

Gestión y Configuración de Bases de Datos SQLite

Eres el (DBA) administrador de una base de datos SQLite para una pequeña aplicación de gestión de frutas en dispositivos móviles. Tendrás que configurar correctamente la base de datos, optimizar su rendimiento, realizar backups/restauraciones y garantizar la seguridad de los datos almacenados.

Finalidad:

- 1. Gestionar y configurar el entorno de la base de datos SQLite.
- 2. Optimizar el rendimiento de la base de datos a través de diferentes técnicas de ajuste.
- 3. Implementar copias de seguridad y restauración.
- 4. Aplicar técnicas de seguridad para proteger los datos sensibles.

Requisitos del entorno:

- 5. **Instalación de SQLite:** Asegúrate de tener SQLite instalado en tu entorno de trabajo. Puedes verificarlo con `sqlite3 --version`
- 6. Crea una base de datos llamada "frutitas.db" que almacenará la información de mandarinas, melocotones y caquis.

1. Configuración de la base de datos (2 puntos)

1.1. Creación de tablas y estructura

Crea las siguientes tablas básicas con las restricciones adecuadas:

- **Mandarinas:**
 - id_mandarina (INTEGER, PK, AUTOINCREMENT): Identificador único para cada mandarina.
 - color (TEXT, NOT NULL): Color de la mandarina.
 - tipo (TEXT, NOT NULL): Tipo de mandarina.
 - size (TEXT, NOT NULL): Tamaño de la mandarina.
 - fecha_recogida (DATE, NOT NULL): Fecha en la que se recogió la mandarina.
- **Melocotones:**
 - id_melocoton (INTEGER, PK, AUTOINCREMENT): Identificador único para cada melocotón.
 - tipo (TEXT, NOT NULL): Tipo de melocotón.
 - suave (BOOLEAN, NOT NULL): Indica si el melocotón es suave (1) o no (0).
- **Caquis:**
 - id_caqui (INTEGER, PK, AUTOINCREMENT): Identificador único para cada caqui.
 - id_mandarina (INTEGER, FK, NOT NULL): Relacionado con id_mandarina en la tabla mandarinas.
 - id_melocoton (INTEGER, FK, NOT NULL): Relacionado con id_melocoton en la tabla melocotones.
 - color (TEXT, NOT NULL): Color del caqui.
 - pedunculo (BOOLEAN, NOT NULL): Indica si el caqui tiene pedúnculo (1) o no (0).
 - tiempo_maduracion (INTEGER, NOT NULL): Tiempo de maduración en días.

1.2. TAMAÑO Y MÁS...

Justifica cuanto ocupa tu base de datos en disco. ¿En qué momento aumenta de tamaño si se mete más info? ¿como DBA harías algo para optimizar esto?

2. Optimización del rendimiento y mantenimiento (5 puntos)

Se ha encontrado un trozo de log de consultas a la base de datos durante el día 27 de octubre de 2024. Se pide que optimizaciones realizarías y por qué para mejorar la base de datos en función de la información que te ofrece el log y justificalo.

PISTA: UTILIZA EXPLAIN QUERY PLAN Y CHEQUEA TODOS LOS PRAGMAS DE SQLITE.

3. Copias de seguridad y restauración (2.5 puntos)

- 3.1. Realización de una copia de seguridad con el nombre frutitas_bonitas_bonitas.db
- 3.2. Realiza la restauración desde la copia de seguridad
- 3.3. Automatización de backups

Crea un script para backups automáticos y que ejecute todos los días a las 23:59:59

4. Preguntas sin Chatgpt ni IAs: (0.5 puntos)

- ¿Cómo impactan los cambios en el PRAGMA en el rendimiento y la seguridad de la base de datos?
- ¿Qué mecanismos adicionales usarías para proteger una base de datos SQLite en un entorno de producción?
- ¿Qué diferencia existe entre los modos de journaling como DELETE, TRUNCATE y WAL en SQLite?

