

Chap13. 객체 종류

13-1 데이터베이스를 위한 데이터를 저장한 데이터사전

“ 데이터 사전이란?

“ 데이터베이스를 구성하고 운영하는데 필요한 정보

“ 데이터 사전 뷰

접두어	설명
USER_XXXX	현재 데이터베이스에 접속한 사용자가 소유한 객체 정보
ALL_XXXX	현재 데이터베이스에 접속한 사용자가 소유한 객체 또는 다른 사용자가 소유한 객체 중 사용 허가를 받은 객체, 즉 사용 가능한 모든 객체 정보
DBA_XXXX	데이터베이스 관리를 위한 정보(데이터베이스 관리 권한을 가진 SYSTEM, SYS 사용자만 열람 가능)
V\$_XXXX	데이터베이스 성능 관련 정보(X\$_XXXX 테이블의 뷰)

13-1 데이터베이스를 위한 데이터를 저장한 데이터사전

“ **USER_** 접두어를 가진 데이터 사전

“ 접속한 사용자가 소유한 객체 정보

“ **ALL_** 접두어를 가진 데이터 사전

“ 접속한 사용자가 소유하거나, 사용을 허가받은 객체 정보

“ **DBA_** 접두어를 가진 데이터 사전

“ 데이터베이스 관리 권한을 가진 사용자가 조회가능

13-2 더 빠른 검색을 위한 인덱스

“ 인덱스

“ 데이터 검색 성능 향상을 위해 테이블 열에 사용하는 객체

“ USER_INDEXES

“ USER_IND_COLUMNS



13-2 더 빠른 검색을 위한 인덱스

“ 인덱스 생성

```
CREATE INDEX 인덱스 이름  
ON 테이블 이름(열 이름1 ASC or DESC,  
열 이름2 ASC or DESC,  
... );
```

“ 인덱스 삭제

“ DROP

13-3 테이블처럼 사용하는 뷰

“ 뷰란?

“ SELECT문을 저장한 객체

“ 뷰의 사용 목적

“ 편의성 : SELECT문의 복잡도 완화

“ 보안성 : 테이블의 일부 데이터만 노출

13-3 테이블처럼 사용하는 뷰

“ 뷰의 생성

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW 뷰 이름 (열 이름1, 열 이름2, ...)  
AS (저장할SELECT문)  
[WITH CHECK OPTION [CONSTRAINT 제약 조건]]  
[WITH READ ONLY [CONSTRAINT 제약 조건]];
```

Diagram illustrating the components of the CREATE VIEW statement:

- 1: [OR REPLACE]
- 2: [FORCE | NOFORCE]
- 3: VIEW
- 4: 뷰 이름
- 5: (열 이름1, 열 이름2, ...)
- 6: AS (저장할SELECT문)
- 7: [WITH CHECK OPTION [CONSTRAINT 제약 조건]]
- 8: [WITH READ ONLY [CONSTRAINT 제약 조건]];



13-3 테이블처럼 사용하는 뷰

- “ USER_VIEWS

- “ 뷰 삭제

 - “ DROP

- “ 인라인 뷰를 사용한 TOP-N SQL문

13-4 규칙에 따라 순번을 생성하는 시퀀스

“ 시퀀스란?

“ 번호 생성기

“ 시퀀스 생성

CREATE SEQUENCE 시퀀스 이름 —①

[INCREMENT BY n] —②

[START WITH n] —③

[MAXVALUE n | NOMAXVALUE] —④

[MINVALUE n | NOMINVALUE] —⑤

[CYCLE | NOCYCLE] —⑥

[CACHE n | NOCACHE] —⑦

13-4 규칙에 따라 순번을 생성하는 시퀀스

“ 시퀀스 사용

“ CURRVAL

“ 시퀀스 마지막 생성 번호

“ NEXTVAL

“ 다음 번호 생성

“ 시퀀스 수정 : ALTER

“ 시퀀스 삭제 : DROP



13-5 공식 별칭을 지정하는 동의어

“ 동의어란?

“ 객체 이름 대신 사용할 수 있는 다른 이름



The diagram shows the SQL syntax for creating a synonym. It consists of two lines of text. The first line is `CREATE [PUBLIC] SYNONYM 동의어 이름`. The second line is `FOR [사용자.][객체 이름];`. Red lines and circles with numbers 1 through 4 indicate the components: 1 points to `[PUBLIC]`, 2 points to `동의어 이름`, 3 points to `[사용자.]`, and 4 points to `[객체 이름]`.

```
CREATE [PUBLIC] SYNONYM 동의어 이름
FOR [사용자.][객체 이름];
```

“ 동의어 삭제 : DROP

Chap14. 제약 조건

14-1 제약 조건 종류

“ 제약 조건이란?

“ 테이블 열에 저장될 데이터의 특성,조건을 지정

종류	설명
NOT NULL	지정한 열에 NULL을 허용하지 않습니다. NULL을 제외한 데이터의 중복은 허용됩니다.
UNIQUE	지정한 열이 유일한 값을 가져야 합니다. 즉 중복될 수 없습니다. 단 NULL은 값의 중복에서 제외됩니다.
PRIMARY KEY	지정한 열이 유일한 값이면서 NULL을 허용하지 않습니다. PRIMARY KEY는 테이블에 하나만 지정 가능합니다.
FOREIGN KEY	다른 테이블의 열을 참조하여 존재하는 값만 입력할 수 있습니다.
CHECK	설정한 조건식을 만족하는 데이터만 입력 가능합니다.



14-2 빈값을 허락하지 않는 NOT NULL

“ NOT NULL

“ 지정된 열에 **NULL** 저장 불가

“ NULL외 **데이터의 중복**은 허용

“ 제약 조건 확인

“ USER CONSTRAINTS

열 이름	설명
OWNER	제약 조건 소유 계정
CONSTRAINT_NAME	제약 조건 이름(직접 지정하지 않을 경우 오라클이 자동으로 지정함)
CONSTRAINT_TYPE	제약 조건 종류 C : CHECK, NOT NULL U : UNIQUE P : PRIMARY KEY R : FOREIGN KEY
TABLE_NAME	제약 조건을 지정한 테이블 이름



14-2 빈값을 허락하지 않는 NOT NULL

- “ 제약 조건 지정

- “ 테이블을 생성하며 제약조건 지정

- “ 제약 조건 이름 직접 지정

- “ 이미 생성한 테이블에 제약 조건 지정

- “ 제약 조건 삭제

- “ DROP



14-3 중복되지 않는 값 UNIQUE

- “ UNIQUE

- “ 지정된 열에 중복 데이터 저장 불가
- “ NULL 저장은 가능

- “ 제약 조건 확인 : `USER_CONSTRAINTS`



14-3 중복되지 않는 값 UNIQUE

“ 제약 조건 지정

“ 테이블을 생성하며 제약조건 지정

“ 제약 조건 이름 직접 지정

“ 이미 생성한 테이블에 제약 조건 지정

“ 제약 조건 삭제

“ DROP



14-4 유일하게 하나만 있는 값 PRIMARY KEY

“ PRIMARY KEY

“ 지정된 열에 중복 데이터 저장 불가

“ NULL 저장 불가

“ 자동으로 인덱스 생성



14-4 유일하게 하나만 있는 값 PRIMARY KEY

“ 제약 조건 지정

“ 테이블을 생성하며 제약 조건 지정하기

“ 테이블을 생성하며 제약 조건 이름 직접 지정하기



14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

“ FOREIGN KEY

- “ 다른 테이블의 **PRIMARY KEY**를 참조
- “ 참조하고 있는 키의 데이터와 **NULL**만 저장 가능



14-5 다른 테이블과 관계를 맺는 FOREIGN KEY

- “ 제약 조건 지정

- “ 테이블을 생성하며 제약 조건 지정하기
- “ 테이블을 생성하며 제약 조건 이름 직접 지정하기

- “ FOREIGN KEY로 참조 행 데이터 삭제하기

- “ ON DELETE CASCADE
- “ ON DELETE SET NULL



14-6 데이터 형태와 범위를 정하는 CHECK

“ CHECK

- “ 열에 저장할 수 있는 값의 범위 또는 패턴을 정의
- “ 조건식을 지정



14-7 기본값을 정하는 DEFAULT

“ DEFAULT

- “ 저장값이 없을 경우 기본 값을 지정



Chap15. 사용자, 권한, 롤 관리

15-1 사용자 관리

“ 사용자란?

- “ 데이터베이스에 접속하여 데이터를 사용/관리하는 계정
- “ 업무의 분할과 효율, 보안을 고려하여 생성

“ 데이터베이스 스키마란?

- “ 데이터를 저장 및 관리하기 위해 정의한 데이터베이스 구조의 범위를 분류



15-1 사용자 관리

“ 사용자 생성

CREATE USER 사용자 이름(필수)

IDENTIFIED BY 패스워드(필수)

DEFAULT TABLESPACE 테이블 스페이스 이름(선택)

TEMPORARY TABLESPACE 테이블 스페이스(그룹) 이름(선택)

QUOTA 테이블 스페이스크기 **ON** 테이블 스페이스 이름(선택)

PROFILE 프로파일 이름(선택)

PASSWORD EXPIRE(선택)

ACCOUNT [LOCK/UNLOCK](선택);



15-1 사용자 관리

“ 사용자 정보 조회

“ ..._USERS

“ 오라클 사용자의 변경과 삭제

“ 변경 :ALTER

“ 삭제 :DROP

“ CASCADE



15-2 권한 관리

“ 권한

- “ 접속 사용자에게 따라 접근할 수 있는 데이터 영역을 지정
- “ 시스템 권한, 객체 권한

“ 시스템 권한

- “ 사용자 생성, 정보 수정, 삭제, 데이터베이스 접근
- “ 여러 자원과 객체 생성 및 관리 등의 권한을 포함



15-2 권한 관리

“ 시스템 권한 부여

```
GRANT [시스템 권한] TO [사용자 이름/롤(Role)이름/PUBLIC]  
[WITH ADMIN OPTION];
```

Diagram illustrating the syntax for granting system privileges:

- 1: [시스템 권한] (System Privilege)
- 2: [사용자 이름/롤(Role)이름/PUBLIC] (User Name/Role Name/PUBLIC)
- 3: [WITH ADMIN OPTION]; (Optional clause)

“ 시스템 권한 취소

```
REVOKE [시스템 권한] FROM [사용자 이름/롤(Role)이름/PUBLIC];
```

15-2 권한 관리

“ 객체 권한이란?

“ 특정 사용자가 생성한 객체와 관련된 권한

“ SCOTT이 아닌 사용자가 SCOTT 테이블에 INSERT



15-2 권한 관리

“ 객체 권한 부여

```
GRANT [객체 권한/ALL PRIVILEGES] — 1  
ON [스키마.객체 이름] — 2  
TO [사용자 이름/롤(Role)이름/PUBLIC] — 3  
[WITH GRANT OPTION]; — 4
```

“ 객체 권한 취소

```
REVOKE [객체 권한/ALL PRIVILEGES](필수)  
ON [스키마.객체 이름](필수)  
FROM [사용자 이름/롤(Role) 이름/PUBLIC](필수)  
[CASCADE CONSTRAINTS/FORCE](선택);
```



15-3 롤 관리

“ 롤이란?

“ 여러 종류의 권한을 묶어 놓은 그룹

“ 사전 정의된 롤

“ CONNECT

“ RESOURCE

“ DBA



15-3 롤 관리

“ 사용자 정의 롤

“ 필요에 의해 직접 권한을 포함시켜 생성한 롤

- ① CREATE ROLE문으로 롤을 생성합니다.
- ② GRANT 명령어로 생성한 롤에 권한을 포함시킵니다.
- ③ GRANT 명령어로 권한이 포함된 롤을 특정 사용자에게 부여합니다.
- ④ REVOKE 명령어로 롤을 취소시킵니다.

Chap16. PL/SQL 기초

16-1 PL/SQL 구조

“ 블록

“ PL/SQL 프로그램의 기본 단위

구성 키워드	필수/선택	설명
DECLARE(선언부)	선택	실행에 사용될 변수·상수·커서 등을 선언
BEGIN(실행부)	필수	조건문·반복문·SELECT·DML·함수 등을 정의
EXCEPTION(예외 처리부)	선택	PL/SQL 실행 도중 발생하는 오류(예외 상황)를 해결하는 문장 기술

DECLARE

[실행에 필요한 여러 요소 선언];

BEGIN

[작업을 위해 실제 실행하는 명령어];

EXCEPTION

[PL/SQL수행 도중 발생하는 오류 처리];

END;

16-1 PL/SQL 구조

“ 주석

“ 실행되지 않는 문장

“ 한줄 주석 : '--'

“ 여러 줄 주석 : '/* */'

“ SQL문에서도 사용 가능



16-2 변수와 상수

“ 변수

- “ 데이터를 일시적으로 저장
- “ 저장되는 값의 변경 가능
- “ 이름, 저장할 자료형을 지정

변수 이름 자료형 := 값 또는 값이 도출되는 여러 표현식;

1

2

3

4

16-2 변수와 상수

“ 상수

- “ 한번 저장한 값이 프로그램 종료시까지 유지
- “ **CONSTANT** 키워드 사용

변수 이름 **CONSTANT** 자료형 := 값 또는 값을 도출하는 여러 표현식;

1

2

3

4

5

16-2 변수와 상수

“ 변수의 기본값 지정

변수 이름 자료형 **DEFAULT** 값 또는 값 도출되는 여러 표현식;

1

2

3

4

“ 변수에 **NULL** 값 저장 막기

변수 이름 자료형 **NOT NULL** := 또는 **DEFAULT** 값 또는 값이 도출되는 여러 표현식;

1

2

3

4

5

16-2 변수와 상수

“ 변수의 자료형

“ 스칼라형 : 내부 구성 요소가 없는 단일 값

분류	자료형	설명
숫자	NUMBER	소수점을 포함할 수 있는 최대 38자리 숫자 데이터
문자열	CHAR	최대 32,767바이트 고정 길이 문자열 데이터
	VARCHAR2	최대 32,767바이트 가변 길이 문자열 데이터
날짜	DATE	기원전 4712년 1월 1일부터 서기 9999년 12월 31일까지 날짜 데이터
논리 데이터	BOOLEAN	PL/SQL에서만 사용할 수 있는 논리 자료형으로 true, false, NULL을 포함



16-2 변수와 상수

“ 변수의 자료형

“ 참조형 : 특정 테이블의 열, 행 구조를 참조

“ %TYPE : 열 참조

변수 이름 테이블이름.열이름%TYPE;

1

2

3

“ 변수 이름 테이블 이름%ROWTYPE;

1

2

3

16-2 변수와 상수

“ 변수의 자료형

“ 복합형

분류	자료형	설명
컬렉션	TABLE	한 가지 자료형의 데이터를 여러 개 저장(테이블의 열과 유사)
레코드	RECORD	여러 종류 자료형의 데이터를 저장(테이블의 행과 유사)

“ LOB형

“ Large Object



16-3 조건 제어문

“ IF 조건문

종류	설명
IF-THEN	특정 조건을 만족하는 경우 작업 수행
IF-THEN-ELSE	특정 조건을 만족하는 경우와 반대 경우에 각각 지정한 작업 수행
IF-THEN-ELSIF	여러 조건에 따라 각각 지정한 작업 수행

“ IF-THEN

```
IF 조건식 THEN —①  
    수행할 명령어; —②  
END IF; —③
```

16-3 조건 제어문

“ IF-THEN-ELSE

```
IF 조건식 THEN —①  
    수행할 명령어; —②  
ELSE  
    수행할 명령어; —③  
END IF; —④
```

“ IF-THEN-ELSIF

```
IF 조건식 THEN —①  
    수행할 명령어; —②  
ELSIF 조건식 —③  
    수행할 명령어; —④  
ELSIF —⑤  
    수행할 명령어; —⑥  
...  
ELSE —⑦  
    수행할 명령어; —⑧  
END IF; —⑨
```

16-3 조건 제어문

“ CASE 조건문

종류	설명
단순 CASE문	비교 기준이 되는 조건의 값이 여러 가지일 때 해당 값만 명시하여 작업 수행
검색 CASE문	특정한 비교 기준 없이 여러 조건식을 나열하여 조건식에 맞는 작업 수행

단순 CASE

```
CASE 비교 기준 —①
  WHEN 값1 THEN —②
    수행할 명령어; —③
  WHEN 값2 THEN —④
    수행할 명령어; —⑤
  ...
  ELSE —⑥
    수행할 명령어; —⑦
END CASE; —⑧
```

검색 CASE

```
CASE
  WHEN 조건식1 THEN —①
    수행할 명령어; —②
  WHEN 조건식2 THEN —③
    수행할 명령어; —④
  ...
  ELSE —⑤
    수행할 명령어; —⑥
END CASE;
```

16-4 반복 제어문

“ 반복문

종류	설명
기본 LOOP	기본 반복문
WHILE LOOP	특정 조건식의 결과를 통해 반복 수행
FOR LOOP	반복 횟수를 정하여 반복 수행
Cursor FOR LOOP	커서를 활용한 반복 수행

“ 반복 수행 제어

종류	설명
EXIT	수행 중인 반복 종료
EXIT-WHEN	반복 종료를 위한 조건식을 지정하고 만족하면 반복 종료
CONTINUE	수행 중인 반복의 현재 주기를 건너뛴
CONTINUE-WHEN	특정 조건식을 지정하고 조건식을 만족하면 현재 반복 주기를 건너뛴



16-4 반복 제어문

“ 기본 LOOP

LOOP

반복 수행 작업;

END LOOP;

“ EXIT : 즉시 반복 종료

“ EXITWHEN : 종료 조건식 사용

“ WHILE LOOP

WHILE 조건식 LOOP

반복 수행 작업;

END LOOP;



16-4 반복 제어문

“ FOR LOOP

```
FOR i IN 시작 값 .. 종료 값 LOOP  
    반복 수행 작업;  
END LOOP;
```

“ CONTINUE

“ 즉시 다음 반복 주기로 이동

“ CONTINUE-WHEN

“ 특정 조건식 만족 여부에 따라 다음 반복 주기로 이동

