

목차

- 1.** 페이징이 필요한 이유
- 2.** 페이징 분석
 - a.** 페이징 구조 분석
 - b.** 페이징 공식 도출

1. 페이지가 필요한 이유

뉴스

커뮤니티

하늘의 궤적 빛낸 코드 이벤트 진행

리뷰

게이머존

라이브/영상

게이밍/IT

게임스토어

FM26 예약 판매 중

보더랜드4 인벤게임즈 입점

5685661

[수다] 저그가 이론상 제일 사기여야하는 종족 아닌가 [14]

75

깜냥

13:55

95

0

5685660

[수다] 비틱) 플라즈마 하트 균형의 수호자들에게 [2]

74

피영

13:55

72

1

5685659

[수다] 요즘 메봉이들 답글도 안달아주고 말이야 [16]

84

아내

13:55

34

0

5685658

[수다] 스타포스 성공적

31

루이드엘

13:55

47

0

5685657

[수다] 솥솥이 아가방 어찌고 보고 충격 받음 [10]

86

말랭이

13:55

122

0

5685656

[수다] 츠츠지지 스타포스 안 보이는데 어떻게 보나요 [3]

62

Wound

13:55

37

0

5685655

[수다] 신규 에픽던전 보스 엄청새네

72

에스페라

13:54

45

0

5685654

[수다] 알색 드롭률보니까 마지막주쯤에 개비싸질거같음

13

도체리아

13:54

53

0

5685653

[수다] 귀칼골라보 1년만 늦게했다면!!!! [4]

82

엔젤릭버스터

13:54

95

0

5685652

[수다] 그 웨폰3렙단들이 문제역자너

70

문주윤

13:54

29

0

5685650

[수다] 언컨 가격 보면 풀하는 그냥 혜자로보임 [1]

45

꾸여역

13:53

77

0

5685649

[수다] 부캐 폴드메템 팔까요? ㅇㅇ 체력이 안됨 [2]

48

Wakeup

13:53

81

0

목록

글쓰기

< 이전

1

2

3

4

5

6

7

8

9

10

다음 >

제목

위 사진은 한 커뮤니티 사이트의 게시판입니다. 게시판을 만들려면 당연히 페이지를 해야 합니다.왜냐하면, 데이터를 보는 화면은 고정되어 있고 데이터는 엄청 많기 때문입니다.

모든 데이터를 화면에 보여줄 수 없기 때문에 또는 그렇다 할지라도 모든 데이터를 압축해서 보여주는 식은 옳바르지 않습니다. 이게 바로 페이징이 필요한 이유입니다.

2-a. 페이징 구조 분석

배경 모델 분석 : 저는 우선 배경 모델을 우선적으로 파악을 했습니다. 저같은 경우에는 웹개발입니다. 그럼 먼저 생각해 볼 점이 하나 있는데, 웹개발은 어떠한 배경으로 이루어지는가입니다.

웹개발은 서버-클라이언트 모델을 사용하고 있습니다. 그럼 서버-클라이언트 모델만 생각하면 되지 않느냐라고 하실수 있지만, 넓게 보면은 맞습니다만 우리는 **DB**또한 고려를 해야 합니다. 왜냐하면 데이터연동을 하기 위해서입니다.

즉, 우리는 다음과 같은 배경 모델을 생각해 볼 수 있습니다.

클라이언트-서버-DB (이하 배경모델이라 함.)

위의 사진을 통해 우린 배경모델에서 클라이언트의 역할, 서버의 역할, **DB**의 역할을 생각해야 합니다.

클라이언트 분석: 그럼 위의 사진은 어떤 사진인가요? 그냥 게시판 사진이라고 생각하실 수 있겠지만 좀더 구체적으로 말하면 웹브라우저 사진입니다. 그리고 웹브라우저는 사실상 클라이언트의 역할을 수행하고 있습니다.

클라이언트가 무엇인지 정해졌으니 클라이언트가 무엇을 요청하는지에 대해 아래와 같은 그림을 생각해 볼 수 있습니다. 브라우저 사용자의 행동(마우스 클릭)을 중점으로 생각하면 됩니다.

요청1

요청2

5685661	[수다] 저그가 이론상 제일 사기여야하는 종족 아닌가 [14]	깜냥	13:55	95	0
5685660	[수다] 비틱! 플라즈마 하트 균형의 수호자들에게 [2]	피영	13:55	72	1
5685659	[수다] 요즘 메봉이들 답글도 안달아주고 말이야 [16]	아내	13:55	34	0
5685658	[수다] 스타포스 성공적	루이드엘	13:55	47	0
5685657	[수다] 솜솜이 아가방 어찌고 보고 충격 받음 [10]	말랭이	13:55	122	0
5685656	[수다] 추추지지 스타포스 안 보이는데 어떻게 보나요 [3]	Wound	13:55	37	0
5685655	[수다] 신규 에픽던전 보스 엄청쌔네	에스페라	13:54	45	0
5685654	[수다] 알색 드롭률보니까 마지막주쯤에 개비싸질거같음	도체리아	13:54	53	0
5685653	[수다] 귀칼콜라보 1년만 늦게했다면!!!! [4]	엔젤릭버스터	13:54	95	0
5685652	[수다] 그 웨폰3렙단들이 문제역자녀	문주윤	13:54	29	0
5685650	[수다] 언컨 가격 보면 풀하는 그냥 혜자로보임 [1]	꾸여억	13:53	77	0
5685649	[수다] 부캐 풀드메템 팔까요? ㅇㅇ 체력이 안됨 [2]	Wakeup	13:53	81	0

목록

글쓰기

< 이전 1 2 3 4 5 6 7 8 9 10 다음 >

제목

Q

크게 **2**가지 요청으로 나뉘게 됩니다. **요청1**은 페이징 이후의 특정 게시판에 대한 데이터를 요청하고, **요청2**는 페이징을 하기위해 해당 페이지 번호를 요청을 하게 됩니다. **요청1**은 페이징 이후의 처리이므로 우리는 **요청2**를 먼저 고려해야 합니다.

서버 분석 : 클라이언트의 역할이 정해 졌으니 서버의 역할을 찾아야 합니다. 그런데 클라이언트에서 요청한게 페이지번호 하나뿐입니다.

어떻게 해야 할지 막막합니다. 게다가 옆친데 똥친격으로 또 고려해야 할 사항이 있는데, 각각의 파란색 테두리의 보여주는 개수(게시글 보여주는 개수, 페이지 보여주는 개수) 또한 변수이므로 고려해야 합니다.

이로써 총 **3**개의 변수를 생각해야 합니다.

3개의 변수는 다음과 같이 나뉩니다.

- 서버용 변수 : 게시글 보여주는 개수, 페이지 보여주는 개수
- 클라이언트 요청 변수 : 페이지 번호

그리고 공식 도출에서 설명하겠지만 사실상 페이징에서는 서버쪽이 메인이라 생각하면 됩니다.

***페이징 공식 존재 여부에 대한 생각** : 규칙이 있는 곳에는 웬만하면 공식이 있다라는 생각으로 접근을 했습니다.

***페이징 변수 위치 여부** : 위의 서버용 변수를 클라이언트에 할당해도 되나 이러한 데이터들은 보안상의 이유로 서버쪽에 놓는 게 좋다 생각했습니다.

DB분석 : **DB**는 서버의 요청으로 데이터를 얼마큼 가져올 것인가를 생각해야 합니다. 그럼 어떻게 얼마큼 가져올것에 대해 설명을 하자면,

우선 **DB** 테이블은 **2차원**인데 우린 데이터를 가져오는것만이 목표이므로 **1차원(배열)**으로 생각해 해결해 나가는게 좋다 생각이 듭니다.

즉, 전체 테이블을 하나의 배열로 생각해 각 페이지별로 구역을 정해 데이터를 가져오는게 **DB**의 역할입니다.

2-b. 페이징 공식 도출

- 변수 설정
 - **limitBoardNum(> 0)** : 게시글 보여지는 개수(서버용 변수)
 - **limitPageNum(> 0)** : 페이지 보여주는 개수(서버용 변수)
 - **totalBoardNum(>= 0)** : 전체 게시글 갯수(서버용 변수, DB연동 필요)
 - **reqPage(integer)** : 요청 페이지

***totalBoardNum** 설명 : 이 변수는 제가 처음에 뭘 할지 모르다가 나중에 생각해낸 변수입니다. 최대페이지등을 구할 때 사용 됩니다.

페이징을 통해 구역을 일정하게 나누려면 당연히 나눌대상이 필요한데 이 대상이 **totalBoardNum** 입니다.

최대 페이지 도출 방법

totalBoardNum	totalBoardNum / limitBoardNum	maxPage
0 = limitBoardNum * 0	0	1
limitBoardNum * 0 + 1	0	1
~	~	~
limitBoardNum * 1	1	1
limitBoardNum * 1 + 1	1	2
~	~	~
limitBoardNum * 2	2	2
limitBoardNum * 2 + 1	2	3
~	~	~
limitBoardNum * 3	3	3

설명 : totalBoardNum을 생각할 때 단순히 **0, 1, 2, 3, 4, ...** 이렇게 보는게 아니라 구역을 나누기 위해 위와 같이 규칙을 두어 만드는데 좋습니다.

빨간색 테두리 영역은 개수가 **1**개라 일반적인 테두리 영역과 다릅니다만, 빨간색 테두리 영역이 있는데 끝부분만 남기고 나머지는 잘렸다고 생각하시면 됩니다.

그리고 **n**을 각 테두리 영역의 시퀀스(**>= 0**)라 가정을 한다면 다음과 같은 식이 나오게 됩니다.

each totalBoardNum range = limitBoardNum * (n - 1) + x,
(1 <= x <= limitBoardNum)

그런다음 **each totalBoardNum range**를 **limitBoardNum**로 나머지 연산을 하게 되면

$1 \leq x < \text{limitBoardNum}$ 경우에는 나머지가 **0**이 아니며, **$x = \text{limitBoardNum}$** 인 경우에는 나머지가 **0**입니다. 즉, **$\text{totalBoardNum} \% \text{limitBoardNum}$** 값이 **0**인지 아닌지에 따라 **maxPage**가 결정됩니다. 따라서, **maxPage**는 다음과 같이 쓸 수 있습니다.

$\text{maxPage} = (\text{totalBoardNum} / \text{limitBoardNum}) + 1$
when $\text{totalBoardNum} \% \text{limitBoardNum}$ is not 0.

$\text{maxPage} = \text{totalBoardNum} / \text{limitBoardNum}$
when $\text{totalBoardNum} \% \text{limitBoardNum}$ is 0.

가져오는 범위 도출 방법

range	reqPage
$0 \sim \text{limitBoardNum} - 1$	1
$\text{limitBoardNum} \sim 2 * \text{limitBoardNum} - 1$	2
$2 * \text{limitBoardNum} \sim 3 * \text{limitBoardNum} - 1$	3

위 표는 아래표와 같습니다.

range	reqPage
$(\text{reqPage} - 1) * \text{limitBoardNum} \sim \text{reqPage} * \text{limitBoardNum} - 1$	1
$(\text{reqPage} - 1) * \text{limitBoardNum} \sim \text{reqPage} * \text{limitBoardNum} - 1$	2
$(\text{reqPage} - 1) * \text{limitBoardNum} \sim$	3

$\text{reqPage} * \text{limitBoardNum} - 1$	
---	--

따라서,

$\text{startIdx} = (\text{reqPage} - 1) * \text{limitBoardNum}$

$\text{endIdx} = \text{reqPage} * \text{limitBoardNum} - 1$

***추가설명 :** **startIdx** 와 **endIdx** 는 가져오는 범위를 나타내는데 전체를 가져오는 범위를 의미합니다. 이렇게 하면 **out of range** 와 같은 오류가 날 수가 있으나 **ORACLE DBMS**같은 경우에는 **BETWEEN** 절에 **out of range**가 발생하더라도 오류가 나지 않으니 오류체크를 해도되고 안해도 됩니다.

시작 페이지 도출 방법

startPage	reqPage
$1 = \text{limitPageNum} * 0 + 1$	$1 = \text{limitPageNum} * 0 + 1$
~	~
$1 = \text{limitPageNum} * 0 + 1$	$\text{limitPageNum} * 1$
$\text{limitPageNum} * 1 + 1$	$\text{limitPageNum} * 1 + 1$
~	~
$\text{limitPageNum} * 1 + 1$	$\text{limitPageNum} * 2$
$\text{limitPageNum} * 2 + 1$	$\text{limitPageNum} * 2 + 1$
~	~
$\text{limitPageNum} * 2 + 1$	$\text{limitPageNum} * 3$

설명 : 테두리 영역의 시퀀스를 **n (>= 0)** 이라 가정한다면,

startPage = limitPageNum * n + 1 이 됩니다. 하지만, n은 임의의
가정된 수 이기 때문에 n을 다른 실질적인 변수로 치환해줘야 합니다.
 또한, 위의 표에서 다음과 같은 식을 얻을 수 있습니다.

$$\text{each reqPage - 1 range} = \text{limitPageNum} * n + x, \\ (0 \leq x < \text{limitPageNum})$$

reqPage - 1을 한 이유는 동일한 값을 얻기 위해서 입니다.

그러면, 해당 식을 나누기(몫) 연산으로 나눠주면 됩니다. 나머지 연산이
아닌 나누기 연산을 하는 이유는 나머지 연산은 자연수 등차수열에
대해서 각각의 구역에 대해 일정한 값을 갖지 못하지만 나누기 연산은
그것이 가능하기 때문에 나누기 연산을 해야 합니다.

나누기 연산을 하게 되면 다음과 같은 식을 얻을 수 있습니다.

$$n = ((\text{reqPage} - 1) / \text{limitPageNum})$$

따라서, 시작 페이지는 다음의 식으로 쓸 수 있습니다.

$$\text{startPage} = \text{limitPageNum} * ((\text{reqPage} - 1) / \text{limitPageNum}) + 1$$

끝, 이전, 다음 페이지 도출 방법

endPage	reqPage
limitPageNum * 1	1 = limitPageNum * 0 + 1
~	~
limitPageNum * 1	limitPageNum * 1
limitPageNum * 2	limitPageNum * 1 + 1
~	~
limitPageNum * 2	limitPageNum * 2
limitPageNum * 3	limitPageNum * 2 + 1
~	~

limitPageNum * 3	limitPageNum * 3
-------------------------	-------------------------

설명 : 시작페이지 도출 방법과 유사하기 때문에 자세한 설명은 생략하겠습니다. 테두리 영역 시퀀스 **n (>= 1)** 에 대하여,

endPage = limitPageNum * n,

each reqPage - 1 range = limitPageNum * (n - 1) + x, (0 <= x < limitPageNum),

(reqPage - 1) / limitPageNum = n - 1

endPage = limitPageNum * (((reqPage - 1) / limitPageNum) + 1)

다음, 이전 페이지는 쉽습니다.

nextPage = endPage + 1

prevPage = startPage - 1