PostgreSQL

Chapter 1

Introduction
PostgreSQLの世界へようこそ

PostgreSQL

1-1 PostgreSQL誕生の経緯

PostgreSQLはpostgresというデータベースに，いろいろな人たちがさまざまな改良と機能追加を行うことによって生み出されました．ここではPostgreSQLの「ご先祖様」であるpostgresから始まって，PostgreSQLが誕生するまでの過程を順を追って見ていくことにします．

1.1.1 postgres

postgresは，カリフォルニア大学バークレー校（UCB）において，Michael Stonebraker教授の率いるデータベース研究プロジェクトの中で生まれました．UCBといえば，BSD版のUNIXを世に送り出したことで有名ですが，Michael Stonebraker教授の指導により，データベース研究でも大きな成果を上げています．たとえば，最も初期のRelationl Database Management System（RDBMS）[注1]であるIngresはUCBで開発されたものです．もちろんUCBで生まれたpostgresも平凡なものではあり得なく，数々の斬新な特徴を持っていました．

●注1
リレーショナルデータモデル（日本語では「関係モデル」）を採用したデータベース管理システムのこと．現在，最も広く使われているデータベースのタイプです．

①配列のサポート
②拡張可能な型システム
③ルールベースのアクティブデータベース
④大容量の不揮発性メモリ，光ディスクなどの（当時としては）新しいデバイスのサポート

このうち①と②はとくに重要で，PostgreSQLにも受け継がれている特徴なので，ここで説明します．

配列

伝統的なRDBMSで扱えるデータ型は，整数や実数，文字など，単純なものに限られます．しかし，実世界の存在をモデル化し，データベースに格納しようとすると，これだけでは不便です．たとえば，毎日の天気を「晴」「曇」「雨」「雪」のどれかの値を取るデータ型「weather」で表し，1ヵ月分のデータをデータベースに登録したいとします．伝統的なRDBMSでは，

```
create table 天気 (
        年月日 date,
        天気 weather
);
```

のようにしておいて，1レコードに1日分のデータを入れておき，1998年11月のデータが欲しければ，1998年11月1日から1998年11月31日までのレコードを取り出すようにするか（効率がよいとは言えません），さもなければ

```
create table 天気 (
        年月 date,
        天気1 weather,  -- 1日の天気
        天気2 weather,  -- 2日の天気
        天気3 weather,  -- 3日の天気
           :
           :
        天気31 weather  -- 31日の天気
);
```

のように1ヵ月分のデータを1レコードに入れるしかありません（スマートさに欠けますし，31日までない月は無駄が生じます）．しかし，postgresなら次のように書けます．

```
create table 天気 (
        年月 date,
        天気 weather[]  -- その月の毎日の天気
);
```

ここで

```
weather[]
```

は，データ型weatherの可変長配列を表します．これならある月の天気は一度のデータベースアクセスで求められますし，28日までしかない月でも無駄が出ません．

ここで注意していただきたいのは，postgresでは任意のデータ型の配列，つまり次に述べるようなユーザ定義型の配列の作成が可能だという点です．

拡張可能な型システム

すでに述べたように，伝統的なRDBMSでは単純なデータ型しか使えません．ここで，複素数をデータベース化することを考えましょう．ご存知のように，複素数は，実数部と虚数部から成ります．実数部と虚数部の値は浮動小数点で表現できますので，通常のRDBMSでは次のようになるでしょう．

```
create table 複素数 (
実数部 float,
虚数部 float
);
```

これでも一応表現できていますが，少々不便です．まず，実数部と虚数部を2個のカラムとして分けて書かなければなりません．複素数を扱うテーブルが1個ならまだ我慢できますが，数十のテーブルに同じ複素数が出てきたら嫌になりますね．データの入力もしかり．

```
insert into 複素数 values (1.0, 2.0);
```

のような感じですので，1.0が実数部なのか2.0の方が実数部なのか間違えそうです．できれば，

```
insert into 複素数 values ('1+2i');
```

のようにわかりやすくしたいものです．これらの問題は「抽象データ型」という機能を用いれば解決できます．抽象データ型とは，簡単に言うと任意のデータ型とそのデータに伴う演算をまとめて定義することです．postgresでは，ユーザが自由にデータ型を定義できます．すなわち，

- ユーザから見える入力／出力データをどのようなフォーマットにするか
- コンピュータ内部で用いるデータ表現をどうするか[注2]
- このデータ型に対する加算，減算など各種演算とオペレータ

などが定義できます．さらに，データベース特有のものとして以下の定義が可能です．

- 索引（インデックス）を定義するために必要な情報
- そのデータ型にとって効率のよい新たなアクセスメソッド[注3]

postgresの開発は1986年に始まりましたが，最初のプロトタイプはLisp言語で書かれており，冗談のようなスピードで動いたそうです．まもなくpostgresはC言語で書き直され，1988年には外部にも公開されるようになりました．

筆者が最初にpostgresに触れたのは1990年にリリースされたバージョン2ですが，このころはまだバグが多く，ずいぶん泣かされたことを覚えています．その後1992年になってバージョン4がリリースされ，性能と信頼性が大幅に向上しました．筆者がデータベースエンジンとして使っていたのは主にこのバージョンです．postgresの完成度が高まるにつれ，postgresはUCBの内外でますます多くの研究プロジェクトで採用されるようになり，ユーザ数もどんどん増えていきました．

●注2
これはC言語の構造体を定義することによって行います．

●注3
アクセスメソッドとは，データに高速にアクセスするための機構です．postgresでサポートされているアクセスメソッドには，通常使われているBtreeやhashに加え，地理情報などの2次元データを効率よく管理できるRtreeがあります．また，必要ならばユーザ定義のアクセスメソッドが作成できます．

postgresの終焉

しかし，postgresが有名になり，ユーザ数が増えるにつれてサポートの負担が重くのしかかるようになりました．もともとpostgresは研究用のデータベースであり，研究で得られた理論を実証するのが本来の目的です．postgresの維持自体が目的になるのは本末転倒と言えます．また，postgresは相次ぐ機能追加によって巨大にふくれ上がり，保守するのが困難になっていました．これらの理由から，postgresの開発はバージョン4.2を最後に打ち切られてしまいました．

1.1.2 postgres95とPostgreSQL

postgres95

このまま終わればPostgreSQLもこの世に存在することはなかったわけですが，再びpostgresを復活させたのが当時UCBの大学院生であり，postgresプロジェクトにも参加していたAndrew YuとJolly Chenです．彼らはpostgresに大胆な改造を加え，

実用的なデータベースソフトウェアとして生まれ変わらせました．これがPostgreSQLの原型になるpostgres95です．postgresに加えられた改良点を以下に列挙します．

- 問い合わせ言語をPOSTQUELから標準のSQLに変更
- ソースコードの構造の見直しと大幅な改良

  その結果，プログラムサイズは25％も小さくなり，見通しがよくなりました．また，BSD makeの代わりにGNU makeが使われるようになり，移植性が向上しました．
- ANSI C準拠のCへの書き直し
- 性能の向上：postgresに比べ，30～50%のスピードアップ
- Tcl/Tkインターフェースの追加
- ライセンス条件がより緩やかになり，商用利用が可能となった

Illustraの設立とPostgreSQL

さて，postgresから生まれたのはpostgres95だけではありません．postgresプロジェクトの中心的人物だったMichael Stonebraker教授は，postgresを商用化すべく，Illustraという会社を設立しました．Illustra社はpostgresに改良を加え，Illustraデータベースとして製品化しました．のちにIllustra社はInformix社に買収され，Informix Universal serverという製品に統合されました．

一方，postgres95はAndrew YuとJolly ChenがUCBを卒業後，開発から手を引いたために危機を迎えました．そこでpostgres95の開発を引き継いだのが，カナダ在住のMarc G. Fournier氏を世話役とする現在の開発チームです．

1996年になると，postgres95という名前がふさわしくなってしまったので，1997年1月のバージョン6.0からpostgres95はPostgreSQLという名前に変更されました．つまり，PostgreSQLのバージョン番号は6.0からスタートしたわけですが，これは，postgresの最後のバージョンが4.2であり，postgres95を5.xとみなせば，PostgreSQLは6.0にあたる，ということではないかと筆者は解釈しています．そこで本書では安定バージョンの6.3.2を中心に解説しますが，必要に応じて本書執筆時点の最新バージョンである6.4についても触れることにします．

ここで，PostgreSQLのバージョン番号について説明します．小数点以下1桁目がメジャーバージョン番号で，小数点以下2桁目がマイナーバージョン番号と呼ばれます．たとえば，バージョン6.3.2なら，メジャーバージョン番号が3で，マイナーバージョン番号が2となります．メジャーバージョン番号が上がると，機能に大きな変更が加わり，基本的にデータやプロトコルの互換性がなくなります[注4]．マイナーバージョ

●注4
既存のデータを移行するためのツールは提供されます．

ン番号の変更は主にバグ修正のために利用されます．したがって，マイナーバージョン番号の大きいバージョンほど安定していると言えます．

1.1.3 日本語対応から国際化対応へ

マルチバイトパッチ

当然ながら，PostgreSQLは，もともとは日本語のことはまったく考慮されておらず，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．PostgreSQLには正規表現検索機能があります．これは，LIKE検索をより強力にしたようなもので非常に便利な機能です．そこで，筆者は既存の日本語対応の正規表現ライブラリを組み込むことにより，正規表現検索に日本語を使えるようにしてみました．その後，メーリングリストのメンバである前田氏が，テーブル名やカラム名に日本語を使えるようにするパッチ[注5]を作成されました．これらの改造はまとめて「日本語パッチ」として，メーリングリストを通じて配布されるようになりました．ただし，このパッチには以下のような問題点がありました．

- 使用している正規表現ライブラリのライセンス条件がGPL[注6]であり，PostgreSQLとは合わない．そのため，日本語パッチを正式にオリジナルソースに取り込むことができない．
- 日本語しか使えないので，日本以外の非英語圏では適用できない

そこで，筆者は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，この改造を「マルチバイト[注7]パッチ」と呼ぶことにしました．マルチバイトパッチでは，LIKEも正しくマルチバイト文字が処理されるようになりました．このパッチは，PostgreSQLのバージョン6.3.1からオリジナルソースに取り込まれたので，今では改めてパッチを当てる必要がなくなりました．

日本語版ODBCドライバ

ODBCとは，Open Data Base Connectivityの略で，主としてPC上のアプリケー

●注5
パッチとは，オリジナルのソースコードと，変更部分の差分を取り出したものです．改造，バグ修正などをインターネットで配布するために広く使われている方法です．

●注6
Free Software Foundationの定めるライセンス条件のことです．

●注7
日本語などの非ラテン文字は，1文字を複数のバイトで表現することが多いため，こう呼んでいます．

ションがデータベースに接続するための規約です．利用したいデータベースに対応したODBCドライバを利用することにより，ExcelやAccessなどからもデータベースが使えるようになります．

PostgreSQLに付属のODBCドライバでは英語しか使えませんが，メーリングリストのメンバーの片岡氏が日本語版を作成／維持されています．片岡氏の改造により，日本語が使えるようになっただけでなく，オリジナルのODBCドライバのバグも修正されています．ぜひ，このODBCドライバをお使いになるようにお勧めします．

インターネット上では

```
http://www.interwiz.koganei.tokyo.jp/software/PsqlODBC/index.html
```

から取得できます．また，片岡氏のご厚意により本書付属CD-ROMにも日本語ODBCドライバを収録しました（packages/ODBC）．

PostgreSQLの最新事情

■6.4.2登場！

1999年1月2日，最新バージョンの6.4.2がリリースされました．6.4.2は6.4に対してバグ修正バージョンという位置付けなので，基本的に機能の追加はありません．6.4.2は6.4で作成したデータベースがそのまま使えますので，移行は容易だと思います．

■CobaltQube，NetBSD/macppc，NetBSD/m68kへの移植

最近話題のCobaltQube，それに新旧Mac用のNetBSDであるNetBSD/macppcおよびNetBSD/m68kに6.4.2が移植されました．このほか，6.4.2の各種バグ修正などを含めたパッチ適用済みソースが

```
ftp://ftp.sra.co.jp/pub/cmd/postgres/6.4.2/patches
```

で公開されています．なお，本コラム執筆時点（1999年2月中旬）のsnapshotが本書付属のCD-ROMのpgsql/に収録されてます．

■開発中の6.5

開発中の次期バージョン6.5では，今までになかったほどの大幅な性能強化が行われる見込みです．目玉はMVCC（Multi Versioning Concurrency Control）という機能です．6.4以前では更新処理を行うトランザクションがあると，テーブル全体がロックされ，他のトランザクションが待たされるのが大きな弱点でしたが，6.5では同じ行を更新する場合でない限り，複数のトランザクションを同時に実行することができます．またオプティマイザ（問い合わせの最適化を行う部分）の見直しが行われ，最高1300倍の高速化が達成されています．

もちろんこれは極端な例ですし，開発中のバージョンの話ですから正式版が出てみないと実際のところはわからないのですが，それにしても楽しみなことです．

1-2 PostgreSQLの特徴

本節ではPostgreSQLの特徴を説明します．

1.2.1 SQL92のサポート

PostgreSQLはデータベースの標準言語であるSQL92をサポートしています．トランザクションはもちろん，副問い合わせ，主キーなどのSQL92の重要な機能のほとんどをサポートします．表1.2.1にSQL92への対応状況を示します．

表1.2.1 PostgreSQLのSQL92サポート状況

◆スキーマ定義関係

CREATE SCHEMA	×
DROP SCHEMA	×
CREATE DOMAIN	×
ALTER DOMAIN	×
DROP DOMAIN	×
CREATE TABLE	○
ALTER TABLE	○
DROP TABLE	○
CREATE VIEW	○
DROP VIEW	○
CREATE ASSERTION	×
DROP ASSERTION	×
CREATE CHARACTER SET	×
DROP CHARACTER SET	×
CREATE COLLATION	×
DROP COLLATION	×
CREATE TRANSLATION	×
DROP TRANSLATION	×
GRANT	○
REVOKE	○

◆データ操作文

INSERT	○
DELETE	○
UPDATE	○

◆カーソル関係

DECLARE TABLE	×
DECLARE CURSOR	○
ALLOCATE CURSOR	×
OPEN	×
SELECT	○
FETCH	○
CLOSE	○

◆トランザクション関係

SET TRANSACTION	×
SET CONSTRAINTS	×
COMMIT	○
ROLLBACK	○

◆コネクション関係

CONNECT TO	×
SET CONNECTION	×
DISCONNECT	×

◆セッション関係

SET CATALOG	×
SET SCHEMA	×
SET NAMES	×
SET SESSION AUTHORIZATION	×
SET TIME ZONE	○

◆データ型

INT[EGER]	○
SMALLINT	○
DEC[IMAL] [(precision [,scale])]	△*
NUMERIC [(precision [,scale])]	△*
FLOAT [(precision)]	○
REAL	○
DOUBLE PRECISION	○
CHAR[ACTER] [(length)]	○
N[ATIONAL] CHAR[ACTER] [(length)]	×
[VAR]CHAR[ACTER] VARYING (length)	○
N[ATIONAL] CHAR[ACTER] VARYING (length)	×
BIT [(length)]	×
BIT VARYING (length)	×
DATE	○
TIME	○
TIMESTAMP	○
INTERVAL	○

*) scale=0固定

◆論理演算

AND	○
OR	○
NOT	○

◆特殊なオペレータ

BETWEEN	○
EXISTS	○
IN	○
IS NULL	○
LIKE	○
MATCH	×
UNIQUE	×

◆算術演算

加算(+)	○
減算(-)	○
乗算(*)	○
除算(/)	○

◆比較

=	○
<>	○
<	○
>	○
<=	○
>=	○

◆文字列の結合

||	○

◆スカラー関数

BIT_LENGTH ()	×
OCTET_LENGTH()	6.4から
CHAR[ACTER]_LENGTH()	○
LOWER ()	○
UPPER ()	○
POSITION(string IN string)	○
SUBSTRING(x FROM y FOR z)	○
TRIM(xxx pad FROM)	○
EXTRACT (x FROM y)	○
CAST (AS ¦ Domain)	○
CASE WHEN ... ELSE ... END	×
TRANSLATE (USING)	×
COLLATE	×
CONVERT	×
OVERLAPS	×

◆USER関数

CURRENT_USER	○
USER	×
SESSION_USER	×
SYSTEM_USER	×

◆日時関数

CURRENT_DATE	○
CURRENT_TIME	○
CURRENT_TIMESTAMP	○

◆集合関数

AVG	○
COUNT	○
MAX	○
MIN	○
SUM	○

◆限定子

ANY	○
SOME	×
ALL	○

◆制約

UNIQUE	○
PRIMARY KEY	○
FOREIGN KEY	×
CHECK	○
NOT NULL	○

◆その他

GROUP BY	○
HAVING	△*
UNION	○
EXCEPT	×
INTERSECT	×

*）6.4から利用可能

1.2.2 無償利用＆ソースコードの公開

PostgreSQLは無償で利用でき，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，PostgeSQLはいわゆるパブリックドメインソフトウェアではありません．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．ライセンス条件はソースに付属のCOPYRIGHTというファイルに書かれています（リスト1.2.1）．

●リスト1.2.1 PostgreSQLのライセンス条件

```
PostgreSQL Data Base Management System (formerly known as Postgres, then
as Postgres95).

Copyright © 1994-7 Regents of the University of California

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose, without fee, and without a written agreement
is hereby granted, provided that the above copyright notice and this
paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING
LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS
DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE.  THE SOFTWARE PROVIDED HEREUNDER IS
ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
```

1.2.3　数多くのプラットフォームで稼働

PostgreSQLはほとんどのUNIXプラットフォームに移植されています．現在のところ，メーリングリストなどを通じてPostgreSQLの稼働が確認されているプラットフォームを表1.2.2に示します．

なお，クライアント用のライブラリのみに限定すれば，以下のプラットフォームでも利用が可能です．

- NEXTSTEP
- Win32（Windows）注1

●注1
6.4からのサポートになります．

筆者が試した限りでは，MachTen（MacOS上で動くUNIX）でもクライアントライブラリを動作させることができました．

●表1.2.2　PostgreSQLの稼働するプラットフォーム

FreeBSD
NetBSD
OpenBSD
BSD/OS
Linux ELF：Intel x86，Sparc/Linux，MkLinux，LinuxPPC
SunOS 4.1.x
Solaris 2.4/2.5/2.6：SPARCとx86アーキテクチャの両方をサポート
HP-UX 9.0/10
IRIX 5.3
IBM AIX

1.2.4 多様なプログラミングインターフェースをサポート

PostgreSQLは，標準で以下のプログラミング言語をサポートします．

- C
- C++
- Tcl/Tk
- Perl
- Java（タイプ4 JDBC ドライバ）
- Python
- ecpg（C 言語にSQL を埋め込むためのプリプロセッサ）

サードパーティからも各種インターフェースが提供されています．以下に代表的なものを挙げておきます．

●PHP/FI（Personal Home Page Construction Kit/Form Interpreter）

フリーで配布されているHTMLファイルに埋め込んで使うスクリプト言語です．バージョン3.0からは単に「PHP」と呼ぶようになりました．1次配布元は

```
http://www.php.net/
```

ですが，以下に挙げた日本語によるサポートページを利用すると便利でしょう．

```
http://www.cityfujisawa.ne.jp/%7Elouis/apps/phpfi/
```

●ruby

まつもと ゆきひろ氏が作成したオブジェクト指向スクリプト言語です．PostgreSQLへのアクセス用モジュール[注2]はまつもと えいじ氏が作成され，以下のURLで公開されています．

```
ftp://ftp.netlab.co.jp/pub/lang/ruby/contrib/postgres-0.5.tar.gz
```

なお，ruby本体は

```
http://www.netlab.co.jp/ruby/jp/
```

から入手することができます．

●注2
モジュールが新しくなった場合はバージョン番号（0.5）がより大きなものになりますので，そちらを取得してください．

1.2.5　クライアント／サーバアーキテクチャ

データベースを利用するクライアントと，データベースエンジンを提供するサーバが完全に独立しています．このことにより，以下のメリットがあります．

- 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．

1.2.6　日本語化＆国際化対応

コンパイル時に指定することにより，以下の文字コードのうちの1つを使用することができます．

①各国のEUC（Extended Unix Code）
- 日本語
- 中国語
- 中国語（台湾で使用されているもの）
- 韓国語

②Unicode（UTF-8）
③マルチリンガルエディタMuleの内部コード

とくに②または③を使用した場合，各国語を混在して使用できるようになります．
なお，6.4ではさらに以下の機能が追加されます．

q 使用する文字コードがコンパイル時ではなく，データベース生成時に決定できる
w サーバ側とクライアント側で異なる文字コードが使用できる
e クライアント側の文字コードにSJISが利用可能
r SQL92の文字列関数character_length(), position(), substring() でのマルチバイトサポート

t 英語以外のラテン系言語ISO-8859-1から5までのサポート

このうちwerは，6.3.2リリース後，筆者がβ版として発表したパッチの中に含まれています．第2章ではこのパッチも適用します．

1.2.7 ユーザ定義関数

PostgreSQLでは，C言語，SQL言語を使ってユーザ定義の関数を作ることができます[注3]．関数を定義するには，

```
create function 関数名(引数型,...)
    returns データ型
    as '関数本体'
    language '言語';
```

の構文を使います．PostgreSQLでは，引数の数／型が異なれば，同じ名前を持つ関数を複数定義できます．関数本体はSQL関数の場合にはSQL文，C言語の場合には関数をコンパイルした結果のオブジェクトファイル名です[注4]．このオブジェクトファイルはデータベースエンジンが動的にロードします．作成した関数は，システム組み込みの関数とまったく同様に扱うことができます．実は，ユーザ定義のデータ型も，この機能を使っています．

ユーザ定義関数は非常に有用な機能で，とくにC言語のユーザ定義関数を使えば，ほとんど何でもできます．それだけに，使い方を誤るとデータベースエンジンがクラッシュしたり，セキュリティホールになる可能性があるので，注意が必要です．

●注3
これ以外にTcl/Tkのユーザ定義関数があります．また6.4からPL/pgSQLという制御構造を記述できる，ユーザ定義関数SQL言語によるユーザ定義関数機能が追加されました．

●注4
正確にはシェアードライブラリ（shared library）になっている必要があります．

1.2.8 リーレショナルデータモデルを超えるオブジェクト指向機能

PostgreSQLは伝統的なリレーショナルモデルベースに，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（参考文献1）をサポートします．このモデルは，リレーショナルモデルにもとづくデータベースの特長であるトランザクション，大量データの高速処理，セキュリティ，といった点をそのまま維持しながら，オブジェクト指向データモデルの柔軟性，拡張性を取り入れたものです．主な特徴を挙げると

- 配列のサポート
- 拡張可能な型システム

となります．

また，PostgreSQLはオブジェクト指向モデルの特徴である「継承」もサポートします．たとえば，

```
create database 人間 (
        名前 varchar(128),
        生年月日 date,
        住所 varchar(128)
);
```

と定義したテーブルを，「社員」データベースで継承することができます．

```
create database 社員 inherits 人間 (
        社員番号 integer,
        所属部署 varchar(128)
);
```

この場合，「社員」テーブルは，「人間」テーブルを継承しているので，社員番号，所属部署のカラムに加え，名前，生年月日，住所というカラムも持つことになります．

PostgreSQLを使用するにあたっての注意点

このようにPostgreSQLはすばらしいデータベースですが，利用にあたって考慮すべき点もあります．

■無保証であること

これはどんなフリーソフトウェアにもいえることですが，PostgreSQLを利用して何か問題が起きても誰にもいっさい責任は問えません．自己責任において使ってください．

■サポート

フリーソフトウェアですから，ベンダのサポートはありません．自分で解決できない問題はメーリングリストなどで相談してください．ただ，これは必ずしも欠点とは言えません．なぜなら，ほとんどの場合，メーリングリストで相談することによりすばやく解決方法が見つかりますし，商用ソフトウェアと違って次のバージョンまでバグ修正を待つ必要もないからです．

なお，PostgreSQLの既知のバグと，対処方法については第5章をご覧ください．