

# Лекция №8

- 1. Запросы – действия или запросы вносящие изменения**
- 2. Построение *SQL* - запросов**

# **Запросы- действия или запросы, вносящие изменения**

***Запрос, вносящий изменения,  
можно рассматривать как  
обычный запрос на выборку,  
который должен ещё  
выполнять некоторые  
операции над указанной  
группой записей в  
результатирующей таблице.***

Для преобразования **запроса-выборки в запрос действие** , необходимо выбрать соответствующий тип запроса из меню.

В окне базы данных запросы-действия имеют каждый свою пиктограмму.

Т.к. действия запросов, вносящих изменения, **необратимы**, то следует придерживаться следующей последовательности действий:

1. создать запрос выборку
2. *перейти в режим просмотра результирующей таблицы и просмотреть записи, выбранные в запросе*

3. *преобразовать запрос выборку в запрос действие*
4. выполнить запрос: кнопка !
5. проверить внесенные изменения в режиме просмотра результирующей таблицы.

**ЗАМЕЧАНИЕ:** нужно создавать резервные копии обрабатываемых таблиц!!!

# Создание запроса для изменения (обновления) значений

**Пример 11:** Предположим, мы хотим сделать 15% скидку по **конкретной услуге**, заказанной после определенной даты.

Для создания такого запроса, следует пройти 2 этапа:

- ❑ создать запрос на выборку и просмотреть данные, подлежащие изменению;
- ❑ преобразовать запрос-выборку в запрос-обновление и выполнить.

В нашем случае : выберем услуги с кодом 3,  
заказанные после 1 января 2010 года

Бланк запроса QBE будет выглядеть следующим  
образом:

Поле:	Код услуги	Дата исполнения заказа	Скидка
Имя таблицы:	<b>Услуги</b>	<b>Заказы</b>	<b>Заказы</b>
Условие отбора:	3	>#01.01.10#	
Вывод на экран:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Преобразуем запрос на выборку в запрос на обновление:

- выберем команду **Запрос → обновление**, в запрос будет добавлена строка **Обновление**.
- в строке **Обновление** в поле **Скидка** введем 0,15.
- на панели инструментов щелкнуть кнопку **!**
- для завершения выполнения запроса нужно ответить **Да**.

# Запрос на создание таблицы

**Пример 15:** Необходимо составить таблицу всех клиентов, сделавших заказ за последние шесть месяцев

Поле:	Наименование клиента	Дата принятия заказа	Наименование услуги
Имя таблицы:	Клиенты	Заказы	Услуги
Условие отбора:		>#5.06.2012#	
Вывод на экран:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- ✓ Построим запрос-выборку;
- ✓ Выбрать в меню **Запрос → создание таблицы**;
- ✓ В поле **Имя таблицы** – новое имя таблицы
- ✓ Перейти в режим просмотра результирующей таблицы, а затем в конструктор.
- ✓ Вывод дат лучше отключить.
- ✓ Кнопка **!** → Да.

# Создание запроса для удаления записей

Это наиболее опасный запрос!!!

Удаляющий запрос может удалять записи из нескольких таблиц одновременно.

Для этого нужно чтобы:

- было определено отношение между таблицами;
- включена опция **Обеспечение целостности данных**;
- включена опция **Каскадное удаление связанных записей**.

## Пример 15: Удалить заказы, обработанные в прошлом году.

1. *Необходимо построить запрос выборку;*
2. *Запрос/удаление;*

- В таблице Заказы выбрать поле Дата принятия заказа;
- Для этого поля задайте критерий **>=#1/01/11# and <#1.01.12#;**
- Нужно проверить, что результирующая таблица содержит записи, относящиеся только к 2008 году;
- Режим конструктора;
- Кнопка **!**;
- Кнопка ДА. Записи будут удалены из таблиц.

# Построение SQL - запросов

SQL – выражения вводятся в режиме SQL, для этого нужно выбрать Запрос/Запрос SQL (кнопки нет).

С помощью SQL запросов можно выполнить следующие действия:

- **Объединяющий запрос** комбинирует общие поля из одной или нескольких таблиц.
- **Отправляющий запрос** отправляет SQL-инструкции на SQL-сервер в формате базы данных MsA.
- **Запрос, воздействующий на данные** используются реже всего. Этот запрос создает или изменяет объекты непосредственно в базе данных MsA.

## Просмотр SQL-выражений в запросах

**SQL** – это стандартизированный язык, поддерживаемый многими реляционными базами данных. Этот язык весьма прост в изучении и использовании. Для просмотра **SQL**-выражений выберите Вид/Режим **SQL**.

**SQL**-выражения можно использовать в выражениях, макросах, формах и отчетах.

Один из способов научиться строить **SQL**-выражения – это строить запросы, а затем изучать соответствующие им запросы **SQL**.

# Основные команды SQL:

- **SELECT** – команда/ключевое поле, начинающая SQL-выражения; предшествует названию поля или полей, выбираемых из полей таблиц.
- **FROM** – указывает имя таблицы или таблиц, содержащих поля, перечисленные после команды **SELECT**.
- **ORDER BY** – команда, указывающая порядок вывода данных.

# Команда SELECT

**SELECT** Первое\_поле, второе\_поле,  
третье\_поле...

Имя поля и имя таблицы берется в [ ], если  
имя содержит пробел.

**SELECT** [Справочник клиентов].[Название  
клиента], [Справочник клиентов].Страна,  
[Справочник заказов].[Дата исполнения  
заказа]

# Предикаты SELECT

- ***ALL;***
- ***DISTINCT;***
- ***DISTINCTROW;***
- ***TOP.***

Эти предикаты служат для ограничения количества возвращаемых записей. В SQL-выражениях их можно использовать с командой WHERE.

- ❖ ***ALL*** – выбирает все записи, которые удовлетворяют условию ***WHERE***.
- ❖ ***DISTINCT*** – необходимо включать, когда из запроса следует исключить одинаковые записи.
- ❖ ***DISTINCTROW*** – проверяет совпадение в таблицах всех полей, а не только выбранных.
- ❖ ***TOP*** – ограничивает число выводимых записей, удовлетворяющих ***WHERE***.  
***Например:***  
***TOP 10 – первые 10 записей.***

**Пример:** при создании запроса, выводящего код клиента и день, в который он сделал заказ, нужно использовать следующую SQL-информацию:

***SELECT DISTINCT Заказы.[Код клиента],  
Заказы.[Дата принятия заказа]***

Данная конструкция проверяет дублирование для полей, указанных для просмотра и исключает из результирующей таблицы дублированные записи.

## Команда FROM

Команда FROM указывает таблицы (или запросы), содержащие поля в выражении SELECT.

**Пример: (для каждой таблицы)**  
***SELECT [наименование клиента],***  
***Страна,***  
***FROM [Клиенты].***

При работе с несколькими таблицами в команде FROM устанавливается объединение между несколькими таблицами, которые будут соединены в результирующей таблице.

***INNER JOIN...ON*** – обычное  
объединение (внутреннее);

*Смотрим пред. Пример!*

***FROM Клиенты INNER JOIN Заказы  
ON Клиенты.[Код клиента] =  
Заказы.[Код клиента];***

## Команда WHERE

Служит для указания требуемых условий.  
Условие WHERE может быть любым допустимым выражением, т.е. содержащим одно и более условий.

**Пример: Введем ограничение на код клиента.**

WHERE (Клиенты. [Код клиента] = 16)

**В SQL-выражениях команда WHERE должна следовать за командой FROM.**

## Команда ORDER BY

Используется для сортировки запроса.

**Пример:** ORDER BY Клиенты.[Код клиента];

**Оканчивать SQL-выражения нужно (;).**

**Пример:**

Построить SQL-выражение, которое будет выполнять следующие действия:

- 1. Выбрать поля: Название клиента, страна, код заказа и дата заказа.**
- 2. Объединить таблицы: Клиенты и Заказы по полю Код клиента.**
- 3. Вывести только те записи, которые связаны с определенным клиентом.**
- 4. Сортировка по полю дата заказа.**

```
SELECT Клиенты.[Наименование  
клиента], Заказы.[Дата принятия  
заказа], Клиенты . Страна, Заказы.[Код  
заказа]  
FROM Клиенты INNER JOIN Заказы ON  
Клиенты.[Код клиента] = Заказы.[Код  
клиента]  
WHERE (((Клиенты.[Наименование  
клиента])="Меньшов"))  
ORDER BY Заказы.[Дата принятия  
заказа];
```