

FileMaker® 8

ODBC および JDBC デベロッパーズガイド



© 2004-2005 FileMaker, Inc. All Rights Reserved.

FileMaker, Inc.
5201 Patrick Henry Drive
Santa Clara, California 95054

FileMaker 及び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。ScriptMaker 及びファイルフォルダロゴは、FileMaker, Inc. の商標です。

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。

また、製品及びサンプルファイル等に登場する会社名、氏名、住所などのデータは全て架空のもので、実在する企業、人物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。

詳細情報については www.filemaker.co.jp をご覧ください。

目次

第1章	
概要	7
このガイドについて	7
FileMaker での ODBC と JDBC の使用	7
データソースとしての FileMaker データベースファイルの使用	7
ネットワークを使用する場合の条件	8
クライアントアプリケーションとしての FileMaker Pro の使用	9
旧バージョンのファイルの更新	9
以前に FileMaker データベースファイルをデータソースとして共有していた場合	9
以前に FileMaker Pro をクライアントとして使用してデータソースにアクセスしていた場合	9
第2章	
ODBC を使用した FileMaker データの共有	11
ODBC について	11
ODBC クライアントドライバの使用	12
FileMaker データベースファイルへのアクセス - 概要	12
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	13
ODBC クライアントドライバのインストール (Windows)	13
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)	13
ODBC によるアクセスの確認 (Windows)	14
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	15
ODBC クライアントドライバのインストール (Mac OS)	15
ODBC クライアントドライバの設定 (Mac OS)	15
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)	16
ODBC によるアクセスの確認 (Mac OS)	16
第3章	
JDBC を使用した FileMaker データの共有	17
JDBC について	17
JDBC クライアントドライバの使用	17
JDBC クライアントドライバのインストール	17
JDBC クライアントドライバについて	18
JDBC URL を使用したデータベースへの接続	18
URL サブネームでのドライバプロパティの指定	19
JDBC によるアクセスの確認	20

第4章	
サポートされている標準	21
Unicode 文字のサポート	21
SQL ステートメントと句	21
SELECT ステートメント	21
DELETE ステートメント	24
INSERT ステートメント	24
UPDATE ステートメント	25
CREATE TABLE ステートメント	25
ALTER TABLE ステートメント	26
CREATE INDEX ステートメント	26
DROP INDEX ステートメント	27
FROM 句	27
WHERE 句	27
GROUP BY 句	27
HAVING 句	28
UNION 演算子	28
ORDER BY 句	28
FOR UPDATE 句	29
SQL 統計関数	29
SQL 式	29
フィールド名	29
定数とリテラル	30
指数または科学表記	30
数値演算子	30
文字演算子	31
日付演算子	31
リレーショナル演算子	31
論理演算子	32
関数	32
演算子の優先順位	34
ODBC カタログ関数	35
JDBC メタデータ関数	35
付録 A	
FileMaker フィールドタイプの ODBC データタイプへのマッピング	37
付録 B	
FileMaker Pro フィールドタイプの JDBC データタイプへのマッピング	39

付録 C	
ODBC および JDBC エラーメッセージ	41
ODBC エラーメッセージ	41
ODBC ドライバのエラーメッセージ	41
ODBC ドライバマネージャのエラーメッセージ	41
SequeLink Client のエラーメッセージ	41
SequeLink Server のエラーメッセージ	41
データソースのエラーメッセージ	42
JDBC エラーメッセージ	42
JDBC ドライバのエラーメッセージ	42
SequeLink Server のエラーメッセージ	42
データソースのエラーメッセージ	42
索引	43

第 1 章

概要

『FileMaker® ODBC および JDBC デベロッパーズガイド』をお読みいただきありがとうございます。このガイドでは、ODBC (Open Database Connectivity) および JDBC (Java Database Connectivity) を使用して他のアプリケーションと FileMaker データを共有する場合に役立つ概念と詳細事項について説明します。また、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、JDBC、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も説明します。

FileMaker Pro、FileMaker Pro Advanced、または FileMaker Server Advanced を使用して、データベースソリューションを作成およびテストできます。その後、FileMaker データベースソリューションをデータソースとして ODBC および JDBC に準拠したアプリケーションと共有したり、FileMaker データベースソリューションをクライアントアプリケーションとして使用して他の ODBC データソースにアクセスしたりすることができます。

このガイドについて

- このドキュメントには、FileMaker Pro 8、FileMaker Pro 8 Advanced、および FileMaker Server 8 Advanced で ODBC および JDBC を使用してデータにアクセスしたり、データを共有したりするための情報のみが含まれます。
- このガイドは、読者が 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。
- ODBC を使用した他のデータベースソースへのアクセスを含む、FileMaker Pro の機能についての手順ごとの説明については、FileMaker Pro ヘルプを参照してください。
- FileMaker Pro のドキュメントでは、インターネットまたはイントラネット上で Web ブラウザを使用してアクセスできるデータベースのことを指す場合に、「Web 公開」という用語を使用します。
- 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。
- このガイドでは、「FileMaker Server」を FileMaker Server 8 Advanced を指すものとして使用しています。FileMaker Server 8 Advanced では、ODBC および JDBC を使用して他のアプリケーションと FileMaker データベースファイルを共有できます。

FileMaker での ODBC と JDBC の使用

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。これらの API は、FileMaker Pro および FileMaker Server 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 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。

SQL は、ODBC と JDBC のインターフェースを介してデータソースの FileMaker ホストに渡され、SELECT 姓, 名 FROM 顧客 WHERE 都市='パリ' といったクエリーを実行したり、INSERT INTO 顧客 (姓, 名) VALUES ('高木, '幸子') を使用した新しいレコードの作成といった更新処理を行います。

データソースとしての FileMaker データベースファイルの使用

FileMaker Server 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。FileMaker Server では、最大 50 の接続を使用でき、ローカルアクセス (同じコンピュータ)、およびリモートアクセス (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) をサポートしています。

FileMaker Pro を使用して FileMaker データベースファイルを実データソースとしてホストし、ODBC および JDBC を使用してクライアントアプリケーションとデータを共有できます。FileMaker Pro では最大5つの接続を使用でき、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。

他のアプリケーションとのデータの共有に必要な xDBC プラグインコンポーネントは、FileMaker Server および FileMaker Pro で自動的にインストールされます。

ホストされている FileMaker データベースファイルへアクセスするには、対応する ODBC または JDBC 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、FileMaker CD の「xDBC」フォルダにある独立したインストールプログラムを実行することで利用できるようになります。また、最新クライアントドライバや FileMaker 8 に関する情報については、www.filemaker.co.jp を参照してください。FileMaker Server 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。

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ことが必要です。

クライアントアプリケーションでは、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、[Get external data] や [SQL query] などの名前メニュー項目があります。各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 および JDBC クライアントドライバを FileMaker Pro や FileMaker Server で使用した場合にサポートされる SQL ステートメントについては、第4章「サポートされている標準」で説明されています。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。

重要 ODBC/JDBC 共有がすでにオンになっていた場合に共有を無効にすると、FileMaker Server 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）。エラーは報告されないため、データソースを利用できなくなったために処理を完了できないことは、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時、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。Mac OS X 10.3 で FileMaker データベースファイルをホストする場合、ODBC/JDBC 共有を再び有効にする前に、FileMaker Pro または FileMaker Server を再起動してください（Mac OS X 10.4 では再起動する必要はありません）。

FileMaker データソースにアクセスするために必要なドライバファイルをインストールするには、『FileMaker ODBC および JDBC クライアントドライバのインストール』ガイドを参照してください。

サードパーティ製ツールに関する制限

Microsoft Query ウィザード

- FileMaker データソースでは、拡張 ASCII 文字または2バイト文字を含むテーブル名または列名にアクセスすることはできません。代わりに Microsoft Query を使用して、文字をダブルクォーテーションマークで囲んで手動で入力します。

Microsoft Access

- FileMaker データソースでは、拡張 ASCII 文字または2バイト文字を含むテーブル名または列名にアクセスすることはできません。
- FileMaker データソースをリンクすることはできませんが、インポートすることはできます。

ネットワークを使用する場合の条件

FileMaker Server 8 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 8 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。

クライアントアプリケーションとしての FileMaker Pro の使用

FileMaker Pro を ODBC クライアントアプリケーションとして使用して、同じコンピュータ上やネットワーク上でデータソースと情報交換を行うことができます。

サードパーティ製ツールに関する制限

Microsoft SQL Server

- FileMaker データベースファイルに decimal、float、money、または numeric のデータをエクスポートすると、入力値の自動化フィールドオプション [計算値] がデータベースに設定されます。このオプションにより、計算式 `Truncate(fieldname;0)` で値が小数点以下 0 桁に切り詰められます。作成されたデータベースでこのオプションの選択を解除してからもう一度エクスポートを実行すると、入力値の自動化フィールドオプションは解除された状態のままになります。

旧バージョンのファイルの更新

FileMaker Pro 8 では、旧バージョンのファイルメーカーで作成されたファイルを開くことができます。詳細については、『旧バージョンの FileMaker データベースの変換』ガイドを参照してください。

注意 FileMaker Server 8 では、FileMaker Pro 8 および FileMaker Pro 7v3 データベースファイルをホストできます。FileMaker Server 8 では、旧バージョンの FileMaker Pro データベースファイルをホストまたは変換することはできません。

以前に FileMaker データベースファイルをデータソースとして共有していた場合

旧バージョンで、LDAC (Local Data Access コンパニオン) または RDAC (Remote Data Access コンパニオン) を使用してデータベースファイルを共有していた場合は、2つの変更を行う必要があります。まず、アカウントのアクセス権セットの一部として [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カウントに属するようにユーザを設定する必要があります。次に、FileMaker Server または FileMaker Pro ホストで ODBC/JDBC 共有を有効にする必要があります。このためには、FileMaker Pro で、[編集] メニューの [共有設定] サブメニューから [ODBC/JDBC...] を選択して、共有を有効にします。

ODBC/JDBC を使用した共有、およびアカウントとアクセス権セットの設定の詳細については、FileMaker Pro ヘルプを参照してください。

注意 旧バージョンのファイルメーカー Server とファイルメーカー Pro では、ODBC を使用したアクセスのためにドライバを設定する場合、ホストアプリケーションに対して作成する必要がある DSN (データソース名) は 1 つだけでした。この 1 つの DSN で、すべての FileMaker データベースファイルにデータソースとしてアクセスすることが可能でした。FileMaker Server 7 および FileMaker Pro 7 以降から、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それぞれ 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。

以前に FileMaker Pro をクライアントとして使用してデータソースにアクセスしていた場合

データソースにアクセスするように設定されているドライバは引き続き動作します。更新されたドライバを提供している他社へのリンクについては、www.filemaker.com/odbc を参照してください。さらに、Windows では、SQL Server ドライバなどのクライアントドライバがオペレーティングシステムに付属しています。

重要 Mac OS では、OpenLink iODBC ドライバは、旧バージョンの FileMaker で提供されるドライバと競合します。OpenLink iODBC ドライバをインストールすると、旧バージョンの FileMaker Pro に含まれているテキスト、Oracle、および SQL Server のドライバを使用できなくなります。

FileMaker Pro を使用して ODBC によって他のデータソースにアクセスする場合の詳細については、FileMaker Pro ヘルプを参照してください。

第 2 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を使用して、別のアプリケーションから FileMaker データソースに接続します。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。Windows では、FileMaker ODBC クライアントドライバは DataDirect 32-BIT SequeLink 5.4 です。Mac OS では、FileMaker ODBC クライアントドライバは ivslk18.dylib です。

また、FileMaker Pro を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 によるデータソースへのアクセスの詳細については、FileMaker Pro ヘルプを参照してください。

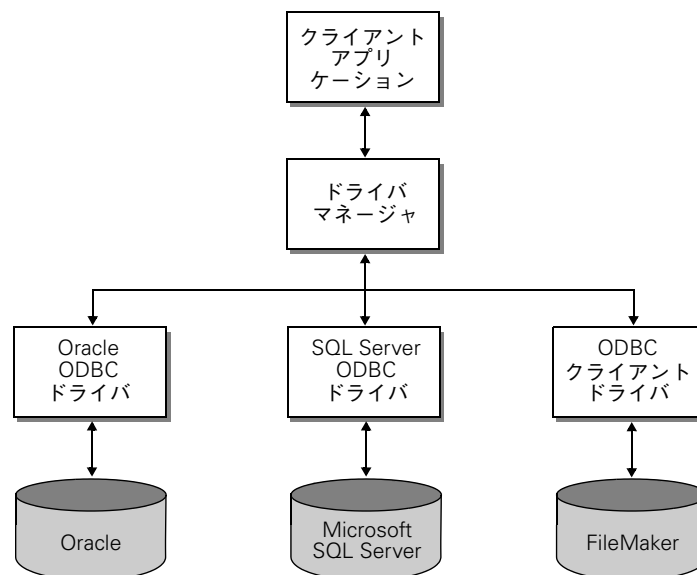
ODBC について

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）ステートメントの共通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 4 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 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 のコンポーネント



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 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 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ファイルメーカー Pro ODBC ドライバは、ODBC 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。

FileMaker データベースファイルへのアクセス - 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 (FileMaker Server Admin 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 (Windows) : [操作] メニューから [プロパティ] を選択し、[クライアント] タブをクリックします。続いて、[Server Advanced 設定...] をクリックし、[ODBC および JDBC 接続を許可] を選択します。
FileMaker Server Admin (Mac OS) : [設定] ボタンをクリックし、[クライアント] タブをクリックしてから、[ODBC および JDBC 接続を許可] を選択します。
FileMaker Pro (Windows) : [編集] メニューの [共有設定] サブメニューから [ODBC/JDBC...] を選択し、[ODBC/JDBC 共有] を [オン] にします。
FileMaker Pro (Mac OS) : [FileMaker Pro] メニューの [共有設定] サブメニューから [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コンピュータ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。

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、[Get external data] や [SQL query] 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。

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。

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。

注意 Windows の場合、Microsoft Access で、FileMaker のデータベースファイルから ODBC 経由で一度にインポートできるのは、最大 32 フィールドです。データベースファイルに 32 を超えるフィールドが含まれる場合は、32 フィールドずつインポートしてください。

重要 Mac OS 上で FileMaker データベースファイルをホストする場合、クライアントドライバは、ASCII 上位文字、2 バイト文字、または日本語文字を含むデータベース名やテーブル名にアクセスすることができません。FileMaker データベースファイルにこれらの文字が使用されている場合は、2 番目のデータベースを作成して、ファイル名やテーブル名に ASCII 文字のみを使用してください。2 番目のデータベースでは、元のデータベースファイルにあるデータを参照するファイルを作成します。両方のファイルを ODBC/JDBC で共有し、2 番目のファイルに基づいて DSN を作成します。

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

ODBC クライアントドライバのインストール (Windows)

ODBC クライアントドライバは、FileMaker Server Web Publishing CD および FileMaker Pro CD の「¥xDBC¥ODBC Client Driver Installer」フォルダにある独立したインストールプログラムを実行することで利用できるようになります。また、最新の ODBC クライアントドライバや FileMaker ODBC に関する情報については、www.filemaker.co.jp を参照してください。

インストール手順については、『FileMaker ODBC および JDBC クライアントドライバのインストール』ガイドを参照してください。このガイドは、FileMaker Pro CD の「¥xDBC¥マニュアル(PDF)」フォルダおよび FileMaker Server をインストールした「¥日本語エキストラ¥マニュアル(PDF)」フォルダに PDF 形式のドキュメントで提供されています。

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、それぞれ DSN を作成します (旧バージョンのファイルメーカー Server とファイルメーカー Pro では、該当するアプリケーションでホストされるすべての FileMaker データベースファイルに対して、DSN を1つのみ作成していました)。この DSN で、FileMaker ODBC クライアントドライバ、ホストアプリケーション (FileMaker Server または FileMaker Pro)、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。

ODBC クライアントドライバのプロパティを設定または変更するには、次の操作を行います。

1. [ODBC データ ソース アドミニストレータ] コントロールパネルを開きます。

Windows の [コントロールパネル] で、[管理ツール]、[データ ソース (ODBC)] の順に選択します。

Windows XP の場合、[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。

2. [システム DSN] タブをクリックします。

データソースをユーザ DSN またはファイル DSN として設定する場合は、対応するタブをクリックします。

3. [追加...] をクリックします。

既存のデータソースのプロパティを変更する場合は、データソースを選択してから [構成...] をクリックし、手順 6 に進みます。

4. ドライバの一覧から [DataDirect 32-BIT SequeLink 5.4] を選択します。

ODBC データソース アドミニストレータの一覧にドライバが含まれていない場合は、Regedit を使用してレジストリ キー HKEY_LOCAL_MACHINE¥SOFTWARE¥ODBC¥ODBCINST.INI¥ODBC Drivers の「(既定)」という文字列を削除します ([スタート] メニューの [ファイル名を指定して実行] を選択し、“Regedit” を指定します)。このエントリのデータは、[(値の設定なし)] です。データが空白の場合は、[(既定)] を右クリックして [削除] を選択し、[はい] をクリックして削除を確認します。この操作により、エントリが [(値の設定なし)] にリセットされます。ODBC データソース アドミニストレータに ODBC クライアントドライバが表示されるように、ODBC クライアントドライバをアンインストールして再インストールします。

5. [完了] をクリックします。

[ODBC DataDirect SequeLink Setup] ダイアログボックスが表示され、[General] タブが選択された状態になります。

6. [Data Source Name:] 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。

[Description:] はオプションです。

[Use LDAP] の選択が解除されていることを確認してください。ODBC クライアントドライバにはトランスレータ .DLL は含まれません。そのため、[Translate...] をクリックしても、ODBC トランスレータを設定することはできません。

7. [SequeLink Server Host:] 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 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。

8. [SequeLink Server Port:] に、“2399” と入力します。

9. [Server Data Source:] には、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。ファイル拡張子は入力しないでください。

FileMaker Pro または FileMaker Server ホストアプリケーションで ODBC/JDBC による共有が有効になっている場合は、[Server Data Source:] の右側にあるボタンをクリックすると、現在開いている FileMaker データベースファイルのファイル名が表示され、ここからデータベースファイルを選択できます。

データベース名にスペースが使用されている場合は、スペースをエスケープ文字 %20 に置き換えます。たとえば、“serverdatasource=MY%20DATABASE” のように入力します。

10. [OK] をクリックしてデータソース情報を保存します。

別の FileMaker データベースファイルを共有する場合は、手順 3 に戻って、データベースファイルをデータソースとして設定します。

11. [OK] をクリックして、[ODBC データソース アドミニストレータ] ダイアログボックスを閉じます。

重要 FileMaker データソースを設定する場合、[SequeLink Server Port:] に “2399” を指定する必要があります。

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. [ODBC データソース アドミニストレータ] コントロールパネルを開きます。

Windows の [コントロールパネル] で、[管理ツール]、[データソース (ODBC)] の順に選択します。

Windows XP の場合、[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。

2. [システム DSN] タブをクリックします。

データソースをユーザ DSN またはファイル DSN として設定する場合は、対応する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 [DataDirect 32-BIT SequeLink 5.4] が表示されます。

4. [構成...] をクリックします。

[DataDirect SequeLink for ODBC Setup] ダイアログボックスが表示されます。

5. [Test Connect] 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 [Database User Name:] と [Database Password:] 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、「Connection test was successful」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

カスタムアプリケーションを作成する場合は、バージョン 3.52.1 の iODBC ヘッドおよびライブラリを使用します。バージョン 3.51 で作成されたアプリケーションは、クライアントドライバをロードできない可能性があります。

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は、FileMaker Server Web Publishing CD および FileMaker Pro CD の「¥xDBC¥ODBC Client Driver Installer」フォルダにある独立したインストールプログラムを実行することで利用できるようになります。詳細については、PDF ファイル『FileMaker ODBC および JDBC クライアントドライバのインストール』を参照してください。

ODBC クライアントドライバをインストールするには、「SequeLink.bundle」ファイルをシステムまたはユーザのライブラリにコピーします。「/ODBC」フォルダがない場合は手動で作成してください。

ライブラリ	SequeLink.bundle のコピー先フォルダ
システム	/Library/ODBC
ユーザ	/Users/<ユーザ>/Library/ODBC

追加情報については、FileMaker Pro CD の「¥xDBC¥マニュアル(PDF)」フォルダおよび FileMaker Server をインストールした「¥日本語エキストラ¥マニュアル(PDF)」フォルダにある『FileMaker ODBC および JDBC クライアントドライバのインストール』ガイドを参照してください。

ODBC 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用に推奨される iODBC ドライバマネージャソフトウェアとともにバンドルされている ODBC Administrator を使用して、クライアントドライバを設定します。OpenLink ODBC Administrator は、OpenLink の Web サイトからダウンロード可能な ODBC-JDBC Lite Bridge にバンドルされています。

このクライアントドライバは、次の ODBC Administrator でテスト済みです。

- OpenLink ODBC Administrator 3.52.1 (OpenLink Software から入手可能)
- Apple ODBC Administrator 1.0.1 (Mac OS X 10.3 に付属)

クライアントドライバを設定する際は、簡単な説明とドライバファイルへのパスを入力するよう要求されます。

SequeLink.bundle のコピー先ライブラリ	設定時に使用するドライバパス
/Library/ODBC	/Library/ODBC/SequeLink.bundle/Contents/MacOS/ivslk18.dylib
/Users/<ユーザ>/Library/ODBC	/Users/<ユーザ>/Library/ODBC/SequeLink.bundle/Contents/MacOS/ivslk18.dylib

また、ODBC Administrator でオプションとしてキーワードと設定ファイルへのパスを定義することができますが、クライアントドライバにはその情報は必要ありません。

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、それぞれ DSN を作成します (旧バージョンのファイルメーカー Server とファイルメーカー Pro では、該当するアプリケーションでホストされるすべての FileMaker データベースファイルに対して、DSN を 1 つのみ作成していました)。この DSN で、FileMaker ODBC ドライバ、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。

重要 Mac OS 用のクライアントドライバは、ASCII 上位文字、2 バイト文字、または日本語文字を使用したデータベース名やテーブル名をサポートしません。FileMaker データベースファイルにこれらの文字が使用されている場合は、2 番目のデータベースを作成して、ファイル名やテーブル名に ASCII 文字のみを使用してください。2 番目のデータベースでは、元のデータベースファイルにあるデータを参照するファイルを作成します。両方のファイルを ODBC/JDBC で共有し、2 番目のファイルに基づいて DSN を作成します。

さらに、DSN に次のキーワード値を指定する必要があります。

キーワード	値
Host	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 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 アドレスまたはホスト名を入力します。
Port	“2399” を入力します。
ServerDataSource	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 (ファイル拡張子は入力しないでください)。データベース名にスペースが使用されている場合は、スペースをエスケープ文字 %20 に置き換えます。たとえば、“ServerDataDource=MY%20DATABASE” のように入力します。2 バイト文字は使用できません。

重要 ODBC Administrator で DSN を追加する際に「Could not load the driver or translator setup library」というメッセージが表示される場合は、OpenLink の Web サイトから ODBC-JDBC Lite Bridge (m615jzzz.dmg) をダウンロードしてインストールしてください。

ODBC によるアクセスの確認 (Mac OS)

OpenLink ODBC Administrator では、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ことができます。

第 3 章

JDBC を使用した FileMaker データの共有

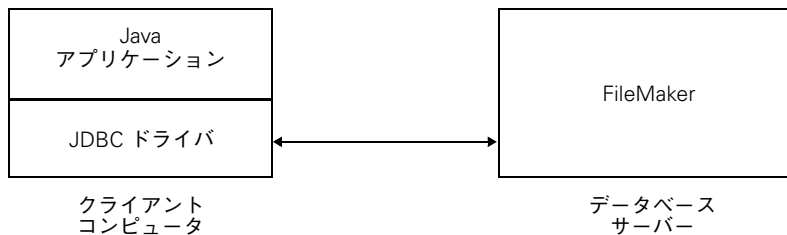
Java プログラマは、JDBC クライアントドライバを RAD (Rapid Application Development) ツールとともに使用して、FileMaker データソースに接続する Java アプリケーションやアプレットを視覚的に作成できます。この JDBC クライアントドライバを使用する Java アプリケーションやアプレットは、FileMaker データベースファイル内のデータに直接アクセスすることができます。

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。

DBMS 固有のプロトコル



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のインストール

JDBC クライアントドライバは、クラスファイルを含む Java アーカイブファイル (.jar ファイル拡張子) としてパッケージされています。最新バージョンの JDBC クライアントドライバ (Windows および Mac OS) や、FileMaker JDBC に関する情報については、www.filemaker.co.jp を参照してください。

インストールの詳細については、『FileMaker ODBC および JDBC クライアントドライバのインストール』ガイドを参照してください。このガイドは、FileMaker Pro CD の「¥xDBC¥マニュアル(PDF)」フォルダおよび FileMaker Server をインストールした「¥日本語エキストラ¥マニュアル(PDF)」フォルダに PDF 形式のドキュメントで提供されています。

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 の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 を参照してください。また、www.datadirect.com から入手可能な『SequeLink Developer's Reference』も参照してください。

JDBC クライアントドライバは、JDK (Java Development Kit) 1.4 で動作するように設計されています。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。Java 2 環境で JDBC 1.2 呼び出しのみを使用する限り、このドライバは JDK 1.3 および Java 2 で動作します。

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

```
com.ddtek.jdbc.sequelink.SequeLinkDriver
```

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ファイルメーカー 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。

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:<subprotocol>:<subname>
```

JDBC URL の最初の部分は常に JDBC プロトコル (「jdbc」) です。subprotocol 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、subprotocol は sequelink です。subname 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

次に、JDBCTest を使用して次の操作を行う場合の例を示します。

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します。
2. FileMaker データソースとの接続を確立します。JDBC URL は jdbc:sequelink://17.184.17.170:2399 です。
3. エラーコードを返します。

```
import java.sql.*;
class FMPJDBCTest
{
    public static void main(String[ ] args)
    {
        // JDBC クライアントドライバを登録します。
        try {
            Driver d =
                (Driver)Class.forName("com.ddtek.jdbc.sequelink.SequeLinkDriver").newInstance();
        } catch (Exception e) {
            System.out.println(e);
        }
        // FileMaker との接続を確立します。
        Connection con;
        try {
            con =
                DriverManager.getConnection("jdbc:sequelink://17.184.17.170:2399;
                user=some user;password=some password",userName,password);
        } catch (Exception e) {
            system.out.println(e);
        }
    }
}
```

```

}
// 接続に関する警告を取得します。
SQLWarning warning = null;
try {
    warning = con.getWarnings();
    if (warning == null) {
        System.out.println("警告はありません");
        return;
    }
    while (warning != null) {
        System.out.println("警告: "+warning);
        warning = warning.getNextWarning();
    }
} catch (Exception e) {
    System.out.println(e);
}
}
}
}

```

注意 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、Properties 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

基本的な JDBC URL 接続

形式: jdbc:sequelink://<sequelink ホストの IP アドレス>:<ポート>

これは、パスワードを使用せずに FileMaker データベースファイルに接続するための URL です。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。

ファイルが存在するコンピュータと同じコンピュータ上で JDBC URL 接続を実行する場合は、jdbc:sequelink://localhost:2399 を使用できます。

例: jdbc:sequelink://17.184.17.170:2399

URL 内にユーザ名とパスワードが定義された JDBC URL 接続

形式:

jdbc:sequelink://<sequelink ホストの IP アドレス>:<ポート>;user=<ユーザ名>;password=<パスワード>

例: jdbc:sequelink://17.184.17.170:2399;user=phil;password=jsp

URL 内にデータベース名が指定された JDBC URL 接続

形式:

jdbc:sequelink://<sequelink ホストの IP アドレス>:<ポート>;serverDataSource=<データベース名>

例: jdbc:sequelink://17.184.17.170:2399;serverDataSource=publications

データベース名にスペースが使用されている場合は、スペースをエスケープ文字 %20 に置き換えます。

例: jdbc:sequelink://17.184.17.170:2399;serverDataSource=MY%20DATABASE

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 1 (データストアのユーザ名とパスワードを使用) : jdbc:sequelink://<sequelink ホストの IP アドレス>:<ポート>;serverDataSource=<データベース名>;DBUser=<データベースユーザ名>; DBPassword=<データベースパスワード>

形式 2 (ホストのユーザ名とパスワードを使用) : jdbc:sequelink://<sequelink ホストの IP アドレス>:<ポート>;serverDataSource=<データベース名>;HUser=<ユーザ名>;HPassword=<パスワード>

注意 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ファイル参照、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。

注意 SSL (Secure Socket Layer) 暗号化はサポートされていません。より安全な JDBC ソリューションを作成するには、Web サーバーが https を使用して FileMaker データベースファイルをホストし、ファイアウォールの後方にあるデータベースファイルと通信する .ASP や .JSP のような環境を設定します。

JDBC によるアクセスの確認

JDBC クライアントドライバをインストールする際に、JDBCTest をインストールするオプションを選択すると、接続を確認することができます。JDBC クライアントドライバのインストール時に、[Install Developer's Tools] を選択して JDBCTest をインストールします。詳細については、PDF ファイル『FileMaker ODBC および JDBC クライアントドライバのインストール』を参照してください。FileMaker JDBC に関する情報については、www.filemaker.co.jp を参照してください。

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
- 使用する FileMaker アカウントで、FileMaker データベースファイル内の [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 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
- 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)

Java と JDBC を使用して一般的なデータ交換を行ったり、FileMaker データを Web 上に公開したりする場合の追加情報と例については、www.filemaker.co.jp を参照してください。

第 4 章

サポートされている標準

この章では、FileMaker ODBC および JDBC クライアントドライバでサポートされる SQL のステートメントと構造について説明します。ODBC または JDBC に準拠した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には、クライアントドライバを使用します。FileMaker データベースソリューションは、FileMaker Pro または FileMaker Server でホストできます。

ODBC クライアントドライバは、ODBC 3.5 レベル 1 の機能とレベル 2 の一部の機能をサポートします。JDBC クライアントドライバは、JDBC 3.0 の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 を参照してください。また、www.datadirect.com から入手可能な『SequeLink Developer's Reference』も参照してください。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。

注意 Unicode のデータを挿入および取得するには、SQL_C_WCHAR を使用します (SQL_C_BINARY データタイプはサポートされません)。

SQL ステートメントと句

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	DELETE	INSERT	UPDATE
CREATE TABLE	ALTER TABLE	CREATE INDEX	DROP INDEX

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、付録 A 「FileMaker フィールドタイプの ODBC データタイプへのマッピング」および付録 B 「FileMaker Pro フィールドタイプの 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。

注意 ODBC および JDBC クライアントドライバでは、繰り返しフィールドの最初の繰り返しのみが認識されます。また、FileMaker Pro のポータルはサポートされません。

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] 列別名], ... }
FROM テーブル名 [テーブル別名], ...
[ 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。

注意 より大きなデータベースでの「SELECT *」、およびテーブル別名または射影リストのリテラルを使用した SELECT 処理は、正しく機能しない場合があります。混乱を防ぐため、ワイルドカードを避け、テーブルおよび列の名前はエイリアスを使用せずに指定してください。

列別名を使用して、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」に別名「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別名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」の別名です。

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。次に例を示します。

```
SELECT DISTINCT 部門 FROM 従業員名簿
```

SQL 句	目的
FROM	SELECT ステートメントで使用するテーブルを指定します。
WHERE	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	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	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。この句は、GROUP BY 句がすでに定義されている場合にのみ有効です。
UNION	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	レコードのソート方法を指定します。
FOR UPDATE	SELECT ステートメントで選択されたデータベーステーブルのレコードをロックします。

注意 列なしのテーブルからデータを取得しようとする、SELECT ステートメントは失敗します。

例

次の例では、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
```

次の例では、ORDER BY 句を使用して姓と名を昇順にソートします。

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

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 '2004-06-05' 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 '2004-06-05 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

使用する列	SQL の例
タイムスタンプ列	<code>SELECT 売上タイムスタンプ FROM 営業データ</code> <code>SELECT DISTINCT 売上タイムスタンプ FROM 営業データ</code>
BLOB ^a 列	<code>SELECT 会社パンフレット FROM 営業データ</code> <code>SELECT GETAS(会社ロゴ, 'JPEG') FROM 営業データ</code>
ワイルドカード*	<code>SELECT * FROM 営業社員</code> <code>SELECT DISTINCT * FROM 営業社員</code>

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

定数は、SELECT ステートメントに含めることができる単一のリテラル値であるのに対して、列は、FileMaker データベースファイルのフィールドへの参照です（フィールドには複数の固有の値を含めることができます）。

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。

注意 より大きなデータベースでの `SELECT *` ステートメントは、正しく機能しない場合があります。混乱を防ぐため、ワイルドカードを避け、テーブルおよび列の名前はエイリアスを使用せずに指定してください。

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。次に例を示します。

```
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。次に例を示します。

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

この例の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にファイルを挿入し、ファイルへの参照のみを保存した場合、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た場合、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をインポートした場合、SELECT ステートメントによって「?」が表示されます（ファイルは FileMaker Pro に `Untitled.dat` として表示されます）。

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 関数を使用してファイルタイプを指定します。次に例を示します。

```
SELECT GetAs(会社ロゴ, 'JPEG') FROM 営業データ
```

FileMaker データベースファイルでオブジェクトフィールドから、次のファイルタイプを取得できます（大文字と小文字は区別されます）。

ファイルタイプ	説明	ファイルタイプ	説明
'BMPf'	ビットマップ	'PCD'	Kodak PhotoCD
'EMBO'	OLE オブジェクトデータ	'PDF'	Portable Document Format
'EMF+	Windows 拡張メタファイルプラス	'PICT'	Mac OS (512-バイトのファイルベースのヘッダはなし)

ファイル タイプ	説明	ファイル タイプ	説明
'EPS'	Embedded PostScript	'PNGf'	Bitmap イメージ形式
'FILE'	[ファイルを挿入] コマンドの結果	'PNTG'	MacPaint
'FPix'	Flash (FPX)	'qtif'	QuickTime イメージファイル
'FORK'	リソースフォーク (Mac OS)	'SGI'	一般的なビットマップ形式
'GIF'	Graphics Interchange Format	'snd'	標準サウンド (Mac OS raw 形式)
'JPEG'	写真のようなイメージ	'TIFF'	デジタルイメージのラスタファイル形式
'JP2'	JPEG 2000	'TPIC'	Targa
'META'	Windows メタファイル (拡張)	'XMLO'	レイアウトオブジェクト
'METO'	Windows メタファイル (オリジナル)	'8BPS'	PhotoShop (PSD)
'moov'	QuickTime 旧形式 (Mac OS)		

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 } ]
```

注意 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、ほか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, ...)
```

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。

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')。日付、時刻、およびタイムスタンプの値は、中カッコ {} で囲む必要があります。文字の論理値は、ピリオドで囲む必要があります (例: .T. または .F.)。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, {2004/6/5})
```

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が割り当てられます。

注意 オブジェクトフィールドで INSERT 処理を行うことができるのはテキストのみです。

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というようになります。

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')。日付、時刻、およびタイムスタンプの値は、中カッコ {} で囲む必要があります。文字の論理値は、ピリオドで囲む必要があります (例: .T. または .F.)。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。

注意 オブジェクトフィールドで UPDATE 処理を行うことができるのはテキストのみです。

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 [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。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 [データベースの定義] 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	<code>CREATE TABLE テーブル1 (列1 VARCHAR, 列2 VARCHAR (50), 列3 VARCHAR (1001), 列4 VARCHAR (500276))</code>
テキスト列、NOT NULL	<code>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)</code>
数値列	<code>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))</code>
日付列	<code>CREATE TABLE テーブル3 (列1 DATE, 列2 DATE, 列3 DATE, 列4 DATE)</code>
時刻列	<code>CREATE TABLE テーブル4 (列1 TIME, 列2 TIME, 列3 TIME, 列4 TIME)</code>
タイムスタンプ列	<code>CREATE TABLE テーブル5 (列1 TIMESTAMP, 列2 TIMESTAMP, 列3 TIMESTAMP, 列4 TIMESTAMP)</code>
BLOB 列	<code>CREATE TABLE テーブル6 (列1 BLOB, 列2 BLOB, 列3 BLOB, 列4 BLOB)</code>

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 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	<code>ALTER TABLE 営業社員 ADD 列1 VARCHAR</code>
列を削除する	<code>ALTER TABLE 営業社員 DROP 列1</code>

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 [インデックス名] [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] の索引オプション [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] の索引オプション [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] の索引オプション [必要時に索引を自動設定] が選択されます。

例

`CREATE INDEX 索引 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

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] の索引オプション [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

FROM 句

FROM 句は、SELECT ステートメントで使用され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 [テーブル別名]

テーブル名には、1つまたは複数の単純なテーブル名を現在の作業ディレクトリまたは完全パス名で指定できます。テーブル別名を使用して、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することができます。

フィールド名に、テーブル名またはテーブル別名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、テーブル別名を使用する必要があります。次に例を示します。

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。次に例を示します。

**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**

注意 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

注意 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 列式

列式は、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、あるいは1つまたは複数の式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 円以上の部門だけを表示する場合などです。この句は、GROUP BY 句がすでに定義されている場合にのみ有効です。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。次に例を示します。

```
SELECT 姓, 給与, 入社年月日 FROM 従業員名簿 UNION SELECT 名前, 給与, 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的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、SELECT ステートメントで選択されたデータベーステーブルのレコードをロックします。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
```

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て使用することができます。

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。次に例を示します。

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。

重要 SQL 関数名には大文字を使用します (大文字と小文字が区別される場合もあります)。

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	任意の数値式の最大値。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	任意の数値式の最小値。たとえば、MIN(給与)は、給与フィールドの最小値を返します。

例

```
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
定数とリテラル	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。

定数とリテラル

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。

リテラルは別の種類の定数です。ただし、割り当て済みの値を持つのではなく、「大阪市」や「14:35:10」のように、リテラル自体が値になります。リテラルは、指定した内容がそのまま使用されるという点で「WYSIWIG」な定数です。

リテラルなどの文字定数は、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')。

日付、時刻、およびタイムスタンプ定数は、{D '2005-06-05'}、{14:35:10}、{TS '2005-06-05 14:35:10'} のように、中カッコ ({}) で囲む必要があります。唯一の例外は SQL-92 の構文で、次のように、ISO 形式の日付および時刻を中カッコで囲まずに指定する必要があります。

- 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 '2005-06-05' { D '2005-06-05' } { 2005/06/05 } { 05/06/05 }
時刻	TIME '14:35:10' { T '14:35:10' } {14:35:10}
タイムスタンプ	TIMESTAMP '2005-06-05 14:35:10' { TS '2005-06-05 14:35:10' } { 2005/06/05 14:35:10 } {2005-06-05 14:35:10} { 05/06/05 14:35:10 } この2桁の西暦の構文を使用する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、[タイプ:]で [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。

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。

文字である論理値は、ピリオドで囲む必要があります。論理定数は、真を表す .T. と 1、および偽を表す .F. と 0 です。データを簡単に移動できるようにするには、1 と 0 を使用してください。

指数または科学表記

指数表記を含めることができます。

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 {2004/01/30} 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	{2004/02/04}
-	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 - {2004/01/01} 入社年月日 - 10	29 {2004/01/20}

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

ODBC および JDBC クライアントドライバでは、式内で使用できる多くの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。

重要 SQL 関数名には大文字を使用します（大文字と小文字が区別される場合があります）。

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。
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	文字列の左側から指定された数の文字を返します。	LEFT('Mattson', 3) は Mat を返します。
RIGHT	文字列の右側から指定された数の文字を返します。	RIGHT('Mattson', 4) は tson を返します。
SUBSTR	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({2004/12/25}) は 2004/12/25 を返します。
TIME	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	接続時に指定されたログイン ID を返します。	

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
SELECT TIME() AS 統計 FROM 営業データ WHERE 営業データ.請求書番号 = 1
SELECT USERNAME() 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	説明	例
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。
LEN	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({2004/01/30}) は 1 を返します。
DAY	日付の日の部分を返します。	DAY({2004/01/30}) は 30 を返します。
YEAR	日付の年の部分を返します。	YEAR({2004/01/30}) は 2004 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。
DAYOFWEEK	日付式の曜日を 1 から 7 の数字で返します。	DAYOFWEEK({2004/05/01}) は 7 を返します。

数字を返す関数	説明	例
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。
POW	数字を乗数で累乗します。	POW(7, 2) は 49 を返します。
INT	数字の整数の部分返します。	INT(6.4321) は 6 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。
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。
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。
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。
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。

日付を返す関数	説明	例
DATE	今日の日付を返します。	今日が 2005/11/21 の場合、 DATE() は {2005/11/21} を返します。
DATEVAL	文字列を日付に変換します。	DATEVAL('2006/01/30') は {2006/01/30} を返します。

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 > {2004/01/30} AND 部門 = 'D101'
```

AND が最初に評価されるため、このクエリーでは、2004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。次に例を示します。

```
WHERE (給与 > 4000000 OR 入社年月日 > {1989/01/30}) AND 部門 = 'D101'
```

この句では、給与が4,000,000円を超えるか、または2004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

JDBC では、Java の `dmd.getFunction` コマンドを使用してメタデータに対してクエリーを実行します。

付録 A

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、空の文字列はヌルとして返されます。
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。

備考

- FileMaker データベースファイルから一度に SELECT 処理を実行できるフィールドは最大 170 です。一度に UPDATE 処理を実行できるフィールドは最大 100 です。
- FileMaker Pro では、繰り返しフィールド、つまり配列データタイプがサポートされていますが、ODBC ではこのタイプはサポートされていません。繰り返しは、FileMaker Pro によってタブまたはコンマで区切られたファイルにエクスポートされ、それぞれの繰り返しがグループの区切り文字 (Unicode 10 進値 29) で区切られます。グループの区切り文字で区切られたテキスト列は連結されます。他のデータタイプでは、すべて最初の繰り返しのみが返されます。

付録 B

FileMaker Pro フィールドタイプの 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 1.4 のドキュメント Web ページ (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。

付録 C

ODBC および JDBC エラーメッセージ

この付録では、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。エラー番号と説明の一覧については、www.datadirect.co.jp を参照してください。

FileMaker におけるエラー処理の詳細については、FileMaker Pro ヘルプで Get (最終エラー) 関数または Get (最終 ODBC エラー) 関数の説明を参照してください。

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
- ODBC ドライバマネージャのエラー
- SequeLink Client のエラー
- SequeLink Server のエラー
- データソースまたはデータベース管理システム

ODBC ドライバのエラーメッセージ

SequeLink ODBC ドライバで報告されるエラーの形式は、次のとおりです。

[DataDirect] [ODBC SequeLink driver] メッセージ

次に例を示します。

[DataDirect] [ODBC SequeLink driver] Invalid precision specified

このタイプのエラーを受け取った場合は、アプリケーションで最後に実行した ODBC 呼び出しに問題がないかどうかを確認するか、ODBC アプリケーションベンダに連絡してください。

ODBC ドライバマネージャのエラーメッセージ

ODBC ドライバマネージャで報告されるエラーの形式は、次のとおりです。

[Microsoft] [ODBC Driver Manager] メッセージ

次に例を示します。

[Microsoft] [ODBC Driver Manager] 関数シーケンスエラーです。

このタイプのエラーが表示された場合は、適切な ODBC サポートファイルおよびドライバが使用されていることを確認してください。

SequeLink Client のエラーメッセージ

SequeLink ODBC Client で報告されるエラーの形式は、次のとおりです。

[DataDirect] [ODBC SequeLink driver] [SequeLink Client] メッセージ

次に例を示します。

[DataDirect] [ODBC SequeLink driver] [SequeLink Client] The specified transliteration module is not found

SequeLink Server のエラーメッセージ

SequeLink Server で報告されるエラーの形式は、次のとおりです。

[DataDirect] [ODBC SequeLink driver] [SequeLink Server] メッセージ

次に例を示します。

[DataDirect] [ODBC SequeLink driver] [SequeLink Server] Only SELECT statements are allowed in this read-only connection.

データソースのエラーメッセージ

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。

[DataDirect] [ODBC SequeLink driver] [データソース] 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。

[DataDirect] [ODBC SequeLink driver] [FileMaker] 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、間違っただけの列名が表示されることがあります。

JDBC エラーメッセージ

SequeLink for 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
- SequeLink Server のエラー
- データソースまたはデータベース管理システム

JDBC ドライバのエラーメッセージ

JDBC ドライバで報告されるエラーの形式は、次のとおりです。

[DataDirect] [SequeLink JDBC Driver] メッセージ

次に例を示します。

[DataDirect] [SequeLink JDBC Driver] Timeout expired

このタイプのエラーが表示された場合は、アプリケーションで最後に実行した JDBC 呼び出しに問題がないかどうかを確認するか、JDBC アプリケーションの製造元に連絡してください。

SequeLink Server のエラーメッセージ

SequeLink Server で報告されるエラーの形式は、次のとおりです。

[DataDirect] [SequeLink JDBC Driver] [SequeLink] メッセージ

SequeLink Server エラーが存在しない場合は、次のように表示されます。

[DataDirect] [JDBC SequeLink driver] [SequeLink]

データソースのエラーメッセージ

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。

[DataDirect] [SequeLink JDBC Driver] [データソース] 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。

[DataDirect] [SequeLink JDBC Driver] [FileMaker] 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

ALTER TABLE (SQL ステートメント) 26

C

CAST 関数 23, 37

CREATE INDEX (SQL ステートメント) 26

CREATE TABLE (SQL ステートメント) 25

D

Data Source Name 14

DELETE (SQL ステートメント) 24

DROP INDEX (SQL ステートメント) 27

F

FileMaker データソースの共有 7

FileMaker データソースの設定

JDBC 経由 19

ODBC 経由 (Mac OS) 16

ODBC 経由 (Windows) 13

FileMaker データソースのホスト 7

FOR UPDATE (SQL 句) 29

FROM (SQL 句) 27

G

GetAs 関数 23, 37

GROUP BY (SQL 句) 27

H

HAVING (SQL 句) 28

I

INNER JOIN 27

INSERT (SQL ステートメント) 24

J

JDBC クライアントドライバ

JDBC URL の指定 18

JDBC ドライバマネージャでの登録 18

Unicode のサポート 21

アクセスの確認 20

インストール 17

繰り返しフィールド 21

データタイプのマッピング 39

ドライバクラスおよびメインエントリーポイント 18

メタデータ関数 35

JDBC クライアントドライバの URL (Uniform Resource Locator) 18

JDBC クライアントドライバの登録 18

JDBC のメタデータ関数 35

JDBC、説明 17

Java Development Kit (JDK) 18

L

Local Data Access コンパニオン (LDAC) 9

M

Microsoft Access クライアントアプリケーション 8

Microsoft Query ウィザード 8

Microsoft SQL Server 9

O

ODBC

繰り返しフィールド 37

説明 11

ODBC クライアントドライバ

FileMaker フィールドの最大数 37

Unicode のサポート 21

アクセスの確認 (Mac OS) 16

アクセスの確認 (Windows) 14

インストール (Mac OS) 15

インストール (Windows) 13

カタログ関数 35

繰り返しフィールド 21

データタイプのマッピング 37

ODBC のカタログ関数 35

ORDER BY (SQL 句) 28

OUTER JOIN 27

R

Remote Data Access コンパニオン (RDAC) 9

Rapid Application Development (RAD) ツール 17

S

SELECT (SQL ステートメント) 21

SQL 式 29

演算子の優先順位 34

関数 32

指数または科学表記 30

数値演算子 30

定数 30

日付演算子 31

フィールド名 29

文字演算子 31

リテラル 30

リレーショナル演算子 31

論理演算子 32

SQL 式の演算子の優先順位 34

SQL 式の科学表記 30
SQL 式の関数 32
SQL 式の指数表記 30
SQL 式の数値演算子 30
SQL 式の定数 30
SQL 式の日付演算子 31
SQL 式のフィールド名 29
SQL 式の文字演算子 31
SQL 式のリテラル 30
SQL 式のリレーショナル演算子 31
SQL 式の論理演算子 32
SQL ステートメント
 ALTER TABLE 26
 CREATE INDEX 26
 CREATE TABLE 25
 DELETE 24
 DROP INDEX 27
 INSERT 24
 SELECT 21
 UPDATE 25
 クライアントドライバによるサポート 21
SQL 統計関数 29
SQL-92 への準拠 21
SSL (Secure Sockets Layer) 暗号化 20
SequeLink Server Host 14
SequeLink Server Port 14
Server Data Source 14

U

UNION (SQL 演算子) 28
UPDATE (SQL ステートメント) 25
Unicode のサポート 21

W

WHERE (SQL 句) 27

あ

アクセスの確認
 JDBC クライアントドライバ 20
 ODBC クライアントドライバ (Mac OS) 16
 ODBC クライアントドライバ (Windows) 14
アクセスのテスト
 JDBC クライアントドライバ 20
 ODBC クライアントドライバ (Mac OS) 16
 ODBC クライアントドライバ (Windows) 14

え

エスケープ文字 16, 19

お

オブジェクトフィールド 23
オペレーティングシステム 8

き

共有された FileMaker データベースファイルの無効化 8

く

クライアントアプリケーション、FileMaker の使用 9
繰り返しフィールド 21, 37

け

結合 27

し

式、SQL 29

て

データソース
 FileMaker データベースファイルの使用 7
 JDBC によるアクセスの確認 20
 JDBC によるアクセスのための設定 19
 ODBC によるアクセスの確認 (Mac OS) 16
 ODBC によるアクセスの確認 (Windows) 14
 ODBC によるアクセスのための設定 (Mac OS) 16
 ODBC によるアクセスのための設定 (Windows) 13
 各 FileMaker データベースファイルに対して
 1つの DSN 9
 共有された FileMaker データベースファイルの
 無効化 8
データタイプのマッピング
 JDBC クライアントドライバ 39
 ODBC クライアントドライバ 37
データベース名のスペース 16, 19

と

統計関数、SQL 29
ドライバのプロパティ
 JDBC クライアントドライバ 19
 ODBC クライアントドライバ (Mac OS) 16
 ODBC クライアントドライバ (Windows) 13

ね

ネットワークの必要条件 8

ま

マッピング、データタイプ
 JDBC クライアントドライバ 39
 ODBC クライアントドライバ 37