

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Тульский государственный университет»

Интернет-институт

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине

«Базы данных 1»

Семестр 5

Вариант 3

Выполнил: студент гр. ИБ262521-ф

Артемов Александр Евгеньевич

Проверил: канд. техн. наук, доц.

Французова Юлия Вячеславовна

Тула 2024

Практическая работа № 1.

Название работы: Инфологическое проектирование.

Цели работы: Получение навыков проектирования модели на основе инфологического проектирования.

Задание:

1. Выделить необходимый набор сущностей, отражающих предметную область и информационные потребности пользователей.
2. Определить необходимый набор атрибутов каждой сущности, выделив идентифицирующие атрибуты.
3. Классифицировать атрибуты каждой сущности (описательные, указывающие, вспомогательные).
4. Определить сущности вида подтип/супертип, где это необходимо.
5. Определить связи между сущностями.
6. Проанализировав структуру связей, исключить избыточные.
7. Определить множественность и условность связей.
8. Дать формулировку связей с точки зрения каждой участвующей сущности.
9. Формализовать связи вида 1:1, 1:M, M:N.
10. Классифицировать сущности, разделив их на стержневые, ассоциативные, характеристические, обозначающие.
11. Построить ER-диаграмму модели базы данных.
12. Описать модель базы данных на языке инфологического проектирования.

Выполнение лабораторной работы.

Изучены теоретические сведения практической работы по проектированию модели на основе инфологического проектирования.

Так как в задании практической работы не указана предметная область, мною было принято решение рассмотреть в упрощенном виде систему доставки продуктов из магазина.

1. Выделение необходимого набора сущностей, отражающих предметную область и информационные потребности пользователей.

Рассмотрим доставку продуктов с бытовой точки зрения:

- заказчик при помощи мобильного приложения или веб-сайта магазина просматривает список доступных продуктов;
- заказчик добавляет необходимые ему продукты в корзину (указывая количество);

- заказчик оформляет заказ, оплачивая его (сделаем допущение, что доступна только оплата онлайн, ковидные времена и все такое);
- заказ поступает доставщику;
- доставщик собирает заказ в ближайшем магазине;
- доставщик выполняет доставку заказчику.

В данной предметной области можно выделить такие сущности как **ЗАКАЗЧИК**, **ЗАКАЗ**, **ПРОДУКТ**, **ПРОДУКТЫ_В_ЗАКАЗЕ**, **ДОСТАВЩИК**.

2. Определение необходимого набора атрибутов каждой сущности, выделение идентифицирующих атрибутов.

ЗАКАЗЧИК. Заказчик — это любой человек, который делает заказ в приложении, оплачивает и принимает его. Характерными атрибутами данной сущности являются имя и телефон заказчика.

Сущность **ЗАКАЗЧИК** имеет идентификатор **Номер_заказчик**, и атрибуты **Имя** и **Телефон**.

ДОСТАВЩИК. Доставщик — это сотрудник службы доставки, который принимает заказ, собирает продукты, указанные в заказе, и доставляет по указанному адресу заказчику. Характерными атрибутами данной сущности являются имя и телефон доставщика.

Сущность **ДОСТАВЩИК** имеет идентификатор **Номер_доставщик**, и атрибуты **Имя** и **Телефон**.

ЗАКАЗ. В сущности **ЗАКАЗ** необходимо наличие атрибутов указывающих на доставщика, выполняющего заказ, и заказчика. Так же для заказа необходим атрибут даты и времени создания заказа, так как доставщики имеют довольно сжатые сроки доставки продуктов. Так как заказ может находиться в различных стадиях выполнения (например, получен, в сборке, доставляется, выполнен), необходим атрибут статуса заказа. В качестве идентифицирующего атрибута можно использовать уникальный номер заказа, автоматически генерируемый СУБД по определенным правилам.

Таким образом, сущность **ЗАКАЗ** имеет идентификатор **Номер_заказ**, и атрибуты **Номер_заказчик**, **Адрес**, **Номер_доставщик**, **Дата_время**, **Статус**.

ПРОДУКТ. Продукт — это то, что мы покупаем в магазине, например, еда или одежда. Каждый продукт уникален своими характеристиками, хотя их возможно семантически объединять в группы. Например, помидоры бывают разных сортов, а джинсы различных моделей и размеров. Поэтому каждый продукт имеет собственную позицию или артикул. Этот атрибут является идентификатором продукта. Так же каждый продукт имеет наименование и цену за одну единицу товара (например, за пару джинсов или за килограмм помидоров).

Сущность **ПРОДУКТ** имеет идентификатор **Артикул**, и атрибуты **Наименование** и **Цена**.

ПРОДУКТЫ_В_ЗАКАЗЕ. Для хранения информации что именно заказано необходима вспомогательная сущность, которая связывает

сущности заказ и продукт. В большинстве случаев заказываются сразу несколько продуктов в разных количествах. В данном случае сущность должна иметь атрибуты указывающие как на заказ, так и на продукт, а так же количество продукта.

Сущность **ПРОДУКТЫ_В_ЗАКАЗЕ** имеет атрибуты Номер_заказ, Артикул, Количество. Атрибуты Номер_заказ и Артикул являются составным идентификатором.

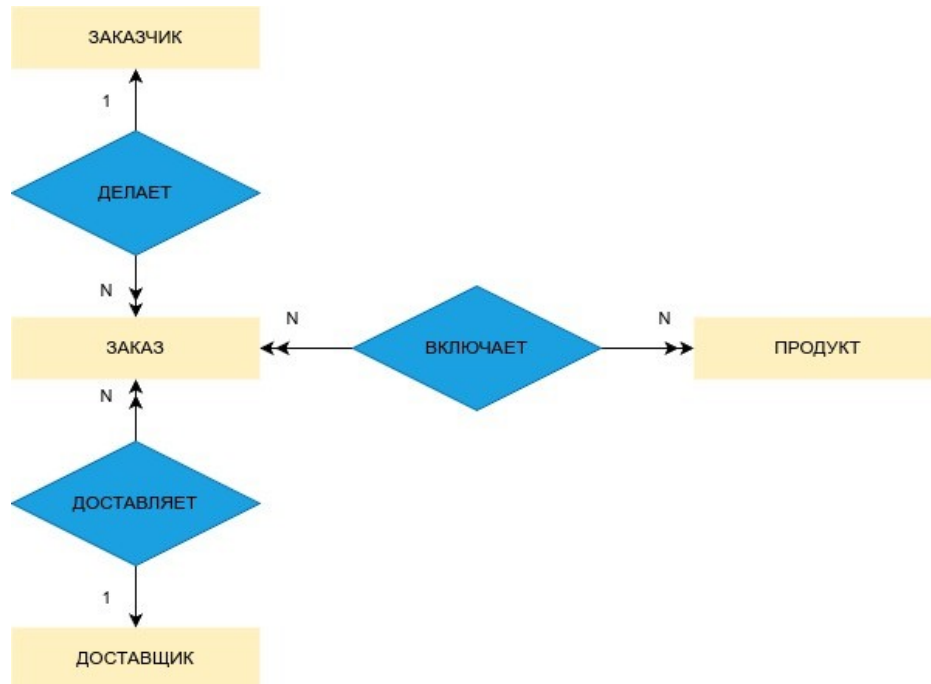
3. Классификация атрибутов сущностей (описательные, указывающие, вспомогательные).

Сущность	Атрибут	Классификация
ДОСТАВЩИК	Номер_доставщик	Указывающий
	Имя	Описательный
	Телефон	Описательный
ЗАКАЗЧИК	Номер_заказчик	Указывающий
	Имя	Описательный
	Телефон	Описательный
ЗАКАЗ	Номер_заказ	Указывающий
	Номер_заказчик	Вспомогательный
	Адрес	Описательный
	Номер_доставщик	Вспомогательный
	Дата_время	Описательный
	Статус	Описательный
ПРОДУКТ	Артикул	Указывающий
	Наименование	Описательный
	Цена	Описательный
ПРОДУКТЫ_В_ЗАКАЗЕ	Номер_заказ	Вспомогательный
	Артикул	Вспомогательный
	Количество	Описательный

4. Определение сущностей вида подтип/супертип, где это необходимо.

В данной предметной области нет необходимости определения сущностей вида подтип/супертип.

5. Определение связи между сущностями.



6. Анализ структуры связей, исключение избыточных связей.

Сущность ЗАКАЗЧИК делает ЗАКАЗ. ЗАКАЗ включает в себя экземпляры сущности ПРОДУКТ. Сущность ДОСТАВЩИК доставляет ЗАКАЗ ЗАКАЗЧИКУ.

7. Определение множественности и условности связей.

Существует три вида связи: один-к-одному, один-ко-многим и многие-ко-многим.

Связь один-к-одному существует, когда один экземпляр одного объекта связан с единственным экземпляром другого. Связь один-к-одному обозначается стрелками.

Связь один-ко-многим существует, когда один экземпляр первого объекта связан с одним (или более) экземпляром второго объекта, но каждый экземпляр второго объекта связан только с одним экземпляром первого. Множественность связи изображается двойной стрелкой.

Связь многие-ко-многим существует, когда один экземпляр первого объекта связан с одним или большим количеством экземпляров второго и каждый экземпляр второго связан с одним или многими экземплярами первого. Этот тип связи изображается двусторонней стрелкой.

Помимо множественности, связи могут подразделяться на безусловные и условные. В безусловной связи для участия в ней требуется каждый экземпляр объекта. В условной связи принимают участие не все экземпляры объекта. Связь может быть условной как с одной, так и с обеих сторон.

Эти три фундаментальных вида связи относятся к числу безусловных связей и требуют участия каждого экземпляра сущности.

8. Формулировка связей с точки зрения каждой участвующей сущности.

Связь ДЕЛАЕТ между сущностями ЗАКАЗЧИК и ЗАКАЗ является связью вида один-ко-многим, так как ЗАКАЗЧИК может делать множество ЗАКАЗОВ, но у каждого ЗАКАЗА может быть только один ЗАКАЗЧИК. Эта связь является безусловной, так как требует участия каждого экземпляра сущностей.

Связь ДОСТАВЛЯЕТ между сущностями ДОСТАВЩИК и ЗАКАЗ является связью вида один-ко-многим, так как ДОСТАВЩИК может доставлять множество ЗАКАЗОВ, но у каждого ЗАКАЗА может быть только один ДОСТАВЩИК. Эта связь является безусловной, так как требует участия каждого экземпляра сущностей.

Связь ВКЛЮЧАЕТ между сущностями ЗАКАЗ и ПРОДУКТ является связью вида многие-ко-многим, так как ЗАКАЗ может включать один или множество ПРОДУКТОВ, а каждый ПРОДУКТ может быть включен в один или множество ЗАКАЗОВ. Эта связь является безусловной, так как требует участия каждого экземпляра сущностей.

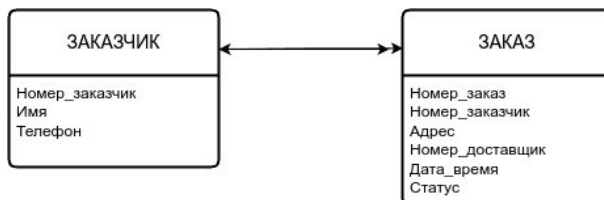
9. Формализация связей вида один-к-одному, один-ко-многим и многие-ко-многим.

Формализация связи вида один-к-одному не приводится, так как не рассматривается в данной предметной области.

Для формализации связи один-ко-многим вспомогательные атрибуты должны быть добавлены к сущности на стороне «многого», поскольку размещение такого вспомогательного атрибута на стороне «один» будет нарушать правило атрибутов (когда сущность имеет составной идентификатор, каждый атрибут, не являющийся частью идентификатора, представляет характеристику всей сущности, а не ее части, а тем более не характеристику чего-либо другого).

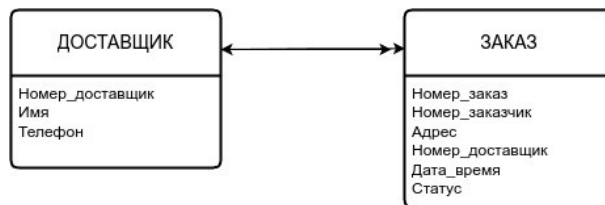
Для формализации связи «многие-ко-многим» создают отдельную ассоциативную сущность, которая содержит ссылки на идентификаторы каждого из участвующих экземпляров.

Формализация связи ДЕЛАЕТ между сущностями ЗАКАЗЧИК и ЗАКАЗ:



Сущность ЗАКАЗ имеет атрибут Номер_заказчик, который содержит ссылку на идентификатор сущности ЗАКАЗЧИК. Связь вида один-ко-многим.

Формализация связи ДОСТАВЛЯЕТ между сущностями ДОСТАВЩИК и ЗАКАЗ:



Сущность ЗАКАЗ имеет атрибут Номер_доставщик, который содержит ссылку на идентификатор сущности ДОСТАВЩИК . Связь вида один-ко-многим.

Формализация связи ВКЛЮЧАЕТ между сущностями ЗАКАЗ и ПРОДУКТ:



В данном случае имеется связь вида многие-ко-многим, поэтому создана отдельная ассоциативная сущность ПРОДУКТЫ_В_ЗАКАЗЕ, которая содержит ссылки на идентификаторы сущностей ЗАКАЗ и ПРОДУКТЫ. Подобно любой другой, ассоциативная сущность ПРОДУКТЫ_В_ЗАКАЗЕ может иметь дополнительные атрибуты и участвовать в связях с другими сущностями.

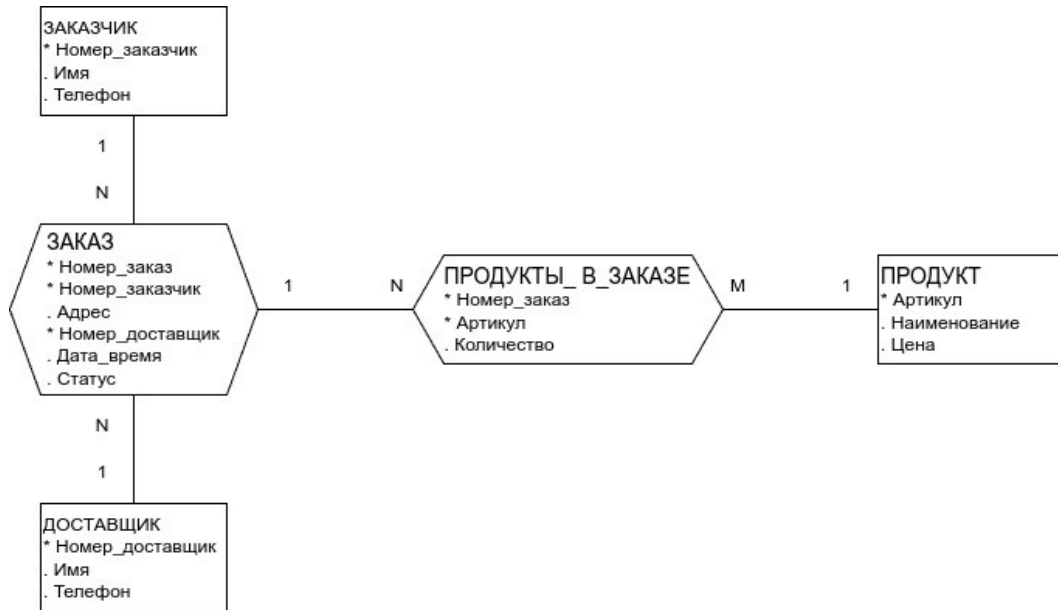
10. Классификация сущностей, разделение их на стержневые, ассоциативные, характеристические, обозначающие.

В данной предметной области стержневыми являются сущности ЗАКАЗЧИК и ДОСТАВЩИК, так как они являются независимыми.

В контексте сущностей ЗАКАЗЧИК и ДОСТАВЩИК сущность ЗАКАЗ является ассоциативной.

В контексте сущностей ЗАКАЗ и ПРОДУКТ сущность ПРОДУКТЫ_В_ЗАКАЗЕ является ассоциативной, а сущности ЗАКАЗ и ПРОДУКТ — стержневыми.

11. Построение ER-диаграммы модели базы данных.



В расширенной ER-модели ассоциация изображается шестигранником.

12. Описание модели базы данных на языке инфологического проектирования.

ЗАКАЗЧИК (Номер_заказчик, Имя, Телефон)

ДОСТАВЩИК (Номер_доставщик, Имя, Телефон)

ЗАКАЗ [**ЗАКАЗЧИК** M, **ДОСТАВЩИК** N] (Номер_заказ, Номер_заказчик, Номер_доставщик, Адрес, Дата_время, Статус)

ПРОДУКТ (Артикул, Наименование, Цена)

ПРОДУКТЫ_В_ЗАКАЗЕ [**ЗАКАЗ** M, **ПРОДУКТ** N] (Номер_заказ, Артикул, Количество)

Ответы на контрольные вопросы.

1. Каковы задачи, решаемые на этапе инфологического проектирования?

Первой задачей инфологического проектирования является определение ПО системы, позволяющее изучить информационные потребности будущих пользователей. Другая задача этого этапа – анализ ПО, который призван сформировать взгляд на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО. Анализ ПО выполняется разработчиком логической базы данных – специалистом в данной ПО.

2. В чем состоит отличие понятия типа сущности и элемента сущности?

Понятие тип сущности относится к набору однородных предметов, событий, личностей, выступающих как единое целое. Элемент сущности относится к конкретной вещи в наборе. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не некоторого конкретного элемента этого типа. Для большей выразительности и лучшего понимания имя сущности может сопровождаться примерами конкретных сущностей этого типа.

3. Каковы способы представления сущности?

В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. Для идентификации конкретных экземпляров сущности используются атрибуты, выполняющие роль идентификаторов.

4. Каковы правила атрибутов?

Один экземпляр сущности имеет одно единственное значение для каждого атрибута в любое данное время. В табличной интерпретации это правило требует, чтобы существовал один и только один элемент данных для каждой строки и столбца. Это делает невозможным появление незаполненных ячеек и ячеек с группой значений.

Атрибут не должен содержать никакой внутренней структуры.

Когда сущность имеет составной идентификатор, каждый атрибут, не являющийся частью идентификатора, представляет характеристику всей сущности, а не ее части, а тем более не характеристику чего-либо другого.

Каждый атрибут, не являющийся частью идентификатора, представляет характеристику экземпляра, указанного идентификатором, а не характеристику некоторого другого атрибута - неидентификатора.

5. Как классифицируются атрибуты?

Описательные атрибуты представляют характеристики, внутренне присущие каждому экземпляру сущности. Если значение описательного атрибута изменяется, то это говорит о том, что некоторый аспект экземпляра сущности изменился, но сам экземпляр остался прежним.

Указывающие атрибуты используются для присвоения имени или обозначения экземпляров сущности. Изменение значения указывающего атрибута говорит о том, что данному экземпляру дается новое имя. Указывающие атрибуты часто используются как идентификатор или как часть идентификатора.

Вспомогательные атрибуты используются для связи экземпляра одного сущности с экземпляром другого. Если значение вспомогательного атрибута меняется, это означает, что теперь другие экземпляры связаны между собой.

6. Каковы фундаментальные виды связей?

Существуют три фундаментальных вида связи: один-к-одному (1:1), один-ко-многим (1:M), многие-ко-многим (M:N). Эти фундаментальные виды связей относятся к числу безусловных связей и требующих участия каждого экземпляра сущности.

7. Как формализуется связь 1:1?

Связь один-к-одному (1:1) существует, когда один экземпляр одной сущности связан с единственным экземпляром другой сущности.

8. Как формализуется связь 1:M?

Связь один-ко-многим (1:M) существует, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан только с одним экземпляром первой сущности.

9. Как формализуется связь M:N?

Связь многие-ко-многим (M:N) существует, когда один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан с одним или более экземпляром первой сущности.

10. Что такое композиция связей?

При композиции родительский класс сущностей (или агрегат) связывается с несколькими дочерними классами сущностей (или компонентами). При этом компоненты агрегата (т.е. компоненты родительского класса сущностей) ссылаются на агрегат посредством внешнего ключа, входящего в состав первичного ключа и, следовательно, не могут существовать вне агрегата.

Композиция представляет собой усиленную форму простой агрегации. Композиция характеризуется тем, что: ссылка на агрегат участвует в

идентификации компонентов; эти компоненты не могут существовать вне агрегата.