Санкт-Петербургский политехнический университет
Кафедра компьютерных систем и программных технологий

Отчёт по вводной лабораторной работе

Дисциплина: Базы данных

Тема: Разработка клиентского приложения.

Выполнил студент гр. 43501/4 _____ Чеботарёв Г.М.
(подпись)
Руководитель _____ Мяснов А.В.
(подпись)

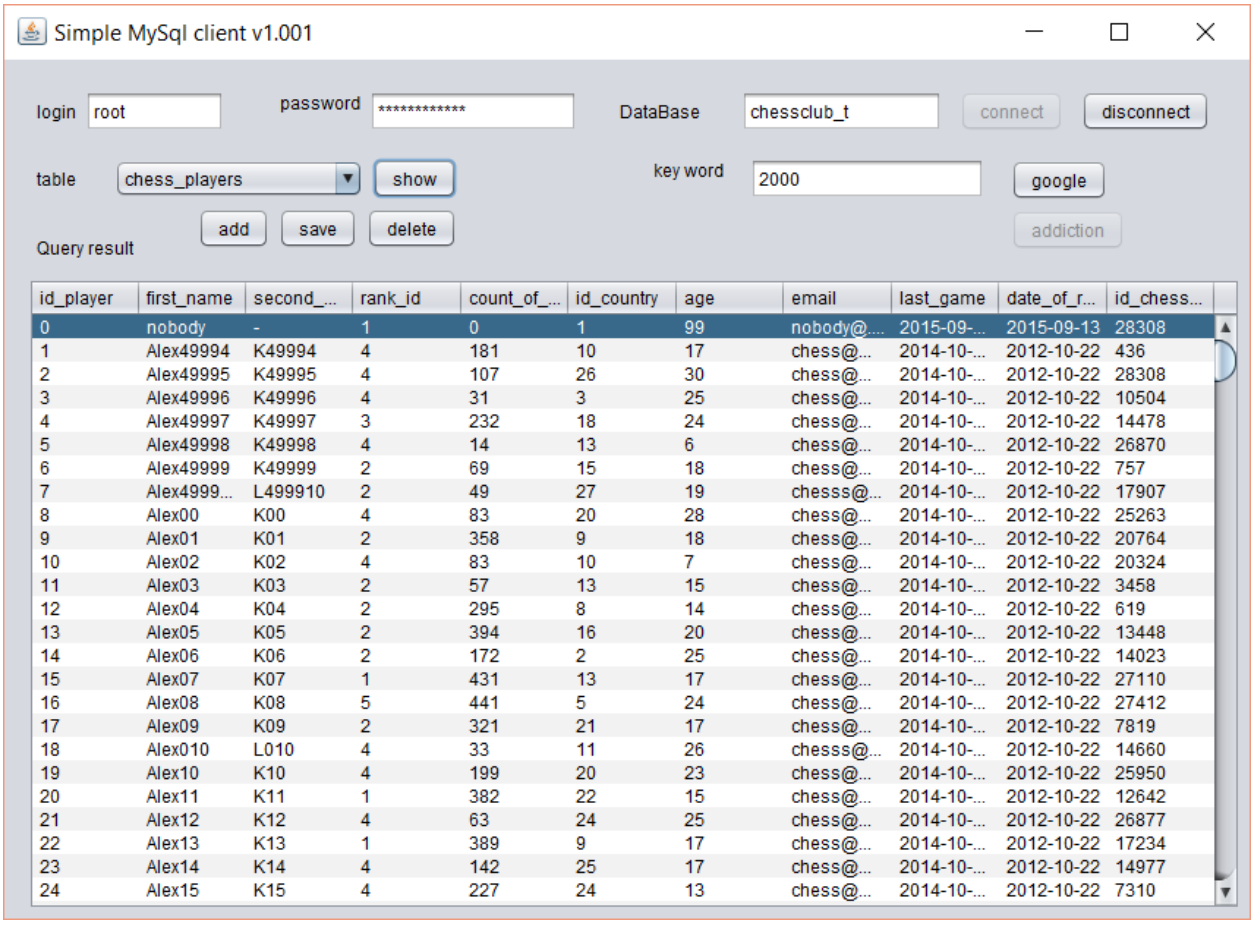
Санкт -Петербург
2016

Программа работы

Варианты приложения:

1. Консольное приложение
   - импорт и экспорт в xml нескольких таблиц
   - проверка с помощью схемы
2. **Графическое приложение с CRUD (технология Swing) для 3-4 таблиц**
   - **master-detail с редактированием**
   - **отчет с использованием запроса**
   - **запуск процедур и получение результатов**
3. Веб-приложение на JSP
   - master-detail с редактированием
   - отчет с использованием запроса
   - запуск процедур и получение результатов
4. Веб-приложение на XML + XSLT
   - master-detail
   - отчет с использованием запроса
   - запуск процедур и получение результатов

Ход работы:



id_player	first_name	second_...	rank_id	count_of_...	id_country	age	email	last_game	date_of_r...	id_chess...
0	nobody	-	1	0	1	99	nobody@....	2015-09-...	2015-09-13	28308
1	Alex49994	K49994	4	181	10	17	chess@...	2014-10-...	2012-10-22	436
2	Alex49995	K49995	4	107	26	30	chess@...	2014-10-...	2012-10-22	28308
3	Alex49996	K49996	4	31	3	25	chess@...	2014-10-...	2012-10-22	10504
4	Alex49997	K49997	3	232	18	24	chess@...	2014-10-...	2012-10-22	14478
5	Alex49998	K49998	4	14	13	6	chess@...	2014-10-...	2012-10-22	26870
6	Alex49999	K49999	2	69	15	18	chess@...	2014-10-...	2012-10-22	757
7	Alex4999...	L499910	2	49	27	19	chesss@...	2014-10-...	2012-10-22	17907
8	Alex00	K00	4	83	20	28	chess@...	2014-10-...	2012-10-22	25263
9	Alex01	K01	2	358	9	18	chess@...	2014-10-...	2012-10-22	20764
10	Alex02	K02	4	83	10	7	chess@...	2014-10-...	2012-10-22	20324
11	Alex03	K03	2	57	13	15	chess@...	2014-10-...	2012-10-22	3458
12	Alex04	K04	2	295	8	14	chess@...	2014-10-...	2012-10-22	619
13	Alex05	K05	2	394	16	20	chess@...	2014-10-...	2012-10-22	13448
14	Alex06	K06	2	172	2	25	chess@...	2014-10-...	2012-10-22	14023
15	Alex07	K07	1	431	13	17	chess@...	2014-10-...	2012-10-22	27110
16	Alex08	K08	5	441	5	24	chess@...	2014-10-...	2012-10-22	27412
17	Alex09	K09	2	321	21	17	chess@...	2014-10-...	2012-10-22	7819
18	Alex010	L010	4	33	11	26	chesss@...	2014-10-...	2012-10-22	14660
19	Alex10	K10	4	199	20	23	chess@...	2014-10-...	2012-10-22	25950
20	Alex11	K11	1	382	22	15	chess@...	2014-10-...	2012-10-22	12642
21	Alex12	K12	4	63	24	25	chess@...	2014-10-...	2012-10-22	26877
22	Alex13	K13	1	389	9	17	chess@...	2014-10-...	2012-10-22	17234
23	Alex14	K14	4	142	25	17	chess@...	2014-10-...	2012-10-22	14977
24	Alex15	K15	4	227	24	13	chess@...	2014-10-...	2012-10-22	7310

Рис. 1. – вид оконного приложения

Данное приложение написано на java с использованием JDBC библиотек. Реализована следующая функциональность:

1) Подключение (переподключение) к некоторой БД.
2) Возможность просматривать (изменять, добавлять, удалять) содержимое таблиц (любой таблицы из выбранной БД).

3) Возможность поиска ключевого слова в выбранной таблице, а также по таблицам, связанными внешними ключами с искомой, с последующим просмотром содержимого поля внешней таблицы.
4) Автоматическая подгрузка данных при прокрутке.
5) Система оповещения об ошибках.

Разработанное клиентское приложение протестировано на нескольких БД с различным содержимым. По результатам тестов, «комфортная работа» с таблицей до 130 Мб. При работе с таблицами 200-400 Мб возникают некоторые «торможения». Это обуславливается ограниченным потенциалом JDBC библиотеки.

Запросы генерируются автоматически, согласно заполняемым полям:

1. **Запрос данных на отображение**
   String query = "select * from " + jComboBox1.getSelectedItem() + " LIMIT "+start_lim+","+Limit+";";

   где ***jComboBox1.getSelectedItem()*** – имя таблицы, введенное пользователем в соответствующую графу,
   Limit – кол-во строк вытягиваемых за запрос.

2. **Запрос на добавление строки в таблицу:**
   String query = "INSERT INTO `" + jComboBox1.getSelectedItem() + "` (", name;
   ```
   for(int i =0; i< count_header; i++){
      query = query + (String)headerVect.get(i);
      if(i+1<count_header)
          query = query +", ";
   }
   query = query +") VALUES ( ";
   for(int i =0; i< count_header; i++){
      query = query + "'"+(String)new_data.get(i)+"'";
      if(i+1<count_header)
          query = query +" , ";
   }
   query = query +");";
   boolean rs = stmt.execute(query);
   ```
   В первом цикле генерируются заголовки, и конкатенируются. Во втором – добавляемые данные, взятые из jTable (т.е. то, что ввел пользователь в таблицу).

3. **Запрос на обновление данных**
   String query = "update `" + jComboBox1.getSelectedItem() + "` set ", name;
   ```
   for(int i =0; i< count_header; i++){
      query = query + (String)headerVect.get(i)+" = '"+(String)new_data.get(i)+"'";
      if(i+1<count_header)
          query = query +", ";
   }
   query = query +" where ";
   for(int i =0; i< count_header; i++){
      query = query + (String)headerVect.get(i)+" = '"+(String)old_data.get(i)+"'";
      if(i+1<count_header)
   ```

```
            query = query +" and ";
        }
        query = query +";";
```

Как и в предыдущем запросе: первый цикл – генерируют данные которые нужно заменить, второй – генерирует условия замены.

4. **Запрос на удаление строки**

```
String query = "DELETE FROM `" + jComboBox1.getSelectedItem() + "` WHERE ", name;
for(int i =0; i< count_header; i++){
   query = query + (String)headerVect.get(i)+" = '"+(String) jTable1.getValueAt(index_deleted,i)+"'";
   if(i+1<count_header)
      query = query +" and ";
}
query = query +";";
```

Запрос содержит ряд переменных:
jComboBox1.getSelectedItem() – имя таблицы,
headerVect – вектор заголовков таблица,
jTable1.getValueAt(index_deleted,i) – данные из конкретной ячейки

5. **Запрос на внешние ключи выбранной таблицы**

```
query = "SELECT COLUMN_NAME as master_col, REFERENCED_TABLE_NAME as name_slave, "
    + "REFERENCED_COLUMN_NAME as slave_col" +
    " FROM information_schema.KEY_COLUMN_USAGE\n" +
    "WHERE TABLE_SCHEMA ='"+jTextField3.getText()+"' AND TABLE_NAME ='"+
    jComboBox1.getSelectedItem() +"' AND\n" +
    "CONSTRAINT_NAME <>'PRIMARY' AND REFERENCED_TABLE_NAME is not null;";
```

Данный запрос возвращает все внешние ключи выбранной таблицы.

6. ***Генерация составного запроса для поиска ключевого слова по выбранной таблице, и таблицам, связанных с ней с помощью внешних ключей.***

6.1. ***Поиск по основной таблице***:

```
query = "select *, 'KEYTABLE', 'KEYWORD' from "+jComboBox1.getSelectedItem()+" where concat("
        + header_slave_table+") like '%"+jTextField2.getText()+"%' LIMIT 0,"+Limit+" ";
```

Основные переменные:
jComboBox1.getSelectedItem() – имя текущие таблицы;
header_slave_table – строка, в которой конкатенированы заголовки таблицы через запятую;
jTextField2.getText() – ключевое слово введенное пользователем;
Limit – кол-во строк вытягиваемых за запрос.

6.2. Поиск по второстипенним таблицам

```
query = query +  " UNION select T1.*,'"+external_keys_vec.get(i+1)+"', "
```

```
                    +
"concat_ws('='," "+external_keys_vec.get(i+2)+"",T2."+external_keys_vec.get(i+2)+
                     ") from "+jComboBox1.getSelectedItem()+ " as T1\n" +
                     "inner join " + external_keys_vec.get(i+1)+ " as T2 on concat"
                    + "(" + header_slave_table + ") like '%"+jTextField2.getText()+"%'\n" +
                     "where                T1."+external_keys_vec.get(i)+"                =
T2."+external_keys_vec.get(i+2) +  " LIMIT 0,"+Limit;
```

external_keys_vec.get(i+2) – имя внешней таблицы;
external_keys_vec.get(i+1) – имя поля во внешней таблице;
jComboBox1.getSelectedItem() – имя текущие таблицы;
header_slave_table – строка, в которой конкатенированы заголовки таблицы через запятую;
jTextField2.getText() – ключевое слово введенное пользователем;
Limit – кол-во строк вытягиваемых за запрос.

Подобные запросы генерируются для каждой из внешних таблиц, после чего «объединяются» с помощью UNION и отправляются на сервер mysql. Так как внешних ключей у таблицы не больше 3(chess_players), подобная архитектура запроса допускается. В случае, если количество таблиц, к которым необходимо было бы послать было бы велико, предпочтительнее было бы отправлять запросы по отдельности. Полный код приложения выложен на GitHub, ссылка добавлена в комментарии к заданию.

Вывод:

В разработанном приложении реализованы следующие требования:

1) CRUD
2) Подгрузка при прокрутке
3) Поиск ключевого слова по выбранной таблице и связанных таблицах.

К положительным сторонам приложения можно отнести ее независимость от БД. Приложение будет одинаково работать как с chessclub_t, так и с любой другой базой данных. При разработке были получены (возможно)полезные навыки. К недостаткам – полная непрактичность и ненадобность.