

## 10. 게시판 구현 – MySQL 설치, Bean 설정

### 1) 요구분석 (게시판 구현)

#### (1)게시판의 종류

- ① 공지사항 : 관리자만 글쓰기, 편집, 삭제가 가능
- ② 자유게시판 : 1) 회원에 가입하지 않고도 글쓰기, 수정, 삭제, 조회 가 가능함  
2) 댓글 달 수 없음
- ③ **Q&A 게시판** : 답전형 게시판 : 변형이 되는 게시판 (댓글이 내부적으로 감춰져있음)  
: 테이블을 2 개를 설계해야함  
: 글쓰기, 글수정, 글삭제, 글조회 + 댓글기능  
: 회원제 운영 (실명제)  
: 글삭제 (관리자가)
- ④ 자료실 겸용 게시판 : Q&A 게시판 + 파일 업&다운로드 : 스프링으로 구현

#### (2) JspBoard 답변형 게시판 구현 : 필드

- ① 눈에 보이는 필드 > 번호, 분류(공지,Q&A,,,,),제목,이름,작성일짜,조회수
- ② 눈에 **보이지 않는** 필드 > 댓글(ref,step,level)
- ③ 게시판의 번호 : 가장 처음(위)에 보이는 글의 번호가 가장 큼.
- ④ 한페이지당 기본 레코드 개수 : 10 개 , (페이지가 15 개 → 총 레코드는 150 개)

JspBoard==>1.페이징 처리기법 2.댓글다는 로직->회원관리와 동일

회원리스트(메서드)

#### ① DB에 연결

-src-> **DBConnectionMgr(DB)->mysql**로 작성

**private String \_driver = "org.gjt.mm.mysql.Driver",->mysql 드라이버**

**\_url = "jdbc:mysql://127.0.0.1:3306/mydb?useUnicode=true&characterEncoding=EUC\_KR",**  
**===(포트번호 3306,3307)**

**/mydb(접속할 DB명)?useUnicode=true&characterEncoding=UTF-8",**

**\_user = "root", (접속계정명)**

**\_password = "1234";(암호)**

#### | ② mysql 드라이버 라이브러리에 추가

-WebContent

-WEB-INF

**-lib->접속할 드라이브에 대한 라이브러리를 추가로 설정**

**mysql-connector-java-5.1.12-bin.jar**

**-list.jsp(글목록보기)->memberlist.jsp(회원리스트) ← 이게 게시판**

## 2) MySQL 설치 및 사용

### (1) 설치 : <http://www.mysql.com> , sql 다운로드

\*\*환경설정을 하면서 세밀하게 설치-> **Custom**

\*\*설치경로-> C:\MySQL 5.1\

Register the MySQL Server now->체크해제

계정생성 X (등록 X)

Configure the Mysql Server now ->체크유지

=>계속 환경설정을 이어서 할 예정

=>언급하지 않은 부분은 그냥 next

Multifunctional Database(다기능 목적)로 사용

Decision Support=>통계목적 사용(X)

Manual Setting =>동시접속수 지정(15) 체크

포트번호 3306 or 3307(다른 mysql 이 설치된경우) <->오라클 1521

언어선택->euckr=>utf8(대세)

어디든 입사하면  
자기 컴퓨터 설정 정도는  
할 줄알아야함

\*\* 서비스 → MySQL

\*\* PATH 에 bin 폴더 등록->경로에 상관없이 mysql 명령어 사용하게 하기위해서

암호 1234 =>관리자 -> root

\*\* 원격접속 체크->Enable root access from remote machine

가능하게(교육 때문에)->보안때문에 제한을 줄 수 있다.

### (2) 사용법 : 간단한 콘솔 작업

1. **show databases;** → 현재 데이터베이스 목록확인 전체 내용을 확인 (오라클의 select \* from tab)

2. use 전환할 DB 명; mysql> **use mysql;** → 특정 DB 에 접속

> MySQL 계정 로그인(root)/1234 → 특정 DB 에 접속 → 테이블 생성

(오라클 : 사용자 관리->계정을 만들어서 테이블 작성=>테이블스페이스 저장)

3. **show tables;** DB 에 들어가 있는 테이블을 보여주세요(목록)

4 .desc user; //구조확인 → 오라클과 동일 → desc 테이블명

5. select \* from user;

-----

자기자신의 root 계정

(암호화-1234)

| localhost | root | \*A4B6157319038724E3560894F7F932C8886EBFCF | Y

| 0 | 0 | 0 | 0 |

| % | root | \*A4B6157319038724E3560894F7F932C8886EBFCF | Y

=====>원격으로 접속할때 사용하는 관리자 계정명

6. mysql>exit; → 종료

### (3) SQLGate for Mysql : MySQL 을 시각적으로 조작하기 좋게 만든 툴

유저명:root

패스워드:1234

서버: localhost // 상대방의 ip 주소를 입력->원격가능

포트: 3306 or 3307

데이터베이스: mysql ->처음에는 존재하는 DB 에 접속

**\*\*캐릭터셋:utf8**

**\*\*유니코드 사용->반드시 체크할것->한글사용 가능**

연결테스트->접속성공

오라클과 mysql의 자료형 비교

	오라클	MySQL
문자	varchar2	varchar
숫자	number	int,smallint
긴 문자	varchhar2	text

### (4)SQLGate for Mysql : MySQL 사용법

-- 한줄 주석 | /\* 여러줄 주석 \*/

--1.DB 생성 **create database** 생성할 DB 명

> **create database imsi;**

--2.DB 삭제 **drop database** 삭제시킬 DB 명;

> drop database imsi;

> create database mydb;

> **use mydb;**

--3.테이블 생성 ->오라클과 형식이 거의 같다.(varchar,int,text)

```
CREATE TABLE sawon (  
  id int(5) NOT NULL,  
  name varchar(13) NOT NULL,  
  dept varchar(14) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

--4.테이블의 구조 (desc 테이블명)

desc sawon

--5.필드를 추가

--alter table 수정 t add column 필드명 자료형(=데이터형) 제약조건  
alter table sawon add column addr varchar(40);

--6.필드를 삭제

--alter table 수정 t drop column 삭제할 필드명  
alter table sawon drop column addr;

--7.필드를 변경

--alter table 수정 t modify column 변경할 필드명 자료형 제약조건  
alter table sawon modify column dept varchar(30);

--8.insert

insert into sawon values(1,'홍길동','영업부');  
insert into sawon values(2,'테스트','개발부');  
insert into sawon values(3,'임시','관리부');  
insert into sawon values(4,'임시 2','총무부');  
insert into sawon values(5,'테스트 2','생산 2 부');

-- 9.select

select \* from sawon where id=2 or dept='영업부';

--10.테이블의 이름을 변경=>rename table 원본 t 명 to 변경 t 명

**rename table sawon to test;**

rename table test to sawon;

--11.테이블의 백업

--create table 백업 t 명 as select \* from 원본 t 명  
create table b\_sawon as select \* from sawon;

--12.필드정렬-> **order by** 정렬할 필드명 정렬기준(asc or desc)

select \* from sawon order by id desc;

### 3) 테이블 설계

#### (1) mydb : 게시판 테이블 작성

```
CREATE TABLE board (
  num int(11) NOT NULL AUTO_INCREMENT,
  writer varchar(10) NOT NULL,
  email varchar(30) DEFAULT NULL,
  subject varchar(50) NOT NULL,
  passwd varchar(12) NOT NULL,
  reg_date datetime NOT NULL,
  readcount int(11) DEFAULT '0',
  ref int(11) NOT NULL,
  re_step smallint(6) NOT NULL,
  re_level smallint(6) NOT NULL,
  content text NOT NULL,
  ip varchar(20) NOT NULL,
  PRIMARY KEY (num)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

##### **auto\_increment**

- **자동**으로 게시물번호가 증가하라.
- 수동으로 입력하는 경우 적용X –
  - > 직접 게시물번호가 중복이 안되게 넣어줘야
- 오라클 > sequence와 기능이 동일  
pk에 부여 :  
중복되면 안되는 필드에 부여한다.

오라클에서 :

=>create sequence 시퀀스명board\_seq; (1,1)  
=>insert into board values (board\_seq.nextval,~)

로그인 구현시 사용했던 DBConnectionMgr.java를 그대로 사용한다

/\* 원격의 MySQL에 접속하는 경우의 환경설정부분 \*/

```
private String _driver = "org.gjt.mm.mysql.Driver",
  _url = "jdbc:mysql://192.168.0.101:3306/mydb2?useUnicode=true&characterEncoding=UTF-8",
  _user = "root",
  _password = "1234";
```

### 4) 게시판 작동 과정

list.jsp 접속 → 리스트 + 페이지 → 글쓰기(클릭) → writeForm.jsp(이동) → 내용입력 후 글쓰기 → writeProc.jsp  
→ list.jsp 로 되돌아옴 (새 글이 등록)

## BoardDTO.jsp (: 게시판의 필드를 정의)

```
package devtaco.board;

import java.sql.Timestamp;
```

mydb 테이블의 필드들을 코딩해준다.

```
public class BoardDTO {
    // 눈에 보이는 필드
    private int num;
    private String writer; // 작성자
    private String subject; // 제목
    private String email; // 이메일
    private String content; // 글내용
    private String passwd; // 글 비밀번호
    // 눈에 보이지 않는 필드 --> 직접 입력할 수 없음 (시스템에서 읽어들이어서 입력)
    private Timestamp reg_date; // sql 내장 객체로, 날짜데이터를 가져온다
    private int readcount; // 조회수
    private String ip; // 작성자의 ip 주소

    // 공지사항, 자유게시판 + 3개의 필드 추가( 답변형 )
    private int ref; // 글 그룹번호 ( 분류별 그룹, 단독성격을 띄게되면 게시물번호와 역할이 같음)
    private int re_step; // 답변글의 '순서' 지정 ( 같은 그룹일때 답변글 순서)
    private int re_level; // 답변글의 답변 -> 글의 깊이(depth)

    ""
    각 필드의 setter, getter 생성
}
```

### 1) ref (reply family)

: \*\*\* 답변글로 사용하는 경우 → **윗글(일반글)**과 **답변글을 묶어준다.**  
(그룹번호 역할을 해준다.)

**\*\* 댓글의 위치를 알려준다.\*\***

: 단독으로 사용하는 경우

**\*\*각각의 게시물을 구분하는 인자로** 사용(=num와 같다)=>pk(중복X)

num	writer	email	ref
1	aaa	a@daum.net	0 or 1부터 시작가능하다.
2	bbb	b@daum.net	1 or 2

### 2) re\_step(reply step) : 몇번째 단계의 답변인지를 나타내는컬럼

: 같은 level 내에서도 답변글을구분하는 인자 (asc)으로 순서 구분  
오름차순으로 구분->0 ,1 2 ,3,4,5

### 3) re\_level(reply level)=> 글의 정렬순서를 지정해주는 컬럼 ==>x축으로 구분(들여쓰기)

: 답변순서를 정해줄때 사용하는 필드

답변이 많아질 수 록 숫자가 증가된다. **depth(깊이)**