

그림으로 공부하는 오라클 구조 -

Ch1. I/O와 디스크의 관계

오라클을 이해하기 위한 필수 키워드

1. 병렬 처리를 가능케 하고 높은 처리량을 실현한다.
2. 응답(response)을 중시한다.
3. 커밋한 데이터는 지킨다.

오라클의 동작 원리를 공부할 때 위의 3가지 관점을 가지고 공부하면 이해하기에 좋다.

디스크의 동작

디스크에 데이터를 넣고 꺼내는 I/O작업이 메모리에 비해 현저히 느리기 때문에 디스크 I/O 작업은 최소화 되어야 한다. 이를 이해하기 위해서는 디스크와 메모리의 차이를 알아야 한다.

메모리(RAM)은 데이터에 대한 빠른 접근을 위해 데이터를 임시로 저장하고 있는 저장소이다. 컴퓨터가 꺼지면 들고 있던 데이터가 사라지는 단점이 있지만 데이터에 대한 빠른 접근을 가능하게 한다.

반면에 디스크는 HDD, SSD와 같이 데이터를 영구적으로 저장하는 저장 장치이다. 메모리의 데이터에 접근하는 것은 전기적 신호로 이루어지지만 디스크의 데이터에 접근하는 것은 암(Arm)을 움직이게 하는 기계적인 동작이 필요하기 때문에 메모리와 비교해서 상당히 느리다.

따라서, 디스크에 접근하는 I/O의 시간과 횟수를 줄여야 시스템의 성능이 좋아진다.

어떻게 디스크 I/O 대기시간을 줄일까?

1. 인덱스 사용하기 : 데이터베이스의 인덱스를 통해 처음부터 모든 데이터를 훑지 않고 원하는 데이터가 어디 있는지 인덱스를 통해 확인한 후 원하는 랜덤 액세스 방식으로 데이터를 찾아간다. 이렇게 하면 데이터를 full scan하는 시퀀셜 액세스 방식보다 빠르게 데이터를 찾을 수 있다.
2. 랜덤 액세스 : 인덱스로 확인한 ROWID를 통해 데이터를 찾아갈 때 한 번에 하나의 블록만 액세스하는 것을 랜덤 액세스라고 한다. 이 때, 찾으려는 데이터가 항상 물리적으로 연속적인 공간에 저장되어 있지는 않으므로 디스크를 회전시켜가며 찾아야 한다. 그래서 적은 양의 데이터에 접근할 때에는 한 번에 하나의 블록만 특정하여 접근하는 랜덤 액세스 방식이 좋으나 데이터의 양이 전체 데이터의 15% 이상(절대치 아님)되면 디스크 계속 회전시켜 왔다갔다 하는 랜덤 액세스 방식보다는 시퀀셜 액세스(데이터를 그냥 처

음부터 훑어 가며 여러 블록 씩 접근하는 것)방식이 효율적일 수 있다. 즉 **인덱스를 사용한 액세스가 무조건 효율적인 것은 아니다!**

※오라클은 랜덤 액세스를 '**db file sequential read**'라고 표시하며, 시퀀셜 액세스는 '**db file scattered read**'라고 표시한다.

- 표현이 정반대로 되어 있는 것 같은데 이는 메모리에 데이터가 적재되는 방식이 다르기 때문이다.
- 오라클은 데이터를 블록 단위로 메모리에 배치한다.
- 랜덤 액세스는 1개의 블록만을 읽어 오므로 읽어온 데이터(블록)은 메모리에서 연속이다 (sequential)
- 시퀀셜 액세스는 여러 개의 블록을 읽어 와 메모리에 분산적으로(scattered) 배치된다.
- 따라서 위와 같이 용어의 혼동이 발생하는 것이다.
- 정리하자면 랜덤 액세스와 시퀀셜 액세스는 '디스크'에서 데이터에 접근하는 방식의 구분이며, **db file sequential read**와 **db file scattered read**은 가져온 데이터(블록)을 메모리에 적재하는 방식을 구분한 것이다.