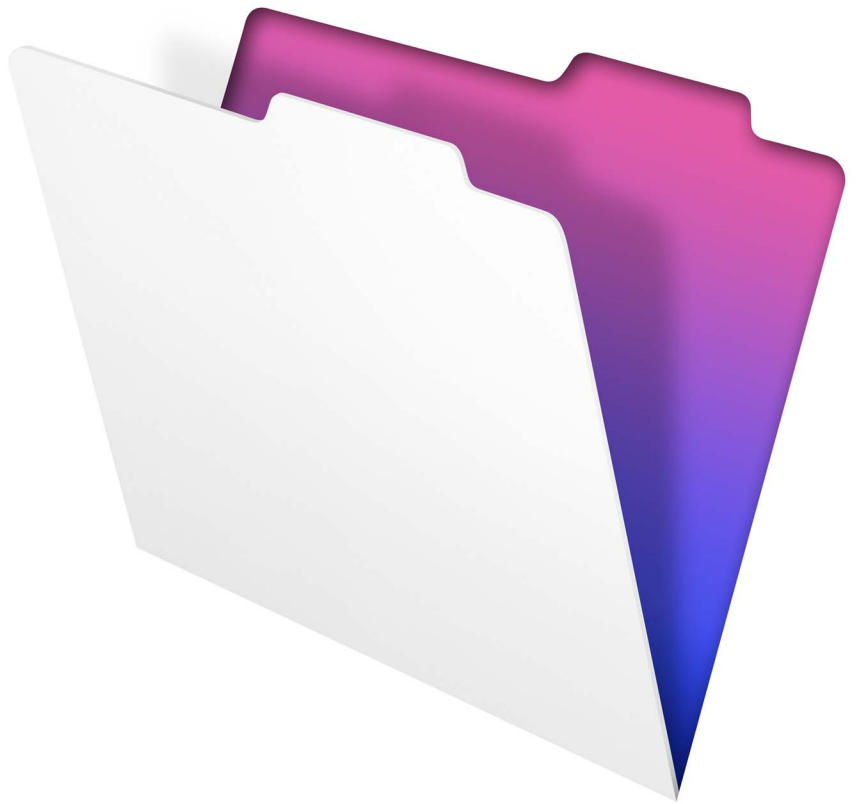


# FileMaker® 11

## ODBC と JDBC ガイド



© 2004-2010 FileMaker, Inc. All Rights Reserved.

FileMaker, Inc.

5201 Patrick Henry Drive

Santa Clara, California 95054

FileMaker、ファイルメーカーは、FileMaker, Inc. の米国及びその他の国における登録商標です。ファイルフォルダロゴは、FileMaker, Inc. の商標です。その他のすべての商標は該当する所有者の財産です。

FileMaker のドキュメンテーションは著作権により保護されています。FileMaker, Inc. からの書面による許可無しに、このドキュメンテーションを複製したり、頒布することはできません。このドキュメンテーションは、正当にライセンスされた FileMaker ソフトウェアのコピーがある場合そのコピーと共にのみ使用できます。

製品及びサンプルファイル等に登場する人物、企業、E メールアドレス、URL などのデータは全て架空のもので、実在する人物、企業、E メールアドレス、URL とは一切関係ありません。スタッフはこのソフトウェアに付属する「Acknowledgements」ドキュメントに記載されます。他社の製品 及び URL に関する記述は、情報の提供を目的としたもので、保証、推奨するものではありません。

詳細情報については [www.filemaker.co.jp](http://www.filemaker.co.jp) をご覧ください。

第 01 版

# 目次

## 第 1 章

### 概要

このガイドについて	7
ODBC および JDBC について	7
ODBC クライアントアプリケーションとしての FileMaker ソフトウェアの使用	8
ODBC データのインポート	8
リレーションシップグラフへの ODBC テーブルの追加	8
データソースとしての FileMaker データベースの使用	8
ホストされる FileMaker Pro データベースへのアクセス	8
サードパーティ製ツールに関する制限	9
ネットワークを使用する場合の条件	9
旧バージョンのファイルの更新	10
最新のドライバのインストール	10

## 第 2 章

### 外部 SQL データソースへのアクセス

ODBC データのインポート	11
ODBC によるデータソースの操作のための SQL の実行	12
リレーションシップグラフでの ODBC テーブルの操作	12
FileMaker 11 でサポートされるデータソース	13
リレーションシップグラフへの ODBC テーブルの追加	13

## 第 3 章

### FileMaker ODBC クライアントドライバのインストール

必要なハードウェアとソフトウェア	15
ODBC クライアントドライバの必要条件 (Windows)	15
ODBC クライアントドライバの必要条件 (Mac OS)	15
ネットワークを使用する場合の必要条件	15
ODBC クライアントドライバのインストール (Windows)	15
クライアントドライバの設定 (Windows)	16
ODBC クライアントドライバのインストール (Mac OS)	17
クライアントドライバの設定 (Mac OS)	18
この後の作業を開始するにあたって	19

## 第4章

### ODBC を使用した FileMaker データの共有

ODBC について	21
ODBC クライアントドライバの使用	22
FileMaker データベースファイルへのアクセスの概要	22
Windows アプリケーションからの FileMaker データベースファイルへのアクセス	23
FileMaker DSN 用の ODBC クライアントドライバのプロパティの指定 (Windows)	23
ODBC によるアクセスの確認 (Windows)	24
Mac OS アプリケーションからの FileMaker データベースファイルへのアクセス	24
FileMaker DSN 用の ODBC クライアントドライバのプロパティの指定 (Mac OS)	24
ODBC によるアクセスの確認 (Mac OS)	25

## 第5章

### FileMaker JDBC クライアントドライバ

必要なソフトウェア	27
ネットワークを使用する場合の必要条件	27
JDBC クライアントドライバのインストール	27
JDBC クライアントドライバの使用	28

## 第6章

### JDBC を使用した FileMaker データの共有

JDBC について	29
JDBC クライアントドライバの使用	29
JDBC クライアントドライバについて	29
JDBC URL を使用したデータベースへの接続	30
URL サブネームでのドライバプロパティの指定	31
複数の FileMaker データベースファイルを使用したソリューション	32
JDBC によるアクセスの確認	32

## 第7章

### サポートされている標準

Unicode 文字のサポート	33
SQL ステートメント	33
SELECT ステートメント	33
SQL 句	34
FROM 句	34
WHERE 句	35
GROUP BY 句	35
HAVING 句	35
UNION 演算子	36
ORDER BY 句	36
FOR UPDATE 句	36
DELETE ステートメント	38
INSERT ステートメント	39
UPDATE ステートメント	40
CREATE TABLE ステートメント	41
ALTER TABLE ステートメント	41
CREATE INDEX ステートメント	42
DROP INDEX ステートメント	42
SQL 統計関数	42
SQL 式	43
フィールド名	43
定数	43
指数または科学表記	44
数値演算子	44
文字演算子	44
日付演算子	45
リレーショナル演算子	45
論理演算子	46
関数	46
文字列を返す関数	47
数字を返す関数	48
日付を返す関数	49
演算子の優先順位	49
ODBC カタログ関数	50
JDBC メタデータ関数	50
予約 SQL キーワード	50

## 第 8 章

### リファレンス情報

FileMaker フィールドタイプの ODBC データタイプへのマッピング	53
FileMaker フィールドの JDBC データタイプへのマッピング	53
ODBC および JDBC エラーメッセージ	54
ODBC エラーメッセージ	54
JDBC エラーメッセージ	54

索引	55
----	----

# 第 1 章

## 概要

このガイドでは、FileMaker® ソフトウェアを ODBC クライアントアプリケーションとして、また ODBC および JDBC アプリケーション用のデータソースとして使用する方法について説明します。

FileMaker ソフトウェアで ODBC と JDBC を使用する方法の概要については、次の表を参照してください。

操作	方法	製品	参照先
<ul style="list-style-type: none"><li>■ ODBC クライアントアプリケーションとしての FileMaker ソフトウェアの使用</li><li>■ 外部 SQL データソースに保存された ODBC データへのアクセス</li></ul>	<ol style="list-style-type: none"><li>1. リレーションシップグラフを介した対話</li><li>2. 一度の静的 ODBC インポート、または [ファイル] メニュー &gt; [開く ...] を選択。また、[レコードのインポート] スクリプトステップおよび [SQL を実行] スクリプトステップ</li></ol>	<ul style="list-style-type: none"><li>■ FileMaker Pro</li><li>■ FileMaker Pro Advanced</li><li>■ FileMaker Server</li><li>■ FileMaker Server Advanced</li></ul>	<ul style="list-style-type: none"><li>■ このガイドの第 2 章。</li><li>■ FileMaker Pro ヘルプ</li></ul>
<ul style="list-style-type: none"><li>■ データソースとしての FileMaker データベースの使用</li><li>■ FileMaker Pro データのサードパーティ ODBC クライアントアプリケーションとの共有</li></ul>	<ol style="list-style-type: none"><li>1. SQL クエリー</li><li>2. ODBC および JDBC</li></ol>	<ul style="list-style-type: none"><li>■ FileMaker Pro</li><li>■ FileMaker Pro Advanced</li><li>■ FileMaker Server Advanced のみ</li></ul>	<p>このガイドの第 3 章から第 8 章。</p>

## このガイドについて

- 旧バージョンの FileMaker Pro での ODBC および JDBC の使用については、[www.filemaker.co.jp](http://www.filemaker.co.jp) を参照してください。
- このガイドは、ユーザが ODBC および JDBC の基本操作と SQL クエリーの構築に精通していることを想定しています。これらのトピックの詳細については、他社の書籍を参照してください。
- このガイドでは、FileMaker Pro Advanced の特定の機能を説明する場合以外は、「FileMaker Pro」を、FileMaker Pro と FileMaker Pro Advanced の両方を指すものとして使用しています。

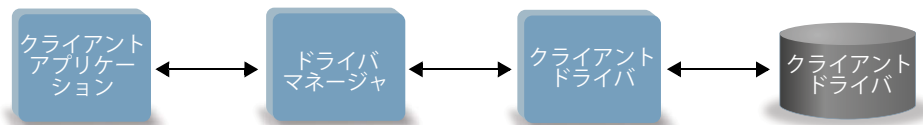
✎ FileMaker に関するドキュメントについては、[www.filemaker.co.jp](http://www.filemaker.co.jp) からダウンロードすることができます。このドキュメントの最新版も、Web サイトから入手できます。

## ODBC および JDBC について

ODBC および JDBC は、API（アプリケーションプログラミングインターフェース）です。ODBC は C 言語で記述されたアプリケーション用の API で、JDBC は Java 言語用に記述された同様の API です。これらの API は、FileMaker Pro および FileMaker Server Advanced を含む各種のデータソースとデータベースサービスに接続するための共通言語をクライアントアプリケーションに提供します。

ODBC および JDBC をサポートするすべてのアプリケーションは、SQL（構造化照会言語）ステートメントの基本的なサブセットを認識します。SQL を使用することで、他のアプリケーション（スプレッドシート、ワードプロセッサ、レポートツールなど）を使用して、データを表示、分析、および修正することができます。

クライアントアプリケーションはデータソースと通信するために、ODBC または JDBC API を使用してクライアントドライバを指定するドライバマネージャと通信します。



FileMaker ソフトウェアはクライアントアプリケーション、またはデータソースとして動作します。

## ODBC クライアントアプリケーションとしての FileMaker ソフトウェアの使用

ODBC クライアントアプリケーションとして、FileMaker ソフトウェアは外部 SQL データソースにアクセスできます。FileMaker ソフトウェアは、ODBC データソース用のクライアントドライバを使用して外部 SQL データソースに接続し、ODBC データをインポートするか、リレーションシップグラフで ODBC テーブルと連携します。

### ODBC データのインポート

ODBC データは次のいずれかの方法でインポートできます。

- [ファイル] メニューから、ODBC データソースを指定し、FileMaker Pro SQL クエリービルダーのダイアログボックスに SQL ステートメントを入力する
- [レコードのインポート] スクリプトステップ、または [SQL を実行] スクリプトステップを使用する FileMaker スクリプトを作成する

いずれの方法でも、ユーザ自身が SQL ステートメントを入力する必要があるため、サポートされる SQL ステートメント、および ODBC データソース用の構文について知る必要があります。ユーザ自身が SQL ステートメントを記述するため、任意の ODBC データソースから ODBC データをインポートすることができます。

### リレーションシップグラフへの ODBC テーブルの追加

リレーションシップグラフに ODBC テーブルを追加する場合、現在のアクティブな FileMaker データベースファイルのデータを操作する場合とよく似た方法で、外部 SQL データソースと接続、連携することができます。たとえば、次のようなことが可能です。

- リレーションシップグラフに ODBC データソース用のテーブルを作成する
- 非保存計算を実行する ODBC テーブルに補助フィールドを追加する、または ODBC テーブルでデータを集計する
- 外部データを対話的に追加、変更、および削除する
- ODBC テーブルで FileMaker テーブルおよびフィールド (「列」とも呼ばれる) のフィールド間にリレーションシップを作成する

FileMaker Pro は、リレーションシップグラフに追加された ODBC テーブルとの通信に使用する SQL ステートメントを生成するため、リレーションシップグラフで FileMaker Pro がサポートする特定の Oracle、SQL Server、および MySQL データソースに限定されます。

**メモ** FileMaker Pro を使用して、外部 ODBC データソースのスキーマを変更することはできません。

第 2 章「外部 SQL データソースへのアクセス」には、FileMaker ソフトウェアを ODBC クライアントアプリケーションとして使用する方法が説明されています。

## データソースとしての FileMaker データベースの使用

データソースとして、FileMaker データは、ODBC および JDBC 準拠のアプリケーションと共有されています。アプリケーションは、FileMaker クライアントドライバを使用して FileMaker データソースに接続し、ODBC または JDBC を使用して SQL クエリーを構築および実行し、FileMaker データベースソリューションから取得したデータを処理します。

### ホストされる FileMaker Pro データベースへのアクセス

FileMaker Server Advanced または FileMaker Pro を使用して、FileMaker データベースファイルをデータソースとしてホストし、ODBC および JDBC を使用して他のアプリケーションとデータを共有できます。次の表に、FileMaker の各製品で実行できる内容について説明します。

FileMaker 製品	操作
FileMaker Server Advanced	最大 50 の接続、およびローカルアクセス (同じコンピュータ) とリモートアクセス (Web サーバーなどのミドルウェアと、デスクトップ生産性向上アプリケーションからのリモートクライアントアクセスの両方) をサポート。
FileMaker Pro	最大 9 の接続、およびローカルアクセス (同じコンピュータ) のみサポート。

FileMaker データベースソリューションで複数の FileMaker データベースファイルを使用する場合、データベースファイルはすべて同じコンピュータ上にある必要があります。



他のアプリケーションとデータを共有する場合に必要な ODBC と JDBC プラグインコンポーネントは、FileMaker Server Advanced および FileMaker Pro と同時にインストールされます。

ホストされる FileMaker データベースファイルにアクセスする場合、対応する ODBC または ODBC クライアントドライバをインストールする必要があります。クライアントドライバは、他社のアプリケーションがインストールされているコンピュータにインストールします。

このガイドでは、ODBC および JDBC クライアントドライバを FileMaker Pro や FileMaker Server Advanced とともに使用する場合に、ODBC（Open Database Connectivity）、JDBC（Java Database Connectivity）、および SQL（構造化照会言語）の業界標準がどのようにサポートされるかについて説明します。

- ODBC を使用した FileMaker データソースへのアクセスに必要なドライバファイルのインストール方法については、第 3 章「FileMaker ODBC クライアントドライバのインストール」で説明しています。
- 別のアプリケーションから FileMaker データソースに接続する FileMaker ODBC クライアントドライバを使用する方法については、第 4 章「ODBC を使用した FileMaker データの共有」で説明しています。
- JDBC を使用した FileMaker データソースへのアクセスに必要なドライバファイルのインストール方法については、第 5 章「FileMaker JDBC クライアントドライバ」で説明しています。
- FileMaker JDBC クライアントドライバを FileMaker データソースに接続する Java アプリケーションまたはアプレットで使用方法については、第 6 章「JDBC を使用した FileMaker データの共有」で説明しています。
- ODBC および JDBC クライアントドライバを FileMaker Pro や FileMaker Server Advanced で使用する場合にサポートされる SQL ステートメントについては、第 7 章「サポートされている標準」で説明しています。

**重要** ODBC/JDBC 共有がすでにオンになっていた場合に無効にすると、FileMaker Server Advanced または FileMaker Pro でホストされているデータソースはただちに利用できなくなります。データソースを利用できなくなったことについて、データベース管理者が ODBC および JDBC クライアントアプリケーションに警告することはできません（管理者は、FileMaker データベースファイルのクライアントとのみ通信できます）。エラーが報告されないため、データソースが利用できなくなったことにより処理が完了できないことをクライアントアプリケーションからユーザに通知する必要があります。利用できない FileMaker データベースファイルにクライアントアプリケーションが接続しようとする、接続に失敗したことを説明するメッセージが表示されます。

## サードパーティ製ツールに関する制限

Microsoft Access: Microsoft Access を使用して FileMaker データソースのデータを表示する場合、集計フィールドからのデータは使用しないでください。集計フィールドのデータは、Microsoft Access で編集することはできないため、Microsoft Access で表示されるデータ値は正確でない可能性があります。

## ネットワークを使用する場合の条件

FileMaker Server Advanced を使用して、ネットワーク上で FileMaker データベースファイルをデータソースとしてホストする場合、TCP/IP ネットワークが必要です。FileMaker Pro では、ローカルアクセス（同じコンピュータ）のみがサポートされます。

## 旧バージョンのファイルの更新

### 最新のドライバのインストール

FileMaker Pro または FileMaker Server Advanced の前のバージョンからのドライバをインストールしている場合は、バージョン 11 用のドライバをインストールする必要があります。

FileMaker バージョン 11 用のドライバは、FileMaker Pro または FileMaker Server Advanced の前のバージョンと互換性がありません。

ドライバのインストールの詳細については、第 3 章「FileMaker ODBC クライアントドライバのインストール」および第 5 章「FileMaker JDBC クライアントドライバ」を参照してください。

**メモ** データソースとしてアクセスする個々の FileMaker データベースファイルに対して、データソース名 (DSN) を作成する必要があります。1 つの DSN を使用してアクセスするように設定し、複数の FileMaker データベースファイル間でテーブルを分散できるようにしていた場合は、それらのテーブルを 1 つのデータベースファイルに統合するか、複数の DSN を作成する必要があります。

## 第 2 章

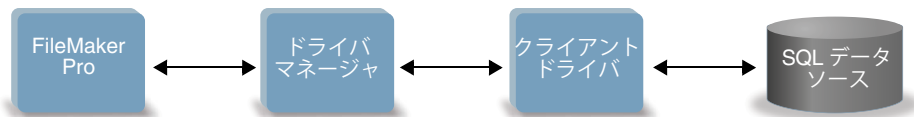
# 外部 SQL データソースへのアクセス

ODBC クライアントアプリケーションとして、FileMaker ソフトウェアは外部 SQL データソースにアクセスできます。FileMaker ソフトウェアは、ODBC データソース用のクライアントドライバを使用して外部 SQL データソースに接続し、ODBC データをインポートするか、リレーションシップグラフで ODBC テーブルと連携します。

ODBC データをインポートしているか、リレーションシップグラフで ODBC テーブルを操作しているかにかかわらず、使用している ODBC データソース用のドライバを設定する必要があります。たとえば、Oracle データベースからレコードにアクセスする場合は、Oracle クライアントドライバを設定します。

### ODBC データのインポート

ODBC データをインポートする場合は、クライアントコンピュータ上で設定された外部 SQL データソース用の ODBC クライアントドライバが必要です。



ODBC クライアントドライバを設定すると、レコードの操作、既存の FileMaker Pro データベースファイルへのレコードのインポート、または ODBC データソース（Oracle または Microsoft Access データベースなど）からの新しい FileMaker Pro データベースファイルの作成が可能になります。

まず、インポート元にするデータソースにアクセスします。次に、そのデータソースからインポートするレコードに対するクエリを構築します。最後に、既存のファイルにデータをインポートする場合は、データソースから FileMaker Pro データベースファイルにフィールドをマップします。

[ファイル]メニュー、[レコードのインポート]スクリプトステップ、または[SQL を実行]スクリプトステップを使用して、ODBC データソースにアクセスできます。

ODBC データをインポートするには、次の一般的な手順に従います。

- アクセスする外部データソース用の特定の ODBC ドライバをインストールして設定する。
- 現在の FileMaker Pro ファイルをホストするコンピュータで、アクセスする各 ODBC データソースに対してシステムデータソース名（DSN）を定義する。
- アクセスする ODBC データソースに関する追加的な考慮事項を決定する（ユーザにユーザ名とパスワードの入力を求めるメッセージを表示するかなど）。

- FileMaker Pro では、次のいずれかを行います。
  - 既存の FileMaker Pro ファイルにインポートするには、[ファイル]メニュー>[レコードのインポート]>[ODBC データソース...]を選択します。
  - データソースレコードから FileMaker Pro ファイルを作成するには、[ファイル]メニュー>[開く...]を選択します。[開く...]ダイアログボックスの、[ファイルの種類:] (Windows) または [表示:] (Mac OS) で [ODBC データソース...]を選択します。

データソースを選択して、ユーザ名とパスワード（ある場合）を入力し、[OK] をクリックして、FileMaker Pro SQL クエリービルダーのダイアログボックスを開きます。

FileMaker Pro SQL クエリービルダーのダイアログボックスを使用してクエリーを構築できます。インポート元にするテーブルを選択し、SQL クエリーで使用する特定の列を選択します。検索条件を構築するには [WHERE] タブ、ソート順を指定するには、[ORDER BY] タブを使用します。

SQL Query ビルダーのダイアログボックスに、SQL ステートメントを直接入力することもできます。

クエリーはただちに実行することも、[レコードのインポート] スクリプトステップ、または [SQL を実行] スクリプトステップを使用して、FileMaker スクリプトの一部として実行することもできます。

**×モ** ODBC インポート、[SQL を実行] スクリプトステップ、および外部 SQL データソースは、FileMaker Pro Advanced で作成されたランタイムソリューションではサポートされていません。

データのインポート、SQL クエリービルダーのダイアログボックスの使用、および FileMaker スクリプトの作成の詳細については、FileMaker Pro ヘルプを参照してください。

## ODBC によるデータソースの操作のための SQL の実行

ODBC による FileMaker Pro データベースファイルへのデータのインポートだけでなく、[SQL を実行] スクリプトステップを介して、SQL ステートメントを使用して、データソースを操作することもできます。INSERT、UPDATE、および DELETE など、データソースによってサポートされる任意の SQL ステートメントを使用できます。

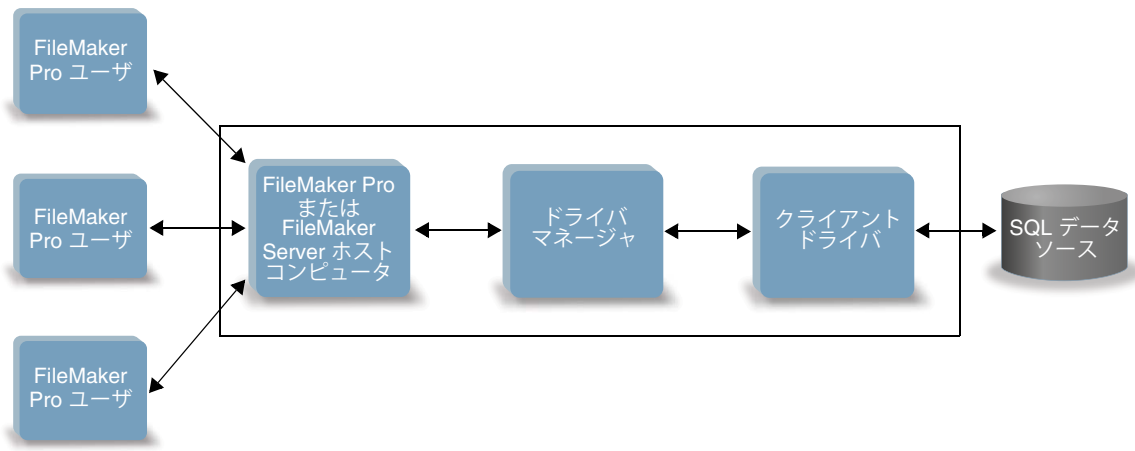
単に FileMaker Pro データベースファイルにデータをインポートする以上の操作に SQL ステートメントを使用することもできます。たとえば、FileMaker Pro データベースファイルからの情報を使用して、SQL Server のデータベーステーブルにレコードを追加する SQL ステートメントを実行することもできます。

[SQL を実行] スクリプトステップを使用する FileMaker スクリプトの作成の詳細については、FileMaker Pro ヘルプを参照してください。

## リレーションシップグラフでの ODBC テーブルの操作

リレーションシップグラフに ODBC テーブルを追加する場合、現在のアクティブな FileMaker データベースファイルのデータを操作する場合とよく似た方法で、外部 SQL データソースと接続、連携することができます。

リレーションシップグラフで ODBC テーブルを含むソリューションのためのホストとして、FileMaker Pro または FileMaker Server を使用する場合は、ホストコンピュータ上で外部 SQL データソース用の ODBC クライアントドライバを設定します。



### FileMaker 11 でサポートされるデータソース

ODBC クライアントアプリケーションとして、FileMaker はリレーションシップグラフで、ODBC テーブルとして、次の外部 SQL データソースをサポートしています。

- Oracle 9i
- Oracle 10g
- Oracle 11g
- SQL Server 2000
- SQL Server 2005
- SQL Server 2008
- MySQL 5.0 Community Edition (無償版)
- MySQL 5.1 Community Edition (無償版)

サポートされているクライアントドライバについては、[www.filemaker.co.jp/support/technologies](http://www.filemaker.co.jp/support/technologies) を参照してください。

### リレーションシップグラフへの ODBC テーブルの追加

サポートされている ODBC データソースのデータにアクセスするために FileMaker Pro データベースを設定するには、次の操作を行います。

- アクセスする外部データソース用の特定の ODBC ドライバをインストールして設定する。
- 現在の FileMaker Pro ファイルをホストするコンピュータで、アクセスする各 ODBC データソースに対してシステムデータソース名 (DSN) を定義する。
- アクセスする ODBC データソースに関する追加的な考慮事項を決定する (ユーザにユーザ名とパスワードの入力を求めるメッセージを表示するかなど)。
- ODBC データソースから、現在の FileMaker Pro ファイルのリレーションシップグラフに 1 つ以上のテーブルを追加する。
- 外部データを表示するには、FileMaker Pro ファイルのレイアウトにフィールドを追加する。
- 外部 ODBC データソースで保存されたデータに基づく計算および集計結果を表示するには、外部テーブルおよびレイアウトにオプションで補助フィールドを追加する。

ODBC クライアントドライバの設定、ODBC データソースへの接続、ODBC データソースの編集、およびリレーションシップグラフでの ODBC テーブルの設定の詳細なステップと追加情報については、FileMaker Pro ヘルプを参照してください。



## 第 3 章

# FileMaker ODBC クライアントドライバのインストール

この説明は、ODBC（Open Database Connectivity）を使用して他社のアプリケーションやカスタムアプリケーションからデータソースとしての FileMaker にアクセスするために必要な ODBC クライアントドライバをインストールする場合に役立ちます。ODBC クライアントドライバは、FileMaker インストールディスク、または電子ダウンロードの「xDBC」フォルダから個別のインストールとして利用できます。

最新バージョンのクライアントドライバは、

[www.filemaker.co.jp/support/technologies](http://www.filemaker.co.jp/support/technologies)

FileMaker Server Advanced を使用して FileMaker データベースファイルをホストする場合は、リモートユーザがクライアントドライバを利用できるようにします。

必要なクライアントドライバをインストールしたら、ドライバを設定して FileMaker データソースにアクセスしたり、SQL（構造化照会言語）クエリーを構築してデータを操作できます。

## 必要なハードウェアとソフトウェア

ODBC クライアントドライバをインストールして使用するには、次のハードウェアとソフトウェアが最低限必要です。

### ODBC クライアントドライバの必要条件（Windows）

- Pentium III 700MHz 以上
- Windows XP の場合は最低 256 MB の RAM、推奨 2 GB。Windows Vista および Windows 7 の場合は、最低 1 GB の RAM
- Microsoft Data Access Components（MDAC）バージョン 2.8 SP1
- Windows MDAC 6.0 for Vista

### ODBC クライアントドライバの必要条件（Mac OS）

- Apple G4、G5、または Intel プロセッサを搭載した Mac OS コンピュータ
- 最低 1 GB の RAM、推奨 2 GB
- Mac OS X 10.5.7 または 10.6（このソフトウェアは、FileMaker によって確認された以降のバージョンでも動作します）

### ネットワークを使用する場合の必要条件

別のコンピュータ上でホストされた FileMaker データソースにアクセスする場合は、TCP/IP によるネットワークアクセスが必要です。

## ODBC クライアントドライバのインストール（Windows）

**メモ** ODBC クライアントドライバを 32 ビット Windows オペレーティングシステムにインストールするには、MDAC 2.8 SP1 が必要です。MDAC は、[www.microsoft.com](http://www.microsoft.com) から入手できます。

ODBC クライアントドライバをインストールするには：

1. 次のいずれかを行います。
  - ソフトウェアを電子的に受け取っている場合は、インストールアイコン（.exe ファイル）をダブルクリックします。
  - インストールディスクがある場合は、ディスクをドライブに挿入します。
2. FileMaker Pro または FileMaker Server ウィンドウで「xDBC」フォルダをダブルクリックします。

3. 「xDBC」フォルダで「ODBC Client Driver Installer」フォルダをダブルクリックします。
4. 「ODBC Client Driver Installer」フォルダで [FMODBC\_Installer\_Win32.msi] ファイルをダブルクリックします。  
FileMaker ODBC Driver Setup Wizard が開きます。
5. 画面の指示に従って、ODBC クライアントドライバをインストールします。
6. インストールが終了したら、[ 閉じる ] をクリックします。  
デフォルトでは、ODBC クライアントドライバは次のフォルダにインストールされます。  
c:\¥ windows ¥ system32  
これで、FileMaker データソースにアクセスするための設定を行う際に、ODBC クライアントドライバ [FileMaker ODBC] を利用できるようになります。

## クライアントドライバの設定 (Windows)

クライアントアプリケーションを使用して FileMaker データソースにアクセスする前に、データソースに対してクライアントドライバを設定する必要があります。これらの設定で、使用するクライアントドライバ、データソースの場所、および接続する方法の詳細を識別します。

**重要** FileMaker クライアントドライバを使用する場合は、ポートとして「2399」を予約する必要があります。

ODBC クライアントドライバを設定するには、次の操作を行います。

1. Windows コントロールパネルで、[ 管理ツール ] > [ データソース (ODBC) ] を選択します。
  - Windows XP では、管理ツールはパフォーマンスとメンテナンスカテゴリに表示されます。
  - Windows Vista および Windows 7 では、管理ツールはシステムとメンテナンス、またはシステムとセキュリティカテゴリに表示されます。[ODBC データソースアドミニストレータ] が開きます。
2. [ システム DSN ] または [ ユーザ DSN ] タブを選択します。
3. [ 追加 ... ] をクリックします。  
[ データソースの新規作成 ] ダイアログボックスが開きます。
4. [FileMaker ODBC] を選択し、[ 完了 ] をクリックします。  
[FileMaker DSN の構成] ダイアログボックスが開きます。
5. [ 次へ ] をクリックします。
6. [ 名前 ] に、FileMaker データソースにアクセスする他のユーザにも分かりやすい名前を入力します。[ 説明 ] に、FileMaker データソースのオプションの説明を入力します。[ 次へ ] をクリックします。
7. [ ホスト ] では、次の設定を行います。
  - FileMaker Pro によってローカルコンピュータ上でホストされているデータベースファイルに接続する場合は、「localhost」または IP アドレス「127.0.0.1」を入力します。
  - FileMaker Server Advanced によってネットワーク上でホストされているデータベースファイルに接続する場合は、FileMaker Server の IP アドレスを入力します。ホストアプリケーションで ODBC/JDBC を経由した共有を有効にしている場合は、[ ホストに接続して使用可能なデータベースの名前を取得 ] を選択できます。[ 次へ ] をクリックします。  
そうでない場合は、[ 完了 ] をクリックしてデータソース情報を保存します。



8. [データベース]には、使用可能なデータベースのリストからデータベースを選択するか、データソースとして使用する FileMaker データベースファイルのファイル名を入力します。

**メモ** FileMaker Server Advanced でホストされているデータベースファイルについては、データベースのリストが [ファイル表示フィルタ] 設定に基づいてフィルタされている場合があります。詳細については、FileMaker Server ヘルプを参照してください。

英語以外のテキストの特殊な処理が必要な場合は [エンコード設定] をクリックします。[エンコード設定オプション] ダイアログボックスが開きます。

- 言語設定を自動検出するには、[アプリケーション用の言語設定を自動検出] オプションを選択します。
- 言語設定を指定するには、[アプリケーション用の言語設定を自動検出] オプションの選択を解除して、使用するシステム設定を選択します。

実行に時間のかかるクエリーのためのログファイルを作成するには、[実行に時間のかかるクエリーをログファイルに保存] オプションを選択して、ログファイルの名前を入力します。

[完了] をクリックしてデータソース情報を保存します。

9. FileMaker DSN に関する情報を確認します。

- [テスト] をクリックして、FileMaker データソースにアクセスするために ODBC クライアントドライバが正しく設定されていることを確認します。

エラーメッセージが返される場合は、接続情報を修正できます。

FileMaker データベースファイルがホストされ利用可能か、指定された FileMaker アカウントで ODBC/JDBC によるアクセス拡張アクセス権が割り当てられたアクセス権セットが使用されているか、およびホストアプリケーション (FileMaker Pro または FileMaker Server Advanced) が ODBC/JDBC で共有できるように設定されていることも確認する必要があります。

- [完了] をクリックしてデータソース情報を保存します。

## ODBC クライアントドライバのインストール (Mac OS)

ODBC クライアントドライバをインストールするには：

1. 次のいずれかを行います。

- ソフトウェアを電子的に受け取っている場合は、ディスクイメージアイコン (.dmg ファイル) をダブルクリックします。
- インストールディスクがある場合は、ディスクをドライブに挿入します。

2. FileMaker Pro または FileMaker Server ウィンドウで「xDBC」フォルダをダブルクリックします。

3. 「xDBC」フォルダで「ODBC Client Driver Installer」フォルダをダブルクリックします。

4. 「ODBC Client Driver Installer」フォルダで [FileMaker ODBC.mpkg] をダブルクリックします。

FileMaker ODBC Driver Installer が開きます。

5. 画面の指示に従って、ODBC クライアントドライバをインストールします。

6. インストールが終了したら、[閉じる] をクリックします。

ODBC クライアントドライバは次のフォルダにインストールされます。

/ライブラリ /ODBC

**メモ** ODBC クライアントドライバのインストールフォルダを変更することはできません。

これで、FileMaker データソースにアクセスするための設定を行う際に、ODBC クライアントドライバ [FileMaker ODBC] を利用できるようになります。

## クライアントドライバの設定 (Mac OS)

クライアントアプリケーションを使用して FileMaker データソースにアクセスする前に、データソースに対してクライアントドライバを設定する必要があります。これらの設定で、使用するクライアントドライバ、データソースの場所、および接続する方法の詳細を識別します。

ここでは、[www.odbcmanager.net](http://www.odbcmanager.net) から利用可能な FileMaker によってサポートされないフリーウェア製品である Actual Technologies の ODBC Manager がインストールされている場合を例に説明します。

Apple の ODBC Administrator Tool for Mac OS X が使用できる場合もあります。Mac OS X バージョン 10.5 の場合は、オペレーティングシステムの一部として ODBC Administrator がインストールされています。Mac OS X バージョン 10.6 の場合、ODBC Administrator は、[www.apple.com/support](http://www.apple.com/support) で利用できます。

**重要** FileMaker クライアントドライバを使用する場合は、ポートとして「2399」を予約する必要があります。

ODBC クライアントドライバを設定するには、次の操作を行います。

1. ODBC Manager ユーティリティを起動します。(ODBC Manager は、アプリケーションフォルダのユーティリティフォルダにインストールされています。)
2. [システム DSN] または [ユーザ DSN] タブを選択して、[追加 ...] をクリックします。  
[ドライバを選択] ダイアログボックスが開きます。
3. [FileMaker ODBC] を選択し、[OK] をクリックします。  
[FileMaker DSN の構成] ダイアログボックスが開きます。
4. [続行] をクリックします。
5. [名前] に、FileMaker データソースにアクセスする他のユーザにも分かりやすい名前を入力します。[説明] に、FileMaker データソースのオプションの説明を入力します。[続行] をクリックします。
6. [ホスト] では、次の設定を行います。
  - FileMaker Pro によってローカルコンピュータ上でホストされているデータベースファイルに接続する場合は、「localhost」または IP アドレス「127.0.0.1」を入力します。
  - FileMaker Server Advanced によってネットワーク上でホストされているデータベースファイルに接続する場合は、FileMaker Server の IP アドレスを入力します。

ホストアプリケーションで ODBC/JDBC を経由した共有を有効にしている場合は、[ホストに接続して使用可能なデータベースの名前を取得] を選択できます。[続行] をクリックします。

そうでない場合は、[完了] をクリックしてデータソース情報を保存します。

7. [データベース] には、使用可能なデータベースのリストからデータベースを選択するか、データソースとして使用する FileMaker データベースファイルのファイル名を入力します。

**メモ** FileMaker Server Advanced でホストされているデータベースファイルについては、データベースのリストが [ファイル表示フィルタ] 設定に基づいてフィルタされている場合があります。詳細については、FileMaker Server ヘルプを参照してください。

英語以外のテキストの特殊な処理が必要な場合は [エンコード設定] をクリックします。[エンコード設定オプション] ダイアログボックスが開きます。

- 言語設定を自動検出するには、[アプリケーション用の言語設定を自動検出] オプションを選択します。
- 言語設定を指定するには、[アプリケーション用の言語設定を自動検出] オプションの選択を解除して、使用するシステム設定を選択します。

実行に時間のかかるクエリーのためのログファイルを作成するには、[実行に時間のかかるクエリーをログファイルに保存] オプションを選択して、ログファイルの名前を入力します。

[完了] をクリックしてデータソース情報を保存します。

## 8. FileMaker DSN に関する情報を確認します。

- [ テスト ] をクリックして、FileMaker データソースにアクセスするために ODBC クライアントドライバが正しく設定されていることを確認します。  
エラーメッセージが返される場合は、接続情報を修正できます。  
FileMaker データベースファイルがホストされ利用可能か、指定された FileMaker アカウントで ODBC/JDBC によるアクセス拡張アクセス権が割り当てられたアクセス権セットが使用されているか、およびホストアプリケーション（FileMaker Pro または FileMaker Server Advanced）が ODBC/JDBC で共有できるように設定されていることも確認する必要があります。
- [ 完了 ] をクリックしてデータソース情報を保存します。

## この後の作業を開始するにあたって

クライアントドライバをインストールして設定したら、SQL クエリーを構築して実行し、FileMaker データソースにアクセスできます。

クライアントアプリケーションでは、ODBC によるデータソースへのアクセスに対して異なる用語が使用されている場合があります。多くのアプリケーションには、[ 外部データの取得 ] や [ SQL クエリー ] などの名前のメニュー項目があります。詳細については、アプリケーションに付属のマニュアルまたはヘルプを参照してください。

ODBC データソースのとしての FileMaker の使用の詳細については、第 4 章「ODBC を使用した FileMaker データの共有」を参照してください。



## 第 4 章

# ODBC を使用した FileMaker データの共有

ODBC クライアントドライバは、他のアプリケーションから FileMaker データソースに接続する場合に使用します。この ODBC クライアントドライバを使用するアプリケーションは、FileMaker データベースファイル内のデータに直接アクセスすることができます。

FileMaker ODBC クライアントドライバは、「FileMaker ODBC」です。

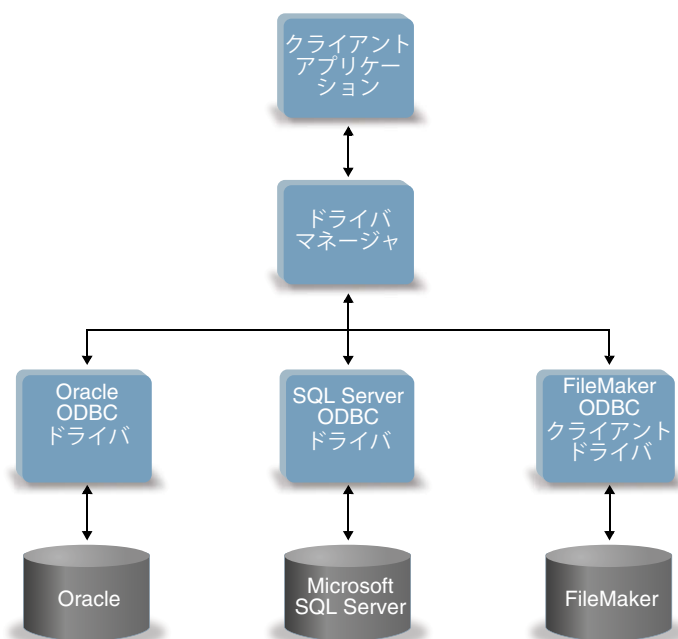
**✕** FileMaker Pro を ODBC クライアントアプリケーションとして使用し、SQL を使用して ODBC によって別のデータソースからレコードを操作することもできます。ODBC を使用した外部 SQL データソースへのアクセスの詳細については、第 2 章「外部 SQL データソースへのアクセス」を参照してください。

## ODBC について

ODBC は、アプリケーションがさまざまなデータベース管理システムのデータにアクセスできるようにする API です。ODBC は、クライアントアプリケーションに、データソースとデータベースサービスに接続するための共通言語を提供します。

ODBC をサポートするすべてのアプリケーションは、SQL（構造化照会言語）ステートメントの基本的なサブセットを認識します。SQL を操作すると、他のアプリケーション（スプレッドシート、ワードプロセッサ、レポートツールなど）を使用して、FileMaker のデータを表示、分析、および修正することができます。ODBC クライアントドライバでサポートされている SQL ステートメント、関数、および式については、第 7 章「サポートされている標準」を参照してください。

ODBC クライアントドライバを使用することで、アプリケーションは FileMaker データベースファイルと直接通信できます。SQL ステートメントがデータベースファイルの FileMaker ホストに配信され、そのステートメントの結果がユーザに返されます。FileMaker Server Advanced を使用して FileMaker データベースファイルをデータソースとしてホストする場合、クライアントアプリケーションをユーザのコンピュータ（クライアントコンピュータ）上に配置したまま、ネットワークに接続された別のコンピュータ（サーバーコンピュータ）上にデータベースファイルを配置することができます。これは、クライアント/サーバー型構成と呼ばれます。



## ODBC クライアントドライバの使用

ODBC クライアントドライバは、ODBC に準拠する任意のアプリケーションで使用できます。FileMaker データベースファイルをデータベースソースとして共有すると、次の処理を行うことができます。

- Microsoft Word で作成された手紙に受取人の住所、氏名の差込印刷を行う
- Microsoft Excel でグラフを作成する
- FileMaker のデータを Microsoft SQL Server などの DBMS に移動する
- クエリーまたはレポートのツールを使用して、FileMaker のデータをさらに分析し、チャートの作成、特別なクエリーの構築、および詳細な分析を実行する
- FileMaker Pro と情報を共有する Microsoft Visual Basic のアプリケーションを作成する

FileMaker データベースファイルをデータソースとして共有するには、FileMaker Pro を使用して、データベースファイルへのアクセスを必要とするアカウントを定義します。続いて、[ODBC/JDBC によるアクセス] 拡張アクセス権を含むアクセス権セットをアカウントに割り当てることによって、データベースファイルへのアクセスを制御します。最後に、FileMaker Server Advanced または FileMaker Pro ホストアプリケーションが ODBC/JDBC によってデータを共有できるようにします。詳細については、FileMaker Pro または FileMaker Server のヘルプを参照してください。

**重要** FileMaker ODBC クライアントドライバの前のバージョンには、FileMaker バージョン 11 との互換性がありません。FileMaker バージョン 11 のデータベースファイルに接続するには、新しい ODBC クライアントドライバをインストールして設定する必要があります。

**メモ** FileMaker xDBC Listener が現在実行されているか確認するためには、Mac OS の Activity Monitor、あるいは Windows のタスクマネージャを使用して FileMaker xDBC Listener プロセスのステータスを確認します。プロセスが開始した際、プロセス名に fmjdbc\_listener という名前が付けられ、同じ名前でもログにイベントが記録されます。FileMaker xDBC Listener プロセスと FileMaker Server プロセスとは区別されます。

## FileMaker データベースファイルへのアクセスの概要

ODBC に準拠したアプリケーションから、SQL クエリーを構築して FileMaker データベースファイルにアクセスできます。SQL クエリーを生成するコンピュータ上に、ODBC クライアントドライバをインストールしておく必要があります。

FileMaker データベースファイルにアクセスするには、次の操作を行います。

1. FileMaker Pro で、データベースファイルにアクセスするアカウントに割り当てたアクセス権セットを確認します。  
アクセスするアカウントには、[ODBC/JDBC によるアクセス] 拡張アクセス権が割り当てられたアクセス権セットを使用する必要があります。
2. FileMaker Server Advanced (FileMaker Server Admin Console から) または FileMaker Pro ホストアプリケーションが、ODBC/JDBC によってデータを共有できるようにします。  
FileMaker Server Admin Console の場合、[ODBC/JDBC] をクリックし、[ODBC/JDBC を有効にする] を選択します。  
FileMaker Pro の場合、[ファイル] メニュー > [共有設定] > [ODBC/JDBC...] を選択し、[ODBC/JDBC 共有] を [オン] にします。
3. アクセスする FileMaker データベースファイルがホストされていて利用可能であることを確認します。  
FileMaker データベースソリューションで複数の FileMaker データベースファイルを使用する場合、データベースファイルはすべて同じマシン上にある必要があります。
4. FileMaker データソースに接続します。
5. クライアントアプリケーションで SQL クエリーを構築して実行します。  
開いていてアクセスできるように設定されている各 FileMaker データベースファイルは、独立したデータソースです。データソースとしてアクセスする各 FileMaker データベースファイルに対して、DSN を作成します。  
各データベースには 1 つまたは複数のテーブルが含まれます。FileMaker のフィールドは、列として表示されます。英数字以外の文字も含む完全なフィールド名が、列名として表示されます。

## Windows アプリケーションからの FileMaker データベースファイルへのアクセス

### FileMaker DSN 用の ODBC クライアントドライバのプロパティの指定 (Windows)

データソースとしてアクセスする個々の FileMaker データベースファイルに対して、DSN を作成します。この DSN で、FileMaker ODBC クライアントドライバ、FileMaker ホストアプリケーションの場所、およびデータソースとしてアクセスする FileMaker データベースファイルを指定します。

新しい ODBC クライアントドライバを作成するには、16 ページの「クライアントドライバの設定 (Windows)」を参照してください。

既存の ODBC クライアントドライバを変更するには、次の操作を行います。

- Windows コントロールパネルで、[ 管理ツール ] > [ データソース (ODBC) ] を選択します。
  - Windows XP では、管理ツールはパフォーマンスとメンテナンスカテゴリに表示されます。
  - Windows Vista および Windows 7 では、管理ツールはシステムとメンテナンス、またはシステムとセキュリティカテゴリに表示されます。

[ODBC データソースアドミニストレータ] が開きます。
- [ システム DSN ] または [ ユーザ DSN ] タブを選択します (前に設定したときに使用したタブを選択します)。
- すでに設定済みの FileMaker データソースを選択します。  
[ 名前 ] には最初に入力したデータソース名が表示され、[ ドライバ ] には [FileMaker ODBC] が表示されます。
- [ 構成 ... ] をクリックします。  
[FileMaker DSN の構成] ダイアログボックスが表示されます。
- [ 名前 ] に、FileMaker データソースにアクセスする他のユーザにも分かりやすい名前を入力します。[ 説明 ] に、FileMaker データソースのオプションの説明を入力します。[ 次へ ] をクリックします。
- [ ホスト ] に、データソースの場所を入力します。  
FileMaker Pro によってローカルコンピュータ上でホストされている FileMaker データベースファイルに接続する場合は、“localhost” (または 127.0.0.1) を入力します。  
FileMaker Server Advanced によってネットワーク上でホストされている FileMaker データベースファイルに接続する場合は、FileMaker Server の IP アドレスを入力します。  
ホストアプリケーションで ODBC/JDBC を経由した共有を有効にしている場合は、[ ホストに接続して使用可能なデータベースの名前を取得 ] を選択します。
- [ データベース ] には、使用可能なデータベースのリストからデータベースを選択するか、データソースとして使用する FileMaker データベースファイルのファイル名を入力します。  
**メモ** FileMaker Server Advanced でホストされているデータベースファイルについては、データベースのリストが [ ファイル表示フィルタ ] 設定に基づいてフィルタされている場合があります。詳細については、FileMaker Server ヘルプを参照してください。  
英語以外のテキストの特殊な処理が必要な場合は [ エンコード設定 ] をクリックします。[ エンコード設定オプション ] ダイアログボックスが開きます。
  - 言語設定を自動検出するには、[ アプリケーション用の言語設定を自動検出 ] オプションを選択します。
  - 言語設定を指定するには、[ アプリケーション用の言語設定を自動検出 ] オプションの選択を解除して、使用するシステム設定を選択します。

実行に時間のかかるクエリーのためのログファイルを作成するには、[ 実行に時間のかかるクエリーをログファイルに保存 ] オプションを選択して、ログファイルの名前を入力します。
- [ 完了 ] をクリックしてデータソース情報を保存します。
- [ 完了 ] をクリックして、[FileMaker DSN の構成] ダイアログボックスを閉じます。



## ODBC によるアクセスの確認 (Windows)

FileMaker データソースにアクセスするために ODBC クライアントドライバが正しく設定されていることを確認するには、次の操作を行います。

1. Windows コントロールパネルで、[ 管理ツール ] > [ データソース (ODBC) ] を選択します。
  - Windows XP では、管理ツールはパフォーマンスとメンテナンスカテゴリに表示されます。
  - Windows Vista および Windows 7 では、管理ツールはシステムとメンテナンス、またはシステムとセキュリティカテゴリに表示されます。

[ODBC データソースアドミニストレータ] が開きます。
2. [ システム DSN ] または [ ユーザ DSN ] タブを選択します (前に設定したときに使用したタブを選択します)。
3. すでに設定済みの FileMaker データソースを選択します。  
[ 名前 ] には最初に入力したデータソース名が表示され、[ ドライバ ] には [FileMaker ODBC] が表示されます。
4. [ 構成 ... ] をクリックします。  
[FileMaker DSN の構成] ダイアログボックスが表示されます。
5. [ まとめ ] ページが表示されるまで、[ 次へ ] をクリックします。
6. [ テスト ] をクリックします。  
FileMaker のアカウント名とパスワードをそれぞれ [ データベースユーザ名 ] と [ データベースパスワード ] に入力するようメッセージが表示されます。

接続に問題がない場合は、「テストは正常に完了しました」というメッセージが表示されます。

接続が失敗した場合は、次の操作を行います。

- FileMaker データベースファイルがホストされていて利用可能であることを確認する
- 接続情報を更新または修正する
- 使用する FileMaker アカウントで、[ODBC/JDBC によるアクセス] 拡張アクセス権が割り当てられたアクセス権セットが使用されていることを確認する
- FileMaker Pro または FileMaker Server ホストアプリケーションが ODBC/JDBC でデータを共有できるように設定されていることを確認する

## Mac OS アプリケーションからの FileMaker データベースファイルへのアクセス

### FileMaker DSN 用の ODBC クライアントドライバのプロパティの指定 (Mac OS)

データソースとしてアクセスする個々の FileMaker データベースファイルに対して、DSN を作成します。この DSN で、FileMaker ODBC クライアントドライバ、FileMaker ホストアプリケーションの場所、およびデータソースとしてアクセスする FileMaker データベースファイルを指定します。

新しい ODBC クライアントドライバを作成するには、16 ページの「クライアントドライバの設定 (Windows)」を参照してください。

既存の ODBC クライアントドライバを変更するには、次の操作を行います。

1. ODBC Manager ユーティリティを起動します。(ODBC Manager は、アプリケーションフォルダのユーティリティフォルダにインストールされています。)
2. [ システム DSN ] または [ ユーザ DSN ] タブをクリックします。
3. すでに設定済みの FileMaker データソースを選択します。  
[ 名前 ] には最初に入力したデータソース名が表示され、[ ドライバ ] には [FileMaker ODBC] が表示されます。
4. [ 構成 ... ] をクリックします。  
[FileMaker DSN の構成] ダイアログボックスが開きます。



5. [ 続行 ] をクリックします。
6. [ 名前 ] に、FileMaker データソースにアクセスする他のユーザにも分かりやすい名前を入力します。  
[ 説明 ] はオプションです。
7. [ ホスト ] に、データソースの場所を入力します。  
FileMaker Pro によってローカルコンピュータ上でホストされている FileMaker データベースファイルに接続する場合は、“localhost”（または 127.0.0.1）を入力します。  
FileMaker Server Advanced によってネットワーク上でホストされている FileMaker データベースファイルに接続する場合は、FileMaker Server の IP アドレスを入力します。  
ホストアプリケーションで ODBC/JDBC を経由した共有を有効にしている場合は、[ ホストに接続して使用可能なデータベースの名前を取得 ] を選択します。
8. [ データベース ] には、使用可能なデータベースのリストからデータベースを選択するか、データソースとして使用する FileMaker データベースファイルのファイル名を入力します。  
**メモ** FileMaker Server Advanced でホストされているデータベースファイルについては、データベースのリストが [ ファイル表示フィルタ ] 設定に基づいてフィルタされている場合があります。詳細については、FileMaker Server ヘルプを参照してください。  
英語以外のテキストの特殊な処理が必要な場合は [ エンコード設定 ] をクリックします。[ エンコード設定オプション ] ダイアログボックスが開きます。
  - 言語設定を自動検出するには、[ アプリケーション用の言語設定を自動検出 ] オプションを選択します。
  - 言語設定を指定するには、[ アプリケーション用の言語設定を自動検出 ] オプションの選択を解除して、使用するシステム設定を選択します。
 実行に時間のかかるクエリーのためのログファイルを作成するには、[ 実行に時間のかかるクエリーをログファイルに保存 ] オプションを選択して、ログファイルの名前を入力します。
9. [ 完了 ] をクリックしてデータソース情報を保存します。
10. [ 完了 ] をクリックして、[ FileMaker DSN の構成 ] ダイアログボックスを閉じます。

## ODBC によるアクセスの確認 (Mac OS)

FileMaker データソースにアクセスするために ODBC クライアントドライバが正しく設定されていることを確認するには、次の操作を行います。

1. ODBC Manager ユーティリティを起動します。（ODBC Manager は、アプリケーションフォルダのユーティリティフォルダにあります。）
2. [ システム DSN ] または [ ユーザ DSN ] タブを選択します（前に設定したときに使用したタブを選択します）。
3. すでに設定済みの FileMaker データソースを選択します。  
[ 名前 ] には最初に入力したデータソース名が表示され、[ ドライバ ] には [ FileMaker ODBC ] が表示されます。
4. [ 構成 ... ] をクリックします。  
[ FileMaker DSN の構成 ] ダイアログボックスが開きます。
5. [ まとめ ] ページが表示されるまで、[ 続行 ] をクリックします。
6. [ テスト ] をクリックします。  
FileMaker のアカウント名とパスワードをそれぞれ [ データベースユーザ名 ] と [ データベースパスワード ] に入力するようメッセージが表示されます。

接続に問題がない場合は、「テストは正常に完了しました」というメッセージが表示されます。

接続が失敗した場合は、次の操作を行います。

- FileMaker データベースファイルがホストされていて利用可能であることを確認する
- 接続情報を更新または修正する
- 使用する FileMaker アカウントで、[ODBC/JDBC によるアクセス] 拡張アクセス権が割り当てられたアクセス権セットが使用されていることを確認する
- FileMaker Pro または FileMaker Server ホストアプリケーションが ODBC/JDBC でデータを共有できるように設定されていることを確認する

# 第 5 章

## FileMaker JDBC クライアントドライバ

この説明は、JDBC（Java Database Connectivity）を使用して他社のアプリケーションやカスタムアプリケーションからデータソースとしての FileMaker にアクセスするために必要なクライアントドライバをインストールする場合に役立ちます。クライアントドライバは、FileMaker DVD、または電子ダウンロードの「xDBC」フォルダから利用できます。最新バージョンのクライアントドライバは、次の URL でも利用可能です。

[www.filemaker.co.jp/support/technologies](http://www.filemaker.co.jp/support/technologies)

FileMaker Server Advanced を使用して FileMaker データベースファイルをホストする場合は、リモートユーザがクライアントドライバを利用できるようにします。

必要なクライアントドライバをインストールしたら、ドライバを設定して FileMaker データソースにアクセスしたり、SQL（構造化照会言語）クエリーを構築してデータを操作できます。

JDBC クライアントドライバは FileMaker ソフトウェアのドライバの部分で、他社のアプリケーションやカスタムアプリケーションから、JDBC データソースとしての FileMaker ファイルにアクセスできるようにします。

### 必要なソフトウェア

JDBC クライアントドライバをインストールして使用するには、JDK 1.4 以降が必要です。

実行されている Java のバージョンを確認するには、コマンドウインドウ（Windows）またはターミナルウインドウ（Mac OS）を開いて、“java -version”と入力します。

### ネットワークを使用する場合の必要条件

別のコンピュータ上でホストされた FileMaker データソースにアクセスする場合は、TCP/IP によるネットワークアクセスが必要です。

### JDBC クライアントドライバのインストール

ユーザは、JDBC クライアントドライバのインストール先として使用するフォルダに対し、書き込みアクセスが必要です。

JDBC クライアントドライバをインストールするには：

1. 次のいずれかを行います。

- Windows: ソフトウェアを電子的に受け取っている場合は、インストールアイコン（.exe ファイル）をダブルクリックします。
- Mac OS: ソフトウェアを電子的に受け取っている場合は、ディスクイメージアイコン（.dmg ファイル）をダブルクリックします。
- インストールディスクがある場合は、ディスクをドライブに挿入します。

2. FileMaker Pro または FileMaker Server ウインドウで「xDBC」フォルダをダブルクリックします。

3. 「xDBC」フォルダで「JDBC Client Driver Installer」フォルダをダブルクリックします。

4. 「fmjdbc.jar」ファイルをオペレーティングシステムの適切なフォルダにコピーします。

- Windows: 「fmjdbc.jar」ファイルを Java 実行ファイル（java.exe）が含まれるフォルダ、または Java アプリケーションの ClassPath に含まれる別のフォルダの場所にコピーします。
- Mac OS: 「fmjdbc.jar」ファイルを「/ライブラリ /Java/Extensions」フォルダ、または Java アプリケーションの ClassPath に含まれる別のフォルダの場所にコピーします。

これで、FileMaker データソースへのアクセスに、JDBC クライアントドライバを利用できるようになります。

## JDBC クライアントドライバの使用

JDBC ドライバマネージャを使用して Java アプリケーションやアプレットに JDBC クライアントドライバを登録し、アプリケーションやアプレット内から正しい JDBC URL を指定する必要があります。

**重要** FileMaker JDBC クライアントドライバには、ポート 2399 を予約する必要があります。ポート番号は常に 2399 です。JDBC 共有を別のポートに変更することはできません。

JDBC クライアントドライバの使用法の詳細については、第 6 章「JDBC を使用した FileMaker データの共有」を参照してください。

## 第 6 章

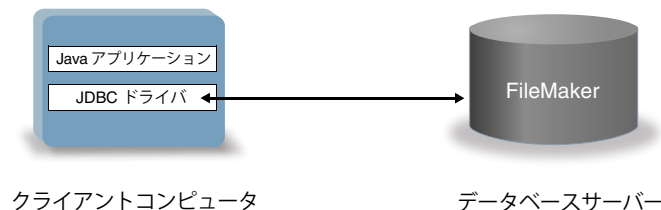
# JDBC を使用した FileMaker データの共有

Java プログラマは、JDBC クライアントドライバを RAD（Rapid Application Development）ツールとともに使用して、FileMaker データソースに接続する Java アプリケーションやアプレットを視覚的に作成できます。この JDBC クライアントドライバを使用する Java アプリケーションやアプレットは、FileMaker データベースファイル内のデータに直接アクセスすることができます。

### JDBC について

JDBC は、リレーショナルデータベースにアクセスするための標準的な言語である SQL ステートメントを実行するための Java API です。JDBC は Java の ODBC に相当する機能であるため「Java Database Connectivity」を表すと考えられていますが、名前であり、頭字語ではありません。JDBC は低レベルインターフェースです。つまり、JDBC は SQL コマンドを直接呼び出すために使用されます。また、高レベルインターフェースおよびツールのためのベースとして使用されるようにも設計されています。

JDBC クライアントドライバを使用することで、Java アプレットやアプリケーションは FileMaker データベースファイルと直接通信できます。SQL ステートメントがデータベースファイルの FileMaker ホストに配信され、そのステートメントの結果がユーザに返されます。FileMaker Server を使用してホストすると、Java アプレットまたはクライアントアプリケーションをユーザのコンピュータ（クライアントコンピュータ）上に配置したまま、データソースとして使用する FileMaker データベースファイルは、ネットワークに接続された別のコンピュータ（サーバーコンピュータ）上に配置することができます。これは、クライアント/サーバー型構成と呼ばれます。



### JDBC クライアントドライバの使用

Java アプリケーションやアプレットのコードをビルドする際に、任意の Java コンパイラや RAD ツールとともに JDBC クライアントドライバを使用して、データベースに接続することができます。Java アプリケーションやアプレットの作成後、アプリケーションやアプレットをデータベースと通信させるためには、JDBC クライアントドライバをファイルと同じ場所に置くか、コード内に含める必要があります。

JDBC クライアントドライバを使用するには、JDBC ドライバマネージャを使用して Java アプリケーションやアプレットにドライバを登録し、アプリケーションやアプレット内から正しい JDBC URL を指定する必要があります。データベースに接続するには、JDBC URL が必要です。

### JDBC クライアントドライバについて

JDBC クライアントドライバは、JDBC 3.0 仕様を部分的にサポートします。次の機能は FileMaker でサポートされません。

- セーブポイントサポート
- 自動生成キーの検索
- 呼び出し可能なステートメントオブジェクトに名前を渡す
- ホールダブルカーソルサポート
- Ref オブジェクトで参照されるオブジェクトの検索と更新
- CLOB、ARRAY、REF データ型を含む列の更新
- 論理データ型

- DATALINK データ型
- グループと型のマッピングの変換
- JDBC SPI とコネクタアーキテクチャの関係

詳細については、[www.filemaker.co.jp/support/technologies](http://www.filemaker.co.jp/support/technologies) を参照してください。

JDBC クライアントドライバは、Java Development Kit (JDK) 1.5 (Mac OS) および 1.6 (Windows) についてテストされています。これはタイプ 4 ドライバで、FileMaker によって使用されるネットワークプロトコルに JDBC 呼び出しを直接変換するネイティブプロトコルのピュア Java ドライバです。このタイプのドライバでは、自動インストール（たとえば、JDBC ドライバを使用するアプレットとともにそのドライバをダウンロードする）などすべての Java の機能を利用できます。

ドライバクラスおよびドライバのメインエントリポイントの名前は、次のとおりです。

`com.filemaker.jdbc.Driver`

**重要** 旧バージョンのファイルメーカーとともにリリースされた FileMaker JDBC ドライバは、JDBC クライアントドライバによって置き換えられます。以前に古いドライバを使用して FileMaker データソースにアクセスするように設定していた場合は、新しいドライバを使用および設定して、アクセスを再定義する必要があります。

**メモ** FileMaker xDBC Listener が現在実行されているか確認するためには、Mac OS の Activity Monitor、あるいは Windows のタスクマネージャを使用して FileMaker xDBC Listener プロセスのステータスを確認します。プロセスが開始した際、プロセス名に `fmjdbc_listener` という名前が付けられ、同じ名前でログにイベントが記録されます。

FileMaker xDBC Listener プロセスと FileMaker Server プロセスとは区別されます。

## JDBC URL を使用したデータベースへの接続

Java では、大部分のリソースは URL (Uniform Resource Locator) を通じてアクセスされます。JDBC URL を使用してデータベースを識別し、JDBC クライアントドライバがデータベースとの接続を認識および確立できるようにします。

JDBC URL は、コロンで区切られた次の 3 つの主要部分から成ります。

`jdbc:<サブプロトコル>:<サブネーム>`

JDBC URL の最初の部分は常に JDBC プロトコル (「`jdbc`」) です。<サブプロトコル> は、ドライバ名または複数のドライバをサポートする機構です。この JDBC クライアントドライバでは、<サブプロトコル> は「`filemaker`」です。<サブネーム> は、FileMaker データソースをホストするコンピュータの IP アドレスです。

### JDBC クライアントドライバの登録と FileMaker データソースへの接続の例

ここで示すのは、次の機能を持つ JDBC クライアントアプリケーションのコードです。

1. JDBC ドライバマネージャを使用して JDBC クライアントドライバを登録する。

2. FileMaker データソースとの接続を確立します。JDBC URL は次のとおりです。

`jdbc:filemaker://192.168.1.1/database`

3. エラーコードを返します。

```
import java.sql.*;
class FMPJDBCTest
{
    public static void main(String[ ] args)
    {
        // register the JDBC client driver
        try {
            Driver d =
            (Driver)Class.forName("com.filemaker.jdbc.Driver").newInstance();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```

    }
    // establish a connection to FileMaker
    Connection con;
    try {
        con = DriverManager.getConnection("jdbc:filemaker://192.168.1.1/mydatabase","username", "password");
    } catch (Exception e) {
        System.out.println(e);
    }
    // get connection warnings
    SQLWarning warning = null;
    try {
        warning = con.getWarnings();
        if (warning == null) {
            System.out.println("No warnings");
            return;
        }
        while (warning != null) {
            System.out.println("Warning: "+warning);
            warning = warning.getNextWarning();
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}
}

```

メモ この例はコンパイルできません。

### URL サブネームでのドライバプロパティの指定

user および password のドライバプロパティを JDBC URL のサブネームに指定します。これらは、プロパティパラメータによって DriverManager.getConnection メソッドを呼び出す際に接続へ渡されるプロパティです。

- user: [ODBC/JDBC によるアクセス] 拡張アクセス権が割り当てられたアクセス権セットを使用する FileMaker データベースファイルのアカウント
- password: FileMaker データベースファイルのアカウントのパスワード

### URL 内にデータベース名が指定された JDBC URL 接続

形式:

jdbc:filemaker://<filemaker ホストの IP アドレス>/<データベース名>

例:

jdbc:filemaker://192.168.1.1/publications

### URL 内にユーザ名、パスワード、およびデータベース名が指定された JDBC URL 接続

形式:

jdbc:filemaker://<filemaker ホストの IP アドレス>/<データベース名>?user=<データベースユーザ名>&password=<データベースパスワード>

例:

jdbc:filemaker://192.168.1.1/customers?user=Collections&password=admin

メモ この構文にはアンパサンド文字 (&) が使用されているため、ユーザ名やパスワードにアンパサンド文字を使用することはできません。

無効なユーザ名の例：

```
jdbc:filemaker://localhost/sales_db?user=ad&min&password=admin
```

無効なパスワードの例：

```
jdbc:filemaker://localhost/sales_db?user=admin1&password=ad&min
```

## 複数の FileMaker データベースファイルを使用したソリューション

1つのソリューションで複数のデータベースファイルを使用する場合は、ソリューションに必要な外部データソース参照、テーブルの別の名前、およびリレーションシップがすべて含まれる追加のデータベースを作成します。その後で、データソースとしてこの追加データベースファイルを JDBC URL に定義します。FileMaker データベースファイルは同じコンピュータ上にある必要があります。

## JDBC によるアクセスの確認

JDBC による FileMaker データベースファイルへのアクセスを確認する場合は、次の点を確認してください。

- FileMaker データベースファイルがホストされていて利用可能である
  - FileMaker アカウントで、[ODBC/JDBC によるアクセス] 拡張アクセス権が割り当てられたアクセス権セットが使用されている
  - FileMaker Pro または FileMaker Server Advanced ホストアプリケーションが ODBC/JDBC でデータを共有できるように設定されている
- FileMaker データベースファイルをデータソースとして共有するには、FileMaker Pro を使用して、データベースファイルへのアクセスを必要とするアカウントを定義します。続いて、[ODBC/JDBC によるアクセス] 拡張アクセス権を含むアクセス権セットをアカウントに割り当てることによって、データベースファイルへのアクセスを制御します。最後に、FileMaker Server Advanced または FileMaker Pro ホストアプリケーションが ODBC/JDBC によってデータを共有できるようにします。詳細については、FileMaker Pro ヘルプを参照してください。
- JDBC クライアントドライバの登録および JDBC URL が正しい（ドライバは、Java アプリケーション内に含めるか、またはクライアントコンピュータ上に配置できます）

JDBC を使用した FileMaker データの共有の詳細については、[www.filemaker.co.jp/support/technologies](http://www.filemaker.co.jp/support/technologies) を参照してください。



## 第 7 章

# サポートされている標準

この章では、FileMaker ODBC と JDBC クライアントドライバでサポートされている SQL ステートメントとコンストラクトについて説明します。これらのクライアントドライバは、ODBC または JDBC 準拠のアプリケーションから FileMaker データベースソリューションにアクセスする場合に使用します。FileMaker データベースソリューションは、FileMaker Pro または FileMaker Server Advanced によりホストされます。

ODBC クライアントドライバは、Level 2 の一部の機能を持つ ODBC 3.5 Level 1 をサポートします。JDBC クライアントドライバは、JDBC 3.0 仕様を部分的にサポートします。詳細については、[www.filemaker.co.jp/support/technologies](http://www.filemaker.co.jp/support/technologies) を参照してください。ODBC および JDBC クライアントドライバは、SQL-92 エントリレベルに準拠しており、SQL-92 中間レベルの機能も部分的にサポートしています。

### Unicode 文字のサポート

ODBC および JDBC クライアントドライバでは、Unicode API がサポートされています。ただし、これらのクライアントドライバを使用するカスタムアプリケーションを作成するときは、Unicode に対応していないクエリツールやアプリケーションが使用される場合に備えて、フィールド名、テーブル名、およびファイル名には ASCII を使用してください。

**メモ** Unicode データの挿入および取得には、SQL\_C\_WCHAR を使用します。

### SQL ステートメント

ODBC および JDBC クライアントドライバでは、次の SQL ステートメントに対するサポートが提供されています。

SELECT (以下を参照)	DELETE (38 ページ)	INSERT (39 ページ)	UPDATE (40 ページ)
CREATE TABLE (41 ページ)	ALTER TABLE (41 ページ)	CREATE INDEX (42 ページ)	DROP INDEX (42 ページ)

また、クライアントドライバでは、FileMaker データタイプと ODBC SQL および JDBC SQL データタイプのマッピングもサポートされています。データタイプの変換については、53 ページの「FileMaker フィールドタイプの ODBC データタイプへのマッピング」および 53 ページの「FileMaker フィールドの JDBC データタイプへのマッピング」を参照してください。SQL クエリーの構築の詳細については、他社の書籍を参照してください。

**メモ** ODBC および JDBC クライアントドライバでは、FileMaker ポータルはサポートされていません。

### SELECT ステートメント

SELECT ステートメントを使用して、要求する列を指定します。SELECT ステートメントの後に、フィールド名に相当する、抽出する列式を指定します（たとえば、姓）。列式には、数式演算または文字操作（たとえば、給与 \* 1.05）を入れることができます。

SELECT ステートメントでは、次のようなさまざまな句を使用できます。

```
SELECT [DISTINCT] { * | 列式 [[AS] 列エイリアス ], ... }  
FROM テーブル名 [ テーブルエイリアス ], ...  
[ WHERE 式 1 リレーショナル演算子 式 2 ]  
[ GROUP BY { 列式, ... } ]  
[ HAVING 式 1 リレーショナル演算子 式 2 ]  
[ UNION [ALL] (SELECT ...) ]  
[ ORDER BY { 列式 [DESC | ASC] }, ... ]  
[ FOR UPDATE [OF { 列式, ... } ] ]
```

角カッコ ([ ]) 内の項目は省略可能です。

列エイリアスを使用して、列によりわかりやすい名前を付けたり、長い列名を短縮したりすることができます。たとえば、列「部門」にエイリアス「所属部門」を割り当てるには、次のように指定します。

```
SELECT 部門 AS 所属部門 FROM 従業員名簿
```

フィールド名に、テーブル名またはテーブルエイリアスの接頭辞を付けることができます。たとえば、「従業員名簿. 姓」または「名簿. 姓」のように指定します。この場合、「名簿」はテーブル「従業員名簿」のエイリアスです。

DISTINCT 演算子は、最初の列式の前に配置できます。この演算子は、クエリー結果から重複する行を除去します。例：

```
SELECT DISTINCT 部門 FROM 従業員名簿
```

## SQL 句

ODBC および JDBC クライアントドライバでは、次の SQL 句に対するサポートが提供されています。

SQL 句	目的
FROM (34 ページ)	SELECT ステートメントで使用するテーブルを指定します。
WHERE (35 ページ)	FileMaker Pro の検索条件のように、抽出するレコードの条件を指定します。
GROUP BY (35 ページ)	返された値をグループ化するための、1 つまたは複数のフィールド名を指定します。この句は、FileMaker Pro の小計のように、それぞれのグループについて 1 行を返すことによって集計値のセットを返すときに使用します。
HAVING (35 ページ)	グループ化の条件を指定します。たとえば、給与総額が 20,000,000 円以上の部門だけを表示する場合などです。
UNION (35 ページ)	2 つ以上の SELECT ステートメントの結果を単一の結果に結合します。
ORDER BY (36 ページ)	レコードのソート方法を指定します。
FOR UPDATE (36 ページ)	SQL カーソルで位置付け更新または位置付け削除を実行する場合

メモ 列なしのテーブルからデータを取得しようとしても、SELECT ステートメントは何も返しません。

## FROM 句

FROM 句は、SELECT ステートメントで使用するテーブルを指定します。形式は次のとおりです。

```
FROM テーブル名 [ テーブルエイリアス ] [, テーブル名 [ テーブルエイリアス ]]
```

テーブル名は、現在のデータベースのテーブルの名前です。

テーブルエイリアスを使用して、テーブルによりわかりやすい名前を付けたり、長いテーブル名を短縮したり、同じテーブルにクエリーを複数回含めることができます（たとえば、自己連結など）。

フィールド名に、テーブル名またはテーブルエイリアスの接頭辞を付けることができます。たとえば、テーブルの指定が FROM 従業員 E の場合は、E. 姓で姓フィールドを参照できます。SELECT ステートメントでテーブルをそれ自身に結合する場合は、テーブルエイリアスを使用する必要があります。例：

```
SELECT * FROM 従業員 E, 従業員 F WHERE E. 管理者番号 = F. 従業員番号
```

等号記号 (=) を指定すると、一致する行のみが結果に含まれます。

複数のテーブルを結合して、両方のソーステーブルに対応する行が存在しない行をすべて破棄する場合は、INNER JOIN を使用できます。例：

```
SELECT *
FROM 営業社員 INNER JOIN 営業データ
ON 営業社員. 営業社員番号 = 営業データ. 営業社員番号
```

2 つのテーブルを結合して、最初のテーブル（「左の」テーブル）の行を破棄しない場合は、LEFT JOIN を使用できます。

```
SELECT *
FROM 営業社員 LEFT JOIN 営業データ
ON 営業社員. 営業社員番号 = 営業データ. 所属部門番号
```

「営業社員」テーブルのすべての行が、結合されたテーブルに表示されます。

2つのテーブルを結合して、2番目のテーブル（「右の」テーブル）の行を破棄しない場合は、RIGHT JOIN を使用できます。

```
SELECT *
FROM 営業社員 RIGHT JOIN 営業データ
ON 営業社員.営業社員番号 = 営業データ.所属部門番号
```

「営業データ」テーブルのすべての行が、結合されたテーブルに表示されます。

#### メモ

- LEFT JOIN はサポートされていますが、LEFT OUTER JOIN 文法は現在サポートされていません。
- RIGHT JOIN はサポートされていますが、RIGHT OUTER JOIN 文法は現在サポートされていません。
- FULL OUTER JOIN については現在サポートされていません。

## WHERE 句

WHERE 句は、抽出するレコードの条件を指定します。WHERE 句には、次の形式で条件を含めます。

```
WHERE 式1 リレーショナル演算子 式2
```

式1 および式2 には、フィールド名、定数値、または式を指定できます。

リレーショナル演算子は、2つの式をリンクするリレーショナル演算子です。たとえば、次の SELECT ステートメントは、給与が 2,000,000 円以上の従業員の名前を抽出します。

```
SELECT 姓, 名 FROM 従業員名簿 WHERE 給与 >= 2000000
```

WHERE 句では、次のような式を使用することもできます。

```
WHERE 式1 IS NULL
WHERE NOT 式2
```

**メモ** SELECT リスト（射影リスト）で完全修飾名を使用する場合は、関連する WHERE 句でも完全修飾名を使用する必要があります。

## GROUP BY 句

GROUP BY 句は、返された値をグループ化するための、1つまたは複数のフィールド名を指定します。この句は、集計値のセットを返すときに使用します。この句の形式は次のとおりです。

```
GROUP BY 列
```

列は、SELECT 句で使用されている列式に一致する必要があります。列式には、データベーステーブルの1つまたは複数のフィールド名をコンマで区切って指定できます。

次の例では、各部門の給与の合計を求めます。

```
SELECT 部門番号, SUM(給与) FROM 従業員名簿 GROUP BY 部門番号
```

このステートメントは、固有の各部門番号に対して1つの行を返します。各行には、部門番号、およびその部門の従業員の給与の合計が含まれます。

## HAVING 句

HAVING 句を使用することで、グループ化の条件を指定できます。たとえば、給与総額が 20,000,000 円を超える部門だけを表示する場合などです。この句の形式は次のとおりです。

```
HAVING 式1 リレーショナル演算子 式2
```

式1 および式2 には、フィールド名、定数値、または式を指定できます。これらの式は、SELECT 句の列式に一致する必要はありません。

リレーショナル演算子は、2つの式をリンクするリレーショナル演算子です。次の例では、給与の合計が 20,000,000 円を超える部門のみを返します。

```
SELECT 部門番号, SUM(給与) FROM 従業員名簿
GROUP BY 部門番号 HAVING SUM(給与) > 20000000
```

## UNION 演算子

UNION 演算子は、複数の SELECT ステートメントの結果を単一の結果に結合します。この単一の結果には、SELECT ステートメントから返されたレコードがすべて入ります。ただし、デフォルトでは、重複したレコードは返されません。重複レコードを返すには、ALL キーワードを使用します (UNION ALL)。形式は次のとおりです。

SELECT ステートメント UNION [ALL] SELECT ステートメント

UNION 演算子を使用する場合、各 SELECT ステートメントの選択リストには、同じデータタイプで同じ数の列式が含まれていて、同じ順序で指定されている必要があります。例：

SELECT 姓, 給与, 入社年月日 FROM 従業員名簿 UNION SELECT 名前, 給料, 生年月日 FROM 従業員

この例では同じ数の列式が使用されていて、各列式は順番に同じデータタイプになっています。

次の例は、列式のデータタイプが異なる (従業員名簿の給与と昇給の姓のデータタイプが異なる) ため、有効ではありません。この例では、各 SELECT ステートメントの列式の数と同じですが、データタイプごとの式の順序が同じではありません。

SELECT 姓, 給与 FROM 従業員名簿 UNION SELECT 給与, 姓 FROM 昇給

## ORDER BY 句

ORDER BY 句は、レコードのソート方法を指定します。形式は次のとおりです。

ORDER BY { ソート式 [DESC | ASC]}, ...

ソート式には、フィールド名、式、または使用する列式の位置を示す数値を指定できます。デフォルトでは、ソートは昇順 (ASC) で実行されます。

たとえば、姓でソートしてから名でソートするには、次の SELECT ステートメントのいずれかを使用できます。

SELECT 従業員番号, 姓, 名 FROM 従業員名簿 ORDER BY 姓, 名

または

SELECT 従業員番号, 姓, 名 FROM 従業員名簿 ORDER BY 2,3

2 番目の例では、姓は SELECT に続く 2 つ目の列式にあたるので、ORDER BY 2 と指定することで、姓に従ってソートされます。

## FOR UPDATE 句

FOR UPDATE 句は、SQL カーソルによる位置付け更新または位置付け削除を実行します。形式は次のとおりです。

FOR UPDATE [OF 列式]

列式は、更新するデータベーステーブル内のフィールド名をコンマで区切ったリストです。列式はオプションで省略可能です。

次の例は、給与フィールドの値が 2,000,000 円を超える、従業員名簿データベース内のすべてのレコードを返します。各レコードは、取得時にロックされます。レコードを更新または削除する場合、ロックは変更を確定するまで維持されます。その他の場合は、次のレコードを取得するとロックは解除されます。

SELECT \* FROM 従業員名簿 WHERE 給与 > 2000000  
FOR UPDATE OF 姓, 名, 給与

次に、その他の例を示します。

使用する列	SQL の例
文字列定数	SELECT 'CatDog' FROM 営業社員
数値定数	SELECT 999 FROM 営業社員
日付定数	SELECT DATE '2011-06-05' FROM 営業社員
時刻定数	SELECT TIME '02:49:03' FROM 営業社員
タイムスタンプ定数	SELECT TIMESTAMP '2011-06-05 02:49:03' FROM 営業社員
テキスト列	SELECT 会社名 FROM 営業データ SELECT DISTINCT 会社名 FROM 営業データ
数値列	SELECT 金額 FROM 営業データ SELECT DISTINCT 金額 FROM 営業データ
日付列	SELECT 売上日 FROM 営業データ SELECT DISTINCT 売上日 FROM 営業データ
時刻列	SELECT 売上時刻 FROM 営業データ SELECT DISTINCT 売上時刻 FROM 営業データ
タイムスタンプ列	SELECT 売上タイムスタンプ FROM 営業データ SELECT DISTINCT 売上タイムスタンプ FROM 営業データ
BLOB <sup>a</sup> 列	SELECT 会社パンフレット FROM 営業データ SELECT GETAS( 会社ロゴ, 'JPEG') FROM 営業データ
ワイルドカード *	SELECT * FROM 営業社員 SELECT DISTINCT * FROM 営業社員

a. BLOB は、FileMaker データベースファイルのオブジェクトフィールドです。

### これらの例に関する注意

列は FileMaker データベースファイルのフィールドの参照です（フィールドには複数の値が含まれている場合があります）。

アスタリスク（\*）のワイルドカード文字は、「すべて」を簡単に指定する方法です。たとえば、SELECT \* FROM 営業社員では、結果は営業社員テーブル内のすべての行になります。SELECT DISTINCT \* FROM 営業社員の例では、結果は、営業社員テーブル内にある固有な（重複しない）行すべてになります。

- FileMaker は、空の文字列のデータは保存しないため、次のクエリーでは常にレコードが返されません。  
SELECT \* FROM テスト WHERE c = ''  
SELECT \* FROM テスト WHERE c <> ''
- バイナリデータで SELECT を使用している場合は、返すストリームを指定する GetAs() 関数を使用する必要があります。詳細については、“オブジェクトフィールドの内容の取得: CAST() 関数と GetAs() 関数,” を参照してください。

### オブジェクトフィールドの内容の取得: CAST() 関数と GetAs() 関数

オブジェクトフィールドから、バイナリデータ、ファイル参照情報、または特定のファイルタイプのデータを取得することができます。

バイナリデータを取得するには標準の SELECT ステートメントを使用します。例:

```
SELECT 会社パンフレット FROM 営業データ
```

ファイルまたは JPEG データが存在する場合、SELECT ステートメントによってデータがバイナリ形式で取得されます。存在しない場合は、SELECT ステートメントから <ヌル> が返されます。

ファイルへのファイルパス、ピクチャ、または Quicktime ムービーなど、オブジェクトフィールドからファイル参照情報（ファイルパスなど）を取得するには、SELECT ステートメントとともに CAST 関数を使用します。例:

```
SELECT CAST( 会社パンフレット AS VARCHAR(NNN)) FROM 営業データ
```

この例で取得する内容は次のとおりです。

- FileMaker Pro を使用してオブジェクトフィールドをファイルに挿入し、ファイルへの参照のみ保存され、SELECT ステートメントによってタイプ SQL\_VARCHAR というファイル参照情報が取得されます。
- FileMaker Pro を使用してオブジェクトフィールドにファイルの内容を挿入して、SELECT ステートメントによってファイル名が取得されます。
- 別のアプリケーションからオブジェクトフィールドにファイルがインポートされると、SELECT ステートメントは '?' を表示します。(FileMaker Pro では、ファイルは Untitled.dat と表示されます。)

オブジェクトフィールドから特定のファイルタイプのデータを取得するには、GetAs 関数を使用して、FileMaker Pro のオブジェクトフィールドへの挿入方法に従いファイルタイプを指定します。

- [挿入]>[ファイル...] コマンドでデータが挿入された場合、GetAs 関数で「ファイル」を指定します。例：  
SELECT GetAs(会社パンフレット,'ファイル') FROM 営業データ
- [挿入]>[サウンド...] コマンド (標準サウンド - Mac OS 未処理形式) でデータが挿入された場合、GetAs 関数で「snd」を指定します。例：  
SELECT GetAs(会社ミーティング,'snd') FROM 会社ニュースレター
- [挿入]>[オブジェクト...] コマンド (OLE コンテナデータ) でデータが挿入された場合、GetAs 関数で「EMBO」を指定します。例：  
SELECT GetAs(会社業績,'EMBO') FROM 年間報告
- [挿入]>[ピクチャ...] コマンドでデータが挿入された場合、ドラッグ&ドロップ、またはクリップボードから貼り付け (ペースト) して、次の一覧からファイルタイプの 1 つを指定します。例：  
SELECT GetAs(会社ロゴ,'JPEG') FROM 会社アイコン

ファイルタイプ	説明	ファイルタイプ	説明
'EMF+	Windows Enhanced Metafile Plus	'PDF'	Portable Document Format
'EPS'	Embedded PostScript	'PICT'	Mac OS (512- バイトのファイルベースのヘッダはなし)
'FPix'	Flash (FPX)	'PNGf'	Bitmap イメージ形式
'FORK'	リソースフォーク (Mac OS)	'PNTG'	MacPaint
'GIFf'	Graphics Interchange Format	'qtif'	QuickTime イメージファイル
'JPEG'	写真イメージ	'SGI'	一般的なビットマップ形式
'JP2'	JPEG 2000	'TIFF'	デジタルイメージのラスタファイル形式
'META'	Windows メタファイル (拡張)	'TPIC'	Targa
'METO'	Windows メタファイル (オリジナル)	'XMLO'	レイアウトオブジェクト
'moov'	QuickTime 旧形式 (Mac OS)	'8BPS'	PhotoShop (PSD)

## DELETE ステートメント

DELETE ステートメントを使用して、データベーステーブルからレコードを削除します。DELETE ステートメントの形式は次のとおりです。

DELETE FROM テーブル名 [ WHERE { 条件 } ]

✕モ WHERE 句は、削除するレコードを決定します。WHERE キーワードを省略すると、テーブル内のすべてのレコードが削除されますが、テーブルには影響ありません。

従業員名簿テーブルに関する DELETE ステートメントの例を次に示します。

DELETE FROM 従業員名簿 WHERE 従業員番号 = 'E10001'

それぞれの DELETE ステートメントは、WHERE 句の条件を満たすすべてのレコードを削除します。この例では、従業員番号 E10001 を持つすべてのレコードが削除されます。従業員番号は従業員名簿テーブル内の固有な値なので、1 レコードだけが削除されます。



## INSERT ステートメント

INSERT ステートメントを使用して、データベーステーブルにレコードを作成します。次のいずれかを指定できます。

- 新しいレコードとして挿入する値のリスト
- 新しいレコードのセットとして挿入するために、他のテーブルのデータをコピーする SELECT ステートメント

INSERT ステートメントの形式は次のとおりです。

```
INSERT INTO テーブル名 [( 列名, ...)] VALUES ( 値式, ...)  
[, VALUES ( 値式, ...)]
```

列名は、省略可能な列名の一覧で、VALUES 句で値が指定される列の名前と順序を示します。列名を省略した場合は、値式（値式）に、テーブルで定義されているすべての列の値を、テーブルで定義されている順序と同じ列順で指定する必要があります。列名では、最終日 [4] などのフィールドの繰り返しも指定することができます。

値式は、新しいレコードの列の値を指定する式のリストです。通常、この式は列について定数値ですが、サブクエリーを指定することもできます。文字列の値は、シングルクォーテーション (') の組で囲む必要があります。シングルクォーテーションで囲まれた文字列の値にシングルクォーテーションを含めるには、同時に 2 つのシングルクォーテーションを使用します（例: 'Don''t'）。

サブクエリーはカッコ (()) で囲む必要があります。

次の例では、式のリストを挿入します。

```
INSERT INTO 従業員名簿 ( 姓, 名, 従業員番号, 給与, 入社年月日 )  
VALUES ( '小田', '弘', 'E22345', 2750000, {2008/06/05} )
```

それぞれの INSERT ステートメントにより、データベーステーブルに 1 レコードが追加されます。この例では、従業員名簿データベーステーブルに 1 レコードが追加されました。5 つの列について値が指定されます。テーブルの残りの列には、ヌルを意味する空白の値が割り当てられます。

**メモ** オブジェクトフィールドでは、パラメータ化されたステートメントを準備し、アプリケーションからデータをストリームしていない限り、INSERT 処理を行うことができるのはテキストのみです。バイナリデータを使用するには、PutAs() 関数でタイプを指定する必要があります（PutAs(col, 'type')）。ここでのタイプの値は、37 ページの「オブジェクトフィールドの内容の取得: CAST() 関数と GetAs() 関数」で説明されているタイプです。

SELECT ステートメントは、列名のリストで指定されたそれぞれの列名の値を返すクエリーです。値式のリストの代わりに SELECT ステートメントを使用すると、あるテーブルから行のセットを選択し、INSERT ステートメントでそれを別のテーブルに挿入することができます。

次に、SELECT ステートメントを使用する INSERT ステートメントの例を示します。

```
INSERT INTO 従業員名簿 1 ( 姓, 名, 従業員番号, 部門, 給与 )  
SELECT 姓, 名, 従業員番号, 部門, 給与 FROM 従業員名簿  
WHERE 部門 = '販売'
```

このタイプの INSERT ステートメントでは、挿入する列の数が SELECT ステートメントの列の数と一致する必要があります。また、他のタイプの INSERT ステートメントで列のリストが値式のリストに対応すると同様に、挿入する列のリストが SELECT ステートメントの列のリストに対応する必要があります。たとえば、挿入される先頭の列は選択された先頭の列に対応し、挿入される 2 番目の列は選択された 2 番目の列に対応するようになります。

これらの対応する列のサイズとデータタイプは、互換性がある必要があります。SELECT ステートメント内のそれぞれの列は、INSERT ステートメント内の対応する列に対する通常の INSERT/UPDATE ステートメントで ODBC または JDBC クライアントドライバが処理できるデータタイプにしてください。SELECT ステートメント内の列の値のサイズが対応する INSERT ステートメント内の列のサイズより大きい場合は、値が制限されます。

SELECT ステートメントは、値が挿入される前に評価されます。

## UPDATE ステートメント

UPDATE ステートメントを使用して、データベーステーブル内のレコードを変更します。UPDATE ステートメントの形式は次のとおりです。

UPDATE テーブル名 SET 列名 = 式, ... [ WHERE { 条件 } ]

列名は、値を変更する列の名前です。1つのステートメントで複数の列を変更できます。

式は、列の新しい値です。

通常、この式は列について定数値ですが、サブクエリーを指定することもできます。文字列の値は、シングルクォーテーション (') の組で囲む必要があります。シングルクォーテーションで囲まれた文字列の値にシングルクォーテーションを含めるには、同時に2つのシングルクォーテーションを使用します (例: 'Don"t')。

サブクエリーはカッコ (()) で囲む必要があります。

WHERE 句は、任意の有効な句です。これによって、更新するレコードを決定します。

従業員名簿テーブルに対する UPDATE ステートメントの例を次に示します。

UPDATE 従業員名簿 SET 給与 = 32000, 控除 = 1 WHERE 従業員番号 = 'E10001'

UPDATE ステートメントは、WHERE 句の条件を満たすすべてのレコードを変更します。この例では、従業員番号 E10001 を持つすべての従業員について、給与と控除が変更されます。従業員番号は従業員名簿テーブル内の固有な値なので、1レコードだけが更新されます。

次に、サブクエリーを使用した例を示します。

UPDATE 従業員名簿 SET 給与 = (SELECT avg( 給与 ) from 従業員名簿 ) WHERE 従業員番号 = 'E10001'

この例では、従業員番号 E10001 を持つ従業員について、給与を会社の平均給与に変更します。

**メモ** オブジェクトフィールドでは、パラメータ化されたステートメントを準備し、アプリケーションからデータをストリームしていない限り、UPDATE 処理を行うことができるのはテキストのみです。バイナリデータを使用するには、PutAs() 関数でタイプを指定する必要があります (PutAs(col, 'type'))。ここでのタイプの値は、37 ページの「オブジェクトフィールドの内容の取得: CAST() 関数と GetAs() 関数」で説明されているタイプです。



## CREATE TABLE ステートメント

CREATE TABLE ステートメントを使用して、データベースファイル内にテーブルを作成します。CREATE TABLE ステートメントの形式は次のとおりです。

CREATE TABLE テーブル名 テーブル要素リスト [DEFAULT 式] [UNIQUE] [NOT NULL]  
ステートメント内で、各列の名前とデータタイプを指定します。

- テーブル名およびテーブル要素リストには、100 文字という制限があります。
- DEFAULT キーワードでは列のデフォルト値を設定できます。CURRENT\_USER、CURRENT\_DATE、CURRENT\_TIME、または CURRENT\_TIMESTAMP などのリテラルを使用することもできます。
- 列を UNIQUE に定義すると、自動的に、FileMaker データベースファイル内の対応するフィールドの入力値の制限オプション [ユニークな値] が選択されます。
- 列を NOT NULL に定義すると、自動的に、FileMaker データベースファイル内の対応するフィールドの入力値の制限オプション [空欄不可] が選択されます。このフィールドには、FileMaker Pro の [データベースの管理] ダイアログボックスの [フィールド] タブで [空欄不可] としてフラグが付けられます。

例

使用する列	SQL の例
テキスト列	CREATE TABLE テーブル 1 ( 列 1 VARCHAR, 列 2 VARCHAR (50), 列 3 VARCHAR (1001), 列 4 VARCHAR (500276))
テキスト列、NOT NULL	CREATE TABLE テーブル 1NN ( 列 1 VARCHAR NOT NULL, 列 2 VARCHAR (50) NOT NULL, 列 3 VARCHAR (1001) NOT NULL, 列 4 VARCHAR (500276) NOT NULL)
数値列	CREATE TABLE テーブル 2 ( 列 1 DECIMAL, 列 2 DECIMAL (10,0), 列 3 DECIMAL (7539,2), 列 4 DECIMAL (497925,301))
日付列	CREATE TABLE テーブル 3 ( 列 1 DATE, 列 2 DATE, 列 3 DATE, 列 4 DATE)
時刻列	CREATE TABLE テーブル 4 ( 列 1 TIME, 列 2 TIME, 列 3 TIME, 列 4 TIME)
タイムスタンプ列	CREATE TABLE テーブル 5 ( 列 1 TIMESTAMP, 列 2 TIMESTAMP, 列 3 TIMESTAMP, 列 4 TIMESTAMP)
BLOB 列	CREATE TABLE テーブル 6 ( 列 1 BLOB, 列 2 BLOB, 列 3 BLOB, 列 4 BLOB)

## ALTER TABLE ステートメント

ALTER TABLE ステートメントを使用して、データベースファイル内の既存のテーブルの構造を変更します。各ステートメントで変更できる列は 1 つだけです。ALTER TABLE ステートメントの形式は次のとおりです。

ALTER TABLE テーブル名 ADD [COLUMN] 列定義

ALTER TABLE テーブル名 DROP [COLUMN] 非修飾の列名

ALTER TABLE テーブル名 ALTER [COLUMN] 列定義 SET DEFAULT 式

ALTER TABLE テーブル名 ALTER [COLUMN] 列定義 DROP DEFAULT

ALTER TABLE ステートメントを使用する前に、テーブルの構造と変更の内容を確認する必要があります。

例

目的	SQL の例
列を追加する	ALTER TABLE 営業社員 ADD 列 1 VARCHAR
列を削除する	ALTER TABLE 営業社員 DROP 列 1
列のデフォルト値を設定する	ALTER TABLE 営業社員 ALTER 会社 SET DEFAULT 'FileMaker'
列のデフォルト値を削除する	ALTER TABLE 営業社員 ALTER 会社 DROP DEFAULT

✕ **モ** SET DEFAULT および DROP DEFAULT はテーブルの既存の行には影響しませんが、その後テーブルに追加される行のデフォルト値を変更します。

## CREATE INDEX ステートメント

CREATE INDEX ステートメントを使用して、データベースファイル内での検索を高速化します。CREATE INDEX ステートメントの形式は次のとおりです。

```
CREATE INDEX ON テーブル名 . 列名
CREATE INDEX ON テーブル名 ( 列名 )
```

CREATE INDEX は、単一の列に対してサポートされています（複数の列の索引はサポートされていません）。FileMaker データベースファイルのオブジェクトフィールドタイプ、集計フィールド、グローバル格納オプションが設定されているフィールド、または非保存の計算フィールドに対応する列に対しては、索引を作成できません。

テキスト列の索引を作成すると、FileMaker データベースファイルの対応するフィールドに対して、自動的に [データの格納] の格納オプション [最小限] が選択されます。テキスト以外の列（または日本語のテキストとして書式設定されている列）の索引を作成すると、FileMaker データベースファイルの対応するフィールドに対して、自動的に [データの格納] の格納オプション [すべて] が選択されます。

どの列の索引を作成した場合でも、FileMaker データベースファイルの対応するフィールドに対して、自動的に [データの格納] の格納オプション [必要時に索引を自動設定] が選択されます。

FileMaker は必要に応じて、索引を自動的に作成します。CREATE INDEX を使用すると、索引はオンデマンドではなくただちに構築されます。

例

```
CREATE INDEX ON 営業社員 . 営業社員番号
```

## DROP INDEX ステートメント

DROP INDEX ステートメントを使用して、データベースファイルから索引を削除します。DROP INDEX ステートメントの形式は次のとおりです。

```
DROP INDEX ON テーブル名 . 列名
DROP INDEX ON テーブル名 ( 列名 )
```

データベースファイルの容量が大きすぎる場合や、特定のフィールドをクエリーで使用する頻度が低い場合は、索引を削除します。

索引設定された多くのテキストフィールドが含まれる非常に容量の大きい FileMaker データベースファイルを操作している場合に、クエリーのパフォーマンスが低いときは、一部のフィールドから索引を削除することを検討してください。また、SELECT ステートメントでほとんど使用しないフィールドからも索引を削除することを検討します。

どの列の索引を削除した場合でも、FileMaker データベースファイルの対応するフィールドに対して、自動的に [データの格納] の格納オプション [なし] が選択され、[必要時に索引を自動設定] の選択は解除されます。

PREVENT INDEX CREATION 属性はサポートされていません。

例

```
DROP INDEX ON 営業社員 . 営業社員番号
```

## SQL 統計関数

統計関数は、レコードのセットから単一の値を返します。統計関数は、SELECT ステートメントの一部として使用するか、AVG(給与)のようにフィールド名とともに使用するか、または AVG(給与 \* 1.07)のように列式と組み合わせることができます。

列式の前に DISTINCT 演算子を指定して、重複する値を取り除くことができます。例：

```
COUNT (DISTINCT 姓)
```

この例では、固有な姓の値のみがカウントされます。

例

統計関数	返される値
SUM	数値フィールド式の値の合計。たとえば、SUM( 給与 ) は、すべての給与フィールドの値の合計を返します。
AVG	数値フィールド式の値の平均。たとえば、AVG( 給与 ) は、すべての給与フィールドの値の平均を返します。
COUNT	任意の数値式の値の数。たとえば、COUNT( 名前 ) は、名前の値の数を返します。COUNT をフィールド名とともに使用した場合、COUNT は、ヌル以外のフィールドの値の数を返します。COUNT(*) は特別な例で、ヌル値が含まれるレコードを含む、セット内のレコードの数を返します。
MAX	任意のフィールド式の最大値。たとえば、MAX( 給与 ) は、給与フィールドの最大値を返します。
MIN	任意のフィールド式の最小値。たとえば、MIN( 給与 ) は、給与フィールドの最小値を返します。

```
SELECT SUM ( 営業データ . 金額 ) AS 統計 FROM 営業データ
SELECT AVG ( 営業データ . 金額 ) AS 統計 FROM 営業データ
SELECT COUNT ( 営業データ . 金額 ) AS 統計 FROM 営業データ
SELECT MAX ( 営業データ . 金額 ) AS 統計 FROM 営業データ
WHERE 営業データ . 金額 < 300000
SELECT MIN ( 営業データ . 金額 ) AS 統計 FROM 営業データ
WHERE 営業データ . 金額 > 300000
```

## SQL 式

SELECT ステートメントの WHERE 句、HAVING 句、および ORDER BY 句で式を使用して、高度で詳細なデータベースクエリーを作成します。有効な式の要素を次に示します。

フィールド名	数値演算子	リレーショナル演算子
定数	文字演算子	論理演算子
指数表記	日付演算子	関数

## フィールド名

最も一般的な式は、計算や営業データ . 請求書番号などの単純なフィールド名です。

## 定数

定数とは、変わらない値です。たとえば、価格 \* 1.05 という式では、値 1.05 が定数です。または、「6 月の日数」という定数に値 30 を割り当てることができます。

文字定数は、シングルクォーテーション (') の組で囲む必要があります。シングルクォーテーションで囲まれた文字定数にシングルクォーテーションを含めるには、同時に 2 つのシングルクォーテーションを使用します (例: 'Don't')。

FileMaker は、次のように中カッコ ({} ) で囲まれた ODBC/JDBC 形式の日付、時刻、およびタイムスタンプ定数を処理します。

- {D '2010-06-05'}
- {T '14:35:10'}
- {TS '2010-06-05 14:35:10'}

FileMaker は、SQL-92 の構文で、次のように中カッコで囲まれていない ISO 形式の日付および時刻も処理します。

- DATE 'YYYY-MM-DD'
- TIME 'HH:MM:SS'
- TIMESTAMP 'YYYY-MM-DD HH:MM:SS'

定数	使用可能な構文の例
テキスト	'大阪市'
数字	1.05
日付	DATE '2010-06-05' { D '2010-06-05' } { 2010/06/05 } { 10/06/05 } <b>メモ:</b> 2 桁の西暦の構文は、ODBC/JDBC 形式、または SQL-92 形式ではサポートされていません。
時刻	TIME '14:35:10' { T '14:35:10' } { 14:35:10 }
タイムスタンプ	TIMESTAMP '2010-06-05 14:35:10' { TS '2010-06-05 14:35:10' } { 2010/06/05 14:35:10 } { 10/06/05 14:35:10 } この 2 桁の西暦の構文を使用したフィールドに対しては、FileMaker データベースファイルの入力値の制限オプションとして、[ 西暦 4 桁の日付 ] が選択されていないことを確認してください。 <b>メモ:</b> 2 桁の西暦の構文は、ODBC/JDBC 形式、または SQL-92 形式ではサポートされていません。

日付と時刻の値を入力する際は、データベースファイルのロケールの書式と同じ書式を使用します。たとえば、データベースがイタリア語のシステムで作成された場合は、イタリア語の日付書式と時刻書式を使用します。

## 指数または科学表記

数字は科学表記を使用して表記できます。

例

```
SELECT 列 1 / 3.4E+7 FROM テーブル 1 WHERE 計算 < 3.4E-6 * 列 2
```

## 数値演算子

数値式には、+、-、\*、/、および ^、または \*\* (指数) を含めることができます。

数値式の前に、単項のプラス (+) またはマイナス (-) を付けることができます。

## 文字演算子

複数の文字を連結することができます。

例

次の例では、姓は「田中」で、名は「一郎」です。

演算子	連結	例	結果
+	後部の空白文字を保持します。	姓 + 名	'田中 一郎'
-	後部の空白文字を末尾に移動します。	姓 - 名	'田中一郎'

## 日付演算子

日付を変更することができます。

### 例

次の例では、入社年月日は {D '2008-01-30'} です。

演算子	日付の処理	例	結果
+	日付に日数を加算します。	入社年月日 + 5	{D '2008-02-04'}
-	2つの日付の間の日数を得るか、または日付から日数を減算します。	入社年月日 - {D '2008-01-01'} 入社年月日 - 10	29 {D '2008/01/20'}

次に、その他の例を示します。

```
SELECT 売上日, 売上日 + 30 AS 統計 FROM 営業データ
SELECT 売上日, 売上日 - 30 AS 統計 FROM 営業データ
```

## リレーショナル演算子

演算子	意味
=	等しい
<>	等しくない
>	大きい
>=	大きいか等しい
<	小さい
<=	小さいか等しい
LIKE	パターンに一致する
NOT LIKE	パターンに一致しない
IS NULL	ヌルに等しい
IS NOT NULL	ヌルに等しくない
BETWEEN	下限と上限の間の値の範囲
IN	指定された値のセットの1つ、またはサブクエリーの1つ
NOT IN	指定された値のセットの1つでない、またはサブクエリーの1つでない
EXISTS	サブクエリーによって少なくとも1つのレコードが返された場合は「真」
ANY	サブクエリーによって返された各値をある値と比較する（演算子の前に、=、<>、>、>=、<、または<=を付ける必要があります）。=AnyはInと同じです。
ALL	サブクエリーによって返された各値をある値と比較する（演算子の前に、=、<>、>、>=、<、または<=を付ける必要があります）。

### 例

```
SELECT 営業データ.請求書番号 FROM 営業データ
  WHERE 営業データ.営業社員番号 = 'SP-1'
SELECT 営業データ.金額 FROM 営業データ WHERE 営業データ.請求書番号 <> 125
SELECT 営業データ.金額 FROM 営業データ WHERE 営業データ.金額 > 300000
SELECT 営業データ.売上時刻 FROM 営業データ
  WHERE 営業データ.売上時刻 < '12:00:00'
SELECT 営業データ.会社名 FROM 営業データ
  WHERE 営業データ.会社名 LIKE '% 大学'
SELECT 営業データ.会社名 FROM 営業データ
  WHERE 営業データ.会社名 NOT LIKE '% 大学'
```

```

SELECT 営業データ.金額 FROM 営業データ WHERE 営業データ.金額 IS NULL
SELECT 営業データ.金額 FROM 営業データ WHERE 営業データ.金額 IS NOT NULL
SELECT 営業データ.請求書番号 FROM 営業データ
  WHERE 営業データ.請求書番号 BETWEEN 1 AND 10
SELECT COUNT( 営業データ.請求書番号 ) AS 統計
  FROM 営業データ WHERE 営業データ.請求書番号 IN (50,250,100)
SELECT COUNT( 営業データ.請求書番号 ) AS 統計
  FROM 営業データ WHERE 営業データ.請求書番号 NOT IN (50,250,100)
SELECT COUNT ( 営業データ.請求書番号 ) AS 統計 FROM 営業データ
  WHERE 営業データ.請求書番号 NOT IN (SELECT 営業データ.請求書番号
  FROM 営業データ WHERE 営業データ.営業社員番号 = 'SP-4')
SELECT *
  FROM 営業データ WHERE EXISTS (SELECT 営業データ.金額
  FROM 営業データ WHERE 営業データ.営業社員番号 IS NOT NULL)
SELECT *
  FROM 営業データ WHERE 営業データ.金額 = ANY (SELECT 営業データ.金額
  FROM 営業データ WHERE 営業データ.営業社員番号 = 'SP-1')
SELECT *
  FROM 営業データ WHERE 営業データ.金額 = ALL (SELECT 営業データ.金額
  FROM 営業データ WHERE 営業データ.営業社員番号 IS NULL)

```

## 論理演算子

2つ以上の条件を結合することができます。次に示すように、AND または OR を使用して条件を関連させる必要があります。

給与 = 4000000 AND 控除 = 1

論理 NOT 演算子は、次に示すように反対の意味にするために使用します。

NOT ( 給与 = 4000000 AND 控除 = 1 )

## 例

```

SELECT * FROM 営業データ WHERE 営業データ.会社名
  NOT LIKE '% 大学' AND 営業データ.金額 > 300000
SELECT * FROM 営業データ WHERE ( 営業データ.会社名
  LIKE '% 大学' OR 営業データ.金額 > 300000)
  AND 営業データ.営業社員番号 = 'SP-1'

```

## 関数

FileMaker SQL では、式で使用可能なさまざまな関数がサポートされています。文字列を返す関数、数字を返す関数、および日付を返す関数があります。

## 文字列を返す関数

文字列を返す関数	説明	例
CHR	ASCII コードを 1 文字の文字列に変換します。	CHR(67) は C を返します。
CURRENT_USER	接続時に指定されたログイン ID を返します。	
DAYNAME	指定した日付に対応する曜日の名前を返します。	
RTRIM	文字列から後部の空白を削除します。	RTRIM(' ABC ') は、「ABC」を返します。
TRIM	文字列から前部および後部の空白を削除します。	TRIM(' ABC ') は、「ABC」を返します。
LTRIM	文字列から前部の空白を削除します。	LTRIM(' ABC ') は、「ABC」を返します。
UPPER	文字列の各文字を大文字に変更します。	UPPER('Allen') は「ALLEN」を返します。
LOWER	文字列の各文字を小文字に変更します。	LOWER('Allen') は「allen」を返します。
LEFT	文字列の最も左側の文字を返します。	LEFT('Mattson',3) は「Mat」を返します。
MONTHNAME	暦月の名前を返します。	
RIGHT	文字列の最も右側の文字を返します。	RIGHT('Mattson', 4) は「tson」を返します。
SUBSTR SUBSTRING	文字列のサブ文字列を返します。文字列、抽出する最初の文字、抽出する文字数（オプション）のパラメータを指定します。	SUBSTR('Conrad', 2, 3) は「onr」を返します。 SUBSTR('Conrad', 2) は「onrad」を返します。
SPACE	空白の文字列を生成します。	SPACE(5) は「     」を返します。
STRVAL	任意のタイプの値を文字列に変換します。	STRVAL('Woltman') は「Woltman」を返します。 STRVAL(5 * 3) は「15」を返します。 STRVAL(4 = 5) は「False」を返します。 STRVAL({D '2008-12-25'}) は「2008-12-25」を返します。
TIME TIMEVAL	時刻を文字列として返します。	午後 9:49 の場合、TIME() は「21:49:00」を返します。
USERNAME USER	接続時に指定されたログイン ID を返します。	

✕ **TIME()** 関数は将来の対応が保証されていません。SQL 標準の **CURRENT\_TIME** を使用してください。

### 例

```

SELECT CHR(67) + SPACE(1) + CHR(70) FROM 営業社員
SELECT RTRIM(' ' + 営業社員. 営業社員番号) AS 統計 FROM 営業社員
SELECT TRIM(SPACE(1) + 営業社員. 営業社員番号) AS 統計 FROM 営業社員
SELECT LTRIM(' ' + 営業社員. 営業社員番号) AS 統計 FROM 営業社員
SELECT UPPER( 営業社員. 営業社員番号) AS 統計 FROM 営業社員
SELECT LOWER( 営業社員. 営業社員) AS 統計 FROM 営業社員
SELECT LEFT( 営業社員. 営業社員, 5) AS 統計 FROM 営業社員
SELECT RIGHT( 営業社員. 営業社員, 7) AS 統計 FROM 営業社員
SELECT SUBSTR( 営業社員. 営業社員番号, 2, 2) + SUBSTR( 営業社員. 営業社員番号, 4, 2) AS 統計 FROM 営業社員
SELECT SUBSTR( 営業社員. 営業社員番号, 2) + SUBSTR( 営業社員. 営業社員番号, 4) AS 統計 FROM 営業社員
SELECT SPACE(2) + 営業社員. 営業社員番号 AS 営業社員番号 FROM 営業社員
SELECT STRVAL('60506') AS 統計 FROM 営業データ WHERE 営業データ. 請求書番号 = 1

```

## 数字を返す関数

数字を返す関数	説明	例
ABS	数値式の絶対値を返します。	
ATAN	引数のアークタンジェントをラジアンで表された角度として返します。	
ATAN2	x および y 座標のアークタンジェントをラジアンで表された角度として返します。	
B	2 進数の 10 進数での値を返します。	B'1001' は「9」を返します。
CEIL CEILING	引数以上で最小の整数値を返します。	
DEG DEGREES	引数の角度の数字、つまりラジアンで表された角度を返します。	
DAY	日付の日の部分を返します。	DAY({d '2010/01/30'}) は「30」を返します。
DAYOFWEEK	日付式の曜日を 1 から 7 の数字で返します。	DAYOFWEEK({d '2004/05/01'}) は「7」を返します。
MOD	2 つの数字を除算して、除算の余りを返します。	MOD(10, 3) は「1」を返します。
EXP	引数で指定された乗数で累乗した自然対数 (e) のベースとなる値を返します。	
FLOOR	引数以下で最大の整数値を返します。	
HOURL	値の時間の部分を返します。	
INT	数字の整数の部分を返します。	INT(6.4321) は「6」を返します。
LEN LENGTH	文字列の長さを返します。	LEN('ABC') は「3」を返します。
MONTH	日付の月の部分を返します。	MONTH({d '2010/01/30'}) は「1」を返します。
LN LOG	引数の自然対数を返します。	
MAX	2 つの数字の大きい方を返します。	MAX(66, 89) は「89」を返します。
MIN	2 つの数字の小さい方を返します。	MIN(66, 89) は「66」を返します。
MINUTE	値の分の部分を返します。	
NUMVAL	文字列を数字に変換します。文字列が有効な数字にならない場合は、0 を返します。	NUMVAL('123') は「123」を返します。
PI	数学的定数 pi の定数値を返します。	
POW	数字を乗数で累乗します。	POW(7,2) は「49」を返します。
RADIANS	角度で表された引数のラジアンの数字を返します。	
ROUND	数字を四捨五入します。	ROUND(123.456,0) は「123」を返します。 ROUND(123.456,2) は「123.46」を返します。 ROUND(123.456, -2) は「100」を返します。
SECOND	値の秒の部分を返します。	
SIGN	引数の記号を示します。- 1 は負、0 は 0、1 は正です。	
SIN	引数のサインを返します。	
SQRT	引数の平方根を返します。	
TAN	引数のタンジェントを返します。	



数字を返す関数	説明	例
VAL	文字列を数字に変換します。文字列が有効な数字にならない場合は、0 を返します。	VAL('123') は「123」を返します。
X	16 進数の 10 進数での値を返します。	X'b9' は「185」を返します。
YEAR	日付の年の部分を返します。	DAY({d '2010/01/30'}) は「2010」を返します。

## 日付を返す関数

日付を返す関数	説明	例
CURDATE CURRENT_DATE	今日の日付を返します。	
CURTIME CURRENT_TIME	現在の時刻を返します。	
CURTIMESTAMP CURRENT_TIMESTAMP TIMESTAMPVAL	現在のタイムスタンプの値を返します。	
DATE TODAY	今日の日付を返します。	今日が 2010/11/21 の場合、DATE() は「2010-11-21」を返します。
DATEVAL	文字列を日付に変換します。	DATEVAL('2011/01/30') は「2011-01-30」を返します。

✕モ DATE() 関数は将来の対応が保証されていません。SQL 標準の CURRENT\_DATE を使用してください。

## 演算子の優先順位

式が複雑になるに連れて、式が評価される順序が重要になってきます。次の表は、演算子が評価される順序を示します。最初に 1 行目の演算子、続いて 2 行目の演算子というように評価されます。同じ行にある演算子は、式の左から右に評価されます。

優先順位	演算子
1	単項の「-」、単項の「+」
2	^, **
3	*, /
4	+, -
5	=, <, <=, >, >=, Like, Not Like, Is Null, Is Not Null, Between, In, Exists, Any, All
6	Not
7	AND
8	OR

次に、優先順位の重要性を示す例を挙げます。

WHERE 給与 > 4000000 OR 入社年月日 > {d '2008/01/30'} AND 部門 = 'D101'  
AND が最初に評価されるため、このクエリーでは、2008 年 1 月 30 日より後に入社した部門 D101 の従業員の他に、部門や入社年月日に関係なく給与が 4,000,000 円を超えるすべての従業員も抽出されます。

この句を強制的に異なる順序で評価するには、カッコを使用して、最初に評価する条件を囲みます。例：

WHERE ( 給与 > 4000000 OR 入社年月日 > {d '2008/01/30'} ) AND 部門 = 'D101'  
この句では、給与が 4,000,000 円を超えるか、または 2008 年 1 月 30 日より後に入社した部門 D101 の従業員が抽出されます。

## ODBC カタログ関数

ODBC クライアントドライバでは、次のカタログ関数がサポートされています。

- SQLTables - カタログ情報は、単一のパート名（テーブル名のみ）として保存および報告されます。
- SQLColumns
- SQLColumnPrivileges
- SQLDescribeCol
- SQLGetTypeInfo

## JDBC メタデータ関数

JDBC クライアントドライバでは、次のメタデータ関数がサポートされています。

- getColumnns
- getColumnPrivileges
- getMetaData
- getTypeInfo
- getTables
- getTableTypes

## 予約 SQL キーワード

次の表に、列、テーブル、エイリアス、またはその他のユーザ定義のオブジェクトの名前に使用できない予約キーワードを一覧にします。構文エラーが表示される場合、エラーはこれらの予約語のいずれかを使用したことによりまゝす。これらのキーワードのいずれかを使用する場合、クエスチョンマークを使用して、その語がキーワードとして処理されるのを防ぐ必要があります。

たとえば、次の Create Table ステートメントは、OID キーワードをデータ要素名として使用する方法を示しています。

```
create table t ("oid" numeric)
```

ABSOLUTE	COMMIT	DOUBLE	NSERT	OF
ACTION	CONNECT	DROP	INT	ON
ADD	CONNECTION	ELSE	INTEGER	ONLY
ALL	CONSTRAINT	END	INTERSECT	OPEN
ALLOCATE	CONSTRAINTS	END_EXEC	INTERVAL	OPTION
ALTER	CONTINUE	ESCAPE	INTO	OR
AND	CONVERT	EVERY	IS	ORDER
ANY	CORRESPONDING	EXCEPT	ISOLATION	OUTER
ARE	COUNT	EXCEPTION	JOIN	OUTPUT
AS	CREATE	EXEC	KEY	OVERLAPS
ASC	CROSS	EXECUTE	LANGUAGE	PAD
ASSERTION	CURDATE	EXISTS	LAST	PART
AT	CURRENT	EXTERNAL	LEADING	PARTIAL
AUTHORIZATION	CURRENT_DATE	EXTRACT	LEFT	POSITION
AVG	CURRENT_TIME	FALSE	LENGTH	PRECISION
BEGIN	CURRENT_TIMESTAMP	FETCH	LEVEL	PREPARE
BETWEEN	CURRENT_USER	FIRST	LIKE	PRESERVE
BINARY	CURSOR	FLOAT	LOCAL	PRIMARY
BIT	CURTIME	FOR	LONGVARBINARY	PRIOR
BIT_LENGTH	CURTIMESTAMP	FOREIGN	LOWER	PRIVILEGES
BLOB	DATE	FOUND	LTRIM	PROCEDURE
BOOLEAN	DATEVAL	FROM	MATCH	PUBLIC
BOTH	DAY	FULL	MAX	READ
BY	DAYNAME	GET	MIN	REAL
CASCADE	DAYOFWEEK	GLOBAL	MINUTE	REFERENCES
CASCADED	DEALLOCATE	GO	MODULE	RELATIVE
CASE	DEC	GOTO	MONTH	RESTRICT
CAST	DECIMAL	GRANT	MONTHNAME	REVOKE
CATALOG	DECLARE	GROUP	NAMES	RIGHT
CHAR	DEFAULT	HAVING	NATIONAL	ROLLBACK
CHARACTER	DEFERRABLE	HOURL	NATURAL	ROUND
CHARACTER_LENGTH	DEFERRED	IDENTITY	NCHAR	ROWID
CHAR_LENGTH	DELETE	IMMEDIATE	NEXT	ROWS
CHECK	DESC	IN	NO	RTRIM
CHR	DESCRIBE	INDEX	NOT	SCHEMA
CLOSE	DESCRIPTOR	INDICATOR	NULL	SCROLL
COALESCE	DIAGNOSTICS	INITIALLY	NULLIF	SECOND
COLLATE	DISCONNECT	INNER	NUMERIC	SECTION
COLLATION	DISTINCT	INPUT	NUMVAL	SELECT
COLUMN	DOMAIN	INSENSITIVE	OCTET_LENGTH	SESSION

SESSION_USER	USAGE
SET	USER
SIZE	USERNAME
SMALLINT	USING
SOME	VALUE
SPACE	VALUES
SQL	VARBINARY
SQLCODE	VARCHAR
SQLERROR	VARYING
SQLSTATE	VIEW
STRVAL	WHEN
SUBSTRING	WHENEVER
SUM	WHERE
SYSTEM_USER	WITH
TABLE	WORK
TEMPORARY	WRITE
THEN	YEAR
TIME	ZONE
TIMESTAMP	
TIMESTAMPVAL	
TIMEVAL	
TIMEZONE_HOUR	
TIMEZONE_MINUTE	
TO	
TODAY	
TRAILING	
TRANSACTION	
TRANSLATE	
TRANSLATION	
TRIM	
TRUE	
UNION	
UNIQUE	
UNKNOWN	
UPDATE	
UPPER	
USAGE	
USER	
USERNAME	
USING	

## 第 8 章

# リファレンス情報

### FileMaker フィールドタイプの ODBC データタイプへのマッピング

次の表は、FileMaker Pro のデータタイプと標準 ODBC のデータタイプのマッピングを示します。

FileMaker フィールドタイプ	変換される ODBC データタイプ	データタイプの説明
テキスト	SQL_VARCHAR	FileMaker でテキストフィールドに対して [ 最大文字数 :] を指定していない限り、テキストの最大列長は 100 万文字です。FileMaker では、空の文字列は NULL として返されます。
数字	SQL_DOUBLE	FileMaker の数字フィールドタイプには、最小 $10^{-308}$ から最大 $10^{+308}$ の正または負の値を最大 15 桁まで含めることができます。
日付	SQL_DATE	
時刻	SQL_TIME	FileMaker の時刻フィールドタイプには、時刻または時間間隔を含めることができます。時間間隔は、時刻として返されます。ただし、ゼロより小さいか 24 時より大きい場合は 0 が返されます。
タイムスタンプ	SQL_TIMESTAMP	
オブジェクト (BLOB)	SQL_LONGVARBINARY	オブジェクトフィールドから、バイナリデータ、ファイル参照情報、または特定のファイルタイプのデータを取得することができます。 SELECT ステートメント内で、CAST 関数を使用してファイル参照情報を取得し、GetAs 関数を使用して特定のファイルタイプのデータを取得します。
計算		結果は、対応する ODBC データタイプにマッピングされます。

テーブル宣言では、文字列の長さはオプションです。すべての文字列は Unicode で保存および取得されます。

✕モ FileMaker の繰り返しフィールドは配列と同じようにサポートされています。例：

```
INSERT INTO mytable(repField[3]) VALUES ('this is rep 3')  
SELECT repField[1], repField[2] FROM mytable
```

### FileMaker フィールドの JDBC データタイプへのマッピング

JDBC クライアントドライバでは、FileMaker データタイプを JDBC SQL タイプに変換する際に、次のマッピングが使用されます。（これらのデータタイプの詳細については、JDK 1.5 のドキュメント Web ページ ([www.javasoft.com](http://www.javasoft.com)) を参照してください。）

FileMaker フィールドタイプ	変換される JDBC SQL タイプ
テキスト	java.sql.Types.VARCHAR
数字	java.sql.Types.DOUBLE
日付	java.sql.Types.DATE
時刻	java.sql.Types.TIME
タイムスタンプ	java.sql.Types.TIMESTAMP
オブジェクト	java.sql.Types.BLOB
計算	計算式の結果のデータタイプによって指定

FileMaker の計算データタイプは、JDBC クライアントドライバによって、計算結果に一致する JDBC SQL タイプに変換されます。たとえば、結果がタイムスタンプデータタイプになる FileMaker の計算式は、JDBC クライアントドライバによって java.sql.Types.TIMESTAMP に変換されます。

## ODBC および JDBC エラーメッセージ

この章では、FileMaker と ODBC/JDBC の使用時に表示されるエラーメッセージの基本的な書式を示します。

### ODBC エラーメッセージ

エラーメッセージは、次のエラーやシステムから生成される可能性があります。

- ODBC ドライバのエラー
- FileMaker および FileMaker xDBC Listener のエラー

#### FileMaker ODBC エラーメッセージ

FileMaker Listener またはデータソースで発生したエラーは、データソース名も含んだ形式で次のように表示されます。

[FileMaker] [FileMaker ODBC] メッセージ

たとえば、FileMaker データソースからは次のようなメッセージが表示される場合があります。

[FileMaker] [FileMaker ODBC] Invalid Username/Password

このタイプのエラーは、データベースシステムで不正な操作を実行した場合に表示されます。詳細については、FileMaker のマニュアルを調べるか、またはデータベース管理者に連絡してください。

異なる列のエラーを参照する一連のメッセージでは、間違った列名が表示されることがあります。

### JDBC エラーメッセージ

FileMaker JDBC ドライバは、SQLException を返すことで、呼出し元のアプリケーションにエラーを報告します。エラーメッセージは、次のエラーやシステムから生成される可能性があります。

- JDBC ドライバのエラー
- FileMaker および FileMaker xDBC Listener のエラー

#### FileMaker JDBC エラーメッセージ

FileMaker Listener またはデータソースで発生したエラーは、データソース名も含んだ形式で次のように表示されます。

[FileMaker] [FileMaker JDBC] メッセージ

たとえば、FileMaker データソースからは次のようなメッセージが表示される場合があります。

[FileMaker] [FileMaker JDBC] Invalid Username/Password

このタイプのエラーは、データベースシステムで不正な操作を実行した場合に表示されます。詳細については、FileMaker のマニュアルを調べるか、またはデータベース管理者に連絡してください。

# 索引

## A

ABS 関数 48  
ALL 演算子 45  
ALTER TABLE (SQL ステートメント) 41  
AND 演算子 46  
ANY 演算子 45  
ARRAY データ型 29  
ATAN2 関数 48  
ATAN 関数 48

## B

BETWEEN 演算子 45  
BLOB データ型  
    CREATE TABLE で使用 41  
    SELECT で使用 37  
B 関数 48

## C

CAST 関数 37, 53  
CEILING 関数 48  
CEIL 関数 48  
CHR 関数 47  
CLOB データ型 29  
CREATE INDEX (SQL ステートメント) 42  
CREATE TABLE (SQL ステートメント) 41  
CURDATE 関数 49  
CURRENT USER 関数 47  
CURRENT\_DATE 関数 49  
CURRENT\_TIMESTAMP 関数 49  
CURRENT\_TIME 関数 49  
CURRENT\_USER 関数 47  
CURTIMESTAMP 関数 49  
CURTIME 関数 49

## D

DATALINK データ型 30  
DATEVAL 関数 49  
DATE 関数 49  
DAYNAME 関数 47  
DAYOFWEEK 関数 48  
DAY 関数 48  
DEGREES 関数 48  
DEG 関数 48  
DELETE (SQL ステートメント) 38  
DISTINCT 演算子 34  
DROP INDEX (SQL ステートメント) 42  
DSN  
    作成 (Mac OS) 24  
    作成 (Windows) 23  
    ファイルごとに 1 つ 10

## E

EXISTS 演算子 45  
EXP 関数 48

## F

FileMaker 製品 8  
FileMaker データソースの設定  
    JDBC 経由 31  
    ODBC 経由 (Mac OS) 24  
    ODBC 経由 (Windows) 23  
FLOOR 関数 48  
FOR UPDATE (SQL 句) 36  
FROM (SQL 句) 34  
FULL OUTER JOIN 35

## G

GetAs 関数 38, 53  
GROUP BY (SQL 句) 35

## H

HAVING (SQL 句) 35  
HOUR 関数 48

## I

INNER JOIN 34  
INSERT (SQL ステートメント) 39  
INT 関数 48  
IN 演算子 45  
IS NOT NULL 演算子 45  
IS NULL 演算子 45

## J

Java Development Kit (JDK) 30  
Java バージョン 27  
JDBC  
    エラーメッセージ 54  
    クライアントドライバ、説明 29  
    使用の概要 7  
    説明 29  
JDBC SPI 30  
JDBC クライアントドライバ  
    JDBC URL の指定 30  
    JDBC ドライバマネージャでの登録 30  
    Unicode のサポート 33  
    アクセスの確認 32  
    データタイプのマッピング 53  
    ドライバクラスおよびメインエントリーポイント 30  
    ポータル 33  
    メタデータ関数 50  
JDBC クライアントドライバの URL (Uniform Resource

Locator) 30  
JDBC クライアントドライバの登録 30  
JDBC のメタデータ関数 50

## L

LEFT JOIN 34, 35  
LEFT OUTER JOIN 34, 35  
LEFT 関数 47  
LEN 関数 48  
LIKE 演算子 45  
LN 関数 48  
LOG 関数 48  
LOWER 関数 47  
LTRIM 関数 47

## M

Mac OS  
    DSN の作成 24  
    JDBC クライアントドライバの必要条件 27  
    ODBC アクセスの確認 25  
    ODBC クライアントドライバの必要条件 15  
MAX 関数 48  
MINUTE 関数 48  
MIN 関数 48  
MOD 関数 48  
MONTHNAME 関数 47  
MONTH 関数 48

## N

NOT IN 演算子 45  
NOT LIKE 演算子 45  
NOT NULL (SQL 句) 41  
NOT 演算子 46  
NUMVAL 関数 48

## O

ODBC  
    エラーメッセージ 54  
    繰り返しフィールド 53  
    使用の概要 7  
    説明 21  
    標準準拠 33  
ODBC Administrator (Mac OS) 25  
ODBC/JDBC によるアクセス拡張アクセス権 22  
ODBC クライアントドライバ  
    Unicode のサポート 33  
    アクセスの確認 (Mac OS) 25  
    アクセスの確認 (Windows) 24  
    カタログ関数 50  
    データタイプのマッピング 53  
    ポータル 33  
ODBC データソースアドミニストレータ (Windows) 24  
ODBC のカタログ関数 50  
ORDER BY (SQL 句) 36  
OR 演算子 46

OUTER JOIN 35

## P

PI 関数 48  
POW 関数 48  
PREVENT INDEX CREATION 42

## R

RADIANS 関数 48  
Rapid Application Development (RAD) ツール 29  
REF データ型 29  
RIGHT JOIN 35  
RIGHT OUTER JOIN 35  
RIGHT 関数 47  
ROUND 関数 48  
RTRIM 関数 47

## S

SECOND 関数 48  
SELECT (SQL ステートメント) 33  
    BLOB データ型 37  
    空の文字列 37  
    バイナリデータ 37  
SIGN 関数 48  
SIN 関数 48  
SPACE 関数 47  
SQL 式の文字演算子 44  
SQL\_C\_WCHAR データ型 33  
SQL-92 33  
SQLExceptions 54  
SQL 式 43  
    演算子の優先順位 49  
    関数 46  
    指数または科学表記 44  
    数値演算子 44  
    定数 43  
    日付演算子 45  
    フィールド名 43  
    文字演算子 44  
    リテラル 43  
    リレーショナル演算子 45  
    論理演算子 46  
SQL 式の演算子の優先順位 49  
SQL 式の科学表記 44  
SQL 式の関数 46  
SQL 式の指数表記 44  
SQL 式の数値演算子 44  
SQL 式の定数 43  
SQL 式の日付演算子 45  
SQL 式のフィールド名 43  
SQL 式のリテラル 43  
SQL 式のリレーショナル演算子 45  
SQL 式の論理演算子 46  
SQL ステートメント  
    ALTER TABLE 41  
    CREATE INDEX 42



CREATE TABLE 41  
 DELETE 38  
 DROP INDEX 42  
 INSERT 39  
 SELECT 33  
 UPDATE 40  
 クライアントドライバによるサポート 33  
 予約キーワード 50

SQL 統計関数 42  
 SQL の式 43  
 SQL 標準準拠 33  
 SQRT 関数 48  
 STRVAL 関数 47  
 SUBSTRING 関数 47  
 SUBSTR 関数 47

## T

TAN 関数 48  
 TIMESTAMPVAL 関数 49  
 TIMEVAL 関数 47  
 TIME 関数 47  
 TODAY 関数 49  
 TRIM 関数 47

## U

Unicode のサポート 33  
 UNION (SQL 演算子) 36  
 UPDATE (SQL ステートメント) 40  
 UPPER 関数 47  
 USERNAME 関数 47

## V

VALUES (SQL 句) 39  
 VAL 関数 48

## W

Windows  
   DSN の作成 23  
   JDBC クライアントドライバの必要条件 27  
   ODBC アクセスの確認 24  
   ODBC クライアントドライバの必要条件 15  
 WHERE (SQL 句) 35

## X

X 関数 49

## Y

YEAR 関数 49

## あ

アカウントとアクセス権 22  
 アクセス権、拡張 22  
 アクセスの確認

JDBC クライアントドライバ 32  
 ODBC クライアントドライバ (Mac OS) 25  
 ODBC クライアントドライバ (Windows) 24  
 アクセスのテスト  
   JDBC クライアントドライバ 32  
   ODBC クライアントドライバ (Mac OS) 25  
   ODBC クライアントドライバ (Windows) 24

## い

位置付け更新および削除 36  
 インストールの必要条件 15, 27  
 インストール必要条件 15, 27

## え

エラーメッセージの形式 54

## お

オブジェクトフィールド  
   INSERT ステートメント 39  
   JDBC データタイプマッピング 53  
   SELECT ステートメント 37  
   UPDATE ステートメント 40  
   マッピング、データタイプ 53  
 オブジェクトフィールドの QuickTime ファイル 38  
 オブジェクトフィールドのイメージファイル 38  
 オブジェクトフィールドのビットマップファイル 38

## か

カーソル  
   JDBC 29  
   ODBC 36  
 概要  
   FileMaker での ODBC と JDBC の使用 7  
   アクセス権と共有の設定 22  
 拡張アクセス権 22  
 空の文字列  
   SELECT で使用 37

## き

キーワード、予約 SQL 50  
 共有された FileMaker データベースファイルの無効化 9  
 共有、ODBC/JDBC の設定 22

## く

空白文字 44  
 クライアントアプリケーション、FileMaker の使用 7  
 繰り返しフィールド 53

## け

結合 34

## こ

構文エラー 50

## さ

サーバーデータソース 25  
サブクエリー 39

## し

時刻の書式 43  
システム必要条件 15, 27  
自動生成キー 29

## せ

セーブポイントサポート 29  
接続、データベース 8

## た

タイムスタンプの書式 43

## て

データソース  
JDBC によるアクセスの確認 32  
JDBC によるアクセスのための設定 31  
ODBC によるアクセスの確認 (Mac OS) 25  
ODBC によるアクセスの確認 (Windows) 24  
ODBC によるアクセスのための設定 (Mac OS) 24  
ODBC によるアクセスのための設定 (Windows) 23  
各 FileMaker データベースファイルに対して 1 つの DSN 10  
共有された FileMaker データベースファイルの無効化 9  
データソース名。DSN を参照  
データタイプのマッピング  
JDBC クライアントドライバ 53  
ODBC クライアントドライバ 53  
データベース接続、サポート数 8  
データベース、DSN 23  
テーブルエイリアス 34

## と

統計関数、SQL 42  
ドライバ  
旧ドライバのアンインストール 10  
ドライバのプロパティ  
JDBC クライアントドライバ 31  
ODBC クライアントドライバ (Mac OS) 24  
ODBC クライアントドライバ (Windows) 23

## ぬ

ヌル値 39, 53

## ね

ネットワークの必要条件 9

## は

バイナリデータ  
SELECT で使用 37  
パスワード  
JDBC 31  
ODBC 24, 25

## ひ

日付の書式 43  
標準準拠 33

## ふ

ファイル  
アクセスの設定 22  
オブジェクトフィールドで使用 38  
1 台のコンピュータ上での編成 8  
フィールド  
JDBC へのマッピング 53  
ODBC へのマッピング 53

## ほ

ポータル 33  
ポート、JDBC 向け指定 28  
ホールダブルカーソル 29  
ホスト、DSN 23, 25

## も

文字列関数 47

## よ

予約 SQL キーワード 50

## り

リモートアクセス 8

## れ

列エイリアス 34  
列の空白の値 39  
列名 22

## ろ

論理データ型 29