

# 「RDBMS を単なるデータストアとして扱う」

表参道.rb #25 ~ Rails アンチパターン~

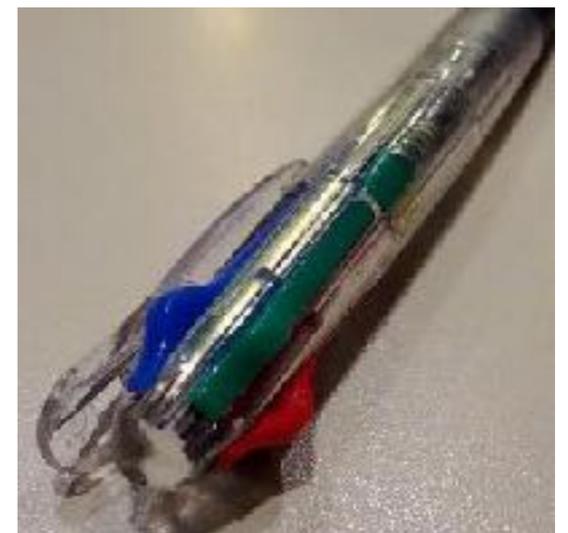
2017-08-03

@s\_osa\_

# 誰？

---

- OSA Shunsuke
- Rails で Web アプリケーションを書いています
  - Working on クックパッド料理教室
- 好きなこと：モデリング
  - オブジェクトやその相互作用を考えるのが好き
- Twitter: @s\_osa\_
- GitHub: s-osa



こんなことで困っていませんか？

# 親リソースの存在確認

---

- belongs\_to で定義されているリソースの存在確認を頻繁にしている
  - if article.blog
  - nil?
  - blank?
  - present?
  - try
  - &

# 1つしかないはずのレコードが複数ある

---

- 本来は unique であるはずの列が unique になっていない
  - 「Favorite を取り消したけど消えない」
  - 調べてみたら2件データができていて、片方しか削除されていなかった

なぜこんなことに？

# 親リソースの削除

---

- 他のリソースによって参照されているリソースを削除してしまう
- Rails では Association の定義に dependent オプションを渡すと削除時の動作が定義できる
  - nullify: 結局 nil check が必要
  - restrict\_with\_\*: (使っている人を見たことがない)

# validation/callback を介さないデータ更新

---

- `update_attribute(:column, value)`
- `save(validate: false)`
- 他のアプリケーション
- SQL

# 多重生成

---

- 同じデータを複数回書き込むと重複データが作成されてしまう
- 「ボタンは1回だけ押してください」

# アプリケーションにはバグが入りやすい

---

- 人間は間違えるのでバグを生んでしまう
- アプリケーションは頻繁に更新される

アプリケーションでは防げない

RDBMS の制約を使う

# FOREIGN KEY 制約

---

- 外部キーの参照先が存在することを保証する
- 参照先が存在しないレコードを INSERT/UPDATE したり、参照されているレコードを DELETE するとエラーになる
- Rails 4.2 でサポートされるようになった 🎉

# UNIQUE 制約

---

- 指定したカラムのデータが一意であることを保証
- 重複レコードを発生させるクエリはエラー
- 外部キーに設定すると便利なことが多い
  - `has_one` されている外部キーに定義して最大1件であることを保証
  - 中間テーブルの外部キーに複合 `UNIQUE` 制約を定義して、同じ組み合わせは保存されないように
    - `users <— favorites —> articles`

# おまけ：その他の制約

---

- NOT NULL 制約
  - デフォルトでつけるべき
  - ミックさんによる「NULLにまつわる諸問題」がオススメ
    - [http://www.geocities.jp/mickindex/database/idx\\_database.html](http://www.geocities.jp/mickindex/database/idx_database.html)
- CHECK 制約
  - 便利だけどルールから外れてしまう
    - schema.rb -> structure.sql
  - MySQL では使えない 😭

制約によって得られるもの

# データを信頼できる

---

- FOREIGN KEY 制約がついている外部キーは確実に  
関連リソースをたどれる
- UNIQUE 制約がついているカラムは確実に unique
- nil check やレコード数の確認が不要になってコード  
がスツキリする

# 誤りに早く気付ける

---

- 不正なデータを生成しようとするクエリが投げられたら、例外によってそれに気づくことができる
- 異常状態に早く気づけば原因を突き止めるのが容易になる
  - 壊れたデータを表示するときまで異常であることに気づけないと、どこで生成・更新されたのかを調べるのは大変

RDBMS と仲良くして  
楽しくコードを書こう