

Федеральное агентство по образованию РФ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

С.Н. Авдеенко, А.Л. Богданов

**ПОСТРОЕНИЕ ЗАПРОСОВ
В СУБД MICROSOFT ACCESS**

Учебное пособие

Томск – 2009

ОДОБРЕНО кафедрой математических методов и информационных технологий в экономике

Зав. кафедрой, д.т.н., профессор _____ В.В. Домбровский

РАССМОТРЕНО И УТВЕРЖДЕНО методической комиссией экономического факультета

ПРОТОКОЛ № ____ от " ____ " _____ 2009 г.

Председатель комиссии, д.э.н., профессор _____ В.С. Цитленок

Учебное пособие посвящено вопросам построения запросов в СУБД Microsoft Access. Рассматриваются теоретические вопросы и подробно разбираются примеры по основным аспектам построения запросов различных видов: запросы на выборку записей, запросы с вычисляемыми полями, агрегированные запросы, многотабличные запросы, запросы на изменение.

Учебное пособие предназначено для студентов всех экономических специальностей и может применяться при изучении таких дисциплин как "Информационные технологии управления", "Информационные системы в экономике", "Автоматизированные технологии в экономике".

РЕЦЕНЗЕНТ:

к.т.н., доцент А.Н. Моисеев

Основные сведения о Microsoft Access

Основными функциями *систем управления базами данных* (СУБД) являются: описание структуры баз данных, обработка и управление данными. В настоящее время наибольшее распространение получили *реляционные СУБД* (РСУБД). В таких СУБД базы данных представляют собой множество взаимосвязанных таблиц. Каждая таблица хранит информацию об объектах одного определенного типа. Строки таблиц называются *записями*, столбцы – *атрибутами*. Каждая строка хранит информацию об одном объекте и состоит из полей, в которых хранятся значения атрибутов объекта.

СУБД Microsoft Access (далее просто Access) относится к классу *настольных РСУБД* и в настоящий момент является одним из самых популярных приложений данного класса. Первой отличительной особенностью Access является способ хранения данных – база данных представляет собой один файл, в котором хранятся таблицы с данными и другие объекты, облегчающие работу с ними: *запросы, экранные формы, отчеты, страницы доступа к данным, макросы и программы*. Второй особенностью Access является наличие множества инструментов, упрощающих работу с данными и другими объектами базы: мастера создания форм и отчетов, построители запросов, фильтры, средства проектирования структуры базы данных и анализа ее эффективности. Третьей особенностью является способность Access использовать данные из внешних источников: файлов, других СУБД и приложений. При этом использование внешних данных может осуществляться двумя способами: путем их импорта в базу данных Access и путем присоединения (если такое допускает источник). В последнем случае пользователь работает с присоединенными данными так, как будто бы эти данные находятся в базе данных Access, хотя фактически их расположение остается неизменным. Эта особенность позволяет организовать параллельную работу сразу нескольких пользователей с одними и теми же данными. Наконец, последней замечательной особенностью Access является наличие встроенного языка программирования Visual Basic for Application (VBA), с помощью которого пользователь при желании (и умении) может разработать собственные как простые, так и достаточно сложные многопользовательские приложения корпоративного уровня, работающие с данными серверных СУБД.

Построение запросов в Microsoft Access

Основные понятия

Запрос является стандартным объектом Access и предназначен для просмотра, анализа данных одной или нескольких таблиц, добавления, внесения изменений в имеющиеся данные или удаления данных. Например, с помощью запроса можно извлечь данные из нескольких таблиц, удовлетворяющие заданным условиям, разбить их на группы по определенному признаку, провести необходимые вычисления и сортировку в требуем порядке.

Запрос представляет собой команду, содержащую указание для СУБД какое действие с какими данными и как именно требуется выполнить. Передача СУБД команды, содержащейся в запросе, называется *выполнением запроса*. *Результат выполнения запроса* всегда представляет собой таблицу (даже, если результатом является одно число), поэтому запрос может выступать в роли *источника данных* для других запросов.

Создать запрос в Access можно двумя способами:

- написать текст запроса на структурированном языке запросов *SQL (Structured Query Language)*;
- построить образец таблицы, которую требуется получить в результате выполнения запроса, с помощью графического языка запросов *QBE (Query By Example)*.

В данном учебном пособии в основном рассматривается второй способ построения запросов. Инструмент, который будет использоваться для построения запросов, называется *Конструктор запросов* и подробно рассматривается далее в разделе "Запрос на выборку".

Запросы в Access бывают следующих типов:

- *запрос на выборку*,
- *перекрестный запрос*,
- *запрос на создание таблицы*,
- *запрос на обновление данных*,
- *запрос на добавление данных*,
- *запрос на удаление данных*,
- *запрос на объединение*.

Далее все типы запросов будут подробно рассмотрены.

Запрос на выборку

Запрос на выборку предназначен для просмотра данных, содержит условия отбора записей и возвращает выборку (*результатиру-*

ций набор), соответствующую указанным условиям, без изменения исходных данных. Это самый часто используемый тип запросов.

Запросы такого типа можно строить с помощью *фильтра*, *мастера построения запросов* и *конструктора запросов*. Во всех случаях производится извлечение заданного подмножества записей источника данных, отличие же заключается в следующем:

- *Фильтр* является простейшим инструментом извлечения данных, удовлетворяющих заданным условиям. Он не позволяет: осуществлять выбор данных сразу из нескольких таблиц, указывать какие поля должны присутствовать в результирующем наборе, вычислять итоговые значения. Кроме того, его нельзя сохранить в базе данных как самостоятельный объект (только в виде запроса).
- *Мастер построения запросов* позволяет быстро построить не очень сложный запрос, при этом допускается выбор полей из нескольких таблиц и вычисление итоговых значений (при определенных условиях).
- *Конструктор запросов* позволяет строить сложные запросы на основе нескольких таблиц, с возможностью выбора требуемых полей, наложением на них условий отбора, указанием порядка следования записей, группировкой, вычислением промежуточных итоговых значений и возможностью построения сложных выражений с применением встроенных в Access функций.

Рассмотрим каждый из инструментов более подробно.

Фильтрация данных

В Access предусмотрено несколько способов отбора записей с помощью фильтров:

- *Фильтрация по выбранному значению*: при просмотре содержимого таблицы выполняется щелчок правой кнопкой мыши по интересующему значению и в появившемся контекстном меню выбирается команда **Фильтр по выделенному**. В результате на экране остаются видимыми только записи, содержащие выбранное значение (рис. 1).
- *Фильтрация, исключающая выбранное значение*: при просмотре таблицы производится щелчок правой кнопкой мыши по интересующему значению, после чего в контекстном меню выбирается команда **Исключить выделенное**. В результате на экране остаются видимыми только записи, не содержащие выбранное значение (рис. 2).

- **Фильтрация по введенному значению:** при просмотре таблицы производится щелчок правой кнопкой мыши по любому значению интересующего столбца и в появившемся контекстном меню в поле **Фильтр для...** вводится требуемое значение или логическое выражение (условие отбора). В результате на экране остаются видимыми только записи, содержащие указанное значение (рис. 3).

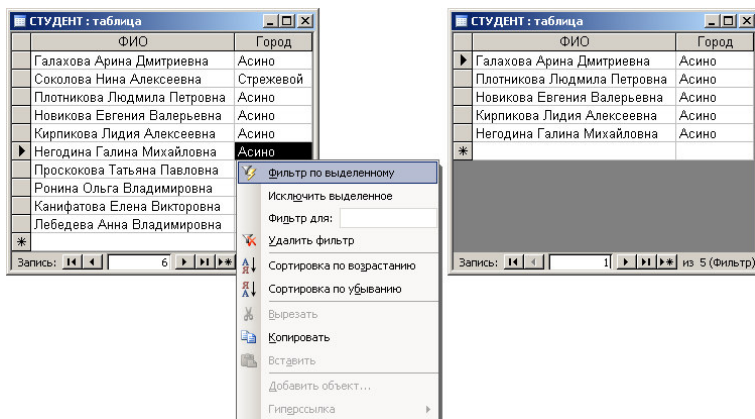


Рис. 1. Исходная таблица (слева)
и результат выполнения команды **Фильтр по выделенному** (справа)

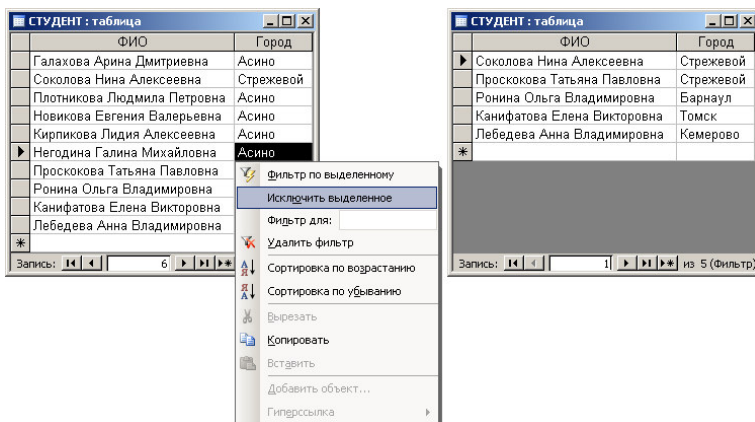


Рис. 2. Исходная таблица (слева)
и результат выполнения команды **Исключить выделенное** (справа)

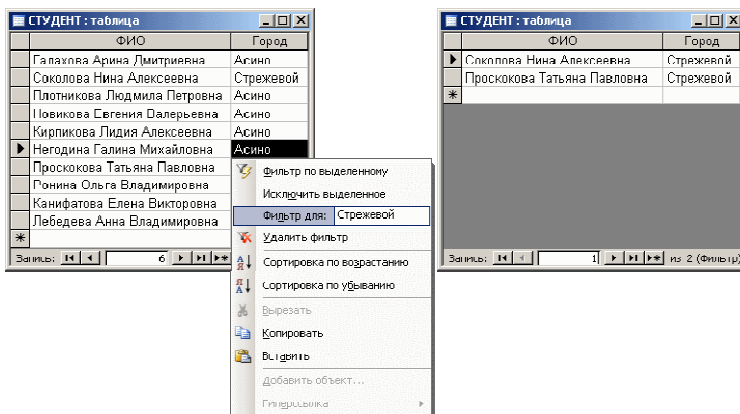


Рис. 3. Исходная таблица (слева) и результат выполнения команды **Фильтр для...** со значением параметра "Стрежевой" (справа)

Допускается повторное применение перечисленных команд к уже полученным результатам фильтрации, что приводит к последовательному уточнению фильтра. Команда **Удалить фильтр** отменяет действие фильтра.

Мастер построения запросов

Мастер построения запросов предназначен для быстрого построения простых запросов на выборку. Чтобы создать запрос с его помощью, необходимо в окне базы данных щелкнуть кнопку **Запросы** (слева на панели **Объекты**), после чего в списке запросов дважды щелкнуть левой кнопкой мыши по ярлыку **Создание запроса с помощью мастера** (рис. 4) или воспользоваться кнопкой **Создать**.

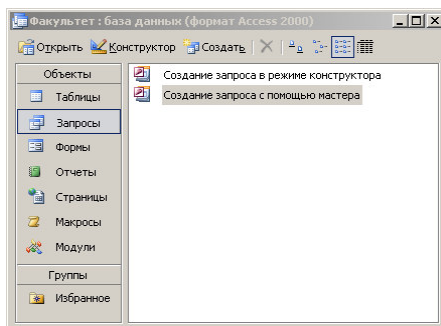


Рис. 4. Для создания запроса с помощью мастера предназначен ярлык **Создание запроса с помощью мастера**

Затем в появившемся окне мастера (рис. 5) выбрать в выпадающем списке **Таблицы и запросы** источник данных (это может быть таблица или ранее построенный запрос), с помощью кнопок со стрелками переместить требуемые поля из списка **Доступные поля** в список **Выбранные поля** (порядок полей в результирующем наборе будет соответствовать порядку полей в этом списке) и нажать кнопку **Далее**.

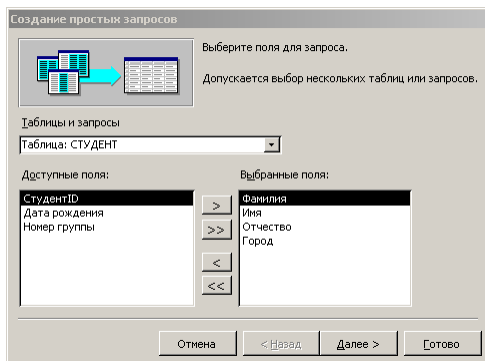


Рис. 5. Первый шаг работы с мастером построения запросов

На последнем шаге работы мастера (рис. 6) необходимо ввести имя для создаваемого запроса в поле **Задайте имя запроса**, установить переключатель в положение **Открыть запрос для просмотра данных**, если работа с запросом завершена, или в положение **Изменить макет запроса**, если требуется провести его дополнительную настройку, и нажать кнопку **Готово**.

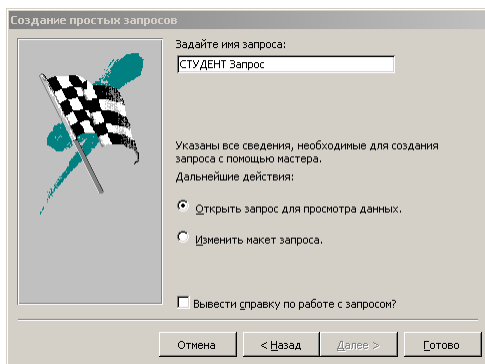


Рис. 6. Заключительный шаг работы с мастером построения запросов

Результат выполнения рассмотренного в примере запроса изображен на рис. 7.

Фамилия	Имя	Отчество	Город
Алексеев	Максим	Геннадьевич	Томск
Грачева	Ксения	Ивановна	Асино
Дмитриев	Леонид	Осипович	Стрежевой
Шишко	Ольга	Ивановна	Барнаул
Щербакова	Людмила	Леонидовна	Томск
Буланова	Людмила	Павловна	Асино
Долганова	Валентина	Викторовна	Томск
Тюнина	Людмила	Алексеевна	Кемерово
Медведев	Евгений	Владимирович	Стрежевой
Бородина	Полина	Тимофеевна	Новосибирск

Запись: 1 из 10

Рис. 7. Результат выполнения запроса

Конструктор запросов

Конструктор запросов предназначен для создания новых и изменения уже существующих запросов. Окно конструктора запросов (рис. 8) состоит из двух частей: *области схемы* (вверху) и *области сетки* (внизу), называемой также *бланк запроса*.

СТУДЕНТ

- *
- СтудентID
- Фамилия
- Имя
- Отчество
- Дата рождения

Поле:

Имя таблицы:

Сортировка:

Вывод на экран:

Условие отбора:

или:

Рис. 8. Окно конструктора запросов

Область схемы предназначена для отображения таблиц и/или запросов, являющихся источниками данных для создаваемого запроса, и связей между ними. Такие таблицы называются *базовыми таблицами запроса*. Область сетки предназначена для выбора столбцов результирующего набора, указания порядка сортировки, условий отбора и группировки строк, ввода выражений для вычисляемых столбцов, указания новых значений в запросах на обновле-

ние/добавление и имен столбцов-получателей в запросах на добавление.

Окно конструктора запросов содержит также *разделительную линию* – тонкую серую полосу между областью схемы и областью сетки, – позволяющую изменять вертикальные размеры этих областей, и полосы прокрутки, позволяющие просматривать информацию не поместившуюся на экране.

При работе с конструктором запросов полезными будут команды, расположенные в пунктах главного меню **Вид**, **Вставка** и **Запрос**, а также кнопки панели инструментов **Конструктор запросов** (рис. 9), которая появляется на экране, как только начинается работа с конструктором. Наиболее важными командами являются: команда **Запуск** (меню **Запрос**), приводящая к *выполнению запроса* и выводу на экран таблицы с *результатами запроса*, и команда **Режим таблицы** (меню **Вид**), приводящая к выводу на экран таблицы с результатами запроса без его фактического выполнения. Невидимое на первый взгляд различие в поведении этих команд проявляется в запросах, приводящих к изменению данных: в запросах на создание, добавление, обновление и удаление переключение в режим таблицы приводит к показу записей, с которыми указанные действия будут произведены, если будет дана команда **Запуск**; сами действия при этом не производятся.

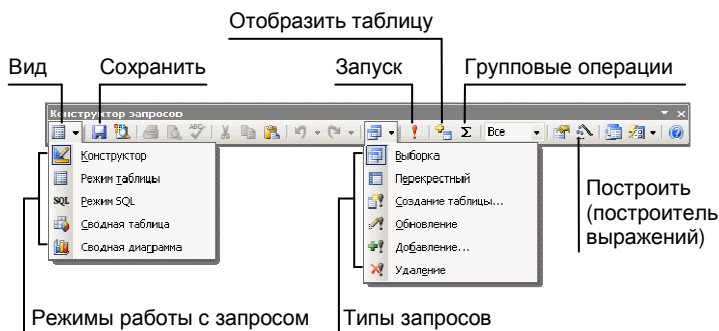


Рис. 9. Панель инструментов **Конструктор запросов**

В конструкторе запросов выполняются следующие операции:

- добавление/удаление базовых таблиц в запрос/из запроса;
- добавление, удаление и изменение порядка полей;
- настройка параметров запроса для решения требуемой задачи.

Далее все эти операции будут подробно рассмотрены.

Добавление/удаление базовых таблиц в запрос/из запроса

Практически всегда (за исключением особых случаев) запросы выполняются над данными, расположенными в таблицах базы данных или являющихся результатами выполнения других запросов, поэтому процедура создания запроса начинается с выбора базовых таблиц. Сразу после команды **Создание запроса в режиме конструктора** на экране появляются пустой бланк нового запроса и окно **Добавление таблицы** (рис. 10), в котором необходимо выбрать требуемые источники данных (таблицы и/или запросы).

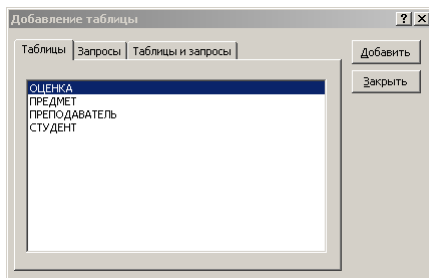


Рис. 10. Окно **Добавление таблицы**

Выбор объектов в этом окне осуществляется стандартным способом: одинарный щелчок левой кнопкой мыши по объекту приводит к его выделению, удержание на клавиатуре во время щелчка клавиши **CTRL** позволяет выбрать несколько объектов, а удержание клавиши **SHIFT** позволяет выбрать группу соседних объектов. Нажатие кнопки **Добавить** приводит к помещению выбранных объектов в область схемы запроса; кнопка **Заккрыть** закрывает окно.

Если, после закрытия окна **Добавление таблицы**, потребовалось добавить в запрос еще один или несколько источников данных (таблиц и/или запросов), то повторно открыть его можно с помощью команды **Отобразить таблицу** в меню **Запрос** или с помощью одноименной кнопки на панели инструментов **Конструктор запросов** (см. рис. 9).

Добавить таблицу в запрос можно также простым перетаскиванием ее пиктограммы из окна базы данных в область схемы окна конструктора запросов.

Удаление базовых таблиц из запроса

Чтобы удалить базовую таблицу из запроса необходимо выделить ее в области схемы, например, щелкнув левой кнопкой мыши ее название, и нажать на клавиатуре клавишу **DELETE**.

Добавление, удаление и изменение порядка полей в запросе

Как уже говорилось, одним из отличий запросов от фильтров является возможность выбора требуемых полей (столбцов) в результатах запроса. Сделать это можно одним из следующих трех способов:

- найти требуемое поле в окне таблицы в области схемы и дважды щелкнуть по нему левой кнопкой мыши;
- найти требуемое поле в окне таблицы в области схемы, выделить его (допускается выделение более одного поля с помощью клавиш **SHIFT** и **CTRL**) и затем перетащить его мышью в область сетки на требуемое место;
- в области сетки установить текстовый курсор в строку **Поле** пустого столбца, нажать появившуюся справа кнопку выпадающего списка и выбрать в списке требуемое поле (рис. 11).

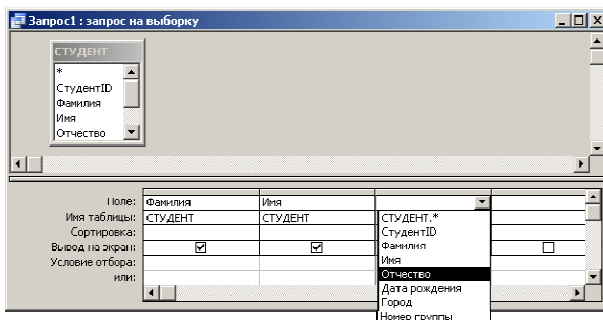


Рис. 11. Выпадающий список, появляющийся в строке **Поле**

Если требуется добавить в запрос все поля, имеющиеся в базовой таблице, то при их большом количестве все перечисленные выше способы могут оказаться утомительными. В этом случае рекомендуется воспользоваться специальным полем * (звездочка), которое имеется в каждой таблице и добавление которого в бланк запроса приводит к выводу всех имеющихся в базовой таблице полей в его результатах.

Чтобы удалить поле из запроса необходимо выделить соответствующий столбец (допускается выделение смежных столбцов) в бланке запроса и нажать на клавиатуре клавишу **DELETE**.

Для выделения столбцов предназначена область выделения столбцов – узкая серая полоска над столбцами в бланке запроса. При помещении курсора мыши в область выделения курсор принимает форму жирной направленной вниз стрелки. Нажатие левой кнопки мыши в этот момент приведет к выделению столбца.

Столбцы в результатах выполнения запроса отображаются в порядке, определенном в бланке запроса. Изменить порядок их следования можно перестановкой столбцов в бланке запроса. Делается это стандартным способом: сначала с помощью зоны выделения производится выделение требуемого столбца или нескольких смежных столбцов, после чего выделенные столбцы перемещаются мышью на новое место (отпускать кнопку мыши следует когда указатель мыши окажется перед столбцом, который должен быть справа от вставляемого столбца). Ширина столбцов в бланке запроса также может меняться. Осуществляется это простым перетаскиванием мышью границы столбца или двойным щелчком мыши по линии границы, как это делается в таблицах Microsoft Excel.

Пример 1. В базе данных (файл **Пример 1.mdb**) имеется таблица **СТУДЕНТ** с данными об обучающихся на факультете студентах (рис. 12).

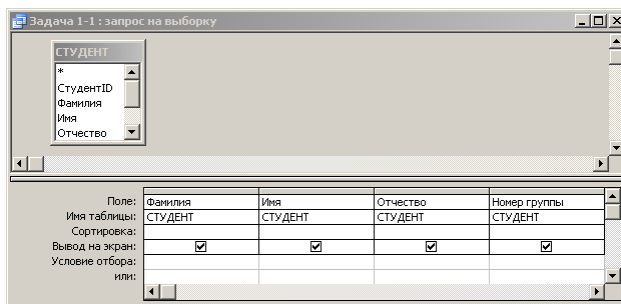


СтудентID	Фамилия	Имя	Отчество	Дата рождения	Город	Номер группы
1	Алексеев	Максим	Геннадьевич	05.08.1988	Томск	101
2	Грачева	Ксения	Ивановна	25.11.1989	Асино	101
3	Дмитриев	Леонид	Осипович	16.10.1987	Стрежевой	104
4	Шишко	Ольга	Ивановна	24.02.1989	Барнаул	102
5	Щербакова	Людмила	Леонидовна	24.11.1989	Томск	102
6	Буланова	Людмила	Павловна	25.09.1987	Асино	104
7	Долганова	Валентина	Викторовна	10.07.1987	Томск	103
8	Тюнина	Людмила	Алексеевна	27.07.1988	Кемерово	101
9	Медведев	Евгений	Владимирович	14.01.1990	Стрежевой	104
10	Бородина	Полина	Тимофеевна	06.07.1987	Новосибирск	103

Рис. 12. Исходные данные (пример 1)

Задача 1.1. Построить запрос, извлекающий выборку со следующими полями: **Фамилия, Имя, Отчество, Номер группы.**

Решение.



Поле:	Фамилия	Имя	Отчество	Номер группы
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				
или:				

Рис. 13. Запрос для решения задачи 1.1

Результат выполнения запроса.

Фамилия	Имя	Отчество	Номер группы
Алексеев	Максим	Геннадьевич	101
Грачева	Ксения	Ивановна	101
Дмитриев	Леонид	Осипович	104
Шишко	Ольга	Ивановна	102
Щербакова	Людмила	Леонидовна	102
Буланова	Людмила	Павловна	104
Долганова	Валентина	Викторовна	103
Тюнина	Людмила	Алексеевна	101
Медведев	Евгений	Владимирович	104
Бородина	Полина	Тимофеевна	103

Рис. 14. Результат выполнения запроса (задача 1.1)

Сортировка

По умолчанию порядок записей в результатах запроса определяется порядком записей исходной базовой таблицы. Изменить его можно с помощью строки **Сортировка**. Для каждого столбца, по которому требуется выполнить сортировку необходимо в выпадающем списке строки установить требуемый порядок: по возрастанию или по убыванию.

Сортировка записей осуществляется по выбранным столбцам в порядке слева направо, т.е. столбцы, расположенные слева, имеют более высокий приоритет перед столбцами, расположенными справа. Если требуется изменить приоритет столбцов, не меняя их взаимного расположения, то сделать это можно, воспользовавшись переключателями в строке **Вывод на экран**. Установленный в положении "выключено" переключатель означает, что соответствующий столбец не отображается в результатах запроса, хотя по-прежнему влияет на его работу, например, на порядок сортировки.

Пример 2. В базе данных (файл **Пример 2.mdb**) имеется таблица **СТУДЕНТ** с данными о студентах (рис. 15).

СтудентID	Фамилия	Имя	Отчество	Дата рождения	Город	Номер группы
1	Васечкин	Алексей	Иванович	25.11.1989	Асино	101
2	Васечкин	Евгений	Осипович	26.10.1987	Стрежевой	104
3	Шишко	Ольга	Ивановна	14.02.1990	Барнаул	102
4	Щербакова	Людмила	Леонидовна	04.12.1991	Томск	102
5	Буланова	Людмила	Павловна	15.10.1987	Асино	104
6	Васечкин	Олег	Петрович	10.07.1987	Томск	103
7	Тюнина	Людмила	Алексеевна	17.12.1988	Кемерово	101
8	Васечкин	Евгений	Владимирович	24.11.1991	Стрежевой	104
9	Бородина	Полина	Тимофеевна	16.07.1987	Новосибирск	103
10	Васечкин	Евгений	Геннадьевич	25.08.1989	Томск	101

Рис. 15. Исходные данные (пример 2)

Задача 2.1. Построить запрос, извлекающий выборку с полями: **Фамилия, Имя, Отчество, Номер группы**. Выборка должна быть упорядочена по полям **Фамилия, Имя, Отчество** в алфавитном порядке.

Решение.

Поле:	Фамилия	Имя	Отчество	Номер группы
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
Сортировка:	по возрастанию	по возрастанию	по возрастанию	по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				
или:				

Рис. 16. Запрос для решения задачи 2.1

Результат выполнения запроса.

Фамилия	Имя	Отчество	Номер группы
Бородина	Полина	Тимофеевна	103
Буланова	Людмила	Павловна	104
Васечкин	Алексей	Иванович	101
Васечкин	Евгений	Владимирович	104
Васечкин	Евгений	Геннадьевич	101
Васечкин	Евгений	Осипович	104
Васечкин	Олег	Петрович	103
Тюнина	Людмила	Алексеевна	101
Шишко	Ольга	Ивановна	102
Щербаклова	Людмила	Леонидовна	102

Рис. 17. Результат выполнения запроса (задача 2.1)

Комментарий. Обратите внимание, что записи о студентах с совпадающими фамилиями в результирующем наборе оказались собранными вместе и упорядоченными между собой по полю **Имя**; записи, в которых одновременно совпали значения полей **Фамилия** и **Имя**, также оказались собранными вместе и упорядоченными между собой по полю **Отчество**.

Задача 2.2. Построить запрос, извлекающий выборку с полями: **Фамилия, Имя, Отчество, Номер группы**, При этом выборка должна быть упорядочена вначале по полю **Номер группы** (по возрастанию) и затем по полям **Фамилия, Имя, Отчество** (в алфавитном порядке).

Решение.

Задача 2-2 : запрос на выборку

СТУДЕНТ

Поле: ☐ Фамилия ☐ Имя ☐ Отчество ☐ Номер группы

Имя таблицы: ☐ СТУДЕНТ ☐ СТУДЕНТ ☐ СТУДЕНТ ☐ СТУДЕНТ ☐ СТУДЕНТ ☐ СТУДЕНТ ☐ СТУДЕНТ

Сортировка: ☒ по возрастанию ☐ по возрастанию ☐ по возрастанию ☐ по возрастанию

Вывод на экран: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Условие отбора: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Рис. 18. Запрос для решения задачи 2.2

Результат выполнения запроса.

Задача 2-2 : запрос на выборку

Фамилия	Имя	Отчество	Номер группы
Васечкин	Алексей	Иванович	101
Васечкин	Евгений	Геннадьевич	101
Тюнина	Людмила	Алексеевна	101
Шишко	Ольга	Ивановна	102
Щербакова	Людмила	Леонидовна	102
Бородин	Полина	Тимофеевна	103
Васечкин	Олег	Петрович	103
Буланова	Людмила	Павловна	104
Васечкин	Евгений	Владимирович	104
Васечкин	Евгений	Осипович	104

Записи: 1 из 10

Рис. 19. Результат выполнения запроса (задача 2.2)

Комментарий. Для решения задачи поля **Фамилия**, **Имя**, **Отчество** добавлены в бланк запроса дважды. Повторно добавленные поля используются только для указания порядка сортировки и в результирующем наборе не присутствуют. Заметим, тот же результат можно получить, добавив перед столбцом **Фамилия** столбец **Номер группы**, включив по нему сортировку и отключив вывод на экран.

Наложение условий отбора

Для наложения условий отбора результирующих записей предназначена строка **Условие отбора** в бланке запроса и все строки, расположенные ниже ее. Если имеющихся строк недостаточно, то увеличить их количество можно командой **Строки** в меню **Вставка**. Условием отбора может быть:

- простое значение (*literal*), например, символьная константа "Иванов", число 10 или дата #28.05.2009#;
- логическое выражение, построенное с применением операций сравнения, таких как > (больше), < (меньше), = (равно),

<> (неравно), >= (больше или равно), <= (меньше или равно), объединенных операциями **AND** (логическое умножение), **OR** (логическое сложение) или **NOT** (операция отрицания или инверсия) и операторов **LIKE**, **IN**, **BETWEEN**,... (см. примеры);

- выражение, построенное с применением встроенных в Access или созданных пользователем функций (см. следующие разделы).

Условия, находящиеся в разных столбцах одной строки, объединяются логической операцией **AND**, находящиеся в разных строках – операцией **OR**.

Пример 3. В базе данных (файл **Пример 3.mdb**) имеется таблица **СТУДЕНТ** с данными о студентах (рис. 20).

СтудентID	Фамилия	Имя	Отчество	Дата рождения	Город	Номер группы	Математика
1	Маслова	Галина	Андреевна	18.10.1988	Стрежевой	101	3
2	Петров	Иван	Иванович	02.10.1989	Стрежевой	104	4
3	Васечкин	Александр	Сергеевич	26.10.1988	Новосибирск	105	3
4	Андрейчук	Татьяна	Владимировна	10.07.1989	Барнаул	103	4
5	Ясницкая	Наталья	Петровна	10.10.1988	Томск	101	3
6	Иванникова	Наталья	Иосифовна	08.12.1988	Стрежевой	101	5
7	Васечкин	Александр	Михайлович	09.08.1987	Асино	102	5
8	Ишина	Ольга	Николаевна	07.09.1987	Асино	102	5
9	Петров	Виктор	Семенович	08.02.1988	Асино	102	4
10	Петров	Алексей	Николаевич	23.04.1990	Томск	105	3
11	Ахмадиева	Ирина	Зуфаровна	09.04.1990	Томск	103	4
12	Степанюк	Надежда	Васильевна	02.06.1989	Асино	101	3

Рис. 20. Исходные данные (пример 3)

Задача 3.1. Построить запрос, извлекающий выборку с результатами по математике всех студентов с фамилией Петров. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Номер группы, Математика.**

Решение.

Задача 3-1 : запрос на выборку

Таблица: **СТУДЕНТ**

Поле: **Фамилия, Имя, Отчество, Номер группы, Математика**

Критерии: **Петров**

Рис. 21. Запрос для решения задачи 3.1

Результат выполнения запроса.

	Фамилия	Имя	Отчество	Номер группы	Математика
▶	Петров	Иван	Иванович	104	4
	Петров	Виктор	Семенович	102	4
	Петров	Алексей	Николаевич	105	3
✱					

Запись: 1 из 3

Рис. 22. Результат выполнения запроса (задача 3.1)

Задача 3.2. Построить запрос, извлекающий выборку с результатами по математике студента с фамилией Петров из группы с номером 102. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Номер группы, Математика.**

Решение.

СТУДЕНТ

Имя: СТУДЕНТ

Номер группы: СТУДЕНТ

Условие отбора: Имя = Петров

Вывод на экран: Имя, Имя, Имя, Имя, Имя

Поле:	Фамилия	Имя	Отчество	Номер группы	Математика
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
Сортировка:					
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:	Имя = Петров			102	

Рис. 23. Запрос для решения задачи 3.2

Результат выполнения запроса.

	Фамилия	Имя	Отчество	Номер группы	Математика
▶	Петров	Виктор	Семенович	102	4
✱					

Запись: 1 из 1

Рис. 24. Результат выполнения запроса (задача 3.2)

Задача 3.3. Построить запрос, извлекающий выборку с результатами по математике всех студентов с фамилией Петров и студентов из группы с номером 102. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Номер группы, Математика.**

Решение.

Задача 3-3 : запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Имя Отчество Номер группы Математика
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ
Сортировка:
Вывод на экран: ☒ ☒ ☒ ☒ ☒
Условие отбора: "Петров" или: 102

Рис. 25. Запрос для решения задачи 3.3

Результат выполнения запроса.

Задача 3-3 : запрос на выборку

Фамилия	Имя	Отчество	Номер группы	Математика
Петров	Иван	Иванович	104	4
Васечкин	Александр	Михайлович	102	5
Ишина	Ольга	Николаевна	102	5
Петров	Виктор	Семенович	102	4
Петров	Алексей	Николаевич	105	3

Записи: 1 из 5

Рис. 26. Результат выполнения запроса (задача 3.3)

Задача 3.4. Построить запрос, извлекающий выборку с результатами по математике студентов Петрова и Васечкина из группы с номером 102. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Номер группы, Математика.**

Решение.

Задача 3-4 : запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Имя Отчество Номер группы Математика
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ
Сортировка:
Вывод на экран: ☒ ☒ ☒ ☒ ☒
Условие отбора: "Петров" или: "Васечкин"

Рис. 27. Запрос для решения задачи 3.4

Результат выполнения запроса.

Фамилия	Имя	Отчество	Номер группы	Математика
Васечкин	Александр	Михайлович	102	5
Петров	Виктор	Семенович	102	4

Рис. 28. Результат выполнения запроса (задача 3.4)

Задача 3.5. Построить запрос, извлекающий фамилии студентов с оценками "4" и "5" по математике. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Номер группы, Математика.**

Решение (вариант 1):

Фамилия	Имя	Отчество	Номер группы	Математика
Васечкин	Александр	Михайлович	102	5
Петров	Виктор	Семенович	102	4

Рис. 29. Первый вариант запроса для решения задачи 3.5

Решение (вариант 2):

Фамилия	Имя	Отчество	Номер группы	Математика
Васечкин	Александр	Михайлович	102	5
Петров	Виктор	Семенович	102	4

Рис. 30. Второй вариант запроса для решения задачи 3.5

Результат выполнения запроса.

Фамилия	Имя	Отчество	Номер группы	Математика
Петров	Иван	Иванович	104	4
Андрейчук	Татьяна	Владимировна	103	4
Иванникова	Наталья	Иосифовна	101	5
Васечкин	Александр	Михайлович	102	5
Ишина	Ольга	Николаевна	102	5
Петров	Виктор	Семенович	102	4
Ахмадиева	Ирина	Зуфаровна	103	4

Записей: 1 из 7

Рис. 31. Результат выполнения запроса (задача 3.5)

Задача 3.6. Построить запрос, извлекающий выборку с результатами по математике студентов из групп со 102 по 104 включительно. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Номер группы, Математика.**

Решение (вариант 1):

Фамилия	Имя	Отчество	Номер группы	Математика
СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
			102	
			103	
			104	

Рис. 32. Первый вариант запроса для решения задачи 3.6

Решение (вариант 2):

Фамилия	Имя	Отчество	Номер группы	Номер группы	Математика
СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
			>=102	<=104	

Рис. 33. Второй вариант запроса для решения задачи 3.6

Решение (вариант 3):

Задача 3-6 (вариант 3): запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Имя Отчество Номер группы Математика
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ
Сортировка:
Вывод на экран: ☒ ☒ ☒ ☒ ☒
Условие отбора:
или: Between 102 And 104

Рис. 34. Третий вариант запроса для решения задачи 3.6

Решение (вариант 4):

Задача 3-6 (вариант 4): запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Имя Отчество Номер группы Математика
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ СТУДЕНТ
Сортировка:
Вывод на экран: ☒ ☒ ☒ ☒ ☒
Условие отбора:
или: In (102;103;104)

Рис. 35. Четвертый вариант запроса для решения задачи 3.6

Результат выполнения запроса.

Задача 3-6 : запрос на выборку

	Фамилия	Имя	Отчество	Номер группы	Математика
▶	Петров	Иван	Иванович	104	4
	Андрейчук	Татьяна	Владимировна	103	4
	Васечкин	Александр	Михайлович	102	5
	Ишина	Ольга	Николаевна	102	5
	Петров	Виктор	Семенович	102	4
	Ахмадиева	Ирина	Зуфаровна	103	4
✱					

Записи: 1 из 6

Рис. 36. Результат выполнения запроса (задача 3.6)

Комментарий. В данной задаче необходимо извлечь выборку, значение одного из полей которой (**Номер группы**) должно находиться в диапазоне чисел [102, 104]. Первый вариант решения – самый простой – подходит в ситуациях, когда рассматриваются только целые числа, а диапазон допустимых значений узкий. Если же диапазон широкий или область допустимых значений поля не огра-

ничивается множеством целых чисел, то условие на отбор записей можно сформулировать в форме неравенств (см. второй вариант решения на рис. 33) или с помощью оператора **BETWEEN**, которому требуется указать в качестве аргументов левую и правую границы интервала (см. третий вариант решения на рис. 34). В последнем варианте решения применяется оператор **IN**, требующий в качестве аргумента множество допустимых значений. Оператор **IN** очень полезен в ситуациях, когда множество допустимых значений является результатом другого запроса.

Задача 3.7. Построить запрос, возвращающий список студентов, проживающих не в Томске и не в Новосибирске. Выборка должна содержать поля: **Фамилия, Имя, Отчество, Город**.

Решение (вариант 1):

Поле:	Фамилия	Имя	Отчество	Город
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	<>'Томск' And <>'Новосибирск'			
или:				

Рис. 37. Первый вариант запроса для решения задачи 3.7

Решение (вариант 2):

Поле:	Фамилия	Имя	Отчество	Город
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Not In ('Томск','Новосибирск')			
или:				

Рис. 38. Второй вариант запроса для решения задачи 3.7

Результат выполнения запроса.

Фамилия	Имя	Отчество	Город
Маслова	Галина	Андреевна	Стрежевой
Петров	Иван	Иванович	Стрежевой
Андрейчук	Татьяна	Владимировна	Барнаул
Иванникова	Наталья	Иосифовна	Стрежевой
Васечкин	Александр	Михайлович	Асино
Ишина	Ольга	Николаевна	Асино
Петров	Виктор	Семенович	Асино
Степанок	Надежда	Васильевна	Асино

Рис. 39. Результат выполнения запроса (задача 3.7)

Задача 3.8. Построить запрос, возвращающий список студентов, родившихся в октябре 1988 года. Результирующий набор должен содержать поля: **Фамилия, Имя, Отчество, Дата рождения.**

Решение.

Рис. 40. Запрос для решения задачи 3.8

Результат выполнения запроса.

Фамилия	Имя	Отчество	Дата рождения
Маслова	Галина	Андреевна	18.10.1988
Васечкин	Александр	Сергеевич	26.10.1988
Ясницкая	Наталья	Петровна	10.10.1988

Рис. 41. Результат выполнения запроса (задача 3.8)

Комментарий. Обратите внимание, что в Access значения дат заключаются между символами # (решетка). Для решения этой задачи также можно было использовать оператор **BETWEEN**.

Задача 3.9. Построить запрос, возвращающий список студентов, фамилии которых начинаются на букву "А". Результирующий набор должен содержать поля: **Фамилия, Имя, Отчество.**

Решение.

Задача 3-9: запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Имя Отчество
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ
Сортировка:
Вывод на экран: ☒ ☒ ☒
Условие отбора: Like "A*" или:

Рис. 42. Запрос для решения задачи 3.9

Результат выполнения запроса.

Фамилия	Имя	Отчество
Андрейчук	Татьяна	Владимировна
Ахмадиева	Ирина	Зуфаровна

Запись: 1 из 2

Рис. 43. Результат выполнения запроса (задача 3.9)

Комментарий. Для решения данной задачи необходим оператор **LIKE**, позволяющий извлекать "похожие" на образец записи. При создании образца можно использовать символ * (звездочка), означающий любой набор символов, и символ ? (вопрос) – любой символ.

Задача 3.10. Построить запрос, возвращающий список всех девушек факультета. Результирующий набор должен содержать поля: **Фамилия, Имя, Отчество.**

Решение.

Задача 3-10: запрос на выборку

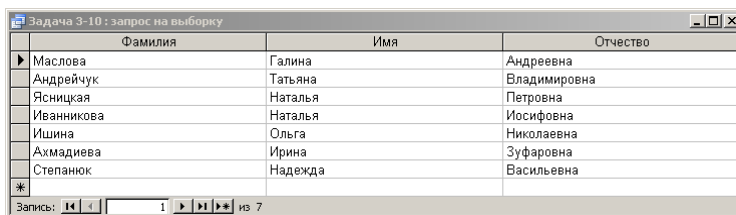
СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: СтудентID Фамилия Имя Отчество
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ
Сортировка:
Вывод на экран: ☒ ☒ ☒
Условие отбора: Like "a*" или:

Рис. 44. Запрос для решения задачи 3.10

Результат выполнения запроса.



Фамилия	Имя	Отчество
Маслова	Галина	Андреевна
Андрейчук	Татьяна	Владимировна
Ясницкая	Наталья	Петровна
Иванникова	Наталья	Иосифовна
Ишина	Ольга	Николаевна
Ахмадиева	Ирина	Зуфаровна
Степанюк	Надежда	Васильевна

Рис. 45. Результат выполнения запроса (задача 3.10)

Комментарий. Решение данной задачи основано на предположении, что отчество женщин всегда оканчивается буквой "а". Условие отбора в этом случае принимает вид "а*", т.е. в результирующую выборку должны попасть записи, в которых отчество состоит из произвольных символов в начале и заканчивается символом "а".

Создание вычисляемых полей

Вычисляемые поля в запросах предназначены для вычисления значения введенного пользователем *выражения*. Результат вычислений не сохраняется в базовых таблицах, пересчитывается каждый раз при выполнении запроса и поэтому всегда отражает текущее состояние базы данных. Вычисляемые поля, кроме того что создаются в момент выполнения запроса, больше ничем не отличаются от обычных полей, поэтому обращаться с ними можно также как с обычными полями, например, их можно использовать в условиях отбора записей.

Различают два типа вычислений:

- *итоговые вычисления*, производимые над группами строк (см. следующий раздел);
- *пользовательские вычисления*, производимые над каждой записью с использованием одного или нескольких полей, арифметических и других операторов и встроенных в Access или определенных пользователем функций.

Для создания вычисляемого поля необходимо: установить текстовый курсор в строку **Поле** свободного столбца бланка запроса, ввести имя для создаваемого поля, поставить двоеточие и построить *выражение*. Выражение представляет собой комбинацию математических и логических операторов, имен полей, констант и функций, в результате обработки которой получается единственное значение (результат). Выражение может выполнять математические вычисле-

ния, обработку текста или производить проверку данных. При работе с выражениями следует помнить, что символьные константы требуются заключать в двойные кавычки ("); названия полей – в квадратные скобки ([,]); даты обрамлять решёткой (#); порядок вычислений определяется приоритетом операций (см. справку по Access) и может быть изменен с помощью скобок. Пример бланка запроса с вычисляемым полем изображен на рис. 46.

Рис. 46. Бланк запроса с вычисляемым полем **ФИО**

Если создаваемое выражение оказывается слишком длинным, то можно либо открыть окно **Область ввода**, нажав на клавиатуре клавиши **Shift + F2** (рис. 47), либо открыть окно **Построитель выражений** (рис. 48), нажав кнопку **Построить** на панели инструментов **Конструктор запросов**.

Рис. 47. Окно **Область ввода**

Окно построителя выражений состоит из трех частей (рис. 48):

- В верхней части окна расположено текстовое поле – *поле выражения*, – в котором формируется текст выражения. Для этого используются расположенные ниже инструменты. Допускается также непосредственная работа с содержимым этого поля.

- В средней части окна находятся кнопки с часто используемыми операторами. Нажатие на любую из них приводит к вставке в поле выражения соответствующего оператора.
- В нижней части окна в левом списке отображаются папки, содержащие: основные объекты базы данных, встроенные и определенные пользователем функции, константы, операторы и общие выражения; в среднем списке отображаются определенные элементы или типы элементов для папки, выбранной в левом списке; в правом списке отображаются значения (если они существуют) для элементов, заданных в левом и среднем списках. Нажатие на кнопку **Вставить** приводит к вставке в текущую позицию поля выражения выбранного элемента.

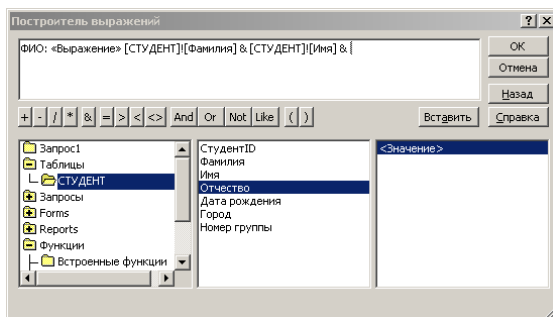


Рис. 48. Окно построителя выражений

Создание вычисляемого поля в построителе начинается с ввода имени создаваемого поля и символа двоеточия, затем строится выражение. Требуемые для вычислений объекты, операторы и функции можно вводить вручную или с помощью инструментов. В последнем случае в поле выражения могут автоматически добавляться заключенные в двойные кавычки метки «**Выражение**» (см. рис. 48), указывающие местоположение пропущенных операндов. Например, такая метка может появиться, если в выражение вставить подряд два оператора сложения (+). Иногда Access "ошибается" и добавляет эту метку, там где она не требуется (см. рис. 48). Удалить такую метку можно нажав на клавиатуре клавишу **DELETE**, предварительно выделив ее.

В Access имеется более сотни встроенных функций, позволяющих производить вычисления и преобразования числовых и символьных значений, работать с датами и временем, обрабатывать пустые и ошибочные значения.

Приведем описания некоторых часто используемых встроенных функций Access:

ABS(X)

Возвращает модуль числа **X**.

Пример:

$ABS(-7,5) = 7,5$

INT(X)

Возвращает целую часть числа **X**.

Пример:

$INT(7,5) = 7$

SQR(X)

Возвращает квадратный корень числа **X**.

Пример:

$SQR(16) = 4$

ROUND(X; P)

Округляет число **X** до указанного числа знаков (**P**) после запятой.

Пример:

$ROUND(7,121212; 3) = 7,121$

DATE()

Возвращает текущую дату.

NOW()

Возвращает текущую дату и время.

DAY(D)

Извлекает номер дня месяца из даты **D**.

Пример:

$DAY(\#21.12.2009\#) = 21.$

MONTH(D)

Извлекает номер месяца из даты **D**.

Пример:

$MONTH(\#21.12.2009\#) = 12.$

YEAR(D)

Извлекает год из даты **D**.

Пример:

YEAR(#21.12.2009#) = 2009.

DATESERIAL(Y; M; D)

Формирует дату из значений года (**Y**), месяца (**M**) и дня (**D**).

Пример:

DATESERIAL(2009; 12; 21) = #21.12. 2009#.

LEFT(S, N)

Возвращает **N** первых символов строки **S**.

Пример:

LEFT("Миру–мир! Риму–Рим!"; 9) = "Миру–мир!"

RIGHT(S, N)

Возвращает **N** последних символов строки **S**.

Пример:

RIGHT("Миру–мир! Риму–Рим!"; 9) = "Риму–Рим!"

TRIM(S)

Удаляет все пробелы в начале и в конце строки **S**.

Пример:

TRIM(" слово ") = "слово"

UCASE(S)

Переводит все символы строки **S** в верхний регистр.

Пример:

UCASE("Миру–мир!") = "МИРУ–МИР!"

LCASE(S)

Переводит все символы строки **S** в нижний регистр.

Пример:

LCASE("Риму–Рим!") = "риму–рим!"

IF(C; E1; E2)

Если логическое выражение **C** истинно, то возвращает значение выражения **E1**, иначе – значение выражения **E2**.

Пример:

IF(2 + 2 < 2 * 2; "Ooppss!"; "It's Ok!") = "It's Ok!"

NZ(V; E)

Если значение V не равно NULL, то возвращает V, иначе – E.

Пример:

NZ(NULL; "Not NULL!") = "Not NULL!"

NZ("NULL", "Not NULL!") = "NULL"

Пример 4. В базе данных (файл **Пример 4.mdb**) имеется таблица **СТУДЕНТ** с данными о студентах и их оценках (рис. 49).

СТУДЕНТ : таблица							
СтудентID	Фамилия	Имя	Отчество	Дата рождения	Математика	Информатика	Статистика
1	Гусев	Вячеслав	Андреевич	18.04.1987	5		5
2	Корольков	Максим	Иванович	02.10.1989	5	4	3
3	Ильченко	Алексей	Сергеевич	16.07.1988		5	
4	Русина	Ольга	Ивановна	10.06.1989	5		5
5	Гуленко	Татьяна	Владимировна	28.11.1988	4	3	

Запись: 1 из 5

Рис. 49. Исходные данные (пример 4)

Задача 4.1. Построить запрос, возвращающий список фамилий студентов. Список должен состоять из одного столбца **ФИО**, являющегося объединением полей **Фамилия**, **Имя**, **Отчество**.

Решение.

Рис. 50. Запрос для решения задачи 4.1

Результат выполнения запроса.

Задача 4-1: запрос на выборку	
ФИО	
Гусев Вячеслав Андреевич	
Корольков Максим Иванович	
Ильченко Алексей Сергеевич	
Русина Ольга Ивановна	
Гуленко Татьяна Владимировна	

Запись: 1 из 5

Рис. 51. Результат выполнения запроса (задача 4.1)

Задача 4.2. Построить запрос, рассчитывающий для каждого студента средний балл как среднее арифметическое оценок по результатам его экзаменов по математике, информатике и статистике. Результирующий набор должен состоять из двух столбцов: **Фамилия** и **Средний балл**.

Решение (вариант 1):

Задача 4-2 (вариант 1): запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Средний балл: ((Математика)+(Информатика)+(Статистика))/3
Имя таблицы: СТУДЕНТ
Сортировка: ☒ ☒
Вывод на экран: ☒
Условие отбора:
или:

Рис. 52. Первый вариант запроса для решения задачи 4.2

Результат выполнения запроса (1-ый вариант):

Задача 4-2 (вариант 1): запрос на выборку

Фамилия	Средний балл
Гусев	
Корольков	4
Ильченко	
Русина	5
Гуленко	

Запись: 1 из 5

Рис. 53. Результат выполнения первого варианта запроса к задаче 4.2

Решение (вариант 2):

Задача 4-2 (вариант 2): запрос на выборку

СТУДЕНТ

*
СтудентID
Фамилия
Имя
Отчество

Поле: Фамилия Средний балл: (NZ([Математика];2)+NZ([Информатика];2)+NZ([Статистика];2))/3
Имя таблицы: СТУДЕНТ
Сортировка: ☒ ☒
Вывод на экран: ☒
Условие отбора:
или:

Рис. 54. Второй вариант запроса для решения задачи 4.2

Результат выполнения запроса (2-ой вариант):

Фамилия	Средний балл
Гусев	4
Корольков	4
Ильченко	3
Руфина	5
Гуленко	3

Рис. 55. Результат выполнения второго варианта запроса к задаче 4.2

Комментарий. Так как в Access результат вычисления арифметического выражения будет **NULL**, если хотя бы один из операндов равен **NULL**, то в первом варианте решения у студентов, не сдавших хотя бы один экзамен, значение среднего балла равно **NULL**. Во втором варианте решения делается предположение, что отсутствие оценки равносильно оценке 2 (двойка). Для этого в выражении для вычисления среднего балла вместо исходных оценок используются значения, преобразованные функцией **NZ**. В результате средний балл удалось рассчитать для всех студентов.

Задача 4.3. Построить запрос, возвращающий список студентов старше 20 лет. Результирующий набор должен содержать столбцы: **Фамилия, Имя, Отчество, Возраст**.

Решение.

Поле:	Фамилия	Имя	Отчество	Возраст: Year(Date()) - Year([Дата рождения])
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				>20

Рис. 56. Запрос для решения задачи 4.3

Результат выполнения запроса.

Фамилия	Имя	Отчество	Возраст
Гусев	Вячеслав	Андреевич	22
Ильченко	Алексей	Сергеевич	21
Гуленко	Татьяна	Владимировна	21

Рис. 57. Результат выполнения запроса (задача 4.3)

Агрегированные запросы

Агрегированные запросы в процессе выполнения разбивают исходный набор данных на группы схожих между собой записей и для каждой группы рассчитывают для заданных столбцов числовые характеристики с применением следующих *статистических (агрегирующих) функций*:

COUNT

Подсчитывает количество значений выбранного столбца.

SUM

Рассчитывает сумму значений выбранного столбца.

AVG

Рассчитывает среднее арифметическое значений выбранного столбца.

VAR

Рассчитывает дисперсию значений выбранного столбца.

STDEV

Рассчитывает среднеквадратичное отклонение значений от среднего значения выбранного столбца.

MAX

Находит значение максимального элемента выбранного столбца.

MIN

Находит значение минимального элемента выбранного столбца.

FIRST

Возвращает первую запись выбранного столбца.

LAST

Возвращает последнюю запись выбранного столбца.

Замечание. Все перечисленные выше функции кроме **FIRST** и **LAST** игнорируют пустое значение (**NULL**). Функции **FIRST** и **LAST** возвращают записи в том порядке, в котором они были введены. Сортировка записей не влияет на выполнение этих функций.

Для построения агрегированного запроса необходимо открыть конструктор запросов и дать команду **Групповые операции** (меню **Вид**) или нажать одноименную кнопку на панели инструментов. В результате в бланке запроса появится дополнительная строка **Групповая операция** (рис. 58), позволяющая для каждого столбца определить одну из следующих *групповых операций*:

- **Группировка**

По значению элементов выбранного столбца производится разбиение исходного множества записей на группы строк с одинаковым значением выбранного столбца.

- **Статистическая функция**

По значениям выбранного столбца производится расчет значения выбранной статистической функции.

- **Выражение**

По значениям выбранного столбца производится расчет сложного выражения, содержащего в себе одну или несколько статистических функций.

- **Условие**

Определяет условия на значения выбранного столбца для строк, которые будут участвовать в разбиении на группы; строки, не удовлетворяющие наложенным условиям, в расчетах участвовать не будут.

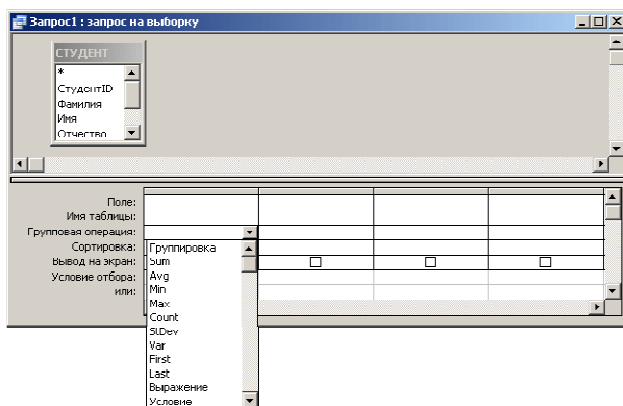
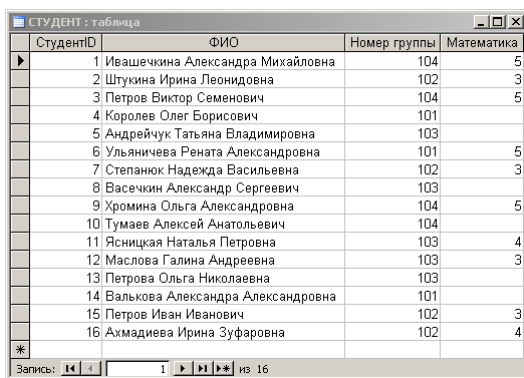


Рис. 58. Вид бланка запроса в режиме групповых операций

Пример 5. В базе данных (файл **Пример 5.mdb**) имеется таблица **СТУДЕНТ** с данными об обучающихся на факультете студентах и полученных ими оценках по математике (рис. 59).



СтудентID	ФИО	Номер группы	Математика
1	Ивашечкина Александра Михайловна	104	5
2	Штукина Ирина Леонидовна	102	3
3	Петров Виктор Семенович	104	5
4	Королев Олег Борисович	101	
5	Андрейчук Татьяна Владимировна	103	
6	Ульяничева Рената Александровна	101	5
7	Степанюк Надежда Васильевна	102	3
8	Васечкин Александр Сергеевич	103	
9	Хромина Ольга Александровна	104	5
10	Тумаев Алексей Анатольевич	104	
11	Ясницкая Наталья Петровна	103	4
12	Маслова Галина Андреевна	103	3
13	Петрова Ольга Николаевна	103	
14	Валькова Александра Александровна	101	
15	Петров Иван Иванович	102	3
16	Ахмадиева Ирина Зуфаровна	102	4

Рис. 59. Исходные данные (пример 5)

Задача 5.1. Определить количество обучающихся на факультете студентов.

Решение.

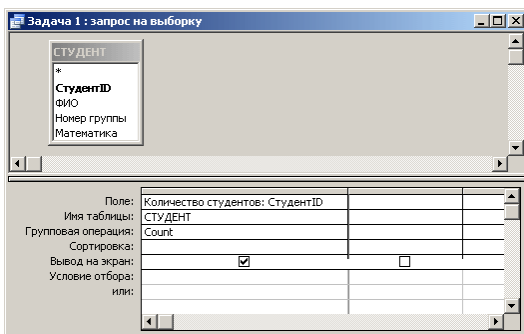
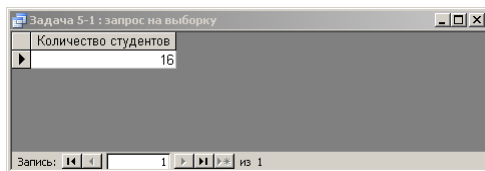


Рис. 60. Запрос для решения задачи 5.1

Результат выполнения запроса.



Количество студентов
16

Рис. 61. Результат выполнения запроса (задача 5.1)

Комментарий. Для решения задачи достаточно определить количество записей в таблице **СТУДЕНТ** с помощью статистической функции **COUNT**. Принимая во внимание, что все статистические функции игнорируют пустые значения (**NULL**), в качестве аргумента функции следует выбрать столбец, гарантированно не содержащий пустых значений. Обычно, наилучшим претендентом на роль такого столбца является первичный ключ таблицы. В данном случае – это столбец **СтудентID**.

Задача 5.2. Для каждой группы определить количество студентов.

Решение.

Задача 5-2: запрос на выборку

СТУДЕНТ

*
СтудентID
ФИО
Номер группы
Математика

Поле: Номер группы Количество студентов: СтудентID
Имя таблицы: СТУДЕНТ СТУДЕНТ
Групповая операция: Группировка Count
Сортировка:
Вывод на экран: ☒ ☒
Условие отбора:
или:

Рис. 62. Запрос для решения задачи 5.2

Результат выполнения запроса.

Задача 5-2: запрос на выборку

Номер группы	Количество студентов
101	3
102	4
103	5
104	4

Запись: 14 | 1 | 1 | >= | из 4

Рис. 63. Результат выполнения запроса (задача 5.2)

Комментарий. Для решения данной задачи необходимо разбить исходный набор данных на подгруппы по значениям столбца **Номер группы**, после чего посчитать количество записей в каждой подгруппе с помощью функции **COUNT**. Как и в предыдущей задаче, наилучшим претендентом для подсчета количества записей в каждой подгруппе является столбец **СтудентID**.

Задача 5.3. Рассчитать средний балл по математике по результатам всех студентов.

Решение.

Рис. 64. Запрос для решения задачи 5.3

Результат выполнения запроса.

Рис. 65. Результат выполнения запроса (задача 5.3)

Комментарий. Для решения задачи применяется статистическая функция **AVG**. Так как требуется рассчитать средний балл по всем результатам, то группировка записей не производится.

Задача 5.4. Рассчитать средний балл по математике для каждой группы.

Решение.

Рис. 66. Запрос для решения задачи 5.4

Результат выполнения запроса.

Номер группы	Средний балл
101	5
102	3,25
103	3,5
104	5

Рис. 67. Результат выполнения запроса (задача 5.4)

Комментарий. В отличие от предыдущей задачи, средний балл требуется рассчитать отдельно для каждой группы студентов, поэтому в запрос добавлена операция группировки данных по столбцу **Номер группы**. При выполнении запроса вначале происходит разбиение исходного множества строк на подгруппы с одинаковым значением поля **Номер группы**, после чего для каждой подгруппы вычисляется среднее значение.

Задача 5.5. Рассчитать средний балл по математике для групп 101 и 103.

Решение.

Рис. 68. Запрос для решения задачи 5.5

Результат выполнения запроса.

Номер группы	Средний балл
101	5
103	3,5

Рис. 69. Результат выполнения запроса (задача 5.5)

Комментарий. Отличие данной задачи от предыдущей заключается в требовании ограничить результаты указанными группами. Для выполнения требования в бланк запроса добавлены условия на значения поля **Номер группы**. Выполнение запроса, как и в предыдущем случае, начинается с разбиения исходного множества на подгруппы с одинаковым значением поля **Номер группы**, затем в каждой из подгрупп вычисляется среднее значение, после чего производится фильтрация в соответствии с наложенными условиями и вывод результатов.

Задача 5.6. Рассчитать средний балл по математике по результатам студентов групп 101 и 103.

Решение.

Поле:	Номер группы	Средний балл:	Математика
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	
Групповая операция:	Условие	Avg	
Сортировка:			
Вывод на экран:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	101		
или:	103		

Рис. 70. Запрос для решения задачи 5.6

Результат выполнения запроса.

Средний балл
4

Запись: 1 из 1

Рис. 71. Результат выполнения запроса (задача 5.6)

Комментарий. Отличие этой задачи от предыдущей заключается в том, что средний балл требуется рассчитать для студентов групп 101 и 103, не делая между ними различий. Для этого в строке **Групповая операция** выбирается значение **Условие** и накладываются условия фильтрации. Выполнение запроса начинается с фильтрации исходного множества строк, после чего осуществляется их группировка и вычисление итогового значения. Обратите внима-

ние, что столбец, по которому осуществляется предварительная фильтрация данных на экран не выводится – Access автоматически сбрасывает флажок **Вывод на экран**.

Задача 5.7. Определить количество студентов в каждой группе, не сдавших экзамен (не имеющих оценки) по математике.

Решение.

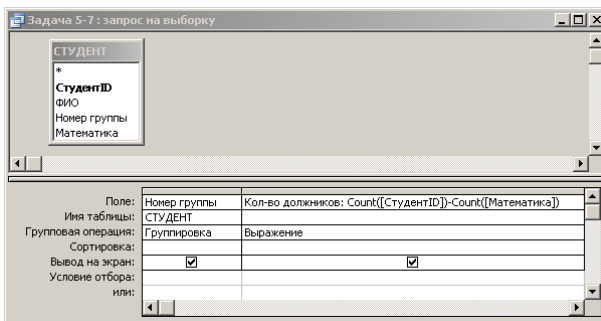


Рис. 72. Запрос для решения задачи 5.7

Результат выполнения запроса.

Номер группы	Кол-во должников
101	2
102	0
103	3
104	1

Запись: 1 из 4

Рис. 73. Результат выполнения запроса (задача 5.7)

Комментарий. В данной задаче требуется разбить исходную выборку на подгруппы, после чего для каждой из них посчитать количество пустых значений. Проблема заключается в том, что статистические функции игнорируют пустые значения. Поэтому для поиска искоемых величин необходимо разбить исходную выборку на подгруппы, найти в каждой подгруппе количество обучающихся студентов и вычесть из него количество студентов, получивших какую-нибудь оценку по математике – это и будет искомая величина.

Многотабличные запросы

Все рассмотренные выше запросы основывались на данных одной таблицы, на практике же чаще всего встречаются ситуации, когда для получения ответа на вопрос требуются данные сразу из нескольких источников данных (таблиц и/или запросов). В этом случае применяются многотабличные запросы. При их построении в бланк запроса добавляется несколько источников данных и определяются способы их объединения, в остальном многотабличные запросы ничем не отличаются от обычных запросов.

Пример бланка многотабличного запроса представлен на рис. 74. Источниками данных для запроса являются две таблицы: **ПРЕПОДАВАТЕЛЬ** и **ПРЕДМЕТ** (рис. 75), хранящие имена преподавателей и названия предметов соответственно. В данном примере предполагается, что между сущностями "Преподаватель" и "Предмет" имеется отношение "один-ко-многим" (т.е. один курс читается одним преподавателем, один преподаватель читает много курсов), которое в соответствии с теорией построения реляционных баз данных реализовано путем добавления в таблицу **ПРЕДМЕТ** внешнего ключа (атрибут **ЛекторID**), ссылающегося на таблицу **ПРЕПОДАВАТЕЛЬ**. Линия, соединяющая таблицы в бланке запроса, называется *линией объединения*. Она показывает, по значениям каких столбцов и каким образом происходит объединение записей исходных таблиц. Для ее создания необходимо мышью "ухватить" одно из полей, формирующих связь, после чего переместить и "бросить" его над вторым полем связи (например, можно "ухватить" поле **ЛекторID** и "бросить" его над полем **ПреподавательID**). Запрос возвращает список преподавателей и названий предметов, которые они преподают (рис. 76).

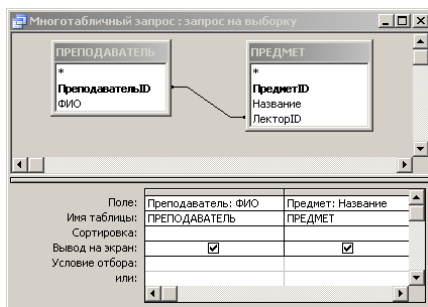


Рис. 74. Многотабличный запрос

ПРЕПОДАВАТЕЛЬ : таблица	
ПреподавательID	ФИО
1	Верников Руслан Андреевич
2	Морозов Илья Алексеевич
3	Алексеев Максим Петрович
4	Понтягин Сергей Иванович
Запись: 1 из 4	

ПРЕДМЕТ : таблица		
ПредметID	Название	ЛекторID
1	Математика	1
2	Информатика	2
3	Статистика	1
4	Литература	3
5	Философия	
Запись: 1 из 5		

Рис. 75. Содержимое таблиц ПРЕПОДАВАТЕЛЬ и ПРЕДМЕТ

Многотабличный запрос : запрос на выборку	
Преподаватель	Предмет
Верников Руслан Андреевич	Математика
Морозов Илья Алексеевич	Информатика
Верников Руслан Андреевич	Статистика
Алексеев Максим Петрович	Литература
Запись: 1 из 4	

Рис. 76. Результат выполнения многотабличного запроса

Таблицы в базе данных могут быть связаны между собой в окне **Схема данных** (меню **Сервис**). В этом случае при их добавлении в бланк запроса происходит автоматическое построение линии объединения. Если в окне **Схема данных** при определении связи между таблицами было также затребовано обеспечение целостности данных, то в бланке запроса такие таблицы будут автоматически связаны между собой линией, на концах которой будут отображаться цифра "1" и символ "∞" (бесконечность). Единица показывает, что таблица находится на стороне "один", а бесконечность – на стороне "много" в отношении "один-ко-многим" (рис. 77).

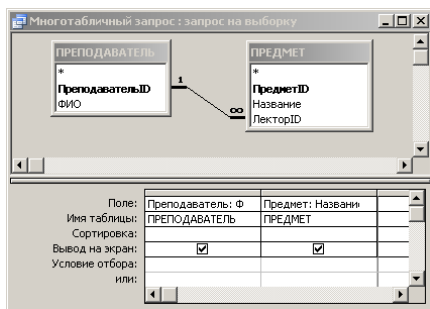


Рис. 77. Вид линии объединения в ситуации, когда между таблицами определена связь с контролем целостности данных

Автоматическое объединение добавляемых в бланк запроса таблиц происходит также в ситуациях, когда поля таблиц имеют одинаковые имена и типы данных или одно из полей является ключе-

вым, при этом иногда Access "ошибается". Чтобы удалить ошибочно добавленную линию объединения необходимо выделить ее одинарным щелчком левой кнопки мыши и нажать на клавиатуре клавишу **DELETE**.

Access поддерживает три типа объединения таблиц:

- *внутреннее объединение*,
- *левое внешнее объединение*,
- *правое внешнее объединение*.

Внутреннее объединение используется по умолчанию для вновь создаваемых связей и означает, что в запросе будут участвовать только те строки объединяемых таблиц, в которых значения связанных столбцов совпадают. В рассмотренном выше примере между таблицами по умолчанию установлено внутреннее объединение, поэтому в результирующую выборку попали только те преподаватели, которые читают лекции, и только те предметы, которые преподают; преподаватель Понтягин и предмет Философия в результатах отсутствуют.

Левое внешнее объединение применяется когда требуется извлечь все записи из таблицы со стороны "один" и те записи из таблицы со стороны "много" в отношении "один-ко-многим", в которых связанные поля совпадают.

Правое внешнее объединение является обратным по отношению к левому, и позволяет извлечь все записи из таблицы со стороны "много" и те записи из таблицы со стороны "один", в которых связанные поля совпадают.

Для построения левого или правого внешнего объединения необходимо вначале создать внутреннее объединение, затем щелкнуть правой кнопкой мыши на линии объединения, выбрать в появившемся контекстном меню команду **Параметры объединения**, после чего в открывшемся диалоговом окне (рис. 78) указать способ объединения таблиц и нажать кнопку **ОК**.

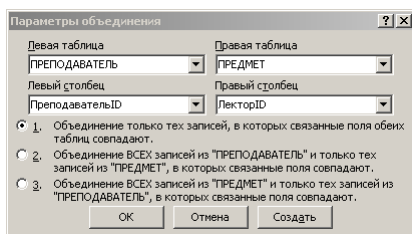


Рис. 78. Диалоговое окно **Параметры объединения** позволяет выбрать поля, по которым производится объединение таблиц и тип объединения

Пример бланка запроса с левым внешним объединением и результатом его выполнения (используются данные предыдущего примера, см. рис. 75) представлены на рис. 79. Запрос возвращает список всех преподавателей с указанием предметов, которые они читают. Обратите внимание, что в результирующий набор попал преподаватель Понтягин, который не ведет занятий ни по одному предмету.

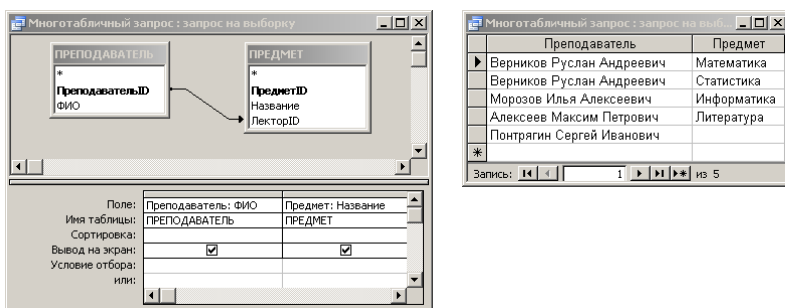


Рис. 79. Бланк запроса с левым внешним объединением и результат его выполнения

Пример бланка запроса с правым внешним объединением и результатом его выполнения (используются данные предыдущего примера, см. рис. 75) представлен на рис. 80. Запрос возвращает список всех предметов и фамилии преподавателей, которые ведут эти предметы. Обратите внимание, что предмет Философия не читает ни один преподаватель.

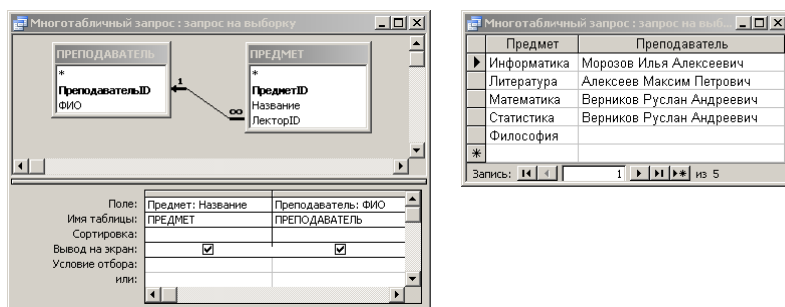


Рис. 80. Бланк запроса с правым внешним объединением и результат его выполнения

Удаление линии объединения в запросе (преднамеренно или случайно) приводит к построению так называемого *декартового произведения* – множества, элементами которого являются всевозможные сочетания элементов исходных множеств. На рис. 81 изображен бланк запроса, в котором отсутствует линия объединения, и результат его выполнения. Количество элементов (строк) в результирующем наборе равно $4 \times 5 = 20$ (четыре записи в таблице **ПРЕДМЕТ** и пять записей в таблице **ПРЕПОДАВАТЕЛЬ**). На практике такие запросы используются редко, т.к. результаты их, как правило, не представляют практического интереса.

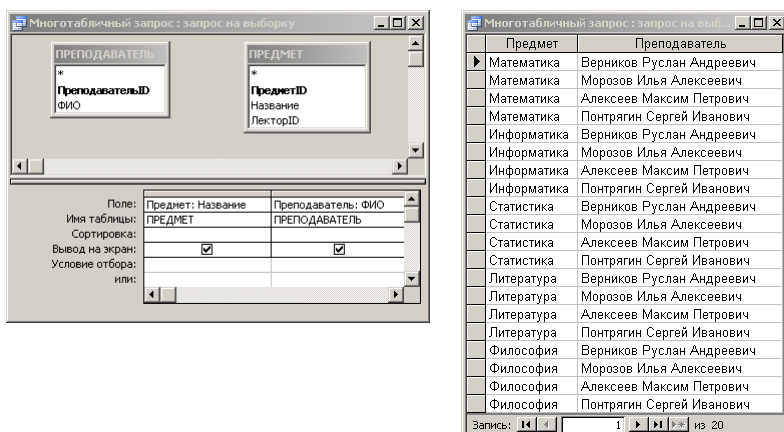


Рис. 81. Бланк запроса с отсутствующей линией объединения и результат его выполнения

Пример 6. Пусть в базе данных рассмотренного выше примера кроме таблиц **ПРЕПОДАВАТЕЛЬ** и **ПРЕДМЕТ** (см. рис. 75) имеется таблица **СТУДЕНТ**, хранящая имена студентов, обучающихся на факультете, и таблица **ОЦЕНКА**, хранящая результаты экзаменов. Содержимое этих таблиц изображено на рис. 82. В таблице **ОЦЕНКА** первичным ключом является пара атрибутов – **СтудентID** и **ПредметID**, – каждый из которых является внешним ключом: **СтудентID** ссылается на таблицу **СТУДЕНТ**, **ПредметID** – на таблицу **ПРЕДМЕТ**. Общая схема базы данных изображена на рис. 83.

СТУДЕНТ : таблица	
СтудентID	ФИО
1	Маслов Иван Петрович
2	Глухов Алексей Иванович
3	Рудин Илья Сергеевич
4	Быков Андрей Алексеевич
5	Ясницкая Ольга Анатольевна
6	Ростова Наталья Петровна
7	Старцев Петр Ильич
Записи: 1 из 7	

ОЦЕНКА : таблица		
СтудентID	ПредметID	Оценка
2	2	4
3	1	4
3	2	5
4	1	4
4	2	5
7	1	4
7	2	4
Записи: 1 из 7		

Рис. 82. Исходные данные для примера 6 (см. также рис. 75)

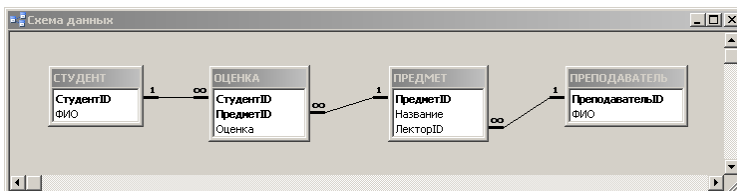


Рис. 83. Схема базы данных (пример 6)

Задача 6.1. Построить запрос, возвращающий список фамилий преподавателей, которым сдавал экзамен и получил оценку студент Рудин Илья Сергеевич.

Решение.

Поле: Преподаватель: ФИО
 Иная таблица: ПРЕПОДАВАТЕЛЬ
 Сортировка: ФИО
 Вывод на экран: ☒
 Условие отбора: "Рудин Илья Сергеевич"
 или:

Рис. 84. Запрос для решения задачи 6.1

Результат выполнения запроса.

Задача 6-1 : запрос на выборку	
Преподаватель	
Верников Руслан Андреевич	
Морозов Илья Алексеевич	
Записи: 1 из 2	

Рис. 85. Результат выполнения запроса (задача 6.1)

Комментарий. В этой задаче требуемые для решения данные расположены в таблицах **СТУДЕНТ** (требуется чтобы наложить условия отбора) и **ПРЕПОДАВАТЕЛЬ** (чтобы получить фамилии преподавателей). Таблицы не связаны между собой напрямую, поэтому для их объединения в запрос требуется добавить "связывающие" их таблицы **ОЦЕНКА** и **ПРЕДМЕТ** (см. схему данных на рис. 83).

Задача 6.2 Построить запрос, возвращающий список фамилий студентов, у которых принимал экзамен и поставил оценку преподаватель Морозов Илья Алексеевич.

Решение.

Поле:	Студент: ФИО	ФИО
Имя таблицы:	СТУДЕНТ	ПРЕПОДАВАТЕЛЬ
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	или:	"Морозов Илья Алексеевич"

Рис. 86. Запрос для решения задачи 6.2

Результат выполнения запроса.

Студент
Быков Андрей Алексеевич
Старцев Петр Ильич
Глухов Алексей Иванович
Рудин Илья Сергеевич

Записи: 1 из 4

Рис. 87. Результат выполнения запроса (задача 6.2)

Комментарий. Запрос аналогичен предыдущему. Изменились только поле, по которому необходимо производить фильтрацию, и поле, которое должно присутствовать в результирующем наборе.

Задача 6.3. Построить запрос, рассчитывающий для каждого обучающегося студента его средний балл, как среднее арифметическое всех его оценок.

Решение.

Задача 6-3: запрос на выборку

СТУДЕНТ

ОЦЕНКА

Поле:

Имя таблицы:

Групповая операция:

Сортировка:

Вывод на экран:

Условие отбора:

или:

Рис. 88. Запрос для решения задачи 6.3

Результат выполнения запроса.

Задача 6-3: запрос на выборку

Студент	Средний балл
Быков Андрей Алексеевич	4,5
Глухов Алексей Иванович	4
Маслов Иван Петрович	
Ростова Наталья Петровна	
Рудин Илья Сергеевич	4,5
Старцев Петр Ильич	4
Ясницкая Ольга Анатольевна	

Запись: 1 из 7

Рис. 89. Результат выполнения запроса (задача 6.3)

Комментарий. Исходные данные для этой задачи находятся в таблицах **СТУДЕНТ** и **ОЦЕНКА**. Для получения списка всех студентов используется левое объединение. Для расчета среднего значения используется статистическая функция **AVG**.

Задача 6.4. Построить запрос, возвращающий список предметов, по которым не было сдано ни одного экзамена (нет оценок).

Решение.

Задача 6-4: запрос на выборку

ПРЕДМЕТ

ОЦЕНКА

Поле:

Имя таблицы:

Групповая операция:

Сортировка:

Вывод на экран:

Условие отбора:

или:

Рис. 90. Запрос для решения задачи 6.4

Результат выполнения запроса.

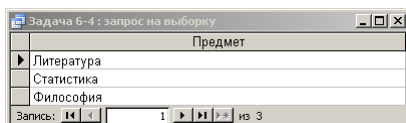


Рис. 91. Результат выполнения запроса (задача 6.4)

Комментарий. Данные для решения этой задачи находятся в таблицах **ПРЕДМЕТ** и **ОЦЕНКА**. Для определения количества оценок используется статистическая функция **COUNT**. Нюанс задачи заключается в том, что количество оценок требуется рассчитать в том числе и для предметов, по которым не было сдано ни одного экзамена, чтобы оставить их в списке, наложив соответствующее условие отбора. Для этого в запросе используется левое внешнее объединение.

Задача 6.5. Построить запрос, возвращающий список всех студентов с оценками и названиями предметов (студенты, не имеющие оценок, также должны присутствовать в списке).

Решение.

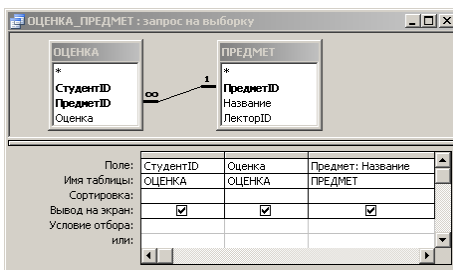


Рис. 92. Вспомогательный запрос для решения задачи 6.5

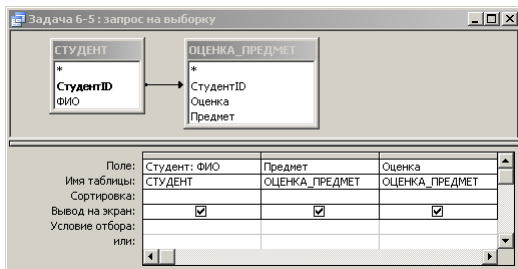


Рис. 93. Основной запрос для решения задачи 6.5

Результат выполнения запроса.

СтудентID	Оценка	Предмет
3	4	Математика
7	4	Математика
4	4	Математика
4	5	Информатика
7	4	Информатика
2	4	Информатика
3	5	Информатика

Рис. 94. Результат выполнения вспомогательного запроса (задача 6.5)

Студент	Предмет	Оценка
Маслов Иван Петрович	Информатика	4
Глухов Алексей Иванович	Информатика	5
Рудин Илья Сергеевич	Математика	4
Рудин Илья Сергеевич	Математика	4
Быков Андрей Алексеевич	Математика	4
Быков Андрей Алексеевич	Информатика	5
Ясницкая Ольга Анатольевна		
Ростова Наталья Петровна		
Старцев Петр Ильич	Математика	4
Старцев Петр Ильич	Информатика	4

Рис. 95. Результат выполнения основного запроса (задача 6.5)

Комментарий. Очевидное на первый взгляд решение этой задачи, содержащее внутреннее и внешнее объединения, представлено на рис. 96. К сожалению, Access не поддерживает такие запросы и при попытке их выполнения выводит на экран сообщение об ошибке (рис. 97). Для решения задачи необходимо создать вспомогательный запрос, в котором реализуется одно из объединений, например, объединение данных таблиц **ОЦЕНКА** и **ПРЕДМЕТ** (см. рис. 92), после чего создается основной запрос (см. рис. 93), в котором реализуется второе объединение, но уже с данными вспомогательного запроса. Результаты вспомогательного и основного запроса представлены на рис. 94 и рис. 95.

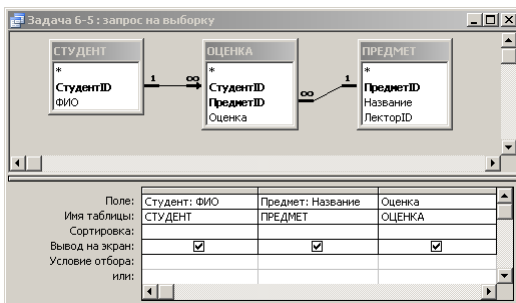


Рис. 96. Запрос с внешним и внутренним объединением (неверное решение)

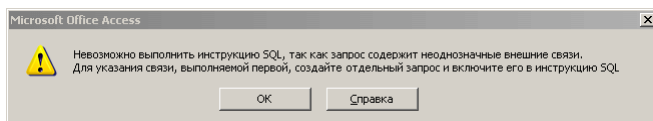


Рис. 97. Сообщение об ошибке выполнения запроса, содержащего внешнее и внутреннее объединения

Перекрестный запрос

Перекрестный запрос – это специальный вид агрегированного запроса, в котором после выполнения всех вычислений производится дополнительное разбиение строк на два набора и затем формируется результирующая таблица, в которой слева располагаются значения одного набора, формируя заголовки строк, сверху – значения другого набора, формируя заголовки столбцов, и в центре – результаты вычислений. Благодаря такой структуре таблица с результатами перекрестного запроса оказывается гораздо проще для восприятия, чем результаты обычного агрегированного запроса. Так на рис. 98 изображены одни и те же данные полученные с помощью обычного агрегированного запроса (вверху) и с помощью перекрестного запроса (внизу). Очевидно, что второй вариант дает более наглядную картину ситуации.

	ФИО	Название	Оценка
▶	Глухов Сергей Сергеевич	Информатика	5
	Глухов Сергей Сергеевич	Математика	4
	Иванов Алексей Иванович	Информатика	4
	Иванов Алексей Иванович	Математика	4
	Кудряшов Максим Андреевич	Литература	4
	Кудряшов Максим Андреевич	Статистика	5
	Литвиненко Ольга Сергеевна	Информатика	5
	Литвиненко Ольга Сергеевна	Математика	4
	Малышева Мария Андреевна	Философия	3
	Пестов Сергей Ильич	Информатика	4
Записи: 1 из 10			

	ФИО	Информатика	Литература	Математика	Статистика	Философия
▶	Глухов Сергей Сергеевич	5		4		
	Иванов Алексей Иванович	4		4		
	Кудряшов Максим Андреевич		4		5	
	Литвиненко Ольга Сергеевна	5		4		
	Малышева Мария Андреевна					3
	Пестов Сергей Ильич	4				
Записи: 1 из 6						

Рис. 98. Результат агрегированного (вверху) и перекрестного (внизу) запросов, построенных по одним и тем же данным

Для построения перекрестного запроса необходимо открыть конструктор запросов, после чего сменить тип запроса на перекрестный одноименной командой в меню **Запрос**. В результате в блан-

ке запроса появятся две строки: **Групповая операция** и **Перекрестная таблица** (рис. 99).

Рис. 99. Выпадающий список строки **Перекрестная таблица** позволяет указать местоположение значений выбранных столбцов

Назначение строки **Групповая операция** осталась прежним (см. раздел "Агрегированные запросы"). Выпадающий список в строке **Перекрестная таблица** позволяет указать для каждого столбца расположение в результирующей таблице: слева (**Заголовки строк**), сверху (**Заголовки столбцов**), в центре (**Значение**); или скрыть значения столбца в результатах.

Замечание. В запросе может быть только один столбец для заголовков столбцов и только один столбец для значений.

Пример 7. База данных (файл **Пример 7.mdb**) содержит экзаменационные оценки студентов по различным предметам в четырех таблицах: **ГРУППА**, **СТУДЕНТ**, **ПРЕДМЕТ** и **ОЦЕНКА**. Схема данных базы изображена на рис. 100, содержимое таблиц представлено на рис. 101.

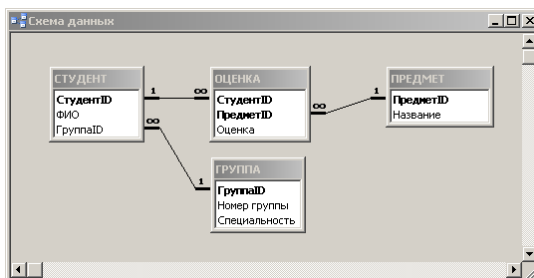


Рис. 100. Схема данных (пример 7)

ГРУППА : таблица		
ГруппаID	Номер группы	Специальность
1	101	Менеджмент
2	102	Мировая экономика
3	103	Бухгалтерский учет
Запись: 1 из 3		

СТУДЕНТ : таблица		
СтудентID	ФИО	ГруппаID
1	Кудряшов Максим Андреевич	1
2	Пестов Сергей Ильич	1
3	Литвиненко Ольга Сергеевна	3
4	Глухов Сергей Сергеевич	2
5	Малышева Мария Андреевна	1
6	Минаев Сергей Петрович	3
7	Иванов Алексей Иванович	1
Запись: 1 из 7		

ПРЕДМЕТ : таблица	
ПредметID	Название
1	Математика
2	Информатика
3	Статистика
4	Литература
5	Философия
Запись: 1	

ОЦЕНКА : таблица		
СтудентID	ПредметID	Оценка
1	3	4
3	4	4
2	2	5
1	1	4
3	2	5
4	1	3
4	2	5
5	5	3
7	1	4
7	3	3
1	2	4
1	5	5
2	5	5
7	5	3
4	4	3
2	3	3
2	4	3
2	1	4
5	3	3
6	5	4
6	3	5
7	2	3
6	1	5
5	4	5
7	4	4
Запись: 1 из 25		

Рис. 101. Исходные данные (пример 7)

Задача 7.1. Построить перекрестный запрос, возвращающий результаты экзаменов студентов.

Решение.

Задача 7-1 : перекрестный запрос		
СТУДЕНТ * СтудентID ФИО ГруппаID	∞	ОЦЕНКА * СтудентID ПредметID Оценка
	∞	ПРЕДМЕТ * ПредметID Название
Поле: Студент: ФИО Название Оценка Имя таблицы: СТУДЕНТ ПРЕДМЕТ ОЦЕНКА Групповая операция: Группировка Группировка Sum Перекрестная таблица: Заголовки строк Заголовки столбцов Значение Сортировка: Условие отбора: или:		

Рис. 102. Запрос для решения задачи 7.1

Результат выполнения запроса.

Задача 7-1 : перекрестный запрос					
Студент	Информатика	Литература	Математика	Статистика	Философия
Глухов Сергей Сергеевич	5	3	3		
Иванов Алексей Иванович	3	4	4	3	3
Кудряшов Максим Андреевич	4		4	4	5
Литвиненко Ольга Сергеевна	5	4			
Малышева Мария Андреевна		5		3	3
Минаев Сергей Петрович			5	5	4
Пестов Сергей Ильич	5	3	4	3	5
Запись: 1 из 7					

Рис. 103. Результат выполнения запроса (задача 7.1)

Комментарий. Для решения задачи необходимы данные из таблиц **СТУДЕНТ**, **ПРЕДМЕТ** и **ОЦЕНКА**. В бланке запроса указанные таблицы объединяются согласно схеме данных, в результате чего получается одна таблица, содержащая имя студента, название предмета и его экзаменационную оценку по этому предмету. Строки полученной таблицы группируются по столбцам **ФИО** и **Предмет**, а для столбца **Оценка** вычисляется значение функции **SUM**. Так как в таблице **ОЦЕНКА** столбцы **СтудентID** и **ПредметID** являются первичным ключом, то можно быть уверенным, что функция **SUM** будет применяться только к одной оценке.

Замечание. Рассмотренный вариант решения строит таблицу в которую попадают студенты, имеющие хотя бы одну оценку, и предметы, по которым сдан хотя бы один экзамен. Студенты, не сдавшие ни одного экзамена и предмет, по которому не было сдано ни одного экзамена, в результирующий набор не попадают. Для устранения этого недостатка необходимо использовать внешние объединения. Решение, получаемое в этом случае достаточно громоздко, и здесь не приводится.

Задача 7.2. Построить перекрестный запрос, рассчитывающий средний балл по каждому предмету для всех специальностей.

Решение.

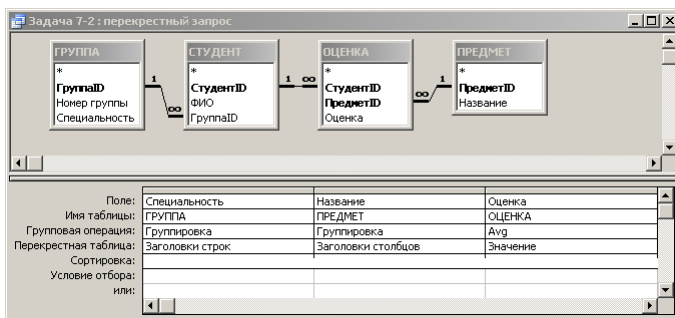


Рис. 104. Запрос для решения задачи 7.2

Результат выполнения запроса.

Специальность	Информатика	Литература	Математика	Статистика	Философия
Бухгалтерский учет	5	4	5	5	4
Менеджмент	4	4	4	3,25	4
Мировая экономика	5	3	3		

Записи: 1 из 3

Рис. 105. Результат выполнения запроса (задача 7.2)

Запрос на создание таблицы

Запрос на создание таблицы позволяет извлечь требуемые данные из одной или нескольких таблиц и поместить их в новую таблицу. Для его построения необходимо открыть конструктор запросов и дать команду **Создание таблицы...** (меню **Запрос**). На экране появится диалоговое окно **Создание таблицы** (рис. 106), в котором необходимо ввести имя для создаваемой таблицы и определиться с местом ее хранения – в текущей или в другой базе данных (в последнем случае потребуется указать файл базы данных). В остальной работе с запросом ничем не отличается от работы по созданию запроса на выборку (допускается наложение условий отбора, создание вычисляемых полей и применение агрегирующих функций).

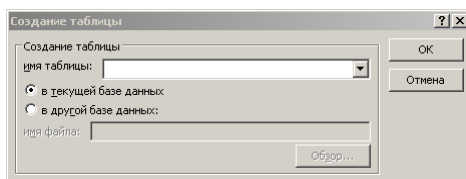


Рис. 106. Окно **Создание таблицы**

Поле **имя таблицы** (см. рис. 106) представляет собой выпадающий список, содержащий имена всех таблиц выбранной базы данных. Если имя создаваемой таблицы совпадает с уже существующей таблицей, то перед выполнением запроса на экране появится запрос на подтверждение удаления существующей таблицы (рис. 107). Положительный ответ на него приведет к удалению из базы данных существующей таблицы, созданию новой с таким же именем и наложению ее данными запроса.

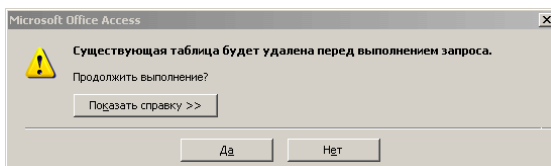


Рис. 107. Запрос на подтверждение удаления существующей таблицы

Замечание. Выполнение запроса на создание таблицы ведет к необратимым изменениям в базе данных. В частности, восстановить удаленную таблицу невозможно. Поэтому настоятельно рекомендуется перед выполнением запроса на создание таблицы создавать резервную копию удаляемой таблицы.

Пример 8. База данных (файл **Пример 8.mdb**) содержит три таблицы **ГРУППА**, **СТУДЕНТ** и **ПРЕПОДАВАТЕЛЬ** (рис. 108). Схема данных и содержимое таблиц изображены на рис. 109 и на рис. 110.

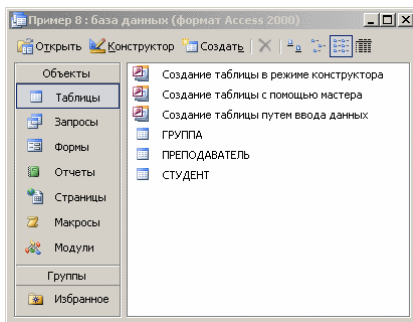


Рис. 108. Окно базы данных (пример 8)

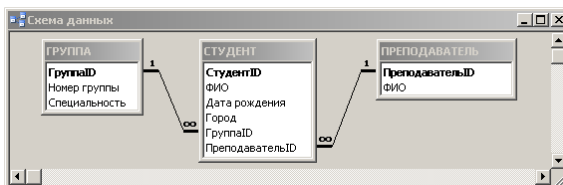


Рис. 109. Схема данных (пример 8)

ГруппаID	Номер группы	Специальность
1	101	Менеджмент
2	102	Мировая экономика
3	103	Бухгалтерский учет
4	104	Налоги и налогообложение
Записи: 1 из 4		

ПреподавательID	ФИО
1	Рябова Татьяна Яковлевна
2	Лукашов Олег Николаевич
3	Шерина Наталья Васильевна
4	Дмитриев Леонид Осипович
Записи: 1 из 4	

СтудентID	ФИО	Дата рождения	Город	ГруппаID	ПреподавательID
1	Чернова Татьяна Александровна	05.08.1988	Томск	1	1
2	Быкова Марина Александровна	25.11.1989	Асино	2	2
3	Лобанов Артур Сергеевич	16.10.1987	Стрежевой	1	1
4	Ворошкович Виктор Иванович	24.02.1989	Барнаул	2	2
5	Галахова Арина Дмитриевна	24.11.1989	Томск	3	2
6	Конев Артем Сергеевич	25.09.1987	Асино	3	1
7	Медведев Евгений Владимирович	10.07.1987	Томск	1	3
Записи: 1 из 7					

Рис. 110. Исходные данные (пример 8)

Задача 8.1. Создать таблицу **ГРУППА 101**, содержащую записи обо всех студентах группы 101. Таблица должны содержать поля: **Студент**, **Дата рождения**, **Город** и **Научный руководитель**.

Решение.

Поле:	Студент: ФИО	Дата рождения	Город	Научный руководитель: ФИО	Номер группы
Имя таблицы:	СТУДЕНТ	СТУДЕНТ	СТУДЕНТ	ПРЕПОДАВАТЕЛЬ	ГРУППА
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:					101
или:					

Рис. 111. Запрос для решения задачи 8.1

Результат выполнения запроса.

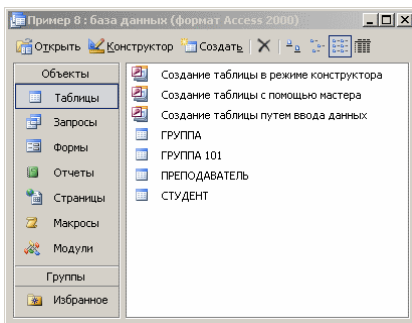


Рис. 112. Результат выполнения запроса (задача 8.1)

Студент	Научный руководитель	Дата рождения	Город
Чернова Татьяна Александровна	Рябова Татьяна Яковлевна	05.08.1988	Томск
Лобанов Артур Сергеевич	Рябова Татьяна Яковлевна	16.10.1987	Стрежевой
Медведев Евгений Владимирович	Шерина Наталья Васильевна	10.07.1987	Томск

Рис. 113. Содержимое таблицы **ГРУППА 101** после выполнения запроса

Комментарий. Требуемые для решения задачи данные находят-ся в таблицах **ГРУППА**, **СТУДЕНТ** и **ПРЕПОДАВАТЕЛЬ**, в бланке запроса они объединяются согласно схеме данных. Чтобы в создаваемую таблицу попали только записи о студентах из группы 101, в бланк запроса добавляется поле **Номер группы**, на которое

накладывается соответствующее условие отбора, и т.к. самого поля в создаваемой таблице быть не должно, переключатель в строке **Вывод на экран** устанавливается в положение "выключено".

Запрос на добавление данных

Запрос на добавление данных позволяет добавить к существующей таблице данные, извлеченные из других таблиц или запросов. Таблица, в которую производится добавление данных – *таблица-получатель*, может располагаться как в текущей, так и в другой базе данных.

Для создания запроса на добавление необходимо открыть конструктор запросов, после чего дать команду **Добавление** (меню **Запрос**). На экране (рис. 114) появится окно **Добавление**, в котором необходимо в выпадающем списке **имя таблицы** выбрать таблицу-получатель (если таблица расположена в другой базе данных, потребуется ввести имя файла базы данных).

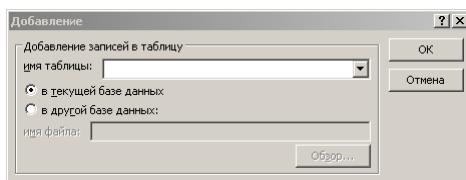


Рис. 114. Окно **Добавление**

После указания таблицы-получателя в окне **Добавление** и нажатия кнопки **ОК** на экране появляется бланк запроса, содержащий строку **Добавление** предназначенную для выбора полей в таблице-получателе, в которые требуется поместить значения полей результирующего набора запроса. В остальном работа с запросом на добавление ничем не отличается от работы по созданию запроса на выборку (допускается наложение условий отбора, создание вычисляемых полей и применение агрегирующих функций).

Замечание. Запрос на добавление является запросом на изменение. Его выполнение ведет к необратимым изменениям в таблице-получателе, о чем предупреждает окно-запрос на продолжение операции, появляющееся на экране в момент выполнения запроса (рис. 115). Так как отменить последствия операции добавления невозможно, настоятельно рекомендуется перед выполнением запроса переключаться в табличный режим просмотра, тщательно изучать результирующий набор на предмет наличия ошибок и, только убедившись в их отсутствии, запускать запрос на выполнение. Также

рекомендуется перед выполнением запроса создавать резервную копию таблицы-получателя.

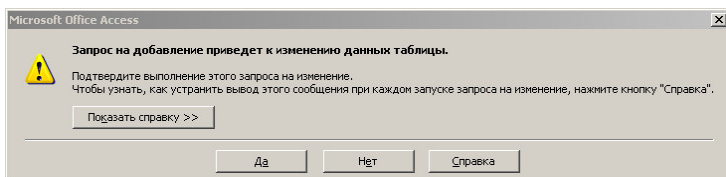


Рис. 115. Запрос подтверждения операции добавления

Пример 9. База данных (файл **Пример 9.mdb**) содержит таблицы: **СТУДЕНТ**, **ПРЕДМЕТ**, **ОЦЕНКА**. Схема данных и содержимое таблиц изображены на рис. 116 и рис. 117. В базе также имеется таблица **ОТЛИЧНИКИ**, структура которой и содержимое изображены на рис. 118 и рис. 119.

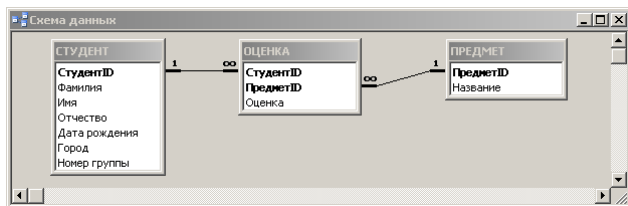


Рис. 116. Схема данных (пример 9)

ПредметID	Название
1	Математика
2	Информатика
3	Статистика
4	Философия
5	Культурология
6	История
7	Менеджмент

СтудентID	ПредметID	Оценка
1	2	5
4	3	4
3	1	5
1	3	3
7	3	5
2	3	4
2	6	5

СтудентID	Фамилия	Имя	Отчество	Дата рождения	Город	Номер группы
1	Полякова	Галина	Васильевна	27.08.1988	Томск	102
2	Иванов	Иван	Иванович	12.09.1990	Абакан	103
3	Круглов	Алексей	Матвеевич	16.09.1986	Кемерово	101
4	Родионова	Ольга	Владимировна	01.03.1987	Асино	104
5	Панкова	Татьяна	Ивановна	11.04.1989	Новосибирск	103
6	Игнатьева	Валентина	Александровна	18.02.1988	Стрежевой	101
7	Косинок	Надежда	Михайловна	19.10.1988	Томск	102

Рис. 117. Исходные данные (пример 9)

Рис. 118. Структура таблицы **ОТЛИЧНИКИ** (пример 9)

Студент	Группа	Предмет
Шумар Андрей Иванович	101	Математика
Николаев Игорь Васильевич	102	Информатика
Вережagina Ольга Андреевна	101	Менеджмент

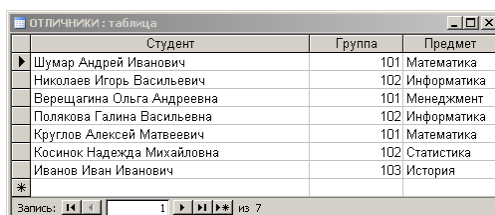
Рис. 119. Содержимое таблицы **ОТЛИЧНИКИ** (пример 9)

Задача 9.1. Создать запрос, добавляющий в таблицу **ОТЛИЧНИКИ** данные обо всех студентах, имеющих оценку "отлично" (5). Фамилия, имя и отчество студента должны быть объединены и помещены в столбец **Студент**, номер группы, в которой учится студент, помещен в столбец **Группа**, а название предмета – в столбец **Предмет**.

Решение.

Рис. 120. Запрос для решения задачи 9.1

Результат выполнения запроса.



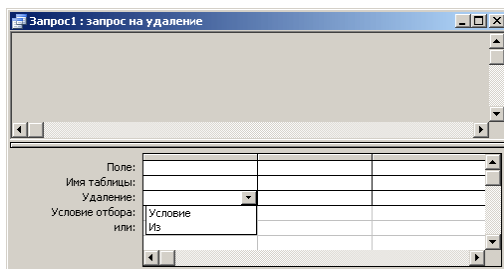
Студент	Группа	Предмет
Шумар Андрей Иванович	101	Математика
Николаев Игорь Васильевич	102	Информатика
Верещагина Ольга Андреевна	101	Менеджмент
Полякова Галина Васильевна	102	Информатика
Круглов Алексей Матвеевич	101	Математика
Косинок Надежда Михайловна	102	Статистика
Иванов Иван Иванович	103	История

Рис. 121. Содержимое таблицы **ОТЛИЧНИКИ** после выполнения запроса

Запрос на удаление данных

Запрос на удаление позволяет произвести удаление записей, удовлетворяющих заданным условиям, из одной или нескольких таблиц.

Для создания запроса на удаление необходимо открыть конструктор запросов, после чего дать команду **Удаление** (меню **Запрос**), при этом в бланке конструктора запросов исчезнут строки **Сортировка** и **Вывод на экран** и появится строка **Удаление** (рис. 122).



Поле:	Иная таблица:	Удаление:	Условие отбора:	или:
		Условие		
		Из		

Рис. 122. Выпадающий список в строке **Удаление** позволяет указать столбцы, используемые для формирования условий отбора удаляемых записей

Выпадающий список в строке **Удаление** содержит два значения:

- **Условие** – позволяет наложить условия отбора на удаляемые записи. Данное значение доступно для любых полей таблиц, участвующих в запросе, кроме поля * (звездочка).
- **Из** – предназначено для указания таблицы, из которой будет производиться удаление записей. Данное значение доступно только для поля * (звездочка).

Основным нюансом при создании запроса на удаление является необходимость учитывать то, в каких отношениях находится таблица, из которой будут удаляться записи, с другими таблицами базы

данных. Обусловлено это необходимостью поддержания в базе логической целостности (непротиворечивости) данных.

Назовем таблицу, состоящую в отношении "один-ко-многим" с другой таблицей на стороне "один", – *родительской*, а таблицу, на стороне "многo" – *дочерней*.

Удаление записей из таблицы, которая не состоит ни в каких отношениях с другими таблицами базы данных или является дочерней таблицей, не приводит к нарушениям целостности данных, поэтому никаких дополнительных действий не требует.

Удаление записей из таблицы, которая хотя бы в одном из отношений является родительской, может привести к появлению в дочерней таблице "потерянных" данных – записей, у которых внешний ключ ссылается на несуществующие данные в родительской таблице. Во избежание этого необходимо перед удалением данных из родительской таблицы, удалить связанные данные из дочерней таблицы. Не выполнение этого требования приводит к нарушению целостности данных, а т.к. пользователь может легко допустить такую ошибку, то на практике обычно ответственность за поддержание логической целостности данных возлагают на Access. Здесь возможны две ситуации:

- При создании связи между таблицами был затребован только контроль целостности. В этом случае при попытке удаления записей из родительской таблицы, у которых имеются связанные записи в дочерней таблице, Access заблокирует операцию и выведет на экран информационное сообщение о наличии связанных данных в подчиненной таблице.
- При создании связи между таблицами был затребован контроль целостности и включено каскадное удаление связанных записей. В этом случае Access выполнит операцию удаления, причем вместе с записями родительской таблицы будут удалены связанные с ними записи из дочерней таблицы.

Замечание. Запрос на удаление, как и запрос на добавление, является запросом на изменение и приводит к необратимым изменениям в таблицах базы данных. Так как отменить последствия операции удаления невозможно, настоятельно рекомендуется перед выполнением запроса переключаться в табличный режим просмотра, тщательно изучать результирующий набор на предмет наличия ошибок и, только убедившись в их отсутствии, запускать запрос на выполнение. Также рекомендуется создавать резервную копию таблиц, из которых производится удаление данных.

Пример 10. База данных (файл **Пример 10.mdb**) содержит таблицы: **ГРУППА**, **ПРЕДМЕТ**, **СТУДЕНТ**, **ОЦЕНКА**. Схема данных изображена на рис. 123, содержимое таблиц – на рис. 124. Для каждой связи между таблицами затребовано обеспечение целостности данных и каскадное удаление связанных записей.

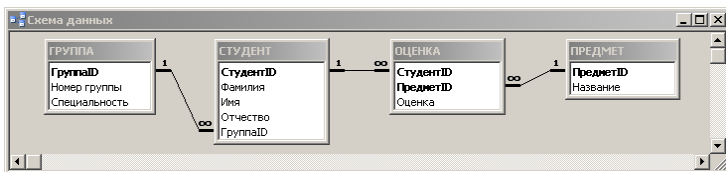


Рис. 123. Схема данных (пример 10)

ГруппID	Номер группы	Специальность
1	101	Менеджмент
2	102	Мировая экономика
3	103	Бухгалтерский учет

Запись: 1 из 3

ПредметID	Название
1	Математика
2	Информатика
3	Статистика

Запись: 1 из 3

СтудентID	Фамилия	Имя	Отчество	ГруппID
1	Ромашова	Оксана	Николаевна	1
2	Иванов	Иван	Иванович	2
3	Родионова	Ольга	Владимировна	2
4	Королев	Максим	Петрович	3
5	Пригожин	Сергей	Владимирович	3
6	Петрова	Ксения	Ивановна	2
7	Уфимцев	Владислав	Петрович	1

Запись: 1 из 7

СтудентID	ПредметID	Оценка
1	2	3
2	2	5
4	2	4
4	3	3
5	1	3
7	1	4
7	3	5

Запись: 1 из 7

Рис. 124. Исходные данные (пример 10)

Задача 10.1. Создать запрос, удаляющий из базы данных все оценки по математике.

Решение.

Задача 10-1 : запрос на удаление

ОЦЕНКА: СтудентID, ПредметID, Оценка

ПРЕДМЕТ: ПредметID, Название

Поле: ОЦЕНКА, *

Имя таблицы: ОЦЕНКА

Удаление: Из: ПРЕДМЕТ

Условие отбора: Условие: "математика"

или:

Рис. 125. Запрос для решения задачи 10.1

Результат выполнения запроса.

СтудентID	ПредметID	Оценка
1	2	3
2	2	5
4	2	4
4	3	3
7	3	5

Рис. 126. Результат выполнения запроса (задача 10.1)

Комментарий. Для решения задачи в бланк запроса необходимо добавить таблицу **ОЦЕНКА**, т.к. ее данные подлежат удалению, и таблицу **ПРЕДМЕТ**, которая требуется для фильтрации данных.

Задача 10.2. Создать запрос, удаляющий из базы студента Королева (удалить все данные, относящиеся к студенту Королеву).

Решение.

Поле: Фамилия

Имя таблицы: СТУДЕНТ

Удаление: Условие

Условие отбора: "Королев"

или:

Рис. 127. Запрос для решения задачи 10.2

Результат выполнения запроса.

СтудентID	Фамилия	Имя	Отчество	ГруппaID
1	Ромашова	Оксана	Николаевна	1
2	Иванов	Иван	Иванович	2
3	Родионова	Ольга	Владимировна	2
5	Пригожин	Сергей	Владимирович	3
6	Петрова	Ксения	Ивановна	2
7	Уфимцев	Владислав	Петрович	1

СтудентID	ПредметID	Оценка
1	2	3
2	2	5
7	3	5

Рис. 128. Результат выполнения запроса (задача 10.2)

Комментарий. Так как таблица **СТУДЕНТ** является родительской по отношению к таблице **ОЦЕНКА** и для связи между ними затребовано обеспечение целостности данных и каскадное удаление

связанных записей, то для решения задачи достаточно удалить в таблице **СТУДЕНТ** запись, относящуюся к студенту Королеву, оценки его будут удалены автоматически.

Задача 10.3. Создать запрос, удаляющий из базы запись о группе 101 и все данные, относящиеся к ней.

Решение.

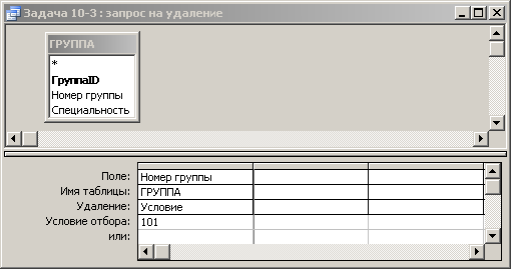


Рис. 129. Запрос для решения задачи 10.3

Результат выполнения запроса.

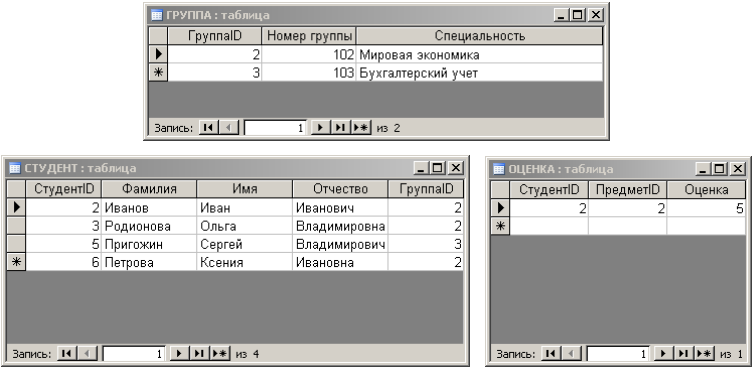


Рис. 130. Результат выполнения запроса (задача 10.3)

Комментарий. Так как таблица **ГРУППА** является родительской по отношению к таблице **СТУДЕНТ**, которая в свою очередь является родительской по отношению к таблице **ОЦЕНКА** и для всех межтабличных связей затребовано обеспечение целостности данных с каскадным удалением связанных записей, то для решения задачи достаточно удалить одну запись в таблице **ГРУППА**, относящуюся к группе 101. Записи о студентах группы и их оценки будут удалены автоматически.

Запрос на обновление данных

Запрос на обновление данных позволяет изменить значения в одном или нескольких столбцах таблицы сразу для множества записей, удовлетворяющих заданным условиям. Для создания запроса на обновление необходимо открыть конструктор запросов, после чего дать команду **Обновление** (меню **Запрос**). В бланке запроса появится строка **Обновление**, предназначенная для ввода выражений обновляемых полей (рис. 131). Назначение остальных строк бланка запроса осталось прежним.

Поле:	Стипендия			
Имя таблицы:	СТУДЕНТ			
Обновление:	100			
Условие отбора:				
или:				

Рис. 131. В данном запросе на обновление всем студентам назначается стипендия в размере 100 руб.

Выражение, как и при создании запросов с вычисляемыми полями, представляет собой комбинацию математических и логических операторов, имен полей, констант и функций, в результате обработки которой получается единственное значение (результат). Символьные константы в выражениях требуется заключать в двойные кавычки ("); названия полей – в квадратные скобки ([,]); даты обрамлять символом решетки (#). Порядок вычислений выражения определяется приоритетом операций (см. справку по Access) и может быть изменен с помощью скобок.

Замечание 1. Изменения, произошедшие в таблице в результате выполнения запроса на обновление, отменить невозможно, поэтому настоятельно рекомендуется перед выполнением запроса переключаться в табличный режим просмотра, тщательно изучать результирующий набор на предмет наличия ошибок и, только убедившись в их отсутствии, запускать запрос на выполнение. Также рекомендуется создавать резервную копию обновляемой таблицы.

Замечание 2. Назначением запросов на обновление является изменение имеющихся в таблице данных, поэтому такие запросы нельзя применить для обновления вычисляемых и итоговых полей. Также нельзя обновить значения полей с типом данных **Счетчик**.

Пример 11. База данных (файл **Пример 11.mdb**) содержит две таблицы: **СТУДЕНТ** и **СТУДЕНТ2** (рис. 132).

СтудентID	ФИО	Телефон	Стипендия	Word	Excel
1	Алексеева Марина Юрьевна		100	зачет	зачет
2	Нестерова Ольга Анатольевна		100	незачет	
3	Сухов Иван Иванович		100		зачет
4	Котов Евгений Владимирович		100	незачет	
5	Маслова Ольга Ивановна		100	зачет	зачет
6	Зверев Игорь Сергеевич		100	незачет	незачет
7	Корнеев Иван Петрович		100		зачет

СтудентID	ФИО	Телефон
1	Алексеева Марина Юрьевна	7-787-876
2	Нестерова Ольга Анатольевна	8-878-657
3	Сухов Иван Иванович	9-987-767
4	Котов Евгений Владимирович	3-293-475
5	Маслова Ольга Ивановна	1-235-987
6	Зверев Игорь Сергеевич	2-244-888
7	Корнеев Иван Петрович	4-444-444

Рис. 132. Исходные данные (пример 11)

Задача 11.1. Увеличить всем студентам стипендию на 100 руб.

Решение.

Задача 11-1 : запрос на обновление

Таблица: **СТУДЕНТ**

Поле: Стипендия

Иная таблица: СТУДЕНТ

Обновление: [Стипендия]+100

Условие отбора:

или:

Рис. 133. Запрос для решения задачи 11.1

Результат выполнения запроса.

СтудентID	ФИО	Телефон	Стипендия	Word	Excel
1	Алексеева Марина Юрьевна		200	зачет	зачет
2	Нестерова Ольга Анатольевна		200	незачет	
3	Сухов Иван Иванович		200		зачет
4	Котов Евгений Владимирович		200	незачет	
5	Маслова Ольга Ивановна		200	зачет	зачет
6	Зверев Игорь Сергеевич		200	незачет	незачет
7	Корнеев Иван Петрович		200		зачет

Рис. 134. Результат выполнения запроса (задача 11.1)

Задача 11.2. Студентам, имеющим два зачета, увеличить стипендию на 100 руб.

Решение.

Задача 11-2: запрос на обновление

СТУДЕНТ

ФИО
Телефон
Стипендия
Word
Excel

Поле: Стипендия Word Excel
Имя таблицы: СТУДЕНТ СТУДЕНТ СТУДЕНТ
Обновление: [Стипендия]+100
Условие отбора: "зачет" "зачет"
или:

Рис. 135. Запрос для решения задачи 11.2

Результат выполнения запроса.

СтудентID	ФИО	Телефон	Стипендия	Word	Excel
1	Алексеева Марина Юрьевна		300	зачет	зачет
2	Нестерова Ольга Анатольевна		200	незачет	
3	Сухов Иван Иванович		200		зачет
4	Котов Евгений Владимирович		200	незачет	
5	Маслова Ольга Ивановна		300	зачет	зачет
6	Зверев Игорь Сергеевич		200	незачет	незачет
7	Корнеев Иван Петрович		200		зачет

Запись: 1 из 7

Рис. 136. Результат выполнения запроса (задача 11.2)

Задача 11.3. Обновить данные о телефонах студентов в таблице СТУДЕНТ данными из таблицы СТУДЕНТ2.

Решение.

Задача 11-3: запрос на обновление

СТУДЕНТ

СтудентID
ФИО
Телефон
Стипендия

СТУДЕНТ2

СтудентID
ФИО
Телефон

Поле: Телефон
Имя таблицы: СТУДЕНТ
Обновление: СТУДЕНТ2.Телефон
Условие отбора:
или:

Рис. 137. Запрос для решения задачи 11.3

Результат выполнения запроса.



СтудентID	ФИО	Телефон	Стипендия	Word	Excel
1	Алексеева Марина Юрьевна	7-787-876	300	зачет	зачет
2	Нестерова Ольга Анатольевна	8-878-657	200	незачет	
3	Сухов Иван Иванович	9-987-767	200		зачет
4	Котов Евгений Владимирович	3-293-475	200	незачет	
5	Маслова Ольга Ивановна	1-235-987	300	зачет	зачет
6	Зверев Игорь Сергеевич	2-244-888	200	незачет	незачет
7	Корнеев Иван Петрович	4-444-444	200		зачет

Рис. 138. Результат выполнения запроса (задача 11.3)

Комментарий. Для решения данной задачи требуется данные столбца **Телефон** таблицы **СТУДЕНТ2** поместить в одноименный столбец таблицы **СТУДЕНТ**. Для этого таблицы объединяются по первичному ключу **СтудентID**. Нюансом задачи является то, что столбцы в обеих таблицах имеют одинаковые имена, в результате чего в выражении для обновления возникает *конфликт имен* – ситуация, в которой по имени столбца не удастся установить какой таблице он принадлежит. Для устранения конфликта достаточно записать полное имя столбца, включающее в себя название таблицы, имя столбца и точку между ними в качестве разделителя.

Запрос на объединение

*Запрос на объединение*¹ позволяет объединить в результатах запроса данные из не связанных между собой таблиц или запросов, имеющих похожую структуру.

Запросы на объединение в отличие от рассмотренных выше запросов создаются только с помощью языка SQL. После открытия окна конструктора запросов и смены типа запроса (команда **Запрос SQL → Объединение** в меню **Запрос**) происходит автоматическое переключение окна конструктора запросов в режим текстового редактора, в котором необходимо набрать текст запроса на языке SQL. Синтаксис запроса на объединение следующий:

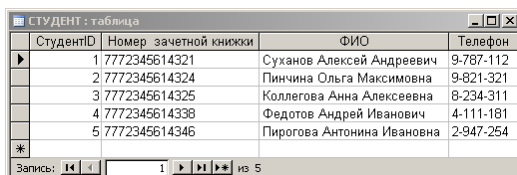
```
SELECT Поле1, Поле2, ...
FROM Таблица1
UNION
SELECT Поле1, Поле2, ...
FROM Таблица2
```

¹ Не следует путать запрос на объединение, которому в языке SQL соответствует команда **UNION** (объединение), с операцией объединения таблиц в запросе, обозначаемой в SQL ключевым словом **JOIN** (соединение).

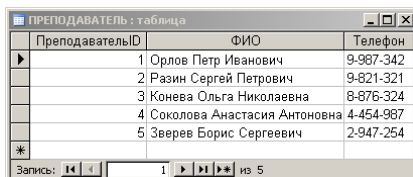
Запрос состоит из трех частей. Первая и третья части образованы парой ключевых слов **SELECT** и **FROM** и представляют собой обыкновенные запросы на выборку. После команды **SELECT** идет перечисление имен столбцов (полей), из которых требуется извлечь данные, после ключевого слова **FROM** указывается имя таблицы, в которой расположены эти столбцы (поля). Между первым и вторым запросом на выборку расположено ключевое слово **UNION**, предполагающее объединить результаты двух запросов в один результирующий набор.

Для успешного выполнения запроса на объединение необходимо, чтобы входящие в него запросы на выборку имели одинаковое количество полей вывода, следующие в одинаковом порядке и имеющие одинаковые или схожие типы данных.

Пример 12. База данных (файл **Пример 12.mdb**) содержит таблицы: **СТУДЕНТ** и **ПРЕПОДАВАТЕЛЬ** (рис. 139).



СтудентID	Номер зачетной книжки	ФИО	Телефон
1	7772345614321	Суханов Алексей Андреевич	9-787-112
2	7772345614324	Пинчина Ольга Максимовна	9-821-321
3	7772345614325	Коллегова Анна Алексеевна	8-234-311
4	7772345614338	Федотов Андрей Иванович	4-111-181
5	7772345614346	Пирогова Антонина Ивановна	2-947-254

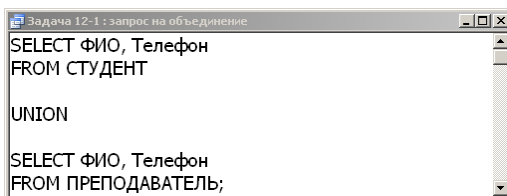


ПреподавательID	ФИО	Телефон
1	Орлов Петр Иванович	9-987-342
2	Разин Сергей Петрович	9-821-321
3	Конева Ольга Николаевна	8-876-324
4	Соколова Анастасия Антоновна	4-454-987
5	Зверев Борис Сергеевич	2-947-254

Рис. 139. Исходные данные (пример 12)

Задача 12.1. Требуется составить список фамилий и телефонов всех преподавателей и студентов.

Решение.



```

SELECT ФИО, Телефон
FROM СТУДЕНТ

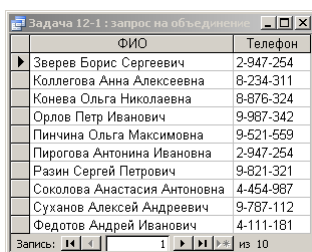
UNION

SELECT ФИО, Телефон
FROM ПРЕПОДАВАТЕЛЬ;

```

Рис. 140. Запрос для решения задачи 12.1

Результат выполнения запроса.



ФИО	Телефон
Зверев Борис Сергеевич	2-947-254
Коллегова Анна Алексеевна	8-234-311
Конева Ольга Николаевна	8-876-324
Орлов Петр Иванович	9-987-342
Пинчина Ольга Максимовна	9-521-559
Пирогова Антонина Ивановна	2-947-254
Разин Сергей Петрович	9-821-321
Соколова Анастасия Антоновна	4-454-987
Суханов Алексей Андреевич	9-787-112
Федотов Андрей Иванович	4-111-181

Записи: 1 из 10

Рис. 141. Результат выполнения запроса (задача 12.1)

Рекомендуемая литература

1. Кузин А.В., Левонисова С.В. Базы данных. М: Academia, 2008.
2. Захарова Л.Официальный учебный курс Microsoft: Microsoft Office Access 2003. М: Эком, 2006.
3. Моисеенко С.И., Соболев Б.В. Разработка приложений в MS Access. М: Вильямс, 2006 г.
4. Гуляев А.К. MS Office XP, 2003. М: Корона Принт, 2006.
5. Моркес Д. Microsoft Access 2003. М: Наука и техника, 2006.
6. Сеннов А. Access 2003. Практическая разработка баз данных. СПб: Питер, 2005.
7. Вейскас Дж. Эффективная работа: Microsoft Office Access 2003. СПб: Питер, 2005.
8. Кошелев В.Е. Access 2003. М: Бином, 2005.

Содержание

Основные сведения о Microsoft Access	3
Построение запросов в Microsoft Access	4
Основные понятия	4
Запрос на выборку	4
Фильтрация данных	5
Мастер построения запросов	7
Конструктор запросов	9
Создание вычисляемых полей	26
Агрегированные запросы	34
Многотабличные запросы	42
Перекрестный запрос.....	52
Запрос на создание таблицы	56
Запрос на добавление данных.....	59
Запрос на удаление данных	62
Запрос на обновление данных	67
Запрос на объединение.....	70
Рекомендуемая литература	73
Содержание	74