



TIPOS DE DATOS

LECCIÓN 2

En esta lección, exploramos qué tipos de datos existen, cómo se pueden convertir entre diferentes tipos de datos y los aspectos a tener en cuenta cuando se almacena información en una variable de una base de datos o de un lenguaje de programación para evitar pérdida de información u operaciones erróneas.

Tipos de Datos



Cada tipo de dato ocupa un espacio determinado en Memoria y tiene límites de qué cantidades puede representar. Los límites pueden variar dependiendo del lenguaje de programación, sin embargo, los tipos de datos tienden a ser similares.

En el estándar SQL los tipos de datos admitidos son:

INTEGER o INT:

Representa números enteros. Puede ser de longitud fija (por ejemplo, INT) o variable (por ejemplo, INTEGER).

VARCHAR o CHAR:

VARCHAR se utiliza para almacenar cadenas de longitud variable, mientras que CHAR, se utiliza para cadenas de longitud fija. VARCHAR es más eficiente en términos de almacenamiento, ya que solo utiliza el espacio necesario para la cadena.

DECIMAL o NUMERICO:

Utilizado para almacenar números decimales. Puedes especificar la precisión (número total de dígitos) y la escala (número de dígitos después del punto decimal).

DATE



Almacena fechas en formato 'YYYY-MM-DD'. Es utilizado para representar días específicos en el calendario.

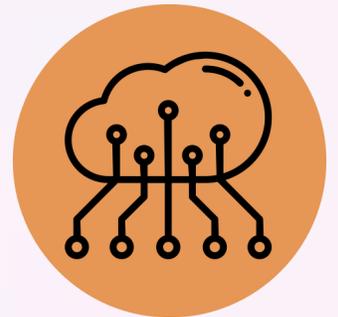
TIME:

Almacena la información de tiempo en formato 'HH:MI:SS'.
Representa la hora del día.

DATETIME o TIMESTAMP:

Almacena tanto la fecha como la hora en formato 'YYYY-MM-DD HH:MI:SS'.
TIMESTAMP a menudo se utiliza para representar momentos específicos en el tiempo.

BOOLEAN o BOOL:



Representa valores de verdad o falsedad (true o false).

BLOB (Binary Large Object):

Almacena datos binarios, como imágenes, audio o archivos.

ENUM:

Define un conjunto de valores posibles para un campo.
Se utiliza para limitar las opciones disponibles.

En los lenguajes de programación asociados a ciencia de datos (Python, Scala y C++) también se emplean tipos de datos. En el caso de Python se hacen implícitamente, es decir, el programador no debe especificar el tipo, sino que el intérprete de Python se encarga de definir los datos. Sin embargo, sí existe una diferencia entre los tipos de datos.

Para el lenguaje de programación Python, algunos de los tipos de datos comunes son:

int:

Representa números enteros, por ejemplo, 5 o -10.

float:

Representa números de punto flotante (decimales), por ejemplo, 3.14 o -0.5.

str:

Representa cadenas de texto, por ejemplo, 'Hola, mundo!' o "Python".

bool:

Representa valores booleanos, True o False.



list:

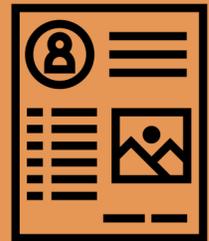
Representa listas ordenadas y mutables de elementos, por ejemplo, [1, 2, 3] o ['manzana', 'banana', 'cereza'].

tuple:

Representa tuplas ordenadas e inmutables de elementos, por ejemplo, (1, 2, 3) o ('rojo', 'verde', 'azul').

dict:

Representa diccionarios, que son colecciones no ordenadas de pares clave-valor, por ejemplo, {'nombre': 'Juan', 'edad': 25}.



set:

Representa conjuntos no ordenados de elementos únicos, por ejemplo, {1, 2, 3}.

NoneType:

Representa el tipo de datos None, que se utiliza para indicar la ausencia de un valor o un objeto nulo.

complex:

Representa números complejos, por ejemplo, $3 + 4j$.



bytes y bytearray:

Representa secuencias de bytes inmutables y mutables respectivamente

Como se ve en las descripciones, cada tipo de dato tiene un uso específico y también tiene límites. Tal es el caso de los enteros, un entero no puede contener información de decimales, así como un número de tipo float no puede contener números imaginarios o complejos. Los tipos de datos les dicen a los programas cómo se deben leer de memoria los datos y qué operaciones hay asociadas a los datos.

La conversión entre tipos de datos es una operación común en programación para adaptar valores de un tipo a otro compatible. En muchos lenguajes de programación, incluido Python, las conversiones explícitas e implícitas son fundamentales para manipular datos de manera efectiva. Las conversiones explícitas, también conocidas como "casting", ocurren cuando un programador indica específicamente la transformación de un tipo de dato a otro. Por ejemplo, en Python, se puede realizar una conversión explícita de un número entero a un número de punto flotante utilizando la función `float()`. Por otro lado, las conversiones implícitas, también llamadas "coerciones", ocurren automáticamente cuando el lenguaje ajusta automáticamente los tipos de datos para realizar operaciones, como sumar un entero y un flotante, donde el entero se convierte implícitamente a flotante para la operación.



La conversión entre tipos de datos es esencial para garantizar la consistencia y la precisión en programas. Sin embargo, es crucial tener en cuenta posibles pérdidas de información al convertir entre tipos, especialmente cuando se trata de conversiones de tipos numéricos, ya que las representaciones y precisiones pueden variar. La elección adecuada entre conversiones explícitas e implícitas depende de las necesidades del programa y la intención del programador para garantizar un comportamiento correcto y eficiente. Tanto los lenguajes de programación como las bases de datos incluye funciones de Casting (conversión) de datos para especificar al software cómo debe manipular datos. Como ejemplos de aplicación se puede emplear Excel y crear varios números enteros iguales . Posteriormente dar formato a cada número de fecha, de porcentaje y de texto. A cada número intentar realizar sumas y restas con otros números y validar cómo es el comportamiento de las operaciones (se pueden o no hacer) y el valor resultante entre datos de diferentes tipos. Esto permitirá a los estudiantes descubrir la importancia de asignar apropiadamente los datos.