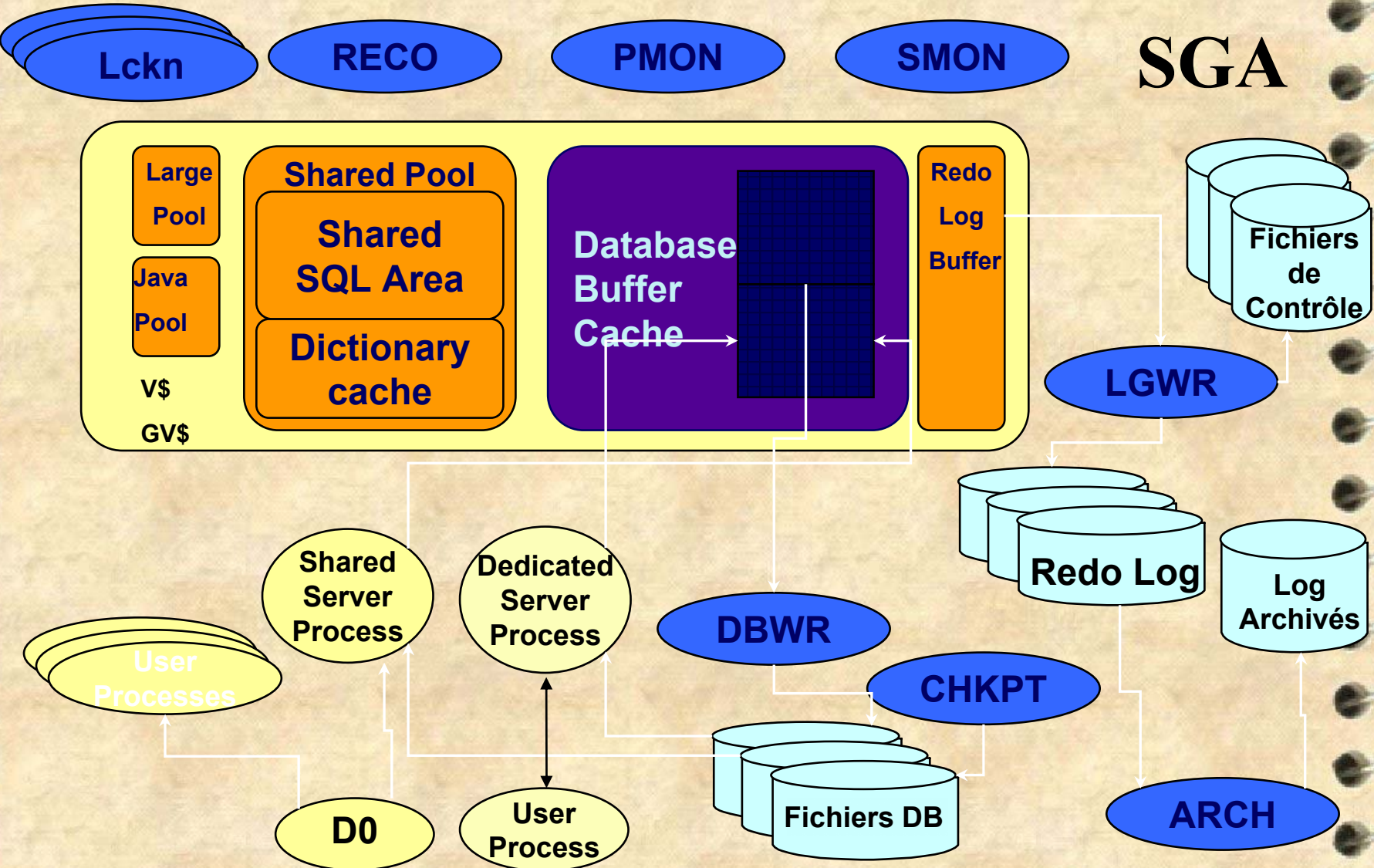


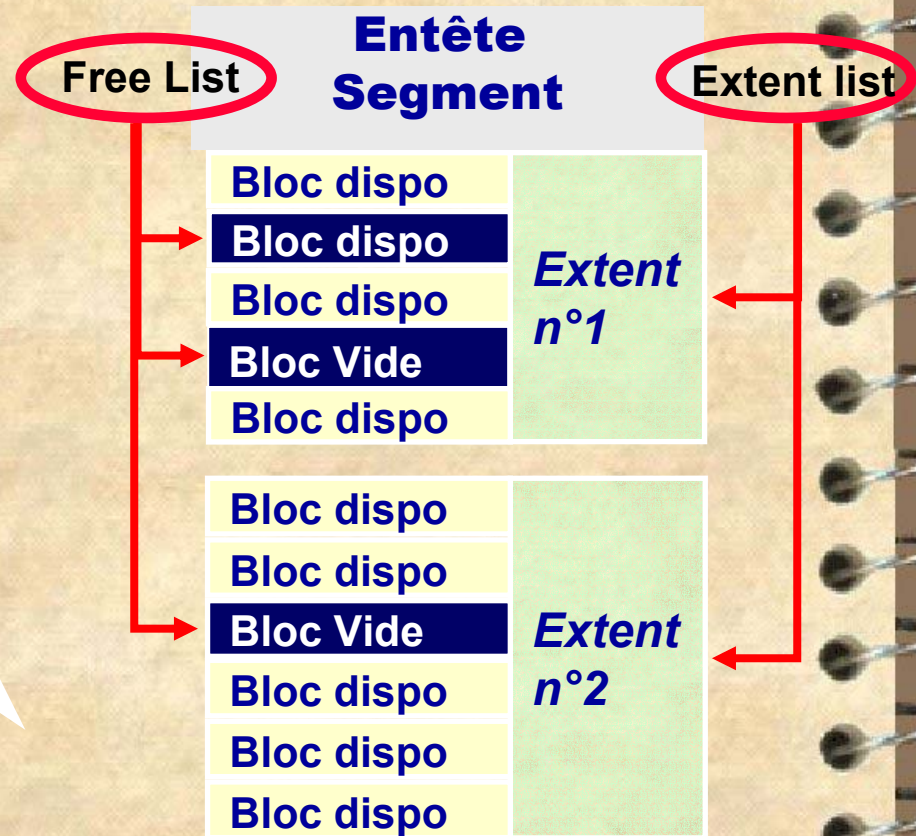
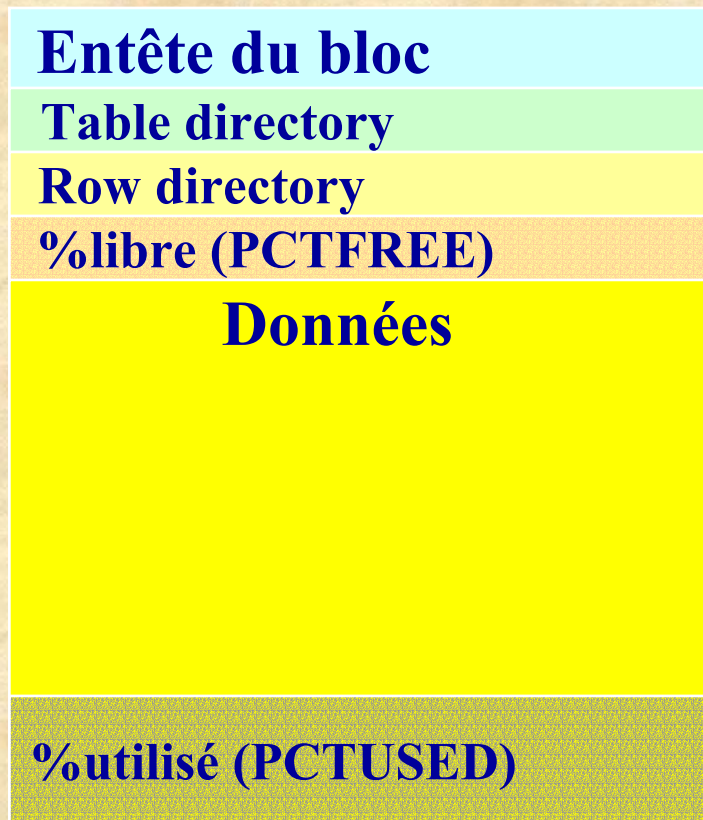
# Le tuning des Performances Oracle

TAFORA

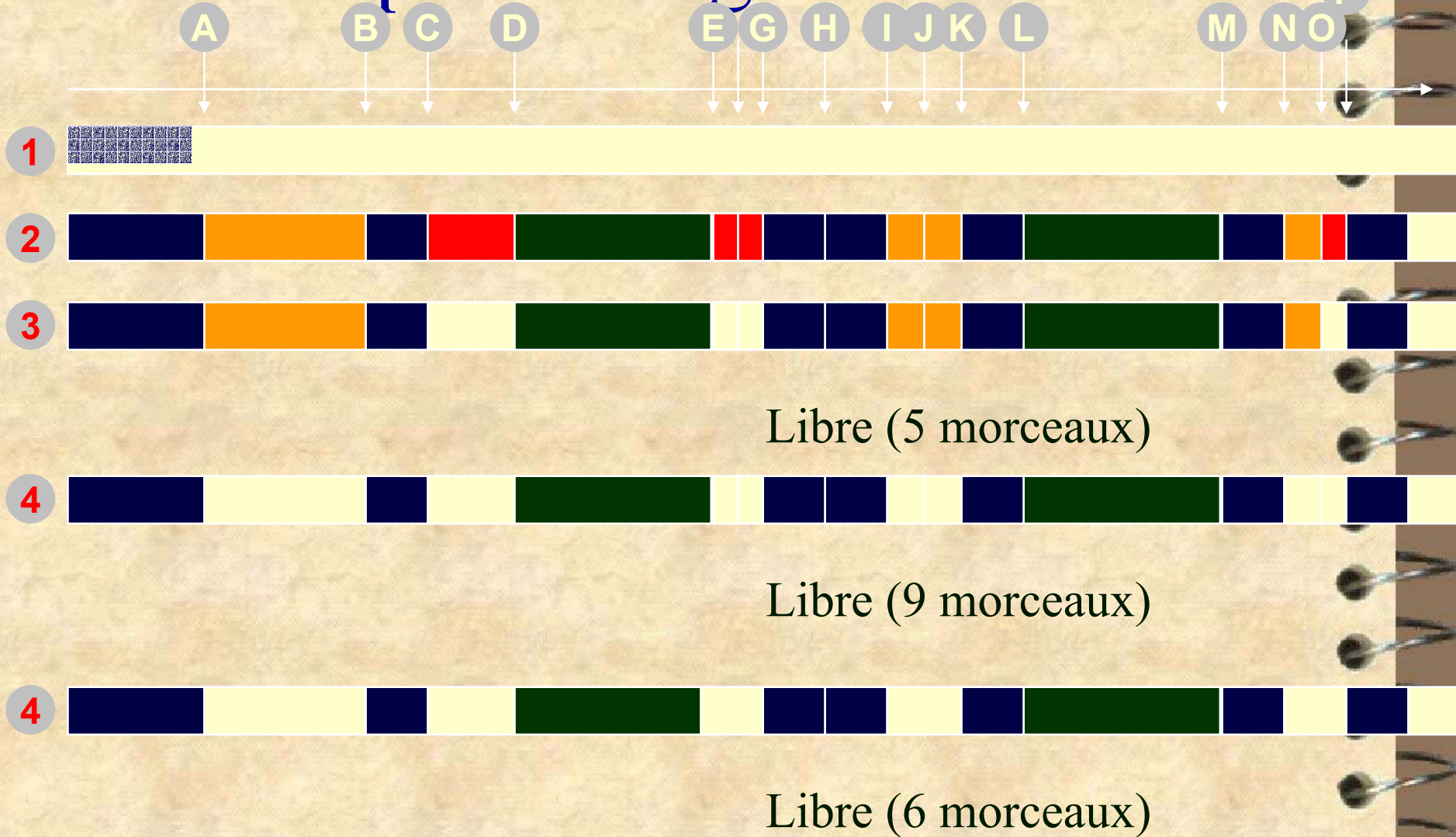
# L'environnement Oracle



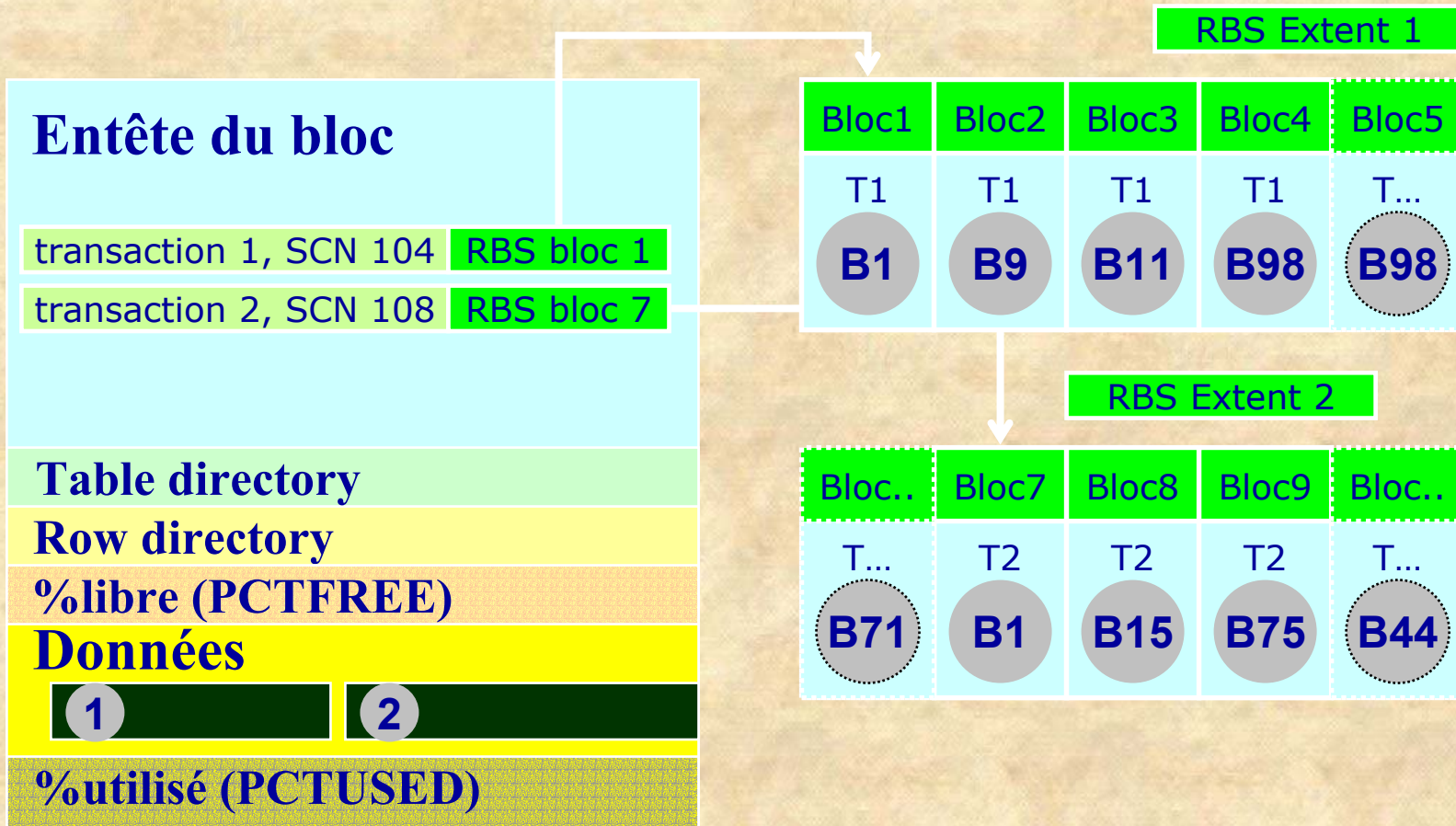
# Les blocs – leur statut



# Tablespace - Segments - Extents



# Les blocs – SCN, RBS, initrans



# *Problèmes de performances?*

## *Il peut bien s'agir de ...*

disque

contention

Buffer du dictionnaire

rollback segment

Buffer des données

fragmentation

sql area

index

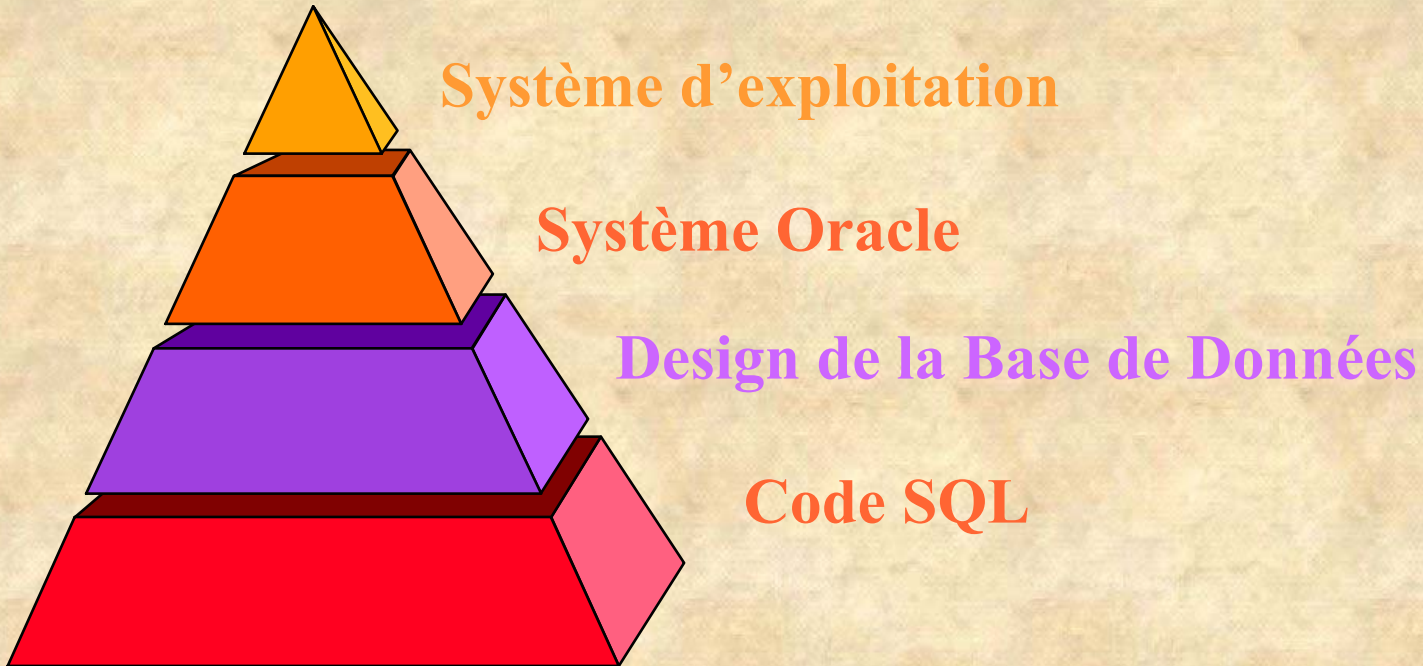
# Agenda

- Introduction
- Le tuning du système d'exploitation
- Le tuning du système Oracle
- Design
- Tuning de la base de données
- Code (SQL et PL/SQL)

# Introduction

- Par où commencer l'optimisation ?
- Quel est votre rôle ?
- Proactif
  - Mesurer périodiquement tous les paramètres
- Réactif
  - Optimiser en fonction des problèmes

# Introduction



# Mode Réactif

Par où commencer le tuning en Mode Réactif ?

La base est complètement ‘à genoux’

- OS , Instance, SQL

Une seule Application a des problèmes ?

- Analyse de la base, du SQL

Une seule Transaction a des problèmes ?

- Revisiter le code (SQL)

# Tuning OS

- Mémoire
- Contrôleur et Entrées/Sorties Disques
- CPU
- Réseau

# OS (Mémoire)

- La mémoire est la PLUS SENSIBLE des ressources dans les opérations de tuning
- Elle affecte le système entier d'une manière dramatique
- La première ressource revue lors de l'analyse des performances.

# OS (Mémoire)

## Règles de base :

- Toujours plus de mémoire, tant que ça ne swappe pas.
- Ne jamais descendre au dessous de 5% de mémoire libre
- Si plus de 10% de mémoire sont libres aux périodes de pointe, alors la mémoire n'est pas utilisée de manière efficace.
  - Affecter plus de mémoire à toutes les zones Caches d'Oracle (détails plus loin)

# OS (I/O)

- Les E/S disques affectent Oracle et le système entier si elles ne sont pas optimisées
- Séparer les disques du système d'exploitation (paging et fichiers système) des disques contenant les datafiles d'Oracle
- Placer l'index sur un autre disque que la table
- Placer les Redo Logs sur des disques séparés
- Placer les Archive Logs sur des disques séparés

# OS - CPU

- Éliminer le Swapping / Réduire le Paging
- Éviter les traitements batch pendant les périodes de pointe

# OS - Réseau

- Si de longs batchs sont exécutés, garder la majorité du travail sur le serveur.
- Si des données ou des rapports sont souvent transférés, utiliser les utilitaires de compression des données.
- Si les utilisateurs déchargent fréquemment des données, envisager les extractions de nuit.
- Pour éviter le trafic volumineux, utiliser les packages, procédures, triggers, fonctions, et contraintes

# Tuning du Système Oracle

- Mémoire
- Entrées/Sorties Disques
- CPU
- Verrouillage
- Contention

# Systeme Oracle

## Mémoire

- La Mémoire est la PLUS SENSIBLE des ressources dans les opérations de tuning d'une Instance Oracle
- La mémoire et les zones de Cache Oracle affecte la performance générale de toutes les applications
- Maîtrisée directement par les paramètres INIT.ORA (spfile.ora) associés

# Systeme Oracle

## Mémoire

- Shared Buffer Cache
  - SHARED\_POOL\_SIZE contrôle la taille du Cache Dictionary/Library et du Cache Shared SQL
  - A surveiller régulièrement pour s'assurer qu'il a la taille adéquate
- Buffer Cache
  - DB\_BLOCK\_BUFFERS (db\_cache\_size) contrôle la zone de Cache données
  - A surveiller régulièrement pour s'assurer qu'il est correctement taillé

# Systeme Oracle

## Mémoire

- Log Buffer
  - LOG\_BUFFER en contrôle la taille
  - Augmenter la taille s'il y a des problèmes d'Entrées/Sorties sur les disques contenant les Redo Logs
  - Augmenter la taille si des traitements batch sont exécutés
- Les Sessions Utilisateurs
  - Contrôler la mémoire par utilisateur avec le RESOURCE\_LIMIT
  - Valoriser le SORT\_AREA\_SIZE en fonction des besoins des traitements

# Systeme Oracle

## Entrées/Sorties Disques

- Redo Logs
  - Séparer les Redo Logs des fichiers Database et des Archive Logs
  - Ventiler les Redo Logs alternés sur différents disques
  - Utiliser le mirroring pour les Archive Logs
- Rollback Segments
  - Répartir les Rollback Segments sur les disques
  - Mettre les Rollback Segments dans des Tablespaces séparés et les assigner en séquence
  - Utiliser le Striping pour les Tablespaces de Rollback Segments

# Systeme Oracle

## Entrees/Sorties Disques

- Faire résider les packages gourmands en mémoire
  - Les packages STANDARD, DBMS\_STANDARD, DBMS\_UTILITY, DBMS\_DESCRIBE, DBMS\_OUTPUT doivent être résidents.
- Créer un Tablespace TEMPORARY séparé
  - S'assurer que la taille des extents «initial» et «next» sont des multiples de la SORT\_AREA\_SIZE (+ 1bloc)
- Créer un Tablespace séparé pour les produits

# Systeme Oracle

## CPU

- S'assurer que les zones Cache sont suffisantes pour éviter les mouvements entre le cache et le disque.
- Séparer les traitements batchs Oracle du traitement OLTP
- Surveiller et identifier les longues requêtes SQL puis les optimiser

# Systeme Oracle

## Verrouillage

- Les verrous sont levés SEULEMENT lors des commits ou des rollbacks
  - S'assurer que les utilisateurs n'éteignent pas leurs station de travail pendant le traitement d'une transaction
  - Utiliser les Resource Profiles pour gérer le temps «idle»
- Verrou Mortel.
  - Eviter les situations de Deadlock
  - Employer une stratégie de verrouillage au niveau du système entier.

# Systeme Oracle

## Verrouillage

- Si les utilisateurs se plaignent de temps de réponse inégaux, la raison en est habituellement due au verrouillage.
- Surveiller continuellement les conditions de lock wait.

# Systeme Oracle

## Contention

- Rollback Segments
  - Augmenter le nombre de Rollback Segments lorsque vous avez des problèmes de contention
  - Ne jamais faire confiance à OPTIMAL pour rétrécir vos Rollback Segments, les tailler correctement
- Buffer de Redo Log
  - Augmenter la taille s'il y a contention sur l'espace
  - Réduire le latch contention sur les machine à CPU multiples

# Design & Tuning des Bases de Données

- Table
  - Quand dénormaliser ?
  - Design de tables volumineuses
  - Chargement volumineux de données
- Index
- Contention
- Fragmentation

# Design & Tuning des Bases de Données

- Table
  - Quand dénormaliser ?
  - Design de tables volumineuses
  - Chargement volumineux  
de données
- Index
- Contention
- Fragmentation

# Design des Bases de Données

## Design des Tables

- Quand dénormaliser ?
  - Jointure de 5 tables ou plus
  - Plusieurs lignes représentées par une seule (un tableau dans une ligne)
  - Créer des tables sommaires
  - Coder en dur les données statiques qui sont fréquemment jointes
  - Enregistrer les totaux calculés.
  - Remplacer les Primary Keys naturelles par les System Assigned Keys
  - Combiner les tables de codes
  - Faire des extractions de nuit pour supporter le reporting batch de jour

# Design des Bases de Données

## Design des Tables

- Tables Volumineuses
  - Toujours concevoir avec un critère de purge en tête.
  - Utiliser soit les vues partitionnées de la 7.3 soit les tables partitionnées de la 8.x
- Exemple de Partitionnement
  - Compagnie d'assurances - Informations liées aux réclamations
  - Chaque année est enregistrée dans une table / partition séparée
  - Chargement mensuel seulement dans la table(partition) de l'année courante

# Design des Bases de Données

## Design des Tables

- Chargements volumineux de données
  - Si les données sont chargées avec SQL\*Loader, utiliser le «direct path»
  - Si l'Import est utilisé, augmenter le paramètre Buffer, et utiliser DIRECT
  - Utiliser le chargement de table, et la construction d'index, PARALLEL
  - Considérer l'utilisation de l'option UNRECOVERABLE (NOLOGGING)
  - Si Oracle 8+ est utilisé, tirer partie des Inserts et des Updates parallèles

# Design des Bases de Données

## Design des Index

- Design des Index
  - Indexer les colonnes avec un Access Path lourd
  - Toujours balancer les statements Select contre les statements Insert, Update, et Delete
  - Indexer les clés étrangères pour éviter les locks globaux de la table
  - Indexer les colonnes composites pour une meilleure sélectivité

# Design des Bases de Données

## Contention

- Contention de Free List
  - Les Free Lists sont allouées dans l'entête de segment
  - Les applications à fort taux d'Insert peuvent générer des contentions sur les Free Lists
  - S'assurer que les Free Lists peuvent s'accommoder du nombre d'Inserts

# Design des Bases de Données

## Fragmentation

- Surveiller et vérifier la raison de fragmentation de vos tables:
  - Suppression de lignes
  - Migration de lignes
  - Allocation d'Extents Dynamique
- Balancement des Index B-Tree
- La solution:
  - UNLOAD, DROP, RECREATE, RELOAD

# Codage de l'application & SQL

- Codage de l'application
- SQL
  - Parsing
  - Optimiseur
  - Tuning

# Codage de l'application

- TOUJOURS essayer d'utiliser les Procédures Stockées
  - Moins de trafic réseau
  - Pré-chargement des procédures et fonctions associées
  - Les variables de packages statiques peuvent être utilisées
  - La sécurité peut être affectée au niveau Package
  - Rendre les Packages les plus fortement utilisés, résidents en mémoire.
  - Attention aux gros Packages qui dominant le «shared pool»

# Codage SQL

## Parsing

- Le processus de parsing SQL
  - Vérifie la syntaxe
  - Recherche dans le «Shared pool»
  - Recherche dans le dictionnaire
  - Détermine le chemin d'accès
  - Sauvegarde le plan d'exécution
- S'assurer que la recherche en «Shared Pool» trouve le SQL
  - Renforcer les standards SQL
  - Encourager l'utilisation des «bind variables»

# Codage SQL

## Optimiseur

- Optimiseur basé sur le coût
  - L'évaluation est souple et les changements sont basés sur les statistiques de l'Analyse
  - L'évaluation est centrée sur le total de lectures logiques
    - Meilleur débit - ALL\_ROWS
    - Meilleur temps de réponse - FIRST\_ROWS
  - La sélectivité des index non-unique est calculée
  - L'évaluation du processus «fourchette» utilise les valeurs haute et basse des colonnes
  - Les Hints peuvent être utilisés pour influencer l'optimiseur

# Codage SQL Tuning

- Utiliser l'information suivante pour déterminer le chemin d'accès
  - Le statement SQL et les prédicats
  - Les Index sur les tables référencées
  - L'Explain Plan
  - Le mode actuel de l'optimiseur
  - Les statistiques sur les tables
  - Disponibilité des Hints

# Les tris

- Trier nécessite de la mémoire.
- Paramétrer Sort Area.
- Bien paramétrer les tablespaces temporaires.

# Les locks

- Oracle maintient la concurrence d'accès, l'intégrité des données ainsi que leur consistance en utilisant nombreux types de verrous.
- Les verrous sont libérés uniquement au commit ou au rollback.

# SQL

- Utilisez le shared pool intelligemment.
- Modifier le cache SQL.
- Comment Oracle accède aux blocs de données ?

# Autres solutions ?

- Partitionnez
- Automatiser l'Export/Import
- Défragmentez

# Maximiser les Performances

- Ne faites que des sauvegardes nécessaires
- Pensez tablespace temporaire lors de la reindexation
- Prealouer l'espace de stockage
- Pretriez les données d'après le plus gros index
- Créez les index séparément

# Fragmentation

- Comprendre le bon et le mauvais de la fragmentation
- Reconstruire si nécessaire
- Trouver une démarche automatisée de la défragmentation et de la reconstruction

# Démarche de suivi d'activité

- 1. Étudier les fichiers traces**
- 2. Suivre la consommation d'espace**
- 3. Vérifier l'intégrité avant les sauvegardes et après de gros traitements**
- 4. Suivre l'activité globale du serveur Oracle**

# Les utilisateurs

- Les utilisateurs Oracle veulent toute la base pour eux.

# Comprendre les structures physiques

- Comprendre l'état des objets
- Interroger souvent les tables dynamiques.
- ANALYZER les tables.
- Approfondir les compétences.
- Réinterroger les tables dynamiques.  
(Quelque chose a pu changer).

# Jointures grammaticales

Peut-être vous lisez à l'envers et du haut vers le bas !

!

le bas

vers

à l'envers

et du

vous

lisez

Peut-être

haut