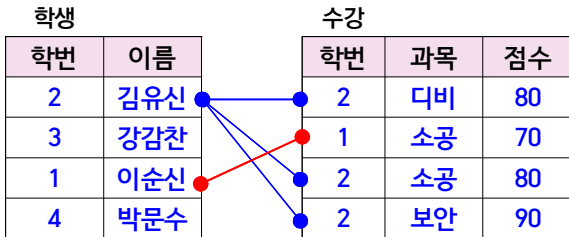


## 20. 여러 가지 조인

### // 중첩루프조인(nested loop join)

- 선행테이블에서 조인할 레코드를 차례로 하나씩 읽으면서
- 후행테이블도 레코드를 차례로 읽으면서 조인조건을 만족하는 레코드를 찾아 조인한다.
- 각 테이블은 정렬되어 있지 않아도 된다.(정렬되어 있으면 더 빠르게 조인할 수 있음)
- 후행테이블이 정렬 또는 인덱스가 없으면 순차탐색을 계속해야 하므로 성능 저하!

Driving table	<ul style="list-style-type: none"> <li>• 조인할 때 먼저 접근되는 테이블 - 선행테이블(외부테이블)</li> <li>• 조인할 두 테이블 중 작은 테이블을 선행테이블로 설정한다.</li> </ul>
Driven table	<ul style="list-style-type: none"> <li>• 조인할 때 나중에 접근되는 테이블 - 후행테이블(내부테이블)</li> <li>• 조인을 위한 적절한 인덱스가 생성되어 있어야 한다.(효율적인 조인)</li> </ul>



중첩루프조인 ↓ 학생 × 학생.학번 = 수강.학번 수강

학번	이름	과목	점수
2	김유신	디비	80
2	김유신	소공	80
2	김유신	보안	90
1	이순신	소공	70

### // 정렬-합병조인(sort-merge join)

- 두 테이블이 정렬되어 있지 않으면, 조인 속성 기준으로 오름차순 정렬 후에 조인한다.
- 두 테이블의 정렬된 조인 속성 키를 scan하면서 조인 결과를 생성한다.
- 정렬되어 있으므로 보다 빠르게 조인할 수 있다.
- 정렬 작업 비용이 발생된다.

// 단일루프조인(single loop join) - 색인된 중첩루프조인

- 후행테이블의 조인 속성에 인덱스가 존재할 경우 사용 - 물리적인 정렬을 이용하지 않는다.
- 선행테이블의 각 레코드에 대하여 후행테이블의 인덱스를 사용하여 직접 검색 후 조인하는 방식
- 테이블 S에 인덱스가 존재할 경우
- 테이블 R의 각 레코드에 대하여 S의 인덱스를 사용하여 S의 레코드 중에서 조인조건을 만족하는 레코드를 직접 검색 - 인덱스를 이용하여 직접 검색이 가능하므로 성능 향상

정렬(sort)	<ul style="list-style-type: none"> <li>• 테이블을 정렬하면 행 순서가 원본과 다를 수 있는 테이블의 복사본 생성</li> <li>• 테이블 크기가 클수록 더 많은 기억공간 필요 - 단점</li> </ul>
인덱스(index)	<ul style="list-style-type: none"> <li>• 테이블에서 하나 이상의 열을 사용하여 인덱스를 만들 수 있다.</li> <li>• 인덱스는 별도의 파일에 저장된다.(인덱스 파일)</li> </ul>

- 정렬과 인덱스는 데이터베이스 테이블에서 데이터 검색 속도를 향상시키는 데 사용된다.
- 단일루프조인은 물리적인 정렬을 이용하지 않고, 인덱스를 이용하여 검색 속도를 향상시킨다.

// 해시조인

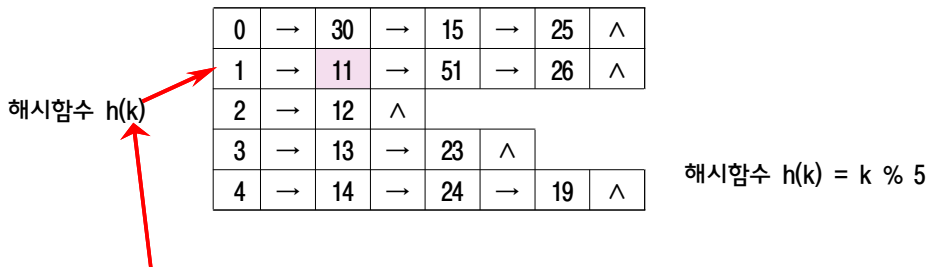
- 해시조인은 해시함수를 사용하여 두 테이블의 자료를 결합하는 조인 방식이다.

작은 테이블 (build input)	<ul style="list-style-type: none"> <li>• 두 테이블 중 작은 테이블을 읽어 해시영역(hash area)에 해시표를 생성한다.</li> <li>• 해시함수에서 리턴 받은 버킷주소의 해시체인에 엔트리를 연결한다.(해시표 작성)</li> <li>• 즉, 두 테이블 중에서 작은 테이블에 대한 해시표만 메모리에 올린다.</li> <li>• 작은 테이블을 생성입력(build input) 또는 드라이빙 테이블이라고도 한다.</li> </ul>
큰 테이블 (probe input)	<ul style="list-style-type: none"> <li>• 큰 테이블을 읽으면서 메모리에 올려진 작은 테이블과 비교하면서 조인한다.</li> <li>• 해서, 큰 테이블을 검증입력(probe input)이라 한다.</li> </ul>

작은 테이블 

15	12	13	14	11	30	51	23	24	19	25	26
----	----	----	----	----	----	----	----	----	----	----	----

↓ 해시표를 생성



// 해시조인 정리

- 해시조인은 한 테이블이 **해시영역(hash area)**에 저장될 정도로 충분히 작아야 효율적이다.
- **해시영역**은 해시조인을 수행하기 위한 메모리 내에 만들어진 영역이다.
- 해시조인은 **해시표**를 생성하는 비용이 수반되므로 **작은 테이블**이 충분히 작아야 효율적이다.
- 해시조인은 관계형 데이터베이스에서 **가장 많은 비용**이 들어가는 조인 방법이다.
- 해시표 생성 및 탐색할 때 모두 해시함수를 사용한다.
- 해시조인은 **대량**의 데이터일 때도 효과가 좋다.
- 해시조인은 CPU 및 Memory 사용을 많이 요구한다.
- 해시조인은 **동등조인(equi-join)**만 적용 가능하다. (**연산자** =만 사용 가능)

**기출문제 분석**

1. 릴레이션 R과 S에 대한 조인 연산의 구현 방법을 설명한 것으로 옳지 않은 것은? [2023년 국가 7급]

- ① 중첩루프조인(nested loop join): R과 S의 레코드들에 대한 별도의 인덱스나 정렬을 필요로 하지 않는다.
- ② 단일루프조인(single loop join): R과 S 중 하나의 릴레이션에 있는 레코드들은 조인 속성 값에 따라 물리적으로 정렬되어 있어야 한다.
- ③ 정렬-합병조인(sort-merge join): R과 S의 레코드들이 조인 속성 값에 따라 물리적으로 정렬되어 있을 때 가장 효율적이다.
- ④ 해시조인(hash join): R과 S의 레코드들은 조인 속성을 해시 키로 하고 동일한 해시함수를 이용해 해시되어 있어야 한다.

☞ 조인

- 
- **단일루프조인(single loop join):** R과 S 중 하나의 릴레이션에 있는 레코드들은 조인 속성 값에 따라 물리적으로 정렬되어 있어야 한다.(x)  
→ **단일루프조인:** 물리적인 정렬이 아니고, 조인 속성 값에 따라 **인덱스**가 생성되어 있어야 한다.
- 

정답 : ②