

**Группа ТЭК 2/2**

**Дата 18.01.23**

**Вид занятия** Лекция

**Тема:** Системы управления базами данных

**Цель занятия:**

- **дидактическая** – выучить системы управления базами данных .
- **воспитательная** – поощрять студентов к изучению предмета с целью последующего использования ПК в учебной и будущей профессиональной деятельности

### **План занятия**

1. Создание базы данных
2. Создание таблиц
3. Связи между таблицами

### **Литература:**

**Основная литература :**

1. Михеева Е.В. Информационные технологии в профессиональной деятельности: учеб. пособие для студ. сред. проф. образования. – 9-е изд., стер. – М.: Издательский центр «Академия», 2014. – 256 с.
2. Михеева Е.В. Практикум по информационным технологиям в профессиональной деятельности: учеб. пособие для студ. учреждений сред. проф. образования. – 10-е изд., стер. – М.: Издательский центр «Академия», 2014. – 192 с.

**Дополнительная литература:**

1. Войтюшенко Н.М. Информатика и компьютерная техника: Уч. пос. баз. подготовки для студ. экон. и техн. специальностей дн. и заоч. форм обучения /Н.М.Войтюшенко, А.И.Остапец. – Донецк: ДонНУЭТ, 2014 – 485 с.

### **1. Создание базы данных**

Для создания своей базы данных, работы с ней или с готовыми примерами, необходимо запустить *Access*.

Кнопка **Пуск** → п. **Программы** → п/п **Microsoft Access** -

Для создания базы данных выбрать **Новая база данных** → указать папку и имя базы.

В первичном окне программы откроется окно базы данных, в котором будут создаваться все остальные объекты (рисунок 1).

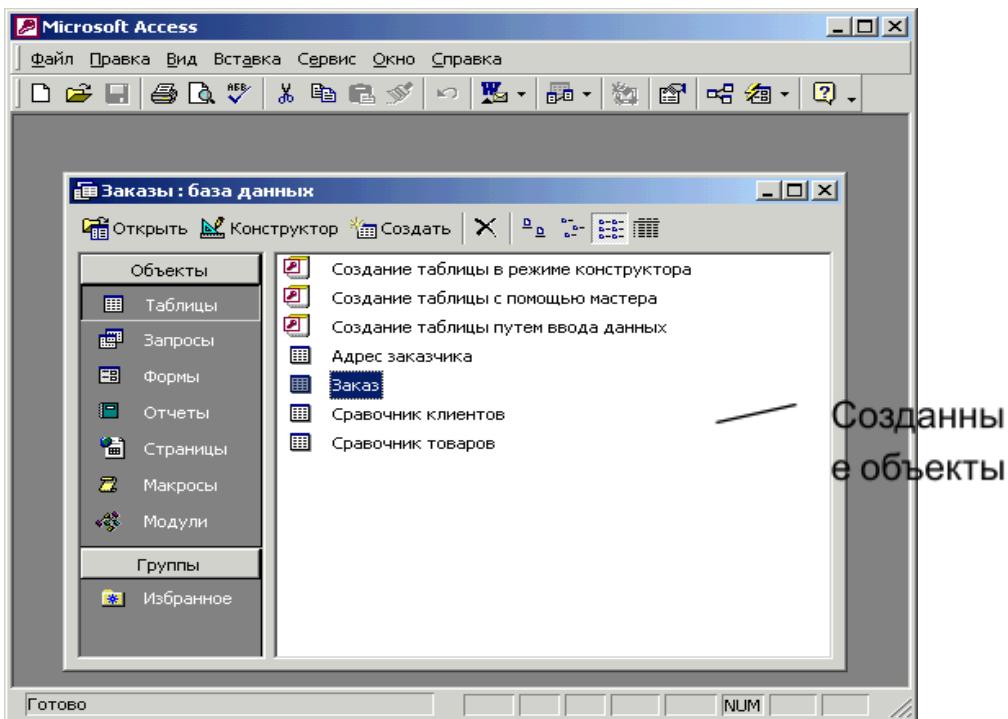


Рисунок 1 - Окно Access и базы данных Заказы

## 2. Создание таблиц

Access предоставляет пользователю 5 способов создания таблиц:

- в режиме таблицы,
- в режиме конструктора,
- с помощью мастера таблиц,
- путем импорта готовой таблицы,
- с помощью установления связи с другой базой данных.

Режим таблицы вызывается с помощью значка **Создание таблицы** путем **ввода данных**. Пользователю предоставляется пустая таблица похожая на обычную электронную таблицу, куда можно непосредственно вводить данные. Стандартные названия колонок **Поле1**, **Поле2** и т.п. заменяют на содержательные путем двойного щелчка по имени поля либо через контекстное меню (м. **Формат** → к. **Переименовать**). Имя поля может быть на языке пользователя, не длиннее 64 символов, без служебных знаков. Пробел в именах использовать можно. Длина самого поля не должна превышать 64 символов и может состоять из букв, цифр, пробелов и некоторых специальных символов. При заполнении таблицы, СУБД анализирует данные и определяет их тип. Ввод данных в строку, форматирование таблицы аналогичен соответствующим операциям в табличном процессоре *Excel*. Access автоматически сохраняет запись (строку)

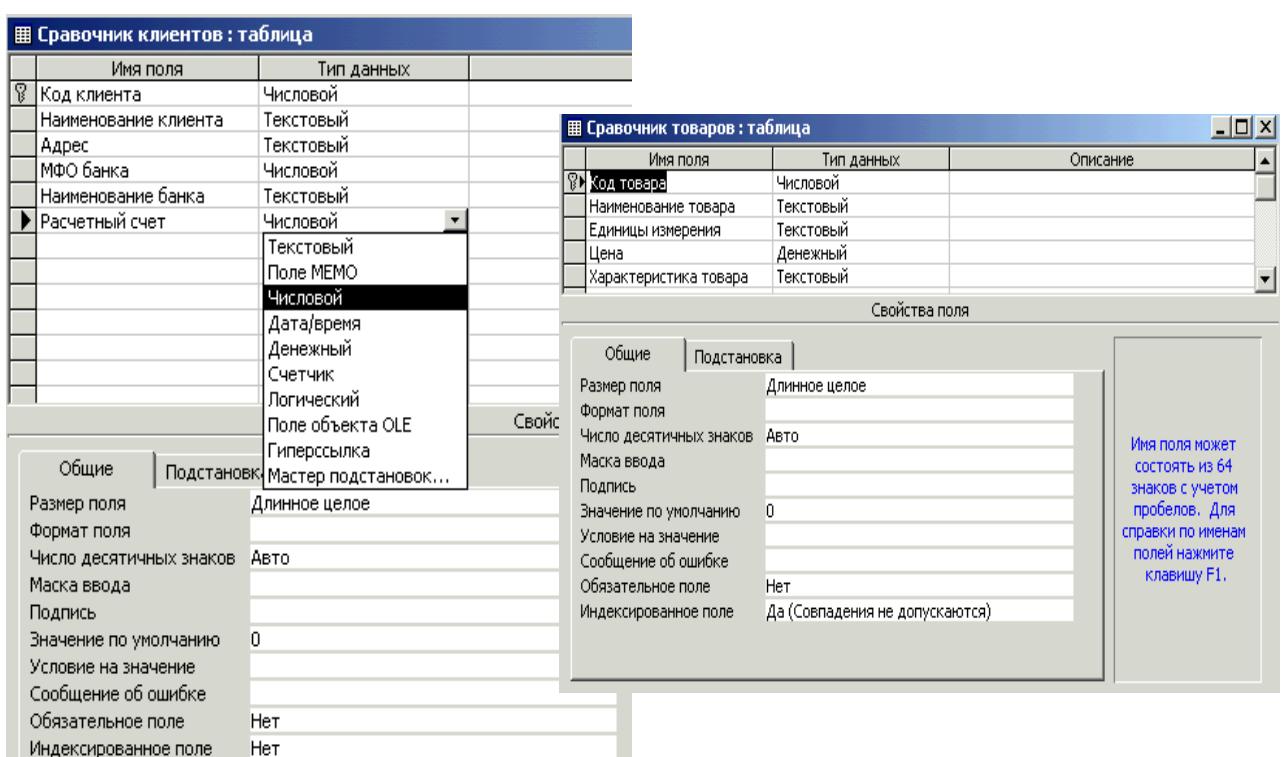
таблицы), как только пользователь покинет её пределы. При закрытии таблицы у пользователя запрашивается её имя. Название таблицы может содержать не более 64 символов, в т.ч. пробелы. Запрещено использовать в именах точки, !, [.]. Каждая таблица должна иметь ключевое поле, которое СУБД предлагает создать автоматически с именем **Код** и типом данных **Счетчик**. В этом случае записи будут нумероваться последовательно. Если данный вариант не подходит, например в наших таблицах ключевые поля имеют другую значность, то следует ответить **Нет**, а после создания таблицы определять ключевое поле самостоятельно в режиме конструктора.

**Режим конструктора** разрешает руководить процессом создания структуры таблицы, задавая каждому полю тип и соответствующие свойства.

Режим конструктора вызывается с помощью значка **Создание таблицы в режиме конструктора**.

Access предлагает заполнить специальный бланк (рисунок 2), в котором определяются поля входящие в таблицу.

При заполнении бланка можно вводить дополнительную информацию в табличке **Свойства поля**, находящейся в нижней части бланка (размер поля, формат поля, число десятичных знаков, подпись, значение по умолчанию, условие на значение, сообщение об ошибке, обязательное поле, индексированное поле).



## Рисунок 2 – Бланки создания таблиц Справочник клиентов и Справочник товаров в режиме конструктора

Каждая таблица должна иметь ключ. Чтобы определить ключевое поле, необходимо выделить щелчком нужное поле и щелкнуть по инструментальной кнопке с нарисованным ключом либо меню **Правка** → п. **Ключевое поле**. *Access*, в случае отсутствия ключевого поля, автоматически добавляет первым ключевое поле в виде счетчика. Если пользователя это не устраивает, то можно переопределить ключевое поле таким же способом, как и его определение.

Пример 1. - Создать таблицы справочников и таблицу заказов

Выполнение. Кнопка **Пуск** → к. **Программы** → п. **Microsoft Access** → селекторная кнопка **Новая база данных** → указать папку и имя базы – **Заказы** → указать объект **Таблицы** → значок **Создание таблицы в режиме конструктора**. → заполнить бланк для **Справочника клиентов** (рис.1) → щелкнуть по полю **Код клиента** → щелкнуть по кнопке → закрыть окно бланка стандартным способом → на вопрос о сохранении таблицы щелкнуть по кнопке **Да** → дать имя таблице **Справочник клиентов** → **OK** → на вкладке таблицы должен появиться значок **Справочник товаров**.

Аналогичным образом создать и сохранить таблицы **Справочник товаров** (бланк конструктора показан на рисунке 2), **Заказ** (бланк конструктора показан на рисунке 3).

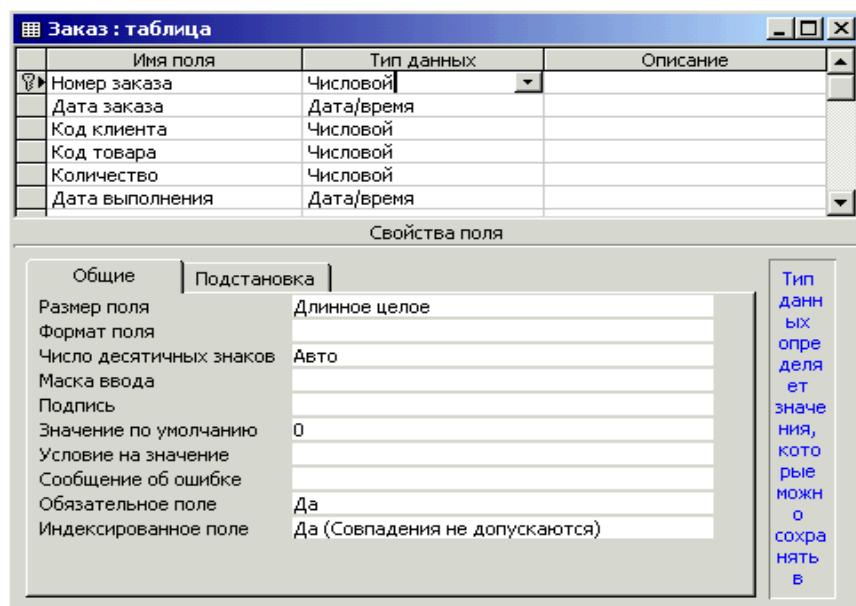


Рисунок 3 – Бланк создания таблицы **Заказы** в режиме конструктора

Созданную форму таблицы нужно сохранить: меню **Файл** → п. **Сохранить** → дать имя таблице (например, **Заказ**) → **OK**. Другие таблицы создаются аналогично. Следует создать **Справочник клиентов** и **Справочник товаров**.

Созданные таблицы пока пусты. Таблицы простой структуры типа справочников, информация которых носит относительно статический характер, могут заполняться стандартными средствами *Access*: вкладка **Таблицы** → выбрать из списка название справочника (например, **Справочник товаров**) → кнопка **Открыть**.

Далее идет простое заполнение строк таблицы с клавиатуры (таблица 1., таблица 2).

Таблица 1. - **Справочник клиентов**

Код клиента	Наименование клиента	Адрес	МФО банка	Наименование банка	Расчетный счет
11	ООО Мечта	г. Харцызск	125632	АК ПИБ	123654789
12	МЧП Радуга	г. Горловка	456321	Аваль	147852369
13	ЧП Корниенко	г. Донецк	235421	ДОД УСБ	741258963
14	ДМЗ	г. Донецк	256642	Ворош.отд. ПИБ	142536987
15	Стирол	г. Горловка	582415	ПИБ	633214548

Таблица 2. - **Справочник товаров**

Код товара	Наименование товара	Единицы измерения	Цена	Характеристика товара
1	маргарин сливочный	пачка, 250	1,50р.	
2	маргарин солнечный	пачка, 250	1,10р.	
3	маргарин молочный	кг	3,40р.	

Код товара	Наименование товара	Единицы измерения	Цена	Характеристика товара
4	шоколадный десерт	пачка, 250	2,80р.	
5	масло сливочное	кг	12,50р.	

Строки таблиц имеют маркировку: с ключом - ключевое поле; с треугольником - текущая запись; с карандашом – изменяемая запись.

Редактирование и удаление записи идет обычным для приложений Microsoft Office способом. Причем *Access* следит за соблюдением целостности базы данных, т.е., чтобы не было нарушений в связанных таблицах.

Мастер таблиц загружает специальную программу, которая помогает пользователю, последовательно выполняя указания, создавать таблицу на основе существующих шаблонов.

Режим импорта разрешает выбрать существующий файл с данными и импортировать его в таблицу *Access*. В частности, этот режим разрешает импортировать таблицы *Excel*.

С помощью установления связи с таблицами другой БД, можно связать рабочую базу данных с одной или несколькими таблицами другой базы данных. При этом связанные таблицы будут использоваться так, будто они принадлежат к одной базе.

### 3. Связи между таблицами

Как правило, СУБД обрабатывает несколько таблиц. Как установить связь между таблицами? Связь между таблицами устанавливается по значениям некоторых полей. В теории СУБД известны четыре варианта связей между двумя таблицами:

1. **Отношение один – к – одному.** При таком отношении каждой записи первой таблицы соответствует не более одной записи второй таблицы, и наоборот. Можно сказать, что две таблицы, связанные отношением один к одному, представляют собой одну таблицу, разделенную надвое одной чертой, так что часть полей оказалось в одной таблице, а часть – в другой.

2. **Отношение один – ко – многим.** При этом каждой записи первичной (базовой) таблицы может соответствовать несколько записей другой таблицы. Например, одному коду товара **Справочника товаров**

может соответствовать множество записей в таблице заказов.

3. **Отношение многие – к – одному.** При таком отношении любой записи той таблицы базы данных, связь с которой мы рассматриваем, может соответствовать любое количество записей созданной нами новой таблицы, но не наоборот. Этот тип отношений наиболее распространен. Именно возможность создавать межтабличные связи типа многие-к-одному и отличает реляционные СУБД от обычных процессоров плоских файлов. Можно сказать, что это отношение один-ко-многим наоборот. Например, повторяющемуся коду товара в таблице заказов соответствует только один код товара в **Справочнике товаров**.

4. **Многие – ко – многим.** В этом случае каждой записи одной из таблиц может соответствовать любое количество записей другой таблицы, и наоборот. Например, связь таблиц заказов и товаров.

*Access* требует, чтобы у полей, по которым связываются две таблицы между собой, были одинаковые типы данных и размер.

Для связываемых таблиц одна таблица выступает в роли главной, другие – подчиненные. В подчиненной таблице надо определить поле **вторичного (внешнего) ключа**, тип данных и размер которого совпадают с полем первичного ключа главной таблицы. Обычно обоим полям присваивают одинаковые имена (но это не обязательно). Если, например, взять таблицу **Заказ**, то это будет **Код клиента**. Вторичные ключи отличаются от первичных тем, что для них допускается наличие одинаковых значений полей. Для задания вторичного ключа в подчиненной таблице в режиме конструктора следует выбрать поле вторичного ключа, например, **Код клиента** (рисунок 3). В области свойств поля перейти в “индексированное поле” (последнее по списку) и выбрать элемент **ДА** (совпадения допускаются). Аналогично должно выглядеть индексированное поле для **Кода товара** таблицы **Заказ**. Измененную структуру таблицы необходимо сохранить.

Далее следует закрыть главную и подчиненную таблицы. Затем выполнить следующие действия: меню **Сервис** → команда **Схема данных**. Двойной щелчок мышью по имени таблицы в диалоговом окне **Добавление таблицы** (рисунок 4), помещает таблицу в окно **Схема данных**.

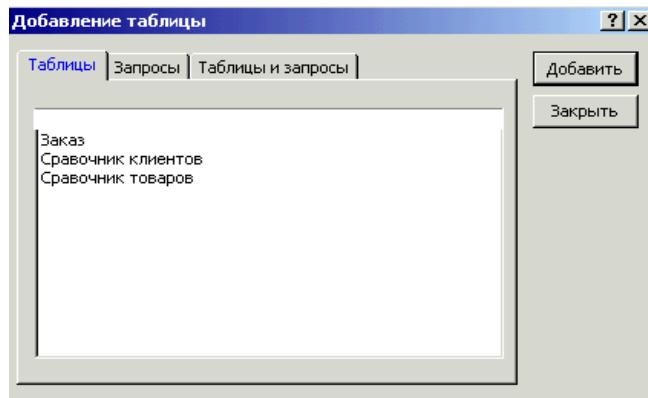


Рисунок 4 - Окно **Добавление таблицы** для перенесения связываемых таблиц в окно **Схема Данных**

Пример 2. - Создать схему связи для таблиц базы данных **Заказы**

Выполнение. 1. Открыть таблицу **Заказ** в конструкторе и проверить, чтобы для полей **Код клиента** и **Код товара** в таблице **Свойства поля** вкладка **Общие** по строке **Индексированное поле** имели параметр **Да**.

2. Меню **Сервис** → команда **Схема данных** → Если окно **Добавление таблицы** не появилось поверх окна **Схема данных**, то щелкнуть по кнопке **Отобразить таблицу**  → выделить имя таблицы **Заказ** → кнопка **Добавить** → аналогично добавить в окно схемы данных справочники клиентов и товаров → кнопка **Закрыть** окна **Добавление таблицы** → перетащить название поля **Код клиента** из таблицы **Справочник клиентов** на поле **Код клиента** таблицы **Заказ** → включить флајок **Обеспечение целостности данных** в диалоговом окне **Связи** → проверить индикатор выбора отношений (должно быть «один-ко-многим») → кнопка **Создать**. Отношение будет представлено на экране в виде линии, проведенной между связующими полями таблиц (рисунок 5). Закройте окно с сохранением выполненных установок. Аналогично следует установить связь по полю **Код товара** между **Справочником товаров** и таблицей **Заказ**.

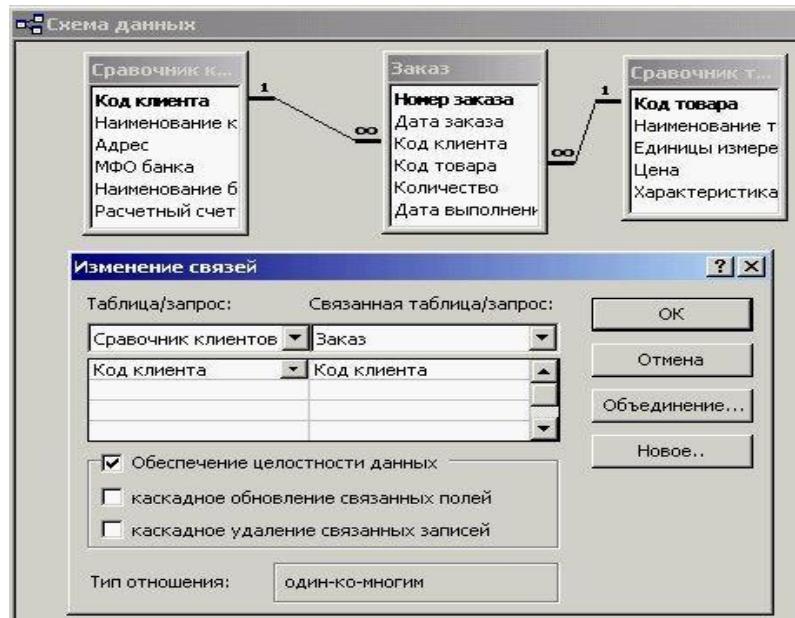


Рисунок 5 – Окно **Схема данных** с установленными связями между таблицами

Редактирование, удаление и добавление записей в связанные таблицы строго контролируется *Access*. Например, если редактируемая запись в подчиненной таблице, то она будет сохранена лишь тогда, когда значение связующего поля присутствует в главной таблице. При редактировании главной таблицы можно удалить запись, если она не связана с записями подчиненной таблицы. Ввод в подчиненные таблицы будет сохранен, если в новых записях используются вторичные ключи со значениями, которые уже имеются в главной таблице. Например, с кодами товаров, которые уже введены в справочник.

Конспект прислать по адресу  
[svetlana.avilova@gmail.com](mailto:svetlana.avilova@gmail.com)