МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВГОРОДСКОЙ ОБЛАСТИ

ОГА ПОУБоровичский техникум строительной индустрии и

экономики

Специальность 09.02.07 «Информационные системы и программирование

Учебная практика по ПМ.11

«Разработка, администрирование и защита баз данных»

УП.11 27 00 00 ДО

Выполнил:

Студент группы П-31

\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В.Шихов

Руководитель практики:

\_\_\_\_\_\_\_\_\_\_ Е.В. Винокурова

2022

Пусть требуется создать программную систему, предназначенную для завуча школы. Она должна обеспечивать хранение сведений о каждом учителе, о предметах, которые он преподает, номере закрепленного за ним кабинета, о расписании занятий. Существуют учителя, которые не имеют собственного кабинета. Об учениках должны храниться следующие сведения: фамилия и имя, в каком классе учится, какую оценку имеет в текущей четверти по каждому предмету. Завуч должен иметь возможность добавить сведения о новом учителе или ученике, внести в базу данных четвертные оценки учеников каждого класса по каждому предмету, удалить данные об уволившемся учителе и отчисленном из школы ученике, внести изменения в данные об учителях и учениках, в том числе поменять оценку ученика по тому или иному предмету. В задачу завуча входит также составление расписания. Завучу могут потребоваться следующие сведения:

Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

Кто из учителей преподает в заданном классе?

В каком кабинете будет 5-й урок в среду у некоторого класса?

В каких классах преподает заданный предмет заданный учитель?

Расписание на заданный день недели для указанного класса?

Сколько учеников в указанном классе?

Необходимо предусмотреть возможность получения документа, представляющего собой отчет о работе школы. В этот отчет следует включить сведения об успеваемости за четверть по каждому предмету для каждого класса. Порядок следования классов в отчете – от младших к старшим. Необходимо подсчитать средний балл по каждому предмету, по каждому классу и по школе в целом, указать общее количество учеников в классе и школе, количество учеников, имеющих в четверти только оценки «5», только «4» и «5», имеющих хотя бы одну оценку «3», хотя бы одну оценку «2». Для каждого класса указать классного руководителя. Кроме того, в отчете должно быть указано количество учителей по каждому предмету и общее количество кабинетов в школе.

1 Предметная область: Школа

2 Пользователь БД: Завуч Школы

Функции:

* Добавление учеников
* Отчисление учеников
* Добавление учителей
* Увольнение учителей
* Составление расписания
* Выставление оценок ученикам
* Возможность изменения данных об учителях
* Возможность изменения данных об учениках
* Возможность изменения оценок учеников
* Формирование отчёта

**Запросы**

1 Выводит учителя преподающего в указанный день недели в указанный урок

2 Выводит количество учеников в конкретной группе

3 Выводит в каком кабинете будет 5-й урок в среду у некоторого класса

4 Выводит расписание на заданный день недели для указанного класса

5 Выводит количество преподавателей ведущих определённый предмет

6 Выводит список учеников получивших 2 за первую четверть по любому предмету

7 Выводит список учеников получивших 2 за первую четверть по конкретному предмету

8 В каких классах преподает заданный предмет заданный учитель

9 Вывод классов, сортированный от меньших к большим

10 Вывод количества учеников в школе

**Сущности базы данных**

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Teacher | Перечень преподавателей |
| 2 | Subject | Перечень предметов |
| 3 | Classroom | Перечень кабинетов |
| 4 | Grade | Описание оценок |
| 5 | Pupil | Перечень учеников |
| 6 | Schedule | Описание расписания |
| 7 | Class | Перечень классов |

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название связи | Сущности | Назначение |
| 1 | 1:1 | Кабинет:Учитель | За 1 кабинетом может быть закреплен 1 учитель |
| 2 | 1:М | Оценка:Предмет | У оценки может быть множество предметов |
| 3 | М:1 | Оценка:Ученик | У оценки много учеников |
| 4 | М:1 | Ученик:Класс | У класса много учеников |
| 5 | 1:М | Расписание:Класс | У расписания может быть только 1 класс |
| 6 | 1:М | Расписание:Предмет | У расписания есть предмет |
| 7 | 1:М | Расписание:Кабинет | У множества расписаний может быть множество кабинетов |
| 8 | 1:М | Расписание:Учитель | У расписания может быть множество учителей |
| 9 | 1:М | Учитель:Предмет | У множества учителей может быть множество предметов |

**Словарь данных**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Класс Class** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **ClassID** | Код класса | int | Да |  |
|  | **ClassName** | Наименование класса | nvarchar(10) | Да |  |
| Внешний | **TeacherID** | Классный руководитель | int | Да | Внешний ключ к таблице Учителя |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Кабинет Classroom** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **Classroom** | Наименование кабинета | int | Да |  |
| Внешний | **TeacherID** | Код учителя | int | Нет | Внешний ключ к таблице учители |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Оценки Grade** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **GradeID** | Код оценки | int | Да |  |
| Внешний | **SubjectID** | Код предмета | Int | Да | Внешний ключ к таблице предметы |
| Внешний | **PupilID** | Код ученика | int | Да | Внешний ключ к таблице ученики |
|  | **Grade** | Оценка | int | Да | Ограничение от 2 до 5 |
|  | **Date** | Дата | date | Да |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ученик Pupil** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **PupilID** | Код ученика | int | Да |  |
|  | **Surname** | Фамилия | nvarchar(50) | Да |  |
|  | **Name** | Имя | nvarchar(50) | Да |  |
|  | **Midname** | Отчество | nvarchar(50) | Нет |  |
| Внешний | **ClassID** | Код класса | Int | Да | Внешний ключ к таблице класс |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Schedule Расписание** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **SheduleID** | Код расписания | int | Да |  |
|  | **DayOfTheWeek** | День недели | nvarchar(20) | Да |  |
| Внешний | **ClassID** | Код класса | int | Да | Внешний ключ к таблице класс |
| Внешний | **SubjectID** | Код предмета | int | Да | Внешний ключ к таблице предмет |
| Внешний | **ClassroomID** | Код кабинета | int | Да | Внешний ключ к таблице кабинет |
|  | **LessonNumber** | Номер урока | int | Да |  |
|  | **TeacherID** | Учитель | int | Да | Внешний ключ к таблице учитель |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Предмет Subject** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **SubjectID** | Код предмета | int | Да |  |
|  | **SubjectName** | Наименование | nvarchar(50) | Да |  |

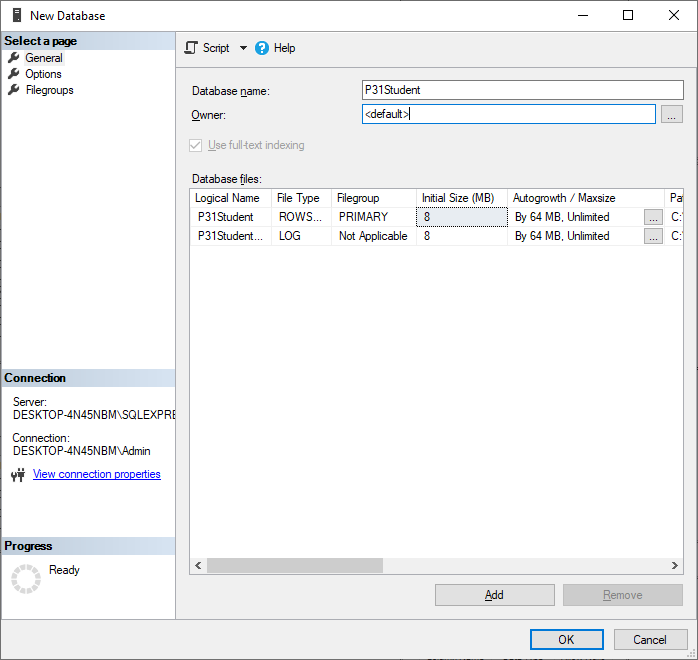
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ПредметыУчителей SubjectTeacher** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Внешний | **SubjectID** | Код предмета | int | Да | Внешний ключ к таблице предмет |
| Внешний | **TeacherID** | Код учителя | int | Да | Внешний ключ к таблице учитель |

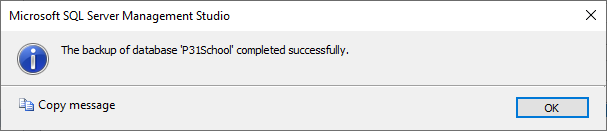
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Учитель Teacher** | | | | | |
| **Ключ** | **Поле** | **Поле (назначение)** | **тип** | **Обязательное** | **Примечание** |
| Первичный | **TeacherID** | Код учитель | int | Да |  |
|  | **Surname** | Фамилия | nvarchar(50) | Да |  |
|  | **Name** | Имя | nvarchar(50) | Да |  |
|  | **MidName** | Отчество | nvarchar(50) | Нет |  |

**Классификация сущностей**

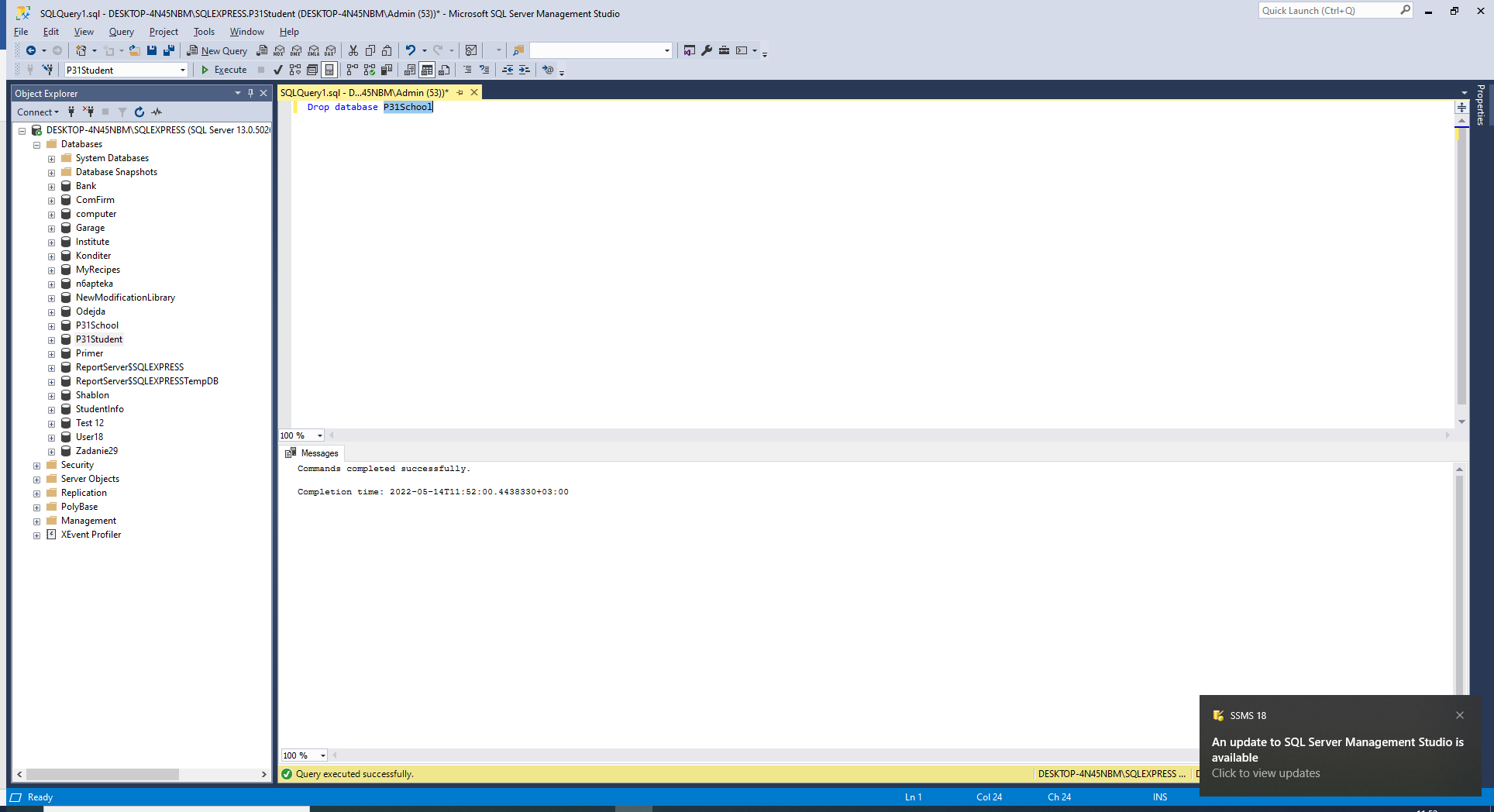
|  |  |  |
| --- | --- | --- |
| **№** | **Сущность** | **Тип** |
| 1 | Subject | Сильная |
| 2 | Teacher | Сильная |
| 3 | Classroom | Слабая |
| 4 | Pupil | Слабая |
| 5 | Grade | Слабая |
| 6 | Class | Сильная |
| 7 | Schedule | Слабая |

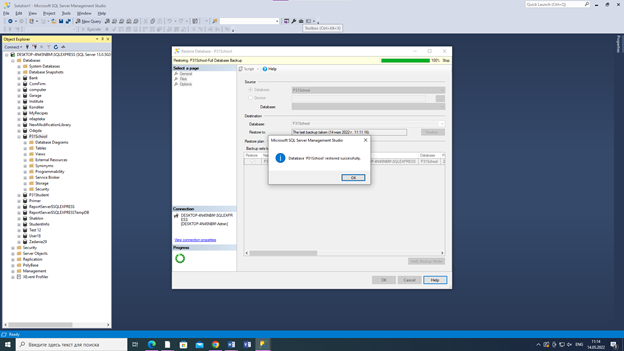
## Задание 2

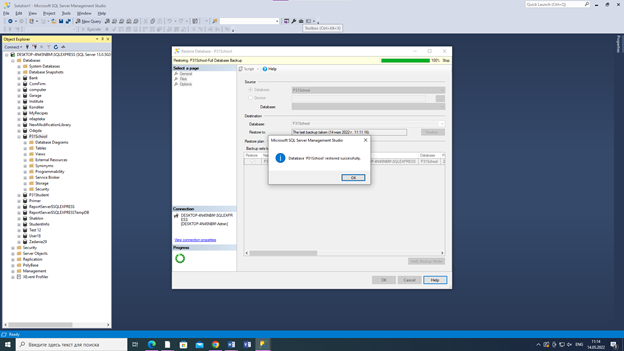




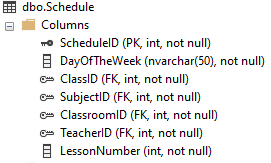
**Резервирование базы данных**

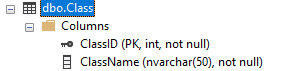


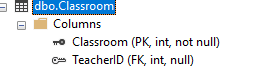
1. 

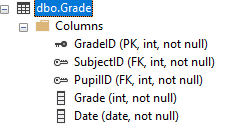


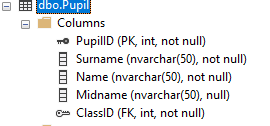
**Восстановление базы данных**

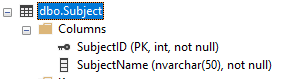
****

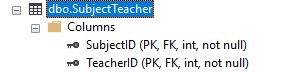
****

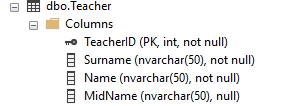
****

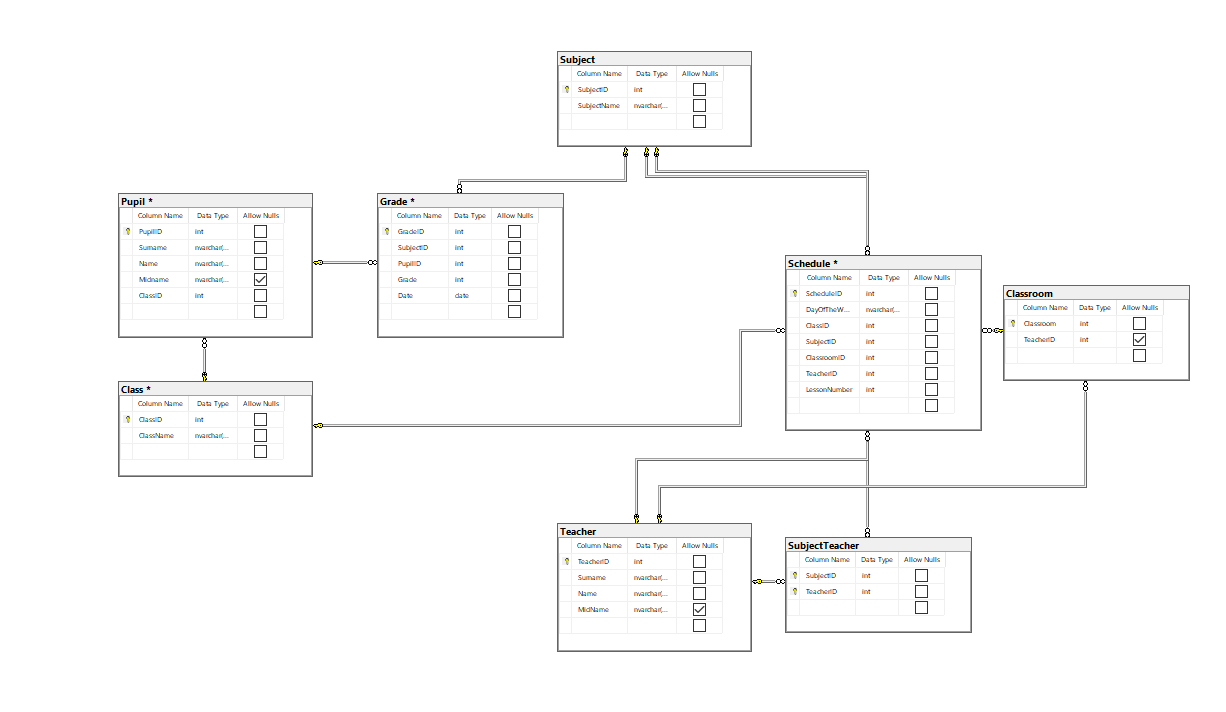
****

****

****

****

****

****

**Диаграмма БД**

## SQL Запросы

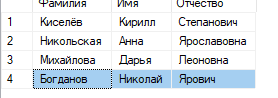
**2 простейших запроса с использованием операторов сравнения;**

Select Surname[Фамилия], [Name][Имя],Midname[Отчество] From Pupil Where ClassID = 1



Данный запрос выводит учеников группы 1А

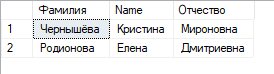
Select Surname[Фамилия], [Name][Имя],Midname[Отчество] From Teacher Where Surname != 'Раков'



Данный запрос выводит всех учеников, чья фамилия не Раков

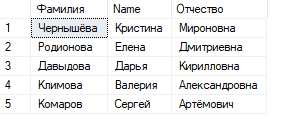
**3 запроса с использованием логических операторов AND,OR,NOT;**

Select p.Surname[Фамилия], p.[Name],p.Midname[Отчество] From Grade g join Pupil p on g.PupilID = p.PupilID Where p.ClassID = 1 Or p.ClassID = 2



Данный запрос выводит ФИО учеников из 1А или 1Б

Select Surname[Фамилия], [Name], Midname[Отчество] From Pupil Where Not Surname = 'Громов'



Данный запрос выводит учеников, чья фамилия не Громов

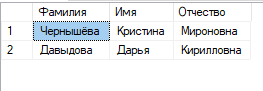
Select Surname[Фамилия], [Name], Midname[Отчество] From Pupil Where Surname = 'Чернышёва' and Midname = 'Мироновна'



Данный запрос выводит учеников чья фамилия – Чернышёва, а отчество Миронова

**1 запрос на использование комбинации логических операторов;**

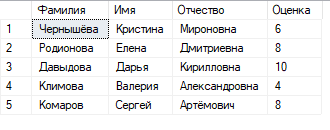
Select Surname[Фамилия], [Name][Имя], Midname[Отчество] From Pupil Where ClassID = 1 Or Not ClassID = 2 And ClassID = 3

****

Данный запрос выводит ФИО учеников, чей класс 1А или не 1Б и 1В

**1 запрос на использование выражений над столбцами;**

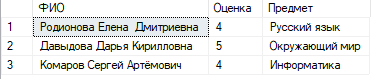
Select Surname[Фамилия], [Name][Имя], Midname[Отчество], Grade\*2[Оценкаф] From Grade g join Pupil p on p.PupilID = g.PupilID



Данный запрос выводит оценки учеников в 10 бальной системе

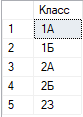
**2 Запроса с проверкой на принадлежность диапазону значений**

Select Concat(Surname ,' ', [Name], ' ', Midname) as 'ФИО', Grade[Оценка], SubjectName[Предмет] From Grade g join Subject s on s.SubjectID = g.SubjectID join Pupil p on p.PupilID = g.PupilID Where Grade between 4 and 5



Данный запрос выводит Учеников, получивших 4 или 5

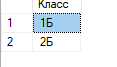
Select ClassName[Класс] from Class where SUBSTRING(ClassName,1,1) between 1 and 2

****

Данный запрос выводит список 1 и 2 классов

**2 Запроса с проверкой на соответствие шаблону**

Select ClassName[Класс] from Class where ClassName like '%Б'



Данный метод выводит Б классы

Select Distinct ClassName from Schedule s join Class c on s.ClassID = c.ClassID Where DayOfTheWeek like 'Понедельник'



Данный запрос выводит список классов, у которых есть уроки в понедельник

1. **1 запрос с проверкой на неопределенное значение**

Select CONCAT(Surname, ' ', [Name], ' ', Midname)[ФИО] From Pupil Where Midname Is Null

****

**Данный запрос выводит ФИО учеников без Отчества**

# **SQL Запросы на соединение таблиц**

1. **1 запрос с использованием декартового произведения двух таблиц;**

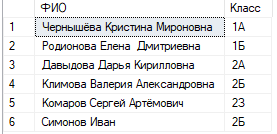
Select Surname[Фамилия], [Name][Имя], Midname[Отчество], ClassName[Класс] From Pupil,Class



Данный запрос выводит декартово произведение учеников и классов

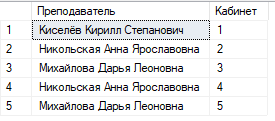
1. **2 запроса с использованием соединения двух таблиц по равенству;**

Select Concat(Surname, ' ', [Name], ' ', Midname)[ФИО], ClassName[Класс] from Pupil p join Class c on c.ClassID = p.ClassID



Данный запрос выводит ФИО и классы учеников

Select Concat(T.Surname, ' ', t.[Name], ' ', t.MidName)[Преподаватель], Classroom[Кабинет] From Teacher t join Classroom cs on t.TeacherID = cs.TeacherID



Данный список выводит Преподавателей и их предметы

1. **1 запрос с использованием соединения двух таблиц по равенству и условием отбора;**

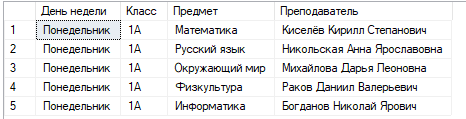
Select DayOfTheWeek[День недели], LessonNumber[Предмет],Concat(T.Surname, ' ', t.[Name], ' ', t.MidName)[Преподаватель] From Schedule s join Teacher t on s.TeacherID = t.TeacherID Join Classroom cs on cs.Classroom = s.ClassroomID where s.TeacherID != cs.TeacherID Order by DayOfTheWeek



Данный запрос выводит Преподавателя, его предмет и день недели

1. **1 запрос с использованием соединения по трем таблицам;**

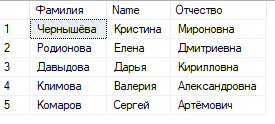
Select DayOfTheWeek[День недели], ClassName[Класс], SubjectName[Предмет], Concat(T.Surname, ' ', t.[Name], ' ', t.MidName)[Преподаватель] From Schedule s Join Class c on c.ClassID = s.ClassID join Teacher t on t.TeacherID = s.TeacherID join Subject sub on sub.SubjectID = s.SubjectID



Данный запрос выводит расписание

1. **1 запрос с использованием левого внешнего соединения;**

Select Distinct p.Surname[Фамилия], p.[Name],p.Midname[Отчество] From Grade g Left join Pupil p on g.PupilID = p.PupilID



Данный запрос выводит список учеников, получивших оценку

1. **1 запрос на использование правого внешнего соединения;**

Select Distinct p.Surname[Фамилия], p.[Name],p.Midname[Отчество] From Grade g Right join Pupil p on g.PupilID = p.PupilID Where SubjectID = 1

****

Данный запрос выводит список учеников, получивших оценку по определенному предмету

# **SQL Запросы на использование функций**

1. **1 запрос с использованием функции COUNT;**

Select Count(\*)[Преподавателей] from Teacher



Данный запрос выводит количество преподавателей

1. **1 запрос с использованием функции SUM;**

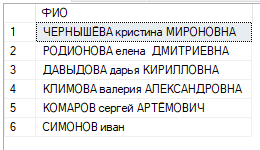
Select Sum(Grade) / Count(\*) as 'Cр балл' From Grade



Данный запрос выводит средний балл в школе

1. **1 запрос с использованием функций UPPER, LOWER;**

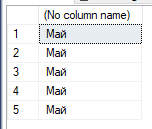
Select Concat(UPPER(Surname),' ', LOWER([Name]),' ', Upper([Midname]))[ФИО] From Pupil



Данный запрос выводит фамилию в верхнем регистре, имя в нижнем и отчество в верхнем учеников

1. **1 запрос с использованием временных функций;**

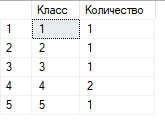
Select DATENAME(MONTH, Date) from Grade



Данный запрос выводит месяца выставленных оценок

1. **1 запрос с использованием группировки по одному столбцу;**

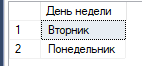
Select ClassID[Класс], Count(\*)[Количество] From Pupil Group By ClassID



Данный запрос выводит Количество учеников в классах

1. **1 запрос на использование группировки по нескольким столбцам;**

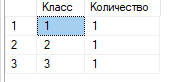
Select Distinct DayOfTheWeek[День недели] From Schedule s Join Class c on c.ClassID = s.ClassID join Teacher t on t.TeacherID = s.TeacherID join Subject sub on sub.SubjectID = s.SubjectID Group by DayOfTheWeek, s.ClassID



Данный запрос выводит дни недели на которые составлено расписание

1. **1 запрос с использованием условия отбора групп HAVING;**

Select ClassID[Класс], Count(\*)[Количество] From Pupil Group By ClassID Having ClassID < 4



Данный запрос выводит количество учеников первых трёх классов

1. **1 запрос с использованием фразы HAVING без фразы GROUP BY;**

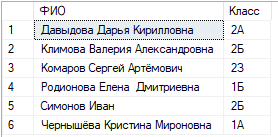
Select Count(\*) From Pupil Having Count(\*) > 5



Данный запрос выводит количество учеников, если их больше 5

1. **1 запрос с использованием сортировки по столбцу;**

Select Concat(Surname, ' ', [Name], ' ', Midname)[ФИО], ClassName[Класс] from Pupil p join Class c on c.ClassID = p.ClassID Order by Surname



Данный запрос выводит сведения об учениках отсортированный по фамилии

1. **1 запрос на добавление новых данных в таблицу;**

Insert into Class(ClassName) values ('3З')

Данный запрос добавляет класс 3З

1. **1 запрос на обновление существующих данных в таблице;**

Update Pupil set Midname = null



Данный запрос стирает отчество всем ученикам

1. **1 запрос на обновление существующих данных по условию WHERE;**

Update Class Set ClassName = '4З' Where ClassName = '3З'



Данный запрос переименовывает класс 3З в 4З

1. **1 запрос на удаление существующих данных.**

Delete From Class Where ClassName = '4З'



Данный запрос удаляет класс 4З

# **SQL Запросы на создание таблиц**

1. **1 запрос на создание любой родительской таблицы (из своего варианта)**

CREATE TABLE [dbo].[Class](

[ClassID] [int] IDENTITY(1,1) NOT NULL Primary Key,

[ClassName] [nvarchar](50) NOT NULL,

)



Данный запрос создаёт родительскую таблицу Class

1. **1 запрос на создание дочерней таблицы, прописать внешние ключи**

CREATE TABLE [dbo].[Pupil](

[PupilID] [int] IDENTITY(1,1) NOT NULL Primary Key,

[Surname] [nvarchar](50) NOT NULL,

[Name] [nvarchar](50) NOT NULL,

[Midname] [nvarchar](50) NULL,

[ClassID] [int] NOT NULL FOREIGN KEY REFERENCES Class(ClassID),

)



Данный запрос создаёт дочернюю таблицу Pupil