

## 1. Join Method 의 선택

대용량의 데이터라면 당연히 Hash Join 을 이용해야 한다.

해쉬조인은 해싱함수 기법을 활용하여 조인을 수행하는 방식으로 Nested Loop 조인이 가진 랜덤 액세스의 단점과 Sort Merge 조인이 가진 정렬의 부담을 해결할 수 있는 대안이다.

랜덤의 부담은 최악의 경우 하나의 로우를 액세스하기 위해 한 블록씩 액세스를 해야 하고, 그 블록이 아직 디스크에 위치하고 있을 수도 있다는 점이다. 정렬의 부담은 정렬처리 그 자체의 부담이라기 보다는 메모리 내에 지정된 정렬영역보다 정렬할 크기가 지나치게 큰 경우에 발생하는 기하급수적인 부담을 말한다.

解

↖ NL Join은 조인 조건으로부터 많은 영향을 받는다.

해쉬조인의 가장 큰 장점은 조인의 연결고리의 인덱스 존재유무에 영향을 받지 않는다는 점이다.

따라서 인라인뷰를 이용한 중간집합을 만들어 가면서 집합의 연산을 통해 결과 집합을 생성해 가다보면 중간에 생성되는 집합들은 일종의 임시 테이블이라고 할 수 있으므로 당연히 인덱스를 가질 수 없다. 이러한 처리에서 해쉬조인의 가치는 크게 부각될 수 밖에 없다.