

## 6.5. Создание реляционной БД в среде СУБД ACCESS

### Общие сведения

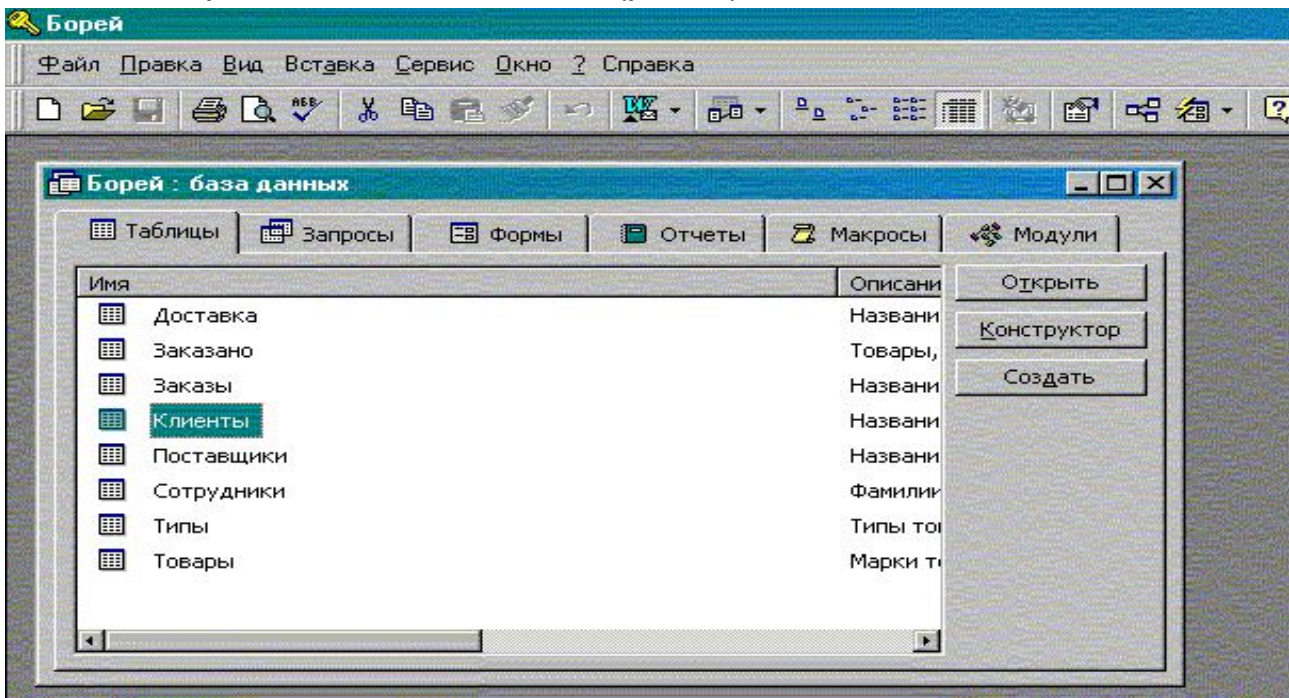
Реляционные отношения в СУБД ACCESS представлены в двух формах: в виде таблиц и в виде связей между таблицами. Количество таблиц и их размеры ограничиваются только объемом памяти компьютера.

Каждая таблица состоит из полей (именованных столбцов) и записей (строк). Записи не имеют имен, но автоматически нумеруются.

Между таблицами, имеющими одинаковые поля, разрешаются связи двух типов: 1:1 и 1 : M. Связи M : M реализуются только через третью, промежуточную таблицу.

### Создание таблиц

Чтобы начать разработку новой базы данных, следует создать пустую базу данных командой \*Файл \* Создать \*Новую БД. На экране появится окно БД (рис. 1).



Щелкнув мышью по экранной кнопке "Создать", Вы перейдете в режим создания новой таблицы. ACCESS предлагает выбрать таблицы и поля из существующих шаблонов либо самостоятельно определить имена, типы, длину полей и прочие их свойства.

Независимо от способа создания таблицы режим "Конструктор" позволяет в любой момент изменить ее структуру и свойства (рис.2).

Независимо от способа создания таблицы режим "Конструктор" позволяет в любой момент изменить ее структуру и свойства (рис.2).

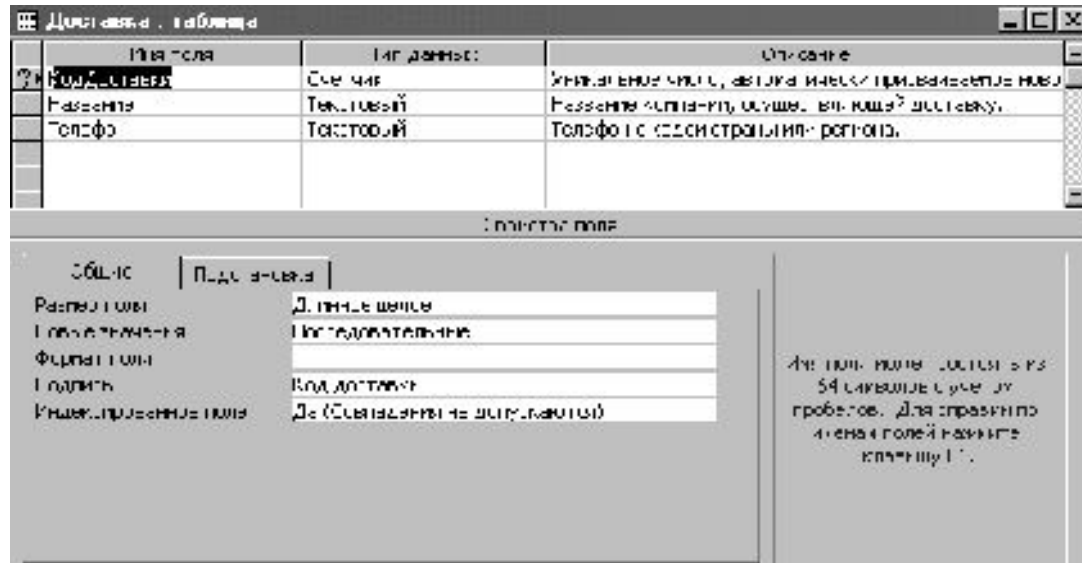


Рис.2. Окно конструктора таблиц

Для создания новой таблицы необходимо дать ей имя и описать структуру, а также выбрать и назначить ключевое поле. Обязательными свойствами каждого поля являются имя, тип и размер.

Таблицы и поля таблиц должны иметь уникальные имена, которые могут включать до 64 любых символов, кроме символов . ! ' [ и ] . Рекомендуется не включать в имя пробелы и избегать слишком длинных имен; имена из нескольких слов рекомендуется писать слитно, без пробелов и каждое слово начинать с заглавной буквы. Примеры имен: Студенты, ПродажиЗаМесяц.

**Тип данных** определяет, какого вида данные будут храниться в поле - текст, числа, даты и т.д. и сколько места они будут занимать на диске. Важно правильно определить тип поля до того, как начнется ввод данных, так как при изменении типа данные могут быть искажены или потеряны. Наиболее часто используются следующие типы данных:

- текстовый (до 255 символов);
- числовой с разной степенью точности,
- дата / время,
- примечания (МЕМО) - до 64000 символов.

Реже применяют типы:

- счетчик (для служебных полей, типа КодТовара и т.п.),
- денежный,
- логический (да / нет)
- OLE (для хранения данных, сформированных другими прикладными программами - рисунков, схем, звукозаписей, форматированных текстов и т.п.)

**Размер поля** назначается системой по умолчанию, автоматически, но разработчик должен подобрать наиболее оптимальный размер, чтобы уменьшить затраты памяти ЭВМ на хранение информации. Например, если номер зачетки включает не более 5 символов, то целесообразно выбрать размер поля = 5, а не 20.

Поля таблицы имеют ряд дополнительных свойств, которые позволяют проверять значения данных, облегчают разработку прикладных программ.

**Ключевые поля** однозначно идентифицируют своим значением каждую конкретную запись, то есть значения ключевого поля в любых двух строках таблицы должны отличаться. Общепринятые правила при выборе ключа:

в качестве ключа чаще всего выбирают числовой или символьный код, который используется только для внутренних целей БД и не доступен для изменения пользователем,

тип ключевого поля - "счетчик" или "числовой",

имя ключевого поля совпадает с именем понятия, для которого составлена таблица (например, имя ключа не КодТовара, а просто Товар).

Для определения ключа служит команда \*Правка \*Ключевое поле, в результате должен появиться значок ключа слева от имени поля.

### **Связи между таблицами**

Связи между таблицами являются необходимым элементом структуры БД. Для того, чтобы связь была возможна, таблицы должны иметь общие поля. Чаще всего в одной таблице для связи используется ключ, а в другой таблице это поле не является ключом и называется "внешним ключом". Например, для сопоставления сведений о сотрудниках и о принятых ими заказах, следует определить связь по полю "Код Сотрудника" в двух таблицах: СОТРУДНИКИ (Код Сотрудника, Фамилия, Имя) и ЗАКАЗЫ (Код заказа, Код клиента, Код сотрудника). В первой таблице общее поле является ключом, а во второй - внешним ключом

После того, как определены поля для связи, следует оценить тип связи.

**Отношение "один-ко-многим"** является наиболее часто используемым типом связи между таблицами. Например, между таблицами "Сотрудники" и "Заказы" существует отношение "один-ко-многим": каждый сотрудник может принять несколько различных заказов, но за каждый заказ отвечает один конкретный сотрудник

**Отношение "многие-ко-многим"** реализуется только с помощью третьей таблицы, ключ которой состоит из ключевых полей тех таблиц, которые необходимо связать. Например, между таблицами "Заказы" и "Товары" имеется отношение "многие-ко-многим", которое реализовано с помощью дополнительной таблицы "Заказано".

**Отношение "один-к-одному"** В этом случае каждая запись в одной таблице может быть связана только с одной записью в другой таблице и наоборот. Этот тип связи используют редко, поскольку такие данные могут быть помещены в одну таблицу. Например, такую связь используют для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты и т.п.

### **Для определения связей между таблицами следует:**

1. закрыть все открытые таблицы и выполнить команду меню \* Сервис \*Схема данных”
2. добавить таблицы в окно “Схема данных”;
3. перенести с помощью мыши ключевое поле одной таблицы в другую,
4. определить тип связи (1:1, 1:M, M:1) (двойной щелчок по линии связи открывает окно определения типа связей).

Обратите внимание, что *нельзя изменить тип данных для поля, которое связывает таблицу с другой таблицей*. Предварительно нужно удалить установленную связь.

## Контрольные вопросы

1. Сколько баз данных может быть открыто одновременно?
2. Каковы правила Access для имен объектов?
3. Какое расширение принято для файлов БД?
4. Каков порядок определения структуры БД в СУБД ACCESS?
5. Сколько таблиц может содержать база данных ACCESS?
6. В каком режиме описывается структура таблицы БД?
7. Можно ли изменить структуру таблицы после ее создания?
8. Какие типы полей БД предусмотрены в ACCESS?
9. Какая информация может храниться в полях типа "объект OLE"?
10. Какие свойства поля должны быть определены обязательно?
11. Что такое ключевое поле?
12. Для чего необходимо описывать связи между таблицами базы данных?
13. Какие типы связей поддерживает СУБД ACCESS?