

Московский Энергетический Институт
(технический университет)

Лабораторная работа № 04
«Практическое применение методологии IDEF1X»

по курсу
«CASE-технологии разработки программных средств»

Выполнила: студентка группы А-13-06

Бражникова Юлия

Проверил: Куриленко И.Е.

Москва 2011 г.

Теоретическое введение.

Методология IDEF1X - один из подходов к семантическому моделированию данных, основанный на концепции "сущность-связь" (Entity-Relationship). Это инструмент для анализа информационной структуры систем различной природы. Информационная модель, построенная с помощью IDEF1X-методологии, отображает логическую структуру информации об объектах системы

Таким образом, концептуальная модель, представленная в соответствии со стандартом IDEF1X, является логической схемой базы данных для проектируемой системы. Основными объектами концептуальной модели являются сущности и связи.

Сущность - некоторый обособленный объект или событие моделируемой системы, имеющий определенный набор свойств - атрибутов. Отдельный элемент этого множества называется "экземпляром сущности". Сущность может обладать одним или несколькими атрибутами, которые однозначно идентифицируют каждый образец сущности, и может обладать любым количеством связей с другими сущностями.

Правила для атрибутов сущности:

1. Каждый атрибут должен иметь уникальное имя.
2. Сущность может обладать любым количеством атрибутов.
3. Сущность может обладать любым количеством наследуемых атрибутов, но наследуемый атрибут должен быть частью первичного ключа сущности-родителя.
4. Для каждого экземпляра сущности должно существовать значение каждого его атрибута (правило обращения в нуль - Not Null).
5. Ни один из экземпляров сущности не может обладать более чем одним значением для ее атрибута.

Сущность изображается на ER-диаграмме в виде прямоугольника, в верхней части которого приводится ее название; далее следует список атрибутов. Ключевые атрибуты могут быть выделены подчеркиванием или иным способом.

Стандарт IDEF1X описывает способы изображения двух типов сущностей - независимой и зависимой, и связей - идентифицирующих и неидентифицирующих.

Каждая сущность может обладать любым количеством связей с другими сущностями.

Сущность является независимой, если каждый ее экземпляр может быть однозначно идентифицирован без определения его связей с другими сущностями.

Сущность называется зависимой, если однозначная идентификация ее экземпляра зависит от его связей с другими сущностями.

Сущность может обладать атрибутами, которые наследуются через связь с родительской сущностью. Последние обычно являются внешними ключами и служат для организации связей между сущностями. Если внешний ключ сущности используется в качестве ее первичного ключа (ПК) или как часть составного первичного ключа, то сущность является зависимой от родительской сущности. Если внешний ключ не является первичным и не входит в составной первичный ключ, то сущность является независимой от родительской сущности.

Если сущность является зависимой, то связь ее с родительской сущностью

называется идентифицирующей, в противном случае - неидентифицирующей.

Связь изображается на ER-диаграмме линией, проводимой между сущностью-родителем и сущностью-потомком с точкой на конце линии у сущности-потомка. идентифицирующая связь изображается сплошной линией, неидентифицирующая - пунктирной.

Связи дается имя, выражаемое грамматической формой глагола. Для связи дополнительно может присутствовать указание мощности: какое количество экземпляров сущности-потомка может существовать для сущности-родителя. Имя связи всегда формируется с точки зрения родителя, так что может быть образовано предложение, если соединить имя сущности родителя, имя связи, выражение мощности и имя сущности-потомка.

Постановка задачи.

Построить логическую и физическую модели в Erwin согласно системе, описанной в ТЗ в лабораторной работе №1.

Цель работы.

Изучить методологию IDEF1X. Освоить практическое применение ERWin.

Решение задачи.

Описание предметной области.

Предметная область включает в себя следующие объекты: Клиент, Оператор склада, Заказ подарков, Подзаказ, Подарок.

Клиент

Свойства:

- Логин
- Пароль
- Адрес электронной почты
- Фамилия
- Имя
- Номер и серия паспорта
- Контактный телефон

Оператор склада

Свойства:

- Логин
- Пароль
- Адрес электронной почты
- Фамилия
- Имя
- Контактный телефон

Заказ подарков

- Свойства:
- Клиент, создавший заказ
 - Оператор склада, обрабатывающий заказ
 - Суммарная стоимость заказа в баллах
 - Дата создания
 - Статус

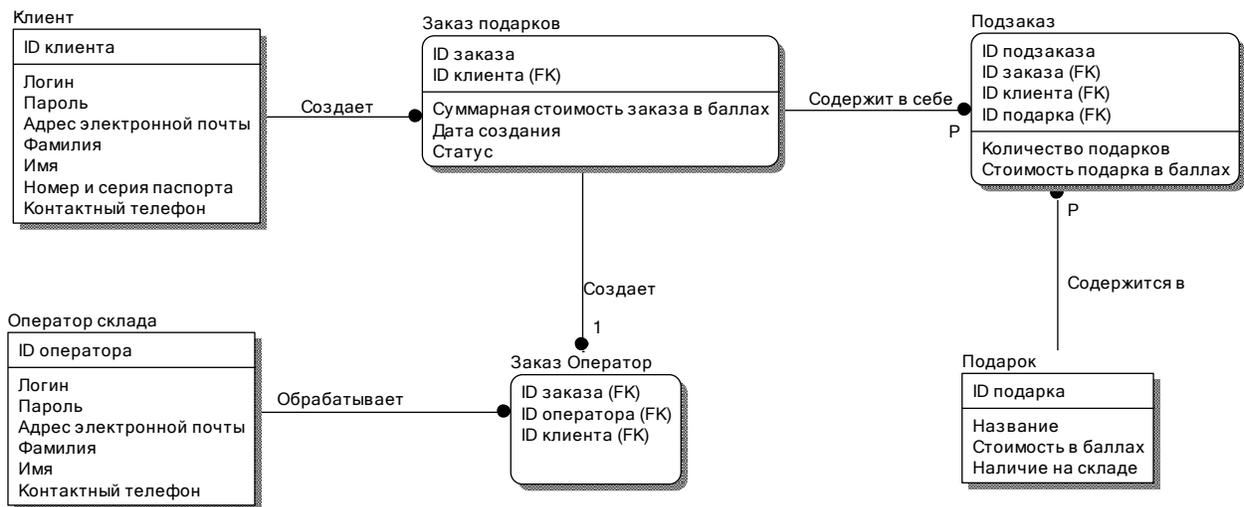
Подзаказ

- Свойства:
- Заказ, который содержит в себе подзаказ
 - Подарок, который содержится в подзаказе
 - Количество подарков
 - Стоимость подарка в баллах

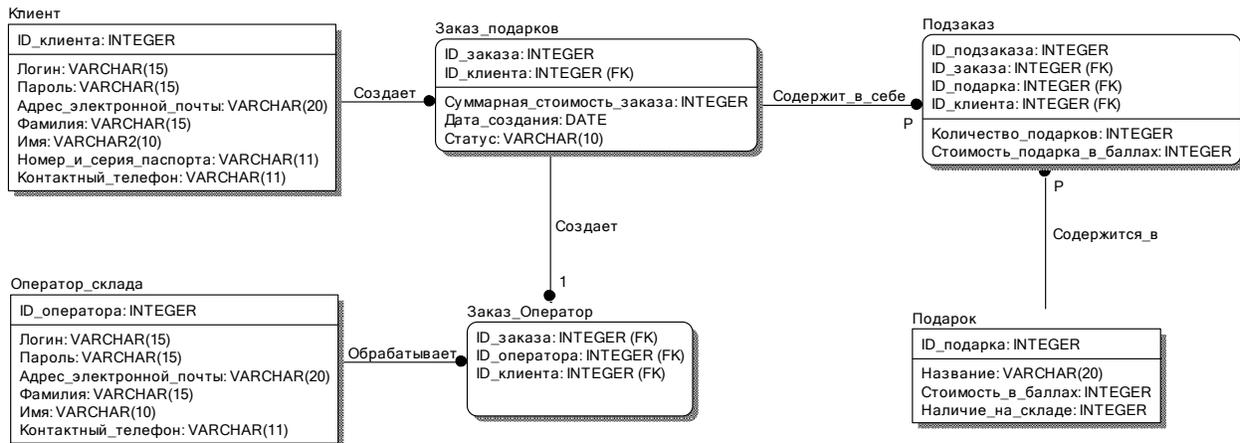
Подарок

- Свойства:
- Название
 - Стоимость в баллах
 - Наличие на складе

Логическая модель БД



Физическая модель БД



Связь Клиент – Заказ подарков

Parent Delete - Restrict

Parent Update - Cascade

Удалять Клиента, для которого есть невыполненный Заказ_подарков, запрещено.

Связь Заказ подарков – Подзаказ

Parent Delete - Cascade

Parent Update - Cascade

При удалении Заказа_подарков, удаляются все его Подзаказы.

Связь Подарок – Подзаказ

Parent Delete – Set Null

Parent Update - Cascade

При удалении Подарка, связанные с ним Подзаказы не удаляются.

Связь Заказ подарков – Заказ Оператор

Parent Delete – Cascade

Parent Update - Cascade

При удалении Заказа_подарков, удаляется Заказ_Оператор.

Связь Оператор склада – Заказ Оператор

Parent Delete – Cascade

Parent Update - Cascade

При удалении Оператора_склада, удаляется Заказ_Оператор.