
SQLite

¿Qué es SQLite?

- Librería compacta y autocontenida de código abierto y distribuida bajo dominio público.
- SGBD embebido, sin configuración y transaccional.

Características de la librería

- Código abierto
- Dominio Público
- Compacta
- Autocontenida
- Disponible versión “algamation”

Características de la BD

- Fichero único
- Manifiesto de tipado
- Registros de longitud variable
- Alta seguridad de los datos

Características del SGBD

- Embebido
- No necesita configuración
- Consultas transaccionales
  - Atomicas
  - Consistentes
  - Aisladas
  - Durables

Diferencias en SQL (I)

Restricciones

- No interpreta claves ajenas
- Soporte parcial de triggers
- Soporte parcial de ALTER TABLE
- Soporte parcial de JOIN
- Vistas de sólo lectura
- No hay comandos GRANT y REVOKE

Diferencias en SQL (II)

Añadidos

- Orden REPLACE
- Cláusula ON CONFLICT <algoritmo>
  - ROLLBACK, ABORT, FAIL, IGNORE o REPLACE
- Ordenes ATTACH y DETACH
- Posibilidad de crear nuevas ordenes SQL

SQLite se recomienda para

- Aplicaciones autónomas
- Formato de almacenamiento
- Pequeños dispositivos
- Información interna temporal

SQLite se desaconseja en

- Aplicaciones cliente/servidor
- Bases de datos muy grandes
- Situaciones de alta concurrencia

Comenzando con SQLite

- Descargar la librería desde:

  www.sqlite.org/download.html

- Paquetes disponibles:
  - Binarios
  - Codigo fuente
  - Fichero C “algamation”

SQLite en Ubuntu 7.10

- Paquetes necesarios:
  - libsqlite3-0
  - libsqlite3-0-dev
- Paquetes opcionales
  - sqlite3

Preparar un proyecto

- Incluir la cabecera sqlite3.h
- Linkar la librería:
  - Con gcc ó g++:

    g++ fuente.c -L/usr/lib -o ejecutable -lsqlite3

SQLite para usuarios de PHP / MySQL

Operación	PHP/MySQL	SQLite
Conectar con el servidor	mysql_connect()	
Seleccionar la base de datos	mysql_select_db()	sqlite3_open()
Realizar una consulta	mysql_query()	sqlite3_prepare()
Obtener una fila de la consulta	mysql_fetch_array()	sqlite3_step()
Liberar el resultado	mysql_free_result()	sqlite3_finalize()
Cerrar la conexión con la BD	mysql_close()	sqlite3_close()

Sencillo ejemplo en C/C++ (I)

Incluir librerías

```
#include <iostream>
#include <iomanip>
#include <string>
#include <sqlite3.h>  /* Incluimos la cabecera de sqlite */

using namespace std;
```

Sencillo ejemplo en C/C++ (II)

main y declaración de variables

```
int main(int argc, char *argv[]){

  sqlite3 *db;                /* Puntero a la base de datos */
  sqlite3_stmt *resultado; /* Puntero a los resultados */
  int msg;                    /* Valor de retorno */
  string orden;               /* Ordenes SQL */
  const char* siguiente;   /* Puntero a la siguiente orden */
  char* error;                /* Mensaje de error de una orden */
```

Sencillo ejemplo en C/C++ (III)

Apertura / creación de la base de datos

```
/* Crear un fichero prueba.bd con la base de datos */
msg = sqlite3_open("prueba.bd",&db);

if (msg!=SQLITE_OK) {
  cout << "Error al crear la base de datos\n" << endl;
  exit(1);
}
```

Sencillo ejemplo en C/C++ (IV)

Crear tabla de ejemplo

```
/* Crear una tabla con valores básicos */
orden = "DROP TABLE IF EXISTS alumnos;";
orden+= "CREATE TABLE alumnos (nombre VARCHAR(50),
orden+=                        " edad NUMERIC(3));";
orden+= "INSERT INTO alumnos values('Daniel Ponsoda',28);";
orden+= "INSERT INTO alumnos values('Valentin Carr.',22);";
orden+= "INSERT INTO alumnos values('Omar Marin',26);";
msg = sqlite3_exec(db,orden.c_str(),NULL,NULL,&error);

if (msg!=SQLITE_OK) {
  cout << error << endl;
  exit(2);
}
```

Sencillo ejemplo en C/C++ (V)

Preparar una consulta

```
/* Preparar una consulta */
orden = "SELECT * FROM alumnos ORDER BY nombre;";
msg = sqlite3_prepare (db,orden.c_str(),orden.length(),
                       &resultado,&siguiente);

if (msg!=SQLITE_OK) {
  cout << "Error en la consulta" << endl;
  exit(3);
}
```

Sencillo ejemplo en C/C++ (VI)

Leer las filas obtenidas

```
/* Leer la informacion fila a fila */
while (sqlite3_step(resultado)==SQLITE_ROW){
  cout << sqlite3_column_text(resultado, 0) << " | ";
  cout << sqlite3_column_int (resultado, 1) << endl;
}
```

Sencillo ejemplo en C/C++ (VII)

Cerrar la BD y terminar

```
  /* Cerrar la base de datos */
  sqlite3_close(db);
  return 0;
}
```

Sencillo ejemplo en C/C++ (VII)

Compilación y resultado

```
$ g++ -Wall bd.c -L/usr/lib -o bd -lsqlite3
$ ./bd

Daniel Ponsoda | 28
Omar Marin | 26
Valentin Carretero | 22
```

Enlaces

Pagina web oficial. - www.sqlite.org

Extensión de SQLite para PHP – www.php.net/sqlite

Extensión de SQLite para Python – http://pysqlite.org

Extensión de SQLite para C++ - http://www.int64.org/sqlite.html

Gestor de consultas - http://sqlitebrowser.sourceforge.net

Administrador más completo - http://sqliteadmin.orbmu2k.de/