Section7. 조인의 원리

- 조인의 원리
  - 중첩 루프조인, 정렬 병합 조인, 해시조인

4.7.1. 중첩 루프조인

- 중첩 for문과 같은 원리로, 조건에 맞는 조인을 하는 방법.
- 랜덤 접근에 대한 비용이 많이 증가하므로, 대용량 테이블에서는 사용하지 않음.

의사 코드

```
for each row in t1 matching reference key {
    for each row in t2 matching reference key {
        if row satisfies join conditions, send to client
    }
}
```

- 예시
  - t1, t2 테이블 조인 시,
    - 첫 테이블에서 행을 한번에 하나씩 읽고,
    - 그 다음 테이블에서도 행을 하나씩 읽어 조건에 맞는 레코드를 찾아 결괏값을 반환
- 중첩 루프조인에서 발전한 조인할 테이블을 작은 블록으로 나눠서,
  - 블록 하나씩 조인하는 블록 중첩 루프 조인 (BNL , Block Nested Loop) 방식도 존재

4.7.2 정렬 병합 조인

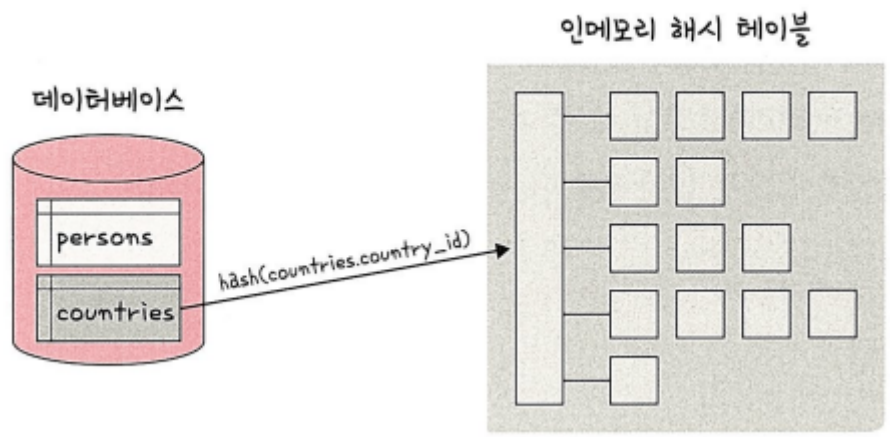
- 각각의 테이블을 조인할 필드 기준으로 정렬하고, 정렬이 끝난 이후에 조인 작업을 수행하는 조인
- 조인할 때 쓸 적절한 인덱스가 없고,
  - 대용량의 테이블들을 조인하고 조인 조건으로 <, > 등 범위 비교 연산자가 있을 때 사용

4.7.3. 해시 조인

- 해시테이블을 기반으로 조인하는 방법.
- 두 개의 테이블을 조인한다고 할 때,
  - 하나의 테이블이 메모리에 온전히 들어가면 보통 중첩 루프조인보다 효율적이다.
  - 단, 동등 조인에서만 사용 가능.
- MySQL의 해시조인 단계는 2단계로 나뉨
  - 1단계 : 빌드 단계
    - 입력 테이블 중 하나를 기반으로, 메모리 내 해시 테이블을 빌드하는 단계.
    - 보통 바이트가 더 작은 테이블을 기반으로 테이블을 빌드.

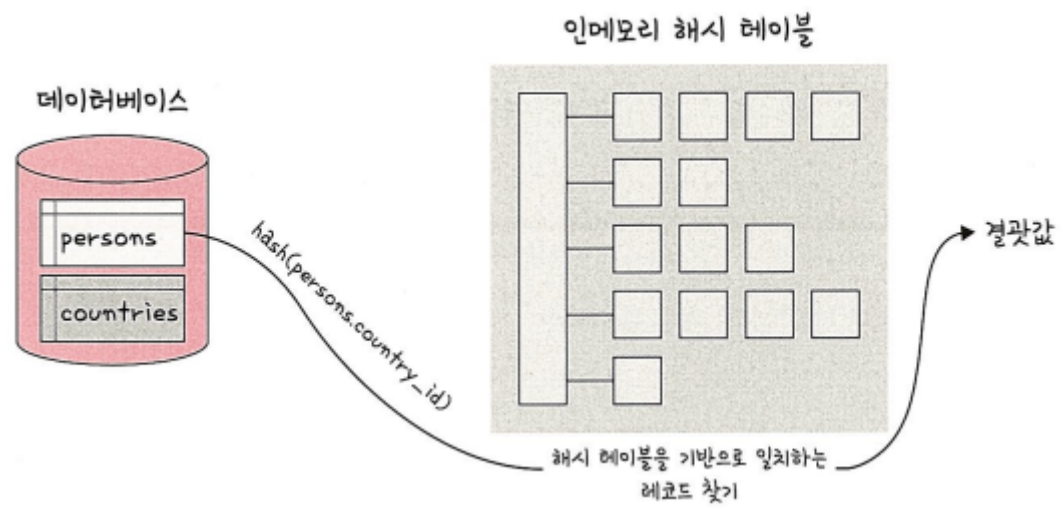
- 조인에 사용되는 필드가 해시 테이블의 키로 사용됨.

▼그림 4-40 빌드 단계



- 2단계 : 프로브 단계
  - 레코드 읽기를 시작하는 단계.
  - 각 레코드에서 키와 일치하는 레코드를 찾아서 결괏값으로 반환.
  - 각 테이블은 한번 씩만 읽게되어서,
    - 중첩해서 두 개의 테이블을 읽는 중첩 루프 조인보다 보통 성능이 더 좋다.
  - 이때, 사용 가능한 메모리 양은 시스템 변수 join_buffer_size에 의해 제어되고, 런타임시 조정가능.

▼그림 4-41 프로브 단계



예상 질문!

데이터베이스란?

- 일정한 규칙 혹은 규약을 통해 구조화되어 저장되는 데이터의 모음.
- 해당 데이터베이스를 제어, 관리하는 통합시스템을 DBMS라고 함.
- 데이터베이스의 데이터들을 특정 DBMS마다 정의된 쿼리언어를 통해 삽입, 삭제, 수정, 조회 등을 수행할 수 있다.
- 실시간 접근과 동시 공유가 가능하다는 특징이 있다.

중첩 루프조인이란?

- NLJ, Nested Loop Join
- 중첩 for문과 같이 원리로 조건에 맞는 조인을 하는 방법.
- 랜덤 접근에 대한 비용이 많이 증가하므로, 대용량 테이블에서는 사용하지 않음.
- t1, t2 테이블을 조인한다라고 하면, 첫 번째 테이블에서 행을 한번에 하나씩 읽고, 그 다음 테이블에서도 행을 하나씩 읽어 조건에 맞는 레코드를 찾아 결괏값을 반환한다.

인덱스를 매 필드마다 설정하는 것이 좋을까?

- 인덱스는 두번의 탐색을 강요하므로, 읽기관련 비용이 더 들게된다.
- 매 필드마다 설정하는 것은 고찰을 해봐야 알 수 있다.
    - 또한, 테이블 수정시 인덱스도 수정되어야 하므로, B트리 높이 조정 비용, 데이터 분산 비용도 소모됨.
- 컬렉션에서 가져와야 하는 양이 많을수록 인덱스를 사용하는 것은 비효율적이다 !