Для реализации базы данных социальной сети с функциями **отправки сообщений**, **постов** и **реакций на посты** можно создать структуру данных с использованием комбинации реляционной базы данных (PostgreSQL) для хранения информации о пользователях, постах и реакциях, и NoSQL базы данных (MongoDB) для хранения большого объема сообщений между пользователями.

**1. Реляционная база данных (PostgreSQL)**

**Таблица users — хранит информацию о пользователях**

|  |  |
| --- | --- |
| user\_id | PRIMARY KEY INT |
| username | VARCHAR (50) NOT NULL |
| email | VARCHAR (50) NOT NULL |
| password\_hash | NOT NULL |
| created\_at | TIMESTAMP |

* **user\_id**: уникальный идентификатор пользователя.
* **username**: имя пользователя
* **email**: электронная почта пользователя
* **password\_hash**: хешированный пароль.
* **created\_at**: дата и время создания профиля.

**Таблица media\_file — хранит публикации пользователей**

|  |  |
| --- | --- |
| media\_id | PRIMARY KEY INT |
| post\_id | INT NOT NULL |
| file\_url | VARCHAR (255) NOT NULL |
| file\_type | VARCHAR (50) NOT NULL |
| file\_size | BIGINT |
| description | TEXT |
| uploaded\_at | TIMESTAMP |

* **media\_url:** уникальный идентификатор файла.
* **post\_id:** уникальный идентификатор поста.
* **file\_url:** Ссылка на файл в облачном хранилище, где сам файл хранится.
* **file\_type:** Тип файла, такой как "image/jpeg" для изображений, "video/mp4" для видео, "application/pdf" для документов и т.д.
* **file\_size:** Размер файла в байтах. Это полезно для управления хранилищем, поскольку можно контролировать ограничение на размер загружаемых файлов.
* **description:** Описание или подпись к файлу, если это изображение или документ.
* **uploaded\_at:** Дата и время загрузки.

**Таблица posts — хранит публикации пользователей**

|  |  |
| --- | --- |
| post\_id | PRIMARY KEY INT |
| user\_id | INT NOT NULL FOREIGN KEY |
| content | TEXT |
| media\_url | VARCHAR (255) FOREIGN KEY |
| created\_at | TIMESTAMP |

* **post\_id**: уникальный идентификатор поста.
* **user\_id**: идентификатор пользователя, который создал пост.
* **content**: содержание поста.
* **media\_url:** ссылка на файл (документ, изображение и т.д.) в облачном хранилище
* **created\_at**: дата и время создания поста.

**Таблица reactions — хранит реакции пользователей на посты**

|  |  |
| --- | --- |
| reaction\_id | SERIAL PRIMARY KEY |
| post\_id | INT NOT NULL |
| user\_id | INT NOT NULL |
| reaction\_type | VARCHAR (50) |
| reacted\_at | TIMESTAMP |

* **reaction\_id**: уникальный идентификатор реакции.
* **post\_id**: идентификатор поста, к которому относится реакция.
* **user\_id**: идентификатор пользователя, оставившего реакцию.
* **reaction\_type**: тип реакции (например, лайк, сердечко, смех и т.д.).
* **reacted\_at**: дата и время, когда была оставлена реакция.

**2. NoSQL база данных (MongoDB)**

Для хранения сообщений между пользователями, которые могут генерироваться в большом количестве, используем коллекцию MongoDB, так как NoSQL базы данных более производительны и гибки для работы с большими объемами данных.

**Коллекция messages — хранит личные сообщения между пользователями**

|  |  |
| --- | --- |
| \_id | Уникальный идентификатор сообщения |
| sender\_id | Идентификатор пользователя, отправившего сообщение |
| receiver\_id | Идентификатор пользователя, получившего сообщение |
| message\_text | Текст сообщения |
| sent\_at | Дата и время отправки сообщения |
| attachments | Массив вложений (каждый элемент содержит file\_url, file\_type, file\_size, description) |

**Поля в массиве attachments:**

|  |  |
| --- | --- |
| file\_url | URL файла, расположенного в облачном хранилище |
| file\_type | MIME-тип файла (например, image/jpeg для изображений, application/pdf для документов) |
| file\_size | Размер файла в байтах |
| description | Подпись или описание файла, если необходимо |