

## UA4B. PL/SQL

---

### 4.12. Variables y constantes

---

Para declarar variables o constantes pondremos el nombre de la variable, seguido del tipo de datos y opcionalmente una asignación. Si es una constante antepondremos la palabra **CONSTANT** al tipo de dato (lo que querrá decir que no podemos cambiar su valor). Podremos sustituir el operador de asignación en las declaraciones por la palabra reservada **DEFAULT**. También podremos forzar a que no sea nula utilizando la palabra **NOT NULL** después del tipo y antes de la asignación. Si restringimos una variable con **NOT NULL** deberemos asignarle un valor al declararla, de lo contrario PL/SQL lanzará la excepción **VALUE\_ERROR** (no te asustes que más adelante veremos lo que son las excepciones, pero como adelanto te diré que es un error en tiempo de ejecución).

```
id SMALLINT;  
hoy DATE := sysdate;  
pi CONSTANT REAL:= 3.1415;  
id SMALLINT NOT NULL; --ilegal, no está inicializada  
id SMALLINT NOT NULL := 9999; --legal
```

El alcance y la visibilidad de las variables en PL/SQL será el típico de los lenguajes estructurados basados en bloques, aunque eso lo veremos más adelante.

### Conversión de tipos

---

Aunque en PL/SQL existe la conversión implícita de tipos para tipos parecidos, siempre es aconsejable utilizar la conversión explícita de tipos por medio de funciones de conversión (TO\_CHAR, TO\_DATE, TO\_NUMBER, ...) y así evitar resultados inesperados.

## Precedencia de operadores

---

Al igual que en nuestro lenguaje matemático se utiliza una precedencia entre operadores a la hora de realizar las operaciones aritméticas, en PL/SQL también se establece dicha precedencia para evitar confusiones. Si dos operadores tienen la misma precedencia lo aconsejable es utilizar los paréntesis (al igual que hacemos en nuestro lenguaje matemático) para alterar la precedencia de los mismos ya que las operaciones encerradas entre paréntesis tienen mayor precedencia. En la tabla siguiente se muestra la precedencia de los operadores de mayor a menor.

*Precedencia de operadores.*

Operador.	Operación.
**, NOT	Exponenciación, negación lógica.
+, -	Identidad, negación.
*, /	Multiplicación, división.
+, -,	Suma, resta y concatenación.
=, <=>, <, >, <=>, >=, IS NULL, LIKE, BETWEEN, IN	Comparaciones.
AND	Conjunción lógica
OR	Disyunción lógica.

Obra publicada con **Licencia Creative Commons Reconocimiento Compartir igual 4.0**  
<<http://creativecommons.org/licenses/by-sa/4.0/>>