

FrontBase ユーザーズガイド

校正前 - ドラフト
2003 年 1 月 24 日版

A10 (<http://www.a10-dev.com/>)

注意 - 重要:

- ・日本語ドキュメントは A10 が翻訳を実施した。翻訳内容に関する問い合わせは A10 をお願いいたします。
- ・校正前のドラフト版です。多くの問題が有ることが考えられます。原文（英語）の参考程度にお使いください。
- ・翻訳作業には多くの時間を必要とします。同時に多くのユーザーが早く日本語で読みたいと言う要望もあります。校正前のドラフト版では間違った解釈を行うリスクも同時に発生します。十分に注意してお使いください。
- ・ドキュメントの内容を利用する場合は、各自の責任でお使いください。A10 は直接的、間接的な総ての損害を負いません。



2002年3月28日に更新しました
FrontBaseは絶えず進歩しているから、このマニュアルの情報のいくつかが正確でないかもしれません。最近の最新の情報についてはFrontBase ディストリビューションでリリースノートを読んでください。

FrontBase は、FrontBase 社とその使用権付与者が © 2000 年に著作権を取りました。すべての権利は当方に帰属します。

これらのページの文書は個人使用のための著作権使用による印刷は行ってもよいです。しかし、前述以外の、この文書の一部をいかなる形式で送信したり、コピーしたり、記録することを含む、電子的にあるいは機械的に、あるいはすべての情報記憶検索システムで、FrontBase 社から文書で許可を得ず複製してはいけません。

FrontBase と FrontBase ロゴは FrontBase 社の登録商標です。

すべての他の商標と登録商標はそれぞれの所有者の財産です。

コンパクトディスクのすべてのソフトウェアと文書は CD パンフレットのライセンス契約の適用を受けています。

FrontBase と連絡を取る方法：

アメリカ合衆国とインターナショナル

FrontBase 社
27042 Towne Center Drive
Suite 265
Foothill Ranch, CA 92610

FrontBase 社
Datavej 52
DK-3460 Birkerød
Denmark

注文 電話：+1 949 455 2913
ファックス：+1 949 457 0966

電話：+45 4582 6262
ファックス：+45 4582 0816

World Wide Web <http://www.frontbase.com>

技術サポート support@frontbase.com

インフォメーション info@frontbase.com

販売、マーケティング sales@frontbase.com

ライセンスを受ける license@frontbase.com

目次

1	手引き	1 3
2	はじめに	1 5
	アーキテクチャ	1 5
	クライアント/サーバ・アーキテクチャ	1 5
	リレーショナル・データベースサーバー	1 5
	高度なスケーラブル (大きなものから小さなものまで)	1 5
	SQL 9 2 の問合せ言語	1 6
	固有のインタフェース	1 6
3	導入	1 7
	概要	1 7
	サポートしているプラットフォーム	1 7
	前もって設計する	1 8
	しっかりした土台	1 8
	規格の適合	2 0
	完全な SQL 9 2 の適合	2 0
	ユニコード文字の表示	2 1
	TCP/IP クライアント/サーバの相互作用	2 1
	ANSI C Codebase	2 1
	キーの特徴	2 2
	テラバイトデータベース、ギガバイトカラム値	2 2
	FrontBase セキュリティ	2 3
	トランザクション	2 5
	列レベルの特権	2 5
	生のバックアップ	2 5
	キャッシュすること	2 5
	マルチサーバデプロイメント	2 6
	ドライバー、アダプタとプラグイン	2 7
	ODBC と JDBC	2 7

WebObjects 5 プラグイン	2 7
PHP3 と PHP4 アダプタ	2 7
perl アダプタ	2 7
Omnis Studio DAM	2 8
REALBasic	2 8
Tcl ドライバ	2 8
EOF アダプタ	2 8
移行	2 9
FileMaker	2 9
MySQL	2 9
FrontBase 関連のプロセス	2 9
FBExec	2 9
FrontBase	3 0
FBWebEnabler	3 1
プロセス識別	3 1

4 インストール 3 3

FrontBase をダウンロードする	3 3
FrontBase をインストールする	3 4
MacOS	3 4
Windows	3 7
Linux	3 9
Unix	5 5
FrontBase のライセンス	6 1
ライセンスを得る	6 1
ライセンスをインストールする	6 2
FrontBase のディレクトリの構造	6 3
ディレクトリの場所	6 3
ディレクトリの内容	6 3
FrontBase を隠す	6 5
MacOS X と MacOS X サーバー 10.x	6 6
WindowsNT / 2 0 0 0	6 7
ソラリス	6 8
FreeBSD	6 9
RedHat	7 0

Mandrake.....	7 1
SuSE.....	7 2
Debian	7 3
最新の状態を保つ	7 4
最新バージョンを決定する	7 4
あなたのFrontBase サーバーをアップグレードする.....	7 5
5 基本概念	7 7
SQL 9 2 の概念	7 7
カタログ.....	7 8
スキーマ.....	7 8
ユーザ.....	8 0
日付、時間と時刻表示	8 1
キーワードと識別子	8 2
もっと多くのSQL 9 2 についての知識.....	8 2
トランザクションを理解する	8 3
並列アクセス	8 3
トランザクション	8 3
更新の力業.....	8 4
アイソレーションレベル.....	8 4
ロックの仕方	8 5
ロックと EOF	8 6
6 アドミニストレーション	8 7
アドミニストレーションツール	8 8
FrontBaseManager	8 8
FBWebManager	8 8
sql92	8 8
FBScriptAgent	8 8
トランザクションのログを書き出す	8 9
実装	9 0
アドミニストレーション.....	9 1
SQL の文法.....	9 2
オプション	9 3

ユーティリティ	9 3
複製	9 5
複製	9 5
クラスタリング	1 0 3
実装	1 0 3
バックアップと復帰	1 0 8
概要	1 0 8
データベースファイルをコピーする	1 0 9
生のデータベースのバックアップ	1 0 9
データベースを復帰する	1 1 0
書式のないファイルへの書き出し	1 1 0
複製	1 1 1
拡張した書式のないファイルの読み込みと書き出しの関数 ..	1 1 1
拡張した書式のないファイルの書き出しの関数	1 1 1
拡張した書式のないファイルの読み込みフィルター	1 1 2
インデックスを付ける	1 1 6
インデックスの管理	1 1 6
戦略	1 1 7
事例:	1 1 7
FrontBase のチューニング	1 2 0
データベースサーバーパフォーマンス	1 2 0
FrontBase のキャッシュするメカニズム	1 2 0
いつキャッシュを調整したらいいですか?	1 2 2
デスクリプタキャッシュ	1 2 2
表をキャッシュする	1 2 3
未使用のデバイスドライバ (RDD)	1 2 6
改善しているパフォーマンス	1 2 8
Mac OS X と未使用のデバイス	1 2 9
メモリの使用法	1 3 0
移行	1 3 2
FileMaker	1 3 2
MySQL	1 3 3
トラブル・シューティング	1 3 4
SQL 文のログ	1 3 4
SQL ログファイルの場所	1 3 6

新しいSQL ログファイル	1 3 6
---------------------	-------

7 FrontBaseManager	1 3 7
---------------------------	--------------

FBManager からの変化	1 3 8
データベースを監視して、そして管理する	1 4 1
データベースを作る	1 4 3
バックアップに復帰する	1 4 5
監視	1 4 7
データベースをスタートして、そしてストップする	1 4 8
データベースに接続する	1 4 9
接続ウインドウ	1 5 1
SQL インタプリター	1 5 2
データベース	1 5 4
セッション	1 5 5
使用法	1 5 6
ライセンス	1 5 8
ユーザ	1 6 0
スキーマ	1 6 1
スキーマオブジェクト	1 6 2
新しいスキーマオブジェクト	1 6 3
新しい表	1 6 3
新しいビュー	1 6 5
新しい手順と新しい関数	1 6 6
新しいコレーション	1 6 7
オープンスキーマオブジェクトの内容	1 6 8
列を挿入して、そしてアップデートする	1 6 9
BLOBとCLOBの取り扱い	1 7 0
オープンスキーマオブジェクトの定義	1 7 1
表を定義するウインドウ	1 7 2
カラム定義パネル	1 7 2
主キーパネル	1 7 3
フォーリンキー(Foreign Key)パネル	1 7 5
ユニークパネル	1 7 7
チェックパネル	1 7 9
インデックスパネル	1 8 1

完全なテキストインデックス	1 8 4
特権	1 8 5
SQL	1 8 8
表をキャッシュする	1 8 9
ブラックリストとホワイトリスト	1 9 1
バックアップ	1 9 4
奇数の n の終わり	1 9 4
プリファレンス	1 9 5
ファイル読み込み	1 9 6
SQL ログ	2 0 2

8 FBWebManager 2 0 5

データベースを監視する	2 0 5
データベースに加える	2 0 5
モニターにデータベースを加える	2 0 7
データベースから隠す	2 0 9
データベースを管理する	2 0 9
データベースを作る	2 0 9
データベースをスタートする	2 1 3
データベースを改名する	2 1 4
モニタービューを新しくする	2 1 4
データベースの管理	2 1 4
データベースをストップして、そして削除する	2 1 6
SQL 9 2	2 1 6
ユーザ管理	2 2 2
データベースパスワード	2 2 6
表のキャッシュの管理	2 2 7
ホワイトリストとブラックリストの管理	2 2 8
取り扱いの管理	2 3 0
キーワード	2 3 2
ライセンスを加える	2 3 4

9 sql92 2 3 5

コマンド文法	2 3 5
--------------	-------

オプション	2 3 5
一般的	2 3 7
sql92 によって解釈したコマンド	2 3 7
接続	2 3 7
データベースを作る	2 3 8
Blob を定義して、そして Clob を定義する	2 3 8
データベースを削除する	2 3 9
切断	2 3 9
スクリプト	2 4 0
出口と終了	2 4 0
接続をセットする	2 4 0
データベースパスワードをセットする	2 4 0
パスワードをセットする	2 4 1
データベースをスタートする	2 4 1
データベースをストップする	2 4 1
Autostart	2 4 2
接続を示す	2 4 3
使用方法を示す	2 4 3
データベースログを示す	2 4 3
データベースを示す	2 4 4
ライセンスを示す	2 4 4
ログを示す	2 4 4
メモリを示す	2 4 4
キャッシュを示す	2 4 5
トランザクションを示す	2 4 5
スキーマを示す	2 4 5
表を示す	2 4 5
ビューを示す	2 4 6
サイズを示す	2 4 6
履歴を示す	2 4 6
複製をスタートする	2 4 6
クライアントを作る	2 4 7
クライアントを加える	2 4 7
クライアントを隠す	2 4 7
クライアントを示す	2 4 8

SMALLINT	2 6 4
INTEGER, INT	2 6 4
LONGINT	2 6 5
DECIMAL[(<precision> [, <scale>])]	2 6 5
NUMERIC[(<precision> [, <scale>])]	2 6 5
FLOAT[(<precision>)]	2 6 6
REAL	2 6 6
DOUBLE PRECISION	2 6 6
CHARACTER, CHAR	2 6 6
NATIONAL CHARACTER, NATIONAL CHAR, NCHAR ...	2 6 7
CHARACTER VARYING, CHAR VARYING, VARCHAR ...	2 6 7
NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, NCHAR VARYING ..	2 6 8
BIT	2 6 8
BIT VARYING	2 6 8
BYTE	2 6 9
DATE	2 6 9
TIME	2 6 9
TIME WITH TIME ZONE	2 7 0
TIMESTAMP	2 7 0
TIMESTAMP WITH TIME ZONE	2 7 0
INTERVAL	2 7 1
BLOB	2 7 1
CLOB	2 7 2
BOOLEAN	2 7 2
FrontBase データタイプの土台のマッピング / Java の オブジェクト ..	2 7 2
文字列	2 7 3
整数	2 7 4
10進数	2 7 4
日付	2 7 5
時間	2 7 5
ストリームデータ	2 7 5
主キー	2 7 6
主キーと自動生成	2 7 6
キーの生成	2 7 6

列レベルの特権	2 7 7
定義	2 7 7
デプロイング	2 7 8
あなたがコレクションのためにできること	2 8 0
国際的な文字	2 8 0
大文字小文字の違いを無視する比較オペレーション	2 8 0
あなた自身のアプリケーション、あるいはソリューションに FrontBase を組み込む	2 8 4
ディレクトリ構造	2 8 4
FrontBase サーバーをスタートする – Windows NT/2000/XP	2 8 7
FrontBase サーバーをスタートする – Windows NT/2000/XP 以外のプラットフォーム	2 8 8

手引き

FrontBase ユーザーズガイドは動作の指針を提供することと同様、使い易さのために考案した文書です。

この文書では次の章が入っています：

- － 15 ページの「はじめに」。
- － 17 ページの「概要」。
- － 33 ページの「インストール」。
- － 87 ページの「アドミニストレーション」。
- － 77 ページの「基本概念」。
- － 137 ページの「FrontBaseManager」。
- － 205 ページの「FBWebManager」。
- － 235 ページの「sql92」。
- － 249 ページの「デベロッパーのための FrontBase」。

はじめに

FrontBase はスケーラブル（大きなものから小さなものまで）なり
レーショナル・データベースサーバーです。少数の一般的な概念はそ
れが何を意味するか説明するのに役立つでしょう。

アーキテクチャ

クライアント／サーバ・アーキテクチャ

これらの条件は World Wide Web で誰にでもよく知られています。
Web と共に、ブラウザ（クライアント）は若干の Web サイト（サー
バー）からページのリクエストをします。Web サイトはリクエストを
処理して、そして結果（Web ページ）を表示します。FrontBase サー
バーは Web サーバの若干の方法と類似しています。それは
FrontBase クライアントからリクエストを聞いて、リクエストを処
理して、そして結果を返します。

リレーショナル・データベースサーバー

しかし、FrontBase はリレーショナル・データベースサーバーです。
Web サーバがブラウザの表示のために一般に Web ページをサポート
する間に、FrontBase はデータを記憶して、そして引き出す請求を
処理します。「リレーショナル」モデルを実行することによって、
FrontBase はデータをアプリケーションによって定義した表にスト
アします、その中に、アプリケーションによって定義した関係が存在
するかもしれません。表はそれらの行によって定義します。例えば、
電話帳の表が次の行を持っているでしょう：名前、phone_number。
実際のデータは表の列で入力されます。電話帳の表の列が名前の行
に「John Doe」と phone_number の行に「5 5 5 - 1 2 1 2」を
持っているかもしれません。

高度なスケーラブル（大きなものから小さなものまで）

FrontBase は高度なスケーラブル（大きなものから小さなものまで）
です、そしてそれは非常に大きいデータのセットとリクエストの高い
負荷を処理することができることを意味します。データのセットす
る大きさは一般に表の列の番号によって表わします。その列に索引
を付けている特徴を使って、FrontBase は効率的に百万の列の表に

データを挿入して、そして参照することができます。リクエストの高い負荷のアプリケーションで、FrontBase の複製とクラスタリングの特徴は複数のサーバで同じデータベースを走らせるために利用することができます。

SQL 92 の問合せ言語

FrontBase は工業規格 SQL 92 の問合せ言語を実装します。FrontBase クライアントはこの言語を記憶するために使って、そしてサーバでデータを引き出します。それらは同様にユーザを管理するために言語を使って、性能を調整して、そして他のアドミニストレーションのタスクをします。FrontBase は非常に厳密に SQL 92 規格を実装しますが、FrontBase 特定の問題を処理するために同様に拡張子を持っています。FrontBase はクライアント/サーバ通信のために TCP/IP のような、他の規格を、そして文字の値の記憶装置のためにユニコードを使います。

固有のインタフェース

スケーラビリティ（大きなものから小さなものまで）、そして規格の執着という特徴で、FrontBase は今日の特別製のクライアント/サーバアプリケーションのためにそしてビルドしているダイナミックな Web サイトのために理想的なデータベースです。アプリケーションの開発者は特別製のアプリケーションを開発するために FrontBase の FBCAccess C ライブラリ、JDBC、ODBC、Tcl ドライバと他のインタフェースを使うことができます。それらはダイナミックな Web サイトを開発するために FrontBase のアダプタの配列を使うことができます。現在 PHP、perl と WebObjects / EOF、REALBasic プラグインと Omnis Studio DAM のためにアダプタがあります。

導入

FrontBase は今日の品質重視の開発者とユーザの規格と要求に従う高性能なリレーショナル・データベースエンジンです。

この章では次のセクションが入っています：

- ・ 17 ページの「概要」。
- ・ 20 ページの「規格の適合」。
- ・ 22 ページの「キーの特徴」。
- ・ 27 ページの「ドライバー、アダプタとプラグイン」。
- ・ 29 ページの「移行」。
- ・ 29 ページの「FrontBase 関連のプロセス」。

概要

エンジンは純粋な ANSI C で書かれています、そしてコンパイラとランタイムシステム、組み込みのシステム、オブジェクト指向プログラミング、データベースシステムとコマンドとコントロールのシステムで15+年の経験から利益を得ます。

このセクションの見出しは次の通りです：

- ・ 17 ページの「サポートしているプラットフォーム」。
- ・ 18 ページの「前もって設計する」。
- ・ 18 ページの「しっかりした土台」。

サポートしているプラットフォーム

- － 34 ページの「MacOS」。
- － 37 ページの「Windows」。
- － 39 ページの「Linux」。

－ 55 ページの「Unix」。

前もって設計する

FrontBase は次のような国際的な、そして事実上の規格に高性能と適合を提供します：

SQL 92：FrontBase はこの重要な国際規格を持っている合致で最初の産業的な強力なデータベースエンジンです。これはエンジンのすぐ中でビルドをチェックして完全に矛盾のない拘束をします。

ユニコード：表示のために UTF-8 という規格を使うことによって、スペースを節約する間に、FrontBase は排他的にユニコード 2.0 をすべてのキャラクタデータを処理するために使います。これは入り混ざったクライアント環境とそれらのさまざまな文字セットに容易なサポートを提供します。

通信：FrontBase はソケットを、開発者が多種多様なクライアントプラットフォームをサポートすることを容易にして、クライアントと通信するために使います。

アドミニストレーション：FrontBase データベースはインターネット上ですべてのコンピュータで管理することができます－標準的な Web ブラウザは必要とするすべてです。

しっかりした土台

根本的に正確なりレーショナル指向のエンジンは非常に効率的に、そして制限条件なしでデータベースの取扱いにしっかりした土台を提供します：

- － 厳しいトランザクションコントロール
- － クラッシュに対して 100% の回復力
- － 超高速の起動時間
- － テラバイトのサイズのデータベース
- － ギガバイトのサイズの CHARACTER / VARCHAR 文字列と BLOBs
- － 複数の行が非常に少ないオーバーヘッドで Bツリーの索引をつけることを最適化しました
- － ホスト OS ファイルシステムの独立

- － メモリに表をキャッシュすること
- － バージョン番号を読むことでアイソレーションする
- － 列レベルをロックする機能
- － 読み込み専用のデータベース

規格の適合

このセクションは FrontBase が執着する国際規格を記述します。これらは SQL 9 2、ユニコード、TCP/IP と ANSI C を含みます。

FrontBase と共にアプリケーションを開発して、そして展開するとき、FrontBase はこれらの規格を影響することができることを保証して、いくつかの国際規格に執着します。

このセクションでは次のことを論じるでしょう：

- 2 0 ページの「完全な SQL 9 2 の適合」。
- 2 1 ページの「ユニコード文字の表示」。
- 2 1 ページの「TCP/IP クライアント／サーバの相互作用」。
- 2 1 ページの「ANSI C Codebase」。

完全な SQL 9 2 の適合

FrontBase は完全な SQL 9 2 の規格を実装します。重大な処理は規格によって定義したすべての特徴を実装して、そして、規格によって定義するように、それらを実装しました。この文書は FrontBase で SQL 9 2 を使うためのいくつかの入門解説の事例を提供します、しかし包括的なリストについては、SQL リファレンス文書を見てください。SQL 9 2 の究極の手引きは規格自身です。あなたは、報酬として ANSI からの規格「**ANSI の SQL 9 2 規格**」を得ることができません：

<http://webstore.ansi.org/>

有名なデータベース専門家 C. J. Date と Hugh Darwen による優秀な本は次の通りです：

SQL 規格の手引き、第 4 版

<http://www.amazon.com/exec/obidos/ASIN/0201964260>

それは理論的に近い提供するので、非常に面白い本です。Date と Darwen は究極の SQL 9 2 規格の判断をもたらした多くのファンではありません。

ユニコード文字の表示

FrontBase はユニコードを使って、すべての文字データ（CLOBs も含める）を記憶します。照合（COLLATIONS）のために FrontBase のサポートと結び付いて、これはサーバーサイドの文字列の比較が標準的な ASCII 以外の文字セットで動作することを保証します。

ユニコードのための FrontBase のサポートはクライアントプラットフォームでスムーズに動作します。文字データは、クライアントオペレーティング・システムのネイティブのユニコードライブラリを使って、クライアント側でユニコードに変換します。ユニコードフォーマットでの文字データは、FrontBase サーバーにパスして、そしてそこで記憶します。クライアントがサーバーから文字データを引き出すとき、クライアントは、クライアントプラットフォームでユニコードフォーマットから使用に適したフォーマットに変換します。

TCP/IP クライアント／サーバの相互作用

FrontBase クライアントとサーバーは TCP / IP インターネットプロトコルの規格を通信するために使います。これは FrontBase サーバーに組み込むシステムを設計して、そして展開する方法でものすごい自在に使うことを可能にします。例えば、WebObjects デプロイメントで、いくつかのアプリケーションサーバーをサーバーエリアネットワーク（SAN）の FrontBase サーバーのクラスターに接続してもよいです。あるいは、もしこのような性能そして／あるいは冗長性を必要としないなら、あなたは単独の FrontBase サーバーで WebObjects、アパッチと FrontBase を走らせることができます。

ANSI C Codebase

FrontBase は ANSI C で書かれます。これは安定したクロスプラットフォーム codebase を保証して、そして我々に FrontBase を最小の努力使って新しい UNIX ベースのプラットフォームに動かすことを許します。これは FrontBase の顧客に開発とデプロイメントのプラットフォームで自在に使うことを提供します。

FBCAccess クライアントライブラリは同様に ANSI C で書かれています。FBCAccess のソースコードは特別なリクエストによって利用可能です、それで、もしあなたが FrontBase サーバーがポートしなかった（PalmOS あるいは PocketPC のような）プラットフォームで FrontBase クライアントを展開する必要があるなら、できます。

キーの特徴

このセクションはインターネットアプリケーションのために理想的なデータベースにする FrontBase のキーの特徴を紹介します。FrontBase は「大きい少年たち」の重要な特徴の大部分をサポートして、そして無料の、そしてオープンソースのやり方よりむしろ規格に従っています。

このセクションでは次のことを論じるでしょう：

- テラバイトデータベース、ギガバイトの行の値
- FrontBase セキュリティ
- トランザクション
- 列レベルの特権
- 生のバックアップ
- キャッシュすること
- マルチサーバーのデプロイメント

テラバイトデータベース、ギガバイトの行の値

FrontBase はテラバイトサイズのデータベース、ギガバイトサイズの文字の行の値とギガバイトサイズのバイナリの大きいオブジェクト（BLOBs）と文字の大きいオブジェクト（CLOBs）をサポートします。

FrontBase はデータベースを記憶するために使ったそれ自身のファイルのファイルシステムを実装します。FrontBase 2.26 のファイルシステムは、有用なスペースの 2 テラバイト以上を与えて、最高 2^{32} バイトのブロックから成り立ちます。FrontBase 2.x の性能は再び増えました。FrontBase <バージョン番号> は今低レベルのディスクアクセスのために 2048 バイトのブロックサイズを使います、そしてそれは、ほとんどの場合、4 つのファクターでディスクアクセスを減らします。これは 4 テラバイトデータベースの方へ同様にイネーブルします。FrontBase の将来のバージョンは 64 ビットアドレスの能力を使うでしょう – FrontBase 開発チームは現在低いレベルディスクアクセスのペタからエクサバイトデータベースがいくつかの物理装置に広めるのに処理するアドレス能力を拡張しています。文字の行の値として、BLOB、あるいは CLOB が 2^{32} バイトまで占領することができます。実際は、データベースのために全部のアクセス可能なスペースを費やさないように、大きいオブジェクトはずっとより小さいでしょう。それで、我々はギガバイトサイズの行の値を公表します。

理論的なファイルサイズがギガバイト範囲を超えることができないプラットフォームで、FrontBase は、必要であるとき、ファイルシステムの限界の中で適していたいくつかのファイルの中のデータベースのためにその記憶装置を壊すでしょう。FrontBase がこれらのファイルの中でそれ自身のファイルシステムを維持するので、このパーティションを切ることは文字の行の値、BLOBs、あるいは CLOBs のサイズに限度を押し付けません。

FrontBase セキュリティ

FrontBase はセキュリティのためにパスワード、暗号とクライアント IP アドレスチェックを使います。

パスワード

FrontBase はデータベースへのアクセスを保護してパスワードの 2 つのレイヤ：データベースパスワードとユーザパスワード、を提供します。

1. データベースパスワード

もしデータベースパスワードをセットしているなら、クライアントが接続プロトコルの一部としてデータベースパスワードをサーバーに送られるはずですが、もしサーバーがパスワードを確かめることができないなら、クライアント接続は即座に閉じます。

2. ユーザパスワード

それぞれのデータベースユーザがパスワードを持つことができます。セッションがそのユーザについて作るとき、パスワードはサーバーによって確認します。もし確認が失敗するなら、セッションは作られません。セッションがうまく作られたとき、SQL 9 2 の規格によって定義した保護は引き継ぎます。

3. 一般的なパスワードの処理

パスワードはどんな長さでもよいです。パスワードはクライアントソフトウェアの外に決してさらされません、そしてそれらはデータベースにさえいません。アプリケーションが FrontBase クライアントライブラリにパスワードを送るとすぐに、一方的な関数がパスワードの要約を発生させるために適用します。要約からパスワードを推定することができないように、関数はパスワードの一部を廃棄するでしょう。ユーザ名は要約の一部です、それで同じパスワードで 2 つのユーザが同じ要約を持たないでしょう。パスワードの要約はサーバーに送られて、そしてパスワードの代わりに確認のために使います。

暗号

暗号は通信チャンネルとデータ記憶装置を保護するために使います。FrontBase データベースを作成するとき、あなたは任意にディスクに記憶したデータが暗号化することを指定してもよいです。あなたは同様に任意にサーバーとそのクライアントの間の通信チャンネルが安全であるはずであることを指定することができます。あなたは指定したそれぞれのオプションに暗号化のキーを提供するはずで

1. データ暗号化

ディスクで記憶したデータが 512 バイトブロックに暗号ブロック連鎖方式で 3 重の DES を使って暗号化します。データの記憶装置自身は 512 バイト / ブロックでブロック指向です、それでこれは効果的に、表の定義、表の内容、文字データと BLOBs を含む、すべてのデータを暗号化します。初期化ベクターは、システムの中で、ブロックの理論的な立場に依存します、それで同じ内容を持っているブロックは決して同じ暗号文のブロックを発生させないでしょう。データの暗号化のために使ったキーは DES の暗号化の 64 ビットの初期化ベクターと 3x56 ビットです。

2. 通信の暗号化

クライアントとサーバーは安全なチャンネルを確立することが可能です。クライアントがサーバーに接続するとき、それはサーバーからパブリックな RSA のキーを受け取ります。クライアントは、ランダムなセッションキー：出て行くデータと入ってくるデータのセットを発生させます。それはパブリックな RSA のキーでそれらのセッションキーを暗号化して、そして結果をサーバーに送ります。サーバーはプライベートなキーを使ってクライアントによって送られたセッションキーを解読します。それで、クライアントとサーバーは秘密キーの共通のセットを確立しました。

通信データの暗号化のために使ったアルゴリズムは暗号文とクリアテキストフィールドバックでバイトストリーム方式の 3 重の DES です。明確なテキストフィールドバックはエラーの後に続くすべてのバイトにエラーが繁殖するであろうことを保証します。これはエラーの単純な検出を保証して、そしてただ小さい量の冗長性だけを導入します。

IP アドレスチェック

FrontBase はどのクライアントが接続するか決定することについてブラックリストとホワイトリスト（同様に「ウイスキーリスト」として知られている）を実装します。

クライアントが FrontBase サーバーに接続するとき、クライアントの IP アドレスはブラックリストとホワイトリストに反してチェックします。もし IP アドレスがブラックリストであるなら、接続は拒否します。もし IP アドレスがホワイトリストであるなら、接続は受け入れます。

もし IP アドレスがホワイトリストであるなら、あなたは安全な通信チャネルがそのアドレスために要求するかどうか指定することができます。ほとんどの場合、ローカル接続が暗号化なしで走ることを可能にすることは問題ないでしょう。

FrontBase は同様にそれがただローカル接続だけを受け入れるモードで走ることができます。これは、WebObjects あるいは PHP で動いている Web サーバをバックアップして、部外者が直接データベースに接続することができないことを保証するとき、有用です。

トランザクション

89 ページの「トランザクションのログを書き出す」セクションは FrontBase のトランザクションサポートを記述します。

列レベルの特権

FrontBase はあなたに個々の列のアクセス権を指定することを可能にする、277 ページの呼び出されたユニークな特徴「列レベルの特権」を与えます。それぞれの列は特定のユーザ、そして特定のグループに所属しているユーザによって所有すると言われています。列のアクセス権（セレクトして、更新して、そして削除する）は、所有者、グループと世界について指定することができます。

生のバックアップ

108 ページの「バックアップと復帰」セクションはそれが生のデータベース（すなわちクライアントがデータベースにアクセスして、そして修正し続ける間）のバックアップを行うことを可能にする FrontBase のバージョン対応システムを記述します。

キャッシュすること

120 ページの「FrontBase のチューニング」セクションは FrontBase のスキームをキャッシュすることを記述します。

マルチサーバーのデプロイメント

95 ページの「複製」セクションは FrontBase の複製の特徴、そしてマルチサーバーのデプロイメントのために、冗長性と拡張した性能のために 103 ページの「クラスタリング」を記述します。

ドライバー、アダプタとプラグイン

次のことはアプリケーション開発について FrontBase を極めて自在に使うデータベースにするドライバーとアダプタの短い概要です。次のことのすべては関連づけられた文書に沿って www.frontbase.com で「ダウンロード」のセクションからダウンロードするためにアクセス可能です。

ODBC と JDBC

ODBC と JDBC ドライバはこれらの規格を当てにするアプリケーションから FrontBase まで一般的な接続性を提供します。ODBC ドライバーは WinNT/2000/ME/98 で使えます。それぞれのドライバー間で接続を確立するための詳細は、ファイルがドライバーにあるリードミーで見つけることができます。

WebObjects 5 プラグイン

WebObjects プラグインはすべてのサポートするオペレーティング・システムで FrontBase に接続を可能にします。Win32 と MacOS X のために個々のプラグインがあります。

PHP3 と PHP4 アダプタ

これらのアダプタのための API は MySQL のためにそれぞれのアダプタに基づいています。関数は MySQL アダプタの `mysql_` から始まり、それらは FrontBase アダプタの `fbsql_` から始まります。この類似の最も明白な長所は FrontBase への PHP アプリケーションをポートすることが比較的単純であるということです。PHP 4 ドライバーは `php4.0.6` (発表している) そして `php4.0.7-dev` (現在の dev バージョン) をサポートします。

perl アダプタ

perl アダプタは perl データベースインタフェース (DBI) です。この抽象レイヤは FrontBase に perl アプリケーションをポートすることを他のデータベースからかなり容易にするでしょう。

Omnis Studio DAM

Omnis Studio 3.x のために発表した DAM（データアクセスモジュール）は、我々のダウンロードページから利用可能です。この DAM は FrontBase での開発に豊富な RAD 環境をデベロッパーに提供します。DAM は MacOS クラシック（8 / 9）、MacOS X と Win32 のために現在利用可能です。文書はダウンロードパッケージに含まれています。

REALBasic

これらのプラグインは MacOS クラシック（8 / 9）で走っている REALbasic 3.0 PPC と MacOS X で走っている REALbasic 3.0 カーボンです。

Tcl ドライバ

FrontBase は今、ツールコマンド言語（Tcl）、言語の利用可能なクロスプラットフォームを書いている単純な、オープンソースで開発をサポートしています。

EOF アダプタ

EOF アダプタはアップルの WebObjects 4 . 5 を使って FrontBase を動かすことを可能にします。

移行

FrontBase は FileMaker と MySQL から読み込んでいるデータベースのツールを持っています。

FileMaker

1 3 2 ページの「FileMaker」の移行は、2つのステッププロセスです。FileMaker データベースの表は FileMaker から書き出されます。書き出されたファイルは MacOS X で動かすことができ、そしてそこでアプリケーションが FrontBase の中にそれらを読み込みます。

MySQL

1 3 3 ページの「MySQL」の移行ツールは、JDBC を MySQL データベースから表データを引き出して、そして FrontBase データベースの中にそれを読み込むために使います。

他の読み込みメカニズムは 1 1 1 ページでの拡張した読み込み機能「拡張した書式のないファイルの読み込みと書き出し関数」によって利用可能です。

FrontBase 関連のプロセス

FrontBase をインストールするために走る3つのメインプロセスがあります：FBExec、FrontBase と FBWebEnabler。このセクションはこれらのプロセス、それらが走っているかどうか決定する方法とそれらをスタートする方法を記述します。

FBExec

FBExec はあなたのコンピュータ上で走っている FrontBase データベースとあなたのコンピュータ上で走っているクライアントソフトウェアの間にあるいはネットワークに関してブローカーを動作します。

あなたが FrontBase をインストールするとき、FBExec がスタートアップに着手できるように、あなたのコンピュータは準備するでしょう。FBExec が UNIX ベースでインストールするために走っている

ことを確認するために、次のことをターミナルセッションに入力してください：

```
ps axc | grep FBExec
```

もし FBExec が走っているなら、システムはきつとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
```

もし FBExec が走っていないなら、あなたは次のことを始めてください（\$PATH に <install dir>/FrontBase/bin を加えたと仮定します）：

```
FBExec &
```

タスクがターミナルセッションで終了しないように、終わりにアンド記号（「&」）を含んでください。

もし、しかし、あなたが FrontBase を Windows NT 上で走らせているなら、あなたは FBExec が走っているかどうか調べるためにタスクマネージャーを使うことができます。もしそれが走っていないなら、サービスマネージャーのところに行って、そして始めてください。それがシステムスタートアップでスタートするように、FBExec はサービスとしてインストールします。

FrontBase

FrontBase の 1 つのインスタンスがあなたのコンピュータ上で走っているそれぞれのデータベースについて走っているでしょう。それぞれの FrontBase のインスタンスがサービスマネージャー（Windows NT）、あるいは FrontBaseManager と FBWebManager ツールを使うことによって、コマンドライン（UNIX をインストールすること）から、直接スタートします。

実行の間にサーバーのビヘビアに影響を与える FrontBase のためのコマンドラインオプションは、250 ページの「利用可能な起動オプション」で、記述しています。これらのオプションは同様に、FrontBase データベースを始めて、そして作るために、137 ページの「FrontBaseManager」と205 ページの「FBWebManager」のツールを使うとき、セットすることができます。

FBWebEnabler

FBWebEnabler プロセスはローカルにあるいはもう1つのコンピュータからすべての Web ブラウザで FrontBase サーバーでアドミニストレーションの関数を行うことを可能にしている FBWebManager システムについて持続的な接続を維持します。

あなたが FrontBase をインストールするとき、FBWebEnabler がスタートアップに着手できるように、あなたのコンピュータは準備するでしょう。もしそれが走るのをやめるなら、あなたは Web ブラウザを通して cgi-bin/FBWebManager にアクセスしようと試みるとき、多分発見するでしょう。205 ページの「FBWebManager」は、FBWebEnabler に接続することができないことを示すでしょう。

もし FBWebEnabler が走っていないなら、あなたは次のことを始めてください（\$PATH に <install dir>/FrontBase/bin を加えたと仮定します）：

FBWebEnabler &

タスクがターミナルセッションで終了しないように、終わりにアンド記号（「&」）を含んでください。

もし、しかし、あなたが FrontBase を Windows NT 上で走らせているなら、あなたはサービスマネージャーのところに行って、そして FBWebEnabler を始めることができます。

FBExec のように、それがシステムスタートアップでスタートするように、FBWebEnabler はサービスとしてインストールします。

プロセスの識別

MacOS 上のスタートアップ時間で FrontBase サーバープロセスの PID を決定するために、情報を提供しているファイルがあります。`.pid` ファイルはデータベースディレクトリで作ります：

`/Library/FrontBase/Databases/<database name>.fb.pid`

インストール

我々はそれぞれのサポートするサーバープラットフォームが違えば、FrontBase の違ったインストールを提供します。それぞれのインストールすることが FrontBase サーバーとサーバーアドミニストレーションツールとプラットフォームのために利用可能なクライアントライブラリが入っています。この章では次のセクションが入っています：

- 33 ページの「FrontBase をダウンロードする」。
- 34 ページの「FrontBase をインストールする」。
- 61 ページの「FrontBase のライセンス」。
- 63 ページの「FrontBase のディレクトリの構造」。
- 65 ページの「FrontBase を隠す」。
- 74 ページの「最新の状態を保つ」。

FrontBase をダウンロードする

次のプラットフォームをサポートします：

- 34 ページの「MacOS」。
- 37 ページの「Windows」。
- 39 ページの「Linux」。
- 55 ページの「Unix」。

FrontBase をインストールする

MacOS

次の管理ツールを提供します：

- ・ 137 ページの「FrontBaseManager」。
- ・ 205 ページの「FBWebManager」。
- ・ 235 ページの「sql92」。

次のクライアントライブラリが利用可能です：

- ・ FBAccess
- ・ FBCAccess
- ・ PHP3/4
- ・ Perl
- ・ ODBC
- ・ JDBC 2.0
- ・ EOF Adaptor

次のオペレーティング・システムをサポートします：

- ・ 34 ページの「MacOS X と MacOS X サーバー 10.x」。

MacOS X と MacOS X サーバー 10.x

下に説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたは新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

1. 「ルート」として MacOS X コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのようにならざるを得ません。それは特別なポートを必要としません、従って、セキュリティ懸念を起すようなことはありません。それはそうするためにすべてのユーザの準備で走らせることができます。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

- もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプトは自動的に FrontBase に関連したプロセスを止めるでしょう。もしあなたが手作業で止めることを望むなら、88 ページの「アドミニストレーションツール」を参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディズエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

- かつて、あなたは .dmg ファイルをダウンロードして、ブラウザのプリファレンスに従ってファイルの位置を示しました。 .dmg ファイルは .pkg ファイルにディスクユーティリティで拡張します

- .dmg ファイルをセレクトして、そしてダブルクリックしてください。 .pkg をインストールして、ファイルをセレクトして、そしてダブルクリックするために、そしてインストーラは FrontBase をインストールする準備ができたなら開始するでしょう。インストーラは詳細指示に従って提供するでしょう。

デフォルトで、インストーラは `/Library/ FrontBase` ディレクトリに FrontBase をインストールするでしょう。

インストーラはアパッチ Web サーバによって使う適切なディレクトリに FBWebManager ファイルをインストールしようと試みるでしょう。

- FrontBase は正常にインストールして、そして、次のことをターミナルウィンドウに入力することによって、スタートしたことを確認してください：

```
ps axc | grep FB
```

もし FBExec プロセス（重要な FrontBase の部品）と FBWebEnabler プロセス（Web アドミニストレーションツールプロセス）が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
375 ? S 0:00 FBWebEnabler
```

もし1つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /Local/Library/FrontBase
```

(FrontBase のデフォルトにインストールする場所)

```
./bin/FBExec &  
./bin/FBWebEnabler &
```

コンピュータを再起動するとき、FBExec が自動的にスタートしてインストールすることに注意してください。FBExec は同様にインストールのプロセスの一部としてスタートします、そして、インストールの後にコンピュータを再起動する必要がありません。もし FBExec あるいは FBWebEnabler を開始することがエラーをもたらすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役立つことがうれしいです。

7. もしあなたが FrontBase の前のバージョンからアップグレードしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア（例えば WebObjects、PHP など）があるなら、あなたは今それらを再起動してください。

最初に、88 ページの「アドミニストレーションツール」で説明しているようにあなたのデータベースがスタートします。その時、あなたのクライアントソフトウェアを再起動してください。

8. `/Library/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にしましょう。
9. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。

Windows

次の管理のツールを提供します：

- ・ 1 3 7 ページの「FrontBaseManager」。
- ・ 2 0 5 ページの「FBWebManager」。
- ・ 2 3 5 ページの「sql92」。

次のクライアントライブラリが利用可能です：

- ・ FBAccess
- ・ FBCAccess
- ・ PHP3/4
- ・ Perl
- ・ ODBC
- ・ JDBC 2.0
- ・ EOF Adaptor

次のオペレーティング・システムをサポートします：

- ・ 3 7 ページの「Windows 2 0 0 0 / N T」。

Windows 2 0 0 0 / N T

下に説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたは新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

1. 「管理者」として Windows NT コンピュータにログインしてください。FrontBase は現在管理者によってインストールする必要があります、その結果、IIS と類似のサービスのように入ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起すようなことはありません。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。
アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。
3. ダウンロードしたアーカイブを圧縮解除するために（PKZip のような）あなたのお気に入りの zip ファイルユーティリティを使ってください。圧縮解除することは 3 つの実行可能な(.exe)

ファイルを与えるでしょう。それらのそれぞれを走らせて、そして提供されたインストールの説明に従ってください。同様にインクルードしたものは Windows 2 0 0 0 / N T の FrontBase と共に WebObjects 5 を使っているそれらのために要求するさらに実行可能な WebObjects プラグインです。

FrontBase は `<drive>:/usr/FrontBase` ディレクトリにインストールされるでしょう、そして `<drive>:` はあなたが FrontBase をインストールしたドライブです。

Windows NT の特定の部品が `<drive>:/Program Files/FrontBase Tools` と `<drive>:/Apple/Library/Frameworks` ディレクトリにインストールされるでしょう。

4. デフォルトで、FrontBase は C : ドライブにインストールすると思います。もしあなたがもう1つのドライブ (例えば F :) にそれをインストールしたなら、あなたは NT にシステム環境変数を加える必要があるでしょう。

スタート -> セッティング -> コントロールパネルに行って、システムアイコンをダブルクリックして、そして、FrontBase (例えば F :) をインストールしたドライブの文字にセットして、`FB_HOME_DRIVE` 環境変数に加えてください。

5. インストールするプロセスは自動的に NT サービスとして FBExec と FBWebEnabler を定義して、そしてそれらをスタートするでしょう。しかし、あなたはそれが比較的単純である若干の理由のために今までに手作業でこのタスクを行う必要がありますか。例えば、FBExec をインストールして、そしてスタートするために、あなたは FBExec を NT サービスと定義する必要がありますでしょう。シェル (例えば Bourne あるいは DOS シェル) を取り出して、そして次のコマンドを入力してください:

```
<drive>:/usr/FrontBase/bin/FBExec -install
```

サービスコントロールマネージャーを使って FBExec サービスをスタートしてください (スタート -> セッティング -> コントロールパネル、サービスアイコンをダブルクリックしてください)。サービスコントロールマネージャーを使って、あなたは同様に、コンピュータが再起動するときはいつでも、FBExec サービスが自動的にスタートするように指定することができます。

「Windows NT タスクマネージャー」(Ctrl-Alt-Del、タスクマネージャーをクリック) を開始することによって、あなたは FBExec が今走っていることを確認することができます。

6. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました! あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすでしょう。

Linux

次の管理のツールを提供します：

- ・ 205 ページの「FBWebManager」。
- ・ 235 ページの「sql92」。

次のクライアントライブラリが利用可能です：

- ・ FBCAccess
- ・ PHP3/4
- ・ Perl
- ・ ODBC
- ・ JDBC 2.0

次のオペレーティング・システムをサポートします：

- ・ 40 ページの「RedHat 7.x Linux (x86)」。
- ・ 43 ページの「SuSE 7.x Linux (x86)」。
- ・ 46 ページの「YellowDog Linux (PPC)」。
- ・ 49 ページの「Debian Linux (x86)」。
- ・ 52 ページの「Mandrake Linux (x86)」。
- ・ もしあなたが IBM S390 のための FrontBase for Linux に興味を持っているなら、どうか info@frontbase.com で我々と連絡を取ってください。

RedHat 7.x Linux (x86)

下に説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたは新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注：RedHat ベースの Linux は RPM で簡単にインストールできます。

1. 「ルート」として RedHat Linux コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのように走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起こすようなことはありません。

2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

3. もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業で止めることを望むなら、基本的なアドミニストレーションを参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディスエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

4. ターミナルから、FrontBase を拡張して、そして次のようにインストールするスクリプトを走らせてください（実際のバージョン番号がより高いかもしれないことに注意してください）。もしあなたが初めて FrontBase をインストールしているなら：

```
rpm -i FrontBase-<version-number>.rpm
```

もしあなたが FrontBase をアップデートしているなら、使ってください：

```
rpm -U FrontBase-<version-number>.rpm
```

— U に加えて、— どのようなエラーでも無視してインストールを続けるオプションを使うことに注意してください。

デフォルトで、スクリプトは `/usr/local/FrontBase/bin` ディレクトリに FrontBase をインストールするでしょう。スクリプトは関連したアパッチディレクトリに FBWebManager の関連したファイルをインストールしようと試みるでしょう。

5. FrontBase が正常にインストールして、そして、次のことをターミナルウインドウに入力することによって、スタートしたことを確認してください：

```
ps axc | grep FB
```

もし FBExec プロセス (重要な FrontBase の部品) と FBWebEnabler プロセス (Web アドミニストレーションツールプロセス) が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
375 ? S 0:00 FBWebEnabler
```

もし 1 つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /usr/local/FrontBase
```

(FrontBase のデフォルトでインストールする場所)

```
./bin/FBExec &
./bin/FBWebEnabler &
```

もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。

6. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア (例えば WebObjects、PHP など) があるなら、今それらを再起動してください。

最初に、88 ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

7. `/usr/local/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にするでしょう。
8. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすでしょう。

SuSE 7.x Linux (x86)

下の説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたは新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注： SuSE Linux は rpm で簡単にインストールできます。

1. 「ルート」としてあなたの SuSE Linux コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのようには走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起こすようなことはありません。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。
アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。
3. もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業で止めることを望むなら、基本的なアドミニストレーションを参照してください。
もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディスエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれない！ あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。
4. ターミナルから、FrontBase を拡張して、そして次のようにインストールするスクリプトを走らせてください（実際のバージョン番号がより高いかもしれないことに注意してください）。もしあなたが初めて FrontBase をインストールしているなら：

```
rpm -i FrontBase-<version-number>.rpm
```

もしあなたが FrontBase をアップデートしているなら、使ってください：

```
rpm -U FrontBase-<version-number>.rpm
```

— U に加えて、— どのようなエラーでも無視してインストールを続けるオプションを使うことに注意してください。

デフォルトで、スクリプトは `/opt/FrontBase` ディレクトリに FrontBase をインストールするでしょう。スクリプトは関連したアパッチディレクトリに FBWebManager の関連したファイルをインストールしようと試みるでしょう。

5. FrontBase が正常にインストールして、そして、次のことをターミナルウィンドウに入力することで、スタートしたことを確認してください：

```
ps axc | grep FB
```

もし FBExec プロセス (重要な FrontBase の部品) と FBWebEnabler プロセス (Web アドミニストレーションツールプロセス) が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
375 ? S 0:00 FBWebEnabler
```

もし 1 つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /opt/FrontBase
```

(FrontBase のデフォルトでインストールする場所)

```
./bin/FBExec &
./bin/FBWebEnabler &
```

もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。

6. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア (例えば WebObjects、PHP など) があるなら、今それらを再起動してください。

最初に、88 ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

7. `/opt/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にするでしょう。

8. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすでしょう。

YellowDog Linux (PPC)

下の説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたが新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注：YellowDog Linux は rpm で簡単にインストールできます。

1. 「ルート」としてあなたの YellowDog Linux コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのように走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起すようなことはありません。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

3. もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業で止めることを望むなら、88 ページの「アドミニストレーションツール」を参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディスエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

4. ターミナルから、FrontBase を拡張して、そして次のようにインストールするスクリプトを走らせてください（実際のバージョン番号がより高いかもしれないことに注意してください）。もしあなたが初めて FrontBase をインストールしているなら：

```
rpm -i FrontBase-<version-number>.rpm
```

もしあなたが FrontBase をアップデートしているなら、使ってください：

```
rpm -U FrontBase-<version-number>.rpm
```

— U に加えて、— どんなエラーでも無視してインストールを続けるオプションを使うことに注意してください。

デフォルトで、スクリプトは `/opt/FrontBase` ディレクトリに FrontBase をインストールするでしょう。スクリプトは関連したアパッチディレクトリに FBWebManager の関連したファイルをインストールしようと試みるでしょう。

5. FrontBase が正常にインストールして、そして、次のことをターミナルウインドウに入力することによって、スタートしたことを確認してください：

```
ps axc | grep FB
```

もし FBExec プロセス（重要な FrontBase の部品）と FBWebEnabler プロセス（Web アドミニストレーションツールプロセス）が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
375 ? S 0:00 FBWebEnabler
```

もし1つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /opt/FrontBase
```

（FrontBase のデフォルトでインストールする場所）

```
./bin/FBExec &
./bin/FBWebEnabler &
```

もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、どうか support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。

6. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア（例えば WebObjects、PHP など）があるなら、今それらを再起動してください。

最初に、88ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

7. `/opt/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にするでしょう。
8. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすでしょう。

Debian Linux (x86)

下の説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたが新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注：シェルスクリプトで Debian Linux をインストールできます。

1. 「ルート」としてあなたの Debian Linux コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのように走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起すようなことはありません。

2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

.deb アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

3. もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めることを望むかどうか尋ねるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業で止めることを望むなら、88 ページの「アドミニストレーションツール」を参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディスエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

4. あなたが .deb ファイルをダウンロードした途端に、次のようにそれ（ターミナルウィンドウがルートとしてログインした）をインストールしてください：

```
dpkg -i FrontBase-3.3.deb
```

FBWebManager の実行可能なプログラムはアパッチインストールの cgi-bin ディレクトリ（/usr/lib/cgi-bin）にインストールするために試みます。FBWebManager によって必要とするイメージと他のファイルは /var/www/FBWebManager にインストールします。もしあなたのコンピュータのアパッチ

が /usr/lib/cgibin と /var/www にインストールしないなら、どうか適切な位置に FBWebManager と関連づけたファイルをコピーして / 移動してください。

5. FrontBase が正常にインストールして、そして、次のことをターミナルウインドウに入力することによって、スタートしたことを確認してください：

```
ps axc | grep FB
```

もし FBExec プロセス (重要な FrontBase の部品) と FBWebEnabler プロセス (Web アドミニストレーションツールプロセス) が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
375 ? S 0:00 FBWebEnabler
```

もし 1 つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd / usr / lib / FrontBase
```

(FrontBase のデフォルトでインストールする場所)

```
./bin/FBExec &
./bin/FBWebEnabler &
```

コンピュータがブートするとき、FBExec が自動的にスタートするためにインストールすることに注意してください。FBExec は同様にインストールするプロセスの一部としてスタートします、すなわちインストールの後にコンピュータを再起動する必要がありません。

もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。

6. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア (例えば WebObjects、PHP など) があるなら、今それらを再起動してください。

最初に、88 ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

7. `/usr/lib/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にしますでしょう。
8. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすでしょう。

Mandrake Linux (x86)

下の説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたが新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注：Mandrake Linux は RPM で簡単にインストールできます。

1. 「ルート」としてあなたの Mandrake Linux コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのように走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起こすようなことはありません。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

3. もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業でことを止めることを望むなら、基本的なアドミニストレーションを参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディスエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

4. ターミナルから、FrontBase を拡張して、そして次のようにインストールするスクリプトを走らせてください（実際のバージョン番号がより高いかもしれないことに注意してください）。もしあなたが初めて FrontBase をインストールしているなら：

```
rpm -i FrontBase-<version-number>.rpm
```

もしあなたが FrontBase をアップデートするなら、使ってください：

```
rpm -U FrontBase-<version-number>.rpm
```

— U に加えて、— **ど**んなエラーでも無視してインストールを続けるオプションを使うことに注意してください。

デフォルトで、スクリプトは `/usr/local/FrontBase` ディレクトリに FrontBase をインストールするでしょう。スクリプトは関連したアパッチディレクトリに FBWebManager の関連したファイルをインストールしようと試みるでしょう。

5. FrontBase が正常にインストールして、そして、次のことをターミナルウインドウに入力することによって、スタートしたことを確認してください：

```
ps axc | grep FB
```

もし FBExec プロセス (重要な FrontBase の部品) と FBWebEnabler プロセス (Web アドミニストレーションツールプロセス) が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec
375 ? S 0:00 FBWebEnabler
```

もし 1 つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /usr/local/FrontBase
```

(FrontBase のデフォルトでインストールする場所)

```
./bin/FBExec &
./bin/FBWebEnabler &
```

もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。

6. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア (例えば WebObjects、PHP など) があるなら、今それらを再起動してください。

最初に、88ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

7. `/usr/local/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にするでしょう。
8. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすでしょう。

Unix

次の管理ツールを提供します：

- ・ 205 ページの「FBWebManager」。
- ・ 235 ページの「sql92」。

次のクライアントライブラリが利用可能です：

- ・ FBCAccess
- ・ PHP3/4
- ・ Perl
- ・ ODBC
- ・ JDBC 2.0

次のオペレーティング・システムをサポートします：

- ・ 55 ページの「Solaris」。
- ・ 58 ページの「FreeBSD (x86)」。

Solaris

下の説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたは新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注：FBManager は Solaris ですぐ利用できません。Solaris の FBAccess と EOF アダプタはアップルの WebObjects にインストールするように要求します。FrontBase for Solaris は SunOS 5.8 を使って作られました。他のバージョンについては、info@frontbase.com で我々と連絡を取ってください

1. 「ルート」としてあなたのソラリスコンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのように走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起こすようなことはありません。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

- もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めることを望むかどうか尋ねるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業でことを止めることを望むなら、88 ページの「アドミニストレーションツール」を参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディセーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

- ターミナルから、FrontBase を拡張して、そして次のようにインストールするスクリプトを走らせてください（実際のバージョン番号がより高いかもしれないことに注意してください）。

```
cd FrontBase-<version-number>  
sh ./install.sh
```

- デフォルトで、スクリプトは `/opt/FrontBase` ディレクトリに FrontBase をインストールするでしょう。スクリプトは関連したアパッチディレクトリに FBWebManager の関連したファイルをインストールしようと試みるでしょう。
- FrontBase が正常にインストールして、そして、次のことをターミナルウインドウに入力することによって、スタートしたことを確認してください：

```
ps - e | grep FB
```

もし FBExec プロセス（重要な FrontBase の部品）と FBWebEnabler プロセス（Web アドミニストレーションツールプロセス）が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec  
375 ? S 0:00 FBWebEnabler
```

もし 1 つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /opt/FrontBase
```

(FrontBase のデフォルトでインストールする場所)

```
./bin/FBExec &  
./bin/FBWebEnabler &
```

もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。

7. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア (例えば WebObjects、PHP など) があるなら、今それらを再起動してください。

最初に、88 ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

8. `/opt/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にするでしょう。

おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすとよいでしょう。

FreeBSD (x86)

下の説明に従って、あなたは数分で FrontBase をインストールすることができるでしょう。FrontBase をインストールした後に、あなたは新しい FrontBase サーバーについて読むことができます。続ける前に、あなたはこのセクションを印刷することが役に立つことに気付くでしょう。

注：FreeBSD の場合、シェルスクリプトを使ってインストールして、そして多少手作業で設定をしなければなりません。

注：FrontBase for FreeBSD は、FreeBSD 4.2 を使って作られました

1. 「ルート」としてあなたの FreeBSD コンピュータにログインしてください。FrontBase は現在ルートによってインストールする必要があります、その結果、アパッチと他のルートレベルサービスのように走ることができます。それは特別なポートを必要としません、従って、セキュリティ懸念を起こすようなことはありません。
2. FrontBase をダウンロードするために、www.frontbase.com に行ってください。

アーカイブは 5 MB 以下の大きさで、そして 56K モデムでダウンロードするのに数分かかります。

3. もしあなたがサーバー上で走っている FrontBase の前のバージョンを持っているなら、インストールスクリプト（次のステップで走る）は FrontBase に関連したプロセスを止めることを望むかどうか尋ねるでしょう。あなたは通常スクリプトにこれらのプロセスを止めさせるでしょう。もしあなたが手作業でことを止めることを望むなら、88 ページの「アドミニストレーションツール」を参照してください。

もしあなたが同様に（WebObjects、PHP などのような）クライアントソフトウェアを走らせているなら、それがアップグレードプロセスの間に FrontBase にアクセスしないように、あなたはそれをディスエーブルしておいてください。若干のクライアントソフトウェアは FrontBase が走っていないことを検出して、そして、あなたがアップグレードしているのに、それを再開しようと試みるかもしれません！あなたがアップグレードする間に、インストールは数分間オフラインにしておいてください。

4. ターミナルから、FrontBase を拡張して、そして次のようにインストールするスクリプトを走らせてください（実際のバージョン番号がより高いかもしれないことに注意してください）。

```
tar xvf FrontBase-<version-number>.tar  
cd FrontBase-<version-number>  
sh install.sh
```

5. デフォルトで、スクリプトは `/usr/local/FrontBase` ディレクトリに FrontBase をインストールするでしょう。スクリプトは関連したアパッチディレクトリに FBWebManager の関連したファイルをインストールしようと試みるでしょう。
6. FrontBase が正常にインストールして、そして、次のことをターミナルウインドウに入力することによって、スタートしたことを確認してください：

```
ps aux | grep FB
```

もし FBExec プロセス（重要な FrontBase の部品）と FBWebEnabler プロセス（Web アドミニストレーションツールプロセス）が走っているなら、システムはきっとこんなふうに答えるでしょう：

```
374 ? S 0:00 FBExec  
375 ? S 0:00 FBWebEnabler
```

7. もし1つあるいは両方ともが走っていないなら、コマンドラインからそれらを開始しようとしてください：

```
cd /usr/local/FrontBase
```

（FrontBase のデフォルトでインストールする場所）

```
./bin/FBExec &  
./bin/FBWebEnabler &
```

- もし FBExec あるいは FBWebEnabler を開始することでエラーを起こすなら、support@FrontBase.com に電子メールを送ってください。我々はあなたの役に立つことがうれしいです。
8. もしあなたが FrontBase の前のバージョンからアップグレードしようとしていて、そして、アップグレードを始める前に、走っているデータベースあるいはクライアントソフトウェア

(例えば WebObjects、PHP など) があるなら、今それらを再起動してください。

最初に、88ページの「アドミニストレーションツール」で説明するようにデータベースがスタートします。その時、クライアントソフトウェアを再起動してください。

9. `/usr/local/FrontBase/bin` ディレクトリに含む検索パスを調整すると、コマンドラインを使うこともなく、あなたを楽にするでしょう。
10. おめでとうございます、あなたは正常にダウンロードして、そして FrontBase をインストールしました！ あなたはコンピュータを再起動する必要がありません。あなたは今 FrontBase サーバーに伴う文書を読んでしばらく過ごすといでしょう。

FrontBase ライセンス

ライセンスを得るには

IP あるいは Mac (イーサネット) アドレスとサーバーのプラットフォームを登録して、そして供給することによって、あなたは FrontBase.com Web サイトからライセンスの文字列を得ることができます。あなたのサーバーに「ライセンスの文字列」と「ライセンスのチェック」が電子メールで送られるでしょう。

すぐ利用できるライセンス

E-Starter

セキュリティ・チェックと列レベルの特権を使う無期限のライセンス。あなたはデプロイメントのためにライセンスを使ってもよいです、それはリモートアクセスを可能にしますが、バックアップについてはサポートされません。

E-Business

セキュリティ・チェック、書き出し / 読み込み、表をキャッシュすること、バックアップ、未使用のデバイスドライバ、暗号、複製クライアント。これは我々が定期的なバックアップが必要である重大な環境のデプロイメントについて勧める最小のライセンスです。

E-Enterprise

セキュリティ・チェック、書き出し / 読み込み、表をキャッシュすること、バックアップ、未使用のデバイスドライバ、暗号化、複製クライアント、複製マスター、クラスタリング、クラスタモニター。複製 / クラスタデザインと設定アシスタンスを含みます。

Developer

E-Enterprise ライセンス - デプロイメントの権利がありません。6カ月の満期で、6カ月ごとに更新することができます。

Embedded

インターネットサービスプロバイダや OEM を希望する会社がそれらの売り物に FrontBase を法人登録するために。完全な詳細は 284 ページの「あなた自身のアプリケーションあるいは売り物の中に FrontBase を組み込む」セクションの中にあります。

無料のライセンスを得るには

FrontBase.com Web サイトから、'Buy' セクションにそして次に 'License' にナビゲートしてください。ここで、あなたは FrontBase の無料の 'E-Starter' ライセンスのリンクを見つけるでしょう。

注：無料のライセンスはバックアップと復帰をサポートしません。

あなたはこのリンクを通して同様に無料のデベロッパーライセンス (デプロイメントの権利がない "Enterprise" ライセンス) を申し込むことができます。

ライセンスを購入するには

FrontBase.com Web サイトから、'Buy' セクションにそして次に 'License' にナビゲートしてください。ここで、あなたは FrontBase で利用可能であるさまざまな商用ライセンスのリンクを見つけるでしょう。

ライセンスをインストールする方法

ライセンス文字列をインストールするには 3 つの方法があります：

1. もしあなたが FrontBaseManager を含むプラットフォームで FrontBase を走らせているなら、あなたはそこからそれをインストールすることができます。ツールメニューからライセンス管理を選択して、サーバーをセレクトして、そしてあなたに電子メールで送られたライセンス文字列とライセンスチェックをペーストしてください。
2. もしあなたがサーバー上で走っている FBWebManager を持っているなら、あなたの Web ブラウザを使ってそれに接続して、そして左の行のライセンスをクリックしてください。あなたに電子メールで送られたライセンス文字列とライセンスチェックをペーストしてください。
3. 大いに勧められないけれども、あなたはコマンドラインからテキストエディタあるいは、例えば、`cat>LicenseString` を使ってライセンスファイル自身を作ることができます。ファイルは `LicenseString` と呼ばれ、そして主に FrontBase ディレクトリに存在します。そのフォーマットは次の通りです：

```
<64 char license string>:<16 char license check>
```

FrontBase のディレクトリの構造

FrontBase のディレクトリは FrontBase をインストールするために必要としたすべてのファイルを含みます。FrontBase のディレクトリのファイルとサブディレクトリはこのセクションで説明します。

ディレクトリの場所

FrontBase のディレクトリの場所は FrontBase をインストールするプラットフォームに依存します

プラットフォーム	パス
3 4 ページの「MacOS」。	/Library/FrontBase
3 7 ページの「Windows」。	<drive>:/usr/local/FrontBase
4 0 ページの「RedHat 7.x Linux (x86)」。	/usr/local/FrontBase
4 3 ページの「SuSE 7.x Linux (x86)」。	/opt/FrontBase
4 6 ページの「YellowDog Linux (PPC)」。	/opt/FrontBase
4 9 ページの「Debian Linux (x86)」。	/usr/lib/FrontBase
5 2 ページの「Mandrake Linux (x86)」。	/usr/local/FrontBase
5 5 ページの「Solaris」。	/opt/FrontBase
5 8 ページの「FreeBSD (x86)」。	/usr/local/FrontBase

FrontBase の将来のバージョンとそのプラットフォーム特定のインストーラがどこでも FrontBase をインストールできるようになるでしょう。現在、FrontBase ディレクトリは「ルート」あるいは「管理者」によって所有する必要があります、しかしそのたいてい不自然な要求は将来のバージョンで同様に姿を消すでしょう。

ディレクトリの内容

FrontBase ディレクトリは通常次のサブディレクトリとファイルを含むでしょう：

1. データベースはホストからサポートしたすべてのデータベースを保つディレクトリです。それぞれのデータベースのために、2

つのファイルがあります : < データベース名 >.fb と < データベース名 >.fb.log 。前者はデータベースの実際のデータを含んでいるのに対して、後者はその他の状態とエラーメッセージを含んでいます。< データベース名 >.fb ファイルを削除することによって、データベースを削除することができます。もちろん、あなたは同様に `_SYSTEM` としてデータベースにログインして、そしてデリートデータベースコマンドを出すことができます。Windows NT で、あなたは、それを削除する前に、あなたがサービスとしてデータベースを隠したことを確認するべきです。

2. **照合 (Collations)** は、すべての定義した照合を保つディレクトリです (ユニコード文字の順番をスキーマのコレクションとして例示することができる)。
3. **トランスレーション** はすべての定義した変換を保つディレクトリです。
4. **bin** は FrontBase の実行可能なプログラムを含むディレクトリです。あなたは簡単にアクセスするために \$PATH 環境変数に FrontBase/bin ディレクトリを加えることを望むでしょう。このドキュメントのすべての事例はあなたがそうしたと想定します。
5. **FBExec.log** は FBExec プロセスがその他の状態とエラーメッセージを付加するファイルです。
6. **FBWebEnablerLog** は FBWebEnabler プロセスがその他の状態とエラーメッセージを付加するファイルです。
7. **ライブラリ** は新しいデータベースを作成することを必要とするその他のファイルを含むディレクトリです。
8. **バックアップ** はバックアップ・ファイルに入っているディレクトリです。
9. **TransactionLogs** は個々のサブディレクトリのすべてのトランザクションログに入っています。
10. **インクルード** は開発のために使われたヘッダーファイルに入っています。
11. **lib** は開発のために使われた FrontBase ライブラリに入っています。
12. **テンプレート** は FBWebManager の html テンプレートに入っています。

FrontBase を取り除く

もしあなたが今までに完全に FrontBase をコンピュータから取り除く必要があったなら、このセクションはどのようにそれをしたらよいかを記述します。

FrontBase インストールを取り外す、あるいはアンインストールすることはプラットフォームに非常に依存しています、そのため、下にそれぞれのサポートするプラットフォームに必要な工程を説明します。

経験的方法は：

- rpm パッケージは rpm コマンドを使うことによって取り除く事が出来ます。
- deb パッケージが dpkg コマンドを使うことによって取り除く事が出来ます。
- FrontBase インストールディレクトリの中の deinstall.sh コマンドを使うことによって、tar パッケージを取り除く事が出来ます。

Mac OS X と MacOS X サーバー 10.x

1. ルートとしてログインしてください (* root でログインは出来ないなので sudo コマンドを使う)
2. すべての FrontBase プロセスと FBExec プロセスを止めてください
3. 除去コマンド (rm) を使って FrontBase インストールを取り去ります (あるいはファインダを使います):

```
rm -r /Applications/FrontBaseManager.app  
rm -r /Applications/FBUnicodeManager.app  
rm -r /Library/Frameworks/FBAccess.framework  
rm -r /Library/Frameworks/FrontBaseEOAdaptor.framework  
rm /Library/StartupItems/FrontBase*  
rm -r /Library/WebServer/Documents/FBWebManager  
rm -r /Library/CGI-Executables/FBWebManager  
rm -r /Library/FrontBase
```

注：最後のステップ 3 の rm はあなたのすべてのデータベースを削除して、それが取り消し出来ないという事に注意してください。

Windows NT / 2 0 0 0

1. アドミニストレーターとしてログインしてください
2. ServiceControlManager アプリケーションを使って FBExec と FBWebEnabler サービスを止めます。
3. FBExec サービスを取り除きます：

```
<drive>:\usr\FrontBase\bin\FBExec -remove
```

4. FBWebEnabler サービスを取り除きます：

```
<drive>:\usr\FrontBase\bin\FBWebEnabler -remove
```

5. すべての FrontBase データベースサービスを取り除きます：

```
<drive>:\usr\FrontBase\bin\FrontBase -remove <database name>
```

6. サービスとしてインストールしたそれぞれのデータベースのためにステップ 4 を繰り返してください（インストール済みのデータベースのリストは Service-ControlManager アプリケーションを使って見つけます）。
7. （例えば Windows NT エクスプローラを使って）次のフォルダーを取り除きます：

```
<drive>:\Apple\Local\Library\Frameworks\FBAccess.framework  
<drive>:\Apple\Local\Library\Frameworks\FrontbaseEOAdaptor.framework  
<drive>:\Apple\Local\Library  
<drive>:\Apple\Local\Library\Executables\FrontbaseEOAdaptor.dll  
<drive>:\Program Files\FrontBaseTools  
<drive>:\usr\FrontBase
```

注：プログラムを加えたり取り除いたりするということはインストールされた FrontBase 構成要素、サーバー、フレームワークとツールのエントリを取り除くことも同時に行います。ステップ 6 の最後のフォルダーを削除することによって、あなたのすべてのデータベースを取り外したことになります。

Solaris

1. ルートとしてログインしてください
2. ディレクトリーを FrontBase インストールディレクトリに変えます。

```
cd /opt/FrontBase
```

3. FrontBase インストールを取り外します。

```
sh ./deinstall
```

4. ペアレントディレクトリに行きます。

```
cd ..
```

5. インストールディレクトリの残りを取り除きます：

```
rm -r FrontBase
```

注：ステップ3の最後のフォルダーを削除することによって、あなたのすべてのデータベースを取り外したことになります。

FreeBSD

1. ルートとしてログインしてください
2. ディレクトリーを FrontBase インストールディレクトリに変えます。

```
cd /opt/FrontBase
```

3. thr FrontBase インストールを取り除きます。

```
sh ./deinstall
```

4. ペアレントディレクトリに行きます。

```
cd ..
```

5. インストールディレクトリの残りを取り除きます：

```
rm -r FrontBase
```

注：ステップ3の最後のフォルダーを削除することによって、あなたのすべてのデータベースを取り外したことになります。

RedHat

1. ルートとしてログインしてください
2. FrontBase インストールを取り除きます

```
rpm -e FrontBase
```

3. インストールディレクトリの残りを取り除きます :

```
rm -r /usr/local/FrontBase
```

Mandrake

1. ルートとしてログインしてください
2. FrontBase インストールを取り除きます

```
rpm -e FrontBase
```

3. インストールディレクトリの残りを取り除きます :

```
rm -r /usr/local/FrontBase
```

SuSE

1. ルートとしてログインしてください
2. FrontBase インストールを取り除きます

```
rpm -e FrontBase
```

3. インストールディレクトリの残りを取り除きます :

```
rm -r /opt/FrontBase
```

Debian

1. ルートとしてログインしてください
2. FrontBase インストールを取り除きます

```
dpkg -r frontbase
```

3. インストールディレクトリの残りを取り除きます：

```
rm -r /usr/lib/FrontBase
```

注：ステップ3の最後のフォルダーを削除することによって、あなたのすべてのデータベースを取り外したことになります。

最新の状態を保つ

FrontBase は絶えず改善されています。もしあなたのオペレーティング・システムあるいはコンピュータを使って、CD-ROM からの FrontBase を得るか、あるいはまとめるなら、あなたはたいていの場合最新のバージョン（あるいは最新のドキュメンテーションを読み込む！）を走らせません。

最新のバージョンを決定する

sql92 を使う

FrontBase データベースへの接続を確立して、それから次の SQL ステートメントを実行してください：

```
VALUES(SERVER_NAME);
```

データベースを作るために使う FrontBase のバージョンと同様、現在走っている FrontBase のバージョンを返すでしょう。

FBWebManager

FrontBase データベースへの接続を確立して、それから次の SQL ステートメントを実行してください：

```
VALUES(SERVER_NAME);
```

データベースを作るために使う FrontBase のバージョンと同様、現在走っている FrontBase のバージョンを返すでしょう。

FrontBaseManager

(MacOS X と Windows NT のみ) FrontBase データベースへの接続を確立して、それから次の SQL ステートメントを実行してください：

```
VALUES(SERVER_NAME);
```

データベースを作るために使う FrontBase のバージョンと同様、現在走っている FrontBase のバージョンを返すでしょう。

コマンドライン / ターミナル

同様に `-v` 起動オプションを使って FrontBase をスタートすることは、使用中の FrontBase のバージョンを詳述します：

```
FrontBase -v <database-name>
```

あなたの FrontBase サーバーをアップグレードする

あなたがアップグレードする必要があるとき、33 ページの「インストール」セクションに進んでください。あなたのプラットフォームを選択して、そして FrontBase ソフトウェアをダウンロードしてください。

注：もしあなたが現在 FrontBase の前のバージョンをインストールして持っていて、そしてサーバー上で走っているなら、アップグレードの説明に慎重な注意を払ってください！

基本概念

この章は次のセクションに分けられています：

- － 77 ページの「SQL 92 の概念」。
- － 83 ページの「トランザクションを理解する」。

SQL 92 の概念

SQL 92 は ANSI / ISO 本体からの SQL の最新の公式の規格です、そしてそれは言語 SQL と種々の実装で何年もの経験からできた汗と涙の結晶です。

FrontBase は SQL 92 のほとんどすべてを実装する最初の産業の強力なデータベースサーバーです。これはあなたがデータベースサーバーと一緒にすることを望む仕事では重要でないでしょう、しかし少なくともあなたが知る必要がある 2 つの問題があります。多少混乱をまねくおそれがあるような大部分の概念はスキーマの概念です。市場に出ている多くの製品がそれらの文献の中の言葉のスキーマを使うのに対して、ほとんど（少なくとも SQL 92 の感覚で）この概念を実装しません。

実際に、カタログと呼ばれるスキーマにレイヤがあります。SQL 92 データベースが、それぞれスキーマの数を保って、カタログの数を構成します。スキーマがオブジェクトの数をコンテナとして見ます：表、ビュー、ドメイン、コレーションなど。

このセクションの見出しは次の通りです：

- ・ 78 ページの「カタログ」。
- ・ 78 ページの「スキーマ」。
- ・ 80 ページの「ユーザ」。
- ・ 81 ページの「日付、時間と時刻表示」。

- ・ 8 2 ページの「キーワードと識別子」。
- ・ 8 2 ページの「もっと多くの SQL 9 2 についての知識」。

カタログ

現在、FrontBase はデータベースの 1 つのカタログにサポートを提供します。このカタログはデータベースの名前を継承します、例えばもしデータベースが Movies.fb という名前を使って作るなら、カタログは Movies と名付けます。このカタログ名は常にデフォルトのカタログとして使います、従ってあなたは本当にカタログのことで心配する必要がありません。

スキーマ

前に述べたとおり、カタログが多くスキーマに入ることができます。スキーマがユーザによって所有します（さらに下を見てください）、そしてこのユーザだけがスキーマでのオブジェクトを加えるか、あるいはドロップすることができます。スキーマオブジェクトは表、ビュー、ドメイン、を使ってできます・・・。GRANT / REVOKE 文を使って、他のユーザが、スキーマ、例えばテーブルの挿入の特権で、オブジェクトに関して特権の番号を与えることができます。

スキーマでのオブジェクトがいわゆる制限された名前によって参照することができます：

```
[ [<catalog name>.] <schema name>.] <object name>
```

SQL 9 2 は「すべてをデフォルトにする」準備を提供します、それであるあなたが、現在のスキーマのリファレンスオブジェクトに、ただ<オブジェクト名>だけを望むときはいつでも与える必要があります。

新しいデータベースを作成するとき、2 つのスキーマを同様に作成します： DEFINITION_SCHEMA と INFORMATION_SCHEMA 。 DEFINITION_SCHEMA はすべての他のスキーマを維持するために使ったオブジェクトのすべてを維持します。

INFORMATION_SCHEMA はさまざまなオブジェクトを維持します、そしてそれは DEFINITION_SCHEMA のオブジェクトと便宜オブジェクトの数へのアクセスを提供します。例えばもしあなたはどの表がデータベースで定義したか見ることを望むなら、次の SQL 9 2 文を実行してください：

```
SELECT * FROM INFORMATION_SCHEMA.TABLES;
```

INFORMATION_SCHEMA.TABLES は実際に次のように定義したビューです：

```
CREATE VIEW INFORMATION_SCHEMA.TABLES AS  
SELECT * FROM DEFINITION_SCHEMA.TABLES;
```

INFORMATION_SCHEMA のビューはすべてアップデートしません、すなわち挿入のような：

```
INSERT INTO INFORMATION_SCHEMA.TABLES ....will fail.
```

DEFINITION_SCHEMA は独占的に FrontBase によって維持して、そしてユーザによってアクセスする、あるいは直接操ることはできません。

(現行のユーザがその時所有するであろう) 新しいスキーマを作成するために：

```
CREATE SCHEMA <schema name>;
```

注：事例がごく小さい断片の完全な文法のために SQL 9 2 の規格を見てください。

スキーマを現在のスキーマにするために：

```
SET SCHEMA '<schema name>;'
```

注：(文字列を使ってスキーマ名を与えることに注意してください)。

定義したスキーマのリストを見るために：

```
SELECT * FROM INFORMATION_SCHEMA.SCHEMATA;
```

現在のスキーマが何であるか見るために、CURRENT_SCHEMA の文字列の関数は利用可能です。CURRENT_SCHEMA が SQL 9 2 への FrontBase 拡張であることに注意してください。

ユーザ

SQL 9 2 のユーザの概念は比較的単純です、しかしとても密接にスキーマの概念と結び付けられます。

データベースにアクセスするために、ユーザ名を必要とします、そうしなければアクセスを否定します (FrontBase は SQL 9 2 の拡張子としてパスワードの保護を提供します)。

新しいデータベースを作成するとき、多くのユーザ名を同様に、次のどちらの間に作成します：_SYSTEM と _PUBLIC。両方のこれらのユーザ名は、特別なユーザ名として、SQL 9 2 によってみなされません、実際、主な下線は通常の識別子で使うことができません、そして使わないはずです。

新規ユーザを作成するために：

```
CREATE USER <user name> [DEFAULT SCHEMA <schema name>];
```

デフォルトのスキーマを変えるために：

```
ALTER USER <user name> SET DEFAULT SCHEMA <schema name>;
```

データベースがアクセスするときにはいつでも、ユーザ名を作成するとき、存在するはずのオプションの <スキーマ名> はユーザのためにデフォルトのスキーマであるでしょう。もしデフォルトの <スキーマ名> を与えないなら、ユーザ名と同じつづりを使ってスキーマを作成して、そしてデフォルト (ユーザがデータベースにアクセスする最初するとき、これは起こるでしょう) として使います。

誰が現行のユーザであるか見るために

ユーザと CURRENT_USER の文字列の関数は使うことができます (ユーザは CURRENT_USER のために簡略なコミュニケーションをとります)。

ユーザ名を現行のユーザにするために :

```
SET SESSION AUTHORIZATION <user name>;
```

定義したユーザ名のリストを見るために :

```
SELECT * FROM INFORMATION_SCHEMA.USERS;
```

日付、時間と時刻表示

SQL 9 2 は次のデータタイプを含む複雑な時間の概念を持っています :

- 日付
- 時間
- タイムゾーンを持っている時間
- 時刻表示
- タイムゾーンを持っている時刻表示

日付は年、月と日、すなわち、時間でない部品を持っています。

時間は時間、分と秒を持っています。

時刻表示は年、月、日、時間、分と秒を持っています。

時間あるいは時刻表示のリテラルをデータベースに挿入するとき、サーバーのタイムゾーンはリテラルに加えます。例：もしサーバーがデンマークで走っていて、そして8月であるなら、サーバーのタイムゾーンは GMT+02:00 です。もし時刻表示を「1999-08-02 11:49:00」と挿入するなら、リテラルは +02:00 に調整します。もし、しかしながら、時刻表示を「1999-01-02 11:49:00」と挿入するなら、リテラルは +01:00 に調整します (1月には夏時間がないから)。

もしあなたがタイムゾーンに関して完全なコントロールを望むなら、あなたはタイムゾーンデータタイプを持っている時刻表示を使うべきです。

Example: `TIMESTAMP '1999-08-02 11:49:00-08:00'`.

同じコメントは時間とタイムゾーンを持っている時間を適用します。

キーワードとアイデンティファイア

SQL 9 2 はキーワードの非常に広範囲のセットを持っています、そして、識別子をつづることをセレクトするとき、あなたは若干の驚きに出くわすでしょう。それは SQL 9 2 のキーワードをつづることで衝突する可能性が非常に高いです。同様に識別子は下線でスタートすることができないことに注意してください。

「衝突の問題」を減らすには 2 つの方法があります：

- ・ 下線で終わっている SQL 9 2 にキーワードがありません、例えば識別子として `SELECT_` を使うことは正しいでしょう。
- ・ ダブルクォートの識別子を囲むことで、本質的にすべてのつづりで、識別子として、使うことができます、例えば `"SELECT"` は正しい識別子です。

SQL 9 2 は大文字小文字の違いを無視します、例えば識別子としての `Movies` は `MOVIES` と同一であるとみなします。

もっと多くの SQL 9 2 についての知識

あなたは常に ANSI あるいは ISO から規格自身のコピーを手に入れることができます、しかし規格は本当にユーザに向けられていません。SQL 9 2 をもっと知れたければ Chris J. Date と Hugh Darwen が書いた「SQL 規格のガイド、第 4 版」を読んでください。この本は時々非常に理論的な観点があり、それにもかかわらず非常に完成した、そして絶対に読める、そして理解できる SQL 9 2 のすべての概念と構築について説明します。

その本はアディソン - ウェスリーによって出版されています、ISBN # : 0-201- 96426-0 です。この本を注文する簡単な方法は www.amazon.com に行くことです、または、あなたの町の書店にあるかもしれません。

トランザクションを理解する

このセクションは手短かにトランザクションのデータベース概念、アイソレーションレベル、ロックの仕方とアップデートする方法を記述します；FrontBase データベースに並列アクセスをコントロールするために使ったすべての概念

並列アクセス

データベースサーバーの基本的な機能の1つが共有したデータに並列アクセスでユーザを提供するはずですが、それで、データにエラーがあるか、あるいはデータを失われないように、データベースサーバーはデータベースに作ったアップデートが規則正しい方法で行うことを保証するはずですが。

トランザクション

トランザクションはデータベースへのコントロールユーザのアクセスに使用します。ユーザはトランザクションなしでデータベースにアクセスすることができません、そしてすべてのオペレーションはトランザクションのコンテキストで行われます。トランザクションを確定するとき、トランザクションのコンテキストのユーザによってデータベースにしたすべての変更は他のユーザに明らかにします。トランザクションは、外部から見える、1つのアトミックのオペレーションです。

トランザクションを失敗するか分からない間、あなたは失敗したトランザクションを確定することができません、とるべき唯一の動作は振り出しに戻ってまたスタートすることです（トランザクションが次に失敗しないであろうという期待を持って）。データベースサーバーが、原則として、思うままにトランザクションに失敗することができます、しかし良いサーバーは、唯一、良い理由のためにトランザクションを失敗するでしょう。唯一の良い理由とはデッドロックのようなアクセスの衝突の場合です。

トランザクションを作るとき、それはアイソレーションレベル、更新の力業とロックの仕方を割り当てます。更新の力業はアクセスが読み込み専用か、あるいは書き込みと読み込みの両方かどうか決定します、そしてロックの仕方はデータベースへのアクセスを同期させるために使ったロックのタイプを決定します。

更新の力業

更新の力業は読み込み専用か、あるいは書き込みと読み込みの両方ができます、読み込み専用の更新の力業を持っているトランザクションはデータベースを修正することができません。読み込み専用のトランザクションは干渉しないから、更新の力業は非常に重要です。

アイソレーションレベル

SQL92 は4つのアイソレーションレベルを定義します：

- － 確定しない読み込み
- － 確定する読み込み
- － 繰り返し可能な読み込み
- － シリアルに伝送できる

そして FrontBase はもう1つを定義します：

- － 作り替える

データベースのデータにアクセスしているユーザが次の現象を経験するでしょう：

無効な (Dirty) 読み込み

1つのトランザクションがデータベースに若干のデータを書き込みます、2番目のトランザクションがその時そのデータを読み込みます、しかし最初はトランザクションを巻き戻します。2番目のトランザクションは今本当に存在しなかったデータを読み込みました。

繰り返しができない読み込み

トランザクションが列を読み込みました。2番目のトランザクションが列の値をアップデートして、そして確定します。もし最初のトランザクションが再び列を読み込むなら、それは違った結果になるでしょう。

ファントム

1つのトランザクションはデータベースの若干のデータをセレクトします、2番目のトランザクションはアップデートするか、あるいは

最初のトランザクションが使った属性を満たす列に挿入します。2番目のトランザクションを確定します。もし最初のトランザクションが再びセレクトしたものを実行するなら、それは違った結果になるでしょう。

下の表は所定のアイソレーションレベルがどの現象を認めるか示します：

	無効な (Dirty) 読み込み	繰り返しできない	ファントム
確定しない読み込み	はい	はい	はい
確定する読み込み	いいえ	はい	はい
繰り返し可能な読み込み	いいえ	いいえ	いいえ
シリアルに伝送できる	いいえ	いいえ	いいえ
作り替える	いいえ	いいえ	いいえ

ロックするデータの量はアイソレーションレベルによって反映します。無意味な確定しない読み込みをロックします。無意味な確定する読み込みをロックします、しかしただ確定するだけのデータは読み込みます。それらをセレクトするとき、繰り返し可能な読み込みは列をロックします。シリアルに伝送できることは表全体をロックします。あなたが FrontBase を使うとき、確定しない読み込みは確定する読み込みにアップグレードします。

作り替えるアイソレーションレベルは読み込み専用のトランザクションに有効なだけで、そして、トランザクションの継続時間のために、データベースの現在のバージョンを保持するでしょう、要するに、その時データベースのスナップ写真を作って、トランザクションを作りました。他のトランザクションがデータベースを修正することがあるでしょう、しかし変更は作り替えるトランザクションには見えないでしょう。作り替えるトランザクションのすべての番号で、データベースの確定したバージョンを共有して、同時に進行中であることができます。

ロックの仕方

更新の作業とアイソレーションレベルのほかに、FrontBase はロックの仕方の概念を導入します。ロックの仕方は次の値を持っています：

- 悲観的

- 延期する（同様に「アップグレードできる」と呼ばれます）

- 楽観的

悲観的にロックすることは所定のオブジェクトを変更するだろうと想定します、すなわち、オブジェクトが（ロックしていないで）利用可能になるまで、トランザクションを待つはずで、トランザクションが待っている時はデッドロックする可能性があります。デッドロックを発見して、そして、デッドロックを起こしているトランザクションの1つが失敗することで、壊れます。

楽観的にロックすることは所定のオブジェクトが他のトランザクションによって変更しないと想定します、そしてすべての変更で面倒なことはこれまでにして行われます。トランザクションを確定するとき、アクセスしたオブジェクトが、トランザクションの間に、変更しなかったことをチェックします、もしそれらが変更したなら、確定は失敗します。

延期することは、オブジェクトが読み込み専用と仮定する悲観的にロックすることのバージョンです、最初は、ロックは読み込みのロックです、そしてもしオブジェクトがアップデートするなら、ロックは書き込みのロックにアップグレードします。

ロックすることと EOF

EOF は楽天的なロックされることを使っています、トランザクションが例えばフェッチによって始められて、そして、変更がセーブされるとき、終えられます。EOF は完全に正しくないアップデートしようとしているオブジェクトをチェックするだけで、アクセスしたすべてのオブジェクトをチェックするべきです。ユーザは列の番号をロードして、若干の計算をして、そして列の結果を記憶します。計算のために使ったすべての列が見つけれない場合、変更するかもしれませんが、そして結果は間違っているでしょう。

FrontBase は楽観的にロックされることを実装します。ネストしたトランザクションをクライアントに割り当てることによって、EOF での問題は解決することができます、ユーザがオブジェクトをセレクトするとき、トランザクションをスタートして、そして、ユーザが変更として保存するとき、それを確定してください。実際のロールバックと確定がユーザにとって利用可能にします。もしサーバーがロックしていることを実装するなら、EOF でロックすることは不必要です、そしてスナップショットなどがたぶん止められるでしょう。

アドミニストレーション

FrontBase は「維持管理なし」でできるように設計しています、そして多くのアプリケーションのために、データベースを始めることはおそらくあなたがする必要のあるすべてであるでしょう。もっと多くの進歩している、あるいは重大なアプリケーションがユーザを管理して、バックアップ、現在のインストールを保っている、チューニングと他のタスクを必要とするでしょう。これらのタスクは単純で、そして簡単に使えるように設計しています、そしてそれらの多くがターミナルウインドウのコマンドラインから実行するでしょう。これらは、この章で説明します。FrontBase はさらに FrontBase サーバーが次のページに詳述した管理するための 3 つの主要なツールを提供します。この章の事例は FrontBase のインストールを所有しているユーザとしてログインしていて、そして FrontBase /bin ディレクトリが \$PATH であると仮定します。

この章は次のセクションを含んでいます： - 88 ページの「アドミニストレーションツール」。

- 89 ページの「トランザクションのログを書き出す」。
- 95 ページの「複製」。
- 103 ページの「クラスタリング」。
- 108 ページの「バックアップと復帰する」。
- 111 ページの「拡張した書式のないファイルの読み込みと書き出しの関数」。
- 116 ページの「インデックスを付ける」。
- 120 ページの「FrontBase を調整する」。
- 132 ページの「移行」。
- 135 ページの「<install dir>/FrontBase/bin/
FrontBase -logSQL [<otheroptions>]
<database name> &」。

アドミニストレーションツール

「sql92」コマンドラインツール、Web ベースの「FBWebManager」ツールと「FrontBaseManager」アプリケーション（MacOS X と MacOS X サーバー 10.x のために）は下に示したようにそれぞれ別の章で説明しています：

FrontBaseManager

1 3 7 ページの「FrontBaseManager」は、ローカル、そしてリモートのデータベースサーバーを監視して、そして管理するアプリケーションを導入します。それは FrontBase の MacOS X のインストールで利用可能です。Windows のインストールはオリジナルの FBManager を使います。

FBWebManager

2 0 5 ページの「FBWebManager」は、FrontBase のすべてのインストールで利用可能であって、そしてダイナミックな HTML（JavaScript としても知られている）をサポートするすべてのブラウザを通してアクセスする FBWebManager Web ベースのアプリケーションを導入します。

sql92

2 3 5 ページの「sql92」は、FrontBase のすべてのインストールで利用可能な sql92 コマンドラインツールを導入します。

FBScriptAgent

FBScriptAgent はコマンドを FrontBase データベースに転送する「scriptable」アプリケーションです。未処理の SQL コマンドを送ることに加えて、FBScriptAgent はスタートして、ストップして、作って、そして FrontBase データベースを削除することができます。FBScriptAgent は前もってもっと多くの複雑なデータベースクライアントライブラリを通して利用可能であった多くのデータベースの機能性を見せます。ユーザはほとんどアップルスクリプトコマンドのキー入力によって FrontBase の強力な特徴にアクセスすることができます。

FBScriptAgent の巧妙なデザインの外観はそれがスクリプターがメタデータとデータハンドラを扱うことを可能にするということです。

例えば、もしデータベースの取出しが1,000,000のレコードをセレクトするなら、これはもしすべてのレコードを返したなら、扱うべきほとんどのアプリケーションのために難しいでしょう。その代わりに、FBScriptAgentの取出しは、ユーザがどのくらいのレコードを選択したか決定することができるメタデータオブジェクトを返します、すべてのエラーあるいは警告は、セレクト、実行時間、行の名前などによって作成します。メタデータは同様に、もし必要とするなら、実際に、バッチの、列を取って来るために使うであろうフェッチ「ハンドル」を含みます。知的にFBScriptAgentを設計することの特徴はデータを大量に使うて容易にユーザを動作することを可能にする方法です。

アプリケーションは、文書と事例で、<http://www.frontbase.com>で「ダウンロード」セクションを公表されています

トランザクションのログを書き出す

デフォルトで、FrontBase（バージョン 3.x とそれ以降）はトランザクションヒストリーログを維持します。これはデータベースのデータあるいは構造を変更したすべてのトランザクションの、SQL 文のフォームの、完成したリストです。

従ってこのようなログはデータベース開発の完全なヒストリーを提供します、そして、本質的にトランザクションログの SQL 文をやり直すことで、特定の開始点からデータベースを再び作ることができます。このような特定の開始点はデータベースの作成あるいはデータベースのバックアップを作成したポイントです。

FrontBase のトランザクションのログを書き出す概念はいくつかの目的を助けます：

1. それはデータの損失に対して余分のレベルのセキュリティを提供します
2. ディスクオペレーションの処理（すなわちクライアントの処理は完了したディスク書き込みのオペレーションを待たなくてもよい）からクライアントインタフェース処理の相互作用を断つことはデータベースサーバーをイネーブルします
3. それはベースとしてクラスタリングと複製をデータベースのためにサポートします。

上記の2番目のポイントはトランザクションログがどんな所定の時点でも実際のデータベース内容よりも「進んでいる」であろうことを意味します。FrontBase が既存のデータベースのためにスタート

するとき、従ってそれは自動的にトランザクションログがデータベースに確定しなかったトランザクションが入っているかどうか調べます、そしてこの場合データベースは最新になります。

自動的に敬意を払った最新のデータベースをそのトランザクションのログにする、このメカニズムはクラスタリングと複製に関連して同様に役に立ちます。次のセクションは FrontBase のトランザクションのログを書き出すさまざまな要素を説明します：

- 90 ページの「実装」。
- 91 ページの「アドミニストレーション」。
- 92 ページの「SQL の文法」。
- 93 ページの「オプション」。
- 93 ページの「ユーティリティー」。

実装

データベースのトランザクションログは FrontBase のインストールの TransactionLogs ディレクトリにあります： TransactionLogs/<dbname>。

このディレクトリはそれぞれが命名する 1 つ以上のトランザクションログディレクトリを含みます： L_yyyy_mm_dd-hh_mm_ss、ディレクトリを作ったとき、時間のポイントを識別します。より多くのものが存在するとき、従って名前はトランザクションログディレクトリーの正しい順序を提供するでしょう。

トランザクションログディレクトリは transactions.log という名前のファイルそして多分 (ttttttt がトランザクション番号である) tttttttt.sql という名前のファイルに入っています。後のファイルは特に大きいトランザクションに入っています。それが継続中であるのに対して、FrontBase は、ただ、トランザクションに確定するとき、ログファイルにそれを書き込むことだけをして、トランザクションのコア内のログを保とうと試みます。しかし、トランザクション自身がログファイルの書き込みを超えたとき、コア内のトランザクションの大きさに限界があります (現在 1 MB)。

新しいトランザクションのログのディレクトリを作ることを超えたとき、トランザクションのログのディレクトリで保たれた (結合した) ファイルのサイズへの限界があります。この限界はデフォルトで 512 MB です、しかしユーザによって変更することができます。

トランザクションのログが特定の開始点からデータベースをアップデートするために使うとき、開始点が明瞭であることは明らかに必要です。データベースのバックアップを作るとき、不可分なトランザクションのログのディレクトリの変更が行われるでしょう。2つの最初のキャラクタ（それぞれ B_ と L_）以外、バックアップ・ファイルと新しいトランザクションのログのディレクトリの名前は同一であるでしょう。

アドミニストレーション

トランザクションのログを書き出すための使用法はオプションです、しかしそれは FrontBase のデフォルトと、付加的な保管について勧めます。トランザクションのログを書き出すことが複製とクラスタリングのコンテキストで FrontBase を走らせて要求することに注意してください。

トランザクションのログを書き出すことがイネーブルであるとき、FrontBase はユーザインタフェースが、トランザクションが物理的にファイルシステムに書き込む前にさえ、継続するのを認めます。これはトランザクションのログがディスクにあるデータベースよりも「進んでいる」ことを意味するかもしれませんが、そしてこの状況でトランザクションのログは維持しているトランザクションのために極めて重要です。ディスク上のデータベースの状態は常に安定しています — すなわちそれは実際にディスクに書き込んだ最新のトランザクションを反映します — しかしそれは実際にトランザクションのログの状態にするのに遅れるかもしれません。

FrontBase サーバーが既存のデータベースでスタートするとき、（もし存在しているなら）データベースがトランザクションのログに関して最新であることを自動的に確認します、そしてもしそうでなければ、それを最新の状態にします。これは最新のトランザクションのログのディレクトリが決定的に常に適切であることを意味します。一般に、それは、トランザクションのログのディレクトリへの変更が行われるとき、正確に予測することが難しいです（ログサイズの制限によって起こっている）、そして、トランザクションのログがデータベースよりもどのくらい進んでいるか予測することは同様に難しいです。従って、トランザクションのログのファイルが最後の明瞭な開始点：データベースの作成あるいはバックアップのポイント、に戻って重要であると述べるのが最も安全です。

ディスク・スペースの節約より他の理由でトランザクションのログファイルを取り除くことは決して必要ではありません。

次の単純な規則はトランザクションのログを書き出すことの管理の処理を規定します：

データベースのバックアップが不可分な新しいトランザクションのログのディレクトリを作って、そしてすべての前のトランザクションのログのディレクトリをすたれたものにします。

これはちょうど作成したバックアップに敬意を払ったことを意味します、すべての前のトランザクションのログのディレクトリはデリートされるでしょう。インストールが物理的にそれらを削除する前にいくつかのバックアップと添付のトランザクションのログの生成を保持することを望むか、個別の問題です。

SQL の文法

次の SQL 文はトランザクションのログを書き出す処理のために導入しました：

```
CREATE TRANSACTION LOG;
```

トランザクションのログを書き出すこと（デフォルト）をイネーブルします。もしトランザクションのログを書き出すことがディスエーブルだったなら、新しいトランザクションのログのディレクトリが作られるでしょう

```
DROP TRANSACTION LOG;
```

(推薦されていない) トランザクションのログを書き出すことをディスエーブルします

```
SET TRANSACTION LOG LIMIT <integer-expr>;
```

n が <integer-expr> によって指定する値である場合は、トランザクションのログのディレクトリのサイズを n MB にセットします

```
SHOW LOGS [ALL | <integer-expr> ];
```

n が <integer-expr> によって指定する値である場合は、n 個の最新のトランザクションのログのディレクトリのまとめを示します。0 の値（あるいはすべて）がすべての既存のトランザクションのログのディレクトリを意味します。<integerexpr> の欠如は 1 を意味します（すなわち最も新しいということです）

```
SWITCH TO NEW TRANSACTION LOG;
```

新しいトランザクションのログのディレクトリを作ります

WRITE DATA;

データベースのバックアップを作って、そして不可分な新しいトランザクションのログのディレクトリを作ります。このように、バックアップはちょうど始めたトランザクションのログに明瞭な出発点を提供します。

オプション

次の新しい起動オプションは FrontBase サーバーで導入しました：

`-tlog=no|yes`

サーバーをスタートするときに、データベースを使って記憶したトランザクションのログを書き出す方式をオーバーライドします；この方式は、永久を変更することができません。

`-keptlog`

サーバーは、もし1つが存在するなら、トランザクションのログをクリアするのを阻止します。

これらのオプションもまた、通常の（ごくわずかの）オペレーションでは役に立ちません。

ユーティリティー

次のユーティリティーは所定のデータベースのためにトランザクションのログの状態と内容を調べるのに役立ちます。それらは / FrontBase/bin ディレクトリにあります：

**FBTLog [-d <transactionlog-directory>] <database-name>
[<transaction-number>]**

デフォルトで、最も古いトランザクションのログのディレクトリからスタートしてトランザクションのログで、トランザクションをリストします。もし指定されるなら、<transaction-number> がリストするであろう最も古いトランザクションの番号を識別します、そして -d オプションは開始点として指定したトランザクションのログのディレクトリを識別します。

FBTLogs [-s <file-count>] <database-name>

n が <file-count> によって指定する値である場合は、n 個の最新のトランザクションのログのディレクトリのまとめを示します。0 (デフォルト) の値はすべての既存のトランザクションのログのディレクトリーを意味します。

複製

複製とクラスタリングは余分（サーバーが利用できなくなる時、休止時間から保護します）と負荷分散（重負荷の状況でもっと良い応答時間）の両方に役立ちます。

複製

FrontBase は、マスターをアップデートするとき、アップデートするいくつかの読み込み専用のクライアントに1つの原版のデータベースの複製をサポートします。クライアントは原版にかなりルーズな結合を持っています；クライアントが、例えば、それがしばらくの間、オフラインであったとき、それは古いでしょう。クライアントが再びオンラインで来るとき、それは、オフラインの間に実行したトランザクションからアップデートするでしょう。

原版のデータベースとクライアントが大ざっぱに結合するだけで、そしてまったく原版のサーバーはクライアントと結び付けるオーバーヘッドを持ちません、原版の複製は時間がかかるバックアップを取り除いて、そして、原版から、書き込みのタスクを報告するために使うことができます、そしてクライアント自身によってバックアップの役目を果たすでしょう。それを別として、複製がデータベースの読み込み専用のコピーのもっと多くの従来の流通のために使うでしょう。

導入

クライアントがすべて読み込み専用であるとき、原版からクライアントへの変更の移行は、プロトコルがクラスタリングのために必要としたツーフェイズコミットのような、複雑な同期化を必要としません、けれども原版とクライアントで実行したトランザクションを数えることによって、簡単にコントロールすることができます。トランザクションが、クライアントで同期なしで、原版に確定することができます、従って、サーバーがトランザクションの必要な一連番号を提供して、そして同様にサーバーの外側の変更の表示を提供する限り、実際の複製のメカニズムはサーバーの外で動かすことができます。この外部表現はデータベースのトランザクションのログによって提供します。

全体的なデザイン

複製のシステムが成り立つには：

- － 「原版のデータベースサーバー。」
- － 「複製のプロセス。」
- － 「クライアント。」

原版のデータベースサーバー

複製をイネーブルするために、FrontBase サーバーは `-rmaster` オプションを使ってスタートするはずです：

```
FrontBase -rmaster <other-options> <master-database-name>
```

複製のプロセス

原版のデータベースの複製は個々のプロセス、複製のデーモンでし
ます、そしてそれは原版が維持するトランザクションのログを読み込み
ます。トランザクションのログは原版へのすべての修正のために
SQL に入っています。トランザクションを確定するとき、トランザ
クションのための SQL はトランザクションのログに書き込みます。
実際にデータベースを修正する SQL だけをログに書き込みます、そ
れでそれは主に挿入と更新の文です。

複製のデーモンは原版のトランザクションのログを監視します、そし
てトランザクションが完全にログファイルに書き込んだときはいつ
でも、この SQL はアクセス可能なすべてのクライアントで複製の
デーモンによって実行します。トランザクションのログを監視する
ことは基本的に複製のデーモンの反応時間をコントロールする構成
可能な間隔でのポーリングです。クライアント間で妨害をしないた
めに、通信の遅れの可能性がある 1 つのクライアントで他のいかなる
クライアントにも影響を与えないように、それぞれのクライアントへ
の複製がコントロールのスレッド自身で行います。

複製のデーモンは報告の状態とクライアントの作成、追加と除去、の
ために小さいツールを持って来ます、そしてトランザクションのログ
は検査のためにツールを持って来ます。

複製のプロセスはコマンドを使ってスタートします：

```
FBReplicator [-v] [-s] [-i <seconds>] [-d <transactionlogdirec-  
tory>]  
<database-name>;
```

- ・ -v
冗長です。複製のデーモンは stdout にイベントをログファイルに書くでしょう。
- ・ -s
静かです。複製のデーモンはイベントをログファイルに書かないでしょう。
- ・ -i <seconds>
間隔。ポーリングの間隔は複製のデーモンがどれくらいよく原版のトランザクションのログを調べるかを決定します。複製のデーモンはトランザクションが差し迫っている複製ではないとき、指定した量の時間をスリープするでしょう。デフォルト値は 10 秒となっています。
- ・ -d <transactionlog-directory>
トランザクションのログのディレクトリは複製のために使うトランザクションのログのディレクトリを指定します。通常の場合、このオプションを指定することは必要ではありません。

複製のデーモンが原版のトランザクションのログを調べて、そして読み込むとき、複製のデーモンは原版のデータベースと同じホストで走るはずです。

クライアント

複製のクライアントは `-rclient` オプションを使ってスタートするはずです：

FrontBase -rclient <other-options> <client-database-name>

それはデータベースをアップデートするために複製のクライアントのユーザを拒否します。クライアントのデータベースは原版を使って複製のセットアップを作るはずです、それで原版とクライアントは普通のヒストリーを共有します。言い替えれば、複製のシナリオを準備する前に、クライアントデータベースは作ることができません。

FBTlogs

FBTlogs[-s <directory-count>] <database-name>

FBTlogs ツールはデータベースのためにトランザクションのログの状態を表示します。ツールが直接ログにアクセスするとき、それはデータベースサーバーと同じホストで走るはずです。

オプション

-s <directory-count>

考慮するためにトランザクションのログのディレクトリの番号を指定します。デフォルトはすべてです。

引数

<database-name>

関連したデータベースの名前。

FBTlog

FBTlog[-d <transactionlog-directory>]<database-name>[
<transaction-number>]

FBTLog ツールはデータベースのためにトランザクションのログの内容をリストします。
ツールが直接ログにアクセスするとき、それはデータベースサーバーと同じホストで走るはずです。

オプション

-d <transactionlog-directory>

考慮するために最初のトランザクションのログのディレクトリを指定します。デフォルトは一番古いです。

引数

<database-name>

関連したデータベースの名前。

[<transaction-number>]

指定した番号を使ってスタートするトランザクションをリストしてください。

FBRAccess ツール

FBRAccess <command> <arguments>

FBRAccess は複製のデーモンを管理するコマンドラインツールです。このツールのために、データベースの名前は一般に、「<db>[@<host>]」として、書くことができます、<db> はデータベースの名前で、そして <host> はホストとして機能しているマシンの名前です。ホストの規格の欠如は localhost、現在のマシンを意味します。ツールは次のコマンドを実行します：

```
FBRAccess add <master-host> <master-database> <client-database>[<databasepassword>[<system-password>]]
```

```
FBRAccess create <master-database> <client-database>[<databasepassword>[<system-password>]]
```

```
FBRAccess remove <master-database> <client-database>
```

```
FBRAccess status <master-database>
```

- Create コマンドは原版にクライアントを加えます。もしクライアントデータベースが存在しないなら、それは作って、そして原版のコピーを使って初期化します。もしクライアントデータベースが存在するなら、複製のプロセスはクライアントサーバーをアップデートするためにスタートするでしょう。もしあなたが大きいデータベースを扱っている、そして新しいクライアントを作ることを望むなら、他の手段によってコピーを提供することは、それが何度も、リソースを消費することができるかもしれないタスクを、複製のプロセスが行うより良いでしょう。もし指定するなら、オプションのパスワードはクライアントデータベースで _SYSTEM ユーザのためにデータベースパスワードとパスワードのはずです。複製はパスワードがクライアントに接続を確立することを必要とします、しかし、もしそれらが原版のために現在のパスワードと異なっているなら、パスワードを指定することが必要なだけです。
- Add コマンドは存在しないクライアントデータベースを作らないであろう、そして、もしそれが走っていないなら、クライアントをスタートしないであろうこと以外、Create コマンドとして動作します。
- Remove コマンドはクライアントを複製のリストから取り除きます、複製のプロセスはクライアントにアップデートを送るのをやめるでしょう、しかしサーバーは走っています。
- Status コマンドはクライアントの状態の情報を返します。最初の行は原版のためにホスト名とデータベース名と現在のトランザクションの番号を印刷します。それぞれのクライアントの

ために、行がクライアントのデータベース名、ホスト名、現在のトランザクションの番号と状態を印刷します。

新しいクライアントを作成する方法

クライアントを作る最も単純な方法は FBRAccess create コマンドを使うはずですが、しかし、それが大きいデータベースのために時間がかかるかもしれないとき、クライアントホストに原版のデータベースをコピーしてインクルードすることは常に望ましくありません。

もしあなたが同じ エンディアンタイプのホストを動作しているなら、あなたは原版のホストからクライアントまでデータベースをコピーしてもよいです；しかし、ファイルをコピーするとき、原版のデータベースが走っていないことを確認してください。もしあなたが原版のデータベースを止めることを望まないなら、あなたは書き込みデータコマンドを使ってバックアップを作って、バックアップをクライアントホストに動かして、そしてそこに復帰することができます。

もしあなたが異なったエンディアンタイプのホストを動作しているなら、あなたは WRITE ALL コマンドを使ってデータベースの書式のないファイルのバージョンを作るはずですが（1 1 1 ページの「拡張した書式のないファイルの読み込みと書き出しの関数」を見てください）、それはデータベース内容のエンディアンの公平なアスキーの表示を作ります。それをクライアントホストに動かして、そしてデータベースに書式のないファイルをロードしてください。

もしあなたが完全なトランザクションのログを持っているなら（通常、若いデータベースのために持っているだけです）、あなたは簡単にクライアントデータベースをスタートすることができます。

クライアントデータベースを確立したとき、ちょうど FBRAccess を使っているクライアントのリストにクライアントを加えてください。

複製の準備をする

新しい原版のデータベースと関連づけたクライアントの複製の準備を行うためのターミナル指向の順序は次の通りです：

1. マシン masterhost に原版のデータベースを、例えば masterDB をスタートしてください：

```
FrontBase -create -rmaster masterDB &
```

2. sql92 ツールを使って若干のトランザクションを確定してください、例えば：

```
CONNECT TO masterDB USER _SYSTEM;
CREATE USER u1;
CREATE SCHEMA AUTHORISATION u1;
ALTER USER u1 SET DEFAULT SCHEMA u1;
DISCONNECT CURRENT;
QUIT
```

3. 複製のデーモンをスタートしてください：

```
FBReplicator masterDB &
```

他のマシン（例えば clienthost）で：

4. クライアントデータベース（例えば clientDB）をスタートしてください

```
FrontBase -create -rclient clientDB &
```

最初のマシンで：

5. 次のように masterDB から clientDB まで接続を確立してください。

```
FBRAccess add masterDB clientDB@clienthost
```

原版のクライアントの関係は確立しています。例えば、行為によって、2つのデータベースのトランザクションログを調べてください：

6. masterhost で：

```
FBTLog masterDB
```

clienthost で：

```
FBTLog clientDB
```

もし、他方では、あなたは新たなクライアントに繰り返すことを望む既存の原版のデータベースを持っています：

- － 上記のポイント 1 から -create を省略してください
- － 上記のポイント 2 を省略してください

複製とパスワード

複製のデーモンは、標準のクライアントライブラリを使って、複製のクライアントのためにあらゆる点で普通のユーザです。それがクライアントと連絡を取ることを確立するとき、それは、_SYSTEM ユーザとして、従って、接続を確立するために _SYSTEM とデータベースパスワードを知るために必要とします。複製のデーモンは所定のクライアントのために現在のパスワードの追跡をします、それで、クライアントがオフラインであるとき、あなたは原版のパスワードを変えてもよいです。

もしあなたが上記に記述するように、若干のソートのコピーを新しいクライアントに作るなら、複製のデーモンはデータベースのためのパスワードを知りません、そして従ってクライアントに接続することができません。従って FBRAccess ツールはあなたに、クライアントを加えて / 作るとき、現在のパスワードを指定することを与えます。

データベースの身元確認チェック

それぞれの FrontBase データベースは 4 8 ビットの唯一の身元確認があります。複製のデーモンは原版のデータベースの唯一の身元確認がそれぞれのクライアントの唯一の身元確認とまったく同じであることを必要とします。このチェックは、同じヒストリーを共有して、原版とクライアントが関連していることを保証します。

複製のコマンド

sql92 のコマンドラインのツールは今、複製の原版とそのクライアントをコントロールして、そして保つためにコマンドをインクルードします。sql92 のコマンドはあなたが原版のサーバーにセッションを持つと想定します、そしてコマンドは対応する複製を動作します。すべての関連したコマンドは 2 4 6 ページの SQL コマンドセクション - 「複製をスタートする」で見ることができます。

事例：

事例は 1 つのクライアント使って原版のデータベースを作ります。

```
create database master options -rmaster;  
connect to master user _system;  
start replicator;  
create client client;  
show clients;
```

```
table t0 ( c0 integer );
insert into t0 values 1;
table t0;
disconnect all;
connect to client user _system;
table t0;
disconnect all;
```

クラスタリング

複製のクラスタは1つのデータベースを実行するデータベースサーバーの番号です、クライアントがクラスタでどんなサーバーでも接続して、そしてデータベースにアクセスして、そしてアップデートすることができます

バックグラウンド

クラスタリングデータベースの理由は次のように最高の記述をすることができます：

1. 拡張したデータのセキュリティ

クラスタのそれぞれのデータベースサーバーがデータベースの完全なコピーを維持します、それで1つのマシンがそのまま残っている限りデータは失われません。

2. 負荷分散

典型的なクライアントアプリケーションは、規模が、読み書きのトランザクションのレートより大きいレートを使って読み込みのトランザクションを実行するでしょう、そして読み込みのトランザクションは、それが関係しているクラスタメンバーにアクセスするだけでしょう、従ってロードを流通することができます。

3. ホットスペア

クライアントがクラスタの1つのデータベースに接続を解放するとき、それはもう1つにも簡単に接続することができます、失われるであろう唯一のデータは未処理のトランザクションのデータです。

実装

1. ツーフイズコミット

クライアントがクラスタのメンバーを修正するとき、修正を確定することができる前に、クラスタの他のメンバーはアッ

プデートするはずです。クラスタのメンバーをアップデートするとき、トランザクションは、いわゆるツーフェイズコミットアルゴリズムを使って確定します、そして、トランザクションはすべてのクラスタメンバーに委託するか、あるいは一つも委託しないことを保証します。

2. クラスタを冷たいにスタートする

クラスタの最初のメンバーがスタートするとき、そのサーバーがデータベースの最新のバージョンを持っていると想定します。サーバーがたった1つのクラスタであるとき、それはクラスタを使って同期させる必要がありません。

3. クラスタにメンバーを加えること

クラスタの残っているメンバーがスタートするとき、それらはクラスタのメンバーを決定して、そしてクラスタを使って同期させるために新しいメンバーの前方を回転するでしょう。もし新しいメンバーとクラスタを同期させることが可能ではないなら、新しいメンバーは中断します。新しいメンバーを同期させるとき、それはクライアントが接続することを可能にするでしょう、そしてそれはクラスタメンバーの残りからアップデートを受け取るでしょう。

4. メンバーをクラスタから取り除くこと

メンバーをクラスタから取り除くことは簡単です：ただそのサーバーを止めて / 抹殺すればよいです。

5. クラスタへのクライアントの接続

クライアントがクラスタに接続することを望むとき、それがメンバーの1つと接続しようとすることができます、そしてもしそれが失敗するなら、有用なサーバーを見つける、あるいはすべてのサーバーが試みて、失敗するまで、2番目を試みます、その場合、クラスタに接続することは不可能です。

事前のチェック

クラスタで使ったサーバー / マシンがネットワークやインターネットによってアクセスできることを調べてください。もしマシンにアクセスすることに問題があるなら、システム管理者と連絡を取ってください、DNS の準備に問題があるかもしれません。もしあなたが使ったマシンですでに走っているデータベースクラスタがあることを知っているなら、問題はないでしょう。

Mac OS X サーバーで、コンピュータの設定とネットワークの選択を行ってください。DNS のセクションに行って、そして「サーチドメイン」ボックスをチェックしてください。もしドメインがプロジェクトを使うために項目がないなら、あなたが必要とするドメイン、例え

ば「hotroof.com」をエンターしてください。これは基本ではありません、しかし後で楽になるでしょう。

新しいデータベースをクラスタリングする

次のことは新しいデータベースのためにクラスタリングを確立することのプロセスを詳述するでしょう。

1. 第一のマシンで最初のクラスターメンバーをスタートしてください

machine1 にログオンして、そして次のコマンドラインを実行してください：

```
FrontBase -rcluster=newdb@machine1,newdb@machine2 newdb
```

これは次のような応答をするでしょう：

```
2002-01-30 12:33:35 FrontBase Server - <version-number>
2002-01-30 12:33:35 Bootstrapping new database: /opt/FrontBase/
Databases/newdb.fb
2002-01-30 12:33:35 Bootstrapping new database
2002-01-30 12:33:35 Bootstrapping DEFINITION SCHEMA done
2002-01-30 12:33:36 Bootstrapping INFORMATION SCHEMA done
2002-01-30 12:33:36 Cleared Transaction Log directory: /opt/
FrontBase/TransactionLogs/newdb
Jan 30 12:33:36 2002 [12630] Synchronized at transaction: 0
```

machine1 の newdb のための FrontBase サーバーは今走っています、そして machine2 で走っている newdb のためのサーバーをクラスターメンバーとして受け入れるでしょう。

2. 第二のマシンで2番目のクラスターメンバーをスタートしてください

machine2 にログオンして、そして次のコマンドラインを実行してください：

```
FrontBase -rcluster=newdb@machine1,newdb@machine2 newdb
```

これは次の追加と一緒に上記のような応答をするでしょう：

```
Jan 30 12:44:12 2002 [12640] Connected to newdb@machine1
Jan 30 12:44:12 2002 [12640] Inherits database ID from
newdb@machine1
```

同様に、machine1 で、あなたは次の出力を得るでしょう：

```
Jan 30 12:44:12 2002 [12630] Connected to newdb@machine2
```

今クラスタに2つのメンバーがあります、そしてそれらはお互いと通信するためにセットします。

もしあなたが2つ以上のメンバーを必要とするなら、`-rcluster` オプションを拡張してください：

```
rcluster=newdb@machine1,newdb@machine2,newdb@machine3
```

そして関連したマシンのそれぞれのポイント2を繰り返してください。

あなたは同様に後ですでに既存のクラスタに新しいクラスタメンバー（例えば、machine7）を加えることができます：

machine7 にログオンして、そして次のコマンドラインを実行してください：

```
FrontBase -rcluster=newdb@machine1 newdb
```

サーバーはクラスタの指定したメンバーからクラスタのメンバーの全リストを引き出すでしょう、それであなたは完全なリストを指定しなくてもよいです。

1つの理由あるいはもう1つの理由のために、上記のコマンド行を使って、止めたことは、クラスタメンバーをいつでも再起動することが可能です。

既存のデータベースをクラスタリングします

すでに既存のデータベースのクラスタリングを準備するために、データベースのコピーがクラスタマシンで最初に確立するはずですが。これは108ページの「バックアップと復帰」セクションで説明されています、とにかくやってみてください。

その時、前に記述した手順は次に続いています。

データベースの身元確認チェック

それぞれの FrontBase データベースは 48 ビットの唯一の身元確認を持っています。データベースクラスタのすべてのメンバーは、それらが共通のヒストリーを共有することを意味して、同じ唯一のデータベースの識別子を共有するはずですが、新しいメンバーがクラスタに加えるときはいつでも、このプロパティをチェックします。

バックアップと復帰

FrontBase は書式のないファイルに完成したデータベースをバックアップすることを可能にします。それは直接あなたのデータベースに ASCII だけのファイルからデータの内容を復帰することを可能にします。FrontBase は「生の」データベースをバックアップすることさえできます。

あなたはハードウェアの失敗、ファイルを失う、あるいは他の大惨事の損失から守るために一定の基準で重要なデータベースをバックアップすべきです。同様に FrontBase データベースを新しいサーバーに移動させるとき、あるいは診断のためにそれらを FrontBase に送るとき、あなたはバックアップする必要があるでしょう。

注：我々は Mac OS X サーバー 1.x で確かめました、重負荷で交換する時、ファイルシステムの不適切な問題を起こすことがあります。これらのファイルシステムの問題は FrontBase データベースファイルに影響を与えることができます。我々はこれらの潜在的な問題を避けるために Mac OS X サーバー 10.x にアップグレードすることを大いに勧めます。

概要

あなたがデータベースを保護するためにとることができる処置がすべてのプラットフォームで同一であるとき、一般にあなたのデータを保護することは意味をなすから、我々はあなたが追うことができる可能な戦略のこの批評記事を作り出しました。

このセクションは次のことが入っています：

- 109 ページの「データベースファイルをコピーする」。
- 109 ページの「生のデータベースのバックアップ」。
- 110 ページの「データベースを復帰する」。
- 110 ページの「書式のないファイルを書き出す」。
- 111 ページの「複製」。

データベースのファイルをコピーする

この戦略は非常に簡単で、基本的で、そして確かに何も無いよりは良いです。

FrontBase はそのデータベースファイルを `<install-path>/FrontBase/Databases` に記憶します。優先のバックアップユーティリティを使うことによって、あなたは、このディレクトリの内容をバックアップすることができます。tar ユーティリティはすべての Linux / Unix システムで利用可能です、それはコマンドラインから実行する必要があります：

```
tar cvf <destination file> <install path>/FrontBase/Databases
```

Mac OS X サーバー 10.x の事例（バックアップは現在のディレクトリで作ります）：

```
tar cvf fbbackup /Library/FrontBase/Databases
```

注：バックアップを作る前に、あなたはすべての FrontBase サーバーを止める必要があります。

生のデータベースのバックアップ

FrontBase で、それが走っている間に、あなたはデータベースのバックアップを作ることができます。バックアップはひどく目立たない、そして、バックアップが進行するのに対して、データベースはアップデートし続けることができます。

あなたは `_SYSTEM` ユーザとしてデータベースに接続して、そして次の SQL 文を出す必要があります：

```
WRITE DATA;
```

ほとんどのシステムで、バックアップをスタートするタスクは自動化することができます。Linux / Unix システムで cron ユーティリティと sql92 コマンドラインツールは、一緒に、プロセスを自動化することができます。

データベースを復帰する

助言：バックアップから、復帰する前に余分な予防の処置として、我々は最初に <install path>/FrontBase/Databases の「古い」データベースファイルをコピーすることを強く勧めます。

バックアップからデータベースを復帰するために、あなたは（ルートとしてログインしている）コマンド行からそうする必要があります：

シンタックス：

```
FrontBase -create -restore <database name>
```

書式のないファイルに書き出す

FrontBase はデータベースが後で再構築することができるから ASCII フォーマットだけのファイルのデータベース全体、スキーマ定義と内容を書き出すことができます。

書式のないファイルの書き出し / 読み込みの機能性の 2 つの目的は次の通りです：

1. ASCII の形式のデータベースのバックアップと復帰。
2. 異なった「エンディアン」タイプ（ビッグエンディアン / リトルエンディアン）のプラットフォームの間にデータベースについて再書き込みする。

次の SQL 9 2 文はそれぞれ完全なデータベースを書き出して、そして読み込むでしょう：

```
WRITE ALL OUTPUT(DIR='path to export directory', CONTENT=TRUE);
```

```
SCRIPT <path to export directory>/schema.sql;
```

このアプローチは 1 1 1 ページの「拡張した書式のないファイルの読み込みと書き出しの関数」セクションで完全に記述されています。

複製

複製のアプローチは95ページの「複製」セクションで完全に記述されています。複製はすべてのアップデートなどを受ける原版のデータベースと原版の読み込み専用のコピーである n 箇所にミラーしたデータベースを持つ方法です。

「サスペンダーとベルト」アプローチが2つのミラーしたデータベースを持つことができます、それは、通常データベーストラフィック andone を処理し、同様にバックアップを処理します。これは、安全について、関連した多数のマシン（マルチプラットフォームで簡単にできる）と多数の安全なガード技術で、非常に高い程度を保証するでしょう。

拡張した書式のないファイルの読み込みと書き出しの関数

FrontBase は前に書式のないファイルの読み込み / 書き出しの機能性を提供しました、しかし我々は FrontBase 3.x において若干の拡張をしてこれを改善しようと努めました。このセクションはこれらの拡張を説明して、そして実証しようと努めます。前の書き出しと読み込みの SQL 文はまだ関数が変化していないけれども、機能性で不完全です、従って確定することができません。

拡張した書式のないファイルの書き出しの関数

書式のないファイルの書き出しの関数の特徴、自在に使うこと、そして安定を改善するために、我々は書き出しの機能性を拡張しました。

シンタックス :

```
WRITE ALL OUTPUT(list of <key value pair>);
```

```
<key value pair> ::= <key> = <value>  
<key> ::= RSEP | CSEP | DIR | FOLDER | CONTENT  
<value> ::= "general expression"
```

RSEP – Row SEPARATOR

<value> は文字列の式のはずです、そしてそれはお互いから実際の入力
で列を分ける文字列を識別します。標準的なバックスラッシュ +
文字 combo を使うことによって、ある特定の制御文字をエスケープ
することができます：

\n 新しい行
\r 復帰
\t 水平に進むタブ

CSEP – 行のセパレータ

<value> は文字列の式のはずです、そしてそれは（列の中で）お互い
から実際の入力
で列を分ける文字列を識別します。標準的なバック
スラッシュ + 文字 combo を使うことによって、ある特定の制御文
字をエスケープすることができます（RSEP 参照）。

DIR あるいはフォルダー

<value> は文字列の式のはずです、そしてそれは書式の無いファイル
の書き出しが作られるであろうディレクトリ（フォルダー）のパスを
意味する文字列を識別します。もしディレクトリが存在しないなら、
それは作られて試みるでしょう。

内容

<value> は TRUE、あるいは FALSE のいずれか判定するブーリア
ン式のはずです。TRUE は同様にテーブルの内容が書き出すであろ
う（テーブル毎の1つのファイル）ことを意味します。

拡張した書式の無いファイルの読み込みフィルター

書式の無いファイルの読み込みフィルターの機能、自在に使うこと、
そして安定を改善するために、我々は読み込みフィルター機能性を拡
張しました。

シンタックス：

```
INSERT INTO <table> [FROM] INPUT(list of <key value pair>) [COM-  
MIT  
<count>];
```

```
<key value pair> ::= <key> = <value>  
<key> ::= RSEP | CSEP | FILE | TYPE | COLUMNS | COUNT |  
SKIP | CHECK | RDQ | STOP  
<value> ::= "general expression"
```

TYPE

<value> は「FrontBase」、「アクセス」、「Sybase」、「Informix」あるいは「MSSQL」の1つのはずです。

この値は読み込みフィルターにヒントを提供して、そしてデフォルト値の番号を意味します：

```
TYPE = 'FrontBase' =>  
RSEP = '$\n', CSEP = '$$', SKIP = 1
```

この値は同様に実際の入力ファイルが FrontBase が表を書式のないファイルに捨てる時作り出すものとまったく同じのフォーマットを持っていることを意味します（書き込みの表を参照）。

```
TYPE = 'Access' =>  
RSEP = '\n', CSEP = ';' ;
```

タイプの値は同様に日付、時間と時刻表示の値が何のフォーマットで与えるかを意味します。

RSEP - 列のセパレータ

<value> は文字列の式のはずです、そしてそれはお互いから実際の入力で列を分ける文字列を識別します。標準的なバックスラッシュ + 文字 combo を使うことによって、ある特定の制御文字をエスケープすることができます：

```
\n 新しい行  
\r 復帰  
\t 水平に進むタブ
```

CSEP - 行のセパレータ

<value> は文字列の式のはずです、そしてそれはお互い（列の中で）から実際の入力で列を分ける文字列を識別します。標準的な

バックスラッシュ + 文字 combo を使うことによって、ある特定の制御文字をエスケープすることができます (RSEP 参照)。

FILE

<value> は文字列の式のはずです、そしてそれは実際の書式のないファイル入力のパスを意味する文字列を識別します。

COLUMNS

<value> はそれぞれ所定の <table> の行を意味する行の名前のリストのはずです。行の名前を与える順序は重要で、そして実際の入力と一致するはずです。

SKIP

<value> は整数式のはずです、そしてそれは、他のいかなる行でも読む前に、無条件に省略されるいくつかの実際の入力ファイルの行を識別します。

COUNT

<value> は整数式のはずです、そしてそれはいくつかの列のために実際の入力に値があるであろうか識別します。もし値が列のために指定しないなら、そしてもしタイプ = 「アクセス」(行の名前が実際の入力の最初のラインに与えると仮定する) なら、カウントは必要です。

CHECK

<value> は TRUE か、あるいは FALSE のいずれかを判定するブーリアン式のはずです。TRUE はそれぞれの列が挿入するために、すべての制約のチェックを実施することを意味します。もし FALSE が指定されるなら、制約のチェックが実施されません。もしチェックが例えば重複する主キーを避けるために FALSE にセットするなら、入力 genuinely 適格であることを確認することは良い考えです。

RDQ - ダブルクォートから取り除く

<value> は TRUE か、あるいは FALSE のいずれかを判定するブーリアン式のはずです。TRUE はもし行の値がダブルクォートで括弧に囲まれるなら、それらが自動的に取り除かれるであろうことを意味します。

STOP – エラーが起こった後止まる

<value> は TRUE か、あるいは FALSE のいずれか判定するブーリアン式のはずです。TRUE は、エラーが見つかった途端に、読み込みセッションが終了するであろうことを意味します。

索引を付ける

索引の管理

```
CREATE INDEX [<index name>] FOR | ON <table name> (<column  
name list>);
```

所定の表と行のために索引を作ります。表のために作ることができる索引の番号に制限がありません。索引の定義に含まれることができるいくつかの行に制限がありません。もし索引が後でドロップするのなら、<index name> は必要です。カレントユーザは指定した表を保つスキーマの所有者のほずです。

索引があなたが待つ間に作られるとき、この文の実行は表がいくつの列を保つかによって少しの時間を要するかもしれません。

CREATE INDEX によって作成された索引がただ SELECTs を最適化するために使うだけであることに注意してください、すなわちこれらの索引は完全な制約のチェックを意味しません。

```
DROP INDEX <index name>;
```

指定した索引をドロップします。カレントユーザはインデックスを保つスキーマの所有者のほずです。

既存の SQL アプリケーションを再書き込みすることで、助けることができる便利な道具として、同様に紹介します：

```
CREATE UNIQUE INDEX [<index name>] FOR | ON <table name>  
(<column name list>);
```

これは意味的に同じです：

```
ALTER TABLE <table name> ADD [CONSTRAINT <index name>] UNIQUE  
(<column name list>) INITIALLY IMMEDIATE NOT DEFERRABLE;
```

上記の文のすべてはトランザクションを始めています、すなわち確定が他のユーザに見える変更をするために必要とします。

戦略

FrontBase は維持している索引に 2 つの戦略を提供します：

1. 持続スペース

デフォルトの戦略、呼び出した持続スペースは、非常に能率的なスペースで、そして、数十万の列まで表を使ってよく動作します。表の索引は、列の番号が、列毎に 5 バイト以下しか負担をかけないので問題はありません。もしあなたが 100,000 の列の表を持っているなら、この表で索引を作ることは、500 KB 以下にディスクの必要面積を増やさずでしょう。メモリの効率は、行の値が索引の情報（最適化した B ツリー）と一緒に記憶しないから、達成します。悪い面は、索引を使うとき、表の列が行の値を得るためにロードするはずで、この索引の方式はたいていのケースでうまく動きます。ユーザはデータベースの少ないディスク・スペースの必要面積が好きです。

2. 持続時間

多くのディスク・スペースを消費して何百万という列を検索するとき、持続時間と呼ばれる代わりの戦略は速いです。大変上手に方式の大きさを調整して、そして、何百万という列を簡単に表を処理することができます。行の値が B ツリーにコピーします、そしてそれはディスク・スペースの必要面積を増やしますが、かなり参照が速くなります。ただ必要とするときだけ、実際の列がロードします。もし所定のセレクトがただ索引の一部である行の値を取って来るだけであるなら、実際の列はまったくロードされません。セレクトは非常に速いです。

例：

単語の表、文書の表とリレーションの表を使って典型的な索引の設定を思ってください：

```
CREATE TABLE WORD(
    WORDPK INT PRIMARY KEY, -- Implies an index
    WORD VARCHAR(64));

CREATE INDEX ON WORD(WORD);
```

```
CREATE TABLE DOCUMENT(  
    DOCUMENTPK INT PRIMARY KEY, -- Implies an index  
    DOCUMENT CLOB);
```

```
CREATE TABLE RELATION(  
    WORDFK INT,  
    DOCUMENTFK INT,  
    PRIMARY KEY(WORDFK, DOCUMENTFK)); -- Implies an index
```

```
CREATE INDEX ON RELATION(DOCUMENTFK, WORDFK);
```

```
COMMIT;
```

所定の単語が見つけた文書を識別して DOCUMENTFKs のリストを得るために：

```
SELECT DOCUMENTFK  
    FROM RELATION, WORD  
    WHERE RELATION.WORDFK = WORD.WORDFK AND WORD.WORD =  
                                                '<some-word>';
```

所定の文書で見つけた単語を識別して WORDFKs のリストを得るために：

```
SELECT WORDFK  
    FROM RELATION  
    WHERE DOCUMENTFK = <some-document-pk>;
```

2つの所定の単語が見つけた文書を識別して DOCUMENTFKs のリストを見つけるために：

```
SELECT DOCUMENTFK  
    FROM RELATION, WORD  
    WHERE RELATION.WORDFK = WORD.WORDFK AND WORD.WORD =  
                                                '<word_1>'  
  
    INTERSECT  
    SELECT DOCUMENTFK  
    FROM RELATION, WORD  
    WHERE RELATION.WORDFK = WORD.WORDFK AND WORD.WORD =  
                                                '<word_2>'
```

これはビューにラップすることができました / できるでしょう。

文書毎に平均して100,000の文書と100の単語を使って合理的な設定で、リレーションの表は10,000,000の列を保持、

そして持続時間の完ぺきな候補です。上記の SELECT 文のすべてで、もし索引が持続時間にセットしたなら、実際の列はロードされないでしょう。索引は実際の行の値を保つでしょう。

単語表は持続時間のためにそれほど有望でない候補者です。この表のために適切なサイズを定めたキャッシュと結合した持続スペースを使うことはおそらくもっと多くの意味をなすでしょう。

FrontBase を調整する

あなたの FrontBase データベースが大きくなる、あるいは性能の要求が増加するとき、あなたは全体的な、そして特定のクエリのパフォーマンスを改善するために FrontBase を調整することができます。このセクションは、表と索引のキャッシュ（特定のクエリの性能）と同様、未使用のデバイスドライバーの特徴（全体的な性能）を使ってどのように FrontBase を調整するかを記述します。

データベースサーバーの性能

データベースサーバーの性能は3つのものに依存します：CPU、RAM とディスクサブシステムのスピード。

あなたのデータベースが小さい（すなわち表毎に10万の列まで）とき、性能は主に CPU のスピードに依存しています：サーバーがなんと速くデータを動かすことができることでしょうか。あなたのデータベースが大きくなるにつれて、その少しがメモリーに適しています。多くのデータベースがいつも RAM に適するにはあまりにも大きいです。キャッシュ、キャッシュする戦略、そしてキャッシュするオプションが大いに認められている性能に影響を与えます。結局は、あなたのデータベースは利用可能な RAM の量より何度も大きくなります。その時点で、データベースサーバーのスピードがディスクスピードに最も依存します：サーバーがなんと速くデータを得ることができることでしょうか。

よく、あなたのマシンをアップグレードするよりむしろ、あなたはもっと多くの RAM をインストールして、そしてディスクサブシステムをアップグレードすることができます。それがいっそう効果的に利用可能な RAM を使うことができるように、どのようにデータをキャッシュするかについて、あなたは同様にデータベースサーバーのヒントを与えることができます。

FrontBase はメカニズムをキャッシュする

FrontBase は、データベース開発者によって調整可能なそしてアクセス可能な2つの主な部品を使って複雑なキャッシュの戦略を提供します：キャッシュしている表、そして未使用のデバイスドライバー（あるいは全体的なキャッシュ）。FrontBase キャッシュの重要な品質はトランザクションによって提供した完全性が維持されるということです。キャッシュの中のデータがアップデートされる時、同じデータは確定を実行するや否や同様にディスクに書き込まれます。ただデータが正常にディスクに書き込んだ後だけ、確定は成功します。

キャッシュのまとめ

FrontBase データベースサーバーは実際に 5 つの明確なキャッシュタイプを持っています：

RDD キャッシュ

RDD（未使用のデバイスドライバー）キャッシュはファイルシステム上に存在するキャッシュを通して書き込みします。RDD キャッシュの主要な目的は根本的な記憶媒体に読み込むことを避けることです、しかしそれは同様に記憶媒体に書き込みの最適化のために基礎を作ります。すべての読み込みと書き込みは RDD キャッシュを通して行います。

デスクリプタのキャッシュ

表のために記憶したデータの実装はデータへの速いアクセスを提供するためにデスクリプタを要求します。デスクリプタはデスクリプタのキャッシュでキャッシュされます。

列のキャッシュ

それぞれの表が表の中の列の修理する部分のためにキャッシュを持っています。これらのキャッシュは頻繁に使う表がコアで存在することを保証します、そしてそれは根本的な記憶媒体から時間がかかる読み込みを避けます。

ドープ（強化する）キャッシュ

表のすべての列が同じ修理するサイズを持っているように、さまざまなサイズの行のデータは行のために実際のデータに修理するサイズの参照によって実行します。可変的なサイズの行のデータはドープ（強化する）キャッシュでキャッシュされます。

索引のキャッシュ

表のための索引は索引の方式によって、表のためにすべての索引によって共有したキャッシュで、あるいは索引毎に 1 つのキャッシュで、キャッシュされます。

列、ドープ（強化する）と索引のキャッシュはひとまとめにして表のキャッシュとして知られています。

キャッシュをいつ調整すればよいですか？

もし、しきりに SELECTs を使ってデータベースサーバーを的中させることが CPU の利用の割合を暗示的に下げるなら、あなたはキャッシュしているメカニズムを調整する必要があります。この場合、データベースサーバーはデータを届けるためにディスクサブシステムを待っています。マシンに RAM を加えることによって、あなたは簡単に性能を増やすことができるか、あるいはキャッシュすることを調整する必要があるでしょう。

FrontBase は次のことをキャッシュすることを提供します：

- 1 2 2 ページの「デスクリプタキャッシュ」。
- 1 2 3 ページの「表をキャッシュする」。
- 1 2 6 ページの「未使用のデバイスドライバ (RDD)」。

デスクリプタキャッシュ

現在 FrontBase はデスクリプタキャッシュを変更する手段を提供しません。FrontBase のデータベースファイルのページのおよそ 0.2% がファイルでのデータへのアクセスを提供するために使います。これらのページが容易に実際のデータを読み込むためにアクセス可能であることが重要です、そうでないときはシステムは最初にディスクそして次にデータからデスクリプタを読み込むでしょう。

コマンドを出すことによって、デスクリプタキャッシュの性能は sql92 コマンドラインツールで調べることができます：

```
SHOW IO DESCRIPTOR;
```

次の見出しをリストします：

キャッシュブロックを使う – ブロックの番号がキャッシュで現在使われています

自由なキャッシュブロック – ブロックの番号が現在使われていません

デスクリプタの読み込み – デスクリプタの読み込みの要求の番号

デスクリプタを的中する – キャッシュから読み込んだデスクリプタの読み込みの要求の相対的な番号

デスクリプタの書き込み – デスクリプタの番号を書き込む

カウントを読み込む – データの読み込みの番号

ブロックを読み込む – 読み込んだデータのブロックの番号

カウントを書き込む – 書き込みの番号

ブロックを書き込む – 書き込みのブロックの番号

最も重要な数字はデスクリプタキャッシュの性能を決定するための的中率です。新たにスタートしたデータベースのためにそれは大きくなって、そしてほぼ100%安定するでしょう。番号の残りがデスクリプタの読み込みと書き込みの間を行き来する感覚を与えます、そしてデータを読み書きします、従ってIOの動きが基準です。

表をキャッシュすること

表をキャッシュすることはキャッシュするために所定の表の列のいくつかを調整するために開発者あるいは管理者を可能にします：

- ・ 最小の列のカウント、キャッシュする列の最小の番号。
- ・ 最大の列のカウント、最大の番号あるいはキャッシュする列。
- ・ パーセンテージ、キャッシュする列のカウントの合計のパーセンテージ。
- ・ 持続的な、トランザクションにキャッシュを取っておく。
- ・ 前もってロード `syry`、表が参照される最初るとき、表をキャッシュする。

若干のシナリオで、あなたが、メモリー処理のある特定の邪魔が起こることを可能にする間に、コントロールに実際のメモリー処理を望むだろうから、持続的な設定は利用可能（そして常に実行されません）です。事例はレポートの作成です、そしてそれぞれのトランザクションが多くのSELECTsに及ぶことができます、そして多分、通常それほど使っていない若干のテーブルを参照するでしょう。もし持続することがそのような表を消すためにセットするなら、トランザクションを確定するとき、キャッシュは使えるようにロードして、そして流すでしょう。

いつ表をキャッシュすることをえばいいですか？

表をキャッシュすることはメモリー処理の展開なしで 50% から 100% キャッシュすることができるより小さい「フィーダーテーブル」で一般に使います。

パフォーマンスインジケータ

コマンドを出すことによって、表のキャッシュの性能は sql92 コマンドラインツールで調べることができます：

```
SHOW CACHE[<schema-name>.]<table-name>;
```

注：名前はワイルドカード %_ が使えます。

もし <table-name> を省略するなら、現在のスキーマですべての持続した表が示されます。SHOW CACHES コマンドは次の見出しをプリントするでしょう：

タイプ — 列のキャッシュは R、ドープ（強化する）キャッシュ、D、索引のキャッシュを最適化したスペース、S、そして、索引のキャッシュを最適化した時間、T のマークが付いています。

項目名 — 項目の名前。列とドープ（強化する）キャッシュのためにそれは表の名前です、索引キャッシュのためにそれは索引の名前です。他の SQL 文で使うとき、_（下線）で始まっている名前は”（ダブルクォート）で囲まれるはずであることに注意してください。

項目 — 構造体の項目の合計の数。列のキャッシュのためにそれは表での列の番号です、ドープ（強化する）キャッシュのためにそれは唯一の文字列の番号です、そして索引のためにそれは索引を実行するために使うページの番号です。

実際の — キャッシュの現在の項目の番号。

バイトサイズ — キャッシュのために使う RAM のバイト数。

最大の項目 — キャッシュを保つ項目の最大数。個々の項目のサイズが変化し、そしてそれを省略するとき、この数はドープ（強化する）キャッシュのために関連していません。

バイトサイズ – 完全にいっぱいになったキャッシュが使うバイトの推定番号。ドープ（強化する）キャッシュのためにこの番号は、同様に、キャッシュがその方法で制限されるキャッシュの限界サイズです。

ルックアップする – 項目をアクセスした時間の番号。

的中する – 項目がキャッシュの中で見つけた参照の数。

的中率 – キャッシュで見つけた参照の相対的な番号、従って1つ以上の読み込みオペレーションを引き起こしませんでした。いっぱいではなかったキャッシュに項目をロードしたキャッシュの参照が的中するとして数えられます。これはキャッシュをロードすることは、それがいっぱいになるまで、ヒットするという結果になります。ロードするの間の的中率は100%でなるでしょう。

Refs（リファレンス）（ドープ（強化する）キャッシュのみ） – ドープ（強化する）が唯一の文字列を記憶し、それぞれの時間に存在する文字列が1度だけ記憶されるでしょう。。Refs はドープ（強化する）キャッシュの文字列に参照の合計の数を与えます。

比率（ドープ（強化する）キャッシュのみ） – 1つの唯一の文字列にリファレンスの平均の番号を与えているリファレンスと項目のカウントの間の比率。もし比率が1に近いなら、それは同一の文字列の複製を避けるためにメカニズムを止める考慮をする価値を持っています。

リストを別としてそれぞれのキャッシュのために、合計のラインが実際のメモリ処理のために計算されます、そしてそれは、キャッシュがどれぐらいメモリを使っているかについて、考えさせるでしょう。

表キャッシュの設定を調整する

sql92 のコマンドラインツールを使って、あなたは FrontBase によって提供された SQL 92 の次の拡張子を使って表のキャッシュを調整することがあります。あなたは同様に FrontBaseManager と FBWebManager ツールを使って表のキャッシュの設定を調整することができます。

```
ALTER TABLE <table name> SET CACHE ([<lower>], [<upper>],  
                                         [<percent>]);
```

所定の表のためにキャッシュのパラメータをセットするか、あるいは調整します。<lower> と <upper> が絶対的な列のカウントとして与えられます。サーバーは最小の <lower> 列と最大の <upper> 列を

メモリに取っておくでしょう。デフォルトは <lower> のための 2000 と <upper> のための 20,000 です。<percent> 値はサーバーに、まだ <upper> 値に従う間に、キャッシュの列のさまざまな番号を取っておくように言います。<percent> のためのデフォルト値は 20 です。

```
ALTER TABLE <table name> SET CACHE PRESERVE FALSE | TRUE;
```

たとえ表へのリファレンスがないとしても、所定の表のためにキャッシュを維持する (TRUE に) サーバーに指示します。FALSE は所定の表に、もうリファレンスがないとき、サーバーがキャッシュを捨てることを意味します。

```
ALTER TABLE <table name> SET CACHE PREPARE FALSE | TRUE;
```

表を参照する最初するとき、所定の表のために十分にキャッシュをロード (TRUE に) するためにサーバーに指示します。所定の表の実際の使用によって要求するように、FALSE はキャッシュがロードされることを意味します。

上記の文はトランザクションを始めています。確定が他のユーザに見える変更をするように要求します。

未使用のデバイスドライバー (RDD)

今日の速いコンピュータを使って、データベースサーバーのたいていの性能の問題はどのくらい速くハードディスクからデータを得ることができるか関連しています。この範囲で、性能を増やす1つの方法はホスト OS ファイルシステムを回避することです。FrontBase でこれは生のデバイス・ドライバ (RDD) モジュールの実装を通してします。RDD は datastore として使う生のパーティションを指定することを可能にします。さらに、RDD はその結合した書き込むことを通じて、そして読み込みのキャッシュの大きさを指定することを可能にします。

RDD はいつ使えばいいですか？

より小さい表が 100% キャッシュするように、RDD は表をキャッシュすることと共に一般に使うのに対して、より大きい表が RDD キャッシュによってキャッシュされます。事例はインデックスを付けているソリューションです。あなたは 3 番目のより大きい表がリレーションの表、的中する状態で、一般に 2 つのより小さい表、単語と文書を持つでしょう。FrontBase で、RDD に的中した表を管理させる間に、あなたは 100% 単語とドキュメントをキャッシュする

でしょう。100 - 300 MB の RDD キャッシュメモリが、的中した表のサイズに依存して、よく動きます。

RDD の設定を調整する

データベースを作るか、あるいはスタートしたとき、RDD の設定が指定されます。FrontBaseManager と FBWebManager 両方がそれぞれの「Start Advanced (先にスタートする)」コマンドを使って RDD 設定を指定します。コマンドラインから、FrontBase プロセスをスタートするとき、あなたは RDD 設定を指定します。

RDD パフォーマンスインジケータ

コマンドを出すことによって、RDD キャッシュの性能は sql92 のコマンドラインツールで調べることができます：

```
SHOW IO;
```

次の見出しがリストされています：

使ったページ - 現在使用中のページの番号

空いたページ - キャッシュの中の空いたページの番号

未処理の pi - 未処理の入力のページの番号

未処理の po - 未処理の出力のページの番号

未処理の ps - 未処理の同期化の番号。この番号は成功して、そしてトランザクションログに書き込まれた、しかしデータベースファイル自身ではない確定の番号を判断します

読み込みのカウント - 読み込みの要求の番号

読み込みのブロック - ブロックの読み込みの番号

読み込みを的中する - キャッシュから読み込んだ読み込みブロックの相対的な番号

読み込みを待つ - 利用可能になるためにブロックを待たなければならない読み込みの番号

読み込みが行き詰まる – ディスクからデータを読み込んで利用可能になるバッファを待たなければならない読み込みの番号

書き込みのカウント – 書き込みの要求の番号

書き込みのブロック – ブロックの書き込みの番号

書き込みが行き詰まる – バッファが利用可能になる書き込みを待たなければならない時間の番号

デバイスの読み込み – ディスクからの読み込みの実際の番号

デバイスを書き込み – ディスクへの書き込みの実際の番号

キャッシュが静的に割り当てられるとき、キャッシュのページの番号が空き、そして使ったページの合計に一定で、そして等しいです。主要なパフォーマンスインジケータは RDD キャッシュの性能の的中率です。より高い比率で、ディスクがあまり活動していません。RDD にもっと多くのメモリーを割り当てることによって、あるいは表のキャッシュの大きさを増やすことによって、的中率は改善します。行き詰まることは 0 になることでしょう；もしそれらがそうでないなら、キャッシュがデータの未処理の出力でいっぱいになっている、そしてキャッシュをより大きくすることが良い考えであることを意味します。

パフォーマンスを改善する

我々は今種々のキャッシュのパフォーマンスのためにすべての番号を持っています。しかしパフォーマンスを改善するための可能性のインジケータは何ですか？

最初に思いつくことはすべてのキャッシュを完成した根本的なデータ構造に受け入れることに関して大きくするはずですが。それは表のキャッシュのレベルとなり、そしてそれは RDD キャッシュになります。もしあなたがサーバーを根本的なファイルのサイズより大きい RDD キャッシュを使って始めるなら、既存のデータは正確にいったん読み込んだら、再び決して読み込まれないでしょう。修正したページをちょうどファイルに書き込んで、そしてキャッシュに取っておきます。あなたは同様に完全な表をキャッシュするサイズについて表のキャッシュをセットすることができます；これは表が開くたびに再構築しない表のための構造としてもっと効率的です。

上記の設定は、十分な RAM を使ってマシンの小さいデータベースのために効率的であるでしょう、しかしマシンがページングし始めると

すぐに、性能は劇的に下がるでしょう、そしてさらにもう少し精巧なスキームが所定のハードウェアコンフィギュレーションの最大の性能を得るために使うはずでず。ほとんどの場合、それがただ、ページングシステムがページを書いて、そして新しいものを読み込まなければならぬに対して、データを再読み込みすればよいとき、FrontBase はページングシステムより良く実行するでしょう。

あなたはめったに用いられた表ではなく表のキャッシュを避けるでしょう。所定のキャッシュの参照の数をざっと調べることによって、あなたはこれを決定して、そしてもしそれが少ないように思うなら、表の維持を止めることができます。

表のデータはその時 RDD キャッシュによってキャッシュされます。あなたは同様に少ない的中率で表を探すことができました。的中率はテーブルのためにキャッシュが受け入れるより多くの項目を使っても 100% 以下しか達成できません。非常に低いの中率は表のスキャンが実行することを提案します、それは、完全な表が返される場合、唯一のすることがキャッシュのサイズを増やすことであるか、あるいは、索引が表に対して実行したクエリの最適化のために要求する場合、表のスキャンの理由を切り離すはずでず。低いの中率とは一般に 20% 以下でず。

的中率を改善することによって、表のキャッシュをもっと効率的にするとき、RDD キャッシュに行かない表のキャッシュによって満たされた読み込みとして、あなたは同様に RDD キャッシュをもっと効率的にするでしょう、従って持続されていないキャッシュを読み込むことはもっと効率的でず。コストとは表のキャッシュが RDD キャッシュより多くの RAM を使うということです。

Mac OS X と未使用のデバイス

FrontBase は、生のディスクデバイスで、直接機能することができます、従ってファイルシステムとバッファのメカニズムを回避します。ファイルシステムのオーバーヘッドが避けられ、そしてディスクに関するより素晴らしいコントロールが目的を達するとき、未使用のデバイスの使用は FrontBase のパフォーマンスと信頼性を拡張します。

未使用のデバイスを作る

望ましいサイズ、そして FrontBaseFS のタイプとあなたが作成することを望むデータベースの名前を使ってディスクにパーティションを作成してください。タイプは FrontBaseFS のはずでず、しかし名前は主に pdisk と類似のプログラムによって使われます。ディスクにパーティションを作成するために pdisk ユーティリティを使いました。

未使用データのパーティションを FrontBase にとってアクセス可能であるようにするために、シンボリックリンクを作ってください：

```
ln -s /dev/disk<x>s<y> /Library/FrontBase/Databases/  
<database-name>.fb
```

<x> がデバイス番号で、そして <y> がスライス番号です。スライス番号は、`pdisk L` コマンドによってリストされるパーティションの番号です。<database-name> は FrontBase データベースにアクセスするために使うはずである名前です。

未使用のデバイスで FrontBase データベースを作る

FrontBase サーバーの未使用デバイスに FrontBase データベースを初期化することは、既存ファイルの内容に上書きすることを FrontBase に可能にする `-create` オプションを使ってスタートするはずですが、データベースを初期化した途端に、FrontBase は通常の方法で使われるかもしれません。しかし、`-rdd` オプションが適切な未使用のデバイス・ドライバキャッシュを指定するために使うことを暗示します。

メモリーの使用法

あなたが使うことを望むかもしれないもう 1 つの統計上のコマンドがあります：

```
SHOW MEMORY [ALL];
```

次の見出しをリストします：

The name – 現在我々は 2 つの別個のメモリゾーンを持っています：IO、IOZone のために、そして他のすべて、DefaultZone のために

VM – ゾーンに割り当てる MB の数

Small-Large – メモリの大きい部分と小さい部分に割り当てる MB の数

Used-Free – 使われた小さい部分と空いた小さい部分のための MB の数。小さい部分に割り当てられた VM が 10 MB 以上であるとき、一般に空いた小さい部分は小さい部分に割り当てたスペースのおよそ 5 - 10% でしょう

Used-Free (2回目の) – 使った、そして空いたメモリの一部の数

Used-Free (3回目の) – 使った、そして空いた小さい部分の平均のサイズ

IOZone の合計サイズは RDD のサイズに近いです。使われた小さい部分のサイズは数 MB で一般にキャッシュの合計サイズに近いです

もし使われた小さい部分の数が大きくなっているなら、表のキャッシュは一般にまだいっぱいになっていません。もし合計の割り当てがマシンの物理的な RAM に近いなら、キャッシュの設定をよく調べるか、あるいはもっと多くの RAM をインストールすることは良い考えでしょう。

移行

このセクションは他のベンダーのデータベースサーバーからデータベースを移行させるために、FrontBase の読み込み機能の使用方法を説明します。

現在 FrontBase は次の読み込みのメカニズムを持っています：

- 1 3 2 ページの「FileMaker」。
- 1 3 3 ページの「MySQL」。
- 他の読み込みメカニズムは 1 1 1 ページの拡張した読み込み機能「拡張した書式のないファイルの読み込みと書き出しの関数」によって利用可能です。

FileMaker

FileMaker Pro の移行ツールはあなたに FrontBase で FileMaker Pro データベースを動かします。それは 2 つのキーの制限を持っています。最初に、FrontBase が初期段階のツールではなく、リレーショナル・データベースサーバーであるとき、ただあなたの FileMaker Pro テーブルだけが移行するでしょう。第二に、SQL 9 2 は、計算した行のための準備を持っていません、それで、もし FileMaker Pro データベースでそれらを持っているなら、それらは FrontBase へと移動することができません、しかしながら、それらはビューに実装することができます。

現在、あなたは別に FileMaker Pro 移行ツールをダウンロードする必要があるでしょう。それは MacOS X のプラットフォームだけ利用可能です。ツールは FrontBase Web サイトのダウンロードのセクションからダウンロードできます。

あなたが FileMaker Pro 移行ツールをダウンロードして、そして圧縮解除するとき、あなたは次の項目を見つけるでしょう：

1. FM2FB.app

これは Mac OS X サーバーでビルドする主なアプリケーションです。それはすぐに使うことができます、あるいはそれはアプリケーションフォルダーの 1 つに動かすことができます。アプリケーションは FrontBase クライアントのフレームワークで走ることを必要とします。あなたが FrontBase パッケージをインストールするとき、これらのフレームワークはインストールされています。

2. MetaDumper.fp3

この FileMaker スクリプトは FileMaker ファイルから情報を引き出します。このスクリプトは FileMaker がそれにアクセスすることができる場所に動かすでしょう。それは FileMaker の MacOS と Windows バージョンの両方で作動します。MacOS においては、スクリプトファイルをダブルクリックすることが可能ではないかもしれませんが。その代わりに、FileMaker をスタートして、そして "Open..." メニューアイテムを使ってそれを開いてください。

3. リードミー

もっと古い「リードミー」ドキュメント。

FM2FB と MetaDumper スクリプトを使う方法についての文書にアクセスするために、FM2FB.app をスタートして、そして「ヘルプ」メニューから「FM2FB Help」を選択してください。これは Web ブラウザで HTML 文書を開くでしょう。HTML ファイルはアプリケーションラッパーに組み込まれています。

MySQL

MySQL の移行ツールはあなたに両方のデータベースのために JDBC ドライバーを使って FrontBase で MySQL データベースを動かします。それは 1 つのキーの制限を持っています。それは enum を使って MySQL を処理する、あるいは行のタイプをセットすることができません。MySQL の移行ツールは Java バージョン 1.2 あるいはそれ以降を必要とします。

現在、あなたは別に MySQL の移行ツールをダウンロードする必要があるでしょう。ツールは FrontBase Web サイト (www.frontbase.com) のダウンロードのセクションからダウンロードできます。結局、それはすべてのプラットフォームのために FrontBase Web サイトでそれぞれダウンロードしなければならないでしょう。

あなたは MySQL のために JDBC ドライバーを得る必要があるでしょう。MySQL の移行ツールの残りが入っているディレクトリで MySQL JDBC ドライバーをインストールしてください。

MySQL の移行ツールを走らせるために、あなたの作業ディレクトリをツールに入っているディレクトリに変更して、そしてコマンドラインから次のことを実行してください：

```
java -cp mysql_2_comp.jar:frontbasejdbc.jar:mysql2fb.jar MySQL2FB
```

それは若干の情報のためにあなたに移行を完了するように頼むでしょう：

ソース

ソースフィールドは転送するための MySQL データベースの場所と名前です。入力はこのフォーマットを持つはずで：
dbName@host。

行き先

行き先のフィールドは FrontBase データベースの望ましい場所と名前です。指定したホストに FrontBase 2.0 の走っているバージョンがなければなりません。入力はこのフォーマットを持つはずで：
dbName@host。

ユーザ名

ユーザ名は MySQL データベースにログオンするために必要なユーザ名です。

パスワード

パスワードは MySQL ユーザ名のためのパスワードです。

読み込みフィルターは間違っ**て**フォーマットした入力に容赦しません。製造のデータベースを動作する前に、テストデータベースを使って実験してください！

トラブルシューティング

SQL 文のログを書き出すこと

FrontBase はあなたにサーバーに送った、そして実行したすべての SQL 文のログを書き出すことをイネーブルすることを認めます。これは順番にただ強力なストレステストツール (Client Simulator) だけではなく、同様に非常に重要なデバッグの探査機を可能にします。最後に、決して軽んじるべきでないこととして、ログを書き出すことは同様にサーバーに課したさまざまなクライアントを変更する状態の原文の表示を使ってあなたに提供します。

FrontBase 3.1 の前に、サーバーをスタートするとき、SQL ログを書き出すことが作動できる状態にしていなければなりません：

```
<install dir>/FrontBase/bin/FrontBase -logSQL [<other options>]  
<database name> &
```

3.1 とそれ以降のバージョンのリリースで、SQL ログを書き出すことは同様に通常の SQL 文で意味するものよりもっとダイナミックな方法でコントロールすることができます：

```
SET WRITE SQL { TRUE | FALSE } [ GLOBAL ];
```

この文の文法のいろいろな組合せは下に記述します。

```
SET WRITE SQL TRUE GLOBAL;
```

全体的な基礎にログを書き出すことをするでしょう、すなわち、もしそれが存在しないなら、ログファイルを作って、そしてすべての新しいエージェントの接続のためにログを書き出します。既存の接続がログを書き出さないことに注意してください。

```
SET WRITE SQL FALSE GLOBAL;
```

すべての既存の、そして新しいエージェントの接続のためにログを書き出すことを止めるでしょう。実際のログファイルがクローズされないことに注意してください。これはもしログを書き出すことが再び後で作動しているなら、ログファイルが（例えばログファイルディレクトリがオフ / オンの間に作られた場合）場所を変えないことを意味します。

```
SET WRITE SQL TRUE;
```

実行しているエージェントの接続のために、しかし、もしログを書き出すことが全体的なオプションを使って最初に作動していたなら、ログを書き出すことをオンにするでしょう。

```
SET WRITE SQL FALSE;
```

実行しているエージェントの接続のためにログを書き出すことを止めるでしょう。

ログを書き出すことを可能にするとき、実際のファイルが取り除かれない、すなわち新しいログの項目が付加されることに気付いてください。ログを書き出すことをすることは、全体的に、サーバーのストップ/リスタートの循環に及ばないでしょう。

SQL ログファイルの場所

ログファイルは次のようにデフォルト毎に作られるでしょう

```
<install dir>/FrontBase/Databases <database name>.fb.sql
```

FrontBase のインストールディレクトリでログファイルと呼ばれるディレクトリを作成することで、この場所を変えることができます：

```
mkdir <install dir>/FrontBase/LogFiles
```

それらをストップして、そしてリスタートするまで、開いているログファイルのサーバーがこの場所の変更を受け取らないでしょう。

ログファイルのディレクトリが実際にファイルシステムの他のどこかに存在するディレクトリへのソフトな（シンボリックな）リンクができることに注意してください。

新しい SQL のログファイル

SQL のログファイルが長い間にかなり大きくなることができます、そしてファイルの数でログを分割することは有利にできます。サーバーは 1) 既存のログファイルを改名する、そして 2) 新しいログファイルを作るように指示することができます：

SWITCH TO NEW SQL LOG;

時刻表示が接尾辞として加えられること以外、改名されたログファイルの名前は同じのままです：

```
<database name>.fb.sql.yyyymmddhhmmss
```

新しいログファイルの名前は、スイッチが作られる前と同じであるでしょう、すなわちディレクトリーの変更はなしです。

FrontBaseManager

FrontBaseManager はあなたにスタートして、ストップして、監視して (Monitor)、作って、取り除かせるアプリケーションです、そして一般に MacOS の FrontBase データベースのすべての外観を管理します。FrontBaseManager はすべてのデータベースの維持管理を行って、表とビューを作って、データベースからデータを検索して、そして SQL プログラミングのために、すべて必要でないあなたのデータベースにデータをアップロードするためにあなたにクリーンなグラフィカルインターフェースを与えます。FrontBaseManager for WinNT/2000 はまもなく利用可能になるでしょう。



この章は次のセクションを持っています：

1 3 8 ページの「FBManager からの変化」。

1 4 1 ページの「データベースを監視して (Monitor) 、そして管理する」。

1 5 1 ページの「接続ウインドウ」。

1 5 2 ページの「SQL インタプリター」。

1 5 4 ページの「データベース」。

1 5 5 ページの「セッション」。

1 5 6 ページの「使い方」。

1 5 8 ページの「ライセンス」。

1 6 0 ページの「ユーザ」。

1 6 1 ページの「スキーマ」。

1 6 2 ページの「スキーマオブジェクト」。

1 8 9 ページの「表のキャッシュ」。

1 9 1 ページの「ブラックリストとホワイトリスト」。

1 9 4 ページの「バックアップ」。

1 9 4 ページの「奇数の n の終わり」。

1 9 5 ページの「プリファレンス」。

1 9 6 ページの「ファイルの読み込み」。

2 0 2 ページの「SQL ログ」。

FBManager からの変化

FrontBase のデータベース管理アプリケーションの最近の改訂版にようこそ！ このセクションは前のデータベース管理アプリケーション (FBManager として知られている) から経験豊かな FrontBase 開発者にどこが変更したのかを示し、そしてあなたに FrontBaseManager の特徴を教えましょう：

- ・ **新しい文書**

新しいアプリケーションの文書は FrontBaseManager のヘルプメニューから利用可能です。

- ・ **マルチスレッド**
FrontBaseManager は今、デッドロックの可能性がない、同じデータベースに、多数の、同時の接続を可能にして、マルチスレッドです。ユーザがフェッチの間に表あるいは列のロックに遭遇するとき、これは同様にメインアプリケーションスレッドがバインドされるのを阻止します。
- ・ **アクアユーザインタフェース**
インタフェースはアクアヒューマンインタフェースガイドラインに適しているデザインに変更しました。そして、我々はアプリケーションにスパイスを利かせるために若干の美しい新しいアイコンを持っています！
- ・ **拡張した接続パネル**
新しい、統合化されたデータベースの接続パネルはデータベース名と TCP ポートの両方によって接続をサポートします。
- ・ **トランザクションの状態**
トランザクションの情報はデータベースの接続ウィンドウで常に利用可能です。トランザクションが、アイソレーションレベル、ロックの仕方、アップデートする方法（あるいはアクセスモード）と確定モードのような SQL 9 2 トランザクションセッティングと同様、進歩にあるかにかかわらず、これはインクルードします。
- ・ **拡張した SQL インタプリタ**
SQL インタプリタは役立つキーワードを強調することを提供して、ヒストリーリストのコピー（コマンド C）をサポートして、そして SQL テキストエリアの中にドラッグするファイルをサポートします。コンテキストメニューが、SQL インタプリタのテキストエリアを右クリックすることによって、利用可能にし、アイソレーションレベル、ロックの仕方、代わる代わる自動的に確定すること、など、変化するような動作への速いアクセスを提供します。
- ・ **容易なデータベースの情報をアクセスする**
データベースパネルは容易にデータベースバージョン、稼働時間、ディスクのデータベースのサイズ、データベースがある方式などのような、データベースについての基本的な情報を提供します。
- ・ **SQL 9 2 セッションのカスタマイゼーション**
セッションパネルを使って、あなたはセッションと結び付けられたトランザクションの設定を配列することができます。

例えば、あなたがもっと寛容なアイソレーションレベルとロックの仕方、あるいは、読み込み専用モードに、イネーブル/ディスエーブルの自動確定を切り替える、などをすることができます。

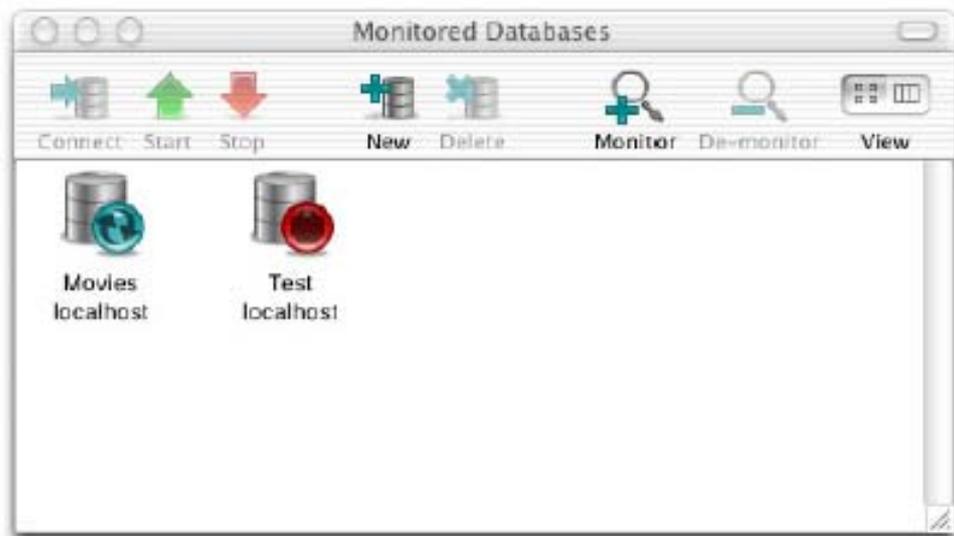
- ・ **拡張した使い方の情報**
使い方のパネルはそれらの SQL 9 2 トランザクションの設定、接続時間、トランザクションの継続時間などのような、接続したユーザについてもっと多くの情報を提供します。
- ・ **ライセンスを受けることが解明される**
ライセンスを受けるパネルはデータベースライセンスについての詳細な情報を示します。付加的な特典として、あなたは間接的にデータベースサーバーにライセンスすることができます。
- ・ **スキーマオブジェクトのブラウザ**
スキーマオブジェクトのブラウザパネルを使って、あなたはデータベースの中で、表、ビュー、手順、関数、照合などを含むオブジェクトのすべてを否定することができます。いかなる点でも、あなたは容易に新しいスキーマオブジェクトを作成するか、あるいは既存のスキーマオブジェクトの内容あるいは定義をブラウズすることができます。マウスの右クリックを使うことはクリップボードに選択したオブジェクトの完全修飾名（fully qualified name）を速くコピーするためにコンテキストメニューを出すでしょう。
- ・ **強力な表のエディタ**
グラフィックモデラー（modeler）ではない間に、FrontBaseManager の表のエディタは、少数のマウス・クリックを使って表についてあなたにほとんどすべてを変更することを可能にして、表の作成に関して大きいコントロールを提供します。それが、表の定義を変えるとき、リアルタイムに ALTER SQL を提供するので、行のエディタは特に役立ちます。これは所定の変更のために何が起きようとしているか知っていることを保証するのに助けとなって、そしてもう少し多くのプロセスでの SQL 9 2 と FrontBase について教えます。
- ・ **いっぱいテキストに索引を付ける**
表の定義エディタを使って、あなたは速く全文に強力な LookSee というインデックスを付けるエンジンを配列して、そしてデータのためにいっぱいテキストの索引を作ることができます。索引を付けているエンジンはテキスト、HTML と XML を処理することができます、そしてステミング（*）こと、大文字小文字を無視する、

トリミングの索引など、いくつかのオプションを持っています。

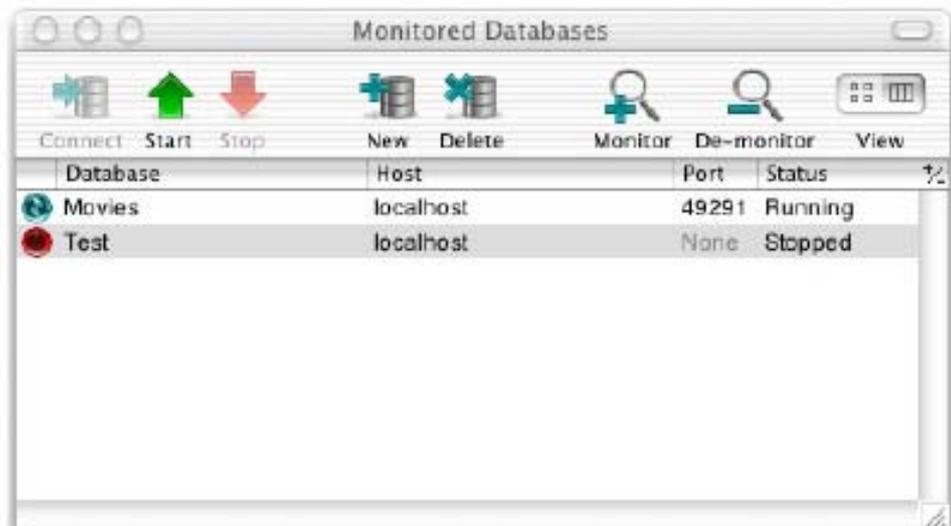
- ・ **強力な表のコンテンツエディタ**
コンテンツエディタを使って、あなたは表のデータをブラウザして、列を挿入して、列をアップデートして、そして列を削除することができます。あなたは挿入しさえすれば、そして、簡単な hyperlink をクリックすることによって、BLOB と CLOB オブジェクトをダウンロードすることができます！
- ・ **ドラッグアンドドロップ**
あなたはコンテンツエディタでのあるいはフェッチ結果のウィンドウの列を選択して、そしてデスクトップに、あるいは TextEdit、メール、エクセルと Word のようなアプリケーションにそれらをドラッグすることができます。
- ・ **容易なバックアップ**
バックアップパネルはマウスのクリックを使ってバックアップの機能性を提供します。バックアップが常に良い考えです！
- **自在に使うファイル読み込みの補助**
いろいろな異なったファイル形式から読み込みデータへの強力なグラフィックインタフェース。あなたは容易に FrontBase、OpenBase から作られたデータ、マイクロソフト・アクセス、マイクロソフト・エクセルあるいは限定したファイル形式に書き出すどんなプログラムでもロードすることができます。

データベースを監視して (Monitor)、管理する

FrontBaseManager は FrontBase サーバーの状態を監視して (Monitor)、そしてコントロールします、すなわちそれらが走っているか、あるいはストップしたかにかかわらず。FrontBaseManager をスタートするとき、次の「監視した (Monitor) データベース」ウィンドウが現われます：



このウィンドウは、データベースの名前とデータベースが位置しているホストコンピュータを使って一緒にデータベースの状態を示しているアイコンとして監視した (Monitor) データベースを示します。上記の事例はホスト localhost でデータベースムービーとテストを示します。Movies データベースは走っています、そしてテストデータベースを止めます。交互に、あなたはビューボタンを使って一覧表示に切り替えることができます。対応する一覧表示のウィンドウは次のように見えます：

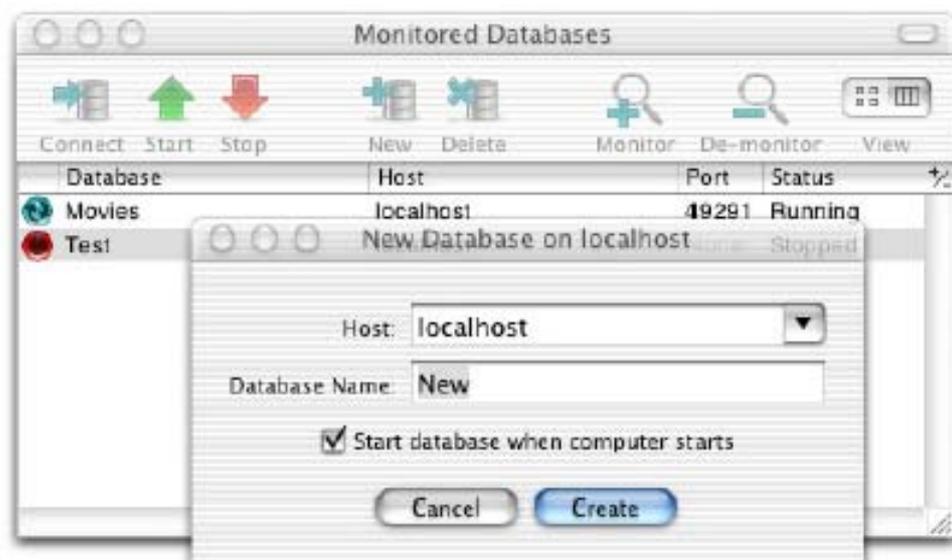


FrontBaseManager は最後にアイコンビューあるいは一覧表示を使ったかどうか覚えているでしょう。

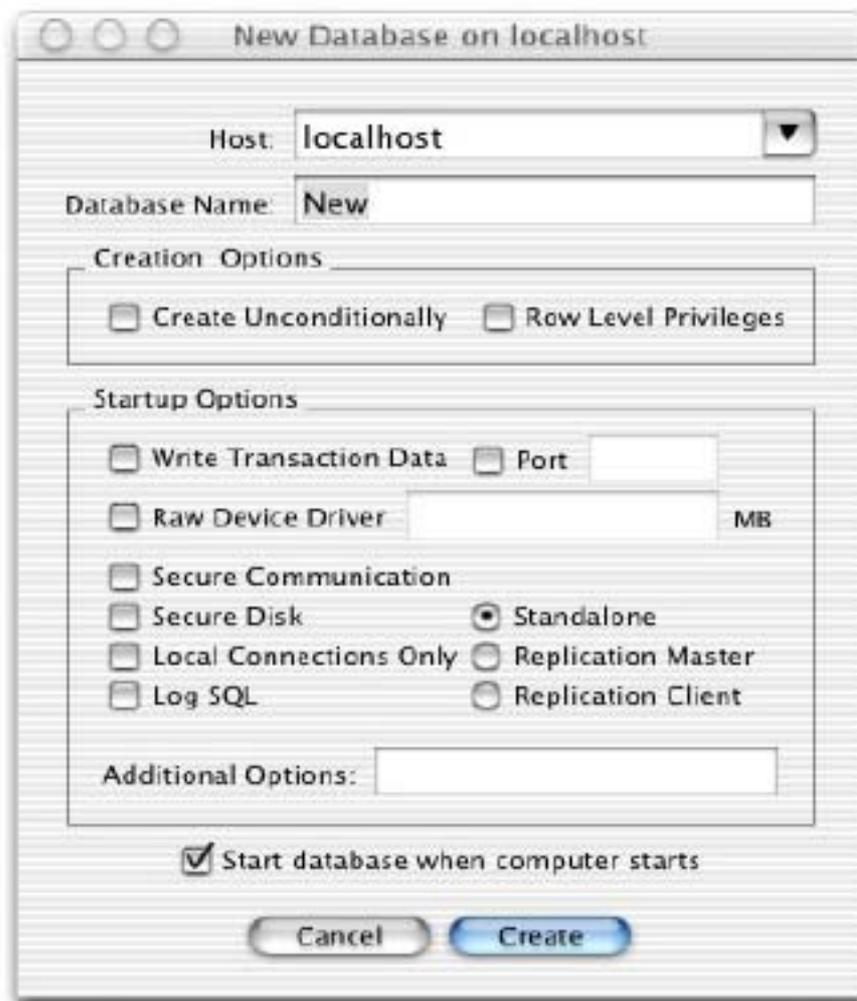
データベースを作る

新規 (New) と削除 (Delete) ボタンはあなたが新しいデータベースを作るか、あるいは既存の、選択したデータベースを削除することを可能にするでしょう。あなたがストップしたデータベースを選択しなかったなら、削除 (Delete) ボタンはイネーブルではないでしょう。あなたが削除 (Delete) をクリックするとき、FrontBaseManager はあなたにデータベースを削除することを望むことを確認するように要求するでしょう。

あなたが新しくクリックするとき、FrontBaseManager はあなたがどのホストでデータベースを作成することを望むか、そして何をデータベースと命名するべきか尋ねてあなたにダイアログ・ボックスを与えるでしょう。そのウィンドウはこれのように見えます：



File->New Database Advanced オプションを使って、あなたは追加のオプションを使って新しいデータベースを作成することができます。すべての利用可能なオプションの詳細な説明について、249ページの「デベロッパーのための FrontBase」を見てください。少数のオプションがここで論じられています：



無条件で作る (*Create Unconditionally*) – (もしそれが存在したなら) 新しいデータベースが同一名を使って既存のデータベースに上書きして、無条件に作ります。警告でこのオプションを使ってください!

列レベルの特権 (*Row Level Privileges*) – FrontBase はデータベースの表でそれぞれの個々の列に、Unix ファイルシステムのファイルのように、特権を割り当てることを可能にする列レベルの特権と呼ばれる唯一の特徴を提供します。列レベルの権利 (*Row Level Privileges*) チェックボックスは、それが作られるとき、データベースのためにこの特徴になります。どのようにこの特徴を使うかについて、もっと多くの情報のために 277 ページの「列レベル特権」を参照してください。

書き込みトランザクションデータ (Write Transaction Data) – バックアップから復帰した後に、あなたはトランザクションがもしトランザクションデータを書き上げたならバックアップから「前進」を実行することができます。

ポート (Port) – ポートを指定することは FrontBase データベースに特定のポート番号を聞くように指示します。

未使用デバイスドライバー (Raw Device Driver) – FrontBase は非常に進歩した書き込みを通してデータの記憶場所として同様に未使用のデバイスの使用 (パーティション) をサポートするキャッシュメカニズムを提供します。もしあなたがこのオプション (そしてあなたのライセンスがそれを許可する) を使いたいなら、未使用のデバイスドライバー (Raw Device Driver) をクリックして、そしてこのキャッシュの大きさを指定してください。

ローカル接続のみ (Local Connections Only) – このオプションを指定することによって、FrontBase のデータベースがデータベースと同じコンピュータで走っているクライアントから接続を受け入れるでしょう。(デフォルトは、ローカルなクライアントと同様、ネットワークで結ばれた接続を受け入れるはずです)

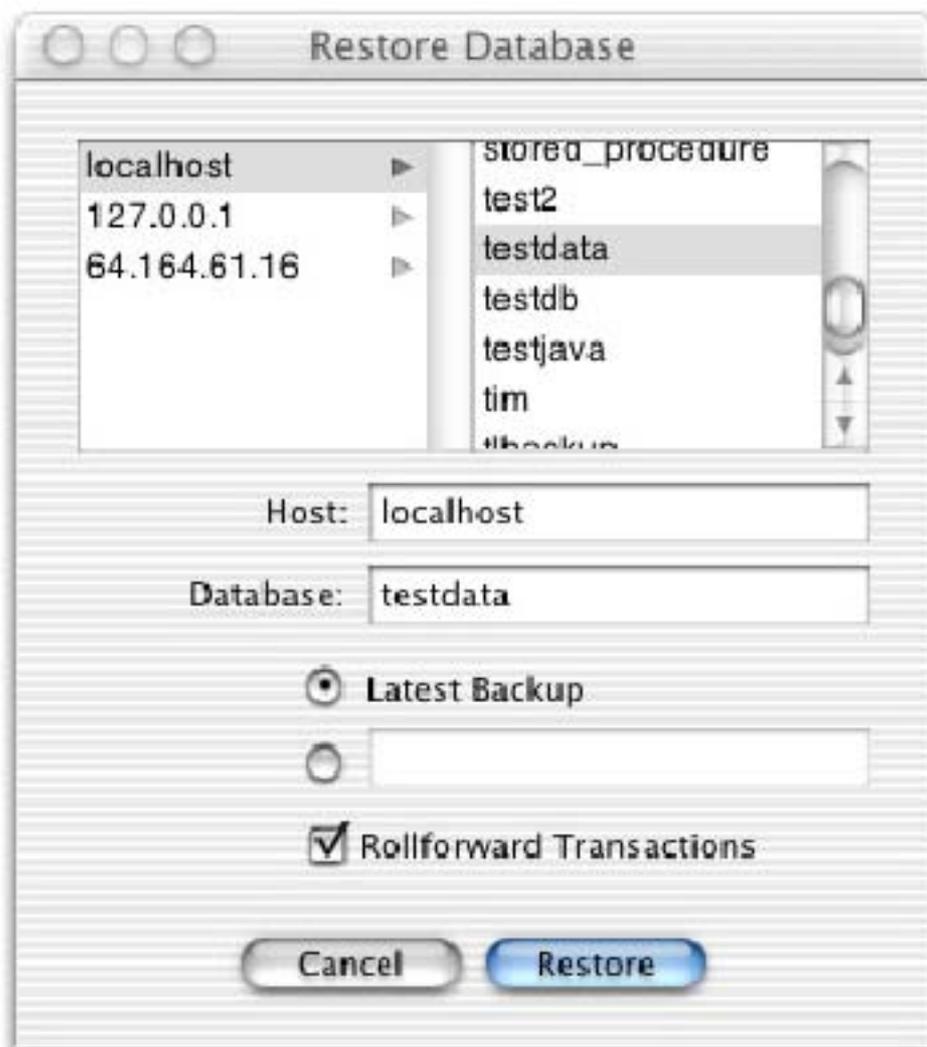
SQL をログファイルに書く (Log SQL) – このオプションを指定することによって、FrontBase のデータベースは SQL 文のファイルを受け入れたすべてをログファイルに書くでしょう。ファイルは FrontBase のインストールのデータベースディレクトリで作成されて、そして <dbname>.fb.sql と名付けます。(デフォルトは SQL 文をログファイルに書けません。) 例えば、あなたはムービーという名前のデータベースで「SQL をログファイルに書く」ことができます、そして (Mac OS X で) あなたのログは /Library/FrontBase/Databases/ Movies.fb.log にあるでしょう。

複製のオプション (Replication option) – FrontBase によって提供された複製のスキームで、あなたはすべてのアップデートが行われるはずである原版のデータベースを持つことができます。読み込み専用の複製クライアントの数を加えることができます。あなたはこの新しいデータベースが New Database Advanced ウィンドウによって独立型 (standalone) のサーバー、複製の原版 (replication master) あるいは複製クライアント (replication client) としてスタートするかどうか指定することができます。

バックアップから復帰する

バックアップから復帰することは FrontBaseManager を使えば簡単です。あなたがデータベースのバックアップを復帰することを望む 2つの事例があります。1つは、あなたは偶然にあなたのデータベースを完全に削除して、そしてバックアップから復帰する必要があります。2つ目は、あなたがまだデータベースを持つが、ちょうどバックアップで現在の内容に上書きすることを望むということであるで

しょう。それで、もしあなたがすでにデータベースを持っているなら、それをストップすることを確かにして、そして次に File->Restore Database を選択してください。

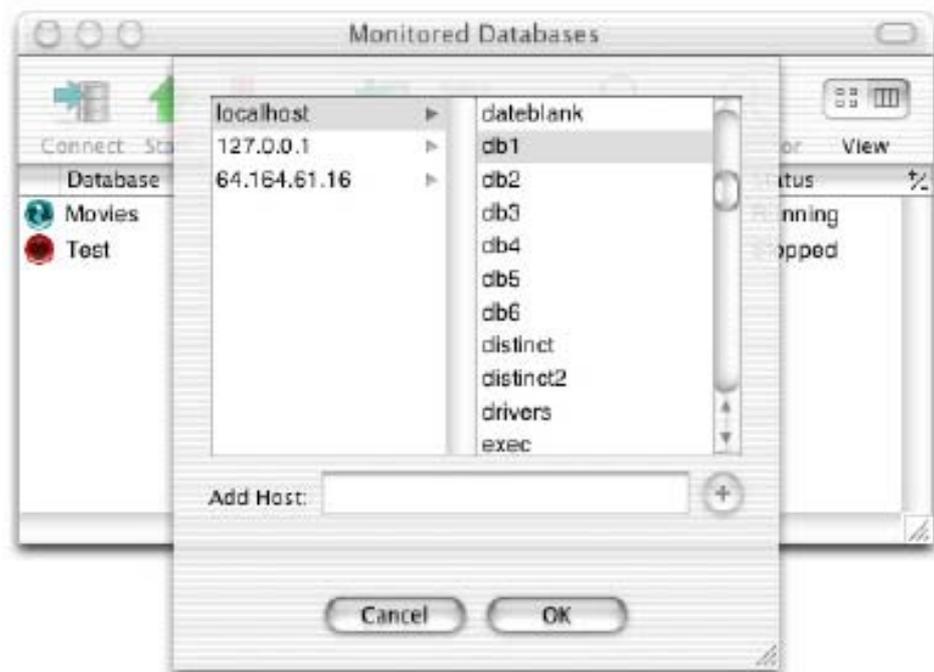


あなたが復帰することを好むホスト名とデータベースを選択してください。最もしばしばあなたはちょうど最近のバックアップを復元することを望むが、もしあなたがもっと古い (の・もの・人) を復活させた方がましであるなら、ファイル名を指定するでしょう。もしあなたがバックアップから前進 (roll forward) トランザクションを望むなら (もしあなたがトランザクションデータを書いているなら)、「Rollforward トランザクション (Rollforward transactions)」オプションを選択してください。あなたが「復帰する (Restore)」ボタンをクリックするとき、(もし必要とされるなら) FrontBaseManager はデータベースを作成して、データベースをスタートして、そしてすべての内容を再ロードするでしょう。

もしデータベースが削除されたなら、バックアップを復帰するために、データベースが使ったホストを選択して、そして次にデータベース textfield のデータベースの名前をタイプしてください。もしあなたが最新のバックアップからトランザクションログでセーブした前進 (roll forward) トランザクションを望むなら、「Rollforward トランザクション」チェックボックスがチェックしてあるか確かめてください。「復帰する (Restore)」ボタンをクリックしてください、そうすればあなたのデータベースは復帰するでしょう、あなたの監視されたデータベースにビューを加えて、そしてスタートしました。

監視 (Monitoring)

FrontBaseManager は積極的にそれが現在監視しているそれらのデータベースを管理することを可能にします。それで、データベースアイコンあるいはあなたが見るリスト項目はそれが監視しているそれらのデータベースに過ぎません。モニタービューに既存のデータベースを加えるために、ツールバーで「付け加える (Add)」ボタンをクリックするか、あるいはメニューから Database->Monitor を選択してください。次のパネルが現われます：

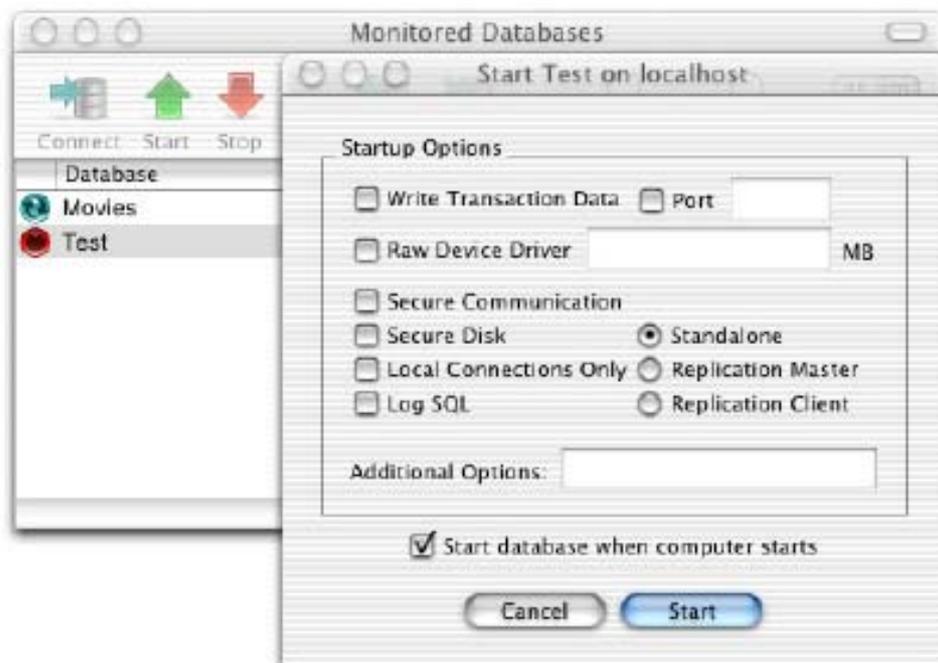


ブラウザの左の行はすべてが現在ホストを監視したことを示します。右の行はすべての存在を示します、しかし、選択したホストの

データベースはまだ監視されていません。まだ示されないホストのデータベースを監視するには、ブラウザの下に textfield のホスト名を入力して、そして<リターン>を押してください。監視されたデータベースのリストにデータベースを加えるには、ブラウザのデータベースを選択して、そして「OK」ボタンをクリックしてください。あなたはすぐに多数のデータベースを加えることができます。それを選択して、そして「取り除く (Remove)」をクリックすることによって、あるいはそれをセレクトして、そしてデータベース -> 「監視をストップする」 (Database->"Stop Monitoring") を選択することによって、あなたは監視したデータベースリストからデータベースを取り除くことができます。

データベースをスタートして、ストップします

まだ走っていない監視されたデータベースをスタートして、モニタービューのそれを選択して、そして、スタートをクリックするか、あるいは、Database->Start あるいは Database->"Start Advanced" を選択してください。スタートをクリックする、あるいはデータベース->スタート (Database->Start) を選択することは即座にデータベースを仕事を始めます。「先にスタートする (Start Advanced)」を選択することはあなたに、実際にデータベースをスタートする前に、オプションを指定することを可能にしてパネルを出します。



あなたはこのウィンドウでスタートオプションを選択することができます。オプションは次に説明します：

書き込みトランザクションデータ (Write Transaction Data) – バックアップから復帰する後に、もしあなたがトランザクションデータを書き上げたなら、バックアップから「前進」のトランザクションを実行することができます。

ポート (Port) – ポートを指定することは FrontBase データベースに特定のポート番号を使うように指示します。

未使用のデバイスドライバー (Raw Device Driver) – FrontBase は非常に進歩した書き込みを通してデータの記憶場所として同様に未使用のデバイスの使用 (パーティション) をサポートするキャッシュメカニズムを提供します。もしあなたがこのオプション (そしてライセンスがそれを許可する) を使いたいなら、未使用のデバイスドライバーをクリックして、そしてこのキャッシュの大きさを指定してください。

ローカル接続のみ (Local Connections Only) – このオプションを指定することによって、FrontBase のデータベースがデータベースと同じコンピュータで走っているクライアントから接続を受け入れるでしょう。(デフォルトは、ローカルなクライアントと同様、ネットワークで結ばれた接続を受け入れます。)

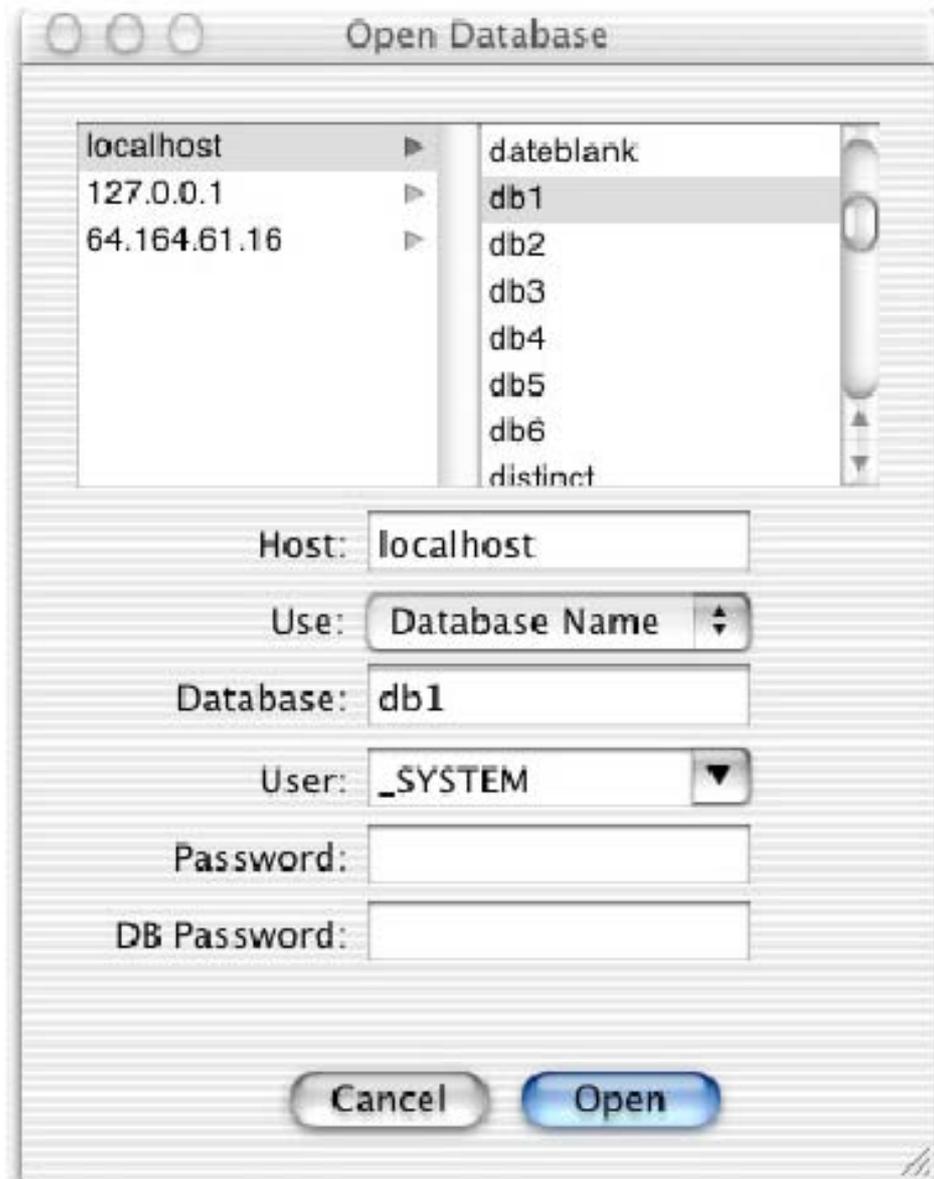
SQL をログファイルに書く (Log SQL) – このオプションを指定することによって、FrontBase のデータベースは SQL 文のファイルを受け入れたすべてをログファイルに書くでしょう。ファイルは FrontBase インストールの Databases ディレクトリで作成されて、そして <データベース名>.fb.sql と命名されます。(デフォルトは SQL 文をログファイルに書けません。) 例えば、あなたはムービーという名前のデータベースで「SQL をログファイルに書く」ことができます、そして (Mac OS X で) あなたのログは /Library/FrontBase/Databases/ Movies.fb.log にあるでしょう。

複製のオプション (Replication options) – FrontBase によって提供された複製のスキームで、あなたはすべてのアップデートが行われるはずである原版のデータベースを持つことができます。読み込み専用の複製クライアントの数を加えることができます。あなたは新しいデータベースが New Database Advanced ウィンドウによって独立型 (standalone) のサーバー、複製の原版 (replication master) あるいは複製クライアント (replication client) としてスタートするかどうか指定することができます。

データベースに接続する

あなたが走っているデータベースを持っている途端に、アイコンを選択して、そして「接続 (Connect)」ボタンをクリックするか、アイコンを選択して、そしてファイル -> オープンデータベース (File->Open Database) を選択することによって、あるいは、データベースアイコンをダブルクリックすることによって、データベースに接続することができます。もしあなたがまだ走っていないデータベース

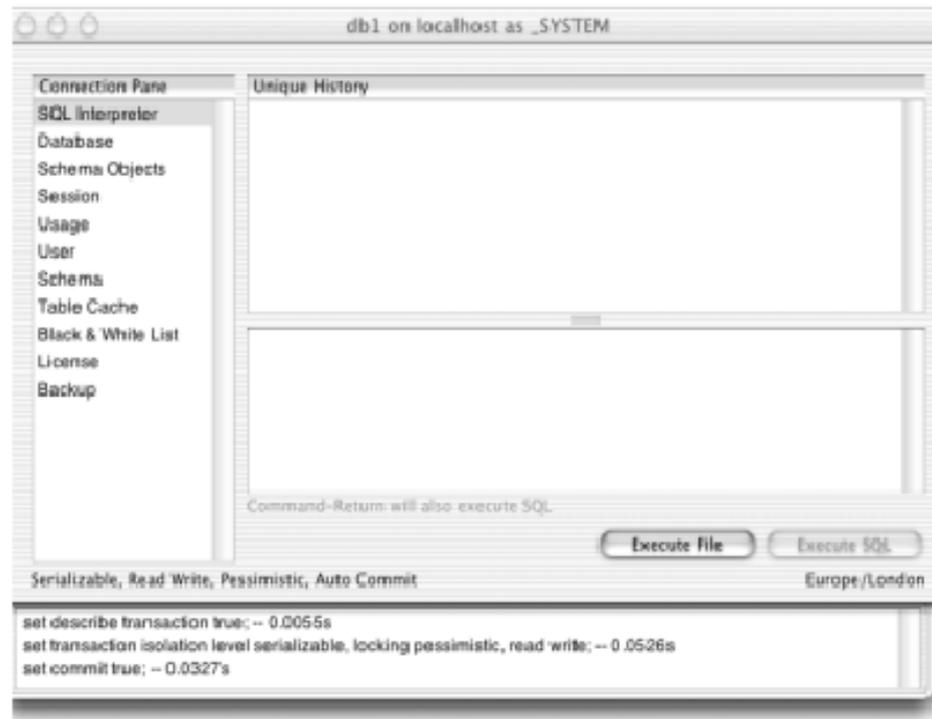
に接続しようと試みるなら、あなたは「接続することが不可能です：データベースが走っていません」というメッセージを得るでしょう。あなたが走っているデータベースに接続しようと試みるとき、あなたは次のダイアログ・ボックスを得るでしょう：



このウィンドウはあなたが接続するために選択したデータベースを示して、そして同様に選択肢を変更することを可能にします。それはあなたにデータベース名あるいは TCP ポートを使って接続することを可能にします。最終的に、それはもしあるならユーザとパスワードと DB パスワードを指定することを可能にします。あなたは前の接続で使われたユーザ名のリストからユーザを選択するか、あるいは

新しいユーザ名でタイプすることができます。あなたが初めて（直接それを作った後で）データベースに接続するとき、スーパーユーザ `_SYSTEM` だけが提供されます。もし接続が確立することができるなら、主な接続ウィンドウが現われます。

接続ウィンドウ



接続ウィンドウは FrontBaseManager の主なウィンドウです。このパネルはあなたに FrontBaseManager が提供する機能性の大部分についてアクセスできます。それで主な接続ウィンドウの最初に要素を調べましょう、そして次に我々は接続パネルのそれぞれの別個の交換したビューを徹底的に調べるでしょう。

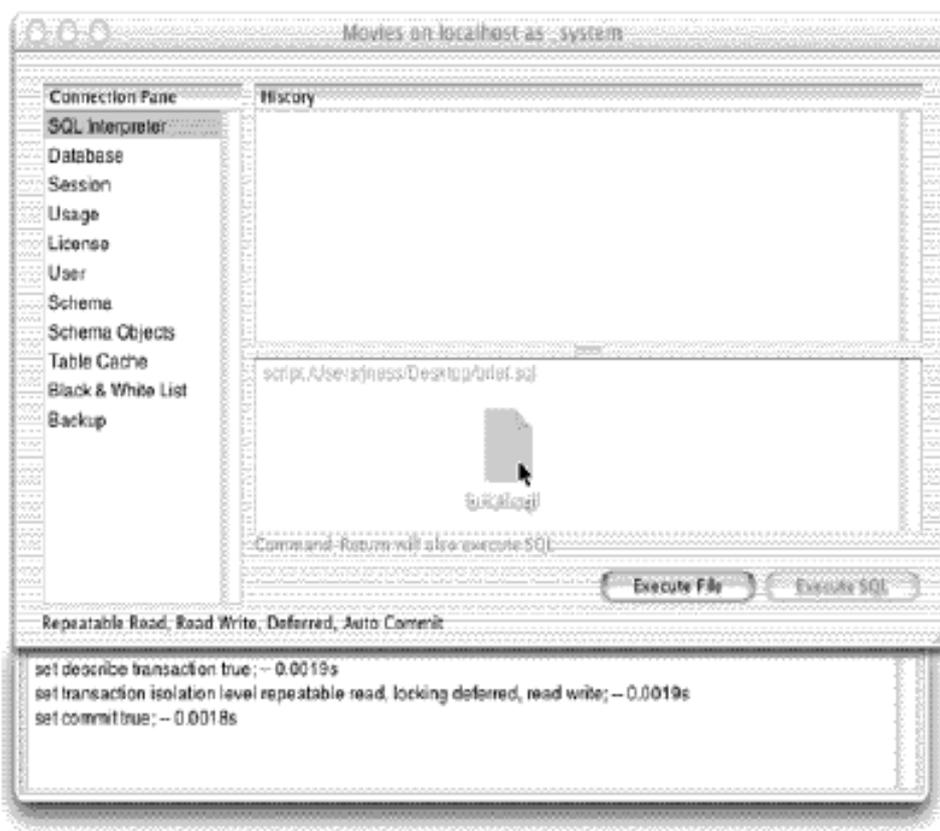
接続ウィンドウのタイトルバーはデータベース名、それが走っているホストと接続のユーザが入っています。接続パネルの左ボタンの角に常にそこで現在のトランザクションの設定を思い出させる

トランザクションの設定があります。これらのトランザクションの設定はセッションパネルを通して変更することができます。(すべて利用可能なトランザクションの設定の定義と説明について付録1を見てください。) デフォルトで、drawer (引き出し) はすべての SQL 文のログがこの接続の中でデータベースに送ることを示している接続ウィンドウに拡張します。(ビュー->「SQL ログを隠す」(View-> "Hide SQL Log") を選択することによって、この drawer (引き出し) は隠すことができます)

SQL インタプリター

接続ウィンドウはサブビューとして選択した SQL インタプリターパネルを使ってスタートします。SQL インタプリターはデータベースサーバーに直接 SQL を送ることを可能にします。そうするために、より少ないテキストエリアをクリックして、あなたの SQL にタイプして、そして「SQL を実行する (Execute SQL)」をクリックします。テキスト入力エリアの状態のメッセージとして、コマンドリターンが同様に SQL を実行するでしょう。あなたは SQL でタイプするように、SQL インタプリターは青の SQL 9 2 が予約したキーワードを強調することに注意するでしょう。これはあなたが SQL 文をデバッグするのを手伝います。あなたがサーバーに SQL を送るとき、SQL 文の履歴が履歴エリアをあなたのために取っておきます。デフォルトで、履歴リストは唯一取っています。すなわち、もしあなたが何か他の SQL 文を「select * from foo;」に出し、それから、再び「select * from foo;」を出すなら、あなたは履歴の 2 番目の「select * from foo;」を得ないでしょう。その代わりに、SQL 履歴は履歴リストの底にすでに履歴の「select * from foo;」を移動させるでしょう。もしあなたが履歴のすべての SQL 文を見る方を好むなら、あなたはプリファレンスでこの SQL の唯一の機能を変えることができます。

あなたは同様に SQL インタプリターを使って SQL のファイルを実行することができます。それには 2 つの方法があります。あなたは実行するファイルを選ぶために「ファイルを実行する (Execute File)」ボタンをクリックすることができます、あるいは、テキスト項目のエリアのファイルにドラッグ&ドロップすることができます。あなたがファイルをドラッグするように、グレースアウトした「script <filename>」がテキストエリアに現われるでしょう。それから、あなたがテキストエリアの中にファイルをドロップするとき、「script <filename>」は実際に実行されるでしょう。



ヒストリーラインの項目をクリックして、そして次に「SQL を実行する (Execute SQL)」をクリックすることによって、あるいはヒストリーラインをダブルクリックすることによって、あなたはヒストリーから容易にコマンドを再実行することができます。

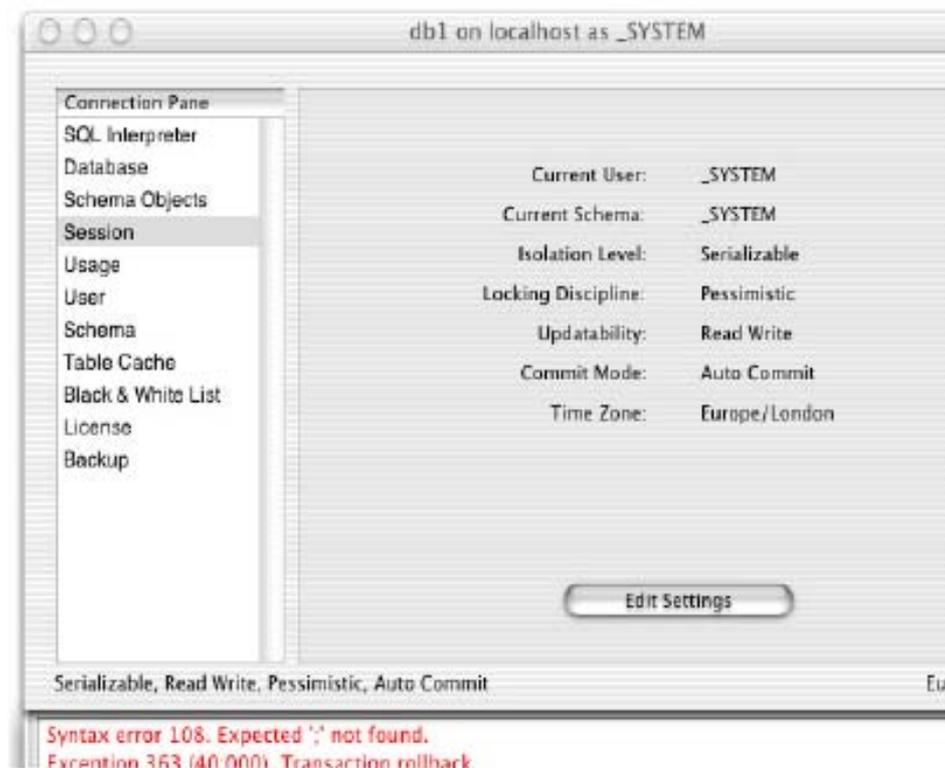
同様にあなたのトランザクションの設定を SQL インタプリターから変更することは容易です。SQL 項目の範囲を右クリックしてください、そうすれば、コンテキストメニューが出てきて、あなたにディスエーブル/イネーブルの自動確定 (Disable/Enable Auto Commit) についてのオプションを与えて、そして、コンフィギュレーションを設定する有用なトランザクションの番号を変更します。

データベース



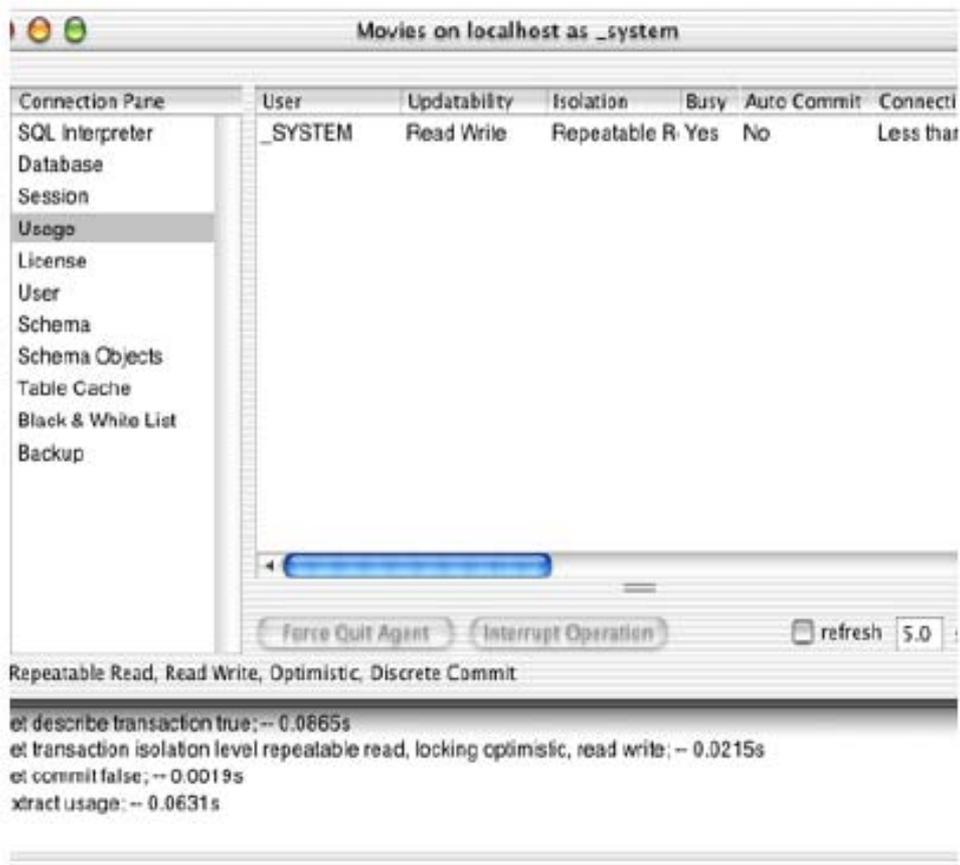
データベースパネルはデータベースについての情報が入っています。この画面の項目の大部分が自明です。サーバーのバージョンは現在のデータベースサーバーのバージョン番号です。データベースバージョンは初めにこのデータベースを作成したサーバーのバージョンです。あなたはデータベースのパスワードを変えるか、あるいはパネルの一番下のボタンを使ってこのパネルでデータベースをストップすることができます。

セッション

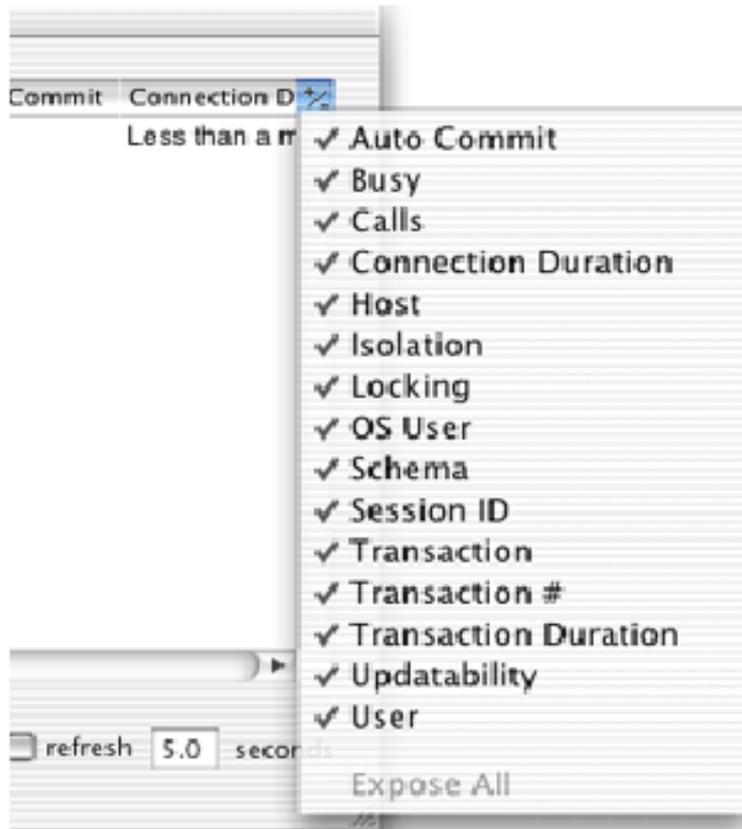


セッションパネルはこのセッションの接続についての情報が入っています。それは現在のユーザ (current user)、現在のスキーマ (current schema)、アイソレーションレベル (isolation level)、ロックの仕方 (locking discipline)、アップデートする方法 (updatability)、そして確定モード (commit mode) を示します。現在のスキーマあるいはトランザクション設定を編集することを可能にする「編集設定 (Edit Settings)」ボタンがあります。

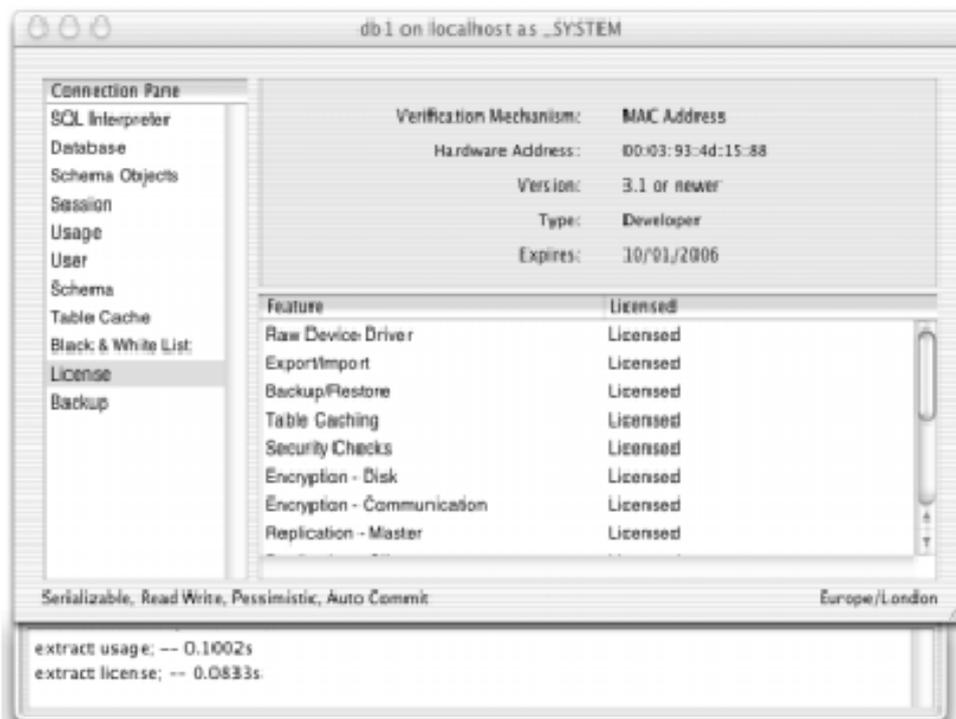
使い方



使い方のパネルは現在、接続されたデータベースに有効なすべての接続を示します。それは接続の情報、トランザクション設定と接続の継続時間を示します。一番下のパネルは選択したトランザクションの現在走っている SQL を示します。あなたは示される情報的なフィールドのリストをフィルターするために右上の +/- ボタンを使うことができます。



ライセンス

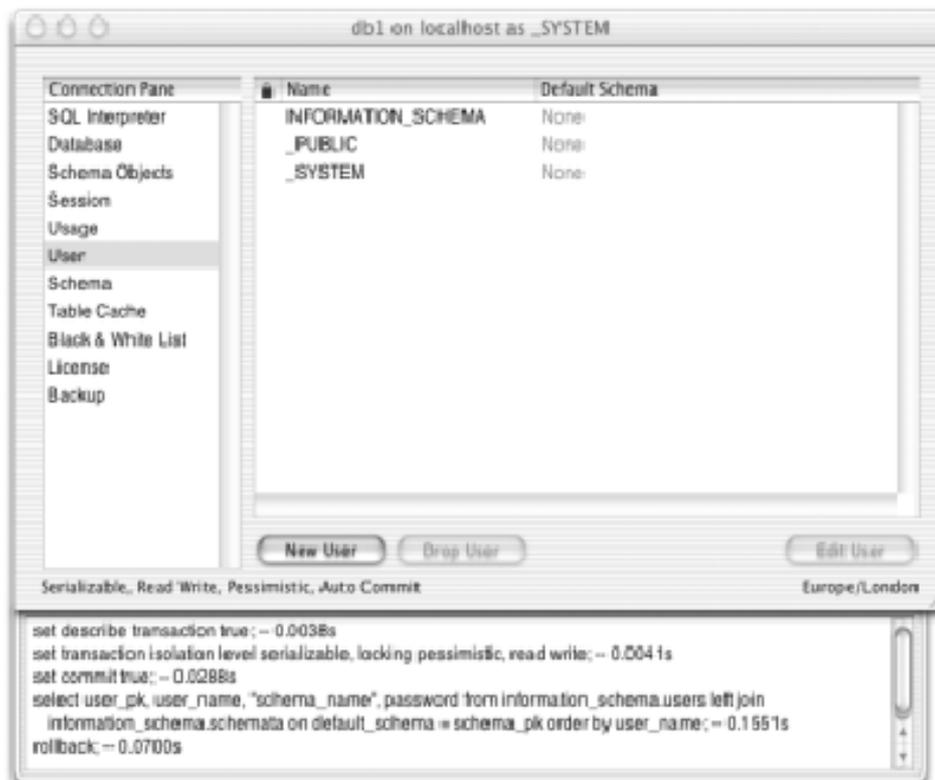


ライセンスパネルはあなたの現在のライセンス情報を示します。それは検証メカニズム（MAC アドレスあるいは IP アドレス）、それが反対の実証している実アドレス、バージョン、ライセンスタイプと満了の日付を示します。それはそれから同様にすべての機能とその機能のライセンスを与えられた状態（ライセンスが与えられた、あるいはライセンスが与えられていない）を示します。ライセンス情報を変更するために、ツール -> 「ライセンス管理」(Tools->"License Management") をクリックして、ホストを指定して、そして「ライセンスを編集する (Edit License)」ボタンをクリックしてください。



あなたはその時ライセンスキーと FrontBase から受け取ったチェックキーを指定することができます。「ライセンスをセットする (Set License)」をクリックしてください、そうすれば、あなたがサーバーを再起動した後、新しいライセンスが有効になるでしょう。新しいライセンスを有効にするためにサーバーを再起動してください！

ユーザ



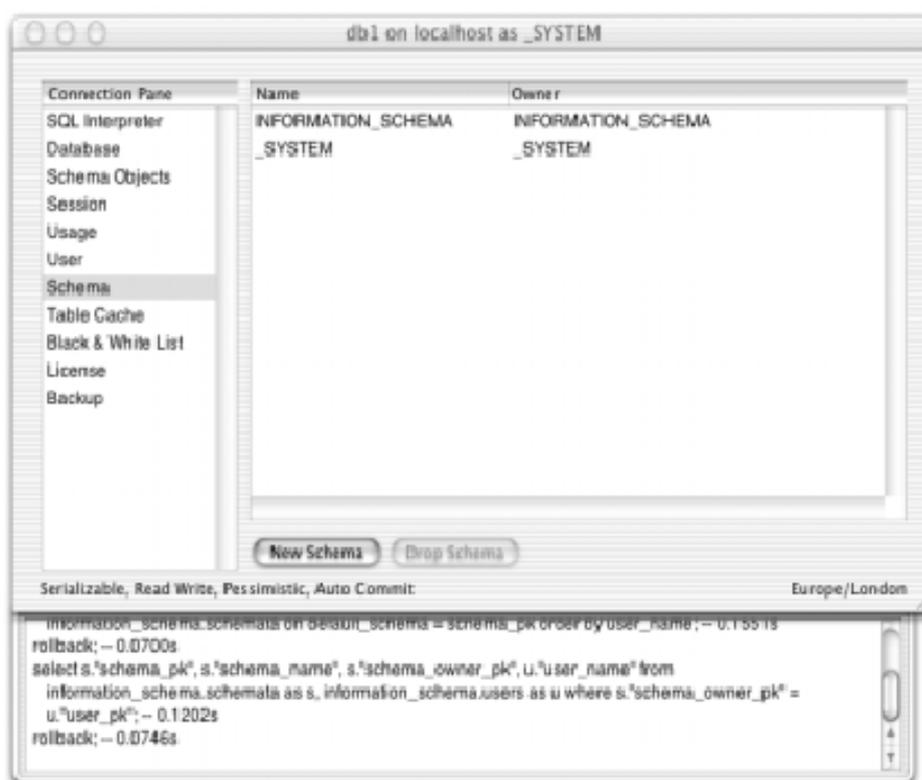
ユーザパネルはユーザを加えて、ドロップして、そして編集するツールを与えます。

新規ユーザを作成するために、「新規ユーザ (New User)」ボタンをクリックして、そしてユーザ名、スキーマオプションとパスワードを指定してください。もしあなたが「デフォルトスキーマを作る」ことを指定するなら、FrontBaseManager はユーザ名と同じユーザ名のためにスキーマを作るでしょう。（「ボブ」という名前の新規ユーザが「ボブ」という名前の新しいスキーマを作って、そして彼のデフォルトスキーマとして割り当てるでしょう。）そうでないときは、あなたはデータベースで定義した他のスキーマのいずれも、あるいはデフォルトスキーマを持っていないためにユーザのデフォルトスキーマを指定することができます。

ユーザをドロップするために、ユーザを選択して、と「ドロップユーザ (Drop User)」をクリックします。シートが「カスケードをドロップする (Drop Cascade)」か、あるいは「リストリクトをドロップする (Drop Restrict)」かどうか要求して来るでしょう。もしユーザがスキーマあるいはスキーマオブジェクトを所有しないなら、あなたはただ「リストリクトをドロップする (Drop Restrict)」を選択することができます。「カスケードをドロップする (Drop Cascade)」を選択することはユーザと関連づけられたスキーマとスキーマオブジェクトをドロップするでしょう。

ユーザを編集することはユーザのデフォルトスキーマとパスワードを変更することを可能にします。

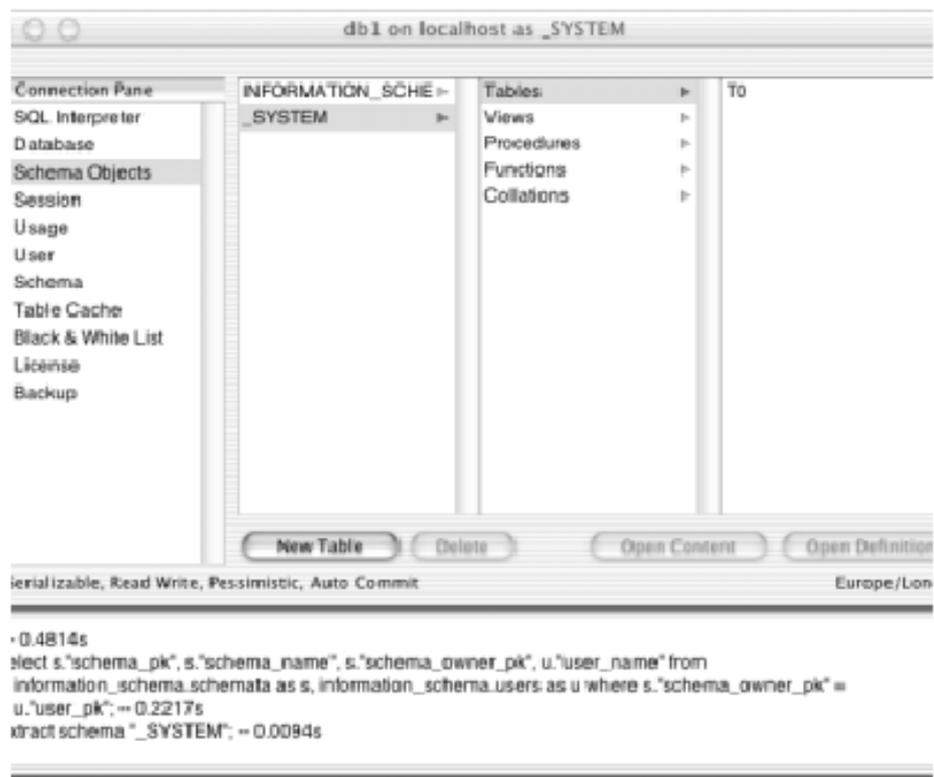
スキーマ



スキーマパネルはあなたにスキーマを加えるか、あるいはドロップすることを可能にします。新しいスキーマを作るために、「新規スキーマ (New Schema)」ボタンをクリックして、そしてスキーマ名と

所有者を指定してください。スキーマをドロップするために、ユーザ自身がログファイルに書いたスキーマを選択して、そして「スキーマをドロップする (Drop Schema)」ボタンをクリックしてください。もしスキーマがスキーマオブジェクトに入っているなら、あなたは同様にすべての関連づけられたスキーマオブジェクトを削除するために「カスケードをドロップする (Drop Cascade)」をするはずですが。もしスキーマがからであるなら、あなたは「リストラクトをドロップする (Drop Restrict)」ことができます。

スキーマオブジェクト



スキーマオブジェクトのブラウザは最も強力な FrontBaseManager の機能性のいくつかが入っています。それはあなたにビューを作って、削除して、そしてあなたが表とデータについて本当に望む物を修正することを可能にします。このパネルはブラウザあるいはファイндаのように編成されます。あなたは最初にスキーマを選択して、それからオブジェクトタイプ、それから、それらの内容を作って、

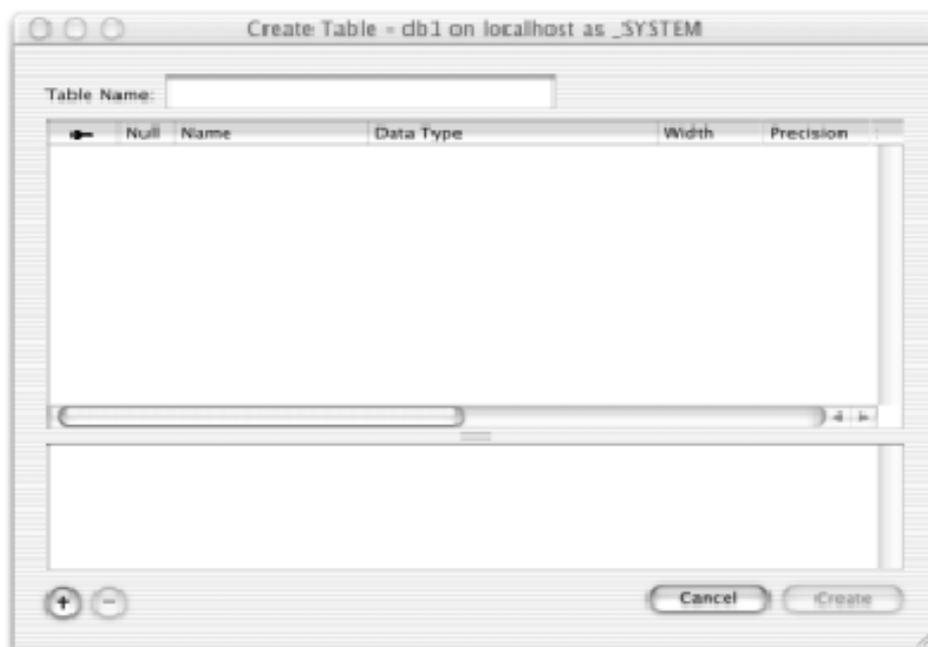
削除して、開くか、あるいはそれらの定義を開くためにスキーマオブジェクトのリストを与えるでしょう。

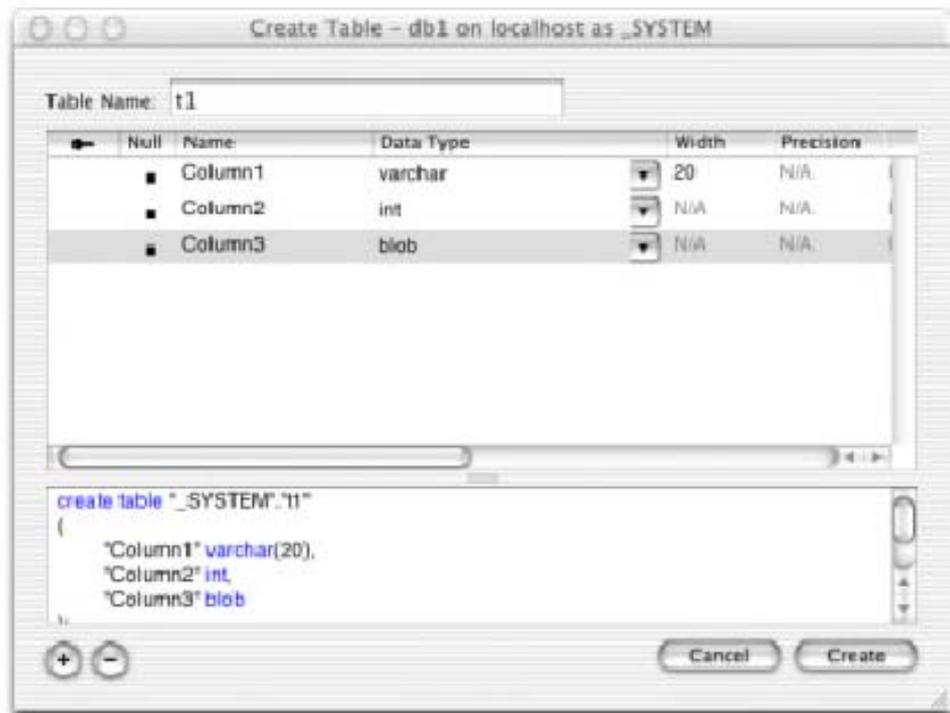
新しいスキーマオブジェクト

FrontBaseManager が巧みに使うことを可能にする5つの違ったスキーマオブジェクト、表、ビュー、手順、関数と照合があります。FrontBaseManager はそれぞれのタイプの新しいオブジェクトを作成するために直観的なビューを提供します。最初に新規テーブルエディタを見ましょう。

新規テーブル

それを得るために、あなたが表、それから「テーブル (Tables)」を選択することを作成したいスキーマを選択してください。「新規テーブル (New Table)」ボタンは使用可能であるでしょう。「新規テーブル」ボタンをクリックします、そしてあなたはこのようなビューを与えられるでしょう：

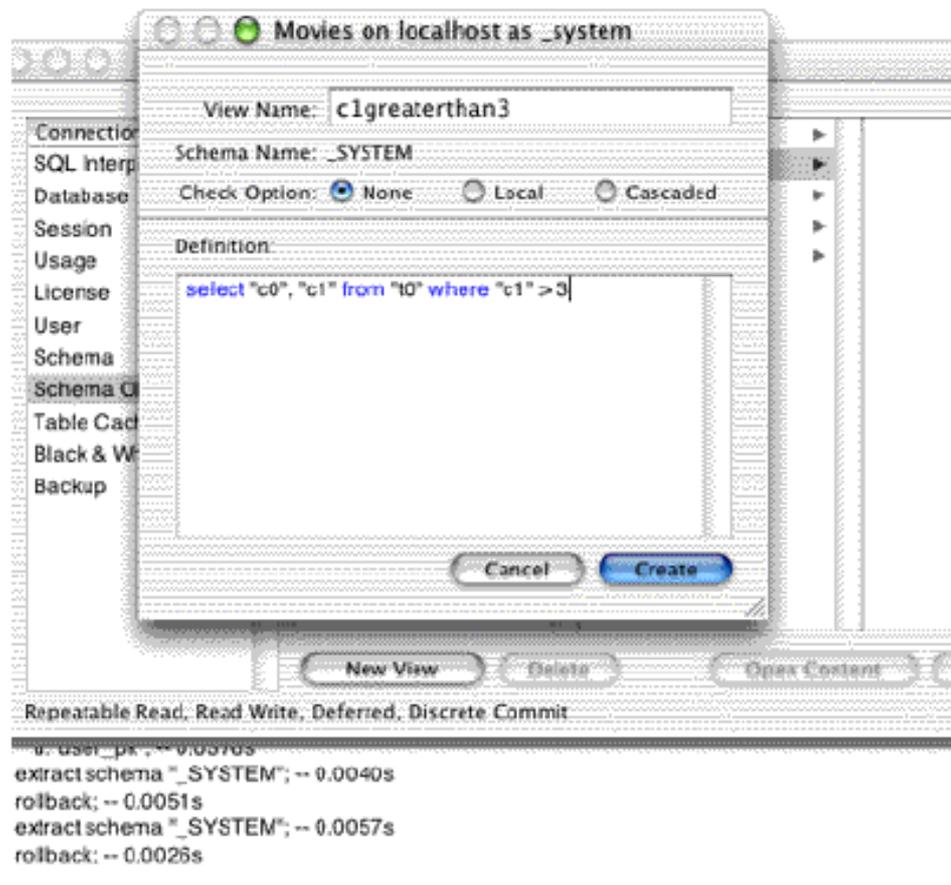




あなたはそれから、表の名前を指定して、そして行をセットするためにウィンドウの左下ので+ ボタンをクリックすることができます。あなたは表の名前を指定して、そして行を加えるとき、実行されるSQL が下のパネルで示されていることに注意するでしょう。これはあなたが正確に何が所定の表の準備のために起こっているか知っていることを保証するのに助けて、そしてプロセスであなたにもう少し多くの SQL 9 2 を教えます。あなたはデータタイプあるいは上のパネルのどんな属性でも変えるように注意するでしょう、下のパネルのSQL は何のSQL が新しい表の定義を与えて実行するか示すために変更します。左から右のラベルは主キー（もしキーが現われるなら、この行は主キーです）です、空（もしインジケータが現われるなら、この行は空を認めます）、行の名前、行のデータタイプ、幅、精密度、寸法、デフォルト値、標準化されるかどうか、そして行の照合の名前、を認めます。あなたが「作る (Create)」ボタンを押すまで、実際に作成するSQL の表がサーバーに送らないことに注意することは

重要です。これは正確にあなたが、それを実際に作る前に、どのように表を設計することを望むか定義するために自在に使うことができるキャンバスを与えます。

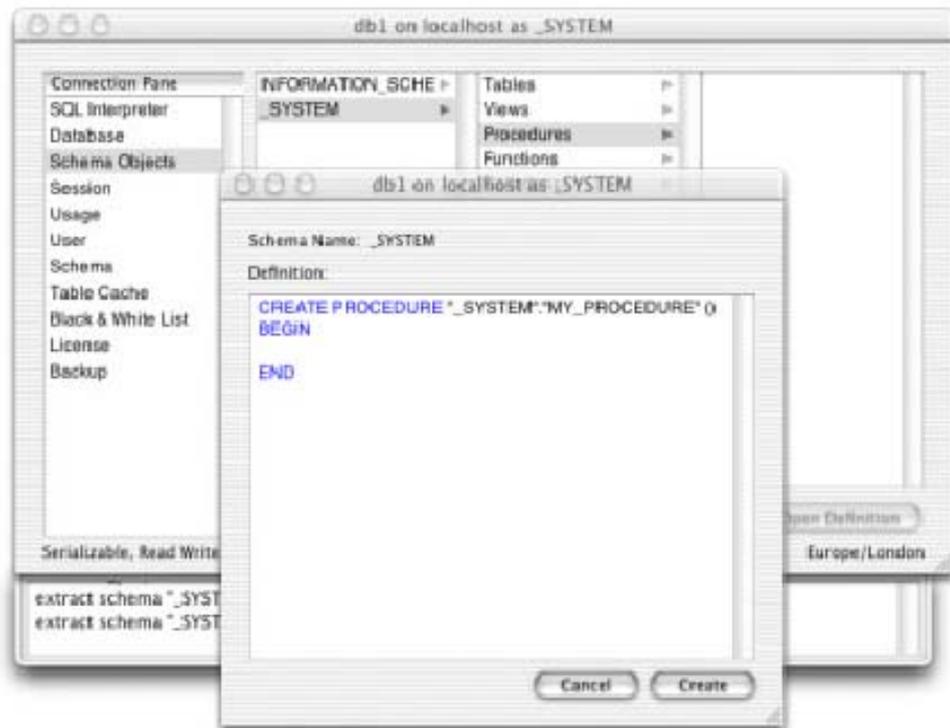
新しいビュー



新しいビューエディタは表のエディタより自由な形式です。それがあなたにビューの名前を指定することを可能にして、スキーマ名を前もって代入します、チェックオプション（SQL92 スタンドで定義した先進的な特徴）とビューの定義を指定するためにあなたにラジオボタンのグループを与えます。例えば、c1（int 行）が3より大きい表 t0 から c0 と c1 という名前の行が入っているビューを作るために、定義の中に次のことを入力してください：

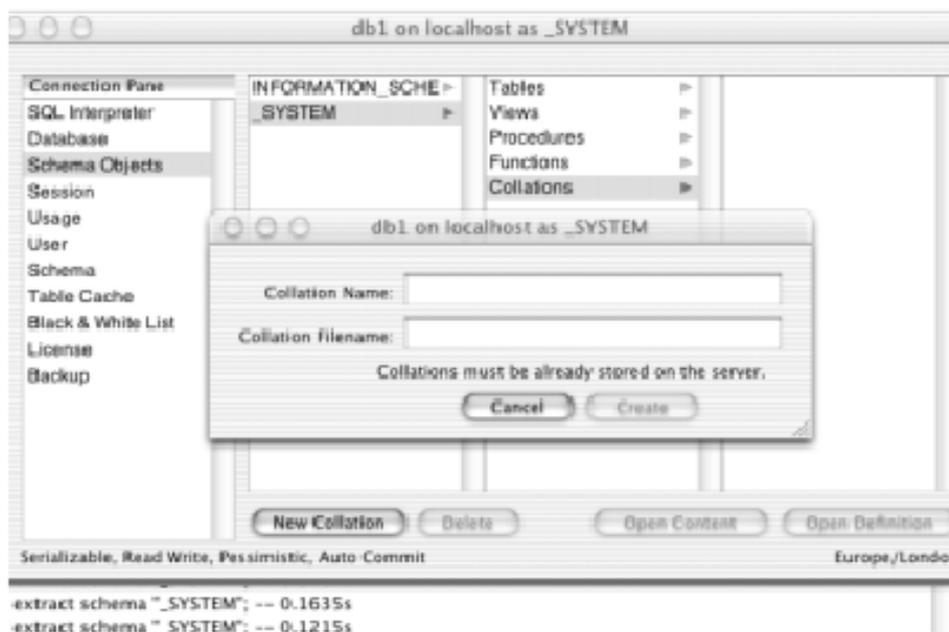
```
select "c0", "c1" from "t0" where "c1" > 3
```

新しい手順と新しい関数



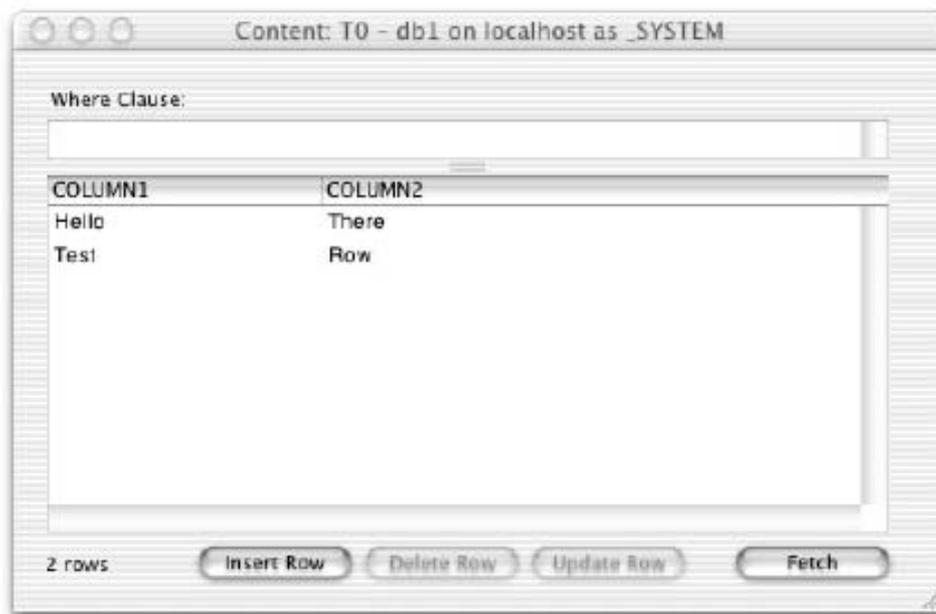
新しい手順と新しい関数のウィンドウの両方とも非常に簡単です。あなたが定義したスキーマの中で手順あるいは関数の項目にスキーマオブジェクトブラウザでナビゲートするとき、「新しい関数 (New Function)」あるいは「新しい手順 (New Procedure)」ボタンが使用可能になるでしょう。あなたがボタンをクリックするとき、結果として出てくるウィンドウは関数あるいは手順を定義することを可能にするでしょう。

新規の照合 (Collation)



スキーマオブジェクトブラウザの「新規の照合 (New Collation)」ボタンをクリックすることはあなたにサーバーの照合名と実際の照合ファイルのファイル名を指定することを可能にする小さいウィンドウを出します。(照合はすべて /Library/FrontBase/Collations に記憶されてます、それであなたはパスを指定する必要がありません。)

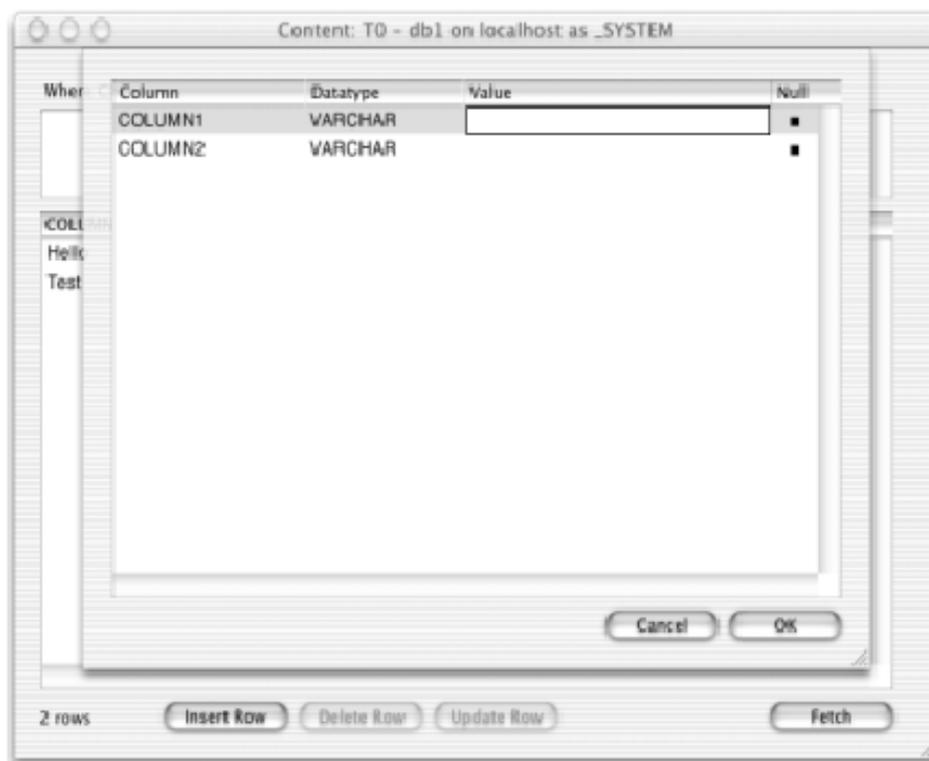
スキーマオブジェクトの内容を開く



あなたが定義された表あるいはビューを選択すると、「内容を開く (Open Content)」ボタンはイネーブルになります。このウィンドウはあなたに内容を見て、そして列を挿入して、削除して、そしてアップデートすることを可能にします。ウィンドウが最初に現われるとき、それは表のすべての列（フェッチの限界まで）を示します。上のパネルで、どこの文節か (where clause) を指定して、そして「フェッチ (Fetch)」ボタンをクリックすることによって、あなたはフェッチされた（そして示される）列を制限することができます。あなたは 'where' の後に記入する SQL 文の部品だけを指定してください。例えば、もしあなたがその表の中で column3 を int 行とする表を持っていたなら、「column3 > 5」は正当な「どこの文節か (where clause)」であるでしょう。列を削除するために、列あるいは列の数を選択して、そして削除 (delete) ボタンをクリックしてください。もしスキーマオブジェクトがアップデート可能である（若干のビューはそうではないが）なら、列は削除されるでしょう。列を挿入して、そしてアップデートすることは、あなたが代入して欲しい行のどれでも値を

指定することを可能にするウィンドウからシートをペロッと出す (pulling down) ことによって、起きます。

列を挿入して、アップデートする



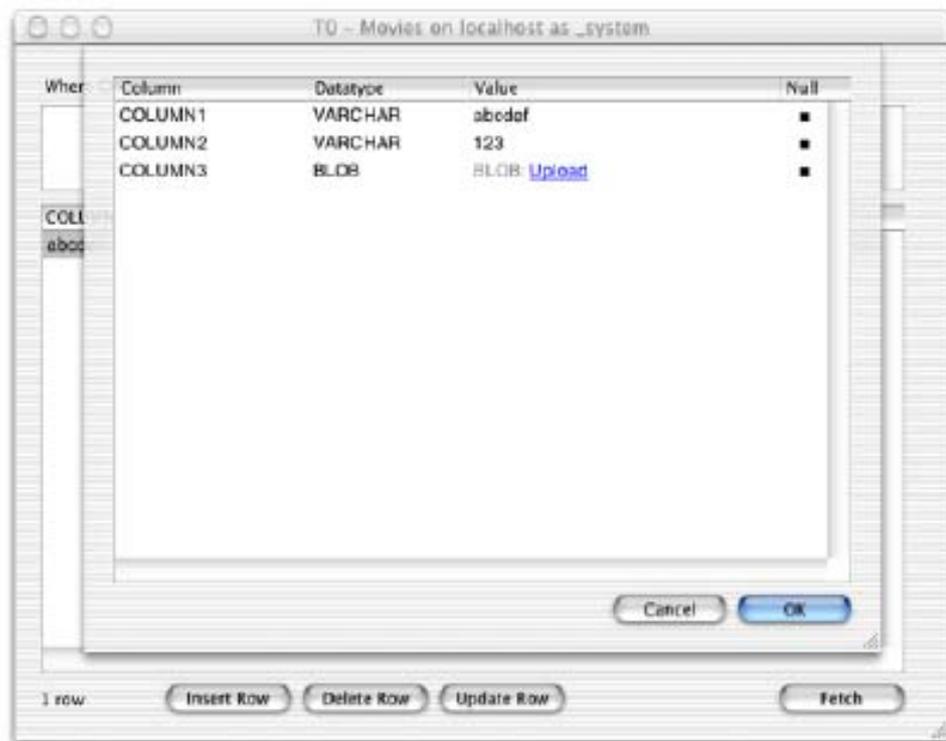
列エディタのシートは表の行の名前、データタイプ、値、そして、ヌルインジケータを可能にするための行の表のビューが入っています。値を入力するとき、実際の内容を入力してください。例えば、日付の行 c0 が入っている表 t0 の中に列を挿入する SQL が次のように見えるでしょう：

```
insert into "t0" ("c0") values (DATE '2001-08-11');
```

しかしながら、あなたは値の部分、2001-08-11を入力する必要があるだけで、列エディタはあなたのために日付を前接続します。列を挿入するとき、もしあなたがゼロを認めない行のために値を指定しなかったなら、「OK」ボタンはディセーブルでしょう。

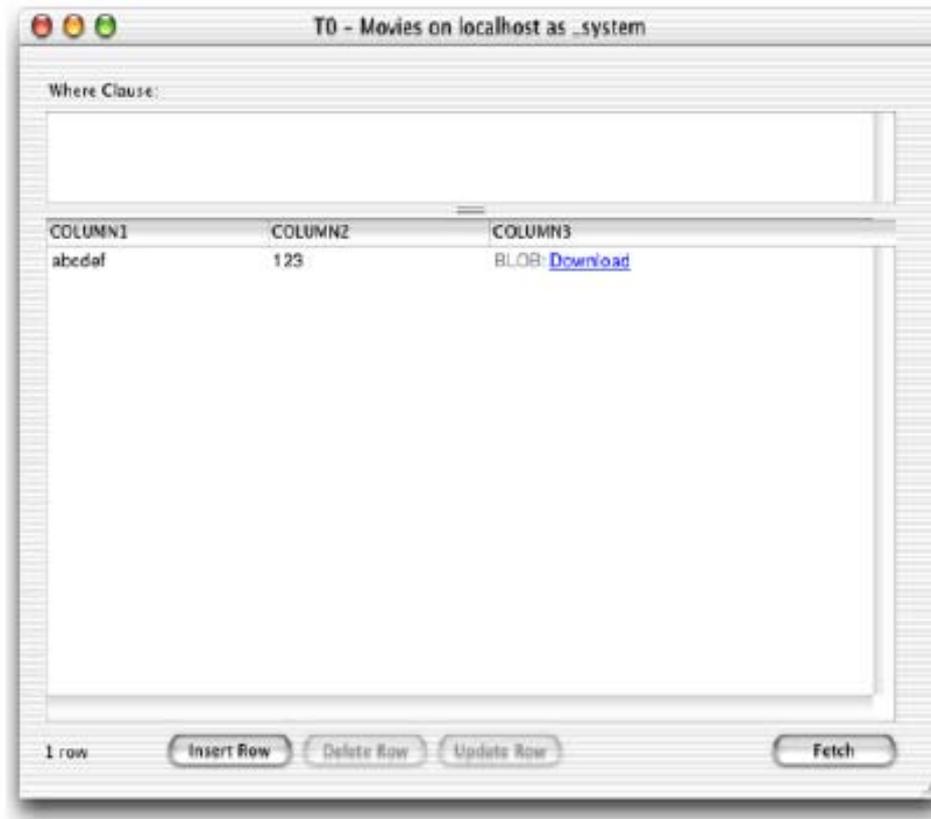
BLOB と CLOB の処理

FrontBaseManager は大きいオブジェクトをアップロードして、そしてダウンロードするために直観的なユーザインタフェースを提供します。もし表あるいはビューがデータタイプ BLOB あるいは CLOB で行を持っているなら、あなたは内容のエディタで直接これらの値を編集することはできません。しかしながら、あなたは CLOB / BLOB 行にあなたが欲しい内容が入っているファイルを指定することができます、そして FrontBaseManager はファイルで読み込んで、そしてその値に行をセットするでしょう。



もし CLOB / BLOB 行に内容が入っているなら、それはあなたのローカルなファイルシステムのファイルにその内容をダウンロードする

ために内容のブラウザのハイパーリンクを示すでしょう。ダウンロードリンクをクリックすることは、あなたに大きいオブジェクトデータを保存するファイル名を要求して、「ファイルをセーブする (Save File)」ダイアログを出すでしょう。



スキーマオブジェクト定義を開く

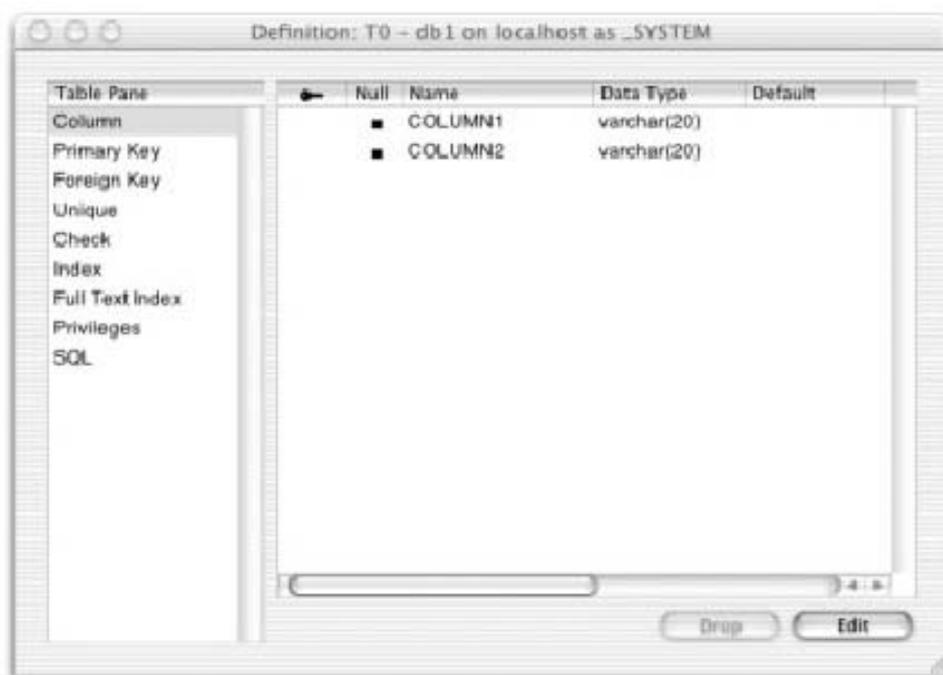
スキーマオブジェクト定義のウィンドウはあなたにスキーマオブジェクト自身についての情報を与えます。ビュー、手順と関数の定義ウィンドウはあなたに定義の情報を与えます (スキーマオブジェクトを作ったとき指定しました) そして、あなたにスキーマオブジェクトの権利を変更することを可能にします。照合の定義のウィンドウは

照合の名前、スキーマ名、タイプと外部名を与えます。表の定義のウィンドウはあなたに我々が詳細で見ることのできるかなり多くの情報を与えます。

表の定義のウィンドウ

表の定義のウィンドウはあなたに表についての多くの情報をアクセスするために与えます。1度、パネルのそれぞれをやってみましょう。表の行、主キー、外部キーなどを変えるために、あなたがトランザクション設定を `Serializable`、`Pessimistic` にセットするはずであることに注意してください。あなたは接続ウィンドウあるいは SQL インタプリターを右クリックすることによってセッションパネルを通してトランザクション設定を変更することができます。

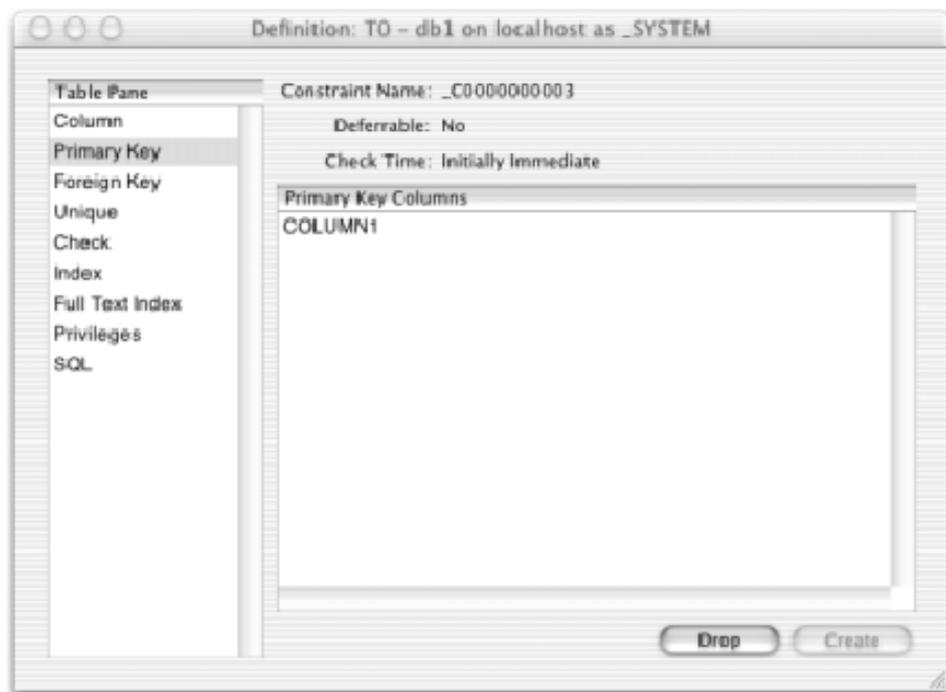
行の定義のパネル



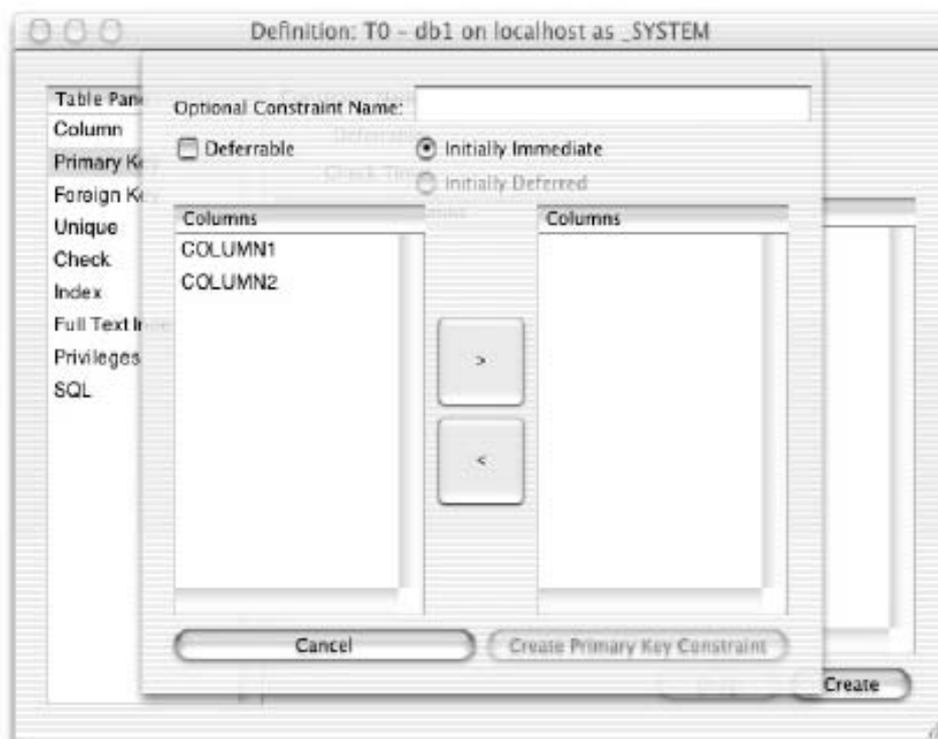
表の定義のウィンドウの行の定義のパネルはあなたに定義した表の行をドロップするか、あるいは編集する機会を与えます。行をドロップするために、行を選択して、そして「ドロップ (Drop)」ボタンをクリックしてください。あなたは、表が少なくとも1つの行を持つはずであるから、表のすべての行をドロップすることができるわけでは

ないことに注意してください。編集ボタンをクリックする、そして空の制約ではなく、行の名前、データタイプ、幅、精密度、寸法、デフォルト値をセットする、標準化される行を指定する、そして行に照合を指定する、主キーをセットすることによって、あなたは表の行を編集することができます。あなたが行の定義パネルの表を変えると、実行する SQL が下のパネルで作られることに注意してください。実際にサーバーに SQL を送るために「アップデート (Update)」をクリックしてください。

主キーのパネル



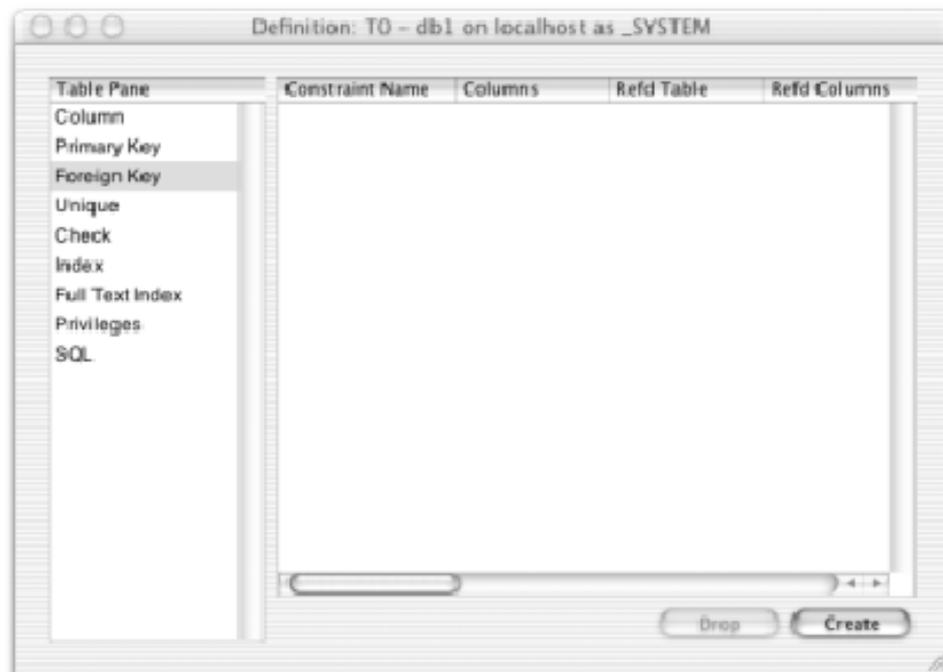
表の定義のウインドウの主キーの定義のパネルはあなたにドロップさせて、そして主キーの制約を作成させます。主キーの制約を作成するために、主キーの作成シートを出すために作成 (Create) ボタンをクリックしてください。



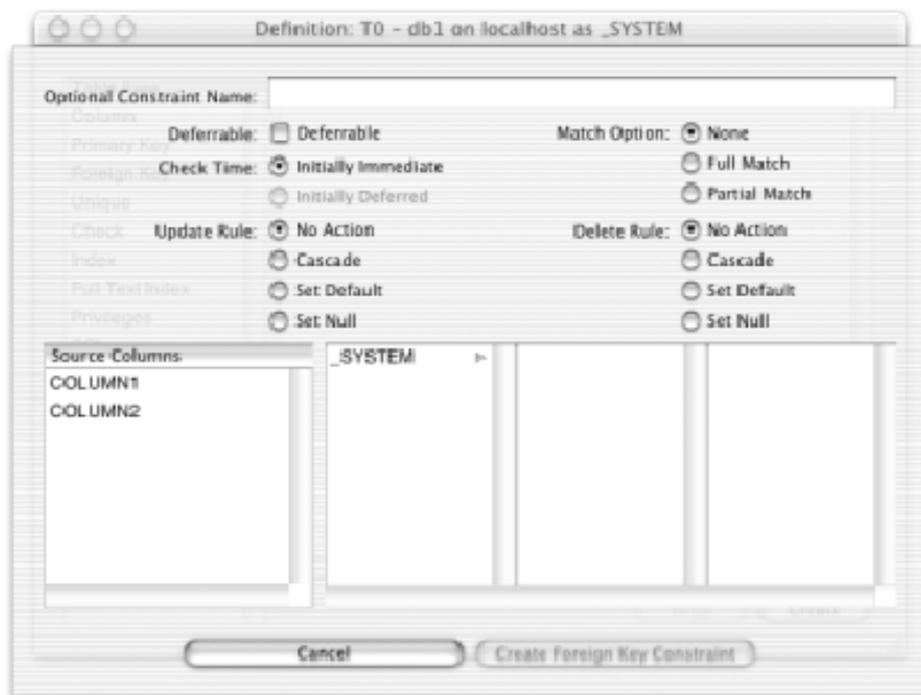
ここからあなたが（任意の）制約名を指定して、制約が延期可能であるか否かにかかわらず、そして、もしそれが延期可能であるなら、それが初めに延期されるか否かにかかわらず、設定することができます。あなたはその時どのコラムが主キーの制約に参加した方がいいか指定します、そうすれば「主キーの制約を作成（Create Primary Key Constraint）」ボタンをクリックすることができるようになります。もしあなたのトランザクション設定が正しかったなら、シートは引っ込むでしょう、そしてあなたは新しい主キーの制約を持つでしょう。

主キーの制約をドロップするためには、「ドロップ（Drop）」ボタンをクリックしてください。

フォーリンキー (Foreign Key) のパネル



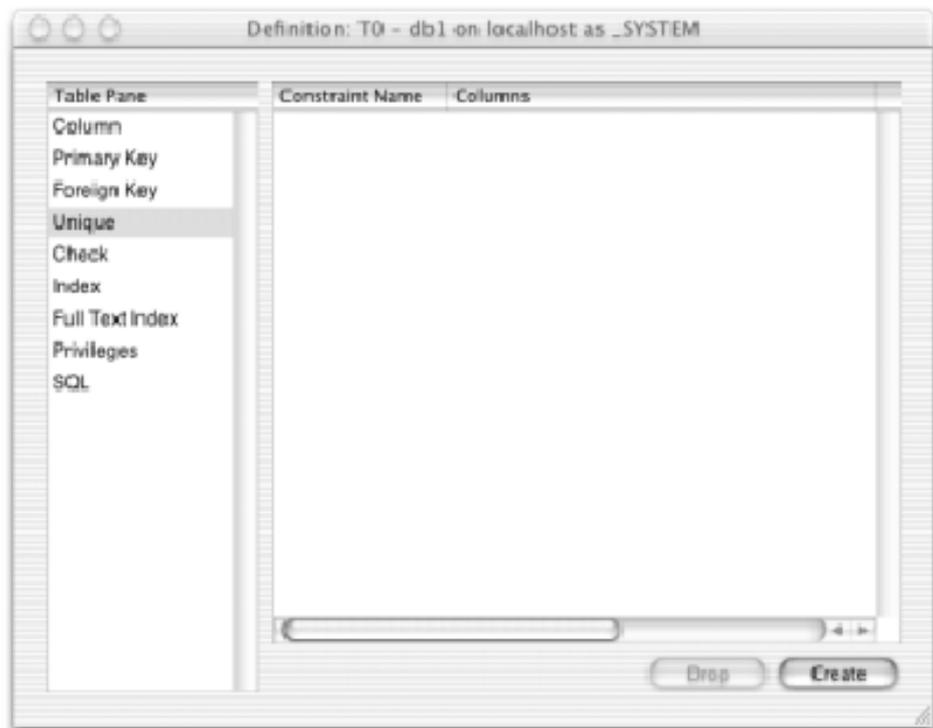
表の定義のウィンドウのフォーリンキー (Foreign Key) の定義のパネルはあなたにドロップさせて、そして外フォーリンキー (Foreign Key) の制約を作成させます。フォーリンキー (Foreign Key) の制約を作成するためには、フォーリンキー (Foreign Key) の作成シートを出すために作成 (Create) ボタンをクリックしてください。



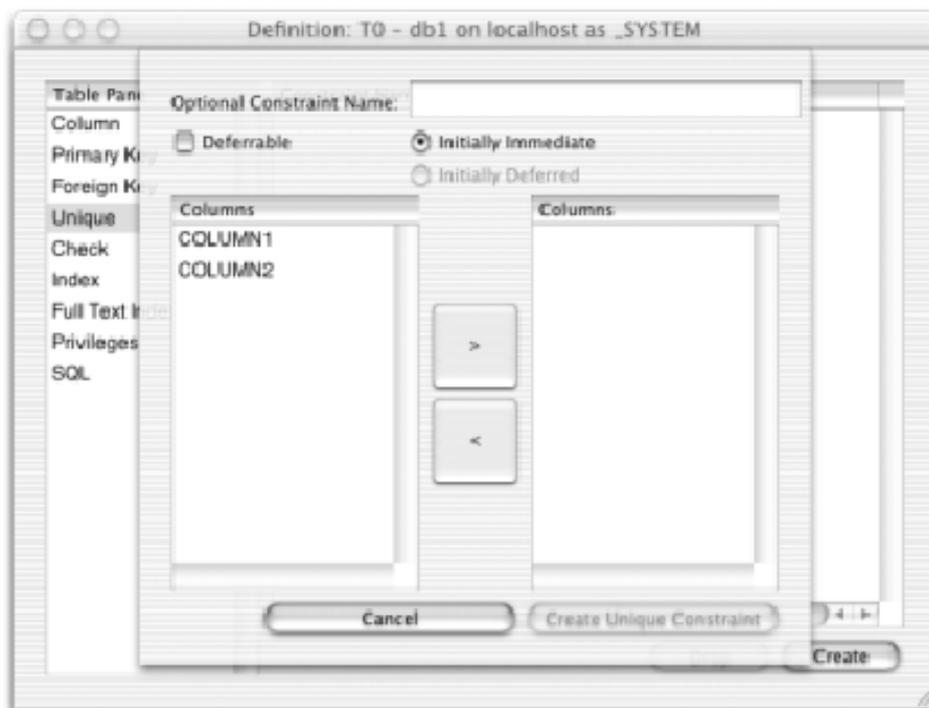
このシートはひるむように見えます、しかしあなたが本当に指定しなければならない2つのものはソース行（「ソース行 (Source Columns)」のブラウザ）と行き先行（「ソース行 (Source Columns)」ブラウザの右のブラウザ）です。ソース行と行き先行の数は、もちろん、一致するはずです。もしあなたがデフォルトビヘビア以外のフォーリンキー (Foreign Key) ビヘビアを必要とするなら、他のオプション（延期可能で、時間をチェックして、慣例をアップデートして、オプションと一致して、慣例を削除する）は変更することができます。

フォーリンキー (Foreign Key) の制約をドロップするためには、ただ制約を選択して、そして「ドロップ (Drop)」ボタンをクリックしてください。

唯一のパネル



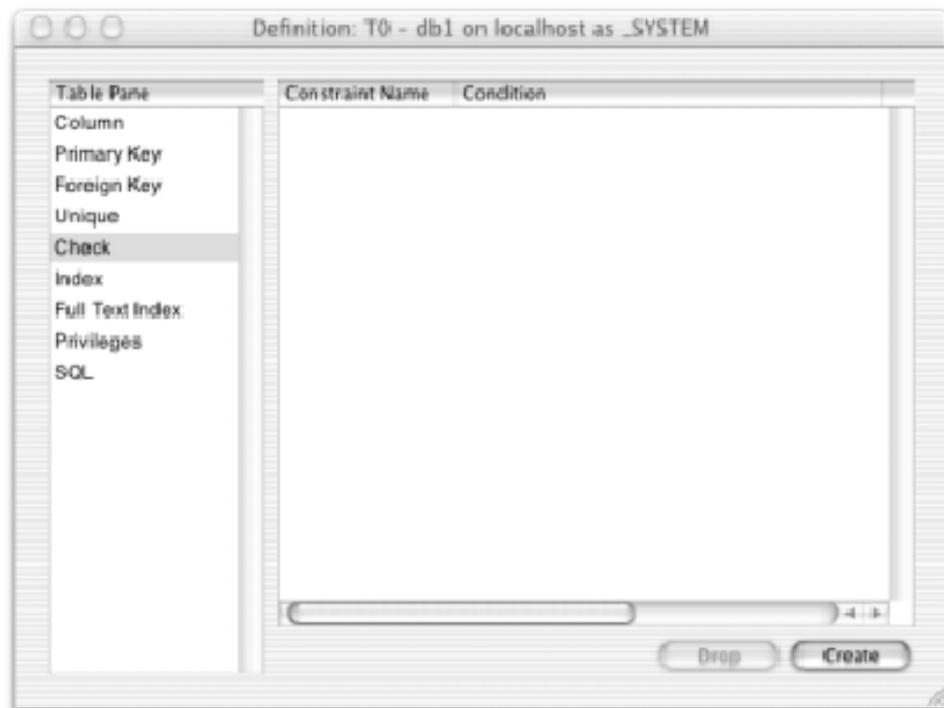
表の定義のウィンドウの唯一の定義のパネルは唯一の制約を作って、そしてドロップするためにインタフェースを提供します。唯一の制約を作るために、唯一の作成のシートを出すために「作成 (Create)」ボタンをクリックしてください。



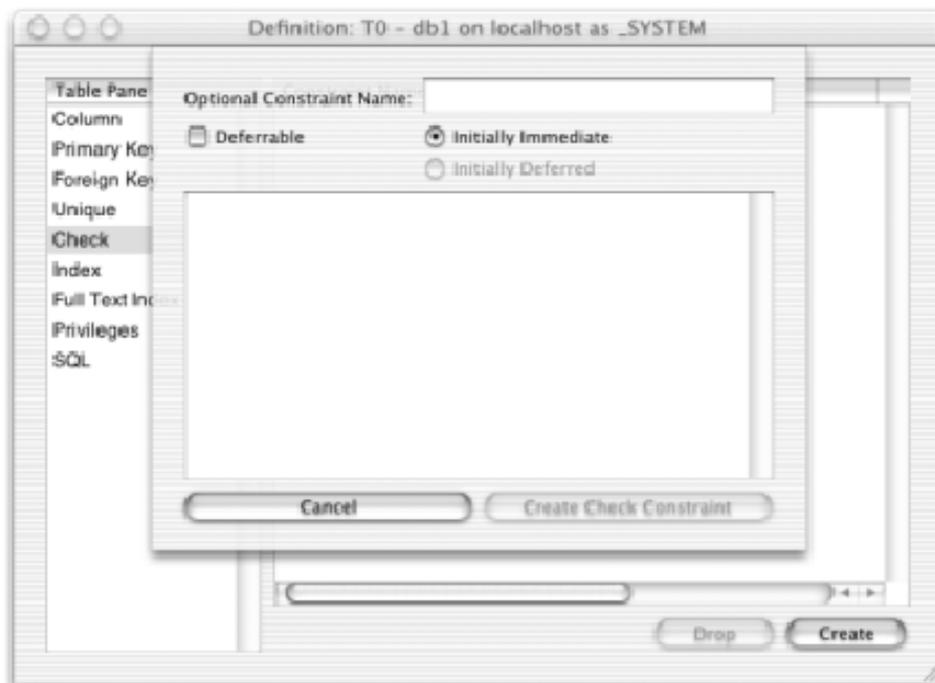
ここからあなたが（任意の）制約名を指定して、制約が延期可能であるか否かにかかわらず、そして、もしそれが延期可能であるなら、それが初めに延期されるか否かにかかわらず設定することができます。あなたはその時どの行が唯一の制約に参加するか指定して、そして「唯一の制約を作る（Create Unique Constraint）」ボタンをクリックすることができます。

唯一の制約をドロップするために、ただ制約を選択して、そして「ドロップ（Drop）」ボタンをクリックしてください。

チェックパネル



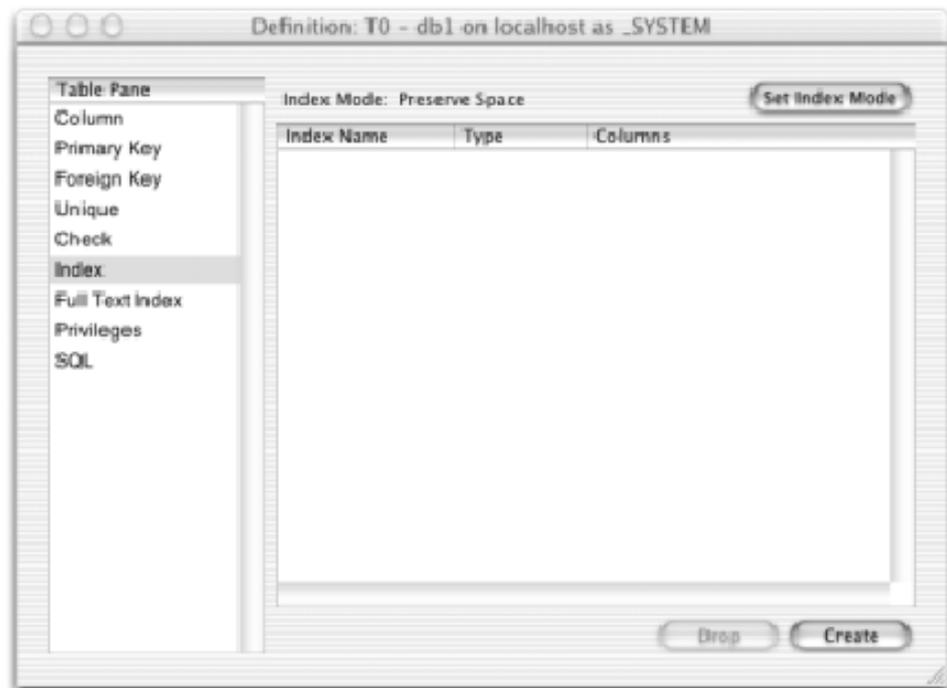
表の定義のウィンドウのチェックの定義のパネルはチェックの制約を作成して、そしてドロップすることを可能にします。チェックの制約を作成するためには、チェックの作成シートを出すために「作成(Create)」ボタンをクリックしてください。



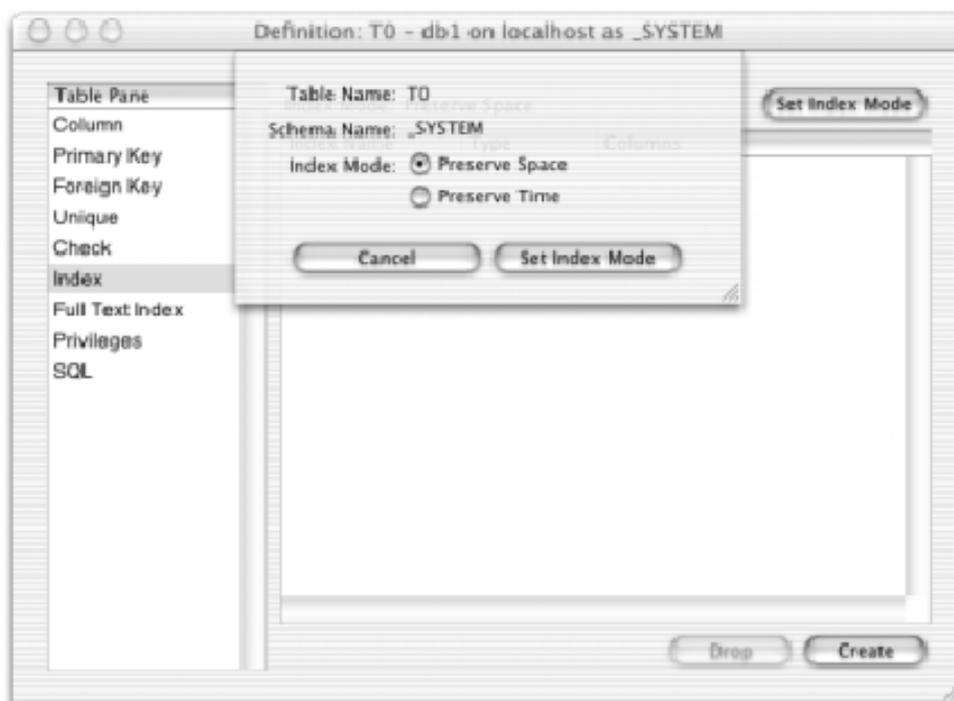
ここからあなたが（任意の）制約名を指定して、制約が延期可能であるか否かにかかわらず、そして、もしそれが延期可能であるなら、それが初めに延期されるか否かにかかわらずセットすることができます。あなたはその時、定義範囲の自由な形式のチェックの制約を指定することができます。（int column1 の「column1 > 3」のように、値 > 3 だけを強制することを望みます）。あなたはチェックの制約を作成するために、「チェックの制約の作成（Create Check Constraint）」がクリックできるようになります。

チェックの制約をドロップするためには、制約を選択して、そして「ドロップ（Drop）」ボタンをクリックしてください。

索引のパネル

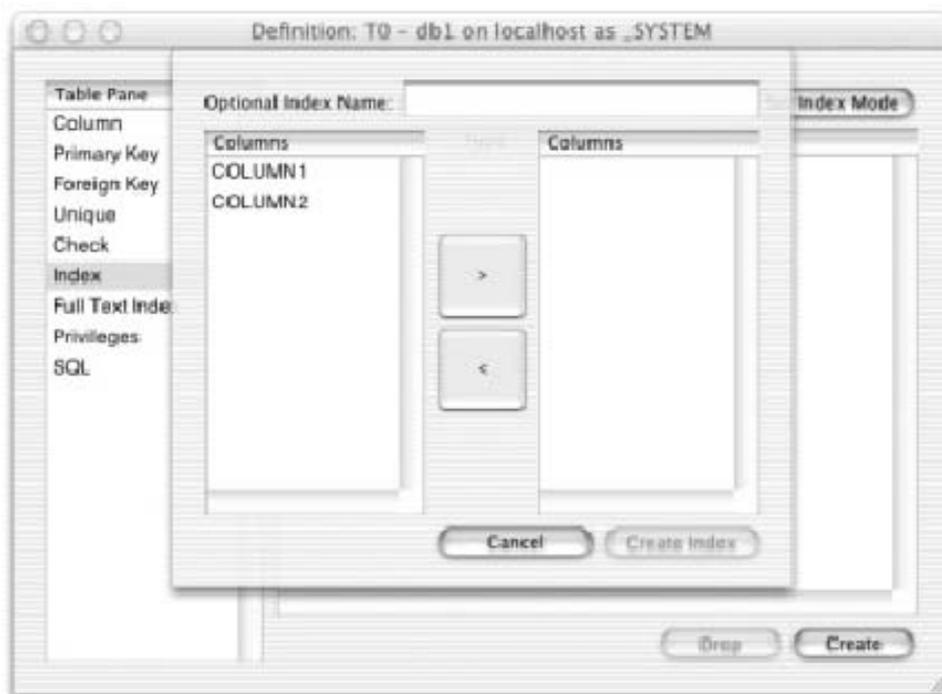


インデックスの定義のパネルはあなたが決まった場所で現在持っているすべてのインデックスを示して、あなたに索引モードを設定することを可能にして、あなたに既存の索引をドロップして、そして新しいインデックスを作成することを可能にします。



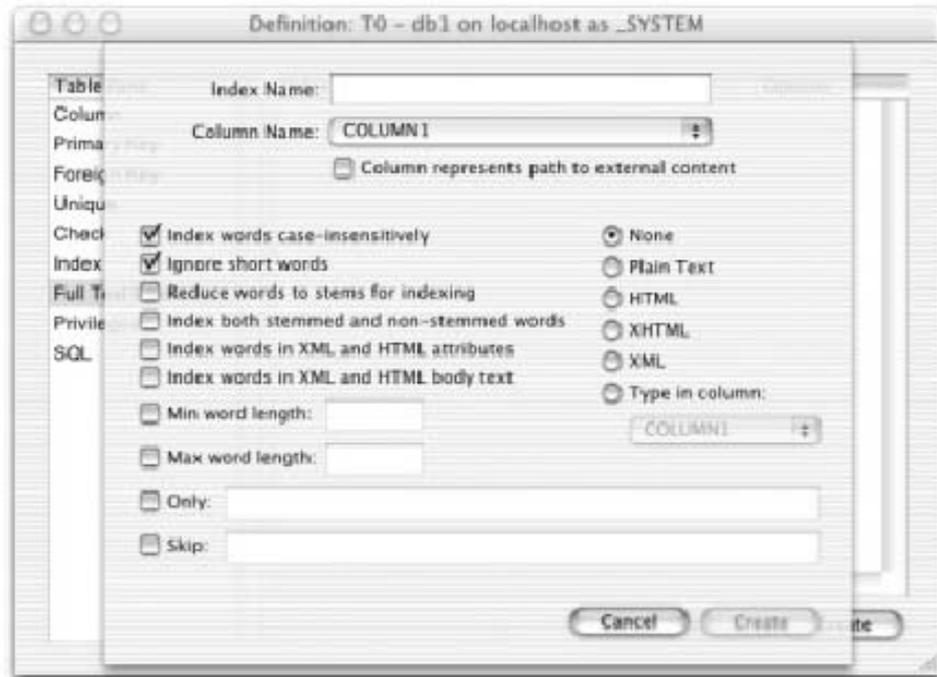
索引モードを設定するために、「索引モードを設定する (Set Index Mode)」ボタンをクリックしてください、そうすればあなたは「持続するスペース」あるいは「持続する時間」の選択を使って与えられるでしょう。あなたの選択をします、「索引モードを設定する (Set Index Mode)」をクリックしてください、そうすれば FrontBaseManager はあなたの索引モードを変更するでしょう。あなたは、索引の設定を変更するためにシリアルな (Serializable)、悲観的な (Pessimistic) モードのはずです。

インデックスをドロップするために、インデックスを選択して、そして「ドロップ (Drop)」をクリックしてください。あなたは索引をドロップするためにシリアルな (Serializable)、悲観的な (Pessimistic) モードであるはずで



索引を作るために、「作成 (Create)」ボタンをクリックしてください。このシートが来るとき、あなたは (任意の) 索引名を指定して、そして次に、どの行が索引に参加して、そして「索引を作成 (Create Index)」ボタンをクリックするか指定することができます。あなたは索引を作るためにシリアルな (Serializable)、悲観的な (Pessimistic) モードであるはずで

全文の索引

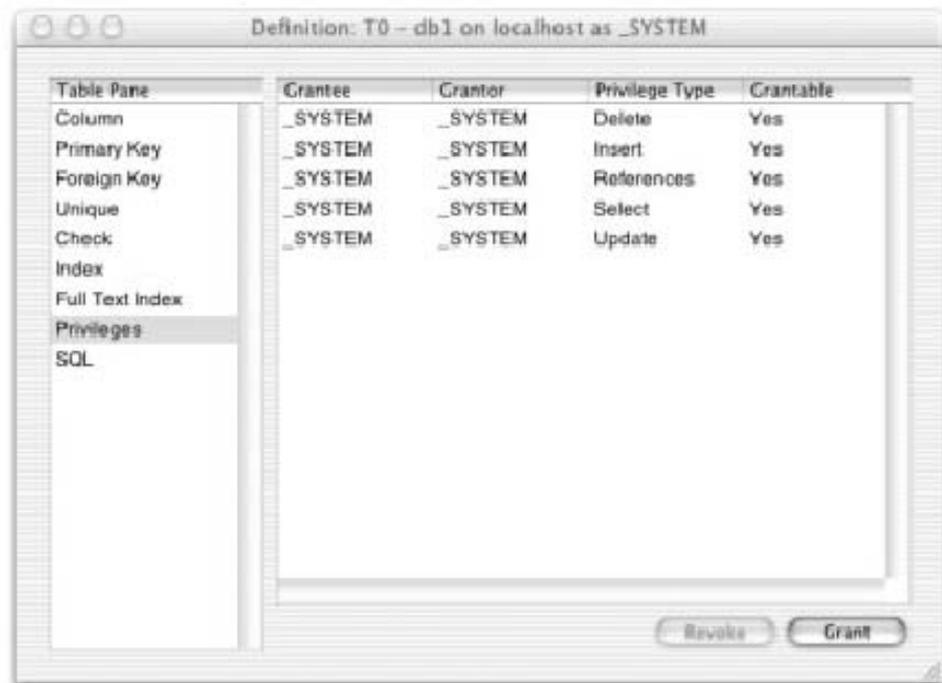


全文の索引のパネルはあなたが全文の索引を作成して、そしてドロップすることを可能にします。

全文の索引を作成するために、全文の索引の作成シートを出すために「作成 (Create)」ボタンをクリックしてください。全文の索引のために、あなたは索引名を指定してください。次に、行の名前を指定してください。すべての異なった全文の索引のオプションの説明のために、LookSee 文書を見てください。（LookSee はエンジンに索引を付ける全文です。）オプションを指定した後で、全文の索引を作成するために「作成 (Create)」をクリックしてください。

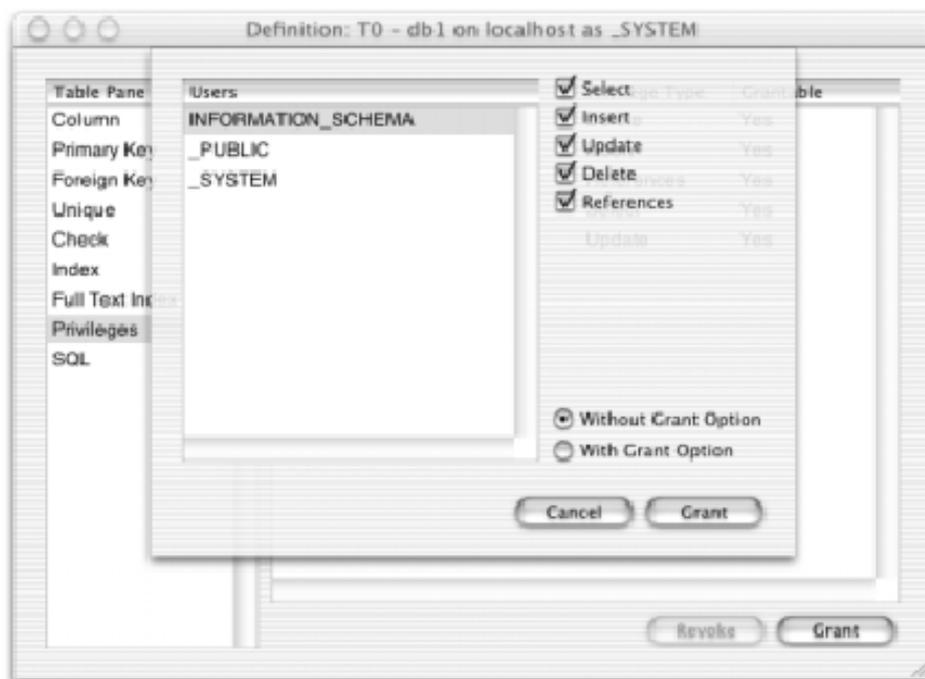
全文の索引をドロップするために、インデックスを選択して、そして「ドロップ (Drop)」ボタンをクリックしてください。

特権



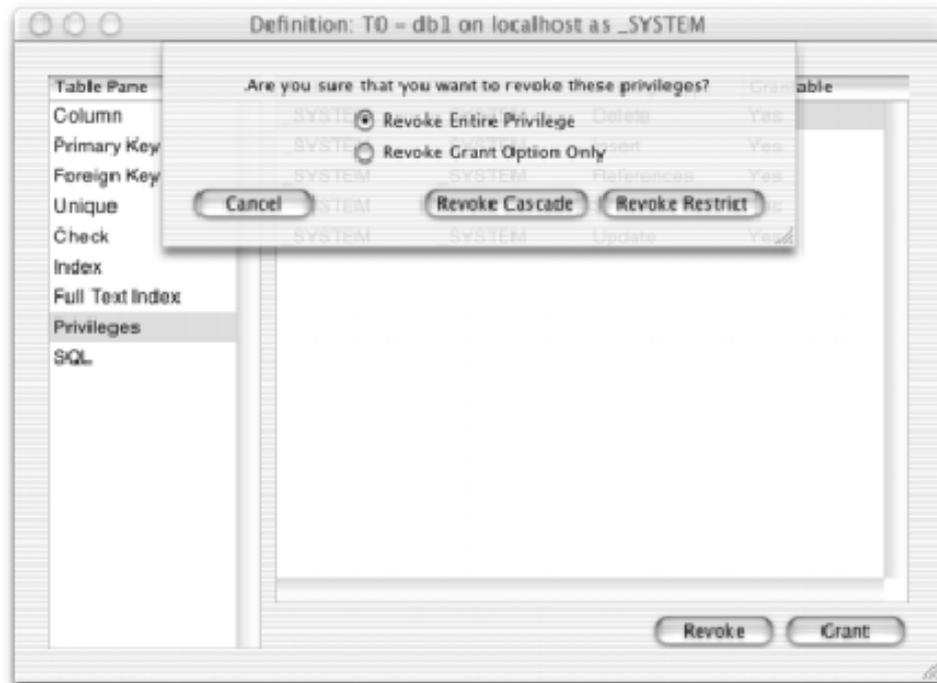
特権のパネルは現在データベースに与えられた特権を示します。行は被特権者 (Grantee)、特権者 (Grantor)、タイプ (Type) と特権ができる (Grantable) です。

特権ができる (Grantable) は被特権者が他のユーザに特権を与えることができるかどうかを意味します。特権を与えるために、接続のユーザが特権の認可を持っていることを確認して、そして「認可 (Grant)」ボタンをクリックしてください。



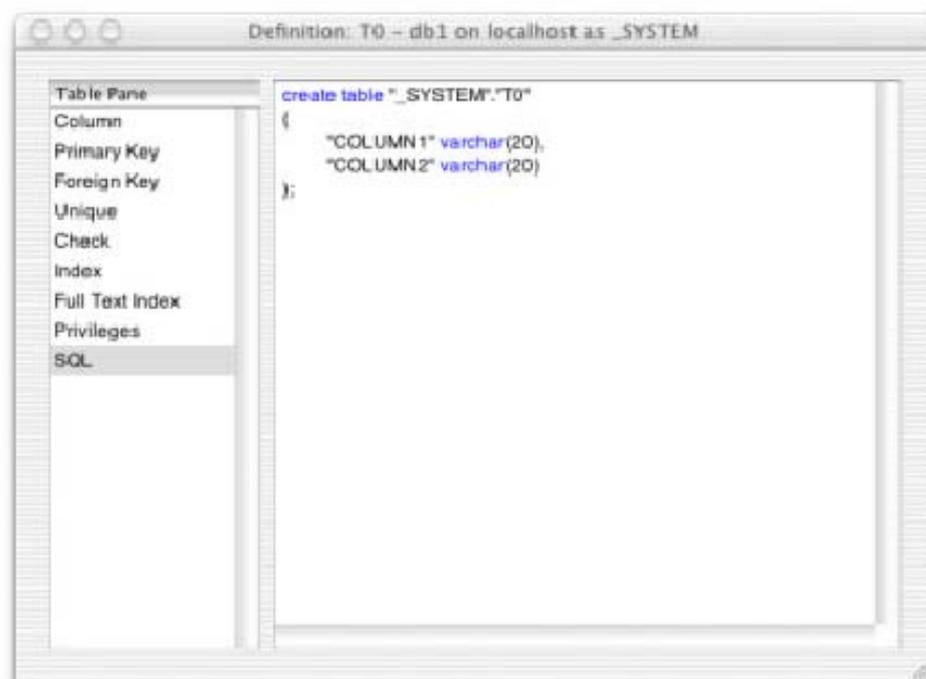
あなたは左のブラウザですべてのデータベースユーザに会うでしょう。被特権者 (grantee) を選択して、与えることを望むすべての特権タイプをチェックします、認可オプションなし (デフォルト) で、あるいは認可オプションで、与えることを望むかどうか選択します、そして「認可 (Grant)」ボタンをクリックしてください。

特権を無効にするために、あなたが無効にするのを望む特権を選択します、そして「無効 (Revoke)」ボタンをクリックしてください。



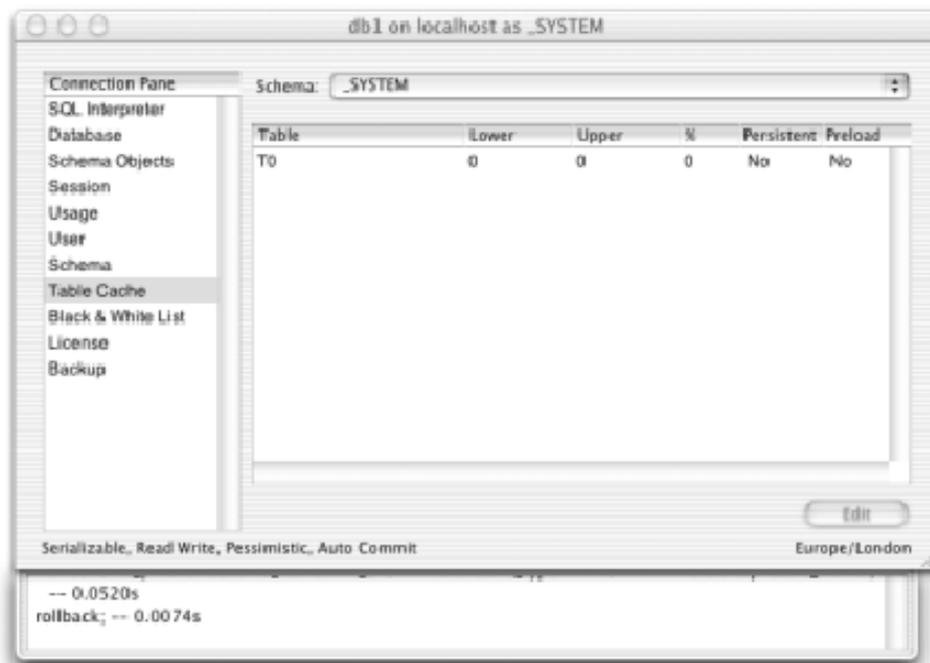
(ユーザが他のユーザに特権を与えることができないように)、あなたは全部の特権を無効にするか、あるいは認可オプションのみを無効にする、のいずれかの選択を与えるでしょう。もしあなたが限定することを無効にするなら、それはあなたが選択した特権を無効にするだけでしょう。もしあなたがカスケードを無効にするなら、それはあなたが選択した特権とその特権を使って与えたすべての特権を無効にするでしょう。(もし BOB が JIM に特権を与え、そして JIM は、SUE に、特権を与えるなら、BOB の特権のカスケードを無効にすることは BOB の、JIM の、そして SUE の特権を無効にするでしょう。)

SQL



このパネルは表を作った SQL を示します。

表のキャッシュ



表のキャッシュのパネルは表のキャッシュについての情報が入っています。

表のキャッシュは表をキャッシュすることを微調整するために使うことができます。選択されたスキーマのすべての表はこのパネルで示されます。パラメータは次の通りです：

下 (Lower) - キャッシュの保持する表の列の最小の数

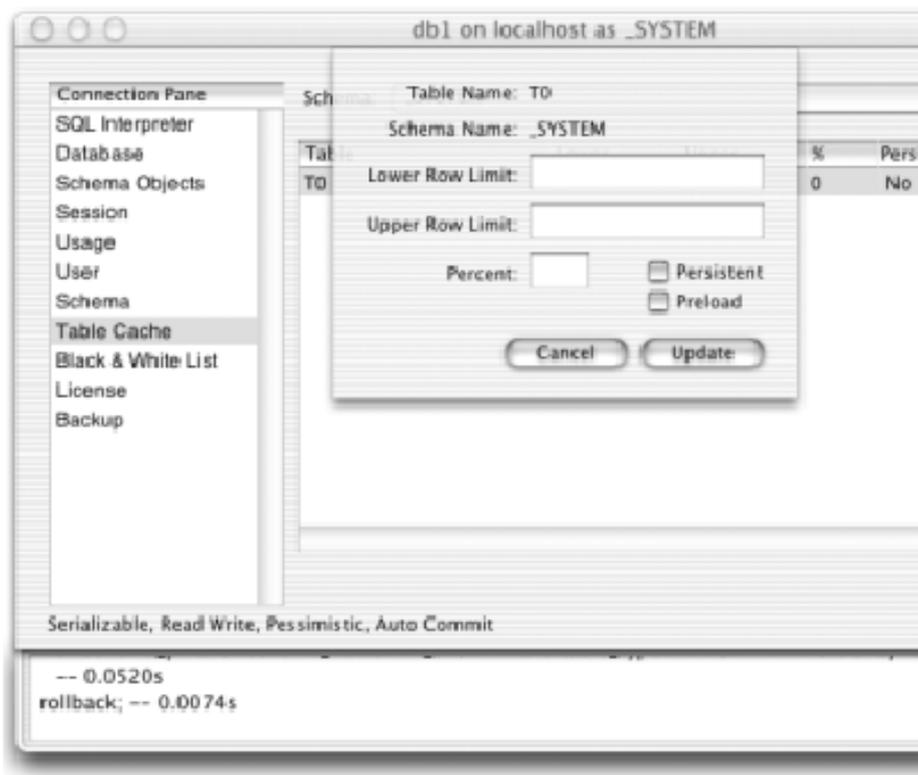
上 (Upper) - キャッシュの保持する表の列の最大の数

% キャッシュの保持する列の合計の数のパーセンテージ

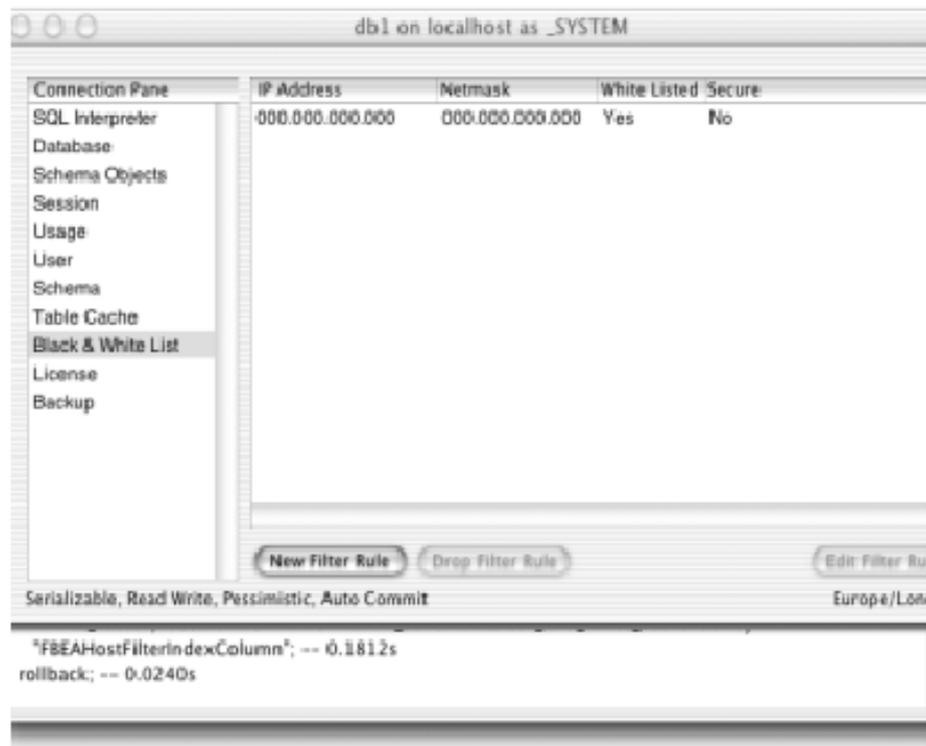
持続的 (Persistent) – もしイエスであるなら、キャッシュはトランザクションの向こう側に保持されます。そうでないときは、キャッシュはそれぞれの COMMIT あるいは ROLLBACK の後にクリアされます。

前もってロードする (Preload) – もしイエスであるなら、サーバーをスタートするとき、表のすべての列はキャッシュにロードされるでしょう。そうでないときは、列はそれらに最初のリファレンスをロードします。

これらの設定を変更するために、あなたが変更することを望む表を選択して、そして「編集 (Edit)」ボタンをクリックしてください。

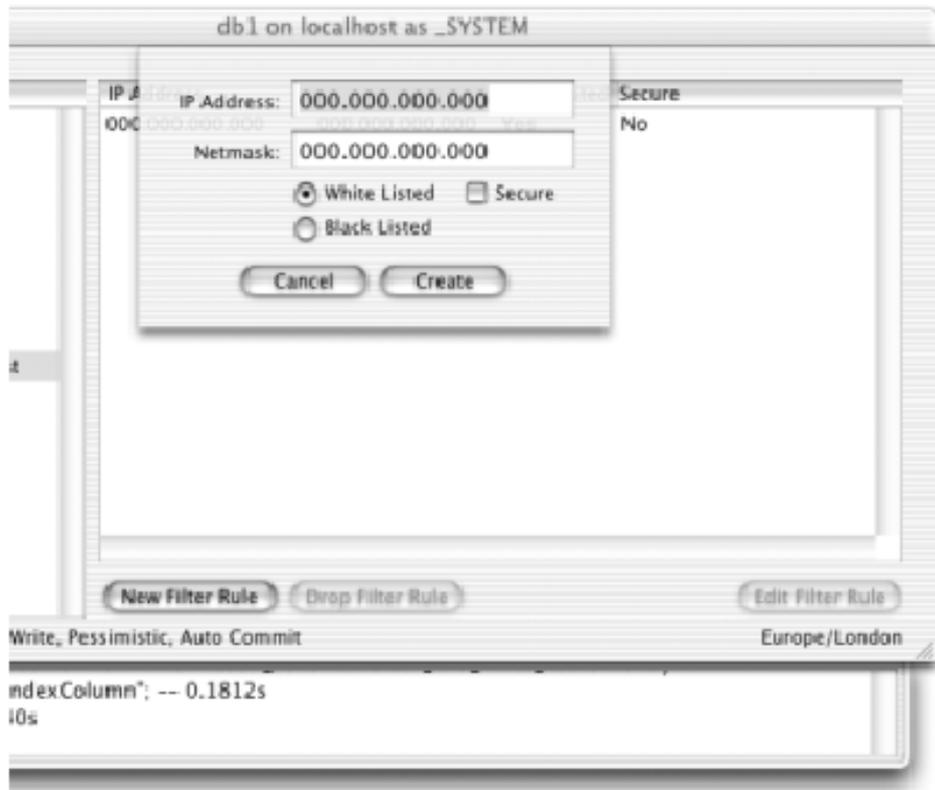


ブラックリストとホワイトリスト



パネルはブラックリストとホワイトリストについての情報が入っています。クライアントが FrontBase サーバーに接続するとき、その IP アドレスは、クライアントが接続することを可能にするかどうか見る、ブラックリストとホワイトリストと照らし合わせます。ブラックリストとホワイトリストのそれぞれの項目のために、クライアントの IP アドレスは netmask を使って AND をとります、そして、結果はブラックリストとホワイトリストの IP アドレスを比較します。もし一致するものを見つけて、そして項目がホワイトリストであるなら、クライアントは接続することを可能にします。そうでないときは、接続は拒否されます。もし一致するものが見つけられないなら、接続は拒否されます。もしイエスが安全なフィールドに入力されるなら、クライアントはサーバーを使ってすべてのそれ以上の通信のために暗号を使うようにリクエストします。

新しいフィルターの規則を加えるために、「新しいフィルターの規則 (New Filter Rule)」 ボタンをクリックしてください。



あなたがホワイトリストあるいはブラックリストに望むホストあるいはネットワークを記述する netmask と IP アドレスを入力してください。ラジオボタンからホワイトリストあるいはブラックリストを選択して、そして、もしクライアントがそれらのコミュニケーションのために暗号を使うことをリクエストすることをサーバーに望むなら、安全な (Secure) チェックボックスをクリックしてください。「作成 (Create)」ボタンをクリックしてください、そうすればあなたは新しいリスト項目を持つでしょう。

フィルターの規則をドロップするために、あなたがドロップすることを望む規則を強調して、そして「フィルターの規則をドロップする (Drop Filter Rule)」ボタンをクリックしてください。

フィルターの規則を編集するために、あなたが編集することを望む規則を強調して、そして「フィルターの規則を編集する (Edit Filter Rule)」ボタンをクリックしてください。シートがあなたにフィルターの規則を編集することを可能にして来るでしょう。あなたが変更することを望む値を変更した後、その変更を保存するために「アップデート (Update)」をクリックしてください。

フィルターの規則がブラックリストとホワイトリストの上からリストの下まで評価します。一致するものが見つかるとすぐに、評価がストップします、そして規則が適用されます。それで、もしホワイトリストである IP 000.000.000.000/Netmask 000.000.000.000 を持っているなら、あなたが最初のホワイトリスト項目に何個かのブラックリスト項目を持っていても、すべての単独の IP が連結することが可能でしょう。どのようにブラックリストとホワイトリストを使うかについて、ここに若干の事例があります

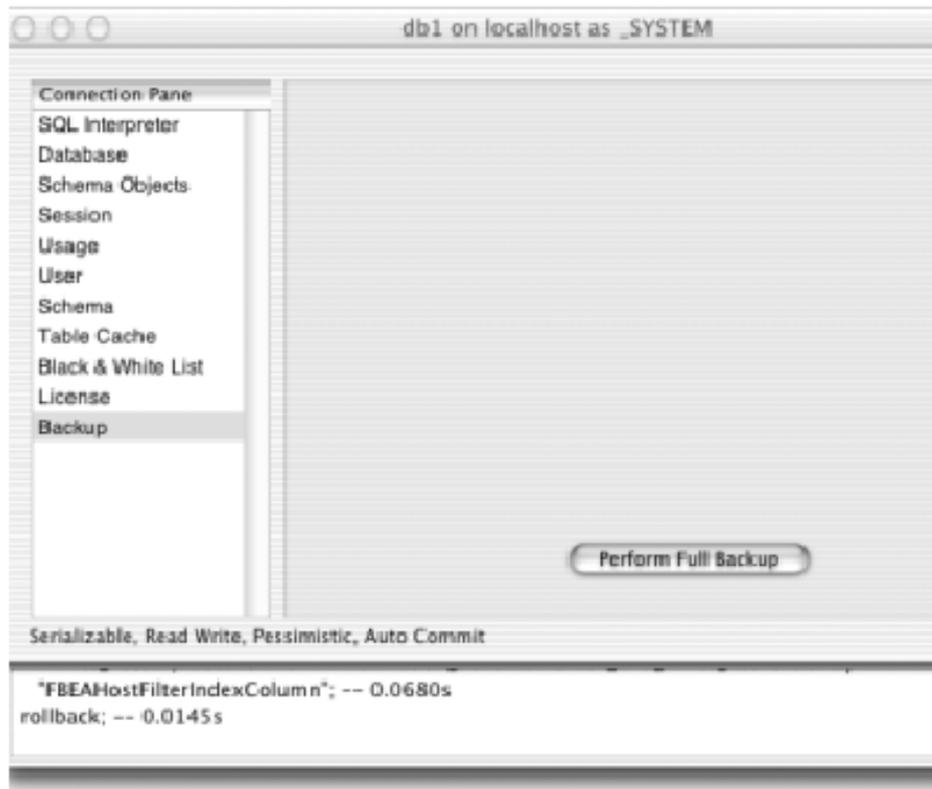
IP	ネットマスク	ホワイトリスト	ソース
24.5.126.145	255.255.255.255	いいえ	いいえ
000.000.000.000	000.000.000.000	はい	いいえ

これは 24.5.126.145 で単独の IP をロックアウトするでしょう、そしてそれ以外のすべての IP から接続を可能にするでしょう。

IP	ネットマスク	ホワイトリスト	ソース
24.5.126.0	255.255.255.0	YES	NO
24.5.127.0	255.255.255.0	YES	NO
128.0.64.2	255.255.255.255	YES	NO

これは 2 つのクラスの C ネットワーク (24.5.126.0-255 と 24.5.127.0-255) と単独の IP アドレス (128.0.64.2) からアクセスを可能にするでしょう、そしてそれ以外のすべての IP の接続を拒絶するでしょう。

バックアップ

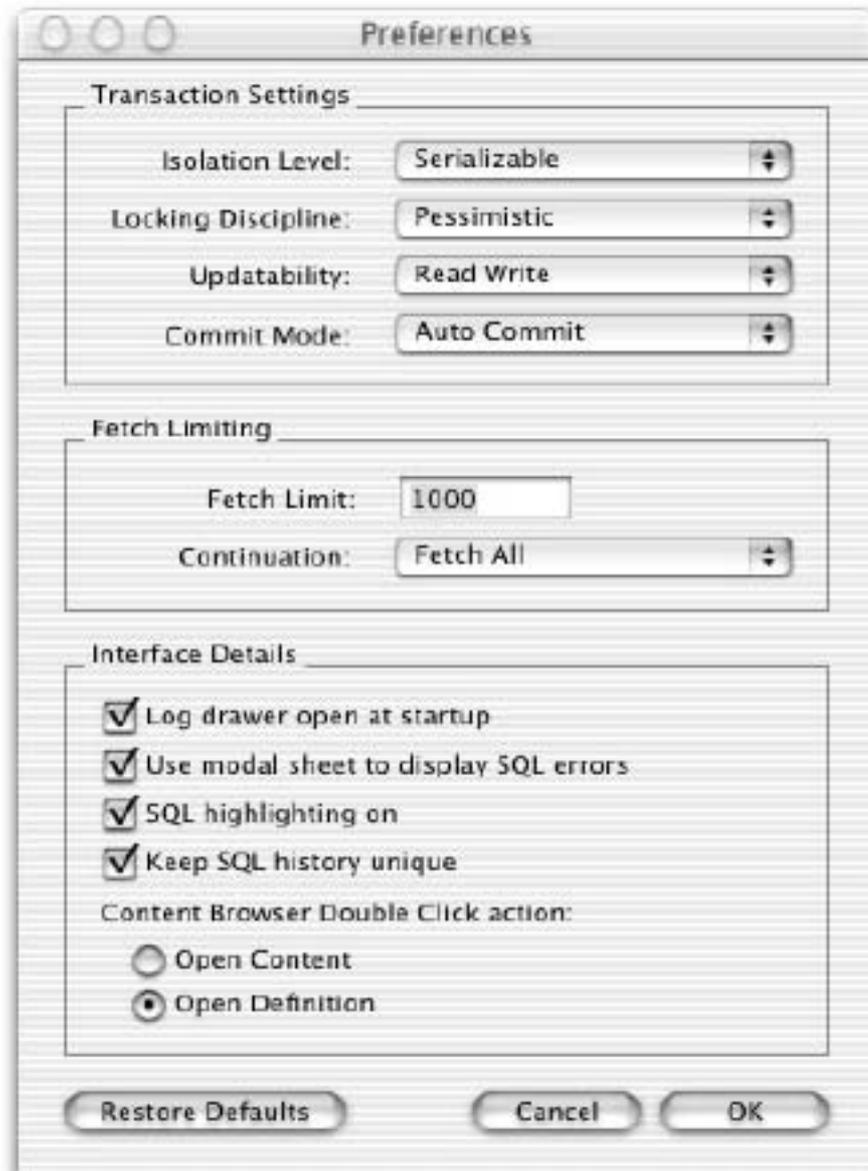


バックアップのパネルは「フルバックアップを行う (Perform Full Backup)」ボタンを持っています。もしあなたがボタンをクリックするなら、シートは「バックアップを始める (Begin Backup)」ボタンを使って来るでしょう。それは同様にサーバーに増加する復帰のためにトランザクションログをとるよう指示するチェックボックス (デフォルトからチェック) を持っています。

奇数 n の終わり

すべての FrontBaseManager の特徴が主な接続ウィンドウパネルを通してアクセス可能であるわけではありません。いくつかはプルダウン・メニューあるいはキーボード・ショートカットからアクセスされます。このセクションはそれらの特徴を詳述します：

プリファレンス



FrontBaseManager の選択システムはあなたにファクトリーのデフォルトから望む準備を交換し続けなくてもよいように起動時のトランザクション設定 (transaction settings)、フェッチの制限 (fetch limiting)、そしてインターフェースの詳細 (interface details) の少しを指定することを可能にします。あなたが新しいデータベース接続

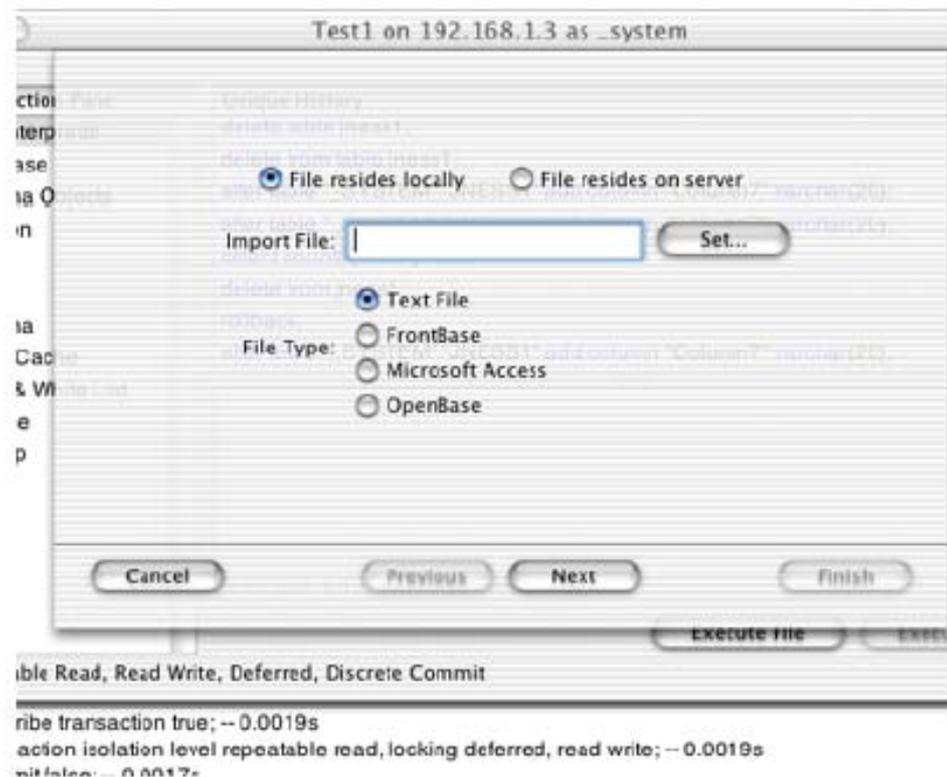
をスタートする時はいつも、あなたのプリファレンスで指定したトランザクションの設定 (transaction settings) はセットされるでしょう。

あなたが新しいデータベース接続をスタートするとき、あなたのフェッチの限界 (fetch limit) は同様に設定されるでしょう。継続 (continuation) の動作はサーバーにそれがフェッチの限界に的中するとき何をしたらいいか示します。2つの選択はフェッチをストップするか、あるいはすべての利用可能な列をフェッチします。

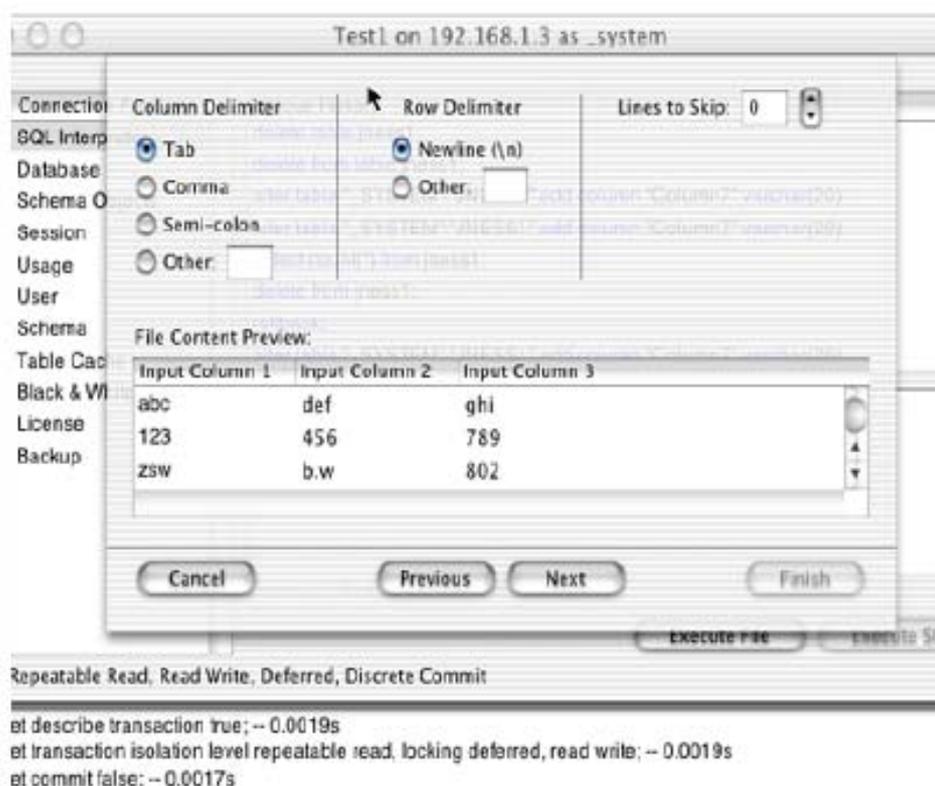
インタフェースの詳細のセクションは、あなたに、接続ウィンドウが最初に開くとき、開くあるいは閉じる SQL ログの引き出し (drawer) を必要とするかどうか指定することを可能にします。(あなたは、もちろん、手作業でそれを変更することができます。そうするには、セクション 4.2 を見てください。) SQL で際だっていることは、あなたにどの単語が SQL92 で予約しているキーワードであるか示すことによって、大いに SQL92 の構文エラーを減らすことができる強力な特徴です。(これの1つの利益は、もしそれらが SQL92 で予約しているキーワードと同じ名前をつけるなら、行の名前をコーテーション (“ ”) で囲むよう思い出させます。) 最終的に、あなたは唯一の SQL インタプリターの SQL ヒストリーを取っておくためにプリファレンスをセットすることができます。すなわち、もしあなたが何か他の SQL 文を「select * from foo;」に出し、それから、再び「select * from foo;」を出すなら、あなたはヒストリーの2番目の「select * from foo;」を得ないことを意味します。その代わりに、SQL ヒストリーはすでにヒストリーリストの下のヒストリーで「select * from foo;」を動かすでしょう。

ファイルの読み込み

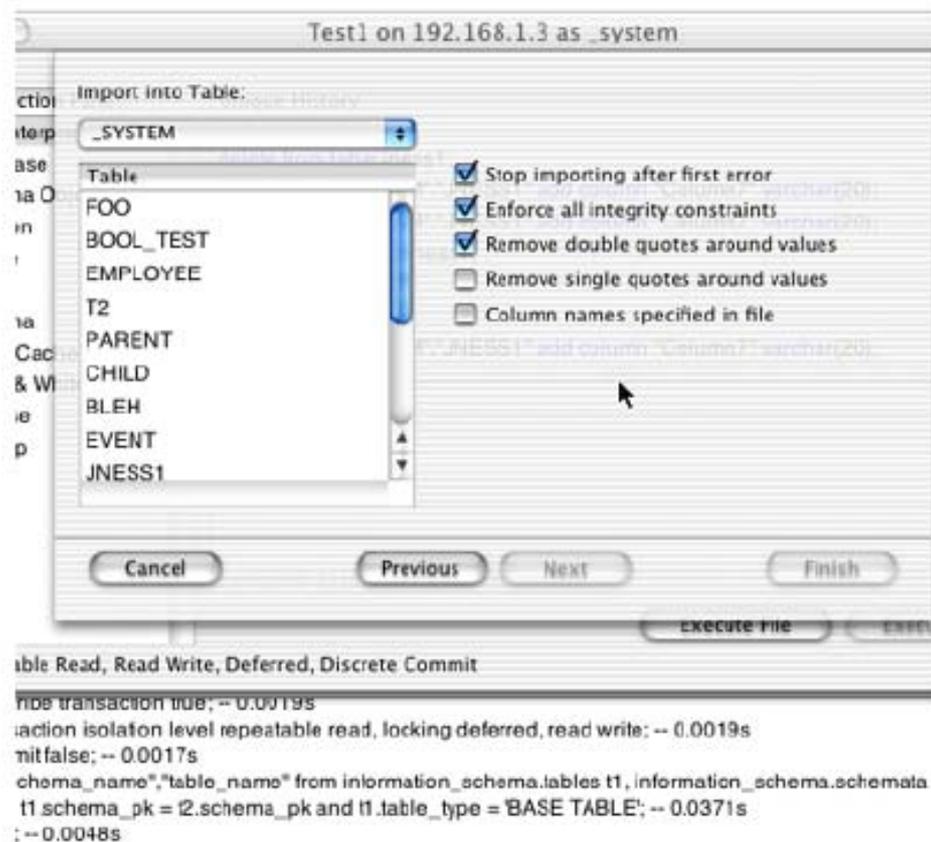
FrontBaseManager バージョン <バージョン番号> が FrontBase の強力な読み込みの機能性を利用するために最初に直観的なユーザインタフェースを導入しました。読み込みの補助のインタフェースにアクセスするために、File->Import を選択してください。



この最初の画面はあなたにファイルがどこにあるか、そしてファイルが何の一般フォーマットであるか指定することを可能にします。あなたはローカルのクライアントマシンであるか、あるいはデータベースサーバーであるファイルから選択することができます。（もしサーバーがローカルマシンであるなら、この選択はディスエーブルになるでしょう。）もしファイルがローカルに存在するなら、あなたは同様にファイルのためにローカルなファイルシステムをブラウズする "Set..." ボタンを使うことができます。「ファイルタイプ」はあなたにファイルの一般フォーマットを指定することを可能にします。もしそれが FrontBase の書き出しであるなら、FrontBase を選択してください。もしマイクロソフト・アクセスでファイルを作り出したなら、マイクロソフト・アクセスを選択してください。たいていの書式のないファイルの読み込みのために、あなたはテキストファイルを選択することを望むでしょう。ファイルの場所とタイプを選択した後で、次の画面に進むために「次に (Next)」をクリックしてください。

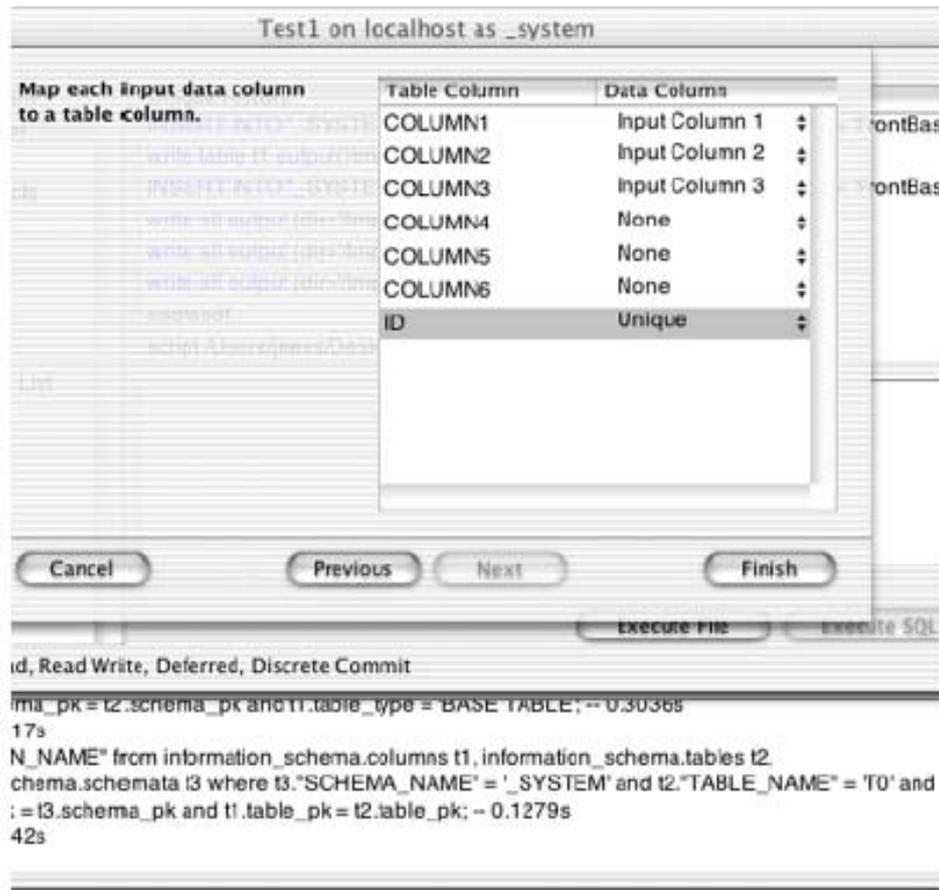


あなたがファイルが位置しているところと一般フォーマットを指定した今、あなたは最初のファイルの行のプレビューを得るでしょう。これは同様にあなたが行及び列の区切り文字（それぞれの入力ファイル行及び列の間のセパレータ）を指定する画面です、そして読み込みプロセスが省略するファイルのトップの行です。あなたが区切り文字と行をスキップするように変更するとき、あなたのプレビューはデータがどのように読み込んでいるか示すために変更するでしょう。あなたはこの読み込みの行き先を選択するために次の画面で設定できます。



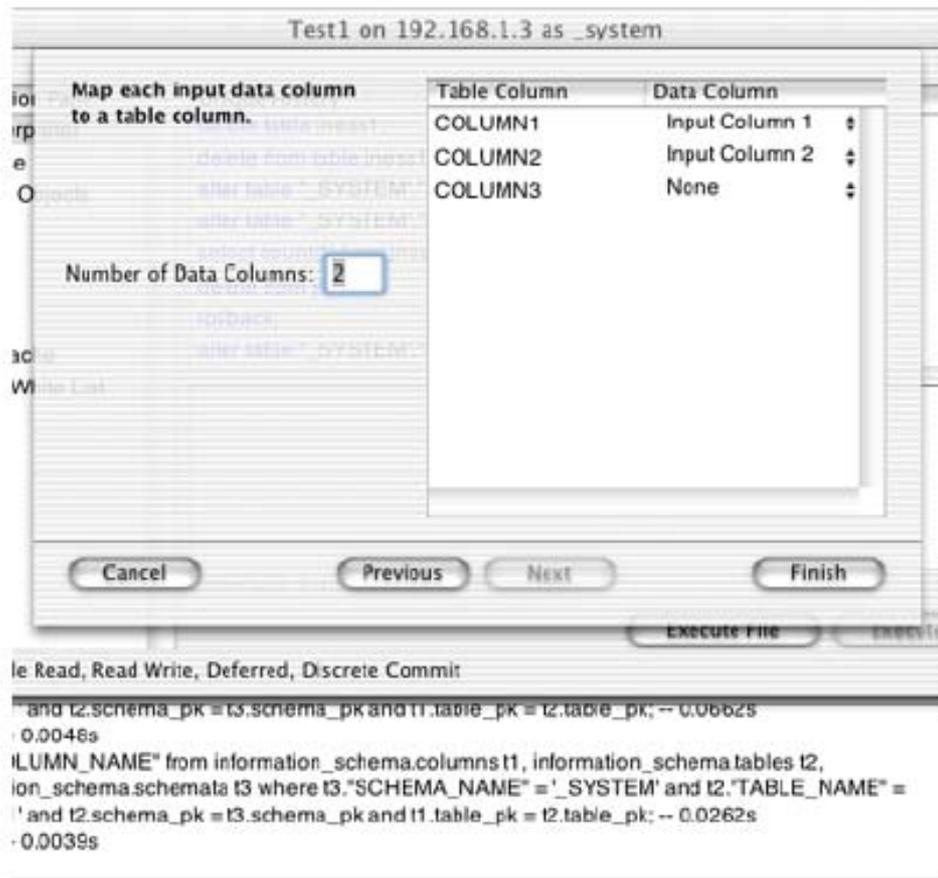
この画面であなたはこの読み込みでどの表を使うか選ぶでしょう。もしあなたが望むなら、あなたは他のスキーマからの表にアクセスするために表のリストの上にドロップしてスキーマを変更することができます。この画面にいくつかの読み込みオプションが同様にあります。あなたは、もしそれがエラーに遭遇するなら、プロセスがストップするか、データベースサーバーが（唯一の制約とチェックの制約のような）すべての完全な制約を実施するか、シングルあるいはダブルコーテーションが、ファイルの値の周りに、取り除かれるか、そして行の名前がファイルで指定されるかどうかを選択することができます。FrontBase と OpenBase のタイプの読み込みはファイルで指定した行の名前を使って明瞭なファイル形式を持っています、それであなたはこの画面で「終了 (Finish)」をクリックすることができます、そしてあなたは終わるでしょう。もしあなたのファイルがマイクロソフト・アクセスで作ったなら、行の名前は、「ファイルで指定した行の名前 (Column names specified in file)」チェックボックスをチェックして、そして「終了 (Finish)」をクリックすることができます。

できるように、ファイルを指定します。しかしながら、もしあなたがファイルに指定した行の名前なしでテキストファイル、書式のないファイル、あるいはマイクロソフト・アクセスのファイルを読み込んでいるなら、あなたはこの読み込みのために使う行を指定する必要があります。それは次の画面でできます。

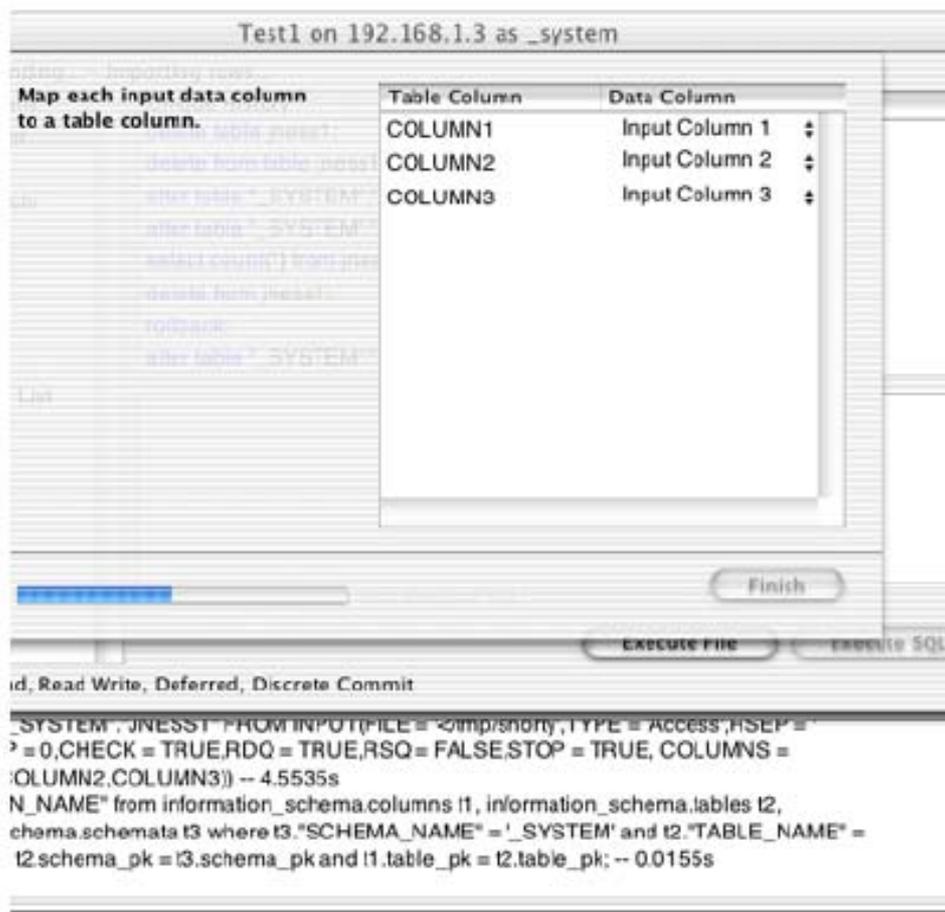


この画面で、あなたは1つの「表のコラム」を使ってそれぞれの「入力する行 (Input Column)」と一致する必要があるでしょう。すべてのデータベースの表の行が指定した入力する行 (input column) を持つ必要があるわけではありません、けれどもすべての入力する行 (input column) がそのために指定した表の行を持っているはずで、もしあなたが望むなら、あなたは表の行の1つを「唯一」に割り当てることができます。この表の行は INTEGER タイプの行であるはずで、そして読み込みのそれぞれの列について唯一の値を得るでしょう。もしデータファイルがローカルにあるか、あるいはサーバーにある、しかしあなたのサーバーがローカルにあるなら、

FrontBaseManager はファイルを開いて、そして、ファイルが持っている入力する行を何個数えることができるでしょう。しかし、もしファイルがデータベースサーバーに存在し、そしてクライアントがファイルにアクセスすることができないなら、あなたはこのファイルに入っている入力する行を何個指定する必要があるでしょう。もしその場合であるなら、ここにあなたが見る画面があります：



データの行の番号（この場合2）を変更することはあなたが右側で見える入力する行を何個変更するか注意してください。あなたが表の列を使ってそれぞれの入力する行と一致したら、「終了」をクリックします、そうすると、あなたの読み込みがスタートするでしょう。もしファイルがローカルに存在するなら、あなたは次の画面のようなプログレスバーを見るでしょう：



もし読み込みの補助があなたの読み込みファイルを動作しないなら、問題の詳細を tools@frontbase.com に送ってください。

SQL のログ

SQL のログの引き出し (drawer) は正確に何の SQL がサーバーに送られるか知る非常に便利な方法です。しかしながら、もしあなたが望むなら、あなたは、View->Hide SQL ログを選択することによって、あるいはコマンド - L. を的中することによって、SQL のログの引き出し (drawer) の下の端から接続ウィンドウへ、ドラッグすることによって、SQL のログの引き出し (drawer) を閉じることができます。もしログの引き出し (drawer) が隠されるなら、View->Show SQL ログを選択することによって、あるいはコマンド - L.

を押すことによって、あなたは再びそれを見せることができます。コマンド - K を押すことによって、あなたはいつでも SQL のログをクリアすることができます。

FBWebManager

この章は FrontBase のすべてのインストールのために利用可能であって、そして（JavaScript としても知られている）ダイナミックな HTML をサポートするブラウザを通してアクセスできる FBWebManager Web ベースのアプリケーションを紹介します。

この章は次のことが入っています：

- 205 ページの「データベースを監視する」。
- 209 ページの「データベースを与える」。
- 214 ページの「データベース管理」。
- 234 ページの「ライセンスを加える」。

データベースを監視する (Monitoring)

FBWebmanager はあなたに FrontBase サーバーの状態を監視して、そしてコントロールさせます、すなわちそれらが走っているか、あるいはストップしたかにかかわらず。FBWebManager によって監視されたデータベースはクライアントのコンピュータでクッキーに記憶します、それで、FBWebmanager がアクセスする次のとき、ユーザは同じデータベースを見るでしょう。FBWebManager を使うために、あなたはサーバーで走っているアパッチのような Web サーバを必要とするでしょう。クライアントのブラウザを FBWebManager と接続するために、あなたは URL を提供する必要があるでしょう。これは、例えば、フォームをとることができました：

`http://localhost/cgi-bin/FBWebManager`

ここは、「localhost」がサーバマシンのホスト名です。

データベースを加える

FBWebmanager が所定のユーザによって最初にスタートするとき、次の「からの」ウィンドウが図 8.1 に現れます。

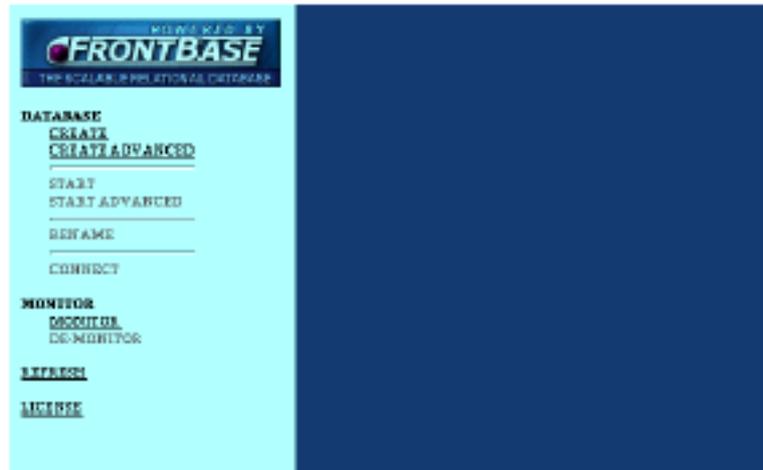


図 8 . 1 WebManager をスタートする

監視されたデータベースが、データベースとデータベースが位置しているホストコンピュータの名前を使って一緒に、データベースの状態を指摘して、アイコンとして示します。下の事例は WhitePages と Pitsaw が macosx と呼ばれるホストで見つけたデータベースを示します。図 8 . 2 を見てください。

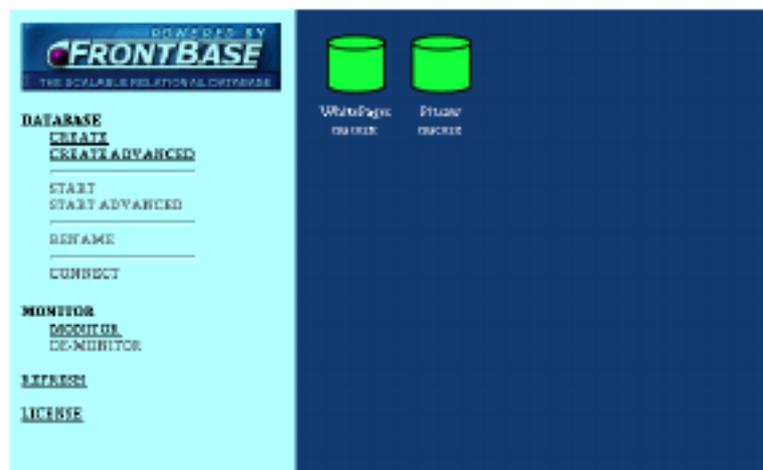


図 8 . 2 監視されたデータベース

モニターにデータベースを加える

図8.3のように、データベースセレクトビューを出すメニュー項目 MONITOR->MONITOR をクリックすることによって、データベースがモニタービューに加えられます。

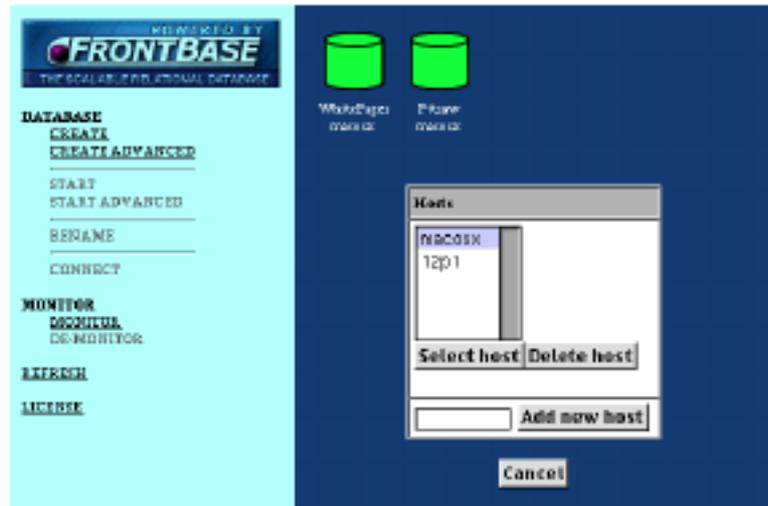


図8.3 データベースを加える

もし欲しいホストがリストにないなら、入力フィールドにホスト名を入力して、そして、新しいホストを加える (Add new host) をクリックしてください。それから、ホストはリストに加えられます。リストから欲しいホストを選択して、そしてホストを選択する (Select host) をクリックしてください。図8.4のように、これは指定したホストでデータベースのリストを起動するでしょう。

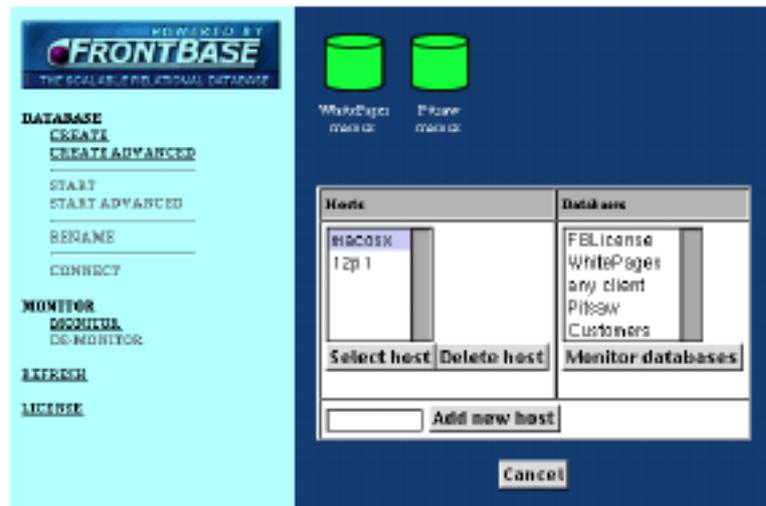


図 8 . 4 ホストを加える

データベースを監視するために選んで、そしてデータベースを監視する (Monitor databases) をクリックしてください。
これは監視したデータベースのリストにデータベースを加えるでしょう、図 8 . 5 を見てください。



図 8 . 5 データベースを選択する

データベースから取り除く

データベースをモニタービューから取り除くために、あなたは最初にそれを選択する必要があります。データベースは真っ新のバックグラウンドで与えられます。データベースを監視しないためには、メニュー項目 MONITOR->DEMONITOR をクリックしてください。この動作がデータベースをストップも、そしてデリートもしない、それはモニタービューから取り除かれるだけです。

データベースを与える

Web ベースのデータベースマネージャーは理解する方法が容易なデータベースを管理する必要なタスクを実行させます。

データベースを作る

新しいデータベースを作るために、メニュー項目 DATABASE->CREATE をクリックして、そして次のビューが図 8.6 のように表示します。

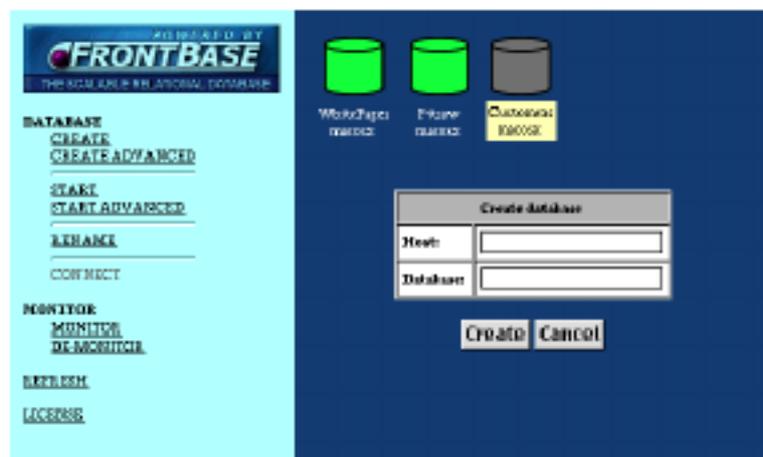


図 8.6 データベースを作る

ホスト名と新しいデータベース名を入力して、そして作成 (Create) ボタンをクリックしてください。図 8 . 7 のように、モニタービューにデータベースを作成して、スタートして、そして、加えます。



図 8 . 7 モニタを加えた

もしあなたがデータベースサーバーにオプションを指定する必要があるなら、メニュー項目の DATABASE->CREATE ADVANCED をクリックしてください。これはあなたに、図 8.8 のように、次のオプションを指定することができるでしょう。

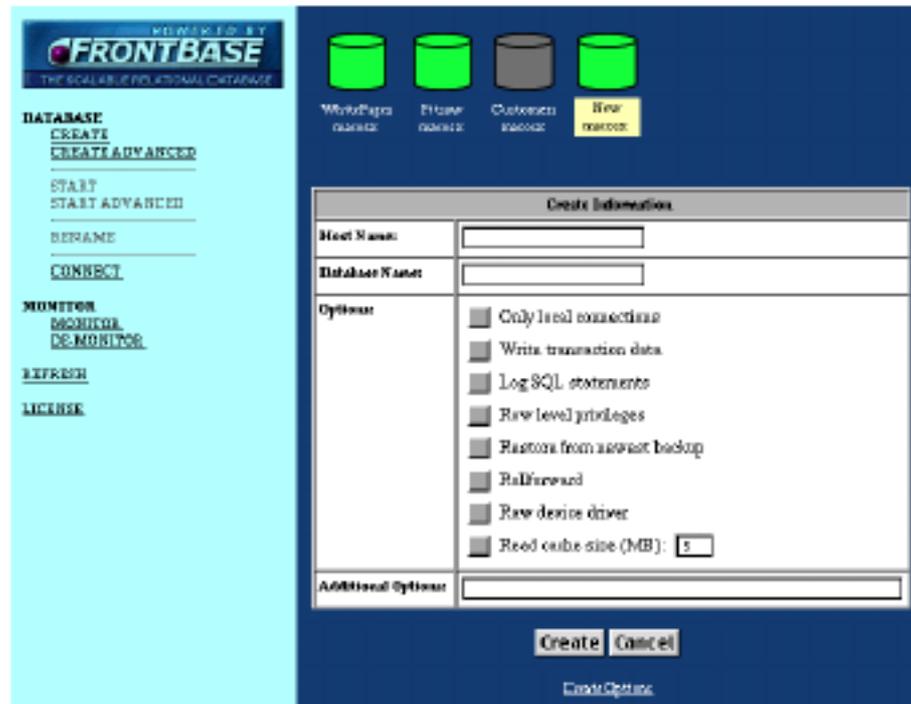


図 8.8 データベースサーバーオプション

1. ローカル接続だけ
データベースサーバーがローカルな接続、すなわちデータベースサーバーが位置している同じホストからの接続を可能にするでしょう。
2. トランザクションデータを書く
データベースサーバーはファイルにすべてのトランザクションをログファイルに書くでしょう。
3. SQL 文をログファイルに書く
データベースサーバーはファイルに実行された SQL ステートメントをログファイルに書くでしょう。ファイルはデータベースディレクトリで見つけて、そして <データベース名>.fb.sql と名前をつけます。
4. 列レベルの特権がユーザが個別の列のためにアクセス権を指定することを可能にします。

ユーザに個々の列のためにアクセス権を指定することを可能にします。

5. 最も新しいバックアップからの復帰

データベースを作成するとき、データの内容は最も新しいバックアップから復帰するでしょう。バックアップを作ることに ついて 89 ページの「トランザクションのログを書き出す」を参照してください。

6. Rollforward

バックアップ・ファイルからデータベースを復帰するとき、このオプションはバックアップが完了したとき、完了しなかったトランザクションを示します。これはバックアップが「トランザクションのデータを使って (WITH TRANSACTION DATA)」オプションを使ってしたことを要求します。

7. 未使用のデバイスドライバ

このオプションはホストの OS のファイルシステムが直接回避することを指定します。この問題についての別個の書類がプロダクションの下にあります。

8. キャッシュサイズを読み込む

「未使用のデバイスドライバ」オプションに関連してこのオプションはキャッシュの読み込むサイズを指定します。

9. 追加のオプション

指定した、上記のほかに、データベースを作るとき、オプションの番号を指定することができます。

データベースをスタートする

すでに作ったが、ストップした、データベースをスタートするために、モニタービューでデータベースを選択して、そしてメニュー項目 DATABASE->START をクリックしてください。もしあなたがデータベースサーバーにオプションを指定する必要があるなら、メニュー項目 DATABASE->START ADVANCED をクリックしてください。これはあなたに、図 8.9 のように、次のオプションを指定することをイネーブルにするでしょう。

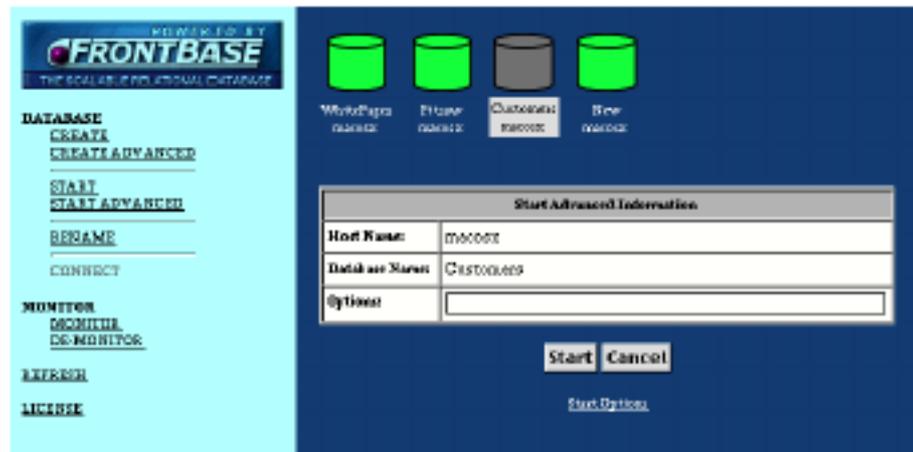


図 8.9 既存のデータベースをスタートする

データベースをスタートするとき、多くのオプションは指定することができます。リンク「オプションをスタートする (Start Options)」をクリックすることによって、オプションは見るすることができます。

データベースを改名する

ストップしたデータベースを改名するために、モニタービューのデータベースを選択して、そして、図8.10で見られるように、メニュー項目 DATABASE->RENAME をクリックしてください。

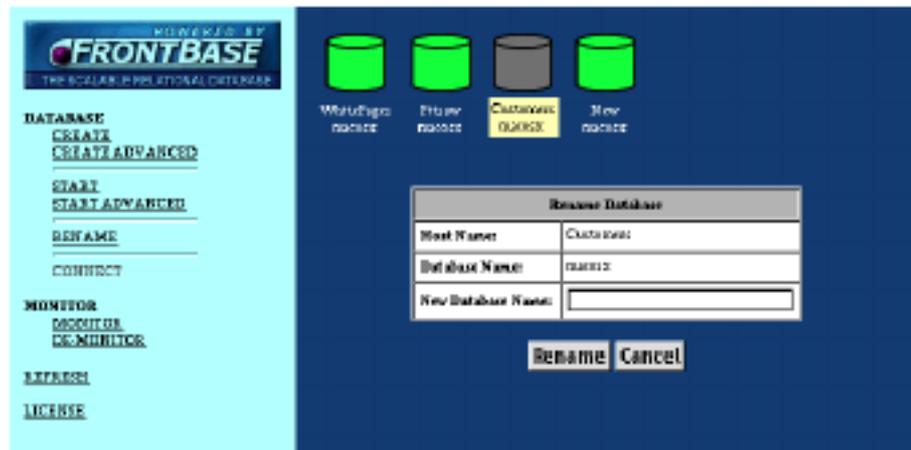


図8.10 データベースを改名する

新しいデータベース名を入力して、そしてリネーム (Rename) ボタンをクリックしてください。

モニタービューを新しくする

モニタービューを作ったときから、データベースが状態を変更したということが起きることがあります。モニタービューをアップデートするために、メニュー項目 REFRESH をクリックしてください。

データベース管理

データベース管理のモジュールを入力するために、モニタービューで走っているデータベースを選択して、そしてメニュー項目 DATABASE->CONNECT をクリックしてください。モニタービューは図8.11で見る接続の書式を表示するでしょう。

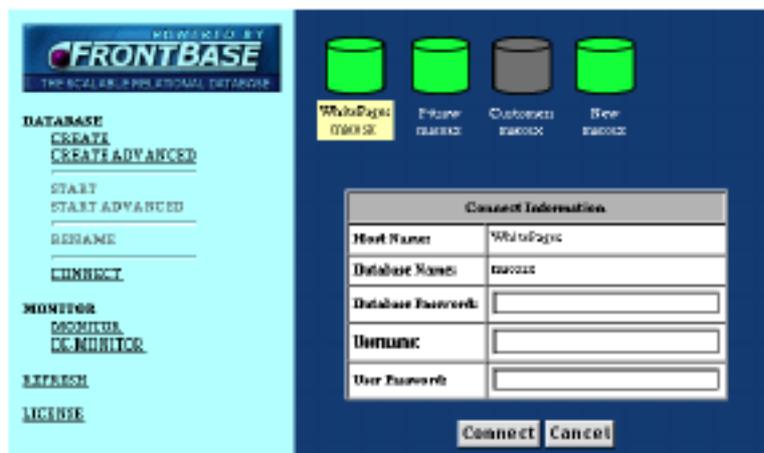


図 8.11 データベース管理モジュールを入力する

データベースパスワード（代わりにあなたはかつて接続していたこれを作ることができます）が、もしあれば、ユーザ名とパスワードを入力します、そして接続（Connect）ボタンをクリックします。もし接続を確立することができるなら、データベース管理ビューを入力します。

管理モジュールによって可能にしたアドミニストレーションタスクは次の通りです：

- 216 ページの「データベースをストップして、削除する」。
- 216 ページの「SQL 92」。
- 222 ページの「ユーザ管理」。
- 226 ページの「データベースパスワード」。
- 227 ページの「表のキャッシュの管理」。
- 228 ページの「ホワイト／ブラックリストの管理」。
- 230 ページの「管理の使い方」。
- 232 ページの「キーワード」。

データベースをストップして、削除する

データベースをストップする、あるいは削除するために、あなたは '_system' ユーザとして接続されているはずです。図 8 . 1 2 のように、メニュー項目 DATABASE->STOP あるいは DATABASE->DELETE をクリックしてください

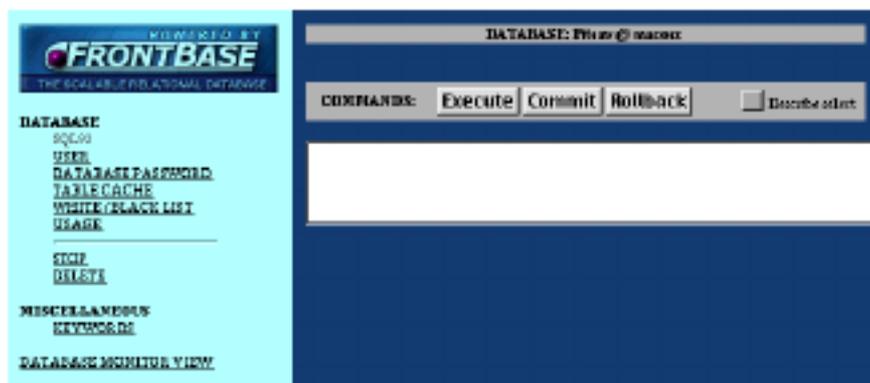


図 8 . 1 2 データベースをストップして、削除する

SQL 9 2

SQL92 の管理ビューは、サーバーを使って SQL ベースの相互作用「コマンドライン」をすることが容易です。SQL92 マネージャーを入力するために、メニュー項目 DATABASE->SQL92 をクリックしてください。最初に、セクションを入力しているコマンドだけが見えます。

後で、コマンドが同様にヒストリーのセクションを実行するとき、そして、図 8 . 1 3 のように、ログのセクションが現われるでしょう。

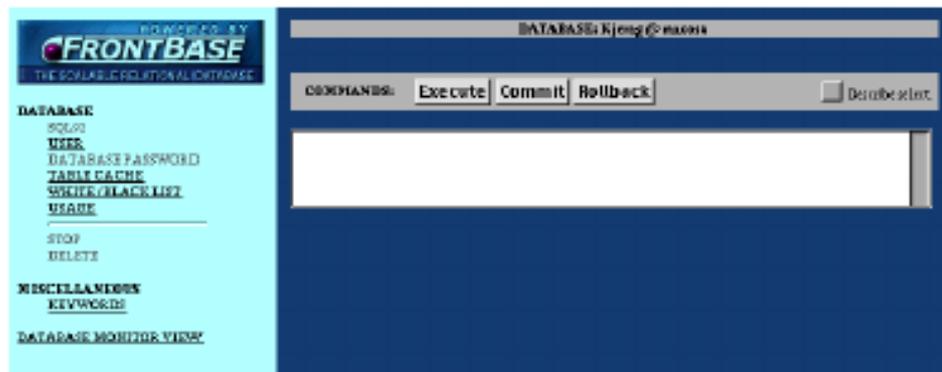


図 8 . 1 3 SQL92 の管理ビュー

セクションを入力しているコマンド

セクションを入力しているコマンドはコマンドと3つのボタンのヘッダーを入力するためにテキストフィールドから成ります：1つは、入力したコマンドを実行する、そして、2つは、確定 (COMMIT) とロールバック (ROLLBACK) コマンドのショートカットボタン。右にチェックボックス (「選択を記述する (Describe select)」) があります。イネーブルで、そして、選択が実行される時、リンク (「問合せの設計図 (Query Plan)」) がコマンドヘッダーで見えるでしょう。あなたがこのリンクをクリックするとき、新しいウィンドウが問合せの設計図を示してポップアップするでしょう。同様に、選択が実行される時、リンク (「結果の設定 (Result Set)」) がコマンドヘッダーで見えます。あなたがこのリンクをクリックする、ウィンドウが選択の結果を使ってポップアップします。

ヒストリーのセクション

ヒストリーのセクションはヘッダーと実行した文のリストから成ります。ヒストリーヘッダーであなたは表示したコマンドの数を変更することができます。同様にあなたはヒストリーリストをクリアすることができます。右にチェックボックス (「新しく加える (Add only new)」) があります。このチェックボックスがイネーブルであるとき、すでにリストにないコマンドだけがリストに加えられます。

ヒストリーリストのそれぞれの項目は、「右向きの三角形」と「下向きの三角形」という2つの小さいイメージを持っています。「右向きの三角形」のアイコンは、文が実行されます。「下向きの三角形」のアイコンをクリックすることはそれを実行の前に修正することができるコマンドフィールドに文をコピーします。



図 8 . 1 4 右向きと下向きの三角形

ログのセクション

ログのセクションはヘッダーとログビューから成ります。ログヘッダーであなたはログをクリアすることができます。図 8 . 1 5 を見てください。



図 8 . 1 5 右向きと下向きの三角形

事例

「選択を記述する (Describe select)」をコマンド 'select * from languages;' の実行がイネーブルした後、次のことが SQL92 管理ビューで示します。最初に SQL92 管理ビューです、それから結果の設定が来ます、そして図 8.16 のように、最後にクエリの実設計図まで来ます。



図 8.16 SQL の管理ビュー

Number of rows: 11

LANGUAGE	LANGUAGE_ID	LOCALIZED
English	1	English
French	2	Français
German	3	Deutsch
Danish	4	Dansk
Swedish	5	Svenska
Dutch	6	Nederlands
Norwegian	7	Norsk
Italian	8	Italiano
Spanish	9	Español
Finnish	10	Suomi
Portuguese	11	Português

Close

No SVI_LITERALS

GROUPS

Schema Name	Table Name	Row Count
ADMINISTRATOR	LANGUAGES	11

No SVI_JOINS

[Close](#)

ユーザ管理

入力するために、ユーザ管理はメニュー項目 DATABASE->USER をクリックします。ユーザ管理を入力するとき、すべての既存のユーザは図 8 . 1 7 で示されます。

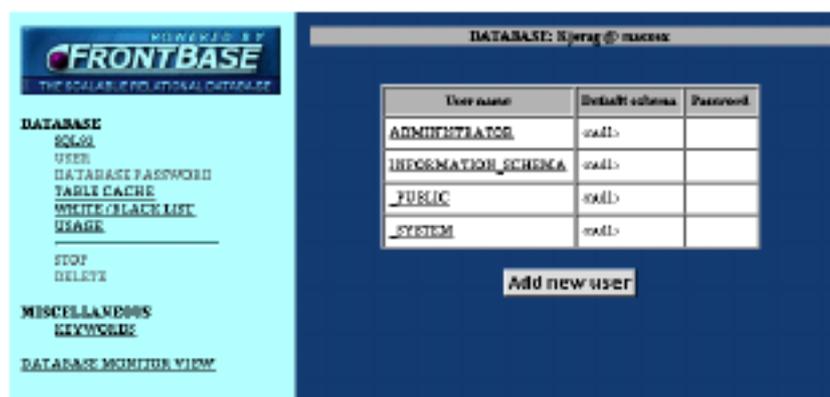


図 8 . 1 7 ユーザ管理のビュー

ユーザマネージメントは次のものを含んでいます：

- 2 2 2 ページの「新規ユーザを加える」。
- 2 2 3 ページの「ユーザを削除する」。
- 2 2 4 ページの「ユーザパスワードを設定する」。
- 2 2 5 ページの「スキーマをデフォルトに設定する」。

新規ユーザを加える

新規ユーザを加えるために、あなたはユーザ '_system' として接続されているはずです。新規ユーザを加える (Add new user) ボタンをクリックしてください、そうすればユーザビューは図 8 . 1 8 のように見えるでしょう。

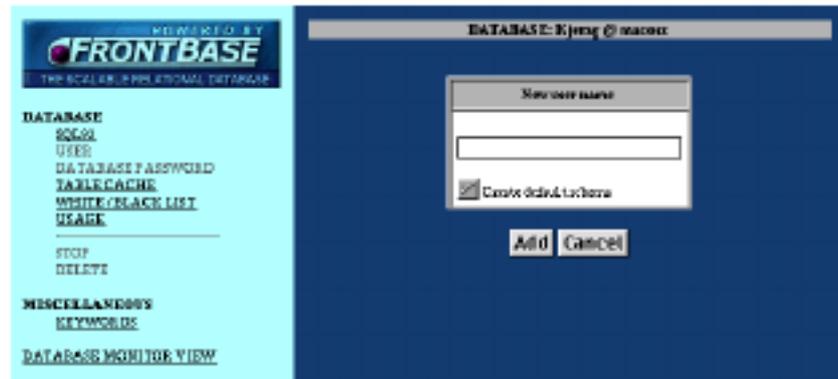


図 8 . 1 8 新規ユーザを加える

もし「スキーマをデフォルトに作成する (Create default schema)」スイッチを設定するなら、ユーザと同じ名前を使ってスキーマが作成されます。すなわち接続が新規ユーザ名を使って確立するように、新規ユーザがカレントユーザになることに注意してください。対応する SQL 文はログのウィンドウで示されます。

ユーザを削除する

ユーザを削除するために、あなたは最初に表のユーザを選択して、そして次に、制限をドロップする (Drop RESTRICT) あるいはドロップカスケードをドロップする (Drop CASCADE) ボタンをクリックします。制限してドロップすること (Dropping restricted) は、もしユーザがスキーマを所有しないなら、成功するでしょう。カスケードしてドロップすること (Dropping cascaded) は、ユーザによって無条件にスキーマの所有したすべてをドロップするでしょう。図 8 . 1 9 を見てください。

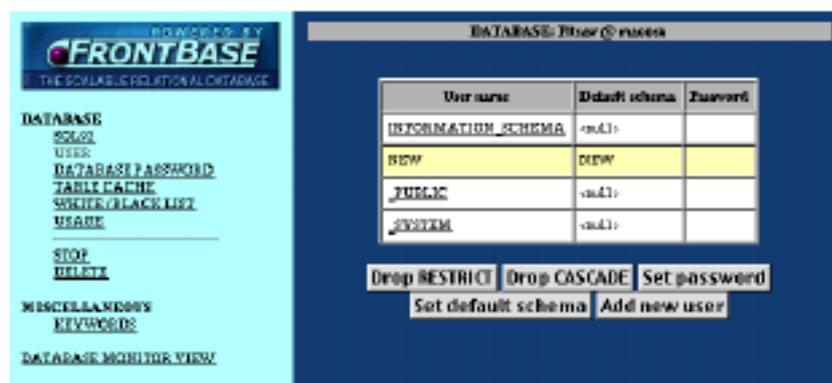


図 8 . 1 9 ユーザを削除する

ユーザパスワードを設定する

ユーザパスワードを設定するか、あるいは変更するために、表のユーザを選択して、設定したパスワード (Set password) ボタンをクリックしてください、そうすれば次のビューは図 8 . 2 0 で示されます。



図 8 . 2 0 ユーザパスワードを設定するか、あるいは変更する

新しいパスワードを入力して、パスワードを確かめます、すなわち、もう一度それをタイプして、そして設定したパスワード (Set password) ボタンをクリックしてください。表のユーザ項目はアップデートされます、そして、図 8 . 2 1 のように、パスワードがユーザのために設定することを示します。

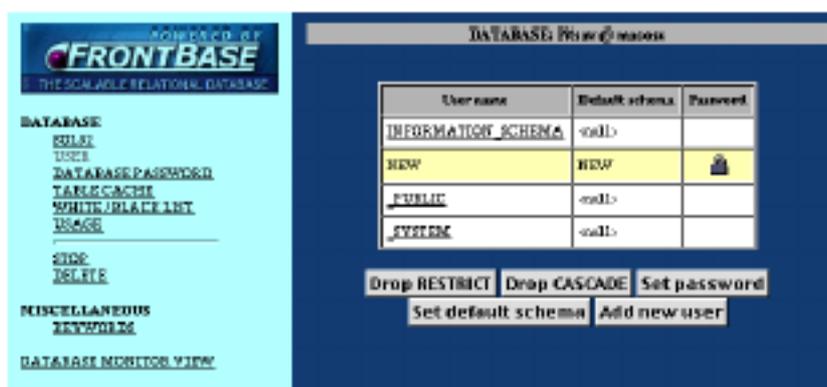


図 8 . 2 1 パスワードの設定を確認してください

スキーマをデフォルトに設定する

データベースに接続をするときはいつでも、デフォルトのスキーマはユーザのために現在のスキーマになるスキーマです。スキーマをデフォルトに設定するか、あるいは変更するために、スキーマをデフォルトに設定する (Set default schema) ボタンをクリックしてください、そうすれば可能なスキーマが図 8 . 2 2 でリストされます。

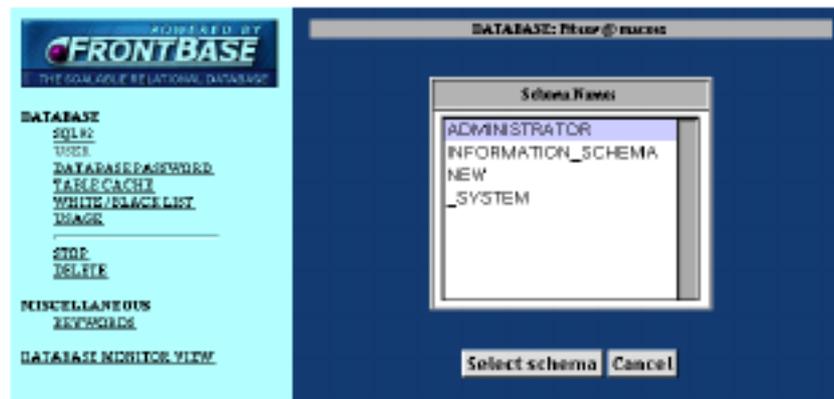


図 8 . 2 2 スキーマをデフォルトに設定する

スキーマを選択して、そしてスキーマを選択する (Select schema) ボタンをクリックしてください、ユーザ項目はアップデートされます、そして図 8 . 2 3 で見られるように、ユーザのために新しいデフォルトのスキーマを示します。

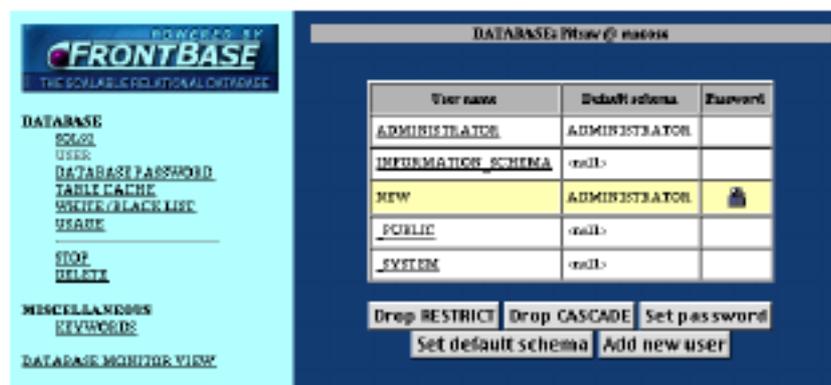


図 8 . 2 3 新しいデフォルトのスキーマ

データベースのパスワード

データベースのパスワードを設定するために、メニュー項目 DATABASE->DATABASE PASSWORD をクリックしてください。あなたはデータベースのパスワードを設定するために '_system' ユーザとして接続されているはずです。図 8 . 2 4 を見てください。

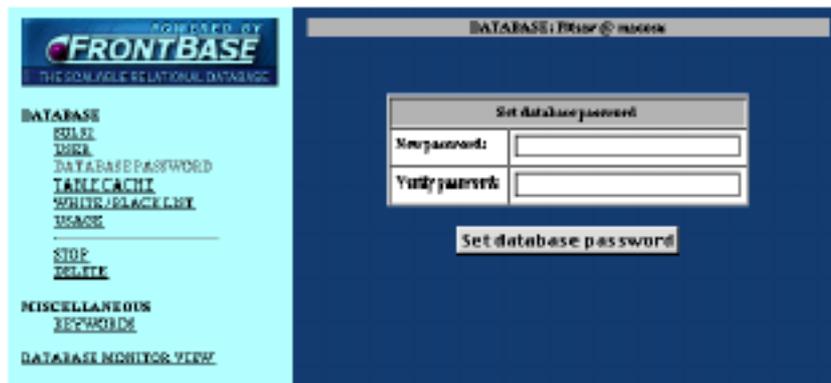


図 8 . 2 4 データベースのパスワードを設定する

新しいパスワードを入力して、パスワードを確かめてください、すなわち、もう一度それをタイプして、そしてデータベースのパスワードを設定する (Set Database Password) ボタンをクリックしてください。

表のキャッシュの管理

表のキャッシュマネージャーを入力するために、図 8.25 のように、メニュー項目データベース -> 表キャッシュをクリックしてください。

Schema	Table	Lower	Upper	%	Persistent	Prefetch
TELEPHON_BOG	ADDRESSNUMBER					
TELEPHON_BOG	AFTERSUFFIX		30000	100	YES	YES
TELEPHON_BOG	FIRMSNR		30000	100	YES	YES
TELEPHON_BOG	FORESUFFIX		30000	100	YES	YES
TELEPHON_BOG	EDMUNUMER		30000	100	YES	YES
TELEPHON_BOG	LANDSUFFIX		30000	100	YES	YES
TELEPHON_BOG	MIDDLEMAYNE		30000	100	YES	YES
TELEPHON_BOG	MAYNE					
TELEPHON_BOG	ORD					
TELEPHON_BOG	POSTDISTRIBTER		30000	100	YES	YES
TELEPHON_BOG	STILLINGER		30000	100	YES	YES
TELEPHON_BOG	VERE		30000	100	YES	YES

図 8.25 表のキャッシュマネージャー

接続したユーザによって所有するすべての表は表で示されます、そしてキャッシュのパラメータはそれぞれの表のために設定することができます。パラメータは次の通りです：

- Lower
キャッシュの中で保たれた表の列の最小数。
- Upper
キャッシュの中で保たれた表の列の最大数。
- %
キャッシュの中で保たれた表の列の合計数のパーセンテージ。
- Persistent
もし YES に設定するなら、キャッシュはトランザクションを保ちます、そうでないときはキャッシュはそれぞれの確定 (COMMIT) あるいはロールバック (ROLLBACK) の後に洗い流されます。

－ Preload

もし YES にセットするなら、サーバーをスタートするとき、表のすべての列はキャッシュにロードされるでしょう、そうでないときは、列はそれらに最初のリファレンスをロードします。

キャッシュパラメータを修正するために、スキーマ名をクリックすることによって、表の列を選択してください。図 8 . 2 6 のように、これはパラメータを編集してイネーブルするでしょう。

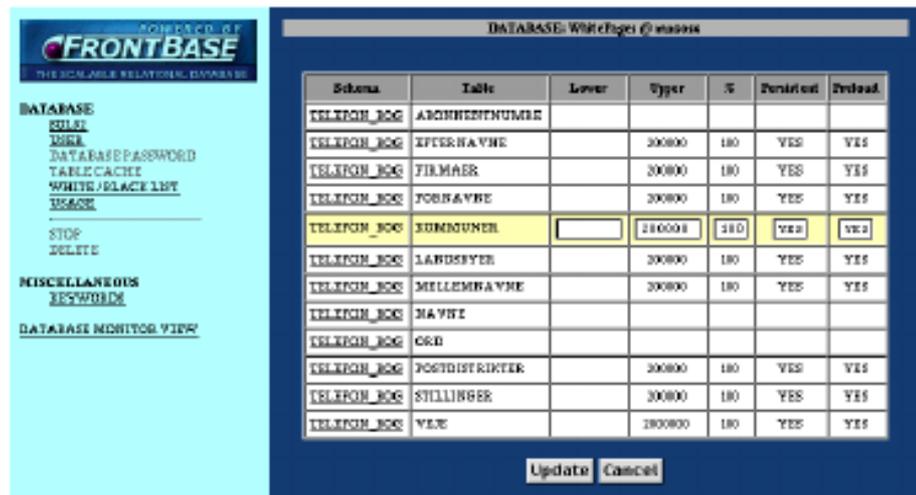


図 8 . 2 6 キャッシュパラメーターを変更する

新しいパラメータを入力した後で、アップデート (Update) ボタンをクリックしてください。

ホワイト／ブラックリストの管理

ホワイトリストとブラックリストのマネージャーを入力するために、図 8 . 2 7 のように、メニュー項目の DATABASE->WHITE BLACK LIST をクリックしてください。

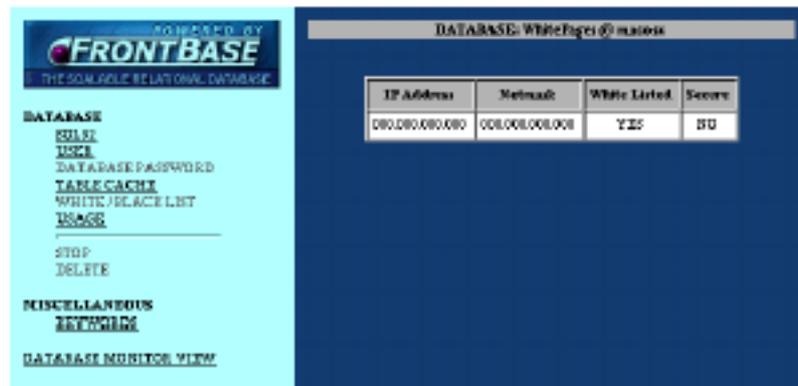


図 8.27 ホワイト/ブラックリストのマネージャー

クライアントが FrontBase サーバーに接続するとき、その IP アドレスは、もしクライアントが接続することを可能にするなら、ホワイトリストとブラックリストをチェックします。ホワイトリストとブラックリストのそれぞれの項目について、クライアントの IP アドレスは netmask を使って AND をとります、そして結果はホワイトリストとブラックリストの IP アドレスを比較します。もし一致するものを見つけて、そして、ホワイトの値が YES なら、クライアントは接続することを可能にします、そうでないときは、接続を拒否します。もし一致するものが見つけられないなら、接続は拒否します。もし YES が安全なフィールドに入力するなら、クライアントはサーバーを使ってそれ以上のコミュニケーションのために使用する暗号をリクエストします。

値をアップデートするために、あなたがユーザ '_system' として接続しなければなりません。リストの項目を修正するために、IP アドレスをクリックすることによって、表の列を選択してください。図 8.28 のように、これは値を編集してイネーブルするでしょう。



図 8 . 2 8 値をアップデートする

値は修正することができます、あるいは、ドロップ (Drop) ボタンをクリックすることによって、選択した項目を削除することができます。新しい項目が新規 (New) ボタンで作られます。

使用法の管理

使用法マネージャーはデータベースのすべての接続を監視することを可能にします。接続を見るために、図 8 . 2 9 のように、メニュー項目の DATABASE->USAGE をクリックしてください。

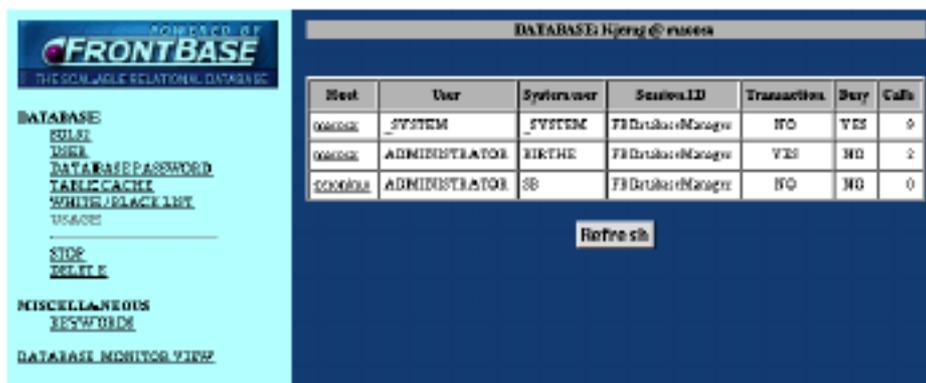


図 8 . 2 9 使用法マネージャー

使用法の表はクライアントが現在データベースとホストに接続したことを表示します、そしてそこからそれらは接続しました。トランザクションと使用中の行はより重要です。トランザクションは接続が

継続中のトランザクションを持っているかどうかを示します。使用中の行はクライアントが使用中であることを示します。

何らかの理由であなたがクライアントをストップする必要があるのは、ちょうどそうすることを望むから、あるいはそれが使用中である、例えば、あなたがユーザ '_system' としてログインしなければならない非常に複雑にした選択であるからです。あなたがストップすることを必要とするクライアントを、図 8.30 のように、そのホスト名をクリックすることによって、選択できます。



図 8.30 クライアントをストップする

もしクライアントが使用中であるなら、使用中の状態をストップする (Stop busy condition) をクリックして、そうでないときはエージェントをストップする (Stop agent) ボタンをクリックしてください。

キーワード

キーワードマネージャーはあなたにデータベースサーバーによって使われたキーワードをルックアップすることを可能にします。キーワードマネージャーをスタートするために、図8.31のように、メニュー項目 MISCELLANEOUS->KEYWORDS をクリックしてください。

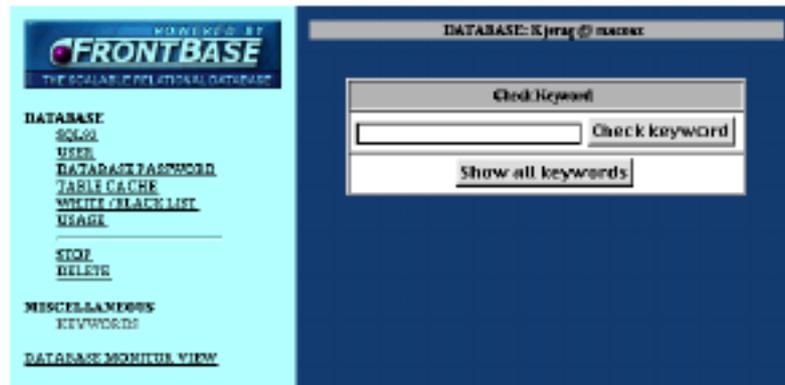


図 8 . 3 1 キーワードをルックアップする

テキストフィールドの単語を入力することによって、あなたは単語がキーワードであるかどうか調べることができます、そしてチェックキーワード (Check keyword) をクリックしてください。

データベースサーバーによって使われたすべてのキーワードを見るために、すべてのキーワードを示す (Show all keywords) ボタンをクリックしてください。図 8.32 のように、キーワードは個々のウィンドウで示されます。

Keywords		
ABSOLUTE	ACTION	ADA
ADD	ALL	ALLOCATE
ALTER	AND	ANY
ARE	AS	ASC
ASSERTION	AT	AUTHORIZATION
AVG	BEGIN	BETWEEN
BIT	BIT_LENGTH	BLOB
BIDIRECTIONAL	BOTH	BY
BYTE	C	CACHE
CALL	CASCADE	CASCADED
CASE	CAST	CATALOG
CATALOG_NAME	CHAR	CHAR_LENGTH
CHARACTER	CHARACTER_LENGTH	CHARACTER_SET_CATALOG
CHARACTER_SET_NAME	CHARACTER_SET_SCHEMA	CHECK
CLASS_ORIGIN	CLOSE	CLOSE
COALESCE	COBOL	COLLATE
COLLATION	COLLATION_CATALOG	COLLATION_NAME
COLLATION_SCHEMA	COLUMN	COLUMN_NAME
COMMAND_FUNCTION	COMMIT	COMMITTED
CONNECTION_NUMBER	CONNECT	CONNECTION
CONNECTION_NAME	CONSTRAINT	CONSTRAINT_CATALOG
CONSTRAINT_NAME	CONSTRAINT_SCHEMA	CONSTRAINTS
CONTINUE	CONVERT	CORRESPONDING
COUNT	CREATE	CROSS
CURRENT	CURRENT_CATALOG	CURRENT_DATE
CURRENT_SCHEMA	CURRENT_TIME	CURRENT_TIMESTAMP
CURRENT_USER	CURSOR	CURSOR_NAME
DATA	DATE	DATE_TIME_INTERVAL_PRECISION
DATE_TIME_INTERVAL_CODE	DAY	DEALLOCATE
DEC	DECIMAL	DECLARE
DEFAULT	DEFERRABLE	DEFERRED
DELETE	DESC	DESCRIBE
DESCRIPTOR	DIAGNOSTICS	DISCONNECT
DISTINCT	DOMAIN	DOUBLE
DROP	DYNAMIC_FUNCTION	ELSE
FIRST	END	END

図 8.32 すべてのキーワードを示す

ライセンスを加える

FBWebManager が走っているサーバーにライセンスを加えるために、図 8 . 3 3 のように、メニュー項目 LICENSE をクリックしてください。

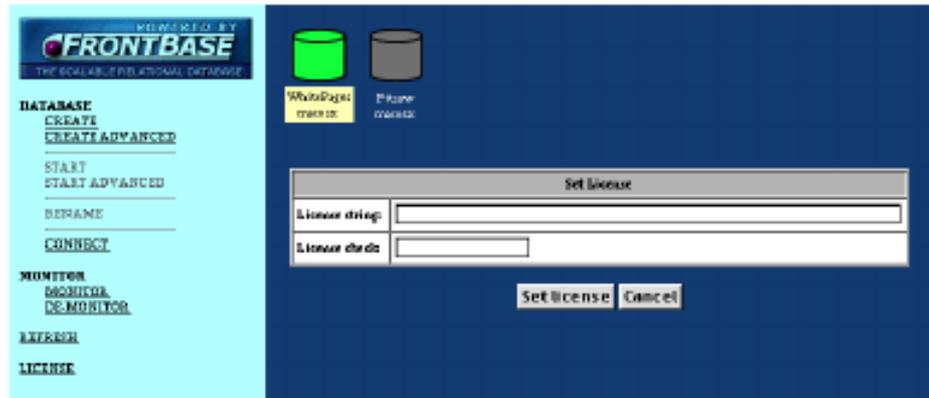


図 8 . 3 3 ライセンスを加える

あなたが FrontBase.com から受け取ったライセンス文字列とライセンスチェックキーを入力してください。そして次に、ライセンスを設定する（Set license）ボタンをクリックしてください。

sql92

この章は FrontBase のすべてのインストールで利用可能な sql92 のコマンドラインツールを紹介します。sql92 とは SQL 9 2 文を実行するためのコマンドラインツールです。

この章では次のことを含んでいます：

- － 2 3 5 ページの「コマンドシンタックス」。
- － 2 3 5 ページの「オプション」。
- － 2 3 7 ページの「一般」。
- － 2 3 7 ページの「sql92 によって解釈するコマンド」。

最も普通のコマンドだけがこのセクションに含まれることに注意してください。FrontBase の SQL の完全なリストについては、今度の SQL リファレンス文書を見てください。

コマンドシンタックス

sql92 <options> [<filename>]

もしファイル名を指定するなら、sql92 は名前付のファイルに記憶した SQL 9 2 文を読み込んで、そして実行します。もし入力ファイルを除くなら、SQL 9 2 文は標準の入力から読み込みます。すべての出力が標準の出力に書き込みます。

オプション

- ・ -a
自動的にトランザクションを確定しないでください。
- ・ -c
比較します。テストを実行して、入力と出力を比較して、もしそれらが等しいなら、パスするコメントを書いてください、そして、そうでない時、失敗します。
- ・ -e
エラーで終了します。

- -i
無視します。ignore オプションを指定しないなら、エラーに遭遇するとき、通常 sql92 は終了するでしょう。
- -l <number>
結果の設定からフェッチされる列の大部分の番号。
- -m
それが sql を実行したとき、サーバーから返されたように、メタデータを印刷してください。
- -n
結果を調べなくてはならない演算子を示す注釈を印刷してください。
- -p
プロンプト。入力ソースが端末装置であるとき、sql92 が文のためにデフォルトにプロンプトを出すでしょう。プロンプトオプションを指定するとき、sql92 は常にプロンプトを出すでしょう。
- -q
クォートの中のコントロール文字を可能にしてください。
- -s
静かです。それらが実行されるとき、コマンドをプリントしないでください。入力ソースが端末装置ではないとき、文がデフォルトにプリントされます。
- -t
タイミング情報をプリントしてください。
- -u <encoding>
入力エンコーディング。<encoding> 引数は入力ファイルをコード化することを指定します。デフォルトは同様に us-ascii を受け入れる UTF8 です。
- -v
冗長です。それらが実行されるとき、文をプリントしてください。
- -w
警告を印刷してください。

一般

sql92 コマンドは常に ';' 文字で終了します。あなたが sql92 に入力をタイプするとき、 ';' 文字が入力されるまで、それは文を解釈しないでしよう。これは SQL 9 2 文がいくつかの行に及ぶことを可能にします。たいていの SQL 9 2 の文は FrontBase SQL サーバーによって解釈されます、しかし多くのコマンドが sql92 によって解釈されます。これらのコマンドは一般に入力ファイルのサーバーとアドミニストレーションに接続します。

コメントが '#' 文字あるいは '-' 文字列でスタートして、そして行の終わりまでに終了します。コメント記号としての '#' はあなたにインタプリタを書くことを可能にします。コメントは入力の一部であるとみなされ、そしてそれで冗長な出力の一部です。文字 '>' で始まっている行は同様にコメントと見なされます、しかし冗長な出力の一部と見なされません。SQL 9 2 文を実行する結果は、 '>' で始まる行を使って、書かれます。これはあなたに、復帰のテストを単純化して、スクリプトの出力が入力と同じであるテストスクリプトを書くことを可能にします。

sql92 が呼び出されるとき、それは、名前 .sql92rc.sql を使って、初期設定ファイルを実行しようと試みます。sql92 は最新の作業ディレクトリとユーザのホームディレクトリを検索します。

sql92 によって解釈したコマンド

接続

シンタックス :

```
CONNECT TO <database-name>  
[DATABASE_PASSWORD <database-password>]  
[ON <host-name>]  
[AS <connection-name>]  
[USER <user-name>]  
[PASSWORD <password>];
```

```
CONNECT TO DEFAULT;
```

最初の形式は <hostname> という名前のホストにオプションのデータベースパスワード <database-password> を使って <database-name> という名前のデータベースへの接続を確立します。もし <hostname> が指定されないなら、ローカルホストを装います。接続は <connection-name> で名前をつけます、そしてそれは同様にプロンプト値として使います。もし接続名が指定されないなら、接続はデータベース名の名前とホストの名前に名前をつけます。<user-name> は確立した SQL 92セッションのために使った識別子の許可を指定します。もし <user-name> が指定されないなら、ホストのオペレーティング・システムのログイン名を使います。<password> はユーザパスワードを指定します。接続コマンドは SQL 92の文です、しかし sql92 によって解釈されます。

2番目の書式はデフォルト接続を確立します。

データベースを作成する

シンタックス :

```
CREATE DATABASE <database> [(ON | @ | HOST) <hostname>]
[PASSWORD <password>]
[OPTIONS <option>...];
```

<hostname> 名で、ホストに <database-name> 名を使って新しいデータベースを作ってください。もしホスト名が指定されないなら、ローカルホストの名前を使います。もしターゲットホストの FBExec がパスワードを要求するなら、それは指定することができます、もしそれがそうでなく、sql92 が相互作用的に走っているなら、ユーザはパスワードの入力を促されるでしょう、そうでないときは、文は失敗するでしょう。データベースを作るとき、オプションが使われるでしょう。

Blob を定義して、そして Clob を定義する

シンタックス :

```
DEFINE BLOB <blob-name> LENGTH <length> VALUE {<hex-bytes>};
DEFINE CLOB <clob-name> LENGTH <length> VALUE {<hex-bytes>};
```

<blob-name> (あるいは <clob-name>) 名、バイトの長さ <length>、そして <hex-bytes> のリストの値を使って blob (あるいは clob) オブジェクトを定義します。<hex-byte> は2つの16進数です。<hex-bytes> のリストは空白スペース (改行、タブ、スペースなど) を含むことができます。blob (あるいは clob) オブジェクトは定義して、そして次の SQL 92 の文で参照することができます。リファレンスが書式 '@<blob-name>' (あるいは '@<clob-name>') を持って、そしてその特定の行のために blob (あるいは clob) オブジェクトを作るでしょう。次の確定、あるいはロールバックが実行されるとき、blob (あるいは clob) オブジェクトは見えなくなるでしょう。新しい BLOB が、(ほとんど) FrontBase データベースの下のレイヤーであるファイルシステムのファイルとして、記憶されます、もしこのようなファイルが (挿入 (INSERT) の間に起きる) 列の行にリンクされないなら、確定 (COMMIT) あるいはロールバック (ROLLBACK) を実行するとき、ファイルは当然削除してください。

データベースを削除する

シンタックス :

```
DELETE DATABASE <database> [(ON | @ | HOST) <hostname>]
[PASSWORD <password>];
```

ホスト <hostname> のデータベース <database> は削除されます。もしホスト名が指定されないなら、ローカルホストを装います。もし th ホストの FBExec を要求するなら、パスワードは、<password> として指定することができます、もしパスワードが指定されないなら、sql92 が相互作用的に走っています、そして FBExec はユーザがプロンプトを出してパスワードを要求します。

切断

シンタックス :

```
DISCONNECT <connection-name>;
DISCONNECT CURRENT;
DISCONNECT ALL;
```

最初の形式は <connection-name> 名を使って接続を切断します。2番目の形式は現在の接続を切断します、そして3番目の形式はすべての接続を切断します。

スクリプト

シンタックス :

```
SCRIPT <path-to-file>/schema.sql;
```

指定ファイル (schema.sql) から sql92 コマンドを読み込んで、実行します

出て終わる

シンタックス :

```
EXIT  
QUIT
```

sql92 から出ます。EXIT と QUIT コマンドは ';' を使って終了する必要があります。唯一のものです。

接続を設定する

シンタックス :

```
SET CONNECTION <connection-name>;  
SET CONNECTION DEFAULT;
```

最初の形式は <connection-name> 名を使って接続を現在の接続にします。2 番目の形式はデフォルト接続を現在の接続にします。

データベースのパスワードを設定する

シンタックス :

```
SET DATABASE_PASSWORD [<password>;
```

データベースのパスワードを <password> にセットします。もし <password> が省略されるなら、将来の接続はパスワードを必要としないでしょう。

パスワードを設定する

シンタックス :

```
SET PASSWORD  
[USER <user-name>]  
[OLD <old-password>]  
[NEW <new-password>];
```

<user-name> 名を使ってユーザのためにパスワードを設定します。もしユーザがすでにパスワードを持っているなら、それを指定してください、そして新しいパスワード <new-password> はユーザのために確立されます。もし <new-password> が省略されるなら、ユーザはパスワードなしでログインできます。

データベースをスタートする

シンタックス :

```
START DATABASE <database> [(ON | @ | HOST) <hostname>]  
[PASSWORD <password>]  
[OPTIONS <option>...];
```

<host-name> 名で、ホストに <database-name> 名を使ってデータベースをスタートします。もしホスト名が指定されないなら、ローカルホストの名前が使われます。もしターゲットホストの FBExec がパスワードを要求するなら、それを指定することができます、もしそれがそうでなく、sql92 が相互作用的に走っているなら、ユーザはパスワードのためにプロンプトを出すでしょう、そうでないときは、文は失敗するでしょう。データベースをスタートするとき、オプションが使われるでしょう。

データベースをストップする

シンタックス :

```
STOP DATABASE;  
STOP DATABASE <database-name> [@|ON|HOST <host-name>];
```

最初の形式は現在接続されたデータベースをストップします。2 番目の形式は指定したホストのデータベースをストップします。データベースを止めるために、あなたはユーザ `_SYSTEM` として接続

してください。コマンドはシステムの接続を作って、そしてもし必要とされるならデータベースパスワードと `_SYSTEM` パスワードのためにプロンプトを出そうと試みるでしょう。

自動にスタートする (Autostart)

シンタックス :

```
AUTOSTART [ON|HOST <host-name>];
```

ユーザがコントロールすることができる `autostart` コマンドで、それがブートされる時、データベースが `<host-name>` を使って所定のホストからスタートします。もしホスト名が省略されるなら、ローカルホストを装います。`autostart` コマンドは次のサブコマンドを持っています :

```
SHOW;
```

自動にスタートしたデータベースのリストを示します。リストは番号が付けられています、そして番号は特定のデータベースのリファレンスに使うことができます。

```
ADD <database-name> [ <options>...];
```

自動にスタートしたデータベースの設定にデータベースを加えます。サーバーは指定したオプションを使ってスタートします。

```
OPTIONS <number> <options>...;
```

選択したデータベースのためにオプションを設定します。

```
DELETE <number>;
```

自動にスタートしたデータベースの設定から指定したデータベースから取り除きます。

```
SAVE [PASSWORD <password>];
```

変更を確定します。もしホストの FBExec がパスワードを要求する
なら、それを指定してください、もしそうではなくて、sql92 が相互
作用的に走っているなら、ユーザがプロンプトを出します、そうでな
いときはコマンドは失敗するでしょう。

EXIT;
QUIT;

autostart コマンドループを終了してください。

接続を示す

シンタックス :

SHOW CONNECTIONS;

確立したすべての接続を示します。

使い方を示す

シンタックス :

SHOW USAGE;

現在接続されたデータベースのためにセッションのまとめをプリン
トします。

データベースのログを示す

シンタックス :

SHOW DATABASE LOG <database-name> [@ | ON | HOST <host-name>];

指定したホストのデータベースのためにデータベースログファイル
の後部を示します。もしホスト名が指定されないなら、ローカルホス
トを装います。

データベースを示す

シンタックス :

```
SHOW DATABASE [FULL];
```

現在接続されたデータベースの情報を示します。もし FULL が指定されるなら、すべての利用可能な情報が示されます。

ライセンスを示す

シンタックス :

```
SHOW LICENSE;
```

ライセンスの詳細を示します。

ログを示す

シンタックス :

```
SHOW LOGS [ALL|<number>];
```

現在接続されたデータベースのトランザクションログファイルのために状態を表示します。もしすべてが指定されるなら、すべてのトランザクションログが示されます、もし番号が指定されるなら、最新の番号がプリントされます。デフォルトは1です。

メモリーを示す

シンタックス :

```
SHOW MEMORY [ALL];
```

現在接続されたデータベースサーバーによって割り当てられたメモリの状態を示します。アドミニストレーションの章の中のメモリー使用のセクションを参照してください。

キャッシュを示す

シンタックス :

```
SHOW CACHES;  
SHOW CACHES <table-name>;  
SHOW CACHES <schema-name>.<table-name>;
```

キャッシュのために統計を示します。最初の形式は現在のスキーマのキャッシュを示します、2番目の形式は指定したテーブルのキャッシュを示します、そして最後の1つは指定したスキーマのテーブルを示します。スキーマ名と表の名前は SQL 92 のワイルドカード文字 % と _ を含むことができます。

トランザクションを示す

シンタックス :

```
SHOW TRANSACTION;
```

現在のトランザクションパラメータを示します。

スキーマを示す

シンタックス :

```
SHOW SCHEMA <schema name>;
```

表とビューが指定したスキーマで定義したのを示します。

表を示す

シンタックス :

```
SHOW TABLE <table name> [FULL];
```

表の定義を印刷します。もし FULL が指定されるなら、すべての利用可能な情報が印刷されます。

ビューを示す

シンタックス :

```
SHOW VIEW <view name> [FULL];
```

ビューの定義を印刷します。もし FULL が指定されるなら、すべての利用可能な情報が印刷されます。

サイズを示す

シンタックス :

```
SHOW SIZE;  
SHOW SIZE <table-name>;  
SHOW SIZE <schema-name>.<table-name>;
```

表が指定した実装するために使われたディスクスペースの量を示します。最初の形式は、現在のスキーマで、すべての表を示します、2番目の形式は、現在のスキーマで、指定した表を示します、そして、最後の形式は、指定したスキーマで指定した表を示します。スキーマ名と表の名前は SQL92 のワイルドカード文字 % と _ を含むことができます。

ヒストリーを示す

シンタックス :

```
SHOW HISTORY;
```

コマンドヒストリーを示します。

複製をスタートする

シンタックス :

```
START REPLICATOR;
```

複製をスタートします。もし複製がすでに走っているなら、エラーが報告されます。

クライアントを作成する

```
CREATE CLIENT <database-name>  
[@|ON|HOST <host-name>]  
[DATABASE PASSWORD <password>]  
[PASSWORD <system-password>];
```

ホストに名前を使って新しいクライアントを作ります。もしホスト名が省略されるなら、ローカルホストが使われます。それが後でクライアントリストに加えられることができるように、create コマンドは新しいクライアントを作って、そして初期化します。

クライアントを加える

```
ADD CLIENT <database-name>  
[@|ON|HOST <host-name>]  
[DATABASE PASSWORDS <password>]  
[PASSWORD <password>];
```

ホストに複製クライアントリストに名前を使ってクライアントを加えます。複製はクライアントを最新の状態に保つでしょう。パスワードはクライアントに接続することが可能である複製によって必要とされる `_SYSTEM` ユーザのためのデータベースパスワードとパスワードです。

クライアントを取り除く

```
REMOVE CLIENT <database-name> [@|ON|HOST <host-name>];
```

クライアントをリストから取り除くと、クライアントはもうアップデートされません、しかし後で再びクライアントリストに加えることができます。

クライアントを示す

```
SHOW CLIENTS;
```

クライアントリストとそれぞれのクライアントの状態を示します。

複製をストップする

```
STOP REPLICATOR;
```

複製をストップします。

開発者のための FrontBase

この章では、それらのデータソースのために FrontBase を統合するアプリケーションの開発者のために重要である問題を扱います。この章は次のセクションに分けられます：

- 250 ページの「利用可能な起動オプション」。
- 256 ページの「FrontBase とセキュリティ」。
- 257 ページの「暗号化」。
- 259 ページの「ツールとオプション」。
- 263 ページの「データタイプ」。
- 272 ページの「FrontBase データタイプのファンデーション / Java オブジェクトをマップすること」。
- 276 ページの「主キーと自動生成」。
- 277 ページの「列レベルの特権」。
- 280 ページの「あなたがコレーシヨンのためにできること」。
- 284 ページの「あなた自身のアプリケーション、あるいはソリューションに FrontBase を埋め込む」。

利用可能な起動オプション

FrontBase は、データベースをスタートするとき、指定することができるいわゆるオプションの大規模なリストをサポートします。オプションは、コマンドライン、あるいは、例えば FBWebManager を使うときに与えることができます。

- autocommit - トランザクションの自動的な確定
自動にそれぞれの文の後にすべてのトランザクションを確定します
- autcreate - 自動的に新規ユーザを作成する
ユーザを接続するとき、自動にユーザ名を作ります
- create - 無条件に新しいデータベースを作る
- fbexec - FBExec の有無にかかわらず走る
FBExec プロセスの有無にかかわらず走ります
- index - 索引モードをデフォルトに設定する
新しい索引のためにデフォルトモードを指定します
- localonly - ローカルな接続だけを受け入れる
- logSQL - ファイルの SQL 文をログに書く
それらを受け取るとき、すべての SQL 文をログファイルに書きます
- port - ポート番号
使うポート番号を指定します
- prvchk - 特権のチェック
特権のチェックをディスエーブルにします
- rlpriv - 列レベルの特権
- restore、rollforward、wtd - バックアップから復帰する
- RDD - 未使用のデバイスドライバ
未使用のデバイスドライバをイネーブルにします
- keys、scomm、sdisk - 暗号
暗号のために使うキーを指定します
- rmaster、rclient - 複製
複製の原版
- rcluster - クラスタ
クラスタの一部になります

autocommit - トランザクションの自動的な確定

Syntax: -autocommit[=`no`|`yes`] `yes` is the default

FrontBase 2.0 は自動的に確定する機能を提供します、それによってトランザクションが、正常に実行したトランザクションを始めた SQL 文の後に、自動的に確定します。

次の SQL 文を実行することによって、自動的に確定する機能は同様に ON・OFF にすることができます：

```
SET COMMIT FALSE; -- Turn the auto commit feature off
SET COMMIT TRUE;  -- Turn the auto commit feature on
```

もしオプションが省略されるなら、FrontBase 2.0 データベースをスタートするとき、自動的に確定する機能を OFF にします。

autocreate – 自動的に新規ユーザを作成する

Syntax: -autocreate[=no|yes] yes is the default

-autocreate=yes を指定することによって、データベースに接続しているすべての未知のユーザが自動的に作られるでしょう。この機能をオンにすることによって、あなたがデータベースのセキュリティに違反していることに注意してください。

もし、FrontBase 2.0 データベースをスタートするとき、オプションが省略されるなら、未知のユーザが接続することは許されません。

create – 無条件に新しいデータベースを作成する

Syntax: -create

-create を指定することによって、(もしそれが存在したなら)、新しいデータベースが無条件に同一名を使って既存のデータベースに上書きして作ります。

警告！ 注意してこのオプションを使ってください。

fbexec – FBExec の有無にかかわらず走る

Syntax: -fbexec[=no|yes] yes is the default

-fbexec=no を指定することによって、スタートしている FrontBase のデータベースは FBExec プロセスを使って通信を確立しようとしていないでしょう。これは FBExec が専門的な理由のために必要としない特別な環境で走るために FrontBase をイネーブルにします。このオプションが -port オプションを使うことを要求し、そして特別製のクライアントサイドのソフトウェアを要求することに注意してください。

通常 FrontBase データベースは FBExec と通信するでしょう、そしてそれは FrontBase がインストールしたホストに情報のゲートウェイとして機能します。

index – 索引モードをデフォルトに設定する

Syntax: `-index[=time|space] space is the default`

FrontBase のデータベースで作ったすべての索引は持続スペース (PRESERVE SPACE) あるいは持続時間 (PRESERVE TIME) モードの設定を使って作ります。持続スペース (PRESERVE SPACE) は、最高 1 . 0 0 0 . 0 0 0 の列を使って表のためによく動作するメモリの非常に効率的なモードです。持続時間 (PRESERVE TIME) はもっと多くのメモリを使います、しかし、素晴らしい性能のために何百万という列の表に遭遇します。

注: デフォルトは持続スペース (PRESERVE SPACE) です。

次の SQL 文を実行することによって、所定の表に作られた索引のモードは変更することができます:

```
ALTER TABLE <table> SET INDEX PRESERVE SPACE;
```

あるいは

```
ALTER TABLE <table> SET INDEX PRESERVE TIME;
```

localonly – ローカルな接続だけを受け入れる

Syntax: `-localonly`

このオプションを指定することによって、Frontbase のデータベースはデータベースと同じコンピュータで走っているクライアントから接続を受け入れるだけでしょう。

デフォルトは、ローカルクライアントと同様、ネットワークで結ばれた接続を受け入れます。

logSQL - ファイルの SQL 文をログに書く

Syntax: `-logSQL`

このオプションを指定することによって、Frontbase のデータベースがファイルの受け取った SQL 文をすべてログファイルに書くでしょう。ファイルは FrontBase のインストールの Databases ディレクトリで作られて、そして名前をつけます：

Databases/<database name>.fbsql Windows NT

Databases/<database name>.fb.sql All other platforms

デフォルトは SQL 文をログファイルに書きません。

port - ポート番号

Syntax: `-port=<port number>`

FrontBase データベースが BSD スタイルソケットを使っているクライアントと通信します、接続をする前にいわゆるポート番号を知るようにクライアントに要求します。クライアントが所定の FrontBase のデータベースのポート番号を得る FBExec プロセスと通信するでしょう。

FrontBase のデータベースは通常、ホストオペレーティング・システムから、ポート番号を得るでしょう、しかし、例えば、スタティックなファイアウォールを使ってセットアップする、そして知っているポート番号を使ってください。`-port` オプションを使うことによって、FrontBase のデータベースは特定のポート番号を使うように指示することができます。

prvchk - 特権のチェック

Syntax: `-prvchk[=no|yes]` Default is yes

FrontBase 2.0 で特権のチェックの番号が、特に、参照した行で適切な SELECT 特権のチェックを強制しました。これは既存のアプリケーションを「壊す」ことができます、それで、`-prvchk=no` オプションを指定することによって、特権のチェックを起こさないでしょう、すなわち、既存のアプリケーションが FrontBase 1.2 と同じように動作します。

rlpriv – 列レベルの特権

Syntax: `-rlpriv`

FrontBase は列レベルの特権と呼ばれる、唯一の特徴を提供します、それはあなたに、データベースの表のそれぞれの個々の列に Unix ファイルシステムのファイルのような特権を割り当てることを可能にします。新しいデータベースを作るとき、`-rlpriv` オプションは与えられることが必要です。

もっと多くの情報について、279 ページの「列のためにアクセス権を選択する」を参照してください。

restore、rollforward、wtd – バックアップから復帰します

Syntax: `-restore[=<file name>]`
`-rollforward`
`-wtd`

すべての3つのオプションは FrontBase によって提供したバックアップメカニズムと関係があります。

RDD – 未使用のデバイスドライバー

Syntax: `-RDD[=<size of cache>]`

FrontBase が非常に進歩した書くことを通してキャッシュメカニズムを提供します、それは同様に、データの記憶装置、したがって名前として未使用のデバイス (パーティション) の使用をサポートします。

キャッシュのサイズは、オプションが所定のデータベースのために指定するか、あるいは、サイズが減少するか、あるいは増えることを必要とする時、初めて与えられることを必要とします。

keys 、 scomm 、 sdisk – 暗号

Syntax: -keys=<file name>
-scomm
-sdisk

唯一の暗号スキームを展開することによって、FrontBase は高度なデータの保護を提供します。単にクライアントと FrontBase データベースの間の通信が暗号化する（ストリーミング）ことができるだけではありません、けれども同様にハードディスクの実際の datastore は暗号化する（ブロックモード）ことができます。

rmaster 、 rclient – 複製

Syntax: -rmaster
-rclient

FrontBase によって提供した複製スキームで、あなたは、すべてのアップデートが行われなくてはならない1つの原版のデータベースを持つことができます。読み込み専用の複製されたクライアントデータベースの数を加えることができます。

rcluster – クラスタ

Syntax: -rcluster=<cluster members>

FrontBase は同様にクラスタ機能を提供します、それによって、クラスタの N データベースが、それらがお互いに変更をすべて同期させる間に、すべてアップデートすることができます。

FrontBase とセキュリティ

FrontBase で、あなたは、盗聴と干渉に対して通信を保護することに加えて、データを保護するために、いくつかのオプションを持っています。保護は、通信の暗号とデータの記憶装置、許可のパスワード、そしてコンピュータのブラックリスト/ホワイトリストを応用することによって、得ることができます。実際に特別な組み込みの認証機能を使って FrontBase サーバーの使用法を保護することは同様に可能です。

パスワードの保護

パスワードの保護の2つのレイヤ、データベースパスワードとユーザパスワードを提供します、。

データベースのパスワード

もしデータベースのパスワードが設定されるなら、クライアントが接続プロトコルの一部としてデータベースのパスワードをサーバーに送ってください、もしサーバーがパスワードを確認することができないなら、クライアントの接続はすぐに終了します。

ユーザのパスワード

それぞれのデータベースのユーザはパスワードを持つことができます、セッションがそのユーザのために作られるとき、パスワードはサーバーによって確認されます、もし確認が失敗する場合は、セッションは作られません。セッションが正常に作られたとき、SQL92の定義した保護を引き継ぎます。

サーバーの認証

FrontBase は、安全な環境で動作しないことは、ほとんどの開発者を苛立たせる事であるから、デフォルトは、イネーブルではないが、認証の機能を持っています。

イネーブルにするために、この特徴は、/Library/FrontBase への書き込み許可を使ってルートあるいはユーザとして走ってください：

- ・ /Library/FrontBase/bin/FBExec -newpasswd=secret (あなたは FBExec -h によって使い方を見ることができます)。

- ・ 走っている FBExec のサービスを再起動してください。

この時点で、データベースを作る、それらをスタートする、それらをストップするなどのことは認証を必要とするでしょう。個々のデータベースはデータベースのパスワードとユーザのパスワードの使用を通してさらに安全に保つことができます。

一般的なパスワードの処理

パスワードはどんな長さでもできます、そして、パスワードは、クライアントソフトウェアの外に、決して露出しません、そして、それらはデータベースにさえいません。アプリケーションがアプリケーションから FrontBase のクライアントソフトウェアまでパスワードを渡すとすぐに、一方向の関数がパスワードの要約を作るために用いられます。関数はパスワードの一部を廃棄するでしょう、それでその要約からパスワードを推論することはできません。ユーザ名は要約の一部です、それで同じパスワードを使った2つのユーザは同じ要約を持たないでしょう。パスワードの要約はパスワードの代わりに確認のために使います。

暗号化

暗号は通信チャンネルとデータの記憶装置を保護するために使います。あなたが FrontBase を作成するとき、あなたはオプションとしてディスクに記憶したデータが暗号化していることを指定して、そしてオプションとしてサーバーとそのクライアントの間の通信チャンネルが安全であることを指定することができます。あなたは指定したそれぞれのオプションに暗号化のキーを提供してください。

データの暗号化

ディスクで記憶したデータは、512バイトのブロックで暗号ブロック連鎖モードのトリプル DES を使って暗号化します、データの記憶装置は512バイト/ブロックでブロック指向です。初期化ベクターはシステムの中のブロックの論理的な位置に依存しません、従って同じ内容を使ったブロックは決して異なった暗号文のブロックを作らないでしょう。

データの暗号化のために使ったキーは DES 暗号規格で64ビットの初期化ベクターと 3x56 ビットです。

安全なチャネル

クライアントとサーバーは安全なチャネルを確立することが可能です。クライアントがサーバーに接続するとき、それは、サーバーから、パブリックな RSA のキーを受け取ります、クライアントは出て行くデータと入ってくるデータのために、ランダムなセッションキーの設定を作ります、パブリックな RSA のキーを使ってそれらを暗号化します、そして結果をサーバーに送ります。サーバーは、そのプライベートキーを使ってクライアントから、結果を解読します、そして従って、クライアントとサーバーは秘密キーの普通の設定を確立しました。

通信データの暗号化のために使ったアルゴリズムは暗号文と明確なテキストのフィードバックを使ったバイトストリームモードのトリプル DES です。明確なテキストのフィードバックはエラーの次のすべてのバイトにエラーを伝えてしまいます、そしてそれはエラーの検出を単純にして、小さい量の冗長性を広めて、そして受け取る最後に確認のためにそれを使います。

ツールとオプション

数少ないツールがキーの管理をサポートするために提供します、そして FrontBase はセキュリティと関係があるオプションの番号を持っています。

FBKeyGenerator

キーの生成は FrontBase によって使うことができるキーの設定を作るために使います：

FBKeyGenerator <RSA-key-bit-size> [<key-filename>]

キービットサイズは 1024 と 4096 ビットの間にしてください。もしキーのファイル名が指定されるなら、キーは、そのファイルに、書き込まれます、そうでないときは、キーは標準の出力に書き込まれます。それぞれのキーの設定は身元確認のために割り当てます

キーテキストの事例：

FrontBase Key Generator Tue Dec 1 12:22:45 1998

Key ident: fc0c719d

DES disk keys:

iv(008):9f68d3d1 e25fb047

k1(008):a2dc2f97 7a1a9b5b

k2(008):f29ba4e0 346bea31

k3(008):16bf92f4 311cc834

RSA private key:

n(128):c7763cbb 9627ff8e 756173f8 2507257c

3b4c349f 0401cc49 8473dff6 9d61752c

020ba963 bd726d91 af249fd0 3b099e3a

543b701c aa26be06 48f7b277 3071db45

9607b9d5 69d2012b 6cc93b62 0536ef69

41b815ce 265e8933 df5ab416 916ee017a

23fe996 99212754 d1168f98 8ba78144
d92bb918 4e5c5c2b d140d79d 602e86ed
e(003):010001
d(128):a61baf84 410e5a63 1719eb6c 31d9fbbb
b3de48cc 4c97fab 16d53124 bdf8158e
6abdaf79 62a1d2e1 ca4be3d3 93f6f490
7ae96cfa 5231257b 32752568 e12d507e
eed424dd dd3e377a a41f8aeb f7158cf7
e5388bf3 adfd2b1a 62bf2377 a1a49809
1d9c79f8 c4d1a25f 26b464da ddbd8e7a
cbb01ca5 847746c2 008821a0 84d93d79
p(064):f8a87b44 77c89ba2 9a70a505 a2618d8c
8814e4d1 e14a920d 868947b0 c10a0542
9477cf9f 7c46e6ed 7c094b8c 85a7f866
81b021c7 06eaa4d5 80c31473 27476b6f
q(064):cd59e60b dfdd7d65 5825ba48 86bcb5b6
7a469109 81bdcb91 4381aa21 83bfe3e5
4db26077 b2001209 e916547a 084dcf2b
299ae74d 5f0fe010 535a1376 bdce3563
dmp1(064):f45a0c98 61e59f1b 910cf4fd ae6e92fc
f09527af 1fe779d3 14cc3ec8 f149f995
4df4d6f0 f0dd73fd 1810c24f 1ae0cf77
5c264b6b 1bdee590 717242dc 4e531ebd
dmq1(064):2e77bf54 c48dac48 788fe0fd 746fd376
bd68a67d 4e3c928a 068a9ae3 069b2f88
bdf2355e 208b5f89 9a9310fe a44e6728
cf3b5c47 e7d101d4 efe79c2b ec7b731d

```
iqmp(064):8fe8b77c 53069c4d 020c2b98 c8fedf59  
5381e3a6 8d437b03 f2950da7 c7d6927b  
ef7b2ea1 e2f17add c5717232 f83bbdcd  
fa373d94 5fc066f3 4453df6d c1250d7a
```

FBKeyGenerator

FrontBase は 3 つのセキュリティオプションを持っています :

- scomm

もし存在しているなら、サーバーとそのクライアントの間の通信は安全です。

- sdisk

もし存在しているなら、ディスクで記憶したデータは暗号化されます

- keys=<file-name>

キーは名前付ファイルから読み込まれます。

もし -sdisk あるいは -scomm の 1 つあるいは両方ともが指定されるなら、サーバーは機能するためにキーの設定を要求します。もし -keys= が指定されるなら、キーはそのファイルから読み込みます、そうでないときはそれらは標準の入力から読み込みます。サーバーがスタートしたとき、それは、指定したキーを、正確に、そして要求するためにチェックします。

FBChangeKey

FBChangeKey ツールは、暗号化のために使ったキーを変更する、あるいは暗号化したデータベースを解読するために、前に暗号化しなかったデータベースを暗号化するために使うことができます。

FBChangeKey <database-file> [<keys-filename>]

もし、提供されないなら、キーは標準の入力から読み込みます。もしあなたが 1 つのキーを提供するなら、もしそれがそうではなかったなら、データベースが暗号化されます、そして、もしそれが暗号化されたなら、解読します。もしあなたが 2 つのキーを提供するなら、データベースは、古いキーによって解読します、そして新しいものを使って暗号化します。2 つのキーのためにテキストを付加することに

よって、2つのキーが作られます。ツールはキーのアイデンティティをチェックします、それであなたがキーテキストを付加する順序は重要ではありません。

IP アドレスチェック

クライアントが FrontBase サーバーに接続するとき、クライアントの IP アドレスはブラックリストとホワイトリストを照らし合わせます。もし IP がブラックリストに載せられたなら、接続は拒否します、もし IP がホワイトリストに載せられたなら、接続は受け入れます。それはホワイトリストとブラックリストの両方で動作するように、リストが取り決めます。もし IP がホワイトリストであるなら、あなたはその IP に安全な通信チャンネルを要求するかどうか指定することができます。ほとんどの場合、ローカル接続が暗号化なしで走ることに問題はないでしょう。

関連したオプションは `-localonly` オプションです、そしてそれは Frontbase が FrontBase サーバーと同じホストで走っているクライアントから接続を可能にするだけであることを確認します。

データタイプ

SQL 9 2 は FrontBase によってサポートされるすべてのデータタイプの広範囲のリストを提供します。さらに、FrontBase は同様に SQL3 から多くのデータタイプをサポートします。データタイプのリストがロング、そして多分混乱するようにさえ思われるけれども、心配しないでください、名前の多くが同じデータタイプ（このように委員会から来る仕事です）を意味します。

- 2 6 4 ページの「TINYINT」。
- 2 6 4 ページの「SMALLINT」。
- 2 6 4 ページの「INTEGER、INT」。
- 2 6 5 ページの「LONGINT」。
- 2 6 5 ページの「DECIMAL[(<precision> [, <scale>])]」。
- 2 6 5 ページの「NUMERIC[(<precision> [, <scale>])]」。
- 2 6 6 ページの「FLOAT[(<precision>)]」。
- 2 6 6 ページの「REAL」。
- 2 6 6 ページの「DOUBLE PRECISION」。
- 2 6 6 ページの「CHARACTER、CHAR」。
- 2 6 7 ページの「NATIONAL CHARACTER、NATIONAL CHAR、NCHAR」。
- 2 6 7 ページの「CHARACTER VARYING、CHAR VARYING、VARCHAR」。
- 2 6 8 ページの「NATIONAL CHARACTER VARYING、NATIONAL CHAR VARYING、NCHAR VARYING」。
- 2 6 8 ページの「BIT」。
- 2 6 8 ページの「BIT VARYING」。
- 2 6 9 ページの「BYTE」。
- 2 6 9 ページの「DATE」。
- 2 6 9 ページの「TIME」。
- 2 7 0 ページの「TIME WITH TIME ZONE」。
- 2 7 0 ページの「TIMESTAMP」。
- 2 7 0 ページの「TIMESTAMP WITH TIME ZONE」。

- 271 ページの「INTERVAL」。
- 271 ページの「BLOB」。
- 272 ページの「CLOB」。
- 272 ページの「BOOLEAN」。

TINYINT

8 ビットの整数として実装します。

事例：

```
CREATE TABLE T0(C0 TINYINT, ...);
```

SMALLINT

16 ビットの整数として実装します。

事例：

```
CREATE TABLE T0(C0 SMALLINT, ...);
```

INTEGER, INT

32 ビットの整数として実装します。明白な使用を別として、このデータタイプはよく単独の行の主キーのために使います。もしあなたが EOF を使っているなら、あなたは EOF の自動で作られた主キーを使うことを調べることができます、そして EOF としてのバイトタイプはデータベースアクセスなしでキーを作ることができます。トレードオフは 12 バイトの主キーです、従って、作ることが理解できないのに対して、32 ビットの整数はかなりシンプルです。

事例：

```
CREATE TABLE T0(C0 INTEGER PRIMARY KEY, ...);
```

LONGINT

64ビットの整数として実装します。

事例：

```
CREATE TABLE T0(C0 LONGINT, ...);  
DECIMAL[ ( <precision> [ , <scale> ] ) ]
```

サインと指数を維持するために128ビットの整数 + 32ビットとして実装します。<precision> のためのデフォルト値は、<scale> のために38（最大）と0です。この表示は NSDecimalNumber のそれとまったく同じです。もしあなたが固定小数点を必要とするなら、これはそのデータタイプです。10進数のポピュラーな用途は通貨の値です。

注：FrontBase は、10の表示をベースを使うことによって、精度を失いません。もしあなたが例えば1.23を挿入するなら、これは1.229994599ではなく、その値を記憶して、返します。これは同様に NUMERIC、FLOAT と REAL と DOUBLE PRECISION（下を参照）データタイプを用います。

事例：

```
CREATE TABLE T0(C0 DECIMAL, PROFITS DECIMAL(20,2), ...);
```

NUMERIC[(<precision> [, <scale>])]

サインと指数を維持するために64ビットの整数 + 32ビットとして実装します。<precision> のためのデフォルト値は <scale> のために19（最大）と0です。もしあなたが38桁の精度を必要としない（そして従って記憶装置の要求を減らす）なら、NUMERIC は10進数の代わりに使うことができます。

事例：

```
CREATE TABLE T0(C0 NUMERIC, SALARY NUMERIC(10,2), ...);
```

FLOAT[(<precision>)]

サインと指数を維持するために 64 ビットの整数 + 32 ビットとして実装します。<precision> のためのデフォルト値は 19 (最大) です。

事例 :

```
CREATE TABLE T0(C0 FLOAT, C1 FLOAT(10), ...);
```

REAL

サインと指数を維持するために 64 ビットの整数 + 32 ビットとして実装します。<precision> のためのデフォルト値は 19 (最大) です。あなたが、FLOAT を使うとき、最大の精度を指定することができますこと以外、REAL と FLOAT が同じく実行されます。

事例 :

```
CREATE TABLE T0(C0 REAL, ...);
```

DOUBLE PRECISION

サインと指数を維持するために 128 ビットの整数 + 32 ビットとして実装します。<precision> のためのデフォルト値は 38 (最大) です。多くの目的のために、これは `NSDecimalNumber/` `java.math.BigDecimal` をマップするために最も良い選択です。詳細については、「FrontBase のファンデーション / Java オブジェクトをマップすること」の文書を見てください。

事例 :

```
CREATE TABLE T0(C0 DOUBLE PRECISION, ...);
```

CHARACTER, CHAR

文字列を従来の固定された長さの文字として実装します。FrontBase がユニコードをサポートして、そしてすべての文字列を UTF8 エンコーディングに記憶することに注意してください。これは ASCII 以外の値の文字列が文字数より多くのバイトを占領することを意味

します。ほとんどの ASCII でない文字、例えば æøǎÆØǺ、UTF8 フォーマットでコード化されるとき、2 バイトを占領します。

注：文字の値の最大の長さは 2 GB です。

事例：

```
CREATE TABLE T0(C0 CHAR(1), C1 CHARACTER(100000), ...);
```

NATIONAL CHARACTER, NATIONAL CHAR, NCHAR

FrontBase がユニコードをサポートするとき、NATIONAL CHARACTER のデータタイプは文字にマップします。

事例：

```
CREATE TABLE T0(C0 NATIONAL CHAR(1), C1 NCHAR(100000), ...);
```

CHARACTER VARYING, CHAR VARYING, VARCHAR

文字列を従来の可変的な長さの文字として実装します。可変的な長さの文字列の実装は非常に効率的です、そして非常に長い文字列を使って関連づけられた余分なオーバーヘッドがありません。（それが固定した長さの文字列であるかのように）16 バイトまでの長さの文字列が列のレコードで直接記憶します。いわゆるスペルの表がそれぞれの表で関連づけられます、そして表の列に挿入したすべて同一の可変的な長さの文字列は1度記憶するだけです。

注：FrontBase が非常に効率的に `varchars` をコード化する、可変的な長さの文字列の使用は一般的に、固定した長さの文字列以上を推薦します。

事例：

```
CREATE TABLE T0(C0 VARCHAR(128), C1 CHARACTER VARYING(200000), ...);
```

NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING, NCHAR VARYING

FrontBase がユニコードをサポートするとき、NATIONAL CHARACTER VARYING のデータタイプはすべて CHARACTER VARYING にマップします。

事例：

```
CREATE TABLE T0(C0 NATIONAL CHAR VARYING(10), C1 NCHAR VARYING(10000),...);
```

BIT

ビットデータタイプは概念的に 1 のと 0 の文字列です、しかし、不透明なバイナリのデータタイプ、すなわちビット（8）が 1 バイトを占領するとき、実行します。

下に関連する EOF と BYTE を見てください。

事例：

```
CREATE TABLE T0(C0 BIT(32), C1 BIT(256)...);
```

BIT VARYING

BIT として、しかしビットの文字列は可変的な長さであるという明白な例外があります。

事例：

```
CREATE TABLE T0(C0 BIT VARYING(32), C1 BIT VARYING(256)...);
```

BYTE

単純なラッパは BIT、すなわち BYTE(n) のために BIT(n*8) とまったく同じです。このデータタイプは SQL92 の規格の一部ではありません、しかしもっと良い EOF の自動的な主キーの生成をサポートするために導入しました。もしあなたが 1 2 バイトのバイナリキーを使うなら、データベースサーバーを往復しないで、EOF は自動的に主キーを作ります（そしてそれでトランザクションを始めさせる）ことができます。

事例：

```
CREATE TABLE T0(C0 BYTE(12), ...);
```

DATE

従来の日付データタイプ。DATE は時間の構成要素を含んでいないことに注意してください。DATE 値は内面的に秒 (2 0 0 1 - 0 1 - 0 1 がゼロです) として示して、そして NUMERIC(0) 値として記憶します。

事例：

```
CREATE TABLE T0(C0 DATE, ...);
```

TIME

完全な時刻表示の時間の構成要素だけを維持します。TIME 値（「12:34:23」）は内面的に秒として示して、そして NUMERIC 値として記憶します。どうか、否定的に TIME 値がサーバertimeゾーンで表されると想定されることに注意してください、すなわち、それが挿入される時、サーバertimeゾーンは時間の値を用います。

事例：

```
CREATE TABLE T0(C0 TIME, ...);
```

TIME WITH TIME ZONE

TIME として、時間帯のオフセット以外は、時間の値（「12:34:23 - 08:00」）を使って含まれ、そして記憶します。明確な時間帯がクライアントに返されます。

事例：

```
CREATE TABLE T0(C0 TIME WITH TIME ZONE, ...);
```

TIMESTAMP

日付と時間の構成要素の両方を含む完全な時刻表示の値を維持します。TIMESTAMP 値（「2001-01-24 12:34:23」）は内面的に秒（2001-01-01 がゼロです）として示して、そして NUMERIC 値として記憶します。TIMESTAMP 値がサーバーの時間帯に表されることに注意してください、すなわち、それが挿入される時、サーバーの時間帯は時間の値を用います。これは TIMESTAMP 値がクライアントと異なっている時間帯を持って終わることを意味します！

事例：

```
CREATE TABLE T0(C0 TIMESTAMP, ...);
```

TIMESTAMP WITH TIME ZONE

TIMESTAMP として、時間帯のオフセット以外は、時間の値（「2001-01-24 12:34:23-08:00」）を使って含まれ、そして記憶されます。明確な時間帯はクライアントに返されます。もしあなたが時間帯の情報をどのように記憶して、そして表示したかに関する完成したコントロールを必要とするなら、このデータタイプを必要とします。

事例：

```
CREATE TABLE T0(C0 TIMESTAMP WITH TIME ZONE, ...);
```

INTERVAL

INTERVAL は実際に 2 つの個々のデータタイプです：年・月の間隔のデータタイプと日・時間の間隔のデータタイプ。

年・月の間隔が内面的に月として示されて、そして 32 ビットの整数として記憶します。

日・時間の間隔が内面的に秒として示されて、そして NUMERIC の値として記憶します。

インタバルを使う 1 つの方法は、日付と時刻表示を操るとき、例えば日あるいは月を加えるとき、次の通りです：

```
DATE '2000-01-25' + INTERVAL '02' MONTH (result: DATE '2000-03-25' )
```

あるいは

```
DATE '2000-02-28' + INTERVAL '02' DAY (result DATE '2000-03-01' )
```

事例：

```
CREATE TABLE T0(C0 INTERVAL YEAR TO MONTH, C1 INTERVAL MONTH, ...);  
CREATE TABLE T1(D0 INTERVAL DAY TO SECOND, C1 INTERVAL HOUR, ...);
```

BLOB

バイナリの大きいオブジェクトは不透明なバイナリのデータタイプです、すなわちあなたが記憶するバイトは方法でインタプリートしなくて、そして挿入するときと同じ形式で返します。FrontBase は、サーバー側のストリーミングをインクルードする、すなわち、必要なコピーを非常に効率的に BLOBs を実行します。クライアント側のストリーミングは FrontBase 2.0 でイネーブルに設計されています。BLOB 値は 2 GB のサイズまでできます。

事例：

```
CREATE TABLE T0(C0 BLOB, ...);
```

CLOB

文字の大きいオブジェクトは、大文字のデータタイプです、すなわちあなたは、検索することを望まない、そして拡大した効率を望む文字列が標準的な CHARACTER / VARCHAR 値（例えば INSERT あるいは UPDATE SQL 文にコピーする）を比較しました。CLOBs は BLOBs と同様に効率的に実行します。CLOB 値は、クライアント側で起きているエンコーディングとデコーディングを使って UTF8 フォーマットにエンコードします。

事例：

```
CREATE TABLE T0(C0 CLOB, ...);
```

BOOLEAN

符号なしのバイトとして実装しました。SQL 9 2 は 3 値論理学を使うことに注意してください、すなわち可能な値は FALSE (0)、TRUE (1) と UNKNOWN (255) です。

事例：

```
CREATE TABLE T0(C0 BOOLEAN, ...);
```

FrontBase データタイプの中の Foundation/Java のマップするオブジェクト

次のことは FrontBase データタイプの最も普通のファンデーション / Java オブジェクトの勧めるマッピングの概要です。

String

NSString java.lang.String java.io.Reader (for CLOB)	提案した データタイプ :	CHARACTER, VARCHAR, CLOB
	推薦 :	VARCHAR

注：もしあなたが大きい文字列を動作しているなら、そしてもしあなたがたいがい、文字列で検索を行うつもりではないなら、CLOB データタイプは良い選択です。CLOB データタイプの主な利点は文字列が別にデータベースに送られるということです。すなわち、むしろ、データベースは、莫大な SQL 文を解析して、そしてコピーしてください、それはその代わりに効率的にバイナリのチャンネルにデータを転送することができます。VARCHAR の実装が非常に効率的で、そして同一の文字列の再利用を可能にするとき、VARCHAR は CHARACTER に勧められた、概要です。あなたはどんな性能のペナルティーでも心配しないで「制限がない」 VARCHAR 定義（例えば VARCHAR(1000000)）を使うことができます。文字列の内面的な表示はクライアント側で起きているエンコーディング/デコーディングで UTF8 です、すなわち NSString/java.lang.String の本来のメソッドによって処理されました。

整数

NSNumber java.lang.Number	提案した データタイプ:	BOOLEAN, SMALLINT, INTEGER, FLOAT, REAL, DOUBLE PRECISION
	推薦:	実際の使用に依存する整数 (INTEGER) あるいは倍精度 (DOUBLE PRECISION)。

注: NSNumber/Number は C/Java 番号のデータタイプのためにカバーするとき、選択する FrontBase タイプは実際の使用に依存します。

10進数

NSDecimalNumber java.math.BigDecimal	提案した データタイプ:	NUMERIC, DECIMAL, REAL, FLOAT, DOUBLEPRECISION
	推薦:	DOUBLE PRECISION, DECIMAL

注: もしあなたが正確な計算のために NSDecimalNumber を使っているなら、それは直接、倍精度 (DOUBLE PRECISION) (あるいはもし整数であるなら 10進数 (DECIMAL)) にマップします。もしあなたが記憶装置の要求を減らすことを望むなら、あなたはその代わりに FLOAT/NUMERIC (縮小は列・値毎に 8 バイトです) を使うことができます。

日付

NSDate java.sql.Date java.sql.Timestamp)	提案した データタイプ:	DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE
	推薦:	TIMESTAMP

注: TIMESTAMP WITH TIME ZONE は直接 NSDate によって提供されたデータと等しいです。もしあなたが異なった時間帯のデータを扱わなくてもよいなら、あるいは、もしあなたが常にクライアントの時間帯を使って表示したデータを必要とするなら、あなたは TIMESTAMP を考慮することができます。

時間

java.sql.Time	提案した データタイプ:	TIME, TIME WITH TIME ZONE
	推薦:	Either.

注: java.sql.Time は java.sql.Date の周りに単純なラッパー（あるいはただ平凡なぞっとするようなハック）です。

ストリームデータ

NSData java.io.InputStream	提案した データタイプ:	BLOB
	推薦:	BLOB

主キー

提案した
データタイプ: INTEGER, BYTE(12)

推薦: Either.

注: BIT(96) のためにカバーである BYTE(12) は、もし主キーの生成が EOF (クライアント側の計算機能) によってされるなら、データタイプを使ってください。どんな FrontBase データタイプでも主キーのために使うことができます、そしてすべての数のデータタイプが正確な数値として示されるので、主キーの行のために、例えば倍精度 (DOUBLE PRECISION) あるいは時刻表示 (TIMESTAMP) をデータタイプとして用いることは安全です。しかし、EOF はある特定のデータタイプ (DOUBLE PRECISION, BLOB と CLOB) が主キーの行のために指定することを可能にしません、それで、もしあなたが EOF を使っているなら、あなたはこれを考慮する必要があるでしょう。FrontBase はマルチコラムであるか、あるいは複合的な主キーをサポートするのに対して、もしあなたが EOF を使って集合キーを使っているなら、あなたは若干の注意を用いてください。

主キーと自動生成

FrontBase は、SQL92 の規格、表 (multiple-column) の主キーによって要求するように、サポートします、しかし、リスト (single-column) の整数の主キーの場合、FrontBase はこのようなキーを生成することを助けることができます。

キーの生成

FrontBase で、それぞれの表が関連づけられたカウンターを持っています。カウンターは次の SQL 文で、アクセスして、そして増やします:

```
SELECT UNIQUE FROM <table>;
```

それは単独の整数の行を使って単独の列を返します。SELECT UNIQUE の概念は同様に scalar サブクエリとして使うことができます:

```
INSERT INTO <table> VALUES(SELECT UNIQUE FROM <table>, ...);
```

表を作るとき、関連づけられたカウンターは1 0 0 0 0 0 0の初期値に設定します。

カウンターは2つの方法で設定することができます：

```
SET UNIQUE=<value> FOR <table>;
```

あるいは

```
SET UNIQUE FOR <table>(<column>;
```

後で、ショートハンドであるために：

```
SET UNIQUE=(SELECT MAX(<column>)+1 FROM <table>) FOR <table>;
```

特別なデフォルト値を指定することによって、新しい列の主キーの行が自動的に設定することができます：

```
ALTER TABLE <table> ALTER <column> SET DEFAULT UNIQUE;
```

もし行のための明確な値が INSERT 文で与えられないなら、デフォルトを使うでしょう。

列レベルの特権

定義すること

FrontBase は唯一の特徴が列レベルの特権を呼び出すことを提供します、そしてそれはあなたに個別の列のためにアクセス権を指定することを可能にします。それぞれの列が特定のユーザと特定のグループに属することによって所有すると言われています。アクセス権 (SELECT、UPDATE、と DELETE) は、列のために所有者、グループと世界のために指定することができます。

注：列レベルの特権は、個々のオプションとしてライセンスを与えません、そしてその機能がフリーのバージョンで利用可能ではありません。

展開する

列レベルの特権の機能を使うために、所定のデータベースが与えた機能をオプションとして初期化してください：

```
/Local/Library/FrontBase/bin/FrontBase -rlpriv <database name>
```

FBDatabaseManager によってデータベースを作るとき、あなたは同様に `-rlpriv` オプションを指定することができます。

作られた途端に、オプションはデータベースに記録します、すなわち、データベースサーバーがその後、ストップして、そしてスタートしたとき、あなたはオプションを指定する必要がありません。

メタデータを管理する

```
CREATE GROUP <group name>;  
    -- CURRENT_USER must be _SYSTEM  
DROP GROUP <group name> RESTRICT|CASCADE;  
    -- CURRENT_USER must be _SYSTEM
```

```
ALTER GROUP <group name> ADD USER <user name>;  
    -- CURRENT_USER must be _SYSTEM  
ALTER GROUP <group name> DROP USER <user name>;  
    -- CURRENT_USER must be _SYSTEM
```

```
ALTER USER <user name> SET DEFAULT GROUP <group name>;  
    -- CURRENT_USER must be _SYSTEM or <user name>  
ALTER TABLE <table name> SET DEFAULT PRIVILEGES(<row privileges>)  
    [USER <user name>];  
    -- CURRENT_USER must be _SYSTEM or <user name>, if no  
    user name  
    -- is given, the current user is used  
<row privileges> ::= <row privs> | <row privileges> , <row privs>  
<row privs>      ::= <owner privs> | <group privs> | <world privs>  
<user privs>    ::= USER = * | <priv mask>  
<group privs>  ::= GROUP = * | <priv mask>  
<world privs>  ::= * = * | <priv mask>
```

<priv mask>	::= <priv>		<priv mask> + <priv>
<priv>	::= SELECT		UPDATE DELETE

事例 :

```
ALTER TABLE TO SET DEFAULT PRIVILEGES(USER=*,  
GROUP=SELECT+UPDATE, *=SELECT);
```

データの内容を管理する

```
UPDATE <table name> SET PRIVILEGES(<row privileges>) [WHERE <cond  
expr>];  
UPDATE <table name> SET GROUP <group name> [WHERE <cond expr>];  
UPDATE <table name> SET USER <user name> [WHERE <cond expr>];  
-- CURRENT_USER has to either own the row or be _SYSTEM
```

列のためにアクセス権を選択する

列の所定の設定のための所有者、グループと特権は次のように取り出すことができます :

```
SELECT USER, GROUP, PRIVILEGES FROM <table> WHERE <cond expr>;
```

VIEW での SELECT をラップすることによって、値はクエリで使うことができます :

```
CREATE VIEW(ROW_OWNER, ROW_GROUP, ROW_PRIVS) TO _PRIVS  
SELECT USER, GROUP, PRIVILEGES FROM T0;  
SELECT * FROM TO _PRIVS WHERE ROW_OWNER = '<user name>';
```

照合 (Collations) があなたのためにできること

照合 (Collations) は基本的にあなたが2つの文字がどのように比較されるか、あるいは、どちらかと言えば2つの所定の文字が、等しいか、それ以下か、それ以上かを比較するかをコントロールする方法です。

なぜこれを気にするのですか？

照合 (collations) を気にしなければならない2つの主な理由があります：

1. 国際的な文字
2. 大文字小文字の違いを無視する比較オペレーション

国際的な文字

FrontBase はユニコードを実行します、従って、漢字、ハングルなどを含むすべてのいわゆる国際的な文字の使用をサポートします。少なくとももし順序が、ほとんどの人々がそれを予想する結果になるなら、ユニコード領域の国際的な文字の位置の値は2文字の順序のために使うことができません。

事例：小文字 C が序数の値として99を持っている間に、フランス語の文字 _ (Cedilla を使ったラテンの小さい文字 C) は231 (10進数) の序数の値を持っています。もし例えば _ と d が比較されるなら、d は _ より小さいために比較されるでしょう、そしてそれはあなたが必要とするものではないかもしれません。

大文字小文字の違いを無視する比較オペレーション

通常、文字列はそれらがユーザによって入力したのと同じ文字を使ってデータベースに記憶します。若干のユーザは、小文字で入力することを好みます、他のユーザは、大文字で入力することを好みます、そして少数は FrontBase のように大文字化を使います。検索するとき、ユーザは一般に文字の大きさがどちらで登録されたか分かりません、すなわち検索がこれをしなければなりません。

次のようにすることによって、これを扱うことができます：

```
SELECT * FROM T0 WHERE UPPER(CITY) = 'COPENHAGEN';
```

上記の SELECT を使って問題は T0 で定義した索引です。CITY を使うことはできません、すなわち SELECT はもし索引が使われているなら、もっとゆっくり実行するでしょう。

いわゆる COLLATION を定義することによって、あなたは効果的に文字がどのように並べられるか決めることができます、すなわち、順序の値の中に序数の値をマップします。これは例えば、もし「a」が「A」と同じ順序の値の中にマップされるなら、「a」は「A」と等しいとみなされることを意味します。

FrontBase ディストリビューションは、CaseInsensitive.coll1 (<FB home>/Collations ディレクトリに位置している) と呼ばれる照合 (collation) 表を含みます、そしてその名前によって意味されるように、この照合 (collation) は大文字小文字の違いを無視する比較を使うことができます。

最初にあなたは照合 (collation) を定義する必要があります：

```
CREATE COLLATION CASE_INSENSITIVE
FOR INFORMATION_SCHEMA.SQL_TEXT
FROM EXTERNAL('CaseInsensitive.coll1');
COMMIT;
```

表を作るとき、照合 (collation) を使います：

```
CREATE TABLE T0(
    ...
    DB VARCHAR(128) COLLATE CASE_INSENSITIVE,
    ...
);
CREATE INDEX ON T0(DB);
COMMIT
```

DB 行の値がもう 1 つの文字列と比較されるときはいつでも、指定した照合 (collation) は自動的に使われるでしょう、これは、索引をビルドするとき、比較することを含みます。

事例：

```
INSERT INTO T0(DB) VALUES 'frontbase', 'FrontBase', 'FRONTBASE';  
COMMIT;  
SELECT DB FROM T0 WHERE DB = 'FrOnTbAsE';
```

– すべての3つの列を返すでしょう

```
SELECT DB FROM T0 WHERE DB LIKE 'f%';
```

– すべての3つの列を返すでしょう

もしあなたが若干の理由のために大文字小文字の違いを比較することを望むなら、あなたは (FBUnicodeManager アプリケーションを使っている) アイデンティティの照合 (collation) を定義して、そして例えば <FB home>/Collations ディレクトリの CaseSensitive.coll1 として照合 (collation) として保存する必要があります。

```
CREATE COLLATION CASE_SENSITIVE  
  FOR INFORMATION_SCHEMA.SQL_TEXT  
  FROM EXTERNAL('CaseSensitive.coll1');  
COMMIT;  
SELECT DB FROM T0 WHERE DB = 'FrontBase' COLLATE CASE_SENSITIVE;
```

– 1つの列を返すでしょう

```
SELECT DB FROM T0 WHERE DB LIKE 'F%' COLLATE CASE_SENSITIVE;
```

– 2つの列を返すでしょう

上記の2つの SELECTs は DB 行で作られた索引を使わないことに注意してください、すなわち大きい表のためにこれらの2つの SELECTs はもし索引を使うなら、もっとゆっくり実行するでしょう。

今もしあなたが大文字小文字の違いを無視して検索することを望んで、そして、正確な文字の一致が同様に用いられることを要求することによって、さらに設定された結果を制限するならどうでしょう？簡単に終わります：

```
SELECT DB FROM T0 WHERE  
DB = 'FrontBase'  
AND  
DB = 'FrontBase' COLLATE CASE_SENSITIVE;
```

最初の WHERE コンポーネントは「frontbase」、「FrontBase」、「FRONTBASE」を返すのに対して、2 番目の WHERE コンポーネントは「FrontBase」に結果を減らすでしょう；

あなた自身のアプリケーションあるいはソリューションの中に FrontBase を組み込む

エンドユーザが所定のプラットフォームのために FrontBase をダウンロードして、そしてインストールするとき、インストールは一般にプラットフォームのために特定のデフォルトの場所に入るでしょう。FrontBase は、インストールの後に、アクセスが認められるすべてのアプリケーションなどにアクセスできます。

FrontBase がもう 1 つのアプリケーションあるいはソリューションの中に組み込まれるとき、通常、FrontBase は所定のアプリケーションを使って機能している、そしてすでにインストールされている FrontBase の標準的なエンドユーザバージョンから独立していることを確認することは望ましいです。

技術的にこれは次のことを意味します：

1) FrontBase は親アプリケーションあるいはソリューションの標準的なディレクトリ構造の中にインストール（「組み込み」）されます、すなわち標準的な FrontBase インストーラパッケージは使われません。

2) アプリケーションあるいはソリューション特定のライセンスの文字列が使われます。

このライセンスの文字列は一般的です、すなわちそれは特定の IP あるいは MAC アドレスに結び付けられません。

3) 所定のホストに FrontBase データベースのための DNS サービスのような FBExec は、親アプリケーションあるいはソリューションがポート番号を使ってデータベースと接続することを意味して、使われません。

ポート番号はライセンスの文字列の中に組み込まれる（変更できない）でしょう。

4) 組み込みのライセンス文字列は (1) の FrontBase データベースの 1 つを使って動作してアプリケーションあるいはソリューションのために可能にします。

ディレクトリの構造

FrontBase が親アプリケーションのディレクトリの構造の中に完全に組み込むことができるけれども、同様に FrontBase ディレクトリの構造の概念がまだなければなりません。FrontBase サーバーは、実際のデータベースファイルを含んで必要とする一部の位置を示す "relative to where I am" スキームを使います。

標準的なエンドユーザインストールのための FrontBase のディレクトリの構造とファイルは通常次の通りです：

```
FrontBase/Collations/CaseInsensitive.coll1
FrontBase/Databases
FrontBase/Java/frontbasejdbc.jar
FrontBase/Library/DefinitionSchema.sql
FrontBase/Library/InformationSchema.sql
FrontBase/Library/OpenBaseImport.sql
FrontBase/Library/KeyWords.txt
FrontBase/Library/FBSQLErrors.array
FrontBase/Library/*.ucm
FrontBase/LicenseString
FrontBase/Templates/*
FrontBase/TransactionLogs
FrontBase/Translations/ToLower.trans
FrontBase/Translations/ToUpper.trans
FrontBase/bin/
FrontBase/include/*
FrontBase/lib/libFBCAccess.a
```

FrontBase のディレクトリのそれぞれのサブディレクトリの関数は次の通りです：

照合 (Collations)

すべての照合 (collation) の定義のためのプレースホルダーが所定のデータベーススキーマによって使われます。もしあなたのスキーマが照合 (collations) を利用しないなら、このディレクトリは空にするあるいは取り出すことができます。

データベース

実際のデータベースファイルを持つでしょう。実際のデータベースファイルはホストファイルシステムのどこにでも位置することができるけれども、それがサポートをより容易にするとき、このディレクトリを使うことを勧めます。

Java

FrontBase のために JDBC ドライバーを持ちます。このディレクトリは空にするあるいは取り出すことができます。

ライブラリ

incl. ファイルは新しいデータベースをブートストラップする間に使った種々のハウスキーピングファイルを持っています。もし親アプリケーションが、ブートストラップする前のデータベースをインクルードするなら、InformationSchema.sql と DefinitionSchema.sql ファイルは必要としません。もし FrontBase がインストールの一部として、新しいデータベースをブートストラップするために、必要とするなら、これらの2つのファイルはライブラリディレクトリで利用可能です。

FBSQLErrors.array ファイルは原文のエラーメッセージをサーバーによって返されるエラーメッセージをマップするためにクライアントアプリケーションで使うパラダイムエラーメッセージのリストです。親アプリケーションは一般に、それ自身の中にこのファイルを組み込む、(あるいは無視する) でしょう。FBSQLErrors.array はローカライズされたエラーメッセージを提供するために編集することができます。このファイルはサーバーによって使われません。

.ucm ファイルは選ばれた文字の設定のサーバーによって返された UTF8 にコード化した文字列をマップするためにクライアントアプリケーションで使われます。

クライアントアプリケーションは所定の文字の設定の中に UTF8 をマップする他の手段を使うために決めることができます。.ucm ファイルはサーバーによって使われません。このディレクトリの他のファイルを取り出すことができます。

テンプレート

HTML ファイルは FBWebManager によって使いました。このディレクトリは空にするあるいは取り出すことができます。

TransactionLogs

サーバーによって独占的に作成して、そして維持します。このディレクトリは削除されません。変換 UPPER と LOWER (SQL 関数) をサポートするために2つの変換ファイルを持ちます。もし親アプリケーションが LOWER と UPPER を使わないなら、このディレクトリを空にする、あるいは取り出すことができます。

bin

Frontbase ディストリビューションの実行可能プログラム (バイナリ) を持ちます。FrontBase サーバーの実行可能プログラムだけを必要とします、しかし実行可能な sql92 (コマンドラインツール) は同様にインクルードするために有利です。

インクルード

開発者によって使われたさまざまなファイルを持ちます。このディレクトリは空にするあるいは取り出すことができます。

lib

デベロッパーによって使われたさまざまなファイルを持ちます。このディレクトリはからあるいは取り出されることがあります。

組み込みの展開

組み込みの状況で展開することができる最小の FrontBase ディレクトリの構造は次の通りです：

```
FrontBase/Databases/<pre-bootstrapped database>  
FrontBase/LicenseString  
FrontBase/TransactionLogs  
FrontBase/bin/FrontBase
```

組み込みの状況で展開することができる推薦された FrontBase ディレクトリの構造は次の通りです：

```
FrontBase/Collations/CaseInsensitive.coll1  
FrontBase/Databases  
FrontBase/Library/DefinitionSchema.sql  
FrontBase/Library/InformationSchema.sql  
FrontBase/LicenseString  
FrontBase/TransactionLogs  
FrontBase/Translations/ToLower.trans  
FrontBase/Translations/ToUpper.trans  
FrontBase/bin/FrontBase  
FrontBase/bin/sql92
```

FrontBase サーバーをスタートします – Windows NT/2000/XP

ウィンドウズプラットフォームは、それ自身の記述を正当化するために十分に他のすべてをサポートするプラットフォームから外れます。

インストールプロセスの間に、FrontBase サーバーとデータベースは標準的なサービスアプリケーションとしてインストールしてください：

```
<drive>:<install path>%FrontBase%bin%FrontBase -install  
<database name>
```

FrontBase サーバーは他のサービスアプリケーションのようにスタートして、そしてストップすることができます、例えば自動的にサービスコントロールマネージャーによって、そして/あるいは親アプリケーションによってプログラマチックに。

現在、サーバーは、デフォルト毎に、C:/usr/FrontBase/Databases の中にデータベースファイルを作成しようとするでしょう。FB_HOME_DRIVE と呼ばれるシステム全体の環境変数を定義することによって、データベースのディレクトリは適切な場所に位置することができます。一般に FB_HOME_DRIVE 変数は、組み込みの状態で、次のように定義します

```
<drive>:<install path>\FrontBase\
```

FrontBase サーバーをスタートします - Windows NT/2000/XP 以外の他のプラットフォーム

サーバーは他のバックグラウンドアプリケーションとしてスタートします：

```
<install path>/FrontBase/bin/FrontBase [<options>] <database  
name> &
```

もしデータベース名を前接続しているパスの情報がないなら、サーバーはデータベースがデータベースディレクトリに存在すると想定するでしょう。

サーバーに何のポート番号を使うか示す方法

ポート番号は LicenseString ファイルで変更できません、 -port<number> オプションの使用は無視されます。