

Проектирование базы данных некоторой предметной области. Нормализация отношений. Создание модели в MS Visio.

Цель работы:

- получить навыки нормализации базы данных некоторой предметной области;
- привести таблицы к третьей нормальной форме;
- создать концептуальную модель средствами языка UML в MS Visio.

Теоретические положения

Процесс преобразования отношений базы данных (БД) к виду, отвечающему нормальным формам, называется **нормализацией**. **Цель нормализации базы данных** - устранение избыточности и дублирования информации. Выделяются **шесть** нормальных форм, пять из которых так и называются: **первая, вторая, третья, четвертая, пятая нормальная форма**, а также нормальная **форма Бойса-Кодда**, лежащая между третьей и четвертой.

Процесс проектирования БД с использованием **метода нормальных форм** является **итерационным** и заключается в последовательном переводе отношений (таблиц) из **первой нормальной формы** в нормальные **формы более высокого порядка по определённым правилам**. База данных считается нормализованной, если ее таблицы (по крайней мере, большинство таблиц) представлены как минимум в третьей нормальной форме.

Первая нормальная форма: таблица соответствует *первой нормальной форме*, если все значения ее полей **атомарны** (простые), **и все записи — уникальны**.

Аномалии данных

Избыточное дублирование данных создает проблемы при обработке кортежей отношения, названное "**аномалиями обновления отношения**".

Выделяют три основных вида аномалий:

- 1) Удаления
- 2) Модификации (редактирования)
- 3) Добавления

Аномалии модификации – изменение значения одного данного может повлечь за собой просмотр всей таблицы и соответствующие изменения других записей таблицы.

Аномалии удаления – удаление какого-либо данного из таблицы может привести к потере информации, которая не связана напрямую с удаляемым данным.

Аномалии добавления – нельзя добавить в таблицу запись, содержащую пустые поля или если добавление записи требует дополнительного просмотра таблицы.

Правила нормализации таблиц

Правило 1: уникальность полей- каждое поле таблицы должно представлять уникальный тип информации

Правило 2: первичные ключи - каждая таблица должна иметь уникальный идентификатор (первичный ключ), который может состоять из одного или нескольких полей

Вторая нормальная форма: Таблица находится во второй нормальной форме, если она находится в первой нормальной форме и **её неключевые поля полностью зависят от всего первичного ключа**.

Правило 3: функциональная зависимость - для каждого значения первичного ключа значения полей (атрибутов) должны относиться к субъекту таблицы и полностью его описывать

Для перехода от первой нормальной формы ко второй, необходимо выполнить следующие действия:

1. Определить, на какие части можно разбить первичный ключ, так чтобы некоторые из неключевых полей зависели от одной из этих частей (*эти части не обязаны состоять из одного поля*).
2. Создать новую таблицу для каждой такой части ключа и зависящих от нее полей, и переместить их в эту таблицу. Часть бывшего первичного ключа станет первичным ключом новой таблицы.
3. Удалить из исходной таблицы поля, перемещенные в другие таблицы, кроме тех, которые станут внешними ключами.

Примечание: Если таблица находится в первой нормальной форме и первичный ключ у неё атомарный, то она автоматически находится и во второй нормальной форме.

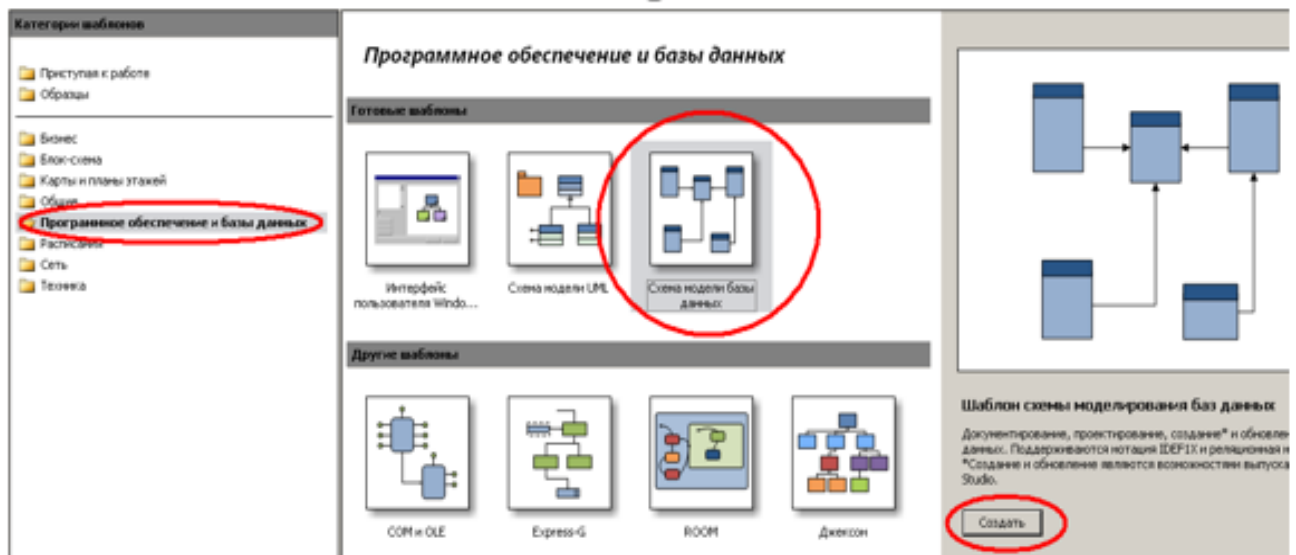
Третья нормальная форма: Таблица находится в третьей нормальной форме, если она находится во второй нормальной форме и **все неключевые поля взаимно независимы и зависят только от первичного ключа**

Правило 4: независимость полей - изменение значения любого поля, не входящего в первичный ключ, не должно влиять на данные других полей.

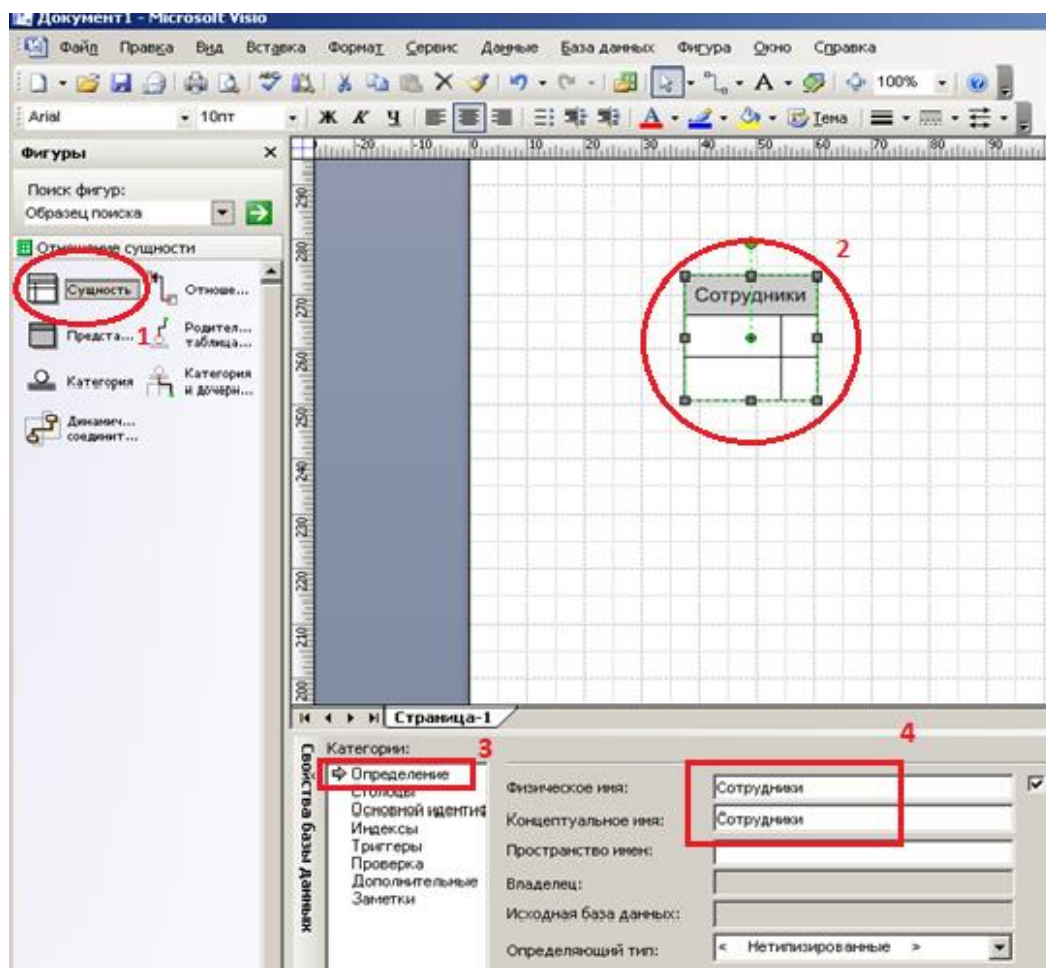
Примечание: Самая распространенная ситуация в данном контексте - это расчетные столбцы, значения которых можно получить путем каких-либо манипуляций с другими столбцами таблицы. Для приведения таблицы в третью нормальную форму такие столбцы из таблиц надо удалить.

Порядок выполнения работы:

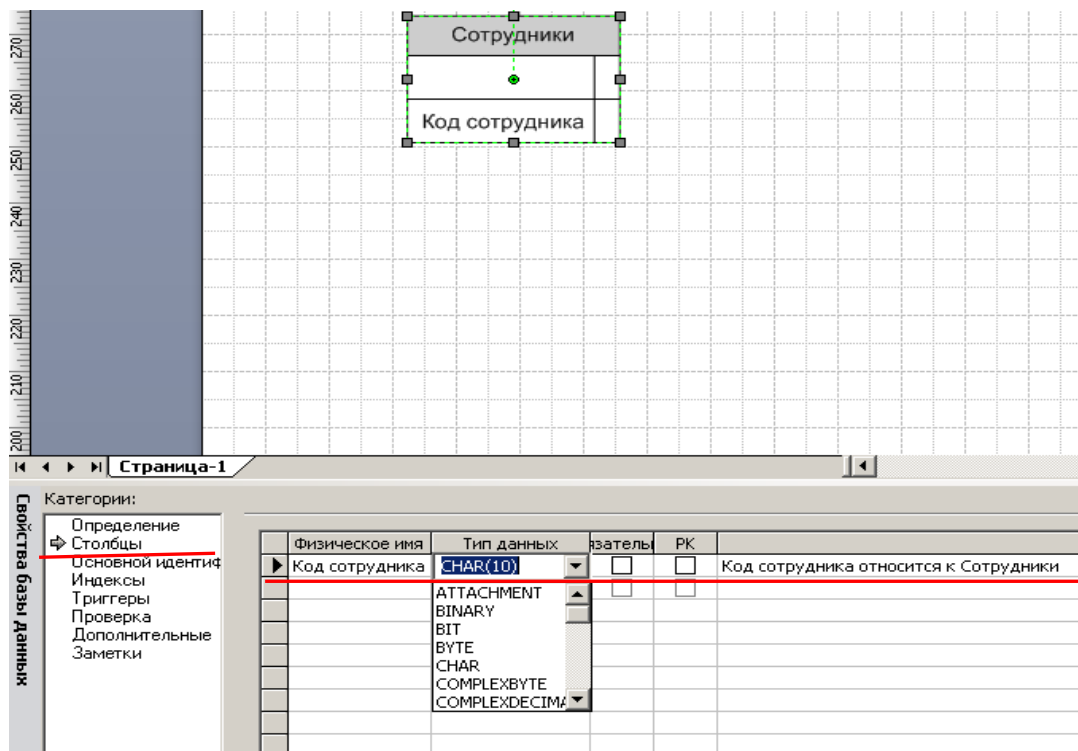
- 1) Ознакомьтесь с предложенной предметной областью.
- 2) Создайте таблицу, занесите в неё все необходимые данные, которые должна хранить база данных. (данные в таблицы должны содержать аномалии)
- 3) Постепенно переводите таблицы от первой нормальной формы к третьей, исключая аномалии.
- 4) Переведя таблицы к третьей нормальной форме средствами MS Visio создайте концептуальную модель.
 - Запустите MS Visio.
 - Выберите → Категории шаблонов → Программное обеспечение и базы данных → готовые шаблоны Схема модели базы данных → кнопка Создать



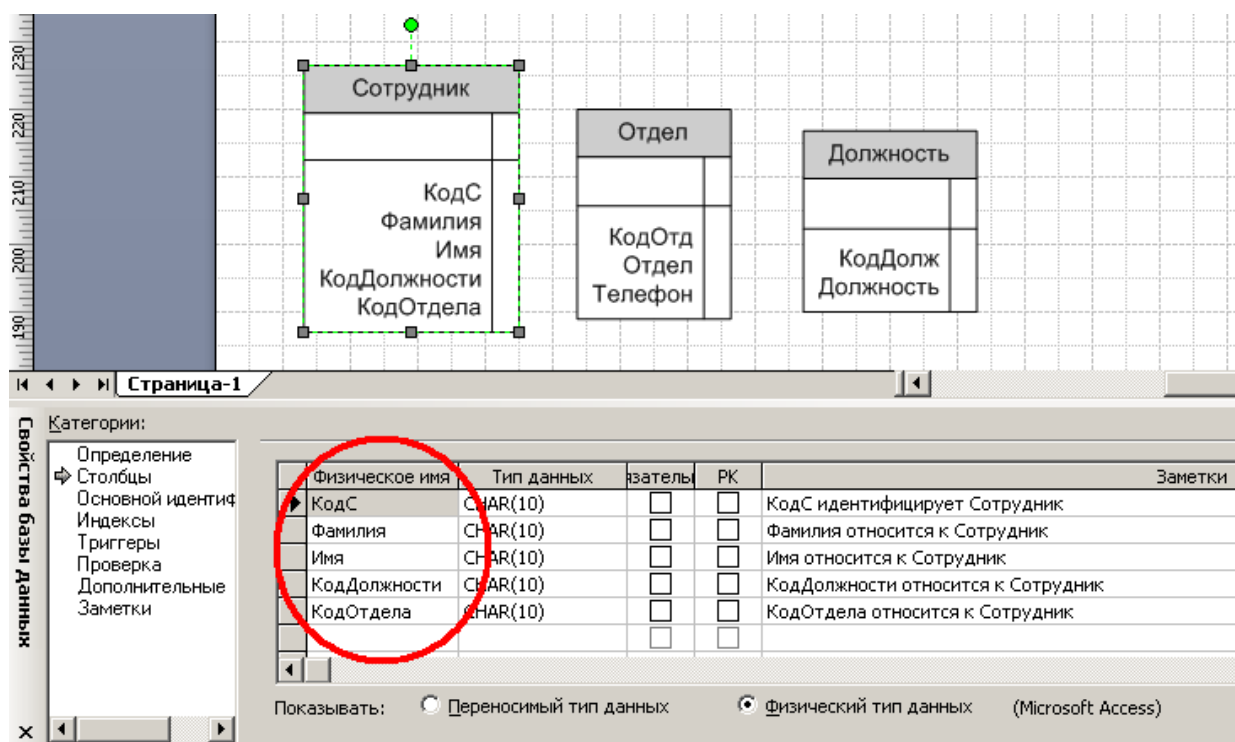
- Для создания новой таблицы во вкладке **Фигуры** → выбрать **Сущность** → зажав ЛКМ перетащите таблицу на рабочий лист.
- В категории выберите **Определение** → задайте таблице **Имя**



- Формируем столбцы таблицы: **Имя столбца** → **Тип данных** (из списка) → **Обязательное** → **Первичный ключ (РК)** → **Заметки** (пояснение, какая информация содержится в поле)

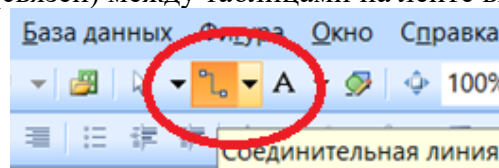


Введите информацию о всех полях формируемой таблицы.



Аналогично, создайте остальные таблицы.

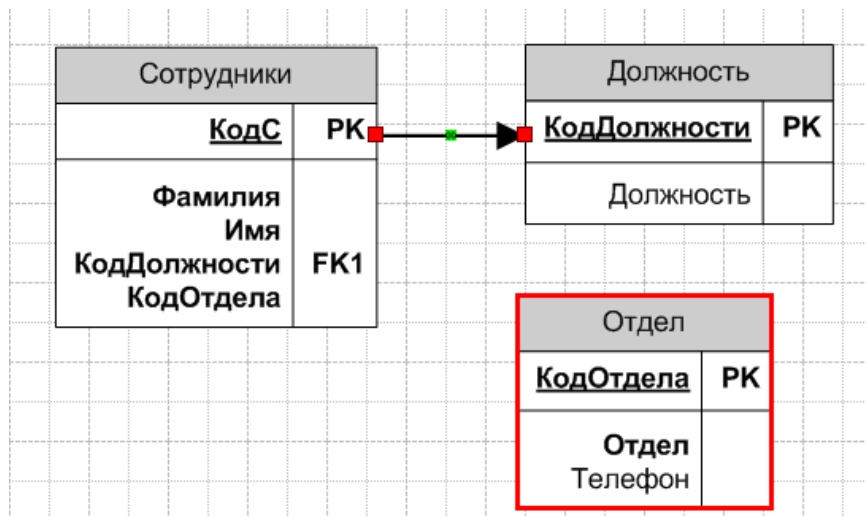
Для создания отношений (связей) между таблицами на ленте выберите



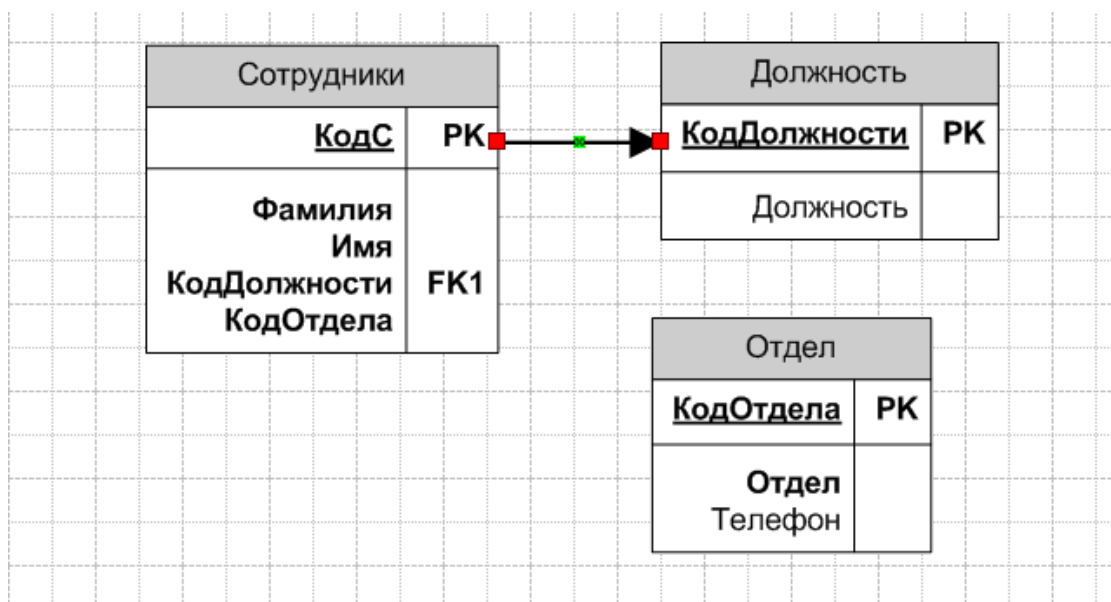
Соединительная линия

→ перетащите её в

центр родительской таблицы до появления красного контура. Зажать левую клавишу мыши и удерживая перетащите в центр дочерней (подчиненной) таблицы, а затем отпустить



В подчиненной таблице появится внешний ключ (FK)

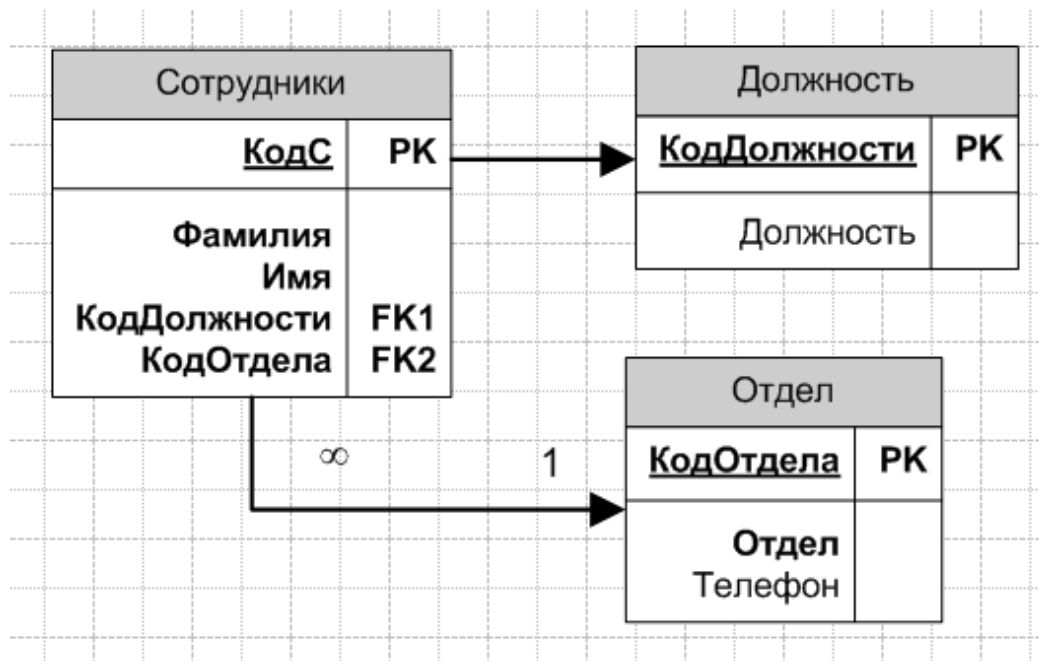


– Теперь укажем типы связей между таблицами:

а) вкладка **Вставка** → **Рисунок** → **Формула** → выбираем символ ∞.



б) на панели инструментов ЭУ **Текст** установите рядом с созданной связью и поставьте 1 и ∞.



– Для удаления таблицы, столбца, связи выделяем этот объект и нажимаем кнопку **Удалить**.

Содержание отчета

1. Название работы
2. Цель работы
3. Описание предметной области
4. Таблица со всеми необходимыми данными (начальная)
5. Таблицы с данными после приведения к 3-ей НФ
6. Концептуальная модель, созданная в MS Visio.