

TP 1 SQL et PL/SQL (2 séances)

Les outils

Vous disposez tous de comptes individuels sur un serveur Oracle.

ATTENTION, les logins sont identiques à votre login INSA, mais les mots de passe ne sont pas gérés par le CAS INSA. C'est à vous des les gérer à la main.

IMPORTANT : 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 :

1. en mode console avec l'application **sqlplus**
2. en mode interface homme/machine avec l'application **SQL_Developer**

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

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 (créez le si nécessaire - vous devez y avoir les droits d'écriture).
- 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) - communiqué par vos enseignants lors du TP.
 - **nomServeur** est le nom DNS de la machine supportant Oracle, ici **if-Oracle11**
 - **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 **DB11**

La commande de connexion doit donc ressembler à

`sqlplus tdupont/kDi3aR23!@//if-Oracle11:1521/DB11`

Si pour une raison, qui peut être raisonnable, vous ne souhaitez pas que votre mot de passe apparaisse en clair lors de votre connexion, vous pouvez utiliser la commande

`sqlplus <loginOracle>@'//<machine>:<port>/<Instance>/'`

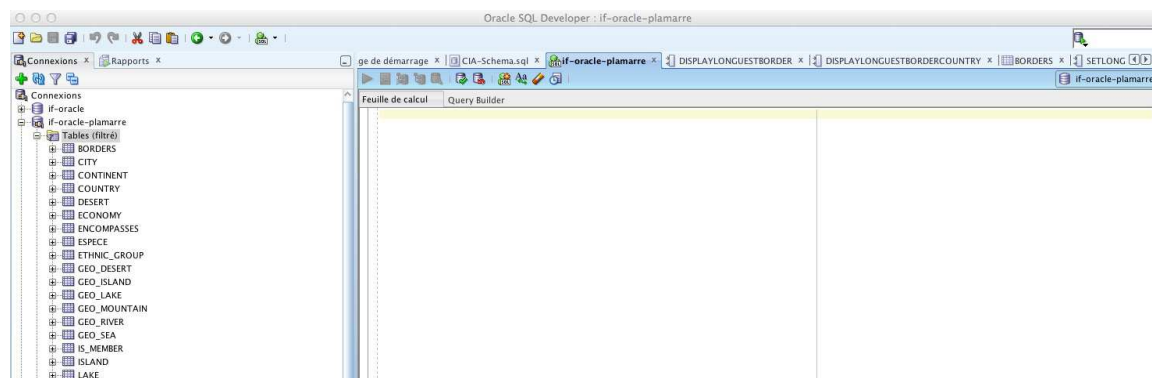
soit `sqlplus tdupont@'//if-Oracle11:1521/DB11/'`

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

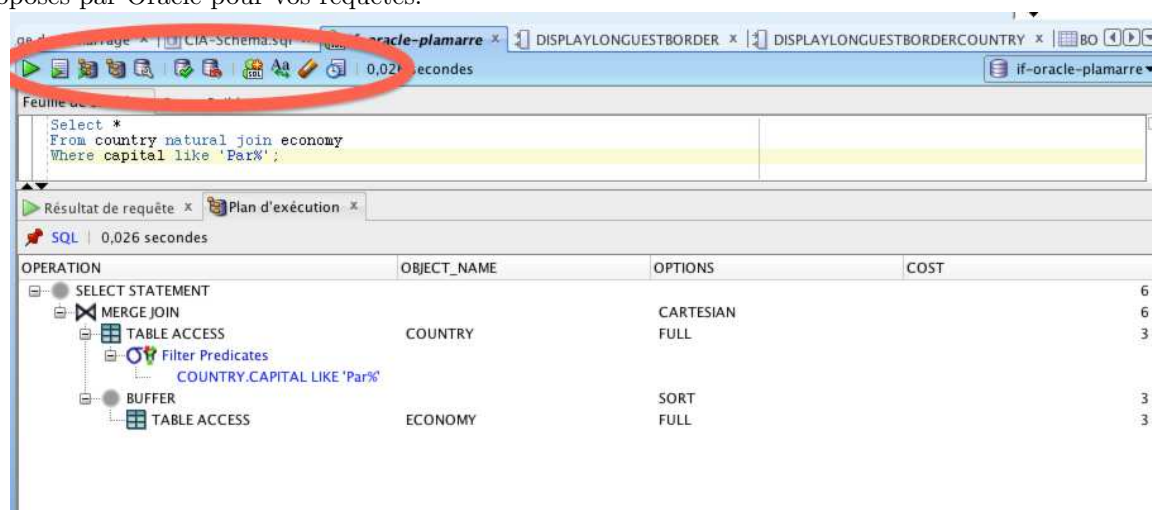
Pour travailler avec une interface graphique

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. La maîtrise de l'interface texte reste préconisée car elles sera la seule à rester utilisable dans des situations tendues.

Pour utiliser le client lourd, lancez l'application **SQL_Developer** qui doit être présente sur vos postes. Configurez l'application pour qu'elle se connecte à votre base de données en utilisant les informations données dans la section précédente. Vous pourrez alors travailler sur 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.



Bonne visite.

Instructions générales

Choisissez le compte d'un des membres du binôme pour travailler et connectez-vous à ce compte. Durant le TP, tenez à jour un cahier (numérique) dans lequel pour chaque exercice vous noterez la question, les problèmes rencontrés et les solutions que vous avez proposées. Cela vous sera utile à la fois pour vos révisions, votre évaluation de TP et votre évaluation de DS.

Jeu de données.

Vous avez à votre disposition la base de données contenant des informations sur tous les pays du monde et toutes les question de cd TP, ainsi que l'évaluation sur machine de la troisième séance, porteront sur ce jeu de données.

ATTENTION, cette base N'a PAS été créée pour ce TP, ni à des fins d'enseignements. C'est une base qui est mise à disposition sur le site <http://www.dbis.informatik.uni-goettingen.de/Mondial/> où vous pouvez la télécharger sous différents formats. Elle résulte d'une compilation, intégration, de plusieurs sources et a été mise à jour fin 2017. A priori, nous utiliserons donc la dernière version. Pour éviter toute divergence de version (le site peut être mis à jour indépendamment du TP, et quelques modifications ont peut-être été effectuées pour des raisons pédagogiques), vous devez télécharger la version qui est disponible sur le dossier du cours Moodle.

Bien que cette base soit largement utilisée en enseignement, il n'en demeure pas moins qu'elle ne répond pas toujours aux normes de qualité que l'on serait en droit d'attendre (et que l'on exigera de vous) que ce soit pour la base ou sa documentation. L'avantage est que vous êtes confrontés à une situation assez proche de ce que l'on rencontre dans la réalité. Soyez attentifs, prenez du recul, et analyser les problèmes avec un esprit critique !

Attention, dans ce TP, vous avez un rôle de gestionnaire de données Ce n'est pas vous qui avez la vérité terrain pour déterminer les données qui doivent être présentes ou non dans la base. Ne sortez pas de votre rôle et ne modifiez ni le schéma ni les données sans l'accord explicite des utilisateurs (ici, les enseignants de TP).

Sur le répertoire Moodle du module, vous trouverez plusieurs fichiers correspondant aux jeu de données lui même (trois fichiers sql correspondant respectivement à la création du schéma, l'insertion des données, et la suppression des tables) ainsi que plusieurs fichiers de documentation (type pdf).

Installation de la base de données sur votre compte

Vous devez récupérer les fichiers de création de la base (un fichier pour la création du schéma et un fichier pour l'insertion des données) sur le répertoire Moodle du module. Enregistrez les fichiers sur votre compte système dans un répertoire prévu à cet effet.

Connectez vous à l'instance oracle en utilisant, à votre convenance, en mode console ou en mode graphique. La commande pour exécuter un script sql est

@<nom_de_fichier>

Si le nom de fichier doit être protégé (par exemple s'il contient des espaces), utilisez des guillemets doubles. Attention, à bien donner le nom absolu de votre fichier.

Exemple : @"/Utilisateurs/Mon Rep principal/BDR/Mondial/script.sql"

Problèmes.

Problème 1.

Pour garantir la qualité des données, on souhaite que le SGBD effectue des vérifications lors des opérations d'ajout, de modification ou de suppression. Les éléments que l'on cherche à garantir sont décrits ci-dessous. Pour chacun de ces éléments, si le schéma actuel ne l'intègre pas déjà, proposez une modification permettant de le garantir. Pensez aussi aux tests permettant de vérifier que votre solution fonctionne correctement.

Si plusieurs solutions s'offrent à vous, listez les différentes possibilités avec leurs différences et sollicitez la personne "métier" (ici l'enseignant) pour faire un choix.

Si, lors de la mise en place de certaines contraintes, vous rencontrez des problèmes liés aux données, identifiez les données responsables du problème avant de soumettre le problème clairement formulé à la personne "métier" pour qu'elle puisse vous aider à prendre la bonne décision (erreur de contrainte ou mauvaise formulation ; données erronées).

Question 1.1. La capitale d'un pays doit être une ville connue, c.-à-d. présente dans la table CITY.

Question 1.2. Tous les pays doivent avoir des noms différents les uns des autres. En d'autres termes, il n'est pas envisageable d'avoir deux pays portant le même nom.

Question 1.3. Le nom d'un pays doit être connu. Il n'est pas possible d'avoir un pays sans nom, ou plus précisément dont le nom est inconnu.

Question 1.4. La capitale d'une province doit être une ville de la table CITY.

Question 1.5. Le pays d'une province doit être un pays de la table COUNTRY.

Question 1.6. Le pays d'une ville doit être un pays de la table COUNTRY.

Question 1.7. La province d'une ville doit être une province de la table PROVINCE.

Question 1.8. La longueur d'une frontière ne peut qu'être strictement positive (sinon, il n'y a pas de frontière).

Question 1.9. Deux villes ne peuvent avoir les mêmes coordonnées (à la fois même longitude et même latitude).

Problème 2.

Requêtes

Question 2.1. Lister tous les pays par ordre alphabétique de leur nom.

Question 2.2. Donner la (ou les) organisation(s) qui a(ont) le plus grand nombre de membres.

Question 2.3. Donner la liste des pays triés par ordre de pourcentage de leur population citadine.

Question 2.4. Donner la liste des continents par ordre croissant de leur population.

Question 2.5. Lister les pays présentant la plus longue frontière. On considérera ici la frontière totale du pays (c.-à-d. la somme de toutes ses frontières avec tous les autres pays).

Question 2.6. Lister les noms de villes qui apparaissent dans plusieurs pays, avec le nombre d'occurrences. La liste doit être triée par ordre décroissant du nombre d'apparitions, puis en cas d'égalité, par ordre alphabétique des villes.

Question 2.7. Chercher les pays (code et nom) dont on ne connaît pas, dans la base, la date d'indépendance.

Question 2.8. Lister les pays (code et nom) qui ne sont membre aucune organisation.

Question 2.9. Pour tous les pays, lister les organisations dont ils sont membres. ATTENTION, demande spécifique : tous les pays doivent apparaître dans la liste, même s'ils ne sont membres d'aucune organisation.

Question 2.10. Donner la longueur totale des frontières sur chaque continent. Note : une frontière est considérée comme étant sur un continent seulement si les deux pays concernés sont sur le continent en question.

Question 2.11. Sur chaque continent, donner les longueurs moyennes des frontières entre deux pays. On ne compte ici que les frontières entre pays du même continent.

Question 2.12. Lister les villes qui sont le siège d'au moins une organisation.

Question 2.13. Donner la moyenne du nombre d'organisations siégeant dans une ville pour les villes accueillant au moins un siège d'organisation.

Question 2.14. Pour les cinq langues parlées par le plus grand nombre de personnes, afficher la langue et le nombre de personnes qui la parlent.

Question 2.15. Pour chacune des cinq langues parlées dans le plus de pays, afficher la langue et les pays où elle est parlée.

Question 2.16. Écrire en PL/SQL une solution qui prend en paramètre un code de pays et qui fournit son nom ainsi que le nombre de pays frontaliers. Pour tester cette solution, écrivez une procédure de test qui peut par exemple afficher les résultats obtenus sur l'écran du terminal.

Question 2.17. Modifier la table country en lui rajoutant une colonne (frontiereLongue). Cette colonne doit contenir le code du pays avec laquelle le pays donné a la frontière la plus longue.

Renseigner cette colonne. Pour cela, 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 sélectionne un arbitrairement.

Question 2.18. Écrire en PL/SQL du code qui fournit la liste des pays ayant acquis leur indépendance un lundi (utiliser to_char pour convertir la date).

Question 2.19. 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.

Question 2.20. Pour un code pays passé en paramètre, faire un récapitulatif. Ce récapitulatif devra mentionner : 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.

Question 2.21. Mesure pour les informations manquantes (optionnelle). 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.

Question 2.22. Utilisation de la mesure précédente (optionnelle). En utilisant la mesure définie pour la question précédente, donner les 10 pays ayant le moins d'informations manquantes dans la base. (optionnelle)

Question 2.23. En partant de la mer Méditerranée, quels sont les mers (ou océans) que l'on peut atteindre en naviguant ?

Question 2.24. 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)

Droits d'accès

Question 2.25. 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.

Contraintes d'intégrité par déclencheurs.

ATTENTION, le sujet indique clairement que certaines contraintes sont implantées grâce à des contraintes SQL et d'autres par des déclencheurs. Dans la réalité, et pour les évaluations, ce sera à vous de déterminer quel est l'outil le mieux adapté. Ici, il vous est demandé explicitement de répondre en utilisant un déclencheur. Cependant, indiquez clairement si la même contrainte peut être réalisée par une contrainte SQL. Si non, expliquez pourquoi. Si oui, précisez la méthode qui serait le mieux adaptée.

Question 2.26. Sur la table Politics, la date d'indépendance doit être inférieure ou égale à la date du jour.

Question 2.27. Le code d'un pays doit être alphabétique et saisi en majuscule. Analysez cette contrainte. Peut-elle être mise en œuvre par une contrainte SQL ? Si oui, y a-t-il un avantage à la traiter avec un déclencheur ?

Traitement automatique des données

Question 2.28. 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.

Question 2.29. 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).

Question 2.30. 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.

Modification des données

Question 2.31. Division d'un pays. Suite à des mouvements politiques, un pays, dont le nom doit encore rester secret, nous dirons donc le pays P, doit être divisé en deux (P2 et P3). Proposez à une méthodologie vous permettant d'effectuer cette modification. A l'issue de vos modifications, il ne doit plus y avoir dans la base que les pays P1 et P2, le pays P ayant été supprimé. Attention aux informations : les villes, provinces, montagnes, rivières, lacs...devront peut être changer de noms, mais ne doivent pas disparaître.
