



# OSS-DB Exam Silver 技術解説無料セミナー

2013/9/28

日本ヒューレット・パッカード株式会社  
HP教育サービス 講師  
金 源基



## ■ 金 源基(キム ケンキ)

- ～1999年 UNIXシステムプログラミング中心に開発業務
- 1999年 外資系大手のServer事業部 システムコンサルタント
- 2000年 フリーISPのカスタマイズ部門シニアマネジャー
- 2002年 BIシステムコンサルタント
- 2008年 OSS DB及びOSS BIツール中心にコンサルタント業務
- 2013年 日本ヒューレット・パッカー株式会社  
教育サービス本部で「PostgreSQL運用管理コース」担当



- OSS-DB技術者認定試験の概要
- PostgreSQLのご紹介
- インストール方法と簡単な利用方法
- OracleデータベースとPostgreSQLのアーキテクチャの違い
- OracleデータベースとPostgreSQLのプログラミング上の違い
- HP教育サービスの研修紹介



# OSS-DB技術者 認定試験の概要



特定非営利活動法人エルピーアイジャパン(LPI-Japan)が、オープンソースデータベース(OSS-DB)に関する技術力と知識を公平かつ厳正に中立的な立場で認定するIT技術者認定資格です。

本認定制度には「Silver」と「Gold」の二つのレベルがあり、それぞれ以下のスキルを備えているIT技術者であることを認定します

## ■OSS-DB Gold

大規模データベースシステムの改善・運用管理・コンサルティングができる技術者

## ■OSS-DB Silver

データベースシステムの設計・開発・導入・運用ができる技術者



## ■ OSS-DB Gold

- ・認定条件: OSS-DB Exam Goldに合格し、かつ有意なOSS-DB Silverを保有していること
- ・認定有意性の期限: 5年間

## ■ OSS-DB Silver

- ・認定条件: OSS-DB Exam Silverに合格すること
- ・認定有意性の期限: 5年間





- 出題数 50問
- 合格点 64点
- 試験時間 90分(アンケート時間などを含む)

## ■ 試験実施方式

コンピュータを使いオンラインで50問の解答を入力します。  
マウスによる選択方式がほとんどですが、キーボード入力問題も多少出題されます。

## ■ 試験日時・会場

テストセンター(ピアソンVUE)でお好きな日時や会場を選択することができます。

- 合否結果 試験終了と同時にわかります



## ■ 一般知識(20%)

- OSS-DBの一般的特徴
- ライセンス
- コミュニティと情報収集
- データベース設計、正規化
- リレーショナルデータベースの一般知識
- リレーショナルデータモデルの基本概念
- SQL一般知識、SQL分類
- データベース管理システムの役割

## ■ 運用管理(50%)

- インストール方法
- 標準付属ツールの使い方
- 基本的な運用管理作業
- 設定ファイル
- バックアップ方法

## ■ 開発/SQL(30%)

- SQLコマンド
- トランザクション概念
- 組み込み関数

## ■ 最新の試験範囲はウェブで確認しましょう!

<http://www.oss-db.jp/outline/examarea.shtml>





# PostgreSQLの紹介



- 無償、オープンソースのリレーショナル・データベース (RDBMS) です
  - PostgreSQLライセンス
    - BSDライセンスに似ています
    - 利用、コピー、配布は自由で制限が緩くなっています
  - 最新バージョンはPostgreSQL 9.2.4 (2013年6月現在)
    - 最初の2つの数字がメジャー・バージョンになります
    - 3つめの数字はマイナー・バージョン(主にバグ修正)です
  - 対応プラットフォームが豊富です
  - 標準SQLにほぼ準拠しています
    - SQL92, 99, 2003, 2008



## ■ PostgreSQLから派生した製品群

### ■ Postgres Plus (EnterpriseDB)

- Oracle Database互換機能をはじめ複数の機能を追加しています
- 独自のツール、保守を提供しています

### ■ Greenplum Database (EMC)

- 並列ストレージを使ったデータ・ウェアハウス用データベースです

### ■ PowerGres (SRA OSS)

- 独自ツールの追加や保守を提供しています



## ■ PostgreSQL 9.2の新機能

- 性能改善
  - Index-Only-Scan
  - 並列処理性能の向上
- 機能追加
  - 同期レプリケーション
  - JSONデータ型
  - 範囲データ型
  - DDL改善
- プロセス、ユーティリティの追加
  - チェックポインター
  - pg\_receivexlogユーティリティ



# インストール方法と 簡単な利用方法



## ■ インストーラ 起動

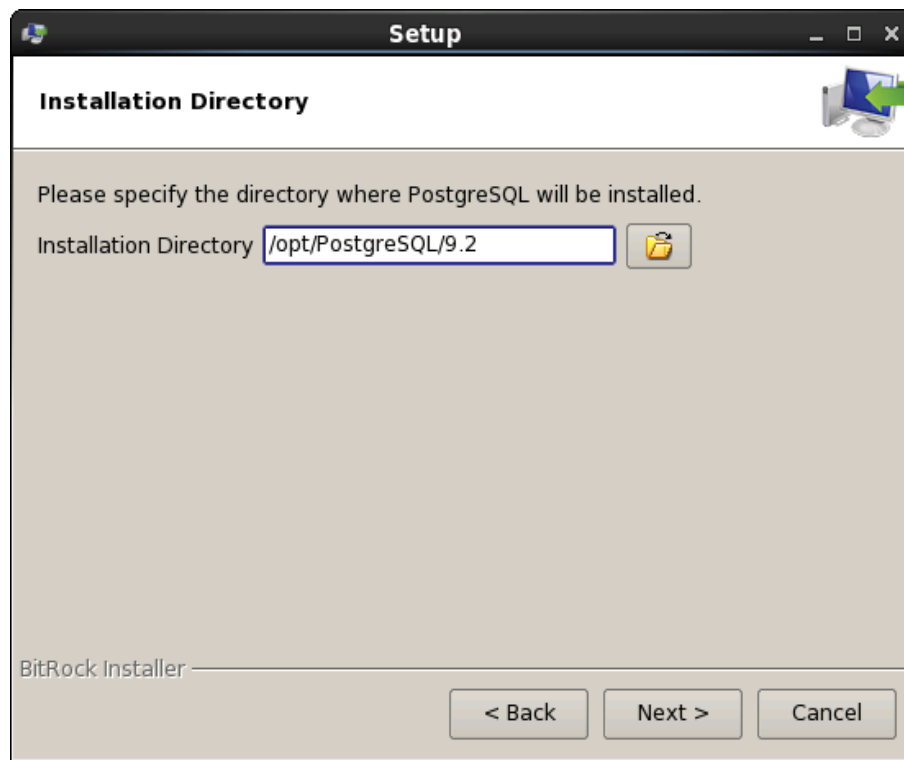
- rootユーザーで実行します。
- テキスト・モードを使用する場合「--mode text」パラメータを指定します。
- その他のオプションを表示する場合「--help」パラメータを指定します。





## ■ バイナリのインストール場所

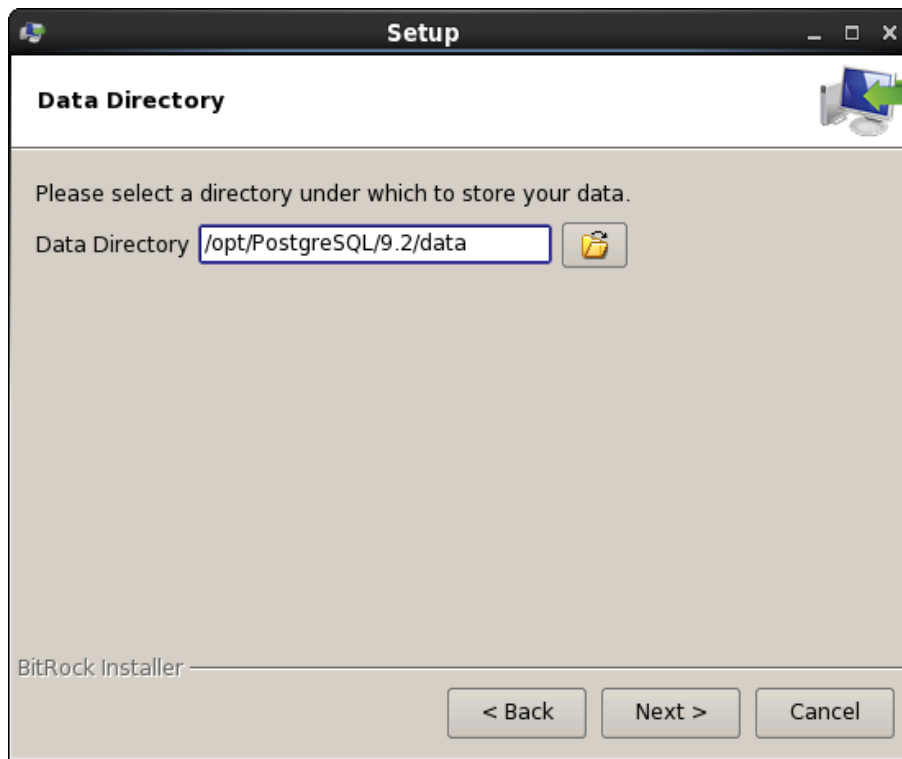
- 存在しなければ自動作成されます。
- ディレクトリの所有者はrootになります。





## ■ 初期データベース保存場所

- データベース・クラスタ用
- 所有者はpostgresになります。

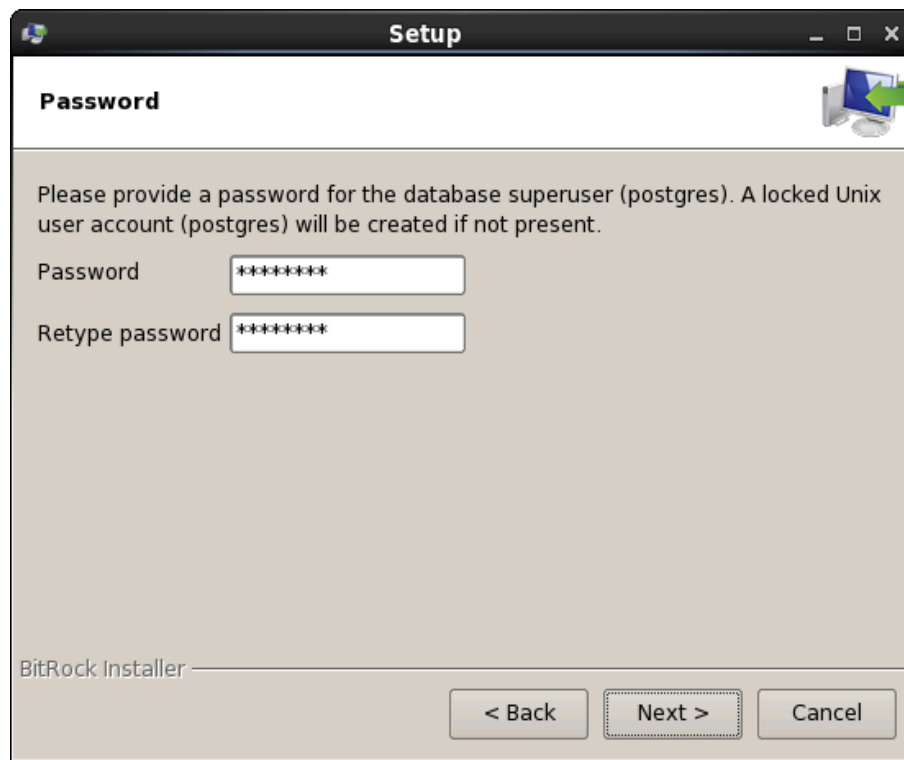






## ■ postgresユーザーのパスワード

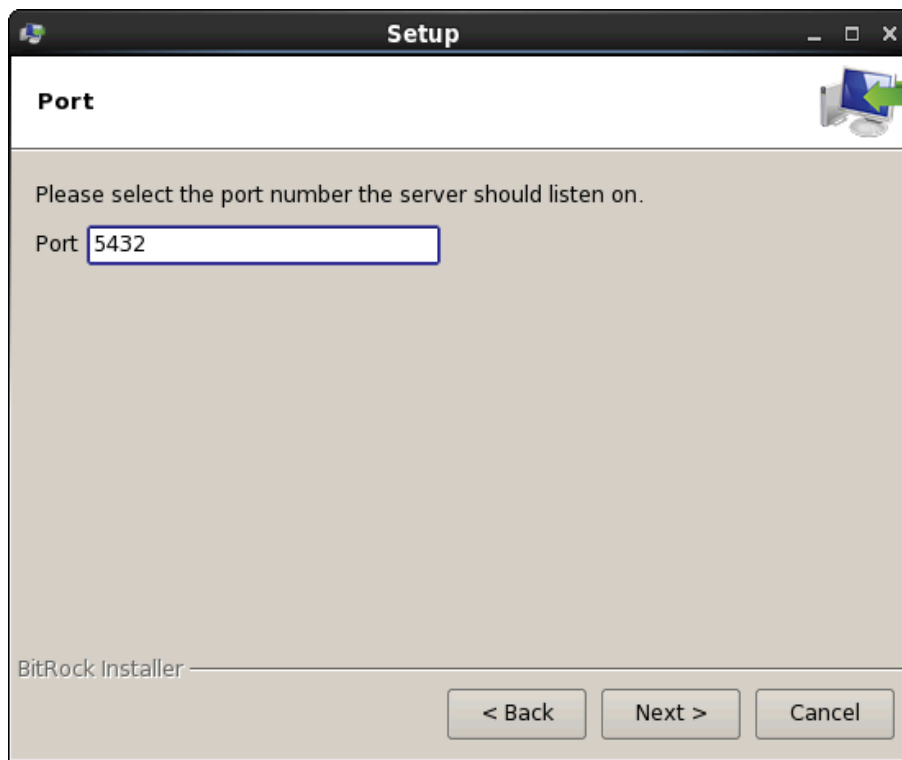
- OSユーザー (postgres) が作成されます。
- アカウントはロック状態です。
- ホーム・ディレクトリは/var/lib/postgresになります。





## ■ 接続待ちTCPポート番号

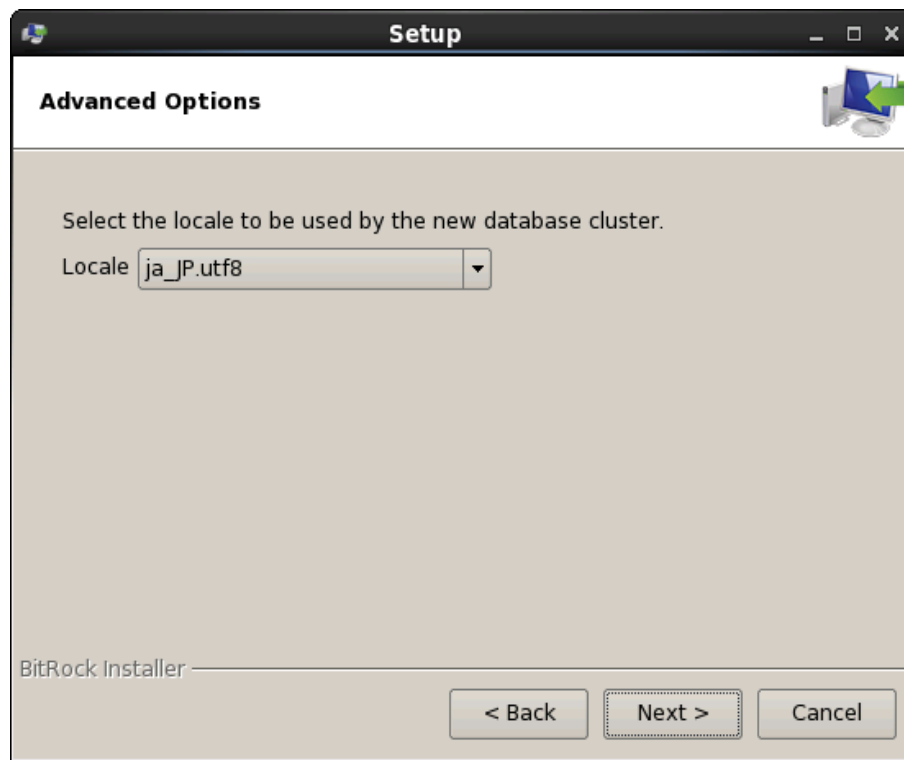
- 標準では5,432を使用します。





## ■ロケール設定

- 標準では環境変数から決定されます。
- 初期データベースの作成ロケールになります。





## ■ スタックビルダの起動

- 更新版ソフトウェアのダウンロードとインストールを行います。
- インターネット接続が必要です。
- PROXY経由の接続が可能です。





## ■ スタックビルダの実行

- 更新版ソフトウェアのダウンロードとインストールを行います。





## ■ ソースコードからのインストール

- postgres ユーザー (postgresグループ) を作成します
- ソース・コードを展開します
- コンパイルを行います (postgresユーザ)
  - `$ ./configure` ← 環境チェックと設定
  - `$ gmake` ← ビルド
  - `$ gmake check` ← ビルド確認
- インストールを行います (rootユーザ)
  - `# gmake install` ← インストール (/usr/local/pgsqlがデフォルト)
- データベース・クラスタを作成します (postgresユーザ)
  - `$ initdb ${PGDATA}`



機能	名前	URL
性能統計情報の取得	pg_statsinfo	<a href="http://pgstatsinfo.projects.postgresql.org/index_ja.html">http://pgstatsinfo.projects.postgresql.org/index_ja.html</a>
バックアップ リカバリ	pg_rman	<a href="http://code.google.com/p/pg-rman/">http://code.google.com/p/pg-rman/</a>
オンライン再編成	pg_reorg	<a href="http://reorg.projects.postgresql.org/index-ja.html">http://reorg.projects.postgresql.org/index-ja.html</a>
データ一括ロード	pg_bulkload	<a href="http://pgbulkload.projects.postgresql.org/index_ja.html">http://pgbulkload.projects.postgresql.org/index_ja.html</a>
レプリケーション	Slony-I	<a href="http://www.slony.info/">http://www.slony.info/</a>
GUIによる管理	pgAdmin3	<a href="http://www.pgadmin.org/">http://www.pgadmin.org/</a>
プロキシによる 負荷分散	pgpool-II	<a href="http://www.pgpool.net/mediawiki/index.php/Main_Page">http://www.pgpool.net/mediawiki/index.php/Main_Page</a>



## ■ 環境変数の設定と確認

```
$ cd ${HOME}           ←ホーム・ディレクトリへ
$ ./pg_env.sh         ←スクリプトの実行
$ env | grep PG       ←環境変数の確認
PGPORT=5432
PGUSER=postgres
PGDATABASE=postgres
PGLOCALEDIR=/opt/PostgreSQL/9.2/share/locale
PGDATA=/opt/PostgreSQL/9.2/data
$
```





## ■標準の設定

```
# TYPE DATABASE USER ADDRESS METHOD
#
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
```



# Oracleデータベースとの違い (アーキテクチャ編)



## ■ 基本機能比較

比較項目	Oracle Database	PostgreSQL
インスタンスとデータベースの関係	1 インスタンス = 1 データベース (RAC環境除く)	インスタンス内に複数データベース
トランザクション・ログ	REDOログ・ファイル	WAL (Write Ahead Logging)
ストレージ領域	<ul style="list-style-type: none"><li>・ ファイルシステム</li><li>・ ASM</li><li>・ RAWデバイス</li></ul>	ファイルシステム
表領域	データベースの一部	データベースの保存先
スキーマ	ユーザーと同一	ユーザーとは独立、省略時は public
読み取り一貫性	UNDO表領域	標準表領域に追記
スタンバイ・データベース	Oracle Data Guard	Streaming Replication
トランザクション分離レベル	Read Committed Serializable	Read Committed Serializable



## ■ プロセス構成

実行処理	Oracle Database	PostgreSQL
クライアントの接続待ち	リスナー	postmaster
SQL実行	サーバー・プロセス	postgres
プロセス監視	PMON	postmaster
トランザクション・ログの書き込み	LGWR	wal writer
データファイルへの書き込み	DBWR	writer
アーカイブログの書き込み	ARCH	archiver
パフォーマンス情報の収集	MMON	stats collector
不要領域の削除	SMON	autovacuum launcher
チェックポイントの実行	CKPT	checkpointer
ログ・ファイルの書き込み	—	logger
トランザクション・ログの送受信	—	wal sender / wal receiver



## ■ストレージ領域

ストレージ領域	Oracle Database	PostgreSQL
データベース全体管理	—	データベース・クラスタ
表領域の構成	データファイル <ul style="list-style-type: none"> <li>・オブジェクトを保存</li> <li>・所有者なし</li> </ul>	ディレクトリ <ul style="list-style-type: none"> <li>・データベースを保存</li> <li>・オブジェクトを保存</li> <li>・所有者あり</li> </ul>
トランザクション・ログ	REDOログ・ファイル <ul style="list-style-type: none"> <li>・固定サイズ</li> <li>・複数メンバー</li> <li>・グループ数固定</li> </ul>	\${PGDATA}/pg_xlogディレクトリ <ul style="list-style-type: none"> <li>・固定サイズ（標準16MB）</li> <li>・単一メンバー</li> <li>・グループ数可変</li> </ul>
制御ファイル	複数ファイル	\${PGDATA}/global/pg_control ファイル
初期化パラメータ・ファイル	\${ORACLE_HOME}/dbs/ - sp\${ORACLE_SID}.ora	\${PGDATA}/postgresql.conf
アーカイブ・ログ・ファイル	log_archive_destで指定	archive_commandで指定



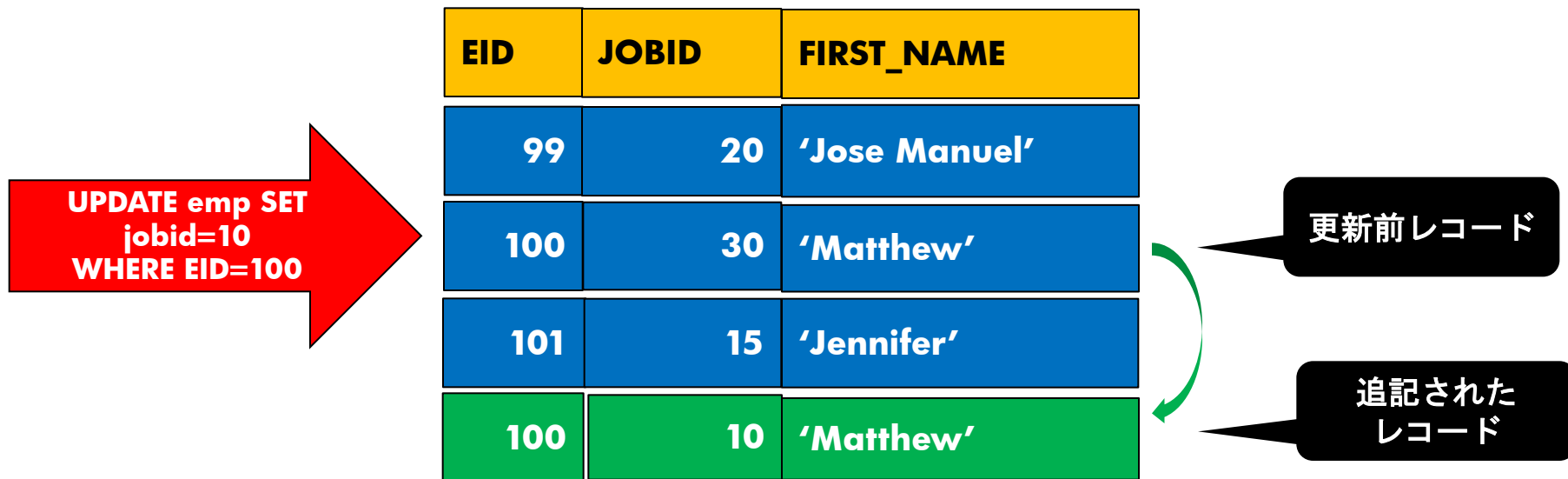
## ■メモリ構造

メモリ領域	Oracle Database	PostgreSQL
データ・バッファ	バッファ・キャッシュ (DEFAULT, KEEP, RECYCLE,非標準ブロック 用DEFAULT)	共有バッファ (shared_buffers)
ディクショナリ情報	共有プール	
SQL実行計画	共有プール	—
パラレル・クエリ情報等	ラージ・プール	—
結果キャッシュ	リザルト・キャッシュ	—
Javaアプリケーション関連	Javaプール (java_pool_size)	—
トランザクション情報	ログ・バッファ	WALバッファ (wal_buffers)
サーバー・プロセス	PGA (pga_aggregate_target)	ワークメモリ (work_mem) 一時メモリ (temp_buffers)
メンテナンス領域	—	メンテナンス用ワークメモリ (maintenance_work_mem)



## ■更新前情報は移動されません

- 更新後情報はレコード単位で追記されます
- 更新前情報は旧データであることを示す情報が付与されます





## ■VACUUM処理

- 不要データを空き領域リストに入れて再利用します (VACUUM CONCURRENT)
- オブジェクト統計が再計算されます (VACUUM CONCURRENT)
- データファイルの縮小を行います (VACUUM FULL)
- インデックスの再構築を行います (VACUUM FULL)
- アクセスが制限されます (VACUUM FULL)
- PostgreSQL 8.3以降は自動的に実行されます (パラメータautovacuum = on)
- 自動VACUUMの実行単位はテーブルです





## ■テキスト

操作	Oracle Database	PostgreSQL
自動コミットの標準	SQL*PlusではOFF ・ SHOW AUTOCOMMIT ・ SET AUTOCOMMIT ON	psqlではON ・ \echo :AUTOCOMMIT ・ \set AUTOCOMMIT off
DDLとトランザクション	DDLの実行により暗黙 COMMIT	自動コミットOFFの場合は DDLもトランザクションの 一部
トランザクション実行中の エラー発生時のCOMMIT	正常部分のみCOMMIT	全てROLLBACK
プロシージャ、ファンクシ ョン内のトランザクション制御	実行可能(SELECT文以外)	実行不可
テーブルに対するDDL実行中 のSELECT	実行可能	ブロックされる



## ■ 主なオブジェクトの比較

オブジェクト	Oracle Database	PostgreSQL
テーブル	Heap, IOT	Heapのみ
インデックス	B*Tree, Bitmap, Function	B*Tree, Hash, Function, 部分
ビュー	○	○
制約	○	○
シーケンス	○	○
シノニム	○	×
パーティション・テーブル	○	△ (複数テーブルの連携)
データベース・リンク	○	△ (dblink機能)
マテリアライズド・ビュー	○	×
ストアド・プロシージャ	○ (PL/SQLのみ)	○ (複数言語、プロシージャなし)
パッケージ	○	×
トリガー	○	○



## ■ 以下の理由により非推奨です。

- 操作がWALに記録されない場合があります
- ストリーミング・レプリケーションでは複製されません
- クラッシュ後には再構築が必要になる場合があります



## ■オブジェクトのグループ化機能です

- 名前と所有者を持ちます
- 複数作成することができます
- オブジェクトを「スキーマ名.オブジェクト名」の形式で特定します

## ■オブジェクトが所属するスキーマは以下の方法で決められます

- スキーマ名を省略した場合、パラメータsearch\_pathで指定された名前を検索して、存在した名前のスキーマに決定されます
- パラメータsearch\_pathの標準は \$user, public となっているため、ユーザ名と同じ名前のスキーマが存在しなければpublicとなります
- 存在しない(またはCREATE権限がない)スキーマを指定したSQLはエラーになります



# Oracleデータベースとの違い (プログラミング編)



- ANSI標準のSQL文には互換性があります
- 一部の構文は書き直しが必要です
  - ALL, AS OF, CONNECT BY, CUBE, CURSOR, FOR UPDATE OF, GROUPING SET, MERGE等
  - サブ・クエリーにおける一部の構文をサポートしません
  - (+) 句による結合が使用できません
  - 疑似列が使用できません
  - 演算子は豊富ですが、使用できない演算子もあります
- Oracle データベース独自の関数は変更が必要です



## ■ NULLと文字列の結合結果はNULLになります

■ `SELECT 'string' || NULL ⇒ NULL` (PostgreSQL)

■ `SELECT 'string' || NULL ⇒ 'string'` (Oracle データベース)

## ■ NULLと長さ0の文字列の比較結果は異なります

■ `" IS NULL ⇒ FALSE` (PostgreSQL)

■ `" IS NULL ⇒ TRUE` (Oracle データベース)



## ■ ストアド・ファンクション記述言語

## ■ 標準で以下の言語を選択できます

- PL/pgSQL
- PL/Tcl
- PL/Perl
- PL/pgSQL
- PL/Python

## ■ モジュール追加で他の言語による開発が可能です

- PL/php
- PL/Java
- PL/R
- PL/Ruby





- C言語用の埋め込みSQLプリプロセッサ
- Pro\*Cと同様の機能を持っています
- EXEC SQL構文による埋め込みSQLを実行します
  - 接続構文が異なります
  - 文字列保存フォーマットが指定できません
  - PL/SQLブロックが指定できません



## ■HP教育サービスはOSS-DBアカデミック認定校です





ご清聴ありがとうございました。

■お問い合わせ■

HP教育サービス

<http://www.hp.com/jp/education/>

e-mail : [dl1.cec@hp.com](mailto:dl1.cec@hp.com)

