

[トップページ](#) -> [研究工具箱と教材](#) -> [リレーショナルデータベース入門\(実践で学ぶ\)](#) -> [データベースの物理構造 \(Physical Structure of Relational Database\)](#)

[\[サイトマップへ\]](#) [\[全文検索へ\]](#) [\[統計情報へ\]](#)

データベースの物理構造 (Physical Structure of Relational Database)

URL: <http://www.db.is.kyushu-u.ac.jp/rinkou/addb/5.html>

```
CREATE TABLE reports (  
  id INTEGER PRIMARY KEY,  
  docid INTEGER,  
  path TEXT,  
  val TEXT,  
  created_at DATETIME NOT NULL );  
INSERT INTO reports (1, 1, '/root/title', 'report A', '2009-12-11 11:20:36');  
INSERT INTO reports (2, 1, '/root/author', 'kaneko', '2009-12-11 11:20:36');  
INSERT INTO reports (3, 1, '/root/date', '2009/11/29', '2009-12-11 11:20:36');
```

■ The attribute 'id' is the **PRIMARY KEY**
The 'id' is **key** that can be used to identify each row.

Table 'reports'

id	docid	path	val	created_at
1	1	/root/title	'report A'	2009-12-11 11:20:36
2	1	/root/author	'kaneko'	2009-12-11 11:20:36
3	1	/root/date	'2009/11/29'	2009-12-11 11:20:36

PRIMARY
KEY

Fig. Logical Structure of the Table

Table 'reports'

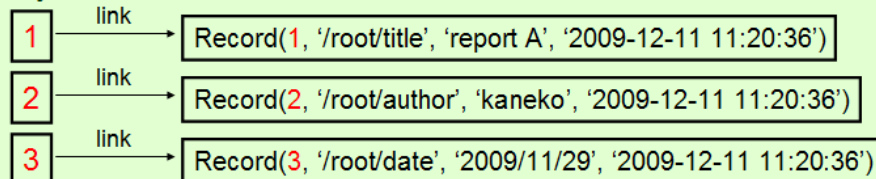
id	docid	path	val	created_at
1	1	/root/title	'report A'	2009-12-11 11:20:36
2	1	/root/author	'kaneko'	2009-12-11 11:20:36
3	1	/root/date	'2009/11/29'	2009-12-11 11:20:36

PRIMARY
KEY

Logical Structure

Keys

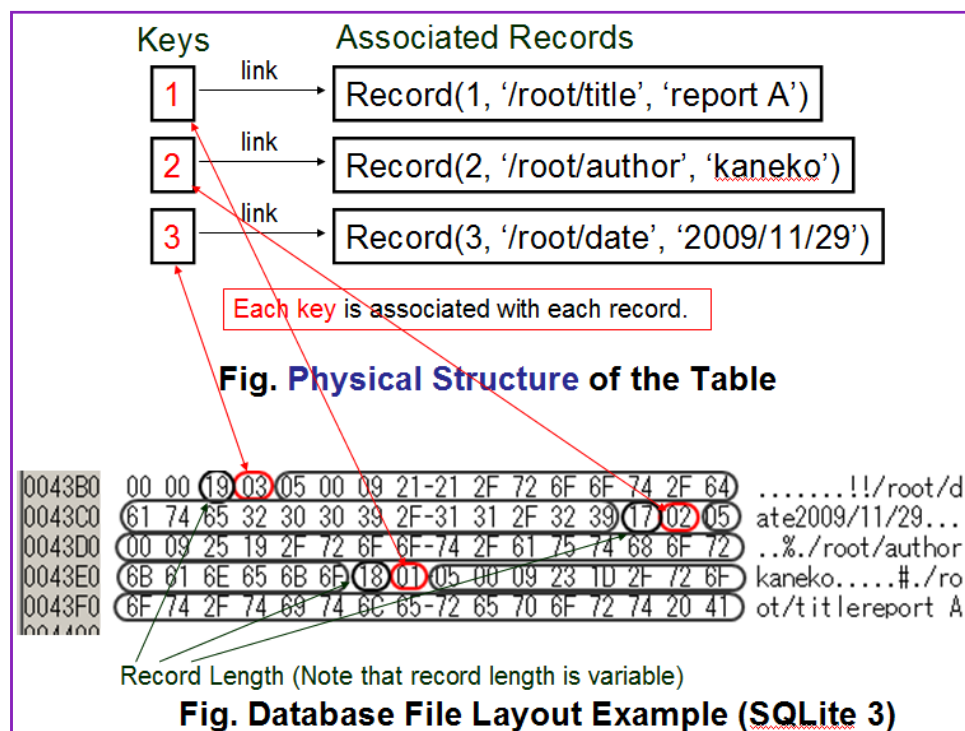
Associated Records



Each key is associated with each record.

Physical Structure (Conceptual View)

Fig. Logical Structure and Physical Structure



演習

演習で行うこと

- ・ SQLite 3 Analyzer のインストール (Install SQLite 3 Analyzer)
- ・ (Windows の場合) バイナリエディタのインストール
- ・ SQLite データベースファイルの物理構造 (Physical Structure of SQLite Database File)
- ・ 二次索引 (secondary index)
- ・ SQL 問い合わせ計画 (SQL query plan)

SQLite 3 Analyzer のインストール (Install SQLite 3 Analyzer)

■ Ubuntu の場合

<http://www.sqlite.org/> から Linux 用のファイル入手.

Precompiled Binaries for Linux	
sqlite3-3.7.3.bin.gz (231.57 KiB)	A command-line program for accessing and modifyi documentation for additional information.
tclsqlite3-3.7.3.so.gz (242.74 KiB)	Bindings for Tcl/Tk . You can import this shared libr database access from Tcl/Tk. See the documentati
sqlite3-3.7.3.so.gz (231.93 KiB)	A precompiled shared-library for Linux without the
sqlite3_analyzer-3.7.2-linux.zip (723.66 KiB)	An analysis program for database files compatible beyond.

■ Windows の場合

1. [sqlite3_analyzer-3.7.1.zip](#) の入手

<http://www.sqlite.org/>

The screenshot shows the SQLite Home Page. The 'Download' button in the navigation bar is highlighted with a red box. The page includes a welcome message, current status (Version 3.7.3), common links, and a section for precompiled binaries for Windows.

Welcome.

SQLite is a software library that implements a [self-contained](#), [serverless](#), [zero-configuration](#), [transactional](#) SQL database engine. SQLite is the [most widely deployed](#) SQL database engine in the world. The source code for SQLite is in the [public domain](#).

Sponsors

Ongoing development and maintenance of SQLite is sponsored in part by [SQLite Consortium](#) members, including:

 **Adobe**

Adobe revolutionizes how the world engages with ideas and information - anywhere, anytime, and through any medium.

Current Status

- [Version 3.7.3](#) of SQLite is recommended for all new development. Upgrading from version 3.7.2 is optional. Upgrading from all other SQLite versions is recommended.

Common Links

- [Features](#)
- [Frequently Asked Questions](#)
- [Well-known Users](#)
- [Getting Started](#)
- [SQL Syntax](#)
 - [Pragmas](#)
 - [SQL functions](#)
 - [Date & time functions](#)
 - [Aggregate functions](#)
- [C/C++ Interface Spec](#)
 - [Introduction](#)
 - [List of C-language APIs](#)
- [The TCL Interface Spec](#)
- [Development Timeline](#)
- [Report a Bug](#)
- [Wiki](#)

(3390.65 KiB)

Precompiled Binaries For Windows
[sqlite-3.7.3.zip](#)
(268.98 KiB)

A command-line program for accessing and modifying database files. See [the documentation](#) for additional information.

[tclsqlite-3.7.3.zip](#)
(339.67 KiB)

Bindings for [Tcl/Tk](#). You can import this shared library to get database access from Tcl/Tk. See [the documentation](#).

[sqldll-3.7.3.zip](#)
(267.18 KiB)

This is a DLL of the SQLite library without the TCL MSVCRT.DLL.

[sqlite3_analyzer-3.7.1-win32.zip](#)
(557.29 KiB)

An analysis program for database files compatible with SQLite 3.7.1 and beyond.

2. **sqlite3_analyzer-3.7.1.zip の解凍**

解凍すると sqlite3_analyzer.exe ができる。

3. **sqlite3_analyzer.exe を C:/SQLite にコピー**

名前	サイズ	種類	更新日時
sqlite3.exe	513 KB	アプリケーション	2009/12/07 12:05
mydb	23 KB	ファイル	2009/12/11 20:20
sqlite3_analyzer.exe	1,119 KB	アプリケーション	2008/08/05 17:33

(Windows の場合) バイナリエディタのインストール

1. バイナリエディタ BZ のファイルの入手
2. ファイルの確認



3. ファイルの解凍 (Extract the file)

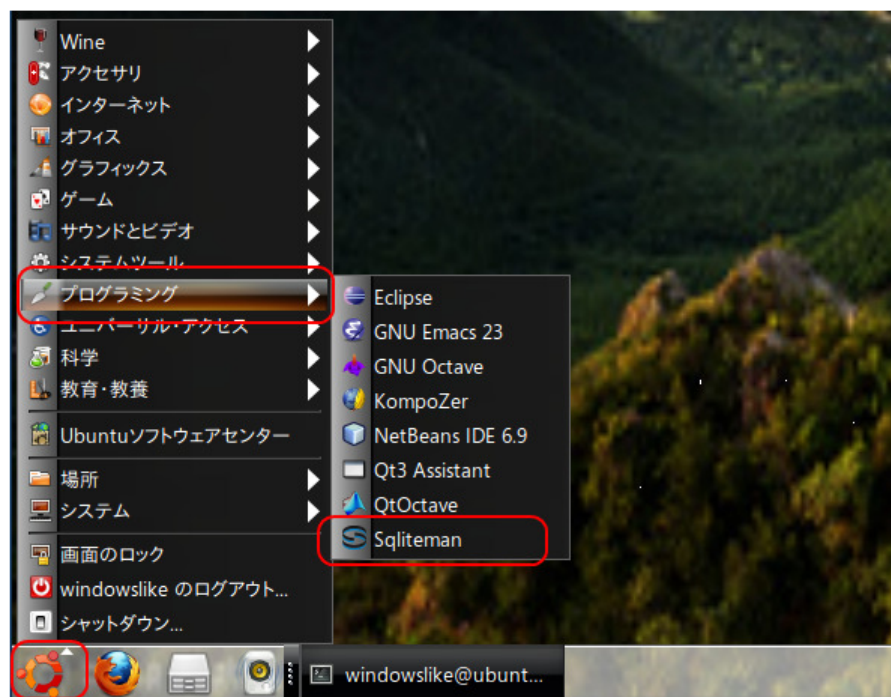


Sqliteman の起動 (Start Sqliteman)

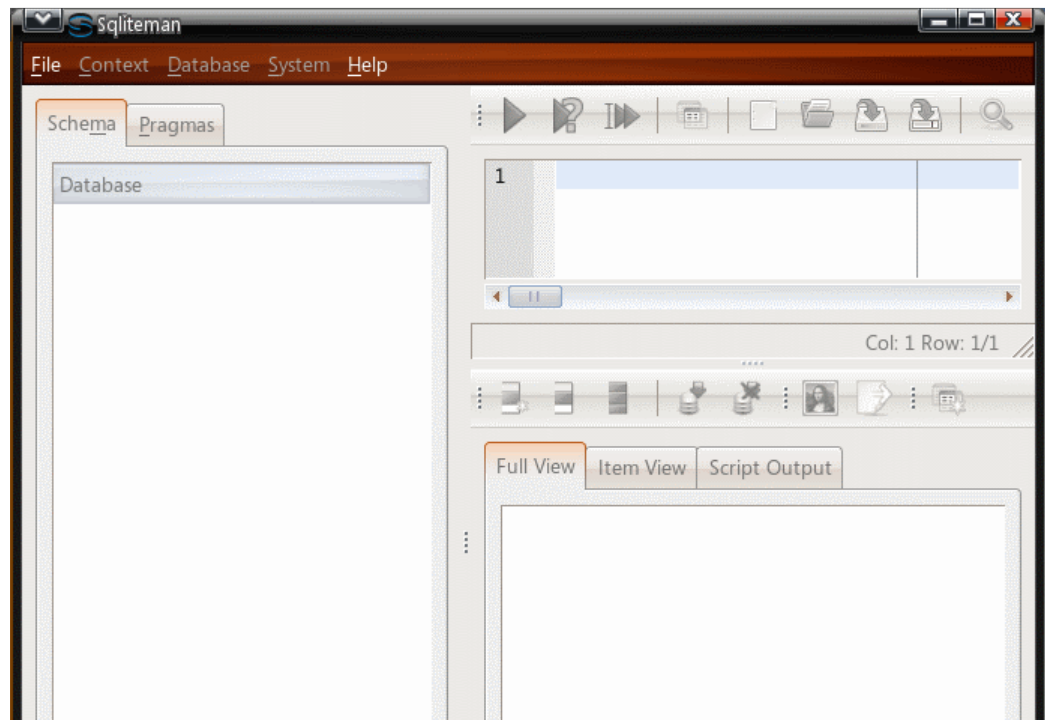
1. **SQLiteman の起動** (Start SQLiteman)

■ Ubuntu での SQLiteman の起動例

「プログラミング」→「Sqliteman」と操作する。



SQLiteman の新しいウィンドウが開く. (A New window appears)

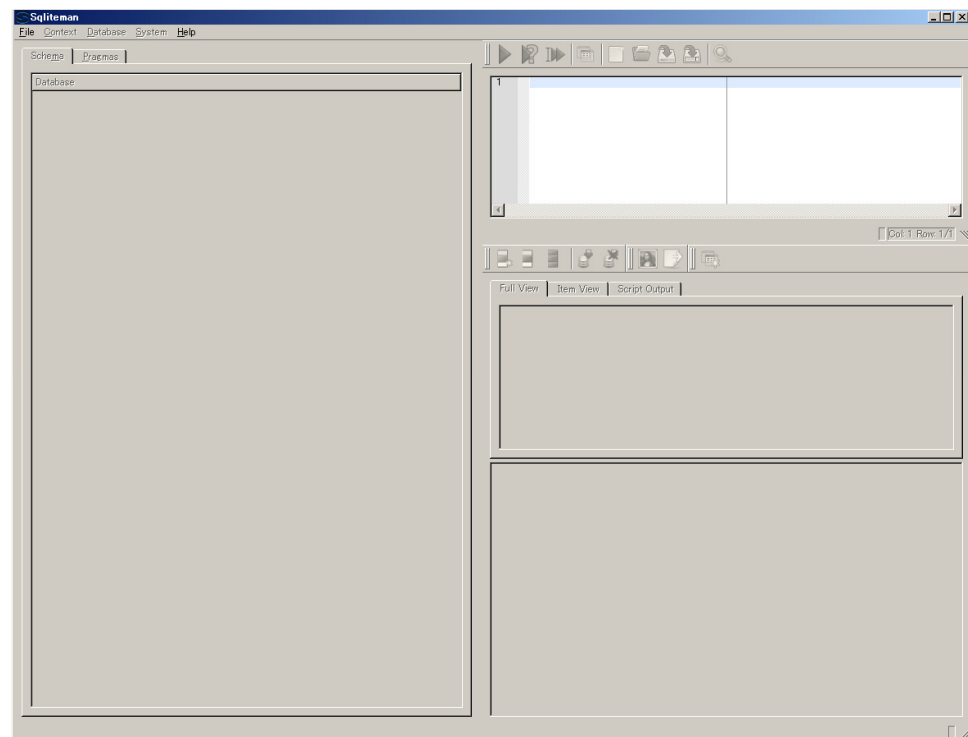


■ Windows での SQLiteMan の起動例

「SQLiteMan」のアイコンをダブルクリック (double click "SQLiteMan.exe")



SQLiteMan の新しいウィンドウが開く. (A New window appears)

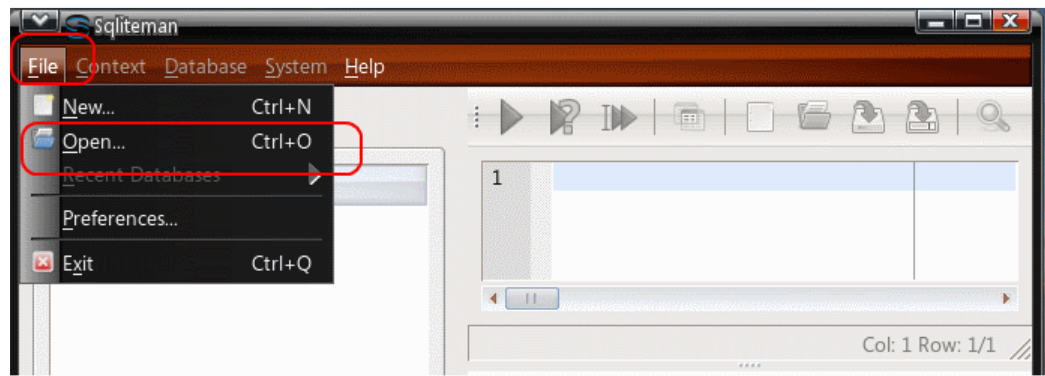


SQLiteMan で既存のデータベースを開く (Open an existing database using SQLiteMan)

すでに作成済みのデータベースを, 下記の手順で開くことができる.

以下の手順で, 既存のデータベースファイルを開く. (Open an existing database file)

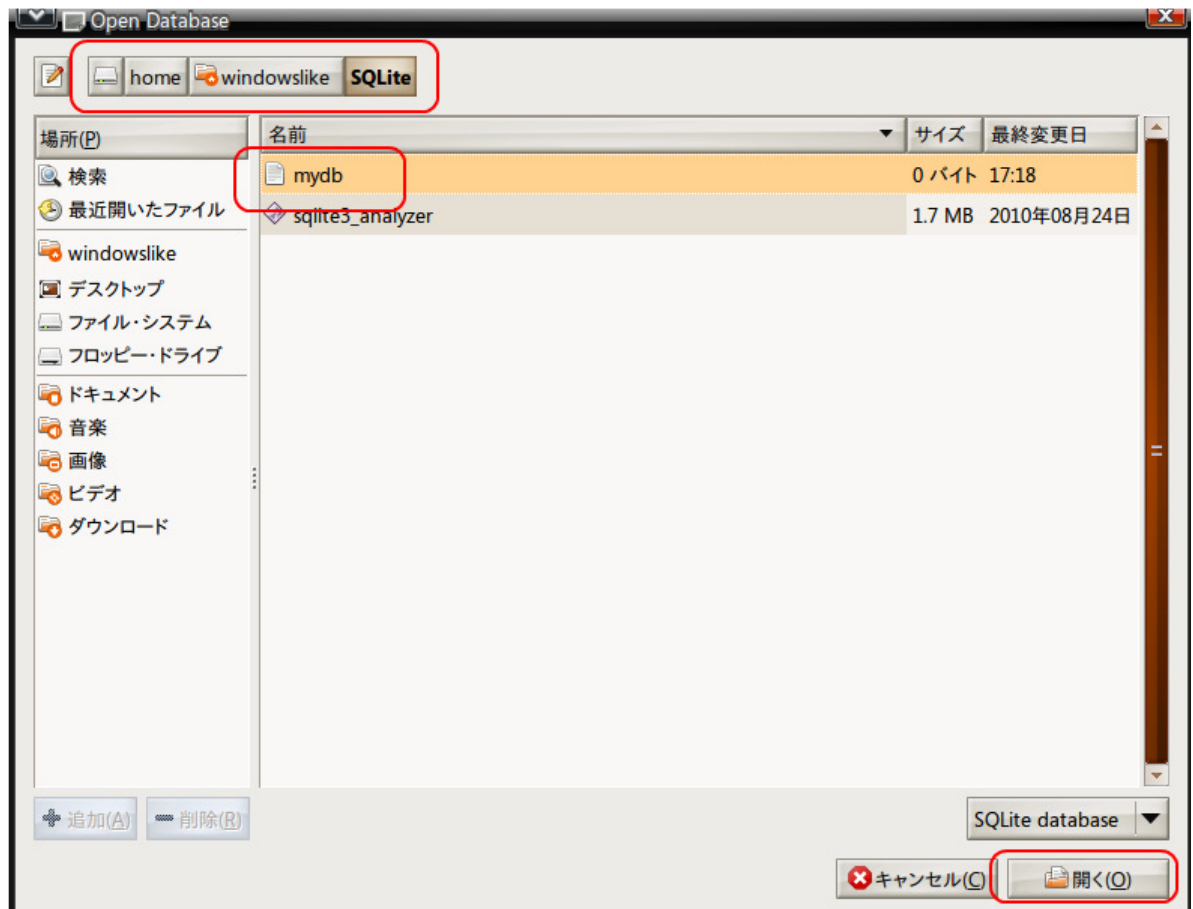
1. 「File」→「Open」



2. データベースファイルを開く (Open Database File)

■ Ubuntu での実行例 (「SQLite/mydb」を開く場合)

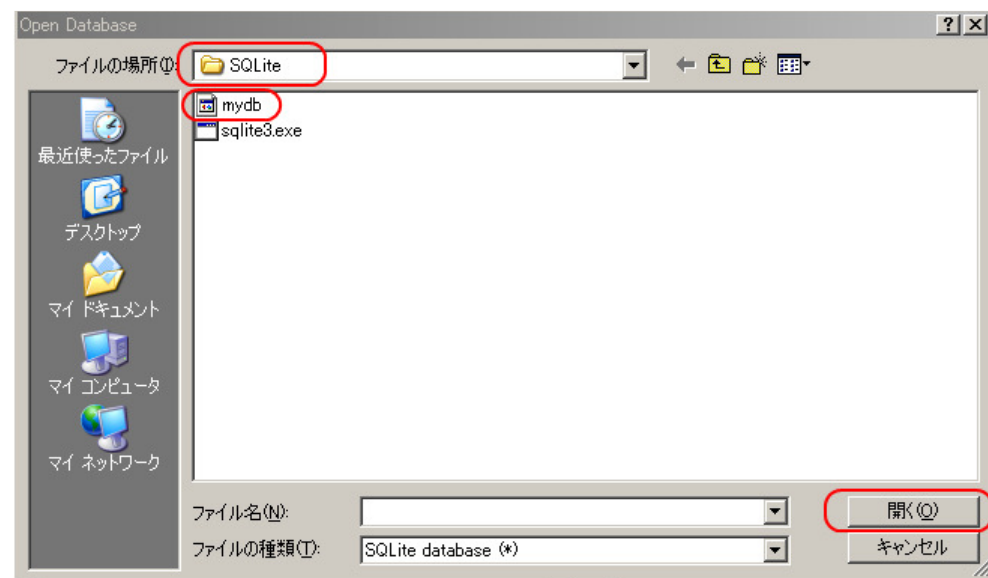
データベースファイル **SQLite/mydb** を選び、「開く」をクリック (Click '開く' after choosing the database file "SQLite/mydb")



■ Windows での実行例 (「C:\SQLite\mydb」を開く場合)

データベースファイル **C:\SQLite\mydb** を選び、「開く」をクリック (Click '開く' after choosing the database file "C:\SQLite\mydb")

要するに、/home/<ユーザ名>/SQLite の下の mydb を選ぶ。



SQL を用いたテーブル定義と一貫性制約の記述 (Table definition and integrity constraint specification using SQL)

SQL を用いて、**products** テーブルを定義し、一貫性制約を記述する。 (Define 'products' table and specify integrity constraints of the table using SQL)

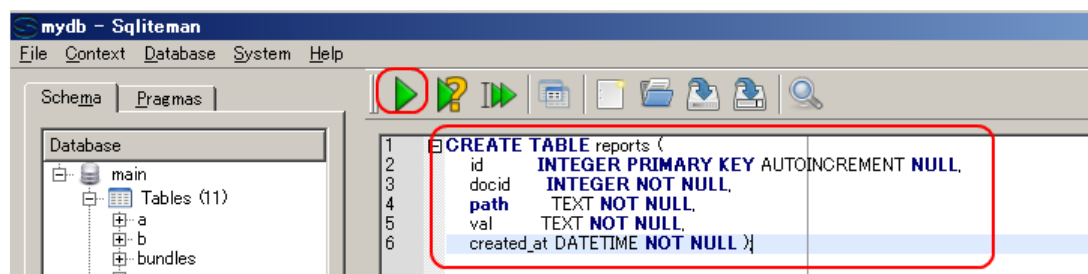
リレーショナル・スキーマ (relational schema): **reports(id, docid, path, val, created_at)**

1. **reports** テーブルの定義 (Define a table)

次の SQL を入力し、「Run SQL」のアイコンをクリック (Write the following SQL, and click "Run SQL" icon).

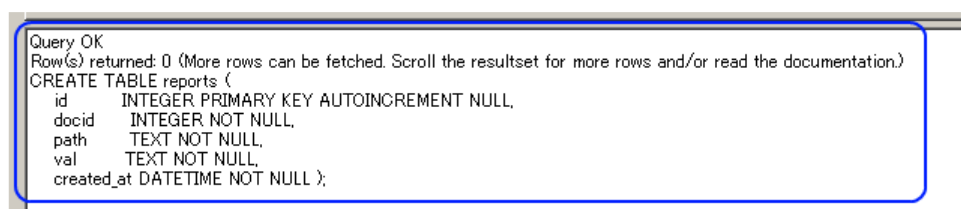
```
CREATE TABLE reports (
  id          INTEGER PRIMARY KEY AUTOINCREMENT NULL,
  docid       INTEGER NOT NULL,
  path        TEXT NOT NULL,
  val         TEXT NOT NULL,
  created_at  DATETIME NOT NULL );
```

※「SQL Editor」のウィンドウには、SQL プログラムを書くことができる。 In the '**SQL string**' window, you can write down **SQL program(s)**.



2. **コンソールの確認** (Inspect console)

エラーメッセージが出ていないことを確認



SQL を用いたテーブルへの行の挿入 (Insert rows into a table using SQL)

次のような **reports** テーブルを作る。 (Construct table 'reports')

id	docid	path	val
1	1	/root/title	report A
2	1	/root/author	kaneko
3	1	/root/date	2009/11/29

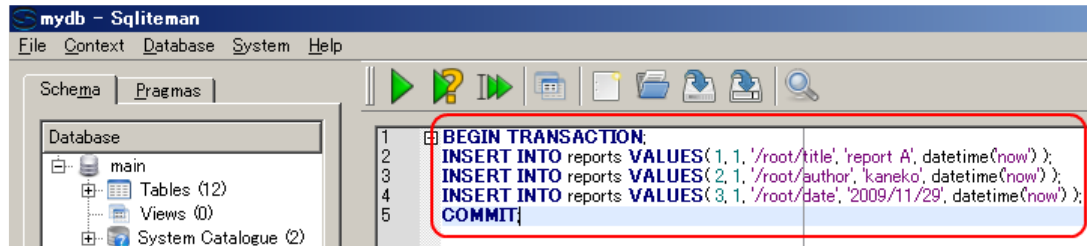
以下の手順で、SQL を用いて **reports** テーブルへの行の挿入を行う (Insert rows into table 'reports' using SQL)

1. **SQL プログラムの記述**

```

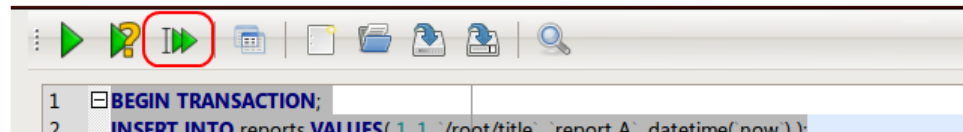
BEGIN TRANSACTION;
INSERT INTO reports VALUES( 1, 1, '/root/title', 'report A', datetime('now') );
INSERT INTO reports VALUES( 2, 1, '/root/author', 'kaneko', datetime('now') );
INSERT INTO reports VALUES( 3, 1, '/root/date', '2009/11/29', datetime('now') );
COMMIT;

```



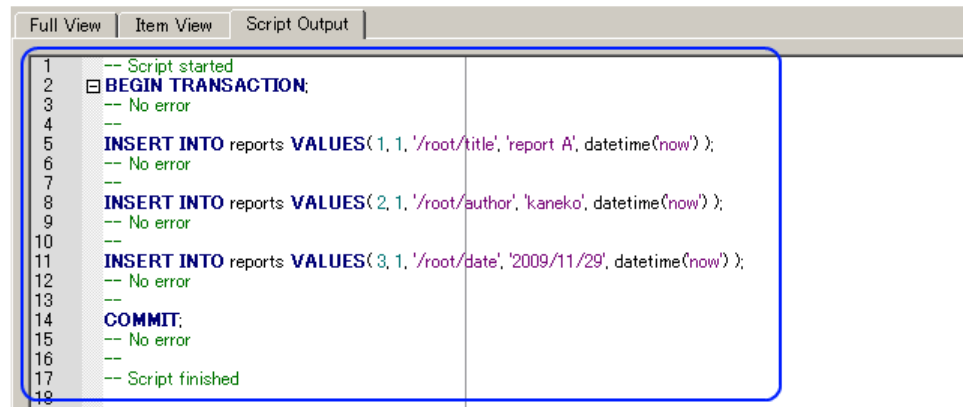
2. 複数の SQL 文の一括実行 (Run multiple SQL statements)

複数の SQL 文を一括実行したいので、カーソルを先頭行に移動した後に、「Run multiple SQL statements ...」のボタンをクリックする。「Move the cursor to the top statement. Click "Run multiple SQL statements from current cursor position in one batch" icon」



3. 「Script Output」ウインドウの確認 (Inspect "Script Output" window)

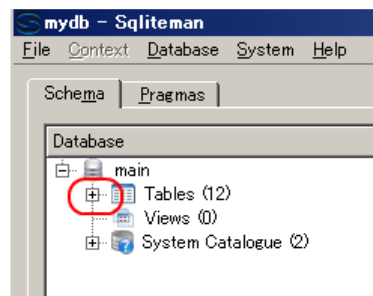
エラーメッセージが出ていないことを確認



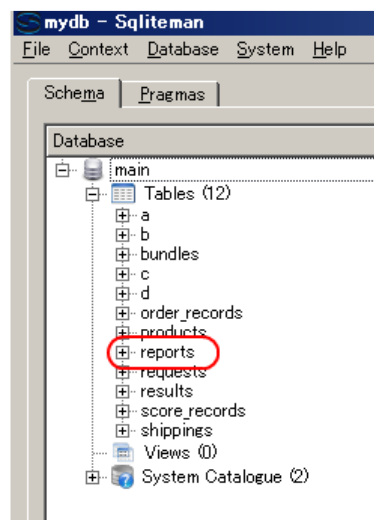
Sqliteman を用いたデータのブラウズ (Browse Data using Sqliteman)

reports テーブル

まず、オブジェクト・ブラウザ (Object Browser) の中の「Tables」を展開 (Click 'Tables')



次に、テーブル reports を選ぶ (Select table 'reports')

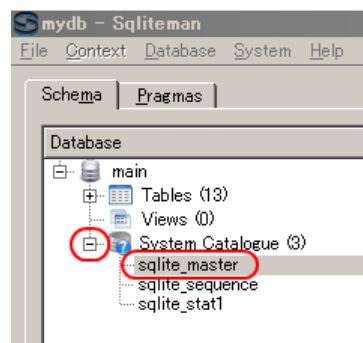


テーブル **reports**が表示される (table 'reports' appears)

	id	docid	path	val	created_at
1	1	1	/root/title	report A	2009-12-11 11:20:36
2	2	1	/root/author	kaneko	2009-12-11 11:20:36
3	3	1	/root/date	2009/11/29	2009-12-11 11:20:36

※ もし、データに間違いがあれば、このウィンドウで修正できる (If you find any mistakes, you can modify the data using this window).

- ・「**System Catalogue**」を展開し、「**sqlite_master**」をクリックすると、データベース・スキーマ (database schema) が表示されるので確認する (Database schema appears by clicking 'sqlite_master')



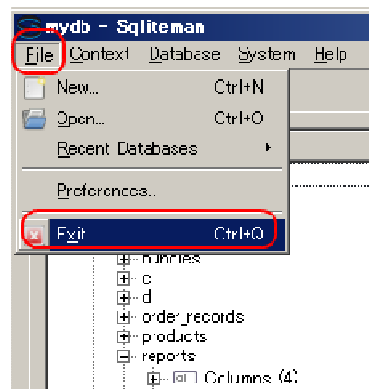
- ・データベーススキーマの中の**reports** テーブルの行を確認する。 (Examine the table 'reports' in the database schema information)

	type	name	tbl_name	rootpage	sql
1	table	sqlite_sequence	sqlite_sequence	4	CREATE TABLE sqlite_sequence(name,seq)
2	table	reports	reports	2	CREATE TABLE reports (id INTEGER PRIMARY KEY AL docid INTEGER NOT NULL, path TEXT NOT NULL, val TEXT NOT NULL, createdat DATETIME NOT NULL)

reports テーブルのルート・ページ番号が分かる。この場合は「2」になっているが、違う値になっていても問題はない。 (The root page number of table 'reports'. In this figure, the number is '2').

ルート・ページ番号は、データベース管理システムが自動的に決める番号である。

- ・ **Sqliteman** を終了 (Exit the Sqliteman)
このときデータベースが**保存**される (Database is saved automatically)



SQLite データベースファイルの物理構造 (Physical Structure of SQLite Database File)

データベースファイルは、レコードを単位とした物理構造になっている。

1-9 bytes		
Child page number	Record Size	Database Record
4 bytes (not present for leaf pages)		<i>record-size</i> bytes, where <i>record-size</i> is the value stored in the previous field.

Figure 5 – Small Record Index B-Tree Cell

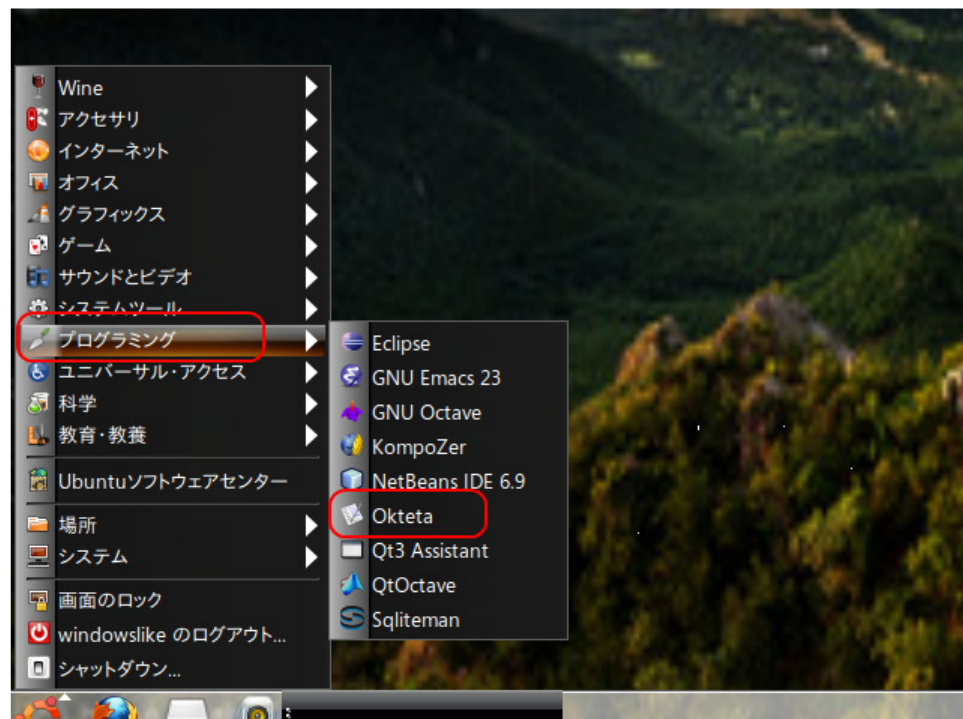
This figure is from the 'SQLite Database File Format' Web page. <http://www.sqlite.org/fileformat.html>

1. **Sqliteman** を終了しておくこと (Exit the Sqliteman)
2. **バイナリエディタの起動** (Execute the binary editor)

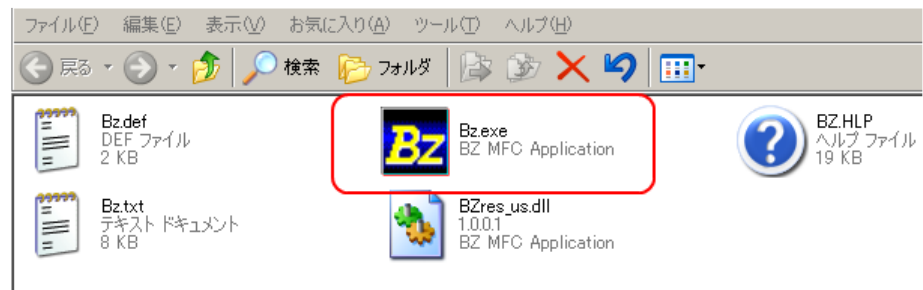
■ Ubuntu の場合 (Okteta を開く)

「プログラミング」→「Okteta」のように操作する

※ 端末を開いて、「okteta」のように実行しても良い。okteta が無いときは手動でインストールする。



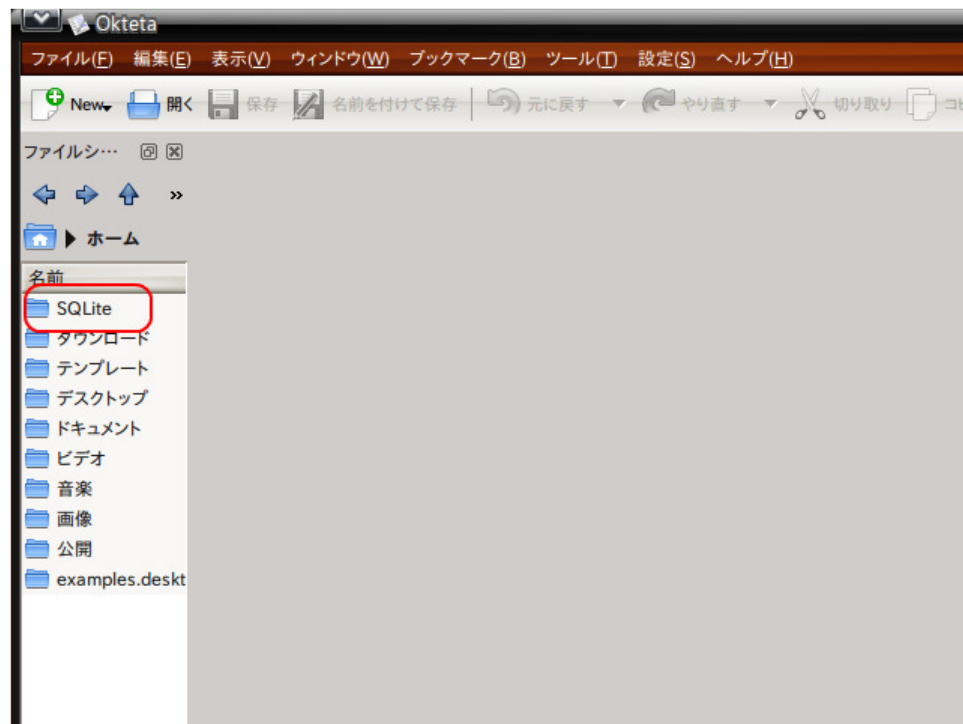
■ Windows の場合 (Bz を開く)



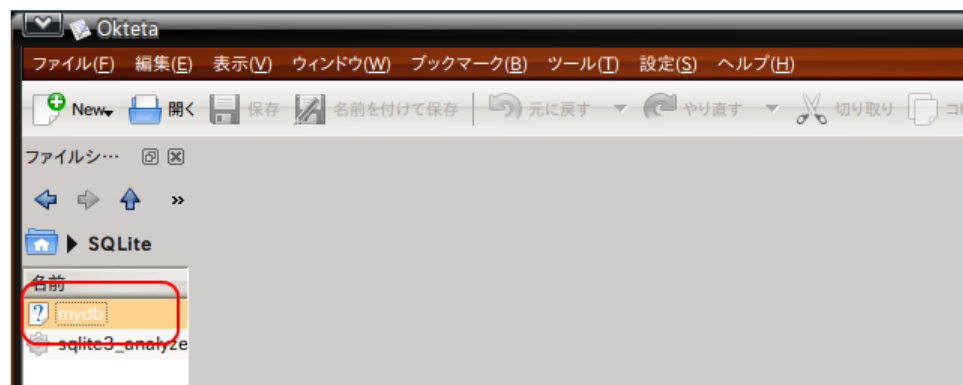
3. データベースファイルを開く (Open the database file)

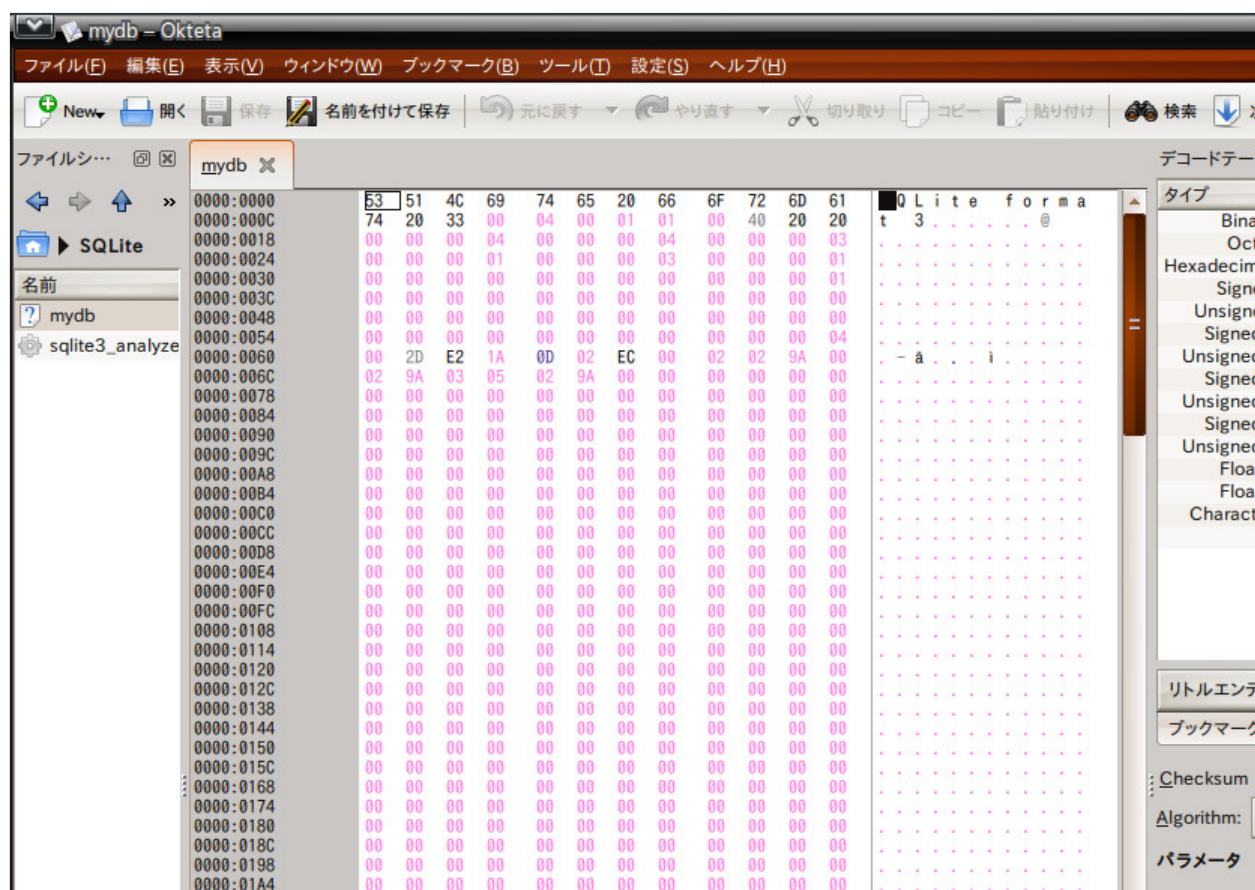
■ Ubuntu の場合 (Okteta を使う場合)

ディレクトリを選ぶ。ここでは、「SQLite」を選んでいる(「SQLite」をダブルクリック)。

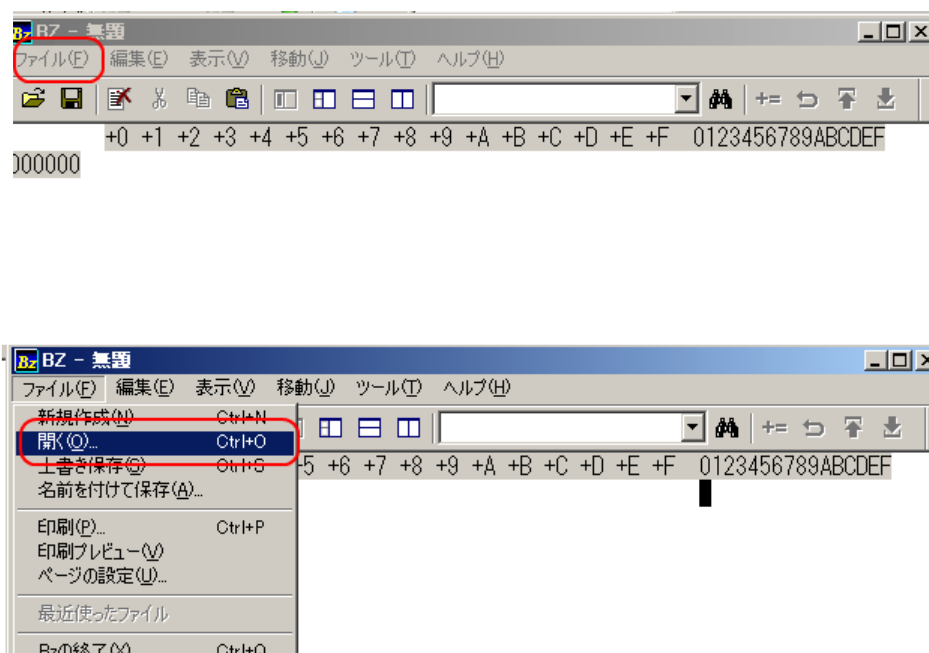


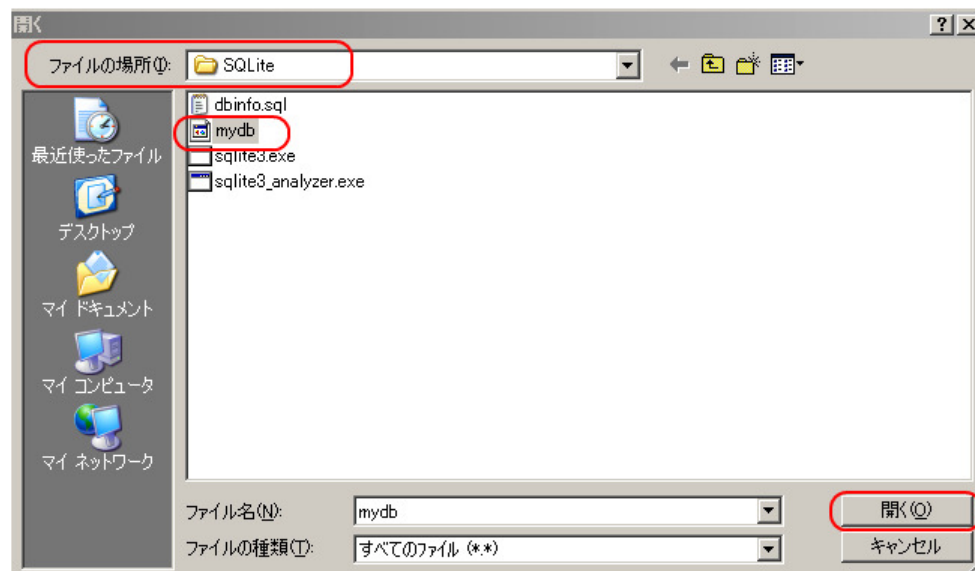
ファイルを選ぶ。ここでは、「mydb」を選んでいる(「mydb」をダブルクリック)。





■ Windows の場合 (Bz)





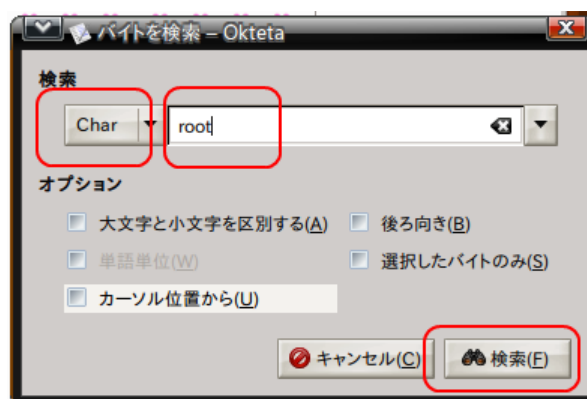
4. データベース・ファイルの中から **reports** テーブルのレコードを探す (Find the table 'records' in a database file manually)

■ Ubuntu の場合 (Okteta を使う場合)

「検索」をクリック



「Char」を選び、「root」を指定して、「検索」をクリック



5. データベースの中身の確認

データベース・ファイルのデータページの中には、レコードが並んでいることが確認できる。データページの中には未使用部分がある。

There a sequence of **records** in data pages in database file.

005B00	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B10	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B20	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B30	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B40	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B50	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B60	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B70	00 00 00 2E 03 06 00 01-21 21 33 01 2F 72 6F 6F	!!3./roo
005B80	74 2F 64 61 74 65 32 30-30 39 2F 31 31 2F 32 39	t/date2009/11/29
005B90	32 30 30 39 2D 31 32 2D-31 31 20 31 31 3A 32 30	2009-12-11 11:20
005BA0	3A 33 36 2C 02 06 00 01-25 19 33 01 2F 72 6F 6F	:36,...%.3./roo
005BB0	74 2F 61 75 74 68 6F 72-6B 61 6E 65 6B 6F 32 30	t/authorkaneko20
005BC0	30 39 2D 31 32 2D 31 31-20 31 31 3A 32 30 3A 33	09-12-11 11:20:3
005BD0	36 2D 01 06 00 01 23 1D-33 01 2F 72 6F 6F 74 2F	6-...#.3./root/
005BE0	74 69 74 6C 65 72 65 70-6F 72 74 20 41 32 30 30	titlereport A200
005BF0	39 2D 31 32 2D 31 31 20-31 31 3A 32 30 3A 33 36	9-12-11 11:20:36
005C00	0D 00 00 00 04 03 44 00-03 D4 03 A8 03 70 03 44	D... ..p.D
005C10	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005C20	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005C30	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005C40	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	

※ データベースの構造

- SQLite 3 では、データベースヘッダがある
- SQLite 3 では、データページのサイズは 1024バイト (16進数で 400)

6. path フィールドの値の確認 ('path' field)

```

005B00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B70 00 00 00 2F 03 06 00 01-21 21 33 01 2F 72 6F 6F .....!!3./roo
005B80 74 2F 64 61 74 65 32 30-30 39 2F 31 31 2F 32 39 t/date2009/11/29
005B90 32 30 30 39 2D 31 32 2D-31 31 20 31 31 3A 32 30 2009-12-11 11:20
005BA0 3A 33 36 2C 02 06 00 01-25 19 33 01 2F 72 6F 6F :36,...%.3./roo
005BB0 74 2F 61 75 74 68 6F 72 6B 61 6E 65 6B 6F 32 30 t/author kaneko20
005BC0 30 39 2D 31 32 2D 31 31-20 31 31 3A 32 30 3A 33 09-12-11 11:20:3
005BD0 36 2D 01 06 00 01 23 1D-33 01 2F 72 6F 6F 74 2F 6-...#.3./root/
005BE0 74 69 74 6C 65 72 65 70-6F 72 74 20 41 32 30 30 title report A200
005BF0 39 2D 31 32 2D 31 31 20-31 31 3A 32 30 3A 33 36 9-12-11 11:20:36
005C00 0D 00 00 00 04 03 44 00-03 D4 03 A8 03 70 03 44 .....D... ..p.D
005C10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

7. val フィールドの値の確認 ('val' field)

```

005B00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B70 00 00 00 2F 03 06 00 01-21 21 33 01 2F 72 6F 6F .....!!3./roo
005B80 74 2F 64 61 74 65 32 30-30 39 2F 31 31 2F 32 39 t/date2009/11/29
005B90 32 30 30 39 2D 31 32 2D-31 31 20 31 31 3A 32 30 2009-12-11 11:20
005BA0 3A 33 36 2C 02 06 00 01-25 19 33 01 2F 72 6F 6F :36,...%.3./roo
005BB0 74 2F 61 75 74 68 6F 72 6B 61 6E 65 6B 6F 32 30 t/author kaneko20
005BC0 30 39 2D 31 32 2D 31 31-20 31 31 3A 32 30 3A 33 09-12-11 11:20:3
005BD0 36 2D 01 06 00 01 23 1D-33 01 2F 72 6F 6F 74 2F 6-...#.3./root/
005BE0 74 69 74 6C 65 72 65 70-6F 72 74 20 41 32 30 30 title report A200
005BF0 39 2D 31 32 2D 31 31 20-31 31 3A 32 30 3A 33 36 9-12-11 11:20:36
005C00 0D 00 00 00 04 03 44 00-03 D4 03 A8 03 70 03 44 .....D... ..p.D
005C10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

8. created_at フィールドの値の確認 ('created_at' field)

created_at には、now を使って現在時刻を入れたので、値が違っているでしょう。

```

005B00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005B70 00 00 00 2F 03 06 00 01-21 21 33 01 2F 72 6F 6F .....!!3./roo
005B80 74 2F 64 61 74 65 32 30-30 39 2F 31 31 2F 32 39 t/date2009/11/29
005B90 32 30 30 39 2D 31 32 2D-31 31 20 31 31 3A 32 30 2009-12-11 11:20
005BA0 3A 33 36 2C 02 06 00 01-25 19 33 01 2F 72 6F 6F :36,...%.3./roo
005BB0 74 2F 61 75 74 68 6F 72 6B 61 6E 65 6B 6F 32 30 t/author kaneko20
005BC0 30 39 2D 31 32 2D 31 31-20 31 31 3A 32 30 3A 33 09-12-11 11:20:3
005BD0 36 2D 01 06 00 01 23 1D-33 01 2F 72 6F 6F 74 2F 6-...#.3./root/
005BE0 74 69 74 6C 65 72 65 70-6F 72 74 20 41 32 30 30 title report A200
005BF0 39 2D 31 32 2D 31 31 20-31 31 3A 32 30 3A 33 36 9-12-11 11:20:36
005C00 0D 00 00 00 04 03 44 00-03 D4 03 A8 03 70 03 44 .....D... ..p.D
005C10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
005C40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

この演習では **docid フィールドは無視**する。整数データはコード化されている。数値データのコード化体系はデータベース管理システムの種類によって違う。In this exercise, ignore the 'docid' field. The integer value is encoded.

9. キーとレコードの長さ (Key and Record Length)

005B00	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B10	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B20	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B30	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B40	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B50	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B60	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005B70	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	!13./roo
005B80	74 2F 64 61 74 65 32 30-30 39 2F 31 31 2F 32 39	t/date2009/11/29
005B90	32 30 30 39 2D 31 32 2D-31 31 20 31 31 3A 32 30	2009-12-11 11:20
005BA0	3A 33 36 2C 02 06 00 01-25 19 33 01 2F 72 6F 6F	:36,...%.3./roo
005BB0	74 2F 61 75 74 68 6F 72-6B 61 6E 65 6B 6F 32 30	t/authorkaneko20
005BC0	30 39 2D 31 32 2D 31 31-20 31 31 3A 32 30 3A 33	09-12-11 11:20:3
005BD0	36 2D 01 06 00 01 23 1D-33 01 2F 72 6F 6F 74 2F	6-...#.3./root/
005BE0	74 69 74 6C 65 72 65 70-6F 72 74 20 41 32 30 30	titlereport A200
005BF0	39 2D 31 32 2D 31 31 20-31 31 3A 32 30 3A 33 36	9-12-11 11:20:36
005C00	0D 00 00 00 04 03 44 00-03 D4 03 A8 03 70 03 44	D... ..p.D
005C10	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005C20	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005C30	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
005C40	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	

SQLite 3 Analyzer を用いたデータベース情報の確認 (Examine Database Information using SQLite 3 Analyzer)

1. **Sqliteman** を終了しておくこと (Exit the Sqliteman)

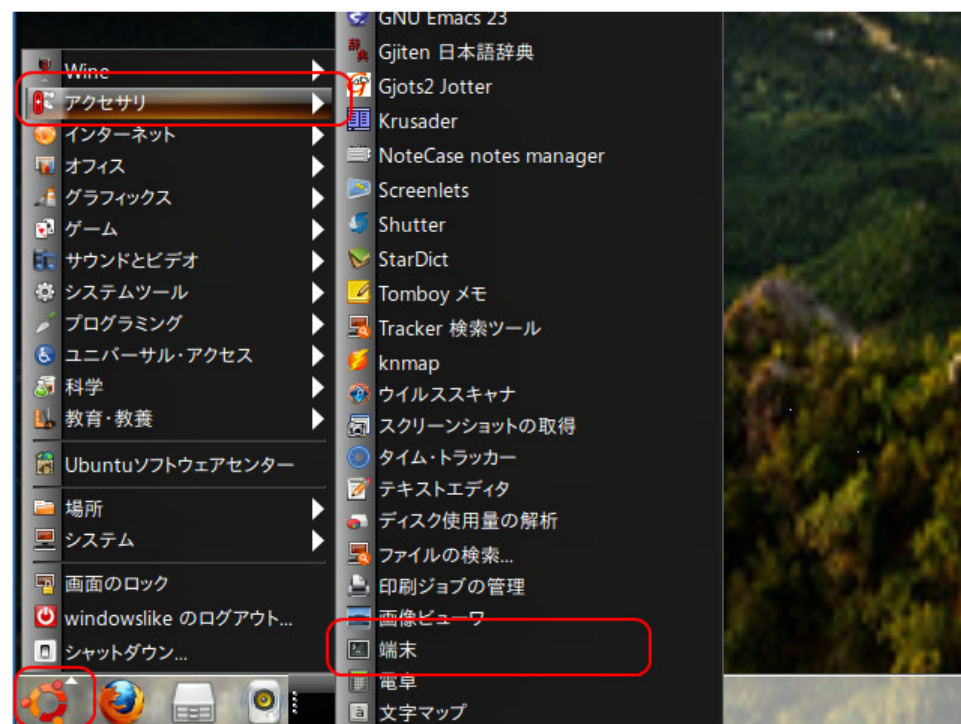
2. **SQLite 3 Analyzer** を用いた「**情報ファイル**」の生成

■ Ubuntu の場合

端末を開き、次のように操作する。

データベースの情報はファイル **dbinfo.sql** に格納することにする。このファイル名はなんでも良いが、アルファベットのみの使用が良い。

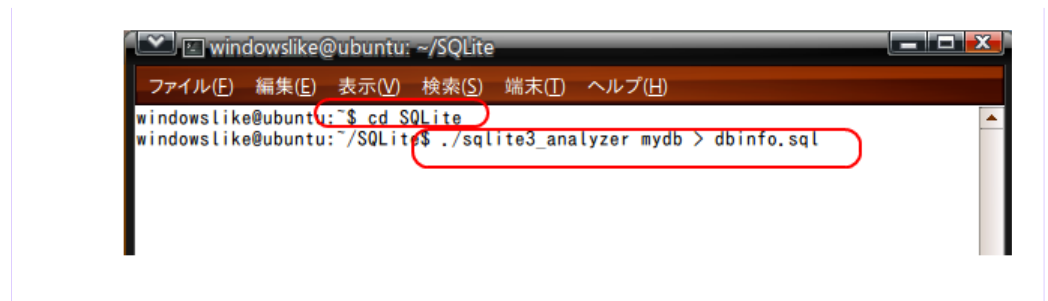
1. 端末を開きたいので、「アクセサリ」→「端末」と操作する



2. SQLite データベース・ディレクトリ **SQLite** に移り、**SQLite 3 Analyzer** を起動する。

このとき、**データベース論理名**として **mydb** を指定する。(The logical database name is 'mydb').

```
cd SQLite
./sqlite3_analyzer mydb > dbinfo.sql
```

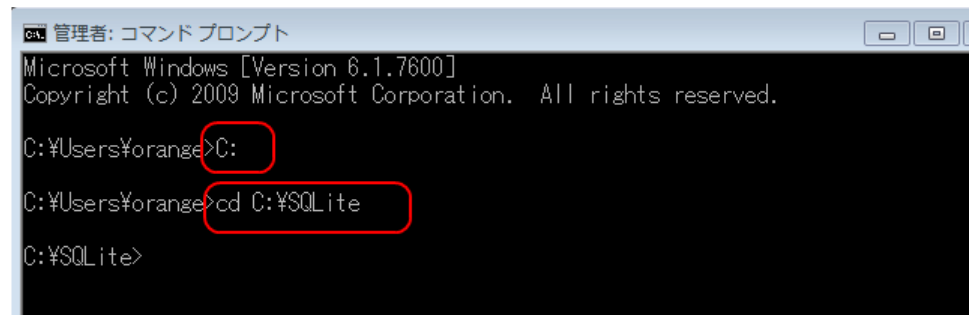


■ Windows の場合

1. Windows のコマンドプロンプトで、次の操作を行う。(Use the Windows command prompt, and do the following).

SQLite データベース・ディレクトリ **C:\SQLite** に移る。

```
C:
cd C:\SQLite
```



2. **SQLite 3 Analyzer の起動** (Start the SQLite 3 Analyzer).

このとき、**データベース論理名**として **mydb** を指定する。(The logical database name is 'mydb').

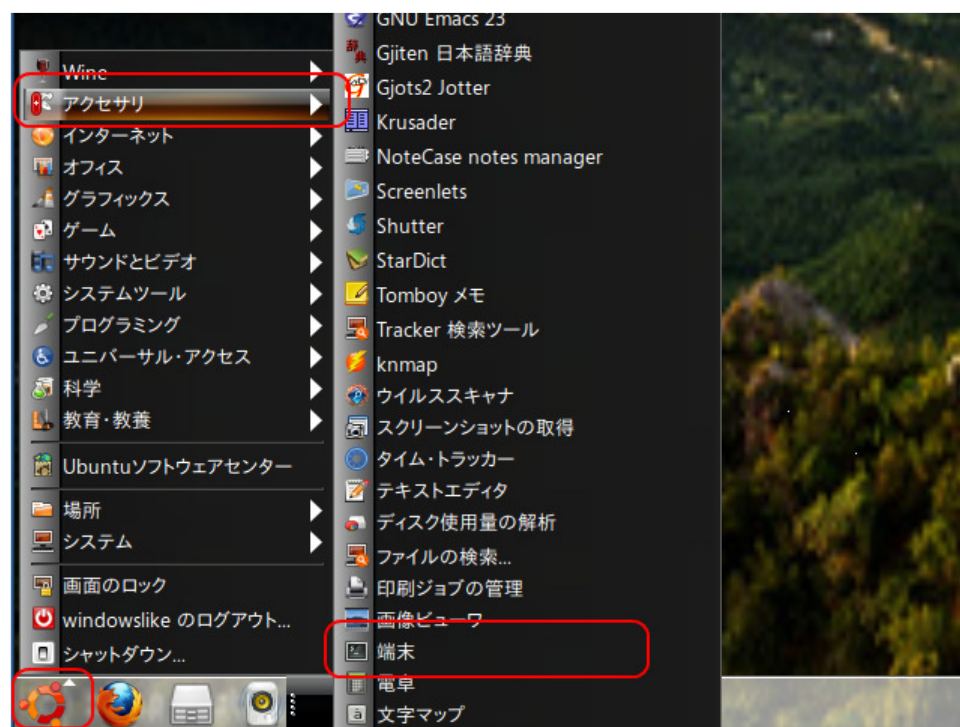
```
.\sqlite3_analyzer.exe mydb > dbinfo.sql
```

```
C:\SQLite>.\sqlite3_analyzer.exe mydb > dbinfo.sql
Analyzing table A...
Analyzing table B...
Analyzing table C...
Analyzing table D...
Analyzing table bundles...
Analyzing table order_records...
Analyzing table products...
Analyzing table reports...
Analyzing table requests...
Analyzing table results...
Analyzing table score_records...
Analyzing table shippings...
Analyzing table sqlite_master...
Analyzing table sqlite_sequence...
Analyzing index sqlite_autoindex_products_1 of table products...
Analyzing index sqlite_autoindex_results_1 of table results...
Analyzing index sqlite_autoindex_results_2 of table results...
Analyzing index sqlite_autoindex_score_records_1 of table score_records...
C:\SQLite>
```

3. **SQLite の起動** (Execute the SQLite. The database name 'mydb' is specified.)

■ Ubuntu の場合

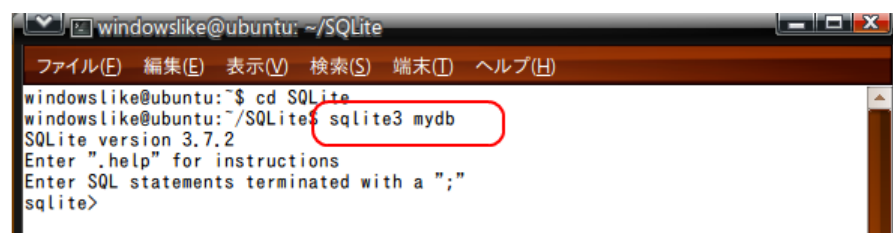
端末を開きたいので、「**アクセサリ**」→「**端末**」と操作する



SQLite データベース・ディレクトリ **SQLite** に移り、**SQLite** を起動する。

このとき、**データベース論理名**として **mydb** を指定する。(The logical database name is 'mydb').

```
cd SQLite
sqlite3 mydb
```

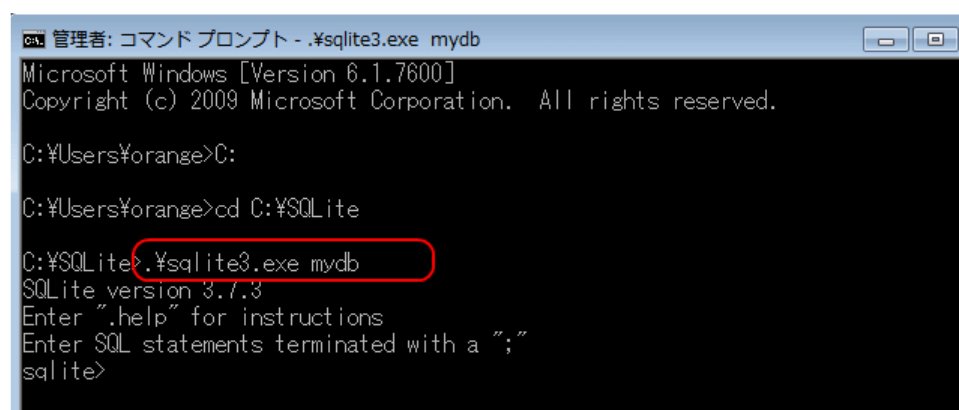


■ Windows の場合

SQLite データベース・ディレクトリ **C:\SQLite** に移り、**SQLite** を起動する。

このとき、**データベース論理名**として **mydb** を指定する。(The logical database name is 'mydb').

```
C:
cd C:\SQLite
.\sqlite3.exe mydb
```



4. **dbinfo.sql** の実行 (Execute dbinfo.sql)

```
.read dbinfo.sql
```

```
C:\SQLite>.\sqlite3.exe mydb
SQLite version 3.6.21
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>.read dbinfo.sql
```

5. **SQLite の終了** (End SQLite)

「.exit」で終了.

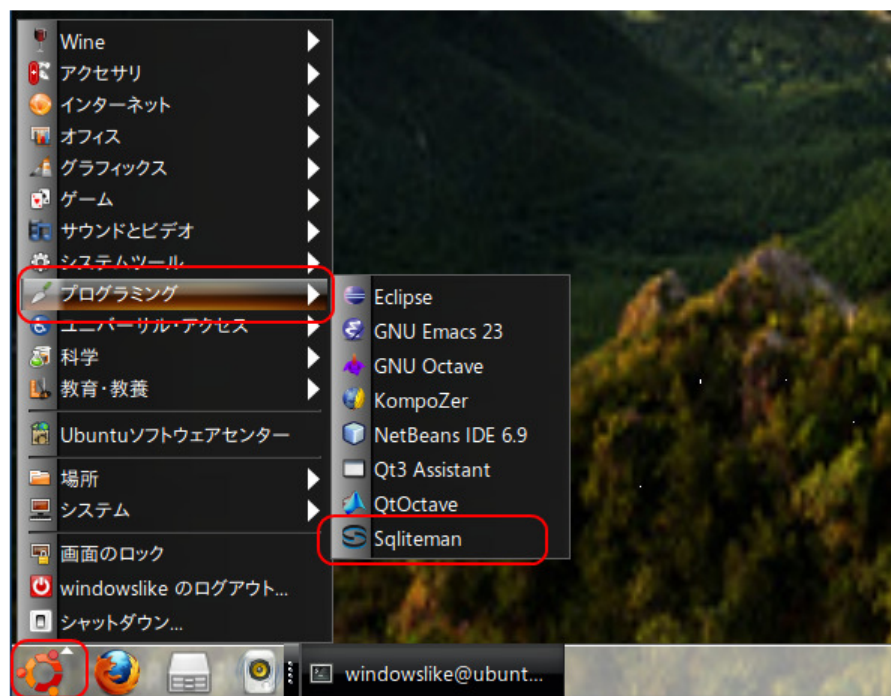
```
C:\¥SQLite>¥sqlite3.exe mydb
SQLite version 3.6.21
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .read dbinfo.sql
sqlite> .exit
C:\¥SQLite>
```

6. **Sqliteman の起動** (Start Sqliteman)

「Sqliteman」のアイコンをダブルクリック (double click "Sqliteman.exe")

■ **Ubuntu での SQLiteman の起動例**

「プログラミング」→「Sqliteman」と操作する.



SQLiteman の新しいウィンドウが開く. (A New window appears)

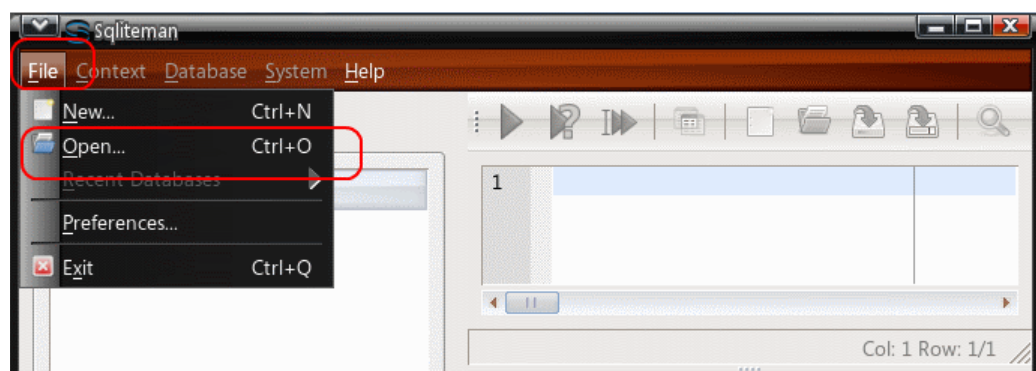
■ **Windows での SQLiteman の起動例**

「SQLiteman」のアイコンをダブルクリック (double click "SQLiteman.exe")



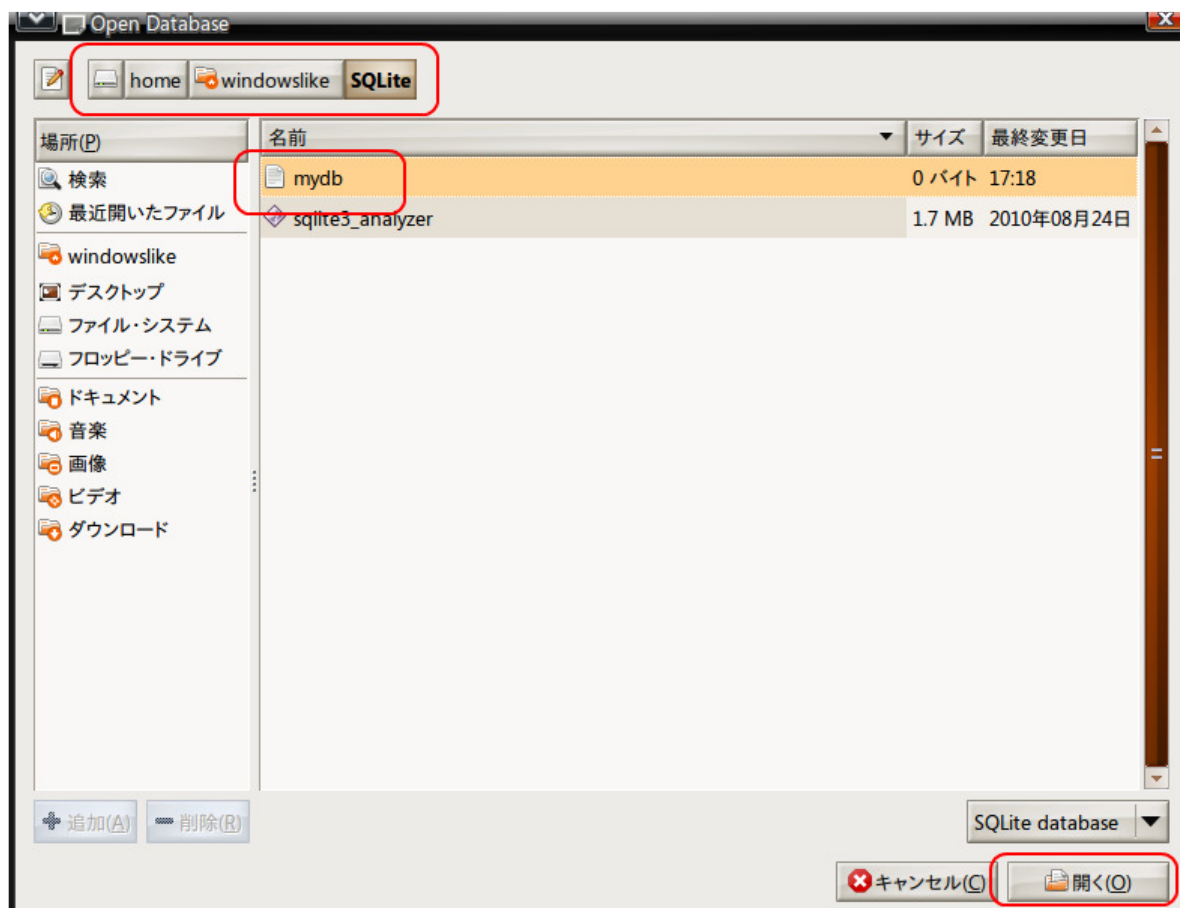
SQLiteman の新しいウィンドウが開く. (A New window appears)

7. 「File」→「Open」

8. **データベースファイルを開く** (Open Database File)

■ Ubuntu での実行例 (「SQLite/mydb」を開く場合)

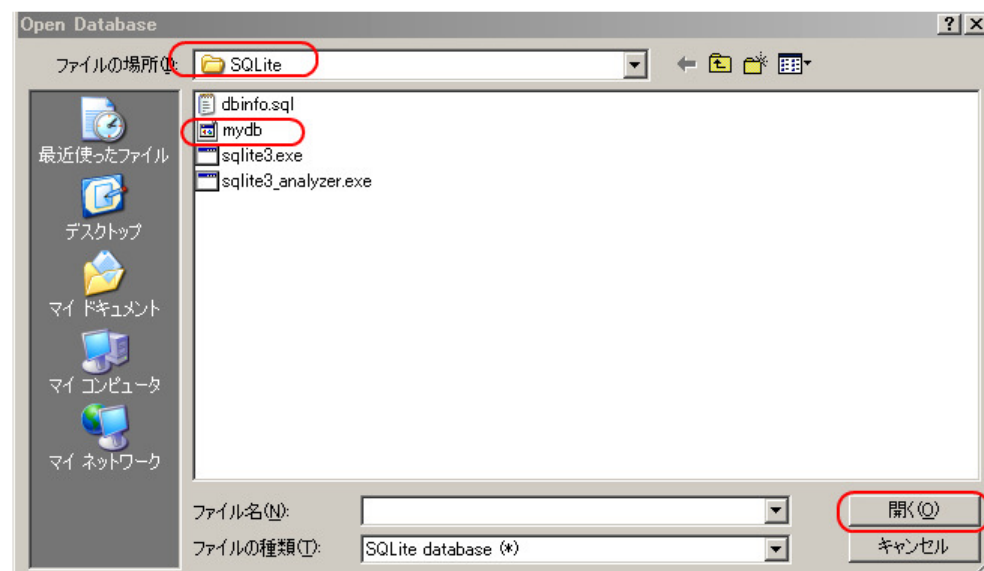
データベースファイル **SQLite/mydb** を選び, 「開く」をクリック (Click '開く' after choosing the database file "SQLite/mydb")



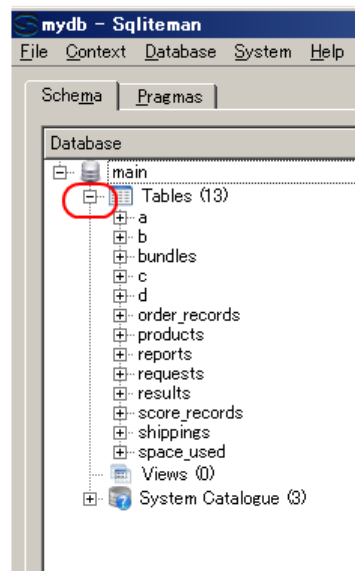
■ Windows での実行例 (「C:\SQLite\mydb」を開く場合)

データベースファイル **C:\SQLite\mydb** を選び, 「開く」をクリック (Click '開く' after choosing the database file "C:\SQLite\mydb")

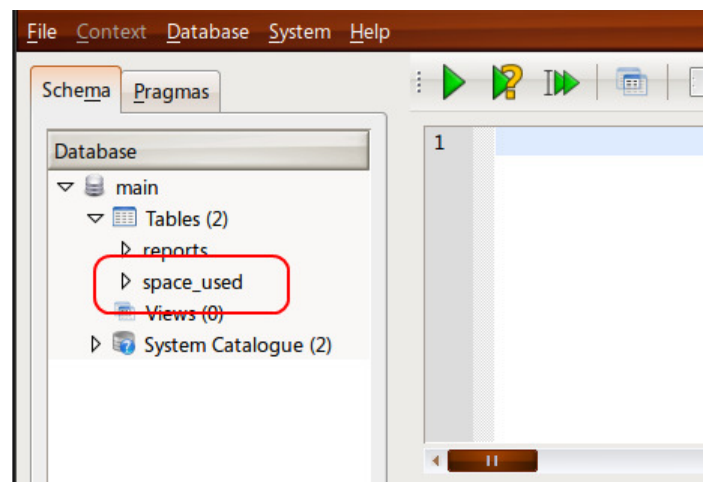
要するに, /home/<ユーザ名>/SQLite の下の mydb を選ぶ.



9. オブジェクト・ブラウザ (Object Browser) の中の「Tables」を展開すると, テーブルの一覧 (List of Tables) が表示される (List of tables appears by clicking 'Tables')



10. 次に、テーブル **space_used** を選ぶ (Select table 'space_used')

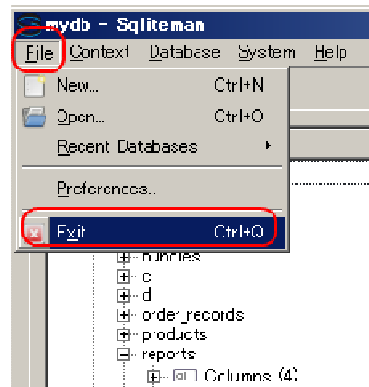


すると、データベースの情報が表示される

int_pages, leaf_pages, ovfl_pages の列は、は**各テーブルが占有しているページの数**を示している。(Number of pages which each table occupies)

	name	tblname	is_index	nentry	leaf_entries	payload	ovfl_payload	ovfl_cnt	mx_payload	int_pages	leaf_pages
1	A	A	0	2	2	27	0	0	15	0	
2	B	B	0	5	5	81	0	0	18	0	
3	C	C	0	5	5	71	0	0	16	0	
4	D	D	0	5	5	40	0	0	8	0	
5	bundles	bundles	0	5	5	145	0	0	29	0	
6	order_records	order_records	0	4	4	223	0	0	58	0	
7	products	products	0	4	4	185	0	0	49	0	
8	reports	reports	0	3	3	135	0	0	46	0	
9	requests	requests	0	4	4	132	0	0	33	0	
10	results	results	0	5	5	91	0	0	20	0	
11	score_records	score_records	0	5	5	186	0	0	39	0	
12	shippings	shippings	0	4	4	116	0	0	29	0	
13	sqlite_master	sqlite_master	0	21	17	3385	0	0	650	1	
14	sqlite_sequence	sqlite_sequence	0	6	6	77	0	0	17	0	
15	sqlite_autoindex_products_1	products	1	4	4	76	0	0	21	0	
16	sqlite_autoindex_results_1	results	1	5	5	81	0	0	18	0	
17	sqlite_autoindex_results_2	results	1	5	5	40	0	0	8	0	
18	sqlite_autoindex_score_re...	score_records	1	5	5	81	0	0	18	0	

11. **Sqliteman** を終了 (Exit the Sqliteman)



演習問題

次の問いに答えよ。 Answer the following questions.

問い (Questions)

次の PTABLE テーブルに関する問題 (About the following 'PTABLE' table)

name	type	color
apple	fruit	red
apple	fruit	blue
rose	flower	white
rose	flower	red
rose	flower	yellow

SQL を用いて PTABLE のテーブル定義を書きなさい (Write the table definition of the table PTABLE using SQL)

SQL を用いて属性 name の二次索引を生成しなさい (Write a SQL to generate a secondary index on the attribute 'name')

このテーブル定義には PRIMARY KEY が無いので、データベース管理システムは、各レコードのキー値 (Key Value) を自動生成する。

Database system generates a key value for each record automatically.