

# *Система для учета проведения занятий в школе*



# Описание предметной области.

**Создание модели данных с сущностями «Школа».**

**Разработка системы для учета проведения занятий в школе в течение одного года.**

В школе есть классы соответствующие разным годам обучения. Также в школе есть классы соответствующие одному году обучения.

Обучение в школе предполагает изучение определенных предметов.

Все классы одного года должны изучать одни и те же предметы. Могут существовать также факультативные предметы.

Каждый учитель школы может вести занятия по нескольким предметам. С классом может быть связан классный руководитель

## Информационные потребности пользователя:

- Отображения в системе состояние учебного процесса школы
- Поиск информации: возможность искать и фильтровать данные о классах, учениках, учителях и предметах.
- Обновление и управление данными: возможность добавлять, редактировать или удалять записи о студентах, учителях, предметах и классах.

Администрация  
(управление данными школы)

Учителя  
Доступ к расписанию

Ученики и родители  
Выбор расписания

## Данные в базе

Академический класс (годовой курс обучения в школе):  
номер класса, список букв класса

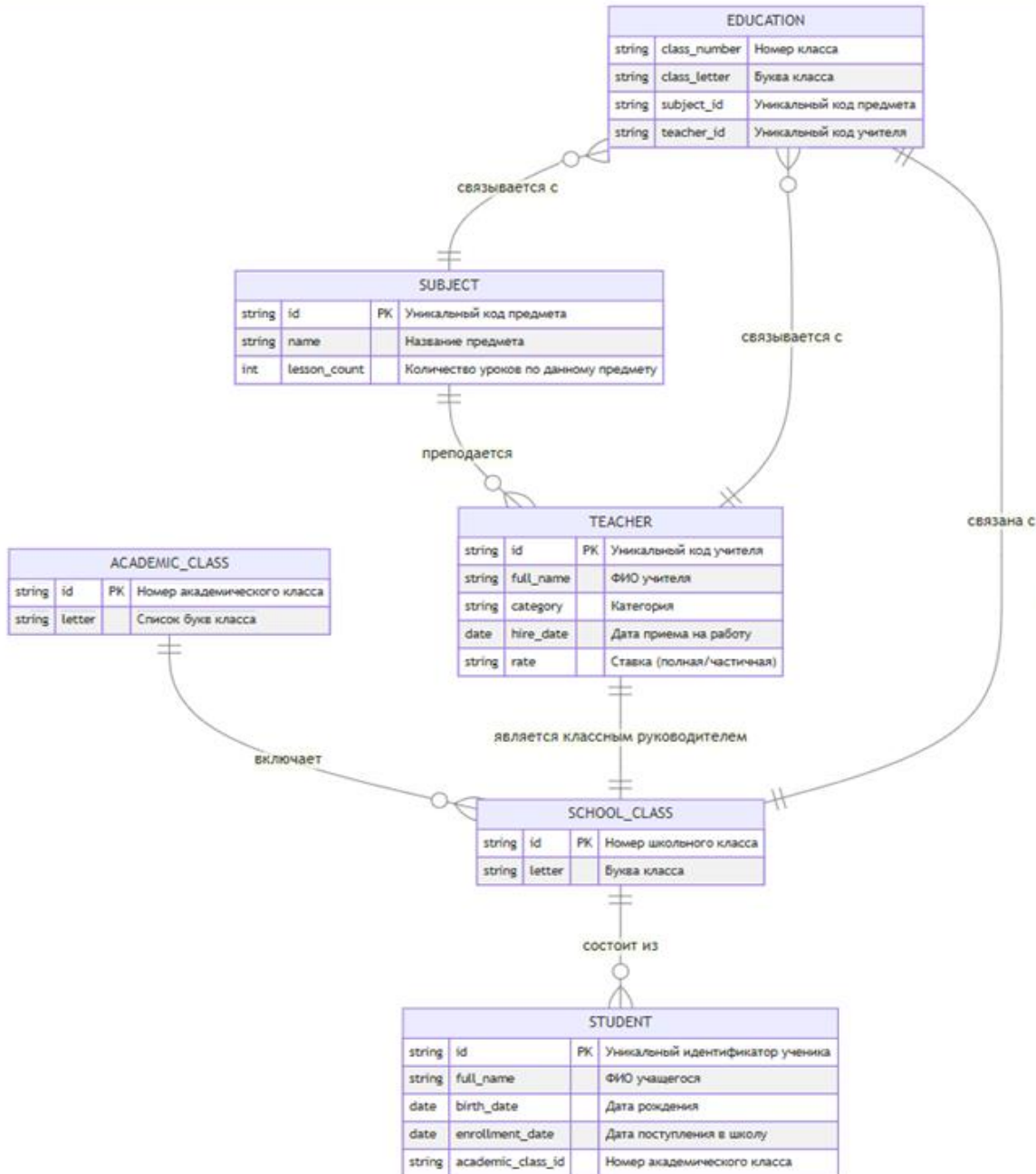
Предметы: ID предмета, название предмета, количество уроков

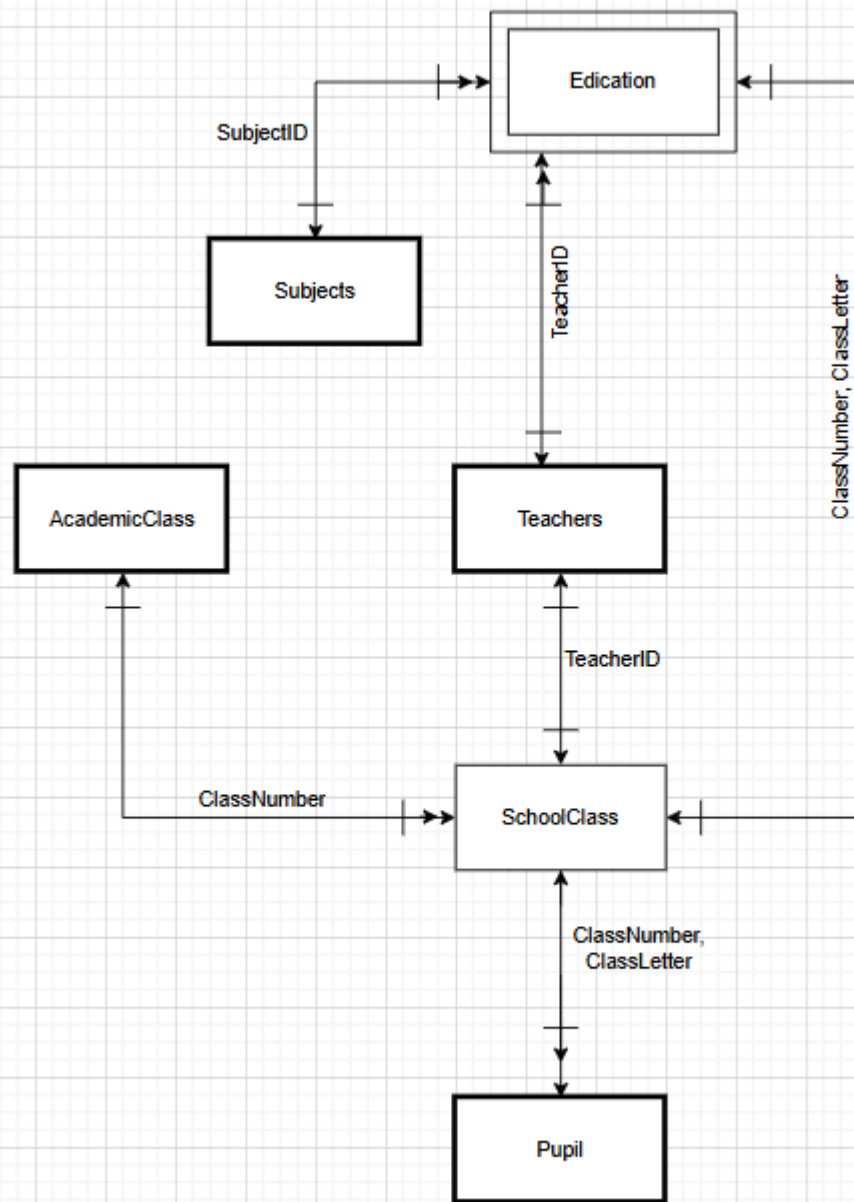
Учителя: ID учителя, ФИО, категория, дата приема на работу, ставка (полная/частичная)

Учащиеся: ID ученика, ФИО учащегося, дата рождения, дата поступления в школу, номер класса

Классы: номер класса, буква класса, ID учителя

Обучение: номер класса, буква класса, ID учителя, ID предмета





### • Описание связей

- 1:M Академический класс-Школьный класс. Каждый академический класс может включать в себя несколько школьных классов
- M:N Предмет-Учитель. Каждый предмет может преподаваться одним и более учителями. Каждый учитель преподаёт один и более предметов
- 1:1 Учитель-Школьный класс. Каждый учитель может являться классным руководителем только одного класса или не являться классным руководителем ни одного школьного класса. Каждый школьный класс может иметь только одного классного руководителя
- 1:M Школьный класс-Ученик. Каждый школьный класс состоит из нескольких учеников. Каждый ученик включен только в один школьный класс.
- 1:1 Образование-Школьный класс. Каждая запись в сводной таблице Образование соответствует конкретному школьному классу, а каждый школьный класс может быть связан только с одной записью в образовании
- M:1 Образование-Предмет. Каждый предмет в таблице Предмет может быть упомянут несколько раз в таблице Образование, но каждая запись в Образовании ссылается на один уникальный предмет
- M:1 Образование-Учитель. Каждый учитель в таблице Учитель может быть упомянут несколько раз в таблице Образование, но каждая запись в Образовании может ссылаться только на одного учителя

Создание физической модели

Множество сущностей	Название атрибута	Имя атрибута	Тип, длина	Примечание	Количество байт	Общее количество байт
Академический класс	Номер класса	ClassNumber	INT(2)	первичный ключ	4	
	Список класса	LetterList	VARCHAR(50)	обязательное поле	50+1	
Предмет	Код предмета	SubjectID	INT(3)	первичный ключ	4	55
	Название	SubjectName	VARCHAR(50)	обязательное поле	50+1	
	Количество уроков	lesson_count	INT(2)		4	
	Тип предмета	SubjectType	VARCHAR(20)	Обязательный или Факультативный	20+1	
	Список класса	ClassesList	VARCHAR(30)	обязательное поле	30+1	
Учитель	Описание	Description	VARCHAR(150)		150+1	262
	Код учителя	teacherID	INT(3)	первичный ключ	4	
	ФИО	full_name	VARCHAR(90)	обязательное поле	90+1	
	Категория	category	VARCHAR(20)		20+1	
	Дата приема на работу	hiring_date	Date	обязательное поле	3	
	Ставка	rate	DECIMAL(3,2)	обязательное поле	5	124
	Школьный класс	Номер класса	ClassNumber	внешний ключ (к AcademicClass)	4	
		Буква класса	ClassLetter	обязательное поле	1	
		Код учителя	teacherID	внешний ключ (к SchoolClass)	4	9
	Ученик	Код ученика	student_code	первичный ключ	4	
Обучение		ФИО	full_name	обязательное поле		
		Дата рождения	birth_date	обязательное поле	3	
		Дата поступления в школу	SchoolEntryDate	обязательное поле	3	
		Буква класса	ClassLetter	внешний составной ключ (к SchoolClass)	1	
		Номер класса	ClassNumber	внешний составной ключ (к SchoolClass)	4	
		Номер класса	ClassNumber	внешний составной ключ (к SchoolClass)	4	106
		Буква класса	ClassLetter	внешний составной ключ (к SchoolClass)	1	
		Код учителя	teacher_code	внешний составной ключ (к Tetchers)	4	
		Код предмета	subject_code	внешний составной ключ (к Subjects)	4	13
						569

## Привидение базы данных к 3НФ

- Избыточных данных нет
- .Каждое поле зависит от первичного ключа
- Транзитивных зависимостей нет

Академический класс	
Номер класса	Список класса
1	А, Б, В, Г
2	А, Б, В, Г
3	А, Б, В, Г
4	А, Б, В, Г
5	А, Б, В
6	А, Б, В
7	А, Б, В
8	А, Б, В
9	А, Б, В
10	А, Б
11	А, Б

Учитель				
Код учителя	ФИО	Категория	Дата приема на работу	Ставка
1	Иванов А.И	NULL	01.09.2023	1
2	Васильев Ю.А.	NULL	02.09.2023	1
3	Попова Е.С.	NULL	03.09.2023	1
3	Сидоров Ю.М.	NULL	04.09.2023	1
5	Петров А.И.	NULL	05.09.2023	1

Класс		
Номер класса	Буква класса	Код учителя
1	А	NULL
1	Б	NULL
1	В	NULL
1	Г	NULL
2	А	NULL
2	Б	NULL
2	В	NULL
2	Г	NULL
3	А	NULL

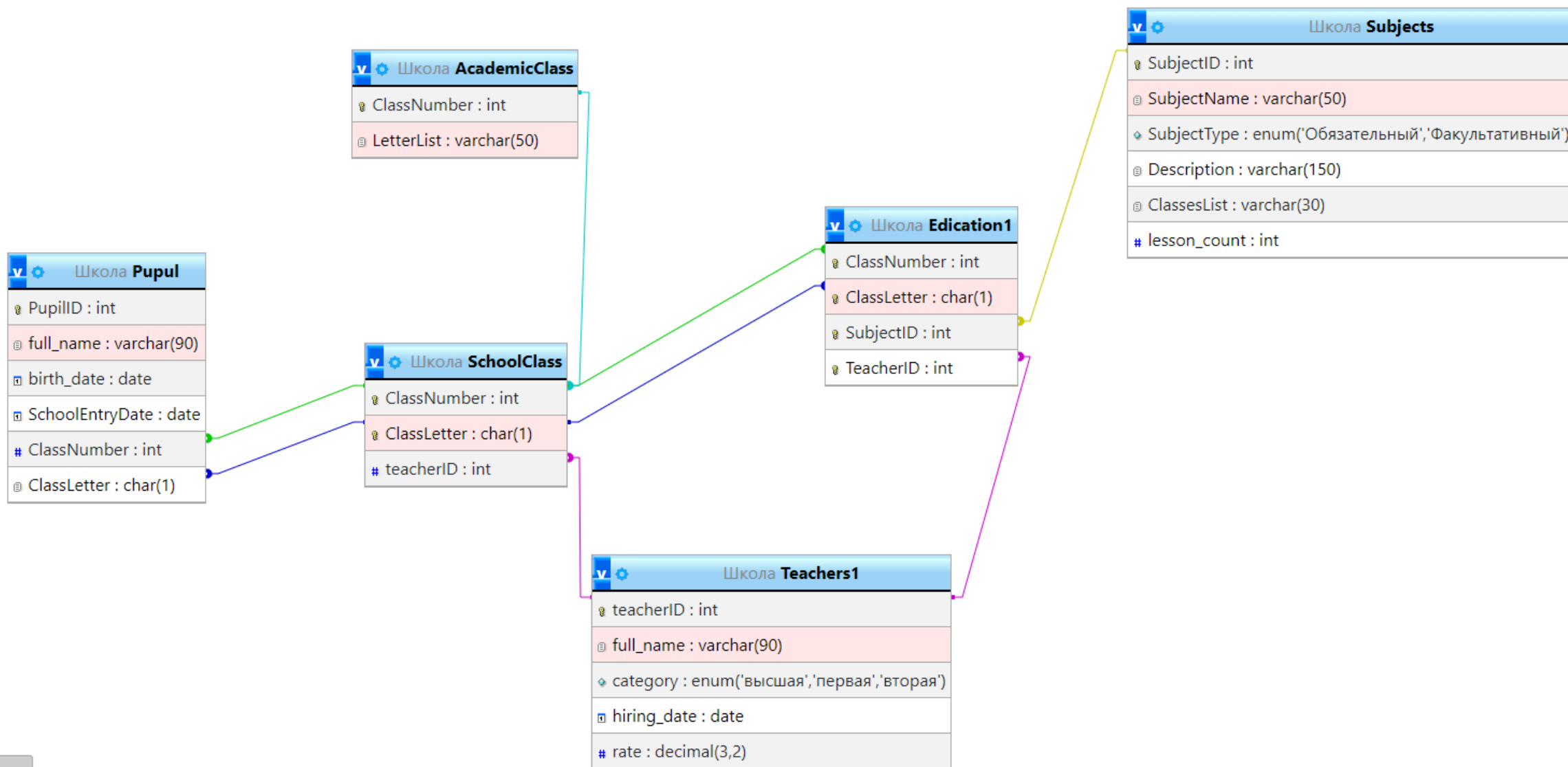
Предмет						
Код предмета	Имя предмета	Количество уроков	Тип предмета	Описание предмета	Список классов	Количество уроков
1	Русский язык	34	Обязательный	NULL	1,2,3,4,5,6,7,8,9,10,11	NULL
2	Литература	19	Обязательный	NULL	1,2,3,4,5,6,7,8,9,10,11	NULL
3	Английский язык	33	Обязательный	NULL	5,6,7,8,9,10,11	NULL
4	Информатика	25	Обязательный	NULL	6,7,8,9,10,11	NULL
5	Математика	34	Обязательный	NULL	1,2,3,4,5,6,7,8,9,10,11	NULL

Ученик					
Код ученика	ФИО	Дата рождения	Дата поступления в колледж	Номер класса	Буква класса
1	Кузнецова Анна Николаевна	13.10.1989	01.09.2023	2	А
2	Михайлова Мария Сергеевна	11.06.1998	01.09.2024	2	А
3	Чистякова Наталья Евгеньевна	14.02.2000	02.09.2024	3	Б
4	ригорьев Виктор Петрович	02.05.1991	01.09.2024	4	Г

Обучение			
Номер класса	Буква класса	Код учителя	Код предмета
1	А	1	1



## Настройка связей и проверка структуры в «Дизайнере»



# Написание SQL- кода для создания страниц

```
CREATE TABLE AcademicClass(  
  ClassNumber INT PRIMARY KEY,  
  
  LetterList VARCHAR(50) NOT  
  NULL  
);
```

```
CREATE TABLE Subjects (  
  SubjectID INT PRIMARY KEY  
  AUTO_INCREMENT,  
  SubjectName VARCHAR(50) NOT  
  NULL,  
  SubjectType  
  ENUM('Обязательный',  
  'Факультативный') DEFAULT  
  'Обязательный',  
  Description VARCHAR(150),  
  ClassesList VARCHAR(30) NOT NULL,  
  lesson_count INT  
);
```

```
CREATE TABLE Teachers(  
  teacherID INT PRIMARY KEY  
  AUTO_INCREMENT,  
  full_name VARCHAR(90) NOT  
  NULL,  
  category ENUM('высшая',  
  'первая', 'вторая') DEFAULT NULL,  
  hiring_date DATE NOT NULL,  
  rate DECIMAL(3,2) DEFAULT 1.00  
);
```

```
CREATE TABLE SchoolClass (  
  ClassNumber INT NOT NULL,  
  ClassLetter CHAR(1) NOT NULL,  
  teacherID INT NOT NULL,  
  PRIMARY KEY (ClassNumber, ClassLetter),  
  CONSTRAINT  
  fk_SchoolClass_AcademicClass FOREIGN  
  KEY (ClassNumber)  
  REFERENCES AcademicClass  
  (ClassNumber)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  CONSTRAINT fk_SchoolClass_Teacher  
  FOREIGN KEY (teacherID)  
  REFERENCES Teachers (teacherID)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE  
);
```

```
CREATE TABLE Pupil (  
    PupilID INT PRIMARY KEY  
    AUTO_INCREMENT,  
    full_name VARCHAR(90) NOT NULL,  
    birth_date DATE NOT NULL,  
    SchoolEntryDate DATE NOT NULL,  
    ClassNumber INT NOT NULL,  
    ClassLetter CHAR(1) NOT NULL,  
    CONSTRAINT fk_Pupil_SchoolClass FOREIGN  
    KEY (ClassNumber, ClassLetter)  
    REFERENCES SchoolClass(ClassNumber,  
    ClassLetter)  
);
```

```
CREATE TABLE Education (  
    ClassNumber INT NOT NULL,  
    ClassLetter CHAR(1) NOT NULL,  
    SubjectID INT NOT NULL,  
    TeacherID INT NOT NULL,  
    PRIMARY KEY (ClassNumber, ClassLetter, SubjectID, TeacherID),  
    CONSTRAINT edication_ibfk_1 FOREIGN KEY (ClassNumber,  
    ClassLetter)  
    REFERENCES SchoolClass (ClassNumber, ClassLetter)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
    CONSTRAINT edication_ibfk_2 FOREIGN KEY (SubjectID)  
    REFERENCES Subjects (SubjectID)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
    CONSTRAINT edication_ibfk_3 FOREIGN KEY (TeacherID)  
    REFERENCES Teachers (teacherID)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT  
);
```

# Запросы на языке SQL

```
INSERT INTO AcademicClass  
(ClassNumber, LetterList) VALUES  
(1, 'А, Б, В, Г'),  
(2, 'А, Б, В, Г'),  
(3, 'А, Б, В, Г'),  
(4, 'А, Б, В, Г'),  
(5, 'А, Б, В'),  
(6, 'А, Б, В'),  
(7, 'А, Б, В'),  
(8, 'А, Б, В'),  
(9, 'А, Б, В'),  
(10, 'А, Б'),  
(11, 'А, Б');
```

```
INSERT INTO Subjects (SubjectID, SubjectName, SubjectType, Description,  
ClassesList, lesson_count) VALUES  
(1, 'Русский язык', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL),  
(2, 'Литература', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL),  
(3, 'Иностранный язык', 'Обязательный', NULL, '5,6,7,8,9,10,11', NULL),  
(4, 'Математика', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL),  
(5, 'Информатика', 'Обязательный', NULL, '6,7,8,9,10,11', NULL),  
(6, 'Всемирная история', 'Обязательный', NULL, '5,6,7,8,9,10,11', NULL),  
(7, 'География', 'Обязательный', NULL, '6,7,8,9,10,11', NULL),  
(8, 'Биология', 'Обязательный', NULL, '6,7,8,9,10,11', NULL),  
(9, 'Химия', 'Обязательный', NULL, '7,8,9,10,11', NULL),  
(10, 'Астрономия', 'Обязательный', NULL, '10,11', NULL),  
(11, 'Физкультура', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL),  
(12, 'Черчение', 'Обязательный', NULL, '7,8,9,10,11', NULL);
```

### Команда UPDATE

Предположим, что вы хотите изменить учителя для определенного класса, например, для класса "1A", чтобы преподаватель с ID 1 был заменен на преподавателя с ID 2:

Вот пример команды

```
UPDATE SchoolClass  
SET TeacherID = 2  
WHERE ClassNumber = 1 AND  
ClassLetter = 'A' AND TeacherID = 1;
```

Этот запрос изменит все записи для класса "1A", где учитель с ID 1, и заменит их на учителя с ID 2.

### Проверка команды DELETE

Теперь давайте удалим записи для класса "2Г", где преподаватель с ID 7.

Вот пример команды

```
DELETE FROM Education  
WHERE ClassNumber = 2 AND  
ClassLetter = 'Г' AND TeacherID = 7;
```

Этот запрос удалит все записи из таблицы Education, относящиеся к классу "2Г", где учитель с ID 7.

# Написание SQL-запросов для выборки, извлечения, сортировки и группировки

## 1. Выборка данных из таблицы

Запрос для извлечения всех данных из таблицы Pupil:

```
SELECT * FROM Pupil;
```

## 4. Использование HAVING и GROUP BY для фильтрации после группировки

Выборка предметов, преподаваемых определенным преподавателем

```
SELECT
    Teachers.full_name,
    COUNT(DISTINCT Education.SubjectID) AS subject_count
FROM
    Teachers
JOIN
    Education ON Teachers.teacherID = Education.TeacherID
GROUP BY
    Teachers.teacherID
HAVING
    COUNT(DISTINCT Education.SubjectID) > 1;
;
```

## 2. Фильтрация данных с использованием WHERE

Этот SQL-запрос извлекает данные о студентах из таблицы Pupil, и выводит только те записи, где студент принадлежит к классу 2A.

```
SELECT full_name, ClassNumber, ClassLetter
FROM Pupil WHERE ClassNumber = 2 AND ClassLetter = 'A';
```

## 3. Сортировка данных с использованием ORDER BY

Запрос для сортировки учеников по имени (по возрастанию):

```
SELECT full_name, ClassNumber, ClassLetter
FROM Pupil
ORDER BY full_name ASC;
```

```
SELECT full_name, ClassNumber, ClassLetter
FROM Pupil
ORDER BY full_name DESC;
```

# Выполнение SQL-запросов

✓ Добавлено 11 строк. (Запрос занял 0,0151 сек.)

```
INSERT INTO AcademicClass (ClassNumber, LetterList) VALUES (1, 'А, Б, В, Г'), (2, 'А, Б, В, Г'), (3, 'А, Б, В, Г'), (4, 'А, Б, В, Г'), (5, 'А, Б, В'), (6, 'А, Б, В'), (7, 'А, Б, В'), (8, 'А, Б, В'), (9, 'А, Б, В'), (10, 'А, Б'), (11, 'А, Б');
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Добавлено 12 строк. (Запрос занял 0,0154 сек.)

```
INSERT INTO Subjects (SubjectID, SubjectName, SubjectType, Description, ClassesList, lesson_count) VALUES (1, 'Русский язык', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL), (2, 'Литература', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL), (3, 'Иностранный язык', 'Обязательный', NULL, '5,6,7,8,9,10,11', NULL), (4, 'Математика', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL), (5, 'Информатика', 'Обязательный', NULL, '6,7,8,9,10,11', NULL), (6, 'Всемирная история', 'Обязательный', NULL, '5,6,7,8,9,10,11', NULL), (7, 'География', 'Обязательный', NULL, '6,7,8,9,10,11', NULL), (8, 'Биология', 'Обязательный', NULL, '6,7,8,9,10,11', NULL), (9, 'Химия', 'Обязательный', NULL, '7,8,9,10,11', NULL), (10, 'Астрономия', 'Обязательный', NULL, '10,11', NULL), (11, 'Физкультура', 'Обязательный', NULL, '1,2,3,4,5,6,7,8,9,10,11', NULL), (12, 'Черчение', 'Обязательный', NULL, '7,8,9,10,11', NULL);
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Добавлено 7 строк. (Запрос занял 0,0023 сек.)

```
INSERT INTO Teachers (teacherID, full_name, category, hiring_date, rate) VALUES (1, 'Иванов Иван Иванович', NULL, '2010-08-15', 1.00), (2, 'Дорохова Елена Александровна', NULL, '2023-01-01', 1.00), (3, 'Сидорова Светлана Александровна', NULL, '2018-09-10', 1.00), (4, 'Кузнецова Анна Викторовна', NULL, '2012-03-22', 1.00), (5, 'Михайлов Михаил Евгеньевич', NULL, '2016-11-05', 1.00), (6, 'Данилов Александр Сергеевич', NULL, '2023-01-10', 1.00), (7, 'Земченко Александр Игоревич', NULL, '2025-01-01', 1.00);
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Добавлено 35 строк. (Запрос занял 0,0155 сек.)

```
INSERT INTO SchoolClass(ClassNumber, ClassLetter, teacherID) VALUES (1, 'A', 1), (2, 'A', 1), (3, 'A', 1), (4, 'A', 1), (1, 'Б', 2), (2, 'Б', 2), (3, 'Б', 2), (4, 'Б', 2), (1, 'В', 3), (2, 'В', 3), (3, 'В', 3), (4, 'В', 3), (1, 'Г', 4), (2, 'Г', 4), (3, 'Г', 4), (4, 'Г', 4), (5, 'А', 4), (6, 'А', 4), (7, 'А', 4), (8, 'А', 4), (9, 'А', 4), (10, 'А', 4), (11, 'А', 4), (5, 'Б', 5), (6, 'Б', 5), (7, 'Б', 5), (8, 'Б', 5), (9, 'Б', 5), (10, 'Б', 5), (11, 'Б', 5), (5, 'В', 6), (6, 'В', 6), (7, 'В', 6), (8, 'В', 6), (9, 'В', 6);
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Добавлено 8 строк. (Запрос занял 0,0151 сек.)

```
INSERT INTO Pupil (PupilID, full_name, birth_date, SchoolEntryDate, ClassNumber, ClassLetter) VALUES (4, 'Кузнецова Анна Николаевна', '2005-07-19', '2011-09-01', 2, 'А'), (5, 'Михайлова Мария Сергеевна', '2005-09-10', '2011-09-01', 2, 'А'), (6, 'Дмитриев Александр Владимирович', '2005-11-30', '2011-09-01', 2, 'Б'), (7, 'Чистякова Наталья Евгеньевна', '2005-04-03', '2011-09-01', 3, 'Б'), (8, 'Григорьев Виктор Петрович', '2005-02-15', '2011-09-01', 3, 'Б'), (9, 'Федорова Дарина Владимировна', '2005-08-25', '2011-09-01', 3, 'В'), (10, 'Ковалев Вячеслав Викторович', '2005-12-02', '2011-09-01', 4, 'В'), (11, 'Лебедев Даниил Артемович', '2005-10-10', '2011-09-01', 4, 'В');
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Добавлена 31 строка. (Запрос занял 0,0154 сек.)

```
INSERT INTO Education (ClassNumber, ClassLetter, SubjectID, TeacherID) VALUES (1, 'А', 1, 1), (1, 'А', 2, 2), (1, 'А', 3, 3), (1, 'А', 4, 4), (1, 'А', 5, 5), (1, 'А', 6, 6), (1, 'А', 7, 7), (1, 'Б', 1, 1), (1, 'Б', 2, 2), (1, 'Б', 3, 3), (1, 'Б', 4, 4), (1, 'Б', 5, 5), (1, 'Б', 6, 6), (1, 'Б', 7, 7), (1, 'В', 1, 1), (1, 'В', 2, 2), (1, 'В', 3, 3), (1, 'В', 4, 4), (1, 'В', 5, 5), (1, 'В', 6, 6), (1, 'В', 7, 7), (1, 'Г', 1, 1), (1, 'Г', 2, 2), (1, 'Г', 3, 3), (1, 'Г', 4, 4), (1, 'Г', 5, 5), (1, 'Г', 6, 6), (1, 'Г', 7, 7), (2, 'А', 1, 1), (2, 'А', 2, 2), (2, 'А', 3, 3);
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Затронута 1 строка. (Запрос занял 0,0010 сек.)

```
UPDATE Education SET TeacherID = 2 WHERE ClassNumber = 1 AND ClassLetter = 'А' AND TeacherID = 1;
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)

✓ Затронута 0 строк. (Запрос занял 0,0003 сек.)

```
DELETE FROM Education WHERE ClassNumber = 2 AND ClassLetter = 'Г' AND TeacherID = 7;
```

[\[ Построчное редактирование \]](#) [\[ Изменить \]](#) [\[ Создать PHP-код \]](#)



SELECT \* FROM Pupil;

Профилирование

[\[ Построчное редактирование \]](#)
[\[ Изменить \]](#)
[\[ Анализ SQL запроса \]](#)
[\[ Создать PHP-код \]](#)
[\[ Обновить \]](#)

☐ Показать все

Количество строк:

Фильтровать строки:

Сортировать по кли

Extra options

	PupilID	full name	birth date	SchoolEntryDate	ClassNumber	ClassLetter
<input type="checkbox"/>	4	Кузнецова Анна Николаевна	2005-07-19	2011-09-01	2	A
<input type="checkbox"/>	5	Михайлова Мария Сергеевна	2005-09-10	2011-09-01	2	A
<input type="checkbox"/>	6	Дмитриев Александр Владимирович	2005-11-30	2011-09-01	2	B
<input type="checkbox"/>	7	Чистякова Наталья Евгеньевна	2005-04-03	2011-09-01	3	B
<input type="checkbox"/>	8	Григорьев Виктор Петрович	2005-02-15	2011-09-01	3	B
<input type="checkbox"/>	9	Федорова Дарина Владимировна	2005-08-25	2011-09-01	3	B
<input type="checkbox"/>	10	Ковалев Вячеслав Викторович	2005-12-02	2011-09-01	4	B
<input type="checkbox"/>	11	Лебедев Даниил Артемович	2005-10-10	2011-09-01	4	B

`SELECT full_name, ClassNumber, ClassLetter FROM Pupil WHERE ClassNumber = 2 AND ClassLetter = 'A';`

☐ Профилирование

[\[ Построение редактирование \]](#) [\[ Изменить \]](#) [\[ Анализ SQL запроса \]](#) [\[ Создать PHP-код \]](#) [\[ Обновить \]](#)

☐ Показать все    Количество строк:     Фильтровать строки:     Сортировать по к...

Extra options

	<a href="#">full_name</a>	<a href="#">ClassNumber</a>	<a href="#">ClassLetter</a>
<input type="checkbox"/> Кузнецова Анна Николаевна	2	A	
<input type="checkbox"/> Михайлова Мария Сергеевна	2	A	

`SELECT full_name, ClassNumber, ClassLetter FROM Pupil ORDER BY full_name ASC;`

☐ Профилирование

[ [Построчное редактирование](#) ] [ [Изменить](#) ] [ [Анализ SQL запроса](#) ] [ [Создать PHP-код](#) ] [ [Обновить](#) ]

☐ Показать все | Количество строк:  | Фильтровать строки:

Extra options

	<a href="#">full_name</a> ↕ 1	<a href="#">ClassNumber</a>	<a href="#">ClassLetter</a>
<input type="checkbox"/>	Григорьев Виктор Петрович	<a href="#">3</a>	<a href="#">Б</a>
<input type="checkbox"/>	Дмитриев Александр Владимирович	<a href="#">2</a>	<a href="#">Б</a>
<input type="checkbox"/>	Ковалев Вячеслав Викторович	<a href="#">4</a>	<a href="#">В</a>
<input type="checkbox"/>	Кузнецова Анна Николаевна	<a href="#">2</a>	<a href="#">А</a>
<input type="checkbox"/>	Лебедев Даниил Артемович	<a href="#">4</a>	<a href="#">В</a>
<input type="checkbox"/>	Михайлова Мария Сергеевна	<a href="#">2</a>	<a href="#">А</a>
<input type="checkbox"/>	Федорова Дарина Владимировна	<a href="#">3</a>	<a href="#">В</a>
<input type="checkbox"/>	Чистякова Наталья Евгеньевна	<a href="#">3</a>	<a href="#">Б</a>

```
SQL-запрос успешно выполнен.

SELECT Teachers.full_name, COUNT(DISTINCT Education.SubjectID) AS subject_count FROM Teachers JOIN Education ON Teachers.teacherID = Education.TeacherID GROUP BY Teachers.teacherID HAVING COUNT(DISTINCT Education.SubjectID) > 1;
```

☐ Профилирование

[\[ Построенное редактирование \]](#) [\[ Изменить \]](#) [\[ Анализ SQL запроса \]](#) [\[ Создать PHP-код \]](#) [\[ Обновить \]](#)

Extra options

full_name	subject_count
Иванов Иван Иванович	2

# Составление представлений

## 1. Представление, которое связывает ученика с его классным руководителем.

```
CREATE VIEW PupilsWithClassTeacher AS
SELECT
    Pupil.PupilID,
    Pupil.full_name AS PupilName,
    Pupil.ClassNumber,
    Pupil.ClassLetter,
    Teachers.full_name AS TeacherName
FROM
    Pupil
JOIN
    SchoolClass ON Pupil.ClassNumber = SchoolClass.ClassNumber
                AND Pupil.ClassLetter = SchoolClass.ClassLetter
JOIN
    Teachers ON SchoolClass.teacherID = Teachers.teacherID;
```

## 2. Представление изучаемых предметов в академическом классе

```
CREATE VIEW AcademicClassSubjects AS
SELECT DISTINCT
    AcademicClass.ClassNumber,
    Subjects.SubjectName
FROM
    AcademicClass
JOIN
    Education ON AcademicClass.ClassNumber =
Education.ClassNumber
JOIN
    Subjects ON Education.SubjectID =
Subjects.SubjectID;
```

### **3.Представление преподаватели-Ставки**

```
CREATE VIEW TeacherRates AS  
SELECT Teachers.full_name, Teachers.category,  
Teachers.rate  
FROM Teachers;
```

### **5. Представление, которое показывает количество учеников в каждом классе**

```
CREATE VIEW ClassStudentCount AS  
SELECT  
    SchoolClass.ClassNumber,  
    SchoolClass.ClassLetter,  
    COUNT(Pupil.PupilID)  
FROM  
    SchoolClass  
LEFT JOIN  
    Pupil ON SchoolClass.ClassNumber =  
Pupil.ClassNumber AND SchoolClass.ClassLetter =  
Pupil.ClassLetter  
GROUP BY  
    SchoolClass.ClassNumber, SchoolClass.ClassLetter;
```

### **4.Представление, которое показывает Количество предметов в каждом классе**

```
CREATE VIEW ClassSubjectCount AS  
SELECT  
    Education.ClassNumber,  
    Education.ClassLetter,  
    COUNT(DISTINCT Education.SubjectID)  
FROM  
    Education  
GROUP BY  
    Education.ClassNumber,  
Education.ClassLetter;
```

# ПРАВА ДОСТУПА

- **Полные права для администрации школы**

- 
- Права на таблицы:
- GRANT ALL PRIVILEGES ON AcademicClass TO admin\_user;
- GRANT ALL PRIVILEGES ON Education TO admin\_user;
- GRANT ALL PRIVILEGES ON Pupil TO admin\_user;
- GRANT ALL PRIVILEGES ON SchoolClass TO admin\_user;
- GRANT ALL PRIVILEGES ON Subjects TO admin\_user;
- GRANT ALL PRIVILEGES ON Teachers TO admin\_user;
- 
- Права на представления
- GRANT ALL PRIVILEGES ON academicclasssubjects TO admin\_user;
- GRANT ALL PRIVILEGES ON classtudentcount TO admin\_user;
- GRANT ALL PRIVILEGES ON classsubjectcount TO admin\_user;
- GRANT ALL PRIVILEGES ON pupilswithclassteacher TO admin\_user;
- GRANT ALL PRIVILEGES ON teacherrates TO admin\_user;

- **Права для учителей:**

- Учителя не могут удалять данные, но могут вставлять, обновлять и просматривать данные в таблицах и представлениях.
- 
- Права на таблицы:
- GRANT SELECT, INSERT, UPDATE ON AcademicClass TO teacher\_user;
- GRANT SELECT, INSERT, UPDATE ON Education TO teacher\_user;
- GRANT SELECT, INSERT, UPDATE ON Pupil TO teacher\_user;
- GRANT SELECT, INSERT, UPDATE ON SchoolClass TO teacher\_user;
- GRANT SELECT, INSERT, UPDATE ON Subjects TO teacher\_user;
- GRANT SELECT, INSERT, UPDATE ON Teachers TO teacher\_user;
- 
- Права на представления:
- GRANT SELECT ON academicclasssubjects TO teacher\_user;
- GRANT SELECT ON classtudentcount TO teacher\_user;
- GRANT SELECT ON classsubjectcount TO teacher\_user;
- GRANT SELECT ON pupilswithclassteacher TO teacher\_user;
- GRANT SELECT ON teacherrates TO teacher\_user

- **Права для учеников:**

- Права на таблицы
- GRANT SELECT ON Pupil TO pupil\_user;
- GRANT SELECT ON SchoolClass TO pupil\_user;
- 
- Права на представления:
- GRANT SELECT ON academicclasssubjects TO pupil\_user;
- GRANT SELECT ON classtudentcount TO pupil\_user;
- GRANT SELECT ON classsubjectcount TO pupil\_user;
- GRANT SELECT ON pupilswithclassteacher TO pupil\_user;





## Преимущества системы

- Улучшение эффективности управления
- Доступность данных
- Удобство хранения и поиска данных
- Анализ и отчетность
- Сокращение ошибок:
- Упрощение взаимодействия между родителями, учителями и администрацией
- Повышение безопасности данных
- Автоматическая поддержка стандартов и нормативных требований

# Заключение

- Внедрение системы управления данными для школы значительно улучшает организацию процессов, ускоряет обработку информации и повышает точность учета.
- Учет проведенных занятий в школе позволяет эффективно управлять учебным процессом, ускоряет доступ к важным данным и способствует улучшению взаимодействия между учениками, преподавателями и администраторами.
- Это делает систему не только удобной для использования, но и легко адаптируемой под будущие нужды учебного заведения.

**Спасибо за внимание!**

