

데이터 정의어(DDL) – 객체 생성



1. 데이터베이스 객체(Objects)

1.1 객체의 종류

객 체	설 명
TABLE	기본 저장 단위이며 행과 열로 구성
VIEW	하나 이상의 테이블에서 데이터 부분 집합을 논리적으로 표시
SEQUENCE	기본 키 값을 생성
INDEX	효율적인 질의를 위해 생성
SYNONYM	객체에 다른 이름을 제공



1.2 객체 이름 지정 규칙

- ☐ 문자로 시작
- ☐ 1~30자까지 가능
- ☐ A~Z, a~z, 0~9, _(밑줄), \$, #만 포함
- ☐ 동일한 사용자가 소유한 다른 객체의 이름과 중복되지 않아야 함
- ☐ Oracle Server의 예약어가 아니어야 함
- ☐ 이름은 대소문자를 구분하지 않음 (EMP = eMP, emp, EmP)



2. 테이블 (Table)

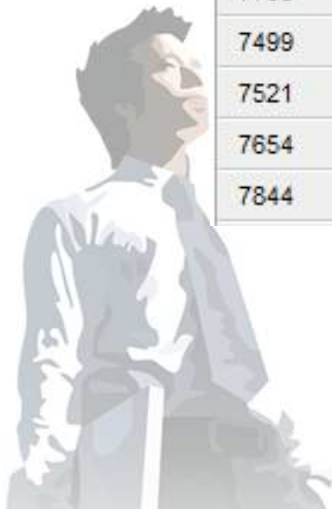
2.1 개요

☐ 테이블 (Table)이란 데이터베이스에서 데이터를 저장하는 가장 기본적인 객체

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	81/11/17	5000	-	10
7566	JONES	MANAGER	7839	81/04/02	2975	-	20
7698	BLAKE	MANAGER	7839	81/05/01	2850	-	30
7782	CLARK	MANAGER	7839	81/06/09	2450	-	10
7902	FORD	ANALYST	7566	81/12/03	3000	-	20
7788	SCOTT	ANALYST	7566	82/12/09	3000	-	20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON



2.2 테이블 구분

☐ 사용자 테이블

- 사용자가 생성 및 유지 관리하는 테이블의 모음
- 사용자 정보를 포함

☐ 데이터 디렉터리

- **Oracle Server**가 생성 및 유지 관리하는 테이블의 모음
- 데이터베이스 정보를 포함
- **sys** 사용자는 모든 데이터 디렉터리 테이블을 소유
- 기본 테이블의 정보는 이해하기 어렵기 때문에 일반적으로 이해하기 쉬운 데이터 디렉터리 뷰 (**view**)를 액세스 (**Access**)
(**USER_**, **ALL_**, **DBA_**, **V\$_**)

2.2 테이블 구분 (계속)

☐ 데이터 디렉터리 질의

- 사용자가 소유한 테이블의 이름 질의

```
SQL> SELECT table_name  
2 FROM user_tables;
```

- 사용자가 소유한 개별 객체 유형 질의

```
SQL> SELECT DISTINCT object_type  
2 FROM user_objects;
```

- 사용자가 소유한 테이블, 뷰, 동의어 및 시퀀스 질의

```
SQL> SELECT *  
2 FROM user_catalog;
```

3. 테이블 생성

3.1 개요

- ☐ CREATE TABLE 문을 실행하여 테이블 생성
- ☐ CREATE TABLE 권한 및 객체 생성에 대한 저장영역이 필요
- ☐ 테이블 이름, 열의 이름, 데이터 유형 및 크기 지정
- ☐ 다른 사용자의 테이블 참조
 - 다른 사용자에게 속한 테이블은 해당 사용자의 스키마에 있지 않음
 - 테이블에 소유자 이름을 접두어로 붙여야 함
- ☐ 테이블 생성 구문

```
CREATE TABLE [schema.]table  
      (column datatype [DEFAULT expr] [, ...]);
```


3.2 데이터 유형

데이터 유형	설 명
VARCHAR2(size)	가변 길이 문자 데이터 (최소크기:1, 최대크기:4000)
CHAR(size)	고정 길이 문자 데이터 (최소크기:1, 최대크기:2000)
NUMBER(p,s)	가변 길이 숫자 데이터 (p:전체 자릿수, s:소수점 이하 자릿수)
DATE	날짜 및 시간 값 (BC 4712년1월1일~AD 9999년 12월31일)
LONG	최고 2GB의 가변 길이 문자 데이터
CLOB	최고 4GB의 단일 바이트 문자 데이터
RAW(size)	길이인 size의 원시 이진 데이터 (최대크기:2000)
LONG RAW	최고 2GB의 가변 길이 원시 이진 데이터
BLOB	최고 4GB의 이진 데이터
BFILE	외부 파일에 저장된 이진 데이터 (최대크기:4GB)
ROWID	테이블에서 행의 고유 주소를 나타내는 64진수

3.3 DEFAULT 옵션

- ☞ 삽입 중에 열 (Column) 의 기본값을 지정
- ☞ DEFAULT datatype은 COLUMN의 datatype과 일치
- ☞ 리터럴 값, 표현식 또는 SQL 함수는 기본값으로 사용 가능 (sysdate, user)
- ☞ 다른 열의 이름이나 의사 열은 사용 불가능 (NEXTVAL, CURRVAL)

```
...,  
hiredate DATE DEFAULT SYSDATE,  
...
```



3.4 테이블 생성

```
SQL> CREATE TABLE sample
2  (sam1    NUMBER(4) ,
3    sam2    VARCHAR2(10) ,
4    sam3    VARCHAR2(20)) ;
```

테이블이 생성되었습니다.

```
SQL> DESCRIBE SAMPLE
```

이름	널?	유형
SAM1		NUMBER (4)
SAM2		VARCHAR2 (10)
SAM3		VARCHAR2 (20)

3.4 테이블 생성 (계속)

- ☐ 서브쿼리 (Subquery) 를 사용한 테이블 생성
 - 지정한 열 (Column) 수를 서브쿼리 열 수와 일치
 - 열 이름 및 기본값을 사용하여 열을 정의

```
CREATE TABLE table  
    [(column, column...)]  
AS subquery;
```

```
SQL> CREATE TABLE    dept30  
2  AS  
3  SELECT    empno, ename, sal*12 ANNSAL, hiredate  
4  FROM      emp  
5  WHERE     deptno = 30;
```

테이블이 생성되었습니다.

데이터 정의를어(DDL) - 객체 수정



1. 테이블 수정

1.1 열(Column) 추가

- ☑ 새로운 열은 마지막 열이 됨
- ☑ 열을 추가할 때 테이블이 이미 행을 포함하고 있으면, 새 열의 모든 행은 널(Null)

```
ALTER TABLE table  
ADD          (column datatype [DEFAULT expr]  
              [, column datatype]...);
```

```
SQL> ALTER TABLE dept30  
2 ADD          (job VARCHAR2(9));
```

테이블이 변경되었습니다.

1.2 열(Column) 수정

- ❑ 열의 데이터 유형, 크기 및 기본값을 변경할 수 있음
- ❑ 숫자 또는 문자 열의 너비 또는 전체 자릿수를 증가
- ❑ 열이 널(**Null**) 값만 포함하고 테이블에 행이 없는 경우 열의 너비 감소
- ❑ 열이 널 값을 포함하면 데이터 유형을 변경
- ❑ 기본값을 변경하면 변경 이후에 테이블에 삽입되는 항목에만 영향

```
ALTER TABLE table  
MODIFY      (column datatype [DEFAULT expr]  
              [, column datatype]...);
```

```
SQL> ALTER TABLE      dept30  
      2  MODIFY          (ename VARCHAR2(15));
```

테이블이 변경되었습니다.

1.3 열(Column) 삭제

- ❑ 테이블에서 더 이상 필요하지 않은 열 삭제
- ❑ 열은 데이터를 포함하거나 포함하지 않음
- ❑ 테이블 변경 후 테이블에 열이 하나 이상 있어야 함
- ❑ 삭제된 열은 복구할 수 없음

```
ALTER TABLE table  
DROP COLUMN column ;  
OR  
ALTER TABLE table  
DROP (column [, column, ...]) ;
```

```
SQL> ALTER TABLE      dept30  
      2 DROP COLUMN      job ;
```

테이블이 변경되었습니다.

2. SET UNUSED 옵션

2.1 개요

- ❑ SET UNUSED 옵션을 사용하여 하나 이상의 열을 사용하지 않음으로 표시
- ❑ DROP UNUSED COLUMNS 옵션을 사용하여 UNUSED로 표시된 열을 제거
- ❑ 실제로 테이블의 각 행에서 대상 열이 제거되지는 않음
- ❑ 사용 구문

```
ALTER TABLE  table  
SET UNUSED    (column) ;  
OR  
ALTER TABLE  table  
SET UNUSED COLUMN column ;
```

```
ALTER TABLE  table  
DROP  UNUSED COLUMNS ;
```

2.2 특징

- ❑ 열을 사용하지 않는 열로 표시한 경우 열 삭제와 동일한 기능을 나타냄
- ❑ 지정된 열이 실제로 디스크 공간에서 삭제되는 것이 아님
- ❑ **DROP** 절을 사용하여 열을 삭제할 때 소요되는 시간보다 빨라짐
- ❑ **DROP UNUSED COLUMNS** 옵션을 통해 디스크에서도 완전히 해제



2.3 활용

```
SQL> ALTER TABLE dept30
      2 SET UNUSED (hiredate);
```

테이블이 변경되었습니다.

```
SQL> DESC dept30
```

이름	날?	유형
-----	-----	-----
EMPNO		NUMBER (4)
EMPNAME		VARCHAR2 (15)
ANNSAL		NUMBER

```
SQL> ALTER TABLE dept30
      2 DROP UNUSED COLUMNS;
```

테이블이 변경되었습니다.

기타 데이터 정의어



1. 테이블 삭제

1.1 개요

- ❑ 테이블의 모든 데이터 및 구조 삭제
- ❑ 테이블을 삭제하면 테이블의 모든 데이터 및 관련된 인덱스도 모두 삭제
- ❑ 해당 테이블을 생성한 사용자와 **DROP ANY TABLE** 권한을 가진 사용자만 테이블 제거
- ❑ 테이블 삭제 구문

```
DROP TABLE table  
[CASCADE CONSTRAINTS];
```



1.2 활용

- ☐ 모든 뷰와 동의어는 그대로 남아 있지만 사용 불가
- ☐ 보류 중인 트랜잭션을 모두 커밋 (COMMIT)
- ☐ DROP TABLE문은 롤백 (ROLLBACK) 할 수 없음

```
SQL> DROP TABLE dept30;
```

테이블이 삭제되었습니다.



2. 객체 이름 변경

- ❑ 데이터베이스 객체 (테이블, 뷰, 시퀀스, 동의어 등)의 이름 변경
- ❑ 객체 이름을 변경하기 위해서는 사용자가 해당 객체의 소유자이어야 함
- ❑ 객체 이름 변경 구문

```
RENAME old_name TO new_name;
```

- ❑ 활용

```
SQL> RENAME sample TO sam;
```

테이블명이 바뀌었습니다.



3. 테이블 절삭

3.1 개요

- ☐ **TABLE**의 모든 행 제거
- ☐ 해당 **TABLE**이 사용하는 저장 공간 해제
- ☐ 테이블 절삭 구문

```
TRUNCATE TABLE table;
```



3.2 활용

- ☐ **TRUNCATE**를 사용한 행 제거 작업은 롤백(**rollback**) 할 수 없음
- ☐ 데이터 조작어(DML)의 **DELETE** 문과 유사
(단, **DELETE** 문은 저장 공간을 해제하지 않음)

```
SQL> TRUNCATE TABLE sam;
```

테이블이 잘렸습니다.



4. 주석 달기

- ☐ COMMENT 문을 사용하여 테이블 또는 열에 주석을 추가할 수 있음

```
COMMENT ON TABLE table | COLUMN table.column  
IS 'text' ;
```

```
SQL> COMMENT ON TABLE emp  
2 IS 'Employee Information' ;
```

주석이 생성되었습니다.

- ☐ 주석은 데이터 디렉터리 뷰를 통해 볼 수 있음

- ALL_COL_COMMENTS
- USER_COL_COMMENTS
- ALL_TAB_COMMENTS
- USER_TAB_COMMENTS