

# Oracle : Architecture et concepts de base

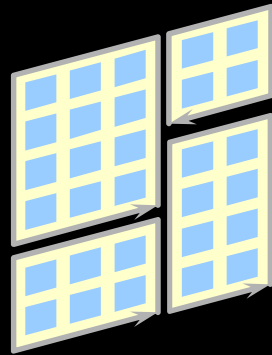


Pr. Omar El Beqqali

omar.elbeqqali@usmba.ac.ma

**elbeqqali.olymppe.in**

# Oracle Enterprise Edition 10G, 11G...



Partitionnement



Real Application  
Clusters



Oracle Enterprise  
Manager Packs

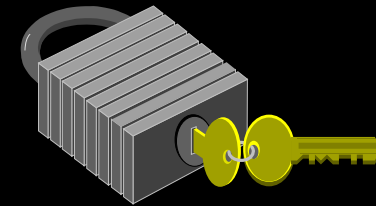


Advanced  
Security

ORACLE®

## Caractéristiques ORACLE (++)

- Performance
- Fiabilité
- Disponibilité sur toutes les plate-formes
- Capacités d'extension (cluster,...)
- Connexion et Sécurité
- Portabilité
- Fonctionnalités étendues

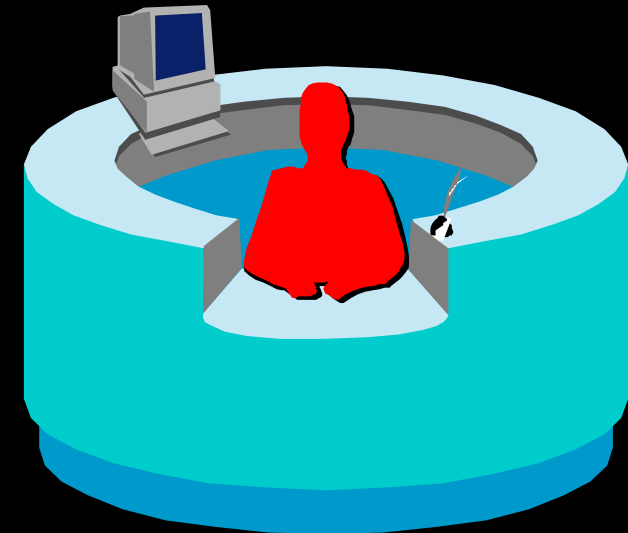


## Caractéristiques (--)

- Prix
- Complexité de mise en œuvre
- Gourmandise système

# Tâches de l'administrateur de base de données

- Planification et création de bases
- Gestion de la disponibilité des bases
- Gestion des structures physiques et logiques
- Gestion de la sécurité
- arrêter et démarrer la base
- Sauvegarde et récupération
- Réglage des bases
- Contrôler et optimiser les performances



# Sommaire

● L'architecture d'Oracle	● Les fichiers journaux
● Les transactions	● Les espaces de disque logiques
● Les processus d'arrière-plan	● La gestion automatique des fichiers
● Les outils d'administration (SQL*Plus et iSQL*Plus)	● La gestion du stockage
● L'architecture Oracle Net	● Les segments UNDO
● Oracle Enterprise Manager	● Les types de données
● L'installation d'Oracle	● La création des tables
● La gestion d'une instance	● La gestion des tables
● La création d'une base de données	● Les index
● Dictionnaire de données	● Les vues et autres objets
● Le fichier de contrôle	● Les profils
	● Les utilisateurs
	● Les privilèges

# PLAN

## Architecture d' Oracle

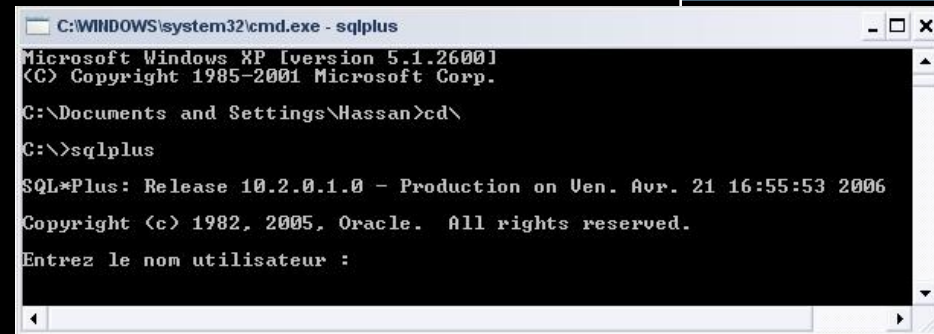
La base de données

L' instance

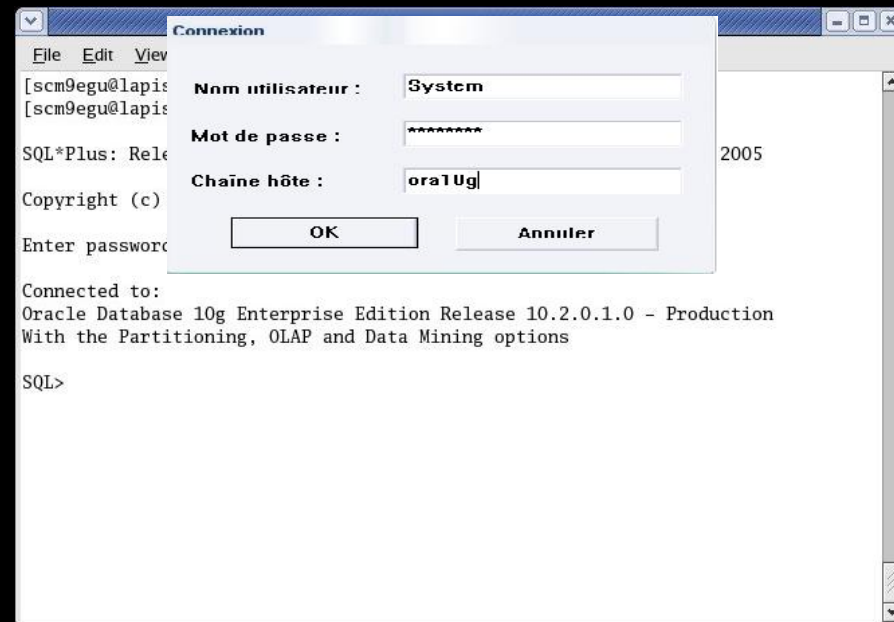
La SGA et la PGA

# Les méthodes de connexion

- SQL\*Plus *Ligne de commande*
  - sqlplus



- SQL\*Plus *Windows*
  - sqlplusw



# Les méthodes de connexion

## iSQL\*Plus

Version Web, accessible à partir d'un navigateur.

<http://localhost:port/isqlplus>



ORACLE  
iSQL\*Plus

Connexion

L'utilisation non autorisée de ce site est interdite et peut faire l'objet de poursuites judiciaires.

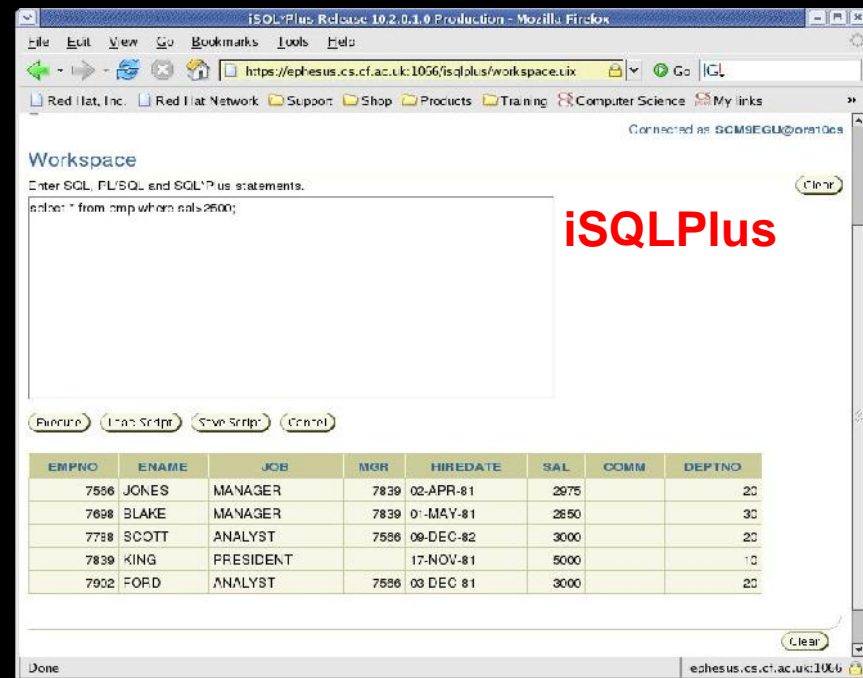
\* Indique un champ obligatoire

\* Nom utilisateur: system

\* Mot de passe:

Identificateur de connexion: ora10g

Connexion



iSQL\*Plus Release 10.2.0.1.0 Production - Mozilla Firefox

Workspace

Enter SQL, PL/SQL, and SQL\*Plus statements.

sql> \* from emp where sal > 2500;

Clear

Execute Run Script Save Script Cancel

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7902	FORD	ANALYST	7566	03-DEC-81	3000		20

Done

Clear

ephesus.cs.cf.ac.uk:1066



# Utilisateurs SYS et SYSTEM

Comptes créés automatiquement avec rôle DBA

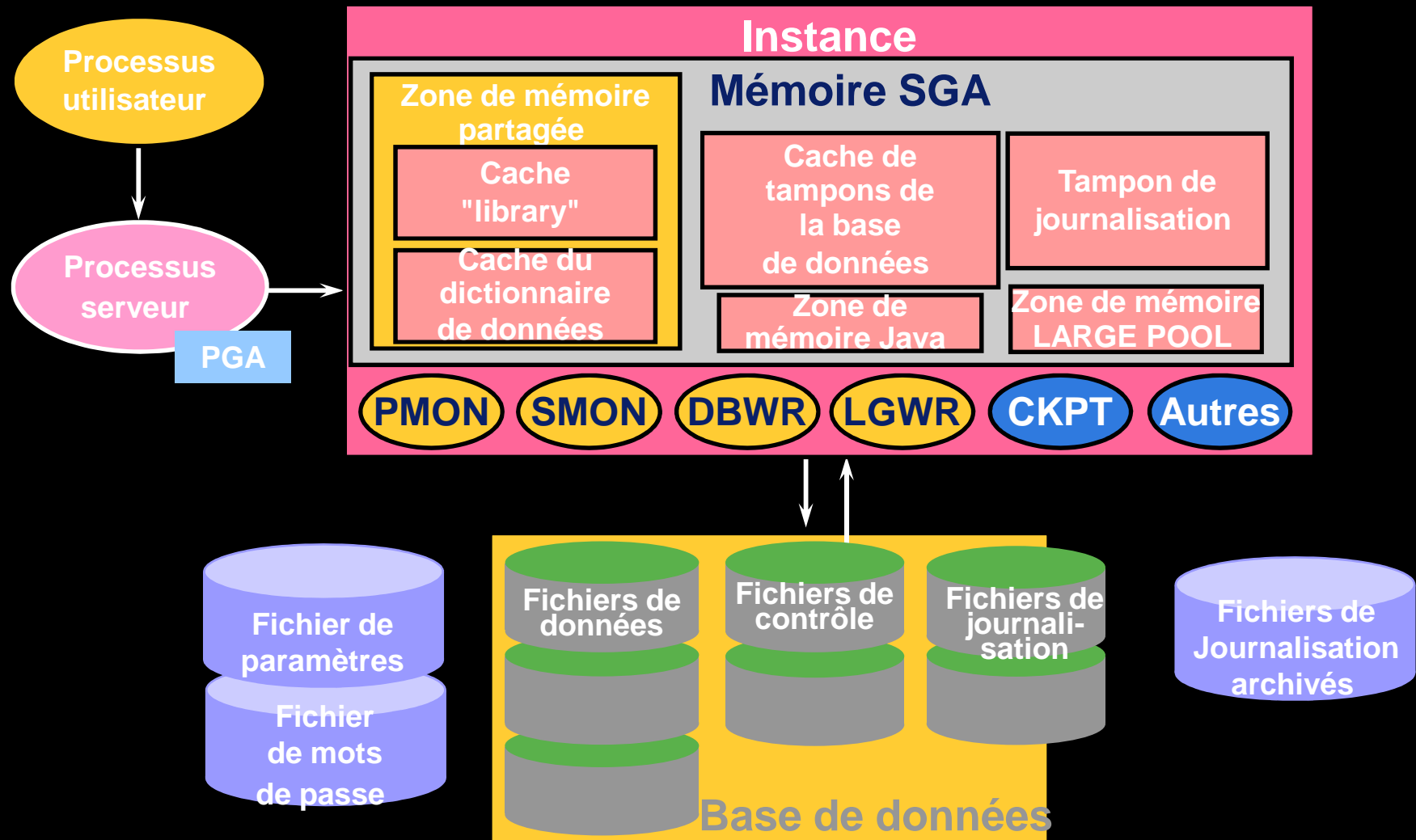
## ■ **SYS**

- **Mot de passe :  
change\_on\_install**
- ***Propriétaire du  
dictionnaire de  
données de la  
base***

## ■ **SYSTEM**

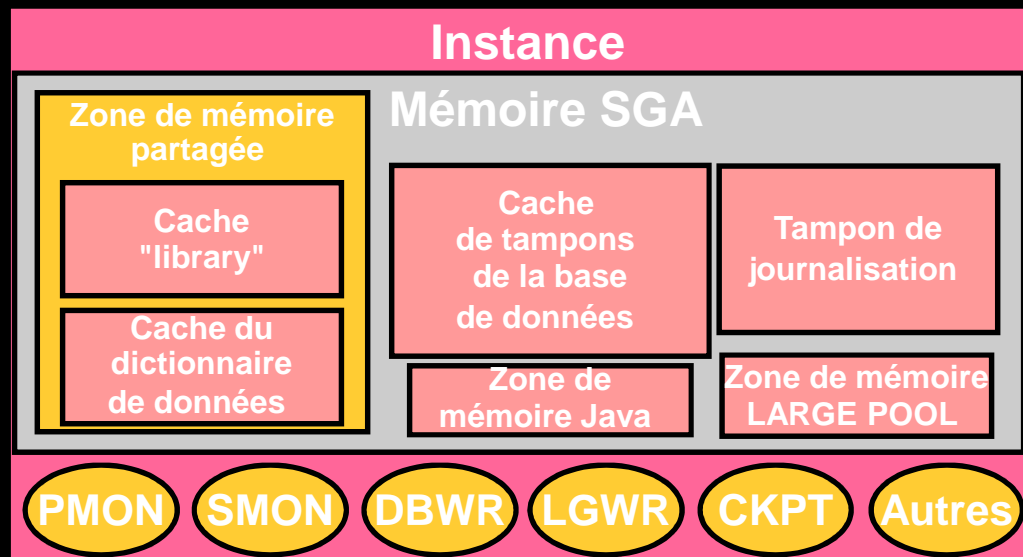
- **Mot de passe :  
manager**
- ***Propriétaire de tables  
internes  
supplémentaires  
utilisées par les outils  
Oracle***

# Présentation des principaux composants



# Instance Oracle

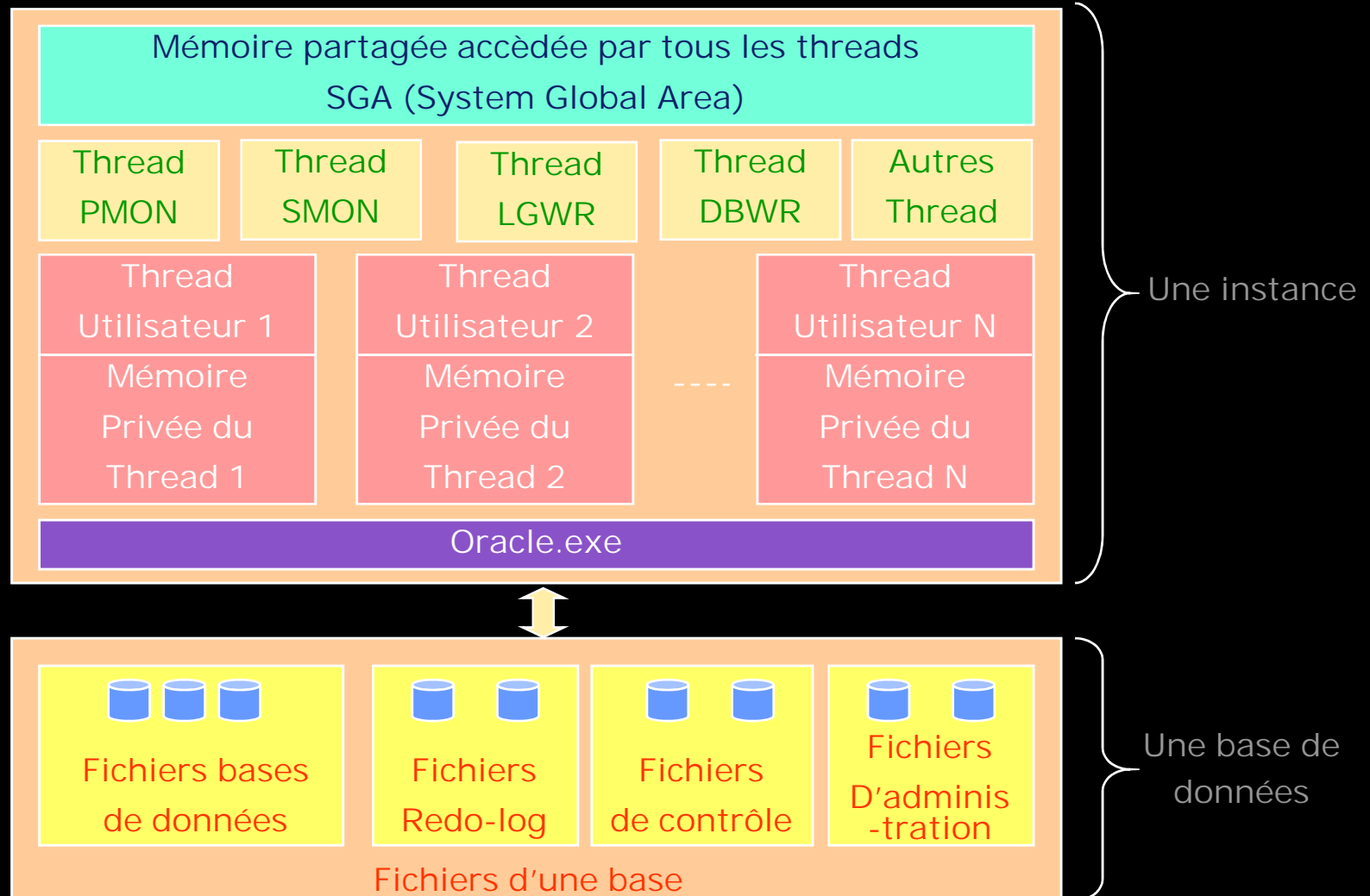
- Une instance Oracle :
  - permet d'accéder à une base de données Oracle,
  - n'ouvre qu'une seule base de données,
  - est constituée de structures de processus d'arrière-plan et de structures mémoire.



Structures  
mémoire

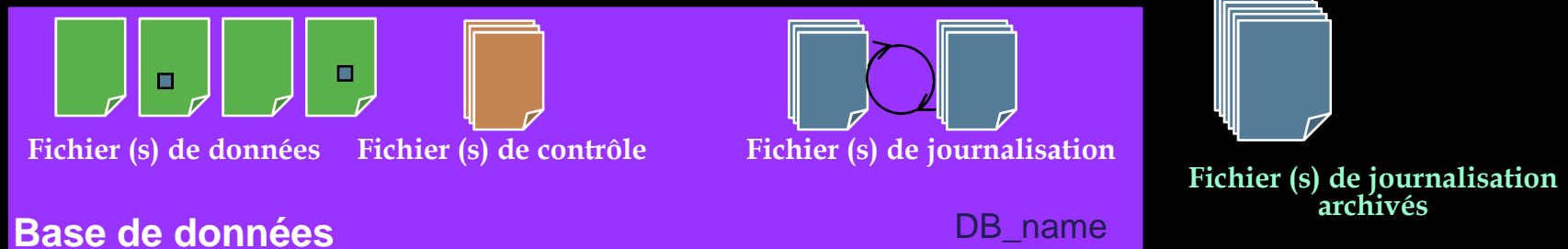
Structures  
de processus  
d'arrière-plan

# Base de données et instance



## Présentation générale (2)

### ■ La base de données



#### ■ **est constituée:**

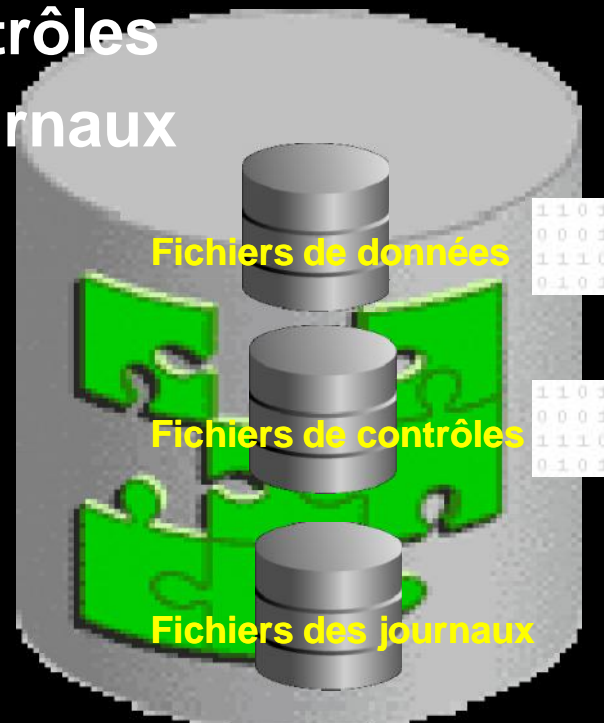
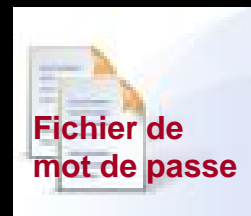
- Un ou plusieurs fichiers de données qui contiennent les données proprement dites.
- D'au minimum un fichier de contrôle qui contient des informations de contrôle sur la BD.
- D'au minimum deux groupes de fichiers de journalisation qui enregistrent toutes les modifications apportées à la base.

#### ■ **Les différentes catégories de base de données**

- BD transactionnelles (ou OLTP pour OnLine Transaction Processing)
- BD décisionnelles (ou DSS pour Decision Support Systems).
- Mixtes (qui sont à la fois OLTP et DSS).

# La base de données

- Une base de données Oracle est l'ensemble de 3 types de fichiers.
  - Les fichiers de données
  - Les fichiers de contrôles
  - Les fichiers des journaux



- Une base de données Oracle est désignée par le nom de la BD, avec le paramètre « db\_name » non modifiable.

■ `sql> show parameter DB_NAME;`

# La base de données

## ■ Fichier de contrôle

- Le nom de la base de données.
- La date/heure de création de la BD.
- Le nom des fichiers constitutifs de la base.
- Les noms des espaces de disques logiques.
- La taille de bloc de données par défaut.
- L'emplacement des autres fichiers de la BD (contrôle, journalisation).
- Le numéro de séquence du fichier journal en cours d'utilisation;
- Des informations sur les points de reprise(contrôle) (*checkpoint*).
- L'emplacement des archives.

# La base de données

## ■ Fichier de contrôle

### ■ L'information du fichier de contrôle

- V\$CONTROLFILE: *affiche l'ensemble des fichiers de ctrl de la BD*
- V\$PARAMETER: *permet d'afficher la valeur du paramètre « control\_files »*
- V\$CONTROLFILE\_RECORD\_SECTION: *affiche l'ensemble des informations concernant les enregistrement dans les fichiers de contrôle de la BD.*

### ■ Le multiplexage

- CONTROL\_FILES = 'CONTROL01.CTL','CONTROL02.CTL'
- En raison de l'importance du fichier de contrôle, Oracle permet de multiplexer ce fichier pour en avoir plusieurs copies, afin d'éviter toute corruption ou perte du fichier.

```
Sql> show parameter control_files
```

Name	type	value
-----		
Control_files	string	
		'C:\oracle\oradata\dba\CONTROLFIE\CONROL01.CTL'

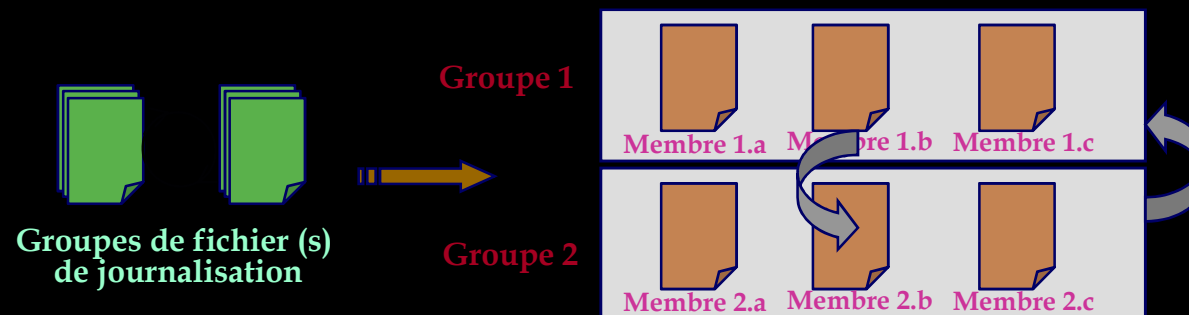
```
Sql> ALTER SYSTEM SET CONTROL_FILES= 'C:\.....\CONTROL01.CTL',  
'D:\.....\CONTROL02.CTL' SCOPE=SPFILE;  
Système modifié.
```



# La base de données

## ■ Fichier de journalisation (redo Log)

- Ils enregistrent toutes les modifications apportées à la BD;
- Ils sont organisés en groupes écrits de manière circulaire, les informations sauvegardées sont donc périodiquement écrasées.
- Ils sont utilisés pour la récupération de l'instance ou ses fichiers après un arrêt anormal.
- Ils sont organisés en groupes (au min 2) composés d'un ou plusieurs membres (min un) et ils sont créés lors de la création de la BD. À l'intérieur d'un groupe, **les membres sont écrits simultanément en miroir par l'instance Oracle (processus LGWR) et contiennent la même information.**



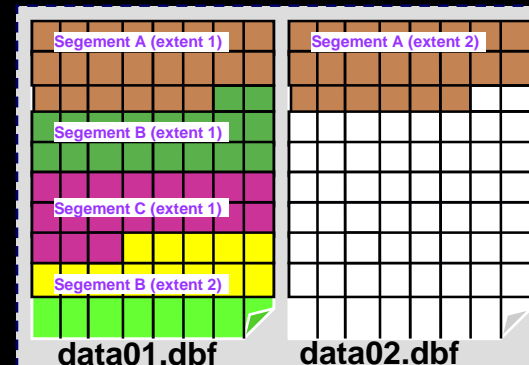
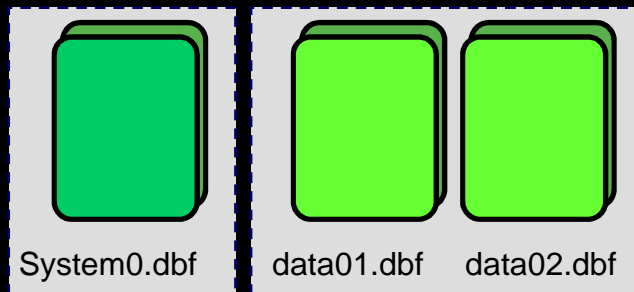
- chaînés circulairement ; ce qui signifie l'un se remplit, puis l'autre, puis le premier, etc
- Les fichiers de journalisation sont très importants pour la sécurité de la BD.
- **Gérés à 100% par Oracle**

# La base de données

## ■ Fichier de données

### Définition

Les fichiers de données contiennent les données proprement dites de la BD (tables et index notamment). Ils sont logiquement regroupés en tablespaces.



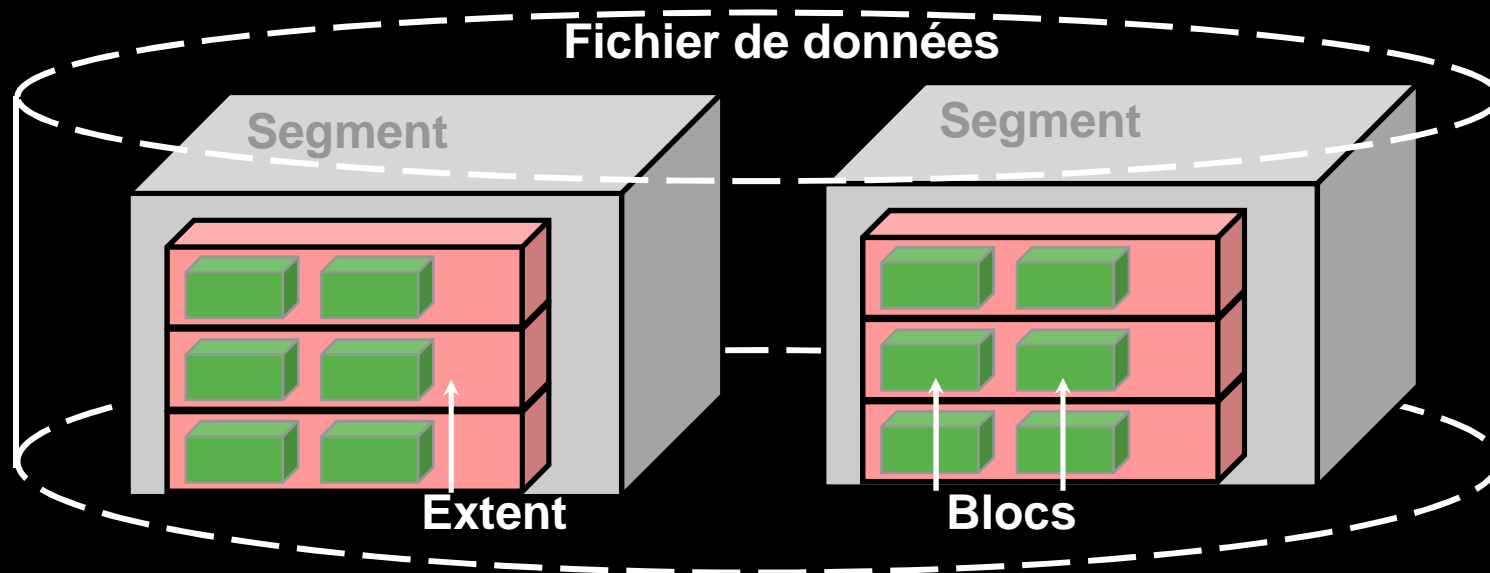
■ Bloc Oracle  
Blanc = libre  
Autre couleur = alloué

- **Un tablespace est une unité logique de stockage composée d'un ou plusieurs fichiers physiques.**
- Une base de données comporte au minimum deux fichiers de données appartenant à deux tablespaces réservés pour Oracle (**SYSTEM** et **SYSAUX**) qui ne doivent contenir aucune donnée applicative.

# Structure logique

- La structure logique définit le mode d'utilisation de l'espace physique d'une base de données.
- Cette structure possède une hiérarchie composée de tablespaces, de segments, d'extents et de blocs.

## Tablespace



## Tablespaces SYSTEM et Non-SYSTEM

### ■ Tablespace **SYSTEM**

contient :

- Des informations relatives au dictionnaire de données
- Un rollback segment SYSTEM

- Tablespace **Temporaire** :  
Utilisé pour les opérations de tri  
Ne peut pas contenir d'objets permanents

### ■ Tablespace **Non-SYSTEM**

contient :

- Les Rollback segments
- Les segments temporaires
- Les données applicatives
- Les index

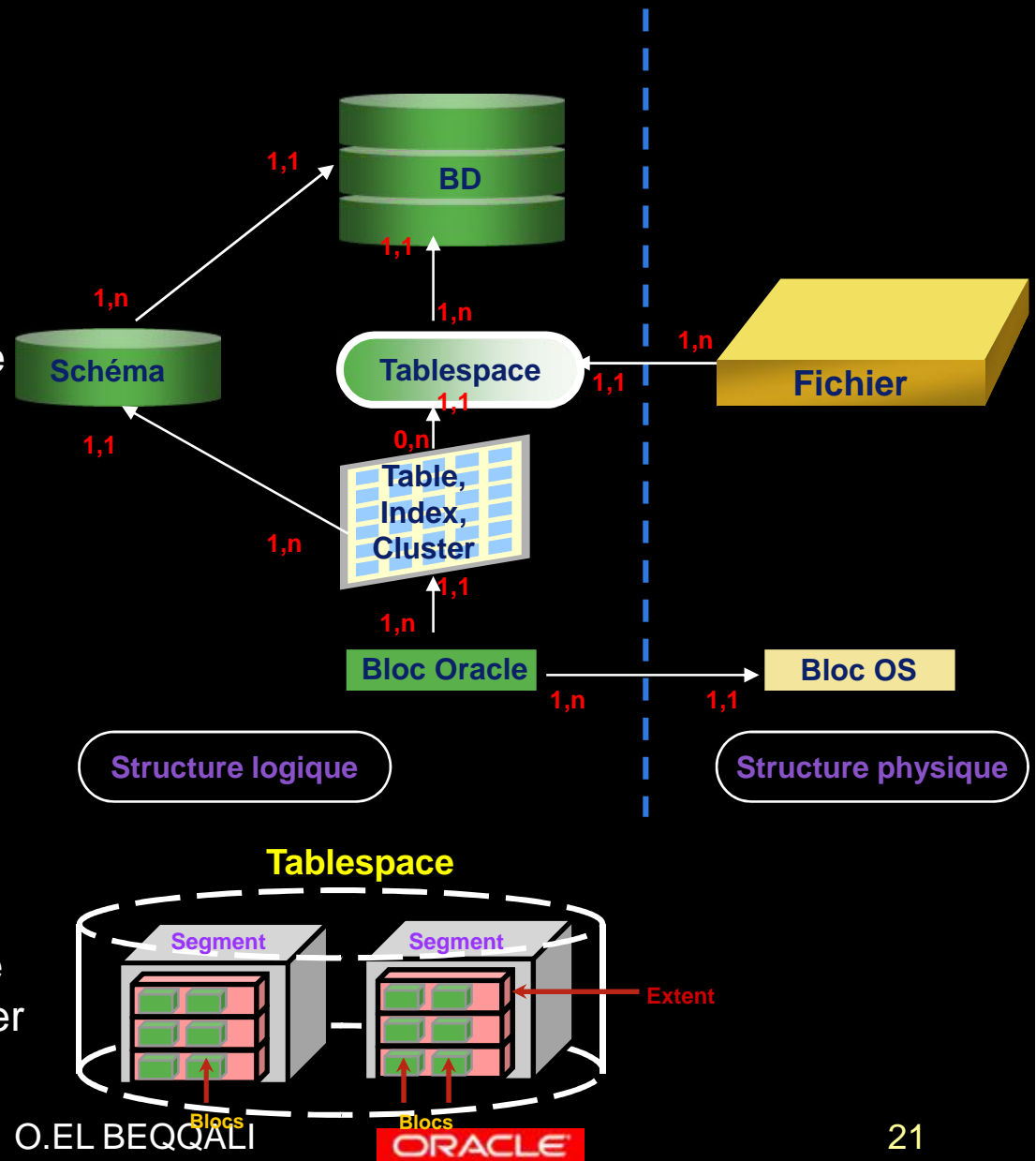
# La base de données

## ■ Organisation du stockage

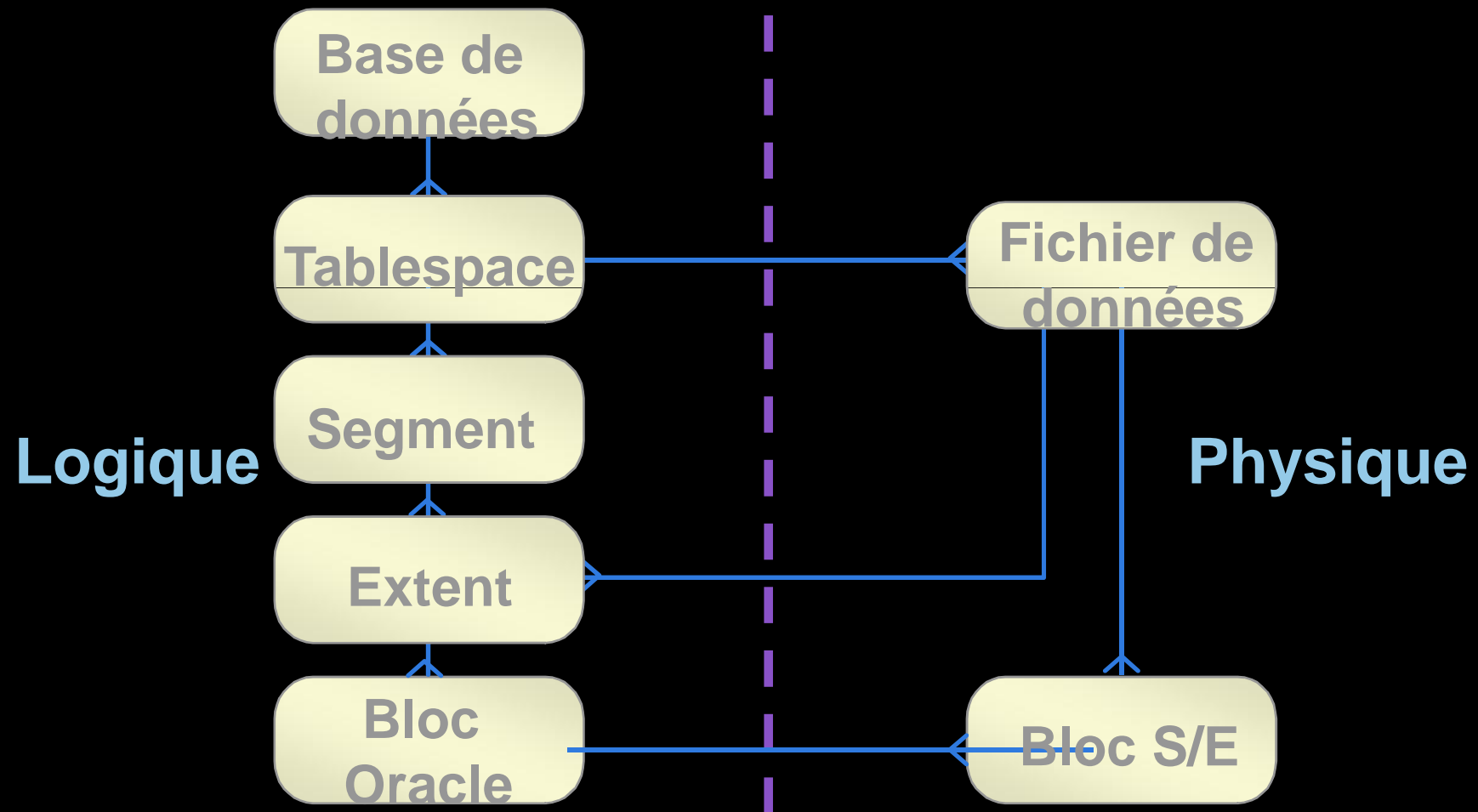
- Les fichiers de données sont découpés en blocs d'une taille donnée (4 ko, 8 Ko, ...)
- L'espace occupé par un objet dans un tablespace est désigné par le terme générique de segment. Il y a 4 types de segments:

- **Les segments de table**
- **Les segments d'index**
- **Les segments d'annulation**
- **Les segments temporaires.**

- Un segment appartient à un tablespace et est constitué d'extensions (extents).
- Une extension est un ensemble de blocs contigus dans un fichier de données.



# Structure Logique d'une Base de Données



# L'instance

## ■ La SGA (Vue d'ensemble)

### Définition

La **SGA (System Global Area)** est une zone de mémoire partagée par les différents processus de l'instance. Elle est allouée au démarrage de l'instance et libérée lors l'arrêt de l'instance. Elle est dimensionnée par un ensemble de paramètres définis dans le fichier de paramètres.

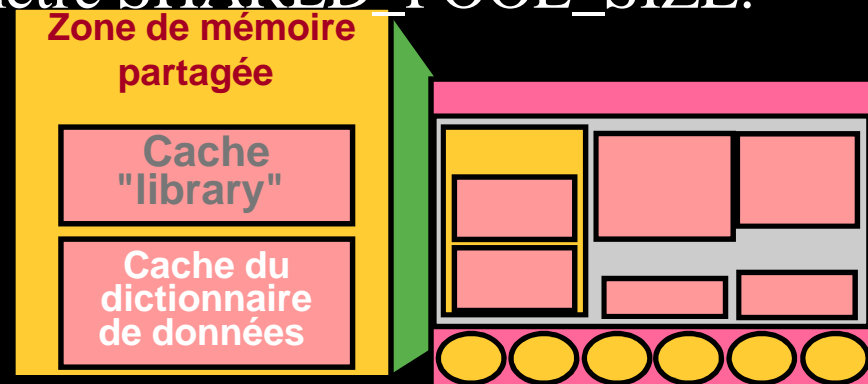
## ■ La SGA comporte les structures suivantes:

- **Database Buffer Cache:** cache de données;
- **Redo Log Buffer:** mémoire tampon pour l'enregistrement des modifications apportées à la BD;
- **Shared Pool:** zone de partage des requêtes et du DD Oracle;
- **Java Pool:** mémoire utilisée par la machine virtuelle Java intégrée;
- **Large Pool:** zone mémoire optionnelle utilisée par différents processus dans des configurations particulières;
- **Streams Pool:** zone de mémoire utilisée par la fonctionnalité Streams (fonctionnalité qui permet de faire circuler des informations entre processus);
- La SGA contient aussi une structure SGA fixe qui contient des informations sur l'état de la BD et le l'instance, et sur les verrous. Elle n'est dimensionnée pae le DBA; sa taille est faible qlq Ko.
- Taille définie à l'aide du paramètre SGA\_MAX\_SIZE

## L'instance

- Shared Pool (Zone de mémoire partagée)
  - Elle permet de stocker :
    - les dernières instructions SQL exécutées,
    - les dernières définitions de données utilisées.
  - Elle est constituée de deux structures mémoire clés liées aux performances :
    - Cache "library"
    - Cache du dictionnaire de données
  - Sa taille est définie par le paramètre `SHARED_POOL_SIZE`.

```
ALTER SYSTEM SET  
SHARED_POOL_SIZE = 64M;
```





# L'instance

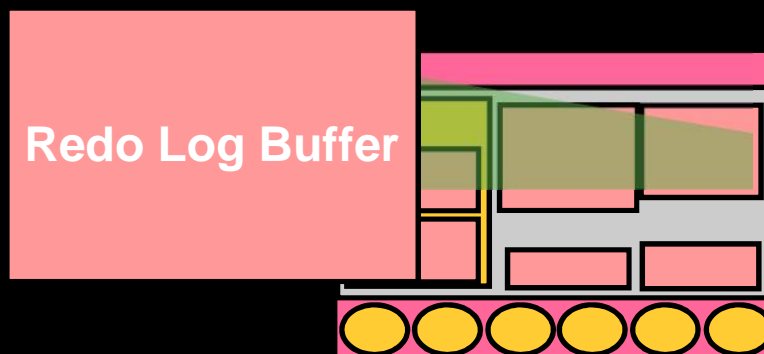
- Cache « Library »
  - Le cache "library" conserve des informations sur les dernières instructions SQL et PL/SQL utilisées.
  - Il permet le partage des instructions fréquemment utilisées.
  - Il est géré par un algorithme LRU.
  - Il est composé de deux structures :
    - la zone SQL partagée,
    - la zone PL/SQL partagée.
  - Sa taille dépend du dimensionnement de la zone de mémoire partagée.

## L'instance

- Cache «dictionary»: Cache du dictionnaire de données
  - Le cache du dictionnaire de données contient les dernières définitions utilisées dans la base.
  - Il contient des informations sur les fichiers, les tables, les index, les colonnes, les utilisateurs, les privilèges et d'autres objets de la base de données.
  - Au cours de l'analyse, le processus serveur recherche les informations dans le cache du dictionnaire pour résoudre les noms d'objet et valider l'accès.
  - La mise en mémoire cache des informations du dictionnaire de données réduit le temps de réponse aux interrogations et aux instructions LMD.
  - La taille du cache dépend du dimensionnement de la zone de mémoire partagée.

# L'instance

- Redo Log Buffer (Tampon de journalisation)
  - Il enregistre toutes les modifications apportées aux blocs de données de la base.
  - Sa principale fonction est la récupération de données.
  - Les modifications enregistrées constituent des entrées de journalisation.
  - Les entrées de journalisation contiennent des informations permettant de reconstruire des modifications.
  - La taille du tampon est définie par le paramètre LOG\_BUFFER.



# L'instance

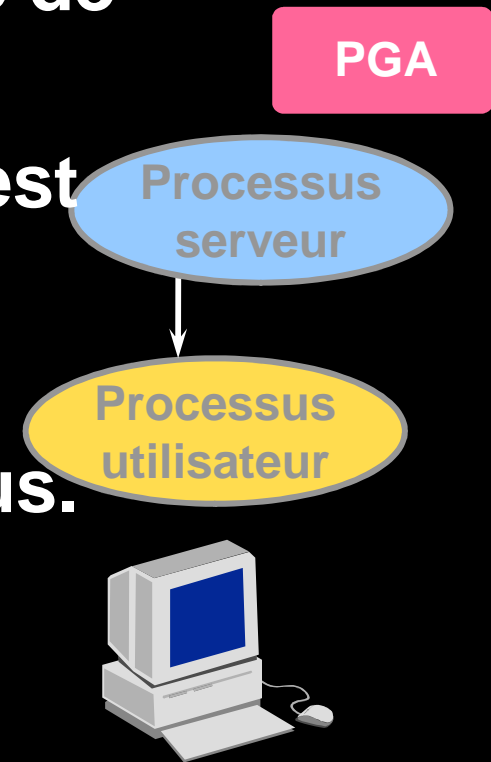
- Zone de mémoire LARGE POOL
  - Zone facultative de la mémoire SGA
  - Elle réduit la charge de la zone de mémoire partagée.
    - la mémoire allouée par session (UGA) au serveur partagé
    - les processus serveur d'E/S
    - les opérations de sauvegarde et de restauration ou RMAN
    - les mémoires tampon des messages d'exécution en parallèle
      - `PARALLEL_AUTOMATIC_TUNING = TRUE`
  - Elle n'utilise pas de liste LRU.
  - Sa taille est définie par le paramètre `LARGE_POOL_SIZE`.

- Zone de mémoire Java
  - La zone de mémoire Java répond aux besoins d'analyse des commandes Java.
  - Elle est nécessaire si Java est installé et utilisé.
  - Sa taille est définie par le paramètre **JAVA\_POOL\_SIZE**.
- Zone Streams Pool
  - *Est dimensionnée par le paramètre STREAMS\_POOL\_SIZE (0 par défaut)*

# L'instance

## Mémoire PGA

- Mémoire réservée à chaque processus utilisateur qui se connecte à une base de données Oracle.
- Elle est allouée lorsqu'un processus est créé.
- Elle est libérée à la fin du processus.
- Elle n'est utilisée que par un processus.
- Les tables v\$sesstat, v\$statname, permettent de déterminer la taille de la PGA pour une session



# L'instance

## ■ Mémoire PGA

- **La taille maximum de la PGA est influencée en plus par les paramètres d'initialisations suivants :**
  - sort\_area\_size
  - hash\_area\_size
  - bitmap\_merge\_area\_size and create\_bitmap\_area\_size
- **D'autres paramètres influencent aussi la taille de la PGA d'une session**
  - OPEN\_LINKS : nombre de databases link ouverts
  - DB\_FILES : nombre de fichiers de données pouvant être ouverts
- **En mode serveur dédié il est difficile de gérer l'allocation des paramètres \*\_area\_size. Depuis la 9i le DBA peut fixer sa PGA maximale grâce au paramètre :**
  - PGA\_AGGREGATE\_TARGET

# L'instance

## ■ La Zone de tri

- **Une zone de tri** est associée à un Serveur (dédié ou non) pour traiter des ordres nécessitant des tris (Group by, Order by, Join, ...)
- **la taille de la zone de tri** est déterminée par le paramètre **SORT\_AREA\_SIZE** (en bytes)
  - Par défaut cette taille est de **65000 bytes**
  - Si cette zone est pleine un **Segment temporaire** est généré
- **SORT\_AREA\_RETAINED\_SIZE** (exprimée en byte, 0 min, Sort\_area\_size par défaut et max): espace à ne pas libérer en cas d'écriture dans le segment temporaire **tuning de la zone de tri** ;  
table v\$sysstat

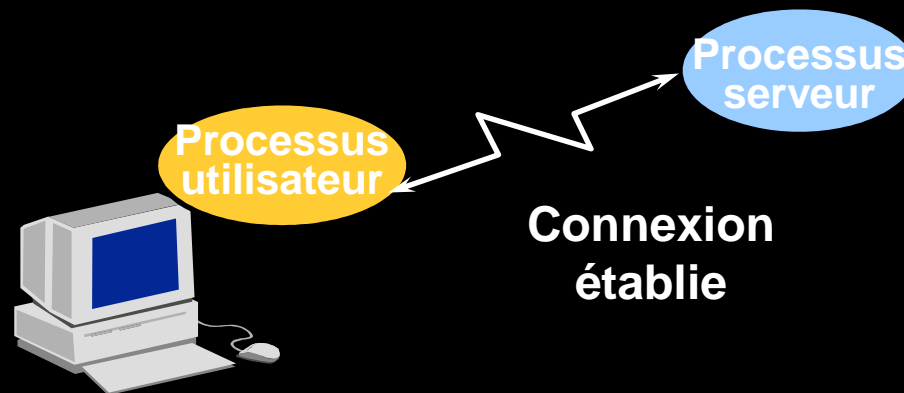
```
Sql> SELECT name, value FROM v$sysstat  
WHERE name in ('sorts (memory)',  
               'sorts (disk)');
```



## L'instance

### ■ Processus utilisateur

- Programme qui demande une interaction avec le serveur Oracle.
- Ce processus doit d'abord établir une connexion.
- Il n'entre pas directement en interaction avec le serveur Oracle.

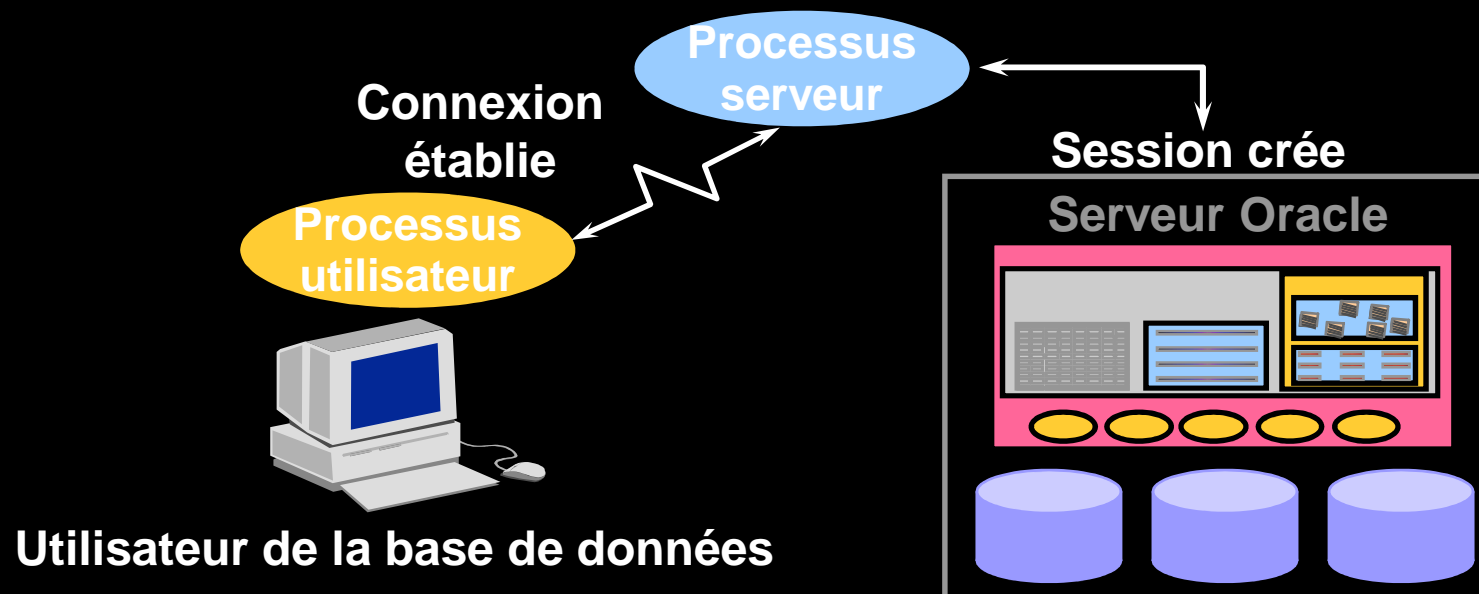


Utilisateur de la base de données

# L'instance

## ■ Processus serveur

- Programme qui entre directement en interaction avec le serveur Oracle.
- Il répond aux appels générés et renvoie les résultats.
- Il peut s'agir d'un serveur dédié ou d'un serveur partagé.



## Processus d'arrière-plan

- Gèrent et appliquent les relations entre les structures physiques et les structures mémoire.

- Processus d'arrière-plan obligatoires

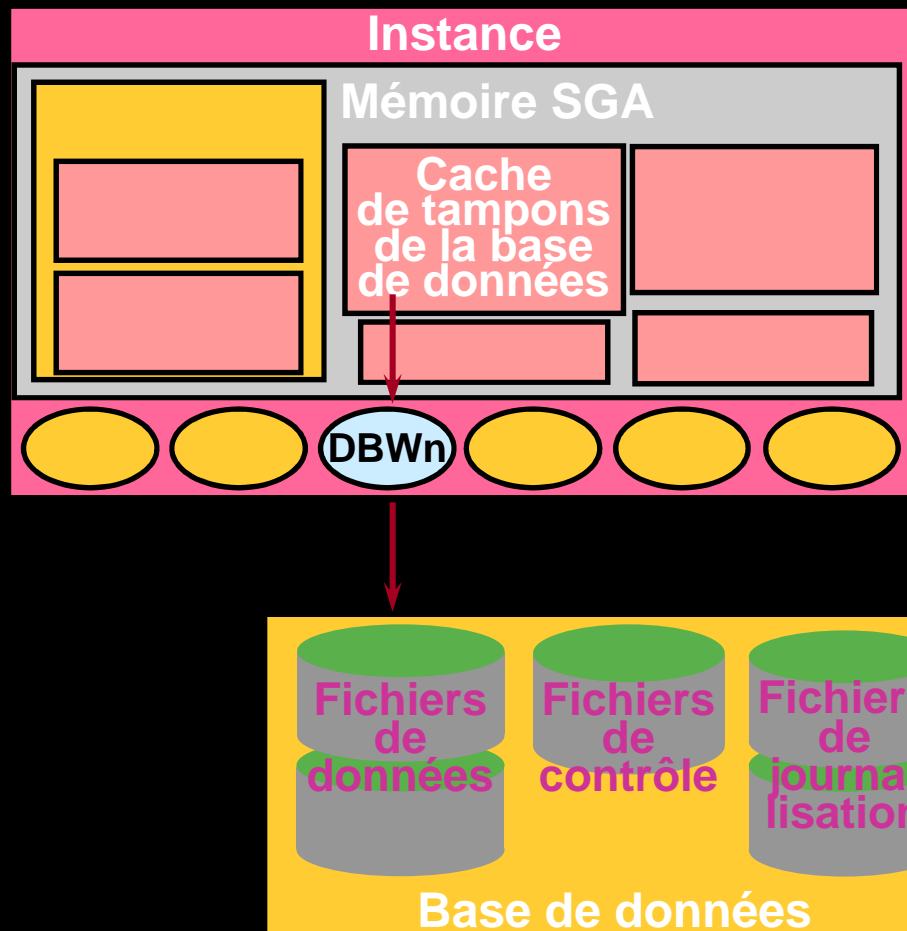
- DBWn
- CKPT
- LGWR
- \*PMON
- \*SMON

- Processus d'arrière-plan facultatifs

- |        |      |      |
|--------|------|------|
| • ARCn | LMDn | RECO |
| • CJQ0 | LMON | Snnn |
| • Dnnn | Pnnn |      |
| • LCKn | QMNn |      |

## Processus d'arrière-plan

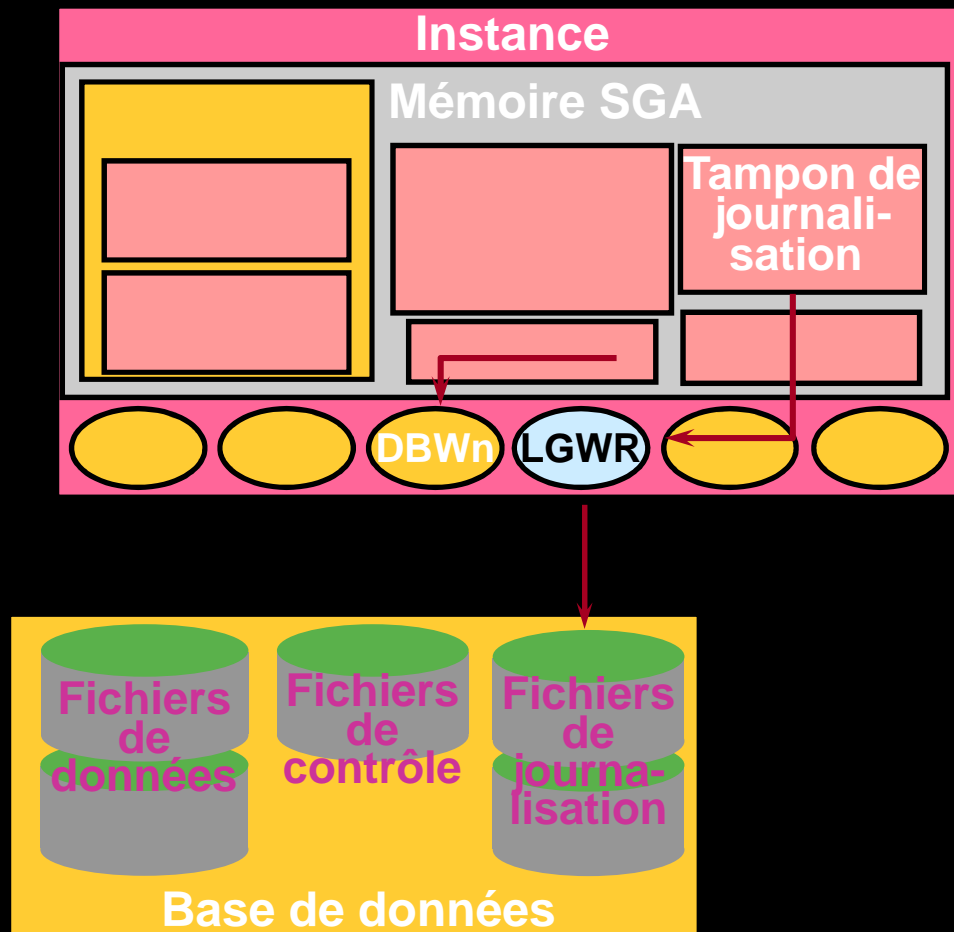
- Processus database writer (DBWn) **DBWn** écrit dans les cas suivants :



- *point de reprise*
- *aucune mémoire tampon disponible*
- *tablespace hors ligne*
- *tablespace en lecture seule*
- *DROP ou TRUNCATE sur une table*
- *BEGIN BACKUP sur un tablespace*

# Processus d'arrière-plan

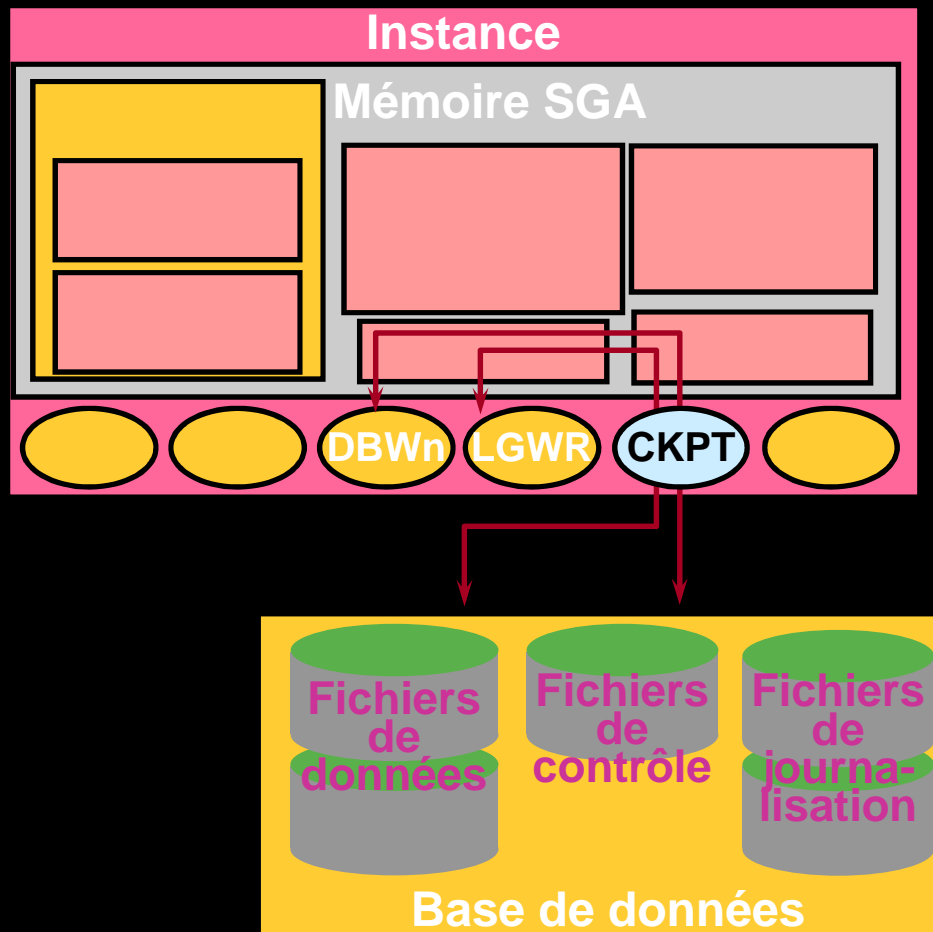
## ■ Processus LGWR (Log Writer)



- **LGWR** écrit dans les cas suivants :
  - validation
  - un tiers du cache est occupé
  - la journalisation atteint 1 Mo
  - avant que le processus DBWn ne procède à une opération d'écriture

## Processus d'arrière-plan

- Processus **CKPT** (Checkpoint)

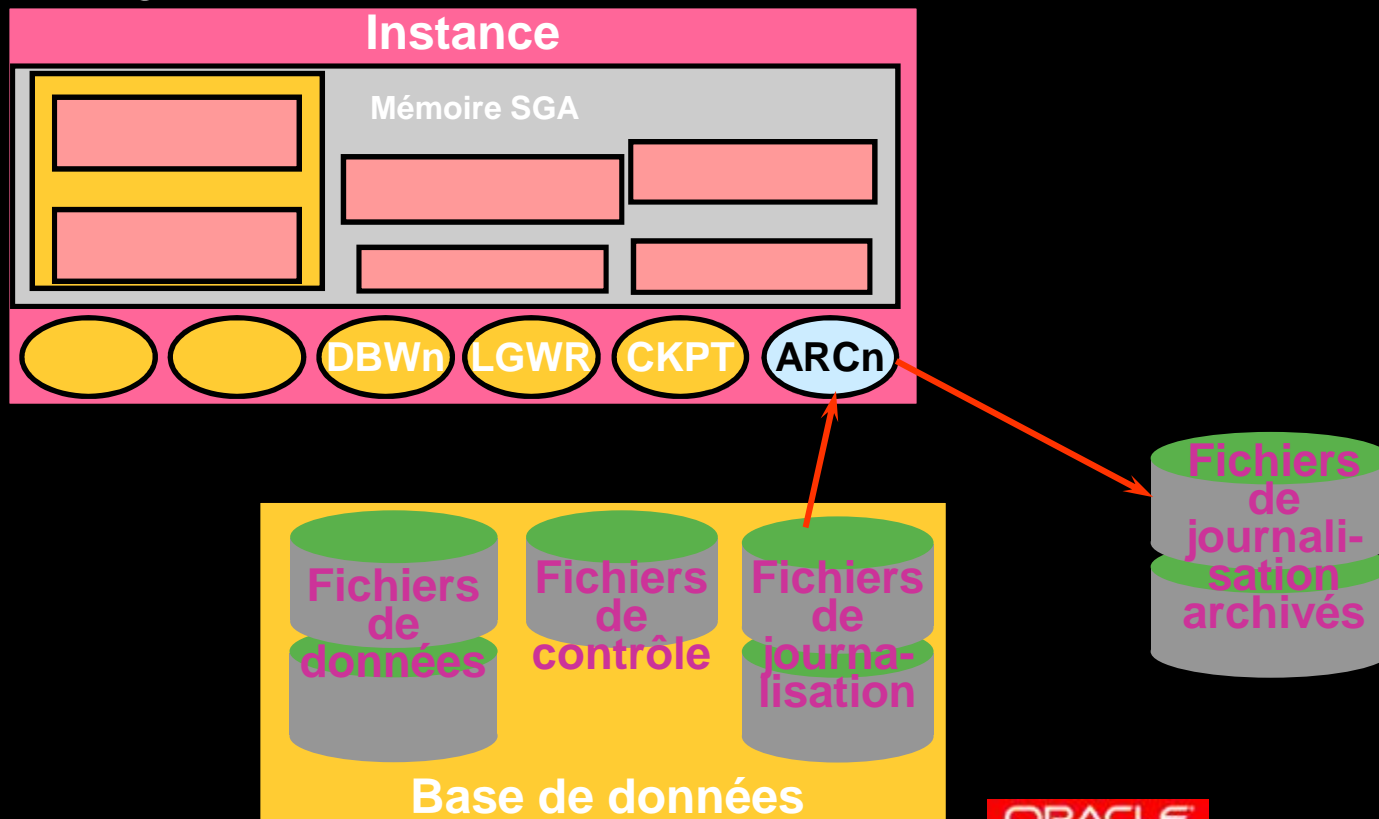


- Ce processus est chargé :

- de signaler DBWn aux points de reprise,
- de mettre à jour les fichiers de contrôle avec les informations sur le point de reprise.

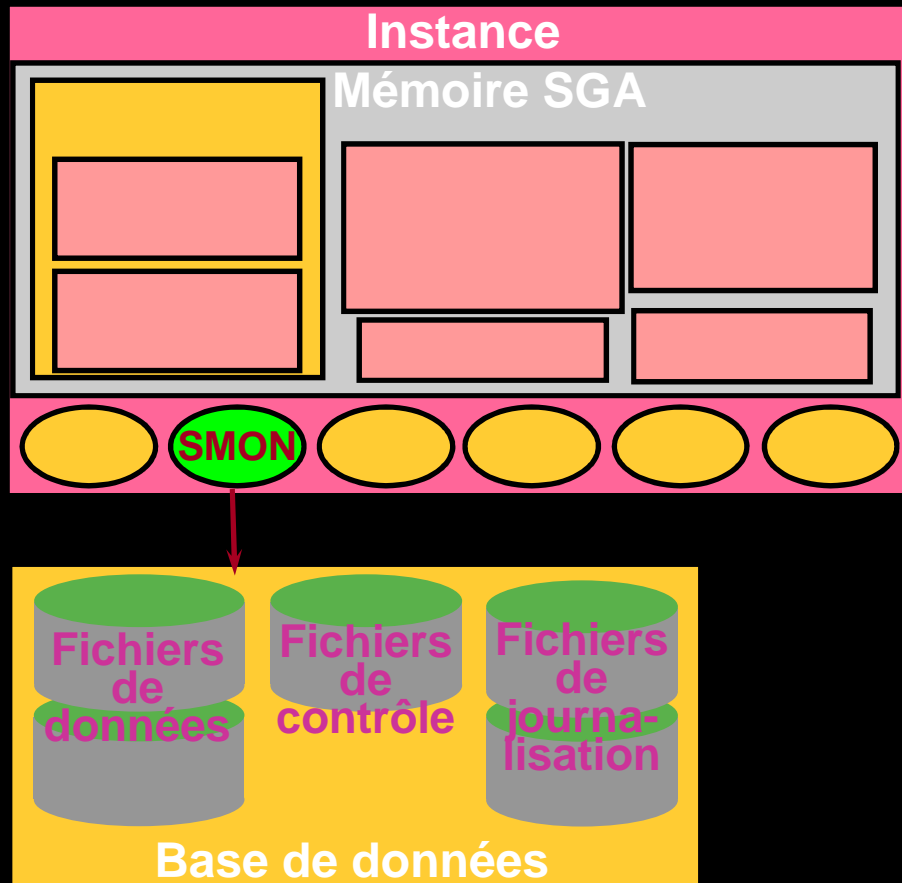
## Processus d'arrière-plan

- Processus **ARCn** (processus d'archivage)
  - Processus d'arrière-plan facultatif
  - En mode **ARCHIVELOG**, il archive automatiquement les fichiers de journalisation en ligne
  - Il enregistre toutes les modifications apportées à la base de données



# Processus d'arrière-plan

## ■ Processus **SMON** (System Monitor)



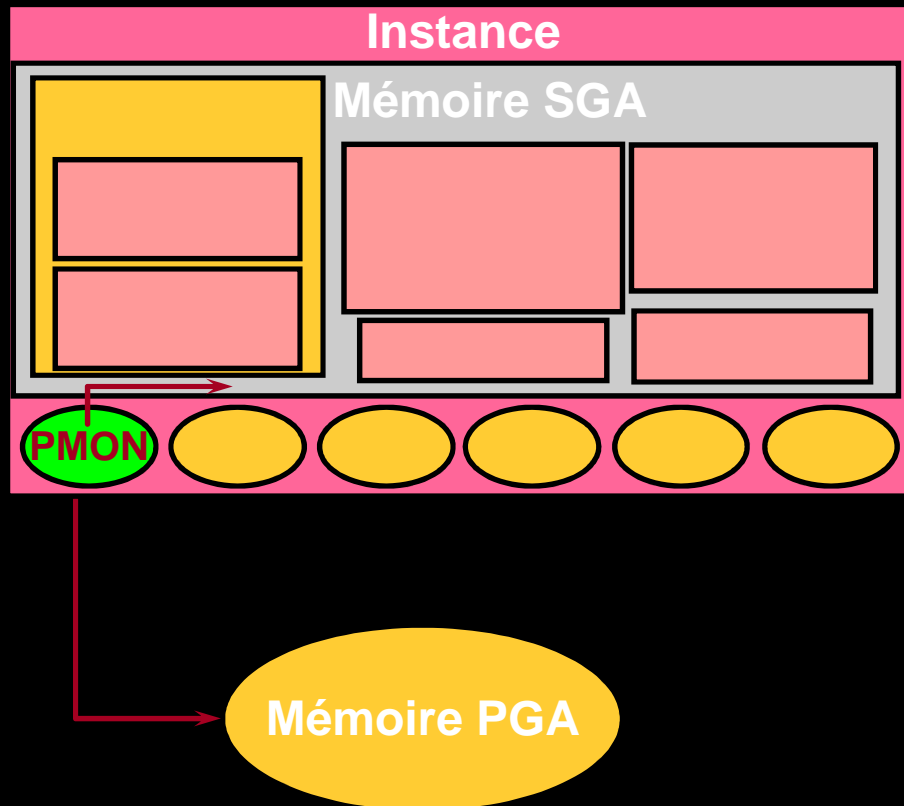
## ■ Responsabilités :

- Récupération de l'instance :
  - réimplémente des modifications dans les fichiers de journalisation,
  - ouvre la base de données pour permettre l'accès aux utilisateurs,
  - annule les transactions non validées.
- Fusion de l'espace libre

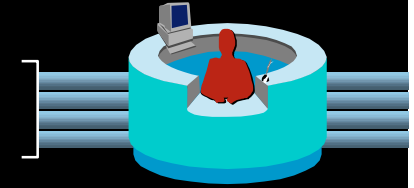


## Processus d'arrière-plan

- Processus **PMON** (Process Monitor)



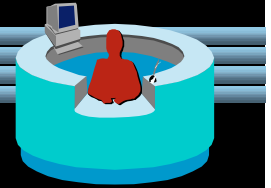
- Suite à l'échec de processus, PMON exécute des opérations de nettoyage :
  - annule la transaction
  - libère des verrous
  - libère d'autres ressources



### ■ Création d'une base

- Variables d'environnement à positionner
  - ORACLE\_SID=nom\_de\_instance
  - ORACLE\_HOME=home\_oracle
  - ORACLE\_BASE=home\_des\_bases\_Oracle
- Outils de lancement des commandes
  - Sqlplus
  - Le *Database Configuration Assistant*. Outil graphique permettant de créer et modifier les bases
  - implicite lors de l'installation du serveur Oracle

# Administration de B.D : création (suite)



## ■ Création d'une base

### ■ Etapes à suivre pour créer une base manuellement

#### 1. Définir l'arborescence de la base

**ora9data->dbtest->admin, tssys, tsusers, tstemp, tsrbs, ...**

#### 2. Définir les scripts de création de la base (crDBTEST.sql)

#### 3. Définir le fichier d'initialisation (initDBTEST.ora)

**Si Win : Créer le service Win pour la base.**

**C:\>oradim -new -sid dbtest -intpwd manager -startmode auto -pfile  
c:\ora9data\dbtest\admin\initDBTEST.ora**

#### 4. Lancer Sqlplus

**c:\> sqlplus**

**username:sys as sysdba password:manager**

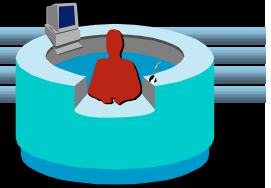
#### 5. Exécuter les commandes contenues dans crDBTEST.sql

- démarrer une instance (nomount)
- exécuter CREATE DATABASE ...
- exécuter catalog.sql (pour les vues du dictionnaires)
- ajouter des tablespaces supplémentaires pour une meilleure organisation ttemp, tsrbs, tsutil
- ajouter des rbs supplémentaires si mode de gestion d'annulations manuelles
- exécuter catproc.sql (pour l'option procédurale)
- exécuter les scripts supplémentaires suivants : catdbsyn.sql
- fixer le tablespace temporaire des users SYS et SYSTEM vers le tablespace temporaire

**Voir Annexes** pour plus de détails sur initSid.ora, spfile et crsid.sql

## Exemple de Fichier des Paramètres

```
■ # Initialization Parameter File: initU15.ora
■ db_name                = U15
■ control_files           = (/DISK1/control01.con,
■                          /DISK2/control02.con)
■ db_block_size           = 8192
■ db_block_buffers        = 2000
■ shared_pool_size        = 30000000
■ log_buffer              = 64K
■ processes               = 50
■ db_files                 = 100
■ log_files                = 10
■ max_dump_file_size      = 10240
■ background_dump_dest    = (/home/disk3/user15/BDUMP)
■ user_dump_dest           = (/home/disk3/user15/UDUMP)
```



### ■ Création d'une base

#### Code SQL

```
CREATE DATABASE [nombase]  
  [CONTROLFILE REUSE]  
  [LOGFILE {[GROUP entier] logFileSpec, ...}]  
  [MAXLOGFILES entier]  
  [MAXLOGMEMBERS entier]  
  [MAXLOGHISTORY entier]  
  [MAXDATAFILES entier]  
  [MAXINSTANCES entier]  
  [{NOARCHIVELOG | ARCHIVELOG}]  
  [CHARACTER SET nomCaracterSet]  
  [NATIONAL CHARACTER SET nomCaracterSet]  
  [DATAFILE {dataFileSpec [ClauseAutoExtend], ...}]  
  [default_temp_tablespace]  
  [undo_tablespace_clause]  
  [SET STANDBY DATABASE {PROTECTED UNPROTECTED}]  
  [set_time_zone_clause]
```

# Le fichier de paramètres

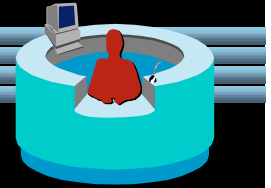
- Consultable via la vue V\$PARAMETER

```
SELECT name, value FROM V$PARAMETER [show  
parameter...]
```

- Il existe deux types de fichiers de paramètres:

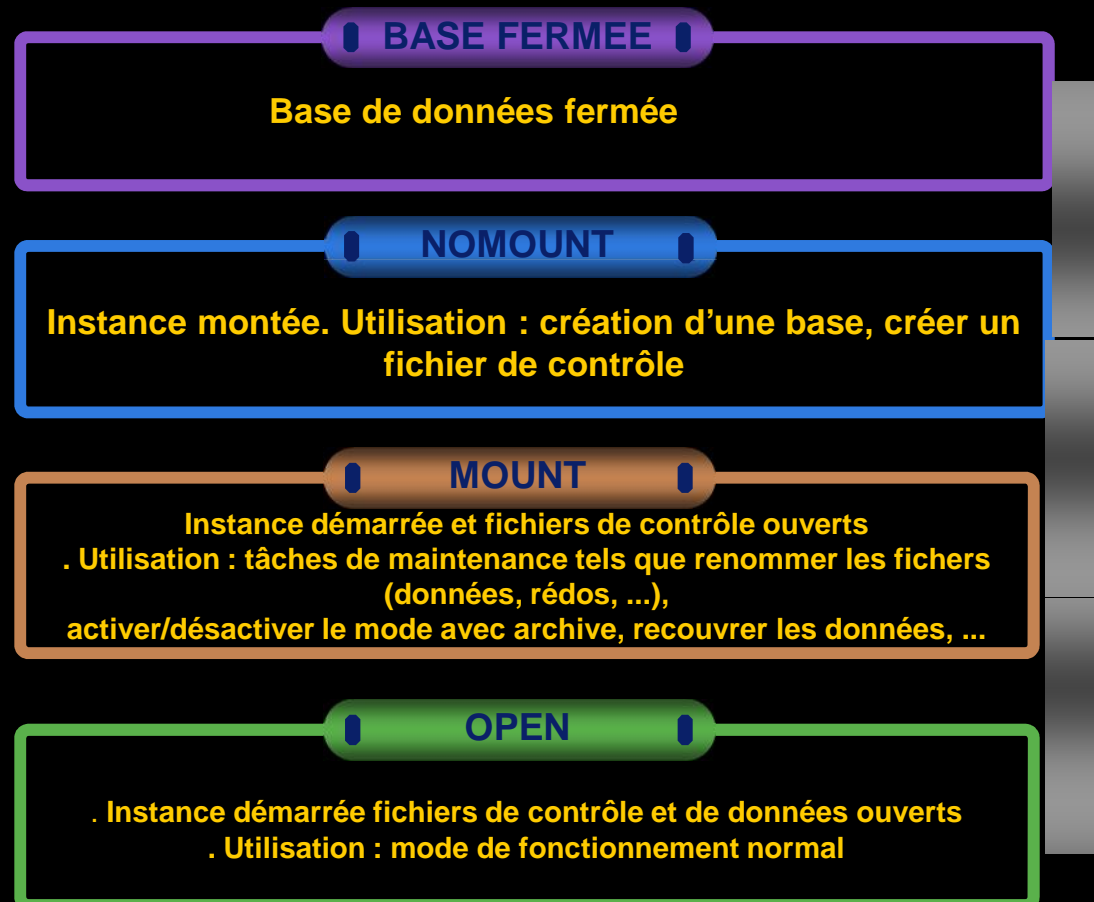
<b>PFILE</b> (parameter file)	<b>SPFILE</b> (server parameter file)
Fichier texte	Fichier binaire
Modifiable via un éditeur texte	Modifiable via SQL
Disponible sur la machine de démarrage	Centralisé (sur le serveur uniquement)
Nommé init%.ora	Nommé spfile%.ora
Modification à froid	Modification à chaud

# Administration de base : états

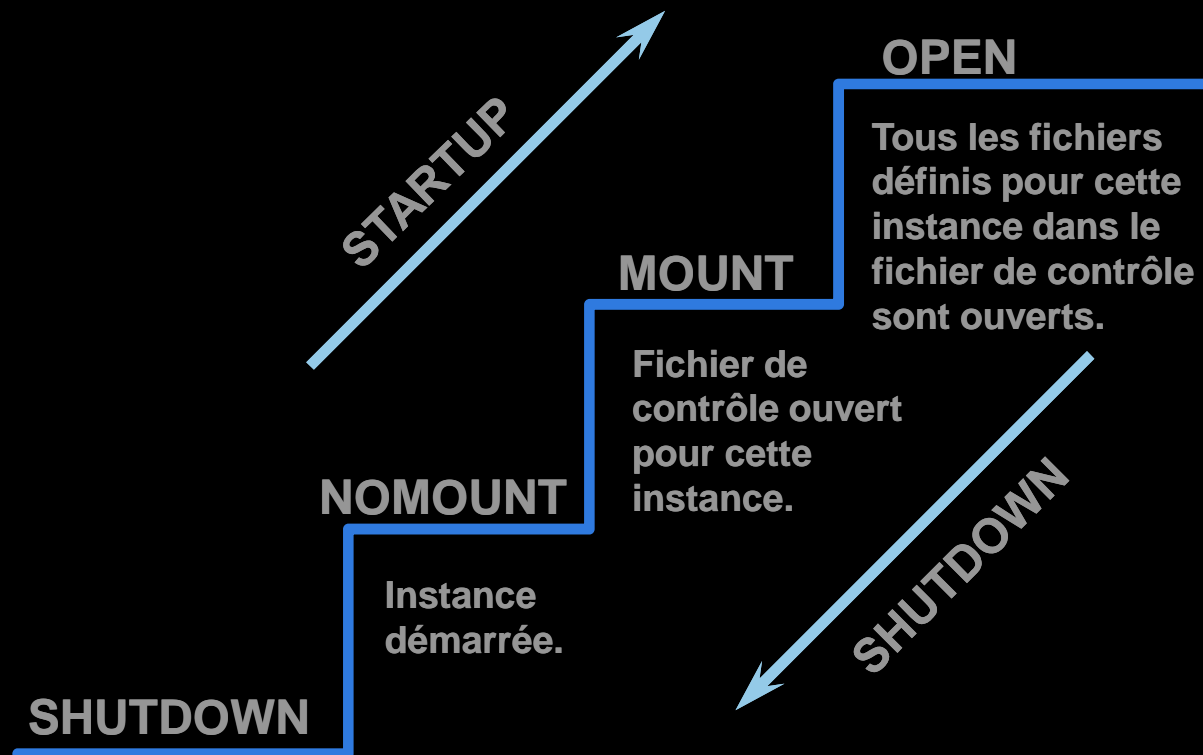


## ■ États d'une base

- Une Base de données comporte les états suivants:



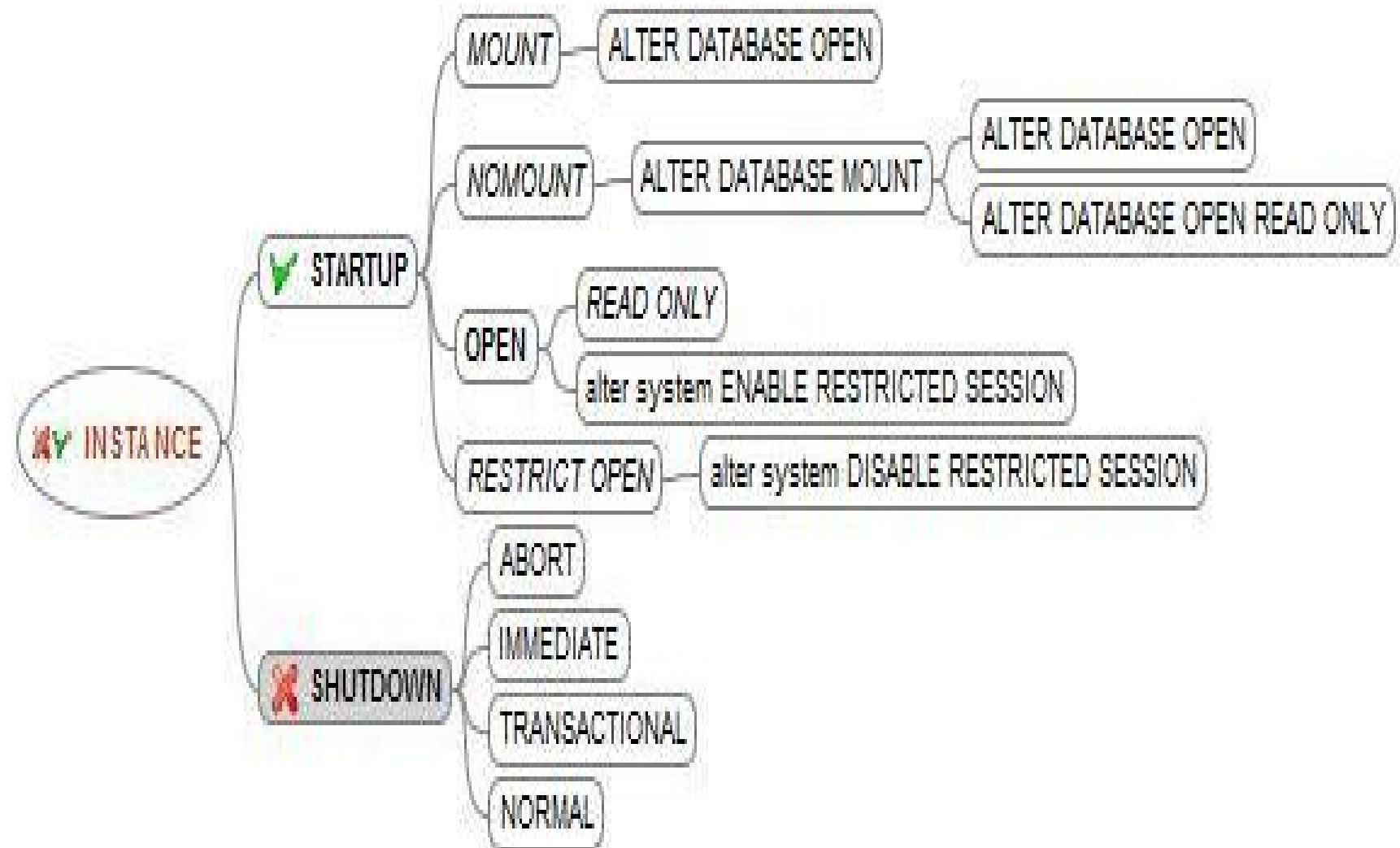
# Démarrage et Arrêt par Etapes

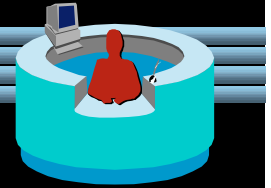


```
STARTUP PFILE=/DISK1/initMABASE.ora
```



## Démarrage et Arrêt par Etapes (2)





## ■ Démarrage d'une base

### ■ Procédure Générale

- Positionner les variables d'environnement ORACLE\_SID et ORACLE\_HOME
- démarrage de la base possible par paliers (***startup nomount, startup mount, et startup open***)
- posséder les privilèges appropriés (SYSDBA, SYSOPER, OSDBA ou OSOPER, ...)
- indiquer si nécessaire le fichier des paramètres
- démarrer la base sous *sqlplus* (Unix et NT) ou via *les services*(NT) ou *Oradim* (NT), à travers Entreprise
- Manager: *database administration assistant* (NT ou Unix)

### ■ Syntaxe de la commande

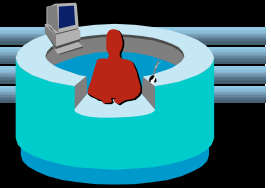
- ***STARTUP [FORCE]***  
***[RESTRICT] [PFILE=fich\_param]***  
***[OPEN | MOUNT | NOMOUNT]***
- **ALTER DATABASE [nom\_base] MOUNT | OPEN**



# Gestion des utilisateurs



# Tâches d'administration de base



## ■ Démarrage d'une base (Exemple)

### ■ Lancer SQLPLUS

\$sqlplus

username : sys as sysdba password : manager

- #démarrer et ouvrir les fichiers de la base

SQL >startup ;

*idem*

SQL>startup OPEN

pfile= /user/oracle/v9/dbs/initCOURS.ora

- #démarrer la base par paliers

\$ SQLPLUS

username : sys as sysdba password : manager

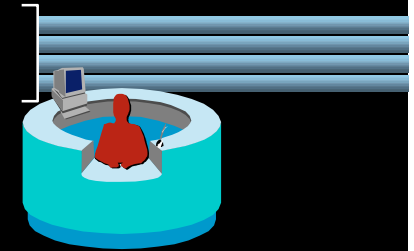
# Instance démarrée, fichiers de contrôles ouverts

SQL >startup mount ;

#En plus les fichiers de données sont ouverts

SQL > alter database open ;

# Administration de BD : SHUTDOWN



## ■ Arrêt d'une base

### ■ Procédure Générale

- Posséder les privilèges OS (osdba, osoper)
- positionner les variables d'environnement (ORACLE\_SID et ORACLE\_HOME)
- Lancer l'outil SQLPLUS

### ■ Syntaxe de la commande

- **SHUTDOWN [NORMAL | IMMEDIATE | ABORT | TRANSACTIONAL]**

#### – NORMAL

**attend la déconnexion de l'ensemble des utilisateurs**

#### – IMMEDIATE

**Invalide les transactions en cours et déconnecte les users**

#### – ABORT

**Arrêt violent de l'instance (recouvrement utile : SMON)**

#### – TRANSACTIONAL

**Contrairement à IMMEDIATE, laisse finir les transactions en cours**

## Mode d'arrêt d'une instance de BD

Mode d'arrêt	A	I	T	N
Permet de nouvelles connexions	Non	Non	Non	Non
Attend la fin des sessions en cours	Non	Non	Non	Oui
Attend la fin des transactions en cours	Non	Non	Oui	Oui
Applique un point de reprise et ferme les fichiers	Non	Oui	Oui	Oui

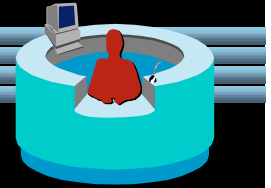
**Mode d'arrêt :** A = ABORT

T = TRANSACTIONAL

I = IMMEDIATE

N = NORMAL

# Administration de BD : SHUTDOWN (suite)



## ■ Arrêt d'une base (Exemple)

### ■ Lancer SQL

**username : sys as sysdba**

**# arrêt normal avec attente de déconnexion**

**SQL>shutdown**

**ou**

**# Arrêt immédiat : Annulation des transactions en cours et déconnexion .**

**# Les utilisateurs ont le message suivant :**

**# ORA-03113 : end-of-file on communication channel**

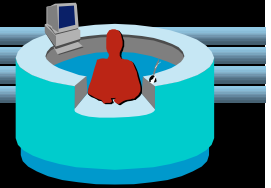
- **SQL>shutdown immediate ;**

**ou**

**# Arrêt brutal (pas d'attente de déconnexion, pas d'invalidation de transactions en cours)**

- **SQL >shutdown abort;**

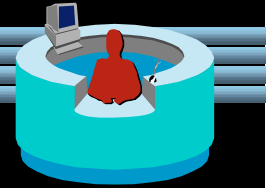
## Administration de BD



- **Suspension et réactivation de la base**
  - Afin de permet une sauvegarde base en ligne sans activité de mise à jour dans les fichiers de données il est possible maintenant de suspendre l'activité de mise à jour
  - Commande à exécuter pour suspendre la base
    - **SQL>ALTER SYSTEM SUSPEND**
  - Commande à exécuter pour revenir en mode normal
    - **SQL>ALTER SYSTEM RESUME**
  - **NOTE : les règles de sauvegarde *base ouverte* doivent être respectées (voir le chapitre Sauvergarde et Restauration)**

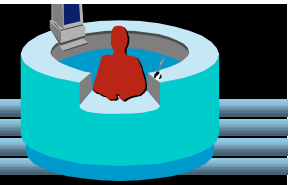


## Administration de BD : DD



- **Le dictionnaire de données d'Oracle** (voir ateliers )
  - un dictionnaire par base de données
  - Ensemble de tables, vues et synonymes permettant la gestion des objets d'une base (propriété de l'utilisateur SYS)
  - Il est accessible via des ordres SQL
  - Il est modifié indirectement via des ordres dits DDL
  - Il ne doit être modifié directement
  - Contient des informations persistantes (objets créés tels que les tables, les index, les clusters, ...) et des informations dynamiques (tels les sessions ouvertes, les E/S effectuées, ...)

## Administration de BD : DD



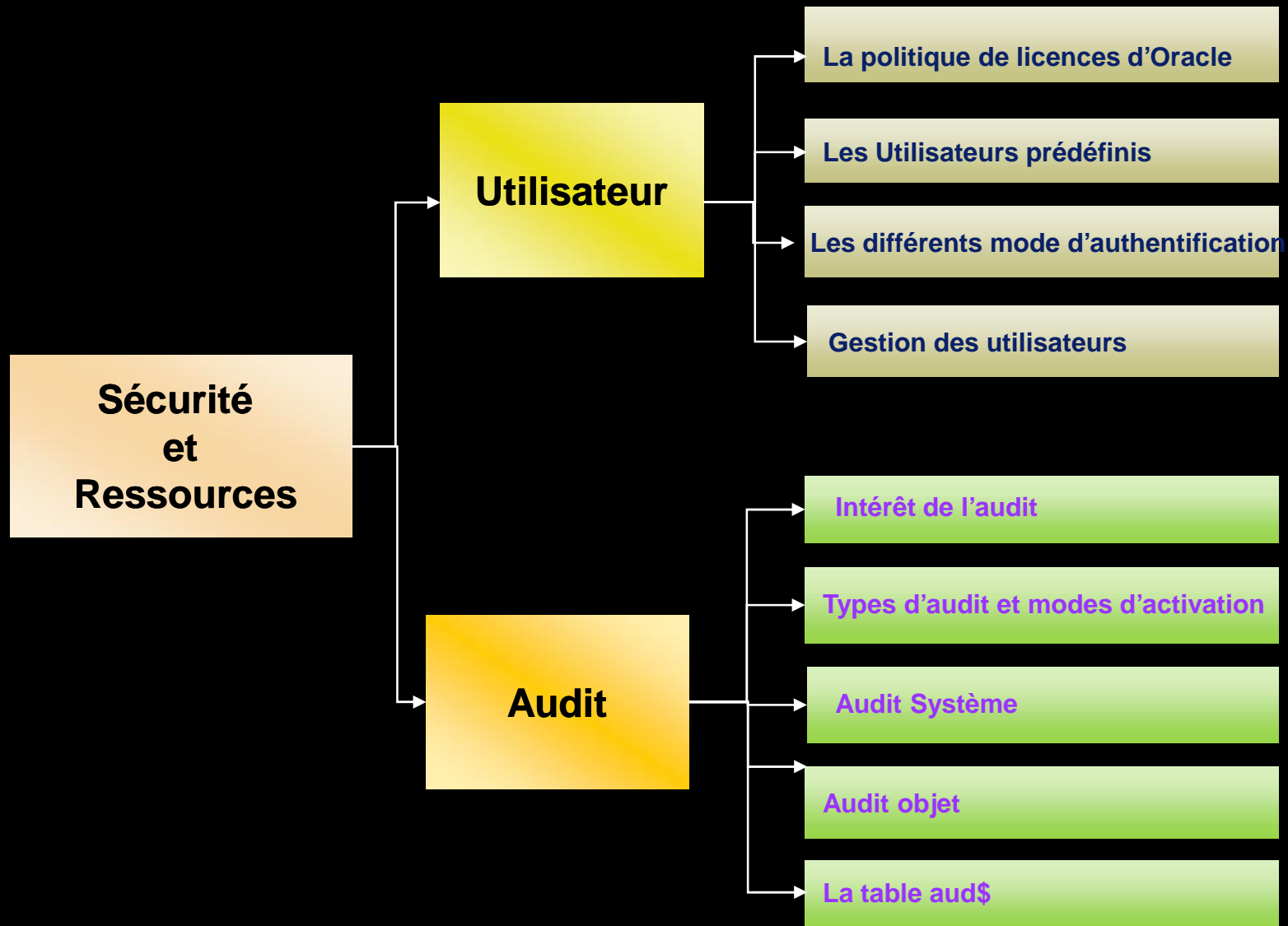
### ■ Le dictionnaire de données d'Oracle (Contenu du DD)

- des **tables de base** du dictionnaire stockées en cluster et finissant par \$. Exemple **tab\$, ind\$, obj\$, seg\$, ...**
- les **vues de performances**

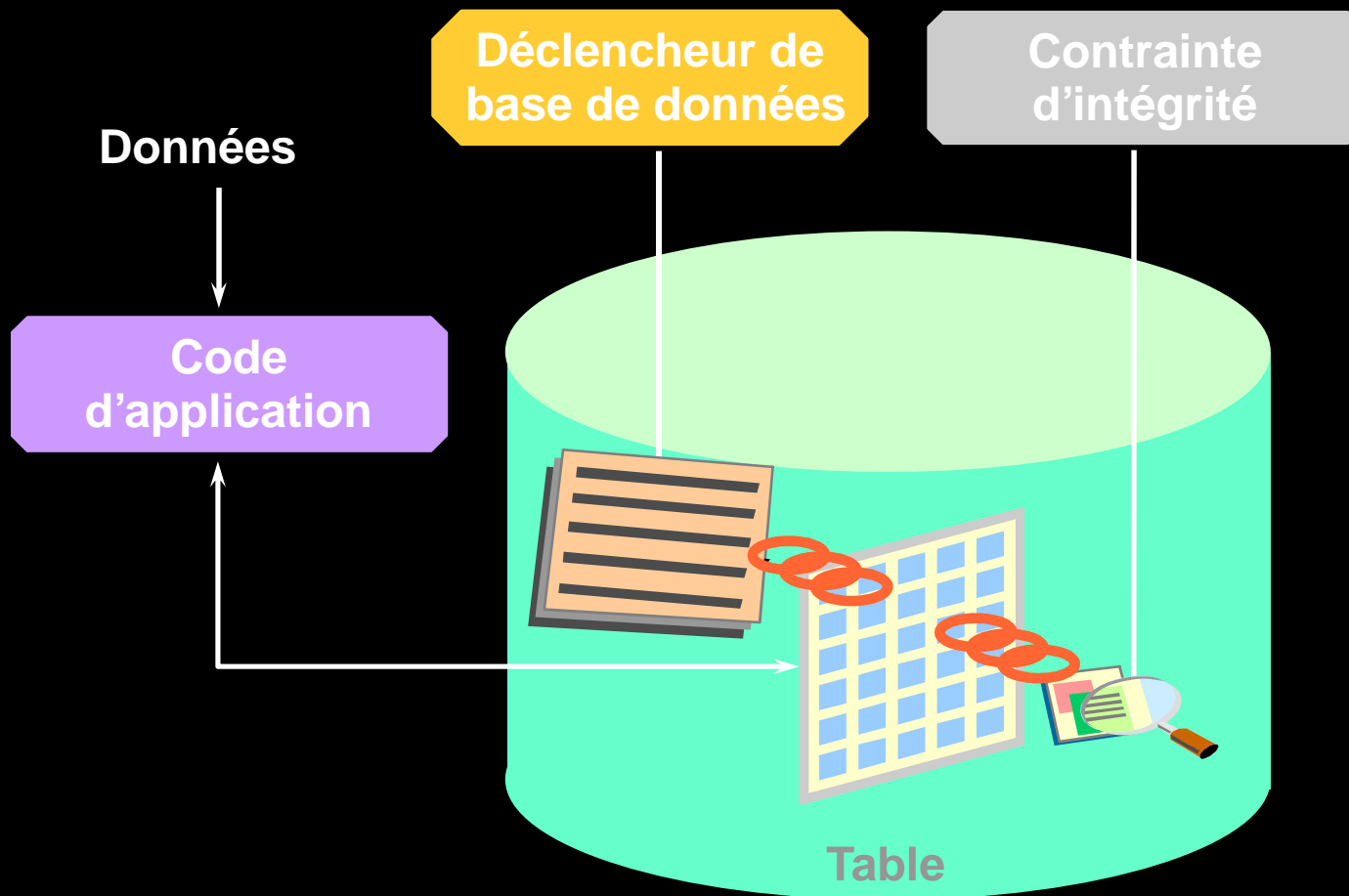
exemples : **v\_\$process, v\_\$session, v\_\$syststat, v\_\$sesstat, ...**

- des **vues** sur le dictionnaire de base commençant par :
  - **user\_\*** : information sur tous les objets dont l'utilisateur connecté est propriétaire
  - **all\_\*** : informations sur les tous les objets accessibles par l'utilisateur connecté
  - **dba\_\*** : information sur tous les objets de la base. Il faut avoir le privilège **SELECT ANY TABLE** pour y accéder
- des **synonymes** sur les vues pour simplifier. Exemple:
  - **v\$process, v\$session, v\$systat, v\$sesstat, ...**

# Gestion de la sécurité et des ressources



# Intégrité des données



### ■ Généralités

#### ■ Rôle de l'Administrateur de sécurité et des ressources

- Définir une politique de sécurité
- Faire les choix du type de sécurité : au niveau système, au niveau Oracle, au niveau Global (LDAP)
- Gérer les utilisateurs
- Gérer les ressources (profiles)
- Assurer l'affectation et le retrait des droits
- Affiner la politique de sécurité par l'utilisation des rôles
- Effectuer les audits

# Gestion de la sécurité et des ressources



## Objectifs

- créer des rôles
- accorder des privilèges à des rôles
- accorder des rôles à des utilisateurs ou à des rôles
- établir des rôles par défaut

O. EL BEQQALI

62

### ■ Les Privilèges

#### ■ Introduction

- Un privilège donne le droit d'exécuter certaines commandes SQL ou le droit d'accéder à certaines ressources
- Oracle possède deux types de privilèges : les *privilèges systèmes* et les *privilèges objets*.
- Un privilège peut être affecté (retiré) à un Utilisateur, un Rôle ou tous les utilisateurs (PUBLIC)

# Gestion de la sécurité et des ressources

## ■ Les Privilèges

### ■ Les privilèges Systèmes

- Oracle possède près de **127 privilèges Systèmes** (la V6 en avait 3 : *Connect, resource, dba*)
- Les privilèges donnent le droit de réaliser des opérations systèmes
- Ces privilèges sont classés par catégories d'objets

– ANALYZE	AUDIT
– CLUSTER	CONTEXT
– DATABASE	DATABASE LINK
– DIMENSION	INDEX
– INDEXTYPE	LIBRARY
– MATERIALIZED VIEW	MISCELLANEOUS
– OPERATOR	OUTLINE
– PRIVILEGE	PROCEDURE
– PROFILE Public	Database Link
– PUBLIC SYNONYM	ROLE
– ROLLBACK SEGMENT	SESSION
– SEQUENCE	SNAPSHOT
– SYNONYM	SYSTEM
– TABLE	TABLESPACE
– TRANSACTION	TRIGGER
– TYPE	USER
– VIEW	

**NOTE** : Voir le site d'oracle pour obtenir la liste complète des privilèges



# Gestion de la sécurité et des ressources

## ■ Les privilèges

### ■ Exemple de privilèges systèmes de la catégorie TABLE:

- CREATE TABLE CREATE ANY TABLE
- ALTER ANY TABLE BACKUP ANY TABLE
- DROP ANY TABLE LOCK ANY TABLE
- LOCK ANY TABLE SELECT ANY TABLE
- INSERT ANY TABLE UPDATE ANY TABLE
- DELETE ANY TABLE COMMENT ANY TABLE
- UNDER ANY TABLE

### ■ Affectation d'un privilège Système

**GRANT { system\_priv | role } TO { user | role | PUBLIC }  
[ WITH ADMIN OPTION ]**

**System\_priv** : nom d'un privilège système

**role** : Nom d'un rôle

**user, role ou PUBLIC** : droit affecté à un utilisateur, un rôle ou public

**With Admin Option** : le rôle pourra être redistribué par celui qui le reçoit

### ■ Les privilèges systèmes (suite)

#### ■ Affectation des privilèges systèmes (suite)

- – L'affectation d'un privilège avec l'option "**WITH ADMIN OPTION**" suit les règles suivantes :
  - Celui qui reçoit le droit peut le redistribuer
  - **Son retrait** à un utilisateur qui lui-même l'a affecté à un autre **ne peut se faire en cascade**
  - ne peut être affecté à un ROLE

#### ■ Exemple

- **GRANT ALTER TABLESPACE TO scott ;**
- **GRANT CREATE USER,**  
**CREATE SESSION TO scott WITH ADMIN OPTION ;**
- **GRANT ALTER ANY TABLE TO PUBLIC ;**

### ■ Les Privilèges Systèmes (suite)

#### ■ Révocation d'un privilège Système

- Syntaxe

```
REVOKE { system_priv | role }
```

```
FROM { user | role | PUBLIC }
```

- Exemple :

```
REVOKE ALTER ANY TABLE FROM PUBLIC ;
```

```
REVOKE CREATE SESSION FROM SCOTT ;
```

#### ■ Les vues du dictionnaire

```
SELECT * FROM DBA_SYS_PRIVS ORDER BY grantee, privilege ;
```

<u>GRANTEE</u>	<u>PRIVILEGE</u>	<u>ADM</u>
CONNECT	ALTER SESSION	NO
CONNECT	CREATE CLUSTER	NO

- **NOTE** : Supposant qu'un utilisateur U1 attribut un privilège P1 "WITH ADMIN OPTION" à un utilisateur U2 et que U2 l'attribut à son tour à U3. La révocation de P1 à U2 n'entraîne pas la révocation de P1 à U3.

# Gestion de la sécurité et des ressources

## ■ Les privilèges Objets

- Ces privilèges contrôlent l'accès aux objets des tables, vues, séquences, procédures, fonctions et packages, vue matérialisée (VM) ....
- Classification selon les types d'objets

Privilèges objets	Libellé	Objets concernés
ALTER	droit de modifier	<i>table, séquence</i>
DELETE	droit de supprimer	<i>table, vue, VM</i>
EXECUTE	droit d'exécuter	<i>procédure, fonction, package, type user, opérateur, indextype, library</i>
INDEX	droit de créer un index ( <i>ne peut être affecté à un rôle</i> )	<i>table</i>
INSERT	droit d'insérer	<i>table, vue, VM</i>
ON COMMIT REFRESH	droit de créer une vue matérialisée ON COMMIT REFRESH sur une table	<i>table</i>
QUERY REWRITE	droit de créer une vue matérialisée sur une table	<i>table</i>
REFERENCES	droit de référencer une table lors d'un alter ou create table ( <i>ne peut être affecté à un rôle</i> )	<i>table</i>
READ	droit de lire dans une directory	<i>directory</i>
SELECT	droit de consulter	<i>table, vue, snapshot, sequence</i>
UPDATE	droit de mise à jour	<i>table ou vue</i>
UNDER	droit de créer des sous vue	<i>vue, type user</i>
WRITE	droit d'écrire dans une directory	<i>directory</i>

# Gestion de la sécurité et des ressources

## ■ Les privilèges Objets

### ■ Affectation de privilèges objets

```
GRANT { object_priv | ALL [ PRIVILEGES ] } [( column [,column ] ... ) ]  
[, { object_priv | ALL [ PRIVILEGES ] } [ ( column [,column] ... ) ] ]  
ON [ schema.] object  
TO { user | role | PUBLIC } [ WITH GRANT OPTION ]
```

#### • Notes :

- *ALL* : n'est pas un privilège mais signifie "tous les privilèges sur un objet"
- *object\_priv*: Nom du privilège
- *column* : Nom d'une colonne si *object\_priv*= *insert*, *update* ou *references*
- *schema.objet* : Nom de l'objet concerné
- *With Grant Option*: L'utilisateur qui reçoit le privilège peut le réaffecter.

#### • Exemple

```
sql> GRANT INSERT (ename, job) ON emp TO scott with grant option ;
```

```
sql> GRANT UPDATE (SAL), DELETE ON emp TO scott ;
```

```
sql> GRANT REFERENCES, UPDATE ON bonus TO BRAHIM ;
```

## ■ Les privilèges Objets

### ■ Révocation de privilèges objets

#### • Syntaxe

REVOKE { object\_priv | ALL [ PRIVILEGES ] }

ON [ schema. ] object

FROM { user | role | PUBLIC } [ CASCADE CONSTRAINTS ]

#### • Notes

- **CASCADE CONSTRAINTS** : s'emploie avec le privilège REFERENCES, supprime les contraintes d'intégrité mises.
- **Retrait d'un privilège et**
- **WITH GRANT OPTION**:
- Si un utilisateur U1 a affecté un privilège P1 à U2 et U2 l'a affecté à U3, le retrait à U2 entraîne le retrait à U3 : le retrait se fait en cascade.

#### • Exemples

```
sql>REVOKE DELETE ON Bonus FROM scott ;
```

```
sql>REVOKE UPDATE ON emp FROM public;
```

```
sql>REVOKE REFERENCES ON scott.emp FROM BRAHIM ;
```

```
sql>REVOKE ALL ON bonus FROM PUBLIC ;
```

# Gestion de la sécurité et des ressources

## ■ Les privilèges Objets

### ■ Visualisation des privilèges objets

DBA_TAB_PRIVS	DBA_COL_PRIVS
ALL_TAB_PRIVS	ALL_COL_PRIVS
USER_TAB_PRIVS	USER_COL_PRIVS
ALL_TAB_PRIVS_MADE	DBA_COL_PRIVS
USER_TAB_PRIVS_MADE	ALL_COL_PRIVS_MADE
USER_TAB_PRIVS_MADE	USER_COL_PRIVS_MADE
ALL_TAB_PRIVS_RECD	ALL_COL_PRIVS_RECD
USER_TAB_PRIVS_RECD	ALL_COL_PRIVS_RECD
TABLE_PRIVILEGES	COLUMN_PRIVILEGES

### ■ Principales Colonnes de vues ci-dessus

GRANTEE	: utilisateur ayant reçu le privilège
OWNER	: propriétaire de la table
TABLE_NAME	: nom de la table
COLUMN_NAME	: Nom de la colonne concerné
GRANTOR	: Utilisateur ayant affecté le privilège
PRIVILEGE	: privilège affecté
GRANT	: privilège reçu.

### ■ Les privilèges Objets

#### ■ Visualisation des privilèges objets

- Visualisation de tous les droits sur les objets de la base

```
SELECT * FROM dba_tab_privs  
WHERE table_name = 'BONUS' OR  
table_name = 'EMP';
```

<u>GRANTEE</u>	<u>OWNER</u>	<u>TABLE</u>	<u>NAME</u>	<u>GRANTOR</u>	<u>PRIVILEGE</u>
BRAHIM	SCOTT	BONUS		SCOTT	ALTER
BRAHIM	SCOTT	BONUS		SCOTT	DELETE
BRAHIM	SCOTT	BONUS		SCOTT	INDEX
BRAHIM	SCOTT	BONUS		SCOTT	INSERT
BRAHIM	SCOTT	BONUS		SCOTT	SELECT
BRAHIM	SCOTT	BONUS		SCOTT	UPDATE
BRAHIM	SCOTT	BONUS		SCOTT	REFERENCES

#### ■ Tous les droits sur toutes les colonnes des tables dans la base

```
SELECT * FROM dba_col_privs ;
```



## ■ Les rôles (Généralités)

### ■ Définition

- Un rôle est un concept Oracle qui permet de regrouper plusieurs privilèges et / ou rôles afin de les affecter ou retirer en bloc à un utilisateur et / ou un rôle.

### ■ un rôle facilite la gestion des privilèges

### ■ l'affectation d'un rôle à un utilisateur peut se faire sous Oracle ou à travers l'OS

### ■ pour des raisons de sécurité, un **mot de passe** peut être assigné à un rôle

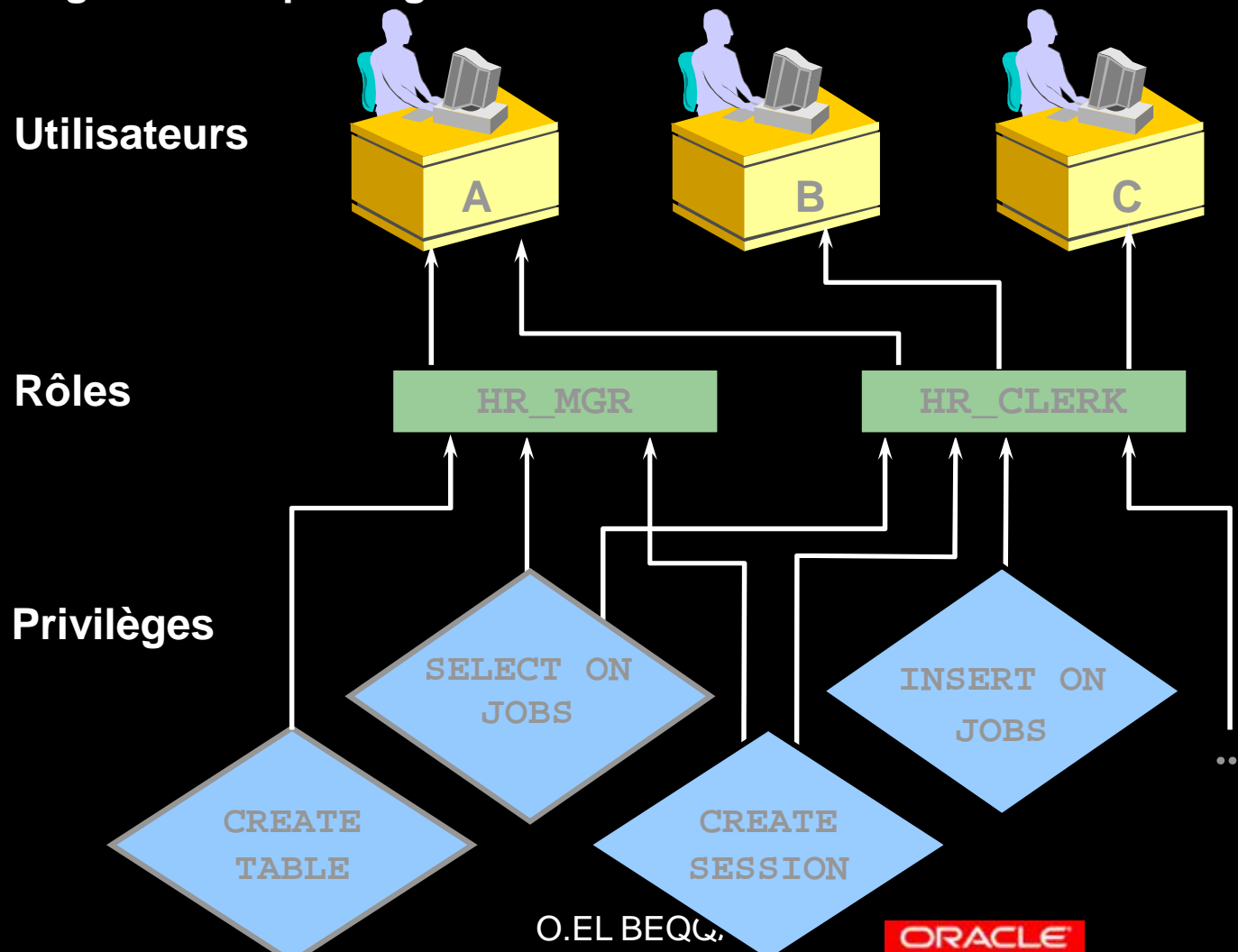
### ■ oracle fournit un certain nombre de **rôles par défaut** (connect, resource, dba, exp\_full\_database, imp\_full\_data\_base, select\_catalog\_role, delete\_catalog\_role / execute\_catalog\_role, ...)

### ■ **pour créer un rôle**, il faut **avoir le privilège "CREATE ROLE"**

# Gestion de la sécurité et des ressources

## ■ Les rôles (Généralités)

- Assignation de privilèges aux utilisateurs : SANS ROLES ou VIA UN ROLE



# Gestion de la sécurité et des ressources

## ■ Les rôles (Création)

- A sa création, un rôle ne contient aucun privilège

- Syntaxe

**CREATE ROLE** *role1*

[ { **NOT IDENTIFIED**

| **IDENTIFIED** { **BY** *password* | **EXTERNALLY** | **GLOBALLY** |

**USING** *package* } ]

- Mots clés et paramètres

- **role** : nom du rôle à créer
- **NOT IDENTIFIED** : permet de créer un rôle sans mot de passe
- **Password** : mot de passe assigné au rôle
- **EXTERNALLY** : mot de passe est contrôlé au niveau de l'OS (*Identifié de manière externe*)
- **GLOBALLY** : Rôle autorisé au niveau de l'annuaire
- **USING package** : rôle applicatif

- Exemple

sql> **CREATE ROLE** *rl\_etudiant* ;

sql> **CREATE ROLE** *rl\_admin\_secu* **IDENTIFIED BY** *secu\_pass* ;

sql> **CREATE ROLE** *hr\_manager* **IDENTIFIED EXTERNALLY**;

## ■ Les rôles (Modification)

- On peut modifier le *niveau de sécurité* d'un rôle
- privilège requis pour modifier un rôle ALTER ANY ROLE.

- **Syntaxe**

```
ALTER ROLE role { NOT IDENTIFIED  
| IDENTIFIED { BY password | EXTERNALLY |  
Globally | USING package} }
```

- **Mots clés et paramètres**

- **role** : nom du rôle à créer
- **NOT IDENTIFIED** : permet d'inhiber le mot de passe d'un rôle
- **Password** : nouveau mot de passe assigné au rôle
- **EXTERNALLY** : mot de passe contrôlé au niveau de l'OS
- **GLOBALLY** : Rôle autorisé au niveau de l'annuaire
- **USING package** : rôle applicatif

- **Exemple**

```
sql> ALTER ROLE ROLE rl_etudiant IDENTIFIED EXTERNALLY ;  
sql> ALTER ROLE rl_admin_backup IDENTIFIED BY backup_pass;  
sql> ALTER ROLE rl_admin_secu NOT IDENTIFIED;
```

### ■ Les rôles (Suppression)

- Un rôle supprimé est retiré IMMEDIATEMENT du domaine de sécurité de l'utilisateur (connecté ou non) ou du rôle l'ayant reçu
- Le privilège DROP ANY ROLE et le fait d'avoir acquis un rôle avec WITH ADMIN OPTION permettent de le supprimer

- **Syntaxe**

**DROP ROLE** Nom\_du\_role ;

- **Exemple**

**DROP ROLE** rl\_admin\_secu ;

# Gestion de la sécurité et des ressources

## ■ Les Rôles

### ■ Affectation de privilèges ou de rôles à un rôle

#### ■ Exemple

- Création de deux rôles

- # rôle rassemblant les privilèges pour se connecter

- # rôle rassemblant les privilèges pour administrer la sécurité

- CREATE ROLE rl\_connect ;**

- CREATE ROLE rl\_admin\_secu ;**

#### Affectation des privilèges aux rôles

- GRANT create session, alter session, Restricted session TO rl\_connect ;**

- GRANT create role, create user, create profile TO rl\_admin\_secu;**

#### Affectation d'un Rôle à un autre Rôle

- GRANT rl\_connect TO rl\_admin\_secu ;**

### ■ Affectation de privilèges à un rôle

#### ■ Privilèges ne pouvant être affectés à un ROLE

- Privilège Système

- **UNLIMITED TABLESPACE**

- Ce privilège inhibe tous les quotas et autorise l'utilisateur à créer des objets dans n'importe quel tablespace.

- Privilèges Objets

- INDEX

- # droit de créer un index sur les tables d'autres utilisateurs

- REFERENCES

- # droit de référencer une table dans le schéma d'autres utilisateurs

# Gestion de la sécurité et des ressources

## ■ Les Rôles (Affectation d'un rôle à un Utilisateur)

- Elle peut se faire au niveau :
  - Oracle
  - du Système d'Exploitation (OS)
  - De l'annuaire de l'entreprise

### ■ Affectation d'un Rôle au niveau Oracle

**GRANT role to user [WITH ADMIN OPTION]**

L'utilisateur ayant reçu le rôle avec WITH ADMIN OPTION peut le réaffecter, supprimer ou modifier.

### ■ Affectation d'un rôle au niveau de l'OS

- Positionner le paramètre OS\_ROLE dans init.ora afin que l'affectation et la révocation des rôles se fassent au niveau de l'OS  
OS\_ROLE = TRUE
- Déclarer (sous UNIX) dans le fichier de groupe chaque rôle comme étant un groupe
  - **Syntaxe** `ora_<SID>_<role>[_[D][A]] : [user1, [user2], [ ...]]`

#### • Avec

<b>SID</b>	: nom de l'instance
<b>rôle</b>	: nom du rôle
<b>D</b>	: rôle par défaut
<b>A</b>	: WITH ADMIN OPTION

#### • Exemple :

- » `ora_COURS_rl_connect_D:scott, mopolos, tintin`
- » `ora_COURS_rl_admin_secu_DA:mopolos,osmani`

## Rôles prédéfinis

NOM DU ROLE	PRIVILEGES AFFECTES AU ROLE
<b>CONNECT</b>	CREATE SESSION,
<b>RESOURCE</b>	CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER
<b>DBA</b>	tous les privilèges WITH ADMIN OPTION
<i>exp_full_database</i>	SELECT ANY TABLE, BACKUP ANY TABLE, INSERT, DELETE AND UPDATE ON tables SYS.INCVID, SYS.INCFIL, SYS.INCEXP
<i>imp_full_database</i>	BECOME USER, WRITEDOWN (trusted Oracle)
<i>Execute_catalog_role</i>	Privilège d'exécuter les procédures du dictionnaire
<i>Select_catalog_role</i>	Privilège de consulter tout le dictionnaire Oracle
<i>Delete_catalog_role</i>	Privilège de supprimer la table d'audit aud\$
<i>Recovery_catalog_owner</i>	Fournit les privilèges pour le propriétaire du catalogue de recouvrement
<i>Aq_administrator_role</i>	Fournit les privilèges pour l'administration des Queue
<i>Snmpagent</i>	Fournit les privilèges utiles à l'agent SNMP d'entreprise manager

**l'affectation des rôles RESOURCE et DBA à un utilisateur entraîne un  
GRANT *unlimited tablespace***



## Gestion de la sécurité et des ressources : RÔLES

### ■ Les rôles ( obtenir des informations)

#### ■ interrogez les vues suivantes du dictionnaire de données :

- **DBA\_ROLES** : Tous les rôles qui existent dans la base de données
- **DBA\_ROLE\_PRIVS** : Rôles accordés à des utilisateurs et à des rôles
- **ROLE\_ROLE\_PRIVS** : Rôles accordés à des rôles
- **DBA\_SYS\_PRIVS** : Privilèges système accordés à des utilisateurs et à des rôles
- **ROLE\_SYS\_PRIVS** : Privilèges système accordés à des rôles
- **ROLE\_TAB\_PRIVS** : Privilèges objet accordés à des rôles
- **SESSION\_ROLES** : Rôles activés par l'utilisateur

**Exemple 1** : listing de tous les rôles de la base

```
sql> SELECT * FROM dba_roles ;
```

<u>ROLE</u>	<u>PASSWORD</u>
CONNECT	NO
RESOURCE	NO
DBA	NO
EXP_FULL_DATABASE	NO
IMP_FULL_DATABASE	NO
MONITORER	NO
RL_ADMIN_SECU	NO
RL_CONNECT	NO

**Exemple 2** : liste des rôles affectés à un role ou un user.

```
sql>SELECT * FROM dba_role_privs WHERE grantee = 'RL_ADMIN_SECU' ;
```

<u>GRANTEE</u>	<u>GRANTED-ROLE</u>	<u>ADM-DEF</u>	<u>Default</u>
RL_ADMIN_SECU	RL_CONNECT	NO	YES

**Exemple 3** : liste des rôles actifs pour la session

```
sql> SELECT * FROM session_roles;
```

ROLE  
DBA.....

## Gestion de la sécurité et des ressources : RÔLES

- Les profiles
  - Un **profile** est un concept Oracle qui **permet** à l'administrateur d'une base de **contrôler la consommation des ressources systèmes et des mots de passes**
  - Il existe un profile par défaut appelé **DEFAULT**. Il est par défaut affecté à un utilisateur lors de sa création
  - Les limites du profile DEFAULT sont positionnées à UNLIMITED
  - Le profile DEFAULT ne peut être supprimé. *Les limites de ce profile peuvent par contre être modifiées*
  - **activation et contrôle des limites :**
    - dans le fichier initSID.ora positionner :  
***RESOURCE\_LIMIT = TRUE***
    - ou dynamiquement faire sous sqlplus par exemple :  
**SQL> ALTER SYSTEM SET resource\_limit = true;**

# Gestion de la sécurité et des ressources : RÔLES

## ■ Les Profiles (Création)

- Privilège requis CREATE PROFILE
- **Syntaxe partie limite des ressources**

CREATE PROFILE profile LIMIT

[ SESSIONS\_PER\_USER { integer | UNLIMITED | DEFAULT } ]

[ CPU\_PER\_SESSION { integer | UNLIMITED | DEFAULT } ]

[ CPU\_PER\_CALL { integer | UNLIMITED | DEFAULT } ]

[ CONNECT\_TIME { integer | UNLIMITED | DEFAULT } ]

[ IDLE\_TIME { integer | UNLIMITED | DEFAULT } ]

[ LOGICAL\_READS\_PER\_SESSION { integer | UNLIMITED | DEFAULT } ]

[ LOGICAL\_READS\_PER\_CALL { integer | UNLIMITED | DEFAULT } ]

[ COMPOSITE\_LIMIT { integer | UNLIMITED | DEFAULT } ]

[ PRIVATE\_SGA { integer [K | M] | UNLIMITED | DEFAULT } ]

## ■ Mots clés et paramètres

- **Session\_per\_user** : Nombre maximum de sessions par utilisateur
- **Logical\_read\_per\_session** : Nbre de blocs de données à lire pour une session
- **cpu\_per\_session** : temps CPU max par session en % de secondes
- **cpu\_per\_call** : temps CPU pour un appel (en cas de parse, execute ou fetch) en % de secondes
- **connect\_time** : temps écoulé maximum (en minutes)
- **idle\_time** : temps maximum d'inactivité.
- **private\_sga** : taille privée de la SGA allouée à un utilisateur
- **unlimited** : limite de la ressource illimitée
- **default** : prend la limite par défaut de la ressource

## Gestion de la sécurité et des ressources : RÔLES

### ■ Les Profiles (Création)

- Privilège requis CREATE PROFILE

- **Syntaxe partie password**

CREATE PROFILE profile LIMIT

[FAILED\_LOGIN\_ATTEMPTS {expr | UNLIMITED | DEFAULT}]

[PASSWORD\_LIFE\_TIME {expr | UNLIMITED | DEFAULT}]

[PASSWORD\_REUSE\_TIME {expr | UNLIMITED | DEFAULT}]

[PASSWORD\_REUSE\_MAX {expr | UNLIMITED | DEFAULT}]

[PASSWORD\_LOCK\_TIME {expr | UNLIMITED | DEFAULT}]

[PASSWORD\_GRACE\_TIME {expr | UNLIMITED | DEFAULT}]

[PASSWORD\_VERIFY\_FUNCTION {function, NULL, DEFAULT}]

- **Mots clés et paramètres**

- **Failed\_login\_attempts** : nombre d'échecs avant le blocage du compte
- **password\_life\_time** : durée en jours avant l'expiration du mot de passe
- **password\_reuse\_time** : durée en jours avant la réutilisation d'un password
- **password\_reuse\_max** : nombre de modif du password avant réutilisation
- **password\_lock\_time** : durée en jours du verrouillage d'un compte
- **password\_grace\_time** : délai de tolérance du password avant son expiration
- **password\_verify\_function** : fonction de contrôle des mots de passes

# Gestion de la sécurité et des ressources : RÔLES

## ■ Les Profiles (Création)

### ■ Exemple 1

```
CREATE PROFILE pf_secretaire LIMIT
sessions_per_user 2
cpu_per_session unlimited
cpu_per_call 1000
logical_reads_per_session unlimited
logical_reads_per_call 100
idle_time 30
connect_time 480 ;
```

### ■ Exemple 2

```
CREATE profile pf_agent LIMIT
sessions_per_user 2
cpu_per_session unlimited
cpu_per_call 1000
composite_limit 20000
private_sga 32K ;
```

### ■ Exemple 3

```
CREATE PROFILE pf_admin
PASSWORD_LIFE_TIME 200
LIMIT PASSWORD_REUSE_MAX DEFAULT
PASSWORD_REUSE_TIME UNLIMITED
CPU_PER_SESSION UNLIMITED
```

## Gestion de la sécurité et des ressources : RÔLES

### ■ Assignation d'un profile à un utilisateur

- A la création d'un nouvel utilisateur  
`CREATE USER rackham IDENTIFIED BY lerouge`  
`PROFILE pf_secretaire ;`
- A la modification d'un utilisateur  
`ALTER USER rackham`  
`PROFILE pf_agent ;`

### ■ Modification d'un profile

- Privilège requis : `ALTER PROFILE`

### ■ Suppression d'un profile

- En cas de suppression d'un profile existant affecté à un utilisateur, ce dernier se verra automatiquement attribué le profile DEFAULT
- Le profile DEFAULT ne peut être supprimé
- Privilège requis : `DROP PROFILE`
- **Syntaxe**  
`DROP PROFILE nom_profile [CASCADE]`
- **Exemple**  
`sql>DROP PROFILE pf_secretaire CASCADE ;`

## Gestion de la sécurité et des ressources : RÔLES

### ■ Les Profiles (Visualisation des informations des profiles)

- Vues contenant les informations sur les profiles :
  - dba\_profiles, resource\_cost, user\_resource\_limit
- **Exemple 1** : Liste de tous les profiles

```
sql> SELECT profile, resource_name, limit FROM dba_profiles  
ORDER BY profile, resource_name;
```

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	COMPOSITE_LIMIT	UNLIMITED
...		
DEFAULT	CPU_PER_SESSION	600
PF_AGENT	COMPOSITE_LIMIT	20000
PF_AGENT	CONNECT_TIME	DEFAULT
PF_AGENT	CPU_PER_CALL	1000
PF_AGENT	PRIVATE_SGA	32768
PF_AGENT	SESSIONS_PER_USER	2
...		
PF_SECRETAIRE	COMPOSITE_LIMIT	DEFAULT
PF_SECRETAIRE	CONNECT_TIME	480
PF_SECRETAIRE	CPU_PER_CALL	1000
PF_SECRETAIRE	CPU_PER_SESSION	UNLIMITED
PF_SECRETAIRE	IDLE_TIME	30
PF_SECRETAIRE	LOGICAL_READS_PER_CALL	100
PF_SECRETAIRE	SESSIONS_PER_USER	2

## Gestion de la sécurité et des ressources

### ■ Les utilisateurs

- La notion d'utilisateur est fondamentale pour accéder aux données d'une base Oracle
- Le site d'un client Oracle doit être tenu à jour au niveau des licences :
  - les paramètres de *init.ora* pour le contrôle de la licence (valeurs par défaut 0) : *license\_max\_session*, *license\_sessions\_warning*, *license\_max\_users*
- Si la limite en nombre de sessions est illimitée conserver les valeurs par défaut
  - Seuls les utilisateurs avec le privilège RESTRICTED SESSION peuvent se connecter en cas de dépassement de la limite
  - en cas de mise en oeuvre de l'architecture parallèle, chaque instance à ses limites mais la somme doit équivaloir à la somme des limites du site
  - visualisation des limites des licences **v\$license**



# Gestion de la sécurité et des ressources

## ■ Les utilisateurs

### ■ Contrôle de la limitation du nombre d'utilisateurs

- Au moment du lancement d'une instance

```
LICENSE_MAX_USERS = 80
```

- Au moment où l'instance tourne

```
sqlplus >ALTER SYSTEM
```

```
SET LICENSE_MAX_USERS=100;
```

### ■ Authentification des utilisateurs

- A partir d'Oracle

```
CREATE USER scott IDENTIFIED BY tiger ;
```

- A partir de l'OS

```
CREATE USER ali IDENTIFIED EXTERNALLY ;
```

- Les utilisateurs authentifiés par l'OS sont précédés d'une chaîne définie par le paramètre de `initsid.ora` `OS_AUTHENT_PREFIX` qui vaut par défaut **OPS\$**

- Globalement à partir de l'annuaire LDAP

```
CREATE USER scott IDENTIFIED GLOBALLY AS
```

```
'CN=scott,OU=division1,O=oracle,C=US'
```

## Etablir des rôles par défaut

- Un utilisateur peut se voir accorder un grand nombre de rôles.
- Un utilisateur peut se voir accorder un rôle par défaut.
- Vous pouvez limiter le nombre de rôles par défaut d'un utilisateur.

```
ALTER USER scott  
        DEFAULT ROLE hr_clerk, oe_clerk;
```

```
ALTER USER scott DEFAULT ROLE ALL;
```

```
ALTER USER scott DEFAULT ROLE ALL EXCEPT  
        hr_clerk;
```

```
ALTER USER scott DEFAULT ROLE NONE;
```

## Rôles d'application

- Seuls les packages PL/SQL autorisés peuvent activer des rôles d'application
- La clause de package `USING` permet de créer un rôle d'application

```
CREATE ROLE admin_role  
IDENTIFIED USING hr.employee;
```

## Activer et désactiver les rôles

- Désactivez un rôle accordé à un utilisateur pour le révoquer temporairement
- Activez un rôle pour l'accorder temporairement
- La commande SET ROLE permet d'activer et de désactiver les rôles
- Les rôles par défaut d'un utilisateur sont activés à la connexion
- Un mot de passe peut être requis pour activer un rôle

## Activer et désactiver les rôles

```
■ SET ROLE hr_clerk;
```

```
SET ROLE oe_clerk  
IDENTIFIED BY order;
```

```
SET ROLE ALL EXCEPT oe_clerk;
```

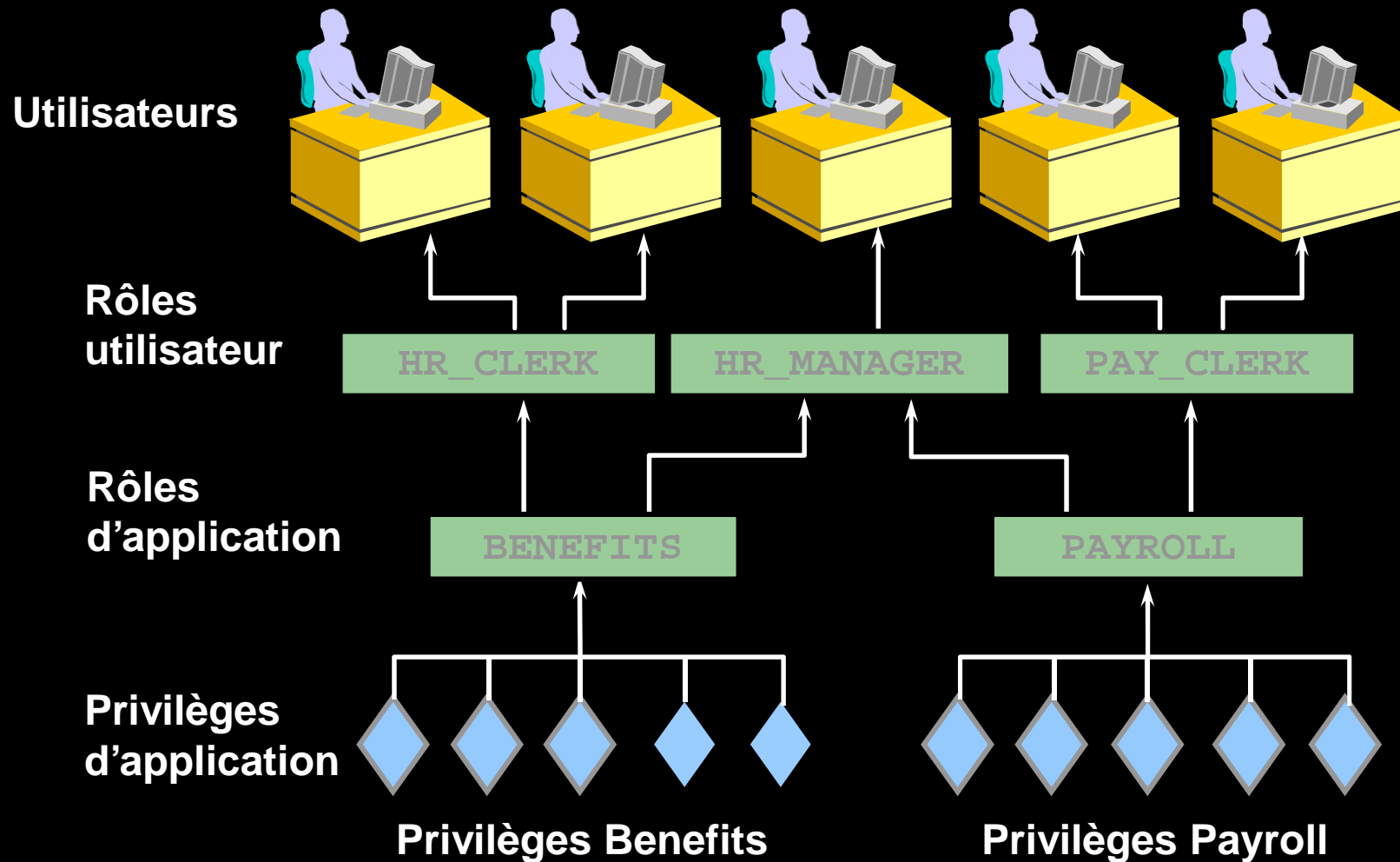
## Révoquer des rôles accordés à des utilisateurs

- La révocation d'un rôle accordé à un utilisateur requiert l'option  
ou le privilège .
- Pour révoquer un rôle, utilisez la syntaxe suivante :

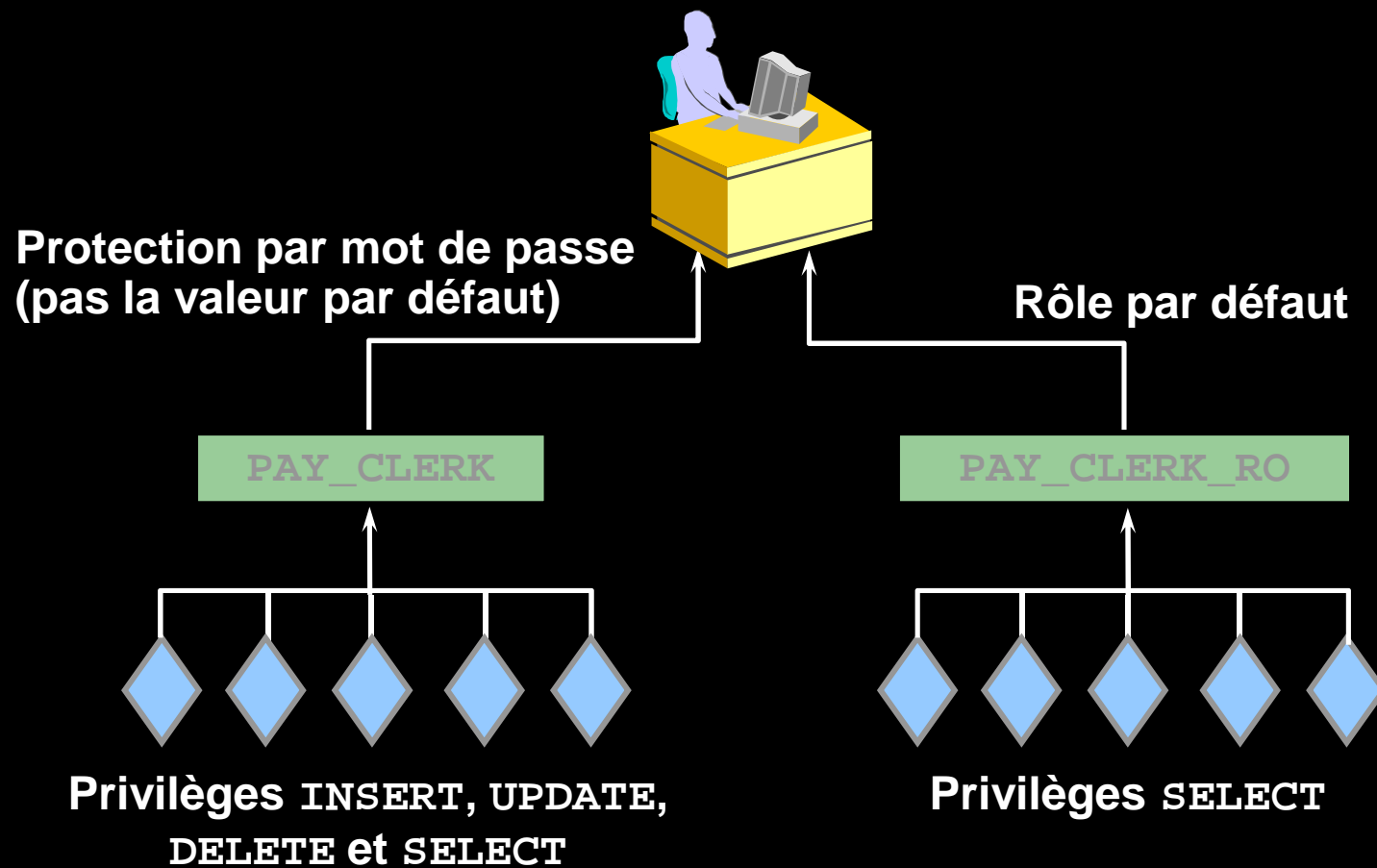
```
REVOKE oe_clerk FROM  
scott;
```

```
REVOKE hr_manager FROM  
PUBLIC;
```

# Instructions relatives à la création de rôles



# Règles d'utilisation des mots de passe et des rôles par défaut





# Utilisateurs et Sécurité



# Schéma de Base de Données

- Tables
- Triggers
- Contraintes
- Index
- Vues
- Séquences
- Procédures stockées
- Synonymes
- Types de données définis par l'utilisateur
- Database Links...

## Liste de Contrôle pour la Création d'utilisateurs

1. Choisissez un nom d'utilisateur et un mécanisme d'authentification
2. Identifiez les tablespaces dans lesquels l'utilisateur doit stocker des objets
3. Décidez des quotas pour chaque tablespace
4. Affectez un tablespace par défaut et un tablespace temporaire
5. Créez le compte utilisateur
6. Accordez des privilèges et des rôles à l'utilisateur

## Création d'un Nouvel utilisateur:

```
CREATE USER USER1  
IDENTIFIED BY user1  
DEFAULT TABLESPACE data01  
TEMPORARY TABLESPACE temp  
QUOTA 15m ON TS1  
PASSWORD EXPIRE;
```

## Les traitements stockés dans la base

- Les traitements sont définis avec le langage PL/SQL et sont stockés dans les tables du dictionnaire de données.
- **types de traitements:**
  - **La procédure** dont l'unique rôle est d'effectuer un traitement.
  - **La fonction** qui effectue un traitement pour renvoyer en retour une valeur.
  - **Le package** qui regroupe un ensemble de procédures et / ou de fonctions.
  - **Le trigger** (déclencheur) qui est déclenché automatiquement lors d'une mise à jour sur une table.

# Les procédures

## Syntaxe de création / modification

```
CREATE          [OR          REPLACE]          PROCEDURE
  [nom_user.]nom_procédure  (arguments IN type[,
  argument IN type, ...])          {IS|AS}
```

```
[Variable_locale type_variable_locale;]
```

```
BEGIN
```

```
{ contenu du bloc PL }
```

```
END  [nom_procédure];
```

```
/
```

### APPEL DIRECT SOUS SQLPLUS

**EXECUTE Nomproc(param,...)**

### DANS UN PROGRAMME PL /SQL OU UNE AUTRE PROCEDURE

**Nom\_proc (param,...);**

## Les procédures (exemple)

*Création d'une procédure qui compte le nombre d'employés pour un numéro de département donné.*

```
CREATE OR REPLACE PROCEDURE
  p_Empl ( v_deptno IN  number(4)) IS
  v_no NUMBER;
BEGIN
  SELECT COUNT(empno)
    INTO v_no
    FROM emp
    WHERE deptno =v_deptno;
  DBMS_OUTPUT.PUT_LINE('Nombre d'employés
  ||v_no);
END;
/
```

*Exécution :*      **Sql> execute p\_Empl(30);**

Procédures dans la base : *SELECT object\_name FROM user\_objects  
WHERE object\_type = 'PROCEDURE';*

## Les procédures (exemple2 )

```
SQL>CREATE OR REPLACE PROCEDURE  
raise_salary  
(v_id IN emp.empno%TYPE) IS  
BEGIN  
    UPDATE emp  
    SET sal = sal * 1.1  
    WHERE empno = v_id;  
END raise_salary;  
/
```

Procedure created.

```
SQL> EXECUTE raise_salary (7389)  
PL / SQL procedure successfully completed
```



## PROCEDURES / PARAMETRES

```
SQL> CREATE PROCEDURE add_dept
(v_name IN dept.dname%TYPE DEFAULT
'inconnu ',
v_loc IN dept.loc%TYPE DEFAULT
'inconnu ')
IS
BEGIN
INSERT INTO dept (dept_deptno.NEXTVAL,
v_name,v_loc);
END add_dept;
```

```
SQL> BEGIN
add_dept (' TRAINING ', 'FES ');
END
/
```

## Les procédures (utilisation)

### Exécution

Il faut avoir le privilège objet: EXECUTE

### Exécution sous SQL\*PLUS

*EXECUTE [nom\_propriétaire.]nom\_procedure  
{(Liste arguments)};*

### Exécution dans le corps d'un programme PL/SQL

BEGIN

    [nom\_propriétaire.]nom\_procedure {(liste  
arguments)}

END

### Suppression

*DROP PROCEDURE nom\_procedure;*

# Les fonctions

## ■ Syntaxe de création / modification

```
CREATE          [OR          REPLACE]          FUNCTION
  [nom_user.]nom_function  (arguments  IN  type
  [,argument      IN      type_arg,...    ]    RETURN
  type_val
  {IS | AS }
```

```
[Variable_locale type_variable_locale;]
```

```
BEGIN
```

```
{ contenu du bloc PL }
```

```
RETURN variable_a_retourner;
```

```
END [nom_function]
```

```
END;
```

## Les fonctions (exemple)

Création d'une fonction qui compte le nombre d'employés qui travaillent dans un service donné.

```
CREATE OR REPLACE FUNCTION F_SERVICE (V_E_EMP IN  
E_EMP. SERVICE_NO% TYPE) RETURN NUMBER  
IS  
V_NO NUMBER(3);  
BEGIN  
SELECT COUNT(NO) INTO V_NO FROM EMP  
WHERE SERVICE_NO = V_E_EMP;  
RETURN(V_NO);  
END ;  
/
```

## **Les fonctions** (exemple/utilisation)

```
CREATE OR REPLACE FUNCTION get_sal  
(v_id IN emp.empno%TYPE)  
RETURN NUMBER IS  
v_salary emp.sal%TYPE := 0;  
BEGIN  
  SELECT sal INTO v_salary FROM emp  
    WHERE empno = v_id;  
RETURN (v_salary);  
END get_sal;  
/
```

```
SQL> select ename from emp where sal >  
get_sal(7369)
```

# COMPARAISON FONCTIONS / PROCEDURES

PROCEDURE	FONCTION
EXECUTE UN BLOC PL / SQL	INVOQUE COMME EXPRESSION
PAS DE DONNEES DE TYPE RETURN	DOIT CONTENIR AU MOINS UNE DONNEE DE TYPE RETURN
PEUT RETOURNER UNE OU PLUSIEURS VALEURS	DOIT RETOURNER UNE VALEUR

# Les fonctions

## Recherche d'une fonction dans la base

```
SELECT object_name  
      FROM user_objects  
     WHERE object_type = 'FUNCTION'
```

## Recherche du code source d'une fonction dans la base

```
SELECT text  
      FROM user_source WHERE name  
     'NOM_FONCTION' ORDER BY line;
```

## Suppression

```
DROP FUNCTION nom_fonction;
```

## Exemple Factoriel....Fonctions / Procédures

*create or replace **procedure** Fact(n IN int, fact OUT int)*  
*is*

*begin*

*fact := 1;*

*for i in 1..n*

*Loop*

*fact := fact \* i;*

*end loop;*

*end;*

*/*

=====Exécution=====

**declare**

**res int;**

**begin**

**Fact(5,res);**

**dbms\_output.put\_line('Résultat : ' || res);**

**end;**

**/**

Version procédure



## Exemple Factoriel....Fonctions / Procédures

*create or replace* **function** *f\_Fact*(*n in int*) *return*  
*integer is*

*begin*

**if** (*n* = 1) *then* *return*(1);  
**else** *return*(*n* \* *f\_Fact*(*n*-1));

**end if**;

*end*;

/

====*Exécution*====

**declare**

*res int*;

**begin**

*res* := *f\_Fact*(5);

*dbms\_output.put\_line*('Résultat : ' || *res*);

**end**;

/

Version Fonction

# Exemple

Soit la table table Patient (id\_P, Température Number);

*Create or replace Function Conversion(Deg\_F in Number)*

*Return number is*

*deg\_c number;*

*Begin*

*Deg\_c := (5.0 / 9.0) \* (Deg\_f - 32);*

*Return Deg\_c;*

*End Conversion*

*/*

EXEMPLE d'utilisation de cette fonction dans Select suivante:

**SQL> Select id\_p, Température ,  
Conversion(Température) From Patient;**

# Les packages

- \* Un package est l'encapsulation d'objets dans une même unité logique de traitement
  - Procédures
  - Fonctions
  - Exceptions
  - variables, curseurs, constantes
- \* Construit en 2 parties:

**SPECIFICATION & BODY**

## Les packages (suite)

Un package est un objet de la base de données qui **encapsule** d'autres objets (procédures, fonctions ..)

- Un package a essentiellement deux parties:
  - **la partie déclaration**
  - **la partie corps** du programme: dans cette partie, sont définies les procédures et les fonctions et les curseurs

## Package E/S

Le package **DBMS\_OUTPUT** : Permet de stocker l'information dans un tampon avec les modules PUT et **PUT\_LINE**.

**Les procédures de DBMS\_OUTPUT:**

- **PUT\_LINE** : affiche une chaîne de caractères
- **GET\_LINE** : extrait une ligne du tampon de sortie.

Autres....

**DBMS\_REFRESH.REFRESH** (....)

```
Sql> execute DBMS_OUTPUT.PUT_LINE('message');  
[set serveroutput on]
```

## PACKAGES / SYNTAXE

- ***CREATE [OR REPLACE] PACKAGE***

package\_name IS|AS  
element\_package;

...

END package\_name;

- ***CREATE [OR REPLACE] PACKAGE BODY***

package\_name IS|AS  
declaration de type ou de variables globales;  
specifications package;  
END package\_name;

## SEQUENCES

- Une séquence est un objet virtuel qui ne contient aucune donnée utilisé pour générer automatiquement des valeurs (NUMBER).

***CREATE SEQUENCE nomSéquence***

***INCREMENT BY entier ]***

***[START WITH entier ]***

***[ { MAXVALUE entier / NOMAXVALUE } ]***

***[ { MINVALUE entier / NOMINVALUE } ]***

***USER\_SEQUENCES [.....]***

- **\*Pseudo-colonnes :**
- CURRVAL retourne la valeur courante
- NEXTVAL incrémente la séquence et retourne la valeur obtenue

## SEQUENCES (suite)

**Exemple :** *create sequence seq1 start with 1 increment by 1 MAXVALUE 30;*

- \* **seq1.CURRVAL** qui retourne la valeur courante de la séquence (lecture seule) ;
- \* **seq1.NEXTVAL** qui incrémente la séquence et retourne la nouvelle valeur de celle-ci (écriture et lecture).
- Modification d'une séquence : *(le privilège ALTER ANY SEQUENCE est requis)*

**ALTER SEQUENCE Nom\_seq INCREMENT BY 5  
MAXVALUE 850;**

**\*Affichage d'une séquence :**

- **SELECT seq2.CURRVAL FROM DUAL;**



## Séquences (*manipulations*)

- Suppression d'une séquence

***DROP SEQUENCE seq2;***

- UTILISATION

- Il s'agit de créer une séquence pour la gestion des numéros des étudiants

- *create table etudiant (id\_etu number primary key , nom varchar2(20), prenom varchar2(20), age number );*

- **---Création d'une séquence pour id\_etu---**

- insertion :

- *insert into etudiant values (seq\_etu.nextval, 'Ali', 'Med', 27);*

- ***SELECT seq\_etu.CURRVAL "valeur courante" from dual ;***

# Les triggers (1)

## Définition

Un trigger stocké est un traitement procédural lié à une table et une seule, et donc répertorié dans le dictionnaire de données.

- Il se déclenche automatiquement lors d'un événement intervenu sur la table dont il dépend: insertion, suppression ou mise à jour
- Il est valide tant que la table existe.
- Il peut être actif ou inactif

## Les triggers (2)

- **2 types de trigger:**
  - Déclenchement sur chaque ligne mise à jour
  - Déclenchement une seule fois pour la mise à jour

### **2 séquencements:**

- **Avant** la mise à jour
  - **Après** la mise à jour
- **12 triggers possibles sur une table**

## Les triggers stockés (3)

### ■ Création / Modification du trigger

```
CREATE [OR REPLACE ] TRIGGER [nom_user.]nom_trigger
    { BEFORE | AFTER } { INSERT | UPDATE | DELETE }
    [OR { INSERT | UPDATE | DELETE } ]
    ON nom_table
    [FOR EACH ROW [ WHEN (prédicat) ]]
```

```
DECLARE
```

```
--Déclaration de variables locales au trigger;
```

```
BEGIN
```

```
    { contenu du bloc PL }
```

```
END;
```

```
/
```

## Les triggers stockés (4)

### Désactivation d'un trigger

```
ALTER TRIGGER [nom_user.]nom_trigger  
DISABLE;
```

### Réactivation d'un trigger

```
ALTER TRIGGER [nom_user.]nom_trigger  
ENABLE;
```

### Désactivation de tous les triggers sur une table

```
ALTER TABLE Nom_table DISABLE ALL  
TRIGGERS;
```

### Réactivation de tous les triggers sur une table

```
ALTER TABLE nom_table ENABLE ALL  
TRIGGERS;
```

### Suppression du trigger

```
DROP TRIGGER [nom_user.]nom_trigger;
```

## Les triggers (Old – New)

### Utilisation des variables «OLD.» et «NEW.»

Elles sont utilisables uniquement dans les triggers FOR EACH ROW.

Dans le bloc PL/SQL, on peut faire référence aux colonnes en les préfixant avec **:NEW.nom\_colonne** ou **:OLD.nom\_colonne** avec les restrictions suivantes:

**:NEW** uniquement dans les triggers INSERT ou UPDATE (On ne peut pas modifier la valeur de :NEW.nom\_colonne dans les triggers AFTER).

**:OLD** uniquement dans les triggers UPDATE ou DELETE (On ne peut pas modifier la valeur de :OLD.nom\_colonne).

## Les triggers (exemple)

*Sauvegarder dans une autre table historique une ligne avant chaque suppression dans la table emp*

```
CREATE OR REPLACE TRIGGER del_emp
BEFORE DELETE ON emp
FOR EACH ROW
BEGIN
INSERT INTO tab_traces VALUES ( :OLD.empno, :OLD.ename,
SYSDATE);
END;
/
```

### Recherche d'un trigger dans la base

```
SELECT trigger_body
FROM user_triggers
WHERE trigger_name = 'NOM TRIGGER'
```

## Exercice :

Soit la table Emp(NumEmp, NomEmp, SalaireEmp)

*Le salaire d'un employé ne peut baisser.*

Cette contrainte ne peut être prise en compte qu'en utilisant les triggers ou bien les procédure stockées.

```
CREATE TRIGGER Mod_Sal  
AFTER UPDATE OF Emp ON Salaire  
FOR EACH ROW  
BEGIN  
    If new.Salaire < old.Salaire Rollback;  
END;    [erreurs à corriger ...atelier]
```

***A suivre part II***





# FEFERENCES

- ***Documents Oracle (white papers)***
- ***<http://www.oracle.com>***
- ***...***