

データベース設計論

リレーショナルデータモデル以外のデータモデル

2020/1/21

本日の内容

- SQLインジェクション
 - （現時点での）サンプルプログラムは実はやばいという話
- リレーショナルデータモデル以外の（最近使われている）データモデルの話
 - NoSQLとドキュメントデータベース
 - グラフデータベース

SQLインジェクション

- アプリケーションの入力フォーム等を使って想定以外のSQL文を実行させること
 - アプリケーションのセキュリティ上の不備によるもの

```
param = <入力フォームから取得した値>
query = "SELECT id, name, email-address
        FROM meibo
        WHERE id = '"+param+"'"
db.all(query, {}, function(req, res, next){...});
```

問題：何が起こる？

検索

検索ボタンを押すと、以下のプログラムが実行される

```
param = " x' or 1=1; --"  
query = "SELECT id, name, email-address  
        FROM meibo  
        WHERE id = ' " + "1' or 1=1; --" + ""  
db.all(query, {}, function(req, res, next){...});
```

何が実行されることになるでしょう？

対策：エスケープ処理

- 入力された値に'や/,;等のエスケープシーケンスがあったとき、それを無効なものに置き換える
- プレースホルダを使う
 - プレースホルダ：値を埋め込む場所を?等の記号にしたSQL文を用意し、APIで用意されたメソッドを使って値を埋め込むようにする

```
param = <入力フォームから取得した値>
query = "SELECT id, name, email-address
        FROM meibo
        WHERE id =?"
db.all(query,param, function(req,res, next){...});
```

リレーショナルデータベース以外の データベース

- 第1世代：
 - 階層データベース
 - ネットワークデータベース
- 第2世代：
 - リレーショナルデータベース
- 第3世代：
 - オブジェクト指向データベース
- それ以降：
 - Key-value-store (KVS)
 - ドキュメント指向データベース
 - グラフデータベース

リレーショナルデータベースモデルの利用で
困ること

1. 何かと結合が必要になる
2. スキーマが固定されている

具体例：ユーザプロフィール

- お茶大の学生プロフィールを共有するデータベースを作ることを想定してみよう
 - 各学生について
 - 氏名
 - ニックネーム
 - 趣味
 - 最近気になっていること
 - 習得したスキル
 - 連絡先
 - メッセージ

演習：リレーショナルデータベースモデルなら
どうやって設計する？

黒板に書きます

困ることその1:

リストをそのまま保存したい

- 1人分のプロフィールを表示するために複数のクエリとたくさんの結合が必要になる
 - ニックネーム
 - 趣味
 - 最近気になっていること
 - 習得したスキル

やってみよう

- 一人分のプロフィールをWebページ上に表示するためのSQL文を書いてみよう

頻繁な結合はパフォーマンスに影響する

- 自己紹介DBが世界展開したら？
 - LinkedIn
 - 日本でもリクナビの持っているデータベースは膨大
- 1人の履歴書を検索するたびに数千万、数億のレコードを持つテーブルとの結合を何度も行うのは非現実的

困ることその2：人によって保存したい情報が異なる

- 習得したスキル：
 - 情報科学科の場合
 - 使ったことのあるプログラミング言語と習得状況
 - 化学科の場合？
 - 物理科の場合？
- 連絡先：
 - 電話番号
 - メールアドレス
 - Facebookアカウント
 - Twitterアカウント
 - 他にも…（人によって連絡手段は様々）

NoSQL

- Not only SQL データベース
 - リレーショナルデータモデル以外のデータモデルでデータを管理するDBMSの大まかな分類後
 - 2010年代に出現した
 - もとはTwitterのハッシュタグから
- 主なNoSQLデータベースのモデル
 - ドキュメント指向データモデル
 - Key-value-store
 - グラフデータモデル

ドキュメント指向データモデル

- 階層型のデータを分解しないまま保存できるようにする
- 例：JSON型でのプロフィールデータ

```
{
  name: '渡辺知恵美',
  birthday: '1975/06/27',
  hobby: ['駅伝観戦', 'コーヒー鑑賞', '自転車', '子育て'],
  carrer: [
    {
      period: ['2003/4', '2004/9'],
      organization: '奈良女子大学',
      position: '助手'
    },
    {
      period: ['2004/10', '2013/4'],
      organization: 'お茶の水女子大学',
      position: '講師'
    }
    ...
  ] ...
}
```

ドキュメント指向データベース

- よく使われているデータベースプロダクト
 - MongoDB, RethinkDB, CouchDB, Espresso
- 特徴
 - スキーマレス：ドキュメントのスキーマはあらかじめ定義しない
 - スキーマは統一しなくてもいい (🤔 後述)
 - 結合がほとんど起きない

問い合わせ例

- 習得した言語にSQLが含まれている人のプロフィールを取得する（MongoDBの場合）

```
db.user.find({mastered_language: 'SQL' })
```

Key-Value-Store (KVS)

- 概要
 - keyとvalueのセットで保存する
 - keyで検索するとvalueが返ってくる
- よく使われているプロダクト

ドキュメント指向データベースは 階層データのみを扱う？🤔

- 多々多の関係は扱えなくなる？

```
{
  name: '渡辺知恵美',
  birthday: '1975/06/27',
  hobby: ['駅伝観戦', 'コーヒー鑑賞', '自転車', '子育て'],
  carrer: [
    {
      period: ['2003/4', '2004/9'],
      organization: '奈良女子大学',
      position: '助手'
    },
    {
      period: ['2004/10', '2013/4'],
      organization: 'お茶の水女子大学',
      position: '講師'
    }
  ] ...
}
```

```
organizations = [
  {
    name: '奈良女子大学',
    category: '国立大学',
    address: '奈良県奈良市北魚屋西町'
  },
  ...
  {
    name: 'お茶の水女子大学',
    category: '国立大学',
    address: '東京都文京区大塚2-1-1'
  }
  ...
]
```

ドキュメント参照を利用して 多対多関係も表現できる

```
{
  name: '渡辺知恵美',
  birthday: '1975/06/27',
  hobby: ['駅伝観戦', 'コーヒー鑑賞', '自転車', '子育て'],
  carrer: [
    {
      period: ['2003/4', '2004/9'],
      organization: ObjectId(' 54759eb3c090d83494e2d804')
      position: '助手'
    },
    {
      period: ['2004/10', '2013/4'],
      organization: ObjectId(' 54759eb3c090d83449482d3a')
      position: '講師'
    }
  ] ...
}

organizations = [
  {
    name: '奈良女子大学',
    category: '国立大学',
    address: '奈良県奈良市北魚屋西町'
  },
  ... ObjectId(' 54759eb3c090d83494e2d804')
  {
    name: 'お茶の水女子大学',
    category: '国立大学',
    address: '東京都文京区大塚2-1-1'
  },
  ... ObjectId(' 54759eb3c090d83449482d3a')
]
```

ドキュメント指向DBはスキーマレス？

- 保存するときに指定しないだけ
 - 問い合わせするときにデータ構造の理解が必要
- スキーマオンライト
 - データを格納する時点でスキーマを確定する
 - プログラミング言語における静的型チェック
- スキーマオンライト
 - データの読み取り時に解釈される
 - プログラミング言語における動的型チェック

結局は…

万能のデータベースは存在しない

“A one size fits all database doesn't fit anyone”

Werner Vogels/CTO – Amazon.com