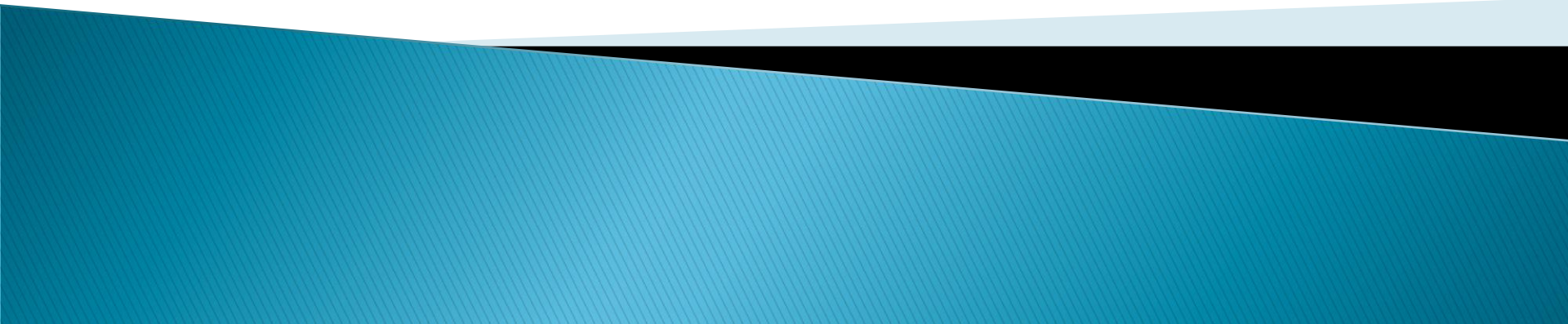


Index Architecture



시스템 구조

논리적인 데이터베이스

[illegible]

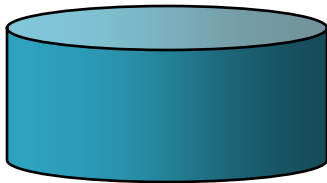
Table Customer

[illegible]

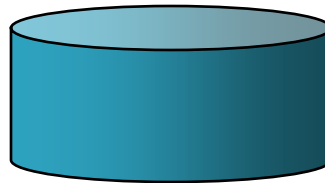
Table Orders

Table Products

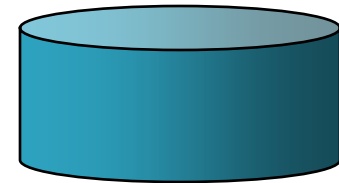
물리적인 데이터베이스



~~USE DBF~~



Example.DBF



Northwnd.LOG

데이터 파일 구조

▶ BLOCK

- SQL에서 데이터를 저장하고 처리하는데 사용되는 가장 기본적인 입출력단위

▶ 익스텐트(Extent)

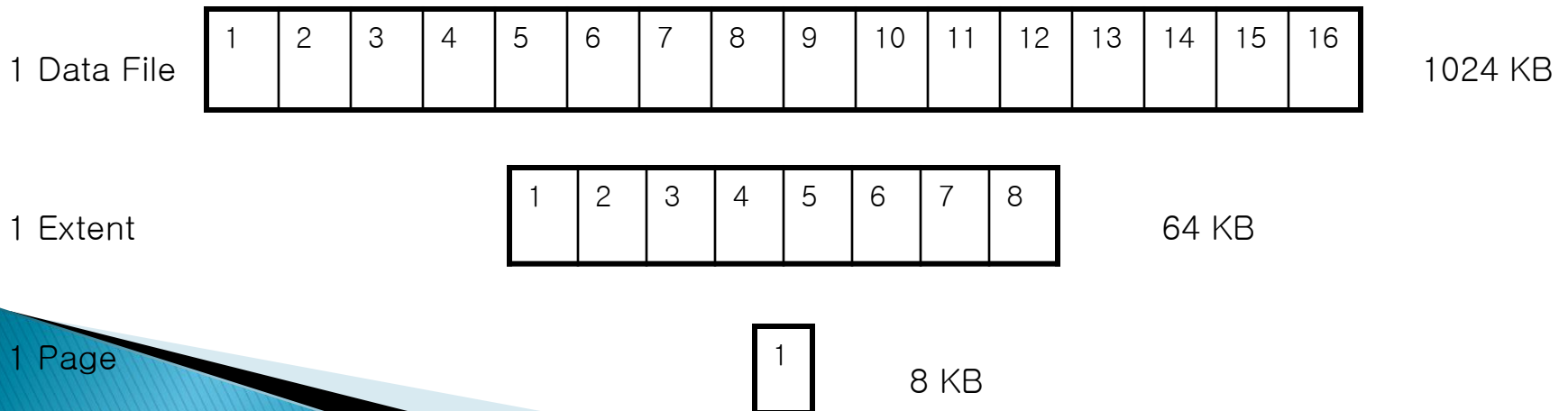
- SQL에서 테이블과 인덱스를 저장하고 관리하는데 사용되는 공간의 기본 할당 단위

BLOCK

- ▶ 총크기 : 각 BLOCK당 8KB
 - 8192Byte 의 저장공간(1024Byte X 8)
- ▶ 헤더 저장공간 : 96Byte
 - BLOCK 종류,여유공간정보,Object ID 저장
- ▶ 실 저장공간 : 8096Byte
 - 1Mbyte = 1024 Kbyte
 - > 128개의 BLOCK

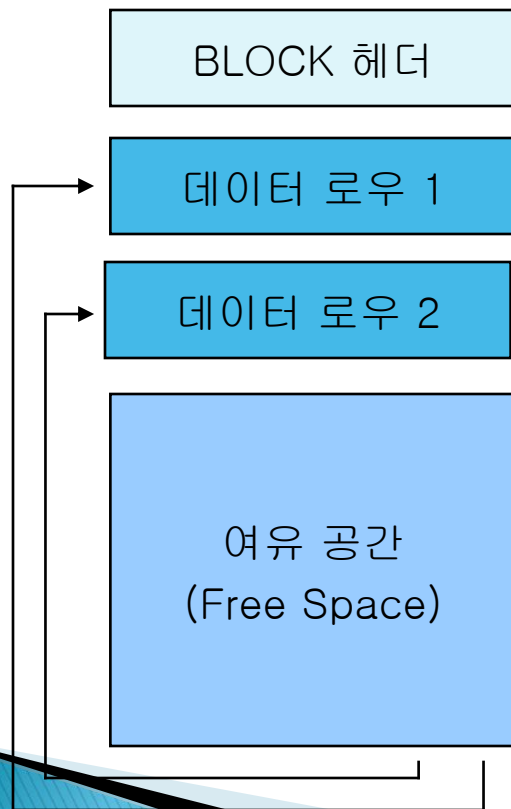
익스텐트(Extent)

- ▶ 연속된 8개의 block으로 구성(64KB)
 - 각 BLOCK이 8 Kbyte * 8개 = 64Kb
- ▶ 각 메가당 16개의 Extent
 - $1024\text{Kb} / 64\text{KB} = 16\text{개}$



BLOCK 파일 구조

▶ 데이터 BLOCK의 구조



- 별도의 BLOCK에 저장되는 text, ntext, image형의 데이터 형태를 제외한 모든 데이터 로우를 저장
(text, ntext, image형은 데이터의 크기가 너무 크기 때문에 데이터 페이지의 부담이 큼)

- ▶ BLOCK 헤더 영역에는 각 BLOCK 의 전후관계에 대한 포인터를 가지고 있다.
 - Linked List 형태



데이터 BLOCK 의 동작 원리

▶ 클러스터드 테이블

◦ Clustered Table

- 인덱스 순서 = 물리적인 데이터 BLOCK 순서
- 테이블에 하나만 적용가능

▶ 언클러스터드 테이블

◦ Non Clustered Table (힙:Heap)

- 인덱스 순서 <> 물리적인 데이터 BLOCK 순서
- 데이터가 삽입되는 무작위 상태
- 테이블에 여러 개 적용가능(최고249개)

Index

- ▶ Index Search Methodology
- ▶ Table Scan Methodology
 - Table = 도서관
 - Extent = 서고
 - Data Block = 서적
 - Index = 색인 철

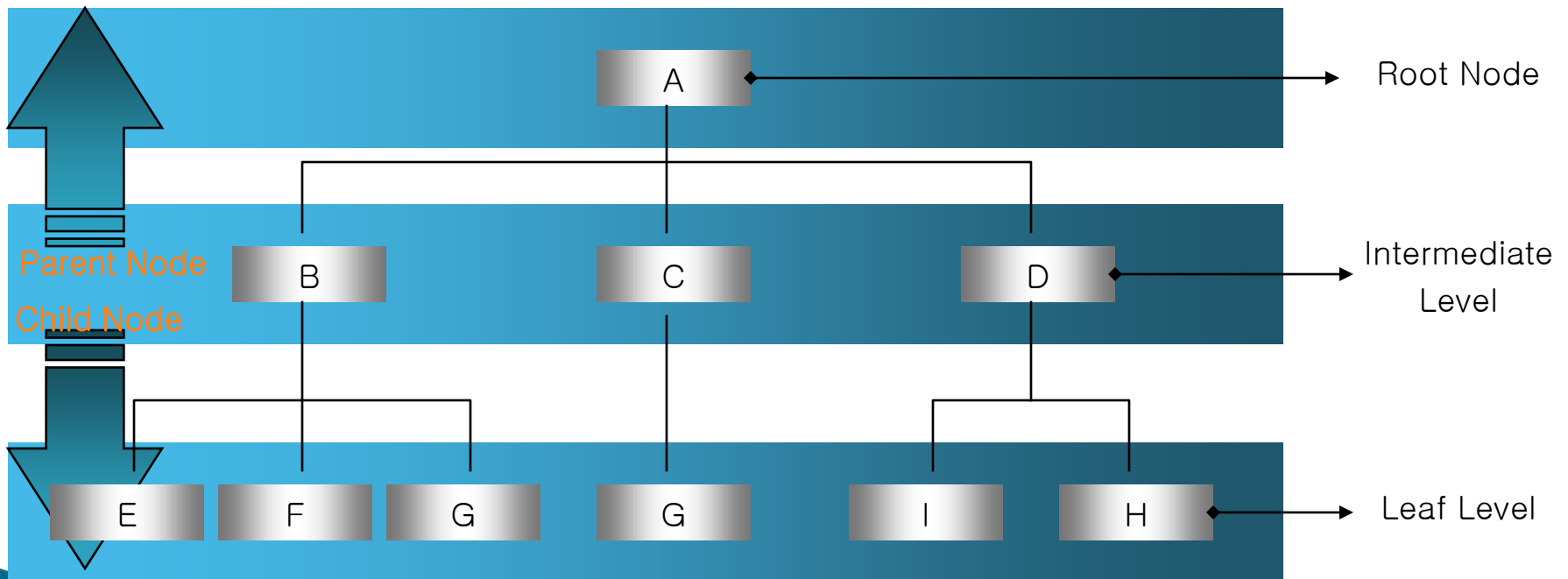
Index로 인한 문제점

- ▶ 추가시간
 - 검색시간 < 추가,수정,삭제시간
- ▶ 별도 저장공간
 - 데이터와는 별도의 장소에 저장
 - 데이터 저장 크기의 5~20%
- ▶ 관리자의 수고
 - 인덱스 재설정(Rebuild), 단편화

Index의 자료 구조

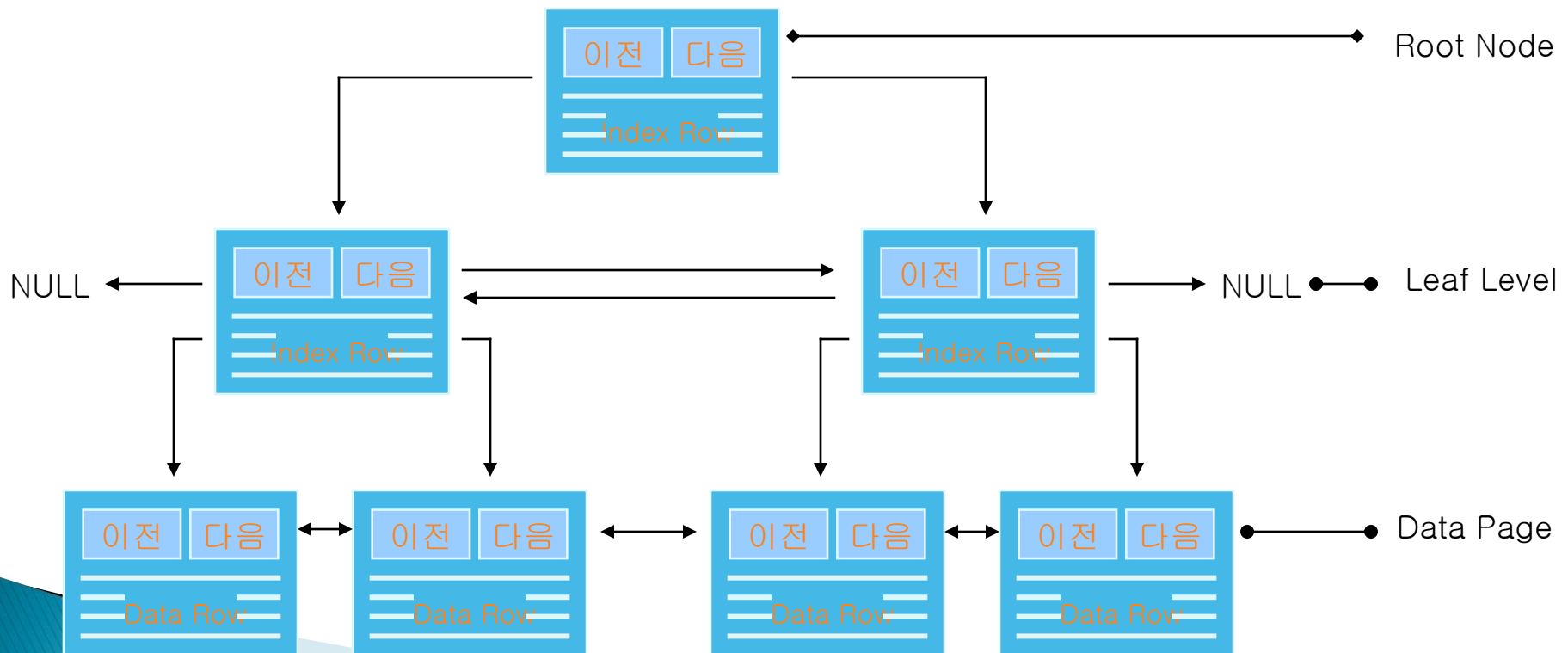
▶ B-Tree(Balanced Tree)

- 나무를 꺼꾸로 뒤집어 놓은 형태



Non-Clustered Index

- ▶ B-Tree를 이용한 인덱스 BLOCK 에 저장
 - 물리적인 데이터 저장 순서가 무작위로 들어감



Data 페이지들은 인덱스의 정렬 순서와 상관없이 저장

번호	이름	주소	나이
1	김창호	서울	40
2	김기배	인천	25
5	이은주	인천	18
7	김은혜	서울	22
15	안우길	청주	27
4	손호성	부산	20
12	최배근	서울	28
3	홍순성	서울	30
9	정순복	광주	28
22	김현철	제주	24
18	반상기	경주	35

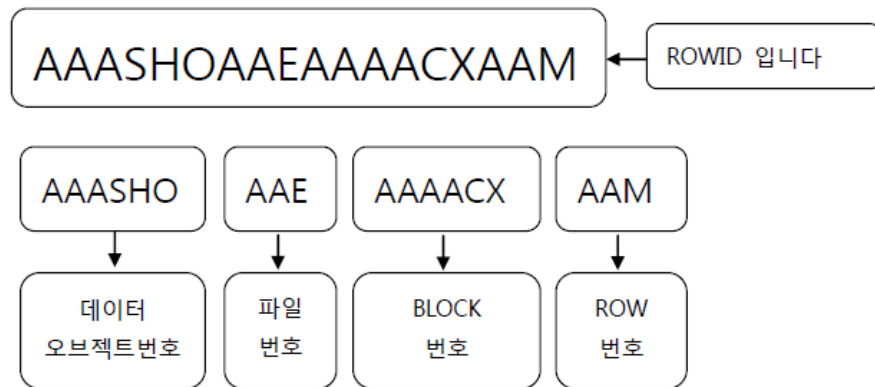
번호	이름	주소	나이
1	김창호	서울	40
2	김기배	인천	25
5	이은주	인천	18
7	김은혜	서울	22

번호	이름	주소	나이
15	안우길	청주	27
4	손호성	부산	20
12	최배근	서울	28
3	홍순성	서울	30

번호	이름	주소	나이
9	정순복	광주	28
22	김현철	제주	24
18	반상기	경주	35

Row ID 생성

- ▶ 데이터 파일
- ▶ 데이터 BLOCK 번호
- ▶ 로우 번호



번호	이름
1	1-1-1
2	1-1-2
5	1-1-3
7	1-1-4
15	1-2-1
4	1-2-2
12	1-2-3
3	1-2-4
9	1-3-1
22	1-3-2
18	1-3-3

- ▶ 인덱스 BLOCK이 데이터 BLOCK 보다 많은 양의 레코드들을 저장
 - 인덱스들은 인덱스된 해당 컬럼과 약간의 오버헤드로 구성

BLOCK 7	
1	4
5	5
15	6

Index BLOCK

BLOCK 4	
1	1-1-1
2	1-1-2
3	1-2-4
4	1-2-2

BLOCK 5	
5	1-1-3
7	1-1-4
9	1-3-1
12	1-2-3

BLOCK 6	
15	1-2-1
18	1-3-3
22	1-3-2

Data BLOCK

데이터 BLOCK 1			
1	김창호	서울	40
2	김기배	인천	25
5	이은주	인천	18
7	김은혜	서울	22

데이터 BLOCK 2			
15	안우길	청주	27
4	손호성	부산	20
12	최배근	서울	28
3	홍순성	서울	30

데이터 BLOCK 3			
9	정순복	광주	28
22	김현철	제주	24
18	반상기	경주	35

- 1. 색인 컬럼의 키값을 정렬 (Non-Clustered : Index컬럼만 정렬)
- 2. 정렬된 인덱스 컬럼을 기준으로 데이터 BLOCK을 만듬 (Row ID 기록)
- 3. Leaf level이 완성되면 Non-Leaf Level을 만듬 (Row ID 기록안함)

키 값 정렬

번호	이름
1	1-1-1
2	1-1-2
5	1-1-3
7	1-1-4
15	1-2-1
4	1-2-2
12	1-2-3
3	1-2-4
9	1-3-1
22	1-3-2
18	1-3-3

Non-Leaf Level

인덱스BLOCK 7	
1	4
5	5
15	6

Leaf Level

인덱스BLOCK 4		인덱스BLOCK 5		인덱스BLOCK 6	
1	1-1-1	5	1-1-3	15	1-2-1
2	1-1-2	7	1-1-4	18	1-3-3
3	1-2-4	9	1-3-1	22	1-3-2
4	1-2-2	12	1-2-3		

데이터 BLOCK 1			
1	김창호	서울	40
2	김기배	인천	25
5	이은주	인천	18
7	김은혜	서울	25

데이터 BLOCK 2			
15	안우길	청주	27
4	손호성	부산	20
12	최배근	서울	28
3	홍순성	서울	30

데이터 BLOCK 3			
9	정순복	광주	28
22	김현철	제주	24
18	반상기	경주	35

데이터 검색 방법

- ▶ 단일 데이터 검색(Point Query)
 - `Select * from where 이름='김현철'`
- ▶ 범위 데이터 검색(Range Query)
 - `Select * from where 지역='서울'`
 - `Select * from
where 번호 between 1 and 15`

단일 데이터 검색(Point Query)

- ▶ Select 번호 from 예제 where 번호=9

BLOCK 4	
1	1-1-1
2	1-1-2
3	1-2-4
4	1-2-2

BLOCK 7	
1	4
5	5
15	6

BLOCK 5	
5	1-1-3
7	1-1-4
9	1-3-1
12	1-2-3

BLOCK 6	
15	1-2-1
18	1-3-3
22	1-3-2

데이터 BLOCK 1			
1	김창호	서울	40
2	김기배	인천	25
5	이은주	인천	18
7	김은혜	서울	25

데이터 BLOCK 2			
15	안우길	청주	27
4	손호성	부산	20
12	최배근	서울	28
3	홍순성	서울	30

데이터 BLOCK 3			
9	정순복	광주	28
22	김현철	제주	24
18	반상기	경주	35

새로운 데이터 추가

BLOCK 5	
5	1-1-3
7	1-1-4
9	1-3-1
12	1-2-3

BLOCK 7	
1	4
5	5
12	8
15	6

Index Page

② Page 분할

BLOCK 4	
1	1-1-1
2	1-1-2
3	1-2-4
4	1-2-2

BLOCK 5	
5	1-1-3
6	1-3-4

BLOCK 8	
7	1-1-4
9	1-3-1
12	1-2-3

BLOCK 6	
15	1-2-1
18	1-3-3
22	1-3-2

Data Page

데이터 BLOCK 1			
1	김창호	서울	40
2	김기배	인천	25
5	이은주	인천	18
7	김은혜	서울	22

데이터 BLOCK 2			
15	안우길	청주	27
4	손호성	부산	20
12	최배근	서울	28
3	홍순성	서울	30

데이터 BLOCK 3			
9	정순복	광주	28
22	김현철	제주	24
18	반상기	경주	35
6	김상경	진주	24

New Data

Clustered Index

Index Page

인덱스 BLOCK 4	
1	1
5	2
15	3

Data Page

데이터 BLOCK 1			
1	김창호	서울	40
2	김기배	인천	25
3	홍순성	서울	30
4	손호성	부산	20

데이터 BLOCK 2			
5	이은주	인천	25
7	김은혜	서울	22
9	정순복	광주	28
12	최배근	서울	28

데이터 BLOCK 3			
15	안우길	청주	27
18	반상기	경주	35
22	김현철	제주	24

새로운 데이터 추가

데이터 BLOCK 1			
1	김창호	서울	40
2	김기배	인천	25
3	홍순성	서울	30
4	손호성	부산	20

데이터 BLOCK 5			
12	최배근	서울	28

데이터 BLOCK 2			
5	이은주	인천	25
7	김은혜	서울	22
9	정순복	광주	28
10	남광훈	서울	29

데이터 BLOCK 3			
15	안우길	청주	27
18	반상기	경주	35
22	김현남	제주	24

Page 분할

인덱스 BLOCK 4	
1	1
5	2
15	3
12	5

New Data

