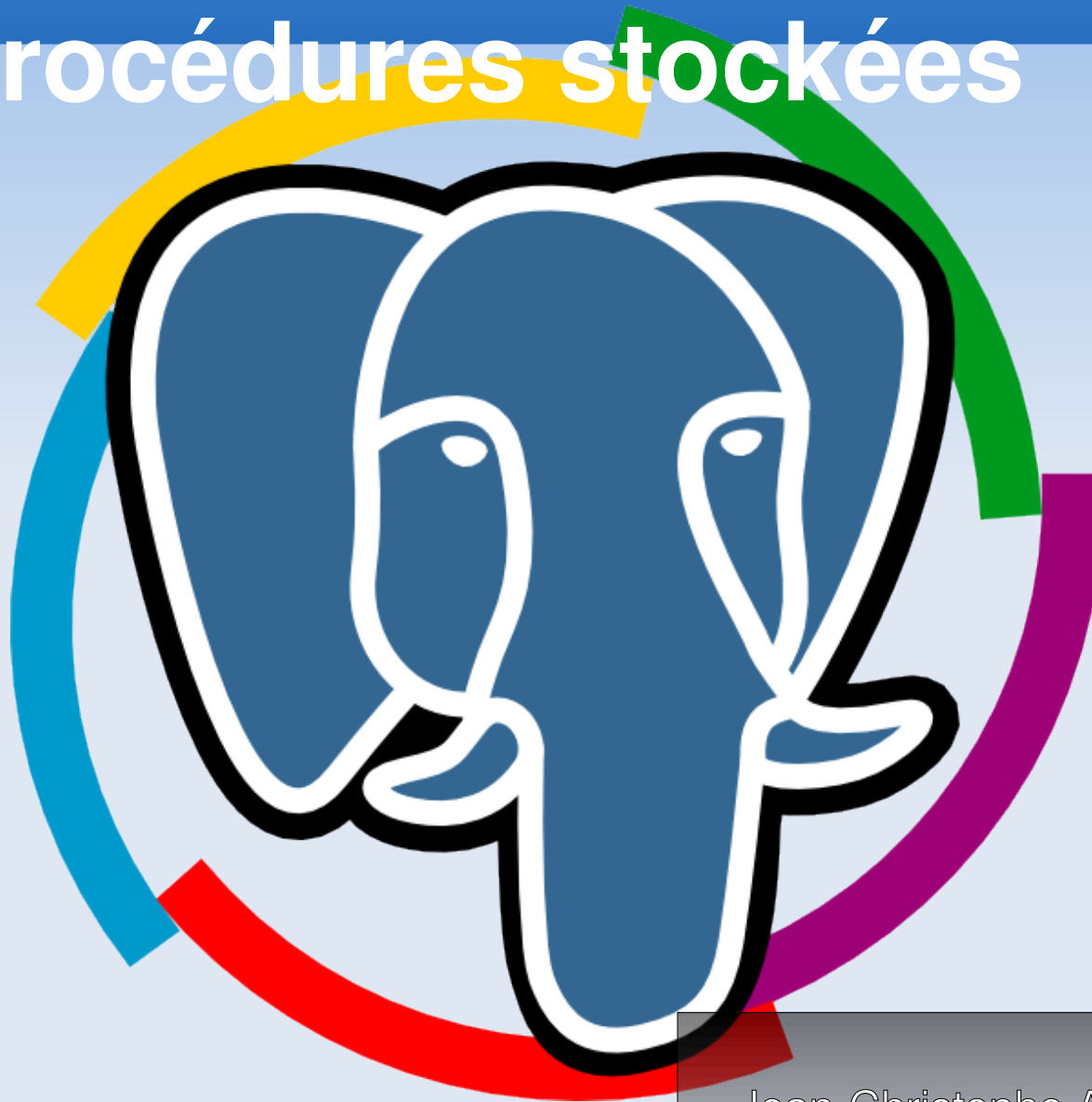


PL/pgsql : langage de procédures stockées



Jean-Christophe Arnu
PostgreSQLFr/CS



Qu'allons nous voir aujourd'hui?

- Quelques généralités sur PostgreSQL
- Des procédures stockées? Pour quoi faire?
- Anatomie d'une procédure stockée
- Syntaxe du langage PL/pgSQL
- Exercices



2008-07-04

Tutoriel PL/pgSQL

PostgreSQL

- Un SGBD open source sous licence BSD
- Conception pour des fonctionnalités professionnelles
 - Contraintes d'intégrité
 - Procédures stockées
 - Réplication



2008-07-04

Tutoriel PL/pgSQL

Les procédures stockées

- Depuis le début (PostgreSQL95)
- Plusieurs langages possibles (plugable)
- Au plus près des données :
 - Tout sur le serveur
 - Tout dans le schéma de base de données
 - Pas de consommation de ressources excessives



2008-07-04

Tutoriel PL/pgSQL

Les procédures stockées

```
CREATE [OR REPLACE] FUNCTION nom (<paramètres>)  
[RETURNS <type>] AS  
$$  
<corps>  
$$  
LANGUAGE plpgsql;
```

- Paramètres IN, OUT, INOUT
- Types de retour de type
 - INTEGER, FLOAT, ...
 - RECORD une ligne
 - SET OF une table



Le langage PL/pgSQL

- Compris entre les here-docs (\$\$)
 - Un bloc DECLARE
 - `nom [CONSTANT] type [NOT NULL] [{DEFAULT | :=} expr]`
 - Type : INTEGER, FLOAT, ... (types classiques)
 - Type : `table.colonne%TYPE` (recopie de type)
 - Type : `table%ROWTYPE` (ligne) ou RECORD (ligne anonyme)
 - Type : `anyelement` (polymorphe)
 - Nom ALIAS FOR \$n (alias sur le paramètres)
 - Un corps (BEGIN ... END)
 - Le traitement à proprement parler!



Affectations/Exécution requêtes

- Variable : = valeur; -- type simple
- SELECT clause INTO variable FROM requête;
-- types simples et lignes
- PERFORM requête -- exécute une requête sans retour
- EXECUTE requête INTO variable -- exécute une requête



Structures de contrôles

- RETURN
- RETURN NEXT expression renvoie plusieurs lignes (SET OF type)
- RETURN QUERY requête
- IF THEN ELSE / ELSEIF END;
- LOOP ... END LOOP [label];
- EXIT [label] [WHEN condition]
- CONTINUE [label] [WHEN condition]



Structures de contrôle

- WHILE condition LOOP ... END LOOP;
- FOR var IN début..fin [REVERSE] [BY pas] LOOP ... END LOOP;
- FOR var_ligne IN EXECUTE requête LOOP ... END LOOP;



Exceptions

- Bloc EXCEPTION:
- RAISE
 - NOTICE
 - EXCEPTION

```
EXCEPTION
```

```
WHEN no_data_found THEN
```

```
-- faire quelque chose
```

```
WHEN division_by_zeo THEN
```

```
--traiter la division par zéro
```



Les curseurs - 1

- Dans le bloc DECLARE
 - Nom [[NO] SCROLL] CURSOR [(param)] FOR requête;
- C1 CURSOR FOR requête
 - OPEN C1;
- C2 REFCURSOR;
 - OPEN C2 FOR requête;
 - OPEN C2 FOR EXECUTE requête;



Les curseurs - 2

- Moncurs CURSOR (maxsal : float) FOR
SELECT * FROM salaire < maxsal;
- FETCH direction FROM curseur INTO cible
 - Cible peut être une ou plusieurs variables, une ligne
 - Direction FORWARD, BACKWARD, RELATIVE,
FIRST, LAST, ABSOLUTE, NEXT, PRIOR
- MOVE direction FROM curseur; -- ~ fetch mais
sans retourner
- CLOSE curseur;



Et les triggers ?

- Ce sont des procédures stockées aussi
- CREATE TRIGGER
- Création de plusieurs variables implicites (principaux)
 - NEW et OLD (RECORD)
 - TG_OP, TG_WHEN, TG_TABLE_NAME, TG_LEVEL
- Le trigger renvoie NULL ou un RECORD de la table sur laquelle on opère



Création d'un trigger

- CREATE FUNCTION nom_fonc()
RETURNS TRIGGER AS \$\$

...
\$\$ LANGUAGE plpgsql;
- CREATE TRIGGER nom_trig
[BEFORE|AFTER] evenement [OR ...]
[FOR [EACH] {ROW|STATEMENT}]
EXECUTE PROCEDURE nom_fonc (args)

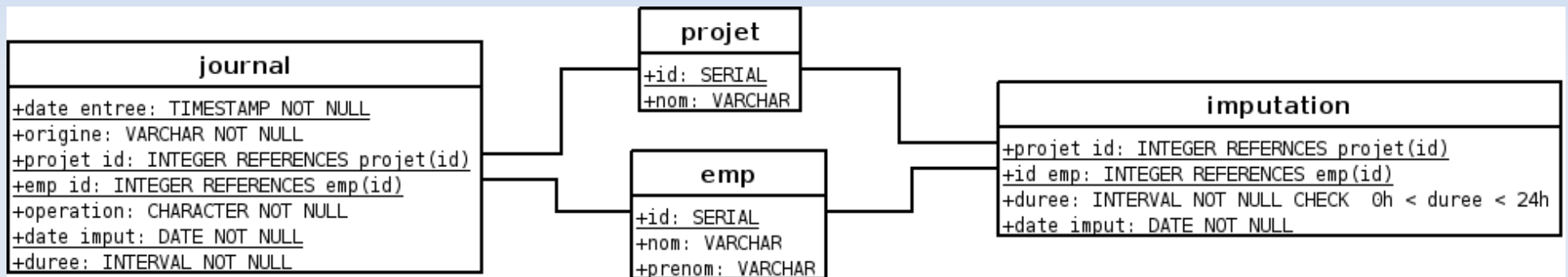


Support d'exercice - 1

- Une entreprise veut récupérer le temps passé sur les projets de chacun de ses employés
- Elle veut aussi vérifier que chacun des employés remplit bien ses imputations
 - Connaître la quantité d'employé en sur-travail (exercice 1) avoir le score de l'employé le plus vaillant (exercice 2)
 - Avoir une liste des heures supplémentaires par employé (exercice 3)
 - Avoir un journal des opérations d'imputations (exercice 4)



Support d'exercice - 2



Exercice N°1

- Installation de PostgreSQL
- Installation de PL/pgSQL
- Installation d'une base
- Lister le nombre d'employés ayant dépassé leur quota paramétrable d'heures hebdomadaires avec une procédure stockée « nb_employees_surtravail »



Exercice N°2

- Retourner l'employé le plus zélé, la semaine et l'année où il a été zélé et le cumul d'heure pour cette semaine.



Exercice N°3

- Afficher la liste des utilisateurs zélés



2008-07-04

Tutoriel PL/pgSQL

19

Exercice N°4

- Créer un trigger permettant de journaliser toutes les opérations d'imputation dans la table journal dès que l'on ajoute, modifie ou supprime une entrée dans la table imputation.



Langages procéduraux?

- Il existe un grand nombre de langages procéduraux :
 - PL/Perl, PL/Python, PL/Tcl sont des langages procéduraux officiels
 - PL/Java, PL/LUA, PL/sh, PL/Scheme sont disponibles sur pgFoundry
 - PL/ruby existe également.



Merci!

- À Vous!
- À OpenOffice.org (PostgreSQL fonctionne bien avec OOo)
- À Keyjnote !



2008-07-04

Tutoriel PL/pgSQL

22