

Работа № 4

Исследование возможностей Microsoft Access по формированию запросов с использованием SQL.

Цель: 1. Изучить порядок формирования запросов с использованием SQL в Microsoft Access.

2. Получить практические навыки формирования запросов с использованием SQL.

3. Исследовать возможности Microsoft Access по формированию запросов к таблицам с использованием SQL.

SQL — это аббревиатура выражения Structured Query Language (язык структурированных запросов), язык, который дает вам возможность создавать и работать в реляционных базах данных.

Преимуществами SQL являются:

1. Он не относится к числу патентованных языков, используемых разработчиками определенных баз данных. Почти все большие СУБД поддерживают SQL.
2. Несмотря на кажущуюся простоту, SQL является очень мощным языком; пользуясь его элементами, можно выполнять очень сложные операции с базами данных.

Запрос - команда, которая сообщает базе данных, чтобы она вывела определенную информацию из таблиц в память. Эта информация обычно посылается непосредственно на экран компьютера, но ее можно также послать принтеру, сохранить в файле или представить как вводную информацию для другой команды или процесса.

Все запросы в SQL состоят из одиночной команды - **SELECT (ВЫБОР)**. Данная команда инструктирует базу данных, чтобы извлечь информацию из таблицы. Например, можно вывести таблицу «Продавцы», напечатав следующее:

**SELECT КодПродавца, Наименование, Адрес, Комиссионные
FROM Продавцы;**

Вывод для этого запроса показывается в Рисунке 1.

Вывод данных из таблицы Продавцы			
КодПродавца	Наименование	Адрес	Комиссионные
1001	Пеел	Лондон	0,12
1002	Серрес	Москва	0,13
1003	Акселрод	Нью Йорк	0,1
1004	Мотика	Лондон	0,11
1007	Рифкин	Барселона	0,15
*			

Рисунок 1 – Запрос «Вывод данных из таблицы «Продавцы»

Объяснение каждой части этой команды:

SELECT	Ключевое слово, которое сообщает базе данных, что эта команда - запрос. Все запросы начинаются этим словом, сопровождаемым пробелом.
КодПродавца, Наименование, Адрес, Комиссионные	Это список столбцов из таблицы, которые выбираются запросом. Любые столбцы, не перечисленные здесь, не будут включены в вывод команды. Это не значит, что они будут удалены или их информация будет стерта из таблиц, потому что запрос не воздействует на информацию в таблицах, он только показывает данные.
FROM Продавцы	Ключевое слово, подобно SELECT, которое должно быть представлено в каждом запросе. Оно сопровождается пробелом и затем именем таблицы используемой в качестве источника информации. В данном случае — это таблица «Продавцы».
;	Точка с запятой используется во всех интерактивных командах SQL, чтобы сообщать базе данных, что команда заполнена и готова выполниться.

Запрос такого характера не обязательно будет упорядочивать вывод данных. Та же самая команда, выполненная с теми же самыми данными, но в разное время не сможет вывести данные в том же порядке. Обычно, строки выводятся в том порядке, в котором они найдены в таблице. Это не обязательно будет тот порядок, в котором данные вводились или сохранялись. В отсутствии явного упорядочения в выводе данных нет никакого определенного порядка.

Оператор SQL можно разбить на несколько строк для облегчения чтения и понимания структуры команд SQL.

Если необходимо вывести все столбцы таблицы, имеется необязательное сокращение, которое можно использовать. Звездочка (*) может применяться для вывода полного списка столбцов следующим образом:


```
SELECT *
FROM Продавцы;
```

Это приведет к тому же результату, что и предыдущая команда.

Команда SELECT способна извлечь строго определенную информацию из таблицы. Есть возможность увидеть только определенные столбцы таблицы. Это выполняется легко, простым исключением столбцов из части команды SELECT. Например, запрос:

SELECT КодПродавца, Наименование FROM Продавцы;

будет производить вывод, показанный на рисунке 2.



КодПродавца	Наименование
1001	Пеел
1002	Серрес
1003	Акселрод
1004	Мотика
1007	Рифкин
*	

Рисунок 2 – Вывод определенных столбцов из таблицы «Продавцы»

Даже если столбцы таблицы, по определению, упорядочены, это не означает, что они будут выводиться в том же порядке. Оператор "SELECT *..." покажет все столбцы в их естественном порядке, но, если указать столбцы отдельно, можно получить их в том порядке, в котором необходимо.

Например, рассмотрим таблицу «Заказы», содержащую поле «ДатаПриобретения», «КодПродавца», «КодЗаказа» и «Сумма»:

SELECT ДатаПриобретения, КодПродавца, КодЗаказа, Сумма FROM Заказы;

Вывод этого запроса показан на рисунке 3.



ДатаПриобретения	КодПродавца	КодЗаказа	Сумма
10.03.2011	1007	3001	18,69
10.03.2011	1004	3002	1900,1
10.03.2011	1001	3003	767,19
10.03.2011	1002	3005	5160,45
10.03.2011	1007	3006	1098,16
10.04.2011	1002	3007	75,75
10.05.2011	1001	3008	4723
10.04.2011	1003	3009	1713,23
10.06.2011	1002	3010	1309,95
10.06.2011	1001	3011	9891,88
*			

Рисунок 3 – Запрос «Вывод данных из таблицы «Заказы»

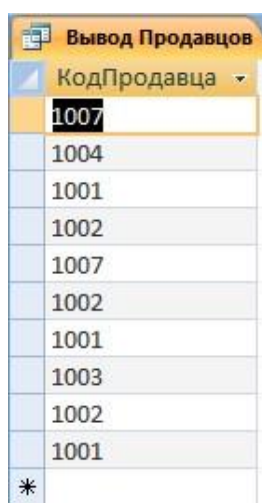
Таким образом, структура информации в таблицах — это просто основа для активной перестройки структуры в SQL.

Удаление избыточных данных.

DISTINCT (ОТЛИЧИЕ) – это аргумент, который обеспечивает устранение двойных значений из предложения SELECT.

Предположим, что необходимо знать, какие продавцы в настоящее время имеют свои заказы в таблице «Заказы». Не нужно знать, сколько заказов имеет каждый, требуется только вывести список номеров продавцов. Для этого создается следующий запрос:

```
SELECT КодПродавца  
FROM Заказы;
```

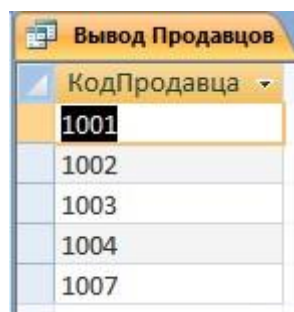


КодПродавца
1007
1004
1001
1002
1007
1002
1001
1003
1002
1001
*

Рисунок 4 – Запрос с дублированием номеров продавцов

Для получения списка без дубликатов, необходимо создать следующий запрос:

```
SELECT DISTINCT КодПродавца  
FROM Заказы;
```



КодПродавца
1001
1002
1003
1004
1007

Рисунок 4 – Запрос без дублирования номеров продавцов

Другими словами, DISTINCT следит за тем, какие значения были ранее, так, чтобы они не были продублированы в списке. Это полезный способ избежать избыточности данных.

Параметры DISTINCT.

DISTINCT может указываться только один раз в предложении SELECT. Если предложение выбирает многочисленные поля, DISTINCT опускает строки, где все выбранные поля идентичны. Строки, в которых некоторые значения одинаковы, а некоторые различны, будут сохранены. DISTINCT, фактически, приводит к показу всей строки вывода, не указывая полей, так что нет никакого смысла, чтобы его повторять.

Оператор WHERE.

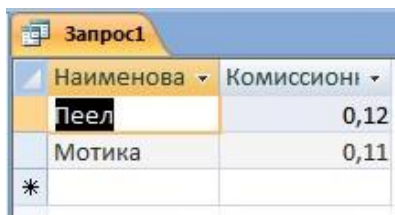
Таблицы имеют тенденцию становиться очень большими, с течением времени все большее и большее количество строк в них добавляется. Поскольку обычно только определенные данные интересуют пользователя в данное время, SQL дает возможность устанавливать критерии, чтобы определить, какие строки будут выбраны для вывода.

WHERE (ГДЕ) — это предложение команды SELECT, которое позволяет устанавливать предикаты, условие которых может быть или верным или неверным для любой строки таблицы. Команда извлекает только те строки из таблицы, для которых такое утверждение верно.

Например, необходимо вывести наименование и комиссионные всех продавцов в Лондоне. Для этого требуется создать следующий запрос:

```
SELECT Наименование, Комиссионные  
FROM Продавцы  
WHERE Адрес = «Лондон»;
```

Когда предложение WHERE представлено, программа базы данных просматривает всю таблицу по одной строке и исследует каждую строку, чтобы определить, верно ли утверждение. Следовательно, для записи «Пеел» программа рассмотрит текущее значение столбца «Адрес», определит, что оно равно 'Лондон', и включит эту строку в вывод. Запись для «Серрес» не будет включена, и так далее.



Наименование	Комиссионные
Пеел	0,12
Мотика	0,11
*	

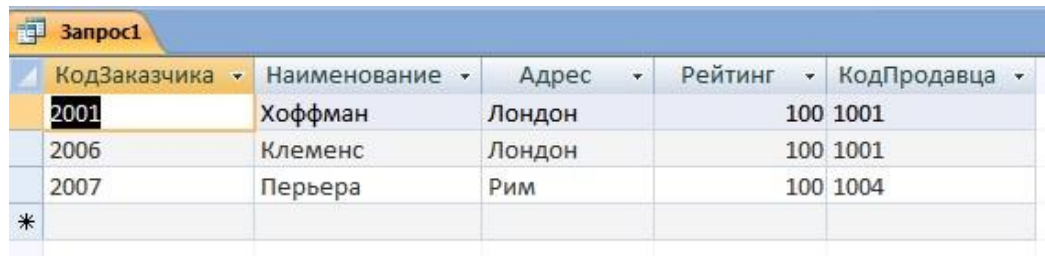
Рисунок 5 – Оператор SELECT с предложением WHERE

В предложении WHERE можно также использовать числовые поля. Поле «Рейтинг» таблицы «Заказчики» предназначено, чтобы разделять заказчиков на группы, основанные на некоторых критериях, которые могут быть получены в итоге через этот номер. Такие числовые коды могут быть полезны в реляционных базах данных как способ подведения итогов сложной

информации. Можно выбрать всех заказчиков с рейтингом 100 следующим образом:

```
SELECT *  
FROM Заказчики  
WHERE Рейтинг = 100;
```

Одиночные кавычки в данном случае не используются, так как «Рейтинг» является числовым полем.



КодЗаказчика	Наименование	Адрес	Рейтинг	КодПродавца
2001	Хоффман	Лондон	100	1001
2006	Клеменс	Лондон	100	1001
2007	Перьера	Рим	100	1004
*				

Рисунок 6 – Оператор SELECT с числовым полем в предикате

Можно использовать номера столбцов, устранять дубликаты или переупорядочивать столбцы в команде SELECT, которая использует WHERE. Однако, можно изменять порядок столбцов для имен только в предложении SELECT, но не в предложении WHERE.

ВНИМАНИЕ! Для того чтобы создать запрос в режиме SQL необходимо перейти на ленте на вкладку «Создание» и нажать на кнопку «Конструктор запросов». После этого открывается конструктор запросов и окно «Добавление таблицы», которое необходимо закрыть. Далее по кнопке «Режимы» выбираем «Режим SQL» (рисунок 7).

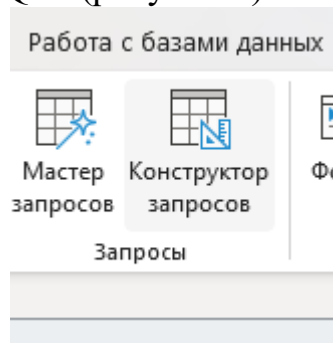


Рисунок 7 – Режимы, создания запросов

Задание. Написать запросы для выполнения следующих условий:

1. Напишите команду SELECT, которая бы вывела часть столбцов одной из ваших таблиц. 1 запрос.
2. Аналогично предыдущему, с использованием функций (например, умножения на число). 1 запрос.
3. Напишите запрос, который вывел бы часть строк таблицы в соответствии с заданным условием. 2 запроса.

4. Напишите запрос, который вывел бы столбец одной из ваших таблиц без дублирования строк. 1 запрос.