

LES OBJETS D'UNE BASE DE DONNÉES

2021/2022

2BI

Les objets

- Les tables
- Les vues
- Les séquences
- Les indexes

1) Les tables

- Créer une table:

- CREATE TABLE nom_de_la_table
(colonne1 type_donnees,
colonne2 type_donnees,
colonne3 type_donnees,
colonne4 type_donnees);

NB: On peut ajouter des contraintes (exp: PRIMARY KEY, UNIQUE, NOT NULL, etc.).

- Modifier une table:

- ALTER TABLE nom_table instruction;

On modifie une table pour ajouter, modifier, supprimer une colonne ou une contrainte.

- Supprimer une table:

- DROP TABLE nom_table;

- Renommer une table:

- RENAME ancien_nom TO nouveau_nom;

1) Les tables

- Une table stocke des données.
 - `SELECT * FROM employees;`

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17/06/87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21/05/91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25/06/97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07/02/99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/94	FI_ACCOUNT	9000	-	108	100

2) Les vues

- Une vue est une table virtuelle. Elle concerne une ou plusieurs tables de base.
- Une vue est une fenêtre à travers laquelle les données peuvent être vues ou changées.
- Elle est le résultat d'une requête SELECT.
- Le contenu des vues change lorsque le contenu des tables qui entrent dans la description de la vue change.

2) Les vues

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17/06/87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21/05/91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25/06/97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07/02/99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/94	FI_ACCOUNT	9000	-	108	100

EMPLOYEE_ID	LAST_NAME	SALARY
107	Lorentz	4200
108	Greenberg	12000
109	Faviet	9000

2) Les vues

- Pourquoi utiliser les vues?
 - **Eviter** les requêtes longues et compliquées.
 - Restreindre l'accès aux données pour garantir la **confidentialité** et l'**intégrité** des données.
 - Augmenter l'**indépendance logique** (une vue donne la possibilité de récupérer des données à partir de plusieurs tables).

2) Les vues

Recréer la vue même si elle existe

Créer la vue même si les tables de base n'existent pas

CREATE [OR REPLACE] [FORCE|**NOFORCE**] VIEW nom_vue [(alias[, alias]...)]

AS subquery

[WITH CHECK OPTION [CONSTRAINT constraint]]

[WITH READ ONLY [CONSTRAINT constraint]];

Les noms pour les expressions sélectionnées par la sous requête

Pour garantir que seulement les lignes accessibles via la vue peuvent être insérées ou mises à jour

La requête SELECT

Rend la vue en mode lecture (interdire l'utilisation du DML)

2) Les vues

- Afficher la structure d'une vue:
 - `DESCRIBE nom_vue;`
- Récupérer les données à partir d'une vue:
 - `SELECT * FROM nom_vue [where condition];`
- Supprimer une vue:
 - `DROP nom_vue;`
- Renommer une vue:
 - `RENAME ancien_nom TO nouveau_nom;`

2) Les vues

- Les vues figures dans les tables systèmes **ALL_CATALOG**, **USER_VIEWS** et **ALL_VIEWS**.

```
SELECT * FROM ALL_VIEWS;
```

- Il existe deux types de vues:

1) Vue simple:

- Récupère des données à partir d'une seule table.
- Ne contient pas de fonctions ou de groupe de données.
- Permet l'exécution des opérations DML.

2) Vue complexe:

- Récupère des données à partir de plusieurs tables.
- Contient des fonctions ou groupe de données.
- Ne permet pas toujours l'exécution des opérations DML.

3) Les séquences

- Une séquence est un objet créé par un utilisateur et peut être partagé avec plusieurs utilisateurs.
- Les séquences sont souvent utilisées pour la création de valeurs pour les clés primaires.
- Une séquence peut être utilisée pour plusieurs tables.

3) Les séquences

```
CREATE SEQUENCE nom_sequence  
[INCREMENT BY n]  
[START WITH n]  
[{MAXVALUE n | NOMAXVALUE}  
{MINVALUE n | NOMINVALUE}]  
[{CYCLE | NOCYCLE}]  
[{CACHE n | NOCACHE}];
```

(par défaut 1)

(par défaut 1)

(la valeur 10^{27} dans le cas asc et -1 dans le cas desc)

(la valeur 1 dans le cas asc et - 10^{26} dans le cas desc)

(par défaut Oracle met en cache 20 valeurs)

3) Les séquences

- Avoir la valeur courante d'une séquence:
 - `SELECT nom_sequence.CURRVAL FROM dual;`
- Avoir la prochaine valeur d'une séquence :
 - `SELECT nom_sequence.NEXTVAL FROM dual;`
- Modifier une séquence:
`ALTER SEQUENCE nom_sequence`
`[INCREMENT BY n]`
`[START WITH n]`
`[{MAXVALUE n | NOMAXVALUE}`
`[{MINVALUE n | NOMINVALUE}]`
`[{CYCLE | NOCYCLE}]`
`[{CACHE n | NOCACHE}];`
- Supprimer une séquence :
 - `DROP SEQUENCE nom_sequence;`

Souvent utilisée quand la valeur maximale est atteinte.

4) Les index

- Un index est un objet de la base de données.
- Il est utilisé par le serveur d'Oracle pour accéder plus rapidement aux lignes d'une table en utilisant des pointeurs.
- Il est utilisé automatiquement par le serveur.
- Les index sont créés:
 - Automatiquement: un index est créé automatiquement quand on définit une clé primaire ou une colonne unique lors de la création et la définition d'une table.
 - Manuellement: on peut créer manuellement des index sur des colonnes pour augmenter la vitesse d'accès aux lignes d'une table.

4) Les index

- Créer un index:

- `CREATE INDEX nom_index ON nom_table(colonne1 [, col2] ...);`

- Supprimer un index :

- `DROP INDEX index_name;`

L'ARCHITECTURE D'ORACLE

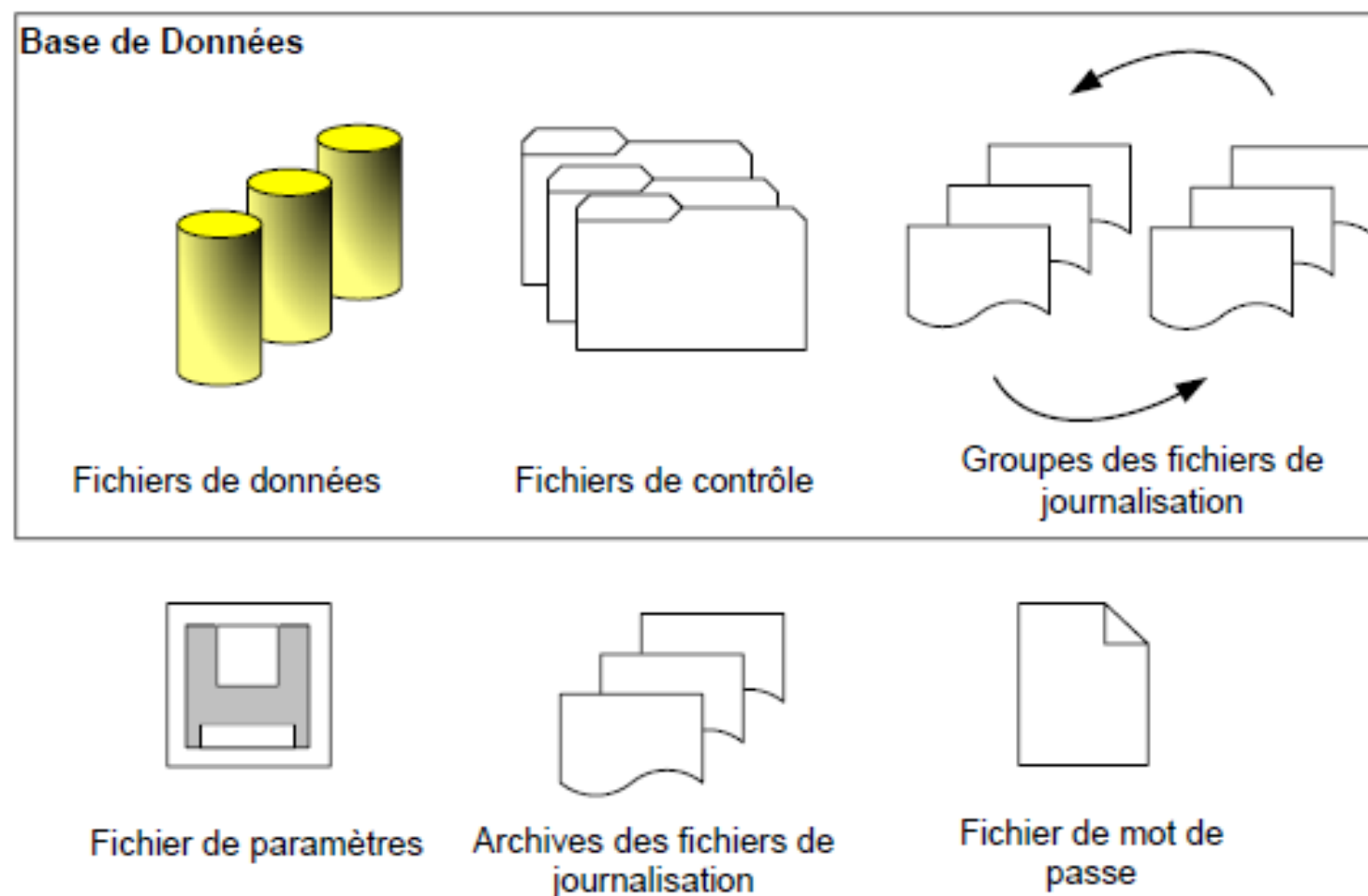
2021/2022

2BI

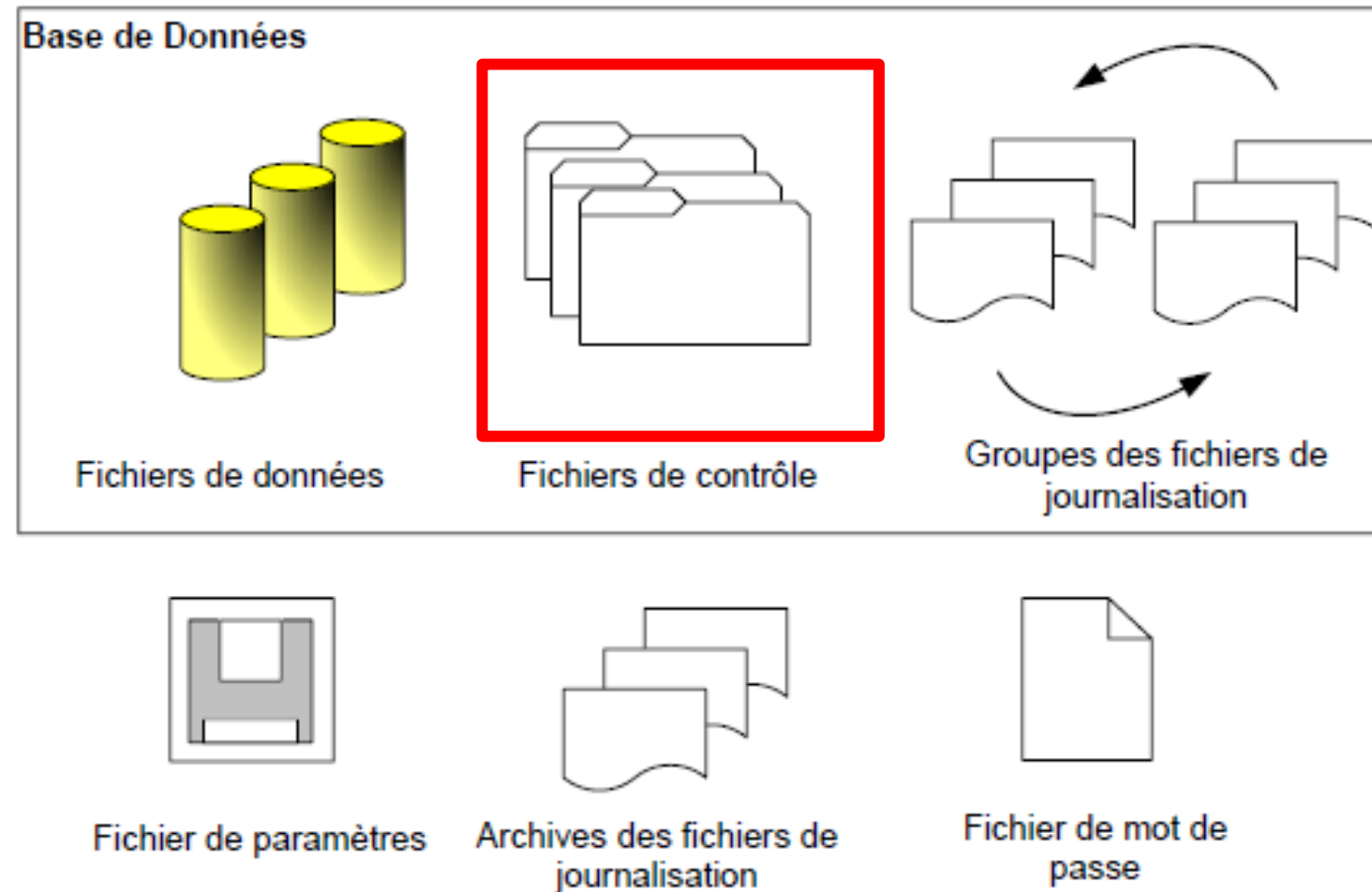
Architecture d'Oracle Database

- Un serveur de base de données oracle inclut deux composantes importantes:
 - **La base de données**: est l'ensemble **de fichiers** qui contiennent entre autres les données, les informations de la base ainsi que le journal de modification des données.
 - **L'instance**: est un ensemble de **processus** et de **zones mémoire** qui facilitent l'accès et la manipulation de la base de données.

La base de données: les fichiers physiques



La base de données: les fichiers physiques



La base de données: Fichiers de contrôle

- Un fichier de contrôle contient des informations sur la structure physique de la base de données:
 - Le nom de la base de données
 - Les noms, les chemins et les tailles des fichiers de données et de journalisation
 - Les informations de restauration de la BD
 - Etc.
- Il est créé pendant la création de la base et il est modifié en permanence.
- Le fichier de contrôle est primordial pour le lancement correcte de l'instance (s'il est endommagé la BD ne peut pas être chargée).
- Les informations sur les fichiers de contrôle sont examinées à partir de la vue `v$CONTROLFILE`

La base de données: Fichiers de contrôle

- Travail à faire :
 - Affichez la description de la vue V\$CONTROLFILE via: DESCRIBE v\$CONTROLFILE

```
SQL> describe v$controlfile
```

Nom	NULL ?	Type
STATUS		VARCHAR2(7)
NAME		VARCHAR2(513)
IS_RECOVERY_DEST_FILE		VARCHAR2(3)
BLOCK_SIZE		NUMBER
FILE_SIZE_BKLS		NUMBER

La base de données: Fichiers de contrôle

Column	Datatype	Description
STATUS	VARCHAR2 (7)	INVALID if the name cannot be determined (which should not occur); NULL if the name can be determined
NAME	VARCHAR2 (513)	Name of the control file
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	Indicates whether the file was created in the fast recovery area (YES) or not (NO)
BLOCK_SIZE	NUMBER	Control file block size
FILE_SIZE_BKLS	NUMBER	Control file size (in blocks)
CON_ID	NUMBER	The ID of the container to which the data pertains. Possible values include:

- 0: This value is used for rows containing data that pertain to the entire CDB. This value is also used for rows in non-CDBs.
- 1: This value is used for rows containing data that pertain to only the root
- *n*: Where *n* is the applicable container ID for the rows containing data

La base de données: Fichiers de contrôle

- Travail à faire :

- Affichez la description de la vue V\$CONTROLFILE via: DESCRIBE v\$CONTROLFILE;

```
SQL> describe v$controlfile
```

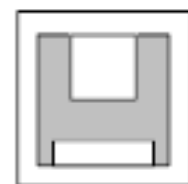
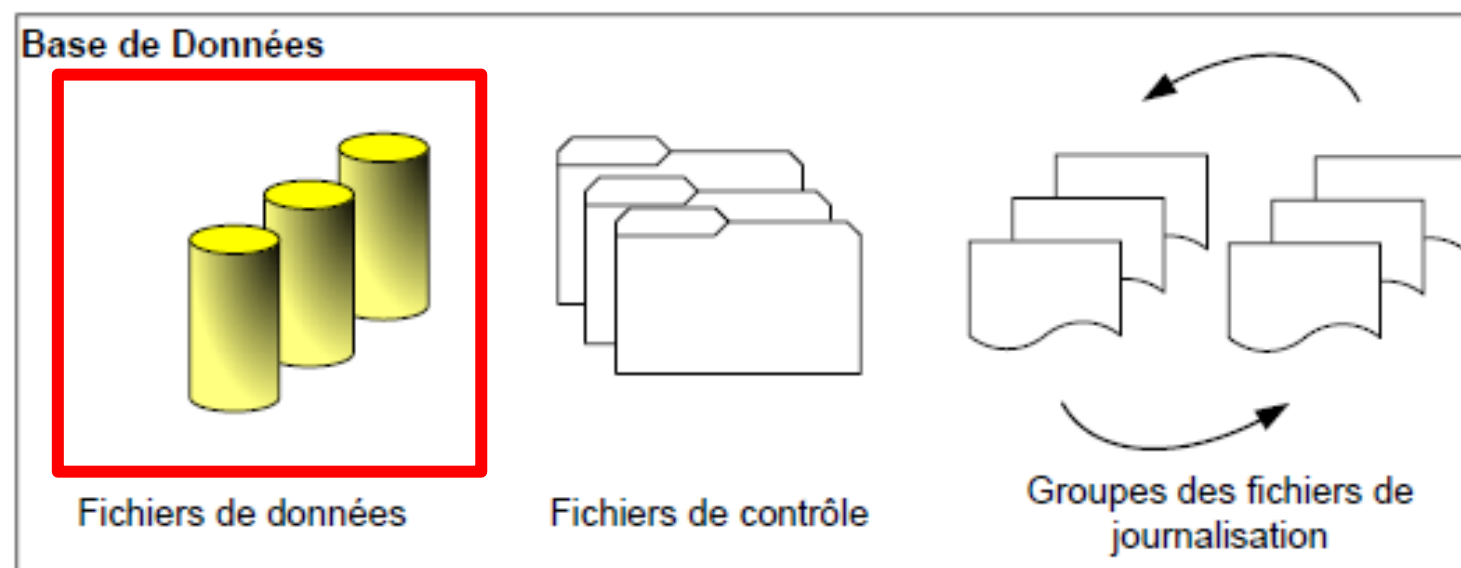
Nom	NULL ?	Type
STATUS		VARCHAR2(7)
NAME		VARCHAR2(513)
IS_RECOVERY_DEST_FILE		VARCHAR2(3)
BLOCK_SIZE		NUMBER
FILE_SIZE_BKLS		NUMBER

- Affichez le contenu de la vue: SELECT * FROM V\$CONTROLFILE;

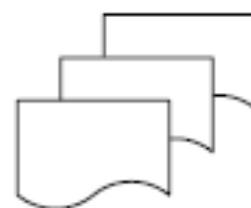
```
SQL> select * from v$controlfile;
```

STATUS	NAME	IS_	BLOCK_SIZE	FILE_SIZE_BKLS
	C:\ORACLE\XE\ORADATA\XE\CONTROL.DBF			
NO			16384	430

La base de données: les fichiers physiques



Fichier de paramètres



Archives des fichiers de
journalisation



Fichier de mot de
passe

La base de données: Fichiers de données

- Des **fichiers physiques** qui stockent les données de la base sous format spécial Oracle.
- Les fichiers de données sont logiquement regroupés en structures logiques appelées **tablespaces**.
- Un fichier de données ne peut être associé qu'à **un seul tablespace** et à **une seule base de données**.
- Les vues `DBA_TABLESPACES` et `DBA_DATA_FILES` incluent toutes les informations respectivement relatives aux tablespaces et aux fichiers de données de la base.

La base de données: Fichiers de données

```
SQL> SELECT tablespace_name, file_name FROM DBA_DATA_FILES  
2 ORDER BY tablespace_name;
```

TABLESPACE_NAME

FILE_NAME

SYSAUX

C:\ORACLEXE\ORADATA\XE\SYSAUX.DBF

SYSTEM

C:\ORACLEXE\ORADATA\XE\SYSTEM.DBF

UNDO

C:\ORACLEXE\ORADATA\XE\UNDO.DBF

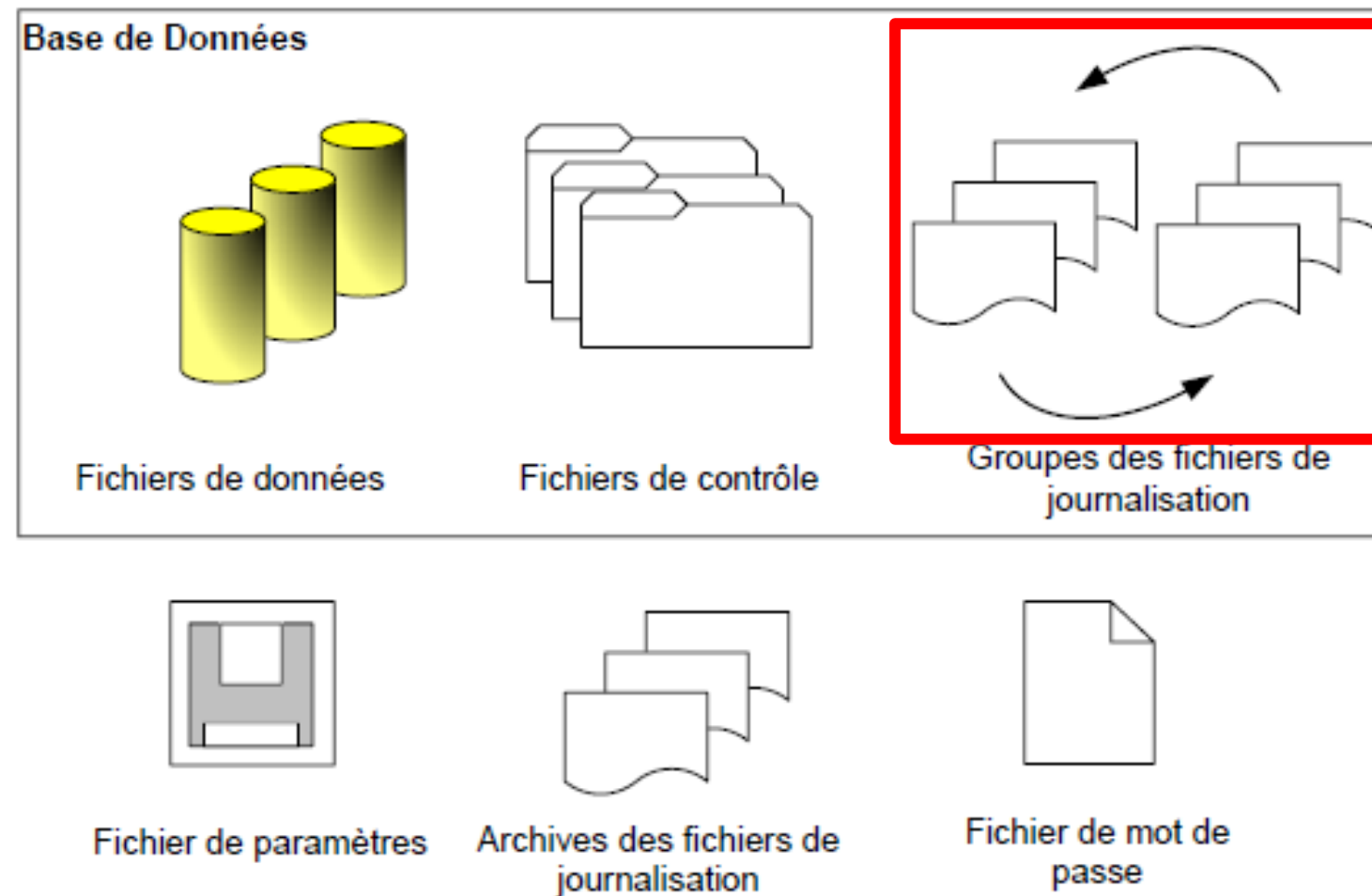
TABLESPACE_NAME

FILE_NAME

USERS

C:\ORACLEXE\ORADATA\XE\USERS.DBF

La base de données: les fichiers physiques



La base de données: Fichiers de journalisation

- Les modifications réalisées par les utilisateurs sur les données, sont sauvegardées dans les fichiers de journalisation (Redo-log).
- Les fichiers de journalisation sont utilisés en cas de crash du système afin de **reconstituer** les données.
- Un fichier de journalisation est sous format spécial Oracle et l'écriture dans ce fichier est assurée par le processus **LGWR** (Log Writer).
- Les informations sur les fichiers journaux existent dans la vue `V$LOGFILE`.

La base de données: Fichiers de journalisation

- Travail à faire :
 - Affichez la description de la vue V\$LOGFILE via: `DESCRIBE V$LOGFILE;`

```
SQL> describe v$logfile;
```

Nom	NULL ?	Type
GROUP#		NUMBER
STATUS		VARCHAR2(7)
TYPE		VARCHAR2(7)
MEMBER		VARCHAR2(513)
IS_RECOVERY_DEST_FILE		VARCHAR2(3)

La base de données: Fichiers de journalisation

Column	Datatype	Description
GROUP#	NUMBER	Redo log group identifier number
STATUS	VARCHAR2 (7)	Status of the log member: <ul style="list-style-type: none">• INVALID - File is inaccessible• STALE - File's contents are incomplete• DELETED - File is no longer used• null - File is in use
TYPE	VARCHAR2 (7)	Type of the logfile: <ul style="list-style-type: none">• ONLINE• STANDBY
MEMBER	VARCHAR2 (513)	Redo log member name
IS_RECOVERY_DEST_FILE	VARCHAR2 (3)	Indicates whether the file was created in the fast recovery area (YES) or not (NO)

La base de données: Fichiers de journalisation

- Travail à faire :
 - Affichez le contenu de la vue V\$LOGFILE via: `SELECT * FROM V$LOGFILE;`

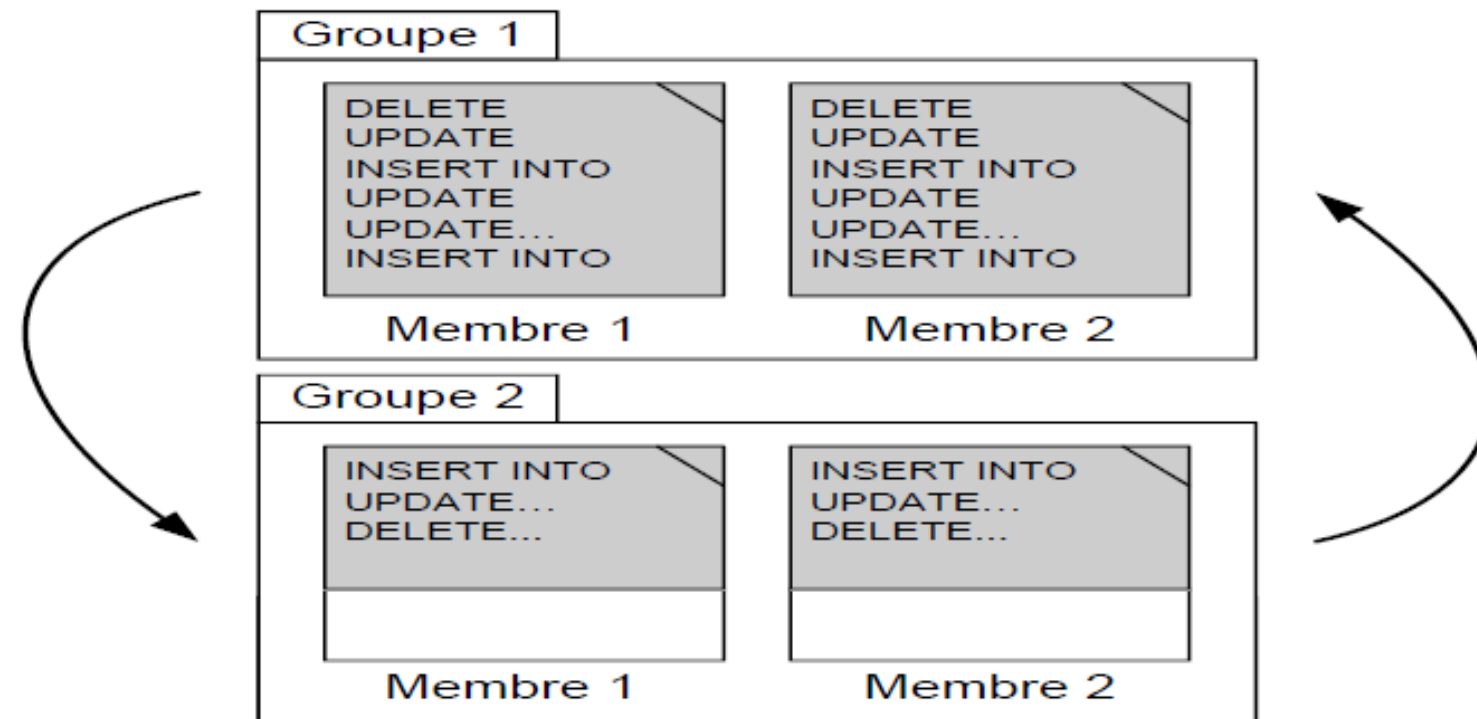
```
SQL> select * from v$logfile;
```

GROUP#	STATUS	TYPE
MEMBER		
IS_		
2	STALE	ONLINE
C:\ORACLE\XE\APP\ORACLE\FLASH_RECOVERY_AREA\XE\ONLINELOG\01_MF_2_K1ZGYSOF_.LOG		
YES		
1	ONLINE	
C:\ORACLE\XE\APP\ORACLE\FLASH_RECOVERY_AREA\XE\ONLINELOG\01_MF_1_K1ZGYRPO_.LOG		
YES		
GROUP#	STATUS	TYPE
MEMBER		
IS_		

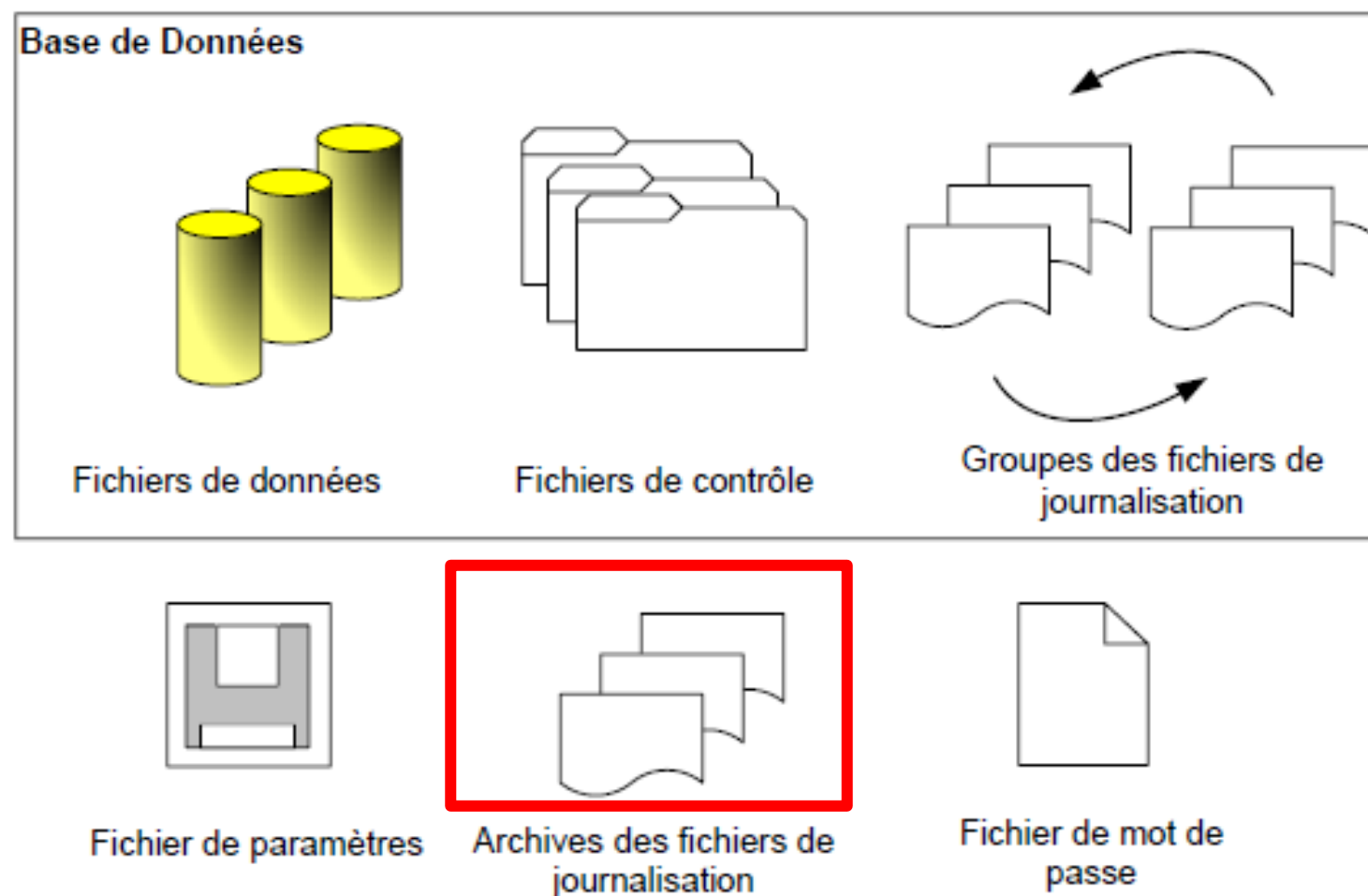
La base de données: Fichiers de journalisation

- L'écriture sur les fichiers journaux est **multiplexée** et **cyclique**.
- L'ensemble des fichiers multiplexés sont appelés **membres** et forment un **groupe**.
- L'écriture est multiplexée à **l'intérieur** d'un groupe, et **cyclique** entre les groupes.
- Il existe au minimum **deux groupes** de fichiers journaux.

La base de données: Fichiers de journalisation

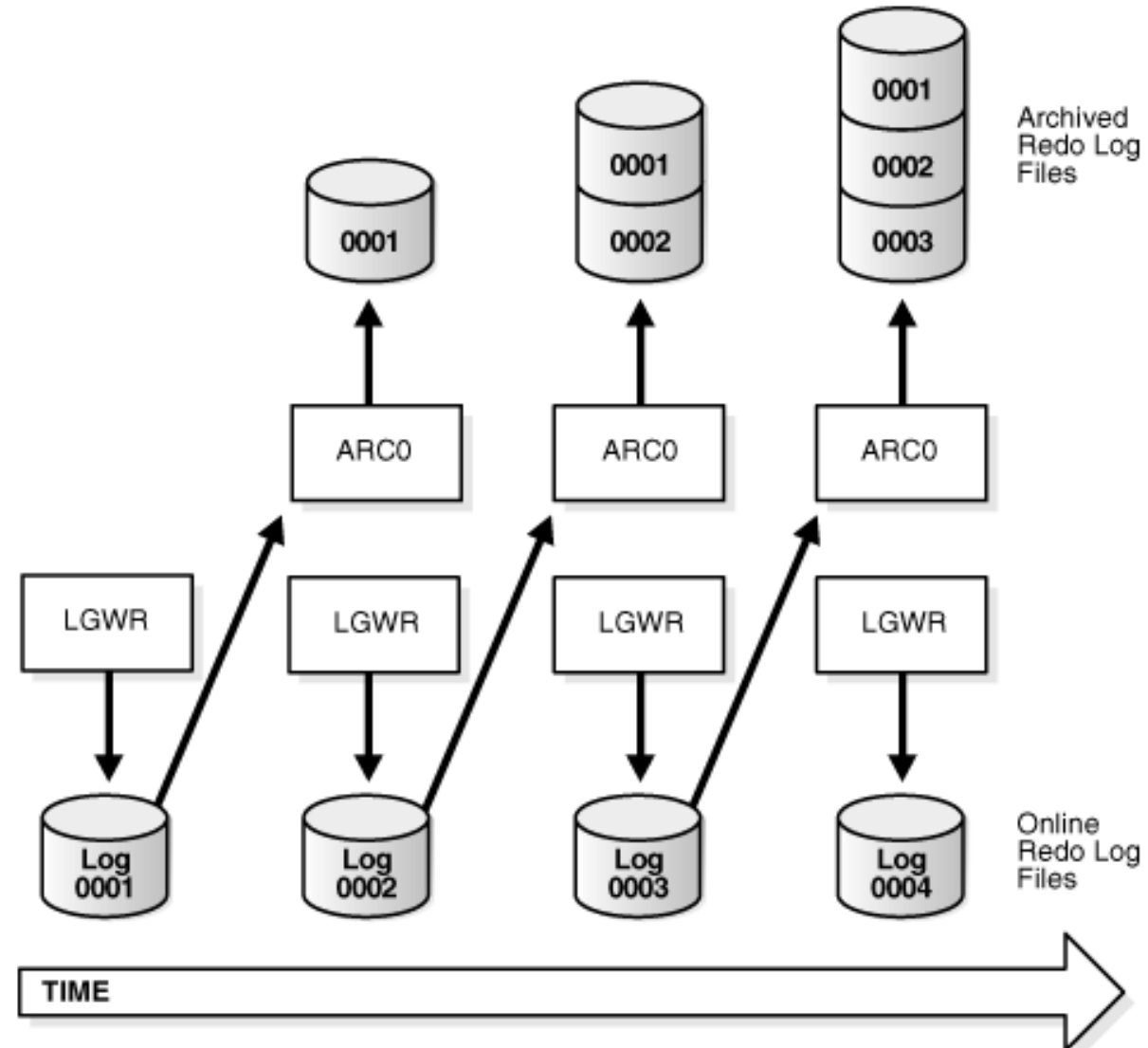


La base de données: les fichiers physiques



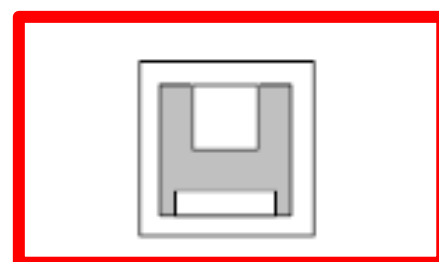
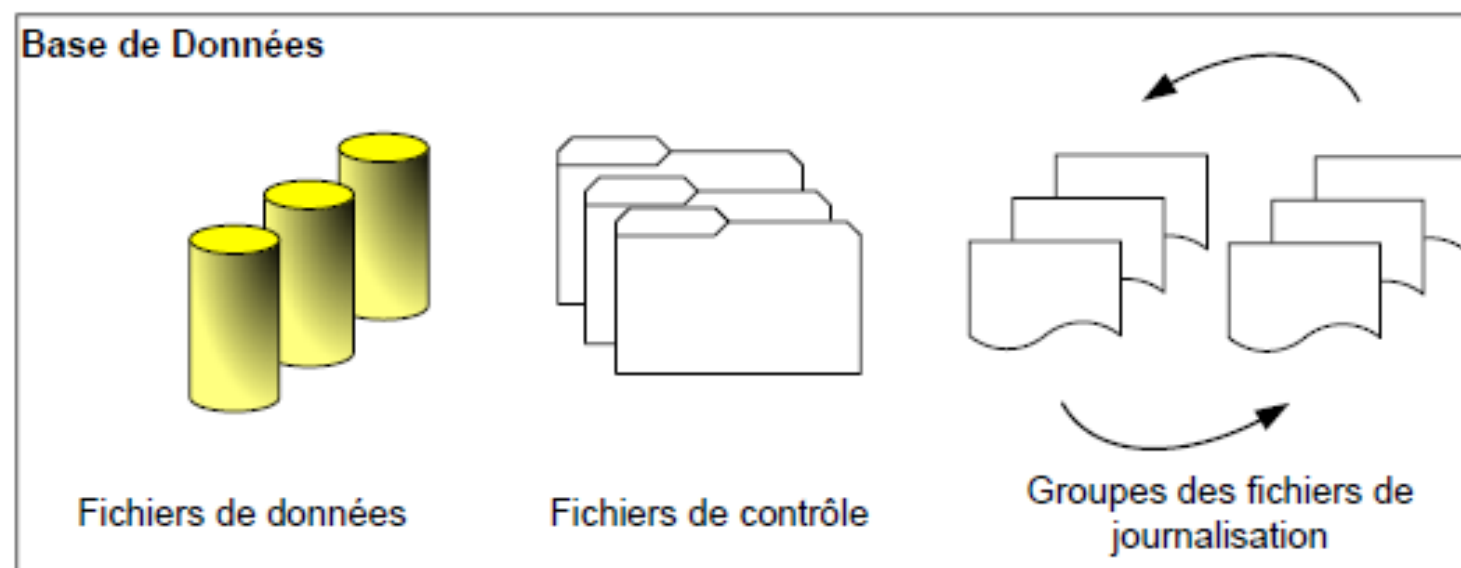
La base de données: Archive des fichiers de journalisation

- L'écriture sur les fichiers journaux est **cyclique** → certaines transactions sont **écrasées**.
- Pour une **sécurité optimale**, il faut archiver les fichiers de journalisation avant de finir le cycle de sauvegarde des transactions (avant d'écraser les fichiers).
- Cette option est offerte par le mode ARCHIVELOG (NB: une BD n'est pas forcément en mode ARCHIVELOG).
- Les fichiers journaux archivés peuvent être stockés sur des disques (serveurs) **distants** pour optimiser la sécurité de la BD.

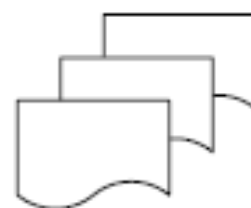


Source: oracle.com

La base de données: les fichiers physiques



Fichier de paramètres



Archives des fichiers de
journalisation

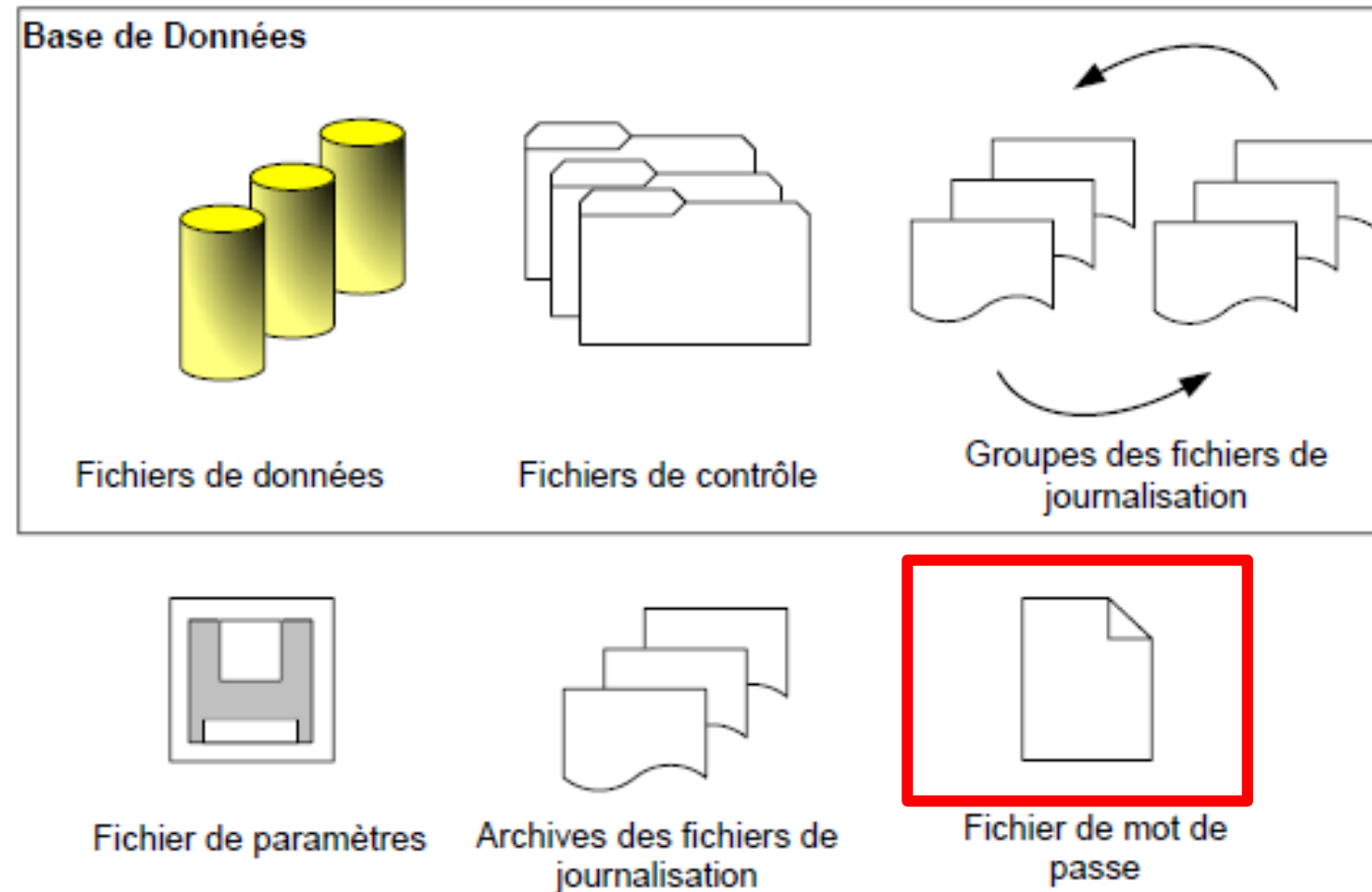


Fichier de mot de passe

La base de données: Fichier de paramètres

- Un fichier de paramètres contient les paramètres de configuration qui conditionnent l'**initialisation** et le **fonctionnement** de l'instance.
- Il existe deux fichiers de paramètres: **SPFILE** et **PFILE**.

La base de données: les fichiers physiques



La base de données: Fichier de mot de passe

- Le fichier de mot de passe est un fichier physique qui contient l'ensemble des administrateurs de la BD avec leurs mots de passe.
- Afin de démarrer l'instance, l'administrateur de la BD doit s'identifier. Son identification se produit à partir du fichier de mot de passe.

L'instance

- L'instance est l'ensemble de **structures mémoires** et de **processus** qui assurent l'accès et la gestion d'une base de données.
- Le **fichier de paramètres** est utilisé pour configurer l'instance lors de son démarrage (par défaut SPFILE mais on peut utiliser PFILE).
- Une instance ne peut ouvrir **qu'une seule** base de données à la fois.

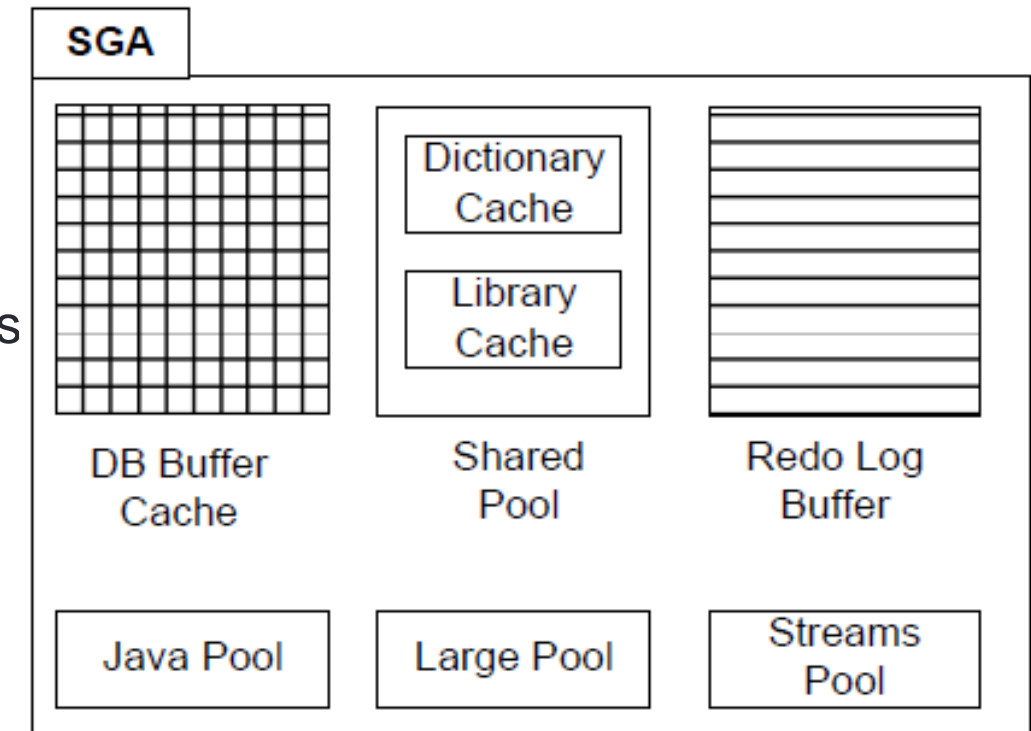
L'instance: les structures mémoires

- Une instance emploie deux structures principales:
 - la **SGA** (System Global Area)
 - la **PGA** (Program Global Area).

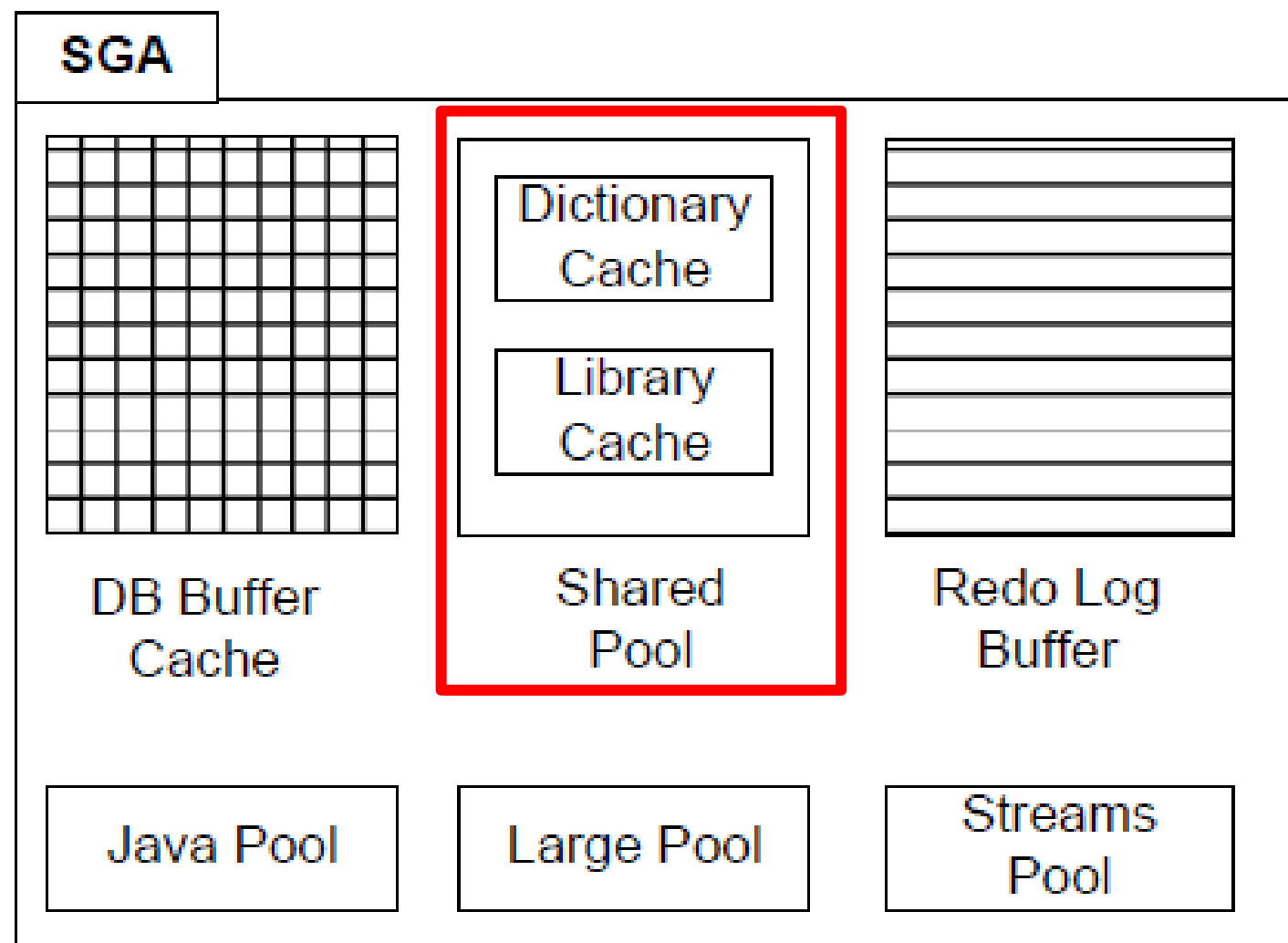
L'instance: les structures mémoires

■ System Global Area (SGA):

- Un espace de la mémoire centrale partagé par les différents processus de l'instance.
- Allouée au démarrage de la base de données et libérée lors de son arrêt.
- Constituée de trois zones obligatoires et de trois zones optionnelles qui sont allouées au démarrage en fonction des valeurs des paramètres du fichier de paramètres.
- Plus la SGA est grande, plus le serveur est performant.
- Sa taille maximale est définie via le paramètre **SGA_MAX_SIZE**.
- Le paramètre **LOCK_SGA** permet à l'administrateur d'obliger (ou non) Oracle à allouer la zone SGA exclusivement en mémoire physique, sans ayant recours à la mémoire virtuelle.
- Les tailles des zones mémoire peuvent être définies manuellement ou automatiquement (**SGA_TARGET**).

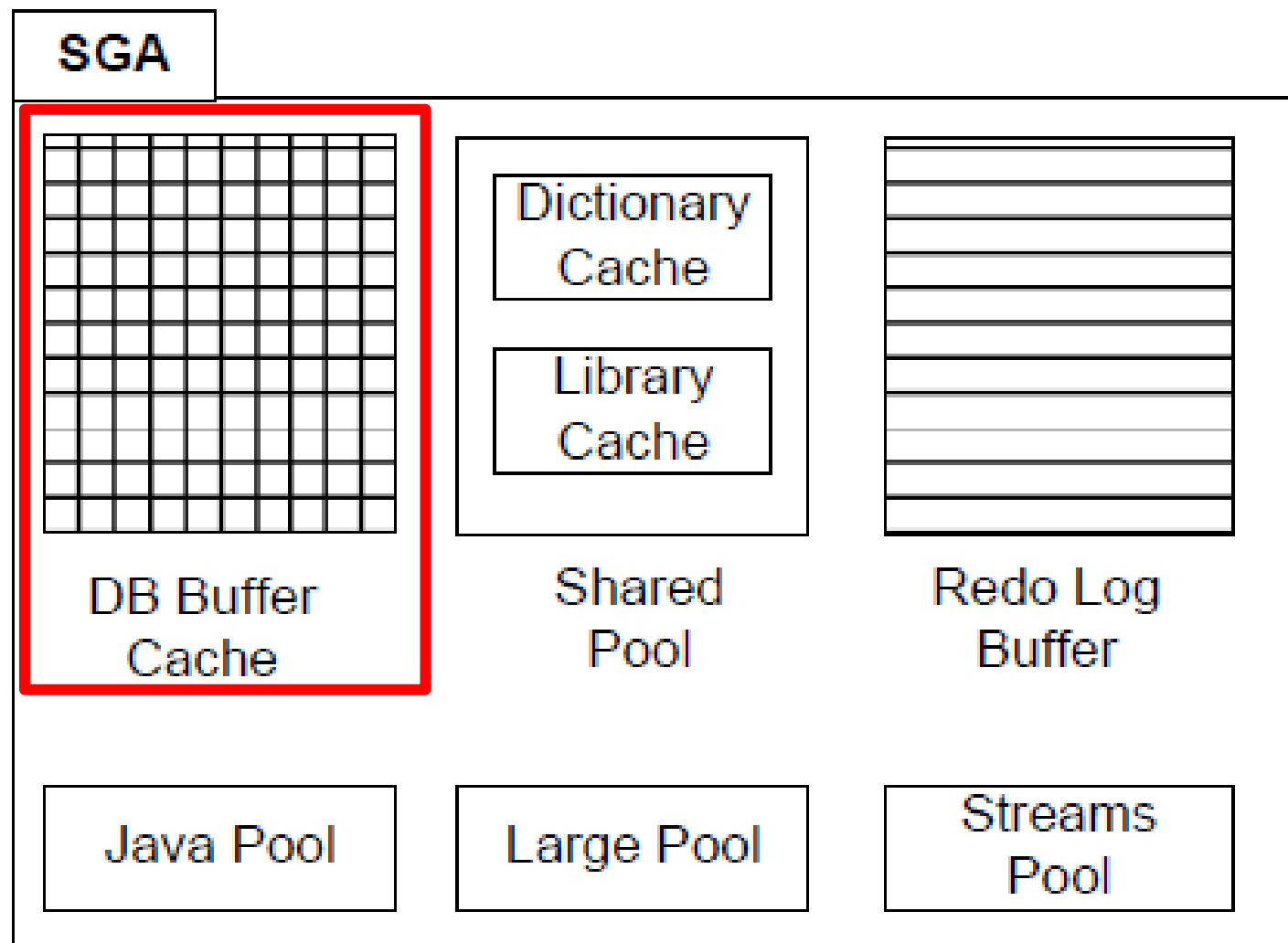


L'instance: les structures mémoires



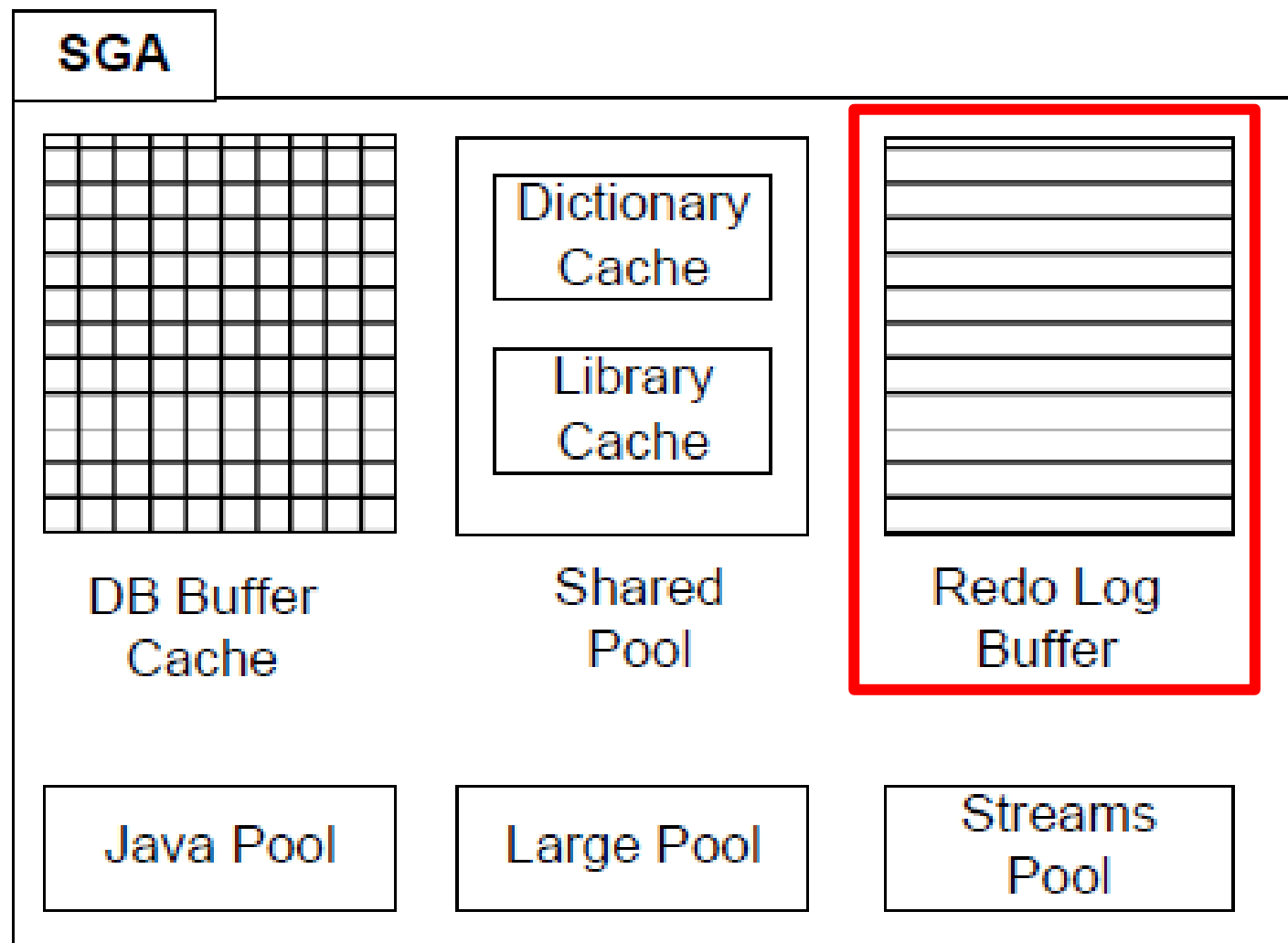
- La taille est fixé par: `SHARED_POOL_SIZE`.
- Le **Library Cache** contient les informations sur les requêtes SQL les plus récemment utilisées (*son texte, sa compilation, son plan d'exécution*).
- Oracle ne perd pas son temps à ré-exécuter une requête qui existe dans le LC.
- Le **Dictionary Cache** contient toutes les informations nécessaires pour l'analyse de la requête tel que l'existence de la table, des colonnes, le droit d'accès etc.

L'instance: les structures mémoire



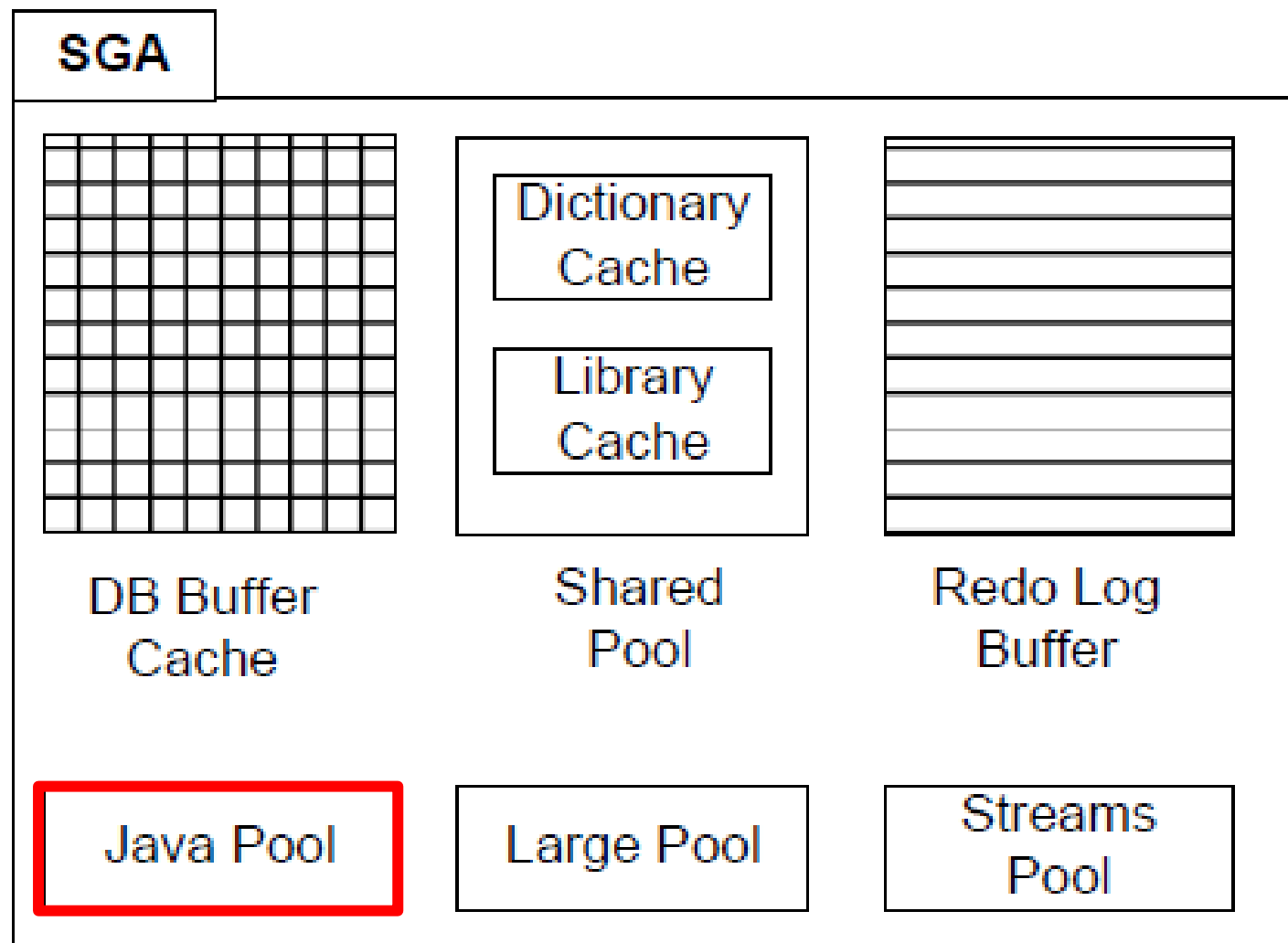
- La taille est fixé par: `DB_CACHE_SIZE`.
- Le **DB Buffer Cache** contient les blocs de données les plus récemment utilisés.
- Lorsqu'une requête est exécutée, Oracle vérifie d'abord si les blocs de données à ramener ne soient pas déjà chargés dans le DB buffer.
- Si les blocs de données n'existent pas, Oracle les y stockent.

L'instance: les structures mémoires



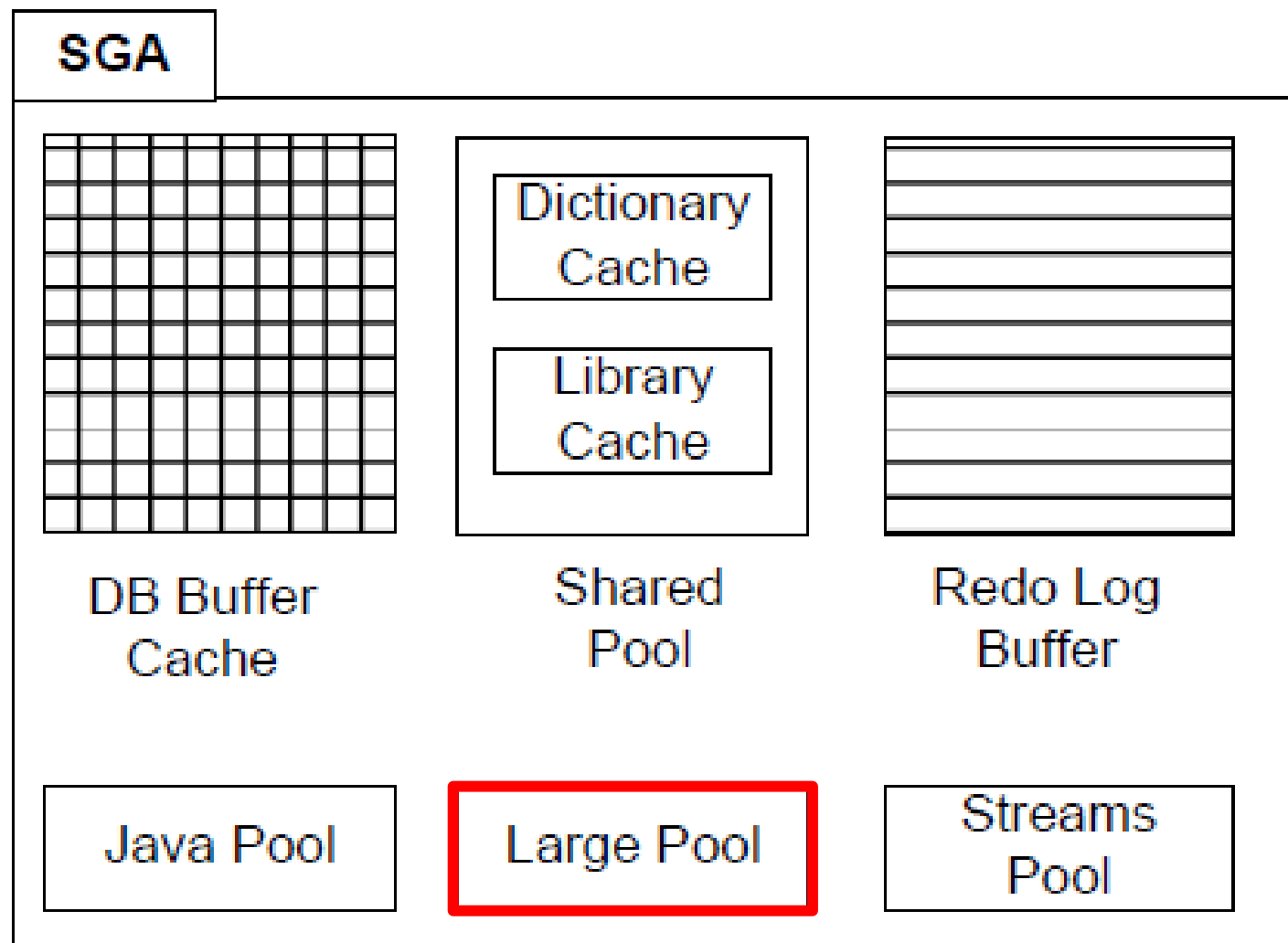
- La taille est fixé par: LOG_BUFFER.
- Le **Redo Log Buffer** contient les mises à jour effectuées sur les données, et donc toutes les informations relatives à toute transaction réalisée par les utilisateurs.
- Le contenu du buffer est écrit par Oracle dans les fichiers de journalisation (Redo Log Files) pour garantir la récupération des données en cas de crash du système.

L'instance: les structures mémoires



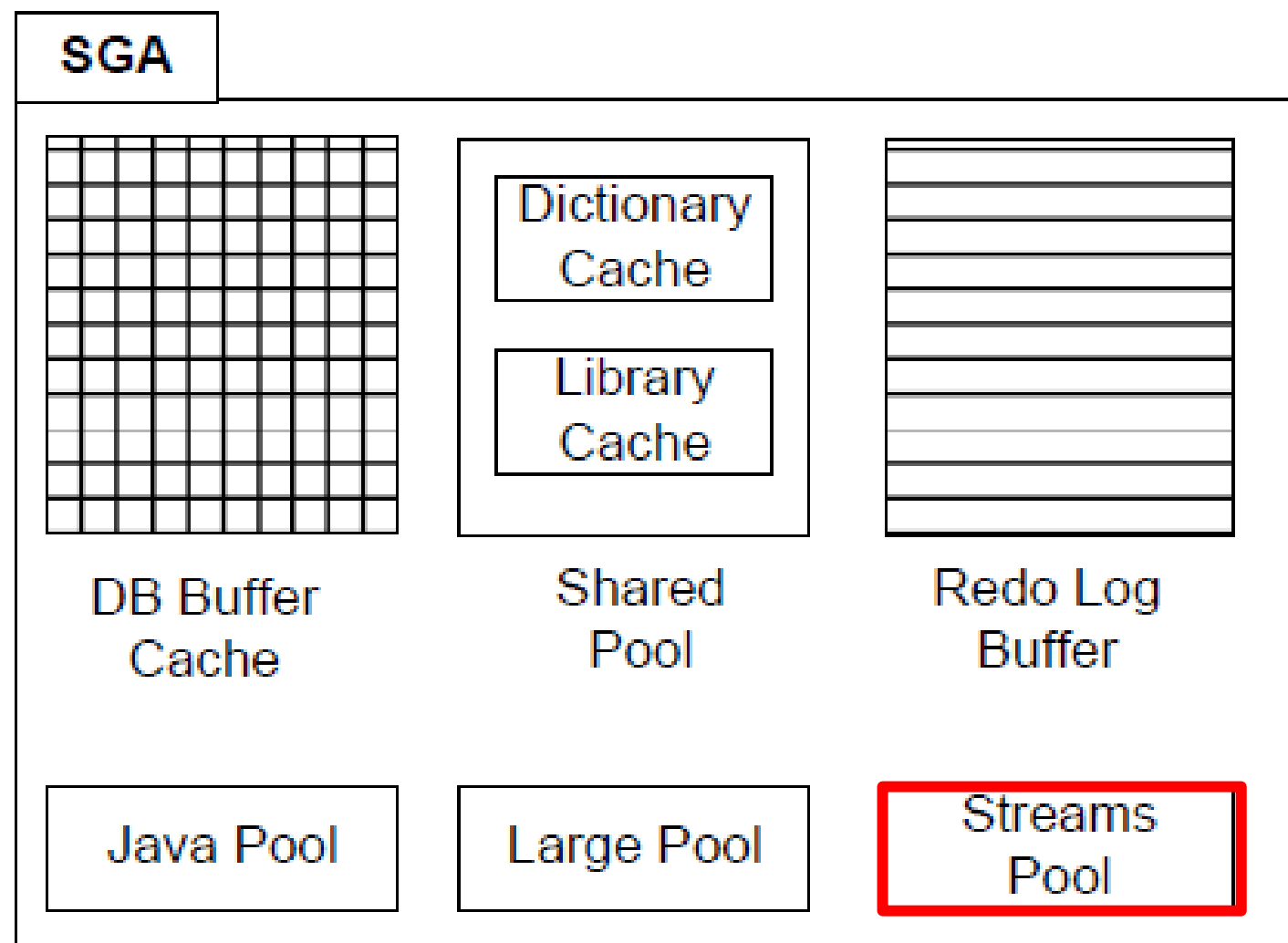
- La taille est fixé par: `JAVA_POOL_SIZE`.
- Le **Java Pool** stocke les objets et applications Java les plus récemment utilisés lorsque la machine virtuelle Java JVM optionnelle est utilisée.

L'instance: les structures mémoires



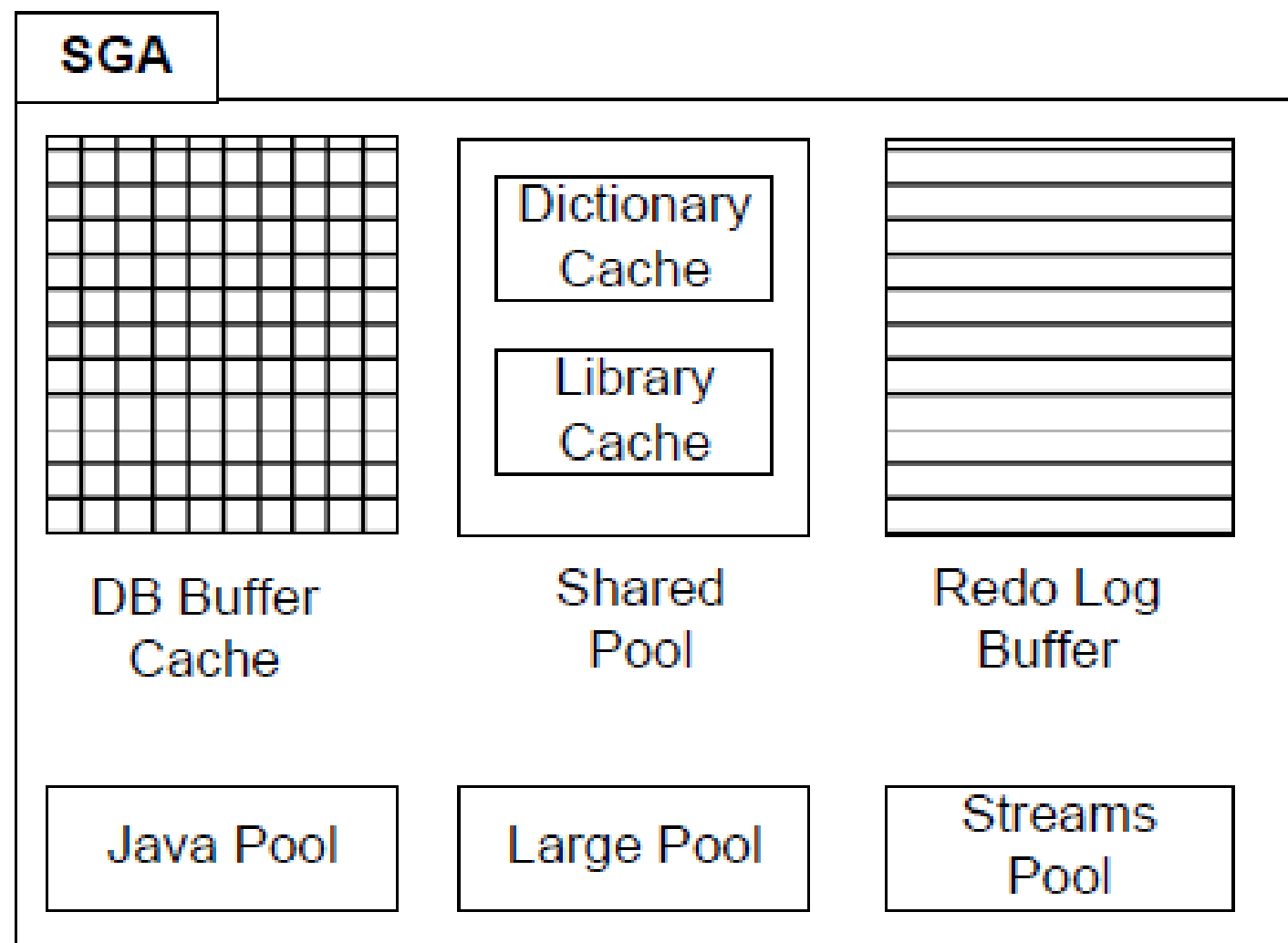
- La taille est fixé par: `LARGE_POOL_SIZE`.
- Le **Large Pool** stocke des données lors de l'exécution d'opérations volumineuses. Il est utilisé dans le cas d'une configuration serveur partagé.

L'instance: les structures mémoires



- La taille est fixé par: STREAMS_POOL_SIZE.
- Le **Streams Pool** est le cache des données relatives à la queue de messages utilisées par Oracle .

L'instance: les structures mémoires



- Une petite zone mémoire (de l'ordre de centaines de Ko) appelée **SGA fixe** existe et inclut des informations sur l'état de la base de données, sur l'instance et sur les verrous.
- Les vues sont **V\$SGA** et **V\$SGA_DYNAMIC_COMPONENTS**

L'instance: les structures mémoires

```
SQL> SELECT * FROM V$SGA;
```

NAME	VALUE
Fixed Size	1289996
Variable Size	218104052
Database Buffers	583008256
Redo Buffers	2904064

Affiche les tailles de zones mémoire composant la SGA:

SGA Fixe, SGA variable, Cache des données et celle des journaux

L'instance: les structures mémoires

■ Program Global Area (PGA):

- Une zone mémoire **privée** dédiée aux utilisateurs.
- Elle est créée pour chaque **processus serveur** quand il est initié.
- Elle stocke les informations spécifiques aux utilisateurs telles que les informations de session etc.
- La PGA est exclusivement accédée par le processus serveur.
- La PGA totale allouée à tous les processus serveurs est appelée **PGA agrégée**.
- Sa taille est dimensionnée par le paramètre **PGA_AGGREGATE_TARGET** et c'est Oracle qui se charge de répartir cette mémoire entre les différents processus serveurs.

L'instance: les processus

- Les processus Oracle permettent aux différentes composantes du serveur d'interagir et d'échanger entre elles ainsi qu'avec les utilisateurs.

- Les types de processus sont:
 - **Processus utilisateur**
 - **Processus serveur**
 - **Processus d'arrière plan**

L'instance: les processus

➤ **Processus utilisateur:**

- s'exécute au niveau client.
- Interprète et exécute les requêtes SQL.

➤ **Processus serveur:**

- s'exécute au niveau serveur BD.
- Il reçoit les requêtes utilisateur, les exécute et renvoie le résultat.

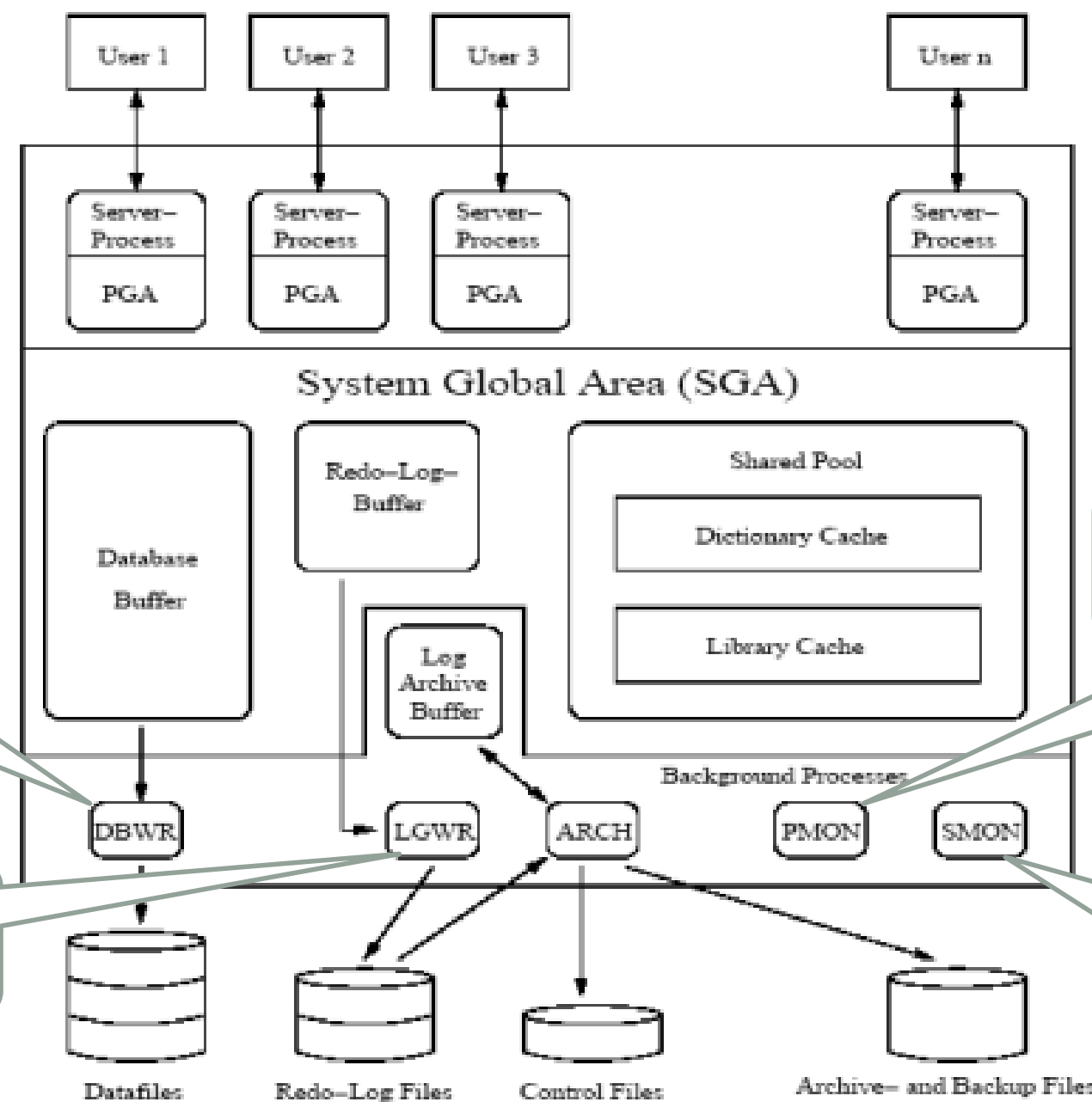
➤ **Processus d'arrière plan:**

- Assurent le bon fonctionnement du serveur.
- Maximisent la performance du serveur.
- Démarrent avec ou après (sur demande) le démarrage de l'instance.
- Certains peuvent être exécutés en n exemplaires.
- Exp: LGWR (écrit le contenu du redo log buffer sur les fichiers de journalisation de manière séquentielle) et SMON (récupère les données au démarrage de l'instance si cette dernière s'est arrêtée de manière anormale).

L'instance: les processus

- **Connexion:** un client est lié au serveur.
- **Session:** un client s'identifie.
- Plusieurs sessions peuvent être ouvertes en même temps. Le nombre est limité à la valeur du paramètre SESSIONS.

Conclusion



écrit le contenu des buffers dans les fichiers de données

écrit le contenu du redo log buffer sur les fichiers de journalisation de manière séquentielle

nettoie les ressources, les verrous et les processus utilisateurs non utilisées

récupère les données au démarrage de l'instance si cette dernière s'est arrêtée de manière anormale