

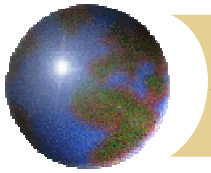
Desarrollo de Portales WEB *Lenguaje PHP*

Gonzalo Anchante Hurtado
Docente

INSTITUTO SUPERIOR TECNOLÓGICO



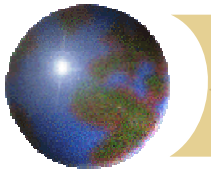
R.M. N° 420 - 94 - ED



TIPOS DE DATOS EN MYSQL.

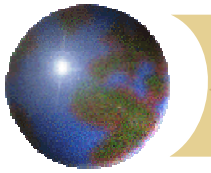
A. Campos numéricos

- ❖ MySQL soporta los tipos numéricos exactos (INTEGER, NUMERIC, DECIMAL, y SMALLINT) y los tipos numéricos aproximados (FLOAT, DOUBLE precision y REAL).
- ❖ Los campos que contienen números enteros admiten el parámetro UNSIGNED que implica que no admita signos por lo que solo aceptaría enteros positivos.



TIPOS DE DATOS EN MYSQL.

- ⊕ Todos los campos numéricos admiten el parámetro ZEROFILL cuya función es completar el campo con ceros a la izquierda hasta su longitud máxima

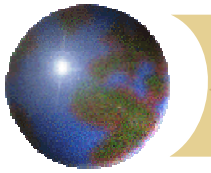


TIPOS DE DATOS EN MYSQL.

TINYINT [(M)] [UNSIGNED] [ZEROFILL]

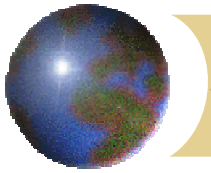
—

- ✚ Número entero muy pequeño. Con la opción UNSIGNED puede tomar valores entre 0 y 255. En caso contrario, puede estar comprendido entre -128 y 127.
- ✚ Si el parámetro ZEROFILL solo tiene sentido junto con la opción UNSIGNED ya que no tiene ningún sentido tratar de rellenar con ceros a la izquierda de un número negativo.



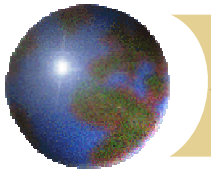
TIPOS DE DATOS EN MYSQL.

- ✿ Si intentamos insertar un valor fuera de rango registraría el valor dentro del rango más próximo a él.
- ✿ P. ej.: Si tratamos de insertar el valor 437 escribiría 127 ó 255, este último en el caso de tener la opción UNSIGNED.
- ✿ Si pretendiéramos insertar -837 con la opción UNSIGNED escribiría 0 y sin ella pondría -128.
- ✿ El tamaño de un campo TINYINT es de 1 byte.



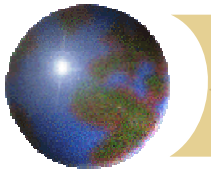
TIPOS DE DATOS EN MYSQL.

- ❖ **SMALLINT [(M)] [UNSIGNED]**
[ZEROFILL] Número entero *pequeño*. Con la opción **UNSIGNED** puede tomar valores entre **0** y **65 535**. En caso contrario, puede estar comprendido entre **-32 768** y **32 767**



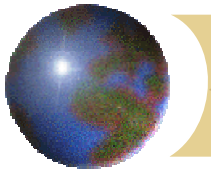
TIPOS DE DATOS EN MYSQL.

- ❖ **MEDIUMINT [(M)] [UNSIGNED]**
[ZEROFILL] Número entero
mediano. Con la opción **UNSIGNED**
puede tomar valores entre **0** y **16 777**
215. En caso contrario, puede estar
comprendido entre **-8 388 608** y **8**
388 607



TIPOS DE DATOS EN MYSQL.

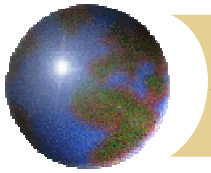
- ❖ **INT [(M)] [UNSIGNED]**
[ZEROFILL] Número entero. Con la opción **UNSIGNED** puede tomar valores entre **0** y **4 294 967 295**. En caso contrario, puede estar comprendido entre **-2 147 483 648** y **2 147 483 647**.



TIPOS DE DATOS EN MYSQL.

- ❖ **BIGINT [(M)] [UNSIGNED]**
[ZEROFILL] Número entero *grandes*. Con la opción **UNSIGNED** puede tomar valores entre **0** y **18 446 744 073 709 551 615**. En caso contrario, puede estar comprendido entre **-9 223 372 036 854 775 808** y **21 474 839 223 372 036 854 775 807 647**, pero al usarlo desde PHP estará sujeto a las limitaciones máximas de los valores numéricos de este.

Su tamaño es de *8 bytes*.



TIPOS DE DATOS EN MYSQL.

B. Números de coma flotante

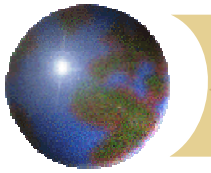
Por la estructura binaria de los microprocesadores y habida cuenta de que algunos números **no enteros** -sin ir más lejos, el **0.1**- requerirían *infinitos caracteres binarios* para su representación exacta, se hace necesario introducir un **redondeo** en su tratamiento informático y como consecuencia de ello asumir que se generan **errores de medida**



TIPOS DE DATOS EN MYSQL.

El Standar de Aritmética de Punto Flotante estableció dos niveles de precisión:

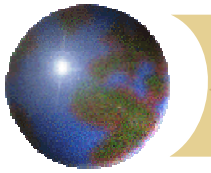
- ❖ **Precisión Simple** en la que **todo número debe ser almacenado en 32 bits**
(4 bytes)
- ❖ **Doble** precisión en la que los **números se almacenan en 64 bits**
(8 bytes).



TIPOS DE DATOS EN MYSQL.

FLOAT(x) [ZEROFILL]

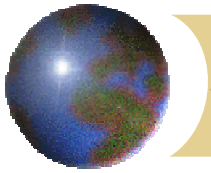
- ❖ Número de coma flotante. Ignora la opción UNSIGNED pero si acepta ZEROFILL por lo que debe prestarse atención a estas opciones ya que no sería demasiado habitual una presentación como esta: 000-3.47



TIPOS DE DATOS EN MYSQL.

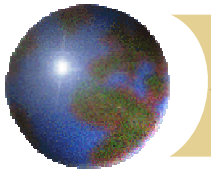
CAMPOS TIPO CADENA DE CARACTERES CHAR (M) [BINARY]

- ❖ Es una cadena de tamaño fijo que se completa a la derecha por espacios si es necesario.
- ❖ El parámetro M puede valer de 1 a 255 caracteres.
- ❖ Los espacios finales son suprimidos cuando la cadena es insertada en el registro.
- ❖ Los valores de tipo CHAR son elegidos y comparados sin tener en cuenta Mayúsculas / Minúsculas y utilizan el juego de caracteres por defecto.



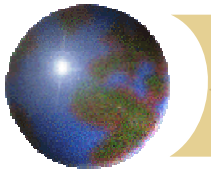
TIPOS DE DATOS EN MYSQL.

- ❖ **VARCHAR(M) [BINARY]**
- ❖ Es una cadena de caracteres de longitud variable. Su tamaño máximo - especificado en el parámetro M- puede estar comprendido entre 1 y 255 caracteres



TIPOS DE DATOS EN MYSQL.

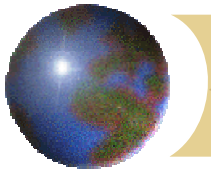
- ❖ **ENUM('valor1','valor2',...)**
- ❖ Es una cadena de caracteres que contiene uno solo de los valores de la lista (valor1, valor2, etc. etc.).



CREAR UNA TABLA EN MYSQL USANDO FUNCIONES PHP

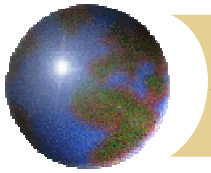
**Los posibles modos de crear tablas
son los siguientes:**

- ❖ `CREATE TABLE IF NOT EXISTS (nombre_campo1
parametros_campo1, nombre_campo2
parametros_campo2,... nombre)`
- ❖ `CREATE TABLE (nombre_campo1
parametros_campo1, nombre_campo2
parametros_campo2,... nombre)`



CREAR UNA TABLA EN MYSQL USANDO FUNCIONES PHP

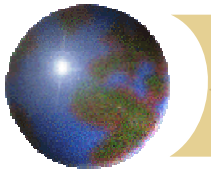
- ⊕ La única diferencia entre ambas opciones es la conocida comprobación/no comprobación de la preexistencia y el devolver/no devolver error en tal caso



CREAR UNA TABLA EN MYSQL USANDO FUNCIONES PHP

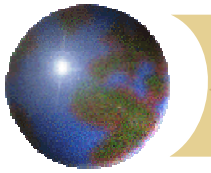
Detalle:

- ❖ **nombre_campoxx** es eso... el **nombre que quieres asignar a ese campo** y para el que normalmente *los expertos* suelen utilizar algo que aluda al contenido tal como: *fec_nac*, *nom_perro* y *lindezas* similares.



CREAR UNA TABLA EN MYSQL USANDO FUNCIONES PHP

parametros_campoxx. Por cada campo que crees tienes que asignarle algunos parámetros **obligatoriamente** algunos (**tipo de campo o tipo de campo y dimensión**) y otros **opcionales** que los entendidos llaman **flags**.

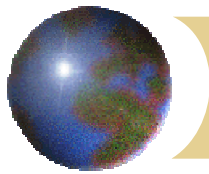


CREAR UNA TABLA EN MYSQL USANDO FUNCIONES PHP

Ejercicio1

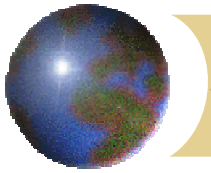
En el archivo: **ejercicio4.php**

- ❖ **Crear una base de datos llamada :
baseclase.**
- ❖ **Dentro de ella crear una tabla:
países.**
- ❖ **Con la estructura siguiente:**



CREAR UNA TABLA EN MYSQL USANDO FUNCIONES PHP

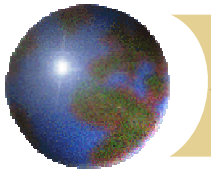
Name	Type	Null	Default
❖ codigo	tinyint(3) unsigned zerofill	No	
❖ nombre	char(40)	No	PERU
❖ poblacion	decimal(14,2) unsigned zerofill	No	
❖ desarrollado	enum('Si','No')	No	No
❖ continente	set('America','Africa','Europa','Asia')	Yes	



VER LA ESTRUCTURA DE UNA TABLA UTILIZANDO MYSQL

- Las sentencia MySQL que permiten visualizar la estructura de una tabla es esta:

SHOW FIELDS from *nombre de la tabla*



VER LA ESTRUCTURA DE UNA TABLA UTILIZANDO MYSQL

En el archivo: ejercicio5.php

⊕ En la base: baseclase.

⊕ En la tabla : paises.

Este ejercicio permite visualizar la estructura de una tabla, ubicando nombre de campo, tipo de dato, valores o propiedades, etc.



VER LA ESTRUCTURA DE UNA TABLA UTILIZANDO MYSQL

La tabla tiene 5 campos

Estructura de la tabla paises de la base de datos baseclase					
Nombre	Tipo	Null	Indice	Por Defecto	Extras
codigo	tinyint(3) unsigned zerofill	NO			
nombre	char(40)	NO		PERU	
poblacion	decimal(14,2) unsigned zerofill	NO			
desarrollado	enum('Si','No')	NO		No	
continente	set('America','Africa','Europa','Asia')	YES			



Ejercicio aplicado

- Crear la estructura siguiente:
- Base de datos: Inventario
- Tabla: Productos
- Estructura:

	Name	Type	Null	Default
	idproducto	tinyint(3) unsigned	No	0
	descripcion	char(40)	No	PROCESADOR
	cantUni	int(4) unsigned	Yes	100
	precio	float(6,2)	Yes	150.00
	idproveedor	tinyint(3) unsigned	No	0
	iddestino	tinyint(3) unsigned	No	0
	idresponsable	tinyint(3) unsigned zerofill	Yes	015