

# ソースコード講読 alter.c

井上優作

2014 年 10 月 12 日

## ALTER TABLE について

SQLite3 の ALTER TABLE コマンドは、テーブルの名前を変更するか、またはカラムを追加することができるコマンドである。以下のようにして実行する。

```
ALTER TABLE tbl RENAME TO new_table_name
ALTER TABLE tbl ADD COLUMN column-def
```

## ソースコード

- L21: このファイルのコードは、ビルド時に ALTER TABLE コマンドを除かない限り存在する。

### LL.37-84 static void renameTableFunc

コメント中にあるコード例の引数が、この関数の第 3 引数に配列 ( \*\*argv ) となって入っている。42-43 行目で、SQL 文とテーブルの名前が取り出されている。テーブル名の特定規則は、TK\_LP または TK\_USING の後に続く最初のスペースでないトークンであるとしており、これを 59 - 78 行目で実行している。この関数自体は戻り値は void だが、80 行目の zRet が実質的な戻り値？

### LL.102-148 static void renameParentFunc

外部キー制約の定義を修正するための関数。120 - 127 行目で古い名前を探して、129 - 138 行目でそれを新しい名前に変更しているような気がする。

### LL.158-224 static void renameTriggerFunc

第 1 引数が CREATE TRIGGER 文、第 2 引数がテーブルの名前で、これが第 3 引数のテーブル名に置き換えられる。CREATE TRIGGER 文の中のテーブル名は TK\_ON か TK\_DOT と TK\_WHEN, TK\_BEGIN または TK\_FOR の間にあるトークンである。この規則に従って、181-215 行目でテーブルの名前を特定している。今までに出てきた関数と違って、古い名前を新しい名前に置き換えているようなコードが見つからない。

## LL.230-247 void sqlite3AlterFunctions

今までに定義した関数をグローバルな関数定義ハッシュテーブルに登録しているらしいということだけ分かった (L245 の `sqlite3FuncDefInsert` は `callback.c:L304` で定義されている)。

## LL.265-274 static char \*whereOrName

複数の `constant` を受け取って `name=<constant1> OR name=<constant2> OR ...` というフォーマットの文字列を返す関数。フォーマット指定子に使われている `%Q` は標準で実装されているものではない？ また、270 行目だけ見ると、3 つ以上の `constant` があるときにどうやって `OR` でつないだ文字列を返すのかが分からない。

## LL.283-290 char \*whereForeignKeys

上で定義した関数を使って `where` 句を返す関数。 `zWhere` という 1 つの変数に代入し続けている。 `*whereOnName` の返し方が再帰的になっている？

## LL.299-323 static char \*whereTempTriggers

全てのテンポラリートリガーを選択する `where` 句文字列を生成するための関数。 309-316 行目でそれを生成して、317-322 行目で返している？

## LL.333-372 void reloadTableSchema

テーブルの内部表現を一旦削除して読み込み直す関数。 341-345 行で `VDBE` とデータベースを準備、347-353 行目で内部スキーマからトリガーを削除、357 行目でテーブルとインデックスを削除、360 - 362 行目でテーブルを再読み込みしている。 368-370 行目では、もしテーブルがテンポラリーデータベースに無ければテンポラリートリガーを読み込む (?)。