

## 1. 인덱스(Index)

데이터베이스 성능을 향상시키기 위한 액세스 메커니즘으로 인덱스(index)를 사용한다. 인덱스는 테이블 내의 행에 대한 빠른 임의 액세스를 허용하기 위해 사용되며, 오라클에서 인덱스는 18자리의 ROWID와 칼럼 값으로 구성된다.

기본 키(primary key) 칼럼과 고유 키(Unique key) 칼럼은 자동으로 인덱스가 생성되므로 인덱스를 생성하지 않는다.

### 1.1. 인덱스 생성 기준

#### ☞ 인덱스는 생성 시 고려사항

- DML문의 WHERE절에 자주 사용하는 칼럼으로 만든다.
- Null 값이 많은 칼럼에 인덱스를 생성하면 실행속도가 빨라진다.
- WHERE절에 의해 검색되는 데이터 분포가 테이블 전체의 약 10~15% 범위에 속할 때 인덱스를 생성한다.
- 테이블의 크기가 큰 경우에 인덱스를 생성한다.
- 주민등록번호와 같이 넓게 분포된 칼럼에 인덱스를 생성한다.
- 불필요한 인덱스를 많이 생성하면, 실행속도가 오히려 지연되므로, 한 테이블에 3~4개 이내로 생성한다.
- 인덱스를 구성하는 칼럼이 16개를 초과할 수 없다.
- 성별과 같이 좁게 분포된 칼럼은 인덱스를 생성하지 않는다.
- 테이블의 데이터가 자주 변경되는 경우에는 생성하지 않는다.

### 1.2. 인덱스의 생성

CREATE [UNIQUE] INDEX문으로 테이블에 대한 인덱스를 생성한다. UNIQUE는 생략할 수 있다.

문법	CREATE [UNIQUE] INDEX 인덱스명 ON            테이블명 ( 칼럼명1, 칼럼명2, ...);
----	--

#### ※ 기술 방법

- 인덱스명 : 생성할 인덱스명을 기술한다.

- 테이블명(칼럼명1, ...): 인덱스가 생성될 테이블명가 칼럼명을 기술한다.

예) Student 테이블의 성명(Name) 칼럼으로 Student\_Name 인덱스를 생성하시오.

### 1.3. 인덱스의 삭제

DROP INDEX문으로 불필요한 인덱스를 삭제한다.

문법	DROP INDEX 인덱스명;
----	------------------

예) Student 테이블에 생성된 Student\_Name 인덱스를 삭제하시오.

## 2. 뷰

### 2.1. 뷰의 정의

하나 이상의 테이블의 쿼리에 기초한 저장된 쿼리이다. 뷰의 구조(혹은 칼럼)는 있으나, 뷰 자체에 데이터를 포함하거나 저장하고 있지 않기 때문에 가상 테이블이라고도 부른다. 사용자는 뷰로부터 테이블과 동일하게 필요한 정보를 검색할 수 있다.

#### ☞ 뷰의 용도

- 테이블의 접근에 관한 권한을 제한하고,
- 개발자나 사용자에게 복잡성을 감추고,
- 칼럼명을 변경하여 단순화할 때 사용한다.

EMPLOYEE

EMPNO	NAME	DEPT
1111	전지현	1
2222	송혜교	2

DEPARTMENT

DEPTNO	NAME	LOC
1111	인사부	서울
2222	전산부	대구

EMPNO	NAME	DEPT_NAME	LOC
1111	전지현	인사	서울
2222	송혜교	전산	대구

V\_EMP\_DEMP

## 2.2. 뷰에 관한 명령문

CREATE VIEW문은 SELECT문에 기술된 테이블이나 뷰 등에서 검색이 가능한 새로운 뷰의 객체를 생성한다.

문법	CREATE VIEW 뷰명  [(칼럼명1, 칼럼명2, ...)]  AS  SELECT문;
----	---

### ※ 기술방법

- 뷰명 : 생성할 뷰명을 기술한다.
- 칼럼명1, ... : 뷰에 포함시킬 칼럼명을 기술한다.
- SELECT문 : 뷰에서 검색할 데이터가 포함되는 SELECT문을 기술한다.

### ☞ 뷰 생성 시 고려사항

- 칼럼명을 생략하면, SELECT문의 SELECT절에 기술한 칼럼명으로 대체된다.
- 함수나, 수식, 리터럴 등이 사용되면 별명을 사용하거나, 칼럼명을 기술해야 된다.
- SELECT문에 사용하는 테이블의 기본 키나 NOT NULL 칼럼을 포함시켜야 데이터의 추가나 , 수정이 가능하다.

문법	[뷰 삭제 구문]  DROP VIEW 뷰명;
----	--------------------------------

예) 1. SG\_Scores 테이블로부터 'C0901' 학생의 성적[학번, 과목코드, 과목명, 성적]을 출력하시오.

2. 개인별 성적 조회를 위한 뷰를 생성하고, 이 뷰로부터 'C0901' 학생의 성적 정보(학번, 과목코드, 과목명, 점수)를 출력하시오.

3. Course 테이블과 Professor 테이블로부터 교수별 담당 과목 수를 출력하는 뷰를 생성하고, 이 뷰로부터 교수번호, 교수명, 과목 수를 출력하시오.

4. Student 테이블로부터 '컴공'학과 학생을 위한 Student\_Computer 뷰를 생성하고, 뷰로부터 (학과코드, 학년, 학번, 성명, 주민등록번호, 전화번호)를 출력하시오.

5. Student\_Computer 뷰에 다음(학과코드, 학년, 학번, 성명, 주민등록번호, 전화번호) 정보를 입력하여 확인하시오.

## 6. Student\_Computer 뷰를 삭제하시오.

### 2.3. 뷰의 제한 조건

#### ☞ 뷰를 통한 데이터의 트랜잭션이 가능하게 하려면

- GROUP BY절, DISTINCT, 그룹함수 등을 사용한 뷰가 아니어야 한다.
- 하나의 테이블에서 생성된 뷰이어야 한다.
- 수식이 사용된 필드는 수정, 삭제할 수가 없다.
- 기본 키, Not Null 로 설정된 칼럼이 모두 포함되어야 한다.

### 2.4. 인라인 뷰

SELECT문의 FROM절에 기술하는 서브 쿼리를 말한다. 서브 쿼리는 또 다른 서브 쿼리를 포함할 수도 있다. FROM절의 서브 쿼리에 별명을 사용하면 상위 쿼리에서 사용될 수 있다.

#### ☞ TOP-N 쿼리 구문

TOP-N 쿼리 또는 특정 위치로부터 일부 행을 검색할 수 있다. ROWNUM 의사 컬럼을 사용하여 인라인 뷰의 검색할 행의 수를 지정한다.

문법	<pre>SELECT  별명.칼럼명1, 별명.칼럼명2, ...  FROM (SELECT  칼럼명1, 칼럼명2, ...        FROM    테이블명, ...        ORDER BY 칼럼명 {ASC   DESC}, ...) 별명  WHERE ROWNUM &lt;= 행의 수;</pre>
----	--

예) 1. SG\_Scores 테이블을 성적과 학점을 내림차순으로 출력하고, 1 페이지에 출력할 성적 상위자 10명(1위~10위)의 학번, 성명, 과목명, 학점수, 성적, 등급을 출력하시오.

2. SG\_Scores 테이블에서 3 페이지에 출력할 성적 상위자 10명(21위~30위)에 대하여 학번, 성명, 과목명, 학점수, 성적을 출력하시오.

## 3. 시퀀스(Sequence)

데이터베이스 객체로, 시퀀스가 생성할 때 규칙에 따라 정수를 생성한다. 시퀀스는 행을 식별하는

기본 키 값을 자동적으로 생성하거나, 난수 생성에 사용한다. 기본 값을 1부터 시작하여 1씩 증가하고, 최대 15개까지 만들 수 있다.

### 3.1. 시퀀스 생성

CREATE SEQUENCE문으로 시퀀스를 생성한다.

문법	CREATE SEQUENCE 시퀀스명
	[ INCREMENT BY     증가값 ]
	[ START WITH        시작값 ]
	[ MAXVALUE          최대값 ]
	[ CYCLE             반복횟수 ]

#### ※ 기술 방법

- 시퀀스명 : 생성할 시퀀스명을 기술한다.
- INCREMENT BY 증가 : 시작값에서 증가할 값을 기술한다.
- START WITH 시작값 : 시퀀스의 초기값을 기술한다.
- MAXVALUE 최대값 : 시퀀스의 최종값을 기술한다.
- CYCLE 반복횟수 : 최대값이 되었을 때 반복횟수를 기술한다.

#### ☞ 시퀀스의 참조변수

시퀀스의 값을 반환하는 참조변수로 NEXTVAL과 CURRVAL이 있다.

기술	시퀀스명.참조변수
----	-----------

- 시퀀스명.NEXTVAL 변수를 시퀀스의 현재 값에 "증가값"을 더한 값을 정수로 반환한다.
- 시퀀스명.CURRVAL 변수를 시퀀스의 현재 값을 정수로 반환한다.

#### ☞ 시퀀스의 데이터를 다루는 참조변수 용도

- INSERT문에서 VALUES절, SELECT문에서 선택 목록, UPDATE문의 SET절에 사용할 수 있다.
- DISTINCT절이 있는 SELECT문, ORDER BY절, GROUP BY절이 있는 SELECT문, 서브 쿼리, 뷰 쿼리에서는 사용할 수 없다.

예) 1. Department 테이블에 대한 Dept\_Seq 시퀀스를 생성하고, Dept\_Seq 시퀀스를 이용하여

Department 테이블의 모든 행에 순서번호를 부여하여 출력하시오.

2. Student 테이블의 데이터를 Computer\_Student 테이블에 입력하여 출력하시오. 단, 학번은 st\_seq 시퀀스를 생성하여 이용한다.

### 3.2. 시퀀스 삭제

DROP SEQUENCE문으로 시퀀스를 삭제한다.

예) St\_Seq 시퀀스를 삭제하시오.

## 4. 생성된 테이블의 관리

사용자가 테이블, 뷰 등 생성된 객체를 관리하기 위한 몇 가지 기능을 알아보자.

### 4.1. 사용자의 객체명 출력

오라클 사용자에게 대하여 생성된 객체들을 조회할 수 있다. 테이블이나 뷰, 인덱스, 동의 등의 모든 객체들이 출력된다.

명령문	SELECT * FROM TAB;  SELECT * FROM USER_CATALOG;
-----	---

예) 사용자가 생성한 객체명을 모두 출력하시오.

### 4.2. 테이블의 구조 출력

Describe 명령어는 생성된 테이블이나 뷰의 구조를 출력한다.

명령어	Describe [ 혹은 desc ] 테이블명
-----	---------------------------

예) Score\_List 뷰의 구조를 출력하시오.

### 4.3. 테이블의 제약조건 출력

USER\_CONSTRAINTS 객체로부터 생성된 테이블에 대한 제약조건 정보를 출력할 수 있다. 제약조건의 Null과 Not Null은 describe 명령어로 출력할 수 있으나, 다른 제약조건들은 USER\_CONSTRAINTS 객체로부터 조회해야 알 수 있다. USER\_CONSTRAINTS 객체의 칼럼은 "Describe USER\_CONSTRAINTS"로 구조를 확인할 수 있다.

#### 🔑 USER\_CONSTRAINTS 객체의 주요 칼럼

- OWNER : 객체를 생성한 오라클 사용자를 반환한다.
- CONSTRAINT\_NAME : 제약조건명을 반환한다.
- CONSTRAINT\_TYPE : 제약조건에 대한 타입을 반환한다. 제약조건에 대한 타입은 기본 키(P), 외부 키(R), 고유 키(U), NOT NULL(C), 체크(C)로 구분.
- SEARCH\_CONDITION : 제약조건이 지정된 내용을 반환한다.
- TABLE\_NAME : WHERE 절에 조회할 테이블명을 기술한다.

문법	<pre>SELECT  * FROM    USER_CONSTRAINTS WHERE   TABLE_NAME = '테이블명';</pre>
----	--

예) Professor 테이블에 지정된 제약조건들을 모두 출력하시오.

#### 4.4. 제약조건의 활성화와 비활성화

테이블에 지정된 제약조건에 대하여 일시적으로 비활성화하거나, 활성화할 수 있다. 단, 기본 키의 제약조건은 비활성화가 불가능하다.

##### 🔑 제약조건 비활성화

테이블에 대한 제약조건을 "DISABLE"시키면 데이터 무결성을 검증하지 않는다.

문법	<pre>ALTER TABLE 테이블명 DISABLE CONSTRAINT 제약조건명;</pre>
----	---

##### 🔑 제약조건 활성화

제약조건이 "DISABLE"되어 입력된 데이터가 제약조건이 위배되면 제약조건을 활성화(ENABLE)할 수 없다.

문법	<pre>ALTER TABLE 테이블명 ENABLE CONSTRAINT 제약조건명;</pre>
----	--

예) 1. 교수(Professor) 테이블의 체크 제약조건(PROF\_CK)을 비활성화하고, 'S11', '강준상', '겸임교수', '컴공', '123-4567'의 값을 입력하시오.

2. Professor의 체크 제약조건(PROF\_CK)을 활성화시키시오.