Image Upload

● 내(웹 서버)겐 너무 큰 당신(이미지)!! 어쩌면 좋을까~
- 웹 환경에서의 이미지 출력은 너무 힘든 작업이다. 왜? 크기 때문에.
- 따라서, 이미지의 크기를 작게 줄여서 처리 한다. 작은 이미지를 썸네일(Thumbnail:엄지손톱)이미지라고 한다.

● 이미지 파일 읽기
- 이미지 파일을 읽는 작업은 ImageIO 클래스가 제공하는 read() 메서드를 사용한다.
- 읽을 수 있는 이미지 파일의 종류는 PNG, JPEG, GIF 등의 세 가지이다.
- c:\image\product.gif 파일을 읽는 예

```
File file = new File("c:\\image\\product.gif");
BufferedImage image = ImageIO.read(file);
                        혹은
FileInputStream input = new FileInputStream("c:\\image\\product.gif");
BufferedImage image = ImageIO.read(input);
        URL을 지원하므로 다른 웹 서버로부터 이미지를 읽을 수도 있다.
URL readURL = new URL("http://어디어디닷컴/main/product.gif");
BufferedImage urlImage = ImageIO.read(readURL);
```

● 작은 이미지 만들기, 즉 썸네일 이미지 만들기
- 썸네일 이미지를 만드는 과정은 다음과 같다.

① ImageIO.read() 메서드를 사용해서 이미지를 BufferedImage에 저장한다.(원본)
② 작은 크기를 갖는 새로운 BufferedImage 객체를 생성한다.(대상)
③ 대상 BufferedImage 객체의 createGraphics() 메서드를 호출해서 대상 BufferedImage 객체에 그림을 그릴 수 있는 객체(Graphics2D)를 얻어온다.
④ 위의 ③에서 얻은 Graphics2D객체를 사용해서 원본 BufferedImage를 대상 BufferedImage에 그린다. 이 때, 작은 크기로 그린다.

- 위의 과정을 프로그램으로 하면 다음과 같다.

```
BufferedImage srcImage = Image().read(imageFile);//과정 1
BufferedImage destImage = new BufferedImage(w, h, BufferedImage.TYPE_3BYTE_BGR);//과정 2
Graphics2D g2 = destImage.createGraphics();//과정 3
g2.drawImage(srcImage, 0, 0, width, height, null);//과정 4.
                        //srcImage를 새로운 width와 height의 크기로 destImage로 그린다.
```

- 변경된 이미지를 파일로 저장한다.

```
File file = new File("new.jpg");
ImageIO.write(destImage, "jpg", file);
```

답변글 게시판

● 글을 읽고 그 글에 대한 답글을 단다.

- 답글 작업은 그리 간단한 문제가 아니다. 즉, 무슨 글에 대한 답변글인지를 파악해야 되며, 또한, 답변글에 대해서 또다른 답변글이 달릴 수 있기 때문이다.
- 답변글 작업시에 중요한 사항은 다음과 같다.

① 어느글에 대한 답변글인지를 저장해야 한다.

② 답변글에 대한 또다른 답변글이 달릴 수 있다.

③ 답변글들 사이의 순서를 유지해야 한다.

- 위의 사항들을 위해 다음의 정보를 유지한다.

· 일련번호 : 글들의 고유번호로 새로운 글이 추가될 때마다 자동으로 1씩 증가한다.

· 그룹번호 : 관련된 글들이 공유하는 번호. 그룹번호가 동일하면 모두 관련된 글들임을 의미한다.

· 부모번호 : 어떤 글에 답변글이 달릴 경우, 그 글의 일련번호가 부모번호가 된다.

· 출력순서 : 같은 그룹 내에서의 출력 순서를 의미한다.

- 다음은 게시판에 올린 두 개의 글에 대한 정보이다.

입력순서	일련번호	그룹번호	부모번호	출력순서
1	1	1	0	0
2	2	2	0	0

1. 가나다라
2. 어쩌구저쩌구

- 1번 글에 답글을 단 후의 정보이다.

입력순서	일련번호	그룹번호	부모번호	출력순서
1	1	1	0	0
3	3	1	1	1
2	2	2	0	0

1. 가나다라
   [RE] 가나다라
2. 어쩌구저쩌구

- 다시 1번 글에 또 다른 답글을 단 후의 정보이다.

입력순서	일련번호	그룹번호	부모번호	출력순서
1	1	1	0	0
3	3	1	1	1
4	4	1	1	2
2	2	2	0	0

1. 가나다라
   [RE] 가나다라
   [RE] 가나다라
2. 어쩌구저쩌구

- 이번에는 3번 답글에 답글을 단 후의 정보이다.

입력순서	일련번호	그룹번호	부모번호	출력순서
1	1	1	0	0
3	3	1	1	1
5	5	1	3	2
4	4	1	1	3
2	2	2	0	0

1. 가나다라
   [RE] 가나다라
   [RE] 가나다라
   [RE] 가나다라
2. 어쩌구저쩌구

● 게시판의 설계

- 다음의 규칙을 갖는 게시판을 설계한다.

① 1단계의 글은 반드시 이미지를 업로드 해야 한다.

② 답변글에는 이미지를 업로드 하지 않아도 된다.

③ 글 목록에는 업로드한 이미지의 썸네일이미지를 보여준다.

④ 글을 등록할 때는 수정과 삭제에 사용되는 암호를 입력한다.

● 데이터 베이스 테이블 설계

- 필요한 테이블은 다음과 같다.

① 게시판의 글에 대한 정보를 갖는 테이블 - WRITING_INFO

② 게시판의 내용을 저장하는 테이블 - WRITING_CONTENT

③ 게시판의 글 번호를 저장하는 테이블 - ID_SEQUENCE

- 게시판의 내용을 별도의 테이블에 저장하는 이유는 검색 속도를 향상시키기 위해서이다. 일반적으로 게시판의 글 목록에서는 글의 내용은 보여주지 않는다. 따라서, 검색할 때에는 필요한 정보만을 검색하도록 한다. 긴 내용을 포함하는 테이블은 검색 속도에서 불리하다.

- WRITING_INFO 테이블의 구조

열 이름	데이터 형태	설 명	비 고
WRITING_ID	NUMBER(5)	일련 번호	PRIMARY KEY
GROUP_ID	NUMBER(5)	그룹 번호	NOT NULL
ORDER_NO	NUMBER(5)	목록에 출력되는 순서	NOT NULL
LEVEL_NO	NUMBER(5)	단계 번호	NOT NULL
PARENT_ID	NUMBER(5)	부모글의 일련 번호	NOT NULL
REGISTER_DATE	VARCHAR2(25)	등록일	NOT NULL
WRITER_NAME	VARCHAR2(20)	작성자 이름	NOT NULL
EMAIL	VARCHAR2(80)	작성자 이메일	NOT NULL
IMAGE_NAME	VARCHAR2(40)	이미지 파일의 이름	NOT NULL
PASSWORD	VARCHAR2(20)	암호	
TITLE	VARCHAR2(100)	글의 제목	NOT NULL

- WRITING_CONTENT 테이블의 구조

열 이름	데이터 형태	설 명	비 고
WRITING_ID	NUMBER	일련 번호	PRIMARY KEY
CONTENT	CLOB	글의 내용 (CLOB는 최대 4GB의 문자 데이터를 저장할 수 있다)	NOT NULL

- ID_SEQUENCE 테이블의 구조

열 이름	데이터 형태	설 명	비 고
TABLE_NAME	VARCHAR2(50)	테이블의 이름	PRIMARY KEY
LAST_ID	NUMBER(5)	마지막 글의 일련 번호	NOT NULL

- 게시판에 글을 쓸때는 ID_SEQUENCE 테이블로부터 마지막에 저장한 글 번호를 읽어와서 1을 더한 값을 새로운 글의 번호로 사용한다. LAST_ID에도 1 증가된 값을 저장한다.

- ID_SEQUENCE 테이블에 TABLE_NAME에는 테이블 이름이 저장되는데, 테이블 마다 별도의 글 번호를 관리하기 위해서이다.