

Лабораторная работа №6

ТЕМА: Построение сложных запросов в инструкции SELECT языка SQL. Обработка данных в SQL.

ЦЕЛЬ: Научить создавать подзапросы, вносить изменение в базу данных и извлекать данные из внешних баз данных используя команды языка SQL.

ПЛАН ЗАНЯТИЯ:

- I.** Основные теоретические сведения.
- II.** Создание дополнительных таблиц БД «фирма» и занесение исходных данных.
- III.** Подзапросы в инструкции SELECT.
- IV.** Использование запросов для работы с формами.
- V.** Выбор данных из внешней базы данных.
- VI.** Объединение результатов нескольких запросов.
- VII.** Инструкции языка обработки данных.

I. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Подзапрос — это запрос, который размещается внутри другого запроса, а точнее, является частью предложения **WHERE** (или **HAVING**) основного запроса и заключается в круглые скобки.

Иногда подзапросы называют **вложенными запросами**. Запрос, содержащий подзапрос, называют **сложным запросом**. При выполнении сложного запроса сначала выполняется подзапрос (он для этого и заключается в скобки!), а затем — основной запрос.

Выходные данные подзапроса обычно используются при оценке выражения в предложении **WHERE** (или **HAVING**) основного (по отношению к данному подзапросу) запроса с использованием таких операторов, как **=**, **>**, **<**, **O**, **IN**, **NOT IN**, **AND**, **OR**. (Эти операторы можно использовать и внутри подзапроса.)

Очень важно понимать, какие данные возвращает подзапрос, поскольку от этого зависит синтаксис использования предложения **WHERE**. Необходимо при написании сложных запросов соблюдать следующие правила:

- 1.** Предложение **SELECT** подзапроса обычно содержит только один столбец и, следовательно, может иметь результатом массив однородных (только строковые значения, только целые значения и т.д.) значений или только одно значение. Очень редко встречаются случаи использования нескольких столбцов. Если подзапрос возвращает массив значений, соответствующее этому подзапросу предложение **WHERE** может содержать только многозначные операторы, например, **IN**.
 - 2.** Предложение **ORDER BY** следует использовать только в основном запросе. В подзапросе для упорядочения данных необходимо применять предложение **GROUP BY**.
 - 3.** Недопустимо использование в основном запросе предложения **BETWEEN**.
Простой синтаксис подзапроса для оператора **SELECT** следующий:
-

```

SELECT { * | table.* | [table.]field1 [AS alias1] [, [table.]field2
[AS alias2] [, ...]]}
FROM table1 [table1Alias] [, table2 [table2Alias] [, ...]
WHERE field1 operator
(SELECT { * | table.* | [table.]field1 [AS alias1] [, [table.]field2 [AS alias2] [, ...]]}
FROM table1 [table1Alias] [, table2 [table2Alias] [, ...]
WHERE [criteria])

```

II. СОЗДАНИЕ ДОПОЛНИТЕЛЬНЫХ ТАБЛИЦ БД «ФИРМА» И ЗАНЕСЕНИЕ ИСХОДНЫХ ДАННЫХ.

Для выполнения лабораторной работы необходимо добавить к ранее созданной базе данных таблицы, позволяющие учитывать перемещение товаров между подразделениями. Это можно сделать при помощи таблиц **КонтрАгенты** (информация о подразделениях, в которые и из которых передаются товары), **НаклЗаголовки** (дата и номер накладной, код подразделения, где сформирована накладная, код подразделения, куда «ушел» или откуда «пришел» товар, и тип накладной: приходная, расходная и т.д.), **ТипыНакладных** (коды и наименования типов накладных, например, отпуска, приема товара и т.д.) и **НаклСпецификации** (сведения о товарах, которые связаны с накладной) (рис. 1). Обратите внимание на то, что для таблицы **НаклЗаголовки** значение совокупности полей **Номер**, **Дата** и **КодПодразделения** является уникальным. В таблице же **НаклСпецификации** для каждой такой совокупности будет ровно столько записей, сколько наименований товаров имеется в накладной.

Задание № 1.

В соответствии с приведенной на рис. 1 схемой необходимо изменить файл **фирма.mdb**, то есть добавьте четыре таблицы. Одна из них — **Контр-Агенты** — очень похожа на таблицу **Подразделения**, и ее содержимое можно просто скопировать из этой таблицы (рис. 2), поскольку одни и те же подразделения фирмы могут выступать и в качестве принимающих товары, и в качестве отпускающих их. На самом деле, в данном случае две таблицы для одной и той же информации используются только для простоты. В реальной ситуации для подобных целей таблицы создаются (или копируются) во время выполнения кода.

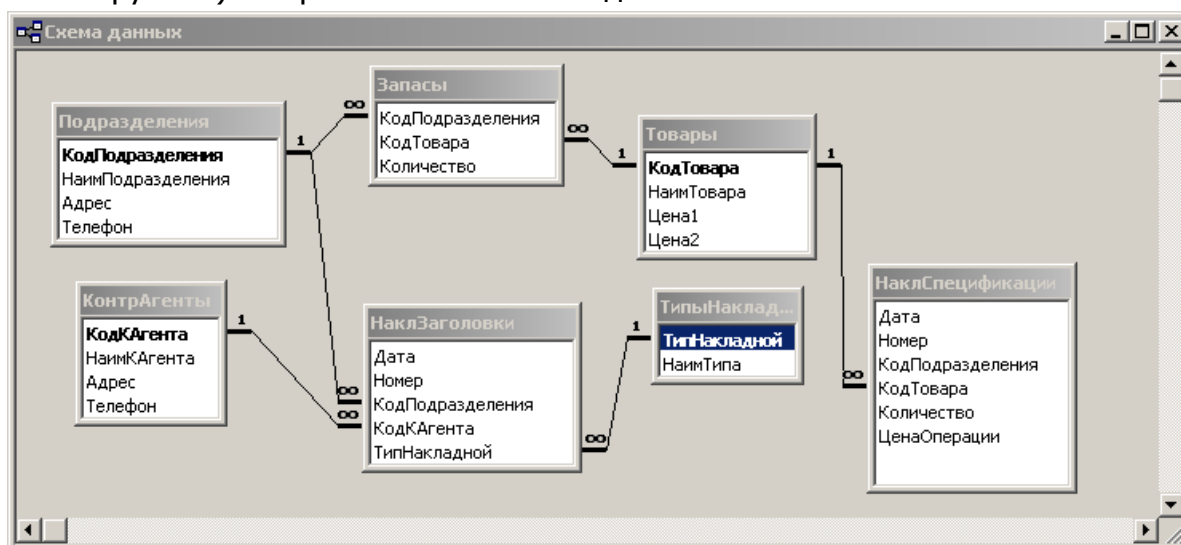


Рис. 1. Дополнительная схема базы «фирма.mdb»

	КодКАгента	НаимКАгента	Адрес	Телефон
+	0429	СКЛАД (Мультимедиа)	ул. П.И. Чайковского, 22	333-33-05
+	1506	СКЛАД (МНЕВНИКИ)	ул. Советская, 77	444-33-45
+	1518	СКЛАД (НЕЙРОН)	пр. Петровича	324-55-45
+	1958	СКЛАД (Брак)	пер. Симоновича	334-33-41
*				

Запись: 1 из 4

Рис. 2. Содержимое таблицы **КонтрАгенты**, скопированное из таблицы **Подразделения**

Названия полей и типы данных дополнительных таблиц приведены в таблице 1.

Таблица 1. Типы полей дополнительных таблиц БД «ФИРМА»

Таблица	Поле	Тип и длина поля
КонтрАгенты	КодКАгента	Текстовое(4)
КонтрАгенты	НаимКАгента	Текстовое(50)
КонтрАгенты	Адрес	Текстовое(60)
КонтрАгенты	Телефон	Текстовое(10)
ТипыНакладных	ТипНакладной	Integer
ТипыНакладных	НаимТипа	Текстовое(20)
НаклЗаголовки	Дата	Date/Time
НаклЗаголовки	Номер	Long Integer
НаклЗаголовки	КодПодразделения	Текстовое(4)
НаклЗаголовки	КодКАгента	Текстовое(4)
НаклЗаголовки	ТипНакладной	Integer
НаклСпецификации	Дата	Date/Time
НаклСпецификации	Номер	Long Integer
НаклСпецификации	КодПодразделения	Текстовое(4)
НаклСпецификации	КодТовара	Текстовое(12)
НаклСпецификации	Количество	Long Integer
НаклСпецификации	ЦенаОперации	Денежный

Заполните таблицы, в которых предполагается хранить информацию о накладных. На рис. 3 приведена заполненная таблица с заголовками накладных. Даты и номера накладных здесь могут быть произвольными, коды полей **КодПодразделения** и **КодКАгента** взяты из таблиц **Подразделения** и **КонтрАгенты**. Совокупность данных первых трех полей должна быть уникальной: одной дате может соответствовать несколько номеров накладных с одним кодом подразделения, но номер накладной не может повторяться для одной и той же даты и кода подразделения.

В качестве примера первую строку таблицы можно интерпретировать так: **11 января 2003 года** на складе с кодом **"0429"** была сформирована накладная с номером **1**, кодом типа **5** (расход на склад) и контрагентом с кодом **"1958"**.

Дата	Номер	КодПодраздел	КодКАгента	ТипНакладной
11.01.2002	5	1506	1518	2
11.02.2002	1	0429	1958	2
11.01.2002	1	0429	1958	5
11.02.2002	2	0429	1506	5
11.02.2002	3	0429	1506	5
11.02.2002	4	0429	1958	5
11.03.2002	1	0429	1518	5
11.03.2002	2	0429	1958	2
11.03.2002	3	0429	1518	5
11.03.2002	4	0429	1958	5

Запись: 11 из 11

Рис. 3. Заполненная таблица с заголовками накладных

Таблицу с типами накладных заполнить очень просто (рис. 4), хотя, на самом деле, типов накладных может быть гораздо больше.

ТипНакладной	НаимТипа
2	Приход со склада
4	Приход из опт. маг.
5	Расход на склад

Запись: 1 из 3

Рис. 4. Заполненная таблица с типами накладных

Дата	Номер	КодПодраздел	КодТовара	Количество	ЦенаОперации
11.01.2002	1	0429	010415090937	100	120,00р.
11.01.2002	1	0429	010414114438	50	135,00р.
11.01.2002	5	1506	028019100222	546	83,00р.
11.02.2002	1	0429	006019165803	100	42,00р.
11.02.2002	1	0429	028019100222	140	88,00р.
11.02.2002	2	0429	186100175232	100	750,00р.
11.02.2002	2	0429	174500175521	200	750,00р.
11.02.2002	2	0429	028019100222	60	89,00р.
11.02.2002	2	0429	010415090732	12	109,00р.
11.02.2002	3	0429	174500175359	200	750,00р.
11.02.2002	3	0429	028019100222	545	85,00р.
11.02.2002	4	0429	174500175521	222	750,00р.
11.02.2002	4	0429	028019100222	334	88,00р.
11.02.2002	4	0429	186100175232	767	745,00р.
11.03.2002	1	0429	028019100222	232	88,00р.
11.03.2002	1	0429	010415090732	222	110,00р.
11.03.2002	2	0429	010415090732	111	111,00р.
11.03.2002	2	0429	174500175359	234	750,00р.
11.03.2002	3	0429	174500175359	432	750,00р.
11.03.2002	4	0429	174500175521	67	750,00р.
11.03.2002	5	0429	186100175232	231	750,00р.

Запись: 18 из 21

Рис. 5. Заполненная таблица со спецификациями накладных

На рис. 5 приведена таблица **НаклСпецификации**. Надо помнить, что эта таблица содержит коды, количества и цены (или производные от цен) товаров, которые перемещаются в соответствии с накладными, заголовки для которых уже запол-

нены. Это означает, что первые три поля заполняются информацией, которая берется из первых трех полей таблицы **НаклЗаголовки** (рис. 3), а в остальные помещаем любой код товара из таблицы **Товары**, произвольное количество и цену, примерно равную любой цене из той же таблицы. Так, например, на рис. 5 первые две записи означают, что спецификация накладной, заголовок которой хранится в первой записи таблицы **НаклЗаголовки**, содержит сведения о двух товарах: с кодом "010415090937" — (VCD) Шоссе в никуда — в количестве 100 шт. и ценой 120 р.; с кодом "010414114438" — (VCD) Патриот — в количестве 50 шт. и ценой 135 р.

III. ПОДЗАПРОСЫ В ИНСТРУКЦИИ SELECT

Рассмотрим использование подзапроса, если он имеет результатом единственное значение. Например, пусть нам необходимо получить список товаров склада, на котором их больше всего. Для начала нам нужно определить этот склад. Создадим запрос, который возвратит код склада с наибольшим количеством товаров:

```
SELECT TOP 1 КодСклада
FROM Инвентарная_ведомость
GROUP BY КодСклада
ORDER BY SUM(Количество) DESC
```

Предложение **TOP 1** указывает на то, что в результирующий набор попадает только одна запись, а поскольку набор сортируется (по убыванию) по суммарным количествам товаров, то в результате мы получаем код склада с наибольшим количеством товаров. Остается использовать ранее рассмотренный запрос, который возвращает список товаров для указанного кода склада, но если ранее мы указывали этот код, то теперь определяем его в подзапросе (результат — на рис. 6):

```
SELECT a.НаимТовара AS [Наименование товара], b.Количество AS Количество
FROM Товары AS a, Запасы AS b
WHERE a.КодТовара=b.КодТовара AND b.КодПодразделения =
(SELECT TOP 1 КодПодразделения
FROM Запасы
GROUP BY КодПодразделения
ORDER BY SUM(Количество) DESC)
```

Наименование товара	Количество
(VCD) Патриот	3300
(VCD) Доберман	4200
(3DC) Street Fighter 3 Wimpact	2200
(3DC) Street Fighter 3 Wimpact	2300
(3DC) Super Runabout	1300

Рис. 6. Результат выполнения сложного запроса

Целью следующего запроса является получение ведомости товаров склада, где имеется больше всего наименований (кодов) товаров (результат — на рис. 7).

```
SELECT a.НаимТовара AS [Наименование товара], b.Количество AS Количество
FROM Товары AS a, Запасы AS b
WHERE a.КодТовара=b.КодТовара AND b.КодПодразделения =
(SELECT TOP 1 КодПодразделения
```

```
FROM Запасы
GROUP BY КодПодразделения ORDER BY COUNT(КодТовара) DESC)
```

Здесь в подзапросе вместо функции **SUM** используется **COUNT** — счетчик количества записей.

Наименование товара	Количество
(S) Road Rash II	200
(S) Tazmania	500
(VCD) Патриот	400
(VCD) Доберман	800
(S) Cool Spot	600
(3DC) Street Fighter 3 Wimpact	600

Рис. 7. Результат выполнения сложного запроса

Задание № 2.

Выполните рассмотренные запросы и сохраните результат под именами, указанными на рисунках 6 и 7.

IV. ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ ДЛЯ РАБОТЫ С ФОРМАМИ

Чтобы понять, какие запросы можно построить с использованием дополненной базы данных, спрогнозируем, какая информация может потребоваться менеджеру для оценки движения товаров на фирме. Например, для проверки некоторой документации, необходимо просто просмотреть накладные какого-либо подразделения. Понятно, что менеджер не будет просматривать накладные в том же режиме, в котором эти таблицы создавались. Менеджеру мы предоставим некоторую форму, созданную в VB.

Если бы нам нужно было создать форму для просмотра накладных, то она, вероятно, была бы похожа на ту, которая показана на рис. 8. Изображенная на этом рисунке форма получена в редакторе VB только с одной целью — понять, какие SQL-запросы были бы необходимы для заполнения такого диалогового окна. Вопросам реализации такой и других форм посвящены следующие лабораторные работы.

Рис. 8. Диалоговое окно для просмотра накладных

Во-первых, для заполнения окна комбинированного списка в секции **Подразделение** необходим запрос, в результате которого получался бы список подразделений, накладные которых имеются в таблице **НаклЗаголовки**. Во-вторых, следует заполнить окно комбинированного списка в секции **Дата** датами (уникальными) из таблицы **НаклЗаголовки**¹.

Далее, после выбора пользователем данных в секциях **Подразделение** и **Дата** в окно комбинированного списка **Накладная** следует поместить список накладных (с указанием номера накладной, ее типа и наименованием контрагента). Для этого необходимо построить запрос с целью получения таблицы номеров, типов, контрагентов накладных для определенного подразделения и указанной даты.

После заполнения данными списка **Накладная** и пользовательского выбора в этом окне останется только для известного подразделения, даты и номера накладной сформировать запрос по таблице **НаклСпецификации**, чтобы получить данные о конкретной накладной.

Формирование списка подразделений

Итак, рассмотрим первый запрос — формирование списка подразделений, накладные которых имеются в таблице **НаклЗаголовки**. Целесообразно получить таблицу наименований и кодов подразделений. Наименованиями следует заполнить окно комбинированного списка для отображения в диалоговом окне. Коды необходимо записать в список, не отображаемый в окне, но синхронизированный с видимым комбинированным списком. После выбора пользователем необходимого подразделения из окна комбинированного списка можно узнать код этого подразделения из не отображаемого списка.

И так, запрос, о котором идет речь, должен быть следующим:

```
SELECT DISTINCT b.КодПодразделения, a.НаимПодразделения
FROM Подразделения a, НаклЗаголовки b
WHERE a.КодПодразделения =b.КодПодразделения
```

Если выполнить этот запрос для данных, отображенных на рис. 3 и 5 (и таблицы **Склады**), то получится набор, приведенный на рис. 9.

КодПодразделения	НаимПодразделения
0429	СКЛАД (Мультимедиа)
1506	СКЛАД (МНЕВНИКИ)

Рис. 9. Склады, накладные которых имеются в таблице **НаклЗаголовки**

Таким образом, если пользователь выберет необходимый склад, то мы (или программа, поддерживающая диалоговое окно) будем знать код этого склада.

Задание № 3.

Сформируйте запрос, результат которого изображен на рис. 9 и сохраните под тем же названием.

¹ Можно было бы, конечно, просто предоставить менеджеру окно для ввода дат, но тогда при отсутствии накладных за указанную дату менеджер получал бы сообщение о том, что накладные отсутствуют, а это не похоже на дружелюбный интерфейс.

Формирования списка дат

Считая известным код склада, создадим очень простой запрос для формирования списка дат, например, для склада с кодом "0429".

```
SELECT DISTINCT Дата
FROM НаклЗаголовки
WHERE КодПодразделения="0429"
```

Результатом этого запроса может быть таблица, приведенная на рис. 10

Дата
11.01.2002
11.02.2002
11.03.2002

Рис. 10. Даты накладных (в таблице **НаклЗаголовки**) для склада с кодом «0429»

Задание № 4.

Создайте запрос для формирования списка дат, как показано на рис. 10.

Формирование списка накладных

После выбора пользователем склада и даты необходимо сформировать запрос для списка **Накладная** с целью получения таблицы номеров, типов, контрагентов накладных для определенного подразделения и указанной даты (например, **11/02/2002**). Правильным будет упорядочить результат запроса по номерам накладных (результатом этого запроса может быть таблица, приведенная на рис. 11):

```
SELECT a.Номер, b.НаимКАгента, a.ТипНакладной
FROM НаклЗаголовки AS a, КонтрАгенты AS b
WHERE a.КодКАгента = b.КодКАгента AND a.КодПодразделения="0429"
AND a.Дата = CDate("11/02/2002") ORDER BY a.Номер
```

Номер	НаимКАгента	ТипНакладной
1	СКЛАД (Брак)	2
2	СКЛАД (МНЕВНИКИ)	5
3	СКЛАД (МНЕВНИКИ)	5
4	СКЛАД (Брак)	5

Рис. 11. Такая таблица может послужить источником данных для списка **Накладная**

Если менеджер имеет хорошую память для того, чтобы помнить все (около десяти) коды типов накладных, то можно не дополнять этот запрос наименованиями типов накладных. Хотя, с другой стороны, зря тогда вообще создавалась таблица **ТипыНакладных**. В следующем запросе раскрываются наименования типов накладных (результат на рис. 12):


```

SELECT a.Номер, b.НаимКАгента, c.НаимТипа
FROM НаклЗаголовки AS a, КонтрАгенты AS b, ТипыНакладных AS c
WHERE a.КодКАгента = b.КодКАгента AND a.КодПодразделения = "0429"
      AND a.Дата = CDate("11/02/2002") AND a.ТипНакладной = c.ТипНакладной
ORDER BY a.Номер

```

Номер	НаимКАгента	НаимТипа
1	СКЛАД (Брак)	Приход со склада
2	СКЛАД (МНЕВНИКИ)	Расход на склад
3	СКЛАД (МНЕВНИКИ)	Расход на склад
4	СКЛАД (Брак)	Расход на склад

Рис. 12. Такая таблица может послужить источником данных для списка **Накладная**

Задание № 5.

Сформируйте описанные выше запросы и сохраните как «Запрос 23» и Запрос 24».

Формирование спецификации выбранной накладной

Осталось сформировать запрос для заполнения таблицы со спецификацией выбранной накладной, например, с номером 2 (результатом запроса может быть таблица, приведенная на рис. 13):

```

SELECT b.НаимТовара, a.Количество, a.ЦенаОперации
FROM НаклСпецификации a, Товары b
WHERE a.КодТовара=b.КодТовара AND a.КодПодразделения="0429"
      AND a.Дата=CDate("11/02/2002") AND Номер =2
ORDER BY b.НаимТовара

```

НаимТовара	Количество	ЦенаОперации
(ЗДС) Star Gladiator 2	200	750,00р.
(ЗДС) Super Runabout	100	750,00р.
(VCD) Голая мишень	60	89,00р.
(VCD) Пастырь	12	109,00р.

Рис. 13. Таблица со спецификацией накладной

Используя таблицы **НаклЗаголовки** и **НаклСпецификации**, можно получать не только накладные. Можно также создавать различные отчеты о поступлениях товаров на склады или расходов со складов. Из рис. 12 следует, что **1 февраля 2002 года** со склада с кодом **"0429"** были отправлены товары на два склада с созданием трех накладных. Следующий запрос суммирует эти накладные в один отчет, в котором содержится информация о всех товарах, отправленных с этого склада указанного числа (результат запроса имеет вид, представленный на рис. 14).

Наименование	Количество	Сумма
(3DC) Star Gladiator 2	422	316 500,00р
(3DC) Street Fighter 3 Wir	200	150 000,00р
(3DC) Super Runabout	867	646 415,00р
(VCD) Голая мишень	939	81 057,00р
(VCD) Пастырь	12	1 308,00р

Рис. 14. Отчет о расходе товаров со склада 11 февраля 2002 года

```
SELECT b.НаимТовара AS Наименование, SUM(a.Количество) AS Количество,
      Format(SUM(a.Количество*a.ЦенаОперации), "### ##0.00р") AS Сумма
FROM НаклСпецификации AS a, Товары AS b
WHERE a.КодТовара=b.КодТовара AND a.КодПодразделения ="0429"
      AND STR(a.Дата) & STR(a.Номер)
IN
  (SELECT STR(Дата) & STR(Номер)
   FROM НаклЗаголовки
   WHERE Дата = CDate ("11/02/2002") AND ТипНакладной=5)
GROUP BY b.НаимТовара;
```

Здесь для определения используемых записей таблицы **НаклСпецификации** применяется подзапрос, формирующий список строк вида **STR(Дата) & STR(Номер)** при условии, что дата равна выражению **CDate("11/02/2002")**, а тип накладной — **5**:

```
(SELECT STR(Дата) & STR(Номер)
 FROM НаклЗаголовки
 WHERE Дата = CDate("11/02/2002" ) AND ТипНакладной=5)
```

Задание № 6.

Выполните и сохраните описанные в разделе «Формирование спецификации выбранной накладной» запросы

V. ВЫБОР ДАННЫХ ИЗ ВНЕШНЕЙ БАЗЫ ДАННЫХ

Иногда бывает необходимо выполнять запросы к базам данных, не открывая их. Для этого в инструкции **SELECT** в предложении **FROM** после имен запрашиваемых полей можно указать полное имя файла с базой данных. Для тестирования запросов к внешней базе данных следует создать пустую (без таблиц) базу данных и написать в этой базе, например, следующий запрос (результат полностью совпадает с таблицей, представленной на рис. 15):

```
SELECT НаимТовара AS [Наименование товара], Цена1 AS [Цена закупочная]
FROM Товары IN 'c:\temp\фирма.mdb'
WHERE НаимТовара LIKE "*мишень*"
```

Наименование товара	Цена закупочная
(VCD) Голая мишень	90,00р.
*	0,00р.

Рис. 15. Выбор данных из внешней базы

Не следует забывать указывать путь к внешней базе данных в подзапросе. Следующий запрос дает в результате ту же таблицу, что представлена на рис. 14.

```
SELECT b.НаимТовара AS Наименование, SUM(a.Количество) AS Количество,  
       Format(SUM(a.Количество*a.ЦенаОперации), "### ##0.00p") AS Сумма  
FROM НаклСпецификации AS a, Товары AS b IN 'c:\temp\фирма.mdb'  
WHERE a.КодТовара=b.КодТовара AND a.КодПодразделения ="0429"  
      AND STR(a.Дата) & STR(a.Номер)  
IN  
  (SELECT STR(Дата) & STR(Номер)  
   FROM НаклЗаголовки IN 'c:\temp\фирма.mdb'  
   WHERE Дата = CDate ("11/02/2002") AND ТипНакладной=5)  
GROUP BY b.НаимТовара;
```

В сложном запросе для каждого предложения **FROM** (т.е. для каждого **SELECT**-запроса) может быть указана своя внешняя база данных. Например:

```
SELECT b.НаимТовара AS Наименование, SUM(a.Количество) AS Количество,  
       Format(SUM(a.Количество*a.ЦенаОперации), "### ##0.00p") AS Сумма  
FROM НаклСпецификации AS a, Товары AS b IN 'c:\temp\фирма.mdb'  
WHERE a.КодТовара=b.КодТовара AND a.КодПодразделения ="0429"  
      AND STR(a.Дата) & STR(a.Номер)  
IN  
  (SELECT STR(Дата) & STR(Номер)  
   FROM НаклЗаголовки IN 'c:\temp\фирма2.mdb'  
   WHERE Дата = CDate ("11/02/2002") AND ТипНакладной=5)  
GROUP BY b.НаимТовара
```

Перед выполнением этого запроса таблица **НаклЗаголовки** должна быть перенесена в базу данных файла **фирма2.mdb**.

Если в одном из **SELECT**-запросов используется текущая база, то ее указывать, как обычно, не нужно. Например:

```
SELECT b.НаимТовара AS Наименование, SUM(a.Количество) AS Количество,  
       Format(SUM(a.Количество*a.ЦенаОперации), "### ##0.00p") AS Сумма  
FROM НаклСпецификации AS a, Товары AS b IN 'c:\temp\фирма.mdb'  
WHERE a.КодТовара=b.КодТовара AND a.КодПодразделения ="0429"  
      AND STR(a.Дата) & STR(a.Номер) IN  
(SELECT STR(Дата) & STR(Номер)  
 FROM НаклЗаголовки  
 WHERE Дата = CDate ( "11/02/2002") AND ТипНакладной=5)  
GROUP BY b.НаимТовара;
```

Перед выполнением этого запроса таблица **НаклЗаголовки** должна быть перенесена в базу данных, из которой выполняется запрос.

Задание № 7.

Создайте и сохраните запросы, иллюстрирующие выбор данных из внешней базы данных

VI. ОБЪЕДИНЕНИЕ РЕЗУЛЬТАТОВ НЕСКОЛЬКИХ ЗАПРОСОВ

Иногда бывает необходимо объединять результаты выполнения двух и более **SELECT**-инструкций. Эту операцию можно выполнить при помощи предложения

UNION. Поскольку результатом всего запроса является один набор, каждый из запросов, объединяемых предложением **UNION**, должен содержать одинаковое количество полей с одинаковыми типами данных — практически, объединяются тождественные наборы. Можно написать сколь угодно сложную **SELECT**-инструкцию, добавить к ней слово **UNION**, за которым будет другая **SELECT**-инструкция и т.д., но все **SELECT**-инструкции должны возвращать одинаковые наборы.

Для демонстрации работы с объединением, представим себе, что по не обсуждаемым здесь причинам мы имеем более одного mdb-файла со структурой рассматриваемого до сих пор файла **фирма.mdb**. Таблицы **Товары** файлов, например, **фирма1.mdb** и **фирма2.mdb** представлены на рис. 16-17 и, в общем-то, могут содержать одинаковые записи.

КодТовара	НаимТовара	Цена1	Цена2
010414114438	(VCD) Патриот	138,00р.	161,00р.
010415090732	(VCD) Пастырь	111,00р.	130,00р.
010415090937	(VCD) Шоссе в никуда	120,00р.	140,00р.
010428144348	(VCD) Доберман	135,00р.	158,00р.
028019095823	(VCD) Diber Marouani	90,00р.	105,00р.
028019100018	(VCD) Wham! The best of	105,00р.	123,00р.
028019100222	(VCD) Голая мишень	90,00р.	105,00р.
036019164741	(S) Cool Spot	42,00р.	49,00р.
174500175359	(3DC) Street Fighter 3 W'ir	750,00р.	875,00р.
174500175521	(3DC) Star Gladiator 2	750,00р.	875,00р.
186100175232	(3DC) Super Runabout	750,00р.	875,00р.
*		0,00р.	0,00р.

Рис. 16. Таблица Товары файла **фирма1.mdb**

КодТовара	НаимТовара	Цена1	Цена2
006019165803	(S) Road Rash II	42,00р.	49,00р.
006019170049	(S) Tazmania	30,00р.	35,00р.
010414114438	(VCD) Патриот	138,00р.	161,00р.
010415090732	(VCD) Пастырь	111,00р.	130,00р.
010415090937	(VCD) Шоссе в никуда	120,00р.	140,00р.
010428144348	(VCD) Доберман	135,00р.	158,00р.
028019095823	(VCD) Diber Marouani	90,00р.	105,00р.
028019100018	(VCD) Wham! The best of	105,00р.	123,00р.
028019100222	(VCD) Голая мишень	90,00р.	105,00р.
174500175521	(3DC) Star Gladiator 2	750,00р.	875,00р.

Рис. 17. Таблица Товары файла **фирма2.mdb**

Для получения общей ведомости товаров можно использовать следующую инструкцию:

```
SELECT НаимТовара AS Наименование,
       Format(Цена1, "### ##0.00р")
AS [Цена оптовая]
FROM Товары IN 'c:\temp\фирма1.mdb'
```

UNION

```
SELECT НаимТовара AS Наименование,  
       Format(Цена1, "### ##0.00p")  
AS [Цена оптовая]  
FROM Товары IN 'c:\temp\фирма2.mdb'
```

Как видно из результата запроса (рис. 18), в полученном наборе нет повторяющихся записей. Это — свойство объединения: записи результирующих наборов не повторяются. Если же вы измените в одной из таблиц для повторяющегося товара значение в поле **Цена1** (**Цена2** не входит в запрос), то в результирующем наборе это наименование появится дважды, но с разными ценами (рис. 19).

Наименование	Цена оптовая
(3DC) Star Gladiator 2	750,00p
(3DC) Street Fighter 3 Wimpact	750,00p
(3DC) Super Runabout	750,00p
(S) Cool Spot	42,00p
(S) Road Rash II	42,00p
(S) Tazmania	30,00p
(VCD) Diber Marouani	90,00p
(VCD) Wham! The best of	105,00p
(VCD) Голая мишень	90,00p
(VCD) Доберман	135,00p
(VCD) Пастырь	111,00p
(VCD) Патриот	138,00p
(VCD) Шоссе в никуда	120,00p

Рис. 18. Результирующий запрос-объединение

Наименование	Цена оптовая
(3DC) Star Gladiator 2	750,00p
(3DC) Street Fighter 3 Wimpact	750,00p
(3DC) Super Runabout	750,00p
(S) Cool Spot	42,00p
(S) Road Rash II	42,00p
(S) Tazmania	30,00p
(VCD) Diber Marouani	90,00p
(VCD) Wham! The best of	105,00p
(VCD) Голая мишень	90,00p
(VCD) Доберман	135,00p
(VCD) Пастырь	111,00p
(VCD) Патриот	138,00p
(VCD) Шоссе в никуда	120,00p
(VCD) Шоссе в никуда	135,00p

Рис. 19. Результирующий запрос-объединение

Обратите внимание на то, каким значком (см. рис. 20) помечен запрос с использованием **UNION**. В подрисовочных надписях к рис. 18 и 19 подобные запросы называются «запросами-объединениями».

При помощи предложения **UNION ALL** можно получить в одном наборе все записи первой и второй таблицы, даже если в таблицах имеются одинаковые записи (одинаковые значения во всех полях).

В запросе, использующем **UNION**, можно упорядочивать результирующие записи предложением **ORDER BY**. При этом допустимо только одно предложение **ORDER BY**, которое следует располагать после всех предложений **SELECT** и которое действует для обоих запросов. Например:

```
SELECT НаимТовара AS Наименование, Цена1 AS [Цена оптовая]
FROM Товары IN 'c:\temp\фирма1.mdb'
UNION
SELECT НаимТовара AS Наименование, Цена1 AS [Цена оптовая]
FROM Товары IN 'c:\temp\фирма2.mdb'
ORDER BY 2
```

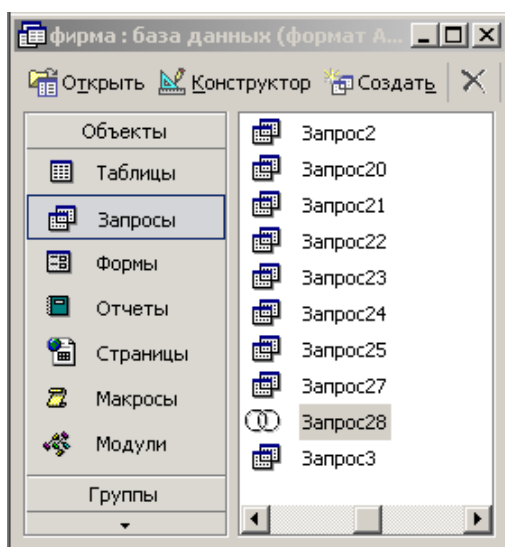


Рис. 20. Для значка запроса-объединения используется соответствующий рисунок

Результат выполнения этого запроса мог бы выглядеть так, как представлено на рис. 21.

Запрос29 : запрос на объединение	
Наименование	Цена оптовая
(S) Tazmania	30,00р.
(S) Cool Spot	42,00р.
(S) Road Rash II	42,00р.
(VCD) Diber Marouani	90,00р.
(VCD) Голая мишень	90,00р.
(VCD) Wham! The best of	105,00р.
(VCD) Пастырь	111,00р.
(VCD) Шоссе в никуда	120,00р.
(VCD) Доберман	135,00р.
(VCD) Шоссе в никуда	135,00р.
(VCD) Патриот	138,00р.
(3DC) Star Gladiator 2	750,00р.
(3DC) Street Fighter 3 Wimpact	750,00р.
(3DC) Super Runabout	750,00р.

Рис. 21. Использование **ORDER BY** в запросе-объединении

Задание № 8.

Создайте БД **фирма1.mdb** и **фирма2.mdb**, скопируйте из БД **фирма.mdb** таблицу **Товары**, внеся в них изменения как показано на рисунках 16 и 17. В БД **фирма.mdb** выполните запросы с использованием предложения **UNION**.

VII. ИНСТРУКЦИИ ЯЗЫКА ОБРАБОТКИ ДАННЫХ

Рассмотрим довольно простые инструкции языка обработки данных (DML), которые используются для добавления, корректировки и удаления строк в таблицах и включают такие инструкции, как:

- **INSERT** (вставить данные в таблицу);
- **UPDATE** (обновить данные);
- **DELETE** (удалить данные).

Эти инструкции мы также продемонстрируем на примерах работы с таблицами базы данных **фирма.mdb** с использованием того же механизма, который применялся для **SELECT**-запросов. Но вы, конечно, никогда не будете использовать этот механизм для реализации приложений. Он хорош только для изучения **SQL**-инструкций. Что же касается **SELECT**-запросов, которые были рассмотрены ранее, то многие из них (а именно те, которые не используют в логических выражениях предложения **WHERE** литеральных значений) могут без изменения применяться в приложениях на языках высокого уровня.

Инструкция INSERT INTO

Инструкция **INSERT INTO** позволяет добавлять записи в таблицу и имеет следующий синтаксис:

для добавления одной записи

```
INSERT INTO targetTable [(field1 [, field2 [...]])]  
[IN externaldatabase ]  
VALUES (value1 [, value2 (, ... )])
```

для добавления нескольких записей

```
INSERT INTO targetTable [(field1 [, field2 [, ... ]])]  
[IN externaldatabase ]  
SELECT [sourceTable. ] field1 [,field2] [, ... ] FROM expression
```

Здесь **targetTable** — имя таблицы или запроса, куда добавляются данные; после **targetTable** могут следовать **field1**, **field2**, ... — имена полей. **sourceTable** — имя таблицы (псевдонима) с исходными данными; после **sourceTable** могут следовать имена полей (**field1**, **field2**, ...). Если вставка выполняется в базу данных, которая не является текущей, следует использовать предложение **IN externaldatabase**, где **externaldatabase** — полное имя внешней базы данных. Предложение **VALUES (value1 [, value2 [, ...]])** определяет значения указанных полей добавляемой записи.

Для добавления одной записи к таблице можно использовать следующий запрос (запрос на добавление):

```
INSERT INTO Товары ( КодТовара, НаимТовара, Цена1, Цена2 )  
VALUES ('000000000002', 'Косолапый мишка', 12, 15)
```

Для добавления записи к таблице неоткрытой базы данных можно использовать следующий запрос:


```
INSERT INTO Товары ( КодТовара, НаимТовара, Цена1, Цена2 ) IN "C:\temp\DB4.MDB"  
VALUES ('000000000003', 'Плюшевый заяц', 12, 15)
```

Поскольку инструкция `INSERT INTO` позволяет вставлять данные, полученные в результате запроса, то нетрудно догадаться, что с ее помощью можно копировать данные из одной (или нескольких) таблицы в другую. Например, следующая инструкция просто копирует данные из таблицы **Товары** базы **фирма2.mdb** в одноименную таблицу базы **фирма3.mdb**:

```
INSERT INTO Товары IN 'c:\temp\фирма3.mdb'  
SELECT* FROM Товары IN 'c:\temp\фирма2.mdb'
```

Понятно, что можно копировать не все, а только определенные записи из нескольких таблиц.

Задание № 9.

Выполните описанные выше примеры запросов, при необходимости создав новую БД (DB4.MDB).

Инструкция UPDATE

Инструкция **UPDATE**, позволяет обновлять записи в таблице, не создает возвращаемый набор и имеет следующий синтаксис:

```
UPDATE table SET newvalue WHERE criteria
```

Здесь **table** — имя таблицы, данные которой следует обновить; **newvalue** — выражение, определяющее значения, которые вставляются в каждое поле; **criteria** — выражение, которое определяет обновляемые записи.

Например, следующая инструкция удваивает значение поля **Цена1** таблицы **Товары** базы данных **C:\temp\фирма3.MDB** для записей, у которых значение поля **Цена1** меньше **100**:

```
UPDATE Товары IN 'c:\temp\фирма3.MDB' SET Цена1=Цена1*2.0 WHERE Цена1 < 100
```

Если необходимо изменить содержимое нескольких полей, то можно использовать, например, такую инструкцию:

```
UPDATE Товары IN 'c:\temp\фирма3.MDB' SET Цена1=Цена1*2.0, Цена2=Цена2*2.3  
WHERE Цена1 < 100
```

Задание № 10.

Создайте и сохраните запросы, использующие инструкцию **UPDATE**

Инструкция DELETE

Инструкция **DELETE** удаляет записи из таблицы и имеет синтаксис:

```
DELETE FROM table WHERE criteria
```

Здесь **table** — имя таблицы, данные из которой следует удалить; **criteria** — выражение, которое определяет удаляемые записи.

Например, следующая инструкция удаляет из таблицы **Товары** базы данных **C:\temp\фирма3.MDB** записи, у которых значение поля **Цена1** больше **100**:

```
DELETE FROM Товары IN 'c:\temp\фирма3.MDB' WHERE Цена1 > 100
```

Задание № 11.

Создайте запрос, который удалит из таблицы **Товары** базы данных **фирма2.MDB** записи, у которых значение поля **Цена2** меньше **500**.

VIII. КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Дайте определение подзапроса.
2. Какой запрос называется сложным?
3. Что позволяет осуществить в запросе предложение **TOP 1**?
4. Как построить запрос — формирование списка?
5. Как осуществить запросы к базам данных, не открывая их?
6. Что собой представляет запрос-объединение, приведите примеры синтаксиса такого рода запросов?
7. Какие инструкции языка обработки данных вы знаете, расскажите о их назначении?
8. Как в запросе, использующем **UNION**, можно упорядочивать результирующие записи?
9. Как работает инструкция **INSERT INTO**, ее синтаксис?
10. Назначение инструкции **UPDATE**, ее синтаксис?
11. С помощью, какой команды можно удалить записи из таблицы?