

**Тема:**  
**Выборка из одной таблицы.**  
**Выборка с добавлением. Выбор уникальных строк.**  
**Извлечение диапазона строк. Сортировка.**  
**Фильтрация данных.**  
**Использование в запросах операторов сравнения, логических операторов**

## Инструкция SELECT.

<https://learn.microsoft.com/ru-ru/sql/t-sql/queries/select-transact-sql?view=sql-server-ver15>

-- Syntax for SQL Server and Azure SQL Database

```
<SELECT statement> ::=
[ WITH { [ XMLNAMESPACES , ] [ <common_table_expression> [,...n] ] } ]
<query_expression>
[ ORDER BY <order_by_expression> ]
[ <FOR Clause> ]
[ OPTION ( <query_hint> [ ,...n ] ) ]
<query_expression> ::=
{ <query_specification> | ( <query_expression> ) }
[ { UNION [ ALL ] | EXCEPT | INTERSECT }
  <query_specification> | ( <query_expression> ) [,...n] ]
<query_specification> ::=
SELECT [ ALL | DISTINCT ]
  [ TOP ( expression ) [ PERCENT ] [ WITH TIES ] ]
  <select_list>
  [ INTO new_table ]
  [ FROM { <table_source> } [ ,...n ] ]
  [ WHERE <search_condition> ]
  [ <GROUP BY> ]
  [ HAVING <search_condition> ]
```

## SELECT (Transact-SQL)

Возвращает строки из базы данных и позволяет делать выборку одной или нескольких строк или столбцов из одной или нескольких таблиц в SQL Server.

Полный синтаксис инструкции SELECT сложен, однако основные предложения можно кратко описать следующим образом:

```
[ WITH { [ XMLNAMESPACES , ] [ <common_table_expression> ] } ]
SELECT выбранный список [ INTO новая таблица ]
[ FROM источник таблицы ] [ WHERE условие поиска ]
[ GROUP BY выражение группирования ]
[ WINDOW выражение окна ]
[ HAVING условие поиска ]
[ ORDER BY выражение упорядочения [ ASC | DESC ] ]
```

Операторы UNION, EXCEPT и INTERSECT можно использовать между запросами, чтобы сравнить их результаты или объединить в один результирующий набор.

**Порядок предложений в инструкции SELECT имеет значение.** Любое из необязательных предложений может быть опущено; но если необязательные предложения используются, они должны следовать в определенном порядке.

### Логический порядок обработки инструкции SELECT

Следующие действия демонстрируют логический порядок обработки или порядок привязки инструкции SELECT. Этот порядок определяет, когда объекты, определенные в одном шаге, становятся доступными для предложений в последующих шагах.

FROM	определяет имена используемых таблиц
JOIN	
GROUP BY	группирует строки, имеющие одинаковые значения в указанном столбце;
HAVING	фильтрует группы строк в соответствии с указанным условием
DISTINCT	
ON	
WHERE	фильтрует строки таблицы в соответствии с заданными условиями
WITH CUBE или WITH ROLLUP	
SELECT	форматирует выходные данные
ORDER BY	сортирует результаты выполнения запроса

Порядок предложений в запросе SELECT не может быть изменен.

**Предложения SELECT и FROM являются обязательными, присутствие остальных зависит от контекста.**

В предложении SELECT указывается список столбцов, которые должны быть возвращены запросом. Можно указать исходные элементы или вычисляемые поля во время выполнения запроса.

**Конструкция DISTINCT | ALL исключает / разрешает вывод повторяющихся строк.**

Конструкция ALL используется по умолчанию.

\* означает вывод всех столбцов указанной таблицы.

В случае, если выборка производится из нескольких таблиц, перед символом звездочки может указываться имя таблицы.

SQL-запрос может содержать вычисляемые столбцы, значения которых могут определяться на основе значений данных, хранящихся в БД конструкции.

**Вычисляемым столбцам следует давать название с помощью ключевого слова AS.**

**Вычисляемый столбец можно создать как: <Новое поле> = <выражение>**

Если название столбца состоит из нескольких слов, разделенных пробелами, следует их записать в квадратных скобках: [].

Сортировка данных выполняется с помощью команды ORDER BY, которая добавляется в конец запроса, после чего перечисляется список столбцов. Для каждого столбца указывается тип сортировки ASC | DESC (ascending – по возрастанию | descending – по убыванию). ASC – по умолчанию, можно не указывать.

Конструкция TOP <N> позволяет выбрать определенное количество строк из таблицы. Дополнительный оператор PERCENT позволяет выбрать процентное количество строк из таблицы. Дополнительный оператор WITH TIES позволяет выбрать все строки с такими же свойствами.

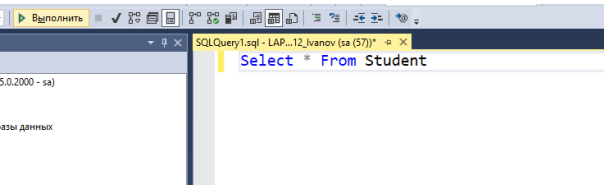
Конструкция OFFSET <N> ROWS указывает число строк, которые необходимо пропустить, прежде чем будет начат возврат строк из выражения запроса.

Конструкция FETCH NEXT <N> ROWS ONLY указывает число строк, возвращаемых после обработки предложения OFFSET.

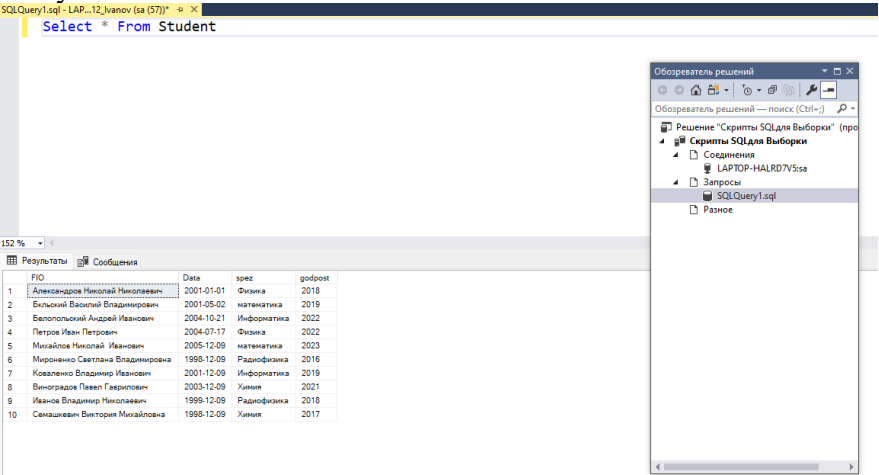
Имеется таблица Student:

ФИО	Дата_рождения	Специальность	Год поступления
Александров Николай Николаевич	01.01.2000	Физика	2018
Бельский Василий Владимирович	02.05.2001	Математика	2019
Белопольский Андрей Иванович	21.10.2004	Информатика	2022
Петров Иван Петрович	17.07.2004	Физика	2022
Михайлов Николай Иванович	09.12.2005	математика	2023
Мироненко Светлана Владимировна	09.12.1998	Радиофизика	2016
Коваленко Владимир Иванович	09.12.2001	Информатика	2019
Виноградов Павел Гаврилович	09.12.2003	Химия	2021
Иванов Владимир Николаевич	09.12.1999	Радиофизика	2018
Семашкевич Виктория Михайловна	09.12.1998	Химия	2017

Вывести список всех СТУДЕНТОВ:

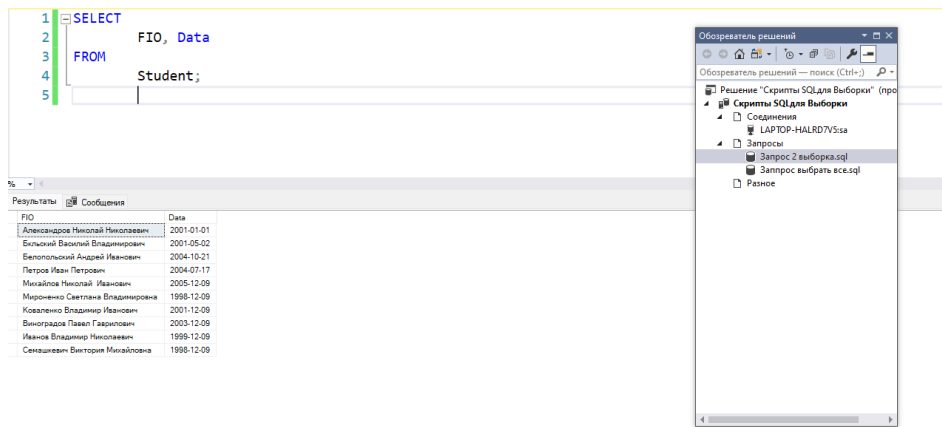


Получим:

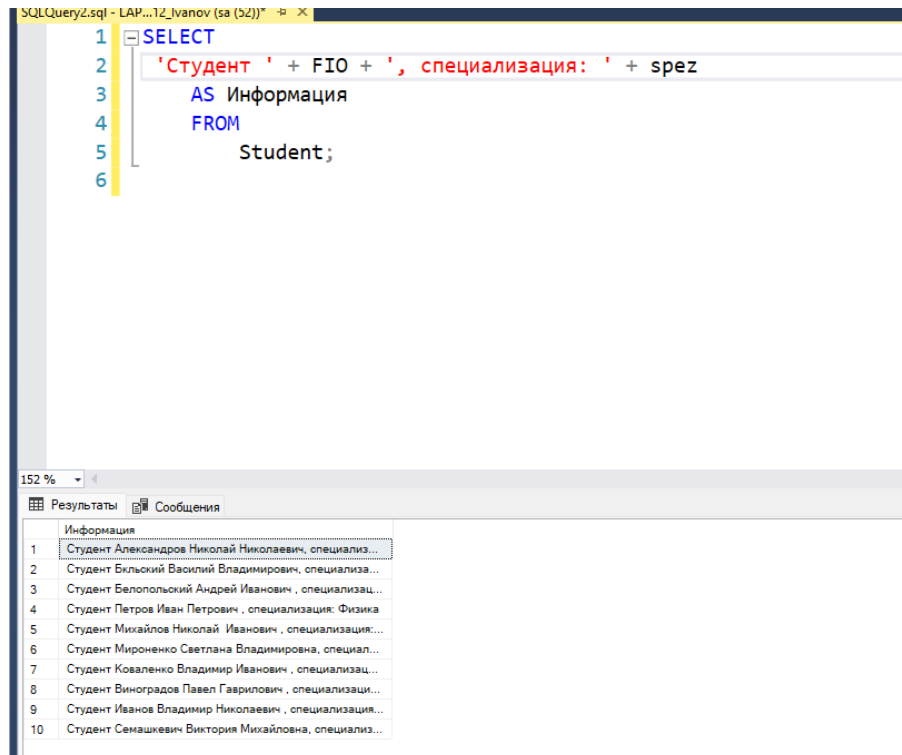


Вывести ФИО и дату рождения всех студентов:

Выполним и сохраним запрос под именем «Запрос 2 выборка» получим:



**Создайте вычисляемое поле «Информация», содержащее информацию об студентах в таком виде: «Студент Петров Петр Петрович, специализация: математика»:**



## Вывести ФИО студентов и номер следующего года после поступления:

Запрос 4 вычисления...2 Ivanov (sa (53))

```
1 SELECT FIO,  
2 [Через год] = godpost + 1  
3 FROM  
4 Student;  
5
```

Обозреватель решений

Обозреватель решений — поиск (Ctrl+;)

Решение "Скрипты SQL для Выборки" (пр...

- Скрипты SQL для Выборки
  - Соединения
    - LAPTOP-HALRD7V5:sa
  - Запросы
    - Запрос 2 выборка.sql
    - Запрос выбрать все.sql
    - Запрос 3 выч\_поле.sql
    - Запрос 4 вычисления.sql
  - Разное

52 %

Результаты Сообщения

	ФИО	Через год
1	Александров Николай Николаевич	2019
2	Бельский Василий Владимирович	2020
3	Белопольский Андрей Иванович	2023
4	Петров Иван Петрович	2023
5	Михайлов Николай Иванович	2024
6	Мироненко Светлана Владимировна	2017
7	Коваленко Владимир Иванович	2020
8	Виноградов Павел Георгиевич	2022
9	Иванов Владимир Николаевич	2019
10	Семашев Виктор Михайлович	2018

## Выведите список специализаций (spez), убрав дубликаты

Запрос 5 дубликат...12 Ivanov (sa (53))

```
1 SELECT DISTINCT  
2 spez FROM  
3 Student;  
4
```

Обозреватель решений

Обозреватель решений — поиск (Ctrl+;)

Решение "Скрипты SQL для Выборки" (пр...

- Скрипты SQL для Выборки
  - Соединения
    - LAPTOP-HALRD7V5:sa
  - Запросы
    - Запрос 2 выборка.sql
    - Запрос выбрать все.sql
    - Запрос 3 выч\_поле.sql
    - Запрос 4 вычисления.sql
    - Запрос 5 дубликат.sql
  - Разное

152 %

Результаты Сообщения

	spez
1	Информатика
2	математика
3	Радиофизика
4	Физика
5	Физика
6	Химия

Вывести список студентов, отсортированный в обратном алфавитном порядке по полю «Spez» и в алфавитном порядке по полю «FIO»

The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

```
1 SELECT *
2 FROM Student
3 ORDER BY
4     spez DESC, fio ASC;
5
```

The results pane shows a table with the following data:

FIO	Data	spez	godpost
Виноградов Павел Гаврилович	2003-12-09	Химия	2021
Семашкевич Виктория Михайловна	1998-12-09	Химия	2017
Александров Николай Николаевич	2001-01-01	Физика	2018
Петров Иван Петрович	2004-07-17	Физика	2022
Иванов Владимир Николаевич	1999-12-09	Радиофизика	2018
Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016
Бильский Василий Владимирович	2001-05-02	математика	2019
Михайлов Николай Иванович	2005-12-09	математика	2023
Белопольский Андрей Иванович	2004-10-21	Информатика	2022
Коваленко Владимир Иванович	2001-12-09	Информатика	2019

The right pane shows the 'Обозреватель решений' (Solution Explorer) window, displaying the 'Скрипты SQL для Выборки' (SQL Scripts for Selection) folder, which contains various query files.

Вывести первые две строки из списка студентов, отсортированного в алфавитном порядке по полю «ФИО»

The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL query:

```
1 SELECT TOP 2 *
2 FROM Student
3 ORDER BY
4     fio ASC
5
```

The results pane shows a table with the following data:

FIO	Data	spez	godpost
Александров Николай Николаевич	2001-01-01	Физика	2018
Белопольский Андрей Иванович	2004-10-21	Информатика	2022

The right pane shows the 'Обозреватель решений' (Solution Explorer) window, displaying the 'Скрипты SQL для Выборки' (SQL Scripts for Selection) folder, which contains various query files.

## Фильтрация данных.

### Использование в запросах операторов сравнения, логических операторов

Для фильтрации данных применяется оператор **WHERE**.

**Синтаксис:**

**WHERE <условие>. В**

условиях поле таблицы сравнивается с константой или с выражением. Символьные константы пишутся в одинарных кавычках. Числовые константы и названия столбцов пишутся без кавычек.

Чтобы строка попала в результат, условие должно быть истинно. В условиях используются операции сравнения.

**В Transact-SQL применяются следующие операции сравнения:**

**= – равенство;**

**<> или != – неравенство;**

**< – меньше;**

**> – больше;**

**!< – не меньше;**

**!> – не больше;**

**<= – меньше или равно;**

**>= – больше или равно.**

Можно использовать несколько условий для фильтрации данных. Для объединения их в одно выражение используются логические операторы.

В Transact-SQL применяются следующие логические операторы:

**AND** – логическое умножение или конъюнкция (И). Бинарный оператор, объединяет два условия; если оба условия истинны, результат – истина, иначе – ложь.

**OR** – логическое сложение или дизъюнкция (ИЛИ). Бинарный оператор, объединяет два условия; если хотя бы одно из этих условий истинно, то общее условие оператора **OR** также будет истинно.

**NOT** – логическое отрицание или инверсия (НЕ). Унарный оператор, применяется к одному условию. Если выражение в этой операции ложно, то общее условие истинно.

Самый высокий приоритет у оператора **NOT**. Самый низкий – у **OR**. Если эти операторы встречаются в одном выражении, **то сначала выполняется NOT, потом AND, а затем OR.**

При записи условий использование скобок – хороший тон программирования.

Оператор **BETWEEN** используется для сравнения с **диапазоном от начального и до конечного значения**. Начальное и конечное значения включены в промежуток.

Оператор **LIKE** используется для **сравнения с шаблоном строки**. Для определения шаблона применяются специальные символы:

**% – любая последовательность символов, в том числе пустая;**

**\_ – любой символ;**

**[ ] – символ, который указан в квадратных скобках;**

**[ - ] – символ из определенного диапазона;**

**[ ^ ] – символ, который не указан после символа ^.**

**В базах данных, для обозначения неизвестного значения, используется понятие NULL.**

Для проверки неизвестного значения нельзя использовать операторы сравнения. Допускается только **IS NULL** или **IS NOT NULL**.

Оператор **IN** используется для сравнения с набором значений. Список значений указывается в скобках.

## Воспользуемся данными из табл. Student

APTOP-HALRD7V5....nov - dbo.Student

	FIO	Data	spez	godpost
	Александров Николай Николаевич	2001-01-01	Физика	2018
✎	Бельский Василий Владимирович	2001-05-02	математика	2019
	Белопольский Андрей Иванович	2004-10-21	Информатика	2022
	Петров Иван Петрович	2004-07-17	Физика	2022
	Михайлов Николай Иванович	2005-12-09	математика	2023
	Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016
	Коваленко Владимир Иванович	2001-12-09	Информатика	2019
	Виноградов Павел Гаврилович	2003-12-09	Химия	2021
	Иванов Владимир Николаевич	1999-12-09	Радиофизика	2018
	Семашкевич Виктория Михайловна	1998-12-09	Химия	2017
*	NULL	NULL	NULL	NULL

## Задания

1. Вывести список студентов, информацию о студентах, которые поступили после 2018 года.

```
1  --присоединение БД
2  use Ucheb_12_Ivanov
3  --список студентов, которые поступили после
4  SELECT fio,
5         data,
6         spez,
7         godpost
8  FROM   Student
9
10 WHERE  godpost > 2018
11
12
```

2 %

Результаты

fio	data	spez	godpost
Бельский Василий Владимирович	2001-05-02	математика	2019
Белопольский Андрей Иванович	2004-10-21	Информатика	2022
Петров Иван Петрович	2004-07-17	Физика	2022
Михайлов Николай Иванович	2005-12-09	математика	2023
Коваленко Владимир Иванович	2001-12-09	Информатика	2019
Виноградов Павел Гаврилович	2003-12-09	Химия	2021

2. Вывести список студентов, информацию о студентах специальность у которых «физика»

```
12 --Вывести список студентов, специальность у которых «физика»
13 use Ucheb_12_Ivanov;
14 SELECT fio,
15        data,
16        spez,
17        godpost
18 FROM   Student
19
20 WHERE  spez='физика';
21
```

Результаты

fio	data	spez	godpost
Александров Николай Николаевич	2001-01-01	физика	2018
Петров Иван Петрович	2004-07-17	физика	2022



3. Вывести список студентов, информацию о студентах которые поступили после 2016 года по специальности математика или студентов, которые поступили после 2018 года по специальности физика.

```
40 --список студентов, которые поступили после 2016 года по специальности математика
41 --или которые поступили после 2018 года по специальности физика
42 use Ucheb_12_Ivanov;
43 SELECT fio,
44        data,
45        spez,
46        godpost
47 FROM Student
48 WHERE (godpost>2016) AND (spez='математика')
49        OR
50        (godpost>2018) AND (spez='физика')
51
52
53
54
```

фio	data	spez	godpost
Бельский Василий Владимирович	2001-05-02	математика	2019
Петров Иван Петрович	2004-07-17	физика	2022
Михайлов Николай Иванович	2005-12-09	математика	2023

4. Вывести список студентов, информацию о студентах которые поступили в промежутке 2019-2022гг.

```
54 -- список студентов, которые поступили в промежутке 2019-2022гг.
55 use Ucheb_12_Ivanov;
56 SELECT *
57 FROM Student
58 WHERE godpost BETWEEN 2019 AND 2022
59
60
61
62
```

ФИО	Data	spez	godpost
Бельский Василий Владимирович	2001-05-02	математика	2019
Белопольский Андрей Иванович	2004-10-21	Информатика	2022
Петров Иван Петрович	2004-07-17	физика	2022
Коваленко Владимир Иванович	2001-12-09	Информатика	2019
Виноградов Павел Гаврилович	2003-12-09	Химия	2021

5. Список студентов, информацию о студентах по дате рождения в промежутке 2001-01-01 и 2005-12-09 отсортированных по фамилии

```
72 -- список студентов, по дате рождения в промежутке 2001-01-01 и 2005-12-09
73 --отсортированных по фамилии
74 use Ucheb_12_Ivanov;
75 SELECT *
76 FROM Student
77 WHERE data BETWEEN '2001-01-01' AND '2005-12-09'
78 Order By
79 Fio;
80
81
82
83
```

ФИО	Data	spez	godpost
Александров Николай Николаевич	2001-01-01	физика	2018
Белопольский Андрей Иванович	2004-10-21	Инфо...	2022
Бельский Василий Владимирович	2001-05-02	мате...	2019
Виноградов Павел Гаврилович	2003-12-09	Химия	2021
Коваленко Владимир Иванович	2001-12-09	Инфо...	2019
Михайлов Николай Иванович	2005-12-09	мате...	2023
Петров Иван Петрович	2004-07-17	физика	2022

6. Список студентов, информацию о студентах фамилия, которых начинается с буквы «М»

```
83 -- Список студентов, информацию о студентах фамилия, которых начинается с буквы «М»
84 SELECT *
85
86 FROM
87     Student
88 WHERE
89     FIO LIKE 'М%';
```

2 %

Результаты Сообщения

	FIO	Data	spez	godpost
1	Михайлов Николай Иванович	2005-12-09	математика	2023
?	Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016

7. Список студентов, информацию о студентах, в специальности у которых вторая буква – «и»

```
91 --Список студентов, информацию о студентах в специальности у которых вторая буква – «и»
92     use Ucheb_12_Ivanov;
93 SELECT *
94
95 FROM
96     Student
97 WHERE
98     spez LIKE '_и%';
```

%

Результаты Сообщения

	FIO	Data	spez	godpost
	Александров Николай Николаевич	2001-01-01	физика	2018
	Петров Иван Петрович	2004-07-17	физика	2022
	Виноградов Павел Гаврилович	2003-12-09	Химия	2021
	Семашкевич Виктория Михайловна	1998-12-09	Химия	2017

8. Список студентов, информацию о студентах, в фамилии у которых третья буква буква «л» или «е» или «м»

```
99 --Список студентов, информацию о студентах, в фамилии у которых
100 --третья буква буква «л» или «е» или «м»
101     use Ucheb_12_Ivanov;
102 SELECT *
103
104 FROM
105     Student
106 WHERE
107     FIO LIKE '___[лем]%' ;
108
```

Результаты Сообщения

	FIO	Data	spez	godpost
	Александров Николай Николаевич	2001-01-01	физика	2018
	Бельский Василий Владимирович	2001-05-02	математика	2019
	Белопольский Андрей Иванович	2004-10-21	Информатика	2022
	Семашкевич Виктория Михайловна	1998-12-09	Химия	2017

9. Фамилии студентов, фамилии, у которых не начинаются с «М» до «С» или с

```
117 -- Фамилии студентов, фамилии которых не начинаются с «М» до «С» или с «Б»
118 use Ucheb_12_Ivanov;
119 SELECT fio
120
121 FROM
122     Student
123 WHERE
124     Fio LIKE '[^М-С,^Б]%' ;
```

152 %

Результаты Сообщения

	fo
1	Александр Николай Николаевич
2	Коваленко Владимир Иванович
3	Виноградов Павел Гаврилович
4	Иванов Владимир Николаевич

10. Список студентов, по специальностям математика, физика

```
126 -- список студентов, по специальностям математика, физика
127 use Ucheb_12_Ivanov;
128 SELECT
129     *
130
131 FROM
132     Student
133 WHERE
134     spez In ('математика', 'физика');
```

%

Результаты Сообщения

FIO	Data	spez	godpost
Александр Николай Николаевич	2001-01-01	физика	2018
Бельский Василий Владимирович	2001-05-02	математика	2019
Петров Иван Петрович	2004-07-17	физика	2022
Михайлов Николай Иванович	2005-12-09	математика	2023

