

EXAMEN PRATIQUE

Session : Principale

Module : Système de Gestion des Bases de données		Documents autorisés : OUI
Enseignants : MME S.YEKKEN, M. F.OUELBANI, M. A.HOSNI, M. G.KHODJET EL KHIL, M. Y.GASRI		Nombre de pages : 2
Date : 04/06/2013	Heure : 09h00	Durée : 1h30
Classes : 3B1 – 3B3 – 3B4 – 3A6 – 3A13 – 3A14		

REMARQUE IMPORTANTE : Enregistrez toutes vos réponses dans un fichier TXT selon la dénomination suivante : **CLASSE_NOM_PRENOM.TXT**

SOIT LE SCHEMA RELATIONNEL SUIVANT

PATIENTS (<u>idPatient</u> , nom, prenom, dateNaissance)
MEDECINS (<u>idMedecin</u> , nom, prenom, specialite)
CONSULTATIONS (<u>idPatient#</u> , <u>idMedecin#</u> , <u>DateConsultation</u> , LieuConsultation)
idPatient et idMedecin sont des clés étrangères qui font respectivement référence aux relations PATIENTS et MEDECINS

Rq : Les clés primaires sont en **gras** et soulignées et les clés étrangères sont suivies par un #.

1. Ecrire les requêtes SQL permettant de créer et de remplir les 3 tables : **(3 PTS)**

PATIENTS	IDPATIENT	NOM	PRENOM	DATENAISSANCE
	101	Ben Salah	Salah	01/12/1983
	102	Ben Salem	Ahmed	05/08/1976

MEDECINS	IDMEDECIN	NOM	PRENOM	SPECIALITE
	1001	Bel Hadj	Salma	Generaliste
	1002	Ben Amor	Mohamed	Dentiste

CONSULTATIONS	IDMEDECIN	IDPATIENT	DATECONSULTATION	LIEUCONSULTATION
	1001	101	To_date(sysdate,'dd/mm/yyyy')	Clinique Hannibal
	1001	102	To_date(sysdate,'dd/mm/yyyy')	Clinique Hannibal

2. Ecrire une fonction stockée « fn_nb_cons » ayant comme paramètres l'identifiant du médecin et une date donnée et permettant de retourner le nombre de consultations réalisées par ce médecin à cette date. **(2 PTS)**
3. Transformer la fonction stockée « fn_nb_cons » par une procédure stockée équivalente nommée « proc_nb_cons ». **(2 PTS)**
4. Ecrire un bloc anonyme permettant d'afficher le nombre de consultations réalisées aujourd'hui par le médecin **1001** En utilisant: **(2 PTS)**
 - a. la fonction stockée « fn_nb_cons »,
 - b. la procédure stockée « proc_nb_cons ».
5. Ecrire une procédure stockée ayant comme paramètre l'identifiant d'un médecin et permettant d'afficher la liste des consultations de la journée courante. La liste doit inclure les informations suivantes : l'identifiant, le nom et le prénom de chaque patient. Cette procédure doit prendre en considération les exceptions suivantes :
 - a. Vérifier si l'identifiant du médecin existe dans la table MEDECINS
 - b. Vérifier si le médecin n'a pas de consultations en utilisant la fonction stockée « fn_nb_cons ». **(5 PTS)**
6. Ajouter une colonne nommée « NbConsultation » de type entier à la table MEDECINS. Ecrire une requête SQL permettant d'alimenter cette colonne à partir de la table CONSULTATIONS en calculant pour chaque médecin le nombre de ces consultations. **(3 PTS)**
7. Ecrire un trigger qui se déclenche après chaque insertion ou suppression effectuée sur la table CONSULTATIONS et met à jour la valeur de la colonne « NbConsultation » correspondante au médecin. **(3 PTS)**

BON TRAVAIL

Correction :

1.

```
Create table patients (idpatient number primary key, nom varchar2(15), prenom
varchar2(15), dateNaissance date );
Create table medecins (idmedecin number primary key, nom varchar2(15), prenom
varchar2(15), specialite varchar2(20) );
Create table consultation (idmedecin number, idpatient number , dateconsultation date,
lieuconsultation varchar2(15) , constraint pk_consultation primary key(idmedecin,idpatient,
dateconsultation), constraint fk_consul_patient foreign key(idpatient) references
patients(idpatient), constraint fk_consul_medecin foreign key(idmedecin) references
medecins(idmedecin));
Insert into patients values(101,'Ben Salah','Salah',to_date('01/12/1983','dd/mm/yyyy'));
Insert into patients values(102,'Ben Salem','Ahmed',to_date('05/08/1976','dd/mm/yyyy'));
Insert into medecins values(1001,'Bel Hadj','Salma','Generaliste');
Insert into medecins values(1002,'Ben Amor','Mohamed','Dentiste');
Insert into consultation values(1001,101, To_date(sysdate,'dd/mm/yyyy'), 'Clinique Hannibal');
Insert into consultation values(1001,102, To_date(sysdate,'dd/mm/yyyy'), 'Clinique Hannibal');
```

2.

```
Create or replace function fn_nb_Cons (idM number, d date)
Return number
Is
A number;
Begin
Select count(*) into A from consultation where idMedecin=idM and dateconsultation=d;
Return A;
End;
/
```

3.

```
Create or replace procedure proc_nb_Cons( idM in out number ,d date)
Is
A number;
Begin
Select count(*) into A from consultation where idMedecin=idM and dateconsultation=d;
idM:=A;
End;
/
```

4.

```
a/
Begin
Dbms_output.put_line(fn_nb_Cons(1001,
to_date(sysdate,'dd/mm/yyyy')));
end;
/
```

```
b/ Declare
A number := 1001;
Begin
proc_nb_cons (a,
to_date(sysdate,'dd/mm/yyyy'));
dbms_output.put_line(a);
end;
/
```

5.

```
Create or replace procedure proc_liste_consultation(idM number)
Is
A number; test boolean:=true; nb_cons exception;
Begin
    begin
        Select idMedecin into a from medecins where idmedecin=idM;
        Exception when no_data_found then dbms_output.put_line('medecin inexistant');
        test:=false;
        End;
    If (test) then
        For I in (Select P.idpatient id, P.nom nom ,P.prenom prenom from patients P inner join
        consultation C on P.idPatient=C.idPatient inner join Medecins M on
        M.idmedecin=C.idmedecin and M.idmedecin=idM and
        C.dateconsultation=to_date(sysdate,'dd/mm/yyyy'))
        Loop
            Dbms_output.put_line(I.id || ' ' || I.nom || ' ' || I.Prenom);
        End loop;
        If (fn_nbrConsultation(idM, to_date(sysdate,'dd/mm/yyyy'))=0) then raise nb_cons;
        End if;
        End if;
        Exception
        When nb_cons then Dbms_output.put_line('pas de consultations pour ce medecin');
        End;
    /
```

6.

```
Alter table medecins add (nbreConsultation number);
```

```
Update medecins M set nbreConsultation=
(select count(idpatient) from consultation where idmedecin= M.idmedecin ) ;
```

7.

```
Create or replace trigger trig_MAJ_NbrConsultation
After insert or delete
On consultation
For each row
Begin
    If inserting then update medecins set nbreConsultation=nbreconsultation+1 where
    idMedecin=:new.idMedecin; end if;
    If deleting then update medecins set nbreConsultation=nbreconsultation-1 where
    idMedecin=:old.idMedecin; end if;
    End;
    /
```