Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**КУРСОВОЙ ПРОЕКТ**

По дисциплине: «Информационное обеспечение систем управления»

на тему:

«Создание базы данных»

Выполнил:

ст. гр. ТУУ-411 Прокопьева Е. Е.

Проверил:

доц. Васильева М.А.

МОСКВА 2021

Содержание

[ЦЕЛЬ КУРСОВОГО ПРОЕКТА 2](#_Toc90502932)

[ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ 2](#_Toc90502933)

[ВВЕДЕНИЕ 2](#_Toc90502934)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc90502935)

[ВЫВОД 16](#_Toc90502936)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc90502937)

# ЦЕЛЬ КУРСОВОГО ПРОЕКТА

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

# ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

В данном курсовом проекте ставится задача разработать РБД в СУБД Microsoft SQL Server по заданной теме. Проектирование РБД проводится с помощью метода «Сущность – связь». Проверка построенной модели РБД осуществляется с помощи метода нормализации отношений. Пояснительная записка должна содержать пункты по проектированию РБД и пункты по разработке РБД в СУБД Microsoft SQL Server: разработка скриптов на создание и заполнение РБД, разработка необходимых функций, процедур, триггеров и представлений (views).

Заданная тема (формализующая заданную предметная область): «Школа»

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 в 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.

# ВВЕДЕНИЕ

Реляционная модель данных основывается на математических принципах, которые вытекают из теории множеств и математической логики. Эти принципы впервые были применены в области моделирования данных в конце 1960-х годов доктором Е.Ф. Коддом, а впервые опубликованы – в 1970 году. Тогда же и появились первые прототипы реляционных систем управления базами данных (СУБД).

Кодд предложил применение реляционной алгебры к системе управления реляционной базой данных (СУРБД), для разбиения данных в связанные наборы. Он организовал свою СУБД вокруг теории, основанной на наборах данных.

Самая распространенная трактовка реляционной модели данных принадлежит известному последователю идей Кодда Кристоферу Дейту. Согласно трактовке Дейта, реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной части, манипуляционной части и целостной части.

В структурной части модели фиксируется, что единственной родовой структурой данных, используемой в РБД, является нормализованное n-арное отношение (кортеж). Определяются понятия доменов, атрибутов, кортежей, заголовка, тела и переменной отношения.

В манипуляционной части модели определяются два фундаментальных механизма манипулирования РБД – реляционная алгебра и реляционное исчисление. Первый механизм базируется на классической теории множеств, а второй – на классическом логическом аппарате исчисления предикатов первого порядка. Основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка РБД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Наконец, в целостной части реляционной модели данных фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД. Первое требование называется требованием целостности сущности (entity integrity). Объекту или сущности реального мира в РБД соответствуют кортеже отношений. Конкретно требование состоит в том, что любой кортеж любого значения-отношения любой переменной отношения должен быть отличим от любого другого кортежа этого значения отношения по составным значениям заранее определенного множества атрибутов переменной отношения, т.е. ***любая переменная отношения должна обладать первичным ключом***. Это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

Модель данных, или концептуальное описание ПрО – начальный уровень проектирования баз данных.

С точки зрения теории РБД, основные принципы реляционной модели на концептуальном уровне можно сформулировать следующим образом:

* все данные представляются в виде структуры, определенной в виде строк и столбцов и называемой отношением;
* все значения являются скалярами. Это означает, что для любой строки и столбца любого отношения существует одно и только одно значение;
* все операции выполняются над целым отношением, и результатом их выполнения также является целое отношение. Этот принцип называется замыканием.

Формулируя принципы реляционной модели, доктор Кодд выбрал термин «отношение» (relation), т.к. он считал, что этот термин однозначен (в то время как, например, термин «таблица» имеет множество различных видов – таблица в тексте, электронная таблица и т.д.).

Каждая строка, содержащая данные, называется ***кортежем***, каждый столбец отношения называется ***атрибутом*** (на уровне практической работы с современными РБД используются термины «запись» и «поле»).

Элементами описания реляционной модели данных на концептуальном уровне являются сущности, атрибуты, домены и связи.

***Сущность*** – некоторый обособленный объект или событие, информацию о котором необходимо сохранять в БД; сущность имеет определенный набор свойств – атрибутов. Сущности могут быть как физические, так и абстрактные. Для сущностей различают ее тип и экземпляр. ***Тип*** характеризуется именем и списком свойств, а ***экземпляр*** – конкретными значениями свойств.

Домен – это набор всех допустимых значений, которые может содержать атрибут. Понятие «домен» часто путают с понятием «тип данных». Необходимо различать эти два понятия. Тип данных – это физическая концепция, а домен – логическая. Например, «целое число» - это тип данных, а «возраст» - это домен.

Существует несколько типов связей между двумя сущностями: это связи «один – к – одному», «один – ко – многим» и «многие – ко – многим».

Каждая связь в реляционной модели характеризуется именем, обязательностью, типов и степенью. Различают *факультативные* и *обязательные* связи. Если сущность одного типа оказывается по необходимости связанной с сущностью другого типа, то между этими типами объектов существует *обязательная* связь. Иначе связь является *факультативной*.

Степень связи определяется количеством сущностей, которые охвачены данной связью.

*Диаграмма «сущности-связи»* (Entity-Relationship diagrams, или ER-diagram) служит для описания схемы базы на концептуальном уровне проектирования. Метод был предложен в 1976 г. Питером Пин Шань Ченом (Peter Pin Shan Chen).

# ОСНОВНАЯ ЧАСТЬ

***Базовые сущности ПрО***

Люди – полное имя и дата рождения;

Роли – название роли (учитель или ученик);

Ученики – имеют имя и дату рождения (через сущность «Люди») и класс, в котором обучаются;

Учителя – имеют имя и дату рождения (через сущность «Люди») и свой предмет, которые ведут только они;

Класс – учебный класс, который имеет название и своего классного руководителя;

Предметы – название предметов;

Успеваемость – у каждого ученика по конкретному предмету своя годовая оценка;

Расписание – содержит день недели, номер урока, класс и предметы.

***ER-диаграмма будущей базы данных***

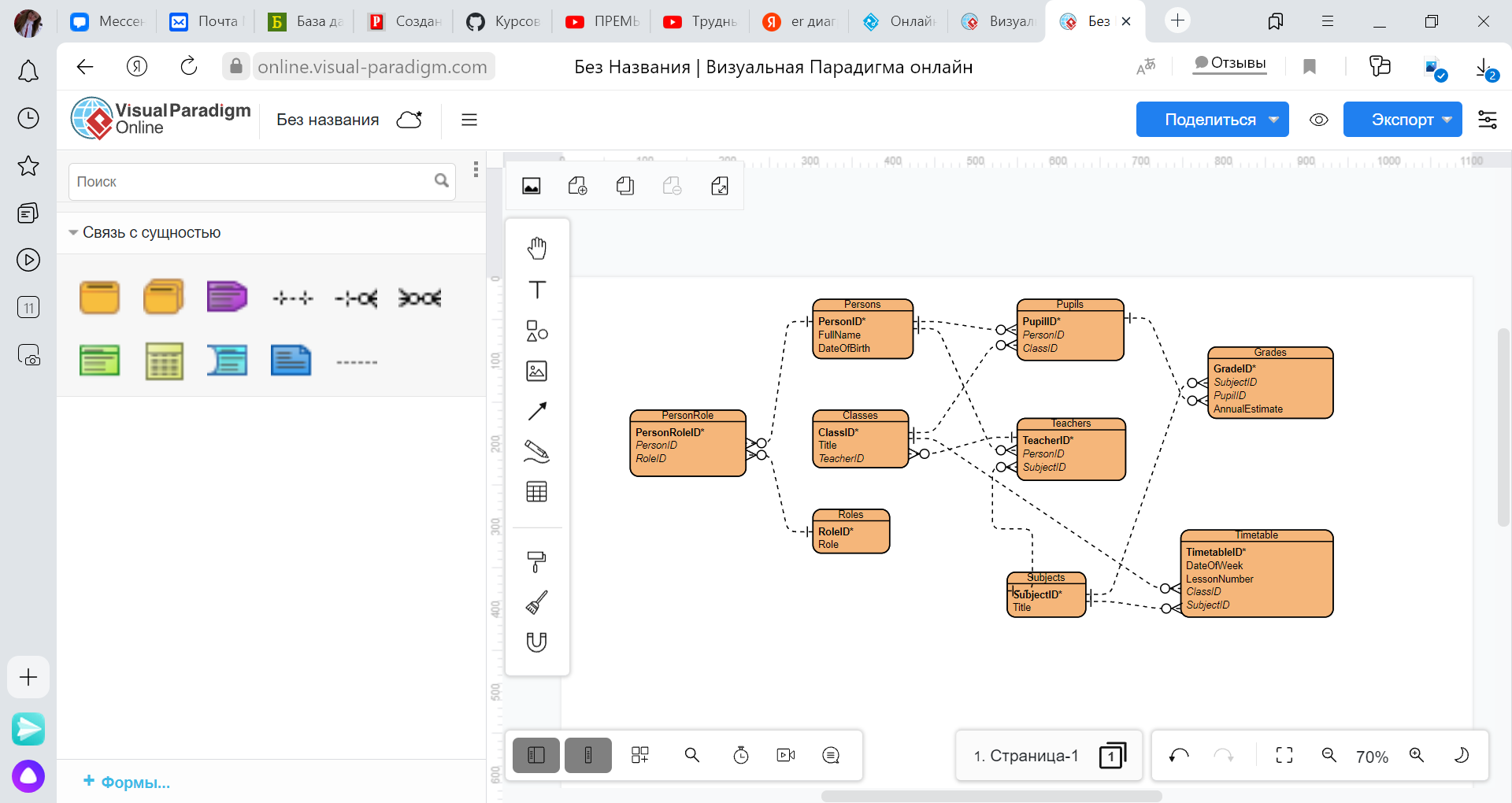


Рисунок 1 – ER-диаграмма будущей БД

***Составление реляционных отношений***

Таблица 1 – Отношение Люди (Persons)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | PersonID | INT | Первичный ключ |
| ФИО | FullName | VARCHAR(255) | Обязательное поле |
| Дата рождения | DateOfBirth | DATA | Обязательное поле |

Таблица 2 – Отношение Роли (Roles)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | RoleID | INT | Первичный ключ |
| Название роли | Role | VARCHAR(50) | Обязательное поле |

Таблица 3 – Отношение Люди и Роли (PersonRole)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | PersonRoleID | INT | Первичный ключ |
| Люди | PersonID | INT | Внешний ключ (Persons)  Обязательное поле |
| Роли | RoleID | INT | Внешний ключ (Roles)  Обязательное поле |

Таблица 4 – Отношение Предметы (Subjects)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | SubjectID | INT | Первичный ключ |
| Название предмета | Title | VARCHAR(255) | Обязательное поле |

Таблица 5 – Отношение Учителя (Teachers)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | TeacherID | INT | Первичный ключ |
| Люди | PersonID | INT | Внешний ключ (Persons)  Обязательное поле |
| Предмет | SubjectID | INT | Внешний ключ (Subjects)  Обязательное поле |

Таблица 6 – Отношение Классы (Classes)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | ClassID | INT | Первичный ключ |
| Название класса | Title | VARCHAR(10) | Обязательное поле |
| Классный руководитель | TeacherID | INT | Внешний ключ (Teachers)  Обязательное поле |

Таблица 7 – Отношение Ученики (Pupils)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | PupilID | INT | Первичный ключ |
| Люди | PersonID | INT | Внешний ключ  (Persons)  Обязательное поле |
| Класс | ClassID | INT | Внешний ключ (Classes)  Обязательное поле |

Таблица 8 – Отношение Успеваемость (Grades)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | GradeID | INT | Первичный ключ |
| Предмет | SubjectID | INT | Внешний ключ  (Subjects)  Обязательное поле |
| Ученик | PupilID | INT | Внешний ключ (Pupils)  Обязательное поле |
| Годовая оценка | AnnualEstimate | INT | Обязательное поле |

Таблица 9 – Отношение Расписание (Timetable)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Примечание** |
| Первичный ключ | Timetable ID | INT | Первичный ключ |
| День недели | DateOfWeek | VARCHAR(50) | Обязательное поле |
| Номер урока | LessonNumber | INT | Обязательное поле |
| Класс | ClassID | INT | Внешний ключ (Classes)  Обязательное поле |
| Предмет | SubjectID | INT | Внешний ключ (Subjects)  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.

Все отношения находятся в 1НФ.

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

***Физическое проектирование БД***

Создание базы данных School:

CREATE DATABASE School;

Создание таблиц

Создание таблицы Persons:

CREATE TABLE Persons(

PersonID INT PRIMARY KEY IDENTITY(1,1)

,FullName VARCHAR(255) NOT NULL

,DateOfBirth Date NOT NULL

);

Создание таблицы Roles:

CREATE TABLE Roles(

RoleID INT PRIMARY KEY IDENTITY(1,1)

,[Role] VARCHAR(50) NOT NULL

);

Создание таблицы PersonRoles:

CREATE TABLE PersonRole(

PersonRoleID INT PRIMARY KEY IDENTITY(1,1)

,PersonID INT NOT NULL

,FOREIGN KEY (PersonID) REFERENCES Persons (PersonID)

,RoleID INT NOT NULL

,FOREIGN KEY (RoleID) REFERENCES Roles (RoleID)

);

Создание таблицы Subjects:

CREATE TABLE Subjects(

SubjectID INT PRIMARY KEY IDENTITY(1,1)

,Title VARCHAR(255) NOT NULL

);

Создание таблицы Teachers:

CREATE TABLE Teachers(

TeacherID INT PRIMARY KEY IDENTITY(1,1)

,PersonID INT NOT NULL

,FOREIGN KEY (PersonID) REFERENCES Persons (PersonID)

,SubjectID INT NOT NULL

,FOREIGN KEY (SubjectID) REFERENCES Subjects (SubjectID)

);

Создание таблицы Classes:

CREATE TABLE Classes(

ClassID INT PRIMARY KEY IDENTITY(1,1)

,Title VARCHAR(10) NOT NULL

,TeacherID INT NOT NULL

,FOREIGN KEY (TeacherID) REFERENCES Teachers (TeacherID)

);

Создание таблицы Pupils:

CREATE TABLE Pupils(

PupilID INT PRIMARY KEY IDENTITY(1,1)

,PersonID INT NOT NULL

,FOREIGN KEY (PersonID) REFERENCES Persons (PersonID)

,ClassID INT NOT NULL

,FOREIGN KEY (ClassID) REFERENCES Classes (ClassID)

);

Создание таблицы Grades:

CREATE TABLE Grades(

GradeID INT PRIMARY KEY IDENTITY(1,1)

,SubjectID INT NOT NULL

,FOREIGN KEY (SubjectID) REFERENCES Subjects (SubjectID)

,PupilID INT NOT NULL

,FOREIGN KEY (PupilID) REFERENCES Pupils (PupilID)

,AnnualEstimate INT NOT NULL

);

Создание таблицы Timetable:

CREATE TABLE Timetable(

TimetableID INT PRIMARY KEY IDENTITY(1,1)

,DateOfWeek VARCHAR(50) NOT NULL

,LessonNumber INT NOT NULL

,ClassID INT NOT NULL

,FOREIGN KEY (ClassID) REFERENCES Classes (ClassID)

,SubjectID INT NOT NULL

,FOREIGN KEY (SubjectID) REFERENCES Subjects (SubjectID)

);

Добавление данных в таблицы

INSERT INTO Persons (FullName,DateOfBirth) VALUES

('Смирнова Татьяна Александровна','2006-01-13')--1 9а

,('Романова Екатерина Матвеевна','2006-02-10')--2 9а

,('Власова Антонина Артемьевна','2006-01-08')--3 9а

,('Иванова Алиса Давидовна','2006-08-11')--4 9а

,('Воробьев Артём Святославович','2006-06-16')--5 9а

,('Самойлов Степан Михайлович','2006-11-03')--6 9а

,('Шестакова Ясмина Мирославовна','2006-03-18')--7 9б

,('Михайлова Виктория Матвеевна','2006-12-01')--8 9б

,('Смирнова Варвара Георгиевна','2006-02-19')--9 9б

,('Миронов Ярослав Даниилович','2006-02-06')--10 9б

,('Калычев Илья Николаевич','2005-10-10')--11 9б

,('Прокофьева Дарина Ильинична','2006-09-06')--12 10

,('Софронова Милана Елисеевна','2005-05-07')--13 10

,('Лобанов Тимофей Глебович','2006-05-05')--14 10

,('Колпаков Александр Дмитриевич','2006-04-15')--15 10

,('Никулин Демьян Михайлович','2006-10-21')--16 11

,('Муратова Екатерина Артёмовна','2006-06-24')--17 11

,('Иванова Василиса Леонидовна','2006-08-27')--18 11

,('Горшкова Виктория Ивановна','2006-07-20')--19 11

,('Зайцева Валентина Федоровна','1988-09-23')--20 русский кл.рук.9а

,('Соболева Лидия Егоровна','1979-11-06')--21 математика кл.рук.9б

,('Соколова Алиса Данииловна','1963-03-20')--22 физика кл.рук.10

,('Осипов Василий Куприянович','1986-05-26')--23 физкультура

,('Гусев Сергей Макарович','1987-06-19');--24 биология кл.рук.11

INSERT INTO Roles([Role]) VALUES

('ученик'),('учитель');

INSERT INTO PersonRole (PersonID,RoleID) VALUES

(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1),(8,1),(9,1),(10,1) ,(11,1),(12,1),(13,1),(14,1),(15,1),(16,1),(17,1),(18,1),(19,1),(20,2)

,(21,2),(22,2),(23,2),(24,2);

INSERT INTO Subjects (Title) VALUES

('Русский язык')--1

,('Математика')--2

,('Физика')--3

,('Физкультура')--4

,('Биология');--5

INSERT INTO Teachers (PersonID, SubjectID) VALUES

(20,1),(21,2),(22,3),(23,4),(24,5);

INSERT INTO Classes (Title,TeacherID) VALUES

('9А',1),('9Б',2),('10',3),('11',5);

INSERT INTO Pupils (PersonID,ClassID) VALUES

(1,1),(2,1),(3,1),(4,1),(5,1),(6,1)

,(7,2),(8,2),(9,2),(10,2),(11,2)

,(12,3),(13,3),(14,3),(15,3)

,(16,4),(17,4),(18,4),(19,4);

INSERT INTO Grades (SubjectID,PupilID,AnnualEstimate) VALUES

--1

(1,1,4),(2,1,4),(3,1,5),(4,1,3),(5,1,4),

--2

(1,2,4),(2,2,4),(3,2,5),(4,2,3),(5,2,4),

--3

(1,3,4),(2,3,4),(3,3,5),(4,3,3),(5,3,4),

--4

(1,1,4),(2,1,4),(3,1,5),(4,1,3),(5,1,4),

--5

(1,5,4),(2,5,4),(3,5,4),(4,5,3),(5,5,4),

--6

(1,6,4),(2,6,4),(3,6,4),(4,6,3),(5,6,4),

--7

(1,7,3),(2,7,4),(3,7,5),(4,7,3),(5,7,4),

--8

(1,8,4),(2,8,4),(3,8,5),(4,8,3),(5,8,4),

--9

(1,9,5),(2,9,5),(3,9,5),(4,9,5),(5,9,5),

--10

(1,10,4),(2,10,4),(3,10,5),(4,10,3),(5,10,4),

--11

(1,11,4),(2,11,4),(3,11,5),(4,11,4),(5,11,4),

--12

(1,12,4),(2,12,4),(3,12,5),(4,12,3),(5,12,4),

--13

(1,13,4),(2,13,4),(3,13,4),(4,13,5),(5,13,4),

--14

(1,14,4),(2,14,4),(3,14,5),(4,14,3),(5,14,4),

--15

(1,15,5),(2,15,5),(3,15,5),(4,15,5),(5,15,5),

--16

(1,16,4),(2,16,4),(3,16,5),(4,16,3),(5,16,4),

--17

(1,17,4),(2,17,4),(3,17,5),(4,17,3),(5,17,4),

--18

(1,18,4),(2,18,4),(3,18,5),(4,18,3),(5,18,4),

--19

(1,19,4),(2,19,4),(3,19,5),(4,19,3),(5,19,4);

INSERT INTO Timetable (DateOfWeek,LessonNumber,ClassID,SubjectID) VALUES

--9а

('Понедельник', 1, 1, 1),('Понедельник', 2, 1, 2)

,('Вторник', 1, 1, 2),('Вторник', 2, 1, 5),('Вторник', 3, 1, 4)

,('Среда', 1, 1, 5),('Среда', 2, 1, 3)

,('Четверг', 1, 1, 1),('Четверг', 2, 1, 2)

,('Пятница', 1, 1, 4),

--9б

('Понедельник', 1, 2, 2)

,('Вторник', 1, 2, 3),('Вторник', 2, 2, 1)

,('Среда', 1, 2, 2),('Среда', 2, 2, 5)

,('Четверг', 1, 2, 4),('Четверг', 2, 2, 3),('Четверг', 3, 2, 1)

,('Пятница', 1, 2, 2),

--10

('Понедельник', 1, 3, 5),('Понедельник', 2, 3, 1),('Понедельник', 3, 3, 4)

,('Вторник', 1, 3, 1),('Вторник', 2, 3, 4)

,('Среда', 1, 3, 4)

,('Четверг', 1, 3, 3),('Четверг', 2, 3, 1)

,('Пятница', 1, 3, 1),('Пятница', 2, 3, 2),

--11

('Понедельник', 1, 4, 3)

,('Вторник', 1, 4, 5),('Вторник', 2, 4, 2)

,('Среда', 1, 4, 3),('Среда', 2, 4, 4)

,('Четверг', 1, 4, 2)

,('Пятница', 1, 4, 5),('Пятница', 2, 4, 1),('Пятница', 3, 4, 3);

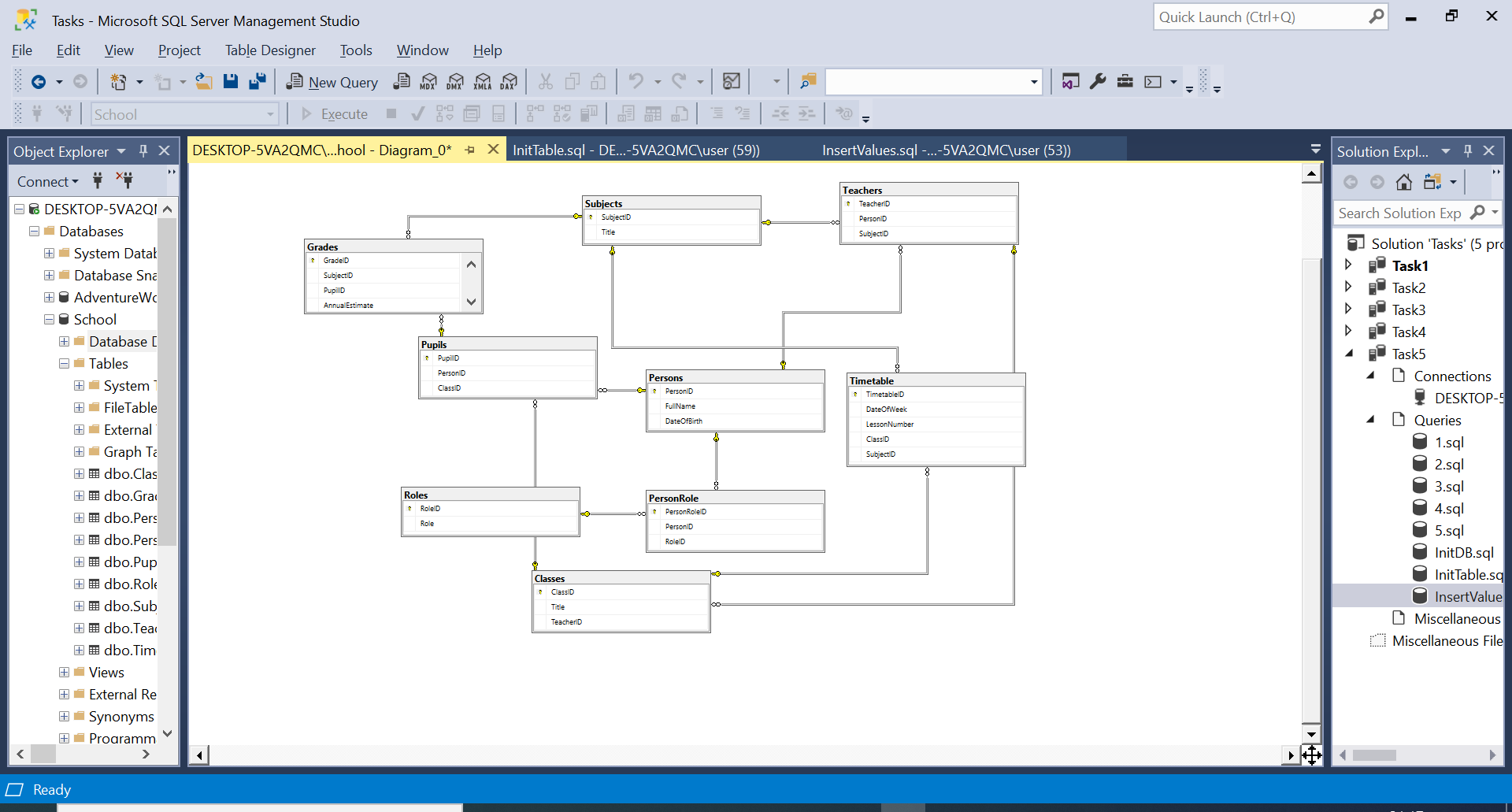


Рисунок 2 – ER-диаграмма, сгенерированная в Microsoft SQL Server Management Studio

***Разработка необходимых запросов***

1. Показывать список учащихся данного класса.

SELECT Classes.Title, Persons.FullName

FROM Classes

INNER JOIN Pupils

ON Pupils.ClassID = Classes.ClassID

INNER JOIN Persons

ON Persons.PersonID = Pupils.PersonID

WHERE Classes.Title = '10';

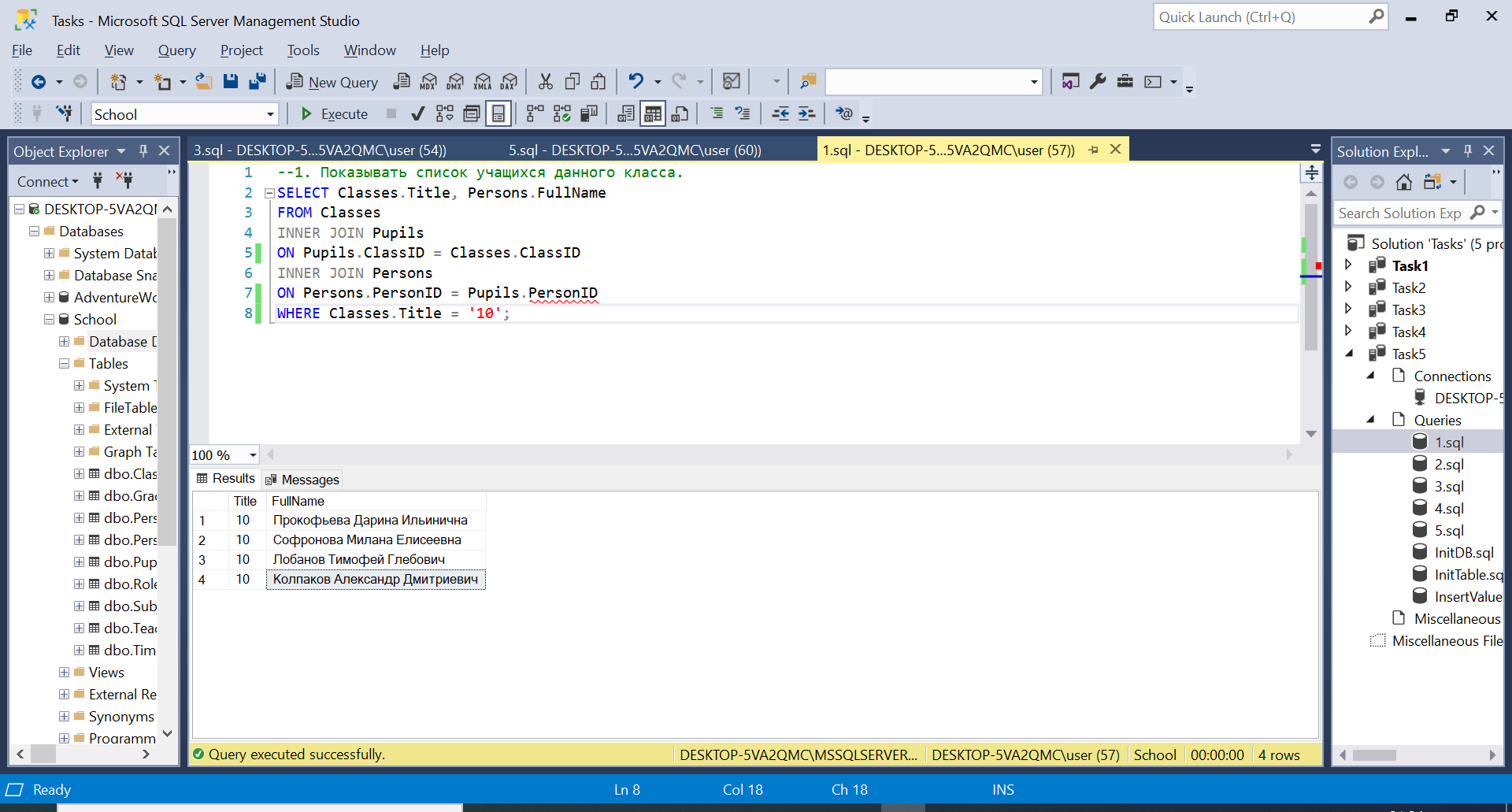


Рисунок 3 – Результат выполнения 1-го запроса

1. Показывать классного руководителя данного класса.

SELECT Classes.Title,Persons.FullName

FROM Classes

INNER JOIN Teachers

ON Classes.TeacherID = Teachers.TeacherID

INNER JOIN Persons

ON Persons.PersonID = Teachers.PersonID

WHERE Classes.Title = '9А';

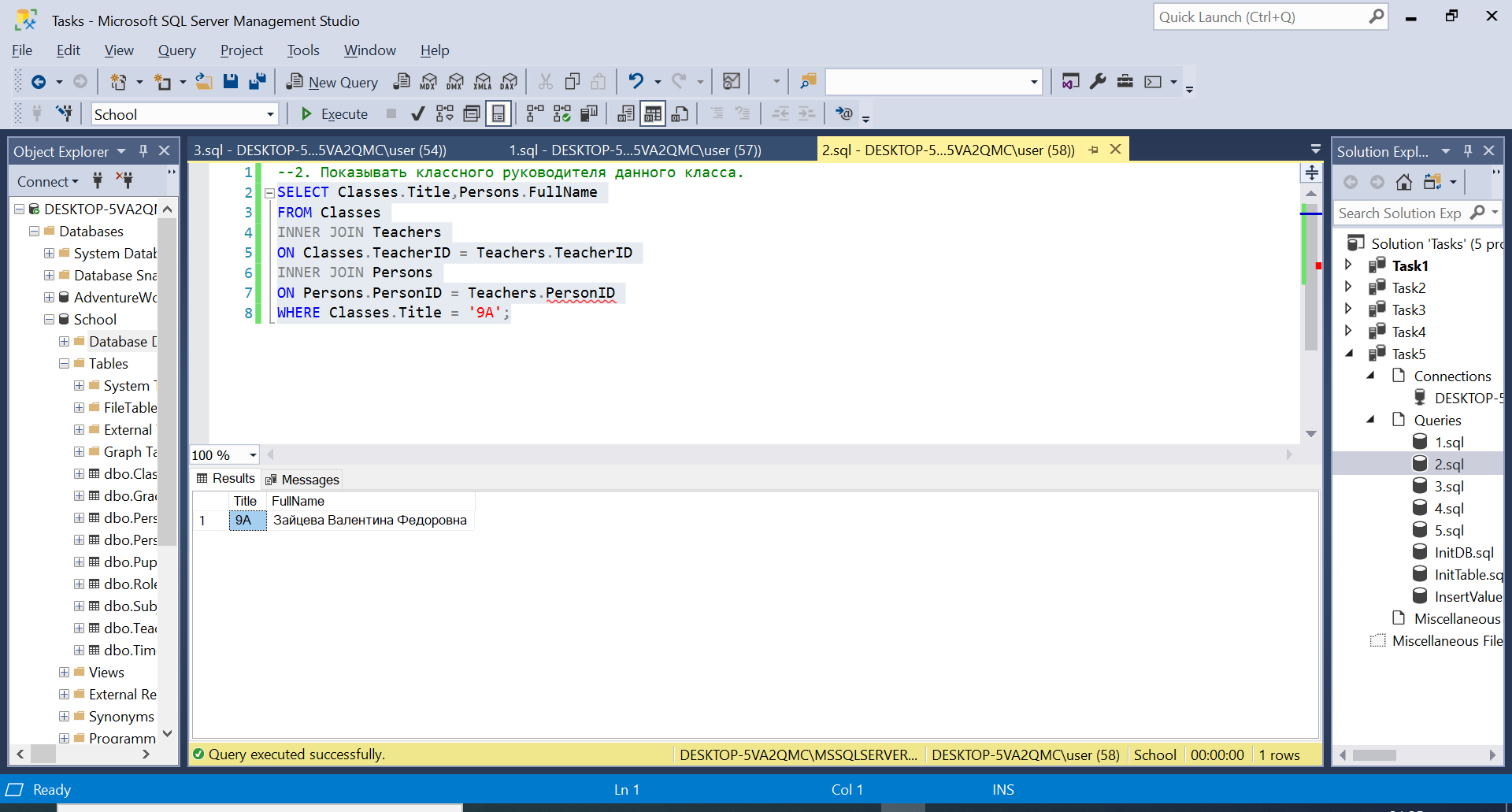


Рисунок 4 – Результат выполнения 2-го запроса

1. Показывать занятость данного учителя.

SELECT Classes.Title

,Timetable.DateOfWeek

,Timetable.LessonNumber

,Subjects.Title

,Persons.FullName

FROM Timetable

INNER JOIN Subjects

ON Timetable.SubjectID = Subjects.SubjectID

INNER JOIN Classes

ON Classes.ClassID = Timetable.ClassID

INNER JOIN Teachers

ON Teachers.SubjectID = Subjects.SubjectID

INNER JOIN Persons

ON Persons.PersonID = Teachers.PersonID

WHERE Teachers.TeacherID = 4;

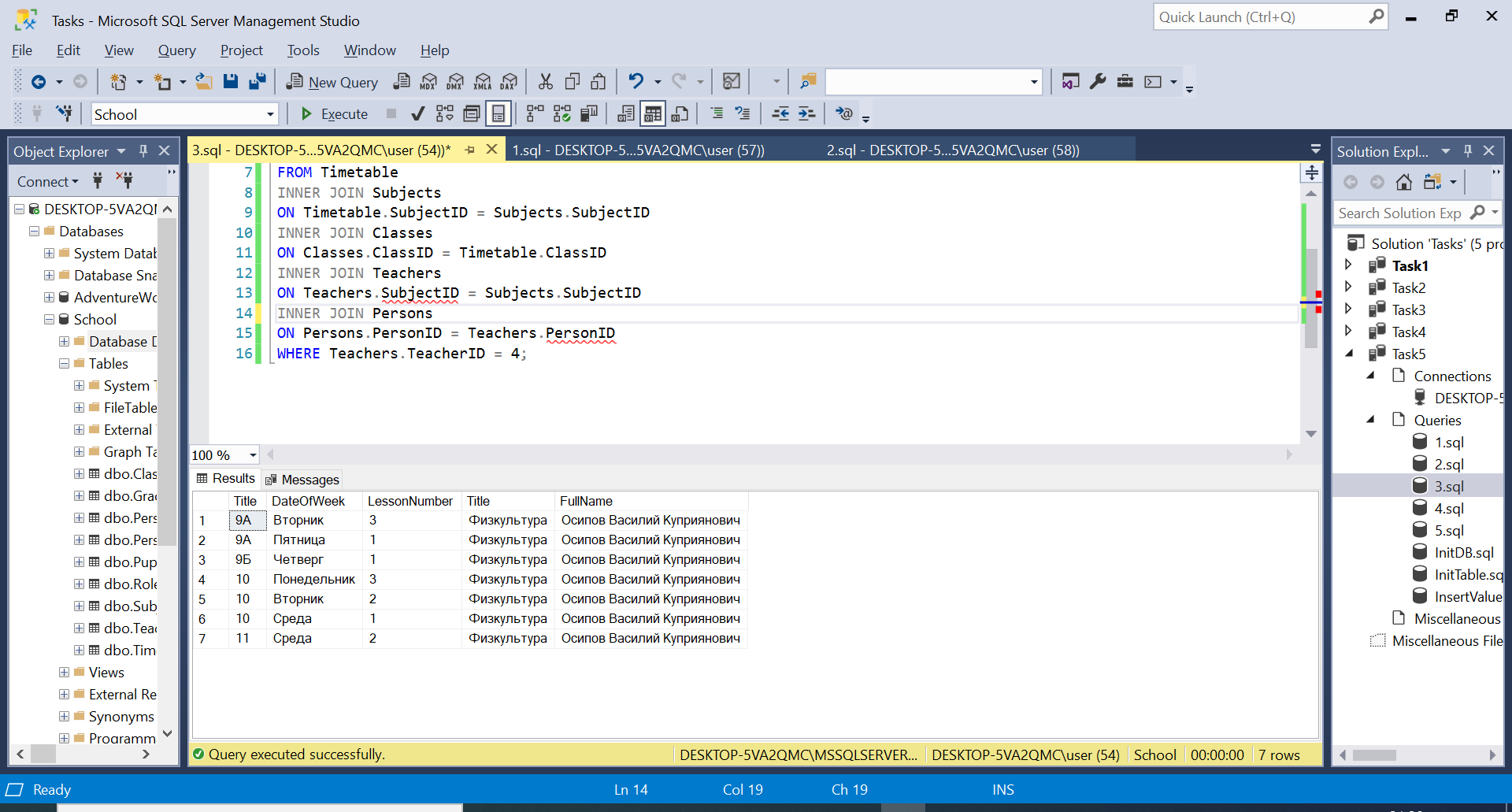


Рисунок 5 – Результат выполнения 3-го запроса

1. Показывать успеваемость данного ученика.

SELECT Persons.FullName

,Subjects.Title

,Grades.AnnualEstimate

FROM Grades

INNER JOIN Pupils

ON Pupils.PupilID = Grades.PupilID

INNER JOIN Subjects

ON Subjects.SubjectID = Grades.SubjectID

INNER JOIN Persons

ON Persons.PersonID = Pupils.PersonID

WHERE Persons.PersonID = 2;

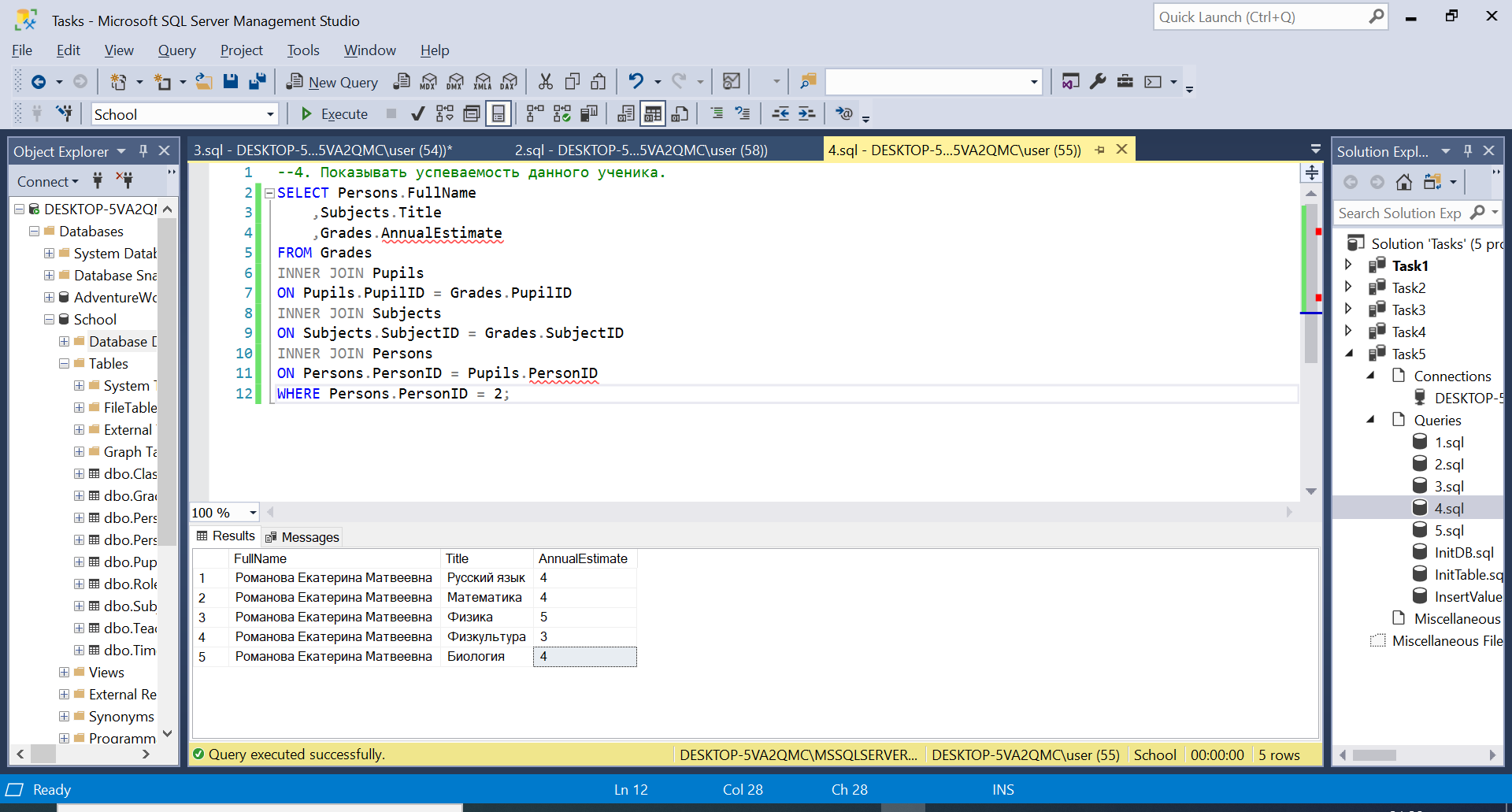


Рисунок 6 – Результат выполнения 4-го запроса

1. Показывать список учеников, учащихся без троек.

SELECT Persons.FullName

FROM Grades

INNER JOIN Pupils

ON Pupils.PupilID = Grades.PupilID

INNER JOIN Subjects

ON Subjects.SubjectID = Grades.SubjectID

INNER JOIN Persons

ON Persons.PersonID = Pupils.PersonID

WHERE Grades.AnnualEstimate <> 3

GROUP BY Persons.FullName

HAVING COUNT(\*)=5;

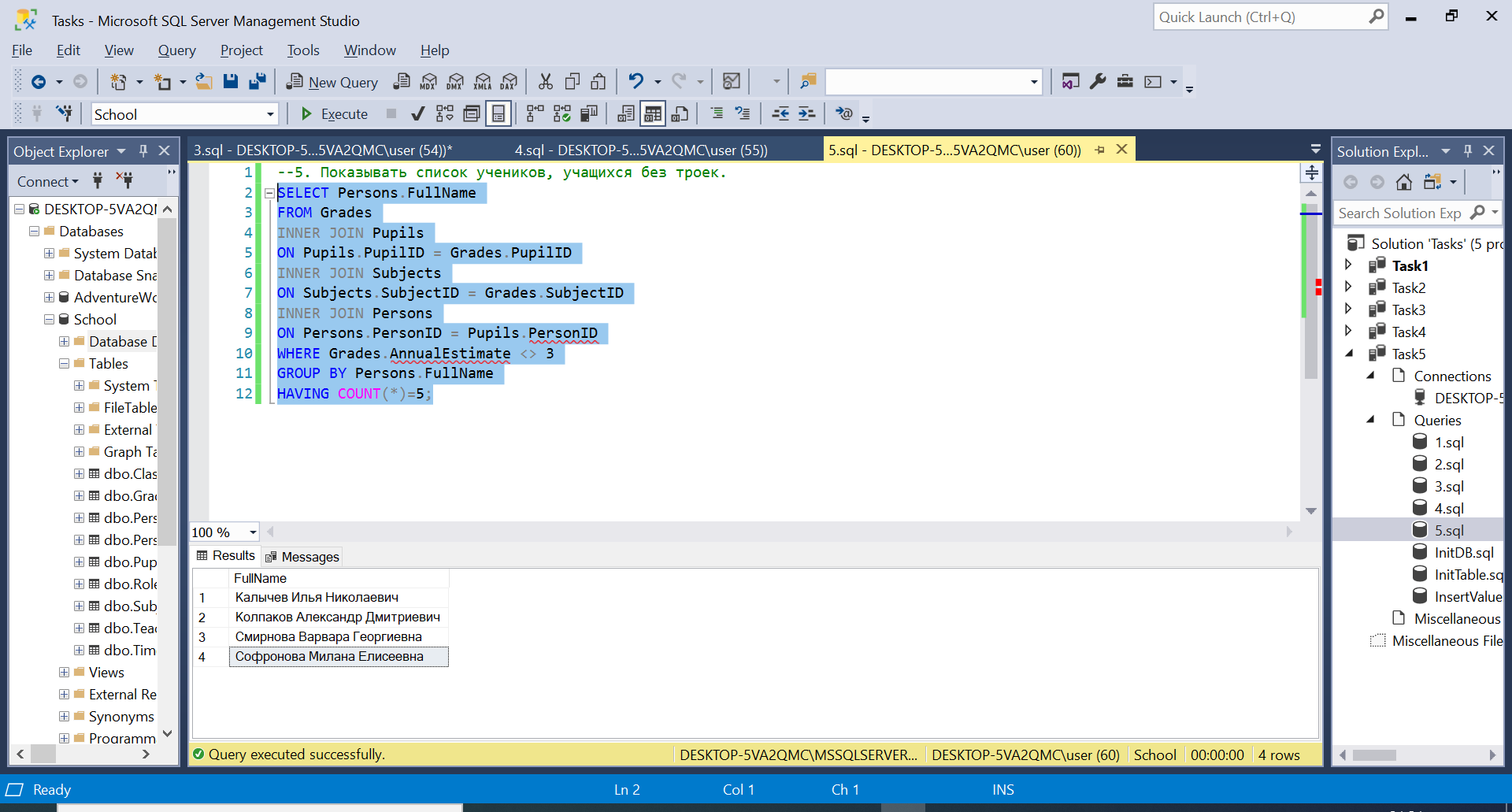


Рисунок 7 – Результат выполнения 5-го запроса

# ВЫВОД

По завершению курсового проекта были изучены методы и закреплены знания в проектировании реляционных баз данных (РБД) с примерами и заданиями для исполнения в системе управления базами данных (СУБД) Microsoft SQL Server для БД «Школа».

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Балакина Е.П., Васильева М.А., Филипченко К.М. Информационное обеспечение систем управления. Учебно-методическое пособие. Издание второе, исправленное и дополненное – М.: РУТ (МИИТ), 2020. –с.