

Compte rendu GAUTIER Titouan

Exercice 1:

A -

```
CREATE or replace TRIGGER modif_salaire after UPDATE ON employe FOR EACH ROW
BEGIN
    if :new.salaire < :old.salaire then
        RAISE_APPLICATION_ERROR (-20001,'viol de la règle : une salaire ne peut pas être diminué');
    END IF ;
END;
```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
modif_salaire	after	update	employe	oui

```
--Test--
update employe set SALAIRE = 2500 where NUEMPL = 28;
```

B -

```
Create or replace TRIGGER duree_hebdo after update on employe for each row
BEGIN
    if :new.hebdo > :old.hebdo then
        RAISE_APPLICATION_ERROR (-20002,'viol de la règle : une durée hebdo ne peut être augmenté');
    end if;
end;
```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
duree_hebdo	after	update	employe	oui

```
--Test--
update employe set HEBDO = 35 where NUEMPL = 28;
```

Exercice 2:

A -

```
alter table travail drop constraint FK_nuempl_travail;
alter table travail add constraint FK_nuempl_travail foreign key (NUEMPL) references employe (NUEMPL) initially deferred ;

CREATE or replace trigger suppr_employe after delete on employe
BEGIN
    delete from travail where NUEMPL not in (select NUEMPL from EMPLOYE);
end;
```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
suppr_employe	after	delete	employe	non

```
--Test--
delete employe where NUEMPL = 21;
```

B -

```

alter table concerne drop constraint FK_nuproj_concerne;
alter table travail drop constraint FK_nuproj;
alter table concerne add constraint FK_nuproj_concerne foreign key (NUPROJ) references projet (NUPROJ) initially deferred ;
alter table travail add constraint FK_nuproj foreign key (NUPROJ) references projet (NUPROJ) initially deferred ;

```

```

Create or replace trigger suppr_projet after delete on projet
BEGIN
    delete from travail where NUPROJ not in (select NUPROJ from PROJET);
    delete from   concerne where NUPROJ not in (select NUPROJ from PROJET);
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
suppr_projet	after	delete	projet	non

```

--Test--
delete projet where NUPROJ = 103;

```

Exercice 3:

A -

```

Create or replace trigger sum_duree after insert or update of duree on travail
declare
    employe_rec employe%RowType;
BEGIN
    select * INTO employe_rec from employe e where e.HEBDO < (select sum(tr.duree) from travail tr where tr.NUEMPL = e.NUEMPL);
    raise_application_error(-20003,'viol de la règle : la somme des durée de travail ne peut pas dépasser la duree hebdo');
exception
    when NO_DATA_FOUND then null;
    when TOO_MANY_ROWS then
        raise_application_error(-20004,'viol de la règle : la somme des durée de travail ne peut pas dépasser la duree hebdo');
end;

```

```

Create or replace trigger sum_hebdo after update of hebdo on employe
declare
    employe_rec employe%RowType;
BEGIN
    select * INTO employe_rec from employe e where e.HEBDO < (select sum(tr.duree) from travail tr where tr.NUEMPL = e.NUEMPL);
    raise_application_error(-20003,'viol de la règle : la somme des durée de travail ne peut pas dépasser la duree hebdo');
exception
    when NO_DATA_FOUND then null;
    when TOO_MANY_ROWS then
        raise_application_error(-20004,'viol de la règle : la somme des durée de travail ne peut pas dépasser la duree hebdo');
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
sum_duree	after	insert,update	travail	non
sum_hebdo	after	insert,update	employe	non

```

--Test--
update EMPLOYE set HEBDO = 2 where NUEMPL = 39;
update TRAVAIL set DUREE = 32 where NUEMPL = 39;

insert into EMPLOYE values(2,'titouan',2,1,2500);
insert into travail values(2,370,32);

```

B -

```

Create or replace trigger resp_max3 after update or insert on employe
declare
    employe_rec employe%RowType;
BEGIN
    select * into employe_rec from employe e where (select count(*) from projet p where e.nuempl=p.resp)> 3 ;
    raise_application_error(-20005,'viol de la règle : le nombre de projet dont l"employée est le responsable a dépassé le max

```

```

exception
when NO_DATA_FOUND then null;
when TOO_MANY_ROWS then
    raise_application_error(-20006,'viol de la règle : le nombre de projet dont l'employée est le responsable a dépassé le max
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
resp_max3	after	update,insert	employe	non

```

--Test--
insert into projet values(100,'projet',30);
update projet set RESP = 30 where NUPROJ = 135;
commit;

```

C -

```

Create or replace trigger service_max3 after update or insert on concerne
declare
    service_rec service%RowType;
BEGIN
    select * into service_rec from service s where (select count(*) from concerne c where s.NUSERV = c.NUSERV) > 3;
    raise_application_error(-20007,'viol de la règle : le service est concerné par plus de 3 projet ');
EXCEPTION
    when NO_DATA_FOUND then null;
    when TOO_MANY_ROWS then
        raise_application_error(-20008,'viol de la règle : le service est concerné par plus de 3 projet ');
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
service_max3	after	update,insert	concerne	non

```

--Test--
insert into concerne values (1,237);
update concerne set NUSERV = 1 where NUPROJ = 103;

```

D -

```

Create or replace trigger chef_salaire_plus after update or insert on employe
declare
    employe_rec EMPLOYE%rowtype;
BEGIN
    select * into employe_rec from employe e1 where e1.NUEMPL in (select s.chef from SERVICE s)
    and e1.SALAIRE < (select max(salaire) from employe e2 where e1.AFFECT = e2.AFFECT);
    raise_application_error(-20009,'viol de la règle : Un employee gagne plus qu'un chef');
EXCEPTION
    when NO_DATA_FOUND then null;
    when TOO_MANY_ROWS then
        raise_application_error(-20010,'viol de la règle : Un employee gagne plus qu'un chef');
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
chef_salaire_plus	after	update,insert	employe	non

```

--Test--
update employe set SALAIRE=3000 where NUEMPL=39;
insert into employe values(1,'titouan',25,1,3000);

```

E -

```

Create or replace trigger chef_salaire_plus_resp after insert or update of salaire on employe
declare
    employe_rec EMPLOYE%rowtype;
BEGIN
    select * into employe_rec from employe e1
        where e1.NUEMPL in (select p.resp from projet p)
    and e1.SALAIRE >
        (select max(e2.salaire) from employe e2
            where e2.NUEMPL in (select s.chef from SERVICE s ) and e1.AFFECT = e2.AFFECT);
    raise_application_error(-200011,'viol de la règle : Un employee responsable gagne plus qu'un chef');
EXCEPTION
    when NO_DATA_FOUND then null;
    when TOO_MANY_ROWS then
        raise_application_error(-20012,'viol de la règle : Un employee responsable gagne plus qu'un chef');
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
chef_salaire_plus	after	insert,update	employe	non

```

--Test--
update employe set salaire = 3000 where NUEMPL=57;
insert into employe values(1,'titouan',25,1,3000);

```

F -

```

Create or replace trigger chef_salaire_plus_resp2 after insert or update of salaire on employe
declare
    employe_rec EMPLOYE%rowtype;
BEGIN
    select * into employe_rec from employe e1
        where e1.NUEMPL in (select p.resp from projet p)
    and e1.SALAIRE >
        (select max(e2.salaire) from employe e2
            where e2.NUEMPL in (select s.chef from SERVICE s ) and e1.AFFECT = e2.AFFECT)
    or
        e1.NUEMPL in (select s.chef from SERVICE s)
        and e1.SALAIRE < (select max(salaire) from employe e2 where e1.AFFECT = e2.AFFECT);

    raise_application_error(-200013,'viol de la règle : Un employee (responsable ou pas) gagne plus qu'un chef');
EXCEPTION
    when NO_DATA_FOUND then null;
    when TOO_MANY_ROWS then
        raise_application_error(-20014,'viol de la règle : Un employee (responsable ou pas) gagne plus qu'un chef');
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
chef_salaire_plus_resp2	after	insert,update	employe	non

```

--Test--
update employe set SALAIRE=3000 where NUEMPL=39;
update employe set salaire = 3000 where NUEMPL=57;
insert into employe values(1,'titouan',25,1,3000);

```

Exercice 4:

```

CREATE Table employe_alert(
    NUEMPL NUMBER(4) not null primary key,
    NOMEMPL CHAR(20) not null,
    HEBDO NUMBER(2) not null,
    AFFECT NUMBER(2) not null,
    SALAIRE NUMBER(4) not null
);

Create or replace trigger employe_alert_add after insert or update of salaire on employe
for each row when ( new.salaire > 5000)
BEGIN

```

```

insert into employe_alert values
(:new.nuempl,:new.nomempl,:new.hebdo,:new.affect,:new.salaire);
end;

```

Nom trigger	Type trigger	Insert,delete,update	Nom table	For each row
employe_alert_add	after	insert,update	employe	oui

```

--Test--
insert into employe values(1,'titouan',25,1,5500);
update employe set SALAIRE = 5500 where NUEMPL = 20;

```