



SQLite

**La (des)conocida
Súper Hormiga**

Ventajas

- **Gratuita** para todo uso (dominio público)
- Compacta y rápida
- **La más implementada** del mundo
- **Confiable**, muy bien testeada
- Un **estándar** como tipo de datos



Contenido

- SQLite y Python: mejores prácticas
- Transacciones, Aislamiento y Concurrencia
- Indices
- Tipos de datos especiales





SQLite

Usos de SQLite

Cuando usar SQLite

- Análisis de datos
- Como formato propietario de archivo
- Caché, temporarios, en **memoria**
- Versión monousuario, educación o experimentos
- Embebido, móviles, IoT

Cuando **NO** usar sqlite

- Muchas escrituras concurrentes
- Bases de datos GIGANTES (más de 281 Tb)
- Se accede a través de red



SQLite

**Mejores prácticas con
Python**

SQLite y Python: **Conexión**

```
con = sqlite3.connect(database[,...])
con.row_factory = sqlite3.Row
row = con.execute("SELECT 1 as a")
      .fetchone()
col1 = row["a"]
```



SQLite y Python: **Usar context managers**

```
con = sqlite3.connect(database[,...])  
with con:  
    con.execute("sql")
```



SQLite y Python: **Optimizar**

- Mejor usar la conexión que crear cursores
- Usar

```
con.execute("sql", (variables,))
```

```
con.executemany("sql", [n-tuplas])
```

```
con.executescript("sql; sql; ...")
```



SQLite y Python: **Comodines** *PLACE HOLDERS*

- **Comodines,**

```
sql = "INSERT INTO tb (a, b) VALUES (?, ?)"
```

```
con.execute(sql, (var_A, var_B))
```

- Evitar errores de conversión
- Prevenir sql injection.



SQLite y Python: **PRAGMAS**

```
con.execute("PRAGMA  
           schema.opción = valor")
```

- Para ver y configurar opciones en la base de datos
- Schema es la base de datos principal por defecto.





SQLite

**Transacciones
Aislamiento
Concurrencia**

Transacciones

- Todas o ninguna
 - Se graban el grupo de modificaciones atómicas

Ejecuta

1) BEGIN

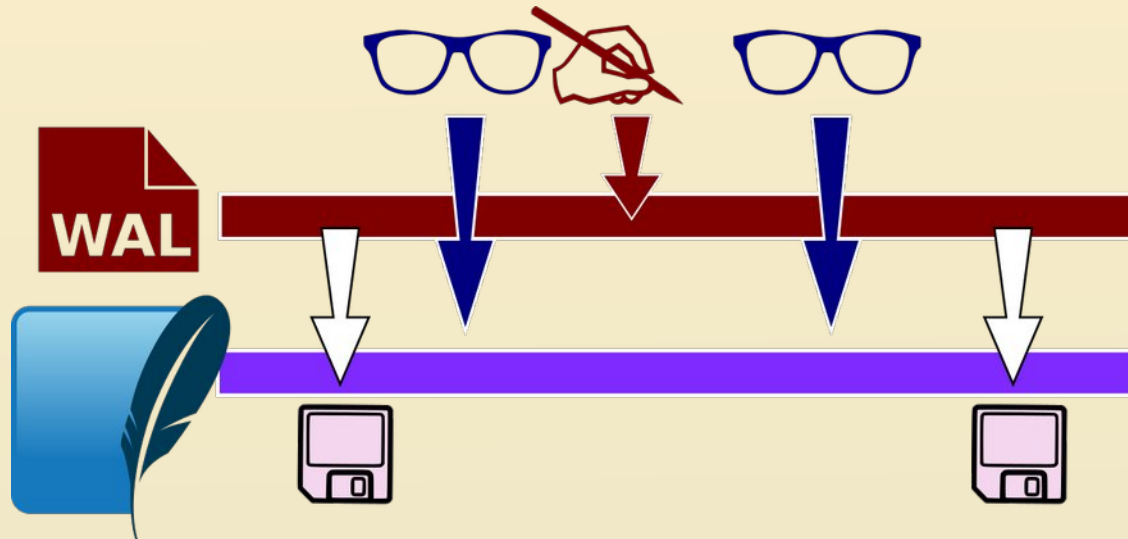
2) Modificaciones

3) COMMIT o ROLLBACK



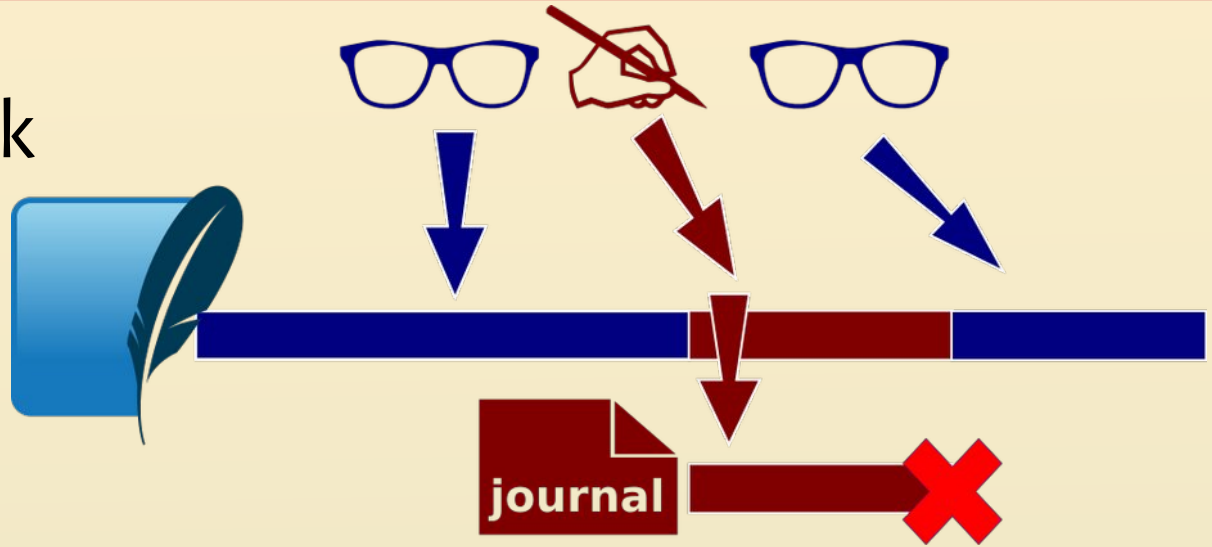
Journal Mode: **W**RITE **A**HEAD **L**OGGING

- Escribe en un archivo de logging. Luego sincroniza



Journal Mode: **Rollback**

- Archivos de rollback
- Pueden ser
 - Temporales
 - Permanentes
 - Truncables



Journal Mode: **En memoria o desactivado**

- Maximiza el rendimiento
- Tienen mayor riesgo de corrupción



Niveles de aislamiento

- Determina cuando los cambios realizados se vuelven visibles para otras conexiones.

Por defecto

```
con = sqlite3.connect(database,  
check_same_thread=True, isolation_level=None)
```

Niveles de aislamiento: **EXCLUSIVE**

- Sólo desde una conexión
- Un solo usuario lector y escritor.



Niveles de aislamiento: **INMEDIATE**

- Solicita escritura
- Completa otras operaciones
- Bloquea BD y escribe
- Libera la BD.



Niveles de aislamiento: **DEFERRED**

- Solicita permiso escritura
- Modo de solo lectura. Siguen las lecturas
- Completa escritura
- Libera la BD para escritura



SQLite

Indices

Cuando usar índices

- Indexar los campos que serán claves foráneas
- Los que se usarán para las búsquedas
- **EVITE** indexar de más
 - Disminuye la performance

Indices parciales

- Indexar sólo algunas filas de una tabla.

```
CREATE UNIQUE INDEX lider  
ON miembros(id_equipo)  
WHERE es_lider_equipo;
```


Indices en expresiones

- Indexa el resultado de funciones **determinísticas**.

```
CREATE INDEX t2xy ON t2(x+y);
```

```
SELECT * FROM t2 WHERE y+x=22;
```

Indice de texto completo: FULL TEXT SEARCH

- Separa por palabras los textos que indexa.
- Genera TABLAS VIRTUALES
- Usar el operador MATCH

```
CREATE VIRTUAL TABLE email USING fts5(sender, title,  
body);
```

```
SELECT * FROM email WHERE email MATCH 'buscar';
```

Indice datos espaciales R*Tree

- Para ubicaciones en espacios N-Dimensionales
- Acelera la búsqueda por espacios delimitados.
- Hasta **5** dimensiones.



SQLite

Tipos de datos

Tipos de datos: standard

- Str
- Int
- Float
- None
- Bytes
- **BLOB**: Grandes objetos binarios
- JSON



Fecha y hora

```
con = sqlite3.connect('db',  
detect_types=sqlite3.PARSE_DECLTYPES)  
con.execute('CREATE TABLE foo (bar, baz timestamp)')  
con.execute('INSERT INTO foo VALUES (?, ?)',  
            (23, datetime.datetime.now()))
```



Fecha y hora: Funciones de SQLite

Convertir str a fecha

- Date, Time,
- Datetime
- Julianday
- Strftime

Modificadores

Permiten agregar o modificar la fecha

- Start of período
- */- x período

Fecha y hora: Funciones de SQLite

Calcular el día de la madre

3º domingo de octubre

```
date('now', 'start of year',  
'+9 months', '+14 days', 'weekday 0')
```




SQLite



María Andrea Vignau