**Caso de análisis: SQLite como Sistema Manejador de Bases de Datos**

**1. Funcionalidad de SQLite**

SQLite es un motor de base de datos relacional de tipo embebido, diseñado para integrarse dentro de programas y dispositivos sin necesidad de un servidor independiente. A diferencia de sistemas cliente-servidor como PostgreSQL o SQL Server, este software funciona mediante una biblioteca que gestiona la información en un único archivo local.  
Sus principales características incluyen:  
- Autocontenido: toda la base de datos se encuentra en un archivo.  
- Cumplimiento de transacciones ACID.  
- Portabilidad: puede utilizarse en navegadores, teléfonos inteligentes, aplicaciones de escritorio o dispositivos IoT.  
- Implementación ligera que requiere muy pocos recursos.  
Esto lo convierte en una alternativa muy usada en contextos donde la simplicidad y la velocidad son prioritarias frente a la escalabilidad.

**2. Ventajas y desventajas en comparación con otros SMBD**

Ventajas:  
- Reducción de redundancia: al centralizar datos en un solo archivo, se evita duplicación innecesaria.  
- Consistencia: las propiedades ACID garantizan que la información se mantenga coherente aun en fallos o interrupciones.  
- Portabilidad y simplicidad: basta con copiar el archivo para migrar o respaldar la base.  
- Bajo costo: es de código abierto y no exige licencias.  
- Mantenimiento sencillo: no requiere administradores especializados.  
- Velocidad en entornos pequeños: su ejecución embebida lo hace más rápido que otros motores en proyectos de tamaño reducido.  
  
Desventajas:  
- Concurrencia restringida: varios procesos pueden leer, pero las escrituras simultáneas generan bloqueos.  
- Seguridad limitada: carece de gestión avanzada de usuarios o roles; depende del sistema operativo.  
- Escalabilidad baja: no es recomendable para bases de gran volumen o sistemas distribuidos.  
- Estandarización parcial: aunque respeta gran parte de SQL, no incluye todas las funcionalidades avanzadas de otros motores.  
- No apto para corporativos: en entornos críticos con múltiples usuarios concurrentes, no ofrece el mismo rendimiento que sistemas robustos como Oracle o MySQL.

**3. Funciones de un SMBD en SQLite**

Funciones que sí incluye:  
- Operaciones básicas de almacenamiento, consulta y actualización.  
- Transacciones con propiedades ACID.  
- Mecanismos de recuperación mediante journaling o WAL.  
- Restricciones de integridad como claves primarias y foráneas.  
  
Funciones limitadas o ausentes:  
- Catálogo de metadatos complejo.  
- Seguridad y autorización a nivel de usuario o rol.  
- Alta concurrencia en sistemas multiusuario.  
- Herramientas avanzadas de replicación o respaldo.

**4. SQLite y la arquitectura ANSI-SPARC**

La arquitectura de tres niveles establece:  
1. Externo: cómo los usuarios ven la información.  
2. Conceptual: el modelo lógico.  
3. Interno: la representación física.

SQLite ofrece un cumplimiento parcial:  
- Soporta vistas y consultas para la capa externa.  
- Permite la definición de tablas, índices y restricciones en el nivel conceptual.  
- Gestiona el almacenamiento físico en un archivo único.  
  
No obstante, al estar diseñado para ser ligero, carece de la sofisticación que otros motores aplican en cada capa.

**5. Conclusión**

SQLite puede clasificarse como un sistema manejador de bases de datos legítimo, aunque con un enfoque especializado que lo distingue de sus contrapartes cliente-servidor. Su diseño embebido, su adherencia a los principios ACID y su capacidad para gestionar datos de manera autónoma lo posicionan como una opción válida dentro del espectro de los sistemas de gestión de bases de datos.

Su mayor fortaleza radica en entornos donde la simplicidad, el bajo consumo de recursos y la portabilidad son críticos, como en aplicaciones móviles, dispositivos IoT, software educativo o prototipos iniciales. No obstante, es importante reconocer sus limitaciones en cuanto a concurrencia, seguridad integrada y escalabilidad, aspectos donde sistemas como MySQL, PostgreSQL u Oracle conservan una clara ventaja.

Por lo tanto, SQLite no debe verse como un reemplazo para los sistemas de bases de datos tradicionales, sino como una herramienta complementaria ideal para casos de uso específicos donde su arquitectura ligera y autónoma ofrece ventajas decisivas. La elección de SQLite versus otros sistemas dependerá siempre de los requisitos particulares del proyecto en cuestión.