

TP Oracle

REMARQUE IMPORTANTE

Pour ce TP, on demande de réaliser **un compte-rendu du travail effectué sous forme d'un PDF** (sous forme nom_etudiant.PDF). Ce compte-rendu comprendra un rapport (manuscrit ou réalisé sur traitement de texte) détaillant les points intéressants abordés à chaque question :

- requêtes SQL correspondant aux questions posées
- résultat de l'exécution de la requête (avec des capture écran)
- réponses aux questions soulevées dans le sujet,
- explication des parties non triviales des requêtes complexes,
- analyse et commentaire des jeux d'essais.

Un script SQL qui contient les instructions SQL de chaque partie sous forme Nom_Etudiant.sql

Le contexte du cas traité est l'internet des objets (Internet of Things, IoT). Dans ce contexte, nous souhaitons modéliser une plateforme de gestion de services. Des utilisateurs, qui possèdent des objets connectés, souscrivent à des services. Lorsque les objets envoient des données, elles sont transmises aux services souscrits par leur propriétaire. Pour savoir à qui les objets connectés appartiennent et quels sont les services souscrits par leur propriétaire, nous utilisons une base de données dont le schéma relationnel est donné ci- dessous :

USER (IDUSER, LASTNAME, FIRSTNAME, EMAIL)

SERVICE (IDSERVICE, NAME, SERVICETYPE)

THING (MAC, IDUSER*, THINGTYPE, PARAM)

SUBSCRIBE (IDUSER*, IDSERVICE*)

Dans ce schéma :

- Chaque objet connecté est identifié par son adresse physique unique (adresse MAC) sur 17 caractères. Le champ THINGTYPE permet d'indiquer si l'objet est particulier (sinon, le champ n'est pas spécifié). Le champ PARAM permet de fournir des paramètres optionnels concernant l'objet.
- Le champ SERVICETYPE de la table service permet d'indiquer si le service est particulier (sinon, le champ n'est pas spécifié).

➤ L'ensemble des instances des différentes tables sont représentées dans les tableaux suivants :

TABLE USER :

IDUSER	LASTNAME	FIRSTNAME	EMAIL
1	Souad	MESBAH	souad.mesbah@gmail.com
2	Younes	CHALAH	younes.chalah@gmail.com
3	Chahinaz	MELEK	chahinaz.melek@gmail.com
4	Samia	OUALI	samia.ouali@gmail.com
5	Djamel	MATI	djamel.mati@gmail.com
6	Assia	HORRA	assia.horra@gmail.com
7	Lamine	MERABAT	Lamine.MERABAT@gmail.com
8	Seddik	HMIA	seddik.hmia@gmail.com
9	Widad	TOUATI	widad.touati@gmail.com

TABLE SERVICE :

IDSERVICE	NAME	SERVICETYPE
1	myKWHome	smarthome
2	FridgAlert	smarthome
3	RUNstats	quantifiedself
4	traCARE	quantifiedself
5	dogWATCH	
6	CarUse	

TABLE THING :

MAC	IDUSER	THINGTYPE	PARAM
f0:de:f1:39:7f:17	1		
f0:de:f1:39:7f:18	2		
f0:de:f1:39:7f:19	2	thingtempo	60
f0:de:f1:39:7f:25	10		
f0:de:f1:39:7f:20	2	thingtempo	1.5
f0:de:f1:39:7f:21	4		
f0:de:f1:39:7f:22	4		

TABLE SUBSCRIBE:

IDUSER	IDSERVICE
2	1
2	2
1	3
3	7

TP N° 2

Création et manipulation d'une BD

Ce TP concerne la création, la manipulation et l'interrogation d'une BD ainsi que les contraintes d'intégrité statiques, l'évolution de schéma ...etc. Il est conseillé de créer un utilisateur ainsi que les objets nécessaires de façon à obtenir plus de souplesse dans l'administration. La création d'un utilisateur nécessite au minimum deux tablespaces, un par défaut ainsi qu'un temporaire. Ces opérations doivent être effectuées en tant que DBA (Data Base Administrateur).

Quelques Syntaxes :

- **Création de Tablespace**

```
CREATE TABLESPACE xxx_tbs DATAFILE 'C:\tbs_xxx.dat' SIZE 100M AUTOEXTEND ON ONLINE;
```

Description :

CREATE TABLESPACE xxx_tbs	Spécifie le nom du tablespace de la base.
DATAFILE 'C:\tbs_xxx.dat' SIZE 100M	Spécifie le nom complet du fichier système ainsi que sa taille en méga octets.
AUTOEXTEND ON	La taille est augmentée automatiquement en cas de saturation
ONLINE	Disponible immédiatement suite à sa création

- **Création de Tablespace temporaire**

```
CREATE TEMPORARY TABLESPACE xxx_TempTBS TEMPFILE 'C:\temp_xxx.dat' SIZE 100M AUTOEXTEND ON;
```

- **Création d'un utilisateur**

```
Create User Nom_User Identified by MotDePasse Default Tablespace Nom_Tablespace  
Temporary Tablespace Nom_Temp_Tablespace;
```

- **Donner tous les droits à l'utilisateur :**

```
GRANT ALL privileges to Nom_User
```

Pour Activer une contrainte

```
ALTER TABLE nomTable ENABLE CONSTRAINTnomContrainte [EXCEPTIONS INTO tableErreurs];
```

Créer la table « tableErreurs » pour identifier les tuples qui ne vérifient pas la contrainte.

```
CREATE TABLE TableErreurs (adresse ROWID, utilisateur VARCHAR2(30), nomTable VARCHAR2(30), nomContrainte VARCHAR2(30));
```

Travail demandé

Partie 1 : Création des Tablespaces et des utilisateurs

1. Créer deux Tablespaces *IOT_TBS* et *IOT_TempTBS*
2. Créer un utilisateur **DBAIOT** en lui attribuant les deux tablespaces créés précédemment
3. Donner tous les privilèges à cet utilisateur.

Partie 2 : Langage de définition de données

4. Créer les relations de base avec toutes les contraintes d'intégrité.
5. Ajouter l'attribut **ADRESSUSER** de type chaîne de caractères dans la relation **USER**.
6. Ajouter la contrainte not null pour les attributs **ADRESSUSER** et **LASTNAME** de la relation **USER**.
7. Modifier la longueur de l'attribut **ADRESSUSER** (agrandir, réduire).
8. Renommer la colonne **ADRESSUSER** dans la table **USER** par **ADRUSER**. Vérifier.
9. Supprimer la colonne **ADRUSER** dans la table **USER**. Vérifier la suppression.
10. Un utilisateur s'inscrit à un service pour une période délimitée par un début et fin. Donner les instructions SQL pour répondre à ce besoin.

Partie 3 : Langage de manipulation de données

11. Remplir toutes les tables par les instances représentées ci-dessus. Quels sont les problèmes rencontrés ?

Partie 4 : Gestion des utilisateurs

Au cours des trois premières parties vous avez travaillé seul dans un schéma de nom égal à votre nom d'utilisateur. Nous allons vérifier que le SGBD gère la concurrence d'accès à des objets de la base entre plusieurs utilisateurs différents. Tout utilisateur qui crée des objets est propriétaire de ces objets. Le créateur d'un objet peut décider de donner (ou de supprimer) certains droits d'accès à tout autre utilisateur de sa connaissance.

GRANT privilège [ON table/vue] TO utilisateur [WITH GRANT OPTION] → privilège sur un objet

GRANT privilège TO utilisateur [WITH ADMIN OPTION] → privilège système

Cet ordre permet de donner le privilège concerné sur la table ou la vue à l'utilisateur.

Un utilisateur ayant accordé un privilège peut le reprendre à tout moment à l'aide de l'ordre **REVOKE** :

REVOKE privilège ON [table/vue] FROM utilisateur → privilège sur un objet

REVOKE privilège FROM utilisateur → privilège système

Les privilèges qui peuvent être donnés sont les suivants :

SELECT droit de lecture, **INSERT** droit d'insertion de lignes, **UPDATE** droit de mise-à-jour de lignes, **DELETE** droit de suppression de lignes,

ALTER droit de modification de la définition de la table, **INDEX** droit de création d'index, **ALL** tous les droits ci-dessus.

*Un utilisateur ayant reçu un privilège avec la mention facultative **WITH GRANT OPTION** peut les transmettre à son tour à un autre utilisateur (User3).*

Questions :

1. Connectez vous avec l'utilisateur **DBAIOT** et créez un autre utilisateur : **Admin** en lui donnant les mêmes tablespace que **DBAIOT**.
2. Connectez-vous à l'aide cet utilisateur. Que remarquez-vous ?
3. Donnez le droit de création d'une session pour cet utilisateur (Create Session) et reconnectez-vous.
4. Donnez les privilèges suivants à **Admin**: créer des tables, des utilisateurs. Vérifiez.
5. Exécutez la requête **Q1** suivante : **Select * from DBAIOT.USERS** ; Que remarquez-vous ?
6. Donnez le droit de lecture à cet utilisateur pour la table **USERS**. Exécutez la requête **Q1** maintenant.
7. On veut créer une vue **USER_THING** qui sauvegarde pour chaque utilisateur ses objets connectés. Que faut-il faire ? Que remarquez-vous ?
8. Donnez le droit de création de vue à cet utilisateur, le droit de lecture sur la table **THING** et réessayez de refaire la création de la vue.
9. Créez un index **NAMESERVICE_IX** sur l'attribut **NAME** de la table **SERVICE**. Que remarquez-vous ?

10. Donnez le droit de création d'index à **Admin** pour la table **SERIVCE**, ensuite réessayez de créer l'index. Que se passe-t-il ?
11. Enlevez les privilèges précédemment accordés.
12. Vérifiez que les privilèges ont bien été supprimés.
13. Créez un profil « **IOT_Profil** » qui est caractérisé par : (3 sessions simultanées autorisées, Un appel système ne peut pas consommer plus de 35 secondes de CPU, Chaque session ne peut excéder 90 minutes, Un appel système ne peut lire plus de 1200 blocs de données en mémoire et sur le disque, Chaque session ne peut allouer plus de 25 ko de mémoire en SGA, Pour chaque session, 30 minutes d'inactivité maximum sont autorisées, 5 tentatives de connexion avant blocage du compte, Le mot de passe est valable pendant 50 jours et il faudra attendre 40 jours avant qu'il puisse être utilisé à nouveau, 1 seul jour d'interdiction d'accès après que les 5 tentatives de connexion ont été atteintes, La période de grâce qui prolonge l'utilisation du mot de passe avant son changement est de 5 jours).
14. Affectez ce profil à l'utilisateur **Admin**.
15. Créez le rôle : « **SUBSCRIBE_MANAGER** » qui peut voir les tables **USERS**, **SERVICE** et peut modifier les lignes de la table **SUBSCRIBE**.
16. Assignez ce rôle à **Admin**. Vérifier que les autorisations assignées au rôle **SUBSCRIBE_MANAGER**, ont été bien transférées sur l'utilisateur à **Admin**.

Partie 5 : Dictionnaire de données

ORACLE dispose d'une structure centralisée (le dictionnaire de données) contenant la description de tous les objets (tables, vues, utilisateurs, ...) gérés par ce SGBD. Ce dictionnaire regroupe donc toutes les informations nécessaires au fonctionnement du SGBD. Il présente la particularité d'être lui-même organisé comme une base de données (on parle de meta-base) et d'être par conséquent accessible directement à travers SQL. Pour des raisons d'intégrité aisément compréhensibles, la plupart des tables du dictionnaire ne sont pas modifiables par les utilisateurs (seul l'administrateur de la base SYS, a accès à ces dernières). A l'opposé, l'accès à leur contenu est généralement autorisé, ce qui va nous permettre d'explorer quelque peu le dictionnaire afin de mieux appréhender cette structure de données essentielle au SGBD. D'une manière plus précise, les tables du dictionnaire sont généralement cryptées. Leur contenu est par contre accessible par l'intermédiaire de vues. Elles sont de quatre grands types:

- Vues relatives aux objets d'un utilisateur : **USER_***
- Vues relatives aux objets accessibles à un utilisateur : **ALL_***
- Vues relatives aux administrateurs : **DBA_***
- Vues relatives au suivi des performances : **V\$***

Au cours de ce TP, nous allons nous intéresser plus particulièrement aux vues de type ALL et USER. Pour vous guider dans vos travaux, vous trouvez dans le tableau suivant l'ensemble des vues du dictionnaire de données Oracle de type USER.

Nom de la vue	Synonyme	Contenu
DICTIONARY	DICT	Toutes les vues du dictionnaire, pour le développeur ou le DBA : Nom de la vue, description
USER_TABLES	TABS	Mes tables : nom, tablespace, stockage, statistiques, cluster éventuel
USER_TAB_COLUMNS	COLS	Colonnes de mes tables : Nom colonne, type, longueur, obligatoire
USER_VIEWS	-	Mes vues : Nom, texte de l'ordre SQL associé, type
USER_INDEXES	IND	Mes indexes : Nom, table indexée, unicité, stockage, statistiques
USER_IND_COLUMNS	-	Nom index, nom table, nom colonne, position et longueur
USER_CLUSTERS	CLU	Mes clusters ; Nom, stockage, statistiques
USER_OBJECTS	OBJ	Mes objets : tables, vues, indexes, clusters, synonymes, procédures, fonction, package, séquence
USER_SEQUENCES	SEQ	Mes séquences : Valeur min, max, incrément, cycle, cache
USER_SYNONYMS	SYN	Mes synonymes : Nom du synonyme, de la table, propriétaire et dblink éventuel
USER_USERS	-	Caractéristiques générales de l'utilisateur: Nom, tablespace par défaut, tablespace temporaire
USER_CONSTRAINTS	-	Mes contraintes : Nom, type, table d'accueil, statut
USER_DB_LINKS	-	Mes database links (liens base distantes) : Nom, user distant, mot de passe, serveur distant, date de création
USER_TAB_PRIVS	-	Des privilèges donnés ou reçus : Bénéficiaire, propriétaire, créateur
USER_EXTENTS	-	Caractéristiques de stockage de mes objets : Nom du segment, de la partition, du tablespace, taille en octets et en blocs
USER_TS_QUOTAS	-	Quota d'écriture autorisé sur les tablespace : Nom du tablespace, taille max en octets et en blocs

Questions :

Connectez vous (les tables du TP précédent étant créées) ;

D'une manière générale il est utile de connaître un minimum du contexte des données que l'on souhaite consulter ou mettre à jour. La liste de mes tables, les noms et types des colonnes d'une de mes tables, les tables accessibles, etc. Ces informations qui concernent les informations, sont appelées méta données (ou META DATA).

Le dictionnaire de données (DD) est le référentiel unique qui décrit tous les objets physiques et logiques d'une base de données Oracle. Il est accédé par les utilisateurs DBA ou non.

Lien du catalogue oracle : https://docs.oracle.com/cd/B10501_01/nav/catalog_views.htm?remark=homepage

1. Connecter en tant que « **System** ». Lister le catalogue « **DICT** ». Il contient combien d'instances ? Donner sa structure ? (Describe DICT; select * from dict;)
2. Donner le rôle et la structure des tables (ou vues) suivantes : ALL_TAB_COLUMNS, USER_USERS, ALL_CONSTRAINTS et USER_TAB_PRIVS. (exemple : describe USER_USERS; select COMMENTS from DICT where TABLE_NAME='USER_USERS';)
3. Trouver le nom d'utilisateur avec lequel vous êtes connecté (sans utiliser show user, en utilisant le dictionnaire)?
4. Comparer la structure et le contenu des tables ALL_TAB_COLUMNS et USER_TAB_COLUMNS ?
5. Vérifiez que les tables de la partie 1 ont été réellement créées (afficher la liste des tables de l'utilisateur connecté) ? Donner toutes les informations sur ces tables ?
6. Lister les tables de l'utilisateur « **system** » et celles de l'utilisateur **DBAIOT** (l'utilisateur de la partie 1).
7. Donner la description des attributs des tables **THING** et **SUBSCRIBE** (Exploiter la table **USER_TAB_COLUMNS**).
8. Comment peut-on vérifier qu'il y a une référence de clé étrangère entre les tables **THING** et **SUBSCRIBE**?
9. Donner toutes les contraintes créées lors de la partie et les informations qui les caractérisent (Exploitez la table **USER_CONSTRAINTS**);
10. Retrouver toutes les informations permettant de recréer la table **SUBSCRIBE**.
11. Trouver tous les privilèges accordés à **Admin** (comme on les a supprimé dans la partie 2, recréez 2 privilèges système et un privilège objet pour admin et les afficher en tant que admin et en tant que system).
12. Trouver les rôles donnés à l'utilisateur **Admin**.
13. Trouver tous les objets appartenant à **Admin**.
14. L'administrateur cherche le propriétaire de la table **SUBSCRIBE**, comment il pourra le trouver ?
15. Donner la taille en Ko de la table **SUBSCRIBE** (utiliser desc user_segments;).
16. Vérifier l'effet produit par chacune des commandes de définition de données de la partie 1 sur le dictionnaire :
Créez un nouvel utilisateur comme dans la partie 1, donner lui tous les privilèges ensuite connectez-vous avec cet utilisateur que vous venez de créer

La syntaxe SQL de création d'un utilisateur

CREATE USER utilisateur

IDENTIFIED {BY motdePasse | EXTERNALLY | GLOBALLY AS 'nomExterne' }

[DEFAULT TABLESPACE nomTablespace [QUOTA {entier [K | M] | UNLIMITED} ON nomTablespace]]

[TEMPORARY TABLESPACE nomTablespace] [PROFILE nomProfil] [PASSWORD EXPIRE] [ACCOUNT {LOCK | UNLOCK}];

- **IDENTIFIED BY mot de Passe** permet d'affecter un mot de passe à un utilisateur local (cas le plus courant et le plus simple).
- **IDENTIFIED BY EXTERNALLY** permet de se servir de l'authenticité du système d'exploitation pour s'identifier Oracle (cas des compte OPS\$ pour Unix).
- **IDENTIFIED BY GLOBALLY** permet de se servir de l'authenticité d'un système d'annuaire.
- **DEFAULT TABLESPACE** nomTablespace associe un espace disque de travail (appelé tablespace) à l'utilisateur.
- **TEMPORARY TABLESPACE** nomTablespace associe un espace disque temporaire (dans lequel certaines opérations se dérouleront) à l'utilisateur.
- **QUOTA** permet de limiter ou pas chaque espace alloué.
- **PROFILE** nomProfil affecte un profil (caractéristiques système relatives au CPU et aux connexions) à l'utilisateur.
- **PASSWORD EXPIRE** pour obliger l'utilisateur à changer son mot de passe à la première connexion (par défaut il est libre). Le DBA peut aussi changer ce mot de passe.
- **ACCOUNT** pour verrouiller ou libérer l'accès à la base (par défaut UNLOCK).

La syntaxe SQL de Création d'un Profil

CREATE PROFILE nomProfil

LIMIT { Paramètre Ressource | Paramètre Mot de Passe } [Paramètre Ressource | Paramètre Mot de Passe];

Paramètre Ressource : {{SESSIONS_PER_USER|CPU_PER_SESSION|CPU_PER_CALL|CONNECT_TIME|
IDLE_TIME|LOGICAL_READS_PER_SESSION|LOGICAL_READS_PER_CALL | COMPOSITE_LIMIT}{entier | UNLIMITED | DEFAULT } | PRIVATE_SGA
{entier[K|M] | UNLIMITED | DEFAULT}}

Paramètre Mot de Passe : { FAILED_LOGIN_ATTEMPTS | PASSWORD_LIFE_TIME | PASSWORD_REUSE_TIME | PASSWORD_REUSE_MAX |
PASSWORD_LOCK_TIME | PASSWORD_GRACE_TIME } { expression | UNLIMITED | DEFAULT } }

Les options principales sont les suivantes :

- SESSIONS_PER_USER : nombre de sessions concurrentes autorisées.
- CPU_PER_SESSION : temps CPU maximal pour une session en centièmes de secondes.
- CPU_PER_CALL : temps CPU autorisé pour un appel noyau en centièmes de secondes.
- CONNECT_TIME : temps total autorisé pour une session en minutes (pratique pour les examens de TP minutés).
- LOGICAL_READS_PER_SESSION : définir le nombre maximal de bloc lus durant une session. On parlera ici des blocs lus sur le disque et dans la mémoire.
- LOGICAL_READS_PER_CALL : définir le nombre maximal de bloc lus durant un "appel serveur". On parlera ici des blocs lus sur le disque et dans la mémoire
- IDLE_TIME : temps d'inactivité autorisé, en minutes, au sein d'une même session (pour les étudiants qui ne clôturent jamais leurs sessions).
- PRIVATE_SGA : espace mémoire privé alloué dans la SGA (System Global Area).
- FAILED_LOGIN_ATTEMPTS : nombre de tentatives de connexion avant de bloquer l'utilisateur.
- PASSWORD_LIFE_TIME : nombre de jours de validité du mot de passe (il expire s'il n'est pas changé au cours de cette période).
- PASSWORD_REUSE_TIME : nombre de jours avant que le mot de passe puisse être utilisé à nouveau. Si ce paramètre est initialisé à un entier, le paramètre PASSWORD_REUSE_MAX doit être passé à UNLIMITED.
- PASSWORD_REUSE_MAX : nombre de modifications de mot de passe avant de pouvoir réutiliser le mot de passe courant. Si ce paramètre est initialisé un entier, le paramètre PASSWORD_REUSE_TIME doit être passé à UNLIMITED.
- PASSWORD_LOCK_TIME : nombre de jours d'interdiction d'accès à un compte après que le nombre de tentatives de connexions a été atteint.
- PASSWORD_GRACE_TIME : nombre de jours d'une période de grâce qui prolonge l'utilisation du mot de passe avant son changement (un message d'avertissement s'affiche lors des connexions). Après cette période le mot de passe expire.

Syntaxe de Création d'un rôle

CREATE ROLE nomRle [NOT IDENTIFIED | IDENTIFIED

{BY motdePasse | USING [sechma.]paquetage | EXTERNALLY | GLOBALLY }]