МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

**«РАЗРАБОТКА ПРОЕКТА ПО СОЗДАНИЮ СИСТЕМЫ**

**ОБРАБОТКИ ЗАКАЗА ДЛЯ КНИЖНОГО МАГАЗИНА»**

Студент .

(Фамилия И.О.)

Факультет Инфокоммуникационные технологии .

Кафедра Программных систем Группа К4120 .

Направление (специальность) 11.04.02 Программное . обеспечение в инфокоммуникациях .

Руководитель Иванов Сергей Евгеньевич, доцент, к.ф.- м.н .

.( Фамилия, И.О., должность, ученое звание, степень )

Дисциплина Разработка и внедрение распределенных . систем .

Санкт-Петербург

2017 г.

Оглавление

[Введение 3](#_Toc478597655)

[ЧАСТЬ 1. Цели проекта 3](#_Toc478597656)

[1.2 Цели и предметная область проекта 3](#_Toc478597657)

[1.2 Обзор информационных систем для реализации проекта 4](#_Toc478597658)

[ЧАСТЬ 2. Формирование образа проекта в целом 5](#_Toc478597659)

[2.1 Функциональные и нефункциональные требования 5](#_Toc478597660)

[2.2 Анализ бизнес-процессов. Модель информационной системы в виде диаграммы потоков данных. Структура информационных потоков. 5](#_Toc478597661)

[2.3 Основная функциональность и ключевые функции системы 7](#_Toc478597662)

[ЧАСТЬ 3. Проектирование 7](#_Toc478597663)

[3.1 Сущности, необходимые атрибуты и атрибуты, являющиеся уникальными идентификаторами для сущностей 7](#_Toc478597664)

[3.2 Отношения между сущностями, представление их графически. Логическая модель данных информационной системы. 9](#_Toc478597665)

[3.3 Нормализации логической модели данных 10](#_Toc478597666)

[3.4 Функциональная архитектура системы 10](#_Toc478597667)

[3.5 Выбор архитектуры для проекта распределенной системы 12](#_Toc478597668)

[3.6 Физическая модель данных системы. Типы данных для атрибутов. Реляционная модель. 14](#_Toc478597669)

[3.7 Основные пункты технического задания на проект 17](#_Toc478597670)

[Заключение 18](#_Toc478597671)

[Список литературы 19](#_Toc478597672)

# Введение

Дистанционная книжная торговля, к которой относится и виртуальная торговля, появилась, с одной стороны, из-за дефицита предложения, с другой – из-за неэффективности для издателей традиционных каналов сбыта, таких, как оптовая и розничная торговля. Издателям далеко не всегда удается успешно реализовать книги через оптовые базы и книжные магазины, прежде всего, из-за их ориентированности на виды литературы, популярной среди всех слоев населения, на бестселлеры и раскрученных авторов. Книги более узкой направленности, весьма востребованные, даже необходимые для определенных категорий покупателей, часто просто не доходят до них по обычным каналам.

Одним из наиболее бурно развивающихся видов книжной торговли с удаленным доступом являются книжные интернет-магазины, которые имеют немало сходства с традиционной почтово-посылочной торговлей, гак как большинство заказов выполняется через почту. Однако в отличие от торговли по каталогам в виртуальном книжном магазине можно сначала посмотреть книгу, познакомиться с ее содержанием, рекомендациями и отзывами других читателей, и только потом сделать заказ. Также в виртуальном книжном магазине имеются более широкие возможности для поиска нужной книги и представлен более широкий ассортимент книг. Эта форма торговли приобретает всё большую популярность и является одной из наиболее перспективных в плане дальнейшего развития.

# ЧАСТЬ 1. Цели проекта

## 1.2 Цели и предметная область проекта

**Целью** данной работы являетсясоздание проекта для системы обработки заказа для книжного магазина, отвечающего функциональным, техническим и операционным требованиям и обеспечивающего необходимые параметры, такие как производительность, масштабируемость, безопасность и управляемость.

Основные цели работы:

• систематизация, закрепление и расширение теоретических и практических знаний по дисциплине «Разработка и внедрение распределенных систем» и применение их при решении конкретных задач;

• приобретение навыков проектирования распределенных систем;

• развитие творческого мышления и навыков самостоятельной работы;

• приобретение навыков использования справочной, нормативной и научной литературы, а также Интернет-ресурсов;

• приобретение навыков оформления пояснительной записки.

**Характеристика предметной области**

Основным профилем деятельности книжного магазина является реализация книжной продукции. Данная система включает в себя описание различных книг, авторов, клиентов и т.п.

Система сохраняет информацию о:

• книгах (ISBN книги, наименование, количество страниц, цена) и их категориях;

• авторах книг (ФИО автора, год рождения);

• издательствах;

• заказах (описание заказов);

• клиентах (ФИО клиента, контактные данные: телефон, адрес);

• сотрудниках (курьерах) (ФИО, контактные данные).

Для обеспечения возможности корректного описания заказов и связей с покупателями необходима следующая информация:

- о покупателях: ФИО, адрес, телефон, логин, пароль;

- о заказах: заказчик, ISBN заказанных книг и их количество, дата заказа;

- об исполнителях заказа (курьерах, осуществляющих доставку заказов).

- цена книги.

## 1.2 Обзор информационных систем для реализации проекта

My-shop.ru – система, в которой представлен большой ассортимент книг разных категорий, канцтоваров, игрушек, сувениров, а также товаров для творчества. Система предлагает возможность поиска книг по различным параметрам (названию, автору, ISBN, категории, цене, наличию). После регистрации у пользователя появляется возможность оформить заказ. В персональном разделе пользователь можно просмотреть свои заказы, сохранить и изменить адреса, на которые может производиться доставка. Система предлагает на выбор разные способы получения заказа: доставка курьером, почтой, а в некоторых городах можно забрать заказ в пункте самовывоза.

Лабиринт.ru – книжный интернет-магазин, в котором помимо книг представлены канцелярские товары, игрушки и сувениры. Поиск осуществляется по жанрам, результаты можно отсортировать по алфавиту или наличию. Пользователи могут оставлять отзывы и рецензии на прочитанные книги, новости и обзоры, добавлять содержания и иллюстрации. Любую рецензию можно оценить: чем больше человек похвалило рецензию, тем выше ее рейтинг. За интересный и содержательный отзыв на товар рецензенту присуждается бонус: от 5-ти до 20-ти рублей, которые можно использовать для оплаты заказа. Оплата может производиться наличными при доставке курьерской службой и при самовывозе, также возможна оплата подарочным сертификатом, через мобильный телефон, доступны электронные формы оплаты.

# ЧАСТЬ 2. Формирование образа проекта в целом

## 2.1 Функциональные и нефункциональные требования

Функциональные требования к проекту:

* внесение информации о товарах (книгах);
* просмотр каталога товаров;
* регистрация пользователей;
* выбор необходимого товара и просмотр подробной информации;
* оформление заказа;
* формирование отчётов.

Нефункциональные требования к проекту:

* система должна обеспечивать безотказную работу в режиме 24х7;
* время формирования отчетов не должно превышать 3 минут;
* система должна быть разработанная с использованием языка С#;
* эффективность и устойчивость к сбоям;
* легкость и простота использования;
* доступный интерфейс.

## 2.2 Анализ бизнес-процессов. Модель информационной системы в виде диаграммы потоков данных. Структура информационных потоков.

Анализ бизнес-процессов осуществлён с помощью программы AllFusion Process Modeler 7 (BPwin). С использованием BPwin строятся диаграммы бизнес-процессов, ясно показывающие бизнес-процессы (блоки), результаты их работы и ресурсы, необходимые для их функционирования. BPwin-модель обеспечивает объединенную картину того, как организация добивается выполнения своих целей, от маленьких отделов до всей компании в целом. BPwin поддерживает методологии IDEF0 (функциональная модель), DFD (DataFlow Diagram) и IDEF3 (Workflow Diagram).

Функциональная модель предназначена для описания существующих бизнес- процессов на предприятии и идеального положения вещей – того, к чему нужно стремиться. Методология IDEF0 предписывает построение иерархической системы диаграмм. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.

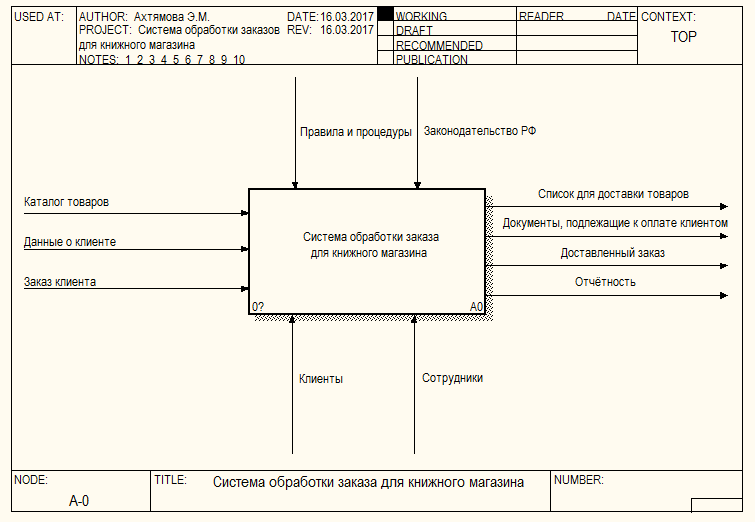


Рисунок 1 – Контекстная диаграмма IDEF0 «Система обработки

заказа для книжного магазина»

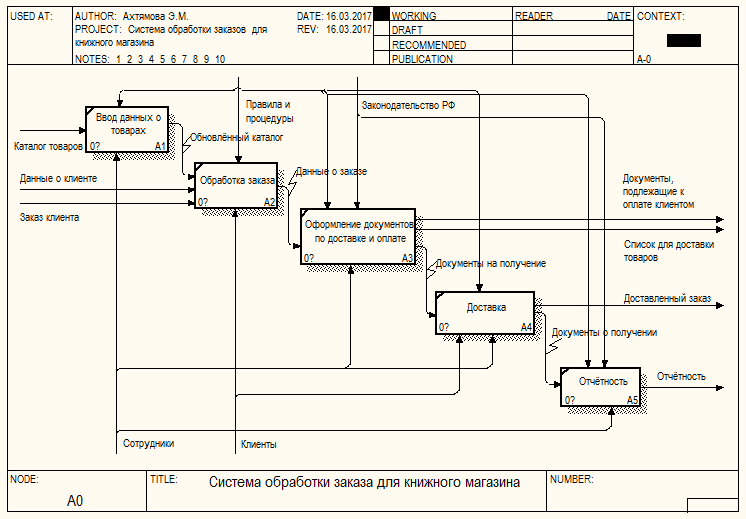


Рисунок 2 – Диаграмма декомпозиции IDEF0

## 2.3 Основная функциональность и ключевые функции системы

Система должна сохранять информацию о: книгах (ISBN книги, наименование, количество страниц, цена) и их категориях, авторах книг (ФИО автора, год рождения), издательствах, заказах (описание заказов), клиентах (ФИО клиента, контактные данные: телефон, адрес), сотрудниках (курьерах) (ФИО, контактные данные).

Ключевые функции системы:

* внесение информации о товарах (книгах);
* просмотр каталога товаров;
* регистрация пользователей;
* выбор необходимо товара и просмотр подробной информации;
* оформление заказа;
* формирование отчётов.

# ЧАСТЬ 3. Проектирование

## 3.1 Сущности, необходимые атрибуты и атрибуты, являющиеся уникальными идентификаторами для сущностей

Выделены следующие сущности: книги, авторы, клиенты, сотрудники (курьеры), заказы, издательства, адреса, категории.

Атрибуты сущности «Книги»:

* ISBN\_book – номер книжного издания;
* name\_book – наименование книги;
* price – цена;
* pages – количество страниц;
* ID\_publ\_house – код издательства;
* ID\_category –код категории.

Атрибуты сущности «Авторы»:

* ID\_author – код автора;
* sname – фамилия автора;
* fname – имя автора;
* otch – отчество автора;
* year\_birth – год рождения;
* ISBN\_book – номер книжного издания.

Атрибуты сущности «Клиенты»:

* ID\_cust – код клиента;
* sname – фамилия клиента;
* fname – имя клиента;
* otch – отчество клиента;
* phone – телефон;
* login – логин;
* password – пароль.

Атрибуты сущности «Курьеры»:

* ID\_cour – код курьера;
* sname – фамилия курьера;
* fname – имя курьера;
* otch – отчество курьера;
* phone – телефон.

Атрибуты сущности «Заказы»:

* ID\_order – код заказа;
* date – дата заказа;
* quantity – количество книг;
* ISBN\_book – номер книжного издания;
* ID\_cour – код курьера;
* ID\_cust – код клиента.

Атрибуты сущности «Издательства»:

* ID\_publ\_house – код издательства;
* name\_ph – название издательства.

Атрибуты сущности «Категории»:

* ID\_category –код категории.
* name – название издательства.

Атрибуты сущности «Адреса»:

* ID\_address – код адреса;
* city – город;
* street – улица;
* housing – корпус;
* num\_house – номер дома;
* ID\_cust – код клиента.

Атрибут или группа атрибутов, которая уникально определяющих данную сущность называется ключом сущности или уникальным идентификатором сущности. Уникальный идентификатор (UID) – это специальный атрибут или группа атрибутов, который однозначно идентифицирует конкретный экземпляр сущности.

Для сущности «Книги» уникальным идентификатором является атрибут номер книжного издания (ISBN\_book), для сущности «Авторы» – код автора (ID\_author), для сущности «Клиенты» – код клиента (ID\_cust), для сущности «Курьеры» – код курьера (ID\_cour), для сущности «Заказы» – код заказа (ID\_ order), для сущности «Издательства» – код издательства (ID\_publ\_house), для сущности «Категории» – код категории (ID\_category), для сущности «Адреса» – код адреса (ID\_address).

## 3.2 Отношения между сущностями, представление их графически. Логическая модель данных информационной системы.

Отношения представляют собой бизнес-правила для связи сущностей. Отношения позволяют по одной сущности находить другие сущности, связанные с ней. Отношение всегда содержит в себе два правила и графически изображаются линией, соединяющей две сущности.

Логическая модель данных является начальным прототипом будущей базы данных, она описывает понятия, используемые в данной предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Концептуальный выбор логической модели данных предопределяет уровень эффективности той или иной программной реализации базы данных. Модель позволяет установить связи между структурными элементами базы данных, описать и классифицировать эти элементы, выделить основные информационные потоки, а также позволяет проводить анализ, дальнейшее развитие системы и имитацию процессов, без реализации непосредственно базы данных.

Для представления логической модели используется нотация Чарльза Бахмана, в которой сущность изображается прямоугольником, внутри которого перечисляются имя сущности и атрибуты. Для каждого атрибута указывается его роль: PK (primary key) – первичный ключ, FK (foreign key) – внешний ключ, PFK (primary and foreign key) – внешний ключ в составе первичного, I (inherited) – атрибут, унаследованный от обобщенной сущности.

Связи в нотации Бахмана изображаются линией с двумя названиями. Мощность связи указывается с помощью стрелок: «––» – мощность «один-к-одному», «→» – мощность «один-ко-многим»,«↔» – мощность «многие-ко-многим», полнота – с помощью пустого или закрашенного кружочка, а идентифицирующая связь – с помощью специальной метки – точки, расположенной около зависимой сущности.

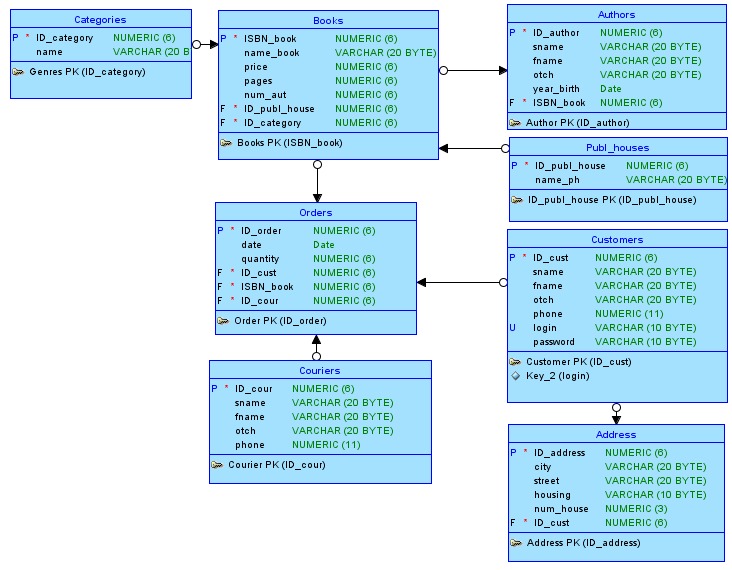


Рисунок 3 – Логическая модель базы данных

## 3.3 Нормализации логической модели данных

Созданная база данных соответствует третьей нормальной форме:

* База данных соответствует первой нормальной форме, так как каждый столбец в строке является атомарным, т.е. столбец содержать одно и только одно значение для заданной строки;
* База данных соответствует второй нормальной форме, так как база данных соответствует первой нормальной форме и все столбцы, не входящие в состав первичного ключа, функционально полно зависят от первичного ключа;
* База данных соответствует третьей нормальной форме, так как база данных соответствует второй нормальной форме и все не ключевые атрибуты функционально зависят только от первичного ключа.

## 3.4 Функциональная архитектура системы

Эффективность функционирования информационной системы во многом зависит от её архитектуры. Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

В качестве архитектуры выбрана архитектура клиент-сервер.

Практические реализации такой архитектуры называются клиент-серверными технологиями. Технология «клиент-сервер» – эᴛᴏ модель вычислений, предусматривающая распределение функций обработки в многопользовательской базе данных по нескольким компьютерам. Распределение выполнения функций обработки между компьютерами осуществляется с использованием протокола сервисных запросов, т.е. один компьютер «клиент» запрашивает обслуживание у другого компьютера «сервера», который реализует обслуживание и отсылает его результаты «клиенту».

Основная задача, которую решает клиентское приложение, – это обеспечение интерфейса с пользователем, т.е. ввод данных и представление результатов в удобном для пользователя виде, и управление сценариями работы приложения. Основные функции серверной СУБД – обеспечение надежности, согласованности и защищенности данных, управление запросами клиентов, быстрая обработка SQL-запросов.

Основные требования, которые предъявляются к компьютеру пользователя – это возможность запускать необходимые пользователю приложения, осуществлять соединение с сервером и обеспечивать нормальную работоспособность разработанной программы.

К серверу базы данных предъявляются гораздо более жесткие требования, чем к компьютеру пользователя. Это связано с тем, что вся важная информация хранится именно на сервере. Основные требования к серверу следующие: во-первых, обеспечение сохранности информации при различных сбоях в системе – зависании, случайной перезагрузке, во-вторых, обеспечение сохранности информации при перебоях в электроснабжении, в-третьих, высокое быстродействие для обработки запросов пользователей. Для соблюдения первого требования применяется дисковая подсистема, состоящая из двух полностью одинаковых жестких дисков. Из этих дисков аппаратными средствами реализуется зеркалирование информации с одного на другой, при выходе из строя одного диска целиком или частично информация на втором полностью сохраняется и может быть восстановлена после замены поврежденного диска. Эта технология носит название RAID. Для соблюдения второго требования применяется источник бесперебойного питания. При отключении электроэнергии он обеспечивает в течение некоторого времени работоспособность компьютера, что позволяет завершить выполняемые операции и корректно завершить работу компьютера до восстановления электроснабжения. Кроме того, источник бесперебойного питания также фильтрует различные помехи в электроснабжении (например, скачки напряжения), которые могут повредить вычислительной технике. Для соблюдения третьего требования применяется достаточно мощный процессор в сочетании с большим объемом оперативной памяти.

## 3.5 Выбор архитектуры для проекта распределенной системы

В качестве системы управления базами данных была выбрана объектно-реляционная система управления базами данных компании Oracle – Oracle Database.

**Oracle Database** – это производительная, масштабируемая, функциональная СУБД для решения широкого спектра задач, доступная под различные архитектуры процессоров и для различных операционных систем.

Ключевые возможности Oracle Database:

* Real Application Cluster (RAC) обеспечивает работу одного экземпляра базы данных на нескольких узлах grid, позволяя управлять нагрузкой и гибко масштабировать систему в случае необходимости.
* Automatic Storage Management (ASM) позволяет автоматически распределять данные между имеющимися ресурсами систем хранения данных, что повышает отказоустойчивость системы и снижает общую стоимость владения (TCO).
* Производительность. Oracle Database позволяет автоматически управлять уровнями сервиса и тиражировать эталонные конфигурации в рамках всей сети.
* Простые средства разработки. Новый инструмент разработки приложений HTML DB позволит простым пользователям создавать эффективные приложения для работы с базами данных в короткие сроки.
* Самоуправление. Специальные механизмы Oracle Database позволяют самостоятельно перераспределять нагрузку на систему, оптимизировать и корректировать SQL-запросы, выявлять и прогнозировать ошибки.
* Большие базы данных. Максимальный размер экземпляра базы данных Oracle может достигать 8 экзабайт.
* Поддержка переносимых табличных пространств, система управления потоками данных Oracle Streams и модель распределенных SQL-запросов. Для переноса существующих баз данных в среду Grid в них не потребуется вносить изменений, что позволяет быстро начать использовать все преимущества Oracle Database.

**Oracle SQL Developer Data Modeler –** это комплексное решение, позволяющее разработчикам проектировать реляционные модели взаимосвязей объектов для последующего преобразования их в полноценные базы данных. Продукт поддерживает логическое, реляционное, многомерное моделирование и моделирование типов данных, предлагая возможности многоуровневого проектирования и построения концептуальных диаграмм сущностей и связей. Пользователи могут создавать, расширять и модифицировать модели, а также сравнивать их с уже существующими.

Oracle SQL Developer Data Modeler предлагает множество функциональных возможностей для моделирования данных и баз данных, включая:

* Визуальное моделирование взаимосвязей между сущностями **–** поддерживает нотации Баркера (Barker) и Бахмана (Bachman), чтобы разработчики могли переключаться между моделями для удовлетворения потребностей клиентов или для создания и сохранения различных визуальных представлений моделей.
* Ускоренное преобразование ERD-моделей в реляционные модели – трансформация всех правил и решений, сделанных на концептуальном уровне, в реляционную модель, детали в которой уточняются и обновляются.
* Разделение реляционной и физической моделей **–** позволяет разработчикам создавать одну реляционную модель для разных версий базы данных или для разных баз данных, включая Oracle Database, IBM DB2 V7 и V8 для платформ Linux, UNIX, Windows и OS/390, а также Microsoft SQL Server 2000 и 2005.
* Полный набор физических определений для баз данных **–** поддерживает такие физические определения, как секции, роли и табличные пространства для конкретных версий базы данных в средах с разными СУБД от разных производителей, обеспечивая большую согласованность и повышение продуктивности разработчиков.

Продукт интегрируется с бесплатным Oracle SQL Developer **–** популярным графическим инструментом Oracle для разработки баз данных, **–** чтобы предоставить разработчикам возможность открывать и просматривать созданные ранее структуры, а также выполнять запросы и формировать отчеты с использованием репозитория отчетов.

**Oracle SQL Developer ‒** интегрированная среда разработки на языках SQL и PL/SQL, с возможностью администрирования баз данных, ориентирована на применение в среде Oracle Database. Oracle SQL Developer написан на языке Java и является кросс-платформенным, т. е. работает во всех операционных системах. Oracle SQL Developer интегрируется с Apex для разработки и администрирования приложений. Oracle SQL Developer позволяет выполнять экспорт и импорт данных и структур. Oracle SQL Developer – это намного больше, чем просто SQL-редактор, такой например, как PL/SQL Developer, Toad и им подобные. Это мощный инструмент для разработки на PL/SQL с контролем версий, инструмент миграции с других баз данных и администрирования связи между различными СУБД, разработки отчетов и публикации в Oracle Apex и многое другое.

**Microsoft Visual Studio** ‒ это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Эти инструменты предназначены для максимально эффективной совместной работы; все они доступны в интегрированной среде разработки (IDE) Visual Studio.Visual Studio можно использовать для создания различных типов приложений, от простых приложений для магазина и игр для мобильных клиентов до больших и сложных систем, обслуживающих предприятия и центры обработки данных. Можно создавать:

1. приложения и игры, которые выполняются не только на платформе Windows, но и на Android и iOS;
2. веб-сайты и веб-службы на основе ASP.NET, JQuery, AngularJS и других популярных платформ;
3. приложения для самых разных платформ и устройств, включая, но не ограничиваясь: Office, Sharepoint, Hololens, Kinect и "Интернета вещей";

* игры и графические приложения для разных устройств Windows, включая Xbox, с поддержкой DirectX.

Для реализации проекта был выбран язык программирования C#.

**C#** – язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции. Разработан в 1998—2001 годах группой инженеров под руководством Андерсa Хейлсбергa в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет строгую статическую типизацию, поддерживает полиморфизм, перегрузку операторов, указатели на функции-члены классов, атрибуты, события, свойства, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Delphi, Modula и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++) или вывода типов (в отличие от Haskell).

## 3.6 Физическая модель данных системы. Типы данных для атрибутов. Реляционная модель.

Основные типы данных, использованных в проекте:

* VARCHAR: тип используется для хранения строк переменной длины. Синтаксис: VARCHAR [(длина)]. Параметр длина является необязательным. Если длина строки не указана явно, она полагается равной 1 символу. Максимальное значение параметра длина – 4000 символов.
* NUMBER: тип используется для представления чисел с заданной точностью. Синтаксис: NUMBER [(точность [,масштаб])]. Если значение параметра точность не указано явно, оно полагается равным 38. Значение параметра масштаб по умолчанию предполагается равным 0. Значение параметра точность может изменяться от 1 до 38, значение параметра масштаб может изменяться от -84 до 127. Использование отрицательных значений масштаба означает сдвиг десятичной точки в сторону старших разрядов.
* DATE: тип DATE используется для хранения даты и времени. Допускаются даты с 1 января 4712 г. до н.э. до 31 декабря 4712 г. н.э. Время хранится с точностью до секунды. Синтаксис: DATE. Поддерживается специальная арифметика дат и времен. Добавление к переменной типа DATE числа интерпретируется Oracle как определение более поздней даты, а вычитание – как определение более ранней. В Oracle одному дню соответствует число 1. Для оперирования одним часом можно использовать выражение 1/24, одной минутой – 1/24/60 и т.д.

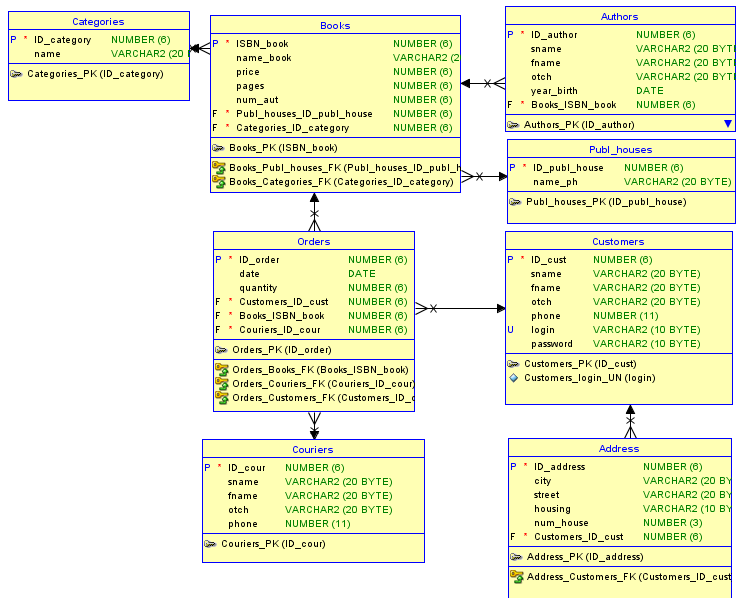


Рисунок 4 – Реляционная модель базы данных

Таблицы базы данных были заполнены:

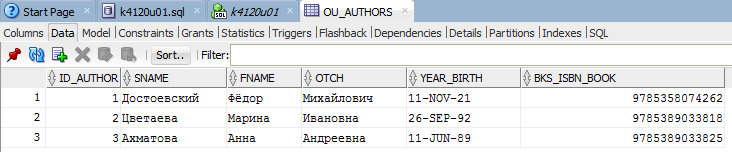


Рисунок 5 – Таблица «OU\_AUTHORS»

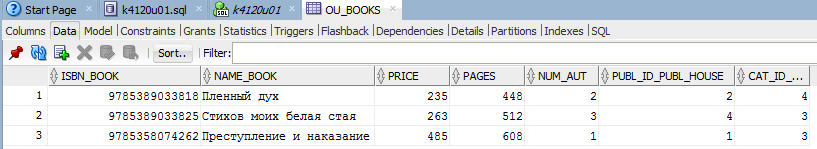


Рисунок 6 – Таблица «OU\_BOOKS»

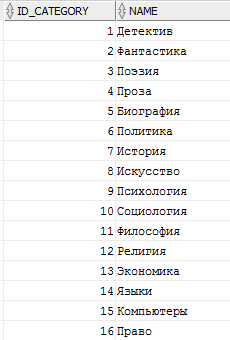


Рисунок 7 – Таблица « «OU\_CATEGORIES»

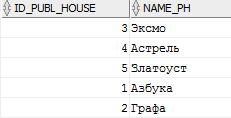


Рисунок 8 – Таблица «OU\_PUBL\_HOUSES»

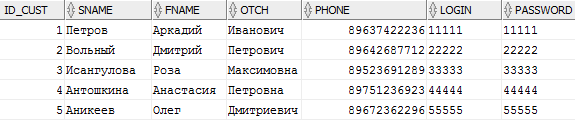


Рисунок 9 – Таблица «OU\_CUSTOMERS»

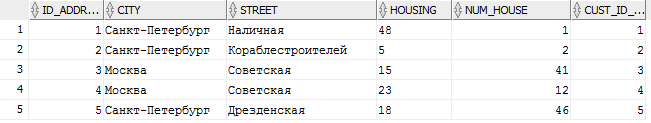


Рисунок 10 – Таблица «OU\_ADDRESS»

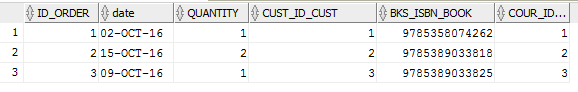


Рисунок 11 –Таблица «OU\_ORDERS»

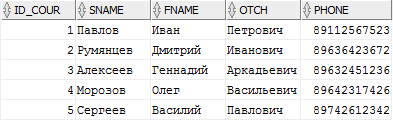


Рисунок 12 – Таблица «OU\_COURIERS»

## 3.7 Основные пункты технического задания на проект

По данному проекту были определены следующие пункты технического задания:

1. Цели и предметная область проекта (Смотреть часть 1 пункт 1.1)

2. Обзор информационных систем для реализации проекта (Смотреть часть 1 пункт 1.2)

3. Функциональные и нефункциональные требования (Смотреть часть 2 пункт 2.1)

4. Формирование образа проекта в целом (Смотреть часть 2)

5. Анализ бизнес-процессов. Сущности и необходимые атрибуты. (Смотреть часть 2 пункт 2.2)

6. Отношения между сущностями, представление их графически. (Смотреть часть 3 пункт 3.2)

7. Атрибуты, которые будут являться уникальными идентификаторами для каждой сущности. (Смотреть часть 3 пункт 3.1)

8. Основная функциональность и ключевые функции системы (Смотреть часть 2 пункт 2.3)

9. Разработка модели информационной системы в виде диаграммы потоков данных (Смотреть часть 2 пункт 2.2)

10. Структура информационных потоков (Смотреть часть 2 пункт 2.2)

11. Функциональная архитектура системы (Смотреть часть 3 пункт 3.4)

12. Выбор архитектуры для проекта распределенной системы (Смотреть часть 3 пункт 3.5)

13.  Логическая модель данных информационной системы (Смотреть часть 3 пункт 3.2)

14. Нормализация логической модели данных (Смотреть часть 3 пункт 3.3)

15. Построение физической модели данных системы. Типы данных для атрибутов. Реляционная модель (Смотреть часть 3 пункт 3.6)

16. Основные пункты технического задания на проект (Смотреть часть 3 пункт 3.7)

# Заключение

В результате выполнения курсовой работы был создан проект для системы обработки заказа для книжного магазина, отвечающий функциональным и техническим требованиям. Для реализации проекта была исследована предметная область, сформированы функциональные и нефункциональные требования, произведён анализ бизнес-процессов, определены ключевые функции системы, определены сущности и необходимые атрибуты, а также созданы логическая и физическая модели данных.

# Список литературы

1. Герасименко, А.А. Оптимальная компенсация реактивной мощности в системах распределения электрической энергии: монография. [Электронный ресурс]: Монографии / А.А. Герасименко, В.Б. Нешатаев. – Электрон. дан. – Красноярск: СФУ, 2012. – 218 с. – Режим доступа: http://e.lanbook.com/book/45701
2. Кирсанов, Э.А. Обработка информации в пространственно-распределенных системах радиомониторинга: статистический и нейросетевой подходы. [Электронный ресурс]: Учебные пособия / Э.А. Кирсанов, А.А. Сирота. – Электрон. дан. – М.: Физматлит, 2012. – 344 с. – Режим доступа: http://e.lanbook.com/book/59646
3. Фредерик П. Брукс. Проектирование процесса проектирования: записки компьютерного эксперта.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2012. – 464 с.
4. Александров, Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы. [Электронный ресурс]: Учебные пособия – Электрон. дан. – М.: Финансы и статистика, 2011. – 224 с. – Режим доступа: http://e.lanbook.com/book/5306
5. Хмара, Г.А. Переходные процессы в электроэнергетических системах с распределенными параметрами для студентов направления 140400.62 «Электроэнергетика и электротехника» профиль «Электроснабжение» очной / заочной/заочно-сокращенной формы обучения. [Электронный ресурс]: Методические указания и рекомендации / Г.А. Хмара, Ю.А. Савиных. – Электрон. дан. – Тюмень : ТюмГНГУ, 2013. – 42 с. – Режим доступа: http://e.lanbook.com/book/61167
6. Birman K. Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services (Texts in Computer Science). Springer, 2012 – 752 p.
7. Cachin Ch., Guerraoui R., Rodrigues L. Introduction to Reliable and Secure Distributed Programming. Springer, 2011 – 339 p.
8. Coulouris G., Dollimore J., Kindberg T., Blair G. Distributed Systems: Concepts and Design. Addison-Wesley, 2011 – 1008 p.
9. Hwang K., Dongarra J., Fox G. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Morgan Kaufmann, 2011 – 672 p.

10. Jain N., Verma S., Khare D. Distributed Systems. Acme Learning Private Limited, 2012 – 305 p.