**1. 인덱스(INDEX)의 개요**

Oracle8 Server의 인덱스는 포인터를 사용하여 행의 검색을 촉진시킬 수 있는 스키마 객체입니다. 인덱스는 명시적 또는 자동적으로 생성 할 수 있으며 Column에 대한 인덱스가 없으면 한 테이블 전체를 검색하게 될 것입니다. 인덱스는 QUERY의 성능을 향상시키려면 인덱스 생성을 고려해야 한다. 또한 Column이나 Column의 집합에 값이 유일하도록 하기 위해 인덱스를 사용할 수 있다.

**1.1 인덱스의 특징**

1. 인덱스는 테이블의 값을 빠르게 액세스 하도록 하는 데이터베이스 객체이다.
2. 데이터를 빠르게 찾기 위한 B\*TREE을 써서 디스크 입출력 횟수를 줄인다.
3. Oracle8 Server가 인덱스를 자동적으로 사용하고 유지 보수 한다.
4. 인덱스를 만들면 사용자가 직접 조작할 필요가 없게 된다.
5. 인덱스는 논리적으로도 물리적으로도 테이블과는 독립적이다.
6. 언제든지 생성하거나 삭제할 수 있으며 이는 테이블이나 다른 인덱스에 영향을 주지 않는다는 의미이다.

♣ 참고

B\* INDEX에 대하여

1. 어떤 행에 대한 액세스 횟수도 동일하게 하는 이진의 균형 탐색 TREE이다.
2. 행이 테이블 시작이나 중간, 또는 끝에 있어도 거의 같은 횟수 내에 지정된 값을 액세스 하는 효율적인 방법이다.
3. Oracle8 Server가 만드는 인덱스는 TREE에 정렬된 여러 개의 저장소 PAGE로 구성된다.
4. 각 페이지는 키 값이 데이터 자체를 가리킬 때까지 구조의 아래 쪽으로 향하는 페이지에 대한 포인터와 일련의 키 값을 갖고있다.

**1.2 인덱스 생성 방법**

**1.2.1 자동 생성**

테이블 정의에 PRIMARY KEY나 UNIQUE 제약 조건을 정의할 때 unique 인덱스가 자동적으로 생성된다.

**1.2.2 사용자가 생성**

행에 대한 액세스 속도를 빠르게 하기 위해 Column에 non\_unique 인덱스 또는 unique 인덱스를 생성한다.

**1.3 인덱스의 종류**

|  |  |
| --- | --- |
| 종 류 | 설 명 |
| Unique index | 지정된 열의 값이 고유함을 보장 |
| Non-unique index | 데이터를 검색할 때 가장 빠른 결과를 보장 |
| Single column index | 하나의 열만 인덱스에 존재 |
| Composite Index | 여러 열을 결합하여 하나의 인덱스를 생성(16개의 열까지) |

**1.4 사용자가 인덱스 생성**

CREATE INDEX문장을 이용함으로써 하나 이상의 열의 인덱스를 생성합니다.

**1.4.1 Syntax**

|  |
| --- |
| **CREATE INDEX** index\_name  **ON** table\_name (column1[,column2,.....]); |

**1.4.2 인덱스 생성을 위한 지침**

**가) 많은 것이 항상 더 좋은 것은 아니다.**

테이블의 많은 인덱스가 질의의 스피드 향상을 꼭 의미하는 것은 아닙니다. 인덱스를 가지고 있는 테이블에 대한 각DML작업은 인덱스도 갱신되어야 함을 의미합니다. 많은 인덱스가 테이블과 관련되어 있으면, ORACLE SERVER은 DML후에 모든 인덱스를 갱신하기 위해 더 많은 노력이 필요하게 됩니다.

**나) 언제 인덱스를 생성하는가?**

1. 열은 WHERE절 또는 조인 조건에서 자주 사용됩니다.
2. 열은 광범위한 값을 포함합니다.
3. 열은 많은 수의 null값을 포함합니다.
4. 둘 또는 이상의 열은 WHERE절 또는 조인 조건에서 자주 함께 사용됩니다.
5. 테이블은 대형이고 대부분의 질의들은 행의 2~4%보다 적게 읽어 들일 것으로 예상됩니다.

**다) 언제 인덱스를 생성해서는 안되는가**

1. 테이블이 작다.
2. 열이 질의의 조건으로 자주 사용되지않는다.
3. 대부분의 질의들은 행의 2~4%이상을 읽어 들일 것으로 예상된다.
4. 테이블은 자주 갱신됩니다. 테이블에 하나 이상 인덱스를 가지고 있다면 테이블을 엑세스하는 DML문장은 인덱스의 유지 때문에 상대적으로 더 많은 시간이 걸리게 됩니다.

♣ 참고

유일성을 강행하기를 원한다면, 테이블 정의에 유일한 제약 조건을 정의해야 함을 명심하십시오. 이때 유일한 인덱스는 자동으로 생성됩니다.

문제1) EMP 테이블에서 ename열에 인덱스를 생성하여라

|  |
| --- |
| SQL> **CREATE INDEX emp\_ename\_idx**  2 **ON emp (ename);**  Index created. |

**1.5 인덱스 생성 확인**

1. USER\_INDEXES는 인덱스 이름과 고유성 정보를 가지고 있다.
2. USER\_IND\_COLUMNS는 인덱스 명, 테이블 명, 열명을 가지고 있다.

문제2) EMP 테이블에 이미 생성되어 있는 인덱스,관련된 열명,고유성 여부를 출력하여라.

|  |
| --- |
| SQL> **SELECT c.index\_name,c.column\_name,**  2 **c.column\_position,i.uniqueness**  3 **FROM user\_indexes i,user\_ind\_columns c**  4  **WHERE c.index\_name = i.index\_name**  5 **AND c.table\_name = 'EMP';**  INDEX\_NAME COLUMN\_NAME COLUMN\_POSITION UNIQUENES  ------------------------ --------------------- --------------- ---------  EMP\_PRIMARY\_KEY EMPNO 1 UNIQUE |

♣ 참고

Oracle에서의 최적화 방법

가) 규칙 기준(Rule-base) 최적화

SQL문장을 실행하기 위한 계획(Execution plan)을 선택할 때 내부적인 규칙에 근거하여 적절한 INDEX를 선정하여 사용하는 방식

나) 비용 기준(Cost-base) 최적화

SQL문장을 실행하기 위한 계획(Execution plan)을 선택할 때 인덱스가 있는 테이블의 통계치를 분석하여 가장 비용이 적게 드는 방식으로 액세스 경로를 결정한다.

**1.6 인덱스 제거**

인덱스를 수정할 수 없습니다. 인덱스를 변경하기 위해서는, 그것을 제거하고 다시 작성해야 합니다. DROP INDEX문장을 생성하여 데이터 사전에서 인덱스 정의를 제거합니다. 인덱스를 제거하기 위해서는 인덱스의 소유자이거나 DROP ANY INDEX권한을 가지고 있어야 합니다.

**1.6.1 Syntax**

|  |
| --- |
| **DROP INDEX**  index\_name; |

문제3) emp\_ename\_idx 인덱스를 삭제하여라.

|  |
| --- |
| SQL> **DROP INDEX emp\_ename\_idx;**  Index dropped. |

☞ Guidelines

1. 인덱스를 수정할 수는 없다.
2. 인덱스를 변경하려면 삭제한 다음 다시 만들어야 한다.
3. DROP INDEX 명령을 사용하여 인덱스를 삭제하라.
4. 인덱스를 삭제하려면 그 인덱스의 소유자이거나 DROP ANY INDEX 권한을 가지고 있어야 한다.

**2. 동의어**

다른 사용자가 소유한 테이블을 참조하기 위해서는 동의어를 생성한 이름 뒤에 점을 찍고 테이블 이름을 써야 합니다. 동의어 생성은 스키마 이름까지 명시할 필요를 제거시키고 테이블, 뷰, 시퀀스, 프로시저, 또는 다른 객체에 대한 또 다른 이름을 제공 합니다. 이 방법은 뷰처럼 긴 이름을 가진 객체한테 유용하게 사용될 수 있습니다.

**2.1 Syntax**

|  |
| --- |
| **CREATE**  [**PUBLIC**] **SYNONYM** synonym\_name  **FOR**  object\_name; |

PUBLIC 모든 사용자에 대해 액세스 가능한 동의어를 생성

synonym\_name 생성 되어야 할 동의어 이름

object\_name 생성된 동의어에 대한 객체를 식별합니다.

☞ Guidelines

1. 객체는 패키지에 포함될 수 없습니다.
2. 개별 동의어 이름은 동일 사용자가 소유한 모든 다른 객체의 이름과 달라야 합니다.

문제4) SALGRADE 동의어로 GUBUN를 생성하고 동의어로 조회하여라.

|  |
| --- |
| SQL> **CREATE SYNONYM gubun**  2 **FOR salgrade;**  Synonym created.  SQL> **SELECT \***  2 **FROM gubun;**  GRADE LOSAL HISAL  --------- --------- ---------  1 700 1200  2 1201 1400  . . . . . . . . . . |

**2.2 동의어 삭제**

동의어를 제거하기 위해 DROP SYNONYM문장을 사용합니다. DBA만 공용(Public) 동의어를 제거할 수 있습니다.

**2.2.1 Syntax**

|  |
| --- |
| **DROP** [**PUBLIC**]  **SYNONYM**  synonym\_name; |

문제5) 앞에서 생성한 동의어를 삭제하여라.

|  |
| --- |
| SQL> **DROP SYNONYM gubun;**  Synonym dropped. |

**◈ 연 습 문 제 ◈**

1. EMP 테이블에서 이름을 가지고 INDEX(emp\_ename\_indx)를 생성하여라

2. INDEX의 장단점을 기술하여라.

3. 테이블 생성시 자동적으로 생성되는 INDEX가 있다. 어떤 제약 조건을 기술하면 생성되는가?

4. INDEX 생성 지침을 설명하여라.

5. 여러 개의 COLUMN으로 인덱스를 사용하는 경우가 있다. 몇개의 COLUMN까지 가능한가.

6. 동의어는 어떤 경우에 사용하는가.

7. EMP 테이블을 EMPLOYEE라는 동의어를 생성하여라.