

Fiche de TP N°1- Introduction :

Oracle-Docker, SQLPlus, SQL Developer, PL/SQL

1. Installation (Oracle Database)

i. Sur docker (Recommandé)

- **docker-desktop** : <https://www.docker.com/products/docker-desktop>
- **WLS: Windows pour Linux** → **wsl_update_x64.msi**
 - https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi
- **In CMD:** Run `bcdedit /set hypervisorlaunchtype auto`

ii. Sur la machine Host :

- Pour installer **Oracle Database 12c/19c** vous devez être connectés en tant qu'administrateur sur votre ordinateur. Le SGBD d'oracle peut être téléchargé à partir de la page officielle d'oracle.
- https://www.youtube.com/watch?v=4y1WRF_4oFQ un lien vers une vidéo qui vous montre comment installer Oracle **12c**.

2. Lancement d'une instance Oracle à travers docker-compose:

- i. Copier le fichier « **docker-compose.yml** » dans un répertoire nommé **OracleTP**
- ii. Exécuter la commande « **docker-compose up -d** » dans le répertoire **OracleTP**

3. SQL*PLUS

- i. Ouvrir un terminal et exécuter la commande suivante :
 - **docker exec -it oracledb /bin/sh**
 - **-> sqlplus**
- ii. Connecter en tant qu'administrateur, en tapant la commande **sys as sysdba;**
- iii. Créer un nouvel utilisateur en tapant : **create user ali identified by aaa ;**
- iv. Affectation des droits : **grant all privileges to ali ;**
 - PS : si vous n'arrivez pas à récupérer votre user, exécutez la commande suivante :
alter system set "_common_user_prefix"="" scope=spfile;
- v. Connecter avec l'utilisateur **ali** : **connect ali**, ensuite vous tapez son mot de passe « **aaa** »
- vi. Taper **show user** ; pour savoir avec quel utilisateur vous êtes connectés
- vii. Créer une table « **etudiant(id,nom)** » et essayer d'insérer certains tuples
- viii. Ajouter **commit** ; pour valider vos insertions (sinon **rollback** pour annuler)
- ix. Créer un autre utilisateur nommé « **ahmed** », et vérifier si vous pouvez créer la même table « **etudiant** » en connectant avec l'user « **ahmed** »

- x. A partir de l'utilisateur « **ahmed** » afficher le contenu de la table « **etudiant** » de l'utilisateur « **ali** » (ou du schéma de ali) : **select * from ali.etudiant ;**

4. Auto-Increment dans oracle : Dans oracle, les attributs **auto-increment** sont gérés par des séquences.

- Soit la table client : **CREATE TABLE client (id int PRIMARY KEY, nom varchar(45))**
- Pour rendre la colonne **id** auto-increment, il faut créer une séquence :

create sequence client_seq

INCREMENT BY 1 START WITH 1 NOMAXVALUE NOCYCLE CACHE 10;

- Ensuite, pour chaque nouvelle insertion l'**id** du nouveau client est calculé en utilisant l'instruction : **client_seq.nextval**

insert into client values(client_seq.nextval, 'Ahmed');

insert into client values(client_seq.nextval, 'Ali'); commit;

Id	nom
1	ahmed
2	ali

5. PL/SQL: (sur SQL Developer <https://www.oracle.com/tools/downloads/sqldev-downloads.html>, Paramètres de connexion : [User: system ou ali, Password: aaa, Nom d'hôte: localhost, Port : 1521, SID : xe])

Soit le schéma relationnel de la base de données PRODUIT (exécuter le fichier **creation_script.sql**)

PRODUIT (IDP, designation, prixUnitaire, #IDCategorie) ;

CATEGORIE (IDCategorie, nom)

Partie 1 : Le prix unitaire dans la table produit est exprimé en Dinars. On souhaite vendre ces produits à l'étranger, par conséquent, nous devons construire, à partir de la table **PRODUIT**, une autre table **PRODUIT_Etranger** telle que :

- Les deux tables ont la même structure ;
- La désignation des produits soit écrite en majuscules ;
- Le prix unitaire en Dinars des produits soit converti en Dollars.
- Si la table PRODUIT est vide, la table PRODUIT_Etranger devra contenir uniquement le tuple (0, 'Pas de produit', NULL, NULL).
- Si un prix de la table produit est NULL, son prix en Dollars doit être 0.

- Ecrivez le code **PL/SQL** qui permet de remplir la table PRODUIT_Etranger

NB: Utiliser les fonctions d'oracle : UPPER, TRUNC et NVL.

Partie 2 (Trigger):

Soit la table **ProduitHistory** (user, idP, operation_date, operation_type).

- Ecrire un trigger permettant, après l'exécution de chaque opération sur la table **Produit**, d'ajouter un tuple concernant cette opération dans la table **ProduitHistory** tout en précisant l'utilisateur est la date de l'opération.