1. index merge scan 이란 ?

여러개의 인덱스를 같이 사용하여 하나의 인덱스만 사용했을때 보다 테이블 엑세스를 줄일 수 있는 인덱스 스캔방법

	인덱스 엑세스 방법	관련 힌트
1	index range scan	index
2	index unique scan	index
3	index full scan	index
4	index skip scan	index_ss
5	index fast full scan	index_ffs
→ 6	index merge scan	and_equal
7	index bitmap merge scan	index_combine

```
select  /*+ gather_plan_statistics index(emp2 emp2_col1) */  count(*)
  from  emp2
  where  col1='A'  and col2='D';
```

emp2_col1 인덱스

COL1	ROWID
A	AAAXDOAANAAAAHjAAJ
A	AAAXDOAANAAAAHjAAL
A	AAAXDOAANAAAAHjAAH
A	AAAXDOAANAAAAHjAAK
A	AAAXDOAANAAAAHjAAM
A	AAAXDOAANAAAAHjAAN
A	**AAAXDOAANAAAAHjAAB**
A	**AAAXDOAANAAAAHjAAC**
A	AAAXDOAANAAAAHjAAD
A	AAAXDOAANAAAAHjAAA
A	AAAXDOAANAAAAHjAAE
A	AAAXDOAANAAAAHjAAF
A	AAAXDOAANAAAAHjAAG
A	AAAXDOAANAAAAHjAAI

emp2 테이블

ROWID	COL1	COL2	...
AAATc1AAHAAAAHeAAA	A	C	···
AAATc1AAHAAAAHeAAB	A	D	···
AAATc1AAHAAAAHeAAC	A	D	···
AAATc1AAHAAAAHeAAD	A	C	···
AAATc1AAHAAAAHeAAE	A	C	···
AAATc1AAHAAAAHeAAF	A	C	···
AAATc1AAHAAAAHeAAG	A	C	···
AAATc1AAHAAAAHeAAH	A	C	···
AAATc1AAHAAAAHeAAI	A	C	···
AAATc1AAHAAAAHeAAJ	A	C	···
AAATc1AAHAAAAHeAAK	A	C	···
AAATc1AAHAAAAHeAAL	A	C	···
AAATc1AAHAAAAHeAAM	A	C	···
AAATc1AAHAAAAHeAAN	A	C	···

```
select  /*+ gather_plan_statistics index(emp2 emp2_col2) */  count(*)
  from  emp2
  where  col1='A'  and col2='D';
```

emp2_col2 인덱스

COL2	ROWID
D	AAAXDOAANAAAAHjAAZ
D	AAAXDOAANAAAAHjAAW
D	AAAXDOAANAAAAHjAAX
D	AAAXDOAANAAAAHjAAY
D	AAAXDOAANAAAAHjAQ
D	AAAXDOAANAAAAHjAAR
D	**AAAXDOAANAAAAHjAAB**
D	**AAAXDOAANAAAAHjAAC**
D	AAAXDOAANAAAAHjAAS
D	AAAXDOAANAAAAHjAAT
D	AAAXDOAANAAAAHjAAU
D	AAAXDOAANAAAAHjAAV
D	AAAXDOAANAAAAHjAAO
D	AAAXDOAANAAAAHjAAP

emp2 테이블

ROWID	COL1	COL2	...
AAATc1AAHAAAAHeAAP	B	D	···
AAATc1AAHAAAAHeAAO	B	D	···
AAATc1AAHAAAAHeAAV	B	D	···
AAATc1AAHAAAAHeAAU	B	D	···
AAATc1AAHAAAAHeAAT	B	D	···
AAATc1AAHAAAAHeAAS	B	D	···
AAATc1AAHAAAAHeAAR	B	D	···
AAATc1AAHAAAAHeAAQ	B	D	···
AAATc1AAHAAAAHeAAY	B	D	···
AAATc1AAHAAAAHeAAX	B	D	···
AAATc1AAHAAAAHeAAB	A	D	···
AAATc1AAHAAAAHeAAC	A	D	···
AAATc1AAHAAAAHeAAZ	B	D	···
AAATc1AAHAAAAHeAAW	B	D	···

```
select /*+ gather_plan_statistics and_equal(emp2 emp2_col1 emp2_col2) */ count(*)
 from emp2
 where col1='A' and col2='D';
```

emp2_col1 인덱스

COL1	ROWID
A	AAAXDOAANAAAAHjAAJ
A	AAAXDOAANAAAAHjAAL
A	AAAXDOAANAAAAHjAAH
A	AAAXDOAANAAAAHjAAK
A	AAAXDOAANAAAAHjAAM
A	AAAXDOAANAAAAHjAAN
A	**AAAXDOAANAAAAHjAAB**
A	**AAAXDOAANAAAAHjAAC**
A	AAAXDOAANAAAAHjAAD
A	AAAXDOAANAAAAHjAAA
A	AAAXDOAANAAAAHjAAE
A	AAAXDOAANAAAAHjAAF
A	AAAXDOAANAAAAHjAAG
A	AAAXDOAANAAAAHjAAI

emp2_col2 인덱스

COL2	ROWID
D	AAAXDOAANAAAAHjAAZ
D	AAAXDOAANAAAAHjAAW
D	AAAXDOAANAAAAHjAAX
D	AAAXDOAANAAAAHjAAY
D	AAAXDOAANAAAAHjAQ
D	AAAXDOAANAAAAHjAAR
D	**AAAXDOAANAAAAHjAAB**
D	**AAAXDOAANAAAAHjAAC**
D	AAAXDOAANAAAAHjAAS
D	AAAXDOAANAAAAHjAAT
D	AAAXDOAANAAAAHjAAU
D	AAAXDOAANAAAAHjAAV
D	AAAXDOAANAAAAHjAAO
D	AAAXDOAANAAAAHjAAP

emp2 테이블

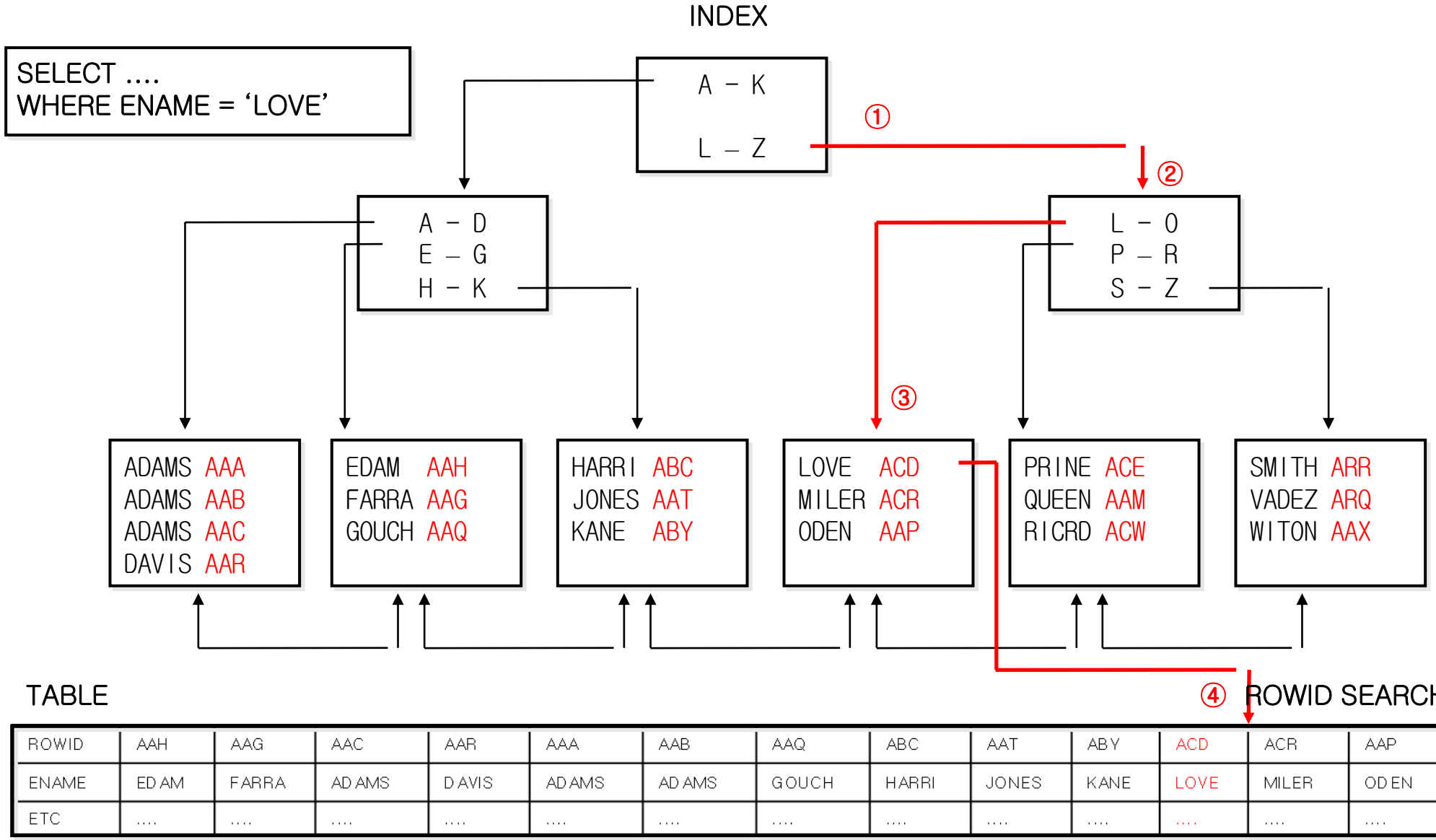
ROWID	COL1	COL2	...
AAATc1AAHAAAAHeAAA	A	C	···
AAATc1AAHAAAAHeAAB	A	D	···
AAATc1AAHAAAAHeAAC	A	D	···
AAATc1AAHAAAAHeAAD	A	C	···
AAATc1AAHAAAAHeAAE	A	C	···
AAATc1AAHAAAAHeAAF	A	C	···
AAATc1AAHAAAAHeAAG	A	C	···
AAATc1AAHAAAAHeAAH	A	C	···
AAATc1AAHAAAAHeAAI	A	C	···
AAATc1AAHAAAAHeAAJ	A	C	···
AAATc1AAHAAAAHeAAK	B	C	···
AAATc1AAHAAAAHeAAL	B	C	···
AAATc1AAHAAAAHeAAM	B	C	···
AAATc1AAHAAAAHeAAN	B	C	···

2. index bitmap merge scan 이란 ?

일반 인덱스를 크기가 아주 작은 비트맵 인덱스로 변환하고 비트맵 인덱스들을 하나로 합쳐서 스캔하는 스캔방법

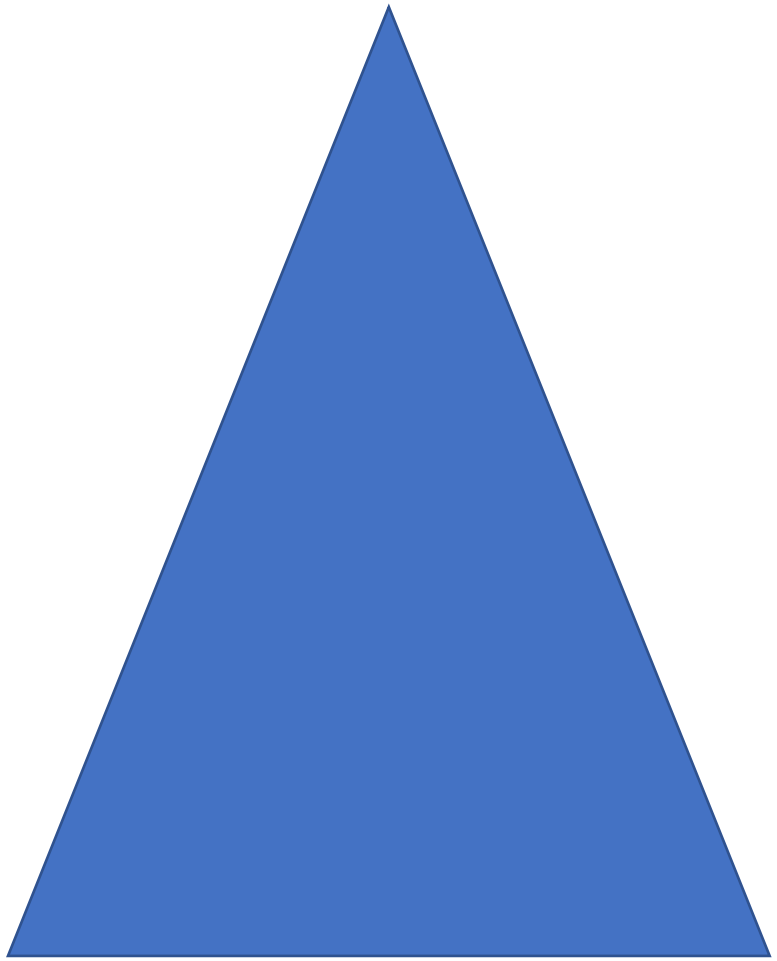
	인덱스 엑세스 방법	관련 힌트
1	index range scan	index
2	index unique scan	index
3	index full scan	index
4	index skip scan	index_ss
5	index fast full scan	index_ffs
6	index merge scan	and_equal
→ 7	index bitmap merge scan	index_combine

일반적인 tree 구조의 인덱스

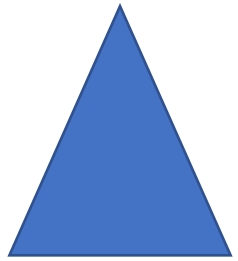


ROWID	AAH	AAG	AAC	AAR	AAA	AAB	AAQ	ABC	AAT	ABY	ACD	ACR	AAP	....
ENAME	EDAM	FARRA	ADAMS	DAVIS	ADAMS	ADAMS	GOUCH	HARRI	JONES	KANE	LOVE	MILER	ODEN	....
ETC	....	....	....	....	....	....	....	....	....	....	....	....	....	....

일반 인덱스 → 비트맵 인덱스로 변환하게 되면?



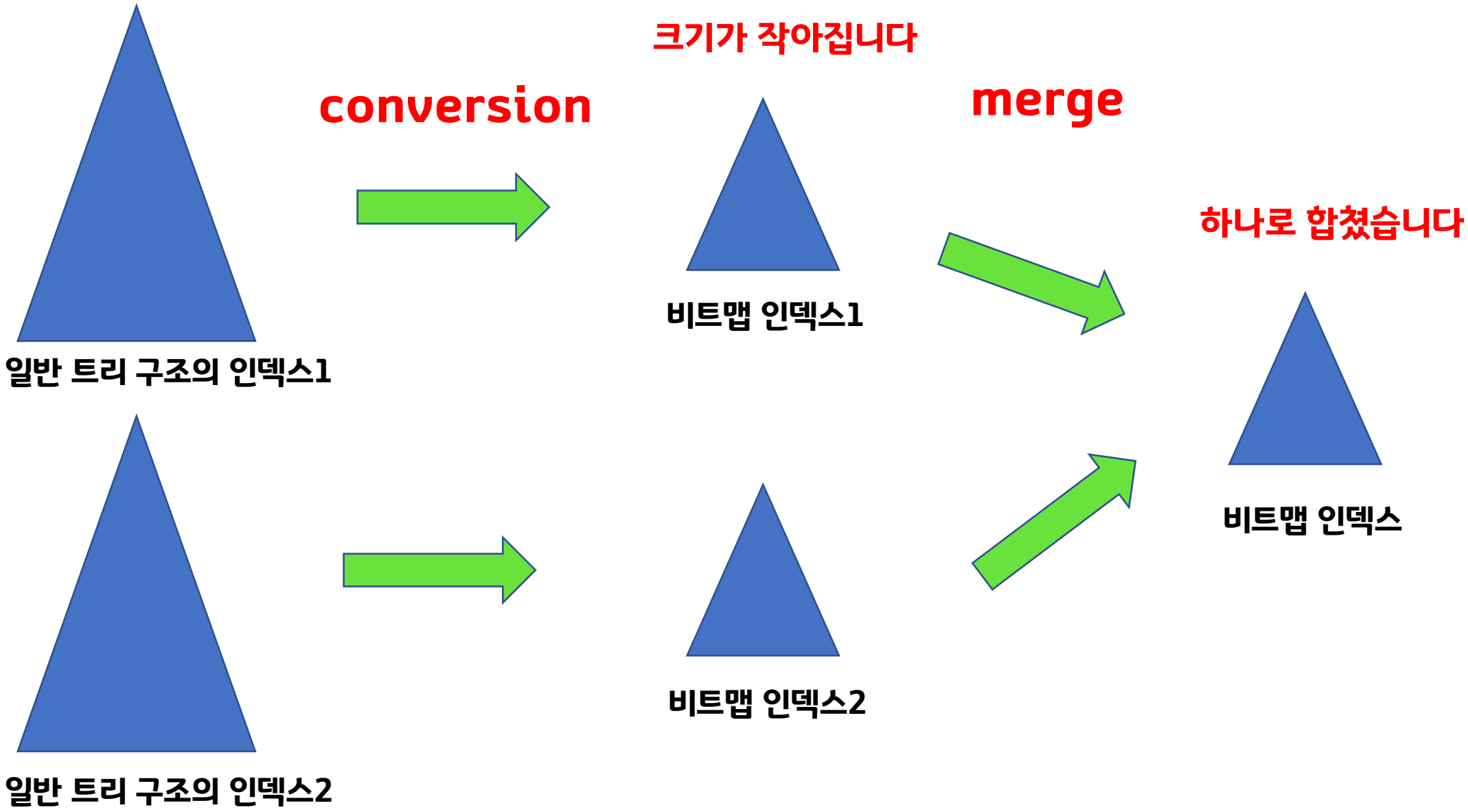
크기가 작아집니다



일반 트리 구조의 인덱스

비트맵 인덱스

비트맵으로 변환하고 합칩니다



```
select /*+ gather_plan_statistics index_combine(emp2) */ count(*)
 from emp2
 where col1='A' and col2='D';
```

bitmap conversion

bitmap conversion

emp_col1 인덱스

emp_col2 인덱스

emp2 테이블

COL1	ROWID
A	AAAXDOAANAAAAHjAAJ
A	AAAXDOAANAAAAHjAAL
A	AAAXDOAANAAAAHjAAH
A	AAAXDOAANAAAAHjAAK
A	AAAXDOAANAAAAHjAAM
A	AAAXDOAANAAAAHjAAN
A	**AAAXDOAANAAAAHjAAB**
A	**AAAXDOAANAAAAHjAAC**
A	AAAXDOAANAAAAHjAAD
A	AAAXDOAANAAAAHjAAA
A	AAAXDOAANAAAAHjAAE
A	AAAXDOAANAAAAHjAAF
A	AAAXDOAANAAAAHjAAG
A	AAAXDOAANAAAAHjAAI

COL2	ROWID
D	AAAXDOAANAAAAHjAAZ
D	AAAXDOAANAAAAHjAAW
D	AAAXDOAANAAAAHjAAX
D	AAAXDOAANAAAAHjAAY
D	AAAXDOAANAAAAHjAQ
D	AAAXDOAANAAAAHjAAR
D	**AAAXDOAANAAAAHjAAB**
D	**AAAXDOAANAAAAHjAAC**
D	AAAXDOAANAAAAHjAAS
D	AAAXDOAANAAAAHjAAT
D	AAAXDOAANAAAAHjAAU
D	AAAXDOAANAAAAHjAAV
D	AAAXDOAANAAAAHjAAO
D	AAAXDOAANAAAAHjAAP

ROWID	COL1	COL2	...
AAATc1AAHAAAAHeAAA	A	D	···
AAATc1AAHAAAAHeAAB	A	D	···
AAATc1AAHAAAAHeAAC	A	D	···
AAATc1AAHAAAAHeAAD	A	D	···
AAATc1AAHAAAAHeAAE	A	D	···
AAATc1AAHAAAAHeAAF	A	D	···
AAATc1AAHAAAAHeAAG	A	D	···
AAATc1AAHAAAAHeAAH	A	D	···
AAATc1AAHAAAAHeAAI	A	D	···
AAATc1AAHAAAAHeAAJ	A	D	···
AAATc1AAHAAAAHeAAK	A	D	···
AAATc1AAHAAAAHeAAL	A	D	···
AAATc1AAHAAAAHeAAM	A	D	···
AAATc1AAHAAAAHeAAN	A	D	···