



HAI502I - TRAVAUX PRATIQUES 1

RAPPELS - SQL

Objectifs du TP :

L'objet de ce TP est de se « remettre dans le bain » avec ORACLE et de répondre à un ensemble de requêtes SQL.

Soit le schéma relationnel exemple suivant qui illustre une base de données dans laquelle des voisins peuvent se rendre un certain nombre de services. Les clés primaires sont en gras souligné et les clés étrangères en italique :

VOISIN (**IDVOISIN**, NOM, PRENOM, VILLE, RUE, CODEPOSTAL)

SERVICE (**IDSERVICE**, LIBELLE)

TRAVAIL (*IDSERV*, *IDVOIS*, **DATESERVICE**, PRIX)

où IDSERV référence IDSERVICE et est défini dans le domaine NUMERIC(3,0), IDVOIS référence IDVOISIN et est défini dans le domaine NUMERIC(3,0), DATESERVICE est défini dans le domaine DATE et il ne peut y avoir qu'un seul service réalisé par un voisin un jour donné, PRIX est défini dans le domaine NUMERIC(3,0).

Création des tables

Pour faciliter le travail les relations et les tuples sont fournis. Récupérer et sauvegarder les fichiers suivants dans le cours sur MOODLE dans le répertoire du TP1 :

- `creationSERVICE.sql`
- `remplissageSERVICE.sql`
- `mesresultatsrequetesTP1.sql`

Rappels :

- Vérifier dans SAPIENS (<https://sapiens.umontpellier.fr/>), après vous être connecté avec votre login et mot de passe usuel, que l'instance ORACLE est bien disponible sur votre compte (menu « Mon Compte »).
- Penser à récupérer son identifiant de la forme e00000....
- Dans une fenêtre, taper la commande suivante pour lancer SQLPlus :

```
rlwrap sqlplus e00000.../motdepasse@oracle.etu.umontpellier.fr:1524/pllicence
```

Exemple (pour un étudiant identifié par e2019012531 avec comme mot de passe ASEDRF3a) :

```
rlwrap sqlplus e2019012531/ASEDRF3a@oracle.etu.umontpellier.fr:1524/pllicence
```

Création de la base :

- Une fois SQLPlus lancé, vous devez créer la base de données. Pour cela, tapez après le prompt SQL>, l'instruction suivante : `@creationSERVICE.sql` (ou bien `start creationSERVICE.sql`)
- Une fois la base créée, vous pouvez insérer un ensemble de tuple en tapant après le prompt SQL>, l'instruction suivante : `@remplissageSERVICE.sql` (ou bien `start remplissageSERVICE.sql`)

Attention : les fichiers `creationSERVICE.sql` et `remplissageSERVICE.sql` doivent être dans le même répertoire que celui où vous avez lancé SQLPlus.



Vous utiliserez l'interface d'accès SQLPlus qui permet l'exécution de commandes spécifiques et de commandes SQL. Chaque machine sous linux possède un client SQLPLUS (interpréteur de commandes SQL et module d'Oracle). Comme l'éditeur de commandes sous SQLPLUS est "ligne à ligne", il est conseillé d'utiliser un éditeur externe (par exemple gedit) pour effectuer et sauvegarder l'ensemble du travail demandé.

Après le signe d'invite `SQL>`, taper les commandes SQL. Les minuscules ne sont pas différenciées des majuscules pour ce qui concerne la syntaxe réservée.

L'exécutable `SQLPLUS` est un interpréteur/éditeur de commandes. Il peut interpréter plusieurs types de commandes : les commandes propres à `SQLPLUS` et les commandes SQL (et PL/SQL).

Les commandes propres à `SQLPLUS` ne sont pas terminées par un ";" et possèdent des abréviations (mises entre parenthèses).

Rappel : Quelques commandes rapides à connaître sur `SQL*Plus` :

- `SET pause ON/OFF` défilement page par page
- `SET LINESIZE n` fixe le nombre de caractères de la ligne
- `SET PAGESIZE n` fixe le nombre de lignes par page
- `DESC nomtable` signifie `DESCRIBE` en anglais et permet d'avoir la description de la table `nomtable`.
- `HOST (!)` pour sortir temporairement de `SQLPLUS` et pouvoir exécuter des commandes Unix ; on retourne à `SQLPLUS` par la commande `EXIT`.
- `SPOOL <nomfichier>` pour enregistrer la trace d'une session dans un fichier `<nomfichier>.lst`
- `SPOOL OFF`
- `QUIT/EXIT` pour quitter `SQLPLUS`.

Les commandes SQL peuvent être saisies sur une ou plusieurs lignes, le point-virgule obligatoire termine la commande. Attention à la syntaxe qui est très rigoureuse. Attention aux parenthèses, virgules et surtout au point-virgule qui termine une commande.

Répondez en SQL aux différentes questions suivantes. Les requêtes sont à écrire dans le fichier `mesresultatsrequetesTP1.sql`.

Ce fichier contient l'ensemble des requêtes à écrire, les résultats à obtenir ainsi que des éléments de mise en page.

Pour exécuter les requêtes : `@ mesresultatsrequetesTP1.sql` (ou `start mesresultatsrequetesTP1.sql`).

Q1 - Quels sont les différents services (LIBELLE) proposés ?

Q2 - Quels sont les services (LIBELLES, DATESERVICE) qui ont été utilisés par le voisin DURAND PAUL (en majuscule) ?

Q3 - Quel est le prix moyen et le prix maximum des services ?

Q4 – Quels sont les voisins qui n'ont pas de ville ?



Q5 - Quels sont les voisins (NOM, PRENOM) qui ont eu au moins un service dont le prix est supérieur au prix moyen des services ?

Q6 - Insérer dans la relation SERVICE, deux nouveaux tuples dont les identifiants (IDSERVICE) sont respectivement 6 et 7 et les libellés (LIBELLE) 'JARDINAGE' et 'MENAGE' (en majuscule).

Q7 - Quels sont les services (LIBELLE) qui ne sont proposés par aucun voisin ?

Q8 - Quels sont les différents services (LIBELLE) proposés ? Cette requête est identique à la Q1 mais proposer 2 solutions différentes pour ne pas voir apparaître 2 fois le libellé JARDINAGE

Q9 - Supprimer les deux derniers tuples insérés précédemment dans la relation SERVICE (C.f. question Q6)

Q10 - Quels sont les voisins (NOM, PRENOM) qui ont fait le plus de prestations ?

Q11 - Quels sont les services (LIBELLE) facturés les plus chers (PRIX) mais par catégorie de service pour chaque voisin (NOM, PRENOM) ?

Par exemple, si Dupond Pierre fait deux fois du Babysitting pour 200 euros et 100 euros puis du Jardinage pour 80 euros, le résultat attendu est :

Babysitting Dupond Pierre 200

Jardinage Dupond Pierre 80

Q12 - Quels sont les voisins (NOM, PRENOM) qui ont fait tous les services ?