**EXEMPLES TRIGGERS**

-- Mise à jour du nombre d'employés dans la table DEPT

Alter table dept add (nb\_emp number(5) default 0) ;

DECLARE

nb number;

num dept.deptno%type;

cursor c\_num is select deptno from dept for update of nb\_emp;

BEGIN

FOR j IN c\_num LOOP

SELECT count(\*) into nb from emp where deptno=j.deptno;

UPDATE dept set nb\_emp=nb where current of c\_num;

END LOOP;

END;

/

OU

DECLARE

Cursor c\_dept is select deptno,count(\*) NB from emp

Group by deptno;

Begin

FOR I in c\_dept loop

Update dept set dept.nb\_emp=i.nb where deptno=i.deptno;

End loop;

End;

/

CREATE OR REPLACE TRIGGER MAJ\_NB\_EMP

AFTER insert or update of deptno or delete

ON EMP

FOR EACH ROW

DECLARE

BEGIN

IF inserting THEN

UPDATE DEPT set NB\_EMP = NB\_EMP + 1 WHERE deptno = :NEW.deptno;

ELSIF updating THEN

UPDATE DEPT set NB\_EMP = NB\_EMP + 1 WHERE deptno = :NEW.deptno;

UPDATE DEPT set NB\_EMP = NB\_EMP - 1 WHERE deptno = :OLD.deptno;

ELSIF deleting THEN

UPDATE DEPT set NB\_EMP = NB\_EMP - 1 WHERE deptno = :OLD.deptno;

END IF;

END;

/

--Modification de salaire

CREATE OR REPLACE TRIGGER MAJ\_SAL

AFTER UPDATE OF SAL

ON SCOTT.EMP

FOR EACH ROW

WHEN (OLD.SAL > NEW.SAL)

DECLARE

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Pas de baisse de salaire !');

END;

/

--Pour effectuer un suivi d’activité dans une table

CREATE TABLE SAISIES(numero number, date\_saisie date,

utilisateur varchar2(50));

CREATE OR REPLACE TRIGGER avant\_saisie

BEFORE INSERT ON SAISIES

REFERENCING NEW AS nouvelle

FOR EACH ROW

BEGIN

SELECT USER INTO :nouvelle.utilisateur FROM DUAL;

END;

/

CREATE TABLE STAT\_UTIL(utilisateur varchar2(50), nombre integer);

CREATE OR REPLACE TRIGGER apres\_saisie

AFTER INSERT ON SAISIES

FOR EACH ROW

DECLARE

nb stat\_util.nombre%type;

BEGIN

SELECT nombre INTO nb FROM stat\_util WHERE utilisateur=:NEW.utilisateur;

UPDATE stat\_util SET nombre=nb+1 WHERE utilisateur=:NEW.utilisateur;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

INSERT INTO stat\_util VALUES (:NEW.utilisateur,1);

END;

/

INSERT INTO SAISIES (numero,date\_saisie) VALUES (1,'26/06/06');

SELECT \* FROM SAISIES;

SELECT \* FROM STAT\_UTIL;

--Pour gérer une exception, pas de salaire inférieur à 500

CREATE OR REPLACE TRIGGER salaire\_minimum

BEFORE INSERT ON emp

FOR EACH ROW

BEGIN

IF :NEW.sal<500 THEN RAISE\_APPLICATION\_ERROR(-20002,'Salaire trop bas');

END IF;

END;

/

DECLARE

bas\_salaire EXCEPTION;

PRAGMA EXCEPTION\_INIT(bas\_salaire,-20002);

BEGIN

INSERT INTO emp(empno,ename,sal,deptno) VALUES(6000,'MARIE',400,4);

EXCEPTION

WHEN bas\_salaire THEN

dbms\_output.put\_line(SQLERRM);

END;

/

ORA-20002: Salaire trop bas

--Pour archiver les modifications des salaires des employés

create table arch\_emp (code number(5),date\_mod date, anc\_salaire number);

create or replace trigger tr\_archivage

after update of sal on emp

for each row

declare

begin

insert into arch\_emp values (:old.empno,sysdate,:old.sal);

end;

/

update emp set sal=4200 where empno=7369;

select \* from arch\_emp;

select \* from emp;

--en fonction de l’heure

create or replace trigger tr\_test\_heure

before insert or update on dept

for each row

when (to\_number(to\_char(sysdate,'HH24'))>=14)

declare

begin

dbms\_output.put\_line('Trop tard');

end;

/

update dept set loc='BREST' where deptno=40;

select \* from dept;

OU

CREATE TABLE T1(code number);

CREATE OR REPLACE TRIGGER t\_heure

BEFORE INSERT ON T1

BEGIN

IF TO\_char(sysdate,'HH24:MI') between '12:00' and '16:00' then

raise\_application\_error(-20500,'Vous devez insérer'

||' uniquement aux heures prévues');

else

dbms\_output.put\_line('Ok');

end if;

end;

/

show errors;

insert into t1 values (444);

--pour stocker des infos DDL dans la base

CREATE TABLE info\_table

(utilisateur varchar2(20),nom\_table varchar2(20),creation date);

CREATE OR REPLACE TRIGGER cre\_table

AFTER DDL ON DATABASE

BEGIN

IF (ora\_dict\_obj\_type='TABLE') THEN

IF (ora\_sysevent='CREATE') THEN

INSERT INTO info\_table VALUES (ora\_login\_user,ora\_dict\_obj\_name,sysdate);

END IF;

END IF;

END;

/

CREATE TABLE TEST (code number(5));

SELECT \* FROM info\_table;

--Suivi des erreurs

CREATE TABLE les\_erreurs

(utilisateur varchar2(50), temps date, num\_erreur varchar2(50));

CREATE OR REPLACE TRIGGER svr\_erreur

AFTER SERVERERROR ON SCHEMA

BEGIN

INSERT INTO les\_erreurs(utilisateur,temps,num\_erreur)

VALUES (ora\_login\_user,sysdate,ora\_server\_error(1));

END;

/

INSERT INTO DEPT VALUES(40);

ERRUER à la ligne 1 :

ORA-00947 : nombre de valeurs insuffisant

col utilisateur for a30;

col num\_erreur for a30;

select \* from les erreurs;

--Infos de Connect ou Disconnect

CREATE TABLE T\_connect

(utilisateur varchar2(50), date\_connect varchar2(50), action varchar2(50));

CREATE OR REPLACE TRIGGER trig\_connect

AFTER LOGON ON database

BEGIN

insert into scott.t\_connect values

(user, to\_char(sysdate,'DD/MM/YYYY HH24:MI'), 'connecte');

end;

/

CREATE OR REPLACE TRIGGER trig\_deconnect

BEFORE LOGOFF ON database

BEGIN

insert into scott.t\_connect values

(user, to\_char(sysdate,'DD/MM/YYYY HH24:MI'), 'deconnecte');

end;

/

--Trigger avec INSTEAD OF sur des vues

CREATE OR REPLACE VIEW EMP\_MANAGER

AS SELECT empno, ename, sal FROM SCOTT.EMP

WHERE upper(job) = 'MANAGER';

INSERT INTO emp\_manager VALUES (5555,'JP',6000);

--JP apparait dans la table EMP mais pas dans la vue EMP\_MANAGER.

CREATE OR REPLACE TRIGGER ins\_emp\_manager

INSTEAD OF INSERT ON EMP\_MANAGER

FOR EACH ROW

BEGIN

INSERT INTO EMP (empno,ename,job,sal)

VALUES (:NEW.empno,:NEW.ename,'MANAGER',:NEW.sal);

END;

/

INSERT INTO emp\_manager VALUES (5555,'JP',6000);

--JP apparait dans la table EMP et dans la vue EMP\_MANAGER.