текже объясни admin-delete-user

Gemini

Конечно, давайте разберём Edge-функцию admin-delete-user. Это мощный административный инструмент, который дополняет логику управления пользователями.

**Объяснение: Edge-функция admin-delete-user**

Эта функция ([d:\ProjectGitHub\DashBoardOrtobumNew\supabase\functions\admin-delete-user\index.ts](code-assist-path:d:\\ProjectGitHub\\DashBoardOrtobumNew\\supabase\\functions\\admin-delete-user\\index.ts" \o "d:\\ProjectGitHub\\DashBoardOrtobumNew\\supabase\\functions\\admin-delete-user\\index.ts)) представляет собой защищённый API для выполнения двух связанных, но разных задач:

1. Удаление пользователя из конкретной организации.
2. (Опционально) Полное удаление аккаунта пользователя из всей системы.

Она предназначена для вызова из доверенных систем (например, панели администратора или автоматизированных скриптов), а не напрямую из клиентского приложения.

**Алгоритм работы**

1. **Безопасность**: Функция принимает POST-запрос и первым делом проверяет наличие секретного x-api-key в заголовках. Это гарантирует, что её могут вызывать только авторизованные сервисы.
2. **Валидация**: Проверяются входные данные: organization\_name и user\_email.
3. **Поиск организации**: Используя сервисный ключ (service\_role), функция находит ID организации по её имени. Если организация не найдена, возвращается ошибка.
4. **Делегирование удаления из организации**: Самый важный шаг. Функция вызывает SQL-функцию remove\_user\_from\_organization, передавая ей ID организации и email пользователя.
   * **Почему это важно?** Вся критическая бизнес-логика находится именно в этой SQL-функции. Она проверяет, не является ли удаляемый пользователь **последним администратором** в организации. Если это так, SQL-функция вернёт ошибку, и Edge-функция остановит выполнение. Это прекрасный пример выноса правил безопасности на уровень базы данных, что делает систему надёжнее.
5. **Опциональное удаление из Auth**: Если в запросе был передан флаг delete\_from\_auth: true, функция выполняет дополнительное, очень ответственное действие:
   * Она находит пользователя в системной таблице auth.users по email.
   * Вызывает sb.auth.admin.deleteUser(), чтобы **полностью и безвозвратно удалить аккаунт пользователя** из системы аутентификации Supabase.
6. **Ответ**: Функция возвращает сообщение об успехе, уточняя, был ли пользователь удалён только из организации или ещё и из системы Auth.

**Ключевые моменты и дизайн**

* **Разделение ответственности**: Это отличный пример правильного проектирования. Edge-функция выступает в роли **оркестратора**: она проверяет доступы, координирует вызовы и обрабатывает опциональные шаги. А SQL-функция (remove\_user\_from\_organization) выступает в роли **исполнителя бизнес-правил**, гарантируя целостность данных (нельзя оставить организацию без администратора).
* **Безопасность**: Использование service\_role ключа оправдано, так как для удаления пользователя из auth.users необходимы максимальные привилегии. Защита через API-ключ — это стандартная и надёжная практика для таких функций.
* **Гибкость**: Флаг delete\_from\_auth делает функцию очень гибкой. Можно просто убрать пользователя из одной из его организаций, не затрагивая его аккаунт, или же полностью удалить его из системы.

**Сравнение с другими процессами**

* **В отличие от удаления через RLS-политику** (которое может инициировать сам пользователь или админ из интерфейса), эта Edge-функция — это **супер-административный** инструмент. Она может быть вызвана для любой организации и любого пользователя, пока у вызывающей системы есть API-ключ.
* **В отличие от invite-user**, которая является созидательной функцией, admin-delete-user — разрушительная, и её использование требует большей осторожности.

В целом, это хорошо спроектированная и безопасная функция для выполнения критически важных административных задач.