

Лабораторная работа №2

Исследование возможностей Microsoft Access по формированию запросов с использованием SQL.

Цель:

1. Получить практические навыки формирования запросов с использованием предикатов оператора SELECT SQL.
2. Исследовать возможности Microsoft Access по формированию запросов к таблицам с использованием SQL.

Предикаты, используемые в запросах:

1. IN:

field IN (список значений)

– определяет множество значений, с которыми будет сравниваться значение указанного поля field. Предикат считается истинным, если значение поля field равно хотя бы одному из элементов множества.

Пример с использованием предиката IN:

Список мониторов продавцов «альфа» и «бета»:

```
select * from монитор where продавец IN ("альфа","бета")
```

2. BETWEEN:

field BETWEEN значение1 AND значение2

– определяет, входит ли значение поля field в указанные границы. Если значение поля меньше, чем значение1, или больше, чем значение2, предикат возвращает "ложь".

Пример с использованием предиката BETWEEN:

Вывести модель, дату выпуска и цену мониторов производства 2017 года:

```
select модель, выпуск, цена from монитор where выпуск  
BETWEEN #01/01/20# AND #12/31/20#
```

3. LIKE:

field LIKE 'образец'

– используется для поиска подстрок, применяется *только к полям символьного типа*. Возможно использование шаблонов: '?' – один любой символ и '*' – произвольное количество символов (в т.ч., ни одного);

Примеры с использованием предиката LIKE:

Вывести модель, дату выпуска и цену мониторов, в названии которых есть сочетание «NEC»:

```
select модель, выпуск, цена from монитор where модель like "*NEC*"
```

Список продавцов, в наименовании которых вторая буква «е»:

```
select * from продавец where наимен like "?е*"
```

4. Оператор SQL NOT служит для задания противоположно заданного условия

В следующем примере показано возвращение списка всех продуктов с ценой меньше 10 долларов и больше 100 долларов.

```
SELECT ProductID, ProductName, UnitPrice FROM Products  
WHERE UnitPrice NOT BETWEEN 10 AND 100
```

5. Функции агрегирования:

COUNT – определяет в результате количество строк (записей).

SUM – определяет арифметическую сумму значений указанного числового поля в результирующем множестве записей.

AVG – определяет среднее арифметическое значений указанного числового поля в результирующем множестве записей.

MAX, MIN – определяет максимальное (минимальное) значение указанного поля в результирующем множестве.

Пример

Посчитать сумму цен мониторов по продавцам (группировка):

```
select продавец, sum(цена) from монитор group by продавец;
```

6. Сортировка результатов осуществляется использованием ORDER BY.

Результирующие записи упорядочивает по значению одного или нескольких полей: ASC – по возрастанию, DESC – по убыванию.

Пример:

Список мониторов, упорядоченный по наименованию продавцов (в алфавитном порядке) и по уменьшению цены для каждого продавца:

```
select * from монитор order by продавец, цена desc
```

7. Запрос SELECT на нескольких таблицах реализует декартово произведение исходных таблиц (или их соединение, если указать условия соответствия значений полей разных таблиц). Для полей нужно указывать имя таблицы перед именем поля.

Пример:

Запрос по двум таблицам. Список наименований моделей мониторов с ценами и адресами продавцов:

select [монитор].[модель], [монитор].[цена], [продавец].[наимен] from
монитор, продавец where [монитор].[продавец]=[продавец].[наимен];

Задание. Написать запросы для выполнения следующих условий:

1. Напишите команду SELECT с использованием предиката IN. 1
запрос.
2. Напишите команду SELECT с использованием предиката BETWEEN.
1 запрос.
3. Напишите команду SELECT с использованием предиката LIKE. 1
запроса.
4. Напишите команду SELECT с использованием оператора SQL NOT.
1 запрос.
5. Напишите команду SELECT с использованием группировки. 1
запрос.
6. Напишите команду SELECT с использованием сортировки. 1 запрос.
7. Напишите команду SELECT к двум таблицам. 1 запрос.