**Caso de estudio de un SMBD: SQLite**

**1. Definición y funcionalidad de SQLite**

SQLite es un sistema de gestión de bases de datos relacional (RDBMS) ligero, autónomo y embebido. A diferencia de otros motores como MySQL o PostgreSQL, no funciona bajo un modelo cliente-servidor, sino que se integra directamente en las aplicaciones. Los datos se almacenan en un solo archivo en disco, lo cual facilita su portabilidad y simplicidad.

**Características principales:**

* No requiere configuración ni instalación de un servidor.
* Toda la base de datos se guarda en un único archivo.
* Cumple con gran parte del estándar SQL-92.
* Ofrece soporte a transacciones ACID (atómicas, consistentes, aisladas y duraderas).
* Es ampliamente utilizada en aplicaciones móviles, navegadores web y sistemas embebidos.

**2. Ventajas y desventajas de SQLite respecto a los SMBD**

**Ventajas:**

* **Control de redundancia de los datos:** al centralizarse en un único archivo, se reduce duplicidad.
* **Coherencia de los datos:** garantiza integridad con transacciones ACID.
* **Compartición de los datos:** el archivo puede copiarse y compartirse fácilmente.
* **Mayor integridad:** mantiene datos consistentes.
* **Economía de escala:** gratuito, de código abierto y sin necesidad de licencias.
* **Accesibilidad y respuesta:** rápido para aplicaciones pequeñas y medianas.
* **Mantenimiento sencillo:** no requiere un administrador especializado.
* **Mayor portabilidad:** se puede usar en móviles, navegadores o software ligero.

**Desventajas:**

* **Concurrencia limitada:** no soporta alta cantidad de usuarios accediendo al mismo tiempo como otros SMBD.
* **Seguridad básica:** no incluye usuarios ni roles avanzados, la seguridad depende del sistema operativo.
* **Escalabilidad reducida:** no es ideal para bases de datos muy grandes (más de 1 TB) o sistemas distribuidos.
* **Imposición de estándares:** aunque sigue SQL, carece de algunas funciones avanzadas de otros motores.
* **Menor productividad en grandes empresas:** no está diseñado para sistemas corporativos de misión crítica.

**3. Funciones de un SMBD presentes y ausentes en SQLite**

**Funciones presentes:**

* Almacenamiento, recuperación y actualización de datos.
* Soporte de transacciones (ACID).
* Servicios básicos de recuperación ante fallos.
* Servicios de control de concurrencia (a nivel de archivo/bloqueo).
* Servicios de integridad (restricciones primarias, foráneas, etc.).

**Funciones ausentes o limitadas:**

* Catálogo avanzado de metadatos accesible al usuario.
* Servicios robustos de recuperación en entornos distribuidos.
* Servicios de autorización detallada (manejo de usuarios y roles).
* Escalabilidad y concurrencia masiva.

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

La arquitectura ANSI-SPARC propone **tres niveles**:

1. **Nivel externo:** cómo los usuarios ven los datos.
2. **Nivel conceptual:** el modelo lógico de la base.
3. **Nivel interno:** cómo se almacenan físicamente los datos.

**SQLite cumple parcialmente con la arquitectura ANSI-SPARC.**

* Permite definir esquemas (nivel conceptual).
* Los usuarios acceden con vistas y consultas SQL (nivel externo).
* Maneja el almacenamiento en archivo único (nivel interno).  
  Sin embargo, sus mecanismos son simplificados y no tan sofisticados como en otros SMBD completos.

**5. Conclusiones**

SQLite **sí puede clasificarse como un sistema manejador de bases de datos** porque cumple las funciones esenciales de almacenamiento, recuperación, actualización de datos y soporte de transacciones.  
Es una solución práctica, ligera y eficiente en escenarios donde la base de datos no requiere una alta concurrencia ni escalabilidad. Por ello, es ideal para aplicaciones móviles, prototipos y software de escritorio.  
Sin embargo, para sistemas empresariales de gran escala o con múltiples usuarios concurrentes, conviene utilizar motores más robustos como MySQL, PostgreSQL o SQL Server.