TP PROCEDURES ET FONCTIONS STOCKEES

I. Fonctions stockées

1)

```
Drop function if exists ToLowerCase;
Delimiter //
Create function ToLowerCase(value varchar(255))
     Returns varchar(255)
     Deterministic
Begin
    Declare chaine varchar(255);
    select concat(upper(left(value, 1)), lower(right(value, length(value)-1))) into chaine;
    return (chaine);
END //
Delimiter ;
```

2) Select ToLowerCase(nom), ToLowerCase(prenom) from Adherents;

3)

```
Drop function if exists GetRetard;
Delimiter //
Create function GetRetard(idA int)
    Returns int
    Deterministic
Begin
    Declare nbJ int;
    Select MIN(dureeMax-to_days(curdate())+to_days(dateEmp)) into nbJ from Emprunter
where dateRet is NULL AND NA=idA;
    return(nbJ);
END //
Delimiter ;
```

4) Select NA, Nom, Prenom, GetRetard(NA) from Adherents;

II. La bibliothèque – procédures stockées

6)

```
Drop procedure if exists ListerEmprunts;
Delimiter //
Create procedure ListerEmprunts(NumA int)
Begin
    select * from Emprunter where NA=NumA;
END //
Delimiter ;
```

7)

```
Drop procedure if exists ListerExempl;
Delimiter //
Create procedure ListerExempl(title varchar(150))
Begin
    select L.* from Livres L left join Oeuvres O on L.NO=O.NO left join Emprunter E on
L.NL=E.NL where dateRet is not null and O.Titre=title group by L.NL;
END //
Delimiter ;

Call ListerExempl(“Le Rouge et le Noir”);
```

8)

```
Drop procedure if exists AuthorsData;
Delimiter //
Create procedure AuthorsData(authorsName varchar(100))
Begin
    select Titre, count(L.NL) as Exemplaires from Oeuvres O left join Livres L on
O.NO=L.NO left join Emprunter E on L.NL=E.NL where dateRet is not null AND
Auteur=authorsName group by O.NO;
END //
Delimiter ;

Call AuthorsData(“Lewis CAROLL”);
```

9)

```
Drop procedure if exists SaveEmprunt;
Delimiter //
```

```
Create procedure SaveEmprunt(NumL int, dateEmpr date, dureeM int, dateReto date, NumA
int)
Begin
    insert into Emprunter(NL, dateEmp, dureeMax, dateRet, NA) values(NumL, dateEmpr,
dureeM, dateReto, NumA);
END //
Delimiter ;

Call SaveEmprunt(5, ' 2015-12-04', 3, ' 2015-12-20', 10);
```

10)

```
Alter table Emprunter modify dureeMax int(11) default 14;
```

11)

```
Drop procedure if exists VerifEmprunt;
Delimiter //
Create procedure VerifEmprunt(NumL int, dureeM int, NumA int, OUT ErrorMessage
varchar(50), OUT ErrorCode int)
Begin
    if ((select count(NL) from Livres where NL=NumL)=0)
        then
            set ErrorMessage='Le livre n’existe pas.';
            set ErrorCode=1;
    elseif ((select count(NA) from Adherents where NA=NumA)=0)
        then
            set ErrorMessage='L’adhérent n’existe pas.';
            set ErrorCode=2;
    elseif ((select count(NL) from Emprunter where NL=NumL and DateRet is null)!=0)
        then
            set ErrorMessage='Le livre est déjà emprunté.';
            set ErrorCode=3;
    elseif ((select count(NA) from Emprunter where NA=NumA and DateRet is null)>=3)
        then
            set ErrorMessage='L’adhérent emprunte déjà trois livres.';
            set ErrorCode=4;
    else
        call SaveEmprunt(NumL, curdate(), dureeM, NULL, NumA);
        set ErrorMessage='Emprunt enregistré.';
        set ErrorCode=0;
    end if;
```

```
END //
Delimiter ;

Call VerifEmprunt(22, 10, 3, @MessageError, @CodeError);

Select @CodeError as CodeError, @MessageError as MessageError;
```

12)

```
Alter table Livres add Emprunte boolean default 0 ;

Drop trigger if exists EmpruntLivre;
Delimiter //
Create trigger EmpruntLivre
before insert on Emprunter
for each row
Begin
    update Livres set Emprunte=1 where NL=new.NL;
END //
Delimiter ;

Drop trigger if exists RetourLivre;
Delimiter //
Create trigger RetourLivre
after update on Emprunter
for each row
Begin
    if (old.dateRet is not null)
        then
             update Livres set Emprunte=0 where NL=old.NL;
     end if;
END //
Delimiter ;


Drop trigger if exists DeleteEmprunt;
Delimiter //
Create trigger DeleteEmprunt
after delete on Emprunter
for each row
Begin
    update Livres set Emprunte=0 where NL=old.NL;
END //
Delimiter ;
```

13)

```
Drop procedure if exists VerifRetour;
Delimiter //
Create procedure VerifRetour (NumL int, NumA int, OUT ErrorMessage varchar(50), OUT
ErrorCode int, OUT nbJRet int)
Begin
    Declare dateE date;
    Declare dureeEmp int;
    set nbJRet=null;
    if ((select count(NL) from Livres where NL=NumL)=0)
        then
            set ErrorMessage='Le livre n’existe pas.';
            set ErrorCode=1;
    elseif ((select count(NA) from Adherents where NA=NumA)=0)
        then
            set ErrorMessage='L’adhérent n’existe pas.';
            set ErrorCode=2;
    elseif ((select count(NL) from Emprunter where NL=NumL and dateRet is null)=0)
        then
            set ErrorMessage='Le livre n’est pas emprunté.';
            set ErrorCode=3;
    else
        update Emprunter set dateRet=curdate() where NL=NumL and dateRet is null;
        Select to_days(dateEmp) into dateE from Emprunter where NL=NumL and dateRet is
null;
        Select to_days(dureeMax) into dureeEmp from Emprunter where NL=NumL and
dateRet is null;
        set nbJRet=to_days(curdate())-dateE-dureeEmp;
        set ErrorMessage='Retour enregistré.';
        set ErrorCode=0;
    end if;
END //
Delimiter ;

Call VerifRetour (2, 28, @MessageError, @CodeError, @nbJ);

Select @CodeError as CodeErreur, @MessageError as MessageErreur, @nbJ as
Nombre_Jours_Retard ;
```