

Use of Variables



1. 변수 사용 및 처리

1-1. 변수 사용

- ☑ 데이터 입력 검증에 사용할 데이터를 하나 이상의 변수로 임시 저장 (**데이터 임시 저장**)
- ☑ 변수를 사용하여 데이터베이스를 액세스하지 않고 계산 및 데이터 조작 가능 (**저장된 값 조작**)
- ☑ 선언한 변수는 선언문 또는 기타 명령문에서 해당 변수를 참조하는 방식으로 응용 프로그램에서 반복 사용 (**재사용**)
- ☑ %TYPE 및 %ROWTYPE을 사용하는 경우 데이터베이스 열의 정의에 따라 변수 선언 (**유지 관리의 용이성**)

1-2. 변수 처리

- ☑ 선언 부분에서 변수 선언 및 초기화
- ☑ 실행 부분에서 변수에 새로운 값 할당
- ☑ 매개변수를 통해 값을 PL/SQL 블록에 전달
- ☑ 출력 변수를 통해 결과 확인



2. 변수 유형

2-1. PL/SQL 변수

- ☐ 스칼라(Scalar) 데이터 유형
 - 단일 값 보유
 - 기본 데이터 유형은 Oracle Server 테이블의 열 유형과 일치
 - PL/SQL은 부울(Boolean) 변수도 지원
- ☐ 조합(Composite) 데이터 유형
 - 조합 데이터 유형을 사용하면 PL/SQL 블록에서 필드 그룹을 정의하고 조작 가능
- ☐ 참조(Reference) 데이터 유형
 - 다른 프로그램 항목을 지정하는 값(포인트) 보유
- ☐ LOB (대형객체) 데이터 유형
 - 한 행을 초과하여 저장되는 대형 객체의 위치 지정 값 보유

2-2. Non-PL/SQL 변수

- ☐ 바인드(Bind) 변수와 호스트(Host) 변수

mb

3. 변수 선언

3-1. 변수 이름 지정 규칙

- ☐ 서로 다른 블록에 있는 두 변수는 동일한 이름 사용 가능
- ☐ 변수 이름은 블록에서 사용되는 테이블 열의 이름과 동일하면 안됨
- ☐ 변수 이름은 30자 이하, 첫 자는 문자이고 나머지는 문자, 숫자, 특수 문자 사용가능
- ☐ 변수에는 v_, 전역변수에는 g_를 접두어로 사용하면 데이터베이스 객체의 이름을 지정할 때 혼란을 방지할 수 있음

3-2. 변수 선언 지침

- ☐ SQL 객체에 사용하는 규칙과 동일한 규칙에 따라 변수 이름 지정
- ☐ 이름 지정 규칙을 따름
- ☐ 할당 연산자 (:=) 또는 DEFAULT 예약어를 사용하여 변수 초기화
- ☐ NOT NULL 및 CONSTANT로 지정된 변수를 초기화
- ☐ 변수를 한 행에 하나씩 선언하면 유지 관리가 용이

mb

3-3. PL/SQL 변수 선언

```
identifier [constant] datatype [not null]
[:= | DEFAULT expr];
```

```
Declare
  v_hiredate      DATE;
  v_deptno        NUMBER(2) NOT NULL := 10;
  v_location      VARCHAR(13) := 'Atlanta';
  v_comm          CONSTANT NUMBER := 1400;
```

- ☑ PL/SQL 블록에서 PL/SQL 식별자를 참조하려면 먼저 선언 부분에서 모든 PL/SQL 식별자 선언
- ☑ 초기값은 선택적이므로 변수 선언을 위해 변수에 값을 할당하지 않아도 됨
- ☑ 선언 부분에서 다른 변수를 참조하는 경우에는 이전 명령문에서 해당 변수를 개별적으로 선언해야 함

mb

4. 변수 초기화 및 값 할당

4-1. 변수 초기화 및 키워드

- ☑ 할당 연산자 (:=)
- ☑ DEFAULT 키워드
- ☑ NOT NULL 제약조건

```
v_hiredate := '15-SEP-99';
```

- ☑ Oracle 8i 부터 가능하며 이전 버전에서는 TO_DATE 함수 사용

```
g_mgr      NUMBER(4) DEFAULT 7839;
```

```
v_location VARCHAR2(13) NOT NULL := 'CHICAGO';
```

mb

4-2. 변수에 값 할당

```
identifier := expr;
```

```
v_hiredate := '31-DEC-98';
```

```
v_ename := 'Maduro';
```

```
SQL> SELECT    sal * 0.10
2  INTO        v_bonus
3  FROM        emp
4  WHERE       empno = 7369;
```

mb

5. 스칼라(Scalar) 데이터 유형

5-1. 스칼라 데이터 유형 개요

- ☐ 단일 값 보유
- ☐ 내부 구성 요소 없음
- ☐ 스칼라 데이터 유형은 숫자, 문자, 날짜, 부울 등 네 가지 범주로 분류

22-MAY-17

256120.08

"Four score and seven years ago our fathers brought forth upon this continent, a new nation, conceived in LIBERTY, and dedicated to the proposition that all are created

TRUE

Yedam

mb

5-2. 기본 스칼라 데이터 유형 (1)

데이터 유형	설 명
VARCHAR2 [(<i>maximum_length</i>)]	<ul style="list-style-type: none"> 가변길이 문자 데이터의 기본 유형 최대 길이 : 32,767 bytes
NUMBER [(<i>precision</i> , <i>scale</i>)]	<ul style="list-style-type: none"> 고정 및 부동 소수점 숫자의 기본 유형
DATE	<ul style="list-style-type: none"> 날짜 및 시간의 기본 유형 날짜 범위 : BC 4712년 ~ AC 9999년
CHAR [(<i>maximum_length</i>)]	<ul style="list-style-type: none"> 고정길이 문자 데이터의 기본 유형 최대 길이 : 32,767 bytes DEFAULT 길이 : 1 bytes
LONG	<ul style="list-style-type: none"> 가변길이 문자 데이터의 기본 유형 최대 길이 : 32,760 bytes LONG 데이터베이스 열의 최대 너비는 2,147,483,647bytes
LONG RAW	<ul style="list-style-type: none"> 이진 데이터 및 바이트 문자열의 기본 유형 최대 길이 : 32,760 bytes PL/SQL에서는 LONG RAW 데이터를 해석하지 않음

mb

5-3. 기본 스칼라 데이터 유형 (2)

데이터 유형	설 명
BOOLEAN	<ul style="list-style-type: none"> 논리 계산에 사용 가능한 세 가지 값 (TRUE, FALSE, NULL) 중 하나를 저장하는 기본유형
BINARY_INTEGER	<ul style="list-style-type: none"> -2,147,483,647 ~ 2,147,483,647 사이에 있는 정수의 기본 유형
PLS_INTEGER	<ul style="list-style-type: none"> -2,147,483,647 ~ 2,147,483,647 사이에 있는 부호 표시 정수의 기본 유형 저장 공간을 적게 사용 NUMBER 및 BINARY_INTEGER 값보다 빠름

mb

5-4. 스칼라 변수 선언 예제

```
v_job          VARCHAR2(9);
v_count        BINARY_INTEGER := 0;
v_total_sal    NUMBER(9,2) := 0;
v_orderdate    DATE := SYSDATE + 7;
c_tax_rate     CONSTANT NUMBER(3,2) := 8.25;
v_valid        BOOLEAN NOT NULL := TRUE;
```

mb

5-5. %TYPE 속성

```
...
v_ename        emp.ename%TYPE;
v_balance      NUMBER(7,2);
v_min_balance   v_balance%TYPE := 10;
...
```

- ☑ 다음에 따라 변수를 선언
 - 데이터베이스 열 정의
 - 이전에 선언한 다른 변수
- ☑ %TYPE에 다음을 접두어로 붙임
 - 데이터베이스 테이블 및 열
 - 이전에 선언한 변수 이름

mb

5-6. 부울(Boolean) 변수

```
v_sal1      := 50000;
v_sal2      := 60000;
v_sal1 < v_sal2 → true 리턴
```


```
v_comm_sal  BOOLEAN := (v_sal1 < v_sal2);
→ boolean 변수를 선언하고 초기화
```

- ☑ 부울 변수에는 TRUE, FALSE, NULL 값만 할당
- ☑ 변수는 논리 연산자 AND, OR, NOT으로 연결
- ☑ 변수의 결과는 항상 TRUE, FALSE, NULL
- ☑ 부울 값을 반환할 때 산술, 문자 및 날짜 표현식 사용 가능

mb

6. 조합(Composite) 데이터 유형

6-1. PL/SQL 레코드 구조

TRUE	22-MAY-17	YEDAM	
------	-----------	-------	---

6-2. PL/SQL 테이블 구조

1	SMITH		
2	JONES		
3	NANCY		
4	TIM		

↑ VARCHAR2

↑ BINARY_INTEGER

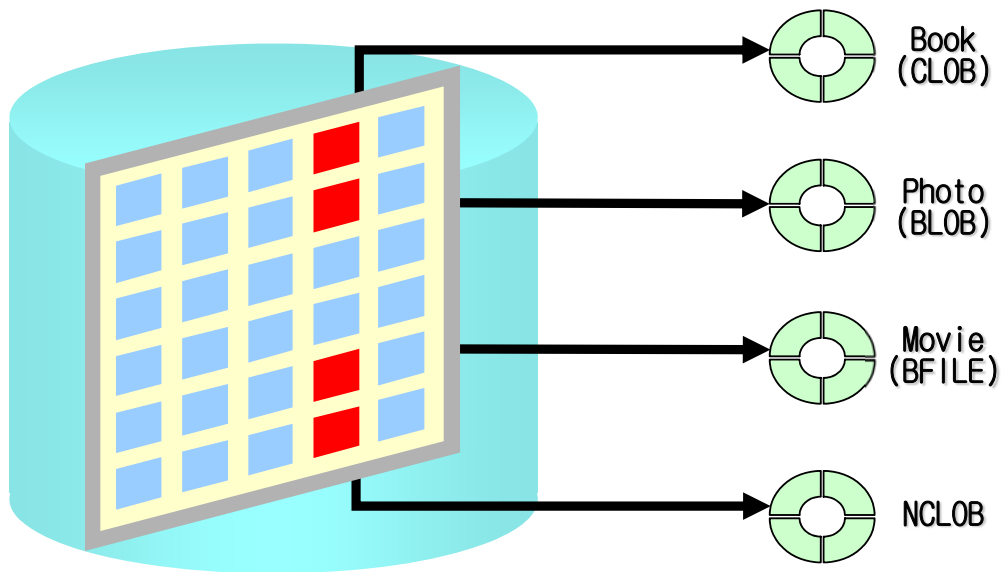
1	5000		
2	2345		
3	12		
4	3456		

↑ NUMBER

↑ BINARY_INTEGER

mb

7. LOB(대형 객체) 데이터 유형

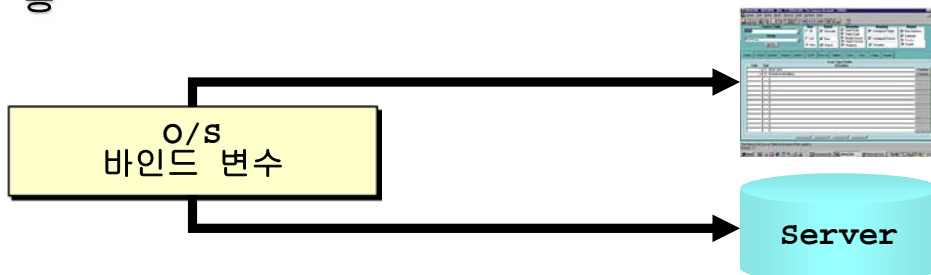


mb

8. Non-PL/SQL 변수

8-1. 바인드(Bind) 변수

- 바인드 변수는 호스트 환경에서 선언한 변수
- 런타임(Runtime) 값을 하나 이상의 PL/SQL 프로그램과 주고 받는데 사용



- PL/SQL 프로그램은 바인드 변수를 다른 변수와 마찬가지로 사용
- 바인드 변수 생성 및 표시

```
SQL> VARIABLE g_n NUMBER
...
SQL> PRINT g_n
```

mb

8-2. Non-PL/SQL 변수 참조

- ☑ Non-PL/SQL 변수를 호스트(Host) 변수로 참조
- ☑ 참조 대상에 콜론(:)을 접두어로 붙임

```
SQL> VARIABLE      g_monthly_sal NUMBER
SQL> ACCEPT        p_annual_sal PROMPT -
> 'Please enter the annual salary: '
Please enter the annual salary: 5000 ← 5000 입력
SQL> DECLARE
2     v_sal      NUMBER(9,2) := &p_annual_sal;
3 BEGIN
4     :g_monthly_sal := v_sal/12;
5 END;
6 /

.....
PL/SQL procedure successfully completed.
SQL> PRINT g_monthly_sal
```

9. DBMS_OUTPUT.PUT_LINE

- ☑ Oracle 지원 패키지 프로시저
- ☑ PL/SQL 블록에서 데이터를 표시하는 또 하나의 방법
- ☑ SET SERVEROUTPUT ON을 사용하여 SQL*Plus에서 사용 가능하도록 설정

```
SQL> SET SERVEROUTPUT ON
SQL> ACCEPT        p_annual_sal PROMPT -
> 'Please enter the annual salary: '
Please enter the annual salary: 5000 ← 5000 입력
SQL> DECLARE
2     v_sal      NUMBER(9,2) := &p_annual_sal;
3 BEGIN
4     v_sal := v_sal/12;
5     DBMS_OUTPUT.PUT_LINE ('The monthly salary is ' ||
6     TO_CHAR(v_sal));
7 END;
8 /

.....
The monthly salary is 416.67
PL/SQL procedure successfully completed.
```