Chap19. 저장 서브프로그램

19-1 저장 서브프로그램

- 저장 서브프로그램이란?
  - 이름을 지정하여 저장해두는 PL/SQL 프로그램

	익명 블록	저장 서브프로그램
이름	이름 없음	이름 지정
오라클 저장	저장할 수 없음	저장함
컴파일	실행할 때마다 컴파일	저장할 때 한 번 컴파일
공유	공유할 수 없음	공유하여 사용 가능
다른 응용 프로그램에서의 호출 가능 여부	호출할 수 없음	호출 가능

19-1 저장 서브프로그램

저장 서브프로그램이란?

서브프로그램	용도
저장 프로시저 (stored procedure)	일반적으로 특정 처리 작업 수행을 위한 서브프로그램으로 SQL문에서는 사용할 수 없습니다.
저장 함수 (stored function)	일반적으로 특정 연산을 거친 결과 값을 반환하는 서브프로그램으로 SQL문에서 사용할 수 있습니다.
패키지 (package)	저장 서브프로그램을 그룹화하는 데 사용합니다.
트리거 (trigger)	특정 상황(이벤트)이 발생할 때 자동으로 연달아 수행할 기능을 구현하는 데 사용합니다.

19-2 프로시저

파라미터를 사용하지 않는 프로시저

```
CREATE [OR REPLACE] PROCEDURE 프로시저 이름
IS | AS -3   1          2
   선언부
BEGIN
   실행부
EXCEPTION -4
   예외 처리부
END [프로시저 이름]; -5
```

19-2 프로시저

- 프로지서 내용 확인
  - USER_SOURCE

USER_SOURCE의 열	설명
NAME	서브프로그램(생성 객체) 이름
TYPE	서브프로그램 종류(PROCEDURE, FUNCTION 등)
LINE	서브프로그램에 작성한 줄 번호
TEXT	서브프로그램에 작성한 소스 코드

- 프로시저 삭제하기
  - DROP

19-2 프로시저

파라미터를 사용하는 프로시저

```
CREATE [OR REPLACE] PROCEDURE 프로시저 이름
                ①                    ②
[(파라미터 이름1 [modes] 자료형 [ := ¦ DEFAULT 기본값],③
  파라미터 이름2 [modes] 자료형 [ := ¦ DEFAULT 기본값],
  ...
  파라미터 이름N [modes] 자료형 [ := ¦ DEFAULT 기본값]
)]
IS ¦ AS ④
  선언부
BEGIN
  실행부
EXCEPTION ⑤
  예외 처리부
END [프로시저 이름]; ⑥
```

19-2 프로시저

파라미터 사용 모드

파라미터 모드	설명
IN	지정하지 않으면 기본값으로 프로시저를 호출할 때 값을 입력받습니다.
OUT	호출할 때 값을 반환합니다.
IN OUT	호출할 때 값을 입력받은 후 실행 결과 값을 반환합니다.

19-2 프로시저

- 프로시저 오류 정보 확인하기
  - SHOW ERRORS
  - USER_ERRORS

19-3 함수

프로시저와 함수

특징	프로시저	함수
실행	EXECUTE 명령어 또는 다른 PL/SQL 서브프로그램 내에서 호출하여 실행	변수를 사용한 EXECUTE 명령어 또는 다른 PL/SQL 서브프로그램에서 호출하여 실행하거나 SQL문에서 직접 실행 가능
파라미터 지정	필요에 따라 지정하지 않을 수도 있고 여러 개 지정할 수도 있으며 IN, OUT, IN OUT 세 가지 모드를 사용할 수 있음	프로시저와 같게 지정하지 않을 수도 있고 여러 개 지정할 수 있지만 IN 모드(또는 생략)만 사용
값의 반환	실행 후 값의 반환이 없을 수도 있고 OUT, IN OUT 모드의 파라미터 수에 따라 여러 개 값을 반환할 수 있음	반드시 하나의 값을 반환해야 하며 값의 반환은 프로시저와 달리 OUT, IN OUT 모드의 파라미터를 사용하는 것이 아니라 RETURN절과 RETURN문을 통해 반환

19-3 함수

함수 생성하기

```
CREATE [OR REPLACE] FUNCTION 함수 이름
[(파라미터 이름1 [IN] 자료형1, ❶
  파라미터 이름2 [IN] 자료형2,
  ...
  파라미터 이름N [IN] 자료형N
)]
RETURN 자료형 ❷
IS | AS
  선언부
BEGIN
  실행부
  RETURN (반환 값); ❸
EXCEPTION
  예외 처리부
END [함수 이름];
```

19-3 함수

- 함수 실행하기
  - PL/SQL로 함수 실행하기
  - SQL문에서 함수 실행하기

- 함수 삭제하기
  - DROP

19-4 패키지

- 패키지
  - 여러 서브프로그램을 통합·관리
  - 모듈성, 설계, 정보 은닉, 기능성 및 성능 향상
- 패키지 구조와 생성
  - 명세
  - 본문

19-4 패키지

패키지 명세

```
CREATE [OR REPLACE] PACKAGE 패키지 이름
IS ¦ AS
   서브프로그램을 포함한 다양한 객체 선언
END [패키지 이름];
```

패키지 본문

```
CREATE [OR REPLACE] PACKAGE BODY 패키지 이름
IS ¦ AS
   패키지 명세에서 선언한 서브프로그램을 포함한 여러 객체를 정의
   경우에 따라 패키지 명세에 존재하지 않는 객체 및 서브프로그램도 정의 가능
END [패키지 이름];
```

19-4 패키지

- 서브프로그램 오버로드

```
CREATE [OR REPLACE] PACKAGE 패키지 이름
IS | AS
   서브프로그램 종류 서브프로그램 이름(파라미터 정의);
   서브프로그램 종류 서브프로그램 이름(개수나 자료형, 순서가 다른 파라미터 정의);
END [패키지 이름];
```

- 패키지 사용
  - 패키지이름.객체이름
- 패키지 삭제
  - DROP

19-5 트리거

- 트리거란?
  - 데이터베이스 내 이벤트 발생시 자동 실행될 기능을 정의
  - 트리거 사용의 장점
    - 간편한 데이터 작업
    - 복잡한 데이터 규칙 정의
    - 보안성, 안정성을 위한 대처 활용
  - 트리거 지정 대상
    - p.508
  - 트리거의 구분
    - p.509

19-5 트리거

- DML 트리거

```
CREATE [OR REPLACE] TRIGGER 트리거 이름 ─1
BEFORE ¦ AFTER ─2
INSERT ¦ UPDATE ¦ DELETE ON 테이블 이름 ─3
REFERENCING OLD as old ¦ New as new ─4
FOR EACH ROW WHEN 조건식 ─5
FOLLOWS 트리거 이름2, 트리거 이름3 ... ─6
ENABLE ¦ DISABLE ─7

DECLARE
   선언부
BEGIN
   실행부
EXCEPTION
   예외 처리부
END;
```

19-5 트리거

- DML 트리거의 제작 및 사용
  - BEFORE
  - AFTER
- 트리거 정보 조회
  - USER_TRIGGERS
- 트리거 변경
  - ALTER
- 트리거 삭제
  - DROP