

Тема: Реляционные БД. Создание реляций между таблицами.

Содержание

Реляционные БД.....	2
Создание реляций между таблицами.....	2
Создание поля с подстановкой значений из другой таблицы	2
Создание связей между таблицами	4
Главный ключ таблицы.....	4
Отношения между таблицами	5
Схема данных.....	7
Методические рекомендации по выполнению практической работы.....	9
Используемая литература:	Ошибка! Закладка не определена.

Реляционные БД

Можно доказать, что любая структура данных поддается преобразованию в простую двумерную таблицу. Мы уже говорили, что такое представление является наиболее удобным и для пользователя, и для машины, — подавляющее большинство современных информационных систем работает именно с такими таблицами. Однако большинство практических задач требует наличия в базе данных нескольких связанных между собой таблиц для управления данными, содержащимися в связанных таблицах. Для них используются те же средства, что и при работе с однотоабличными базами данных. Кроме того, разработаны специальные средства предназначенные для связывания таблиц.

Здесь мы вводим понятие *реляционной БД*, т. е. базы данных, которая является совокупностью описанных выше двумерных таблиц (синоним неуклюжего словосочетания «простые двумерные таблицы»).

Основная идея реляционного подхода состоит в том, чтобы представить произвольную структуру данных в виде простых двумерных таблиц.

1. В реляционных БД любые совокупности данных представляются в виде двумерных *таблиц*.
2. Каждая таблица должна иметь уникальное *имя* в заданном формате.
3. Каждая таблица состоит из фиксированного числа *столбцов (полей)* и некоторого (переменного) количества *строк (записей)*.

Каждый столбец представляет конкретное данное (например, код фирмы, код продукции, номер телефона, факультет). На языке БД столбцы таблицы называются *полями*, причем для каждого поля разработчик должен определить:

- уникальное имя поля;
- тип поля;
- длину (свойства) поля.

Каждая строка таблицы на языке БД называется *записью*. Записи нумеруются по порядку 1, 2, ...n, где n – общее число записей (строк) в таблице на данный момент. В отличие от количества полей (столбцов) в таблице, количество записей в процессе эксплуатации БД может как угодно меняться (от 0 до десятков и сотен тысяч). Количество полей тоже можно изменить, но это уже особая операция, которая называется *реорганизацией* файла.

4. Сила реляционных баз данных заключается в том, что они могут быстро найти и связать данные из разных таблиц при помощи запросов, форм и отчетов. Для этого каждая таблица должна содержать одно или несколько полей, однозначно идентифицирующих каждую запись в таблице. Это поле называется *ключевым полем* таблицы.

Создание реляций между таблицами

Создание поля с подстановкой значений из другой таблицы

1. Откройте таблицу в режиме **Конструктор**.
2. При необходимости вставить в таблицу поле.
Чтобы вставить в таблицу поле, выберите строку, над которой его нужно поместить, и нажмите кнопку **Добавить строки** на панели инструментов. Чтобы добавить поле в конец таблицы, выберите первую пустую строку. В столбце **Имя поля** введите имя поля. В столбце **Тип данных** выберите тип вводимых данных.
3. Если поле уже существует, то в **Типе данных** выберите строку **Мастер подстановок**.

Примечание Если поле, которое будет использоваться как внешний ключ для поля подстановок, уже существует, выберите строку этого поля. Например, если имеется таблица «Товары», содержащая поле «КодПоставщика», которое требуется превратить в поле подстановок, выбирающее названия поставщиков из таблицы «Поставщики», то необходимо выбрать строку поля «КодПоставщика».

4. Выберите переключатель для создания столбца подстановки, использующего значения из таблицы или запроса.
5. С помощью **Мастера подстановок** пошагово, создать список для выбора данных из другой таблицы.

1 Мастер создает столбец подстановки, в котором отображается список значений для выбора. Каким способом столбец подстановки будет получать эти значения?

Объект "столбец подстановки" будет использовать значения из таблицы или запроса.

Будет введен фиксированный набор значений.

2 Выберите таблицу или запрос со значениями, которые будут содержать столбец подстановки.

Таблица: Анкеты
Таблица: Предметы
Таблица: пропуска
Таблица: Удостоверения

Показать Таблицы Запросы Таблицы и запросы

3 Какие поля содержат значения, которые следует включить в столбец подстановки? Отобранные поля станут столбцами в объекте "столбец подстановки".

Доступные поля: Дата поступления, Телефон, Код_курсанта

Выбранные поля: Код_пропуска, **Номер**

4 Допускается сортировка записей по возрастанию или по убыванию, включающая до 4 полей.

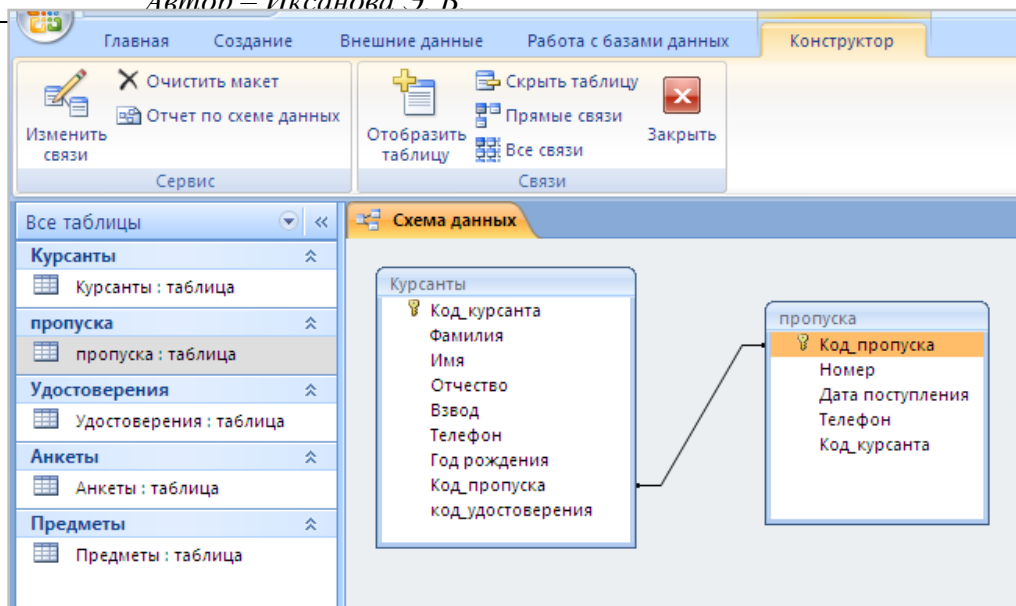
1 (Отсутствует) по возрастанию
2 Код_пропуска по возрастанию
3 Номер по возрастанию
4 по возрастанию

5 Перетащите правую границу заголовка столбца на нужную ширину или дважды щелкните ее для автоматического подбора ширины.

Скрыть ключевой столбец (рекомендуется)

Номер
1/2A
2/2A
3/2A

После нажатия кнопки **Готово** Microsoft Access создаст поле подстановок и установит некоторые свойства этого поля на основании значений, выбранных в мастере. Кроме этого автоматически будет определена связь между таблицами.



Создание связей между таблицами

Главный ключ таблицы

Каждая запись в таблице должна иметь **главный ключ**, т. е. идентификатор, значение которого однозначно определяет эту и только эту запись. Если для таблицы обозначены ключевые поля, то Microsoft Access предотвращает дублирование или ввод пустых значений в ключевое поле. *Ключ может состоять из одного или нескольких полей.* Главный ключ должен обладать двумя свойствами:

1. *Однозначная идентификация записи:* запись должна однозначно определяться значением ключа.
2. *Отсутствие избыточности:* никакое поле нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации.

Каждое значение главного ключа в пределах таблицы должно быть уникальным. Таким образом, указание главного ключа — это и есть единственный способ отличить один экземпляр объекта от другого.

Например,

Составляя таблицу «Кадры», мы можем допустить небрежность, неявно предположив, что в таблице не будет лиц с одинаковыми фамилиями и инициалами. Ясно, что это не так: в большой организации всегда могут найтись два-три Кузнецовых Б.Н. и т. п. и наша небрежность сведет к нулю ценность такой информационной системы. На этом примере видно, что фамилия никогда не может быть ключом такой таблицы — вместо нее всегда используют придуманные разработчиком уникальные цифровые обозначения лица - табельные номера или что-то другое.

Почему мы говорим «Главный ключ таблицы»? А что, бывают «неглавные» ключи? Да, кроме главного, мы можем использовать так называемые **простые** (или **вторичные**) ключи таблицы.

Например, в таблице *телефоны* главный ключ — номер телефона, однако, мы можем просматривать эту таблицу по категории абонента, и тогда мы говорим, что поле категории — простой ключ таблицы. Значение простого ключа может быть **неуникальным**. Главный ключ может быть только один, а простых ключей множество.

В заключение сделаем два важных замечания:

1. Теория требует, чтобы значение главного ключа было уникальным. Однако в реальных БД сплошь и рядом в файлах встречаются дубликаты главных ключей, и системы спокойно работают с ними. Пользователю предоставляется возможность выявить дубликаты и самому решить — исправить ли эти ошибки или мириться с ними.
2. Теория требует наличия главного ключа у всех файлов (в частности, только через аппарат главных ключей можно автоматически устанавливать связи между файлами). Однако в вашей БД могут быть и файлы, для которых понятие главного ключа не существует. Система спокойно работает и с такими файлами.

В Microsoft Access можно выделить три типа ключевых полей:

счетчик,
простой ключ,
составной ключ.

Счетчик

Поле счетчика можно задать таким образом, чтобы при добавлении каждой записи в таблицу в это поле автоматически вносилось порядковое число. Указание такого поля в качестве ключевого является наиболее простым способом создания ключевых полей. Если до сохранения созданной таблицы в Microsoft Access ключевые поля не были определены, то при сохранении будет выдано сообщение о создании ключевого поля. При нажатии кнопки **Да** будет создано ключевое поле счетчика.

Простой ключ

Если поле содержит уникальные значения, такие как коды или инвентарные номера, то это поле можно определить как ключевое. Если выбранное поле содержит повторяющиеся или пустые значения, то оно не будет определено как ключевое.

Составной ключ

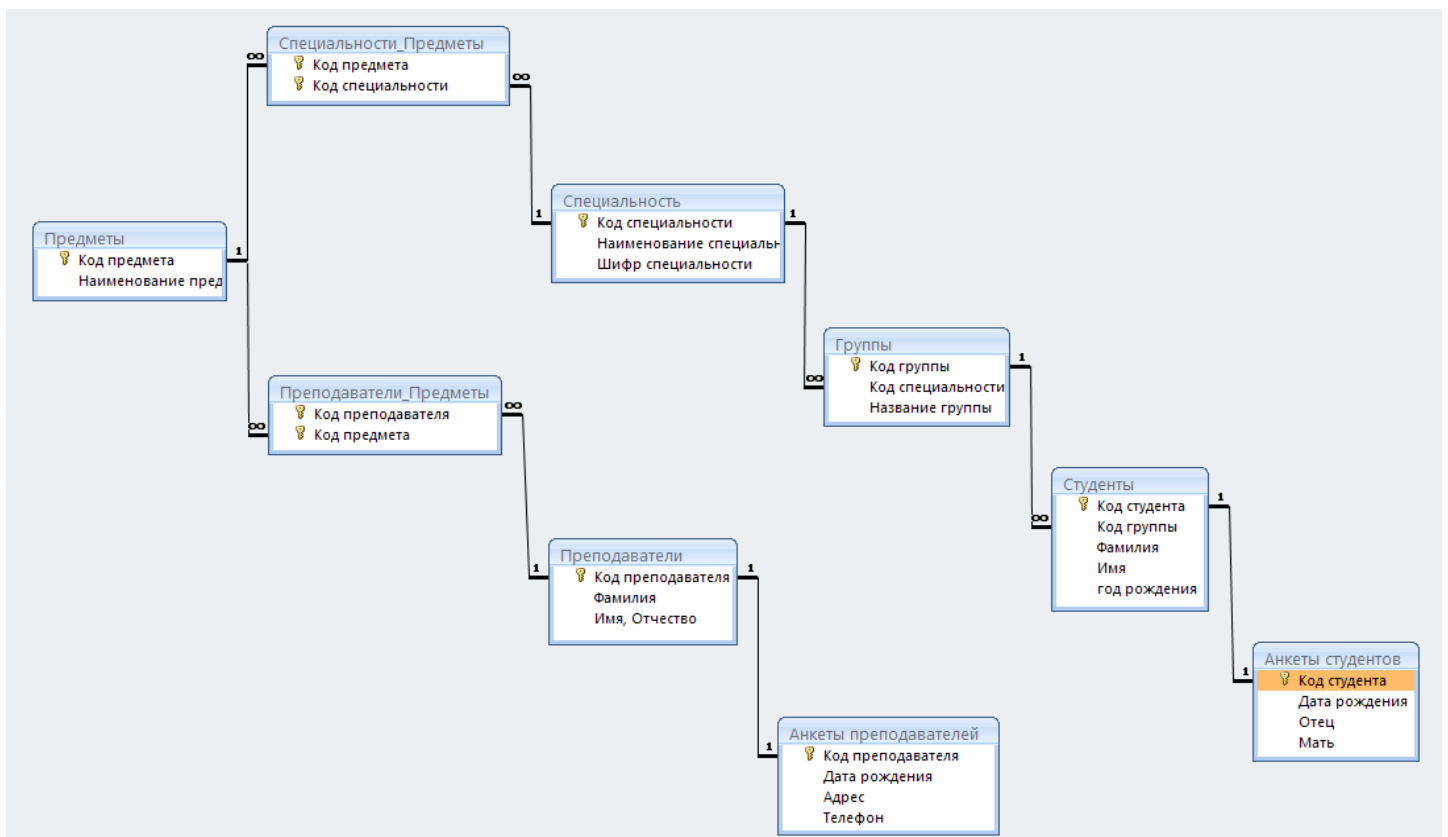
В случаях, когда невозможно гарантировать уникальность значений одного поля, существует возможность создать ключ, состоящий из нескольких полей.

Примером может служить совокупность трех полей: *Фамилия + Имя + Отчество*.

Отношения между таблицами

После создания различных таблиц, содержащих данные, относящиеся к различным аспектам базы данных, разработчик должен продумать, каким образом Microsoft Access будет объединять эти данные при их извлечении из базы данных. Первым шагом при этом является определение связей между таблицами. После этого становится возможным создание запросов, форм и отчетов, в которых выводятся данные из нескольких таблиц сразу.

Пример проекта БД «Колледж».



Как работает связь между таблицами

В приведенном примере данные полей в таблицах должны быть определенным образом скоординированы, так как они содержат сведения об одних и тех же объектах (студентах, преподавателях). Эта координация осуществляется путем установления связей между таблицами. Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях, обычно между полями разных таблиц, имеющими одинаковые имена. В большинстве случаев с ключевым полем одной таблицы, являющимся уникальным идентификатором каждой записи, связывается внешний ключ другой таблицы.

Например,

Две таблицы *Группа* и *Студенты*, связь определяется по полям *Код группы* в обеих таблицах.

Отношение «один-к-одному»

Отношение «один-к-одному» означает, что каждая запись в одной таблице соответствует только одной записи в другой таблице.

Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением «один-к-одному» используют для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

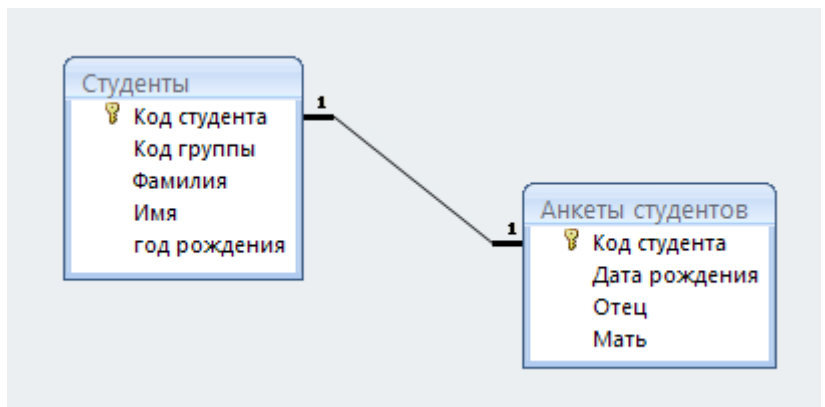
Например,

Две таблицы *Студенты* и *Анкеты студентов*. Между таблицами существуют отношение один-к-одному, поскольку для одного человека может существовать только одна запись, содержащая анкетные данные.

Построение связи между таблицами

Тип и направление создаваемой связи зависит от полей, для которых определяется связь.

- Отношение «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми (или имеют уникальные индексы).



Отношение «один-ко-многим»

Связь с отношением «один-ко-многим» является наиболее часто используемым типом связи между таблицами.

При отношении «один-ко-многим» каждой записи в одной таблице могут соответствовать несколько записей в другой таблице.

Например,

Две таблицы *Группа* и *Студенты*. Между таблицами существуют отношение «один-ко-многим», поскольку в одной группе может учиться множество студентов, но в то же время один студент не может учиться в разных группах.

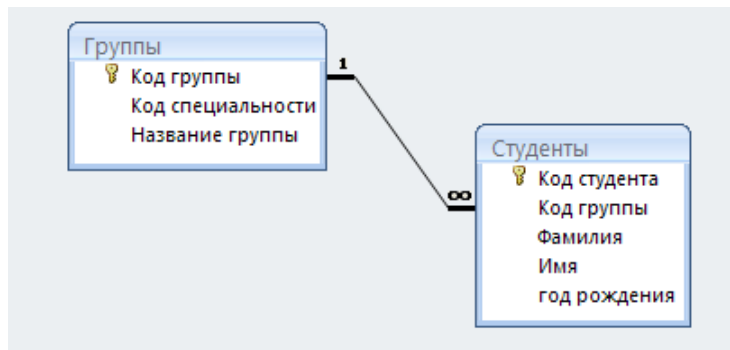
Построение связей между таблицами

При установке отношения «один-ко-многим» необходимо:

- 1). Установить, какая из таблиц является основной (*Группы*), а какая подчиненная (*Студенты*),
- 2). В подчиненной таблице должно быть поле для связи (*Код группы*). Оно должно соответствовать по смыслу, по названию и по типу данных ключевому полю основной таблицы. Но если это поле – счетчик, то в подчиненной таблице оно должно быть числовым.

3). Связь создается между ключевым полем основной таблицы и соответствующим полем подчиненной таблицы:

Код группы (таблица *Группы*) →
Код группы (таблица *Студенты*)



Отношение «много-ко-многим»

Отношение «много-ко-многим» возникает между двумя таблицами в тех случаях, когда:

- одна запись из первой таблицы может быть связана более чем с одной записью из второй таблицы;
- одна запись из второй таблицы может быть связана более чем с одной записью из первой таблицы.

Например,

Две таблицы *Предметы* и *Преподаватели*. Между таблицами может быть реализована связь «много-ко-многим», так как один и тот же предмет могут преподавать разные преподаватели, и в то же время один и тот же преподаватель может преподавать разные предметы.

Такая схема реализуется только с помощью третьей (связующей) таблицы, в состав полей которой входят ключевые поля связуемых таблиц (код предмета и код преподавателя). В то же время эти два поля будут являться составным ключом связующей таблицы.

Построение связей между таблицами

Связь с отношением «многие-ко-многим» фактически является двумя связями с отношением «один-ко-многим» через третью таблицу, ключ которой состоит из по крайней мере двух полей, которые являются полями внешнего ключа в двух других таблицах.

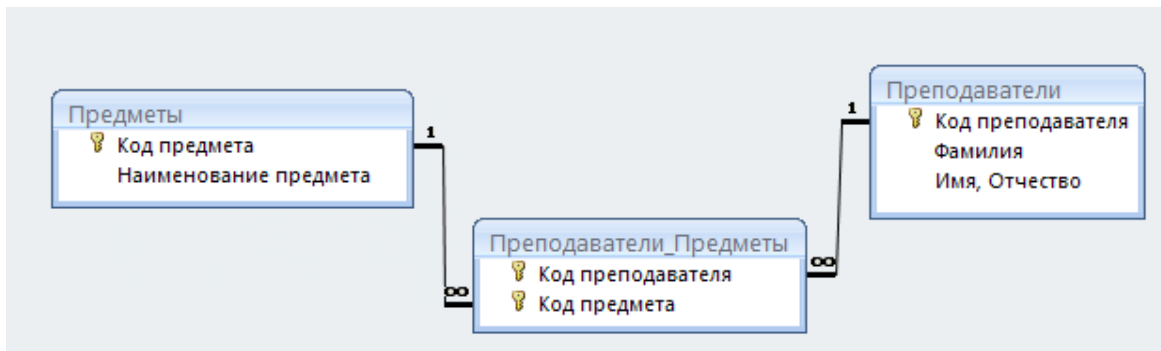
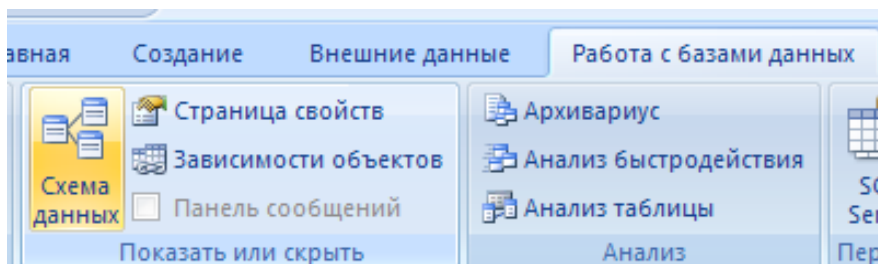


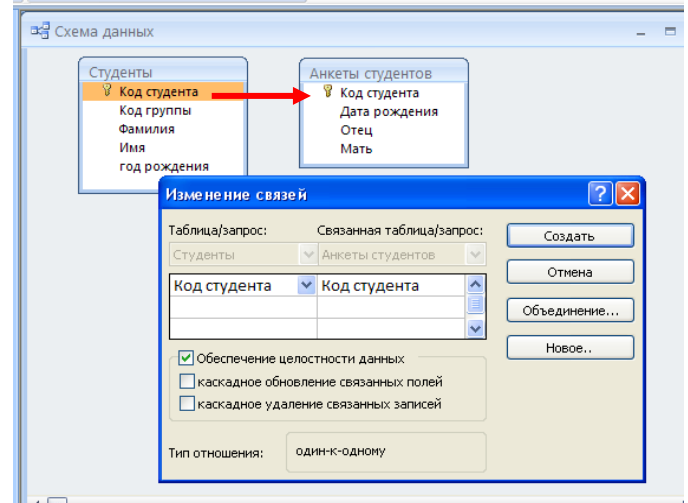
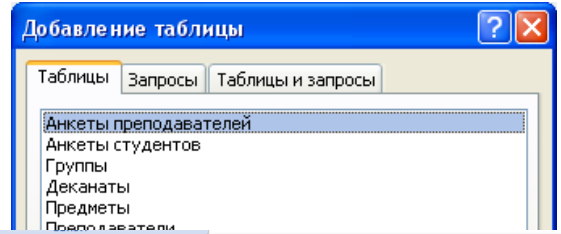
Схема данных

Для того чтобы определить и отредактировать связи между таблицами в Microsoft Access, следует:

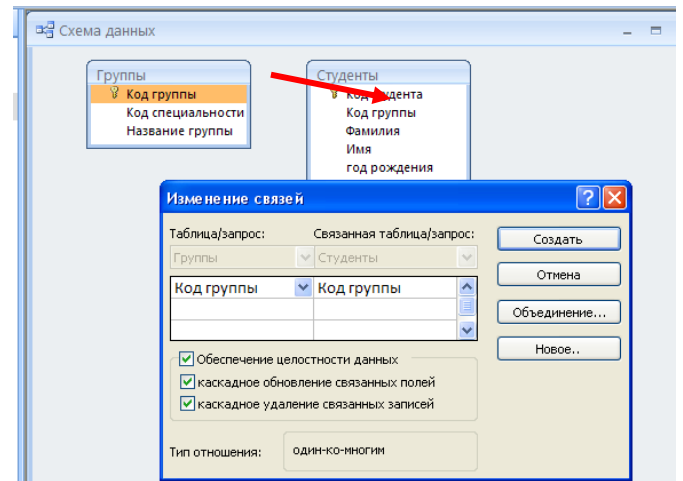
1. Закрывать все открытые таблицы. Создавать или изменять связи между открытыми таблицами нельзя.
2. На вкладке **Работа с базами данных** выбрать инструмент **Схема данных**:



3. На вкладке **Конструктор** с помощью инструмента **Отобразить таблицу** открыть диалоговое окно **Добавление таблицы**. В диалоговом окне выделить таблицы, между которыми нужно создать связи и нажать кнопку **Добавить**.
4. В окне **Схема данных** появятся выделенные таблицы как графические объекты, которые можно передвигать, изменять размеры.
5. Для создания связи «**один-к-одному**» необходимо выделить ключевое поле одной таблицы и совместить с ключевым полем второй таблицы. После отжатия кнопки мыши на экране появится диалоговое окно **Изменение связей**. В окне автоматически определяется **Тип отношения**, связуемые таблицы и поля связи.
Для создания правильной связи необходимо указать параметр:

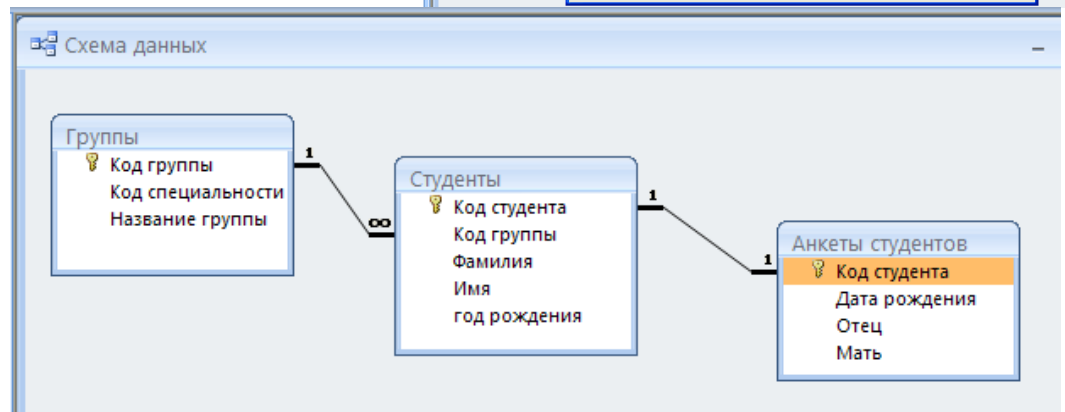


Обеспечение целостности данных.
Для создания связи «**один-ко-многим**» необходимо выделить ключевое поле одной таблицы и совместить с соответствующим полем второй таблицы. После отжатия кнопки мыши на экране появится диалоговое окно **Изменение связей**. В окне автоматически определяется **Тип отношения**, связуемые таблицы и поля связи.



Для создания правильной связи необходимо указать параметры:

- Обеспечение целостности данных,**
- каскадное обновление связанных полей,**
- каскадное удаление связанных записей.**



6. **Изменение связи:**
Открыть режим **Схема данных**, выделить связь и на вкладке **Конструктор** выбрать инструмент **Изменить связь**. Можно аналогичную команду выбрать в контекстном меню.
7. **Удаление связи:**
Открыть режим **Схема данных**, выделить связь и выбрать команду **Удалить** в контекстном меню.
8. **Внести изменения в таблицу:**
Открыть режим **Схема данных**, выделить таблицу выбрать команду **Конструктор** в контекстном меню.

Методические рекомендации по выполнению практической работы

Тема Реляционные БД. Создание реляций между таблицами.

Цель Закрепление на практике знаний по теме.

Практическая часть Выполнить практическую работу.

Критерии оценки:

Критерий	Балл	Степень выполнения работы
работа выполнена на 100%.	5 (отлично)	Созданы таблицы, правильно описаны поля и их свойства. Созданы связи между таблицами.
работа выполнена на 75%	4 (хорошо)	Созданы таблицы, правильно описаны поля и их свойства. Созданы связи между таблицами с ошибками.
работа выполнена на 50%	3 (удовлетворительно)	Созданы таблицы, правильно описаны поля и их свойства. Неправильно созданы связи между таблицами.
работа не выполнена	2 (неудовлетворительно)	Созданы таблицы, не созданы связи между таблицами.

Используемая литература:

1. Трофимов, В. В. Информационные технологии в экономике и управлении: учебник для СПО / под ред. В. В. Трофимова. — 2-е изд., перераб. и доп. — М.: Издательство Юрайт, 2017. — 482 с.— ISBN 978-5-534-09137-3, 978-5-534-09138-0— Режим доступа: <https://biblio-online.ru/book/informacionnye-tehnologii-v-ekonomike-i-upravlenii-v-2-ch-chast-1-427212>
2. Гаврилов, М. В. Информатика и информационные технологии: учебник для СПО / М. В. Гаврилов, В. А. Климов. — 4-е изд., перераб. и доп. — М.: Издательство Юрайт, 2017. — 383 с.— ISBN 978-5-534-03051-8— Режим доступа: <https://biblio-online.ru/book/informatika-i-informacionnye-tehnologii-413451>
3. Трофимов, В. В. Информатика в 2 т. Том 1: учебник для СПО / В. В. Трофимов ; под ред. В. В. Трофимова. — 3-е изд., пер. и доп. — М.: Издательство Юрайт, 2018. — 553 с. — (Серия: Профессиональное образование). — ISBN 978-5-534-02518-7. — Режим доступа: www.biblio-online.ru/book/87EC2130-3EBB-45B7-B195-1A9C561ED9D9.
4. Трофимов, В. В. Информатика в 2 т. Том 2: учебник для СПО / В. В. Трофимов; отв. ред. В. В. Трофимов. — 3-е изд., пер. и доп. — М.: Издательство Юрайт, 2018. — 406 с. — (Серия: Профессиональное образование). — ISBN 978-5-534-02519-4. — Режим доступа: www.biblio-online.ru/book/14FE5928-69CF-41EC-A00B-3979EC8273C8.