*Занятие № 8*

*Номер учебной группы:* П-16

*Фамилия, инициалы учащегося:* Язубец Е.В.

*Дата выполнения работы:* 16.11.2022

*Тема работы:* «Разработка модели «сущность-связь» в нотации Баркера с использованием современных CASE технологий»

*Результат выполнения работы*

**1.Описание работы с программой Dia.**

**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 панельинструментов и рабочая область располагаются в отдельных окнах. Это такназываемый однодокументный интерфейс (CSDI). При работе с несколькимифайлами для каждого открывается отдельное окно, а панель инструментов вэтом случае постоянно находится поверх остальных окон.

Поначалу этонепривычно, 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.

**2.Нотация Баркера.**

Сущность в нотации Баркера представляется прямоугольником любого размера, содержащим внутри себя:

* имя сущности;
* список имен атрибутов (знак oперед именем атрибута);

указатели ключевых атрибутов (знак #перед именем атрибута);

указатели обязательных атрибутов (знак \*перед именем атрибута).

Все связи являются бинарными и представляются линиями с двумя концами (соединяющими сущности), для которых должно быть определено:

* имя связи;
* степень множественности (один или много объектов участвуют в связи: « 0,1» , « 0,\*» , «1» , «1,\*» ;

степень обязательности: обязательная (Mandatory) или необязательная (Optional) связь между сущностями.

Если максимальное кардинальное число отображения равно бесконечности, то линия, представляющая связь, разветвляется, принимая вид «вороньей лапки» и примыкает к прямоугольнику, соответствующему множеству сущностей – области значений, в трех точках; в противном случае линия остается без изменений.

Если минимальное кардинальное число отображения равно нулю, то часть линии-связи, примыкающая к прямоугольнику, изображается при помощи пунктирной линии, в противном случае используется сплошная линия.

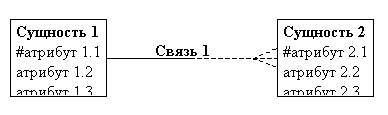
Нотация применяется к обоим отображениям, определяемым связью, при этом линия, изображающая связь, делится пополам, и каждая ее часть оформляется в соответствии со значениями кардинальных чисел.

Для указания степени множественности равной 0 в свойствах связи должна быть включена Target optional.

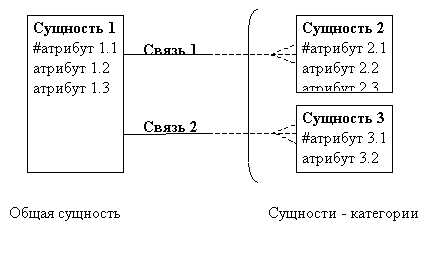
Сущности обозначаются прямоугольниками, внутри которых приводится список атрибутов. Ключевые атрибуты отмечаются символом # (решетка). Связи обозначаются линиями с именами, место соединения связи и сущности определяет кардинальность связи:

|  |  |
| --- | --- |
| **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 |

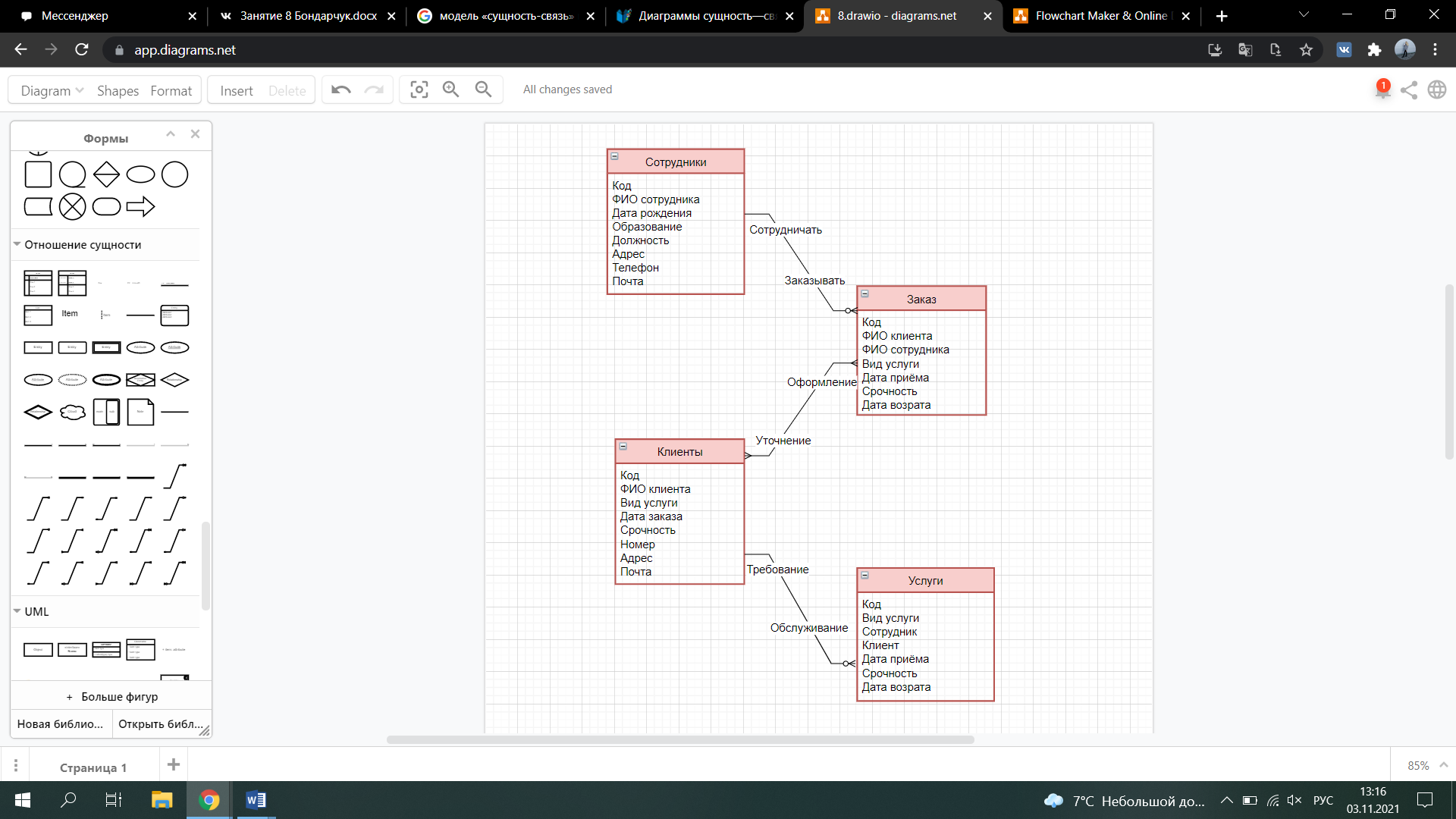
Пример:



Для обозначения отношения категоризации вводится элемент "дуга":



**3.Создала модель «сущность-связь» в нотации Баркера с использованием конструктора Dia.**



**4.Ответы на контрольные вопросы.**

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

**Ответ:** **CASE-средство** - любое программное средство, автоматизирующее ту или иную совокупность процессов жизненного цикла ПО и обладающее следующими особенностями:

· мощные графические средства для описания и документирования ИС, обеспечивающие удобный интерфейс;

· интеграция отдельных компонент CASE-средств, обеспечивающая управляемость процессом разработки ИС;

· использование специальным образом организованного хранилища проектных метаданных (репозитория).

Наиболее трудоемкими этапами разработки ИС являются этапы анализа и проектирования, в процессе которых CASE-средства обеспечивают качество принимаемых технических решений и подготовку проектной документации. При этом большую роль играют методы визуального представления информации

**CASE-средства можно классифицировать по следующим признакам**:

* применяемым методологиям и моделям систем и БД;
* степени интегрированности с СУБД;
* доступным платформам.

**ERwin** - средство разработки структуры базы данных (БД). ERwin сочетает графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логического и физического описания модели данных и прозрачную поддержку ведущих реляционных СУБД и настольных баз данных. С помощью ERwin можно создавать или проводить обратное проектирование (реинжиниринг) баз данных.

Возможны **две точки зрения на информационную** модель и, соответственно, два уровня модели. Первый - **логический** (точка зрения пользователя) - описывает данные, задействованные в бизнесе предприятия. Второй - **физический** - определяет представление информации в БД. ERwin объединяет их в единую диаграмму, имеющую несколько уровней представления.

1. Методология IDEF1X.

**Ответ:** Методология моделирования IDEF1X, являясь расширением стандарта IDEF1, предназначена для описания данных (информации). В ее основе лежит язык семантического моделирования, основанного на концепции "сущность — связь", позволяющей определять данные и связи между ними. Методология используется для создания информационной модели предметной области с помощью идентификации ее сущностей и связей между ними. Чаще всего такая методология используется для описания данных в целях последующей автоматизации их обработки с помощью систем управления базами данных. Таким образом, можно говорить о том, что модели данных в нотации IDEF1X используются для создания баз данных.

Основными элементами модели IDEF1X являются сущности, атрибуты и отношения.

1. Идентификация сущностей. Сущности в ERwin.

**Ответ:** 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.

Сущность представляет собой множество реальных или абстрактных объектов, например: люди, места, события, факты, которые имеют общие характеристики. Сущность - это логическое понятие. Сущности соответствует таблица в реальной СУБД. В ERwin сущность визуально представляет три основных вида информации:

1) атрибуты, составляющие первичный ключ;

2) неключевые атрибуты;

3) тип сущности (независимая/зависимая).

1. Связи в ERwin. Классификация связей.

**Ответ:** **Связь (relationship)** – это функциональная зависимость между двумя сущностями (в частности, возможна связь сущности с самой собой). Например, важно знать фамилию сотрудника, и не менее важно знать, в каком отделе он работает. Таким образом, между сущностями отдел и сотрудник существует связь состоит из (отдел **состоит из** сотрудников). Связь – это понятие логического уровня, которому соответствует внешний ключ на физическом уровне.

Связь является логическим соотношением между сущностями. Каждая связь должна именоваться глаголом (или глагольной фразой), показывающим, как одна сущность соотносится к другой.

Связь показывает, какие именно действия делает клиент. По умолчанию имя связи на диаграмме не показывается. Для отображения имени связи на модели необходимо в меню Format/Relationship Display включить режим Verb Phrase.

В ERwin связи представлены пятью основными элементами информации:

* тип связи (идентифицирующая, неидентифицирующая, полная/неполная категория, неспецифическая связь);
* родительская сущность;
* дочерняя (зависимая) сущность;
* мощность связи (cardinality);
* допустимость пустых (null) значений.

В IDEF1X различают зависимые и независимые сущности. Тип сущности определяется ее связью с другими сущностями.

1. Атрибуты в ERwin. Классификация атрибутов.

**Ответ:** **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.*

Каждый *атрибут*хранит информацию об определенном свойстве *сущности,*а каждый экземпляр *сущности*должен быть уникальным. *Атрибут*или группа *атрибутов,*которые однозначно идентифицируют экземпляр *сущность,*называется *первичным ключом. Атрибуты первичного ключа*на диаграмме не требуют специального обозначения – это те *атрибуты,*которые находятся в списке *атрибутов*выше горизонтальной линии.