

## Практическая работа № 6 Создание *SQL*-запросов

**Цель работы:** научиться создавать запросы на основе языка *SQL* к БД.

### Этапы работы

- Изучить основные операторы специального языка *SQL*-запросов.
- Создать запросы: на выборку; на удаление, добавление и редактирование записей; с вычисляемыми полями; на создание таблицы; создание вычисляемого и подчиненного запроса.

### Теоретические сведения

Что делать, если нам нужны студенты, которые учатся в 10 классе? Да, это просто, если в нашей БД только от 6 до 20 человек, но что, если их будет 6000? Без дополнительных инструментов узнать это будет сложно.

Именно в этой ситуации к нам на помощь приходят *SQL* запросы, которые помогают изъять лишь необходимую информацию.

#### Виды запросов

*SQL* синтаксис реализует принцип **CRUD** (сокр. от англ. **Create, Read, Update, Delete** — «создать, прочесть, обновить, удалить»). Т.е. с помощью запросов вы сможете реализовать все эти функции.

#### Запросы на выборку данных (оператор **SELECT**).

Оператор **SELECT** (выбрать) является одним из основных операторов, применяемых в *SQL*. С помощью него осуществляется выборка данных из таблиц базы данных. Формат *SQL*-запроса с оператором **SELECT** выглядит следующим образом:

```
SELECT [DISTINCT] <список атрибутов>  
FROM <список таблиц>  
[WHERE <условие выборки>]  
[ORDER BY <список атрибутов>]  
[GROUP BY <список атрибутов>]  
[HAVING <условие>]  
[UNION <выражение с оператором SELECT>]
```

В квадратных скобках [] указаны элементы, которые могут отсутствовать в запросе. После слова **SELECT** через запятую перечисляются наименования полей (список атрибутов), содержимое которых запрашивается.

Обязательным ключевым словом в предложении-запросе **SELECT** является слово **FROM** (из). За ключевым словом **FROM** указывается список разделенных запятыми имен таблиц, из которых извлекается информация.

Другие операторы *SQL* запросов будут рассмотрены при их использовании.

### Задание 1. Создание запроса на выборку

Например, нам нужно найти всех студентов, которые учатся в 10 классе. Для этого нужно перейти на вкладку **Запросы** в секции **База данных**, в секции **Задачи** выбрать создать запрос в режиме *SQL*.

В текстовое поле введите команду:

**SELECT \* FROM Список WHERE Класс= 10;**

где «**SELECT \***» означает, что выбираются все студенты, «**FROM Список**» - из какой таблицы, «**Класс=10**» - условие, которое обязательно должно выполняться. Обязательным ключевым словом в предложении-запросе **SELECT** является слово **FROM** (из). За ключевым словом **FROM** указывается список разделенных запятыми имен таблиц, из которых извлекается информация.

**SQL-Запрос ВСЕГДА должен оканчиваться точкой с запятой ; .**

Нажмите кнопку **Выполнить запрос**. Сохраните запрос под именем **Класс\_10**.

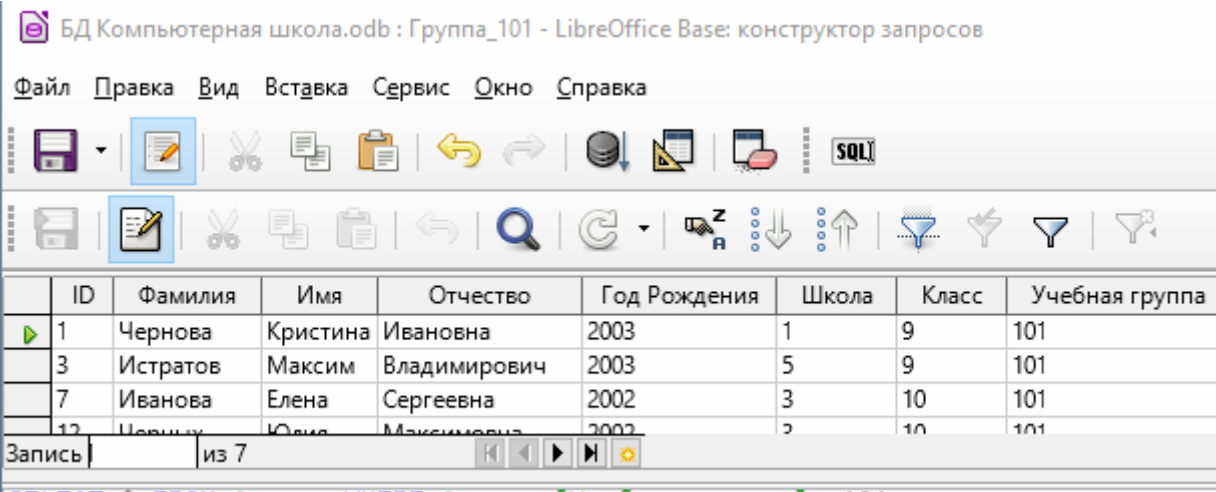
Если вы закрыли запрос и хотите редактировать уже существующий запрос БД, то для его открытия выделите нужный запрос и в контекстном меню выберите **Редактировать в режиме SQL**.

**Создайте запрос на выборку студентов, которые обучаются в 101 группе.** В текстовое поле введите команду:

**SELECT \* FROM Список WHERE Список.[Учебная группа]= 101;**

**Примечание.** Поля, содержащие недопустимые символы – пробел – заключаются в квадратные скобки, название таблиц, имена полей таблиц могут заключаться в кавычки, а сами записи таблиц для определенного поля должны быть заключены в апострофы, причем обязательно используется английский шрифт.

Нажмите кнопку **Выполнить запрос**. В таблице останутся только учащиеся 101 группы. Сохраните запрос под именем **Группа\_101**.



The screenshot shows the LibreOffice Base SQL query editor window titled "БД Компьютерная школа.odb : Группа\_101 - LibreOffice Base: конструктор запросов". The menu bar includes "Файл", "Правка", "Вид", "Вставка", "Сервис", "Окно", and "Справка". The toolbar contains various icons for file operations, editing, and database actions, with the "SQL" icon highlighted. Below the toolbar is a table with the following data:

	ID	Фамилия	Имя	Отчество	Год Рождения	Школа	Класс	Учебная группа
▶	1	Чернова	Кристина	Ивановна	2003	1	9	101
	3	Истратов	Максим	Владимирович	2003	5	9	101
	7	Иванова	Елена	Сергеевна	2002	3	10	101
	12	Черныш	Юлия	Максимовна	2002	2	10	101

Below the table, the "Запись" (Record) bar shows "из 7" (of 7). At the bottom, the SQL query is displayed: **SELECT \* FROM Список WHERE Список.[Учебная группа]= 101;**

А что делать, если нас интересуют студенты из группы 101, которые младше 2002 года рождения? Тогда измените запрос с помощью **AND** (логической операции «И»):

**SELECT \* FROM Список WHERE (((Список.[Учебная группа]) = 101) AND ((Список.[Год Рождения]) > 2002));**

Тогда результирующая таблица **Список** выдаст только учащихся 101 группы младше 2002 года рождения.

Сохраните запрос под тем же именем **Группа\_101**.

### **Выполните самостоятельное задание**

1. Создайте запрос на выборку студентов, которые обучаются в 102 группе компьютерной школы, и в 10 классе общеобразовательной школы.
2. Создайте запрос на выборку девятиклассников, обучающихся в 104 группе компьютерной школы.
3. Создайте запрос на выборку студенты из группы 103, которые старше 2003 года рождения.
4. Создайте запрос на выборку студенты из группы 103 или 105, используя **OR** – логическую операцию «ИЛИ».

Сохраните созданные запросы, выбирая им имена в соответствии с содержанием запросов.

**Примечание.** *Не все SQL-запросы являются запросами на выборку, т.е. используют оператор **SELECT**, тогда проблематично будет сохранить созданные запросы на добавление таблицы, добавление, редактирование и обновление записей таблицы и пр., поэтому создайте отдельный файл в *Writer* с именем **Запросы**, где сохраните текст всех таких последующих запросов с предложенными в работе именами. Для таких запросов используется команда меню **Сервис/SQL** из главного меню базы данных.*

### **Задание 2. Запрос на добавление таблицы**

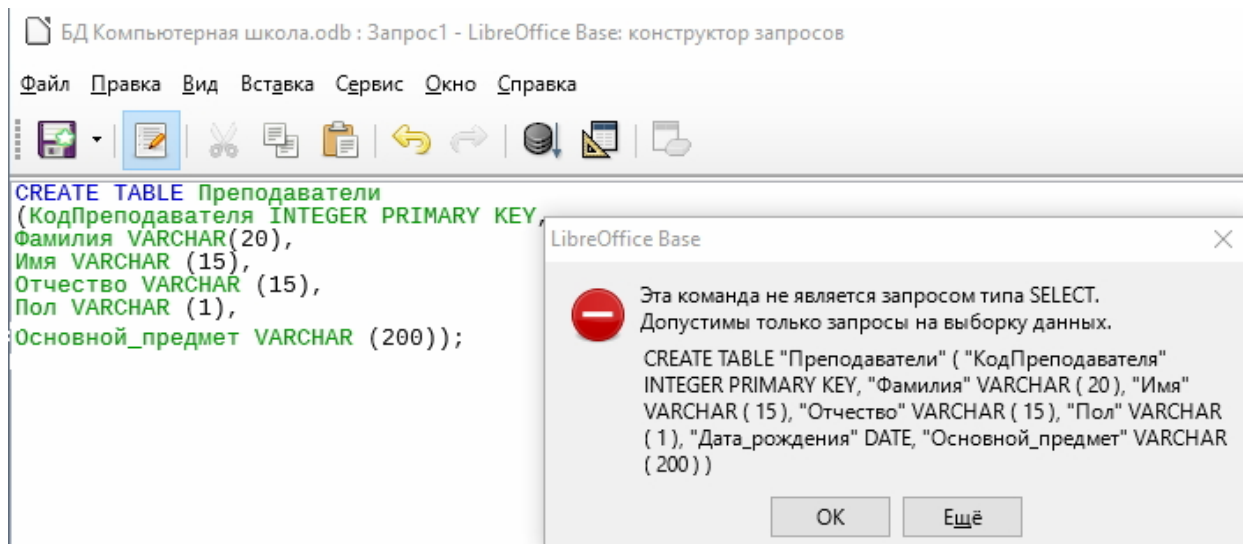
Кроме добавления таблицы с помощью встроенного конструктора, иногда может потребоваться выполнение этой операции с помощью *SQL* запроса. В большинстве случаев это нужно во время выполнения лабораторных или курсовых работ в рамках университетского курса, ведь в реальной жизни необходимости в этом нет. Если вы, конечно, не занимаетесь профессиональной разработкой приложений.

Создайте новый запрос в режиме *SQL*, наберите текст запроса:

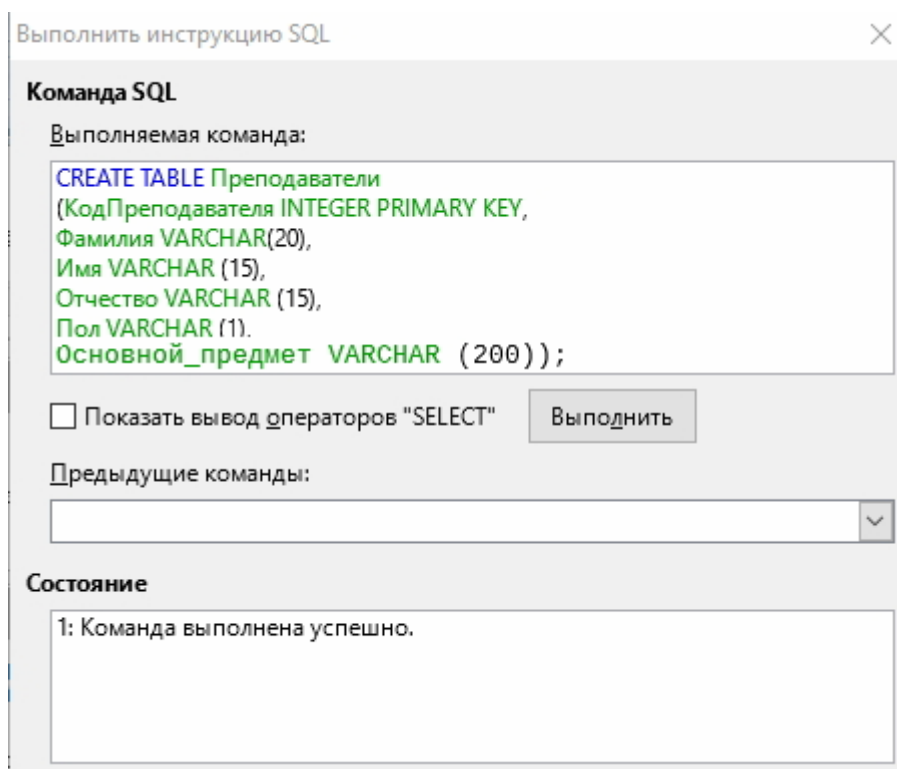
```
CREATE TABLE Преподаватели  
(Код INTEGER PRIMARY KEY,  
Фамилия VARCHAR(20),  
Имя VARCHAR (15),  
Отчество VARCHAR (15),  
Пол VARCHAR (1),  
Основной_предмет VARCHAR (200));
```

Команда «**PRIMARY KEY**» создает **Ключевое поле** в таблице «**Преподаватели**»; «**CREATE TABLE**» означает создание таблицы «**Преподаватели**», а **VARCHAR**, **INTEGER** - типы данных для соответствующих значений: **INTEGER** - означает тип данных поля - целое число, **VARCHAR** - символьный тип данных с количеством выделенных символов, т. е. **VARCHAR(15)** выделяет поле, которое должно содержать слово не более 15 букв.

Если сейчас как ранее в запросе на выборку **Выполнить запрос**, то увидите ошибку.



Для успешного выполнения запроса надо скопировать текст запроса в буфер обмена, в главном окне базы данных «Компьютерная школа» выбрать команду меню **Сервис/SQL**, в поле **Выполняемая команда** вставить из буфера обмена текст запроса и нажать кнопку **Выполнить**, после этого в поле **Состояние** появится сообщение об успешном выполнении запроса.



Нажмите кнопку **Заккрыть**. При попытке сохранить запрос опять появится ошибка, поэтому закройте запрос, сохраните его текст в отдельном файле Writer указав перед текстом имя запроса **Создание\_таблицы**. Если перейти на вкладку **Таблицы**, то вновь созданную таблицу вы не увидите. Для того чтобы увидеть созданную таблицу **Преподаватели**, нужно обновить данные, нажмите команду меню **Вид/ Обновить таблицы**. В результате вы увидите новую таблицу «Преподаватели» в БД, откройте её, вы увидите пустую таблицу.

### Задание 3. Запрос на добавление, удаление и редактирование записей в таблице

Снова перейдите в поле для создания запроса и введите следующие команды:

- для заполнения кортежа (апострофы и кавычки должны быть на английской раскладке):

**INSERT INTO "ПРЕПОДАВАТЕЛИ"**

**VALUES** (1, 'Иванова', 'Иванна', 'Ивановна', 'Ж', 'Информатика');

Сохраните запрос под именем **Запись1**.

- для редактирования записи создайте запрос:

**UPDATE "ПРЕПОДАВАТЕЛИ" SET "ИМЯ" = 'Анна';**

#### Выполните самостоятельное задание

Самостоятельно добавьте в таблицу **ПРЕПОДАВАТЕЛИ** три новых записи для преподавателей. Сохраните запросы **Запись2**, **Запись3**, **Запись4** как текст в созданном отдельном файле Writer.

**Добавьте новое поле в таблицу ПРЕПОДАВАТЕЛИ.** Создайте запрос на добавление поля с помощью команды **ALTER TABLE**. Она может добавлять столбцы к таблице, удалять столбцы, изменять их размеры, а также добавлять или удалять ограничения. Эта команда не часть стандарта ANSI, поэтому в разных системах она имеет разные возможности.

Типичный синтаксис чтобы добавить столбец к таблице:

**ALTER TABLE** <table name> **ADD** <column name> <data type> <size>;

Например:

**ALTER TABLE "ПРЕПОДАВАТЕЛИ" ADD "Часы" INT** – добавляет в таблицу **ПРЕПОДАВАТЕЛИ** новый столбец **Часы** (Количество отведенных преподавателем академических часов по дисциплине) целочисленного типа.

Посмотрите изменения в таблице **Преподаватели**. Сохраните запрос под именем **ПолеЧасы**.

Самостоятельно задайте запрос на добавление столбца **Оплата** (Оплата 1 академ. часа преподавателя в рублях) целочисленного типа. Сохраните запрос под именем **ПолеОплата**.

Убедитесь, что произошло добавление столбцов в таблицу **Преподаватели** и заполните их необходимыми данными.

- для удаления записей из таблицы создайте запрос:

**DELETE FROM "ПРЕПОДАВАТЕЛИ" WHERE "ИМЯ" = 'Анна';**

Убедитесь, что запись о преподавателе Ивановой Анне Ивановне удалена. Сохраните запрос под именем **Удаление\_записи** как текст в созданном отдельном файле Writer.

### Задание 4. Создайте вычисляемый запрос

Такие функции как **SUM** - сумма, **COUNT** - количество, **AVG** – среднее арифметическое значение, **MAX** – максимальное значение, **MIN** – минимальное

значение называются **агрегатными функциями**. Они принимают множество значений, и после их обработки возвращают единственное значение.

Например, для подсчета суммы произведения полей **Часы** и **Оплата** таблицы **Преподаватели** при помощи агрегатной функции **SUM** создайте запрос, выбирая в секции **Задачи** команду **Создание запроса в режиме SQL**:

```
SELECT SUM("Часы"*"Оплата") AS "Общая_Сумма"  
FROM "ПРЕПОДАВАТЕЛИ";
```

В результате вы получите только одно значение – **Общую сумму**. Выполните и сохраните запрос под именем **Общая\_сумма**.

	Общая_Сумма	
▶	16915	
Запись		из 1
<pre>SELECT SUM("Часы"*"Оплата") AS "Общая_Сумма" FROM "ПРЕПОДАВАТЕЛИ";</pre>		

Или можно подсчитать **Сумму** к выплате для какого-либо преподавателя, например, по его коду в таблице, если указать условие:

```
SELECT SUM("Часы"*"Оплата") AS "Выплата"  
FROM "ПРЕПОДАВАТЕЛИ"  
WHERE "КОД" = 3;
```

Выполните запрос и сохраните под именем **Выплата1**.

#### **Выполните самостоятельное задание**

Самостоятельно создайте запрос по Выплате одного из преподавателей, указав его *Фамилию* в условии. Выполните запрос и сохраните как **Выплата2**.

Возможно использование нескольких агрегатных функций в запросе. Например, подсчитаем среднюю оценку учащихся по каждому из предметов и определим количество учащихся, сдавших экзамены:

```
SELECT Avg("Writer") AS "AvgWriter", Avg("Calc") AS "AvgCalc",  
Avg("Base") AS "AvgBase", Count(*) AS "ЧислоСтудентов"  
FROM "Личные данные";
```

Выполните и сохраните запрос под именем **СредняяОценка**.

Файл Правка Вид Вставка Сервис Окно Справка				
	AvgWriter	AvgCalc	AvgBase	ЧислоСтудентов
▶	4	4	4	25
Запись из 1				
<pre>SELECT Avg("Writer") AS "AvgWriter", Avg("Calc") AS "AvgCalc", Avg("Base") AS "AvgBase", Count(*) AS "ЧислоСтудентов" FROM "Личные данные";</pre>				

### Выполните самостоятельное задание

1. Создайте запрос для подсчета количества пятерок по дисциплине Calc. Выполните и сохраните запрос под именем **Calc\_5**.
2. Попробуйте изменить запрос, чтобы средняя оценка рассчитывалась для конкретного учащегося по его коду. Выполните и сохраните запрос под именем **ОценкаУчащегося**.

### Задание 5. Создание запросов на объединение записей

Этот тип запроса позволяет объединить в одном результирующем наборе результаты нескольких запросов, таблиц и инструкций **SELECT**.

Создадим запрос на объединение данных, представленных в двух таблицах. Например, нам необходимо посмотреть оценки учащихся всех групп по текстовому редактору Writer. Тогда нам нужны две таблицы: **Список** (поля *Фамилия*, *Имя*, *Отчество*) и таблица **Личные данные** (поле *Writer*).

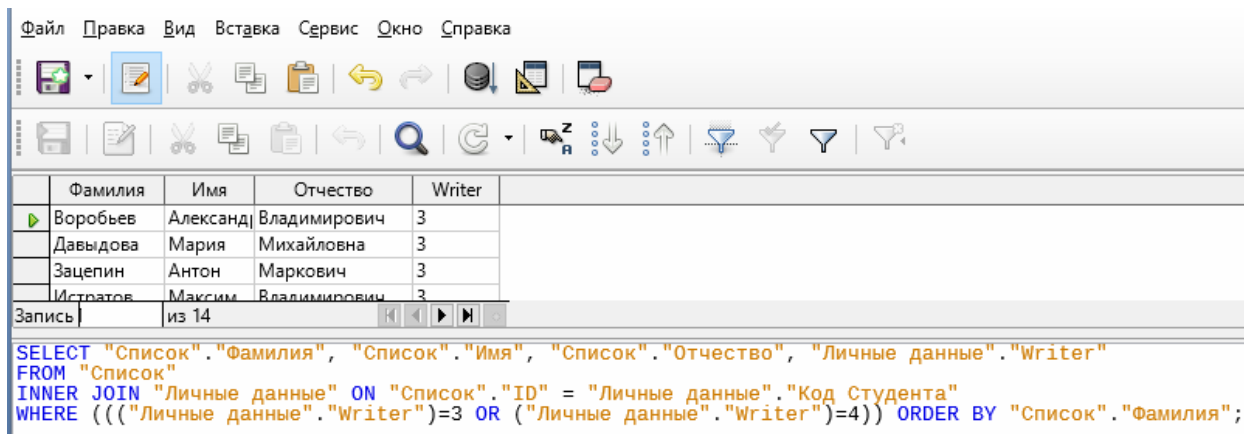
Создадим новый SQL-запрос, на вкладке **Запросы** в секции **Задачи** выберите **Создать запрос в режиме SQL**:

```
SELECT "Список"."Фамилия", "Список"."Имя", "Список"."Отчество", "Личные
данные"."Writer"
FROM "Список"
INNER JOIN "Личные данные"
ON "Список"."ID" = "Личные данные"."Код Студента"
```

Как можно видеть, поля таблицы представлены названием таблицы и названием поля, разделенные точкой. Если таблица или поле содержит запрещенные символы (пробел), то поле или таблица представляются в квадратных скобках. Кавычки нужно набирать на английской раскладке.

Команда **INNER JOIN** означает объединение данных из двух таблиц. Команда **ON** устанавливает связи между таблицами. Нажмите кнопку **Выполнить запрос**. Сохраните запрос с именем **ОценкиWriter**.

Добавим сортировку по полю *Фамилия* (по возрастанию). Команда **ORDER BY**. Выберем тех студентов, оценки которых 3 или 4. Команда **WHERE**. Выполните запрос.



### Выполните самостоятельное задание

1. Самостоятельно создайте запросы по учащимся, (по возрастанию поля *Отчество*) чьи оценки по **Calc** не более 3.
2. Самостоятельно создайте запрос по учащимся, установите сортировку по *Именам* (убыванию), которые получили оценки по **Base** только 5.
3. Сохраните запросы под именами **ОценкиCalc** и **ОценкиBase**.

Инструкция **ASC** используется как дополнение к инструкции **ORDER BY** и служит для определения сортировки по возрастанию. Инструкция **DESC** используется как дополнение к инструкции **ORDER BY** и служит для определения сортировки по убыванию. В случае, когда ни **ASC**, ни **DESC** не указаны, подразумевается наличие **ASC**. Общий вид команды: **ORDER BY** ИмяПоля **DESC**;

### Задание 6. Создайте подчиненный запрос

Некоторые подчиненные запросы можно использовать в перекрестных запросах как предикаты в предложении **WHERE**. Подчиненные запросы, применяемые для вывода результатов (в списке **SELECT**), нельзя применять в перекрестных запросах. В отличие от запросов на объединение, подчиненный запрос можно создать в режиме дизайна. В строке **Критерий** в качестве условия отбора следует указать, например, зарезервированное слово **IN**, а затем ввести инструкцию **SELECT**. А можно сначала создать в режиме дизайна подчиненный запрос, переключиться в режим **SQL**, скопировать получившуюся инструкцию **SQL** в буфер обмена, а затем создать главный запрос и скопировать в строку **Критерий** подчиненный запрос из буфера.

Создадим вложенный запрос, отображающий информацию об учащихся, которые учатся у определенного преподавателя (определим это по фамилии преподавателя).

Попробуем реализовать это через вложенный запрос. Например, мы знаем *Фамилию* преподавателя, но не знаем, в какой группе он ведет занятия. Для начала определим номер группы, если нам известна фамилия преподавателя, например, Белоусов А.И.. Создадим соответствующий **SQL**-запрос, команда **Создать запрос в режиме SQL**:

```
SELECT "Учебная группа" FROM "Группы"
WHERE "Преподаватели" = 'Белоусов А.И.';
```

Убедитесь, что запрос работает, и вы видите номер группы - **102**.

Вернитесь в **SQL**-запрос. Теперь добавим список учеников выбранной группы из таблицы **Список**. Полностью запрос будет иметь вид:



Файл Правка Вид Вставка Сервис Окно Справка


	ID	Фамилия	Имя	Отчество	Год Рождения	Школа	Класс	Учебная группа	Портрет
▶	6	Петрунина	Оксана	Леонидовна	2003	3	9	102	<OBJECT>
	13	Давыдова	Мария	Михайловна	2002	1	10	102	<OBJECT>
	14	Онучина	Евгения	Ивановна	2003	1	9	102	<OBJECT>
	22	Петренко	Артем	Валерьевич	2003	5	9	102	<OBJECT>

Запись 1 из 4

```
SELECT * FROM "Список"
WHERE "Учебная группа" =
(SELECT "Учебная группа" FROM "Группы"
WHERE "Преподаватели" = 'Белоусов А.И. ');
```

Сохраните запрос под именем **Белоусов**.

### Выполните самостоятельное задание

1. Создайте запрос для преподавателя **Новиковой Е.В.** Сохраните запрос. Список какой группы вы увидите при выполнении этого запроса?
2. Создайте запрос для преподавателя **Зачесовой Т.П.** Сохраните запрос. Список какой группы вы увидите при выполнении этого запроса?
3. Сохраните запросы, давая им имена по фамилии преподавателей.

Сохраните созданную вами БД «**Компьютерная школа**».