

## **Fundamentos de Bases de datos**

### **Consulta #1 Bases de datos sql y nosql (propietarios y opensource)**

**Nombre: Andrés Felipe Merino Bravo**

**Fecha: 02/10/2025**

**Consultar los diferentes motores de bases de datos que están disponibles, tanto propietarios como de software libre. características más importantes como: plataforma, memoria, almacenamiento, seguridad.**

### **Bases de datos SQL**

- **Propietarios**

- o **SQL Server**

- Plataforma: Linux, Windows, MacOS
    - Memoria: Buffer pool, configuraciones como “max server memory”. Utiliza AWE/Loc Pages in Memory para fijar memoria física
    - Almacenamiento: Almacenamiento en disco con archivos de datos, logs de transacción, índices (clustered, nonclustered, columnstore)
    - Seguridad: Autenticación (Windows / SQL), roles de servidor, cifrado de datos transparente (TDE) para datos en reposo, conexiones TLS/SSL para el transporte.

- o **Oracle Database**

- Plataforma: Linux, Windows, MacOS
    - Memoria: SGA (System Global Area, compartida) y PGA (Parliamentarians for Global Action).
    - Almacenamiento: Archivos de datos, redo logs, control files, ASM (Automatic Storage Management) para manejar discos/disk groups y redundancia.
    - Seguridad: Cifrado en reposo y en tránsito, control de acceso granular, auditoría, masking, monitoreo de actividad.

- o **SAP ASE**

- Plataforma: Windows, Linux y entornos de nubes
    - Memoria: cachés internos, buffers configurables, cuenta con soporte in-memory (IMRS cache) para mejorar concurrencia.
    - Almacenamiento: Archivos en disco para persistencia: logs, datos, índices; estructuras tradicionales de RDBMS.
    - Seguridad: cifrado de comunicaciones (SSL), encriptación de columna o base de datos completa, autenticación externa (AD/LDAP/Kerberos)

- **Open Source**

- o **MySQL**

- Plataforma: Linux, Windows, macOS

- Memoria: Buffers internos como InnoDB buffer pool para caché de datos e índices; también puede usar motor MEMORY para tablas temporales en memoria.

- Almacenamiento: InnoDB, MyISAM, journaling, índices de disco

- Seguridad: TLS/SSL para conexiones, opciones de cifrado de datos en reposo, auditoría mediante plugins.

- o **PostgreSQL**

- Plataforma: Linux, Windows, macOS

- Memoria: Buffers compartidos, cachés de sistema, trabajo en memoria para operaciones (ordenamientos, joins).

- Almacenamiento: WAL (Write Ahead Log), archivos de datos, índices, soporte para JSON/JSONB

- Seguridad: autenticación (md5, scram, LDAP, etc.), cifrado TLS, control de acceso a nivel de roles, extensiones para cifrado en reposo

- o **SQLite**

- Plataforma: Windows, Linux, macOS

- Memoria: Cachés de página en memoria, toda operación trabaja sobre el archivo local, no hay servidor dedicado.

- Almacenamiento: Único archivo de base de datos en disco, no es cliente-servidor, no hay archivos de log separados en muchas configuraciones estándar.

- Seguridad: Sin sistema de autenticación por defecto, normalmente se utilizan extensiones (SQLCipher)

## **Bases de datos no SQL**

- **Propietarios**

- o **Amazon DynamoDB**

- Plataforma: AWS (nube).

- Memoria: AWS gestiona internamente la memoria, cachés y optimizaciones, entonces el usuario no controla directamente estos detalles.

- Almacenamiento: Datos persistidos en unidades de almacenamiento distribuidas dentro de AWS, redundancias internas, índices secundarios, backups automáticos.

- Seguridad: AWS KMS, cifrado en tránsito (TLS), control de acceso mediante IAM, aislamiento de red (VPC, endpoints).

- **Microsoft Azure Cosmos DB**

- Plataforma: Azure

- Memoria: Memoria gestionada por el servicio.

- Almacenamiento: Particiones lógicas y físicas, particiones físicas de hasta 50GB.

- Seguridad: Cifrado en reposo y en tránsito, control de acceso basado en roles, SLA de latencia y disponibilidad, APIs de seguridad integradas.

- **Google Cloud Firestore**

- Plataforma: Google Cloud

- Memoria: Internamente gestionada por Google, los detalles de caché o memoria no son visibles para el usuario.

- Almacenamiento: Documentos JSON distribuidos, persistencia automática, replicación interna para durabilidad y disponibilidad.

- Seguridad: IAM para permisos, TLS en tránsito.

- **Open Source**

- **MongoDB**

- Plataforma: Linux, Windows, macOS, Dockers.

- Memoria: Caché de índices y datos, puedes configurar el tamaño del caché en el motor WiredTiger

- Almacenamiento: En disco (WiredTiger), con journaling, compresión, índices

- Seguridad: autenticación (SCRAM, LDAP, etc.), cifrado TLS para conexiones, control de roles, auditoría.

- **Redis**

- Plataforma: Linux, Windows, macOS, dockers
- Memoria: Opera en memoria, definición de límites de memoria y políticas de expulsión si se llena la memoria.
- Almacenamiento: Persistencia opcional (snapshot RDB, append-only file AOF), si la persistencia está desactivada, los datos sólo viven en memoria.
- Seguridad: Autenticación con contraseña, TLS para conexiones, ACLs (listas de control de acceso).

- **Apache Cassandra**

- Plataforma: Entorno JVM, comúnmente sobre Linux, desplegable en clusters distribuidos.
- Memoria: Memtables en memoria (heap de Java, caches, bloom filters, etc.).
- Almacenamiento: Commit log en disco, SSTables en disco con compaction, estructuras LSM, persistencia en disco.
- Seguridad: Autenticación y autorización, cifrado TLS para comunicación cliente-nodo y nodo-nodo, cifrado en disco configurable.

- **Apache CouchDB**

- Plataforma: Erlang/OTP, ejecutable en: Linux, Windows, macOS.
- Memoria: Utiliza memoria para caches, índices, vistas; operaciones intermedias requieren memoria.
- Almacenamiento: Utiliza archivos en disco para persistencia con versiones antiguas, compactación, vistas indexadas.
- Seguridad: Autenticación de usuarios (admin, roles), cifrado TLS para conexiones, control de permisos por base de datos, validaciones con funciones de seguridad.

## Bibliografía

1. Microsoft. (s. f.). Memory management architecture guide. SQL Server Documentation. <https://learn.microsoft.com/en-us/sql/relational-databases/memory-management-architecture-guide?view=sql-server-ver17>

2. Microsoft. (s. f.). Monitor Memory Usage. SQL Server Documentation.  
<https://learn.microsoft.com/en-us/sql/relational-databases/performance-monitor/monitor-memory-usage?view=sql-server-ver17>
3. PostgreSQL Global Development Group. (s. f.). Chapter 45.3 Memory Management — SPI Memory. PostgreSQL Documentation.  
<https://www.postgresql.org/docs/current/spi-memory.html>
4. PostgreSQL Global Development Group. (s. f.). Chapter 65. Database Physical Storage. PostgreSQL Documentation.  
<https://www.postgresql.org/docs/current/storage.html>
5. Severalnines. (s. f.). Architecture and Tuning of Memory in PostgreSQL Databases. Severalnines. <https://severalnines.com/blog/architecture-and-tuning-memory-postgresql-databases/>
6. Suzuki, H. (s. f.). Process and Memory Architecture – PostgreSQL. InterDB. <https://www.interdb.jp/pgpgsql02.html>