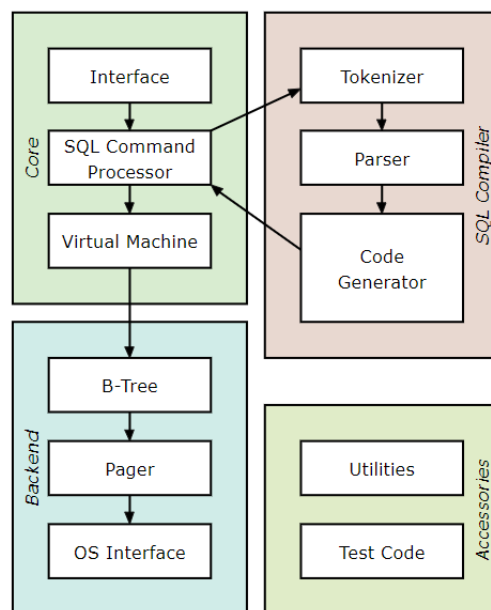


2-A:

1.a 來源: <https://sqlite.org/arch.html>



1.b 來源: <https://sqlite.org/arch.html>

Interface: interface 是可以讓開發者處理資料庫的接口，最主要的是 C interface，讓開發者可以直接使用 C 語言和資料庫互動，除此之外還有許多種，讓 sqlite 可以處理多種語言，並且 Interface 還會利用底線處理重複名稱的問題。

Tokenizer: tokenizer 會將輸入的字串分解並且分別送到 parser 去判斷這行指令是什麼。可以在 tokenize.c 裡面看到，他是利用指標去控制現在要看哪個位置，197 行的 getToken，是去確認接下來的 token 是什麼 type，並且依照讀到的 type 進行不同行為，並且在 567 行的 sqliteRunParser 去進行 parser 的部分，將 token 賦予行動。

Parser: parser 將讀入的指令轉化為有效的行動，sqlite 是用 lemon parser generator，比較不容易出錯，並且在語法錯誤時不會發生 memory leak，parser.y 是文法檔，裡面是各個指令的語法。

Code generator: 產生 bytecode 來執行 statement，有很多檔案都在 code generator 的範圍裡，不同的檔案去解決不同的 statement，會視輸入到 parser 的指令為何調用不同檔案，例如 select.c，就是在處理 select statement，像是 2858 行的 multiselect 就是在處理巢狀的 select。

Bytecode engine: 將先前在 code generator 裡產生的 bytecode 放到

virtual machine 執行。

B-tree: sqlite 是用 B-tree 來儲存資料，btree.c 就是在實作一個 B-tree。在一個 database 裡會用到許多 B-tree，而這些 B-tree 都存在同一個 disk 裡。

Page cache: 他是負責檔案相關的操作，Pager 會將 database 檔案分成固定大小的 page，並且讀入和寫入這些 pages，除此之外，還會快取常用的 pages 讓操作更快，在 pager.c 中，就是在實作存取 disk file，並且還會避免同時對同一個檔案操作。

OS interface: 針對不同的作業系統會有不同的存取指令，因此 sqlite 用 VFS 處理這些問題，VFS 提供在不同作業系統下，各種操作檔案的方法。

Utilities: 各種 utilities 的 function 放在 util.c，在有需要的時候可以直接呼叫裡面的 function，如 memory allocation, caseless string comparison routines 等，除此之外，還有 hash table 在 hash.c，sqlite 自己的 printf 在 print.c，也都算在 utilities 的範圍內。

Test code: 名字開頭有 test 的都是測試專用，於平常使用時不同，在測試時才會使用這些。

1.c 來源: <https://sqlite.org/about.html>

Sqlite 不需要連接到伺服器，他可以直接讀寫檔案，並且還可以在不同的作業系統上運作，再轉換系統時可以不用修改程式，除此之外 sqlite 的大小很小，大約只有幾百 KB，而且也不用下載或是提前設定，可以直接使用。

2.a 來源: <https://www.sqlite.org/howitworks.html>

<https://www.sqlite.org/opcode.html>

Tokenizer: 將輸入的文字分解成一個個 token。

Parser: 將所有 token 除了不需要的空格和註解，放到 parser 裡面，parser 會產生一個語法樹。

Code generator: 將得到的語法樹解析，先找到對應的 table 跟欄位，優化語法樹，再依照語法樹要求的指令選出合適的演算法，輸出 bytecode。

Bytecode engine: 用 sqlite3_step() 執行 bytecode。

2.b.i 來源: <https://sqlite.org/optoverview.html>

Where: 先把 where clause 看成 terms，如果是用 and 連接就將每項個別看成一個 term，如果是用 or 連接就將整體看成一個 term，在 whereInt.h 217 行。接著在 whereexpr.c 132 行，看這個 term 可不可以用 index，在這裡是檢查他的 operator。

Like: 在 whereexpr.c 161 行，like 要可以 optimize，他的右邊就必須事非萬用字元開頭的字串，左邊要是欄位且只能放 NULL、字串、或 BLOB。先檢查 operator 是不是 like，接著算第一個萬用字元前有幾個字元，然後看可不可以 optimize，要能 optimize，他的 pattern 不能以萬用字元開頭，並且萬用字元前不能是 illegal 的字元，或是沒有跳脫字元。之後分成左右邊來看，左邊是欄位，如果左邊不是一般的文字，那他的上下界不能看起來像是文字，不然會被當成文字處理，也就是無法 optimize；如果右邊是變數，且變數的值不需要用到 like，需要加一個 dummy op_variable，不然會出問題。若 like 可以 optimize，就可以直接查找 index，不用去掃描整個 table。

Between: 在 whereexpr.c 1223 行，他會創造兩個新的 virtual terms，分別是他的上下界，在執行的時候，如果 index 符合新的這兩項，就不會用到 between，反過來也是一樣。

2.b.ii 來源: <https://sqlite.org/optoverview.html>

<https://sqlite.org/queryplanner.html>

Query planner 就是要選出最有效率的方法來實作 sql statement，他會考慮 index、估計 cost 來找出方法。當接收到 statement 時，會先檢查他的結構，接著就是去 database 裡找 table 和 columns，並且收集他們的數據，然後計算不同方法的 cost，產出各種 plans，找出最好的 plan。

3.a 來源: <https://sqlite.org/fileformat2.html>

Database file 是由一個或多個 pages 組成，每個 page 的大小是固定的，大小會從 512 到 65536，總共可以有 $2^{32}-2$ 頁，通常 sqlite database 會是 kilobytes 到 gigabytes，每個 page 在 main database 裡只有一種用途分別是這些

- The lock-byte page
- A freelist page
 - A freelist trunk page
 - A freelist leaf page
- A b-tree page
 - A table b-tree interior page
 - A table b-tree leaf page
 - An index b-tree interior page
 - An index b-tree leaf page
- A payload overflow page
- A pointer map page

Database file 裡錢 100 個 bytes 是 header，header 被分為很多個部分，multibytes field 是存在 MSB。

Database Header Format

Offset	Size	Description
0	16	The header string: "SQLite format 3\000"
16	2	The database page size in bytes. Must be a power of two between 512 and 32768 inclusive, or the value 1 representing a page size of 65536.
18	1	File format write version. 1 for legacy; 2 for WAL .
19	1	File format read version. 1 for legacy; 2 for WAL .
20	1	Bytes of unused "reserved" space at the end of each page. Usually 0.
21	1	Maximum embedded payload fraction. Must be 64.
22	1	Minimum embedded payload fraction. Must be 32.
23	1	Leaf payload fraction. Must be 32.
24	4	File change counter.
28	4	Size of the database file in pages. The "in-header database size".
32	4	Page number of the first freelist trunk page.
36	4	Total number of freelist pages.
40	4	The schema cookie.
44	4	The schema format number. Supported schema formats are 1, 2, 3, and 4.
48	4	Default page cache size.
52	4	The page number of the largest root b-tree page when in auto-vacuum or incremental-vacuum modes, or zero otherwise.
56	4	The database text encoding. A value of 1 means UTF-8. A value of 2 means UTF-16le. A value of 3 means UTF-16be.
60	4	The "user version" as read and set by the user_version pragma .
64	4	True (non-zero) for incremental-vacuum mode. False (zero) otherwise.
68	4	The "Application ID" set by PRAGMA application_id .
72	20	Reserved for expansion. Must be zero.
92	4	The version-valid-for number .
96	4	SQLITE_VERSION_NUMBER

3.b

在 os.c 82 行有 `sqlite3OsRead`、`sqlite3OsWrite`、`sqlite3OsClose`，分別對應到 `read`、`write`、`close`，在這三個 function 裡，都會呼叫 `sqlite3_file` 的 function，`xRead`、`xWrite`、`xClose`。

4.a 來源: <https://sqlite.org/lockingv3.html>

Locking: sqlite 有 5 種 locking state，database file 可能是 5 種裡的其中 1 種。Unlock 是指 database file 沒有鎖，不能讀也不能寫，所有的快取資料都是需要被驗證的狀態，是預設的狀態；Shared 是可以讀但不能寫，可以同時很多人讀，啟動多個 shared lock，但不論如何都不可寫入檔案；Reserved 是指之後將要變成可以寫的狀態，但還仍然是只能讀不能寫，一次只能有一個 reserved lock 被啟動，但還是能有多個 shard lock；Pending 是指想要寫入檔案且越快越好，只要等到所有 shared lock 都清空，就可以取得 exclusive lock，啟動 pending lock 時就不能再增加 shared lock；Exclusive 是指可以寫入檔案，一次只能有一個 exclusive lock，並且不能有其他的 lock 存在。

4.b

2-B:

Tokenizer: tokenizer 在 sqlite 裡，會負責將輸入的字串分解一個個 token，並且送到 parser 去判斷這行指令是什麼。

在 tokenize.c 29 行開始，是去定義每個符號的 id，這些 id 會用在判斷這個 token 是甚麼。

第 61 行開始是他的 table，利用對應 table 得知當前為哪個符號。

在第 197 行的 getToken，是要得到 token 的 id，也就是前面定義的。

第 217 行，當 tokenizer 讀到 window、over、filter 的時候，他要特別處理，因為他需要知道這個詞是 keyword 還是 SQL identifier，因此這裡是去辨別這些詞是否為 keyword，他們會分別在以下的情況為 keyword。

WINDOW is treated as a keyword if:

- * the following token is an identifier, or a keyword that can fallback to being an identifier, and
- * the token after than one is TK_AS.

OVER is a keyword if:

- * the previous token was TK_RP, and
- * the next token is either TK_LP or an identifier.

FILTER is a keyword if:

- * the previous token was TK_RP, and
- * the next token is TK_LP.

在 273 行，sqlite3GetToken 回傳的是這個 token 的 size，並且還會將 token 的種類存在 *tokenType，裡面是用 switch case 實作，根據不同種類的 token 做出不同的事，而種類就是前面定義的那些符號。

最後在 567 行去執行 parser，在 tokenizer 裡面去呼叫 parser 會比從 parser 裡面呼叫 tokenizer 還要快，並且比較安全。