

# 조인의 원리

## 중첩 루프 조인 (NLJ, Nested Loop Join)

중첩 for 문과 같은 원리로, 조건에 맞는 조인을 하는 방법

랜덤 접근에 대한 비용이 많이 증가하므로 대용량의 테이블에서는 사용하지 않음

### ex) t1, t2 테이블을 조인하는 경우

첫번째 테이블에서 행을 하나씩 읽고 그 다음 테이블에서도 행을 하나씩 읽어 조건에 맞는 레코드를 찾아 결과값을 반환함

#### 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  
    }  
}
```

중첩 루트 조인에서 발전한 조인할 테이블을 작은 블록으로 나눠서 블록 하나 씩 조인하는

블록 중첩 루프 조인(BNL, Block Nested Loop) 라는 방식도 존재함

## 정렬 병합 조인

각각의 테이블을 조인할 필드 기준으로 정렬하고 정렬이 끝난 후에 조인 작업을 수행하는 조인

조인 시 쓸 적절한 인덱스가 없고 대용량의 테이블들을 조인하고 조인 조건으로 <, > 등 범위 비교 연산자가 있을 때 사용

## 해시 조인

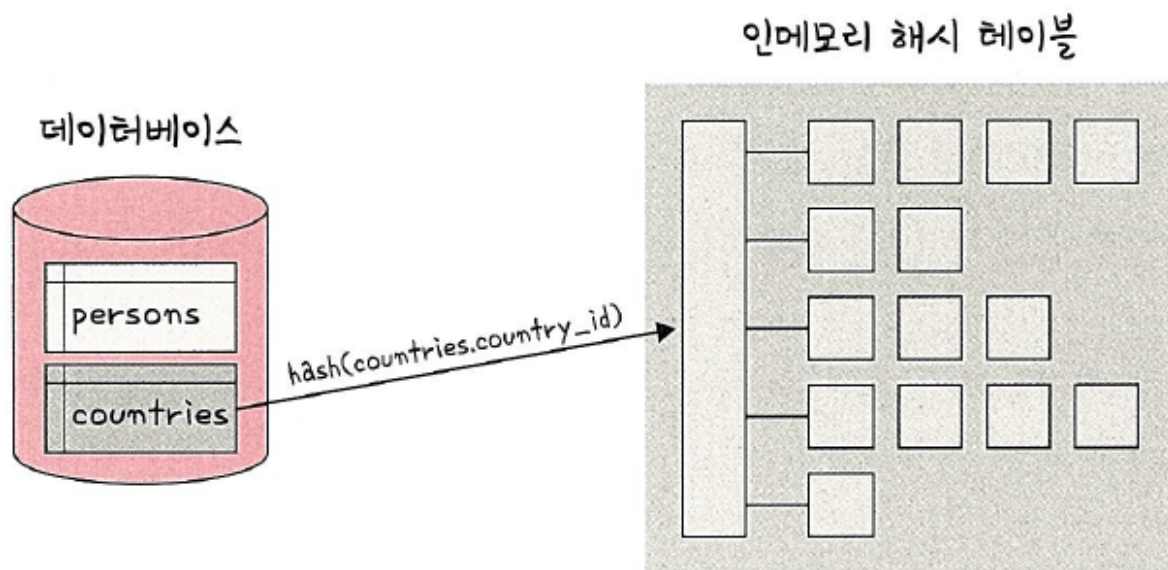
해시 테이블을 기반으로 조인하는 방법

두 개의 테이블을 조인하는 경우 하나의 테이블이 메모리에 온전히 들어간다면, 보통 중첩 루프 조인보다 더 효율적임 (메모리에 올릴 수 없을 정도로 큰 경우, 디스크 사용 비용이 발생함)

동등 조인에서만 사용할 수 있음

MySQL의 경우 8.0.18 이후 사용할 수 있게 되었으며, 빌드 단계, 프로브 단계로 나뉨

## 빌드 단계



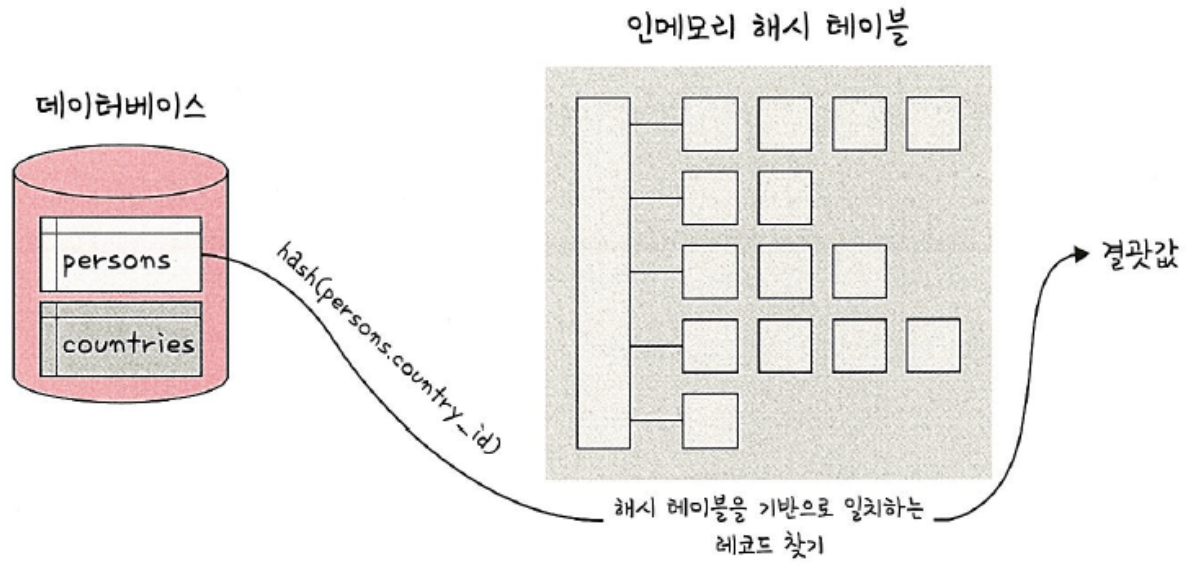
입력 테이블 중 하나를 기반으로 메모리 내 해시 테이블을 빌드하는 단계

ex) persons와 countries 라는 테이블을 조인하는 경우

둘 중에 바이트가 더 작은 테이블을 기반으로 해서 테이블을 빌드함

조인에 사용되는 필드가 해시 테이블의 키로 사용됨

## 프로브 단계



이 단계 동안 레코드 읽기를 시작하며, 각 레코드에서 `persons.country_id` 에 일치하는 레코드를 찾아서 결과값으로 반환함

이를 통해 각 테이블은 한번씩만 읽게 되어 중첩해서 두개의 테이블을 읽는 중첩 루프 조인보다 보통은 성능이 좋음

사용 가능한 메모리 양은 시스템 변수 `join_buffer_size` 에 의해 제어됨. 런타임 시 조정 가능