

Лекция № 30

**ПРЕДСТАВЛЕНИЕ ОБ
ОРГАНИЗАЦИИ БАЗ ДАННЫХ И
СИСТЕМАХ УПРАВЛЕНИЯ ИМИ
(ПРОДОЛЖЕНИЕ)**

Основные понятия организации данных в реляционной БД:

- таблица,
- запись,
- поле,
- тип поля,
- главный (первичный) ключ таблицы.

Данные в базах хранятся в виде таблиц.

Структуру двумерной таблицы образуют столбцы и строки.

Их аналогами в структуре простейшей базы данных являются поля (столбцы таблицы) и записи (строки таблицы).

Если записей в таблице пока нет, значит, ее структура образована только набором полей.

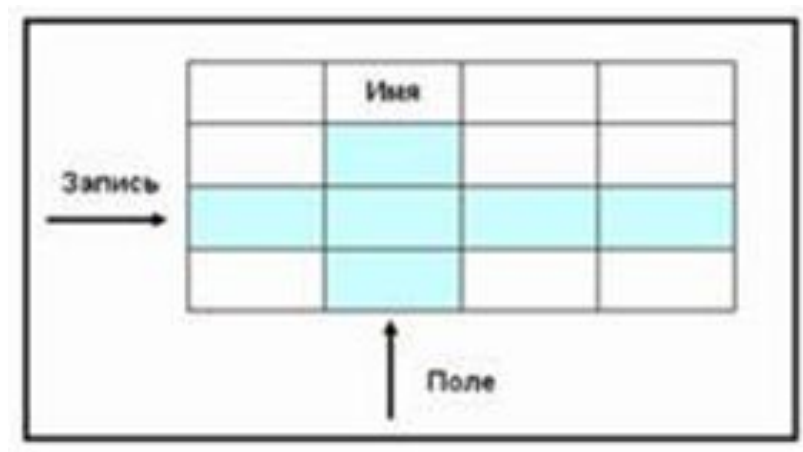
Изменив состав полей базовой таблицы (или их свойства), мы изменяем структуру базы данных и, соответственно, получаем новую базу данных.

Таблица – информационная модель реальной системы.

Каждая таблица имеет *структуру*.

В БД столбцы называют *полями*, а строки – *записями*.

Поля образуют структуру базы данных, а записи составляют информацию, которая в ней содержится.



При создании таблицы одно поле (или одну комбинацию полей) можно назначить ключевым.

С ключевыми полями компьютер работает особо.

Он проверяет их уникальность и быстрее выполняет сортировку по таким полям.

Если вдруг записи в этом поле повторятся, то компьютер просигнализирует об этом.

Для этого и существует понятие ключевое поле.

Например, в библиотечной базе данных таким ключом может быть выбран инвентарный номер книги, который не может совпадать у разных книг.

Уникальное (ключевое) поле – это поле, значения в котором не могут повторяться (например, номер телефона).

Ключевое поле (первичный (главный) ключ) – кандидат для создания связей; это поле, значение которого однозначно определяет запись в таблице.

В качестве ключевого поля чаще всего используют поле, содержащее тип данных *счетчик*.


Однако иногда удобнее в качестве ключевого поля таблицы использовать другие поля: код товара, инвентарный номер, и т.д.

Таким образом, ключевые поля могут быть простыми и составными.

Простой ключ. Если поле содержит уникальные значения, такие как коды или инвентарные номера, то это поле можно определить как первичный ключ. В качестве ключа можно определить любое поле, содержащее данные, если это поле не содержит повторяющиеся значения или значения Null.

Составной ключ. В случаях, когда невозможно гарантировать уникальность значений каждого поля, существует возможность создать ключ, состоящий из нескольких полей. Чаще всего такая ситуация возникает для таблицы, используемой для связывания двух таблиц **МНОГИЕ-КО-МНОГИМ**.


Простой ключ



Номер	Автор	Название	Год	Полка
001	Беляев А.Р.	Звезда КЭЦ	1990	3
002	Олеша Ю.К.	Избранное	1987	5
003	Беляев А.Р.	Избранное	1994	1

В БД «Домашняя библиотека» у разных книг могут совпадать значения полей, но инвентарный номер у каждой книги свой

Составной ключ



Город	№ школы	Директор	Адрес	Телефон
Крюков	1	Иванов А.П.	Пушкина, 5	2-33-56
Шадринск	1	Строев С.С.	Лесная, 14	4-33-11
Шадринск	2	Иванов А.П.	Мира, 34	4-23-24

В БД «Школы области» у разных записей одновременно не могут совпадать только сочетание двух полей: город и номер школы (это составной ключ)

Структура таблицы определяется составом ее *полей* (основные элементы структуры базы данных) и их *свойствами*. Важнейшими свойствами полей являются: *тип поля* и *размер поля*. Для хранения разных типов данных используют *поля соответствующих типов*. Поля базы данных не просто определяют структуру базы – они еще определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей. От свойств полей зависит, какие типы данных можно вносить в поле, а какие нет, а также то, что можно делать с данными, содержащимися в поле.

Основные свойства полей таблиц баз данных (СУБД Microsoft Access):

- **Имя поля** – определяет, как следует обращаться к данным этого поля при автоматических операциях с базой (по умолчанию имена полей используются в качестве заголовков столбцов таблиц).
- **Тип поля** – определяет тип данных, которые могут содержаться в данном поле.
- **Размер поля** – определяет предельную длину (в символах) данных, которые могут размещаться в данном поле.

- **Формат поля** – определяет способ форматирования данных в ячейках, принадлежащих полю.
- **Маска ввода** – определяет форму, в которой вводятся данные в поле (средство автоматизации ввода данных).
- **Подпись** – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется свойство **Имя поля**).

- Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).
- Условие на значение – ограничение, используемое для проверки правильности ввода данных (средство автоматизации ввода, которое используется, как правило, для данных, имеющих числовой тип, денежный тип или тип даты).

- Сообщение об ошибке – текстовое сообщение, которое выдается автоматически при попытке ввода в поле ошибочных данных (проверка ошибочности выполняется автоматически, если задано свойство Условие на значение).
- Обязательное поле – свойство, определяющее обязательность заполнения данного поля при наполнении базы;

- Пустые строки – свойство, разрешающее ввод пустых строковых данных (от свойства Обязательное поле отличается тем, что относится не ко всем типам данных, а лишь к некоторым, например, к текстовым).
- Индексированное поле – если поле обладает этим свойством, все операции, связанные с поиском или сортировкой записей по значению, хранящемуся в данном поле, существенно ускоряются. Кроме того, для индексированных полей можно сделать так, что значения в записях будут проверяться по этому полю на наличие повторов, что позволяет автоматически исключить дублирование данных.

Для каждого поля определяется тип и формат данных. Microsoft Access работают со следующими **типами данных**.

- **Текстовый** – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).
- **Поле Мемо** – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле. Он хранится в другом месте базы данных, а в поле хранится указатель на него, но для пользователя такое разделение заметно не всегда.

- Числовой – тип данных для хранения действительных чисел.
- Дата/время – тип данных для хранения календарных дат и текущего времени.
- Денежный – тип данных для хранения денежных сумм. Теоретически, для их записи можно было бы пользоваться и полями числового типа, но для денежных сумм есть некоторые особенности (например, связанные с правилами округления), которые делают более удобным использование специального типа данных, а не настройку числового типа.

- Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.
- Логический – тип для хранения логических данных (могут принимать только два значения, например, Да или Нет).

- Поле объекта Ole – специальный тип данных, предназначенный для хранения объектов Ole, например мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае полей MEMO, они хранятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной).

- Гиперссылка – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.
- Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод в данных поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

Важнейшими свойствами реальных БД являются возможность получения информации из нескольких таблиц одновременно, а также целостность и непротиворечивость данных.

Таблицы отображают реальные объекты, кроме того, они формируются в результате работы по нормализации исходных таблиц.

После нормализации исходная таблица может быть преобразована в две или более таблицы.

Связи между таблицами позволяют вывести совокупную информацию и автоматически отслеживать целостность данных.

Для связывания таблиц используются ключевые поля – первичные и внешние.

Первичные ключи уникальны, внешние могут повторяться, так как их значения равны значениям первичных ключей в других таблицах.

Рассмотрим в качестве примера простейшую базу данных «Книги», содержащую всего две таблицы.

Издательства : таблица		
	Имя поля	Тип данных
	Код издательства	Счетчик
	Наименование	Текстовый
	Адрес	Поле объекта OLE
	Логотип	Текстовый
	Сайт	Гиперссылка

Книги : таблица		
	Имя поля	Тип данных
	Код книги	Счетчик
	Автор	Текстовый
	Название	Текстовый
	Цена	Денежный
	Код издательства	Числовой
	Дата издания	Дата/время
	Твердый переплет	Логический
	Аннотация	Поле MEMO

При проектировании таблиц задействованы все типы полей, тип поля определяется его назначением.

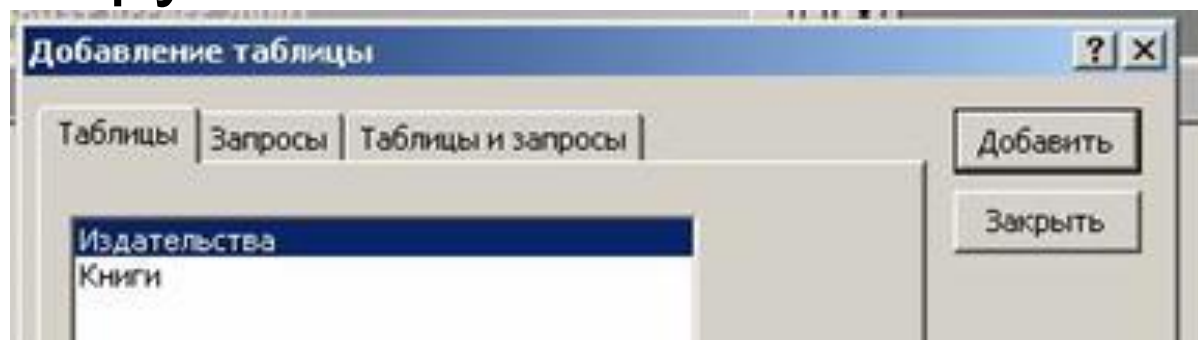
В качестве ключевых полей назначены счетчики, так как счетчики автоматически изменяют свое значение, хотя назначение счетчиков ключевыми полями совсем необязательно.

Кроме того, можно использовать несколько полей в качестве ключа.

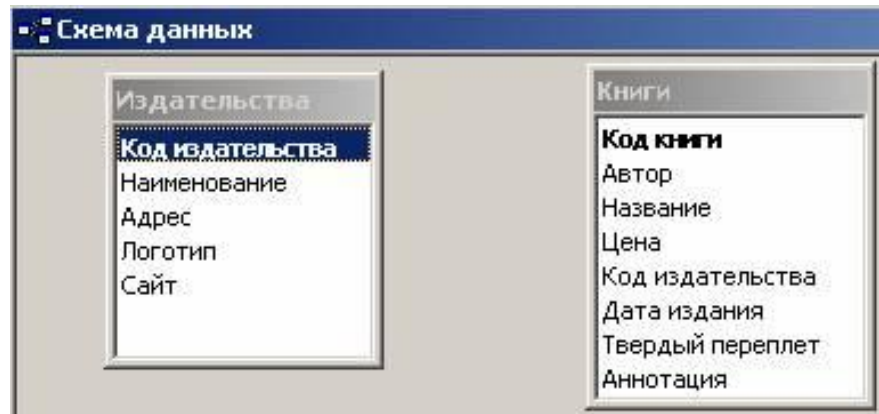
Использование счетчика автоматически обеспечивает уникальность записи и удобство формирования записи.

Код издательства и Код книги являются первичными ключами для соответствующих таблиц. Поле Код издательства присутствует и в таблице Книги в качестве обычного поля. Таким образом, оно может быть использовано в качестве внешнего ключа. Тип этого поля должен совпадать с типом первичного ключа. Счетчик имеет числовой тип (длинное целое). Имена первичного и внешнего ключей не обязательно должны совпадать, но назначение совпадающих имен упрощает дальнейшие действия как СУБД Access , так и разработчика БД.

Для формирования, отображения и редактирования схемы данных используется инструмент , после выбора которого загрузится окно:



Исходная схема данных для двух таблиц, которые теперь надо связать между собой:



Виды логической связи между таблицами

Реляционные таблицы могут быть связаны друг с другом, следовательно, данные могут извлекаться одновременно из нескольких таблиц.

Таблицы связываются между собой для того, чтобы в конечном счете уменьшить объем БД.

Связь каждой пары таблиц обеспечивается при наличии в них одинаковых столбцов.

Существуют **4 типа связей между таблицами:**

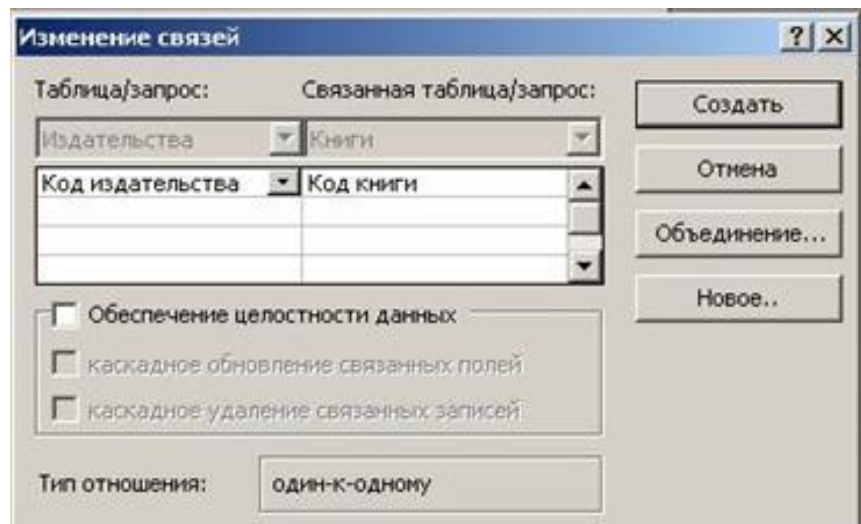
- один к одному,
- один ко многим,
- многие к одному,
- многие ко многим.

Второй и третий типы связей различаются только тем, как по отношению друг к другу располагаются таблицы.

Связь один к одному

Для рассматриваемого случая связь один к одному можно создать, связав первичные ключи таблиц – Код издательства и Код книги. Такая связь будет означать, что каждому издательству ставится в соответствие только одна книга, хотя это и маловероятно. Чтобы связать таблицы, необходимо щелкнуть на ключевом поле одной таблицы и протащить мышью до ключевого поля другой таблицы.

В результате сформируется окно, в котором описывается связь:



Связь один ко многим

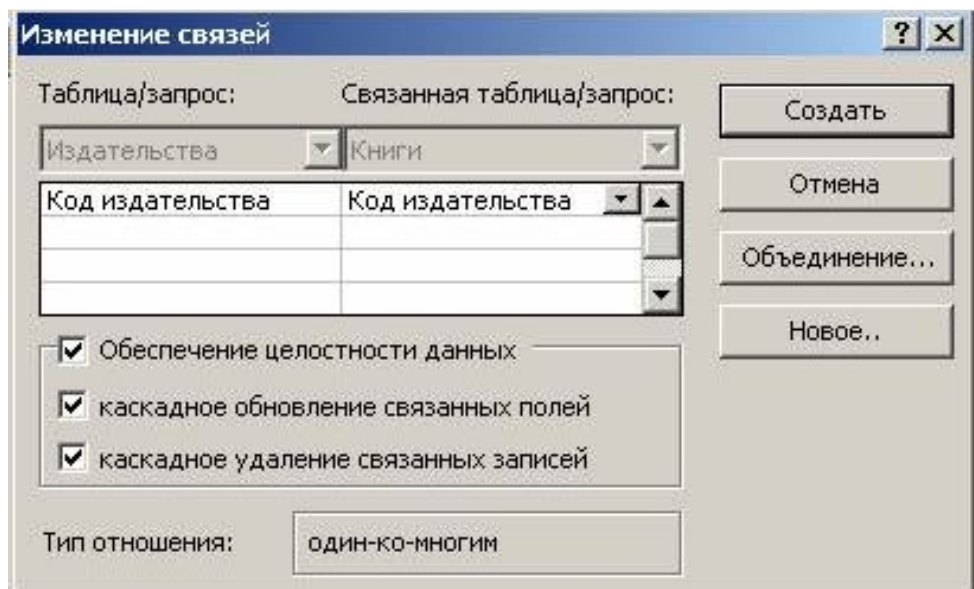
Связь один к одному в рассмотренном примере нелогична. Связь такого типа оправдана, например, для связывания таблиц с личными данными студента и его успеваемостью.

В рассматриваемом случае более логично сформировать связь один ко многим: ведь каждое издательство может издавать множество книг.

Формирование и описание такой связи реализуется по аналогии с предыдущим примером.

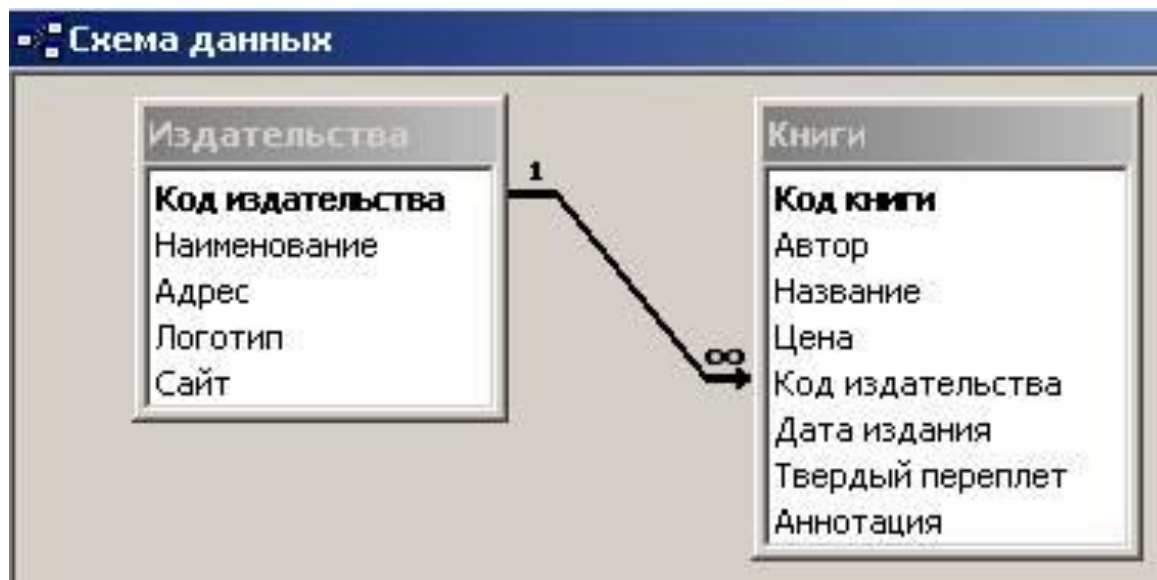
Только теперь связываются первичный ключ «Код издательства» таблицы «Издательства» и внешний ключ «Код издательства» таблицы «Книги».

В этом случае автоматически формируется связь один ко многим.



Внешний вид связи, как и в предыдущем случае, зависит от флажка «Обеспечение целостности данных» и параметров объединения.

Единственное отличие заключается в появлении условного знака бесконечности на конце связи со стороны «многие».



Связь многие ко многим

Здесь более логична связь многие ко многим. Действительно, различные издательства могут издавать одни и те же книги. И наоборот – одна и та же книга может издаваться в различных издательствах.

Для формирования связей такого типа вводится третья таблица, в которой задействованы первичные ключи первых двух таблиц. Если есть необходимость, в эту таблицу вводятся и другие поля.

Например:

Связи : таблица

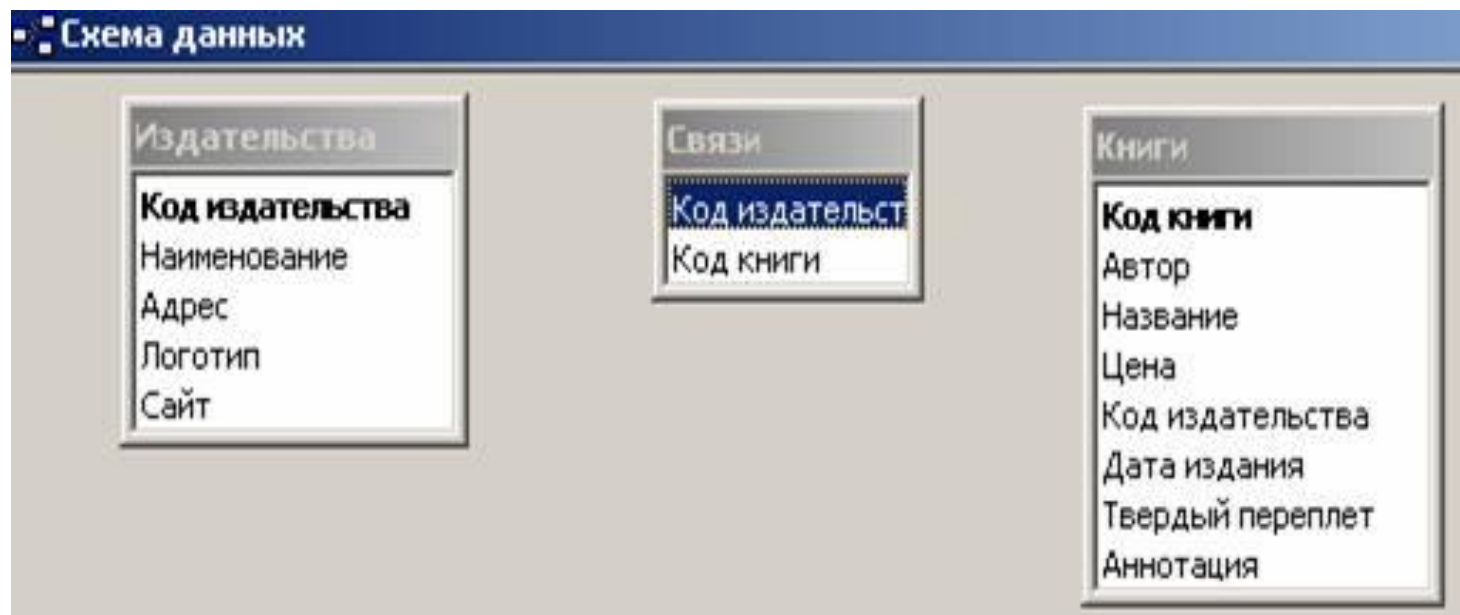
Имя поля	Тип данных
Код издательства	Числовой
Код книги	Числовой

Свойства поля

Общие	Подстановка
Размер поля	Длинное целое
Формат поля	
Число десятичных знаков	Авто
Маска ввода	
Подпись	
Значение по умолчанию	0
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Индексированное поле	Да (Допускаются совпадения)

Поля лучше назвать так же, как и в связываемых таблицах. Типы данных в нашем случае должны быть числовыми, их размер – длинное целое. В свойствах полей «Индексированное поле» необходимо выбрать «Да (Допускаются совпадения).

После описания таблицы «Связи» ее следует включить в схему данных. Для этого необходимо загрузить схему данных, выбрать из контекстного меню команду «Добавить таблицу» и добавить таблицу «Связи» в схему данных. В результате появится окно:



Теперь осталось сформировать связь один ко многим между таблицами «Издательства» и «Связи» и такую же связь – между таблицами «Книги» и «Связи».

В результате получится схема данных:



Между таблицами «Издательства» и «Книги» сформирована связь многие ко многим.

Вопросы

1. Что такое ключевое поле?
2. Что такое простой ключ?
3. Что такое составной ключ?
4. Перечислите основные свойства полей таблиц БД, поясните несколько из них.
5. Какие существуют типы данных полей?
6. Для чего устанавливаются связи между таблицами БД?
7. Какие виды связей между таблицами существуют?
8. В чем заключается связь один к одному и как она устанавливается?
9. В чем заключается связь один ко многим и как она устанавливается?
10. В чем заключается связь многие ко многим и как она устанавливается?