



La validation de migration facilitée par Ora2Pg



Thèmes abordés

- ▷ Vérification des types de données.
- ▷ Vérification des objets migrés.
- ▷ Vérification des données.



1.

Présentation



Présentation

Auteur : Gilles Darold (CTO at MigOps Inc)

Email : gilles@migops.com

MigOps Inc

Entreprise spécialisée dans la migration vers PostgreSQL

- ▷ Finance le développement d'Ora2Pg
- ▷ Support commercial pour Ora2Pg.

Contact : <https://www.migops.com/contact-us/>

Célébration

Joyeux anniversaire Ora2Pg !

20 ans

Première version le 05 mai 2001

Version 23.0 publiée le 15 novembre 2021



Migration vers PostgreSQL

Les étapes



Etapes de la migration

Estimation/Analyse Analyse de la faisabilité et de l'effort de migration

Migration

Mise en oeuvre des tâches déduites de l'analyse, migration du schéma, des données, du SQL, des procédures stockées et de l'application

Testing

Test des objets et données migrées, tests de l'application, des batchs et du workflow complet

Performances

Analyser les problèmes de performances et amener des correctifs, soit au niveau du SQL, de PostgreSQL ou au niveau applicatif

Formation

Les équipes doivent être formées au nouveau SGBD en fonction des besoins de l'entreprise

Support

Un support 24/7 pour la résolution d'incidents, l'aide à la mise en oeuvre opérationnelle ou la réponse à des questions d'exploitation.



Testing

C'est la clé du succès de votre migration

- ▷ Testez, testez et tester encore !

En profiter pour intégrer plus de test unitaires ?

Valider plusieurs fois les étapes de bascule en production



2.

Tests sur les objets

Type d'objets

TYPES

SEQUENCES

TABLES

INDEXES

CONTRAINTEs

TRIGGERS

VIEWS

MATERIALIZED VIEWS

PARTITIONS

FONCTIONS

PROCEDURES

TABLESPACES

PACKAGES => SCHEMA

DBLINKS => dblink/oracle_fdw

SYNONYMS => VIEWS

Validation des types de données

Chargement d'une partie des données permet de détecter :

- ▷ Les problèmes d'integer vs decimal
- ▷ RAW(16) ou RAW(32) vs Uuid
- ▷ Transformation en booléen
- ▷ Champs varchar() avec limite de taille

Chargement d'un nombre limité de données:

```
WHERE          ROWNUM < 10000
```



Décompte des objets

```
ora2pg -c config/ora2pg.conf -t TEST > test_objects.log
```

Principe :

- ▷ Connexions simultanées sur la base Oracle et la base PostgreSQL
- ▷ Extraction et décompte de chaque type d'objet
- ▷ Comparaison entre les deux extractions et status
- ▷ Remontée des erreurs si il y en a



Décompte des objets

- ▷ TABLES
- ▷ TRIGGERS
- ▷ VUES
- ▷ SEQUENCES avec verification LAST_VALUE
- ▷ TYPES UTILISATEURS
- ▷ TABLES EXTERNES (ALL_EXTERNAL_TABLE vs FOREIGN TABLE)

Décompte global du nombre de fonctions :

- PACKAGES
- FONCTIONS
- PROCEDURES



Décompte par tables

- ▷ INDEXES
- ▷ CONTRAINTES UNIQUES
- ▷ CLÉS PRIMAIRES
- ▷ CONTRAINTES CHECK
- ▷ CONTRAINTES NOT NULL
- ▷ COLONNES AVEC VALEUR PAR DÉFAUT
- ▷ IDENTITY COLUMN
- ▷ CLÉS ÉTRANGÈRES
- ▷ TRIGGERS
- ▷ PARTITIONS



EXEMPLES

Exemple action TEST avec migration de la base HR

https://www.ora2pg.com/TEST_example.txt

https://www.ora2pg.com/TEST_example_error.txt

Vérification du nombre de lignes

```
ora2pg -c config/ora2pg.conf -t TEST --count_rows
```

Compte du nombre de lignes de chaque table pendant le décompte des objets.

Action dédiée pour ne faire que le décompte des lignes

```
ora2pg -c config/ora2pg.conf -t TEST_COUNT
```

(utile après un second chargement par exemple)



EXAMPLE

[TEST ROWS COUNT]

ORACLE:actor:200

POSTGRES:actor:200

ORACLE:address:603

POSTGRES:address:603

ORACLE:film_actor:5462

POSTGRES:film_actor:5462

ORACLE:film_category:1000

POSTGRES:film_category:1000

ORACLE:film_text:1000

POSTGRES:film_text:1000

(...)

[ERRORS ROWS COUNT]

OK, Oracle and PostgreSQL have the same number of rows.



3. Tests des vues

Vérification du retour des vues

```
ora2pg -c config/ora2pg.conf -t TEST_VIEW
```

Compte le nombre de lignes retournés par chaque vues.

Pas de control des données retournées, uniquement du nombre de lignes. Une validation au niveau applicatif s'impose.



EXAMPLE

[UNITARY TEST OF VIEWS]

ORACLE:actor_info:200

POSTGRES:actor_info:200

ORACLE:customer_list:599

POSTGRES:customer_list:599

ORACLE:film_list:997

POSTGRES:film_list:997

ORACLE:nicer_but_slower_film_list:997

POSTGRES:nicer_but_slower_film_list:997

ORACLE:sales_by_film_category:16

POSTGRES:sales_by_film_category:16

ORACLE:sales_by_store:2

POSTGRES:sales_by_store:2

ORACLE:staff_list:2

POSTGRES:staff_list:2



4.

Tests des données

Nouveau à partir de la version 23.0

Test du temps de migration

Réduire la fenêtre de coupure nécessaire pour la bascule en production.

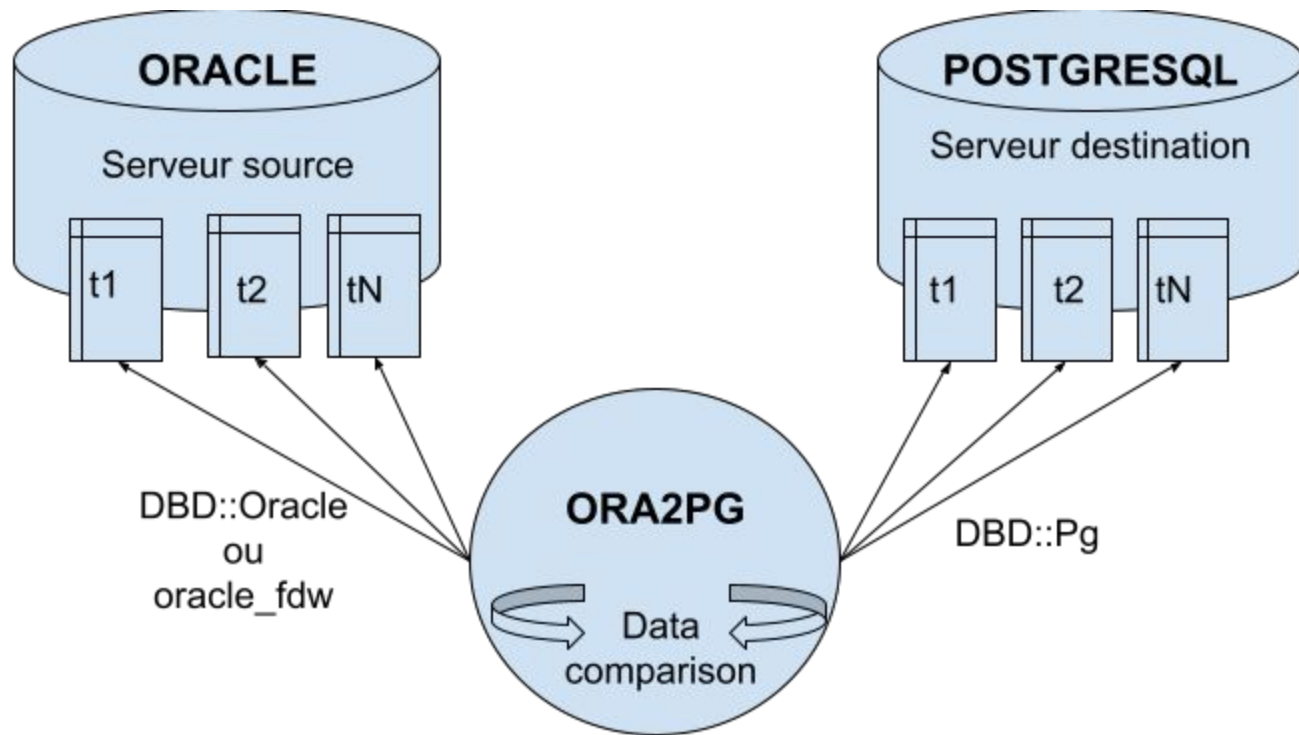
- ▶ Tester le temps de migration des données avec les options :
 - -P : nombre de tables exportées en parallèle
 - -J : nombre de processus Oracle // pour une table
 - -j : nombre de processus d'écriture dans PostgreSQL par table.
- ▶ Sans et avec oracle_fdw (optimum pour BLOB avec -J)
- ▶ Dissocier données d'archive et données "vivantes" pour les bases de plusieurs To

Vérification des données

```
ora2pg -c config/ora2pg.conf -t TEST_DATA
```

Vérifie une à une les valeurs retournées par les deux SGBD.

Utilise oracle_fdw ou mysql_fdw mais aussi une connection directe.



Data validation - TEST_DATA

Prérequis

S'assurer que les colonnes et leurs types de données dans la source de données et la destination coïncide.

- ▷ Table avec clé primaire ou unique pour ORDER BY, sauf chargement initial sans parallélisme.
- ▷ Collation 'C' sur ces clés, sinon difference de tri
- ▷ Aucune modification de données pendant le décompte

Vérification des données

Le résultats des vérifications est enregistré dans le fichier `data_validation.log`.

Dans le répertoire courant ou celui spécifié par l'option `-b | --basedir`

Les erreurs rapportées sont limités à 10 avant l'arrêt de la vérification pour une table en erreur.

Il est possible de paralléliser la vérification avec l'ajout de l'option `-P | --parallel`



Paramétrage

FDW_SERVER	Nom du foreign server pour se connecter à Oracle. Si non défini utilise une connection directe pour requêter les tables.
PG_DSN	Paramètre de connection à la base PostgreSQL
DATA_VALIDATION_ROWS	Nombre de lignes à tester au maximum. Valeur par défaut : 10000 Une valeur de 0 provoque le test de toutes les lignes des tables
DATA_VALIDATION_ERROR	Par défaut la vérification des données d'une table s'arrête après 10 erreur. Ce nombre peut être augmenté si l'on veut traiter plus d'erreur en une passe.
PARALLEL_TABLES	Paralléliser la vérification des données par table, utilise 1 seul processus par défaut
DATA_VALIDATION_ORDERING	Tri les données selon une clé unique. Si désactivé pas de tri.

Vérification des données

Limitations:

- ▷ Pas de vérification multi-schéma, uniquement schéma par schéma.
- ▷ Pas de test de types défini par l'utilisateur, pour l'instant
- ▷ Pas de test partition par partition, seulement par table partitionnée.



5.

Différences de structure

Quid des differences de structure ?

Lors de la vérification Ora2Pg supporte nativement les changements :

- ▷ Du nom de schema des objets (PG_SCHEMA)
- ▷ Du nom des tables (REPLACE_TABLES)
- ▷ Du nom des colonnes (REPLACE_COLS)
- ▷ De la suppression de colonnes (MODIFY_STRUCT)

Exemple de modification de structure

Renommage d'une table :

- ▷ REPLACE_TABLES PRODUCT_TMP:PRODUCT2

Renommage d'une colonne :

- ▷ REPLACE_COLS RAW_INFO(UID_COL:COL_UID)

Suppression d'une colonne à la migration :

- ▷ MODIFY_STRUCT RAW_INFO(ID,UID_COL,INFO_COL)

(contient une quatrième colonne ACTIVE dans la base source)

Quid des differences de types

Lors de la vérification Ora2Pg supporte nativement le changement :

- ▷ En booléen (REPLACE_AS_BOOLEAN et BOOLEAN_VALUES)
- ▷ La transformation des RAW(16) et RAW(32) en uuid (défaut)
- ▷ La redéfinition de types de données (DATA_TYPE)



Merci !

Des questions?

<http://www.ora2pg.com/>

Postez vos rapports de bugs, demandes de fonctionnalités, contributions sur :

<https://github.com/darold/ora2pg/issues>