

## Техническое задание

### Создание базы данных, и необходимой логики

**Проект: Представительство туристической фирмы в зарубежной стране**

#### Оглавление

1. Общие сведения.....	1
1.1.Используемые термины и сокращения.....	1
1.2.Цели изменений (доработок) системы.....	2
1.3.Область применения.....	3
1.4.Описание бизнес-процессов.....	4
1.5.Описание изменений бизнес-процессов.....	4
2. Требования к изменяемой подсистеме.....	14
2.1.Требование к функционалу.....	14
2.2.Требования к ролям.....	16
2.3.Требования к интерфейсам.....	19
2.4.Требования к миграции и изменению потоков данных.....	19
2.5.Нефункциональные требования.....	21
2.6.Справочная информация по объектам (СИ).....	22
2.7.Порядок и виды тестирования.....	35
3. План внедрения.....	43
ПРИЛОЖЕНИЕ .....	46

## 1 Общие сведения

### 1.1 Используемые термины и сокращения

- БД – база данных;
- Турист-отдыхающий – в рамках данной информационной системы, турист, который не использует склад, но записывается на экскурсии.
- Турист-грузоперевозчик – турист шоп-тура, которому необходимы услуги склада, но которому не интересны экскурсии.

## 1.2 Цели изменений (доработок) системы

Цель	Критерий достижения
Управление информацией о гостиницах и расселении туристов	Возможность хранения и обновления данных о туристах, включая персональную информацию, паспортные данные, контактные данные и историю поездок.
Управление информацией о гостиницах и расселении туристов	Возможность хранения данных о гостиницах, включая название, адрес, количество номеров и доступность. Возможность регистрации и учета расселения туристов по гостиницам.
Управление информацией о заказанных экскурсиях	Возможность хранения данных о доступных экскурсиях, включая название, описание, стоимость и продолжительность. Возможность регистрации и учета заказанных экскурсий для каждого туриста.
Управление информацией о грузе	Возможность хранения данных о грузе, включая описание, вес, объем и статус доставки. Возможность отслеживания груза и связи с туристами, отправляющими груз.
Управление финансовой информацией	Возможность учета доходов и расходов представительства, включая данные о оплате туристических услуг, комиссионных, затратах на грузоперевозки, гостиницы и экскурсии. Возможность генерации финансовых отчетов.
Управление статистической информацией	Возможность получения статистических данных о количестве туристов, посещающих страну, популярности экскурсий, загрузке грузовых рейсов и других показателях, необходимых для анализа и принятия управленческих решений.
Управление информацией о визах и документах	Возможность хранения информации о визах, документах и сроках их действия. Возможность генерации документов для получения виз и других необходимых документов.

### 1.3 Область применения

База данных, разработанная на основе представленной предметной области представительства туристической фирмы, может использоваться для создания отчетов и аналитики, связанных с деятельностью представительства. Ниже приведены некоторые примеры областей применения базы данных для формирования отчетов:

1. Финансовый отчет: База данных может использоваться для сбора данных о доходах и расходах представительства. С помощью базы данных можно составить полный финансовый отчет, включающий статьи расходов и доходов, такие как гостиница, перевозки, экскурсии, непредвиденные расходы и расчеты в аэропорту.

2. Отчет о туристической активности: База данных может предоставить информацию о количестве туристов, побывавших в стране за определенный период, включая разделение на различные категории туристов (отдыхающие, грузовые, дети). Это позволит оценить популярность различных типов туристической активности и провести анализ спроса.

3. Отчет о расселении туристов: База данных может использоваться для формирования списков расселения туристов по различным гостиницам. Отчет может содержать информацию о количестве занимаемых номеров и проживавших в них людей за определенный период.

4. Отчет о загрузке самолетов: База данных может предоставить данные о загрузке указанного рейса самолета на определенную дату, включая количество мест, вес.

5. Отчёт о грузообороте склада: База данных может предоставить данные о грузообороте склада за указанный период, о типах самолётов, вывозящих грузы, о типах хранимых грузов, о затратах и прибыли со склада.

## 1.4 Описание бизнес-процессов

Отсутствуют в связи с отсутствием информационной системы.

## 1.5 Описание изменений бизнес-процессов

Разработанная БД автоматизирует основные бизнес-процессы представительства туристической фирмы в зарубежной стране, обеспечивая:

### 1) Управление данными туристов

- Регистрация и хранение персональных данных туристов, включая ФИО, паспортные данные, пол, возраст, детей и предпочтения по размещению.

- Формирование списков туристов по категориям (отдыхающие, shor-туристы, дети) для таможи и расселения в гостиницах.

### 2) Получение виз

- Заполнение и отправка пакетов документов для получения виз в отдел эмиграции.

- Получение и распределение виз по туристам.

### 3) Размещение в гостиницах

- Бронирование номеров в гостиницах на основе предпочтений туристов.

- Составление списков расселения по гостиницам.

- Управление занятостью номеров и проживанием туристов.

### 4) Организация экскурсий

- Предоставление расписания экскурсий и запись туристов на выбранные экскурсии.

- Составление списков туристов, участвующих в экскурсиях, и передача их в агентства организации экскурсий.

### 5) Хранение и отправка груза

- Ведение весовых ведомостей на каждого туриста.
- Маркировка, взвешивание и упаковка груза.
- Составление ведомостей на отправку груза, включая количество мест, вес, стоимость упаковки и страхования.

#### 6) Финансовый учет

- Регистрация всех статей расхода и дохода, включая гостиницу, перевозки, экскурсии, непредвиденные расходы и расчеты в аэропорту.
- Составление подробного финансового отчета для головной фирмы.

#### 7) Анализ и отчетность

- Формирование отчетов о количестве туристов, побывавших в стране за определенный период.
- Получение сведений о конкретных туристах, включая историю посещений, размещение в гостиницах, экскурсии и груз.
- Определение самых популярных экскурсий и экскурсионных агентств.
- Статистика по грузообороту склада, включая количество мест, вес груза и загрузку самолетов.
- Вычисление рентабельности представительства.

Реализованные бизнес-процессы обеспечивают эффективное управление операциями представительства, оптимизируют взаимодействие с туристами, гостиницами и экскурсионными агентствами, а также предоставляют подробную финансовую и аналитическую информацию для принятия обоснованных решений.

Ниже в UML-диаграммах представлены все бизнесы процессы, которые были реализованы при разработке данной информационной системы для представительства туристической фирмы:

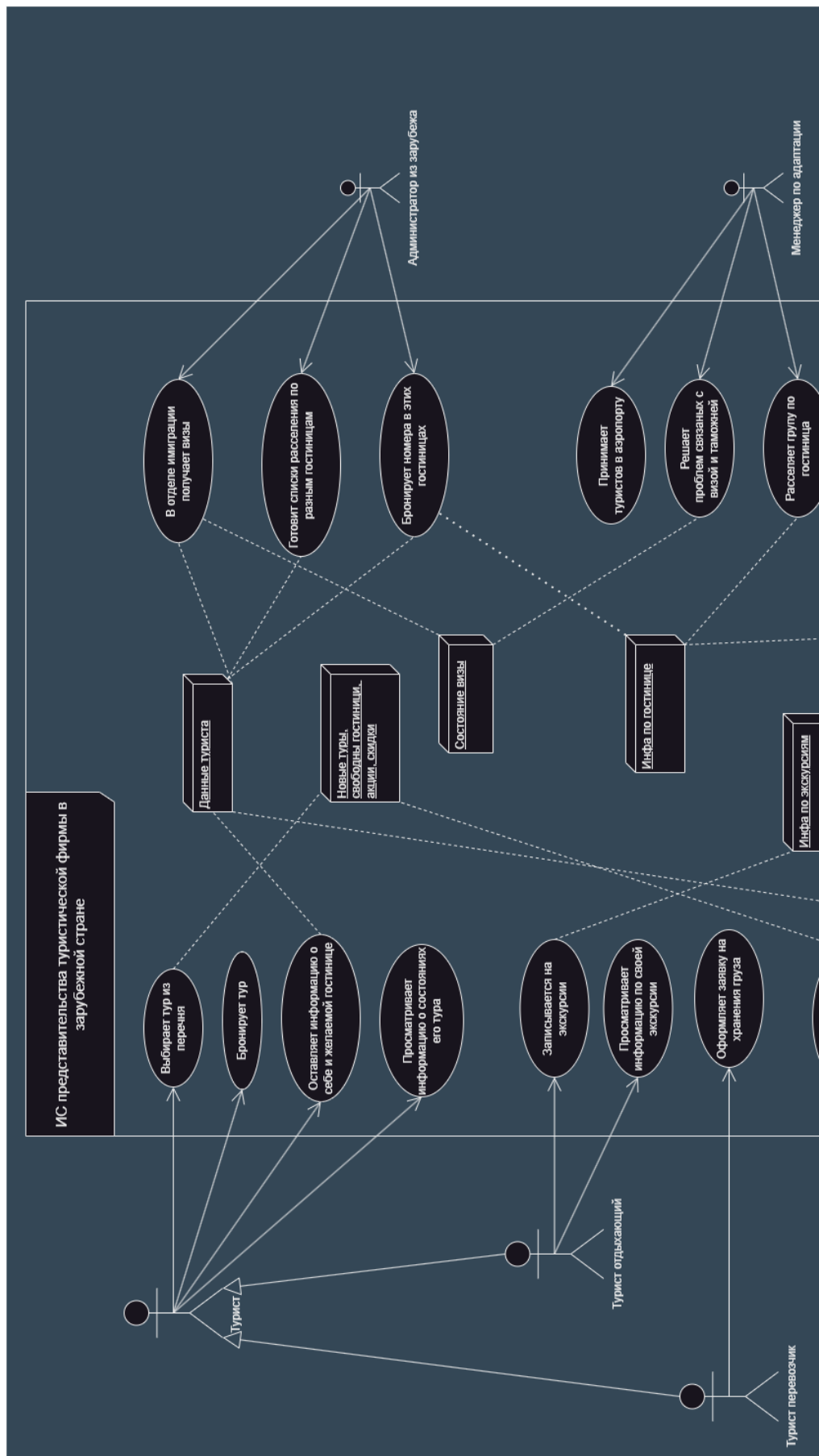


Диаграмма вариантов использования (UseCase) Часть 1

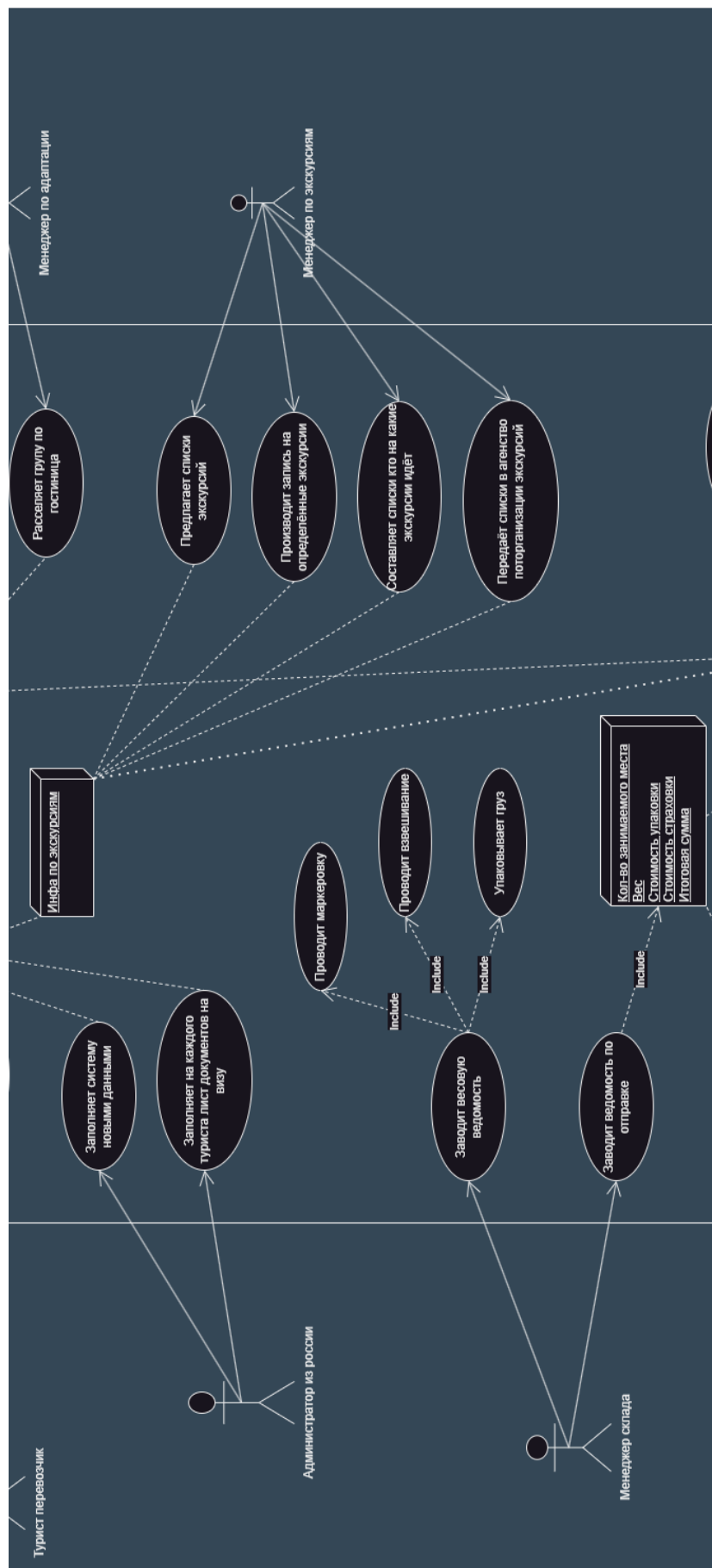


Диаграмма вариантов использования (UseCase) Часть 2

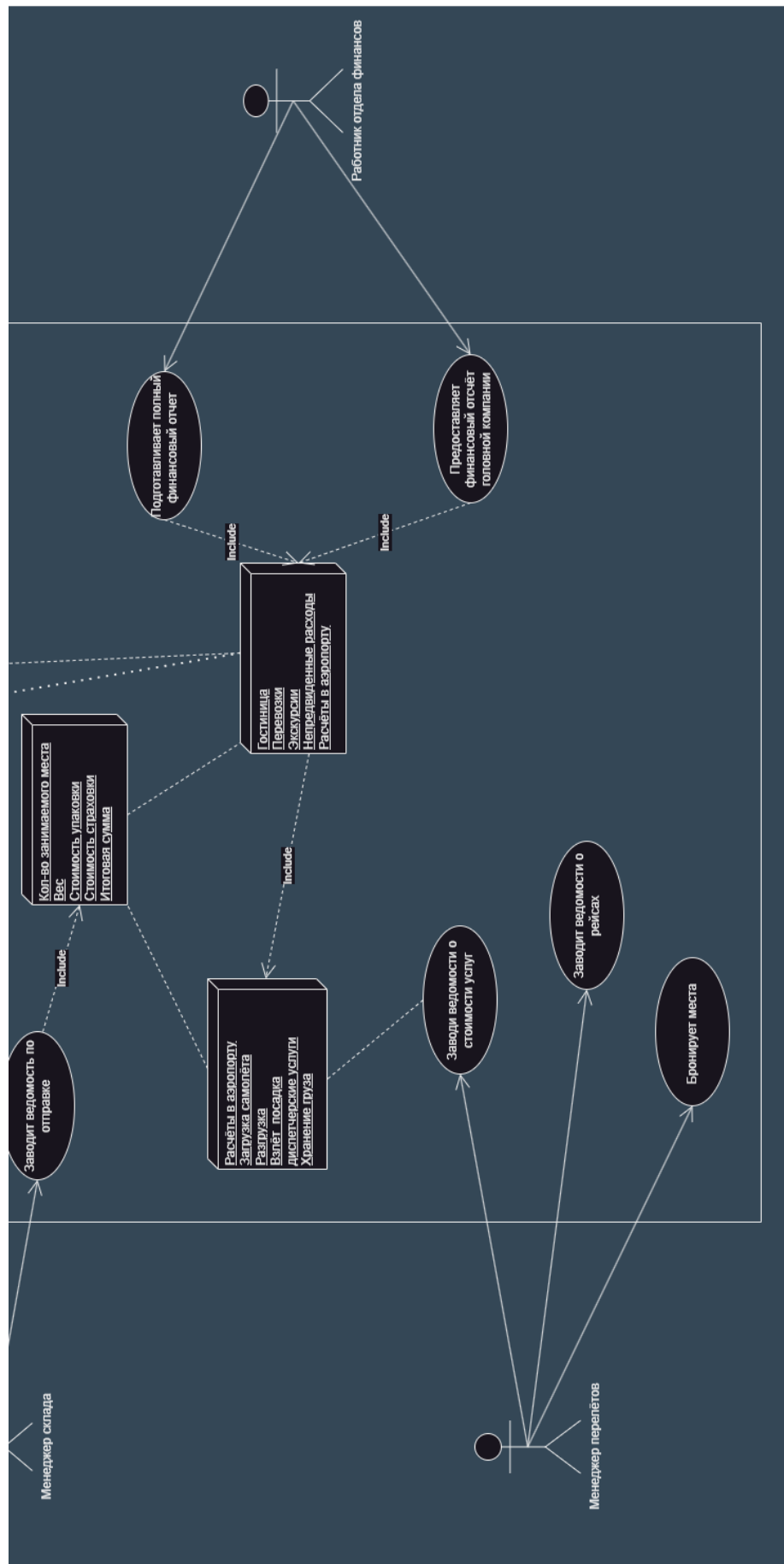


Диаграмма вариантов использования (UseCase) Часть 3



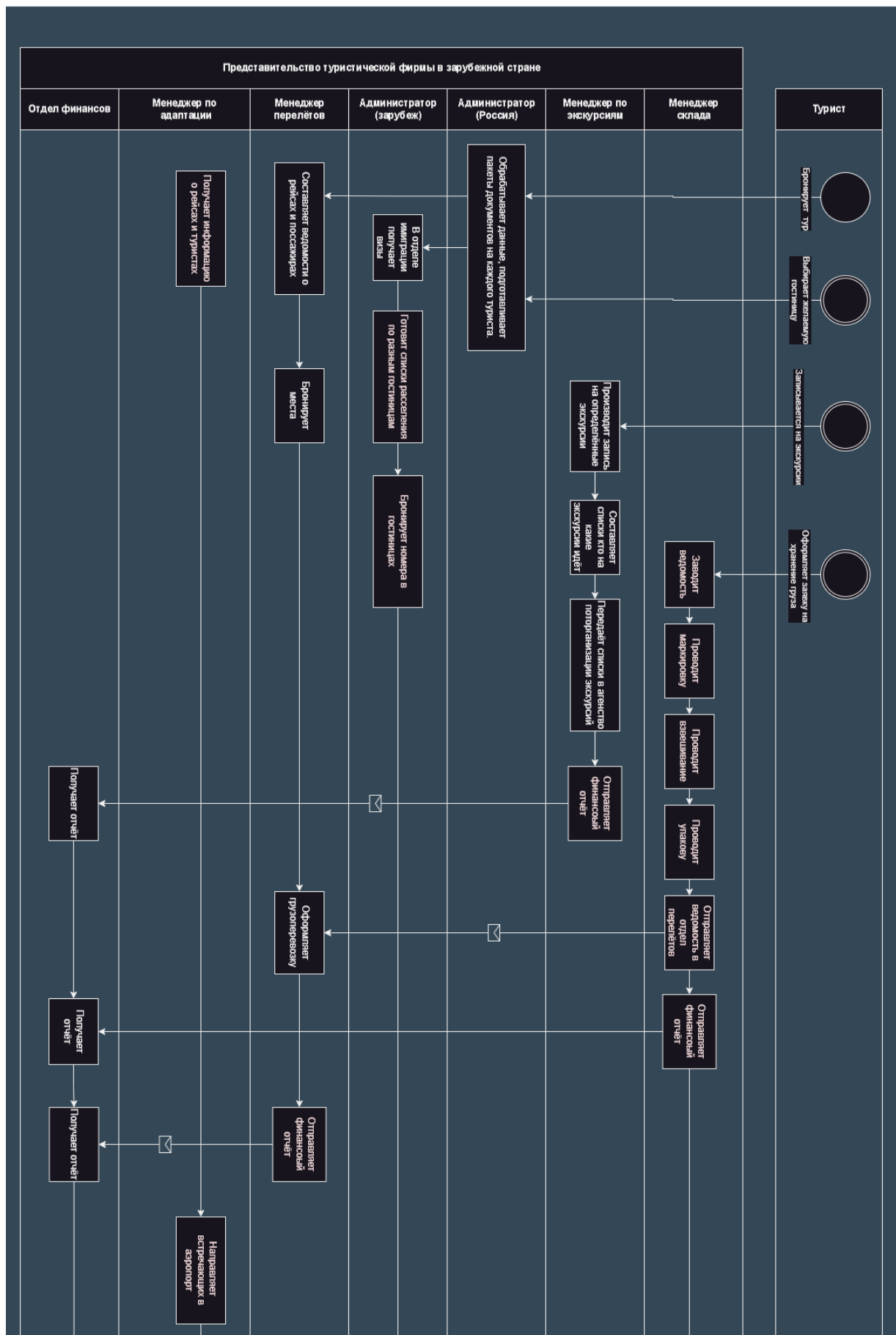


Диаграмма бизнес-процессов (BPMN) Часть 1

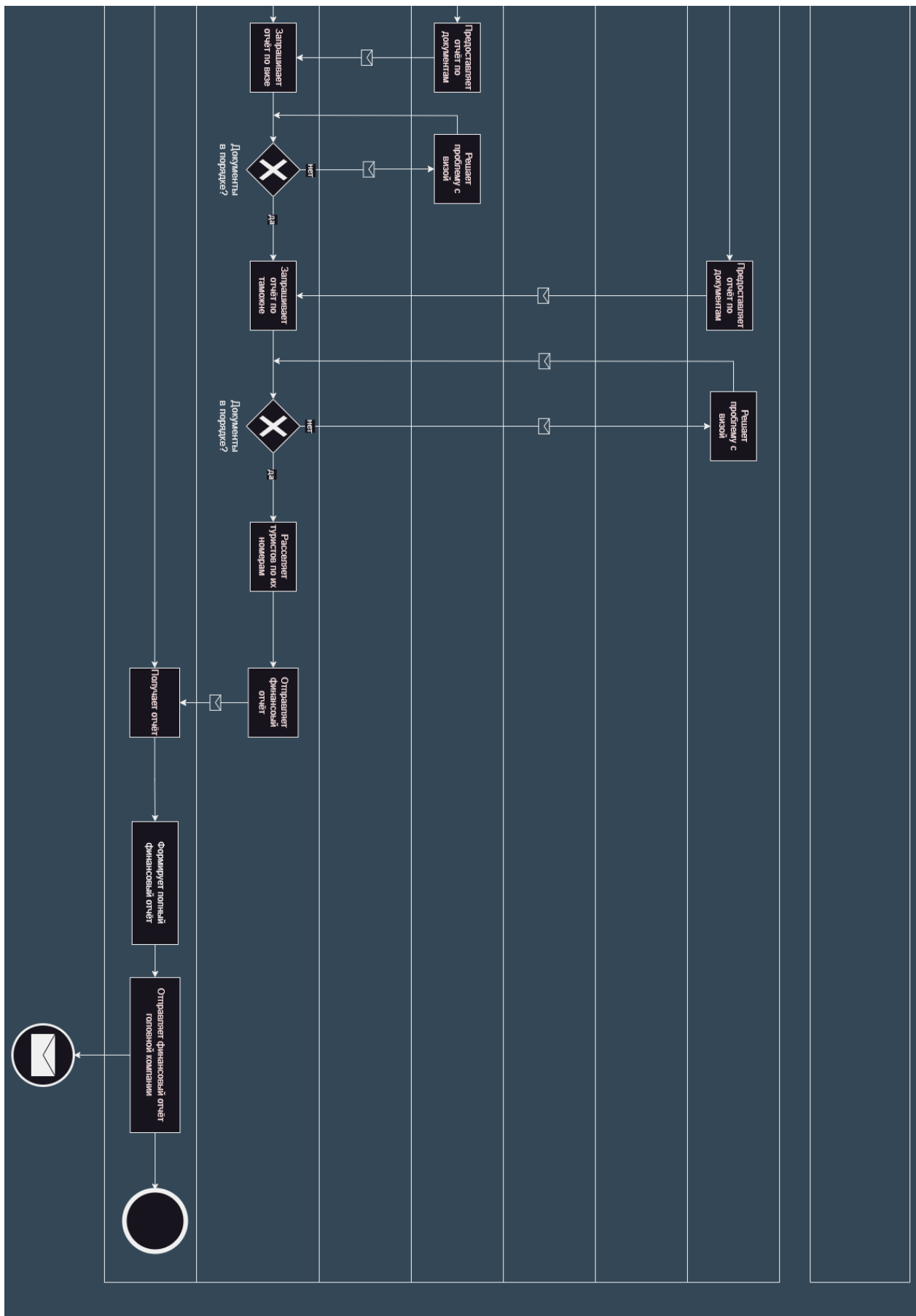


Диаграмма бизнес-процессов (BPMN) Часть 2

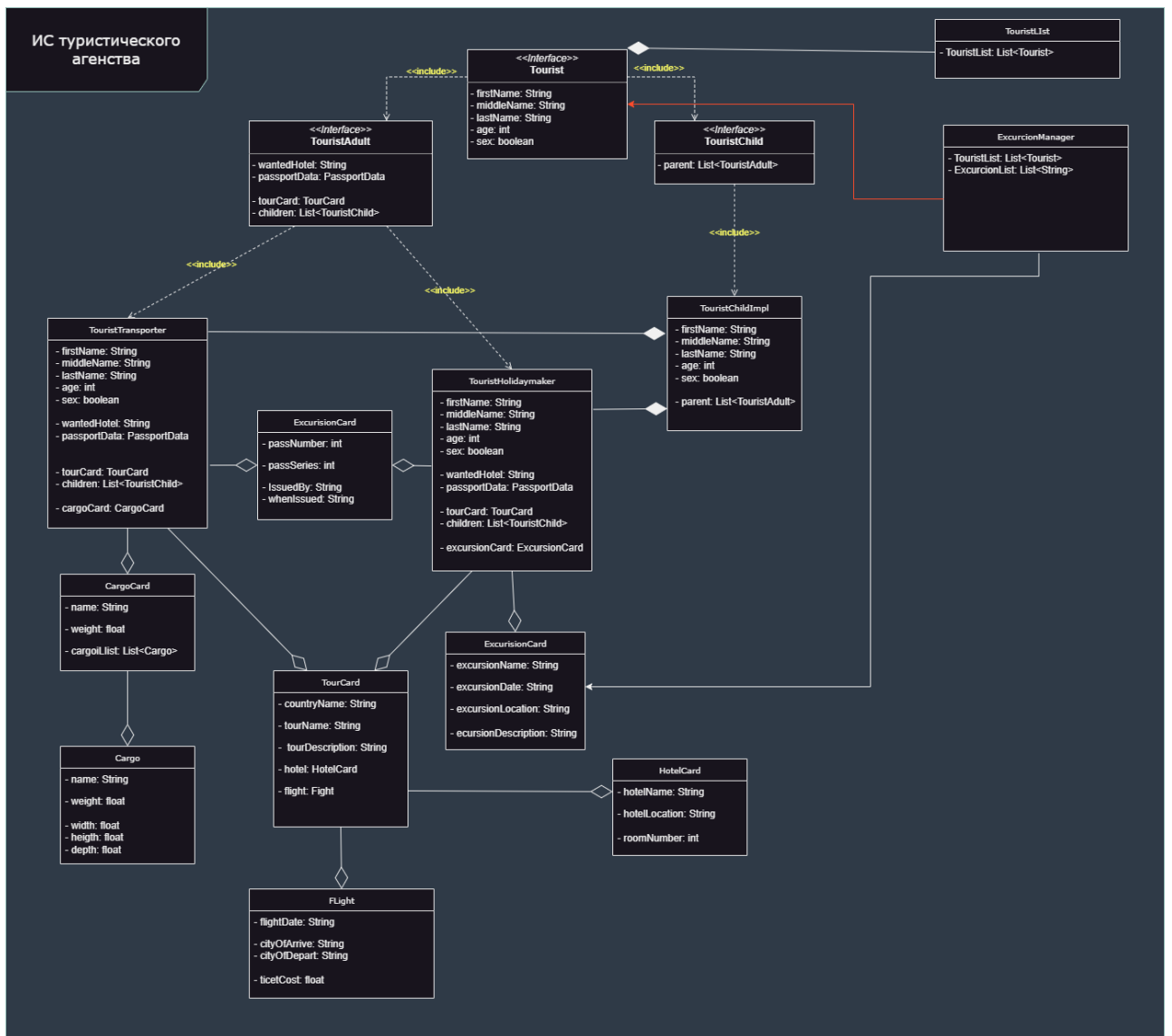


Диаграмма классов (ClassDiagram)

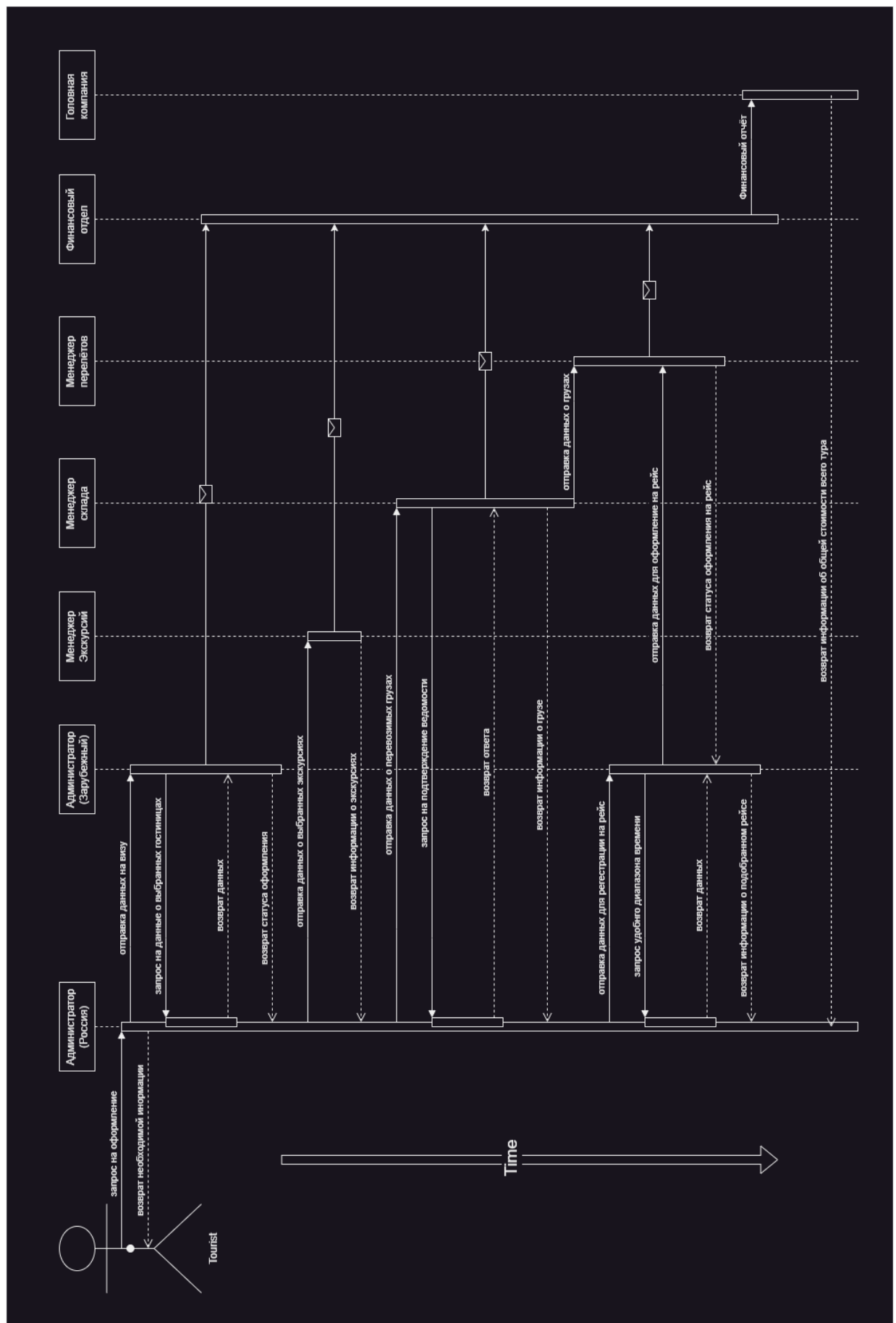


Диаграмма последовательностей (SequenceDiagram)

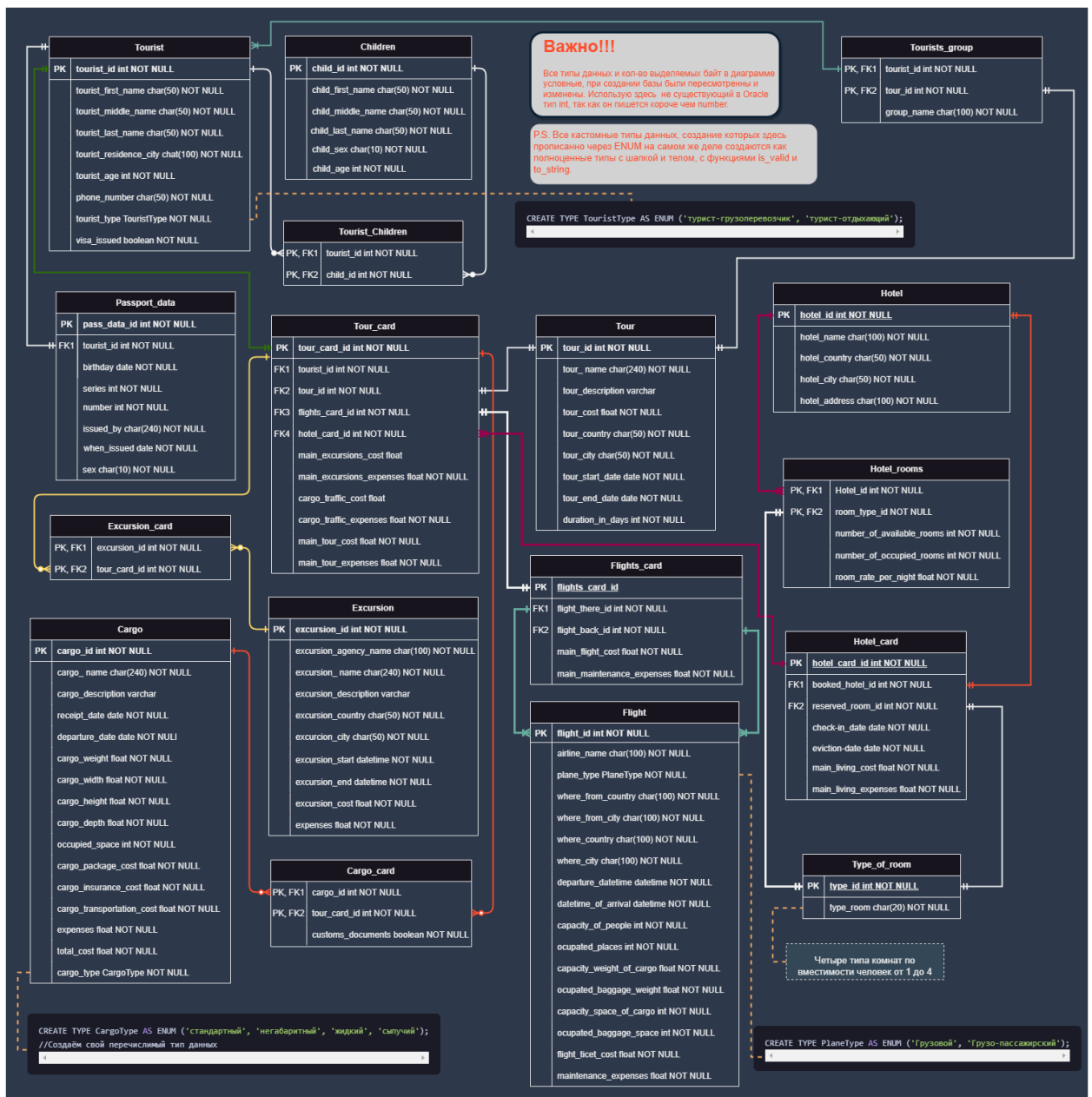


Диаграмма сущностей и связей (ER-диаграмма)

## 2 Требования к изменяемой подсистеме

### 2.1 Требование к функционалу

Общие требования:

а) Модуль управления туристами:

- Хранение личных данных туристов (ФИО, паспортные данные, пол, возраст, дети)

- Управление предпочтениями туристов по размещению в гостиницах

б) Модуль обработки виз:

- Генерация пакетов документов для получения виз на основе данных туристов

- Отслеживание статуса визовых заявлений

- Получение виз в отделе эмиграции

в) Модуль бронирования гостиниц:

- Создание списков расселения туристов по гостиницам

- Бронирование номеров в гостиницах

г) Модуль обслуживания туристов:

- Прием туристов в аэропорту

- Решение проблем, связанных с визами и таможней

- Расселение туристов по гостиницам

- Предложение расписания экскурсий и запись на экскурсии

- Управление списками участников экскурсий

д) Модуль управления категориями туристов:

- Классификация туристов на отдыхающих, туристов за грузом и детей

- Учет специфических характеристик каждой категории туристов

е) Модуль управления складом:

- Ведение весовых ведомостей на каждого туриста
- Маркировка, взвешивание и упаковка груза
- Составление ведомостей на отправку груза с указанием количества мест, веса, стоимости упаковки, страховки и итоговой суммы

ж) Модуль финансовой отчетности:

- Сбор и хранение данных о расходах и доходах (гостиница, перевозки, экскурсии, непредвиденные расходы)
- Формирование финансовых отчетов для головной фирмы

Требования к запросам:

- 1) Возможность сформировать список туристов для таможни в целом и по указанной категории.
- 2) Возможность сформировать списки на расселение по указанным гостиницам в целом и указанной категории.
- 3) Возможность получить количество туристов, побывавших в стране за определенный период в целом и по определенной категории.
- 4) Возможность получить сведения о конкретном туристе: сколько раз был в стране, даты прилета/отлета, в каких гостиницах останавливался, какие экскурсии и в каких агентствах заказывал, какой груз сдавал.
- 5) Возможность получить список гостиниц, в которых производится расселение туристов, с указанием количества занимаемых номеров и проживавших в них человек за определенный период.
- 6) Возможность получить общее количество туристов, заказавших экскурсии за определенный период.
- 7) Возможность выбрать самые популярные экскурсии и самые качественные экскурсионные агентства.
- 8) Возможность получить данные о загрузке указанного рейса самолета на определенную дату: количество мест, вес груза, объемный вес.

9) Возможность получить статистику о грузообороте склада: количество мест и вес груза, сданного за определенный период, количество самолетов, вывозивших этот груз, сколько из них грузовых, а сколько грузопассажирских.

10) Возможность получить полный финансовый отчет по указанной группе в целом и для определенной категории туристов.

11) Возможность получить данные о расходах и доходах за определенный период: обслуживание самолета, гостиница, экскурсии, визы, расходы представительства и т.п.

12) Возможность получить статистику по видам отправляемого груза и удельную долю каждого вида в общем грузопотоке.

13) Возможность вычислить рентабельность представительства (соотношение доходов и расходов).

14) Возможность определить процентное отношение отдыхающих туристов к туристам *shop*-туров в целом и за указанный период (например, в зависимости от времени года).

15) Возможность получить сведения о туристах указанного рейса: список группы, гостиницы, груз, бирки, маркировка.

## **2.2 Требования к ролям**

Роли и привилегии пользователей в базе данных туристического агентства:

Для эффективного управления базой данных туристического агентства необходимо создать несколько ролей с различными привилегиями. Ниже приведен список ролей и привилегий, которые могут потребоваться:

Роль: Администратор базы данных (DBA)

Привилегии:

1) Полный доступ ко всем объектам базы данных



2) Возможность создавать, изменять и удалять пользователей и роли

3) Возможность предоставлять и отзывать привилегии

4) Возможность создавать и управлять резервными копиями базы данных

Роль: Менеджер по работе с туристами

Привилегии:

1) Возможность создавать, изменять и удалять записи о туристах

2) Возможность просматривать и редактировать информацию о визах, гостиницах и экскурсиях

3) Возможность формировать списки туристов для таможи и расселения

4) Возможность получать сведения о конкретных туристах

Роль: Бухгалтер

Привилегии:

1) Возможность просматривать и редактировать финансовые данные

2) Возможность формировать финансовые отчеты

3) Возможность получать статистику о расходах и доходах

Роль: Менеджер по работе со складом

Привилегии:

1) Возможность создавать, изменять и удалять записи о грузе

2) Возможность просматривать и редактировать информацию о маркировке, взвешивании и упаковке груза

3) Возможность формировать ведомости на отправку груза

4) Возможность получать статистику о грузообороте склада

Роль: Аналитик

Привилегии:

1) Возможность просматривать и анализировать данные о туристах, гостиницах, экскурсиях и грузе

2) Возможность получать статистические отчеты

3) Возможность вычислять рентабельность представительства

Роль: Пользователь отчетов

Привилегии:

1) Возможность просматривать и печатать отчеты

2) Возможность экспортировать отчеты в различные форматы

Роль: Разработчик (Пользователь MATTAKVHSI)

Привилегии:

- **CREATE USER:** Создание новых пользователей
- **ALTER USER:** Изменение существующих пользователей
- **DROP USER:** Удаление пользователей
- **GRANT ROLE:** Предоставление ролей другим пользователям
- **REVOKE ROLE:** Отзыв ролей у других пользователей
- **GRANT ANY PRIVILEGE:** Предоставление любых привилегий другим пользователям
- **REVOKE ANY PRIVILEGE:** Отзыв любых привилегий у других пользователей
- **CREATE PROFILE:** Создание профилей для ограничения ресурсов, доступных пользователям
- **ALTER PROFILE:** Изменение существующих профилей
- **DROP PROFILE:** Удаление профилей
- **CREATE JOB:** Создание заданий для автоматизации задач
- **ALTER JOB:** Изменение существующих заданий
- **DROP JOB:** Удаление заданий

Привилегии для управления пакетом DBMS.SCHEDULER

- **EXECUTE DBMS\_SCHEDULER.CREATE\_JOB:** Создание заданий в пакете DBMS.SCHEDULER
- **EXECUTE DBMS\_SCHEDULER.ALTER\_JOB:** Изменение существующих заданий в пакете DBMS.SCHEDULER
- **EXECUTE DBMS\_SCHEDULER.DROP\_JOB:** Удаление заданий в пакете DBMS.SCHEDULER

- **EXECUTE DBMS\_SCHEDULER.RUN\_JOB:** Запуск заданий в пакете DBMS.SCHEDULER
- **EXECUTE DBMS\_SCHEDULER.STOP\_JOB:** Остановка заданий в пакете DBMS.SCHEDULER
- **EXECUTE DBMS\_SCHEDULER.ENABLE\_JOB:** Включение заданий в пакете DBMS.SCHEDULER
- **EXECUTE DBMS\_SCHEDULER.DISABLE\_JOB:** Отключение заданий в пакете DBMS.SCHEDULER

### Дополнительно

Помимо вышеперечисленных привилегий, пользователю MATTAKVSHI также были предоставлены следующие привилегии:

- **DBA:** Полный доступ ко всем объектам базы данных
- **RESOURCE:** Управление ресурсами базы данных
- **CONNECT:** Подключение к базе данных

## 2.3 Требования к интерфейсам

Отсутствуют в связи с отсутствием таковых.

## 2.4 Требования к миграции и изменению поток данных

### Миграция данных

При миграции данных из существующих систем в новую базу данных необходимо учитывать следующие требования:

- **Полнота данных:** Все данные из существующих систем должны быть перенесены в новую базу данных без каких-либо потерь.
- **Точность данных:** Данные должны быть перенесены точно, без каких-либо ошибок или искажений.
- **Согласованность данных:** Данные должны быть перенесены согласованно, чтобы поддерживать целостность базы данных.

- Минимизация времени простоя: Миграция данных должна быть выполнена с минимальным временем простоя для существующих систем.

## Изменение потоков данных

При изменении потоков данных необходимо учитывать следующие требования:

- Эффективность: Потоки данных должны быть оптимизированы для обеспечения высокой производительности и минимального использования ресурсов.
- Надежность: Потоки данных должны быть надежными и гарантировать доставку данных без ошибок.
- Масштабируемость: Потоки данных должны быть масштабируемыми, чтобы справляться с растущим объемом данных.
- Безопасность: Потоки данных должны быть защищены от несанкционированного доступа и утечки данных.

## Процесс миграции и изменения потоков данных

Процесс миграции и изменения потоков данных должен включать следующие шаги:

1. Планирование: Определить объем и сложность миграции, а также разработать план миграции.
2. Подготовка данных: Подготовить данные для миграции, включая очистку и преобразование данных.
3. Миграция данных: Перенести данные из существующих систем в новую базу данных.
4. Тестирование данных: Проверить точность и целостность перенесенных данных.
5. Изменение потоков данных: Изменить потоки данных в соответствии с новыми требованиями.
6. Мониторинг и поддержка: Мониторить потоки данных и предоставлять поддержку для обеспечения бесперебойной работы.

## Инструменты и технологии

Для миграции и изменения потоков данных можно использовать различные инструменты и технологии, такие как:

- **Инструменты миграции баз данных:** Инструменты, которые автоматизируют процесс миграции данных между различными базами данных.
- **Инструменты интеграции данных:** Инструменты, которые позволяют интегрировать данные из различных источников и создавать новые потоки данных.
- **Технологии потоковой передачи данных:** Технологии, которые позволяют передавать данные в режиме реального времени, обеспечивая непрерывную доставку данных.

## Безопасность

При миграции и изменении потоков данных необходимо уделять особое внимание безопасности, чтобы предотвратить несанкционированный доступ, утечку данных и другие угрозы безопасности. Необходимо реализовать следующие меры безопасности:

- **Шифрование данных:** Шифрование данных как при хранении, так и при передаче.
- **Контроль доступа:** Ограничение доступа к данным только авторизованным пользователям.
- **Аудит и мониторинг:** Аудит и мониторинг потоков данных для обнаружения подозрительной активности.
- **Резервное копирование и восстановление:** Резервное копирование данных и возможность их восстановления в случае сбоя.

## 2.5 Нефункциональные требования

1) **Производительность:** Быстрая обработка запросов, особенно интенсивных, таких как формирование списков туристов для таможни или расселения в гостинице.

2) **Доступность:** Постоянная доступность, особенно для интенсивных запросов.

3) **Надежность:** Гарантия целостности данных, особенно для интенсивных запросов.

- 4) Масштабируемость: Возможность поддерживать растущие объемы данных и запросов.
- 5) Безопасность: Защита данных от несанкционированного доступа, особенно для интенсивных запросов.
- 6) Гибкость: Адаптация к изменяющимся требованиям, особенно для интенсивных запросов.
- 7) Простота использования: Удобство обслуживания, особенно для интенсивных запросов.
- 8) Эффективность: Эффективное использование ресурсов, особенно для интенсивных запросов.
- 9) Прозрачность: Понятная и прозрачная архитектура, особенно для интенсивных запросов.
- 10) Документация: Полная и подробная документация, особенно для интенсивных запросов.
- 11) Безопасность: Защита данных от несанкционированного доступа, особенно для интенсивных запросов.

## 2.6 Справочная информация по объектам (СИ)

Объект/инструмент	Описание объекта
Таблица <b>Tour_card</b>	<p><b>Назначение:</b> содержит информацию о туристической карте, связывающей различные аспекты тура.</p> <p><b>Тип таблицы:</b> Heap</p> <p><b>Табличное пространство:</b> SYSAUX</p> <p><b>Принадлежность к БД:</b> SMLab</p> <p><b>Владелец объекта:</b> MATTAKVSHI</p> <p><b>Связанные объекты:</b></p> <p>Таблицы:  Tourist, Tour, Flight_card, Hotel_card, Excursion_card, Cargo_card.</p> <p>Триггеры:  trg_main_flight_cost,  trg_main_maintenance_expenses,</p>

	<p>trg_update_flight_data, calculate_main_tour_cost_expenses, <b>Атрибуты:</b></p> <ul style="list-style-type: none"> <li>○ tour_card_id   <ul style="list-style-type: none"> <li>▪ Описание   Идентификатор туристической карты</li> <li>▪ Наименование   tour_card_id</li> <li>▪ Тип   NUMBER</li> <li>▪ Ограничения целостности уровня столбца   <ul style="list-style-type: none"> <li>▪ Тип   Первичный ключ</li> <li>▪ Наименование   PK_tour_card_id</li> </ul> </li> <li>▪ Индексация   <ul style="list-style-type: none"> <li>▪ Наименование   PK_tour_card_id</li> <li>▪ Уникальность   Да</li> <li>▪ Тип   Нормальный (B* дерево)</li> </ul> </li> <li>▪ Связанные объекты   -</li> </ul> </li> <li>○ tourist_id   <ul style="list-style-type: none"> <li>▪ Описание   Идентификатор туриста</li> <li>▪ Наименование   tourist_id</li> <li>▪ Тип   NUMBER</li> <li>▪ Ограничения целостности уровня столбца   -</li> <li>▪ Индексация   -</li> <li>▪ Связанные объекты   Tourist (tourist_id)</li> </ul> </li> <li>○ tour_id   <ul style="list-style-type: none"> <li>▪ Описание   Идентификатор тура</li> <li>▪ Наименование   tour_id</li> <li>▪ Тип   NUMBER</li> <li>▪ Ограничения целостности уровня столбца   -</li> <li>▪ Индексация   -</li> <li>▪ Связанные объекты   Tour (tour_id)</li> </ul> </li> <li>○ flights_card_id   <ul style="list-style-type: none"> <li>▪ Описание   Идентификатор карточки</li> </ul> </li> </ul>
--	---

	<p>перелетов</p> <ul style="list-style-type: none"> <li>▪ Наименование   flights_card_id</li> <li>▪ Тип   NUMBER</li> <li>▪ Ограничения целостности уровня столбца   -</li> <li>▪ Индексация   -</li> <li>▪ Связанные объекты   Flight_card (flights_card_id)</li> </ul> <ul style="list-style-type: none"> <li>○ hotel_card_id   <ul style="list-style-type: none"> <li>▪ Описание   Идентификатор карточки отеля</li> <li>▪ Наименование   hotel_card_id</li> <li>▪ Тип   NUMBER</li> <li>▪ Ограничения целостности уровня столбца   -</li> <li>▪ Индексация   -</li> <li>▪ Связанные объекты   Hotel_card (hotel_card_id)</li> </ul> </li> </ul> <ul style="list-style-type: none"> <li>• main_excursions_cost   <ul style="list-style-type: none"> <li>○ Описание   Стоимость основных экскурсий</li> <li>○ Наименование   main_excursions_cost</li> <li>○ Тип   FLOAT</li> <li>○ Ограничения целостности уровня столбца   -</li> <li>○ Индексация   -</li> <li>○ Связанные объекты   -</li> </ul> </li> <li>• main_excursions_expenses   <ul style="list-style-type: none"> <li>○ Описание   Затраты на основные экскурсии</li> <li>○ Наименование   main_excursions_expenses</li> <li>○ Тип   FLOAT</li> <li>○ Ограничения целостности уровня столбца   Не может быть NULL</li> <li>○ Индексация   -</li> <li>○ Связанные объекты   -</li> </ul> </li> <li>• cargo_traffic_cost   <ul style="list-style-type: none"> <li>○ Описание   Стоимость грузового трафика</li> <li>○ Наименование   cargo_traffic_cost</li> </ul> </li> </ul>
--	---



	<ul style="list-style-type: none"> <li>○ Тип   FLOAT</li> <li>○ Ограничения целостности уровня столбца   -</li> <li>○ Индексация   -</li> <li>○ Связанные объекты   -</li> <li>• cargo_traffic_expenses   <ul style="list-style-type: none"> <li>○ Описание   Затраты на грузовой трафик</li> <li>○ Наименование   cargo_traffic_expenses</li> <li>○ Тип   FLOAT</li> <li>○ Ограничения целостности уровня столбца   Не может быть NULL</li> <li>○ Индексация   -</li> <li>○ Связанные объекты   -</li> </ul> </li> <li>• main_tour_cost   <ul style="list-style-type: none"> <li>○ Описание   Стоимость основного тура</li> <li>○ Наименование   main_tour_cost</li> <li>○ Тип   FLOAT</li> <li>○ Ограничения целостности уровня столбца   Не может быть NULL</li> <li>○ Индексация   -</li> <li>○ Связанные объекты   -</li> </ul> </li> <li>• main_tour_expenses   <ul style="list-style-type: none"> <li>○ Описание   Затраты на основной тур</li> <li>○ Наименование   main_tour_expenses</li> <li>○ Тип   FLOAT</li> <li>○ Ограничения целостности уровня столбца   Не может быть NULL</li> <li>○ Индексация   -</li> <li>○ Связанные объекты   -</li> </ul> </li> </ul>
Таблица <b>Flight</b>	<p><b>Назначение:</b> содержит всю информацию о рейсах.</p> <p><b>Тип таблицы:</b> Heap</p> <p><b>Табличное пространство:</b> SYSAUX</p> <p><b>Принадлежность к БД:</b> SMLab</p> <p><b>Владелец объекта:</b> MATTAKVSHI</p> <p><b>Связанные объекты:</b></p> <p>Таблицы:</p> <p>Flight_card, Cargo</p>

	<p>Триггеры:  trg_main_flight_cost,  trg_main_maintenance_expenses,   trg_update_flight_data</p> <p>Пользовательские типы: PlaneType</p> <p><b>Атрибуты:</b></p> <ul style="list-style-type: none"> <li>• flight_id <ul style="list-style-type: none"> <li>○ Описание: Уникальный идентификатор рейса</li> <li>○ Тип: NUMBER</li> <li>○ Ограничения целостности уровня столбца: Первичный ключ</li> </ul> </li> <li>• airline_name <ul style="list-style-type: none"> <li>○ Описание: Название авиакомпании</li> <li>○ Тип: VARCHAR2(600)</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• where_from_country <ul style="list-style-type: none"> <li>○ Описание: Страна отправления</li> <li>○ Тип: VARCHAR2(200)</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• where_from_city <ul style="list-style-type: none"> <li>○ Описание: Город отправления</li> <li>○ Тип: VARCHAR2(200)</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• where_country <ul style="list-style-type: none"> <li>○ Описание: Страна прибытия</li> <li>○ Тип: VARCHAR2(200)</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• where_city <ul style="list-style-type: none"> <li>○ Описание: Город прибытия</li> <li>○ Тип: VARCHAR2(200)</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• departure_datetime <ul style="list-style-type: none"> <li>○ Описание: Дата и время отправления</li> <li>○ Тип: TIMESTAMP</li> </ul> </li> </ul>
--	--

	<ul style="list-style-type: none"> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> <li>• datetime_of_arrival <ul style="list-style-type: none"> <li>○ Описание: Дата и время прибытия</li> <li>○ Тип: TIMESTAMP</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• capacity_of_people <ul style="list-style-type: none"> <li>○ Описание: Количество пассажирских мест</li> <li>○ Тип: NUMBER</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• occupied_places <ul style="list-style-type: none"> <li>○ Описание: Количество занятых пассажирских мест</li> <li>○ Тип: NUMBER</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• capacity_weight_of_cargo <ul style="list-style-type: none"> <li>○ Описание: Максимальная грузоподъемность</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• occupied_baggage_weight <ul style="list-style-type: none"> <li>○ Описание: Текущая нагрузка багажа</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• capacity_space_of_cargo <ul style="list-style-type: none"> <li>○ Описание: Количество мест для багажа (1 место = 1 см<sup>3</sup>)</li> <li>○ Тип: NUMBER</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• occupied_baggage_space <ul style="list-style-type: none"> <li>○ Описание: Количество занятых мест для багажа (1 место = 1 см<sup>3</sup>)</li> <li>○ Тип: NUMBER</li> </ul> </li> </ul>
--	---

	<ul style="list-style-type: none"> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> <li>• flight_ticket_cost <ul style="list-style-type: none"> <li>○ Описание: Стоимость билета на рейс</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• maintenance_expenses <ul style="list-style-type: none"> <li>○ Описание: Затраты на обслуживание</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> <li>• plane_type <ul style="list-style-type: none"> <li>○ Описание: Тип самолета</li> <li>○ Тип: PLANETYPE</li> <li>○ Ограничения целостности уровня столбца: Обязательное поле (NOT NULL)</li> </ul> </li> </ul> <p>Ограничения целостности:</p> <ul style="list-style-type: none"> <li>• departure_datetime_less_than_datetime_of_arrival <ul style="list-style-type: none"> <li>○ Описание: Дата и время отправления должны быть меньше или равны дате и времени прибытия (с учетом разницы в 1 час)</li> </ul> </li> <li>• occupied_places_check <ul style="list-style-type: none"> <li>○ Описание: Количество занятых пассажирских мест не должно превышать общее количество пассажирских мест</li> </ul> </li> <li>• occupied_baggage_weight_check <ul style="list-style-type: none"> <li>○ Описание: Текущая нагрузка багажа не должна превышать максимальную грузоподъемность</li> </ul> </li> <li>• occupied_baggage_space_check <ul style="list-style-type: none"> <li>○ Описание: Количество занятых мест для багажа не должно превышать общее количество мест для багажа</li> </ul> </li> </ul> <p>Пользовательский тип данных:</p> <ul style="list-style-type: none"> <li>• PlaneType</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>○ Описание: Пользовательский тип данных для определения типа самолета</li> <li>○ Тип: OBJECT</li> <li>○ Атрибуты: <ul style="list-style-type: none"> <li>▪ type_value <ul style="list-style-type: none"> <li>▪ Описание: Значение типа самолета</li> <li>▪ Тип: VARCHAR2(100)</li> </ul> </li> <li>▪ is_valid() <ul style="list-style-type: none"> <li>▪ Описание: Метод для проверки допустимости значения типа самолета</li> <li>▪ Тип: BOOLEAN</li> </ul> </li> <li>▪ validate_type() <ul style="list-style-type: none"> <li>▪ Описание: Метод для проверки исключительных ситуаций при недопустимом значении типа самолета</li> </ul> </li> <li>▪ to_string() <ul style="list-style-type: none"> <li>▪ Описание: Метод для преобразования типа самолета в строку.</li> </ul> </li> </ul> </li> </ul>
Таблица <b>Cargo</b>	<p><b>Назначение:</b> содержит всю информацию о грузообороте склада.</p> <p><b>Тип таблицы:</b> Heap</p> <p><b>Табличное пространство:</b> SYSAUX</p> <p><b>Принадлежность к БД:</b> SMLab</p> <p><b>Владелец объекта:</b> MATTAKVSHI</p> <p><b>Связанные объекты:</b></p> <p>Таблицы: Flight_card, Cargo_card, Tour_card, Flight</p> <p>Триггеры: trg_update_flight_data</p> <p>Пользовательские типы: CargoType</p> <p><b>Атрибуты:</b></p> <ul style="list-style-type: none"> <li>• cargo_id <ul style="list-style-type: none"> <li>○ Описание: Уникальный идентификатор груза</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Тип: NUMBER</li> <li>○ Ограничения целостности уровня столбца: Первичный ключ</li> <li>• cargo_name <ul style="list-style-type: none"> <li>○ Описание: Название груза</li> <li>○ Тип: VARCHAR2(400)</li> <li>○ Ограничения целостности уровня столбца: Не допускаются значения NULL</li> </ul> </li> <li>• cargo_description <ul style="list-style-type: none"> <li>○ Описание: Описание груза</li> <li>○ Тип: VARCHAR2(600)</li> <li>○ Ограничения целостности уровня столбца: Допускаются значения NULL</li> </ul> </li> <li>• receipt_date <ul style="list-style-type: none"> <li>○ Описание: Дата получения груза</li> <li>○ Тип: TIMESTAMP</li> <li>○ Ограничения целостности уровня столбца: Не допускаются значения NULL</li> </ul> </li> <li>• departure_date <ul style="list-style-type: none"> <li>○ Описание: Дата отправки груза</li> <li>○ Тип: TIMESTAMP</li> <li>○ Ограничения целостности уровня столбца: Не допускаются значения NULL, проверка <code>departure_date &gt;= receipt_date + INTERVAL '1' HOUR</code></li> </ul> </li> <li>• cargo_weight <ul style="list-style-type: none"> <li>○ Описание: Вес груза</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Не допускаются значения NULL</li> </ul> </li> <li>• cargo_width <ul style="list-style-type: none"> <li>○ Описание: Ширина груза</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Не допускаются значения NULL</li> </ul> </li> <li>• cargo_height <ul style="list-style-type: none"> <li>○ Описание: Высота груза</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца:</li> </ul> </li> </ul>
--	--

	<p>Не допускаются значения NULL</p> <ul style="list-style-type: none"> <li>• cargo_depth <ul style="list-style-type: none"> <li>○ Описание: Глубина груза</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Не допускаются значения NULL</li> </ul> </li> <li>• occupied_space <ul style="list-style-type: none"> <li>○ Описание: Занимаемое пространство груза (вычисляемый столбец)</li> <li>○ Тип: NUMBER</li> <li>○ Ограничения целостности уровня столбца: Виртуальный столбец, значение вычисляется автоматически: <math>\text{CEIL}(\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * 0.01)</math></li> </ul> </li> <li>• cargo_package_cost <ul style="list-style-type: none"> <li>○ Описание: Стоимость упаковки груза (вычисляемый столбец)</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Виртуальный столбец, значение вычисляется автоматически: <math>\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * 0.002</math></li> </ul> </li> <li>• cargo_insurance_cost <ul style="list-style-type: none"> <li>○ Описание: Стоимость страховки груза (вычисляемый столбец)</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Виртуальный столбец, значение вычисляется автоматически: <math>\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * \text{cargo\_width} * 0.002</math></li> </ul> </li> <li>• cargo_transportation_cost <ul style="list-style-type: none"> <li>○ Описание: Стоимость транспортировки груза (вычисляемый столбец)</li> <li>○ Тип: FLOAT</li> <li>○ Ограничения целостности уровня столбца: Виртуальный столбец, значение вычисляется автоматически: <math>\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * \text{cargo\_width} *</math></li> </ul> </li> </ul>
--	---

	<p>0.01</p> <ul style="list-style-type: none"> <li>expenses <ul style="list-style-type: none"> <li>Описание: Общие расходы на груз (вычисляемый столбец)</li> <li>Тип: FLOAT</li> <li>Ограничения целостности уровня столбца: Виртуальный столбец, значение вычисляется автоматически: <math>(\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * 0.002) + (\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * \text{cargo\_width} * 0.002) + (\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * \text{cargo\_width} * 0.01)</math></li> </ul> </li> <li>total_cost <ul style="list-style-type: none"> <li>Описание: Общая стоимость груза (вычисляемый столбец)</li> <li>Тип: FLOAT</li> <li>Ограничения целостности уровня столбца: Виртуальный столбец, значение вычисляется автоматически: <math>(((\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * 0.002) + (\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * \text{cargo\_width} * 0.002) + (\text{cargo\_width} * \text{cargo\_height} * \text{cargo\_depth} * \text{cargo\_width} * 0.01)) * 1.1)</math></li> </ul> </li> <li>cargo_type <ul style="list-style-type: none"> <li>Описание: Тип груза</li> <li>Тип: CARGOTYPE (пользовательский тип данных)</li> <li>Ограничения целостности уровня столбца: Не допускаются значения NULL, проверка на допустимые значения</li> </ul> </li> </ul> <p>Связанные объекты:</p> <ul style="list-style-type: none"> <li>Пользовательский тип данных CARGOTYPE</li> </ul>
Триггер <b>trg_update_flight_data</b>	<p><b>Назначение:</b> Обновление данных в таблице Flight при добавлении записи в таблицу Tour_card.</p> <p>Установлен на таблицу: Tour_card</p> <p><b>Тип:</b> AFTER INSERT OR UPDATE</p>

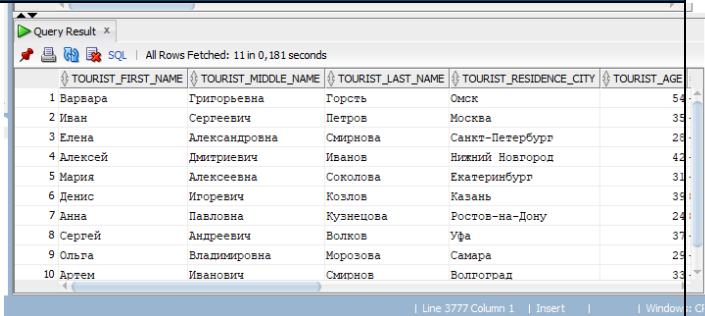


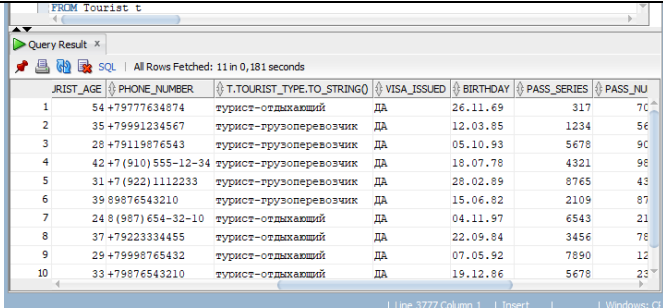
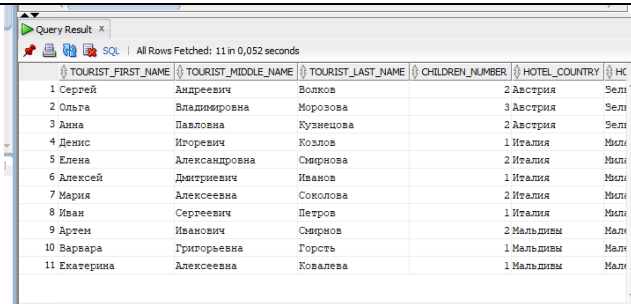
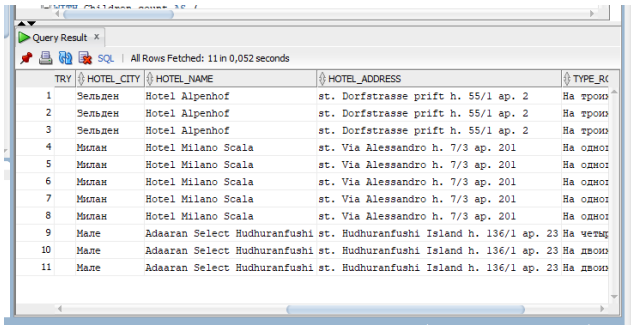
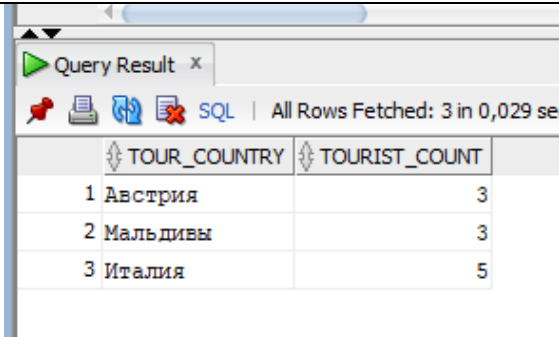
	<p><b>Для каждой строки:</b> Да</p> <p><b>Владелец объекта:</b> МАТТАКVSHI</p> <p><b>Связанные объекты:</b>  Таблицы: Flight, Flight_card, Tour_card, Cargo, Carogo_card, Tourist_Children</p> <p><b>Реализуемый алгоритм:</b>  Для каждой новой или обновленной строки в таблице Tour_card выполняются следующие действия:</p> <ol style="list-style-type: none"> <li>1. Подсчет количества детей для данного туриста и сохранение в переменную v_tourist_children_count.</li> <li>2. Обновление поля occupied_places для ПЕРВОГО рейса, добавляя 1 и значение v_tourist_children_count к текущему значению поля, используя значение flights_card_id из новой строки Tour_card.</li> <li>3. Обновление поля occupied_places для ВТОРОГО рейса, аналогично шагу 2.</li> <li>4. Получение суммарного веса груза для ПЕРВОГО рейса, учитывая только груз, полученный до двух дней до времени отправления рейса. Результат сохраняется в переменную v_cargo_weight.</li> <li>5. Обновление поля occupied_baggage_weight для ПЕРВОГО рейса, добавляя значение v_cargo_weight к текущему значению поля, используя значение flights_card_id из новой строки Tour_card.</li> <li>6. Получение суммарного веса груза для ВТОРОГО рейса, аналогично шагу 4.</li> <li>7. Обновление поля occupied_baggage_weight для ВТОРОГО рейса, аналогично шагу 5.</li> <li>8. Получение суммарного занимаемого места багажа для ПЕРВОГО рейса, учитывая только груз, полученный до двух дней до времени отправления рейса. Результат сохраняется в переменную v_cargo_space.</li> <li>9. Обновление поля occupied_baggage_space для ПЕРВОГО рейса, добавляя значение v_cargo_space к текущему значению поля, используя значение flights_card_id из новой строки Tour_card.</li> <li>10. Получение суммарного занимаемого места багажа для ВТОРОГО рейса, аналогично шагу 8.</li> <li>11. Обновление поля occupied_baggage_space для ВТОРОГО рейса, аналогично шагу 9.</li> </ol>
--	---

<p>Триггер <b>group_name_trigger</b></p>	<p><b>Назначение:</b> Данный триггер предназначен для автоматической генерации наименования группы туристов на основе названия тура и даты его начала.  <b>Установлен на таблицу:</b> Tourists_group  <b>Тип:</b> BEFORE INSERT  <b>Для каждой строки:</b> да  <b>Владелец объекта:</b> MATTAKVSHI  <b>Связанные объекты:</b>          Таблица: Tourist_group  <b>Реализуемый алгоритм:</b></p> <ol style="list-style-type: none"> <li>1. При вставке новой строки в таблицу Tourists_group, перед выполнением операции вставки, триггер срабатывает для каждой строки.</li> <li>2. Внутри триггера выполняется SQL-запрос, который извлекает название тура и дату его начала из таблицы Tour, используя идентификатор тура из вставляемой строки.</li> <li>3. Полученные значения сохраняются в переменные tour_name и tour_start_date.</li> <li>4. Значение поля group_name новой строки устанавливается в соответствии с шаблоном "название тура - дата начала", где дата начала преобразуется в строку в формате 'YYYY-MM-DD'.</li> <li>5. Вставка новой строки в таблицу Tourists_group продолжается с уже измененным значением поля group_name.</li> </ol>
<p>Job <b>check_hotel_card_job</b></p>	<p><b>Назначение:</b> Данное задание предназначено для проверки таблицы Hotel_card и обновления таблицы Hotel_rooms в соответствии с результатами проверки.  <b>Тип:</b> PLSQL_BLOCK  <b>Владелец объекта:</b> MATTAKVSHI  <b>Связанные объекты:</b> Hotel_card, Hotel_rooms  <b>Реализуемый алгоритм:</b></p> <ol style="list-style-type: none"> <li>1. Внутри блока PL/SQL задания выполняется цикл для каждой записи в таблице Hotel_card, где eviction_date (дата выселения) меньше или</li> </ol>

	<p>равна текущей дате и больше, чем дата, предшествующая текущей дате на 1 день.</p> <ol style="list-style-type: none"> <li>Внутри цикла происходит обновление таблицы Hotel_rooms. Уменьшается количество занятых номеров (number_of_occupied_rooms) в соответствующей записи таблицы Hotel_rooms для отеля, указанного в booked_hotel_id, и зарезервированного номера, указанного в reserved_room_id.</li> <li>Задание запускается сразу после создания (start_date = SYSDATE) и повторяется ежедневно в 12:10 (repeat_interval = 'FREQ=DAILY; BYHOUR=12; BYMINUTE=10;').</li> <li>Задание включено (enabled = TRUE).</li> <li>В комментариях указано, что задание предназначено для проверки таблицы Hotel_card и обновления таблицы Hotel_rooms соответствующим образом.</li> </ol>
--	---

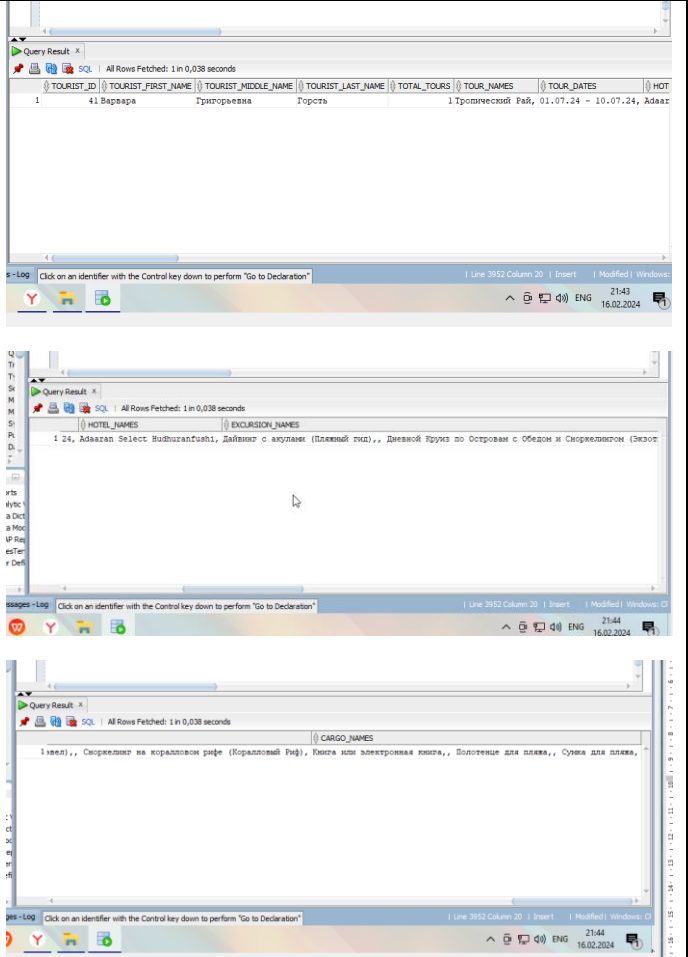
## 2.6 Порядок и виды тестирования

Что проверяем	Как проверяем	Результат
Возможность сформировать список туристов для таможни в целом и по указанной категории.	По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.	

		
Возможность сформировать списки на расселение указанным гостиницам в целом и указанной категории.	По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.	 
Возможность получить количество туристов, побывавших в стране за определенный период в целом и по определенной категории.	По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.	

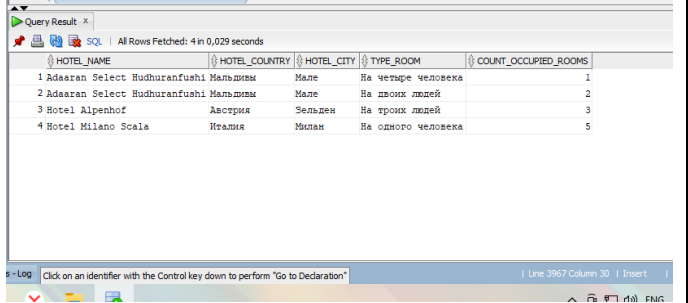
Возможность получить сведения о конкретном туристе: сколько раз был в стране, даты прилета/отлета, в каких гостиницах останавливался, какие экскурсии и в каких агентствах заказывал, какой груз сдавал.

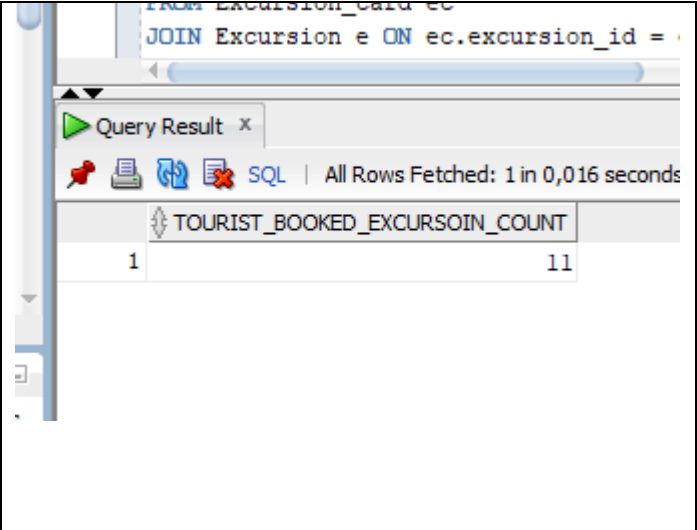
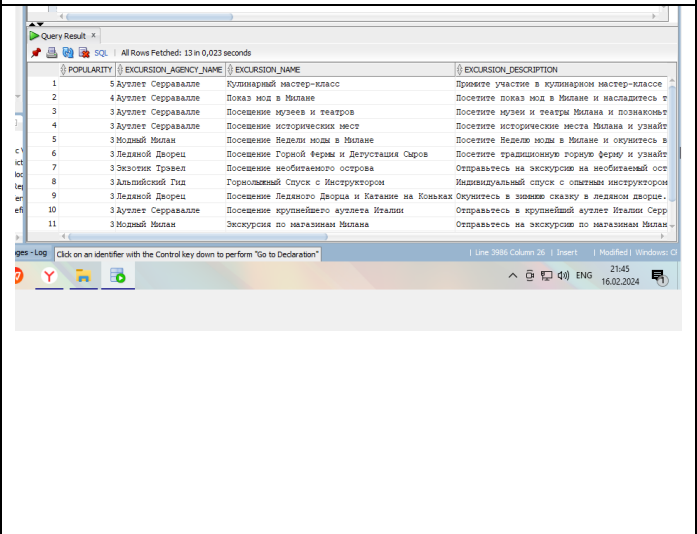
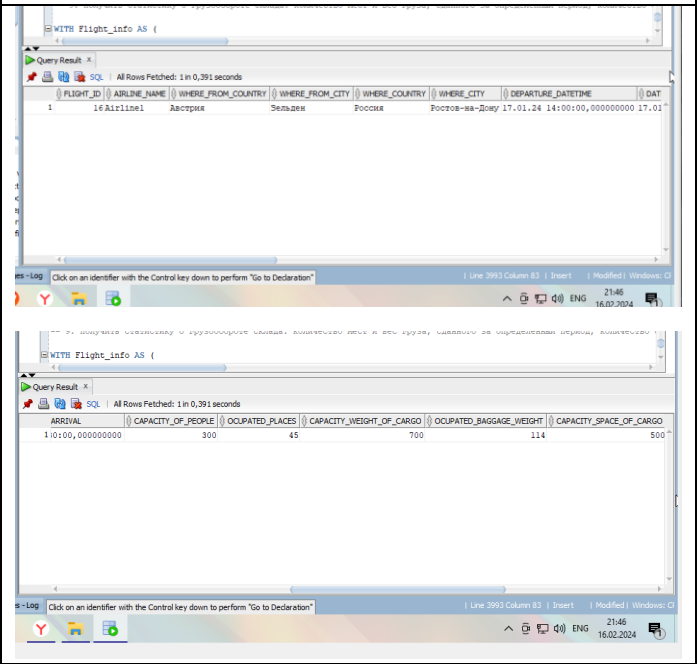
По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.

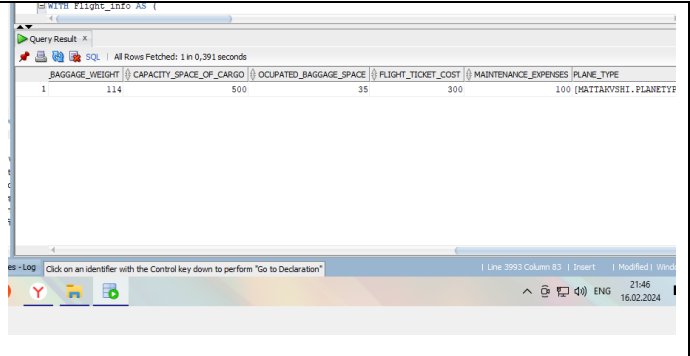
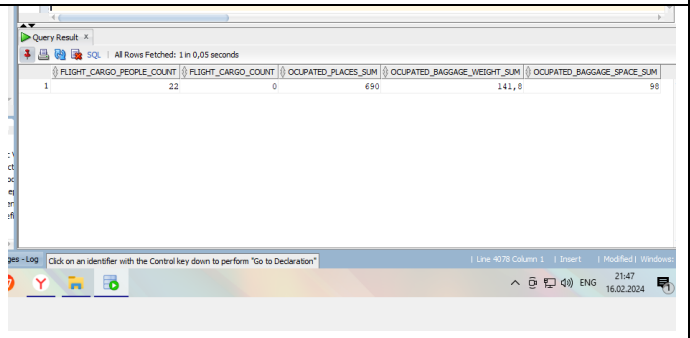
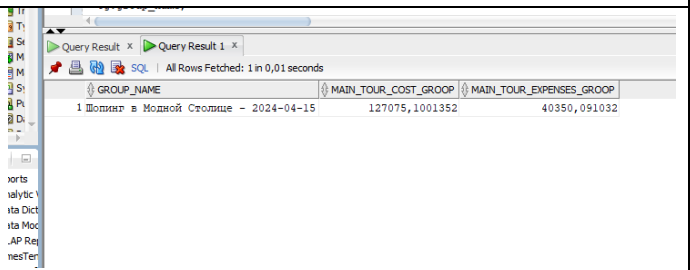


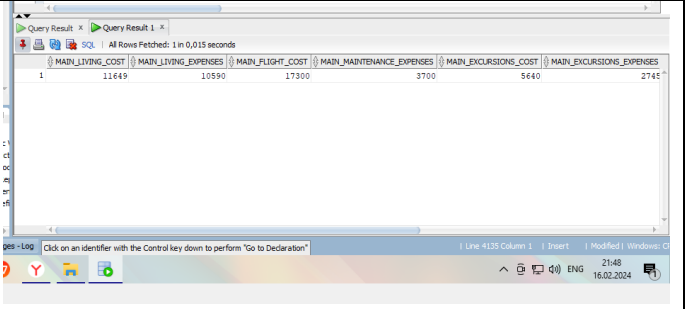
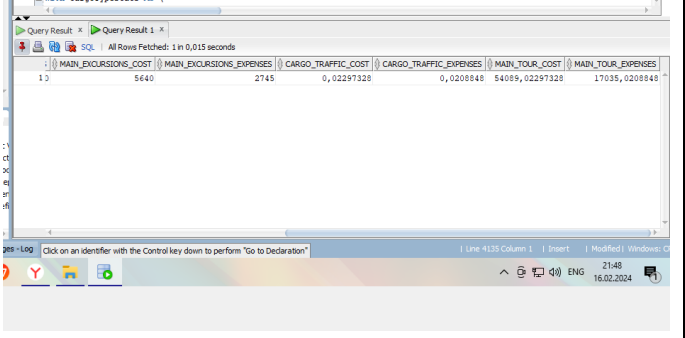
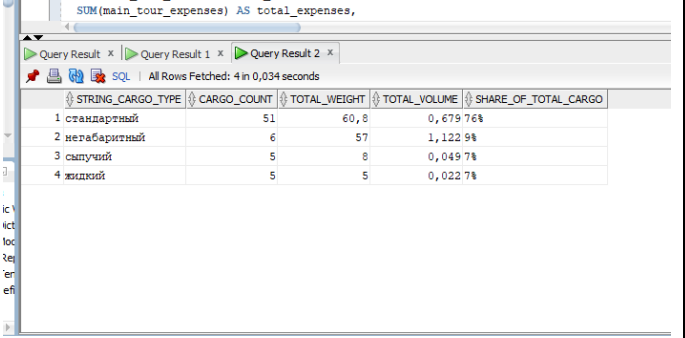
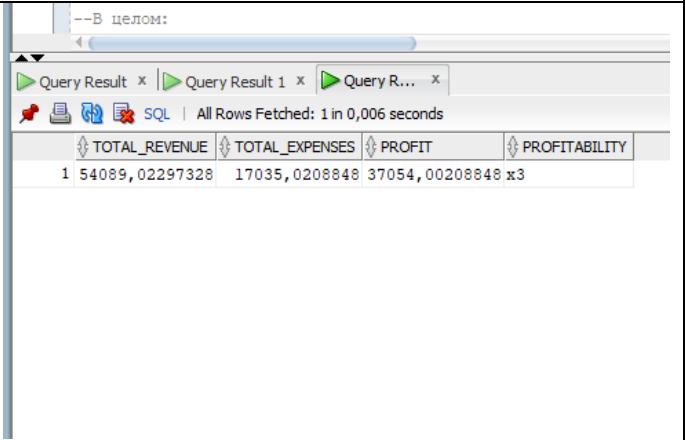
Возможность получить список гостиниц, в которых производится расселение туристов, указанием количества занимаемых номеров и проживавших в них человек за определенный период.

По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.



<p>Возможность получить общее количество туристов, заказавших экскурсии за определенный период.</p>	<p>По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.</p>	
<p>Возможность выбрать самые популярные экскурсии и самые качественные экскурсионные агентства.</p>	<p>По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.</p>	
<p>Возможность получить данные о загрузке указанного рейса самолета на определенную дату: количество мест, вес груза, объемный вес.</p>	<p>По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.</p>	

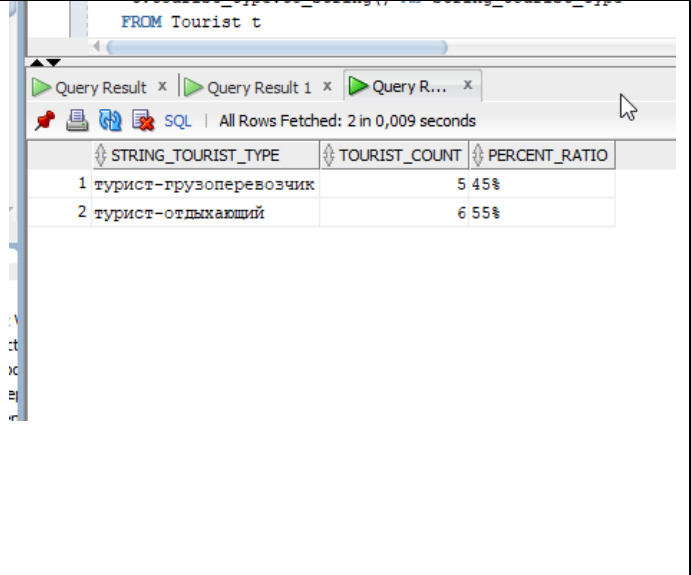
		 <p>The screenshot shows a SQL query result with columns: BAGGAGE_WEIGHT, CAPACITY_SPACE_OF_CARGO, OCCUPATED_BAGGAGE_SPACE, FLIGHT_TICKET_COST, MAINTENANCE_EXPENSES, and PLANE_TYPE. The first row shows values: 1, 114, 500, 35, 300, 100 [HATTARYSHI, PLANETYP].</p>
Возможность получить статистику о грузообороте склада: количество мест и вес груза, сданного за определенный период, количество самолетов, вывозивших этот груз, сколько из них грузовых, а сколько грузопассажирских .	По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.	 <p>The screenshot shows a SQL query result with columns: FLIGHT_CARGO_PEOPLE_COUNT, FLIGHT_CARGO_COUNT, OCCUPATED_PLACES_SUM, OCCUPATED_BAGGAGE_WEIGHT_SUM, and OCCUPATED_BAGGAGE_SPACE_SUM. The first row shows values: 1, 22, 0, 690, 141, 8, 98.</p>
Возможность получить полный финансовый отчет по указанной группе в целом и для определенной категории туристов.	По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.	 <p>The screenshot shows a SQL query result with columns: GROUP_NAME, MAIN_TOUR_COST_GROOP, and MAIN_TOUR_EXPENSES_GROOP. The first row shows values: 1 Шопинг в Модной Столице - 2024-04-15, 127075,1001352, 40350,091032.</p>

<p>Возможность получить данные о расходах и доходах за определенный период: обслуживание самолета, гостиница, экскурсии, визы, расходы представительства и т.п.</p>	<p>По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.</p>	 
<p>Возможность получить статистику по видам отправляемого груза и удельную долю каждого вида в общем грузопотоке.</p>	<p>По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.</p>	
<p>Возможность вычислить рентабельность представительства (соотношение доходов и расходов).</p>	<p>По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.</p>	



Возможность определить процентное отношение отдыхающих туристов к туристам shop-туров в целом и за указанный период (например, в зависимости от времени года).

По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.

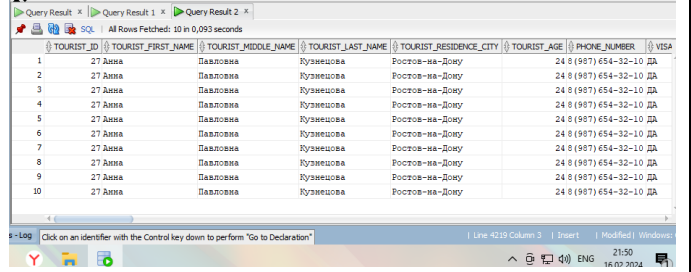


FROM Tourist t

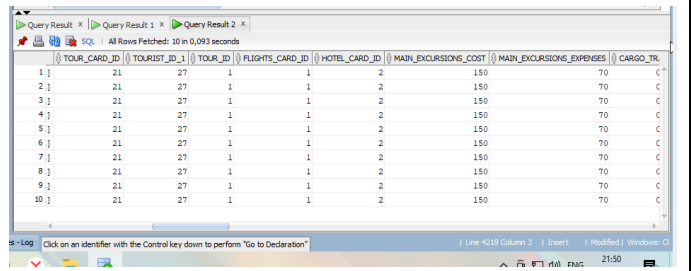
	STRING_TOURIST_TYPE	TOURIST_COUNT	PERCENT_RATIO
1	турист-грузоперевозчик	5	45%
2	турист-отдыхающий	6	55%

Возможность получить сведения о туристах указанного рейса: список группы, гостиницы, груз, бирки, маркировка.

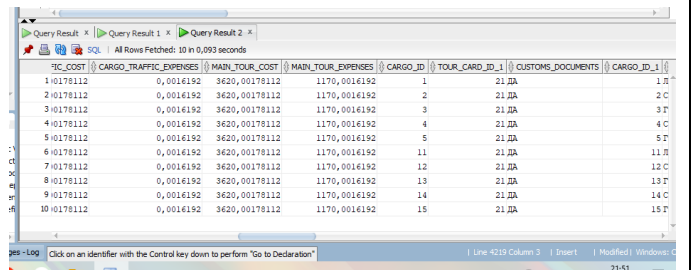
По средством SELECT запроса, текст запроса представлен в последнем разделе приложения с. 148.



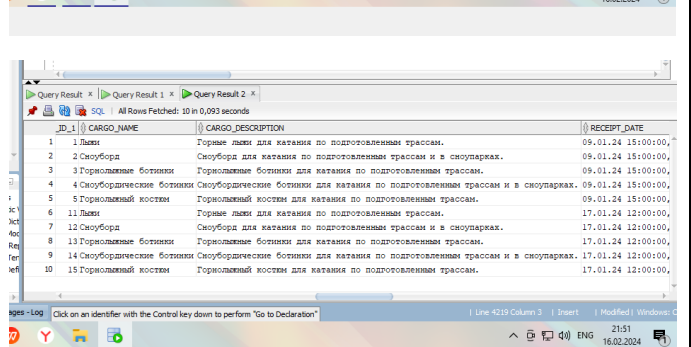
	TOURIST_ID	TOURIST_FIRST_NAME	TOURIST_MIDDLE_NAME	TOURIST_LAST_NAME	TOURIST_RESIDENCE_CITY	TOURIST_AGE	PHONE_NUMBER	VISA
1	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
2	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
3	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
4	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
5	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
6	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
7	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
8	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
9	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА
10	27	Анна	Павловна	Кузнецова	Ростов-на-Дону	24	8 (987) 654-32-10	ДА



	TOUR_CARD_ID	TOURIST_ID	TOUR_ID	FLIGHTS_CARD_ID	HOTEL_CARD_ID	MAIN_EXCURSIONS_COST	MAIN_EXCURSIONS_EXPENSES	CARGO_TR
1	21	27	1	1	2	150	70	C
2	21	27	1	1	2	150	70	C
3	21	27	1	1	2	150	70	C
4	21	27	1	1	2	150	70	C
5	21	27	1	1	2	150	70	C
6	21	27	1	1	2	150	70	C
7	21	27	1	1	2	150	70	C
8	21	27	1	1	2	150	70	C
9	21	27	1	1	2	150	70	C
10	21	27	1	1	2	150	70	C



	TC_COST	CARGO_TRAFFIC_EXPENSES	MAIN_TOUR_COST	MAIN_TOUR_EXPENSES	CARGO_ID	TOUR_CARD_ID_1	CUSTOMS_DOCUMENTS	CARGO_ID_1
1	0178112	0,0016192	3620,00178112	1170,0016192	1	21,ДА		1,П
2	0178112	0,0016192	3620,00178112	1170,0016192	2	21,ДА		2,С
3	0178112	0,0016192	3620,00178112	1170,0016192	3	21,ДА		3,Т
4	0178112	0,0016192	3620,00178112	1170,0016192	4	21,ДА		4,С
5	0178112	0,0016192	3620,00178112	1170,0016192	5	21,ДА		5,Т
6	0178112	0,0016192	3620,00178112	1170,0016192	11	21,ДА		11,Л
7	0178112	0,0016192	3620,00178112	1170,0016192	12	21,ДА		12,С
8	0178112	0,0016192	3620,00178112	1170,0016192	13	21,ДА		13,Т
9	0178112	0,0016192	3620,00178112	1170,0016192	14	21,ДА		14,С
10	0178112	0,0016192	3620,00178112	1170,0016192	15	21,ДА		15,Т



	ID_1	CARGO_NAME	CARGO_DESCRIPTION	RECEIPT_DATE
1	1	Льки	Горные льки для катания по подготовленным трассам.	09.01.24 15:00:00
2	2	Сноуборд	Сноуборд для катания по подготовленным трассам и в сноупарках.	09.01.24 15:00:00
3	3	Горнолыжные ботинки	Горнолыжные ботинки для катания по подготовленным трассам.	09.01.24 15:00:00
4	4	Сноубордические ботинки	Сноубордические ботинки для катания по подготовленным трассам и в сноупарках.	09.01.24 15:00:00
5	5	Горнолыжный костюм	Горнолыжный костюм для катания по подготовленным трассам.	09.01.24 15:00:00
6	11	Льки	Горные льки для катания по подготовленным трассам.	17.01.24 12:00:00
7	12	Сноуборд	Сноуборд для катания по подготовленным трассам и в сноупарках.	17.01.24 12:00:00
8	13	Горнолыжные ботинки	Горнолыжные ботинки для катания по подготовленным трассам.	17.01.24 12:00:00
9	14	Сноубордические ботинки	Сноубордические ботинки для катания по подготовленным трассам и в сноупарках.	17.01.24 12:00:00
10	15	Горнолыжный костюм	Горнолыжный костюм для катания по подготовленным трассам.	17.01.24 12:00:00



### 3 План внедрения

#### Этап 1: Анализ требований

- Провести тщательный анализ требований, изложенных в техническом задании.
- Определить функциональные и нефункциональные требования к базе данных.
- Определить заинтересованные стороны и их потребности в данных.

#### Этап 2: Проектирование базы данных

- Выбрать подходящую модель данных (реляционная).
- Провести декомпозицию предметной области и изобразить все бизнес-процессы в UML-диаграммах.
- Разработать схему базы данных, которая будет поддерживать все необходимые сущности и отношения.
- Определить поля, типы данных, ограничения и индексы.

#### Этап 3: Создание базы данных

- Создать базу данных в выбранной системе управления базами данных (СУБД Oracle).
- Настроить параметры СУБД для обеспечения оптимальной производительности и надежности.

#### Этап 4: Заполнение базы данных

- Разработать процедуры импорта данных для заполнения базы данных существующими данными.
- Обеспечить механизмы ввода данных для добавления новых данных в базу данных.

#### Этап 5: Тестирование и развертывание

- Провести всестороннее тестирование базы данных.
- Развернуть базу данных производственной среде.

#### Этап 6: Обучение и поддержка

- Обучить пользователей работе с базой данных и пользовательским интерфейсом.

- Предоставить документацию и поддержку для обеспечения бесперебойной работы.

#### Этап 7: Мониторинг и обслуживание

- Регулярно отслеживать производительность базы данных.
- Проводить техническое обслуживание для обеспечения оптимальной работы и безопасности.

#### Дополнительные соображения

- Рассмотреть использование инструментов резервного копирования и восстановления для защиты данных.
- Обеспечить безопасность базы данных с помощью мер контроля доступа и шифрования.
- Рассмотреть возможность интеграции с другими системами, такими как система управления взаимоотношениями с клиентами (CRM) или система бронирования.

## ПРИЛОЖЕНИЕ

-- Установка для работы с датами

```
ALTER SESSION SET nls_territory='RUSSIA';
ALTER SESSION SET nls_language='RUSSIAN';
ALTER DATABASE SET TIME_ZONE = 'Europe/Moscow'; -- запускал с
SYS`а
```

-- СОЗДАНИЕ ВСЕХ ТАБЛИЦ -----

-- Создание таблицы туристов и последовательности и триггера для реализации авто инкремента первого столбца

```
CREATE TABLE Tourist (
  tourist_id NUMBER NOT NULL,
  tourist_first_name VARCHAR2(300) NOT NULL,
  tourist_middle_name VARCHAR2(300) NOT NULL,
  tourist_last_name VARCHAR2(300) NOT NULL,
  tourist_residence_city VARCHAR2((300) NOT NULL,
  tourist_age NUMBER NOT NULL,
  phone_number VARCHAR2(50) NOT NULL,
  visa_issued CHAR(10) NOT NULL DEFAULT 'HET' CHECK
(visa_issued IN ('ДА', 'HET')),
  tourist_type TOURISTTYPE NOT NULL,

  CONSTRAINT phone_number_check CHECK
(REGEXP_LIKE(phone_number, '^(+7|8)?[-.\s]?(\d{3}))?[-.\s]?d{3}[-.\s]?d{2}[-.\s]?d{2}'))
);
```

-- Добавление автоинкрементного первого столбца с помощью последовательности и триггера

```
ALTER TABLE Tourist ADD (
  CONSTRAINT tourist_pk PRIMARY KEY (tourist_id));
```

```
CREATE SEQUENCE tourist_seq START WITH 1;
```

```
CREATE OR REPLACE TRIGGER tourist_bir
BEFORE INSERT ON Tourist
FOR EACH ROW
```

```

BEGIN
  SELECT tourist_seq.NEXTVAL
  INTO :new.tourist_id
  FROM dual;
END;
/

```

```

-- Создание пользовательского типа данных и определение его тела
CREATE OR REPLACE TYPE TouristType AS OBJECT (
  type_value VARCHAR2(100),
  MEMBER FUNCTION is_valid RETURN BOOLEAN,
  MEMBER PROCEDURE validate_type,
  MEMBER FUNCTION to_string RETURN VARCHAR2
) NOT FINAL;

```

```

CREATE OR REPLACE TYPE BODY TouristType AS
  MEMBER FUNCTION is_valid RETURN BOOLEAN IS
  BEGIN
    RETURN (type_value IN ('турист-грузоперевозчик', 'турист-
отдыхающий'));
  END;

```

```

  MEMBER PROCEDURE validate_type IS
  BEGIN
    IF NOT self.is_valid THEN
      RAISE_APPLICATION_ERROR(-20000, 'Недопустимое значение
для типа TouristType');
    END IF;
  END;

```

```

  MEMBER FUNCTION to_string RETURN VARCHAR2 IS
  BEGIN
    RETURN type_value;
  END;
END;

```

-- Создание таблицы детей для авто инкремента использую  
последовательность на прямую без триггера

```
CREATE SEQUENCE children_seq START WITH 1;
```

```
CREATE TABLE Children (  
  child_id NUMBER DEFAULT children_seq.nextval NOT NULL,  
  child_first_name VARCHAR2(100) NOT NULL,  
  child_middle_name VARCHAR2(100) NOT NULL,  
  child_last_name VARCHAR2(100) NOT NULL,  
  child_sex CHAR(6) NOT NULL,  
  child_age NUMBER NOT NULL,  
  
  CONSTRAINT child_sex CHECK (child_sex IN ('МУЖ', 'ЖЕН'))  
);
```

```
ALTER TABLE Children ADD (  
  CONSTRAINT children_pk PRIMARY KEY (child_id));
```

-- Создание таблицы связки детей и родителей для авто инкремента  
использую IDENTITY столбец, (который доступен с Oracle 12c +),  
далее во всех таблицах буду использовать именно этот вариант.

```
CREATE TABLE Tourist_Children (  
  tourist_children_id NUMBER GENERATED ALWAYS AS IDENTITY  
  PRIMARY KEY,  
  tourist_id NUMBER NOT NULL,  
  child_id NUMBER NOT NULL,  
  
  CONSTRAINT fk_tourist FOREIGN KEY (tourist_id) REFERENCES  
  Tourist (tourist_id) ON DELETE CASCADE,  
  CONSTRAINT fk_children FOREIGN KEY (child_id) REFERENCES  
  Children (child_id) ON DELETE CASCADE  
);
```

-- Удаляем столбец tourist\_children\_id потому что он избыточен в данной  
ситуации.

-- tourist\_id и children\_id здесь являются составным первичным ключём.

```
ALTER TABLE Tourist_Children DROP COLUMN tourist_children_id;
```

-- Создание таблицы ПАСПОРТНЫХ ДАННЫХ для авто инкремента  
используя IDENTITY столбец

```
CREATE TABLE Passport_data (  
  pass_data_id NUMBER GENERATED ALWAYS AS IDENTITY  
  PRIMARY KEY,  
  tourist_id NUMBER NOT NULL,  
  birthday DATE NOT NULL,  
  pass_series NUMBER NOT NULL,  
  pass_number NUMBER NOT NULL,  
  issued_by VARCHAR2(300) NOT NULL,  
  when_issued DATE NOT NULL,  
  sex CHAR(10) NOT NULL,  
  
  CONSTRAINT fk_tourist_for_pass FOREIGN KEY (tourist_id)  
  REFERENCES Tourist (tourist_id) ON DELETE CASCADE,  
  
  CONSTRAINT sex_check CHECK (sex IN ('МУЖ', 'ЖЕН')),  
  
  CONSTRAINT when_issued_format CHECK (when_issued =  
  TO_DATE(TO_CHAR(when_issued, 'YYYY-MM-DD'), 'YYYY-MM-DD'),  
  CONSTRAINT birthday_format CHECK (birthday =  
  TO_DATE(TO_CHAR(birthday, 'YYYY-MM-DD'), 'YYYY-MM-DD'),  
  
  CONSTRAINT pass_series_format CHECK (pass_series >= 0 AND  
  pass_series <= 9999),  
  CONSTRAINT pass_number_format CHECK (pass_number >= 0 AND  
  pass_number <= 999999)  
);
```

-- Создание триггера для обновления значения поля visa\_issued в  
таблице Tourist с учётом соответствующей записи в таблице  
Passport\_data

```
CREATE OR REPLACE TRIGGER update_visa_issued  
AFTER INSERT ON Passport_data  
FOR EACH ROW  
BEGIN  
  UPDATE Tourist  
  SET visa_issued = 'ДА'  
  WHERE tourist_id = :new.tourist_id  
  AND tourist_first_name IS NOT NULL  
  AND tourist_middle_name IS NOT NULL
```



```

AND tourist_last_name IS NOT NULL
AND tourist_residence_city IS NOT NULL
AND tourist_age IS NOT NULL
AND phone_number IS NOT NULL
AND tourist_type IS NOT NULL;
END;
/

```

-- Создание таблицы ТУРОВ для авто инкремента использую  
IDENTITY столбец, вычисляю сколько дней длится тур, по средствам  
разницы дат начала и конца

```

CREATE TABLE Tour (
  tour_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY
  KEY,
  tour_name VARCHAR2(600) NOT NULL,
  tour_description VARCHAR2(1000),
  tour_cost NUMBER NOT NULL,
  tour_country VARCHAR2(200) NOT NULL,
  tour_city VARCHAR2(200) NOT NULL,
  tour_start_date DATE NOT NULL,
  tour_end_date DATE NOT NULL,
  duration_in_days NUMBER GENERATED ALWAYS AS ((tour_end_date
- tour_start_date) + 1), --ПРОДОЛЖИТЕЛЬНОСТЬ ТУРА В ДНЯХ

  CONSTRAINT tour_date_check CHECK (tour_start_date <
tour_end_date),
  CONSTRAINT tour_duration_check CHECK (duration_in_days >= 2),

  CONSTRAINT tour_start_date_format_check CHECK (tour_start_date =
TO_DATE(TO_CHAR(tour_start_date, 'YYYY-MM-DD HH24:MI:SS'),
'YYYY-MM-DD HH24:MI:SS'),
  CONSTRAINT tour_end_date_format_check CHECK (tour_end_date =
TO_DATE(TO_CHAR(tour_end_date, 'YYYY-MM-DD HH24:MI:SS'),
'YYYY-MM-DD HH24:MI:SS')

);

```

-- Создание таблицы РЕЙСОВ для авто инкремента использую  
IDENTITY столбец

```
CREATE TABLE Flight (  
    flight_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY  
    KEY,  
    airline_name VARCHAR2(600) NOT NULL,  
    where_from_country VARCHAR2(200) NOT NULL,  
    where_from_city VARCHAR2(200) NOT NULL,  
    where_country VARCHAR2(200) NOT NULL,  
    where_city VARCHAR2(200) NOT NULL,  
    departure_datetime TIMESTAMP NOT NULL,  
    datetime_of_arrival TIMESTAMP NOT NULL,  
  
    capacity_of_people NUMBER NOT NULL, --Количество пассажирских  
мест  
    ocupated_places NUMBER DEFAULT 0 NOT NULL,      --Количество  
занятых пассажирских мест --Вычисляем триггером  
    capacity_weight_of_cargo FLOAT NOT NULL, --Максимальная  
грузоподъёмность  
    ocupated_baggage_weight FLOAT DEFAULT 0 NOT NULL, --Текущая  
нагруженность --Вычисляем триггером  
    capacity_space_of_cargo NUMBER NOT NULL, --Количество мест  
багажа (1 место 1 см^3)  
    ocupated_baggage_space NUMBER DEFAULT 0 NOT NULL, --  
Количество занятых мест багажа (1 место 1 см^3) --Вычисляем  
триггером  
  
    flight_ticket_cost FLOAT NOT NULL, -- Это вводится  
администратором  
    maintenance_expenses FLOAT NOT NULL, ---- Это вводится  
администратором  
    plane_type PLANETYPE NOT NULL,  
  
    CONSTRAINT departure_datetime_less_than_datetime_of_arrival  
CHECK (departure_datetime <= datetime_of_arrival - INTERVAL '1'  
HOUR),  
  
    CONSTRAINT ocupated_places_check CHECK (ocupated_places <=  
capacity_of_people),  
  
    CONSTRAINT ocupated_baggage_weight_check CHECK  
(ocupated_baggage_weight <= capacity_weight_of_cargo),
```

```
CONSTRAINT ocupated_baggage_space_check CHECK
(ocupated_baggage_space <= capacity_space_of_cargo)
);
```

```
-- Создание пользовательского типа данных и определение его тела
CREATE OR REPLACE TYPE PlaneType AS OBJECT (
    type_value VARCHAR2(100),
    MEMBER FUNCTION is_valid RETURN BOOLEAN,
    MEMBER PROCEDURE validate_type,
    MEMBER FUNCTION to_string RETURN VARCHAR2
) NOT FINAL;
```

```
CREATE OR REPLACE TYPE BODY PlaneType AS
    MEMBER FUNCTION is_valid RETURN BOOLEAN IS
    BEGIN
        RETURN (type_value IN ('грузовой', 'грузо-пассажирский'));
    END;
```

```
    MEMBER PROCEDURE validate_type IS
    BEGIN
        IF NOT self.is_valid THEN
            RAISE_APPLICATION_ERROR(-20000, 'Недопустимое значение
для типа PlaneType');
        END IF;
    END;
```

```
    MEMBER FUNCTION to_string RETURN VARCHAR2 IS
    BEGIN
        RETURN type_value;
    END;
END;
```

```
-- Создание таблицы Карты Рейсов для авто инкремента используя
IDENTITY столбец
```

```
CREATE TABLE Flight_card (
    flights_card_id NUMBER GENERATED ALWAYS AS IDENTITY
PRIMARY KEY,
    flight_there_id NUMBER NOT NULL,
    flight_back_id NUMBER NOT NULL,
```

```

    main_flight_cost FLOAT DEFAULT 0 NOT NULL, --Вычисляем
триггером
    main_maintenance_expenses FLOAT DEFAULT 0 NOT NULL, --
Вычисляем триггером

```

```

    CONSTRAINT fk_flight_there FOREIGN KEY (flight_there_id)
REFERENCES Flight (flight_id) ON DELETE CASCADE,
    CONSTRAINT fk_flight_back FOREIGN KEY (flight_back_id)
REFERENCES Flight (flight_id) ON DELETE CASCADE
);

```

```

-- Создание таблицы ОТЕЛЕЙ для авто инкремента использую
IDENTITY столбец, использую регулярное выражение для задания
формата адреса

```

```

CREATE TABLE Hotel (
    hotel_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY
KEY,
    hotel_name VARCHAR2(600) NOT NULL,
    hotel_country VARCHAR2(200) NOT NULL,
    hotel_city VARCHAR2(200) NOT NULL,
    hotel_address VARCHAR2(600) NOT NULL,

    CONSTRAINT chk_hotel_address CHECK
(REGEXP_LIKE(hotel_address, '^(\ул.|st.) ([A-ZА-Я][a-zA-Za-яА-
Я]*\s+[a-zA-Za-яА-Я]*) (д.|h.) ([1-9][0-9]{0,3}/[1-9][0-9]{0,1})?)
(кв.|ap.)? ([1-9][0-9]{0,3})?$',))
);

```

```

-- Создание таблицы ТИПЫ КОМНАТ

```

```

CREATE TABLE Type_of_room (
    type_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY
KEY,
    type_room VARCHAR2(40) NOT NULL
);

```

```

-- Добавление 4 типов номеров

```

```

INSERT INTO Type_of_room (type_room) VALUES ('На одного
человека');

```

```

INSERT INTO Type_of_room (type_room) VALUES ('На двоих людей');

```

```

INSERT INTO Type_of_room (type_room) VALUES ('На троих людей');

```

```

INSERT INTO Type_of_room (type_room) VALUES ('На четыре
человека');

```

--Создание таблицы, СВЯЗИ ОТЕЛЕЙ С КОМНТАМИ в которую выносим поля из таблицы отелей, для более правильной логики бд

```
CREATE TABLE Hotel_rooms (  
    hotel_rooms_id NUMBER GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
    hotel_id NUMBER NOT NULL,  
    room_type_id NUMBER NOT NULL,  
    number_of_available_rooms NUMBER NOT NULL,          --Заполняется  
    администратором  
    number_of_occupied_rooms NUMBER DEFAULT 0 NOT NULL,  --  
    Вычисляется триггером  
    room_rate_per_night FLOAT NOT NULL,                 --Заполняется  
    администратором  
  
    CONSTRAINT check_rooms_available CHECK  
(number_of_available_rooms >= number_of_occupied_rooms),  
  
    CONSTRAINT hotel_id_fk_for_hotel_rooms FOREIGN KEY (hotel_id)  
    REFERENCES Hotel (hotel_id) ON DELETE CASCADE,  
    CONSTRAINT room_type_id_fk_for_hotel_rooms FOREIGN KEY  
(room_type_id) REFERENCES Type_of_room (type_id) ON DELETE  
    CASCADE  
);
```

--Удаляем столбец hotel\_rooms\_id потому что он избыточен в данной ситуации.

--hotel\_id и room\_type\_id здесь являются составным первичным ключём.

```
ALTER TABLE Hotel_rooms DROP COLUMN hotel_rooms_id;
```

-- Создание таблицы ОТЕЛЬНАЯ КАРТА для авто инкремента использую IDENTITY столбец

```
CREATE TABLE Hotel_card (  
    hotel_card_id NUMBER GENERATED ALWAYS AS IDENTITY  
    PRIMARY KEY,  
    booked_hotel_id NUMBER NOT NULL,  
    reserved_room_id NUMBER NOT NULL,  
    check_in_date DATE NOT NULL,  
    eviction_date DATE NOT NULL,  
    main_living_cost FLOAT DEFAULT 0 NOT NULL, --Вычисляем  
    триггером
```

main\_living\_expenses FLOAT DEFAULT 0 NOT NULL, --Вычисляем триггером

CONSTRAINT fk\_booked\_hotel FOREIGN KEY (booked\_hotel\_id)  
REFERENCES Hotel (hotel\_id) ON DELETE CASCADE,  
CONSTRAINT fk\_reserved\_room FOREIGN KEY (reserved\_room\_id)  
REFERENCES Type\_of\_room (type\_id) ON DELETE CASCADE,  
CONSTRAINT check\_in\_date\_format CHECK (check\_in\_date =  
TO\_DATE(TO\_CHAR(check\_in\_date, 'YYYY-MM-DD'), 'YYYY-MM-DD'),  
CONSTRAINT eviction\_date\_format CHECK (eviction\_date =  
TO\_DATE(TO\_CHAR(eviction\_date, 'YYYY-MM-DD'), 'YYYY-MM-DD'),

CONSTRAINT check\_in\_eviction\_dates CHECK (eviction\_date >=  
check\_in\_date + INTERVAL '2' DAY)  
);

-- САМАЯ СЛОЖНАЯ И ОТВЕТСТВЕННАЯ ТАБЛИЦА, КОТОРАЯ  
СВЯЗЫВАЕТ ВСЁ ОСТАЛЬНОЕ

-- Создание таблицы КАРТА ТУРА для авто инкремента использую  
IDENTITY столбец

CREATE TABLE Tour\_card (  
tour\_card\_id NUMBER GENERATED ALWAYS AS IDENTITY  
PRIMARY KEY,  
tourist\_id NUMBER NOT NULL,  
tour\_id NUMBER NOT NULL,  
flights\_card\_id NUMBER NOT NULL,  
hotel\_card\_id NUMBER NOT NULL,

main\_excursions\_cost FLOAT DEFAULT 0, --Вычисляем триггером  
main\_excursions\_expenses FLOAT DEFAULT 0 NOT NULL, --  
Вычисляем триггером  
cargo\_traffic\_cost FLOAT DEFAULT 0, --Вычисляем триггером  
cargo\_traffic\_expenses FLOAT DEFAULT 0 NOT NULL, --Вычисляем  
триггером  
main\_tour\_cost FLOAT DEFAULT 0 NOT NULL, --Вычисляем  
триггером  
main\_tour\_expenses FLOAT DEFAULT 0 NOT NULL, --Вычисляем  
триггером

```

        CONSTRAINT fk_tourist_for_tour_card FOREIGN KEY (tourist_id)
REFERENCES Tourist (tourist_id) ON DELETE CASCADE,
        CONSTRAINT fk_tour_for_tour_card FOREIGN KEY (tour_id)
REFERENCES Tour (tour_id) ON DELETE CASCADE,
        CONSTRAINT fk_flight_card_for_tour_card FOREIGN KEY
(flights_card_id) REFERENCES Flight_card (flights_card_id) ON DELETE
CASCADE,
        CONSTRAINT fk_hotel_card_for_tour_card FOREIGN KEY
(hotel_card_id) REFERENCES Hotel_card (hotel_card_id) ON DELETE
CASCADE
);

```

-- Создание таблицы ГРУПП ТУРИСТОВ для авто инкремента  
используя IDENTITY столбец

```

CREATE TABLE Tourists_group (
    tourist_id NUMBER NOT NULL,
    tour_id NUMBER NOT NULL,
    group_name VARCHAR2(700) NOT NULL,

    CONSTRAINT fk_tourist_for_tourist_group FOREIGN KEY (tourist_id)
REFERENCES Tourist (tourist_id) ON DELETE CASCADE,
    CONSTRAINT fk_tour_for_tourist_group FOREIGN KEY (tour_id)
REFERENCES Tour (tour_id) ON DELETE CASCADE
);

```

```

-- Создание триггера для автоматического заполнения таблицы
"Tourists_group" при добавлении новой записи в таблицу "Tour_card"
-- DROP TRIGGER tourists_group_trigger;
CREATE OR REPLACE TRIGGER tourists_group_trigger
AFTER INSERT ON Tour_card
FOR EACH ROW
BEGIN
    INSERT INTO Tourists_group (tourist_id, tour_id)
    VALUES (:new.tourist_id, :new.tour_id);
END;
/

```

```

-- Создание триггера для генерации наименования группы туристов по
названию тура и дате начала
--DROP TRIGGER group_name_trigger;
CREATE OR REPLACE TRIGGER group_name_trigger
BEFORE INSERT ON Tourists_group

```

```

FOR EACH ROW
DECLARE
    tour_name VARCHAR2(300);
    tour_start_date DATE;
BEGIN
    SELECT t.tour_name, t.tour_start_date
    INTO tour_name, tour_start_date
    FROM Tour t
    WHERE t.tour_id = :new.tour_id;

    :new.group_name := tour_name || ' - ' || TO_CHAR(tour_start_date,
'YYYY-MM-DD');
END;
/

```

-- Создание таблицы экскурсий для авто инкремента используя  
IDENTITY столбец, используются более простые ограничения формата  
чем выше

```

CREATE TABLE Excursion (
    excursion_id NUMBER GENERATED ALWAYS AS IDENTITY
PRIMARY KEY,
    excursion_agency_name VARCHAR2(500) NOT NULL,
    excursion_name VARCHAR2(500) NOT NULL,
    excursion_description VARCHAR2(1000),
    excursion_country VARCHAR2(200) NOT NULL,
    excursion_city VARCHAR2(200) NOT NULL,
    excursion_start TIMESTAMP NOT NULL,
    excursion_end TIMESTAMP NOT NULL,
    excursion_cost FLOAT NOT NULL, -- Заполняются экскурсионными
агенствами
    expenses FLOAT NOT NULL, -- Заполняются экскурсионными
агенствами

    CONSTRAINT excursion_start_end_check CHECK (excursion_start <=
excursion_end - INTERVAL '30' MINUTE)
);

```



-- Создание таблицы Связки экскурсий с карточкой Тура человека

```
CREATE TABLE Excursion_card (  
    excursion_id NUMBER NOT NULL,  
    tour_card_id NUMBER NOT NULL,  
  
    CONSTRAINT fk_excursion_id FOREIGN KEY (excursion_id)  
REFERENCES Excursion(excursion_id) ON DELETE CASCADE,  
    CONSTRAINT fk_tour_card_id_for_excursion_card FOREIGN KEY  
(tour_card_id) REFERENCES Tour_card(tour_card_id) ON DELETE  
CASCADE  
);
```

-- Создание таблицы Грузов для авто инкремента использую IDENTITY столбец, здесь все вычисляемые данные реализованы посредством (Computed column) вычисляемых столбцов.

```
CREATE TABLE Cargo (  
    cargo_id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY  
KEY,  
    cargo_name VARCHAR2(400) NOT NULL,  
    cargo_description VARCHAR2(600),  
    receipt_date TIMESTAMP NOT NULL,  
    departure_date TIMESTAMP NOT NULL,  
    cargo_weight FLOAT NOT NULL,  
    cargo_width FLOAT NOT NULL,  
    cargo_height FLOAT NOT NULL,  
    cargo_depth FLOAT NOT NULL,  
  
    occupied_space NUMBER GENERATED ALWAYS AS  
(CEIL(cargo_width * cargo_height * cargo_depth * 0.01)) VIRTUAL,  
    cargo_package_cost FLOAT GENERATED ALWAYS AS (cargo_width *  
cargo_height * cargo_depth * 0.002) VIRTUAL,  
    cargo_insurance_cost FLOAT GENERATED ALWAYS AS (cargo_width *  
cargo_height * cargo_depth * cargo_width * 0.002) VIRTUAL,  
    cargo_transportation_cost FLOAT GENERATED ALWAYS AS  
(cargo_width * cargo_height * cargo_depth * cargo_width * 0.01)  
VIRTUAL,  
    expenses FLOAT GENERATED ALWAYS AS ((cargo_width *  
cargo_height * cargo_depth * 0.002) + (cargo_width * cargo_height *  
cargo_depth * cargo_width * 0.002) + (cargo_width * cargo_height *  
cargo_depth * cargo_width * 0.01)) VIRTUAL,
```

```

total_cost FLOAT GENERATED ALWAYS AS (((cargo_width *
cargo_height * cargo_depth * 0.002) + (cargo_width * cargo_height *
cargo_depth * cargo_width * 0.002) + (cargo_width * cargo_height *
cargo_depth * cargo_width * 0.01)) * 1.1) VIRTUAL,
cargo_type CARGOTYPE NOT NULL,

```

```

CONSTRAINT departure_date_check CHECK (departure_date >=
receipt_date + INTERVAL '1' HOUR)
);

```

```

-- Создание пользовательского типа данных и определение его тела
CREATE OR REPLACE TYPE CargoType AS OBJECT (
    type_value VARCHAR2(100),
    MEMBER FUNCTION is_valid RETURN BOOLEAN,
    MEMBER PROCEDURE validate_type,
    MEMBER FUNCTION to_string RETURN VARCHAR2
) NOT FINAL;

```

```

CREATE OR REPLACE TYPE BODY CargoType AS
    MEMBER FUNCTION is_valid RETURN BOOLEAN IS
    BEGIN
        RETURN (type_value IN ('стандартный', 'негабаритный', 'жидкий',
'сыпучий'));
    END;

```

```

    MEMBER PROCEDURE validate_type IS
    BEGIN
        IF NOT self.is_valid THEN
            RAISE_APPLICATION_ERROR(-20000, 'Недопустимое значение
для типа CargoType');
        END IF;
    END;

```

```

    MEMBER FUNCTION to_string RETURN VARCHAR2 IS
    BEGIN
        RETURN type_value;
    END;
END;

```

-- Создание таблицы Связки грузов с карточкой Тура человека

```
CREATE TABLE Cargo_card (  
    cargo_id NUMBER NOT NULL,  
    tour_card_id NUMBER NOT NULL,  
    customs_documents CHAR(6) NOT NULL CHECK (customs_documents  
IN ('ДА', 'НЕТ')),  
  
    CONSTRAINT fk_cargo_id FOREIGN KEY (cargo_id) REFERENCES  
Cargo (cargo_id) ON DELETE CASCADE,  
    CONSTRAINT fk_tour_card_id FOREIGN KEY (tour_card_id)  
REFERENCES Tour_card (tour_card_id) ON DELETE CASCADE  
);
```

-- СОЗДАНИЕ ВСЕХ БОЛЬШИХ ТРИГГЕРОВ, КОТОРЫМ  
ТРЕБУЕТСЯ МНОГИЕ ТАБЛИЦЫ (ПАРАЧКУ МАЛЕНЬКИХ  
ТРИГГЕРОВ СОЗДАВАЛ И СРЕДИ ТАБЛИЦ)

-- Создание триггера для подсчёта стоимости за перелёт туда обратно

```
CREATE OR REPLACE TRIGGER trg_main_flight_cost
BEFORE INSERT OR UPDATE ON Tour_card
FOR EACH ROW
DECLARE
    v_flight_ticket_cost FLOAT;
    v_num_of_children NUMBER;
    v_flight_there_id NUMBER;
    v_flight_back_id NUMBER;
    new_main_flight_cost FLOAT;
BEGIN
    -- Проверка наличия детей
    SELECT COUNT(*) INTO v_num_of_children
    FROM Tourist_Children tc
    WHERE tc.tourist_id = :NEW.tourist_id;

    -- Получение flight_there_id и flight_back_id из таблицы Flight_card
    SELECT flight_there_id, flight_back_id INTO v_flight_there_id,
    v_flight_back_id
    FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id;

    IF v_num_of_children = 0 THEN
        SELECT flight_ticket_cost INTO v_flight_ticket_cost FROM Flight
        WHERE flight_id = v_flight_there_id;
        new_main_flight_cost := v_flight_ticket_cost;
        SELECT flight_ticket_cost INTO v_flight_ticket_cost FROM Flight
        WHERE flight_id = v_flight_back_id;
        new_main_flight_cost := new_main_flight_cost + v_flight_ticket_cost;
    ELSE
        SELECT flight_ticket_cost INTO v_flight_ticket_cost FROM Flight
        WHERE flight_id = v_flight_there_id;
        new_main_flight_cost := v_flight_ticket_cost + ((v_flight_ticket_cost / 2)
        * v_num_of_children);
        SELECT flight_ticket_cost INTO v_flight_ticket_cost FROM Flight
        WHERE flight_id = v_flight_back_id;
        new_main_flight_cost := new_main_flight_cost + v_flight_ticket_cost +
        ((v_flight_ticket_cost / 2) * v_num_of_children);
    END IF;
    new_main_flight_cost := new_main_flight_cost;
END;
```

```

END IF;

-- Обновление записи в таблице Flight_card
UPDATE Flight_card
SET main_flight_cost = new_main_flight_cost
WHERE flights_card_id = :NEW.flights_card_id;
END;
/

-- Создание триггера для подсчёта затрат на перелёт туда обратно

CREATE OR REPLACE TRIGGER trg_main_maintenance_expenses
BEFORE INSERT OR UPDATE ON Tour_card
FOR EACH ROW
FOLLOWS trg_main_flight_cost
DECLARE
    v_maintenance_expenses FLOAT;
    v_num_of_children NUMBER;
    v_flight_there_id NUMBER;
    v_flight_back_id NUMBER;
    new_main_maintenance_expenses FLOAT;
BEGIN
    -- Проверка наличия детей
    SELECT COUNT(*) INTO v_num_of_children
    FROM Tourist_Children tc
    WHERE tc.tourist_id = :NEW.tourist_id;

    -- Получение flight_there_id и flight_back_id из таблицы Flight_card
    SELECT flight_there_id, flight_back_id INTO v_flight_there_id,
    v_flight_back_id
    FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id;

    IF v_num_of_children = 0 THEN
        SELECT maintenance_expenses INTO v_maintenance_expenses FROM
        Flight WHERE flight_id = v_flight_there_id;
        new_main_maintenance_expenses := v_maintenance_expenses;
        SELECT maintenance_expenses INTO v_maintenance_expenses FROM
        Flight WHERE flight_id = v_flight_back_id;
        new_main_maintenance_expenses := new_main_maintenance_expenses +
        v_maintenance_expenses;
    ELSE

```

```

        SELECT maintenance_expenses INTO v_maintenance_expenses FROM
        Flight WHERE flight_id = v_flight_there_id;
        new_main_maintenance_expenses := v_maintenance_expenses +
        ((v_maintenance_expenses / 2) * v_num_of_children);
        SELECT maintenance_expenses INTO v_maintenance_expenses FROM
        Flight WHERE flight_id = v_flight_back_id;
        new_main_maintenance_expenses := new_main_maintenance_expenses +
        v_maintenance_expenses + ((v_maintenance_expenses / 2) *
        v_num_of_children);
        END IF;

        -- Обновление записи в таблице Flight_card
        UPDATE Flight_card
        SET main_maintenance_expenses = new_main_maintenance_expenses
        WHERE flights_card_id = :NEW.flights_card_id;
    END;
/

```

-- Создание триггера для обновления данных в таблице Flight при добавлении записи в таблицу Tour\_card

```

CREATE OR REPLACE TRIGGER trg_update_flight_data
AFTER INSERT OR UPDATE ON Tour_card
FOR EACH ROW
DECLARE
    v_tourist_children_count NUMBER;
    v_cargo_weight FLOAT;
    v_cargo_space NUMBER;
BEGIN
    --ЗАНЯТЫЕ ПАССАЖИРСКИЕ МЕСТА
    -- Проверка наличия детей и подсчёт их кол-ва
    SELECT COUNT(*) INTO v_tourist_children_count
    FROM Tourist_Children tc
    WHERE tc.tourist_id = :NEW.tourist_id;

    -- Обновление поля occupied_places для ПЕРВОГО рейса
    UPDATE Flight SET occupied_places = occupied_places + 1 +
    v_tourist_children_count
    WHERE flight_id = (
        SELECT flight_there_id FROM Flight_card
        WHERE flights_card_id = :NEW.flights_card_id
    );

```

```

-- Обновление поля occupied_places для ВТОРОГО рейса
UPDATE Flight SET occupied_places = occupied_places + 1 +
v_tourist_children_count
WHERE flight_id = (
    SELECT flight_back_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
);

--БЕС ГРУЗА
-- Получение информации о грузе для ПЕРВОГО рейса
SELECT SUM(cargo_weight) INTO v_cargo_weight
FROM Cargo
WHERE cargo_id IN (
    SELECT cargo_id
    FROM Cargo_card
    WHERE tour_card_id = :NEW.tour_card_id
) AND receipt_date <= (SELECT departure_datetime FROM Flight
WHERE flight_id = (
    SELECT flight_there_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
)) - INTERVAL '2' DAY;

-- Обновление поля occupied_baggage_weight для ПЕРВОГО рейса
UPDATE Flight SET occupied_baggage_weight =
occupied_baggage_weight + COALESCE(v_cargo_weight, 0)
WHERE flight_id = (
    SELECT flight_there_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
);

-- Получение информации о грузе для ВТОРОГО рейса
SELECT SUM(cargo_weight) INTO v_cargo_weight
FROM Cargo
WHERE cargo_id IN (
    SELECT cargo_id
    FROM Cargo_card
    WHERE tour_card_id = :NEW.tour_card_id
) AND receipt_date <= (SELECT departure_datetime FROM Flight
WHERE flight_id = (
    SELECT flight_back_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
)) - INTERVAL '2' DAY;

```

```

-- Обновление поля ocupated_baggage_weight для ВТОРОГО рейса
UPDATE Flight SET ocupated_baggage_weight =
ocupated_baggage_weight + COALESCE(v_cargo_weight, 0)
WHERE flight_id = (
    SELECT flight_back_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
);
--ЗАНИМАЕМОЕ МЕСТО БАГАЖА
-- Получение информации о занимаемом месте багажа для ПЕРВОГО
рейса
SELECT SUM(occupied_space) INTO v_cargo_space
FROM Cargo
WHERE cargo_id IN (
    SELECT cargo_id
    FROM Cargo_card
    WHERE tour_card_id = :NEW.tour_card_id
) AND receipt_date <= (SELECT departure_datetime FROM Flight
WHERE flight_id = (
    SELECT flight_there_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
)) - INTERVAL '2' DAY;
-- Обновление поля ocupated_baggage_space для ПЕРВОГО рейса
UPDATE Flight SET ocupated_baggage_space = ocupated_baggage_space
+ COALESCE(v_cargo_space, 0)
WHERE flight_id = (
    SELECT flight_there_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
);
-- Получение информации о занимаемом месте багажа для ВТОРОГО
рейса
SELECT SUM(occupied_space) INTO v_cargo_space
FROM Cargo
WHERE cargo_id IN (
    SELECT cargo_id
    FROM Cargo_card
    WHERE tour_card_id = :NEW.tour_card_id
) AND receipt_date <= (SELECT departure_datetime FROM Flight
WHERE flight_id = (
    SELECT flight_back_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
)) - INTERVAL '2' DAY;

-- Обновление поля ocupated_baggage_space для ВТОРОГО рейса

```



```

UPDATE Flight SET occupied_baggage_space = occupied_baggage_space
+ COALESCE(v_cargo_space, 0)
WHERE flight_id = (
    SELECT flight_back_id FROM Flight_card
    WHERE flights_card_id = :NEW.flights_card_id
);
END;
/

```

-- Создание триггера для таблицы Hotel\_card

```

CREATE OR REPLACE TRIGGER calculate_expenses_cost
BEFORE INSERT OR UPDATE ON Hotel_card
FOR EACH ROW
DECLARE
    total_children NUMBER;
    room_rate_per_night FLOAT;
BEGIN
    SELECT COUNT(*) INTO total_children
    FROM Tourist_Children
    WHERE tourist_id = (
        SELECT tourist_id FROM Tour_card
        WHERE hotel_card_id = :NEW.hotel_card_id
    );

    SELECT room_rate_per_night
    INTO room_rate_per_night
    FROM Hotel_rooms
    WHERE hotel_id = :NEW.booked_hotel_id
    AND room_type_id = :NEW.reserved_room_id;

    :NEW.main_living_cost := (room_rate_per_night *
    CEIL(:NEW.eviction_date - :NEW.check_in_date) + (room_rate_per_night
    * CEIL(:NEW.eviction_date - :NEW.check_in_date) * 0.5 * total_children))
    * 1.1;

    :NEW.main_living_expenses := room_rate_per_night *
    CEIL(:NEW.eviction_date - :NEW.check_in_date) + (room_rate_per_night
    * CEIL(:NEW.eviction_date - :NEW.check_in_date) * 0.5 * total_children);
END;
/

```

-- Создание триггера для таблицы Hotel при изменении таблицы Hotel\_card

```
CREATE OR REPLACE TRIGGER update_hotel_occupied_rooms
AFTER INSERT ON Hotel_card
FOR EACH ROW
BEGIN
    UPDATE Hotel_rooms
    SET number_of_occupied_rooms = number_of_occupied_rooms + 1
    WHERE hotel_id = :NEW.booked_hotel_id
    AND room_type_id = :NEW.reserved_room_id;
END;
/
```

-- Триггер для вычисления полей main\_excursions\_cost и main\_excursions\_expenses для таблицы Tour\_card

```
CREATE OR REPLACE TRIGGER calculate_main_excursions
AFTER INSERT OR UPDATE ON Excursion_card
FOR EACH ROW
--FOLLOWS trg_main_maintenance_expenses
DECLARE
    v_excursions_cost FLOAT;
    v_excursions_expenses FLOAT;
    old_main_excursions_cost FLOAT;
    old_main_excursions_expenses FLOAT;
BEGIN
    SELECT excursion_cost, expenses
    INTO v_excursions_cost, v_excursions_expenses
    FROM Excursion ex
    WHERE ex.excursion_id = :NEW.excursion_id;

    SELECT main_excursions_cost, main_excursions_expenses
    INTO old_main_excursions_cost, old_main_excursions_expenses
    FROM Tour_card tc
    WHERE tc.tour_card_id = :NEW.tour_card_id;

    IF v_excursions_cost IS NULL THEN
        v_excursions_cost := 0;
    END IF;

    IF v_excursions_expenses IS NULL THEN
        v_excursions_expenses := 0;
    END IF;
```

```

IF old_main_excursions_cost IS NULL THEN
    old_main_excursions_cost := 0;
END IF;

```

```

IF old_main_excursions_expenses IS NULL THEN
    old_main_excursions_expenses := 0;
END IF;

```

```

UPDATE Tour_card
SET main_excursions_cost = old_main_excursions_cost +
v_excursions_cost,
    main_excursions_expenses = old_main_excursions_expenses +
v_excursions_expenses
WHERE tour_card_id = :NEW.tour_card_id;
END;
/

```

-- Триггер для вычисления полей cargo\_traffic\_cost и cargo\_traffic\_expenses  
для таблицы Tour\_card //COMPILED

```

CREATE OR REPLACE TRIGGER calculate_cargo_traffic
BEFORE INSERT OR UPDATE ON Cargo_card
FOR EACH ROW
--FOLLOWS calculate_main_excursions
DECLARE
    v_cargo_cost FLOAT;
    v_cargo_expenses FLOAT;
    old_cargo_traffic_cost FLOAT;
    old_cargo_traffic_expenses FLOAT;
BEGIN
    SELECT total_cost, expenses
    INTO v_cargo_cost, v_cargo_expenses
    FROM Cargo c
    WHERE c.cargo_id = :NEW.cargo_id;

    SELECT cargo_traffic_cost, cargo_traffic_expenses
    INTO old_cargo_traffic_cost, old_cargo_traffic_expenses
    FROM Tour_card tc
    WHERE tc.tour_card_id = :NEW.tour_card_id;

    IF v_cargo_cost IS NULL THEN
        v_cargo_cost := 0;

```

```
END IF;
```

```
IF v_cargo_expenses IS NULL THEN
```

```
    v_cargo_expenses := 0;
```

```
END IF;
```

```
IF old_cargo_traffic_cost IS NULL THEN
```

```
    old_cargo_traffic_cost := 0;
```

```
END IF;
```

```
IF old_cargo_traffic_expenses IS NULL THEN
```

```
    old_cargo_traffic_expenses := 0;
```

```
END IF;
```

```
UPDATE Tour_card
```

```
SET cargo_traffic_cost = old_cargo_traffic_cost + v_cargo_cost,
```

```
    cargo_traffic_expenses = old_cargo_traffic_expenses +
```

```
v_cargo_expenses
```

```
WHERE tour_card_id = :NEW.tour_card_id;
```

```
END;
```

```
/
```

-- Создания триггера подсчёта итоговых трат и итоговой стоимости тура для  
таблицы Tour\_card

```
CREATE OR REPLACE TRIGGER calculate_main_tour_cost_expenses
```

```
BEFORE INSERT OR UPDATE ON Tour_card
```

```
FOR EACH ROW
```

```
FOLLOWS trg_main_maintenance_expenses
```

```
DECLARE
```

```
    v_tour_cost FLOAT;
```

```
    v_main_living_cost FLOAT;
```

```
    v_main_living_expenses FLOAT;
```

```
    v_main_flight_cost FLOAT;
```

```
    v_main_maintenance_expenses FLOAT;
```

```
BEGIN
```

```
-- Получение стоимости тура из таблицы Tour
```

```
SELECT tour_cost INTO v_tour_cost
```

```
FROM Tour
```

```
WHERE tour_id = :NEW.tour_id;
```

```
-- Получение стоимости проживания из таблицы Hotel_card
```

```

SELECT main_living_cost, main_living_expenses INTO
v_main_living_cost, v_main_living_expenses
FROM Hotel_card
WHERE hotel_card_id = :NEW.hotel_card_id;

-- Получение стоимости перелета из таблицы Flight_card
SELECT main_flight_cost, main_maintenance_expenses INTO
v_main_flight_cost, v_main_maintenance_expenses
FROM Flight_card
WHERE flights_card_id = :NEW.flights_card_id;

-- Вычисление main_tour_cost и main_tour_expenses
:NEW.main_tour_cost := v_tour_cost + v_main_living_cost +
v_main_flight_cost + :NEW.main_excursions_cost +
:NEW.cargo_traffic_cost;
:NEW.main_tour_expenses := v_main_living_expenses +
v_main_maintenance_expenses + :NEW.main_excursions_expenses +
:NEW.cargo_traffic_expenses;
END;
/

```

```

-- СОЗДАНИЕ JOB`s -----
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name      => 'CHECK_HOTEL_CARD_JOB',
    job_type      => 'PLSQL_BLOCK',
    job_action     => 'BEGIN
                        FOR rec IN (SELECT * FROM Hotel_card
                        WHERE eviction_date <= SYSDATE AND eviction_date >
TRUNC(SYSDATE) - INTERVAL 1 DAY)
                        LOOP
                          UPDATE Hotel_rooms
                          SET number_of_occupied_rooms =
number_of_occupied_rooms - 1
                          WHERE booked_hotel_id = rec.booked_hotel_id
                          AND reserved_room_id = rec.reserved_room_id;
                        END LOOP;
                        END;',
    start_date     => SYSDATE,
    repeat_interval => 'FREQ=DAILY; BYHOUR=12; BYMINUTE=10;',
    enabled        => TRUE,
    comments       => 'Job to check Hotel_card table and update Hotel_rooms
table accordingly.'
  );
END;
/

```

```

-- ВСТАВКИ В ДАННЫХ В ТАБЛИЦЫ -----

-- Заполнение таблицы ТУРИСТОВ

-- Запись 1
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,
tourist_last_name, tourist_residence_city, tourist_age, phone_number,
tourist_type)
VALUES ('Иван', 'Сергеевич', 'Петров', 'Москва', 35, '+79991234567',
TouristType('турист-грузоперевозчик'));

-- Запись 2
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,
tourist_last_name, tourist_residence_city, tourist_age, phone_number,
tourist_type)
VALUES ('Елена', 'Александровна', 'Смирнова', 'Санкт-Петербург', 28,
'+79119876543', TouristType('турист-отдыхающий'));

-- Запись 3
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,
tourist_last_name, tourist_residence_city, tourist_age, phone_number,
tourist_type)
VALUES ('Алексей', 'Дмитриевич', 'Иванов', 'Нижний Новгород', 42,
'+7(910)555-12-34', TouristType('турист-грузоперевозчик'));

-- Запись 4
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,
tourist_last_name, tourist_residence_city, tourist_age, phone_number,
tourist_type)
VALUES ('Мария', 'Алексеевна', 'Соколова', 'Екатеринбург', 31,
'+7(922)1112233', TouristType('турист-отдыхающий'));

-- Запись 5
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,
tourist_last_name, tourist_residence_city, tourist_age, phone_number,
tourist_type)
VALUES ('Денис', 'Игоревич', 'Козлов', 'Казань', 39, '89876543210',
TouristType('турист-грузоперевозчик'));

-- Запись 6
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,
tourist_last_name, tourist_residence_city, tourist_age, phone_number,
tourist_type)

```

```
VALUES ('Анна', 'Павловна', 'Кузнецова', 'Ростов-на-Дону', 24,  
'8(987)654-32-10', TouristType('турист-отдыхающий'));
```

```
-- Запись 7
```

```
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,  
tourist_last_name, tourist_residence_city, tourist_age, phone_number,  
tourist_type)  
VALUES ('Сергей', 'Андреевич', 'Волков', 'Уфа', 37, '+79223334455',  
TouristType('турист-грузоперевозчик'));
```

```
-- Запись 8
```

```
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,  
tourist_last_name, tourist_residence_city, tourist_age, phone_number,  
tourist_type)  
VALUES ('Ольга', 'Владимировна', 'Морозова', 'Самара', 29,  
'+79998765432', TouristType('турист-отдыхающий'));
```

```
-- Запись 9
```

```
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,  
tourist_last_name, tourist_residence_city, tourist_age, phone_number,  
tourist_type)  
VALUES ('Артем', 'Иванович', 'Смирнов', 'Волгоград', 33,  
'+79876543210', TouristType('турист-грузоперевозчик'));
```

```
-- Запись 10
```

```
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,  
tourist_last_name, tourist_residence_city, tourist_age, phone_number,  
tourist_type)  
VALUES ('Екатерина', 'Алексеевна', 'Ковалева', 'Пермь', 26,  
'+79221112233', TouristType('турист-отдыхающий'));
```

```
-- Запись 11
```

```
INSERT INTO Tourist (tourist_first_name, tourist_middle_name,  
tourist_last_name, tourist_residence_city, tourist_age, phone_number,  
tourist_type)  
VALUES ('Варвара', 'Тригорьевна', 'Торсть', 'Омск', 54, '+79777634874',  
TouristType('турист-отдыхающий'));
```

```
-- Заполнение таблицы ПАСПОРТНЫХ ДАННЫХ
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)
```



```
VALUES (21, TO_DATE('1985-03-12', 'YYYY-MM-DD'), 1234, 567890,  
'МВД г. Москва', TO_DATE('2010-05-20', 'YYYY-MM-DD'), 'МУЖ');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (22, TO_DATE('1993-10-05', 'YYYY-MM-DD'), 5678, 901234,  
'ОВД г. Санкт-Петербург', TO_DATE('2016-08-15', 'YYYY-MM-DD'),  
'ЖЕН');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (24, TO_DATE('1978-07-18', 'YYYY-MM-DD'), 4321, 987654,  
'ОВД г. Нижний Новгород', TO_DATE('2000-11-30', 'YYYY-MM-DD'),  
'МУЖ');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (25, TO_DATE('1989-02-28', 'YYYY-MM-DD'), 8765, 432109,  
'МВД г. Екатеринбург', TO_DATE('2014-06-08', 'YYYY-MM-DD'),  
'ЖЕН');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (26, TO_DATE('1982-06-15', 'YYYY-MM-DD'), 2109, 876543,  
'ОВД г. Казань', TO_DATE('2008-04-23', 'YYYY-MM-DD'), 'МУЖ');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (27, TO_DATE('1997-11-04', 'YYYY-MM-DD'), 6543, 210987,  
'МВД г. Ростов-на-Дону', TO_DATE('2020-09-10', 'YYYY-MM-DD'),  
'ЖЕН');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (28, TO_DATE('1984-09-22', 'YYYY-MM-DD'), 3456, 789012,  
'ОВД г. Уфа', TO_DATE('2006-03-17', 'YYYY-MM-DD'), 'МУЖ');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (29, TO_DATE('1992-05-07', 'YYYY-MM-DD'), 7890, 123456,  
'МВД г. Самара', TO_DATE('2012-10-25', 'YYYY-MM-DD'), 'ЖЕН');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (30, TO_DATE('1986-12-19', 'YYYY-MM-DD'), 5678, 234567,  
'ОВД г. Волгоград', TO_DATE('2011-07-05', 'YYYY-MM-DD'), 'МУЖ');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (31, TO_DATE('1995-08-08', 'YYYY-MM-DD'), 9876, 345678,  
'МВД г. Пермь', TO_DATE('2018-12-12', 'YYYY-MM-DD'), 'ЖЕН');
```

```
INSERT INTO Passport_data (tourist_id, birthday, pass_series,  
pass_number, issued_by, when_issued, sex)  
VALUES (41, TO_DATE('1969-11-26', 'YYYY-MM-DD'), 0317, 708708,  
'МВД г. Омск', TO_DATE('2015-12-06', 'YYYY-MM-DD'), 'ЖЕН');
```

-- Заполнение таблицы ДЕТИ

-- Запись 1

```
INSERT INTO Children (child_first_name, child_middle_name,  
child_last_name, child_sex, child_age)  
VALUES ('Александр', 'Иванович', 'Смирнов', 'МУЖ', 5);
```

-- Запись 2

```
INSERT INTO Children (child_first_name, child_middle_name,  
child_last_name, child_sex, child_age)  
VALUES ('Анна', 'Ивановна', 'Смирнова', 'ЖЕН', 7);
```

-- Запись 3

```
INSERT INTO Children (child_first_name, child_middle_name,  
child_last_name, child_sex, child_age)  
VALUES ('Максим', 'Александрович', 'Соколов', 'МУЖ', 10);
```

```

-- Запись 4
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Алиса', 'Александровна', 'Соколова', 'ЖЕН', 8);

-- Запись 5
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Дмитрий', 'Сергеевич', 'Кузнецов', 'МУЖ', 6);

-- Запись 6
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Екатерина', 'Сергеевна', 'Кузнецова', 'ЖЕН', 4);

-- Запись 7
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Илья', 'Алексеевич', 'Морозов', 'МУЖ', 8);

-- Запись 8
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Мария', 'Алексеевна', 'Морозова', 'ЖЕН', 7);

-- Запись 9
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Артем', 'Алексеевич', 'Морозов', 'МУЖ', 5);

-- Запись 10
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('София', 'Дмитриевна', 'Ковалева', 'ЖЕН', 4);

-- Запись 11
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Иван', 'Вадимович', 'Горсть', 'МУЖ', 6);

```

```

---28
-- Запись 14
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Марта', 'Сергеевна', 'Волкова', 'ЖЕН', 11);

-- Запись 15
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Август', 'Сергеевич', 'Волков', 'МУЖ', 9);

----30
-- Запись 16
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Эйприл', 'Артёмовна', 'Смирнова', 'ЖЕН', 9);

-- Запись 17
INSERT INTO Children (child_first_name, child_middle_name,
child_last_name, child_sex, child_age)
VALUES ('Махон', 'Артёмович', 'Смирнов', 'МУЖ', 12);

-- Заполнение таблицы ТУРИСТЫ ДЕТИ:

-- Запись 1-2: Турист-отдыхающий с двумя детьми
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (22, 2);

INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (22, 3);

-- Запись 3-4: Турист-отдыхающий с двумя детьми
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (25, 4);

INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (25, 5);

-- Запись 5-6: Турист-отдыхающий с двумя детьми
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (27, 6);

```

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (27, 7);
```

----

```
-- Запись 7-9: Турист-отдыхающий с тремя детьми
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (29, 8);
```

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (29, 9);
```

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (29, 10);
```

```
-- Запись 10: Турист-отдыхающий с одним ребенком
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (31, 11);
```

```
-- Запись 11: Турист-отдыхающий с одним ребенком
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (41, 12);
```

----

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (28, 21);
```

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (28, 22);
```

-----

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (30, 23);
```

```
INSERT INTO Tourist_Children (tourist_id, child_id)
VALUES (30, 24);
```

-- Заполнение таблицы ТУРЫ

-- Вставка записи для тура "Зимний Горнолыжный Рай" (турист-отдыхающий, зимний регион)

```
INSERT INTO Tour (tour_name, tour_description, tour_cost, tour_country,
tour_city, tour_start_date, tour_end_date)
VALUES ('Зимний Горнолыжный Рай', 'Отправляйтесь в волшебный
зимний курорт с горнолыжными склонами и трассами. Насладитесь
катанием на лыжах и сноуборде, окунитесь в сказочную атмосферу
заснеженных гор и прогулок по зимнему лесу. Расслабьтесь в уютных
горнолыжных отелях.', 1500, 'Австрия', 'Зельден', TO_DATE('2024-01-
10', 'YYYY-MM-DD HH24:MI:SS'), TO_DATE('2024-01-17', 'YYYY-MM-
DD HH24:MI:SS'));
```

-- Вставка записи для тура "Тропический Рай" (турист-отдыхающий, жаркий летний регион)

```
INSERT INTO Tour (tour_name, tour_description, tour_cost, tour_country,
tour_city, tour_start_date, tour_end_date)
VALUES ('Тропический Рай', 'Отправляйтесь в путешествие на жаркие
тропические острова! Насладитесь пляжами, морем и пальмовыми
рощами. Попробуйте экзотические фрукты, плавайте с дельфинами и
окунитесь в мир подводного снорклинга.', 2000, 'Мальдивы', 'Мале',
TO_DATE('2024-07-01', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-07-10', 'YYYY-MM-DD HH24:MI:SS'));
```

-- Вставка записи для тура "Шопинг в Модной Столице" (турист-шоппер)

```
INSERT INTO Tour (tour_name, tour_description, tour_cost, tour_country,
tour_city, tour_start_date, tour_end_date)
VALUES ('Шопинг в Модной Столице', 'Отправляйтесь в шопинг-тур в
Милан - модную столицу мира! Откройте для себя бутики известных
дизайнеров и насладитесь итальянской модой.', 1800, 'Италия', 'Милан',
TO_DATE('2024-04-15', 'YYYY-MM-DD HH24:MI:SS'),
TO_DATE('2024-04-22', 'YYYY-MM-DD HH24:MI:SS'));
```

-- Вставка записи для тура "Романтический Париж" (турист-отдыхающий, весенний регион)

```
INSERT INTO Tour (tour_name, tour_description, tour_cost, tour_country,
tour_city, tour_start_date, tour_end_date)
VALUES ('Романтический Париж', 'Погрузитесь в атмосферу любви и
романтики в самом романтическом городе мира - Париже! Гуляйте по
узким улочкам Монмартра, наслаждайтесь видом на Эйфелеву башню с
берегов Сены, посетите известные музеи и сады.', 1700, 'Франция',
```

```
'Париж', TO_DATE('2024-05-10', 'YYYY-MM-DD HH24:MI:SS'),  
TO_DATE('2024-05-17', 'YYYY-MM-DD HH24:MI:SS'));
```

-- Заполнение таблицы РЕЙСОВ

-- Запись 1

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline1', 'Россия', 'Ростов-на-Дону', 'Австрия', 'Зельден',  
TIMESTAMP '2024-01-10 08:00:00', TIMESTAMP '2024-01-10 10:30:00',  
300, 700, 500, 300.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 2

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline2', 'Россия', 'Уфа', 'Австрия', 'Зельден', TIMESTAMP  
'2024-01-10 08:00:00', TIMESTAMP '2024-01-10 10:30:00', 150, 600, 300,  
600.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 3

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline1', 'Россия', 'Самара', 'Австрия', 'Зельден', TIMESTAMP  
'2024-01-10 06:00:00', TIMESTAMP '2024-01-10 10:30:00', 300, 500, 300,  
400.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 4

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)
```

```
VALUES ('Airline1', 'Россия', 'Волгоград', 'Мальдивы', 'Мале',  
TIMESTAMP '2024-07-01 05:30:00', TIMESTAMP '2024-07-01 10:30:00',  
150, 700, 300, 300.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 5

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline1', 'Россия', 'Пермь', 'Мальдивы', 'Мале', TIMESTAMP  
'2024-07-01 04:30:00', TIMESTAMP '2024-07-01 10:30:00', 300, 800, 400,  
400.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 6

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline2', 'Россия', 'Омск', 'Мальдивы', 'Мале', TIMESTAMP  
'2024-07-01 03:00:00', TIMESTAMP '2024-07-01 10:30:00', 200, 1000,  
1000, 500.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 7

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline1', 'Россия', 'Москва', 'Италия', 'Милан', TIMESTAMP  
'2024-04-15 06:30:00', TIMESTAMP '2024-04-15 10:30:00', 150, 1000,  
1000, 800.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 8

```
INSERT INTO Flight (airline_name, where_from_country,  
where_from_city, where_country, where_city, departure_datetime,  
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,  
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,  
plane_type)  
VALUES ('Airline2', 'Россия', 'Санкт-Петербург', 'Италия', 'Милан',  
TIMESTAMP '2024-04-15 07:30:00', TIMESTAMP '2024-04-15 10:30:00',  
300, 500, 400, 600.0, 100.0, PlaneType('грузо-пассажирский'));
```



-- Запись 9

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Россия', 'Нижний Новгород', 'Италия', 'Милан',
TIMESTAMP '2024-04-15 09:00:00', TIMESTAMP '2024-04-15 10:30:00',
100, 1000, 500, 400.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 10

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline2', 'Россия', 'Екатеринбург', 'Италия', 'Милан',
TIMESTAMP '2024-04-15 07:00:00', TIMESTAMP '2024-04-15 10:30:00',
300, 900, 700, 500.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Запись 11

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Россия', 'Казань', 'Италия', 'Милан', TIMESTAMP
'2024-04-15 07:30:00', TIMESTAMP '2024-04-15 10:30:00', 150, 300, 500,
500.0, 100.0, PlaneType('грузо-пассажирский'));
```

---ГРУЗОВЫЕ РЕЙСЫ

-- Запись 12

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Россия', 'Санкт-Петербург', 'Китай', 'Пекин',
TIMESTAMP '2024-04-15 04:00:00', TIMESTAMP '2024-04-15 10:30:00',
0, 1000, 1500, 100.0, 30.0, PlaneType('грузовой'));
```

-- Запись 13

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline2', 'Россия', 'Екатеринбург', 'Ирландия', 'Дублин',
TIMESTAMP '2024-04-15 07:00:00', TIMESTAMP '2024-04-15 10:30:00',
0, 1900, 1700, 50.0, 30.0, PlaneType('грузовой'));
```

-- Запись 14

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Россия', 'Казань', 'Белоруссия', 'Минск',
TIMESTAMP '2024-04-15 07:30:00', TIMESTAMP '2024-04-15 10:00:00',
0, 1300, 1500, 200.0, 30.0, PlaneType('грузовой'));
```

-- Запись 15

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Россия', 'Москва', 'Казахстан', 'Астана',
TIMESTAMP '2024-04-15 07:30:00', TIMESTAMP '2024-04-15 9:30:00', 0,
1300, 1500, 50.0, 30.0, PlaneType('грузовой'));
```

--ОБРАТНЫЕ РЕЙСЫ

-- Запись 16 (обратный рейс для рейса из Ростова-на-Дону в Зельден)

```
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Австрия', 'Зельден', 'Россия', 'Ростов-на-Дону',
TIMESTAMP '2024-01-17 14:00:00', TIMESTAMP '2024-01-17 16:30:00',
300, 700, 500, 300.0, 100.0, PlaneType('грузо-пассажирский'));
```

```

-- Обратный рейс для рейса из Уфы в Зельден
-- Запись 17 (обратный рейс для рейса из Уфы в Зельден)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline2', 'Австрия', 'Зельден', 'Россия', 'Уфа', TIMESTAMP
'2024-01-17 14:00:00', TIMESTAMP '2024-01-17 16:30:00', 150, 600, 300,
600.0, 100.0, PlaneType('грузо-пассажирский'));

-- Обратный рейс для рейса из Самары в Зельден
-- Запись 18 (обратный рейс для рейса из Самары в Зельден)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Австрия', 'Зельден', 'Россия', 'Самара', TIMESTAMP
'2024-01-17 14:00:00', TIMESTAMP '2024-01-17 16:30:00', 300, 500, 300,
400.0, 100.0, PlaneType('грузо-пассажирский'));

-- Обратные рейсы для тура "Тропический Рай"

-- Запись 19 (обратный рейс для рейса из Волгограда в Мале)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Мальдивы', 'Мале', 'Россия', 'Волгоград',
TIMESTAMP '2024-07-10 14:00:00', TIMESTAMP '2024-07-10 18:00:00',
150, 700, 300, 300.0, 100.0, PlaneType('грузо-пассажирский'));

-- Запись 20 (обратный рейс для рейса из Перми в Мале)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Мальдивы', 'Мале', 'Россия', 'Пермь', TIMESTAMP
'2024-07-10 14:00:00', TIMESTAMP '2024-07-10 18:00:00', 300, 800, 400,
400.0, 100.0, PlaneType('грузо-пассажирский'));

```

```
-- Запись 21 (обратный рейс для рейса из Омска в Мале)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline2', 'Мальдивы', 'Мале', 'Россия', 'Омск', TIMESTAMP
'2024-07-10 14:00:00', TIMESTAMP '2024-07-10 18:00:00', 200, 1000,
1000, 500.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Обратные рейсы для тура "Шопинг в Модной Столице"

```
-- Запись 22 (обратный рейс для рейса из Москвы в Милан)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Италия', 'Милан', 'Россия', 'Москва', TIMESTAMP
'2024-04-22 16:00:00', TIMESTAMP '2024-04-22 20:00:00', 150, 1000,
1000, 800.0, 100.0, PlaneType('грузо-пассажирский'));
```

```
-- Запись 23 (обратный рейс для рейса из Санкт-Петербурга в Милан)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline2', 'Италия', 'Милан', 'Россия', 'Санкт-Петербург',
TIMESTAMP '2024-04-22 16:00:00', TIMESTAMP '2024-04-22 20:00:00',
300, 500, 400, 600.0, 100.0, PlaneType('грузо-пассажирский'));
```

```
-- Запись 24 (обратный рейс для рейса из Нижнего Новгорода в Милан)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Италия', 'Милан', 'Россия', 'Нижний Новгород',
TIMESTAMP '2024-04-22 16:00:00', TIMESTAMP '2024-04-22 20:00:00',
100, 1000, 500, 400.0, 100.0, PlaneType('грузо-пассажирский'));
```

```
-- Запись 25 (обратный рейс для рейса из Екатеринбурга в Милан)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline2', 'Италия', 'Милан', 'Россия', 'Екатеринбург',
TIMESTAMP '2024-04-22 16:00:00', TIMESTAMP '2024-04-22 20:00:00',
300, 900, 700, 500.0, 100.0, PlaneType('грузо-пассажирский'));
```

```
-- Запись 26 (обратный рейс для рейса из Казани в Милан)
INSERT INTO Flight (airline_name, where_from_country,
where_from_city, where_country, where_city, departure_datetime,
datetime_of_arrival, capacity_of_people, capacity_weight_of_cargo,
capacity_space_of_cargo, flight_ticket_cost, maintenance_expenses,
plane_type)
VALUES ('Airline1', 'Италия', 'Милан', 'Россия', 'Казань', TIMESTAMP
'2024-04-22 16:00:00', TIMESTAMP '2024-04-22 20:00:00', 150, 300, 500,
500.0, 100.0, PlaneType('грузо-пассажирский'));
```

-- Заполнение таблицы КАРТЫ РЕЙСОВ

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (1, 16);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (2, 17);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (3, 18);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (4, 19);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (5, 20);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (6, 21);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (7, 22);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (8, 23);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (9, 24);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (10, 25);
```

```
INSERT INTO Flight_card (flight_there_id, flight_back_id)
VALUES (11, 26);
```

-- Заполнение таблицы ОТЕЛЕЙ

-- Вставка записи для отеля в Зельдене, Австрия (для тура "Зимний Горнолыжный Рай")

```
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('Hotel Alpenhof', 'Австрия', 'Зельден', 'st. Dorfstrasse prift h.
55/1 ap. 2');
```

-- Вставка записи для отеля в Мале, Мальдивы (для тура "Тропический Рай")

```
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('Adaaran Select Hudhuranfushi', 'Мальдивы', 'Мале', 'st.
Hudhuranfushi Island h. 136/1 ap. 23');
```

-- Вставка записи для отеля в Милане, Италия (для тура "Шопинг в Модной Столице")

```
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('Hotel Milano Scala', 'Италия', 'Милан', 'st. Via Alessandro h. 7/3
ap. 201');
```

-----

-- Вставка записи для отеля в Париже, Франция (для тура "Романтический Париж")

```
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('Hotel Raphael', 'Франция', 'Париж', 'st. Avenue Kleber h. 17/1
ap. 45');
```

-- Вставка записи для отеля в Барселоне, Испания  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Arts Barcelona', 'Испания', 'Барселона', 'st. Carrer Marina  
h. 1920/1 ap. 65');

-- Вставка записи для отеля в Риме, Италия  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Eden', 'Италия', 'Рим', 'st. Via Ludovisi h. 49/4 ap. 123');

-- Вставка записи для отеля в Лиссабоне, Португалия  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Ritz Lisbon', 'Португалия', 'Лиссабон', 'st. Rodrigo  
Fonseca h. 88/2 ap. 456');

-- Вставка записи для отеля в Праге, Чехия  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Aria', 'Чехия', 'Прага', 'st. Celetna has h. 1036/11 ap. 265');

-- Вставка записи для отеля в Будапеште, Венгрия  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Corinthia Budapest', 'Венгрия', 'Будапешт', 'st. Erzsebet  
korut h. 434/56 ap. 76');

-----

-- Вставка записи для отеля в Вене, Австрия  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Sacher Wien', 'Австрия', 'Вена', 'st. Philharmonicstrasse  
h. 4/1 ap. 1');

-- Вставка записи для отеля в Берлине, Германия  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Hotel Adlon Kempinski Berlin', 'Германия', 'Берлин', 'st. Unter  
Linden h. 77/1 ap. 1');

-- Вставка записи для отеля в Лондоне, Великобритания  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('The Ritz London', 'Великобритания', 'Лондон', 'st. Piccadilly h.  
150/1 ap. 1');

-- Вставка записи для отеля в Дубае, ОАЭ  
INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('Burj Al Arab Jumeirah', 'ОАЭ', 'Дубай', 'st. Jumeirah Road h.  
33/1 ap. 1');

```

-- Вставка записи для отеля в Абу-Даби, ОАЭ
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('Emirates Palace', 'ОАЭ', 'Абу-Даби', 'st. West Corniche h. 92/1
ap. 1');

-- Вставка записи для отеля в Дохе, Катар
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('The St. Regis Doha', 'Катар', 'Доха', 'st. Corniche Road h. 54/1
ap. 1');

-- Вставка записи для отеля в Маскате, Оман
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('The Chedi Muscat', 'Оман', 'Маскат', 'st. North Ghubrah h. 18/1
ap. 1');

-- Вставка записи для отеля в Куала-Лумпуре, Малайзия
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('The Ritz-Carlton, Kuala Lumpur', 'Малайзия', 'Куала-Лумпур',
'st. Jalan Imbi h. 168/1 ap. 1');

-- Вставка записи для отеля в Сингапуре
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('Marina Bay Sands', 'Сингапур', 'Сингапур', 'st. Bayfront Avenue
h. 10/1 ap. 1');

-- Вставка записи для отеля в Гонконге
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('The Peninsula Hong Kong', 'Гонконг', 'Гонконг', 'st. Salisbury
Road h. 2/1 ap. 1');

-- Вставка записи для отеля в Токио, Япония
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('The Ritz-Carlton, Tokyo', 'Япония', 'Токио', 'st. Aka saka h. 6/1
ap. 1');

-- Вставка записи для отеля в Нью-Йорке, США
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)
VALUES ('The Plaza Hotel', 'США', 'Нью-Йорк', 'st. Fifth Avenue h. 768/1
ap. 1');

-- Вставка записи для отеля в Лос-Анджелесе, США
INSERT INTO Hotel (hotel_name, hotel_country, hotel_city, hotel_address)

```



VALUES ('The Beverly Hills Hotel', 'США', 'Лос-Анджелес', 'st. Sunset Boulevard h. 9648/1 ap. 1');

-- Вставка записи для отеля в Сан-Франциско, США

INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('The Ritz-Carlton, San Francisco', 'США', 'Сан-Франциско', 'st. Stockton Street h. 600/1 ap. 1');

-- Вставка записи для отеля в Майами, США

INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('The Setai Miami Beach', 'США', 'Майами', 'st. Collins Avenue h. 2001/1 ap. 1');

-- Вставка записи для отеля в Лас-Вегасе, США

INSERT INTO Hotel (hotel\_name, hotel\_country, hotel\_city, hotel\_address)  
VALUES ('The Bellagio', 'США', 'Лас-Вегас', 'st. Lagas Boulevard h. 3600/1 ap. 1');

-- Заполнение таблицы ОТЕЛИ КОМНАТЫ

--Для отеля в Зельдене

INSERT INTO Hotel\_rooms (hotel\_id, room\_type\_id,  
number\_of\_available\_rooms, room\_rate\_per\_night)  
VALUES (8, 1, 20, 100.0);

INSERT INTO Hotel\_rooms (hotel\_id, room\_type\_id,  
number\_of\_available\_rooms, room\_rate\_per\_night)  
VALUES (8, 2, 50, 100.0);

INSERT INTO Hotel\_rooms (hotel\_id, room\_type\_id,  
number\_of\_available\_rooms, room\_rate\_per\_night)  
VALUES (8, 3, 25, 100.0);

INSERT INTO Hotel\_rooms (hotel\_id, room\_type\_id,  
number\_of\_available\_rooms, room\_rate\_per\_night)  
VALUES (8, 4, 30, 100.0);

--Для отеля в Мале

INSERT INTO Hotel\_rooms (hotel\_id, room\_type\_id,  
number\_of\_available\_rooms, room\_rate\_per\_night)  
VALUES (11, 1, 15, 120.0);

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (11, 2, 40, 120.0);
```

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (11, 3, 20, 120.0);
```

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (11, 4, 20, 120.0);
```

--Для отеля в Милане

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (16, 1, 10, 150.0);
```

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (16, 2, 40, 150.0);
```

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (16, 3, 25, 150.0);
```

```
INSERT INTO Hotel_rooms (hotel_id, room_type_id,  
number_of_available_rooms, room_rate_per_night)  
VALUES (16, 4, 40, 150.0);
```

-- Заполнение таблицы ОТЕЛЬНАЯ КАРТА

-- Турист 1:

```
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,  
check_in_date, eviction_date)  
VALUES (8, 3, TO_DATE('2024-01-10', 'YYYY-MM-DD'),  
TO_DATE('2024-01-17', 'YYYY-MM-DD'));
```

-- Турист 2:

```
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,  
check_in_date, eviction_date)  
VALUES (8, 3, TO_DATE('2024-01-10', 'YYYY-MM-DD'),  
TO_DATE('2024-01-17', 'YYYY-MM-DD'));
```

```
-- Турист 3:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
VALUES (8, 3, TO_DATE('2024-01-10', 'YYYY-MM-DD'),
TO_DATE('2024-01-17', 'YYYY-MM-DD'));
```

---

```
-- Турист 4:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
VALUES (11, 4, TO_DATE('2024-07-01', 'YYYY-MM-DD'),
TO_DATE('2024-07-10', 'YYYY-MM-DD'));
```

```
-- Турист 5:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
VALUES (11, 2, TO_DATE('2024-07-01', 'YYYY-MM-DD'),
TO_DATE('2024-07-10', 'YYYY-MM-DD'));
```

```
-- Турист 6:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
VALUES (11, 2, TO_DATE('2024-07-01', 'YYYY-MM-DD'),
TO_DATE('2024-07-10', 'YYYY-MM-DD'));
```

-----

```
-- Турист 7:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
VALUES (16, 1, TO_DATE('2024-04-15', 'YYYY-MM-DD'),
TO_DATE('2024-04-22', 'YYYY-MM-DD'));
```

```
-- Турист 8:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
VALUES (16, 1, TO_DATE('2024-04-15', 'YYYY-MM-DD'),
TO_DATE('2024-04-22', 'YYYY-MM-DD'));
```

```
-- Турист 9:
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,
check_in_date, eviction_date)
```

```
VALUES(16, 1, TO_DATE('2024-04-15', 'YYYY-MM-DD'),  
TO_DATE('2024-04-22', 'YYYY-MM-DD'));
```

-- Турист 10:

```
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,  
check_in_date, eviction_date)  
VALUES (16, 1, TO_DATE('2024-04-15', 'YYYY-MM-DD'),  
TO_DATE('2024-04-22', 'YYYY-MM-DD'));
```

-- Турист 11:

```
INSERT INTO Hotel_card (booked_hotel_id, reserved_room_id,  
check_in_date, eviction_date)  
VALUES (16, 1, TO_DATE('2024-04-15', 'YYYY-MM-DD'),  
TO_DATE('2024-04-22', 'YYYY-MM-DD'));
```

-- Заполнение таблицы КАРТА ТУРА

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (27, 1, 1, 2);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (28, 1, 2, 3);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (29, 1, 3, 4);
```

-----

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (30, 2, 4, 5);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (31, 2, 5, 6);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (41, 2, 6, 7);
```

-----

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (21, 3, 7, 8);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (22, 3, 8, 9);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (24, 3, 9, 10);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (25, 3, 10, 11);
```

```
INSERT INTO Tour_card (tourist_id, tour_id, flights_card_id,  
hotel_card_id)  
VALUES (26, 3, 11, 12);
```

-- Заполнение таблицы ЭКСКУРСИИ

```
-- Вставка экскурсий для тура "Зимний Горнолыжный Рай"  
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES ('Альпийский Гид', 'Горнолыжный Спуск с Инструктором',  
'Индивидуальный спуск с опытным инструктором по живописным  
горным склонам. Улучшите свои навыки катания на лыжах или  
сноуборде и насладитесь захватывающими видами.', 'Австрия',  
'Зельден', TO_TIMESTAMP('2024-01-11 10:00:00', 'YYYY-MM-DD  
HH24:MI:SS'), TO_TIMESTAMP('2024-01-11 12:00:00', 'YYYY-MM-DD  
HH24:MI:SS'), 50, 20);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES ('Альпийский Гид', 'Снегоходный Тур по Горным Трейлам',  
'Исследуйте заснеженные пейзажи на мощном снегоходе. Откройте для  
себя скрытые уголки гор и насладитесь незабываемыми видами.',  
'Австрия', 'Зельден', TO_TIMESTAMP('2024-01-12 14:00:00', 'YYYY-
```

MM-DD HH24:MI:SS'), TO\_TIMESTAMP('2024-01-12 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 70, 30);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name, excursion\_description, excursion\_country, excursion\_city, excursion\_start, excursion\_end, excursion\_cost, expenses)  
VALUES ('Ледяной Дворец', 'Посещение Ледяного Дворца и Катание на Коньках', 'Окунитесь в зимнюю сказку в ледяном дворце. Насладитесь катанием на коньках под мерцающими огнями и полюбуйте льдяными скульптурами.', 'Австрия', 'Зельден', TO\_TIMESTAMP('2024-01-13 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO\_TIMESTAMP('2024-01-13 20:00:00', 'YYYY-MM-DD HH24:MI:SS'), 30, 15);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name, excursion\_description, excursion\_country, excursion\_city, excursion\_start, excursion\_end, excursion\_cost, expenses)  
VALUES ('Ледяной Дворец', 'Посещение Горной Фермы и Дегустация Сыров', 'Посетите традиционную горную ферму и узнайте о местном производстве сыров. Насладитесь дегустацией различных видов сыров и познакомьтесь с фермерским бытом.', 'Австрия', 'Зельден', TO\_TIMESTAMP('2024-01-14 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO\_TIMESTAMP('2024-01-14 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 40, 20);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name, excursion\_description, excursion\_country, excursion\_city, excursion\_start, excursion\_end, excursion\_cost, expenses)  
VALUES ('Ледяной Дворец', 'Посещение Зимнего Рынка и Рождественской Ярмарки', 'Погрузитесь в атмосферу зимнего праздника на рождественской ярмарке. Насладитесь традиционными угощениями, сувенирами и праздничной музыкой.', 'Австрия', 'Зельден', TO\_TIMESTAMP('2024-01-15 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO\_TIMESTAMP('2024-01-15 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 20, 10);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name, excursion\_description, excursion\_country, excursion\_city, excursion\_start, excursion\_end, excursion\_cost, expenses)  
VALUES ('Ледяной Дворец', 'Строительство Снежного Форта и Снежные Бои', 'Соберите команду и постройте снежный форт. Устройте веселые снежные бои и насладитесь зимними забавами.', 'Австрия', 'Зельден', TO\_TIMESTAMP('2024-01-16 10:00:00', 'YYYY-MM-DD

HH24:MI:SS'), TO\_TIMESTAMP('2024-01-16 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 30, 15);

-- Вставка экскурсий для тура "Тропический Рай"

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Экзотик Трэвел', 'Дневной Круиз по Островам с Обедом и
Сноркелингом', 'Отправьтесь в круиз по живописным островам.
Насладитесь обедом на борту и исследуйте подводный мир, занимаясь
сноркелингом.', 'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-02
10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-02
16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 80, 40);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Экзотик Трэвел', 'Дайвинг-Экскурсия к Подводным Рифам',
'Исследуйте подводный мир Мальдивских островов с помощью
дайвинга. Погрузитесь в кристально чистые воды и познакомьтесь с
морскими обитателями.', 'Мальдивы', 'Мале', TO_TIMESTAMP('2024-
07-03 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-
07-03 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 100, 50);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Экзотик Трэвел', 'Массаж и Спа-Процедуры на Пляже',
'Побалуйте себя массажем и спа-процедурами на берегу океана.
Расслабьтесь и насладитесь тропическим блаженством.', 'Мальдивы',
'Мале', TO_TIMESTAMP('2024-07-04 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'), TO_TIMESTAMP('2024-07-04 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 50, 25);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Экзотик Трэвел', 'Вечерняя Рыбалка на Тропических Водах',
'Отправьтесь на рыбалку на закате и насладитесь красотой
тропического вечера. Попробуйте поймать экзотических рыб и
насладитесь свежими морепродуктами.', 'Мальдивы', 'Мале',
TO_TIMESTAMP('2024-07-05 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
```

TO\_TIMESTAMP('2024-07-05 20:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
60, 30);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name,  
excursion\_description, excursion\_country, excursion\_city, excursion\_start,  
excursion\_end, excursion\_cost, expenses)  
VALUES ('Экзотик Трэвел', 'Подводный мир Мальдив', 'Исследуйте  
подводный мир Мальдив во время захватывающей экскурсии на лодке  
со стеклянным дном. Наблюдайте за красочными рыбами, кораллами и  
морскими черепахами.', 'Мальдивы', 'Мале', TO\_TIMESTAMP('2024-07-  
02 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO\_TIMESTAMP('2024-07-  
02 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 50, 20);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name,  
excursion\_description, excursion\_country, excursion\_city, excursion\_start,  
excursion\_end, excursion\_cost, expenses)  
VALUES ('Экзотик Трэвел', 'Сафари на острове', 'Отправьтесь на сафари  
по острову и познакомьтесь с уникальной фауной Мальдив. Увидите  
экзотических птиц, летучих лисиц и гигантских ящериц.', 'Мальдивы',  
'Мале', TO\_TIMESTAMP('2024-07-04 14:00:00', 'YYYY-MM-DD  
HH24:MI:SS'), TO\_TIMESTAMP('2024-07-04 18:00:00', 'YYYY-MM-DD  
HH24:MI:SS'), 70, 30);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name,  
excursion\_description, excursion\_country, excursion\_city, excursion\_start,  
excursion\_end, excursion\_cost, expenses)  
VALUES ('Коралловый Риф', 'Рыбалка в открытом море', 'Испытайте  
удачу на рыбалке в открытом море. Поймайте рыбу-меч, тунца или  
барракуду и насладитесь свежим уловом на ужин.', 'Мальдивы', 'Мале',  
TO\_TIMESTAMP('2024-07-06 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
TO\_TIMESTAMP('2024-07-06 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
80, 40);

INSERT INTO Excursion (excursion\_agency\_name, excursion\_name,  
excursion\_description, excursion\_country, excursion\_city, excursion\_start,  
excursion\_end, excursion\_cost, expenses)  
VALUES ('Коралловый Риф', 'Круиз на закате', 'Насладитесь  
романтическим круизом на закате вдоль побережья Мальдив.  
Наблюдайте за живописными островами и наслаждайтесь ужином под  
открытым небом.', 'Мальдивы', 'Мале', TO\_TIMESTAMP('2024-07-08  
17:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO\_TIMESTAMP('2024-07-08  
19:00:00', 'YYYY-MM-DD HH24:MI:SS'), 60, 25);



```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Коралловый Риф', 'Сноркелинг на коралловом рифе',
'Исследуйте подводный мир кораллового рифа во время экскурсии по
сноркелингу. Увидите красочных рыб, кораллы и морских черепах.',
'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-03 10:00:00', 'YYYY-
MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-03 12:00:00', 'YYYY-
MM-DD HH24:MI:SS'), 50, 20);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Экзотик Трэвел', 'Посещение необитаемого острова',
'Отправьтесь на экскурсию на необитаемый остров и проведите день
вдали от цивилизации. Наслаждайтесь пляжем, морем и тишиной.',
'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-05 10:00:00', 'YYYY-
MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-05 16:00:00', 'YYYY-
MM-DD HH24:MI:SS'), 90, 45);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Пляжный гид', 'Дайвинг с акулами', 'Испытайте острые
ощущения от дайвинга с акулами. Погрузитесь в морские глубины и
понаблюдайте за этими хищниками в их естественной среде обитания.',
'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-07 10:00:00', 'YYYY-
MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-07 12:00:00', 'YYYY-
MM-DD HH24:MI:SS'), 100, 50);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Пляжный гид', 'Пляжный отдых на частном острове',
'Проведите день на частном острове и наслаждайтесь пляжным
отдыхом. Загорайте, купайтесь в море и занимайтесь водными видами
спорта.', 'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-09 10:00:00',
'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-09 16:00:00',
'YYYY-MM-DD HH24:MI:SS'), 120, 60);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
```

```
VALUES ('Пляжный гид', 'Экскурсия на гидросамолете', 'Отправьтесь на экскурсию на гидросамолете и полюбуйтесь Мальдивами с высоты птичьего полета. Наслаждайтесь панорамными видами на острова и лагуны.', 'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-02 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-02 16:00:00', 'YYYY-MM-DD HH24:MI:SS'), 150, 75);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name, excursion_description, excursion_country, excursion_city, excursion_start, excursion_end, excursion_cost, expenses)  
VALUES ('Пляжный гид', 'Посещение местной деревни', 'Посетите местную деревню и познакомьтесь с культурой и традициями мальдивского народа. Узнайте о местном образе жизни, ремеслах и кулинарии.', 'Мальдивы', 'Мале', TO_TIMESTAMP('2024-07-04 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-07-04 12:00:00', 'YYYY-MM-DD HH24:MI:SS'), 40, 15);
```

-- Экскурсии для тура "Шопинг в Модной Столице"

```
INSERT INTO Excursion (excursion_agency_name, excursion_name, excursion_description, excursion_country, excursion_city, excursion_start, excursion_end, excursion_cost, expenses)  
VALUES ('Модный Милан', 'Экскурсия по магазинам Милана', 'Отправьтесь на экскурсию по магазинам Милана и посетите самые известные бутики итальянских дизайнеров. Насладитесь шопингом в роскошных торговых центрах и аутлетах.', 'Италия', 'Милан', TO_TIMESTAMP('2024-04-16 10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-04-16 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 100, 50);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name, excursion_description, excursion_country, excursion_city, excursion_start, excursion_end, excursion_cost, expenses)  
VALUES ('Модный Милан', 'Посещение Недели моды в Милане', 'Посетите Неделю моды в Милане и окунитесь в мир высокой моды. Наблюдайте за показами последних коллекций известных дизайнеров и наслаждайтесь атмосферой модного праздника.', 'Италия', 'Милан', TO_TIMESTAMP('2024-04-18 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-04-18 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 150, 75);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Модный Милан', 'Прогулка по улице моды', 'Прогуляйтесь по
знаменитой улице моды Виа Монтенаполеоне и посетите самые
экслюзивные бутики Милана. Насладитесь шопингом в окружении
роскоши и гламура.', 'Италия', 'Милан', TO_TIMESTAMP('2024-04-20
10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-04-20
14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 80, 40);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Аутлет Серравалле', 'Посещение крупнейшего аутлета
Италии', 'Отправьтесь в крупнейший аутлет Италии Серравалле и
воспользуйтесь скидками на товары известных брендов. Насладитесь
шопингом в комфортной обстановке и приобретите дизайнерские вещи
по выгодным ценам.', 'Италия', 'Милан', TO_TIMESTAMP('2024-04-17
10:00:00', 'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2024-04-17
18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 120, 60);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Аутлет Серравалле', 'Посещение музея дизайна', 'Посетите
музей дизайна в Милане и познакомьтесь с историей и развитием
итальянского дизайна. Насладитесь экспозицией мебели, одежды,
аксессуаров и других предметов дизайна.', 'Италия', 'Милан',
TO_TIMESTAMP('2024-04-19 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-04-19 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
60, 30);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES ('Аутлет Серравалле', 'Показ мод в Милане', 'Посетите показ
мод в Милане и насладитесь творчеством итальянских дизайнеров.
Наблюдайте за дефиле моделей и оцените последние тенденции моды.',
'Италия', 'Милан', TO_TIMESTAMP('2024-04-21 19:00:00', 'YYYY-MM-
DD HH24:MI:SS'), TO_TIMESTAMP('2024-04-21 21:00:00', 'YYYY-MM-
DD HH24:MI:SS'), 100, 50);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES ('Аутлет Серравалле', 'Кулинарный мастер-класс', 'Примите  
участие в кулинарном мастер-классе и научитесь готовить  
традиционные блюда итальянской кухни. Насладитесь вкусом пасты,  
пиццы и других итальянских деликатесов.', 'Италия', 'Милан',  
TO_TIMESTAMP('2024-04-15 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
TO_TIMESTAMP('2024-04-15 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
80, 40);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES ('Аутлет Серравалле', 'Прогулка по архитектурным  
достопримечательностям', 'Отправьтесь на прогулку по архитектурным  
достопримечательностям Милана и полюбуйте шедеврами  
итальянской архитектуры. Посетите Дуомо, Галерею Виктора  
Эммануила II и другие известные здания.', 'Италия', 'Милан',  
TO_TIMESTAMP('2024-04-17 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
TO_TIMESTAMP('2024-04-17 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
70, 35);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES ('Аутлет Серравалле', 'Посещение исторических мест',  
'Посетите исторические места Милана и узнайте о богатом прошлом  
города. Посетите замок Сфорца, базилику Сант-Амброджо и другие  
исторические достопримечательности.', 'Италия', 'Милан',  
TO_TIMESTAMP('2024-04-19 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
TO_TIMESTAMP('2024-04-19 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
60, 30);
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES ('Аутлет Серравалле', 'Посещение музеев и театров', 'Посетите  
музеи и театры Милана и познакомьтесь с культурным наследием  
города. Посетите музей театра Ла Скала, музей современного искусства  
и другие культурные достопримечательности.', 'Италия', 'Милан',  
TO_TIMESTAMP('2024-04-21 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),
```

```
TO_TIMESTAMP('2024-04-21 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
90, 45);
```

```
-- Вставка записей для экскурсий тура "Романтический Париж"
```

```
-- Экскурсия 1
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
```

```
VALUES (
```

```
  'Мулен Руж',
```

```
  'Однодневная поездка на Французскую Ривьеру',
```

```
  'Посетите гламурные города Канны и Ниццу, полюбуйтесь
потрясающими видами на Средиземное море и насладитесь роскошной
атмосферой Французской Ривьеры.',
```

```
  'Франция',
```

```
  'Париж',
```

```
  TO_TIMESTAMP('2024-05-11 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),
```

```
  TO_TIMESTAMP('2024-05-11 20:00:00', 'YYYY-MM-DD HH24:MI:SS'),
```

```
  150,
```

```
  50
```

```
);
```

```
-- Экскурсия 2
```

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
```

```
VALUES (
```

```
  'Версаль',
```

```
  'Посещение знаменитых музеев Парижа',
```

```
  'Посетите всемирно известные музеи Парижа, включая Лувр, Музей
Орсе и Центр Помпиду, и откройте для себя богатую историю и
культуру Франции.',
```

```
  'Франция',
```

```
  'Париж',
```

```
  TO_TIMESTAMP('2024-05-12 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
```

```
  TO_TIMESTAMP('2024-05-12 18:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
```

```
  120,
```

```
  40
```

```
);
```

-- Экскурсия 3

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES (  
    'Версаль',  
    'Посещение дворцов и парков Парижа',  
    'Посетите великолепные дворцы и парки Парижа, включая Версаль,  
Фонтенбло и Люксембургский сад, и окунитесь в историю и красоту  
французской монархии.',  
    'Франция',  
    'Париж',  
    TO_TIMESTAMP('2024-05-13 09:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    TO_TIMESTAMP('2024-05-13 17:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    130,  
    45  
);
```

-- Экскурсия 4

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES (  
    'Версаль',  
    'Ночная экскурсия по Парижу',  
    'Откройте для себя очарование Парижа ночью, посетив Эйфелеву  
башню, Монмартр и другие достопримечательности, которые оживают  
в свете вечерних огней.',  
    'Франция',  
    'Париж',  
    TO_TIMESTAMP('2024-05-14 19:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    TO_TIMESTAMP('2024-05-14 23:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    100,  
    30  
);
```

-- Экскурсия 5

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES (  
    'Мулен Руж',  
    'Речная прогулка по Сене',  
    'Насладитесь живописной речной прогулкой по Сене и полюбуйте  
потрясающими видами на достопримечательности Парижа, включая  
Эйфелеву башню, собор Парижской Богоматери и Лувр.',  
    'Франция',  
    'Париж',  
    TO_TIMESTAMP('2024-05-15 11:00:00', 'YYYY-MM-DD HH24:MI:SS'),  
    TO_TIMESTAMP('2024-05-15 13:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    60,  
    20  
);
```

-- Экскурсия 6

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)  
VALUES (  
    'Мулен Руж',  
    'Винный тур в Шампань',  
    'Отправьтесь в однодневную поездку в регион Шампань, посетите  
виноградники и погреба, продегустируйте знаменитое шампанское и  
узнайте о его производстве.',  
    'Франция',  
    'Париж',  
    TO_TIMESTAMP('2024-05-16 08:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    TO_TIMESTAMP('2024-05-16 19:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
    180,  
    60  
);
```

-- Экскурсия 7

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,  
excursion_description, excursion_country, excursion_city, excursion_start,  
excursion_end, excursion_cost, expenses)
```

```
VALUES (
    'Мулен Руж',
    'Посещение Диснейленда Париж',
    'Подарите себе и своим детям незабываемый день в Диснейленде
Париж, где вас ждут аттракционы, шоу и встречи с любимыми
персонажами Disney.',
    'Франция',
    'Париж',
    TO_TIMESTAMP('2024-05-11 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),
    TO_TIMESTAMP('2024-05-11 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
    120,
    40
);
```

-- Экскурсия 8

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES (
    'Версаль',
    'Посещение Версальского дворца',
    'Посетите великолепный Версальский дворец, бывшую резиденцию
французских королей, и полюбуйте его роскошными интерьерами,
садами и фонтанами.',
    'Франция',
    'Париж',
    TO_TIMESTAMP('2024-05-12 09:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-05-12 17:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    100,
    30
);
```

-- Экскурсия 9

```
INSERT INTO Excursion (excursion_agency_name, excursion_name,
excursion_description, excursion_country, excursion_city, excursion_start,
excursion_end, excursion_cost, expenses)
VALUES (
    'Мулен Руж',
    'Посещение кабаре Мулен Руж',
    'Посетите легендарное кабаре Мулен Руж и насладитесь красочным
шоу с участием танцоров, певцов и акробатов.',
```



```
'Франция',  
'Париж',  
TO_TIMESTAMP('2024-05-13 20:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
TO_TIMESTAMP('2024-05-13 22:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),  
80,  
25  
);
```

-- Заполнение таблицы ЭКСКУРСИОННАЯ КАРТА

```
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 1, 21);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 3, 21);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 4, 21);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 6, 21);
```

```
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 1, 22);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 2, 22);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 3, 22);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 4, 22);
```

```
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 1, 23);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 2, 23);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 3, 23);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 4, 23);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 5, 23);  
INSERT INTO Excursion_card (excursion_id, tour_card_id)  
VALUES ( 6, 23);
```

```

INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 7, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 8, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 10, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 11, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 12, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 14, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 16, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 19, 24);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 20, 24);

```

```

INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 8, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 9, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 10, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 11, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 13, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 14, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 16, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 17, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 18, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 19, 25);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 20, 25);

```

```

INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 7, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 9, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 12, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 14, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 15, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 16, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 17, 26);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 18, 26);

```

```

INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 21, 27);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 22, 27);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 24, 27);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 26, 27);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 27, 27);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 28, 27);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 29, 27);

```

```

INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 21, 28);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 22, 28);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 25, 28);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 26, 28);
INSERT INTO Excursion_card (excursion_id, tour_card_id)
VALUES ( 27, 28);
INSERT INTO Excursion_card (excursion_id, tour_card_id)

```

VALUES ( 30, 28);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 23, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 24, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 25, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 26, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 27, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 29, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 30, 29);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 27, 30);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 28, 30);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 29, 30);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 30, 30);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 21, 31);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 22, 31);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 23, 31);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 24, 31);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 25, 31);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 26, 31);

INSERT INTO Excursion\_card (excursion\_id, tour\_card\_id)  
VALUES ( 27, 31);

-- Заполнение таблицы ГРУЗЫ

-- Вставка записей для грузов тура "Зимний Горнолыжный Рай"

-- Груз 1 - Лыжи

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Лыжи',
    'Горные лыжи для катания по подготовленным трассам.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    5.0,
    0.2,
    1.7,
    0.1,
    CargoType('стандартный')
);
```

-- Груз 2 - Сноуборд

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Сноуборд',
    'Сноуборд для катания по подготовленным трассам и в сноупарках.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    4.0,
    0.3,
    1.5,
    0.1,
    CargoType('стандартный')
);
```

-- Груз 3 - Горнолыжные ботинки

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Горнолыжные ботинки',
    'Горнолыжные ботинки для катания по подготовленным трассам.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    2.5,
    0.3,
    0.2,
    0.3,
    CargoType('стандартный')
);

```

-- Груз 4 - Сноубордические ботинки

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Сноубордические ботинки',
    'Сноубордические ботинки для катания по подготовленным трассам и
в сноупарках.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    2.0,
    0.3,
    0.2,
    0.2,
    CargoType('стандартный')
);

```

-- Груз 5 - Горнолыжный костюм

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Горнолыжный костюм',
    'Горнолыжный костюм для катания по подготовленным трассам.',

```

```

    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.5,
    0.5,
    0.3,
    0.2,
    CargoType('стандартный')
);

```

```

-- Груз 6 - Сноубордический костюм
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Сноубордический костюм',
    'Сноубордический костюм для катания по подготовленным трассам и в
сноупарках.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.3,
    0.4,
    0.2,
    0.2,
    CargoType('стандартный')
);

```

```

-- Груз 7 - Горнолыжная маска
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Горнолыжная маска',
    'Горнолыжная маска для защиты глаз от солнца и снега.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.2,
    0.2,

```

```

0.1,
0.1,
CargoType('стандартный')
);

```

```

-- Груз 8 - Сноубордическая маска

```

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
'Sноубордическая маска',
'Sноубордическая маска для защиты глаз от солнца и снега.',
TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-01-10 07:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
0.2,
0.2,
0.1,
0.1,
CargoType('стандартный')
);

```

```

-- Груз 9 - Термос

```

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
'Термос',
'Термос для хранения горячих напитков на склоне.',
TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-01-10 05:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
0.5,
0.2,
0.2,
0.2,
CargoType('стандартный')
);

```

```

-- Груз 10 - Рюкзак

```



```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Рюкзак',
    'Рюкзак для переноски снаряжения и личных вещей на склоне.',
    TO_TIMESTAMP('2024-01-09 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-10 05:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.3,
    0.5,
    0.2,
    CargoType('стандартный')
);

```

-- Обратные записи для грузов тура "Зимний Горнолыжный Рай"

-- Груз 1 - Лыжи

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Лыжи',
    'Горные лыжи для катания по подготовленным трассам.',
    TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    5.0,
    0.2,
    1.7,
    0.1,
    CargoType('стандартный')
);

```

-- Груз 2 - Сноуборд

```

INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (

```

```

'Сноуборд',
'Сноуборд для катания по подготовленным трассам и в сноупарках.',
TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
4.0,
0.3,
1.5,
0.1,
CargoType('стандартный')
);

```

```

-- Груз 3 - Горнолыжные ботинки
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
'Горнолыжные ботинки',
'Горнолыжные ботинки для катания по подготовленным трассам.',
TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
2.5,
0.3,
0.2,
0.3,
CargoType('стандартный')
);

```

```

-- Груз 4 - Сноубордические ботинки
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
'Сноубордические ботинки',
'Сноубордические ботинки для катания по подготовленным трассам и
в сноупарках.',
TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),

```

```

2.0,
0.3,
0.2,
0.2,
CargoType('стандартный')
);

```

```

-- Груз 5 - Горнолыжный костюм
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Горнолыжный костюм',
    'Горнолыжный костюм для катания по подготовленным трассам.',
    TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.5,
    0.5,
    0.3,
    0.2,
    CargoType('стандартный')
);

```

```

-- Груз 6 - Сноубордический костюм
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Сноубордический костюм',
    'Сноубордический костюм для катания по подготовленным трассам и в
сноупарках.',
    TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.3,
    0.4,
    0.2,
    0.2,
    CargoType('стандартный')
);

```

```

-- Груз 7 - Горнолыжная маска
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Горнолыжная маска',
    'Горнолыжная маска для защиты глаз от солнца и снега.',
    TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.2,
    0.2,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 8 - Сноубордическая маска
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Сноубордическая маска',
    'Сноубордическая маска для защиты глаз от солнца и снега.',
    TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.2,
    0.2,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 9 - Термос
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Термос',

```

```

        'Термос для хранения горячих напитков на склоне.',
        TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
        TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
        0.5,
        0.2,
        0.2,
        0.2,
        CargoType('стандартный')
    );

```

```

-- Груз 10 - Рюкзак
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Рюкзак',
    'Рюкзак для переноски снаряжения и личных вещей на склоне.',
    TO_TIMESTAMP('2024-01-17 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-01-17 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.3,
    0.5,
    0.2,
    CargoType('стандартный')
);

```

```

-- Груз 1 - Купальник
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Купальник',
    'Стильный купальник для плавания и принятия солнечных ванн.',
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-01 03:00:00', 'YYYY-MM-DD
HH24:MI:SS'),

```

```

0.3,
0.2,
0.1,
0.1,
CargoType('стандартный')
);

```

```

-- Груз 2 - Плавки
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Плавки',
    'Удобные плавки для плавания и принятия солнечных ванн.',
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-01 03:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.2,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 3 - Солнцезащитный крем
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Солнцезащитный крем',
    'Солнцезащитный крем для защиты кожи от вредного воздействия
ультрафиолетовых лучей.',
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-01 03:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.2,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 4 - Солнцезащитные очки
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Солнцезащитные очки',
    'Солнцезащитные очки для защиты глаз от вредного воздействия
ультрафиолетовых лучей.',
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-01 02:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.1,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 5 - Головной убор
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Головной убор',
    'Головной убор для защиты головы от солнца и перегрева.',
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-01 02:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.1,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 6 - Легкая одежда
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (

```

```

'Легкая одежда',
'Легкая одежда для жаркой погоды.',
TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-07-01 02:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
1.0,
0.3,
0.2,
0.2,
CargoType('стандартный')
);

-- Груз 7 - Обувь для пляжа
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
'Обувь для пляжа',
'Обувь для пляжа и прогулок по песку.',
TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-07-01 02:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
0.5,
0.2,
0.1,
0.2,
CargoType('стандартный')
);

-- Груз 8 - Сумка для пляжа
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
'Сумка для пляжа',
'Сумка для переноски пляжных принадлежностей.',
TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-07-01 05:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
0.5,

```



```
0.3,  
0.2,  
0.2,  
CargoType('стандартный')  
);
```

```
-- Груз 9 - Полотенце для пляжа
```

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,  
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,  
cargo_type)
```

```
VALUES (
```

```
    'Полотенце для пляжа',
```

```
    'Полотенце для пляжа и бассейна.',
```

```
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    TO_TIMESTAMP('2024-07-01 05:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    0.3,
```

```
    0.2,
```

```
    0.1,
```

```
    0.1,
```

```
    CargoType('стандартный')
```

```
);
```

```
-- Груз 10 - Книга или электронная книга
```

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,  
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,  
cargo_type)
```

```
VALUES (
```

```
    'Книга или электронная книга',
```

```
    'Книга или электронная книга для чтения на пляже или в самолете.',
```

```
    TO_TIMESTAMP('2024-06-30 15:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    TO_TIMESTAMP('2024-07-01 05:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    0.5,
```

```
    0.2,
```

```
    0.2,
```

```
    0.1,
```

```
    CargoType('стандартный')
```

```
);
```

-- Обратные записи для грузов тура "Тропический Рай"

-- Груз 1 - Купальник

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Купальник',
    'Стильный купальник для плавания и принятия солнечных ванн.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.3,
    0.2,
    0.1,
    0.1,
    CargoType('стандартный')
);
```

-- Груз 2 - Плавки

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Плавки',
    'Удобные плавки для плавания и принятия солнечных ванн.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.2,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);
```

-- Груз 3 - Солнцезащитный крем

```
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
```

```

        'Солнцезащитный крем',
        'Солнцезащитный крем для защиты кожи от вредного воздействия
        ультрафиолетовых лучей.',
        TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
        HH24:MI:SS'),
        TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
        HH24:MI:SS'),
        0.2,
        0.1,
        0.1,
        0.1,
        CargoType('стандартный')
    );

```

```

-- Груз 4 - Солнцезащитные очки
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Солнцезащитные очки',
    'Солнцезащитные очки для защиты глаз от вредного воздействия
    ультрафиолетовых лучей.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    0.1,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 5 - Головной убор
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Головной убор',
    'Головной убор для защиты головы от солнца и перегрева.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),

```

```

    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.1,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 6 - Легкая одежда
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Легкая одежда',
    'Легкая одежда для жаркой погоды.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.3,
    0.2,
    0.2,
    CargoType('стандартный')
);

```

```

-- Груз 7 - Обувь для пляжа
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Обувь для пляжа',
    'Обувь для пляжа и прогулок по песку.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.5,
    0.2,
    0.1,
    0.2,
    CargoType('стандартный')
);

```

```
);

-- Груз 8 - Сумка для пляжа
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Сумка для пляжа',
    'Сумка для переноски пляжных принадлежностей.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.5,
    0.3,
    0.2,
    0.2,
    CargoType('стандартный')
);
```

```
-- Груз 9 - Полотенце для пляжа
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
    'Полотенце для пляжа',
    'Полотенце для пляжа и бассейна.',
    TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.3,
    0.2,
    0.1,
    0.1,
    CargoType('стандартный')
);
```

```
-- Груз 10 - Книга или электронная книга
INSERT INTO Cargo (cargo_name, cargo_description, receipt_date,
departure_date, cargo_weight, cargo_width, cargo_height, cargo_depth,
cargo_type)
VALUES (
```

```

'Книга или электронная книга',
'Книга или электронная книга для чтения на пляже или в самолете.',
TO_TIMESTAMP('2024-07-10 12:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-07-10 13:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
0.5,
0.2,
0.2,
0.1,
CargoType('стандартный')
);

```

---- Обратные записи для грузов тура "Шоппинг в модной столице"  
(Для этого тура только обратные)

```

-- Груз 1 - Новая дизайнерская одежда
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Платье от Prada',
    'Элегантное платье от всемирно известного бренда Prada. Идеально
    подойдет для любого особого случая.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    2.0,
    0.4,
    0.3,
    0.2,
    CargoType('стандартный')
);

```

```
-- Груз 2 - Туфли от Gucci
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Туфли от Gucci',
    'Стильные туфли от известного бренда Gucci. Идеально подойдут для
создания элегантного образа.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.5,
    0.3,
    0.2,
    0.2,
    CargoType('стандартный')
);
```

```
-- Груз 3 - Сумка от Louis Vuitton
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Сумка от Louis Vuitton',
    'Вместительная сумка от известного бренда Louis Vuitton. Идеально
подойдет для путешествий или повседневного использования.',
```

```

    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    3.0,
    0.5,
    0.4,
    0.3,
    CargoType('стандартный')
);

```

-- Груз 4 - Кошелек от Bottega Veneta

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Кошелек от Bottega Veneta',
    'Стильный кошелек от известного бренда Bottega Veneta. Идеально
подойдет для хранения денег и карт.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.5,
    0.2,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

-- Груз 5 - Украшения от Cartier

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,

```



```

receipt_date,
departure_date,
cargo_weight,
cargo_width,
cargo_height,
cargo_depth,
cargo_type
)
VALUES (
    'Украшения от Cartier',
    'Элегантные украшения от известного бренда Cartier. Идеально
    подойдут для создания стильного образа.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    0.2,
    0.1,
    0.1,
    0.1,
    CargoType('стандартный')
);

```

```

-- Груз 6 - Косметика от Giorgio Armani
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Косметика от Giorgio Armani',
    'Элитная косметика от известного бренда Giorgio Armani. Идеально
    подойдет для создания безупречного макияжа.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),

```

```
1.0,  
0.3,  
0.2,  
0.2,  
CargoType('стандартный')  
);
```

```
-- Груз 7 - Парфюм от Versace
```

```
INSERT INTO Cargo (
```

```
    cargo_name,  
    cargo_description,  
    receipt_date,  
    departure_date,  
    cargo_weight,  
    cargo_width,  
    cargo_height,  
    cargo_depth,  
    cargo_type
```

```
)
```

```
VALUES (
```

```
    'Парфюм от Versace',
```

```
    'Элитный парфюм от известного бренда Versace. Идеально подойдет  
для создания незабываемого образа.',
```

```
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    0.5,
```

```
    0.2,
```

```
    0.2,
```

```
    0.2,
```

```
    CargoType('стандартный')
```

```
);
```

```
-- Груз 8 - Электроника от Apple
```

```
INSERT INTO Cargo (
```

```
    cargo_name,  
    cargo_description,  
    receipt_date,  
    departure_date,  
    cargo_weight,  
    cargo_width,  
    cargo_height,
```

```

    cargo_depth,
    cargo_type
)
VALUES (
    'iPhone 15 Pro Max',
    'Новейший смартфон от Apple с мощным процессором, отличной
камерой и длительным временем автономной работы.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.3,
    0.2,
    0.2,
    CargoType('стандартный')
);

```

-- Груз 9 - Очки от Dolce&Gabbana

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Очки от Dolce Gabbana',
    'Стильные очки от известного бренда Dolce Gabbana. Идеально
подойдут для защиты глаз от солнечных лучей.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    0.3,
    0.2,
    0.1,
    0.1,
    CargoType('стандартный')
)

```

```
);
```

```
-- Груз 10 - Негабаритный груз - Велосипед
```

```
INSERT INTO Cargo (
```

```
    cargo_name,
```

```
    cargo_description,
```

```
    receipt_date,
```

```
    departure_date,
```

```
    cargo_weight,
```

```
    cargo_width,
```

```
    cargo_height,
```

```
    cargo_depth,
```

```
    cargo_type
```

```
)
```

```
VALUES (
```

```
    'Велосипед',
```

```
    'Торный велосипед известного бренда Specialized. Идеально подойдет  
для любителей активного отдыха.',
```

```
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD  
HH24:MI:SS'),
```

```
    15.0,
```

```
    1.2,
```

```
    0.8,
```

```
    0.5,
```

```
    CargoType('негабаритный')
```

```
);
```

```
-- Груз 11 - Жидкий груз - Вино
```

```
INSERT INTO Cargo (
```

```
    cargo_name,
```

```
    cargo_description,
```

```
    receipt_date,
```

```
    departure_date,
```

```
    cargo_weight,
```

```
    cargo_width,
```

```
    cargo_height,
```

```
    cargo_depth,
```

```
    cargo_type
```

```
)
```

```
VALUES (
```

```

'Вино',
'Бутылка вина известного итальянского бренда Barolo. Идеально
подойдет для ценителей изысканных напитков.',
TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
1.5,
0.1,
0.3,
0.1,
CargoType('жидкий')
);

-- Груз 12 - Сыпучий груз - Кофе
INSERT INTO Cargo (
cargo_name,
cargo_description,
receipt_date,
departure_date,
cargo_weight,
cargo_width,
cargo_height,
cargo_depth,
cargo_type
)
VALUES (
'Кофе',
'Упаковку кофе известного итальянского бренда Lavazza. Идеально
подойдет для любителей ароматного напитка.',
TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
1.0,
0.2,
0.2,
0.2,
CargoType('сыпучий')
);

-- Груз 13 - Стандартный груз - Одежда
INSERT INTO Cargo (

```

```

cargo_name,
cargo_description,
receipt_date,
departure_date,
cargo_weight,
cargo_width,
cargo_height,
cargo_depth,
cargo_type
)
VALUES (
'Одежда',
'Набор одежды известного итальянского бренда Gucci. Идеально
подойдет для создания стильного образа.',
TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
5.0,
0.5,
0.4,
0.3,
CargoType('стандартный')
);

```

```

-- Груз 14 - Негабаритный груз - Сноуборд
INSERT INTO Cargo (
cargo_name,
cargo_description,
receipt_date,
departure_date,
cargo_weight,
cargo_width,
cargo_height,
cargo_depth,
cargo_type
)
VALUES (
'Сноуборд',
'Сноуборд известного бренда Burton. Идеально подойдет для
любителей зимнего спорта.',
TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),

```

```

    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    10.0,
    1.5,
    0.2,
    0.2,
    CargoType('негабаритный')
);

```

-- Груз 15 - Жидкий груз - Оливковое масло

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Оливковое масло',
    'Бутылка оливкового масла известного итальянского бренда Bertolli.
Идеально подойдет для приготовления вкусных блюд.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.2,
    0.3,
    0.1,
    CargoType('жидкий')
);

```

-- Груз 16 - Сыпучий груз - Рис

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,

```

```

cargo_width,
cargo_height,
cargo_depth,
cargo_type
)
VALUES (
    'Рис',
    'Упаковку риса известного итальянского бренда Riso Gallo. Идеально
    подойдет для приготовления вкусных блюд.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    2.0,
    0.3,
    0.2,
    0.2,
    CargoType('сыпучий')
);

```

-- Груз 17 - Стандартный груз - Косметика

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Косметика',
    'Набор косметики известного итальянского бренда Kiko Milano.
    Идеально подойдет для создания безупречного макияжа.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    2.0,
    0.3,
    0.2,

```



```

0.2,
CargoType('стандартный')
);

-- Груз 18 - Негабаритный груз - Лыжи
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Лыжи',
    'Лыжи известного бренда Atomic. Идеально подойдут для любителей зимнего спорта.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD HH24:MI:SS'),
    12.0,
    1.8,
    0.2,
    0.2,
    CargoType('негабаритный')
);

```

```

-- Груз 19 - Негабаритный груз - Картина
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type

```

```

)
VALUES (
  'Картина',
  'Картина известного итальянского художника. Идеально подойдет для
украшения интерьера.',
  TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
  TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
  5.0,
  1.0,
  0.8,
  0.2,
  CargoType('негабаритный')
);

```

-- Груз 20 - Жидкий груз - Духи

```

INSERT INTO Cargo (
  cargo_name,
  cargo_description,
  receipt_date,
  departure_date,
  cargo_weight,
  cargo_width,
  cargo_height,
  cargo_depth,
  cargo_type
)
VALUES (
  'Духи',
  'Флакон духов известного итальянского бренда. Идеально подойдет
для создания неповторимого образа.',
  TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
  TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
  0.5,
  0.1,
  0.1,
  0.1,
  CargoType('жидкий')
);

```

-- Груз 21 - Сыпучий груз - Специи

INSERT INTO Cargo (

cargo\_name,  
cargo\_description,  
receipt\_date,  
departure\_date,  
cargo\_weight,  
cargo\_width,  
cargo\_height,  
cargo\_depth,  
cargo\_type

)

VALUES (

'Специи',

'Набор специй известного итальянского бренда. Идеально подойдет  
для приготовления вкусных блюд.',

TO\_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD  
HH24:MI:SS'),

TO\_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD  
HH24:MI:SS'),

1.0,

0.2,

0.2,

0.2,

CargoType('сыпучий')

);

-- Груз 22 - Негабаритный груз - Скульптура

INSERT INTO Cargo (

cargo\_name,  
cargo\_description,  
receipt\_date,  
departure\_date,  
cargo\_weight,  
cargo\_width,  
cargo\_height,  
cargo\_depth,  
cargo\_type

)

VALUES (

'Скульптура',

'Скульптура известного итальянского скульптора. Идеально подойдет  
для украшения сада или интерьера.',

```

    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    10.0,
    0.5,
    1.0,
    0.5,
    CargoType('негабаритный')
);

```

-- Груз 23 - Жидкий груз - Виноградное масло

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Виноградное масло',
    'Бутылка виноградного масла известного итальянского бренда.
Идеально подойдет для приготовления вкусных блюд и ухода за
кожей.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.2,
    0.3,
    0.1,
    CargoType('жидкий')
);

```

-- Груз 24 - Сыпучий груз - Кофе в зернах

```

INSERT INTO Cargo (
    cargo_name,
    cargo_description,

```

```

receipt_date,
departure_date,
cargo_weight,
cargo_width,
cargo_height,
cargo_depth,
cargo_type
)
VALUES (
  'Кофе в зернах',
  'Упаковку кофе в зернах известного итальянского бренда. Идеально
подойдет для приготовления ароматного напитка.',
  TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
  TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
  2.0,
  0.3,
  0.2,
  0.2,
  CargoType('сыпучий')
);

```

-- Груз 25 - Негабаритный груз - Музыкальный инструмент

```

INSERT INTO Cargo (
  cargo_name,
  cargo_description,
  receipt_date,
  departure_date,
  cargo_weight,
  cargo_width,
  cargo_height,
  cargo_depth,
  cargo_type
)
VALUES (
  'Музыкальный инструмент',
  'Гитара известного итальянского бренда. Идеально подойдет для
любителей музыки.',
  TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
  TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),

```

```

5.0,
1.0,
0.5,
0.2,
CargoType('негабаритный')
);

-- Груз 26 - Жидкий груз - Оливковое масло с трюфелем
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,
    cargo_depth,
    cargo_type
)
VALUES (
    'Оливковое масло с трюфелем',
    'Бутылка оливкового масла с трюфелем известного итальянского
бренда. Идеально подойдет для приготовления вкусных блюд.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
HH24:MI:SS'),
    1.0,
    0.2,
    0.3,
    0.1,
    CargoType('жидкий')
);

-- Груз 27 - Сыпучий груз - Ризотто
INSERT INTO Cargo (
    cargo_name,
    cargo_description,
    receipt_date,
    departure_date,
    cargo_weight,
    cargo_width,
    cargo_height,

```

```

    cargo_depth,
    cargo_type
)
VALUES (
    'Ризотто',
    'Упаковку ризотто известного итальянского бренда. Идеально
    подойдет для приготовления вкусного блюда.',
    TO_TIMESTAMP('2024-04-22 10:00:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    TO_TIMESTAMP('2024-04-22 17:45:00', 'YYYY-MM-DD
    HH24:MI:SS'),
    2.0,
    0.3,
    0.3,
    0.1,
    CargoType('сыпучий')
);

```

-- Заполнение таблицы КАРТА ГРУЗОВ

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 1, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 2, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 3, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 4, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 5, 21, 'ДА');

```

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 6, 22, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 7, 22, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 8, 22, 'ДА');

```

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 9, 23, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 10, 23, 'ДА');

```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 11, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 12, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 13, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 14, 21, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 15, 21, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 16, 22, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 17, 22, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 18, 22, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 19, 23, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 20, 23, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 21, 24, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 22, 24, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 23, 24, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 24, 25, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 25, 25, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 26, 25, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 27, 25, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 28, 26, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
```



```
VALUES ( 29, 26, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 30, 26, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 31, 24, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 32, 24, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 33, 24, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 34, 25, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 35, 25, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 36, 25, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 37, 25, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 38, 26, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 39, 26, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 40, 26, 'ДА');
```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 41, 27, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 42, 27, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 43, 27, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 44, 27, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 45, 27, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 46, 27, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 47, 27, 'ДА');
```

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 48, 28, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 49, 28, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 50, 28, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 51, 28, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 52, 28, 'ДА');

```

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 53, 29, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 54, 29, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 55, 29, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 56, 29, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 57, 29, 'ДА');

```

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 58, 30, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 59, 30, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 60, 30, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 61, 30, 'ДА');

```

```

INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 62, 31, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 63, 31, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 64, 31, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 65, 31, 'ДА');
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 66, 31, 'ДА');

```

```
INSERT INTO Cargo_card (cargo_id, tour_card_id, customs_documents)
VALUES ( 67, 31, 'ДА');
```

-- ЗАПРОСЫ К БАЗЕ -----

-- 1. Сформировать список туристов для таможи в целом и по указанной категории.

-- Сформировать таблицу "Список туристов для таможи"

```
SELECT t.tourist_first_name, t.tourist_middle_name, t.tourist_last_name,
       t.tourist_residence_city, t.tourist_age,
       t.phone_number, t.tourist_type.to_string(), t.visa_issued,
       p.birthday, p.pass_series, p.pass_number, p.issued_by, p.when_issued,
       p.sex
FROM Tourist t
JOIN Passport_data p ON t.tourist_id = p.tourist_id;
```

-- Сформировать таблицу "Список туристов по категории турист-грузоперевозчик"

```
SELECT t.tourist_first_name, t.tourist_middle_name, t.tourist_last_name,
       t.tourist_residence_city, t.tourist_age, t.phone_number, t.visa_issued,
       p.birthday, p.pass_series, p.pass_number, p.issued_by, p.when_issued,
       p.sex
FROM Tourist t
JOIN Passport_data p ON t.tourist_id = p.tourist_id
WHERE t.tourist_type = TouristType('турист-грузоперевозчик');
```

-- Сформировать таблицу "Список туристов по категории турист-отдыхающий"

```
SELECT t.tourist_first_name, t.tourist_middle_name, t.tourist_last_name,
       t.tourist_residence_city, t.tourist_age, t.phone_number, t.visa_issued,
       p.birthday, p.pass_series, p.pass_number, p.issued_by, p.when_issued,
       p.sex
FROM Tourist t
JOIN Passport_data p ON t.tourist_id = p.tourist_id
WHERE t.tourist_type = TouristType('турист-отдыхающий');
```

-- 2. Сформировать списки на расселение по указанным гостиницам в целом и указанной категории.

```

-- В целом:
WITH Children_count AS (
    SELECT t.tourist_id, COUNT(*) AS children_number
    FROM Tourist_Children tc
    RIGHT OUTER JOIN Tourist t ON tc.tourist_id = t.tourist_id
    GROUP BY t.tourist_id
)

SELECT
    t.tourist_first_name, t.tourist_middle_name, t.tourist_last_name,
    cc.children_number,
    h.hotel_country, h.hotel_city, h.hotel_name, h.hotel_address,
    tfr.type_room, hc.main_living_cost
FROM Tourist t
JOIN Children_count cc ON t.tourist_id = cc.tourist_id
JOIN Tour_card trc ON t.tourist_id = trc.tourist_id
JOIN Hotel_card hc ON trc.hotel_card_id = hc.hotel_card_id
JOIN Hotel h ON hc.booked_hotel_id = h.hotel_id
JOIN Type_of_room tfr ON hc.reserved_room_id = tfr.type_id
ORDER BY h.hotel_country, h.hotel_city, h.hotel_name;

-- По категории "турист-отдыхающий":
WITH Children_count AS (
    SELECT t.tourist_id, COUNT(*) AS children_number
    FROM Tourist_Children tc
    RIGHT OUTER JOIN Tourist t ON tc.tourist_id = t.tourist_id
    GROUP BY t.tourist_id
)

SELECT
    t.tourist_first_name, t.tourist_middle_name, t.tourist_last_name,
    cc.children_number,
    h.hotel_country, h.hotel_city, h.hotel_name, h.hotel_address,
    tfr.type_room, hc.main_living_cost
FROM Tourist t
JOIN Children_count cc ON t.tourist_id = cc.tourist_id
JOIN Tour_card trc ON t.tourist_id = trc.tourist_id
JOIN Hotel_card hc ON trc.hotel_card_id = hc.hotel_card_id
JOIN Hotel h ON hc.booked_hotel_id = h.hotel_id
JOIN Type_of_room tfr ON hc.reserved_room_id = tfr.type_id
WHERE t.tourist_type.to_string() = 'турист-отдыхающий'

```

```
ORDER BY h.hotel_country, h.hotel_city, h.hotel_name;
```

```
-- По категории "Турист-грузоперевозчик":
```

```
WITH Children_count AS (  
    SELECT t.tourist_id, COUNT(*) AS children_number  
    FROM Tourist_Children tc  
    RIGHT OUTER JOIN Tourist t ON tc.tourist_id = t.tourist_id  
    GROUP BY t.tourist_id  
)
```

```
SELECT  
    t.tourist_first_name, t.tourist_middle_name, t.tourist_last_name,  
    cc.children_number,  
    h.hotel_country, h.hotel_city, h.hotel_name, h.hotel_address,  
    tfr.type_room, hc.main_living_cost  
FROM Tourist t  
JOIN Children_count cc ON t.tourist_id = cc.tourist_id  
JOIN Tour_card trc ON t.tourist_id = trc.tourist_id  
JOIN Hotel_card hc ON trc.hotel_card_id = hc.hotel_card_id  
JOIN Hotel h ON hc.booked_hotel_id = h.hotel_id  
JOIN Type_of_room tfr ON hc.reserved_room_id = tfr.type_id  
WHERE t.tourist_type.to_string() = 'турист-грузоперевозчик'  
ORDER BY h.hotel_country, h.hotel_city, h.hotel_name;
```

```
-- 3. Получить количество туристов, побывавших в стране за  
определенный период в целом и по определенной категории.
```

```
-- В целом:
```

```
SELECT t.tour_country, COUNT(*) as tourist_count  
FROM Tour_card tc  
JOIN Tour t ON tc.tour_id = t.tour_id  
WHERE t.tour_start_date BETWEEN TO_DATE('2024-01-01', 'YYYY-  
MM-DD') and TO_DATE('2024-12-31', 'YYYY-MM-DD')  
GROUP BY t.tour_country;
```

```
-- По категории "турист-отдыхающий":
```

```
SELECT t.tour_country, COUNT(*) as tourist_count  
FROM Tour_card tc  
JOIN Tour t ON tc.tour_id = t.tour_id  
JOIN Tourist tt ON tc.tourist_id = tt.tourist_id
```

```

WHERE tt.tourist_type.to_string() = 'турист-отдыхающий' and
t.tour_start_date BETWEEN TO_DATE('2024-01-01', 'YYYY-MM-DD')
and TO_DATE('2024-12-31', 'YYYY-MM-DD')
GROUP BY t.tour_country;

```

```

-- По категории "турист-грузоперевозчик":
SELECT t.tour_country, COUNT(*) as tourist_count
FROM Tour_card tc
JOIN Tour t ON tc.tour_id = t.tour_id
JOIN Tourist tt ON tc.tourist_id = tt.tourist_id
WHERE tt.tourist_type.to_string() = 'турист-грузоперевозчик' and
t.tour_start_date BETWEEN TO_DATE('2024-01-01', 'YYYY-MM-DD')
and TO_DATE('2024-12-31', 'YYYY-MM-DD')
GROUP BY t.tour_country;

```

-- 4. Получить сведения о конкретном туристе: сколько раз был в стране, даты прилета/отлета, в каких гостиницах останавливался, какие экскурсии и в каких агентствах заказывал, какой груз сдавал.

```

WITH TouristInfo AS (
  SELECT
    t.tourist_id,
    t.tourist_first_name,
    t.tourist_middle_name,
    t.tourist_last_name,

    COUNT(DISTINCT tc.tour_id) AS total_tours,

    LISTAGG(tt.tour_name, ', ')
    WITHIN GROUP (ORDER BY tt.tour_name) AS tour_names,

    LISTAGG(tt.tour_start_date || ' - ' || tt.tour_end_date, ', ')
    WITHIN GROUP (ORDER BY tt.tour_start_date) AS tour_dates,

    LISTAGG(h.hotel_name, ', ')
    WITHIN GROUP (ORDER BY h.hotel_name) AS hotel_names,

    LISTAGG(e.excursion_name || ' (' || e.excursion_agency_name || ')', ', ')
    WITHIN GROUP (ORDER BY e.excursion_name) AS
excursion_names,

    LISTAGG(c.cargo_name, ', ')
    WITHIN GROUP (ORDER BY c.cargo_name) AS cargo_names

```

```

FROM
    Tourist t
LEFT JOIN
    Tour_card tc ON t.tourist_id = tc.tourist_id
LEFT JOIN
    Tour tt ON tc.tour_id = tt.tour_id
LEFT JOIN
    Hotel_card hc ON tc.hotel_card_id = hc.hotel_card_id
LEFT JOIN
    Hotel h ON hc.booked_hotel_id = h.hotel_id
LEFT JOIN
    Excursion_card ec ON tc.tour_card_id = ec.tour_card_id
LEFT JOIN
    Excursion e ON ec.excursion_id = e.excursion_id
LEFT JOIN
    Cargo_card cc ON tc.tour_card_id = cc.tour_card_id
LEFT JOIN
    Cargo c ON cc.cargo_id = c.cargo_id
WHERE
    t.tourist_id = 41 -- Айди конкретного туриста
GROUP BY
    t.tourist_id, t.tourist_first_name, t.tourist_middle_name,
t.tourist_last_name
)
SELECT
    ti.tourist_id,
    ti.tourist_first_name,
    ti.tourist_middle_name,
    ti.tourist_last_name,
    ti.total_tours,
    REGEXP_REPLACE(ti.tour_names, '^(,)([,]+)(,2)+($,)', '\1\4') AS
tour_names,
    REGEXP_REPLACE(ti.tour_dates, '^(,)([,]+)(,2)+($,)', '\1\4') AS
tour_dates,
    REGEXP_REPLACE(ti.hotel_names, '^(,)([,]+)(,2)+($,)', '\1\4') AS
hotel_names,
    REGEXP_REPLACE(ti.excursion_names, '^(,)([,]+)(,2)+($,)', '\1\4')
AS excursion_names,
    REGEXP_REPLACE(ti.cargo_names, '^(,)([,]+)(,2)+($,)', '\1\4') AS
cargo_names
FROM
    TouristInfo ti;

```



-- 5. Получить список гостиниц, в которых производится расселение туристов, с указанием количества занимаемых номеров и проживавших в них человек за определенный период.

```
SELECT h.hotel_name, h.hotel_country, h.hotel_city, tor.type_room,
COUNT(*) AS count_occupied_rooms
FROM Hotel_card hc
JOIN Hotel h ON hc.booked_hotel_id = h.hotel_id
JOIN Type_of_room tor ON hc.reserved_room_id = tor.type_id
WHERE hc.check_in_date BETWEEN TO_DATE('2024-01-01','YYYY-MM-DD') and TO_DATE('2024-12-31','YYYY-MM-DD') -- За 24 год
GROUP BY h.hotel_name, h.hotel_country, h.hotel_city, tor.type_room
ORDER BY h.hotel_name, h.hotel_country, h.hotel_city;
```

-- 6. Получить общее количество туристов, заказавших экскурсии за определенный период.

```
WITH Excursion_card_count AS(
SELECT COUNT(ec.tour_card_id)
FROM Excursion_card ec
JOIN Excursion e ON ec.excursion_id = e.excursion_id
WHERE e.excursion_start BETWEEN TO_DATE('2024-01-01','YYYY-MM-DD') and TO_DATE('2024-12-31','YYYY-MM-DD') -- За 24 год
GROUP BY tour_card_id
)
SELECT COUNT(*) AS Tourist_booked_excursoin_count
FROM Excursion_card_count;
```

-- 7. Выбрать самые популярные экскурсии и самые качественные экскурсионные агентства.

```
SELECT COUNT(*) AS popularity, e.excursion_agency_name,
e.excursion_name, e.excursion_description
FROM Excursion_card ec
JOIN Excursion e ON ec.excursion_id = e.excursion_id
WHERE e.excursion_start BETWEEN TO_DATE('2024-01-01','YYYY-MM-DD') and TO_DATE('2024-12-31','YYYY-MM-DD') -- За 24 год
GROUP BY e.excursion_agency_name, e.excursion_name,
e.excursion_description
HAVING COUNT(*) > 2 -- отбрасываем самые не популярные
ORDER BY popularity DESC;
```

-- 8. Получить данные о загрузке указанного рейса самолета на определенную дату: количество мест, вес груза, объемный вес.

```
SELECT *
FROM Flight f
WHERE f.flight_id = 16 and f.departure_datetime = TIMESTAMP '2024-01-17 14:00:00';
```

-- 9. Получить статистику о грузообороте склада: количество мест и вес груза, сданного за определенный период, количество самолетов, вывозивших этот груз, сколько из них грузовых, а сколько грузопассажирских.

```
WITH
Flight_cargo_people AS (
    SELECT
        tc.tour_card_id,
        COUNT(*) AS flight_cargo_people_count,
        f.departure_datetime,
        SUM(f.ocupated_places) AS ocupated_places_sum,
        SUM(f.ocupated_baggage_weight) AS ocupated_baggage_weight_sum,
        SUM(f.ocupated_baggage_space) AS ocupated_baggage_space_sum
    FROM Tour_card tc
    JOIN Flight_card fc ON tc.flights_card_id = fc.flights_card_id
    JOIN Flight f ON fc.flight_there_id = f.flight_id or fc.flight_back_id = f.flight_id
    WHERE f.plane_type.to_string() = 'грузо-пассажирский'
    GROUP BY tc.tour_card_id, f.departure_datetime
),
Flight_cargo AS (
    SELECT
        tc.tour_card_id,
        COUNT(*) AS flight_cargo_count,
        f.departure_datetime,
        SUM(f.ocupated_places) AS ocupated_places_sum,
        SUM(f.ocupated_baggage_weight) AS ocupated_baggage_weight_sum,
        SUM(f.ocupated_baggage_space) AS ocupated_baggage_space_sum
    FROM Tour_card tc
    JOIN Flight_card fc ON tc.flights_card_id = fc.flights_card_id
    JOIN Flight f ON fc.flight_there_id = f.flight_id or fc.flight_back_id = f.flight_id
    WHERE f.plane_type.to_string() = 'грузовой'
    GROUP BY tc.tour_card_id, f.departure_datetime
)
```

```

SELECT
COALESCE(SUM(fcp.flight_cargo_people_count), 0) AS
flight_cargo_people_count,
COALESCE(SUM(fc.flight_cargo_count), 0) AS flight_cargo_count,
COALESCE(SUM(fcp.ocupated_places_sum), 0) AS ocupated_places_sum,
COALESCE(SUM(fcp.ocupated_baggage_weight_sum), 0) AS
ocupated_baggage_weight_sum,
COALESCE(SUM(fcp.ocupated_baggage_space_sum), 0) AS
ocupated_baggage_space_sum
FROM Flight_cargo_people fcp
FULL JOIN Flight_cargo fc ON fcp.tour_card_id = fc.tour_card_id
WHERE fcp.departure_datetime BETWEEN TIMESTAMP '2024-01-01
00:00:00' and TIMESTAMP '2024-12-31 00:00:00'; -- За 24 год

```

-- 10. Получить полный финансовый отчет по указанной группе в целом и для определенной категории туристов.

-- В целом:

```

SELECT
tg.group_name,
SUM(tc.main_tour_cost) AS main_tour_cost_group,
SUM(tc.main_tour_expenses) AS main_tour_expenses_group
FROM Tourists_group tg
JOIN Tour_card tc ON tg.tour_id = tc.tour_id
WHERE group_name = 'Шопинг в Модной Столице - 2024-04-15' --
указанная группа
--WHERE group_name = 'Зимний Горнолыжный Рай - 2024-01-10' --Ещё
вариант группы
--WHERE group_name = 'Тропический Рай - 2024-07-01' --Ещё вариант
группы
GROUP BY tg.group_name;

```

-- По категории "турист-отдыхающий" в группе: 'Тропический Рай - 2024-07-01'

```

SELECT
tg.group_name,
SUM(tc.main_tour_cost) AS main_tour_cost_group,
SUM(tc.main_tour_expenses) AS main_tour_expenses_group
FROM Tourists_group tg
JOIN Tour_card tc ON tg.tour_id = tc.tour_id
JOIN Tourist t ON tc.tourist_id = t.tourist_id

```

```

WHERE group_name = 'Тропический Рай - 2024-07-01' --указанная
группа
and t.tourist_type.to_string() = 'турист-отдыхающий' --указанная
категория
GROUP BY tg.group_name;

```

```

-- По категории "турист-грузоперевозчик" в группе: 'Шопинг в Модной
Столице - 2024-04-15'
SELECT
    tg.group_name,
    SUM(tc.main_tour_cost) AS main_tour_cost_group,
    SUM(tc.main_tour_expenses) AS main_tour_expenses_group
FROM Tourists_group tg
JOIN Tour_card tc ON tg.tour_id = tc.tour_id
JOIN Tourist t ON tc.tourist_id = t.tourist_id
WHERE group_name = 'Шопинг в Модной Столице - 2024-04-15' --
указанная группа
and t.tourist_type.to_string() = 'турист-грузоперевозчик' --указанная
категория
GROUP BY tg.group_name;

```

-- 11. Получить данные о расходах и доходах за определенный период:  
обслуживание самолета, гостиница, экскурсии, визы, расходы  
представительства и т.п.

```

SELECT
    SUM(hc.main_living_cost) AS main_living_cost,
    SUM(hc.main_living_expenses) AS main_living_expenses,
    SUM(fc.main_flight_cost) AS main_flight_cost,
    SUM(fc.main_maintenance_expenses) AS main_maintenance_expenses,
    SUM(tc.main_excursions_cost) AS main_excursions_cost,
    SUM(tc.main_excursions_expenses) AS main_excursions_expenses,
    SUM(tc.cargo_traffic_cost) AS cargo_traffic_cost,
    SUM(tc.cargo_traffic_expenses) AS cargo_traffic_expenses,
    SUM(tc.main_tour_cost) AS main_tour_cost,
    SUM(tc.main_tour_expenses) AS main_tour_expenses
FROM Tour_card tc
JOIN Flight_card fc ON tc.flights_card_id = fc.flights_card_id
JOIN Hotel_card hc ON tc.hotel_card_id = hc.hotel_card_id
WHERE check_in_date BETWEEN TIMESTAMP '2024-01-01 00:00:00'
and TIMESTAMP '2024-12-31 00:00:00'; -- За 24 год

```

-- 12. Получить статистику по видам отправляемого груза и удельную долю каждого вида в общем грузопотоке.

```
WITH CargoTypeStats AS (  
  SELECT  
    CargoType.to_string(cargo_type) AS string_cargo_type,  
    COUNT(*) AS cargo_count,  
    SUM(cargo_weight) AS total_weight,  
    SUM(cargo_width * cargo_height * cargo_depth) AS total_volume  
  FROM Cargo  
  GROUP BY CargoType.to_string(cargo_type)  
)  
SELECT  
  cts.string_cargo_type,  
  cts.cargo_count,  
  cts.total_weight,  
  cts.total_volume,  
  TO_CHAR(ROUND(  
    (cts.cargo_count / (SELECT SUM(cargo_count) FROM CargoTypeStats))  
    * 100  
  )) || '%' AS share_of_total_cargo  
FROM CargoTypeStats cts  
ORDER BY cts.cargo_count DESC;
```

-- 13. Вычислить рентабельность представительства (соотношение доходов и расходов).

```
SELECT  
  SUM(main_tour_cost) AS total_revenue,  
  SUM(main_tour_expenses) AS total_expenses,  
  (SUM(main_tour_cost) - SUM(main_tour_expenses)) AS profit,  
  'x' || TO_CHAR(ROUND(  
    (SUM(main_tour_cost) / SUM(main_tour_expenses))  
  )) AS profitability  
FROM Tour_card;
```

-- 14. Определить процентное отношение отдыхающих туристов к туристам shor-туров в целом и за указанный период (например, в зависимости от времени года).

--В целом:

```
SELECT  
  TouristType.to_string(tourist_type) AS string_tourist_type,  
  COUNT(*) AS tourist_count,
```

```

    TO_CHAR(ROUND(COUNT(*) / (SELECT COUNT(*) FROM Tourist) *
100)) || '%' AS Percent_ratio
FROM Tourist
GROUP BY TouristType.to_string(tourist_type);

```

--За период Весна-Лето 2024

```

WITH TouristInfo AS(
    SELECT
        t.tourist_type.to_string() AS string_tourist_type
    FROM Tourist t
    JOIN Tourists_group tg ON t.tourist_id = tg.tourist_id
    JOIN Tour tt ON tg.tour_id = tt.tour_id
    WHERE tt.tour_start_date BETWEEN DATE '2024-03-01' and DATE
'2024-09-01'
)
SELECT
    string_tourist_type,
    COUNT(*) AS tourist_count,
    TO_CHAR(ROUND(COUNT(*) / (SELECT COUNT(*) FROM
TouristInfo) * 100)) || '%' AS Percent_ratio
FROM TouristInfo
GROUP BY string_tourist_type;

```

-- 15. Получить сведения о туристах указанного рейса: список группы, гостиницы, груз, бирки, маркировка.

```

WITH TourCard_Flight_info AS (
    SELECT tc.tourist_id, f.flight_id, f.departure_datetime
    FROM Tour_card tc
    JOIN Flight_card fc ON tc.flights_card_id = fc.flights_card_id
    JOIN Flight f ON fc.flight_there_id = f.flight_id or fc.flight_back_id =
f.flight_id
)
SELECT *
FROM Tourist t
JOIN Tour_card tc ON t.tourist_id = tc.tourist_id
JOIN Cargo_card cc ON cc.tour_card_id = tc.tour_card_id
JOIN Cargo c ON cc.cargo_id = c.cargo_id
JOIN Hotel_card hc ON tc.hotel_card_id = hc.hotel_card_id
JOIN Hotel h ON hc.booked_hotel_id = h.hotel_id
WHERE t.tourist_id = (
    SELECT tourist_id

```

```
FROM TourCard_Flight_info
WHERE flight_id = 16 and departure_datetime = TIMESTAMP '2024-01-
17 14:00:00'
-- Так получается, что я так заполнил базу, что у меня все туристы на
разных рейсах,
--даже если они прилетают на одни тур в одной группе, вылетают они
из разных городов,
--поэтому информация будет выведена об одном туристе, какой бы
рейс не выбрали,
--но суть логики это не меняет.
);
```