

第1回DBゼミ

指導:谷村 担当:吉田 小野 若森

2000年4月24日

目次

第1章 データベースとは	3
1.1 DB はなぜ用いられるのか？	3
1.2 DB を用いたシステム	4
1.2.1 DBMS の種類	4
1.2.2 SQL のみを使うクライアントサーバモデル	4
1.2.3 Web データベース	4
1.2.4 スクリプト, CGI について	5
1.3 PHP とデータベース	7
1.4 ODBC によるデータベースアクセス	8
1.4.1 ODBC の構造	8
1.4.2 ODBC によって何ができるか？	9
1.4.3 ODBC の処理について	9
1.5 JAVA によるデータベース接続	9
1.5.1 JDBC の構造	9
1.5.2 JDBC の役割	10
第2章 DB ソフトの種類について	11
2.1 PostgreSQL	11
2.1.1 ライセンス	11
2.1.2 日本語対応	11
2.1.3 性能	11
2.2 MySQL	12
2.2.1 ライセンス	12
2.2.2 日本語対応	12
2.2.3 性能	12
2.3 Oracle	12
2.3.1 ライセンス	13
2.3.2 日本語対応	13
2.3.3 性能	13
2.4 SQL Server	14
2.4.1 ライセンス	14
2.4.2 日本語対応	14
2.4.3 性能	14
2.5 まとめ	15
第3章 データベースの使い道	16
3.1 企業での使い道	16
3.2 生活の中での使い道	16
3.3 英辞郎について	17

3.3.1	英辞郎とは	17
3.3.2	英辞郎の利点	17
3.3.3	英辞郎の問題点	18

第1章 データベースとは

今回のゼミでの最終的な目標は実用的なデータベースシステムを構築することです。ただし DB としては実用的に使われているリレーショナルデータベースを対象とします。¹さまざまな場面で DB を聞いたことがあると思います。しかし DB を作るとなると Microsoft 社の ACCESS しか知らないというのが現状ではないでしょうか？本ゼミでは、どのようなソフトウェアが用いられ DB が導入され、どのように利用されているかを学ぼうと思います。

1.1 DB はなぜ用いられるのか？

一般に DB とはさまざまな利用応用を考慮して格納されたデータの集まりです。でもこれだけなら Word や Excel でファイルを作ってデータを集めておけばいいではないか？ということになります。そこでファイルとデータベースを用いることは何が違うかをまとめてみます。

まずファイルの意味は DB と同じデータの集合体です。しかし以下に示すような特徴があります。

1. ファイルはアプリケーション単位で利用され、アプリケーションとの依存度が高いためファイルの構造を変化させると、アプリケーションの変化もしなくてはならない。
2. 異なるファイル間の情報の重複や関連は系統的にまったく考慮されない。
3. ファイルをシステムの複数のユーザで共有する場合、細かな機密保護を行えない

いっぽう DB ではレコードやフィールド²といった物理的な定義情報のほかにデータのもつ意味などを DB の中に保持しています。そのためフィールドの追加変更のようにデータの構造定義が物理的、論理的に変更されてもアプリケーションを変更する必要がなくなります。

またデータの整合性を保ち、アプリケーションとの依存性をなくすために DB を利用する際には、DBMS(Data Base Management System) というソフトウェアシステム (ミドルウェア) がアプリケーションと DB の橋渡しをしてくれます。以下に DBMS の役割をまとめます。

1. DB の構造を変化させても、以前まで利用していたアプリケーションがそのまま動く
2. DB を同時に複数のユーザが利用しても、整合性を維持し、機密を保護しなくてはならない
3. 障害が発生しても少ない時間で障害前の DB の状態に戻さなくてはならない
4. DB に対する操作³を迅速に行う

このように DBMS を用いて DB を構築することによりデータを一元管理し、またデータを効率よく使用できるので、また現在では DB と DBMS をそれほど区別せずデータベースと呼ぶことが多いです。

¹RDB はデータモデルに表構造を用いる DB です。RDB は数学的基盤により明確であり、また独立性が高いという特徴をもちます。ただし詳細については独学してください。

²ファイルにおけるデータ項目の組み合わせをレコードと呼び、ひとつのデータ項目をフィールドと呼びます。概念的な大きさとしてレコード > フィールドです。

³DB に対する操作とはデータの登録、検索、更新、削除などである

1.2 DB を用いたシステム

1.2.1 DBMS の種類

DBMS には二つのタイプがあります．ひとつは ACCESS などのようなファイル共有型．もうひとつは本ゼミで取り組むクライアントサーバ型です．この節ではその二つの違いを説明します．この二つの違いはデータを処理するプログラムがどこで動くかという点です．ACCESS のようなファイル共有型 DB ではプログラムはクライアント側で稼動し，データ処理はクライアントが行います．サーバとつないだとしても，サーバの役割はデータファイルの共有のみで，DB 検索などの処理はサーバからネットワークをとうしてクライアントにファイルが送られ，実際の処理をクライアントが行います．そのためこの種の DBMS ではクライアントの処理能力が問われます．

いっぽうクライアントサーバ型では，クライアントから SQL が投げられ，サーバはその SQL を実行しその結果をクライアントに返します．そのため必要最小限のネットワークトラフィックしか発生しないが，サーバの処理能力が問われます．この型のソフトウェアとしては Oracle , Sybase , PostgreSQL などがあります．この形の DBMS としては直接 SQL を発行するものや，Web をとうして利用するものがあります．また ODBC(Open Database Connectivity) というクライアントからサーバへのデータ要求するための API(Application Programming Interface) を用いて DB を利用する方式もあります．各方法についていかに示します．

1.2.2 SQL のみを使うクライアントサーバモデル

サーバに DBMS を導入し，クライアントは DB サーバに Telnet などを用いて SQL 文を投げかけるタイプです．この場合 DB サーバに導入されている DBMS の種類によってコマンドの種類が異なりますが，SQL の文法の基本部分は異なりません．図 1.1 に Telnet を問う下 SQL を用いている例を示します．この利用形態の場合にはユーザイン

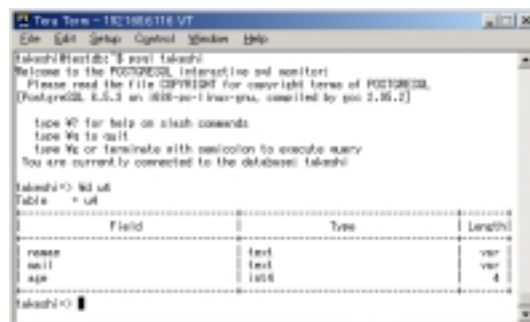


図 1.1: コンソールからの DB アクセス

タフェースの部分で問題があり，また SQL 文に対しての知識が必要となるため，一般ユーザに対しては便利ではありません．

1.2.3 Web データベース

皆さんがよく触れる DB は Web ブラウザ上から用いるものが多いのではないのでしょうか？この Web データベースはクライアントサーバ型の DBMS が用いられることが主です．そこでこの節ではクライアントから Web ブラウザを通して利用されるデータベースについて説明する．WWW サーバと DB サーバを連携させたシステムを図 1.2 に示します．図 1.2 に示す DataBase と DB サーバが一般的に DBMS と呼ばれるものです．このように私たちが Web から利用している DB は WWW サーバと DB サーバの組み合わせによるクライアントサーバモデルが元になっています．図 1.2 の各操作についてもう少し煮詰めます．

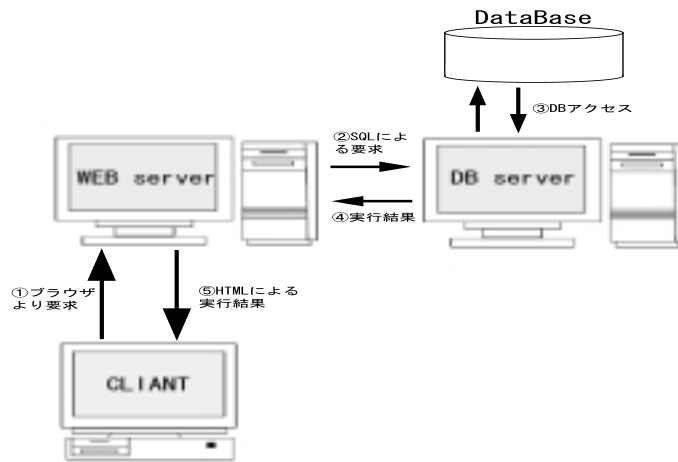


図 1.2: Web データベースの仕組み

1. クライアントの Web ブラウザから Web サーバに URL⁴を要求します。
2. Web サーバが発信するページ (以下 Web ページ) はサーバのディスク内に格納されている*.html や*.htm といったファイルは静的なものです。そのため DB 検索といった動的な処理を行うことができません。そこでサーバサイドで、URL で指定されたファイルからプログラムを実行し、DB 検索を要求するメッセージを DB サーバに送ります。このメッセージにはデータベースを操作する SQL 文を埋め込むことによって DB アクセスします。また、このようなプログラムには CGI や PHP などがあげられるが、それぞれの特徴は後ほど述べる。
3. Web サーバから受け取ったメッセージに埋め込まれた SQL を元に検索などの DB アクセスを行います。
4. DB アクセスの結果を Web サーバに渡し、クライアントから参照できるように静的な形式、つまり HTML の形に変換する。
5. クライアントの Web ブラウザに HTML による処理結果を配信する

このような形式で Web データベースは構成されています。本ゼミが目標とするシステムはこのようなものである。このようなシステムを構築するにあたり必要なものはまず Web サーバ、DB(DBMS)を提供するソフトウェア、そして DB サーバと Web サーバをどのようにつなぐかということです。まず本ゼミではサーバマシンには Linux を搭載したマシンを用います。そこで Web サーバソフトウェアとして最も普及率の高い Apache Web Server を用います。また DB サーバのソフトウェアとしては PostgreSQL を用います。PostgreSQL については第 2 章において説明します。次に DB サーバと Web サーバのインタフェースについて説明します。

1.2.4 スクリプト, CGI について

Web ページは HTML 言語で記述されており、一般に静的なものしか作れません。つまり Web サーバは静的なファイルしかクライアントに配信できないのです。しかしこれでは何かを入力して、そのたびに変化するような Web ページを見ることはできません。このことは Yahoo などの検索サイトが、すべての検索キーワードごとにファイルを用意できないことを考えると明らかです。そこで動的な出力を得るためにかんがえだされたのがスクリプトや CGI などです。スクリプトとはソフトウェアに処理をさせるプログラムのことで、HTML 内に埋め込まれる JavaScript や VBScript などがあります。スクリプトにも 2 種類ありひとつがクライアントサイドスクリプト、もうひとつがサーバサイドスクリプトがあります。いっぽう CGI は Web ページから外部プログラムを呼び出し、その結果を Web ページに反映させる仕組みのことで、WWW サーバと外部プログラムのインタフェースであるといえます。本ゼミでは PHP という

⁴URL とはネットワークリソースのアドレスのことです。Web に限って言えば特定のコンテンツの場所を示すための表記方法のことを示します。

サーバサイドスクリプト言語を用い、DB を利用するシステムを構築するため、ここで少し 3 つの違いを紹介します。

クライアントサイドスクリプト

サーバからクライアントに対して動的な画面を生成するためのスクリプト (ソースプログラム) を送り、クライアント側のブラウザがそれらを逐次解析実行する形式です。サーバの処理は軽くなりますが、クライアント側にそのスクリプトを処理する能力が必要になる。JavaScript や VBScript がそれにあたる。図 1.3 にその概念を示します。

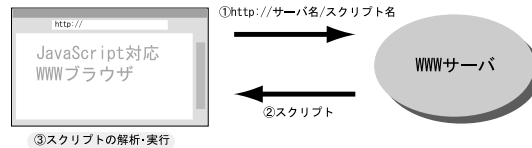


図 1.3: クライアントサイドスクリプト

CGI(Common Gateway Interface)

Web サーバが、URL で指定されたファイルを外部プログラムとして起動する形式です。この外部プログラムとは Web サーバが別プロセスを作成し実行するというので、サーバに大きな負荷がかかります。CGI プログラム自体はサーバの OS で動作可能なものであればどんなものでもかまいませんが、各種スクリプト言語が使われることが多いです。起動されたプログラムは HTML を作成し、その出力は Web サーバをとうして、そのままクライアント側に送られます。図 1.4 にその概念を示します。

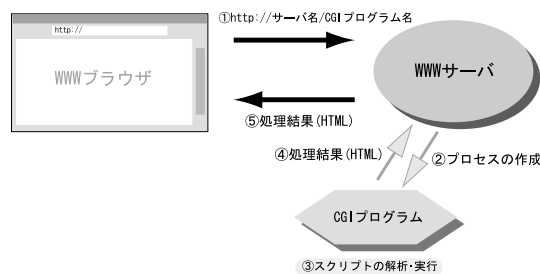


図 1.4: CGI

サーバサイドスクリプト

サーバサイドスクリプトは CGI とよく似ていますが、スクリプトの実行を Web サーバプロセス自身が行うという点で異なります。本ゼミで将来的に使用する PHP はこのタイプで、プロセスを作成しないためサーバ側の処理は軽くなります。しかしサーバプログラム自体が大きくなります。図 1.5 にその概念を示します。

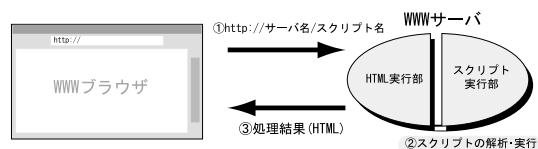


図 1.5: サーバサイドスクリプト

この 3 つ動的 Web ページの方式を表 1.1 に簡単にまとめます。

表 1.1: 動的 Web ページの実装方法

タイプ	クライアントサイドスクリプト	CGI	サーバサイドスクリプト
メリット	サーバの負担が軽い	ブラウザを選ばない	サーバの負荷が CGI より軽い
	GUI に関してきめ細かい記述が可能	記述言語を選ばない	プログラム作成が CGI より容易
		サーバの種類を選ばない	
デメリット	ブラウザを選ぶ	サーバに負担がかかる	サーバプログラムが大きくなる
		プログラム作成に OS の知識がある程度必要	インストールがやや難しい
実装	JavaScript	スクリプト言語として Perl がよく使われる	PHP

1.3 PHP とデータベース

先ほどまで述べたように本ゼミで使用する PHP は HTML 埋め込み型のサーバサイドスクリプト言語です。これを用いると WWW と DB を連携させたアプリケーションを比較的簡単に作成することができます。他にも PHP の特徴としては C や Perl と似ている部分が多く取り組みやすいこと、各種の DB とのインタフェースに優れている点があります。

では PHP を実際にどのように DB と連携させてシステムを構築していくかを説明します。図 1.6 に PHP と DBMS の連携したシステムについて説明します。DBMS にアクセスするためには SQL (Structured Query Language) を使う

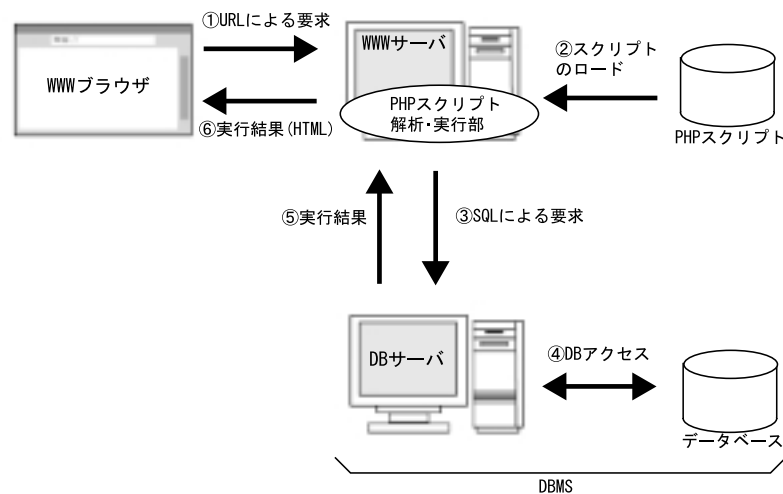


図 1.6: PHP と DBMS との連携

のが標準になっています。PHP では SQL 文を埋め込むことができ、その結果 DB サーバに DB アクセスの要求をすることができます。PHP がサポートする DBMS には Oracle, dBase, MySQL や本ゼミで使用する PostgreSQL があります。また CGI を用いて DB を検索するシステムを構築する場合、形としては 1.6 と同じになります。ただし 1.2.4 節で説明したようにサーバ上でプロセスを作成するかどうかの違いがあります。

1.4 ODBC によるデータベースアクセス

ODBC(Open Database Connectivity) はMicrosoft 社のデータベース API です。ODBC は CLI(Call Level Interface) と呼ばれる DB へのファンクションコールの規格に準拠したライブラリです。これは DB エンジンなどの DB プログラムや DB に関係しないプログラムを互いに接続して通信するために用いられています。他にもデータベース API はありますが、ODBC は PC システムで最も広く使われています。ODBC があれば、どのようなプログラムに DB ソフトウェアを接続する場合でも、通信インタフェースを開発する必要はなくなります。本ゼミでは ODBC は使いませんが、ODBC を知っておくことは損にはならないためここで説明します。

1.4.1 ODBC の構造

ODBC は図 1.7 に示すように ODBC データソース、ODBC ドライバ、ODBC ドライバマネージャ、クライアントアプリケーションと、主に 4 つのコンポーネントから構成されます。各コンポーネントについて説明していきます。

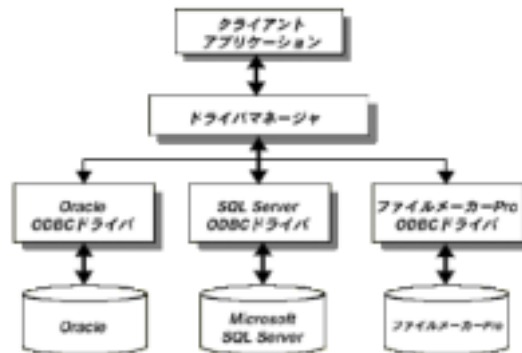


図 1.7: ODBC の構図とコンポーネント

1. ODBC データソース

データソースとはデータそのものの事です。関連する DBMS や、使用するプラットフォームやそのプラットフォームでアクセスする際のネットワーク情報などいくつかのパートから構成されることもあります。

2. ODBC ドライバ

ODBC ドライバは Windows OS 上では Dynamically Linked Libraries (DLL) などデバイスマネージャとデータソースの間の ODBC の機能連絡を処理します。アクセスするクライアントアプリケーションはそのデータソースにアクセスするようドライバに指示し、SQL によるリクエストはそのドライバにより適切に翻訳された後そのデータソースに渡されます。また、リクエストの結果についても同様にしてクライアントアプリケーションに返されます。

3. ODBC ドライバマネージャ

ODBC ドライバマネージャは、ODBC ドライバと同様 Windows OS 上では Dynamically Linked Libraries (DLL) などで、個々のドライバにアクセスを提供します。ドライバマネージャは、アクセスするクライアントアプリケーションの機能連絡を処理し、その結果を認識可能な適切なドライバに指示します。

4. クライアントアプリケーション

クライアントアプリケーションは、クライアント側でデータを処理するもので、このプログラムが ODBC 機能を呼び出し、様々なデータソースと連絡します。

1.4.2 ODBC によって何ができるか？

では ODBC を用いることによって何ができるのかに触れる．クライアント側から ODBC データソースに対して処理をする場合，複雑な途中の処理をあまり考慮せずにクライアントアプリケーションから DB を利用することができる．現在では非常に多くのアプリケーションが ODBC をサポートしており，そのようなアプリケーションの中にもクライアント側に対応したもの，サーバ側に対応したもの，両方に対応したものなどがあります．以下に代表例を示しておきます．

1. **SQL 問い合わせ型の DB** このタイプはサーバ側で使用され、Oracle、SYBASE、Microsoft SQL Server、Microsoft Access などが代表的です。これらはデータテーブルを ODBC に準拠したクライアントに返すことも、独自の SQL を使用するクライアントに返すこともできます．
2. **デスクトップアプリケーション** このタイプは ODBC を経由して DB から作成する書類などにデータを取り組むことができるもので、Microsoft Word や Microsoft Excel などが代表的です．
3. **Web データベースアプリケーション** このタイプは通常 Web サーバのバックエンドとして働き、SQL 型データベースからデータを取り組むもので、Microsoft Active Server Page などが代表的です．これらは DB テーブルから Web ページとして排出します．

1.4.3 ODBC の処理について

ODBC の処理は以下のような流れになります．

1. クライアントアプリケーションが ODBC ドライバマネージャに ODBC リクエストを送る
2. ODBC ドライバマネージャがこのリクエストに適合するデータソースを決める
3. ODBC ドライバマネージャがクライアントアプリケーションのリクエストを適切な ODBC ドライバに渡す
4. ODBC ドライバが ODBC リクエストを受け取り、これをアクセス先が解読可能なネイティブデータベースリクエストに翻訳する
5. サーバ側のデータベースが ODBC ドライバ、ODBC ドライバマネージャを通じて、このリクエストの処理結果をクライアントに返す

このようにいくつかのコンポーネントに仕事を割り振ることで、各アプリケーションはつなぐコンポーネントとのインタフェースを考えることでデータベースアクセスできる．またさまざまなアプリケーションで ODBC がサポートされているため、標準化して使用できることが ODBC の特徴です．

1.5 JAVA によるデータベース接続

Java と DB を接続する場合 JDBC(Java Database Connectivity) と呼ばれるデータベース API を用います．JDBC も ODBC と同じく CLI(Call Level Interface) の規格に準拠したライブラリです．機能的には ODBC とさほど変わりませんが、特定の DBMS に依存しないアプリケーションが作成できるという特徴があります．役割としてはアプリケーションが DBMS に SQL ステートメントを送信し、その実行結果を受け取るために使用するクラスやインタフェースを提供しています．

1.5.1 JDBC の構造

JDBC では、大きく分けてアプリケーション、ドライバマネージャ、ドライバ、そしてデータソースの 4 つのコンポーネントから構成されています．

1. アプリケーション

アプリケーションの主な役割は、JDBC のメソッドを呼び出して DB に SQL ステートメントを送り、処理結果を

受け取ることです。

2. ドライバマネージャ

ドライバマネージャはアプリケーションに代わって適切なドライバをロードします。ここではデータベースとドライバの適切な対応付けを行います。

3. ドライバ

ドライバの主な役割はマネージャから来る JDBC メソッドに対するコールを処理して SQL ステートメントを、適切なデータソースに送り、処理結果をアプリケーションに送ることです。ドライバには3つの種類があります。ひとつが Java を部分的に用い、データベースベンダから提供されるライブラリを用いてドライバが構成されています。2つ目が DB とのコミュニケーションに必要なソフトウェアがすべて Java でかかれているため、移植性が非常に高いドライバです。最後のひとつはブリッジドライバと良ければ、ODBC などのほかの CLI に準拠した API との橋渡しをするドライバです。

4. データソース

データソースは ODBC で説明したようにデータそのものです。

JDBC のさまざまな種類をまとめたものを図 1.8 に示します。

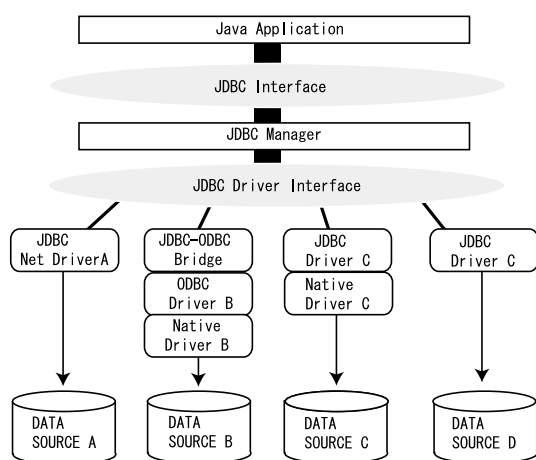


図 1.8: JDBC と 4 つのコンポーネントの関係

1.5.2 JDBC の役割

JDBC には、アプリケーションを DB に接続して問い合わせを行い、結果を受け取るためのさまざまなメソッドが用意されています。つまり JDBC はデータベースドライバと Java アプリケーションとの間のインタフェースなのです。最後に JDBC を用いることによるメリットをあげます。

- JDBC のコードをクライアントに置くため、あらゆる DBMS から完全に独立になります。
- JDBC を使用することによって、DBMS ドライバにあらゆる種類のクエリを送信することができる。そのためアプリケーションは SQL のすべての機能を利用することができる
- JDBC を用いることによって Java とリレーショナルデータベースへの問い合わせ言語である SQL の間の障壁が、ほとんどなくなります。
- 一般的な SQL レベルの API(通常は ODBC) を基礎としてインプリメントされることが多い。

第2章 DBソフトの種類について

2.1 PostgreSQL

PostgreSQL は多くのプラットフォームで動作するリレーショナルデータベース管理システムです．基本的な SQL 言語が使える，膨大なデータの検索にも十分耐えられる機能を持った本格的な RDBMS として，多くのユーザに評価されています．また，ソースコードも公開されているため，インターネット上の多くの開発者により常に改良，バグフィックスが行われていますので，サポート情報も多く安心して利用することができます．日本でもメーリングリストなどを通じて活発にユーザ間のコミュニケーションが行われています．

2.1.1 ライセンス

PostgreSQL は無料で利用でき，しかも完全なソースコードが公開されているソフトウェアです．個人での使用はもちろん，商利用も可能で，商品に組み込んで販売することもできます．ただし，PostgreSQL はいわゆる PDS ではありません．利用にあたってはライセンス条件を厳守する必要があります．

2.1.2 日本語対応

もともとは日本語のことはまったく考慮されおらず，EUC コードを使えば，たまたま日本語データの入出力が可能であるという程度でした．そこで，既存の日本語正規表現ライブラリの流用をやめ，オリジナルの PostgreSQL のコードを改造することにしました．これにより，ライセンス問題を解決するとともに，日本語，中国語，韓国語をはじめ，世界の多くの言語を使えるようになりました．日本語以外の言語も扱えるようになったので「日本語パッチ」という呼び方をやめ，この改善を「マルチパッチ」と呼ぶようになりました．

2.1.3 性能

- FreeBSD, NetBSD, BSD/OS などの数多くのプラットフォームで稼動
- c, c++, perl などの多様なプログラミングインタフェースをサポート
- クライアント/サーバアーキテクチャ
 - － クライアント用のライブラリが軽量になるので，アプリケーションプログラムのサイズが減少する．
 - － データベースエンジンのプログラムが変わっても，クライアント側が影響を受けにくくなる．
 - － ネットワーク対応である．
 - － クライアントとサーバが異なるプラットフォームでもかまわない．
- ユーザ定義関数
 - c 言語，SQL 言語を使ってユーザ定義の関数を作る事が出来る．
- オブジェクト指向機能

PostgreSQL は伝統的なリレーショナルデータベースに，オブジェクト指向の機能をプラスした新しいタイプのオブジェクトリレーショナルデータベースと言われるモデルをサポートします．

2.2 MySQL

MySQL は、スウェーデンの T.c.X.DataKonsalt AB により開発され、ソースコードがネット上で配布されています。

MySQL は多くの UNIX システム、その他のプラットフォームで動作し、その処理能力は他のデータベースにひけをとみません。MySQL は日本では最近になって少しずつ使用されはじめましたが、まだその他のデータベースに比べて知名度はあまりないようです。しかし 1 度 MySQL を使用し始めると、その快適さにあともどりはできなくなるでしょう。

2.2.1 ライセンス

クライアントコードは Public Domain か GPL になっており、クライアントコードを市販プログラムに含めたとしても、代価を払う必要はありません。

サーバ本体を売る場合、あるいは、製品に MySQL のサーバが含まれる場合、TeX にライセンス代を払う必要があります。一部の商用データベースとは違い、サーバにアクセスするクライアントの数は無関係です。あくまでも売られるサーバの数で判断されます。

2.2.2 日本語対応

MySQL のデータ保存構造はもともと 8 ビットクリーンで設計されています。よって、日本語データに入れようとして入れられない事はありません。しかし、OS によって以下のような違いがあります。

OS	コード
UNIX	EUC
MS Window , Machintosh	SJIS

2.2.3 性能

- ・ 早い DB エンジン
- ・ Solaris,SunOS 等の数多くのプラットフォームで動作している
- ・ マルチスレッドを使用している
- ・ メモリリークがない
- ・ c,c++,Java などの多様なプログラム言語のインタフェースが用意されている

2.3 Oracle

少し前まではツリー型やネットワーク型構造のデータベースなどのモデルが作成/実行されていましたが、最近ではデータベースと言えば一般にリレーショナルデータベースを指します。今現在主流のデータベース構造は RDB であり、その最も代表的なソフトが Oracle です。Oracle は、Oracle 社が開発したリレーショナルデータベース管理システムで、他社の製品に先駆けてパソコンサーバ上で稼働した RDBM です。現在オープン系のデータベース市場において、トップクラスのシェアを誇っています。また、性能や機能もトップクラスです。

2.3.1 ライセンス

Oracle データベース製品のライセンス

Oracle のライセンスは、「同時ユーザ数」と「クライアント数」という 2 つの概念から構成されています。これはサーバ側にインストールするデータベースと、それに接続するためのクライアントモジュールから構成されているためです。

「5 同時ユーザ/10 クライアント」、「10 同時ユーザ/100 クライアント」、「25 同時ユーザ/50 クライアント」などがあります。

接続ユーザ数とインストールユーザ数

2 つ目に気をつけておかなければならないことはデータベースの同時ユーザ数とそのデータベースをインストールするサーバのユーザ数との関係です。

WindowsNT サーバの Oracle からデータを取り出すときには、Oracle にログインする必要がありますが、WindowsNT にログインする必要はありません。つまり、WindowsNT という OS の資源をしようしていないということです。ライセンスに関しては Oracle のものだけ気を使えばいいでしょう。

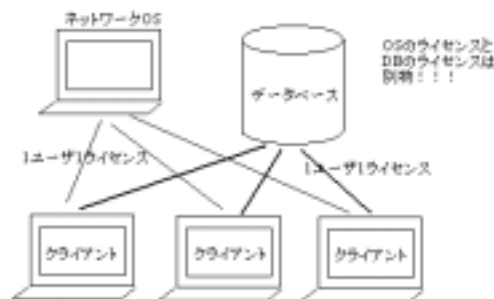


図 2.1: ライセンス

2.3.2 日本語対応

Oracle は選択可能な言語の数は実に 30ヶ国にも及ぶ。つまり日本語対応はされています。ただし、OS でつかわれている言語を選択しないと正しく表示されません。

2.3.3 性能

- ・ マルチプラットフォーム
- ・ シェアが高い
- ・ 拡張性
- ・ ディスク I/O バランスを図るために、表や牽引ごとに格納先のディスクドライブを指定できます。またフラグメンテーション防止のために、あらかじめ領域を表や牽引に割り当てることができます。メモリの使用方法についても細かく指定できるので、ハードウェア性能を極限までひきだすことができます。

- 信頼性

つねに変更ログを取っているので、たとえマシンがダウンしても、再起動したときには自動的に正常な状態まで回復します。ほかにもさまざまな信頼性のための機能が搭載されています。「時間ベースの回復」や「ログファイルベースの回復」などです。

- クライアントサーバ型

クライアントサーバ型データベースでは、サーバ側でプログラムが稼動し、データ処理はサーバが行います。必要最低限のネットワークトラフィックしか発生しないので、サーバマシンの能力を十分に活用することが出来ます。

2.4 SQL Server

SQL Server は Microsoft 社が開発した関係データベースサーバの代表で、ローカルエリアネットワーク、クライアント/サーバコンピューティングといった多くの技術を取りこんだ関係データベースサーバの代表格です。

2.4.1 ライセンス

SQL Server はフリーソフトウェアではありませんので利用する際は代価を払う必要があります。ライセンスモードとして以下のようなモードがあります。

- 同時使用ユーザ数ライセンス
単数の SQL Server にのみアクセスできるライセンス
- 接続クライアント数ライセンス
複数の SQL Server にアクセスするライセンス

2.4.2 日本語対応

SQL Server は日本語対応されています。

2.4.3 性能

- Access との融合
Access 2000 と SQL Server がより強力に統合されたことによって、Access の使いやすいインターフェイスを通じて SQL Server の高い性能と機能を活用することができます。
- サーバクライアントシステム
 - － 検索などの処理速度がアップする
 - － 同時に複数のユーザが利用できる
 - － バックアップが定期的に自動で行える
 - － 万が一損傷しても復元できる
- プラットフォームは Windows98, Windows95, Windows2000professional, WindowsNT workstation4.0, WindowsNT workstation3.51 に限定されます。

2.5 まとめ

今までに挙げた 4 つのデータベースサーバについてまとめます。

	ライセンス	日本語対応	特徴
PostgreSQL	フリー	○	機能が充実
MySQL	フリー	○	速い
Oracle	商用	○	シェアが高い
SQL Server	商用	○	Office との融合

表 2.1: データベースサーバの比較

第3章 データベースの使い道

ここでは、データベースが実際にどのような場面において、活用されているか紹介します。

3.1 企業での使い道

利用目的

- それぞれの社員の予定や会議をまとめたい。
- 営業日誌などをもっと活用し、業務の効率化をはかりたい。
- 社員間のコミュニケーションを活性化させたい。 など

これらの事を目的として、次のようなデータベースが構築されています。

- 在庫管理システム
物流を管理するため、データベースの更新がしばしば行われます。また、在庫状況のデータ参照も行われます。
 - スケジュール管理システム
スケジュールは、変化する物なので、データベースの更新、参照の両方が行われます。
 - 掲示板
掲示板に寄せられた情報をデータベース化する事により、必要な情報の検索が可能になります。
 - 顧客管理システム
 - 設備予約管理システムなど
- どのデータベースの場合も、誰かが更新のため書き込みを行っている時、ほかの人は書き込みを出来ません。

3.2 生活の中での使い道

身近な生活で活用可能で、実際に構築されているデータベースを紹介します。

私たちの生活に有効な情報を簡単に集められるデータベースがたくさん存在しています。

- 図書検索システム
ユーザーは参照するためにこのデータベースを使います。管理者側のみが、書籍の追加、削除などの更新を行います。
- 郵便番号検索システム
一度、データベース化されれば、めったに更新される事はありません。ほとんど、ユーザーが参照するために利用されます。
- 辞書や翻訳のシステム
ユーザは、検索・参照のみ可能となります。追加、削除などの更新は、管理者側のみが行います。
- 映画、テレビ検索システム など

3.3 英辞郎について

データベースソフトの中から，Web 上でも認知度の高く，生活の中でも活用出来る，英辞郎について詳細を示します．

3.3.1 英辞郎とは

英辞郎とは，プロの翻訳者・通訳者で構成されるグループ（EDP）¹が制作する英和辞書データの名称です．英辞郎の他に辞郎シリーズとして，和英辞書の「和英辞郎」，略語辞書の「略語郎」，擬音語・擬態語辞書の「音辞郎」があります．英辞郎単体そのものを英辞郎と呼んだり，辞郎シリーズを総称して英辞郎と呼んだりすることがあります．

英辞郎には，一般的な単語はもちろんのこと，スラング，イディオム，ビジネス用語，経済用語，法律用語，特許用語，コンピュータ用語，科学技術用語，医学用語，固有名詞（組織名・企業名・人名・国名・映画名）などが含まれています．英辞郎は本来，翻訳・通訳の仕事に必要なものとして個人的趣味で作っていたものですが，パソコン通信で知り合った人たちの要望から，@nifty の英会話フォーラムのデータライブラリを通じて無料で配布されています．

また，アルク社が開発した検索システム「英辞郎 on the Web」にアクセスすれば，インターネット上で無料で検索することができます．ここでは，この「英辞郎 on the Web」を紹介します．



図 3.1: 英辞郎の検索画面

英辞郎の検索画面は，Fig.3.1 に示す画面です．

この画面を用いて，「花見」日本語から，英語へと翻訳します．すると Fig.3.2 に示す画面が表示されます．このように，データベースを用いて簡単に翻訳をする事が可能です．



図 3.2: 英辞郎の検索結果

3.3.2 英辞郎の利点

【利点】ネットワーク対応

TCP/IP²ベースのネットワーク上で動作可能となっています．サーバーには Linux を基本 OS に使っている
ので，ネットワークに対する親和性の高いシステムが実現されています．

¹Electronic Dictionary Project

²インターネットやイントラネットで標準的に使われるプロトコル．

【利点】 改変・拡張自由

いつでも、新しいデータを加えられるので、ユーザー側も最新のデータを得られます。

【利点】 クライアントはブラウザを使用

システムに詳しくない人でも安心です。Internet Explorer や Netscape などの Web ブラウザをクライアントに使用する事で、分かりやすいアイコンとシンプルなメニューで、視覚的にも機能が理解できる使いやすさを提供しています。また、管理もブラウザ上から行う事が出来ます。クライアントにプログラムをインストールすることなく利用できる所以、TCO³の削減に繋がります。

3.3.3 英辞郎の問題点

英辞郎は、Web 上からの検索が可能であるのは大変有効な事ではありますが、問題点もあります。

【問題点】 該当件数が 50 件を超えた場合に、先頭の 50 件だけしか表示されない。

「英辞郎 / 和英辞郎 on the Web」のユーザーは非常に多いため、該当データのすべてを表示するようにしますと、ホストコンピュータの負荷が重くなりすぎるという問題が発生するため、表示件数を制限されています。

【問題点】 検索語を入力しても検索結果がなかなか返ってこなかったり、エラーメッセージが画面上に表示されることがあります。

日々多くの人々が「英辞郎 on the Web」を利用できますが、アクセスが集中すると反応が鈍くなったり、エラーメッセージが表示される場合があります。この場合、しばらく経ってから再度検索する必要が生じます（平日の午後はアクセスが集中する傾向があります）

【問題点】 検索した結果が利用者のニーズにうまく合った物とは限らない。

ニーズに対応出来なければ、データベースとしての価値は低下してしまいます。どこまで、ニーズに対応出来るかを考えなければなりません。

【問題点】 データの更新・サーバメンテナンス

突然のデータの更新やサーバのメンテナンスなどによりサービスを予告なく停止される事があり、突然利用不可になる可能性があります。

この他にも、次の事に注意しなければなりません。

- 英辞郎シリーズの辞書データの一部または全部を第三者にコピー、配布、転載することは禁じられている。
- 個人の範囲内における英語学習・研究・翻訳などの目的だけに利用できる。
- 営利目的の場合、無料で利用することはできない。

この注意事項は、英辞郎だけでなく Web 上に存在するフリーのデータベースに当てはまる事です。利点と欠点を十分に理解して、データベースを利用しなければなりません。また、なぜオープンなデータベース環境が必要かと言えば、「必要な」データを「必要な」形式で、「必要な」ときに「必要な」人に渡すための物と言う事も忘れてはいけません。

³ コンピュータシステムの導入、維持・管理などにかかる費用の総額。

関連図書

- [1] 堀田倫英, 石井達夫, 広川類著 『PHP 徹底攻略』(ソフトバンクパブリッシング, 1999 年)
- [2] 清水省悟著 『はじめる SQLserver』(アブライドナレッド発行, 1998 年)
- [3] A. ナス著 『SQL Server ガイド』, (トッパン発行, 1995 年)
- [4] 石井達夫著, 『PostgreSQL』((株) 技術評論社発行, 1999 年)
- [5] 山田精一著 『Oracle8』((株) アスキー発行, 1998 年)
- [6] 片岡巖発行 [データベース活用法] 『Software Design』165 号 ((株) 技術評論社)
- [7] Bernard Van Haecke 著 夏目 大訳 『JDBC パーフェクトガイド』(1998 年 (株) コンピュータエージ)
- [8] 英辞郎 (<http://www.alc.co.jp/eijiro/>)
- [9] 日本 PostgreSQL ユーザー会 (<http://www.jp.postgresql.org/>)
- [10] SKY BORD (<http://www.tms-px.co.jp/sky/sky.html>)