페이징 처리 관련 DAO의 보조 클래스 작성 - PagingControl

저장된 글의 총 개수에 따라서 몇 개의 페이지로 나눌 것인지도 결정해야 하고 특정 페이지 선택 시 각 페이지에 알맞은 글을 DB 서버에 요청해서 가져와야 하기 때문이다.

PagingControl이라는 클래스를 생성하여 페이징 처리에 필요한 기능들을 구현해서 적용한다. 반드시 그래야 하는 것은 아니지만 PagingControl 클래스를 EmpDAO의 내부 클래스로 정의한다. 우선 PagingControl 클래스에 정의하는 멤버 변수들이다.

public int lineCnt = 0; // 한 화면에 구성하는 글 개수

public int pageCnt = 0; // 한 화면의 페이지 개수

public int postCnt = 0; // 요구 사항을 만족하는 글의 전체 개수

public int pgNum = 0; // 현재 페이지 넘버

● 글의 개수에 따른 페이지 개수 계산: int getPageCount( )

**((요구사항을 만족하는 글의 전체 개수-1)/한 화면을 구성하는 게시판 글 개수)+1**

나눗셈 연산 시 몫만 구하고 나머지 값은 버려지므로 + 1을 하고 나눗셈 연산 시 나머지 값 없이 딱 떨어지는 경우를 대비하여 요구사항을 만족하는 글의 전체 개수에서 1을 차감하여 처리한다.

● 각 화면별 시작 페이지 번호: int getPageStart( )

**((현재 페이지넘버-1)/ 한 화면의 페이지 개수)\*한 화면의 페이지 개수+1**

● 각 화면별 종료 페이지 번호: int getPageEnd( )

**각 화면 별 시작 페이지 번호 + 한 화면의 페이지 개수 - 1 과 전체 페이지 개수 중에서 작은 값**

● 이전 페이지들을 출력하는 left 이미지 출력 여부 결정: boolean isPreData( )

화면별 시작 페이지 번호가 1이 아니면 출력

● 이전 페이지들을 출력하는 right 이미지 출력 여부 결정: boolean isNextData( )

각 화면별 종료 페이지 번호가 전체 페이지 개수보다 작으면 출력

● 현재 페이지 넘버에 따라 DB 서버에서 가져올 글의 시작 행 넘버와 종료 행 넘버

- 시작 행 넘버: **종료 행 넘버 – 한 화면을 구성하는 게시판 글 개수 + 1**

int getWritingStart( )

- 종료 행 넘버: **현재 페이지 넘버 \* 한 화면을 구성하는 게시판 글 개수**

int getWritingEnd

[ 오라클의 rownum 가상컬럼 사용 ]

오라클에서 지원하는 가상컬럼(Pseudo Column)으로 쿼리의 결과에 가상으로 행 번호를 1부터 부여한 순번.

1. 오라클에서만 사용가능

2. 주로 페이징처리, TOP-N QUERY에 사용

[ rownum 사용 예 ]

select \* from visitor;

select rownum, name, writedate from visitor;

select rownum, name, writedate from visitor where to\_char(writedate, 'yyyymm') = '202001';

select rownum, name, writedate from visitor order by name;

select rownum, name, writedate from visitor where to\_char(writedate, 'yyyymm') = '202001' and rownum >= 1 and rownum <= 5;

select rownum, name, writedate from visitor where to\_char(writedate, 'yyyymm') = '202001' and rownum >= 6 and rownum <= 10;

select oldrownum, name, writedate from (select rownum oldrownum, name, writedate from visitor where to\_char(writedate, 'yyyymm') = '202001') where oldrownum >= 1 and oldrownum <= 5;

select oldrownum, name, writedate from (select rownum oldrownum, name, writedate from visitor where to\_char(writedate, 'yyyymm') = '202001') where oldrownum >= 6 and oldrownum <= 10;

[ rownum 컬럼에 값이 부여되는 시점 ]

FROM

WHERE <- ROWNUM이 처리되는 시점

GROUP BY

HAVING

SELECT

ORDER BY

[ 월급이 적은 순에서 6번째 ~ 10번째 행을 추출 : 우리가 SQL 공부할 때 사용한 테이블과 좀 다름 ]

select eno, ename, job, hyear, sal, rnum

from ((select eno, ename, job, hyear, sal, rownum rnum

from (select eno, ename, job, to\_char(hdate, 'yyyy') hyear, sal

from emp order by sal)))

where rnum between 6 and 10

[ 페이징에서의 활용 SQL ]

select eno, ename, job, hyear, sal, rnum

from (select eno, ename, job, to\_char(hdate, 'yyyy') hyear, sal, rownum rnum from emp)

where rnum between 1 and 10;

select eno, ename, job, hyear, sal, rnum

from (select eno, ename, job, to\_char(hdate, 'yyyy') hyear, sal, rownum rnum from emp)

where rnum between 11 and 20;