

오라클을 이용한 DB 구축 초급

강의 노트

테이블 생성과 관리

Lesson 1 DB 객체와 데이터 형

1. DB 객체
2. 객체이름 지정 규칙
3. 객체이름 변경 방법
4. 대표적인 데이터 형(Type)

1. DB 객체

(1) DB 객체란?

논리적인 데이터 저장 영역 구조로서, 스키마(Schema) 객체라고도 칭한다.(2) DB 객체의 종류
대표적인 DB 객체로는 다음과 같은 것들이 있다.

객체	특성
테이블 (Table)	행과 열로 구성된 기본적인 데이터 저장 영역의 기본 단위
뷰 (View)	하나 이상의 테이블로부터 획득한 논리적인 관점에서 의 데이터의 부분 집합
시퀀스 (Sequence)	자동 생성되는 일련 번호로, 대개 기본 키 값 생성에 사용
인덱스 (Index)	SQL 문의 실행 속도를 향상시키기 위해 생성하는 색 인
동의어 (Synonym)	DB 객체에 부여한 별칭

2. 객체이름 지정 규칙

- 객체 생성 시에 이름을 부여할 때는 반드시 다음 규칙을 따라야 한다. 1) 첫 문자 : 반드시 영문자로 시작. (단, 한글 이름은 예외)
- 2) 길이 : 이름의 길이는 1 ~ 30 자 사이
- 3) 사용문자 : A-Z, a-z, 0-9, _(underscore), \$, # (비권장), 한글 4) 이름 중복 여부 : 동일한 사용자가 소유한 객체이름은 중복될 수 없음 5) 예약어 사용 여부 : 객체 이름으로 예약어를 사용할 수 없음
- 6) 대소문자 구분 여부 : 대소문자를 구분하지 않음 (자동으로 대문자로 변경됨.)
예) EMP와 Emp, emp는 같은 이름으로 간주
- 7) 대소문자 구분 방법 : 대소문자를 구분하고 싶으면, 이중 인용부호로 묶어서 지정
예) "My_Emp"와 "my_emp"는 같지 않음

3. 객체이름 변경 방법

객체이름을 변경할 때는 다음과 같은 원칙과 형식을 준수해야 한다. (1) 변경 원칙

- 1) 인덱스를 제외한 테이블, 뷰, 시퀀스, 동의어 이름의 변경이 가능
- 2) 객체의 소유자만 변경 가능

(2) 형식

old_name(원래 이름)을 new_name(새 이름)으로 변경한다.

```
RENAME old_name TO new_name ;
```

4. 대표적인 데이터 형(Type)

테이블 생성 시에 중요한 정보가 되는 대표적인 데이터 형의 종류를 살펴보면 다음과 같다.

데이터 형	특징
CHAR	고정길이 문자를 기억함 : 최대 2000 바이트
VARCHAR2	가변길이 문자를 기억함 : 최대 4000 바이트
VARCHAR	VARCHAR2와 같음 : 추후 다른 목적으로 사용될 예정
DATE	날짜와 시간을 기억함 : 기본형식은 YY/MM/DD
NUMBER(n)	길이가 n인 숫자를 기억함
LONG	가변길이 문자를 기억함 : 최대 2G 바이트

Lesson 2 테이블 생성과 데이터 사전

1. 테이블 생성
2. 데이터 사전(Data Dictionary)
3. 서브쿼리를 이용한 테이블 생성

1. 테이블 생성

CREATE TABLE 명령을 이용해서 새로운 테이블을 생성할 수 있다.(1) 테이블 생성 원칙

- 1) 데이터 정의어(DDL) 가운데 하나인 CREATE TABLE 명령으로 생성
- 2) 사용자가 테이블 생성 권한을 가진 경우에만 테이블 생성 가능
(이 권한은 DB 관리자가 사용자에게 부여한다.)
- 3) 테이블 생성 후에 자동으로 커밋(COMMIT)됨

(2) CREATE TABLE 명령의 형식

- 테이블 이름 뒤에 테이블을 구성하는 열 이름과 데이터 형, 길이 등을 괄호 안에 기술한다.
- DEFAULT 표현식 : INSERT 명령으로 행을 삽입할 때, 열의 값을 생략할 경우 열에 할당할 디폴트 값을 명시한다.
- 데이터 형 : 열의 데이터 형과 길이를 명시한다.

```
CREATE TABLE 테이블 이름
    (열 이름 데이터 형 [DEFAULT 표현식]
    [, 열 이름 데이터 형, ...]);
```

(3) CREATE TABLE 명령의 DEFAULT 옵션의 역할

→ 행 삽입 시에 열의 값을 생략하면 지정할 디폴트 값을 명시한다.

예) employee 테이블에 새로운 열이 삽입될 때, 입사일이 생략되면 오늘 날짜를 삽입하도록 테이블을 생성한다.

```
CREATE TABLE employee  
(hiredate DATE DEFAULT SYSDATE, ...)
```

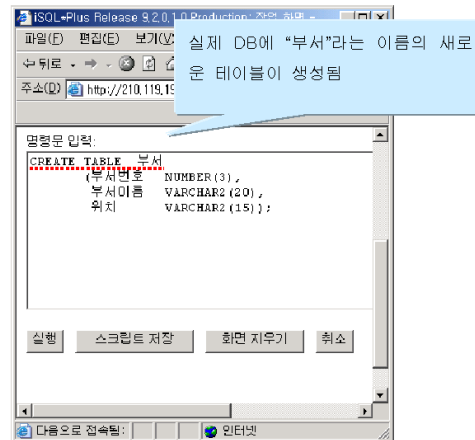
→ 디폴트 값을 지정함으로써, 열에 NULL 값이 입력되는 것을 방지할 수 있다.

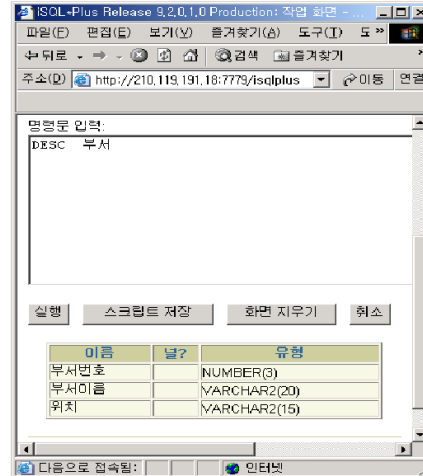
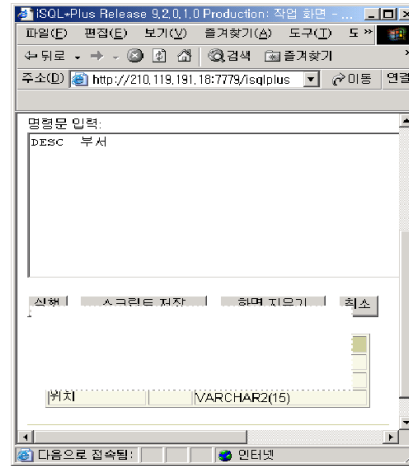
(4) 디폴트 값의 특징

- 1) 가능한 디폴트 값 : 문자열, 표현식, SQL 함수
- 2) 불가능한 디폴트 값 : 다른 열의 이름이나 의사열(NEXTVAL, CURRVAL 등)
- 3) 디폴트 값의 데이터 형은 반드시 열의 데이터 형과 일치해야 함

(5) 테이블 생성 예

→ 부서번호와 부서이름, 위치라는 세 개의 열을 포함하는 “부서”라는 테이블을 생성한다.





2. 데이터 사전(Data Dictionary)

(1) 데이터 사전

데이터 사전이란? :

- DB 관리를 위해 자동으로 생성되는 읽기 전용 테이블의 집합
- 시스템 내에 있는 모든 객체들에 대한 정의와 명세에 관한 정보를 수록하고 있으므로, DB 관리에 있어서 매우 중요한 역할을 함
- 사용자와 시스템이 공동으로 사용함

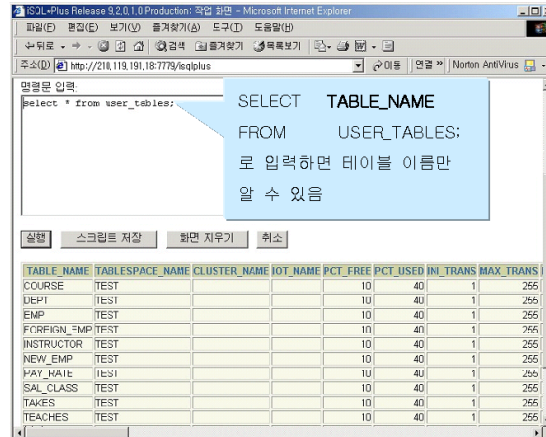
(2) 사용자 객체와 관련된 데이터 사전 테이블

- 데이터를 포함하는 테이블을 포함해서 사용자 객체와 관련된 유용한 정보를 제공하는 데이터 사전 테이블에는 다음과 같은 것들이 있다.

데이터 사전의 테이블 이름	용도
USER_TABLES	사용자가 소유한 모든 테이블의 이름을 알 수 있음
USER_OBJECTS	사용자가 소유한 서로 다른 객체 형을 알 수 있음
USER_CATALOG(또는 CAT)	사용자가 소유한 테이블, 뷰, 동의어 및 시퀀스를 알 수 있음

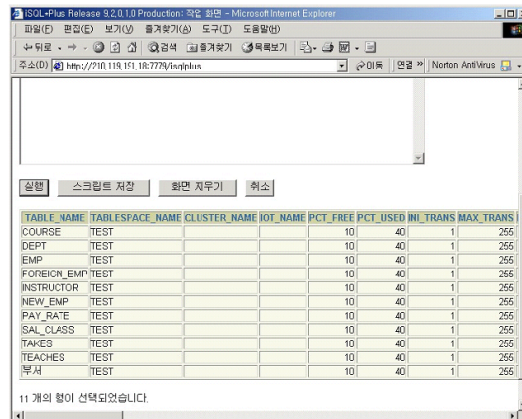
(3) 데이터 사전의 테이블 확인 예

→ 여러분이 소유한 테이블의 이름과 서로 다른 객체 형 및 테이블, 뷰, 동의어 및 시퀀스를 확인한다.



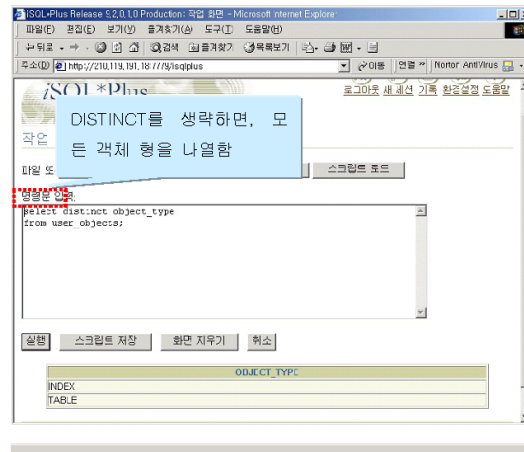
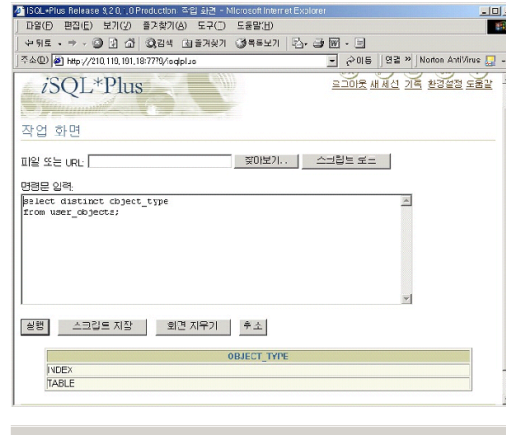
The screenshot shows the SQL*Plus interface with a query entered in the command window: `select * from user_tables;`. A blue callout box highlights the query, stating: "SELECT TABLE_NAME FROM USER_TABLES; 로 입력하면 테이블 이름만 알 수 있음" (If you enter SELECT TABLE_NAME FROM USER_TABLES;, you can only know the table names). Below the command window, the results of the query are displayed in a table format.

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS
COURSE	TEST			10	40	1	255
DEPT	TEST			10	40	1	255
EMP	TEST			10	40	1	255
FOREIGN_EMP	TEST			10	40	1	255
INSTRUCTOR	TEST			10	40	1	255
NEW_EMP	TEST			10	40	1	255
PAY_RATE	TEST			10	40	1	255
SAL_CLASS	TEST			10	40	1	255
TAKES	TEST			10	40	1	255
TEACHES	TEST			10	40	1	255



This screenshot shows the same SQL*Plus interface with the same query results. At the bottom of the window, a message states: "11 개의 행이 선택되었습니다." (11 rows selected).

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS
COURSE	TEST			10	40	1	255
DEPT	TEST			10	40	1	255
EMP	TEST			10	40	1	255
FOREIGN_EMP	TEST			10	40	1	255
INSTRUCTOR	TEST			10	40	1	255
NEW_EMP	TEST			10	40	1	255
PAY_RATE	TEST			10	40	1	255
SAL_CLASS	TEST			10	40	1	255
TAKES	TEST			10	40	1	255
TEACHES	TEST			10	40	1	255
부서	TEST			10	40	1	255





3. 서브쿼리를 이용한 테이블 생성

테이블 생성과 동시에 행을 삽입하기 위해서 서브쿼리(Subquery)를 이용할 수 있다.

(1) 서브쿼리를 포함하는 CREATE TABLE 명령의 형식

→ 예약어인 AS 뒤에 새로운 테이블에 삽입할 행(들)을 반환하는 SELECT 문장인 서브쿼리를 괄호 없이 기술한다.

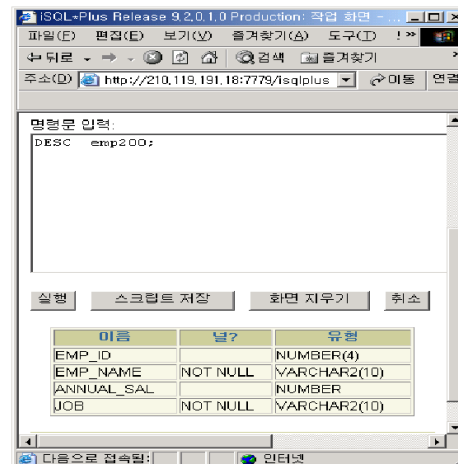
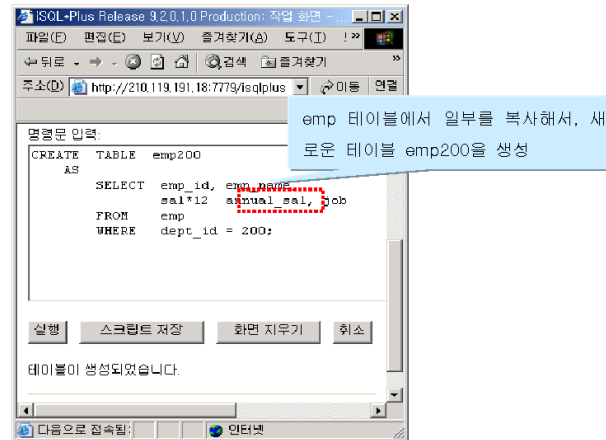
```
CREATE TABLE 테이블 이름 [열 이름1 (, 열 이름2, ...)]  
AS 서브쿼리 ;
```

(2) 테이블 생성 시 주의사항

- 1) 서브쿼리는 괄호로 묶지 않음
- 2) 열 이름 목록을 기술한 경우, 서브쿼리가 반환하는 열의 순서가 CREATE TABLE 절에 명시된 열의 순서와 반드시 일치해야 함
- 3) 열 이름 목록을 생략하면, 테이블이 생성될 때의 열의 수 및 순서가 서브쿼리가 반환한 열의 수 및 순서와 동일해야 함

(3) 테이블 생성과 동시에 행을 삽입하는 예제

→ emp 테이블에서 200 번 부서의 직원 데이터 가운데 직원번호와 직원이름, 업무를 복사하고, 급여에 12 를 곱해서 연봉(ANNUAL_SAL)으로 저장하는 'emp200'이라는 테이블을 생성한다.



Lesson 3 테이블 구조 변경과 테이블 삭제

1. 테이블 구조 변경
2. 테이블 삭제

1. 테이블 구조 변경

ALTER TABLE 명령을 이용해서 기존 테이블의 구조를 변경할 수 있다.(1) 변경 허용 범위

- 새로운 열을 추가한다.
- 기존 열을 변경한다.
- 열에 대한 디폴트 값을 새로 정의한다.
- 테이블에서 열의 삭제는 불가능하다.

(2) ALTER TABLE 명령의 형식

- 열 추가 형식 ADD 절에 추가할 하나 이상의 열 정보를 기술한다.ALTER TABLE 테이블 이름

```
ADD (열 이름1 데이터형 [DEFAULT 표현식]
    [, 열 이름2 데이터형] ... ) ;
```

- 열 변경 형식

MODIFY 절에 변경할 하나 이상의 열 정보를 기술한다.ALTER TABLE 테이블 이름

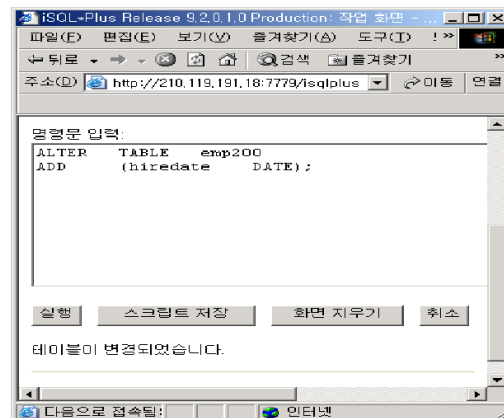
```
MODIFY (열 이름1 데이터형 [DEFAULT 표현식]
    [, 열 이름2 데이터형] ... ) ;
```

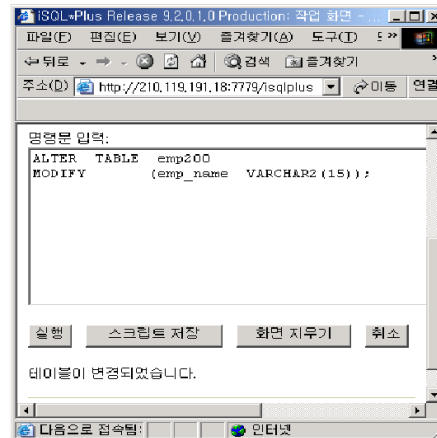
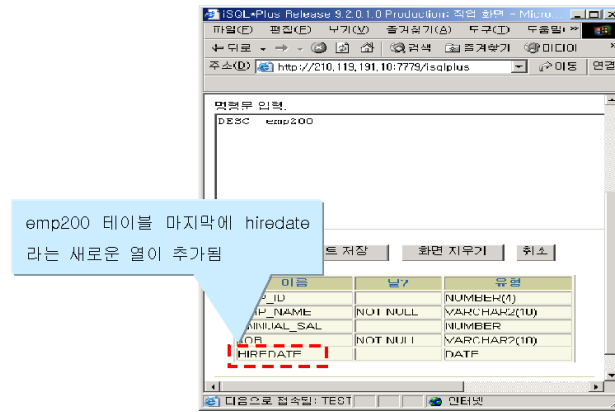
(3) 열 추가/변경 방법

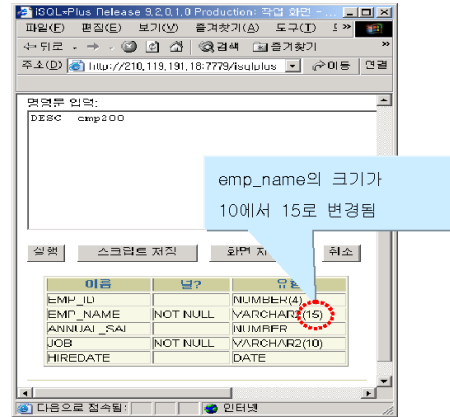
열 추가 방법

- ① **ALTER TABLE ~ ADD** 명령으로 열을 추가함
- ② 열의 위치는 명시할 수 없으며, 새로 추가된 열은 테이블의 마지막 열이 됨
열 변경 방법
- ① **ALTER TABLE ~ MODIFY** 명령으로 기존 열을 변경함
- ② 열의 데이터 형과 크기, 디폴트 값을 변경할 수 있음
- ③ 크기 확장은 항상 가능
- ④ 크기 축소나 데이터 형의 변경은 데이터 값이 없는 경우에만 가능
- ⑤ 디폴트 값을 변경한 경우, 변경된 이후에 삽입되는 데이터에만 적용됨

(4) 열 추가/변경 예







2. 테이블 삭제

DROP TABLE 명령이나 TRUNCATE TABLE 명령으로 테이블을 삭제할 수 있다.

(1) DROP TABLE 명령에 의한 삭제

→ 불필요한 테이블과 함께 테이블에 저장된 데이터도 한꺼번에 삭제한다. 1) DROP TABLE

명령의 형식

→ DROP TABLE 뒤에 삭제할 테이블의 이름을 기술한다.

```
DROP TABLE table_name ;
```

2) 테이블 삭제의 원칙

- 테이블 생성자나 권한이 부여된 사람만 삭제 가능
- 테이블 구조와 함께 테이블에 저장된 모든 데이터가 삭제됨
- 모든 미결정된 트랜잭션이 커밋(COMMIT)됨
- 모든 인덱스가 함께 삭제됨
- 삭제된 후에는 복구(RELLBACK)가 불가능

(2) TRUNCATE TABLE 명령에 의한 삭제

→ 테이블 구조는 남겨두고, 테이블에 저장된 데이터만 삭제할 수 있다.

1) TRUNCATE TABLE 명령의 형식

→ TRUNCATE TABLE 뒤에 데이터만 삭제할 테이블의 이름을 기술한다.

```
TRUNCATE TABLE table_name ;
```

2) 테이블 삭제의 원칙

- 테이블 생성자나 권한이 부여된 사람만 삭제 가능
- 테이블 구조는 남겨두고, 모든 행만 삭제함
- 테이블이 사용한 기억공간을 해제함
- 모든 미결정된 트랜잭션이 자동 커밋됨
- 삭제한 행의 복구가 불가능

◆데이터 조작어인 DELETE 명령으로 데이터를 삭제하면, 기억공간을 해제하지 않게 되며,
따라서 데이터 복구(ROLLBACK)가 가능하다.

◆ 학습 정리

DB 객체의 종류

→ DB 객체란 논리적인 데이터 저장 영역 구조로, 다음과 같은 종류와 특성이 있다.

종류	특성
테이블(Table)	행과 열로 구성된 기본적인 데이터 저장 영역의 기본 단위
뷰(View)	하나 이상의 테이블로부터 획득한 논리적인 관점에서의 데이터의 부분 집합
시퀀스(Sequence)	자동 생성되는 일련 번호로, 대개 기본 키 값 생성에 사용
인덱스(Index)	SQL 문의 실행 속도를 향상시키기 위해 생성하는 색인
동의어(Synonym)	DB 객체에 부여한 별칭

테이블 생성

→ 다음과 같은 형식의 CREATE TABLE 명령을 이용해서 새로운 테이블을 생성할 수 있다.

```
CREATE TABLE 테이블이름  
    (열 이름 데이터형 [DEFAULT 표현식]  
    [, 열 이름 데이터형, ...]);
```

- ① 데이터 정의어(DDL) 가운데 하나인 CREATE TABLE 명령으로 생성한다.
- ② 사용자가 테이블 생성 권한을 가진 경우에만 테이블을 생성할 수 있다.
(이 권한은 DB 관리자가 사용자에게 부여한다.)
- ③ 테이블 생성 후에 자동으로 커밋(COMMIT)된다.

-서브쿼리를 이용한 테이블 생성

→ 테이블 생성과 동시에 데이터 행을 삽입하기 위해서, CREATE TABLE 명령의 AS 절에 서브쿼리를 기술한다.

```
CREATE TABLE 테이블이름 [열 이름1(, 열 이름2, ...)]
AS
서브쿼리;
```

-테이블 구조 변경

→ ALTER TABLE 명령을 이용해서 기존 테이블에 열을 추가하거나 변경해서 구조를 변경할 수 있으며, 열 삭제는 허용되지 않는다.

열 추가 형식	<p>→ ADD 절에 추가할 하나 이상의 열 정보를 기술한다.</p> <pre>ALTER TABLE 테이블이름 ADD (열 이름1 데이터형 [DEFAULT 표현식] [, 열 이름2 데이터형] ...);</pre>
열 변경 형식	<p>→ MODIFY 절에 변경할 하나 이상의 열 정보를 기술한다.</p> <pre>ALTER TABLE 테이블이름 MODIFY (열 이름1 데이터형 [DEFAULT 표현식] [, 열 이름2 데이터형] ...);</pre>

-테이블 삭제

→ DROP TABLE 명령이나 TRUNCATE TABLE 명령으로 테이블을 삭제할 수 있다.

(1) DROP TABLE 명령에 의한 삭제

불필요한 테이블과 함께 테이블에 저장된 데이터도 한꺼번에 삭제한다

```
DROP TABLE table_name ;
```

(2) TRUNCATE TABLE 명령에 의한 삭제

테이블 구조는 남겨두고 테이블에 저장된 데이터만 삭제할 수 있다

```
TRUNCATE TABLE table_name ;
```