

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Экономическая теория и моделирование экономических процессов»

# ***ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ***

Методические указания

к выполнению практических и самостоятельных работ  
для студентов очной формы обучения специальности 080502

Часть II

Курган 2009

Кафедра: «Экономическая теория и моделирование экономических процессов»

Дисциплина: «Информационные технологии в экономике» (специальность 080502)

Составил: старший преподаватель С.А. Хмелев

Методические указания утверждены на заседании кафедры  
« 29 » июня 2009 г.

Рекомендованы методическим советом университета  
« 2 » июля 2009 г.

## Содержание

Введение.....	4
1 Теоретические основы проектирования информационной системы.....	5
1.1 Некоторые определения.....	5
1.2 Обследования предметной области.....	8
1.3 Системное проектирование.....	8
1.3.1 Типы моделей данных.....	9
1.3.2 Взаимосвязи в модели.....	11
2 Проектирование базы данных.....	13
2.1 Этап I Определение сущностей.....	15
2.2 Этап II Определение взаимосвязей между сущностями.....	15
2.3 Этап III Задание первичных и альтернативных ключей, определение атрибутов сущностей.....	16
2.4 Этап IV Приведение модели к требуемому уровню нормальной формы.....	18
2.5 Этап V Физическое описание модели.....	23
3 Варианты заданий для самостоятельной работы.....	24
Список литературы.....	31

## **Введение**

Цель курса «Информационные технологии в экономике» - помочь освоить организацию и использование автоматизированных информационных технологий в экономике; познакомиться с современными методами проектирования программного обеспечения экономических информационных систем.

Программа курса «Информационные технологии в экономике» составлена в соответствии со стандартом по специальности 080502 «Экономика и управление на предприятии (в машиностроении)». В соответствии с Государственным образовательным стандартом обязательный минимум содержания дисциплины составляет: понятие информации; методы ее оценки; информационные модели; информатика и информационные технологии; структура базовой информационной технологии; базы данных (БД); принципы построения; жизненный цикл БД; типы БД; объектно-ориентированные БД; организация проектирования информационной технологии на предприятии.

Каждой цивилизации приходится иметь дело с обработкой информации. С развитием экономики и ростом численности населения возрастает объем взаимосвязанных данных, необходимых для решения коммерческих и административных задач. Взаимосвязанные данные называют информационной системой. Такая система в первую очередь призвана облегчить труд человека, но для этого она должна как можно лучше соответствовать очень сложной модели реального мира.

Ядром информационной системы являются хранимые в ней данные. На любом предприятии данные различных отделов, как правило, пересекаются, то есть используются в нескольких подразделениях или являются общими. Например, для целей управления часто нужна информация по всему предприятию. Заказ комплектующих невозможен без наличия информации о запасах. Хранящиеся в информационной системе данные должны быть легко доступны том виде, в каком они нужны для конкретной производственной деятельности предприятия.

Практический курс «Информационные технологии в экономике» разделен на две части. К первой части курса относится технология работы в среде Microsoft Access. Студенты знакомятся с методикой работы в этой среде, а затем выполняют самостоятельную работу по созданию и использованию базы данных, исходя из предложенных вариантов.

Вторая часть практического курса знакомит студентов с методикой организации работы с данными и создания автоматизированной информационной системы небольшого предприятия. Также рассматриваются проблемы, возникающие при планировании автоматизированной системы обработки информации, вопросы аппаратного и программного обеспечения будущей системы, даются рекомендации по построению информационной модели и реляционной базы данных.

Автоматизированная информационная технология (АИТ) имеет свои цели, методы и средства реализации.

Целью автоматизированной информационной технологии является создание из информационного ресурса качественного информационного продукта, удовлетворяющего требованиям пользователя.

Методами АИТ являются методы обработки и передачи данных. Средства АИТ - это математические, программные, информационные, технические и др.

При таком определении целей, методов и средств под автоматизированной информационной технологией будем понимать целостную техническую систему, обеспечивающую целенаправленное создание, передачу, хранение и отображение информационного продукта (данных, идей, знаний) с наименьшими затратами и в соответствии с закономерностями той социальной среды, где развивается информационная технология.

## **1 Теоретические основы проектирования информационной системы**

### **1.1 Некоторые определения**

Хранящиеся в информационной системе данные должны быть легко доступны в том виде, в каком они нужны для конкретной производственной деятельности предприятия. При этом не имеет существенного значения способ хранения данных. Сегодня на предприятии мы можем встретить систему обработки данных традиционного типа, в которой служащий вручную помещает данные в скоросшиватель, и рядом с ней — современную систему с применением самой быстродействующей ЭВМ, сложнейшего оборудования и программного обеспечения. Несмотря на поразительную несхожесть, обе эти системы обязаны предоставлять достоверную информацию в определенное время, определенному лицу, в определенном месте и с ограниченными затратами, чтобы понять процесс построения информационной системы, необходимо знать ряд терминов, которые применяются при описании и представлении данных.

**Предметной областью** называется часть реальной системы, представляющая интерес для данного исследования.

При проектировании автоматизированных информационных систем предметная область отображается моделями данных нескольких уровней. Число используемых уровней зависит от сложности системы, но в любом случае включает логический и физический уровни. Предметная область может относиться к любому типу организации (например, банк, университет, больница или завод). Необходимо различать полную предметную область (крупное производственное предприятие, склад, универмаг и т.д.) и организационную единицу этой предметной области. Организационная единица в свою очередь может представлять свою предметную область (например, цех по производству кузовов автомобильного завода или отдел обработки данных предприятия по производству ЭВМ). В данном случае цехи и отделы сами могут соответствовать определенным предметным областям.

Информация, необходимая для описания предметной области, зависит от реальной модели и может включать сведения о персонале, заработной плате,

товарах, накладных, счетах, отчетах по сбыту, лабораторных тестах, финансовых сделках, историях болезней, то есть сведения о людях, местах, предметах, событиях и понятиях.

**Объектом** называется элемент информационной системы, информацию о котором мы сохраняем. В реляционной теории баз данных объект называется сущностью.

Объект может быть реальным (например, человек, какой-либо предмет или населенный пункт) и абстрактным (например, событие, счет покупателя или изучаемый студентами курс). Так, в области продажи автомобилей примерами объектов могут служить МОДЕЛЬ АВТОМОБИЛЯ, КЛИЕНТ и СЧЕТ. На товарном складе - это ПОСТАВЩИК, ТОВАР, ОТПРАВЛЕНИЕ и т.д. Каждый объект обладает определенным набором свойств, которые запоминаются в информационной системе. При обработке данных часто приходится иметь дело с совокупностью однородных объектов, например, таких, как служащие, и записывать информацию об одних и тех же свойствах для каждого из них.

**Классом объектов** называют совокупность объектов, обладающих одинаковым набором свойств.

Таким образом, для объектов одного класса набор свойств будет одинаков, хотя значения этих свойств для каждого объекта, конечно, могут быть разными. Например, класс объектов МОДЕЛЬ АВТОМОБИЛЯ будет иметь одинаковый набор свойств, описывающих характеристики автомобилей, и каждая модель будет иметь различные значения этих характеристик.

Объекты и их свойства являются понятиями реального мира. В мире информации, существующем в представлении программиста, говорят об атрибутах объектов.

**Атрибут** — это информационное отображение свойств объекта. Каждый объект характеризуется рядом основных атрибутов.

Например, модель автомобиля характеризуется типом кузова, рабочим объемом двигателя, количеством цилиндров, мощностью, габаритами, названием и т.д. Клиент магазина, продающего автомобили, имеет такие атрибуты, как фамилия, имя, отчество, адрес и, возможно, идентификационный номер. Каждый атрибут в модели должен иметь уникальное имя — идентификатор. Атрибут при реализации информационной модели на каком-либо носителе информации часто называют элементом данных, полем данных или просто полем.

**Значение данных** представляет собой действительные данные, содержащиеся в каждом элементе данных.

В зависимости от того, как элементы данных описывают объект, их значения могут быть количественными, качественными или описательными. Информацию о некоторой предметной области можно представить с помощью нескольких объектов, каждый из которых описывается несколькими элементами данных. Принимаемые элементами данных значения называются данными. Единичный набор принимаемых элементами данных значений называется экземпляром объекта. Объекты связываются между собой определенным образом. Соответствующая модель объектов с составляющими их элементами данных и взаимосвязями называется концептуальной моделью.

**Ключевым элементом данных** называется такой элемент, по которому можно определить значения других элементов данных.

Однозначно идентифицировать объект могут два и более элемента данных. В этом случае их называют «кандидатами» в ключевые элементы данных. Вопрос о том, какой из кандидатов использовать для доступа к объекту, решается пользователем или разработчиком системы. Выбирать ключевые элементы данных следует тщательно, поскольку правильный выбор способствует созданию достоверной концептуальной модели данных.

**Первичный ключ** — это атрибут (или группа атрибутов), которые единственным образом идентифицируют каждую строку в таблице.

Понятие первичного ключа является исключительно важным в связи с понятием целостности баз данных, которое мы подробно рассмотрим в конце этого параграфа.

**Запись данных** — это совокупность значений связанных элементов данных. Тип данных характеризует вид хранящихся данных.

Понятие типа данных в информационной модели полностью адекватно понятию типа данных в языках программирования. Обычно в современных СУБД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (например, суммы в денежных единицах), а также данных специального формата (дата, время, временной интервал и пр.)

**Связь** — это функциональная зависимость между сущностями.

Если между некоторыми сущностями существует связь, то факты из одной сущности ссылаются или некоторым образом связаны с фактами из другой сущности. Поддержание непротиворечивости функциональных зависимостей между сущностями называется ссылочной целостностью.

**Ссылочная целостность** — это обеспечение соответствия значения внешнего ключа экземпляра дочерней сущности значениям первичного ключа в родительской сущности.

Ссылочная целостность может контролироваться при всех операциях, изменяющих данные.

**Нормализация отношений** — это процесс построения оптимальной структуры таблиц и связей в реляционной БД.

В процессе нормализации элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых могут быть представлены те же данные.

**Словарь данных** — это централизованное хранилище сведений об объектах, составляющих их элементах данных, взаимосвязях между объектами, их источниках, значениях, использовании и форматах представления.

## 1.2 Обследования предметной области

Временные технологии обследования предметной области базируются на концепции и программных средствах реинжиниринга бизнес-процессов.

**Реинжиниринг бизнес-процессов** — это фундаментальное переосмысление и радикальное перепланирование критических бизнес-процессов в процессе внедрения средств информатизации, имеющее целью резко улучшить качество функционирования с точки зрения затрат, качества обслуживания и скорости выполнения.

**Бизнес-процесс** — это упорядоченное множество операций (видов) деятельности, реализуемых в организационной структуре в соответствии с ее миссией и целями, начинающихся с одной или более входных операций и заканчивающихся созданием продукции, необходимой клиенту.

На этапе предпроектного обследования проводится сбор данных о предприятии и осуществляемых им видах деятельности, описываются бизнес-процессы объекта автоматизации, т.е. в том виде, в котором они осуществляются на текущий момент. Далее описания бизнес-процессов тем или иным способом формализуются и подвергаются тщательному анализу. На основе этого анализа и возможностей внедряемой компьютерной системы выдвигаются предложения по оптимизации бизнес-процессов в условиях комплексной автоматизации. Иначе говоря, формируются предложения о том, как должно быть.

## 1.3 Системное проектирование

На стадии системного проектирования осуществляется организация работы с данными, которая является одним из основных этапов создания прикладной программы. Именно от этого первого этапа, действительно очень важного и многими недооцениваемого, зачастую зависит создание всей системы автоматизированной обработки данных и успех ее эксплуатации у конечного пользователя.

Для обсуждения этой проблемы нам придется хотя бы слегка погрузиться в теорию информационных систем и начать со средств описания организации данных — информационных моделей.

**Информационные модели (ИМ)** — это средство формирования представления о данных, их составе и использовании в конкретных условиях. Используя современные пакеты прикладных программ, конечно, можно, не прибегая к ИМ, создать достаточно сложные БД, но если вы хотите, чтобы информационная система работала быстро и занимала минимальный объем, вам не обойтись без предварительного анализа задачи с помощью ИМ.

Для описания информационной задачи используются три вида представления ИМ:

1 концептуальное — охватывает всю задачу с точки зрения администратора информационной системы, т.е. лица, ответственного в целом за работу БД;

2 внешнее — отражает представление о задаче с точки зрения конкретного пользователя, т.е. лица, решающего узкую задачу работы системы на конкретном рабочем месте; следовательно, каждая информационная система будет иметь несколько внешних представлений ИМ;

3 внутреннее — отражает представление об информационной задаче разработчика (программиста) с учетом особенностей и возможностей конкретной СУБД и компьютеров, на которых будет реализовываться решаемая задача.

### 1.3.1 Типы моделей данных

Различия в способе представления взаимосвязей между объектами определяют тип модели данных. Для описания информационных процессов используют три основных типа моделей: иерархическую, сетевую и реляционную.

Иерархическая модель организует данные в виде древовидной структуры и является реализацией логических связей по типу «целое-часть». Хорошим примером иерархической модели является любая административная структура. В этой модели выделяется главный объект, через который обеспечивается доступ ко всем остальным объектам системы.

Сетевая модель позволяет любому объекту быть связанным с любым другим объектом. Сетевые модели образуют сложные структуры, что создает определенные трудности при необходимости модернизации или развития ИМ.

Реляционная модель представляет объекты и взаимосвязи между ними в виде таблиц. Впервые данная модель была предложена в 1970 году доктором Э. Ф. Коддом (E.F. Codd). На этой модели данных базируются практически все современные СУБД для ПЭВМ. Существенными преимуществами реляционной модели являются простота представления данных, так как таблицы являются для этого привычной формой, естественный характер формирования запросов, так как они не привязаны жестко к заранее определенной структуре, и высокая степень независимости данных, так как интерфейс пользователя не связан с деталями физической структуры памяти и стратегией доступа. На реляционной модели данных базируется и СУБД Microsoft Access. В Microsoft Access используемая модель получила дальнейшее развитие в плане приближения к классическим канонам построения реляционной модели.

Причина различных видов представлений ИМ заключается в том, что только философы могут позволить себе иметь дело с реальным миром. При решении конкретных задач реальная действительность воспроизводится с существенными ограничениями, зависящими от области деятельности, поставленных целей и мощности вычислительных средств. В дальнейшем реальные информационные процессы в системе будем называть предметной областью данных. Взаимосвязь трех видов представлений ИМ можно показать в виде схемы, приведенной на рисунке 1.

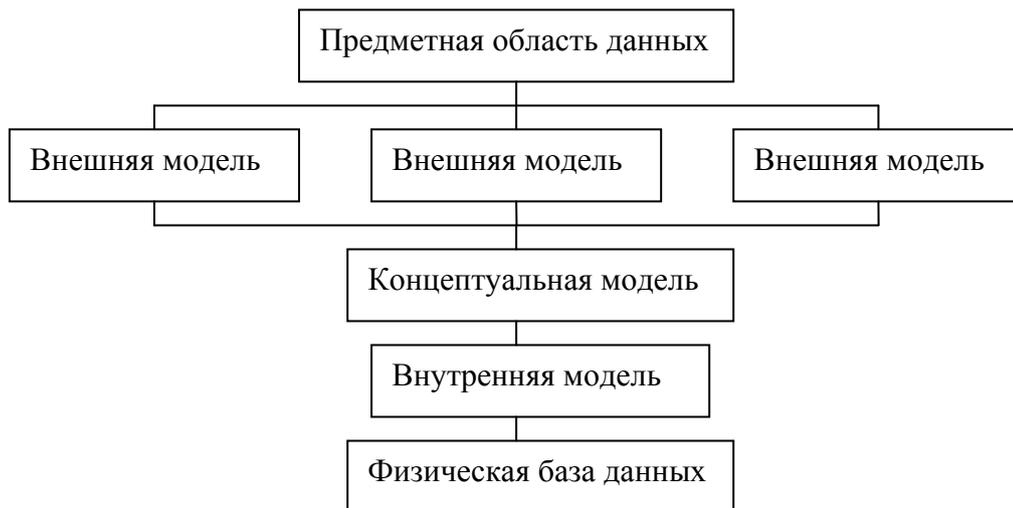


Рисунок 1 - Взаимосвязь трех видов представления ИМ

Таким образом, реальные информационные процессы отображаются через восприятие служебных функций отдельных пользователей, объединяются с точки зрения работы информационной системы в целом, и на этой основе разрабатывается ИМ, которая физически реализуется в виде программы для компьютера и баз данных, размещенных на физическом носителе информации (магнитном диске).

Процесс создания информационной модели начинается с определения концептуальных требований ряда пользователей (рисунок 2).

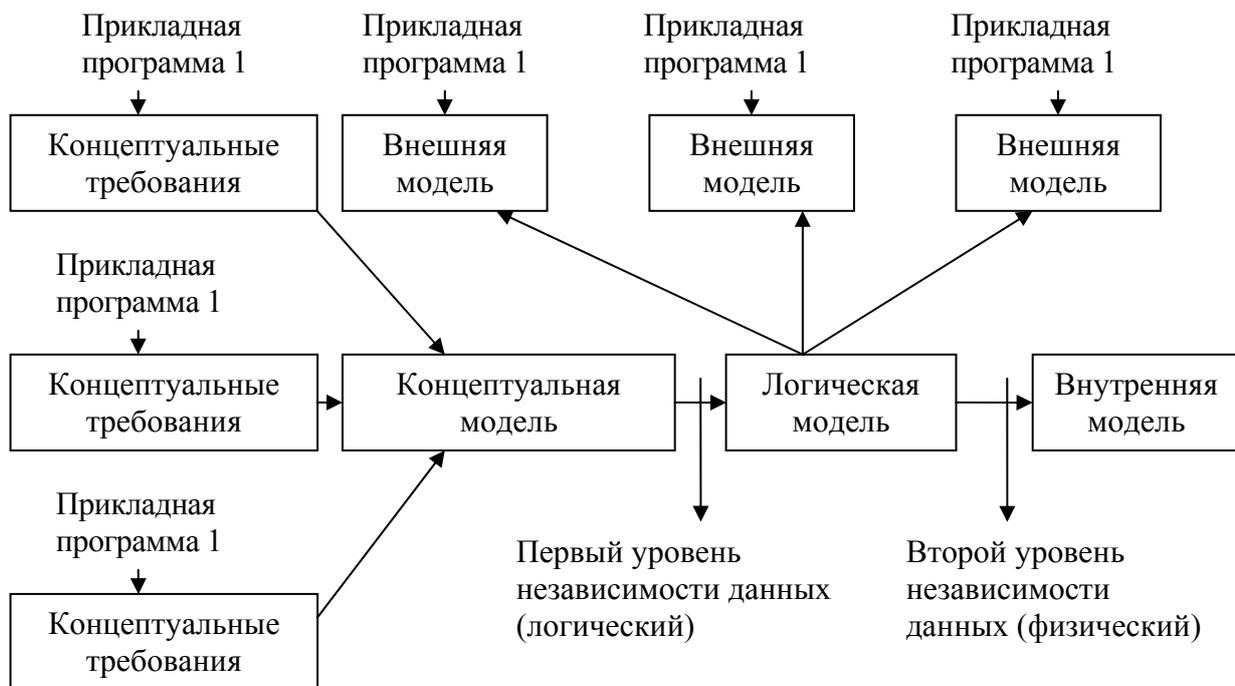


Рисунок 2 - Процесс проектирования информационной модели

Требования отдельных пользователей интегрируются в едином «обобщенном представлении», которое называется концептуальной моделью. Концептуальная модель представляет объекты и их взаимосвязи без указания способов их физического хранения. Таким образом, концептуальная модель является, по существу, моделью предметной области. При проектировании концептуальной модели все усилия должны быть направлены на структуризацию данных и выявления взаимосвязей между ними. Проектирование концептуальной модели основано на анализе на предприятии задач по обработке данных.

Концептуальная модель транслируется затем в модель данных, совместимую с выбранной СУБД. Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называется логической моделью.

Логическая модель отражает логические связи между элементами данных вне зависимости от их содержания и среде хранения.

Логическая модель отображается в физическую память, такую как диск, лента или какой-либо носитель информации.

Физическая модель, определяющая размещение данных, метод доступа и технику индексирования, называется внутренней моделью системы.

### **1.3.2 Взаимосвязи в модели**

Взаимосвязь выражает отображение или связь между двумя множествами данных. Различают взаимосвязи типа «один к одному», «один ко многим» и «многие ко многим».

Рассмотрим задачу по автоматизации управления работой дилера по продаже легковых автомобилей.

Фирма «Фронтон» занимается продажей легковых автомобилей на заказ. Процесс продажи происходит следующим образом. Покупатель производит заказ на покупку автомобиля, пользуясь предоставленным ему фирмой каталогом легковых автомобилей. Представитель фирмы выписывает счет на выбранную модель автомобиля и одновременно с этим отправляет запрос о приобретении данного автомобиля на завод-изготовитель (фирме — поставщику). Фирма «Фронтон» заключила юридические соглашения о поставке автомобилей с рядом заводов-изготовителей легковых автомобилей и крупных дистрибьюторов. После оплаты по соответствующему счету фирма «Фронтон» подтверждает запрос о приобретении и обязуется в течение четырехнедельного периода предоставить покупку соответствующему покупателю.

Если клиент производит заказ на покупку автомобиля впервые, осуществляется первичная регистрация его данных и сведений о сделанном заказе. Если же клиент производит заказ повторно, осуществляется регистрация только данного заказа. Вне зависимости от того, сколько раз данный клиент производил заказы, он имеет уникальный идентификационный номер (уникальный ключ клиента). Информация о каждом клиенте включает наименование клиента, адрес, телефон, факс, фамилию, имя, отчество, признак юридического лица и примечание. Таким образом, атрибутами объекта КЛИЕНТ

являются «УНИКАЛЬНЫЙ КЛЮЧ КЛИЕНТА», «НАИМЕНОВАНИЕ КЛИЕНТА», «АДРЕС КЛИЕНТА» и т.д.

Следующий представляющий для нас интерес объект — МОДЕЛЬ АВТОМОБИЛЯ. Этот объект имеет атрибуты «УНИКАЛЬНЫЙ КЛЮЧ МОДЕЛИ», «НАИМЕНОВАНИЕ МОДЕЛИ» и т.д.

Третий рассматриваемый объект — ЗАКАЗ. Его атрибутами являются «НОМЕР ЗАКАЗА», «КЛЮЧ КЛИЕНТА» и «КЛЮЧ МОДЕЛИ».

Четвертый рассматриваемый объект — ПРОДАВЕЦ. Его атрибутами являются «УНИКАЛЬНЫЙ КЛЮЧ ПРОДАВЦА», «ИМЯ ПРОДАВЦА», «ФАМИЛИЯ» и «ОТЧЕСТВО».

Взаимосвязь «один к одному» (между двумя типами объектов). Допустим, в определенный момент времени один клиент может сделать только один заказ. В этом случае между объектами КЛИЕНТ и ЗАКАЗ устанавливается взаимосвязь «один к одному», обозначаемая одинарными стрелками, как это показано на рисунке 3.

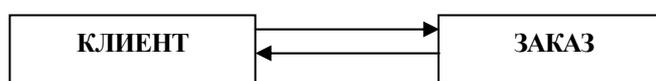


Рисунок 3 - Взаимосвязь «один к одному»

Между данными, хранящимися в объектах КЛИЕНТ и ЗАКАЗ, будет существовать взаимосвязь, в которой каждая запись в одном объекте будет однозначно указывать на запись в другом объекте.

Взаимосвязь - «один ко многим» (между двумя типами объектов). В определенный момент времени один клиент может стать обладателем нескольких моделей автомобилей, при этом несколько клиентов не могут являться обладателями одного автомобиля. Взаимосвязь «один ко многим» можно обозначить с помощью одинарной стрелки в направлении к «одному» и двойной стрелки в направлении ко «многим», как это показано на рисунке 4.

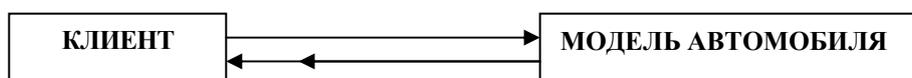


Рисунок 4 - Взаимосвязь «один ко многим»

В этом случае одной записи данных первого объекта (его часто называют родительским или основным) будет соответствовать несколько записей второго объекта (дочернего или подчиненного). Взаимосвязь «один ко многим» очень распространена при разработке реляционных баз данных. В качестве родительского объекта часто выступает справочник, а в дочернем объекте хранятся уникальные ключи для доступа к записям справочника. В нашем примере в качестве такого справочника можно представить объект КЛИЕНТ, в котором хранятся сведения обо всех клиентах. При обращении к записи для определенного клиента нам доступен список всех покупок, которые он сделал, и

сведения о которых хранятся в объекте МОДЕЛЬ АВТОМОБИЛЯ, как это показано на рисунке 4. В случае, если в дочернем объекте будут какие-то записи, для которых нет соответствующих записей в объекте КЛИЕНТ, то мы их не увидим. В этом случае говорят, что объект содержит потерянные (одинокие) записи. Это не допустимо, и в дальнейшем вы узнаете, как избегать подобных ситуаций.

Взаимосвязь «многие ко многим» (между двумя типами объектов). За определенный промежуток времени каждый продавец может обслуживать нескольких клиентов. С другой стороны, приобретая автомобили в различное время, каждый клиент может обслуживаться разными продавцами. Между объектами КЛИЕНТ и ПРОДАВЕЦ существует взаимосвязь «многие ко многим». Такая взаимосвязь обозначается двойными стрелками, как это показано на рисунке 5.

Согласно теории реляционных баз данных для хранения взаимосвязи «многие ко многим» требуются три объекта: по одному для каждой сущности и один для хранения связей между ними (промежуточный объект). Промежуточный объект будет содержать идентификаторы связанных объектов.

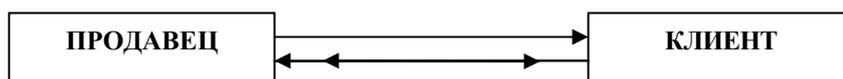


Рисунок 5 - Взаимосвязь «Многие ко многим»

Взаимосвязи между объектами являются частью концептуальной модели и должны отображаться в базе данных. Наряду с взаимосвязями между объектами существуют взаимосвязи между атрибутами объекта. Здесь также различают взаимосвязи типа «один к одному», «один ко многим» и «многие ко многим».

## 2 Проектирование базы данных

Все тонкости построения информационной модели преследуют одну единственную цель — получить хорошую базу данных. А что это такое? Существует очень простое понятие базы данных как большого по объему хранилища, в которое организация помещает все используемые данные и из которого различные пользователи могут их получать, используя различные приложения. Такая единая база данных представляется идеальным вариантом, хотя на практике это решение по различным причинам труднодостижимо. Поэтому чаще всего под базой данных понимают любой набор хранящихся в компьютере взаимосвязанных данных.

В основу проектирования БД должны быть положены представления конечных пользователей конкретной организации — концептуальные требования к системе. Именно конечный пользователь в своей работе принимает решения с учетом получаемой в результате доступа к базе данных информации. От оперативности и качества этой информации будет зависеть эффективность работы

организации. Данные, помещаемые в базу данных, также предоставляет конечный пользователь.

При рассмотрении требований конечных пользователей необходимо принимать во внимание следующее:

- база данных должна удовлетворять актуальным информационным потребностям организации. Получаемая информация должна по структуре и содержанию соответствовать решаемым задачам;
- база данных должна обеспечивать получение требуемых данных за приемлемое время, т.е. отвечать заданным требованиям производительности;
- база данных должна удовлетворять выявленным и вновь возникающим требованиям конечных пользователей;
- база данных должна легко расширяться при реорганизации и расширении предметной области;
- база данных должна легко изменяться при изменении программной и аппаратной среды.

Этапы проектирования базы данных с учетом рассмотренных выше аспектов представлены на рисунке 6.

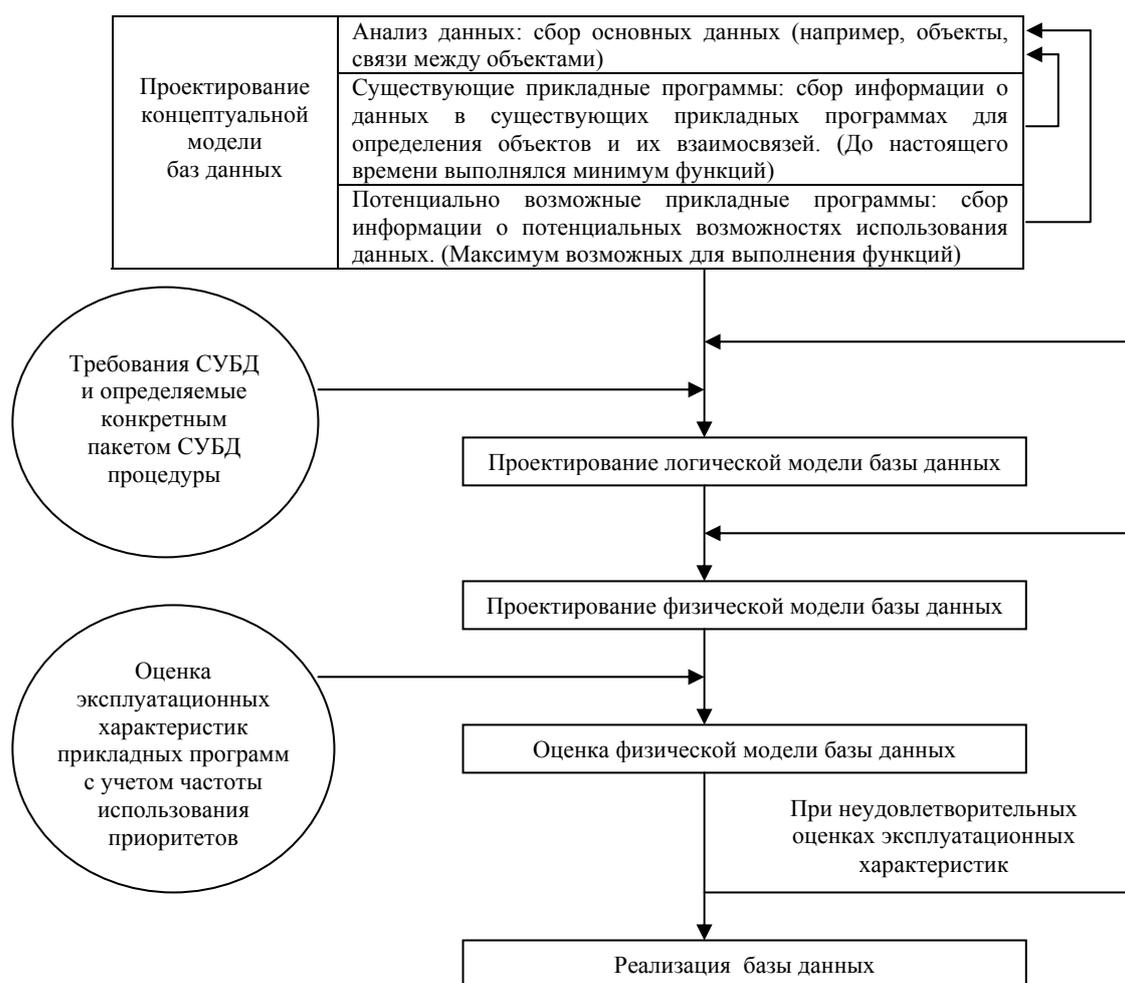


Рисунок 6 - Проектирование базы данных

В результате анализа поставленной задачи и обработки требований конечных пользователей составляется концептуальная модель. При разработке логической модели базы данных, прежде всего, необходимо решить, какая модель данных наиболее подходит для отображения конкретной концептуальной модели предметной области. Коммерческие системы управления базами данных поддерживают одну из известных моделей данных или некоторую их комбинацию. Почти что все популярные системы для персональных компьютеров поддерживают реляционную модель данных.

Отображение концептуальной модели данных на реляционную модель производится относительно просто. Каждый прямоугольник концептуальной модели отображается в одно отношение, которое отражает представление пользователя в удобном для него табличном формате. Простота отображения связана с тем, что при разработке концептуальной модели использовался реляционный подход. Рассмотрим этапы проектирования базы данных, которые должны обеспечить необходимую независимость данных и выполнение эксплуатационных требований (пожеланий пользователей).

### **2.1 Этап I Определение сущностей**

Исходя из задачи, описанной в первой главе, выделим следующие сущности:

- 1) МОДЕЛЬ;
- 2) АВТОМОБИЛЬ;
- 3) КЛИЕНТ;
- 4) ПРОДАВЕЦ;
- 5) ЗАКАЗ;
- 6) ПРОДАЖА;
- 7) СЧЕТ.

### **2.2 Этап II Определение взаимосвязей между сущностями**

Определим для включенных в модель сущностей взаимосвязи в соответствии с рекомендациями, данными в предыдущем параграфе. Полученная после этого информационная модель представлена на рисунке 7.



Рисунок 7 - Информационная модель задачи на втором этапе проектирования

Необходимо отметить, что на рисунке 7 взаимосвязь между объектами КЛИЕНТ и ЗАКАЗ рассматривается в определенный момент времени, для примера связи «один к одному». Однако, анализируя данную взаимосвязь более широко, получим, что один клиент в разное время может производить несколько заказов. С другой стороны, один заказ принадлежит только одному клиенту и поэтому между сущностями КЛИЕНТ и ЗАКАЗ установлена взаимосвязь «один ко многим».

### ***2.3 Этап III Задание первичных и альтернативных ключей, определение атрибутов сущностей***

Для каждой сущности определим атрибуты, которые мы будем хранить в БД. При этом необходимо учитывать тот факт, что при переходе от логической модели к физической модели данных может произойти усечение числа объектов. На самом деле, как правило, значительное число данных, необходимых пользователю, может быть достаточно легко подсчитано в момент вывода информации. В то же время, в связи с изменением алгоритмов расчета или исходных величин, некоторые расчетные показатели приходится записывать в БД, чтобы гарантированно обеспечить фиксацию их значений. Выбор показателей, которые обязательно следует хранить в БД, достаточно сложен. Нечасто можно найти однозначное решение этой проблемы, и в любом случае оно потребует тщательного изучения работы предприятия и анализа концептуальной модели.

Атрибуты, включаемые в состав БД для рассматриваемой модели, приведены в таблице 1.

Таблица 1 - Атрибуты и первичные ключи сущностей информационной модели

Сущность	Первичный ключ	Атрибуты
МОДЕЛЬ	Уникальный ключ модели	Уникальный ключ модели
		Наименование модели
		Наименование фирмы
		Наименование страны
		Рабочий объем двигателя
		Количество цилиндров
		Мощность
		Крутящий момент
		Наименование топлива
		Максимальная скорость
		Время разгона до 100 км/ч
		Наименование шин
		Наименование кузова
		Количество дверей
		Количество мест
		Длина
		Ширина

Продолжение таблицы 1

		Высота
		Расход топлива при 90 км/ч
		Расход топлива при 120 км/ч
		Расход топлива при городском цикле
АВТОМОБИЛЬ	Уникальный ключ автомобиля	Уникальный ключ автомобиля
		Уникальный ключ модели
		Дата выпуска
		Стоимость
КЛИЕНТ	Уникальный ключ клиента	Уникальный ключ клиента
		Наименование клиента
		Адрес
		Фамилия
		Имя
		Отчество
		Признак юридического лица
		Примечание
ПРОДАЖА	Счет	Счет
		Дата продажи
		Сумма
СЧЕТ	Номер записи	Номер записи
		Счет
		Уникальный ключ клиента
		Уникальный ключ автомобиля
		Дата выписки
		Пометка об оплате
		Сумма
ЗАКАЗ	Уникальный ключ заказа	Уникальный ключ заказа
		Уникальный ключ клиента
		Уникальный ключ модели
		Уникальный ключ продавца
ПРОДАВЕЦ	Уникальный ключ продавца	Уникальный ключ продавца
		Фамилия
		Имя
		Отчество

Информационная модель после третьего этапа проектирования приведена на рисунке 8.

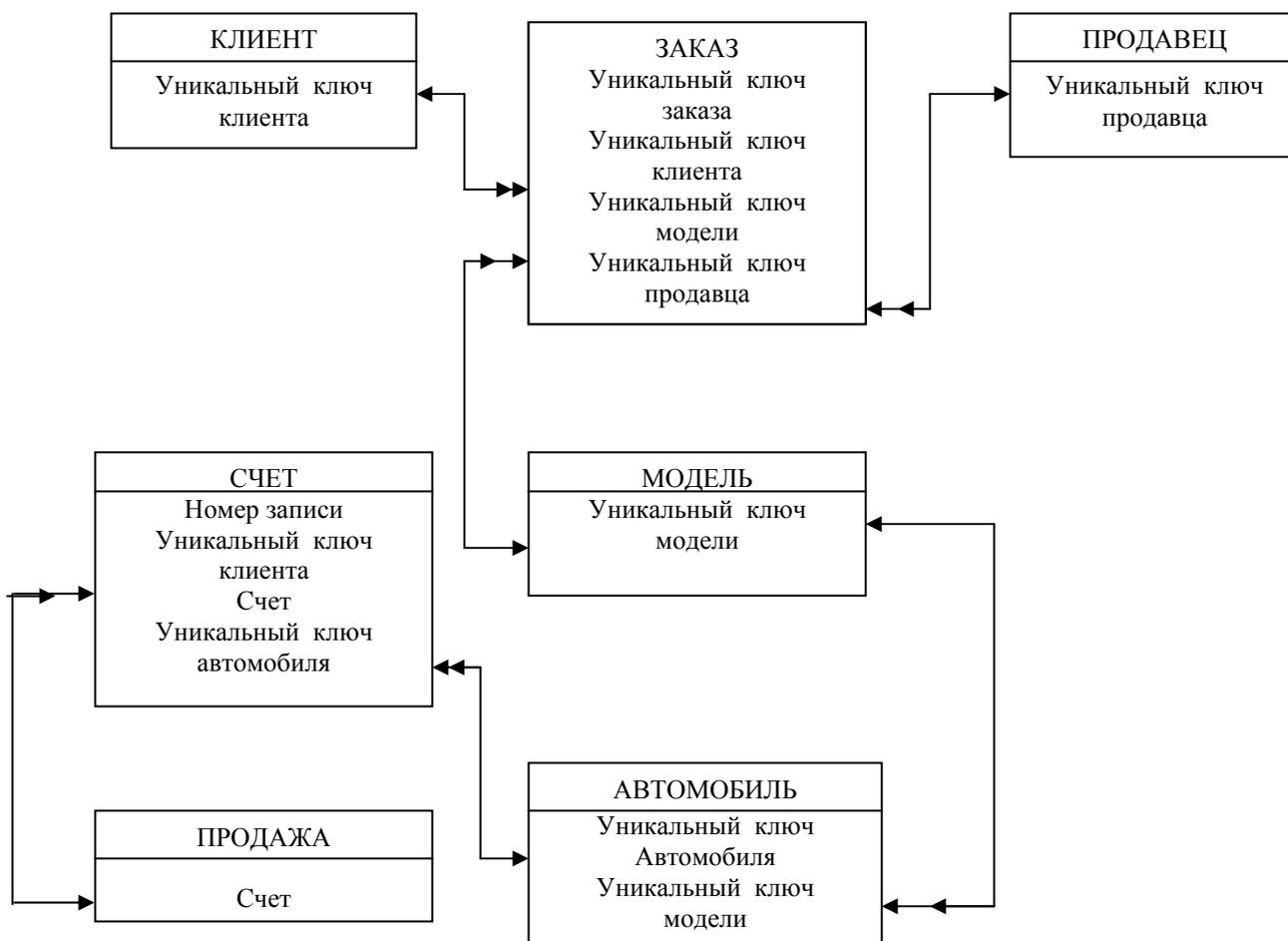


Рисунок 8 - Взаимосвязи между атрибутами сущностей

#### ***2.4 Этап IV Приведение модели к требуемому уровню нормальной формы***

Приведение модели к требуемому уровню нормальной формы является основой построения реляционной БД.

В процессе нормализации элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор таблиц обладает лучшими свойствами при включении, модификации и удалении данных, чем все остальные наборы таблиц, с помощью которых могут быть представлены те же данные. Введение нормализации отношений при разработке информационной модели обеспечивает минимальный объем физической, то есть записанной на каком-либо носителе БД и ее максимальное быстродействие, что напрямую отражается на качестве функционирования информационной системы. Нормализация информационной модели выполняется в несколько этапов.

Данные, представленные в виде двумерной таблицы, являются первой нормальной формой реляционной модели данных.

Первый этап нормализации заключается в образовании двумерной таблицы, содержащей все необходимые атрибуты информационной модели, и в выделении ключевых атрибутов. Очевидно, что полученная весьма внушительная таблица будет содержать очень разнородную информацию. В этом случае будут наблюдаться аномалии включения, обновления и удаления данных, так как при выполнении этих действий нам придется уделить внимание данным (вводить или заботиться о том, чтобы они не были стерты), которые не имеют к текущим действиям никакого отношения. Например, может наблюдаться такая парадоксальная ситуация. При включении в каталог продаж новой модели автомобиля нам сразу придется указать купившего ее клиента.

Отношение задано во второй нормальной форме, если оно является отношением в первой нормальной форме и каждый атрибут, не являющийся первичным атрибутом в этом отношении, полностью зависит от любого возможного ключа этого отношения.

Если все возможные ключи отношения содержат по одному атрибуту, то это отношение задано во второй нормальной форме, так как в этом случае все атрибуты, не являющиеся первичными, полностью зависят от возможных ключей. Если ключи состоят более чем из одного атрибута, отношение, заданное в первой нормальной форме, может не быть отношением во второй нормальной форме. Приведение отношений ко второй нормальной форме заключается в обеспечении полной функциональной зависимости всех атрибутов от ключа, за счет разбиения таблицы на несколько таблиц, в которых все имеющиеся атрибуты будут иметь полную функциональную зависимость от ключа этой таблицы. В процессе приведения модели ко второй нормальной форме в основном исключаются аномалии дублирования данных.

Отношение задано в третьей нормальной форме, если оно задано во второй нормальной форме и каждый атрибут этого отношения, не являющийся первичным, не транзитивно зависит от каждого возможного ключа этого отношения.

Транзитивная зависимость выявляет дублирование данных в одном отношении. Если  $A$ ,  $B$  и  $C$  — три атрибута одного отношения и  $C$  зависит от  $B$ , а  $B$  от  $A$ , то говорят, что  $C$  транзитивно зависит от  $A$ . Преобразование в третью нормальную форму происходит за счет разделения исходного отношения на два. Например, если все данные о моделях автомобилей и самих поступающих автомобилях хранятся в одном отношении, то для нескольких автомобилей одной модели пришлось бы многократно указывать тип кузова, количество дверей и другие технические характеристики. В этом случае технические характеристики зависят от модели автомобиля и, при наличии нескольких автомобилей одной модели, будут дублироваться. Дублирование исчезает, если из одного отношения выделить отношение, в котором будут храниться данные о моделях, и отношение, в котором будут храниться данные об автомобилях.

Давайте сформулируем основные правила, которым нужно следовать при проектировании базы данных:

- исключайте повторяющиеся группы - для каждого набора связанных атрибутов создайте отдельную таблицу и снабдите ее первичным ключом.

Выполнение этого правила автоматически приведет ко второй нормальной форме. Помимо теоретических указаний в этом правиле есть и чисто практический смысл. Представьте, что в вашем списке заказов вы указываете имена ваших клиентов. Клиент «Хитрая лиса» достаточно активен и часто делает у вас заказы. Бьемся об заклад, что найдутся считанные люди, которые в десяти упоминаниях напишут это имя абсолютно одинаково. Кто-то где-нибудь да напишет «Хитрый лис», а для СУБД это уже другой клиент. Поэтому гораздо лучше вести список своих клиентов в отдельной таблице, а в списке заказов использовать только присвоенные им уникальные идентификаторы;

- исключайте избыточные данные - если атрибут зависит только от части составного ключа, переместите атрибут в отдельную таблицу. Это правило помогает избежать потери одних данных при удалении каких-то других. Везде, где возможно использование идентификаторов вместо описания, выносите в отдельную таблицу список идентификаторов с пояснениями к ним.

- исключайте столбцы, которые не зависят от ключа - если атрибуты не вносят свою лепту в описание ключа, переместите их в отдельную таблицу.

С учетом выше изложенного в нашей модели необходимо видоизменить список атрибутов сущности МОДЕЛЬ и добавить такие новые сущности, как ТОПЛИВО, ШИНЫ, КУЗОВ, ФИРМА, СТРАНА.

В основном изменения в модели связаны с введением искусственных атрибутов, которые в виде кодов участвуют в отношениях вместо естественных атрибутов (вид топлива, марка шин и т.п.). К необходимости введения в модель искусственных атрибутов мы пришли в процессе нормализации. В общем случае мы рекомендуем использовать вместо естественных атрибутов коды в следующих случаях:

- в предметной области может наблюдаться синонимия, то есть естественный атрибут отношения не обладает свойством уникальности. Например, среди сотрудников фирмы могут быть однофамильцы или даже полные тезки. В этом случае решить проблему помогает уникальный табельный номер;

- если отношение участвует во многих связях, то для их отображения создается несколько таблиц, в каждой из которых повторяется идентификатор отношения. Для того, чтобы не использовать во всех таблицах длинный естественный атрибут объекта, можно применять более короткий код. Это также будет способствовать повышению быстродействия вашей системы;

- если естественный атрибут может изменяться во времени (например, фамилия), то это может вызвать очень большие сложности при эксплуатации системы. Представьте, что ваш продавец поменял фамилию. Что будет с данными, которые, привязаны к его фамилии? Использование неизменяемого кода (табельного номера) позволит избежать этих сложностей.

Атрибуты, включаемые в измененные или добавленные в модель сущности, приведены в таблице 2.

Таблица 2 - Атрибуты и первичные ключи измененных или добавленных сущностей информационной модели

Сущность	Первичный ключ	Атрибуты
МОДЕЛЬ	Уникальный ключ модели	Уникальный ключ модели Наименование модели Уникальный ключ фирмы Наименование страны Рабочий объем двигателя Количество цилиндров Мощность Крутящий момент Уникальный ключ топлива Максимальная скорость Время разгона до 100 км/ч Уникальный ключ шин Уникальный ключ кузова Количество дверей Количество мест Длина Ширина Высота Расход топлива при 90 км/ч Расход топлива при 120 км/ч Расход топлива при городском цикле
ТОПЛИВО	Уникальный ключ топлива	Уникальный ключ топлива Наименование топлива
ШИНЫ	Уникальный ключ шин	Уникальный ключ шин Наименование шин
КУЗОВ	Уникальный ключ кузова	Уникальный ключ кузова Наименование кузова
ФИРМА	Уникальный ключ фирмы	Уникальный ключ фирмы Наименование фирмы Уникальный ключ страны
СТРАНА	Уникальный ключ страны	Уникальный ключ страны Наименование страны

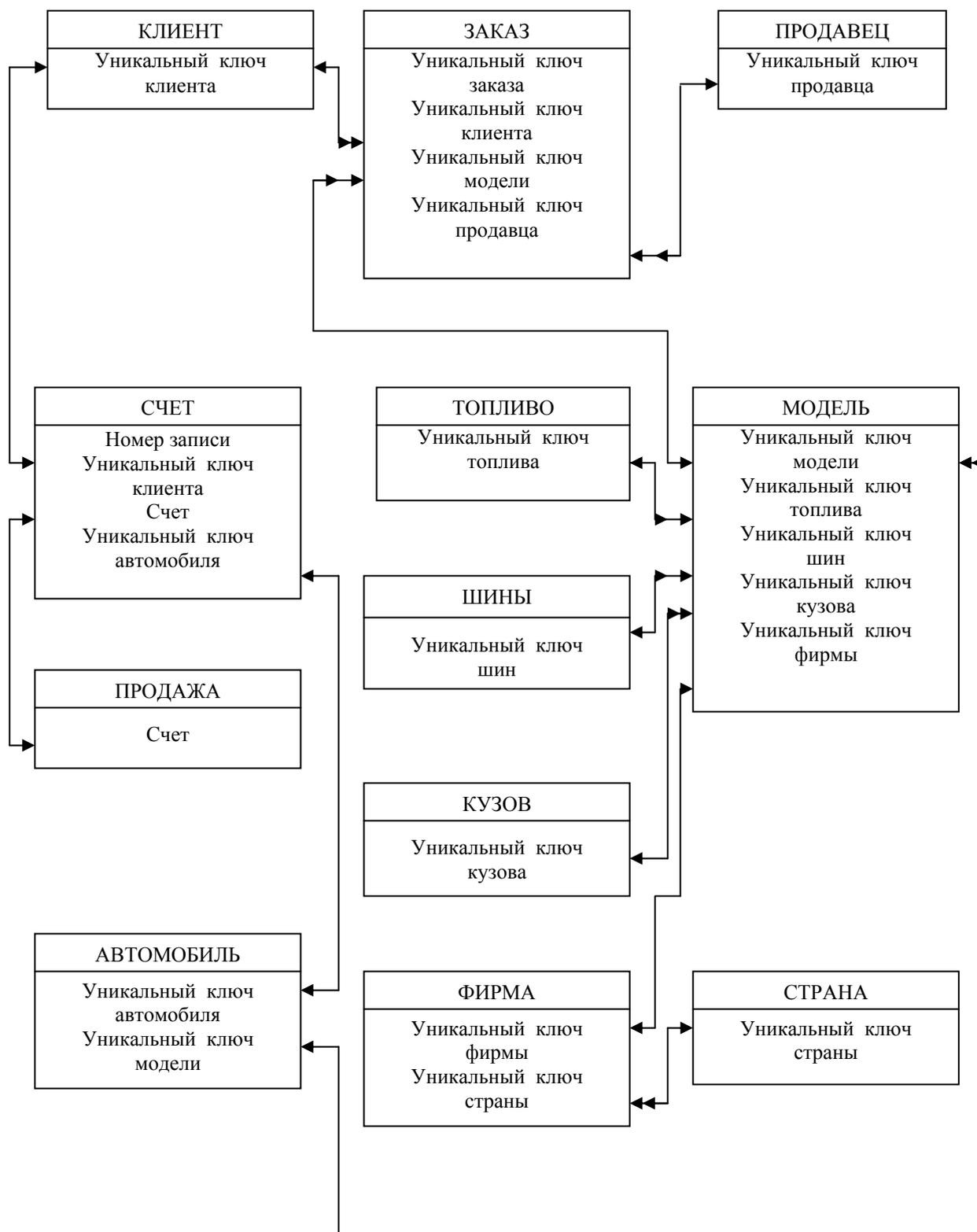


Рисунок 9 - Взаимосвязи между атрибутами сущностей после нормализации модели

## ***2.5 Этап V Физическое описание модели***

На этом этапе мы должны составить проекты таблиц, которые будут в дальнейшем реализовываться в конкретной СУБД.

На этапе физического проектирования мы должны задуматься о такой серьезной проблеме, как обеспечение безошибочности и точности информации, хранящейся в БД. Это называется обеспечением целостности базы данных.

Обеспечением целостности базы данных называется система мер, направленных на поддержание правильности данных в базе в любой момент времени.

Мы привыкли доверять достоверности данных, помещаемых в печатных изданиях. При подготовке книги все данные, помещаемые в нее, проверяют несколько редакторов. Они же стараются сделать так, чтобы книга была написана литературным языком и соответствовала неким нормам, с которыми читатели подходят к книгам различного жанра. Удивительно было бы при чтении детектива встретить ссылки на научные источники в конце страницы. При общении человека с компьютером наблюдается некий стойкий феномен. Мы приравниваем компьютер к книге из издательства с самой известной маркой. Мы априори верим всем данным, которые выдает нам это славное устройство. Мы не хотим ничего слышать о программных ошибках, сбоях и неправильном вводе данных.

Если же трезво взглянуть на проблему, ситуация начинает выглядеть совершенно иначе. При подготовке книги затраты на ее редактирование составляют незначительную часть от общих затрат на ее «выпуск в свет». В системах обработки данных наоборот. Затраты на проверку и поддержание достоверности данных могут составлять значительную часть от общих эксплуатационных затрат. Например, в транспортных предприятиях для контроля правильности ввода данных с путевой документации практикуется параллельный ввод одних и тех же данных несколькими операторами. Считается, что вероятность совершения одной и той же ошибки в этом случае будет крайне мала и простое сравнение результатов ввода различных операторов поможет получить безошибочные данные.

Естественно появляется соблазн избежать дорогостоящей верификации данных. Как сказал один рабочий судовой верфи другому, объясняя, почему он не заделал дыру в корпусе корабля: «Ее никто не увидит, она же под водой». Ошибки в данных неприятны тем, что они остаются незамеченными до тех пор, пока не приведут к тяжелым последствиям, если только вы не позаботитесь обнаружить эти ошибки раньше. Достаточно убедительный довод, чтобы заранее воспользоваться предоставляемыми СУБД мерами для блокирования появления возможных ошибок в разрабатываемой базе данных.

В СУБД целостность данных обеспечивается набором специальных предложений, называемых ограничениями целостности.

Ограничения целостности - это набор определенных правил, которые устанавливают допустимость данных и связей между ними.

Ограничения целостности в большинстве случаев определяются особенностями предметной области. Например, мощность двигателя серийного легкового автомобиля вряд ли может быть ниже 30 л. с.

Ограничения целостности могут относиться к разным объектам БД: атрибутам (полям), записям, отношениям, связям между ними и т.п. Для полей могут использоваться следующие виды ограничений:

- тип и формат поля автоматически допускают ввод только данных определенного типа. Выбор типа поля Date в формате ДД.ММ.ГГ позволит пользователю ввести только шесть чисел. При этом первая пара цифр не сможет превысить в лучшем случае значения 31, а вторая — 12;

- задание диапазона значений, как правило, используется для числовых полей. Диапазон допустимых значений может быть ограничен с двух сторон (закрытый диапазон), а может с какой-то одной: верхней или нижней (открытый диапазон);

- недопустимость пустого поля позволяет избежать появления в БД «ничейных» записей, в которых пропущены какие-либо обязательные атрибуты;

- задание списка значений позволяет избежать излишнего разнообразия данных, если его можно ограничить. Например, для указания типа кузова мы можем ограничить фантазию пользователя только общепринятыми названиями: седан, кабриолет и т. д.;

- проверка на уникальность значения какого-то поля позволяет избежать записей-дубликатов. Вряд ли будет удобно в справочнике клиентов иметь несколько записей для одного и того же лица.

Для реализации ограничений целостности, имеющих отношение к записи, таблицам или связям между ними, в СУБД используются триггеры.

### **3 Варианты заданий для самостоятельной работы**

**Вариант 1.** Разработать прикладное программное обеспечение деятельности депо по ремонту пассажирских вагонов. Депо выполняет несколько видов ремонта. Деповской ремонт – после пробега вагоном 450 тыс. км или два года эксплуатации (что наступит раньше). ТО-2 – подготовка вагона к зимним или летним условиям эксплуатации. ТО-3 – текущее обслуживание – после пробега 150 тыс. км или один год эксплуатации. Текущий ремонт – круглосуточно, при котором ремонтируются вагоны всех дорог России. Основные причины поступления вагона в текущий ремонт: неисправность колесной пары, неисправность буксового узла и т.д. Каждый вагон имеет уникальный номер. Тип вагона также имеет значение при ремонте: купейный, СВ, плацкартный, почтовый, багажный. Каждый вагон приписан к дирекции по обслуживанию пассажиров (ДОП-1, ДОП-2, ДОП-3 и т.д.). Текущий ремонт выполняют ремонтные бригады в четыре смены. Для выполнения остальных ремонтов привлекается, как правило, одна бригада. За высокое качество ремонта члены бригады получают премию.

**Вариант 2.** Разработать прикладное программное обеспечение деятельности ремонтно-эксплуатационного локомотивного депо. Депо выполняет несколько видов ремонта: текущий ремонт (ТР), средний ремонт (СР), техническое обслуживание (ТО) и внеплановый ремонт. При внеплановом ремонте локомотив снимается с рейса и заменяется резервным, поэтому сроки внепланового ремонта должны быть минимальными, а сам ремонт проводится порой в четыре смены. Каждый локомотив имеет уникальный номер и приписан к определенному локомотивному депо. Технология ремонта зависит от типа локомотива (пассажирский или грузовой). Для выполнения первых трех видов ремонта привлекается, как правило, одна бригада. За высокое качество выполненных работ члены бригады получают дополнительное вознаграждение (квартальная премия, месячная премия, 13 и 14 зарплата). За переработку (сверхурочные) также выплачиваются дополнительные суммы.

**Вариант 3.** Разработать прикладное программное обеспечение деятельности судоходной компании «Балтика». Эта крупная компания занимается перевозками грузов между континентами. В ее собственности несколько десятков судов различного класса и грузоподъемности. К услугам этой компании обращаются тысячи клиентов из различных стран мира. На судне может находиться несколько партий грузов для различных грузополучателей из различных стран и городов. Одна партия груза может состоять из нескольких разновидностей грузов. У одной партии груза может быть только один отправитель и только один получатель. Судно следует по маршруту. Маршрут разрабатывается главным менеджером компании и проходит через несколько портов. В очередном порту назначения производится лишь частичная погрузка и выгрузка грузов, и судно следует дальше.

**Вариант 4.** Разработать прикладное программное обеспечение деятельности учреждения юстиции. По существующему законодательству на это учреждение возложена обязанность регистрации прав юридических и физических лиц на недвижимое имущество (здания, квартиры, земельные участки). В этом задании вам необходимо разработать лишь часть программного комплекса, обеспечивающего регистрацию прав граждан на квартиры. Имейте в виду - в здании несколько квартир! В одной квартире несколько собственников, причем в базе данных должна храниться история перехода квартиры от одних собственников к другим. Кадастровый номер здания однозначно определяет его среди других зданий города. Смело используйте его в качестве первичного ключа таблицы зданий.

**Вариант 5.** Разработать прикладное программное обеспечение деятельности малого научно-внедренческого предприятия «Квадро». Это предприятие занимается прокладкой компьютерных сетей и разработкой программных комплексов для организаций нашего города. Численность работников в «Квадро» – примерно 80 человек. Одновременно находится в разработке до 30 проектов. Один разработчик может участвовать в нескольких проектах одновременно, но зарплата его от этого не зависит. Одна организация

может заказать в «Квадро» несколько разработок. Стоимость каждого проекта оговаривается отдельно. При досрочном выполнении работы заказчик перечисляет научно-внедренческому предприятию определенный, заранее оговоренный, процент премии.

**Вариант 6.** Разработать прикладное программное обеспечение деятельности ООО «Киновидеопрокат». Это предприятие фактически контролирует демонстрацию кинофильмов в кинотеатрах города. Отдел маркетинга, изучив ситуацию на рынке кинофильмов, принимает решение о покупке тех или иных кинолент. Отдел закупок претворяет эти решения в жизнь, причем лента может быть куплена как у производителя, так и у посредника. Отдел аренды киновидеопроката сдает закупленные фильмы кинотеатрам города в аренду. Так как всегда закупается только одна копия фильма, он не может демонстрироваться одновременно в нескольких кинотеатрах. У одного поставщика может быть куплено несколько фильмов. Также несколько лент может быть в аренде у одного кинотеатра одновременно.

**Вариант 7.** Разработать прикладное программное обеспечение деятельности предприятия LADA-сервис. Эта крупная компания занимается продажей автомобилей марки ВАЗ в нашем городе. Она имеет несколько филиалов в разных районах. Автомобиль может быть продан как со склада компании, так и на заказ с завода-изготовителя по предоплате. Покупатель может заказать модель, цвет, тюнинг и оговорить срок поставки заказанного автомобиля. Одновременно с новыми авто на площадках компании имеется большой выбор подержанных автомобилей, как отечественных, так и иностранных. Покупателем может быть как физическое лицо, так и организация. В первом случае расчет наличными деньгами, во втором – через банк. Расчет производится в рублях.

**Вариант 8.** Разработать прикладное программное обеспечение торговой-посреднической фирмы «Столица». Бизнес этого предприятия предельно прост: «покупай дешевле – продавай дороже» или состыкуй продавца и покупателя и получи «комиссионные». Основной упор фирма делает на закупки продуктов питания в других регионах страны и за рубежом – там, где они производятся и стоят дешевле, чем в нашем регионе. Часть продукции может быть закуплена и у местных продавцов. В этом случае фирма получает прибыль за счет того, что крупные партии товара стоят дешевле, чем мелкие. Имейте в виду, что товар не может быть продан дешевле, чем он куплен.

**Вариант 9.** Разработать прикладное программное обеспечение деятельности отдела гарантийного ремонта товаров фирмы «Народная торговая компания». Это предприятие – лидер продаж кондиционеров, телевизоров и другой бытовой техники в городе. Хорошо известно, что техника часто выходит из строя, причем уже в период гарантийного срока, а в этом случае продавец товара должен бесплатно отремонтировать его. Ежедневно в отдел гарантийного ремонта обращаются несколько десятков человек, купивших

технику в этой компании. Вы, скорее всего, также побывали в отделе гарантийного ремонта, что очень поможет Вам при разработке программного обеспечения.

**Вариант 10.** Разработать прикладное программное обеспечение деятельности отдела учета личного состава батальона железнодорожных войск. Это фактически отдел кадров воинской части. Батальон расквартирован на отдельной территории. В батальоне несколько рот, в каждой роте несколько взводов, каждый взвод состоит из трех отделений. В мирное время батальон занимается изучением техники и поддержанием ее в рабочем состоянии. Часть технических ресурсов «законсервирована». Поддержание такой техники в отличном состоянии также входит в обязанности личного состава батальона. В настоящее время существует три вида службы: срочная, сверхсрочная и по контракту. Каждый офицер части имеет удостоверение личности, которое заменяет паспорт, а военнослужащий срочной службы – военный билет.

**Вариант 11.** Разработать прикладное программное обеспечение деятельности отдела учета домовладений «Бюро технической инвентаризации». В состав домовладения входят земельный участок и несколько строений. Их называют литерами: жилой дом, летняя кухня, гараж, колодец, забор и т.д. Для жилого дома составляется экспликация, в которой указываются данные по каждому помещению. Экспликация может быть составлена и для других крупных строений. В ее состав входят: номер квартиры, номер помещения на плане, этаж, назначение помещения, площадь, высота и т.п. Для вспомогательных литер (забор, тротуар, колодец) экспликация не заполняется.

**Вариант 12.** Разработать прикладное программное обеспечение деятельности отдела кадров университета. В отделе кадров университета находятся данные всех сотрудников: от преподавателя до ректора, и их трудовой деятельности. Наряду с такими данными, как специальность сотрудника и занимаемая должность, обязательно учитываются сведения об ученой степени сотрудника (кандидат наук, доктор) и ученом звании (доцент, профессор). Также в отделе кадров хранится информация о трудовой деятельности сотрудника: о предыдущих местах работы, сроке работы и предприятии. Отдел кадров занимается подготовкой трудовых договоров с преподавателями после избрания их по конкурсу на очередной срок. Также в его ведении находятся сведения о наложении взысканий на сотрудников и их поощрениях. Взыскания в трудовую книжку не заносятся, а хранятся в электронном виде.

**Вариант 13.** Разработать прикладное программное обеспечение деятельности биржи труда. На биржу труда обращаются люди, не сумевшие самостоятельно устроиться на работу, но все ещё желающие найти работу по специальности. Организации предоставляют бирже список свободных вакансий. Каждый обратившийся ставится на учет. В день обращения ему предлагается список вакансий. Если свободных вакансий нет или они не

устраивают того, кто ищет работу, то ему будет предложено подождать пока подходящее свободное место работы не появится. Зарегистрированный на бирже получает пособие по безработице до тех пор, пока не будет трудоустроен. После этого его данные переносятся в архив, и выплата ему пособия прекращается.

**Вариант 14.** Разработать прикладное программное обеспечение деятельности отдела учета квартир «Бюро технической инвентаризации». В нашем городе имеется 6000 зданий, в которых расположено 199000 квартир. Помещений в этих квартирах – 1 500 000 шт. Кадастровый номер здания является уникальным. Используйте его в качестве простого первичного ключа таблицы зданий. Можете работать и с составным первичным ключом (адресом здания), но в данном случае – это не лучший вариант.

**Вариант 15.** Разработать прикладное программное обеспечение деятельности аптечного склада. Аптечный склад занимается оптовой продажей лекарств больницам и аптекам города. В его ассортименте – тысячи наименований лекарств, а также различных аптечных принадлежностей (градусники, шприцы, бинты и т.д.) Возможна продажа лишь тех лекарств, которые одобрены Минздравом РФ, т.е. имеют регистрационный номер Минздрава РФ. Поступающие лекарства сопровождаются документами – приходными накладными ведомостями. Покупатель получает счет-фактуру на выбранный товар, оплачивает сумму, указанную в ней, и после оплаты получает выходную накладную ведомость, по которой получает выбранный товар.

**Вариант 16.** Разработать прикладное программное обеспечение деятельности отдела учета нежилых помещений «Бюро технической инвентаризации». В 2000 г. в нашем городе была проведена сплошная инвентаризация, в ходе которой было выявлено 16000 нежилых помещений. Это магазины (встроенные, пристроенные и отдельно стоящие), офисы, учреждения, мастерские и т. д. Помещение может состоять из отдельных частей (кабинет, проходная, коридор). Составных частей помещений выявлено 265 000 штук. В одном здании может быть несколько помещений, а помещение может состоять из нескольких частей. Любое здание имеет уникальный кадастровый номер, однозначно определяющий его положение в городе.

**Вариант 17.** Разработать прикладное программное обеспечение деятельности отдела учета налогообложения физических лиц городской налоговой инспекции. По существующему законодательству некоторые категории граждан должны представить в налоговую инспекцию декларацию о полученных доходах. Налоговый инспектор должен проверить ее, занести в базу данных и выписать платежное извещение на уплату подоходного налога с доходов физического лица. Лица, заполнившие декларацию, должны доплатить в бюджет некоторую сумму. С 2002 г. шкала налогообложения – линейная (13 % со всей заработанной суммы за год), но лицам, затратившим средства на

обучение, покупку лекарств и т.д., из бюджета должна быть возвращена некоторая сумма, рассчитываемая по специальной методике.

**Вариант 18.** Разработать прикладное программное обеспечение деятельности телеателье «Спектр». Эта организация занимается послегарантийным ремонтом теле-радиоаппаратуры отечественного и импортного производства. Клиенты этого телеателье – жители и организации нашего города и близлежащих сел. Расчет с физическими лицами ведется наличными, а с организациями – через банк. Выдача отремонтированной техники производится после полной оплаты выполненного ремонта. Отремонтированное изделие получает гарантию. Если в течение гарантийного срока произойдет поломка изделия, то повторный ремонт выполняется за счет телеателье. Если брак допустил мастер, то часть суммы удерживается из его зарплаты. Клиент, обратившийся к услугам ателье несколько раз с ремонтом разной аппаратуры, получает дисконтную карту, дающую право на скидку при ремонте очередного изделия.

**Вариант 19.** Разработать прикладное программное обеспечение деятельности отдела заселения муниципальных общежитий администрации города. В ведении администрации города находится несколько десятков общежитий. Раньше они принадлежали предприятиям города, а теперь, после банкротства предприятий, все эти общежития переданы муниципальным властям. В последние годы бесплатные квартиры гражданам города практически не предоставляются, а количество малоимущих жителей, нуждающихся в жилье, растет. Хоть как-то улучшить жилищные условия этой категории граждан позволяет наличие муниципальных общежитий. Получить четкую картину их заселения позволит данное программное обеспечение. База данных отдела содержит информацию об общежитиях, комнатах общежитий и проживающих.

**Вариант 20.** Разработать прикладное программное обеспечение деятельности Государственной автомобильной инспекции по безопасности дорожного движения города. База данных ГИБДД содержит сведения обо всех транспортных средствах города и их владельцах. В нее заносятся сведения о технических осмотрах транспортных средств и об угонах. Описание угнанного автомобиля не удаляется из базы данных. Истории переходов транспортных средств от одних владельцев к другим не накапливаются. Сведения об автомобилях, снятых с учета, навсегда удаляются из базы данных.

**Вариант 21.** Разработать прикладное программное обеспечение для ведения реестра имущества университетского городка. В состав имущества входит несколько зданий. В зданиях располагаются аудитории, кафедры, лаборатории, вычислительные центры, деканаты и т.д. Любое помещение университета относится к какому-либо подразделению. Все движимое имущество, находящееся в помещении, состоит на балансе материально ответственного лица. Каждая аудитория закреплена за определенной кафедрой

университета, так же в ведении кафедр находятся и лаборатории. По истечении определенного времени имущество, находящееся в помещениях, списывается. Архив списанного имущества не ведется.

**Вариант 22.** Разработать прикладное программное обеспечение деятельности туристической компании «Вояж». Эта компания формирует туристические группы для заграничных поездок и обеспечивает им полную поддержку на маршруте. Количество туристов в группе заранее известно и ограничено. Маршрут группы может пролегать через несколько городов страны назначения. Экскурсии одновременно в несколько стран не проводятся. При обращении в «Вояж» группы из нескольких человек компания предоставляет скидку, которая зависит от количества туристов в группе. Вместе с группой следует представитель компании, который несет полную ответственность за качество услуг, предоставляемых компанией. При возникновении каких-либо неудобств на маршруте, возникших по вине компании, турист получает назад заранее оговоренную в контракте сумму.

**Вариант 23.** Разработать прикладное программное обеспечение деятельности регистратуры ведомственной поликлиники «Эскулап». Работники регистратуры организуют запись пациентов на прием к врачам поликлиники. Так как поликлиника ведомственная, медицинское обслуживание работников предприятия – бесплатное (за счет средств предприятия). «Посторонние» пациенты также могут воспользоваться услугами поликлиники, полностью оплатив затраты на лечение. Определение стоимости лечения и выдача платежных документов для таких больных входит в круг обязанностей работников регистратуры. Врач ведет прием всегда в одном кабинете. Приемные дни занесены в расписание работы поликлиники. На каждого пациента в регистратуре заводится карточка. В начале приема карточки больных, записавшихся на прием, доставляются работником регистратуры в кабинет врача.

**Вариант 24.** Разработать прикладное программное обеспечение деятельности рекламного агентства «Rapid». В собственности этого агентства находится примерно около сотни рекламных щитов, расположенных по всему городу. Установка их согласована с администрацией города, и все необходимые формальности выполнены. На этих щитах может быть размещена реклама по заказу любой организации города. Срок размещения, стоимость аренды щита и стоимость изготовления самой рекламы – договорные. Одна организация может арендовать несколько рекламных щитов. Один щит не сдается в аренду нескольким арендаторам, так как является неделимой рекламной единицей. Договор размещения рекламы может быть продлен по взаимной договоренности сторон.

**Вариант 25.** Разработать прикладное программное обеспечение деятельности ООО «Центр оценки и продажи недвижимости». Одним из источников прибыли этой организации является покупка и продажа квартир.

Центр оценки имеет большой штат специалистов, позволяющий этой организации проводить сделки купли-продажи на высоком профессиональном уровне. Владелец квартиры, желающий ее продать, заключает договор с Центром, в котором указывается сумма, срок продажи и процент отчислений в пользу Центра оценки и продажи недвижимости в случае успешного проведения сделки. Один клиент может заключить с Центром более одного договора купли-продажи одновременно, если он владеет несколькими квартирами. Обмен квартир специалисты центра непосредственно не производят. Для этих целей используется вариант купли-продажи.

### **Список литературы**

- 1 Бакаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2002. – СПб.: БХВ-Петербург, 2002. – 468 с.
- 2 Бакаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2006. – СПб.: БХВ-Петербург, 2006. – 720 с.
- 3 Бакаревич Ю.Б., Пушкина Н.В. MS Access 2006 за 30 занятий. – СПб.: БХВ-Петербург, 2006. – 510 с.
- 4 Винтер Р. Microsoft Access 2005: Справочник. – СПб.: Питер, 2008. – 416 с.
- 5 Информатика. Базовый курс / Под ред. С.В. Симоновича. – СПб.: Питер, 2007. – 640 с.
- 6 Леонтьев Ю. Microsoft Office 2005: Краткий курс. – СПб.: Питер, 2005. – 288 с.
- 7 Microsoft Access 2005. Шаг за шагом: Практическое пособие / Пер. с англ. – М.: ЭКОМ, 2005. – 352 с.

ХМЕЛЕВ СТАНИСЛАВ АНДРЕЕВИЧ

# ***ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ***

Методические указания

к выполнению практических и самостоятельных работ  
для студентов очной формы обучения специальности 080502

Часть II

Редактор Н.М. Устюгова

---

Подписано к печати	Формат 60x84 1/16	Бумага тип. № 1
Печать трафаретная	Усл. печ. л. 2,0	Уч. - изд. л. 2,0
Заказ	Тираж 60	Цена свободная

---

Редакционно-издательский центр КГУ.  
640669, г. Курган, ул. Гоголя, 25.  
Курганский государственный университет.