



COMPRENDRE LES DIFFERENTS TYPES DE CONNEXION LORS DE LA DEFINITION D'UNE BIBLIOTHEQUE D'ACCES A UNE BASE DE DONNEES

Cet article s'adresse aux administrateurs SAS en charge de la mise à disposition des données au sein du référentiel SAS et du paramétrage de la connexion aux bases de données externes à SAS (telles que Oracle, DB2, Teradata...).



Caractéristiques :

Catégories : SAS/Access
OS : Windows, Unix
Version : SAS® 9.3 M2
Vérifié en juin 2013

Une bonne compréhension des options positionnables dans la définition des bibliothèques SAS utilisant un moteur spécifique est primordiale.

Nous allons, dans cet article, nous focaliser sur les types de connexion à la base de données, quel que soit le SGBD. Ce paramétrage a un impact direct sur le nombre de connexions et de sessions créées lors de l'interaction avec la base de données. En effet, lorsque vous créez plusieurs bibliothèques sur plusieurs schémas d'une même base de données, vous pouvez être confronté à des problèmes d'accès à cette base de données. Mais le nombre de sessions autorisées sur une base est souvent limité : si cette limite est atteinte, plus aucun utilisateur ne peut se connecter à la base.

Il peut donc être nécessaire de connaître le nombre de sessions et de connexions effectuées à la base à chaque création de bibliothèque ou accès aux tables de cette bibliothèque. En effet, le nombre de connexions et de sessions dépend des manipulations effectuées dans SAS mais surtout des options des définitions de bibliothèques dans SAS Management Console.

Cet article passe en revue l'impact du choix du type de connexion sur le nombre de sessions et de connexions.

Table des matières

1. Qu'est-ce qu'une CONNEXION, une SESSION et un PROCESSUS ?.....	2
Combien y a-t-il de SESSIONS par CONNEXION ?	3
Comment tracer les connexions et les sessions ?	3
2. Comment modifier les options des BIBLIOTHEQUES dans SAS ?	4
3. Type de connexion et impacts sur les CONNEXIONS/SESSIONS	5
SHAREDREAD (mode par défaut)	5
GLOBALREAD	11
UNIQUE	13
GLOBAL	17
4. Exécution de code SAS sans consultation préalable des bibliothèques ORACLE	19
SHAREREAD :	19
5. Autres options	20
6. Conclusion	20
7. Boîte à outils	21
Connaître le nombre de sessions versus le nombre de sessions maximum	21
Connaître le nombre de sessions par utilisateur	21
Lister les sessions	21
8. Liens utiles	22

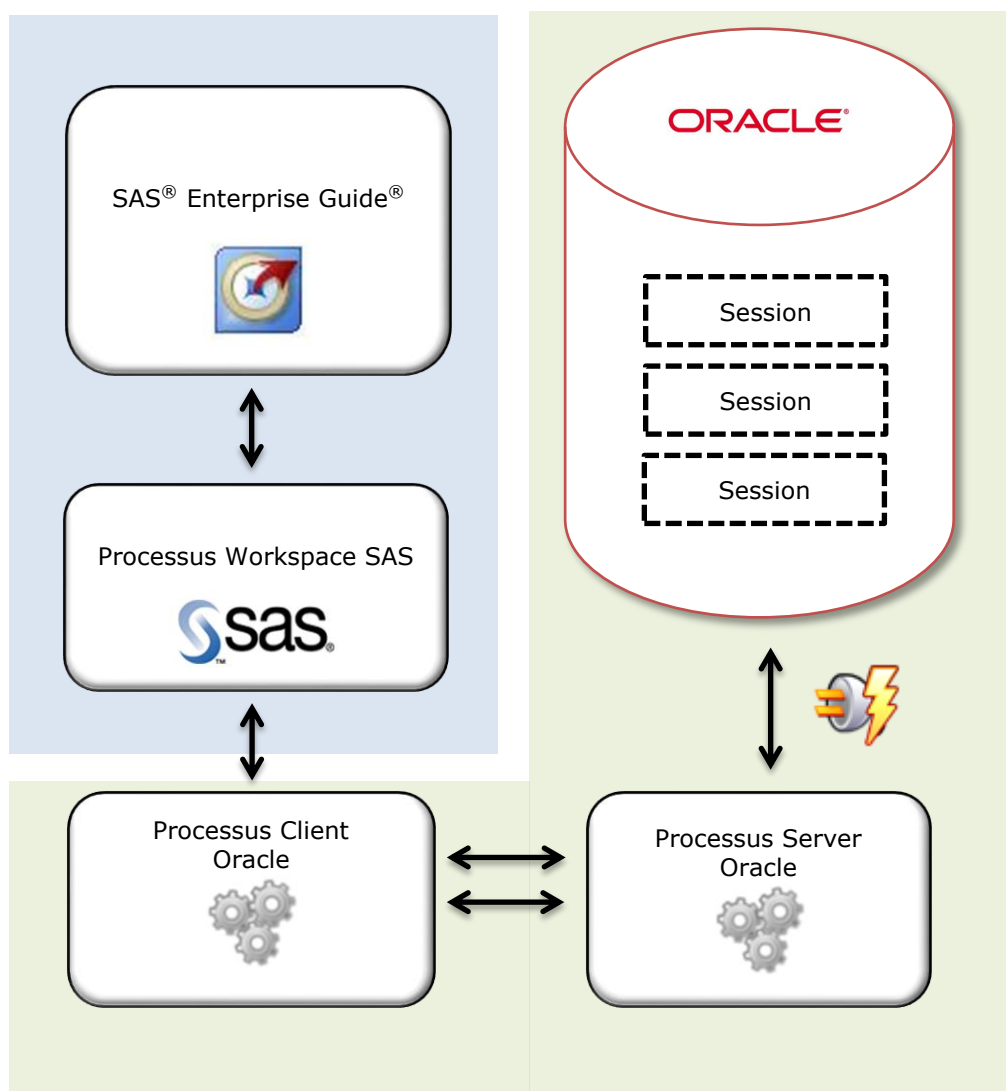
1. Qu'est-ce qu'une CONNEXION, une SESSION et un PROCESSUS ?

Par l'intermédiaire d'un client SAS, les utilisateurs peuvent accéder à un SGBD (Oracle, MySQL, DB2, Teradata ...), sous condition d'avoir le module « SAS/Access Interface to » correspondant. Dans le cadre de cet article, nous utilisons SAS Enterprise Guide comme client, et Oracle comme base de données. Cette connexion entre SAS Enterprise Guide et le serveur Oracle permet la récupération de données via le code SAS Base, retranscrit en requêtes SQL par SAS. Ces requêtes SQL, envoyées par SAS à la base de données, sont souvent transparentes pour l'utilisateur.

Lorsque SAS Enterprise Guide initie une connexion, un « lien » est alors créé entre le workspace server et le serveur Oracle. Grâce à ce lien, le **processus CLIENT** communique avec Oracle via un **processus SERVEUR Oracle**. Le client tourne sur le poste du client et le **processus SERVEUR** tourne sur le serveur.

C'est ce **processus SERVEUR** qui exécute les requêtes SQL soumises par le client. Sous Oracle par exemple, ce mécanisme s'appelle une « *Dedicated Server connection* ». Une configuration alternative consiste à permettre à plusieurs processus CLIENT de partager un seul processus SERVER (« *Shared Server connection* »).

Lorsqu'un utilisateur est connecté à un serveur de base de données, on parle de **SESSION**. Une **SESSION** démarre lorsque le serveur valide la **CONNEXION** de l'utilisateur (c'est-à-dire que l'utilisateur est correctement authentifié). Cette **SESSION** se termine lorsque l'utilisateur se déconnecte ou lorsque le lien entre le client et le serveur est rompu (problème réseau ou incident sur le poste utilisateur par exemple).

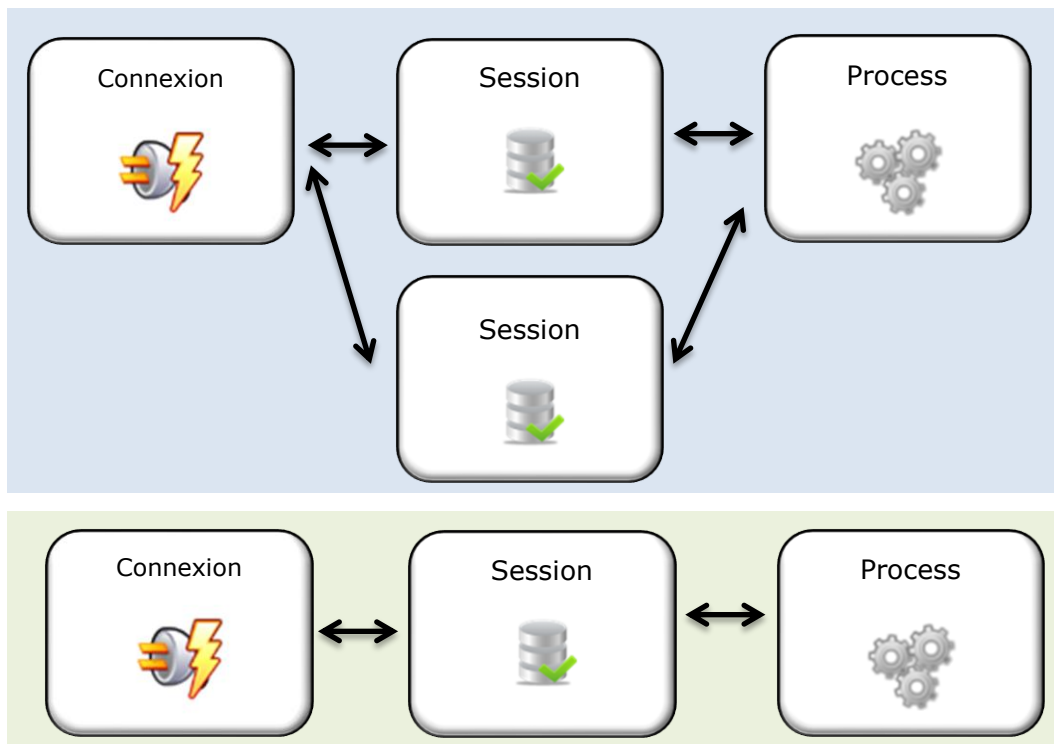


Combien y a-t-il de SESSIONS par CONNEXION ?

A partir d'une seule CONNEXION, il peut y avoir aucune, une ou plusieurs SESSIONS.

Chaque SESSION va créer un PROCESS. C'est ce PROCESS qui exécute les requêtes SQL. Parfois, il est possible d'avoir une connexion, deux sessions mais un seul process.

Enfin, plusieurs SESSIONS peuvent être créées pour un même utilisateur de base de données. Par exemple, les utilisateurs peuvent créer autant de connexions simultanées avec le même utilisateur SCOTT/TIGER, sur la même instance de base de données.



Comment tracer les connexions et les sessions ?

Vous pourriez être amené à devoir tracer les connexions à la base de données ainsi que les sessions créées ou existantes. Plusieurs méthodes peuvent être mises en œuvre.

Toutes les connexions sont tracées dans le fichier de log du SGBD (Pour Oracle, ce fichier se trouve dans le répertoire `/ORACLE_HOME/network/log/listener.log`). En effet, sur votre serveur vous avez un « LISTENER » qui écoute les demandes de connexion à la base ORACLE et qui en conserve une trace. Néanmoins, nous n'avons dans ce fichier que les demandes et donc très peu d'informations au sujet des connexions et sessions.

En étant administrateur de la base de données, vous avez accès à des tables systèmes donnant des informations plus précises.

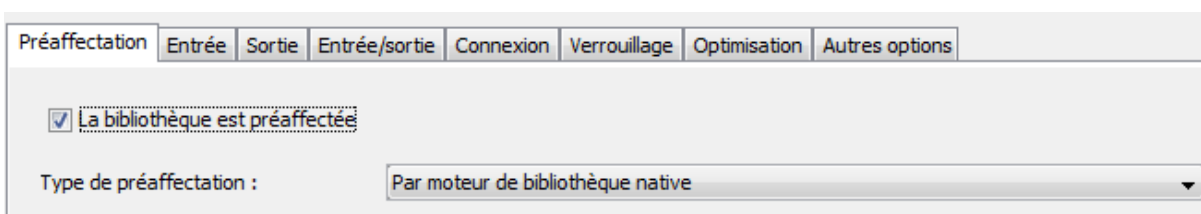
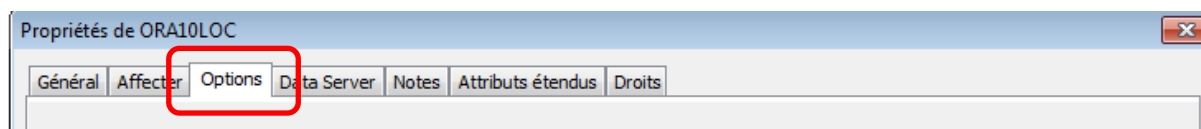
Les tables `DBA_AUDIT_SESSION`, `V$SESSION`, `V$DISPATCHER`, `V$CIRCUIT`, `V$SHARED_SERVER` et `V$OPEN_CURSOR` fournissent des informations détaillées sur les connexions et sessions.

Vous trouverez à la fin de cet article une « boîte à outils » listant des requêtes SQL utilisées pour suivre les connexions et sessions. Ces requêtes sont celles utilisées dans les exemples de cet article.

2. Comment modifier les options des BIBLIOTHÈQUES dans SAS ?

Revenons maintenant dans SAS. Pour être accessible depuis SAS Enterprise Guide, les bibliothèques – dont celles pointant sur une base de données - peuvent être définies via la SAS Management Console. Nous allons voir qu'en fonction des options choisies, le comportement peut ensuite être radicalement différent.

Dans la SAS Management Console, pour accéder à ces options, il faut éditer les propriétés de la bibliothèque, aller dans l'onglet « Options », puis cliquer sur « Options avancées ». La fenêtre se présente alors comme ceci :



3. Type de connexion et impacts sur les CONNEXIONS/SESSIONS

Afin d'éviter l'erreur oracle « *ORA-00018 maximum number of sessions exceeded* », il est possible de jouer sur certains paramètres afin de limiter le nombre de connexions et de créations de sessions. L'idée est de créer des sessions uniquement si l'utilisateur en a la nécessité.

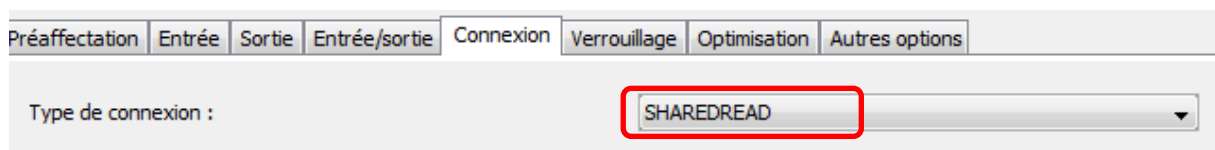
Nous allons tester les différents modes : SHAREDREAD, GLOBALREAD, UNIQUE, SHARED, GLOBAL.

Pour réaliser ces tests, nous avons utilisé les versions suivantes :

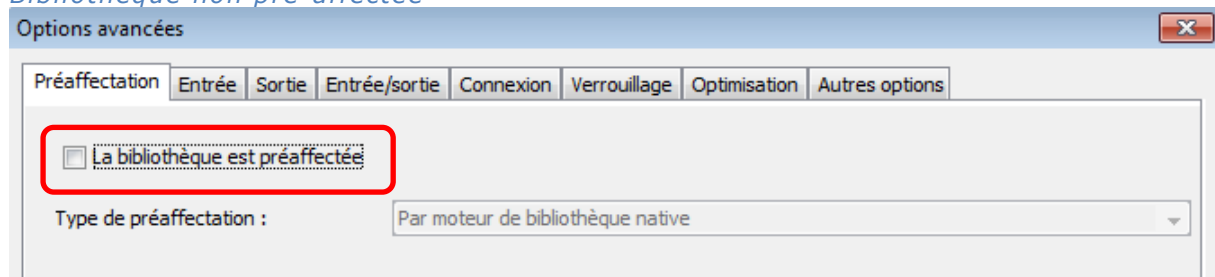
- SAS Enterprise Guide 5.1
- SAS 9.3M2
- Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - 64bit

SHAREDREAD (mode par défaut)

Le mode SHAREDREAD est le mode défini par défaut lors de la création d'une bibliothèque.



Bibliothèque non pré-affectée

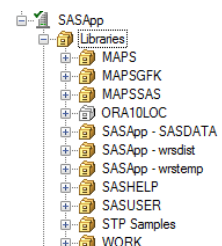


Dans ce premier exemple nous allons simplement naviguer dans SAS Enterprise Guide, sans soumettre de code SAS.

- Une seule bibliothèque Oracle est définie dans le référentiel.
- Cette bibliothèque, ORA10LOC, n'est pas pré-affectée à l'activation du Workspace server.
- Ses autres options sont laissées aux valeurs par défaut.

Cet exemple nous permet de comprendre de façon simple le mécanisme de CONNEXION/DECONNEXION et de création de SESSION lorsqu'une base de données Oracle est interrogée à partir de SAS Enterprise Guide.

Après avoir lancé SAS Enterprise Guide, puis en double-cliquant sur le serveur d'application SASApp, un Workspace Server est démarré et la bibliothèque Oracle ORA10LOC est dépliée.



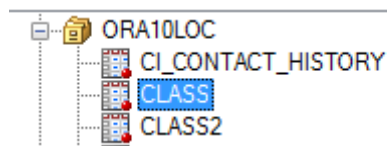
Nous avons alors une CONNEXION/DECONNEXION à la base de données au moment de l'affectation des bibliothèques.

Cependant il n'y **AUCUNE** création de SESSION.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 11:09:52	09/01/2013 11:09:51	651464	27	0	10

Ouvrons maintenant une table de cette bibliothèque ORA10LOC.



Nous avons alors deux CONNEXIONS à la base de données :

- La première CONNEXION correspond à l'activation de la bibliothèque.
- La deuxième à l'affichage de la table CLASS.

CLASS						
	NAME	SEX	AGE	HEIGHT	WEIGHT	
1	Alfred	M	14	69	112.5	
2	Alice	F	13	56.5	84	
3	Barbara	F	13	65.3	98	
4	Carol	F	14	62.8	102.5	
5	Henry	M	14	63.5	102.5	
6	James	M	12	57.3	83	
7	Jane	F	12	59.8	84.5	
8	Janet	F	15	62.5	112.5	
9	Jeffrey	M	13	62.5	84	
10	John	M	12	59	99.5	
11	Joyce	F	11	51.3	50.5	
12	Judy	F	14	64.3	90	
13	Louise	F	12	56.3	77	
14	Mary	F	15	66.5	112	
15	Philip	M	16	72	150	
16	Robert	M	12	64.8	128	
17	Ronald	M	15	67	133	
18	Thomas	M	11	57.5	85	
19	William	M	15	66.5	112	

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 11:09:52	09/01/2013 11:09:51	651464	27	0	10
SCOTT	09/01/2013 11:10:28	09/01/2013 11:10:28	651480	23	0	6
SCOTT	09/01/2013 11:10:29		458888			

Nous avons maintenant une session INACTIVE à l'état WAITING :

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	146	458888	INACTIVE	WAITING	40	2013/01/09 11:10:28	40

Si aucune nouvelle action n'est opérée sur les tables, alors la session reste INACTIVE jusqu'à la fermeture de SAS Enterprise Guide.

On peut également constater que les indicateurs LAST_CALL et SECONDS_IN_WAIT (nombre de secondes depuis le dernier accès à la table) évoluent avec le temps :

STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
WAITING	6	2013/01/09 11:10:28	6

Si une seconde table de cette bibliothèque ORA10LOC est ouverte :

- Il n'y a pas de nouvelle CONNEXION faite à la base de données.
- La session utilisée est la même.
- L'indicateur LAST_CALL est réinitialisé à 0.

Enfin, à la fermeture de SAS Enterprise Guide :

- La dernière connexion ouverte est fermée :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 11:09:52	09/01/2013 11:09:51	651464	27	0	10
SCOTT	09/01/2013 11:10:28	09/01/2013 11:10:28	651480	23	0	6
SCOTT	09/01/2013 11:10:29	09/01/2013 11:28:10	458888	54	0	6

- La session est supprimée.

Dans ce premier exemple, nous constatons que l'utilisation d'une bibliothèque non-pré-affectée au démarrage du WORKSPACE n'est pas coûteux en termes de connexions et de sessions.



En effet, une connexion est créée uniquement à l'ouverture d'une table de la bibliothèque et qu'une seule session est utilisée pour accéder à l'ensemble des tables d'une même bibliothèque.

Bibliothèque pré-affectée

Effectuons maintenant quelques tests mais cette fois, la bibliothèque OR10LOC est pré-affectée au démarrage du Workspace server.

Préaffectation	Entrée	Sortie	Entrée/sortie	Connexion	Verrouillage	Optimisation	Autres options
<input checked="" type="checkbox"/> La bibliothèque est préaffectée							

Dans SAS Enterprise Guide, un code simple est exécuté à l'ouverture de l'application :

```
proc print data=ORA10LOC.class;  
run;
```

L'exécution de ce programme génère une CONNEXION et une SESSION :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 14:50:27		663646			

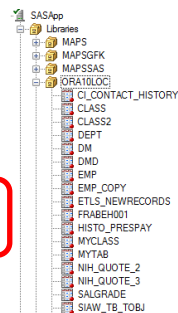
SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	142	663646	INACTIVE	WAITING	117	2013/01/09 14:50:27	117

Si un autre programme est exécuté sur cette même table, la même SESSION sera toujours utilisée.

Maintenant, en dépliant la bibliothèque OR10LOC dans l'explorateur :

Une seconde CONNEXION et une nouvelle SESSION sont créées au moment de l'ouverture de la bibliothèque.



CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 14:50:27		663646			
SCOTT	09/01/2013 14:53:14		426168			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	149	426168	INACTIVE	WAITING	20	2013/01/09 14:53:13	20
SCOTT	bi93	142	663646	INACTIVE	WAITING	183	2013/01/09 14:50:27	183

Nous nous retrouvons maintenant avec 2 SESSIONS pour le même utilisateur.

Si nous analysons les requêtes exécutées par chaque SESSION, nous pouvons visualiser celles associées à l'ouverture de la bibliothèque Oracle pré-affectée (session 149) :

REQUETE(S) :

SID	LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
142	09/01/2013 14:50:30	280	SELECT * FROM CLASS
149	09/01/2013 14:53:13	7251	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))

Maintenant, si nous exécutons un programme de CREATION DE TABLE dans cette bibliothèque Oracle OR10LOCHES, une nouvelle CONNEXION est instanciée ainsi qu'une nouvelle SESSION :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 14:50:27		663646			
SCOTT	09/01/2013 14:53:14		426168			
SCOTT	09/01/2013 14:59:17		458990			

SESSIONS(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	149	426168	INACTIVE	WAITING	16	2013/01/09 14:53:13	16
SCOTT	bi93	148	458990	ACTIVE	WAITED SHORT TIME	0	2013/01/09 14:59:17	0
SCOTT	bi93	142	663646	INACTIVE	WAITING	541	2013/01/09 14:50:27	541

REQUETE(S) :

LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
09/01/2013 15:00:01	1000000	INSERT INTO TEST1100 (CLE_PRESPAYE) VALUES (:CLE_PRESPAYE)

Cette nouvelle SESSION est ACTIVE durant l'exécution du programme. Cette SESSION est ensuite supprimée.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 14:50:27		663646			
SCOTT	09/01/2013 14:53:14		426168			
SCOTT	09/01/2013 14:59:17	09/01/2013 15:00:01	458990	162504	0	239560

A la déconnexion de l'utilisateur ou à la fermeture de SAS Enterprise Guide, les SESSIONS sont détruites et les connexions encore ouvertes, sont fermées.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 14:50:27	09/01/2013 15:06:26	663646	203	0	7
SCOTT	09/01/2013 14:53:14	09/01/2013 15:06:26	426168	1335	0	6
SCOTT	09/01/2013 14:59:17	09/01/2013 15:00:01	458990	162504	0	239560

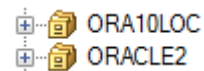


Comme lorsque la bibliothèque n'est pas pré-affectée, nous avons une connexion et une session créées lorsque nous accédons pour la première fois à l'une des tables de cette bibliothèque.

Ce second exemple montre également qu'une nouvelle session est nécessaire si l'on souhaite mettre à jour une table d'une bibliothèque déjà ouverte en lecture.

Deux bibliothèques sur la même base

Partons maintenant du principe que deux bibliothèques ORACLE pré-affectées sont définies dans notre environnement.



Ces deux bibliothèques pointent sur la même base ORACLE.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 16:05:16		663606			
SCOTT	09/01/2013 16:05:16		442620			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	149	442620	INACTIVE	WAITING	66	2013/01/09 16:05:16	66
SCOTT	bi93	142	663606	INACTIVE	WAITING	66	2013/01/09 16:05:16	66

En dépliant les deux bibliothèques dans SAS Enterprise Guide, donc en listant simplement les tables de cette bibliothèque, on obtient deux nouvelles CONNEXIONS :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 16:05:16		663606			
SCOTT	09/01/2013 16:05:16		442620			
SCOTT	09/01/2013 16:08:05		458986			
SCOTT	09/01/2013 16:08:17		413880			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	148	413880	INACTIVE	WAITING	47	2013/01/09 16:08:17	47
SCOTT	bi93	149	442620	INACTIVE	WAITING	228	2013/01/09 16:05:16	228
SCOTT	bi93	141	458986	INACTIVE	WAITING	59	2013/01/09 16:08:05	59
SCOTT	bi93	142	663606	INACTIVE	WAITING	228	2013/01/09 16:05:16	228

REQUETE(S) :

SID	LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
141	09/01/2013 16:08:17	7461	SELECT OBJECT_NAME ,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
148	09/01/2013 16:08:17	7461	SELECT OBJECT_NAME ,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))

En ouvrant une table de la bibliothèque ORACLE2, on peut noter que la SESSION 149 est utilisée :

REQUETE(S) :

SID	LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
141	09/01/2013 16:13:31	7531	SELECT OBJECT_NAME ,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
148	09/01/2013 16:13:31	7531	SELECT OBJECT_NAME ,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
149	09/01/2013 16:13:34	300	SELECT * FROM CLASS

Si nous ouvrons une table de la bibliothèque ORACLE2, la SESSION 142 est utilisée :

REQUETE(S) :

SID	LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
141	09/01/2013 16:16:01	7601	SELECT OBJECT_NAME ,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
142	09/01/2013 16:16:01	320	SELECT * FROM CLASS
148	09/01/2013 16:16:01	7601	SELECT OBJECT_NAME ,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
149	09/01/2013 16:16:01	320	SELECT * FROM CLASS

Il faut donc être prudent si vous multipliez les bibliothèques pré-affectées sur la même base de données.

Avec, par exemple, 4 bibliothèques pré-affectées et dans le cas où les utilisateurs navigueraient dans certaines, le nombre de sessions augmenterait rapidement, même si les utilisateurs ne font aucune opération sur les tables contenues dans les bibliothèques :

SESSION(S) :

	USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
▶	SCOTT	bi93	148	413882	INACTIVE	WAITING	242	2013/01/09 16:21:38	242
	SCOTT	bi93	142	442622	INACTIVE	WAITING	195	2013/01/09 16:21:38	195
	SCOTT	bi93	149	454810	INACTIVE	WAITING	204	2013/01/09 16:22:16	204
	SCOTT	bi93	141	458988	INACTIVE	WAITING	198	2013/01/09 16:21:38	198
	SCOTT	bi93	144	630982	INACTIVE	WAITING	195	2013/01/09 16:22:18	195
	SCOTT	bi93	139	647292	INACTIVE	WAITING	198	2013/01/09 16:22:20	198

REQUETE(S) :

	SID	LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
▶	139	09/01/2013 16:22:25	7881	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
	141	09/01/2013 16:22:22	3002	SELECT * FROM TEST1100
	142	09/01/2013 16:22:25	8506	SELECT * FROM CL_CONTACT_HISTORY
	144	09/01/2013 16:22:25	7881	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
	149	09/01/2013 16:22:16	0	SELECT OBJECT_NAME,OBJECT_TYPE FROM ALL_OBJECTS OBJ WHERE (OBJ.OWNER ='ORABIS') AND (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))



Si sur votre environnement, SAS Enterprise Guide est utilisé par de nombreux utilisateurs simultanément, le nombre de sessions peut augmenter rapidement, voire même atteindre les limites autorisées sur la base de données.

Ces nombreux accès à la base de données, qui peuvent être inutiles, ont un coût et peuvent pénaliser les performances de la base.



Lorsque vous utilisez le mode SHAREDREAD, chaque table d'une même bibliothèque partage la même CONNEXION/SESSION. Ce comportement est vrai lors de la **lecture** d'une table.

Une connexion à part est établie lorsque que vous mettez à jour une table.

Enfin, si votre environnement contient plusieurs bibliothèques, autant de CONNEXION/SESSION seront créées sur votre base de données.

GLOBALREAD

Maintenant que nous avons pu constater le fonctionnement lorsque le mode SHAREDREAD est sélectionné, nous allons réaliser une série d'exemples en testant chaque mode de connexion. Pour chaque mode, nous allons respecter le même protocole de tests :

1. Lancer SAS Entreprise Guide
2. Activer un Workspace Server
3. Lister les tables de deux bibliothèques Oracle ORA10LOC et ORACLE2
4. Ouvrir une table dans chacune des deux bibliothèques (ORA10LOC.CLASS et ORACLE2.CLASS)
5. Lister une table dans chacune des deux bibliothèques via du code SAS
6. Créer une table dans la bibliothèque ORACLE2

Préaffectation	Entrée	Sortie	Entrée/sortie	Connexion	Verrouillage	Optimisation	Autres options
Type de connexion : GLOBALREAD ▼							

En positionnant le type de connexion à GLOBALREAD :

1/ Activation du Workspace server :

CONNEXION(S) :

▶ SCOTT	09/01/2013 21:07:19		741434			
---------	---------------------	--	--------	--	--	--

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
▶ SCOTT	bi93	140	741434	INACTIVE	WAITING	91	2013/01/09 21:07:19	91

2/ Ouverture des deux bibliothèques ORA10LOC et ORACLE2 :

CONNEXION(S) :

▶ SCOTT	09/01/2013 21:07:19		741434			
SCOTT	09/01/2013 21:09:17		647178			
SCOTT	09/01/2013 21:09:34		454894			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
▶ SCOTT	bi93	141	454894	INACTIVE	WAITING	65	2013/01/09 21:09:33	65
SCOTT	bi93	139	647178	INACTIVE	WAITING	81	2013/01/09 21:09:17	81
SCOTT	bi93	140	741434	INACTIVE	WAITING	199	2013/01/09 21:07:19	199

REQUETE(S) :

▶ 139 09/01/2013 21:09:33	8021	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
141 09/01/2013 21:09:33	8021	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))

3/ Affichage du contenu des tables :

Nous affichons le contenu de la table ORA10LOC.CLASS. Nous constatons que la session utilisée est la SESSION ID 140

REQUETE(S) :

▶ 140 09/01/2013 21:12:17	340	SELECT * FROM CLASS
141 09/01/2013 21:12:14	8091	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
139 09/01/2013 21:12:14	8091	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))

Ouvrons maintenant la table ORACLE2.CLASS2.

La lecture des tables de deux bibliothèques se fait dans la même SESSION ID 140.

REQUETE(S) :

140	09/01/2013 21:14:08	133	SELECT * FROM CLASS2
140	09/01/2013 21:14:08	8161	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
139	09/01/2013 21:14:08	8161	SELECT OBJECT_NAME,OBJECT_TYPE FROM USER_OBJECTS OBJ WHERE (OBJ.OBJECT_TYPE IN ('TABLE','VIEW'))
140	09/01/2013 21:12:17	340	SELECT * FROM CLASS

4/ Lecture des tables via du code SAS :

```
proc print data=ORA10LOC.class;
run;
```

Les indicateurs LAST_CALL_ET de la session 140 sont réinitialisés et la requête est bien prise en charge par la session 140.

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	141	454894	INACTIVE	WAITING	270	2013/01/09 21:09:33	270
SCOTT	bi93	139	647178	INACTIVE	WAITING	462	2013/01/09 21:09:17	462
SCOTT	bi93	140	741434	INACTIVE	WAITING	21	2013/01/09 21:07:19	21

REQUETE(S) :

140	09/01/2013 21:19:35	360	SELECT * FROM CLASS
-----	---------------------	-----	---------------------

Donc, quelle que soit l'action, la SESSION 140 sera toujours utilisée pour exécuter les requêtes :

REQUETE(S) :

140	09/01/2013 21:23:17	0	SELECT * FROM ETLS_NEWRECORDS
140	09/01/2013 21:23:14	9757	SELECT * FROM CI_CONTACT_HISTORY
140	09/01/2013 21:23:05	0	SELECT * FROM FRABEH001

5/ Création d'une nouvelle table dans ORA10LOC :

Maintenant nous créons une table. Nous constatons la création d'une nouvelle session :

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	144	442438	ACTIVE	WAITED SHORT TIME	0	2013/01/09 21:25:46	0
SCOTT	bi93	141	454894	INACTIVE	WAITING	7	2013/01/09 21:09:33	7
SCOTT	bi93	139	647178	INACTIVE	WAITING	817	2013/01/09 21:09:17	817
SCOTT	bi93	140	741434	INACTIVE	WAITING	43	2013/01/09 21:07:19	43

CONNEXION(S) :

SCOTT	09/01/2013 21:25:46	09/01/2013 21:26:33	442438	168491	46	242758
-------	---------------------	---------------------	--------	--------	----	--------

Note : Cette SESSION est supprimée après la création de la table.



Avec le mode GLOBALREAD, nous constatons que toutes les opérations de lectures sur les tables de plusieurs bibliothèques partagent une seule connexion.

Une connexion séparée est initialisée pour les opérations de mises à jour.

UNIQUE

Préaffectation	Entrée	Sortie	Entrée/sortie	Connexion	Verrouillage	Optimisation	Autres options
Type de connexion :				UNIQUE			

1/ Activation du Workspace server :

A l'activation du Workspace server, 2 connexions et 2 sessions sont créées.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	159	577556	INACTIVE	WAITING	22	2013/01/16 23:14:10	22
SCOTT	bi93	149	680170	INACTIVE	WAITING	22	2013/01/16 23:14:10	22

2/ Ouverture des deux bibliothèques ORA10LOC et ORACLE2 :

L'ouverture de la bibliothèque ORA10LOC va occasionner la création d'une troisième session.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			
SCOTT	16/01/2013 23:14:55		696564			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	159	577556	INACTIVE	WAITING	67	2013/01/16 23:14:10	67
SCOTT	bi93	149	680170	INACTIVE	WAITING	67	2013/01/16 23:14:10	67
SCOTT	bi93	141	696564	INACTIVE	WAITING	23	2013/01/16 23:14:54	23

L'ouverture de la bibliothèque ORACLE2 va occasionner la création d'une quatrième session.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			
SCOTT	16/01/2013 23:14:55		696564			
SCOTT	16/01/2013 23:15:36		651294			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	159	577556	INACTIVE	WAITING	103	2013/01/16 23:14:10	103
SCOTT	bi93	140	651294	INACTIVE	WAITING	17	2013/01/16 23:15:36	17
SCOTT	bi93	149	680170	INACTIVE	WAITING	103	2013/01/16 23:14:10	103
SCOTT	bi93	141	696564	INACTIVE	WAITING	59	2013/01/16 23:14:54	59

3/ Affichage du contenu des tables :

Nous affichons le contenu de la table ORA10LOC.CLASS. Nous constatons qu'une cinquième connexion est ouverte sur la base. Nous constatons également la création d'une cinquième session.

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			
SCOTT	16/01/2013 23:14:55		696564			
SCOTT	16/01/2013 23:15:36		651294			
SCOTT	16/01/2013 23:16:23		741470			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	159	577556	INACTIVE	WAITING	148	2013/01/16 23:14:10	148
SCOTT	bi93	140	651294	INACTIVE	WAITING	62	2013/01/16 23:15:36	62
SCOTT	bi93	149	680170	INACTIVE	WAITING	148	2013/01/16 23:14:10	148
SCOTT	bi93	141	696564	INACTIVE	WAITING	18	2013/01/16 23:14:54	18
SCOTT	bi93	151	741470	INACTIVE	WAITING	15	2013/01/16 23:16:23	15

Ouvrons maintenant la table ORACLE2.CLASS2 :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			
SCOTT	16/01/2013 23:14:55		696564			
SCOTT	16/01/2013 23:15:36		651294			
SCOTT	16/01/2013 23:16:23		741470			
SCOTT	16/01/2013 23:17:04		430260			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON TIME	LAST CALL ET
SCOTT	bi93	137	430260	INACTIVE	WAITING	20	2013/01/16 23:17:03	20
SCOTT	bi93	159	577556	INACTIVE	WAITING	193	2013/01/16 23:14:10	193
SCOTT	bi93	140	651294	INACTIVE	WAITING	21	2013/01/16 23:15:36	21
SCOTT	bi93	149	680170	INACTIVE	WAITING	193	2013/01/16 23:14:10	193
SCOTT	bi93	141	696564	INACTIVE	WAITING	63	2013/01/16 23:14:54	63
SCOTT	bi93	151	741470	INACTIVE	WAITING	60	2013/01/16 23:16:23	60

Nous avons maintenant 6 connexions ouvertes sur la table et 6 sessions « INACTIVE ».

4/ Lecture des tables via du code SAS :

```
proc print data=ora10loc.CI_CONTACT_HISTORY;
run;
```

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			
SCOTT	16/01/2013 23:14:55		696564			
SCOTT	16/01/2013 23:15:36		651294			
SCOTT	16/01/2013 23:16:23	16/01/2013 23:18:38	741470	30	0	6
SCOTT	16/01/2013 23:17:04	16/01/2013 23:18:38	430260	30	0	6
SCOTT	16/01/2013 23:18:39		413838			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	137	413838	INACTIVE	WAITING	39	2013/01/16 23:18:39	39
SCOTT	bi93	139	577556	INACTIVE	WAITING	313	2013/01/16 23:14:10	313
SCOTT	bi93	140	651294	INACTIVE	WAITING	141	2013/01/16 23:15:36	141
SCOTT	bi93	149	680170	INACTIVE	WAITING	313	2013/01/16 23:14:10	313
SCOTT	bi93	141	696564	INACTIVE	WAITING	183	2013/01/16 23:14:54	183

Les deux connexions ayant servi à afficher les tables ORA10LOC.CLASS et ORACLE2.CLASSE2 ont été fermées.

Une nouvelle session ID 137 a été instanciée pour interroger la table ORA10LOC.CI_CONTACT_HISTORY.

La connexion est fermée à la fin de la récupération des données :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:14:10		680170			
SCOTT	16/01/2013 23:14:10		577556			
SCOTT	16/01/2013 23:14:55		696564			
SCOTT	16/01/2013 23:15:36		651294			
SCOTT	16/01/2013 23:16:23	16/01/2013 23:18:38	741470	30	0	6
SCOTT	16/01/2013 23:17:04	16/01/2013 23:18:38	430260	30	0	6
SCOTT	16/01/2013 23:18:39	16/01/2013 23:28:32	413838	10640	5935	6

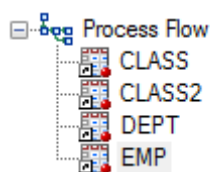
SID	LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
137	16/01/2013 23:18:44	1100953	SELECT * FROM CI_CONTACT_HISTORY

5/ Création d'une nouvelle table dans ORA10LOC :

Nous observons la création d'une connexion/session pour la création de la table.

La connexion est fermée à la fin de l'opération

Note : dans le cas de l'ouverture de nombreuses tables :

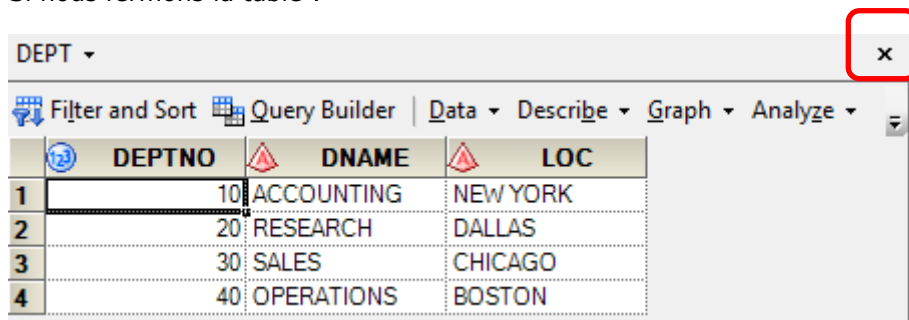


Une connexion/session est créée pour chaque table :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:38:38		413874			
SCOTT	16/01/2013 23:38:38		651300			
SCOTT	16/01/2013 23:38:45		577558			
SCOTT	16/01/2013 23:38:47		426162			
SCOTT	16/01/2013 23:38:49		401474			
SCOTT	16/01/2013 23:39:07		405556			
SCOTT	16/01/2013 23:39:12		430294			

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	137	401474	INACTIVE	WAITING	79	2013/01/16 23:38:49	79
SCOTT	bi93	151	405556	INACTIVE	WAITING	62	2013/01/16 23:39:06	62
SCOTT	bi93	141	413874	INACTIVE	WAITING	91	2013/01/16 23:38:37	91
SCOTT	bi93	159	426162	INACTIVE	WAITING	81	2013/01/16 23:38:47	81
SCOTT	bi93	138	430294	INACTIVE	WAITING	57	2013/01/16 23:39:11	57
SCOTT	bi93	140	577558	INACTIVE	WAITING	84	2013/01/16 23:38:44	84
SCOTT	bi93	149	651300	INACTIVE	WAITING	91	2013/01/16 23:38:37	91

Si nous fermons la table :



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

La session et la connexion associée sont fermées :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	16/01/2013 23:38:38		413874			
SCOTT	16/01/2013 23:38:38		651300			
SCOTT	16/01/2013 23:38:45		577558			
SCOTT	16/01/2013 23:38:47		426162			
SCOTT	16/01/2013 23:38:49		401474			
SCOTT	16/01/2013 23:39:07	16/01/2013 23:41:49	405556	30	0	6
SCOTT	16/01/2013 23:39:12	16/01/2013 23:40:27	430294	30	0	6



Avec le mode UNIQUE, chaque table utilise sa propre connexion/session.

GLOBAL

Préaffectation	Entrée	Sortie	Entrée/sortie	Connexion	Verrouillage	Optimisation	Autres options
Type de connexion :				GLOBAL			

1/ Activation du Workspace server :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME+4/24	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	12/01/2013 23:35:42		438320			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	143	438320	INACTIVE	WAITING	65	2013/01/06 23:35:41	65

2/ Ouverture des deux libraires ORA10LOC et ORACLE2 :

CONNEXION(S) :

USERNAME	TIMESTAMP	LOGOFF_TIME+4/24	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	12/01/2013 23:35:42		438320			
SCOTT	12/01/2013 23:37:17		741398			
SCOTT	12/01/2013 23:37:19		532588			

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	143	438320	INACTIVE	WAITING	137	2013/01/06 23:35:41	137
SCOTT	bi93	145	532588	INACTIVE	WAITING	39	2013/01/06 23:37:19	39
SCOTT	bi93	147	741398	INACTIVE	WAITING	42	2013/01/06 23:37:16	42

En lecture, le comportement est similaire à celui observé lorsque le mode GLOBALREAD est sélectionné.

Nous pouvons passer directement au test :

3/ Création d'une nouvelle table dans ORA10LOC :

Lors de la création de la nouvelle table, et contrairement au mode GLOBALREAD, il n'y a aucune création de nouvelle SESSION pour les opérations de mises à jour :

SESSION(S) :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	143	438320	ACTIVE	WAITED SHORT TIME	0	2013/01/06 23:35:41	0
SCOTT	bi93	145	532588	INACTIVE	WAITING	174	2013/01/06 23:37:19	174
SCOTT	bi93	147	741398	INACTIVE	WAITING	426	2013/01/06 23:37:16	426

REQUETE(S) :




LAST_ACTIVE_TIME	ROWS_PROCESSED	SQL_TEXT
12/01/2013 23:47:01	5635530	INSERT INTO TESTGLOBALREAD22 (CLE_PRESPAYE,ID,CLE) VALUES (:CLE_PRESPAYE,:ID,:CLE)



Avec le mode GLOBAL, une connexion est partagée par toutes les tables de toutes les bibliothèques pour lesquelles vous spécifiez CONNECTION = GLOBAL.

Utilisez cette option avec prudence. Lorsque vous utilisez une connexion globale pour ouvrir plusieurs tables, effectuer un commit / rollback sur une table qui est mise à jour s'applique également à toutes les autres tables ouvertes. Même si vous ouvrez une table seulement pour la lire, son curseur en lecture pourrait être resynchronisé à la suite de cette validation / annulation. Si le curseur est resynchronisé, il n'y a aucune garantie que le contenu de la nouvelle table corresponde au contenu de la table ouverte initialement.



Type de connexion 	Définition 	Remarques & points d'attention 
SHAREDREAD	Toutes les opérations de lectures sur les tables d'une même bibliothèque partagent la même connexion.	Une connexion séparée est initialisée pour les opérations de mises à jour.
GLOBALREAD	Toutes les opérations de lectures sur les tables de plusieurs bibliothèques partagent une seule connexion.	Une connexion séparée est initialisée pour les opérations de mises à jour.
UNIQUE	Chaque table utilise sa propre connexion/session.	<p>Le nombre de sessions peut augmenter rapidement et atteindre les limites autorisées sur la base de données.</p> <p>Ces nombreux accès à la base de données ont un coût et peuvent pénaliser les performances de la base.</p>
GLOBAL	La connexion est partagée par toutes les tables de toutes les bibliothèques pour lequel lesquelles vous spécifiez CONNECTION = GLOBAL.	<p>Utilisez cette option avec prudence. Lorsque vous utilisez une connexion globale pour ouvrir plusieurs tables, effectuez un commit / rollback sur une table qui est mise à jour s'applique également à toutes les autres tables ouvertes.</p> <p>Même si vous ouvrez une table seulement pour la lire, son curseur en lecture pourrait être resynchronisé à la suite de cette validation / annulation.</p> <p>Si le curseur est resynchronisé, il n'y a aucune garantie que le contenu de la nouvelle table corresponde au contenu de la table ouverte initialement.</p>



4. Exécution de code SAS sans consultation préalable des bibliothèques ORACLE

Maintenant nous exécutons un programme SAS utilisant deux bibliothèques Oracle, mais sans avoir au préalable consulté leur contenu (via la fenêtre « liste des serveurs » de SAS Enterprise Guide).

```
proc print data=ORA10LOC.class;
run;

proc print data=ORACLE2.class2;
run;
```

SHAREREAD :

Nous avons 2 CONNEXIONS ouvertes sur la base de données :

USERNAME	TIMESTAMP	LOGOFF_TIME	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	09/01/2013 14:07:56		458902			
SCOTT	09/01/2013 14:07:56		442548			

Ainsi que 2 SESSIONS :

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	146	442548	INACTIVE	WAITING	45	2013/01/09 14:07:56	45
SCOTT	bi93	149	458902	INACTIVE	WAITING	45	2013/01/09 14:07:56	45

Si l'on regarde ce que fait chaque session, nous constatons que la session 149 est en train de lire la table ORA10LOC.CLASS et la 146 la table ORACLE2.CLASS2 :

SID	FIRST_LOAD_TIME	LAST_LOAD_TIME	ROWS_PROCESSED	SQL_TEXT	COMMAND_TYPE	LAST_ACTIVE_TIME
146	2013-01-09/13:43:52	2013-01-09/13:43:52	114	SELECT * FROM CLASS2	3	09/01/2013 14:07:56
149	2013-01-09/11:35:20	2013-01-09/11:35:20	240	SELECT * FROM CLASS	3	09/01/2013 14:07:56

Jointure entre deux tables :

ORACLE10.CLE_A

	CLE	ID
1	0000007503 00000032 000017 090	1
2	0000003571 00000022 000078 039	2
3	0000001246 00000018 000077 043	3
4	0000009675 00000026 000071 055	4
5	0000005312 00000086 000014 086	5
6	0000006530 00000076 000070 032	6
7	0000005220 00000093 000004 061	7
8	0000006575 00000037 000058 031	8
9	0000005392 00000028 000052 055	9
10	000000982 00000010 000058 067	10

ORACLE2.CLE_B

	CLE	ID
1	0000007503 00000032 000017 090	8
2	0000003571 00000022 000078 039	9
3	0000001246 00000018 000077 043	10
4	0000009675 00000026 000071 055	11
5	0000005312 00000086 000014 086	12
6	0000006530 00000076 000070 032	13
7	0000005220 00000093 000004 061	14
8	0000006575 00000037 000058 031	15
9	0000005392 00000028 000052 055	16
10	000000982 00000010 000058 067	17
11	0000004827 00000014 000075 096	18
12	0000007579 00000096 000009 067	19
13	0000008920 00000028 000066 079	20

```

PROC SQL;
    CREATE TABLE WORK.RESULTAT AS
        SELECT t1.CLE,t2.ID,t1.ID
        FROM ORA10LOC.base_cleA t1
            INNER JOIN ORACLE2.base_cleB t2 ON (t1.ID = t2.ID);
QUIT;

```

USERNAME	TIMESTAMP-4+8/24	LOGOFF_TIME+4/24	OS_PROCESS	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE
SCOTT	06/01/2013 23:05:06		680166			
SCOTT	06/01/2013 23:05:06		532494			

USERNAME	OSUSER	SID	SPID	STATUS	STATE	SECONDS_IN_WAIT	LOGON_TIME	LAST_CALL_ET
SCOTT	bi93	145	532494	INACTIVE	WAITING	116	2013/01/06 23:05:06	116
SCOTT	bi93	147	680166	INACTIVE	WAITING	116	2013/01/06 23:05:06	116

SID	L.LAST_ACTIVE_TIME-4+8/24	ROWS_PROCESSED	SQL_TEXT
145	06/01/2013 23:05:07	13	SELECT * FROM BASE_CLEB
147	06/01/2013 23:05:07	10	SELECT * FROM BASE_CLEA

5. Autres options

Différer une connexion jusqu'à ce qu'elle soit nécessaire

Comme nous l'avons vu dans [le premier exemple de cet article](#), une CONNEXION à la base de données est faite à « l'ouverture » de la bibliothèque, avant même d'avoir ouvert une table de cette bibliothèque.

Ce comportement est modifiable par l'intermédiaire de l'option DEFER. Cette option est accessible dans la SMC, dans les options avancées des bibliothèques :

Différer une connexion jusqu'à ce qu'elle soit nécessaire :

Oui

Oui

Non

En positionnant l'option suivante à OUI, la CONNEXION et la création de la SESSION sont faites uniquement à l'ouverture de la table.

Cette option fonctionne si :

- La bibliothèque n'est PAS pré-affectée.
- Le type de connexion n'est pas défini à UNIQUE. Pour rappel, avec UNIQUE une connexion est initialisée à chaque ouverture de table.

6. Conclusion

Comme nous avons pu le voir au travers de nos exemples, le choix du mode de connexion n'est pas à prendre à la légère et demande réflexion. Il est primordial de réfléchir à ces choix et de tester en environnement de recette avant d'appliquer ces modifications en production. L'aide des équipes DBA peut être nécessaire afin de vous aider à monitorer l'activité et pour vous aider dans votre choix.

7. Boîte à outils

Voici quelques informations utiles et astuces.

Dans oracle, une session peut avoir 2 statuts :

- **Active** : La session est en train d'exécuter une requête SQL
- **Inactive** : Aucune requête en cours. Cela signifie que le client est connecté au serveur mais ne fait aucun appel à la base de données.

Connaître le nombre de sessions versus le nombre de sessions maximum

```
SELECT 'Currently, ' || (SELECT COUNT(*) FROM V$SESSION) || ' out of ' ||  
DECODE(VL.SESIONS_MAX,0,'unlimited',VL.SESIONS_MAX) || ' connections are used.'  
AS USAGE_MESSAGE FROM V$LICENSE VL
```

Connaître le nombre de sessions par utilisateur

```
SELECT 'USERNAME : ' || USERNAME || ' - NB SESSIONS : ' || count(*)  
FROM V$SESSION where USERNAME is not null  
GROUP BY USERNAME;
```

Lister les sessions

```
SELECT 'ORACLE USERNAME : ' || s.USERNAME || ' | OS _USERNAME : ' ||  
s.OSUSER || ' | SESSION ID : ' ||  
s.SID || ' | ' ||  
s.STATUS  
|| ' | LOGON_TIME : ' ||  
TO_CHAR(s.LOGON_TIME, 'hh24:mi:ss') || ' | LAST_CALL : ' ||  
s.LAST_CALL_ET || ' second(s)'  
FROM V$PROCESS p,  
V$SESSION s,  
V$DISPATCHER d,  
V$CIRCUIT c,  
V$SHARED_SERVER ss  
WHERE p.ADDR = s.PADDR  
AND s.SADDR=c.SADDR (+)  
AND c.DISPATCHER=d.PADDR (+)  
AND c.SERVER=ss.PADDR (+)  
AND s.USERNAME IS NOT NULL  
ORDER BY s.USERNAME, p.SPID;
```

Il existe d'autres moyens de tracer les connexions à la base de données. En fonction du paramétrage de cette base et si l'audit des sessions est activé, les connexions et déconnexions peuvent être enregistrées.

Sous Oracle, il est possible de requêter la vue **dba_audit_sessions**. Cette vue permet ainsi de connaître l'ensemble des connexions à la base.

8. Liens utiles

Les processus dans un système de base de données Oracle et les différentes configurations possibles : [Oracle Database Concepts - Process Architecture](#).

Matrice de compatibilité SAS et les différents systèmes de base de données : [SAS/ACCESS Validation Matrix](#).

Présentation des différents modes de connexion : [SAS/ACCESS\(R\) 9.3 for Relational Databases \(CONNECTION= Libname Option\)](#).

Nicolas Housset
Consultant Support Clients SAS France