

ЛАБОРАТОРНАЯ РАБОТА №1

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ СРЕДСТВАМИ MICROSOFT ACCESS

Цель: Изучение возможностей **Access** для реализации проекта конкретной базы данных. Создание реляционных таблиц и установление межтабличных связей.

Краткие теоретические сведения

База данных (БД)- это специальным образом организованное компьютерное хранилище данных

Система управления БД (СУБД) — комплекс программных и языковых средств (программная система), предназначенный для:

- создания БД;
- обеспечения эффективного доступа к данным в БД;
- управления данными в БД;
- поддержания БД в актуальном состоянии для предметной области.

Microsoft Access – система управления базами данных (СУБД).

Microsoft Access – система управления реляционными базами данных (РСУБД).

Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц - реляционных таблиц. Таблицы организованы в виде строк и столбцов, где заголовки каждого из столбцов называются полями, а строки – записями.

Реляционная таблица – Таблица Сотрудники

Код сотрудника	Фамилия	Имя	Отчество	Должность
1	Петров	Иван	Иванович	Менеджер
2	Петрова	Ирина	Юрьевна	Бухгалтер
3	Петров	Иван	Иванович	Менеджер

Рисунок 1 – Пример реляционной таблицы БД

Каждая реляционная таблица БД хранит данные об определенном объекте (сущности) предметной области.

Объекты (сущности) – это вовлеченные в деятельность люди, предметы, события, понятия, документы, места и т.п., информация о которых

должна накапливаться в разрабатываемой БД и которые представляют интерес для пользователей.

Поле – способ представления типа данных определяющий информационное свойство объекта.

На поля в таблицах БД могут быть наложены ограничения. Например:

- уникальность значений в атрибуте;
- значения по умолчанию;
- обязательное указание;
- диапазон значений;
- список значений;
- условия на значения.

Для однозначного определения каждой записи таблица должна **иметь уникальный ключ (первичный ключ)**.

Первичный ключ – это поле или набор полей, значения в которых позволяют однозначно определять запись в таблице.

Требования к первичному ключу:

- должен однозначно определять строку таблицы;
- должен быть уникален;
- не должен содержать пустых значений (должен быть обязательно указан, NOT NULL);
- не должен быть избыточным.

Выбор первичного ключа лежит на разработчике БД.

Поскольку первичный ключ должен быть уникальным, для него могут использоваться не все поля таблицы. Если в таблице нет полей, значения которых уникальны, для создания первичного ключа в неё обычно вводят дополнительное числовое поле.

На рисунке 1 в качестве первичного ключа может быть выбрано поле Код сотрудника при условии, что значения в данном поле будут уникальны, не будут пустыми. Поля по отдельности Фамилия, Имя, Отчество, Должность не могут быть выбраны в качестве первичного ключа, так как могут содержать повторяющиеся значения. Совокупность полей Фамилия, Имя, Отчество, Должность так же не могут быть выбраны в качестве первичного ключа, так как могут быть полные однофамильцы на одной и той же должности.

Связи между реляционными таблицами БД соответствуют связям между объектами (сущностями) предметной области.

Объекты (сущности) предметной области **состоят в связи**, если хотя бы одному экземпляру одного из них можно поставить в соответствии (по определенному правилу) один или более экземпляров другого.

Для реляционных таблиц БД: реляционные таблицы БД **состоят в связи**, если хотя бы одной записи одной из них можно поставить в соответствии (по определенному правилу) одну или более записей в другой таблице.

Внешний ключ (англ. foreign key (FK)) – поле или набор полей, предназначенное для организации связи между данными в таблицах БД.

Внешний ключ - это ссылка полей одной таблицы на поля другой таблицы.

Ограничение FOREIGN KEY обычно должно быть связано с ограничением PRIMARY KEY в другой таблице. Кроме того, с помощью этого ограничения могут указываться столбцы ограничения UNIQUE (уникальность) в другой таблице. Ограничение FOREIGN KEY может содержать пустые значения (значения NULL).

Колонка или колонки внешнего ключа одной таблицы ссылаются на первичный ключ (одна или несколько колонок) в другой таблице. При вставке строки в таблицу с ограничением FOREIGN KEY значения, которые должны быть внесены в колонку или колонки, определенные как внешний ключ, сравниваются со значениями в первичном ключе ссылочной (родительской) таблицы. Если ни одна из строк ссылочной таблицы не соответствует значениям во внешнем ключе, то вставка новой строки не выполняется. Но если значения внешнего ключа, которые нужно внести в таблицу, все же имеются в первичном ключе другой таблицы, то вставка новой строки будет выполнена. Если значение, которое должно быть занесено в таблицу с ограничением FOREIGN KEY, равно NULL, то это тоже допустимо.

Проверка ограничений FOREIGN KEY происходит также в тех случаях, когда вы хотите обновить какую-либо строку в ссылочной таблице или в таблице с внешним ключом. Вы не сможете обновить какое-либо значение первичного ключа или внешнего ключа, если это приведет к нарушению ограничения. Существует одно исключение из этого правила, когда вы обновляете данные, используя **каскадное обновление** - при котором изменяется РК родителя и значения FK связанных экземпляров потомка.

Кроме того, ограничения FOREIGN KEY проверяются, если вы хотите удалить какую-либо строку из ссылочной таблицы. Вы не сможете удалить строку из ссылочной (родительской) таблицы, если строка какой-либо таблицы с внешним ключом (таблицы, содержащей ограничение FOREIGN KEY) содержит ссылку на значение в колонке внешнего ключа.

Иными словами, для каждой строки в таблице с внешним ключом должна существовать соответствующая строка в ссылочной (родительской) таблице, и эту строку нельзя удалить, пока на нее имеется ссылка. Существует также исключение из этого правила: вы можете удалить строку из ссылочной (родительской) таблицы с помощью **каскадного удаления** - при котором удаляется экземпляр родителя вместе со связанными экземплярами потомка.

Внешний ключ может ссылаться только на те колонки, которые содержат в ссылочной (родительской) таблице ограничение PRIMARY KEY или UNIQUE. Если вы попытаетесь создать внешний ключ, который ссылается на колонку, не являющуюся частью одного из этих ограничений, то СУБД возвратит сообщение об ошибке. Кроме того, тип данных и размер

колонки или колонок внешнего ключа должны совпадать со ссылочной колонкой или колонками.

Перечисленные особенности отражают правила **ссылочной целостности**.

Для ссылочной целостности характерно:

- первичный ключ (РК) таблицы должен содержать уникальные непустые значения для данной таблицы;
- внешний ключ (FK) потомка должен содержать только те значения, которые уже имеются среди значений PRIMARY KEY родителя (или колонок UNIQUE родителя);
- нельзя удалить экземпляр родителя, имеющий хотя бы один связанный с ним экземпляр потомка. Можно установить механизм **каскадного удаления** при котором удаляется экземпляр родителя вместе со связанными экземплярами потомка;
- нельзя изменить значение в РК родителя, имеющем хотя бы один связанный с ним экземпляр потомка. Можно установить механизм **каскадного обновления** при котором изменяется РК родителя и значения FK связанных экземпляров потомка.

Внешний ключ в таблице может ссылаться и на саму эту таблицу. В таких случаях говорят о **рекурсивном внешнем ключе**. Рекурсивные внешние ключи используются для хранения древовидной структуры данных в реляционной таблице.

Связь характеризуется мощностью связи.

Мощностью связи называется максимальное количество записей одной таблицы БД, связанных с одной записью другой таблицы БД.

Наиболее распространены мощности связи:

- один к одному 1:1;
- один ко многим 1:M;
- многие ко многим M:M.

Связь 1:1: одной записи в одной таблице соответствует одна запись в другой таблице.

Данное отношение используют, если не хотят, чтобы таблица БД «распухала» от второстепенной информации.

Например: **Сотрудники 1 : 1 Дополнительно**

Сотрудники				
Код сотрудника	Фамилия	Имя	Отчество	Должность
1	Петров	Иван	Иванович	Менеджер
2	Сидорова	Ирина	Петровна	Бухгалтер
3	Петров	Иван	Иванович	Менеджер

Дополнительно		
Код сотрудника (FK)	Адрес	Паспортные данные
1	г. Томск	123456 и далее
2	г. Сургут	654321 и далее
3	г. Томск	789123 и далее

Связь 1:M: наиболее распространена. Одной записи в одной таблице (родительской) соответствует несколько записей в другой таблице (дочерней). Запись в дочерней таблице связана с одной записью в родительской таблице.

Например: **Сотрудники 1:M Продажи**

Сотрудники				
Код сотрудника	Фамилия	Имя	Отчество	Должность
1	Петров	Иван	Иванович	Менеджер
2	Сидорова	Ирина	Петровна	Бухгалтер
3	Петров	Иван	Иванович	Менеджер

Продажи		
Номер продажи	Дата	Код сотрудника (FK)
1П	12.02.2008	1
2П	12.02.2008	1
3П	12.02.2008	2
4П	20.02.2008	3
5П	21.02.2008	1

Связь M:M: одной записи первой таблицы может соответствовать несколько записей во второй таблице. И одной записи второй таблицы может соответствовать несколько записей в первой таблице.

Например: **Продажи M:M Блюда**

Продажи

Номер продажи	Дата	Код сотрудника (FK)
1П	12.02.2008	1
2П	12.02.2008	1
3П	12.02.2008	2
4П	20.02.2008	3
5П	21.02.2008	1

Блюда

Код блюда	Название	Тип	Цена меню	Выход
1Б	Оливье	Салаты	100 руб.	250 гр.
2Б	Борщ	Супы	90 руб.	300 гр.
3Б	Цезарь	Салаты	120 руб.	270 гр.

Для моделирования связи М:М в БД вводится дополнительная связующая таблица, которая разбивает связь М:М на связи 1:М и М:1:

Продажи

Номер продажи	Дата	Код сотрудника (FK)
1П	12.02.2008	1
2П	12.02.2008	1
3П	12.02.2008	2
4П	20.02.2008	3
5П	21.02.2008	1

Блюда

Код блюда	Название	Тип	Цена меню	Выход
1Б	Оливье	Салаты	100 руб.	250 гр.
2Б	Борщ	Супы	90 руб.	300 гр.
3Б	Цезарь	Салаты	120 руб.	270 гр.

Продано

Номер продажи (FK)	Код блюда (FK)	Количество	Цена	Скидка
1П	1Б	2	100 руб.	0%
1П	3Б	1	120 руб.	0%
2П	2Б	10	90 руб.	10%
3П	1Б	3	100 руб.	0%
3П	2Б	1	90 руб.	0%
3П	3Б	1	120 руб.	5%

Вторичные ключи устанавливаются по полям, которые часто используются при поиске или сортировке данных: построенные по вторичным ключам индексы помогут системе значительно быстрее найти нужные значения, хранящиеся в соответствующих полях. В отличие от первичных ключей, поля для вторичных ключей могут содержать не уникальные значения.

Например, можно проиндексировать в таблице Блюда поля Тип, Название и др.

Access хранит все объекты - **таблицы, запросы, формы, отчеты, макросы и модули** - в одном файле (с расширением mdb).

Таблицы создаются пользователем для хранения данных. В каждой записи собраны сведения об одном экземпляре определенного объекта.

Запросы создаются пользователем для выборки нужных данных. С помощью запроса можно также обновить, удалить или добавить данные в таблицы или создать новые.

Формы предназначены для редактирования, ввода и просмотра данных в удобном виде; также их можно применять для создания панелей управления в приложении.

Отчеты используются для формирования выходного документа, предназначенного для вывода на печать.

Прежде чем создавать таблицы, необходимо создать базу данных (БД), в которой будут храниться таблицы и другие объекты Access.

Способы создания баз данных в Access

При запуске Access появляется диалоговое окно, в котором можно:

- выбрать БД;
- создать новую.

Для создания БД реализуйте один из способов:

- Выбрать пункт **Базы данных (Запуск мастера)**: автоматическое создание типовой БД.
- Выбрать пункт **Новая база данных**.
- Закрыв диалоговое окно (кн. **Отмена**), можно открыть БД. (**Файл / Открыть БД**) или создать (**Файл / Создать БД**).

Создание таблиц

После точного определения структуры таблицы формируются необходимые **поля** в режиме *Конструктор*. Непосредственное заполнение таблицы осуществляется в режиме *Таблица*. Для ее создания выполняем следующие действия:

- Открываем вкладку **Таблицы**.
- Кн. **Создать**. Появится диалоговое окно **Новая таблица**.
- Выбрать нужный пункт, **ОК**.

Расшифруем эти пункты:

Режим таблицы: на экране появляется таблица, похожая на электронную.

Мастер таблиц: Запускает мастер, который создает таблицу после ответа на все его вопросы.

Импорт таблицы или *Связь с таблицами*: запускает мастер, позволяющий использовать таблицу, хранящуюся в другом формате, как основу для новой таблицы.

Конструктор: позволяет самостоятельно управлять процессом создания таблицы. При этом необходимо:

1. Определить имена полей:

Длина имени не должна превышать 64 символа, может содержать пробелы, но не может содержать . ! [] и не может начинаться с пробела.

2. Выбрать тип данных:

Текстовые поля могут содержать до 255 символов (50 по умолчанию).

Поле МЕМО может содержать до 32 тыс. символов.

Числовые поля (кроме денежных сумм) обычно содержат значения, над которыми впоследствии выполняются вычисления.

Даты/время: над значениями таких полей тоже можно выполнять вычисления. Проверка правильности дат и времени осуществляется автоматически.

Денежный используется для хранения числовых значений денежных сумм.

Счетчик используется для автоматической нумерации добавляемой записи. После добавления записи значения этих полей изменить нельзя.

Логический используется для хранения логических величин, которые могут принимать только два возможных значения: «да» или «нет».

3. Сохранить таблицу:

Файл / Заккрыть или дважды щелкнуть на пиктограмме таблицы в левом верхнем углу окна, или комбинация кн. <Ctrl+F4>, в диалоговом окне подтвердить сохранение.

В следующем диалоговом окне ввести имя таблицы, **ОК**.

Таблица в режиме *Конструктора* всегда содержит три колонки: *имя поля, тип поля и описания поля*. В нижней части окна указываются **Свойства полей**.

Для их определения надо: установить курсор на нужное поле в верхней части окна и перейти в нижнюю часть (F6 или мышью), а затем вручную ввести характеристику этого свойства или выбрать элемент из раскрывающегося списка.

Увеличение размеров области свойств осуществляется нажатием <SHIFT+F2>. Заполнение некоторых свойств можно выполнить с помощью вспомогательного окна построителя (мастера), вызываемого кнопкой **Построить**, расположенной справа от ячейки соответствующего свойства.

Свойства полей

1. Формат поля задает формат представления данных при выводе на экран или печать и может быть задан в формате: *Текстовый, Числовой, Денежный, Счетчик, Дата/Время, Логический*.

Для типов данных *Числовой, Денежный и Счетчик* существует набор форматов:

- *стандартный* – отсутствуют разделители тысяч и знаки валют, число десятичных знаков зависит от точности данных (устанавливается по умолчанию);
- *денежный* – символы валют и два знака после десятичного разделителя;
- *фиксированный* – один знак до и два знака после десятичного разделителя;
- *с разделителями тысяч* – два знака после десятичного разделителя и разделители тысяч;
- *процентный* и
- *экспоненциальный*.

Для типов *Дата/Время* существует набор форматов:

- полный (12.06.96 05:30:10 PM);
- длинный (Среда, 12 июнь 1996);
- средний (12-июн-96);
- краткий (12.06.96);
- длинный формат времени (11:30:10 PM);
- средний формат времени (11:30 PM);
- краткий формат времени (13:30).

Константы типа *Дата/Время* ограничиваются символами "#".
Например, #03.09.02#.

Для *Логического типа*: Да/Нет (по умолчанию);
Истина/Ложь; Вкл/Выкл.

Число десятичных знаков задает число знаков после разделителя (от 0 до 15).

2. Размер поля задает максимальный размер данных в поле. Поле с текстовым типом может иметь размер от 1 до 255 символов (по умолчанию — 50).

Размер поля для числового типа данных:

Тип	Размер
Байт - (1 байт)	Целые числа от 0 до 255
Целое (2 байта)	Целые числа от -32768 до +32767
Длинное целое - (4 байта)	Целые числа от -2 147 483 648 до +2 147 483

	647
С плавающей точкой (4 байта)	С точностью до 6 знаков от - 3.4×10^{38} до $+3.4 \times 10^{38}$
С плавающей точкой (8 байтов)	С точностью до 10 знаков от 1.797×10^{308} до $+1.797 \times 10^{308}$

3. **Маску ввода** можно ввести вручную или использовать *Мастер масок* при помощи кнопки *Построить*. В окне построителя из списка готовых масок следует выбрать подходящую и заполнить пустые позиции.
4. **Подпись поля** определяет подпись для использования в формах и отчетах, если она отличается от имени поля.
5. **Значение по умолчанию** автоматически добавляется во все новые записи.
6. **Условие на значение** задаётся выражением, состоящим из операторов сравнения и значений, используемых для сравнения (операндов).
7. **Сообщение об ошибке** позволяет задать текст, выводимый на экран, если значение не удовлетворяет условию на значение.
8. **Обязательное поле** определяет необходимость заполнения при вводе данных.
9. Свойство **Пустые строки** позволяет хранить пустые строки в текстовых и МЕМО полях.
10. **Индексированное поле** задает построение индекса для полей с типом данных *Текстовый*, *Числовой*, *Денежный*, *Дата/Время* и *Счетчик*. Назначение индекса - ускорение выполнения запросов, поиска и сортировки. Возможные значения этого свойства:
 - **Нет** – не создает индекс или удаляет существующий индекс;
 - **Да (допускаются совпадения)** – создает индекс, если в нем допускаются совпадения значений для разных записей;
 - **Да (совпадения не допускаются)** – создает уникальный индекс.

Алгоритм установления связей:

Выполнить команду **Сервис/Схема данных**.

- Появится диалоговое окно с таблицами без связей или
- если связи устанавливаются впервые, то появится диалоговое окно **Добавление таблицы**. Выбрать таблицу, которая используется при установке связи, щелкнуть по кнопке **Добавить**. Повторить эти действия для каждой добавляемой таблицы и затем кн. **Заккрыть**.
- Для создания связей между таблицами переместить поле, которое нужно связать, из исходной (главной) таблицы или запроса в соответствующее поле другой (подчинённой) таблицы или запроса (как правило, связываются ключевые поля обеих таблиц). Следует учесть, что связываемые поля должны иметь один тип (кроме поля счетчика,

которое можно связывать с числовым полем). Появится диалоговое окно **Связи**.

- Необходимо проверить, стоит ли флажок у опции **Целостность данных** и тогда линия связи станет темнее и появятся значки "1" и "∞", означающий отношение один-ко-многим.
- Целостность обеспечивает контроль правильности ссылок и блокирует выполнение ошибочных операций.

Если связь определена и система взяла на себя поддержку целостности данных, то при просмотре главной таблицы (отношение «один») слева появится колонка со знаком «+». По щелчку на знаке «+» можно открыть подчинённую таблицу.

Практические задания

Цель – создайте базу данных для своей предметной области.

Требования к базе данных (БД):

- БД должна быть реляционной;
- БД должна быть нормализована;
- БД должна состоять минимум из 3-х связанных таблиц (на каждого студента);
- в таблицах БД должны быть наложены ограничения на поля таблиц (по усмотрению разработчиков на какие поля будут наложены ограничения). Должно быть не менее 3-х ограничений на разные поля таблиц (без учета ограничений PRIMARY KEY, FOREIGN KEY);
- должна быть обеспечена ссылочная целостность.

Ниже приведен пример создания БД для кафе в СУБД Microsoft Access.

Пример:

Цель – создание базы данных для автоматизации деятельности кафе.

Создайте отдельную папку на диске для Вашей дальнейшей программы. Например, D:\Temp\Petrov\Proga_Kaffe.

Запустите программу Microsoft Access.

Нажмите меню Файл Создать База данных. В диалоговом окне Сохраните базу данных с именем Кафе.mdb в созданную Вами папку (например, D:\Temp\Petrov\Proga_Kaffe).

Дальнейшие задания выполняются в созданной Вами базе данных Кафе.mdb.

Задание 1

Создайте проект из таблиц: «Блюда», «Сотрудники», «Заказы», «Заказано» со следующими макетами:

Таблица «Блюда»

А. Используя вкладку *Таблицы* создать в режиме конструктора поля, типы данных и свойства полей, по образцу:

Имя поля	Тип данных поля	Описание
Код_блюда	Счетчик	Код блюда в кафе. Первичный ключ. Уникальный идентификатор блюда
Название	Текстовый	Название блюда
Тип	Текстовый	Тип блюда (салаты, десерты, супы и др.)
Цена_меню	Денежный	Цена блюда по меню
Выход	Числовой	Количество готового блюда в граммах
Состав	Поле МЕМО	Описание состава блюда
Фото	Поле объекта OLE	Фотография готового блюда

В. Задайте характеристики полей:

Название поля	Код_блюда
Первичный ключ	Код_блюда
Размер поля	Длинное целое
Новые значения	Последовательные
Индексированное поле	Да (Совпадения не допускаются)

Название поля	Название
Размер поля	40
Обязательное поле	Да
Пустые строки	Нет
Индексированное поле	Да (Допускаются совпадения)

Название поля	Тип
Размер поля	30
Обязательное поле	Да
Пустые строки	Нет
Индексированное поле	Да (Допускаются совпадения)

Название поля	Цена_меню
Формат поля	Денежный
Число десятичных знаков	Авто
Значения по умолчанию	0

Обязательное поле	Да
Индексированное поле	Да (Допускаются совпадения)

Название поля	Выход
Размер поля	Длинное целое
Число десятичных знаков	Авто
Значения по умолчанию	0
Обязательное поле	Нет
Индексированное поле	Нет

Название поля	Состав
Обязательное поле	Нет
Пустые строки	Да

Название поля	Фото
Обязательное поле	Нет

С. В качестве первичного ключа используйте поле **Код_блюда**, пометив ключом через соответствующую команду контекстно-зависимого меню (правая кнопка на имени выделенной строки поля в режиме *Конструктор*).

Д. Сохраните таблицу с именем **Блюда**.

Таблица «Сотрудники»

А. Используя вкладку *Таблицы* создать в режиме конструктора поля, типы данных и свойства полей, по образцу:

Имя поля	Тип данных поля	Описание
Код_сотрудника	Счетчик	Уникальный идентификатор сотрудника. Первичный ключ.
Фамилия	Текстовый	Фамилия сотрудника
Имя	Текстовый	Имя сотрудника
Отчество	Текстовый	Отчество сотрудника
Должность	Текстовый	Должность сотрудника
Пол	Текстовый	Пол сотрудника
Семейное_положение	Текстовый	Семейное положение сотрудника
Дети	Текстовый	Имеются ли дети у сотрудника

В. Задайте характеристики полей:

Название поля	Код_сотрудника
Первичный ключ	Код_сотрудника
Размер поля	Длинное целое

Новые значения
Индексированное поле

Последовательные
Да (Совпадения не допускаются)

Название поля
Размер поля
Обязательное поле
Пустые строки
Индексированное поле

Фамилия
30
Да
Нет
Да (Допускаются совпадения)

Название поля
Размер поля
Обязательное поле
Пустые строки
Индексированное поле

Имя
30
Да
Нет
Да (Допускаются совпадения)

Название поля
Размер поля
Обязательное поле
Пустые строки
Индексированное поле

Отчество
30
Нет
Да
Нет

Название поля
Размер поля
Обязательное поле
Пустые строки
Индексированное поле

Должность
30
Да
Нет
Да (Допускаются совпадения)

Название поля
Размер поля
Условие на значение
Сообщение об ошибке

Обязательное поле
Пустые строки
Индексированное поле

Пол
10
"мужской" Or "женский"
Допустимы значения "мужской" или "женский"
Нет
Да
Нет

Название поля
Размер поля
Условие на значение

Сообщение об ошибке

Семейное_положение
20
"Женат" Or "Замужем" Or "Холост"
Or "Не замужем"
Допустимы значения Женат или

	Замужем или Холост или Не замужем
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Нет
Название поля	Дети
Размер поля	15
Условие на значение	"Есть дети" Or "Детей нет"
Сообщение об ошибке	Допустимы значения Есть дети или Детей нет
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Нет

С. В качестве первичного ключа используйте поле **Код_сотрудника**, пометив ключом через соответствующую команду контекстно-зависимого меню (правая кнопка на имени выделенной строки поля в режиме *Конструктор*).

Д. Сохраните таблицу с именем **Сотрудники**.

Таблица «Продажи»

А. Используя вкладку *Таблицы* создать в режиме конструктора поля, типы данных и свойства полей, по образцу:

Имя поля	Тип данных поля	Описание
Номер_продажи	Счетчик	Уникальный идентификатор продаж. Первичный ключ.
Дата	Дата/время	Дата продажи
Код_сотрудника	Числовой	Идентификатор сотрудника, который оформил продажу. Ссылается на таблицу Сотрудники. Внешний ключ.
Номер_столика	Текстовый	Номер столика в кафе

В. Задайте характеристики полей:

Название поля	Номер_продажи
Первичный ключ	Номер_продажи
Размер поля	Длинное целое
Новые значения	Последовательные
Индексированное поле	Да (Совпадения не допускаются)

Название поля	Дата
Формат поля	Краткий формат даты
Обязательное поле	Нет
Индексированное поле	Нет
Название поле	Код_сотрудника
Размер поля	Длинное целое
Число десятичных знаков	Авто
Обязательное поле	Да
Индексированное поле	Да (Допускаются совпадения)
Название поля	Номер_столика
Размер поля	10
Обязательное поле	Нет
Пустые строки	Да
Индексированное поле	Нет

С. В качестве первичного ключа используйте поле **Номер_продажи**, пометив ключом через соответствующую команду контекстно-зависимого меню (правая кнопка на имени выделенной строки поля в режиме *Конструктор*).

D. Сохраните таблицу с именем **Продажи**.

Таблица «Продано»

A. Используя вкладку *Таблицы* создать в режиме конструктора поля, типы данных и свойства полей, по образцу:

Имя поля	Тип данных поля	Описание
Номер_продажи	Числовой	Номер продажи. Ссылается на таблицу Продажи. Внешний ключ. Входит в первичный ключ.
Код_блюда	Числовой	Код блюда. Ссылается на таблицу Блюда. Внешний ключ. Входит в первичный ключ.
Количество	Числовой	Количество проданных блюд
Цена	Денежный	Цена блюда на момент продажи
Скидка	Числовой	Скидка на конкретное блюдо в продажи

B. Задайте характеристики полей:

Название поля	Номер_продажи
Размер поля	Длинное целое
Число десятичных знаков	Авто
Обязательное поле	Да
Индексированное поле	Да (Допускаются совпадения)

Название поля	Код_блюда
Размер поля	Длинное целое
Число десятичных знаков	Авто
Обязательное поле	Да
Индексированное поле	Да (Допускаются совпадения)

Название поля	Количество
Размер поля	Длинное целое
Число десятичных знаков	Авто
Значения по умолчанию	1
Обязательное поле	Да
Индексированное поле	Да (Допускаются совпадения)

Название поля	Цена
Формат поля	Денежный
Число десятичных знаков	Авто
Обязательное поле	Да
Индексированное поле	Да (Допускаются совпадения)

Название поля	Скидка
Размер поля	Одинарное с плавающей точкой
Формат поля	Процентный
Число десятичных знаков	Авто
Значения по умолчанию	0
Условие на значение	≥ 0 And ≤ 1
Сообщение об ошибке	Не соответствует диапазону скидок от 0 до 1
Обязательное поле	Нет
Индексированное поле	Нет

С. В качестве первичного ключа используйте поля **Номер_продажи** и **Код_блюда**, пометив ключом через соответствующую команду контекстно-зависимого меню (правая кнопка на имени выделенной строки поля в режиме *Конструктор*).

Д. Сохраните таблицу с именем **Продано**.

Задание 2

Схема связи

Ход работы:

1. В открытой базе данных Кафе.mdb выполните команду, **Сервис/Схема данных** добавьте сформированные таблицы Сотрудники, Блюда, Продажи, Продано.
2. Свяжите ключевое поле Код_сотрудника простым перетаскиванием названия поля от таблицы Сотрудники к таблице Продажи.
3. Определите параметр связи **Обеспечение целостности данных** (по умолчанию установится отношение один-ко-многим) и кнопка **Создать**.
4. Свяжите ключевое поле Код_блюда простым перетаскиванием названия поля от таблицы Блюда к таблице Продано.
5. Определите параметр связи **Обеспечение целостности данных** (по умолчанию установится отношение один-ко-многим) и кнопка **Создать**.
6. Свяжите ключевое поле Номер_продажи простым перетаскиванием названия поля от таблицы Продажи к таблице Продано.
7. Определите параметр связи **Обеспечение целостности данных** (по умолчанию установится отношение один-ко-многим) и кнопка **Создать**.
8. Закройте ДО **Схема данных**.
9. Убедитесь, что в главных таблицах Блюда, Сотрудники, Продажи появился столбец с «+», позволяющий просмотреть подчинённые записи.

Задание 3

Заполните по 3-5 строк в каждой созданной таблице. Отследите обеспечение целостности.

Задание 4

Оформить отчёт со следующим содержанием:

1. Титульный лист.
2. Цель работы.
3. Постановка задачи.
4. Краткая теория и ход выполнения заданий.
5. Описание результатов.
Приведите скриншоты полученной схемы данных, конструкторов таблиц, режимов таблиц с заполненными данными. Опишите наложенные ограничения на Вашу БД. Покажите отработку ограничений целостности в Вашей БД.
6. Заключение (выводы).