인덱스란?

레코드를 빠르게 찾을 수 있도록
도와주는 보조 화일

화일에 대한 또 다른 접근 경로

인덱스 엔트리

인덱스 화일의 레코드

인덱스 엔트리의 구조

[키 값들, 레코드 주소]

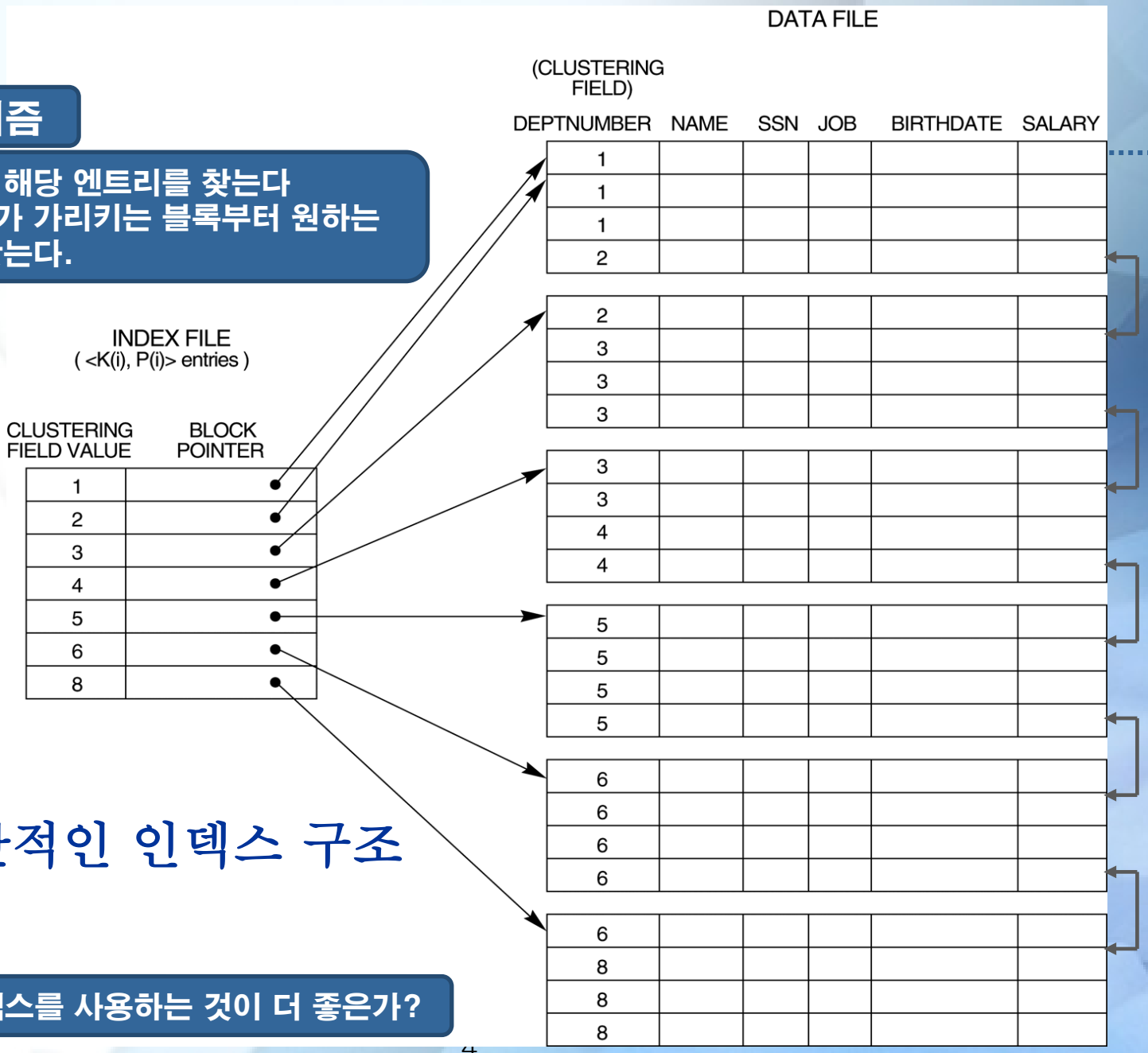
인덱스 엔트리의 구조

[키 값들, 레코드 주소]

일반적인 인덱스 구조

검색 알고리즘

1. 인덱스에서 해당 엔트리를 찾는다
2. 블록포인터가 가리키는 블록부터 원하는 레코드를 찾는다.



일반적인 인덱스 구조

검색할 때 인덱스를 사용하는 것이 더 좋은가?

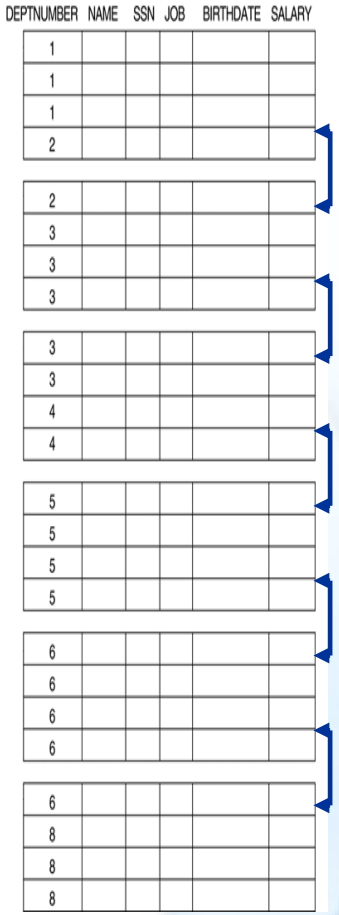
검색 : 순서화일 vs 인덱스

순서화일의 **처음부터** 순차검색 한다.

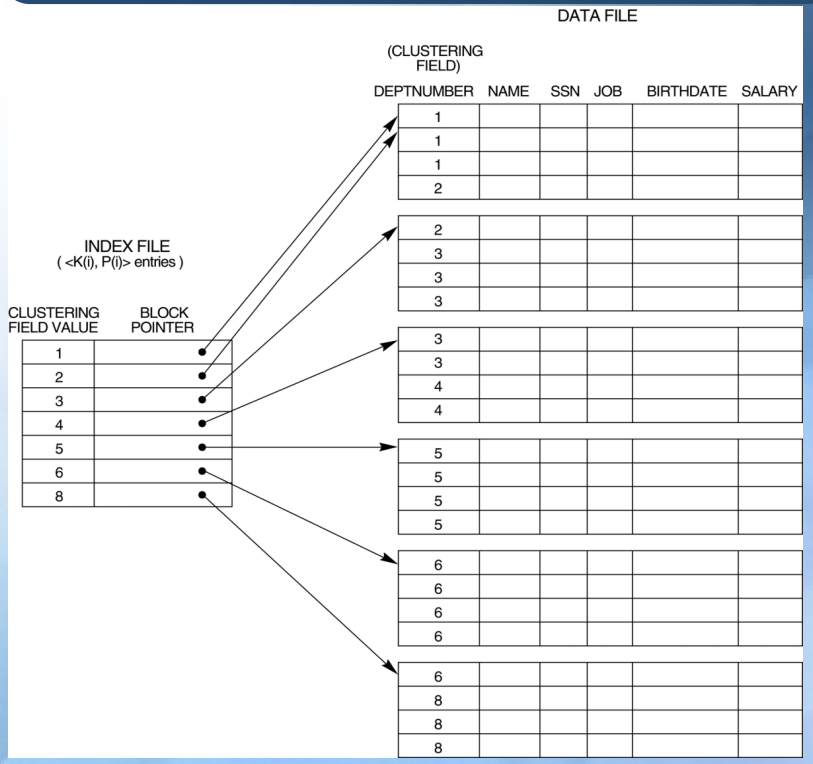
Search 1

Search 3

Search 8

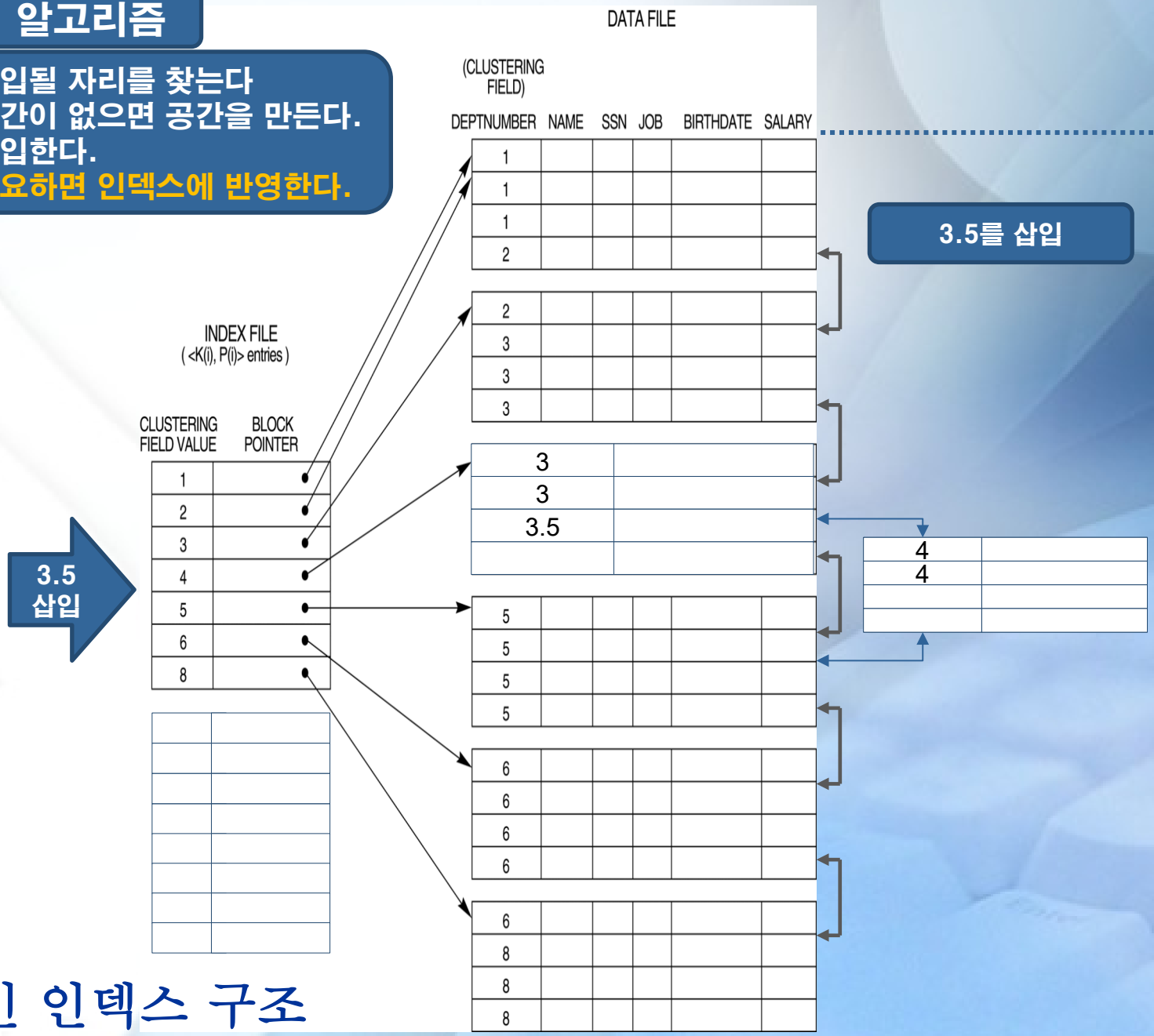


1. 인덱스에서 해당 엔트리를 찾는다
2. **블록포인터가 가리키는 블록부터** 순차검색 한다.



삽입 알고리즘

1. 삽입될 자리를 찾는다
2. 공간이 없으면 공간을 만든다.
3. 삽입한다.
4. 필요하면 인덱스에 반영한다.

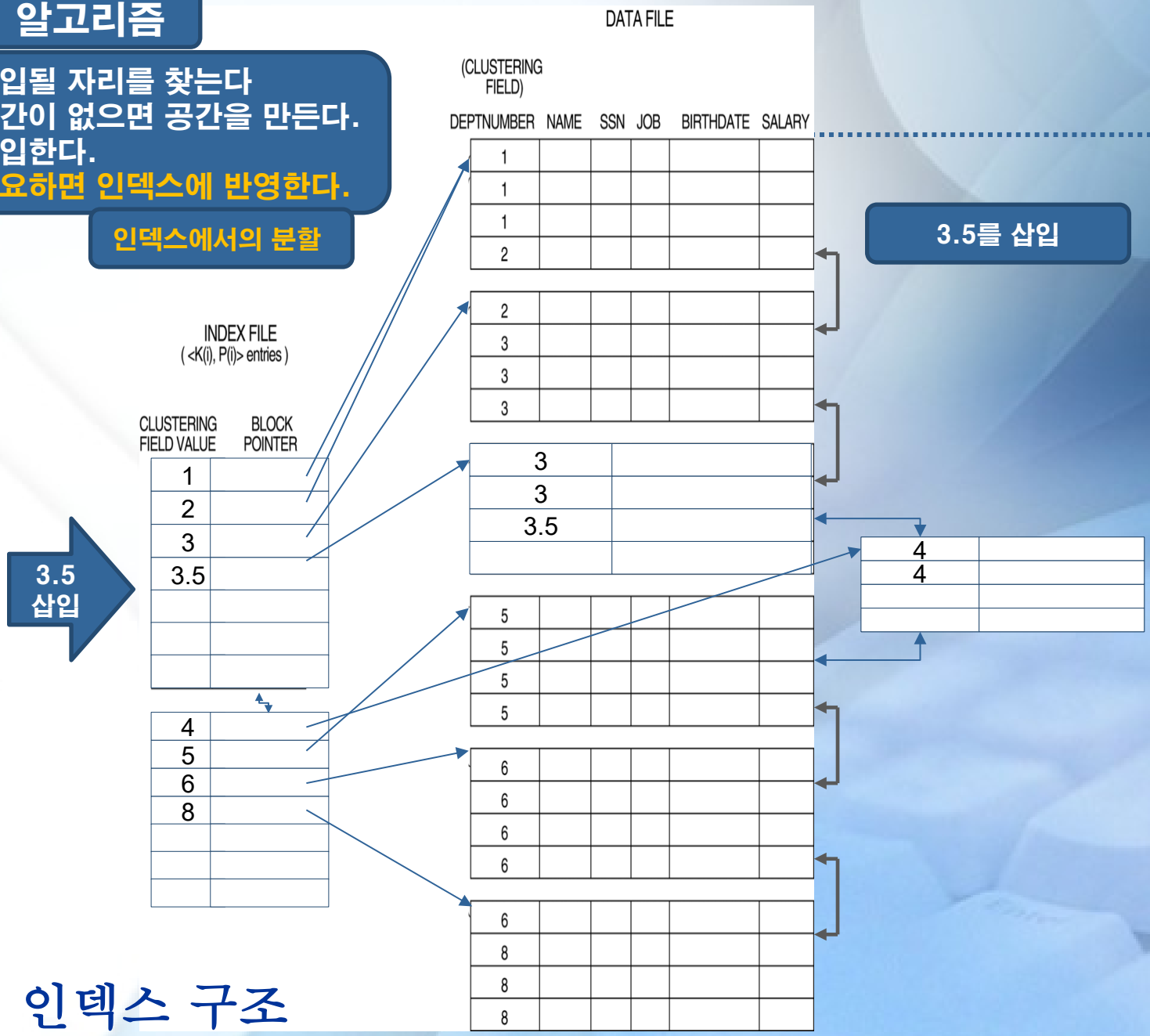


일반적인 인덱스 구조

삽입 알고리즘

1. 삽입될 자리를 찾는다
2. 공간이 없으면 공간을 만든다.
3. 삽입한다.
4. 필요하면 인덱스에 반영한다.

인덱스에서의 분할

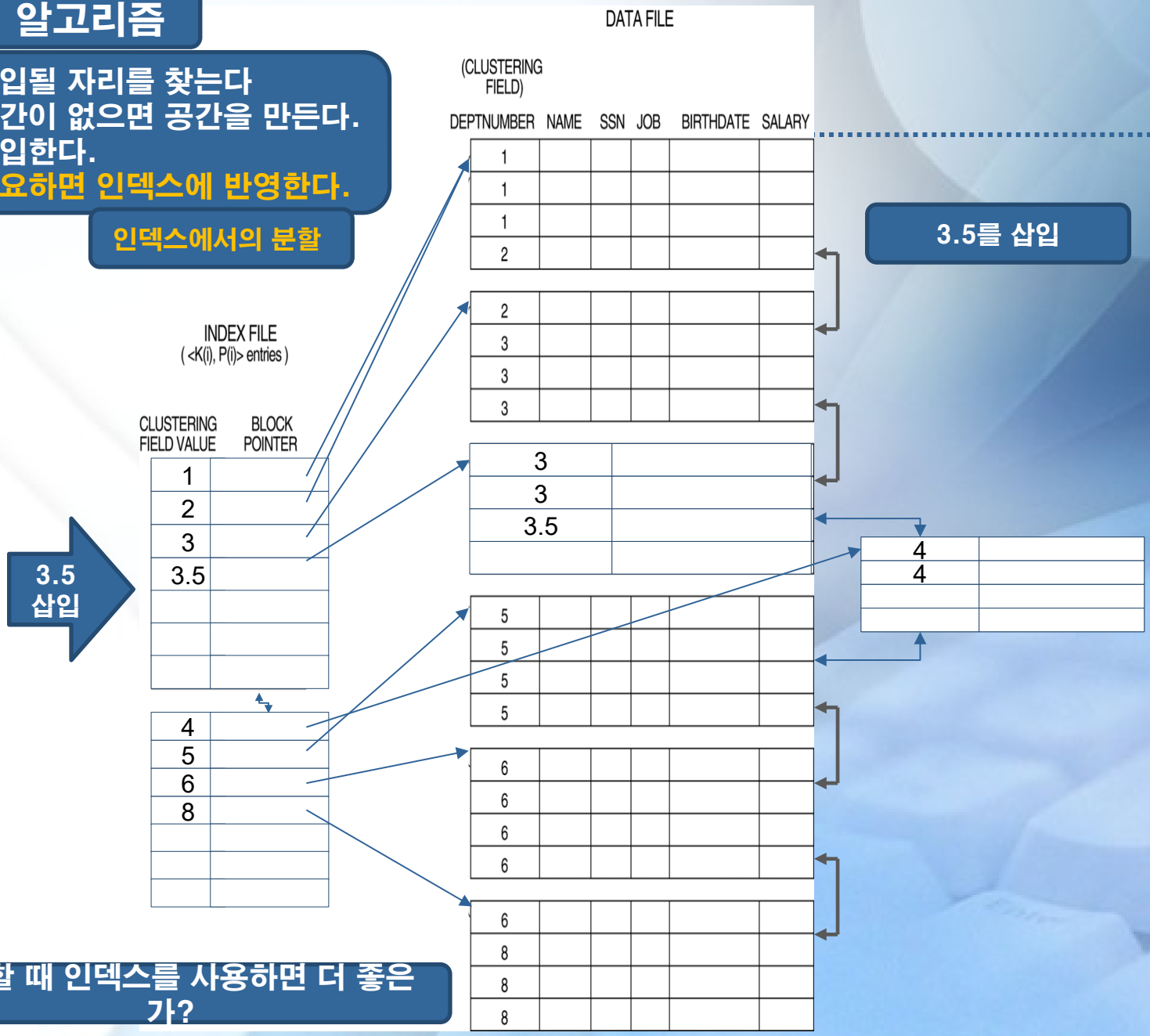


일반적인 인덱스 구조

삽입 알고리즘

1. 삽입될 자리를 찾는다
2. 공간이 없으면 공간을 만든다.
3. 삽입한다.
4. 필요하면 인덱스에 반영한다.

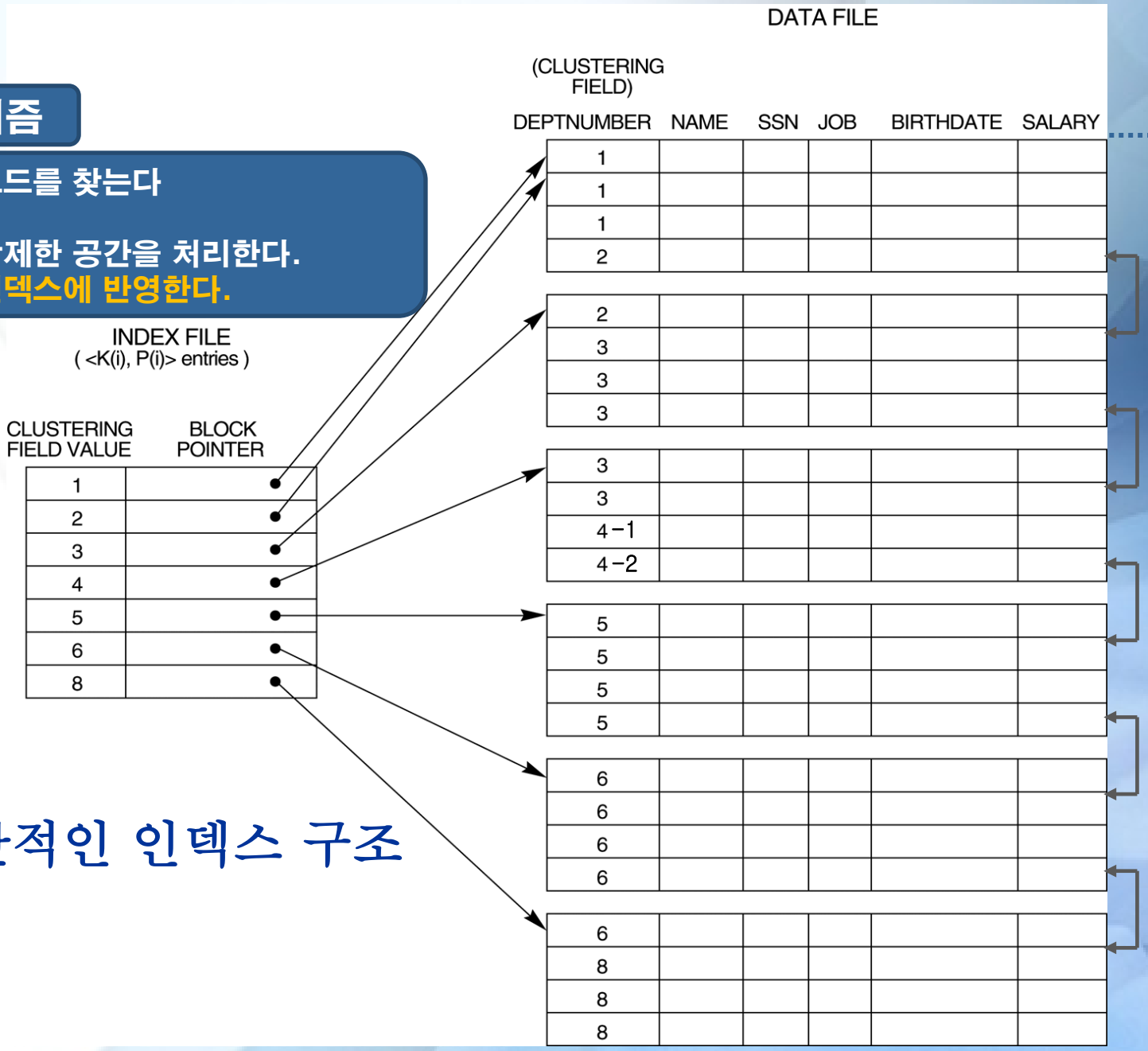
인덱스에서의 분할



삽입할 때 인덱스를 사용하면 더 좋은가?

삭제 알고리즘

1. 삭제할 레코드를 찾는다
2. 삭제한다.
3. 필요하면 삭제한 공간을 처리한다.
4. 필요하면 인덱스에 반영한다.

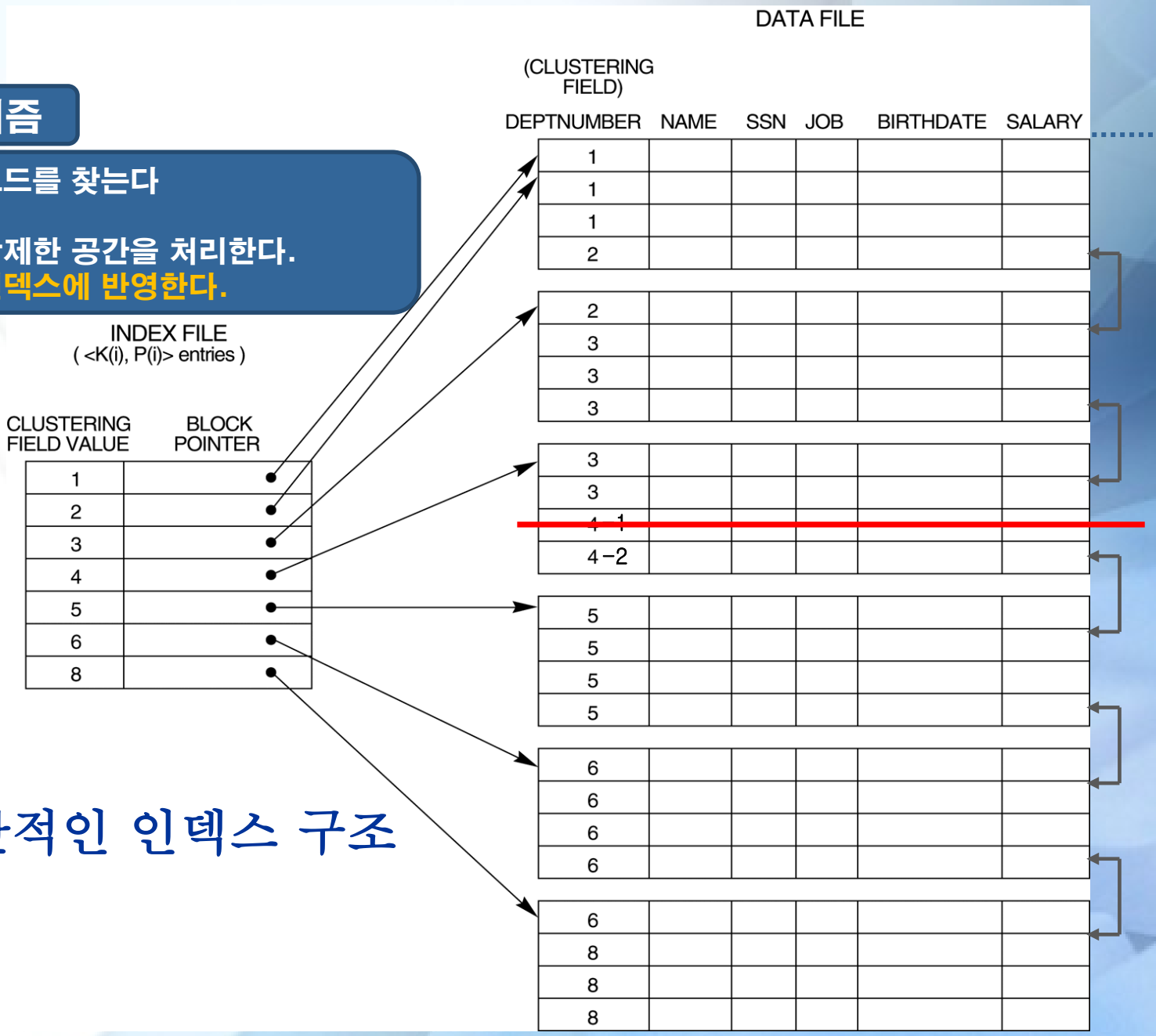


일반적인 인덱스 구조

삭제 알고리즘

1. 삭제할 레코드를 찾는다
2. 삭제한다.
3. 필요하면 삭제한 공간을 처리한다.
4. 필요하면 인덱스에 반영한다.

Delete 4-1



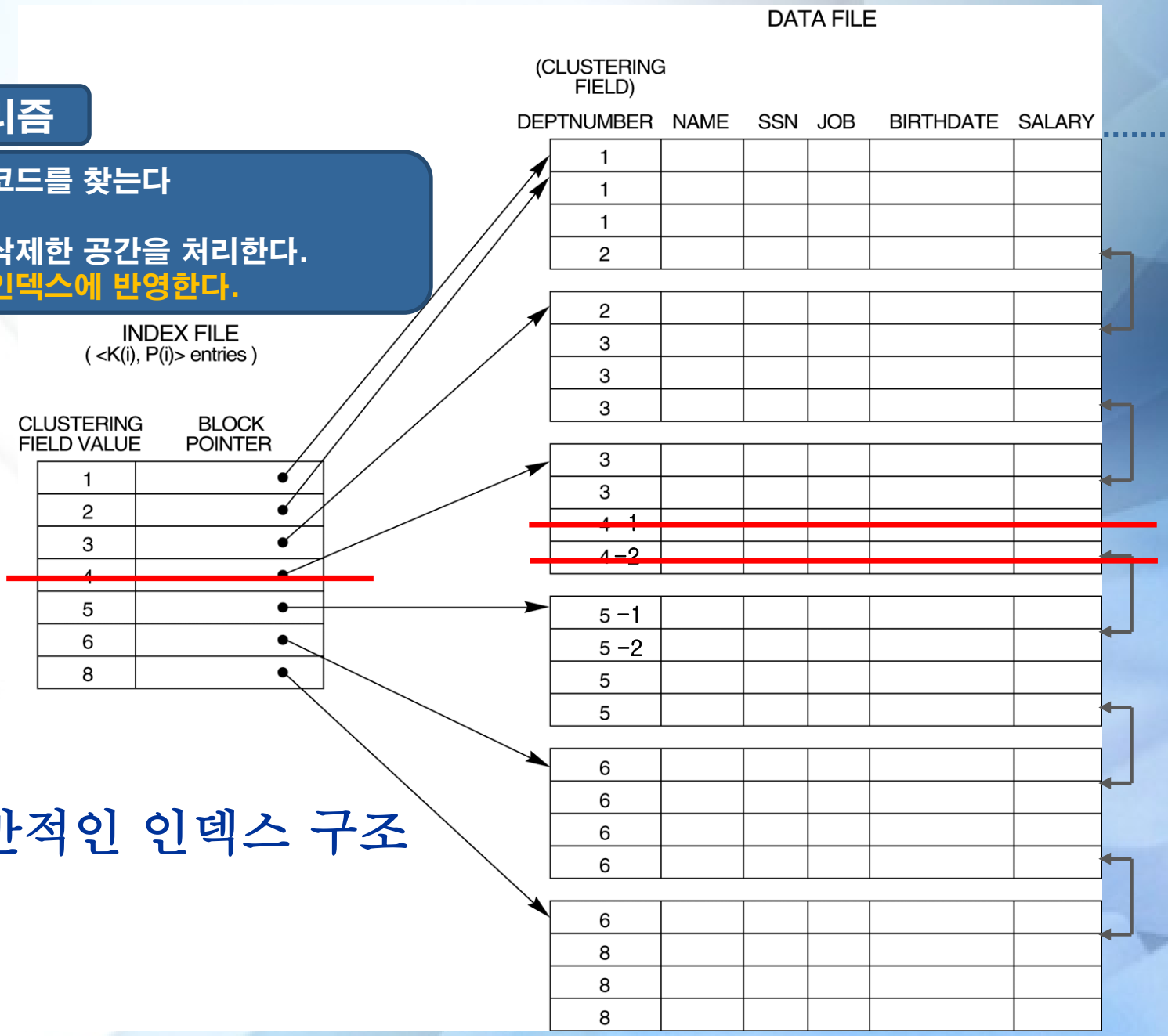
일반적인 인덱스 구조

삭제 알고리즘

1. 삭제할 레코드를 찾는다
2. 삭제한다.
3. 필요하면 삭제한 공간을 처리한다.
4. 필요하면 인덱스에 반영한다.

Delete 4-1

Delete 4-2



일반적인 인덱스 구조

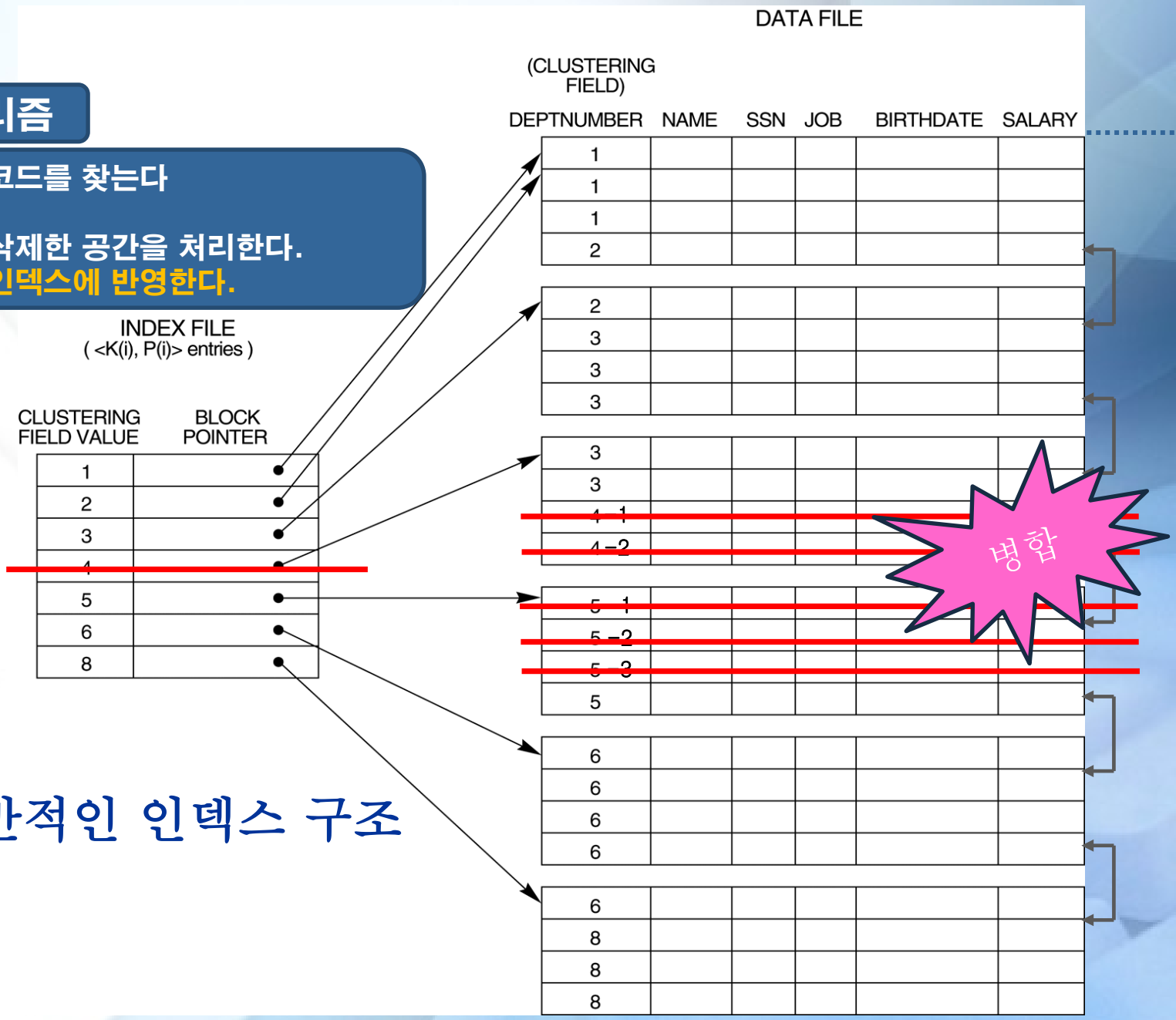
삭제 알고리즘

1. 삭제할 레코드를 찾는다
2. 삭제한다.
3. 필요하면 삭제한 공간을 처리한다.
4. 필요하면 인덱스에 반영한다.

Delete 4-1

Delete 4-2

Delete 5-1
Delete 5-2
Delete 5-3



일반적인 인덱스 구조

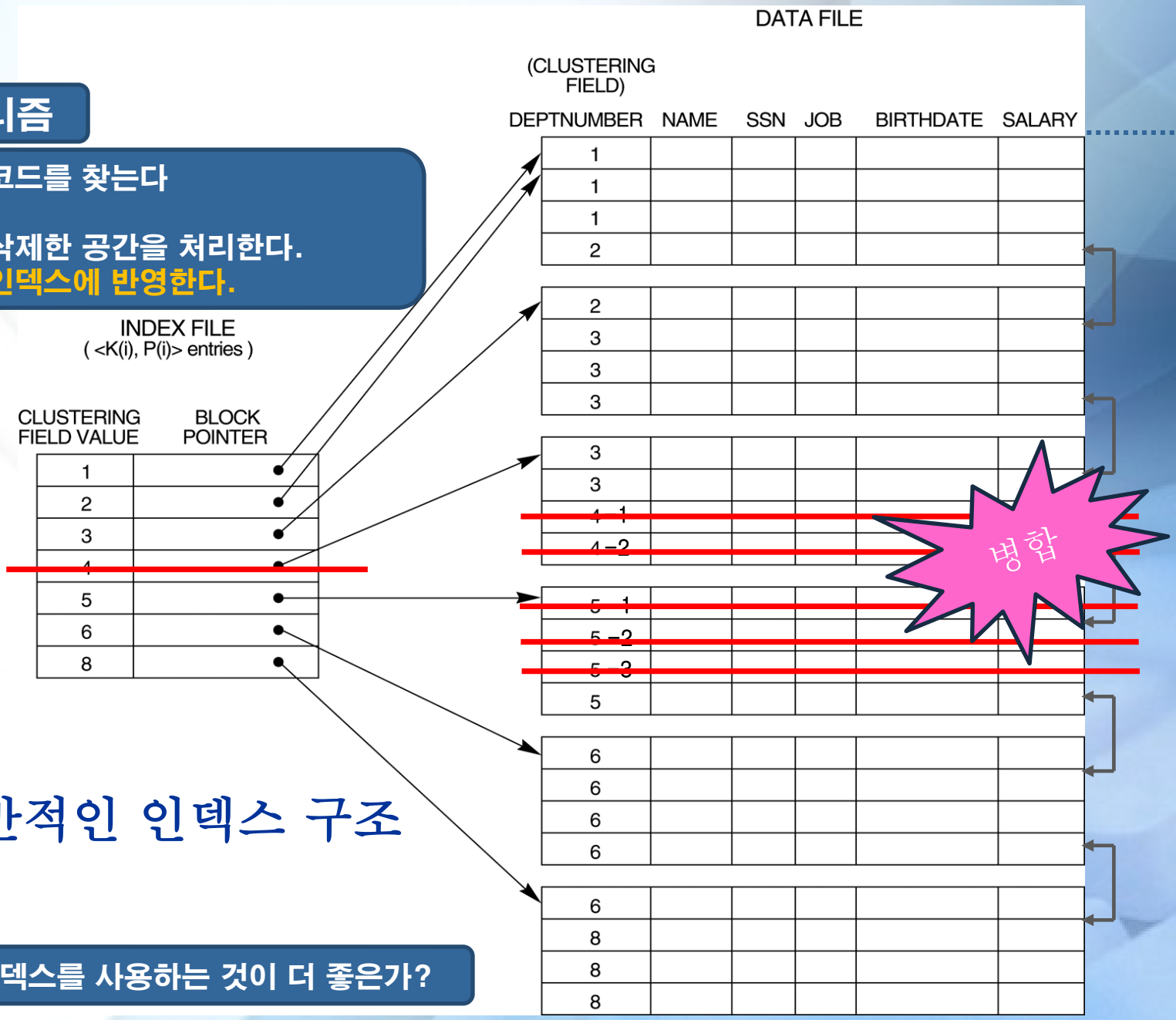
삭제 알고리즘

1. 삭제할 레코드를 찾는다
2. 삭제한다.
3. 필요하면 삭제한 공간을 처리한다.
4. 필요하면 인덱스에 반영한다.

Delete 4-1

Delete 4-2

Delete 5-1
Delete 5-2
Delete 5-3



일반적인 인덱스 구조

삭제할 때 인덱스를 사용하는 것이 더 좋은가?

생각해 봅시다.

1. 인덱스 화일의 크기는 어느 정도 일까?
2. 인덱스 엔트리의 개수는 몇 개인가?
3. 하나의 화일에 몇 개의 인덱스를 만들수 있을까?