

# Ora2Pg : Quick tour

PgSession #4 - 2012

# Qu'est-ce qu'Ora2Pg ?

- Outil créé en 2000
- D'abord un outil de duplication de données Oracle vers PostgreSQL
- Outil d'aide à la migration des bases Oracle vers PostgreSQL (mai 2001)
- Site officiel : <http://ora2pg.darold.net/>
- Version actuelle 9.2

# Prérequis à son utilisation

- Oracle  $\geq$  8i client ou serveur installé
- PostgreSQL  $\geq$  8.4 client ou serveur installé
- Perl 5.8+ et les modules DBI + DBD::Oracle
- Windows : Strawberry Perl 5.10+ ou ActiveState
- Modules Perl optionnels : DBD::Pg et Compress::Zlib
- Multi-threading : Perl compilé avec le support des threads

# Qu'est-ce qu'Ora2Pg permet de faire ? 1/3

- Export du schéma Oracle en ordres DDL compatibles PostgreSQL
- Conversion des types Oracle en types compatibles avec PostgreSQL
- Optimisation des conversions de types NUMBER
- Export des contraintes (clés primaire et clés uniques et clés étrangères)
- Export des contraintes CHECK

# Qu'est-ce qu'Ora2Pg permet de faire ? 2/3

- Export des séquences
- Export des index non implicites
- Export des tablespaces
- Export des triggers
- Export des tables et vues
- Export des utilisateurs et droits sur les objects
- Export des types utilisateurs

# Qu'est-ce qu'Ora2Pg permet de faire ? 3/3

- Export des fonctions et paquets de procédures stockées (PACKAGE)
- Réécriture complète des entêtes de fonction, passage de paramètre et déclaration de variables.
- Conversion automatique d'une partie du code PL/SQL en PL/PGSQL
- Export / Import des données dans des fichiers ou directement dans PostgreSQL.

# Les points forts d'Ora2Pg

- Code libre, gratuit et facilement extensible
- Simple et rapide de mise en œuvre
- Comportement vis à vis d'Oracle et de PostgreSQL paramétrable (encodage, etc.)
- Rapide au chargement des données (utilise COPY)
- Conversion automatique d'une partie du code PL/SQL en PL/PGSQL

# Les points faibles

- Pas de parallélisation de l'export des données
  - Uniquement parallélisation de l'encodage du bytea
- Pas de chargement incrémental des données
  - Même si la clause WHERE le permet en partie
- Ne permet pas la réécriture du code SQL complexe (fonction avec requêtes imbriquées, CONNECT BY, outer join (+), ...)
- Pas de transformation des données



# Fonctionnalités avancées 1/2

- Export du partitionnement Oracle en partitionnement par héritage dans PostgreSQL

```
CREATE TABLE ventes (annee integer, produit varchar(25), ...);
```

```
CREATE TABLE ventes_2010 ( CHECK ( annee = 2010 ) )
```

```
INHERITS (ventes);
```

```
CREATE TABLE ventes_2011 ( CHECK ( annee = 2011 ) )
```

```
INHERITS (ventes);
```

- Création des triggers pour envoyer les enregistrements dans la bonne partition

# Fonctionnalités avancées 2/3

- Export des tables avec ordres DDL pour ora\_fdw

```
CREATE FOREIGN TABLE ora_table (  
    id      integer      NOT NULL,  
    text    character varying(30),  
    floating double precision NOT NULL  
    ) SERVER ora_fdw_server OPTIONS (table 'ORA_TABLE');
```

- La table Oracle distante peut être utilisée normalement comme une table PostgreSQL

# Fonctionnalités avancées 3/3

- Export des vues matérialisées avec mises à jour par snapshot
  - reconstruction complète uniquement
- Pas de vues matérialisées à mise à jour synchrone
  - Nécessite la création de trigger
- Pas de vues matérialisées à mise à jour asynchrone
  - Nécessite de tracer les changements et de les appliquer à intervalle régulier (trigger + fonction)

# Conversion automatique du code PL/SQL - 1/3

- Paquet de procédure stockées convertis en fonctions associées à un schéma
  - PACKAGE.FONCTION => SCHEMA.FONCTION
- Entêtes et paramètres des triggers, séparation du code dans les fonctions
- Conversion des types des variables et réécriture dans un bloc DECLARE
- Réécriture des entêtes et paramètres des fonctions

# Conversion automatique du code PL/SQL - 2/3

- Fonctions équivalentes dont le nom change :
  - NVL() ⇒ coalesce()
- Fonctions dont les paramètres changent :
  - to\_number(num) ⇒ to\_number(num, '9999D9999')
- Fonctions dont les noms et les paramètres changent :
  - substr( string, start\_position, length ) ⇒  
substring(string from start\_position for length)

# Conversion automatique du code PL/SQL - 3/3

- Les fonctions dont la réécriture est complète :
  - `add_months` ⇒ `'N months'::interval`
  - `decode(user_status,'active',username,null)` ⇒  
(`CASE WHEN user_status='active' THEN  
username ELSE NULL END`)
- Bibliothèque de fonctions : ORAFCE
- Et bien d'autres réécritures.

# Comment l'utiliser ?

- Export du schéma et procédures stockées

```
EXPORT_TYPE="GRANT TABLESPACE TYPE TABLE  
SEQUENCE TRIGGER VIEW FUNCTION PROCEDURE  
PACKAGE"
```

```
for etype in $(echo $EXPORT_TYPE | tr " " "\n")
```

```
do
```

```
    mkdir schema/$etype
```

```
    ora2pg -p -t $etype -o $etype.sql -b schema/$etype \  
        -c /etc/config/ora2pg.conf
```

```
done
```

# Comment l'utiliser ?

- Création du propriétaire de la base cible
  - `createuser --no-superuser --no-createrole --no-createdb userpg`
- Création de la base de données :
  - `createdb -E UTF-8 --owner userpg pgdb`
- Création du schéma (espace de nommage)
  - `psql -U userpg pgdb -c "CREATE SCHEMA exported_schema;"`
  - `psql -U userpg pgdb -c "ALTER ROLE userpg SET search_path TO exported_schema,public;"`



# Comment l'utiliser ?

- Import du schéma dans la base PostgreSQL
  - Import des tables, vues, etc...
  - `psql -U userpg pgdb < schema/TABLE.sql`
  - ...
- Chargements différé de certains objets
  - Séquences
  - Contraintes
  - Déclencheurs
  - Index

# Comment l'utiliser ?

- Export des données de la base Oracle :
  - `ora2pg -t COPY -o data.sql -c /etc/ora2pg.conf`
- Import des données :
  - `psql pgdb < data/data.sql`
- Import direct des données dans PostgreSQL
  - `ora2pg -t COPY`

# Comment l'utiliser ?

- Chargement des procédures stockées une à une.
- Etudes des erreurs et débogage du code SQL
- Le temps de migration du schéma et des données est très rapide et marginal par rapport au temps de la migration du code PL/SQL

# Conclusion

- Projet en constante évolution
- Préférez toujours la dernière version d'Ora2Pg
- Faites un retour d'expérience de votre migration à l'auteur
  
- Code en développement :
  - <https://github.com/darold/ora2pg>

# Ora2Pg : Quick tour

Des questions ?

