Да, если ожидается большое количество документов или обновлений, имеет смысл завести базу данных. Это позволит:

1. **Эффективно хранить и структурировать данные**:
   * Организовать документы по категориям, дате, типу, источнику и другим параметрам.
2. **Обеспечить быстрый доступ**:
   * Поиск, фильтрация и сортировка данных станет проще.
3. **Устранить дублирование**:
   * Легко проверять, добавлен ли документ ранее.
4. **Поддерживать масштабируемость**:
   * Возможность обработки большого объема данных без снижения производительности.

**Выбор базы данных**

1. **Для структурированных данных** (документы с предсказуемыми полями, например, название, дата, ссылка):
   * **PostgreSQL** или **MySQL**: Подходят для стандартных операций с таблицами.
2. **Для гибкой структуры данных** (например, текст документа, метаданные):
   * **MongoDB**: Прекрасно подходит для работы с JSON-документами.
3. **Для локального использования или минимальных ресурсов**:
   * **SQLite**: Легковесная база данных для небольших проектов.

**Структура базы данных**

Пример структуры таблиц для PostgreSQL:

**Таблица documents (основная таблица):**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | SERIAL | Уникальный идентификатор |
| title | TEXT | Название документа |
| link | TEXT | Ссылка на документ |
| date\_added | TIMESTAMP | Дата добавления в базу |
| source | TEXT | Источник (например, pravo.gov.ru) |
| type | TEXT | Тип документа (ФЗ, ГОСТ и т.д.) |
| content | TEXT | Текст документа (при необходимости) |

**Таблица categories (для классификации документов):**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | SERIAL | Уникальный идентификатор |
| name | TEXT | Категория (например, "ФЗ", "ГОСТ") |

**Интеграция с Python**

1. **Установка библиотеки для работы с базой данных**:

bash

Копировать код

pip install psycopg2

1. **Пример подключения к PostgreSQL и сохранения данных**:

python

Копировать код

import psycopg2

def save\_to\_database(documents):

# Подключение к базе данных

conn = psycopg2.connect(

dbname="documents\_db",

user="your\_user",

password="your\_password",

host="localhost"

)

cursor = conn.cursor()

# SQL-запрос для вставки данных

for doc in documents:

cursor.execute("""

INSERT INTO documents (title, link, date\_added, source, type)

VALUES (%s, %s, NOW(), %s, %s)

ON CONFLICT (link) DO NOTHING

""", (doc['title'], doc['link'], doc['source'], doc['type']))

# Сохранение изменений и закрытие подключения

conn.commit()

cursor.close()

conn.close()

# Пример использования

documents = [

{"title": "ФЗ №152", "link": "https://pravo.gov.ru/document/12345", "source": "pravo.gov.ru", "type": "ФЗ"}

]

save\_to\_database(documents)

1. **Пример извлечения данных**:

python

Копировать код

def get\_documents():

conn = psycopg2.connect(

dbname="documents\_db",

user="your\_user",

password="your\_password",

host="localhost"

)

cursor = conn.cursor()

cursor.execute("SELECT \* FROM documents ORDER BY date\_added DESC")

results = cursor.fetchall()

cursor.close()

conn.close()

return results

# Выводим данные

for doc in get\_documents():

print(doc)

**Преимущества использования базы данных**

* Возможность добавлять метаданные (например, источник, тип документа).
* Легкость интеграции с веб-интерфейсом или отчетностью.
* Обеспечение целостности данных и удобство работы при увеличении объема.

Если потребуется помощь в проектировании базы данных или настройке, готов помочь!