Отчет о проделанной работе
студента 328 группы

Лебедева Никиты

Спецсеминар «Интернет, распределенные информационные системы и цифровые библиотеки»

Москва, 2019

Содержание

Введение

В двадцать первом веке важным средством общения стали мессенджеры и социальные сети. Они используются для обмена информацией не только подавляющим большинством людей, но и многими коммерческими и некоммерческими организациями. Трудно представить, какой огромный объем данных человечество создает ежеминутно и как часто оно вынуждено обращаться к этим данным. Для хранения и обработки всей этой информации используются базы данных, одну из которых мы и представим в данной работе.

Глоссарий

Канал - общий чат для пользователей, являющихся его участниками. У каждого канала существует владелец, который имеет право добавлять и исключать участников.

Учетная запись - набор уникальных для каждого пользователя данных, позволяющих системе его авторизовать.

Участник канала - пользователь, имеющий возможность чтения и отправки сообщений в канал.

Иметь доступ к каналу - иметь возможность по собственному желанию стать *участником канала.*

Роль - статус пользователя, определяющий действия, которые он может выполнять в канале. Например, добавлять или удалять пользователей.

Вложения - файлы, прикрепленные к сообщению.

Цель работы

Целью работы является создание базы данных, обеспечивающей надежное хранение и быстрый доступ к используемым мессенджером данным.

Специфика приложения

1. Мессенджер используется для обмена сообщениями, которые могут содержать вложения.

2. Приложение имеет открытый исходный код, который может быть изменен перед разверткой на сервере. Чтобы, даже изменив код, администратор сервера не мог получить доступ к пересылаемым данным, все сообщения и вложения должны передаваться и храниться в зашифрованном виде. Шифрование и

дешифрование сообщений должно происходить исключительно в браузере пользователя.

3. Каналы в мессенджере имеют иерархическую структуру: каждый канал может иметь произвольное количество подканалов. Каждый участник отцовского канала *имеет доступ* ко всем его подканалам.

4. Каждый канал имеет собственный ключ шифрования (EncKey), которым, при помощи симметричного алгоритма, шифруются все сообщения в нем.

5. Каждый пользователь имеет свои публичный и приватный (PubKey и PrivKey) ключи шифрования.

6. Копия ключа EncKey каждого канала, в котором состоит пользователь, шифруются его публичным ключом PubKey, после чего отправляется на сервер, откуда в любой момент может быть запрошена пользователем.

7. При приглашении нового участника в канал в браузере приглашающего пользователя происходят следующие действия: Скачивается зашифрованная пользовательская копия EncKey, расшифровывается при помощи пользовательского PrivKey, после чего зашифровывается при помощи PubKey приглашаемого пользователя и отправляется на сервер к другим зашифрованным ключам приглашаемого пользователя.

8. Каждый участник каждого канала имеет (внутри этого канала) роль, определяющую уровень его прав по редактированию канала: изменить имя канала, добавлять новых пользователей и удалять старых.

9. Каждый пользователь имеет логин и пароль. На основании пароля генерируется ключ UnlockKey, которым шифруют пользовательский PrivKey при помощи симметричного алгоритма. После этого зашифрованный PrivKey отправляют храниться на сервер. Позже PrivKey может быть расшифрован пользователем при помощи того же UnlockKey, для генерации которого достаточно снова ввести пароль.

10. При регистрации пользователю предлагается сохранить копию своего PrivKey в надежном месте. В случае утери пароля пользователь может использовать PrivKey для восстановления доступа к учетной записи. В этом случае создается новый пароль, из него - новый UnlockKey, а им шифруется старый PrivKey. Таким образом, доступ к зашифрованным EncKey каналов не теряется.

БД будет использоваться

Для добавления, хранения и получения:

1. данных о каналах
2. данных об учетных записях пользователей
3. сообщений пользователей, а также прикрепленных к сообщениям вложений
4. связей между пользователями и каналами, а также пользовательских настроек этих каналов.
5. публичных ключей шифрования в открытом виде, а также зашифрованных приватных.

Требования к базе данных

1. База должна быть создана при помощи СУБД PostgresQL.
2. Высокая скорость (не более 3 секунд) получения, вставки, модификации и удаления информации в БД.
3. Целостность данных - БД должна содержать полную и непротиворечивую информацию
4. Сокращение избыточности и дублирования данных, кроме тех случаев, когда они позволяют избежать лишних запросов

Задача

1. Составить схему базы данных, удовлетворяющей требованиям и подходящую для указанных сценариев использования.
2. Рассмотреть возможность и необходимость денормализации базы данных.
3. Написать скрипты создания таблиц и индексов.
4. Если решено денормализовать базу данных, то создать триггеры для осуществления корректного удаления, обновления и добавления данных
5. Увеличить производительность БД путем секционирования и создания индексов

Выполнение работы

Выбор алгоритмов шифрования - тема отдельного доклада. Здесь нам достаточно знать, где будут храниться ключи шифрования.
Обозначение: _val - зашифрованное значение val
Оранжевым фоном выделен первичный ключ

Таблица **CHANNEL** содержит информацию о канале

Название столбца	**Описание**
channel_id	уникальный id канала
channel_name	имя канала
parent_channel_id	id канала-отца
_EncKey_parent	Ключ шифрования канала, зашифрованный ключом шифрования канала-отца

Таблица **USER** содержит информацию о пользователе

Название столбца	**Описание**
user_id	уникальный id пользователя
email	при помощи эл. почты можно будет восстановить доступ к учетной записи
email_pass_hash	хэш от пары (email:пароль), используется для авторизации
PubKey	Публичный ключ. Им шифруются доступные пользователю ключи шифрования каналов (enc_key)
_PrivKey	Приватный ключ
name	Имя пользователя
surname	Фамилия пользователя

Таблица **USER_IN_CHANNEL** будет использоваться для хранения информации об участниках каналов и их персональных настройках.

Название столбца	**Описание**
user_id	id пользователя
channel_id	id канала
is_in	true - состоит в канале, false - имеет доступ к каналу
user_role	Права пользователя в этом канале
_EncKey_user	Копия ключа шифрования канала, зашифрованная ключом пользователя
preferences	JSON-файл с пользовательскими настройками. Обрабатывается только во фронтенде (?).

Таблица **MESSAGE** содержит сообщения

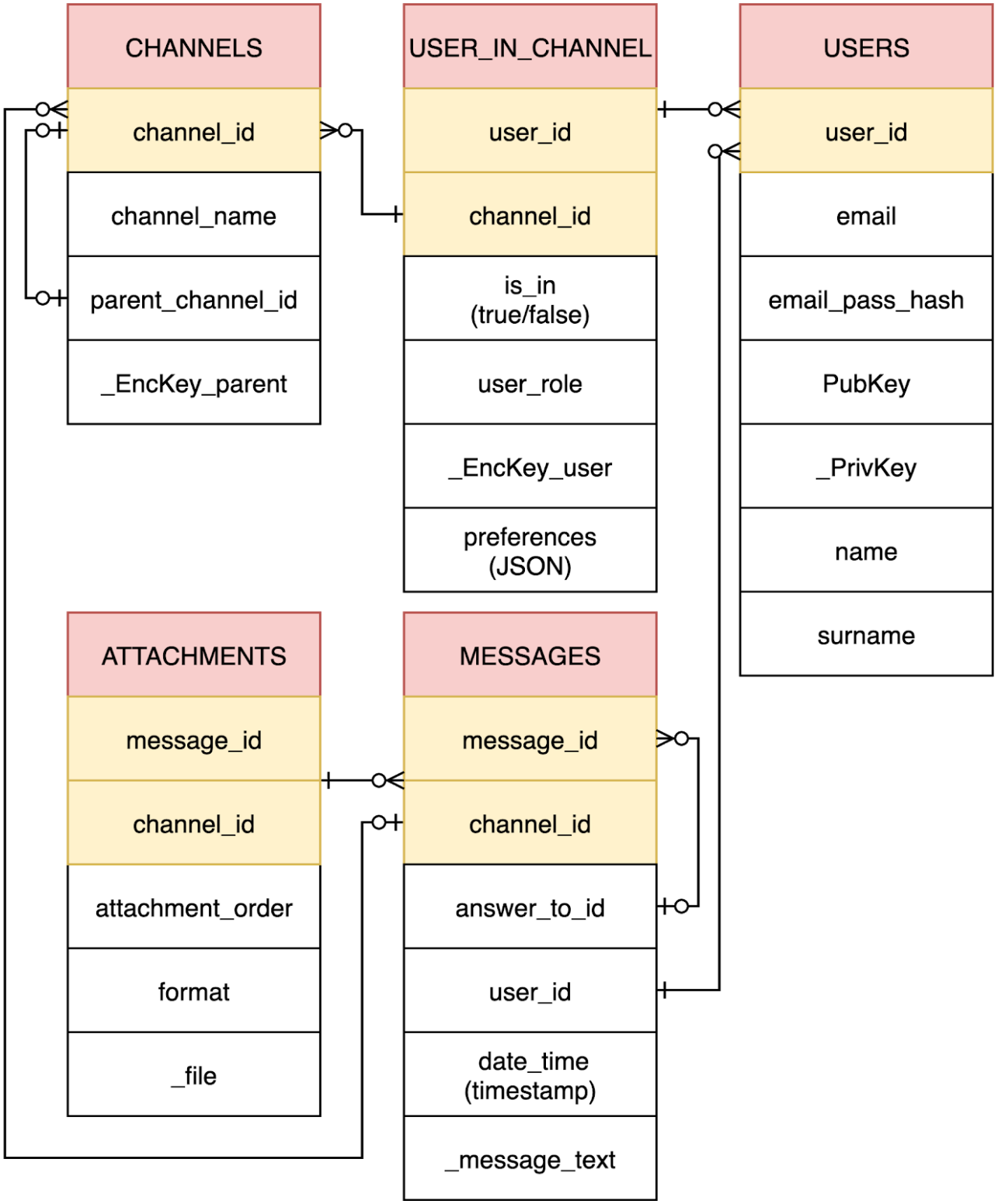
Название столбца	**Описание**
message_id	уникальный id сообщения в канале
channel_id	id канала
answer_to_id	id сообщения, на которое отвечает это сообщение
user_id	логин автора сообщения
date_time	дата и время (timestamp with timezone)
_message_text	текст сообщения, зашифрованный ключом канала

Таблица **ATTACHMENT** содержит вложения, прикрепленные к сообщениям

Название столбца	**Описание**
message_id	id сообщения, к которому прикреплено вложение
channel_id	id канала с сообщением
attachment_order	номер вложения в письме (чтобы прикрепленные фотографии не перемешивались)
format	формат файла (png, mp4, …)
_file	сам файл (в зашифрованном виде)

Схема полученной базы данных

выглядит так:



Оптимизация

СУБД автоматически создает индексы по первичным и внешним ключам.
Требуется создать индексы по тем остальным полям, по которым часто будет происходить поиск строк.

Для ускорения выполнения типовых запросов следует:

1. Для таблицы **CHANNELS** создать индекс по полю **parent_channel_id**
2. Для таблицы **USERS** создать индексы по полю **email**
3. Для таблицы **MESSAGES** создать индекс по полям **channel_id**, **date_time** и **answer_to_id**
4. Для таблицы **ATTACHMENTS** создать индексы по полю **message_id**