

Проектирование таблицы People.....	2
Тестовые сценарии добавления данных.....	3
Тестовые сценарии выборки данных.....	4
Тест-кейсы для изменения данных.....	5
Тест кейсы для удаления данных.....	6
Проектирование таблицы People2.....	7
Тестовые сценарии для таблицы People2.....	8
Нагрузочное тестирование.....	9

# Проектирование таблицы People

Название столбца	Тип	Свойства	Описание
id	SERIAL	PRIMARY KEY	Необходим для однозначной идентификации записей.
Name	VARCHAR(100)	NOT NULL	Хранение ФИО пользователя системы.
DateOfBirth	DATE	NOT NULL	Хранение даты рождения пользователя системы.

# Тестовые сценарии добавления данных

## Позитивные сценарии:

1. Добавление одной записи в пустую таблицу
2. Добавление нескольких записей в пустую таблицу
3. Добавление одной записи в непустую таблицу
4. Добавление нескольких записей в непустую таблицу
5. Добавление записи с одинаковыми значениями полей Name, DateOfBirth
6. Добавление набора записей с одинаковыми значениями полей Name, DateOfBirth в одном запросе
7. Добавление записи с количеством символов в поле Name равной 100

## Негативные сценарии:

1. Добавление записи с отсутствующим значением поля DateOfBirth
2. Добавление записи с отсутствующим значением поля Name
3. Добавление записи с значением будущей даты в поле DateOfBirth
4. Добавление записи со значением Name длиной более 100

# Тестовые сценарии выборки данных

## Позитивные сценарии:

1. Подсчет количества записей при отсутствии записей в таблице
2. Подсчет количества записей при наличии одной записи в таблице
3. Подсчет количества записей при большом количестве записей в таблице
4. Выборка одного элемента с выборкой по id
5. Выборка элементов с применением фильтрации по полю Name
6. Выборка элементов с применением фильтрации по полю Name с использованием LIKE
7. Выборка элементов с применением фильтрации по полю DateOfBirth
8. Выборка элементов с применением фильтрации по полю DateOfBirth с заданием промежутка

## Негативные сценарии:

1. Выборка по отсутствующему значению id
2. Применение некорректного синтаксиса в запросе SELECT
3. Применение некорректного символа в запросе
4. Выборка из отсутствующей таблицы
5. Применение некорректного правила фильтрации при использовании оператора WHERE (несоответствие типов данных)

# Тест-кейсы для изменения данных

## Позитивные:

1. Обновление записи с существующим ID
2. Обновление нескольких записей
3. Обновление всех записей в таблице

## Негативные:

1. Обновление записи с несуществующим ID
2. Обновление с некорректным типом данных
3. Обновление с нарушением уникальности (constraint violation)
4. Обновление с нарушением внешнего ключа (foreign key constraint)
5. Обновление с использованием неправильного имени таблицы или столбца

# Тест кейсы для удаления данных

## Позитивные:

1. Удаление записи по существующему ID
2. Удаление нескольких записей с фильтром
3. Удаление всех записей в таблице
4. Удаление записи с подтверждением наличия внешних ключей (ON DELETE CASCADE)

## Негативные:

1. Удаление записи с несуществующим ID
2. Удаление записи с нарушением внешнего ключа (без ON DELETE CASCADE)
3. Удаление с использованием неправильного имени таблицы
4. Удаление всех записей с ошибкой в условии WHERE
5. Удаление с использованием неправильного типа данных в фильтре

5.

# Проектирование таблицы People2

Название столбца	Тип	Свойства	Описание
id	SERIAL	PRIMARY KEY	Необходим для однозначной идентификации записей
FirstName	VARCHAR(100)	NOT NULL	Имя
FamilyName	VARCHAR(100)	NOT NULL	Фамилия
DateOfBirth	DATE	NOT NULL	Дата рождения
PlaceOfBirth	VARCHAR(100)	NOT NULL	Место рождения
Occupation	VARCHAR(100)		Информация о роде занятий (работе). Носит справочный характер.
Hobby	VARCHAR(100)		Информация об увлечениях пользователей. Носит справочный характер.

# Тестовые сценарии для таблицы People2

Переименование таблицы:

- Корректное переименование таблицы
- Переименование таблицы с ошибкой:
  - Использование названия уже существующей таблицы
  - Использование в новом названии таблицы некорректного символ

Переименование столбцов:

- Корректное переименование одного столбца
- Корректное переименования столбцов
- Переименование столбца с ошибкой:
  - Использование названия уже существующего столбца
  - Использование в новом названии столбца некорректного символ

Добавление столбца:

- Корректное добавление столбца
- Воспроизведение ошибки при добавлении столбца
  - Использование названия уже существующего столбца
  - Использование в новом названии столбца некорректного символа

Изменение столбца:

- Корректное изменение столбца
  - Переименование столбца
  - Изменение типа
- Некорректное изменение столбца
  - Переименование столбца
  - Изменение типа



# Нагрузочное тестирование

Нефункциональные требования к производительности таблицы в базе данных описывают, каким образом система должна работать с точки зрения скорости, масштабируемости, стабильности и отказоустойчивости. Эти требования обычно устанавливают параметры, которым должна соответствовать база данных при выполнении различных операций. Возможные НФТ для таблицы **People2**:

Требования к времени отклика:

- **Вставка записи:** Время вставки одной записи в таблицу **People2** не должно превышать 100 мс при размере таблицы до 1 млн записей.
- **Обновление записи:** Время обновления одной записи не должно превышать 150мс при размере таблицы до 1 млн записей.
- **Выборка записи:** Время выборки одной записи по ключу (например, по id) не должно превышать 50 мс.
- **Выборка множества записей:** Время выборки до 10 тыс. записей по параметрам (например, по PlaceOfBirth или DateOfBirth) не должно превышать 500 мс.

Масштабируемость:

- Система должна поддерживать эффективную работу с таблицей, содержащей до 10 млн записей

Одновременные сеансы:

- Таблица **People2** должна поддерживать до 100 одновременных операций вставки и обновления без значительного увеличения времени выполнения

Использование ресурсов:

- **ЦП:** Использование процессора при выполнении операций с таблицей не должно превышать 70% на сервере при стандартной нагрузке (до 10 млн записей, до 100 одновременных подключений).
- **Память:** Объем оперативной памяти, используемой для операций с таблицей, не должен превышать 50% доступной памяти сервера.
- **Диск:** Использование дискового пространства для хранения данных в таблице должно быть оптимизировано, а рост объема данных не должен приводить к увеличению использования диска более чем на 10% после добавления 1 млн записей.

PostgreSQL предоставляет несколько встроенных утилит и функций, которые помогут отслеживать и контролировать производительность базы данных:

- **pg\_stat\_activity** - Отображает информацию о текущих подключениях, запросах, и их состоянии;
- **pg\_stat\_statements** - Расширение, собирающее статистику по выполненным SQL-запросам, включая количество выполнений, среднее время выполнения и использование ресурсов;
- **pg\_stat\_user\_tables** - Показывает статистику по пользовательским таблицам, включая количество операций вставки, обновления, удаления и выборки;
- **pg\_stat\_bgwriter** - Предоставляет информацию о работе фоновых процессов, которые отвечают за запись данных на диск (background writer);
- **pg\_stat\_database** - Отображает общую статистику по базе данных, включая количество хитов кэша, чтений с диска, количество выполненных транзакций и ошибок;