

情報科学プロジェクト実験レポート課題

S142063 佐藤涼亮

平成 28 年 12 月 2 日

SQL(Structured Query Language)

1 課題の内容

C++プログラムによるデータベース操作

1.1 要点

- 住所から郵便番号を検索するプログラムを作成
- 郵便番号データ用 db を作成し、ファイル 13tokyo.csv の内容を insert するプログラムを作成
- 郵便番号から住所を検索するプログラムの作成
- 住所から郵便番号を検索するプログラムの作成
- db のエントリを更新したり削除したりするプログラムの作成

2 プログラムの説明

2.1 課題 1

郵便番号データ用 db を作成し、ファイル 13tokyo.csv の内容を insert するプログラムの作成。
テーブル名やスキーマは、事前に作成しておき、以下の形式を用いる。

```
create table post (  
    code integer,num5 char(5), num char(7),  
    kana1 text, kana2 text, kana3 text,  
    kanji1 text, kanji2 text, kanji3 text,  
    d1 integer, d2 integer, d3 integer,  
    d4 integer, d5 integer, d6 integer  
);
```

sqlite3 のライブラリ関数を用い、テーブルにデータの挿入 (INSERT) を行う。また、多量のデータを挿入するため、トランザクションを用い一つの SQL 文として扱い、一度のデータベースへの書き込みとして実行することで、研究室でのネットワーク間の実行もスムーズに行うことができる。

2.2 課題 2

- ・郵便番号から住所を検索するプログラム
- ・住所から郵便番号を検索するプログラム
- ・dbのエントリを更新したり削除したりするプログラム

以上の3つのプログラムを実装した。

実装方法は課題1と大差ないが、すべて、インジェクションを回避するためにバインドメカニズムを用いて実装した。sql文にプレースホルダを置きバインド変数として扱うことで、入力された文字列を数値定数や文字列定数として組み込まれる。よって、変数に; や'などの特殊文字が含まれていたとしても、ただの文字として扱われる。更新、削除のプログラムでは、条件を設定する関数を作成し、ANDとORにも対応させた。また、更新や削除で指定できる要素は団体コードと郵便番号、住所の3つに絞った。

2.3 目的

プログラムにおけるデータベースの操作。

2.4 方法

C++プログラムによるデータベースを操作するプログラムの実装。

2.5 結果

最小限の実行結果を記述する。

— 一連の実行結果 —

```
$ ./addrSearch 1002211
郵便番号:1002211 の住所は 東京都小笠原村母島 です。
$ ./postalSearch 東京都小笠原村母島
住所:東京都小笠原村母島 の郵便番号は 1002211 です。
$ ./addrSearch 1000001
郵便番号:1000001 の住所は 東京都千代田区千代田 です。
$ ./postalSearch 東京都千代田区千代田
住所:東京都千代田区千代田 の郵便番号は 1000001 です。
$ ./dlud
1 : Update
2 : Delete
3 : Quit
操作命令を番号で選択してください: 1
```

更新する要素の選択と入力

1 : 全国地方公共団体コード

2 : 郵便番号

3 : 住所

番号を入力してください : 2

郵便番号の入力 -> 1234567

条件の入力

1 : 全国地方公共団体コード

2 : 郵便番号

3 : 住所

番号を入力してください : 3

住所の入力 -> 東京都小笠原村母島

1 : AND

2 : OR

3 : Quit

番号を入力してください : 3

Quit

\$./addrSearch 1002211 <---郵便番号が更新されたため表示されない

\$./addrSearch 1234567 <---郵便番号が更新され見つかったので表示される

郵便番号:1234567 の住所は 東京都小笠原村母島 です。

\$./dlud

1 : Update

2 : Delete

3 : Quit

操作命令を番号で選択してください : 2

条件の入力

1 : 全国地方公共団体コード

2 : 郵便番号

3 : 住所

番号を入力してください : 2

郵便番号の入力 -> 1000001

1 : AND

2 : OR

3 : Quit

番号を入力してください : 3

Quit

\$./addrSearch 1000001 <---エントリが削除されたため表示されない

\$./postalSearch 東京都千代田区千代田 <---エントリが削除されたため表示されない

2.6 考察

郵便番号や住所の検索、データの更新や削除を行うことができ、期待通りの結果が得られた。

3 感想

データベースの操作は以前に行ったことがあったが、sql 文を入力するだけであった。今回は、C++のプログラムを用いてデータベースの操作を学んだ。プログラムを用いてデータベースの操作をする上で、SQL インジェクションの脅威があり、セキュリティ上簡単に実装できない。これらを踏まえてプログラムを実装するのは、少し骨が折れた。今後、必ず今回のようにデータベースを扱うことがあると思うので、インジェクションの脅威をしっかりと対策をすることで、セキュリティの高いプログラムを作成し、他者の信頼を得られるような人物になれるように精進したい。

4 プログラム

ソースコード 1: report1.cpp

```
1 #include <iostream>
2 #include <fstream>
3 #include <MyDBS3.hpp> // MyDBS3.hpp はコンピュータシステム研独自のヘッダファイル
4 using namespace std;
5
6 int main(int argc, char* argv[])
7 {
8     if(argc != 2){
9         cerr << "Error" << endl;
10        return 1;
11    }
12
13    // データベースファイルを開く
14    MyDBS d("tokyo.db");
15    if (!d) {
16        cout << d.error() << "\n";
17        return 1;
18    }
19
20    // ファイルオープン
21    ifstream fin(argv[1]);
22    if (!fin){
23        cerr << "fin" << endl;
24        return 1;
25    }
26
27    // テーブルの初期化
28    if (d.exec("delete_from_post") != SQLITE_OK) {
29        cout << d.error() << "\n";
30        return 1;
31    }
32
33    // トランザクションの開始
34    if (d.exec("begin_transaction;") != SQLITE_OK) {
35        cout << d.error() << "\n";
36        return 1;
37    }
38
39    // データの挿入
40    string line;
41    while(getline(fin,line)){
42        string sql = "insert_into_post_values_(" + line + ")";
```

```

43         if (d.exec(sql) != SQLITE_OK) {
44             cout << d.error();
45             return 1;
46         }
47     }
48 }
49
50 // トランザクションの終わり
51 if (d.exec("commit_transaction;") != SQLITE_OK) {
52     cout << d.error() << "\n";
53     return 1;
54 }
55
56 return 0; // MyDBS のデストラクタが DB を閉じてくれる
57 }

```

ソースコード 2: report2-1.cpp

```

1  #include <iostream>
2  #include <string>
3  #include <MyDBS3.hpp> // コンピュータシステム研独自のヘッダファイル
4  using namespace std;
5
6  int main(int argc, char* argv[]){
7
8      if(argc != 2){
9          cerr << "Error" << endl;
10         return 1;
11     }
12
13     // データベースファイルを開く
14     MyDBS d("tokyo.db");
15     if (!d) {
16         cout << d.error() << "\n";
17         return 1;
18     }
19
20     string postal_code = argv[1];
21
22     // 郵便番号が一致するものを表示する
23     string sql = "select_kanji1,kanji2,kanji3_from_post_where_num_=?";
24     if(d.prepare(sql,postal_code) != SQLITE_OK){
25         cout << d.error() << endl;
26         return 1;
27     }
28     string kanji1,kanji2,kanji3;
29     while(d.step(&kanji1,&kanji2,&kanji3) == SQLITE_ROW){
30         cout << "郵便番号:" << postal_code << "の住所は"
31             << kanji1 << kanji2 << kanji3 << "です." << endl;
32     }
33
34     return 0; // MyDBS のデストラクタが DB を閉じてくれる
35 }

```

ソースコード 3: report2-2.cpp

```

1  #include <iostream>
2  #include <MyDBS3.hpp> // MyDBS3.hpp はコンピュータシステム研独自のヘッダファイル
3  using namespace std;
4
5  int main(int argc, char* argv[]){
6
7      if(argc != 2){
8          cerr << "Error" << endl;
9          return 1;
10     }
11 }

```

```

12 // データベースファイルを開く
13 MyDBS d("tokyo.db");
14 if (!d) {
15     cout << d.error() << "\n";
16     return 1;
17 }
18
19 // プレースホルダとバインドメカニズムを用いることで
20 // 特殊文字を含むすべての文字をただの文字として扱う
21 // 住所が一致するものを表示
22 string addr = argv[1];
23 string sql = "select num, from post where kanji1||kanji2||kanji3=?";
24 if(d.prepare(sql,addr) != SQLITE_OK){
25     cout << d.error() << endl;
26     return 1;
27 }
28 string num;
29 while(d.step(&num) == SQLITE_ROW){
30     cout << "住所:" << addr << "の郵便番号は" << num << "です." << endl;
31 }
32
33 return 0; // MyDBS のデストラクタが DB を閉じてくれる
34 }

```

ソースコード 4: report2-3.cpp

```

1 #include <iostream>
2 #include <vector>
3 #include <MyDBS3.hpp> // MyDBS3.hpp はコンピュータシステム研独自のヘッダファイル
4 using namespace std;
5
6 // 更新する要素の選択
7 string set(vector<string> &v){
8     cout << "更新する要素の選択と入力\n";
9     int x;
10    string w;
11    string sql = "SET";
12    cout << "1:全国地方公共団体コード\n2:郵便番号\n3:住所\n番号を入力してください:"
13    ;
14    cin >> x;
15    if(x == 1) {
16        sql += "code=?";
17        cout << "全国地方公共団体コードの入力->";
18    }
19    else if(x == 2) {
20        sql += "num=?";
21        cout << "郵便番号の入力->";
22    }
23    else if(x == 3){
24        sql += "kanji1=?,kana1=?,kanji2=?,kana2=?,kanji3=?,kana3=?";
25        cout << "都道府県名の入力->"; cin >> w; v.push_back(w);
26        cout << "読み仮名 (カナ)の入力->"; cin >> w; v.push_back(w);
27        cout << "市区町村の入力->"; cin >> w; v.push_back(w);
28        cout << "読み仮名 (カナ)の入力->"; cin >> w; v.push_back(w);
29        cout << "町域名の入力->"; cin >> w; v.push_back(w);
30        cout << "読み仮名 (カナ)の入力->";
31    }
32    else {
33        cerr << "Error" << endl;
34        return "";
35    }
36    cin >> w;
37    v.push_back(w);
38    return sql;
39 }
40 // 条件の指定
41 string setWhere(vector<string> &v){

```

```

42 cout << "条件の入力\n";
43 int x;
44 string w;
45 string sql = "WHERE";
46 while(cout << "1:全国地方公共団体コード\n2:郵便番号\n3:住所\n
    n 番号を入力してください:")
47     && cin >> x){
48     if(x == 1) {
49         sql += "code=";
50         cout << "全国地方公共団体コードの入力->";
51     }
52     else if(x == 2) {
53         sql += "num=";
54         cout << "郵便番号の入力->";
55     }
56     else if(x == 3){
57         sql += "kanji1||kanji2||kanji3=";
58         cout << "住所の入力->";
59     }
60     else{cerr << "Error" << endl; continue;}
61     cin >> w;
62     v.push_back(w);
63     cout << "1:AND\n2:OR\n3:Quit\n n 番号を入力してください:";
64     cin >> x;
65     if(x == 1) sql += "AND";
66     else if(x == 2) sql += "OR";
67     else if(x == 3) break;
68     else cerr << "Error" << endl;
69     continue;
70 }
71 cout << "Quit" << endl;
72 return sql;
73 }
74
75 // 最終的なsql文の実行
76 int execute(MyDBS d,string sql,vector<string> v){
77     if(d.prepare(sql) != SQLITE_OK){
78         cout << d.error() << endl;
79     }
80     for(int idx = 0;idx<v.size();idx++){
81         if(d.bind(idx+1,v[idx]) != SQLITE_OK){
82             cout << d.error() << endl;
83             return 1;
84         }
85     }
86     int rc = d.step();
87     if(rc != SQLITE_DONE && rc != SQLITE_ROW) {
88         cout << d.error() << endl;
89         return 1;
90     }
91     if(d.reset() != SQLITE_OK){
92         cout << d.error() << endl;
93         return 1;
94     }
95     return 0;
96 }
97
98 int main(int argc, char* argv[]){
99
100     // データベースファイルを開く
101     MyDBS d("tokyo.db");
102     if (!d) {
103         cout << d.error() << "\n";
104         return 1;
105     }
106
107     int act;
108     cout << "1:Update\n2:Delete\n3:Quit\n n 操作命令を番号で選択してください:";

```

```

109     cin >> act;
110
111     string sql;
112     vector<string> v; // バインド変数の格納
113
114     switch(act){
115     case 1:
116         sql = "UPDATE_post" + set(v);
117         sql += setWhere(v);
118         if(execute(d,sql,v))return 1;
119         break;
120
121     case 2:
122         sql = "DELETE_FROM_post" + setWhere(v);
123         if(execute(d,sql,v))return 1;
124         break;
125
126     case 3:
127         cout << "Quit" << endl;
128         break;
129
130     default:
131         cerr << "Error" << endl;
132         return 1;
133     }
134
135     return 0; // MyDBS のデストラクタが DB を閉じてくれる
136 }

```