

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Математическое обеспечение и применение ЭВМ»

Курсовая работа

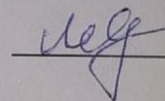
По дисциплине «Базы и хранилища данных»

на тему «Приложение учета и анализа заявок клиентов для сервиса
турагентства»

ПГУ 09.03.04 – 07КП201.14 ПЗ

Направление подготовки - 09.03.04 Программная инженерия

Выполнил студент:

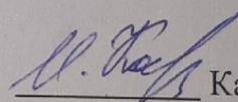


Левин М.В.

Группа:

20ВП1

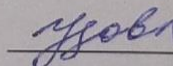
Руководитель работы



Казакова И.А.

доцент

Работа защищена с оценкой



Преподаватели

Дата защиты

8.06.2024г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Пензенский государственный университет

«УТВЕРЖДАЮ»

Зав. кафедрой МОиПЭВМ

_____ А. Ю. Козлов

ЗАДАНИЕ
на курсовую работу по дисциплине
«Базы и хранилища данных»

Студенту _____ Левину М.В. _____ Группа 20ВП1

Тема: Приложение учета и анализа заявок клиентов для сервиса турагентства

Исходные данные (технические требования) на проектирование

1. Провести анализ и определить границы предметной области (ПрОбл), выявить основные информационные задачи (процессы), выделить категории пользователей информационной системы (ИС), разработать сценарий выполнения информационных задач в виде диаграммы деятельности UML.
2. Провести моделирование автоматизируемых бизнес-процессов, для соответствующей категории пользователей ИС выделить не менее трех бизнес-процессов (БП) учетного характера (в т. ч., по крайней мере, один БП создания и добавления в БД коллекции информационных объектов в рамках единой транзакции; один БП, реализующий модификацию одного или нескольких информационных объектов в базе данных; один БП, реализующий явное или виртуальное удаление одного или нескольких информационных объектов в базе данных), один БП, выполняемый автоматически для приложения, один БП, выполняемый автоматически системой при наступлении в базе данных определенного события, связанного с модификацией данных) и не менее двух БП, которые должны формировать агрегированные группы данных, актуальные для анализа информации и возможного принятия решений.
3. Определить варианты использования приложения ИС и оформить соответствующую диаграмму UML.
4. Определить состав приложений ИС, определить архитектуру и состав средств реализации приложения ИС, в т. ч., выбрать целевую систему управления базами данных (СУБД).
5. На основе результатов анализа ПрОбл провести моделирование БД ИС с применением современных case-средств, оформить соответствующие диаграммы и отчеты по моделям БД, развернуть БД в среде целевой СУБД.
6. Разрабатываемое приложение должно реализовывать выборку данных из БД для отображения и реализации аналитических БП, реализацию выделенных на этапе моделирования БП учетного характера, а также автоматических БП.
7. Пользовательский интерфейс приложений должен обеспечивать привычное для ПрОбл и удобное для пользователя представление информации, простое и эффективное выполнение основных функций, доступных пользователям.
8. Программное обеспечение разрабатываемого приложения должно быть полностью отлажено и протестировано, необходимо привести описание процесса и результаты тестирования.

1. Объем работы по курсовому проекту

1. Анализ предметной области и постановка задачи на разработку приложения информационной системы
2. Моделирование бизнес-процессов и построение концептуальной модели базы данных информационной системы
3. Логическое и физическое моделирование базы данных информационной системы с применением современных case-средств
4. Развертывание базы данных информационной системы на сервере MS SQL Server и наполнение ее тестовыми данными
5. Разработка и отладка компонентов клиент-серверного приложения информационной системы
6. Тестирование клиент-серверного приложения информационной системы

2. Графическая часть

Пояснительная записка должна включать:

- диаграмму деятельности сценария выполняемых информационных задач;
- диаграмму и спецификации вариантов использования;
- диаграммы «сущность-связь» для различных уровней моделей БД;
- диаграмма для отображения архитектуры ИС;

3. Экспериментальная часть

1. Разработка, развертывание и наполнение тестовыми данными базы данных ИС;
2. Разработка, отладка и тестирование приложения информационной системы.

Срок выполнения проекта по разделам

1	Анализ предметной области и постановка задачи на разработку приложения ИС	к	29 сентября	2023 г.
2	Моделирование бизнес-процессов и разработка концептуальной модели БД	к	20 октября	2023 г.
3	Логическое и физическое моделирование БД ИС	к	27 октября	2023 г.
4	Разработка, отладка и тестирование программного обеспечения для реализации учетных и автоматических БД	к	24 ноября	2023 г.
5	Разработка, отладка и тестирование программного обеспечения аналитических БП	к	1 декабря	2023 г.
6	Оформление пояснительной записки	к	8 декабря	2023 г.
7	Защита курсовой работы	к	14 декабря	2023 г.

Дата выдачи задания 8 сентября 2023 г.

Дата защиты проекта «__» _____ 2024 г.

Руководитель И.А. Казакова /И.А. Казакова/

Задание получил _____ 8 сентября 2023 г.

Студент М.В. Левин /М.В. Левин/

Реферат

Пояснительная записка содержит 67 страниц, 44 рисунка, 22 таблицы, 5 использованных источников и 5 приложений.

БАЗА ДАННЫХ, ИНФОРМАЦИОННАЯ СИСТЕМА, КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ, СУБД, ПРЕДМЕТНАЯ ОБЛАСТЬ, MS SQL SERVER, ПРОЦЕДУРА, ТРИГГЕР, ФУНКЦИЯ, ПРЕДСТАВЛЕНИЕ, VISUAL PARADIGM.

Объект разработки – клиент-серверное приложение TravelAgency для информационной системы учета и анализа заявок клиентов для сервиса туристического агентства, созданное с использованием средств, предоставляемых современными СУБД реляционного типа.

Цель разработки: изучить предметную область и провести её анализ, выделить бизнес-процессы, разработать концептуальную, логическую и физическую модель базы данных в среде Visual Paradigm, а также реализовать и протестировать клиент-серверное приложение информационной системы.

Результаты разработки исследована и проанализирована предметная область туристического агентства, выделены основные бизнес-процессы, разработана база данных и клиент-серверное приложение TravelAgency для информационной системы учета и анализа заявок клиентов для сервиса туристического агентства.

Средства разработки: СУБД MS SQL SERVER 2018, case-средство Visual Paradigm, язык программирования TypeScript, среда разработки Visual Studio Code.

					ПГУ 09.03.04 – 07КП201.14 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Приложение учета и анализа заявок клиентов для сервиса турагентства Пояснительная записка	Лит.	Лист	Листов	
Разраб.		Левин М.В.	<i>М.В. Левин</i>						
Провер.		Казакова И.А.	<i>И.А. Казакова</i>				3	67	
Реценз.						Группа 20ВП1			
Н. Контр.									
Утверд.									

Содержание

ВВЕДЕНИЕ	6
1 Анализ предметной области и разработка концептуальной модели базы данных информационной системы	7
1.1 Постановка задачи на разработку приложения.....	7
1.2 Моделирование бизнес-процессов предметной области	12
1.3 Разработка концептуальной модели базы данных средствами Visual Paradigm	16
2 Моделирование и развертывание базы данных	18
2.1 Логическое моделирование базы данных средствами Visual Paradigm	18
2.2 Физическое моделирование базы данных средствами Visual Paradigm	24
2.3 Развертывание и наполнение базы данных тестовыми данными в среде MS SQL Server	25
2.3.1 Развертывание базы данных в среде MS SQL Server.....	25
2.3.2 Наполнение базы данных тестовыми данными	26
3 Разработка и тестирование приложения	36
3.1 Разработка серверной части приложения.....	36
3.1.1 Разработка запросов к базе данных и представлений для реализации аналитических бизнес-процессов	36
3.1.2 Разработка функций и хранимых процедур для реализации учетных пользовательских бизнес-процессов	37
3.1.3 Разработка триггеров для реализации учетных автоматических бизнес-процессов и обновления представлений	38
3.2 Разработка клиентской части приложения.....	39
3.2.1 Разработка пользовательского интерфейса клиентского приложения	39
3.2.2 Разработка программного обеспечения клиентского приложения	44
3.3 Тестирование приложения.....	45
3.3.1 Выбор режима тестирования приложения.....	45
3.3.2 Тестирование серверной части приложения	46
3.3.3 Тестирование клиентской части приложения	48
ЗАКЛЮЧЕНИЕ.....	52
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	53
ПРИЛОЖЕНИЕ А	54
ПРИЛОЖЕНИЕ Б	57
ПРИЛОЖЕНИЕ В.....	61
ПРИЛОЖЕНИЕ Г	64
ПРИЛОЖЕНИЕ Д.....	66

ВВЕДЕНИЕ

В современном мире организации все больше осознают важность эффективного управления информацией для успешного функционирования и развития своего бизнеса. Особенно это касается сферы туризма, где конкуренция постоянно растет, а клиентская база играет ключевую роль в привлечении и удержании клиентов. В контексте современных технологий и требований рынка, эффективное управление данными становится фундаментальным аспектом работы туроператоров и агентств

В данной курсовой работе будет реализовано приложение для учета и анализа заявок клиентов турагентства. Помимо серверной части приложения, также будет разработано клиентское приложение, работающее на персональном компьютере на операционной системе Windows 7 и выше.

При выполнении данной курсовой работы будут достигнуты следующие цели:

- формулировка и анализ требований;
- проектирование концептуальной схемы базы данных;
- построение логической и физической моделей базы данных;
- создание базы данных информационной системы и заполнение её тестовыми данными;
- создание аналитических запросов и представлений, создание хранимых процедур и функций, создание триггеров;
- разработка клиентского приложения;
- тестирование и отладка приложения.

В качестве СУБД будет использоваться MS SQL SERVER 2018. Разработка клиентского приложения будет осуществляться на языке программирования TypeScript при помощи технологии NextJS в среде разработки Visual Studio Code.

1 Анализ предметной области и разработка концептуальной модели базы данных информационной системы

1.1 Постановка задачи на разработку приложения

Турагентство - это организация, которая предоставляет услуги по организации и продаже туров, путешествий и отдыха для клиентов. Они работают с различными туроператорами, гостиницами, перевозчиками и другими поставщиками услуг, чтобы создать индивидуальные туры и пакеты для своих клиентов.

Основные функции турагентства включают в себя:

- консультации и планирование: турагенты помогают клиентам выбрать подходящий маршрут и услуги, учитывая их желания, бюджет и временные рамки;
- продажа туров и бронирование: турагенты продают туры и бронируют услуги для клиентов, такие как поездки, размещение в гостиницах, экскурсии и другие;
- организация документов: турагенты помогают клиентам получить все необходимые документы для путешествия, такие как визы, билеты, страховку и т. д.;
- оказание информационной поддержки: турагенты предоставляют информацию о местах назначения, культуре, климате, транспорте и других аспектах путешествия;
- поддержка во время путешествия: турагенты могут предоставлять поддержку во время путешествия, например, в случае непредвиденных обстоятельств или вопросов, связанных с бронированием.

Структурная организация предметной области "турагентство" зависит от размера, специализации и целевой аудитории турагентства. Однако, общая структура может включать в себя следующие элементы:

- руководство и администрация: отвечает за общую стратегию, развитие бизнеса и взаимоотношения с клиентами и партнерами;
- отдел маркетинга и продаж: отвечает за продвижение услуг турагентства, создание рекламных кампаний, разработку скидок и предложения, а также за продажу туров и услуг клиентам;
- отдел по работе с туроператорами: отвечает за бронирование туров, гостиниц, перевозок и других услуг, а также за обработку заявок и запросов клиентов;
- отдел документооборота: отвечает за обработку документов, связанных с путешествиями, таких как визы, билеты, страховку и т.д.;
- финансовый отдел: отвечает за финансовую политику, бухучет, контроль затрат и платежи.

На рисунке 1 показана структурная схема предметной области.

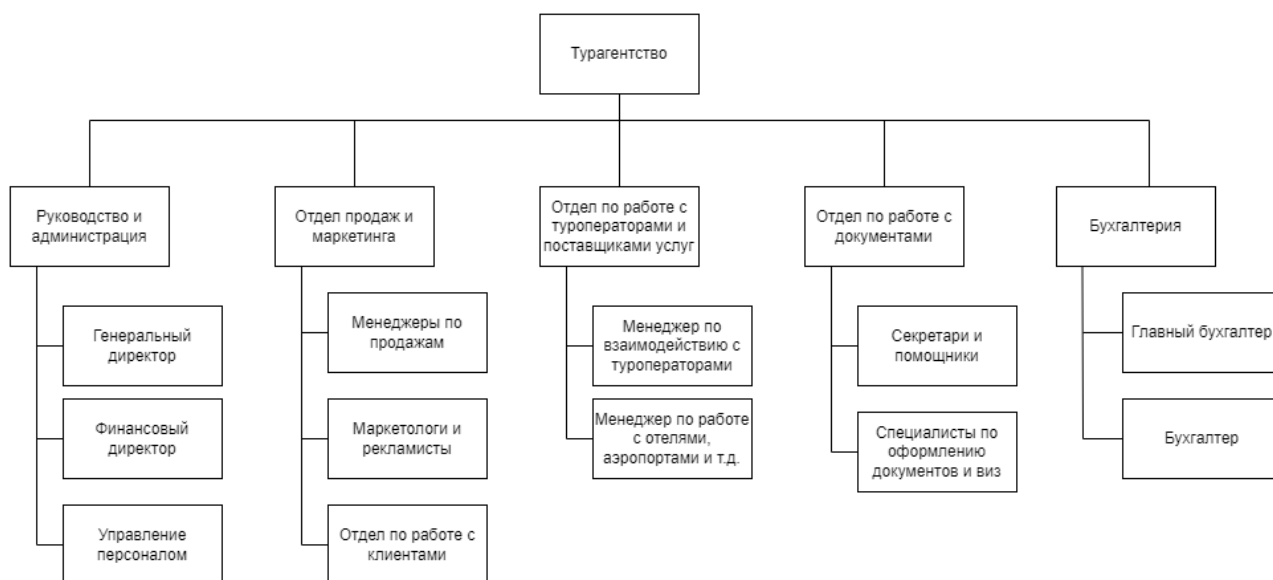


Рисунок 1 – структурная схема предметной области

В предметной области "Турагентство" создаются и обрабатываются различные бумажные и электронные документы. Перечень и описание некоторых из них приведены в таблице 1.

Таблица 1 - перечень и описание структуры бумажных и электронных документов, создаваемых и обрабатываемых в предметной области

Наименование документа	Описание документа
Условия продажи туров	описывают условия и положения, связанные с продажей туров, такие как цены, сроки, гарантии и т. д.
Расчётные счета и договоры о платежах	регулируют процесс оплаты туров и услуг, а также условия возврата денежных средств в случае отмены тура.
Визы и паспорта клиентов	копии документов, необходимых для получения виз и других документов, связанных с путешествием
Электронная корреспонденция и документы	электронные письма, запросы, обращения и другие документы, связанные с коммуникацией
Электронная заявка на туристические услуги	содержит информацию о клиенте, его желаниях и предпочтениях по путешествию. Он используется для определения предлагаемых туров и организации путешествия
Электронная квитанция о оплате	этот документ подтверждает оплату туристических услуг. Он содержит информацию о платеже, банковских данных и других необходимых деталях
Туристическая программа	содержит информацию о туре, включая маршрут, время, места пребывания, виды транспорта и другие необходимые детали

В целом, разработка ИС должна быть направлена на повышение эффективности и качества работы турагентства, улучшение обслуживания клиентов и партнеров, а также на упрощение и автоматизацию процессов. Цели могут включать в себя следующие элементы [1]:

- **автоматизация процессов** - система должна автоматизировать многие процессы, связанные с обработкой заявок, что уменьшит возможность ошибок и повысит точность работы;
- **улучшение качества обслуживания** - ИС должна обеспечить быструю и точную информацию о заявках клиентов, что позволит сотрудникам турагентства быстрее и качественнее обслуживать клиентов;
- **упрощение взаимодействия с клиентами** - ИС должна предоставить клиентам возможность заполнять заявки и получать информацию о своих турах электронным способом, что упростит взаимодействие и улучшит опыт пользования услугами турагентства;
- **аналитика и отчетность** - система должна предоставить различные отчеты и аналитические данные, которые помогут турагентству лучше понимать свой бизнес, определять тренды и принимать информированные решения.

На рисунке 2 показана диаграмма деятельности для информационных задач.

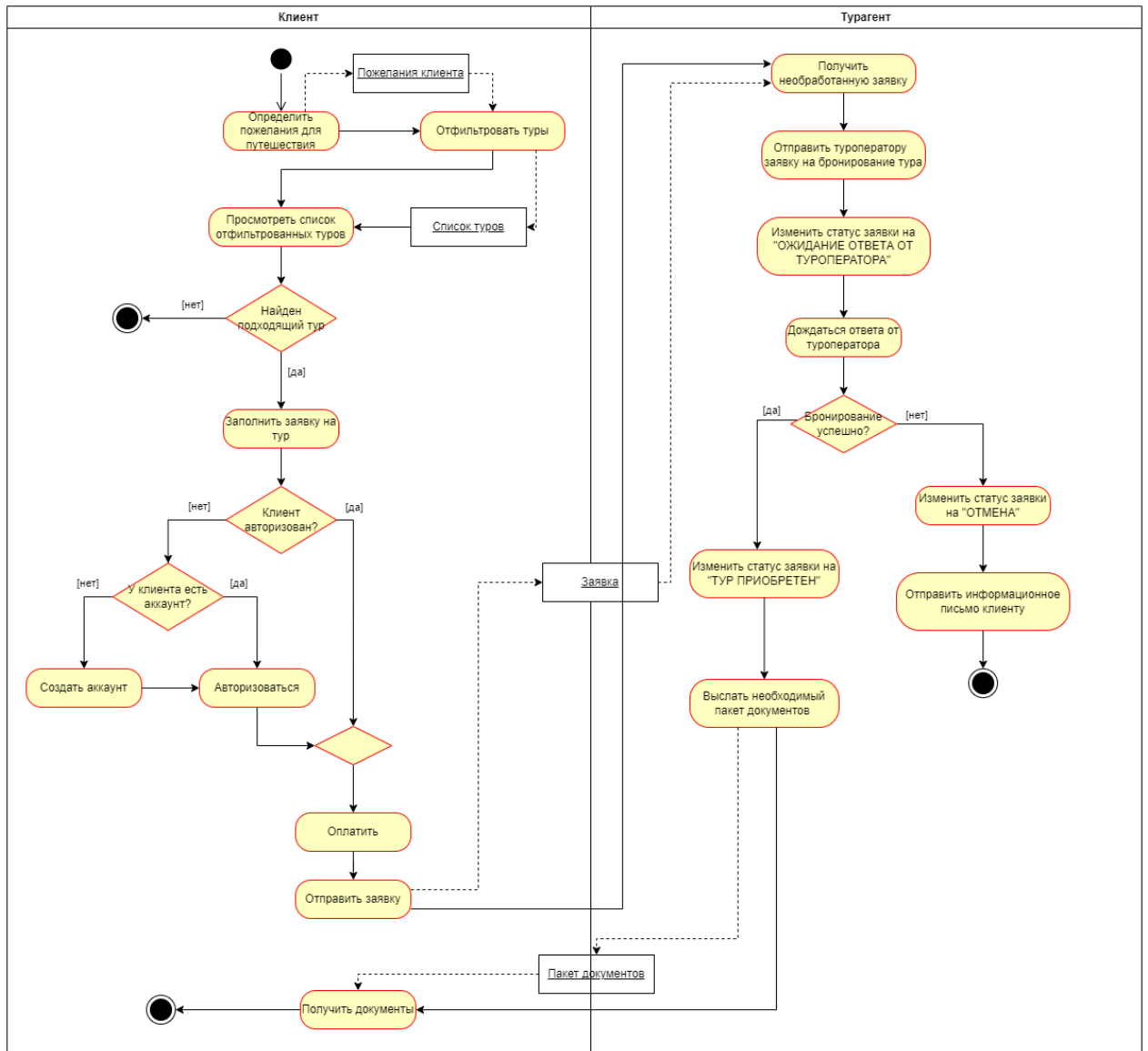


Рисунок 2 - диаграммы деятельности UML

Выделенные категории пользователей и приложения (подсистемы) информационной системы представлены в таблице 2.

Таблица 2 - Категории пользователей информационной системы

№ к. п.	Категория пользователей	Наименование приложения	Автоматизируемые бизнес-процессы	Классы информационных объектов
1	Клиент	Приложение клиента для выбора туристических услуг	Просмотр интересных туров	Тур, Страна, Отель
			Оформление заявки	Турист, Заявка
			Просмотр статуса заявки	Заявка, Статус заявки

Продолжение таблицы 2

2	Турагент	Приложение турагента для учета и анализа заявок клиентов	Просмотр заявок	Заявка, Клиент, Турист
			Изменение статуса заявки	Заявка, Статус заявки
			Получение статистики	Заявка, Тур, Клиент, Обратная связь

В данной курсовой работе будет реализовано приложение турагента для учета и анализа заявок клиентов.

1.2 Моделирование бизнес-процессов предметной области

Спецификация бизнес-процессов предметной области показана в таблице 3.

Таблица 3 - Спецификации бизнес-процессов предметной области

№ п/п	Бизнес-процесс	Исполнитель	Входные данные		Выходные данные		Ограничения
			Поставщик	Содержание	Потребитель	Содержание	
1	Создать аккаунт	Клиент	Клиент	Данные о себе	Система	Данные о клиенте	У одного клиента – один аккаунт
2	Просматривать список туров	Клиент	Система	Список туров	Клиент	Список туров	
3	Фильтровать список туров	Клиент	Клиент	Список фильтров	Клиент	Отфильтрованные туры	
4	Оформить заявку на тур	Клиент	Клиент	Данные о туристах и о себе	Система	Заполненная заявка	Клиент должен быть авторизован в системе
5	Отменить ранее поданную заявку	Клиент	Клиент	Номер заявки	Система	Номер заявки	Заявка с указанным номером должна существовать
6	Просматривать список заявок от клиентов	Турагент	Система	Список заявок	Турагент	Список заявок	
7	Направлять заявку на бронирование тура у туроператора	Турагент	Турагент	Заявка от клиента	Туроператор	Заявка от турагента	
8	Направлять заявку на отмену бронирования тура у туроператора	Турагент	Турагент	Заявка на отмену бронирования тура	Туроператор	Обновленный список туров	Заявка на бронирование должна существовать

Продолжение таблицы 3

9	Изменить состояние заявки	Турагент	Турагент	Заявка	Система	Заявка с измененным статусом	
10	Отправить готовый пакет документов клиенту	Турагент	Турагент	Пакет документов	Клиент	Пакет документов	
11	Получать статистику по продажам путевок в определенные страны	Турагент	Система	Данные для аналитики	Турагент	Статистика	

На рисунке 3 изображена диаграмма вариантов использования.

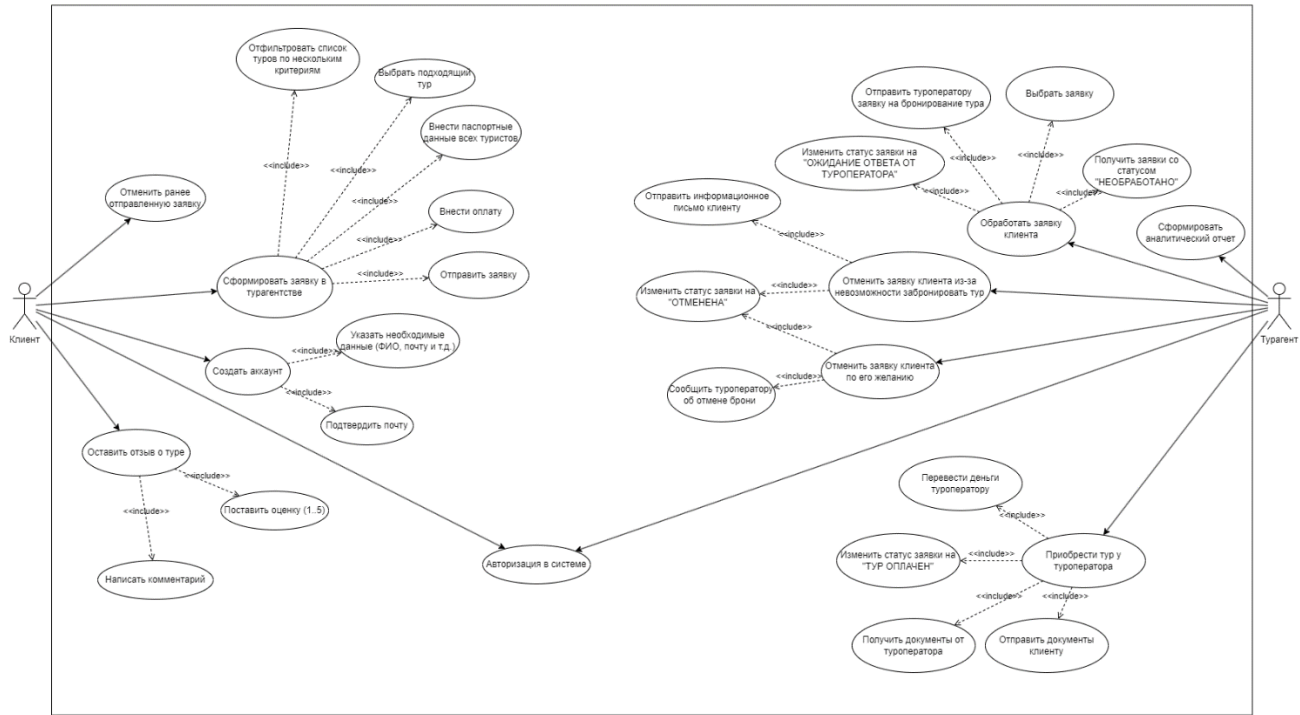


Рисунок 3 - Диаграмма вариантов использования UML

Сценарий варианта использования «Создать аккаунт» представлен на рисунке 4.

Создать аккаунт
Описание: Клиент создает аккаунт
Действующие лица: Клиент
Предусловие: Отсутствие аккаунта
Основной поток: <ol style="list-style-type: none"> 1. Клиент вносит данные о себе 2. Клиент указывает номер телефона и адрес электронной почты 3. Отправляет запрос на создание аккаунта 4. Подтверждает электронную почту
Постусловие: Создан аккаунт для клиента

Рисунок 4 - Сценарий варианта использования «Создать аккаунт»

Сценарий варианта использования «Сформировать заявку в турагентстве» представлен на рисунке 5.

Сформировать заявку в турагентстве
Описание: Система принимает и сохраняет заявку, поступившую от клиента
Действующие лица: Клиент
Предусловие: Клиент должен быть авторизован
Основной поток: <ol style="list-style-type: none"> 1. Клиент фильтрует туры 2. Клиент выбирает понравившийся тур 3. Клиент авторизуется в системе 4. Клиент вносит паспортные данные всех туристов 5. Клиент оплачивает тур 6. Клиент отправляет заявку
Постусловие: Система получает новую заявку со статусом «НЕОБРАБОТАНА»

Рисунок 5 - Сценарий варианта использования «Сформировать заявку в турагентстве»

Сценарий варианта использования «Обработать заявку клиента» представлен на рисунке 6.

Обработать заявку клиента
Описание: Турагент получает заявку клиента и начинает с ней работать
Действующие лица: Турагент
Предусловие: Турагент должен выбрать только необработанные заявки
Основной поток: <ol style="list-style-type: none"> 1. Турагент выбирает заявку 2. Турагент делает заявку на бронирование тура у туроператора 3. Турагент меняет статус заявки на «ОЖИДАНИЕ ОТВЕТА ОТ ТУРОПЕРАТОРА» 4. Система отправляет клиенту информационное сообщение
Постусловие: Заявка изменила статус

Рисунок 6 - Сценарий варианта использования «Обработать заявку клиента»

Сценарий варианта использования «Формирование аналитических отчетов» представлен на рисунке 7.

Формирование аналитических отчетов
Описание: Турагент получает данные в нужном виде для анализа
Действующие лица: Турагент, Система
Предусловие: В системе должно быть необходимое количество данных для анализа
Основной поток: <ol style="list-style-type: none"> 1. Турагент выбирает вид отчета 2. Система получает и агрегирует необходимые данные 3. Система формирует таблицу с данными 4. Система строит график для наглядности
Постусловие: Турагент получил агрегированные данные для анализа

Рисунок 7 - Сценарий варианта использования «Формирование аналитических отчетов»

Полученные результаты являются исходными данными для реализации запросов на выборку и представлений, функций, хранимых процедур и триггеров, как компонентов серверного приложения

1.3 Разработка концептуальной модели базы данных средствами Visual Paradigm

Концептуальная модель данных (КМД) – это общая информационная модель предметной области, охватывающая вопросы классификации, структуризации и семантической целостности (достоверности и согласованности данных) [2].

Концептуальная модель данных разрабатывается независимо от ограничений, вытекающих из модели данных, поддерживаемая той или иной СУБД. По результатам анализа предметной области была построена диаграмма концептуальной модель базы данных, представленная на рисунке 8.

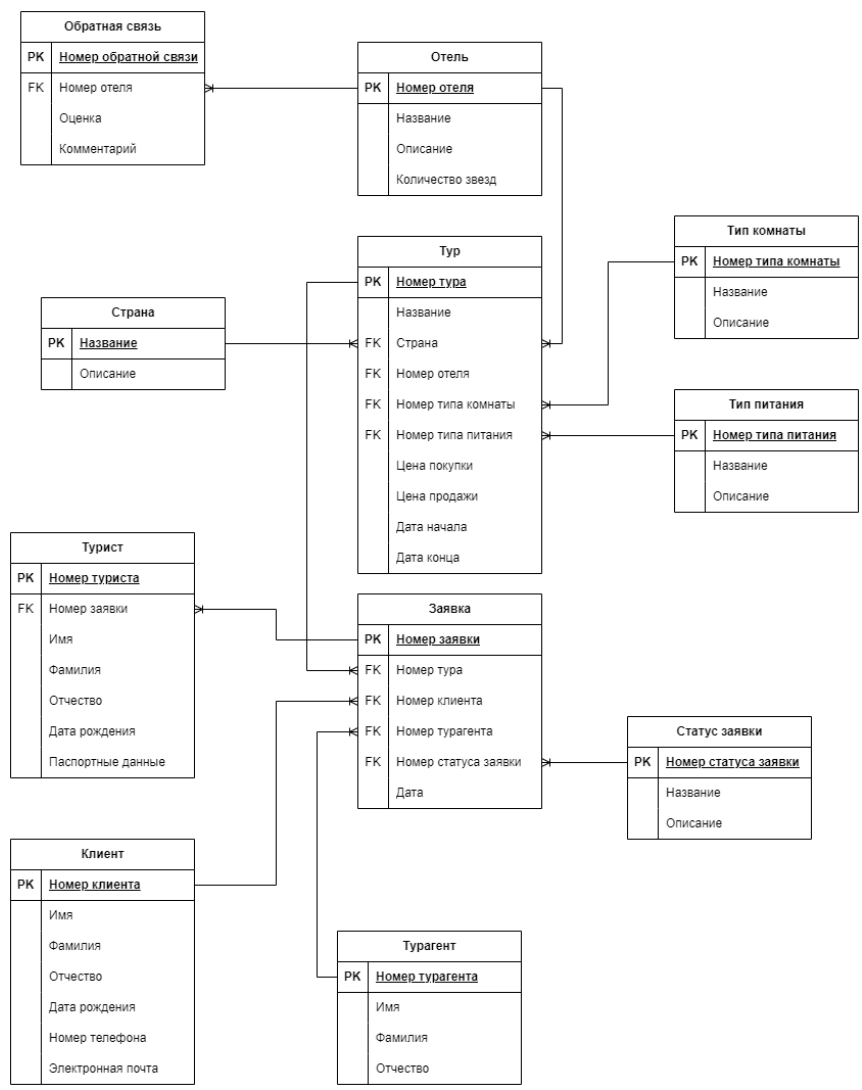


Рисунок 8 - Диаграмма концептуальной модели (схемы) базы данных

В таблице 4 представлена таблица спецификации сущностей

Таблица 4 - Спецификация сущностей

№ п/п	Наименование	Описание	Категория	Количество экземпляров	Режим обновления
1	Клиент	Списочный состав клиентов	Стержневая	Неограниченно	15% в год
2	Турист	Списочный состав Туристов	Стержневая	Неограниченно	15% в год
3	Статус заявки	Виды статусов заявки	Справочная	4	нет
4	Заявка	Списочный состав заявок	Стержневая	Неограниченно	Каждый год
5	Турагент	Списочный состав заявок	Стержневая	Неограниченно	5% в год
8	Тур	Списочный состав туров	Стержневая	Неограниченно	Каждый год
9	Страна	Списочный состав стран	Справочная	30	нет
10	ОТЕЛЬ	Списочный состав отелей	Стержневая	100	нет
11	Обратная связь	Списочный состав обратной связи	Стержневая	Неограниченно	Каждый год
12	Тип комнаты	Виды комнат	Справочная	10	нет
13	Тип питания	Виды питания	Справочная	5	нет

Полученные результаты являются исходными данными для создания логической и физической схем базы данных.

2 Моделирование и развертывание базы данных

2.1 Логическое моделирование базы данных средствами Visual Paradigm

Логическая модель данных является визуальным графическим представлением структур данных, их атрибутов и связей [3]. Логическая модель представляет данные таким образом, чтобы они легко воспринимались бизнес-пользователями. Проектирование логической модели должно быть свободно от требований платформы и языка реализации или способа дальнейшего использования данных.

На рисунке 9 изображена логическая модель базы данных для информационной системы.



Рисунок 9 – Логическая схема базы данных

Для сущностей предметной области были сформированы связи, описание которых приводится в таблице 5.

Таблица 5 - Спецификации связей между сущностями

№ п/п	Сущность 1	Сущность 2	Семантика связи	Мощность	Обязательность	Ограничения
1	Клиент	Заявка	Клиент формирует заявки. Заявка формируется клиентом	1 : M	Необязательная для клиента. Обязательна для заявки	
2	Статус заявки	Заявка	Заявка имеет статус заявки	1 : M	Обязательна для заявки	
3	Турагент	Заявка	Турагент обрабатывает заявки. Заявка обрабатывается турагентом	1 : M	Необязательно для турагента. Обязательно для заявки	
4	Заявка	Турист	Заявка содержит туристов. Турист прикреплен к заявке	1 : M	Обязательно для заявки. Обязательно для туриста.	
5	Тур	Заявка	На тур есть несколько заявок. Заявка оформляется на тур	1 : M	Необязательно для тура. Обязательно для заявки	
6	Страна	Тур	В одной стране может быть несколько туров. Тур проходит в одной стране	1 : M	Обязательно для страны. Обязательно для тура.	
7	Отель	Тур	Отель используется в разных турах. Для тура определен отель	1 : M	Необязательно для отеля. Обязательно для тура	
8	Отель	Обратная связь	На отель оставлена обратная связь. Обратная связь оставлена для отеля	1 : M	Необязательно для отеля. Обязательно для обратной связи	
9	Тип комнаты	Тур	Тур имеет тип комнаты	1 : M	Обязательно для тура	
10	Тип питания	Тур	Тур имеет тип питания	1 : M	Обязательно для тура	

В таблицах 6 – 16 приведены описания всех атрибутов для каждой сущности

Таблица 6 - Спецификации атрибутов сущности «Клиент»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер клиента	Номер клиента	Целый	4	РК	Да	1-99999	
2	Имя	Имя	Символьный	30		Да	Буквы 'А-я'	
3	Фамилия	Фамилия	Символьный	30		Да	Буквы 'А-я'	
4	Отчество	Отчество	Символьный	30		Нет	Буквы 'А-я'	
5	Дата рождения	Дата рождения	Дата	12		Да	уууу/mm/dd	
6	Номер телефона	Номер телефона	Символьный	12		Да	Цифры 0-9 и символ '+'	
7	Электронная почта	Электронная почта	Символьный	64		Да	Буквы 'A-z', Цифры 0-9, Символы ';- @.'	

Таблица 7 - Спецификации атрибутов сущности «Турист»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер туриста	Номер туриста	Целый	4	РК	Да	1-99999	
2	Имя	Имя	Символьный	30		Да	Буквы 'А-я'	
3	Фамилия	Фамилия	Символьный	30		Да	Буквы 'А-я'	
4	Отчество	Отчество	Символьный	30		Нет	Буквы 'А-я'	
5	Дата рождения	Дата рождения	Дата	12		Да	1940.01.01 - 2024.01.01	
6	Паспортные данные	Паспортные данные	Символьный	8		Да	Цифры 0-9	

Таблица 8 - Спецификации атрибутов сущности «Турагент»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер_турагента	Номер турагента	Целый	4	РК	Да	1-9999	
2	Имя	Имя	Символьный	30		Да	Буквы 'А-я'	
3	Фамилия	Фамилия	Символьный	30		Да	Буквы 'А-я'	
4	Отчество	Отчество	Символьный	30		Нет	Буквы 'А-я'	

Таблица 9 - Спецификации атрибутов сущности «Статус заявки»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер статуса заявки	Номер статуса заявки	Целый	4	РК	Да	1-4	
2	Название	Название	Символьный	64		Да	Буквы 'А-я'	
3	Описание	Описание	Символьный	256		Да	Буквы 'А-я'	

Таблица 10 - Спецификации атрибутов сущности «Заявка»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер заявки	Номер заявки	Целый	4	РК	Да	1-99999	
2	Номер тура	Номер тура	Целый	4	FK	Да	1-99999	
3	Номер клиента	Номер клиента	Целый	4	FK	Да	1-99999	
4	Номер турагента	Номер турагента	Целый	4	FK	Да	1-9999	
5	Номер статуса заявки	Номер статуса заявки	Целый	4	FK	Да	1-4	
6	Дата	Дата	Дата	12		Да	1940.01.01 - 2024.01.01	

Таблица 11 - Спецификации атрибутов сущности «Тур»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер тура	Номер тура	Целый	4	PK	Да	1-99999	
2	Страна	Страна	Символьный	64		Да	Буквы 'А-я'	
3	Номер отеля	Номер отеля	Целый	4	FK	Да	1-99999	
4	Номер типа комнаты	Номер типа комнаты	Целый	4	FK	Да	1-20	
5	Номер типа питания	Номер типа питания	Целый	4	FK	Да	1-20	
6	Цена	Цена	Денежный	16		Да	> 0	
7	Дата начала	Дата начала	Дата	12		Да	> 2024.01.01	
8	Дата конца	Дата конца	Дата	12		Да	< 2030.01.01	

Таблица 12 - Спецификации атрибутов сущности «Страна»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Название	Название	Символьный	64	PK	Да	Буквы 'А-я'	
2	Описание	Описание	Символьный	256		Да		

Таблица 13 - Спецификации атрибутов сущности «Тип комнаты»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер типа комнаты	Номер типа комнаты	Целый	4	PK	Да	1-20	
2	Название	Название	Символьный	64		Да		
3	Описание	Описание	Символьный	256		Да		

Таблица 14 - Спецификации атрибутов сущности «Тип питания»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер типа питания	Номер типа питания	Целый	4	PK	Да	1-20	
2	Название	Название	Символьный	64		Да		
3	Описание	Описание	Символьный	256		Да		

Таблица 15 - Спецификации атрибутов сущности «Отель»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер отеля	Номер отеля	Целый	4	PK	Да	1-99999	
2	Название	Название	Символьный	64		Да		
3	Описание	Описание	Символьный	256		Да		
4	Количество звезд	Количество звезд	Целый	4		Да	[1, 5]	

Таблица 16 - Спецификации атрибутов сущности «Обратная связь»

№ п/п	Наименование	Описание	Тип данных	Размер	Роль	Обязательный	Ограничения значений	Значение по умолчанию
1	Номер обратной связи	Номер обратной связи	Целый	4	PK	Да	1-99999	
2	Номер отеля	Номер отеля	Целый	4	FK	Да	1-99999	
3	Оценка	Оценка	Целый	4		Да	[1, 5]	
4	Комментарий	Комментарий	Символьный	256		Да		

Полученные результаты являются исходными данными для развертывания базы данных и разработки приложения информационной системы.

2.2 Физическое моделирование базы данных средствами Visual Paradigm

Физическая модель данных — это реализация логической модели, выполняемая администраторами и разработчиками баз данных [4]. Модель разрабатывается для определенных СУБД, технологий хранения и соединителей данных, чтобы по мере необходимости предоставлять данные через бизнес-системы пользователям. Это конечный результат всех остальных моделей – фактическая реализация массива данных.

На рисунке 10 изображена физическая модель базы данных для информационной системы.

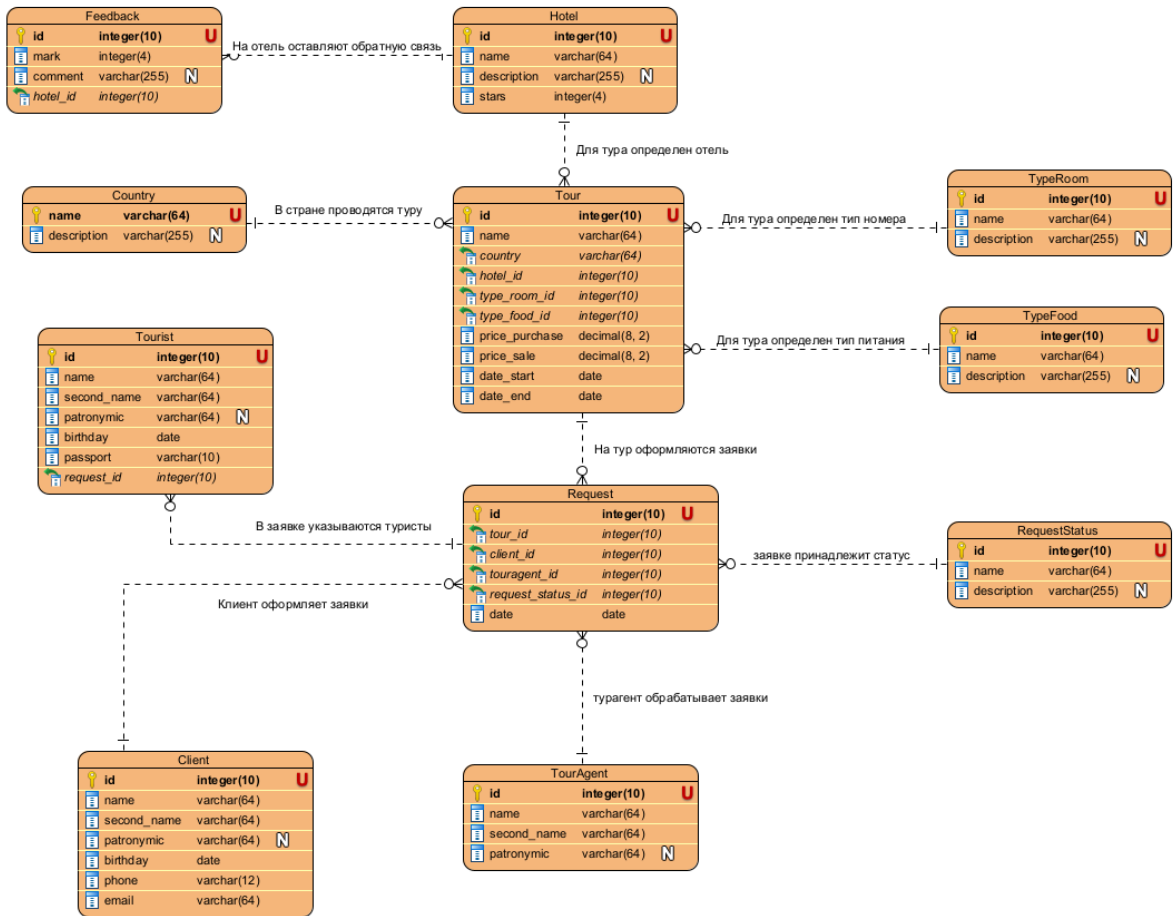


Рисунок 10 – физическая модель базы данных

На основе физической модели данных, а также таблиц, описывающих связи и атрибуты сущностей, будут созданы SQL запросы для развертывания базы данных.

2.3 Развертывание и наполнение базы данных тестовыми данными в среде MS SQL Server

2.3.1 Развертывание базы данных в среде MS SQL Server

Процесс создания базы данных и ее объектов в среде целевой СУБД осуществлялся при помощи скрипта на языке SQL, сгенерированного по физической модели базы данных в среде Visual Paradigm. Скрипты создания базы данных представляются в соответствующем Приложении А.

После создания объектов базы данных, была создана диаграмма данных, представленная на рисунке 11.

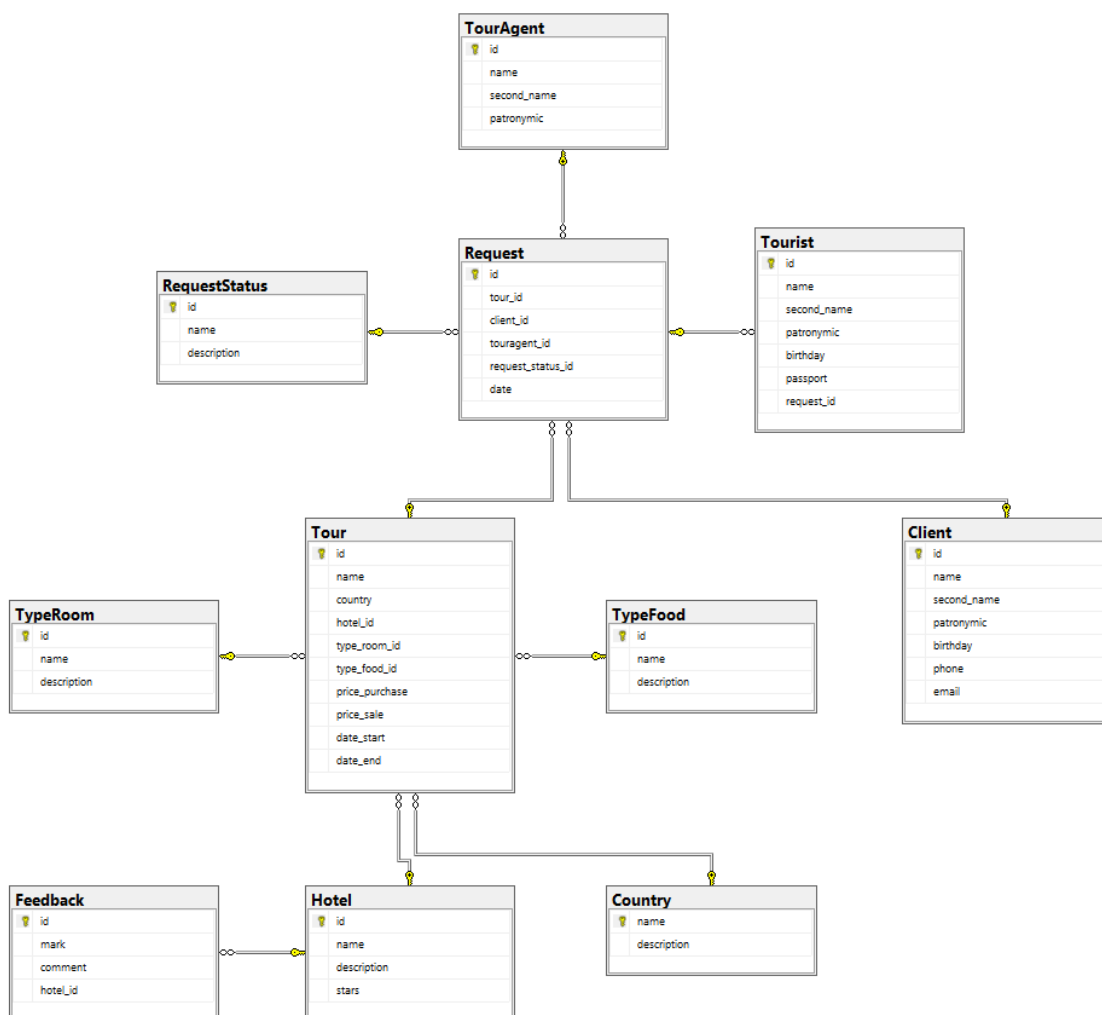


Рисунок 11 - Диаграмма базы данных

Полученные результаты используются в ходе наполнения базы данных и в процессе разработки приложения.

2.3.2 Наполнение базы данных тестовыми данными

Процесс наполнения таблиц базы данных осуществлялся через вызовы соответствующих команд добавления данных в таблицы базы данных.

В листингах ниже приведены скрипты для последовательности вызовов команд добавления данных, скриншоты выполнения и результаты заполнения таблиц (рисунки 12 - 21). Наполненная база данных используется в дальнейшем для отладки и тестирования запросов и представлений, функций и хранимых процедур, триггеров.

Листинг 1 – Заполнение таблицы TypeRoom

```
insert into TypeRoom (name, description) values
('одиночный', 'одноместный номер с одной кроватью подойдет, если тех,
кто путешествует в одиночестве'),
('двойной', 'однокомнатный номер с двуспальной кроватью – отличный
вариант для пары, путешествующей без детей'),
('сдвоенный', 'для проживания двух человек с двумя отдельными
односпальными кроватями. На нем можно остановить выбор в том случае,
если вы заселяетесь в номер, например, с другом/ подружкой, братом /
сестрой или коллегой'),
('тройной', 'семейный вариант для трех человек. В номере могут быть
либо 3 отдельные постели, либо одна большая и односпальная'),
('четырехместный', 'четырехместное размещение для семьи с двумя
детьми'),
('Дополнительная кровать', 'дополнительное спальное место (раскладная
кровать или диван), которое по запросу может быть установлено в любом
типе номера');
```

Results			
Messages			
	id	name	description
1	1	одиночный	одноместный номер с одной кроватью подойдет, если ...
2	2	двойной	однокомнатный номер с двуспальной кроватью – отли...
3	3	сдвоенный	для проживания двух человек с двумя отдельными одн...
4	4	тройной	семейный вариант для трех человек. В номере могут б...
5	5	четырехместный	четырехместное размещение для семьи с двумя детьми
6	6	Дополнительная кровать	дополнительное спальное место (раскладная кровать ...

Рисунок 12 – Заполненная таблица TypeRoom

Листинг 2 – Заполнение таблицы TypeFood

```
insert into TypeFood (name, description) values
('завтраки', 'один из самых распространенных вариантов для туристов,
который подразумевает только завтрак'),
```


('полупансион', 'питание по типу 'полупансион', включающий завтрак и ужин. Такой тип не предполагает спиртных напитков, перекусов и обеда. Алкогольные напитки туристы могут заказать на месте или в номер с оплатой на месте'),
 ('полный пансион', 'предполагает трехразовое питание в формате шведского стола, с разнообразием салатов, первых и вторых блюд, рыбных и овощных продуктов, а также безалкогольные напитки'),
 ('тройной', 'семейный вариант для трех человек. В номере могут быть либо 3 отдельные постели, либо одна большая и односпальная'),
 ('все включено', 'завтрак-второй завтрак-обед-полдник-ужин + алкоголь и закуски');

	id	name	description
1	20	завтраки	один из самых распространенных вариантов для тури...
2	21	полупансион	питание по типу «полупансион», включающий завтрак...
3	22	полный пансион	предполагает трехразовое питание в формате шведс...
4	23	все включено	завтрак-второй завтрак-обед-полдник-ужин + алкоголь...

Рисунок 13 – Заполненная таблица TypeFood

Листинг 3 – Заполнение таблицы Country

```
INSERT INTO Country (name, description)
VALUES
('Испания', 'трансконтинентальное суверенное государство в Южной Европе'),
('Франция', 'Республика в Западной Европе'),
('Кипр', 'Островное государство, расположенное в восточной части Средиземного моря'),
('Куба', 'Островное государство в Латинской Америке, в Карибском бассейне'),
('Турция', 'Турция известна, прежде всего, как один из самых востребованных курортов. Здесь не только красивые пляжи и уникальная природа');
```

	name	description
1	Испания	трансконтинентальное суверенное государство в Юж...
2	Кипр	Островное государство, расположенное в восточной ...
3	Куба	Островное государство в Латинской Америке, в Кари...
4	Турция	Турция известна, прежде всего, как один из самых в...
5	Франция	Республика в Западной Европе

Рисунок 14 – Заполненная таблица Country

Листинг 4 - Заполнение таблицы Hotel

```
INSERT INTO Hotel (name, description, stars)
VALUES
```

```
( 'ОТЕЛЬ АЛЫЕ ПАРУСА', 'Прекрасный отель на берегу моря', 4),
( 'Гостиница Центральная', 'Уютное место в самом центре города', 3),
( 'ОТЕЛЬ ЗОЛОТАЯ ДОЛИНА', 'Отличный выбор для любителей природы', 5),
( 'Гостевой дом Подкова', 'Тихое уединенное место в горах', 2),
( 'ОТЕЛЬ ЛУННЫЙ СВЕТ', 'Идеальное место для романтического уикенда',
4),
( 'Мини отель Радуга', 'Приветливый персонал и уютная атмосфера', 3),
( 'ОТЕЛЬ ЗВЕЗДНЫЙ ГОРОД', 'Для ценителей науки и космоса', 5),
( 'Гостиница Полярная Ночь', 'Уникальный опыт видеть полярное сияние',
4),
( 'ОТЕЛЬ СОЛНЕЧНЫЙ БЕРЕГ', 'Идеальное место для семейного отдыха', 4),
( 'Гостиница Веселый Хомяк', 'Для любителей домашнего уюта', 2);
```

Results		Messages		
	id	name	description	stars
1	7	ОТЕЛЬ АЛЫЕ ПАРУСА	Прекрасный отель на берегу моря	4
2	8	Гостиница Центральная	Уютное место в самом центре города	3
3	9	ОТЕЛЬ ЗОЛОТАЯ ДОЛИНА	Отличный выбор для любителей природы	5
4	10	Гостевой дом Подкова	Тихое уединенное место в горах	5
5	11	ОТЕЛЬ ЛУННЫЙ СВЕТ	Идеальное место для романтического уикенда	4
6	12	Мини отель Радуга	Приветливый персонал и уютная атмосфера	3
7	13	ОТЕЛЬ ЗВЕЗДНЫЙ ГОРОД	Для ценителей науки и космоса	5
8	14	Гостиница Полярная Ночь	Уникальный опыт видеть полярное сияние	4
9	15	ОТЕЛЬ СОЛНЕЧНЫЙ БЕРЕГ	Идеальное место для семейного отдыха	4
10	16	Гостиница Веселый Хомяк	Для любителей домашнего уюта	3

Рисунок 15 – Заполненная таблица Hotel

Листинг 5 - Заполнение таблицы Feedback

```
INSERT INTO Feedback (mark, comment, hotel_id)
VALUES
(4, 'Отличный сервис!', 7),
(5, 'Прекрасный вид из окна', 8),
(3, 'Немного шумно ночью', 9),
(2, 'Не понравился завтрак', 10),
(4, 'Уютные номера', 11),
(5, 'Приветливый персонал', 12),
(3, 'Среднее качество обслуживания', 13),
(4, 'Отличное расположение', 14),
(5, 'Идеально для семейного отдыха', 15),
(1, 'Неудобные кровати', 16),
(5, 'Очень чисто!', 7),
(4, 'Отличный выбор завтраков', 8),
(3, 'Нужно улучшить качество Wi-Fi', 9),
(2, 'Проблемы с кондиционером', 10),
(4, 'Прекрасный бассейн', 11),
(5, 'Шикарный интерьер', 12),
(3, 'Долгое ожидание в ресторане', 13),
(4, 'Уютные номера с видом на город', 14),
(5, 'Отличный детский клуб', 15),
(1, 'Не работал телевизор', 16),
(4, 'Отличный сервис!', 7),
```

(5, 'Отличный опыт', 8),
 (3, 'Комната нуждается в улучшении', 9),
 (2, 'Персонал был недружелюбен', 10),
 (5, 'Настоятельно рекомендуется!', 11),
 (4, 'Хорошее соотношение цены и качества', 12),
 (3, 'Средний срок пребывания', 13),
 (1, 'Ужасный опыт', 14),
 (5, 'Приеду еще раз', 15),
 (4, 'Дружелюбный персонал', 16),
 (3, 'Требуется лучшая чистота', 7),
 (2, 'Плохие удобства', 8),
 (5, 'Очень понравился вид', 9),
 (4, 'Комфортный отдых', 10),
 (3, 'Шумная обстановка', 11),
 (1, 'Разочаровывающее пребывание', 12),
 (5, 'Потрясающее место!', 13),
 (4, 'Чистые помещения', 14),
 (3, 'Могло быть лучше', 15),
 (2, 'Неудовлетворительное обслуживание', 16),
 (5, 'Незабываемый опыт', 7),
 (4, 'Доволен обслуживанием', 8),
 (3, 'Средние удобства', 9),
 (1, 'Избегайте этого места', 10),
 (5, 'Исключительный сервис', 11),
 (4, 'Ухоженная недвижимость', 12),
 (3, 'Не соответствует ожиданиям', 13),
 (2, 'Плохо управляется', 14),
 (5, 'Идеальный отдых!', 15),
 (4, 'Внимательный персонал', 16),
 (3, 'Ограниченные возможности', 7),
 (1, 'Худший отель на свете', 8),
 (5, 'Выдающийся опыт', 9),
 (4, 'Удовлетворительное пребывание', 10),
 (3, 'Чистота могла бы улучшиться', 11),
 (2, 'Разочарован комнатой', 12),
 (5, 'Рекомендую другим', 13),
 (4, 'Быстрое обслуживание', 14),
 (3, 'Требуется ремонт', 15),
 (1, 'Ужасный опыт', 16),
 (5, 'Невероятный отдых!', 7),
 (4, 'Вежливый персонал', 8),
 (3, 'Средний сервис', 9),
 (2, 'Неприятное пребывание', 10),
 (5, 'Прекрасная атмосфера', 11),
 (4, 'Просторные комнаты', 12),
 (3, 'Не лучший выбор', 13),
 (1, 'Сожалею, что остался здесь', 14),
 (5, 'Незабываемый опыт', 15),
 (4, 'Эффективный персонал', 16),
 (3, 'Можно было бы больше удобств', 7),
 (2, 'Ниже ожиданий', 8),
 (5, 'Идеальный отдых!', 9),
 (4, 'Хорошее расположение', 10),
 (3, 'Честное служение', 11),
 (1, 'Ужасное пребывание', 12),
 (5, 'Фантастический опыт', 13),
 (4, 'Комфортные номера', 14),

```
(3, 'Средний стаж', 15),
(2, 'Неудовлетворительное пребывание', 16);
```

Results		Messages		
	id	mark	comment	hotel_id
7	8	3	Среднее качество обслуживания	13
8	9	4	Отличное расположение	14
9	10	5	Идеально для семейного отдыха	15
10	11	1	Неудобные кровати	16
11	12	5	Очень чисто!	7
12	13	4	Отличный выбор завтраков	8
13	14	3	Нужно улучшить качество Wi-Fi	9
14	15	2	Проблемы с кондиционером	10
15	16	4	Прекрасный бассейн	11
16	17	5	Шикарный интерьер	12
17	18	3	Долгое ожидание в ресторане	13
18	19	4	Уютные номера с видом на город	14
19	20	5	Отличный детский клуб	15
20	21	1	Не работал телевизор	16
21	22	4	Отличный сервис!	7
22	23	5	Отличный опыт	8
23	24	3	Комната нуждается в улучшении	9
24	25	2	Персонал был недружелюбен	10
25	26	5	Настоятельно рекомендуется!	11
26	27	4	Хорошее соотношение цены и к...	12
27	28	3	Средний срок пребывания	13
28	29	1	Ужасный опыт	14
29	30	5	Приеду еще раз	15
30	31	4	Дружелюбный персонал	16
31	32	3	Требуется лучшая чистота	7

Рисунок 16 – Заполненная таблица Feedback

Листинг 6 - Заполнение таблицы Client

```
INSERT INTO Client (name, second_name, patronymic, birthday, phone,
email)
VALUES
('Иван', 'Иванов', 'Иванович', '1990-05-15', '+79991234567',
'ivanov@mail.com'),
('Петр', 'Петров', 'Петрович', '1985-10-20', '+79992223344',
'petrov@mail.com'),
('Мария', 'Сидорова', 'Александровна', '1988-03-25', '+79993334455',
'sidorova@mail.com'),
('Елена', 'Козлова', 'Владимировна', '1995-08-12', '+79994445566',
'kozlova@mail.com'),
('Алексей', 'Смирнов', 'Андреевич', '1982-01-30', '+79995556677',
'smirnov@mail.com'),
('Ольга', 'Игнатьева', 'Сергеевна', '1987-06-07', '+79996667788',
'ignatieva@mail.com'),
('Виктор', 'Васильев', 'Игоревич', '1993-11-18', '+79997778899',
'vasiliev@mail.com'),
('Наталья', 'Лебедева', 'Петровна', '1984-04-22', '+79998889900',
'lebedeva@mail.com'),
```



```
( 'Дмитрий', 'Кузнецов', 'Максимович', '1991-09-03', '+79990001122',
'kuznetsov@mail.com'),
( 'Анна', 'Федорова', 'Алексеевна', '1989-12-10', '+79991112233',
'fedorova@mail.com'),
( 'Сергей', 'Морозов', 'Дмитриевич', '1986-07-14', '+79992223344',
'morozov@mail.com'),
( 'Евгения', 'Николаева', 'Ивановна', '1996-02-28', '+79993334455',
'nikolaeva@mail.com'),
( 'Владимир', 'Павлов', 'Сергеевич', '1983-05-05', '+79994445566',
'pavlov@mail.com'),
( 'Алина', 'Соловьева', 'Олеговна', '1992-10-09', '+79995556677',
'solovieva@mail.com'),
( 'Артем', 'Григорьев', 'Владимирович', '1981-03-17', '+79996667788',
'grigoriev@mail.com'),
( 'Юлия', 'Тимофеева', 'Анатолевна', '1988-08-23', '+79997778899',
'timofeeva@mail.com'),
( 'Максим', 'Беляков', 'Игоревич', '1994-01-27', '+79998889900',
'belyakov@mail.com'),
( 'Екатерина', 'Мельникова', 'Денисовна', '1990-06-02', '+79990001122',
'melnikova@mail.com'),
( 'Артур', 'Семенов', 'Александрович', '1987-11-12', '+79991112233',
'semenov@mail.com'),
( 'Людмила', 'Калинина', 'Александровна', '1985-04-15', '+79992223344',
'kalinina@mail.com');
```

Results		Messages					
	id	name	second_name	patronymic	birthday	phone	email
1	1	Иван	Иванов	Иванович	1990-05-15 00:00:00.000	+79991234567	ivanov@mail.com
2	2	Петр	Петров	Петрович	1985-10-20 00:00:00.000	+79992223344	petrov@mail.com
3	3	Мария	Сидорова	Александровна	1988-03-25 00:00:00.000	+79993334455	sidorova@mail.com
4	4	Елена	Козлова	Владимировна	1995-08-12 00:00:00.000	+79994445566	kozlova@mail.com
5	5	Алексей	Смирнов	Андреевич	1982-01-30 00:00:00.000	+79995556677	smimov@mail.com
6	6	Ольга	Игнатьева	Сергеевна	1987-06-07 00:00:00.000	+79996667788	ignatieva@mail.com
7	7	Виктор	Васильев	Игоревич	1993-11-18 00:00:00.000	+79997778899	vasiliev@mail.com
8	8	Наталья	Лебедева	Петровна	1984-04-22 00:00:00.000	+79998889900	lebedeva@mail.com
9	9	Дмитрий	Кузнецов	Максимович	1991-09-03 00:00:00.000	+79990001122	kuznetsov@mail.com
10	10	Анна	Федорова	Алексеевна	1989-12-10 00:00:00.000	+79991112233	fedorova@mail.com
11	11	Сергей	Морозов	Дмитриевич	1986-07-14 00:00:00.000	+79992223344	morozov@mail.com
12	12	Евгения	Николаева	Ивановна	1996-02-28 00:00:00.000	+79993334455	nikolaeva@mail.com
13	13	Владимир	Павлов	Сергеевич	1983-05-05 00:00:00.000	+79994445566	pavlov@mail.com
14	14	Алина	Соловьева	Олеговна	1992-10-09 00:00:00.000	+79995556677	solovieva@mail.com
15	15	Артем	Григорьев	Владимирович	1981-03-17 00:00:00.000	+79996667788	grigoriev@mail.com
16	16	Юлия	Тимофеева	Анатолевна	1988-08-23 00:00:00.000	+79997778899	timofeeva@mail.com
17	17	Максим	Беляков	Игоревич	1994-01-27 00:00:00.000	+79998889900	belyakov@mail.com
18	18	Екатерина	Мельникова	Денисовна	1990-06-02 00:00:00.000	+79990001122	melnikova@mail.com
19	19	Артур	Семенов	Александрович	1987-11-12 00:00:00.000	+79991112233	semenov@mail.com
20	20	Людмила	Калинина	Александровна	1985-04-15 00:00:00.000	+79992223344	kalinina@mail.com

Рисунок 17 – Заполненная таблица Client

Листинг 7 - Заполнение таблицы TourAgent

```
INSERT INTO TourAgent (name, second_name, patronymic)
VALUES
( 'Иван', 'Иванов', 'Иванович'),
( 'Петр', 'Петров', 'Петрович'),
( 'Мария', 'Сидорова', 'Александровна'),
( 'Елена', 'Козлова', 'Владимировна'),
( 'Алексей', 'Смирнов', 'Андреевич');
```

Results		Messages		
	id	name	second_name	patronymic
1	1	Иван	Иванов	Иванович
2	2	Петр	Петров	Петрович
3	3	Мария	Сидорова	Александровна
4	4	Елена	Козлова	Владимировна
5	5	Алексей	Смирнов	Андреевич

Рисунок 18 – Заполненная таблица TourAgent

Листинг 8 - Заполнение таблицы RequestStatus

```

INSERT INTO RequestStatus (name, description)
VALUES
('НЕОБРАБОТАНА', 'Новая заявка, поступившая в систему'),
('ОЖИДАНИЕ ОТ ТУРОПЕРАТОРА', 'Турагент ожидает ответа от
туроператора'),
('ОТМЕНЕНА', 'Заявка отменена'),
('ОПЛАЧЕНА', 'Заявка оплачена');

```

Results		Messages	
	id	name	description
1	1	НЕОБРАБОТАНА	Новая заявка, поступившая в систему
2	2	ОЖИДАНИЕ ОТ ТУРОПЕРАТОРА	Турагент ожидает ответа от туроператора
3	3	ОТМЕНЕНА	Заявка отменена
4	4	ОПЛАЧЕНА	Заявка оплачена

Рисунок 19 – Заполненная таблица RequestStatus

Листинг 9 - Заполнение таблицы Tour

```

INSERT INTO Tour (name, country, hotel_id, type room_id, type food_id,
price_purchase, price_sale, date_start, date_end)
VALUES
('Отдых на море', 'Испания', 7, 1, 20, 150000.00, 200000.00,
'20240615', '20240630'),
('Экскурсионный тур', 'Франция', 8, 2, 21, 180000.00, 220000.00,
'20240710', '20240725'),
('Спортивный тур', 'Испания', 9, 3, 22, 120000.00, 150000.00,
'20240805', '20240820'),
('Гастрономический тур', 'Куба', 10, 4, 23, 250000.00, 300000.00,
'20240901', '20240915'),
('Приключенческий тур', 'Куба', 11, 5, 20, 220000.00, 260000.00,
'20241010', '20241025'),
('Спа туры', 'Франция', 12, 6, 21, 180000.00, 210000.00, '20241105',
'20241120'),
('Культурный тур', 'Турция', 13, 1, 22, 160000.00, 190000.00,
'20241201', '20241215'),
('Экстрим туры', 'Франция', 14, 2, 23, 200000.00, 240000.00,
'20250110', '20250125'),

```

```

('Лыжные туры', 'Турция', 15, 3, 21, 230000.00, 270000.00, '20250205',
'20250220'),
('Круизы', 'Кипр', 16, 4, 23, 280000.00, 320000.00, '20250301',
'20250315'),
('Отдых на море', 'Испания', 7, 5, 22, 150000.00, 200000.00,
'20240715', '20240730'),
('Экскурсионный тур', 'Франция', 8, 6, 23, 180000.00, 220000.00,
'20240710', '20240725'),
('Спортивный тур', 'Испания', 9, 1, 20, 120000.00, 150000.00,
'20240805', '20240820'),
('Гастрономический тур', 'Куба', 10, 2, 21, 250000.00, 300000.00,
'20240901', '20240915'),
('Приключенческий тур', 'Куба', 11, 3, 22, 220000.00, 260000.00,
'20241010', '20241025'),
('Спа туры', 'Франция', 12, 4, 23, 180000.00, 210000.00, '20241105',
'20241120'),
('Культурный тур', 'Турция', 13, 5, 21, 160000.00, 190000.00,
'20241201', '20241215'),
('Экстрим туры', 'Франция', 14, 6, 20, 200000.00, 240000.00,
'20250110', '20250125'),
('Лыжные туры', 'Турция', 15, 1, 23, 230000.00, 270000.00, '20250205',
'20250220'),
('Круизы', 'Кипр', 16, 2, 22, 180000.00, 220000.00, '20250301',
'20250315'),
('Отдых на море', 'Испания', 7, 3, 23, 150000.00, 200000.00,
'20240715', '20240730'),
('Экскурсионный тур', 'Франция', 8, 4, 20, 180000.00, 220000.00,
'20240710', '20240725'),
('Спортивный тур', 'Испания', 9, 5, 21, 120000.00, 150000.00,
'20240805', '20240820'),
('Гастрономический тур', 'Куба', 10, 6, 22, 250000.00, 300000.00,
'20240901', '20240915'),
('Приключенческий тур', 'Куба', 11, 1, 23, 220000.00, 260000.00,
'20241010', '20241025'),
('Спа туры', 'Франция', 12, 2, 20, 180000.00, 210000.00, '20241105',
'20241120'),
('Культурный тур', 'Турция', 13, 3, 21, 160000.00, 190000.00,
'20241201', '20241215'),
('Экстрим туры', 'Франция', 14, 4, 22, 200000.00, 240000.00,
'20250110', '20250125'),
('Лыжные туры', 'Турция', 15, 5, 23, 230000.00, 270000.00, '20250205',
'20250220'),
('Круизы', 'Кипр', 16, 6, 20, 180000.00, 220000.00, '20250301',
'20250315');

```

Results Messages

	id	name	country	hotel_id	type_room_id	type_food_id	price_purchase	price_sale	date_start	date_end
1	1	Отдых на море	Испания	7	1	20	150000.00	200000.00	2024-06-15 00:00:00.000	2024-06-30 00:00:00.000
2	2	Экскурсионный тур	Франция	8	2	21	180000.00	220000.00	2024-07-10 00:00:00.000	2024-07-25 00:00:00.000
3	3	Спортивный тур	Испания	9	3	22	120000.00	150000.00	2024-08-05 00:00:00.000	2024-08-20 00:00:00.000
4	4	Гастрономический тур	Куба	10	4	23	250000.00	300000.00	2024-09-01 00:00:00.000	2024-09-15 00:00:00.000
5	5	Приключенческий тур	Куба	11	5	20	220000.00	260000.00	2024-10-10 00:00:00.000	2024-10-25 00:00:00.000
6	6	Спа туры	Франция	12	6	21	180000.00	210000.00	2024-11-05 00:00:00.000	2024-11-20 00:00:00.000
7	7	Культурный тур	Турция	13	1	22	160000.00	190000.00	2024-12-01 00:00:00.000	2024-12-15 00:00:00.000
8	8	Экстрим туры	Франция	14	2	23	200000.00	240000.00	2025-01-10 00:00:00.000	2025-01-25 00:00:00.000
9	9	Лыжные туры	Турция	15	3	21	230000.00	270000.00	2025-02-05 00:00:00.000	2025-02-20 00:00:00.000
10	10	Круизы	Кипр	16	4	23	280000.00	320000.00	2025-03-01 00:00:00.000	2025-03-15 00:00:00.000
11	11	Отдых на море	Испания	7	5	22	150000.00	200000.00	2024-07-15 00:00:00.000	2024-07-30 00:00:00.000
12	12	Экскурсионный тур	Франция	8	6	23	180000.00	220000.00	2024-07-10 00:00:00.000	2024-07-25 00:00:00.000
13	13	Спортивный тур	Испания	9	1	20	120000.00	150000.00	2024-08-05 00:00:00.000	2024-08-20 00:00:00.000
14	14	Гастрономический тур	Куба	10	2	21	250000.00	300000.00	2024-09-01 00:00:00.000	2024-09-15 00:00:00.000
15	15	Приключенческий тур	Куба	11	3	22	220000.00	260000.00	2024-10-10 00:00:00.000	2024-10-25 00:00:00.000
16	16	Спа туры	Франция	12	4	23	180000.00	210000.00	2024-11-05 00:00:00.000	2024-11-20 00:00:00.000
17	17	Культурный тур	Турция	13	5	21	160000.00	190000.00	2024-12-01 00:00:00.000	2024-12-15 00:00:00.000
18	18	Экстрим туры	Франция	14	6	20	200000.00	240000.00	2025-01-10 00:00:00.000	2025-01-25 00:00:00.000
19	19	Лыжные туры	Турция	15	1	23	230000.00	270000.00	2025-02-05 00:00:00.000	2025-02-20 00:00:00.000
20	20	Круизы	Кипр	16	2	22	180000.00	220000.00	2025-03-01 00:00:00.000	2025-03-15 00:00:00.000
21	21	Отдых на море	Испания	7	3	23	150000.00	200000.00	2024-07-15 00:00:00.000	2024-07-30 00:00:00.000
22	22	Экскурсионный тур	Франция	8	4	20	180000.00	220000.00	2024-07-10 00:00:00.000	2024-07-25 00:00:00.000
23	23	Спортивный тур	Испания	9	5	21	120000.00	150000.00	2024-08-05 00:00:00.000	2024-08-20 00:00:00.000
24	24	Гастрономический тур	Куба	10	6	22	250000.00	300000.00	2024-09-01 00:00:00.000	2024-09-15 00:00:00.000
25	25	Приключенческий тур	Куба	11	1	23	220000.00	260000.00	2024-10-10 00:00:00.000	2024-10-25 00:00:00.000

Рисунок 20 – Заполненная таблица Tour

Листинг 10 - Заполнение таблицы Request

```
INSERT INTO Request(tour_id, client_id, touragent_id,
request_status_id, date)
VALUES
(1, 1, 1, 1, '20240501'),
(2, 2, 2, 1, '20240502'),
(3, 3, 3, 1, '20240503'),
(4, 4, 4, 1, '20240504'),
(5, 5, 5, 1, '20240505'),
(6, 6, 1, 1, '20240506'),
(7, 7, 2, 1, '20240507'),
(8, 8, 3, 1, '20240508'),
(9, 9, 4, 1, '20240509'),
(10, 10, 5, 1, '20240510'),
(11, 11, 1, 1, '20240511'),
(12, 12, 2, 1, '20240512'),
(13, 13, 3, 2, '20240513'),
(14, 14, 4, 2, '20240514'),
(15, 15, 5, 2, '20240515'),
(16, 16, 1, 2, '20240516'),
(17, 17, 2, 2, '20240517'),
(18, 18, 3, 2, '20240518'),
(19, 19, 4, 2, '20240519'),
(20, 20, 5, 4, '20240520'),
(21, 1, 1, 4, '20240521'),
(22, 2, 2, 4, '20240522'),
(23, 3, 3, 4, '20240523'),
(24, 4, 4, 4, '20240524'),
(25, 5, 5, 4, '20240525'),
(26, 6, 1, 4, '20240526'),
```

```
(27, 7, 2, 4, '20240527'),  
(28, 8, 3, 4, '20240528'),  
(29, 9, 4, 4, '20240529'),  
(30, 10, 5, 3, '20240530');
```

Results		Messages				
	id	tour_id	client_id	touragent_id	request_status_id	date
1	1	1	1	1	1	2024-05-01 00:00:00.000
2	2	2	2	2	1	2024-05-02 00:00:00.000
3	3	3	3	3	1	2024-05-03 00:00:00.000
4	4	4	4	4	1	2024-05-04 00:00:00.000
5	5	5	5	5	1	2024-05-05 00:00:00.000
6	6	6	6	1	1	2024-05-06 00:00:00.000
7	7	7	7	2	1	2024-05-07 00:00:00.000
8	8	8	8	3	1	2024-05-08 00:00:00.000
9	9	9	9	4	1	2024-05-09 00:00:00.000
10	10	10	10	5	1	2024-05-10 00:00:00.000
11	11	11	11	1	1	2024-05-11 00:00:00.000
12	12	12	12	2	1	2024-05-12 00:00:00.000
13	13	13	13	3	2	2024-05-13 00:00:00.000
14	14	14	14	4	2	2024-05-14 00:00:00.000
15	15	15	15	5	2	2024-05-15 00:00:00.000
16	16	16	16	1	2	2024-05-16 00:00:00.000
17	17	17	17	2	2	2024-05-17 00:00:00.000
18	18	18	18	3	2	2024-05-18 00:00:00.000
19	19	19	19	4	2	2024-05-19 00:00:00.000

Рисунок 21 – Заполненная таблица Request

Наполненная база данных используется в дальнейшем для отладки и тестирования запросов и представлений, функций и хранимых процедур, триггеров.

3 Разработка и тестирование приложения

3.1 Разработка серверной части приложения

3.1.1 Разработка запросов к базе данных и представлений для реализации аналитических бизнес-процессов

Для приложения были разработаны запросы для реализации аналитических бизнес-процессов. Спецификации запросов приведены в таблице 17.

Таблица 17 - Спецификации запросов к базе данных

№ п/п	Содержание	Используемые таблицы	Параметры группировки	Параметры выбора данных
1	Получить список туров в заданную страну и в отель с заданным количеством звезд	Tour, Hotel, TypeFood, TypeRoom		Название страны, Количество звезд в отеле (точное количество или интервал)
2	Получить список отелей, отсортированный по средней оценке в порядке убывания	Feedback, Hotel	Название отеля	
3	Получить количество туров в определенные страны, а также минимальную и максимальную цену на тур	Tour	Название страны	
4	Получить список оплаченных туров за заданный временной интервал	Request, RequestStatus, Tour		Временной интервал

Для приложения были разработаны представления. Спецификации запросов на их создание приведены в таблице 18.

Таблица 18 - Спецификации представлений

№ п/п	Имя представления	Назначение	Используемые таблицы	Параметры
1	AllRequestWithFullInformation	Получить все заявки с полной и удобно читаемой информацией	Request	ИдентификаторЗаявки, НазваниеТура, ФИО_Клиента, ФИО_Турагента, СтатусЗаявки, Дата
2	SaledToursByCountry	Получить количество проданных туров в определенные страны и их общую выручку в этой стране	Request, RequestStatus, Tour	НазваниеСтраны, ОбщаяВыручка, КоличествоКупленныхТуров
3	ClientStat	Получить количество купленных туров для каждого клиента и общую потраченную им сумму	Request, RequestStatus, Client, Tour	ID_Клиента, ФИО_Клиента, КоличествоКупленныхТуров, ОбщаяПотраченнаяСумма

Скрипты запросов и представлений представлены в Приложении Б.

3.1.2 Разработка функций и хранимых процедур для реализации учетных пользовательских бизнес-процессов

В таблице 19 представлены спецификации разработанных функций и хранимых процедур.

Таблица 19 - Спецификации функций и хранимых процедур учетных бизнес-процессов

№ п/п	Имя функции (процедуры)	Назначение	Описание логики работы	Параметры и возвращаемые данные
1	CreateTour	Создание тура	Если введенной страны нет в таблице Country, то добавляем соответствующую запись. По строковым значениям получаем идентификаторы для отеля, типа номера и типа питания. Конечная дата получается прибавлением к начальной дате количество дней. Добавляется запись в таблицу Tour	Возвращаемого значения нет; Параметры: -Название тура; -Страна; -Название отеля; -Тип номера; -Тип питания; -Цена покупки -Цена продажи; -Дата начала -Количество дней
2	ToWaitRequest	Взять заявку в обработку	По идентификатору заявки, получаем идентификатор тура и статуса заявки. Если статус заявки «НЕОБРАБОТАНА», то изменяем ее на «ОЖИДАНИЕ ОТ ТУРОПЕРАТОРА» и помечаем, что тур больше не доступен для покупки	Возвращаемого значения нет; Параметры: -Идентификатор заявки

Полный тексты скриптов функций и хранимых процедур представлены в Приложении В.

3.1.3 Разработка триггеров для реализации учетных автоматических бизнес-процессов и обновления представлений

В данном разделе представлены спецификации триггера AFTER для реализации автоматически исполняемого учетного бизнес-процесса, а также триггера INSTEAD OF для модификации данных при выполнении операций

UPDATE для необновляемого представления. Спецификации триггеров приведены в таблице 20.

Таблица 20 - Спецификации триггеров

№ п/п	Назначение триггера	Вид триггера	Событие триггера	Таблица триггера	Описание работы триггера
1	Обновление видимости тура, при отмене тура	after	update	Request	Если статус заявки изменился на «ОТМЕНЕНА», то у соответствующего тура изменяется видимость для других клиентов
2	Удаление заявки при обращении к представлению AllRequestWithFullInformation	Instead of	delete	Request	Находит заявку по идентификатору и удаляет ее

Полный текст скриптов триггеров представляются в Приложении Г.

3.2 Разработка клиентской части приложения

3.2.1 Разработка пользовательского интерфейса клиентского приложения

Пользовательский интерфейс клиентского приложения должен обеспечивать привычное и удобное представление информации, простое и эффективное выполнение основных функций пользователей.

Интерфейс приложения состоит из нескольких страниц:

- страница, содержащая информацию о клиентах (рис. 22);
- страница, содержащая информацию о турах (рис. 23);
- страница для создания тура (рис. 24);
- страница, содержащая сводную информацию для каждой из стран, в которую есть тур (рис. 25);
- страница, содержащая информацию об отелях (рис. 26);

- страница, содержащая детальную информацию об отеле (рис. 27);
- страница, содержащая информацию о заявках (рис. 28);
- страницы, содержащие статистику (рис. 29-31).



Рисунок 22 – страница с клиентами

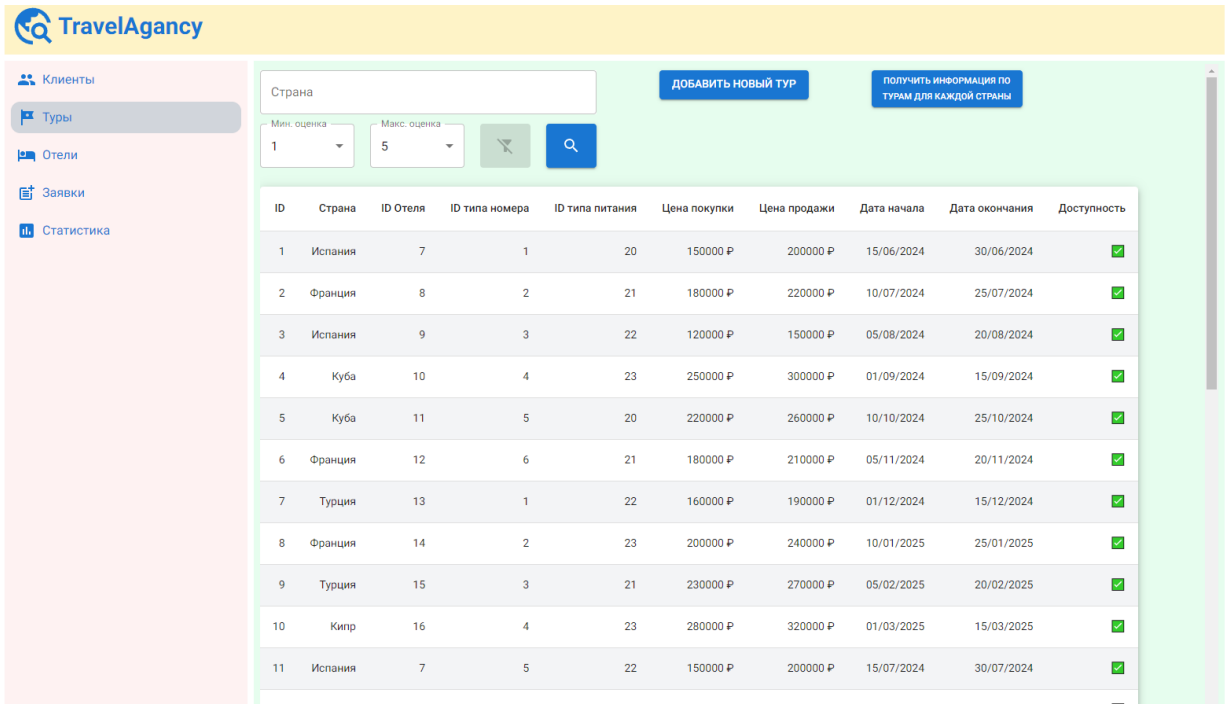


Рисунок 23 – страница с турами

TravelAgency

Клиенты

Туры

Отели

Заявки

Статистика

НАЗАД

Название тура

Страна

Отель

Тип комнаты

Тип питания

Цена покупки

0

Цена продажи

0

28 . 05 . 2024

Количество дней

1

СОЗДАТЬ

Рисунок 24 – страница для создания тура

TravelAgency

Клиенты

Туры

Отели

Заявки

Статистика

Информация по турам для каждой страны

СМОТРЕТЬ ВСЕ ТУРЫ

Страна	Количество путевок	Минимальная цена тура	Максимальная цена тура
Франция	10	175000 Р	240000 Р
Турция	6	190000 Р	270000 Р
Испания	6	150000 Р	200000 Р
Куба	4	260000 Р	300000 Р
Кипр	3	220000 Р	320000 Р
Россия	2	125000 Р	150000 Р

Рисунок 25 – страница со сводной информацией по странам

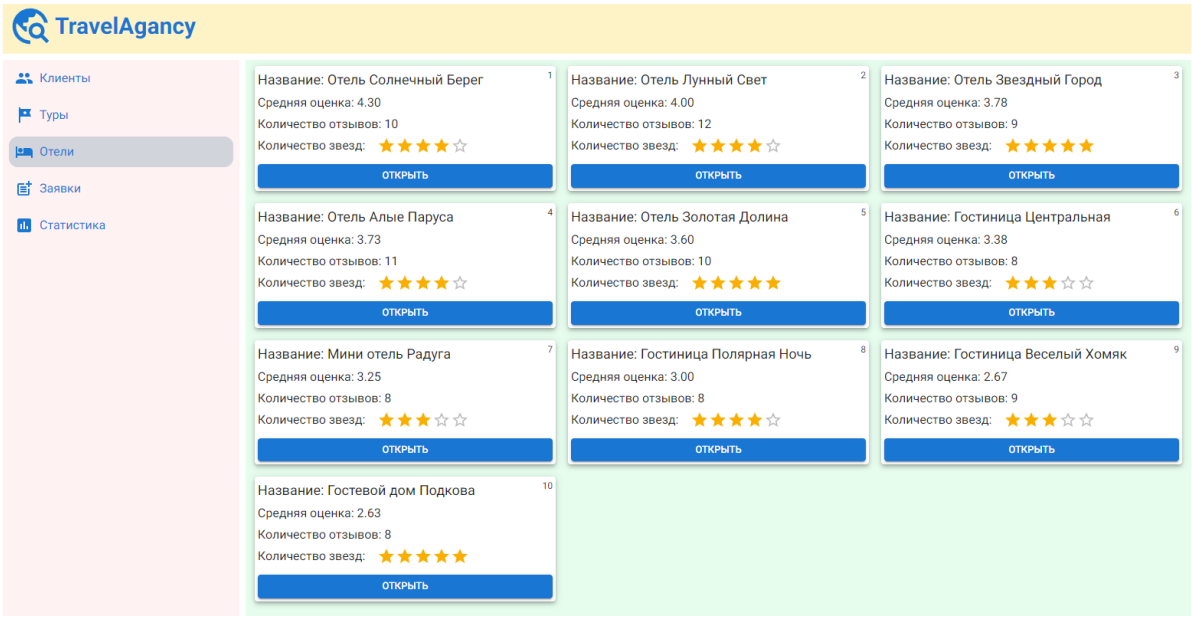


Рисунок 26 – страница со всеми отелями

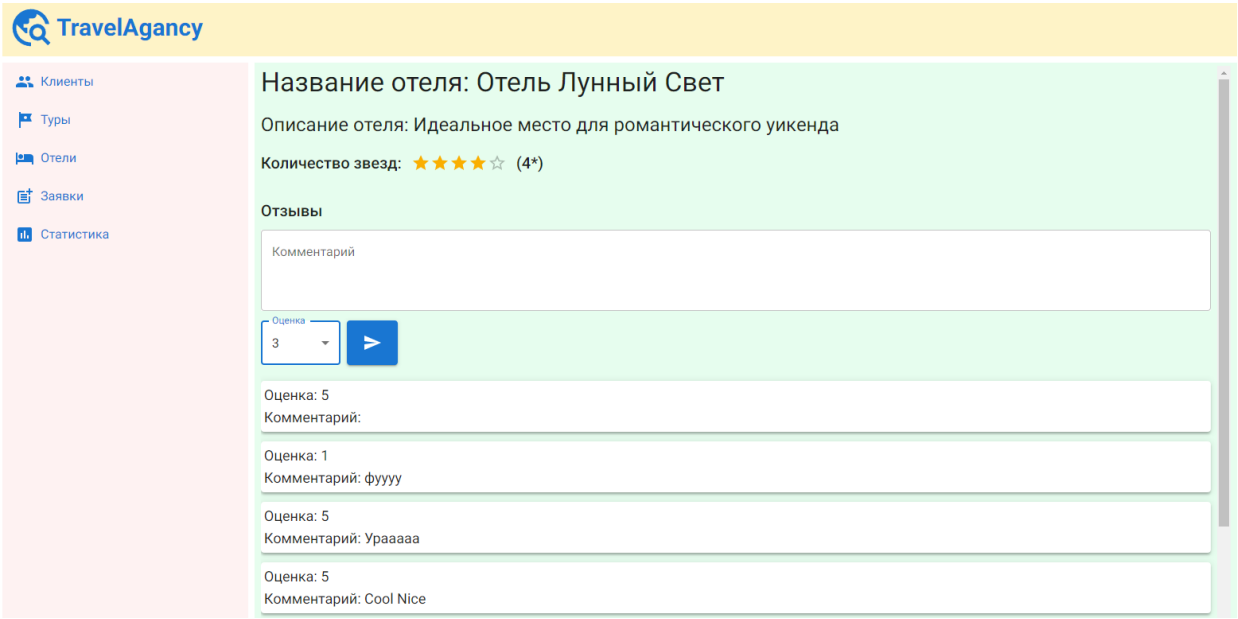


Рисунок 27 – страница с определенным отелем

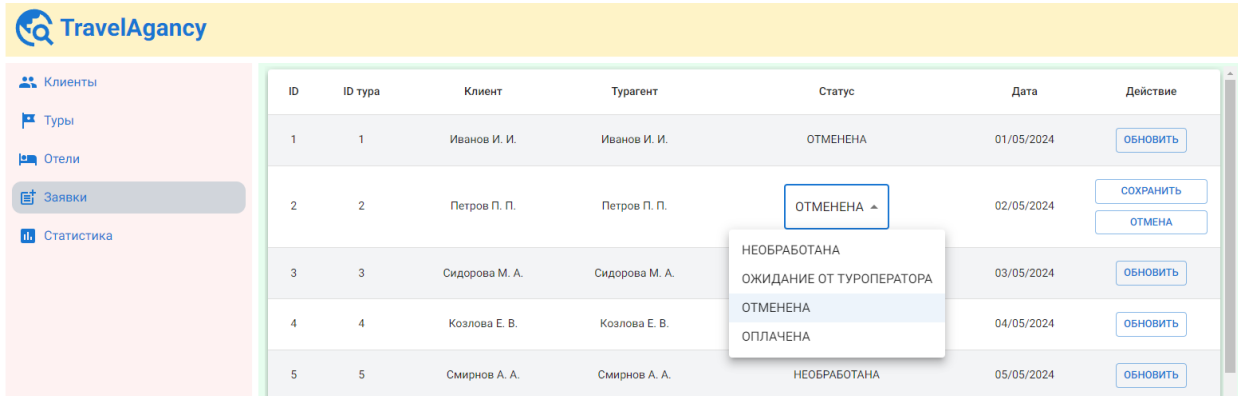


Рисунок 28 – страница с заявками

Статистика по всем странам [НАЗАД](#)

Страна	Количество купленных туров	Общая потраченная сумма
Куба	3	140000 Р
Франция	3	110000 Р
Испания	2	80000 Р
Турция	2	70000 Р
Кипр	1	40000 Р

Рисунок 29 – статистика по странам

Статистика по всем клиентам [НАЗАД](#)

ID клиента	ФИО клиента	Количество купленных туров	Общая потраченная сумма
4	Козлова Е. В.	2	600000 Р
9	Кузнецов Д. М.	1	270000 Р
5	Смирнов А. А.	1	260000 Р
8	Лебедева Н. П.	1	240000 Р
2	Петров П. П.	1	220000 Р
20	Калинина Л. А.	1	220000 Р
6	Игнатьева О. С.	1	210000 Р
1	Иванов И. И.	1	200000 Р
7	Васильев В. И.	1	190000 Р
3	Сидорова М. А.	1	150000 Р

Рисунок 30 – статистика по клиентам



Рисунок 31 – статистика по заявкам

3.2.2 Разработка программного обеспечения клиентского приложения

Было реализовано клиентское приложения. Исходный код программы на языке TypeScript представлен в Приложении Д. Код также доступен по ссылке <https://github.com/martin1917/BIHD>

Диаграмма классов приложения представлена на рисунке 32.

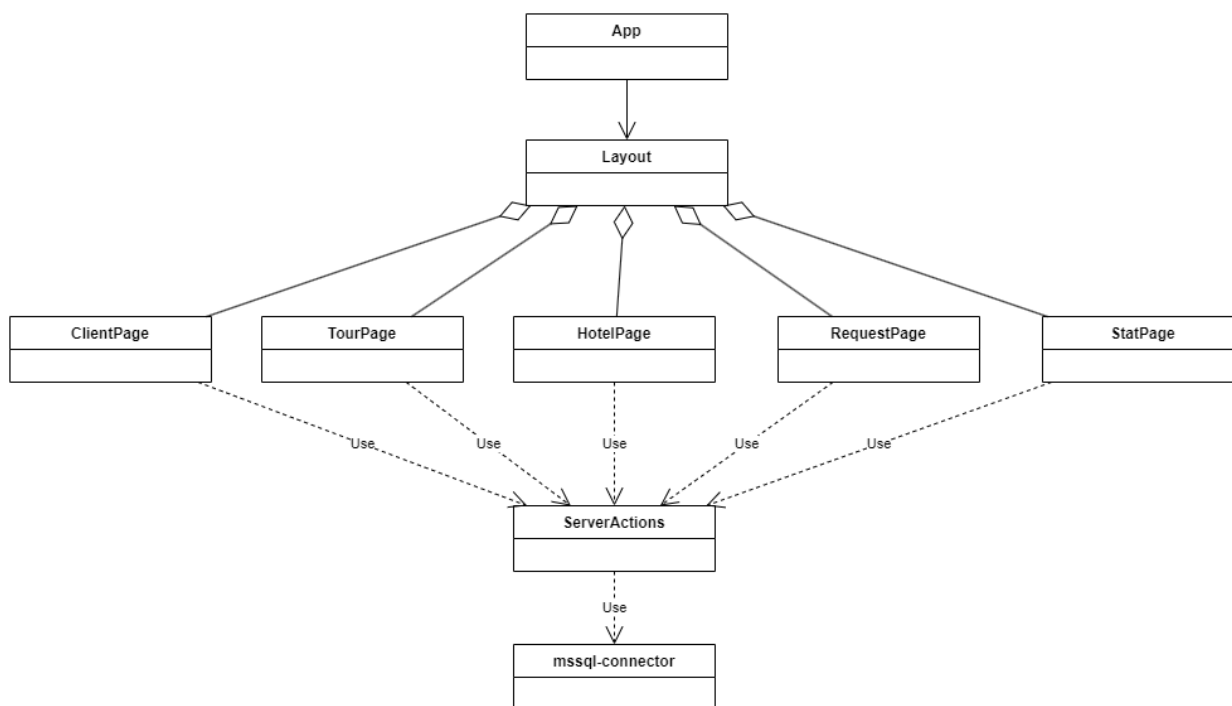


Рисунок 32 – диаграмма классов клиентского приложения

На данной диаграмме представлены следующие классы:

- mssql-connector – Класс для взаимодействия с базой данных;
- ServerActions – Класс для выполнения sql запросов на стороне сервера;
- ClientPage, TourPage, HotelPage, RequestPage, StatPage – Классы, представляющие собой страницы приложения;
- Layout – Класс, содержащий общие элементы для всех страниц;
- App – Класс, с помощью которого происходит запуск всего приложения.

3.3 Тестирование приложения

3.3.1 Выбор режима тестирования приложения

Функциональное тестирование - процесс обеспечения качества (QA) в рамках цикла разработки программного обеспечения, необходимый для проверки реализуемости функциональных требований, согласно спецификации тестируемого программного обеспечения. Оно проводится для оценки соответствия системы или компонента заданным функциональным

требованиям. Функциональное тестирование проводится по принципу черного ящика, в связи с чем функциональность ПО можно протестировать, не зная принципа его внутренней работы. Это снижает требования к тестировщикам в части знания языков программирования или конкретных аспектов реализации программного обеспечения [5].

При решении поставленной задачи будет проведено функциональное тестирование, так как оно наиболее наглядно отображает корректность выполнения функций приложения, соответствующих функциональным требованиям.

3.3.2 Тестирование серверной части приложения

В процессе разработки серверной части программы было проведено функциональное тестирование. Результаты тестирования представлены в таблице 21.

Таблица 21 – Результаты тестирования серверной части приложения

№ п/п	Назначение теста	Описание теста	Результаты теста
1	Проверка получения отфильтрованных туров	Получить список туров в «Испанию» в отель от 4 до 5 звезд	Тест пройден (рис. 33)
2	Проверка получения списка отелей	Получить список отелей, отсортированный по средней оценке в порядке убывания	Тест пройден (рис. 34)
3	Проверка получения сводной информации по странам	Получить количество туров в определенные страны, а также минимальную и максимальную цену на тур	Тест пройден (рис. 35)

Продолжение таблицы 21

4	Проверка получения статистики по доходам	Получить список оплаченных туров с 20/05/2024 по 25/05/2024	Тест пройден (рис. 36)
5	Проверка работы процедуры CreateTour	Добавление нового тура	Тест пройден (рис. 37)
6	Проверка работы триггера changeVisableTour	После отмены заявки, тур становится видимым	Тест пройден (рис. 38-39)

На представленных ниже скриншотах (рисунки 33-39) приведены результаты функционального тестирования серверной части программы.

Results		Messages									
	id	name	country	hotel	stars(*)	type_room	type_food	price_purchase	price_sale	date_end	date_end
1	1	Отдых на море	Испания	Отель Алые Паруса	4	одиночный	завтраки	150000.00	200000.00	2024-06-30 00:00:00.000	2024-06-30 00:00:00.000
2	3	Спортивный тур	Испания	Отель Золотая Долина	5	сдвоенный	полный пансион	120000.00	150000.00	2024-08-20 00:00:00.000	2024-08-20 00:00:00.000
3	11	Отдых на море	Испания	Отель Алые Паруса	4	четыrehместный	полный пансион	150000.00	200000.00	2024-07-30 00:00:00.000	2024-07-30 00:00:00.000
4	13	Спортивный тур	Испания	Отель Золотая Долина	5	одиночный	завтраки	120000.00	150000.00	2024-08-20 00:00:00.000	2024-08-20 00:00:00.000
5	21	Отдых на море	Испания	Отель Алые Паруса	4	сдвоенный	все включено	150000.00	200000.00	2024-07-30 00:00:00.000	2024-07-30 00:00:00.000
6	23	Спортивный тур	Испания	Отель Золотая Долина	5	четыrehместный	полупансион	120000.00	150000.00	2024-08-20 00:00:00.000	2024-08-20 00:00:00.000

Рисунок 33 – результат теста 1

Results		Messages		
	hotel	avg_mark	count	
1	Отель Солнечный Берег	4,25	8	
2	Отель Алые Паруса	4	8	
3	Отель Лунный Свет	4	8	
4	Отель Звездный Город	3,75	8	
5	Отель Золотая Долина	3,75	8	
6	Гостиница Центральная	3,375	8	
7	Мини отель Радуга	3,25	8	
8	Гостиница Полярная Ночь	3	8	
9	Гостевой дом Подкова	2,625	8	
10	Гостиница Веселый Хомяк	2,375	8	

Рисунок 34 – результат теста 2

Results Messages				
	country	count_tour	max_price	min_price
1	Франция	9	240000.00	210000.00
2	Испания	6	200000.00	150000.00
3	Куба	6	300000.00	260000.00
4	Турция	6	270000.00	190000.00
5	Кипр	3	320000.00	220000.00

Рисунок 35 – результат теста 3

Results Messages							
	client_id	tour_name	date_request	status	price_purchase	price_sale	profit
1	20	Круизы	2024-05-20 00:00:00.000	ОПЛАЧЕНА	180000.00	220000.00	40000.00
2	1	Отдых на море	2024-05-21 00:00:00.000	ОПЛАЧЕНА	150000.00	200000.00	50000.00
3	2	Экскурсионный тур	2024-05-22 00:00:00.000	ОПЛАЧЕНА	180000.00	220000.00	40000.00
4	3	Спортивный тур	2024-05-23 00:00:00.000	ОПЛАЧЕНА	120000.00	150000.00	30000.00
5	4	Гастрономический тур	2024-05-24 00:00:00.000	ОПЛАЧЕНА	250000.00	300000.00	50000.00
6	5	Приключенческий тур	2024-05-25 00:00:00.000	ОПЛАЧЕНА	220000.00	260000.00	40000.00

Рисунок 36 – результат теста 4

28	28	Экстрим туры	Франция	14	4	22	200000.00	240000.00	2025-01-10 00:00:00.000	2025-01-25 00:00:00.000	1
29	29	Лыжные туры	Турция	15	5	23	230000.00	270000.00	2025-02-05 00:00:00.000	2025-02-20 00:00:00.000	1
30	30	Круизы	Кипр	16	6	20	180000.00	220000.00	2025-03-01 00:00:00.000	2025-03-15 00:00:00.000	1
31	34	Крутой тур для всей семьи	Россия	14	5	22	100000.00	125000.00	2024-04-05 00:00:00.000	2024-04-19 00:00:00.000	1

Рисунок 37 – результат теста 5

Results Messages											
	id	name	country	hotel_id	type_room_id	type_food_id	price_purchase	price_sale	date_start	date_end	isActive
1	1	Отдых на море	Испания	7	1	20	150000.00	200000.00	2024-06-15 00:00:00.000	2024-06-30 00:00:00.000	0
2	2	Экскурсионный тур	Франция	8	2	21	180000.00	220000.00	2024-07-10 00:00:00.000	2024-07-25 00:00:00.000	1

Рисунок 38 – результат теста 6 (1 часть)

Results Messages											
	id	name	country	hotel_id	type_room_id	type_food_id	price_purchase	price_sale	date_start	date_end	isActive
1	1	Отдых на море	Испания	7	1	20	150000.00	200000.00	2024-06-15 00:00:00.000	2024-06-30 00:00:00.000	1

Рисунок 39 – результат теста 6 (2 часть)

3.3.3 Тестирование клиентской части приложения

В процессе разработки клиентской части программы было проведено функциональное тестирование. Результаты тестирования представлены в таблице 22.

Таблица 22 – Результаты тестирования клиентской части приложения

№ п/п	Назначение теста	Описание теста	Результаты теста
1	Проверка получения списка туров в определенную страну	Получить список туров во «Францию» в отель от 2 до 4 звезд	Тест пройден (рис. 40)
2	Проверка получения статистики	Получить статистику по доходу за период с 20/05/24 по 23/05/24	Тест пройден (рис. 41)
3	Проверка добавления отзыва	Оставить отзыв с оценкой 5. Средняя оценка и количество отзывов должно измениться	Тест пройден (рис. 42-44)

На представленных ниже скриншотах (рисунки 40-41) приведены результаты функционального тестирования клиентской части программы.

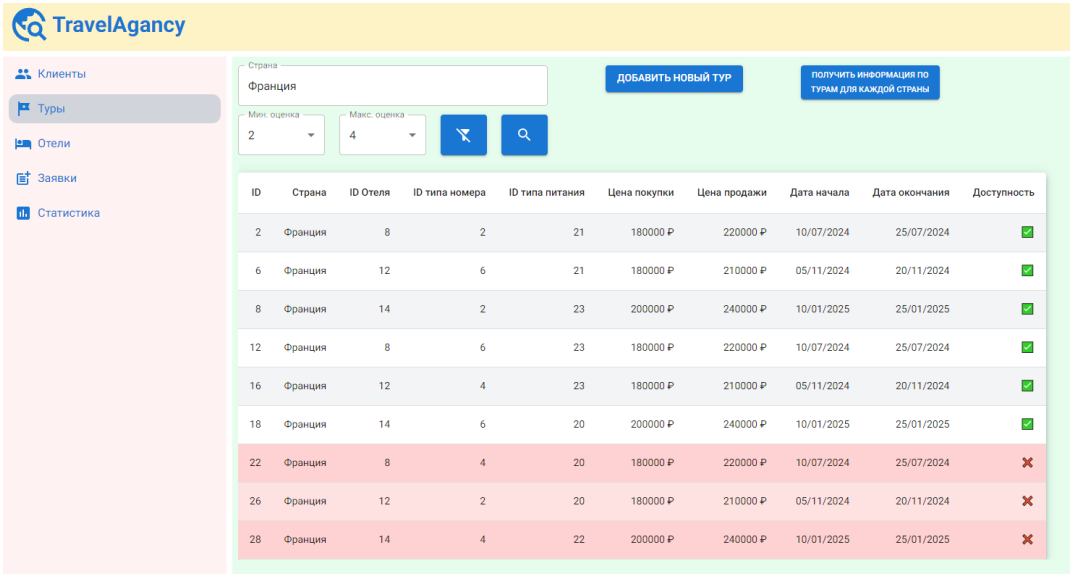


Рисунок 40 – результат теста 1

От

До

ПРИМЕНИТЬ

НАЗАД

20.05.2024

23.05.2024

ОЧИСТИТЬ

Общая выручка: **160000 Р**

ID клиента	ID тура	Дата	Статус	Выручка
20	20	20/05/2024	ОПЛАЧЕНА	40000 Р
1	21	21/05/2024	ОПЛАЧЕНА	50000 Р
2	22	22/05/2024	ОПЛАЧЕНА	40000 Р
3	23	23/05/2024	ОПЛАЧЕНА	30000 Р

Рисунок 41 – результат теста 2

Название: Гостиница Полярная Ночь

Средняя оценка: 3.00

Количество отзывов: 8

Количество звезд: ★★★★★

ОТКРЫТЬ

Рисунок 42 – результат теста 3 (часть 1)

Оценка: 5

Комментарий: Хороший отель. Все понравилось

Рисунок 43 – результат теста 3 (часть 2)

Название: Гостиница Полярная Ночь

Средняя оценка: 3.22

Количество отзывов: 9

Количество звезд: ★★★★★

ОТКРЫТЬ

Рисунок 44 – результат теста 3 (часть 3)

По результатам выполнения функционального тестирования ошибок в исполнении приложения не обнаружено, следовательно, программа работает корректно.

ЗАКЛЮЧЕНИЕ

В рамках данной курсовой работы было разработано приложение учета и анализа заявок клиентов для сервиса турагентства, которое представляет собой важный инструмент для эффективного управления клиентскими запросами и обеспечения высокого уровня обслуживания. Помимо серверной части приложения, также разработано клиентское приложение, работающее на персональном компьютере на операционной системе Windows 7 и выше.

При выполнении данной курсовой работы достигнуты следующие цели:

- сформулированы и проанализированы требования;
- спроектирована концептуальная модель базы данных;
- построена логическая и физическая модели базы данных;
- создана база данных информационной системы и заполнена тестовыми данными;
- созданы аналитические запросы и представления, созданы хранимые процедуры и функции, созданы триггеры;
- разработано клиентское приложение;
- протестировано и отлажено приложение.

Разработанное приложение предназначено для турагентов. Оно позволяет учитывать и обрабатывать заявки от клиентов, а также формировать аналитику по заявкам.

В дальнейшем систему можно улучшить путем интеграции с онлайн платформами для бронирования (например, Booking.com, Airbnb), это позволит автоматизировать процесс бронирования и улучшить доступность услуг для клиентов. Также возможно расширение аналитических возможностей, что позволит выявлять тенденции и предлагать персонализированные путевки на основе данных клиентов.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Информационные системы [Электронный ресурс]. URL: <https://otus.ru/journal/informacionnye-sistemy-opredelenie-i-metodologii-sozdaniya/> (дата обращения: 31.01.2024)
2. Концептуальная модель. [Электронный ресурс]. URL: <https://monographies.ru/ru/book/section?id=16242> (дата обращения: 31.01.2024)
3. Логическая модель. [Электронный ресурс]. URL: <https://erfa.ru/logicheskie-modeli-dannyh-fizicheskaya-i-logicheskaya-modeli-bd.html> (дата обращения: 31.01.2024)
4. Физическая модель. [Электронный ресурс]. URL: <https://powerbi.microsoft.com/ru-ru/what-is-data-modeling/#:~:text=Физическая%20модель%20данных%20—%20это,—%20фактическая%20реализация%20массива%20данных> (дата обращения: 31.01.2024)
5. Лайза Криспин, Джанет Грегори. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams. — М.: «Вильямс», 2010. — 464 с. — (Addison-Wesley Signature Series).

ПРИЛОЖЕНИЕ А

(Исходные скрипты создания базы данных)


```
CREATE TABLE Client (id int IDENTITY NOT NULL CHECK(id < 99999),
name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я]',
LEN(name))), second_name varchar(64) NOT NULL CHECK(second_name like
replicate( '[А-Яа-я]', LEN(second_name))), patronymic varchar(64) NULL
CHECK(patronymic like replicate( '[А-Яа-я]', LEN(patronymic))),
birthday datetime NOT NULL CHECK(birthday > '19000101'), phone
varchar(12) NOT NULL CHECK(phone like '+[1-9][0-9][0-9][0-9][0-9][0-
9][0-9][0-9][0-9][0-9][0-9]'), email varchar(64) NOT NULL CHECK(email
like '%@%'), PRIMARY KEY (id));
```

```
CREATE TABLE Country (name varchar(64) NOT NULL CHECK(name like
replicate( '[А-Яа-я]', LEN(name))), description varchar(255) NULL,
PRIMARY KEY (name));
```

```
CREATE TABLE Feedback (id int IDENTITY NOT NULL CHECK(id <
99999), mark int NOT NULL CHECK(mark >= 1 AND mark <= 5), comment
varchar(255) NULL, hotel_id int NOT NULL, PRIMARY KEY (id));
```

```
CREATE TABLE Hotel (id int IDENTITY NOT NULL CHECK(id < 99999),
name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я ]',
LEN(name))), description varchar(255) NULL, stars int NOT NULL
CHECK(stars >= 1 AND stars <= 5), PRIMARY KEY (id));
```

```
CREATE TABLE Request (id int IDENTITY NOT NULL CHECK(id < 99999),
tour_id int NOT NULL, client_id int NOT NULL, touragent_id int NOT
NULL, request_status_id int NOT NULL, [date] datetime NOT NULL
CHECK([date] > '19000101'), PRIMARY KEY (id));
```

```
CREATE TABLE RequestStatus (id int IDENTITY NOT NULL CHECK(id <
10), name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я ]',
LEN(name))), description varchar(255) NULL, PRIMARY KEY (id));
```

```
CREATE TABLE Tour (id int IDENTITY NOT NULL CHECK(id < 99999),
name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я ]',
LEN(name))), country varchar(64) NOT NULL, hotel id int NOT NULL,
type_room_id int NOT NULL, type_food_id int NOT NULL, price_purchase
decimal(8, 2) NOT NULL CHECK(price_purchase > 0.00), price_sale
decimal(8, 2) NOT NULL CHECK(price_sale > 0.00), date_start datetime
NOT NULL CHECK(date_start > '19000101'), date_end datetime NOT NULL
CHECK(date_end > '19000101'), PRIMARY KEY (id));
```

```
CREATE TABLE TourAgent (id int IDENTITY NOT NULL CHECK(id <
99999), name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-
я]', LEN(name))), second name varchar(64) NOT NULL CHECK(second name
like replicate( '[А-Яа-я]', LEN(second_name))), patronymic varchar(64)
NULL CHECK(patronymic like replicate( '[А-Яа-я]', LEN(patronymic))),
PRIMARY KEY (id));
```

```
CREATE TABLE Tourist (id int IDENTITY NOT NULL CHECK(id < 99999),
name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я]',
LEN(name))), second name varchar(64) NOT NULL CHECK(second name like
replicate( '[А-Яа-я]', LEN(second_name))), patronymic varchar(64) NULL
CHECK(patronymic like replicate( '[А-Яа-я]', LEN(patronymic))),
birthday datetime NOT NULL CHECK(birthday > '19000101'), passport
varchar(10) NOT NULL CHECK(passport like '[0-9][0-9][0-9][0-9] [0-
9][0-9][0-9][0-9]'), request_id int NOT NULL, PRIMARY KEY (id));
```

```
CREATE TABLE TypeFood (id int IDENTITY NOT NULL CHECK(id < 10),
name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я ]',
LEN(name))), description varchar(255) NULL, PRIMARY KEY (id));
```

```

CREATE TABLE TypeRoom (id int IDENTITY NOT NULL CHECK(id < 10),
name varchar(64) NOT NULL CHECK(name like replicate( '[А-Яа-я ]',
LEN(name))), description varchar(255) NULL, PRIMARY KEY (id));

ALTER TABLE Tourist ADD CONSTRAINT [В заявке указываются туристы]
FOREIGN KEY (request_id) REFERENCES Request (id);

ALTER TABLE Tour ADD CONSTRAINT [В стране проводятся туры]
FOREIGN KEY (country) REFERENCES Country (name);

ALTER TABLE Tour ADD CONSTRAINT [Для тура определен отель]
FOREIGN KEY (hotel_id) REFERENCES Hotel (id);

ALTER TABLE Tour ADD CONSTRAINT [Для тура определен тип номера]
FOREIGN KEY (type_room_id) REFERENCES TypeRoom (id);

ALTER TABLE Tour ADD CONSTRAINT [Для тура определен тип питания]
FOREIGN KEY (type_food_id) REFERENCES TypeFood (id);

ALTER TABLE Request ADD CONSTRAINT [заявке принадлежит статус]
FOREIGN KEY (request_status_id) REFERENCES RequestStatus (id);

ALTER TABLE Request ADD CONSTRAINT [Клиент оформляет заявки]
FOREIGN KEY (client_id) REFERENCES Client (id);

ALTER TABLE Feedback ADD CONSTRAINT [На отель оставляют обратную
связь] FOREIGN KEY (hotel_id) REFERENCES Hotel (id);

ALTER TABLE Request ADD CONSTRAINT [На тур оформляются заявки]
FOREIGN KEY (tour_id) REFERENCES Tour (id);

ALTER TABLE Request ADD CONSTRAINT [турагент обрабатывает заявки]
FOREIGN KEY (touragent_id) REFERENCES TourAgent (id);

```

ПРИЛОЖЕНИЕ Б

(Исходные коды запросов и представлений)

Листинг Б.1 – SQL запрос на получение списка туров в заданную страну в отель, у которого количество звезд находится в заданном интервале.

```
select
    t.id,
    t.name,
    t.country,
    h.name as 'hotel',
    h.stars as 'stars(*)',
    tr.name as 'type_room',
    tf.name as 'type_food',
    t.price_purchase,
    t.price_sale,
    t.date_end,
    t.date_end
from Tour t
    join Hotel h on t.hotel_id = h.id
    join TypeFood tf on t.type_food_id = tf.id
    join TypeRoom tr on t.type_room_id = tr.id
where
    t.country = 'Испания' and
    h.stars between 4 and 5;
```

Листинг Б.2 – SQL запрос на получение списка отелей, отсортированных по средней оценке в порядке убывания.

```
select
    h.name as 'hotel',
    avg(cast(fb.mark as float)) as 'avg_mark',
    count(1) as 'count'
from Feedback fb join Hotel h on fb.hotel_id = h.id
group by h.name order by avg_mark desc
```

Листинг Б.3 – SQL запрос на получение количества туров в определенные страны, а также минимальную и максимальную цену на тур в этой стране.

```
select
    country,
    count(1) as 'count_tour',
    max(price_sale) as 'max_price',
    min(price_sale) as 'min_price'
from Tour
group by country
order by count_tour desc
```

Листинг Б.4 – SQL запрос на получение списка оплаченных туров в определенный интервал времени.

```
select
    client_id,
    t.name as 'tour_name',
    r.date as 'date_request',
    rs.name as 'status',
    t.price_purchase,
```

```

        t.price_sale,
        (t.price_sale - t.price_purchase) as 'profit'
from Request r
join Tour t on r.tour_id = t.id
join RequestStatus rs on r.request_status_id = rs.id
where
    rs.name = 'ОПЛАЧЕНА'
    and r.date between '20240520' and '20240525'

```

Листинг Б.5 – SQL запрос на создание представления, которое получает все заявки с полной и удобно читаемой информацией.

```

create view AllRequestWithFullInformation as
select
    r.id as 'ИдентификаторЗаявки',
    t.name as 'НазваниеТура',
    c.second_name + ' ' + substring(c.name, 1, 1) + '.' +
substring(c.patronymic, 1, 1) + '.' as 'ФИО_Клиента',
    ta.second_name + ' ' + substring(ta.name, 1, 1) + '.' +
substring(ta.patronymic, 1, 1) + '.' as 'ФИО_Турагента',
    rs.name as 'СтатусЗаявки',
    r.date as 'Дата'
from Request r
join RequestStatus rs on r.request_status_id = rs.id
join Tour t on r.tour_id = t.id
join Client c on r.client_id = c.id
join TourAgent ta on r.touragent_id = ta.id

```

Листинг Б.6 – SQL запрос на создание представления, которое получает количество проданных туров в определенные страны и их общую выручку в этой стране

```

create view SaledToursByCountry as
select
    t.country as 'НазваниеСтраны',
    sum(t.price_sale - t.price_purchase) as 'ОбщаяВыручка',
    count(1) as 'КоличествоКупленныхТуров'
from Request r
join RequestStatus rs on r.request_status_id = rs.id
join Tour t on r.tour_id = t.id
where rs.name = 'ОПЛАЧЕНА'
group by t.country

```

Листинг Б.7 – SQL запрос на создание представления, которое получает количество купленных туров для каждого клиента и общую потраченную им сумму

```

create view ClientStat as
select
    cs.client_id as 'ID_Клиента',
    c.second_name + ' ' + substring(c.name, 1, 1) + '.' +
substring(c.patronymic, 1, 1) + '.' as 'ФИО_Клиента',
    cs.count_sales as 'КоличествоКупленныхТуров',

```

```

        cs.total_sum as 'ОбщаяПотраченнаяСумма'
from Client c
join
(
    select
        r.client_id,
        count(1) as 'count_sales',
        sum(t.price_sale) as 'total_sum'
    from Request r
    join RequestStatus rs on r.request_status_id= rs.id
    join Client c on r.client_id = c.id
    join Tour t on r.tour_id = t.id
    where rs.name = 'ОПЛАЧЕНА'
    group by r.client_id
) cs on c.id = cs.client_id

```

ПРИЛОЖЕНИЕ В

(Исходные коды скриптов хранимых процедур)

Листинг В.1 – процедура создание тура «CreateTour»

```

CREATE PROCEDURE CreateTour
@NameTour NVARCHAR(64), @Country NVARCHAR(64),
@HotelName NVARCHAR(64), @TypeRoom NVARCHAR(64),
@TypeFood NVARCHAR(64), @PricePurchase DECIMAL(8,2),
@PriceSale DECIMAL(8,2), @DateStart DATETIME,
@CountDays INT
AS
BEGIN
    IF NOT EXISTS(SELECT 1 FROM Country WHERE name = @Country)
        INSERT INTO Country (name, description) VALUES (@Country, '');

    DECLARE @hotel_id int = (SELECT id FROM Hotel WHERE name =
@HotelName);
    IF @hotel_id IS NULL
        THROW 50002, 'Hotel with name not exist', 1;

    DECLARE @type_room_id int = (SELECT id FROM TypeRoom WHERE name =
@TypeRoom);
    IF @type_room_id IS NULL
        THROW 50003, 'TypeRoom with name not exist', 1;

    DECLARE @type_food_id int = (SELECT id FROM TypeFood WHERE name =
@TypeFood);
    IF @type_food_id IS NULL
        THROW 50004, 'TypeFood with name not exist', 1;

    DECLARE @date_end datetime = @DateStart + @CountDays;

    INSERT INTO Tour (name, country, hotel_id, type_room_id,
type_food_id, price_purchase, price_sale, date_start, date_end) VALUES
    (
        @NameTour,
        @Country,
        @hotel_id,
        @type_room_id,
        @type_food_id,
        @PricePurchase,
        @PriceSale,
        @DateStart,
        @date_end
    );
END

```

Листинг В.2 – процедура взятия заявки в обработку «ToWaitRequest»

```

CREATE PROCEDURE ToWaitRequest
@requestId INT
AS
BEGIN
    DECLARE @tour_id INT, @request_status_id INT;
    SELECT
        @tour_id=tour_id,
        @request_status_id=request_status_id
    FROM Request WHERE id = @requestId;

```



```

    DECLARE @status NVARCHAR(64);
    SELECT @status=name FROM RequestStatus WHERE id =
@request_status_id
    if @status != 'НЕОБРАБОТАНА'
        throw 50005, 'Заявка уже обработана', 1;

    DECLARE @res_status_id INT;
    SELECT @res_status_id=id FROM RequestStatus WHERE name = 'ОЖИДАНИЕ
ОТ ТЫПОНЕРАТОРА'

    UPDATE Tour SET isActive = 0 WHERE id = @tour_id;
    UPDATE Request SET request_status_id = @res_status_id WHERE id =
@requestId;
END

```

ПРИЛОЖЕНИЕ Г

(Исходные коды скриптов триггеров)

Листинг Г.1 – триггер на обновление видимости тура, при его отмене

```
CREATE TRIGGER changeVisableTour
ON Request AFTER UPDATE
AS
DECLARE @tour_id INT, @request_id INT, @request_status_id
INT, @cancel_status_id INT
BEGIN
    SELECT @cancel_status_id=id FROM RequestStatus WHERE name =
'ОТМЕНЕНА';
    SELECT @request_id=id, @request_status_id=request_status_id FROM
inserted
    IF @request_status_id = @cancel_status_id
    BEGIN
        SELECT @tour_id=tour_id FROM Request WHERE id = @request_id;
        UPDATE Tour SET isActive = 1 WHERE id = @tour_id;
    END;
END
```

Листинг Г.2 – триггер на удаление заявки при обращении к представлению

AllRequestWithFullInformation

```
CREATE TRIGGER deleteRequest
ON AllRequestWithFullInformation instead OF DELETE
AS
BEGIN
    DELETE FROM Request WHERE id IN
    (
        SELECT ИдентификаторЗаявки FROM deleted
    )
END
```

ПРИЛОЖЕНИЕ Д

(Исходные коды программного обеспечения клиентского приложения)

Файл ./app/layout.tsx

```
import "../globals.css";
import type { Metadata } from "next";
import { AppRouterCacheProvider } from "@mui/material-nextjs/v14-appRouter";
import { ThemeProvider } from "@mui/material/styles";
import CssBaseline from "@mui/material/CssBaseline";
import theme from "../theme";
import { Box, Container, Stack, Link, Typography } from "@mui/material";
import Header from "@components/header";
import Sidebar from "@components/sidebar";

export const metadata: Metadata = {
  title: "Create Next App",
  description: "Generated by create next app",
};

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  return (
    <html lang='en'>
      <body>
        <AppRouterCacheProvider options={{ enableCssLayer: true }}>
          <ThemeProvider theme={theme}>
            <CssBaseline />
            <Container className='h-screen max-w-[1600px]'>
              <Header />
              <Box className='flex gap-2 h-[calc(100vh-5rem)]'>
                <Box className='w-1/5 bg-red-50 py-1 px-2'>
                  <Sidebar />
                </Box>
                <Box className='w-4/5 bg-green-100/70 py-1 px-2'>{children}</Box>
              </Box>
            </Container>
          </ThemeProvider>
        </AppRouterCacheProvider>
      </body>
    </html>
  );
}
```

Файл ./app/clients/page.tsx

```
import { getAllClients } from "@actions/db-clients";
import * as utils from "@utils";
import {
  Box,
  Paper,
  Table,
  TableBody,
```

```

    TableCell,
    TableContainer,
    TableHead,
    TableRow,
    Typography,
  } from "@mui/material";

export default async function ClientsPage() {
  const clients = await getAllClients();

  return (
    <Box className='p-1 h-full w-full overflow-auto'>
      <Box>
        <Typography variant='h6' className='mb-3'>
          Клиенты (общее количество: <b>{clients.length}</b> )
        </Typography>
        <TableContainer component={Paper} elevation={5} className='w-
max'>
          <Table>
            <TableHead>
              <TableRow>
                <TableCell align='right'>ID</TableCell>
                <TableCell align='right'>Фамилия</TableCell>
                <TableCell align='right'>Имя</TableCell>
                <TableCell align='right'>Отчество</TableCell>
                <TableCell align='right'>Дата рождения</TableCell>
                <TableCell align='right'>Почта</TableCell>
                <TableCell align='right'>Телефон</TableCell>
              </TableRow>
            </TableHead>
            <TableBody>
              {clients.map((client) => (
                <TableRow key={client.id} className='odd:bg-gray-100'>
                  <TableCell align='right'>{client.id}</TableCell>
                  <TableCell
align='right'>{client.second_name}</TableCell>
                  <TableCell align='right'>{client.name}</TableCell>
                  <TableCell
align='right'>{client.patronymic}</TableCell>
                  <TableCell
align='right'>{utils.formatDate(client.birthday)}</TableCell>
                  <TableCell align='right'>{client.email}</TableCell>
                  <TableCell align='right'>{client.phone}</TableCell>
                </TableRow>
              ))}
            </TableBody>
          </Table>
        </TableContainer>
      </Box>
    </Box>
  );
}

```

Файл ./app/tours/page.tsx

```
import { Box } from "@mui/material";
```

```
import ToursTable from "../ToursTable";
import { getAllTours } from "@actions/db-tours";

export default async function ToursPage() {
  const tours = await getAllTours();

  return (
    <Box className='h-full w-full overflow-auto'>
      <ToursTable tours={tours} />
    </Box>
  );
}
```

Файл ./app/tours/pageTable.tsx

```
"use client";

import * as utils from "@/utils";
import { clsx } from "clsx";
import {
  Box,
  Button,
  Link,
  Paper,
  Table,
  TableBody,
  TableCell,
  TableContainer,
  TableHead,
  TableRow,
} from "@mui/material";
import { TourEntity } from "@types/tour";
import { useState } from "react";
import TourInfoModal from "../TourInfoModal";
import TourFilter from "../TourFilter";
import NextLink from "next/link";

interface ToursTableProps {
  tours: TourEntity[];
}

export default function ToursTable({ tours }: ToursTableProps) {
  const [toursState, setTours] = useState(tours);
  const [modalIsOpen, setModalIsOpen] = useState<boolean>(false);
  const [selectedTourId, setSelectedTourId] = useState<number>();

  const handleClose = () => setModalIsOpen(false);
  const handleOpen = () => setModalIsOpen(true);

  const handleDoubleClickOnRow = (tourId: number) => {
    setSelectedTourId(tourId);
    handleOpen();
  };

  return (
    <Box className='overflow-auto mr-1'>
```

```

    <TourInfoModal tourId={selectedTourId} open={modalIsOpen}
    handleClose={handleClose} />
    <Box className='mb-4 py-2 flex gap-20'>
      <TourFilter setTours={setTours} />
      <Link component={NextLink} href={`\tours/create`}>
        <Button variant='contained'>Добавить новый тур</Button>
      </Link>
      <Link component={NextLink} href={`\tourStats`}>
        <Button variant='contained' size='small' className='text-
[0.7rem] w-[12rem]'>
          Получить информация по турам для каждой страны
        </Button>
      </Link>
    </Box>
    <TableContainer component={Paper} elevation={5} className='w-
max'>
      <Table>
        <TableHead>
          <TableRow>
            <TableCell align='right'>ID</TableCell>
            <TableCell align='right'>Страна</TableCell>
            <TableCell align='right'>ID Отеля</TableCell>
            <TableCell align='right'>ID типа номера</TableCell>
            <TableCell align='right'>ID типа питания</TableCell>
            <TableCell align='right'>Цена покупки</TableCell>
            <TableCell align='right'>Цена продажи</TableCell>
            <TableCell align='right'>Дата начала</TableCell>
            <TableCell align='right'>Дата окончания</TableCell>
            <TableCell align='right'>Доступность</TableCell>
          </TableRow>
        </TableHead>
        <TableBody>
          {toursState.map((tour) => (
            <TableRow
              key={tour.id}
              onClick={() => handleDoubleClickOnRow(tour.id)}
              className={clsx(
                "select-none odd:bg-gray-100 hover:cursor-pointer
                hover:bg-blue-200 active:bg-blue-300",
                {
                  "bg-red-100 odd:bg-red-200/85": !tour.isActive,
                }
              )}
            >
              <TableCell align='right'>{tour.id}</TableCell>
              <TableCell align='right'>{tour.country}</TableCell>
              <TableCell align='right'>{tour.hotel_id}</TableCell>
              <TableCell
                align='right'>{tour.type_room_id}</TableCell>
              <TableCell
                align='right'>{tour.type_food_id}</TableCell>
              <TableCell align='right'>{`$${tour.price_purchase}
Р`}</TableCell>
              <TableCell align='right'>{`$${tour.price_sale}
Р`}</TableCell>
              <TableCell
                align='right'>{utils.formatDate(tour.date_start)}</TableCell>

```



```

        <TableCell
align='right'>{utils.formatDate(tour.date_end)}</TableCell>
        <TableCell align='right'>{tour.isActive ? "✔" :
"✗"}</TableCell>
    </TableRow>
    )}
</TableBody>
</Table>
</TableContainer>
</Box>
);
}

```