


# База данных

---

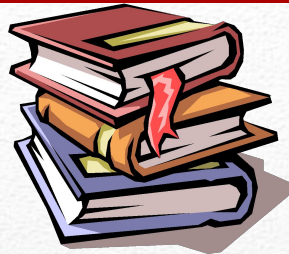


**База данных ( БД )**– это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

---

# Примеры Баз данных

- БД книжного фонда библиотеки
- БД кадрового состава учреждения
- БД законодательных актов в области уголовного права
- Информационные системы по продаже и резервированию авиа- и железнодорожных билетов
- Электронные энциклопедии со сведениями, **например: о муз. инструментах, шедеврах Эрмитажа, кулинарных рецептах, химических элементах и соединениях**



- **Табличные** БД содержат перечень объектов одного типа, т.е. объектов, имеющих одинаковый набор свойств.
- **Иерархические** БД графически представляет собой перевернутое дерево, состоящее из объектов различных уровней.
- **Сетевые** БД имеют более одного предка, т.е. каждый элемент вышестоящего уровня может быть связан одновременно с любыми элементами следующего уровня.
- **Реляционные** БД

# Модели БД

---

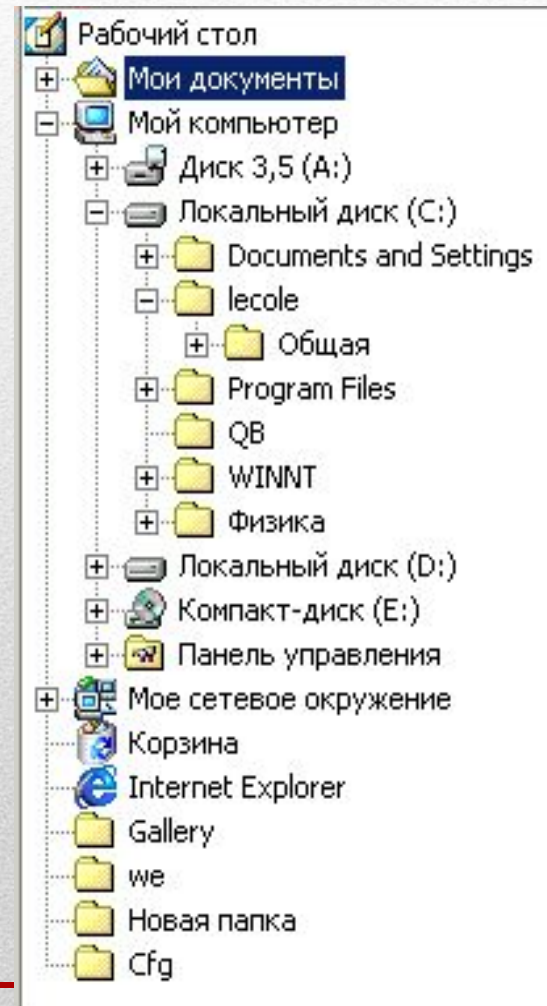
# Иерархическая БД



# Иерархическая модель

- это организация данных в виде древовидной структуры: один элемент в записи является главным, остальные — подчинёнными.

**Запись** – группа связанных между собой элементов данных (совокупность связанных полей) **логически**



# Реляционная БД

	Наименование	Автор	Объем	Цена	Примечание
▶	Война и мир	Л.Н. Толстой	784	200,00р.	Роман
	Идиот	Ф. М. Достоевский	372	125,00р.	Роман
	Лолита	Набоков	435	96,00р.	Роман
*			0	0,00р.	

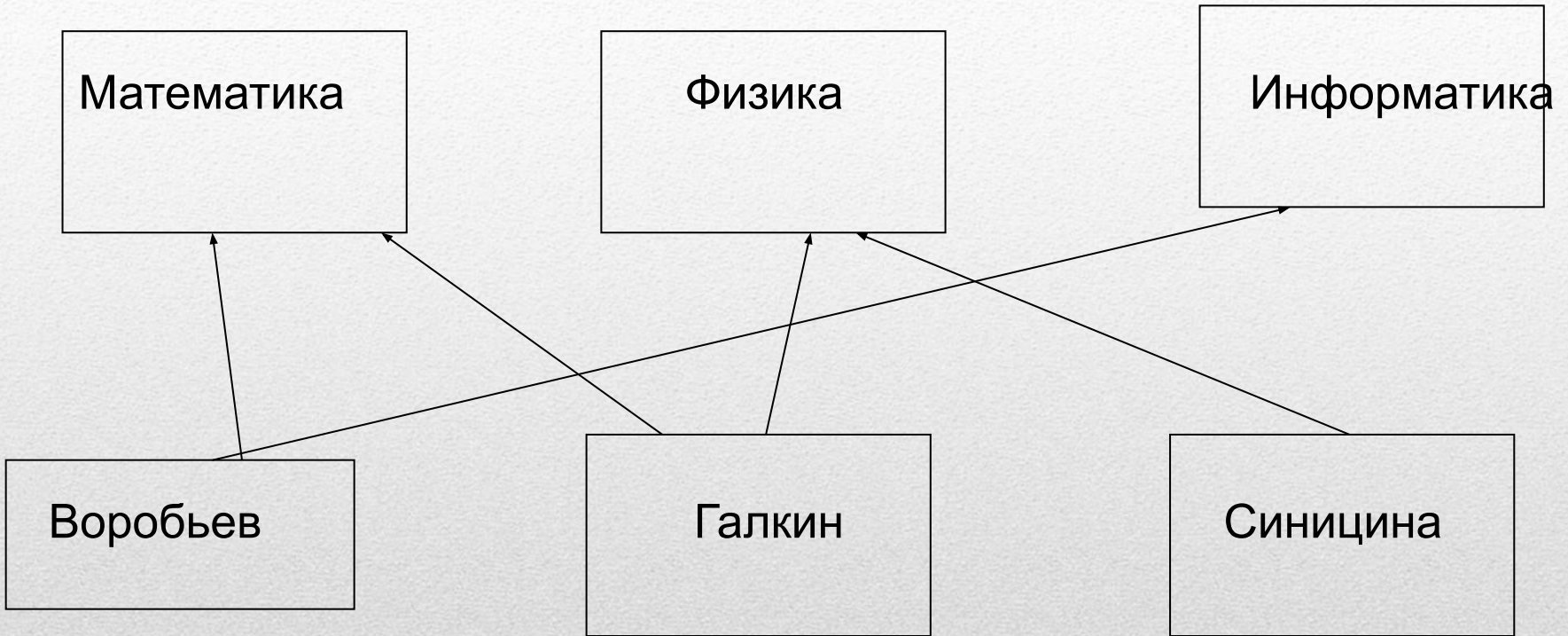
Примечание

:


ЗАПИСЬ

ПОЛЕ

# Сетевые БД





# Системы управления базами данных (СУБД)

- Пользователями БД могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных.
- В современной технологии БД предполагается, что создание БД, ее поддержка и обеспечение доступа пользователей к ней осуществляется централизованно с помощью специального программного инструментария - **СУБД**

# Системы управления реляционными базами данных

- СУБД, работающие с реляционными БД, называются **реляционными СУБД** или **СУРБД**
- Примеры СУРБД:
  - Dbase (в рус.версии – «РЕБУС»)
  - FoxBase (в рус.версии – «КАРАТ»)
  - FoxPro
  - CLIPPER
  - CLARION
  - Paradox
  - MS Access (приложение MS Office)
  - SQL-сервер

# СУБД (системы управления базами данных)

СУБД – это программа, позволяющая создавать БД, а также обеспечивающая обработку (сортировку) и поиск данных.

---

- **Создавать новые БД**
- **Изменять данные**
- **Дополнять данные**
- **Осуществлять поиск**
- **упорядочивать информацию в базе данных**
- **Записывать обновленные данные на диски**
- **Выполнять печать данных и ответов на запросы.**

# **Возможности СУБД**

---

- **Это мощная, высокопроизводительная СУРБД, предназначенная для разработки настольных (desktop) БД и создания приложений БД архитектуры «клиент-сервер».**
- Работает под управлением ОС Windows.
- Как компонент MS Office, имеет усовершенствованный интерфейс пользователя, что обеспечивает совместимость с другими программами MS Office

# MS Access

- **Таблицы** – это основные объекты БД, предназначенные для хранения данных (реляционная БД может иметь несколько взаимосвязанных таблиц)
- **Запросы** – это структуры, предназначенные для обработки данных базы
- **Формы** – это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся
- **Отчеты** – это объекты, с помощью которых данные выводят на принтер в удобном наглядном виде
- **Макросы** – это макрокоманды. Если какие-то операции с БД производятся особенно часто, имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш
- **Модули** – это программные продукты, написанные на языке Visual Basic. Если стандартных средств MS Access пользователю не хватает, он может расширить возможности системы, написав необходимые модули

# Объекты MS Access

- Поле (столбец) – это минимальный элемент данных.
- Запись (строка) – это полный набор данных об определенном объекте.
- Первичный (главный) ключ – это одно или несколько полей, совокупность значений которых однозначно определяет любую запись. Иначе говоря, значение первичного ключа не должно повторяться в разных записях.

## **Основные понятия базы данных**

---

Ф.И.О. ученика	Домашний адрес	Телефон
Петров П. П.	Ул. Садовая. 16. 27	3-15-16
Иванов И. И.	Ул. Луговая, 11, 2	3-16-17
Сидоров С. С.	Ул. Цветочная, 6, 1	3-17-18

**Поле БД** – это столбец таблицы, содержащий значения определенного свойства.

**Запись БД** – это строка таблицы, содержащая набор значений определенного свойства, размещенный в полях БД.

**Ключевой поле** – это поле, значения которого однозначно определяют каждую запись в таблице.

---



- **Поля** – это основные элементы **структуры** БД.
- **Строки** таблиц называются **записями** и содержат информацию об одном объекте.
- На практике приходится иметь дело с более сложными структурами, образованными из **нескольких таблиц**
- Структура должна быть такой, чтобы при работе с БД требовалось вводить как можно меньше данных. Если ввод каких-то данных приходится повторять неоднократно, то базу делают из **нескольких связанных таблиц**. Структуру каждой таблицы разрабатывают отдельно.

# Создание структуры БД – режим Конструктор

**Каждое поле характеризуется именем  
(название свойства) и типом.**

**Тип определяет множество значений,  
которые может принимать данное поле в  
различных записях.**

**Поле**

---

- **Счетчик.** Содержит последовательность целых чисел, которые задаются автоматически при вводе записей. Эти числа не могут быть изменены пользователем.
- **Текстовый.** Содержит до 255 символов.
- **Денежный.** Содержит числа в денежном формате.
- **Гиперссылка.** Содержит ссылку на информационный ресурс в Интернете (например, Web-сайт).
- **Мастер подстановок.** Предлагает перечень значений для выбора

# Типы полей

---

1. **Символьный**  
(текстовый, длина  $\leq 256$  символов)
2. **Поле МЕМО** (для хранения больших текстов,  $\leq 65535$  символов)
3. **Числовой**  
(целые, вещественные)
4. **Дата/время**
5. **Логический** (для хранения логических данных, принимающих лишь 2 значения: true или false, Истина или Ложь, 1 или 0, Да или Нет)
6. **Поле объекта OLE** (для хранения графической, видео, аудио информации и т.п.)

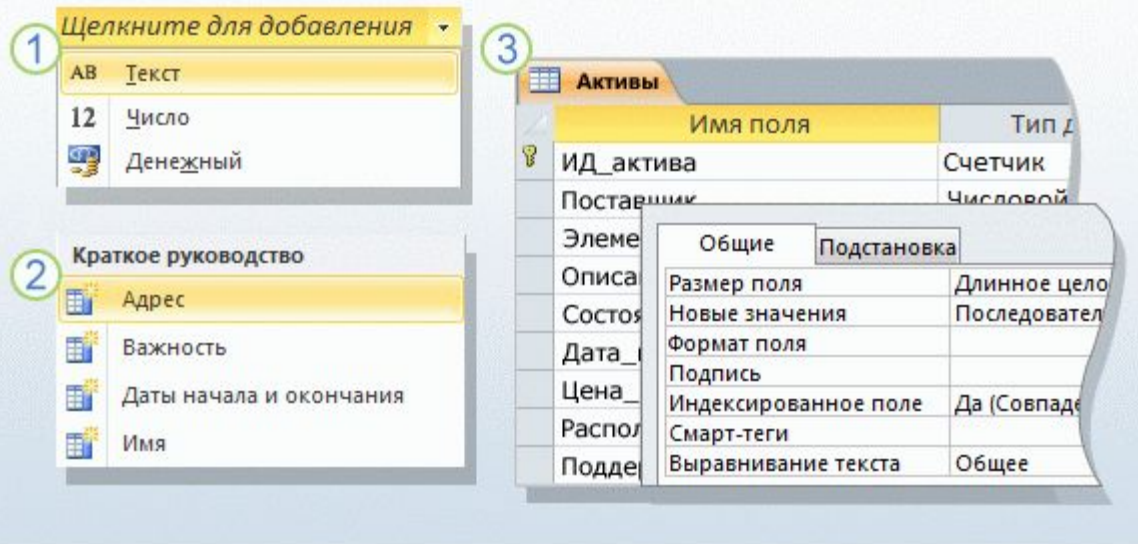
# Типы полей

---

# Свойства полей:

- **Размер поля.** Определяет максимальную длину текстового или числового поля.
  - **Формат поля.** Устанавливает формат данных.
  - **Обязательное поле.** Указывает на то, что данное поле обязательно надо заполнить.
-

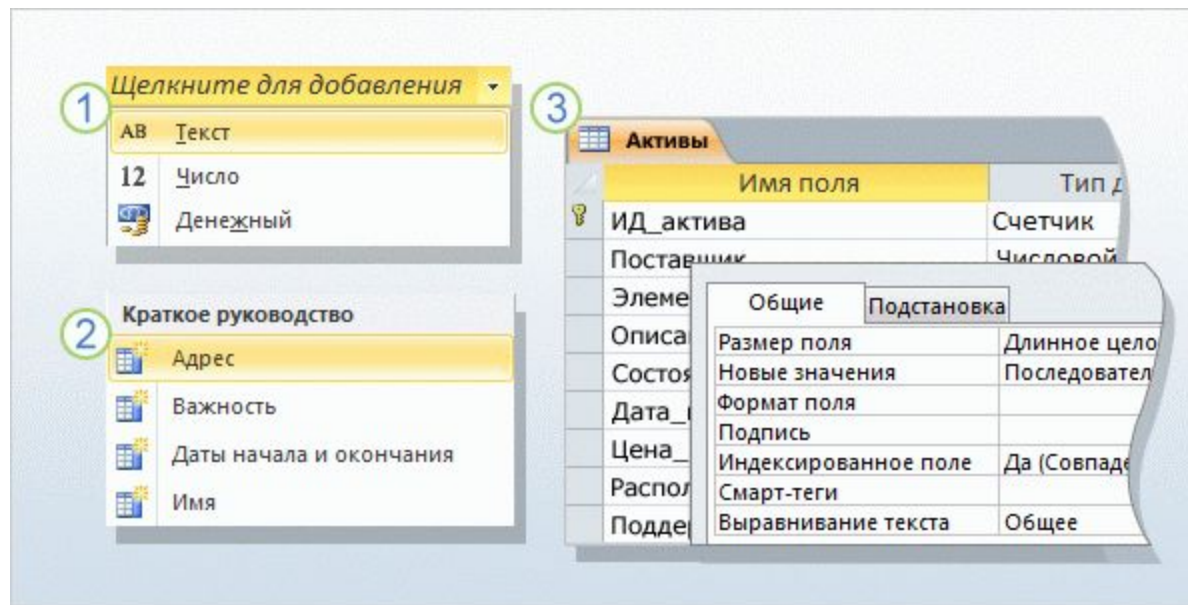
# Создание таблиц



В реляционной базе данных все сведения хранятся исключительно в таблицах, что делает их ключевым компонентом базы данных.

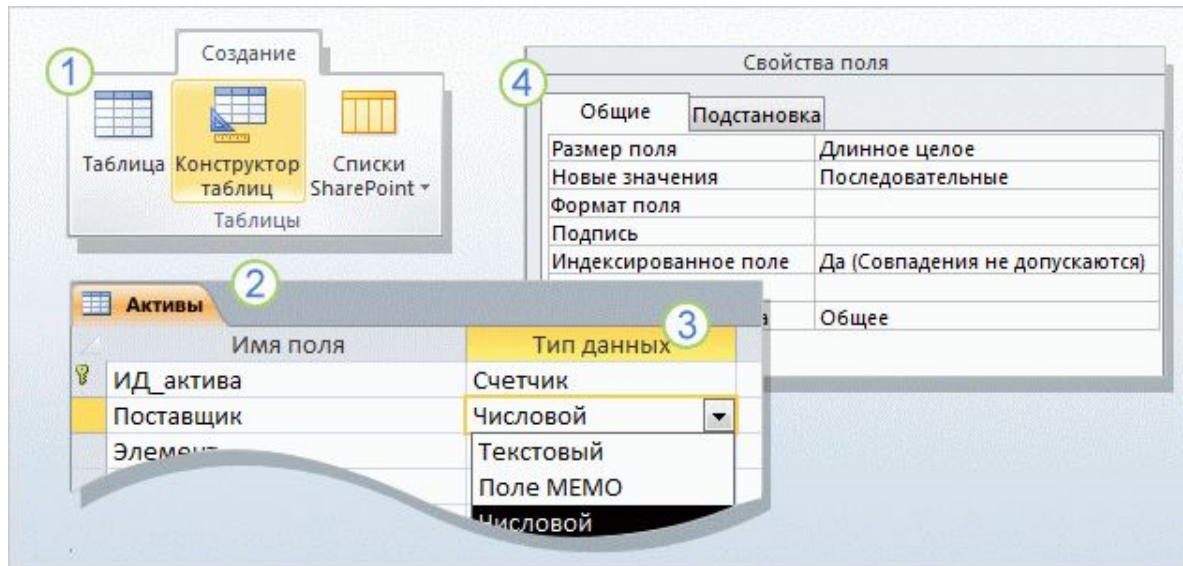
Отсутствие таблиц означает отсутствие базы данных. Для создания таблиц применяются режимы **таблицы** и **конструктор**.

# Создание таблиц в режиме конструктор



- 3 В отличие от режима таблицы, в режиме конструктора можно управлять всеми полями и свойствами таблицы. В рамках этого курса в режиме конструктора будет создана таблица и изменены значения в **поле подстановки** (поле, содержащем список вариантов выбора).

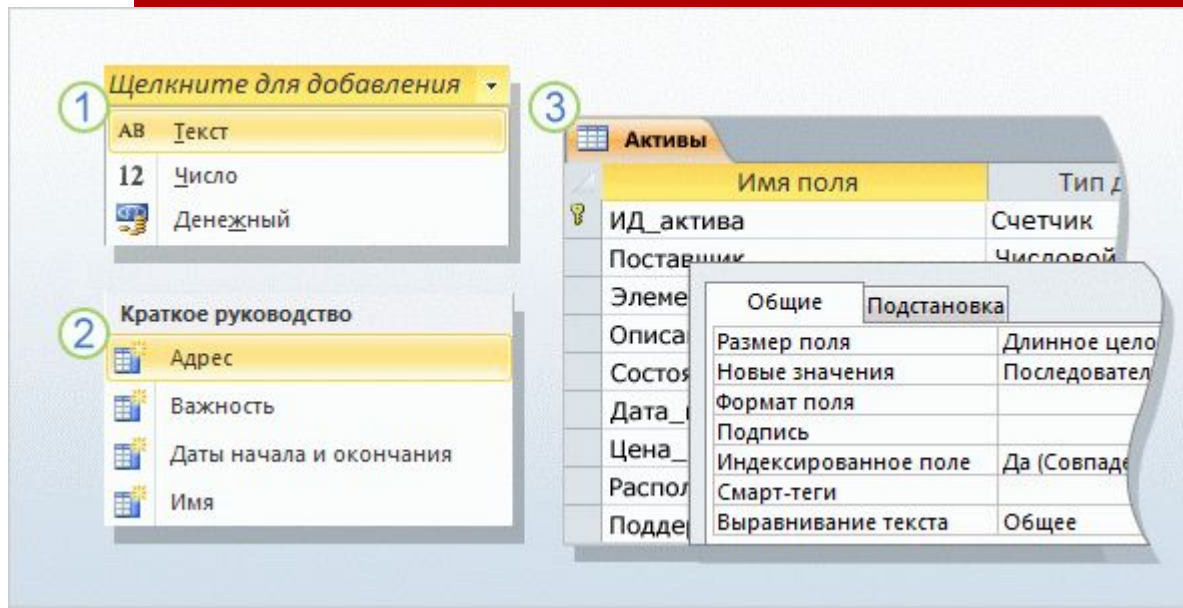
# Создание таблицы в режиме конструктора



**В режиме конструктора** можно создавать таблицы с нуля, а также устанавливать и изменять любые свойства каждого из полей.

Кроме того, в режиме конструктора можно открывать существующие таблицы для добавления, удаления и редактирования полей.





# Создание таблиц

- 1 В режиме **таблицы** для создания нового поля необходимо щелкнуть заголовок пустого поля, выбрать тип данных и ввести имя поля.

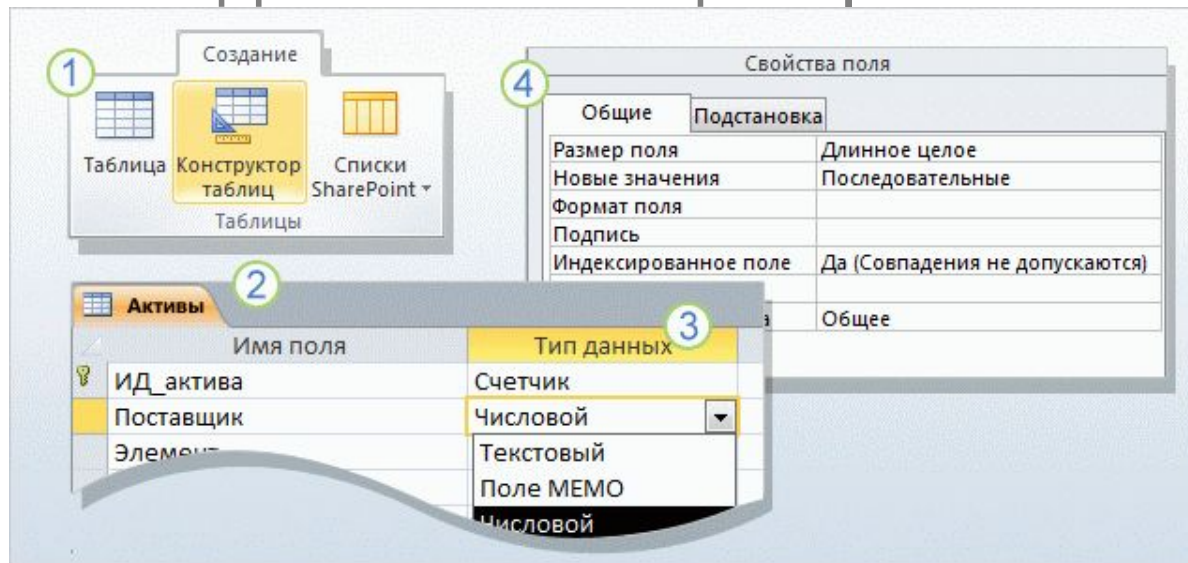
# Создание таблицы в режиме таблицы



**Режим таблицы** является наглядным инструментом для создания таблиц.

- Как при создании новой пустой базы данных, так и при добавлении таблицы в существующую новая таблица открывается в режиме таблицы.
1. Обратите внимание: новая таблица содержит поле «Код». Это первичный ключ, поэтому создавать его уже не требуется.

# Создание таблицы в режиме конструктора



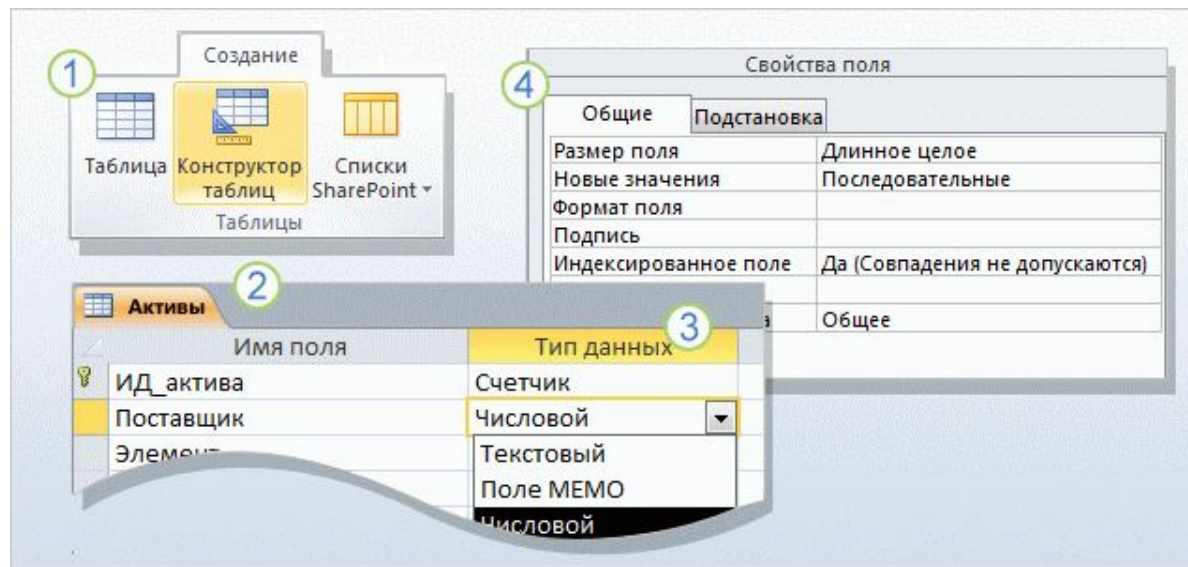
1

На вкладке **Создание** в группе **Таблицы** нажмите кнопку **Конструктор таблиц**.

2

В столбце **Имя поля** конструктора введите имена полей таблицы. Как правило, первым создаваемым полем является поле первичного ключа. Помните о том, что на данном этапе не нужно добавлять внешние ключи: это можно будет сделать при создании связей.

# Создание таблицы в режиме конструктора



- 3 В столбце **Тип данных** в списке рядом с именем поля выберите для него тип данных.
- 4 В области **Свойства поля** можно задавать свойства отдельных полей.

# Добавление и сохранение данных

1

Fabrikam	Стул
Fabrikam	Стул
Fabrikam	Стул
Contoso	Копир

2

Александр Туманов

Начальник отдела снабжения

92 из 92

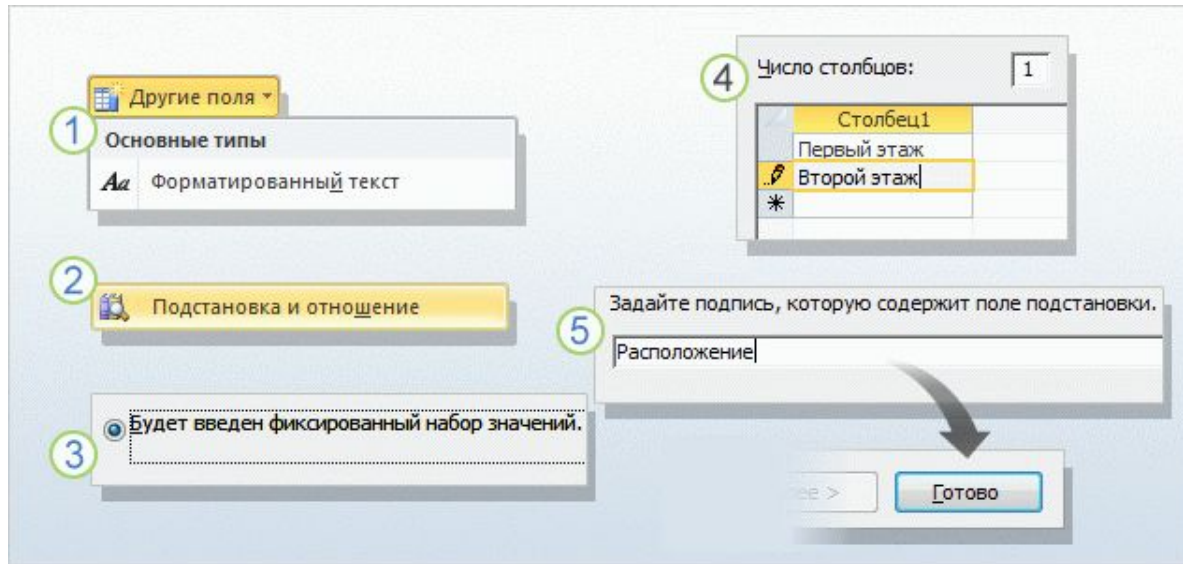
Следующая запись

Завершив создание таблиц, можно добавить несколько записей. Так проще всего протестировать таблицы и убедиться в том, что в них сохраняются нужные данные.

При этом необходимо помнить некоторые правила.

Процесс сохранения данных.

# Добавление поля подстановки в таблицу

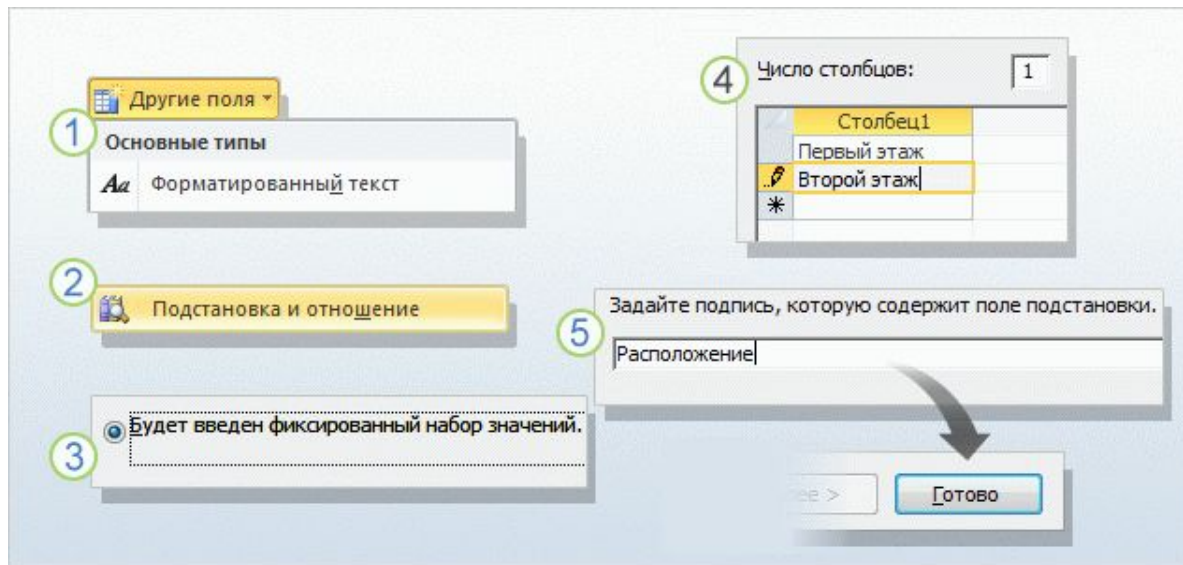


Вместо таблицы также можно использовать **поле подстановки**. Предположим, что требуется фиксировать сведения о местах хранения активов компании.

Если имеется большое количество расположений, например комнаты на нескольких этажах, для этих данных можно создать таблицу, чтобы упростить работу с ними. Однако если есть всего несколько расположений, сведения о них целесообразно хранить в поле подстановки.

Использование мастера подстановок.

# Добавление поля подстановки в таблицу

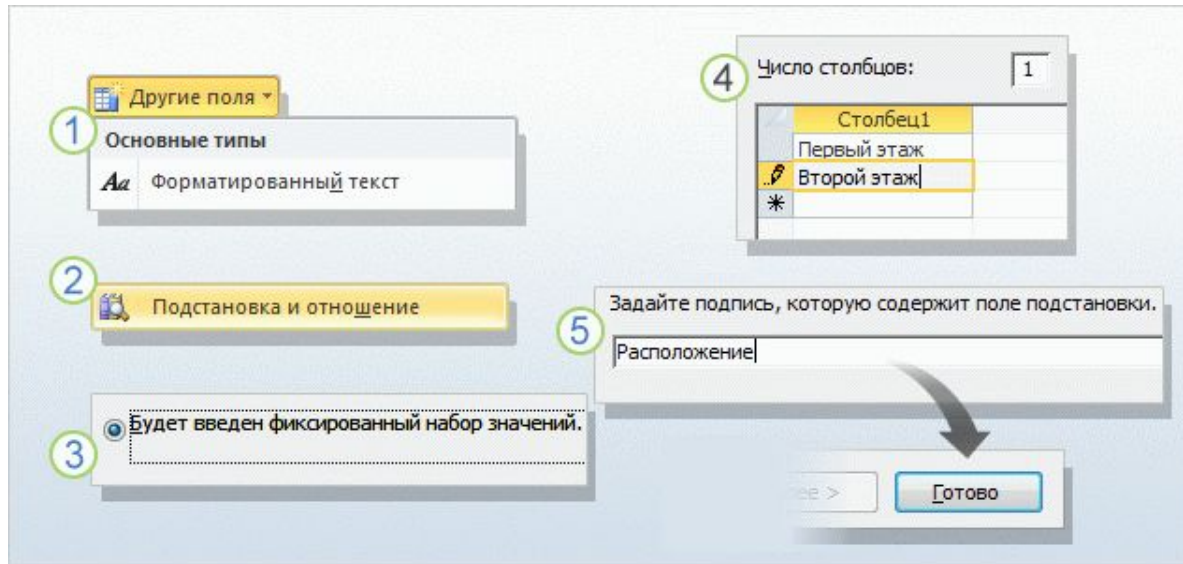


Список вариантов может храниться в самом поле подстановки либо загружаться из поля в другой таблице.

Ниже приведены инструкции по созданию поля подстановки с внутренним списком вариантов выбора (в приложении Access это называется **списком значений**).

Использование мастера подстановок.

# Добавление поля подстановки в таблицу

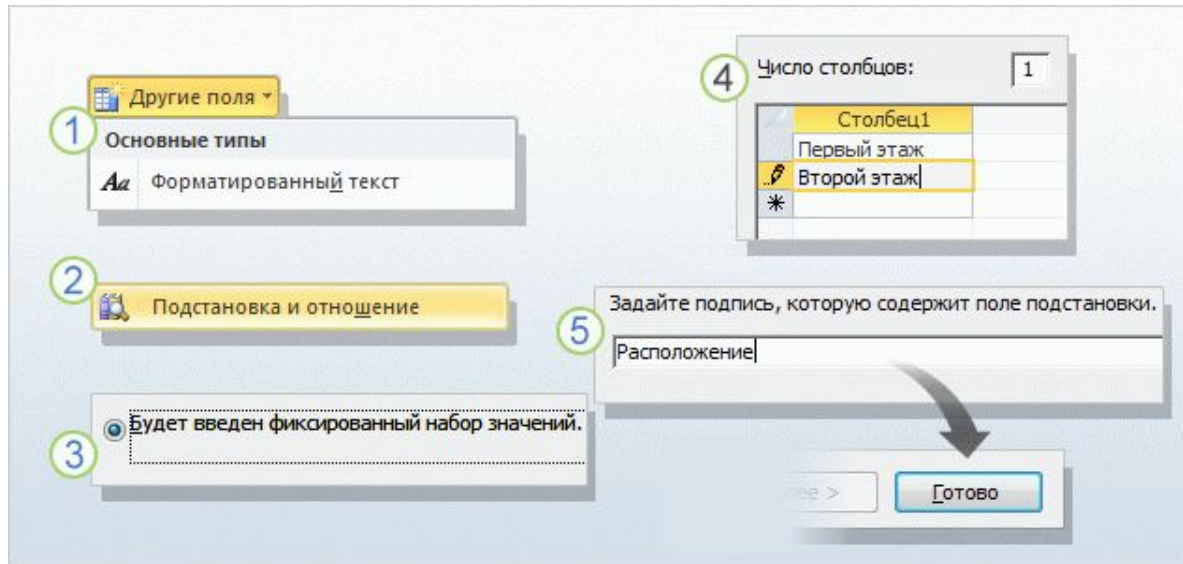


- 1 Откройте таблицу в режиме таблицы и на вкладке **Поля** в группе **Добавление и удаление** выберите команду **Другие поля**.
- 2 Выберите в меню пункт **Подстановка и отношение**. Будет запущен **мастер подстановок**.
- 3 На первой странице мастера выберите параметр **Будет введен фиксированный набор значений** и нажмите кнопку **Далее**.

Использование мастера подстановок.



# Добавление поля подстановки в таблицу

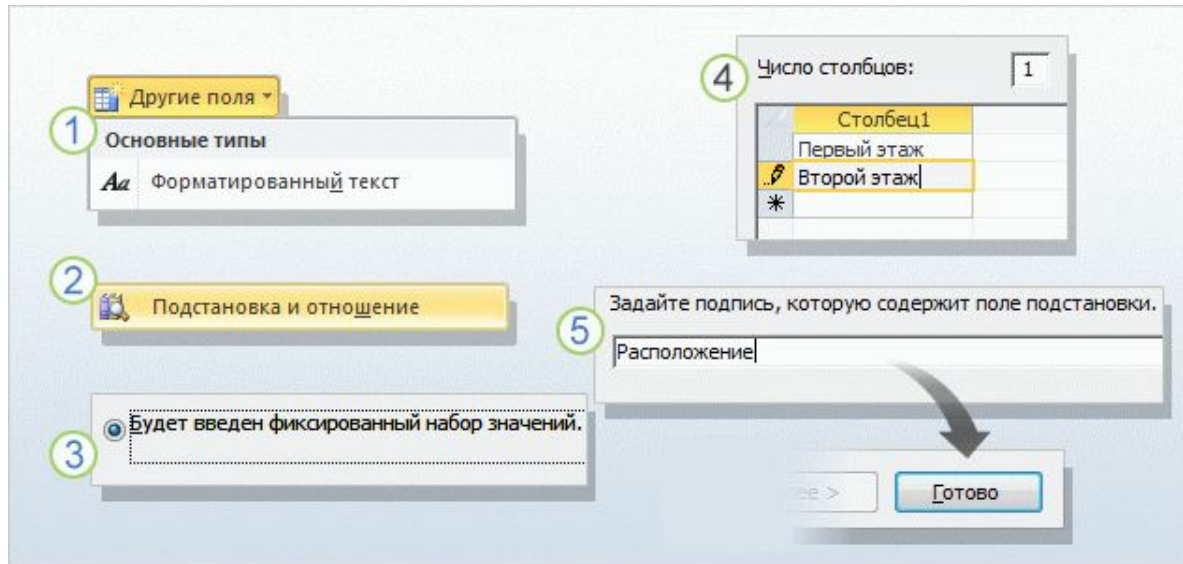


Список вариантов может храниться в самом поле подстановки либо загружаться из поля в другой таблице.

Ниже приведены инструкции по созданию поля подстановки с внутренним списком вариантов выбора (в приложении Access это называется **списком значений**).

Использование мастера подстановок.

# Добавление поля подстановки в таблицу



- 4 На следующей странице мастера введите в поле **Число столбцов** значение 1, а затем введите варианты выбора в таблицу (по одному в строке).
- 5 На третьей странице мастера введите имя для нового поля и нажмите кнопку **Готово**.

Использование мастера подстановок.