Plain verbatim listing. It works for UTF8 input but does not break lines, format syntax or update listings list.

```
z ← keyn SQLiteFrDBI dbi;t;n;st;sn;sc;sql;rf;rm;nrc;⎕io

⍝ Returns a SQLite table creation statement that maps the
⍝ structure of APL+WIN inverted DBI files to SQLite tables.
⍝
⍝ monad: ev ← SQLiteFrDBI cvDBIFileName
⍝
⍝   sqlv ← SQLiteFrDBI 'C:\BCA\bcadev\CA\ULTCL.DBI'

 ⎕io ← 1

⍝ default SQLite primary key name
 :if 0=⎕nc 'keyn' ⋄ keyn ← 'rwkey' ⋄ :endif

⍝ open DBI file - the DBI open function creates
⍝ a number of global variables that are used to
⍝ to access data stored in these files. These
⍝ variable contain '∆_' in their names.
 ⎕error (1∊'∆_'⍷⎕nl 2)/'DBI globals present - erase all DBI globals'

 DBIOpen dbi                         ⍝ open DBI file
 z ← (∨/'∆_'⍷⎕nl 2)⌿⎕nl 2            ⍝ varibles set by DBI open

⍝ we need table name, column names, types, repeat codes

 t ← ⍎(∨/'∆_fty'⍷z)⌿z        ⍝ column types
 n ← ⍎(∨/'∆_fnm'⍷z)⌿z        ⍝ column names prefixed by table
rf ← ⍎(∨/'∆_fnc'⍷z)⌿z     ⍝ (0≤) indicates a DBI repeated column (numeric matrix)
 z ← ⎕ex z                   ⍝ clear DBI open globals
 z ← ⎕ex ⊃'∆DBIFL' '∆DBIFN'

⍝ correspondence between DBI and SQLite column types - SQLite does
⍝ not distinguish between integer types and ignores all fixed length
⍝ declarations in SQL column declarations this is ideal for APL data
⍝ C=text, I=integer, U=integer, F=real, D=date)
 st ← ('text' 'integer' 'integer' 'real' 'date')['CIUFD'⍳t]
 sn ← (¯1 + (↑n)⍳'∆') ↑ ↑n          ⍝ table name without '∆'
 nrc ← sc ← (n ⍳¨ '∆') ↓¨ n         ⍝ just column names

⍝ expand any repeated numeric columns
 :if ∨/rm ← 0 < rf
    sc ← ⊂¨ sc
    (rm/sc) ← (rm/sc) ,¨¨  ⍕¨¨ ⍳¨ rm/rf
    sc ← ⊃ ,/ sc
    st ← (1⌈rf) / st
 :endif
```

```
⍝ SQLite tables require a primary key DBI files
⍝ do not necessarily have a primary key
 ⎕error ((⊂keyn)∊sc)/'(',keyn,') key name occurs in DBI file - use another name'
 sql ← 'create table ',sn,' (',keyn,' integer primary key, '
 sql ← sql , (¯2 ↓ ∊ sc , ' ' , st ,[1.5] ⊂', '), ')'

⍝ return sql, table name, SQLite types, repeating and non-repeating columns
 z ← sql sn st sc nrc
```

The UTF8 APL "trouble makers" properly handled within a `lstlisting` environment. Characters following the APL comment "lamp" character ⍝ get comment coloring. Syntax coloring is one of the features of `lstlisting`.

```
!*+-/<=>?\^|~¨¯×÷←↑→
↓∆∇∊∘∨∩∪∵∼≠≡≢≤≥⊂⊃⊖⊢⊣
⊤⊥⋄⌈⌊⌶⌷⍠⌼⌽⌿⍀⍁⍃⍄⍉⍋⍎⍐⍒
⍕⍗⍙⍝⍞⍟⍡⍢⍣⍤⍥⍨⍪⍫⍬⍱⍲⍳⍴⍵
⍷⍸⍺⎕⎕◇○₧
```

`lstlisting` verbatim with extended UTF8 APL characters.

```
z ← keyn SQLiteFrDBI dbi;t;n;st;sn;sc;sql;rf;rm;nrc;⎕io

⍝ Returns a SQLite table creation statement that maps the
⍝ structure of APL+WIN inverted DBI files to SQLite tables.
⍝
⍝ monad: ev ← SQLiteFrDBI cvDBIFileName
⍝
⍝   sqlv ← SQLiteFrDBI 'C:\BCA\bcadev\CA\ULTCL.DBI'

 ⎕io ← 1

⍝ default SQLite primary key name
 :if 0=⎕nc 'keyn' ⋄ keyn ← 'rwkey' ⋄ :endif
```

```
⍝ open DBI file - the DBI open function creates
⍝ a number of global variables that are used to
⍝ to access data stored in these files. These
⍝ variable contain '∆_' in their names.
 ⎕error (1∊'∆_'⍷⎕nl 2)/'DBI globals present - erase all DBI globals'

 DBIOpen dbi                          ⍝ open DBI file
 z ← (∨/'∆_'⍷⎕nl 2)⌿⎕nl 2             ⍝ varibles set by DBI open

⍝ we need table name, column names, types, repeat codes

 t ← ⍎(∨/'∆_fty'⍷z)⌿z         ⍝ column types
 n ← ⍎(∨/'∆_fnm'⍷z)⌿z         ⍝ column names prefixed by table
 rf ← ⍎(∨/'∆_fnc'⍷z)⌿z        ⍝ (0≤) indicates a DBI repeated column (numeric matrix)
 z ← ⎕ex z                    ⍝ clear DBI open globals
 z ← ⎕ex ⊃'∆DBIFL' '∆DBIFN'

⍝ correspondence between DBI and SQLite column types - SQLite does
⍝ not distinguish between integer types and ignores all fixed length
⍝ declarations in SQL column declarations this is ideal for APL data
⍝ C=text, I=integer, U=integer, F=real, D=date)
 st ← ('text' 'integer' 'integer' 'real' 'date')['CIUFD'⍳t]
 sn ← (¯1 + (↑n)⍳'∆') ↑ ↑n           ⍝ table name without '∆'
 nrc ← sc ← (n ⍳¨ '∆') ↓¨ n          ⍝ just column names

⍝ expand any repeated numeric columns
 :if ∨/rm ← 0 < rf
    sc ← ⊂¨ sc
    (rm/sc) ← (rm/sc) ,¨¨  ⍕¨¨ ⍳¨ rm/rf
    sc ← ⊃ ,/ sc
    st ← (1⌈rf) / st
 :endif

⍝ SQLite tables require a primary key DBI files
⍝ do not necessarily have a primary key
 ⎕error ((⊂keyn)∊sc)/'(',keyn,') key name occurs in DBI file - use another name'
 sql ← 'create table ',sn,' (',keyn,' integer primary key, '
 sql ← sql , (¯2 ↓ ∊ sc , ' ' , st ,[1.5] ⊂', '), ')'

⍝ return sql, table name, SQLite types, repeating and non-repeating columns
 z ← sql sn st sc nrc
```