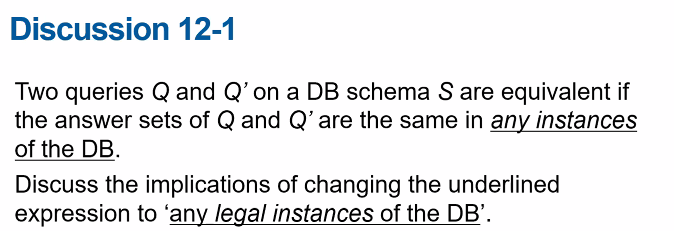
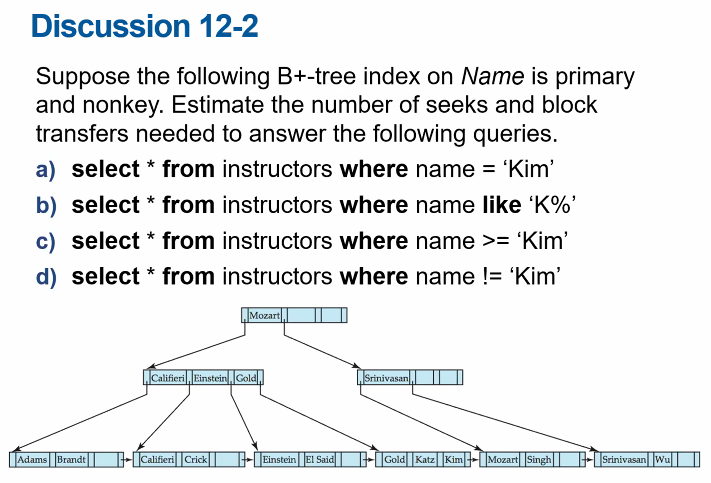
**Discussion 05/27**



Legal instance는 integrity constraints를 만족하는 instance들…

Schema를 정의하고 record를 insert, update, delete 하는 과정에서 모든 instance가 legal이지 않을까? 그러면 legal instance만 확인해도 될 것 같고 효율성이 더 높아질 것 같다.

=> query processing을 하는 시점에서 legal인지 아닌지의 여부는 이미 판단이 되어 있음. Legal instance에서 equivalent한 경우는 훨씬 많아짐. 그래서 선택지가 더 많아지기 때문에 더 많은 선택지 중 cost가 싼 걸 고르면 됨.

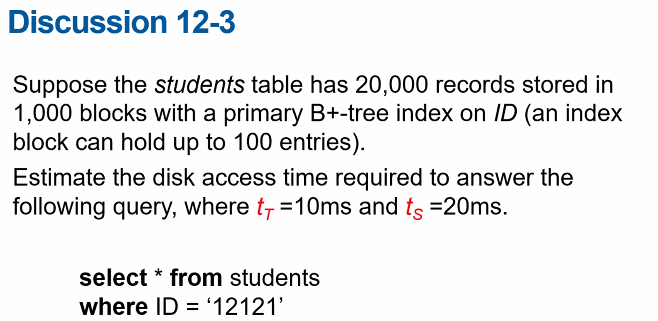


a) 높이가 3 이므로.. estimated seek은 4번. Block transfer는 3 + n번 . n은 kim을 찾는 데 필요한 block들의 개수.

b) seek은 3번. K로 시작하는 record가 x개라고 하면 block transfer는 3 + x번.

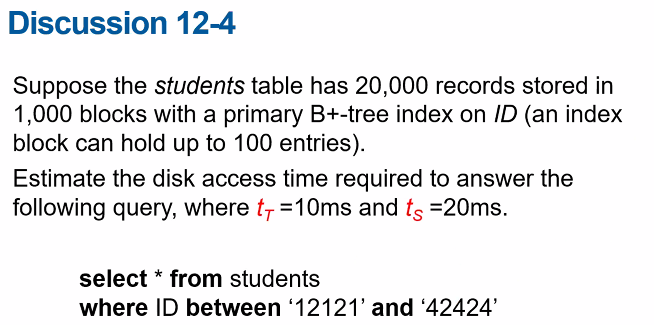
c) seek는 3번. KIM보다 큰 record가 x개면 block transfer는 3+X번.

=> 확실하지 않음..



B+트리의 높이는 log\_50 (20000) 이니까 3. Primary index이고 key이기 때문에

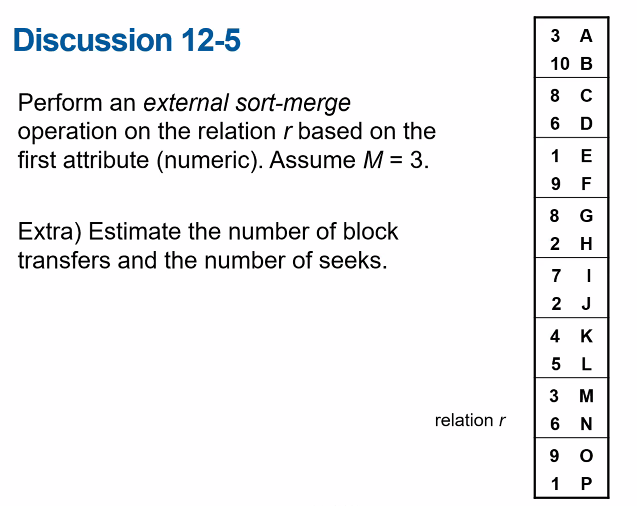
Cost = (4) \* (10ms + 20ms) = 120ms.



위의 문제와 같고, seek는 12121에 대해 index로 수행하면 4번. 여기서부터 순차적으로 relation을 스캔하니까 seek는 여기서 끝. Block transfer는 트리에서 3번 + 42424까지의 개수를 n이라고 하면 n개를 돌아야 하므로 3 + n.

Cost = 4 \* 10ms + (3 + n) \* 20ms.

=> n을 어떻게 estimate할 것인가? # of distinct value를 알고 있으면 좋음. 그리고 전체 구간 대비 해당 구간의 비율을 알면 2만개의 record에서 비율을 구해서 예상 개수를 구함.



1. 처음에 run을 만들면

3개의 block이 되니까 (1E 3A 6D 8C 9F 10B), (2H 2J 4K 5L 7I 8G), (1P 3M 6N 9O)의 run들이 만들어짐.

2. N이 3이므로 merge pass.

(1E 3A 6D 8C 9F 10B), (2H 2J 4K 5L 7I 8G) 둘이 합쳐져서 (1E 2H 2J 3A 4K 5L 6D 7I 8C 8G 10B)가 되고, 뒤에껀 그대로 (1P 3M 6N 9O). 그리고 이 둘이 합쳐져서 최종적으로 (1E 1P 2H 2J 3A 3M 4K 5L 6D 6N 7I 8C 8G 9O 10B)가 됨.

Block transfer는 . 즉 b\_r \* (2\*pass + 1)인데 b\_r = 8, pass = 2이므로 40.

Seek는 . 2\*[8/3] + [8/1]\*(3) = 6 + 24 = 30. b\_b는 1.