8장. 테이블 생성 수정 제거하기

이번 장에서는 테이블을 생성하는 명령문과 테이블의 구조를 변경하는 명령문, 기존 테이블의 존재 자체를 제거하는 명령문, 기존에 사용하던 테이블의 모든 로우를 제거하는 명령문을 학습합니다.

학습 내용

- 테이블 구조를 만드는 CREATE TABLE 문
- 테이블 구조를 변경하는 ALTER TABLE 문
- 테이블명 변경하는 RENAME문
- 테이블 구조를 제거하는 DROP TABLE 문
- 테이블의 모든 데이터를 제거하는 TRUNCATE TABLE 문
- 데이터 사전

학습목표

- 테이블을 생성할 수 있습니다.
- 테이블의 구조를 변경할 수 있습니다.
- 테이블을 제거하거나 이름을 변경할 수 있습니다.
- 테이블에 저장된 모든 데이터를 제거할 수 있습니다.

1-1. 테이블 구조를 만드는 CREATE TABLE 문

- DDL(데이터 정의어 : Data Definition Language)은 테이블을 생성, 수정, 제거하도록 하는 명령문 집합입니다.

- CREATE TABLE 문을 통해 데이터를 저장할 테이블을 생성할 수 있습니다.

형 식	CREATE TABLE [schema.] table (column datatype [DEFAULT expression] [column_constraint clause] [,...]) ;

- schema : 소유자의 이름(사용자 계정)
- column : 테이블에 포함되는 칼럼명
- datatype : 칼럼에 대한 데이터 타입과 길이
- DEFAULT expression: 데이터 입력 값이 생략된 경우 입력되는 기본 값
- column_constraint_clause : 칼럼에 정의되는 무결성 제약 조건

1-1. 테이블 구조를 만드는 CREATE TABLE 문

[예제] 각 부서의 부서번호, 부서이름, 위치를 입력할 수 있는 테이블을 생성하여 봅시다.

예 제	CREATE TABLE dept (dno number(2), dname varchar2(14), loc varchar2(13)) ;

1-2. 테이블 구조를 만드는 CREATE TABLE 문

- 앞에서 배운 서브 쿼리를 이용하여 다른 테이블로부터 테이블을 생성하는 것이 가능합니다. 다음을 입력하게 되면 테이블의 구조와 데이터를 복사하여 새로운 테이블을 생성합니다.

형 식	CREATE TABLE table [column [,column, ...]] AS subquery ;

- **주의 사항 : 컬럼명을 명시적으로 언급할 경우 지정한 칼럼 수, 데이터 타입이 SELECT 문에 의해 검색된 칼럼과 일치하여야 합니다. 언급하지 않을 경우 서브 쿼리의 칼럼명이 그대로 적용됩니다.**

1-2. 테이블 구조를 만드는 CREATE TABLE 문

- 앞에서 배운 서브 쿼리를 이용하여 다른 테이블로부터 테이블을 생성하는 것이 가능합니다. 다음을 입력하게 되면 테이블의 구조와 데이터를 복사하여 새로운 테이블을 생성합니다.

형 식	CREATE TABLE table [column [,column, ...]] AS subquery ;

- **주의 사항 : 컬럼명을 명시적으로 언급할 경우 지정한 칼럼 수, 데이터 타입이 SELECT 문에 의해 검색된 칼럼과 일치하여야 합니다. 언급하지 않을 경우 서브 쿼리의 칼럼명이 그대로 적용됩니다.**

2-1. 테이블 구조를 변경하는 ALTER TABLE 문 – 칼럼 추가

- ALTER TABLE 문을 사용하여 칼럼을 추가, 수정 또는 삭제할 수 있습니다.
- ALTER TABLE … ADD 구문을 사용하여 새로운 칼럼을 추가합니다.

형 식	ALTER TABLE table_name ADD ([column_name data_type DEFAULT expr] [, column_name data_type] …) ;

2-1. 테이블 구조를 변경하는 ALTER TABLE문 – 칼럼 추가

[예제] 사원 테이블에 date 타입을 가지는 birth라는 칼럼을 추가하여 봅시다.

예 제	ALTER TABLE dept20 ADD (birth DATE) ;

2-2. 테이블 구조를 변경하는 ALTER TABLE 문 – 칼럼 변경

❖ ALTER TABLE … MODIFY 구문을 사용하여 칼럼의 데이터 타입, 크기, 기본 값을 변경할 수 있습니다.

형 식	ALTER TABLE table_name MODIFY [column_name data_type DEFAULT expr] [, column_name data_type] …) ;

- 기존 칼럼에 데이터가 없는 경우에는 칼럼 타입이나 크기 변경이 자유롭지만, 데이터가 존재하는 경우 타입 변경은 CHAR, VARCHAR2만 허용되며 변경한 칼럼의 크기는 저장된 데이터의 크기와 같거나 클 경우에만 변경할 수 있습니다. 숫자 타입은 폭 혹은 전체 자릿수를 늘릴 수 있습니다.

- 변경 내용은 변경 후 입력되는 데이터부터 적용이 됩니다.

2-2. 테이블 구조를 변경하는 ALTER TABLE문 – 칼럼 변경

[예제] 사원 테이블의 칼럼의 크기를 변경해 봅시다.

예 제	ALTER TABLE dept20 MODIFY ename varchar2(30) ;

2-3. 테이블 구조를 변경하는 ALTER TABLE 문 – 칼럼 제거

- ALTER TABLE … DROP COLUMN 구문을 사용하여 특정 칼럼과 칼럼 내의 데이터를 제거할 수 있습니다.

- 2개 이상의 칼럼이 존재하는 테이블에서만 가능하며 한 번에 하나의 칼럼만 삭제할 수 있습니다. 삭제된 칼럼은 복구할 수 없습니다.

형 식	ALTER TABLE table_name DROP COLUMN column_ name ;

2-3. 테이블 구조를 변경하는 ALTER TABLE문 – 칼럼 제거

[예제] 부서 테이블에서 사원명 칼럼을 제거하여 봅시다.

예 제	ALTER TABLE dept20 DROP COLUMN ename ;

2-4. 테이블 구조를 변경하는 ALTER TABLE 문 – SET UNUSED

- SET UNUSED는 시스템 요구가 적을 때 칼럼을 제거할 수 있도록 하나 이상의 칼럼을 UNUSED로 표시합니다. 실제로는 테이블에서 해당 칼럼이 제거되지는 않습니다.
- DROP 명령보다 응답시간이 빨라집니다.
- 삭제된 것으로 처리하기 때문에 SELECT나 DESCRIBE를 사용할 수 없습니다.

2-4. 테이블 구조를 변경하는 ALTER TABLE문 – SET UNUSED

[예제] 부서 테이블에서 사원번호 칼럼을 UNUSED 상태로 만들어 봅시다.

예 제	ALTER TABLE dept20 SET UNUSED (eno) ;

[예제] DROP UNUSED COLUMNS를 사용하여 UNUSED로 표시된 모든 칼럼을 제거하여 봅시다.

예 제	ALTER TABLE dept20 DROP UNUSED COLUMNS ;

3. 테이블 명을 변경하는 RENAME 문

- RENAME 문은 테이블을 포함한 모든 객체의 이름을 변경하는 DDL명령문입니다.

형 식	RENAME old_name TO new_name ;

- old name은 기존 객체의 이름이고 new_name은 변경할 새 객체의 이름입니다.

[예제] RENAME 문을 이용하여 테이블 명을 변경하여 봅시다.

예 제	RENAME dept20 TO emp20 ;

4. 테이블 구조를 제거하는 DROP TABLE 문

- DROP TABLE 문을 사용하여 기존의 테이블과 데이터를 모두 제거할 수 있습니다.
- 삭제할 테이블의 기본 키나 고유 키를 다른 테이블에서 참조하고 있는 경우 삭제가 불가능하기 때문에, 그럴 경우 참조 중인 자식 테이블을 먼저 제거하여야 합니다.

[예제] DROP TABLE을 이용하여 테이블을 제거하여 봅시다.

예 제	DROP TABLE emp20 ;

5. 테이블의 모든 데이터를 제거하는 TRUNCATE문

- TRUNCATE 문은 기존의 테이블의 모든 로우를 제거하게 합니다.
- 테이블의 구조는 그대로 유지되며, 테이블의 데이터와 이에 할당된 공간이 해제됩니다.
- 제약 조건, 인덱스, 뷰, 동의어 등은 그대로 유지됩니다

형 식	TRUNCATE TABLE table_name ;

5. 테이블의 모든 데이터를 제거하는 TRUNCATE문

[예제] TRUNCATE 문을 이용하여 테이블의 모든 데이터를 제거하여 봅시다.

예 제	TRUNCATE TABLE dept_second ;

6. 데이터 사전

- 데이터 사전은 사용자와 데이터베이스 자원의 효율적 관리를 위한 다양한 정보를 저장하는 시스템 테이블의 집합입니다.
- 사용자가 테이블을 생성하거나 사용자를 변경하는 등의 작업을 할 때 데이터베이스 서버에 의해 자동으로 갱신되는 테이블입니다.
- 사용자는 데이터 사전의 내용을 직접 수정하거나 삭제할 수 없고 읽기 전용 뷰 형태로 사용자에게 제공됩니다.

접두어	의미
USER_	자신의 계정이 소유한 객체 등에 관한 정보 조회
ALL_	자신의 계정 소유했거나 권한을 부여 받은 객체 등에 관한 정보 조회
DBA_	데이터베이스 관리자만 접근 가능한 객체 등의 정보 조회

6-1. 데이터 사전 – USER_ 데이터 사전

- 접두어로 USER_가 붙은 데이터 사전은 사용자와 가장 밀접하게 관련된 뷰로서 자신이 생성한 테이블, 인덱스, 뷰, 동의어 등의 객체나 해당 사용자에게 부여된 권한 정보를 제공합니다.

[예제] user_table로 사용자가 소유한 테이블에 대한 정보를 조회하여 봅시다.

예 제	SELECT table_name FROM user_tables ;

- USER_SEQUENCES : 사용자가 소유한 시퀀스의 정보
- USER_INDEXES : 사용자가 소유한 인덱스 정보를 조회
- USER_VIEW : 사용자가 소유한 뷰 정보를 조회할 수 있는 데이터 사전

6-2. 데이터 사전 – ALL_ 데이터 사전

- 접두어로 ALL_이 붙은 데이터 사전은 전체 사용자와 관련된 뷰로서 사용자가 접근할 수 있는 모든 객체에 대한 정보를 조회할 수 있습니다.

- 조회 중인 객체가 누구의 소유인지 확인하도록 하기 위해서 OWNER 컬럼을 제공합니다.

[예제] all_table로 테이블에 대한 정보를 조회하여 봅시다.

예 제	SELECT owner, table_name FROM all_tables ;

6-3. 데이터 사전 – DBA_ 데이터 사전

- 접두어로 DBA_가 붙은 데이터 사전은 시스템과 관련된 뷰로, DBA나 시스템 권한을 가진 사용자만 접근할 수 있습니다.

- 현재 사용자가 HR이라면 DBA_로 시작하는 데이터 사전을 조회할 권한이 없기 때문에 DBA 권한을 가진 SYSTEM 계정으로 접속해야 합니다.

[예제] dba_tables로 테이블에 대한 정보를 조회하여 봅시다.

예 제	SELECT owner, table_name FROM dba_tables ;