

# Лабораторная работа №8. Access

## Цель работы

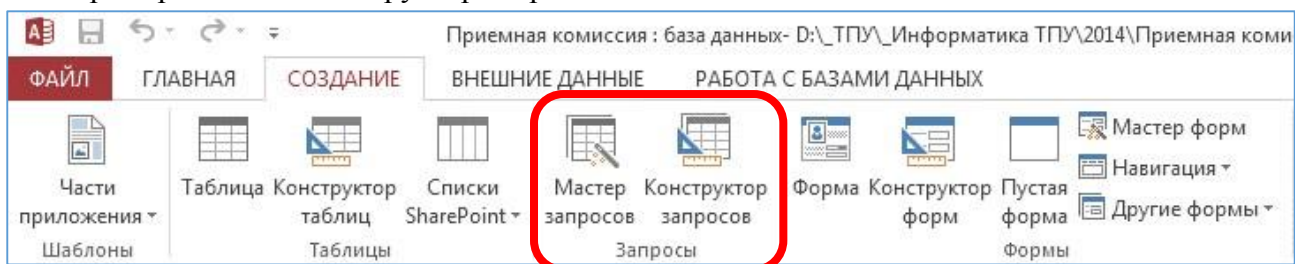
Освоение приемов работы с Microsoft Access, создание простых и сложных запросов.

## Работа с запросами

**Запрос** – это отбор записей в разнообразных формах, в соответствии с выбранными условиями. Запросы служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. Виды запросов:

- на выборку;
- запрос с параметром (критерий задаёт сам пользователь);
- итоговые запросы (производят вычисления по заданному полю и выдают результат);
- запросы на изменение (позволяют автоматизировать заполнение полей таблиц);
- перекрёстные запросы (позволяют создавать результирующие таблицы на основе результатов расчётов, полученных при анализе группы таблиц);
- специфические запросы – запросы к серверу БД, написанные на языке запросов SQL.

Для подготовки запроса используем вкладку **Создание** и выбираем способ создания запроса: Мастер запросов или Конструктор запросов.



## Запросы и фильтры

Запрос на выборку содержит условия отбора данных и возвращает выборку, соответствующую указанным условиям, без изменения возвращаемых данных. Фильтр также позволяет отбирать подмножество записей или сортировать их.

Сходство между запросами на выборку и фильтрами заключается в том, что и в тех и в других производится извлечение подмножества записей из базовой таблицы или запроса.

Основные отличия запросов и фильтров заключаются в следующем.

- Фильтры не позволяют в одной строке отображать данные из нескольких таблиц, т.е. объединять таблицы.
- Фильтры не дают возможности указывать поля, которые должны отображаться в результирующем наборе записей, они всегда отображают все поля базовой таблицы.
- Фильтры не могут быть сохранены как отдельный объект в окне базы данных (они сохраняются только в виде запроса).
- Фильтры не позволяют вычислять суммы, средние значения, подсчитывать количество записей и находить другие итоговые значения.

Запросы могут использоваться только с закрытой таблицей или запросом. Фильтры обычно применяются при работе в режиме **Формы** или в режиме **Таблицы** для просмотра или изменения подмножества записей.

## Задание

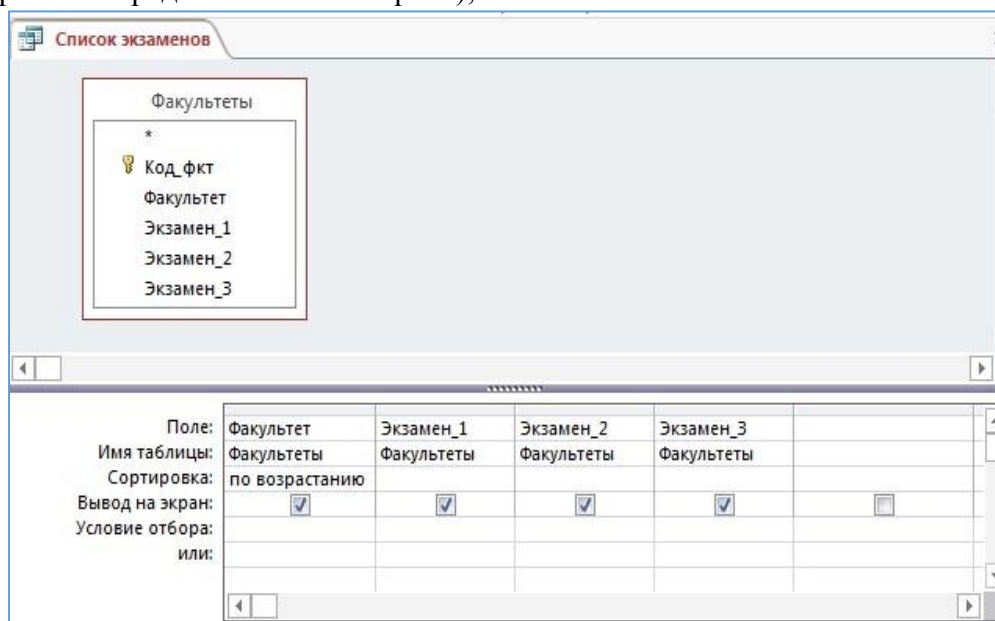
## Реализация простых запросов

**Запрос I.** Создать с помощью Конструктора запросов запрос к базе данных "Приемная комиссия", который позволяет получить список всех экзаменов на всех факультетах.

Список отсортировать в алфавитном порядке названий факультетов.

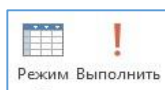
### Порядок выполнения

1. **Создание запроса.** Для этого запроса достаточно одной таблицы **Факультеты**.
  - на вкладке **Создание** выбрать **Конструктор запросов**;
  - в окне **Добавление таблицы** во вкладке **Таблицы** выбрать название таблицы **Факультеты**, щелкнуть на кнопках **Добавить** и **Заккрыть**;
  - сохранить запрос под именем "Список экзаменов";
  - заполнить бланк запроса, перетащив мышью нужные поля из верхней части бланка в нижнюю часть (или двойным щелчком на названии поля);
  - щелкнуть на строке **Сортировка** и в раскрывающемся списке выбрать направление сортировки; возможна многоуровневая сортировка (в этом случае поля сортируются в строгой очерёдности слева направо);

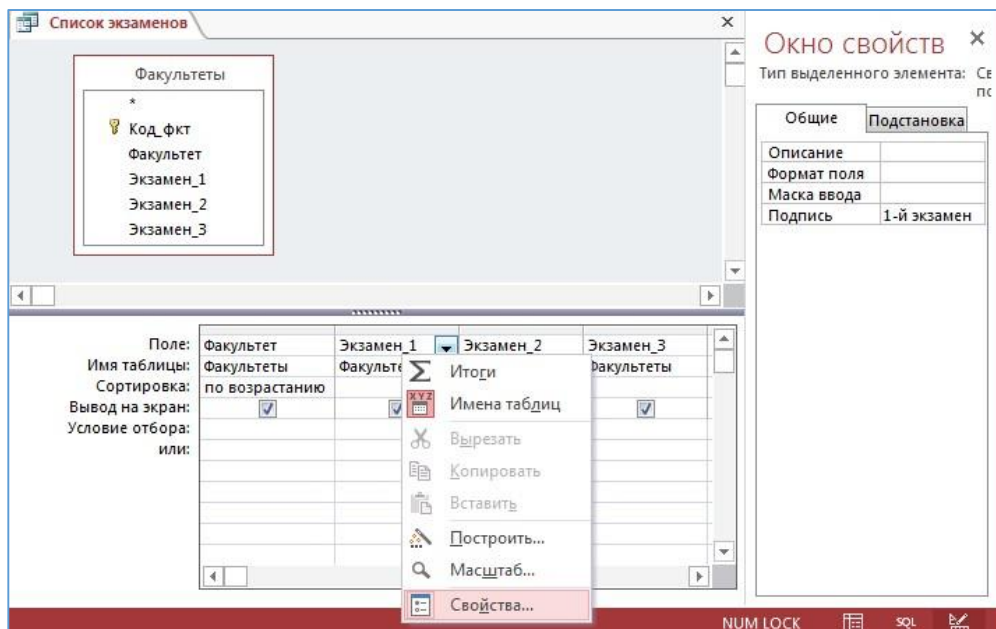


Для просмотра результата нажмите кнопку **Режим таблицы** или кнопку

**Выполнить**

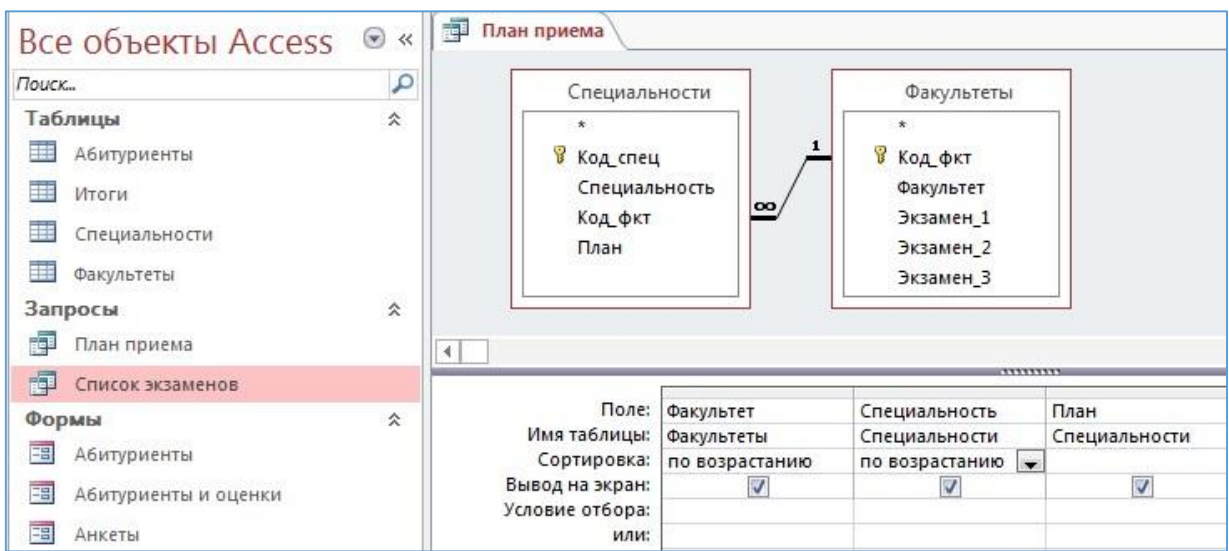


2. **Оформление запроса.** Сменим заголовки столбцов (полей) запроса, при этом имена полей в БД не изменятся.
  - В режиме конструктора запроса в контекстном меню поля **Экзамен\_1** выберите пункт **Свойства....**
    - В **Свойствах** поля в строке **Подпись** введите текст «1-й экзамен».



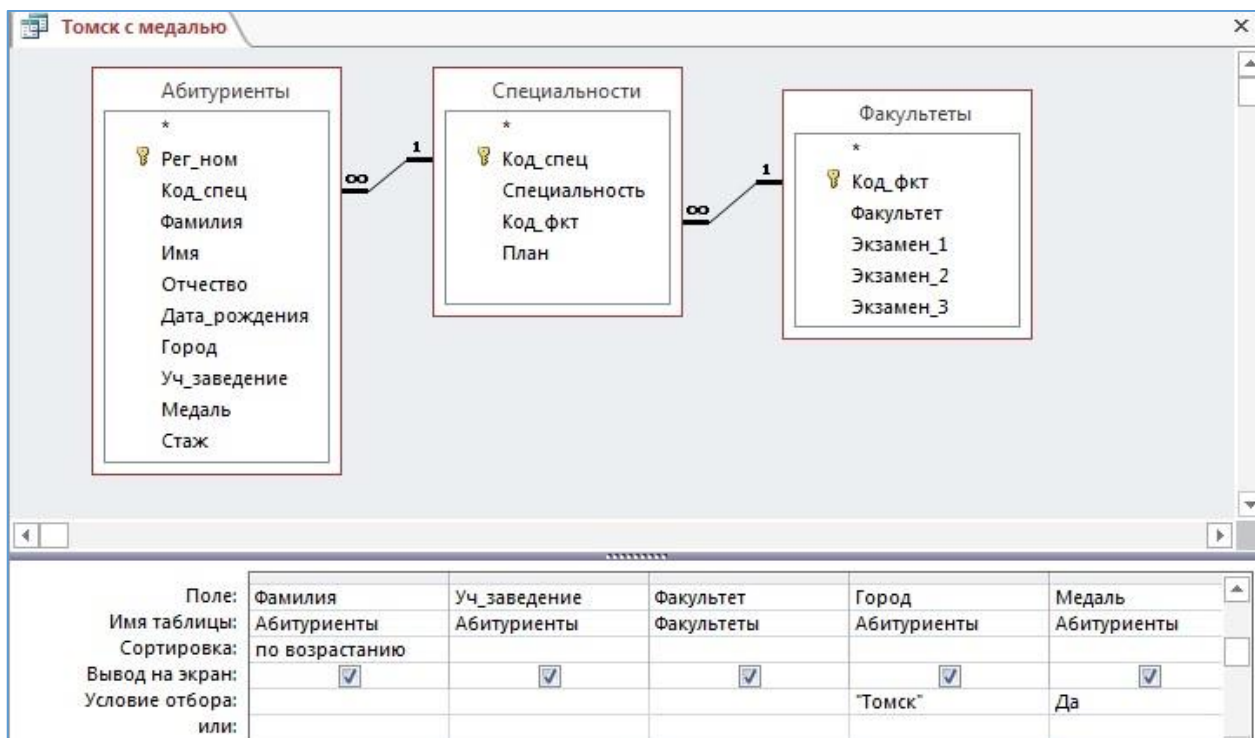
- Аналогично замените заголовки для других экзаменов.
- Сохраните и просмотрите результат выполнения запроса. Получим таблицу, которая отличается от предыдущей лишь заголовками. Обратите внимание, что заголовки меняются только в режиме просмотра, в конструкторе они остаются прежними.

**Запрос II.** Создать запрос *План приема*, который выводит список всех специальностей с указанием факультета и плана приема. Отсортировать список в алфавитном порядке по названию факультета (первый уровень сортировки) и по названию специальности (второй уровень).



Факультет	Специальность	План
Базового телекоммуникационного образования	Радиосвязь, радиовещание и телевидение	25
Базового телекоммуникационного образования	Радиотехника	25
Базового телекоммуникационного образования	Сети связи и системы коммутации	50
Заочное	Защищенные системы связи	10
Заочное	Связь с общественностью	10
Заочное	Экономика в телекоммуникации	10
Информационных систем и технологий	Информационные системы и технологии	15
Информационных систем и технологий	Программное обеспечение	25
Информационных систем и технологий	Управление инновациями	15
Телекоммуникации и радиотехники	Средства связи с подвижными объектами	10
Телекоммуникации и радиотехники	Физика и техника оптической связи	10

**Запрос III.** Получить список всех абитуриентов, живущих в Томске (можно выбрать любой город) и имеющих медали. В списке указать фамилию, номер школы и факультет, на который они поступают. Отсортировать список в алфавитном порядке фамилий.



Обратите внимание на то, что в запросе используются поля только из двух таблиц **Факультеты** и **Абитуриенты**, а в реализации запроса участвует таблица **Специальности**, т.к.

таблица Абитуриенты связана с таблицей Факультеты через таблицу Специальности.

### Самостоятельно:

**Запрос IV.** Получить список всех абитуриентов, поступающих в ВУЗ имеющих производственный стаж (больше 0). Указать фамилию, город, специальность, стаж и факультет на который поступают. Отсортировать фамилии по возрастанию.

**Запрос V.** Получить список абитуриентов, поступающих в ВУЗ имеющих производственный стаж и медаль. Указать фамилию, специальность и факультет, на который поступают. Отсортировать фамилии по возрастанию.

### Применение операторов *or* и *and*

В предыдущих запросах мы уже использовали неявно логическую операцию *and* (*и*), например, когда выбирали абитуриентов, проживающих в Томске **и** имеющих медаль. Обратите внимание на то, что эти условия в конструкторе запросов записываются в одной строку. Для того чтобы выбрать абитуриентов, либо проживающих в Томске, либо имеющих медаль, потребуется оператор *or* (*или*), и тогда одно из условий надо записать ниже, т.е. в строке **или**:

Поле:	Фамилия	Уч_заведение	Факультет	Город	Медаль
Имя таблицы:	Абитуриенты	Абитуриенты	Факультеты	Абитуриенты	Абитуриенты
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				"Томск"	
или:					Да

Для того чтобы выбрать абитуриентов, проживающих либо в Томске, либо в Северске можно поступить аналогично

Поле:	Фамилия	Уч_заведение	Факультет	Город	Медаль
Имя таблицы:	Абитуриенты	Абитуриенты	Факультеты	Абитуриенты	Абитуриенты
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				"Томск"	
или:				"Новосибирск"	

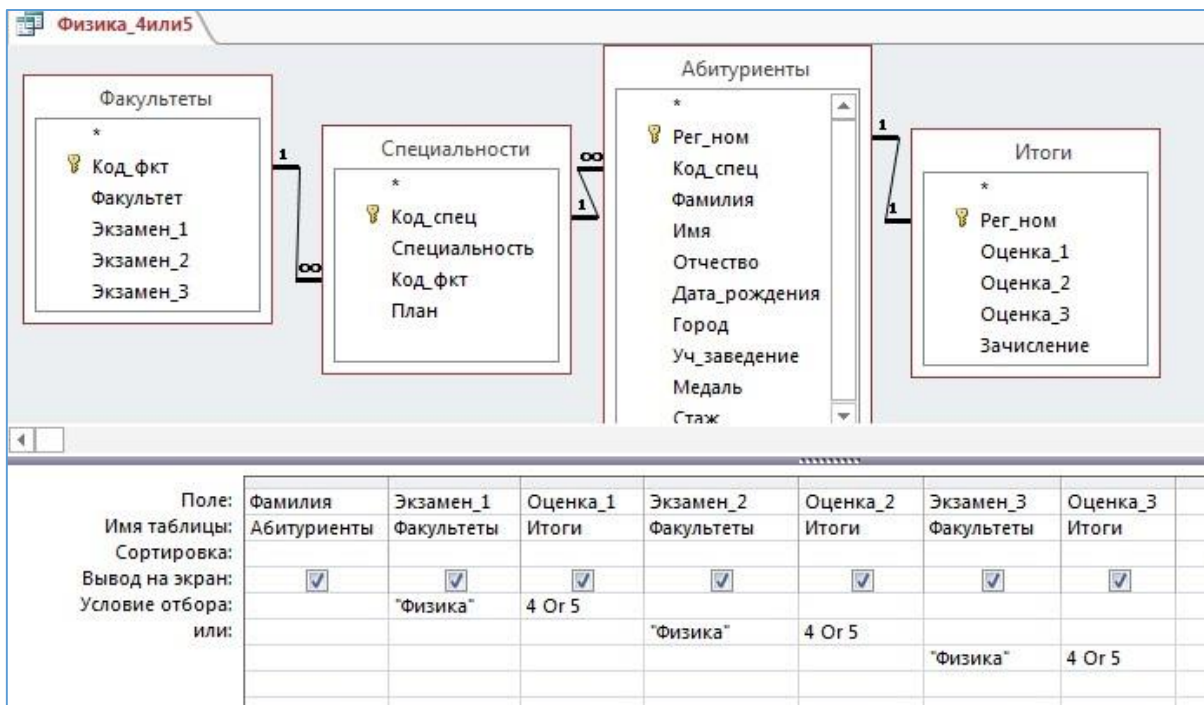
или записать операцию *or* в явном виде

Поле:	Фамилия	Уч_заведение	Факультет	Город	Медаль
Имя таблицы:	Абитуриенты	Абитуриенты	Факультеты	Абитуриенты	Абитуриенты
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				"Томск" Or "Новосибирск"	
или:					

**Запрос VI.** Получить список всех абитуриентов, сдавших физику с оценкой хорошо и отлично.

- В данном запросе следует применить оператор *or* для оценок. А для того чтобы учесть то, что физика могла быть либо первым, либо вторым, либо третьим экзаменом, воспользуемся строкой **или** в **условии отбора**:





**Самостоятельно:**

**Запрос VII.** Получите список абитуриентов, сдавших математику с оценкой хорошо и отлично по факультетам 01 и 03.

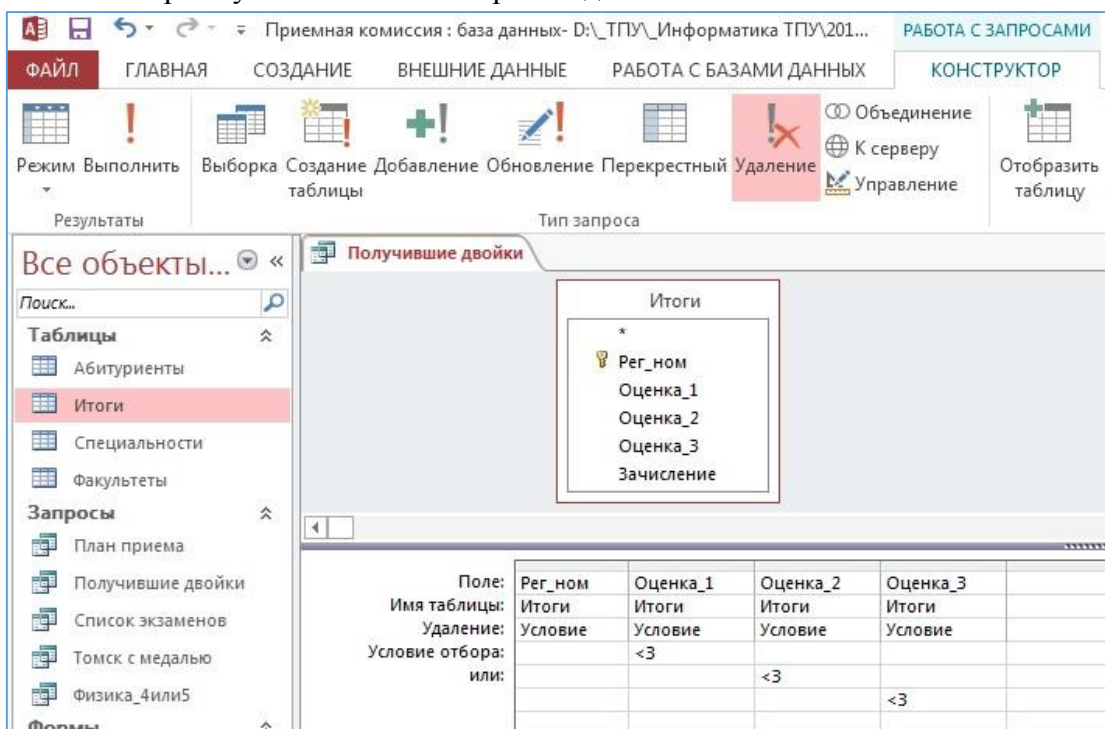
**Реализация запросов на удаление**

**Внимание! Ошибки в следующем запросе могут привести к удалению всех данных. Поэтому закройте базу данных, сделайте копию файла и затем вернитесь к выполнению лабораторной работы.**

**Запрос VIII.** Удалите из таблицы **Итоги** сведения об абитуриентах, получивших двойки или не явившихся на экзамены (в поле оценки значение 0 или пустое поле).

Алгоритм выполнения запроса.

- Создайте запрос, который позволяет выбрать из таблицы **Итоги** абитуриентов, получивших хотя бы по одному экзамену оценку меньше 3.
- В конструкторе запроса установите тип запроса **Удаление**.



- Нажмите кнопку **Выполнить**. Внимательно читайте сообщения!

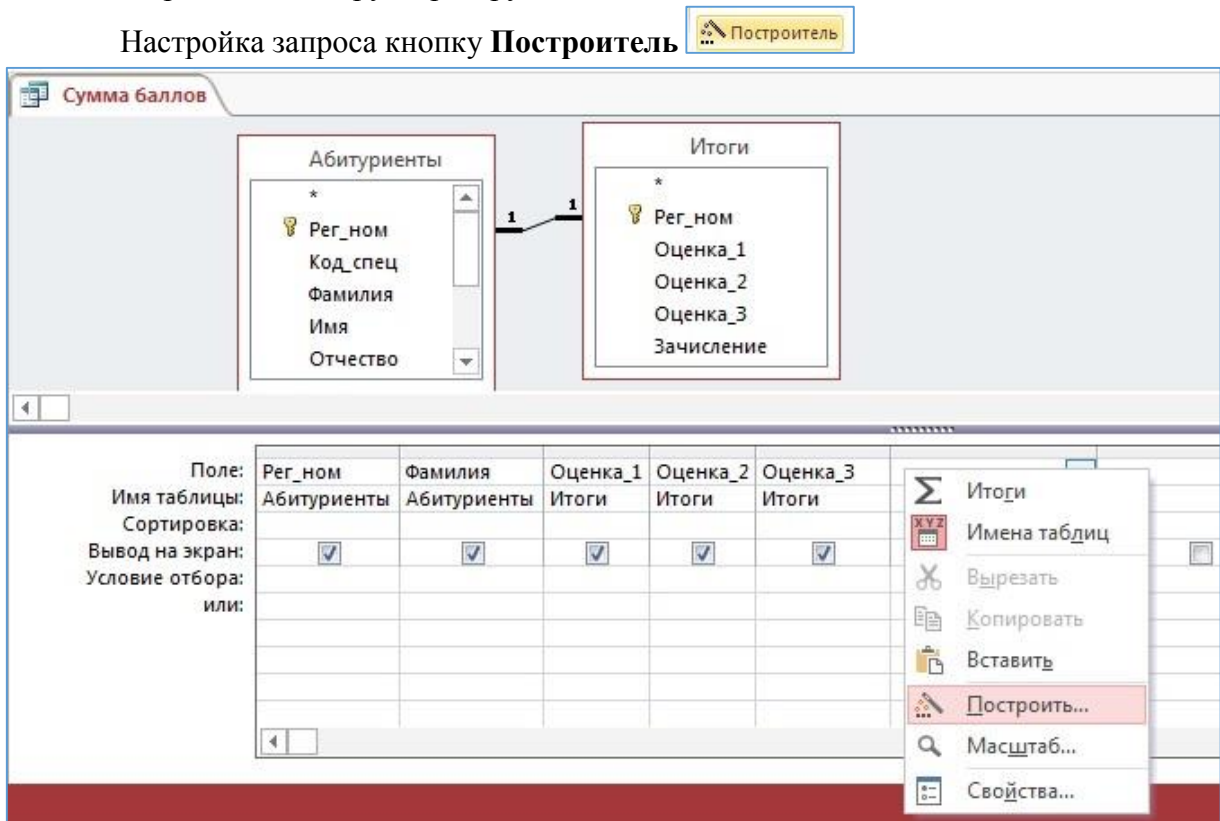
### **Использование вычисляемых полей**

**Запрос IX.** Выведите таблицу со значениями суммы баллов, включив в неё регистрационный номер, фамилию и сумму баллов. Отсортировать по убыванию суммы.

Алгоритм выполнения запроса.

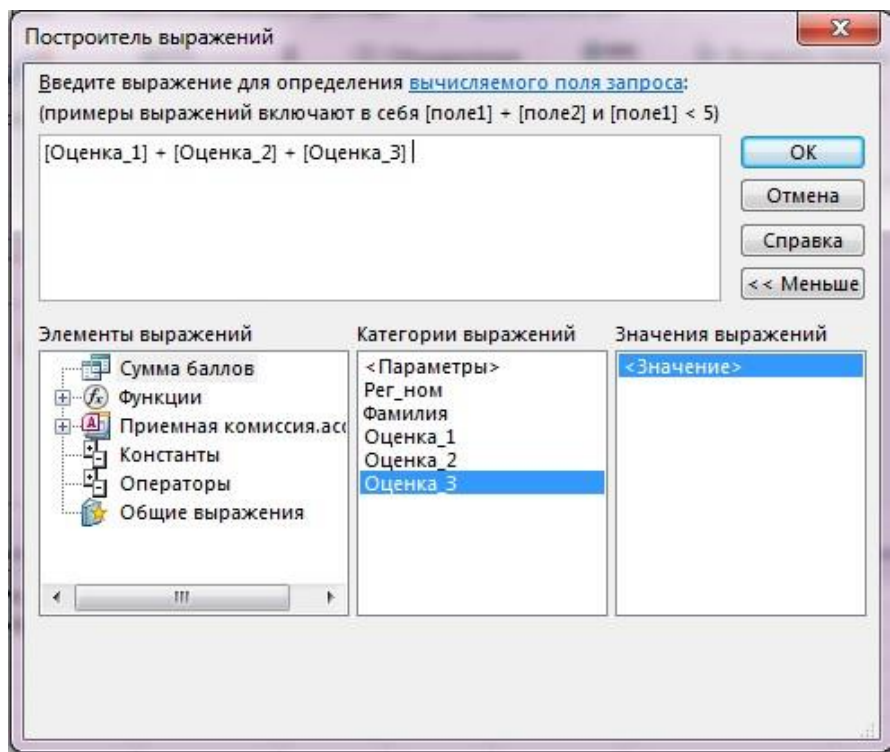
- Создайте запрос, который позволяет получить регистрационный номер, фамилию и все оценки.
- Щелкните в следующее за оценками поле и выберите команду **Построить** в контекстном меню или на вкладке Работа с запросами/Конструктор в группе

Настройка запроса кнопку **Построитель**



- В построителе выражений в области **Элементы выражений** найдите таблицу **Оценки**.
- Щелкая мышью на имена полей Оценка1 и т.д. сформируйте выражение

[Оценка\_1] + [Оценка\_2] + [Оценка\_3]



- Нажмите ОК.

- Переименуйте получившееся вычисленное поле **Выражение1** в **Сумма**:


Поле:	Рег_ном	Фамилия	Оценка_1	Оценка_2	Оценка_3	Сумма: [Оценка_1]+[Оценка_2]+[Оценка_3]
Имя таблицы:	Абитуриенты	Абитуриенты	Итоги	Итоги	Итоги	
Сортировка:						
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:						

- Выполните сортировку по убыванию суммы баллов.
- Запомните какой балл имеет пятый студент (пусть это будет проходной балл, который потребуется в запросе 13).

*Замечание.* Выражение можно вводить непосредственно в ячейке конструктора, не используя построитель выражений.

### Использование групповых операций

При обработке данных бывает необходимо учитывать в запросах итоги, которые вычисляются определенным образом для различных групп записей в таблице.

Для итоговых вычислений необходимо создать новый запрос в режиме конструктора, добавить в него необходимые таблицы, а затем воспользоваться кнопкой **Итоги** , которая находится на вкладке *Работа с запросами/Конструктор* в группе *Показать или Скрыть*. При этом в строке **Групповая операция** бланка запроса по умолчанию устанавливается значение **Группировка**.



При выполнении групповых операций можно использовать итоговые функции, которые следует выбирать из списка в добавленном поле **Групповые операции**. Основные итоговые функции:

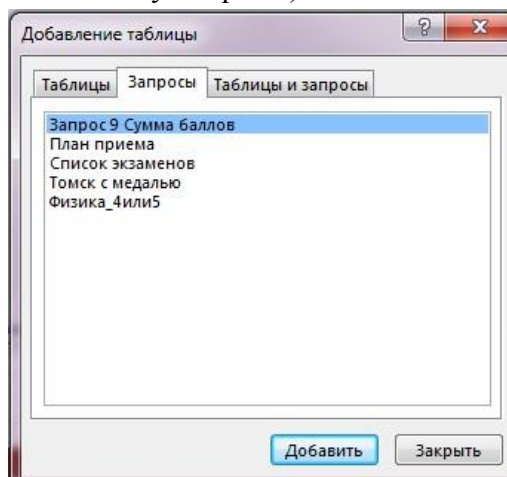
- **Sum** – суммирование числа значений в группе (в столбце),
- **Avg** – среднее значение для группы,
- **Min** – минимальное значение для группы,
- **Max** – максимальное значение для группы,
- **Count** – подсчет числа значений для группы,
- **First** – значение поля в первой записи группы,
- **Last** – значение поля в последней записи группы.



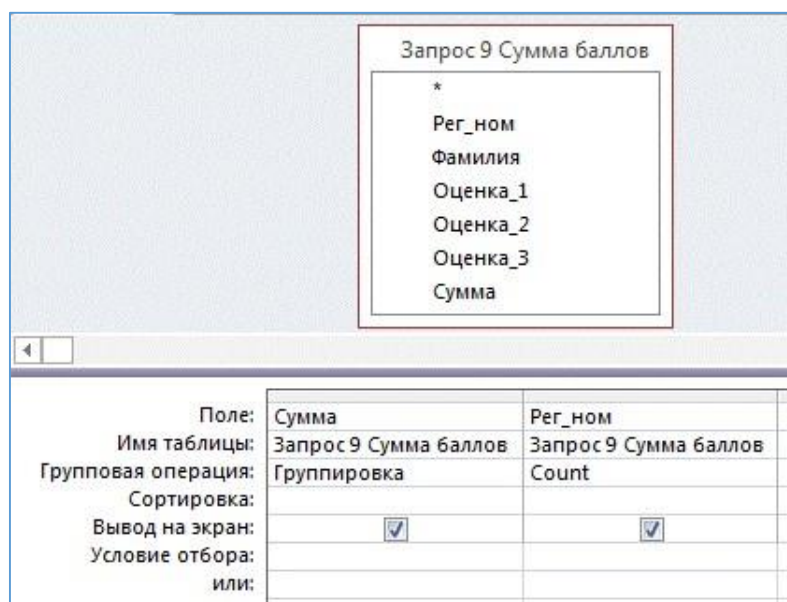
**Запрос X.** Сгруппировать абитуриентов по количеству набранных баллов. Сколько абитуриентов в каждой группе?

Для этого необходимо

- создать запрос на основе предыдущего запроса (при добавлении таблицы в запрос обратите внимание на вкладку Запросы)



- применить групповую операцию **Count**



- переименовать второй столбец

Сумма	Количество
9	2
11	1
12	2
13	1

**Самостоятельно:**

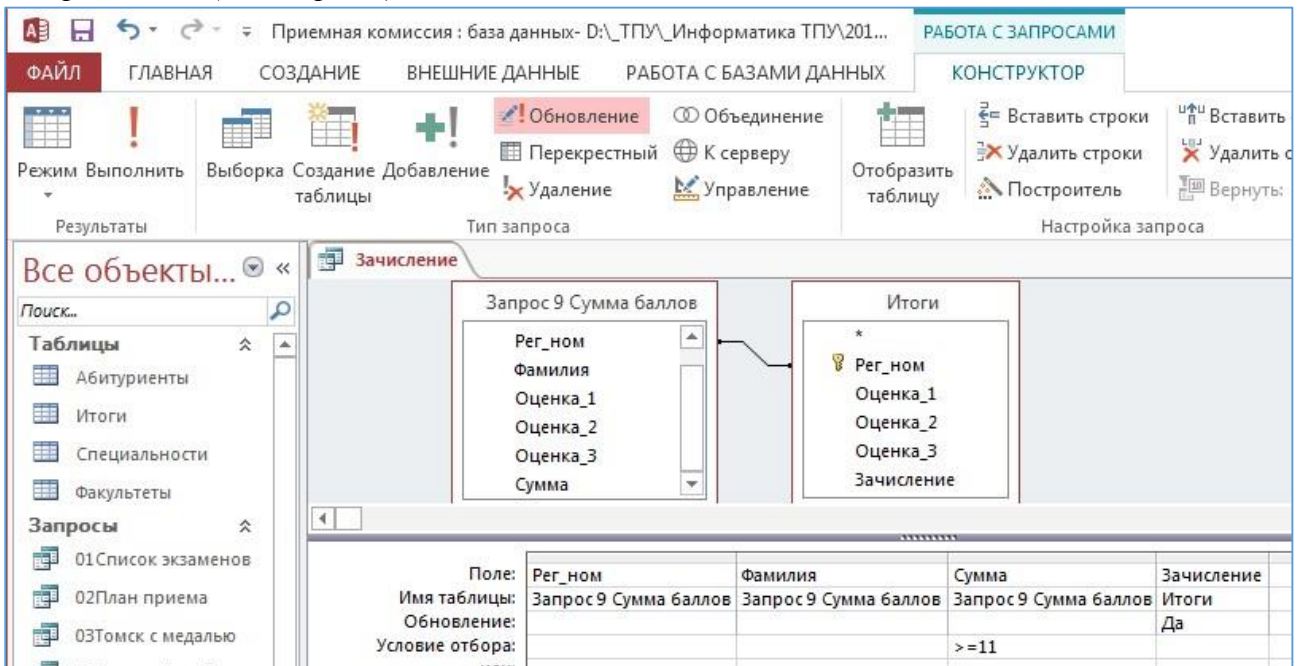
**Запрос XI.** Найдите среднюю сумму баллов.

**Запрос XII.** Найдите количество студентов, сдавших русский язык на 5.

### **Запрос на обновление**

Для создания запроса на обновление необходимо указать тип запроса Обновление (в режиме конструктора запросов на вкладке Работа с запросами/Конструктор в группе Тип запроса). Затем в строке Обновление в нижней части бланка указать нужное значение поля. При этом можно обновлять не все строки таблицы, а только удовлетворяющие некоторому условию, которое указывается в строке Условие отбора.

**Запрос XIII.** Создайте запрос, который позволяет автоматически заполнить столбец *Зачисление* таблицы *Итоги*: галочки поставим абитуриентам, набравшим балл не менее проходного (см. запрос 9).



### **Контрольные вопросы**

- Способы создания запросов.
- Виды запросов к БД MS Access.
- Простые запросы.
- Сложные запросы.
- Применение операторов "or", "and", "between".
- Запрос на удаление.
- Использование групповых операций. □ Использование вычисляемых полей.