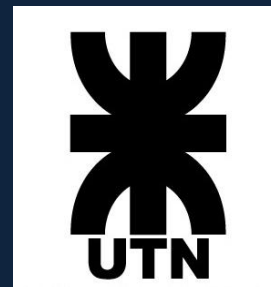


SQLite

versión 3



SQLite

Motor (o librería) de Base de Datos **que implementa el estándar SQL**, y además:

- Es **Autocontenido**
- Es **Transaccional**
- Es **Embebible**
- Carece de servidor
- Carece de configuración

Funcionalidades

Persistencia

Indexación

Integridad

Transaccionado

~~Acceso~~ → "Archivo"

~~Concurrencia~~

~~Autenticación~~

~~Escalabilidad~~

~~Particionamiento~~

~~Replicación~~

Se usa en

- Persistencia en aplicaciones “stand-alone” o de escritorio
- Websites de bajo tráfico
- Reemplazo de formatos de archivo *ad-hoc*
- Prototipado, Experimentos
- Bases de datos internas o temporarias
- **Sistemas Móviles**
- **Sistemas Embebidos**

Pero... por ejemplo?

- Mozilla Firefox
- Skype
- Android (y prácticamente todos los smartphones)
- CoreData, el ORM propietario de iOS / iPhone
- Clientes de correo electrónico de escritorio
- Clientes de chat de escritorio
- Las 145,341,124,272 aplicaciones para organizar y reproducir música y/o películas
- ...etc...

No se usa en

- Aplicaciones cliente/servidor
- Websites de alto tráfico
- Volúmenes de datos muy grandes
- Aplicaciones con alta concurrencia

Propiedades

- **Todo el motor** está contenido en una librería de C de **350KB**
- La base de datos **es un archivo**, y nada más.
- Una base de datos vacía es simplemente **un archivo vacío**.
- El archivo de base de datos es **compatible de forma binaria**.
 - Al ser compatible, es transportable. Con sólo copiarlo funciona en cualquier sistema operativo.
- El “archivo” puede abstraerse del filesystem y existir solamente en memoria, para bases de datos temporarias.

Compatibilidad

- SELECT, INSERT, UPDATE, DELETE: la sintaxis de uso de la base de datos es **exactamente igual**.
- CREATE, DROP: la sintaxis del modelado de datos es **casi idéntica**, salvando atributos específicos (p.e. "InnoDB")
 - **ALTER TABLE es soportado, pero muy limitado**
- Administración: la sintaxis es propia de SQLite. No forma parte del estándar SQL. Cada motor implementa esta parte a su manera.

Tipos de Datos

“Affinity”	Posibles tipos en CREATE TABLE
INTEGER	INT, INTEGER, TINYINT, SMALLINT, MEDIUMINT BIGINT, UNSIGNED BIG INT, INT2, INT8
TEXT	CHARACTER(20), VARCHAR(255) VARYING CHARACTER(255), NCHAR(55) NATIVE CHARACTER(70), NVARCHAR(100) TEXT, CLOB
NONE	BLOB <i>no datatype specified</i>
REAL	REAL, DOUBLE, DOUBLE PRECISION, FLOAT
NUMERIC	NUMERIC, DECIMAL(10,5), BOOLEAN, DATE, DATETIME

Tipos de Datos

- **Todo** se castea a los 4 tipos o “afinidades” básicas.
 - El resto de las nomenclaturas aparecen por cuestiones de compatibilidad con otros motores.
- No tiene tipo **BOOLEAN**
- No tiene tipos de **DATE/TIME**.
 - Recomendán guardar un UNIX Timestamp en un entero, y luego usar las funciones de fecha y hora que sí vienen incluidas.