

ГЛАВА I. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ В ОБЛАСТИ РАЗРАБОТКИ БАЗ ДАННЫХ

1.1 Основная терминология теории баз данных

Базу данных можно охарактеризовать как совместно используемый набор логически связанных данных, предназначенный для удовлетворения информационных потребностей организации [1].

Подобной точки зрения придерживается и К.Дж.Дейт, определяя базу данных как «некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия» [2].

Поддержку и доступ пользователей к базе данных осуществляется с помощью специального программного обеспечения – системы управления базами данных.

Система управления базами данных (СУБД) – программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ [1].

Также система управления базами данных позволяет:

- Управлять данными во внешней памяти;
- Управлять буферами оперативной памяти;
- Структурировать и восстанавливать базу данных после сбоев;

Концептуальная модель базы данных описывает сущности и связи между ними, независимо от системы управления базами данных.

По Оскеркову В.С., сущность – это объект реального мира, который может существовать независимо. Сущность имеет экземпляры, которые должны отличаться друг от друга значениями атрибутов [3].

Атрибут – свойство сущности [3].

Между сущностями существует связь. Связь представляет собой взаимодействие между сущностями [3]. Возможны связи на основе отношений:

- Один-к-одному;
- Один-ко-многим;
- Многие-ко-многим.

Также в справочнике [8] связь определяется, как соединение двух таблиц, с помощью общего поля – идентификатора записи.

Организация данных рассматривается с позиции той или иной модели данных. Благодаря модели данных могут быть представлены объекты базы данных и связи между ними.

Модель данных – это совокупность структур данных и операций из обработки [4].

Всего различают 3 типа модели данных:

- Иерархическая;
- Сетевая;
- Реляционная;

Модель иерархического типа представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое по структуре дерево [4].

Сетевая модель – расширение иерархического подхода. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков [5].

Реляционная модель характеризуется простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных [4]. В реляционной модели данных отношения представляются в виде совокупности таблиц связанных между собой.

Для управления и изменения таблиц (преобразованных сущностей) используются хранимые процедуры и триггеры.

Хранимая процедура – это специальный вид процедуры, который выполняется сервером баз данных. Хранимые процедуры пишутся на процедурном языке, который зависит от конкретной СУБД. Они могут вызывать друг друга, читать и изменять данные в таблицах, и их можно вызвать из клиентского приложения, работающего с базой данных [6].

Триггер – это специальный тип хранимой процедуры, который автоматически выполняется в СУБД и сопровождает процесс модификации указанной таблицы [9].

1.2 Анализ предметной области

Нумизматика – это наука, которая изучает отдельные монеты, историю их чеканки и обращения. Помимо этого, в понятие «нумизматика» включается и изучение банкнот, жетонов, наград и значков. Монеты позволяют узнать об экономической истории общества того времени, в котором она была изготовлена.

С ростом популярности этой науки появились магазины нумизматики, которые обладают коллекциями древних и редких монет и банкнот, доступных потребителю для получения достоверной информации и последующей покупки. Всё это способствовало появлению интернет-магазинов, благодаря которым можно получить информацию, хранящуюся в базе данных, в считанные секунды.

Чтобы разработать правильно-структурированную базу данных необходимо как можно больше информации о предметной области.

Информация о покупателях должна быть полной. База данных должна обеспечить выполнение следующих действий:

- Хранение данных о зарегистрированных пользователях;
- Обновление данных пользователей;
- Учет новых заказов и отслеживание активных;

У каждого покупателя должны быть контактные данные. Также, при регистрации пользователь обязан вносить следующие данные:

- Имя;
- Фамилия;
- Отчество;
- E-mail;
- Телефон;
- Адрес.

Каждый интернет-магазин имеет товар. Для создания карточки с товаром необходимо знать его описание и количество. Также таблица «Товары» разделяется на несколько типов:

- Монеты;
- Банкноты;
- Награды.

Посредством языка запросов из базы данных интернет-магазина осуществляется:

- Выборка товаров по типу, стране, году изготовления;
- Проверка наличия товара.

В свою очередь, администратор сайта может вносить некоторые изменения в базу данных, а в частности: добавлять товар, изменять информацию о нем и удалять его.

Каждый тип товара имеет свои характеристики, которые должны будут учитываться в базе данных.

Монеты имеют такие характеристики:

- Страну;
- Год чеканки;
- Номинал;

- Металл;
- Диаметр;
- Вес;
- Тираж.

Банкноты, в свою очередь имеют схожие характеристики:

- Страну;
- Год печати;
- Номинал.

Так как монету или банкноту можно подделать, каждая единица товара такого типа должна иметь дополнительную характеристику, которая зависит от экспертизы – гарантия подлинности.

Тип «награды», включающие в себя жетоны, медали и значки, немного отличается от предыдущих двух:

- Страна;
- Металл;
- Вес;
- Год;
- Диаметр.

Товары этого типа не имеют такие характеристики, как гарантия подлинности и номинал.

Проанализировав предметную область «Магазин нумизматики», актуально разработать базу данных, предназначенную для хранения и обновления информации о купленных, заказанных товарах, а также о пользователях, покупающих товар. Разработанный продукт позволит автоматизировать взаимодействие покупателя с продавцом.

1.3 Инфологическое проектирование

На этапе инфологического проектирования базы данных строится инфологическая модель предметной области, которая должна отражать смысл взаимосвязей объектов предметной области.

Диаграмма, отражающая связи объектов предметной области, называется диаграммой ER-типа (Entry – сущность, Relationship – связь).

Определение основных сущностей базы данных «Магазин нумизматики»:

1. «Пользователь»
2. «Заказы»
3. «Монеты»
4. «Банкноты»
5. «Награды»
6. «Товары»

Рассмотрение сущности «Пользователь» (рис. 1.1).



Рис.1.1 Сущность «ПОЛЬЗОВАТЕЛЬ»

Таким образом, сущность «Пользователь» обладает статическим атрибутом «Логин» и остальными динамическими атрибутами.

- Статический атрибут – постоянный атрибут, неизменяемый.
- Динамический атрибут – атрибут, который может изменяться.

Также сущность обладает составным атрибутом «Адрес» (рис. 1.2).

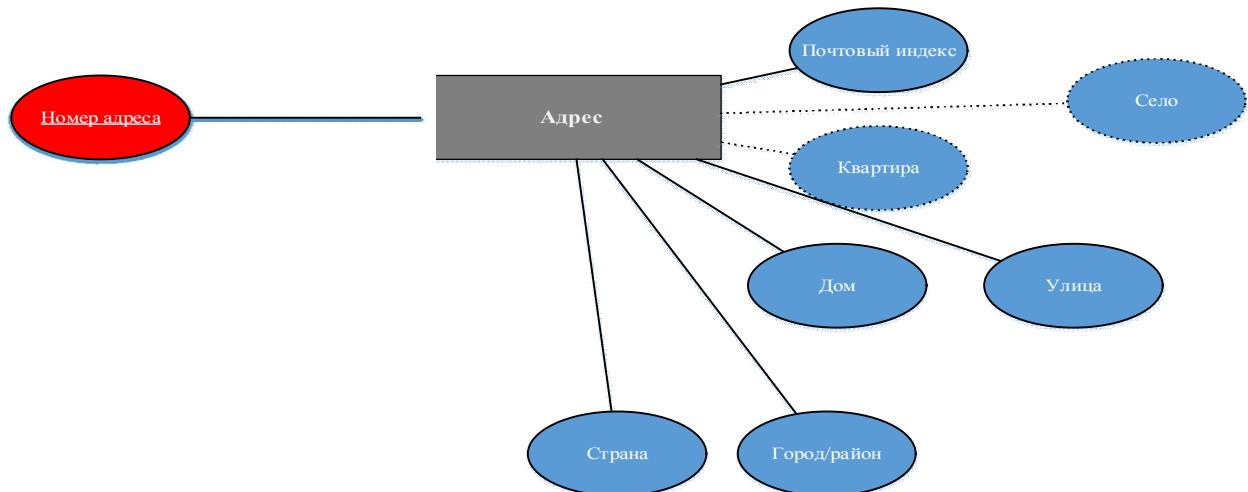


Рис. 1.2 Составной атрибут «АДРЕС»

Далее следует рассмотреть сущность «Заказы» (рис. 1.3).

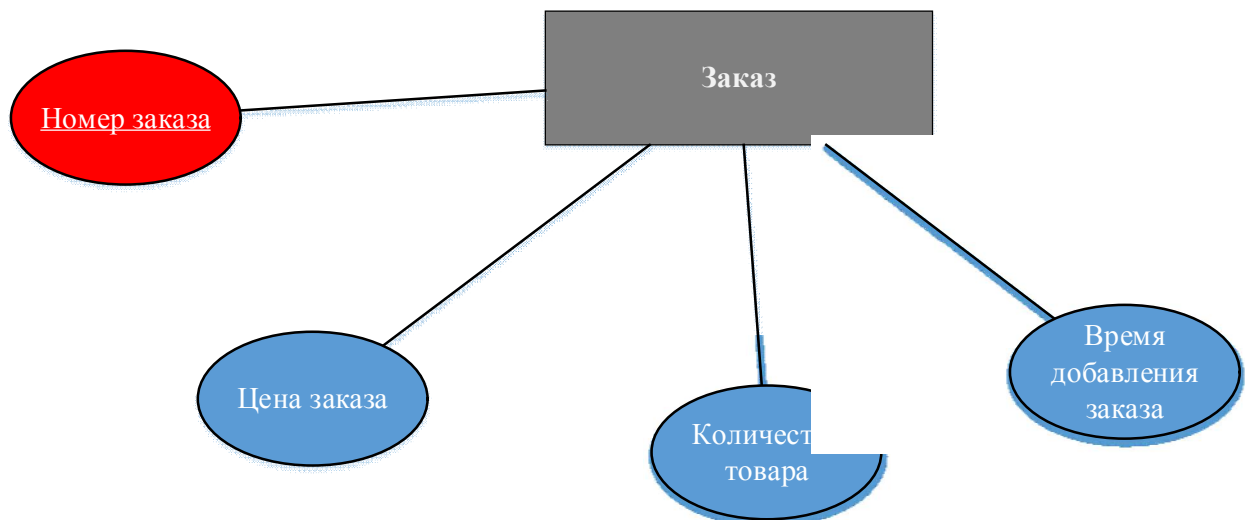


Рис.1.3 Сущность «ЗАКАЗЫ»

Сущность «Заказы» имеет условный атрибут, т.е. атрибут который может иметь пустое значение в поле – Комментарий покупателя, а также все остальные атрибуты являются статическими.

Рассмотрение сущности «Товары» (рис.1.4).

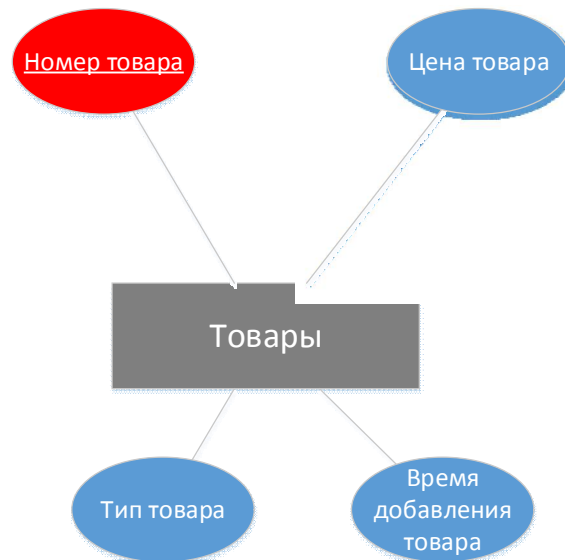


Рис. 1.4 Сущность «ТОВАРЫ»

Сущность «Товары» имеет все динамические товары, кроме атрибута «Время добавления товара», который является статическим и задается только один раз.

Далее необходимо рассмотреть такие сущности, как «Банкноты» (рис.1.5), «Награды» (рис. 1.6) и «Монеты» (рис.1.7).

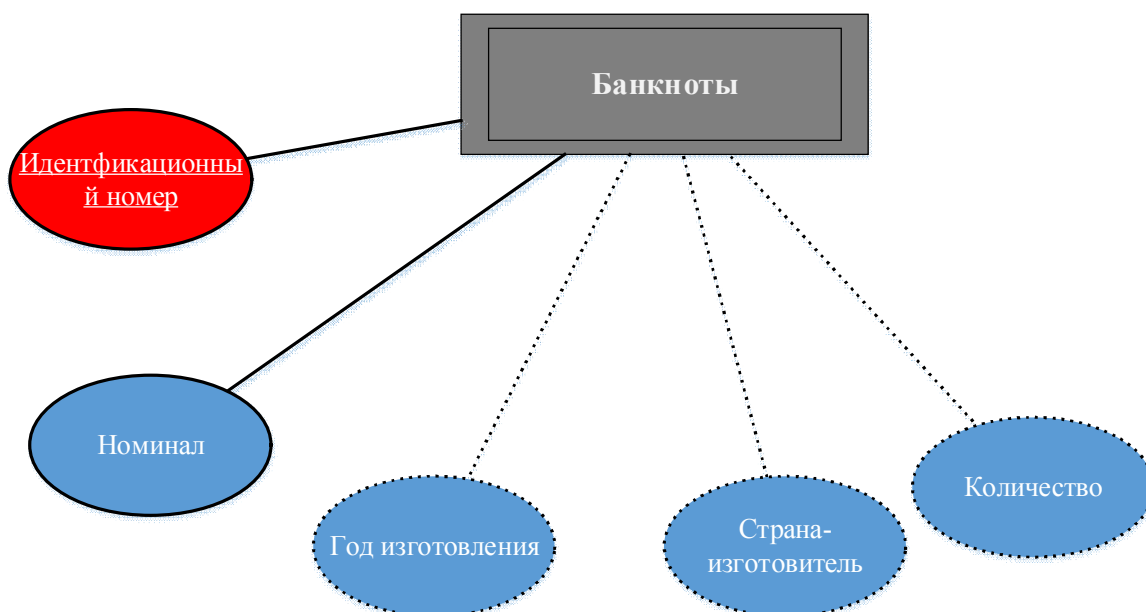


Рис.1.5 Сущность «БАНКНОТЫ»

Сущность «Банкноты» имеет условные атрибуты: «Год изготовления», «Страна-изготовитель» и «Количество», а также статический атрибут «Номинал».

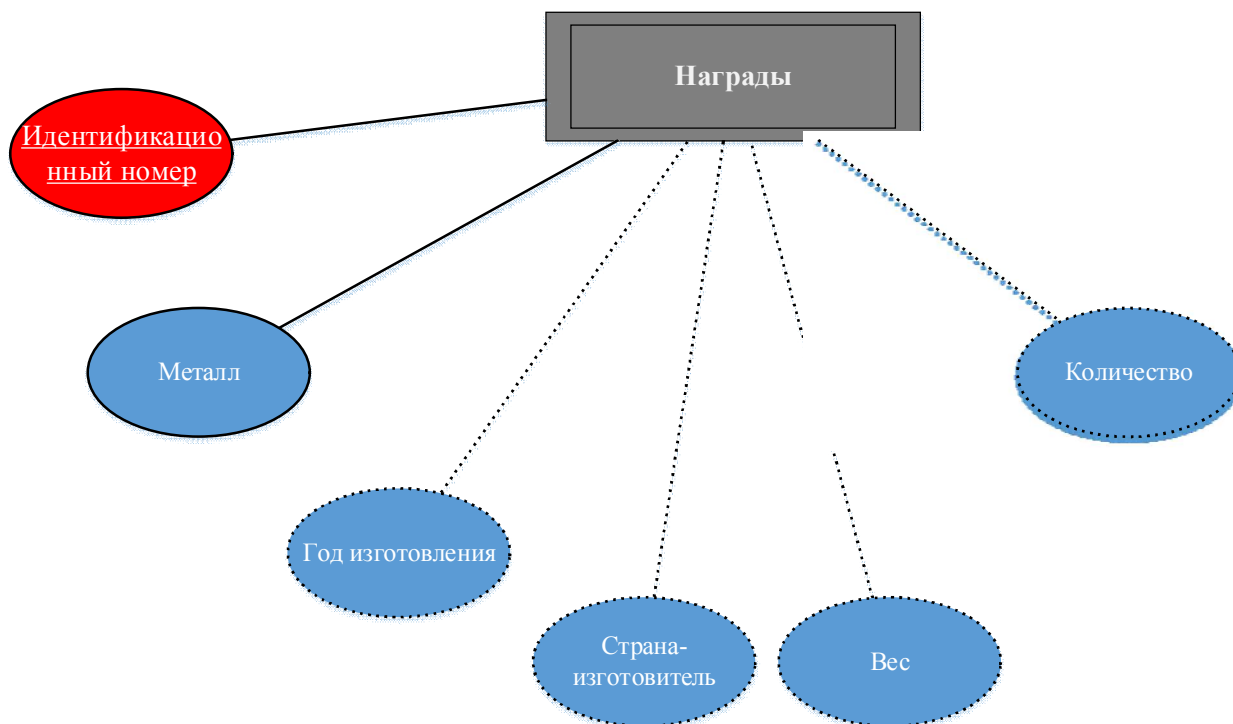


Рис. 1.6 Сущность «НАГРАДЫ»

В сущности «Награды» присутствуют все условные атрибуты, кроме статического «Металл».

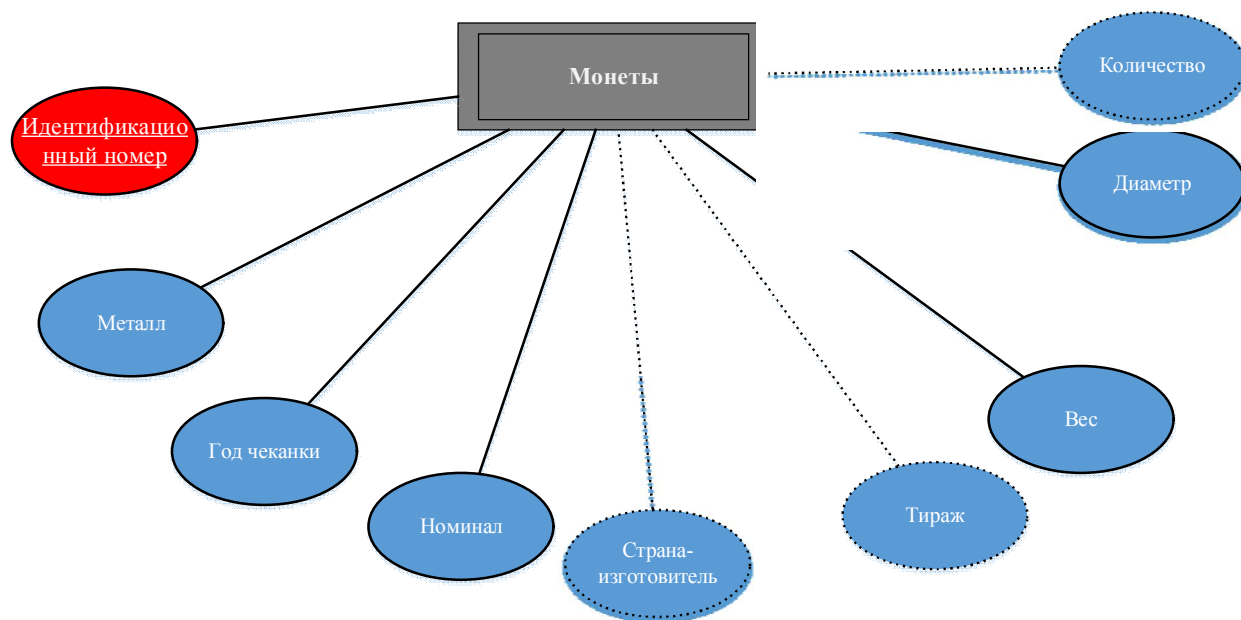


Рис. 1.7 Сущность «МОНЕТЫ»

Сущность «Монеты» обладает динамическим атрибутом «Количество», и всеми остальным статическими. Также у сущности существует условные атрибуты: «Страна-изготовитель» и «Тираж».

Описание связей (рис.1.8).

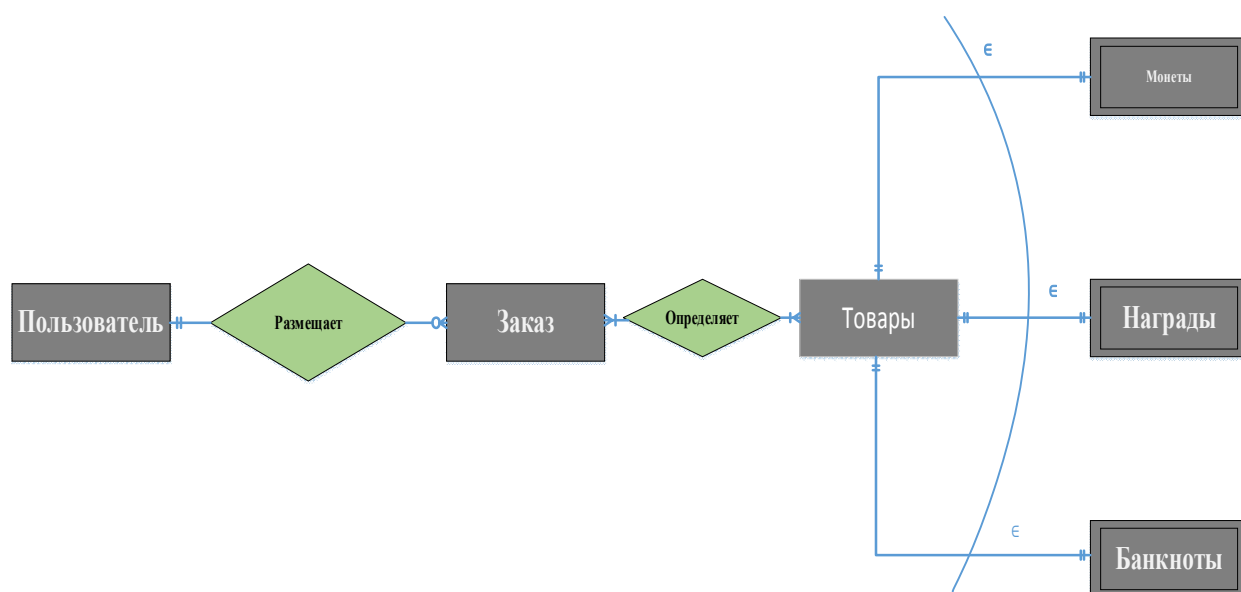


Рис.1.8 Связи между сущностями

Сущность «Пользователь» содержит информацию о всех пользователях, зарегистрированных в базе данных. Каждый экземпляр этой сущности соответствует конкретному пользователю. Между сущностью «Пользователь» и сущностью «Заказ» существует связь типа «1:М», необязательная со стороны заказа (пользователь может не размещать заказ) и обязательная со стороны пользователя (заказ в обязательном порядке должен иметь пользователя (покупателя)).

Сущность «Заказ» хранит данные о заказах. Существует связь между сущностями «Заказ» и «Товары» типа «М:М», обязательную с обеих сторон (заказ должен иметь хотя бы 1 или более товаров, товар должен содержаться в 1 или более заказов).

Сущность «Товары» хранит данные о цене и типе любого товара. Также сущность является сильной сущностью. Существует связь между сущностями «Товары» и «Банкноты», «Товары» и «Награды», «Товары» и «Монеты» типа «1:1», обязательную с двух сторон (каждый товар должен относиться к определенному типу и каждая монета, банкнота или награда должна являться товаром).

Сущности «Монеты», «Банкноты» и «Награды» являются подтипами сущности «Товары». То есть некоторые атрибуты являются необязательным для той или иной сущности (атрибут «Номинал» для сущности «Награды»).

1.4 Логическое проектирование

Цель логического этапа проектирования – организация информации, выделенной на этапе инфологического проектирования, в форму определенной модели.

Задачей логического проектирования является отображение объектов предметной области в объекты выбранной модели данных, в данном проекте –

реляционной, поэтому задача логического проектирования состоит в том, чтобы выбрать:

1. Из каких отношений должна состоять база данных;
2. Какие атрибуты должны быть в этих отношениях.

Выделенные сущности базы данных на инфологическом проектировании:

- Сущность «Заказ»;
- Сущность «Пользователь»;
- Сущность «Товары»;
- Сущность «Банкноты»;
- Сущность «Монеты»;
- Сущность «Награды».

При выборе реляционной модели обязательной процедурой является нормализация.

Под нормализацией понимают процесс построения оптимальной структуры таблиц и связей в реляционной базе данных. Также нормализация направлена на уменьшение избыточности в базе данных.

В терминологии реляционной модели сущности – это отношение – двумерная таблица. Каждая строка в таблице содержит данные, относящиеся к некоторой вещи или какой-то ее части [7].

Отношение находится в 1НФ, если каждый элемент отношения имеет, и всегда будет иметь атомарное значение. Также таблица не должна содержать повторяющихся столбцов.

Разделим ФИО на 3 атрибута – Фамилия, Имя, Отчество. Номинал состоит из самого номинала и денежной единицы. Разделим Номинал на 2 атрибута – Номинал и Денежная единица. Разделим составной атрибут Адрес пользователя на 7 атрибутов – страна, город/район, село, улица, дом, квартира, почтовый индекс. Таблицы, после разбиения атрибутов будут следующими (таблица 1.1).

Таблица 1.1

Приведение к 1НФ

Пользователь	Монеты	Банкноты	Награды
Номер пользователя	Номер монеты	Номер банкноты	Номер награды
Фамилия	Год чеканки	Год изготовления	Металл
Имя	Денежная единица	Денежная единица	Год изготовления

Продолжение таблицы 1.1

Отчество	Номинал	Номинал	Вес
Номер адреса	Вес		Страна- изготовитель
Страна	Диаметр		Количество
Город/район	Количество		Название
Село	Страна- изготовитель		

Улица	Металл		
Дом	Тираж		
Квартира			
Почтовый индекс			
Контактный телефон			
Логин			
Пароль			
E-Mail			

В представленной таблице выполняются все требования к первой нормально форме, описанные выше:

- На любом пересечении таблиц, строки и столбца находится единое значение – атомарное.
- В таблице нет повторяющихся строк.

Таблица находится во второй нормальной форме, если она находится в первой нормальной форме и каждое поле, не входящее в первичный ключ, связано полной функциональной зависимостью с ним. То есть, любой столбец, который не является ключом, должен зависеть от первичного ключа.

Выделение ключевых полей (таблица 1.2).

Таблица 1.2

Выделение первичных ключей

Заказы	Пользователь	Товары
Номер заказа(*)	Номер пользователя(*)	Номер товара(*)
Монеты	Награды	Банкноты

Номер монеты(*)	Номер награды(*)	Номер банкноты(*)
-----------------	------------------	-------------------

Произведем декомпозицию сущности «Пользователь» на 2 сущности «Пользователь» и «Адрес» (таблица 1.3). Атрибуты: страна, город/район, село, улица, дом, квартира, почтовый индекс зависят от номера адреса, то есть зависимость от первичного ключа неполная.

Таблица 1.3

Декомпозиция сущности «ПОЛЬЗОВАТЕЛЬ»

Пользователь
Номер пользователя(*)
Фамилия
Имя
Отчество
Контактный телефон
Логин
Пароль
e-mail
Адрес
Номер адреса(*)
Страна
Город/Район
Село
Дом
Квартира
Почтовый индекс

Также выделим внешние ключи для таблиц (таблица 1.4).

Таблица 1.4

Внешние ключи

Пользователи	Заказы
Номер адреса	Номер товара
	Номер пользователя
Монеты	Банкноты
Номер монеты	Номер банкноты
Награды	
Номер награды	

Таким образом, отношения приведены ко второй нормальной форме, посредством выполнения двух процедур:

- определение первичных ключей;
- выделение в отдельные таблицы атрибуты, неполно зависимые от первичного ключа.

Отношение находится в третьей нормальной форме, когда оно находится во второй нормальной форме и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа

В каждом отношении, приведенном в таблице 1.5, отсутствуют транзитивные зависимости.

Таблица 1.5

БД «Магазин нумизматики» после нормализации

Пользователи	Адрес	Заказы	Товары
id(*)	address_id(*)	cart_id(*)	id_products(*)
Логин	Страна	Номер товара	Страна
Пароль	Город/Район	Номер пользователя	Год
Имя	Село	Цена товара	Цена

Продолжение таблицы 1.5

Фамилия	Улица	Количество товара	Картинка
Отчество	Дом	Время добавления в корзину	
Email	Квартира		Время добавления
Телефон	Почтовый индекс		Количество
Дата и Время регистрации			Тип товара
Номер адреса(address_id)			
Монеты	Банкноты	Награды	
id_products(*)	id_products(*)	id_products(*)	
Номинал	Номинал	Название	
Денежная единица	Денежная единица	Металл	
Металл		Вес	
Диаметр			
Вес			
Тираж			

Все отношения приведены к 3НФ.

1.5 Обзор систем управления базами данных

Главные функции системы управления базами данных – это надежность и производительность. Пользователь должен быть уверен в надежности данных. Сбои должны быть минимизированы и не должны приводить к полным потерям данных. Также, любая СУБД должна обеспечить высокую производительность для решения сложных задач.

Выделяют следующие типы систем управления базами данных по способу доступа к базе данных:

- Файл-серверные;
- Клиент-серверные;
- Встраиваемые.

В файл-серверных СУБД каждый пользователь имеет неограниченный доступ ко всем файлам базы данных. Системы управления базами данных такого типа имеет ряд недостатков:

1. При каждом запросе пользователя обновляется все данные в его локальной копии, вследствие чего, производительность ухудшается.
2. Целостность данных зависит от программ клиентов. Пользователь может внести ошибочные данные в базу данных, которые могут отразиться на других пользователях.

На данный момент файл-серверные СУБД не пользуются популярностью и считаются устаревшими.

В клиент-серверных СУБД, наоборот, все данные хранятся и обрабатываются на одном сервере, при этом к данным имеет доступ только один сервер. Клиенты не имеют прямого доступа к файлам данных; клиенты посылают запросы на получение данных и получают ответ. Именно СУБД этого типа пользуется большой популярностью.

Встраиваемые СУБД являются частью готового программного обеспечения. Встраиваемые СУБД служат для локального хранения информации и ими не пользуются группа клиентов. Пример использования СУБД этого типа является СУБД SQLite, которая используется в ОС Android.

На сегодняшнее время крупными компаниями-разработчиками реляционных СУБД является Oracle, Microsoft и IBM. Из документоориентированных СУБД – MongoDB.

На протяжении многих лет «MySQL» – СУБД компании «Oracle», является лидером среди других систем управления базами данных. Эта СУБД успешно работает с веб-сайтами и веб-приложениями. На «MySQL» работают большинство коммерческих веб-сайтов, а также она служит серверной СУБД для многих приложений. Благодаря популярности MySQL, существует огромное количество плагинов и расширений, которые позволяют облегчить работу с системой.

Также MySQL легко работает с большими объемами информации, и при этом, благодаря упрощения некоторых стандартов позволяет держать производительность на высоком уровне.

«Oracle» – это реляционная СУБД, которая отличалась своей надежностью работы с данными. Следует помнить, что «Oracle» – платное решение, следовательно, разработчик может получить ответы на любые вопросы от самих разработчиков. Также, «Oracle» позволяет восстанавливать данные после сбоев и также легко справляется с большим количеством данных. Следует отметить, компания «Яндекс» работала с этой СУБД, вплоть до 2015 года.

«Microsoft SQL Server» – продукт компании Microsoft. Получила свою популярность из-за простоты обращения с самой СУБД, а также тесная интеграция с операционной системой Windows, что является преимуществом для компаний, пользующихся продуктами Microsoft.

«DB2» – СУБД компании IBM. Существенных преимуществ у СУБД нет, однако некоторые эксперименты, говорят о существенной экономии при использовании DB2.

«MongoDB» – СУБД одноименной компании. Это «нереляционная» СУБД отличается своей скоростью разработки, а также не нужно синхронизировать схему в базе данных и приложении.

Рейтинг популярных, на данный момент, систем управления базами данных (рис. 1.9).

Rank			DBMS	Database Model	Score		
Dec 2017	Nov 2017	Dec 2016			Dec 2017	Nov 2017	Dec 2016
1.	1.	1.	Oracle +	Relational DBMS	1341.54	-18.51	-62.86
2.	2.	2.	MySQL +	Relational DBMS	1318.07	-3.96	-56.34
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1172.48	-42.59	-54.17
4.	4.	4.	PostgreSQL +	Relational DBMS	385.43	+5.51	+55.41
5.	5.	5.	MongoDB +	Document store	330.77	+0.29	+2.09
6.	6.	6.	DB2 +	Relational DBMS	189.58	-4.48	+5.24
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	125.88	-7.43	+1.18
8.	↑ 9.	↑ 9.	Redis +	Key-value store	123.24	+2.05	+3.34
9.	↓ 8.	↓ 7.	Cassandra +	Wide column store	123.21	-1.00	-11.07
10.	10.	↑ 11.	Elasticsearch +	Search engine	119.78	+0.37	+16.51

Рис. 1.9 Рейтинг популярных СУБД

Проанализировав основные СУБД, оказалось, что наиболее подходящей, а также быстрой в освоении для создания базы данных курсового проекта – СУБД «MySQL». MySQL имеет ряд преимуществ перед другими реляционными СУБД:

- MySQL одна из самых быстродействующих СУБД.
- MySQL не требует больших затрат ресурсов компьютера.
- Довольно удобный интерфейс phpMyAdmin – веб-приложения для работы с СУБД MySQL.
- phpMyAdmin не требует установки.
- Для упрощения работы с базой данных: создание БД, создание таблиц и их обновление, существует ряд вкладок с понятным интерфейсом.

Выводы к главе 1

В первой главе были выявлены основные термины, к которым приведены подробные определения.

В результате выполнения исследовательской работы, было выявлено, что для достижения автоматизации процессов магазина нумизматики с использованием большого количества информации требуется разработка базы

данных, позволяющая решить проблему безопасного хранения и систематизации информации согласно современным требованиям.

На этапе инфологического проектирования были выделены сущности и их атрибуты. Также были определены подтипы сущностей и связи между ними.

В результате логического анализа сущностей и их атрибутов, выделенных в инфологическом проектировании, появилась новая сущность, а также были добавлены внешние ключи.

На этапе сравнения систем управления базами данных, было выявлено то, что наиболее популярными СУБД являются реляционные СУБД. Одним из самых популярных СУБД для веб-разработки является MySQL, характеристики которой позволяют использовать её в разработке базы данных курсового проекта.