

Variables

Índice

- 01 [Declaración de variables](#)
- 02 [Variables definidas por el usuario](#)
- 03 [Variables locales](#)
- 04 [Variables del sistema](#)



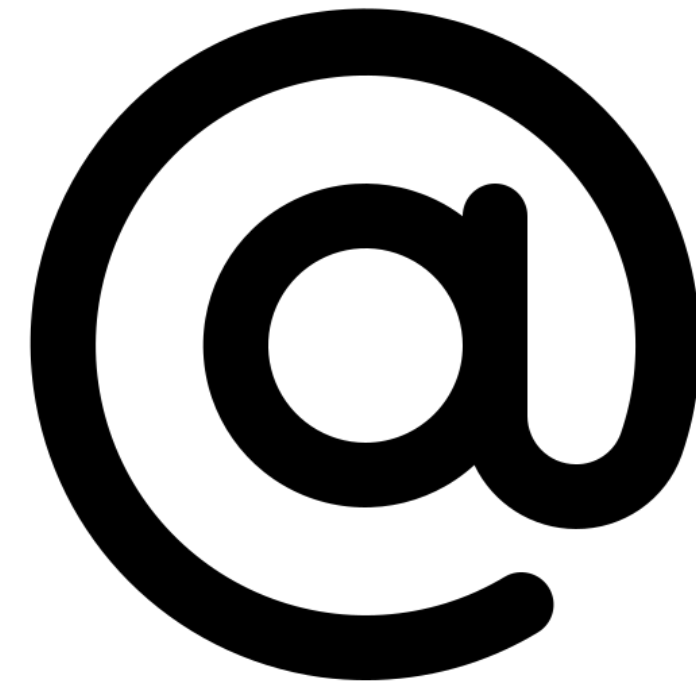
01

Declaración de variables

Declaración de variables

Como ya hemos visto en clases anteriores, las variables son elementos que almacenan **datos** que pueden ir cambiando a lo largo de la ejecución.

Para hacer esto, almacena el valor de la misma en su primera declaración y se refiere a ella en las siguientes declaraciones.



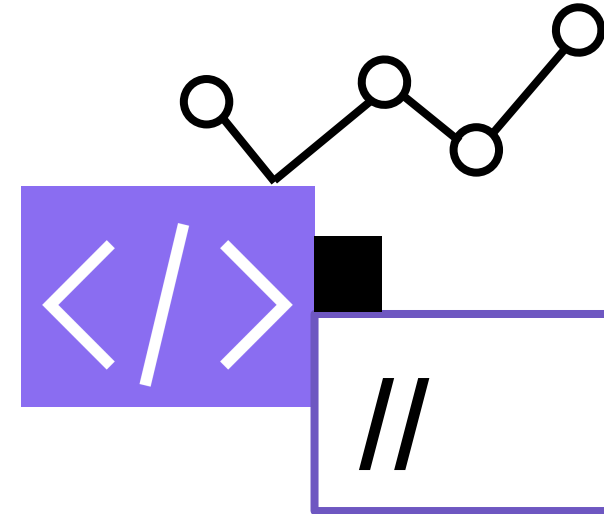
02

Variables definidas por el usuario

Variables definidas por el usuario

MySQL reconoce diferentes tipos de variables. El primer tipo son las variables definidas por el usuario, identificadas por un símbolo @ usado como prefijo.

En MySQL, se puede acceder a las variables definidas por el usuario sin necesidad de declararlas o inicializarlas previamente. Si no son inicializadas, tienen por defecto el valor NULL.



Variables definidas por el usuario

Para inicializar una variable definida por el usuario, se necesita usar una **declaración SET o SELECT**. Se puede inicializar muchas variables a la vez, separando cada declaración de asignación con una coma.

```
SQL SET @nombre_variable1 = valor, @nombre_variable2 = valor;
```

Una vez que se asigna un valor a una variable, tendrá un tipo de dato de acuerdo al valor dado. En el ejemplo anterior, si a @nombre_variable1 lo igualamos a 1 y @nombre_variable2 lo igualamos a 2, las variables se convierten del tipo INT.

Variables definidas por el usuario

La vida útil de una variable definida por el usuario dura mientras la sesión esté activa, y es invisible para otras sesiones. Una vez que la sesión se cierra, la variable desaparece.

Hay 5 tipos de datos que se pueden asignar a una variable definida por el usuario:

- Varchar o Char
- Int
- Decimal
- Float
- Null, que puede ser asociado con cualquier tipo

03

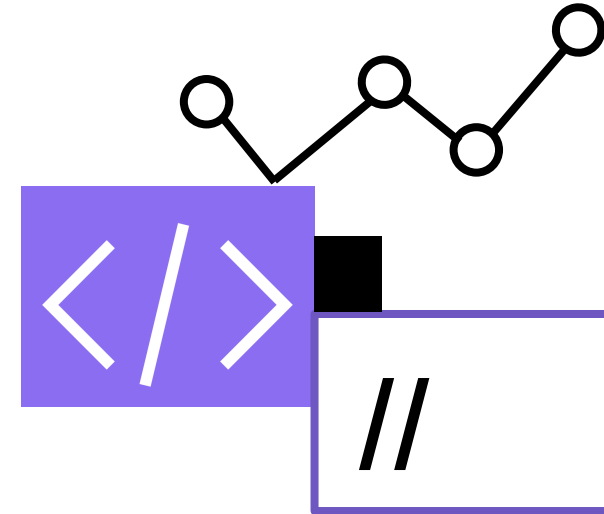
Variables locales

Variables locales

Las variables locales **no necesitan el prefijo @** en sus nombres, pero deben ser declaradas antes de que puedan ser usadas. Para declarar una variable local, se puede usar la declaración **DECLARE** o usarla como un parámetro dentro de una declaración **STORED PROCEDURE** o **FUNCTION**.

Cuando se declara una variable local, opcionalmente, se le puede asignar un valor por defecto. Si no se asigna ningún valor por defecto, la variable se inicializa con un valor **NULL**.

Cada variable vive dentro de un ámbito, delimitado por el bloque **BEGIN ... END** que contiene su declaración.



Variables locales

En el siguiente ejemplo, podemos ver que se utiliza la variable *precio_sin_impuestos* como parámetro para pasar un valor a la función. Esa variable se usa luego en la sentencia RETURN para multiplicarla con la variable local IVA. Luego, la función retorna el cálculo.

```
SQL CREATE FUNCTION agregar_IVA(precio_sin_impuestos DOUBLE(10,12))
      RETURNS DOUBLE(10,12)
      BEGIN
          DECLARE IVA INT DEFAULT 21;
          RETURN ((precio_sin_impuestos * IVA) / 100) + precio_sin_impuestos;
      END
```

04

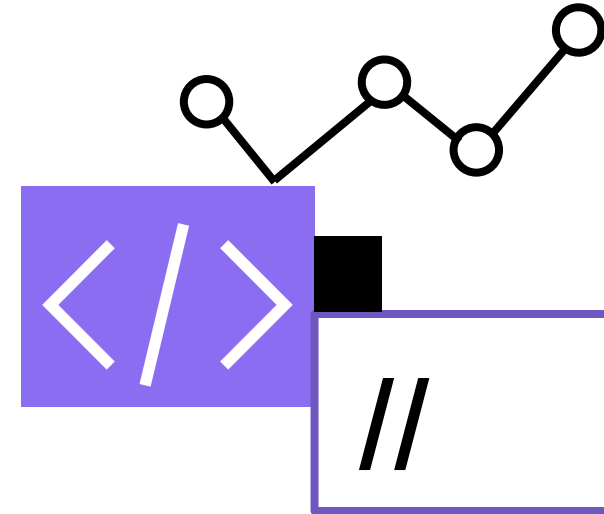
Variables del sistema

Variables del sistema

Hay un tercer tipo de variables llamadas variables del sistema que se usan para almacenar valores que afectan a las conexiones individuales de los clientes —variables de **SESSION**— o que afectan a toda la operación del servidor —variables **GLOBAL**—.

Las variables del sistema pueden ser identificadas usando un doble signo @ como prefijo o usando las palabras GLOBAL o SESSION en la sentencia **SET**.

Otra forma de diferenciar las variables del sistema GLOBAL y SESSION es usar un segundo prefijo: global o session

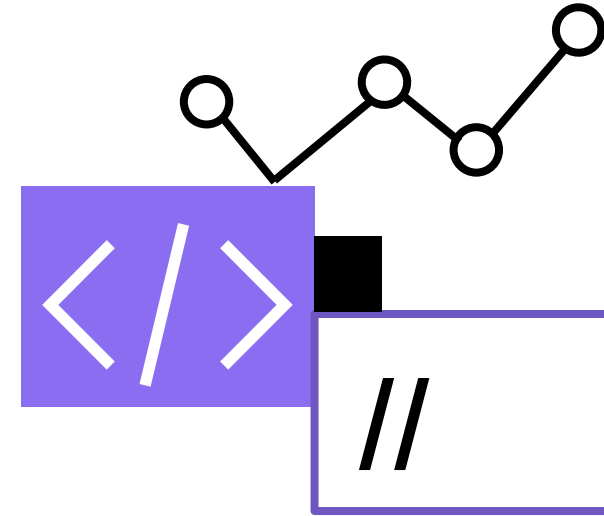


Variables del sistema

Aquí hay algunos ejemplos de cómo se puede asignar valores a las variables del sistema:

```
SQL  - Session
      SET interactive_timeout=30000;
      SET SESSION interactive_timeout=30000;
      SET @@interactive_timeout=30000;
      SET @@local.interactive_timeout=30000;

      -- Global
      SET @@global.interactive_timeout=30000;
      SET GLOBAL interactive_timeout=30000;
```



Variables del sistema

Para ver las variables de sistema en uso dentro de una sesión o en el servidor, se puede usar la sentencia **SHOW VARIABLES**. Se pueden añadir los operadores que se usan en el WHERE para filtrar esta lista, si queremos obtener el valor de algunas variables específicas.

```
SQL SHOW VARIABLES LIKE '%timeout%'
```

¡Muchas Gracias!