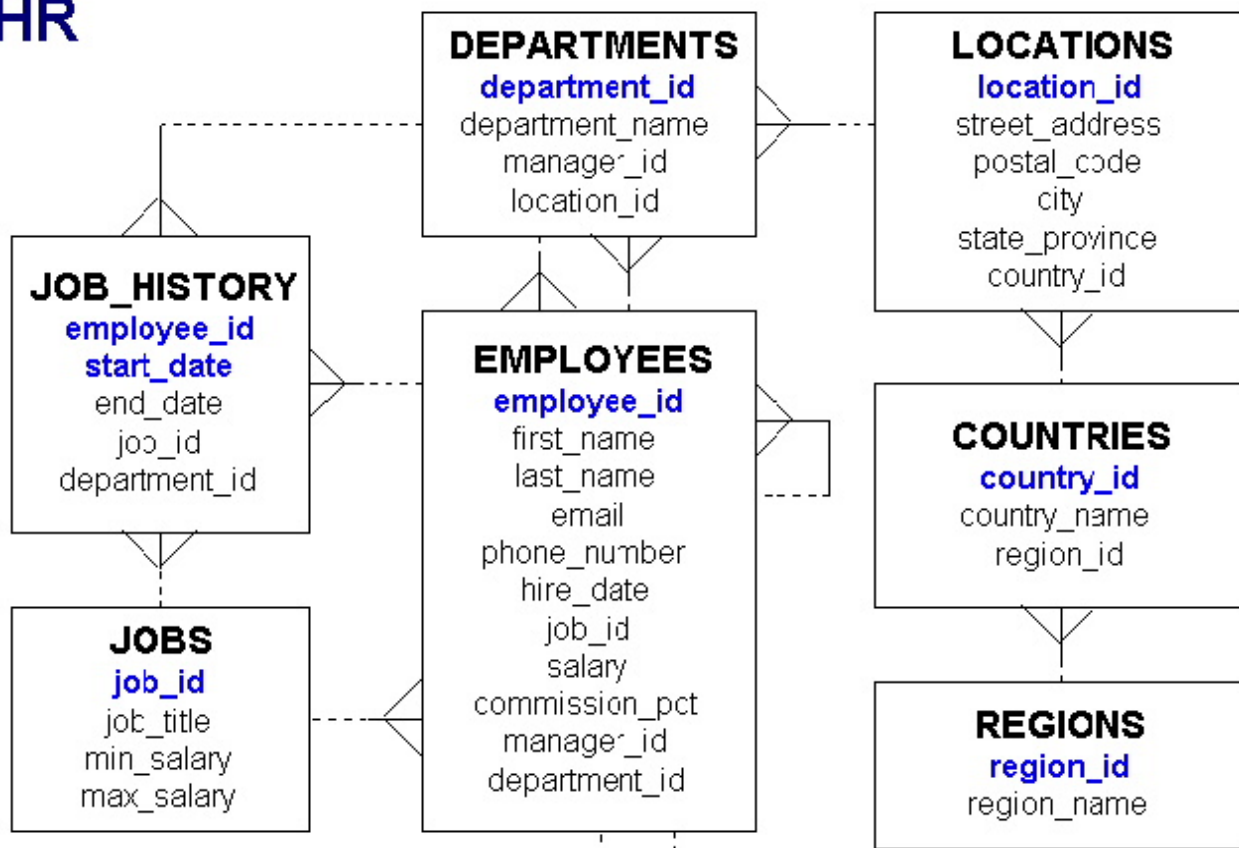


Contrôle Continue N°2 EN PL/SQL

HR



1. Évaluez chacune des déclarations suivantes. Déterminez celles qui ne sont pas valides et expliquez pourquoi.

- DECLARE
v_name,v_dept VARCHAR2(14);
- DECLARE
v_test NUMBER(5);
- DECLARE
V_MAXSALARY NUMBER(7,2) = 5000;
- DECLARE
V_JOINDATE BOOLEAN := SYSDATE;

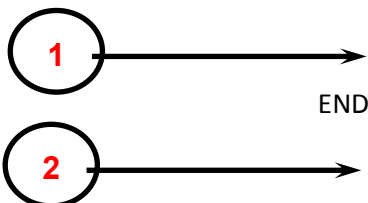
2. Dans chacune des affectations suivantes, déterminez le type de l'expression obtenue.

- v_email := v_firstname || to_char(v_empno);
- v_confirm := to_date('20-JAN-1999', 'DD-MON-YYYY');
- v_sal := (1000*12) + 500
- v_test := FALSE;
- v_temp := v_temp1 < (v_temp2/ 3);
- v_var := sysdate;

3. DECLARE

```
v_custid      NUMBER(4) := 1600;
v_custname    VARCHAR2(300) := 'Women Sports Club';
v_new_custid  NUMBER(3) := 500;

BEGIN
  DECLARE
    v_custid      NUMBER(4) := 0;
    v_custname    VARCHAR2(300) := 'Shape up Sports Club';
    v_new_custid  NUMBER(3) := 300;
    v_new_custname VARCHAR2(300) := 'Jansports Club';
  BEGIN
    v_custid := v_new_custid;
    v_custname := v_custname || ' ' || v_new_custname;
  END;
END;
/
```



Observez le bloc PL/SQL ci-dessus et déterminez le type de données et la valeur de chacune des variables suivantes d'après les règles de portée :

- La valeur de V_CUSTID à la position 1 est :
- La valeur de V_CUSTNAME à la position 1 est :
- La valeur de V_NEW_CUSTID à la position 2 est :
- La valeur de V_NEW_CUSTNAME à la position 1 est :
- La valeur de V_CUSTID à la position 2 est :
- La valeur de V_CUSTNAME à la position 2 est :

4. Ecrivez un bloc PL/SQL qui accepte une année en tant que donnée d'entrée, et qui vérifie s'il s'agit d'une année bissextile. Par exemple, si l'année entrée est 1990, le résultat doit être "1990 is not a leap year".

Indice : L'année doit être divisible par 4, mais pas par 100, ou bien elle doit l'être par 400.

Testez la solution en utilisant les années suivantes :

1990	Année non bissextile
2000	Année bissextile
1996	Année bissextile
1886	Année non bissextile
1992	Année bissextile
1824	Année bissextile

5. a. Pour réaliser l'exercice suivant, vous devez disposer d'une table temporaire où seront stockés les résultats. Vous pouvez la créer vous-même ou exécuter le script labAp_05.sql pour le faire à votre place. Créez une table nommée TEMP, qui comporte les trois colonnes suivantes :

b. Ecrivez un bloc PL/SQL qui contienne deux variables, MESSAGE et DATE_WRITTEN. Déclarez MESSAGE en tant que type de données VARCHAR2 d'une longueur de 35 caractères et DATE_WRITTEN en tant que type de données DATE. Affectez les valeurs suivantes aux variables :

Variable	Contenu
MESSAGE	'This is my first PL/SQL program'
DATE_WRITTEN	Date en cours

Stockez les valeurs dans les colonnes appropriées de la table TEMP. Vérifiez les résultats en interrogeant la table TEMP.

NUM_STORE	CHAR_STORE	DATE_STOR
	This is my first PLSQL Program	24-SEP-01

6. a. Stockez un numéro de service dans une variable de substitution *iSQL**Plus

b. Ecrivez un bloc PL/SQL pour afficher le nombre de personnes travaillant dans ce service.

Indice : Activez DBMS_OUTPUT dans *iSQL**Plus en utilisant SET SERVEROUTPUT ON.

```
old 3: V_DEPTNO DEPARTMENTS.department_id%TYPE := &P_DEPTNO;
new 3: V_DEPTNO DEPARTMENTS.department_id%TYPE := 30;
6 employee(s) work for department number 30
PL/SQL procedure successfully completed.
```

7. Ecrivez un bloc PL/SQL qui déclare une variable nommée v_salary contenant le salaire d'un employé. Dans la section exécutable du programme, effectuez les opérations suivantes :

- Stockez le nom de l'employé dans une variable de substitution *iSQL**Plus
- Stockez son salaire dans la variable v_salary
- Si le salaire est inférieur à 3 000, augmentez l'employé de 500 et affichez le message "<Employee Name>'s salary updated" dans la fenêtre.
- Si le salaire de l'employé est supérieur à 3 000, affichez-le au format suivant "<Employee Name> earns".
- Testez le bloc PL/SQL avec les noms suivants :

LAST_NAME	SALARY
Pataballa	4800
Greenberg	12000
Ernst	6000

Remarque : Supprimez la définition de la variable stockant le nom d'employé à la fin du script.

8. Ecrivez un bloc PL/SQL permettant le stockage du salaire d'un employé dans une variable de substitution *iSQL**Plus. Dans la section exécutable du programme, effectuez les opérations suivantes :

- Calculez le salaire annuel de la façon suivante : salaire * 12.

- Calculez la prime comme indiqué ci-dessous :

Salaire annuel (Annual Salary)	Primes (Bonus)
>= 20000	2000
19999 - 10000	1000
<= 9999	500

- Affichez le montant de la prime dans la fenêtre, en respectant le format suivant :

"The bonus is \$....."

- Testez le bloc PL/SQL avec les cas suivants :

SALARY	BONUS
5000	2000
1000	1000
15000	2000

9. a. Ecrivez un bloc PL/SQL permettant le stockage d'un numéro d'employé, du nouveau numéro de service et du pourcentage d'augmentation du salaire dans des variables de substitution *iSQL*Plus*.

b. Mettez à jour le numéro de service de l'employé en utilisant la nouvelle valeur correspondante, et procédez de même pour le salaire. Utilisez la table EMP pour effectuer les mises à jour. Une fois celles-ci terminées, affichez le message, 'Update complete' dans la fenêtre. Si aucun enregistrement correspondant n'est trouvé, affichez le message, 'No Data Found'. Testez le bloc PL/SQL avec les cas suivants :

EMPLOYEE_ID	NEW_DEPARTMENT ID	% INCREASE	MESSAGE
100	20	2	Updation Complete
10	30	5	No Data found
126	40	3	Updation Complete

10. Créez un bloc PL/SQL qui déclare le curseur EMP_CUR et permet de sélectionner le nom, le salaire et la date d'embauche d'un employé, à partir de la table EMPLOYEES. Traitez chaque ligne à partir du curseur ; si le salaire est supérieur à 15 000 et que la date d'embauche est postérieure à 01-FEB-1988, affichez le nom, le salaire et la date d'embauche de l'employé dans la fenêtre, en respectant le format de l'exemple de résultat ci-dessous :

11. Créez un bloc PL/SQL permettant d'extraire, à partir de la table EMPLOYEES, le nom de famille et le numéro de service de chaque employé possédant une valeur EMPLOYEE_ID inférieure à 114. A l'aide des valeurs extraites de la table EMPLOYEES, remplissez deux tables PL/SQL, la première avec les enregistrements correspondant aux noms de famille des employés et la deuxième avec leur numéro de service. En utilisant une boucle, extrayez à partir des tables PL/SQL, les informations relatives au nom et au salaire de chaque employé ; affichez ensuite les résultats dans une fenêtre grâce à DBMS_OUTPUT.PUT_LINE. Procédez de la sorte pour les 15 premiers employés contenus dans les tables PL/SQL.

Employee Name: King Department_id: 90
 Employee Name: Kochhar Department_id: 90
 Employee Name: De Haan Department_id: 90
 Employee Name: Hunold Department_id: 60
 Employee Name: Ernst Department_id: 60
 Employee Name: Austin Department_id: 60
 Employee Name: Pataballa Department_id: 60
 Employee Name: Lorentz Department_id: 60
 Employee Name: Greenberg Department_id: 100
 Employee Name: Faviet Department_id: 100
 Employee Name: Chen Department_id: 100
 Employee Name: Sciarra Department_id: 100
 Employee Name: Urman Department_id: 100
 Employee Name: Popp Department_id: 100
 Employee Name: Raphaely Department_id: 30
 PL/SQL procedure successfully completed.

12. a. Créez un bloc PL/SQL qui déclare un curseur appelé DATE_CUR. Transmettez un paramètre de type DATE au curseur et affichez les détails de tous les employés embauchés après cette date.

```
DEFINE P_HIREDATE = 08-MAR-00
```

- b. Testez le bloc PL/SQL avec les dates d'embauche suivantes : 08-MAR-00, 25-JUN-97, 28-SEP-98, 07-FEB-99.

166 Ande 24-MAR-00
 167 Banda 21-APR-00
 173 Kumar 21-APR-00
 PL/SQL procedure successfully completed.

13. Créez un bloc PL/SQL permettant d'accorder une promotion aux employés gagnant plus de 3 000, en leur donnant l'intitulé de poste SR. CLERK ainsi qu'une augmentation de salaire de 10 %. Dans cet exercice, utilisez la table EMP et interrogez-la pour vérifier les résultats. **Indice** : Utilisez un curseur avec la syntaxe FOR UPDATE et CURRENT OF.

14. a. Pour réaliser l'exercice suivant, vous devez disposer d'une table nommée ANALYSIS où seront stockés les résultats. Vous pouvez la créer vous-même ou exécuter le script labAp_14a.sql pour le faire à votre place. Créez une table nommée ANALYSIS, qui comporte les trois colonnes suivantes :

Nom de colonne	ENAME	YEARS	SAL
Type de clé			
Valeurs NULL/uniques			
Table FK			
Colonne FK			
Type de données	VARCHAR2	NUMBER	NUMBER
Longueur	20	2	8, 2

- b. Créez un bloc PL/SQL permettant d'alimenter la table ANALYSIS avec les informations contenues dans la table EMPLOYEES. Utilisez une variable de substitution *iSQL*Plus* pour stocker le nom de famille d'un employé.
- c. Interrogez la table EMPLOYEES pour déterminer si l'employé possède plus de cinq années d'ancienneté dans l'organisation, et si son salaire est inférieur à 3 500 ; si tel est le cas, déclenchez une exception. Cette dernière doit être traitée à l'aide d'un gestionnaire d'exceptions approprié, qui insère les valeurs suivantes dans la table ANALYSIS : nom de famille, nombre d'années d'ancienneté et salaire en cours de l'employé. Dans le cas contraire, affichez Not due for a raise dans la fenêtre. Vérifiez les résultats en interrogeant la table ANALYSIS. Testez le bloc PL/SQL avec les cas suivants.

LAST_NAME	MESSAGE
Austin	Not due for a raise
Nayer	Not due for a raise
Fripp	Not due for a raise
Khoo	Due for a raise

Donner tous les réponses sous forme de fichiers script PL/SQL réel issu d'un utilitaire de développement SQL (iSQLPLUS, SQL Developer...), rassemblés dans un fichier compressé portant le nom de l'étudiant et déposé dans le drive de google classroom, toute autre dépôt ne sera pas accepté.