

효율적인 자격증 애플리케이션을 위한 데이터베이스 설계에 관한 연구

진아현*, 이덕규*

*서원대학교 정보보안학과

0720dkfusdlchlrh@naver.com, deokgyulee@seowon.ac.kr

A Study on Database Design for Efficient Data Filter for Certification Application

Ah hyun Jin *, Deok Gyu Lee*

*Dept. of Information Security, Seowon University

요 약

자격증 준비는 필수적 요소라고 볼 수 있다. 하지만 자격증을 처음 접할 때 다수의 자격증으로 인해 사용자가 원하는 자격증을 확인하지 못하는 경우가 많고, 자격증 일정을 따로 찾아봐야 하는 번거로움이 있다. 자격증 애플리케이션은 잘 알려지지 않은 민간자격증이나 신규자격증 등 관련자격증 정보를 제공하는 자격증 애플리케이션에 데이터 필터를 구축하고, 사용자에게 최적의 서비스와 데이터를 제공할 수 있도록 함이 목적이다. 어떻게 효율적으로 데이터베이스를 설계하는지와 IDE에서의 데이터베이스 Connect 코드에 초점을 두고 고찰 해보고자 한다.

데 있어 시간을 절약할 수 있다.

1. 서론

1-1 서비스 분야 애플리케이션 흐름

IT산업 기술이 발전함에 따라 스마트폰 역시 나날이 발전하고 있다. 사람들에게 더 빠른 정보와 효율적인 일상생활을 위해 애플리케이션 또한 다양한 종류가 개발되고 있다.

1-2 자격증 애플리케이션

처음 자격증을 접할 때 자격증의 종류가 여러 가지이기 때문에 원하는 분야의 자격증 정보를 찾기에 많은 시간이 소요된다.

또한, 자격증의 일정을 따로 찾아봐야 하는 번거로움이 있다. 이러한 불편한 점을 보완하고자 자격증 애플리케이션은 자격증에 관한 여러 정보를 한 곳에 모아 놓아 누구나 손쉽게 정보를 빠르고 정확하게 얻을 수 있는 서비스를 제공하고 국가, 민간 필터를 통해 자격증 관련 정보를 찾는 번거로움을 줄이고 최적의 서비스를 사용자에게 제공하려고 한다.

사용자는 이러한 자격증 애플리케이션의 서비스와 직관적인 인터페이스를 통해 자격증 일정 등을 포함한 분산 되어있는 정보를 한눈에 확인할 수 있다. 자격증 취득 및 공부를 하기 위한 계획을 구성하는

2. 본론

2-1 구성도



[그림1] 구성도

데이터베이스와 애플리케이션은 서로 직접 통신할 수 없다. 중간 매개체가 필요한데 PHP 즉, 웹 서버가 그 역할을 담당하게 된다. 웹 서버와 데이터베이스를 결합하고 관리가 수월한 XAMPP 응용 소프트웨어를 활용한다.

2-2 PHP

PHP는 “Hypertext Preprocessor”의 약자로 1994년 라스머스 레돌프가 처음 선을 보인 서버 측 웹 프로그래밍언어이다.

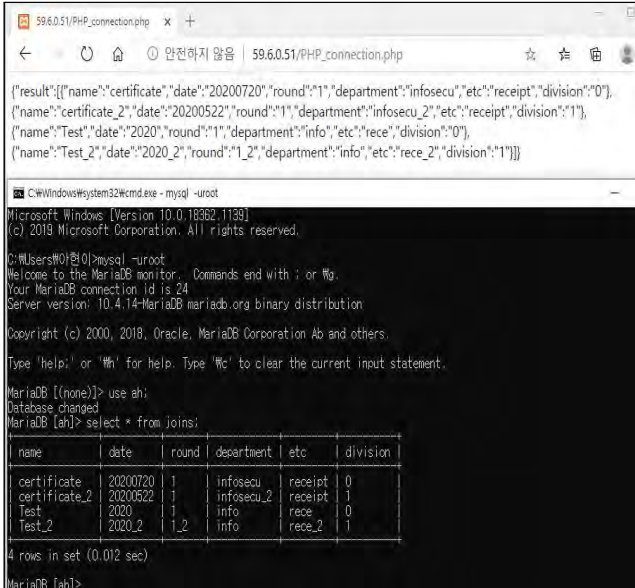
2-3 Database MySQL

데이터베이스에 모아 놓은 데이터들의 저장, 읽기, 검색 등의 처리를 위해서는 전용프로그램이 필요하게 되는데 이것을 데이터베이스 관리 시스템 즉,

DBMS(Database Management System)라 한다. 즉 사용자의 응용프로그램과 데이터의 저장장소인 데이터베이스를 연결해주는 프로그램이라 할 수 있다.

MySQL은 일반 사용자가 무료로 사용할 수 있는 DBMS 중의 하나이며, 기종과 운영체제와 관계없이 활용할 수 있고 보안성이 뛰어나다. 또한, 대용량의 데이터를 빠른 속도로 손쉽게 처리할 수 있으며, 설치가 쉽고 사용이 쉬우므로 초보자도 이른 시일 내에 사용법을 익힐 수 있다. [1]

2-4 PHP와 Database 연동



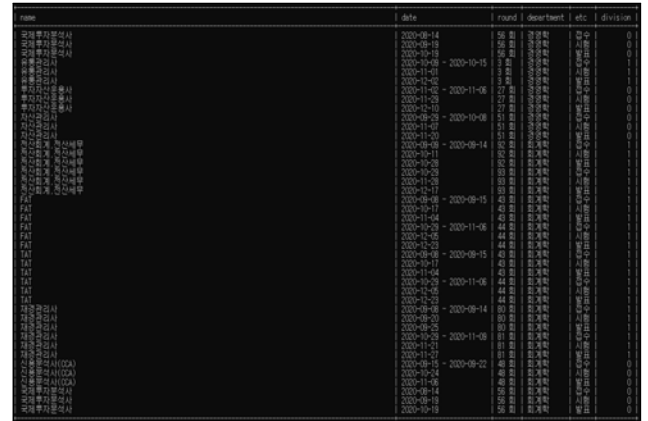
[그림2] 연동 성공 화면 PHP(상), Mysql(하)

[그림 2]는 SQL 언어로 작성한 테이블 속성값이 정상적으로 웹 서버에 연동되어 출력되는 모습이다. 이후 애플리케이션 내부 코드가 웹 서버의 데이터를 참고하여 사용자에게 데이터를 제공해줄 수 있다. 또한, PHP에서 제공해주는 서비스로 MySQL Administrator 페이지를 통해 GUI 기반 데이터베이스

를 관리할 수 있고 오류 Log 및 수많은 Status를 확인하고 수정할 수 있다.

PHP로 제작된 웹 서버는 데이터베이스와 자격증 애플리케이션의 데이터 통신의 매개체가 됨으로써 데이터베이스 및 애플리케이션의 독자적인 개발에 힘을 쓸 수 있게 된다.

2-5 Databeas 설계



[그림3] Database 속성 분류

데이터베이스 속성들은 다음과 같이 구성되어 있다.

- [1] 자격증 명을 표시할 Name 속성
- [2] 자격증 접수, 필기시험, 발표날 등의 날짜를 저장하고 있는 date 속성
- [3] 자격증 시험의 회차를 저장하는 round 속성
- [4] 표시할 자격증에 연관된 학과를 저장하고 있는 department 속성
- [5] 표시하는 해당 값들이 접수, 시험, 발표날인지를 알려주기 위한 etc 속성
- [6] 표시하는 자격증이 민간(0)자격증인지, 국가(1)자격증인지 분류하기 위한 임의 값을 저장하고 있는 division 속성

필요한 정보만 정확하고 빠르게 확인할 수 있도록 체계적인 속성값들과 필터링 기능을 위한 임의값을 지정해주어 효율적인 데이터베이스 설계가 가능하다.

자격증 애플리케이션은 학과 또는 기술별로 자격증 정보를 제공한다. 데이터베이스 내 테이블 또한

학과 또는 기술별로 만들어서 각각의 테이블에 자격증 데이터를 저장할 수 있지만 본 논문에서는 하나의 테이블 내 모든 자격증 정보를 저장하고 임의 값을 저장하는 속성과 필터링 기능을 위한 코드만으로 학과 또는 기술별로 필터링 기능을 제공한다. 일련의 적은 처리 과정만으로 데이터베이스에서 하나의 테이블만 관리할 수 있게 되므로 효율적인 데이터베이스 설계가 실현된다.

2-6 DB Connect 코드

```
public void getData(String uri) {
    class GetDataJSON extends AsyncTask<String, Void, String> {
        @Override
        protected String doInBackground(String... params) {
            String uri = params[0];
            Log.d("joia", "msg 'uri' = " + uri);
            BufferedReader bufferedReader = null;
            try {
                URL url = new URL(uri);
                HttpURLConnection con = (HttpURLConnection) url.openConnection();
                if (con != null) {
                    Log.d("joia", "msg '연결 성공'");
                } else {
                    Log.d("joia", "msg '연결 실패'");
                }
                StringBuilder sb = new StringBuilder();
                bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));
                String json;
                while ((json = bufferedReader.readLine()) != null) {
                    sb.append(json + "\n");
                    Log.d("joia", "msg 'json = " + json);
                }
                return sb.toString().trim();
            } catch (Exception e) {
                Log.e("joia", "msg '데이터 못가져옴 ' + e);
                return null;
            }
        }
    }
}
```

[그림4] Get Date 코드

```
protected void showList() {
    try {
        Log.d("joia", "msg 'myJSON = " + myJSON);
        JSONObject jsonObj = new JSONObject(myJSON);
        JSONObject peoples = jsonObj.getJSONObject(TAG_RESULTS);
        for (int i = 0; i < peoples.length(); i++) {
            JSONObject c = peoples.getJSONObject(i);
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String address = c.getString(TAG_ADDR);
            HashMap<String, String> persons = new HashMap<>();
            persons.put(TAG_ID, id);
            persons.put(TAG_NAME, name);
            persons.put(TAG_ADDR, address);
            personList.add(persons);
        }
        ListAdapter adapter = new SimpleListAdapter(
            MainActivity.this, personList, R.layout.item,
            new String[]{TAG_ID, TAG_NAME, TAG_ADDR},
            new int[]{R.id.id, R.id.name, R.id.address}
        );
        list.setAdapter(adapter);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

[그림5] Show List 코드

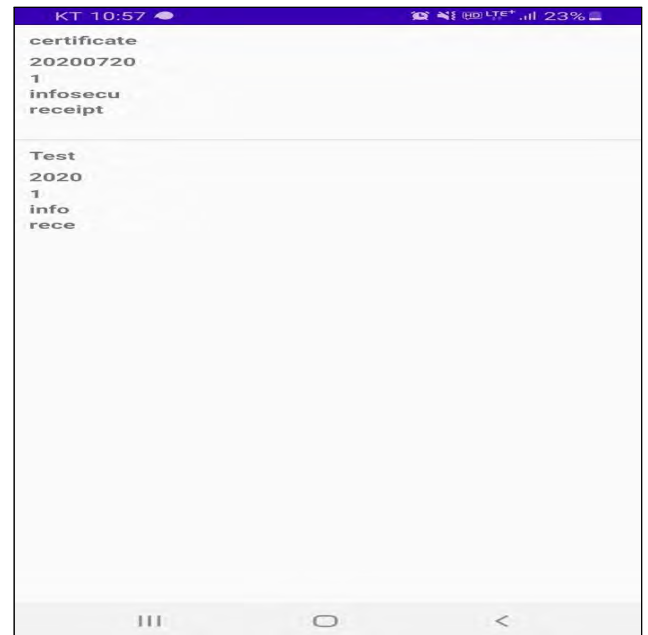
[그림4]는 PHP로 구축한 웹 서버로부터 데이터를 가져온다. 데이터베이스는 웹 서버에 데이터값을 올리게 되는데, 웹 서버의 외부 IP주소와 PHP 파일을 결합한 URL 정보를 통해 데이터를 가져오는 원리다.

[그림5]는 [그림4]의 처리 과정을 통해 가져온 데이터를 하나씩 배열에 담아, 구성된 애플리케이션

레이아웃에 맞게 데이터를 변환하고 사용자에게 정해진 데이터 값을 보여줄 수 있도록 처리한다.

지속적인 코드의 개발을 위해 변수의 역할을 헛갈리지 않도록 직관적이고 누구나 알아보기 쉬운 변수명으로 지정하였다. 객체를 유연하게 사용하고 재사용성 쉽게 가능하도록 함수 또는 역할별로 객체를 분류하였다.

3. 결론



[그림6] 애플리케이션 UI

PHP 웹 서버, 데이터베이스, 안드로이드 프로그래밍이 적절하게 설계되어 정상적으로 애플리케이션에 데이터베이스의 데이터가 출력되는 화면이다. 필터링 기능을 위한 필터 기능을 하는 버튼과 데이터베이스 임의값을 저장하고 있는 division 속성을 통해 많은 데이터베이스 데이터값들을 사용자의 요구에 따라 빠르게 정보를 찾아볼 수 있도록 개발 중이다. 향후 자격증 정보의 양이 점점 많아짐에 따라 효율적인 데이터 처리 방식을 고안하고 필터링 기술을 고도화할 것이다. 또한, 현재 연도별 자격증 정보를 제공해주고 있으나 연도가 바뀔 때마다 많은 양의 데이터를 수정해야 하는 수고가 있으므로 스크립트 로직 개발이 필요할 것 같다.

참고문헌

- [1] (전용주, 채경전, 최현중, 김태영), PHP와 MySQL을 활용한 영재관찰추진 웹 시스템 개발 및 적용 p4-4, 학술논문, 2013.02
- [2] (한새론, 윤속은, 김영지, 최관순, 임중식, 김동식, 전창완, 전홍구), PHP 학습 시스템 구현 p2-2, 한국정보기술학회, 2011.05