데이터베이스 term project 보고서

2017130471 심리학부 최기성

* 벌금 걷기 프로젝트

1. 필요성

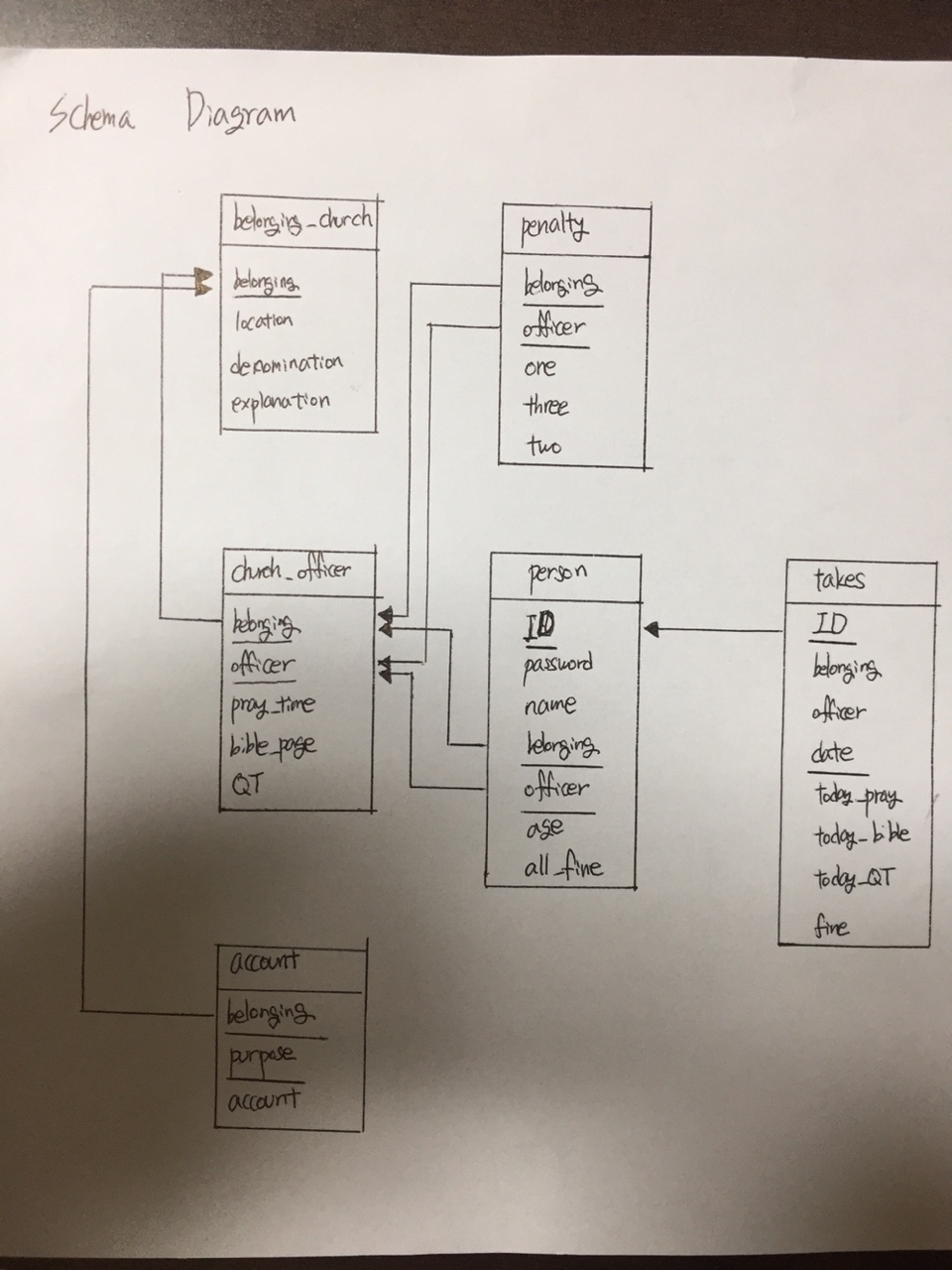
교회에서는 기도, 말씀, 큐티라는 것을 매일 하는 것으로 지향점으로 할 수 있기를 유도한다. 그러나, 강압적인 것이 아니다 보니 실제로 매일을 유지하기가 어렵다. 그렇기에 일정 기간 동안 꾸준히 실천한 인원들에게 선물을 주는 식으로 참여를 독려한다. 이런 방법 중 가장 많이 사용되는 방법은 ‘벌금 걷기’와 같은 방법이다. 실제로 벌금을 걷는 방식을 통해 무언의 압박을 느껴 참여율이 증가하는 경우가 많다.

이를 꾸준히 하기 위해서 본인이 실천한 내용들을 저장하고, 그에 따른 벌금 추산 방법이 다른다. 교회와 직분에 따라서도 달라진다. 이를 위해서 웹 데이터베이스 어플리케이션을 통해서 활용의 편리성을 증대할 수 있게 된다.

해당 애플리케이션을 구현하기 위해서 flask 와 psycopg2를 사용하였다. 또한 이때 데이터 베이스의 이름을 church로 한다.

1. 사용된 데이터 베이스를 위한 설명

* Schema diagram (사진 첨부)



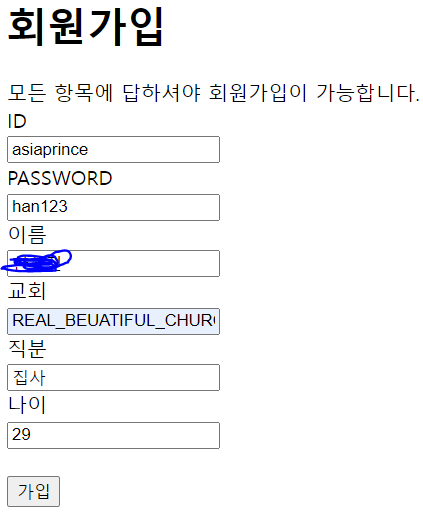
* 각 relation에 대한 설명
* Belonging\_church : 교회와 관련된 내용이 담긴 table로, belonging은 교회 이름, location은 교회 위치, denomination은 교파 이름, explanation은 교회의 방향성 및 부연 설명이 들어간다. Belonging 칼럼을 primary key로 활용한다.
* Church\_officer : 교회의 직분에 대한 내용이 담긴 table로, belonging은 교회 이름, officer은 직분 이름, pray\_time은 해당 직분에서 매일 요구되는 기도 시간, bible\_page는 해당 직분에서 매일 요구되는 성경 읽는 장수, QT는 Boolean 값으로 true와 false로 실행했는지 유무만 확인한다. Belonging은 belonging\_church의 primary key인 belonging 칼럼을 외래키로 설정하였고, belonging과 officer 칼럼을 primary key로 설정한다.
* Person : 현재 프로그램을 사용 중인 인원을 설명하는 테이블이다. ID와 PASSWORD는 사용자를 식별하여 로그인이 가능할 수 있도록 해준다. Name은 사용자의 이름, belonging은 속한 교회 이름, officer은 사용자의 직분, age는 사용자의 나이, all\_fine은 사용자가 아직 지불하지 않은 벌금을 의미한다. Belonging과 officer은 church\_officer 테이블의 belonging과 officer 칼럼을 참조한다. ID, belonging, officer 칼럼을 primary key로 설정한다.
* Takes : 사용자가 매일 실천한 내용들에 대해서 나타낸다. ID는 사용자를 식별할 수 있게 해주는 역할을 하고, belonging은 사용자가 속해있는 교회, officer는 사용자의 직분, date는 사용자가 실천한 날짜, today\_pray는 해당 날짜에 기도한 시간, today\_bible은 해당 날짜에 읽는 성경 장수, today\_QT는 해당 날짜에 큐티를 했는지에 대한 유무, fine은 해당 날짜에 실천한 내용에 따른 벌금이 나타난다. ID는 person 테이블의 ID를 참조하고 있으며, ID, date를 primary key로 설정한다.
* Penalty : 각 직분에 따른 벌금 산정 기준을 결정한다. Belonging과 officer는 각각 교회와 직분을 의미하고, 각 직분에 따라서 기도.말씀.큐티 중 요구되는 수준보다 미달이 된 종목의 개수에 따라 one, two, three라는 칼럼으로 나뉠 수 있다. 각 칼럼에는 미달이 된 종목의 개수에 따른 벌금이 나타난다. Belonging과 officerd은 church\_officer의 belonging과 officer 칼럼을 참조하고 있으며, primary key로 설정한다.
* Account : 각 교회에서 사용하고 있는 계좌를 의미한다. Belonging은 교회, purpose는 계좌의 목적, accout는 계좌 번호이다. Belonging은 belonging\_church의 belonging 칼럼을 참조하며, belonging과 purpose 칼럼을 primary key로 설정한다.

1. 페이지 구성과 기능 소개
   1. 첫 화면

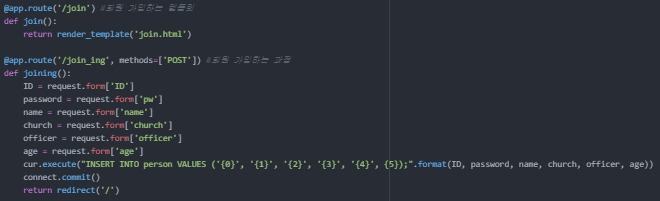


첫 화면으로, 로그인과 회원가입, 다시 첫 화면으로 돌아올 수 있는 항목으로 설정했다.

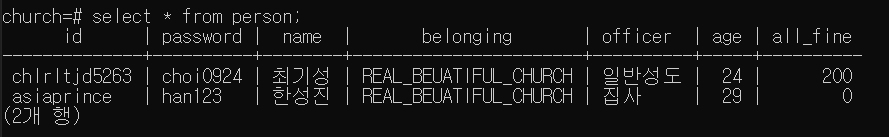
* 1. 회원가입



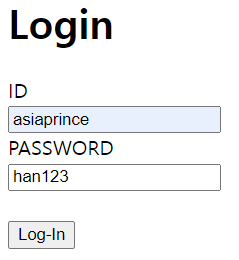
회원 가입을 할 수 있는 페이지로, 해당 항목들에 맞게 기입을 한 이후, 가입 버튼을 눌러서 person 테이블에 튜플을 insert할 수 있게 한다.



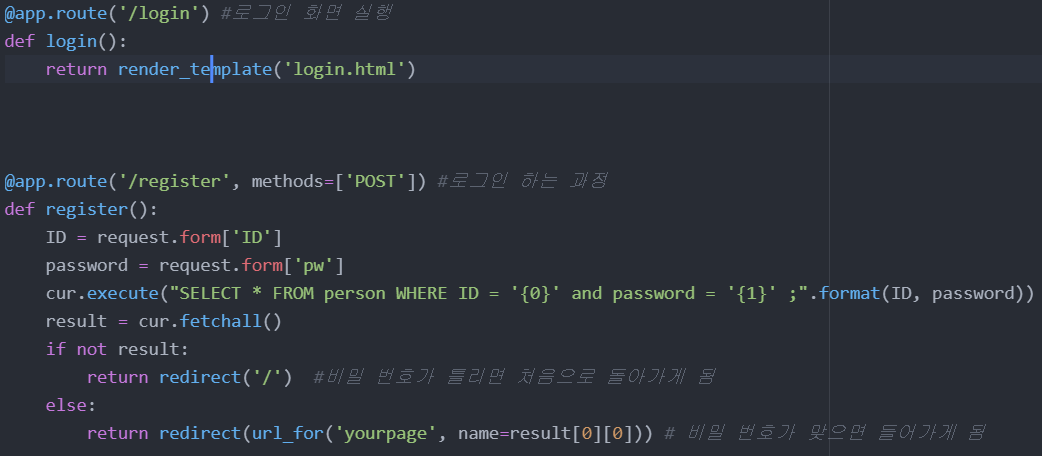
이때 회원 가입 페이지에 들어가게 되면 ‘/join’ 페이지에 접근하게 되고, 해당 페이지에서 가입 버튼을 누르면 ‘/join\_ing’ 으로 연결되어 joining이라는 함수가 실행된다. 이때 가입 버튼을 누르게 된다면 insert sql이 발생하여 person 테이블에 튜플을 insert할 수 있게 된다. 그리고 회원 가입이 된 이후에 즉시 처음 화면으로 돌아가게 된다.



* 기입한 내용대로 person 테이블에 추가가 된 것을 확인할 수 있다.
  1. 로그인

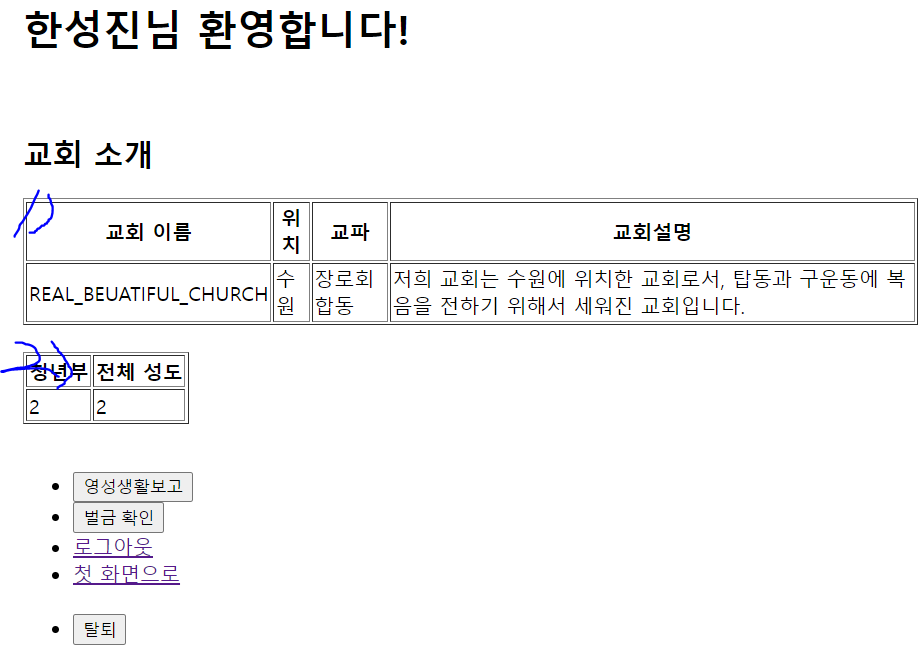


ID와 PASSWORD 값을 기입할 수 있는 LOGIN페이지이다.

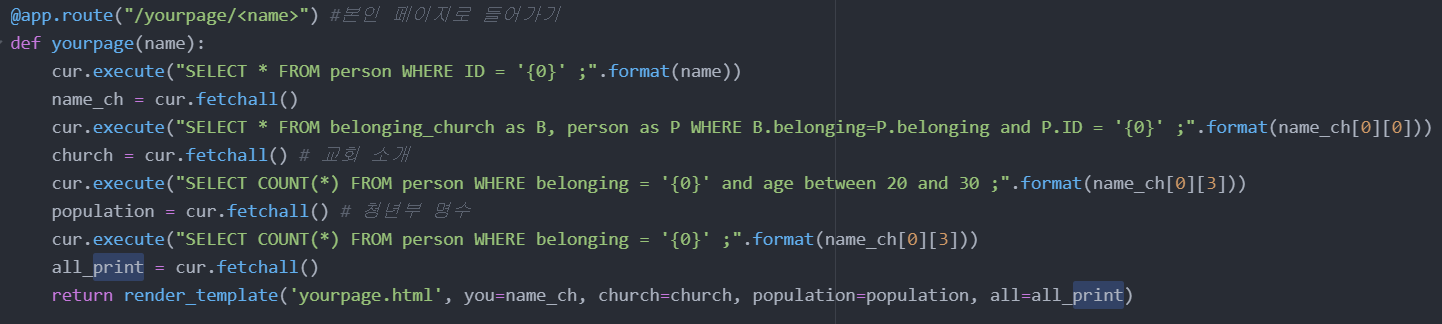


첫 화면에서 로그인 화면을 누르면 /login으로 이동하여 login.html 템플릿이 렌더링이 되고, LOG-IN버튼을 누른 이후에 /register로 이동하여 person 테이블에서 ID와 PASSWORD의 값을 통해 SELECT SQL을 통하여 원하는 튜플을 가져올 수 있다. 만약 잘못 기입하여 튜플을 가져올 수 없었던 경우에는 첫 화면으로 돌아가고, 올바르게 기입하여 튜플을 가져온 경우에는 개인 홈페이지로 이동시킨다.

2-1) 개인 홈페이지



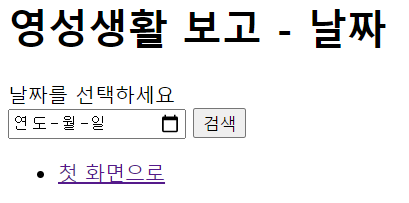
개인 홈페이지이다. 1)을 통해서 개인이 다니고 있는 교회의 대한 정보가 나타나고, 2)를 통해서 속해 있는 교회의 명부 수를 나타낸다. 밑의 나타나는 버튼을 통해서 위에 나타난 실천사항을 보고하고, 개인의 벌금을 확인할 수 있게 된다. 또한 로그아웃을 통해서 첫 화면으로 돌아갈 수 있으며, ‘첫 화면으로’를 통해서 본인의 개인 홈페이지에 들어올 수 있다. 마지막으로 탈퇴 버튼을 통해서 person 테이블에 개인에 대한 정보를 데이터베이스에서 삭제할 수 있다.



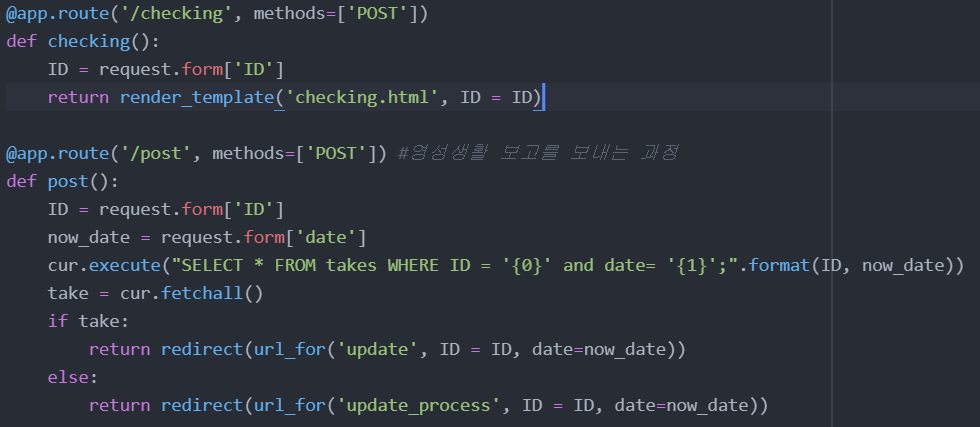
이때 로그인 때 받았던 정보인 name을 인자와 함께 yourpage라는 함수를 실행한다. 이때 두번째 select문을 통해서 Cartesian 곱을 활용하여 교회와 사람에 대한 테이블을 잇고, name을 통해 본인이 다니는 교회에 관련한 정보를 얻을 수 있다. 이는 위 화면의 1)과 연결이 된다. 세번째와 네번째 쿼리는 count(\*)를 통해서 각각 청년부의 인원(20<age<30인 조건을 만족하는)과 전체 인원을 얻어낸다. 이는 위 화면의 2)를 나타낼 수 있게 한다.

2-1) 영성생활 보고

* 첫번째 페이지

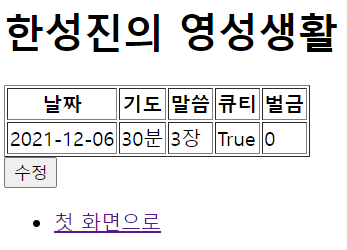


개인 홈페이지에서 ‘영성생활 보고’ 버튼을 누르면 나타나는 화면으로 날짜를 선택할 수 있게 한다.

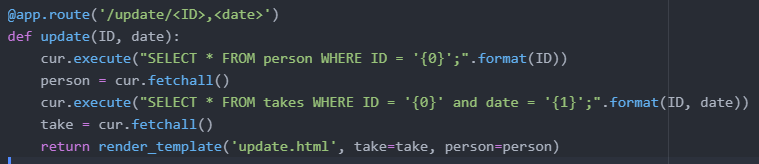


이때 checking 이라는 함수가 실행하였고, 개인 홈페이지에서 전송된 ID값을 가져오게 된다. 해당 날짜를 선택해서 검색을 할 경우에 가져왔던 ID값과 선택한 날짜 값이 /post로 이동하여 post함수가 실행이 된다. SELECT 쿼리를 활용하여 가져온 정보들을 통해서 이미 보고를 했던 기록이 있는지를 확인한다. 보고를 했던 정보가 있는 경우에는 /update로 이동하고, 정보가 없는 경우에는 /update\_process로 이동하게 된다.

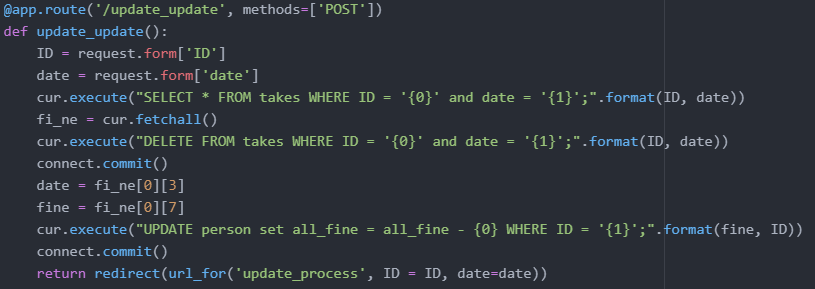
* 보고된 정보가 있었던 경우



해당 날짜에 보고되었던 기록이 있다면, 표 형식을 통해서 어떻게 보고를 했는지를 확인할 수 있다. 이때 수정 버튼을 통해서 해당 테이블을 수정할 수 있도록 한다.

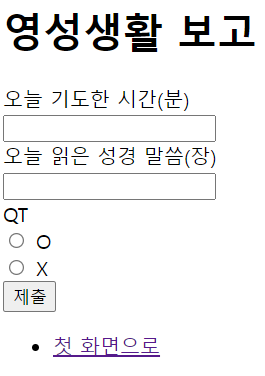


이전 페이지에서 받아온 ID와 DATE의 값을 통해서 takes 테이블에 해당하는 튜플을 찾는다.

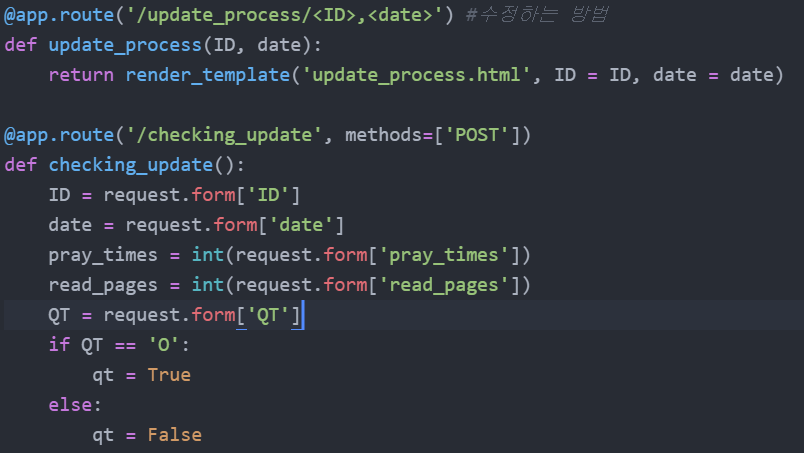


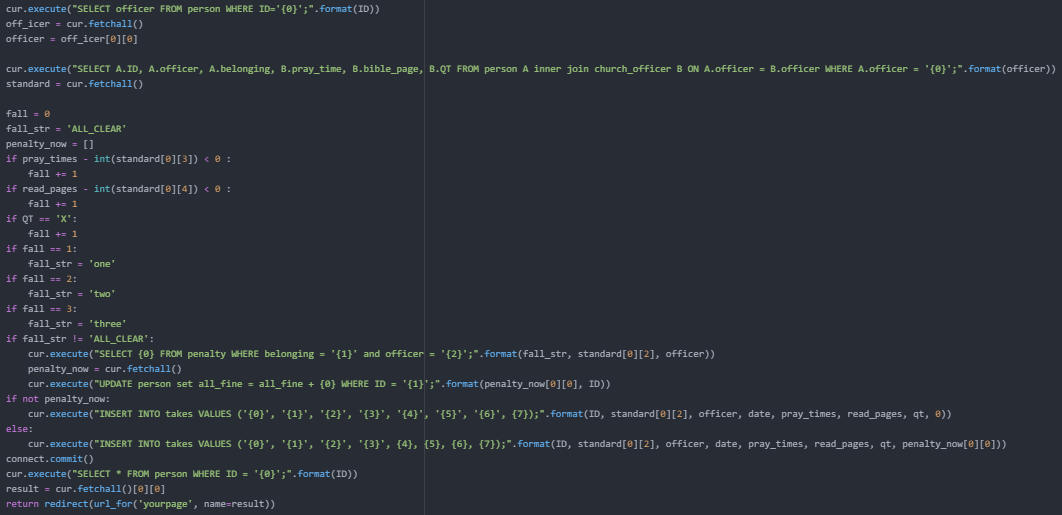
만약 수정을 누른 경우에는, DELETE 쿼리를 통해서 takes 테이블에 존재하던 튜플을 삭제하고, UPDATE 쿼리를 통해서 person 테이블에 존재하는 총 벌금의 값을 감해주도록 업데이트 해주는 과정을 거친다. 그 이후에 /update\_process 로 이동시킨다.

* 보고된 경우가 없거나, 수정 버튼을 누른 경우



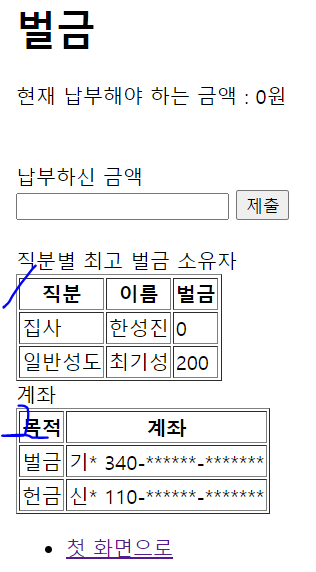
보고된 정보가 존재하지 않거나, 새롭게 보고를 할 수 있는 형식이 주어진다. 기입 후 제출 버튼을 누르면 takes 테이블에 저장이 된다.



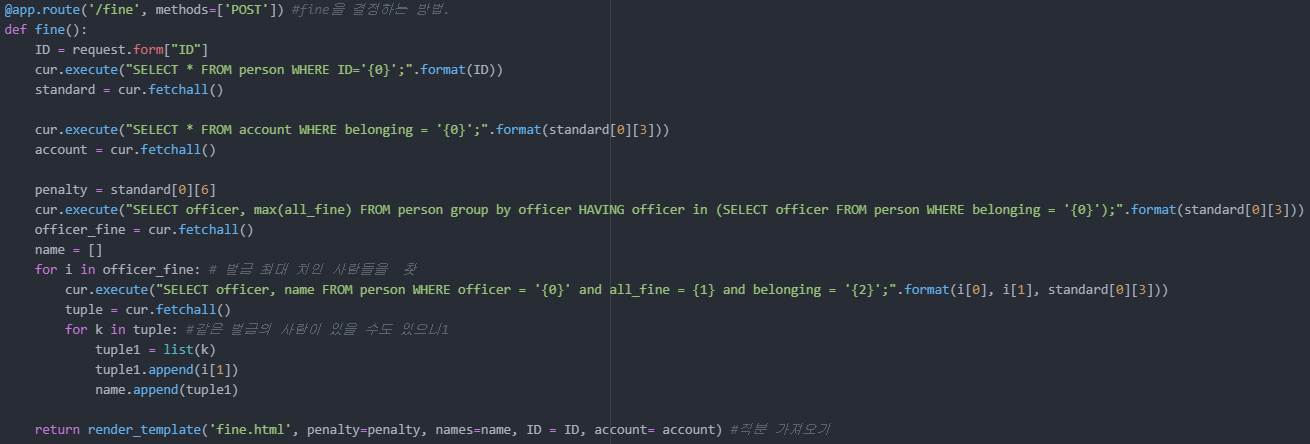


제출 버튼을 누르게 되면 checking\_update 함수가 실행이 된다. 두번째 사진에 나타나는 첫번째 select 쿼리를 join을 활용하여 각 직분에 맞는 영성생활 수준의 정보와 각 개인의 정보를 연결하게 된다. 이 정보를 통해서 본인이 기입한 정보(post를 통해서 얻은 값들)들과 비교하여 미달된 개수를 확인하게 된다. 이때 insert 쿼리를 통해서 takes 테이블에 새로운 값을 넣어주고, person 테이블에도 벌금을 내야 하는 상황이면, 벌금을 추가해주기도 한다. 그리고 다시 개인 홈페이지로 넘어오게 된다.

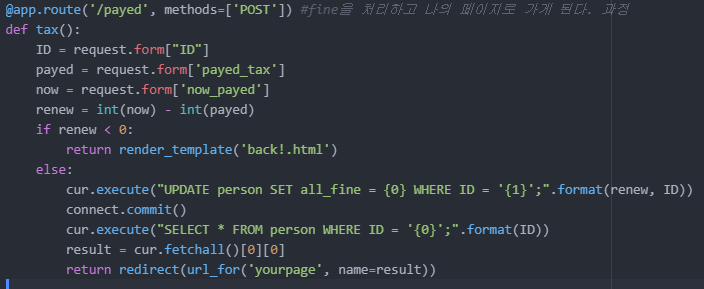
2-2) 벌금 납부



개인 홈페이지에서 벌금 확인 버튼을 누르게 되면 해당 페이지가 나타나게 된다. 맨 위에는 납부해야 하는 금액이 적혀 있고, 다음 칸은 납부한 이후의 금액을 작성해서 제출을 하면 person테이블에 존재하는 개인의 벌금이 줄어들게 된다. 1)은 속해있는 교회의 직분에 따라서 최고 벌금을 소유하고 있는 사람이 나타나게 된다. 2)는 해당 교회에서 갖고 있는 계좌의 정보가 나타난다.



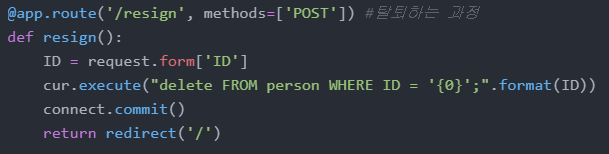
두번째 select 쿼리를 활용하여 account라는 변수에 페이지에 계좌의 정보가 나타나게 하였고, 세번째 select 쿼리는 서브 쿼리를 활용하여 해당 교회에 존재하고 직분에 해당하는 사람이 존재할 때 해당 직분들에 따라서 최고 벌금을 가진 자들을 찾을 수 있도록 했다. 밑의 for 문을 통해서 최고 벌금을 가진 사람의 이름을 가져오고, 최대 벌금을 가진 자들이 여러 명일 때를 대비한 코드이다.



또한 제출 버튼을 누르게 되면 tax함수가 실행하여 update 쿼리를 활용하여 지불한 벌금에 맞게 업데이트할 수 있도록 한다.

3-3) 탈퇴

개인 홈페이지를 통해서 탈퇴 버튼을 눌러서 person 테이블에 해당하는 자신의 기록을 없앨 수 있다. 이때 개인에 해당하는 takes의 값도 동일하게 cascade를 통해서 삭제 된다.



Delete 쿼리를 통해서 person 테이블에 저장되어 있는 개인과 관련된 정보가 삭제가 되고, 맨 처음 화면으로 돌아가게 된다.