

# Module: OSI, INTERNET ET PROGRAMMATION WEB

## TP 5 – Annexe- Utilisation d'ORACLE

**Objectif** : Base de Donnée sous ORACLE et PHP

**Correction** : la correction sera diffusée la semaine qui suit : refaire les exercices avec la correction

**Note** : vous créez sur linserv1.unice.fr un répertoire ~/web/TP5/oracle dans lequel vous placerez vos fichiers html et php ainsi qu'une page index.php

**Note** : Vous manipulerez vos bases de données avec ORACLE sur quad.unice.fr - 134.59.22.8

En ce qui concerne l'utilisation :

- soit vous utilisez l'interface SQLPLUS installée sur chaque poste (attention : la chaîne de connexion est : TDINFO)
- soit vous utilisez l'interface ISQLPLUS depuis un navigateur quelconque (fortement conseillé) : il suffit de charger l'application depuis le lien : <http://quad.unice.fr:5560/isqlplus/>

et entrez le login et password du compte Oracle qui vous est fourni.

## 1. Créer une table sous Oracle

1.1. Dans une fenêtre **telnet** sur quad.unice.fr, ouvrir une session SQL\*Plus sous Oracle avec la commande :

```
$ sqlplus  
saisir votre nom d'utilisateur et votre mot de passe oracle
```

1.2. Sous SQL\*Plus changer votre mot de passe oracle avec la commande :

```
SQL> GRANT CONNECT TO login IDENTIFIED BY passwd ;
```

1.3. Dans un bloc note rédiger la requête SQL permettant de créer une table ANNUAIRE donnant les numéros de poste téléphonique du personnel en fonction de leurs noms et prénoms.

- La table comportera trois champs : nom, prénom, numéro de poste, déclarés en VARCHAR2 (*taille*)
- Les numéros de poste sont de la forme 99.99
- La clé primaire sera composée des deux champs nom et prénom
- Les champs nom et prénom seront déclarés NOT NULL

1.4. A l'aide d'un Copier/Coller, introduire la requête SQL sous SQL\*Plus et l'exécuter.

1.5. Vérifier la structure de votre table ANNUAIRE sous SQL\*Plus

1.6. Insérer quelques tuples dans votre table ANNUAIRE, tester.

1.7. Donner le droit de consultation publique sur votre table ANNUAIRE

## 2. Créer les pages PHP pour consulter et mettre à jour la table annuaire

2.1. Créer un script php permettant d'effectuer à partir de pages web, les opérations suivantes :

1. Rechercher une entrée dans la table annuaire à partir des premières lettres du nom et/ou du prénom
2. Ajouter une nouvelle entrée dans l'annuaire
3. Modifier un numéro de poste
4. Supprimer une entrée de l'annuaire

Vous utiliserez une feuille de style pour la présentation des pages web.

### 3. Créer des menus déroulants à partir de données puisées dans des tables

Nous considérons les deux relations suivantes :

- CATEGORIE (id\_cat, designation, taux\_tva)  
une ligne représente une catégorie de produit (clé id\_cat)

- PRODUIT (id\_pro, id\_cat, designation, marque, prix\_uht, qstock)  
une ligne représente un produit (clé id\_pro)  
un produit appartient à une seule catégorie (clé étrangère id\_cat référence une catégorie)

Un jeu d'essai est créé sous oracle dans le compte ET (le script ayant permis de créer les tables et le jeu d'essai est donné en Annexe2). Les tables sont accessibles en lecture pour tous les utilisateurs d'oracle.

Nous voulons créer une page comportant trois cadres comme dans l'exemple ci-dessous :

Ordinateurs ▼	mac ▼
<b>ID: 20</b> <b>Désignation: mac</b> <b>Marque: apple</b> <b>Prix: 7,500.00</b> <b>Qte stock: 10</b>	

- le cadre en haut à gauche contient un menu permettant de sélectionner une catégorie
- le cadre en haut à droite contient un deuxième menu permettant de sélectionner un produit de la catégorie qui a été sélectionnée à gauche
- le cadre en bas affiche le produit sélectionné

3.1. Créer le script menu.php qui découpe la fenêtre en trois cadres.

3.2. Créer le script menu1.php qui affiche dans le cadre en haut à gauche le menu permettant de sélectionner une catégorie.

Lorsque l'utilisateur sélectionne une catégorie, une fonction javascript permettra d'envoyer au serveur une requête menu2.php avec en paramètre l'identifiant de la catégorie sélectionnée. Ce script menu2.php renverra dans le cadre à droite le menu permettant de sélectionner un produit de la catégorie passée en paramètre.

3.3. Créer le script menu2.php qui renvoie dans le cadre à droite le menu permettant de sélectionner un produit de la catégorie qui est passée en paramètre.

Lorsque l'utilisateur sélectionne un produit, une fonction javascript permettra d'envoyer au serveur une requête fiche.php avec en paramètre l'identifiant du produit sélectionné. Ce script fiche.php renverra dans le cadre en bas la fiche du produit sélectionné.

3.4. Créer le script fiche.php qui renvoie dans le cadre en bas la fiche du produit sélectionné.

## 4. Annexe1 – Fonctions OCI (*Oracle Call Interface*) de PHP

### Etablit une connexion à un serveur Oracle

resource ocilogon (string **username**, string **password**, string **db** )

ocilogon() retourne un identifiant de connexion, nécessaire à la plus part des fonctions OCI. Si l'option ORACLE\_SID n'est pas précisée, PHP utilisera la variable d'environnement ORACLE\_SID pour déterminer le serveur de connexion.

Les connexions sont partagées, à l'intérieur d'une même page avec ocilogon(). Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions.

### Déconnexion d'un serveur Oracle

int ocilogoff (resource **connection**)

ocilogoff() ferme la connexion Oracle.

### Analyse une requête

int ociparse (ocifreedesc **conn**, string **query**)

ociparse() analyse la requête **query** sur la connexion **conn**, et retourne TRUE si la requête **query** est valide, et FALSE, si ce n'est pas le cas. **query** peut être n'importe quelle requête SQL.

### Exécute une commande

int ociexecute (resource **statement**, int **mode** )

ociexecute() exécute une commande déjà préparée (voir ociparse()). L'option **mode** vous permet de spécifier le mode d'exécution (par défaut, il est à OCI\_COMMIT\_ON\_SUCCESS). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI\_DEFAULT.

### Retourne le nombre de lignes affectées

int ocirowcount (resource **statement**)

ocirowcount() retourne le nombre de lignes affectées par une commande de modification. Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées.

### Retourne la valeur d'une colonne dans une ligne lue

mixed ocireult (resource **statement**, mixed **column**)

ocireult() retourne les données de la colonne **column** dans la ligne courante (voir ocifetch()).  
ocifetch() retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILES).

### Modifie la prochaine ligne dans le pointeur interne de résultat.

int ocifetch (resource **statement**)

ocifetch() place la prochaine ligne (d'une commande SELECT) dans le pointeur interne de résultat.

### Retourne la ligne suivante dans un tableau

int ocifetchinto (resource **stmt**, array **&result**, int **mode** )

ocifetchinto() retourne la ligne suivante (pour une commande SELECT) dans le tableau **result**. ocifetchinto() écrasera le contenu de **result**. Par défaut, **result** sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL.

L'option **mode** vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI\_ASSOC+OCI\_RETURN\_NULLS). Les modes valides sont :

- OCI\_ASSOC Retourne un tableau associatif.
- OCI\_NUM Retourne un tableau à index numérique (DEFAULT, valeur par défaut)
- OCI\_RETURN\_NULLS Retourne les colonnes vides.

- OCI\_RETURN\_LOBS Retourne la valeur des objets LOB plutôt que leur descripteur.

**Retourne toutes les lignes d'un résultat**

int ocifetchstatement (resource **stmt**, array &**variable**)

ocifetchstatement() retourne toutes les lignes d'un résultat dans le tableau **variable**. ocifetchstatement() retourne le nombre de lignes retournées.

**Teste si la valeur d'une colonne est NULL**

int ocicolumnisnull (resource **stmt**, mixed **column**)

ocicolumnisnull() retourne TRUE si la colonne **col** du résultat **stmt** est NULL. Vous pouvez utiliser le numéro de colonne (l'indexation des colonnes commence à 1) ou le nom de la colonne, pour le paramètre **col**.

**Valide les transactions en cours**

int ocicommit (resource **connection**)

ocicommit() valide toutes les transactions en cours sur la connexion Oracle **connection**.

**Annule les transactions en cours**

int ocirollback (resource **connection**)

ocirollback() annule les transactions en cours sur la connexion Oracle **connection**.

**Libère toutes les ressources occupées par une commande.**

int ocifreestatement (resource **stmt**)

ocifreestatement() retourne TRUE en cas de succès, et FALSE en cas d'échec.

**Retourne la dernière erreur de stmt|conn|global.**

array ocierror (int **stmt|conn|**)

ocierror() retourne la dernière erreur trouvée. Si l'option **stmt|conn** n'est pas fournie, la dernière erreur rencontrée est retournée. Si aucune erreur n'est trouvée, ocierror() retourne FALSE.

## 5. Annexe2 - jeu d'essai (compte ET sous oracle)

```
DROP TABLE produit;
DROP TABLE categorie;

CREATE TABLE categorie
(
  id_cat          VARCHAR2 (2)   PRIMARY KEY ,
  designation     VARCHAR2 (50)  ,
  taux_tva       NUMBER (4,2)
) ;

CREATE TABLE produit
(
  id_pro         VARCHAR2 (8)   PRIMARY KEY ,
  id_cat         VARCHAR (2)   REFERENCES categorie ,
  designation    VARCHAR2 (50) ,
  marque        VARCHAR2 (50) ,
  prix_uht      NUMBER (9,2) ,
  qstock        NUMBER (2)
) ;

/* jeu d'essai */
INSERT INTO categorie VALUES ('C1', 'Ordinateurs', 10) ;
INSERT INTO categorie VALUES ('C2', 'Logiciels', 20) ;
INSERT INTO categorie VALUES ('C3', 'Imprimantes', 30) ;

INSERT INTO produit VALUES ('10', 'C1', 'ps', 'ibm', 500.00, 10);
INSERT INTO produit VALUES ('20', 'C1', 'mac', 'apple', 7500.00, 10);
INSERT INTO produit VALUES ('30', 'C1', 'aptiva', 'ibm', 12000.00, 10);
INSERT INTO produit VALUES ('40', 'C1', 'power mac', 'apple', 2000.00, 10);
INSERT INTO produit VALUES ('50', 'C2', 'word', 'microsoft', 100.00, 10);
INSERT INTO produit VALUES ('60', 'C2', 'access', 'microsoft', 100.00, 10);
INSERT INTO produit VALUES ('70', 'C2', 'paradox', 'borland', 80.00, 30);
```