**7.1 통계정보와 비용 계산 원리** – 옵티마이저의 통계정보 활용 방법 확인

**1. 선택도와 카디널리티**

● 선택도(Selectivity): 전체 레코드 중에서 조건절에 의해 선택되는 레코드 비율. 컬럼 값 종류 개수(Number of Distinct Values, 이하 ‘NVL’)를 이용 – **옵티마이저 비용 계산의 출발점**

☞ **선택도 = 1 / NVL** : 등치(‘=’) 조건으로 검색할 때의 선택도

● 카디널리티(Cardinality): 전체 레코드 중에서 조건절에 의해 선택되는 레코드 개수

☞ **카디널리티 = 총 로우 수 \* 선택도 = 총 로우 수 / NVL**

Ex) 전체 레코드가 10만 건이며, 상품분류 컬럼에 ‘가전’, ‘의류’, ‘식음료’, ‘생활용품’ 네 개의 값이 있을 때 아래 조건 절에 대한 선택도와 카디널리티

『 WHERE 상품분류 = ‘가전’ 』

=> NVL = 4 | 선택도 = 1/4 = 25% | 카디널리티 = 100,000 \* 1/4 = 25,000

**2. 통계정보**

통계정보

시스템 통계

오브젝트 통계

테이블 통계

인덱스 통계

컬럼 통계(히스토그램 포함)

1) 테이블 통계

① 테이블 통계 수집 명령어

|  |
| --- |
| **BEGIN**  dbms\_stats.gather\_table\_stats('scott', 'emp');  **END**; |

② 수집된 테이블 통계정보 조회

|  |
| --- |
| **SELECT** num\_rows, blocks, avg\_row\_len, sample\_size, last\_analyzed  **FROM** ALL\_TABLES -- or all\_tab\_statistics 뷰 사용  **WHERE** OWNER = 'SCOTT'  **AND** TABLE\_NAME = 'EMP'; |

> 실행 결과(DBever 실행 결과)

7.1.2.2_new.PNG

> 주요 테이블 통계항목: [표 7-1]

|  |  |
| --- | --- |
| 통계항목 | 설명 |
| NUM\_ROWS | 테이블에 저장된 총 레코드 개수 |
| BLOCKS | 테이블 블록 수 = ‘사용된’ 익스텐트(데이터가 한 건이라도 입력된 적이 있는 모든 익스텐트)에 속한 총 블록 수 |
| AVG\_ROW\_LEN | 레코드당 평균 길이(Bytes) |
| SAMPLE\_SIZE | 샘플링한 레코드 수 |
| LAST\_ANALYZED | 통계정보 수집일시 |

※ 테이블에 ‘할당된’ 총 블록 수는 dba\_segments 또는 user\_segments 뷰에서 확인 가능

2) 인덱스 통계

① 인덱스 통계 수집 명령어

|  |
| --- |
| -- 인덱스 통계만 수집  **BEGIN**  dbms\_stats.gather\_index\_stats (ownname => 'scott', indname => 'PK\_EMP');  **END**;  -- 테이블 통계를 수집하면서 인덱스 통계도 같이 수집  **BEGIN**  dbms\_stats.gather\_table\_stats ('scott', 'emp', **CASCADE**=>**true**);  **END**; |

② 수집된 인덱스 통계정보 조회

|  |
| --- |
| **SELECT** blevel, leaf\_blocks, num\_rows, distinct\_keys  , avg\_leaf\_blocks\_per\_key, avg\_data\_blocks\_per\_key, clustering\_factor  , sample\_size, last\_analyzed  **FROM** ALL\_INDEXES -- or all\_ind\_statistics 뷰 사용  **WHERE** OWNER = 'SCOTT'  **AND** TABLE\_NAME = 'EMP'  **AND** INDEX\_NAME = 'PK\_EMP'; |

> 수집된 인덱스 통계정보 조회 결과

7.1.2.5.PNG

> 인덱스 통계항목: [표 7-2]

|  |  |  |
| --- | --- | --- |
| 통계항목 | 설명 | 용도 |
| BLEVEL | 브랜치 레벨의 약자.  인덱스 루트에서 리프 블록에 도달하기 직전까지 읽게되는 블록 수 | 인덱스 수직적 탐색 비용 계산(2.1, 3.3) |
| LEAF\_BLOCKS | 인덱스 리프 블록 총 개수 | 인덱스 수평적 탐색 비용 계산(2.1,3.3) |
| NUM\_ROWS | 인덱스에 저장된 레코드 개수 |
| DISTINCT\_KEYS | 인덱스 키값의 조합으로 만들어지는 값의 종류 개수. 인덱스 키값을 모두 ‘=’ 조건으로 조회할 때의 선택도(Selectivity)를 계산하는데 사용. |
| AVG\_LEAF\_BLOCKS\_PER\_KEY | 인덱스 키값을 모두 ‘=’ 조건으로 조회할 때 읽게 될 리프 블록 개수 |
| AVG\_DATA\_BLOCKS\_PER\_KEY | 인덱스 키값을 모두 ‘=’ 조건으로 조회할 때 읽게 될 테이블 블록 개수 | 테이블 액세스 비용 계산(3.1.2) |
| CLUSTERING\_FACTOR | 인덱스 키값 기준으로 테이블 데이터가 모여 있는 정도.  인덱스 전체 레코드를 스캔하면서 테이블 레코드를 찾아갈 때 읽게 될 테이블 블록 개수를 미리 계산해 놓은 수치(3.1.2) |

※ 테이블 인덱스 조회 Query

**SELECT** table\_name, index\_name, column\_name **FROM** ALL\_IND\_COLUMNS **WHERE** TABLE\_NAME='EMP';

3) 컬럼 통계: 테이블 통계 수집할 때 함께 수집됨.

① 수집된 통계정보 조회

|  |
| --- |
| **SELECT** num\_distinct, density, avg\_col\_len, low\_value, high\_value, num\_nulls  , sample\_size, last\_analyzed  **FROM** ALL\_TAB\_COLUMNS -- or all\_tab\_col\_statistics  **WHERE** OWNER = 'SCOTT'  **AND** TABLE\_NAME = 'EMP' **AND** COLUMN\_NAME = 'DEPTNO'; |

> 컬럼 통계 조회 실행 결과

7.1.2.6.PNG

> 주요 컬럼 통계항목: [표 7-3]

|  |  |
| --- | --- |
| 통계항목 | 설명 |
| NUM\_DISTINCT | 컬럼 값의 종류 개수(NDV). |
| DENSITY | ‘=’ 조건으로 검색할 때의 선택도를 미리 구해 놓은 값.  히스토그램이 없거나, 있더라도 100% 균일한 분포를 갖는다면, 1/NUM\_DISTINCT 값과 일치 |
| AVG\_COL\_LEN | 컬럼 평균 길이(Bytes) |
| LOW\_VALUE | 최소 값 |
| HIGH\_VALUE | 최대 값 |
| NUM\_NULLS | 값이 NULL인 레코드 수 |

▣ 컬럼 히스토그램

: 데이터 분포가 균일하지 않은(‘=’ 조건에 대한 올바른 선택도를 구할 수 없는) 컬럼에 대한 선택도를 구할 경우 사용.

① 히스토그램 - 도수분포표. (p506. [그림 7-1] 참조)

컬럼 값별로 데이터 비중 또는 빈도를 미리 계산해 놓은 통계정보.

실제 데이터를 읽어서 계산해 둔 값이므로 데이터 분포가 많이 변하지 않는 한 거의 정확.

② 히스토그램 유형: [표 7-4]

|  |  |
| --- | --- |
| 히스토그램 유형 | 설명 |
| 도수분포(FREQUENCY) | 값별로 빈도수 저장 |
| 높이균형(HIGHT-BALANCED) | 각 버킷의 높이가 동일하도록 데이터 분포 관리 |
| 상위도수분포(TOP-FREQUENCY) | 많은 레코드를 가진 상위 n개 값에 대한 빈도수 저장 (12c 이상) |
| 하이브리드(HYBRID) | 도수분포와 높이균형 히스토그램의 특성 결합 (12c 이상) |

③ 히스토그램 수집

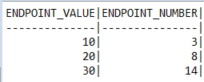
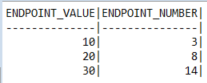
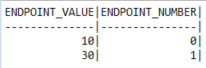
- 테이블 통계 수집할 때 method\_opt 파라미터를 지정하여 설정

|  |
| --- |
| -- P.1  **BEGIN**  dbms\_stats.gather\_table\_stats ('scott', 'emp', **CASCADE**=>**FALSE**  , method\_opt=>'for columns ename size 10, deptno size 4');  **END**;  -- P.2  **BEGIN**  dbms\_stats.gather\_table\_stats ('scott', 'emp', **CASCADE**=>**FALSE**  , method\_opt=>'for all columns size 75');  **END**;  -- P.3  **BEGIN**  dbms\_stats.gather\_table\_stats ('scott', 'emp', **CASCADE**=>**FALSE**  , method\_opt=>'for all columns size auto');  **END**; |

④ 수집된 컬럼 히스토그램 조회

|  |
| --- |
| **SELECT** endpoint\_value, endpoint\_number  **FROM** ALL\_HISTOGRAMS -- or all\_tab\_histograms  **WHERE** OWNER = 'SCOTT' **AND** TABLE\_NAME = 'EMP' **AND** COLUMN\_NAME = 'DEPTNO'  **ORDER** **BY** endpoint\_value; |

> 수집된 컬럼 히스토그램 조회 실행 결과(P.1, P.2, P.3 순)

4) 시스템 통계

- 애플리케이션 및 하드웨어 성능 특성을 측정한 것

① 시스템 통계 측정 대상

* CPU 속도
* 평균적인 Single Block I/O 속도
* 평균적인 MultiBlock I/O 속도
* 평균적인 MultiBlock I/O 개수
* I/O 서브시스템의 최대 처리량(Throuhput)
* 병렬 Slave의 평균적인 처리량(Throuhput)

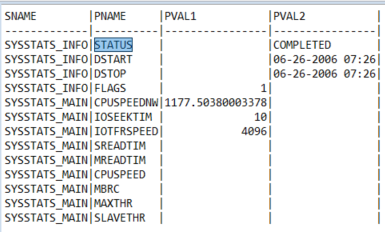
=> 옵티마이저는 시스템 사양뿐만 아니라 애플리케이션 특성(OLTP, DW) 및 동시 트랜잭션 발생량에 따라서도 이들 성능 특성의 달라짐.

옵티마이저가 보다 합리적으로 작동할 수 있게 하려고 9i부터 시스템 통계 수집 기능을 도입

② 시스템 통계 조회 – 『sys.aux\_stats$』 뷰에서 조회

|  |
| --- |
| **SELECT** SNAME, PNAME, PVAL1, PVAL2 **FROM** SYS."AUX\_STATS$"; |

> 시스템 통계 조회 결과



> 시스템 통계 조회 항목

|  |  |  |
| --- | --- | --- |
| 항목 | 설명 | Workload 여부 |
| CPUSPEEDNW | CPU speed | No Workload (NW) stats |
| IOSEEKTIM | The I/O seek time in milliseconds |
| IOTFRSPEED | I/O transfer speed in milliseconds |
| SREADTIM | Single block read time in milliseconds | Workload-related stats |
| MREADTIM | Multiblock read time in ms |
| CPUSPEED | CPU speed |
| MBRC | Average blocks read per multiblock read |
| MAXTHR | Maximum I/O throughput (for OPQ only) |
| SLAVETHR | OPQ Factotum (slave) throughput (OPQ only) |

※ OPQ: Oracle Parallel Query

**3. 비용 계산 원리**

1) 단일 테이블에 대한 인덱스 액세스할 때의 비용 계산 원리

① 인덱스 키값을 모두 ‘=’ 조건으로 검색할 경우

|  |
| --- |
| 비용 = BLEVEL -- 인덱스 수직적 탐색 비용  + AVG\_LEAF\_BLOCKS\_PER\_KEY -- 인덱스 수평적 탐색 비용  + AVG\_DATA\_BLOCKS\_PER\_KEY -- 테이블 랜덤 액세스 비용 |

② 인덱스 키값이 모두 ‘=’ 조건이 아닐 경우: 컬럼 통계까지 활용

|  |
| --- |
| 비용 = BLEVEL -- 인덱스 수직적 탐색 비용  + LEVEL\_BLOCKS \* 유효 인덱스 선택도 -- 인덱스 수평적 탐색 비용  + CLUSTERING\_FACTOR \* 유효 테이블 선택도 -- 테이블 랜덤 액세스 비용 |

=> BLEVEL, LEVEL\_BLOCKS, CLUSTERING\_FACTOR는 인덱스 통계에서 얻을 수 있고, 유효 인덱스 선택도와 유효 테이블 선택도는 컬럼 통계 및 히스토그램을 이용해 계산

▶ 유효 인덱스 선택도

- 전체 인덱스 레코드 중 액세스 조건(스캔 범위를 결정하는 조건절)에 의해 선택될 것으로 예상되는 레코드 비중(%) – p185: [그림 3-30] 참조

▶ 유효 테이블 선택도

- 전체 인덱스 레코드 중 인덱스 컬럼에 대한 모든 조건절에 의해 선택 될 것으로 예상되는 레코드 비중(%) : 『인덱스 액세스 조건 + 인덱스 필터 조건』– p185: [그림 3-30] 참조

※ 비용(Cost)의 정확한 의미

* I/O 비용 모델: 예상 I/O Call 횟수
* CPU 비용 모델: Single Block I/O를 기준으로 한 상대적 시간을 표현 - Single Block I/O 처리 횟수로 환산하여 표시