
TP 1 SQL et PL/SQL (2 séances)

Introduction

Pour chaque personne, connectez-vous à Oracle sur votre compte et changez le mot de passe.

Pour se connecter sur les postes windows vous disposez de deux solutions : en mode console avec l'application `sqlplus` ; 2 - en mode interface homme/machine avec l'application `SQL_Developer`¹

Pour travailler en mode console :

- lancez une invite de commande (menu : Démarrer : exécuter - cmd)
- positionnez-vous dans un répertoire de votre compte qui servira pour le module (vous devez y avoir les droits d'écriture, créez le si nécessaire).
- connectez-vous au serveur oracle à l'aide de la commande
`sqlplus <login>/<motDePasse>@//<nomServeur>:<port>/<nomInstanceOracle>`
où :
 - `login` est votre login INSA.
 - `motDePasse` est votre mot de passe Oracle (non géré par le CAS).
 - `nomServeur` est le nom DNS de la machine supportant Oracle, ici `if-oracle`
 - `port` est le port sur lequel le serveur oracle écoute pour la connexion `sqlplus`. Ici c'est le port standard 1521.
 - `nomInstanceOracle`... comme le nom l'indique. Ici `if`

La commande de connexion doit donc ressembler à

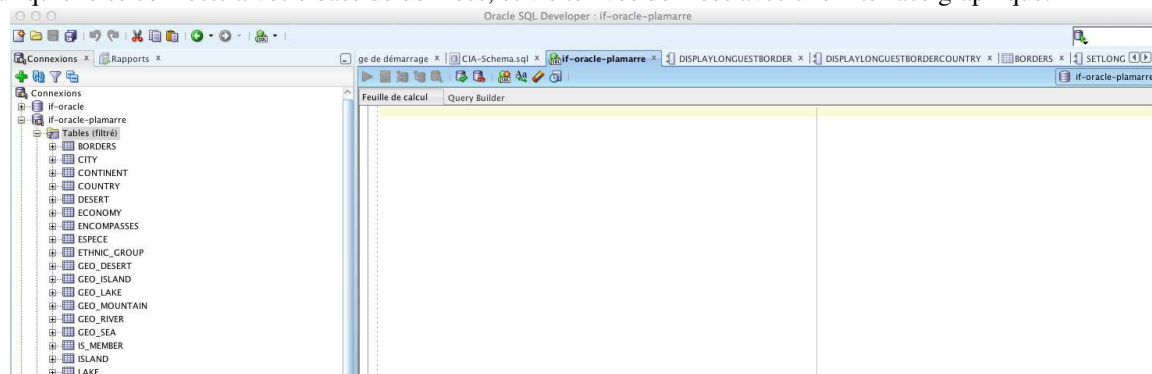
```
sqlplus tdupont/kDi3aR23!@//if-oracle:1521/if
```

La commande SQL vous permettant de changer le mot de passe Oracle est `password`.

Problème 1.

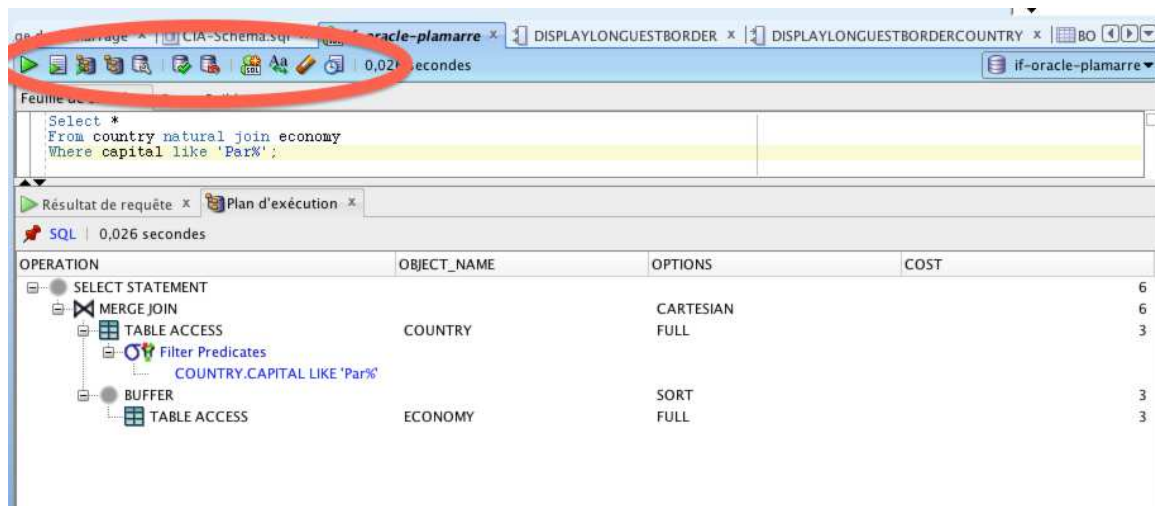
Vous pouvez utiliser une deuxième manière de vous connecter à une base de données Oracle. Plus récente, mais pas toujours disponible, cette interface est bien plus conviviale pour l'utilisateur, mais plus lourde pour le client et le réseau. Pour cette raison l'interface texte reste utilisable dans des situations plus tendues et de ce fait, reste l'outil de base. Il est donc fortement conseillé de savoir l'utiliser.

Lancez l'application `SQL_Developer` qui doit être présente sur vos postes, si nécessaire configurez l'application pour qu'elle se connecte à votre base de données, et visitez vos données avec une interface graphique.



Après configuration et quelques essais personnels, vous pourrez par exemple regarder les plans d'exécution proposés par Oracle pour vos requêtes.

1. Ces deux clients sont téléchargeables chez Oracle et installables gratuitement sur vos machines Windows, Linux, Mac.



Bonne visite.

Problème 2.

Choisissez le compte d'un des membres du binôme pour travailler et connectez-vous à ce compte.

Vous avez à votre disposition la base de données contenant des informations sur tous les pays du monde.

ATTENTION, cette base N'a PAS été créée dans le but de ce TP. C'est une base qui est mise à disposition sur le site <http://www.dbis.informatik.uni-goettingen.de/Mondial/> où vous pouvez peut-être trouver une version plus récente. Elle résulte d'une compilation, intégration, de plusieurs sources.

La base peut, par certains aspects, être un peu déroutante. Le schéma ne répond pas nécessairement à toutes les attentes, la documentation fournie n'est pas toujours complète et certaines informations présentent des caractéristiques qui pourraient amener certains à les qualifier d'incohérentes. Ces défauts ont l'avantage de constituer un terrain d'expérience très proche de ce qui se présente dans la réalité. Cette situation vous impose aussi de prendre un certain recul et de quelquefois remettre en cause certains points. Attention, ne sortez pas de votre rôle. Vous ne devez pas modifier le schéma ou des données sans l'accord de l'administrateur de la base et des utilisateurs. Ces rôles sont tenus par les enseignants de TP...

La structure de la base peut être schématisée comme suit :

```

table Country (Name, Code, Capital, Province, Area, Population)
table City (Name, Country, Province, Population, Longitude, Latitude)
table Province (Name, Country, Population, Area, Capital, CapProv)
table Economy (Country, GDP, Agriculture, Service, Industry, Inflation)
table Population (Country, Population_Growth, Infant_Mortality)
table Politics (Country, Independence, Government)
table Language (Country, Name, Percentage)
table Religion (Country, Name, Percentage)
table Ethnic_Group (Country, Name, Percentage)
table Continent (Name, Area)
table Borders (Country1, Country2, Length)
table encompasses (Country, Continent, Percentage)
table Organization (Abbreviation, Name, City, Country, Province, Established)
table is_member (Country, Organization, type)
table Mountain (Name, Height, Coordinates)
table Desert (Name, Area)
table Island (Name, Islands, Area, Coordinates)
table Lake (Name, Area)
table Sea (Name, Depth)
table River (Name, River, Lake, Sea, Length)
table geo_Mountain (Mountain, Country, Province)
table geo_Desert (Desert, Country, Province)
table geo_Island (Island, Country, Province)
table geo_River (River, Country, Province)
table geo_Sea (Sea, Country, Province)
table geo_Lake (Lake, Country, Province)
  
```

```
table merges_with (Sea1, Sea2)
table located (City, Province, Country, River, Lake, Sea)
```

Question 2.1. Installation de la base sur le compte d'un binôme.

1. Le script de création des tables (mondial-schema_a_distribuer.sql) est disponible sur moodle. Créez la structure de la base en exécutant ce script sur votre compte. Pour cela, sous sqlplus, tapez la commande @nomFichier. Attention, vérifiez bien que votre fichier est bien de type sql.
2. Le script mondial-inputs_a_distribuer.sql permettant d'alimenter la base est lui aussi disponible sur moodle. Utilisez-le pour alimenter votre base de données.

Question 2.2. Contraintes

1. Modifiez la définition des tables en rajoutant les contraintes
 - (a) de clé étrangère (en particulier sur les tables : Country, City et Province) avec d'éventuels effets de cascade
 - (b) d'unicité
 - (c) sur les attributs
 - i. table Borders, les frontières doivent avoir une longueur strictement positive
 - ii. table Country, le nom du pays doit être connu et deux pays ne peuvent avoir le même nom.
 - iii. Quelle que soit la table, tous les pourcentages doivent être entre 0 et 100. Modifiez au moins une table en conséquence.
2. Pour chaque erreur bloquant un des processus ci-dessus, identifiez clairement le problème et proposez des solutions.
3. Modification des données : suite à des mouvements politiques, le pays Atlantis se divise en deux pays (Atlan et Tis). Modifiez la base de sorte à ce qu'elle reflète ces modifications : supprimez le pays Atlantis et créez les 2 nouveaux pays ².

Question 2.3. Requêtes Traduisez les requêtes suivantes en SQL Oracle :

1. Lister tous les pays par ordre alphabétique.
2. Donner la ou les organisation/s qui a/ont le plus de pays membres.
3. Donner la liste des pays triés par ordre de pourcentage de leur population citadine.
4. Lister les continents par ordre croissant de leur population.

2. Pour faire ce qui est demandé, pensez bien à l'ordre dans lequel vous allez réaliser vos opérations.

5. Lister les pays présentant la plus longue frontière.
6. Lister les noms de villes apparaissant dans plusieurs pays ainsi que le nombre d'occurrences.
7. Pour chaque nom de ville, donner le nombre de pays dans lequel elle apparaît. Triez par ordre décroissant du nombre d'apparition(s), puis en cas d'égalité, du nom de la ville.
8. Chercher les pays (code et nom) dont on ne connaît pas, dans la base, la date d'indépendance.
9. Lister les pays (code et nom) qui ne sont dans aucune organisation.
10. Pour tous les pays, lister les organisations dont ils sont membres (un tuple par couple pays/organisation, au moins un tuple pour chaque pays).
11. Donner la longueur totale des frontières sur chaque continent.
12. Donner les longueurs moyennes des frontières présentes sur chaque continent.
13. Lister les villes qui sont le siège d'au moins une organisation.
14. Donner la moyenne du nombre d'organisations siégeant dans une ville pour les villes accueillant au moins un siège d'organisation.
15. Pour chacune des cinq langues parlées dans le plus de pays, afficher la langue et les pays où elle est parlée.
16. Pour les cinq langues parlées par le plus grand nombre de personnes, afficher la langue et le nombre de personnes qui la parlent.
17. Écrire en PL/SQL une solution qui prend en paramètre un code de pays et qui affiche son nom ainsi que le nombre de pays frontaliers.
18. Modifier la table country en lui rajoutant une colonne (frontiereLongue) contenant le nom du pays avec laquelle le pays donné a la frontière la plus longue.
Renseigner cette colonne.
Il est suggéré de programmer pour automatiser cette tâche.
Attention, il peut y avoir plusieurs pays qui ont des frontières de la même longueur. Dans ce cas, on en prend un. . .
19. Écrire en PL/SQL du code affichant les pays qui ont acquis leur indépendance un lundi (utiliser to_char pour convertir la date).
20. Écrire du code qui permet d'insérer un pays dans la base. Cette procédure recevra comme paramètres le nom du pays, son code, la capitale, la superficie, la population, le type du gouvernement et la date d'indépendance. Traiter les exceptions.

21. Pour un code pays passé en paramètre, faire un récapitulatif de : nombre d'habitants, langues parlées avec le nombre d'habitants qui les parlent, PIB (GDP), voisins, continent(s), organisations dont il est membre, date d'indépendance. Traiter par exception le cas où le pays passé en paramètre n'existe pas.
22. Dans la base de données un certain nombre d'informations sont manquantes. Définir une mesure pour la quantité d'informations manquantes pour un pays. Ecrire une fonction qui retourne pour le code de pays passé en paramètre la valeur de cette mesure. (optionnelle)
23. En utilisant la mesure définie pour la question précédente, donner les 10 pays ayant le moins d'informations manquantes dans la base. (optionnelle)
24. En partant de la mer méditerranée, quels sont les mers (ou océans) que l'on peut atteindre en naviguant ?
25. Trouver les différents chemins (sans boucle) permettant d'aller d'un pays à l'autre, les codes des deux pays étant passés en paramètre (optionnelle)

Question 2.4. Donnez les droits en lecture/mise à jour à votre binôme.

La commande pour donner des droits à une personne est la suivante :

```
grant select on <table_name> to <user>;
grant update on <table_name> to <user>;
```

Deux solutions s'offrent à vous. 1 - taper à la main toutes ces commandes les unes après les autres (copier/coller/substitution...) ou 2 - automatiser le processus. La deuxième solution présente l'avantage d'être réutilisable en cas de rajout de tables.

Pour automatiser, vous disposez du catalogue qui vous permet de récupérer un grand nombre d'informations. En particulier, la table `user_tables` contient la description de toutes les tables de l'utilisateur.

Trouvez la requête permettant d'avoir les noms de toutes les tables de l'utilisateur.

Si l'on sait afficher les noms de toutes les tables, l'obtention des lignes de commandes souhaitées n'est pas si loin. Une bonne utilisation d'outils système (sed, awk... sous unix) peut être une piste. L'adaptation de la requête précédente peut être bien plus directe.

Écrivez une requête permettant d'afficher toutes les lignes de commande souhaitées.

Un indice : concaténation de chaîne obtenue par "||".

Si ce qu'affiche cette requête vous convient, vous êtes peut être tentés de copier/coller son résultat pour l'exécuter. Il est plus raisonnable de rediriger la sortie dans un fichier. Pour ce faire, il faut modifier quelques variables puis dire que les sorties doivent être envoyées vers un fichier, puis finalement, demander l'exécution de ce fichier.

Quelques indices : `SPOOL` permet de rediriger la sortie vers un fichier, pour éviter l'affichage des entêtes `SET HEADING OFF`. D'autres variables pouvant présenter un intérêt `ECHO`, `PAGESIZE`, `FEEDBACK`...

Faites de la documentation votre amie. Suite à quelques recherches, vous devriez être rapidement en mesure d'exécuter la requête précédente en plaçant son résultat dans un fichier que vous nommerez `grant.sql`. Bonne recherche.

Une fois satisfait du résultat, vous pouvez créer un fichier `genereGrant.sql` qui contient ces lignes. Une fois cela fait, exécuter le script sera possible dans `sqlplus` par la commande `@genereGrant.sql`. Exécuter le résultat du script, c'est à dire donner effectivement les droits, se fait de la même manière `@grant.sql`. Si vous voulez limiter au maximum le nombre de commandes tapées et que les deux étapes se fassent en une seule, il vous suffit de rajouter la commande `@grant.sql` dans votre fichier `genereGrant.sql` après que le fichier `grant.sql` ait été créé.

Petit récapitulatif. Vous venez d'écrire un script qui génère un autre script via une requête sql puis qui l'exécute. C'est une pratique assez courante que ce soit en BD ou dans d'autres domaines.

Question 2.5. Déclencheurs

1. Sur la table Politics, ajouter la contrainte que la date d'indépendance doit être inférieure ou égale à la date du jour.
 2. Écrire un déclencheur qui assure que le code d'un pays soit alphabétique et en majuscule. Indiquez en quoi cette manipulation peut-être utile et intéressante.
 3. Modifier la table Country en y ajoutant une colonne : border_length (longueur de frontières). Créer un déclencheur qui met à jour la valeur de cette colonne à chaque modification de la table Borders.
 4. Créer un déclencheur nommé Pas_de_modification_soir_et_we qui rend impossible de modifier la table organization en dehors des heures de bureau (lundi-vendredi, 9h00-17h00). Différenciez les messages selon le type de modification (insertion, suppression, mise à jour).
 5. Créer une table audit_modif_organisation (idModif, date, utilisateur, type_modification, ancien_abbreviation, nouvelle_abbreviation) pour tracer toutes les modifications de la table organization. Ecrivez le trigger TAudit_modif_resultat qui insère un tuple dans la table audit_modif_resultat à chaque modification de résultat. Indication : utiliser la variable user ; utiliser une séquence pour gérer l'identifiant de modification.
-