

Contrôle EAB

Formation L3 SITW
Durée : 1h, Mars 2021

Nom :
Prénom :
Groupe :

Exercice N°1 : (10 pts)(1pt*10)

Parmi les réponses proposées, cocher là où les bonnes réponses :

1) Pour créer une base de données Oracle on doit avoir le privilège :

- ☐ SYSASM ☒ SYSDBA ☐ RESSOURCE ☐ DBA

2) Dans le cadre d'administration Oracle, un processus serveur :

- ☐ Est un processus d'arrière-plan
☐ Permet le transfert des blocs de données du disque en mémoire
☐ Est un processus associé à un ensemble de processus utilisateurs
☒ Permet l'analyse et l'exécution des requêtes SQL soumises par les applications

3) Quelle est l'utilisation du « Library cache » dans Oracle?

- ☐ Contient les modifications apportées à la base de données
☒ Utilisé pour stocker le shared SQL
☐ Contient des informations sur l'utilisateur, telles que des privilèges d'utilisateur
☐ Toutes les réponses sont vraies

4) Le « redo log buffer » et « Shared pool » sont des éléments de:

- ☐ Cache PGA et Buffer ☒ SGA ☐ PGA ☐ Buffer cache

5) Quelle commande est utilisée pour supprimer une table existante?

- ☐ DELETE
☐ TRUNCATE TABLE
☐ Les deux : TRUNCATE et DELETE
☒ Aucun de ces réponses

6) Le script suivant permet de :

```
SQL>SPOOL mon_fichier.sql  
SQL> select * from employe;  
SQL> spool off»
```

- ☒ Créer le fichier mon_fichier
☐ Ouvrir le fichier mon_fichier
☒ Stocker les résultats d'une requête sur le fichier mon_fichier

7) Quelle doit être la valeur de la colonne « SALARY » après l'exécution du script suivant ?

```
SQL> UPDATE HR.EMPLOYEES SET SALARY=5000
2 WHERE EMPLOYEE_ID = 200;

1 ligne mise à jour.

SQL> SAVEPOINT SP1;

Savepoint créé.

SQL> UPDATE HR.EMPLOYEES SET SALARY=6000
2 WHERE EMPLOYEE_ID = 200;

1 ligne mise à jour.

SQL> ROLLBACK TO SAVEPOINT SP1;

Annulation (rollback) effectuée.

SQL> SELECT FIRST_NAME, LAST_NAME, SALARY
2 FROM HR.EMPLOYEES
3 WHERE EMPLOYEE_ID = 200;
```

FIRST_NAME	LAST_NAME	SALARY
Jennifer	Whalen	?

5000

8) Quelle doit être la valeur de la colonne « SALARY » après l'exécution du script suivant ?

```
SQL> SELECT FIRST_NAME, LAST_NAME, SALARY
2 FROM HR.EMPLOYEES
3 WHERE EMPLOYEE_ID = 200;
```

FIRST_NAME	LAST_NAME	SALARY
Jennifer	Whalen	4400

```
SQL> UPDATE HR.EMPLOYEES SET SALARY=6000
2 WHERE EMPLOYEE_ID = 200;

1 ligne mise à jour.

SQL> DROP TABLE SCOTT.EMP;

Table supprimée.

SQL> ROLLBACK;

Annulation (rollback) effectuée.

SQL> SELECT FIRST_NAME, LAST_NAME, SALARY
2 FROM HR.EMPLOYEES
3 WHERE EMPLOYEE_ID = 200;
```

FIRST_NAME	LAST_NAME	SALARY
Jennifer	Whalen	?

6000

9) Quels sont les fichiers mis à jour par le processus « DBWn » pour écrire les blocs modifiés ?

- ☐ Les fichiers de données et les fichiers journaux
- ☐ Les fichiers de données et les fichiers de contrôles
- ☒ Les fichiers de données
- ☐ Les fichiers journaux et les fichiers de contrôles

10) Vous avez besoin pour une colonne de vérifier qu'il n'existe pas deux fois la même valeur dans la table en même temps, la colonne ne doit pas contenir des valeurs nulles. Quel est le

type de contraintes que vous devez utiliser pour satisfaire les deux conditions ?

- ☐ CHECK
- ☐ UNIQUE
- ☐ NOT NULL
- ☒ PRIMARY KEY
- ☐ FOREIGN KEY

Exercice N°2: (10 pts)

Dans un bloc PL/SQL anonyme,

1) Définir un type enregistrement (RECORD) avec les champs suivants :

- **NOM**, du même type que l'attribut **ENAME** de la table EMP ;
- **FONCTION**, du même type que l'attribut **JOB** de la table EMP ;
- **SALAIRE**, du même type que l'attribut **SAL** de la table EMP.

2) Déclarez une table INDEX BY pour permettre le stockage temporaire des informations (nom, fonction, salaire) relatifs aux employés dont le numéro est compris entre **7900** et **8000** selon les conditions suivantes :

- Augmentation de **10%** si le salaire < **500**
- Augmentation de **5%** si le salaire < **900**
- Augmentation de **2 %** si le salaire > **900**

Utilisez la commande **DEFINE** pour fournir les deux valeurs **inf** et **sup** des numéros d'employés. Transmettre la valeur du numéro de l'employé au bloc PL/SQL via une variable de substitution.

3) Afficher les valeurs de la table créée.

Solution

```
DEFINE inf=7900  
DEFINE sup = 8000  
DECLARE
```

```
    TYPE e IS RECORD  
    (NOM EMP.ename%type,  
     FONCTION emp.job%type,  
     SALAIRE emp.sal%type);  
    emp_enr e;
```

```
    TYPE tab IS TABLE OF e  
    INDEX BY BINARY_INTEGER;
```

```
    v_tab tab;  
    v_sal emp.sal%type;  
    j BINARY_INTEGER := 1;  
    v_inf NUMBER := &inf;  
    v_sup NUMBER := &sup;  
    V_aug emp.sal%type;  
    n_sal emp.sal%type;  
    v_ename emp.ename%type;  
    v_fonction emp.job%type;  
BEGIN
```

0.5

2

1

1.5

```

FOR i IN v_inf..v_sup LOOP 0.5
    SELECT ename, job, sal 0.5
    INTO v_ename, v_fonction, v_sal 0.5
    FROM emp WHERE empno=i;_ 0.5
CASE
    WHEN v_sal<500 THEN v_aug:=.10; 0.5
    WHEN v_sal <900 THEN v_aug:=.50; 0.5
    WHEN v_sal> 900 THEN v_aug:=.20; 0.5
END CASE;
n_sal:= v_sal+v_sal*v_aug;
v_tab(j).SALAIRE:=n_sal;
v_tab(j).NOM:=v_ename;
v_tab(j).FONCTION:=v_fonction;
DBMS_OUTPUT.PUT_LINE ('le nom: ||v_tab(j).nom|| ' la fonction : ' ||
v_tab(j).fonction||' le nouveau salaire : ' ||v_tab(j).salaire);
j:=j+1;
END LOOP;
END;
/

```

2

1.5

1

0.5