

Investigación de Fundamentos de Programación Orientada a Objetos

Alejandro Vargas García

Tecnología en Informática, Facultad de Ingeniería, Corporación Universitaria Minuto de Dios

NRC-10-73963: Fundamentos de Programación Orientada a Objetos

Profe. HECTOR ANDRES LADINO TORRES

11 de abril de 2025

¿Qué Debo Hacer?

Hacer la siguiente investigación:

- I. ¿Cuántas Cadenas de Caracteres Existen en los motores de Base Datos?
- II. ¿Cuántos gestores de Base datos existen? y haga un cuadro comparativo entre ellas.

Desarrollo

¿Cuántas Cadenas de Caracteres Existen en los Motores de Base Datos?:

Si hablamos de los motores de bases de datos, las cadenas de caracteres son unos de los tipos de datos que más se utilizan, ya que permiten abarcar y almacenar varios textos, símbolos, números y más. Estos tipos de datos cambian según el motor (MySQL, PostgreSQL, Oracle, SQL Server, entre otros), cada uno con diferentes cadenas de caracteres, cada una con sus propias características y funcionalidades. Aunque pensemos que deben de ser muy complicado diferenciar entre ellas, muchas veces solo les cambia el nombre según la plataforma, todo esto sumado, que las cadenas de caracteres están divididas en grupos, los grandes grupos, que pueden ser de longitud fija y de longitud variable, y pueden tener versiones especiales para Unicode o para texto extenso.

Así que, podemos definir que las cadenas de caracteres son las siguientes:

1. **CHAR(n) - Cadena de longitud fija:**

Este tipo denominado CHAR almacena una cadena de longitud fija, en la cual n representa la cantidad de caracteres. Además, entendemos que, si se dejan caracteres de n sin rellenar, estos espacios se rellenan en espacios en blanco.

Usos: útil para campos donde todos los valores tienen el mismo tamaño.

Como ventajas podemos destacar que es muy rápido para comparaciones y además que es ideal para valores predecibles. Como desventajas podemos determinar una, que es que ocupa espacio innecesario si las cadenas no cumplen con el espacio estipulado, ósea, que son muy cortas.

Dato: El tamaño máximo de la cadena de caracteres CHAR varía según su motor, por ejemplo, en el caso de MySQL, que permite hasta 255 caracteres; en SQL server, que permite hasta 8000 bytes y en Oracle, que permite 2000 bytes.

2. **VARCHAR(n) - Cadena de longitud variable:**

VARCHAR puede almacenar textos de longitud variable, hasta un máximo de n caracteres. Se puede diferenciar con CHAR, ya que esta no desperdicia valores si la cadena es corta.

Como ventajas podemos destacar que ahorra mucho el espacio y que además es flexible a datos de longitud variable. Como desventajas no encontramos ninguna, ya que es un tipo de dato muy viable y confiable.

Dato: El tamaño máximo de la cadena de caracteres VARCHAR puede variar según con que estemos trabajando, por ejemplo, en el caso de MySQL, que permite hasta hasta 65,535 bytes; en SQL server, que permite hasta 8,000 bytes o VARCHAR(MAX) hasta 2GB y en Oracle, que permite en VARCHAR2 hasta 4,000 caracteres

3. TEXT, TINYTEXT, MEDIUMTEXT, LONGTEXT - Texto largo (MySQL)

Estas cadenas de caracteres se usan para almacenar grandes volúmenes de texto. No se recomienda para búsquedas frecuentes o índices y además tiene limitaciones, como el hecho de que no se puede indexar correctamente y además que tiene un menor rendimiento a comparación de la cadena de caracteres VARCHAR. Cada uno tienen un máximo y ese es en el caso de TINYTEXT: 255 bytes, el del TEXT: 65,535 bytes, el del MEDIUMTEXT: 16 MB y el del LONGTEXT: 4 GB.

Como usos podemos destacar a los Comentarios, a las descripciones, a los blogs, a los cuerpos de correo y a los documentos.

4. NCHAR(n) y NVARCHAR(n) – Cadenas Unicode

Tienen versiones que son compatibles con Unicode de CHAR y VARCHAR. Utilizan más espacio por carácter que generalmente puede ser de 2 bytes o más.

Usos: Te pueden servir cuando necesites almacenar idiomas como chino, japonés, árabe, emojis, etc.

Dato: El tamaño máximo de estas cadenas de caracteres varía según su motor, por ejemplo, en el caso de SQL Server, NCHAR puede almacenar hasta 4,000 caracteres y en cambio NVARCHAR(MAX) puede hasta 2GB de texto.

5. CLOB (Objeto Largo)

Este es un tipo especial de objeto de texto grande, usado principalmente en Oracle, DB2 y PostgreSQL.

Usos: Puede ser utilizado para almacenar grandes volúmenes de texto donde necesitas funciones LOB (como streaming de texto). Cuenta con capacidad máxima en Oracle de hasta 128 TB.

6. VARCHAR2 (Oracle)

Este es un tipo exclusivo de Oracle, similar a VARCHAR pero con algunas reglas en su interior que garantizan la compatibilidad futura y un manejo de espacio más eficiente.

Dato: Oracle promete y recomienda trabajar VARCHAR2 en vez de VARCHAR, así el trabajo va a ser más impecable. Además de que cuenta con un espacio máximo de unos 4,000 caracteres y si le aplicamos una configuración especial, puede llegar hasta los 32,767.

7. JSON y XML (no son tan comunes, pero pueden almacenar texto)

JSON: Este almacena datos en formato de texto JSON. Se suele usar en PostgreSQL (json y jsonb) y MySQL (JSON).

XML: Este es un tipo especial para documentos XML en Oracle y SQL Server.

¿Cuántos gestores de Base datos existen? y haga un cuadro comparativo entre ellas?

Hay decenas de gestores de bases de datos (también conocidos como *Database Management Systems* o DBMS) disponibles, tanto comerciales como open source. Hasta ahora, más o menos hay 424 gestores de bases de datos diferentes registrados oficialmente, según la página de DB-Engines (que es como un ranking de estas cosas). Ahí agrupan todos: los más conocidos como MySQL, PostgreSQL, SQL Server, y otros que casi nadie usa o que son más de nicho.

Estos se pueden dividir en tipos, los cuales son:

1. **Relacionales (RDBMS):** usan SQL, tablas, filas y columnas.
 - Ej: MySQL, PostgreSQL, Oracle, SQL Server, MariaDB.
2. **No Relacionales (NoSQL):** orientados a documentos, clave-valor, grafos, columnas.
 - Ej: MongoDB, Redis, Cassandra, Neo4j.
3. **NewSQL:** intentan combinar lo mejor de relacionales y NoSQL.
 - Ej: Google Spanner, CockroachDB.
4. **Multimodelo:** soportan más de un tipo de modelo de datos.
 - Ej: ArangoDB, OrientDB.

Todos los ejemplos que ya dimos se van a utilizar para crear el cuadro comparativo, como son 424, no puedo colocarlos completos, entonces decidí incluir los más importantes.

Figura 1

Cuadro de Revisión, para ver efectivamente los gestores que vamos a comparar.

Tipo de Gestor de Base de Datos	Ejemplos
Relacionales (RDBMS)	MySQL, PostgreSQL, Oracle, SQL Server, MariaDB
No Relacionales (NoSQL)	MongoDB, Redis, Cassandra, Neo4j
NewSQL	Google Spanner, CockroachDB
Multimodelo	ArangoDB, OrientDB

Nota: Este cuadro fue hecho por mí, aunque, la información en este cuadro fue tomada de <https://www.ibm.com/cloud/learn/database> y de <https://learn.microsoft.com/es-es/azure/architecture/data-guide/technology-choices/data-storage>.

Figura 2

Cuadro Comparativo de los Gestores de Datos

Gestor de Base de Datos	Modelo de Datos	Lenguaje Principal	Ventajas	Desventajas
MySQL	Relacional (tablas)	SQL	Muy popular, gratuito, multiplataforma	No ideal para big data
PostgreSQL	Relacional (tablas)	SQL	Potente, open source, soporta funciones avanzadas	Curva de aprendizaje más alta
Oracle	Relacional (tablas)	SQL / PL-SQL	Robusto, ideal para grandes empresas	Licencia costosa
SQL Server	Relacional (tablas)	T-SQL	Integración con Microsoft, buena seguridad	Solo en Windows / Azure
MariaDB	Relacional (tablas)	SQL	Rápido, seguro, gratuito, basado en MySQL	Menor soporte empresarial
MongoDB	No relacional (documentos)	JSON / BSON	Flexible, ideal para datos semi estructurados	No usa transacciones completas como SQL
Redis	No relacional (clave-valor)	Propio	Muy rápido, ideal para caché	No apto para datos complejos
Cassandra	No relacional (columnas)	CQL (tipo SQL)	Escalable, resistente a fallos	Difícil de configurar
Neo4j	No relacional (grafos)	Cypher	Potente en relaciones complejas	No sirve para tablas tradicionales
Google Spanner	NewSQL	SQL	Escalabilidad global, consistencia fuerte	Solo en Google Cloud
CockroachDB	NewSQL	SQL	Resistente a fallos, fácil de escalar	Comunidad aún en crecimiento
ArangoDB	Multimodelo	AQL / JSON / Gremlin	Soporta grafos, documentos, clave-valor	Complejo de administrar
OrientDB	Multimodelo	SQL / Gremlin / JSON	Multiuso, soporta varios modelos	Poco popular, documentación limitada

Nota: Este cuadro fue hecho por mí, aunque, la información en este cuadro fue tomada de <https://www.ibm.com/cloud/learn/database> y de <https://learn.microsoft.com/es-es/azure/architecture/data-guide/technology-choices/data-storage>.

Dato importante: No podemos decir con certeza la cantidad que existen, aunque hasta la hora tenemos 424.

Referencias

- DB-Engines. (n.d.). DBMS popularity ranking.
<https://db-engines.com/en/ranking>
- GeeksforGeeks. (n.d.). SQL Data Types.
<https://www.geeksforgeeks.org/sql-data-types/>
- IBM. (2023). Character large object (CLOB). IBM Documentation.
<https://www.ibm.com/docs/en/db2/11.5?topic=types-character-large-object-clob>
- Microsoft. (n.d.). Data types (Transact-SQL). Microsoft Learn.
<https://learn.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql>
- Microsoft. (n.d.). nchar and nvarchar (Transact-SQL). Microsoft Learn.
<https://learn.microsoft.com/en-us/sql/t-sql/data-types/nchar-and-nvarchar-transact-sql>
- MySQL. (n.d.). String Types. MySQL Documentation.
<https://dev.mysql.com/doc/refman/8.0/en/string-types.html>
- Oracle. (2023). CLOB Data Type. Oracle Database Documentation.
<https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/CLOB-Data-Type.html>
- Oracle. (2023). Data Types. Oracle Database Documentation.
<https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/Data-Types.html>
- PostgreSQL Global Development Group. (n.d.). Character Types. PostgreSQL Documentation.
<https://www.postgresql.org/docs/current/datatype-character.html>
- W3Schools. (n.d.). SQL Data Types.
https://www.w3schools.com/sql/sql_datatypes.asp