

リレーショナルデータモデルの実装例

設計

spec PK varchar(60),option longtext,comment longtext

DBモデル : specは実データより最大値が想定できたのでvarchar(60), option, comment はいずれもTEXT型の可変長カラムとして設計

▼基本テーブル

| spec | option | comment |
|-------|---------|----------|
| spec1 | option1 | comment1 |
| spec2 | option1 | comment2 |
| spec3 | option2 | comment3 |
| spec2 | option1 | comment4 |
| spec1 | option1 | comment5 |
| spec3 | option3 | comment6 |
| spec1 | option3 | comment7 |

Spec1を抽出し集計する
optionをすべて取得する

▼spec1リスト

| spec | option | comment |
|-------|---------|----------|
| spec1 | option1 | comment1 |
| spec1 | option1 | comment5 |
| spec1 | option3 | comment7 |

option1を抽出し、集計する
commentをすべて取得する

▼spec1・option1リスト

| spec | option | comment |
|-------|---------|----------|
| spec1 | option1 | comment1 |
| spec1 | option1 | comment5 |

▼全取得内容

```
spec1_option1=[comment1,comment5]
spec1_option3=[comment7]
spec2_option1=[comment2,comment4]
spec3_option2=[comment3]
spec3_option3=[comment6]

spec1_count=3 spec1_option1_count=2
spec1_option3_count=1
spec2_count=2 spec2_option1_count=2
spec3_count=2 spec3_option2_count=1
spec3_option3_count=1
```

(注)実際には再帰問い合わせ等を利用して処理を1回で済むように実装する。

specの値の分
繰り返す

optionの値の分
繰り返す

CDSを用いたJavaプログラムでの開発

設計

$\sum \text{spec} \times (\sum \text{option} \times \sum \text{comment})$

DBモデル : 設計における型の定義が不要

spec、option、commentそれぞれの和をツリー構造で表現する
各spec値は各option値を持ち、各option値は各comment値を持つ

▼CDSデータ例

spec1 × option1 × comment1+
spec2 × option1 × comment2+
spec3 × option2 × comment3+
spec2 × option1 × comment4+
spec1 × option1 × comment5+
spec3 × option3 × comment6+
spec1 × option3 × comment7

spec値,option値,comment値の
組み合わせ (×) の和 (+) を作成

CDS-APIによる組み換え、各specの部分式に注目し、項の数より各specの集計が取得できる

spec1 × (option1 × comment1+option1 × comment5+option3 × comment7)+
spec2 × (option1 × comment2+option1 × comment4)+
spec3 × (option2 × comment3+option3 × comment6)

赤囲内はspec1が持つoption値とcomment値
の全ての組み合わせの和を表している。和の数を取得
するAPIによりspec1の集計が得られる。

CDS-APIによる組み換え、各optionの部分式に注目し、項の数より各spec・option
の集計とcommentのリストが取得できる

spec1 × (option1 × (comment1+comment5)+option3 × comment7)+
spec2 × (option1 × (comment2+comment4))+
spec3 × (option2 × comment3+option3 × comment6)

赤囲内はspec1のoption1におけるcomment値
の和を表している。和の数を取得するAPIと和の各
値をリスト化するAPIによりspec1・option1の組
み合せの集計、commentリストが取得できる。

CDS-APIによる組み換え

```
spec1×option1×(comment1+comment5)+
spec1×option3×comment7+
spec2×option1×(comment2+comment4)+
spec3×option2×comment3+
spec3×option3×comment6
spec1×count=3
spec2×count=2
spec3×count=2
spec1×option1×count=2
spec1×option3×count=1
spec2×option1×count=2
spec3×option2×count=1
spec3×option3×count=1
```

相違点

- ・リレーショナルデータモデル側は繰り返しの矢印毎に処理が入るが、CDSモデル側は構造情報を含む全データが内包されているので一度の処理で結果を取得する
- ・リレーショナルデータモデル側は設計時、実データに沿った内容で型の定義を行う必要があるが、CDSモデル側は型の定義が不要