

Среднесрочные системы управления базами данных

- Создание подробного отчета со скриншотами кода и результатами задания
- Загрузка отчета в <https://forms.gle/3NdbGbUwchLQRNbV9>

<https://sqliteonline.com> – онлайн компилятор если у вас не работает pgadmin4
dump файл скачайте с github для задач 5-9.

Вариант 1: Управление аэропортами и пассажирами

Задача 1а: Создать таблицу для аэропортов (только код напишите)

Цель: Разработать таблицу для хранения информации об аэропортах с соответствующими типами данных и ограничениями.

Столбцы: airport_id (целое число), airport_name (строка), страна (строка), штат (строка), город (строка), created_at (временная метка), updated_at (временная метка).

Требования: Используйте соответствующие типы данных, задайте airport_id в качестве первичного ключа и обеспечьте автоматическое управление временными метками для создания и обновления.

Задача 2а: Вставьте данные в таблицу Passengers(только код напишите)

Цель: Заполните таблицу passengers различными записями, чтобы отразить разнообразие пассажиров.

Укажите имя, фамилию, пол, дату рождения, гражданство, место жительства и номер паспорта.

Задача: Вставьте как минимум 2 записи с разными данными, чтобы обеспечить учет разных национальностей и возрастных групп.

Задача 3а: Обновить информацию о пассажирах в таблице пассажиров(только код напишите)

Цель: Изменить существующие записи, чтобы отразить изменения, такие как изменение имени или уточнение места жительства.

Сценарий: Обновить фамилию и страну проживания для конкретных пассажиров, идентифицированных по номеру паспорта.

Задача 4а: Удалить пассажира из таблицы пассажиров(только код напишите)

Цель: Удалить записи из базы данных на основе определенных условий.

Условие: Удалите пассажиров, которые не обновляли свою информацию в течение последних 5 лет.

Требование: Используйте условный оператор, основанный на временной метке `updated_at`.

Задача 5а: Получить информацию обо всех пассажирах с соответствующей информацией о багаже (используя файл дампа)

Цель: Использовать ВНУТРЕННЕЕ объединение для отображения объединенных данных из таблиц "Пассажиры" и "багаж".

Подробности: Отобразить все столбцы из обеих таблиц, в которых есть записи.

Задача 6а: Составить список всех пассажиров и любого багажа, который они могли зарегистрировать (используя файл дампа)

Цель: Реализовать объединение СЛЕВА, чтобы отобразить всех пассажиров и их багаж, включая тех, у кого нет багажа.

Результат: Включить нулевые значения, с которыми не связан багаж.

Задача 7а: Составить список всего багажа и связанных с ним пассажиров (используя файл дампа)

Цель: Использовать ПРАВОЕ соединение для отображения всего багажа и привязки каждого из них к пассажиру, если это возможно.

Задача: Отобразить данные о пассажирах для каждого места багажа, даже если данные о пассажирах не связаны.

Задача 8а: Объединить информацию о пассажирах с их багажом, отобразив все записи (используя файл дампа)

Цель: Использовать ПОЛНОЕ ВНЕШНЕЕ объединение для создания полного списка всех записей о пассажирах и багаже.

Сложность: Обрабатывать случаи, когда есть пассажиры без багажа и багаж без зарегистрированных пассажиров.

Задача 9а: Создать индекс по стране проживания в таблице пассажиров (используя файл дампа)

Цель: Повысить производительность запросов для поиска по стране проживания.

Требование: Создать индекс по столбцу `country_of_residence` для ускорения поиска.а

Вариант 2: Управление полетами и авиакомпаниями

Задача 1b: Создать таблицу для рейсов(только код напишите)

Цель: Создать таблицу для управления информацией о рейсах с необходимыми полями и взаимосвязями.

Столбцы: Содержат идентификаторы рейсов, выходов на посадку, авиакомпаний, аэропортов и временные метки для запланированного и фактического вылета/прибытия.

Требования: Убедитесь, что внешние ключи правильно связаны с авиакомпаниями и аэропортами.

Задача 2b: Добавьте запись о рейсе в таблицу рейсов(только код напишите)

Цель: Вставьте данные, представляющие новый рейс, включая все необходимые детали.

Элементы данных: идентификатор рейса, пункты вылета и прибытия, привязанная авиакомпания и временные метки.

Задача: Убедиться, что данные соответствуют ограничениям, установленным внешними ключами и расписанием.

Задача 3b: Изменить время отправления рейса по расписанию(только код напишите)

Цель: Обновить время отправления для указанного рейса.

Сценарий: Измените запланированное время вылета в зависимости от оперативных требований или задержек.

Задача 4b: Удалите рейс из таблицы рейсов(только код напишите)

Цель: Удалите запись о рейсе на основе определенных оперативных критериев, таких как отмена.

Условие: Используйте условие, основанное на статусе рейса или датах вылета.

Задача 5b: Перечислить все рейсы вместе с названиями авиакомпаний (используя файл дампа)

Цель: Использовать ВНУТРЕННЕЕ объединение для сопоставления рейсов с соответствующими авиакомпаниями.

Подробности: Отобразить информацию о рейсах рядом с названиями авиакомпаний.

Задача 6b: Отобразить все рейсы, в том числе те, на которых не указана авиакомпания (используя файл дампа)

Цель: Реализовать объединение СЛЕВА для составления списка всех рейсов, гарантируя, что будут включены рейсы без привязки к авиакомпании.

Результат: Отобразить рейсы с нулевыми значениями для авиакомпаний, где это применимо.

Задача 7b: Отобразить все авиакомпании и их рейсы (используя файл дампа)

Цель: Использовать соединение СПРАВА, чтобы отобразить все авиакомпании и связанные с ними рейсы.

Задача: Убедиться, что в списке указаны все авиакомпании, даже те, у которых нет текущих рейсов.

Задача 8b: Объединить сведения о рейсах с авиакомпаниями, отобразив все записи (используя дам-файл)

Задача: Использовать ПОЛНОЕ ВНЕШНЕЕ объединение для отображения полного набора записей как по рейсам, так и по авиакомпаниям.

Сложность: Обработать несоответствия, при которых на рейсах может отсутствовать авиакомпания, и наоборот.

Задача 9b: Проиндексировать пункты отправления и прибытия в таблице рейсов (используя файл дампа).

Цель: Повысить производительность запросов для операций, связанных с информацией о выходе на посадку.

Требование: Создать индексы для столбцов `departing_gate` и `arriving_gate`.

Вариант 3: Управление багажом и безопасностью

Задача 1с: Определить таблицу для багажа

Задача: Создать таблицу для хранения сведений о багаже, связанных с бронированиями и пассажирами.

Столбцы: Идентификатор багажа, вес, идентификатор бронирования и временные метки.

Требования: Включить внешние ключи для привязки багажа к бронированиям.

Задача 2с: Ввести информацию о багаже в таблицу багажа

Цель: Заполнить таблицу сведениями о багаже, зарегистрированном на различных рейсах.

Основные данные: Указать вес багажа и соответствующий идентификатор бронирования.

Задача: Убедиться, что все записи действительны и соответствуют правилам провоза багажа авиакомпаниями.

Задача 3с: Обновить вес багажа

Цель: Изменить записи, чтобы скорректировать заявленный вес багажа, который может потребоваться после повторной регистрации.

Сценарий: Обновить вес на основе дополнительных предметов, добавленных или изъятых после первоначальной регистрации.

Задача 4с: Удалить информацию о багаже из таблицы багажа.

Цель: Удалить записи о багаже, который был изъят или не был погружен на рейс.

Условие: Укажите такие условия, как наличие багажа, не связанного с текущим бронированием.

Задача 5с: Отобразить все проверки безопасности вместе с данными о пассажирах (используя файл дампа).

Цель: Использовать ВНУТРЕННЕЕ объединение, чтобы связать результаты проверки безопасности с соответствующей информацией о пассажирах.

Подробности: Показать полную информацию о безопасности и пассажирах вместе.

Задача 6с: Отобразить всех пассажиров, включая тех, кто не проходил проверку безопасности (используя файл дампа).

Задача: Реализовать объединение по ЛЕВОМУ краю, чтобы отобразить список всех пассажиров, а также тех, у кого нет записи о проверке безопасности.

Результат: Включить нулевые значения для сведений о проверке безопасности, где проверки не проводились.

Задача 7с: Отобразить все проверки безопасности и соответствующую информацию о пассажирах (используя файл дампа).

Цель: Используйте ПРАВИЛЬНОЕ объединение, чтобы убедиться, что все проверки безопасности перечислены вместе со связанными данными о пассажирах.

Задача: Отобразить все записи о проверке безопасности, заполнив данные о пассажирах, где это возможно.

Задача 8с: Связать проверки безопасности с пассажирами, отобразив все результаты (используя файл дампа).

Задача: Использовать ПОЛНОЕ ВНЕШНЕЕ объединение, чтобы отобразить полный список, включающий все проверки безопасности и записи о пассажирах.

Сложность: Обрабатывать случаи, когда могут быть проверки безопасности или пассажиры, перечисленные без соответствующих совпадений.

Задача 9с: Настроить индекс веса багажа (используя файл дампа).

Цель: Повысить эффективность поиска и выполнения запросов, связанных с весом багажа.

Требование: Создать индекс для столбца weight_in_kg.

