

Corporación universitaria UNIMINUTO de Dios

investigación de cadenas y gestores de bases de datos

Juan Carlos Farfán Riaño

1011457

Fundamentos de programación

Héctor Andrés ladino torres

73963

11/04/2025

1. Cuantas Cadenas de Caracteres Existen en los motores de Base Datos

Los tipos de cadenas de caracteres son CHAR, VARCHAR, BINARY, VARBINARY, BLOB, TEXT, ENUM, y SET:

CHAR: es un tipo de dato que representa un carácter o parte de un carácter. Se puede usar para contener cadenas de caracteres en programación y en bases de datos.

VARCHAR: es un tipo de dato que almacena cadenas de caracteres de longitud variable. Se utiliza para guardar datos que pueden incluir letras, números y símbolos

BINARY: Una cadena binaria es una secuencia de 0 y 1 que representa datos binarios, no caracteres.

VARBINARY: es un tipo de dato que almacena cadenas de bytes binarios, en lugar de cadenas de caracteres no binarios. Es similar al tipo VARCHAR, pero se basa en valores numéricos de bytes en lugar de caracteres.

TEXT: Este tipo se utiliza para textos largos, como descripciones o comentarios. Los motores de bases de datos suelen tener variantes como TINYTEXT, MEDIUMTEXT y LONGTEXT, que varían en la máxima cantidad de texto que pueden almacenar.

BLOB: Aunque se utiliza principalmente para datos binarios, BLOB también puede almacenar grandes cantidades de texto. Al igual que TEXT, tiene variantes como TINYBLOB, MEDIUMBLOB y LONGBLOB.

NCHAR Y NVARCHAR: Son similares a CHAR y VARCHAR, pero están diseñados para almacenar caracteres Unicode. Esto permite representar una gama más amplia de caracteres de diferentes idiomas, lo que es esencial para aplicaciones multilingües.

2.cuantos gestores de Base datos existen y haga un cuadro comparativo entre ellas

características	MySQL	MariaDB	PostgreSQL	MongoDB	SQLite
Modelo de datos	Relacional (SQL)	Relacional (SQL)	Relacional (SQL)	NoSQL (Documental)	Relacional (SQL)
Rendimiento	Bueno para lectura, moderado para escritura.	Optimizado para rendimiento, mejora a MySQL	Alto rendimiento en consultas complejas	Alto rendimiento en grandes volúmenes de datos no estructurados	Rápido para operaciones locales
Potencia	Amplia funcionalidad, maduro	Mejoras y nuevas funcionalidades sobre MySQL	Muy potente, soporta tipos de datos avanzados.	Flexible, ideal para datos no estructurados.	Ligero, ideal para aplicaciones embebidas
Función	Aplicaciones web, OLTP	Reemplazo directo de MySQL, OLTP, análisis	OLTP, OLAP, sistemas de información geográfica	Aplicaciones web modernas, Big Data, IoT	Almacenamiento local de datos en aplicaciones
Escalabilidad	Escalabilidad vertical limitada	Mejor escalabilidad vertical que MySQL	Alta escalabilidad vertical y horizontal	Alta escalabilidad horizontal	No está diseñado para escalabilidad
Ventajas	Amplia comunidad, madurez, compatibilidad	Compatibilidad con MySQL, mejoras de rendimiento	Potencia, integridad de datos, extensibilidad	Flexibilidad, escalabilidad, manejo de datos no estructurados.	Ligero, fácil de usar, sin servidor.
Desventajas	Desarrollo más lento, algunas limitaciones de escalabilidad	Menor adopción que MySQL	Curva de aprendizaje empinada, mayor consumo de recursos	Menor madurez que los SGBD relacionales, menor soporte para transacciones ACID	Limitaciones de concurrencia, no apto para aplicaciones de alto tráfico

