

E-4 RESOLUTION + CHART 1956

The Land Of Tcl/Tk

The Land Of Tcl/Tk

Other

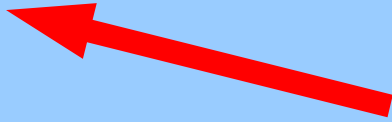
Civilization



Civilization

**Teaming Hordes
of Unwashed
Heathen Barbarians**

The Land Of Tcl/Tk

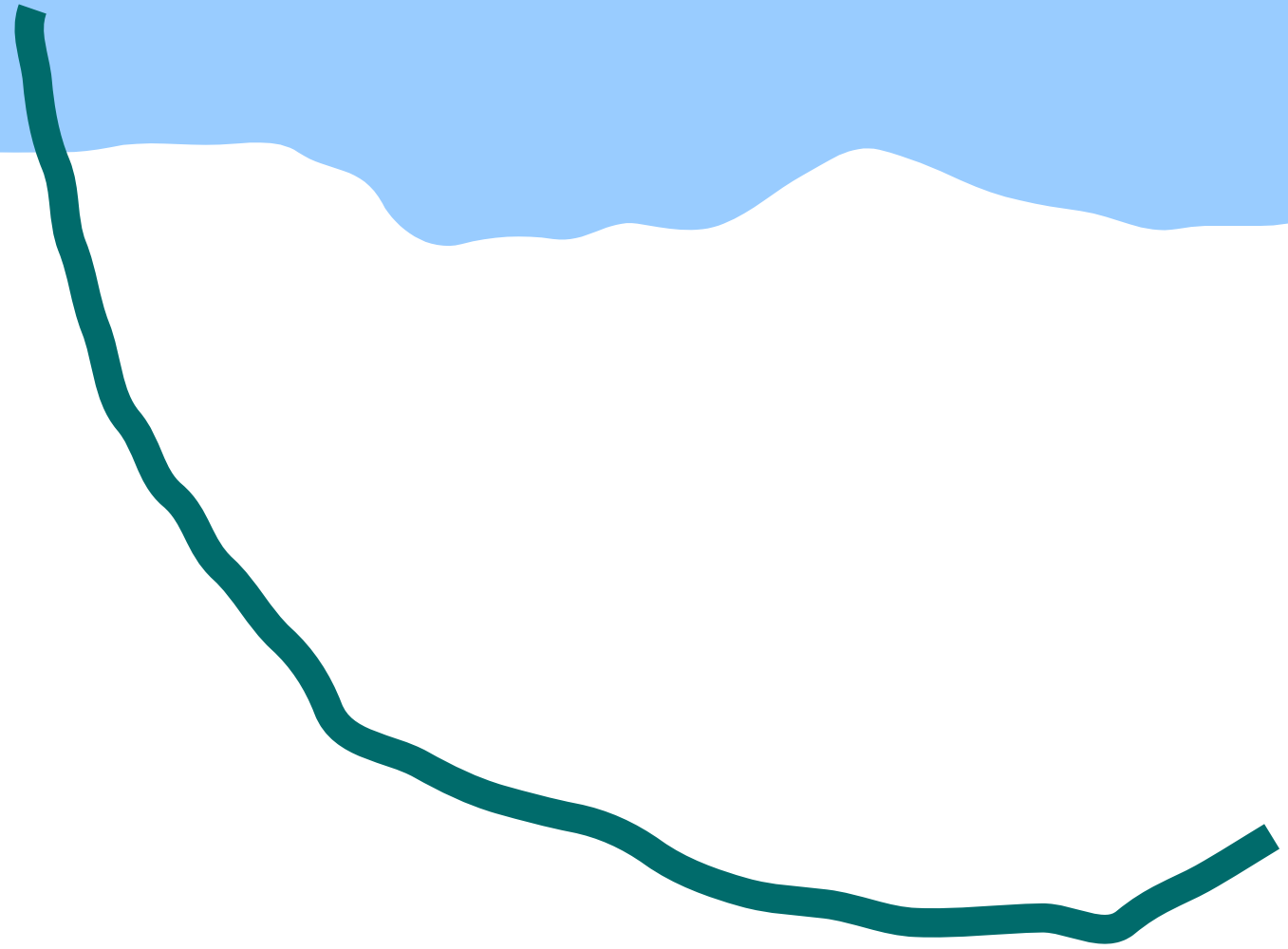


Begin Here

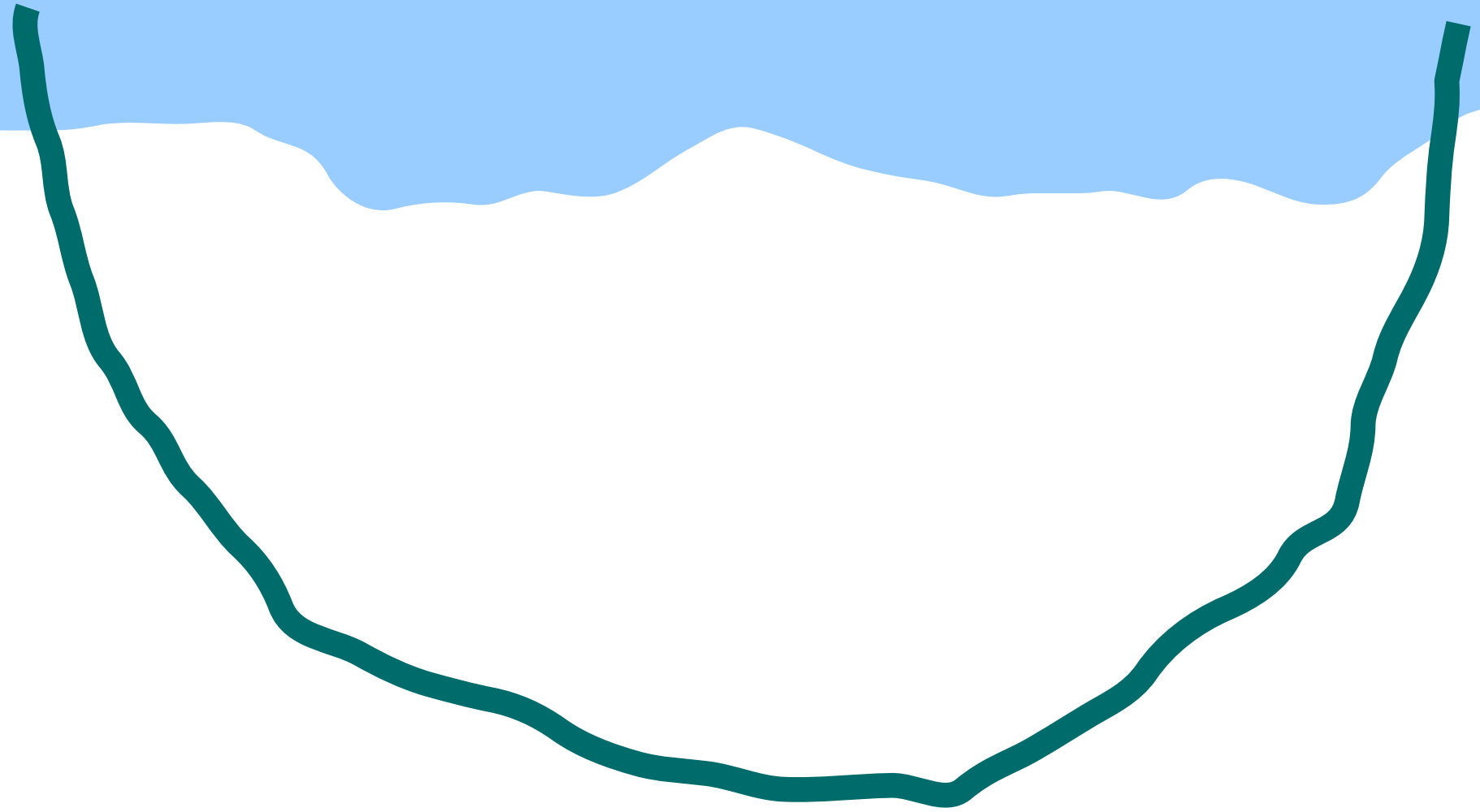
The Land Of Tcl/Tk



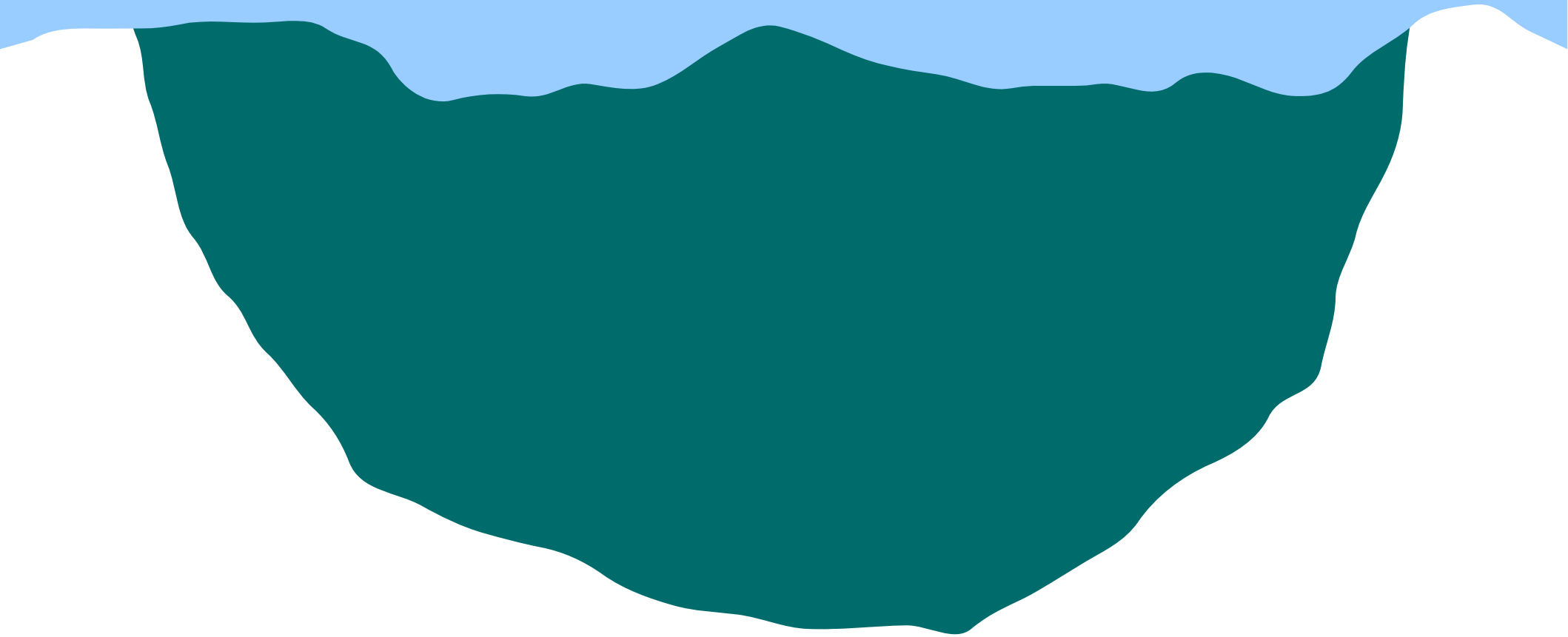
The Land Of Tcl/Tk



The Land Of Tcl/Tk



The Land Of Tcl/Tk



The Land Of Tcl/Tk



SQLite 



- ACID
- Most of SQL-92
- Zero-configuration
- Single disk file
- Small footprint (225KiB)
- Cross-platform databases
- Serverless
- Unlimited length BLOBs and CLOBs
- Manifest typing
- User-defined functions
- Fast
- Public Domain

SQLite 

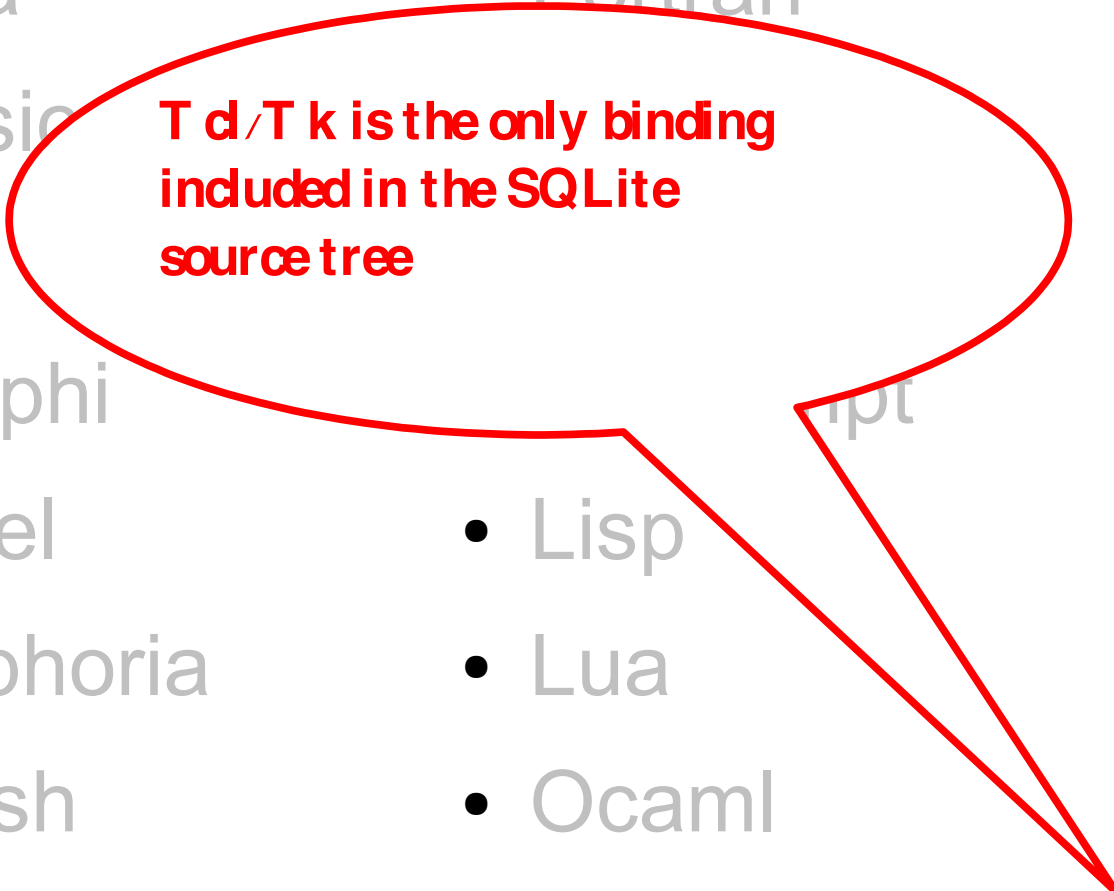
written in and for



SQLite Language Bindings

- Ada
- Basic
- C#
- Delphi
- Eiffel
- Euphoria
- Flash
- Forth
- Fortran
- Haskell
- Java
- Javascript
- Lisp
- Lua
- Ocaml
- Perl
- PHP
- Pike
- Python
- Rebol
- Ruby
- Scheme
- Smalltalk
- Tcl/Tk

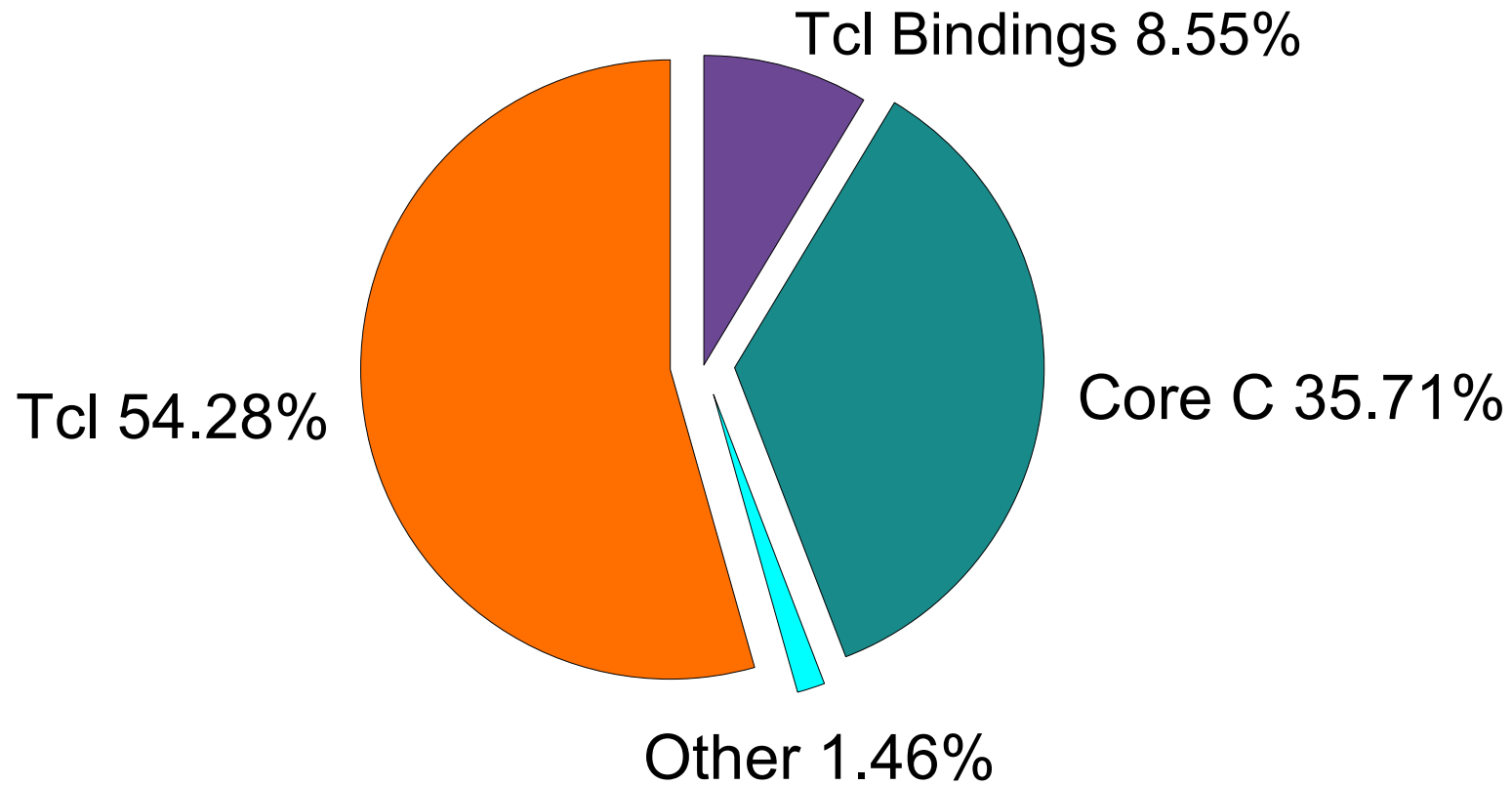
SQLite Language Bindings

- Ada
 - Basic
 - C#
 - Delphi
 - Eiffel
 - Euphoria
 - Flash
 - Forth
 - Fortran
 - Lisp
 - Lua
 - Ocaml
 - Perl
 - PHP
 - Pike
 - Python
 - Rebol
 - Ruby
 - Scheme
 - Smalltalk
 - **Tcl/Tk**
- 
- Tcl/Tk is the only binding included in the SQLite source tree**

Similar Design Philosophy

- Manifest Typing
- No arbitrary limits
- Designed for embedding
- “Just Works”
- Readable source code

Non-Comment Source Lines In SQLite 3.3.7





GENERAL DYNAMICS



Microsoft®

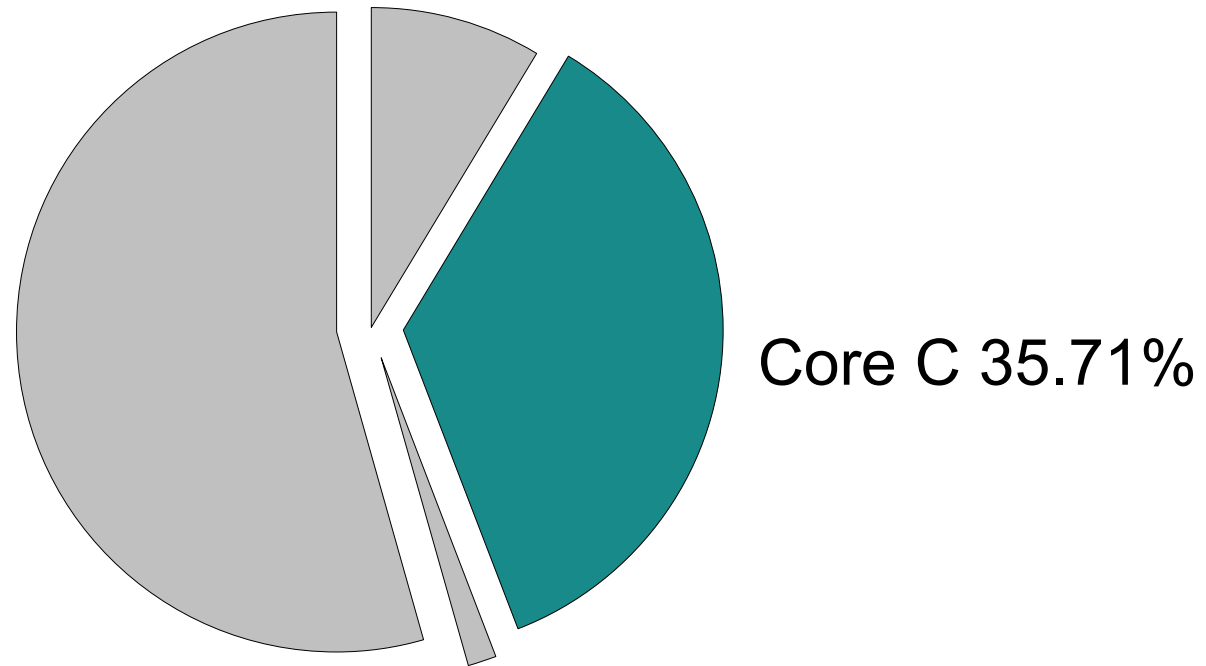
TOSHIBA

symbian





Non-Comment Source Lines In SQLite 3.3.7



Uses For SQLite

- Replacement for client-server RDBMS
- Stand-in for enterprise RDBMS during testing and demos
- Local cache of enterprise RDBMS data
- Persistence of objects, configuration options, etc.
- Complex data structures
- Application file format

Uses For SQLite

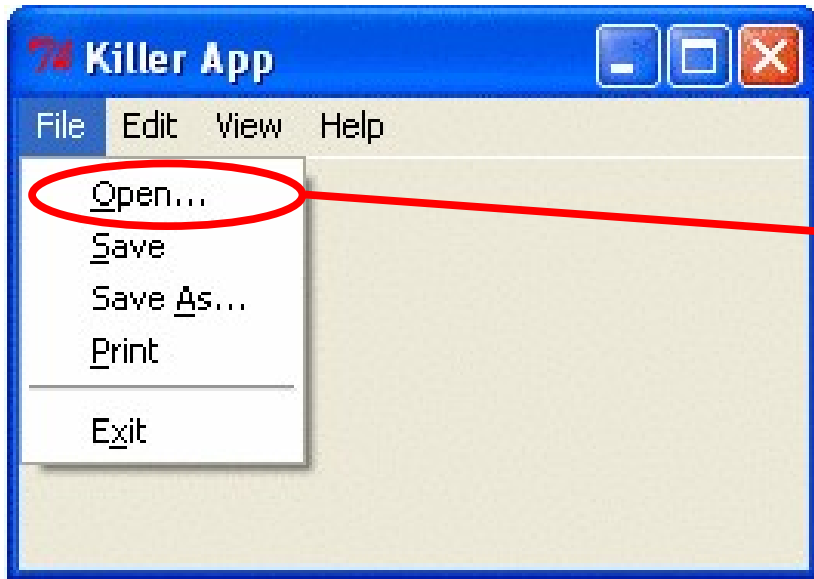
- Replacement for client-server RDBMS
- Stand-in for enterprise RDBMS during testing and demos
- Local cache of enterprise RDBMS data
- Persistence of objects, configuration options, etc.
- Complex data structures
- Application file format



Traditional Uses

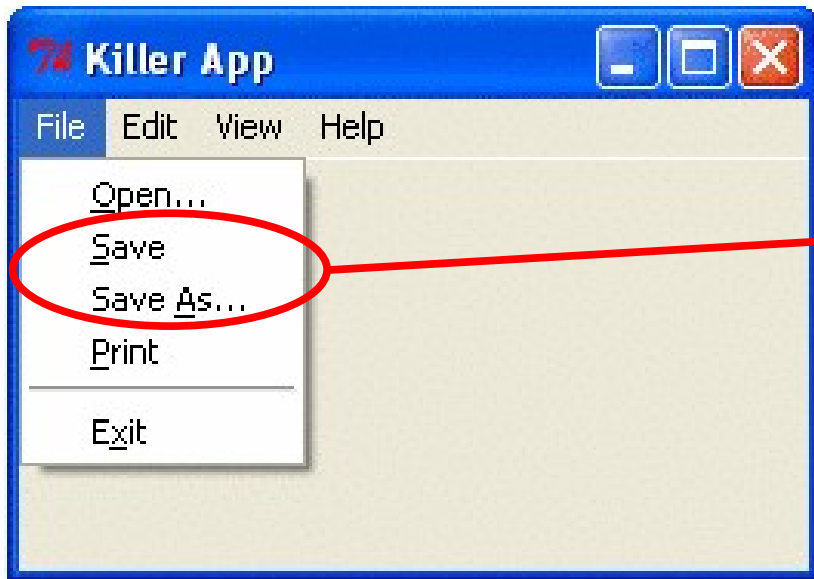


Non-traditional Uses



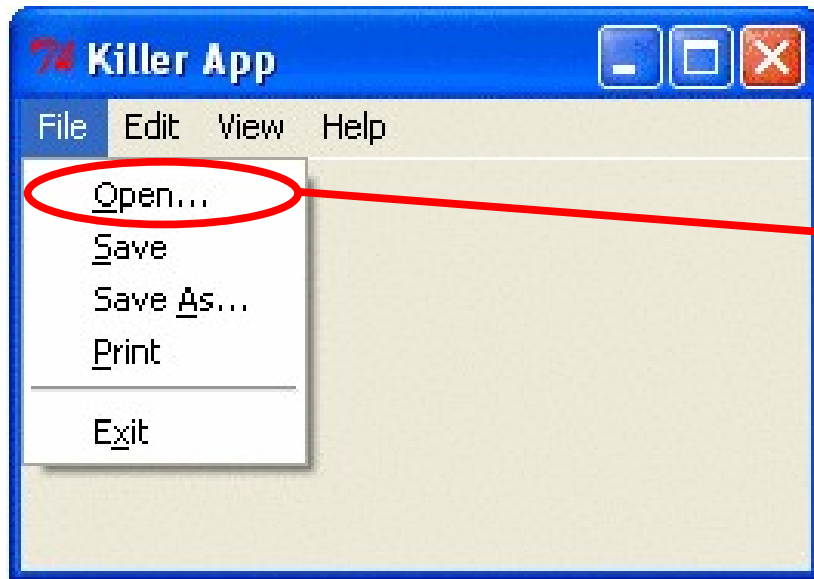
```
set fn [tk_getOpenFile ...]
if {$fn!=""} {
    set fd [open $fn]
    # read and parse
    close $fd
}
```

- File/Open reads and parses the entire application file
- Error prone
- Lots of code to maintain
- Application crash causes data loss



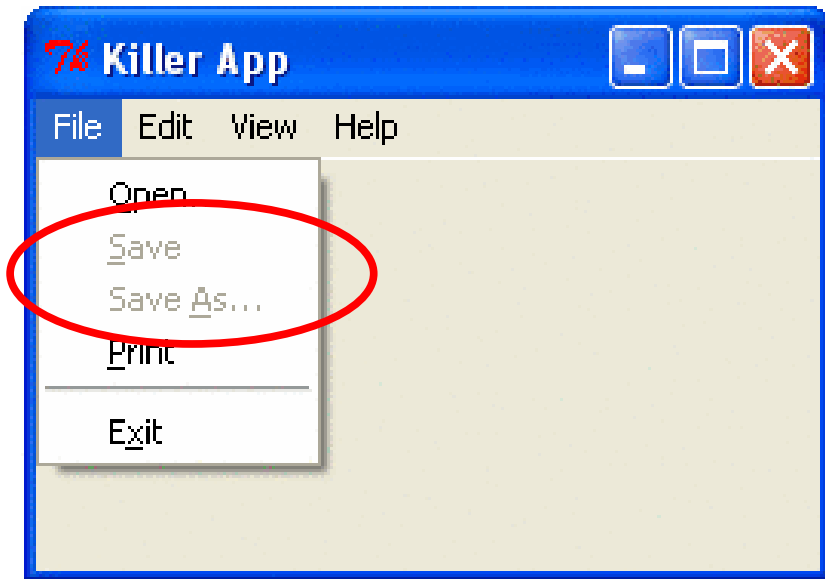
```
set fn [tk_getSaveFile ...]
if {$fn!=""} {
    set fd [open $fn w]
    # Write the file
    close $fd
}
```

- Requires the user to remember to save
- Possible overwrite of independent changes
- Corruption if file generator and parser do not match



```
set fn [tk_getOpenFile ...]
if {$fn!=""} {
    sqlite3 db $fn
}
```

- No need to read and parse file content
- Updates are atomic, consistent, isolated, and durable
- Automatic concurrency control



- Changes are written to disk immediately
- No data loss after unexpected power failure
- The Save and Save As options are obsolete – remove them.

Key Concept



SQLite wants to replace
fopen() not Oracle

Key Concept



SQLite wants to replace
[open] not oratcl

Go here

SQLite Download Page

http://www.sqlite.org/download.html

sqlite-source-3 3 8.zip (586.44 KiB)

This ZIP archive contains pure C source code for the SQLite library. Unlike the tarballs below, all of the preprocessing and automatic code generation has already been done on these C source code, so they can be processed directly with any ordinary C compiler. This file is provided as a service to MS-Windows users who lack the build support infrastructure of Unix.

sqlite-3 3 8-tea.tar.gz (672.73 KiB)

A tarball of proprocessed source code together with a Tcl Extension Architecture (TEA) compatible configure script and makefile.

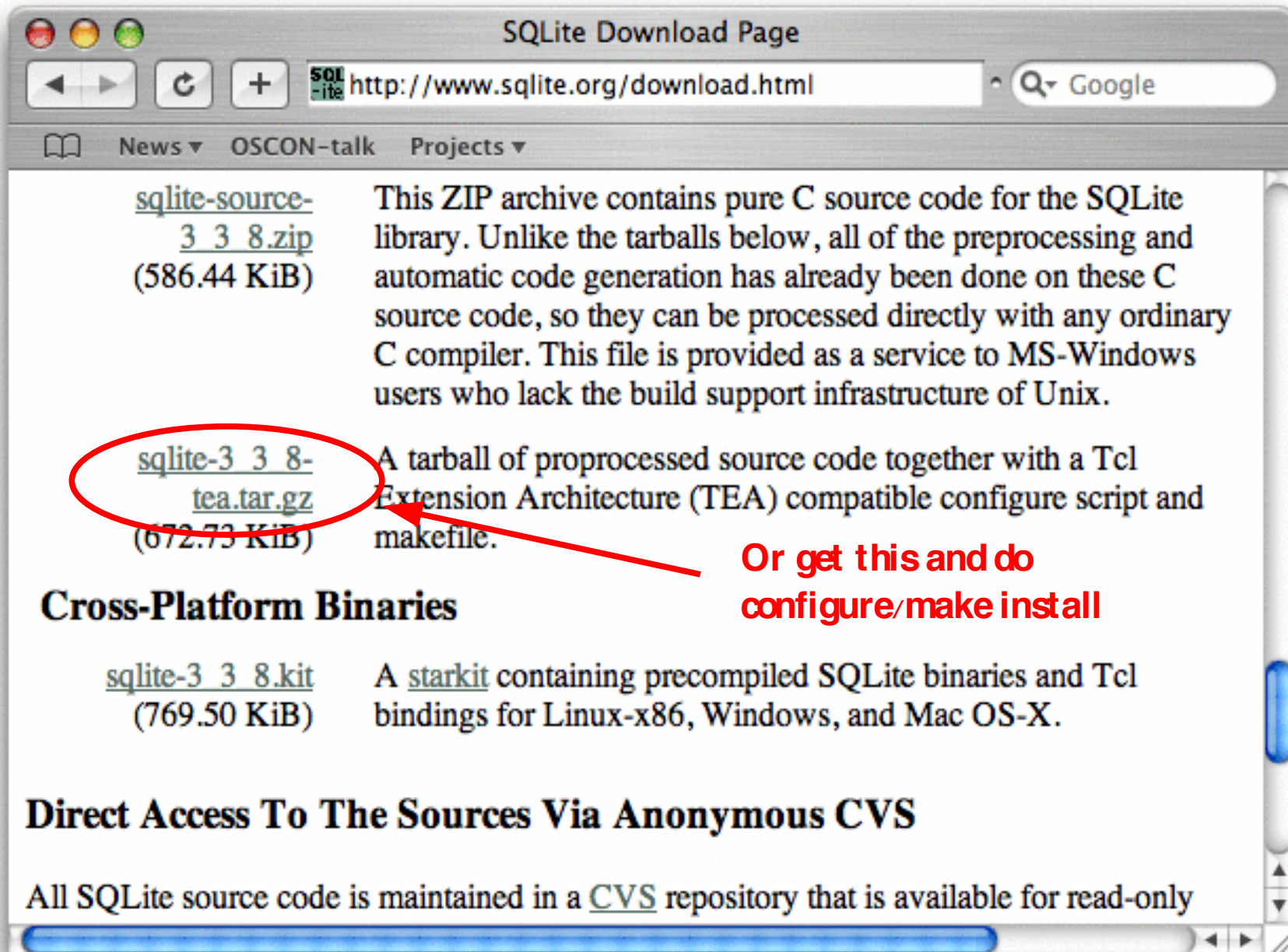
Cross-Platform Binaries

sqlite-3 3 8.kit (769.50 KiB)

A starkit containing precompiled SQLite binaries and Tcl bindings for Linux-x86, Windows, and Mac OS-X.

Direct Access To The Sources Via Anonymous CVS

All SQLite source code is maintained in a CVS repository that is available for read-only



Can be omitted if SQLite
is installed on your system

```
% source sqlite-3_3_8.kit  
% package require sqlite3  
3.3.8  
% sqlite3 db database.db  
%
```

New object for
controlling the database

Name of the database file.
A new one is created if it does
not already exist.

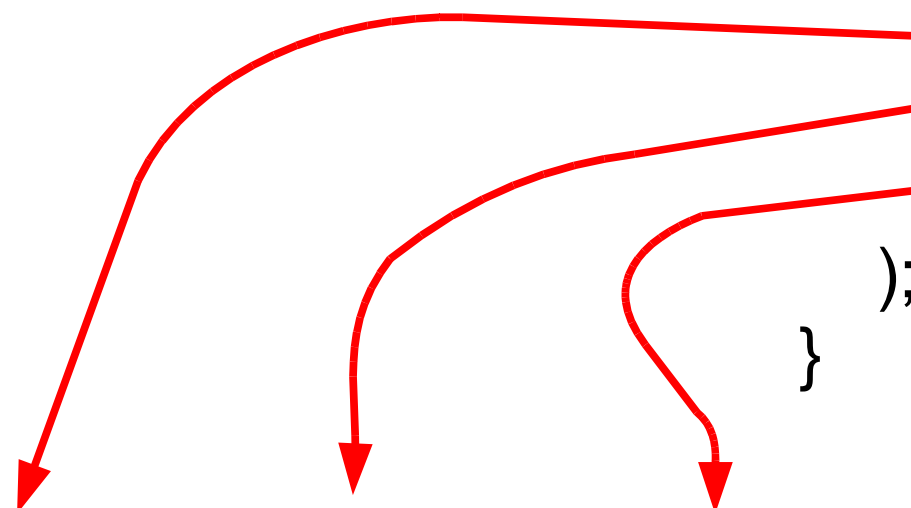
Use the "eval" method to run SQL

```
db eval {  
    CREATE TABLE users(  
        userid INTEGER,  
        first_name VARCHAR(30),  
        last_name VARCHAR(40)  
    );  
}
```

user	first_name	last_name

Semicolon separates multiple SQL statements.
Final semicolon is optional.

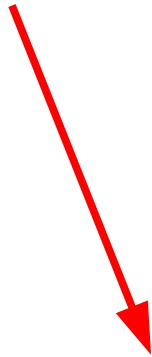
```
db eval {  
  CREATE TABLE users(  
    userid INTEGER,  
    first_name VARCHAR(30),  
    last_name VARCHAR(40)  
  );  
}
```



user	first_name	last_name

```
db eval {  
  CREATE TABLE users(  
    userid INTEGER,  
    first_name VARCHAR(30),  
    last_name VARCHAR(40)  
  );  
}
```

Fixed set of columns



user	first_name	last_name



Arbitrary number of rows

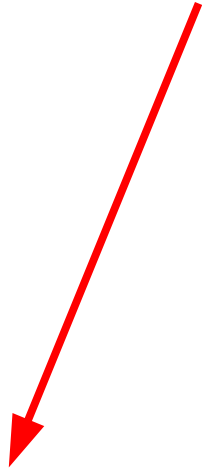
```
db eval {  
  CREATE TABLE users(  
    userid INTEGER,  
    first_name VARCHAR(30),  
    last_name VARCHAR(40)  
  );  
}
```

user	first_name	last_name

Datatypes are ignored, mostly

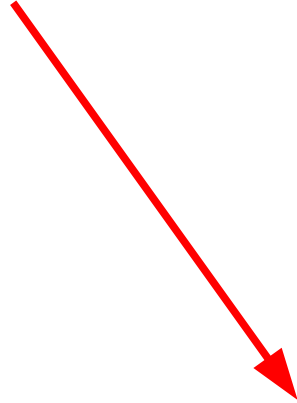


Use an INSERT statement to add data



```
db eval {  
  INSERT INTO users  
  VALUES(1, 'D. Richard', 'Hipp')  
}
```

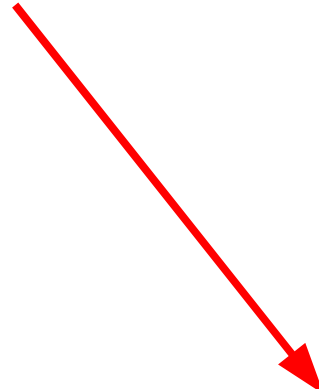
Better to specify column names



```
db eval {  
  INSERT INTO users(user,first_name,last_name)  
  VALUES(1, 'D. Richard', 'Hipp')  
}
```



Columns can occur in any order



```
db eval {  
  INSERT INTO users(last_name, user, first_name)  
  VALUES('Hipp', 1, 'D. Richard')  
}
```



```
db eval {  
  INSERT INTO users  
  VALUES(1, 'D. Richard', 'Hipp')  
}
```

user	first_name	last_name
1	D. Richard	Hipp

Use a SELECT statement to extract data from the database

```
% db eval {  
    SELECT *  
    FROM users  
}  
1 {D. Richard} Hipp  
%
```

“*” means get all columns



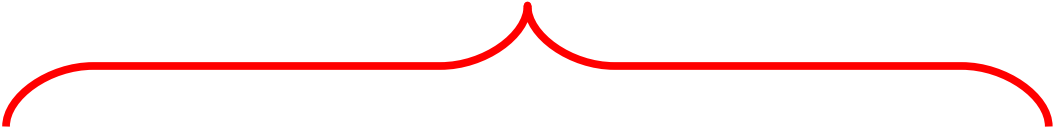
```
% db eval {  
    SELECT *  
    FROM users  
}  
1 {D. Richard} Hipp  
%
```



Data returned in a T CL list

Better to specify specific column names
rather than use "*"

```
% db eval {  
    SELECT user, first_name, last_name  
    FROM users  
}  
1 {D. Richard} Hipp  
%
```



```
db eval {  
    INSERT INTO users  
    VALUES(2, 'Ginger', 'Wyrick')  
}
```

user	first_name	last_name
1	D. Richard	Hipp
2	Ginger	Wyrick

```
% db eval {  
    SELECT * FROM users  
}  
1 {D. Richard} Hipp 2 Ginger Wyrick  
%
```



**Additional rows of data just
make the returned list longer**

% db close

% rename db {}

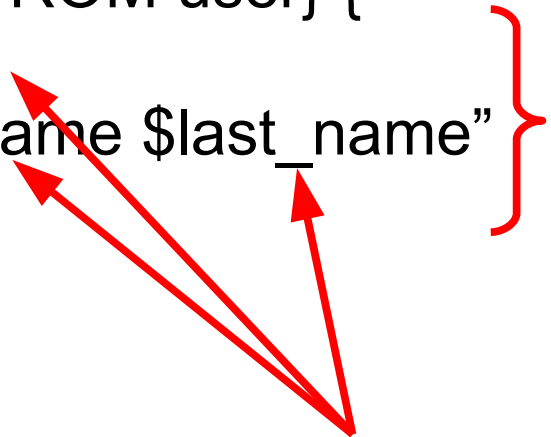


These do the same thing

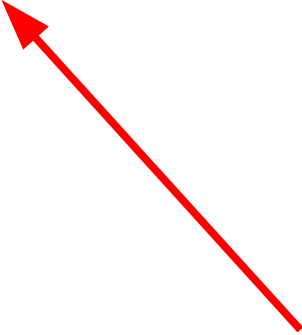
```
% sqlite3 db database.db
% db eval {SELECT * FROM user} {
  puts userid=$userid
  puts "name=$first_name $last_name"
}
userid=1
name=D. Richard Hipp
userid=2
name=Ginger Wyrick
%
```

Script runs once for each row in result set

Column contents store in T CL variables




```
% sqlite3 db database.db
% foreach {userid first_name last_name} \
    [db eval {SELECT * FROM user}] {
    puts userid=$userid
    puts "name=$first_name $last_name"
}
userid=1
name=D. Richard Hipp
userid=2
name=Ginger Wyrick
%
```



**Similar to the previous
except entire result set
is held in memory here**

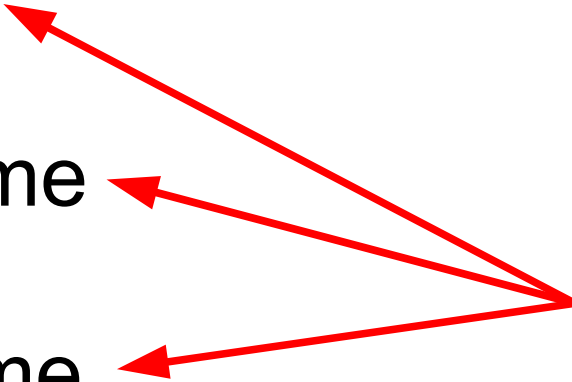

```
% sqlite3 db database.db
% db eval {SELECT * FROM user} {
  puts userid=$userid
  puts "name=$first_name $last_name"
  break
}
userid=1
name=D. Richard Hipp
%
```

**“break” and “continue” work
in the usual way**



```
% db eval {SELECT * FROM user} break
% set userid
1
% set first_name
D. Richard
% set last_name
Hipp
%
```

**Variables persist after
the last iteration of the
loop**



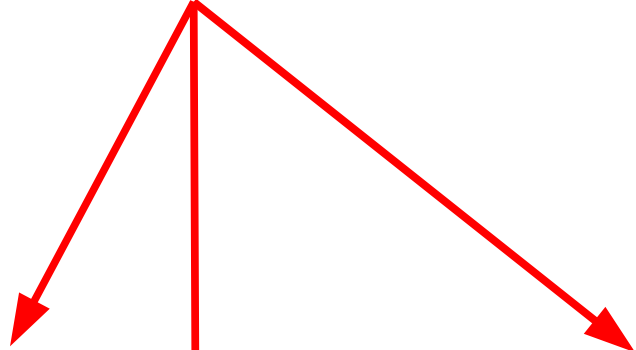
```
% db eval {SELECT * FROM user} var {  
  puts userid=$var(userid)  
  puts "name=$var(first_name) $var(last_name)"  
}  
userid=1  
name=D. Richard Hipp  
userid=2  
name=Ginger Wyrick  
%
```

```
% db eval {SELECT * FROM user} var break
% parray var
var(*)          = userid first_name last_name
var(first_name) = D. Richard
var(last_name)  = Hipp
var(userid)     = 1
%
```

List of result set column names



Use AS to specify alternative column names
in the result set



```
% db eval {  
    SELECT user AS id, first_name AS fname,  
           last_name AS lname  
    FROM user} var break  
  
% parray var  
var(*)      = id fname lname  
var(fname) = D. Richard  
var(lname) = Hipp  
var(id)     = 1  
%
```

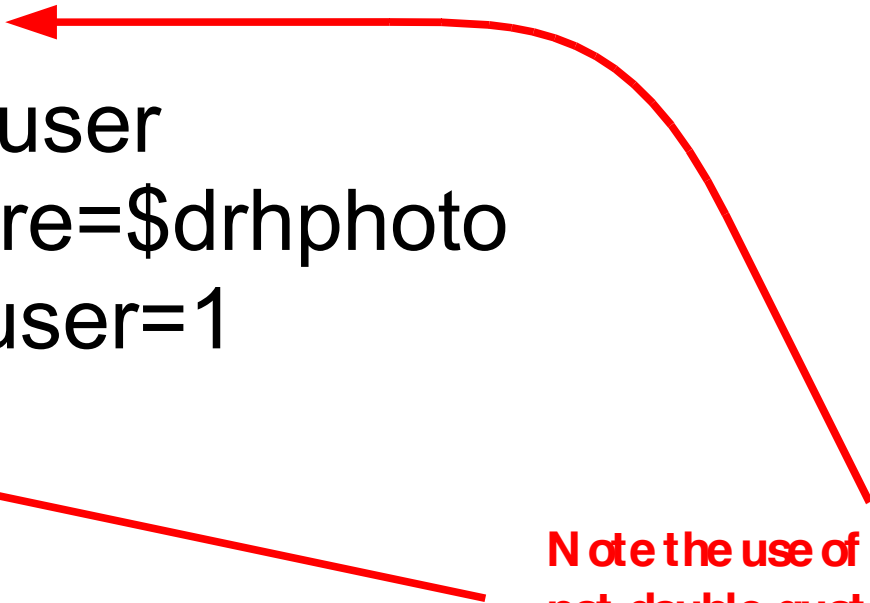
```
db eval {  
  ALTER TABLE user  
  ADD COLUMN picture;  
}
```

user	first_name	last_name	picture
1	D. Richard	Hipp	



New Column Added

```
% set in [open drh.jpg]
% fconfigure $in -translation binary
% set drhphoto [read $in]
% close $in
% db eval {
    UPDATE user
    SET picture=$drhphoto
    WHERE user=1
}
%
```



Note the use of curly-braces,
not double-quotes

SQLite CVSTrac - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.sqlite.org/cvstrac/wiki?p=ConsoleForTclTk

News Projects Talks Stuff next

```
package require sqlite3
set dbname [lindex $argv 0]
sqlite3 db $dbname
set title [file tail $dbname]
source sqlitecon.tcl
sqlitecon::create .console {sqlite> } $title db
wm withdraw .
bind .console <Destroy> {+if {"%W"=="console"} exit}
```

See Also

- [TclDbEdit](#)
- [InteractiveTextFieldEditing](#)

Attachments:

- [sqlitecon.txt](#) 18317 bytes added by drh on 2005-Sep-23 11:41:57 UTC.
- [demo.gif](#) 13365 bytes added by drh on 2005-Sep-23 11:52:43 UTC.

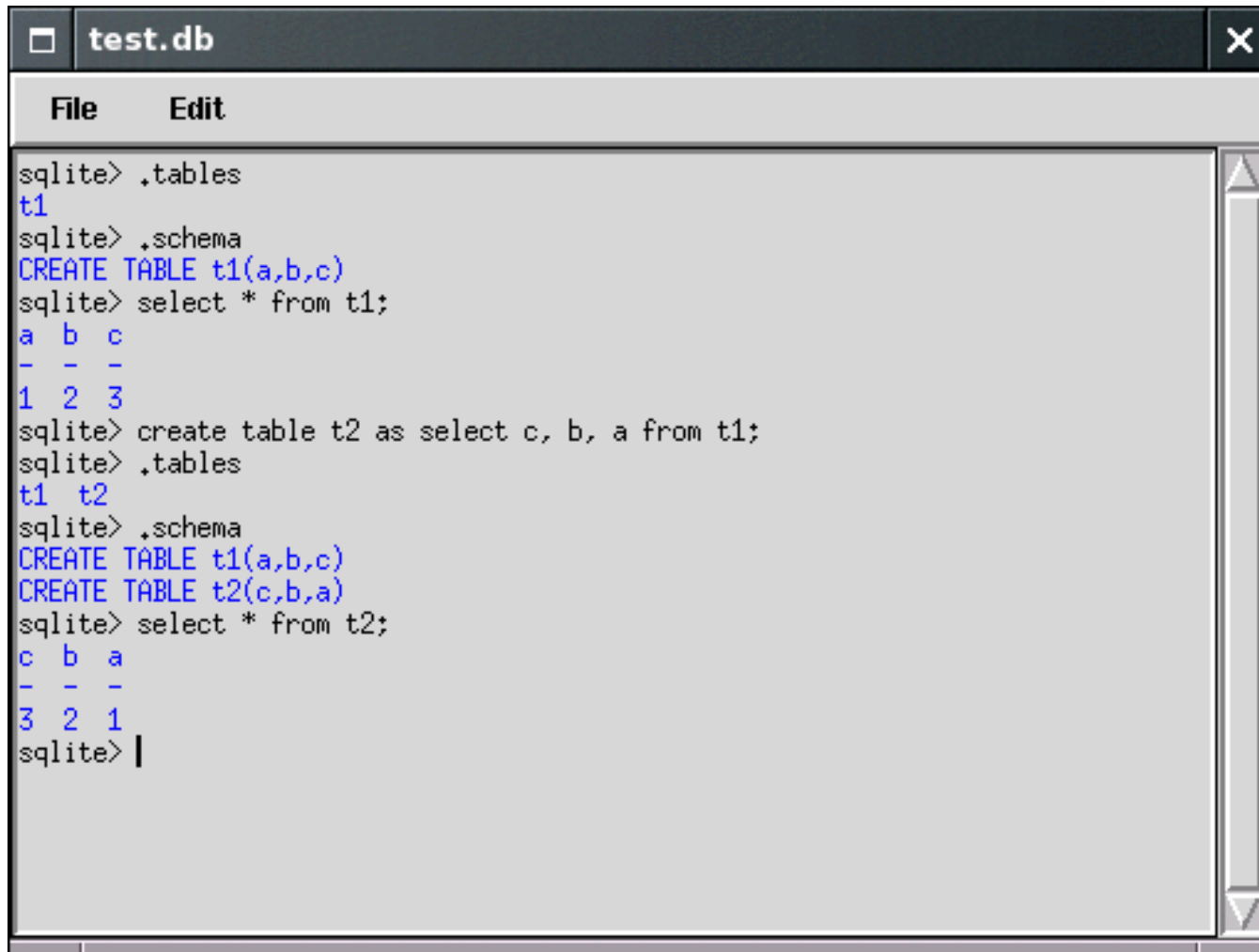
CVSTrac version 1.2.1

Done

Go Here

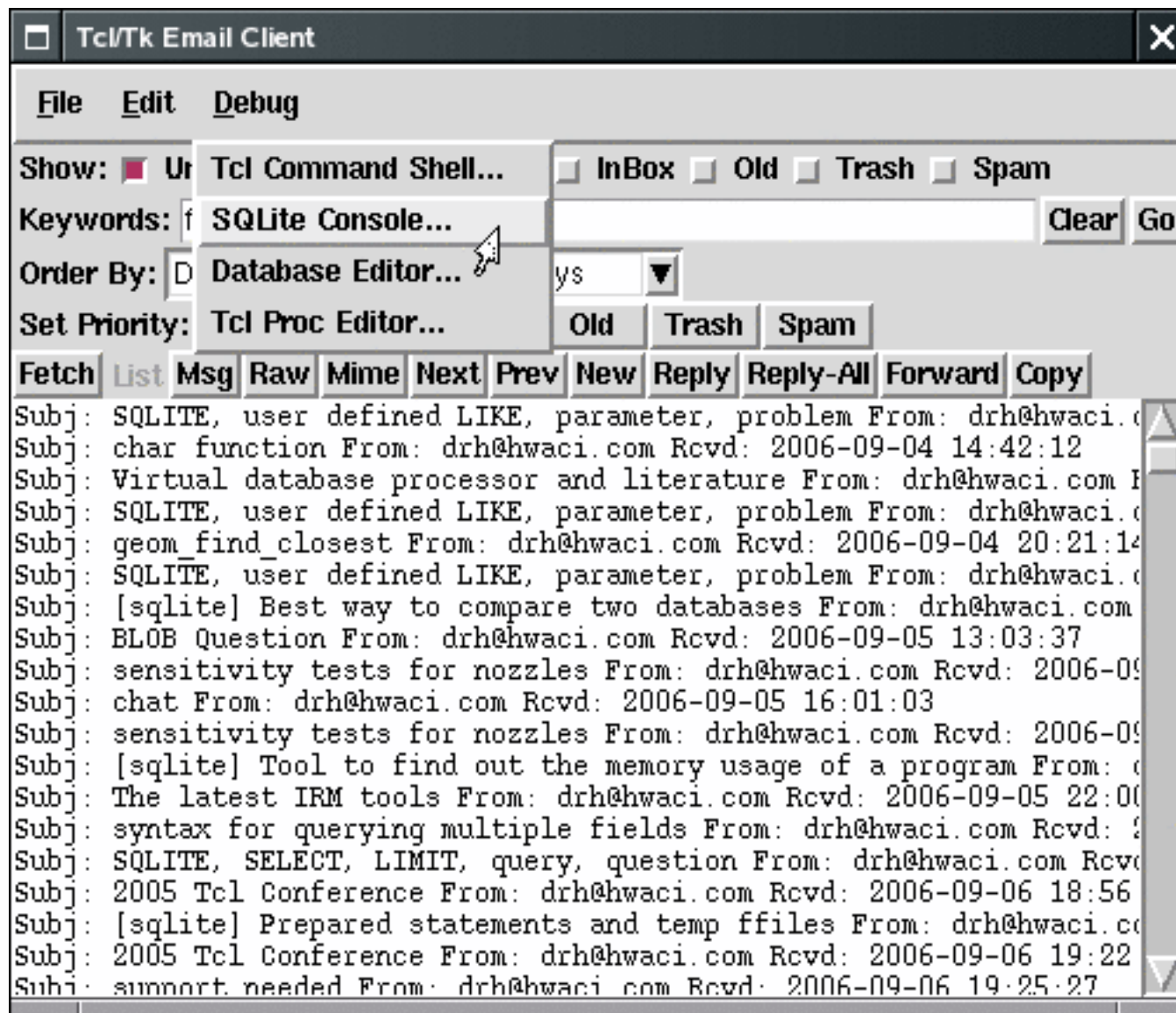
Get This

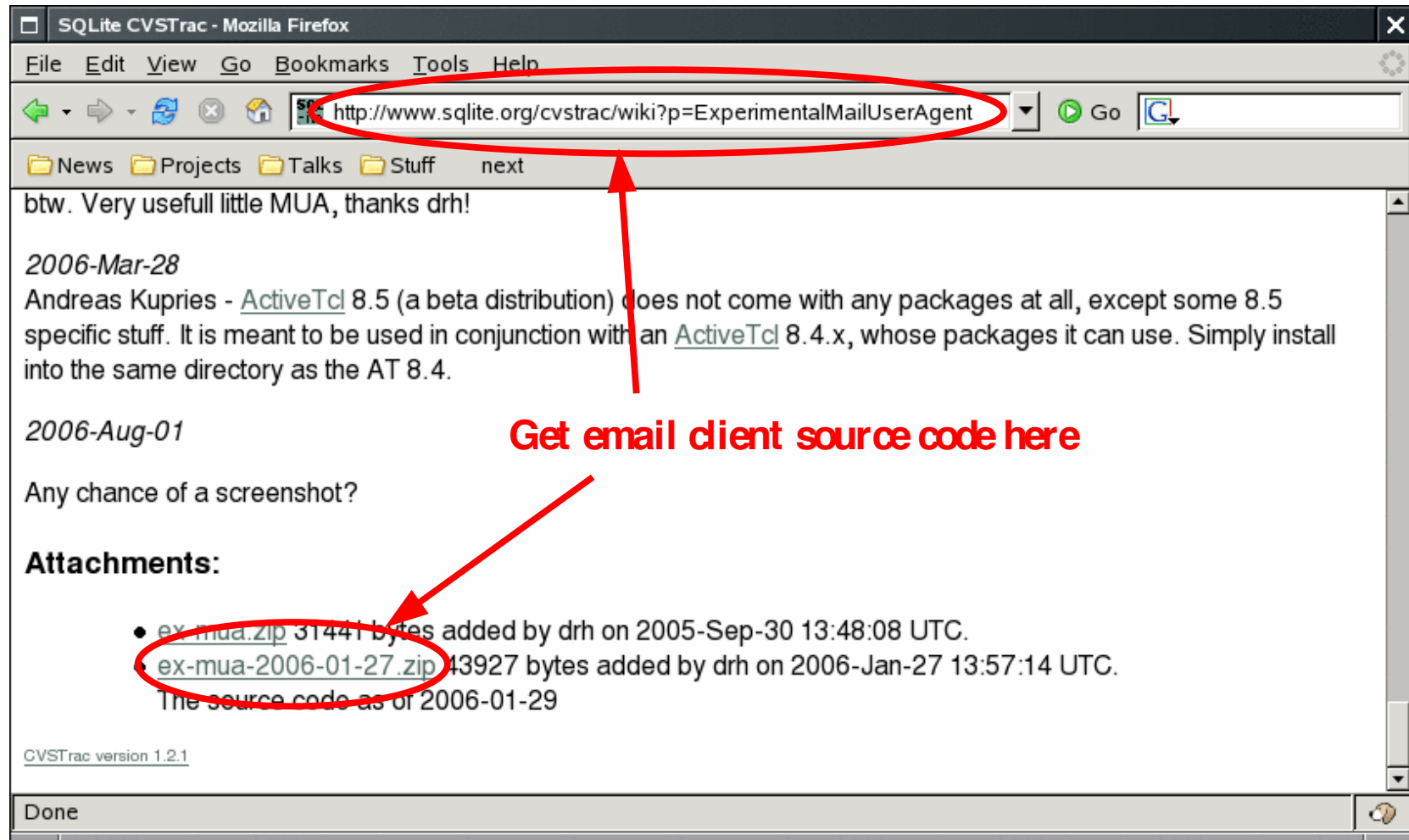

```
% package require Tk
% source sqlitecon.txt
% sqlitecon::create .console {sqlite> } test.db db
%
```



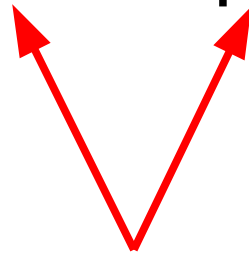
```
File Edit

sqlite> .tables
t1
sqlite> .schema
CREATE TABLE t1(a,b,c)
sqlite> select * from t1;
a b c
- - -
1 2 3
sqlite> create table t2 as select c, b, a from t1;
sqlite> .tables
t1 t2
sqlite> .schema
CREATE TABLE t1(a,b,c)
CREATE TABLE t2(c,b,a)
sqlite> select * from t2;
c b a
- - -
3 2 1
sqlite> |
```

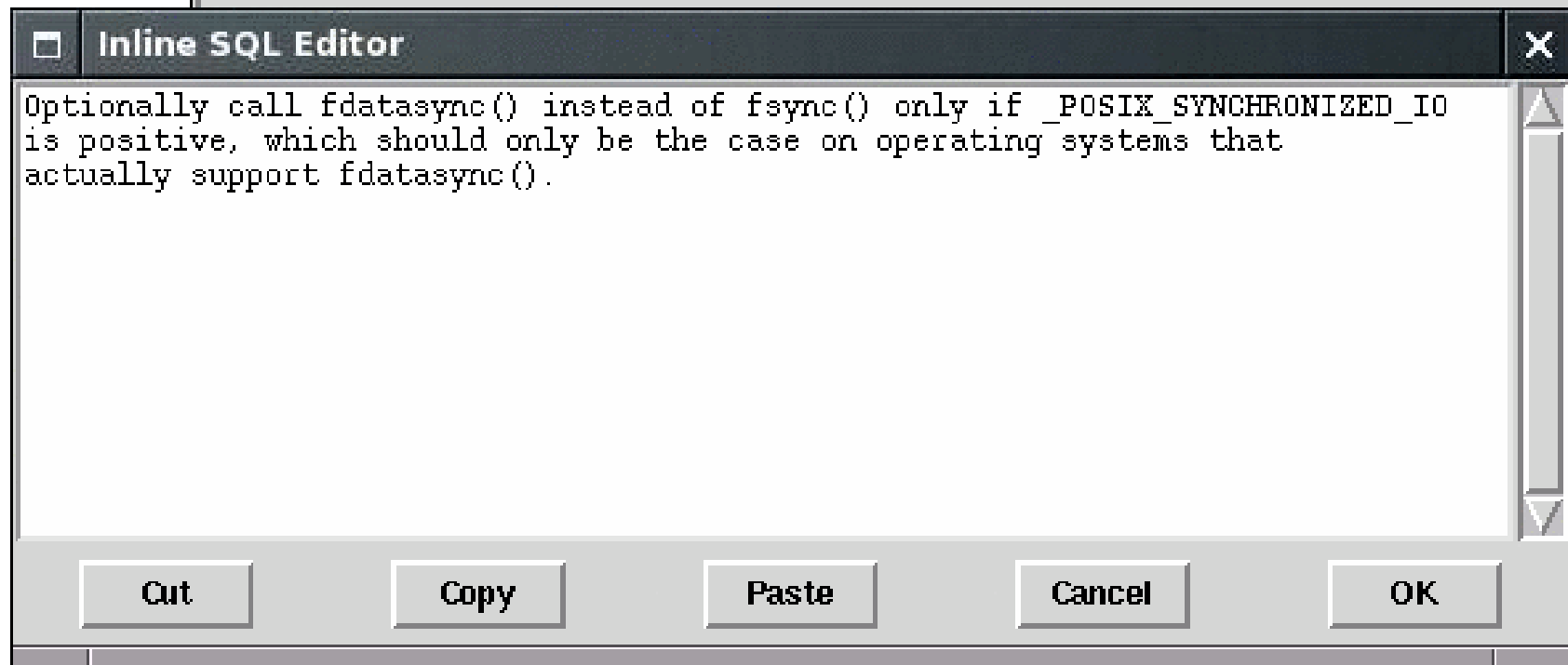
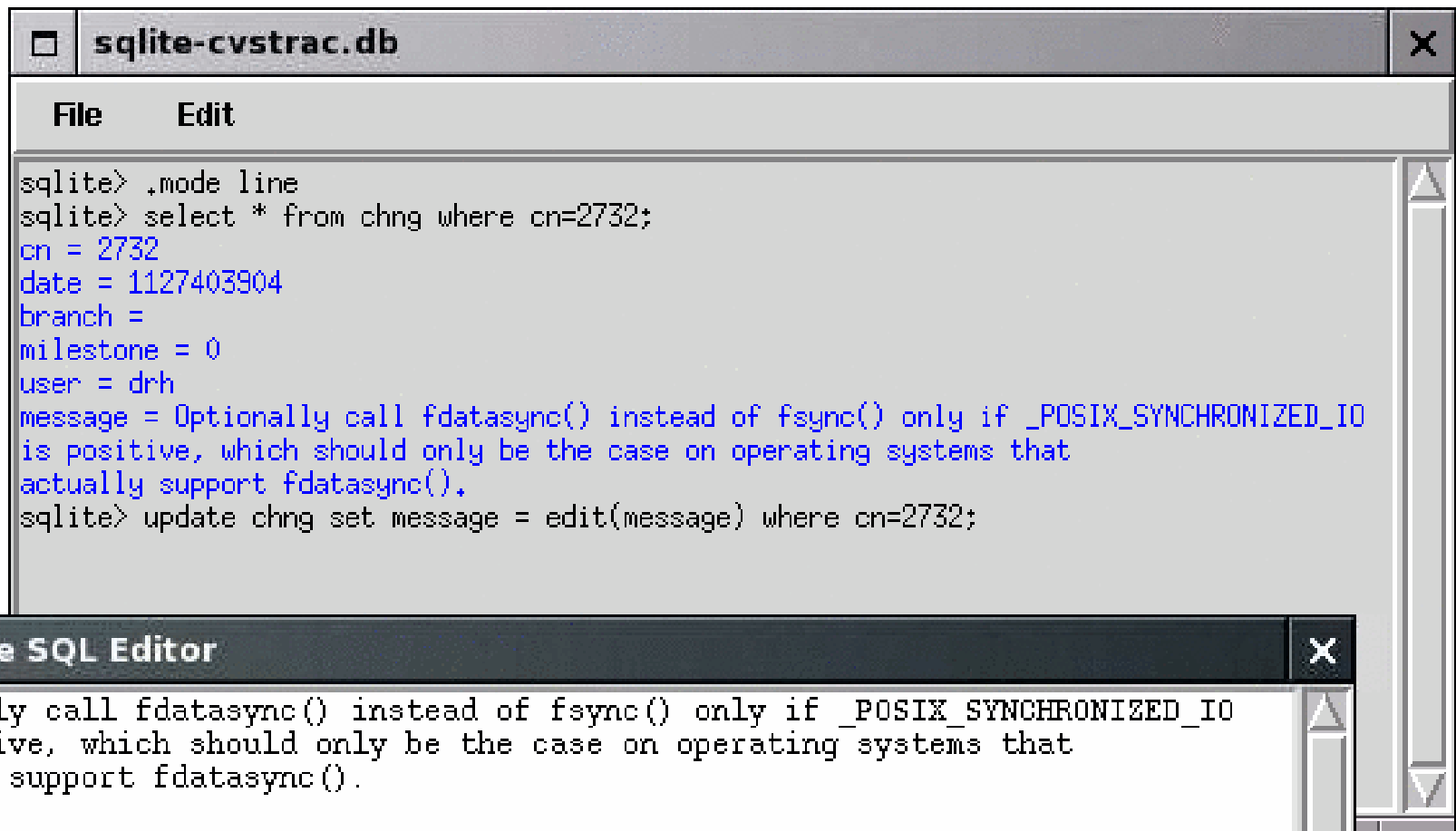


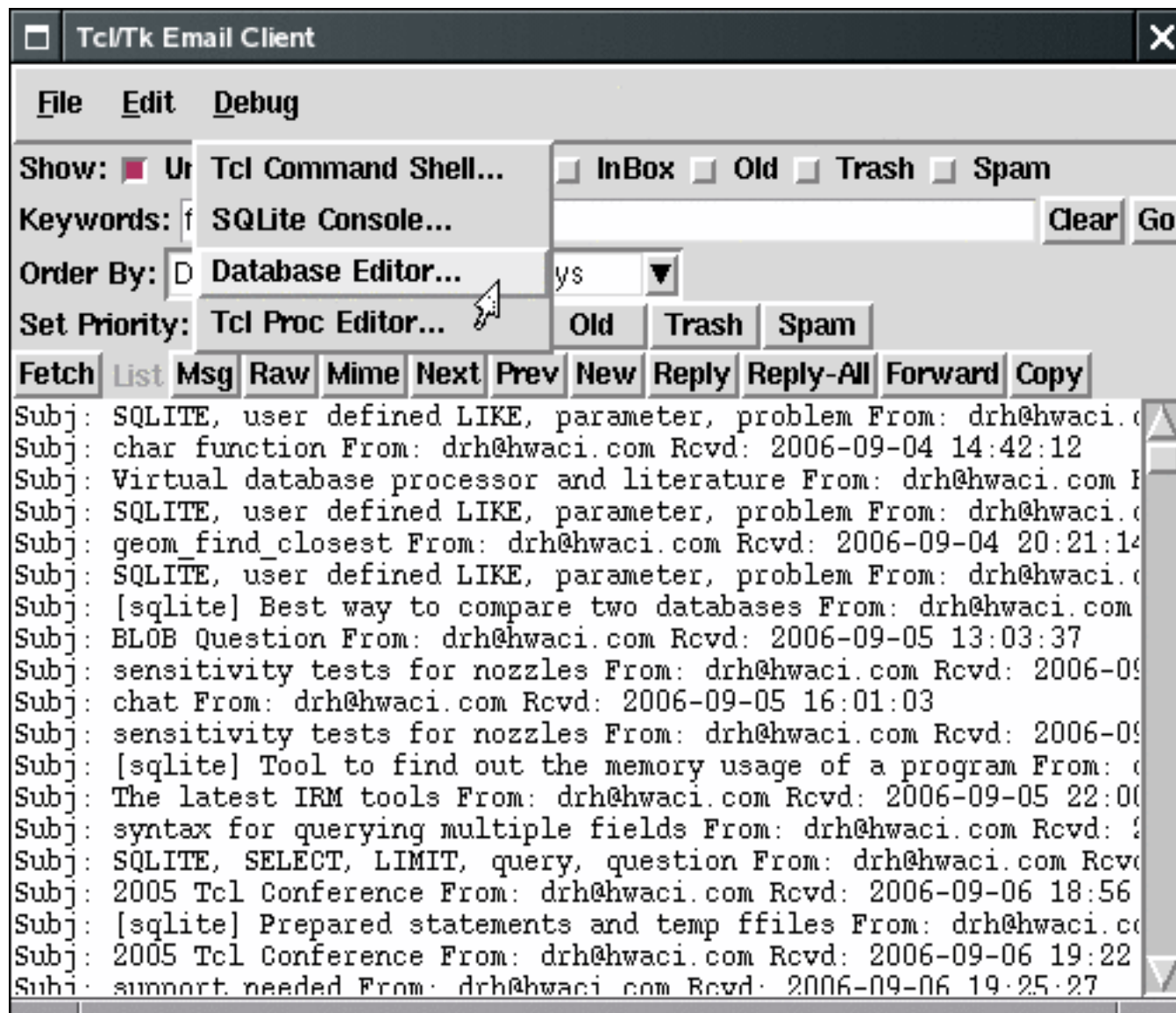


```
proc sqlitecon::_edit {original_text} {  
    # Code here to implement a GUI editor  
    # for $original_text and return the result.  
}  
db function edit ::sqlite::_edit
```



**Create a new SQL function named "edit"
implemented by the T CL proc "::sqlite::_edit"**





Database Editor

Table: **msg** Pattern:

msgid	8
priority	trash
rcvd	2454005.41185
size	2570
nattach	1
userid	4
subject	One Year written replica watches warranty!

Next Prev New Revert Save Delete Shell Close

Database Edit

Table: Item:

msgid	8
priority	trash
rcvd	2454005.41165
size	2570
nattach	1
userid	4
subject	One Year written replica watches warranty!

Next Prev New Revert Save Delete Shell Close

- blob
- email
- email_content
- email_term
- map
- mime
- msg**
- msg_uuid
- user
- var

Database Editor

Table: **msg** Pattern:

msgid	8
pr	005.41185
nattach	1
userid	4
subject	One Year written replica watches warranty!

Search
Edit
Index

Next Prev New Revert Save Delete Shell Close

Database Editor

Table: **msg** Pattern:

8354	msgid	8368
8355	priority	sent
8356	rcvd	2453913.95919
8357	size	946
8358	nattach	0
8359	userid	2
8360	subject	Unit test failures w/legacy file format
8361		
8362		
8363		
8364		
8365		
8366		
8367		
8368		
8369		
8370		
8371		
8372		
8373		

Next Prev New Revert Save Delete Shell Close

Database Editor

Table: **msg** Pattern:

msgid	8
priority	trash
rcvd	2454005.41185
size	2570
nattach	1
userid	4
subject	One Year written replica watches warranty!

Search

Edit New Revert Save Delete Shell Close

Index

Database Editor X

Table: Pattern:

msgid	<input type="text" value="8"/>
priority	<input type="text" value="trash"/>
rcvd	<input type="text" value="2454005.41185"/>
size	<input type="text" value="2570"/>
nattach	<input type="text" value="1"/>
userid	<input type="text" value="4"/>
subject	<input type="text" value="One Year written replica watches warranty!"/>

Edit field "subject" X

Database Editor

Table: msg Pattern: 4591

msgid	8
ppid	005.41185
size	1070
nattach	1
userid	4
subject	One Year written replica watches warranty!

Search
Edit
Index

Next Prev New Revert Save Delete Shell Close

Database Editor

Table: **msg** Pattern: 4591

msgid	4591
priority	old
rcvd	2453657.19627
size	1995
nattach	0
userid	1131
subject	[sqlite] os_unix.c / vxworks

Next Prev New Revert Save Delete Shell Close

Database Editor

Table: **msg** Pattern:

msgid	4591
priority	old
rcvd	2453657.19627
size	1995
nattach	0
userid	1131
subject	[sqlite] os_unix.c / vxworks

Next Prev New Revert Save Delete Shell Close

Database Editor

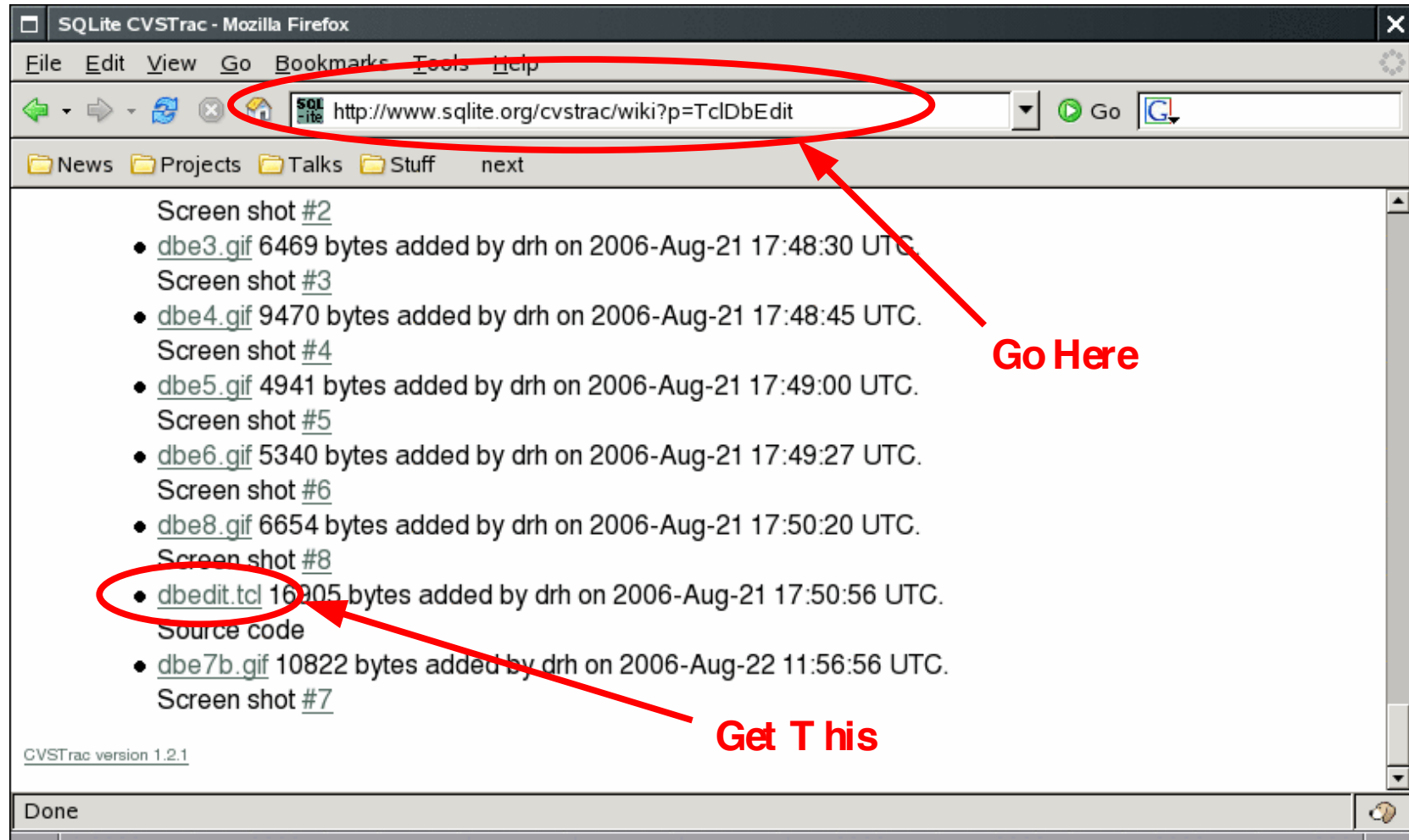
Table: msg Pattern:

msgid 4591

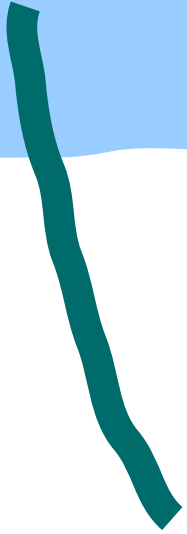
SQLite Console

File Edit

```
sqlite> |
```

The Land Of Tcl/Tk



Full Text Search

Full Text Search

Q: What is Full Text Search?

-

- **A:** In brief: what Google does.

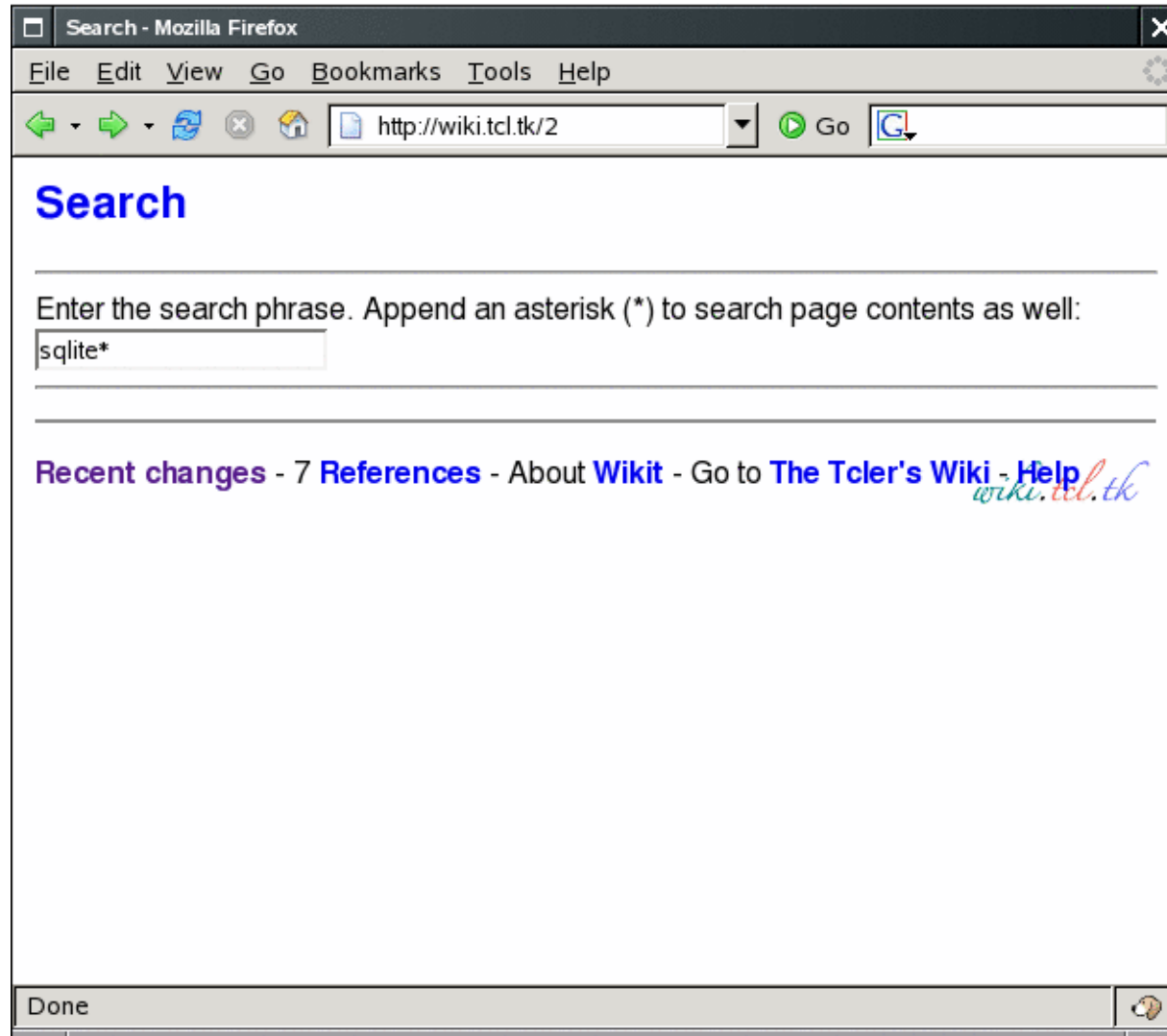
Full Text Search

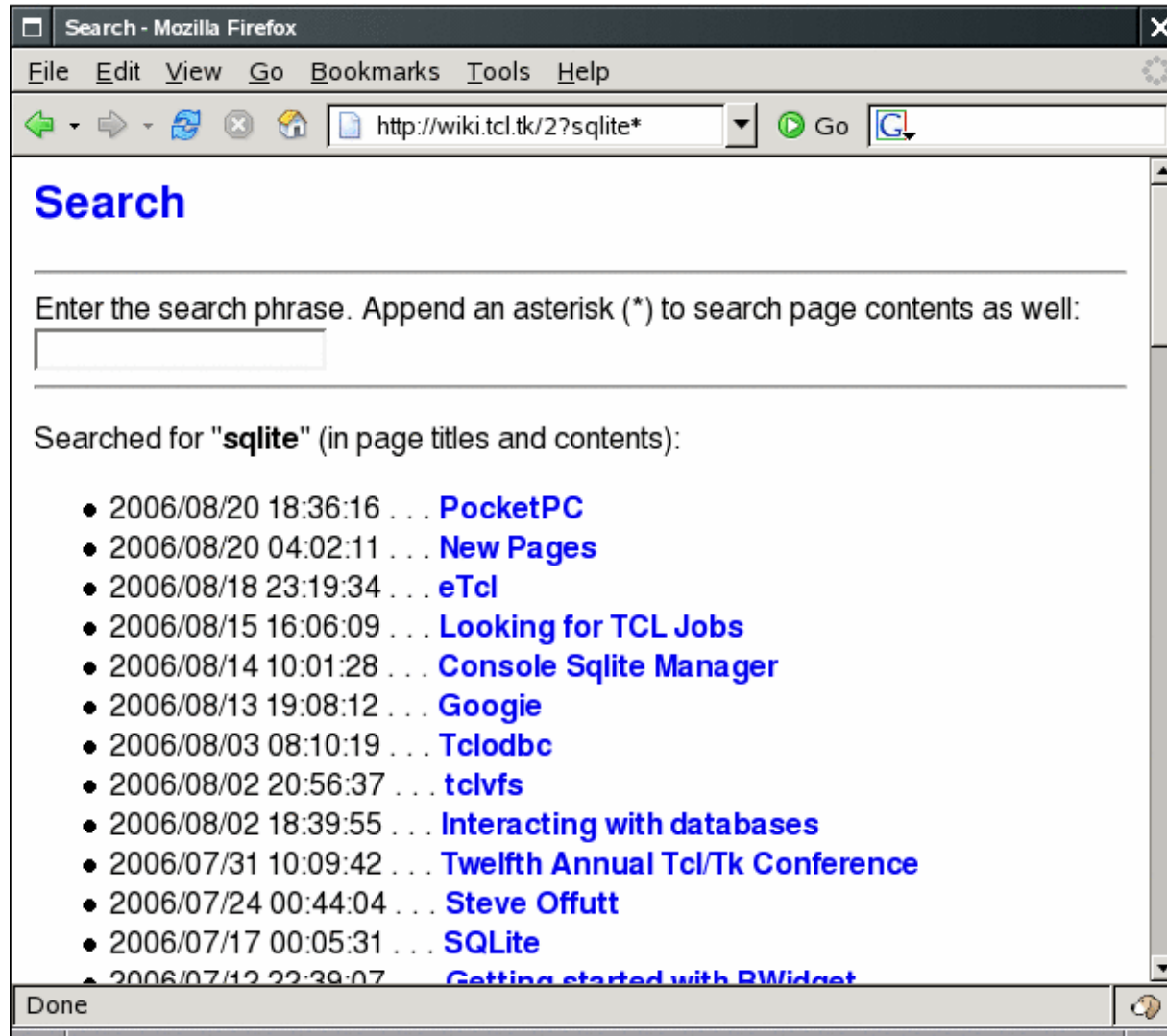
- **Q:** Why is full text search important?
-
- **A:** Internet search engines have users spoiled. Modern applications need to support full text search in order to be competitive

Full Text Search

- **Q:** Is it difficult to implement?
-
- **A:** It is tricky to make it efficient for large data sets and getting internationalization right is hard

Single Term Search





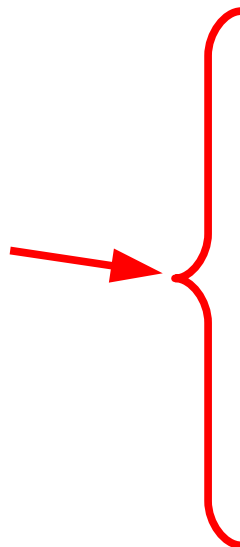
Multiple Term Search

Two or more
search terms
per query



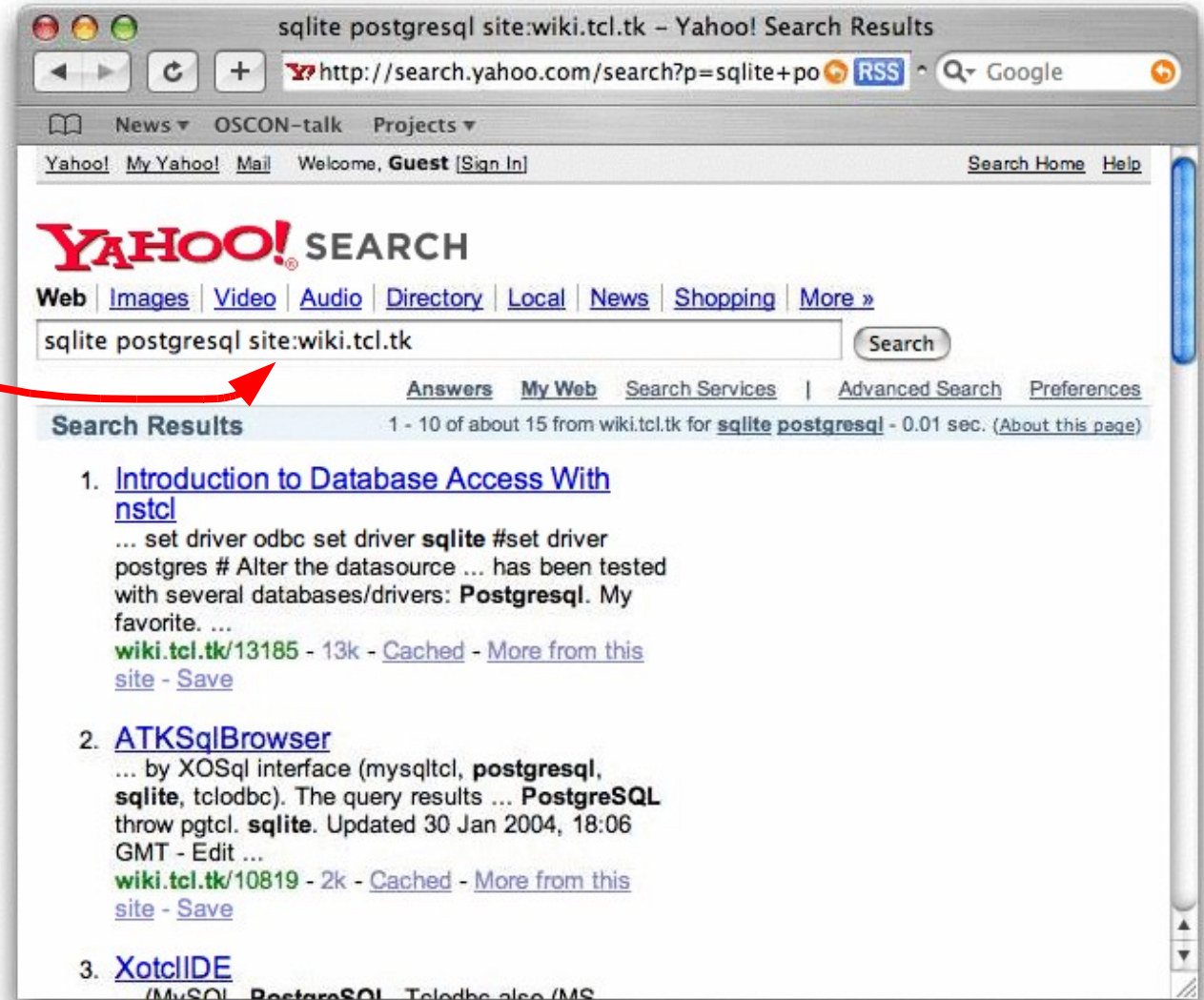
The screenshot shows a web browser window titled "sqlite postgresql - Yahoo! Search Results". The address bar contains the URL "http://search.yahoo.com/search?p=sqlite+po". The search input field contains the text "sqlite postgresql" and is circled in red. Below the search bar, the search results are displayed. The first result is "SQLite Database Speed Comparison" with a snippet: "... the synchronous SQLite times are for comparison against PostgreSQL (which is ... Prior versions of SQLite used to be slower than PostgreSQL and MySQL on this ...". The second result is "sqlite: A command-line access program for SQLite databases" with a snippet: "... PostgreSQL, so you can also use the .dump command to export an SQLite database ... way and export a PostgreSQL database into SQLite using the pg_dump utility. ...". The third result is "dba: Linux SQLite driver - alpha version" with a snippet: "OpenOffice.org: The Open Office Suite".

Each result
contains every
search term



Restrict Search Terms

“site:” keyword
restricts search to
a single website



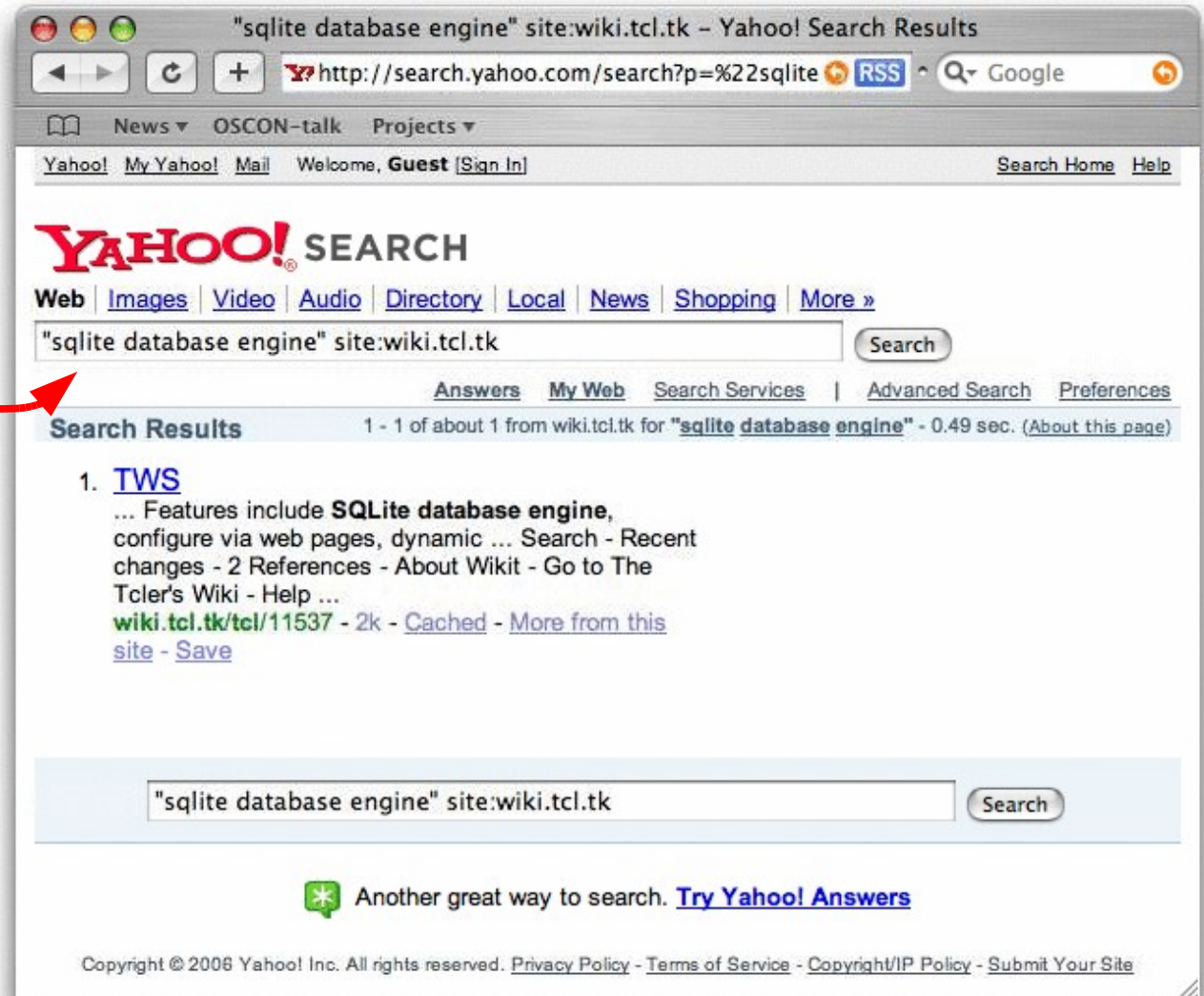
The screenshot shows a web browser window with the title "sqlite postgresql site:wiki.tcl.tk - Yahoo! Search Results". The address bar contains "http://search.yahoo.com/search?p=sqlite+po" and a search button. The search results are displayed on the page, showing three results from wiki.tcl.tk. A red arrow points from the text on the left to the search query in the search box.

Search Results

- [Introduction to Database Access With nstcl](#)
... set driver odbc set driver **sqlite** #set driver postgres # Alter the datasource ... has been tested with several databases/drivers: **Postgresql**. My favorite. ...
wiki.tcl.tk/13185 - 13k - [Cached](#) - [More from this site](#) - [Save](#)
- [ATKSqlBrowser](#)
... by XOSql interface (mysqcl, **postgresql**, **sqlite**, tclodbc). The query results ... **PostgreSQL** throw pgcl. **sqlite**. Updated 30 Jan 2004, 18:06 GMT - Edit ...
wiki.tcl.tk/10819 - 2k - [Cached](#) - [More from this site](#) - [Save](#)
- [XotclIDE](#)
(MySQL, PostgreSQL, Tclodbc also /MS

Phrase Search

Two or more words
within double-quotes
must appear as
written

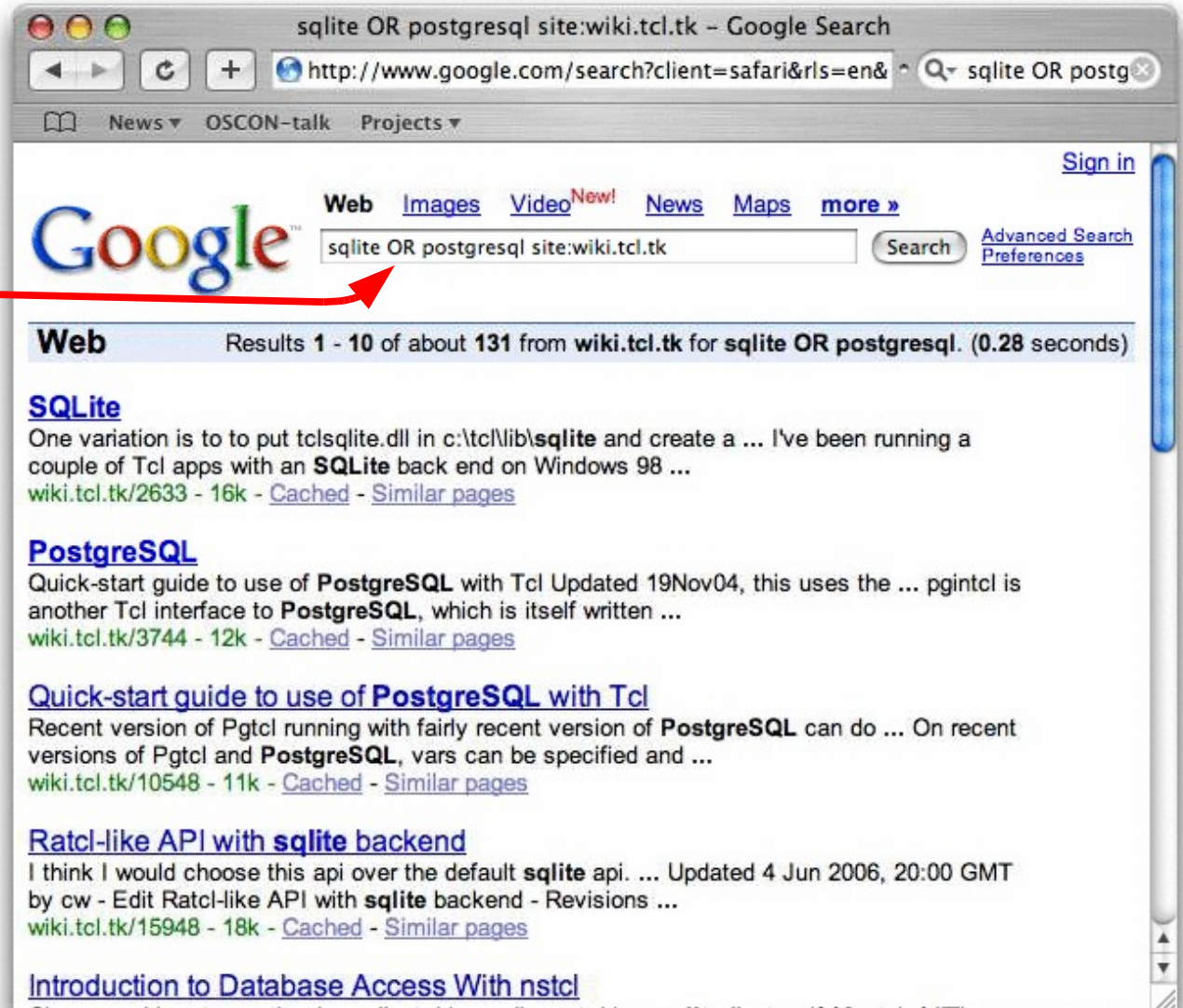


The screenshot shows a browser window with the title "sqlite database engine" site:wiki.tcl.tk - Yahoo! Search Results. The address bar contains the URL "http://search.yahoo.com/search?p=%22sqlite". The search bar contains the query "sqlite database engine" site:wiki.tcl.tk. The search results show one result from wiki.tcl.tk for "sqlite database engine".

1. [TWS](#)
... Features include **SQLite database engine**, configure via web pages, dynamic ... Search - Recent changes - 2 References - About Wikit - Go to The Tcler's Wiki - Help ...
wiki.tcl.tk/tcl/11537 - 2k - [Cached](#) - [More from this site](#) - [Save](#)

OR Search

Two or more words connected by "OR" means only one is required per page

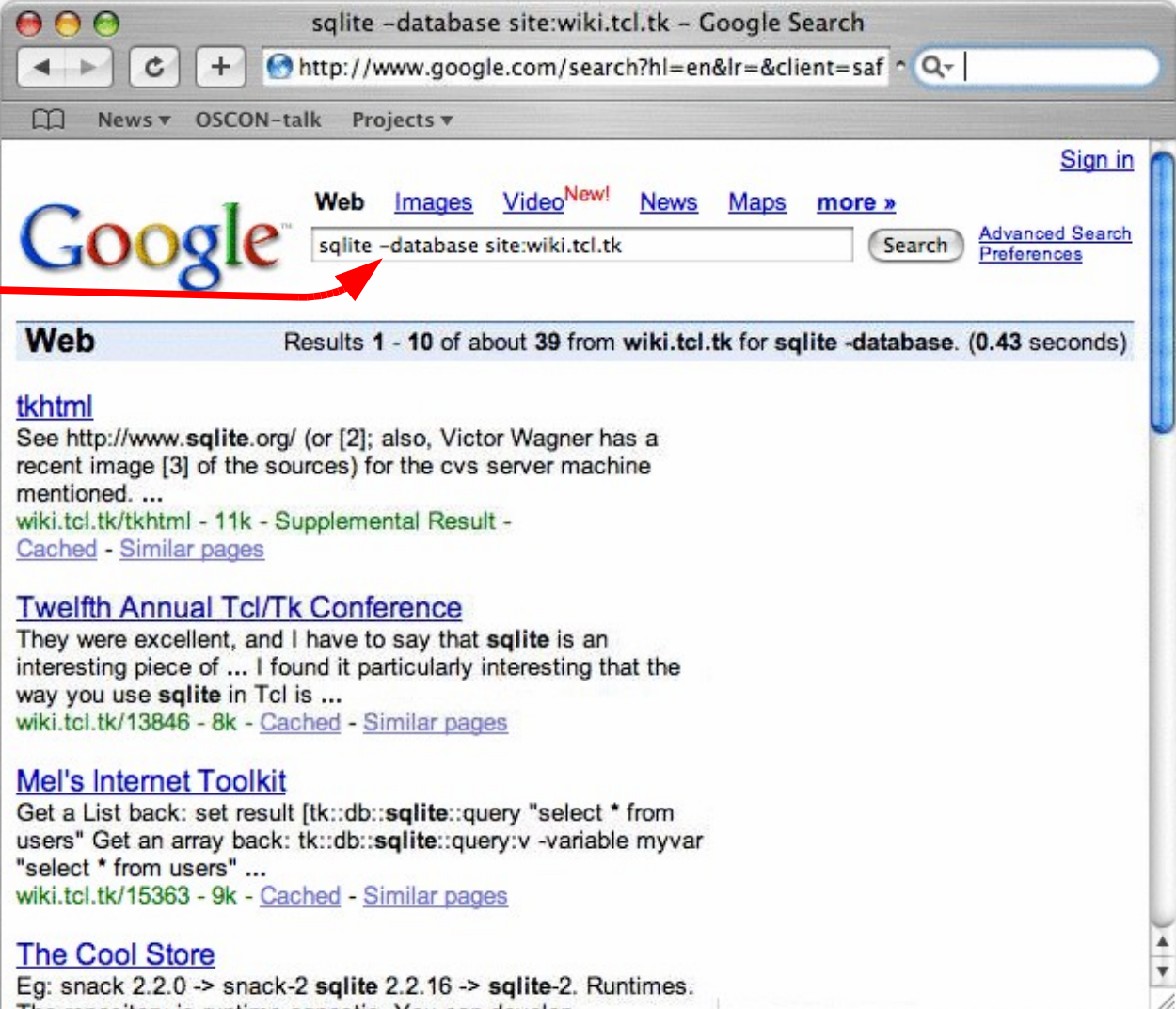


The screenshot shows a Safari browser window with the address bar containing 'sqlite OR postgresql site:wiki.tcl.tk - Google Search'. The search bar contains the same text. The search results are displayed under the 'Web' tab, showing 'Results 1 - 10 of about 131 from wiki.tcl.tk for sqlite OR postgresql. (0.28 seconds)'. The results list includes:

- SQLite**
One variation is to to put tclsqlite.dll in c:\tcl\lib\sqlite and create a ... I've been running a couple of Tcl apps with an **SQLite** back end on Windows 98 ...
wiki.tcl.tk/2633 - 16k - [Cached](#) - [Similar pages](#)
- PostgreSQL**
Quick-start guide to use of **PostgreSQL** with Tcl Updated 19Nov04, this uses the ... pgintcl is another Tcl interface to **PostgreSQL**, which is itself written ...
wiki.tcl.tk/3744 - 12k - [Cached](#) - [Similar pages](#)
- Quick-start guide to use of PostgreSQL with Tcl**
Recent version of Pgtcl running with fairly recent version of **PostgreSQL** can do ... On recent versions of Pgtcl and **PostgreSQL**, vars can be specified and ...
wiki.tcl.tk/10548 - 11k - [Cached](#) - [Similar pages](#)
- Ratcl-like API with sqlite backend**
I think I would choose this api over the default **sqlite** api. ... Updated 4 Jun 2006, 20:00 GMT by cw - Edit Ratcl-like API with **sqlite** backend - Revisions ...
wiki.tcl.tk/15948 - 18k - [Cached](#) - [Similar pages](#)
- Introduction to Database Access With nstcl**

Excluding Terms

“-” before a word means show only documents that lack that word



The screenshot shows a Google search interface. The search bar contains the query "sqlite -database site:wiki.tcl.tk". A red arrow points to the minus sign before "database". Below the search bar, the results are displayed under the heading "Web". The first result is titled "tkhtml" and includes a snippet: "See http://www.sqlite.org/ (or [2]; also, Victor Wagner has a recent image [3] of the sources) for the cvs server machine mentioned. ...". The second result is titled "Twelfth Annual Tcl/Tk Conference" and includes a snippet: "They were excellent, and I have to say that **sqlite** is an interesting piece of ... I found it particularly interesting that the way you use **sqlite** in Tcl is ...". The third result is titled "Mel's Internet Toolkit" and includes a snippet: "Get a List back: set result [tk::db::sqlite::query "select * from users" Get an array back: tk::db::sqlite::query:v -variable myvar "select * from users" ...". The fourth result is titled "The Cool Store" and includes a snippet: "Eg: snack 2.2.0 -> snack-2 **sqlite** 2.2.16 -> **sqlite**-2. Runtimes. The repository is runtime agnostic. You can develop".

Snippets

The screenshot shows a web browser window with the title 'sqlite -database site:wiki.tcl.tk - Google Search'. The address bar contains the URL 'http://www.google.com/search?hl=en&lr=&client=saf'. The search bar contains the query 'sqlite -database site:wiki.tcl.tk'. The search results are displayed under the heading 'Web' and show 'Results 1 - 10 of about 39 from wiki.tcl.tk for sqlite -database. (0.43 seconds)'. Three red arrows point to the first three search results:

- tkhtml**
See <http://www.sqlite.org/> (or [2]; also, Victor Wagner has a recent image [3] of the sources) for the cvs server machine mentioned. ...
[wiki.tcl.tk/tkhtml](#) - 11k - Supplemental Result - [Cached](#) - [Similar pages](#)
- Twelfth Annual Tcl/Tk Conference**
They were excellent, and I have to say that **sqlite** is an interesting piece of ... I found it particularly interesting that the way you use **sqlite** in Tcl is ...
[wiki.tcl.tk/13846](#) - 8k - [Cached](#) - [Similar pages](#)
- Mel's Internet Toolkit**
Get a List back: set result [tk::db::sqlite::query "select * from users" Get an array back: tk::db::sqlite::query:v -variable myvar "select * from users" ...
[wiki.tcl.tk/15363](#) - 9k - [Cached](#) - [Similar pages](#)

The fourth result is partially visible:

- The Cool Store**
Eg: snack 2.2.0 -> snack-2 **sqlite** 2.2.16 -> **sqlite**-2. Runtimes. The repository is runtime-agnostic. You can develop

“Snippets” are excerpts of the document that match the search terms

Proximity search

- Some search engines allow queries like this:
 - sqlite NEAR postgresql
- Matches when “sqlite” occurs near (within 10 words of) “postgresql” in the document
- No more technically difficult than a phrase search

Elements Of Full Text Search

- Tokenizing
- Lexicon mapping
- Indexing and querying
- Snippet generation
- Scoring

2006-Jun-19 - New Book About SQLite

<p>

<i>The Definitive Guide to SQLite</i>, a new book by

Mike Owens.

is now available from Apress.

The books covers the latest SQLite internals as well as

the native C interface and bindings for PHP, Python,

Perl, Ruby, Tcl, and Java. Recommended.

</p>

<hr width="50%">

2006-Jun-19 - New Book About SQLite

<p>

<i>The Definitive Guide to SQLite</i>, a new book by

Mike Owens.

is now available from Apress.

The books covers the latest SQLite internals as well as the native C interface and bindings for PHP, Python, Perl, Ruby, Tcl, and Java. Recommended.

</p>

<hr width="50%">

2006-Jun-19 - New Book About SQLite

<http://www.apress.com/book/bookDisplay.html?bID=10130>

The Definitive Guide to SQLite, a new book by

<http://www.mikesclutter.com> Mike Owens.

is now available from <http://www.apress.com> Apress.

The book covers the latest SQLite internals as well as the native C interface and bindings for PHP, Python, Perl, Ruby, Tcl, and Java. Recommended.



SQLite

SQLite 是個超輕量級的資料庫程式,因為我想在NB上寫Rails,但是又不想裝MySQL好重,第一個就想到SQLite。關於SQLite的中文介紹,可以參考 [簡介SQLite](#) 和 [SQLite使用教學](#)。基本上 sqlite 就是一隻 command-line 程式而已,再加上一個資料檔(即一個資料庫)。

要在Rails中使用,首先去 [SQLite](#) 網站下載執行檔 跟 DLL檔,即 sqlite3.exe 跟 sqlite3.dll,放到 C:/ruby/bin 下。

接著安裝 sqlite for Ruby gem,執行 `gem install sqlite3-ruby`,並選擇 win32 的版本。

最後設定 database.yml :

<p>

</p>

<p>SQLite 是個超輕量級的資料庫程式，因為我想在 NB 上寫 Rails ，
但是又不想裝 MySQL 好重，第一個就想到 SQLite 。關於 SQLite 的中文介紹，可以參考

 簡介 SQLite

和

 SQLite 使用教學

。基本上 sqlite 就是一隻 command-line 程式而已，再加上一個資料檔

(即一個資料庫) 。 </p>

<p> 要在 Rails 中使用，首先去

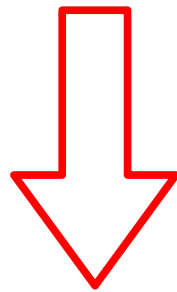
SQLite 網站下載 執行檔 跟 DLL 檔

，即 sqlite3.exe 跟 sqlite3.dll ，放到 C:/ruby/bin 下。 </p>

<p> 接著安裝 sqlite for Ruby gem ，執行 gem install sqlite3-ruby ，並選擇 win32
的版本。 </p>

<p> 最後設定 database.yml :</p>

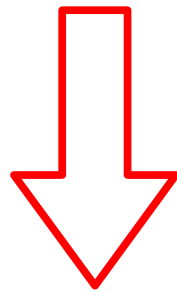
2006 Jun 19 New Book About SQLite The Definitive Guide to SQLite a new book by Mike Owens is now available from Apress The books covers the latest SQLite internals as well as the native C interface and bindings for PHP Python Perl Ruby Tcl and Java Recommended



Case Folding

2006 jun 19 new book about sqlite the definitive guide to sqlite a new book by mike owens is now available from apress the books covers the latest sqlite internals as well as the native c interface and bindings for php python perl ruby tcl and java recommended

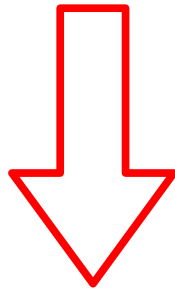
2006 jun 19 new book about sqlite the definitive guide to sqlite a new book by mike owens is now available from apress the books covers the latest sqlite internals as well as the native c interface and bindings for php python perl ruby tcl and java recommended



Stemming

2006 jun 19 new book about sqlite the definit guid to sqlite a new book by mike owen is now avail from apress the book cover the latest sqlite intern as well as the nativ c interfac and bind for php python perl rubi tcl and java recommend

2006 jun 19 new book about sqlite the definit guid
to sqlite a new book by mike owen is now avail from
apress the book cover the latest sqlite intern as well
as the nativ c interfac and bind for php python perl
rubi tcl and java recommend



Remove Stop Words

2006 jun 19 new book about sqlite --- definit guid
-- sqlite - new book -- mike owen -- now avail ---
apress --- book cover --- latest sqlite intern -- well
-- --- nativ - interfac --- bind --- php python perl
rubi tcl --- java recommend

Posting Lists

2006: doc52, doc871, doc1128, doc1137, doc2351

jun: doc551, doc2351

19: doc88, doc92, doc93, doc1443, doc2351

new: doc11, doc31, doc35, ..., doc2337, doc2351

book: doc192, doc331, doc409, ..., doc2196, doc2351

■
■
■

recommend: doc1872, doc2351

Queries

Recommendation For A New Book



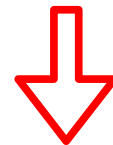
CaseFolding

recommendation for a new book



stemmer

recommend for a new book

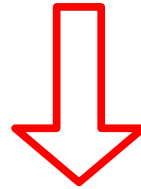


Stop words

recommend new book

Queries

recommend new book

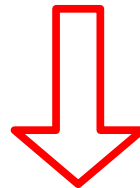


**Retrieve posting lists for
each word in the query**

recommend: doc1872, doc2351

new: doc11, doc31, doc35, ..., doc2337, doc2351

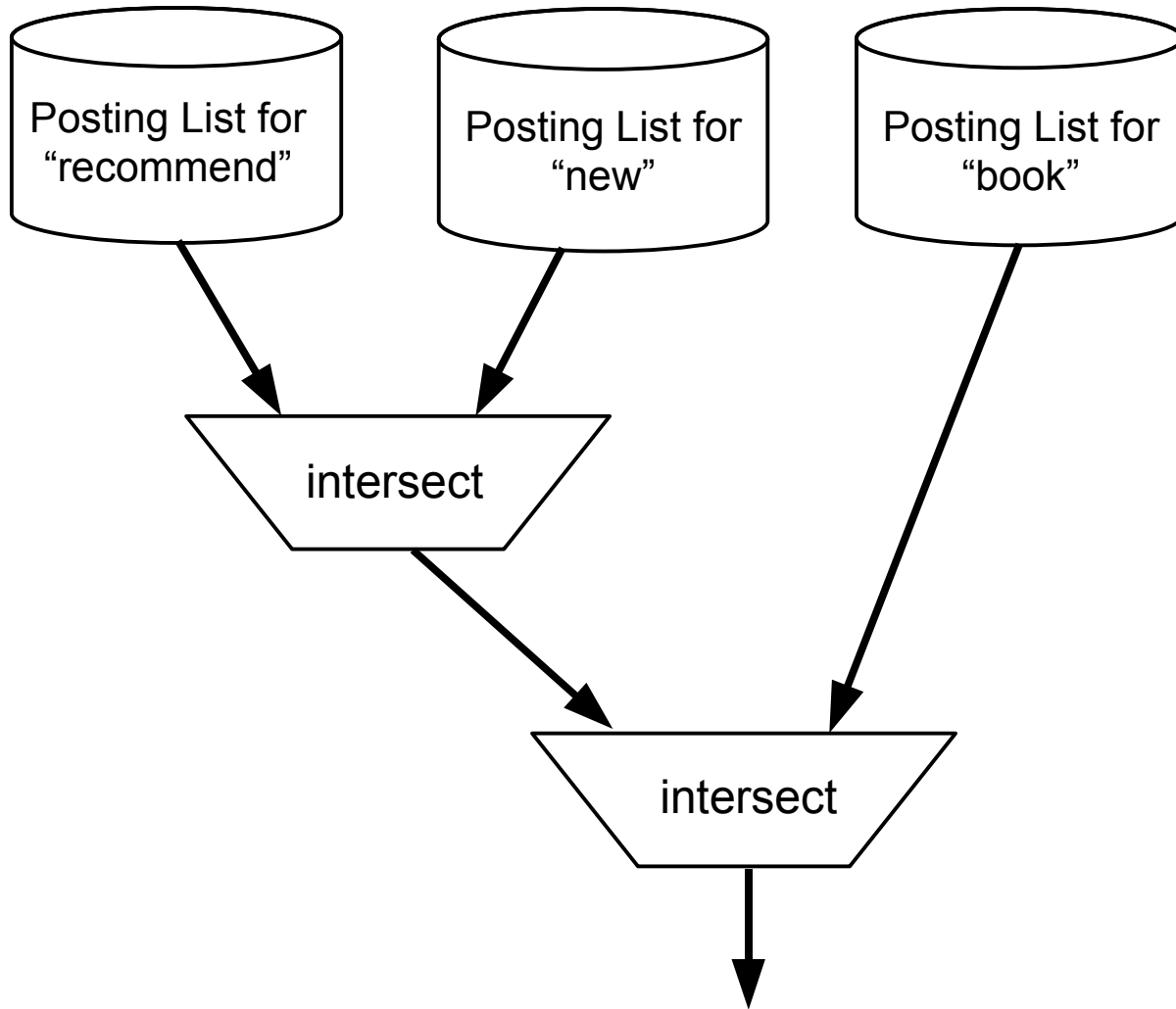
book: doc192, doc331, doc409, ..., doc2196, doc2351



**Take the intersection of
the posting lists**

doc2351

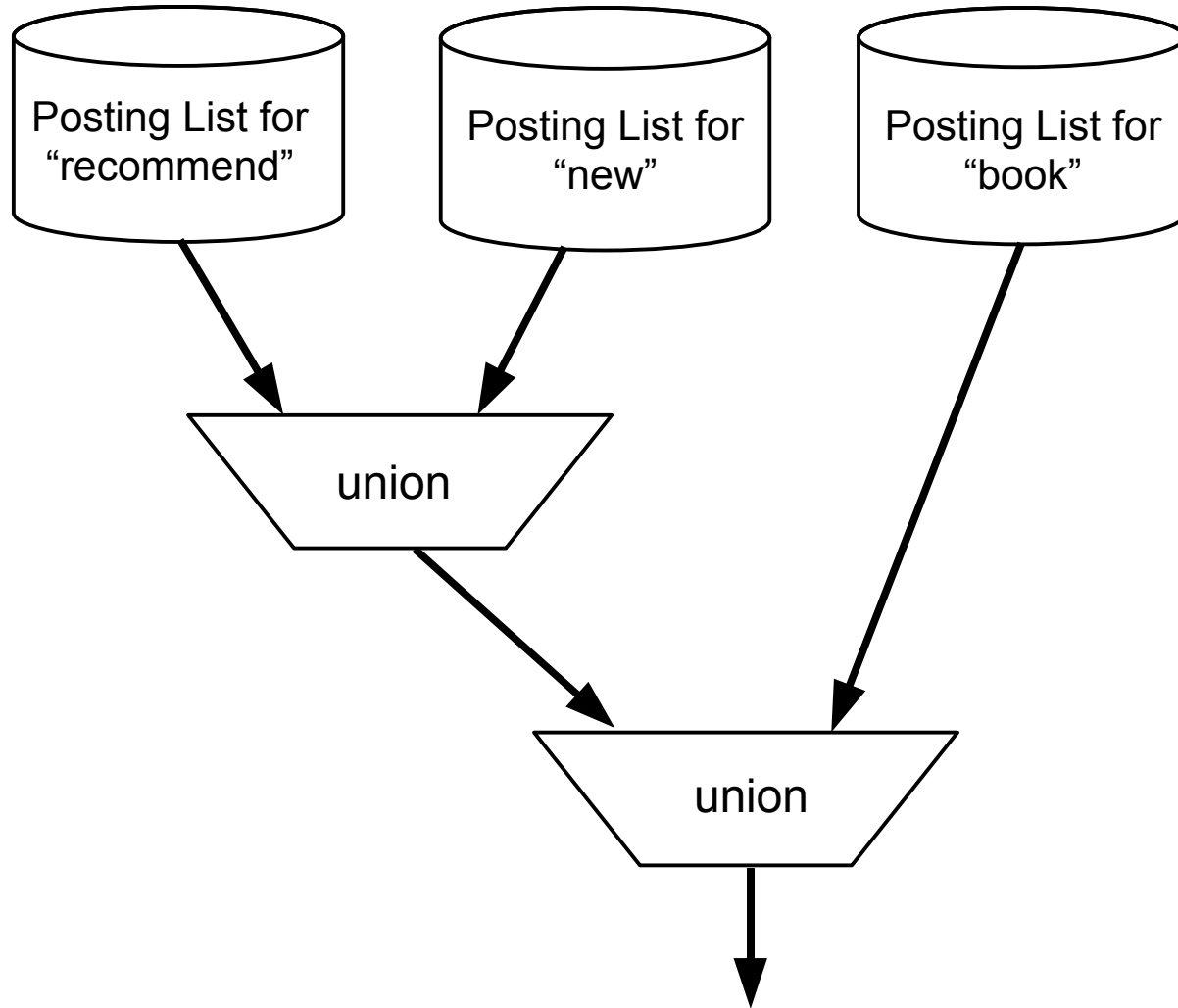
recommend new book



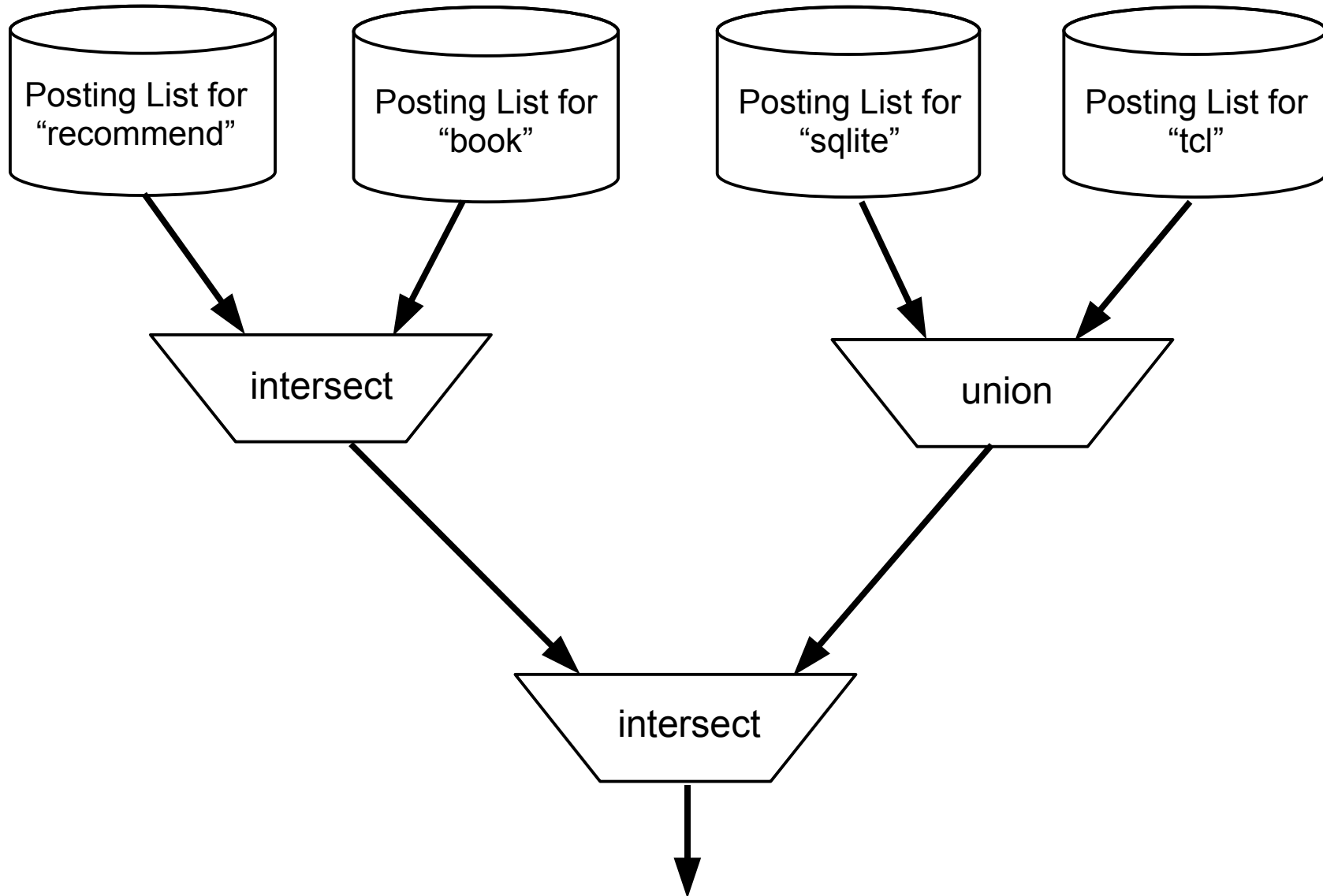
OR Queries

- Simply take the union of the posting lists instead of the intersection

recommend OR new OR book



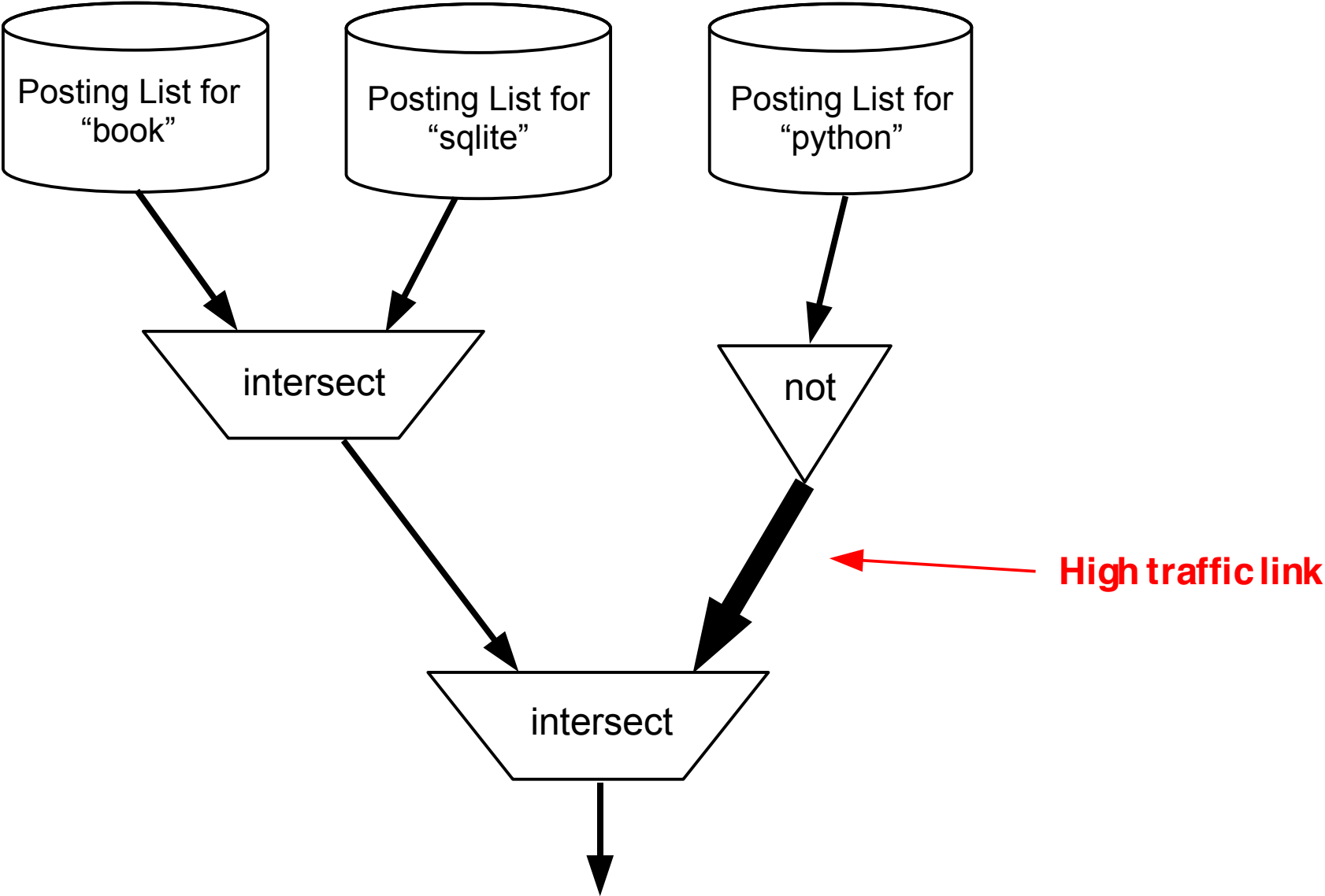
recommend book (sqlite OR tcl)



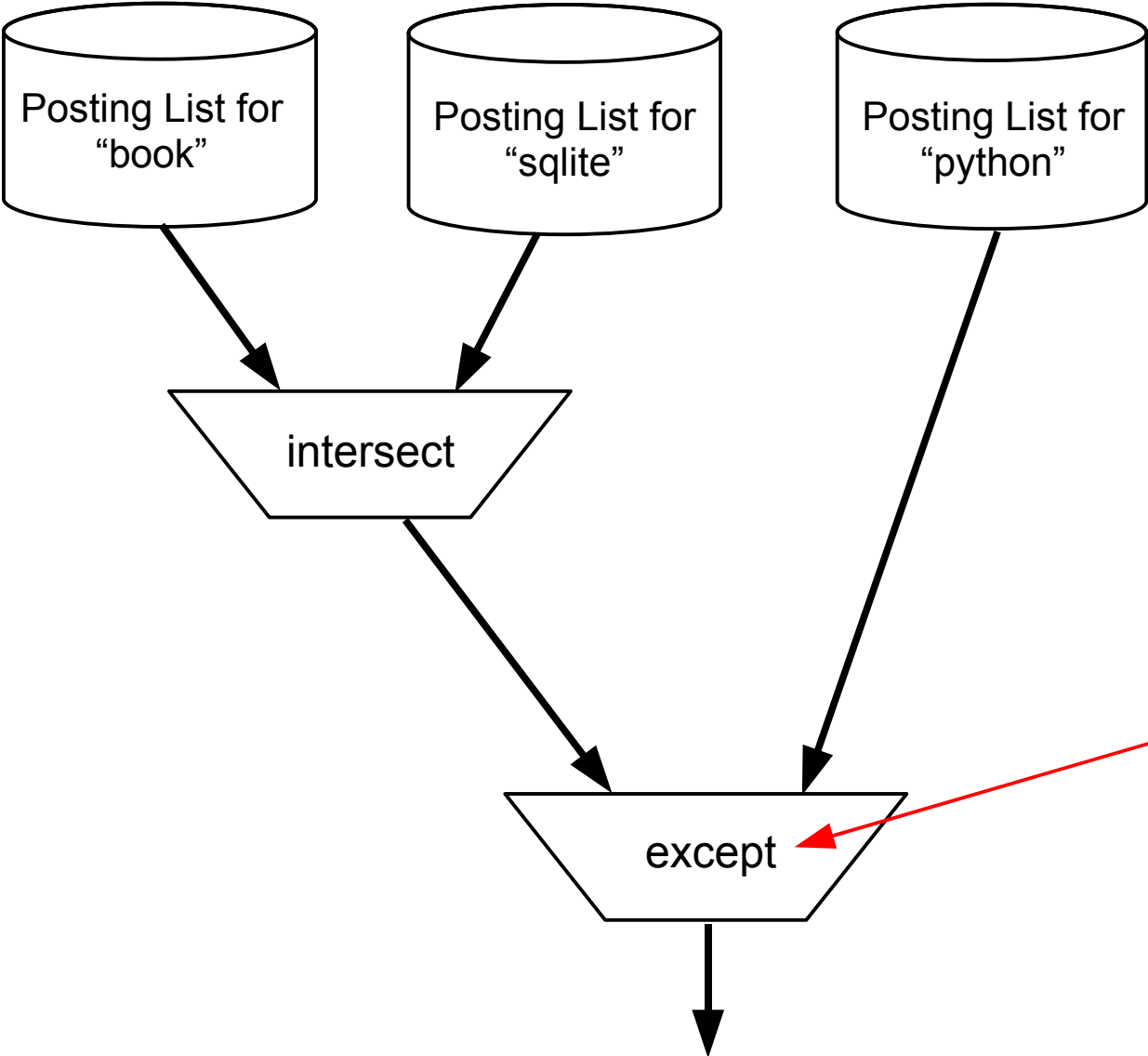
NOT Queries

- Use the complement of the posting list
 - Or better, use an “except” operator

book sqlite -python



book sqlite -python



Works just like the EXCEPT operator in SQL

Phrase Queries

(naïve method)

- Do an ordinary AND query, then examine every document in the result set in a second pass and eliminate those that lack the phrase

Consider this phrase query:

- “the and but”

2006¹ jun² 19³ new⁴ book⁵ about⁶ sqlite⁷ --- definit⁹ guid¹⁰
-- sqlite¹² - new¹⁴ book¹⁵ -- mike¹⁷ owen¹⁸ -- now²⁰ avail²¹ --
apress²³ --- book²⁵ cover²⁶ --- latest²⁸ sqlite²⁹ intern³⁰ -- well³²
-- --- nativ³⁵ - interfac³⁷ --- bind³⁹ --- php⁴¹ python⁴² perl⁴³
rubi⁴⁴ tcl⁴⁵ --- java⁴⁷ recommend⁴⁸



2006: doc52:15, doc871:81, ... , doc2351:1

jun: doc52:16, doc2351:2

19: doc88:96, doc92:6, doc93:15, doc1443:31, doc2351:3

new: doc11:7, ..., doc2337:51, doc2351:4, doc2351:14

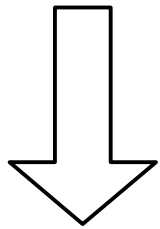
book: doc192:33, ..., doc2196:1, doc2351:5, doc2351:15, doc2351:25

■
■
■

recommend: doc1872:17, doc2351:48

“new book”

The query



Lookup posting lists
for each search term

doc11:7, ..., doc2337:51,
doc2351:4, doc2351:14



A:B

Phrase search combiner



If A==C
and D==B+1
then A:D



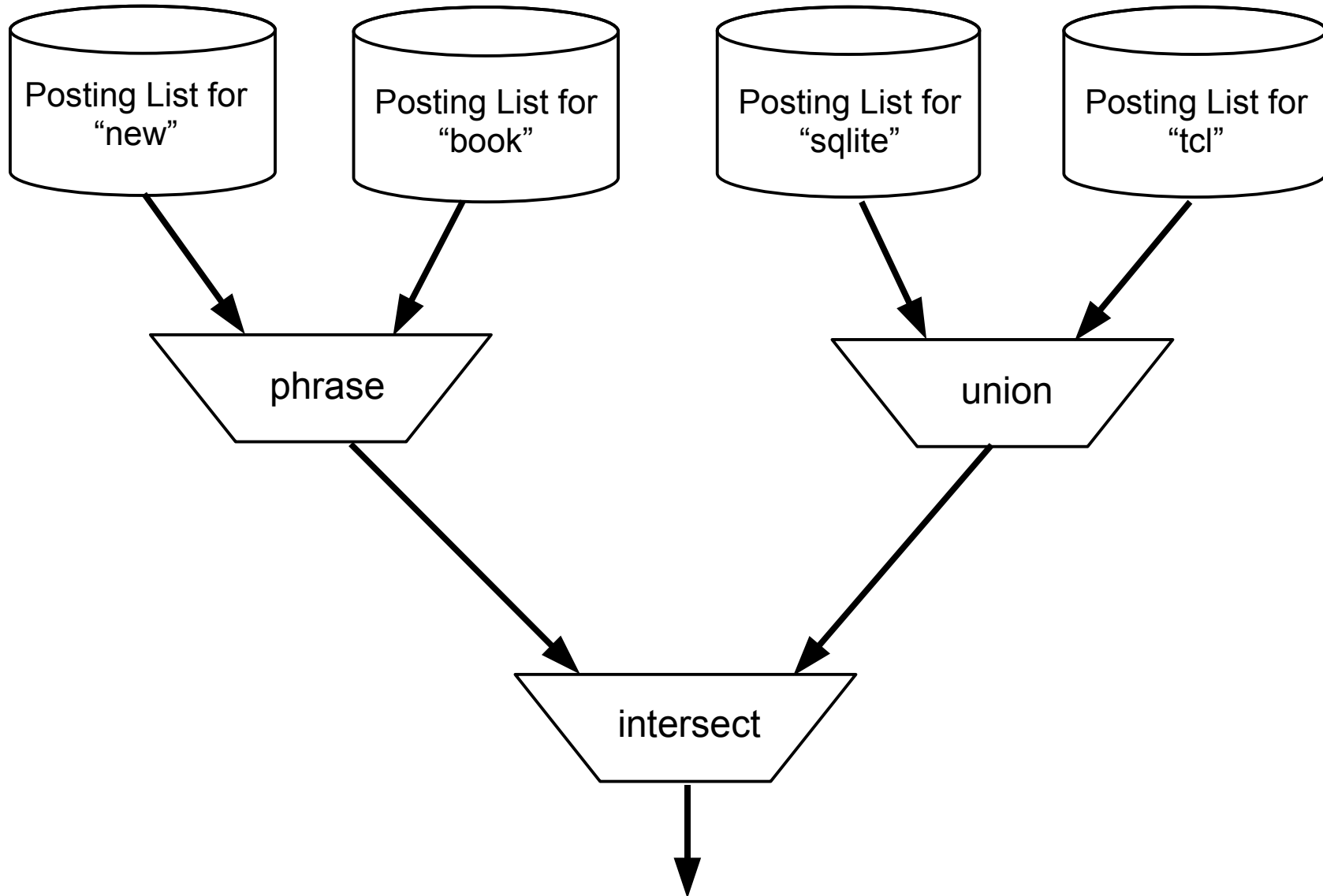
doc2351:5,
doc2351:15

doc192:33, ..., doc2196:1,
doc2351:5, doc2351:15,
doc2351:25



C:D

“new book” (sqlite OR tcl)



Basic Operations

1. Insert A New Document:

- Break the document up into words
- Append the document ID to the posting list for each word

2. Perform A Query:

- Lookup the posting list for each word in the query
- Combine posting lists

What's so hard about that?

Appending DocIDs to Posting Lists

- Average email contains about 200 unique words
- Compressed posting lists measure about 20-30% of the size of the original documents.
- A multi-gibibyte index will not fit in cache
- Each word lookup will require at least one revolution of the disk platter: 8ms
- Total time to insert one email: 1.6 seconds

To get good performance, you need to keep
your working set smaller than our cache

The working set for a full-text index can get
really big really fast

Keeping The Working Set Small

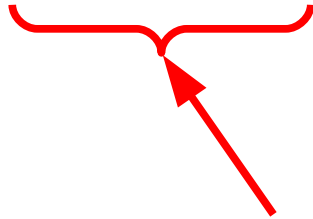
- Limit size of the lexicon
 - Stemming
 - Stop words
- Compress the posting lists aggressively
- Spill stationary parts of posting lists into a separate table that is not cached

Full Text Search

in

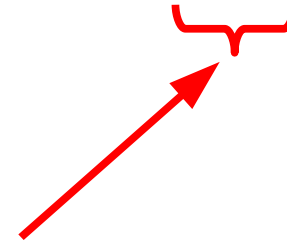
The SQLite logo is displayed in a light green, handwritten-style font. The word "SQLite" is written in a bold, rounded script. To the right of the text is a stylized graphic of a quill pen, also in the same green color, with several curved lines representing the feathers or the tail of the quill.

CREATE VIRTUAL TABLE email USING fts1(content);



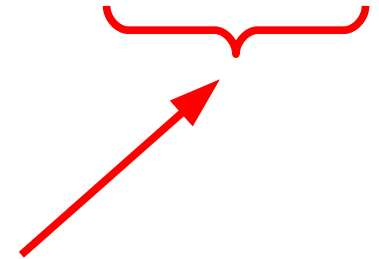
Looks and acts like a table
but is not really a table.

CREATE VIRTUAL TABLE email USING fts1(content);



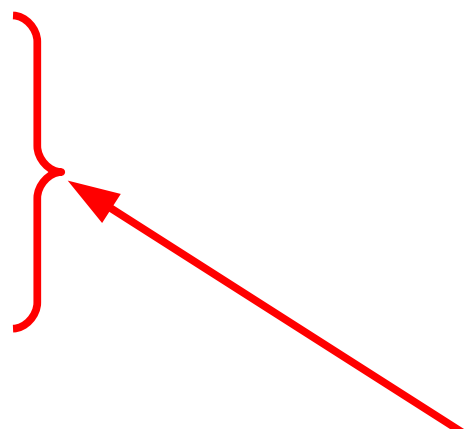
Implemented using the “fts1”
module.

CREATE VIRTUAL TABLE email USING fts1(content);



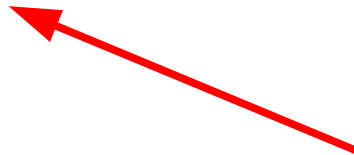
Stores and indexes a single column named "content".


```
CREATE VIRTUAL TABLE email USING fts1(  
  "from" TEXT,  
  "to" TEXT,  
  subject TEXT,  
  body TEXT  
);
```



Able to index multiple columns.

```
CREATE VIRTUAL TABLE email USING fts1(  
  "from" TEXT,  
  "to" TEXT,  
  subject TEXT,  
  body TEXT,  
  tokenize porter  
);
```



Use the Porter stemmer.

```
db eval {  
  INSERT INTO email(rowid,[from],[to],subject,body)  
  VALUES($msgid,$from,$to,$subj,$body)  
}
```



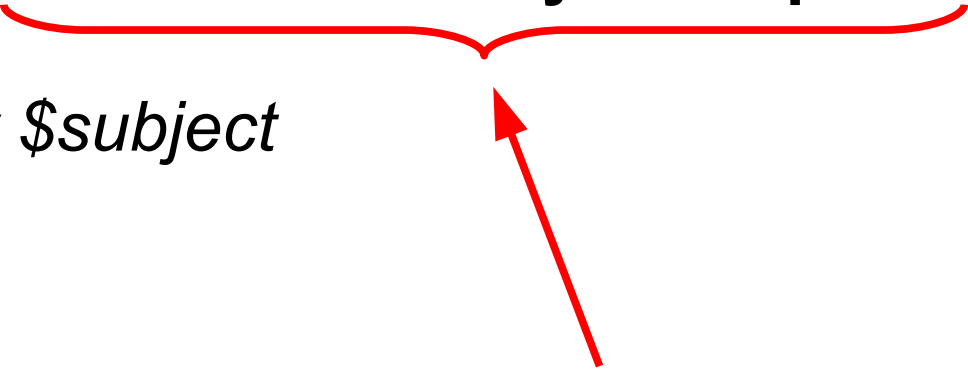
Insert just like a regular table

```
db eval {  
  DELETE FROM email WHERE rowid=$msgid  
}
```



Delete works the same, too

```
db eval {  
  SELECT rowid, subject FROM email  
  WHERE email MATCH 'wyrick sqlite'  
} {  
  # Display $subject  
}
```



Full-text search query using the
MATCH clause.

```
db eval {  
  SELECT rowid, subject FROM email  
  WHERE email MATCH 'wyrick sqlite'  
} {  
  ...  
}
```

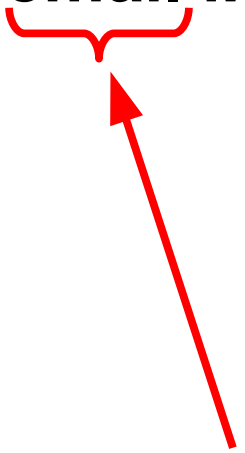
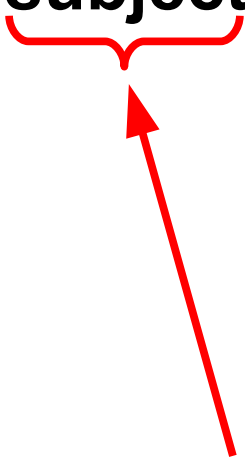


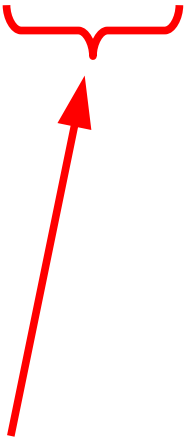
Table name as left operand means
match against any column of the table

```
db eval {  
  SELECT rowid, subject FROM email  
  WHERE subject MATCH 'wyrick sqlite'  
} {  
  ...  
}
```



Use a particular column name to limit
the search to that one column

```
db eval {  
  SELECT rowid, subject FROM email  
  WHERE email MATCH 'from:wyrick sqlite'  
} {  
  ...  
}
```

A red bracket is positioned above the search term 'from:wyrick sqlite' in the SQL query. A red arrow points upwards from below the text to the center of the bracket.

Qualifiers limit an individual search term
to a particular column

Built-in snippet generator

```
db eval {  
    SELECT rowid, snippet(email) FROM email  
    WHERE email MATCH 'from:wyrick sqlite'  
} {  
    ...  
}
```

SQLite Download Page

http://www.sqlite.org/download.html

News ▾ OSCON-talk Projects ▾

[sqlite-source-3.3.8.zip](#) (586.44 KiB) This ZIP archive contains pure C source code for the SQLite library. Unlike the tarballs below, all of the preprocessing and automatic code generation has already been done on these C source code, so they can be processed directly with any ordinary C compiler. This file is provided as a service to MS-Windows users who lack the build support infrastructure of Unix.

[sqlite-3.3.8-tea.tar.gz](#) (672.73 KiB) A tarball of proprocessed source code together with a Tcl Extension Architecture (TEA) compatible configure script and makefile.

Cross-Platform Binaries

[sqlite-3.3.8.kit](#) (769.50 KiB) A starkit containing precompiled SQLite binaries and Tcl bindings for Linux-x86, Windows, and Mac OS-X.

FT S1 comes standard with TCL bindings

Direct Access To The Sources Via Anonymous CVS

All SQLite source code is maintained in a CVS repository that is available for read-only

Potential Uses

- Search for private websites
- Email Clients
- On-line documentation search
- Searchable history in web browsers
- Chatroom archive search
- Search version diffs in a CM system
- Text editors and IDEs

Pervasive Full-Text Search

Pervasive Full-Text Search

Made easy using Tcl and SQLite!

The Land Of Tcl/Tk

The Territory of Pervasive Full-Text Search

The Land Of Tcl/Tk



The Territory of Pervasive Full-Text Search