', 'Вторник', 11, 11, 3),

('10Б', 'Вторник', 1, 1, 3),

('11А', 'Вторник', 2, 2, 3),

('11Б', 'Вторник', 5, 5, 3),

('10А', 'Вторник', 1, 1, 4),

('10Б', 'Вторник', 2, 2, 4),

('11А', 'Вторник', 5, 5, 4),

('11Б', 'Вторник', 11, 11, 4),

('10А', 'Вторник', 2, 2, 5),

('10Б', 'Вторник', 10, 10, 5),

('11А', 'Вторник', 1, 1, 5),

('11Б', 'Вторник', 6, 6, 5),

('10А', 'Вторник', 10, 10, 6),

('10Б', 'Вторник', 7, 7, 6),

('11А', 'Вторник', 5, 5, 6),

('11Б', 'Вторник', 13, 13, 6);

*Заполнение таблицы* *Оценка:*

INSERT INTO [Score] ([date], [score], [student\_id], [subject\_id]) VALUES

('2022-05-11', 5, 1, 4),

('2022-05-11', 4, 2, 4),

('2022-05-11', 5, 3, 4),

('2022-05-11', 5, 4, 4),

('2022-05-11', 3, 8, 4),

('2022-05-11', 4, 9, 4),

('2022-05-11', 2, 10, 4),

('2022-05-11', 3, 1, 1),

('2022-05-11', 5, 2, 1),

('2022-05-11', 4, 3, 1),

('2022-05-11', 3, 4, 1),

('2022-05-11', 3, 6, 1),

('2022-05-11', 3, 8, 1),

('2022-05-11', 5, 9, 1),

('2022-05-11', 2, 10, 1),

('2022-05-11', 3, 11, 2),

('2022-05-11', 5, 12, 2),

('2022-05-11', 3, 13, 2),

('2022-05-11', 3, 14, 2),

('2022-05-11', 5, 15, 2),

('2022-05-11', 3, 16, 2),

('2022-05-11', 4, 17, 2),

('2022-05-11', 5, 18, 2);

### (14, 14);Разработка необходимых запросов, представлений (view), процедур, функций и триггеров

*Показывать список учащихся данного класса.:*

SELECT B.surname, B.[name], B.midname, A.[name] as [classname]

FROM Student as B

INNER JOIN Student\_class as A

ON student\_class\_id = A.id

WHERE student\_class\_id = 1;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.

*Показывать классного руководителя данного класса.:*

SELECT B.surname, B.[name], B.midname

FROM Student\_class as A

INNER JOIN Teacher as B

ON A.teacher\_id = B.id

WHERE A.[name] = '10А';



Рисунок 3.

*Показывать занятость данного учителя:*

SELECT T.surname, T.[name], T.midname, classname\_id, day\_id, S.[name], Convert(Time(0),L.time\_start,0) as time\_start, Convert(Time(0),L.time\_end,0) as time\_end

FROM Workout as A

INNER JOIN Teacher as T ON A.teacher\_id = T.id

INNER JOIN [Subject] as S ON A.subject\_id = S.id

INNER JOIN Lesson as L ON A.time\_id = L.id

WHERE A.teacher\_id = 1;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4.

*Показывать успеваемость данного ученика:*

SELECT B.[name], score, [date]

FROM SCORE as A

INNER JOIN [Subject] as B ON A.subject\_id = B.id

WHERE student\_id = (

SELECT id

FROM Student

WHERE [name] = 'Денис'

AND surname = 'Антонов')

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5.

*Показывать список учеников, учащихся без троек:*

SELECT DISTINCT B.[name], B.[surname]

FROM Score as A

INNER JOIN Student as B ON A.subject\_id = B.id

WHERE A.score > 3;

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6

Построим диаграмму спроектированной базы данных (Рисунок 7).

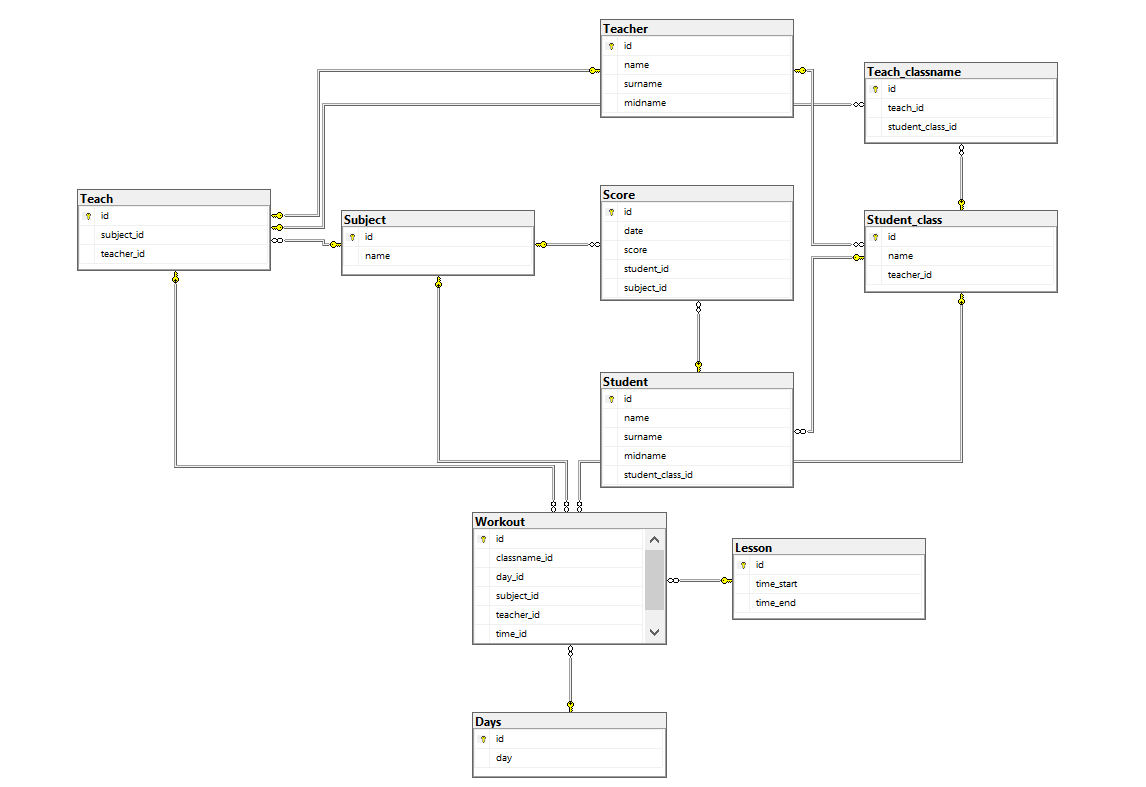


Рисунок 7

# Список литературы

**Visual Paradigm** Entity Relation [Online] // Visual Paradigm Online. - @2020 by Visual Paradigm, 2020. - 2 2, 2020. - https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=ERDiagram.

ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления [В Интернете] // Официальный сайт компании «КонсультантПлюс». - 24 Март 2020 г.. - http://www.consultant.ru/document/cons\_doc\_LAW\_292293/.