**Занятие № 8**

**Номер учебной группы:** П-16

**Фамилия, инициалы учащегося:** Хомич В.И.

**Дата выполнения работы:** 16.11.2021.

**Тема работы:** «Разработка концептуальной модели базы данных»

**Цель работы:** Создание модели «сущность-связь» в нотации Баркера с использованием современных CASE технологий.

**Ход работы**

**Задание 1**

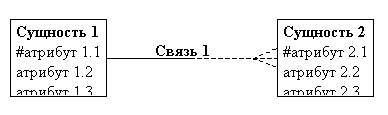
Изучить теоретический материал по работе в программе Dia для создания декомпозиции функциональных диаграмм. Описать в отчете работу с программой. Описать элементы используемые при создании модели сущность-связь в нотации Баркера.

**Ответ:** Dia – бесплатный редактор для создания диаграмм и схем. Коммерческим аналогом этой программы является продукт компании Microsoft – Visio. С помощью Dia возможно создание многих видов структурированных диаграмм и схем, в том числе: - блок-схемы; - диаграммы UML; - сетевые диаграммы; - ER-диаграммы (проектирование баз данных); - упрощенные схемы электрических цепей и другие. В программе поддерживается множество языков и региональных стандартов, среди прочих есть русский и украинский. Dia позволяет экспортировать данные в более чем 25 форматов векторной и растровой графики, в том числе векторные SVG, DXF, FIG, VDX (MS Visio), PDF и растровые рисунки BMP, GIF, JPG, PNG, TIF. «Родной» формат программы - Dia Native Diagram (DIA). Интерфейс у Dia простой, создать диаграмму достаточно просто даже пользователю, впервые работающему с программой. До версии 0.97 панель инструментов и рабочая область располагаются в отдельных окнах. Это так называемый однодокументный интерфейс (CSDI). При работе с несколькими файлами для каждого открывается отдельное окно, а панель инструментов в этом случае постоянно находится поверх остальных окон. Поначалу это непривычно, но в процессе использования оказывается вполне удобно. Dia предоставляет на выбор пользователя большой набор геометрических фигур, библиотеку клипартов, электрические схемы, пиктограммы по компьютерным сетям Cisco, а также кибернетические, гидравлические, логические и многие другие символы. Среди доступных возможностей можно выделить рисование кривых Безье, поддержку слоев, поиск элементов схемы, введение новых символов, определяемых в XML-файлах с помощью подмножества тегов SVG для изображения фигур, загрузка и сохранение диаграммы в своем XML-формате.

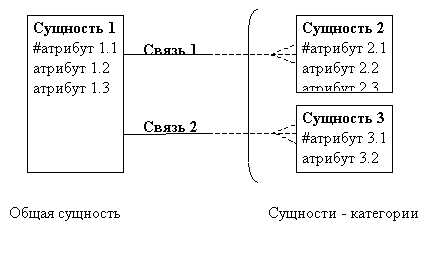
**Нотация Баркера**. Сущности обозначаются прямоугольниками, внутри которых приводится список атрибутов. Ключевые атрибуты отмечаются символом # (решетка). Связи обозначаются линиями с именами, место соединения связи и сущности определяет кардинальность связи:

|  |  |
| --- | --- |
| **Обозначение** | **Кардинальность** |
| http://www.mstu.edu.ru/study/materials/zelenkov/image160.gif | 0,1 |
| http://www.mstu.edu.ru/study/materials/zelenkov/image148.gif | 1,1 |
| http://www.mstu.edu.ru/study/materials/zelenkov/image161.gif | 0,N |
| http://www.mstu.edu.ru/study/materials/zelenkov/image162.gif | 1,N |

Пример:

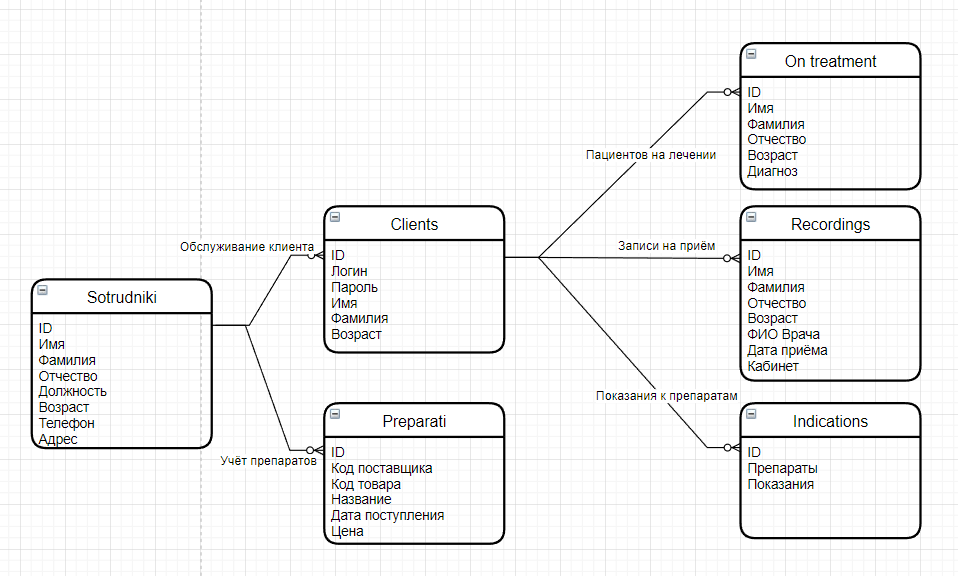


Для обозначения отношения категоризации вводится элемент "дуга":



**Задание 2**

На основании созданной информационной модели (занятие 7) создал модель «сущность-связь» в нотации Баркера с использованием конструктора Dia.

****

**Задание 3**

Ответил на контрольные вопросы:

1. Case-средства. ERwin. Его функции и задачи.

Процесс построения информационной модели состоит из следующих шагов:

* определение сущностей;
* определение зависимостей между сущностями;
* задание первичных и альтернативных ключей;
* определение атрибутов сущностей;
* приведение модели к требуемому уровню нормальной формы;
* переход к физическому описанию модели:назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы; задание триггеров, процедур и ограничений;
* генерация базы данных.

ERwin создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения как части документации, необходимой в цикле разработки. Однако ERwin далеко не только инструмент для рисования. ERwin автоматически создает базу данных (таблицы, индексы, хранимые процедуры, триггеры для обеспечения ссылочной целостности и другие объекты, необходимые для управления данными).

1. Методология IDEF1X.

IDEF1X является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы. Концептуальной схемой мы называем универсальное представление структуры данных в рамках коммерческого предприятия, независимое от конечной реализации базы данных и аппаратной платформы. Будучи статическим методом разработки, IDEF1X изначально не предназначен для динамического анализа по принципу "AS IS", тем не менее, он иногда применяется в этом качестве, как альтернатива методу IDEF1. Использование метода IDEF1X наиболее целесообразно для построения логической структуры базы данных после того, как все информационные ресурсы исследованы (скажем с помощью метода IDEF1) и решение о внедрении реляционной базы данных, как части корпоративной информационной системы, было принято. Однако не стоит забывать, что средства моделирования IDEF1X специально разработаны для построения реляционных информационных систем, и если существует необходимость проектирования другой системы, скажем объектно-ориентированной, то лучше избрать другие методы моделирования.

1. Идентификация сущностей. Сущности в ERwin.

На диаграмме сущность изображается прямоугольником. В зависимости от режима представления диаграммы прямоугольник может содержать имя сущности, ее описание, список ее атрибутов и другие сведения. Для внесения сущности в модель необходимо (убедившись предварительно, что Вы находитесь на уровне логической модели - переключателем между логической и физической моделью служит раскрывающийся список в правой части панели инструментов) нажать на кнопку сущности на панели инструментов (ERwin Toolbox), затем нажать на то место на диаграмме, где Вы хотите расположить новую сущность. Нажав правую кнопку мыши по сущности и выбрав из всплывающего меню пункт Entity Editor. можно вызвать диалог Entity Editor, в котором определяются имя, описание и комментарии сущности.

1. Связи в ERwin. Классификация связей.

На ERwin Toolbox есть пять типов связей: идентифицирующая, неидентифицирующая, полный кластер подтипов, неполный кластер подтипов, неопределенная (связь”многие-ко-многим”). Идентифицирующая связь - такая связь, при которой экземпляр дочерней сущности идентифицируется через свою ассоциацию с родительской сущностью.

1. Атрибуты в ERwin. Классификация атрибутов

Entity Editor в контекстном меню для сущности позволяет определить имя, описание, комментарии, иконку. Для описания атрибутов сущности выбирается пункт Attribute Editor. Здесь можно указать имя нового атрибута и домен, который будет использоваться при определении типа колонки на уровне физической модели. Атрибуты должны именоваться в единственном числе, иметь четкое смысловое значение и быть достаточно важными для того, чтобы их моделировать. Именование сущности в единственном числе облегчает в дальнейшем чтение модели. Каждый атрибут должен быть определен (закладка Definition), при этом следует избегать циклических определений и производных атрибутов. Для внесения дополнительных комментариев и определений к сущности служат свойства, определенные пользователем (UDP). Соблюдение этого правила позволяет частично решить проблему нормализации данных уже на этапе определения атрибутов.