

## Лабораторная работа № 2

### Создание базы данных, состоящей из двух таблиц

#### Теоретические сведения к лабораторной работе.

**I. Создание и использование индексов.** С целью ускорения поиска и сортировки данных в любой СУБД используются **индексы**. **Индекс** является средством, которое обеспечивает быстрый доступ к данным в таблице на основе значений одного или нескольких столбцов. Индекс представляет собой упорядоченный список значений и ссылок на те записи, в которых хранятся эти значения. Чтобы найти нужные записи, СУБД сначала ищет требуемое значение в индексе, а затем по ссылкам быстро отбирает соответствующие записи. Индексы бывают двух типов: **простые** и **составные**. **Простые индексы** представляют собой индексы, созданные по одному столбцу. **Индекс**, построенный по нескольким столбцам, называется **составным**. Примером составного индекса может быть индекс, построенный по столбцам «**Фамилия**» и «**Имя**». Однако применение индексов приносит не только преимущества, но и недостатки. Главным среди них является то, что при добавлении и удалении записей или при обновлении значений в индексном столбце требуется обновлять индекс, что при большом количестве индексов в таблице может замедлять работу. Поэтому индексы обычно рекомендуется создавать только для тех столбцов таблицы, по которым наиболее часто выполняется поиск записей. Во многих СУБД (например, FoxPro) индексы хранятся в отдельных файлах и являются предметом заботы разработчиков, т. к. при нарушении индекса поиск данных выполняется некорректно. В Microsoft Access индексы хранятся в том же файле базы данных, что и таблицы и другие объекты Access. Индексировать можно любые поля, кроме MEMO-полей, полей типа Гиперссылка и объектов OLE.

Чтобы создать **простой индекс**, необходимо:

1. Открыть таблицу в режиме **Конструктора**.
2. Выбрать поле, для которого требуется создать индекс.
3. Открыть вкладку **Общие (General)** и выбрать для свойства **Индексированное поле (Indexed)** значение **Да (Допускаются совпадения) (Yes (Duplicates OK))** или **Да (Совпадения не допускаются) (Yes (No duplicates))** (рис. 2.1).

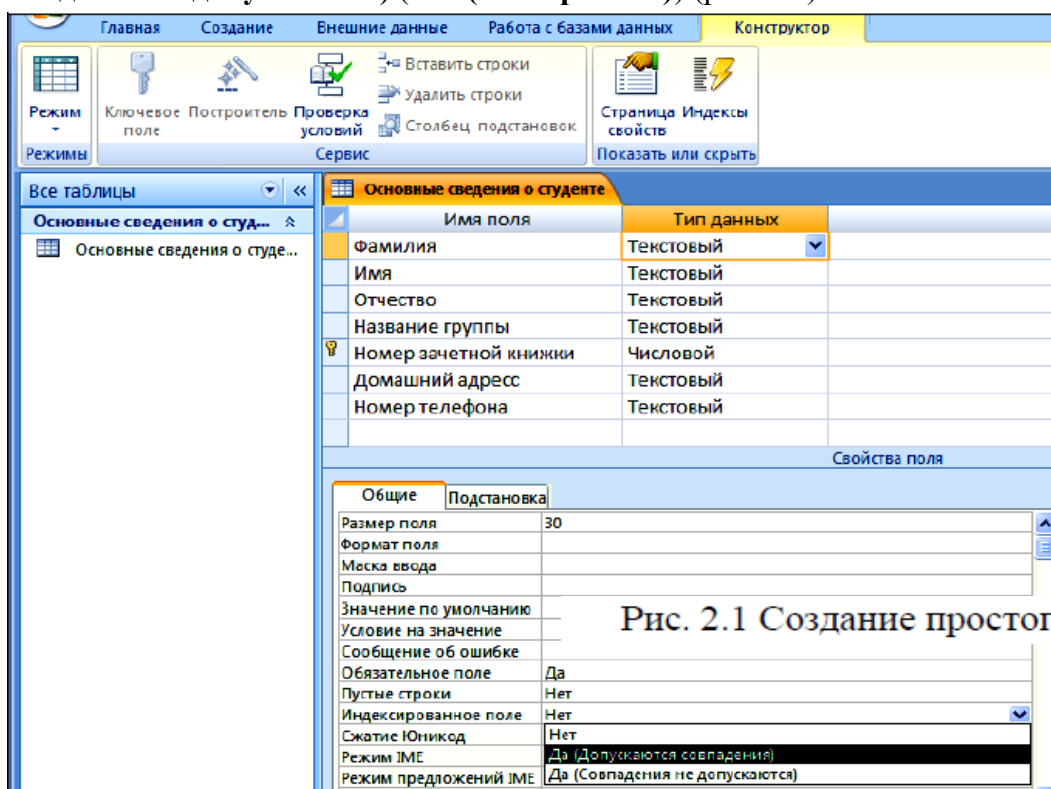


Рис. 2.1 Создание простого индекса

Ключевое поле таблицы автоматически индексируется и свойству **Индексированное поле (Indexed)** присваивается значение **Да (Совпадения не допускаются) (Yes (No duplicates))**.

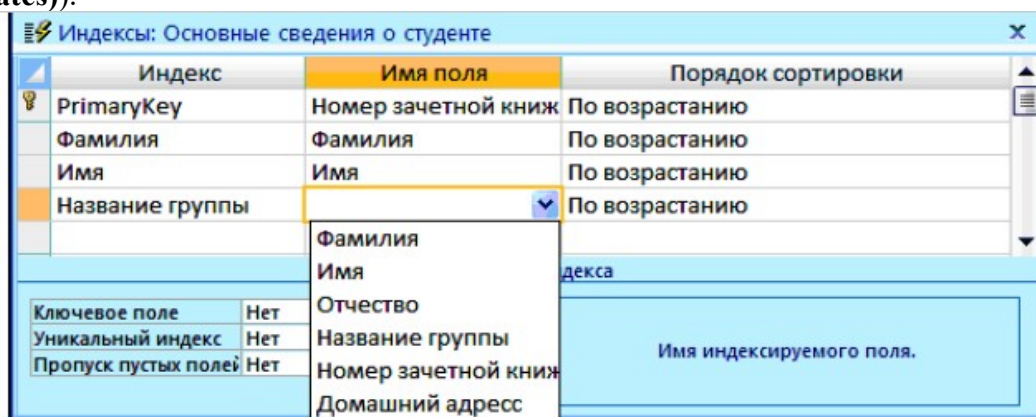


Рис. 2.2. Диалоговое окно **Индексы**

Например, создадим составной индекс «**Фамилия и номер телефона**» (рис. 2.3).

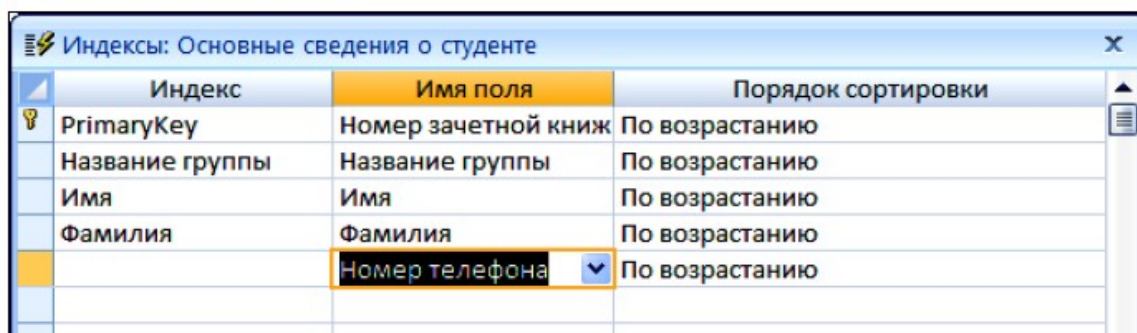


Рис. 2.3. Диалоговое окно **Индексы**. Составной индекс

**Примечание.** По умолчанию устанавливается порядок сортировки **По возрастанию (Ascending)**.

Диалоговое окно **Индексы** используется также для просмотра, изменения и удаления существующих индексов. Изменить можно:

1. Название индекса в поле **Индекс (Index Name)**.
2. Поле таблицы, соответствующее данному индексу, выбрав новое поле из списка в поле **Имя поля (Field Name)**.
3. Порядок сортировки в поле **Порядок сортировки (Sort Order)**.
4. Свойства данного индекса в нижней части окна (см. рис. 2.3):
  - **Ключевое поле (Primary)** определяет, является ли индексированное поле ключевым;
  - **Уникальный индекс (Unique)** определяет, должно ли быть каждое значение в этом поле уникальным.

**Пропуск пустых полей (Ignore Nulls)** определяет, включаются или не включаются в индекс записи с пустым (**Null**) значением данного поля.

## II. Создание связей между таблицами в БД

Существует несколько типов отношений (связей) между таблицами:

- 1) при отношении «**один-к-одному**» каждой записи ключевого поля в первой таблице соответствует только одна запись в связанном поле другой таблицы, и наоборот. Отношения такого типа используются не очень часто. Иногда их можно использовать для разделения таблиц, содержащих много полей, для отделения части таблицы по соображениям безопасности;
- 2) при отношении «**один-ко-многим**» каждой записи в первой таблице соответствует несколько записей во второй, но запись во второй таблице не может иметь более одной связанной записи в первой таблице;
- 3) при отношении «**многие-ко-многим**» одной записи в первой таблице могут соответствовать несколько записей во второй таблице, а одной записи во второй таблице могут соответствовать несколько записей в первой.

При установлении связей между таблицами предварительно закройте все открытые таблицы, так как создавать или изменять связи между открытыми таблицами нельзя.

### Этапы работы

1. Проектирование структуры базы данных, состоящей из двух таблиц.
2. Конструирование пустых таблиц базы данных. БД содержит две таблицы: **Список** из 7 полей (код, фамилия, имя отчество, год рождения, школа, класс, учебная группа). Номера групп и фамилии преподавателей будут храниться в отдельной таблице **Группы** в виде двух столбцов (поля: учебная группа и преподаватель).
3. Создание схемы базы данных.  
В данном случае таблицы связаны связью «один-ко-многим». Это значит, что в таблице **Группы** каждое значение может встречаться только один раз, а в таблице **Список** – сколько угодно (несколько человек из одной группы). **Связи следует устанавливать при пустых таблицах. Если таблицы заполнены, могут возникнуть проблемы при создании связей и свойств связей. Для связи в обеих таблицах должны быть ключевые поля.** В таблице **Список** – поле **Код**, а в таблице **Группы** – поле **Учебная группа**.
4. Ввод данных в таблицы.  
Создадим **Форму** для ввода данных и воспользуемся ею. **При наличии связанных таблиц имеет значение порядок заполнения их значениями.**

### Ход работы:

#### Задание 1.

- **Создайте две пустые таблицы: Список и Группы**, в соответствии с заданными полями, одним из способов: в режиме Таблицы, с помощью Конструктора.
- **Задайте ключевые поля в таблицах:** в **Списке** – это поле **Код**, в таблице **Группы** – это поле **Учебная группа**. Для этого перейдите в режим **Конструктор** и посмотрите, правильно ли заданы поля. Сделайте поле **Код** в таблице **Список** ключевым, поместив курсор на имя поля и нажав кнопку с изображением ключа (Ключевое поле). Тип поля: **Код** – счетчик, **Фамилия, Имя, Отчество** – текстовые, остальные поля – числовые.

- **Создайте подстановку в таблице Список:**

Особое поле в таблице – поле **Учебная группа**. Его тип числовой. Общие свойства поля не меняем. Значения этого поля надо набивать не вручную, а выбирать из списка, содержащегося в таблице **Группы**. Для этого в свойствах поля следует указать, что здесь имеет место подстановка по следующей схеме: выберите вкладку **Подстановка**, тип элемента управления – **Поле со списком**, источник строк – **Группы**. Получите значения элементов, как показано в таблице

**Подстановка**

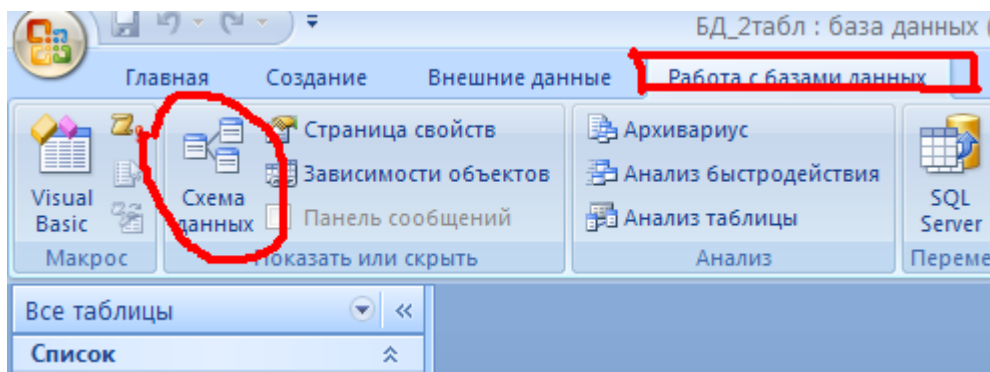
Тип элемента управления	Поле со списком
Тип источника строк	Таблица/запрос
Источник строк	Группы

Остальные свойства оставить как есть в таблице.

- Сохраните изменения, щелкнув по кнопке с дискеткой.
- В результате вы должны получить две несвязанные таблицы.

**Задание 2. Создайте схему данных.**

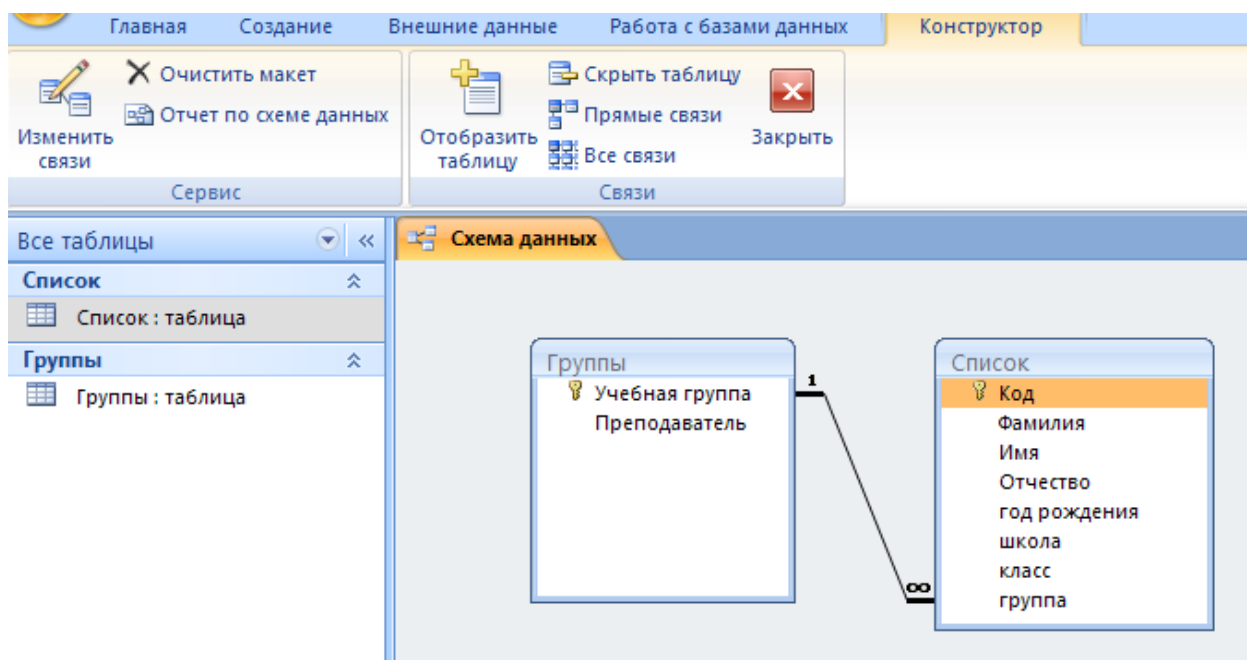
- Выберите команду меню **Работа с базами данных** и найдите в ленте команд создание **Схемы данных**:



• В

появившемся окне **Добавление таблицы** выделите таблицу **Группы** и щелкните по кнопке **Добавить**. Выделите таблицу **Список** и щелкните по кнопке **Добавить**. В окне **Схема данных** появится условный вид этих таблиц. Щелкните по кнопке **Заккрыть** окна **Добавление таблицы**.

- Увеличьте окно таблицы **Список** так, чтобы были видны все поля.
- Поставьте мышку на имя поля **Учебные группы** в таблице **Группы** и, не отпуская кнопку мышки, перетащите ее на поле **Учебные группы** в таблице **Список**. Отпустите мышку. Появится диалоговое окно **Связи**.
- Включите значок **Обеспечение целостности данных**. Это невозможно будет сделать, если типы полей заданы не одинаково.
- Включите значок **Каскадное обновление связанных полей**. Это приведет к тому, что при изменении номера группы в таблице **Группы** автоматически изменится соответствующий номер в таблице **Список**.
- Включите значок **Каскадное удаление связанных полей**. Это приведет к тому, что при удалении записи с номером группы в таблице **Группы** будут удалены все записи из таблицы **Список**, в которой стояли соответствующие номера групп.
- Щелкните по кнопке **Создать**, появится связь «один-ко-многим».
- Закройте схему данных и ответьте утвердительно на вопрос о сохранении схемы данных.



**Задание 3.** Заполните таблицу **Группы**, а затем **Список** значениями из таблицы 1 и 2.

Таблица 1. Группы

Учебная группа	Преподаватель
101	Верзаков С.А.
102	Белоусов А.И.
103	Масалова В.А.
104	Новикова Е.В.
105	Зачесова Т.П.

Таблица 2. Список

Код	Фамилия	Имя	Отчество	Год рождения	Школа	Класс	Учебная группа
1	Чернова	Кристина	Ивановна	1984	1	9	101
2	Терещенко	Инна	Алексеевна	1983	3	10	103
3	Истратов	Максим	Владимирович	1984	5	9	101
4	Бондарь	Ольга	Петровна	1983	1	10	104
5	Новоселов	Алексей	Иванович	1984	3	9	105
6	Петрунина	Оксана	Леонидовна	1984	3	9	102
7	Иванова	Елена	Сергеевна	1983	3	10	101
8	Воробьев	Александр	Владимирович	1984	5	9	103
9	Никифоров	Олег	Федорович	1984	1	9	105
10	Шульга	Константин	Аркадьевич	1984	1	9	103


**Задание 4.** Создайте форму для ввода данных.


- Выберите и откройте таблицу **Список**.
- Выберите меню **Создать**.
- В ленте команд в разделе **Формы** выберите значок **Форма**.
- Щелкните по кнопке **ОК**. Появится пустая форма ввода данных в таблицу **Список**.
- Сохраните форму под именем **Список**.
- Добавьте в базу данных записи, используя форму. Таблица 3.



Таблица 3.

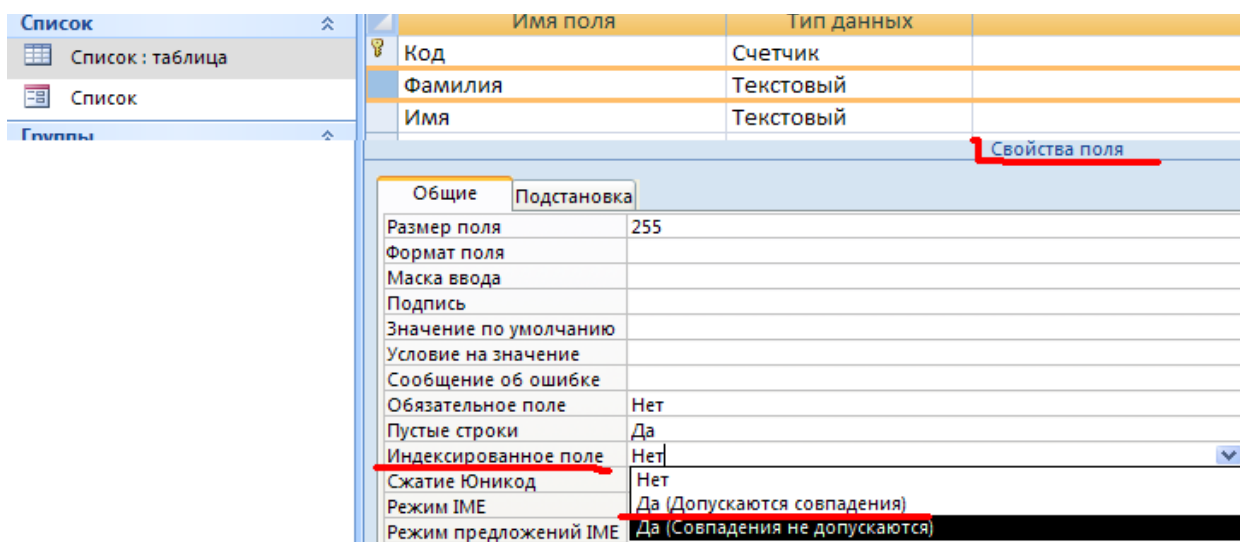
Код	Фамилия	Имя	Отчество	Год рождения	Школа	Класс	Учебная группа
11	Зацепин	Антон	Маркович	1983	5	10	104
12	Черных	Юлия	Максимовна	1983	3	10	101
13	Давыдова	Мария	Михайловна	1983	1	10	102
14	Онучина	Евгения	Ивановна	1984	1	9	102
15	Смирнов	Алексей	Николаевич	1983	5	10	104
16	Воропаев	Анатолий	Николаевич	1984	5	9	105
17	Кислицына	Анна	Михайловна	1984	5	9	103
18	Новиков	Михаил	Сергеевич	1983	3	10	105
19	Сухих	Борис	Евгеньевич	1983	3	10	101
20	Хабибуллин	Руслан	Рахимович	1984	1	9	101

**Замечание.** Переход от поля ввода к следующему полю производится клавишей [Tab], [Enter] или мышкой. Для создания новой записи следует щелкнуть по кнопке . Значения поля **Код** будут заполняться автоматически. Значения поля **учебная группа** выбирайте из ниспадающего списка.

- Сохраните введенные данные. Закройте форму.
- Перейдите в окно **Таблицы**. Откройте таблицу **Список** и убедитесь, что в таблицу внесены новые записи.
- Сохраните текущую таблицу, щелкнув мышкой по кнопке с дискеткой . .

**Задание 5.** Создайте составной индекс в таблице **Список** по двум полям.

- Откройте таблицу **Список** в режиме Конструктора.
- Выделите поле **Фамилия**.
- В разделе **Свойства поля**, расположенного внизу таблицы, установите в **Индексированном поле** значение **Да** и значения **Да (Совпадения допускаются)**.



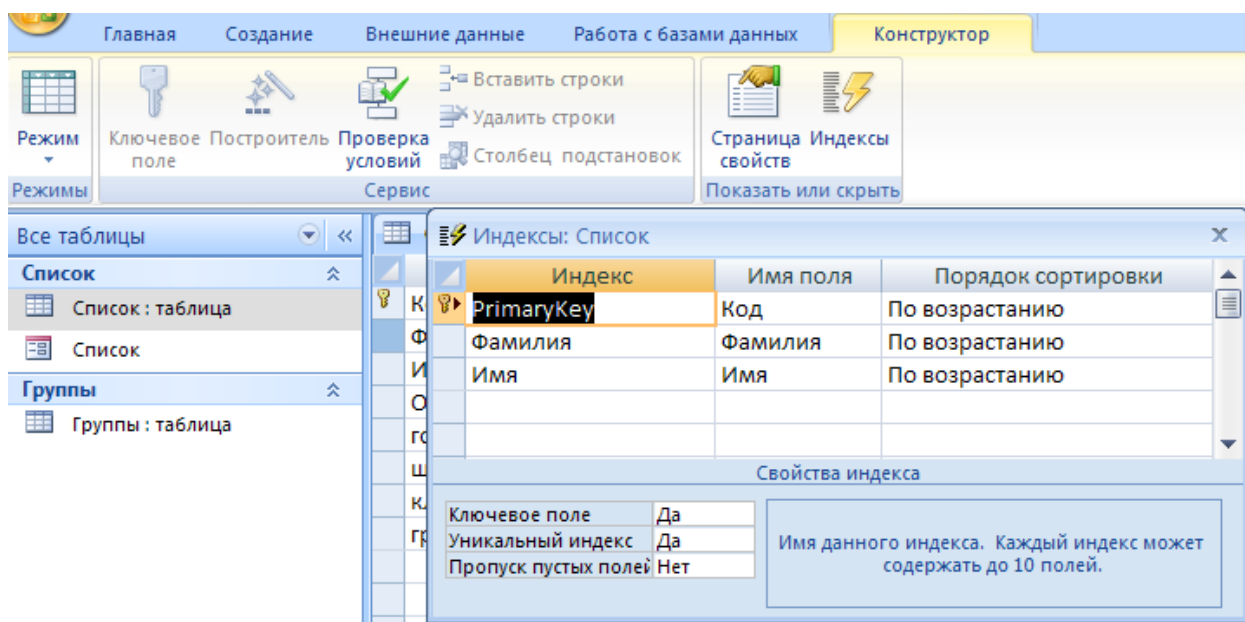
Свойства поля

Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый

Общие Подстановка

Размер поля	255
Формат поля	
Маска ввода	
Подпись	
Значение по умолчанию	
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Да
Сжатие Юникод	Нет
Режим IME	Да (Допускаются совпадения)
Режим предложений IME	Да (Совпадения не допускаются)

- Те же действия выполните для поля **Имя**.
- Выполните команду **Индексы**. Откроется диалоговое окно.



- Убедитесь, что во 2 столбце **Имя поля** заданы поля **Фамилия** и в следующей записи – поле **Имя**. Порядок сортировки выберите по возрастанию.
- Закройте окно индексов и сохраните изменения в таблице **Список**.

#### Задание 6. Проверьте каскадное обновление связанных полей.

- Откройте таблицу **Группы**.
- Исправьте учебные группы на **201, 202, 203, 204, 205**.
- Сохраните и закройте таблицу **Группы**.
- Откройте таблицу **Список**. Убедитесь, что значения групп изменились.
- Закройте таблицу **Список**.

#### Задание 7. Проверьте каскадное удаление связанных полей.

- Откройте таблицу **Группы**.
- Удалите первую запись (всю строку выделите и нажмите на клавишу [Delete]).
- Согласитесь с проверочным вопросом. Закройте таблицу **Группы**.
- Откройте таблицу **Список**. Убедитесь, что исчезли записи с номером группы 201. Закройте таблицу **Список**.

#### **Предъявите преподавателю:**

1. таблицу **Список**;
2. таблицу **Группы**;
3. форму **Список**.

**Завершите работу программы Access.** Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте на него утвердительно.

#### **Контрольные вопросы к лабораторной работе**

1. Что такое индекс?
2. Какие индексы бывают и чем отличаются?
3. Для чего используется поле подстановки?
4. Какие типы отношений между таблицами могут быть?
5. Что значит отношение «один-к-одному»?

6. Что значит связь между таблицами «многие-ко-многим»?
7. Что означает отношение «один-ко-многим»?
8. Как установить связи между таблицами в базе данных в Access?
9. Что значит **«каскадное обновление связанных полей»**?
10. Что значит **«каскадное удаление связанных полей»**?