

ソフトウェア総合演習 I

書籍管理システム フェーズ3

課題 9 データベースによる永続化

書籍紹介ページに含まれる情報や、計算したページランクなど、書籍の推薦に関連する情報を、データベースによって永続化するプログラムを作成せよ。

入力

課題 7 と同様。

永続化を行った後、空白で区切られた複数の単語からなる 1 行のクエリを入力する。

出力

推薦ページをスコアの降順で出力する。

入力例

```
list queue bfs
```

出力例

```
word  page    score
1.00 : 0.94 : 1.94 : graph.html
0.67 : 1.00 : 1.67 : data.html
0.00 : 0.70 : 0.70 : search.html
0.00 : 0.42 : 0.42 : sort.html
0.00 : 0.09 : 0.09 : algorithm.html
```

実装例

例えば、MySQL を利用し、各言語のデータベース接続 API を利用する。データベースとして、簡易的なデータベース管理システム SQLite を利用することができる。

データベースのスキーマは以下のように定義できる。

```
create table url (url varchar(100))
create table word (word varchar(100))
create table link (source int, target int)
create table location (wordid int, urlid int)
```

- テーブル url はドキュメント（書籍）の URL のリストである。属性 url は URL を表す文字列で、その ID が rowid によって対応付けられている。
- テーブル word はドキュメントの中に含まれる単語のリストである。属性 word は単語を表す文字列で、その ID が rowid によって対応付けられている。
- テーブル link は、リンクを表し、source がリンク元の url ID、target がリンク先の url ID を表す。
- テーブル location は、各単語が現れるドキュメントを保持するもので、wordid が単語の ID、urlid がその単語が含まれるドキュメントの URL の ID を表す。

以下に、クエリの入出力例を示す。

```
select rowid,* from url;
```

```
1|algorithm.html
2|sort.html
3|search.html
4|data.html
5|graph.html
```

```
select rowid,* from word;
```

```
1|program
2|algorithm
3|data
4|structure
5|sort
6|merge
7|array
8|binary
9|list
10|queue
11|stack
12|bfs
13|dfs
14|node
15|edge
```

```
select rowid,* from link;
```

```
1|1|2
2|1|3
3|1|4
4|2|4
5|2|3
```

6|3|4
7|3|2
8|4|5
9|5|4
10|5|3

```
select rowid,* from location;
```

1|1|1
2|2|1
3|3|1
4|4|1
5|5|2
6|6|2
7|7|2
8|7|3
9|8|3
10|9|4
11|7|4
12|10|4
13|11|4
14|7|4
15|9|5
16|12|5
17|13|5
18|10|5
19|11|5
20|14|5
21|15|5

このように、検索対象となるドキュメントの永続化ができる。
さらに、課題7で計算したページランクなども、永続化できる。

```
select rowid,* from pagerank;
```

1|1|0.15
2|2|0.664057632054087
3|3|1.1096486754485
4|4|1.58124936251411
5|5|1.49406195813699

これらデータベースの情報を利用して、入力されたキーワードに対して課題8のように書籍の推薦を行う。

課題 10 ユーザインタフェースの実装

これまでの小課題で作成したプログラムを統合し、図書管理システムを完成させよ。例えば、以下の機能拡張が考えられる。

- 全ての項目を永続化し、ユーザ、図書、友人関係、ドキュメント等を追加・更新・削除できるユーザインタフェースを構築する
- インタラクティブに推薦を行うユーザインタフェースを構築する
- インタラクティブに検索を行うユーザインタフェースを構築する

※ユーザインタフェースは CUI でもよい（GUI にする必要はない）

課題 11 推薦機能・検索エンジンの強化

想定される用途を考慮し、推薦機能・検索エンジンを強化せよ。例えば、以下の機能拡張が考えられる。

- 友人関係に距離を定義し、距離を限定した友人のみを考慮する
- 仲の良いグループを定義し（例えば、グラフの強連結成分）、同じグループに属する友人のみを考慮する
- クローラを実装し、永続化・検索対象を絞り込む
- 単語のみではなく、“priority queue” のような文章による検索を可能にする
- ワイルドカードを用いた検索を可能にする