

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Проектирование базы данных для типографии ИТ Адимов Ж.И.

Специальность 5В070400 ВТнПО

Выполнил (а) Адимов М.Ж. ВТ-12-2  
(Фамилия и инициалы) группа

Научный руководитель Мусалирова Т.Р. к.т.н. стар. преподаватель  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмурза А.И., к.э.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 03 » 05 20 16 г.  
(подпись)

по безопасности жизнедеятельности:

Трихобин И.Г. Д.х.н., проф.  
(Фамилия и инициалы, ученая степень, звание)  
« 21 » 04 20 16 г.  
(подпись)

по применению вычислительной техники:

Мусалирова Т.Р. к.т.н. старшей преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
« 30 » 05 20 16 г.  
(подпись)

Нормоконтролер: Мусалирова Т.Р. к.т.н. ст. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
« 30 » 05 20 16 г.  
(подпись)

Рецензент: Усаибаева С.Я., ст. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ 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 Аджимов Максим Юсупович  
(фамилия, имя, отчество)

Тема проекта Проектирование базы данных для типографии И.Т. Аджимов Ж.Ш.

утверждена приказом ректора № 21 от «10» марта 2016 г.

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

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

Проектирование базы данных для типографии И.Т. Аджимов Ж.Ш.  
Разработка приложения для базы данных

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

Анализ предметной области в сфере полиграфических услуг  
Постройка логической структуры типографии  
Обзор существующих аналоговых программ  
Анализ имеющегося программного обеспечения  
Разработка базы данных  
Разработка приложения  
Защита законченной части дипломной работы  
Анализ и расчет статистических и вероятностных факторов



Перечень графического материала (с точным указанием обязательных чертежей)

Сайт для заказа приложения для теплоармии  
 Диаграмма развертывания  
 Диаграмма прецедентов  
 ER-диаграмма  
 Семантическая ER-диаграмма для бизнес-правила  
 Генерация кода для MySQL  
 Создание базы данных

Рекомендуемая основная литература

Вешини Л. PHP и MySQL Web-разработчиком  
 Айдарханова М. Основы дискретной теории  
 Салганин З.Б. Овещение рабочих мест

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БЭИД	Тришадзев И.Г.	01.04 - 21.04.16	<i>[Подпись]</i>
Эконом. часть	Бекмурзаев А.У.	15.03 - 03.05.16	<i>[Подпись]</i>
Поприм. ВТ.	Алисаириева Г.Д.	15.03 - 16.04.16	<i>[Подпись]</i>
Кермаконтроллер	Алисаириева Г.Д.	26.03.16	<i>[Подпись]</i>

**Г Р А Ф И К**  
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1.	Теоретическая часть	01.03.16 - 15.03.16	
2.	Выбор программно-аппаратных средств	18.03.16 - 29.03.16	
3.	Работа с базой данных	01.04.16 - 14.04.16	
4.	Разработка интерфейса	15.04.16 - 25.04.16	
5.	Технико-экономические обоснования	26.04.16 - 04.05.16	
6.	Безопасность телекоммуникации	05.05.16 - 06.05.16	

Дата выдачи задания « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_\_\_ г.

Заведующий кафедрой \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Руководитель \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

## **Аннотация**

В данной дипломной работе выполнено проектирование базы данных для типографии ИП «Адимов М.Ж.», а также разработка веб-приложения для спроектированной базы с целью автоматизации работы на предприятии.

В разделе «безопасность жизнедеятельности» был произведен анализ труда, также выполнен расчет искусственного освещения двумя методами и расчет уровня шума.

В экономической части приведено технико-экономическое обоснование данной работы.

Спроектированная база данных и веб-приложение были внедрены в типографию ИП «Адимов Ж.Ш.»

## **Abstract**

In this thesis work done database design for the printing of IE " Adimov Z.S." as well as the development of Web applications designed to automate the database in the enterprise .

In the " life safety " was produced by labor analysis also calculated the artificial lighting the two methods and the calculation of the noise level.

The economic part is given a feasibility study of this work .

Designed database and web application have been introduced to the printer IE " Adimov Z.S. "

## **Андатпа**

Бұл дипломдық жұмыста ЖК «Адимов Ж.Ш.» баспаханасына арналған деректер қоры жобаланды, сонымен қоса кәсіпорындағы жұмысты автоматтандыру мақсатында жобаланған деректер қорына веб-қосымшасы құрастырылды.

«Тіршілік қауіпсізділігі» бөлімінде еңбек талдауы жүргізілген, сондай-ақ жасанды жарықтандыру екі тәсілмен есептелді және шу деңгейі есептелді.

Экономикалық бөлімде осы жұмыстың технико-экономикалық негіздеуі келтірілген.

Жабаланған деректер қоры және веб-қосымшасы ЖК «Адимов Ж.Ш.» баспаханасына енгізілген.

## Содержание

Введение	12
1. Теоретическая часть	13
1.1. Анализ предметной области	13
1.2 Анализ аналогов	13
1.3 Постановка задачи	18
1.3.1 Техническое задание	18
1.3.1 Требования к программному продукту	18
1.4 UML диаграммы	18
1.4.1 Диаграмма развертывания	20
1.4.2 Диаграмма прецедентов	20
2 Выбор программно-аппаратных средств	21
2.1 Google Chrome	21
2.2 PhpMyAdmin	21
2.3 ERwin Data Modeler	22
2.4 Анализ СУБД	22
2.4.1 Microsoft Access	22
2.4.2 Oracle	24
2.4.3 FileMaker	24
2.4.4 SQL Server	25
2.5 Применяемое СУБД MySQL	25
3 Работа с базой данных	27
3.1 Построение ER- диаграммы	27
3.2 Бизнес-правила	27
3.3 Определение атрибутов каждой сущности	31
3.4 Составление реляционных отношений	33
3.5 Генерация кода	37
3.6 Создание базы данных в phpMyAdmin	39
3.7 Заполнение базы данных в phpMyAdmin	41
4. Разработка интерфейса	41
4.1 Описание интерфейса приложения для базы данных	41
5. Техничко-экономическое обоснование	44
5.2 Расчет затрат на разработку информационных технологий	44
5.3 Расчет цены программного продукта	52
5.4 Оценка социально – экономических результатов	53
6. Безопасность жизнедеятельности	53
6.1 Анализ условий труда	53
6.1.1 Пожаробезопасность	54
6.1.2 Электромагнитное и ионизирующее излучения	54
6.1.3 Освещение	54
6.1.4 Параметры микроклимата	55
6.1.5. Эргономические требования	55
6.1.6 Рабочее помещение, в котором ведется разработка.	56

6.2 Технические решения обеспечения безопасности жизнедеятельности	57
6.2.1 Расчёт искусственного освещения методом коэффициента использования.	57
6.2.2 Расчет искусственного освещения точечным методом	59
6.2.3 Расчет уровня шума	61
Заключение	63
Список литературы	64
Приложение А	65
Приложение Б	68
Приложение В	74

## **Введение**

В настоящее время практически любое предприятие нуждается в собственном программном обеспечении, с целью автоматизации некоторых процессов на предприятии для упрощения работы. В век компьютерных технологий очень трудно проделывать работу без помощи специализированных программ. Учитывая нынешние объемы работы человек не может быстро и безошибочно справляться с поставленными задачами, как это делает автоматизированная система или компьютер. База данных является одной из самых распространенных автоматизированных систем, помогающих в работе предприятия. Типография относится к таким предприятиям. Типографии являются предприятиями, имеющие широкий спектр полиграфических услуг. Например, производство печатных СМИ, книг, визиток, буклетов и многих других канцелярских товаров в настоящее время.

Необходимость использования своего программного обеспечения, базы данных, приложения для работы с клиентами и ведения интерактивного учета заказов обусловлена расширением самого предприятия и его клиентской базы. Таким образом, целью данной дипломной работы является разработка базы данных и соответствующего приложения для типографии ИП «Адимов Ж.Ш.».

Для достижения поставленной цели сформируем план задач:

- анализ предметной области в сфере полиграфических услуг;
- построить логическую структуру типографии;
- обзор существующих аналогичных программ;
- анализ используемого программного обеспечения;
- разработка базы данных;
- разработка приложения;
- расчет экономической части дипломной работы;
- анализ и расчет опасных и вредоносных факторов предприятия.



## **1 Теоретическая часть**

### **1.1 Анализ предметной области**

Требуется разработать базу данных для типографии ИП «Адимов Ж.Ш.». В базе данных должна храниться полная информация о сотрудниках предприятия, о заказах и заказчиках, о товарах и услугах, о финансовых процессах и оборотах внутри предприятия. Система также должна вести отчет бухгалтерии.

Типография предоставляет полиграфические услуги и изготавливает товары. Сотрудники на предприятии работают в цехе, на складе, в доставке, проектируют дизайны, занимаются рекламой и т.д. Одним словом, каждый сотрудник имеет свое рабочее место, при этом один сотрудник не может выполнять две или более разных работ. При заказе товара или услуги, информация записывается в базу, причем каждый заказ выполняет какой-то сотрудник. Заработная плата сотрудника напрямую зависит от заказов, выполненных им. Сотрудник может посмотреть в базе свои данные, отчет по своей работе, список заказов, которые он выполнил, свою заработную плату. Директор предприятия также имеет возможность просматривать информацию о работе сотрудников, чтобы выявлять сотрудников с низкой работоспособностью, либо наоборот, для поощрения трудолюбивых сотрудников.

Проведя анализ предметной области, можно выделить базовые сущности:

- Клиенты – сущность содержит информацию о клиентах и способах связи с ними;
- Сотрудники – содержит информацию о сотрудниках, их личные данные и заработную плату;
- Товары – содержит информацию о товарах, описание, цену и дизайн;
- Услуги – содержит информацию об услугах, описание и цену;
- Заказы – содержит информацию о заказах, товаре или услуге, входящих в него, работнике, дате и сумму выплаты.

Система имеет следующие группы пользователей:

- Администратор – имеет все привилегии;
- Оператор БД – возможность изменения полей в БД;
- Сотрудник – может просматривать информацию только о своей работе.

### **1.2 Анализ аналогов**

1. Программное обеспечение для учета и управления в типографии. На сайте [usu.kz](http://usu.kz) можно заказать приложение для своей типографии, которое дает возможность ведения отчетности и управления на предприятии (рисунок 1).

# ПРОГРАММА ДЛЯ ТИПОГРАФИИ

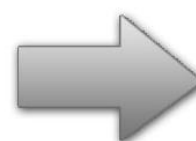
Программное обеспечение для учета  
и управления в типографии

Учет заказов в типографии

Главная Программы Отзывы Поддержка Блог Контакты



Программа УСУ нового поколения  
Версия 4.0 - 10 Факторов успеха



Главная > Программы для учета > Программа для типографии

Рисунок 1 – Сайт для заказа приложения для типографии.

На сайте имеется можно скачать демоверсию приложения, с целью ознакомления с принципом его работы. После скачивания данной демоверсии был проведен анализ приложения.

После открытия приложения, на экране появляется окно авторизации (рисунок 2).

Вход в систему

Пользователь База данных

Аутентификация пользователя

Пользователь \* NIKOLAY

Пароль \*

Укажите роль MAIN

OK Отмена

Рисунок 2 – Окно авторизации

В зависимости от того, какой авторизуется, предоставляются определенные привилегии. Например, при входе через «Администратора» можно не только просматривать таблицы, как при входе через «Сотрудника»,

но и изменять саму базу, а также выдача привилегий тем или иным пользователям. После входа открывается сама база данных (рисунок 3).

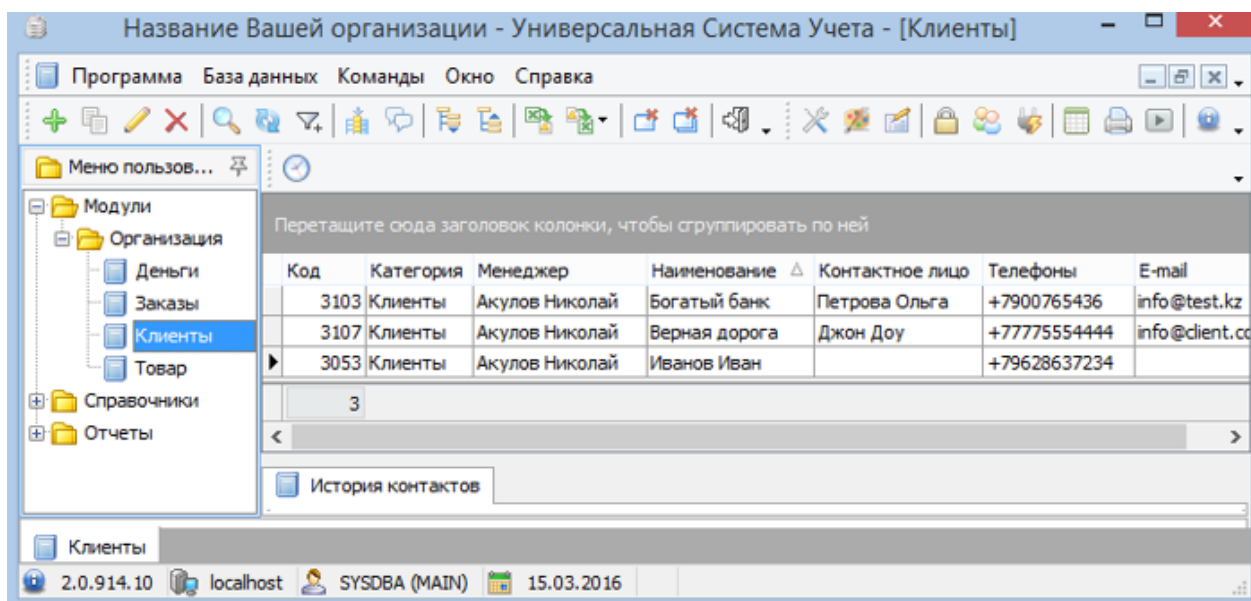


Рисунок 3 – Окно базы данных

Приложение имеет множество функций таких, как просмотр таблиц, проведение отчетов, калькуляции заказа и так далее. Так как вход был выполнен от имени администратора, имеются функции добавления полей, изменения базы. Пример калькуляции заказа приведен на рисунке 4.

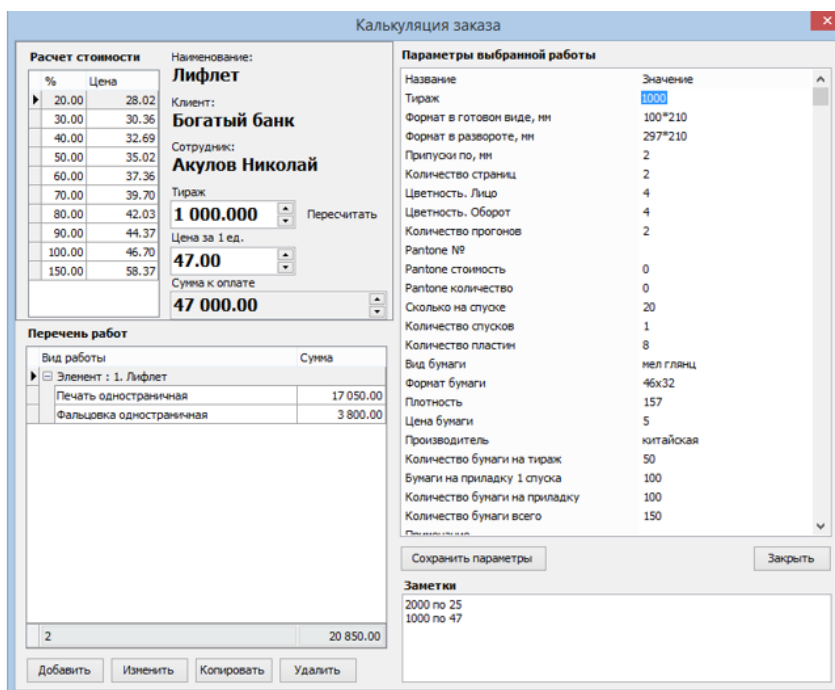


Рисунок 4 – Калькуляция заказа

В целом приложение удобное и просто в использовании, однако к одним из его недостатков можно отнести старомодный интерфейс.

## 2. База данных для типографии «50 КОПЕЕК».

Следующее приложение, подлежащее анализу, это приложение для типографии «50 КОПЕЕК». Сама база данных спроектирована на Microsoft Access. Пример таблицы из этой базы приведен на рисунке 5.

№Заказа	K3	ДатаЗаказа	Дизайнер	Менеджер	ВидОплаты	Скидка,%	Предоплата,%	Макет
1		15.01.2008	ФАЛЬКЕНГОФ Д С	ОНИСИМОВА С Е	р\с	3	100	нет
2		15.01.2008	ФАЛЬКЕНГОФ Д С	ОНИСИМОВА С Е	р\с	0	100	нет
3	3	15.01.2008	НЕТ	КОВАЛЕНКО М Д	р\с	0	100	нет
4	3	15.01.2008	ПАВЛОВСКИЙ С В	МОРГУНОВ А Д	р\с	5	100	нет
5	4	16.01.2008	НЕТ	КОНОВАЛОВА О Н	р\с	0	100	есть
6	5	16.01.2008	НЕТ	КАПЦОВА Г Л	р\с	0	100	есть
7	6	16.01.2008	ОГАНЕСЯН Р Д	МОРГУНОВ А Д	р\с	3	100	нет
8	7	16.01.2008	ПАВЛОВСКИЙ С В	КОВАЛЕНКО М Д	нал	0	100	нет
9	7	17.01.2008	НЕТ	ИГОРЕВА М Д	нал	0	80	есть
10	8	17.01.2008	ИВАНОВА О И	МОРГУНОВ А Д	нал	3	100	нет
11	9	17.01.2008	НЕТ	КАПЦОВА Г Л	нал	0	100	нет
12	9	17.01.2008	ПАВЛОВСКИЙ С В	ИГОРЕВА М Д	нал	0	50	нет
13	10	18.01.2008	ГУДКОВА С В	КАПЦОВА Г Л	кредит	3	100	нет
14	10	18.01.2008	НЕТ	ГУДКОВА Д Ю	кредит	0	100	есть
15	11	18.01.2008	ФАЛЬКЕНГОФ Д С	КОВАЛЕНКО М Д	нал	0	100	нет
16	11	19.01.2008	ГУДКОВА С В	ИГОРЕВА М Д	нал	0	100	нет
17	13	19.01.2008	БОЧКОВ П М	ИГОРЕВА М Д	кредит	0	100	нет
18	12	20.01.2008	ПАВЛОВСКИЙ С В	МОРГУНОВ А Д	нал	5	100	нет
19	12	30.01.2008	НЕТ	КАПЦОВА Г Л	р\с	5	100	есть
20	12	12.02.2008	ГУДКОВА С В	КАПЦОВА Г Л	нал	3	100	нет
21	17	23.02.2008	ГУДКОВА С В	КОВАЛЕНКО М Д	кредит	0	50	нет
22	18	25.02.2008	ПАВЛОВСКИЙ С В	КАПЦОВА Г Л	нал	0	50	нет
23	18	29.02.2008	НЕТ	КОВАЛЕНКО М Д	нал	0	100	есть
24	19	02.03.2008	ФАЛЬКЕНГОФ Д С	ИГОРЕВА М Д	кредит	3	80	нет
25	20	10.03.2008	ИВАНОВА О И	МОРГУНОВ А Д	нал	0	100	нет

Рисунок 5 – Таблица «Заказы»

Само приложение написано на языке C#. Главная форма приложения показана на рисунке 6.

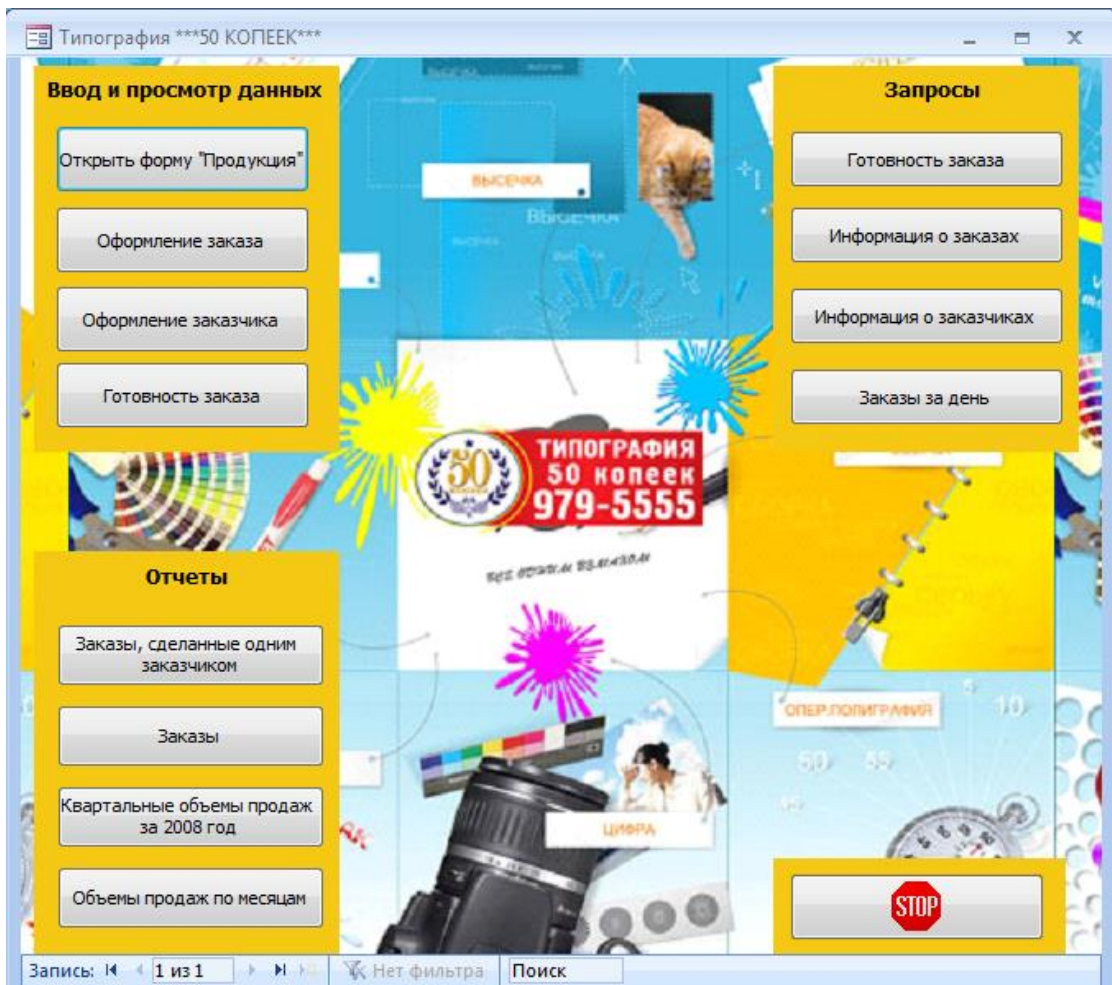


Рисунок 6 – Главная форма

Приложение имеет возможность ведение отчетов о финансовых операциях (рисунок 7).

Информация о заказах 14 декабря 2008 г.  
2:48:56

№Заказа	КЗ	ДатаЗаказа	КодПродукции	СтоимостьПродукции	Скидка,%	Количество	СтоимостьЗаказа
1	1	15.01.2008	1	310,00р.	3	3	902,1
2	1	15.01.2008	3	360,00р.	0	1	360
3	3	15.01.2008	10	1 450,00р.	0	6	8700
4	3	15.01.2008	3	360,00р.	5	4	1368
5	4	16.01.2008	12	1 560,00р.	0	6	9360
6	5	16.01.2008	13	10 800,00р.	0	1	10800
7	6	16.01.2008	14	11 420,00р.	3	3	33232,2
8	7	16.01.2008	15	12 460,00р.	0	4	49840
9	7	17.01.2008	16	15 580,00р.	0	2	31160
10	8	17.01.2008	17	6 230,00р.	3	1	6043,1
11	9	17.01.2008	18	6 650,00р.	0	10	66500
12	9	17.01.2008	19	6 960,00р.	0	5	34800
13	10	18.01.2008	20	8 410,00р.	3	10	81577
14	10	18.01.2008	25	5 190,00р.	0	20	103800
15	11	18.01.2008	23	3 630,00р.	0	4	14520
16	11	19.01.2008	25	5 190,00р.	0	6	31140

Рисунок 7 – Пример формирования отчета



Данное приложение позволяет автоматизировать работу менеджеров по работе с клиентами в типографии. К недостаткам можно отнести старый интерфейс, а также то, что база построена на MicrosoftAccess, что ограничивает возможность использования приложения на других операционных системах.

### **1.3 Постановка задачи**

#### **1.3.1 Техническое задание**

Основной задачей данной дипломной работы является проектирование базы данных для типографии ИП «Адимов Ж.Ш.», которая могла бы хранить данные о товарах и полиграфических услугах, данные о клиентах, данные о сотрудниках, важную информацию о заказах. База данных проектируется с целью автоматизации работы на предприятии и хранения информации в цифровом виде. С помощью БД должны вестись отчеты о работе сотрудников, финансовых оборотах в предприятии.

Необходимо разработать приложение для удобного пользования базой данных. Спроектированная база данных должна быть связана с разработанным приложением.

#### **1.3.2 Требования к программному продукту**

База данных должна осуществлять:

- хранение информации о товарах;
- хранение информации об услугах;
- хранение информации о заказах;
- хранение информации о заказчиках;
- хранение информации о работниках;
- хранение информации о поставщиках;
- отчет работы сотрудников;
- хранение информации о дизайнах.

Требования к приложению:

- удобный в пользовании и современный интерфейс;
- возможность авторизации как от администратора, так и от обычного сотрудника;
- безопасное хранение ценных данных;
- минимальные затраты ресурсов.

### **1.4 UML диаграммы**

Диаграмма UML представляет собой частичное графическое представление (вид) модели системы по разработке, внедрению или уже

существующих систем . UML диаграмма содержит графические элементы (символы) - UML узлов, соединенных с ребрами (также известные как пути или потоки) - которые представляют элементы в проектируемой модели системы UML. Модель системы UML также может содержать другие документы, такие как прецедентов, написанных в качестве шаблонных текстов. Вид диаграммы определяется первичными графическими символами, показанными на диаграмме.

### 1.4.1 Диаграмма классов

Диаграмма классов UML является структурная схема , которая показывает структуру проектируемой системы на уровне классов и интерфейсов , показывает их особенности , ограничения и отношения - ассоциации, обобщения, зависимости и т.д.

Некоторые распространенные типы диаграмм классов являются:

- Диаграмма модели домена ,
- Диаграмма классов реализации .

Диаграмма объекта можно рассматривать как диаграммы классов уровня экземпляра , который показывает характеристики экземпляра классов и интерфейсов ( объектов ) , слоты со спецификациями значений , а также ссылки ( экземпляры ассоциации ) .

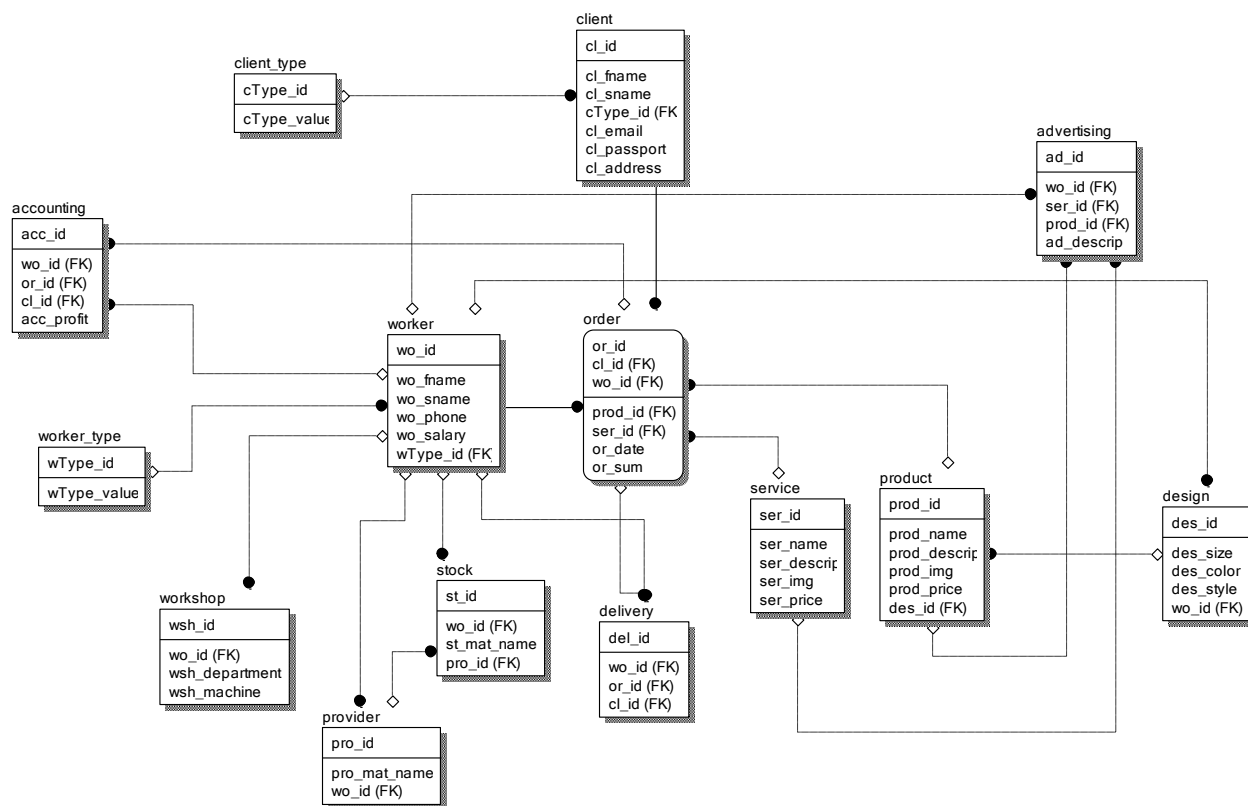


Рисунок 8 – Диаграмма классов

## 1.4.2 Диаграмма развертывания

Диаграмма развертывания представляет собой структурную схему, которая показывает архитектуру системы как развертывание (распределение) программных артефактов для целей развертывания.

Артефакты представляют собой конкретные элементы в физическом мире, которые являются результатом процесса развития. Примерами артефактов являются исполняемые файлы, библиотеки, архивы, схемы баз данных, файлы конфигурации и т.д.

Цель развертывания обычно представлен узлом, который является либо аппаратное устройство или некоторое выполнение программной среды. Узлы могут быть соединены через канал связи для создания сетевых систем произвольной сложности.

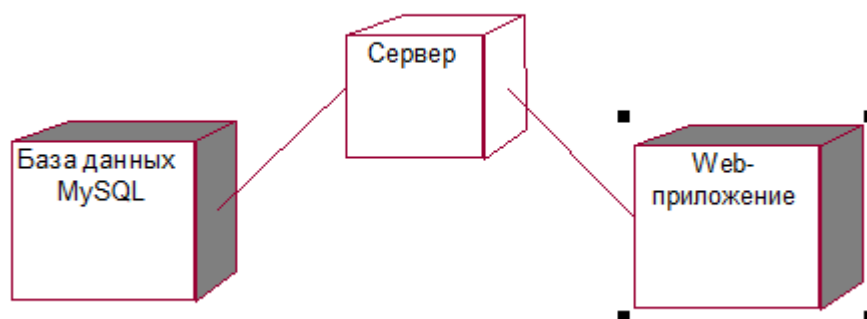


Рисунок 9 – Диаграмма развертывания для базы данных ИП «Адимов Ж.Ш.»

## 1.4.3 Диаграмма прецедентов

Диаграммой прецедентов, как правило, называют диаграммы поведения, используемый для описания набора действий (варианты использования), что некоторые системы или субъект должен или может выполнять в сотрудничестве с одним или несколькими внешними пользователями системы (актеры). Каждый случай использования должен обеспечивать некоторую наблюдаемый и ценный результат актеров или других заинтересованных сторон системы.

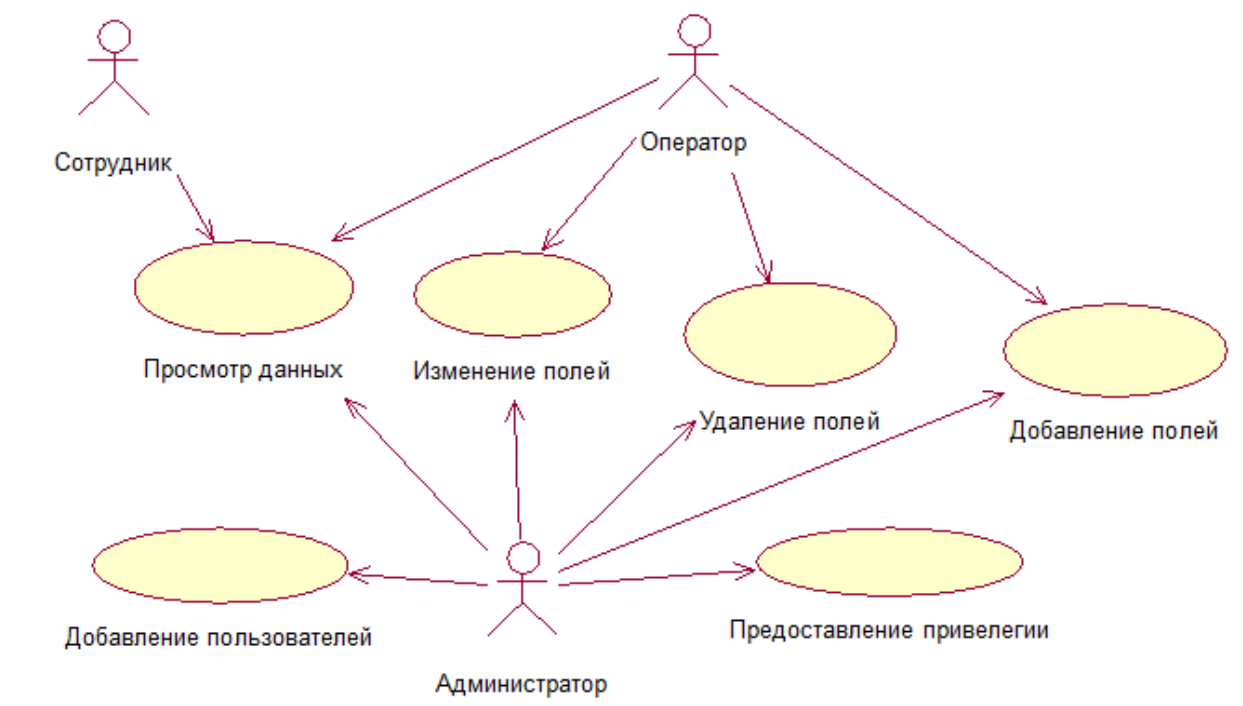


Рисунок 10 – Диаграмма прецедентов для базы данных ИП «Адимов Ж.Ш.»

## 2 Выбор программно-аппаратных средств

### 2.1 Google Chrome

Интернет-браузер, разработанный Google Inc., который сочетает в себе минималистичный дизайн и сложные технологии, чтобы сделать работу в Интернете быстрее, безопаснее и легче. Браузер Google Chrome предлагает функции, включая доступ к любимым страницам мгновенно с эскизами, на рабочем столе для запуска веб-приложений и независимо друг от друга работают вкладки в браузере, чтобы предотвратить сбой браузера.

В консоли предоставляется две функции тестирования веб-страниц и приложений:

- Консоль может настроить отладку, используя такие методы консоли API, как например `console.log ( )` и `console.profile ( )` .

- оболочка для консоли, в которую вы можете вводить команды автоматически, также использовать функции автозаполнения, и `document`, и Chrome DevTools . Вы также можете выполнять JavaScript непосредственно в консоли и используя методы API в командной строке.

### 2.2 PhpMyAdmin

PhpMyAdmin является одним из наиболее популярных приложений для управления базами данных MySQL. Это бесплатный инструмент,

написанный на PHP. С помощью этой программы вы можете создавать, изменять, удалять, импортировать и экспортировать таблицы базы данных MySQL. Вы можете выполнять запросы MySQL, оптимизировать, ремонт и проверить таблицы, изменить параметры сортировки и выполнять другие команды управления базами данных. Все клиенты могут управлять своими базами данных MySQL через предустановленное программное обеспечение PhpMyAdmin.

## **2.3 ERwin Data Modeler**

ERwin Data Modeler позволяет визуализировать сложные структуры данных, информационные ресурсы инвентаризации и установления стандартов в масштабах всего предприятия для управления данными. ERwin упрощает процесс проектирования, предоставляя простую в использовании графическую среду, которая упрощает разработку и ведение базы данных, позволяет автоматизировать множество трудоемких задач и улучшает коммуникацию, повышая эффективность и качество данных, в то же время при одновременном снижении расходов.

Функции ERwin Data Modeler:

- Визуализация сложных структур данных
- Проектирование баз данных, генерация кода
- Стандарты определения
- Модель базы данных и сравнение

## **2.4 Анализ СУБД**

Базы данных – это организованный механизм для хранения, управления и извлечения информации. Проще говоря, это организованная совокупность данных. Система управления базами данных (СУБД), такие как Access, FileMakerPro, Oracle, SQL Server и MySQL предоставляет вам программные инструменты, необходимые для организации этих данных на гибкой основе. Она включает в себя объекты добавления, изменения или удаления данных из базы данных, выполнять запросы о данных, хранящихся в базе данных и создавать отчеты, суммирующих выбранное содержимое.

### **2.4.1 MicrosoftAccess**

MicrosoftAccess является реляционной системой управления базами данных (СУБД), как Oracle и SQL / Server. Эта СУБД сочетает в себе ядро базы данных, известный как JET, разработке форм, отчетов, графики и VisualBasic для выполнения пользовательских программ.



Microsoft Access предоставляет пользователям с одним из самых простых и гибких решений СУБД на рынке сегодня. Microsoft Access доступен с Microsoft Office Профессиональный набор бизнес-продуктов, поэтому никакого дополнительного программного обеспечения базы данных не требуется, если ваша компания покупает компьютеры с этим набором продуктов, которые уже установлены. Базы данных MS Access, вероятно, будут доступны

MS Access является наиболее широко используемой настольной системой базы данных в мире. Типичная стоимость составляет менее \$ 150 на одного пользователя в месяц.

Если поддержка баз данных важна для вас, то доступ может быть вашим лучшим выбором, так как доступ имеет больше консультантов поддержки и развития, чем любая другая система базы данных рабочего стола. Доступ значительно дешевле внедрить и поддерживать по сравнению с более крупными системами баз данных, таких как Oracle или SQL Server.

Microsoft Access хорошо интегрируется с другими членами Microsoft офисного пакета продуктов (Excel, Word, Outlook и т.д.).

Ниже приведены некоторые причины, почему Microsoft Access не может быть вашим лучшим выбором для разработки приложений баз данных вашей компании. В некоторых случаях ваши компьютерные системы не могут быть достаточно, чтобы справиться с любой системой управления базами данных надежно надежным.

Microsoft Office набор программных продуктов не в утвержденном наборе технологий вашей компании.

Вы должны захватить время критических операций - например, от лабораторного прибора.

Любая одна таблица в базе данных может иметь более 1 млн большой и средней ширины записей, или 5 миллионов малых ширины записи (т.е. 3-4 числовых или коротких текстовых полей).

Вы регулярно иметь более 15 человек, активно использующие базу данных, в то же время - к примеру, быстрого ввода данных и / или запросов и отчетов.

Ваши серверы баз данных испытывают периодические, неожиданные сбои - например, более чем на 1 месяц (в этом случае вы будете иметь проблемы с любой системой баз данных).

Вы требуют обширной безопасности базы данных из-за очень чувствительной природы данных. (Хотя вы можете обойти эту проблему с помощью SQL Server в качестве движка базы данных с доступом переднего конца.)

Ваша компания должна работать с несколькими базами данных Access, некоторые с безопасностью доступа включена и некоторые без.

Вы будете делить базу данных с различными операционными системами (Windows 98, Windows ME, Windows 2000, Windows XP, и т.д.).

## 2.4.2 Oracle

Oracle состоит из множества процессов, запущенных в вашей операционной системе. Это процессы управления, хранения и доступа к данным.

Oracle Developer представляет собой приложение Builder 4GL GUI. С помощью Developer, вы можете создавать формы, отчеты и графики. Отчеты состоят из двух компонентов Oracle Developer. Более ранние версии создали клиент-серверные приложения, но более поздние версии создают веб-приложения, которые работают под Oracle Application Server (OAS).

OAS лицензируется отдельно и очень дорого (как и его конкуренты с закрытым кодом).

Oracle Enterprise Manager (OEM) является инструментом Enterprise GUI для управления базой данных. С помощью этого инструмента вы можете выполнить какие-либо действия в базе данных, которые требуются. Этот инструмент используется в основном для введения, но также могут быть предоставлены разработчикам в настройке приложений и мониторинга.

В качестве примечания, Oracle предоставляет все свое программное обеспечение с лицензией разработчика. Эта лицензия позволяет опробовать программное обеспечение бесплатно. Вы не можете создавать производственные приложения без платной лицензии, но разработка с инструментами является приемлемым.

## 2.4.3 FileMaker

FileMaker (или его последняя версия, FileMaker Pro) представляет собой реляционную базу данных, приложение известно как простое в использовании и за его способность динамически обслуживать веб-страницы без необходимости использования дополнительных приложений сторонних разработчиков. С помощью встроенного в веб-сервер (так называемый "Web Companion") и вне коробки бизнес-шаблоны, FileMaker Pro получил высокую оценку за то, что приложение идеально для начинающих администраторов веб-сайтов, которые хотят быстро (буквально одним щелчком мыши) добавить возможности публикации веб-базы данных на их веб-сайт или интранет компании.

FileMaker Pro (ФСМ) имеет встроенные возможности совместного использования файлов. Ваша база данных FileMaker Pro может быть доступна любому человеку с подключенного к сети компьютера и другой копии той же версии приложения ФМП.

FileMaker Pro также позволяет пользователям надежно публиковать в интернете и обмениваться их базы данных с другим пользователями. Кроме того, его интуитивный интерфейс делает его легким и простым в использовании, которая предоставляет пользователям более приятным.

#### 2.4.4 SQL Server

SQL является стандартным языком для доступа и управления базами данных.

Microsoft SQL Server является компьютерным приложением, используемое для создания настольных ПК, корпоративных и веб - приложений на основе баз данных. Он используется на различных уровнях и с различными целями.

Базы данных SQL Server являются одними из наиболее распространенных баз данных в использовании, частично благодаря тому, что насколько легко создавать и поддерживать их. С помощью бесплатного графического интерфейса программы (GUI) пользователя, такие как управление SQL Server , вам не нужно беспокоиться о командной строке.

Функции SQL:

- выполнение запросов к базе данных
- извлечение данных из базы данных
- выполнение записи в базе данных
- обновление записи в базе данных
- удаление записи из базы данных
- создание новой базы данных
- создание новых таблиц в базе данных
- создание хранимых процедур в базе данных
- создание представлений в базе данных
- устанавливать разрешения на таблицы, хранимые процедуры и представления

#### 2.5 Применяемое СУБД MySQL

База данных MySQL представляет собой базу данных веб-хостинг, который используется для хранения информации веб-сайта, как сообщения в блоге или информации о пользователе. Это наиболее распространенный тип реляционной базы данных в интернете сегодня. Это отчасти потому, что она полностью свободна, но и очень мощная.

В основных условиях, база данных MySQL может хранить любой тип данных, который вы хотите. Это позволит вам быстро хранить и извлекать информацию и несколько посетителей веб-сайта могут использовать его в одно время. Вы будете использовать операторы SQL для выполнения всего этого. В более технических терминах, MySQL имеет следующие особенности:

- Кроссплатформенная поддержка
- Хранимые процедуры
- Триггеры
- Курсоры

- Обновляемые просмотры
  - Поддержка VARCHAR
  - INFORMATION\_SCHEMA
  - Строгий режим
  - X / Open XA поддержка распределенной обработки транзакций (DTP); двухфазной фиксации как часть этого, используя движок InnoDB от Oracle
  - Независимые механизмы хранения (MyISAM для скорости чтения, InnoDB для транзакций и ссылочной целостности, MySQL архив для хранения исторических данных в ограниченном пространстве)
  - Операции с двигателями InnoDB, BDB и хранения данных кластера; с точки сохранения InnoDB
  - Поддержка SSL
  - Кэширование запросов
  - Суб-SELECTS
  - Полнотекстовая индексация и поиск с помощью MyISAM двигателя
  - Встроенные в базу данных библиотеки
  - Частичная поддержка Unicode
  - ACID соответствия с использованием InnoDB, BDB и кластерный двигатель
  - Без совместного использования кластеризации через MySQL Cluster
- MySQL является базой данных выбора для нескольких различных языков веб- программирования, включая PHP, Ruby On Rails и Python. Эти языки программирования очень легко подключаются к базе данных MySQL.
- Базой данных может быть трудно управлять, используя только отдельные операторы SQL. Гораздо проще управлять базой данных с помощью графического интерфейса пользователя (GUI). Один из лучших и наиболее популярный графический интерфейс для управления базой данных MySQL называется PhpMyAdmin.

### 3 Работа с базой данных

#### 3.1 Построение ER- диаграммы

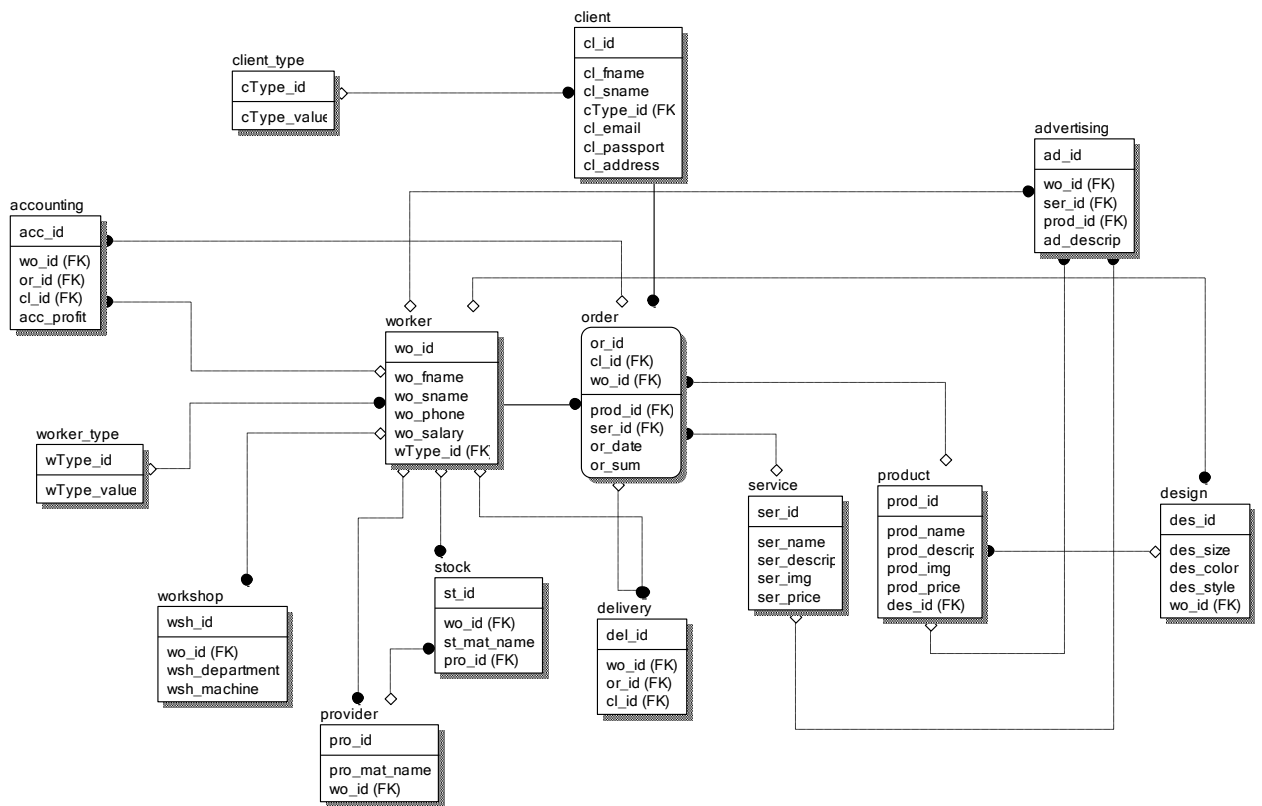


Рисунок 11 – ER-диаграмма

#### 3.2 Бизнес-правила

1. Каждый работник имеет свой тип (профессию), в то же время у каждой профессии может быть несколько работников (рисунок 12).

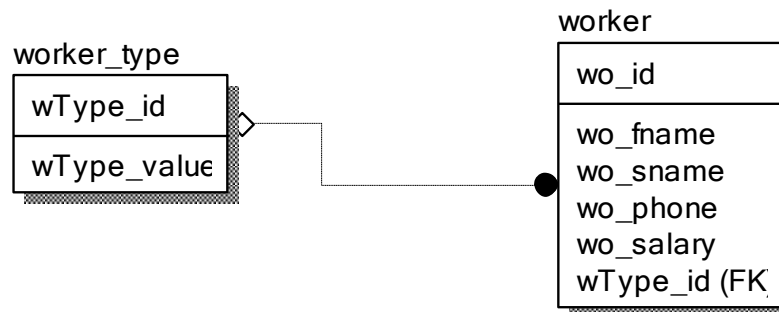


Рисунок 12 – Сегмент ER-диаграммы для бизнес-правила 1

2. Каждый клиент имеет свой тип (юридическое или физическое лицо), а у каждого типа может быть несколько клиентов (рисунок 13).





Рисунок 13 – Сегмент ER-диаграммы для бизнес-правила 2

3. У каждого заказа имеется свой клиент, в то же время любой клиент может сделать несколько заказов (рисунок 14).

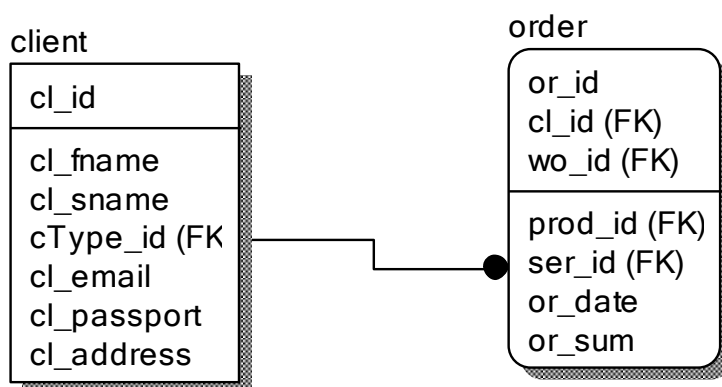


Рисунок 14 – Сегмент ER-диаграммы для бизнес-правила 3

4. Каждый заказ выполняет какой-то работник, а любой работник может выполнять несколько заказов (рисунок 15).

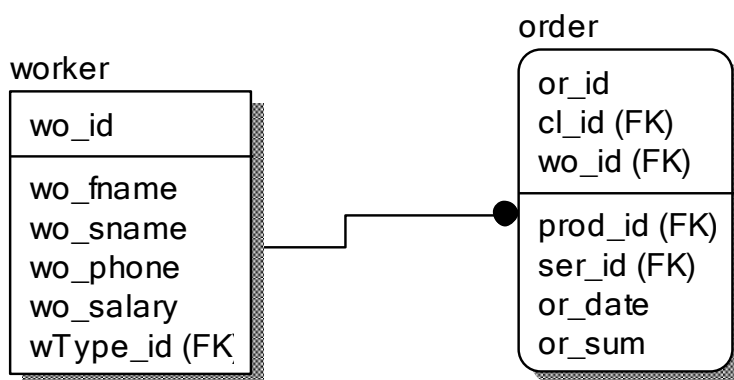


Рисунок 15 – Сегмент ER-диаграммы для бизнес-правила 4

5. Каждый заказ включает в себя либо товар, либо услугу, либо то и другое. Любой товар или услуга могут входить в несколько заказов (рисунок 16).

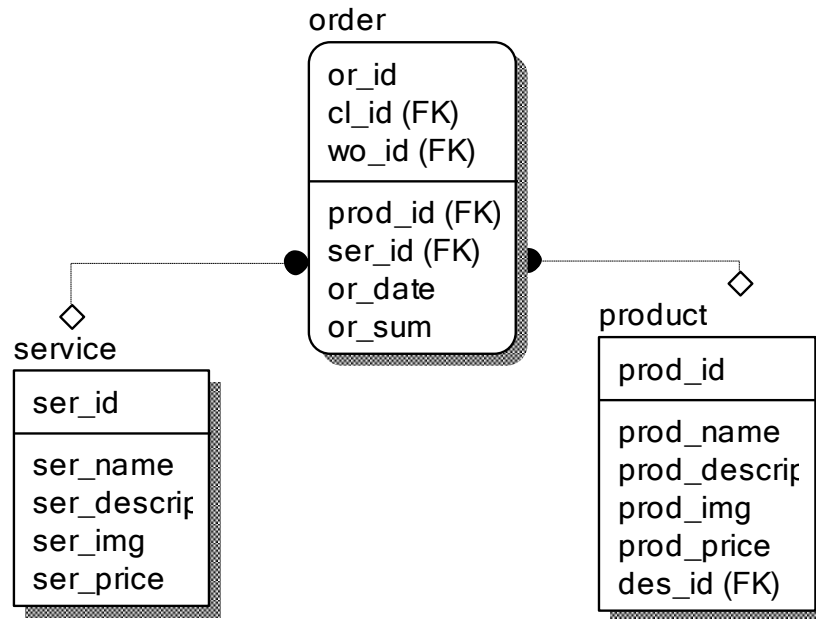


Рисунок 16 – Сегмент ER-диаграммы для бизнес-правила 5

6. Каждый продукт (товар) имеет свой дизайн, каждый дизайн может принадлежать нескольким товарам сразу (рисунок 17).

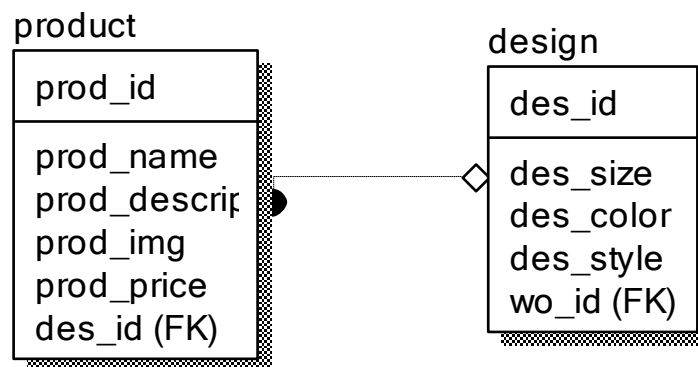


Рисунок 17 – Сегмент ER-диаграммы для бизнес-правила 6

7. Каждую рекламную работу проводит только один работник, однако любой работник рекламы может выполнять несколько рекламных работ (рисунок 17).

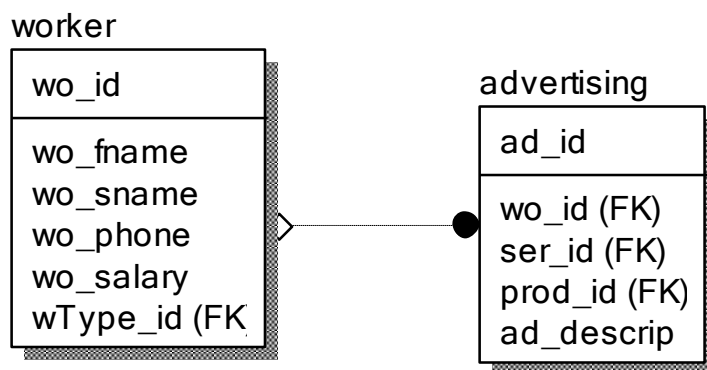


Рисунок 18 – Сегмент ER-диаграммы для бизнес-правила 7

8. Доставкой того или иного заказа занимается один работник. В то же время один работник может заниматься доставкой нескольких заказов (рисунок 19).

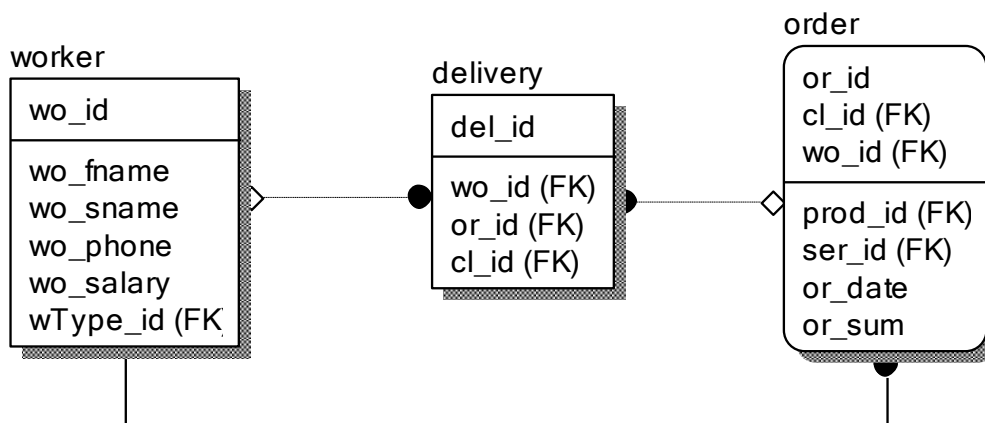


Рисунок 19 – Сегмент ER-диаграммы для бизнес-правила 8

9. Каждый работник в цехе привязан только к своему станку, причем к каждому станку могут быть привязаны несколько работников цеха (рисунок 20).

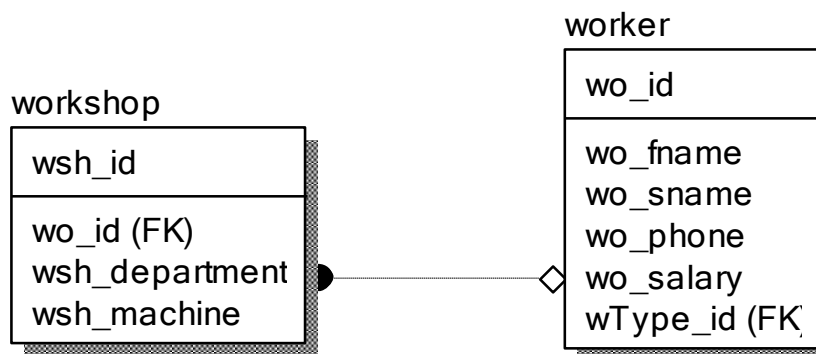


Рисунок 20 – Сегмент ER-диаграммы для бизнес-правила 9

### 3.3 Определение атрибутов каждой сущности

В таблице 1 приведен определение атрибутов каждой сущности.  
Таблица 1 - Определение атрибутов каждой сущности

Сегмент ER модели	Описание							
<p>worker</p> <table border="1"> <tr><td>wo_id</td></tr> <tr><td>wo_fname</td></tr> <tr><td>wo_lname</td></tr> <tr><td>wo_phone</td></tr> <tr><td>wo_salary</td></tr> <tr><td>wType_id (FK)</td></tr> </table>	wo_id	wo_fname	wo_lname	wo_phone	wo_salary	wType_id (FK)	<p>Таблица «Работники» содержит следующие данные:  Wo_id – идентификатор работника  Wo_fname – имя работника  Wo_lname – фамилия работника  Wo_salary – заработная плата работника  WType_id – идентификатор типа работника</p>	
wo_id								
wo_fname								
wo_lname								
wo_phone								
wo_salary								
wType_id (FK)								
<p>worker_type</p> <table border="1"> <tr><td>wType_id</td></tr> <tr><td>wType_value</td></tr> </table>	wType_id	wType_value	<p>Таблица «Тип работника» содержит следующие данные:  WType_id – идентификатор типа работника  WType_value – значение</p>					
wType_id								
wType_value								
<p>client_type</p> <table border="1"> <tr><td>cType_id</td></tr> <tr><td>cType_value</td></tr> </table>	cType_id	cType_value	<p>Таблица «Тип клиента» содержит следующие данные:  CType_id – идентификатор типа клиента  CType_value – значение</p>					
cType_id								
cType_value								
<p>client</p> <table border="1"> <tr><td>cl_id</td></tr> <tr><td>cl_fname</td></tr> <tr><td>cl_lname</td></tr> <tr><td>cType_id (FK)</td></tr> <tr><td>cl_email</td></tr> <tr><td>cl_passport</td></tr> <tr><td>cl_address</td></tr> </table>	cl_id	cl_fname	cl_lname	cType_id (FK)	cl_email	cl_passport	cl_address	<p>Таблица «Клиенты» содержит следующие данные:  Cl_id – идентификатор клиента  Cl_fname – имя клиента  Cl_lname – фамилия клиента  CType_id – идентификатор типа клиента  Cl_passport – паспорт клиента  Cl_email – почта клиента  Cl_address – адрес клиента</p>
cl_id								
cl_fname								
cl_lname								
cType_id (FK)								
cl_email								
cl_passport								
cl_address								
<p>workshop</p> <table border="1"> <tr><td>wsh_id</td></tr> <tr><td>wo_id (FK)</td></tr> <tr><td>wsh_department</td></tr> <tr><td>wsh_machine</td></tr> </table>	wsh_id	wo_id (FK)	wsh_department	wsh_machine	<p>Таблица «Цех» содержит следующие данные:  Wsh_id – идентификатор цеха  Wo_id – идентификатор работника  Wsh_department – отдел  Wsh_machine – станок (машина)</p>			
wsh_id								
wo_id (FK)								
wsh_department								
wsh_machine								
<p>delivery</p> <table border="1"> <tr><td>del_id</td></tr> <tr><td>wo_id (FK)</td></tr> <tr><td>or_id (FK)</td></tr> <tr><td>cl_id (FK)</td></tr> </table>	del_id	wo_id (FK)	or_id (FK)	cl_id (FK)	<p>Таблица «Доставка»:  Del_id – идентификатор доставки  Wo_id – идентификатор работника  Or_id – идентификатор заказа  Cl_id – идентификатор клиента</p>			
del_id								
wo_id (FK)								
or_id (FK)								
cl_id (FK)								

*продолжение таблицы 1*

<p>provider</p> <table border="1"> <tr><td>pro_id</td></tr> <tr><td>pro_mat_name</td></tr> <tr><td>wo_id (FK)</td></tr> </table>	pro_id	pro_mat_name	wo_id (FK)	<p>Таблица «Поставщики» содержит следующие данные:          Pro_id – идентификатор поставщика          Pro_mat_name – название материала          Wo_id – идентификатор работника</p>				
pro_id								
pro_mat_name								
wo_id (FK)								
<p>stock</p> <table border="1"> <tr><td>st_id</td></tr> <tr><td>wo_id (FK)</td></tr> <tr><td>st_mat_name</td></tr> <tr><td>pro_id (FK)</td></tr> </table>	st_id	wo_id (FK)	st_mat_name	pro_id (FK)	<p>Таблица «Склад» содержит следующие данные:          St_id – идентификатор склада          St_mat_name – название материала          Wo_id – идентификатор работника          Pro_id – идентификатор поставщика</p>			
st_id								
wo_id (FK)								
st_mat_name								
pro_id (FK)								
<p>accounting</p> <table border="1"> <tr><td>acc_id</td></tr> <tr><td>wo_id (FK)</td></tr> <tr><td>or_id (FK)</td></tr> <tr><td>cl_id (FK)</td></tr> <tr><td>acc_profit</td></tr> </table>	acc_id	wo_id (FK)	or_id (FK)	cl_id (FK)	acc_profit	<p>Таблица «Бухгалтерия» содержит следующие данные:          Acc_id – идентификатор учета          Wo_id – идентификатор работника          Or_id – идентификатор заказа          Cl_id – идентификатор клиента          Acc_profit – прибыль</p>		
acc_id								
wo_id (FK)								
or_id (FK)								
cl_id (FK)								
acc_profit								
<p>order</p> <table border="1"> <tr><td>or_id</td></tr> <tr><td>cl_id (FK)</td></tr> <tr><td>wo_id (FK)</td></tr> <tr><td>prod_id (FK)</td></tr> <tr><td>ser_id (FK)</td></tr> <tr><td>or_date</td></tr> <tr><td>or_sum</td></tr> </table>	or_id	cl_id (FK)	wo_id (FK)	prod_id (FK)	ser_id (FK)	or_date	or_sum	<p>Таблица «Заказы» содержит следующие данные:          Or_id – идентификатор заказа          Cl_id – идентификатор клиента          Wo_id – идентификатор работника          Prod_id – идентификатор товара          Ser_id – идентификатор услуги          Or_Date – дата заказа          Or_Sum – сумма выплаты</p>
or_id								
cl_id (FK)								
wo_id (FK)								
prod_id (FK)								
ser_id (FK)								
or_date								
or_sum								
<p>service</p> <table border="1"> <tr><td>ser_id</td></tr> <tr><td>ser_name</td></tr> <tr><td>ser_descrip</td></tr> <tr><td>ser_img</td></tr> <tr><td>ser_price</td></tr> </table>	ser_id	ser_name	ser_descrip	ser_img	ser_price	<p>Таблица «Услуги» содержит следующие данные:          Ser_id – идентификатор услуги          Ser_name – название услуги          Ser_descrip – описание          Ser_img – изображение          Ser_price – цена услуги</p>		
ser_id								
ser_name								
ser_descrip								
ser_img								
ser_price								
<p>product</p> <table border="1"> <tr><td>prod_id</td></tr> <tr><td>prod_name</td></tr> <tr><td>prod_descrip</td></tr> <tr><td>prod_img</td></tr> <tr><td>prod_price</td></tr> <tr><td>des_id (FK)</td></tr> </table>	prod_id	prod_name	prod_descrip	prod_img	prod_price	des_id (FK)	<p>Таблица «Товары» содержит следующие данные:          Prod_id – идентификатор товара          Prod_name – название товара          Prod_descrip – описание товара          Prod_img – изображение          Prod_price – цена товара          Des_id – идентификатор дизайна</p>	
prod_id								
prod_name								
prod_descrip								
prod_img								
prod_price								
des_id (FK)								

*продолжение таблицы 1*

<p>design</p> <table border="1"> <tr><td>des_id</td></tr> <tr><td>des_size</td></tr> <tr><td>des_color</td></tr> <tr><td>des_style</td></tr> <tr><td>wo_id (FK)</td></tr> </table>	des_id	des_size	des_color	des_style	wo_id (FK)	<p>Таблица «Дизайны» содержит следующие данные:          Des_id – идентификатор дизайна          Des_size – размеры          Des_color – цвет          Des_style – стиль          Wo_id – идентификатор работника</p>
des_id						
des_size						
des_color						
des_style						
wo_id (FK)						
<p>advertising</p> <table border="1"> <tr><td>ad_id</td></tr> <tr><td>wo_id (FK)</td></tr> <tr><td>ser_id (FK)</td></tr> <tr><td>prod_id (FK)</td></tr> <tr><td>ad_descrip</td></tr> </table>	ad_id	wo_id (FK)	ser_id (FK)	prod_id (FK)	ad_descrip	<p>Таблица «Реклама» содержит следующие данные:          Ad_id – идентификатор рекламы          Wo_id – идентификатор работника          Ser_id – идентификатор услуги          Prod_id – идентификатор товара          Ad_descrip – описание</p>
ad_id						
wo_id (FK)						
ser_id (FK)						
prod_id (FK)						
ad_descrip						

### 3.4 Составление реляционных отношений

Таблица 2 - Схема отношения «Работники»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Wo_id	number	Первичный ключ
Имя	Wo_fName	string	null
Фамилия	Wo_lName	string	null
Зарплата	Wo_salary	number	null
Код типа работника	WType_id	number	Внешний ключ к таблице «Тип работника»

Таблица 3 - Схема отношения «Тип работника»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	WType_id	number	Первичный ключ
Значение	WType_value	string	null

Таблица 4 - Схема отношения «Тип клиента»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	CType_id	number	Первичный ключ
Значение	CType_value	string	null

Таблица 5 - Схема отношения «Клиенты»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Cl_id	number	Первичный ключ
Имя	Cl_fName	string	null
Фамилия	Cl_lName	string	null
Код типа клиента	CType_id	number	Внешний ключ к таблице «Тип клиента»
Паспорт	Cl_passport	number	null
Почта	Cl_email	string	null
Адрес	Cl_address	string	null

Таблица 6 - Схема отношения «Цех»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Wsh_id	number	Первичный ключ
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»
Отдел	Wsh_department	string	null
Станок	Wsh_machine	string	null

Таблица 7 - Схема отношения «Склад»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	St_id	number	Первичный ключ
Название материала	St_mat_name	string	null
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»
Код поставщика	Pro_id	number	Внешний ключ к таблице «Поставщики»

Таблица 8 - Схема отношения «Доставка»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Del_id	number	Первичный ключ
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»
Код заказа	Or_id	number	Внешний ключ к таблице «Заказы»
Код клиента	Cl_id	number	Внешний ключ к таблице «Клиенты»

Таблица 9 - Схема отношения «Поставщики»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Pro_id	number	Первичный ключ
Имя	Pro_name	string	null
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»

Таблица 10 - Схема отношения «Бухгалтерия»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Acc_id	number	Первичный ключ
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»
Код заказа	Or_id	number	Внешний ключ к таблице «Заказы»
Код клиента	Cl_id	number	Внешний ключ к таблице «Клиенты»
Прибыль	Acc_profit	number	null

Таблица 11 - Схема отношения «Заказы»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Or_id	number	Первичный ключ
Код клиента	Cl_id	number	Внешний ключ к таблице «Клиенты»

*продолжение таблицы 11*

Код работника	Wo_id	number	Внешний ключ к таблице «Работники»
Код товара	Prod_id	number	Внешний ключ к таблице «Товары»
Код услуги	Ser_id	number	Внешний ключ к таблице «Услуги»
Дата	Or_Date	datetime	null
Сумма	Or_Sum	number	null



Таблица 12 - Схема отношения «Услуги»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Ser_id	number	Первичный ключ
Название	Ser_name	string	null
Описание	Ser_descrip	string	null
Изображение	Ser_img	string	null
Цена	Ser_price	number	null

Таблица 13 - Схема отношения «Товары»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Prod_id	number	Первичный ключ
Название	Prod_name	string	null
Описание	Prod_descrip	string	null
Изображение	Prod_img	string	null
Цена	Prod_price	number	null
Код дизайна	Des_id	number	Внешний ключ к таблице «Дизайны»

Таблица 14 - Схема отношения «Дизайны»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Des_id	number	Первичный ключ
Размеры	Des_size	number	null
Цвет	Des_color	string	null
Стиль	Des_style	string	null
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»

Таблица 15 - Схема отношения «Реклама»

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечание</i>
Код	Ad_id	number	Первичный ключ
Код работника	Wo_id	number	Внешний ключ к таблице «Работники»
Код услуги	Ser_id	number	Внешний ключ к таблице «Услуги»
Код товара	Prod_id	number	Внешний ключ к таблице «Товары»
Описание	Ad_descrip	string	null

### 3.5 Генерация кода

В программе ERwin имеется такая функция, как генерация кода. Она позволяет после построения ER-диаграммы базы сгенерировать код создания этой самой базы для многих СУБД. Это является одним из преимуществ ERwin-a, т.к. генерация кода намного упрощает работу по созданию базы данных.

На рисунках 21-24 показан процесс генерации кода.

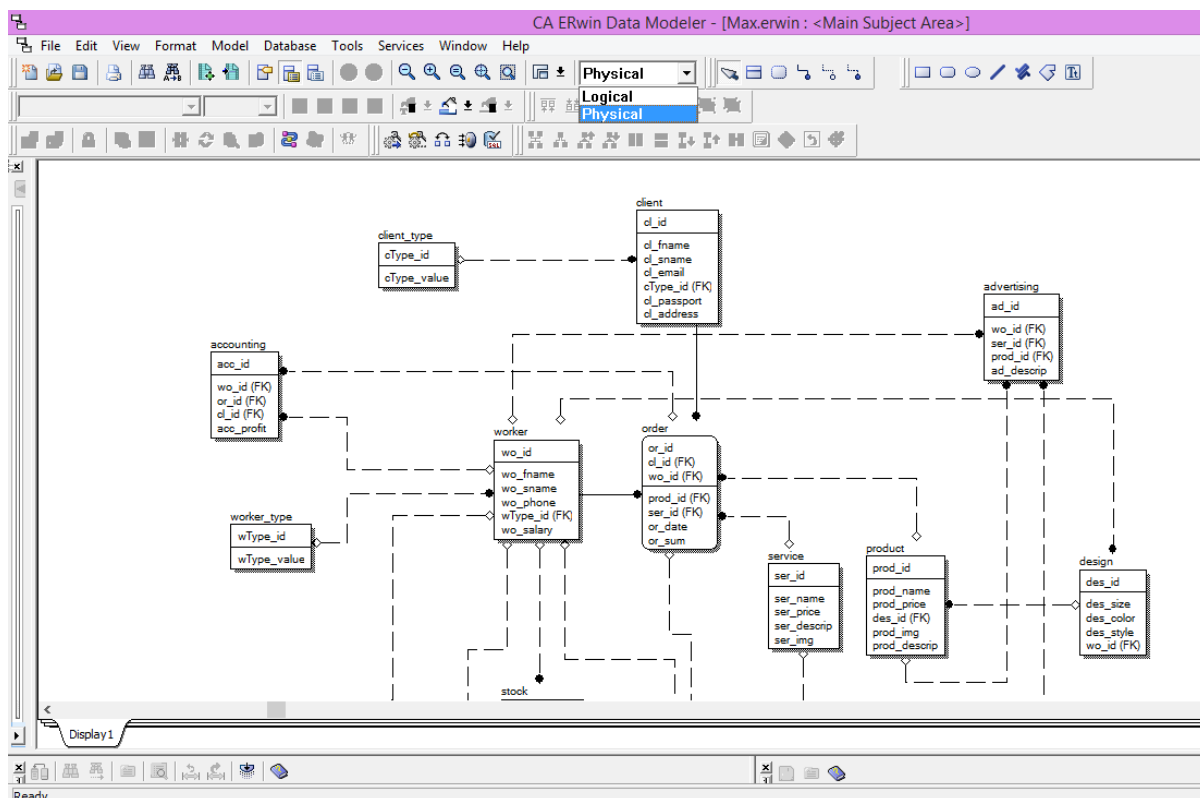


Рисунок 21 – Смена схемы с логического на физический

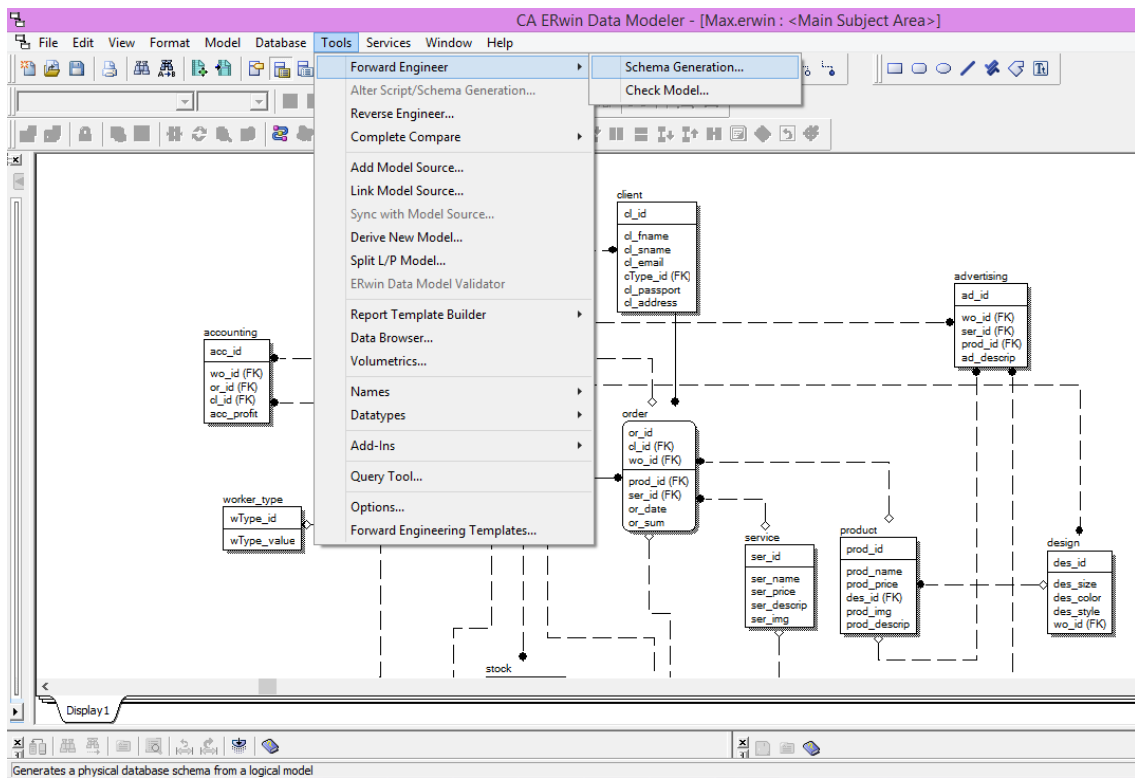


Рисунок 22 – Выбор «генерация схемы» в меню

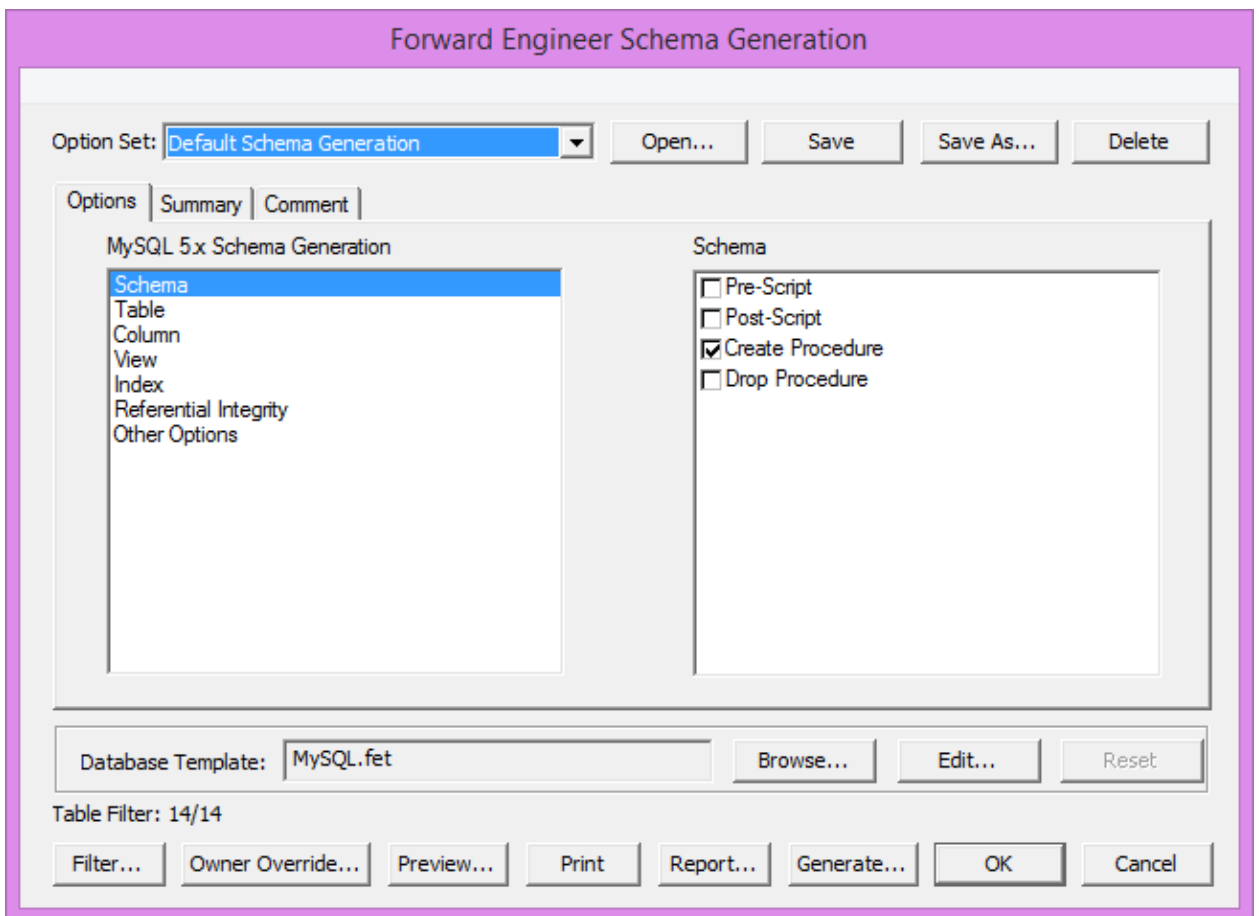


Рисунок 23 – Окно генерации схемы

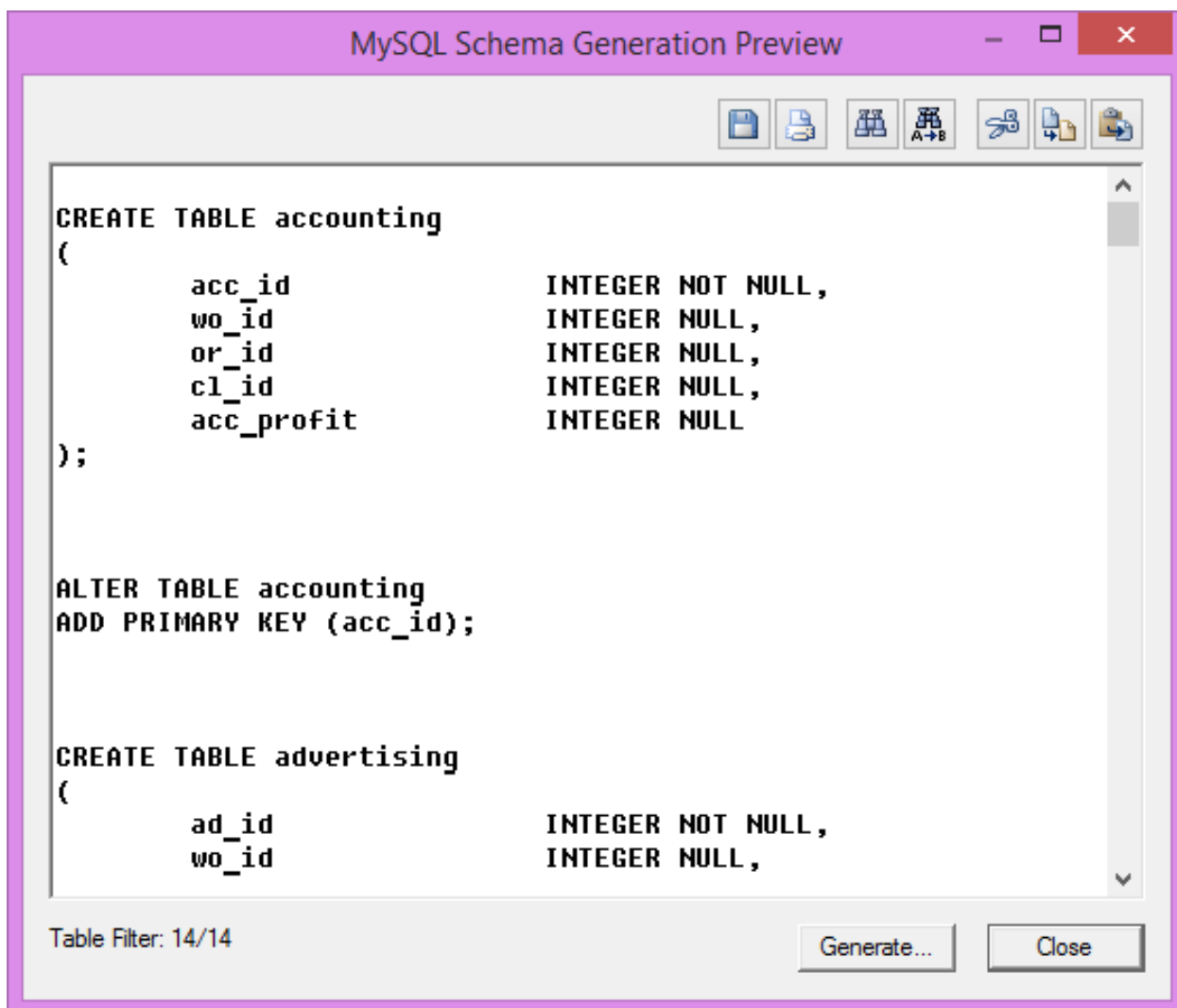


Рисунок 24 – Генерация кода для MySQL

### 3.6 Создание базы данных в phpMyAdmin

После настройки Xampp, включаем Apache и MySQL. Выполняем вход в phpMyAdmin через браузер. После этого создаем базу данных под названием tyrography (рисунок 25).

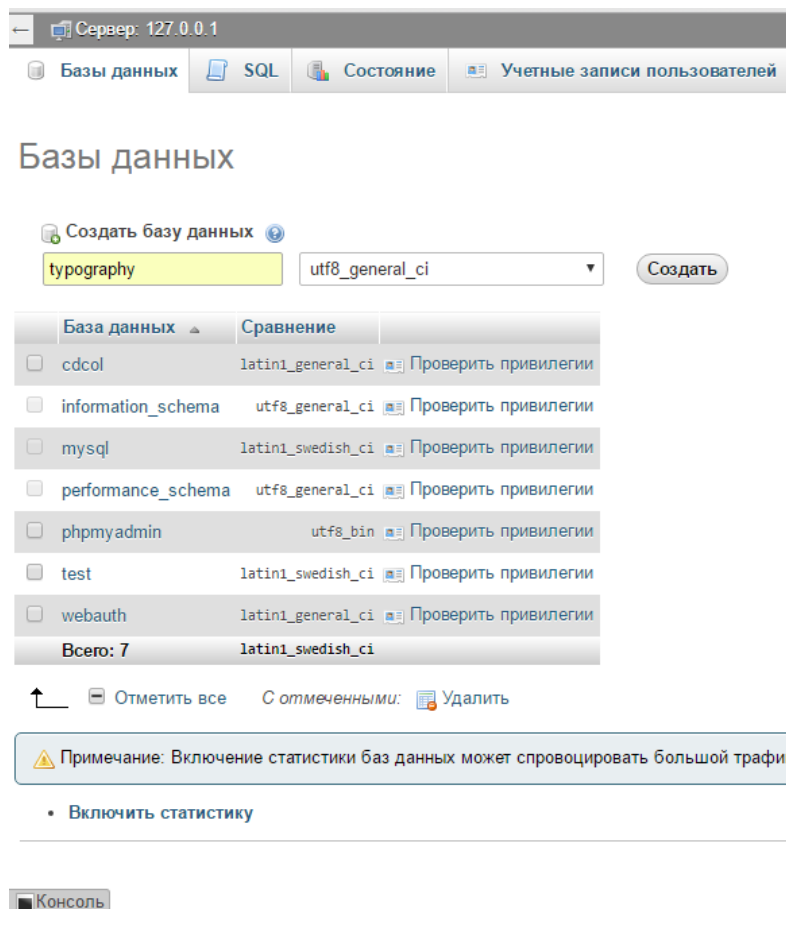
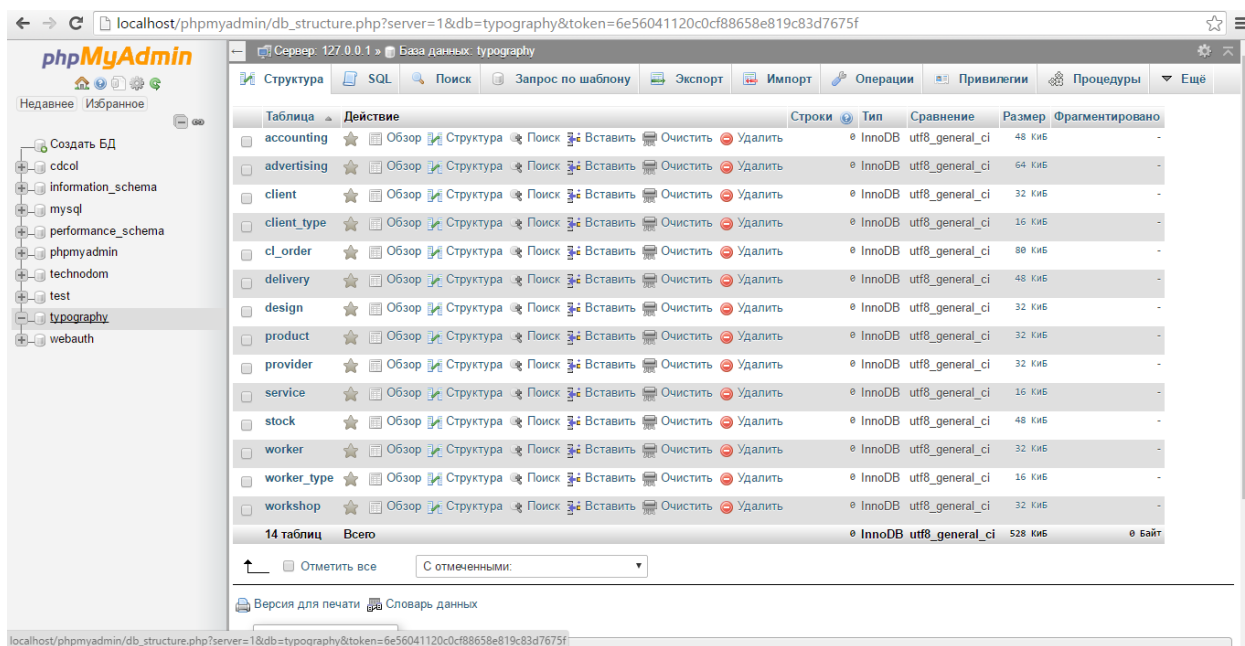


Рисунок 25 – Создание базы данных

После того, как мы создали базу, нужно создать таблицы. Для этого мы используем сгенерированный код. В итоге мы получаем базу, показанную на рисунке 26.



### 3.7 Заполнение базы данных в phpMyAdmin

Заполнение базы данных выполняется во вкладке «Вставить». Пример показан на рисунке 27.

Столбец	Тип	Функция	Null	Значение
wo_id	int(11)		<input type="checkbox"/>	12
wo_fname	varchar(40)		<input type="checkbox"/>	Allen
wo_sname	varchar(40)		<input type="checkbox"/>	Woody
wo_phone	varchar(40)		<input type="checkbox"/>	877774515
wType_id	int(11)		<input type="checkbox"/>	3
wo_salary	int(11)		<input type="checkbox"/>	500000

Рисунок 27 – Пример заполнения таблицы «Работники»

## 4. Разработка интерфейса

### 4.1 Описание интерфейса приложения для базы данных

Данная программа предназначена для упрощения работы с базой данных для типографии ИП «Адимов Ж.Ш.». Приложение связано с этой базой с помощью кода. После открытия приложения появляется окно авторизации (рисунок 28).

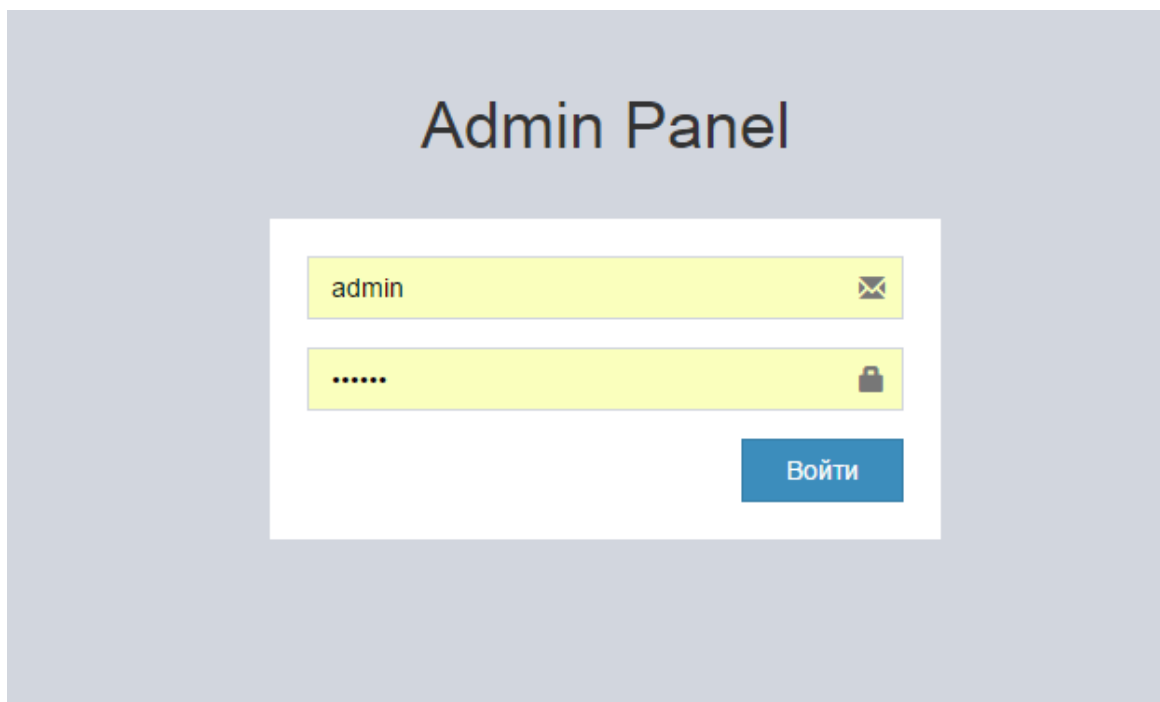


Рисунок 28 – Авторизация

В системе имеется 3 вида пользователя: администратор (admin), оператор (oper) и сотрудник (фамилия сотрудника). У каждого вида пользователя свои привилегии. Например, администратор имеет все права, оператор только редактирование, удаление, добавление полей, а сотрудник имеет право просматривать информацию только о себе.

Зайдем в систему под пользователем «оператор» для ознакомления (рисунок 29).

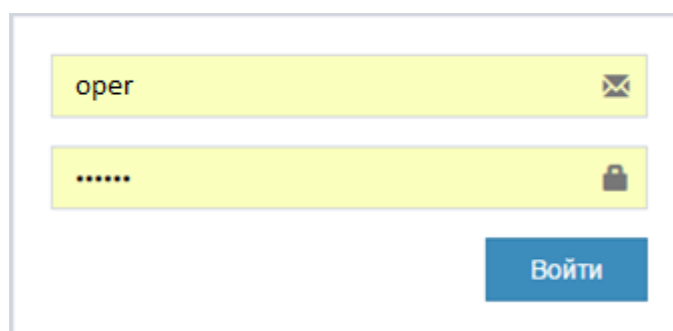


Рисунок 29 – Авторизация оператора

Открывается главное окно приложения (рисунок 30).



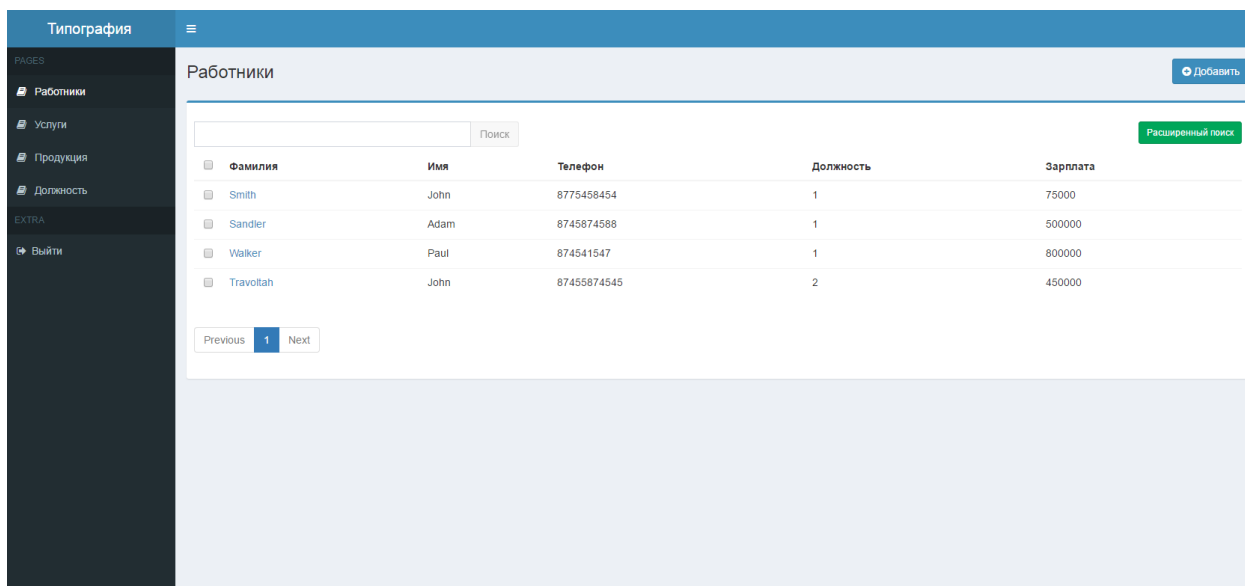


Рисунок 30 – Главное окно для оператора

В левой части представлен список таблиц нашей базы. При нажатии на каждую из них в центре приложения открывается выбранная таблица и значение полей в ней. Имеется поиск и расширенный поиск. Т.к. вход выполнен от имени оператора, имеется возможность добавления поля в любую таблицу (кнопка «Добавить»).

Нажав на кнопку «Выйти» приложение закрывается.

Теперь зайдем под именем какого-то сотрудника. Допустим, это будет Smith (рисунок 31).

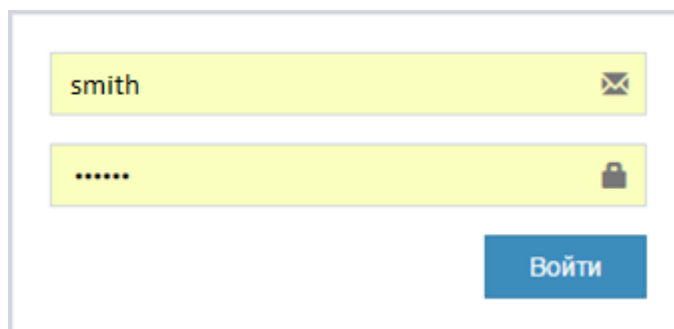


Рисунок 31 – Авторизация сотрудника

Открывается уже другой вид интерфейса (рисунок 32).

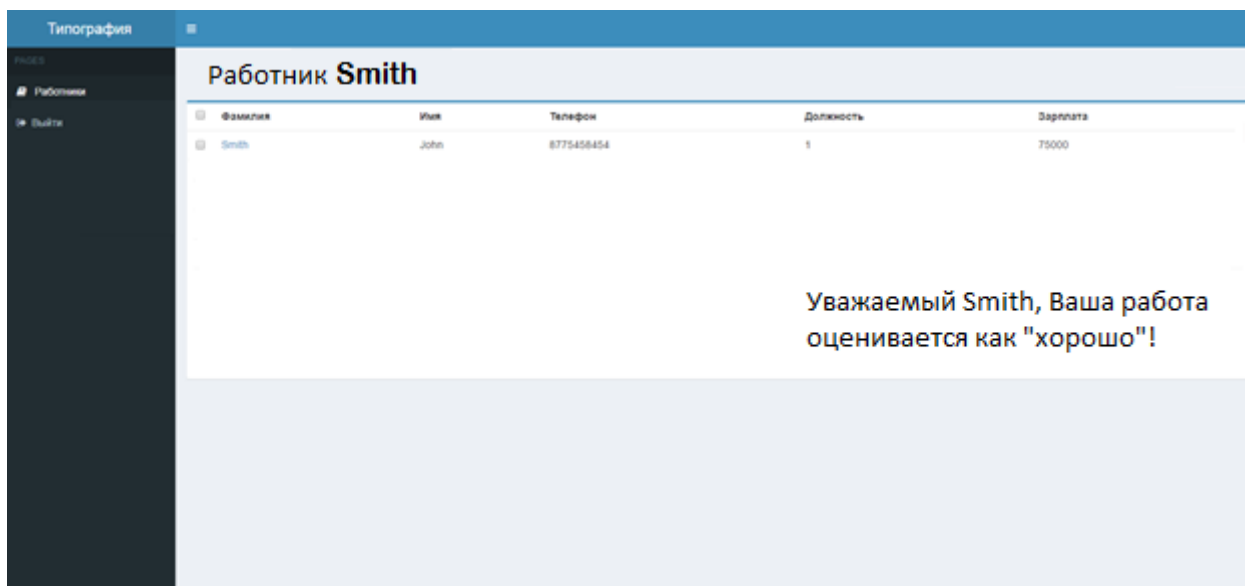


Рисунок 32 – Главное окно для сотрудника

Как мы видим, в правой части уже нет тех таблиц, что были у оператора. Нет и поиска, и функции добавление и редактирования полей. Есть только информация о самом сотруднике и об успешности его работы.

## 5. Технико-экономическое обоснование

### 5.1 Описание работы и обоснование необходимости

Тема этой работы заключается в разработке баз данных для типографии ИП «Адимов Ж.Ш.». Полиграфическая промышленность на сегодняшний день является одной из отраслей, постоянно растущей в связи с большим спросом со стороны базы потенциальных потребителей. База данных необходима для автоматизации процесса данных, управление записями клиентов, полиграфические услуги и продукты на предприятии.

### 5.2 Расчет затрат на разработку информационных технологий

Расчет общей стоимости решений по разработке проекта в виде информационно-коммуникационных технологий (СИО), считается согласно формуле

$$C_{ni} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + P_{ci} + P_{mi} + P_{\text{нки}} + P_{zi} + P_{ni}$$

где  $Z_{\text{фот}}$  – общий фонд оплаты труда разработчиков, тенге;

$Z_{\text{сзи}}$  – отчисления по социальному налогу, тенге;

$M_i$  – затраты на материалы, тенге;

$P_{ci}$  – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

$P_{mi}$  – затраты, связанные с эксплуатацией техники, тенге;

$P_{nki}$  – затраты на научные командировки, тенге;

$P_{zi}$  – прочие затраты, тенге;

$P_{ni}$  – накладные расходы, тенге.

Размер фонда заработной платы разработчиков ( $Z_{фот}$ ) рассчитывается следующим образом:

$$Z_{фот} = Z_{oi} + Z_{di}$$

где  $Z_{oi}$  – основная заработная плата, тенге;

$Z_{di}$  – дополнительная заработная плата, тенге.

Общая трудоемкость проекта рассчитывается следующим образом:

$$T_o = T_n * K_c * K_T * K_n$$

где  $K_c$  – коэффициент, учитывающий сложность ПО;

$K_T$  – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

$K_n$  – коэффициент, учитывающий степень новизны ПО;

$T_n$  – нормативная трудоемкость.

Вычисление коэффициента сложности проводится по таблице 1 и  $K_c = 0,26$ . В программном обеспечении присутствуют более чем 3 характеристик, обеспечение безопасного хранения, управления и поиска сложных структур данных, интерактивный доступ, функционирование программного обеспечения в расширенной операционной среде.

Таблица 16 – Дополнительные коэффициенты сложности ПО

Характеристика ПО	Значения $K_c$
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,08
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,07
4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г	
– 4.1 2 характеристики	0,12
– 4.2 3 характеристики	0,18
– 4.3 Свыше 3-х характеристик	0,26

Поправочный коэффициент, который учитывает уровень использования при разработке стандартных модулей ( $K_T$ ), определяется на основании данных, представленных в таблице 2 и составляет 0,8.

Таблица 17 – Значения поправочного коэффициента, учитывающего использование стандартных модулей типовых программ и ПО ( $K_T$ )

Степень охвата реализуемых функций разрабатываемого ПО стандартными модулями, типовыми программами и ПО	Значения $K_T$
1. От 60 % и выше	0,6
2. От 40 % до 60	0,7
3. От 20 % до 40 %	0,8
4. До 20 %	0,9
5. Типовые программы и ПО не используемые для реализации функций разрабатываемого ПО	1,0

Поправочный коэффициент, который учитывает новизну проекта ( $K_H$ ), определяется на основании данных находящихся в таблице 3 и составляет 0.7.

Таблица 18 – Поправочные коэффициенты, учитывающие новизну ПО ( $K_H$ )

Категория новизны	Степень новизны	Использование		Значение $K_H$
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	+	+	1,75
		–	+	1,6
		+	–	1,2
		–	–	1,0
Б	ПО, являющиеся развитием определенного параметрического ряда ПО	+	+	1,0
		–	–	0,9
		+	–	0,8
В	ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС	–	–	0,7

Основой для определения нормативной трудоемкости являются, нормы времени на разработку проекта, в зависимости от объема программного обеспечения и сложности группы. Нормативная трудоемкость проекта ( $T_H$ ) определяется на основе исходного объема и сложности ПП, подлежащего официальному утверждению, должно определяться с учетом сложности и новизны проекта и использование стандартных модулей в разработке. Учитывая информацию в приложении А: для первой категории сложности программного обеспечения

$$T_n = 520$$

Рассчитаем общий объем трудоемкости на основе формулы

$$T_o = 520 \cdot 0,26 \cdot 0,8 \cdot 0,7 = 76 \text{ чел./дн.}$$

На основе сложности определяется количеством планируемых разработчиков ( $Ч_p$ ) и время, необходимого для разработки проекта в целом ( $T_p$ ).

Можно решить следующие задачи:

- Подсчет количества исполнителей по условиям проектной документации;
- Время необходимое для разработки при заданном количестве исполнителей.

Численность исполнителей проекта ( $Ч_p$ ) подсчитывается по формуле:

$$Ч = T_o / (T_p * \Phi_{эф})$$

где  $\Phi_{эф}$  – эффективный фонд времени работы одного работника в течение года (дн.);

$T_o$  – общая трудоемкость разработки проекта (чел./дн.);

$T_p$  – срок разработки проекта (лет).

Срок разработки проекта ( $T_p$ ) определяется по формуле

$$T_p = T_o / (Ч_p * \Phi_{эф})$$

где  $Ч_p$  – плановое число разработчиков.

Эффективный фонд времени на одного работника ( $\Phi_{эф}$ ) рассчитывается следующим образом:

$$\Phi_{эф} = D_r - D_n - D_v - D_o$$

где  $D_r$  – количество дней в году;

$D_n$  – количество праздничных дней в году;

$D_v$  – количество выходных дней в году;

$D_o$  – количество дней отпуска.

Т.к., в соответствии с производственным календарем на 2016 год:

$D_r = 366$ ;

$D_n = 16$ ;

$D_v = 103$ ;

$D_o = 14$ , эффективный фонд времени одного работника составит

$$\Phi_{эф} = 366 - 16 - 103 - 14 = 233 \text{ дня}$$

Плановое число разработчиков  $Ч_p = 1$ , следовательно, по формуле

$$T_p = 76 / (1 * 233) = 0,33 = 121 \text{ день}$$

Таким образом, согласно произведенным расчетам и в соответствии с формулой

$$Ч = \frac{76}{0,33 \cdot 233} = 1 \text{ человек}$$

Основная заработная плата исполнителей на специальное ПО рассчитывается по следующей формуле:

$$З_{oi} = \sum_{i=1}^n T_{чи} \cdot T_{ч} \cdot К$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

$T_{чи}$  – часовая тарифная ставка i-го исполнителя (тыс.тенге);

$\Phi_n$  – плановый фонд рабочего времени i-го исполнителя (дней), 121 раб день;

$T_{ч}$  – количество часов работы в день (час), 8 часов;

К – коэффициент премирования, составляет 1,38.

По характеру и сложности функций сотрудников, участвующих в разработке ПО составляется штат сотрудников с определением образования, специальности, квалификации и должности (таблица 4).

Таблица 19 – Сведения по работникам, задействованным в проекте

Специалист– Исполнитель	Количество, человек	Зарботная плата в месяц, тенге
Программист	1	90000
Итого	1	90000

Средний дневной заработок каждого работника рассчитывается по следующей формуле:

$$D = \frac{З_m}{D_p}$$

где  $З_m$  – ежемесячный размер заработной платы;

$D_p$  – количество рабочих дней в месяце (это 21 дней – пятидневная рабочая неделя).

$$D = \frac{90000}{21} = 4285 \text{ тенге}$$

Часовая тарифная ставка рассчитывается путем деления месячной заработной платы, установленную при 40-часовой недельной норме рабочего времени и общего фонда времени ( $\Phi_p$ )

$$T_{ч} = T_m / \Phi_p$$

где  $T_{\text{ч}}$  – часовая тарифная ставка (тыс.тенге);  
 $T_{\text{м}}$  – месячная тарифная ставка (тыс.тенге).  
Общий фонд времени:

$$\Phi_{\text{р}} = T_{\text{ч}} * \Phi_{\text{п}}$$

Таким образом

$$\Phi_{\text{р}} = 8 * 21 = 168 \text{ час. в месяц}$$

Общее количество часов составит:

$$\Phi_{\text{р}} = 8 * 121 = 968 \text{ час.}$$

Рассчитаем тарифную ставку программиста

$$T_{\text{ч}} = \frac{90\,000}{168} = 535 \text{ тенге/час}$$

В соответствии с формулой основная заработная плата программиста составляет

$$З_{oi} = 520 * 8 * 121 * 1,38 = 694636,8 \text{ тенге}$$

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

$$З_{сзи} = (\text{ФОТ} - \text{ПО}) * 11\%$$

Где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются

$$\text{ПО} = \text{ФОТ} * 10\%$$

Таким образом

$$\text{ПО} = 694636,8 * 0,1 = 69463,6 \text{ тенге}$$

$$З_{сзи} = (694636,8 - 69463,6) * 0,11 = 68769 \text{ тенге}$$

Затраты на материалы определяются по формуле

$$M_i = (З_{\text{осн}} * N_{\text{мз}}) / 100\%$$

где  $N_{\text{мз}}$  – норма расхода материалов основной заработной платы (3–5%).

$$M_i = 694636,8 * 0,03 = 20839 \text{ тенге}$$

Расходы по спецоборудованию ( $P_{ci}$ ) включают стоимость средств, предназначенных для покупки специального программного обеспечения и вспомогательное оборудование требуемого для специальной разработки программного обеспечения, включая их расходы на производство, отладку, монтаж и эксплуатацию:



$$P_{ci} = \sum_{i=1}^n C_{ci}$$

Таблица 20 – Перечень оборудования, необходимого для разработки

Наименование изделий	Характеристика	Количество единиц	Цена за единицу, тенге	Общая сумма, тенге
Ноутбук	Lenovo g50 Intel Core –i7	1	130000	130000
Многофункциональное устройство	Canon MF-3010	1	30000	30000
Итого				160000

Таблица 21 – Программное обеспечение, используемое в работе

Программное обеспечение	Стоимость, тенге
MySQL	бесплатно
XAMPP ControlPanel	бесплатно
Erwin Data Modeler r7	бесплатно
Итого	0

Амортизационные отчисления по используемому оборудованию рассчитываются по следующей формуле:

$$A_i = \frac{N_A * C_{ПЕР} * N}{100 * n}$$

где  $N_A$  – норма амортизации;

$C_{ПЕР}$  – первоначальная стоимость оборудования;

$N$  – количество дней на выполнение работ;

$n$  – количество рабочих дней в году.

Таким образом

$$A_{об} = \frac{160000 * 25 * 121}{100 * 366} = 13224 \text{ тенге}$$

Расходы по статье « Прочие расходы »  $P_{zi}$  на специальное программное обеспечение, входящее в стоимость приобретения и подготовки специализированной научно-технической информации и специализированных издания . Определены на основе стандартов, разработанных в рамках всей организации , в процентах от основной заработной платы

$$P_{zi} = Z_{oi} * H_{пз}/100$$

Где  $H_{пз}$  – норматив прочих затрат в целом по организации (20%).  
Таким образом

$$П_{zi} = 694636,8 \cdot 0,2 = 138927,4 \text{ тенге}$$

Расходы на накладные расходы ( $P_{ни}$ ), будут основываться на критериях ( $H_{рн}$ ), рассчитанных как процент от основной зарплаты исполнителей .  
Стандарты устанавливаются по всей организации

$$P_{ни} = Z_{oi} * H_{рн} / 100\%$$

где  $P_{ни}$  – накладные расходы на конкретную ПО (тыс.тенге);

$H_{рн}$  – норматив накладных расходов в целом по организации (70%).

Таким образом накладные расходы составляют

$$P_{ни} = 694636,8 \cdot 0,7 = 486245,7 \text{ тенге}$$

Полные затраты на проектирование базы данных для типографии ИП «Адимов Ж.Ш.», составляют

$$C_{ni} = 694636,8 + 68769 + 20839 + 13224 + 138927,4 + 486245,7 = 1422641,9 \text{ тенге.}$$

Сводные результаты расчета затрат на разработку ПО и их структура представлены в таблице 22 и на рисунке 33.

Таблица 22 – Затраты на проектирование базы данных для типографии ИП «Адимов Ж.Ш.»

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{фот}$	694636,8	48,8
Социальный налог	$Z_{сзи}$	68769	4,8
Материалы	$M_i$	20839	1,5
Амортизационные отчисления	$A_{об}$	13224	0,9
Прочие затраты	$П_{zi}$	138927,4	9,7
Накладные расходы	$P_{ни}$	486245,7	34,1
Итого:		1422641,9	100

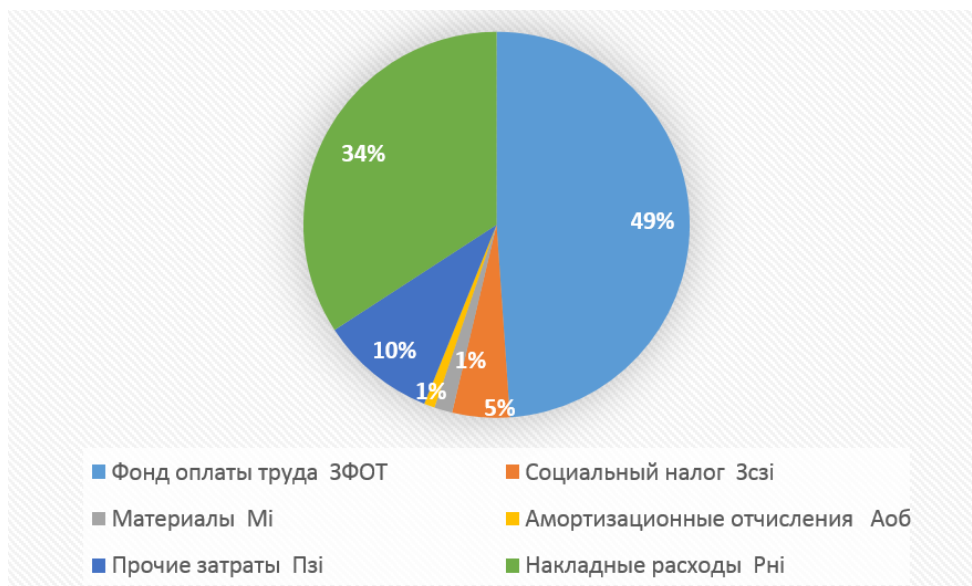


Рисунок 33 – Структура затрат на разработку ПП

### 5.3 Расчет цены программного продукта

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, рассчитывается согласно формуле

$$C_{\text{ПП}} = Z_{\text{РПР}} + \text{П}_{\text{П}} + \text{НДС}$$

где  $C_{\text{ПП}}$  – цена программного продукта, тенге;

$Z_{\text{РПР}}$  – затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$\text{П}_{\text{П}}$  – планируемая прибыль, тенге;

НДС – налог на добавленную стоимость, тенге.

Планируемая прибыль составляет (20%) от себестоимости разработки

$$\text{П}_{\text{П}} = Z_{\text{РПР}} \cdot 0,2$$

$$\text{П}_{\text{П}} = 1422641,9 \cdot 0,2 = 284528,4 \text{ тенге}$$

НДС, начисленный на  $\text{П}_{\text{П}}$ , определяется следующим образом

$$\text{НДС} = (\text{П}_{\text{П}} + Z_{\text{РПР}}) \cdot k_{\text{НДС}}$$

где  $k_{\text{НДС}}$  – ставка налога на добавленную стоимость.

Подставив данные в формулу получаем

$$\text{НДС} = (1422641,9 + 284528,4) \cdot 0,12 = 204860,4 \text{ тенге}$$

Цена реализации  $P_{\Pi}$  по формуле составляет

$$C_{\Pi\Pi} = 1422641,9 + 284528,4 + 204860,4 = 1912030,7 \text{ тенге}$$

#### **5.4 Оценка социально – экономических результатов**

Разработанная база данных выполняет следующие задачи:

- просмотр справочных данных;
- редактирование данных;
- способность быстро находить нужную информацию и отображать ее удобно;
- восстановление данных из базы данных;
- архивирование данных;
- возможность вывода формы отчетности;
- расчет и проверка реальной нагрузки работников;
- формирование электронной документации;
- экономия времени и затрат.

Представлен расчет затрат на разработку базы данных. База данных это более надежный способ хранения и обработки информации о сотрудниках, клиентах и процессах на предприятии.

### **6. Безопасность жизнедеятельности**

#### **6.1 Анализ условий труда**

В данной дипломной работе будет разрабатываться база данных типографии ИП «Адимов Ж.Ш.». База данных будет разрабатываться в офисном помещении и при работе используется ПК. Таким образом, мы проведем анализ потенциально опасных и вредных факторов, воздействующих на работника во время выполнения трудовой обязанности. В связи с этим необходимо создать благоприятные условия труда для разработчика, отвечающие требованиям техники безопасности.

При работе с персональным компьютером разработчик подвергается следующим вредным факторам, воздействующим на его здоровье:

- электромагнитные излучения;
- ионизация воздуха;
- плохая освещенность помещения;
- искривление осанки и ухудшение зрения при несоблюдении эргономических требований;
- несоблюдение техники пожарной и электрической безопасности;
- повышенный уровень шума;

- неблагоприятный микроклимат в помещении.

### 6.1.1 Пожаробезопасность

Здание относится к I степени огнестойкости: здания с несущими и ограждающими конструкциями из природных сооружений или искусственных материалов, бетона или железобетона с применением листовых и плиточных негорючих материалов;

### 6.1.2 Электромагнитное и ионизирующее излучения

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в табл. 23.

Максимальный уровень рентгеновского излучения на рабочем месте программиста компьютера обычно не превышает 10 мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10...100 мВт/м<sup>2</sup>.

Таблица 23. Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

Наименование параметра	Допустимые значения
Напряженность электрической составляющей электромагнитного	10 В/м
Напряженность магнитной составляющей электромагнитного	0,3 А/м
Напряженность электростатического поля не должна превышать:	20 кВ/м
для взрослых пользователей	
для детей дошкольных учреждений и учащихся средних специальных и высших учебных заведений	15 кВ/м

Для уменьшения влияния этого типа радиационных излучений желательно использовать мониторы с низким уровнем излучения (MPR - II , TCO - 92 , TCO - 99 ) рекомендуется установить защитные экраны , а также соблюдать регулярно режим работы и перерыва .

### 6.1.3 Освещение

Согласно СНиП РК 2.04-05-2002 (Естественное и искусственное освещение), относятся к категории работы 1а, производимые сидя и не

требующие физического нагрузки, при которых потребляемые расходы составляет до 120 кДж/ч; Помещение по визуальным условиям работы относится к IV разряду (размер различаемых объектов при работе от 1 до 10 мм и выше);

#### **6.1.4 Параметры микроклимата**

В соответствии со стандартами по охране труда ГОСТ 12.1.005-88. Лучшими и допустимыми нормами микроклимата, в зависимости от категории работы, работы в эталонном помещении для легкой тяжести (1а), который управляется дистанционным пультом на компьютере. Для того, чтобы создать стабильные условия для работы оператора, установлены нормы климата. Условия, которые необходимо соблюдать:

- В холодный период года:
  - оптимальная температура 22-24 °С, допустимая – 18-26 °С;
  - относительная влажность воздуха 40-60%, допустимая 75%.
- В теплый период года:
  - оптимальная температура 23-25 °С, допустимая 20-30 °С;
  - относительная влажность воздуха 40-60%, допустимая влажность 55%.

#### **6.1.5. Эргономические требования**

Создание рабочих мест базируется на стандартных эргономических требованиях. Работа конструкции мебели (столы и стулья) должны позволять индивидуальную настройку на рост и работу, чтобы создать удобную позу. Объекты частого использования труда и средств связи в пределах оптимальной деятельности.

Место для выполнения работы в сидячем состоянии отвечает требованиям. Конструкция его компонентов учитывать характер работы, психологические особенности личности и антропометрические данные.

Во время работы оператору запрещается:

- касаться одновременно экрана монитора и клавиатуры; прикасаться к задней панели системного блока при включенном питании;
- переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;
- загромождать верхние части устройств посторонними предметами;
- производить отключение питания во время выполнения активной задачи;
- производить частые переключения питания.

В процессе работы оператор находится в положении, показанном на рисунке 34.

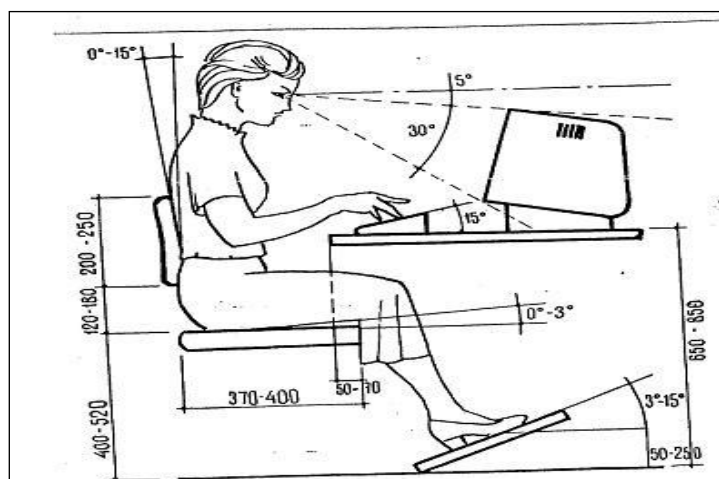


Рисунок 34 – Параметры рабочей мебели для оператора (в мм и градусах)

### 6.1.6 Рабочее помещение, в котором ведется разработка.

Помещение имеет следующие параметры:

- расположен на втором этаже двухэтажного здания;
- рабочее пространство: длина 5 м, ширина 3 м, высота 3,3 м;
- искусственное освещение – светильники: 2 светильника, в каждом по 2 люминесцентные лампы (ПВЛМ-1×40);
- помещение по зрительным условиям работы относится к IV разряду, т. к. наименьший объект различения от 1 до 10 мм;
- количество рабочих мест – 1.

План помещения показан на рисунке 35.

Для обеспечения нормальных условий труда санитарные нормы СН 245-71 устанавливают объём производственного помещения не менее 15 м<sup>3</sup>; Общая площадь помещения, выгороженного стенами или глухими перегородками, не менее 4,5 м<sup>2</sup>. В данном помещении имеется минимальная площадь, составляющая порядка 15 м<sup>2</sup>, а объём - порядка 49,5 м<sup>3</sup>, которые отвечают условиям санитарной нормы.

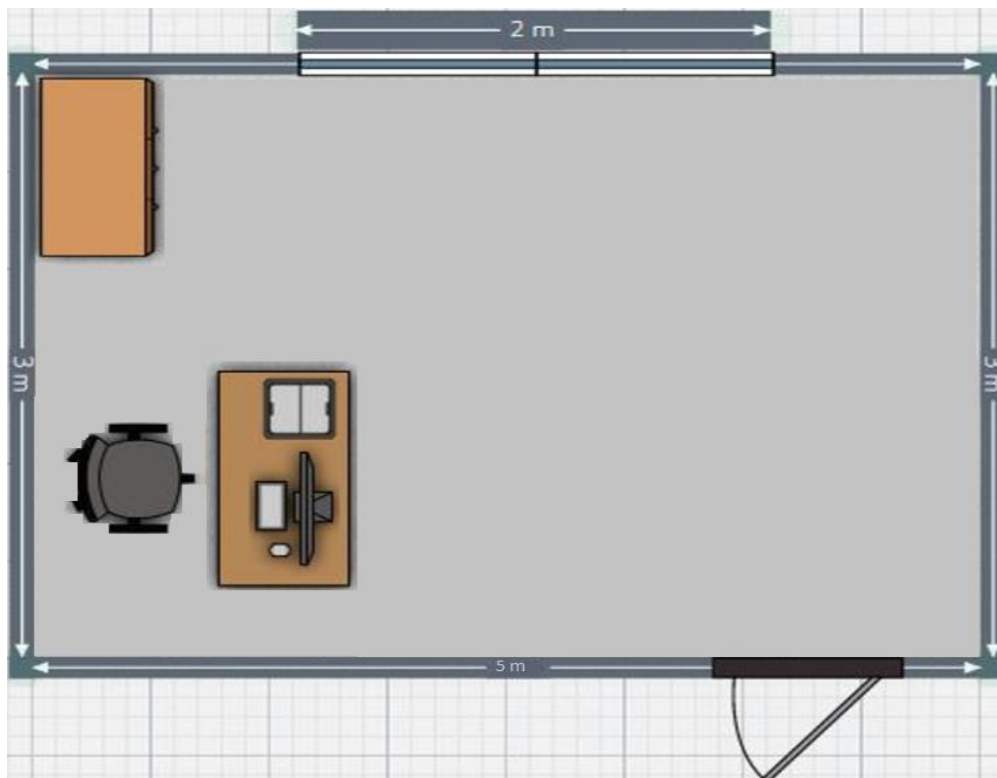


Рисунок 35 – План помещения

## 6.2 Технические решения обеспечения безопасности жизнедеятельности

### 6.2.1 Расчёт искусственного освещения методом коэффициента использования.

При разработке искусственного освещения помещения для определения площади световых проемов, обеспечивая нормальное значение в соответствии с необходимыми требованиями.

Разряд зрительной работы –IV.

Нормируемая освещённость - 400лк.

Используем систему общего освещения с люминесцентными лампами.

Вычислим высоту подвеса лампы над рабочей поверхностью

$$H = h - h_p - h_c$$

$h_c$  - расстояние от светильника до перекрытия,  $h_c=0,1$  м;

$h_p$  - высота рабочей поверхности над полом,  $h_p=0,7$  м;

$h$  - высота помещения,  $h=3,3$  м.



$$H = 3,3 - 0,7 - 0,1 = 2,5 \text{ м}$$

Наиболее выгодное расстояние между лампами определяется по формуле

$$L = \lambda * H$$

где  $\lambda = 1,2 \div 1,4$ . Возьмем  $\lambda = 1,3$

$$L = 1,3 * 2,5 = 3,25 \text{ м}$$

Расстояние от стены до ближайшей лампы, когда работа не выполняется рядом со стеной, определяется по формуле

$$li = (0,4 \div 0,5) * L$$

$$li = 0,45 * 3,25 = 1,46 \text{ м}$$

Определяем индекс помещения:

$$i = \frac{a * b}{H * (a + b)}$$

где  $a$  – длина помещения;

$b$  – ширина.

$$i = \frac{5 * 3}{2,5 * (5 + 3)}$$

Коэффициенты отражения от потолка, стен и пола соответственно равны:  $r_{\text{пот}} = 70\%$ ;  $r_{\text{ст}} = 50\%$ ;  $r_{\text{пол}} = 30\%$ .

Коэффициент использования в данном случае равен  $\eta = 40\%$ , коэффициент запаса равен  $k_3 = 1,2$ . Определим количество люминесцентных ламп по формуле

$$N = \frac{E * K_3 * S * Z}{n * \Phi_{\text{л}} * \eta}$$

где  $S$  – площадь помещения,  $S=15 \text{ м}^2$ ;

$K_3$ – коэффициент запаса,  $K_3=1,2$ ;

$E$  – заданная минимальная освещенность,  $E=400 \text{ лк}$ ;

$Z$  – коэффициент неравномерности освещения,  $Z=1,1$ ;  
 $n$  – количество ламп в светильнике,  $n=2$ ;  
 $\Phi_{л}$  – световой поток выбранной лампы,  $\Phi_{л}=3120$  лм;  
 $\eta$  – коэффициент использования,  $\eta=0,4$ .

$$N = \frac{400 * 1,2 * 15 * 1,1}{2 * 3120 * 0,4} = 3,17 \approx 3$$

Итого, чтобы создать нормальную освещенность 400 лк необходимо 3 светильника серии ПВЛМ с 2 лампами в светильнике, итого 6 люминесцентных ламп, мощность каждой лампы составляет не меньше 40 Вт, что не соответствует действительности, а значит имеющегося в наличии освещения не достаточно для удовлетворения санитарным нормам. Так как в помещении назначено всего 2 светильника, а это нарушение санитарных норм, необходимо дополнительно установить еще один светильник. В результате было реконструировано искусственное освещение, в итоге насчитывается 3 светильника серии ПВЛМ с 2 лампами в светильнике, что соответствует санитарным нормам.

### **6.2.2 Расчет искусственного освещения точечным методом**

Сделаем проверочный расчет точечным методом. Точечный метод расчета используется при применении точечных излучателях (лампы накаливания и ДРЛ). Давайте применим этот метод и

для люминесцентной лампы, что мы выбрали в качестве источника света в помещении. Световой поток лампы в каждом светильнике определяется: Разряд зрительной работы – V. Нормируемая освещённость – 400 лк. Высота подвеса светильников над освещаемой поверхностью  $H = 2,45$  м, коэффициент запаса равен (таблица 24). План освещённости показан на рисунке 36.

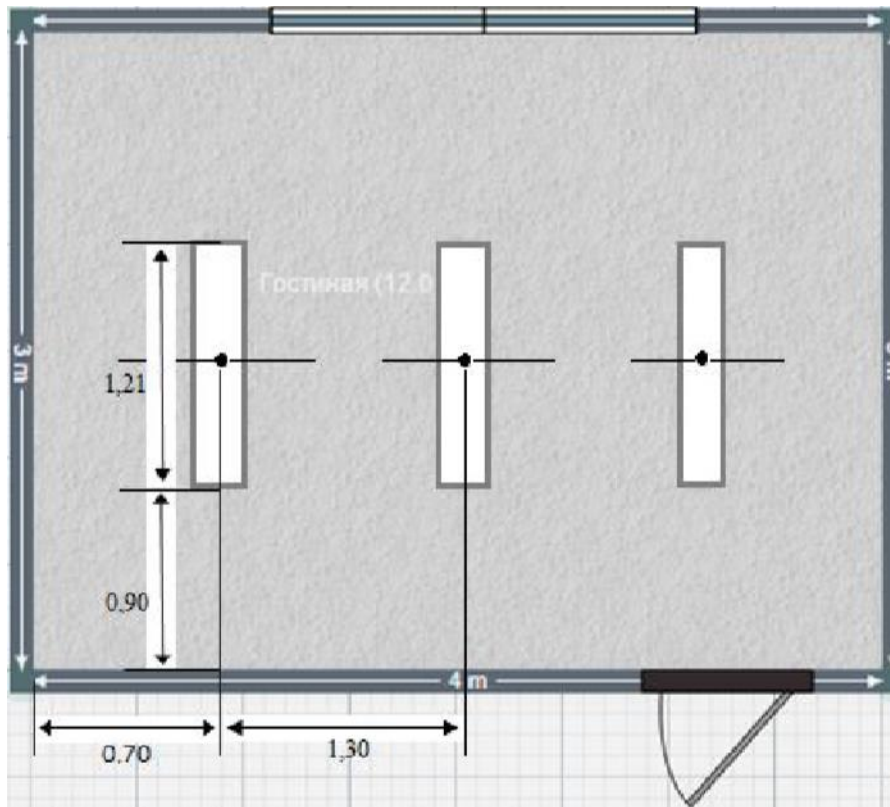


Рисунок 36 – Схема освещенности

Таблица 24 – Светораспределение светильников

Тип св-ка	Сила света $I_\alpha$ , кд в направлении угла $\alpha$										
			5	5	5	5	5	5	5	5	0
ПВЛМ- 1×10	39	35	32	15	04	4	3	4	2	-	0

Намечаем контрольную точку А. Для него, определяется сумма всех светильников после естественного освещения следующим образом.

Находим проекцию расстояния на потолок от точки А до светильника— d. Затем мы должны определить угол между потолком и прямой d. По этому углу находим условную освещенность. Пусть контрольная точка будет находится под лампой № 2. Обозначим  $d_i$  – расстояние от контрольной точки до i–ой лампы

$$\alpha = \arctg\left(\frac{d_i}{H}\right),$$

$$e_i = \frac{I_\alpha \cdot \cos^3(\alpha)}{H^2}$$

1–я лампа  $d_1 = 0$  м  
 $\alpha = 0^\circ$ ;

$$ei = \frac{139 \cdot 2 \cdot \cos^3(0)}{2.25^2} = 54,9 \text{ лк}$$

1, 3-я лампы  $d1 = d3 = 1,3 \text{ м}$ ;  
 $\alpha = 30^\circ$ .

$$ei = \frac{110 \cdot 2 \cdot \cos^3(30)}{2.25^2} = 28,22 \text{ лк}$$

Суммарная условная освещенность равна

$$\Sigma E = 54,9 + 2 \cdot 28,22 = 111,34 \text{ лк}$$

Суммарная освещенность равна

$$E_{AG} = \frac{\mu \cdot F_F}{1000 \cdot K_z} \cdot \Sigma E,$$

$$E_{AG} = \frac{1.1 \cdot 2 \cdot 3120}{1000 \cdot 1.2} \cdot 111.34 \approx 640 \text{ лк}$$

В результате проверки мы убедились, что искусственного освещения вполне достаточно.

### 6.2.3 Расчет уровня шума

Шум ухудшает рабочую среду, обеспечивая вредное воздействие на организм человека. Работа в таких условиях длительного воздействия шума приводят к раздражительности, головной боли, головокружение, потеря памяти, усталость, потеря аппетита, боли в ушах, и так далее. Такие нарушения в ряде органов и систем организма могут привести к неблагоприятным изменениям в эмоциональном состоянии, может привести к стрессу. Под влиянием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость из-за увеличения затрат энергии и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума (выше 80 дБ (А)) на слух человека приводит к его частичной или полной потере.

Основным источником шума является системный блок ПК. Шум издают такие механические устройства, как жесткий диск, CD/DVD приводы, а также кулеры (вентиляторы). Принтер и сканер тоже относятся к источникам шума, но ввиду того что они используются крайне редко, рассматриваться не будут.

Жесткий диск. В системе используется один жесткий диск, со скоростью вращения шпинделя 7200 об/мин. Согласно паспорту максимальный уровень шума не превышает 25 дБ при бездействии, и 32 дБ во время дисковой активности (чтение/запись/поиск).

CD/DVD приводы. В системе установлен один привод - DVD-RW. Максимальный шум из него при максимальных оборотах вращения шпинделя не должен превышать 40-45 дБ, в настоящее время данные приводы используются крайне редко.

Системы активного охлаждения. Для обеспечения циркуляции воздуха внутри системного блока и качественного охлаждения компонентов системы, в корпусе насчитывается 6 кулеров разных размеров и скоростей вращения, более подробное описание ниже:

1) Кулер на центральном процессоре - 1шт., Ø 60мм, 2700об/мин, 22дБА.

2) Кулер на северном мосту материнской платы - 1шт., Ø 50мм, 4500об/мин, 23дБА.

3) Кулер графического ядра видеокарты - 1шт., Ø 40мм, уровень шума и скорость вращения не известны. Но шум данного кулера практически не различим.

4) Кулер блока питания - 1шт., Ø 80мм, 4000об/мин, 33дБА.

5) Кулеры, расположенные на корпусе - 2шт., 80мм, 2500об/мин, 28дБА.

$$L_{\Sigma} = 10 \lg \sum_{i=1}^{i=n} 10^{0,1L_i} \quad (6.7)$$

где  $L_i$  – уровень звукового давления  $i$ -го источника шума;

$n$  – количество источников шума.

Суммарный шум будет равен

$$L_{\text{сум}} = 10 * \lg(10^{32/10} + 10^{22/10} + 10^{23/10} + 10^{33/10} + 10^{28/10}) = 36,6 \text{дБА}$$

## **Заключение**

В ходе выполнения дипломной работы была спроектирована база данных для типографии ИП «Адимов Ж.Ш.», а также разработан интерфейс для удобного пользования базой. В качестве СУДБ был выбран MySQL, а администрирование базы данных велось на phpMyAdmin. Подключение приложения к базе данных происходило без особых проблем.

Программный продукт предназначен для автоматизации работы на предприятии, улучшения качества обслуживания, ведение финансовых отчетов и бухгалтерии.

Спроектированное ПО было внедрено в типографию ИП «Адимов Ж.Ш.». Предприятие успешно пользуется разработанным мной продуктом.

В технико-экономической части произведен расчет затрат на проектирование базы данных. Итоговая сумма затрат составила 1 422 641,9 тенге. Также был проведен расчет стоимости программного продукта, цена реализации составляет 1 912 030,7 тенге. По итогам проведенных расчетов программный продукт является экономически выгодным.

При выполнении данной дипломной работы был проведен анализ условия труда, расчет искусственного освещения методом коэффициента использования и точечным методом, расчет уровня шума.

## Список литературы

1. Белов С.В., Ильницкая А.В., Козьяков А.Ф. и др. Безопасность жизнедеятельности. Учебник для вузов. – М.: Высш.шк., 2005. – 448 с.
2. Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эверо, 2007. – 274с.
3. Байзакова А.А., Санатова Т.С. Охрана труда: Методические указания к выполнению расчетно-графических работ. – 2015
4. Самгин Э.Б. Освещение рабочих мест. – М.: МИРЭА, 1989. – 186с.
5. Мананбаева С.Е. Производственное освещение. Методические указания к выполнению раздела «Безопасность жизнедеятельности» в выпускной работе.- Алматы: АИЭС. 2004г. – 312 с.
6. СНиП РК 2.04-05-2002, Естественное и искусственное освещение. Общие требования. Комитет по делам строительства Министерства индустрии РК. - Астана, 2002г. – 512 с
7. Банк В.Р.,Зверев В.С. Информационные системы в экономике. - М.:Экономистъ, 2005. - 480 с.
8. Носова С.С. Экономическая теория. - М.: Кнорус, 2010 (Электронная версия).
9. Айдарханова М. Основы экономической теории. - Алматы: Фолиант,2010.
10. Веллинг Л. PHP и MySQL Web-разработчикам. - М.: SAMS, 2001. – 893с.
11. Крамер Д. Электронный учебник «HTML: наглядный курс Webдизайна». - СПб.: ИКС, 1999. - 238 с.
12. <http://www.php.net>
13. <http://www.mysql.com>

## Приложение А

```
CREATE TABLE client_type
```

```
(  
    cType_id      INTEGER NOT NULL PRIMARY KEY,  
    cType_value   VARCHAR(40) NULL  
);
```

```
CREATE TABLE client
```

```
(  
    cl_id         INTEGER NOT NULL PRIMARY KEY,  
    cl_fname     VARCHAR(40) NULL,  
    cl_sname     VARCHAR(40) NULL,  
    cl_email     VARCHAR(40) NULL,  
    cType_id     INTEGER NULL,  
    cl_passport  INTEGER NULL,  
    cl_address   VARCHAR(50) NULL,  
    FOREIGN KEY R_1 (cType_id) REFERENCES client_type (cType_id)  
);
```

```
CREATE TABLE worker_type
```

```
(  
    wType_id     INTEGER NOT NULL PRIMARY KEY,  
    wType_value  VARCHAR(40) NULL  
);
```

```
CREATE TABLE worker
```

```
(  
    wo_id        INTEGER NOT NULL PRIMARY KEY,  
    wo_fname     VARCHAR(40) NULL,  
    wo_sname     VARCHAR(40) NULL,  
    wo_phone     INTEGER NULL,  
    wType_id     INTEGER NULL,  
    wo_salary    INTEGER NULL,  
    FOREIGN KEY R_2 (wType_id) REFERENCES worker_type (wType_id)  
);
```

```
CREATE TABLE service
```

```
(  
    ser_id       INTEGER NOT NULL PRIMARY KEY,  
    ser_name     VARCHAR(40) NULL,  
    ser_descrip  VARCHAR(40) NULL,  
    ser_img      VARCHAR(40) NULL,  
    ser_price    INTEGER NULL  
);
```

```
CREATE TABLE design
```



```

(
    des_id          INTEGER NOT NULL PRIMARY KEY,
    des_size        VARCHAR(40) NULL,
    des_color       VARCHAR(40) NULL,
    des_style       VARCHAR(40) NULL,
    wo_id           INTEGER NULL,
    FOREIGN KEY R_3 (wo_id) REFERENCES worker (wo_id)
);
CREATE TABLE product
(
    prod_id         INTEGER NOT NULL PRIMARY KEY,
    prod_name       VARCHAR(40) NULL,
    prod_descrip   VARCHAR(40) NULL,
    des_id          INTEGER NULL,
    prod_img        VARCHAR(40) NULL,
    prod_price      INTEGER NULL,
    FOREIGN KEY R_4 (des_id) REFERENCES design (des_id)
);
CREATE TABLE cl_order
(
    or_id           INTEGER NOT NULL,
    cl_id           INTEGER NOT NULL,
    wo_id           INTEGER NOT NULL,
    prod_id         INTEGER NULL,
    ser_id          INTEGER NULL,
    or_date         DATE NULL,
    or_sum          INTEGER NULL,
    PRIMARY KEY (or_id, cl_id, wo_id),
    FOREIGN KEY R_5 (cl_id) REFERENCES client (cl_id),
    FOREIGN KEY R_6 (wo_id) REFERENCES worker (wo_id),
    FOREIGN KEY R_7 (prod_id) REFERENCES product (prod_id),
    FOREIGN KEY R_8 (ser_id) REFERENCES service (ser_id)
);
CREATE TABLE advertising
(
    ad_id           INTEGER NOT NULL PRIMARY KEY,
    ser_id          INTEGER NULL,
    prod_id         INTEGER NULL,
    ad_descrip     VARCHAR(40) NULL,
    wo_id           INTEGER NULL,
    FOREIGN KEY R_9 (wo_id) REFERENCES worker (wo_id),
    FOREIGN KEY R_10 (ser_id) REFERENCES service (ser_id),
    FOREIGN KEY R_11 (prod_id) REFERENCES product (prod_id)
);

```

);

CREATE TABLE workshop

```
(
    wsh_id          INTEGER NOT NULL PRIMARY KEY,
    wo_id           INTEGER NULL,
    wsh_machine     INTEGER NULL,
    wsh_department  VARCHAR(40) NULL,
    FOREIGN KEY R_12 (wo_id) REFERENCES worker (wo_id)
);
```

CREATE TABLE provider

```
(
    pro_id          INTEGER NOT NULL PRIMARY KEY,
    pro_mat_name    VARCHAR(40) NULL,
    wo_id           INTEGER NULL,
    FOREIGN KEY R_13 (wo_id) REFERENCES worker (wo_id)
);
```

CREATE TABLE stock

```
(
    st_id           INTEGER NOT NULL PRIMARY KEY,
    pro_id          INTEGER NULL,
    wo_id           INTEGER NULL,
    st_mat_name     VARCHAR(40) NULL,
    FOREIGN KEY R_14 (wo_id) REFERENCES worker (wo_id),
    FOREIGN KEY R_15 (pro_id) REFERENCES provider (pro_id)
);
```

CREATE TABLE delivery

```
(
    del_id          INTEGER NOT NULL PRIMARY KEY,
    wo_id           INTEGER NULL,
    or_id           INTEGER NULL,
    cl_id           INTEGER NULL,
    FOREIGN KEY R_16 (wo_id) REFERENCES worker (wo_id),
    FOREIGN KEY R_17 (or_id, cl_id, wo_id) REFERENCES cl_order (or_id,
cl_id, wo_id)
);
```

CREATE TABLE accounting

```
(
    acc_id          INTEGER NOT NULL PRIMARY KEY,
    wo_id           INTEGER NULL,
```

```

        or_id          INTEGER NULL,
        cl_id          INTEGER NULL,
        acc_profit     INTEGER NULL,
        FOREIGN KEY R_18 (wo_id) REFERENCES worker (wo_id),
        FOREIGN KEY R_19 (or_id, cl_id, wo_id) REFERENCES cl_order (or_id,
cl_id, wo_id)
);

```

## Приложение Б

```
<?php
```

```

namespace CrudKit\Controllers;

use CrudKit\CrudKitApp;
use CrudKit\Pages\BasePage;
use CrudKit\Util\FlashBag;
use CrudKit\Util\TwigUtil;
use CrudKit\Util\ValueBag;
use CrudKit\Util\RouteGenerator;
use CrudKit\Util\UrlHelper;
use Exception;

class BaseController {

    /**
     * @var UrlHelper
     */
    protected $url = null;

    /**
     * @var CrudKitApp
     */
    protected $app = null;

    /**
     * @var TwigUtil
     */
    protected $twig = null;

    /**
     * @var BasePage
     */
    protected $page = null;

    /**

```

```

* Route generator
* @var RouteGenerator
*/
protected $routeGen = null;

/**
* @param $app CrudKitApp
*/
public function __construct ($app) {
    $this->app = $app;
    $this->url = new UrlHelper();
    $this->routeGen = new RouteGenerator();
    $this->twig = new TwigUtil();
}

public function handle () {
    $action = $this->url->get("action", "default");

    $whoops = new \Whoops\Run();
    if($this->url->get("ajax", false)) {
        $whoops->pushHandler (new \Whoops\Handler\JsonResponseHandler());
    }
    else {
        $whoops->pushHandler (new \Whoops\Handler\PrettyPageHandler());
    }
    $whoops->register();
    $result = null;
    if(method_exists($this, "handle_".$action)) {
        $result = call_user_func(array($this, "handle_". $action));
    }
    else {
        throw new Exception ("Unknown action");
    }
    $output = "";

    if(is_string($result)) {
        $newResult = array(
            'type' => 'transclude',
            'content' => $result
        );
        $result = $newResult;
    }
}

```

```

switch($result['type']) {
    case "template":
        $output = $this->twig->renderTemplateToString($result['template'],
$result['data']);
        break;
    case "json":
        $this->app->setJsonResponse(true);
        $data = $result['data'];
        $data['flashbag'] = FlashBag::getFlashes();
        $output = json_encode($data);
        break;
    case "redirect":
        $this->app->_requestRedirect ($result['url']);
        return;
        break;
    case "transclude":
        $pageMap = [];
        /** @var BasePage $pageItem */
        foreach($this->app->getPages() as $pageItem) {
            $pageMap []= array(
                'id' => $pageItem->getId(),
                'name' => $pageItem->getName()
            );
        }
        ValueBag::set("flashbag", FlashBag::getFlashes());
        $data = array(
            'valueBag' => json_encode(ValueBag::getValues()),
            'staticRoot' => $this->app->getStaticRoot(),
            'pageMap' => $pageMap,
            'defaultUrl' => $this->routeGen->defaultRoute(),
            'title' => $this->app->getAppName (),
            'userParams' => $this->app->getUserParams (),
            'pageTitle' => "
        );
        if($this->page !== null) {
            $data['page'] = $this->page;
            $data['currentId'] = $this->page->getId();
            $data['pageTitle'] = $this->page->getName();
        }
        else {
            $data['currentId'] = -1;
        }
        $data['page_content'] = $result['content'];
        $data['dev'] = false; // change to true to load unminified js

```

```

        $output = $this->twig->renderTemplateToString("main_page.twig",
$data);
        break;
    default:
        throw new Exception ("Unknown result type");
    }

    return $output;

}
}
<?php

```

```

require "vendor/autoload.php";
use CrudKit\CrudKitApp;
use CrudKit\Pages\MySQLTablePage;
use CrudKit\Pages\BasicLoginPage;

```

```

// Create a new CrudKitApp object
$app = new CrudKitApp ();
$app->setStaticRoot ("static/crudkit/");
$app->setAppName ("Admin Panel");

```

```

//
// HANDLE LOGIN
//
$login = new BasicLoginPage ();
$login->setWelcomeMessage ("Use credentials admin/demo or user/demo");
if ($login->userTriedLogin ()) {
    $username = $login->getUserName ();
    $password = $login->getPassword ();

```

```

// TODO: you should use your own authentication scheme here
if ($username === 'admin' && $password === 'demo') {
    $login->success();
}
else if ($username === 'user' && $password === 'demo') {
    $login->success();
}
else {
    $login->fail ("Please check your password (admin/demo) or (user/demo)");
}
}

```

```

$app->useLogin ($login);
//
// END HANDLING LOGIN.
//

if ($login->getLoggedInUser () === "user") {
    // If the user isn't `admin` then use read-only
    $app->setReadOnly (true);
}

// Create a new page mapped to a table
$workerPage = new MySQLTablePage ("workers_page", "root", "",
"typography");
$workerPage->setName("Работники")
    ->setTableName ("worker") // Set the table name
    ->setPrimaryColumn ("wo_id","ID_Работника")
    ->addColumn ("wo_fname", "Фамилия")
    ->addColumn ("wo_sname", "Имя")
    ->addColumn ("wo_phone", "Телефон")
    ->addColumn ("wType_id", "Должность")
    ->addColumn ("wo_salary", "Зарплата", [ // Additional configuration using
options
    'validation' => [
        'required' => false,
    ]
    ])
    ->setSummaryColumns (["wo_fname",
"wo_sname","wo_phone","wType_id","wo_salary"]);

$app->addPage($workerPage);

$servicePage = new MySQLTablePage ("service_page", "root", "", "typography");
$servicePage->setName("Услуги")
    ->setTableName ("service") // Set the table name
    ->setPrimaryColumn ("ser_id","ID_Услуги")
    ->addColumn ("ser_name", "Название услуги")
    ->addColumn ("ser_descrip", "Описание услуги")
    ->addColumn ("ser_price", "Цена")
    ->setSummaryColumns (["ser_name", "ser_descrip","ser_price"]);

// Add the page to the app
$app->addPage($servicePage);

```

```

$productPage = new MySQLTablePage ("product_page", "root", "",
"typography");
$productPage->setName("Продукция")
    ->setTableName ("product") // Set the table name
    ->setPrimaryColumn ("prod_id","ID_Продукции")
    ->addColumn ("prod_name", "Название товара")
    ->addColumn ("prod_descrip", "Описание товара")
    ->addColumn ("prod_price", "Цена")
    ->setSummaryColumns (["prod_name", "prod_descrip","prod_price"]);

// Add the page to the app
$app->addPage($productPage);

$wtypePage = new MySQLTablePage ("wtype_page", "root", "", "typography");
$wtypePage->setName("Должность")
    ->setTableName ("worker_type") // Set the table name
    ->setPrimaryColumn ("wType_id","ID_должности")
    ->addColumn ("wType_value", "Название Должности")
    ->setSummaryColumns (["wType_id","wType_value"]);

// Add the page to the app
$app->addPage($wtypePage);

// Render the app. This will display the HTML
$app->render ();
?>
<?php

namespace CrudKit\Controllers;

use CrudKit\Util\FlashBag;
use CrudKit\Util\ValueBag;

class MainController extends BaseController {
    public function handle_default () {
        $firstPage = $this->app->getDefaultPage ();
        if($firstPage !== null) {
            return array(
                'type' => 'redirect',
                'url' => $this->routeGen->openPage ($firstPage->getId())
            );
        }
    }
}

```



```

    );
}
else {
    return "";
}
}
}
public function handle_view_page () {
    // Handle the view page action
    $pageId = $this->url->get('page');
    $this->page = $this->app->getPageById($pageId);
    $this->page->init($this->app);
    ValueBag::set("pageId", $this->page->getId());

    return $this->page->render();
}

public function handle_page_function () {
    $pageId = $this->url->get('page');
    $this->page = $this->app->getPageById($pageId);
    $this->page->init($this->app);
    $func = $this->url->get("func");
    ValueBag::set("pageId", $this->page->getId());

    if(method_exists($this->page, "handle_".$func))
    {
        return call_user_func(array($this->page, "handle_".$func));
    }
    else {
        throw new \Exception("Unknown method");
    }
}
}
}

```

### Приложение В

Укрупненные нормы времени на разработку ПО в зависимости от уточненного объема ПО и группы сложности ПО

Таблица В.1 – Укрупненные нормы времени на разработку ПО ( $T_H$ ) в зависимости от уточненного объема ПО ( $V_y$ ) и группы сложности ПО (чел./дн.)

Объем ПО(строки исходного кода, LOC)	Категории сложности ПО			Категории сложности ПО
	1-я	2-я	3-я	

200	—	—	21	1
300	—	—	23	2
400	—	—	25	3
500	—	—	27	4
600	—	33	28	5
700	—	36	30	6
800	—	38	32	7
900	—	40	34	8
1000	51	43	36	9
1200	54	45	38	10
1400	57	48	40	11
1600	60	50	42	12
1800	64	54	45	13
2000	68	57	48	14
2200	73	61	51	15
2400	76	64	54	16
2600	81	68	57	17
2800	86	72	60	18
3000	91	76	64	19
3200	97	81	68	20
3400	103	86	72	21
3600	110	92	77	22
3800	117	98	82	23
4000	124	104	87	24

4200	133	111	93	25
4400	141	118	99	26
4600	151	126	105	27
4800	160	134	112	28
5000	170	142	119	29
5500	182	152	127	30
6000	194	162	135	31
6500	206	172	144	32
7000	220	184	154	33
7500	235	196	164	34
8000	252	210	175	35
8500	268	224	187	36
9000	288	240	200	37
9500	307	256	214	38
10000	327	273	228	39
11000	349	291	243	40
12000	374	312	260	41
13000	399	333	278	42
14000	427	356	297	43
15000	456	380	317	44
16000	487	406	339	45
18000	520	434	362	46
20000	556	464	387	47
22000	595	496	414	48

24000	636	530	442	49
26000	679	566	472	50
28000	727	606	505	51
30000	775	646	540	52
32000	830	692	577	53
34000	888	740	617	54
36000	950	792	660	55
38000	1016	847	706	56
40000	1087	906	755	57
42000	1161	968	807	58
44000	1242	1035	863	59
46000	1328	1107	923	60
48000	1420	1184	987	61
50000	1620	1267	1056	62