

11. 인덱스의 구조와 유형



(1) 인덱스 및 기본 액세스 방식

ROWID

- ROWID는 행의 주소를 나타냅니다.
- 테이블마다 ROWID 의사(pseudo) 열이 있습니다.

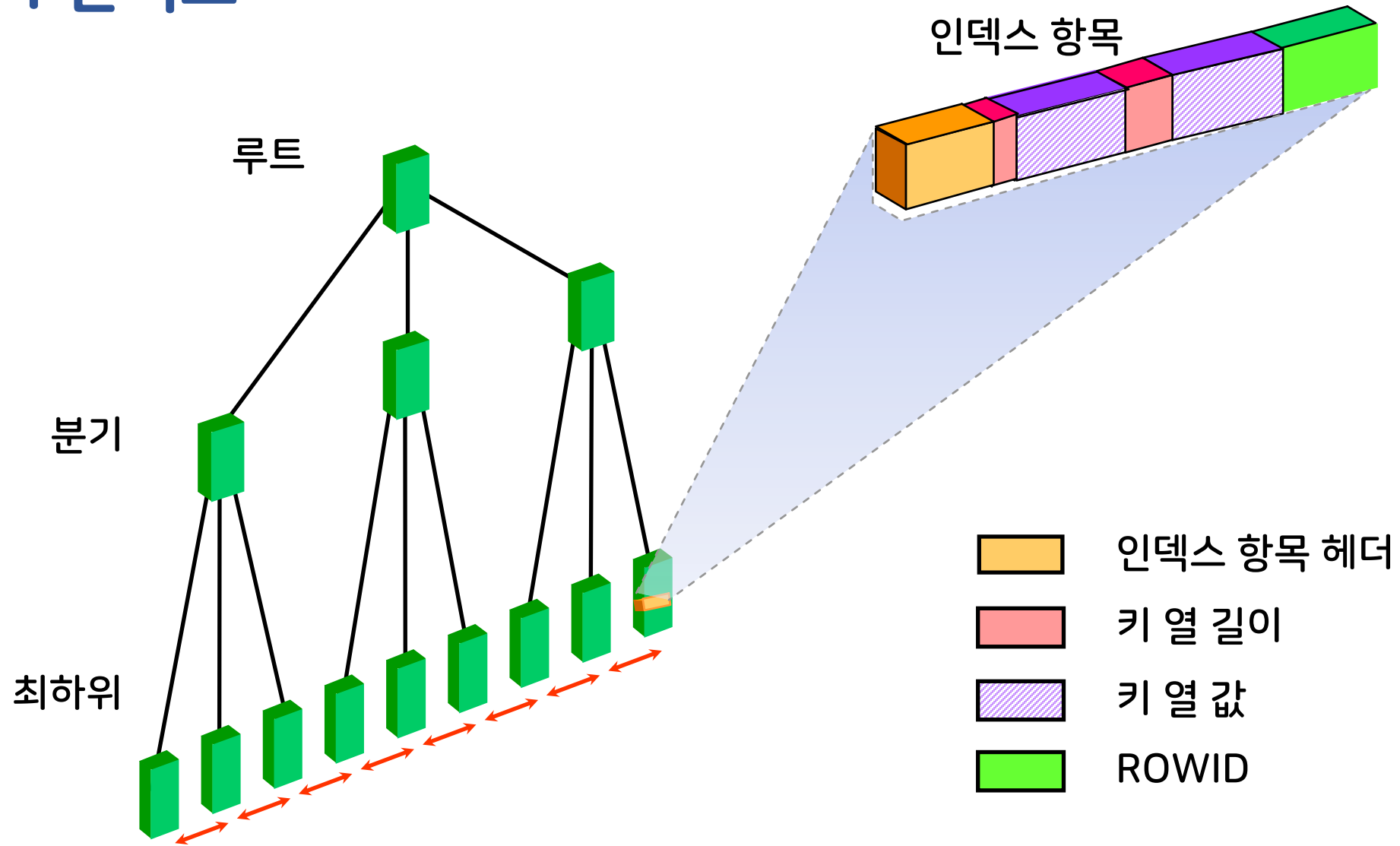
```
select rowid, c.* from channels c;
```

ROWID	C	CHANNEL_DESC	CHANNEL_CLASS
-----	-	-----	-----
AAABQDAABAAAJ4yAAA	S	Direct Sales	Direct
AAABQDAABAAAJ4yAAB	T	Tele Sales	Direct
AAABQDAABAAAJ4yAAC	C	Catalog	Indirect
AAABQDAABAAAJ4yAAD	I	Internet	Indirect
AAABQDAABAAAJ4yAAE	P	Partners	Others

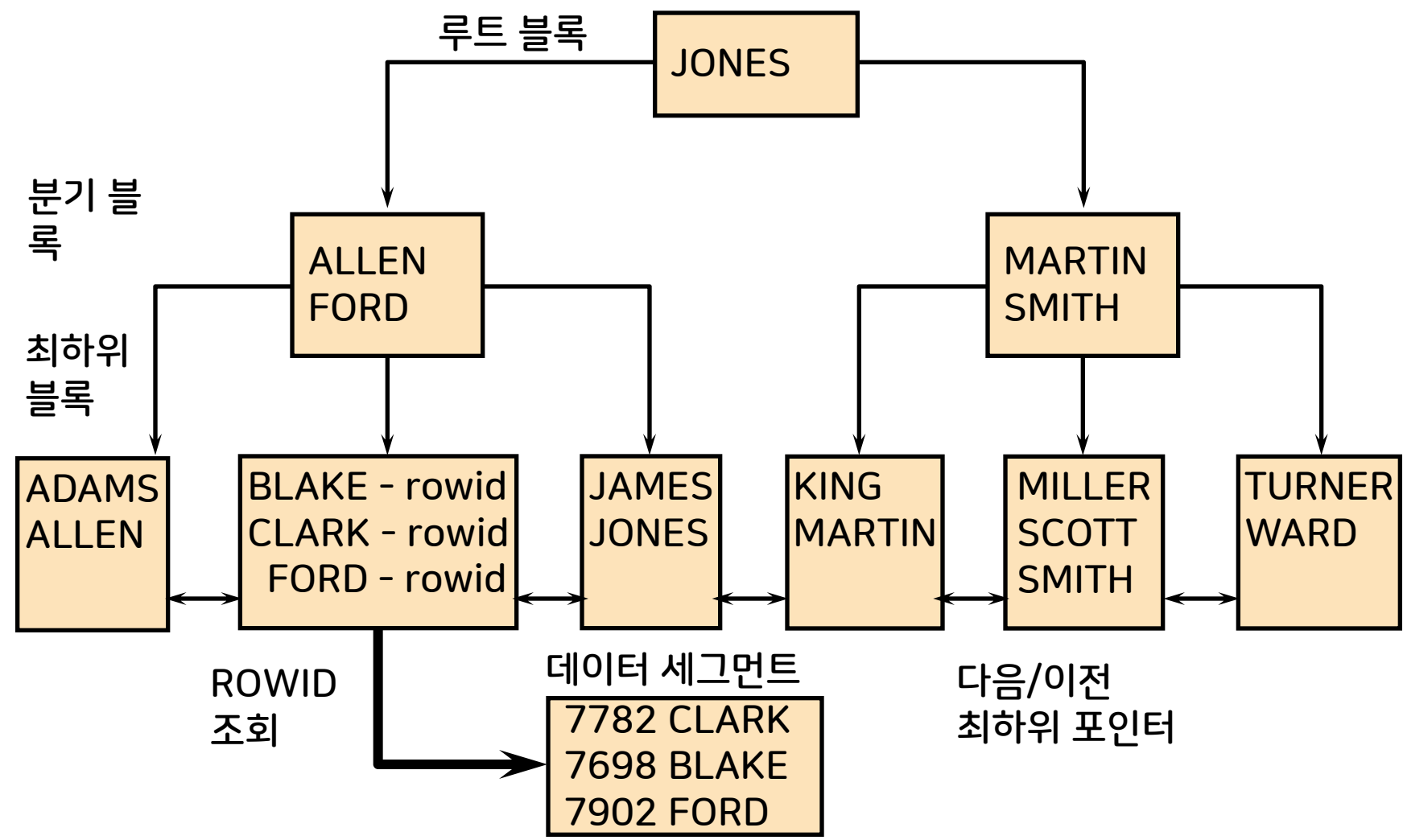
인덱스

- 고유 인덱스 및 고유하지 않은 인덱스
- 조합 인덱스 : 한 테이블에 있는 여러 열에 대해 생성하는 인덱스입니다
- 인덱스 저장 기법
 - B* 트리 : Oracle에서는 표준 인덱스의 경우 액세스 시간을 균등화하는 균형 잡힌 B* 트리 인덱스를 사용합니다.
 - 역방향 키
 - 내림차순
 - 함수 기반
 - 비트맵

B* 트리 인덱스



B* 트리 인덱스 예제



인덱스 및 제약 조건

- Oracle 서버는 다음 사항이 정의될 때 B* 트리 인덱스를 자동으로 생성하거나 사용합니다.
 - 기본 키 제약 조건
 - 고유 제약 조건
- 오라클 데이터베이스는 외래 키 제약 조건에 대해 인덱스를 자동으로 생성하지 않습니다.
- 외래 키 열이 조인 조건에서 자주 사용되면 이 열에 대해 인덱스를 생성하여 조인 프로세스를 개선해야 합니다.

```
CREATE TABLE channels
(
    channel_id CHAR(1)
        CONSTRAINT chan_pk PRIMARY KEY
    , channel_desc VARCHAR2(20)
        CONSTRAINT chan_desc_nn NOT NULL
    , channel_class VARCHAR2(20)
    , channel_total VARCHAR2(13)
);
```

조합 인덱스 고려사항

■ 조합인덱스 지침

- 가장 자주 질의되는 열을 맨 앞(인덱스의 선행 부분(leading part)이라고도 함)에 둡니다.
- 전체 키를 지정하려면 가장 제한을 많이 받는 열을 맨 앞에 둡니다.
- 기본 테이블에 액세스하지 않고 질의 결과를 검색할 수 있도록 여분의 열을 인덱스에 추가합니다.
- 조합 인덱스를 생성할 경우 저장 영역을 줄이기 위해 COMPRESS 옵션을 사용하여 반복되는 키 열 값을 제거할 수 있습니다.

■ 인덱스 스킵 스캐닝

- 스킵 스캐닝은 조합 인덱스에서 첫 번째 열의 조건이 없을 때, 옵티마이저가 첫 열의 모든 가능한 값을 반복하며 뒤쪽 열의 조건을 사용해 인덱스를 탐색하는 방식입니다.
- 주로 첫 번째 열의 카디널리티가 낮을 때 효율적으로 사용됩니다.
- 역방향 키 또는 비트맵 인덱스에서는 지원되지 않습니다.

기본 액세스 방식

- 전체 테이블 스캔(Full Table Scan):

- 테이블의 모든 블록을 순차적으로 읽으며 조건에 맞는 행을 찾는 방식으로, 인덱스를 사용할 수 없거나 조건이 거의 모든 행에 해당될 때 효율적입니다.

- 인덱스 스캔(Index Scan):

- 인덱스를 조건에 따라 순차적으로 탐색하여 해당 행의 위치(RID 또는 ROWID)를 찾아 접근하는 방식으로, 범위 조건이나 정렬 시 효율적입니다.

- 빠른 전체 인덱스 스캔(Fast Full Index Scan):

- 인덱스를 리프 블록 기준으로 전체 읽되, 순서를 고려하지 않고 병렬 또는 효율적으로 읽어들이는 방식으로, 테이블 풀스캔을 대체할 수 있으며 정렬이 필요 없는 경우 사용됩니다.

참고) Full Table Scan과 Fast Full Index Scan은 다중 블록 I/O(Multi-block I/O) 및 병렬 쿼리(Parallel Query)가 가능합니다.

(2) 고급 인덱스

함수 기반 인덱스

- 함수 기반 인덱스는 열 표현식(가상 열)에 대해 인덱스를 생성합니다.
- 함수기반 인덱스는 WHERE 절에 표현식을 포함하는 명령문을 효율적으로 평가하는 방식을 제공합니다.
- 함수기반 인덱스의 용도
 - 계산 집중 표현식 구체화
 - 대소문자를 구분하지 않는 검색

```
CREATE INDEX emp_upper_lastname_ix  
ON employees(upper(last_name));
```

```
SELECT *  
FROM   employees  
WHERE  upper(last_name) = 'KING';
```

비트맵 인덱스

- 일반 B* 트리 인덱스와 비교해 볼 때 저기수 열의 경우는 비트맵 인덱스가 더 빠르며 더 적은 공간을 사용합니다.
- 각 비트맵 인덱스는 비트맵이라는 저장 단위로 구성됩니다.
- 각 비트맵에는 인덱스화된 각 열의 특정 값에 대한 정보가 포함되어 있습니다.
- 비트맵은 내부적으로 B* 트리 구조로 저장되어 액세스 성능을 최대화합니다.
- 비트맵 인덱스는 열의 구분 키 수가 테이블 총 행 수의 1% 이하인 저 기수 열에 적합합니다.

비트맵 인덱스 구조

- 비트맵의 각 위치는 특정 행에 대한 정보를 저장합니다.
- 비트맵 인덱스를 생성하려면 다음과 같이 CREATE INDEX 명령을 사용하되, 키워드인 BITMAP을 추가합니다.

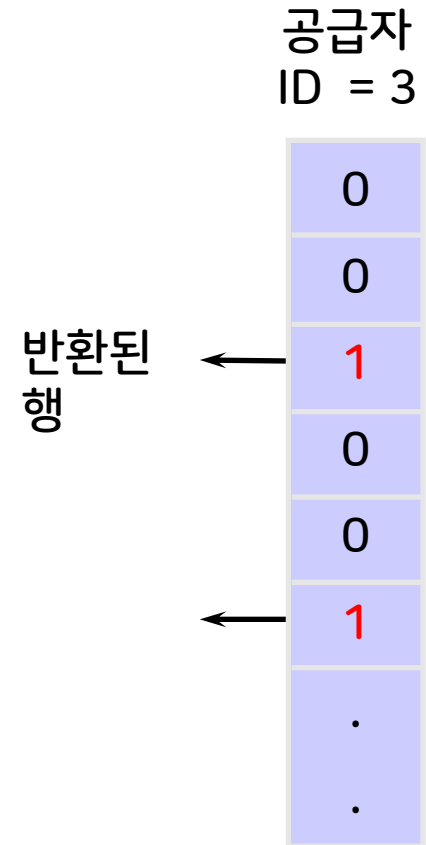
```
CREATE BITMAP INDEX prod_supplier_id  
ON products (supplier_id);
```

행 값	Supplier ID 1	Supplier ID 2	Supplier ID 3	Supplier ID 4	...
'1'	1	0	0	0	
'2'	0	1	0	0	
'3'	0	0	1	0	
'4'	0	0	0	1	
'2'	0	1	0	0	
'3'	0	0	1	0	
.	
.	

질의에 비트맵 인덱스 사용

- 비트맵 열과 관련된 WHERE 조건이 포함된 질의는 비트맵 인덱스를 사용할 수 있습니다.
- SUPPLIER_ID가 '3'인 행 수를 계산하는 질의가 있을 경우, 질의에서는 해당 비트맵을 사용하여 1의 수만 계산하면 될 뿐, 실제 행은 필요하지 않을 수 있습니다.

```
SELECT *  
FROM products  
WHERE supplier_id = 3;
```



비트맵 인덱스 결합

- 속도가 빠른 비트 단위의 AND, MINUS, OR 연산으로 인해 비트맵 인덱스가 효율적입니다.
 - IN을 사용하는 경우(value_list)
 - 술어가 AND/OR로 결합되는 경우

```
SELECT * FROM products WHERE supplier_id IN (1, 3);  
SELECT * FROM products  
WHERE supplier_id = 3 AND prod_pack_size = 'card box';
```

A	B	A and B	A or B	not A
1	1	1	1	0
0	1	0	1	1
1	0	0	1	0
0	0	0	0	1
0	1	0	1	1

비트맵 인덱스의 장점

- 적절히 사용할 경우 비트맵 인덱스는 다음과 같은 장점을 제공합니다.
 - 많은 임시 질의에 대해 응답 시간 감소
 - 다른 인덱스 기법에 비해 공간 사용을 상당히 감소
 - 성능이 낮은 하드웨어에서도 성능을 현저하게 개선
- 다음과 같은 경우에 비트맵 인덱스를 사용합니다.
 - 저기수 열
 - 다음 위치에서 자주 사용되는 열:
 - 복합 WHERE 절 조건
 - 그룹 함수(COUNT, SUM 등)
 - 대용량 테이블
 - 임시 질의가 많고 동시 DML 변경이 거의 없는 DSS 시스템

DML 작업이 인덱스에 주는 영향

- 삽입으로 인해 적절한 블록에 인덱스 항목이 삽입됩니다(블록이 분할될 수 있음).
- 삭제로 인해 인덱스 항목이 논리적으로 삭제됩니다(빈 블록은 사용할 수 있게 됨).
- 키 열을 갱신하면 인덱스에 논리적인 삭제 및 삽입이 발생합니다.

인덱스 사용 모니터 활성화 및 비활성화

- Oracle은 인덱스 사용 통계를 수집할 수 있는 기능을 제공합니다.
 - DML 작업 중 불필요한 오버헤드 제거를 통한 성능 향상

- 인덱스 사용 모니터

- 인덱스 사용 모니터를 시작하는 방법:

```
ALTER INDEX customers_pk MONITORING USAGE;
```

- 인덱스 사용 모니터를 정지하는 방법:

```
ALTER INDEX customers_pk NOMONITORING USAGE;
```

- 새 V\$OBJECT_USAGE 뷰에서는 다음 열과 함께 사용되지 않는 인덱스를 식별할 수 있습니다

Thank you 😊