**Exercice 1 :**

* Afficher à l’aide d’un bloc PL/SQL le message suivant  
  « Premier exercice de PL/SQL effectué à la date du ….date du jour….. »
* Créer un programme PL/SQL qui insère une nouvelle catégorie de produit avec les valeurs suivantes : code =5 et libelle = « composants »
* Valider la transaction

**Exercice 2 :**

* Afficher en utilisant les variables les renseignements utilisateur et la date système. Le nom et le prénom de l’étudiant doivent être définis avec une constante.
* Modifier la valeur de la constante dans le corps du bloc PL/SQL
* Déclarer une nouvelle variable âge avec la contrainte not null : quand faut-il obligatoirement lui assigner une valeur ?

**Exercice 3 :**

* Utiliser les variables de substitution afin de demander l’introduction de deux nombres à l’écran. Les nombres sont récupéré dans des variables.
* Le script aura pour but d’afficher le résultat du calcul : nombre1\* nombre2
* Si le résultat est inférieur à 100, afficher le message ‘trop facile » sinon afficher ‘trop dur !’

**Exercice 4**

**Les variables scalaires**

Réaliser les deux scripts suivants (utiliser les variables de liaison)

* Demander à l’utilisateur d’introduire son nom et deux nombres au choix
* Premier script :  
  Afficher un message de bienvenue à l’utilisateur contenant son nom et les nombres introduits
* Deuxième script :  
  Si les deux nombres sont inférieurs à 100 : afficher « la somme du nombre 1…..et du nombre 2 est ……. »  
  Si le premier nombre est supérieur à 100 et le deuxième, afficher la différence entre les deux nombres.  
  Sinon afficher le produit des deux nombres.
* Réaliser des conversions implicites et des conversions explicites sur les valeurs introduites à l’écran

**Exercice 5**

Explique de façon complète (schéma + commentaires) l’utilité du PL/SQL avec Oracle (TRANSACT/SQL avec SQLSERVER) :

* Qu’est-ce que le PL/SQL
* Pour quelles raisons a-t-on introduit cette extension « programmation » au SQL ?
* Comment intégrer ces « programmes » SQL à un langage de programmation (ex : java ?

**Exercice 6 : les blocs imbriqués**

Réaliser le bloc PLSQL suivant :

* Travailler avec 3 blocs imbriqués
* Utiliser deux variables : la première est une variable de liaison qui contiendra la date système et sera définie dans le bloc principal. La deuxième variable (MaVariable sera redéfinie au niveau de chaque bloc et permettra d’afficher dans quel bloc on se situe.

**Exercice 7**

Réaliser le script permettant de créer :

* Deux types composés « record »
  + Le premier « adresse » contient : rue, numéro, cp, localité »
  + Le deuxième « client » contient : nom du client, adresse
* Déclarer une variable du type client
* Afficher le contenu de cette variable

**Exercice 8**

Réaliser un programme PLSQL permettant à l’utilisateur d’introduire un nombre au choix. Ce dernier permettra au programme d’afficher la table de multiplication par rapport au nombre choisi.

Dans un deuxième temps, le programme devra permettre à l’utilisateur de vérifier ses connaissances par rapport à la table de multiplication.

**Exercice 9**

Ecrire un script permettant :

* De définir un type numérique borné avec deux décimales
* D’introduire un nombre au clavier
* De conserver la valeur de ce nombre en mémoire
* D’affecter la valeur introduite au clavier à la variable bornée
* D’afficher le contenu de cette variable

**Exercice 10**

Ecrire un script permettant :

* De déclarer un tableau de nombres
* A l’aide d’une structure répétitive allant jusque 10, remplir ce tableau : la première colonne contient la valeur de l’indice, la deuxième colonne contient la valeur de l’indice multipliée par 100
* Afficher les valeurs contenues dans le tableau
* Afficher le nombre d’éléments contenus dans le tableau
* Afficher la première et la dernière ligne du tableau
* Supprimer la 5ème ligne du tableau

**Exercice 11**

Ecrire un script permettant :

* De définir une variable de type structure permettant d’insérer un enregistrement dans la table Salaries (travailler avec les variables basées colonnes)
* De créer un tableau de type record (ou structure) à partir de la variable précédente
* Remplir le tableau avec 3 nouveaux salariés
* Insérer le contenu du tableau dans la table salaries (avec structure répétitive)
* Vérifier le contenu de la table salaries
* Annuler la transaction

**Exercice 12**

Idem exercice 11 mais travailler avec les variables basées lignes

**Exercice13**

Le script a pour but d’afficher le nombre d’articles dans la table articles

**Exercice 14**

Afficher le prix hors tva par article et le prix total htva pour la quantité en stock

**Exercice 15**

Quel est le salaire moyen des employés si le type de l’attribut salaire n’est pas connu

**Exercice 16**

Afficher le code catégorie et le libellé de la catégorie d’un article choisi par l’utilisateur ;

Travailler avec des variables de type structure

**Exercice 17**

Afficher le nombre d’articles par catégorie : …. Problème ?

**Exercice 18**

Afficher le nombre d’articles différents concernant chaque commande (numéro de commande et nom du client) si ce nombre est supérieur à 2

**Exercice 19**

Ecrire un script permettant de récupérer toutes les commandes qui sont traitées.

**Exercice 20**

* **Ecrire un script permettant à l’utilisateur d’introduire la désignation d’un article. Le script doit permettre d’afficher les commandes relatives à cet article (numéro de commande, désignation de l’article, total à payer pour cet article)**

**Exercice 21  
  
Créer un bloc PL/SQL permettant :**

* **De mettre à jour un enregistrement de la table commande (le numéro de la cde est introduit par l’utilisateur)**
* **Si la date de la commande est nulle, introduire la date système et modifier l’état de la commande qui devient ‘traitée’**
* **Travailler avec les variables basées**
* **Afficher la commande avant la mise à jour et après la mise à jour**

**Exercice 22: instruction bulk collect et mise à jour  
  
créer un bloc PL/SQL permettant :**

* **De récupérer l’ensemble des enregistrements de la table salarié et d’afficher les renseignements concernant les salariés dont le nom est choisi par l’utilisateur.**
* **De récupérer les numéros de commandes, les taux de tva et les taux de remise et les numéros de client. Afficher ces renseignements pour une commande choisie par l’utilisateur**
* **De modifier cette dernière commande : le taux de tva passe à 21% et si le taux de remise est nul, il passe à 5%**
* **Afficher les renseignements après la modification.**

**Exercice 23**

Écrire un script permettant

* D’introduire le numéro d’un salarié
* De mettre à jour le salaire du salarié (le % d’augmentation est introduit à l’écran)
* Si la mise à jour a été effectuée de récupérer le nouveau salaire
* Travailler avec les variables basées

**Exercice 24**

On demande de modifier le script précédent pour gérer le cas d’une mise à jour non exécutée

**Exercice 25**

On demande d’écrire le script permettant de modifier la commission des vendeurs (diviser le salaire par 3) si cette commission est non nulle.

Si la modification est validée, afficher le nombre de modifications.

**Exercice26 : la clause returning**

On demande d’écrire le script permettant d’introduire un nouveau salarié et d’afficher le nom et le prénom et la fonction du nouveau salarié.

Travailler avec les variables basées (2 façons)

**Exercice 27**

On demande d’écrire le script permettant :

* Mettre à jour **toute** la table Salaries en modifiant les commissions (salaire \* pourcentage)
* Afficher les nouvelles commissions

**Exercice 28 : la clause returning**

* On demande d’écrire le script permettant d’introduire un nouveau salarié et d’afficher le nom et le prénom et la fonction du nouveau salarié.
* Travailler avec les variables basées (2 façons)