Digital Huge Technology

LPI-JAPAN
Linux Professional Institute Japan

OSS-DB Exam Silver 技術解説無料セミナー

2014/5/25

株式会社デジタル・ヒュージ・テクノロジー
技術開発部　サブマネージャー
豊田 健次

自己紹介

DHT Digital Huge Technology　OSS-DB

■名前

豊田　健次　31歳

■所属

株式会社デジタル・ヒュージ・テクノロジー

技術開発部　サブマネージャ

■自己紹介

2007年、株式会社デジタル・ヒュージ・テクノロジーに入社。
以来、様々なプロジェクトへ参加し、開発業務を経験。
近年では、講師なども務めるようになり、人材育成にも力を入れている。

■最近の出来事

HTML5プロフェッショナル認定試験に合格しました。

OSS-DB 試験概要

DHT Digital Huge Technology

OSS-DB

- **出題数**
  - 50問
- **試験時間**
  - 90分（アンケート等を含むため、実質80分程度）
- **合格点**
  - 64点以上

32問/50問

- **試験会場・試験方式**
  - 全国のテストセンターにてCBT方式による受験
- **申し込み**
  - ピアソンVUEからオンライン申し込み

OSS-DB Silver 出題範囲

出題範囲

- 一般知識（16%）
  - データベースの基本的な知識
  - 一般知識、データベース設計　など
- 運用管理（52%）
  - インストール
  - バックアップ
  - 設定ファイル
  - 基本的な運用管理　など
- 開発/SQL（32%）
  - SQLコマンド
  - 組み込み関数
  - トランザクション概念　など

本日の概要

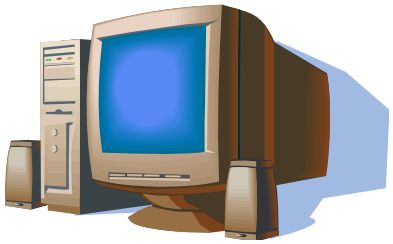
DHT Digital Huge Technology　OSS-DB

- 設定ファイル
  - -postgresql.conf
  - -pg_hba.conf
- 基本的な運用管理
  - -ユーザの追加、削除、変更
  - -VACUUM、ANALYZEの利用
  - -運用管理で必要となる情報

1時間ごとに10分程度の休憩を入れます

設定ファイル

postgresql.conf



設定ファイル

DHT Digital Huge Technology　OSS-DB

postgresql.conf　とは

PostgreSQLにおけるメインの設定ファイル
（initdbを実行すると作成される）

通常の手順でインストールした場合は、
環境変数“$PGDATA”配下に作成される

設定ファイル

$PGDATA

ソースをコンパイルしてインストールした場合は

/usr/local/pgsql

パッケージをインストールした場合は

RH系Linux→ /var/lib/pgsql

Debian系Linux→ /var/lib/postgresql

→インストールする環境や、インストールする方法によって、ファイルの場所が変化するので、実環境を作成して勉強する際には注意しましょう。

実際にファイルを見てみましょう

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

記述方法

行ごとに以下の書式で記述

“変数名 = 値” (#以降はコメントとして扱われる)

例)

```
#listen_address =  ‘localhost’
max_connections = 100
```

値には**ブーリアン**、**整数**、**浮動小数点**、**文字列**の
いずれかが入る

設定ファイル(postgresql.conf)

DHT Digital Huge Technology　OSS-DB

ブーリアンについて

ブーリアン (boolean) とは、「真」/「偽」のどちらかの状態しか持たないデータ型のこと
(スイッチのON/OFFようなイメージ)

PostgreSQLでは、デフォルトは on/off と記述しますが、
yes/no, true/false, 1/0 のような記載もOK

設定ファイル(postgresql.conf)

DHT Digital Huge Technology　OSS-DB

■文字列について

文字列は必ずシングルクォート(’)で囲む

100　　→　整数

‘100’　→　文字列

■整数について

整数には単位を用いることができる

GB　　→　ギガバイト

ms　　→　ミリ秒　　...etc

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

設定の反映

postgresql.conf を編集しても、設定はすぐに反映されない。
以下のコマンドを実行し、反映させる。

```
pg_ctl reload
```

→設定ファイルを読み込み直す

```
pg_ctl -w restart
```

→プログラムを再起動する

※PostgreSQLが起動していない状態なら、上記のコマンドは不要

設定ファイル(postgresql.conf)

設定の反映

postgresql.confの各項目のコメント部分に
「change requires restart」と記載されている項目は、
“pg_ctl -w restart”による再起動が必要

それ以外の項目については、
“pg_ctl reload”で反映

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

主な設定項目(10項目)

- listen_address
- max_connections
- silent_mode
- port
- shared_buffers
- max_files_per_process
- log_destination
- log_connections
- logging_collector
- client_encoding

設定ファイル（postgresql.conf）

DHT Digital Huge Technology　OSS-DB

listen_address（文字列）

接続を受け付けるIPアドレスを指定する。

＊ とだけ書いた場合、すべての接続を許可。

192.168.100.＊ と書いた場合、192.168.100.0～255のIPアドレスからの接続のみを許可する。

空文字（‘’）を書いた場合、外部からの接続はすべて拒否。
UNIXドメインソケットでの接続のみとなる。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

UNIXドメインソケット

PostgreSQLが動作しているサーバと同じサーバ内からの接続に用いられる通信方式。

→Webサーバとしても動作している場合などに利用

外部からの接続を想定しない場合、不要なアドレスからの接続を受け付けないように設定することが望ましい。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology

OSS-DB

max_connections（整数）

データベースへの最大同時接続数を設定する。

この設定値を超える接続を受けた場合、PostgreSQLは、エラーを返す。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

silent_mode (ブーリアン)

標準出力、標準エラー出力にログを出力するか否かを設定。

標準出力、標準エラー出力とは、サーバマシンに接続しているターミナル(ディスプレイ)を指す。

サーバとして運用する場合、ログファイルへ出力するため、通常はoffに設定する。

設定ファイル（postgresql.conf）

DHT Digital Huge Technology　OSS-DB

port（整数）

外部からの接続を受け付けるポート番号を指定する。

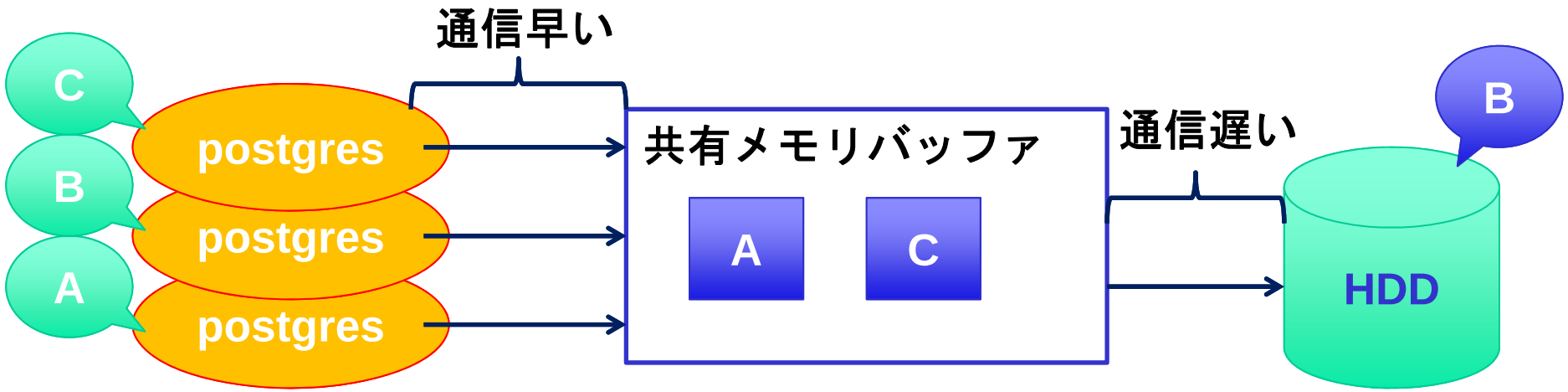
ポート番号とは、サーバ内部のどのサービスがその接続を受け付けるのかを識別するための番号。

デフォルトでは、5432番となっており、特に理由が無ければそのままの利用で問題ない。

shared_buffers（整数）

共有メモリバッファのサイズを設定。

共有メモリバッファは、ディスクから読みだしたデータを
バッファリング(一時保管)しておく領域のこと。
このサイズが大きいほど、ディスクアクセス回数が減少するため、
高速に動作する。



設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

shared_buffers (整数) 続き

割り当てたサイズに比例して速度が向上するわけではないことに注意。

最低でも128KB以上に設定する必要があり、
デフォルトは32MB

サーバがデータベース専用マシンの場合、マシンに搭載しているメモリの1/4を割り当てることが推奨。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

- max_files_per_process (整数)

  PostgreSQLが同時に開くことができるファイルの数を設定する。

  “Too many open files”というエラーが発生した場合は、この設定値を見直す必要がある。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology

OSS-DB

log_destination (文字列)

ログの出力先を設定する。

デフォルトは"stderr" (標準エラー出力)
他に"syslog"、"csvlog"を設定可能。

syslogは、Linuxのログ機能を利用してログファイルへ出力。
csvlogは、ログをCSV形式で出力。

※CSV形式は、ログが解析しやすいメリットがある。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

log_connection (ブーリアン)

クライアントからの接続をログに出力する設定。

デフォルトはoffなので、外部からの接続を受け付ける場合はonに設定しておくべき。

設定ファイル(postgresql.conf)

DHT Digital Huge Technology OSS-DB

logging_collector (ブーリアン)

syslog、csvlogをログファイルへ出力する設定。

通常運用では、ログはファイルへの出力を行うのでON
ログ収集に外部ツールを利用する場合はOFF

設定ファイル（postgresql.conf）

DHT Digital Huge Technology

OSS-DB

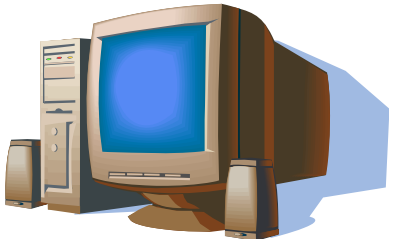
client_encoding（文字列）

サーバとクライアントで利用する文字コードが異なる場合に設定すると、自動的にクライアント側に合わせた文字コードに変換する。

この設定値は、後でコマンド等で変更可能。

設定ファイル

pg_hba.conf



設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

pg_hba.conf

クライアントの認証方法を設定するファイル。

データベースの起動時に設定が読み込まれる。
(pg_ctl reloadで反映可能)
一行ごとに認証対象と認証方式を記述する。
複数行記述した場合、上から順番に評価され、該当する行が見つかった時点でその設定が適用される。
その行の認証がNGとなった場合は、その時点で認証失敗となる。

該当行が無かった場合、接続は「拒否」となる。

設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

pg_hba.conf (一行の構成)

一行は以下の項目をスペース(タブ)で区切って記述する。

- 接続タイプ
- データベース名
- ユーザ名
- IPアドレス(範囲)
- 認証タイプ

実際にファイルを見てみましょう

設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

pg_hba.conf

記述パターンは以下の7通り

```
local      database  user  auth-method  [auth-options]
host       database  user  CIDR-address  auth-method  [auth-options]
hostssl    database  user  CIDR-address  auth-method  [auth-options]
hostnossl  database  user  CIDR-address  auth-method  [auth-options]
host       database  user  IP-address  IP-mask  auth-method  [auth-options]
hostssl    database  user  IP-address  IP-mask  auth-method  [auth-options]
hostnossl  database  user  IP-address  IP-mask  auth-method  [auth-options]
```

上記のパターンはオンラインマニュアルにも記載されている。

http://www.postgresql.jp/document/9.0/html/auth-pg-hba-conf.html

設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

接続タイプ

以下の4タイプがある。

- local

  ローカル接続(UNIXドメインソケット)に適用される。

- host

  ホスト接続(TCP/IP)で、SSL/非SSLの両方に適用される。

- hostssl

  ホスト接続で、SSL接続の場合のみ適用される。

- hostnossl

  ホスト接続で、非SSL接続の場合のみ適用される。

設定ファイル(pg_hba.conf)

データベース名 (database)

データベース名を直接指定する以外に、以下の設定が可能。
(複数指定時はカンマで区切る)

- all
  全てのデータベースが対象
- sameuser
  接続ユーザと同名のデータベースが対象
- samerole
  接続ユーザが属しているロールと同名のデータベースが対象
- @filename
  別ファイルに記載したデータベース名のリスト

設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

ユーザ名 (user)

ユーザ名を直接指定する以外に、以下の設定が可能。
(複数指定時はカンマで区切る)

- all
  全てのユーザが対象
- +groupname
  指定したグループに属しているメンバが対象
- @filename
  別ファイルに記載したユーザ名のリスト

IPアドレス

IPアドレスごとに設定が可能。(範囲指定も可)

接続タイプがlocalの場合、ここは空白にする。

記述方法が2通りある

①IP-address IP-mask

192.168.100.0　　255.255.255.0

IPアドレス　　サブネットマスク

②CIDR-address

192.168.100.0/24

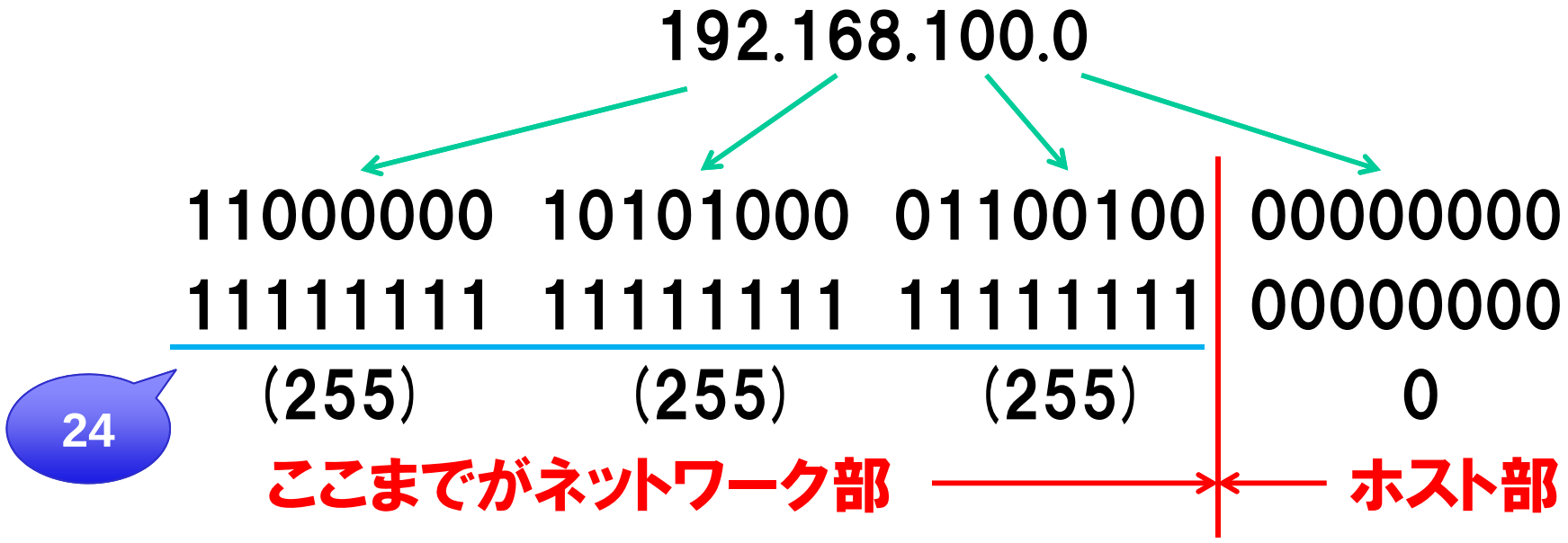
IPアドレス　　サブネットマスク

IPアドレス　続き

IPアドレスは、ネットワーク部とホスト部に分けられる。

どこまでがネットワーク部かを識別する情報

→サブネットマスク (IP-mask)



IPアドレス　続き

範囲を示す場合、ホスト部は0で埋める。

これで正しい

192.168.100.0/24

範囲ではなく特定のIPアドレスを指定する場合は、
以下のように記述する。

192.168.100.5/32

それ以外に以下の設定が可能。

samehost	PostgreSQLサーバと同一ホストからの接続
samenet	PostgreSQLサーバが接続しているネットワーク内のクライアントからの接続

設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

認証タイプ

• trust	無条件で許可
• reject	無条件で拒否
• md5	MD5暗号化パスワード認証
• password	平文パスワード認証
• gss	GSSAPI認証
• sspi	SSPI認証
• krb5	Kerberosバージョン5認証
• ident	ident認証
• ldap	LDAP認証
• radius	RADIUS認証
• cert	SSLクライアント証明書認証
• pam	PAM認証

設定ファイル(pg_hba.conf)

DHT Digital Huge Technology OSS-DB

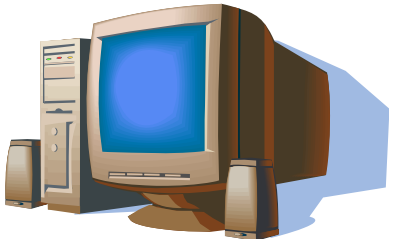
認証タイプ

認証方式は接続方式によって利用できないものや、
OSによって利用できないものがあるため注意が必要。

詳細はオンラインマニュアルを確認
http://www.postgresql.jp/document/9.0/html/auth-pg-hba-conf.html

基本的な運用管理

ユーザの追加、削除、変更



基本的な運用管理(ユーザ)

DHT Digital Huge Technology

OSS-DB

PostgreSQLにおけるユーザ

PostgreSQLでのユーザは、**オペレーティングシステムのユーザとは完全に別に管理**される。

新たにユーザを追加する場合は、オペレーティングシステムとは別に、PostgreSQL側でもユーザの追加作業が必要。

基本的な運用管理(ユーザ)

DHT Digital Huge Technology OSS-DB

■ロール

PostgreSQLでは、「ユーザ」、「グループ」という明確な分類が存在せず、「ロール」という概念ですべてを管理。

グループ:複数のユーザの権限などをまとめて設定

“ロールA(ユーザ)がロールB(グループ)に属する。”

一般的にログイン権限の有無で大別することができる。

ログイン権限を持つロール　　≒　ユーザ

ログイン権限を持たないロール ≒　グループ

基本的な運用管理(ユーザ)

DHT Digital Huge Technology OSS-DB

ユーザの確認方法

以下の2つの方法で確認が可能

＜SQL＞

```
SELECT rolname FROM pg_roles;
```

＜psqlメタコマンド＞

```
¥du
```

基本的な運用管理(ユーザの作成)

DHT Digital Huge Technology

OSS-DB

ユーザの作成

SQLによる追加と、コマンドによる追加の2通り

＜SQL＞

```
CREATE ROLE
```

SQLによりユーザを追加。

様々なオプションが用意されている。

ユーザを追加することができるのは、CREATEROLE権限を持つユーザか、スーパーユーザのみ。

基本的な運用管理(ユーザの作成)

DHT Digital Huge Technology OSS-DB

ユーザの作成

＜コマンド＞

createuser

コマンドラインからユーザを追加。
createuserを実行すると、作成するユーザに与える権限について質問される。

```
-bash-4.1$ createuser toyoda
新しいロールをスーパーユーザとしますか？ (y/n)n
新しいロールにデータベース作成権限を与えますか？ (y/n)n
新しいロールにロールを作成する権限を与えますか？ (y/n)n
-bash-4.1$
```

内部的には前述のSQL"CREATE ROLE"が実行されている。

ユーザの作成　続き

SQL同様、様々なオプションが用意されている。

ユーザを追加することができるのは、CREATEROLE権限を持つユーザか、スーパーユーザのみ。

基本的な運用管理(ユーザの作成)

DHT Digital Huge Technology OSS-DB

CREATEROLE権限について

CREATEROLE権限を持つユーザは新たにユーザを作成することが可能。

自ユーザがデータベース作成権限を持っていなくても、
データベース作成権限を持っているユーザを作成できる。
(※スーパーユーザ権限は例外)

CREATEROLE権限の付与は慎重に!

基本的な運用管理(ユーザの削除)

DHT Digital Huge Technology OSS-DB

ユーザの削除

SQLによる削除と、コマンドによる削除の2通り

＜SQL＞

```
DROP ROLE
```

SQLによりユーザを削除。

スーパーユーザを削除する場合は、自分自身もスーパーユーザの必要がある。

一般ユーザを削除するには、CREATEROLE権限が必要。

基本的な運用管理(ユーザの削除)

DHT Digital Huge Technology OSS-DB

ユーザの削除

＜コマンド＞

dropuser

コマンドラインからユーザを削除。

内部的には前述のSQL"DROP ROLE"が実行されている。

基本的な運用管理(ユーザの変更)

DHT Digital Huge Technology

OSS-DB

ユーザの変更

ユーザの変更はSQLでのみ可能。

＜SQL＞

ALTER ROLE

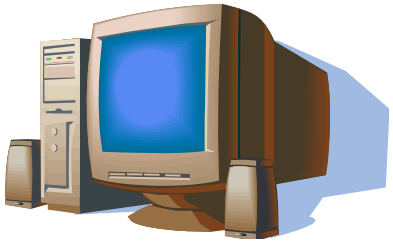
新たな権限の付与や剥奪。

後からパスワードを追加/変更する場合などにも利用。

様々なオプションが用意されている。

基本的な運用管理

VACUUM、ANALYZE



VACUUMを説明する前に...

- MVCC(MultiVersion Concurrency Control)

  複数のユーザから同時に処理を要求された場合でも、
  データの一貫性を保証し、且つ処理を平行して実施する
  ための仕組み。

基本的な運用管理(VACUUM)

DHT Digital Huge Technology OSS-DB

■追記型(マルチバージョン方式)アーキテクチャ

レコードの削除が行われても、内部的にはレコードを削除せず、見かけ上「削除した」ことにする。
更新が行われた場合も、実際には「削除」と「挿入」の組み合わせ。

更新・削除が頻繁に行われるテーブルでは、不要なデータが残り続ける。

→データベースの肥大化、パフォーマンスの低下

基本的な運用管理(VACUUM)

DHT Digital Huge Technology OSS-DB

VACUUM

以下の3つの役割を持っている。

① データベースの不要領域の回収(本当の意味での削除)
→データベースの肥大化、パフォーマンス低下の防止

② XID(トランザクションID)周回エラー防止
→詳細は後述

③ 統計情報の収集
→詳細は後述

XID周回エラー

追記型アーキテクチャ → 古いレコードを消さない

最新のレコードがどれだかわからない！
→レコードにIDを割り当て世代管理 → XID

XIDは無限に増え続けないため、いつかリセットされる。
→古いレコードのIDと混ざり、データが破損！

VACUUMの実施でそれまでのXIDを過去のものとして恒久化
現在のXIDと切り離すことができる。

■統計情報

データベースを最適な方法で検索

→データベース(テーブル)の状態を把握する必要がある。

統計情報

① 定期的な統計情報の収集を推奨
（マシンスペックや更新頻度によるが、１日１回程度）

② テーブルの内容が大幅に変更された場合も実施推奨

基本的な運用管理(VACUUM)

DHT Digital Huge Technology　OSS-DB

VACUUM

SQL、コマンドのいずれかで実行

＜SQL＞

```
VACUUM [ANALYZE] [テーブル名];
```

テーブル名を指定すれば、対象テーブルのみVACUUM実施。
未指定の場合は、全テーブルをVACUUM。

ANALYZEをつけることで、統計情報も同時に収集。

基本的な運用管理(VACUUM)

DHT Digital Huge Technology　OSS-DB

VACUUM

＜コマンド＞

vacuumdb [-z] [-t テーブル名] [データベース名]

“-z” オプションで、ANALYZEを同時実行。

内部的には、前述のSQL”VACUUM”を実行する。

基本的な運用管理(ANALYZE)

DHT Digital Huge Technology

OSS-DB

ANALYZE

ANALYZEを単独で実行することも可能

＜SQL＞

```
ANALYZE [テーブル名];
```

基本的な運用管理(自動バキューム)

DHT Digital Huge Technology OSS-DB

■自動バキューム

前述のVACUUMは手動実行の必要がある。

autovacuum機能を利用すると、更新量に合わせて、適宜VACUUM/ANALYZEを実行してくれる。

postgresql.conf にて設定が可能(デフォルトでON)

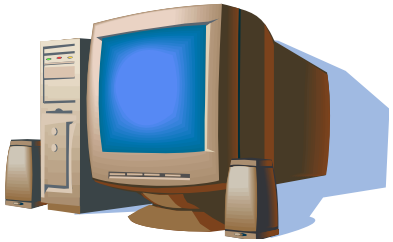
```
#------------------------------
# AUTOVACUUM PARAMETERS
#------------------------------

#autovacuum = on
```

他にも多数の設定項目により、細かい設定が可能。

http://www.postgresql.jp/document/9.3/html/runtime-config-autovacuum.html

基本的な運用管理

システム情報関数



基本的な運用管理(システム情報関数)

DHT Digital Huge Technology OSS-DB

■システム情報関数

データベースの現在の状態を確認するための手段。

主なシステム情報関数

version ()	データベースサーバのバージョン取得
pg_backend_pid ()	接続中のバックエンドプロセスID取得
inet_server_addr ()	サーバのIPアドレスを取得
inet_client_addr ()	クライアントのIPアドレスを取得
current_schema	現在のスキーマ名を取得
current_database ()	現在のデータベース名を取得
current_user	現在のユーザ名を取得

※一部のシステム情報関数は、括弧が不要であることに注意

■システム情報関数　利用方法

SELECT文と組み合わせて利用

例）

SELECT version();

```
postgres=# select version();
                                                      version
------------------------------------------------------------------------------------------------------------
 PostgreSQL 8.4.13 on x86_64-redhat-linux-gnu, compiled by GCC gcc (GCC) 4.4.6 20120305
(Red Hat 4.4.6-4), 64-bit
(1 行)

postgres=#
```

基本的な運用管理(システムカタログ)

DHT Digital Huge Technology OSS-DB

■システムカタログ (システムテーブル)

テーブルや列ごとの情報など、内部情報を格納している。
PostgreSQL独自の仕様のため、ほかのDBへは移植不可。

主なシステムカタログ

pg_class	テーブル情報を管理
pg_roles	ロール情報の表示
pg_authid	ロール情報の管理
pg_proc	関数の管理
pg_type	データ型の管理
pg_index	インデックスの管理

■システムカタログ（システムテーブル） 利用方法

SELECT文と組み合わせて利用

例）

```
SELECT * FROM pg_roles;
```

```
postgres=# select * from pg_roles;
 rolname  | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcatupdate | rolcanl
ogin | rolconnlimit | rolpassword | rolvaliduntil | rolconfig | oid
----------+----------+------------+---------------+-------------+--------------+--------
-----+--------------+-------------+---------------+-----------+-----
 postgres | t        | t          | t             | t           | t            | t
     |           -1 | ********    |               |           |  10
(1 行)

postgres=#
```

基本的な運用管理(情報スキーマ)

DHT Digital Huge Technology OSS-DB

情報スキーマ

テーブルや列ごとの情報など、内部情報を格納している。
システムカタログと類似だが、情報スキーマは移植可能。
ただし、データ量はシステムカタログに比べ少なくなる。

主な情報スキーマ

information_schema.tables	テーブル一覧
information_schema.views	ビュー一覧
information_schema.triggers	トリガー一覧
information_schema.schemata	スキーマ一覧

```
SELECT * FROM information_schema.tables;
```

■学習のポイント

とにかく実際に操作する！

- 文章だけでは理解不足になりがち。
- 実体験は忘れにくい。
- 特に運用管理は実際に操作しないと覚えない。

フリーなPostgreSQLを導入して、どんどん操作しよう！

弊社のことを少しだけ・・・

弊社では土曜日限定スクールを開催中

- LPIC Level1 ～ Level3
- OSS-DB Silver/Gold
- NetCommons

詳細は弊社HPをご確認ください。

http://www.dht-jpn.co.jp/training.php

ご清聴ありがとうございました。

■お問い合わせ■
デジタル・ヒュージ・テクノロジー　トレーニング担当
training@dht-jpn.co.jp