

SGBD - 2^e

PL-SQL - Chapitre 1 - Généralités

Daniel Schreurs

14 février 2022

Haute École de Province de Liège

Table des matières du chapitre i

1. Introduction
2. PL/SQL
3. Déf de blocs de programmation SQL
4. Procédures stockées en PL/SQL
5. Architecture PL/SQL

Introduction

Table des matières de la section : Introduction i

1. Introduction

1.1 Définition

2. PL/SQL

3. Déf de blocs de programmation SQL

4. Procédures stockées en PL/SQL

5. Architecture PL/SQL

- PL/SQL : Procedural Language extensions to SQL ;
- PL/SQL est un langage procédural qui permet de traiter de manière structurée (conditionnelle ou itérative) les données retournées par une instruction SQL ;
- Il s'agit d'un langage propriétaire, PL/SQL est la solution proposée par Oracle.

- Au niveau syntaxe, un programme est constitué de procédures et de fonctions ; des variables permettent l'échange d'information entre les requêtes SQL et le reste du programme.
- **PL/SQL n'a aucun aspect normatif** contrairement à SQL.¹
PL/SQL permet de **stocker du code sur le serveur** et sert principalement à **programmer des procédures stockées et des déclencheurs** (triggers)

1. Mais avec SQL3, la norme SQL a prévu les éléments de langage procédural normatif propre au langage SQL.

PL/SQL

Meilleure performance en écrivant des blocs de programmation PL/SQL :

- Les ordres SQL ne sont plus transmis un à un au moteur de base de données Oracle mais par bloc de programmation moins de trafic réseau
- Le moteur PL/SQL a été optimisé ce qui améliore encore les performances globales des applications

PL/SQL : Pourquoi

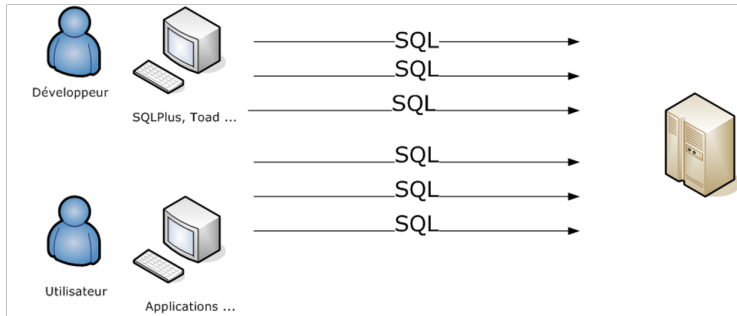


Figure 1 – Sans PL/SQL : envoi d'instructions SQL

PL/SQL : Pourquoi

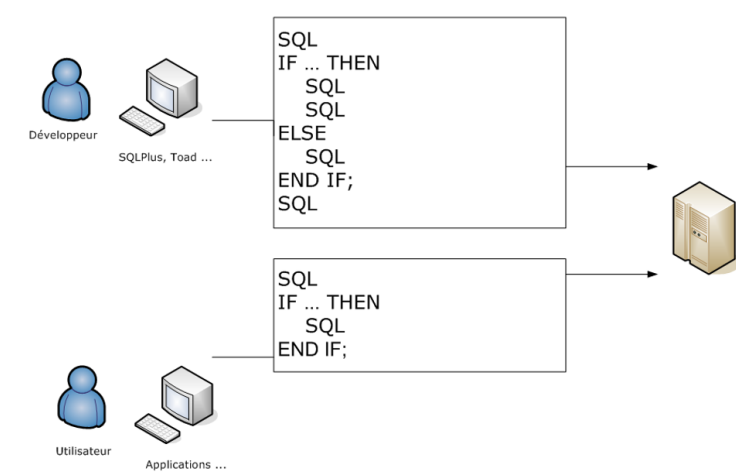


Figure 2 – Avec PL/SQL : écriture de blocs de programmation intégrant SQL

Une des principales forces du PL/SQL est son intégration avec SQL, surtout au niveau des différents types de données manipulés :

- Une donnée de type **DATE** ou **VARCHAR** stockée dans la base de données va être stockée dans une variable de même type dans du code PL/SQL
- Une instruction de sélection SQL retourne un result set qui peut être défini et exploité en PL/SQL

Déf de blocs de programmation SQL

- Les portions de code PL/SQL peuvent être **définies dans n'importe quelle interface SQL**
- Il suffit d'encoder la portion de code **comme s'il s'agissait d'une instruction SQL** et de l'exécuter.
- Dans cette portion de code, il est possible de réaliser des affichages, d'intercepter des exceptions et de les traiter.

Pour un développeur :

- Programmation SQL dynamique, déclencheurs, méthodes des TAD (types abstraits de données)
- Intégration aisée dans les langages classiques OO grâce à sa forte coloration OO

Pour un administrateur, possibilité :

- d'écrire des routines d'administration ;
- de définir des jobs récurrents,etc.

Procédures stockées en PL/SQL

- PL/SQL permet la création de procédures stockées : terme générique qui signifie qu'une portion de code peut être stockée sur le serveur de base de données dans la base de données elle-même.
- Une partie du code peut donc être portée sur le serveur et exécutée par celui-ci. Dans l'architecture classique "client-serveur", le client supporte toute la partie interface (ou présentation) de l'application ainsi que la logique de celle-ci.

Procédures stockées en PL/SQL

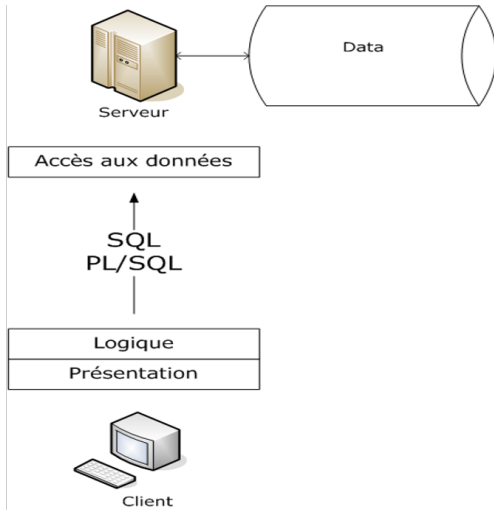


Figure 3 – Architecture client-serveur de base

Procédures stockées en PL/SQL i

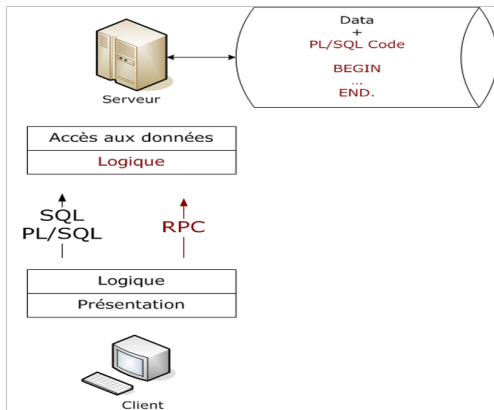


Figure 4 – Les procédures stockées favorisent l'architecture client-serveur de deuxième génération

On peut voir sur ce schéma différents avantages :

- Réutilisabilité : une même procédure ou fonction peut être appelée par plusieurs applications différentes
- Optimisation : le DBA peut utiliser toutes les techniques d'optimisation mises à sa disposition pour les procédures les plus utilisées
- Efficacité : une procédure stockée est compilée une fois et stockée sous une forme exécutable. Les appels peuvent être
- efficaces et se font sous forme de RPC (Remote Procedure Call)

En résumé :

- On obtient une **meilleure performance** et moins de trafic réseau ainsi que de meilleurs temps de réponse
- Les procédures stockées font partie de la mémoire cache de la base de données et peuvent être partagées entre les multiples utilisateurs
- L'**indépendance données-programmes s'en trouve renforcée** ! (Les détails du MRD peuvent être cachés aux différents clients. Il suffit de communiquer au développeur la procédure à appeler ainsi que ses différents paramètres et codes d'erreur)

Architecture PL/SQL

- Le moteur de base de données Oracle coordonne tous les appels en direction de la base de données. **Le SQL et le PL/SQL comportent chacun un moteur d'exécution associé.**
- Lorsqu'un serveur reçoit un appel pour exécuter un programme PL/SQL, **la version compilée du programme est chargée** en mémoire puis exécutée par les moteurs PL/SQL et SQL.
- Le moteur PL/SQL gère les structures en mémoire et le flux logique du programme tandis que le moteur SQL transmet les requêtes à la base de données.

- Un code PL/SQL peut être **appelé** à partir de **SQL*Plus**, Oracle Forms, Oracle Reports, inclus dans le langage hôte C, C++, Java,...
- Dans ce cas, les instructions PL/SQL sont traitées par le moteur PL/SQL embarqué dans l'outil de développement, les **ordres SQL étant évidemment toujours traités par la base de données**.