ACTIVIDAD 2.2 BAE

**Elabora una tabla como la que aparece en este PDF, pero con los tipos de datos de MySQL y comenta las principales diferencias que aprecies con SQL Server (tipos de datos comunes, tipos de datos faltantes, mayor o menor desglose y detalle, etc.)**



# INTRODUCCIÓN

Las bases de datos MySQL permiten que diferentes tipos de datos sean utilizados de distintas maneras. Los principales tipos de datos en MySQL son con **caracteres, numéricos y fechas y horas.**

# TIPOS DE DATOS DE MYSQL

# **Datos numéricos**

| **TIPOS DE DATOS** | **BYTES** | **Valor mínimo** | **Valor máximo** |
| --- | --- | --- | --- |
| TINYINT | 1 | -128 | 127 |
| SMALLINT | 2 | -32768 | 32767 |
| MEDIUMINT | 3 | -8388608 | 8388607 |
| INT o INTEGER | 4 | -2147483648 | 2147483647 |
| BIGINT | 8 | -9223372036854775808 | 9223372036854775807 |
| DECIMAL | M+2bytes si d>0,m+1 bytes si d=0 |  |  |
| FLOAT (X) | 4 u 8 bytes |  |  |
| FLOAT | 4 bytes |  |  |
| DOUBLE | 8 bytes |  |  |
| NUMERIC | M+2bytes si d>0,m+1 bytes si d=0 |  |  |

# **Los tipos de datos DECIMAL, FLOAT y DOUBLE en MySQL son tipos de datos de punto flotante o decimal, lo que significa que pueden almacenar valores con diferentes precisiones y escalas según cómo se definan al crear la columna en una tabla. Estos tipos de datos pueden considerarse "variables" en el sentido de que puedes ajustar la precisión y la escala según tus necesidades. Datos alfanuméricos** Para almacenar datos alfanuméricos (cadenas de caracteres) en MySQL poseemos los siguientes tipos de datos:

| **TIPO DE DATO** | **DESCRIPCIÓN** | **ALMACENAMIENTO** |
| --- | --- | --- |
| **CHAR** | permite almacenar textos breves, de hasta 255 caracteres de longitud como máximo | N bytes |
| **VARCHAR** | La longitud máxima permitida era de 255 caracteres hasta MySQL 5.0.3. pero desde esta versión cambió a un máximo de 65.535 caracteres. | N + 1 |
| **BINARY Y VARBINARY** | Estos dos tipos de datos son identicos a CHAR y VARCHAR, respectivamente, salvo que almacenan bytes en lugar de caracteres | Longitud+1 bytes |
| **TEXT** | Antes de la versión 5.0.3. de MySQL, este campo era el utilizado "por excelencia" para descripciones de productos, comentarios, textos de noticia, y cualquier otro texto largo.  Pero, a partir de la posibilidad de utilizar VARCHAR para longitudes de hasta 65.535 caracteres, es de esperar que se utilice cada vez menos este tipo de campo. | Longitud +2 bytes |
| **BLOB** | Permite almacenar datos de tipo BLOB (Binary Large Object). Admite una longitud máxima de 65,535 bytes de datos. | Longitud +2 bytes |
| **TINYBLOB** | 255 bytes (máximo) | Longitud+1 bytes |
| **MEDIUMBLOB** | 16.777.215 bytes (máximo) |  |
| **LONGBLOB** | 4 gb (o lo máximo que permita el sistema operativo) | Longitud+4 bytes |
| **ENUM** | permite establecer cuáles serán los valores posibles que se le podrán insertar.Es decir, crearemos una lista de valores permitidos, y no se autorizará el ingreso de ningún valor fuera de la lista, y se permitirá elegir solo uno de estos datos como valor del campo | 1 ó dos bytes dependiendo del número de valores |
| **SET** | una lista de valores entre comillas simples, separados por comas. Pero, a diferencia de ENUM, sí podemos llegar a dejarlo vacío, sin elegir ninguna opción de las posibles. | 1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores |

**Datos de fecha y hora:**

| **TIPO DE DATOS** | **DESCRIPCIÓN** | **ALMACENAMIENTO** |
| --- | --- | --- |
| **DATE** | Almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. | El formato de almacenamiento es de año-mes-dia |
| **DATETIME** | Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. | El formato de almacenamiento es de año-mes-dia horas:minutos:segundos |
| **TIMESTAMP** | Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo: | El formato de almacenamiento depende del tamaño del campo: |
| **TIME** | Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. | El formato de almacenamiento es de 'HH:MM:SS' |
| **YEAR** | Almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. | El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos. |

# **DIFERENCIAS SQL SERVER Y MYSQL**

**Entorno**

SQL Server funciona mejor con .NET, mientras que MySQL se puede emparejar con casi cualquier otro idioma, generalmente con PHP

**Sintaxis**

Para la mayoría de las personas, esta es la diferencia más importante entre los dos sistemas. Estar familiarizado con un conjunto de reglas de sintaxis sobre otro puede influir enormemente en su decisión en cuanto a qué sistema le conviene más.

**Zonas horarias**

SQL Server tiene un tipo de dato DATETIMEOFFSET que incluye información de zona horaria, mientras que MySQL utiliza TIMESTAMP sin zona horaria y requiere que la gestión de zonas horarias se realice a nivel de la aplicación.

# **Tipo de dato ENUM:** MySQL ofrece el tipo de dato ENUM, que permite definir un conjunto de valores permitidos para una columna.

**Tipos de datos de cadena:**

**MySQL** ofrece el tipo de dato VARCHAR para almacenar cadenas de longitud variable, mientras que **SQL Server** utiliza VARCHAR o NVARCHAR para soportar cadenas Unicode.

SQL Server tiene el tipo de dato TEXT para almacenar datos de texto grandes, mientras que MySQL utiliza TEXT o LONGTEXT.

**Tipos de datos numéricos:**

**MySQL**:tiene un tipo de dato TINYINT UNSIGNED que no tiene un equivalente exacto en SQL Server.

**SQL Serve**r:ofrece el tipo de dato MONEY para representar valores monetarios, mientras que MySQL generalmente utiliza DECIMAL o DOUBLE para este propósito.

**MEDIUMINT no existe en SQL SERVER**

**Tipos de datos de fecha y hora:**

Ambas bases de datos tienen tipos de datos para manejar fechas y horas. MySQL utiliza DATE, DATETIME y TIMESTAMP, mientras que SQL Server utiliza DATE, DATETIME y DATETIME2.