МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
 ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ ЭКОНОМИКИ И МЕНЕДЖМЕНТА  
КАФЕДРА БИЗНЕС-ИНФОРМАТИКИ

**Проектирование базы данных "Турфирма"**

Пояснительная записка

Курсовой проект по дисциплине  
«База данных»

ТПЖА 230700.182 ПЗ

ТПЖА 230700.380 ПЗ

Выполнили

студенты гр. ПИ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Гриценко И.А./

­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Бабкин А.С./

Проверил

преп.каф.БИ \_\_\_\_\_\_\_\_\_\_\_\_\_ / Кузьминых М.М./

Оценка «\_\_\_\_\_\_\_»

Киров  
2015

**Содержание**

**под изменение!!!**

1. *Введение. Постановка задачи курсовой работы.3*
2. *Анализ предметной области4*
   1. *Общее описание предметной области4*
   2. *Перечень источников информации для моделирования 7*
   3. *Инфологическая модель предметной области 7*
   4. *Модель процессов предметной области 8*
   5. *Модель состояний ключевых объектов предметной области 11*
   6. *Общесистемные требования и решения12*
3. *Выбор СУБД и средств реализации.14*
   1. *Оценка требований к СУБД14*
   2. *Сравнение и выбор СУБД.15*
   3. *Выбор платформы, языка программирования и среды разработки.17*
4. *Проектирование базы данных18*
   1. *Преобразование инфологической модели в даталогическую 18*
   2. *Определение основных транзакций для базы данных.19*
   3. *Оценка возможности оптимизации структуры базы данных.19*
5. *Разработка приложения для работы с базой данных.20*
   1. *Проектирование экранных форм20*
   2. *Разработка программного кода..23*
   3. *Запросы на языке SQL42*
6. *Заключение43*
7. *Приложение А. Запросы на создание базы данных в выбранной СУБД.44*
8. **Введение. Постановка задачи курсовой работы**

В современном мире увеличивается значение информационных систем, позволяющих обеспечить информационную поддержку процессов принятия решений. Базы данных являются одним из основных элементов систем, без них век информационных технологий не мог бы существовать и прогрессивно развиваться.

В нашем мире возрастает значение информационных систем, которые помогают обеспечить поддержку на информационном уровне

В базе данных могут храниться миллионы записей. Базы данных необходимы для многих областей деятельности человека, поэтому данная курсовая работа крайне актуальна и важна.

Целесообразный расчет стоимости тура важная часть для работы турфирм. Каждый клиент для поездки выбирает те или иные критерии комфортного отдыха. Для удобства расчета стоимости тура с учетом всех капризов клиентов, нами было предложено разработать систему, которая облегчит процесс расчета стоимости и учета критериев самих туров.

Целью курсовой работы является построение базы данных, закрепление основ и углубление знаний в области проектирования структуры базы данных, разработке и реализации модели базы данных, а также изучение основных элементов языка SQL.

Задачи курсовой работы состоят в следующем:

* Реализовать базу данных;
* Реализовать функции, выявленные в рамках анализа предметной области и построения модели процессов;
* Организовать разграничение полномочий пользователей, так, чтобы для каждой функциональной роли были доступны только соответствующие операции для каждой роли реализовать отдельный пользовательский интерфейс;
* Описание автоматизированных функций информационной системы
* Построение математической модели
* Проектирование алгоритма работы с программой и алгоритмы работы пользователя с программой
* Формирование экранных форм и программного кода в соответствии с алгоритмами

1. **Анализ предметной области**

**2.1. Общее описание предметной области**

Среда функционирования информационной системы «Расчет стоимости тура».

Информационная система расчета стоимости тура позволяет упорядочить работу туристического агентства, т.е. при расчете учесть не только все аспекты комфортного отдыха клиентов, но и все аспекты оборотных средств предприятия. Информационная система позволяет печатать данные о путешествии на типографском бланке - нет необходимости заполнять бланк от руки, намного облегчается ведение и анализ введенных данных, упрощается работа и расчеты с клиентами: расчетные документы легко могут быть оформлены и учтены. Информация о путешествии вносятся в базу один раз и используются для оформления всех необходимых документов: путевок, заявок, договоров, памяток и т.д. При этом значительно уменьшается вероятность ошибок при заполнении документов. Информационная система позволяет легко собирать и анализировать статистические данные

Объект управления представляет собой подсистему материальных элементов экономической деятельности и хозяйственных процессов.

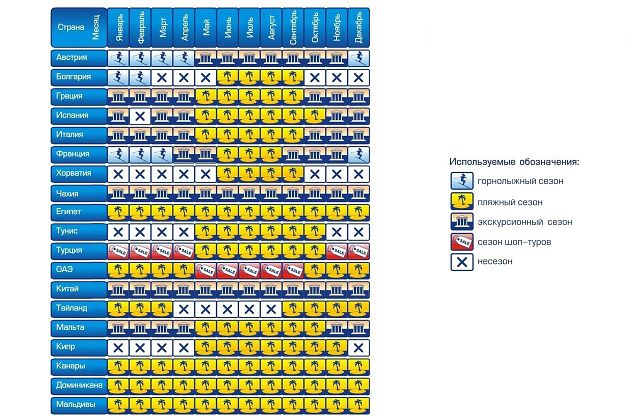
Субъектами управления системы «Расчет стоимости тура» являются работники туристического агентства. Но, в первую очередь, данная информационная система адресована руководителям туристических агентств, как инструмент эффективного управления офисами продаж.

Цель расчетов заключается в повышении качества обслуживания клиентов за счет сокращения сроков обслуживания на всех этапах, начиная с подбора подходящего тура и заканчивая оформлением документов, а также за счет сокращения ошибок персонала. Все это, в конечном итоге, должно привести к увеличению прибыли предприятия

Обычно, для расчета стоимости тура указываются следующие данные:

* страна
* сезон
* проживание
* страховка
* транспорт
* оборотные средства турагенства
* скидки

1. Для учета актуальности посещения данный страны в сезоне клиентам будет представлена следующая таблица:



Таким образом, чтобы учесть актуальность сезона делается следующее:

1. Если в стране актуален пляжный сезон, то стоимость тур пакета увеличивается в 1,25 раза
2. Если в стране актуален сезон шоп-туров, то стоимость увеличивается в 1,20 раза
3. Если в стране актуален сезон экскурсий, то цена увеличивается в 1,15 раза
4. Если в стране актуален горнолыжный сезон, то цена увеличивается в 1,10 раза
5. Если в стране не сезон, то цена увеличивается в 1,02 раза

Страны Европы, к стоимости тура прибавляется 1500. Страны Африки-2000. ОАЭ к стоимости прибавляется 3500.

1. Для учета стоимости транспорта клиенту будет представлена следующая таблица (отдельная по каждой стране):

|  |  |  |
| --- | --- | --- |
| Вид транспорта: | Класс: | Стоимость: |
| Авиакомпания «UTair» | Бизнес-класс | 15000-17000 |
|  | Эконом-класс | 8000-15000 |
| РЖД: | Купе | 6000 |
|  | Плацкарт | 3500 |
| Автобус: | - | 3000 |

1. Для учета размера скидки клиента должна быть представлена следующая таблица:

|  |  |
| --- | --- |
| Скидка для детей до 7 лет | 25% |
| Скидка для студентов | 10% |
| Скидка для пенсионеров | 25% |
| Скидка для постоянных клиентов | 7% |
| Скидка для клиентов с подарочными картами | 5% |

1. Для учета стоимости проживания клиенту должна быть представлена следующая таблица:

(отдельно по каждой стране)

|  |  |
| --- | --- |
| Место проживания: | Стоимость |
| Отель 5звезд | 3000 |
| Отель 4 звезд | 2000 |
| Отель 3 звезды | 1500 |
| Отель 2 звезды | 800 |
| Хостел | 600 |

1. Для учета стоимости страховки клиенту будет предоставлена следующая таблица:

Таким образом, для расчета стоимости тура будут использованы следующие формулы

**Ц =**

Где: Ц - тариф турпакета на одного туриста;

С - себестоимость турпакета;

Н - косвенные и прямые налоги, пошлины и сборы;

П - прибыль туроператора;

Д - дисконты, скидки и бонусы с цены;

КВ - комиссионное отчисление турагенту;

Ч - количество туристов;

Г- количество сопровождающих лиц.

**С=**

**Н=налоги на имущество(у нас это будет фиксированная цена, 10000)+0,3\*С**

**КВ=0,15\*С**

**П=Выручка туроператора-С-Н**

Пример:

Чтобы рассчитать стоимость тура на 5 человек, которые едут в Египет на 10 дней в пляжный сезон, летят эконом-классом, отель 3 звезды и один из них ребенок 5 лет.

С=1500+1500\*10+1000=26500

Н=10000+0,3\*26500=17950

П=66300-26500-17950=21850

Д=26500/4=6625

КВ=3975

Ч=5

Г=0

В итоге, информационная система должна выполнять следующие задачи:

* Ввод/Вывод заданных параметров расчета стоимости тура, соответствуя определенной требованиям клиента и турфирмы
* Формирование документов по критериям

В соответствии с данными задачами будут сформированы:

* экранные формы программы,
* основы взаимодействия с пользователем
* документы расчетов, сохраненные в отдельный файл для дальнейшего распространения или переноса на бумажные носители.

**2.2. Перечень источников информации для моделирования**

Одним из условий успешного функционирования любого предприятия является наличие необходимой информации, которую необходимо обрабатывать. Ядром информационной системы являются хранимые данные.

Источниками являются документы, описывающие бизнес-процессы турфирмы: годовой план турпакетов, договор оказания услуг, договор поставки, продолжительность тура, цена тура, код страны, название страны, чек

Формирование документов производится персоналом, обладающим соответствующими правами, со своего рабочего места.

**2.3. Инфологическая модель предметной области**

Рисунок 2 — Инфологическая модель

**2.4. Модель процессов предметной области**

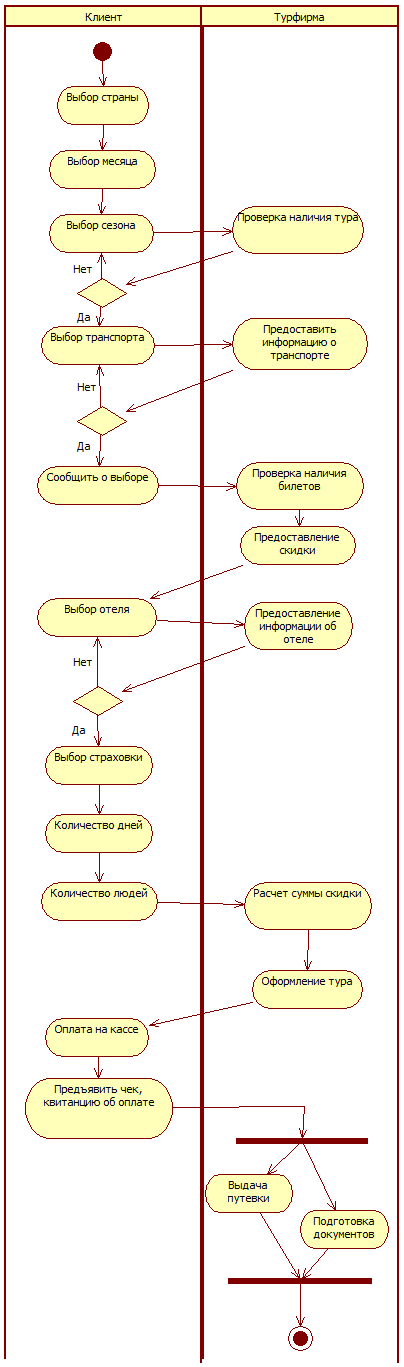


Рисунок 3 — Модель приобретения тура

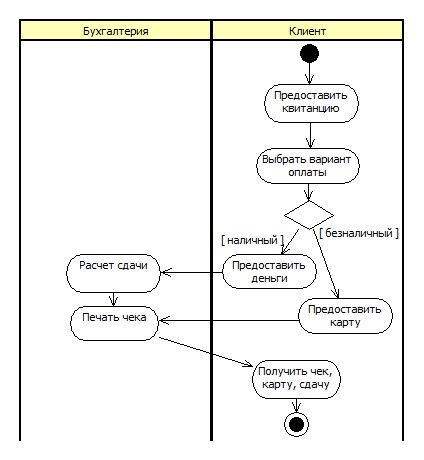
****

Рисунок 5 — Модель оплаты на кассе

**2.5. Модель состояний ключевых объектов предметной области**

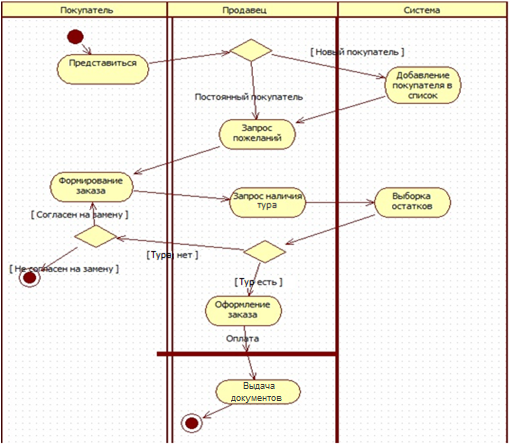
****

Рисунок 6 —– Модель состояния ключевых объектов

* 1. **Общесистемные требования и решения**

|  |  |
| --- | --- |
| **Показатели** | **Выявленные значения** |
| Требования по обеспечению безопасности и разграничению  доступа к базе данных: | |
| Защита базы данных от несанкционированного доступа | Требуется |
| Роли для разных групп пользователей | Требуется |
| Возможности шифрования содержимого базы данных | Не требуется |
| Требования по надежности работы базы данных: | |
| Количество сбоев, которое допускается в определенный период времени | 1 в год |
| Период, в течение которого должна быть восстановлена работоспособность системы | В кратчайшие строки (максимум день) |
| Специальные средства резервного копирования и восстановления данных | Требуется |
| Требования по активности работы с данными: | |
| Количество операций в единицу времени, которое предполагается выполнять в проектируемой базе данных | Приблизительно 500 операций в день |
| Частость добавления новых данных и изменений существующих | При каждом обращении |
| Формирование сложных запросов и выборок из базы данных | Да |
| Требования, позволяющие оценить объем базы данных: | |
| Объем данных, который попадет в базу данных изначально | Приблизительно 300 записей |
| Объем данных, который будет добавляться в базу данных в течение месяца, года | Приблизительно 10000 записей в год |
| Период времени, который должны храниться данные в базе данных (старые данные могут архивироваться) | 2 года |
| Требования, позволяющие оценить сетевую конфигурацию системы: | |
| Структура предприятия, организация доступа к базе данных для удаленных подразделений | Требуется |
| Интенсивность запросов данных через сеть | Высокая, требуется высокоскоростная передача данных разного рода |
| Создание дополнительных копий базы данных в удаленных подразделениях и синхронизация данных между ними | Требуется |
| Требования по максимально возможному числу активных пользователей базы данных | |
| Максимальное число активных пользователей | 100 пользователя |
| Требования по производительности системы: | |
| Количество операций в единицу времени, которое должна выполнять система | 500 операций в день |
| Время выполнения отдельных операций (вставки, обновления, выборки), которое требуется от базы данных | В пределах минуты |
| Требования к возможностям доработки базы данных: | |
| Перенос данных на другую платформу, использование системы в многоплатформенной среде | Не требуется |

Точное количество пользователей предугадать невозможно. Что касается размера базы данных, то 100 Гб под хранение данных должно хватить, так как СУБД в основном будет активно заполняться текстовой информацией (договора и т.д.), а медиаданные будут присутствовать относительно в меньшем количестве, и частота их обновления будет реже. Количество узлов в сети (подразделений) — 12 узлов локальной сети, планируется использовать на каждый отдел + резерв. Таким образом, маска сети будет иметь вид 255.255.240.0, что позволит содержать до 15 узлов и включать 4096 хвостовых компьютеров.

Характеристика технических деталей такова: СУБД: MySQL. Из оборудования используется серверная машина под управлением ms windows + my sql + ftp и персональные рабочие станции под управлением ms windows + .net 4 и старше. Также применяется смешанная топология сети. А разработка приложения ведётся с использованием net framework 4 и win forms.

1. **Выбор СУБД и средств реализации**

**3.1. Оценка требований к СУБД**

|  |  |
| --- | --- |
| **Критерий** | **Важность критерия** |
| Моделирование данных: | |
| Используемая модель данных | 5 |
| Возможность применения триггеров и хранимых процедур | 4 |
| Средства поиска | 2 |
| Предусмотренные типы данных | 4 |
| Реализация языка запросов | 5 |
| Особенности архитектуры и функциональные возможности: | |
| Переносимость, то есть независимость системы от среды, в которой она работает | 3 |
| Масштабируемость | 4 |
| Распределенность, то есть возможность разделения системы на несколько частей, физически удаленных друг от друга | 3 |
| Сетевые возможности. | 4 |
| Контроль работы системы: | |
| Ведение журналов производительности и контроль использования ресурсов системы | 4 |
| Имеющийся инструментарий управления и настройки | 5 |
| Особенности разработки приложений: | |
| Многие производители СУБД выпускают также средства разработки приложений для своих систем | 3 |
| Многоязыковая поддержка | 4 |
| Поддерживаемые языки программирования | 4 |
| Производительность: | |
| Рейтинг TPC | 2 |
| Возможности оптимизирования запросов | 3 |
| Надежность: | |
| Восстановление после сбоев | 5 |
| Резервное копирование | 3 |
| Наличие средств управления и контроля доступа пользователей к данным | 3 |
| Требования к рабочей среде: | |
| Поддерживаемые аппаратные платформы | 3 |
| Минимальные требования к оборудованию | 3 |
| Операционные системы, под управлением которых способна работать СУБД | 2 |
| Смешанные критерии: | |
| Качество и полнота документации | 3 |
| Локализованность | 3 |
| Стоимость | 3 |
| Стабильность производителя | 4 |
| Распространенность СУБД | 3 |

**3.2. Сравнение и выбор СУБД**

Для сравнения выберем три современные системы управления базами данных: Oracle RDBMS, IBM DB2, MySQL. Все СУБД соответствуют тем требованиям, которые были выявлены в ходе анализа предметной области.

А именно Oracle RDBMS обладает такими характеристиками:

* Возможность разбиения крупных баз данных на разделы, что дает возможность эффективно управлять гигантскими гигабайтными базами;
* Наличие универсальных средств защиты информации;
* Эффективные методы максимального повышения скорости обработки запросов;
* Свободные таблицы (в других СУБД все таблицы заполняются сразу при создании);
* Распараллеливание операций в запросе;
* Наличие широкого спектра средств разработки, мониторинга и администрирования;
* Средство передачи журналов, позволяющее синхронизировать отдельные системы посредством автоматизированного обновления журнала транзакций, что повышает надежность системы и устойчивость к авариям;
* Улучшенные средства отказоустойчивости, упрощающие создание и управление отказоустойчивыми кластерами, переключение между узлами кластера при сбое и восстановление узла кластера, не оказывающее влияния на остальные узлы;
* Поддержка большого объема памяти и симметричной многопроцессорной обработки, позволяющая управлять приложениями с высокими нагрузками, связанными с большим объемом транзакций

Характеристики IBM DB2:

* Распараллеливание обработки запроса;
* Полный набор средств тиражирования;
* Сводные таблицы запросов для повышения производительности;
* Оснащена полым набором мультимедиа-расширений

MySQL:

* Простота администрирования;
* Быстродействие и функциональные возможности механизма сервера СУБД;
* Многопоточность, поддержка нескольких одновременных запросов;
* Оптимизация связей с присоединением многих данных за один проход;
* Записи фиксированной и переменной длины;;
* Гибкая система привилегий и паролей;
* гибкая поддержка форматов чисел, строк переменной длины и меток времени;
* быстрая поддержка транзакций через механизм InnoDB

Оценка требований по 3-ех бальной шкале:

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Важность критерия** | | |
| для Oracle | для DB2 | для MySQL |
| Моделирование данных: | | | |
| Используемая модель данных | 3 | 3 | 3 |
| Возможность применения триггеров и хранимых процедур | 3 | 2 | 3 |
| Средства поиска | 2 | 2 | 2 |
| Предусмотренные типы данных | 2 | 2 | 2 |
| Реализация языка запросов | 3 | 3 | 3 |
| Особенности архитектуры и функциональные возможности: | | | |
| Переносимость, то есть независимость системы от среды, в которой она работает | 3 | 3 | 2 |
| Масштабируемость | 3 | 3 | 3 |
| Распределенность, то есть возможность разделения системы на несколько частей, физически удаленных друг от друга | 3 | 2 | 3 |
| Сетевые возможности. | 3 | 3 | 3 |
| Контроль работы системы: | | | |
| Ведение журналов производительности и контроль использования ресурсов системы | 2 | 1 | 3 |
| Имеющийся инструментарий управления и настройки | 2 | 1 | 3 |
| Особенности разработки приложений: | | | |
| Многие производители СУБД выпускают также средства разработки приложений для своих систем | 1 | 1 | 3 |
| Многоязыковая поддержка | 2 | 2 | 2 |
| Поддерживаемые языки программирования | 3 | 2 | 1 |
| Производительность: | | | |
| Рейтинг TPC | 2 | 2 | 2 |
| Возможности оптимизирования запросов | 2 | 2 | 3 |
| Надежность: | | | |
| Восстановление после сбоев | 2 | 2 | 2 |
| Резервное копирование | 3 | 2 | 3 |
| Наличие средств управления и контроля доступа пользователей к данным | 3 | 3 | 3 |
| Требования к рабочей среде: | | | |
| Поддерживаемые аппаратные платформы | 3 | 3 | 3 |
| Минимальные требования к оборудованию | 2 | 2 | 2 |
| Операционные системы, под управлением которых способна работать СУБД | 3 | 2 | 1 |
| Смешанные критерии: | | | |
| Качество и полнота документации | 2 | 3 | 3 |
| Локализованность | 3 | 2 | 3 |
| Стоимость | 3 | 3 | 3 |
| Стабильность производителя | 3 | 3 | 3 |
| Распространенность СУБД | 3 | 1 | 2 |
| Сумма баллов: | 60 | 69 | 60 |
| **Итоговые оценки:** | 241 | 210 | 244 |

MySQL и Oracle относятся к классу многпользовательских систем. Они развиваются и совершенствуются уже в течение многих лет и реализуются как коммерческие программные продукты.

Исходя из получившихся итоговых оценок, можно сделать вывод, что для нашего проекта лучшей СУБД является MySQL. Одним из преимуществ MySQL является простота его применения, в частности администрирования. В принципе данная СУБД представляет собой современную полнофункциональную база данных, которая идеально подходит для малых и средних организаций, а значит и идеально подойдет для нашего автосалона.

**3.3. Выбор платформы, языка программирования и среды разработки**

Для решения поставленной задачи необходимо использовать функциональную, эффективную и удобную платформу для разработки, в качестве такой платформы была выбрана среда .NET Framework. Платформа состоит из двух частей. Основой является исполняющая среда Common Language Runtime (CLR), которая может выполнять как обычные программы, так и серверные приложения. Вторая, не менее важная часть, это библиотека классов Framework Class Library (FCL), содержащая в себе множество компонентов для работы с базами данных, сетью, вводом/выводом, файлами, пользовательским интерфейсом и т.д. К тому же набор компонентов .NET, известный под общим названием ADO.NET, позволяет получать эффективный доступ к реляционным базам данных и многим другим источникам данных.

Среды разработки, поддерживающие .NET: Microsoft Visual Studio, SharpDevelop, MonoDevelop, Embarcadero RAD Studio, Zonnon, PascalABC.NET. Из данных сред разработки мы решили остановиться на Microsoft Visual Studio, которая позволяет разработчикам быстро создавать подключаемые к базам данных приложения, способные обеспечить широчайшие возможности для работы пользователей, также содержит усовершенствованные средства разработки, отладчики, функции работы с базами данных.

Выбранный язык программирования - C#.

1. **Проектирование базы данных**

**4.1. Преобразование инфологической модели в даталогическую**

Схема данных в графическом виде, созданная в SQL Server, представлена на рисунке 7.

Рисунок 7 —Датологическая модель

**4.2. Определение основных транзакций для базы данных**

Основной транзакцией является транзакция на добавление тура. Имея уровень транзакции "SERIALIZABLE" объект, над которым идет манипуляция, полностью блокируется от воздействия на него других запросов, обеспечив тем самым целостность и достоверность информации. Транзакция средней сложности занимает очень малое количество времени, поэтому доступ к данным не будет затрудняться.

**4.3. Оценка возможности оптимизации структуры базы данных**

Одна из основных задач оптимизации заключается в том, чтобы данные (и индексы) занимали как можно меньше места на диске (и в памяти). Это дает значительные преимущества в работе, поскольку ускоряется чтение диска, а оперативная память, как правило, используется меньше. Индексирование также требует меньших ресурсов, если оно выполняется на меньших столбцах. MySQL поддерживает большое количество различных типов таблиц и форматов строк.

Путем использования правильной ссылочности и типов полей объём данных был сокращен до минимума. Кроме того большие таблицы были разделены на более мелкие, что помогло сократить время на выборку данных, тем самым повышена производительность БД. Также не допускалась избыточность информации, то есть предполагалось не хранить одинаковую информацию в нескольких таблицах, а объединять данные в этих таблицах числовыми ссылками. Первичные индексы в таблице были сделаны как можно короче, чтобы идентификация конкретной записи стала простой и эффективной. К тому же объявлялись везде, где возможно, столбцы как NOT NULL, что позволило ускорить все операции и сэкономить по одному биту для каждого столбца.

1. **Разработка приложения для работы с базой данных**

**4.1. Проектирование экранных форм**

На рисунке 8.1 представлена форма выбора в процессе работы приложения.

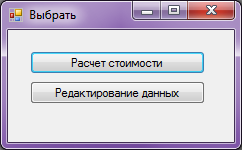
****

Рис 8.1 - Форма выбора во время работы приложения

На рисунке 8.1 предоставлена форма приложения в процессе работы приложения. Данная форма предназначена для выбора страны и сезона тура.

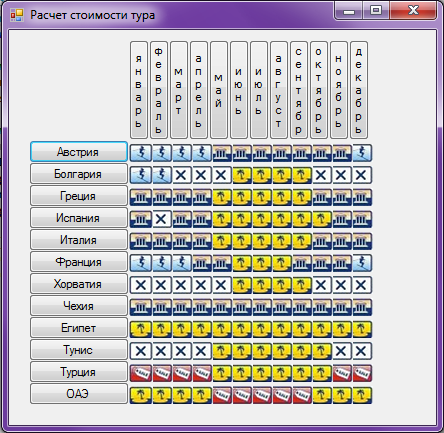


Рисунок 8.1 — Форма выбора страны и сезона тура во время работы приложения

На рисунке 9.1 показан выбор транспорта и его стоимость (в пример взят горнолыжный сезон в Австралии). Выбор транспорта отдельный по каждой стране.

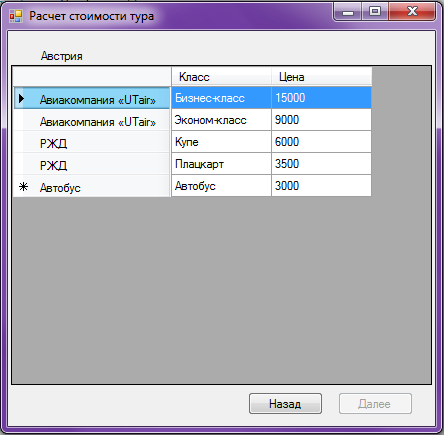


Рисунок 9.1 — Форма выбора транспорта и его стоимость в процессе разработки

На рисунке 9.2 показан размер скидки клиента(ов).

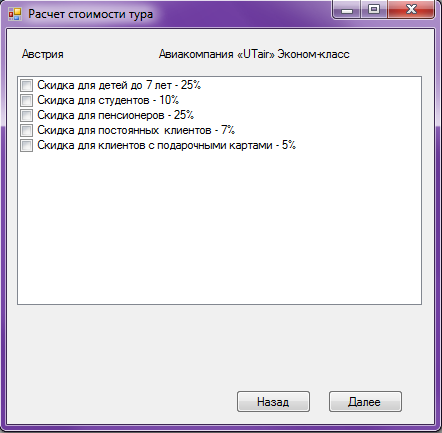


Рисунок 9.2 — Форма размера скидки клиента в процессе разработки

На рисунке 10.1 представлена форма выбора отеля, категории от одной звезды отеля до пятизвездочного.

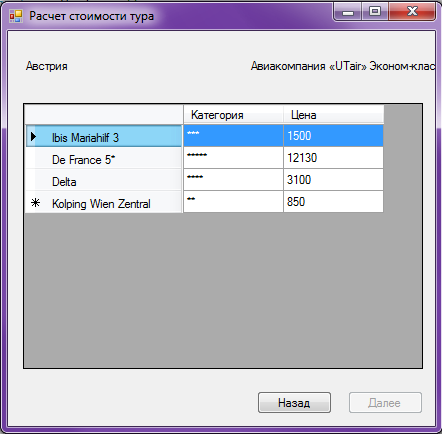


Рисунок 10.1 — Форма выбора отеля в процессе разработки

На рисунке 10.2 представлена форма выбора страховки для клиента(ов).

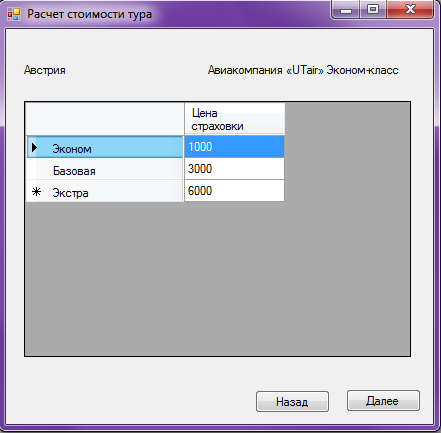


Рисунок 10.2 — Форма выбора страховки в процессе разработки

На рисунке 10.3 представлена форма расчета тура, с выбором дней, количеством взрослых и детей.

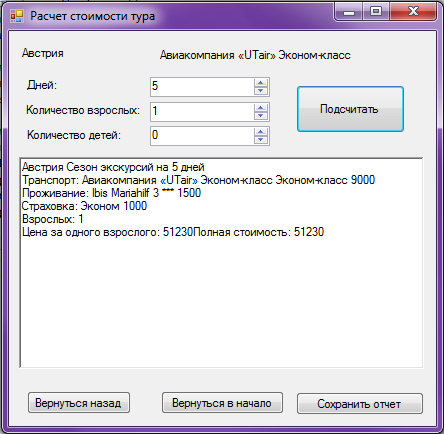


Рисунок 10.3 — Форма расчета тура во время работы приложения

На рисунке 8.1 представлена форма редактирования стран. Данная форма предназначена для добавления/удаления страны и ее описания, а так же редактирования сезона.

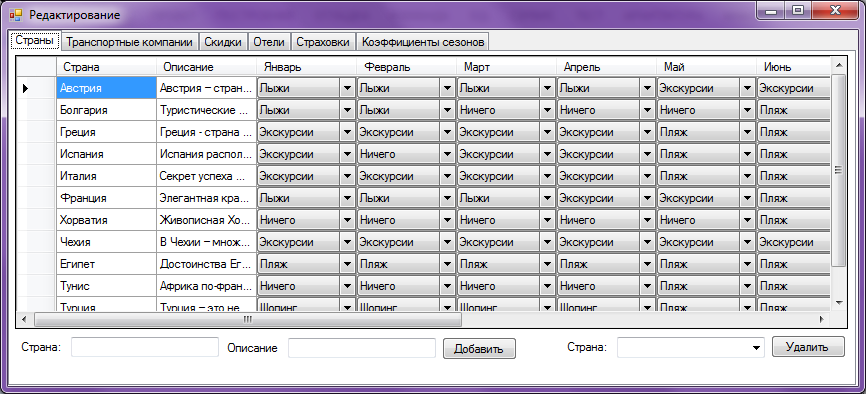


Рис 8.1 Форма редактирования страны

На рисунке 8.1 представлена форма редактирования транспортной компании. Данная форма предназначена для добавления/удаления компании с ее наименованием и стоимостью.

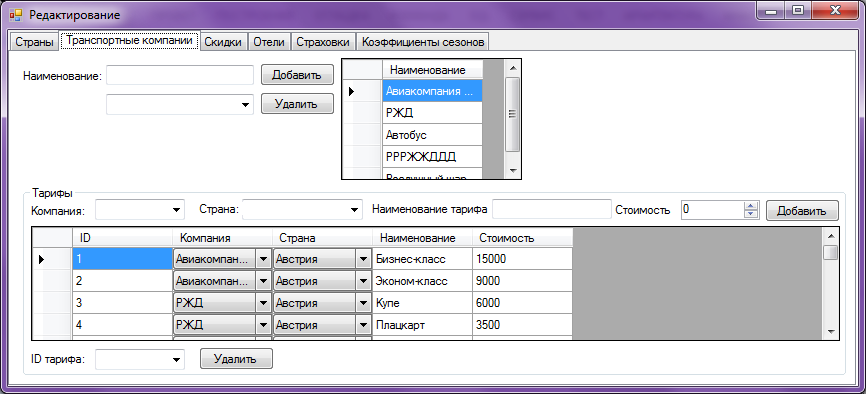


Рис 8.1 - Форма редактирования транспортной компании.

На рисунке 8.1 представлена форма редактирования скидки. В этой форме можно добавлять и удалять скидки

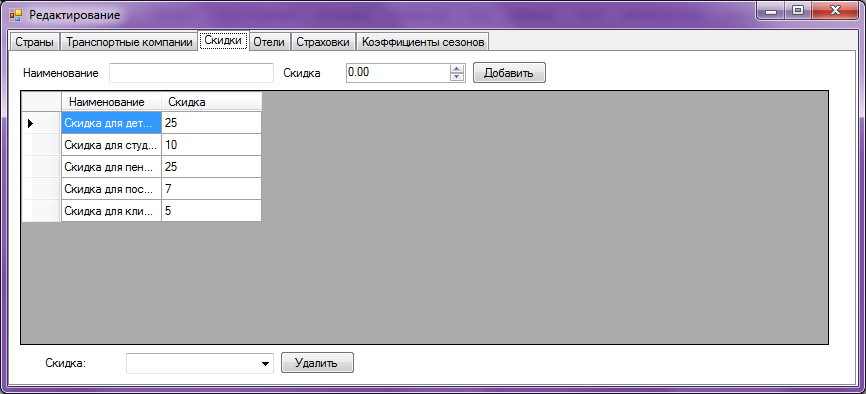


Рис 8.1 - Форма редактирования скидки

На рисунке 8.1 представлена форма редактирования отелей. Форма предназначена для добавления/удаления отелей. Так же можно указать страну и количество звезд для отеля.

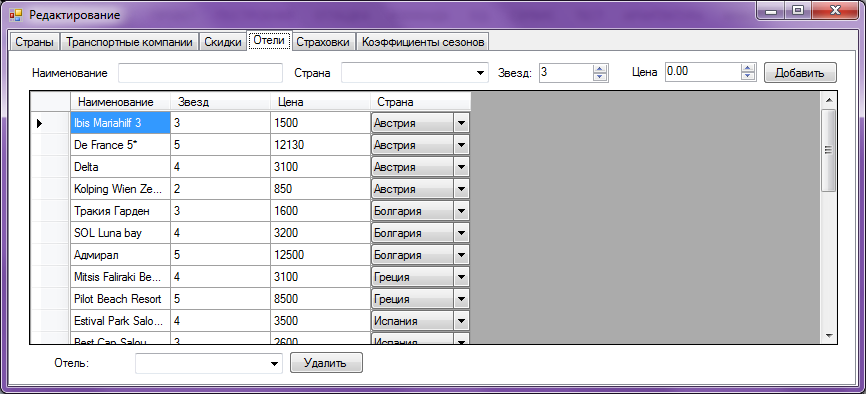


Рис 8.1 - Форма редактирования отеля

На рис 8.1 представлена форма редактирования страховки. На данной форме можно добавлять/удалять страховку.

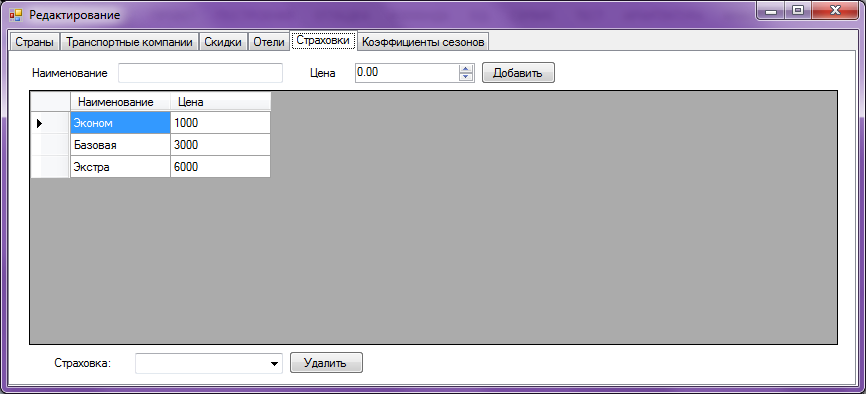


Рис 8.1 - Форма редактирования страховки

**4.2. Разработка программного кода**

public partial class Form1 : Form

{

Tours tours;

public Form1()

{

InitializeComponent();

}

private void click(object sender, EventArgs e)

{

int x, y;

PictureBox btn = (sender as PictureBox);

string str = (string)btn.Tag;

string[] coords = str.Split('-');

y = int.Parse(coords[0]);

x = int.Parse(coords[1]);

tours.select(x, y);

dataGridView1.RowCount = tours.selected.transport.Count;

dataGridView1.ColumnCount = 2;

dataGridView1.RowHeadersWidth = 160;

dataGridView1.Columns[0].HeaderCell.Value = "Класс";

dataGridView1.Columns[1].HeaderCell.Value = "Цена";

for (int i = 0; i < tours.selected.transport.Count; i++)

{

dataGridView1.Rows[i].HeaderCell.Value = tours.selected.transport[i].company;

dataGridView1.Rows[i].Cells[0].Value = tours.selected.transport[i].classType;

dataGridView1.Rows[i].Cells[1].Value = tours.selected.transport[i].cost;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

if (!MySql.init()) {

MessageBox.Show("Не удалось подключиться к базе данных");

return;

}

tours = new Tours();

tours.loadCountries();

for (int i = 0; i < tours.counries.Count; i++)

{

Button b = new Button();

b.Text = tours.counries[i].name;

b.Width = 100;

b.Location = new System.Drawing.Point(20, 110 + i \* 22);

this.panel1.Controls.Add(b);

}

string[] months = {"январь","февраль","март","апрель","май","июнь","июль","август","сентябрь","октябрь","ноябрь","декабрь" };

string[] files = { "plyag.png", "shop.png", "ekskursion.png", "ligi.png", "nothing.png" };

for (int i = 0; i < 12; i++)

{

Button b = new Button();

b.Text = months[i];

b.Width = 20;

b.Height = 100;

b.Location = new System.Drawing.Point(120 + i \* 20, 10);

this.panel1.Controls.Add(b);

}

for (int i = 0; i < tours.counries.Count; i++)

for (int j = 0; j < 12; j++)

{

PictureBox b = new PictureBox();

b.Tag = i+"-"+j;

b.ImageLocation = files[tours.counries[i].seasons[j]];

b.Width = 24;

b.Height = 18;

b.Location = new System.Drawing.Point(120 + j \* 20, 112 + i \* 22);

b.Click += new EventHandler(click);

this.panel1.Controls.Add(b);

}

private void button9\_Click(object sender, EventArgs e)

{

float summa = 0;

string report = "";

float cost = 0;

report += label1.Text + " " + tours.selected.getSeasonName( tours.season) +

" на " + numericUpDown1.Value + " дней \n";

report += "Транспорт: " + label3.Text + " " + dataGridView1.SelectedRows[0].Cells[0].Value + " "

+ dataGridView1.SelectedRows[0].Cells[1].Value + " \n";

report += "Проживание: " + dataGridView2.SelectedRows[0].HeaderCell.Value + " " + dataGridView2.SelectedRows[0].Cells[0].Value + " "

+ dataGridView2.SelectedRows[0].Cells[1].Value + " \n";

report += "Страховка: " + dataGridView3.SelectedRows[0].HeaderCell.Value + " " + dataGridView3.SelectedRows[0].Cells[0].Value + " "

+ " \n";

int peoples = (int)numericUpDown2.Value;

report += "Взрослых: " + peoples + " \n";

for (int i = 0; i < peoples; i++)

{

float s = tours.selected.getAdditionalCost() +

tours.strahovka[dataGridView3.SelectedRows[0].Index].cost +

tours.selected.transport[dataGridView1.SelectedRows[0].Index].cost \* 2+

((int)numericUpDown1.Value) \* tours.selected.hotels[dataGridView2.SelectedRows[0].Index].cost;

float n = 10000 + 0.3f \* s;

float kv = 0.15f \* s;

float p = kv - s - n;

float act = tours.selected.getSeasonK(tours.season);

float d = 0;

for (int j = 1; j < checkedListBox1.Items.Count; j++)

if (checkedListBox1.GetItemChecked(j))

{

d += tours.discounts[j].cost;

report += "Скидка: " + checkedListBox1.GetItemText(checkedListBox1.Items[j]) + "\n";

}

if (d > 0)

{

d = s \* (1 - d / 100);

report += "Общая скидка: " + d + "\n";

}

cost = s + n - d + kv \* act;

summa += cost;

}

report += "Цена за одного взрослого: " + cost;

peoples = (int)numericUpDown3.Value;

if (peoples != 0)

{

report += "Детей: " + peoples + " \n";

for (int i = 0; i < peoples; i++)

{

float s = tours.selected.getAdditionalCost() +

tours.strahovka[dataGridView3.SelectedRows[0].Index].cost +

tours.selected.transport[dataGridView1.SelectedRows[0].Index].cost \* 2 +

((int)numericUpDown1.Value) \* tours.selected.hotels[dataGridView2.SelectedRows[0].Index].cost;

float n = 10000 + 0.3f \* s;

float kv = 0.15f \* s;

float p = kv - s - n;

float act = tours.selected.getSeasonK(tours.season);

float d = 0;

for (int j = 1; j < checkedListBox1.Items.Count; j++)

if (checkedListBox1.GetItemChecked(j))

{

d += tours.discounts[j].cost;

report += "Скидка: " + checkedListBox1.GetItemText(checkedListBox1.Items[j]) + "\n";

}

d = s\*(1-(d+tours.discounts[0].cost)/100);

report += "Общая скидка: " + d + "\n";

cost = s + n - d + kv \* act;

summa += cost;

}

report += "Цена за одного ребенка: " + cost;

}

report += "Полная стоимость: " + summa;

richTextBox1.Text = report;

}

private void button10\_Click(object sender, EventArgs e)

{

object oMissing = System.Reflection.Missing.Value;

Word.\_Application oWord;

Word.\_Document oDoc;

oWord = new Word.Application();

oDoc = oWord.Documents.Add(ref oMissing, ref oMissing,

ref oMissing, ref oMissing);

Word.Paragraph oPara1;

oPara1 = oDoc.Content.Paragraphs.Add(ref oMissing);

oPara1.Range.Text = richTextBox1.Text;

oPara1.Format.SpaceAfter = 24;

oPara1.Range.InsertParagraphAfter();

}

}

**Класс страна**

namespace Turism

{

class Country

{

public int id;

public string name;

public string description;

public int[] seasons;

public List<Transport> transport;

public List<Hotel> hotels;

public Country(int id, string name, string description)

{

this.id = id;

this.name = name;

this.description = description;

seasons = new int[12];

}

public float getAdditionalCost()

{

return MySql.getAdditionalCost(id);

}

public void addSeason(int month, int season)

{

seasons[month] = season;

}

public void loadTransport()

{

transport = MySql.getTransport(id);

}

public List<string> getTransportCompanies()

{

List<string> result = new List<string>();

for (int i = 0; i < transport.Count; i++)

{

bool isEl = false;

for (int j = 0; j < result.Count; j++)

{

if (transport[i].company == result[j])

{

isEl = true;

break;

}

}

if (!isEl)

{

result.Add(transport[i].company);

}

}

return result;

}

public void loadHotels()

{

hotels = MySql.getHotels(id);

}

public float getSeasonK(int id)

{

return MySql.getSeasonK(seasons[id]);

}

public string getSeasonName(int id)

{

return MySql.getSeasonName(seasons[id]);

}

}

}

**4.3. Запросы на языке SQL**

Пример запроса на добавление объекта:

"INSERT INTO transporttarif (id,company,name,country,cost) VALUES (NULL, " + companyId + ", '" + name + "', " + countryId + "," + cost + ")";

Пример запроса на изменение объекта:

"UPDATE discount SET name='" + name + "', discount="+disc+" WHERE name='" + name + "'"

Пример запроса на удаление объекта:

"DELETE FROM discount WHERE name='"+name+"'"

Выборка объекта:

SELECT transporttarif.id, transportcompany.transport, transporttarif.name, transporttarif.cost " +" FROM transporttarif,transportcompany WHERE transportcompany.id=transporttarif.company AND country=" + id

1. **Заключение**

В ходе создания данного курсового проекта были закреплены теоретические знания, полученные в процессе обучения, а также углублены имеющиеся. Были выполнены все поставленные задачи, и учтены все общие требования, которым должно отвечать разрабатываемое приложение в рамках выполнения курсовой работы. К тому же был проведен анализ деятельности турфирмы.

Разработанный проект позволяет быстро и эффективно работать с данными. Удобный интерфейс программы, с одной стороны, позволяет легко ориентироваться в программе, не требуя от пользователя каких-либо специальных навыков работы с электронно-вычислительными машинами, а с другой стороны предоставляет пользователю оперативную информацию о стоимости товара и прочую полезную для пользователя информацию.