

Desarrollo de un Bloque Simple de PL/SQL

Declaración de Variables y Constantes en PL/SQL

- Es necesario declarar todos los identificadores dentro de la sección de declaración, antes de hacer referencia a ellos dentro del bloque PL/SQL.
- Las declaraciones asignan espacio para almacenar un valor, especificar su tipo de dato, y nombrar la ubicación del almacenamiento para poder referenciarlo.
- La sintaxis es la siguiente:

```
identificador [CONSTANT] tipo_dato [NOT NULL] [:= | DEFAULT expresión];
```

Declaración de Variables y Constantes en PL/SQL

- Uso de la Cláusula DEFAULT:
 - Se puede utilizar la palabra reservada DEFAULT en lugar del operador de asignación para inicializar variables.
 - Ejemplos:

```
tipo_sangre CHAR := 'O';  
valido      BOOLEAN := FALSE;
```

```
tipo_sangre CHAR DEFAULT 'O';  
valido      BOOLEAN DEFAULT FALSE;
```

Declaración de Variables y Constantes en PL/SQL

- Uso de la Cláusula NOT NULL:
 - Las declaraciones pueden imponer la restricción NOT NULL.
 - No se puede asignar un valor nulo a una variable definida como NOT NULL.
 - La restricción NOT NULL debe ser seguida por una cláusula de inicialización.
 - Ejemplo:

```
cuenta_id INTEGER(4) NOT NULL := 9999;
```

Declaración de Variables y Constantes en PL/SQL

- Restricciones:
 - Es necesario declarar una variable o constante antes de hacer referencia a ella en otras sentencias.
 - No se puede declarar una lista de variables que tengan el mismo tipo de datos en una sola línea.

Declaración de Variables y Constantes en PL/SQL

- Tipos de Datos:
 - Cada constante y variable tiene un tipo de datos, que especifica un formato de almacenamiento, restricciones y rango de valores válidos.
 - Existen tipos de datos escalares, compuestos, referencia y LOBs.

Declaración de Variables y Constantes en PL/SQL

- Tipos de Datos Escalares:
 - No tienen componentes internos.
 - Los tipos de datos escalares se clasifican en cuatro categorías:
 - numéricos,
 - de caracteres,
 - lógicos (booleanos) y
 - fecha y hora.

Declaración de Variables y Constantes en PL/SQL

- Tipos de Datos Escalares - Tipos Numéricos
 - Los tipos numéricos permiten almacenar datos numéricos (enteros, números reales, y números de punto flotante), representan cantidades y se pueden realizar cálculos sobre ellos.
 - Son tipos numéricos: BINARY_INTEGER y NUMBER.

Declaración de Variables y Constantes en PL/SQL

- Tipos de Datos Escalares - Tipos Carácter
 - Los tipos carácter permiten almacenar datos alfanuméricos, representar palabras y texto, y manipular cadenas de caracteres.
 - Son tipos carácter: CHAR, LONG, RAW, LONG RAW ROWID y VARCHAR2.

Declaración de Variables y Constantes en PL/SQL

- Tipos de Datos Escalares - Tipos Lógicos (Booleanos)
 - El tipo de datos Lógico se usa para almacenar valores lógicos TRUE, FALSE y NULL. Sólo se permiten operaciones lógicas sobre variables "booleanas".
 - Es tipo Lógico o booleano: BOOLEAN.

Declaración de Variables y Constantes en PL/SQL

- Tipos de Datos Escalares - Tipos Fecha y Hora
 - El tipo de datos Fecha y hora se usa para almacenar valores de fecha y hora de longitud fija. Los valores incluyen la hora del día en segundos desde la medianoche.
 - Es tipo fecha y hora: DATE.

Tipos de Datos Escalares

Tipo de Datos Escalar		Descripción
Numéricos	BINARY_INTEGER	Números enteros entre -2.147.483.647 y 2.147.483.647.
	NUMBER[(precisión,escala)]	Números de punto fijo y flotante. Rango :1,0E-130 a 9,99E125. Precisión máxima: 38 dígitos decimales. Escala: rango: -84 a 127.
de Caracteres	CHAR[(longitud_máxima)]	Datos de caracteres de longitud fija. Longitud máxima: hasta 32.767 bytes. Si no se especifica longitud máxima, por defecto será de 1.
	LONG	Cadenas de caracteres de longitud variable. Longitud máxima: 32.760 bytes.
	RAW(longitud_máxima)	Datos binarios o cadenas de bytes. Longitud máxima: hasta 32.767 bytes.
	LONG RAW	Datos binarios o cadenas de bytes. Longitud máxima: 32.760 bytes. El dato LONG RAW no es interpretado por PL/SQL.
	ROWID	Representa la dirección de almacenamiento de una fila.
	VARCHAR2(longitud_máxima)	Caracteres de longitud variable. Rango: 1 a 32.767 bytes.
Booleanos	BOOLEAN	Almacena los valores lógicos TRUE, FALSE y NULL.
Fecha y hora	DATE	Valores de fecha y hora de longitud fija. Rango: 01/01/4712 AC al 31/12/4712 DC.

Tipos de Datos Escalares

- Ejemplos de Declaraciones de Variables Escalares:

v_sexo	CHAR(1);	-- almacena el código de sexo (F o M)
v_contador	BINARY_INTEGER := 0;	-- cuenta el nro. de iteraciones en un loop
v_total_salario	NUMBER(9,2) := 0;	-- acumulador de salario total de un dpto
v_fecha_orden	DATE := SYSDATE + 7;	-- almacena fecha de envío de la orden
c_iva_resp_insc	CONSTANT NUMBER(4,2) := 21.5;	-- porcentaje alícuota IVA
v_valido	BOOLEAN NOT NULL := TRUE;	-- indicador si un dato es válido

Declaración de Variables y Constantes en PL/SQL

- El Atributo %TYPE:
 - El atributo %TYPE provee el tipo de dato de una variable o columna de la base de datos.
 - Ejemplos:

```
credito  REAL(7,2);  
debito   credito%TYPE;
```

```
balance          NUMBER(7,2);  
balance_minimo   balance%TYPE := 10.00;
```

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos:
 - Un tipo de datos compuesto contiene componentes internos y es reusable.
 - PL/SQL ofrece dos clases de tipos compuestos:
 - TABLE y
 - RECORD.

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Tablas PL/SQL:
 - Las tablas PL/SQL son objetos del tipo TABLE.
 - Una tabla PL/SQL es una colección ordenada de elementos del mismo tipo (tipo escalar).
 - Las tablas PL/SQL usan una clave primaria para dar acceso a las filas. Cada elemento tiene un número de índice único, de tipo `BINARY_INTEGER`, que determina su posición en la colección ordenada.

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Creación de Tablas PL/SQL:

```
TYPE nombre_tipo_tabla IS TABLE OF tipo_escalor [NOT NULL]  
    INDEX BY BINARY_INTEGER;
```

```
nombre_tabla_plsql    nombre_tipo_tabla;
```

```
TYPE tabla_de_caracteres IS TABLE OF VARCHAR2(25)  
    INDEX BY BINARY_INTEGER;
```

```
tabla_nombres          tabla_de_caracteres;  
tabla_apellidos        tabla_de_caracteres;
```

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Tablas PL/SQL:
 - Las tablas PL/SQL no tienen un tamaño limitado, pueden crecer dinámicamente.
 - No requieren números índices consecutivos.
 - No se puede inicializar una tabla PL/SQL en su declaración.
 - Para referenciar elementos en una tabla PL/SQL, se especifica un número de índice

```
nombre_tabla_plsql(índice)
```

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Tablas PL/SQL:
 - Cada tabla PL/SQL tiene los atributos EXISTS, COUNT, FIRST, LAST, PRIOR, NEXT y DELETE, que hacen fácil su uso y mantenimiento.
 - La primer referencia a un elemento en una tabla PL/SQL debe ser una asignación, previo a esto el elemento no existe. Si se hace referencia a un elemento que no existe, se genera la excepción predefinida NO_DATA_FOUND.

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Registros PL/SQL:
 - Un registro es un grupo de elementos de datos relacionados, almacenados en campos, cada uno con su propio nombre y tipo de datos, que se pueden tratar como una unidad lógica.
 - Los registros son útiles cuando se necesita almacenar una fila de datos de una tabla para su procesamiento en un bloque PL/SQL.

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Creación de Registros PL/SQL:

```
TYPE nombre_tipo_registro IS RECORD
  (nombre_campo      tipo_campo [[NOT NULL] {:= | DEFAULT} expresión]
  [, nombre_campo    tipo_campo [[NOT NULL] {:= | DEFAULT} expresión]]
  );

nombre_registro      nombre_tipo_registro;
```

```
TYPE tipo_registro_emp IS RECORD
  (apellido          VARCHAR2(25)
  , nombre           VARCHAR2(25)
  , sexo             CHAR(1)
  );

registro_empleado    tipo_registro_emp;
```

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - El Atributo %ROWTYPE:
 - El atributo %ROWTYPE proporciona un tipo de registro que representa una fila en una tabla (o vista).
 - El registro puede almacenar una fila entera de datos seleccionados de una tabla o recuperados de un cursor.
 - Los campos del registro y las columnas correspondientes de la fila de la tabla tienen los mismos nombres y tipos de datos.

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - Ventajas de Utilizar el Atributo %ROWTYPE:
 - La cantidad y el tipo de datos de las columnas referenciadas de la base de datos pueden ser desconocidos y pueden cambiar durante la ejecución.
 - Es útil cuando se recupera una fila mediante la sentencia SELECT, ya que permite declarar registros estructurados como filas de una tabla de la base de datos.

Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - El Atributo %ROWTYPE:
 - La sintaxis para declarar un registro utilizando el atributo %ROWTYPE es la siguiente:

```
nombre_registro      {nombre_cursor | nombre_tabla}%ROWTYPE;
```


Declaración de Variables y Constantes en PL/SQL

- Tipos de datos compuestos - El Atributo %ROWTYPE:
 - Ejemplo en el que se declara un registro que puede almacenar una fila entera de la tabla emp y se selecciona una fila de dicha tabla.

```
DECLARE
    emp_reg    emp%ROWTYPE;
...
BEGIN
    SELECT * INTO emp_reg FROM emp WHERE empno = mi_empno;
...
END;
```

Reglas de Sintaxis de los Bloques PL/SQL

- Los identificadores pueden contener un máximo de 30 caracteres, pero deben comenzar con un carácter alfabético.
- Las palabras reservadas no pueden ser usadas como identificadores a menos que se encuentren encerradas entre comillas dobles (por ejemplo, "SELECT").

Reglas de Sintaxis de los Bloques PL/SQL

- No se debe elegir el mismo nombre para un identificador, que el nombre de alguna columna de una tabla usada en el bloque.
- Una sentencia puede extenderse en más de una línea, pero las palabras reservadas no.

Reglas de Sintaxis de los Bloques PL/SQL

- Las unidades lexicográficas (por ejemplo, identificadores y literales) pueden estar separados por uno o más espacios u otro delimitador que no pueda ser confundido con una parte de la unidad lexicográfica.
- Los literales de caracteres y fechas deben encerrarse entre apóstrofes.

Reglas de Sintaxis de los Bloques PL/SQL

- Los literales numéricos pueden expresarse como un valor simple (por ejemplo, 2E5, que significa $2 * 10^5 = 200.000$).
- Los comentarios multilínea pueden estar encerrados entre los símbolos `/*` y `*/`. Un comentario de una línea comienza con `--` y finaliza con la marca de fin de línea.

Alcance y Visibilidad de las Variables

- Las referencias a un identificador se resuelven de acuerdo con su alcance y visibilidad.
- El alcance de un identificador es la región de una unidad de programa (bloque, subprograma, paquete) desde la cual se puede referenciar al identificador.
- Un identificador es visible sólo en las regiones desde las cuales se lo puede referenciar utilizando un nombre no calificado.

Alcance y Visibilidad de las Variables

Scope

Outer x

```
DECLARE
  X REAL;
BEGIN
  ...
  DECLARE
    X REAL;
  BEGIN
    ...
  END;
  ...
END;
```

Visibility

```
DECLARE
  XREAL;
BEGIN
  ...
  DECLARE
    XREAL;
  BEGIN
    ...
  END;
  ...
END;
```

Alcance y Visibilidad de las Variables

Inner x

```
DECLARE  
    X REAL;  
BEGIN
```

```
    ...  
    DECLARE  
        X REAL;  
    BEGIN  
        ...  
    END;
```

```
    ...  
END;
```

```
DECLARE  
    X REAL;  
BEGIN
```

```
    ...  
    DECLARE  
        X REAL;  
    BEGIN  
        ...  
    END;
```

```
    ...  
END;
```


Alcance y Visibilidad de las Variables

```
DECLARE
  a CHAR;
  b REAL;
BEGIN
  -- identificadores disponibles aquí: a (CHAR), b

  DECLARE
    a INTEGER;
    c REAL;
  BEGIN
    -- identificadores disponibles aquí: a (INTEGER), b, c
  END;

  DECLARE
    d REAL;
  BEGIN
    -- identificadores disponibles aquí: a (CHAR), b, d
  END;

  -- identificadores disponibles aquí: a (CHAR), b
END;
```

Asignación de Valores a las Variables

- Para asignar o reasignar un valor a una variable, se debe escribir una sentencia de asignación PL/SQL.
- Se debe indicar en forma explícita la variable que recibirá el nuevo valor a la izquierda del operador de asignación (:=).

Asignación de Valores a las Variables

- La sintaxis de asignación es la siguiente:

```
identificador := expr;  
nombre_tabla_plsql(valor_clave_primaria) := expr;  
nombre_registro_plsql.nombre_campo := expr;
```

- Las variables y constantes se inicializan cada vez que se ingresa a un bloque o subprograma.
- Por defecto, las variables se inicializan a NULL. A menos que expresamente se inicialice una variable, su valor es indefinido.

Operadores

- Los operadores lógicos, aritméticos y de concatenación usados en PL/SQL son los mismos que se utilizan en SQL. Existe además un operador exponencial (**).
- Se puede controlar el orden en que se realizan las operaciones mediante el uso de paréntesis como en SQL.

Funciones

- La mayoría de las funciones válidas en SQL se pueden utilizar en las expresiones de PL/SQL.
 - Funciones numéricas de una fila.
 - Funciones de caracteres de una fila.
 - Funciones de conversión de tipos de datos.
 - Funciones de fecha y hora.
 - Funciones misceláneas.

Conversión de Tipos de Datos

- Dentro de una expresión, debe asegurarse que los tipos de datos son los mismos.
- Si se mezclan tipos de datos en una misma expresión, se deberán usar las funciones de conversión de tipos de datos que correspondan para realizar la conversión.

```
TO_CHAR(valor, fm);  
TO_DATE(valor, fm);  
TO_NUMBER(valor, fm);
```

Referencias a Variables que no son de PL/SQL

- Se pueden referenciar variables declaradas en el ambiente de invocación dentro de sentencias PL/SQL.
- Esto incluye variables de lenguajes embebidos declarados en los programas de precompilación, campos de una pantalla de una aplicación de Forms y variables ligadas de SQL*Plus.

```
:g_salario_anual := v_salario * 12;
```

Comentarios

- Para comentar el código fuente, se utilizan dos guiones (--) si el comentario es de una sola línea, o se encierra el comentario entre los símbolos /* y */ si el comentario abarca varias líneas.