

## ВАРИАТИВНОЕ ЗАДАНИЕ 4.1 КУРСА «БАЗЫ ДАННЫХ»

### НАПИСАНИЕ ЗАПРОСОВ НА ЯЗЫКЕ SQL

Данная лабораторная работа посвящена созданию запросов. Мы уже работали с запросами с помощью конструктора, сегодня же мы рассмотрим еще один режим работы с запросами – режим SQL. Знание SQL нередко позволяет быстрее написать сложные запросы или разобраться при возникновении каких-то проблем.

*Задание.*

1. *Добавьте в таблицу Book несколько книг одного автора. (Добавление происходит либо через форму, либо напрямую в таблицу).*

2. *В конструкторе создайте запрос, выводящий список авторов книг из таблицы Book (нам нужны только авторы, без названий книг и т.д.). Выполните запрос и обратите внимание на то, что если автор написал несколько книжек, то в результатах запроса он будет фигурировать тоже несколько раз.*

3. *Перейдите в режим SQL, в котором Вы увидите то предложение SQL, которое и является запросом:*

`SELECT Book.Author FROM Book;`

(Сохраните запрос 0\_Все авторы\_повторения)

В SQL для указания на то, что нужно отбрасывать повторения, служит ключевое слово DISTINCT.

*Задание.*

1. *Измените запрос так, как показано ниже и проверьте его работу.*

`SELECT DISTINCT Book.Author FROM Book;`

(Сохраните запрос 1\_Авторы без повторений)

Обратите внимание, что при выполнении второго запроса, кроме отбрасывания повторяющихся записей, может поменяться порядок их вывода. Если требуется задать определенный порядок сортировки выводимых записей, в режиме SQL это можно сделать, добавив инструкцию ORDER BY, после указывается столбец или столбцы, по которым проводится сортировка. По

умолчанию или при явном добавлении ключевого слова ASC сортировка проводится по возрастанию, если стоит DESC – по убыванию.

*Задание.*

*1. Создайте запрос, который выводит список авторов без повторений в алфавитном порядке.*

```
SELECT DISTINCT Book.Author FROM Book ORDER BY Book.Author;
```

(Сохраните запрос 2\_Авторы без повторений сортировка 1)

*2. Создайте запрос, который выводит без повторения перечень авторов и годов издания книг, причем сортировка по первому столбцу производится по возрастанию, а среди записей с одинаковыми значением первого поля, сортировка производится по году издания книг по убыванию*

```
SELECT DISTINCT Book.Author, Book.BookYear  
FROM Book  
ORDER BY Book.Author ASC, Book.BookYear DESC;
```

(Сохраните запрос 3\_Авторы без повторений сортировка 2)

*3. Напишите запрос, выводящий информацию о книгах из таблицы Book в порядке от самых старых к самым новым (все поля таблицы Book должны быть отображены). Книги с одинаковым годом издания упорядочивайте в списке по названию в алфавитном порядке.*

(Сохраните запрос 4\_Книги сортировка)

В некоторых случаях нужно вывести не весь сортированный список, а какую-то его часть. Например, нас могут интересовать первые 5 записей из списка или первые 10 процентов списка. В подобных случаях, в запрос SELECT добавляют соответственно «TOP n» или «TOP n PERCENT», где n – количество записей или процент. Например, следующий запрос выведет список из двух самых новых изданий из Book (символ \* указывает на то, что нам нужны все столбцы).

*Задание.*

*1. Создайте новый запрос так, как показано ниже и проверьте его работу.*

```
SELECT TOP 2 *  
FROM Book  
ORDER BY Book.BookYear DESC;
```

(Сохраните запрос 5\_Две последние книги)

*2. Напишите запрос, выводящий информацию о книгах, составляющих первую половину общего списка из таблицы Book, упорядоченного по возрастанию года (т.е. самые старые книги).*

(Сохраните запрос 6\_Половина списка)

Следующий запрос, использует данные из нескольких связанных таблиц.

```
SELECT BookInLib.LibID, Book.Author, Book.Title
```

```
FROM Book INNER JOIN BookInLib
```

```
ON Book.BookID = BookInLib.BookID;
```

(Сохраните запрос 7\_Запрос к связанным таблицам)

Здесь инструкция INNER JOIN указывает на то, что надо соединить таблицы по условию совпадения значений в столбцах Book.BookID и BookInLib.BookID. Причем соединение должно быть «внутренним» – соответствующие значения должны встречаться как в одной, так и в другой таблице. Кроме внутреннего, Access также поддерживает левое и правое внешнее соединение – LEFT JOIN и RIGHT JOIN. В первом случае в выборку берутся все записи из левой таблицы и соответствующие им записи из правой, если они есть. Во втором случае – наоборот.

*Задание.*

*1. Скопируйте приведенный выше запрос в свою базу. Выполните его. При необходимости измените набор данных в таблицах так, чтобы в таблице Book были записи не связанные с записями из таблицы BookInLib.*

(Сохраните запрос 7\_Запрос к связанным таблицам Inner)

*2. В запросе замените INNER JOIN на LEFT JOIN, проанализируйте результат.*

(Сохраните запрос 8\_Запрос к связанным таблицам Left)

В предложениях SELECT, условия, накладываемые на отбираемые записи, указываются после ключевого слова WHERE.

*Задание.*

*1. Создайте запрос на выборку книг, выпущенных с 1997 по 2001 год.*

```
SELECT Book.Author, Book.Title, Book.Publisher, Book.BookYear
```

```
FROM Book
```

```
WHERE Book.BookYear BETWEEN 1997 and 2001;
```

(Сохраните запрос 9\_книги с 1997 по 2001)

Как мы разбирали в предыдущей лабораторной работе, условия могут включать проверку на равенство, неравенство, принадлежность множеству (IN), соответствие текстовой маске (LIKE) и т.д. Условие также может быть составным и включать логические операторы NOT (отрицание), AND (и), OR (или); их приоритет убывает именно в такой последовательности. Например, пусть кроме условия на год издания, надо проверить принадлежность издательства заранее определённого списку. Запрос может выглядеть следующим образом.

```
SELECT Book.Author, Book.Title, Book.Publisher, Book.BookYear
FROM Book
WHERE Book.BookYear BETWEEN 1997 AND 2001 AND Book.Publisher IN
('Азбука', 'Политехника');
```

*Задание.*

*1. Запустите запрос описанный в примере выше.*

*(Сохраните запрос 10\_книги с 1997 по 2001 IN)*

*2. Создайте запрос, эквивалентный предыдущему с использованием логических операторов (AND, OR). (Первые строчки остаются неизменными. Меняется лишь часть условия – часть, отвечающая за Book.Publisher). Обратите внимание на использование скобок!*

*(Сохраните запрос 11\_книги с 1997 по 2001 OR)*

*3. Напишите на SQL запрос, который выведет в обратном алфавитном порядке названия книг, выпущенных после 1999 года, фамилия автора (значение поля Author) которых начинается с буквы «Г» или название издательства заканчивается буквой «а».*

*Порядок следования команд: SELECT, FROM, WHERE, ORDER BY*

*(Сохраните запрос 12\_Авторы Like)*

*4. Опишите, что на Ваш взгляд делает приведённый ниже запрос. Скопируйте его код в Access и выполните.*

```
SELECT *
FROM Book
WHERE Author LIKE 'A*' OR BookYear > 2000 AND NOT Publisher LIKE
'[И,П]*';
```