**2.3 인덱스 확장기능 사용법**

**1. Index Range Scan**

* B\*Tree 인덱스의 가장 일반적이고 정상적인 형태의 액세스 방식
* 인덱스 루트에서 리프 블록까지 수직적으로 탐색한 후 ‘필요한 범위(Range)만’ 스캔
* Index Range Scan 주의 및 참고사항
* 선두 컬럼을 가공하지 않은 상태로 조건 절에 사용
* 성능은 인덱스 스캔 범위, 테이블 액세스 횟수에 좌우됨

\* set autotrace traceonly explain: SQL을 실행하지않고 실행계획만을 출력

\* autotrace 옵션 참고: <http://wiki.gurubee.net/display/DBSTUDY/Chapter.03+AutoTrace>

**2. Index Full Scan**

* 수직적 탐색 없이 인덱스 리프 블록을 처음부터 끝까지 수평적으로 스캔
* 데이터 검색을 위한 최적의 인덱스가 없을 때 차선으로 선택/적용
* 결합 인덱스로 구성된 인덱스에서 특정 인덱스 컬럼(선두 컬럼)이 없을 경우 - p112 SQL 참조

1. Index Full Scan의 효용성

* 데이터 저장공간: ‘가로 \* 세로’: ‘컬럼 길이 \* 레코드 수’
* 인덱스 스캔 단계에서 대부분의 레코드를 필터링하고 아주 일부만 테이블을 액세스 하는 상황에서 사용(데이터 저장공간의 크기: 테이블 > 인덱스)
* 테이블 액세스 수행빈도가 낮은(조회 대상 데이터가 적은) SQL의 경우 그 효용성이 높음 – [그림 2-17]: p113 참조 ⬄ 반대의 경우 Table Full Scan

1. 인덱스를 이용한 소트 연산 생략 – [그림 2-18]: p114 참조 - /\*+first\_rows\*/ 힌트 사용

* 결과집합이 인덱스 컬럼 순으로 정렬되므로 옵티마이저가 전략적으로 선택한(사용자가 특정 힌트를 사용한) 경우, Sort Order By 연산 과정이 생략 됨 - p114 SQL 참조
* 부분범위 처리가 가능한 상황에서 극적인 성능 개선 효과를 볼 수 있음.
* 옵티마이저 모드: 7.2.2 에서 자세히 다룸
* 부분범위 처리 개념: 3.2 에서 자세히 다룸

**3. Index Unique Scan**

* 수직적 탐색만으로 데이터를 찾는 스캔 방식
* Unique 인덱스를 ‘=’ 조건으로 탐색하는 경우에 작동
* From ‘Index Unique Scan’ to ‘Index Range Scan’ 변환
* Unique 인덱스에 범위검색 조건(between, 부등호, like)을 사용하여 검색할 경우
* Unique 결합 인덱스에 대해 일부 컬럼만으로 검색할 경우

Ex) 주문상품 PK 인덱스:『주문일자+고객ID+상품ID』 | 사용 조회 조건: 『주문일자+고객ID』

**4. Index Skip Scan**

* 인덱스 선두 컬럼을 조건절에 사용하지 않을 경우 사용
* 조건절에 빠진 인덱스 선두 컬럼의 Distinct Value 개수가 적고 후행 컬럼의 Distinct Value 개수가 많을 때 유용.
* B\*Tree 인덱스 구조에서 루트 또는 브랜치 블록에서 읽은 컬럼 값 정보를 이용해 조건절에 부합하는 레코드를 포함할 ‘가능성이 있는’ 리프 블록만 골라서 액세스 하는 스캔 방식 사용 – [그림 2-20]: p117 참조 – Index Range Scan | Index Skip Scan 비교
* index\_ss 힌트 사용

1. Index Skip Scan이 작동하기 위한 조건

* 선두 컬럼에 대한 조건절은 있고, 중간 컬럼에 대한 조건절이 없는 경우

Ex) 일별업종별거래\_PK: 업종유형코드+업종코드+기준일자 / 조건절: 업종유형코드+기준일자

* 선두 컬럼이 범위검색(부등호, BETWEEN, LIKE) 조건일 경우

**5. Index Fast Full Scan** -[그림 2-22, 2-23] : p122~123 참조

* 논리적인 인덱스 트리 구조를 무시하고 인덱스 세그먼트 전체를 물리적으로 디스크에 저장된 순서로 Multiblock I/O 방식으로 스캔
* index\_ffs와 no\_index\_ffs 힌트 사용
* Index Full Scan과 Index Fast Full Scan 비교 – [표 3-4]: p124

|  |  |
| --- | --- |
| Index Full Scan | Index Fast Full Scan |
| 1. 인덱스 구조를 따라 스캔  2. 결과집합 순서 보장  3. Single Block I/O  4. (파티션 돼 있지 않다면) 병렬스캔 불가  5. 인덱스에 포함되지 않은 컬럼 조회 시에도 사용 가능 | 1. 세그먼트 전체를 스캔  2. 결과집합 순서 보장 안 됨  3. Multiblock I/O  4. 병렬스캔 가능  5. 인덱스에 포함된 컬럼으로만 조회할 때 사용 가능 |

**6. Index Range Scan Descending**

* Index Range Scan과 기본적으로 동일한 스캔 방식
* Index Range Scan과 달리 인덱스를 뒤에서부터 앞쪽으로 스캔하기 때문에 내림차순으로 정렬된 결과집합을 얻음
* Index\_desc 힌트를 사용: 옵티마이저가 인덱스를 거꾸로 읽지 않을 경우 사용
* 특정 칼럼에 인덱스가 있을 MAX 갑을 구하는 경우에 사용(FIRST ROW 읽음): p125 참조
* 5.3.3 에서 자세히 다룸