

MySQL과 JSP를 기반으로 한 관측자료의 데이터베이스화
CONSTRUCTION OF ASTRONOMICAL DATABASE BASED ON MySQL AND JSP

성현일, 김상철, 임인성, 김봉규, 안영숙, 남현웅, 손상모, 양홍진
한국천문연구원

HYUN-IL SUNG, SANG CHUL KIM, IN SUNG YIM, BONG GYU KIM, YOUNG SUK AHN,
HYUN-WOONG NAM, SANGMO TONY SOHN, AND HONG-JIN YANG
Korea Astronomy Observatory

(Received November 11, 2004; Accepted November 29, 2004)

ABSTRACT

As a core project of the Korean Astronomical Data Center (KADC) in Korea Astronomy Observatory (KAO), we have constructed a framework of astronomical database using MySQL and JSP. MySQL database server is a powerful open source database management system, and JSP technology provides a simplified, fast way to create dynamic web content. Combined together, MySQL and JSP enable us to develop server-independent web-based applications efficiently in a short amount of time. The MySQL tables and JSP programs embedded in our database are built/written in a way to benefit both users and managers. We plan to expand our database by supplementing additional data in the near future. Our newly constructed database is expected to be the prototype of the Korean Virtual Observatory (KVO).

Keywords: astronomical databases

1. 서론

우리나라의 광학관측연구는 1978년의 소백산천문대 준공과 함께 본격적으로 시작되어 1996년의 보현산천문대 건립 이후 더욱 활발하게 이루어졌다. 그동안 많은 관측이 이루어졌으나 이 관측자료들은 관측제안을 한 연구자들에 의해서만 이용되어져 왔다. 현재, 외국의 대부분 천문대들은 관측자료를 데이터베이스화하여 공개하고 있으며 가상천문대(김상철 등 2003)를 구축하여 자료의 활용도를 극대화시키려는 노력 중에 있다. 우리나라에서도 관측자료의 공개와 데이터베이스화의 필요성이 제시되었으며 이에 한국천문연구원의 천문우주정보센터(Korean Astronomical Data Center, KADC, <http://kadc.kao.re.kr>)에서는 관측자료의 데이터베이스화에 대한 연구를 수행하게 되었다.

관측자료의 데이터베이스화는 소백산천문대와 보현산천문대에서 지금까지 관측된 많은 자료들(경제만 등 2002; 박병곤 & 박윤희 2002)을 체계적으로 정리하여 연구자들이 공유할 수 있도록 함으로써 연구의 보조자 역할을 하는 동시에 자료의 활용도를 높이려는데 목적이 있다(성현일 등 2003). 본 연구에서 확립된, 데이터베이스 구축을 위한 시스템은 국내의 천문대에서 관측된 자료뿐만 아니

라 미국 아리조나주의 레몬산천문대에서 관측된 자료들을 데이터베이스화하는데도 활용될 것이다. 이 연구는 한국 가상천문대(Korean Virtual Observatory, KVO, <http://kvo.kao.re.kr>)의 구축을 위한 초기단계의 연구활동에 해당한다.

이 연구를 위해 외국 데이터베이스의 미러링에 관하여 조사하고 실제로 미러사이트를 구축함으로써 데이터베이스의 기초기술을 미리 확보하였다(성현일 등 2002). 국내의 천문대에서 확보된 관측자료를 각각 데이터베이스로 구축하는 것이 일차적인 목표이고, 더 나아가 국내의 모든 관측자료를 하나의 데이터베이스로 묶어 한국가상천문대를 구축하는 것이 이차적 목표이다. 향후에는 국제가상천문대연맹(International Virtual Observatory Alliance, IVOA, <http://www.ivoa.net>)과 연계하여 국제표준에 따른 데이터베이스화를 추진할 계획이다(Sung et al. 2004).

2장에서는 MySQL과 JSP의 특징 및 동작과정 등에 관하여 설명하고, 3장에서는 데이터베이스를 구성하고 있는 JSP 프로그램과 MySQL 테이블에 관하여, 4장에서는 JSP 프로그램의 기능별 프로그래밍 방법에 관하여 다루며, 5장에서는 본 연구에서 확립된 데이터베이스 구축 시스템

을 기반으로 실제 DB 구축에 응용한 사례를 , 6장에서는 결론을 논하고자 한다.

2. MySQL과 JSP

2.1. MySQL

데이터베이스를 관리하는 데이터베이스 시스템은 크게 다음과 같은 요소들로 구성되고 유지·관리 된다: 스키마, 데이터베이스, 데이터베이스 언어, 데이터베이스 사용자, 데이터베이스 관리 시스템(Database Management System, DBMS). 이 중 DBMS는 응용 프로그램의 데이터 증재자로서 모든 응용 프로그램들이 데이터베이스에 접근하여 공용할 수 있도록 관리해 주는 소프트웨어 시스템이다(이춘식 2002). DBMS의 운영에는 데이터베이스의 생성과 갱신, 삭제뿐만 아니라 사용자의 질의에 응답하고 효율적으로 관리하는 것까지 포함된다. 잘 알려진 DBMS로는 Oracle, Infomix, Sybase 등이 있으며 무료로 사용할 수 있는 공개 DBMS로는 MySQL, PostgreSQL, MSQL 등이 있다(Thomson & Welling 2001; Welling & Thomson 2003).

MySQL은 관계형 데이터베이스 관리 시스템인 SQL (Structured Query Language)의 하나로, 사용자는 유틸리티를 이용하여 테이블 구조로 저장되어 있는 MySQL 데이터베이스의 데이터들에 접근할 수 있다(Kimula & Takahashi 2002). MySQL은 UNIX와 Solaris, AIX, FreeBSD, Linux, Windows 등 다양한 플랫폼에서 사용 가능하다(조준익 2000). MySQL의 수행구조는 그림 1과 같다(김호 2003; 허정수 2000)

2.1.1. 특징

MySQL의 특징 중 주요한 것은 다음과 같다: 1) Multi thread를 지원하므로, CPU가 여러 개일 경우 이 CPU들을 잘 활용할 수 있다. 2) 다양한 플랫폼을 지원한다. 3) 큰 데이터베이스도 다룰 수 있다. 4) 다양한 나라의 언어로 메시지를 표시한다. 5) 최적화된 라이브러리를 사용한다. 6) 속도가 빠르다. MySQL의 가장 큰 특징은 빠른 속도로, 시스템의 자원을 많이 소모하는 ANSI SQL을 구현하지 않으므로 속도가 향상된다.

2.1.2. 버전

MySQL 데이터베이스 서버는 네 가지 버전이 있다: 1) MySQL Standard 버전은 표준 MySQL 스토리지 엔진과 InnoDB 스토리지 엔진을 포함하고 있다. InnoDB는 안전한 트랜잭션과, ACID를 지원하는 스토리지 엔진으로 commit, rollback, crash recovery 그리고 row-level locking

기능을 제공한다. 2) MySQL Max 버전은 표준 MySQL 스토리지 엔진과 InnoDB 스토리지 엔진, 그리고 Berkeley 테

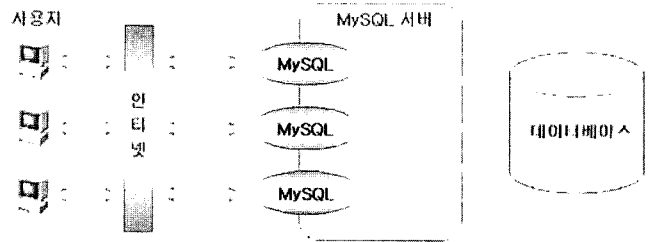


그림 1. MySQL의 수행구조.

이터베이스(BDB) 스토리지 엔진, SSL transport-layer encryption, 다중 파일로의 테이블 분할 지원과 같은 부가 기능들을 포함하고 있다. 3) MySQL Pro는 InnoDB 지원을 포함하여 MySQL Standard 버전의 상업용 라이선스 버전이다. 4) MySQL Classic은 MySQL Pro와 MySQL Standard 버전과는 달리 InnoDB 스토리지 엔진이 빠진 표준 MySQL 스토리지 엔진만을 포함하고 있다.

2.2. JSP

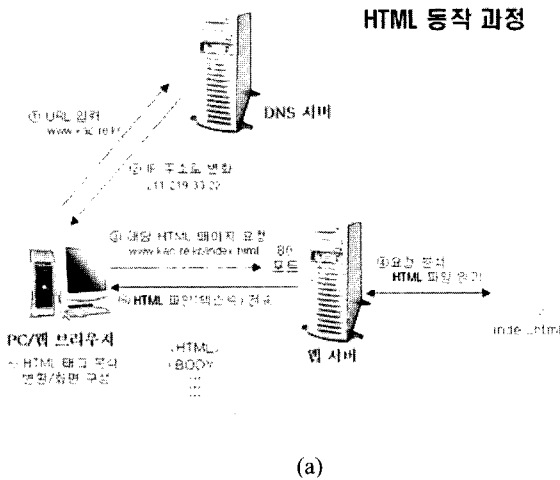
JSP(Java Server Pages)는 자바를 기반으로 한 동적인 웹 사이트 개발 언어로서, ASP(Active Server Pages)와 PHP(Professional Hypertext Preprocessor), CGI(Common Gateway Interface)와 같은 서버측 스크립트 언어이다. 웹 사이트를 개발할 때 많이 사용하는 HTML과 자바스크립트(Javascript)가 클라이언트(사용자)의 웹 브라우저 상에서 실행되는 반면에 JSP는 서버에서 실행되고, 실행된 결과를 클라이언트에 전달해 준다(Field et al. 2002).

JSP는 HTML과 자바스크립트 등의 클라이언트 사이드 스크립트 언어들과 연결된 동적인 웹 페이지를 만들어주는 자바 코드이다. 클라이언트가 웹 서버에 처음으로 JSP 페이지를 요청하면 웹 서버에 있는 JSP 엔진에서 자동으로 서블릿 코드로 변환을 시키고, 그 서블릿 코드를 컴파일 한다. 그리고 컴파일한 클래스를 실행하고, 그 결과를 클라이언트의 웹 브라우저에 보내주게 된다. 이 일련의 모든 과정은 TOMCAT과 같은 JSP 엔진에서 자동으로 처리하며, JSP는 자바 클래스, 서블릿, 애플릿, 웹 서버 등과의 다양한 통신방법을 제공한다(Graham 2002).

2.2.1. JSP 동작과정

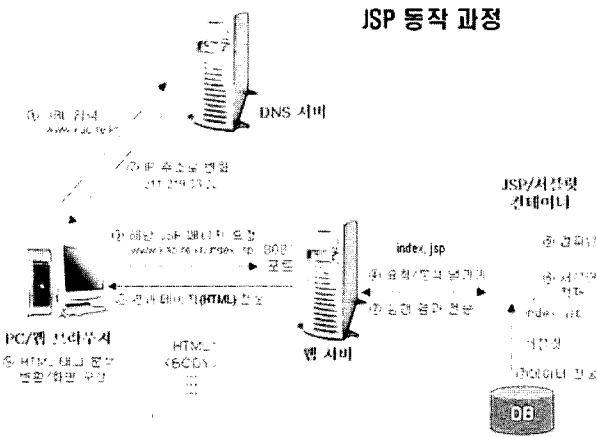
일반적인 HTML 문서로 구성된 웹 페이지의 동작 과정과 JSP로 구성된 웹 페이지의 동작 과정은 그림 2의 (a), (b)와 같은 차이점이 있다.

HTML 동작 과정



(a)

JSP 동작 과정



(b)

그림 2. HTML과 JSP의 동작 과정 비교. (a) HTML로 구성된 웹 페이지의 동작 과정. (b) JSP로 구성된 웹 페이지의 동작 과정.

JSP의 동작 과정은 다음과 같은 세 단계로 요약된다. 첫째, 서버 안에 만들어진 JSP 파일(*.jsp)을 실행한다. 둘째, TOMCAT(JSP 엔진 역할)에 의해서 *.jsp 파일을 서블릿용 java 소스인 *.java 파일로 변경한 다음 컴파일을 통해서 *.class 파일을 생성한다. 셋째, TOMCAT(서블릿 엔진 역할)에 의해서 서블릿용 *.class 파일을 해석하여 처리 결과를 브라우저에 보여준다. TOMCAT은 JSP와 서블릿을 모두 실행 시켜주는 엔진이다.

2.2.2. JSP 특징

JSP는 자바를 바탕으로 한 언어이기 때문에 자바의 모든 특징을 가지고 있다고 볼 수 있다. 특히, 자바의 모든 API(Application Program Interface)를 사용할 수 있는 점과 플랫폼 독립성, 효율성, 메모리 관리 등의 장점이 있

다.

플랫폼 독립성 : JSP와 서블릿도 자바 언어와 마찬가지로 플랫폼에 독립적이다. 이것은 JSP와 서블릿이 웹 서버에 있는 JVM(Java Virtual Machine)에서 컴파일 되고 실행 되기 때문이다. 특히 컴파일 된 서블릿 바이트 코드는 자바를 지원하는 어떤 플랫폼으로 이동해도 실행이 가능하다.

효율성 : CGI와 비교해서 살펴보면 JSP는 클라이언트의 요청을 상당히 효율적으로 처리한다. CGI는 클라이언트의 요청이 있을 때마다 새로운 프로세서를 생성하기 때문에 속도가 느려지고 메모리에 많은 부담을 주는 반면, JSP는 클라이언트의 요청을 받으면 매번 새로운 프로세서를 생성하지 않고 웹 서버에서 실행중인 동일한 프로세서에 의해서 처리되기 때문에 수행속도와 웹 서버의 성능을 크게 향상시킨다. 많은 사용자들이 동시에 접속을 하여도 CGI에 비하여 웹 서버의 기능을 다 할 수 있는 장점을 갖고 있다.

메모리 관리 : 자동으로 메모리를 관리해 주는 것은 자바의 기본적인 특징이다. JVM은, GC(Garbage Collector)가 사용되지 않는 객체들을 사용 가능한 메모리 공간으로 만드는 동작을 자동으로 수행한다.

2.3. MySQL과 JSP의 연동

데이터베이스란 유용한 자료를 모아 놓은 데이터의 집합체를 말한다. JSP는 서버측 언어로, 사용자가 전송한 자료 등을 일정한 저장소에 저장하고 관리하는 프로그램을 작성할 수는 있지만 직접 저장소를 설계하거나 관리할 수는 없다. 따라서 데이터 저장소를 관리하는 MySQL과 같은 DBMS 프로그램을 설치하고, JSP와 연동되도록 환경을 설정하면 데이터베이스에 관련된 프로그램을 개발할 수 있다.

3. 프로그램 구성

데이터베이스 프로그램은 사용자를 위한 부분과 관리자를 위한 부분으로 나누어 제작하였다.

3.1. 사용자 프로그램

데이터베이스의 사용자는 파일을 다운로드 받기까지 네 개의 웹 페이지를 거치게 되며 그 각각에 사용되는 JSP 프로그램 목록과 기능은 표 1과 같다.

3.2. 관리자 프로그램

관리자를 위한 프로그램은 사용자 관리와 데이터 관리, 코드 관리, 관리자 환경 관리 등 크게 네 부분으로 나누

표 1. 사용자 프로그램 목록과 기능

분류	프로그램 이름	프로그램의 기능
검색 부분	index.jsp	검색 메인페이지 프로그램
	db_conn.jsp	DB에 연결하는 프로그램
결과 부분	search.jsp	검색결과를 보여주는 프로그램
	save.jsp	Basket에서 FTP 디렉토리로 저장하는 프로그램
	viewImage.jsp	미리보기 이미지를 보여주는 프로그램
	cartList.jsp	Basket에 담는 프로그램
계정 부분	login.jsp	사용자 로그인 웹 페이지 프로그램
	login_process.jsp	사용자 로그인 프로그램
	logout.jsp	사용자 로그아웃 프로그램
	userEdit.jsp	회원정보 수정 웹 페이지 프로그램
	userUpdate.jsp	회원정보 수정 프로그램
	userInsert.jsp	회원정보 등록 웹 페이지 프로그램
	userReg.jsp	회원정보 등록 프로그램
FTP 부분	deleteFile.jsp	FTP 디렉토리의 파일을 지우는 프로그램
	downloadList.jsp	Basket과 FTP 디렉토리의 목록을 보여주는 프로그램
	selfList.jsp	FTP 디렉토리의 목록을 보여주는 프로그램
	download.jsp	파일을 다운로드하는 프로그램

어지며 그 각각에 사용되는 JSP 프로그램 목록과 기능은 표 2와 같다.

3.3. MySQL 테이블

웹 페이지에서의 구현을 위해 표 1과 표 2에서 언급한 JSP 프로그램을 제작하였고, 이 프로그램에서 필요로 하는 MySQL 테이블의 목록과 내용은 표 3과 같다. 테이블 중 expos_catalogue_soao는 관측된 fits 파일의 정보와 목록을 담고 있으며, user_t와 data_type_t, instrument_t, manager_t, download_t는 각각 사용자 목록, 데이터 형태 목록, 관측기기 목록, 관리자 목록, 다운로드 목록이다.

4. 기능별 프로그래밍

데이터베이스를 구축하기 위한 각종 기능을 갖는 JSP 프로그램 목록을 표 1과 표 2에서 보였다. 사용자를 위한 프로그램은 검색과 결과, 계정, 그리고 FTP 부분으로 크게 나누어진다. 관리자를 위한 프로그램은 사용자관리와 데이터관리, 코드관리, 관리자 환경 부분으로 나누어진다.

표 2. 관리자 프로그램 목록과 기능

분류	프로그램 이름	프로그램의 기능
사용자 관리 부분	userList.jsp	사용자 목록 보여주는 프로그램
	managerYn.jsp	사용자 승인하는 웹 페이지 프로그램
	managerYuUpdate.jsp	사용자 승인 처리 프로그램
	downloadList.jsp	사용자가 다운로드한 목록 보여주는 프로그램
데이터 관리 부분	exposData.jsp	데이터 추가하는 웹 페이지 프로그램
	exposDataInsert.jsp	데이터 추가 프로그램
	exposEdit.jsp	데이터 수정하는 웹 페이지 프로그램
	exposUpdate.jsp	데이터 수정 프로그램
	exposDelete.jsp	데이터 삭제 프로그램
	exposReg.jsp	데이터 등록 프로그램
	exposList.jsp	데이터 목록 보여주는 프로그램
코드 관리 부분	filUpdate.jsp	데이터의 파일명 생성 프로그램
	codeEdit.jsp	코드 수정하는 웹 페이지 프로그램
	codeUpdate.jsp	코드 수정 프로그램
	codeReg.jsp	코드 등록하는 웹 페이지 프로그램
	codeInsert.jsp	코드 추가 프로그램
	codeDelete.jsp	코드 삭제 프로그램
	codeList.jsp	코드 목록 보여주는 프로그램
관리자 환경 부분	index.jsp	관리자 로그인 웹 페이지 프로그램
	login.jsp	관리자 로그인 처리 프로그램
	logout.jsp	관리자 로그아웃 프로그램
	managerEdit.jsp	관리자 수정 웹 페이지 프로그램

사용자와 관리자의 입장에서 웹 페이지 각 부분의 기능에 해당하는 프로그램 제작시 고려한 사항들과 프로그램 소스는 구체적으로 다음과 같다.

4.1. 검색 부분

4.1.1. 필수조건과 추가조건

검색화면은 그림 3과 같고, 필수 검색조건과 추가 검색조건으로 구성되어 있다. 검색대상의 천체명 또는 좌표는 반드시 입력하여야 하는 필수 검색조건이며, 관측날짜, 관측자, 데이터 형태, 관측기기 등은 좀 더 세밀한 검색을 위한 추가 검색조건에 해당한다.

표 3. MySQL 테이블 목록과 내용

테이블 명 : user_t
Primary Key : userid

칼럼 ID	내용	데이터형
userid	사용자 ID	varchar(50)
pswd	암호	varchar(12)
lastname	사용자 이름 (last)	varchar(20)
firstname	사용자 이름 (first)	varchar(20)
middlename	사용자 이름 (middle)	varchar(20)
email	email 주소	varchar(50)
tel	전화번호	varchar(20)
fax	fax번호	varchar(20)
institution	소속기관	varchar(50)
address	주소	varchar(100)
country	국가	varchar(30)

테이블 명 : data_type_t
Primary Key : code

칼럼 ID	내용	데이터형
code	데이터 형태	varchar(20)
cname	데이터 명	varchar(20)

테이블 명 : instrument_t
Primary Key : code

칼럼 ID	내용	데이터형
code	데이터 형태	varchar(20)
cname	데이터 명	varchar(20)

테이블 명 : manager_t
Primary Key : login_id

칼럼 ID	내용	데이터형
login_id	관리자 ID	varchar(20)
pswd	암호	varchar(20)
manager_name	관리자 이름	varchar(20)

테이블 명 : download_t
Primary Key : userid

칼럼 ID	내용	데이터형
userid	사용자 ID	varchar(50)
catalog_code	카타로그 번호	varchar(20)
filename	파일명	varchar(50)
download_list	다운로드 목록	datetime

표 3. 계속

테이블 명 : expos_catalogue_soao
Primary Key : catalog_code

칼럼 ID	내용	데이터형
catalog_code	자료 일련번호	varchar(20)
data_type	자료 종류	varchar(20)
object	관측대상 이름	varchar(20)
ra	RA(관측일)	varchar(20)
dec	DEC(관측일)	varchar(20)
date_obs	관측날짜	varchar(20)
start_ut	관측시간(UT)	varchar(20)
filter	사용필터	varchar(20)
exp_time	노출시간	varchar(20)
observer	관측자 이름	varchar(20)
instrument	관측기기	varchar(20)
filename	파일 이름	varchar(50)
rareult	RA(2000)	decimal(20,17)
decresult	DEC(2000)	decimal(20,17)
rareult1950	RA(1950)	decimal(20,17)
decresult1950	DEC(1950)	decimal(20,17)
ra2000	RA(2000)	varchar(20)
dec2000	DEC(2000)	varchar(20)

4.1.2. 검색대상 선정

필수 검색조건인 검색대상은 천체명 또는 좌표범위 중 한 가지를 선택할 수 있다. 이를 위해 radio button을 사용했고, 천체명과 좌표범위는 직접 입력하도록 하였다. 천체명을 선택할 경우 MySQL 내의 expos_catalogue_soao 테이블의 object 항목에서 조건과 일치하는 자료가 있는지 검색하게 되며, 대소문자와 문자 사이의 여백 유무에 상관없이 검색이 가능하도록 했다. JSP 프로그램 내의 스크립트는 check 함수에서 radiochoose를 수행하여 두 가지 중 하나를 선택할 수 있도록 작성하였다. JSP 프로그램 소스는 아래와 같다.

(radio button을 이용한 검색방법 선택부분)

```
function check()
{
    if(document.form1.radiochoose[0].checked){
        if(document.form1.object.value == "")
            alert("Type in Object name");
    }
}
```


프로그램은 유사한 구조를 갖는다.

(추가조건 중 관측날짜를 결정하는 부분)

```
<TD align right>Observation date <input type=checkbox name=date_obs_utchoose
value="1" <% date_obs_utchoose.equals("1")?checked:"" %>>
</TD>
<TD><input type=text name=date_obs_ut
onkeydown="date_obs_utchoose.checked=true; if(event.keyCode==13){check();}
class="w130" value="<% date_obs_ut%>">
</TD>
```

(추가조건 중 데이터 형태를 결정하는 부분)

```
<TD align right>Data type
<input type=checkbox name=data_typechoose value="1"
</TD>
<select name=data_type onclick="data_typechoose.checked=true" style="width:70px">
<option value="">Any</option>
Connection con= DB_Connection();
PreparedStatement pstmt= null;
ResultSet rset= null;
try{
pstmt= con.prepareStatement("select * from data_type_t order by code");
rset= pstmt.executeQuery();
while(rset.next()) {
String ename= Kr.chk(rset.getString("ename"));
String selected="";
if(ename.equals(data_type)) selected="selected";
out.println("<option value='"+ename+"'" + selected+" ">" + ename +
"</option>");
}
rset.close();
pstmt.close();
</select>
```

4.1.5. 필드갯수 선택

결과화면의 한 페이지에 표시할 목록의 개수를 사용자가 임의로 결정할 수 있도록 하였다.

(필드갯수를 입력하는 부분)

```
if(rows.equals("")){ rows="50"; }
<td align right>Return a maximum of</td>
<td colspan=4><input type=text name=rows class="w65" value="<%=rows%>"
style="text-align:right" onkeydown="if(event.keyCode==13){check();}">rows </td>
```

4.2. 결과 부분

4.2.1. 결과화면 구성

결과화면은 그림 4와 같고, 관측자료의 천체명, 좌표, 관측날짜, 관측시간, 필터, 노출시간, 관측기기 등의 정보를 나타내며 항목별 정렬 기능과 영상 미리보기 기능을 포함하고 있다.

4.2.2. 항목별 정렬 기능

결과화면의 표에서 각각의 항목명을 클릭하면 내림차순과 올림차순을 반복하며 정렬할 수 있도록 제작하였다. 검색

후 많은 목록을 결과로 얻게 되는 경우, 특정한 정보를 찾아내는데 이 정렬 기능은 유용하게 사용된다.

(항목별 정렬 기능 중 파일명과 관측날짜 부분)

```
String sortKind= Kr.chk(request.getParameter("sortKind"));
if(sortKind.equals("")){ sortKind="catalog_code"; }

String sql="select * from expos_catalogue_soa" + whereCond + " order by " +
sortKind + " " + ascdesc;

function sort(sortKind, ascdesc){
document.form1.cpage.value="1";
document.form1.sortKind.value=sortKind;
document.form1.ascdesc.value=ascdesc;
document.form1.submit();
}

<td><A HREF="javascript:sort('catalog_code', '<% sortKind.equals("catalog_code")
&& ascdesc.equals("desc")? "asc":"desc"%>')">File names</A></td>
<td nowrap><A HREF="javascript:sort('date_obs_ut',
'<% sortKind.equals("date_obs_ut") && ascdesc.equals("asc")?
"desc":"asc"%>')">Date Obs</A></td>
<FORM METHOD=POST ACTION="search.jsp?<% request.getQueryString()%>"
name=form1>
<input type=hidden name=sortKind value="<% sortKind%>">
</FORM>
```

4.2.3. 영상 미리보기 기능

영상 미리보기를 위해 관측된 모든 fits 파일에서 jpg 파일을 만들었다. 어두운 천체를 관측한 경우 배경하늘과 천체의 밝기비가 크지 않으므로 선명한 이미지를 얻기가 어렵다. 따라서 한 프레임 속의 모든 픽셀에서 가장 큰 값과 가장 작은 값을 기준으로 밝기분포를 변환시킨 뒤 jpg 파일을 만들었다. 이러한 변환은, 원래의 fits 파일에서 이미지의 절대적인 밝기를 표현하지는 못하지만 이미지의 형태와 fits 파일의 효율성을 판단하기 위한 미리보기의 기능에 적합한 방식이다.

(영상 미리보기 기능 구현 부분)

```
function viewImage(theURL){
self.open("viewImage.jsp?imageName="+
theURL+"&scrollbars=yes,resizable=yes,width=920,height=700");
}
<% if(imgFile.exists() %>
<TD width=60 nowrap><A
HREF="javascript:viewImage('<% imgname%>')">View</A></TD>
<% else %>
<TD width=60 nowrap>N/A</TD>
<% %>
```

4.2.4. 파일선택 기능

검색결과 조건에 맞는 관측자료를 찾아내면 미리보기를 통해 파일의 효율성을 판단하게 되고, 원하는 자료는

Basket에 담게 된다. Basket에 담기 위해 fits 파일 각각을 선택하는 과정이 필요한데, 이 부분도 check box의 형태로 프로그래밍 하였다.

(Basket에 담기 위한 파일 선택 부분)

```
function chkselect()
{
    if(document.form2.filename)
    {
        if(document.form2.filename[0])
        {
            for(i=0; i<document.form2.filename.length; i++)
            {
                document.form2.filename[i].checked = true;
            }
        }
        else if(document.form2.filename)
        {
            document.form2.filename.checked = true;
        }
    }
}

<TD width=40 nowrap><input type=checkbox name=filename
value="<% dcatalog_code%">" ></TD>
<TD width=40 nowrap></TD>
```

4.3. 사용자 계정 부분

데이터베이스에서의 검색은 누구나 할 수 있으나, 검색된 파일을 다운로드 받기 위해서는 사용자 계정을 갖도록 하였다.

4.3.1. 로그인 기능

사용자 계정에 로그인 하는 방법은 웹 상의 일반적인 회원 로그인과 동일하게 만들었다.

(입력된 계정과 암호를 계정목록에서 확인하는 부분)

```
try{
    sql = "select * from user_t where userid = ? and pswd = ?";
    pstmt = con.prepareStatement(sql);
    pstmt.setString(1, id);
    pstmt.setString(2, pswd);
    rset = pstmt.executeQuery();
    if(rset.next())
    {
        loginChk = true;
        File file = new File("pub" + id);
        if(file.isDirectory())
        {
            approvalChk = true;
            session.putValue("USERID", id);
        }
    }
}
rset.close();
pstmt.close();
}
```

4.4. FTP 부분

사용자 계정에 접속하면 자신의 Basket 목록과 FTP 목록의 현황을 함께 비교하면서 볼 수 있게 하였다.

4.4.1. FTP 디렉토리로 올리기 기능

검색결과 화면에서 선택한 파일을 Basket에 임시로 담아 놓은 뒤, 이들 파일 중 다운로드가 필요한 파일을 선별하

여 FTP 디렉토리에 넣을 수 있도록 하였다. 원하는 파일의 선택은 check box로 수행하였다.

(Basket에서 선택된 파일을 FTP로 보내는 부분)

```
function save()
{
    if(document.form2.sumValue.value <= 200){
        alert("Please, note the 200MB limit of the FTP directory.");
        return;
    }
    chk = false;
    if(document.form2.filename)
    {
        if(document.form2.filename[0])
        {
            for(i=0; i<document.form2.filename.length; i++)
            {
                if(document.form2.filename[i].checked = true){
                    chk = true;
                    break;
                }
            }
        }
        else if(document.form2.filename)
        {
            if(document.form2.filename.checked = true){
                chk = true;
            }
        }
    }
    if(chk = false){
        alert("Select save items.");
        return;
    }

    document.form2.action = "save.jsp";
    savewin = self.open("save.jsp", "savewin");
    document.form2.target = "savewin";
    savewin.document.write("<BODY>");
    savewin.document.write("<TABLE width=100% height=100%
style='font-size:12px;' align=center>");
    savewin.document.write("<TR>");
    savewin.document.write("<TD align=center><B><FONT
COLOR='0080FF'>Now Saving</FONT></B><BR><BR>");
    savewin.document.write(" * Wait for a minute.");
    savewin.document.write("</TD>");
    savewin.document.write("</TR>");
    savewin.document.write("</TABLE>");
    document.form2.submit();
    document.form2.target = "_self";
}

<TD colspan=4 align=center>
<A HREF="javascript:save()"></A>
</TD>
```

4.4.2. FTP 디렉토리에서 다운로드 기능

파일을 FTP 디렉토리에 넣으면 이들은 서버 내의 하드 디스크에 담겨지고 웹 또는 FTP로 다운로드할 수 있는 상태가 된다. 사용자 일인당 할당되는 FTP 용량은 총 200MB이며, 사용자가 디렉토리를 관리할 수 있도록 하였다. 즉, 다운로드가 완료된 파일을 지우면 즉시 그 만큼의 공간을 확보하게 된다.

(FTP 디렉토리의 파일을 웹에서 다운로드 하는 부분)


```
String year = "";
String fitsname = "";
if(!ddate_obs_ut.equals("")){
    year = ddate_obs_ut.substring(0,4);
    fitsname = year + "/" + fits + " + dcatalog_code.toLowerCase() + ".fits";
    fitsFile = new File("/kadc/kadc_data/soao/" + fitsname);
}
<TD width=40 nowrap><input type=checkbox name=filename
value="<%=fitsFile.getAbsolutePath().substring(fitsFile.getAbsolutePath().lastIndexOf(
"/")+1)%>" ></TD>
<TD width=60 nowrap><%=Math.round(fileSize/1000000.0*10)/10.0%></TD>
<TD width=60 nowrap><A
href="expos/exposEdit.jsp?catalog_code=<%=dcatalog_code%>"><%=dcatalog_co
de%></a></TD>
<TD width=60 nowrap><A
href="download.jsp?filename=<%=dcatalog_code%>"><%=dcatalog_code%></a>
</TD>
```

4.4.3. 용량표시 기능

FTP 디렉토리 및 Basket 내의 파일 용량을 한 눈에 볼 수 있도록 하였다. 파일 각각의 크기 뿐만 아니라 디렉토리의 사용 중인 용량과 남은 용량을 각각 표시하여 Basket에 있는 파일 중 얼마를 FTP 디렉토리에 담을 수 있는지 쉽게 파악할 수 있게 만들었다.

(Basket과 FTP 디렉토리의 용량을 테이블로 보여주는 부분)

```
function viewSizeOf()
    tot = 0
    if(document.form2.filename){
        if(document.form2.filename[0]){
            for(i = 0; i<document.form2.filename.length; i++){
                if(document.form2.filename[i].checked == true){
                    tot = parseInt(document.form2.filesize[i].value);
                }
            }
        }
        else if(document.form2.filename){
            if(document.form2.filename.checked == true){
                tot = parseInt(document.form2.filesize.value);
            }
        }
    }
    document.form2.selectedValue.value = Math.round(tot/1000000*10)/10.0;
    document.form2.sumValue.value = Math.round((fileTotal-tot)/1000000*10)/10.0;
    if(document.form2.selectedValue.value.indexOf(".") != -1){
        document.form2.selectedValue.value = document.form2.selectedValue.value + ".0";
    }
    if(document.form2.sumValue.value.indexOf(".") != -1){
        document.form2.sumValue.value = document.form2.sumValue.value + ".0";
    }
    document.form2.selectedValue2.value = document.form2.selectedValue.value
    document.form2.sumValue2.value = document.form2.sumValue.value
}
```

4.5. 관리 부분

관리자는 데이터베이스에 관측자료를 추가하고, MySQL 내의 각종 테이블을 업데이트할 뿐만 아니라, 사용자 계정에 대해 사용허가를 부여하고, 다운로드 현황을 파악할 수 있어야 한다. 이러한 기능을 갖는 프로그램을 각각 제

작하였다.

4.5.1. 다운로드 현황 보기

다운로드를 하면 MySQL의 download_t 테이블에 그 기록이 저장되고 이 목록에서 조건에 맞는 내용을 불러와서 화면에 나타내도록 하였다. 다운로드 현황은 사용자별, 월별로 파악할 수 있으며 이를 통해 데이터베이스의 활용도를 객관적 수치로 확인할 수 있다.

(테이블에서 다운로드 현황을 불러오는 부분)

```
String whereCond = "";
if(!userid.equals("")){
    whereCond = " and userid = " + userid + " ";
}
if(!year.equals("")){
    whereCond = " and year(download_date) = " + year + " and
month(download_date) = " + month;
}
if(!whereCond.equals("")){
    whereCond = " where " + whereCond.substring(4);
}
String sql = "select * from download_t";
try{
    pstmt = con.prepareStatement("select count(*) cnt from download_t" +
whereCond, ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE);
    rset = pstmt.executeQuery();
    int tot = 0;
    if(rset.next()){
        tot = rset.getInt("cnt");
    }
    rset.close();
    pstmt.close();
    pstmt = con.prepareStatement(sql + whereCond + " order by userid,
download_date desc", ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE);
    rset = pstmt.executeQuery();
}
```

4.5.2. 자료 추가 기능

새로운 관측자료를 데이터베이스에 추가할 필요가 있을 때 fits 파일과 미리보기용 jpg 파일을 함께 해당 디렉토리에 넣은 뒤 MySQL 테이블을 수정하여야 한다. 기존의 테이블에 새로운 정보를 추가하는 방식으로 진행하며 MySQL에 직접 접속하지 않고 웹 상에서 테이블의 내용을 수정할 수 있도록 프로그래밍 하였다.

(파일로 된 목록을 테이블에 추가 하는 부분)

```
savepath = request.getRealPath("/kadc/expos/data");
String filename = myUpload.getFiles().getFile(0).getFileName();
int filesize = myUpload.getFiles().getFile(0).getSize();
int count = 0;
if(!filename.trim().equals("") && filename != null){
    try {
        if(new File(savepath).isDirectory()){
```

```

    new File(savepath).mkdir();
}
while(new File(savepath+"/"+filename).isFile())
int ext = filename.lastIndexOf(".");
if(ext != -1)
    filename = filename.substring(0, ext) + "_1" + filename.substring(ext);
else
    filename = filename + "_1";
}
myUpload.getFile(0).saveAs(savepath+"/"+filename);
File upl = new File(savepath+"/"+filename);
} catch (Exception e) {
out.println(e.toString());
}
}
}

```

5. DB 구축에 응용

지금까지 언급한 JSP 프로그램들을 모두 제작한 뒤 MySQL을 DBMS로 하여 각 항목에 적합한 데이터형으로 테이블을 만들었다.

5.1. 구축 대상

소백산천문대의 61cm 광학망원경으로 2001년 9월부터 2002년 7월까지 관측한 약 70GB 분량의 관측자료를 대상으로 실제 데이터베이스를 구축하였다.

5.2. 자료의 형태

일반적으로 데이터베이스에서 관측자료를 제공하는 방법에는 (1) 대상천체의 자료와 함께 영점(bias) 영상, 암잡음(dark) 영상, 바닥고르기(flat fielding) 영상 등 전처리에 필요한 자료를 원시자료의 형태로 제공하는 방식과 (2) 영점보정, 암잡음 제거, 바닥고르기 등의 전처리를 수행해서 얻어진 대상천체의 자료만을 제공하는 두 가지 방식이 있다.

원시자료의 형태로 제공하게 될 경우는 데이터베이스에 많은 관측자료를 저장하여야 하며 전송하는 파일의 용량이 커지는 부담이 있다. 한편, 전처리를 수행하는 경우에는 기상조건과 기기상태의 변화 등 관측조건에 따른 특성들을 고려하기가 어렵기 때문에 사용자가 요구하는 정확도를 달성하기 어려운 단점이 있다. 따라서 소백산천문대 관측자료 데이터베이스의 모든 자료는 원시자료의 형태로 보관하고 이를 사용자에게 제공한다.

5.3. 구축 과정

소백산천문대의 관측자료는 모두 2KCCD로 관측되었으며, IRAF를 이용하여 fits 파일의 헤더에서 필요한 정보를 추출하여 목록을 만든 후, MySQL에서 필요로 하는 테이블

의 형태로 전환하였다. MySQL 테이블로 전환하기 이전에 관측 로그북의 내용과 생성된 목록의 내용을 비교하여 목록의 정확도를 검증하였다.

본 연구에서 제작된 데이터베이스 구축 시스템에 맞게 목록을 배열하고, 이 시스템에서 규격화된 MySQL 데이터베이스 테이블의 항목과 포맷에 따르는 형태로 각각의 내용을 제작하였다. 데이터베이스의 디렉토리 구조도 규격화된 시스템의 요구에 맞게 연도별로 배치하였다. 또한 중복된 파일명을 고유한 이름으로 변환하여 정리하였다.

미리보기 파일은 IDL(Interactive Data Language)을 이용하여 jpg 형태로 제작한 뒤 데이터베이스화함으로써, 검색 요구시 해당 fits 파일을 개별적으로 연산 처리하여 사용자의 웹 브라우저로 전송하는 방식에 비해 처리속도가 빠르다.

소백산천문대 관측자료 데이터베이스의 웹 검색화면과 결과화면은 그림 3과 그림 4에 나타낸 것과 같고, 각 부분의 기능은 3장과 4장에서 자세히 언급한바와 같다.

5.4. DB 테스트

데이터베이스의 정상작동 여부는 사용자 측면과 관리자 측면으로 나누어 각 항목별로 실시하였다.

데이터베이스는 대용량의 자료를 다양한 검색조건을 이용하여 검색하고, 그 결과를 다운로드 받을 수 있도록 제작되었다. 따라서 사용자 측면에서는 검색엔진의 정상 작동, FTP 서버 내에서의 파일 복사, 자료의 다운로드 등을 점검하였다. 검색엔진 점검에서는 입력된 각종 검색어에 대하여 올바른 결과가 나오는지 확인하기 위해 데이터베이스에 입력된 관측대상 이름과 좌표를 그룹별로 세분화하여 각각을 입력하는 방법으로 테스트하였다.

관측대상의 좌표로 검색하는 경우, 좌표의 범위를 달리 할 때 정확한 결과를 나타내는지 확인하였다. DB 내에서 검색을 한 뒤 그 결과를 웹으로 표시하는데 걸리는 시간은 일반적으로 1-2초 정도이고, Basket에 담긴 200MB의 자료를 한꺼번에 FTP 디렉토리에 담는 경우에 걸리는 시간은 10초 정도인 것으로 나타났다.

관리자 측면에서는 DB의 관리와 사용자 관리로 나누어 각 항목별로 점검을 하였다. 먼저 DB의 관측자료 목록에서 각 항목들이 정확하게 입력되었는지 여부와 새로운 관측자료 목록을 추가하는 기능이 정상적으로 작동하는지를 확인하였다. 새로운 관측자료 목록은 fits 파일의 헤더에서 읽어낸 정보를 텍스트 형태로 입력할 수 있도록 제작되었으며, DB 내에서의 파일명 생성과 RA, DEC 변환 등을 거쳐 각 항목별로 DB의 테이블에 정확히 입력됨을 확인하였다. 사용자 리스트와 각 사용자의 정보 관리, 사용자별 사용내역 등과 월별 데이터베이스 사용내역 등 각종 통계자료가 정확히 산출되고 있음도 확인하였다.

6. 결론

그동안 운영되어 온 국내의 천문대뿐만 아니라 국외에 건설되고 있는 천문대에서도 방대한 양의 관측자료가 축적되어질 예정이다. 이들 관측자료의 활용도를 높이기 위해 관측자료의 데이터베이스 구축에 대한 요구가 제기되어 왔고, 한국천문연구원 천문우주정보센터에서는 MySQL과 JSP를 이용한 데이터베이스 구축 연구를 수행하였다.

MySQL은 공개 프로그램일 뿐만 아니라 빠른 속도로 많은 자료를 처리할 수 있는 장점을 갖고 있어 DBMS 중 가장 많이 사용되고 있다. HTML의 기능으로는 데이터베이스 내의 자료를 검색할 수 없으므로, 서버에서 실행되고 실행된 결과를 클라이언트에 전달해주는 동적 스크립트 언어인 JSP를 함께 활용하는 것이 효율적이다.

본 연구에서는 관측자료의 데이터베이스화에 필요한 MySQL 테이블들을 제작하고, 이들 자료를 기반으로 웹에서 검색이 가능한 JSP 프로그램을 기능별로 각각 작성하였다. 테이블과 프로그램은 사용자 부분과 관리자 부분으로 나누어지며, user interface(Stucky 2001)와 데이터베이스의 확장성을 높일 수 있도록 제작하였다.

앞으로, 관측자료의 검색부분을 더욱 강화할 계획이다. 천체명으로 검색하는 경우, 동일한 천체에 대해 다양한 이름으로 검색이 가능하도록 CDS(Centre de Données astronomiques de Strasbourg)의 천체 카타로그와 같은 자료를 이용한 데이터베이스를 구축할 예정이다. 또한 국외에 설치한 천문대에서 얻어지는 관측자료의 경우 전처리를 통해 일차 가공된 결과를 데이터베이스화할 수 있도록 할 계획이다.

본 연구를 통해 확립한, MySQL과 JSP를 이용한 데이터베이스 구축 방안은 다양한 관측자료에 적용할 수 있으며, 한국가상천문대의 초기시스템으로 채택이 가능하며, 국제가상천문대연맹의 협력사업에 참여하는 한국의 대표적인 데이터베이스 모델이 될 것이다.

참고문헌

경재만, 김호일, 김승리, 권순길, 민병희 2002, 천문우주 DB 구축 및 자료처리, 한국천문연구원, 166
 김상철, 성현일, 김영수, 임인성, 김종수, 남현웅, 김봉규, 안영숙 2003, 가상천문대의 기본 구조와 세계적 흐름, 한국천문연구원 기술보고서 No. 03-005-055
 김호 2003, Kimho의 MySQL로 배우는 데이터베이스 프로그래밍, 영진.COM
 박병곤 & 박윤호 2002, 천문우주 DB 구축 및 자료처리, 한국천문연구원, 211

성현일, 김상철, 남현웅, 김봉규, 임인성, 윤요나 2003, 천문학논총, 18, 43
 성현일, 임인성, 김봉규, 김상철, 김종수 2002, 천문우주 DB 구축 및 자료처리, 한국천문연구원, 1
 이춘식 2002, 업무 영역별 데이터베이스 설계와 구축, 한빛미디어
 조준익 2000, UNIX 서버와 DB구축, 미래컴
 허정수 2000, 아주 특별한 웹데이터베이스 MySQL & WEB DB 연동, 베스트북
 Field, D. K., Kolb, M. A, & Bayern, S. 2002, Web Development with Java Sever Pages, 인포북 (곽용재 옮김)
 Graham, S. 2002, 자바를 이용한 웹 서비스 구축, 인포북 (석광진 옮김)
 Kimula, H. & Takahashi, M. 2002, 입문 SQL, 영진.COM (오준혁 옮김)
 Stucky, M. 2001, MySQL : Building User Interfaces, New Riders
 Sung, H.-I., Kim, S. C., Yim, I. S., Nam, H.-W., Kim, B. G., Kim, J., & Lee D.-J. 2004, in ASP Conf. Ser. Vol. 314, Astronomical Data Analysis Software and Systems XIII, ed. F. Ochsenbein, M. G. Allen, and D. Egret (San Francisco: ASP), 165
 Thomson, L. & Welling, L. 2001, PHP와 MySQL을 이용한 웹사이트 개발, 인포북 (황영준 옮김)
 Welling, L. & Thomson, L. 2003, 성공적인 웹 프로그래밍 - PHP와 MySQL, 인포북 (류정욱 옮김)