

## **Практическая работа №3.**

### **Индексирование и сортировка таблиц.**

**Цель занятия:** Научиться создавать индексы для таблиц базы данных, организовывать простую и сложную сортировку записей таблицы.

#### *Теоретическая часть*

### **Индексирование полей и записей в базе данных Access**

При помощи индексов ускоряется сортировка и поиск записей, а также обеспечивается уникальность данных. Индексы таблиц Microsoft Access используются так же, как и предметные указатели в книгах: при поиске данных выполняется их поиск в индексе. Индексы можно создавать по одному или нескольким полям. Составные индексы позволяют пользователю различать записи, в которых первые поля могут иметь одинаковые значения.

Концептуально действие индекса состоит в следующем. В индексе содержатся упорядоченный список значений колонки или комбинации колонок, а также сведения о местонахождении на жестком диске соответствующих этим значениям строк таблицы. Значения колонки в индексе упорядочены. Несмотря на то, что порядок строк в таблице случаен, индекс можно быстро просмотреть, чтобы найти конкретное значение. Упорядоченный индекс можно просмотреть во много раз быстрее, чем неупорядоченную таблицу.

Индексы занимают место в БД. При вводе новых данных или удалении данных СУБД приходится обновлять и таблицы, и индексы. Это может замедлить выполнение операций модификации данных, особенно для таблиц с большим числом строк, как в ХД. Таким образом, может появиться проблема, суть которой состоит в возникновении конфликта между скоростью обновления данных в таблице и скоростью ее считываний. При разрешении этой проблемы следует придерживаться следующего эмпирического правила: создавать индексы для колонок первичных ключей и других колонок, часто используемых в тех запросах, в которых для выборки данных применяются логические критерии. Если в результате скорость обновления данных ухудшается, то можно рассмотреть вопрос об удалении некоторых индексов.

### **Выбор полей для индексирования**

В основном, требуется индексировать поля, в которых часто осуществляется поиск, поля сортировки или поля, объединенные с полями из других таблиц в запросах.

Хорошими кандидатами для индексирования обычно являются:

- Поля первичного ключа. По определению, поля первичного ключа должны иметь уникальный индекс.
- Поля внешнего ключа. Они дают хороший индекс по двум причинам. Во-первых, они часто применяются для выполнения соединений с родительскими таблицами. Во-вторых, они могут быть использованы СУБД при поддержке ссылочной целостности в операциях удаления строк родительской и дочерних таблиц.
- Любые поля, которые содержат уникальные значения.
- Поля, запросы или соединения, которые используют от 5 до 10% строк таблицы.
- Поля, которые часто входят как аргументы в функции агрегирования.
- Поля, где предполагается выполнение поиска значений.
- Поля, где предполагается выполнение сортировки значений.

### **Составные индексы**

Если предполагается частое выполнение одновременной сортировки или поиска в нескольких полях, можно создать для этих полей составной индекс. Например, если в одном и том

же запросе часто задаются условия для полей «Имя» и «Фамилия», то для этих двух полей имеет смысл создать составной индекс.

*При сортировке таблицы по составному индексу Microsoft Access сначала выполняет сортировку по первому полю, определенному для данного индекса. Если в первом поле содержатся записи с повторяющимися значениями, то выполняется сортировка по второму полю, определенному для данного индекса, и так далее. В составной индекс можно включить до 10 полей.*

## Создание индекса

Перед созданием индекса необходимо решить, следует ли создать индекс для одного поля или составной индекс. Индекс для одного поля создается с помощью установки свойства Индексированное поле. В следующей таблице приведены возможные параметры свойства Индексированное поле.

**Таблица 1. Параметры свойства "Индексированное поле"**

Параметр свойства	Значение
Нет	Не создавать индекс для этого поля (или удалить существующий индекс)
Да (Допускаются совпадения)	Создать индекс для этого поля
Да (Совпадения не допускаются)	Создать уникальный индекс для этого поля

При создании уникального индекса невозможно ввести новое значение в определенном поле, если такое значение уже существует в том же поле другой записи. В Access уникальный индекс автоматически создается для первичных ключей, однако может понадобиться, чтобы создание значений, совпадающих со значениями в других полях, было невозможным. Например, можно создать уникальный индекс для поля, в котором содержатся серийные номера, чтобы двум продуктам не мог быть присвоен один и тот же серийный номер.

## Создание индекса для одного поля

- Откройте таблицу в режиме **Конструктор**.
- Щелкните на поле, которое следует индексировать.
- В разделе **Свойства поля** вкладка **Общие** выберите свойство **Индексированное поле** и выберите режим индексации.
- Сохраните изменения, в **Конструкторе**.

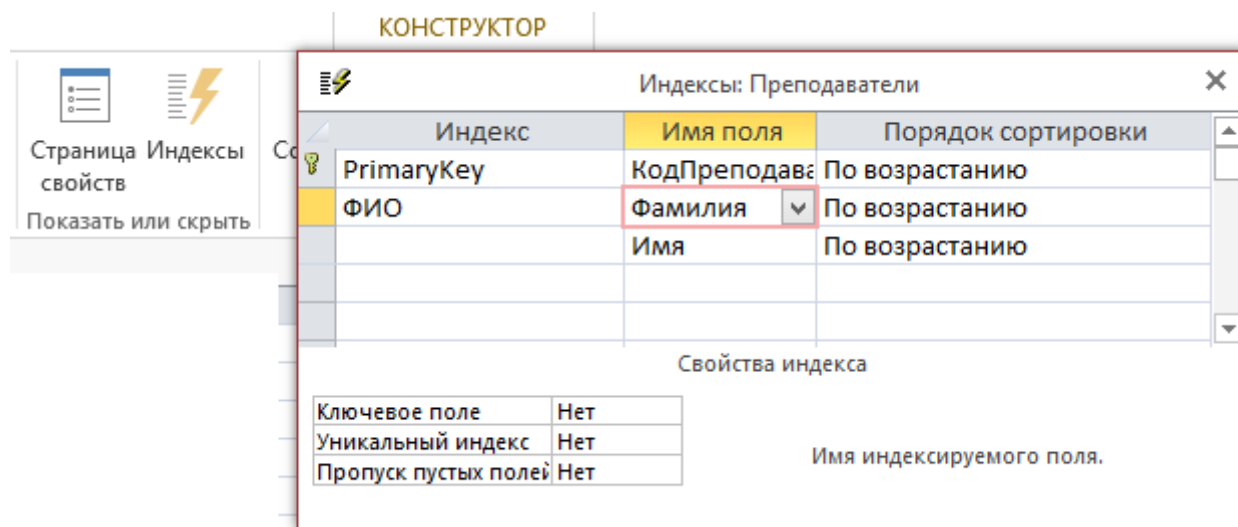
## Создание составного индекса

Чтобы создать составной индекс, необходимо включить в него строку для каждого поля в индексе и поместить название индекса только в самой первой строке. Все строки обрабатываются как часть одного индекса до тех пор, пока не будет обнаружена строка с другим названием индекса.

- Откройте таблицу в режиме **Конструктор**.
- На вкладке **Конструктор** в группе **Показать или скрыть** щелкните кнопку **Индексы**. Появится окно «Индексы».
- Введите имя индекса. Для индекса можно использовать либо имя одного из индексируемых полей, либо другое подходящее имя.
- В столбце **Имя поля** выберите первое поле, которое следует использовать в индексе.
- Следующую строку столбца **Индекс** оставьте пустой, затем в столбце **Имя поля** выберите второе индексируемое поле. Повторите этот шаг для всех полей, которые необходимо включить в индекс.
- В разделе **Свойства индекса** установите свойства индекса в соответствии со следующей таблицей.
- Сохраните изменения, в **Конструкторе**.

**Таблица 2. Свойства индексов**

Надпись	Значение
Ключевое поле	Если Да, то индекс является первичным ключом.
Уникальный индекс	Если Да, то каждое индексируемое значение должно быть уникальным.
Пропуск пустых полей	Если Да, то записи с пустыми значениями в индексируемых полях будут исключены из индекса.



## Сортировка записей

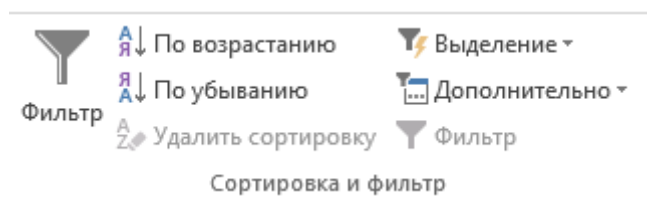
Существует два вида сортировки, которые можно выполнить: простая сортировка и сложная сортировка.

- **Простая сортировка.** Выполняется в режиме формы, в режиме таблицы, все записи одного поля сортируются по возрастанию или по убыванию (но не в том и другом порядке сортировки одновременно).

- **Сложная сортировка.** Выполняется в режиме конструктора запроса, в окне расширенного фильтра, в режиме конструктора отчета, в режиме конструктора страницы, в режиме сводной диаграммы или сводной таблицы. Это означает, что по некоторым полям допускается сортировка по возрастанию, а по другим полям сортировка по убыванию.

### Простая сортировка

- В режиме таблицы выберите поле и установите режим сортировки.



### Сложная сортировка

Рассмотрим один из способов создания сложной сортировки:

- Для выбранной таблицы выберите **Дополнительно – Расширенный фильтр**. Отобразится выбранная таблица, и конструктор фильтра с настройки параметров отображения полей.
- Двойным щелчком мыши выберите из таблицы поля для настройки, которые отобразятся в нижней части конструктора.

- Укажите порядок сортировки для выбранных полей. Сохраните полученный запрос и дайте ему имя.

Зачет

- \*
  - НомСтуденческогоБилета
  - КодПредмета
  - Семестр
  - Допуск
  - Оценка
  - ДатаСдачи

Поле: КодПредмета Семестр

Имя таблицы: Зачет Зачет

Сортировка: по возрастанию по возрастанию

Вывод на экран: ☐ ☐ ☐

Условие отбора:

или:

### Практическое задание

- Отсортируйте таблицу «*Студенты*» по убыванию Даты рождения. Скопируйте полученную таблицу в текстовый документ.
- Отсортируйте таблицу «*Студенты*» по возрастанию Номера группы. Скопируйте полученную таблицу в текстовый документ.
- Добавьте в таблицу «*Экзамен*» следующие записи:

№ студ. билета	Код предмета	Семестр	Допуск	Оценка	Дата сдачи
123432	7	4	Да	5	22.07.2016
233114	6	3	Нет		
232123	9	5	Нет		
234212	11	8	Нет		
234212	10	8	Да	4	22.07.2016
234212	1	3	Да	5	21.09.2014

- Отсортируйте в таблице «*Экзамен*» записи по возрастанию Кода предмета и семестра. Запрос назовите *СортировкаПредметСеместр*. Скопируйте полученную таблицу в текстовый документ.
- Переиндексируйте таблицу «*Экзамен*» следующим образом, чтобы сортировка и поиск записей сначала осуществлялись по полю Семестр (по возрастанию), затем по Коду предмета (по возрастанию), и в последнюю очередь по Номеру студенческого билета (по убыванию). Откройте таблицу «*Экзамен*» в режиме просмотра записей, скопируйте таблицу в текстовый документ.

Индексы: Экзамен

Индекс	Имя поля	Порядок сортировки
PrimaryKey	Семестр	По возрастанию
	Код предмета	По возрастанию
	№ студенческого би	По убыванию

Свойства индекса

Ключевое поле: Да

Уникальный индекс: Да

Пропуск пустых полей: Нет

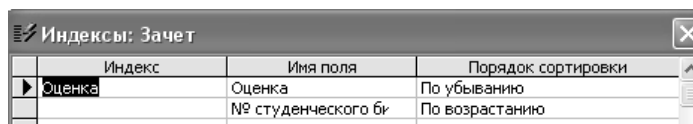
Имя данного индекса: Каждый индекс имеет

- В конструкторе таблиц для таблицы «*Зачет*» отмените Первичный ключ таблицы.

7. Добавьте следующие строки в таблицу «**Зачет**». Дату сдачи заполните самостоятельно.

№ студ. билета	Код предмета	Семестр	Допуск	Оценка	Дата сдачи
232123	4	1	Да	4	
234213	11	6	Да	4	
234213	1	2	Да	3	
232123	4	2	Да	5	
232123	9	4	Да	5	
234213	1	1	Да	5	
233114	5	1	Да	5	
233114	5	2	Да	5	
233114	6	3	Да	5	
234213	11	7	Да	3	
234213	3	3	Да	3	
234213	3	2	Да	3	

8. Создайте для таблицы «**Зачет**» новый индекс по полям - Оценка, в котором Оценки отсортированы по убыванию, а № студенческого билета по возрастанию. Полученную таблицу скопируйте в текстовый документ.



9. Переиндексируйте таблицу «**Зачет**» таким образом, чтобы Оценка была отсортирована по убыванию, Код предмета по возрастанию, Семестр - по убыванию. Индекс назовите – Семестр. Результат скопируйте в текстовый документ.

10. Отмените для таблицы «**РодителиСтудентов**» ключевое поле. Создайте для таблицы новый индекс - Родители, в котором КодРодителя отсортированы по убыванию, а № студенческого билета по возрастанию. Результат скопируйте в текстовый документ.

11. При помощи расширенного фильтра создайте сложную сортировку записей в таблице «**Преподаватели**», в которой производится сортировка записей по полю Стаж работы по убыванию, а по полю Категория - по возрастанию. Сохраните данный расширенный фильтр в виде запроса СортСтажКатегория. Результат запроса при этом будет находиться в объектах Запросы. Скопируйте результат в текстовый документ.

12. При помощи расширенного фильтра задайте сортировку таблицы «**Студенты**» по Группе - по убыванию, по Дате рождения - по возрастанию. Результат скопируйте в текстовый документ. Запрос назовите СортГруппаДата»

13. При помощи расширенного фильтра задайте сложную сортировку таблицы «**Предмет**» по полям КодСпециальности по убыванию, КоличествоЧасов по возрастанию. Сохраните его как запрос «СортСпецЧасы».

14. В таблице «**Зачет**». Установите, напротив полей НомСтуденческогоБилета, КодПредмета, Семестр идентификатор составного первичного ключа.

15. В таблице «**Родители студентов**». Установите, напротив полей НомСтуденческогоБилета, СтатусРодителя идентификатор составного первичного ключа

16. Оформите полученный отчет в текстовом документе, где для каждой таблицы укажите номер пункта задания к какому относится эта таблица и способ реализованной сортировки.

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое индекс и для чего применяется индексирование полей?
2. Какие поля таблицы индексируются автоматически?

3. В каком порядке выполняется сортировка записей в сложных индексах?
4. Сколько существует видов сортировки, и в чем их отличие?
5. Каким образом сортируются записи типа Дата\время?
6. Как сортируются числа, хранящиеся в текстовых полях?