연락처 관리 시스템 만들기

(Assignment 2: RDBMS 와 web 을 이용)

2015004493
김형순

0. 개발/테스트 환경

① 연락처 관리 시스템은 postgres database 를 사용하여 연락처를 관리하고, psycopg 와 flask 를 이용하여 서버를 구성한 뒤, web page 를 이용하여 시스템을 사용 할 수 있다.

② postgres, python3 모두 native로 설치하였고, mac OS 환경에서 개발하였다.
(python 3.7.2, postgreSQL 11.5, MacOs 10.14.6)

③ 시스템을 사용하기 위해서는 flask, psycopg2, jinja2 의 패키지가 필요하다
(Flask 1.1.1, psycopg2 2.8.4, Jinja2 2.10.3 사용)

1. 프로그램 설명 및 사용방법 (서버 관리자)

① 시스템을 사용하기 위해 우선 postgres database 를 실행시켜 놓는다.

② 실행 환경에 따라 app.py 의 main 의 app.run 의 argument 를 다르게 준다.
(Native 에서 실행하기 때문에 debug = True argument 만 주었다.)

③ postgres에 연결하기 위한 도 환경에 따라 helper.py 의 db_connector 부분을 수정한다.
(Mac OS 에 native 에 설치 하였기 때문에
{ 'host': "localhost", 'user': "postgres", 'dbname': "postgres", } 로 설정)

④ postgres database 가 정상적으로 실행 되었다면 시스템을 실행시킬 수 있다. assignment2 폴더에서 app.py 를 실행시킨다.

⑤ app.py 가 정상적으로 실행 되면, http://127.0.0.1:5000/ 로 접속하여 시스템을 이용할 수 있다.

⑥ 최초로 시스템을 실행할 때 초기 연락처들을 받아와야 한다. 최초로 시스템을 실행한다면 contact.csv 를 같은 폴더에 두고 실행해야 한다.

⑦ http://127.0.0.1:5000/ 에 접속하면 시스템은 데이터베이스에서 연락처 테이블 contact 가 있는지 확인하고, 없다면 contact 테이블을 생성하고, contact.csv 에서 파일을 읽어와 저장한다. **(psycopg2 의 copy_from 사용)**

2. 프로그램 설명 및 사용방법 (웹 애플리케이션 이용자)

: 연락처 프로그램은 4개의 기능을 할 수 있다.

1) 연락처 검색 : 이름을 입력 받아 입력된 이름을 첫 부분에 포함하고 있는 모든 연락처의 이름과 전화번호를 출력한다. 이름을 입력하지 않으면 이름을 입력하라는 팝업이 나타난다.

2) 연락처 수정 : 연락처 검색 후, 이름과 전화번호 오른쪽에 수정하기 버튼을 사용 할 수 있다. 수정하기 버튼을 누르면 변경될 이름과 전화번호를 입력 받는 프롬프트가 나타난다. 두 항목 모두 입력해야 하고, 이때 전화번호는 11 자리의 숫자이고 처음은 010 으로 시작되어야 한다. **(정규 표현식 적용)**

3) 연락처 등록 : 이름과 전화번호를 입력 받아 연락처 시스템에 추가한다. 이름과 전화

번호 둘 다 입력해야 하고, 전화번호는 11자리의 숫자이고 처음은 010으로 시작되어야 한다. **(정규 표현식 적용)**

4) 연락처 삭제 : 이름을 입력 받아 해당하는 연락처를 삭제 한다. 이때 이름은 정확하게 전체 일치 되어야 한다.

3. 구현 방법, 함수 별 코드 설명

1) index.html

① body 설명: 연락처 검색/수정(id = search), 등록(id = register), 삭제(id = delete)를 위한 form 이 존재하고, 검색 결과를 출력하기 위한 table (id = search-result) 이 존재한다.

② head 부분에 jquery 와 javascript 를 사용하였다.

③ function submit (form, destination)
: register, delete 시에 이용하는 함수 이다. form 을 이용해 data 를 받아오고, destination 에 해당하는 url 로 이동한다. data 를 전송하기 전에 form 의 정보를 받아와 이름과 전화번호가 입력 되었는지, regex 를 이용하여 전화번호가 010으로 시작하는 11자리 숫자인지 확인한다. 위 조건들이 만족 되지 않으면 alert 를 이용하여 문제를 표시한다.
데이터 확인이 끝난 후에 ajax 를 이용하여 비동기적으로 data 를 서버로 보내고, 완료되었다면 search-result table 에 결과를 반영한다.

④ function submitAndPrint (form, destination)
: search 시에 이용하는 함수 이다. submit 함수와 같은 방법으로 data 를 확인하고, ajax 를 이용해 서버로 data 를 보낸다. 서버의 동작이 완료되면 search-result table 에 결과를 반영한다. 이때 table 의 각 row 마다 수정하기 button 을 추가하여 button 이 선택되면 modify 함수를 호출한다.

⑤ function modify(ctl)
: 수정하기를 선택한 row 의 data 를 받아와 각 행의 data 를 받아오고, 2개의 prompt이용하여 수정될 새 이름과 전화번호를 받아온다. register 때와 마찬가지로 각 data 가 잘 입력되었는지, regex 를 이용하여 전화번호가 010으로 시작하는 11자리 숫자인지 확인한다. 배열을 이용하여 서버로 data 를 보내고, 서버의 동작이 완료되면 search-result table 에 결과를 반영한다.

⑥ $(document).ready(()
: register, delete, search 의 form 이 submit 될 때마다 각 기능에 맞는 함수를 적절한 url 을 이용하여 실행 하고, 완료 시 완료 메시지를 출력한다.

2) app.py

① 각 url 에 따라 서버가 수행 할 함수들을 지정한다.

② /

: 시스템이 시작 될 때 접근되는 url 이다. 시스템의 첫 실행 일수 있으므로 helper.create_table() 을 이용해 database 에 contact table 이 있는지 확인하고 없다면 table 을 생성한다.

contact table 이 처음 생성 되었다면 helper.copy_csv() 를 이용해 contact.csv 에서 파일을 불러오고, helper.make_id() 를 이용해 table 의 각 element 에 id 를 부여한다.

동작이 완료되면 index.html 을 페이지에 출력한다.

③ /register

: web 으로부터 받아온 form data 를 이름과 전화번호로 나눈 뒤 helper.insert_address 를 이용해 연락처를 등록한다. 등록이 완료되면 / 를 반환하여 원래 페이지가 계속 보일 수 있도록 한다.

④ /search

: web 으로부터 받아온 form data 에서 이름을 뽑아 낸 뒤 helper.search_contact() 를 이용하여 연락처를 검색한다. 검색 후 data 를 web page 에 출력해야 하므로 함수의 수행 결과를 return 한다.

⑤ /delete

: web 으로부터 받아온 form data 에서 이름을 뽑아 낸 뒤 helper.delete_address 를 이용해 연락처를 삭제한다. 삭제가 완료되면 / 를 반환하여 원래 페이지가 계속 보일 수 있도록 한다.

⑥ /modify

: web 에서 modify 는 배열을 이용해 data 를 서버로 전달 하였다. request.args.get() 을 이용하여 기존 이름, 전화번호와 새로운 이름, 전화번호를 받아온다. 검색 후 data 를 web page 에 출력해야 하므로 함수의 수행 결과를 return 한다.

3) helper.py (자세한 sql 문은 code 를 참고해주세요. 보고서가 너무 길어질 것같아 생략했습니다.)

① db_connector, db_string

: Data base 를 이용하기 위한 연결 정보.

② create_table(table_name)

: table_name 에 해당하는 이름의 table 을 database 에서 확인하여 (pg_class 에 relname 이 존재하는지 확인하는 sql) table 이 존재하지 않는다면 table 을 생성한다. (Create table sql 사용) table 의 attribute 는 이름과 전화번호 이다. Sql 들은 cursor 를 이용해 실행하고, sql 실행이 모두 끝난 뒤 table 생성여부를 반환한다.

③ copy_csv(file_name, table_name)

: 같은 폴더에 있는 contact.csv 파일에서 데이터를 불러와 database contact table 에 저장한다. cur.copy_from(f, table_name, sep=',') 를 사용한다. (psycopg2 cursor 에 있는 method)

④ make_id (table_name)

: 주어진 table_name 에 해당하는 table 에 primary key 가 되는 id 를 추가해준다. ALTER TABLE sql 을 사용하고, SERIAL 을 이용하여 데이터가 추가 될 때 마다 입력되는 자동으로 key 를 증가시켜준다.

⑤ search_contact(name)

: SELECT sql 을 사용하여 이름이 주어진 이름 + %, 즉, 주어진 이름으로 시작되는 모든 이름을 찾아 반환한다. Sql 문에서는 ORDER BY 를 이용해 정렬하여 출력할 수 있다.

⑥ insert_contact(name, pnum)

: INSERT INTO sql 을 사용하여 주어진 이름과 전화번호로 contact table 의 entry 를 추가한다.

⑦ delete_adress(name)

: DELETE sql 을 사용하여 주어진 이름과 같은 이름을 가진 element를 삭제한다. 이때, 한번에 하나의 element 만 삭제해야 하므로 nested sql 문을 이용하여 하나의 element의 id 만 선택 한 뒤 삭제한다.

⑧ modify_address(name, pnum, newName, newPnum)

: 기존 이름과 전화번호를 이용하여 해당하는 table entry 를 '하나' 만 찾은 뒤, 새로운 이름과 전화번호로 그 table entry 를 수정한다. UPDATE sql 을 사용하고, 하나의 table entry 만 수정하기 위해 nested sql 을 사용하여 하나의 데이터만 선택한다.

4. 결과물 test

* 맥 환경에서 test 하였고, postgresSQL 11.5, python 3.7.2, 웹 브라우저는 safari 12.1.2 에서 test 하였습니다. (chrome 78.0.3904.97 에서도 정상 실행 확인)

① 데이터 베이스 실행, 연락처 관리 시스템 실행

```
gimhyeongsuns-MacBook-Pro:assignment2 Rdolf$ pg_ctl -D /usr/local/var/postgres start
waiting for server to start....2019-11-09 23:49:03.119 KST [30702] LOG:  listening on IPv6 address "::1", port 5432
2019-11-09 23:49:03.119 KST [30702] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2019-11-09 23:49:03.121 KST [30702] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5432"
2019-11-09 23:49:03.142 KST [30703] LOG:  database system was shut down at 2019-11-09 23:48:54 KST
2019-11-09 23:49:03.152 KST [30702] LOG:  database system is ready to accept connections
 done
server started
gimhyeongsuns-MacBook-Pro:assignment2 Rdolf$ ls
app.py                  contact.csv             pypg                    requirements.txt        templates
gimhyeongsuns-MacBook-Pro:assignment2 Rdolf$ python3 app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 756-889-915
```

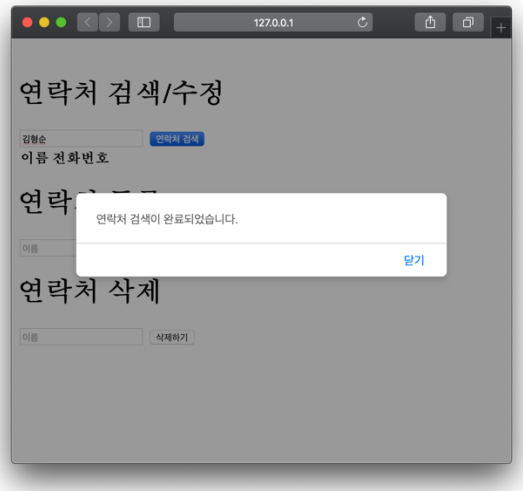
② http://127.0.0.1:5000/ 에 접속하여 연락처 관리 시스템 접속 및 contact table 생성, contact.csv upload.

```
postgres=> DROP TABLE contact;
DROP TABLE
postgres=> \dt
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | student  | table | postgres
 public | students | table | postgres
(2 rows)

postgres=>
```

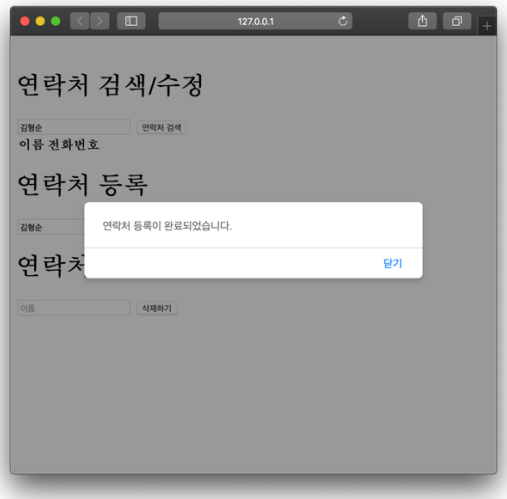
(시스템 실행 전)

③ 이름 김형순 검색 - 검색 결과가 없다는 것을 확인 하였다.



④ 초기 연락처에서 '김형' 검색
('김'으로 시작하는 이름은 너무 숫자가 많아 '김형'으로 검색 하였습니다.)

⑤ 김형순 연락처 삽입 – 검색 table 을 보면 연락처가 등록되었다는 것을 알 수 있다.

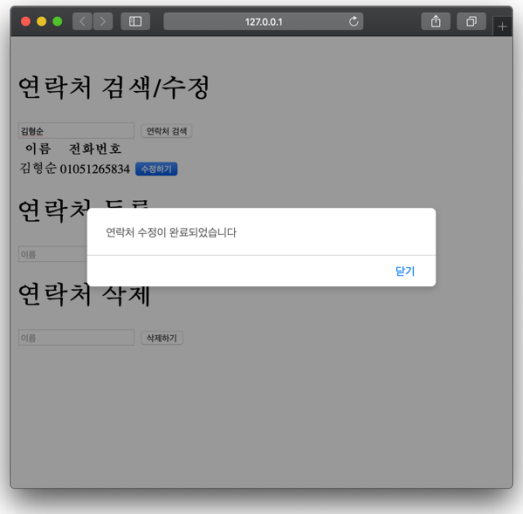


⑥ 시스템 종료 후 재실행

```
gimhyeongsuns-MacBook-Pro:assignment2 Rdolf$ python3 app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 756-889-915
127.0.0.1 - - [10/Nov/2019 00:09:18] "POST /search HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2019 00:09:20] "POST /search HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2019 00:09:32] "POST /register HTTP/1.1" 302 -
127.0.0.1 - - [10/Nov/2019 00:09:33] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2019 00:09:33] "POST /search HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2019 00:09:36] "POST /search HTTP/1.1" 200 -
^Cgimhyeongsuns-MacBook-Pro:assignment2 Rdolf$ python3 app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 756-889-915
```

⑦ '김형순' 검색 – data 가 사라지지 않고 남아있는 것을 확인 할 수 있었다.

⑧ '김형순' 연락처 -> '김형수'로 이름 및 전화번호 수정



수정 후 '김형순' 연락처는 검색되지 않았고, '김형수' 연락처가 수정된 연락처로 검색되었다

⑨ 프로그램 종료, 재실행

- 삽입 시와 마찬가지로 프로그램 종료, 재실행 후에도 연락처를 정상적으로 검색 할 수 있었다

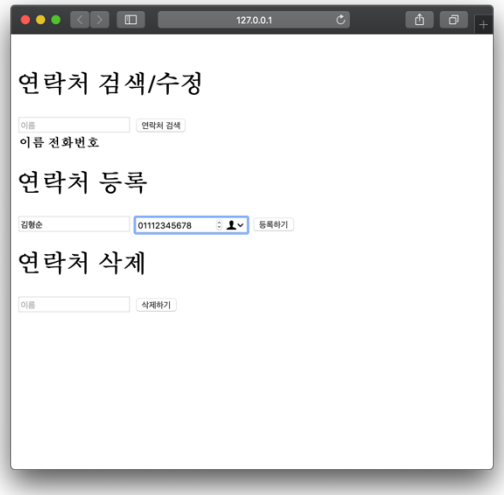
```
127.0.0.1 - - [10/Nov/2019 00:14:30] "POST /search HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2019 00:14:32] "POST /search HTTP/1.1" 200 -
^Cgimhyeongsuns-MacBook-Pro:assignment2 Rdolf$ python3 app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 756-889-915
```

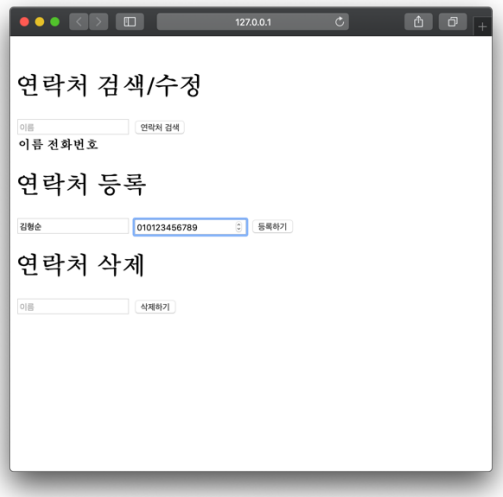
⑩ '김형수' 연락처 삭제

연락처를 삭제하고 검색하면 해당 연락처가 보이지 않는다는 것을 확인하였다.

⑪ 입력시 010으로 시작하는 11자리 전화번호 만 받는지 확인

011 로 시작하는 번호거나 뒷자리가 7자리여서 총 10자리 인 전화번호를 입력하면 입력에 실패하는 것을 확인 하였다.





** 과제 추가 수정

이제 연락처 검색 결과의 개수를 표시합니다. (이름 입력 창 위에 검색된 연락처 개수가 나옴)