



[한국ICT인재개발원] 스프링 프레임워크

6. OracleSQL의 페이징 처리

前) 광고데이터 분석 1년

前) IT강의 경력 2년 6개월

前) 머신러닝을 활용한 데이터 분석 프로젝트반 운영 1년

前) 리그오브 레전드 데이터 분석 등...

現) 국비반 강의 진행중

90972	[일반] 화가 많으면 [4]	toheaven	21/03/19	2049	2
90971	[일반] 남의 밥그릇을 깨기 전에 필요한 고민의 크기 [29]	눈팅만일년	21/03/19	5703	87
90970	[일반] [완전스포] 스나이더컷 2017 버전과 차이점에 중점을 둔 정리 [61]	나주꿀	21/03/18	4472	8
90969	[일반] [슬램덩크] 강백호의 점프슛 이야기 [32]	라울리스타	21/03/18	3142	37
90968	[일반] 그 때 너를 붙잡았더라면... [4]	조공플레이	21/03/18	1620	3

[목록](#) [이전](#) [다음](#) [글쓰기](#)

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [NEXT 10](#)

페이징 처리는 게시판의 모든 글을 보여주지 않고 n개씩 끊어서 보여주는 처리를 의미합니다.

Oracle sql에서는 rownum이라는 개념을 이용해 처리합니다.

```
insert into board_tbl (bno, title, content, writer)
(select BOARD_NUM.nextval, title, content, writer from board_tbl);
```

80개 행 이 (가) 삽입되었습니다.

160개 행 이 (가) 삽입되었습니다.

먼저 위 구문을 이용해 게시물의 글을 기하급수적으로 늘려보겠습니다.

위 구문을 실행하면 서브쿼리 안에서 조회된 모든 **row**를 이용해

Insert구문을 **row**갯수만큼 실행해 결과적으로 데이터가 두 배씩 늘어납니다.

```
select  
/*+ 힌트가 들어갈 자리 */  
* from board_tbl;
```

이제 이 구문들을 조회할 때 **order by bno desc** 대신 인덱스를 이용해 더 빠르게 처리하도록 만들어보겠습니다.

우선 **hint**라는 구문을 쿼리문에 삽입해서 처리합니다.

```
select  
/*+ 힌트가 들어갈 자리 */  
* from board_tbl;
```

힌트는 어떤 요소를 이용해 정렬시킬지를 적는것으로

위에 보시다시피 /*+ 가 여는부분이고 */가 닫는부분인데

주석이나 마찬가지로 잘못된 힌트를 삽입해도 구문에러는 나지 않습니다.

```
select
/*+ INDEX_DESC(board_tbl pk_board) */
* from board_tbl;
```

우리는 예전에 **pk_board**라는 이름으로 **bno**에 **primary key**를 부여했습니다.

따라서 **board_tbl**테이블의 **pk_board**를 기준으로 인덱싱을 하라는 힌트를 넘겨보겠습니다.

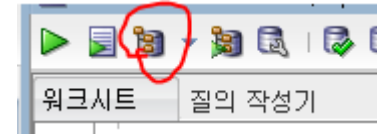
```
select  
/*+ INDEX_DESC(board_tbl pk_board) */  
* from board_tbl;
```

우리는 예전에 **pk_board**라는 이름으로 **bno**에 **primary key**를 부여했습니다.

따라서 **board_tbl**테이블의 **pk_board**를 기준으로 인덱싱을 하라는 힌트를 넘겨보겠습니다.


```
select  
/*+ INDEX_DESC(board_tbl pk_board) */  
* from board_tbl;
```

```
select * from board_tbl order by bno desc;
```



과거 저 두 구문의 결과는 달랐습니다. 위쪽 힌트를 받은 구문의 효율이 훨씬 뛰어났습니다만(11g버전 기준)

현재 사용하는 18c버전에서는 인덱싱 스캔을 기본으로 하도록 설계가 바뀌었기 때문에

위의 계획을 봐도 큰 차이는 없습니다만 위쪽이 더 효율적이었다는 사실은 알고 계시는것이 좋습니다.

```
select  
/*+ INDEX_DESC(board tbl pk_board) */  
rownum, rowid, board_tbl.* from board_tbl;
```

그렇다면 이제 인덱스는 어떻게 얻어오는가 하면

바로 **rownum**, **rowid**를 **select**문의 조회컬럼에 추가하여 조회할 수 있습니다.

단 **rownum**, **rowid**와 함께 *과같은 전체 컬럼을 조회할때는 반드시 테이블명.*로 넣어주셔야 합니다.

ROWNUM	ROWID	BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATEDATE
1	AAASfFAAMAAAByyAA7	655380	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
2	AAASfFAAMAAAByyAA6	655379	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
3	AAASfFAAMAAAByyAA5	655378	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
4	AAASfFAAMAAAByyAA4	655377	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
5	AAASfFAAMAAAByyAA3	655376	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
6	AAASfFAAMAAAByyAA2	655375	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
7	AAASfFAAMAAAByyAA1	655374	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
8	AAASfFAAMAAAByyAA0	655373	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
9	AAASfFAAMAAAByyAAz	655372	테스트글	테스트본문	글쓴이	21/09/24	21/09/24

위와 같이 **rowid**라는 저장 번지수를 통해 조회할 수 있고,

정렬된 **row**에 1번부터 번호를 매겨주는 **rownum**을 같이 조회할 수 있습니다.

붙는 순번은 **rownum**이 가장 마지막에 붙습니다.

```
select /*+ INDEX_DESC(board_tbl pk_board) */  
rownum rn, board_tbl.* from board_tbl WHERE rownum <= 20;
```

이제 저 **rownum**을 이용해 페이지네이션 구문을 작성해보겠습니다.

먼저 위 구문은 **rownum**이 20번 미만인 자료만 스캔하는 자료입니다.

만약 페이지당 10개의 글이 있다면 2페이지 분량의 글을 스캔한것입니다.

```
select * from
```

```
(select /*+ INDEX_DESC(board_tbl pk_board) */  
rownum rn, board_tbl.* from board_tbl WHERE rownum <= 20)  
where rn > 10;
```

RN	BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATED
1	655380	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
2	655379	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
3	655378	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
4	655377	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
5	655376	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
6	655375	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
7	655374	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
8	655373	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
9	655372	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
10	655371	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
11	655370	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
12	655369	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
13	655368	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
14	655367	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
15	655366	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
16	655365	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
17	655364	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
18	655363	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
19	655362	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
20	655361	테스트글	테스트본문	글쓴이	21/09/24	21/09/24

20번까지 얻은 구문을 먼저 구해놓고, 서브쿼리 형태로 감쌉니다.

지금 저렇게 작성된 시점에 rn 1~20번은 스냅샷이 찍힙니다.

```
select * from  
(select /*+ INDEX_DESC(board_tbl pk_board) */  
rownum rn, board_tbl.* from board_tbl WHERE rownum <= 20)  
where rn > 10;
```

RN	BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATEDATE
11	655370	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
12	655369	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
13	655368	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
14	655367	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
15	655366	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
16	655365	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
17	655364	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
18	655363	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
19	655362	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
20	655361	테스트글	테스트본문	글쓴이	21/09/24	21/09/24

스냅샷이 찍힌 상태에서 다시 10보다 큰 rn만 남기면

11~20번까지(2페이지)에 해당하는 글들만 조회할 수 있습니다.

```
select * from  
(select /*+ INDEX_DESC(board_tbl pk_board) */  
rownum rn, board_tbl.* from board_tbl WHERE rownum <= (5*10))  
where rn > (5-1)*10;
```

RN	BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATEDATE
41	655340	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
42	655339	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
43	655338	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
44	655337	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
45	655336	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
46	655335	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
47	655334	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
48	655333	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
49	655332	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
50	655331	테스트글	테스트본문	글쓴이	21/09/24	21/09/24

만약에 5페이지 글을 보고 싶다면, 1페이지가 1~10, 2페이지가 11~20...
5페이지는 41~50페이지 까지이므로 먼저 (페이지 * 10)을 서브쿼리 내부
조건절로, 그리고 ((페이지-1) * 10) 을((5-1) * 10 이므로 40) 바깥쪽 쿼리의
조건절로 넣으면 5페이지를 구할 수 있습니다.

```
select * from  
(select /*+ INDEX_DESC(board_tbl pk_board) */  
rownum rn, board_tbl.* from board_tbl WHERE rownum <= (5*10))  
where rn > (5-1)*10;
```

RN	BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATEDATE
41	655340	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
42	655339	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
43	655338	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
44	655337	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
45	655336	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
46	655335	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
47	655334	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
48	655333	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
49	655332	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
50	655331	테스트글	테스트본문	글쓴이	21/09/24	21/09/24

만약에 5페이지 글을 보고 싶다면, 1페이지가 1~10, 2페이지가 11~20...
5페이지는 41~50페이지 까지이므로 먼저 (페이지 *10)을 서브쿼리 내부
조건절로, 그리고 ((페이지-1) * 10) 을((5-1) * 10 이므로 40) 바깥쪽 쿼리의
조건절로 넣으면 5페이지를 구할 수 있습니다.


```
select * from  
(select /*+ INDEX_DESC(board_tbl pk_board) */  
rownum rn, board_tbl.* from board_tbl WHERE rownum <= (5*10))  
where rn > (5-1)*10;
```

RN	BNO	TITLE	CONTENT	WRITER	REGDATE	UPDATEDATE
41	655340	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
42	655339	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
43	655338	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
44	655337	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
45	655336	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
46	655335	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
47	655334	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
48	655333	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
49	655332	테스트글	테스트본문	글쓴이	21/09/24	21/09/24
50	655331	테스트글	테스트본문	글쓴이	21/09/24	21/09/24

여기까지 이해하셨다면 페이징 처리의 공식인

안쪽쿼리 : 페이지번호 * 글 개수

바깥쪽쿼리 : 페이지번호-1 * 글갯수

로만 쿼리문을 처리해주면 맞는 페이지번호대로 얻어올 수 있습니다.