

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»  
Заведующий кафедрой Курманбаев З.Ж.  
д.ф.н., проф.  
(Ф.И.О., ученая степень, звание)  
«    »    20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной системы для торговой  
компании «Мелан»

Специальность Вычислительная техника и программное обеспечение - 580704.00

Выполнил (а) Малайсаров С.Д. ITD-12-2  
(Фамилия и инициалы) группа

Научный руководитель Мусатаева Т.Т., ст.преп.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмурзаева А.С., к.э.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
А.С. « 26 » 05 2016 г.  
(подпись)

по безопасности жизнедеятельности:

Дроздов Н.С., д.т.н., проф.  
(Фамилия и инициалы, ученая степень, звание)  
Н.С. « 19 » 05 2016 г.  
(подпись)

по применению вычислительной техники:

Мусатаева Т.Т., ст.преп.  
(Фамилия и инициалы, ученая степень, звание)  
Т.Т. « 20 » 05 2016 г.  
(подпись)

Нормоконтролер: Мусатаева Т.Т., ст.преп.  
(Фамилия и инициалы, ученая степень, звание)

Т.Т. « 20 » 05 2016 г.  
(подпись)

Рецензент: Бейтурганов Ш.Б., к.ф.-м.н., доцент  
(Фамилия и инициалы, ученая степень, звание)

Ш.Б. « 26 » 05 2016 г.  
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и информационных технологий  
Специальность Вычислительная техника и программное обеспечение  
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Малайсаров Санжар Густенович  
(фамилия, имя, отчество)

Тема проекта Разработка информационной системы для  
торговой компании «Меломан»

утверждена приказом ректора № 148 от «19» октября 2015 г.

Срок сдачи законченной работы «\_\_» \_\_\_\_\_ 20\_\_ г.

Исходные данные к проекту требуемые параметры результатов  
проектирования (исследования) и исходные данные объекта

Разработка информационной системы для торговой компании  
«Меломан»

Перечень подлежащих разработке дипломного проекта вопросов или  
краткое содержание дипломного проекта:

1. Требования к разрабатываемой информационной системе
2. Разработка информационной системы
3. Технико-экономическая часть
4. Безопасность жизнедеятельности

Перечень графического материала (с точным указанием обязательных чертежей)

1. Схема работы АИС
2. Диаграмма компонентов
3. Диаграмма прецедентов
4. Диаграммы классов, развертывания
5. Обзор ER-диаграмма
6. Семанты ER-диаграмм для бизнес-правил
7. Изображения работы программы
8. Таблицы технико-экономической части
9. Таблицы и рисунки по части безопасности жизнедеятельности

Рекомендуемая основная литература

1. Сатимова С.Т. Проектирование баз данных. Методические указания к выполнению лабораторных работ. - Алматы: АИЭС, 2009
2. Microsoft SQL server 2008. Реализация и обслуживание. - М.: "Русская редакция", 2007
3. Шиндт Т. Полный справочник по С#, - М.: "Вильямс", 2004.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БД	Бурдубекое И.Г.	18.03-19.05.16	[Подпись]
Эконом. часть	Бекмезова А.С.	19.04-26.05.16	[Подпись]
Юридический контроль	Мусатаева Т.Т.	31.05.16	[Подпись]





## **Аннотация**

В данной дипломной работе проводится разработка информационной системы для торговой компании «Меломан», занимающейся продажей мультимедийной продукции.

В ходе работы была построена система, позволяющая пользователям вести учет продукции, поиск нужного товара по категориям и отображение справочной информации о продукте. Система спроектирована с использованием новейших технологий и отвечает всем современным требованиям к разработке программного обеспечения.

В завершении были рассмотрены вопросы безопасности жизнедеятельности, также был проведен расчет технико-экономического обоснования проекта, выявляющий целесообразность разработки проекта и его цену на рынке.

## **Annotation**

This diploma work will show the development of an information system for “Meloman” trading company, which is engaged in multimedia products sales.

The work involved creation of the system, allowing users to keep count of products, to search necessary product by categories and to get background information about the product. The system was built using the newest technologies and meets all modern requirements for software developing.

In conclusion, problems of life safety were considered in addition to feasibility study of the project, determining the relevance of the project development and its price on the market.

## **Андатпа**

Ұсынылып отырған дипломдық жұмыста, мультимедиялық өнімдерді сатумен айналысатын, «Меломан» сауда компаниясына арналған ақпараттық жүйе әзірленеді.

Жұмыс барысында қолданушыларға өнімдердің есебін жүргізуге, санаты бойынша керекті тауарды тауып және ол туралы анықтамалық ақпаратты көрсету мүмкіндіктері бар жүйе құрылған.

Жүйе ең соңғы технологияларын қолданумен жобаланған және ақпараттық қамтамасыздандыруды әзірлеудегі барлық талаптарға жауап береді.

Қорытындысында тіршілік қауіпсіздік сұрақтары қарастырылған, сонымен қатар жобаның мақсатын және нарықтағы құнын көрсететін техника-экономикалық негіздеме есептеулері жүргізілді.

## Содержание

Введение.....	12
1 Требования к разрабатываемой информационной системе.....	13
1.1 Назначение системы .....	13
1.2 Требования к данной информационной системе.....	13
1.3 Сравнение с аналогами на рынке .....	16
2 Разработка информационной системы.....	17
2.1 Постановка задачи .....	17
2.2 Этап начальной разработки базы данных.....	17
2.2.1 Анализ предметной области .....	17
2.2.2 Базовые сущности базы данных.....	17
2.2.3 UML диаграммы.....	20
2.2.4 Диаграмма компонентов .....	20
2.2.5 Диаграмма прецедентов .....	21
2.2.6 Диаграмма классов.....	21
2.2.7 Диаграмма развертывания .....	22
2.3 Проектирование базы данных.....	23
2.3.1 Бизнес-правила .....	23
2.3.2 Разработка и построение ER-диаграммы на основании бизнес-правил .....	26
2.3.3 Определение типов каждой сущности.....	27
2.3.4 Определение атрибутов каждой сущности .....	28
2.4 Составление реляционных отношений.....	30
2.5 Расчет места для хранения базы данных .....	33
2.6 Используемые языки программирования.....	38
2.6.1 Среда разработки Visual Studio .....	38
2.6.2 Язык программирования C# .....	38
2.7 Расчетная часть.....	39
2.7.1 Логическое проектирование .....	39
2.7.2 Физическое проектирование.....	39

2.8 Описание интерфейса приложения и его функциональности.....	63
3 Технико-экономическая часть .....	66
3.1 Описание работы и обоснование необходимости .....	66
3.2 Трудовые ресурсы, используемые в работе .....	66
3.3 Оборудование, используемое в работе .....	66
3.4 Программное обеспечение, используемое в работе .....	67
3.5 Сроки реализации проекта .....	68
3.6 Расчет стоимости работы по разработке .....	68
3.6.1 Расчет затрат на оплату труда .....	69
3.6.2 Расчет затрат по социальному налогу .....	72
3.6.3 Расчет амортизационных отчислений .....	73
3.6.4 Расчет затрат на электроэнергию .....	74
3.6.5 Расчет накладных и прочих расходов.....	75
3.6.6 Расчет стоимости по всем статьям затрат и определение структуры затрат .....	76
3.7 Цена интеллектуального труда.....	77
4 Безопасность жизнедеятельности.....	79
4.1 Анализ потенциально опасных и вредных производственных факторов, воздействующих на персонал .....	79
4.2 Анализ помещения.....	81
4.3 Расчет системы кондиционирования .....	83
4.4 Расчет системы автоматического пожаротушения.....	85
Заключение .....	88
Список используемой литературы .....	89
Приложение А .....	90

## **Введение**

В современном мире, с развитием новейших технологий, для каждого предприятия, работающего в сфере продаж, требуется своя система учета продукции, полностью удовлетворяющая требованиям заказчика. Каждый год сильными темпами набирается оборот продаж мультимедийной продукции, такой, как фильмы и музыка, что вызывает за собой необходимость в ведении учета огромных массивов данных.

В настоящее время торговым предприятиям приходится функционировать в сложных финансово-экономических условиях. Залогом успешной деятельности предприятия в таких условиях является максимально эффективная организация деятельности предприятия на всех уровнях. Существующие информационные технологии позволяют автоматизировать деятельность практически на всех уровнях предприятия. Исходя из этого, становится ясно, что автоматизация деятельности становится ключевой задачей, которую необходимо решить руководству для того, чтобы предприятие работало максимально эффективно.

Для выработки и принятия соответствующих складывающейся обстановке решений необходимы информация и знания, которые должны удовлетворять требованиям полноты, достоверности, своевременности (актуальности), полезности. Основополагающую роль в подготовке принятия решений играет его обоснование по имеющейся информации. Ее, как правило, получают из различных внутренних и внешних источников. В интересах выработки адекватного решения используются внутренние информационные ресурсы, которые складываются из отражения деятельности (функционирования) объекта в документах, других видах и способах сбора, обработки, хранения информации.

Целью дипломного проекта является разработка информационной системы, которая предназначена для учета мультимедийной продукции торговой компании «Меломан».

Создание данной системы востребовано и целесообразно, так как применяемые технологии соответствуют нуждам заказчика и имеет ряд преимуществ перед аналогами на рынке.



# **1 Требования к разрабатываемой информационной системе**

## **1.1 Назначение системы**

Важнейшая задача руководителей информационных служб и высшего руководства в сфере информатизации заключается в том, чтобы взаимно увязать деятельность подчиненных им ИТ-подразделений с задачами бизнес-подразделений организации. Однако большинство руководителей, будучи техническими специалистами, не владеют в полной мере искусством формирования и воплощения в жизнь организационной стратегии. Кроме того, сама организация унаследованных систем не позволяет легко интегрировать данные из разных систем для предоставления потребителям новой информации. Поэтому, увязать информационные технологии и бизнес для предоставления бизнес-подразделениям новой информации за счет внесения изменений в оперативные системы – сложная задача. ИТ-специалист может обозначить некоторую ситуацию, как очевидно требующую изменений, но он не знает, как выбрать стратегию или реализовать изменения, связанные с существующей технологией [1].

В настоящее время наблюдается повсеместное внедрение информационных систем в сферу управления предприятием. Это вполне понятно, так как автоматизация управления приносит наиболее ощутимый эффект. В сфере автоматизации крупных предприятий эти системы получили название корпоративных информационных систем. К предприятиям такого типа можно отнести, например, промышленные предприятия с разветвленной структурой производства, торгово-закупочные объединения, предприятия энергоснабжения и др.

Основной целью всех прикладных задач ИТ является достижение максимальной эффективности интеллектуального труда на основе оптимального распределения рабочего времени в интересах реализации поставленных задач. Автоматизация основных функций и процессов в деятельности торговой компании позволяет не только повысить производительность труда персонала и освободить высококвалифицированных специалистов от выполнения многих рутинных операций, но и создать необходимые условия для широкого внедрения маркетингового инструментария в интересах дальнейшего развития бизнеса.

## **1.2 Требования к данной информационной системе**

Основными требованиями к информационной системе являются:

- поддержка высокой скорости;
- поддержка внутренней непротиворечивости данных;
- автоматизированный ввод информации по первичным документам;
- ввод информации в реальном времени;

- автоматическое получение необходимой информации о продуктах;
- наличие удобных утилит просмотра и работы с данными, наглядность представления информации;
- полнота и достоверность хранимой информации;
- поддержка качественного процесса пополнения данных.

Схема работы системы, учитывая вышеперечисленные требования представлена на рисунке 1.

Для выполнения поставленных задач необходим инструментарий для преобразования больших объемов детализированных данных в форму, которая удобна для стратегического планирования и реорганизации бизнеса и необходима специалисту, ответственному за принятие решений. При этом происходит слияние из разных источников различных сведений в требуемую предметно-ориентированную форму с использованием различных методов анализа [2].

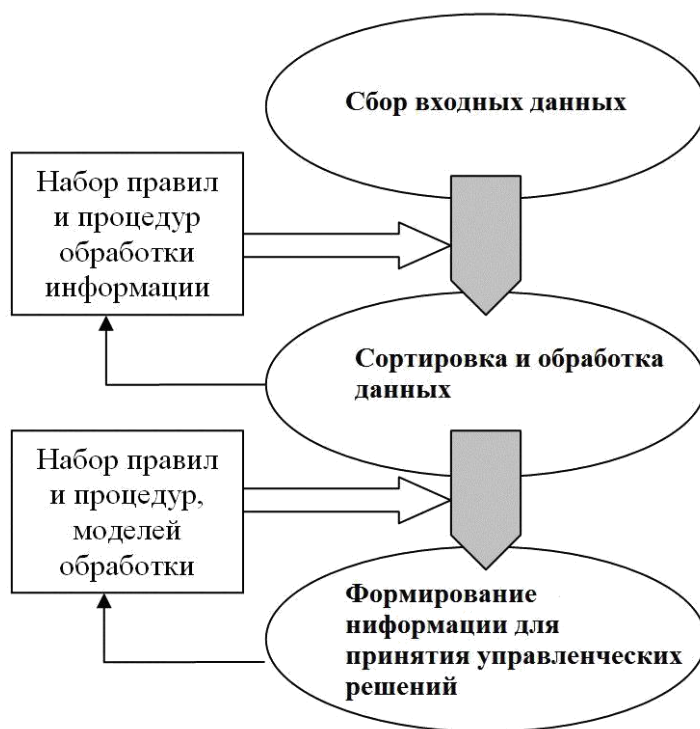


Рисунок 1– Схема работы АИС

Использование информационных технологий предоставляет следующие преимущества:

- а) Экономия рабочего времени.
- б) Отсутствие дублирования.
- в) Избежание ошибок.
- г) Гарантия оперативного обновления форматов представления информации в электронном виде.
- д) Возможность получения информационной выписки.
- е) Оперативное информирование.

ж) Повышение оперативности обработки информации, исключение технических ошибок.

Ядром информационной системы являются хранимые данные. Совокупность информационных объектов, информационно отображающих свойства системы и протекающие в ней процессы, называют информационным пространством. Характерным свойством информационного пространства является его структурированность. Это означает, что выделены его элементы, установлены связи между ними, введены обозначения, элементы и связи упорядочены. Свойство структурированности в разных видах информационных пространств может быть выражено в разной степени.

В качестве элементов структуры информационного пространства выступают единицы информации. Это понятие, которое рассматривается в теории экономических информационных систем (ЭИС), выражает сущностное или смысловое наполнение элемента ИП. Под единицей информации понимают «набор символов, которому придается определенный смысл». Рассматривается система единиц информации, которая имеет довольно сложную иерархическую структуру. Выделяют несколько уровней единиц информации в зависимости от смыслового (семантического) значения, его наполненности. По возрастанию содержательности понятия определены следующие единицы информации: реквизит и составная единица информации (СЕИ), которая включает в себя такие единицы как показатель и база данных.

Модель "сущность-связь" была разработана для моделирования данных, относящихся к некоторым предметным областям или объектам, которые представляют интерес для организации, атрибутов, которыми обладают эти объекты, и отношений между этими объектами [3].

Объекты представляются в модели сущностями, атрибуты – полями, а отношения – связями между сущностями. Можно также сказать, что моделирование в терминах "сущность-связь" представляет собой определенный способ мышления. Поскольку в рассматриваемой модели данных каждый атрибут или поле присутствует единожды, то различные значения в разных полях никак не могут представлять одни и те же данные. Построенная на основе ER-модели компьютерная система для обновления данных, относящихся к определенному экземпляру объекта, просто обновляет одно поле реляционной базы данных. Например, почтовый адрес клиента хранится в определенной ячейке, представляющей собой то единственное место, в котором следует при необходимости обновить почтовый адрес клиента. Это позволяет избежать проблемы нерациональной загрузки системного процессора (system processing unit – SPU) или центрального процессора (central processing unit – CPU) избыточными заданиями [4]. В конечном итоге обработка и обновление данных ускоряются. Все эти трудности, связанные с бункерами данных, неуправляемостью среды оперативных данных и среды поддержки принятия решений, сделали корпоративную модель данных (*enterprise data model*) весьма популярной, однако и ее постигла неудача. Главная идея, лежащая в основе корпоративной

модели данных, состояла в том, чтобы начинать проект с инвентаризации всех данных, которые используются в компьютерной системе. Данные о данных, так называемые *метаданные*, должны были размещаться в специальном хранилище – *репозитории*.

Одним из подходов, позволивших снизить объем избыточных данных в организации, стала методология, в основу которой была положена так называемая модель "сущность-связь". В литературе эта методология получила название *ER-моделирования (entity-relationship (ER) modeling)* [5].

### 1.3 Сравнение с аналогами на рынке

Главным конкурентом разрабатываемой информационной системы является программа 1С, вследствие своей популярности на рынке.

Ниже представлен рейтинг автоматизированных программ, используемых юридическими и физическими лицами:

- 1С-65%
- БЭСТ-7%
- Инфо-бухгалтер - 6%
- Парус - 3%
- Турбо-бухгалтер - 3%
- Средства MS Office - 3%
- Галактика - 2%
- Самописные - 2%
- Другое - 9%

Программа «1С: Бухгалтерия» имеет определенные преимущества, такие как:

- Наличие готовых настроек
- За небольшой период времени, возможно, получение необходимых документов, с использованием определенных форм;
- Отсутствие необходимости в высокой квалификации специалистов;
- Получение программы, в определенный момент максимально отвечающей поставленным задачам.

Недостатками программы «1С: Бухгалтерия» являются:

- Невозможность ведения учета в динамике;
- Относительно невысокая скорость проведения вычислений;
- Для определенной деятельности используют определенную версию, т.е. программа «1С» не является универсальной.

## **2 Разработка информационной системы**

### **2.1 Постановка задачи**

Требуется разработать информационную систему для учета продукции торговой компании «Меломан», позволяющую улучшить работу компании. Система должна предусматривать отражение типа диска продукта, информацию о дате релиза продукта, справочную информацию о режиссерах, актерах, сценаристах, исполнителях и т.д. Информация о продуктах предназначена для просмотра и редактирования администратором и пользователем, но она также исключает возможность модификации, обновления или удаления данных гостем. Таким образом, вводится необходимость разграничения прав доступа

Пользователь системы сможет просматривать данные о хранимой продукции, проводить поиск продуктов по названию, жанрам, типам носителя информации, получать справочную информацию и т.д.

Администратор может вносить новые данные, редактировать имеющиеся, внедрять вспомогательную информацию.

### **2.2 Этап начальной разработки базы данных**

#### **2.2.1 Анализ предметной области**

«Меломан» – казахстанская ритейлинговая, дистрибьюторская и производственная компания, специализирующаяся на производстве и продаже аудио- и видеопродукции, книг, бытовой электроники и сопутствующих товаров.

Компания «Меломан» обладает правами на дистрибьюцию на территории Республики Казахстан продукции ряда крупнейших мировых и российских дистрибьюторов аудио- и видеопродукции: Walt Disney Pictures, Warner Bros. Home Video, Sony Pictures, «Крупный план» («Мосфильм»), Universal Pictures, Paramount Pictures, DreamWorks Pictures, «Союз-видео», Paradise Digital, Sony Music Russia, Universal Music Russia, 20th Century Fox и многих других<sup>[1][2]</sup>.

#### **2.2.2 Базовые сущности базы данных**

Система содержит сведения о продуктах, которые характеризуются следующими параметрами (реализовано в таблице «Продукты (Products)»):

- уникальный код продукта;
- название продукта;
- код типа диска;
- код жанра;
- код исполнителя;
- код композитора;
- код продюсера;
- код звукозаписывающего лейбла;
- код режиссера;
- код сценариста;
- код оператора;
- код кинокомпании;
- дата релиза продукта;
- длительность;
- дата последней поставки;
- наличие;
- цена;

В каждом фильме снимаются актеры, характеризующиеся следующими параметрами (реализовано в таблице «Актеры (Actors)»):

- уникальный код актера;
- отчество;
- имя;
- фамилия;
- страна;

Исполнители песен в музыкальных дисках характеризуются следующими параметрами (реализовано в таблице «Исполнители (Bands)»):

- уникальный код исполнителя;
- имя исполнителя (группы);
- страна;
- код жанра;

У каждого фильма есть выпускающая его кинокомпания, характеризующаяся следующими характеристиками (реализовано в таблице «Компании(Companies)»):

- уникальный код компании;
- название компании;
- страна;

В большинстве фильмов принимает участие собственный композитор, обладающий следующими характеристиками (реализовано в таблице «Композиторы (Composers)»):

- уникальный код композитора;
- отчество;
- имя;



- фамилия;
- страна;

При создании фильма обязательно принимает участие режиссер, который имеет следующие характеристики (реализовано в таблице «Режиссеры (Directors)»):

- уникальный код режиссера;
- отчество;
- имя;
- фамилия;
- страна;

Каждый продукт (диск) можно охарактеризовать типом диска (реализовано в таблице «Типы дисков (DiscTypes)»):

- уникальный код типа диска;
- название типа диска;

Каждый продукт имеет свой определенный жанр. Информация о жанрах представлена следующими характеристиками (реализовано в таблице «Жанры (Genres)»):

- уникальный код жанра;
- название жанра;

В ходе съемок каждого фильма всегда принимает участие оператор, имеющий следующие характеристики (реализовано в таблице «Операторы (Operators)»):

- уникальный код оператора;
- отчество;
- имя;
- фамилия;
- страна;

Для создания каждого фильма должен иметь место продюсер, имеющий следующие характеристики (реализовано в таблице «Продюсеры (Producers)»):

- уникальный код продюсера;
- отчество;
- имя;
- фамилия;
- страна;

Для создания любого фильма обязательно нужен сценарист, имеющий следующие характеристики (реализовано в таблице «Сценаристы (Screenwriters)»):

- уникальный код сценариста;
- отчество;
- имя;

- фамилия;
- страна;

Непосредственное участие в создании любого музыкального продукта принимает звукозаписывающий лейбл, имеющий следующие характеристики (реализовано в таблице «Звукозаписывающие лейблы (RecordLabels)»):

- уникальный код лейбла;
- название;
- страна;

Каждый музыкальный продукт имеет в своем содержании определенное количество песен, имеющих следующие характеристики (реализовано в таблице «Песни (Songs)»):

- код продукта;
- номер песни в продукте;
- название песни;

В фильмах может принимать участие любое количество актеров, которое можно отразить в списке актеров, имеющем следующие характеристики (реализовано в таблице «Список актеров (Starring)»):

- код продукта;
- код актера;

### **2.2.3 UML диаграммы**

Для моделирования статических объектов в объектно-ориентированной концепции UML существует 4 основных типа диаграмм:

1. Диаграмма компонентов.
2. Диаграмма прецедентов
3. Диаграмма классов.
4. Диаграмма развертывания.

### **2.2.4 Диаграмма компонентов**

Диаграмма компонентов показывает набор компонентов и отношений между ними. Компонент – это физически заменяемая часть системы, которая имеет набор интерфейсов и обеспечивает их реализацию.

В данном случае исполняемый модуль Catalog.exe обращается за получением или добавлением информации к базе данных Catalog.mdf. Но обращается не напрямую, а через SQL SERVER. Таким образом SQL SERVER является интерфейсом между программой и базой данных, обеспечивая их взаимодействие.

Диаграмма компонентов представлена на рисунке 2.1.

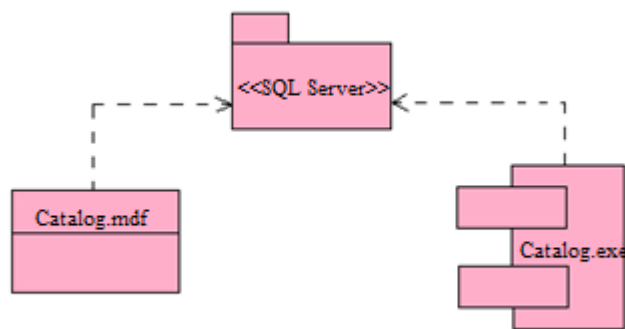


Рисунок 2.1 – Диаграмма компонентов.

### 2.2.5 Диаграмма прецедентов

Для характеристики взаимодействия пользователей с базами данных применяется диаграмма прецедентов или вариантов использования. Диаграмма прецедентов представлена на рисунке 2.2.

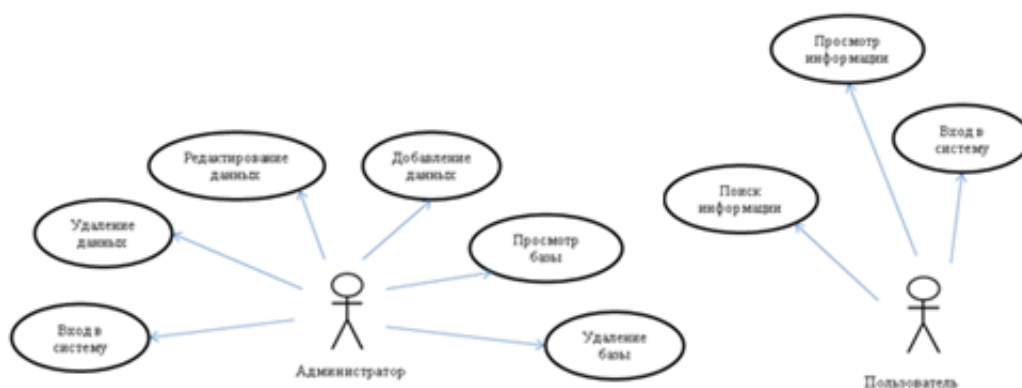


Рисунок 2.2 – Диаграмма прецедентов

Из данной диаграммы видно, что пользоваться базой могут 2 группы пользователей: администратор и пользователь.

### 2.2.6 Диаграмма классов

Диаграмма классов в UML является частным случаем ER диаграммы. ER диаграммы используются для логического проектирования баз данных. Главное их отличие: В ER диаграмме уделяется внимание структуре данных, а в диаграмме классов – поведению классов.

Диаграмма классов представлена на рисунке 2.3.

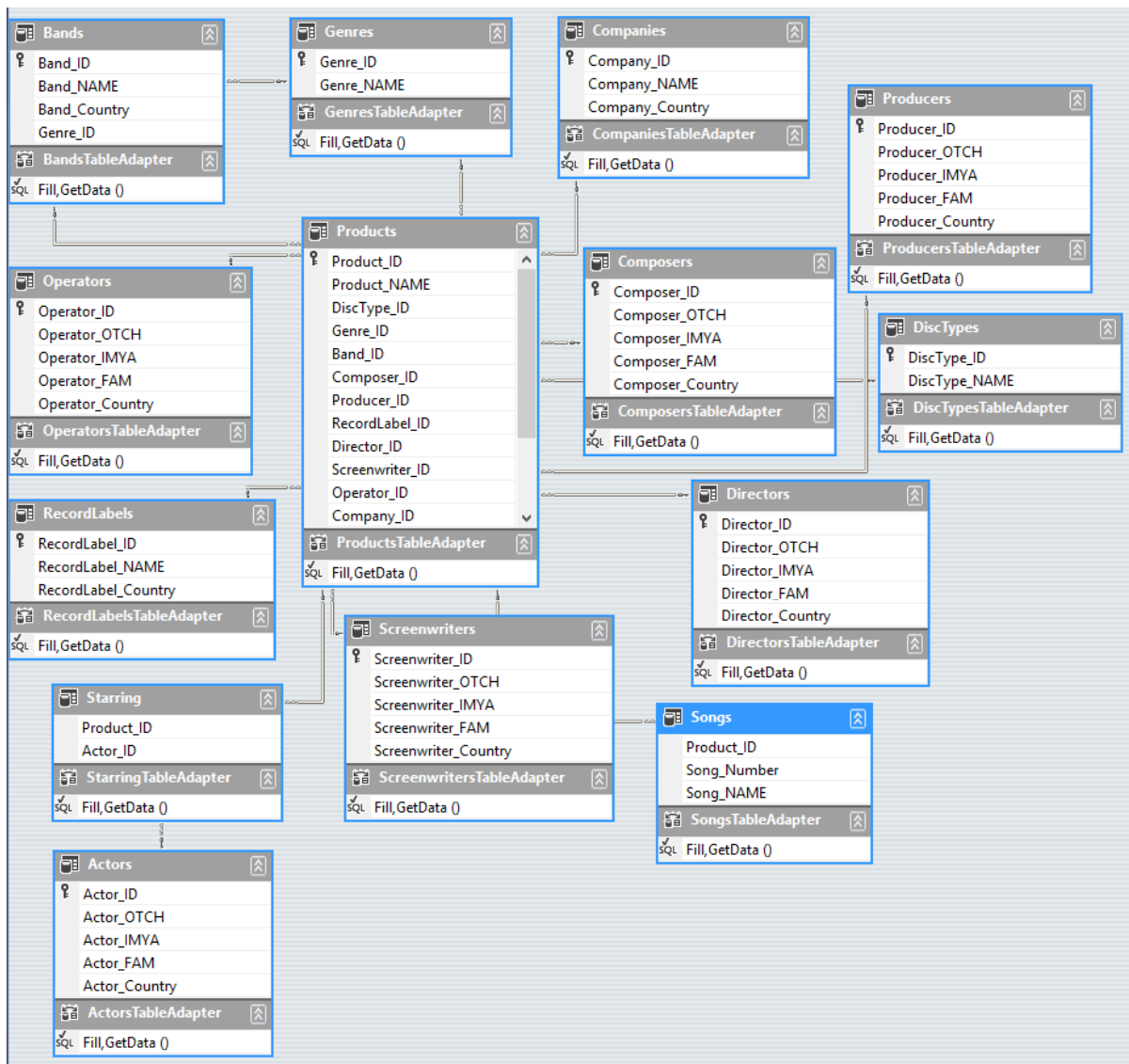


Рисунок 2.3 – Диаграмма классов

## 2.2.7 Диаграмма развертывания

Диаграмма развертывания, представленная на рисунке 2.4, Deployment diagram в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать приложение диаграмма развертывания должна показывать, какие аппаратные компоненты ("узлы") существуют (например сервер базы данных, сервер приложения), какие программные компоненты ("артефакты") работают на каждом узле (например, приложение, хранилище данных), и как различные части этого комплекса соединятся друг с другом (например, JDBC, OLEDB, RMI). Диаграмма развертывания представлена на рисунке 2.4.

Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы

развертывания может концептуально представлять множество физических узлов, таких как кластер серверов баз данных.

Существует два типа узлов:

- узел устройства;
- узел среды выполнения.

Узлы устройств – это физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны. Узел среды выполнения - это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.

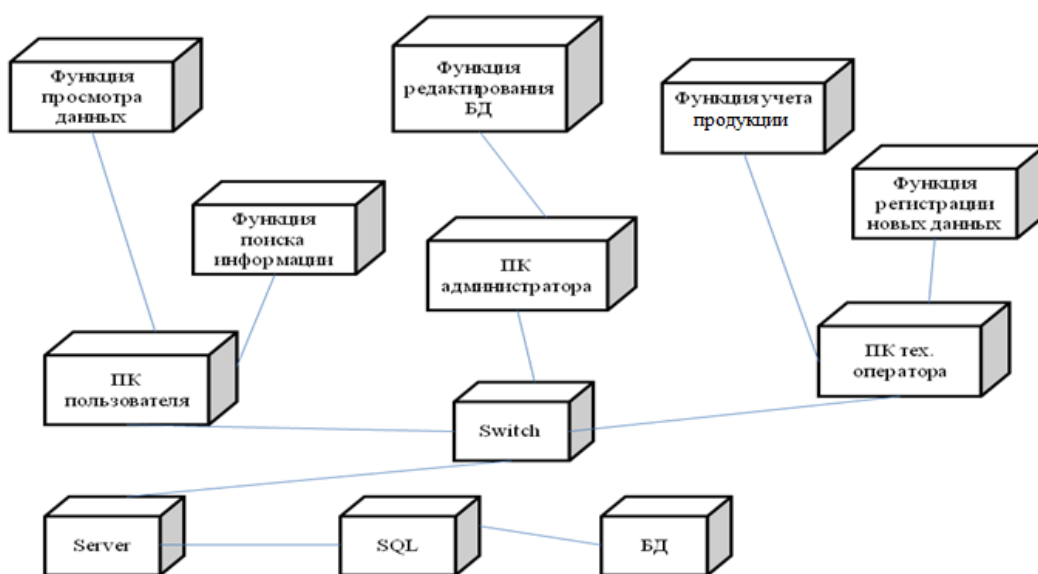


Рисунок 2.4 – Диаграмма развертывания

## 2.3 Проектирование базы данных

### 2.3.1 Бизнес-правила

Вследствие проектирования изначальной ER-модели, необходимо определить связи между сущностями, опираясь на операции. На основе бизнес-правил можно базировать связи между сущностями:

1) Каждый актер может сниматься во многих фильмах и в каждом фильме может принимать участие многие актеры (рисунок 2.5).

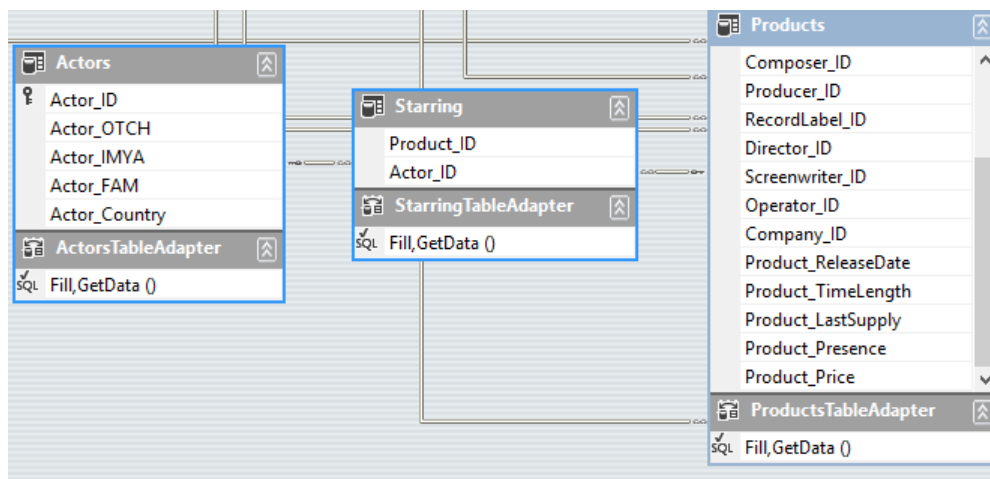


Рисунок 2.5 – Сегмент ER-модели для бизнес-правила 1

2) Каждый музыкальный продукт может включать в себя множество песен, но песня не может принадлежать нескольким продуктам (рисунок 2.6).

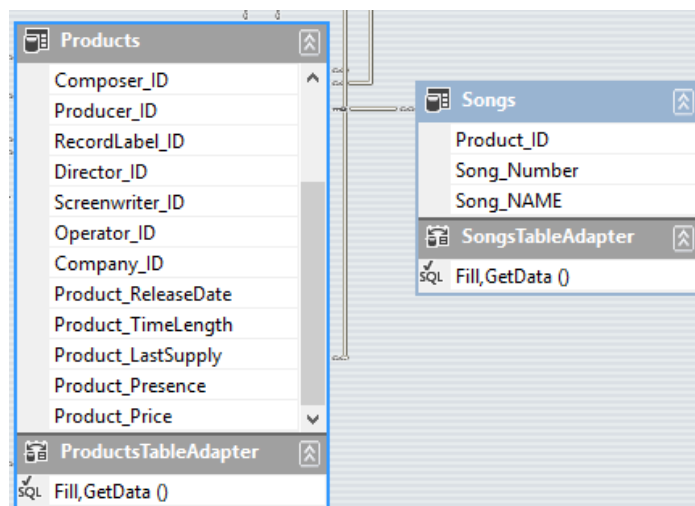


Рисунок 2.6 – Сегмент ER-модели для бизнес-правила 2

3) Жанр могут иметь как исполнители, так и фильмы и музыка (продукты) (рисунок 2.7).



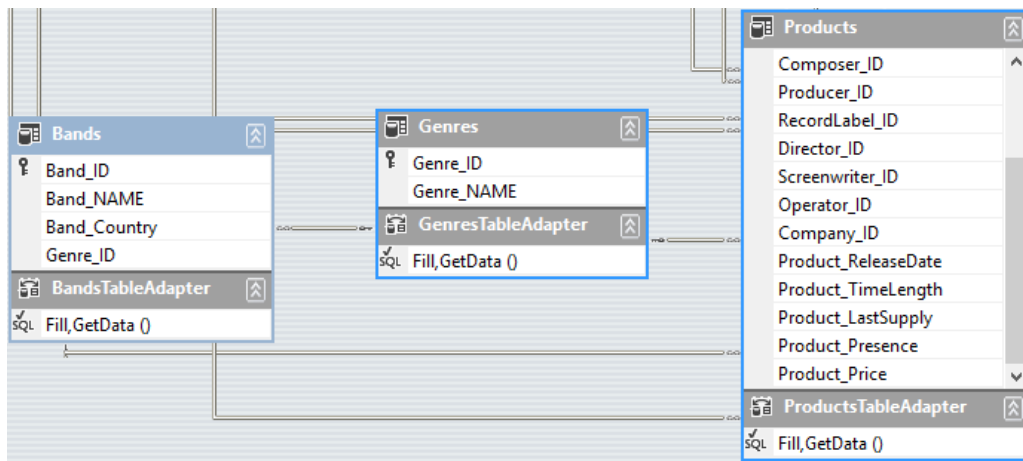


Рисунок 2.7 – Сегмент ER-модели для бизнес-правила 3

4) Каждый фильм может иметь только одного сценариста. Но сценарист может участвовать в создании множества фильмов (рисунок 2.8).

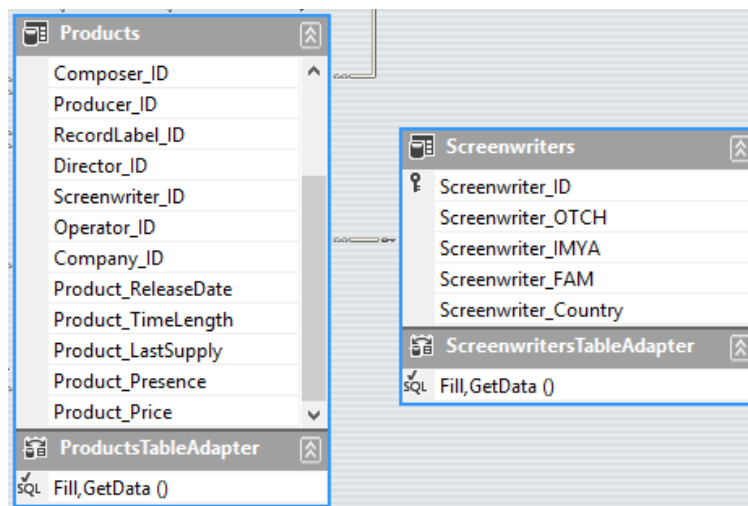


Рисунок 2.8 – Сегмент ER-модели для бизнес-правила 4

5) Фильм (продукт) может выпустить только одна кинокомпания, но одна кинокомпания может выпускать множество фильмов (рисунок 2.9).

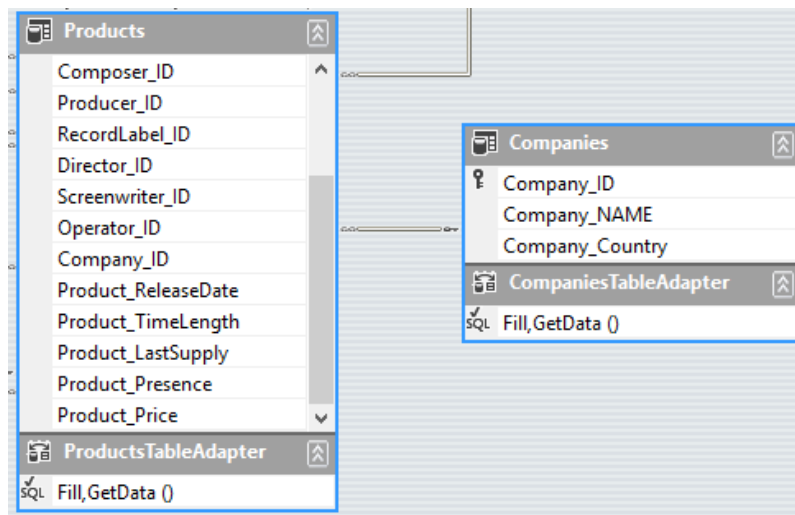


Рисунок 2.9 – Сегмент ER-модели для Бизнес правила 5

### 2.3.2 Разработка и построение ER-диаграммы на основании бизнес-правил

На основе этих бизнес-правил составим конечную концептуальную модель (рисунок 2.10).

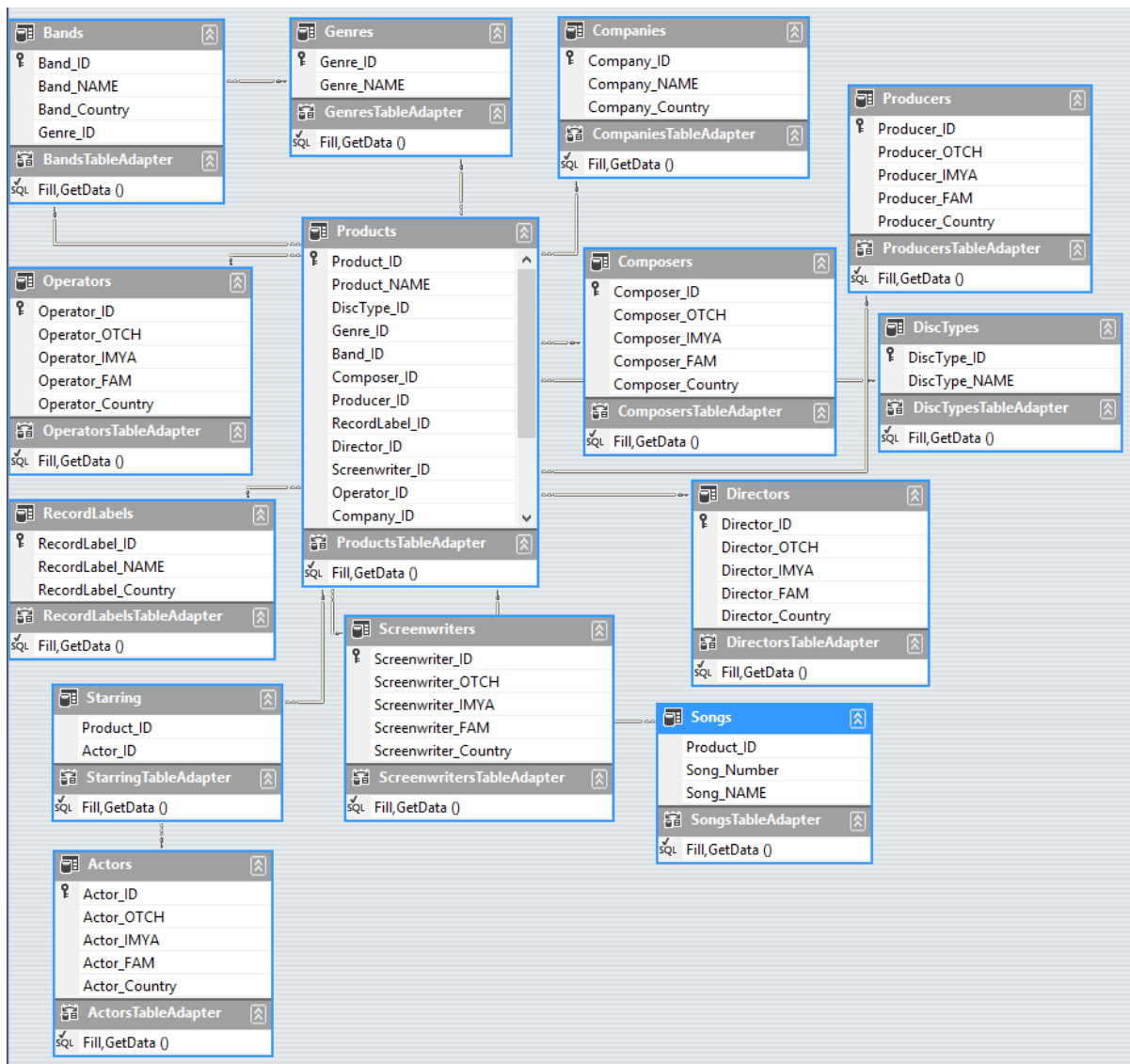


Рисунок 2.10 – Конечная концептуальная модель

### 2.3.3 Определение типов каждой сущности

Исходя из конечной концептуальной модели (рисунок 3.6) определены необходимые сущности, а также их тип. В данной работе будут использоваться сущности, представленные в таблице 3.1.

Таблица 2.1 – Описание назначения таблиц

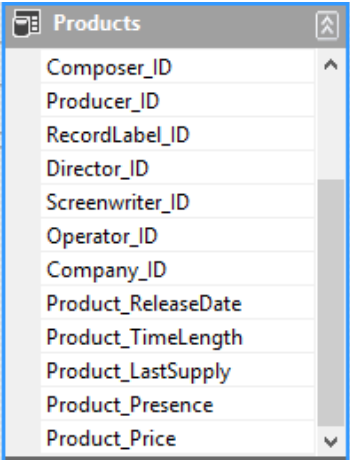
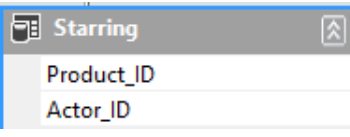

Имя сущности	Описание сущности	Тип сущностей
Actors	Данные об актерах	Сильная
Bands	Данные об исполнителях	Сильная
Companies	Данные о кинокомпаниях	Слабая
Composers	Данные о композиторах	Слабая
Directors	Данные о режиссерах	Слабая
DiscTypes	Данные о типах диска	Сильная

Genres	Данные о жанрах	Сильная
Operators	Данные об операторах	Слабая
Producers	Данные о продюсерах	Слабая
Products	Данные о продуктах	Сильная
RecordLabels	Данные о звукозаписывающих лейблах	Слабая
Screenwriters	Данные о сценаристах	Слабая
Songs	Данные о песнях	Слабая
Starring	Данные о том, какие актеры играют в определенных фильмах	Слабая

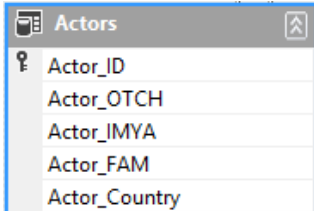
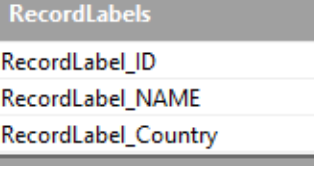
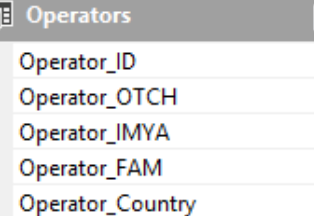
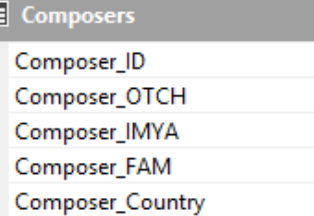
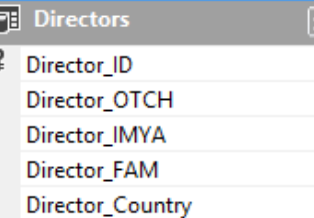
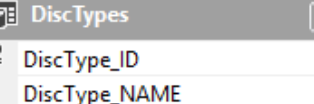
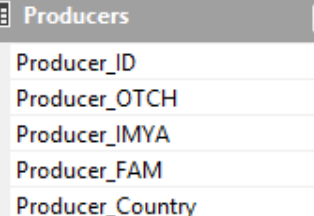
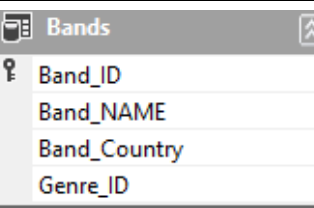
### 2.3.4 Определение атрибутов каждой сущности

Описание сущностей разработанной ER-модели и их атрибутов представлено в таблице 2.2.

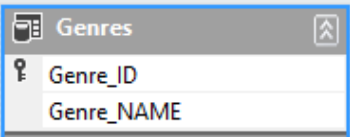

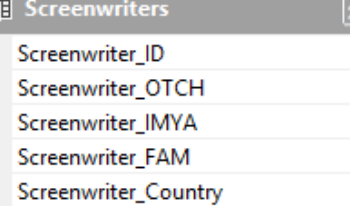
Таблица 2.2 – Описание атрибутов

Сегмент ER модели	Описание
	<p>Таблица Продукты содержит следующие данные:</p> <ul style="list-style-type: none"> <li>Product_ID – уникальный код продукта</li> <li>Product_NAME – название продукта</li> <li>DiscType_ID – код типа диска</li> <li>Genre_ID – код жанра</li> <li>Band_ID – код исполнителя</li> <li>Composer_ID – код композитора</li> <li>Producer_ID – код продюсера</li> <li>RecordLabel_ID – код звукозаписывающего лейбла</li> <li>Director_ID – код режиссера</li> <li>Screenwriter_ID – код сценариста</li> <li>Operator_ID – код оператора</li> <li>Company_ID – код кинокомпании</li> <li>Product_ReleaseDate – дата релиза</li> <li>Product_TimeLength – продолжительность</li> <li>Product_LastSupply – дата последней поставки</li> <li>Product_Presence – наличие</li> <li>Product_Price – цена</li> </ul>
	<p>Таблица Список главных актеров содержит следующие данные:</p> <ul style="list-style-type: none"> <li>Product_ID – код продукта</li> <li>Actor_ID – код актера</li> </ul>
	<p>Таблица Песни содержит следующие данные:</p> <ul style="list-style-type: none"> <li>Product_ID – код продукта</li> <li>Song_Number – номер песни</li> <li>Song_NAME – название песни</li> </ul>

Продолжение таблицы 2.2

	<p>Таблица Актеры содержит следующие данные:          Actor_ID – уникальный код актера          Actor_OTCH – отчество          Actor_IMYA – имя          Actor_FAM – фамилия          Actor_COUNTRY – страна</p>
	<p>Таблица Звукозаписывающие лейблы содержит следующие данные:          RecordLabel_ID          RecordLabel_NAME          RecordLabel_Country</p>
	<p>Таблица Операторы содержит следующие данные:          Operator_ID – уникальный код оператора          Operator_OTCH – отчество          Operator_IMYA – имя          Operator_FAM – фамилия          Operator_COUNTRY – страна</p>
	<p>Таблица Композиторы содержит следующие данные:          Composer_ID – уникальный код композитора          Composer_OTCH – отчество          Composer_IMYA – имя          Composer_FAM – фамилия          Composer_COUNTRY – страна</p>
	<p>Таблица Режиссеры содержит следующие данные:          Director_ID – уникальный код режиссера          Director_OTCH – отчество          Director_IMYA – имя          Director_FAM – фамилия          Director_COUNTRY – страна</p>
	<p>Таблица Типы дисков содержит следующие данные:          DiscType_ID          DiscType_NAME</p>
	<p>Таблица Продюсеры содержит следующие данные:          Producer_ID – уникальный код продюсера          Producer_OTCH – отчество          Producer_IMYA – имя          Producer_FAM – фамилия          Producer_COUNTRY – страна</p>
	<p>Таблица Исполнители содержит следующие данные:          Band_ID – уникальный код исполнителя          Band_NAME – название          Band_Country – страна          Genre_ID – код жанра</p>

## Продолжение таблицы 2.2

	<p>Таблица Жанры содержит следующие данные: Genre_ID – уникальный код жанра Genre_NAME – название жанра</p>
	<p>Таблица Кинокомпании содержит следующие данные: Company_ID – уникальный код кинокомпании Company_NAME – название кинокомпании Company_Country – страна</p>
	<p>Таблица Сценаристы содержит следующие данные: Screenwriter_ID – уникальный код сценариста Screenwriter_OTCH – отчество Screenwriter_IMYA – имя Screenwriter_FAM – фамилия Screenwriter_COUNTRY – страна</p>

## 2.4 Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть).

Отношения приведены в таблицах 2.3–2.16. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Обязательное поле для краткости обозначено `not null`, необязательное – `null`. Типы данных обозначаются так: `bigint` – числовой, `char` – символьный, `Date` – дата.

Таблица 2.3 – Схема отношения Продукт (Product)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код продукта	Product_ID	bigint	Первичный ключ
Название продукта	Product_NAME	char (20)	not null
Код типа диска	DiscType_ID	char (20)	Внешний ключ (к Типам диска)
Код жанра	Genre_ID	char(7)	Внешний ключ (к Жанрам)
Код исполнителя	Band_ID	date	Внешний ключ (к Исполнителям)
Код композитора	Composer_ID	int	Внешний ключ (к Композиторам)
Код продюсера	Producer_ID	int	Внешний ключ (к Продюсерам)
Код звукозаписывающего лейбла	RecordLabel_ID	int	Внешний ключ (к Звукозаписывающим лейблам)
Код режиссера	Director_ID	int	Внешний ключ (к Режиссерам)
Код сценариста	Screenwriter_ID	int	Внешний ключ (к Сценаристам)



Код оператора	Operator_ID	int	Внешний ключ (к Операторам)
Код кинокомпании	Company_ID	int	Внешний ключ (к Кинокомпаниям)
Дата релиза	Product_ReleaseDate	datetime	not null
Продолжительность	Product_TimeLength	varchar(20)	not null
Дата последней поставки	Product_LastSupply	datetime	not null
Наличие	Product_Presence	int	not null
Цена	Product_Price	int	not null

Таблица 2.4 – Схема отношения Список главных актеров (Starring)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код продукта	Product_ID	int	not null
Код актера	Actor_ID	int	not null

Таблица 2.5 – Схема отношения Песни (Songs)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код продукта	Product_ID	int	Внешний ключ (к Продуктам)
Номер песни	Song_Number	int	not null
Название песни	Song_NAME	varchar(20)	Not null

Таблица 2.6 – Схема отношения Сценаристы (Screenwriters)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код сценариста	Screenwriter_ID	Int	Первичный ключ
Отчество	Screenwriter_OTCH	varchar(20)	not null
Имя	Screenwriter_IMYA	varchar(20)	not null
Фамилия	Screenwriter_FAM	varchar(20)	not null
Страна	Screenwriter_COUNTRY	varchar(20)	not null

Таблица 2.7 – Схема отношения Актеры (Actors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код актера	Actor_ID	bigint	Первичный ключ
Отчество	Actor_OTCH	varchar(20)	not null
Имя	Actor_IMYA	varchar(20)	not null
Фамилия	Actor_FAM	varchar(20)	not null
Страна	Actor_COUNTRY	varchar(20)	not null

Таблица 2.8 – Схема отношения Звукозаписывающие лейблы (RecordLabels)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код лейбла	RecordLabel_ID	bigint	Первичный ключ
Название лейбла	RecordLabel_NAME	varchar(20)	not null
Страна	RecordLabel_Country	varchar(20)	not null

Таблица 2.9 – Схема отношения Операторы (Operators)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код оператора	Operator_ID	bigint	Первичный ключ
Отчество	Operator_OTCH	char (20)	not null
Имя	Operator_IMYA	char (20)	not null
Фамилия	Operator_FAM	varchar(20)	not null
Страна	Operator_COUNTRY	varchar(20)	not null

Таблица 2.10 – Схема отношения Композиторы (Composers)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код композитора	Composer_ID	bigint	Первичный ключ
Отчество	Composer_OTCH	varchar(20)	not null
Имя	Composer_IMYA	varchar(20)	not null
Фамилия	Composer_FAM	varchar(20)	not null
Страна	Composer_COUNTRY	varchar(20)	not null

Таблица 2.11 – Схема отношения Режиссеры (Directors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код режиссера	Director_ID	bigint	Первичный ключ
Отчество	Director_OTCH	varchar(20)	not null
Имя	Director_IMYA	varchar(20)	not null
Фамилия	Director_FAM	varchar(20)	not null
Страна	Director_COUNTRY	varchar(20)	not null

Таблица 2.12 – Схема отношения Типы дисков (DiscTypes)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код типа диска	DiscType_ID	bigint	Первичный ключ
Название типа диска	DiscType_NAME	varchar(20)	not null

Таблица 2.13 – Схема отношения Продюсеры (Producers)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код продюсера	Producer_ID	bigint	Первичный ключ
Отчество	Producer_OTCH	varchar(20)	not null
Имя	Producer_IMYA	varchar(20)	not null
Фамилия	Producer_FAM	varchar(20)	not null
Страна	Producer_COUNTRY	varchar(20)	not null

Таблица 2.14 – Схема отношения Исполнители (Bands)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код исполнителя	Band_ID	bigint	Первичный ключ
Название	Band_NAME	varchar(20)	not null
Страна	Band_Country	varchar(20)	not null
Код жанра	Genre_ID	int	Внешний ключ (к Жанрам) not null

Таблица 2.15 – Схема отношения Жанры (Genres)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код жанра	Genre_ID	bigint	Первичный ключ
Название жанра	Genre_NAME	varchar(20)	not null

Таблица 2.16 – Схема отношения Кинокомпании (Companies)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Уникальный код кинокомпании	Company_ID	bigint	Первичный ключ
Название кинокомпании	Company_NAME	varchar(20)	not null
Страна	Company_Country	varchar(20)	not null

## 2.5 Расчет места для хранения базы данных

До запуска базы данных требуется определить необходимые размеры изначальной базы данных и ее последующего роста.

Объём внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные (М<sub>д</sub>). Наиболее существенным обычно является М<sub>д</sub>.

Объём памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объёмом самих данных, поэтому может не учитываться. В проекте рассчитывается предполагаемый максимальный объем памяти занимаемой БД (таблицы 2.17–2.30).

Таблица 2.17 – Расчет физической памяти для таблицы Продукт (Product)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код продукта	Product_ID	bigint	8
Название продукта	Product_NAME	char (20)	20
Код типа диска	DiscType_ID	char (20)	20
Код жанра	Genre_ID	char(7)	7
Код исполнителя	Band_ID	date	8
Код композитора	Composer_ID	int	4
Код продюсера	Producer_ID	int	4
Код звукозаписывающего лейбла	RecordLabel_ID	int	4
Код режиссера	Director_ID	int	4
Код сценариста	Screenwriter_ID	int	4
Код оператора	Operator_ID	int	4
Код кинокомпании	Company_ID	int	4
Дата релиза	Product_ReleaseDate	datetime	8
Продолжительность	Product_TimeLength	varchar(20)	20
Дата последней поставки	Product_LastSupply	datetime	8
Наличие	Product_Presence	int	4
Цена	Product_Price	int	4

Общая длина строки: 135 байт

Число строк: ~ 10 000 000

Общий объем требуемой памяти: ~ 1 350 000 000 байт

Таблица 2.18 – Расчет физической памяти для таблицы Список главных актеров (Starring)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Код продукта	Product_ID	bigint	8
Код актера	Actor_ID	int	4

Общая длина строки: 12 байт

Число строк: ~ 10 000 000

Общий объем требуемой памяти: ~ 12 000 000 байт

Таблица 2.19 – Расчет физической памяти для таблицы Песни (Songs)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Код продукта	Product_ID	bigint	8
Номер песни	Song_Number	int	4
Название песни	Song_NAME	varchar(20)	20

Общая длина строки: 32 байт

Число строк: ~ 50 000 000

Общий объем требуемой памяти: ~ 160 000 000 байт

Таблица 2.20 – Расчет физической памяти для таблицы Сценаристы (Screenwriters)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код сценариста	Screenwriter_ID	bigint	8
Отчество	Screenwriter_OTCH	varchar(20)	20
Имя	Screenwriter_IMYA	varchar(20)	20
Фамилия	Screenwriter_FAM	varchar(20)	20
Страна	Screenwriter_COUNTRY	varchar(20)	20

Общая длина строки: 88 байт

Число строк: ~ 2000

Общий объем требуемой памяти: ~ 176 000 байт

Таблица 2.21 – Расчет физической памяти для таблицы Актеры (Actors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код актера	Actor_ID	bigint	8
Отчество	Actor_OTCH	varchar(20)	20
Имя	Actor_IMYA	varchar(20)	20
Фамилия	Actor_FAM	varchar(20)	20
Страна	Actor_COUNTRY	varchar(20)	20

Общая длина строки: 88 байт

Число строк: ~ 20 000

Общий объем требуемой памяти: ~ 1 760 000 байт

Таблица 2.22 – Расчет физической памяти для таблицы Звукозаписывающие лейблы (RecordLabels)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код лейбла	RecordLabel_ID	bigint	8
Название лейбла	RecordLabel_NAME	varchar(20)	20
Страна	RecordLabel_Country	varchar(20)	20

Общая длина строки: 48 байт

Число строк: ~ 500

Общий объем требуемой памяти: ~ 24 000 байт

Таблица 2.23 – Расчет физической памяти для таблицы Операторы (Operators)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код оператора	Operator_ID	bigint	8
Отчество	Operator_OTCH	char (20)	20
Имя	Operator_IMYA	char (20)	20
Фамилия	Operator_FAM	varchar(20)	20
Страна	Operator_COUNTRY	varchar(20)	20

Общая длина строки: 88 байт

Число строк: ~ 2000

Общий объем требуемой памяти: ~ 176 000 байт

Таблица 2.24 – Расчет физической памяти для таблицы Композиторы (Composers)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код композитора	Composer_ID	bigint	8
Отчество	Composer_OTCH	varchar(20)	20
Имя	Composer_IMYA	varchar(20)	20
Фамилия	Composer_FAM	varchar(20)	20
Страна	Composer_COUNTRY	varchar(20)	20

Общая длина строки: 88 байт

Число строк: ~ 10 000

Общий объем требуемой памяти: ~ 880 000 байт

Таблица 2.25 – Расчет физической памяти для таблицы Режиссеры (Directors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код режиссера	Director_ID	bigint	8
Отчество	Director_OTCH	varchar(20)	20
Имя	Director_IMYA	varchar(20)	20
Фамилия	Director_FAM	varchar(20)	20
Страна	Director_COUNTRY	varchar(20)	20

Общая длина строки: 88 байт

Число строк: ~ 10 000

Общий объем требуемой памяти: ~ 880 000 байт

Таблица 2.26 – Расчет физической памяти для таблицы Типы дисков (DiscTypes)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код типа диска	DiscType_ID	bigint	8
Название типа диска	DiscType_NAME	varchar(20)	20

Общая длина строки: 28 байт

Число строк: ~ 10

Общий объем требуемой памяти: ~ 280 байт

Таблица 2.27 – Расчет физической памяти для таблицы Продюсеры (Producers)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код продюсера	Producer_ID	bigint	8
Отчество	Producer_OTCH	varchar(20)	20
Имя	Producer_IMYA	varchar(20)	20
Фамилия	Producer_FAM	varchar(20)	20
Страна	Producer_COUNTRY	varchar(20)	20

Общая длина строки: 88 байт

Число строк: ~ 2000

Общий объем требуемой памяти: ~ 176 000 байт

Таблица 2.28 – Расчет физической памяти для таблицы Исполнители (Bands)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код исполнителя	Band_ID	bigint	8
Название	Band_NAME	varchar(20)	20
Страна	Band_Country	varchar(20)	20
Код жанра	Genre_ID	int	4



Общая длина строки: 52 байт  
 Число строк: ~ 1 000 000  
 Общий объем требуемой памяти: ~ 52 000 000 байт

Таблица 2.29 – Расчет физической памяти для таблицы Жанры (Genres)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код жанра	Genre_ID	bigint	8
Название жанра	Genre_NAME	varchar(20)	20

Общая длина строки: 28 байт  
 Число строк: ~ 2000  
 Общий объем требуемой памяти: ~ 56 000 байт

Таблица 2.30 – Расчет физической памяти для таблицы Кинокомпании (Companies)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Длина, байт</i>
Уникальный код кинокомпании	Company_ID	bigint	8
Название кинокомпании	Company_NAME	varchar(20)	20
Страна	Company_Country	varchar(20)	20

Общая длина строки: 48 байт  
 Число строк: ~ 10 000  
 Общий объем требуемой памяти: ~ 480 000 байт

В результате табличных расчетов, можно получить приблизительный максимальный занимаемый объем базы данных. Данный объем будет равен сумме всех объемов таблиц:

$$\begin{aligned}
 M_d &= 1350000000 + 12000000 + 160000000 + 1760000 + 24000 + 176000 + \\
 &880000 + 880000 + 280 + 176000 + 52000000 + 56000 + 480000 = \\
 &= 1578608280 \text{ (байт)} \sim 1,47 \text{ (Гб)}
 \end{aligned}$$

Объём памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объёмом самих данных, поэтому может не учитываться. Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных.

Исходя из данных, полученных во время физического проектирования можно сделать вывод, что размер базы данных за первые 5 лет работы системы будет равен 1,47 Гб. Через 5 лет этот размер будет увеличен в 2 раза. Перед запуском базы данных необходимо выставить необходимые размеры первоначальной базы и ее роста.

## **2.6 Используемые языки программирования**

### **2.6.1 Среда разработки Visual Studio**

Microsoft Visual Studio – линейка продуктов компании Майкрософт, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом.

Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

### **2.6.2 Язык программирования C#**

Язык программирования C# был создан в конце 1990-х годов и стал частью общей .NET-стратегии Microsoft. Впервые он увидел свет в качестве б-версии в середине 2000 года. Главным архитектором C# был Андерс Хейлсберг (Anders Hejlsberg) – один из ведущих специалистов в области языков программирования, получивший признание во всем мире. Достаточно сказать, что в 1980-х он был автором весьма успешного продукта Turbo Pascal, изящная реализация которого установила стандарт для всех будущих компиляторов. C# непосредственно связан с C, C++ и Java. И это не случайно. Эти три языка -- самые популярные и самые любимые языки программирования в мире. Более того, почти все профессиональные программисты сегодня знают C и C++, и большинство знает Java. Поскольку C# построен на прочном, понятном фундаменте, то переход от этих "фундаментальных" языков к "надстройке" происходит без особых усилий со стороны программистов. Так как Андерс Хейлсберг не собирался изобретать свое "колесо", он сосредоточился на введении усовершенствований и новшеств.

Свой синтаксис C# во многом унаследовал от C++ и Java. Разработчики, имеющие опыт написания приложений на этих языках, найдут в C# много знакомых черт. Но вместе с тем он является во многом новаторским -

атрибуты, делегаты и события, прекрасно вписанные в общую идеологию языка, прочно заняли место в сердцах .NET - разработчиков. Их введение позволило применять принципиально новые приемы программирования. От С язык С# унаследовал синтаксис, многие ключевые слова и операторы. Кроме того, С# построен на улучшенной объектной модели, определенной в С++. С# и Java связаны между собой несколько сложнее. Как упоминалось выше, Java также является потомком С и С++. У него тоже общий с ними синтаксис и сходная объектная модель. Подобно Java С# предназначен для создания переносимого кода.

## **2.7 Расчетная часть**

### **2.7.1 Логическое проектирование**

Задача логического проектирования состоит в создании реляционных табличных структур на языке DDL.

Ниже приведен код, создающий табличное пространство:

```
CREATE DATABASE diplom
ON PRIMARY
(NAME = diplom, FILENAME='G:\Sanzhar\diplom.mdf', size =100,maxsize=200,
filegrowth=5 mb)
LOG ON
(NAME=education_log, FILENAME='G:\Sanzhar\diplom.ldf', size=100, maxsize=200,
filegrowth=5 mb);
```

### **2.7.2 Физическое проектирование**

#### *Обоснование выбора СУБД*

Для разработки базы данных была выбрана СУБД SQL Server 2012. SQL Server является всеобъемлющим, интегрированным сквозным решением, которое наделяет пользователей организации безопасной, надежной, и продуктивной платформой для обработки промышленной информации и приложений, касающихся интеллектуальных ресурсов предприятия. SQL Server 2012 предоставляет мощные, знакомые инструменты для профессионалов информационных технологий так же, как и для работников информационной сферы, уменьшая сложность создания, развёртывания, управления и использования данных предприятия и аналитических приложений на платформах от мобильных устройств до информационных систем предприятия. Благодаря исчерпывающему набору функций, взаимодействию с существующими системами и автоматизации типовых задач, SQL Server 2012 предоставляет полное решение в области хранения данных для предприятий всех масштабов.

*Меры по обеспечению безопасности*

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа. Для настоящей базы данных на основании анализа предметной области создаются 2 учетные записи: администратора и пользователя. Администратору дается полный контроль над БД, в права пользователя вносятся ограничения на изменение и обновление таблиц.

```
CREATE LOGIN administrator WITH PASSWORD = 'administrator';
CREATE USER administrator FOR LOGIN administrator;
EXEC sp_addrolemember Db_owner, administrator
```

```
CREATE LOGIN users WITH PASSWORD = 'users';
CREATE USER users FOR LOGIN users;
```

```
deny update, insert, delete on Bands to users
deny update, insert, delete on Genres to users
deny update, insert, delete on Companies to users
deny update, insert, delete on Producers to users
deny update, insert, delete on Operators to users
deny update, insert, delete on Products to users
deny update, insert, delete on Composers to users
deny update, insert, delete on DiscTypes to users
deny update, insert, delete on RecordLabels to users
deny update, insert, delete on Starring to users
deny update, insert, delete on Screenwriters to users
deny update, insert, delete on Directors to users
deny update, insert, delete on Songs to users
deny update, insert, delete on Actors to users
```

```
grant select on Bands to users
grant select on Genres to users
grant select on Companies to users
grant select on Producers to users
grant select on Operators to users
grant select on Products to users
grant select on Composers to users
grant select on DiscTypes to users
grant select on RecordLabels to users
grant select on Starring to users
grant select on Screenwriters to users
grant select on Directors to users
grant select on Songs to users
grant select on Actors to users
```

Ниже приведены скрипты для создания таблиц:

```
create table DiscTypes(  
DiscType_ID int identity(1,1) primary key,  
DiscType_NAME varchar(20)  
)
```

```
create table Genres(  
Genre_ID int identity(1,1) primary key,  
Genre_NAME varchar(20)  
)
```

```
create table RecordLabels(  
RecordLabel_ID int identity(1,1) primary key,  
RecordLabel_NAME varchar(20),  
RecordLabel_Country varchar(20)  
)
```

```
create table Companies(  
Company_ID int identity(1,1) primary key,  
Company_NAME varchar(20),  
Company_Country varchar(20),  
)
```

```
create table Directors(  
Director_ID int identity(1,1) primary key,  
Director_OTCH varchar(20),  
Director_IMYA varchar(20),  
Director_FAM varchar(20),  
Director_Year datetime,  
Director_Country varchar(20)  
)
```

```
create table Operators(  
Operator_ID int identity(1,1) primary key,  
Operator_OTCH varchar(20),  
Operator_IMYA varchar(20),  
Operator_FAM varchar(20),  
Operator_Year datetime,  
Operator_Country varchar(20)  
)
```

```
create table Producers(  
Producer_ID int identity(1,1) primary key,  
Producer_OTCH varchar(20),  
Producer_IMYA varchar(20),  
Producer_FAM varchar(20),  
Producer_Year datetime,  
Producer_Country varchar(20)  
)
```

```
create table Screenwriters(  

```

```
Screenwriter_ID int identity(1,1) primary key,  
Screenwriter_OTCH varchar(20),  
Screenwriter_IMYA varchar(20),  
Screenwriter_FAM varchar(20),  
Screenwriter_Year datetime,  
Screenwriter_Country varchar(20)  
)
```

```
create table Composers(  
Composer_ID int identity(1,1) primary key,  
Composer_OTCH varchar(20),  
Composer_IMYA varchar(20),  
Composer_FAM varchar(20),  
Composer_Year datetime,  
Composer_Country varchar(20)  
)
```

```
create table Actors(  
Actor_ID int identity(1,1) primary key,  
Actor_OTCH varchar(20),  
Actor_IMYA varchar(20),  
Actor_FAM varchar(20),  
Actor_Year datetime,  
Actor_Country varchar(20)  
)
```

```
create table Bands(  
Band_ID int identity(1,1) primary key,  
Band_NAME varchar(20),  
Band_Country varchar(20),  
Genre_ID int foreign key references Genres(Genre_ID)  
)
```

```
create table Artists(  
Artist_ID int identity(1,1) primary key,  
Artist_OTCH varchar(20),  
Artist_IMYA varchar(20),  
Artist_FAM varchar(20),  
Artist_Year datetime,  
Artist_Country varchar(20),  
Genre_ID int foreign key references Genres(Genre_ID),  
Band_ID int foreign key references Bands(Band_ID)  
)
```

```
create table Products(  
Product_ID int primary key,  
Product_NAME varchar(40),  
DiscType_ID int foreign key references DiscTypes(DiscType_ID),  
Genre_ID int foreign key references Genres(Genre_ID),  
Band_ID int foreign key references Bands(Band_ID),  
Artist_ID int foreign key references Artists(Artist_ID),
```

```

Composer_ID int foreign key references Composers(Composer_ID),
Producer_ID int foreign key references Producers(Producer_ID),
RecordLabel_ID int foreign key references RecordLabels(RecordLabel_ID),
Director_ID int foreign key references Directors(Director_ID),
Screenwriter_ID int foreign key references Screenwriters(Screenwriter_ID),
Operator_ID int foreign key references Operators(Operator_ID),
Company_ID int foreign key references Companies(Company_ID),
Product_ReleaseDate datetime,
Product_TimeLength varchar(20),
Product_LastSupply datetime,
Product_Presence int,
Product_Price int
)

```

```

create table Songs(
Product_ID int foreign key references Products(Product_ID),
Song_Number int,
Song_NAME varchar(20)
)

```

```

create table Starring(
Product_ID int foreign key references Products(Product_ID),
Actor_ID int foreign key references Actors(Actor_ID)
)

```

Созданная БД содержит все еще пустые табличные структуры. Для заполнения базы данных первоначальными значениями был использован SQL скрипт, написанный вручную:

```

insert into Actors values
(null,'Джей Кей','Симмонс','США'),
(null,'Майлз','Теллер','США'),
(null,'Мэттью','Макконахи','США'),
(null,'Джессика','Честейн','США'),
(null,'Майкл','Кейн','Великобритания'),
(null,'Энн','Хэтэуэй','США'),
(null,'Жан','Рено','Франция'),
(null,'Натали','Портман','США'),
(null,'Гэри','Олдман','Великобритания'),
(null,'Дэни','Айелло','США'),
(null,'Том','Хэнкс','США'),
(null,'Кристиан','Бэйл','Великобритания'),
(null,'Хит','Леджер','Австралия'),
(null,'Клинт','Иствуд','США'),
(null,'Илай','Уоллак','США'),
(null,'Ли','Ван Клиф','США'),
(null,'Лиам','Нисон','США'),
(null,'Бен','Кингсли','Великобритания'),
(null,'Рэйф','Файнс','Великобритания'),
(null,'Робин','Райт','США'),

```

(null,'Гэри','Синиз','США'),  
(null,'Энтони','Перкинс','США'),  
(null,'Джанет','Ли','США'),  
(null,'Вера','Майлз','США'),  
(null,'Роберт','Де Ниро','США'),  
(null,'Джо','Пеши','США'),  
(null,'Шэрон','Стоун','США'),  
(null,'Рэй','Лиотта','США'),  
(null,'Лоррейн','Бракко','США'),  
(null,'Джек','Николсон','США'),  
(null,'Луиза','Флетчер','США'),  
(null,'Брэд','Дуриф','США'),  
(null,'Киану','Ривз','США'),  
(null,'Лоуренс','Фишборн','США'),  
(null,'Керри-Энн','Мосс','Канада'),  
(null,'Хьюго','Уивинг','Австралия'),  
(null,'Чарльз','Чаплин','США'),  
(null,'Арнольд','Шверценеггер','США'),  
(null,'Эдвард','Ферлонг','США'),  
(null,'Линда','Хэмилтон','США'),  
(null,'Роберт','Патрик','США'),  
(null,'Дэвид','Морс','США'),  
(null,'Бонни','Хант','США'),  
(null,'Майкл Кларк','Дункан','США'),  
(null,'Леонардо','Ди Каприо','США'),  
(null,'Мэтт','Деймон','США'),  
(null,'Марлон','Брандо','США'),  
(null,'Роберт','Дюваль','США'),  
(null,'Мартин','Шин','США'),  
(null,'Майкл Джей','Фокс','США'),  
(null,'Кристофер','Ллойд','США'),  
(null,'Джейми','Фокс','США'),  
(null,'Кристоф','Вальц','Австрия'),  
(null,'Шелли','Дювалл','США'),  
(null,'Дэнни','Ллойд','США'),  
(null,'Аль','Пачино','США'),  
(null,'Джеймс','Каан','США'),  
(null,'Малкольм','Макдауэлл','Великобритания'),  
(null,'Кир','Дулли','США'),  
(null,'Александрович','Алексей','Чадов','Россия'),  
(null,'Ингеборга','Дапкунайте','Литва'),  
(null,'Сергеевич','Сергей','Бодров','Россия'),  
(null,'Энтони','Хопкинс','Великобритания'),  
(null,'Джоди','Фостер','США'),  
(null,'Скотт','Гленн','США');

insert into Starring values

(100001,30),(100001,31),(100001,32),  
(100002,33),(100002,34),(100002,35),(100002,36),  
(100003,63),(100003,64),(100003,65),  
(100004,37),



(100005,38),(100005,39),(100005,40),(100005,41),  
(100006,42),(100006,43),(100006,44),(100006,11),  
(100007,45),(100007,46),(100007,30),  
(100008,47),(100008,48),(100008,49),(100008,34),  
(100009,52),(100009,53),  
(100010,54),(100010,55),(100010,30),  
(100011,56),(100011,57),(100011,47),(100011,48),  
(100012,58),  
(100013,59),  
(100014,60),(100014,61),(100014,62),  
(100015,1),(100015,2),  
(100016,12),(100016,13),(100016,9),  
(100017,50),(100017,51),  
(100018,3),(100018,4),(100018,5),(100018,6),  
(100019,7),(100019,8),(100019,9),(100019,10),  
(100020,11),  
(100021,14),(100021,15),(100021,16),  
(100022,17),(100022,18),(100022,19),  
(100023,20),(100023,21),(100023,11),  
(100024,22),(100024,23),(100024,24),  
(100025,25),(100025,26),(100025,27),  
(100026,25),(100026,26),(100026,28),(100026,29);

insert into Songs values

(200001,1,'Intro'),  
(200001,2,'Last Living Souls'),  
(200001,3,'Kids With Guns'),  
(200001,4,'O Green World'),  
(200001,5,'Dirty Harry'),  
(200001,6,'Feel Good Inc.'),  
(200001,7,'El Manana'),  
(200001,8,'Every Planet We'),  
(200001,9,'November Has Come'),  
(200001,10,'All Alone'),  
(200001,11,'White Light'),  
(200001,12,'Dare'),  
(200001,13,'Fire Coming Out'),  
(200001,14,'Dont Get Lost In'),  
(200001,15,'Demon Days'),

(200002,1,'Black Dog'),  
(200002,2,'Rock n Roll'),  
(200002,3,'The Battle Of'),  
(200002,4,'Stairway To Heaven'),  
(200002,5,'Misty Mountain Hop'),  
(200002,6,'Four Sticks'),  
(200002,7,'Going To Cali'),  
(200002,8,'When The Levee'),

(200003,1,'Speak To Me'),  
(200003,2,'Breathe'),

(200003,3,'On the Run'),  
(200003,4,'Time'),  
(200003,5,'The Great Gig'),  
(200003,6,'Money'),  
(200003,7,'Us and them'),  
(200003,8,'Any Colour You Like'),  
(200003,9,'Brain Damage'),  
(200003,10,'Eclipse'),

(200004,1,'Airbag'),  
(200004,2,'Paranoid Android'),  
(200004,3,'Subterranean'),  
(200004,4,'Exit Music'),  
(200004,5,'Let Down'),  
(200004,6,'Karma Police'),  
(200004,7,'Fitter Happier'),  
(200004,8,'Electioneering'),  
(200004,9,'Climbing Up The'),  
(200004,10,'No Surprises'),  
(200004,11,'Lucky'),  
(200004,12,'The Tourist');

insert into Screenwriters values

(null, 'Дамьен','Шазель',null),  
(null, 'Кристофер','Нолан',null),  
(null, 'Люк','Бессон',null),  
(null, 'Роберт','Родэт',null),  
(null, 'Серджо','Леоне',null),  
(null, 'Стивен','Заиллян',null),  
(null, 'Эрик','Рот',null),  
(null, 'Джозеф','Стефано',null),  
(null, 'Николас','Пиледжи',null),  
(null,'Бо','Голдман','США'),  
(null,'Лана','Вачовски','США'),  
(null,'Тед','Тэлли','США'),  
(null,'Чарльз','Чаплин','США'),  
(null,'Джеймс','Кэмерон','США'),  
(null,'Фрэнк','Дарабонт','США'),  
(null,'Уильям','Монахан','США'),  
(null,'Джон','Милиус','США'),  
(null,'Роберт','Земекис','США'),  
(null,'Квентин','Тарантино','США'),  
(null,'Стэнли','Кубрик','США'),  
(null,'Марио','Пьюзо','США'),  
( 'Октябринович','Алексей','Балабанов','Россия');

insert into RecordLabels values

('Virgin records','Великобритания'),  
( 'Atlantic records','США'),  
( 'Harvest','Великобритания'),  
( 'Parlophone','Великобритания');

insert into

Products(Product\_ID,Product\_NAME,DiscType\_ID,Genre\_ID,Band\_ID,Composer\_ID,Producer\_ID,RecordLabel\_ID,Product\_ReleaseDate,Product\_TimeLength,Product\_LastSupply,Product\_Presence,Product\_Price) values

(200001,'Demon Days',1,14,1,null,23,1,'24-05-2005','50:44',getdate(),6,1500),  
(200002,'Led Zeppelin IV',2,15,2,null,24,2,'08-11-1971','42:38',getdate(),3,3000),  
(200003,'The Dark Side Of The Moon',3,16,3,null,25,3,'24-03-1973','43:00',getdate(),5,1200),  
(200004,'OK Computer',4,17,4,null,26,4,'21-05-1997','52:27',getdate(),15,2000);

insert into

Products(Product\_ID,Product\_NAME,DiscType\_ID,Genre\_ID,Director\_ID,Producer\_ID,Screen writer\_ID,Operator\_ID,Composer\_ID,Company\_ID,Product\_ReleaseDate,Product\_TimeLength,Product\_LastSupply,Product\_Presence,Product\_Price) values

(100001,'Пролетая над гнездом кукушки',7,2,13,10,10,11,8,1,'19-11-1975','133 минуты',getdate(),12,2000),  
(100002,'Матрица',8,12,14,11,11,12,9,2,'31-04-1999','136 минут',getdate(),7,7000),  
(100003,'Молчание ягнят',7,9,15,12,12,13,10,3,'14-02-1991','118 минут',getdate(),1,3000),  
(100004,'Огни большого города',7,11,16,13,13,14,11,4,'30-01-1931','87 минут',getdate(),6,4000),  
(100005,'Терминатор 2: Судный день',7,12,17,14,14,15,12,5,'03-07-1991','136 минут',getdate(),2,2000),  
(100006,'Зеленая миля',7,2,18,15,15,16,13,2,'10-12-1999','188 минут',null,0,4000),  
(100007,'Отступники',8,7,11,16,16,10,10,2,'06-10-2006','151 минута',getdate(),5,9000),  
(100008,'Апокалипсис сегодня',7,4,1,17,17,17,14,6,'15-08-1979','179 минут',getdate(),23,3000),  
(100009,'Джанго освобожденный',7,6,4,19,19,9,5,8,'25-12-2012','165 минут',getdate(),1,2000),  
(100010,'Сияние',7,9,19,20,20,19,15,2,'23-05-1980','143 минуты',getdate(),5,2000),  
(100011,'Крестный отец',8,2,1,21,21,20,16,9,'24-03-1972','175 минут',getdate(),6,9000),  
(100012,'Заводной апельсин',8,2,19,20,20,19,15,2,'19-12-1971','136 минут',getdate(),8,9000),  
(100013,'Космическая одиссея 2001 года',8,12,19,20,20,21,17,10,'03-04-1968','142 минуты',null,0,9000),  
(100014,'Война',7,4,3,22,22,22,null,14,'14-03-2002','120 минут',getdate(),6,3000),  
(100015,'Одержимость',7,1,12,8,1,1,1,11,'16-01-2014','106 минут',getdate(),8,3000),  
(100016,'Темный рыцарь',7,5,6,2,2,5,2,2,'18-07-2008','152 минуты',getdate(),10,3000),  
(100017,'Назад в будущее',7,13,9,18,18,18,6,7,'01-10-1985','116 минут',getdate(),3,3000),  
(100018,'Интерстеллар',7,12,6,2,2,2,2,9,'05-11-2014','169 минут',getdate(),3,3000),  
(100019,'Леон',7,3,7,3,3,3,3,12,'14-09-1994','133 минуты',getdate(),7,3000),  
(100020,'Спаси рядового Райана',7,4,8,4,4,4,4,9,'24-07-1998','169 минут',getdate(),3,3000),  
(100021,'Хороший, плохой, злой',7,6,5,9,5,6,5,16,'15-12-1966','177 минут',getdate(),3,3000),  
(100022,'Список Шиндлера',7,7,8,4,6,4,4,7,'15-12-1993','197 минут',getdate(),11,3000),  
(100023,'Форрест Гамп',7,8,9,8,7,7,6,9,'06-07-1994','142 минуты',getdate(),12,3000),  
(100024,'Психо',7,9,10,7,8,8,7,13,'08-09-1960','109 минут',getdate(),10,3000),  
(100025,'Казино',7,2,11,5,9,9,null,7,'22-11-1995','178 минут',getdate(),2,3000),  
(100026,'Славные парни',7,2,11,6,9,10,null,2,'19-09-1990','146 минут',getdate(),6,3000);

insert into Producers values

(null,'Gorillaz',null,'Великобритания'),  
(null,'Джимми','Пейдж','Великобритания'),  
(null,'Pink Floyd',null,'Великобритания'),  
(null,'Найджел','Годрич','Великобритания');

```
insert into Producers
values
(null,'Джейсон','Блам','США'),
(null,'Кристофер','Нолан',null),
(null,'Патрис','Леду','Франция'),
(null,'Стивен','Спилберг',null),
(null,'Барбара','Де Фина',null),
(null,'Ирвин','Уинклер',null),
(null,'Альфред','Хичкок',null),
(null,'Стив','Старки',null),
(null,'Альберто','Гримальди',null),
(null,'Майкл','Дуглас','США'),
(null,'Джоэл','Сильвер','США'),
(null,'Кеннет','Утг','США'),
(null,'Чарльз','Чаплин','США'),
(null,'Джеймс','Кэмерон','США'),
(null,'Фрэнк','Дарабонт','США'),
(null,'Брэд','Питт','США'),
(null,'Фрэнсис Форд','Коппола','США'),
(null,'Боб','Гейл','США'),
(null,'Стейси','Шер','США'),
(null,'Стэнли','Кубрик','США'),
(null,'Альберт','Рудди','США'),
('Михайлович','Сергей','Сельянов','Россия');
```

```
insert into Operators values
(null, 'Шэрон','Мэир', null),
(null, 'Хойте','Ван Хойтема', null),
(null, 'Тьерри','Арбогаст', null),
(null, 'Януш','Камински', null),
(null, 'Уолли','Пфистер', null),
(null, 'Тонино Делли','Колли', null),
(null, 'Дон','Берджесс', null),
(null, 'Джон Л.','Расселл', null),
(null, 'Роберт','Ричардсон', null),
(null, 'Михаэль','Балльхаус', null),
(null,'Хаскелл','Уэкслер','США'),
(null,'Билл','Поуп','США'),
(null,'Так','Фудзимото','США'),
(null,'Гордон','Поллок','США'),
(null,'Адам','Гринберг','США'),
(null,'Дэвид','Таттерсолл','Великобритания'),
(null,'Витторио','Стораро','Италия'),
(null,'Дин','Канди','США'),
(null,'Джон','Олкотт','Великобритания'),
(null,'Гордон','Уиллис','США'),
(null,'Джеффри','Ансуорт','США'),
('Валентинович','Сергей','Астахов','Россия');
```

```
insert into Genres values
```

('Электроника'),  
( 'Хард рок'),  
( 'Прогрессивный рок'),  
( 'Альтернативный рок');

insert into Genres values

('Музыкальный'),  
( 'Драма'),  
( 'Боевик'),  
( 'Военный'),  
( 'Супергеройский'),  
( 'Вестерн'),  
( 'Исторический'),  
( 'Трагикомедия'),  
( 'Триллер'),  
( 'Криминальный'),  
( 'Комедия'),  
( 'Фантастика'),  
( 'Приключенческий');

insert into DiscTypes

values

('Audio CD'),  
( 'Audio DVD'),  
( 'MP3 CD'),  
( 'MP3 DVD'),  
( 'Vinyl'),  
( 'Video CD'),  
( 'Video DVD'),  
( 'Video Blu-Ray');

insert into Directors

values

(null, 'Фрэнсис Форд', 'Коппола', 'США'),  
(null, 'Дэвид', 'Финчер', 'США'),  
( 'Октябринович', 'Алексей', 'Балабанов', 'Россия'),  
(null, 'Квентин', 'Тарантино', 'США'),  
(null, 'Серджо', 'Леоне', 'Италия'),  
(null, 'Кристофер', 'Нолан', 'США'),  
(null, 'Люк', 'Бессон', 'Франция'),  
(null, 'Стивен', 'Спилберг', 'США'),  
(null, 'Роберт', 'Земекис', 'США'),  
(null, 'Альфред', 'Хичкок', 'США'),  
(null, 'Мартин', 'Скорсезе', 'США'),  
(null, 'Дамьен', 'Шазель', 'США'),  
(null, 'Милош', 'Форман', 'США'),  
(null, 'Эндрю', 'Вачовски', 'США'),  
(null, 'Джонатан', 'Демми', 'США'),  
(null, 'Чарльз', 'Чаплин', 'США'),  
(null, 'Джеймс', 'Кэмерон', 'США'),  
(null, 'Фрэнк', 'Дарабонт', 'США'),

(null,'Стэнли','Кубрик','США');

insert into Composers values

(null,'Джастин','Гервиц',null),  
(null,'Ханс','Циммер',null),  
(null,'Эрик','Серра',null),  
(null,'Джон','Уильямс',null),  
(null,'Эннио','Морриконе',null),  
(null,'Алан','Сильвестри',null),  
(null,'Бернард','Херрманн',null),  
(null,'Джек','Ницше','США'),  
(null,'Дон','Дэвис','США'),  
(null,'Говард','Шор','Канада'),  
(null,'Чарльз','Чаплин','США'),  
(null,'Брэд','Фидель','США'),  
(null,'Томас','Ньюман','США'),  
(null,'Кармайн','Коппола','США'),  
(null,'Венди','Карлос','США'),  
(null,'Нино','Рота','Италия'),  
(null,'Рихард','Штраус','Германия');

insert into Companies

values

('Fantasy Films','США'),  
( 'Warner Bros.','США'),  
( 'Orion Pictures','США'),  
( 'United Artists','США'),  
( 'TriStar Pictures','США'),  
( 'American Zoetrope','США'),  
( 'Universal Pictures','США'),  
( 'Columbia Pictures','США'),  
( 'Paramount Pictures','США'),  
( 'Metro-Goldwyn-Mayer','США'),  
( 'Bold Films','США'),  
( 'Gaumont','Франция'),  
( 'Shamley Productions','США'),  
( 'СТВ','Россия'),  
( 'Constantin Film','Италия');

insert into Bands values

('Gorillaz','Великобритания',14),  
( 'Led Zeppelin','Великобритания',15),  
( 'Pink Floyd','Великобритания',16),  
( 'Radiohead','Великобритания',17);

Для запроса информации из БД были созданы специальные представления:

1. Служит для представления информации об актерах, играющих в конкретных фильмах. Результат работы представления «starring\_view» представлен на рисунке 2.10.

```

create view starring_view
as
select Pr.Product_NAME as 'Фильм', (Ac.Actor_IMYA+' '+Ac.Actor_FAM) as 'Актер' from
Starring St
join Actors Ac
on Ac.Actor_ID = St.Actor_ID
join Products Pr
on Pr.Product_ID = St.Product_ID

```

	Фильм	Актер
1	Пролетая над гнездом кукушки	Джек Николсон
2	Пролетая над гнездом кукушки	Луиза Флетчер
3	Пролетая над гнездом кукушки	Брэд Дуриф
4	Матрица	Киану Ривз
5	Матрица	Лоуренс Фишборн
6	Матрица	Керри-Энн Мосс
7	Матрица	Хьюго Уивинг
8	Молчание ягнят	Энтони Хопкинс
9	Молчание ягнят	Джоди Фостер
10	Молчание ягнят	Скотт Гленн
11	Огни большого города	Чарльз Чаплин
12	Терминатор 2: Судный день	Арнольд Шварценеггер
13	Терминатор 2: Судный день	Эдвард Ферлонг
14	Терминатор 2: Судный день	Линда Хэмилтон
15	Терминатор 2: Судный день	Роберт Патрик

Рисунок 2.10 – Результат работы представления «starring\_view»

2. Служит для представления информации о песнях, содержащихся в конкретном музыкальном продукте. Результат работы представления «songs\_view» представлен на рисунке 2.11.

```

create view songs_view
as
select Pr.Product_NAME as 'Альбом', S.Song_Number as '№', S.Song_NAME as 'Песня' from
Songs S
join Products Pr
on Pr.Product_ID = S.Product_ID

```

	Альбом	№	Песня
1	Demon Days	1	Intro
2	Demon Days	2	Last Living Souls
3	Demon Days	3	Kids With Guns
4	Demon Days	4	O Green World
5	Demon Days	5	Dirty Harry
6	Demon Days	6	Feel Good Inc.
7	Demon Days	7	El Manana
8	Demon Days	8	Every Planet We
9	Demon Days	9	November Has Come
10	Demon Days	10	All Alone
11	Demon Days	11	White Light
12	Demon Days	12	Dare
13	Demon Days	13	Fire Coming Out
14	Demon Days	14	Dont Get Lost In

Рисунок 2.11 – Результат работы представления «songs\_view»

3. Служит для представления информации о музыкальных продуктах. Результат работы представления «music\_view» представлен на рисунке 2.12.

```

create view music_view
as
select Pr.Product_ID as 'ID продукта',Pr.Product_NAME as 'Название',Disc.DiscType_NAME
as 'Тип',Genre.Genre_NAME as 'Жанр',Band.Band_NAME as
'Исполнитель',(Prod.Producer_IMYA+' '+Prod.Producer_FAM) as
'Продюсер',Rec.RecordLabel_NAME as 'Звукозаписывающая
компания',Pr.Product_ReleaseDate as 'Дата релиза',Pr.Product_TimeLength as
'Длительность',Pr.Product_LastSupply as 'Последний завоз',Pr.Product_Presence as
'Наличие',Pr.Product_Price as 'Цена'
from Products Pr
join DiscTypes Disc on
Disc.DiscType_ID = Pr.DiscType_ID
join Genres Genre on
Genre.Genre_ID = Pr.Genre_ID
join Bands Band on
Band.Band_ID = Pr.Band_ID
join Producers Prod on
Prod.Producer_ID = Pr.Producer_ID
join RecordLabels Rec on
Rec.RecordLabel_ID = Pr.RecordLabel_ID

```

	ID продукта	Название	Тип	Жанр	Исполнитель	Продюсер	Звукозаписывающая компания
1	200001	Demon Days	Audio CD	Электроника	Gorillaz	NULL	Virgin records
2	200002	Led Zeppelin IV	Audio DVD	Хард рок	Led Zeppelin	Джимми Пейдж	Atlantic records
3	200003	The Dark Side Of The Moon	MP3 CD	Прогрессивный рок	Pink Floyd	NULL	Harvest
4	200004	OK Computer	MP3 DVD	Альтернативный рок	Radiohead	Найджел Годрич	Parlophone

Рисунок 2.12 – Результат работы представления «music\_view»

4. Служит для представления информации о видео продуктах. Результат работы представления «films\_view» представлен на рисунке 2.13.



```

create view films_view
as
select Pr.Product_ID as 'ID продукта',Pr.Product_NAME as 'Название',Disc.DiscType_NAME
as 'Тип',Genre.Genre_NAME as 'Жанр',(Dir.Director_IMYA+' '+Dir.Director_FAM) as
'Режиссер',(Prod.Producer_IMYA+' '+Prod.Producer_FAM) as
'Продюсер',(Screen.Screenwriter_IMYA+' '+Screen.Screenwriter_FAM) as
'Сценарист',(Oper.Operator_IMYA+' '+Oper.Operator_FAM) as
'Оператор',(Comp.Composer_IMYA+' '+Comp.Composer_FAM) as
'Композитор',Company.Company_NAME as 'Кинокомпания',Pr.Product_ReleaseDate as 'Дата
релиза',Pr.Product_TimeLength as 'Длительность',Pr.Product_LastSupply as 'Последний
заказ',Pr.Product_Presence as 'Наличие',Pr.Product_Price as 'Цена'
from Products Pr
join DiscTypes Disc on
Disc.DiscType_ID = Pr.DiscType_ID
join Genres Genre on
Genre.Genre_ID = Pr.Genre_ID
join Directors Dir on
Dir.Director_ID = Pr.Director_ID
join Producers Prod on
Prod.Producer_ID = Pr.Producer_ID
join Screenwriters Screen on
Screen.Screenwriter_ID = Pr.Screenwriter_ID
join Operators Oper on
Oper.Operator_ID = Pr.Operator_ID
left join Composers Comp on
Comp.Composer_ID = Pr.Composer_ID
join Companies Company on
Company.Company_ID = Pr.Company_ID

```

	ID продукта	Название	Тип	Жанр	Режиссер	Продюсер	Сценарист
1	100001	Пролетая над гнездом кукушки	Video DVD	Драма	Милош Форман	Майкл Дуглас	Бо Голдман
2	100002	Матрица	Video Blu-Ray	Фантастика	Эндрю Вачовски	Джозел Сильвер	Лана Вачовски
3	100003	Молчание ягнят	Video DVD	Триллер	Джонатан Демми	Кеннет Утт	Тед Тэлли
4	100004	Огни большого города	Video DVD	Комедия	Чарльз Чаплин	Чарльз Чаплин	Чарльз Чаплин
5	100005	Терминатор 2: Судный день	Video DVD	Фантастика	Джеймс Кэмерон	Джеймс Кэмерон	Джеймс Кэмерон
6	100006	Зеленая миля	Video DVD	Драма	Фрэнк Дарабонт	Фрэнк Дарабонт	Фрэнк Дарабонт
7	100007	Отступники	Video Blu-Ray	Исторический	Мартин Скорсезе	Брэд Питт	Уильям Монро
8	100008	Апокалипсис сегодня	Video DVD	Военный	Фрэнсис Форд Coppola	Фрэнсис Форд Coppola	Джон Милиус
9	100009	Джанго освобожденный	Video DVD	Вестерн	Квентин Тарантино	Стейси Шер	Квентин Тарантино
10	100010	Сияние	Video DVD	Триллер	Стэнли Кубрик	Стэнли Кубрик	Стэнли Кубрик

Рисунок 2.13 – Результат работы представления «films\_view»

5. Служит для представления информации о режиссерах. Результат работы представления «directors\_view» представлен на рисунке 2.14.

```

create view directors_view
as
select Director_ID as 'ID режиссера',Director_OTCH as 'Отчество',Director_IMYA as
'Имя',Director_FAM as 'Фамилия', Director_Country as 'Страна' from Directors

```

	ID режиссера	Отчество	Имя	Фамилия	Страна
1	1	NULL	Фрэнсис Форд	Коппола	США
2	2	NULL	Дэвид	Финчер	США
3	3	Октябринович	Алексей	Балабанов	Россия
4	4	NULL	Квентин	Тарантино	США
5	5	NULL	Серджо	Леоне	Италия
6	6	NULL	Кристофер	Нолан	США
7	7	NULL	Люк	Бессон	Франция
8	8	NULL	Стивен	Спилберг	США
9	9	NULL	Роберт	Земекис	США
10	10	NULL	Альфред	Хичкок	США
11	11	NULL	Мартин	Скорсезе	США

Рисунок 2.14 – Результат работы представления «directors\_view»

6. Служит для представления информации об исполнителях. Результат работы представления «bands\_view» представлен на рисунке 2.15.

```
create view bands_view
as
select B.band_id as 'ID исполнителя',B.band_name as 'Название',B.band_country as 'Страна',
G.genre_name as 'Жанр' from Bands B
join Genres G
on G.genre_id = B.genre_id
```

	ID исполнителя	Название	Страна	Жанр
1	1	Gorillaz	Великобритания	Электроника
2	2	Led Zeppelin	Великобритания	Хард рок
3	3	Pink Floyd	Великобритания	Прогрессивный рок
4	4	Radiohead	Великобритания	Альтернативный рок

Рисунок 2.15 – Результат работы представления «bands\_view»

7. Служит для представления информации об актерах. Результат работы представления «actors\_view» представлен на рисунке 2.16.

```
create view actors_view
as
select Actor_ID as 'ID актера',Actor_OTCH as 'Отчество',Actor_ИМЯ as 'Имя',Actor_FAM as
'Фамилия',Actor_Country as 'Страна' from Actors
```

	ID актера	Отчество	Имя	Фамилия	Страна
1	1	NULL	Джей Кей	Симмонс	США
2	2	NULL	Майлз	Теллер	США
3	3	NULL	Мэттью	Макконахи	США
4	4	NULL	Джессика	Честейн	США
5	5	NULL	Майкл	Кейн	Великобритания
6	6	NULL	Энн	Хэтзуэй	США
7	7	NULL	Жан	Рено	Франция
8	8	NULL	Натали	Портман	США
9	9	NULL	Гэри	Олдман	Великобритания
10	10	NULL	Дэни	Айелло	США
11	11	NULL	Том	Хэнкс	США

Рисунок 2.16 – Результат работы представления «bands\_view»

### Процедуры:

1. Служит для добавления нового списка актеров в главных ролях в конкретном видео продукте.

```
create procedure New_Starring
(@Product_NAME varchar(20),
@Actor_IMYA varchar(20),
@Actor_FAM varchar(20))
as
exec New_Actor null,@Actor_IMYA,@Actor_FAM,null
insert into Starring values
(dbo.GetProduct(@Product_NAME),dbo.GetActor(@Actor_IMYA,@Actor_FAM))
```

2. Служит для добавления новой песни для конкретного музыкального продукта.

```
create procedure New_Song
(@Product_NAME varchar(20),
@Song_NUMBER int,
@Song_NAME varchar(20))
as
insert into Songs values
(dbo.GetProduct(@Product_NAME),@Song_NUMBER,@Song_NAME)
```

3. Служит для добавления нового сценариста.

```
create procedure New_Screenwriter
(@otch varchar(20),
@imyа varchar(20),
@fam varchar(20),
@country varchar(20))
as
if not exists (select Screenwriter_Imya, Screenwriter_Fam from Screenwriters where
Screenwriter_Imya=@imyа and Screenwriter_Fam=@fam)
begin
```

```

        insert into Screenwriters values
        (@otch,@imya,@fam,@country)
    end

```

4. Служит для добавления нового звукозаписывающего лейбла.

```

create procedure New_RecordLabel
(@name varchar(20),
@country varchar(20))
as
if not exists (select RecordLabel_NAME from RecordLabels where
RecordLabel_NAME=@name)
    begin
        insert into RecordLabels values
        (@name,@country)
    end

```

5. Служит для добавления нового музыкального продукта.

```

create procedure AddMusic
(@DiscType varchar(10),
@Product_ID int,
@Product_NAME varchar(40),

@Genre_NAME varchar(20),
@Band_NAME varchar(20),
@Producer_IMYA varchar(20),
@Producer_FAM varchar(20),
@RecordLabel_NAME varchar(20),
@Product_ReleaseDate datetime,
@Product_TimeLength varchar(10),
@Product_LastSupply datetime,
@Product_Presence int,
@Product_Price int)
as
if not exists (select Product_NAME from Products where Product_NAME=@Product_NAME)
    begin
        exec New_Genre @Genre_NAME
        exec New_Band @Band_NAME,null,@Genre_NAME
        exec New_Producer null,@Producer_IMYA,@Producer_FAM,null
        exec New_RecordLabel @RecordLabel_NAME,null

        insert into
Products(DiscType_ID,Product_ID,Product_NAME,Genre_ID,Band_ID,Producer_ID,RecordLabel_ID,Product_ReleaseDate,Product_TimeLength,Product_LastSupply,Product_Price) values

        (@DiscType,@Product_ID,@Product_NAME,dbo.GetGenre(@Genre_NAME),dbo.GetBand(@Band_NAME),dbo.GetProducer(@Producer_IMYA,@Producer_FAM),dbo.GetRecordLabel(@RecordLabel_NAME),@Product_ReleaseDate,@Product_TimeLength,@Product_LastSupply,@Product_Presence,@Product_Price)
    end

```

end

6. Служит для добавления нового видео продукта.

```
create procedure AddFilm
```

```
((@DiscType varchar(10),
@Product_ID int,
@Product_NAME varchar(40),
@Genre_NAME varchar(20),
@Director_IMYA varchar(20),
@Director_FAM varchar(20),
@Producer_IMYA varchar(20),
@Producer_FAM varchar(20),
@Screenwriter_IMYA varchar(20),
@Screenwriter_FAM varchar(20),
@Operator_IMYA varchar(20),
@Operator_FAM varchar(20),
@Composer_IMYA varchar(20),
@Composer_FAM varchar(20),
@Company_NAME varchar(20),
@Product_ReleaseDate datetime,
@Product_TimeLength varchar(10),
@Product_LastSupply datetime,
@Product_Presence int,
@Product_Price int)
as
if not exists (select Product_NAME from Products where Product_NAME=@Product_NAME)
begin
    exec New_Genre @Genre_NAME
    exec New_Director null,@Director_IMYA,@Director_FAM,null
    exec New_Producer null,@Producer_IMYA,@Producer_FAM,null
    exec New_Screenwriter null,@Screenwriter_IMYA,@Screenwriter_FAM,null
    exec New_Operator null,@Operator_IMYA,@Operator_FAM,null
    exec New_Composer null,@Composer_IMYA,@Composer_FAM,null
    exec New_Company @Company_NAME

    insert into
Products(DiscType_ID,Product_ID,Product_NAME,Genre_ID,Director_ID,Producer_ID,Screen
writer_ID,Operator_ID,Composer_ID,Company_ID,Product_ReleaseDate,Product_TimeLength,P
roduct_LastSupply,Product_Presence,Product_Price) values

    (@DiscType,@Product_ID,@Product_NAME,dbo.GetGenre(@Genre_NAME),dbo.GetD
irector(@Director_IMYA,@Director_FAM),dbo.GetProducer(@Producer_IMYA,@Producer_F
AM),dbo.GetScreenwriter(@Screenwriter_IMYA,@Screenwriter_FAM),dbo.GetOperator(@Ope
rator_IMYA,@Operator_FAM),dbo.GetComposer(@Composer_IMYA,@Composer_FAM),dbo.
GetCompany(@Company_NAME),@Product_ReleaseDate,@Product_TimeLength,@Product_L
astSupply,@Product_Presence,@Product_Price)
end
```

7. Служит для добавления нового продюсера.

```

create procedure New_Producer
(@otch varchar(20),
@imya varchar(20),
@fam varchar(20),
@country varchar(20))
as
if not exists (select Producer_Imya, Producer_Fam from Producers where Producer_Imya=@imya
and Producer_Fam=@fam)
begin
insert into Producers values
(@otch,@imya,@fam,@country)
end

```

8. Служит для добавления нового оператора.

```

create procedure New_Operator
(@otch varchar(20),
@imya varchar(20),
@fam varchar(20),
@country varchar(20))
as
if not exists (select Operator_Imya, Operator_Fam from Operators where Operator_Imya=@imya
and Operator_Fam=@fam)
begin
insert into Operators values
(@otch,@imya,@fam,@country)
end

```

9. Служит для добавления нового жанра.

```

create procedure New_Genre
(@genre_name varchar(20))
as
if not exists (select Genre_ID from Genres where Genre_NAME=@Genre_NAME)
begin
insert into Genres values
(@genre_name)
end

```

10. Служит для добавления нового режиссера.

```

create procedure New_Director
(@otch varchar(20),
@imya varchar(20),
@fam varchar(20),
@country varchar(20))
as
if not exists (select Director_Imya, Director_Fam from Directors where Director_Imya=@imya
and Director_Fam=@fam)
begin

```

```

        insert into Directors values
        (@otch,@imya,@fam,@country)
    end

```

### 11. Служит для добавления нового композитора.

```

create procedure New_Composer
(@otch varchar(20),
@imya varchar(20),
@fam varchar(20),
@country varchar(20))
as
if not exists (select Composer_Imya, Composer_Fam from Composers where
Composer_Imya=@imya and Composer_Fam=@fam)
    begin
        insert into Composers values
        (@otch,@imya,@fam,@country)
    end

```

### 12. Служит для добавления новой кинокомпании.

```

create procedure New_Company
(@name varchar(20),
@country varchar(20))
as
if not exists (select Company_NAME from Companies where Company_NAME=@name)
    begin
        insert into Companies values
        (@name,@country)
    end

```

### 13. Служит для добавления нового исполнителя.

```

create procedure New_Band
(@name varchar(20),
@country varchar(20),
@genre_name varchar(20))
as
declare @genre_ID int

if not exists (select Genre_ID from Genres where Genre_NAME=@Genre_NAME)
    begin
        insert into Genres values
        (@genre_name)
    end
set @genre_ID = (select Genre_ID from Genres where Genre_NAME=@Genre_NAME)

if not exists (select Band_NAME from Bands where Band_NAME=@name)
    begin
        insert into Bands values

```

```
(@name,@country,@genre_ID)
end
```

#### 14. Служит для добавления нового актера.

```
create procedure New_Actor
(@otch varchar(20),
@imya varchar(20),
@fam varchar(20),
@country varchar(20))
as
if not exists (select Actor_Imya, Actor_Fam from Actors where Actor_Imya=@imya and
Actor_Fam=@fam)
begin
insert into Actors values
(@otch,@imya,@fam,@country)
end
```

Для разрабатываемой базы данных также были созданы необходимые функции:

1. Служит для получения ID актера при вводе его имени и фамилии. Результат выполнения функции «GetActor» представлен на рисунке 2.17.

```
create function GetActor(@Actor_IMYA varchar(20),@Actor_FAM varchar(20))
returns int
as
begin
declare @Actor_ID int
select @Actor_ID = Actor_ID from Actors where Actor_IMYA=@Actor_IMYA and
Actor_FAM=@Actor_FAM
return @Actor_ID
end
```

```
select actor_imya, actor_fam, dbo.GetActor(actor_imya,actor_fam) as 'ID' from Actors
```

	actor_imya	actor_fam	ID
1	Джей Кей	Симмонс	1
2	Майлз	Теллер	2
3	Мэттью	Макконахи	3
4	Джессика	Честейн	4
5	Майкл	Кейн	5
6	Энн	Хэтзуэй	6
7	Жан	Рено	7
8	Натали	Портман	8
9	Гэри	Олдман	9
10	Дэни	Айелло	10
11	Том	Хэнкс	11

Рисунок 2.17 – Результат выполнения функции «GetActor»



2. Служит для получения ID исполнителя при вводе его названия. Результат выполнения функции «GetBand» представлен на рисунке 2.18.

```
create function GetBand(@Band_NAME varchar(20))
returns int
as
begin
    declare @Band_ID int
    select @Band_ID = Band_ID from Bands where Band_NAME=@Band_NAME
    return @Band_ID
end
```

```
select band_name, dbo.GetBand(band_name) as 'ID' from Bands
```

	band_name	ID
1	Gorillaz	1
2	Led Zeppelin	2
3	Pink Floyd	3
4	Radiohead	4

Рисунок 2.18 – Результат выполнения функции «GetBand»

Создание триггеров:

1. Служит для запрета ввода ID фильмов больше 199999. Результат выполнения триггера «prod\_trig» представлен на рисунке 2.19.

```
Create trigger prod_trig on Products for update,insert
as
declare @id int, @disctype int
select @id=I.Product_ID, @disctype=I.DiscType_ID from inserted I
if (@id not between 100000 and 199999 and @disctype between 6 and 8)
begin
rollback tran
raiserror('ID фильмов не должен быть больше 199999',16,10)
end
```

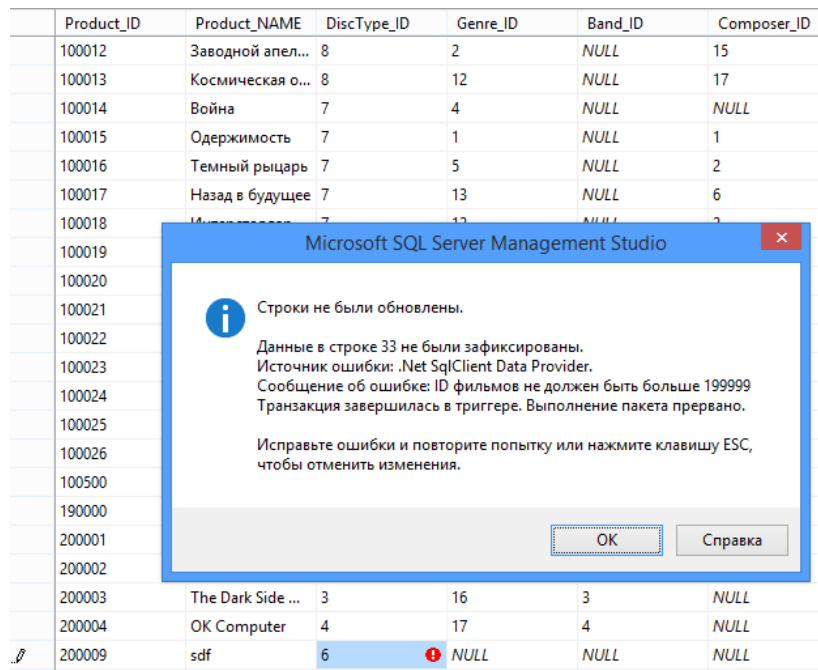


Рисунок 2.19 – Результат выполнения триггера «Prod\_trig»

2. Служит для запрета ввода ID музыки меньше 200000. Результат выполнения триггера «prod\_trig1» представлен на рисунке 2.20.

```

create trigger prod_trig1 on Products for update,insert
as
declare @id int, @disctype int
select @id=I.Product_ID,@disctype=I.DiscType_ID from inserted I
if (@id between 100000 and 199999 and @disctype between 1 and 5)
begin
rollback tran
raiserror('ID музыки не должен быть меньше 200000',16,10)
end

```

	Product_ID	Product_NAME	DiscType_ID	Genre_ID	Band_ID	Composer_ID	Prod
	100012	Заводной апел...	8	2	NULL	15	20
	100013	Космическая о...	8	12	NULL	17	20
	100014	Война	7	4	NULL	NULL	22
	100015	Одержимость	7	1	NULL	1	8
	100016	Темный рыцарь	7	5	NULL	2	2
	100017	Назад					
	100018	Интер					
	100019	Леон					
	100020	Спас					
	100021	Хоро					
	100022	Спис					
	100023	Форр					
	100024	Псих					
	100025	Казин					
	100026	Слав					
	100500	sdf					
	190000	sdf					
	200001	Demon Days	1	14	1	NULL	23
	200002	Led Zeppelin IV	2	15	2	NULL	24
	200003	The Dark Side ...	3	16	3	NULL	25
	200004	OK Computer	4	17	4	NULL	26
	185689	sdf	2	NULL	NULL	NULL	NULL

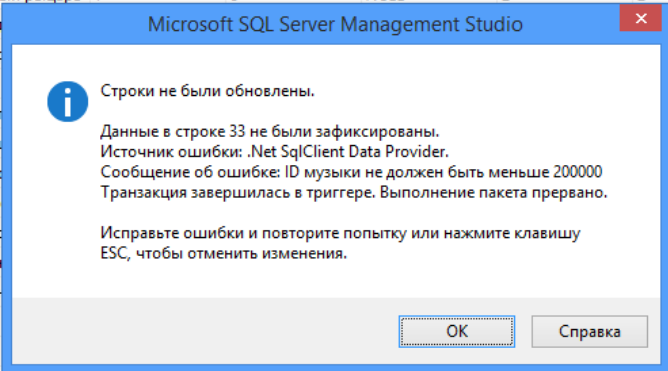


Рисунок 2.20 – Результат выполнения триггера «Prod\_trig1»

## 2.8 Описание интерфейса приложения и его функциональности

Интерфейс данного приложения очень удобен для использования. С приложением могут работать три вида пользователей: администратор, пользователь и гость. Администратор может редактировать все виды таблиц, пользователь – только некоторую ее часть, а гость может только просматривать таблицы фильмов и музыки.

После запуска приложения открывается первая форма – форма входа. На данном этапе производится авторизация пользователя (рисунок 2.21).

Рисунок 2.21 – Форма входа

При осуществлении входа под логином «Администратор (Administrator)» пароль представляет собой комбинацию – «А», открывается вторая форма (рисунок 2.22).

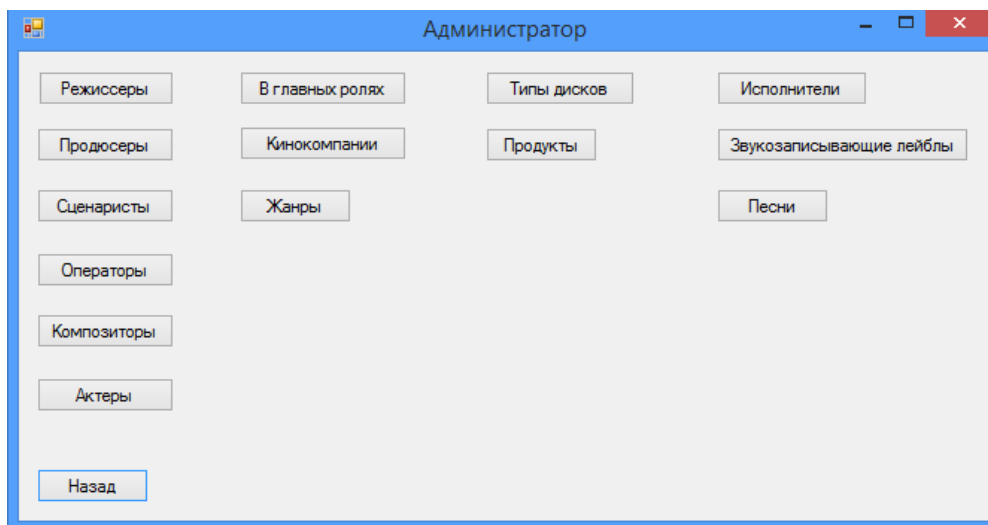


Рисунок 2.22 – Форма администратора

Администратору предоставляется возможность добавления новых продуктов, добавления всех видов справочной информации, такие как актеры, режиссеры, сценаристы и т.д. Также администратор может выводить любую информацию. Для того, чтобы добавить новый продукт, необходимо заполнить соответствующие поля и нажать на кнопку "Сохранить" в верхней панели управления (рисунок 2.23).

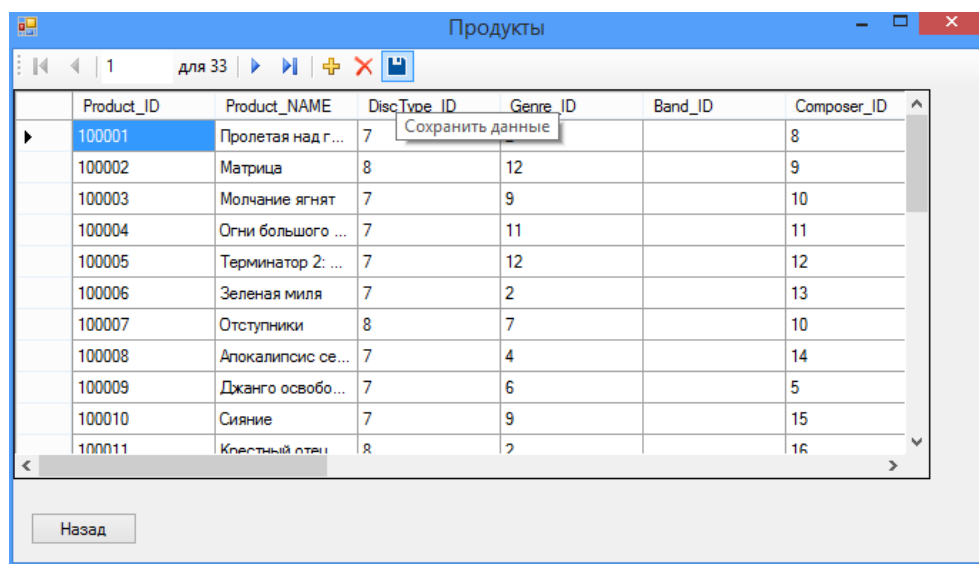
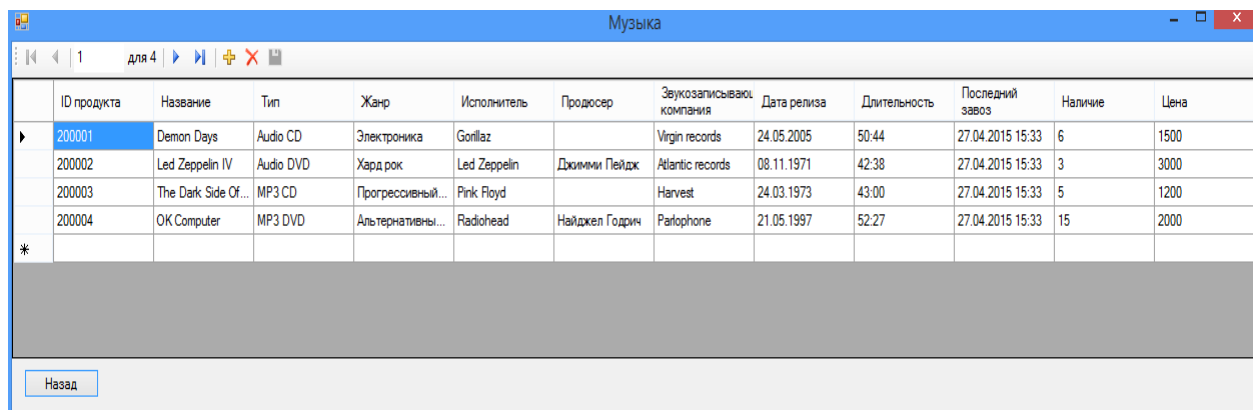


Рисунок 2.23 – Пример добавления нового продукта

При нажатии на кнопку «Музыка» под логином «Гость» выводится таблица без права редактирования, содержащая музыкальные продукты (рисунок 2.24).



ID продукта	Название	Тип	Жанр	Исполнитель	Продюсер	Звукозаписывающая компания	Дата релиза	Длительность	Последний завоз	Наличие	Цена
200001	Demon Days	Audio CD	Электроника	Gorillaz		Virgin records	24.05.2005	50:44	27.04.2015 15:33	6	1500
200002	Led Zeppelin IV	Audio DVD	Хард рок	Led Zeppelin	Джимми Пейдж	Atlantic records	08.11.1971	42:38	27.04.2015 15:33	3	3000
200003	The Dark Side Of...	MP3 CD	Прогрессивный...	Pink Floyd		Harvest	24.03.1973	43:00	27.04.2015 15:33	5	1200
200004	OK Computer	MP3 DVD	Альтернативны...	Radiohead	Найджел Годрич	Parlophone	21.05.1997	52:27	27.04.2015 15:33	15	2000
*											

Назад

Рисунок 2.24 – Пример вывода информации о музыке

### **3 Технико-экономическая часть**

#### **3.1 Описание работы и обоснование необходимости**

Целью дипломной работы является разработка информационной системы для учета мультимедийной продукции. Профилем торговой компании «Меломан» является широкий ассортимент фильмов и музыки, выпускаемых на CD, DVD и Blu-Ray дисках.

На сегодняшний день оборот мультимедийных продуктов растет сильными темпами. Поэтому, каждому предприятию в сфере продаж должна прилагаться надежная база данных с пользовательским интерфейсом для эффективного учета и реализации своей продукции.

В данном разделе приводится описание экономической составляющей для создания данного проекта, отражающей временные, трудовые и финансовые затраты на проект.

#### **3.2 Трудовые ресурсы, используемые в работе**

В данной работе используется интеллектуальный труд, стоимость затрат которого выше, чем физического труда.

В разработке приняли участие:

- Руководитель – постановка задачи, разработка управленческих решений, финансовых составляющих проекта.
- Программист – разработка алгоритмов и программирование;
- Торговый менеджер – консультация в создании информационной системы.

Количество сотрудников, задействованных в разработке, представлено в таблице 3.1.

Таблица 3.1 – Сотрудники и их заработная плата

Должность	Количество человек	Заработная плата в месяц, тенге
Руководитель	1	120000
Программист	1	100000
Торговый менеджер	1	60000
Итого	3	280000

#### **3.3 Оборудование, используемое в работе**

Оборудование, используемое при создании информационной системы, представлено в таблице 3.2.

Таблица 3.2 – Перечень оборудования, необходимого для разработки

Наименование изделий	Характеристика	Количество единиц	Цена за единицу, тенге	Общая сумма, тенге
Ноутбук	HP Pavilion 15; Core i5-4210U (1,7 GHz); 1 TB HDD; 8192 Mb DDR3 SDRAM; Windows 8.1	1	130000	130000
Многофункциональное устройство	HP LaserJet M1132 MFP	1	30000	30000
Итого				160000

Цены на оборудование представлены без учета НДС.

### 3.4 Программное обеспечение, используемое в работе

При разработке информационной системы было использовано следующее программное обеспечение:

- Windows 8.1 – операционная система;
- Kaspersky Internet Security – антивирусная система;
- MS Visual Studio 2012 Professional – среда разработки ПО.

Программное обеспечение, использованное при разработке информационной системы, представлено в таблице 3.3.

Таблица 3.3 – перечень программного обеспечения, необходимого для разработки информационной системы

Программное обеспечение	Стоимость, тенге
Windows 8.1	Бесплатно, входит в предустановленный пакет при покупке ноутбука
Kaspersky Internet Security	11200
MS Visual Studio 2012 Professional	72500
Итого	83700

Цены на ПО приведены без учета НДС.

### 3.5 Сроки реализации проекта

Процесс разработки и сроки реализации информационной системы состоит из 10 этапов и включает в себя:

- Разработка алгоритма;
- проектирование и создание базы данных;
- заполнение базы данных;
- разработка и создание пользовательского интерфейса системы;
- тестирование программного обеспечения;
- оформление отчета.

Этапы и сроки реализации проекта отображены в таблице 3.4.

Таблица 3.4 - Этапы и сроки реализации проекта

Этап работы	Наименование работы	Недели от начала работ												
		1	2	3	4	5	6	7	8	9	10			
1	Постановка задачи	■												
2	Подбор и изучение литературы	■	■											
3	Разработка алгоритма		■	■										
4	Проектирование и создание базы данных		■	■										
5	Заполнение базы данных			■	■									
6	Разработка и создание пользовательского интерфейса системы				■	■								
7	Тестирование ПО							■						
8	Отладка ПО								■					
9	Оформление отчета									■	■			
10	Проверка и сдача отчета											■	■	

### 3.6 Расчет стоимости работы по разработке

Расчет стоимости работы по разработке - это наиболее важная часть экономического анализа, так как на основе этого расчета определяются затраты рабочего времени на разработку проекта на каждом этапе, а также трудовые затраты [10].



Затраты на разработку данного проекта определяются по формуле

$$C = \text{ФОТ} + O_c + A + \text{Э} + C_{\text{пр}} + H \quad (3.1)$$

где ФОТ - фонд оплаты труда;  
 $O_c$  - социальный налог;  
A - амортизационные отчисления;  
Э - затраты на электроэнергию;  
 $C_{\text{пр}}$  - прочие расходы;  
H - накладные расходы.

### 3.6.1 Расчет затрат на оплату труда

Затраты на оплату труда персонала, задействованного в разработке проекта, рассчитываются по формуле:

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}} \quad (3.2)$$

где  $Z_{\text{осн}}$  - основная заработная плата;  
 $Z_{\text{доп}}$  - дополнительная заработная плата.

Труд руководителя проекта, программиста и торгового менеджера принят условно, на договорной основе в размере 120000, 100000 и 60000 тенге соответственно.

На этапах разработки информационной системы участники разработки задействованы неравноценно, для этого необходимо рассчитать средний дневной заработок, а затем и общий размер заработной платы, в зависимости от их фактического участия.

Средний дневной заработок каждого работника рассчитывается по формуле:

$$D = \frac{Z_{\text{Пм}}}{D_{\text{р}}} \quad (3.3)$$

где  $Z_{\text{Пм}}$  – ежемесячный размер заработной платы;  
 $D_{\text{р}}$  – количество рабочих дней в месяце (это 26 дней – шестидневная рабочая неделя).

1) Для руководителя проекта

$$D = \frac{120000}{26} = 4615,4 \text{ тенге/день}$$

2) Для программиста

$$D = \frac{100000}{26} = 3846,2 \text{ тенге/день}$$

3) Для торгового менеджера

$$D = \frac{60000}{26} = 2307,7 \text{ тенге/день}$$

Заработная плата за один час вычисляется по формуле:

$$H = \frac{ЗП_M}{D_p * Ч_p} \quad (3.4)$$

где  $ЗП_M$  – ежемесячный размер заработной платы;

$D_p$  – количество рабочих дней в месяце.

$Ч_p$  – Количество часов рабочего дня ( $Ч_p = 8$ )

1) Для руководителя проекта

$$H = \frac{120000}{26 * 8} = 576,9 \text{ тенге/час}$$

2) Для программиста

$$H = \frac{100000}{26 * 8} = 480,8 \text{ тенге/час}$$

3) Для торгового менеджера

$$H = \frac{60000}{26 * 8} = 288,5 \text{ тенге/час}$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n * z * K} \quad (3.5)$$

где  $T$  – трудоемкость этапа, норма-час;

$q_n$  – количество исполнителей по этапу;  
 $z$  – продолжительность рабочего дня,  $z = 8$  часов;  
 $K$  – коэффициент выполнения норм времени,  $K = 1,1$ .

Полученную величину  $t_n$  округляем в большую сторону до целых дней.

$$t_1 = \frac{24}{1 * 8 * 1,1} \approx 3 - \text{руководитель проекта, постановка задачи};$$

Для каждого вида работ произведены аналогичные расчеты.

$$t_n = 3 + 5 + 10 + 14 + 7 + 10 + 2 + 3 + 3 + 3 = 60 \text{ дней}$$

Таким образом, для проведения всех работ необходимо 60 дней.

Сводные данные по расчету заработной платы персонала, задействованного в разработке проекта, приведены в таблице 3.5.

Таблица 3.5 – Сводные данные по расчету основной заработной платы персонала задействованного в разработке проекта.

Наименование этапов	Исполнитель	Трудовая норма, часов	Длительность цикла, дни	Заработная плата за час работы, тенге	Сумма заработной платы, тенге
Постановка задачи	Руководитель проекта	24	3	576,9	13845,6
Подбор и изучение литературы	Программист	40	5	480,8	19232
Разработка алгоритма	Программист	80	10	480,8	38464
Проектирование и создание базы данных	Программист	112	14	480,8	53849,6
Заполнение базы данных	Торговый менеджер	56	7	288,5	16156
Разработка и создание пользовательского интерфейса	Программист	80	10	480,8	38464

Тестирование ПО	Программист	16	2	480,8	7692,8
Отладка ПО	Программист	24	3	480,8	11539,2
Оформление отчета	Программист	24	3	480,8	11539,2
Проверка и сдача отчета	Руководитель проекта	24	3	576,9	13845,6
Итого		480	60	4807,9	224628

Дополнительная заработная плата составляет 10% от основной заработной платы и вычисляется по формуле:

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1 \quad (3.6)$$

и составит

$$Z_{\text{доп}} = 224628 * 0,1 = 22462,8 \text{ тенге};$$

Таким образом, затраты на оплату труда согласно произведенным расчетам и в соответствии с формулой 3.2 составит

$$\text{ФОТ} = 224628 + 22462,8 = 247090,8 \text{ тенге}$$

### 3.6.2 Расчет затрат по социальному налогу

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$O_c = (\text{ФОТ} - \text{ПО}) * 11\% \quad (3.7)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются, вычисляются отчисления по формуле

$$\text{ПО} = \text{ФОТ} * 10\% \quad (3.8)$$

$$\text{ПО} = 247090,8 * 0,1 = 24709,08 \text{ тенге.}$$

Таким образом, в соответствии с произведенными расчетами и согласно формуле 3.7 размер отчислений на социальные нужды составит

$$O_c = (241166,64 - 24116,664) * 0,11 = 24462 \text{ тенге}$$

### 3.6.3 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле

$$A_i = \frac{H_A * C_{ПЕР} * N}{100 * n} \quad (3.9)$$

где  $H_A$  – норма амортизации;

$C_{ПЕР}$  – первоначальная стоимость оборудования;

$N$  – количество дней на выполнение работ;

$n$  – количество рабочих дней в году.

Норма амортизации на компьютерную технику составляет 40% от стоимости всего оборудования.

Таким образом, амортизационные отчисления по используемому оборудованию, в соответствии с формулой 3.9 составят

- на ноутбук

$$A_1 = \frac{40 * 130000 * 60}{100 * 365} = 8547,95 \text{ тенге}$$

- на многофункциональное устройство

$$A_2 = \frac{40 * 30000 * 7}{100 * 365} = 230,14 \text{ тенге}$$

- на программное обеспечение

$$A_3 = \frac{15 * 83700 * 60}{100 * 365} = 2063,84 \text{ тенге}$$

$$A_i = 8547,95 + 230,14 + 2063,84 = 10841,93 \text{ тенге}$$

Сводные результаты расчета амортизационных отчислений представлены в таблице 3.6.

Таблица 3.6 – Сводные данные по расчету затрат на амортизацию

Наименование оборудования	Количество	Норма амортизации, %	Сумма амортизации, тенге
Ноутбук	1	40	8547,92
Многофункциональное устройство	1	40	230,14
Программное обеспечение	-	15	2063,84
Итого			10841,93

### 3.6.4 Расчет затрат на электроэнергию

Поскольку в процессе производства используется электрооборудование, необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию для производственных нужд включают в себя расходы электроэнергии на оборудование и дополнительные нужды. И рассчитываются по формуле

$$\mathcal{E} = \mathcal{E}_{\text{эл.эн.об.}} + \mathcal{E}_{\text{доп.}} \quad (3.10)$$

где  $\mathcal{E}_{\text{эл.эн.об.}}$  – затраты на электроэнергию для оборудования;

$\mathcal{E}_{\text{доп.}}$  – затраты на электроэнергию для дополнительных нужд.

Расходы электроэнергии для оборудования рассчитываются по формуле

$$\mathcal{E}_{\text{эл.эн.об.}} = W * T * S * K_{\text{исп}} \quad (3.11)$$

где  $W$  – потребляемая мощность, Вт;

$T$  – время работы, часы;

$S$  – тариф (1кВт = 16,02 тенге);

$K_{\text{исп}}$  – коэффициент использования ( $K_{\text{исп}} = 0,9$ ).

$$\mathcal{E}_{\text{эл.эн.об.}}(\text{ноутбук}) = 0,7 * 480 * 16,02 * 0,9 = 4844,45 \text{ тенге}$$

$$\mathcal{E}_{\text{эл.эн.об.}}(\text{МФУ}) = 0,9 * 56 * 16,02 * 0,9 = 726,67 \text{ тенге}$$

Общая сумма затрат на электроэнергию основного оборудования согласно формуле 3.11 составляет

$$Z_{\text{эл.эн.об.}} = 4844,45 + 726,67 = 5571,11 \text{ тенге}$$

Затраты на дополнительные нужды берутся по показателю от затрат на оборудование в размере 5% и рассчитывается по формуле

$$Z_{\text{доп}} = Z_{\text{эл.эн.об.}} * 5\% \quad (3.12)$$

и составляют

$$Z_{\text{доп}} = 5571,11 * 0,05 = 278,56 \text{ тенге}$$

Таким образом, суммарные затраты на электроэнергию, согласно формуле 3.11, составляют

$$Э = 5571,11 + 278,56 = 5849,67 \text{ тенге}$$

Сводные результаты расчета затрат на электроэнергию представлены в таблице 3.7.

Таблица 3.7 – Сводные данные о затратах на электроэнергию

Наименование приборов	Количество	Потребляемая мощность, Вт	Число рабочих дней	Коэффициент использования	Время работы оборудования, часы	Сумма затрат, тенге
Ноутбук	1	0,7	60	0,9	480	4844,45
МФУ	1	0,9	7	0,9	56	726,67
$Z_{\text{доп}}$	-	-	-	-	-	278,56
Итого						5849,67

### 3.6.5 Расчет накладных и прочих расходов

Прочие расходы включают в себя:

1) расходы на интернет на 60 дней (2месяца). Стоимость интернета за один месяц составляет 4700 тенге. Получаем:

$$4700 * 2 = 9400 \text{ тенге}$$

2) расходы на канцелярские товары:

- упаковка бумаги формата А4 – 750 тенге;
- упаковка ручек – 300 тенге;
- 5 карандашей – 250 тенге;

- ластик – 100 тенге;

Общая стоимость канцелярских товаров:

$$750 + 300 + 250 + 100 = 1400 \text{ тенге}$$

Прочие расходы составляют

$$C_{\text{ПР}} = 9400 + 1400 = 10800 \text{ тенге}$$

Накладные расходы составляют 50% от всех затрат и рассчитываются по формуле

$$H = (\text{ФОТ} + O_c + A + \text{Э} + C_{\text{ПР}}) * 50\% \quad (3.13)$$

$$H = (247090,8 + 24462 + 10841,93 + 5849,67 + 10800) * 0,5 = \\ = 149522,2 \text{ тенге}$$

### 3.6.6 Расчет стоимости по всем статьям затрат и определение структуры затрат

В соответствии с формулой 3.1 суммарные затраты по разработке моего проекта составляют

$$C = 247090,8 + 24462 + 10841,93 + 5849,67 + 10800 + 187611,7 = \\ = 448566,6 \text{ тенге}$$

Смета затрат по разработке информационной системы, а также структура расходов представлены в таблице 3.7 и на рисунке 3.1

Таблица 3.8 – Суммарные данные по стоимости разработки проекта

Наименование статьи затрат	Сумма, тенге
Фонд оплаты труда	247090,8
Социальный налог	24462
Амортизация	10841,93
Затраты на электроэнергию	5849,67
Прочие расходы	10800
Накладные расходы	149522,2
Итого	448566,6



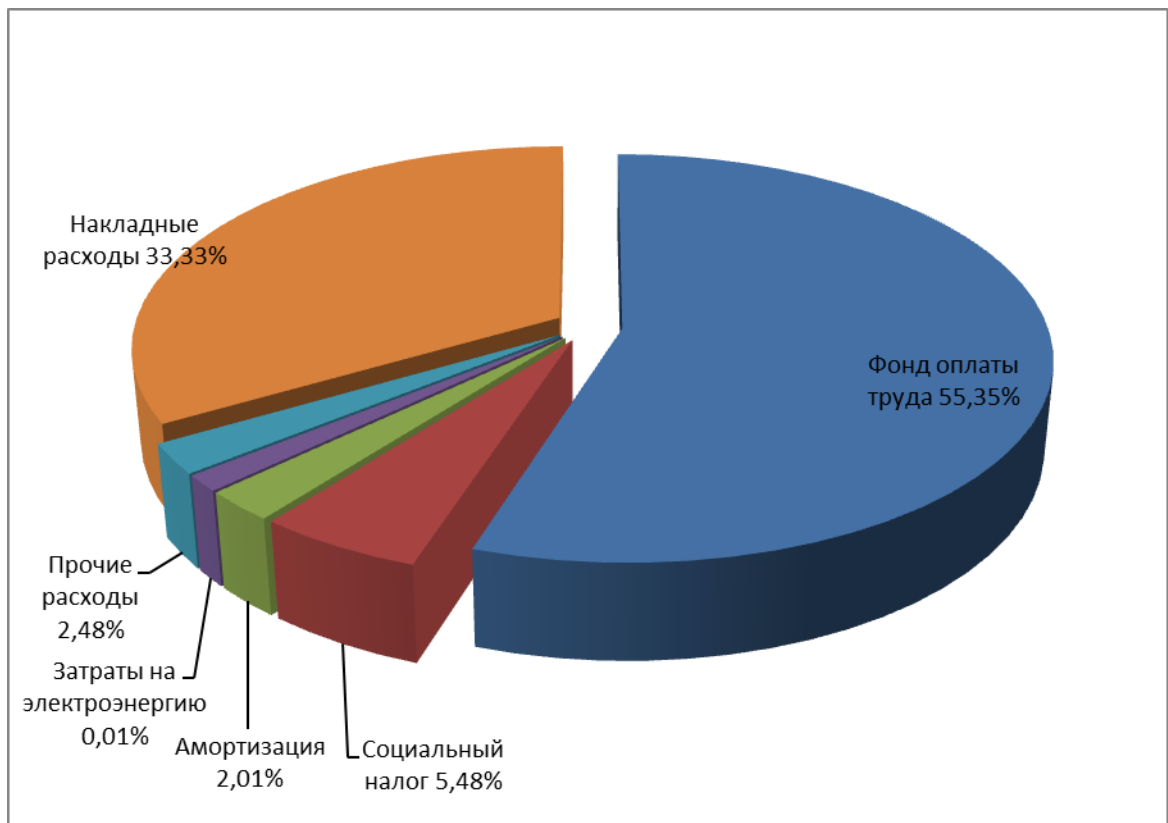


Рисунок 3.1 – Структура затрат по разработке информационной системы

### 3.7 Цена интеллектуального труда

Цена реализации проекта складывается из стоимости и чистого дохода

$$Ц = С + П \quad (3.14)$$

где С – стоимость продукта;  
 П – чистый доход.

При определении первоначальной цены следует задаться желаемым уровнем рентабельности (здесь 20%) реализации программных продуктов

$$Ц_{П} = С * \left(1 + \frac{P}{100}\right) \quad (3.15)$$

где P – рентабельность.

$$Ц_{П} = 448566,6 * \left(1 + \frac{20}{100}\right) = 538279,92 \text{ тенге}$$

Цена реализации проекта рассчитывается по формуле

$$C_p = C_{\Pi} + \text{НДС} \quad (3.16)$$

где НДС – налог на добавленную стоимость по ставке 12%

$$\text{НДС} = C_{\Pi} * 12\% \quad (3.17)$$

$$\text{НДС} = 538279,92 * 0,12 = 64593,6 \text{ тенге}$$

В соответствии с формулой (3.16) цена реализации проекта составит

$$C_p = 538279,92 + 64593,6 = 602873,5 \text{ тенге}$$

В зависимости от спроса, окончательная цена на интеллектуальный труд в условиях рыночных отношений будет варьироваться и изменяться.

Таким образом, разработка информационной системы для компании, занимающейся продажей мультимедийной продукции, является сложным и трудоемким процессом, требующих больших затрат интеллектуального труда. Стоимость разработки включает в себя следующие категории затрат: фонд оплаты труда, социальные отчисления, амортизационные отчисления, затраты на электроэнергию.

Большую часть в общей себестоимости разработки программного продукта составляют затраты на оплату труда в размере 247090,8 тенге, что составляет 55,35% от суммы разработки всего проекта.

## **4 Безопасность жизнедеятельности**

### **4.1 Анализ потенциально опасных и вредных производственных факторов, воздействующих на персонал**

Создание дипломного проекта «Разработка информационной системы для торговой компании Меломан» проводится в помещении без повышенного уровня опасности. Помещение сухое, температура поддерживается на уровне 20 °С – 25 °С с влажностью, не выходящей за пределы нормы в 60%. Все условия работы в помещении соответствуют ГОСТу: в нем отсутствует сырость и токопроводящая пыль.

Дипломный проект представляет собой программу для торговой компании. Следовательно, для работы с программой требуется персональный компьютер (ПК).

Современные ПК являются эргонасыщенными аппаратами с потреблением до 200 - 250 Вт, содержащими несколько электро- и радиоэлектронных устройств. ПК создает вокруг себя поля с широким частотным спектром и пространственным распределением, такие как:

1. электростатическое поле;
2. переменные низкочастотные электрические поля;
3. переменные низкочастотные магнитные поля.

Многие пользователи ПК связаны с воздействием таких психофизических факторов, как умственное перенапряжение, перенапряжение зрительных и слуховых анализаторов, монотонность труда, эмоциональные перегрузки. Воздействие указанных неблагоприятных факторов приводит к снижению работоспособности, вызванное развивающимся утомлением. Появление и развитие утомления связано с изменениями, возникающими во время работы в центральной нервной системе, с тормозными процессами в коре головного мозга. Сильный шум вызывает трудности с распознаванием цветовых сигналов, снижает быстроту восприятия цвета, остроту зрения, зрительную адаптацию, нарушает восприятие визуальной информации, уменьшает на 5 - 12 % производительность труда. Длительное воздействие шума с уровнем звукового давления 90 дБ снижает производительность труда на 30 - 60 % .

Безопасное использование персонального компьютера может быть осуществлено при следующих разновидностях освещения:

– Общим люминесцентном освещении, когда видеомониторы располагаются по периметру помещения или при центральном расположении рабочих мест в два ряда по длине класса с экранами, обращенные в противоположные стороны.

– Совмещенном освещении (естественное + искусственное) только при одном и трех рядном расположении рабочих мест, когда экран и поверхность рабочего стола находятся перпендикулярно светонесущей стене.

– Естественное освещение, когда рабочие места с ПК располагаются в один ряд по длине помещения на расстоянии 0,8 – 1,0м от стены с оконными проемами, и экраны находятся перпендикулярно этой стены.

– Основной поток естественного света при этом должен быть слева. Не допускается направление основного светового потока естественного света справа, сзади и спереди работающего на ПЭВМ.

– Безопасное расстояние глаз до монитора компьютера должно быть 60–70 см, допустимое не менее 50 см.

Так как работа программиста прямо связана с электрооборудованием, необходимо уметь пользоваться им правильно и соблюдать меры безопасности от поражения электрического тока.

Существует множество мер от повреждения электрическим током. Одно из них защитное заземление. Защитное заземление – преднамеренно электрическое заземление с землей или ее эквивалентом металлических нетоковедущих частей, которые могут оказаться под напряжением вследствие замыкания на корпус и по другим причинам.

Также есть и такая мера защиты от поражения электрического тока, называемая занулением. Опасность поражения током при прикосновения к корпусу и другим нетоковедущим металлическим частям электрооборудования, оказавшимся под напряжением вследствие замыкания на корпусе и по др. причинам, может быть устранена быстрым отключением поврежденной электроустановки от питающей сети и вместе с тем снижением напряжения корпуса относительно земли. Этой цели и служит зануление. Зануление – преднамеренное электрическое соединение с нулевым защитным проводником металлических нетоковедущих частей, которые могут оказаться под напряжением.

Принцип действия зануления – превращение замыкание с целью вызвать большой ток, способный обеспечить срабатывание защиты и тем самым автоматически отключить поврежденную электроустановку от питающей сети.

Также есть различные электрические защитные средства от поражения током. Защитные средства могут быть условно разделены на три группы: изолирующие, ограждающие и предохранительные.

Изолирующие – изолируют человека от токоведущих или заземленных частей, а также от земли. Они делятся на основные и дополнительные.

Без строгого учёта правил техники безопасности и производственной санитарии, неточного выполнения требований техники безопасности может привести к аварии, либо к профессиональным заболеваниям и производственному травматизму. Охрана труда обеспечивается системой законодательных актов, социально-экономических, организационных, технических, гигиенических и лечебно-профилактических мероприятий и средств, направленных на создание таких условий труда, при которых исключено воздействие на работающих опасных и вредных производственных факторов. Создание наиболее благоприятных, комфортных условий труда,

улучшение охраны труда и техники безопасности, без сомнения, ведет к более высокой производительности труда, социальному развитию и повышению благосостояния.

При планировке рабочего места необходимо учитывать удобство расположения дисплеев, принтеров, пульта ПК, а также зоны досягаемости рук оператора. Эти зоны, установленные на основании антропометрических данных тела человека, дают возможность рационально разместить компьютер, его клавиатуру и дисплей.

Высота рабочей поверхности стола должна регулироваться в пределах 680–800 мм; при отсутствии такой возможности должна составлять 725 мм.

Дисплей должен удовлетворять следующим требованиям:

1) важнейшие элементы конструкции должны быть расположены в центре поля зрения (клавиатура);

2) элементы должны быть сгруппированы по функциональному признаку;

3) рабочие поверхности должны быть расположены наклонно, по возможности перпендикулярно взгляду оператора;

4) экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600–700 мм, но не ближе 500 мм с учётом размеров знаков и символов.

В помещении на пользователя могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея.

## **4.2 Анализ помещения**

- Рабочее помещение находится на 3 этаже.
- Тип аудитории: кабинет.
- Помещение представляет собой комнату, размеры которой 4x5x3 метров (ширина-длина).
- Источники света: 2 люминесцентные лампы.
- Одно окно (1x2).
- Окраска кабинета и мебели по стандарту должна способствовать созданию благоприятных условий для зрительного восприятия.

– Для защиты от избыточной яркости с окон могут быть применены жалюзи.

Схема помещения приведена на рисунке 4.1.

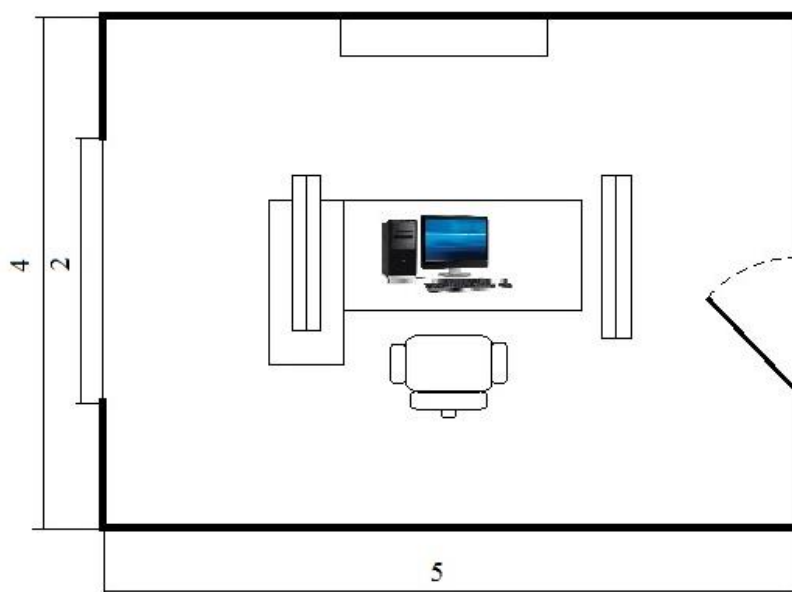


Рисунок 4.1 – План рабочего места

В таблице 4.1 приведены оптимальные нормы параметров микроклимата с учетом периода года. ССБТ для легкой физической работы. Оборудование, установленное в рабочем помещении, не является источником выделения тепла (очень незначительное выделение тепла аппаратурой никаким образом не оказывает влияние на микроклимат рабочего помещения) [11].

Таблица 4.1 – Оптимальные нормы микроклимата для помещений с ПК

Период года	Категория работ	Температура воздуха °С не более	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая – 1а	22-24	40-60	0,1
Теплый	Легкая – 1а	23-25	40-60	0,1

Здание относится к I степени огнестойкости. Рабочее помещение по вопросам пожарной безопасности относится к классу “Д”. В соответствии с типовыми правилами пожарной безопасности административные здания и отдельные помещения, и технологические установки обеспечиваются первичными средствами пожаротушения согласно нормативам.

### 4.3 Расчет системы кондиционирования

Кондиционирование обеспечивает улучшение микроклимата в помещении и должно выполняться в соответствии с нормами. Норма пространства помещения на одного человека составляет 6 м<sup>2</sup>. В офисе предполагается наличие воздуховодов, которые размещены за подвесными потолками. Воздух плавно подается и удаляется из офиса либо через вентиляционные решетки в стенах, либо через специальные розетки, установленные на потолке. Также установлен кондиционер, который освежает воздух.

$V_{\text{вент}}$  – объем воздуха, необходимый для обмена;

$V_{\text{пом}}$  – объем рабочего помещения.

Для расчета примем следующие размеры рабочего помещения (рисунок 4.1):

Длина  $a = 4$  м;

Ширина  $b = 5$  м;

Высота  $h = 3$  м;

Окно = 2x1 м.

Соответственно, объем помещения равен

$$V_{\text{пом}} = a \cdot b \cdot h = 60 \text{ м}^3 \quad (4.1)$$

Необходимый для обмена объем воздуха  $V_{\text{вент}}$  определим исходя из уравнения теплового баланса.

$$V_{\text{вент}} \cdot C (t_{\text{уход}} - t_{\text{приход}}) \cdot Y = Q \cdot 3600 \quad (4.2)$$

где  $V_{\text{вент}}$  - объем воздуха, необходимый для обмена;

$Q_{\text{избыт}}$  - избыточная теплота (Вт);

$C$  - удельная теплопроводность воздуха,  $C=1000$  (Дж/кгК);

$Y$  - плотность воздуха,  $Y=1,2$  мг/см<sup>3</sup>.

$t_{\text{приход}}$  - 18 градусов;

Температура уходящего воздуха определяется по формуле

$$t_{\text{уход}} = t_{\text{р.м.}} + (h - 2)t \quad (4.3)$$

где  $t = 1-5$  градусов – превышение  $t$  на 1м высоты помещения;

$t_{\text{р.м.}} = 26$  градусов – температура на рабочем месте;

$h = 3$  м – высота помещения;

$$t_{\text{уход}} = 26 + (3 - 2) 2 = 28 \text{ }^\circ\text{C}$$

$$Q_{\text{избыт}} = Q_{\text{изб.1}} + Q_{\text{изб.2}} + Q_{\text{изб.3}} \quad (4.4)$$

где  $Q_{\text{изб.}}$  – избыток тепла от электрооборудования и освещения

$$Q_{\text{изб.1}} = E \cdot P \quad (4.5)$$

где  $E$  – коэффициент потерь электроэнергии на теплоотвод  $E=0,55$ ;

$P$  – мощность

$$P = 60 \text{ Вт} \cdot 5 = 300 \text{ Вт},$$

$$Q_{\text{изб.1}} = 0.55 \cdot 300 = 165 \text{ Вт}$$

$Q_{\text{изб.2}}$  – теплоступление от солнечной радиации,

$$Q_{\text{изб.2}} = m \cdot S \cdot k \cdot Q_c \quad (4.6)$$

где  $m$  – число окон,  $m = 1$ ;

$S$  – площадь окна,  $S = 2 \cdot 1 = 2 \text{ м}^2$

$K$  – коэффициент, учитывающий остекление.

Для двойного остекления  $k = 0,6$ ;

$Q_c = 200 \text{ Вт/м}^2$  – теплоступление от окон.

$$Q_{\text{изб.2}} = 2 \text{ м}^2 \cdot 1 \cdot 0.6 \cdot 200 \text{ Вт/м}^2 = 240 \text{ Вт}$$

где  $Q_{\text{изб.3}}$  – тепловыделения людей

$$Q_{\text{изб.3}} = n \cdot q \quad (4.7)$$

где  $q = 80 \text{ Вт/чел}$ ,

$n$  – число людей,  $n = 1$

$$Q_{\text{изб.3}} = 1 \cdot 80 = 80 \text{ Вт}$$

$$Q_{\text{избыт}} = 165 + 240 + 80 = 485 \text{ Вт}$$

Из уравнения теплового баланса следует

$$V_{\text{вент}} = 3600 \cdot 485 / (1000 \cdot 1,2 \cdot (28-18)) = 145,5 \text{ м}^3 = 146 \text{ м}^3$$

Необходимо тщательно продумать месторасположение кондиционера в офисе. Можно установить канальный кондиционер за подвесным потолком и развести воздух в разные точки комнаты через воздуховоды. Это обеспечит равномерное распределение воздуха и температуры. При этом за счет объединения с системой вентиляции снижается стоимость затрат на



оборудование. Блоки могут располагаться на расстоянии до 20 м друг от друга.

В нашем случае высота потолков не позволяет установить канальный кондиционер, поэтому можно выбрать установку сплит системы. Конструкция кондиционера сплит-системы предполагает наличие двух агрегатов - наружного и внутреннего блока. Как следует из названия, внутренний блок должен быть установлен внутри помещения, в то время как наружный блок устанавливается снаружи здания. Сплит-системы отличает низкий уровень шума, небольшие размеры и разнообразие функций.

#### **4.4 Расчет системы автоматического пожаротушения**

Автоматическое пожаротушение подразумевает под собой наличие средств предупреждения пожара, устройств обнаружения пожара, сигнализации и средств пожаротушения. Совокупность перечисленных средств, установленных специалистами–пожарниками, позволяет считать данный комплекс не иначе как системой автоматического пожаротушения. При этом роль человека сводится к проведению перманентных диагностических работ и замене средств пожаротушения по истечению сроков годности. Разрабатываемая система автоматического пожаротушения является аппаратной. В данном зале располагается электронное оборудование, кросс и место оператора. Опасность возгорания истекает только от попадания постороннего напряжения на платы, или же прохождение токов высоких номиналов по проводникам, что вызовет возгорание изоляции. Учитывая тот момент, что оборудование весьма дорогостоящее и критичное к прямому попаданию воды и соли, то автоматическая система пожаротушения должна использовать соответствующие огнетушащие вещества и иметь высокий уровень надежности [12].

Применение системы автоматического пожаротушения (САП) основано на использовании установки газового пожаротушения, что позволит снизить материальные потери. САП газового пожаротушения использует следующие вещества:

- двуокись углерода;
- хладон 114В(2) / тетрофтордибромэтан;
- хладон 13В(1) / бромтрифторметан;
- комбинированный углекислотно–хладоновый состав;
- азот;
- аргон.

САП должна отвечать следующим требованиям:

- дистанционное и местное включение;
- выполнять функции пожарной сигнализации;
- соответствовать требованиям помещения, где установлена САП.

Тип САП и огнетушащие вещества выбираются с учётом пожарной опасности и физико–химических свойств, а также в зависимости от принадлежности помещения по СНиП (Строительные нормы и правила).

В данном случае необходимо выбрать соответствующую САП, пригодную для применения в условиях наличия установок, находящихся под напряжением и в закрытом помещении.

Наиболее часто используемые газовые САП в нашем случае, это:

- установки объёмного пожаротушения;
- установки локального пожаротушения по площади;
- установки локального пожаротушения по объёму.

Расчетная масса комбинированного углекислотно–хладонового состава  $m_d$  кг, для объёмного пожаротушения определяется по формуле

$$m_d = k_6 \cdot g_n \cdot V \quad (4.8)$$

где  $k_6$  – коэффициент компенсации не учитываемых потерь состава,

$k_6=1,13$ ;

$g_n$  – нормативная массовая концентрация состава –  $0,27$  кг/м,

$V$  – объём защищаемого помещения ( $m^3$ ).

$$m_d = 1,13 \cdot 0,27 \cdot 60 = 18,3 \text{ кг.}$$

Расчетное число баллонов  $E_2$  определяется из расчета вместимости в 40–литровый баллон 25 кг углекислотно–хладонового состава. Количество баллонов – 1 (25кг). Внутренний диаметр магистрального трубопровода  $d_i$ , мм, определяется по формуле

$$d_i = d_1 \cdot E_2 \quad (4.9)$$

где  $d_i$  – диаметр сифонной трубки баллона, мм;

$E_2$ – число одновременно разрезаемых баллонов.

$$d_i = 10 \cdot 2 = 14,2 \text{ мм.}$$

Эквивалентная длина магистрального трубопровода  $l_2$  м, определяется по формуле

$$l_2 = k_7 \cdot l \quad (4.10)$$

где  $k_7$  – коэффициент увеличения длины трубопровода для компенсации

$l$  – длина трубопровода по проекту,  $l=60$  м.

$$l_2 = 1,2 \cdot 60 = 72 \text{ м}$$

Площадь сечения выходного отверстия оросителя определим по формуле, м<sup>3</sup>:

$$A_3 = S / E_1 \quad (4.11)$$

где  $S$  – площадь сечения магистрального трубопровода, мм<sup>2</sup>;

$E_1$  – число оросителей;

$$A_3 = 122,5 / 6 = 20,4 \text{ мм}^2$$

Расход углекислотно–хладонового состава  $Q$  кг/с, зависит от эквивалентной длины и диаметра трубопровода,  $Q=0,8$ .

Расчетное время подачи углекислотно–хладонового состава  $t$ , мин, определится по формуле

$$t = md / 60Q, \quad (4.12)$$

$$t = 18,3 / 60 \cdot 0,8 = 0,38 \text{ мин.}$$

Масса основного запаса углекислотно–хладонового состав  $m$ , кг, определяется по формуле

$$m = 1,1 \cdot md \cdot (1 + k_8 / k_6) \quad (4.13)$$

где  $k_8$  – коэффициент, учитывающий остаток углекислотно–хладонового состава в баллонах и трубопроводах,  $k_8=0,2$ .

$$m = 1,1 \cdot 18,3 \cdot (1+0,2/1,13) = 23,7 \text{ кг.}$$

Исходя из расчетов, в помещение с оборудованием, для соблюдения требуемых параметров и установки системы автоматического пожаротушения необходимо установить 1 баллон с углекислотно–хладоновым составом.

В данном разделе был произведен анализ условий труда в рабочем помещении. Уровень условий труда признан допустимым, и данные, полученные из расчетов полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

Также был произведен расчет всех необходимых параметров для кондиционирования воздуха в помещении, т.е. автоматическое поддержание его состояния в помещении в соответствии с определенными требованиями независимо от изменения состояния наружного воздуха и условий в самом помещении.

Электротехническое оборудование в помещении является потенциальным источником возникновения пожара.

## **Заключение**

В результате написания дипломного проекта была спроектирована информационная система для учета мультимедийной продукции торговой компании «Меломан».

Система представляет собой базу данных с пользовательским интерфейсом, позволяющую сотрудникам компании вести учет продукции, выполнять поиск необходимого товара по расширенным категориям, получать справочную информацию о продукте. Интуитивно понятный для пользователей интерфейс программы позволяет быстро и эффективно использовать все функции системы. Конечный продукт предполагает сопровождение программы на долгий период, так как создана с учетом вносимых желаемых изменений заказчика.

В технико-экономической части была рассчитана общая стоимость затрат на разработку системы.

В разделе безопасности жизнедеятельности был произведен анализ условий труда в рабочем помещении, а также расчет всех необходимых параметров для систем кондиционирования и пожаротушения.

В ходе выполнения дипломной работы была достигнута поставленная цель и исполнены все поставленные задачи.

Данная информационная система позволит повысить эффективность в отношении поиска и классифицирования, а также дальнейшего складирования товаров, что, в свою очередь, позволит оптимизировать процесс продаж, а также повысит количество продаваемой продукции. Что приведет к повышению конкурентоспособности данного предприятия на рынке в целом.

## Список используемой литературы

1. Герберт Шилдт Полный справочник по С#. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 752 с. :
2. Дж. Гофф, П. Вайнберг SQL: Полное руководство: Пер. с англ. -2-е изд. – К.: Изд. BHV. 2001
3. Форта, Бен. Освой самостоятельно SQL. 10 минут на урок, 3-е изд. : Пер. с англ. — М. : Издательский дом "Вильямс", 2005.
4. <http://www.imdb.com>
5. Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон. Visual Studio 2010 для профессионалов = Professional Visual Studio 2010. — М.: «Диалектика», 2011. — С. 1184.
6. Алекс Макки. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов = Introducing .NET 4.0: with Visual Studio 2010. — М.: «Вильямс», 2010. — С. 416.
7. Сатимова Е.Г. Проектирование баз данных. Методические указания к выполнению лабораторных работ (для студентов всех специальностей). – Алматы: АИЭС, 2009
8. MICROSOFT SQL SERVER 2008. Реализация и обслуживание. Учебный курс Microsoft/ Пер. с английского – М. «Русская редакция», Спб.: «Питер», 2007.
9. Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Издательский дом «Вильямс», 2000.
10. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. З.Д. Еркешева, Г.Ш. Боканова Алматы: АУЭС, 2013 – 40 с.
11. ГОСТ 12.1.005-88 ОБЩИЕ САНИТАРНО-ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ВОЗДУХУ РАБОЧЕЙ ЗОНЫ
12. ГОСТ 30494-96 ПАРАМЕТРЫ МИКРОКЛИМАТА В ПОМЕЩЕНИЯХ
13. <http://www.mkc-ltd.ru/index.asp?id=1572>
14. <http://www.meloman.kz>

## Приложение А

### Листинг программной части

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Kursach_BD
{
    public partial class Form1 : Form
    {
        public string p3 { get; set; }

        public Form1()
        {
            InitializeComponent();
        }

        public string password;

        private void button1_Click(object sender, EventArgs e)
        {
            password = textBox1.Text;
            if (comboBox1.SelectedIndex == 0)
            {
                if (password.Equals("A"))
                {
                    Form2 f2 = new Form2(p3);
                    f2.Show();
                    Visible = false;
                }
                else
                {
                    MessageBox.Show("Неверный пароль. Попробуйте снова.");
                }
            }
            if (comboBox1.SelectedIndex == 1)
            {
                if (password.Equals("U"))
                {
                    Form3 f3 = new Form3();
                    f3.Show();
                }
            }
        }
    }
}
```

```

        Visible = false;
    }
    else
    {
        MessageBox.Show("Неверный пароль. Попробуйте снова.");
    }
}
if (comboBox1.SelectedIndex == 2)
{
    if (password.Equals("G"))
    {
        Form4 f4 = new Form4();
        f4.Show();
        Visible = false;
    }
    else
    {
        MessageBox.Show("Неверный пароль. Попробуйте снова.");
    }
}
}

private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
}
}

namespace Kursach_BD
{
    public partial class Form2 : Form
    {
        public string constring2;
        public string p3 { get; set; }

        public Form2()
        {
            InitializeComponent();
        }

        public Form2(string zx)
        {
            InitializeComponent();
            p3="Data Source=ERS;Initial Catalog=diplom;Integrated Security=True";
        }
    }
}

```

```
private void button1_Click(object sender, EventArgs e)
{
    Form1 f1 = new Form1();
    f1.Show();
    this.Close();
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    Form21 f2 = new Form21();
    f2.Show();
    Visible = false;
}
```

```
private void button3_Click(object sender, EventArgs e)
{
    Form22 f2 = new Form22();
    f2.Show();
    Visible = false;
}
```

```
private void button4_Click(object sender, EventArgs e)
{
    Form23 f2 = new Form23();
    f2.Show();
    Visible = false;
}
```

```
private void button5_Click(object sender, EventArgs e)
{
    Form24 f2 = new Form24();
    f2.Show();
    Visible = false;
}
```

```
private void button6_Click(object sender, EventArgs e)
{
    Form25 f2 = new Form25();
    f2.Show();
    Visible = false;
}
```

```
private void button7_Click(object sender, EventArgs e)
{
    Form26 f2 = new Form26();
    f2.Show();
    Visible = false;
}
```

```
private void button8_Click(object sender, EventArgs e)
{
```



```

    Form27 f2 = new Form27();
    f2.Show();
    Visible = false;
}

private void button9_Click(object sender, EventArgs e)
{
    Form28 f2 = new Form28();
    f2.Show();
    Visible = false;
}

private void button10_Click(object sender, EventArgs e)
{
    Form29 f2 = new Form29();
    f2.Show();
    Visible = false;
}

private void button11_Click(object sender, EventArgs e)
{
    Form30 f2 = new Form30();
    f2.Show();
    Visible = false;
}

private void button12_Click(object sender, EventArgs e)
{
    Form31 f2 = new Form31();
    f2.Show();
    Visible = false;
}

private void button13_Click(object sender, EventArgs e)
{
    Form32 f2 = new Form32();
    f2.Show();
    Visible = false;
}

private void button14_Click(object sender, EventArgs e)
{
    Form33 f2 = new Form33();
    f2.Show();
    Visible = false;
}

private void button15_Click(object sender, EventArgs e)
{
    Form34 f2 = new Form34();
    f2.Show();
}

```

```

        Visible = false;
    }
}
}

```

```

namespace Kursach_BD

```

```

{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form1 f1 = new Form1();
            f1.Show();
            this.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form9 f9 = new Form9();
            f9.Show();
            Visible = false;
        }

        private void button7_Click(object sender, EventArgs e)
        {
            Form7 f7 = new Form7();
            f7.Show();
            Visible = false;
        }

        private void button8_Click(object sender, EventArgs e)
        {
            Form11 f11 = new Form11();
            f11.Show();
            Visible = false;
        }

        private void button12_Click(object sender, EventArgs e)
        {
            Form8 f8 = new Form8();
            f8.Show();
            Visible = false;
        }

        private void button14_Click(object sender, EventArgs e)
        {

```

```

        Form10 f10 = new Form10();
        f10.Show();
        Visible = false;
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Form17 f7 = new Form17();
        f7.Show();
        Visible = false;
    }

    private void button4_Click(object sender, EventArgs e)
    {
        Form18 f7 = new Form18();
        f7.Show();
        Visible = false;
    }
}
}

namespace Kursach_BD
{
    public partial class Form33 : Form
    {
        public Form33()
        {
            InitializeComponent();
        }

        private void songsBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
            this.songsBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.diplomDataSet);
        }

        private void Form33_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "diplomDataSet.Songs". При необходимости она может быть перемещена или удалена.
            this.songsTableAdapter.Fill(this.diplomDataSet.Songs);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form2 f2 = new Form2();
            f2.Show();
            Visible = false;
        }
    }
}

```

}  
}  
}