# 『とりあえず削除フラグ』をやめる

# 話すこと

- 論理削除の悪い点
- 『削除』を実現する色々な方法
- どう向き合っていくか

# はじめに 物理削除と論理削除

# 物理削除

DB上のデータを直接消す

```sql
DELETE FROM orders WHERE id = 1
```

DB上からデータが消える → 復旧できない

# 論理削除

DB上のデータは消さない

```
UPDATE orders SET deleted_at = "2016-12-26 09:53:57"
```

DB上にデータは残り続ける → 復旧できる

# 何が問題なのか？

- クエリが複雑になる
- データが残り続ける
- 削除してよいと思い込んでしまう
- 削除でない操作に対してdeletedという単語が用いられてしまう

# クエリが複雑になる

```sql
SELECT * FROM orders WHERE deleted_at IS NOT NULL;
```

毎回whereが必要

# データが残り続ける

DB上のデータが消えずに残り続ける
→ データ増大によって検索などが遅くなる

# 削除してよいと思いこんでしまう

deleted_atカラムが存在することで
削除され得る値なんだなという印象を持ってしまう

# 削除でない操作に対してdeletedという単語が用いられてしまう

- 注文を「キャンセル」する
- 商品を「廃止」する
- 社員が「退職」する

ユーザーはその操作を「削除」と呼んでいるか？ 事実は消えない

# どうすればよいか

削除フラグやdeleted_at以外の選択肢を挙げてみます

- ドメインの言葉を使う
- 状態にする
- 削除しない
- 物理削除する

# ドメインの言葉を使う

- is_canceled
- is_active
- is_retired

正しい言葉、明確な言葉を使おう

# フラグではなく状態にする

status (cancel, complete)

そもそも二値で表せないことのほうが多い

# 削除しない

CRUDのU(Update)とD(DELETE)はコストが高い

- 履歴テーブル
- イミュータブルデータモデル

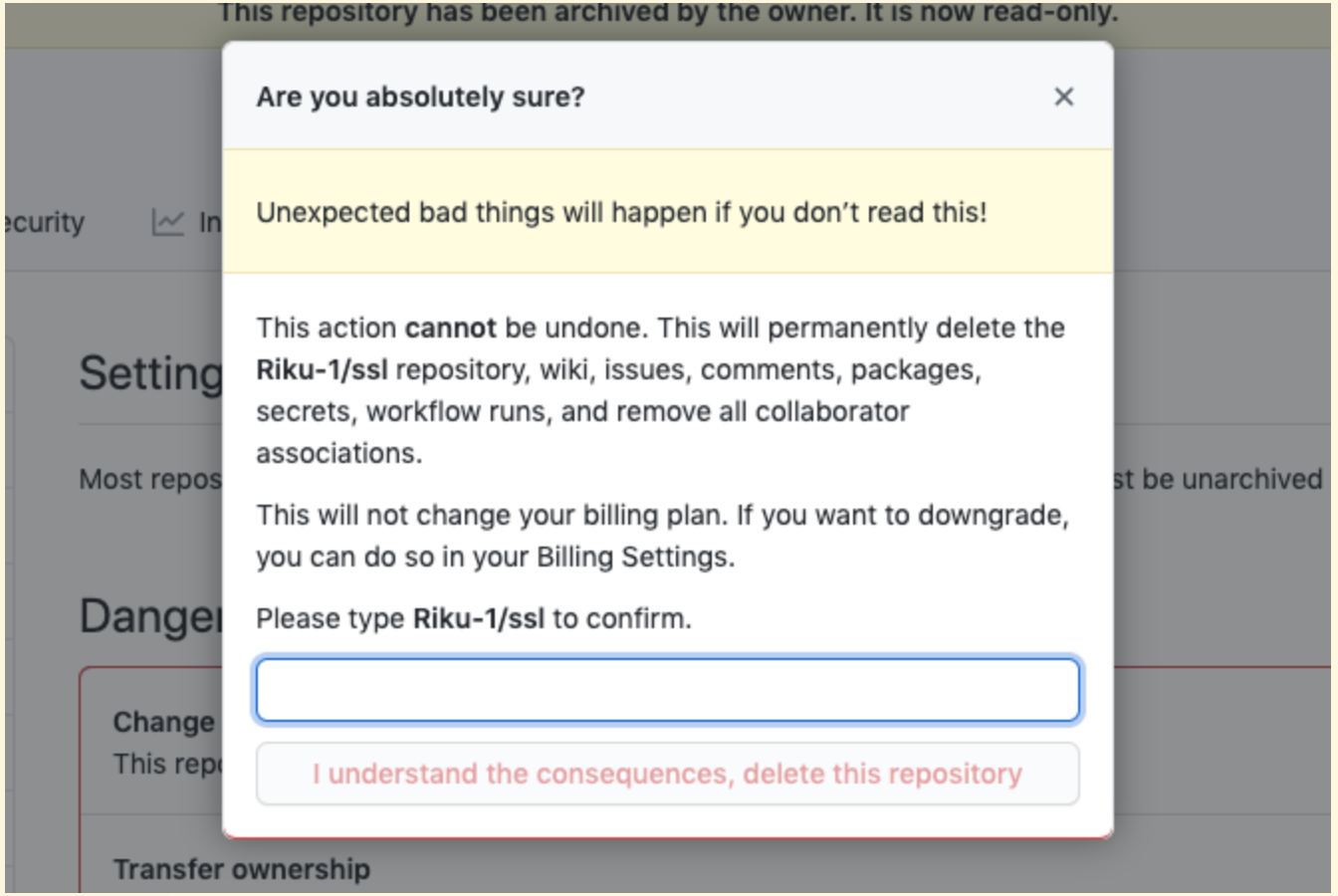
# 物理削除する

ただ物理削除したのでは「元に戻す」ことができない...
→ 削除前に念入りに確認する

# AWSの場合


tdekq2mlenht36 インスタンス を削除しますか?
DB インスタンス tdekq2mlenht36 を削除してよろしいですか?
最終スナップショットを作成しますか?
DB インスタンスの削除前に最終 DB スナップショットを作成するかどうかを決定します。
最終スナップショット名
新しい DB のスナップショットの DBSnapshotIdentifier が作成されました。
tdekq2mlenht36-final-snapshot
削除を確認するには、delete me というフレーズを以下のフィールドに入力します
キャンセル
削除

# Githubの場合


This repository has been archived by the owner. It is now read-only.
Are you absolutely sure?
Unexpected bad things will happen if you don’t read this!
This action cannot be undone. This will permanently delete the Riku-1/ssl repository, wiki, issues, comments, packages, secrets, workflow runs, and remove all collaborator associations.
This will not change your billing plan. If you want to downgrade, you can do so in your Billing Settings.
Please type Riku-1/ssl to confirm.
I understand the consequences, delete this repository
Transfer ownership

# 物理削除にもデメリットはある

- 時間がかかる
- 戻せない

# どの方法も銀の弾丸ではない

どれが適切かちゃんと考えよう
よく考えた上での削除フラグ、論理削除ならOK

# まとめ

- 論理削除の悪い点
- 「削除」を実現するいろいろな方法
- どう向き合っていくか

どの方法が適切かちゃんと考えるといいシステムが作れるんではないでしょうか

# 参考

SQLアンチパターン 幻の26章 「とりあえず削除フラグ」