Principes d'utilisation des systèmes de gestion de bases de données

PL/SQL : les bases

M1, Informatique
Emmanuel Waller, LRI, Orsay

Rappel : les deux parties du cours

- création et gestion de la base
 - les problèmes BD
 - les traitements bas niveau en mode programme : PL/SQL
 - Introduction
 - Les bases
 - Curseurs
 - SQL dynamique
 - Triggers
 - Exceptions, gestion des erreurs BD
 - PL/SQL et les problèmes du mode programme
- accès à la base depuis un programme généraliste :
 - les problèmes MP étudiés à travers (outre PL/SQL) : Java, PHP
- XML, XSQL, XSLT, intégration avec relationnel

PL/SQL, les bases

- Exemples
- le langage :
 - partie programmation : bloc, types, variables, opérateurs, tests, boucles
 - partie BD : ordres SQL, types
- procédures stockées

exemple

- On considère un centre de loisir, représenté par une table loisir(nom, âge, activité)
- Quand c'est l'anniversaire de quelqu'un, il faut :
 - Augmenter son âge de 1
 - Et toutes les personnes qui ont son âge font son activité
- Ecrire une procédure stockée qui prend en paramètre le nom de la personne dont c'est l'anniversaire, et effectue ce traitement
 - On suppose qu'il y en a exactement une

- Code, appel en mode interactif
 - démonstration

Brefs commentaires

- PL/SQL :
 - On reviendra sur tout cela en détail
 - Create or replace
 - Paramètres, corps, is
 - Variables
 - Ordres SQL (variante select)
 - Casse pas significative

- Devant les machines :
 - Charger sous SQL*Plus provoque :
 - Compile
 - Stocke (si pas d'erreur de compilation)
 - Return + slash + return
 - Message :
 - Correct
 - Erreur => débbuguer
 - Set serveroutput on
 - aAppel

exemple

- Ecrire une procédure stockée qui renvoie le nombre de lignes de la table :
 - Rem : ce sera une fonction
- Code, appel
 - démonstration

PL/SQL, les bases

- Exemples
- le langage :
 - partie programmation : bloc, types, variables, opérateurs, tests, boucles
 - partie BD : ordres SQL, types
- procédures stockées

Le bloc PL/SQL

- Séquence d'ordres SQL et instructions PL/SQL
- Unité de base :
 - Structure des procédures stockées
 - exécution hors procédures : compilé et exécuté en une seule fois
- 3 sections
- Commentaires :
 - ceci est un commentaire, sur une ligne
 - /* et cela
 - Aussi, sur plusieurs */

Structure du bloc

- declare
 - mot clé remplacé par is dans procédures stockées)
 - ... /* variables, constantes, exceptions, curseurs */
- begin
 - ... /* Ordres SQL, instructions PL/SQL, structures de contrôle */
- exceptions
 - ... /* Traitement des exceptions */
- end

Exemple

- declare

```
x varchar2(10);
```

```
begin
```

```
x:= 'Hello World';
```

```
dbms_output.put_line(x);
```

```
end;
```

- démonstration

Quels ordres SQL dans PL/SQL ?

- Interrogation : select
- Mise à jour (instance) : insert, update, delete
- Persistance et transactions
- Fonctions : to_char, to_date, round, etc.
- Rien d'autre

- Syntaxe : identique à SQL (sauf certains cas)

Déclaration des variables

- Dans section « declare »
- Variable = nom + type (+ contrainte BD)
- Types =
 - BD : tous types (integer, varchar2, etc.)
 - Programmation :
 - Non BD (ex : booléen)
 - Dérivé

Exemple

-- programmation :

adresse varchar2(20);

x integer := 1;

-- BD :

nom varchar2(10) not null;

Types dérivés

- Référence à une entité existante :
 - Programmation : variable
 - BD : colonne, table, curseur
- Exemple :
 - programmation :
 - x integer;
 - y x%type;
 - BD :
 - vnom client.nom%type;
 - cli client%rowtype;

Utilisation des variables

- Affectation à une variable de :
 - Programmation : expression
 - BD : résultat requête : select into, fetch into (curseur)
 - Rappel : sous SQL*Plus : à l'écran
 - Dans PL/SQL : dans une variable (dans curseur si plusieurs lignes)

Exemple (programmation)

$x := 0;$

$\text{vnom} := \text{'Monsieur'} \parallel \text{vnom};$

$y := (x + 5) * y;$

declare Exemple (BD)

```
vref varchar2(10);  
vprix articles.prixht%type;  
cli client%rowtype;
```

begin

```
select refart, prixht  
    into vref, vprix  
    from articles  
    where nom = 'cravate';
```

```
select *  
    into cli  
    from client  
    where nom = 'Toto';
```

end;

- Résultat de la requête :
 - Exactement une ligne
 - Zéro ou plusieurs : erreur

Exemple (BD)

declare

 vnocli client.nocli%type := 10;

 vnom client.nom%type;

begin

 vnom := 'Dupont';

 update client

 set nom = vnom

 where nocli = vnocli;

end;

Portée des variables

- Concept programmation
- Leur bloc PL/SQL
- (il existe cas plus généraux)

Opérateurs, tests

- Opérateurs :
 - Programmation : =, <, >, !, >=, <=, between, like, and, or, etc.
 - BD : is (not) null
 - (comme SQL)

- Tests :

```
if vnocli = 10
then
    update ... ;
    insert ... ;
else delete ... ;
end if;
```

Boucles

- Exemple : créer des clients
 - for n in 100 .. 110 loop
 - insert into client(nocli) values (n);
 - end loop;
 - n := 100;
 - while n <= 110 loop
 - insert into client(nocli) values (n);
 - n := n + 1;
 - end loop;

PL/SQL, les bases

- Exemples
- le langage :
 - partie programmation : bloc, types, variables, opérateurs, tests, boucles
 - partie BD : ordres SQL, types
- procédures stockées

Procédures stockées : syntaxe

create or replace procedure p

(x1 in | out t1, x2 . . . , . . .) is

. . . -- déclarations

begin

. . . -- corps

[exception . . .] -- gestion des exceptions

end;

- Out : passage par adresse

Fonction stockée

create or replace function f (...)

return integer /* ou autre */ is

begin

...

end;

remarque

- SQL :
 - drop procédure p
 - drop fonction f

- Appel :

- Depuis bloc PL/SQL :

- declare x char;

- begin . . . suppr(x); . . . end;

- Sous SQL*Plus :

- SQL> execute suppr('W');

- Rem : fonction : utiliser la valeur de retour

vocabulaire

- Rappel : modes d'utilisation : appel procédures stockées depuis :
 - une procédure stockée
 - interactif
 - un programme généraliste
 - bref, comme... un ordre SQL
- ordre BD =
 - un ordre SQL
 - appel de procédure stockée PL/SQL

PL/SQL, les bases

- Exemples
- le langage :
 - partie programmation : bloc, types, variables, opérateurs, tests, boucles
 - partie BD : ordres SQL, types
- procédures stockées

PL/SQL, les bases : que faut-il se rappeler ?

- PL/SQL :
 - Conçu : couplage fort avec SQL et manipulations BD
(gestion des problèmes du mode programme)
 - Expressivité : toutes fonctions calculables
- Les parties de code purement BD d'une appli généraliste doivent être :
 - Ecrites en PL/SQL
 - Stockées dans la base

PL/SQL, les bases : compétences à acquérir

- Savoir écrire et appeler des procédures stockées utilisant tous ces éléments :
 - Ordres SQL
 - Tests, boucles
- Dans le cadre d'un cahier des charges

Résoudre un exercice

- Exercice : écrire une procédure stockée
- Écrire ordres BD sur un exemple comme si interactif
- Choix procédure ou fonction
- Choix variables :
 - Paramètres
 - Passage information entre ordres
- Appel
- show errors