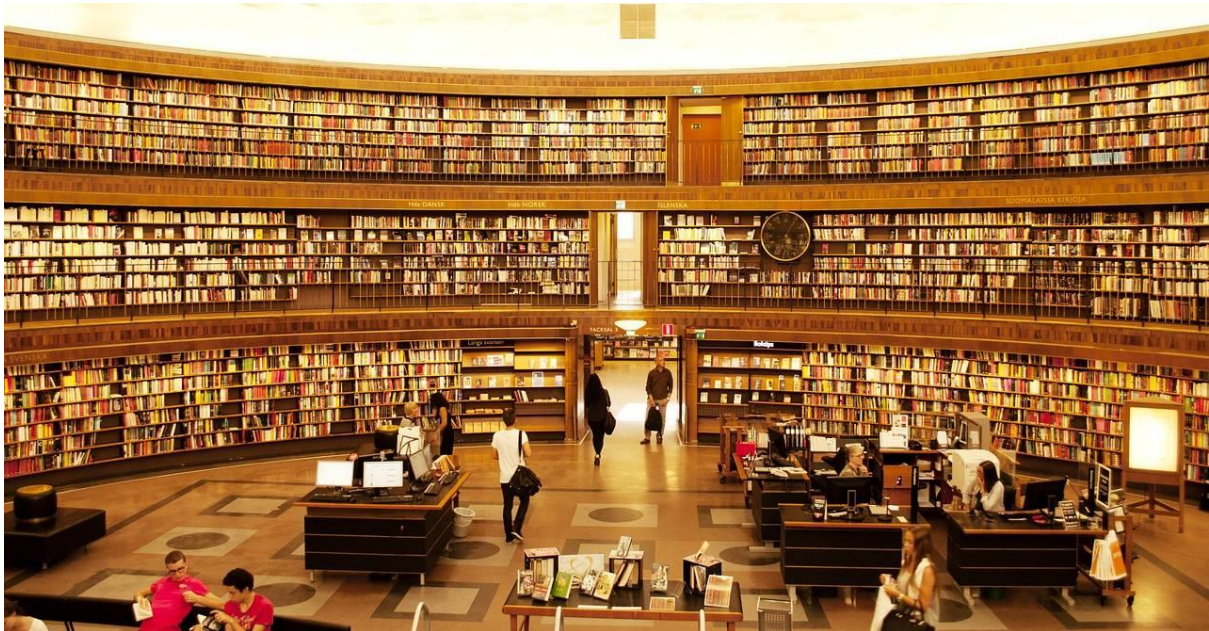


Rapport du projet du Système de Gestion de Base des données



Gestion d'une Bibliothèque



Réalisé par : Wafed TABAI

2018-2019

Introduction

Le fait de gérer une bibliothèque paraît au début une tâche assez facile, mais quand on est sur terrain on remarque la difficulté avec le nombre énorme des ouvrages à gérer, les adhérents qui augmentent chaque jour et surtout le facteur emprunt qui reste le noyau dur de cette gestion. Du coup, la réalisation d'une base de données simple mais solide paraît indispensable, ceci est le but de notre projet qui sera présenté dans ce rapport.

Les cas d'utilisations choisis

4.1. Le cas d'utilisation « Gestion des adhérents »

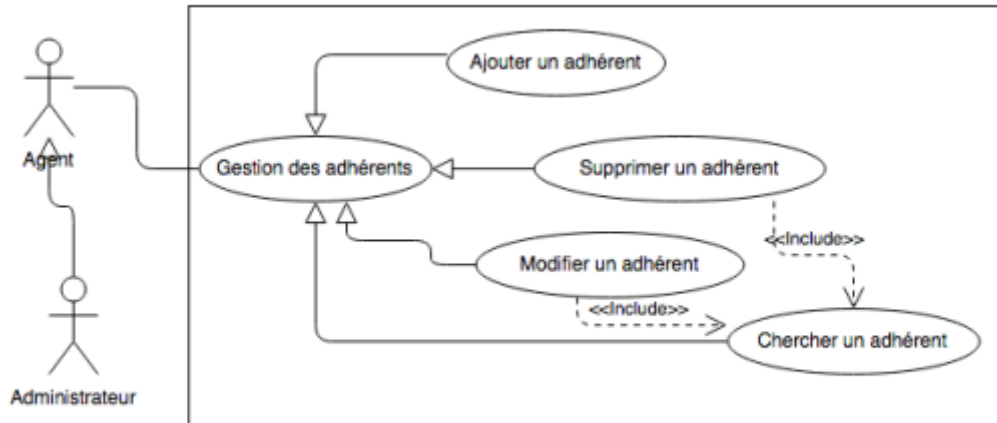


Fig.2 : Diagramme de cas d'utilisation « Gestion des adhérents »

4.2. Le cas d'utilisation « Gestion du catalogue »

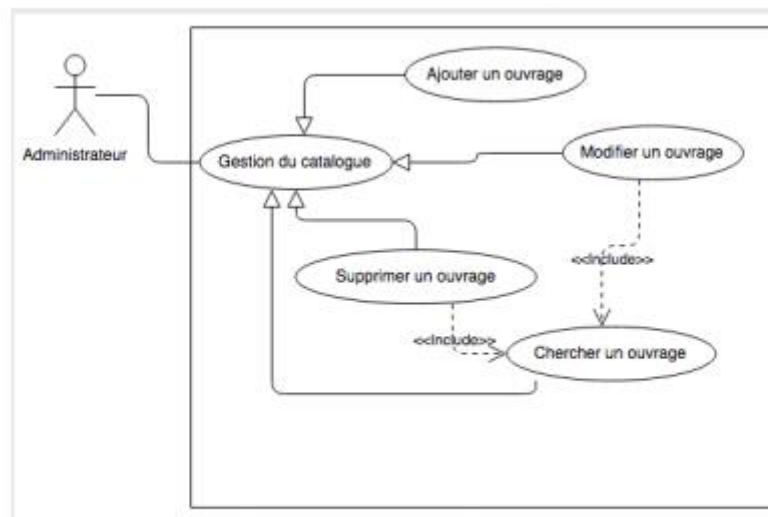


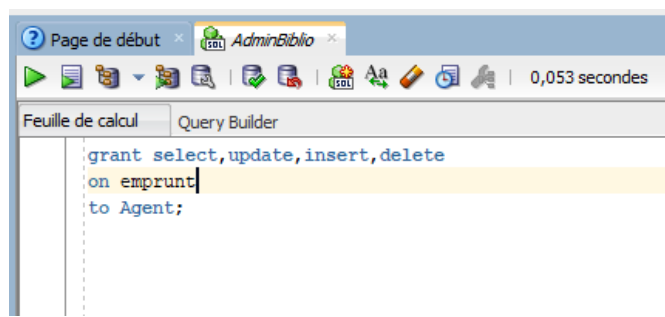
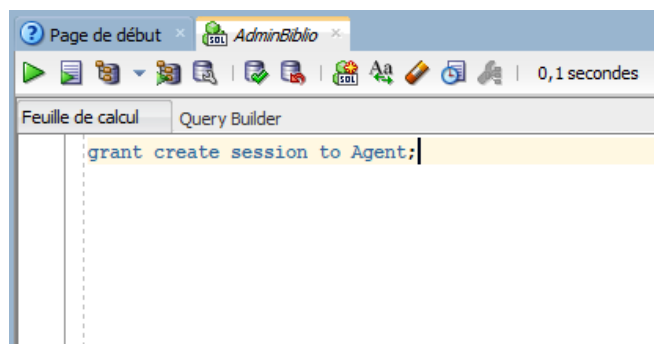
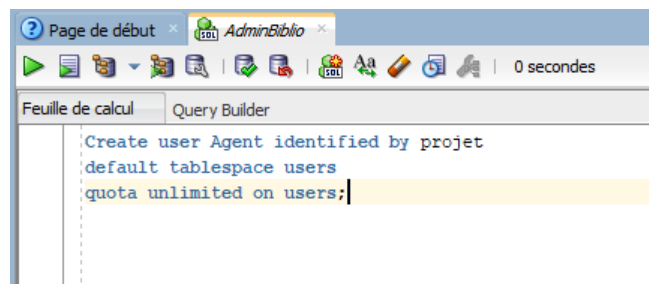
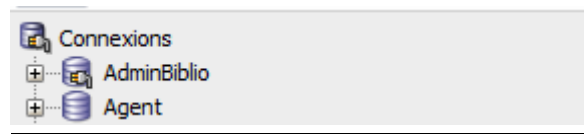
Fig.3 : Diagramme de cas d'utilisation « Gestion du catalogue »

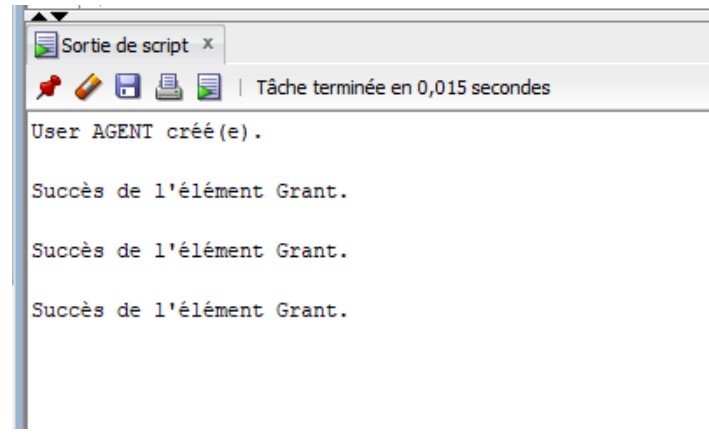
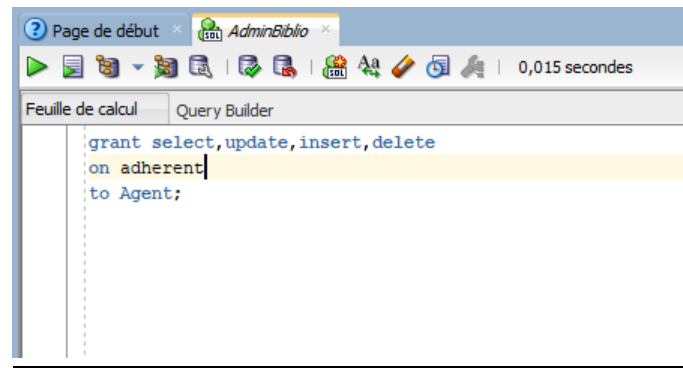
Les utilisateurs du système

Création de deux sessions d'utilisateurs :

« AdminBiblio » : L'Administrateur

« Agent » : L'Agent





Les tables

Pour réaliser cette base de données nous avons élaboré 5 tables comme suit :

Table Adhérent

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	NOADH	NUMBER(6,0)	No	(null)	1	(null)
2	NOM	VARCHAR2(80 BYTE)	No	(null)	2	(null)
3	PRENOM	VARCHAR2(80 BYTE)	No	(null)	3	(null)
4	ADRESSE	VARCHAR2(200 BYTE)	No	(null)	4	(null)
5	NCIN	NUMBER(8,0)	Yes	(null)	5	(null)
6	TEL	NUMBER(10,0)	Yes	(null)	6	(null)
7	DATEADH	DATE	No	(null)	7	(null)
8	EMAIL	VARCHAR2(80 BYTE)	No	(null)	8	(null)

Table Catalogue

❖	COLUMN_NAME	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODEG	NUMBER(10,0)	No	(null)		1	(null)		
2	TITRE	VARCHAR2(200 BYTE)	No	(null)		2	(null)		
3	NOMAUT	VARCHAR2(80 BYTE)	No	(null)		3	(null)		
4	PRENOMAUT	VARCHAR2(80 BYTE)	No	(null)		4	(null)		
5	ANED	NUMBER(4,0)	Yes	(null)		5	(null)		
6	EDITEUR	VARCHAR2(50 BYTE)	No	(null)		6	(null)		
7	PRIX	NUMBER(8,3)	Yes	(null)		7	(null)		

Table Exempleire

❖	COLUMN_NAME	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODEXP	VARCHAR2(10 BYTE)	No	(null)		1	(null)		
2	CODEG	NUMBER(10,0)	No	(null)		2	(null)		
3	ETAT	CHAR(8 BYTE)	No	(null)		3	(null)		
4	DISP	CHAR(3 BYTE)	No	(null)		4	(null)		

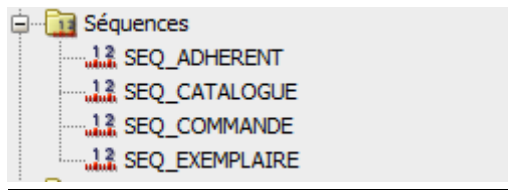
Table Emprunt

❖	COLUMN_NAME	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CODEXP	VARCHAR2(10 BYTE)	No	(null)		1	(null)		
2	DATEEMP	DATE	No	(null)		2	(null)		
3	NOADH	NUMBER(6,0)	No	(null)		3	(null)		
4	DATERPREVUE	DATE	No	(null)		4	(null)		
5	DATEREFFECTIVE	DATE	Yes	(null)		5	(null)		

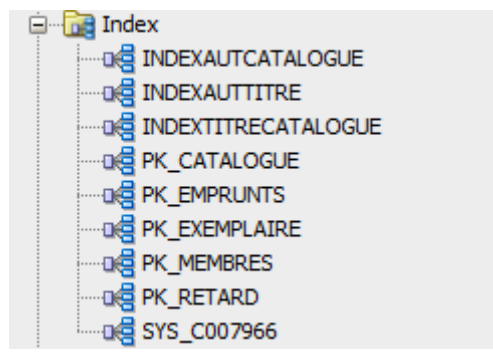
Table Retard

❖	COLUMN_NAME	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	NOADH	NUMBER(6,0)	No	(null)		1	(null)		
2	CODEG	NUMBER(10,0)	No	(null)		2	(null)		
3	DATEEMP	DATE	Yes	(null)		3	(null)		
4	DATEREFFECTIVE	DATE	Yes	(null)		4	(null)		
5	PENALITÉ	NUMBER	Yes	(null)		5	(null)		
6	ENCOURS	CHAR(3 BYTE)	Yes	(null)		6	(null)		

Les Séquences



Les Index



Les règles d'intégrité

Dans la table catalogue, l'année d'édition (anEd) doit être après 1950 :

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
1 CK_CATALOGUE_ANED	Check	aned > 1950	(null)	(null)	(null)	(null)	ENABLED

La date d'emprunt (date système) doit être précédente à celle de retour :

```
create or replace TRIGGER "TRIG_EMP_ANT_RET1"
before insert on emprunt
for each row
begin
if (:new.dateRprevue < :new.dateemp ) then
raise_application_error(-20000,'Il faut que la date d'emprunt soit antérieure à celle du retour');
end if;
if ( to_char(:new.dateemp,'dd')<>to_char(sysdate,'dd')) then
raise_application_error(-20300,'Il faut que la date d'emprunt soit celle d'aujourd'hui');
end if;

end;
```

```

create or replace TRIGGER TRIG_EMP_ANT_RET2
before update of dateprevue,dateemp on emprunt
for each row
begin
if (:new.dateRprevue < :new.dateemp ) then
raise_application_error(-20000,'Il faut que la date d'emprunt soit antérieure à celle du retour');
end if;
end;

```

On ne peut emprunter un titre qu'après avoir adhéré à la bibliothèque. (La date d'adhésion est antérieure ou égale à celle de l'emprunt) :

```

create or replace TRIGGER "TRIG_ADH_ANT_EMP" before insert or update on emprunt
for each row
Declare
v_dateAdh date;
begin
select dateAdh into v_dateAdh from adherent where noAdh = :new.noAdh;
if (v_dateAdh > :new.dateEmp)
then
raise_application_error(-20001,'La date d'adhésion doit etre antérieure ou égale à celle de l'emprunt');
end if;
end;

```

Un emprunt ne doit pas dépasser un mois et la disponibilité d'un exemplaire doit passer de 'oui' à 'non' :

```

create or replace TRIGGER "TRIG_EMP_J_MAX" before insert or update on emprunt
for each row
Declare
DateDiff NUMBER;
begin
select :new.dateRprevue - sysdate into DateDiff
from dual;
if DateDiff >31
then
raise_application_error(-20002,'Un emprunt ne doit pas dépasser 31 jours');
else
update exemplaire e set disp ='non' where e.codexp = :new.codexp and e.disp='oui';
end if;
end;

```

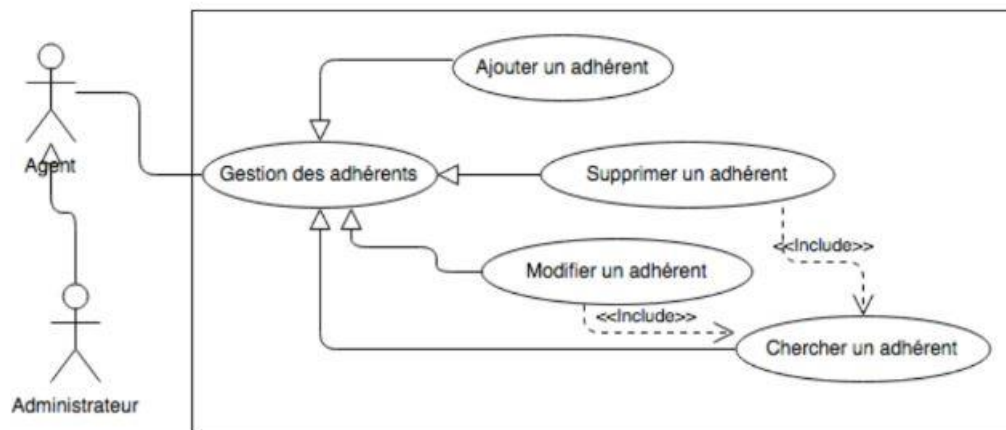

Un adhérent ne peut emprunter qu'au plus 5 titres à la fois et un seul exemplaire pour chaque titre :

```
create or replace TRIGGER "TRIG_EMP_MAX" before insert or update of codexp, noadh on emprunt
for each row
Declare
pragma autonomous_transaction;
nb_empl number;
nb_emp2 number;
v_codeg exemplaire.codeg$type;
begin

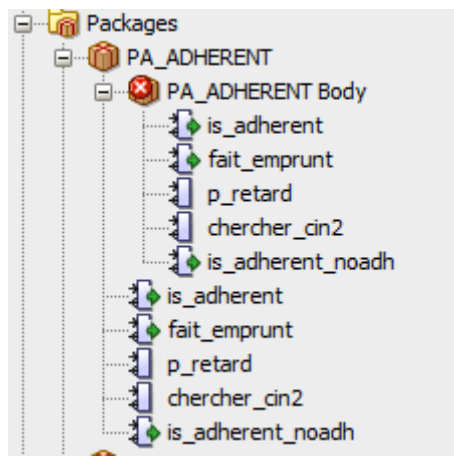
select codeg into v_codeg from exemplaire where upper(codexp)=upper(:new.codexp);
select count(*) into nb_empl from emprunt e, exemplaire ex
where ex.codeg=v_codeg and upper(ex.codexp)=upper(e.codexp) and e.noadh=:new.noadh and e.DATERPREVUE>=sysdate;
if (nb_empl>0) then raise_application_error(-20011,'Vous ne pouvez pas effectuer plus d'un emprunt sur un même livre');
end if;

select count(:new.noadh) into nb_emp2 from emprunt
where :new.dateEmp =sysdate ;
if(nb_emp2 >= 5) then raise_application_error(-20007,'Vous ne pouvez pas effectuer plus de 5 emprunts');
end if;
end;
```

Gestion des Adhérents



Package



1-Ajout d'un adhérent

```
create or replace TRIGGER "TRIG_AJOUT_ADHERENT"
before insert or update on adherent
for each row
declare
pragma autonomous_transaction;
begin
if PA_ADHERENT.is_adherent(:new.ncin) then
raise_application_error(-20004,'Ce numéro de CIN existe déjà dans la base!');
end if;
end;
```

2-Suppression d'un adhérent

```
create or replace TRIGGER "TRIG_SUPPRESSION_ADHERENT" before delete on adherent
for each row
declare
pragma autonomous_transaction;
begin
if PA_ADHERENT.fait_emprunt(:old.noadh) then
raise_application_error(-20005,'Ne peut pas être supprimé car il a un emprunt en cours');
end if;
end;
```

3- Modification d'un adhérent

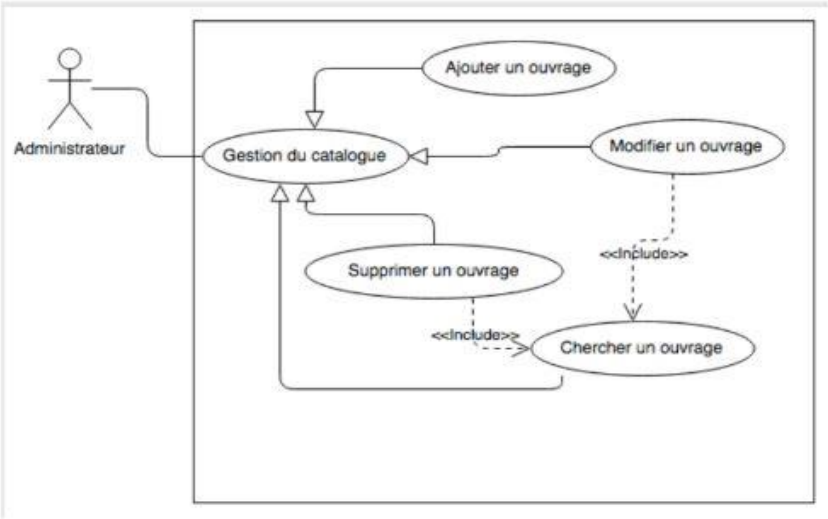
```
create or replace TRIGGER "TRIG_MODIF_ADHERENT"  
BEFORE UPDATE OF noadh,nom,prenom,ncin,dateadh ON adherent  
FOR EACH ROW  
BEGIN  
/*ajouter la fonction is_existant pour respecter la charte */  
if (:OLD.noadh != :NEW.noadh) or (:OLD.nom != :NEW.nom) or (:OLD.prenom != :NEW.prenom)  
or (:OLD.ncin != :NEW.ncin) or (:OLD.dateadh != :NEW.dateadh)  
then  
RAISE_APPLICATION_ERROR(-20006, 'Ce champ ne peut pas etre modifié!') ;  
end if;  
END;
```

4- Chercher un adhérent

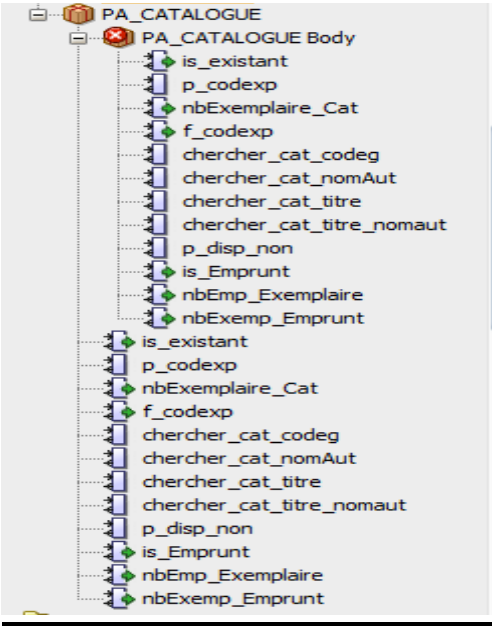
```
function is_adherent (x ADHERENT.NCIN%type)  
return boolean  
is  
v number;  
begin  
select count(*) into v  
from adherent  
where adherent.ncin=x;  
if (v>0) then  
return true;  
end if;  
if (x=null or v<=0) then  
return false;  
end if ;  
end;
```

Dans ce premier cas d'utilisation on a présenté le volet de gestion des adhérents qui présente une tâche principale au sein d'une bibliothèque

Gestion du catalogue



Package



1- Ajout d'un ouvrage

```
create or replace TRIGGER "TRIG_AJOUT_CATALOGUE"  
before insert or update on catalogue  
for each row  
declare  
pragma autonomous_transaction;  
begin  
if PA_CATALOGUE.is_existant(:new.titre, :new.nomAut, :new.prenomAut, :new.anEd) then  
raise_application_error(-20008, 'Ce livre existe déjà dans la base!');  
end if;  
end;
```

2- Suppression d'un ouvrage

```
create or replace TRIGGER TRIG_SUPPRESSION_CATALOGUE  
BEFORE DELETE OR UPDATE ON CATALOGUE  
FOR EACH ROW  
declare  
begin  
if (is_Emprunt(:old.Codeg) > 0) then  
update exemplaire e set e.disp = 'non' where e.codeg = :new.codeg and e.disp = 'oui';  
raise_application_error(-20009, 'Ce catalogue ne peut pas être supprimé car il des exemplaires empruntés');  
end if;  
end;
```

3- Modification d'un ouvrage

```
create or replace TRIGGER "TRIG_MODIF_CATALOGUE"  
BEFORE UPDATE ON catalogue  
FOR EACH ROW  
BEGIN  
if (PA_CATALOGUE.NBEXEMPLAIRE_CAT(:old.codeg) >= 1)  
then  
RAISE_APPLICATION_ERROR(-20010, 'Livre non-modifiable, car il a des exemplaires') ;  
end if;  
END;
```

4- Chercher un ouvrage

```
procedure chercher_cat_codeg (x catalogue.codeg%type)
is
errisnotcatalogue EXCEPTION;
v_record catalogue%rowtype;
y NUMBER;
Begin
select count(*) into y
from catalogue
where catalogue.codeg=x;
if (y=0) then RAISE errisnotcatalogue;
else
select * into v_record from catalogue where catalogue.codeg=x;
DBMS_OUTPUT.PUT_LINE('Titre: ' || v_record.titre || ' Nom auteur: ' || v_record.nomaut || ' Prenom auteur: ' || v_record.prenomaut ||
' Année d"edition: ' || v_record.aned || ' Editeur: ' || v_record.editeur || ' prix: ' || v_record.prix);
end if;
EXCEPTION
WHEN errisnotcatalogue THEN DBMS_OUTPUT.PUT_LINE('Adherent inexistant');
end;
```

```
procedure chercher_cat_nomAut(x catalogue.nomaut%type)
is
errisnotcatalogue EXCEPTION;
v_record catalogue%rowtype;
cursor curnomaut is
select * from catalogue
where catalogue.nomaut=x;

y NUMBER;
Begin
select count(*) into y
from catalogue
where UPPER(catalogue.nomaut)=UPPER(x);
if (y=0) then RAISE errisnotcatalogue;
else
open curnomaut;
loop
fetch curnomaut into v_record;
exit when curnomaut%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Codeg: ' || v_record.codeg || ' Titre: ' || v_record.nomaut || ' Prenom auteur: ' || v_record.prenomaut ||
' Année d"edition: ' || v_record.aned || ' Editeur: ' || v_record.editeur || ' prix: ' || v_record.prix);
end loop;
end if;
EXCEPTION
WHEN errisnotcatalogue THEN DBMS_OUTPUT.PUT_LINE('catalogue inexistant');
end;
```

```

procedure chercher_cat_titre (x catalogue.titre%type)
is
errisnotcatalogue EXCEPTION;
v_record catalogue%rowtype;
cursor curtitre is
select * from catalogue
where upper(catalogue.titre)=upper(x);

y NUMBER;
Begin
select count(*) into y
from catalogue
where upper(catalogue.titre)=upper(x);
if (y=0) then RAISE errisnotcatalogue;
else
open curtitre;
loop
fetch curtitre into v_record;
exit when curtitre%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('Codeg: ' || v_record.codeg || ' Nom auteur: ' || v_record.nomaut || ' Prenom auteur: ' || v_record.prenomaut ||
' Année d"edition: ' || v_record.aned || ' Editeur: ' || v_record.editeur || ' prix: ' || v_record.prix);
end loop;
end if;
EXCEPTION
WHEN errisnotcatalogue THEN DBMS_OUTPUT.PUT_LINE('catalogue inexistant');
end;

```

```

procedure chercher_cat_titre_nomaut (x catalogue.titre%type, v catalogue.nomaut%type)
is
errisnotcatalogue EXCEPTION;
v_record catalogue%rowtype;
cursor curtitre_nomaut is
select * from catalogue
where upper(catalogue.titre)=upper(x) and upper(catalogue.nomaut)=upper(v);

y NUMBER;
Begin
/*select count(*) into y
from catalogue
where catalogue.titre=x and catalogue.nomaut=v;
if (y=0) then RAISE errisnotcatalogue;*/

open curtitre_nomaut;
loop
fetch curtitre_nomaut into v_record;
if curtitre_nomaut%rowcount=0 then raise errisnotcatalogue;
end if;
/*exit when curtitre_nomaut%NOTFOUND;*/

DBMS_OUTPUT.PUT_LINE('Codeg: ' || v_record.codeg || ' Prenom auteur: ' || v_record.prenomaut ||
' Année d"edition: ' || v_record.aned || ' Editeur: ' || v_record.editeur || ' prix: ' || v_record.prix);
end loop;

EXCEPTION
WHEN errisnotcatalogue THEN DBMS_OUTPUT.PUT_LINE('catalogue inexistant');
end;

```

Dans ce deuxième cas d'utilisation, on a traité un facteur primordial dans une bibliothèque et qui est le noyau de toute opération : Les Ouvrages.

Les Vues

« VEmprunt » :

```
CREATE OR REPLACE FORCE VIEW "PROJET"."VEMPRUNT" ("NOADH", "NOM", "NCIN", "CODEG", "TITRE", "DATEEMP", "DATERPREVUE", "DATEREFFECTIVE") AS
Select a.noadh, a.nom, a.ncin, c.codeg, c.titre, e.dateemp, e.dateRprevue, e.dateReffective
from adherent a, catalogue c, emprunt e, exemplaire ex
where a.noadh=e.noadh and c.codeg=ex.codeg and ex.codexp=e.codexp;
```

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS	❖ INSERTABLE	❖ UPDATABLE	❖ DELETABLE
1 NOADH	NUMBER(6)	No	(null)	1 (null)	NO	NO	NO	NO
2 NOM	VARCHAR2(80)	No	(null)	2 (null)	NO	NO	NO	NO
3 NCIN	NUMBER(8)	Yes	(null)	3 (null)	NO	NO	NO	NO
4 CODEG	NUMBER(10)	No	(null)	4 (null)	NO	NO	NO	NO
5 TITRE	VARCHAR2(200)	No	(null)	5 (null)	NO	NO	NO	NO
6 DATEEMP	DATE	No	(null)	6 (null)	YES	YES	YES	YES
7 DATERPREVUE	DATE	No	(null)	7 (null)	YES	YES	YES	YES
8 DATEREFFECTIVE	DATE	Yes	(null)	8 (null)	YES	YES	YES	YES

Ajout, modification à travers la vue :

```
create or replace TRIGGER "TRIG_VUE_EMP" instead of insert on vemprunt
for each row
declare
v_codexp EXEMPLAIRE.CODEXP&type;
v_nb1 number;
v_nb2 number;
begin
select count(*) into v_nb1 from exemplaire where codeg=:new.codeg;
if (v_nb1>0) then
select count(*) into v_nb2 from exemplaire where codeg=:new.codeg and disp='oui';
if (v_nb2>0) then
v_codexp:=PA_CATALOGUE.f_codexp(:new.codeg);
if (not PA_ADHERENT.is_adherent(:new.ncin)) then insert into adherent values (:new.noadh,:new.nom,' ',' ',:new.ncin,0,sysdate,' ');
end if;
insert into emprunt values(v_codexp,:new.dateemp,:new.noadh,:new.daterprevue,:new.datereffective);
else raise_application_error(-20030,'Pas d'exemplaire disponible pour ce catalogue ');
end if;

else raise_application_error(-20031,'Exemplaires inexistants pour ce catalogue');
end if;

end;
```


Calcul de pénalité à l'aide d'un trigger en cas de retard :

```
create or replace TRIGGER TRIG_RETARD_DR_EFFECTIVE
BEFORE update of datereffective on emprunt
for each row
declare
begin
if(:new.datereffective > sysdate) then
raise_application_error(-20035,'Date de retour effective non-valide (supérieur à la date actuelle)!');
end if;

end;
```

```
create or replace TRIGGER TRIG_RETARD
after update of DATEREFFECTIVE on EMPRUNT
for each row
declare

vcodeg exemplaire.codeg$type;
x number;
begin
select codeg into vcodeg from exemplaire e where :new.codexp=e.codexp;
select count(*) into x from retard where noadh=:new.noadh and encours='oui';
if (x=1) then
delete from retard where noadh=:new.noadh and encours='oui';
PA_ADHERENT.p_retard(vcodeg,:new.daterprevue,:new.datereffective,:new.dateemp,:new.noadh);
else
PA_ADHERENT.p_retard(vcodeg,:new.daterprevue,:new.datereffective,:new.dateemp,:new.noadh);
end if;

end;
```

```
procedure p_retard (vcodeg exemplaire.codeg$type,dp date, df date,de date,vnoadh adherent.noadh$type)
is
x number;
begin
x:=(df-dp)*2;
if (df-dp)>0 and (df-dp)<=30 then
insert into RETARD values (vnoadh,vcodeg,de,df,df-dp,'oui');

end if;

if (df-dp)>30 and (df-dp)<90 then
insert into RETARD values (vnoadh,vcodeg,de,df,x,'oui');
end if;

if (df-de)>=90 then
insert into RETARD values (vnoadh,vcodeg,de,df,0,'oui');
/* Il doit payer le prix du livre*/
end if;
end;
```

Expiration de la pénalité et autorisation d'ajout d'un emprunt à travers un trigger :

```

create or replace TRIGGER TRIG_AJOUT_EMPRUNT
before insert on emprunt
for each row
declare
vcodeg exemplaire.codeg$type;
x number;
v_datereffective retard.datereffective$type;
v_penalite retard.penalite$type;
v_encours retard.encours$type;
v_nb_exemp number;
v_nb_exemp_disp number;
begin
select count(:new.noadh) into x from RETARD;
if (x<>null) then
select datereffective,penalite,encours into v_datereffective,v_penalite,v_encours from retard where noadh=:new.noadh;
if ((sysdate - v_datereffective > v_penalite) and( v_penalite > 0)) then
update retard set retard.encours='non' where retard.noadh=:new.noadh;
end if;
if ( ( sysdate - v_datereffective < v_penalite) and( v_penalite > 0)) then
raise_application_error(-20018,'Cet adhérent est encore pénalisé!');
end if;
if ( v_penalite=0 and upper(v_encours)='OUI') then
raise_application_error(-20021,'Cet adhérent doit payer le prix du livre d'abord!');
end if;
else

```

```

else
select count(*) into v_nb_exemp from exemplaire ex where upper(:new.codexp)=upper(ex.codexp);
if (v_nb_exemp >0) then
select count(*) into v_nb_exemp_disp from exemplaire ex where upper(:new.codexp)=upper(ex.codexp) and upper(dispo)='OUI';
if (v_nb_exemp_disp >0) then
PA_CATALOGUE.p_codexp(:new.codexp);
else raise_application_error(-20022,'Exemplaire non-disponible pour le moment!');
end if;
else raise_application_error(-20023,'Exemplaire inexistant!');
end if;
end if;
end;




```

« VStatistiques2018 »

```

CREATE OR REPLACE FORCE VIEW "PROJET"."VSTATISTIQUES2018" ("NB_TOTAL", "NB_MEDIOCRE", "NB_EMPRUNT", "POURC_RETARD", "CODEG") AS
SELECT count(e.CODEG) as nb_total, || select count(al.CODEG) from exemplaire al where al.etat='mediocre' and al.CODEG = e.CODEG as nb_mediocre ,
(select count( vem.codeg) from vemprunt vem where e.codeg=vem.codeg ) as nb_emprunt,
((select count(codeg) from RETARD where e.codeg=retard.codeg)*100/(select count( vem.codeg) from vemprunt vem where e.codeg=vem.codeg having count(vem.codeg)>0 )) as pourc_retard,
e.CODEG from exemplaire e group by e.CODEG;

```

Colonnes	Données	Droits	Dépendances	Détails	Déclencheurs	SQL			
   Actions...									
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	NB_TOTAL	NUMBER	Yes	(null)	1 (null)		NO	NO	NO
2	NB_MEDIOCRE	NUMBER	Yes	(null)	2 (null)		NO	NO	NO
3	NB_EMPRUNT	NUMBER	Yes	(null)	3 (null)		NO	NO	NO
4	POURC_RETARD	NUMBER	Yes	(null)	4 (null)		NO	NO	NO
5	CODEG	NUMBER(10)	No	(null)	5 (null)		NO	NO	NO

Interdiction d'accès le 29 et 30 de chaque mois

```
create or replace trigger interdiction_dacces_catalogue
before delete or update or insert on catalogue
begin
if ( to_char(sysdate,'dd') = '29' or to_char(sysdate,'dd') = '30') then
raise_application_error(-20555,'Acces à la table "CATALOGUE" interdit ce jour du mois!');
end if;
end;
```

```
create or replace trigger interdiction_dacces_emprunt
before delete or update or insert on emprunt
begin
if ( to_char(sysdate,'dd') = '29' or to_char(sysdate,'dd') = '30') then
raise_application_error(-20555,'Acces à la table "EMPRUNT" interdit ce jour du mois!');
end if;

end;
```

```
create or replace trigger interdiction_dacces_exemplaire
before delete or update or insert on exemplaire
begin
if ( to_char(sysdate,'dd') = '29' or to_char(sysdate,'dd') = '30') then
raise_application_error(-20555,'Acces à la table "EXEMPLAIRE" interdit ce jour du mois!');
end if;

end;
```

Conclusion

Dans ce rapport, nous avons exposé les étapes d'élaboration de notre base des données qui consiste à gérer des tâches effectuées dans une bibliothèque numériquement.

Pour couronner le tout, cette base des données vise à diminuer le temps que prend une validation verbale ou manuelle, et éviter de tomber dans l'erreur.

Ce projet était une véritable expérience de travail en collaboration, qui nous a renforcé l'esprit de partage des connaissances.

A partir de ces éléments, nous estimons avoir réussi à créer une application fonctionnelle qui peut être utilisée par toutes les bibliothèques.

Certainement, quelques imperfections sont encore présentes sur l'application, mais elles n'altèrent en rien son fonctionnement global.