# Les Triggers Geedoc Version PG

Fait le 10/04/2024

# Liste des triggers dans Geedoc



Un trigger est une action qui va s’exécuter après ou avant un évènement (un INSERT, un UPDATE ou un DELETE) dans une table.

Dans Geedoc, il n’existe que des triggers AFTER INSERT ou UPDATE sur un ou plusieurs champs

AFTER INSERT : Quand on ajoute (INSERT) un enregistrement dans une table, le trigger va ajouter ou mettre à jour une autre table.

AFTER UPDATE : quand on va modifier (UPDATE) le champ indiqué dans le trigger, le fonction va ajouter ou modifier une autre table. Exemple quand on modifie le sexe d’un animal (table ANIMAL), on garde la trace dans la table SEXE\_HIST du sexe avant et après changement.

Attention, on ne peut pas travailler sur la même table, vrai dans ORACLE en tout cas.

On ne peut pas faire un select max (d\_saillie) from saillie si le trigger est sur la table SAILLIE

# Comment créer un trigger ?











J’ai mis le code ici :

DECLARE

nb\_enreg integer:=0;

v\_last\_repro\_id geedoc.repro.repro\_id%TYPE;

v\_last\_camprepro\_id geedoc.campagne.camprepro\_id%TYPE;

v\_campagne geedoc.campagne.campagne%TYPE;

v\_saison geedoc.campagne.saison%TYPE;

v\_max\_campagne character(5); -- 4 + 1

BEGIN

 SELECT last\_camprepro\_id INTO v\_last\_camprepro\_id FROM geedoc.PRESENT

 WHERE present\_id=new.present\_id;

 IF (v\_last\_camprepro\_id IS NULL) THEN

 UPDATE geedoc.PRESENT

 SET last\_camprepro\_id=new.camprepro\_id WHERE present\_id=new.present\_id;

 ELSE

 -- Si la nouvelle campagne (id) est supérieure à la dernière campagne

 -- Est-ce vraiment vrai ? Pas tj si on rattrape des Repro

 SELECT max(CA.campagne||CA.saison) INTO v\_max\_campagne

 FROM geedoc.repro R

 INNER JOIN geedoc.campagne CA ON R.camprepro\_id=CA.camprepro\_id

 WHERE R.present\_id=new.present\_id;

 -- En déduire campagne saison

 v\_campagne:=substring(v\_max\_campagne,1,4);

 v\_saison:=substring(v\_max\_campagne,5,1);

 SELECT CA.camprepro\_id INTO v\_last\_camprepro\_id

 FROM geedoc.repro R

 INNER JOIN campagne CA ON R.camprepro\_id=CA.camprepro\_id

 WHERE R.present\_id=new.present\_id

AND CA.campagne=v\_campagne AND CA.saison=v\_saison;

 -- MAJ dans la table Present

 UPDATE geedoc.PRESENT

 SET last\_camprepro\_id=new.camprepro\_id WHERE present\_id=new.present\_id;

 END IF;

RETURN NULL;

END;

Cliquer sur « **Save** ».



Il faut ensuite lui dire comment exécuter ce trigger

create trigger tr\_maj\_repro

after insert on geedoc.repro for each row execute procedure geedoc.maj\_repro();

Quand on regarde la table Repro



## Les programmes plqsql

Le programme est composé de plusieurs parties

**DECLARE**

BEGIN

**RETURN NULL ; ou RETURN 0 ;**

**END ;**

Partie DECLARE

On déclare les variables à utiliser

// Plusieurs type integer, character , …etc..

nb\_enreg integer:=0;

// du même type de données que le champ repro\_id de la table repro du schema geedoc

v\_last\_repro\_id geedoc.repro.repro\_id%TYPE;

Entre BEGIN et END : Le programme à s’exécuter

new.<nom du champ> indique la nouvelle valeur du champ (car un fait un AFTER UPDATE ou un AFTER INSERT)

old.<nom du champ> indique l’ancien valeur du champ

 IF (old.sexe=new.sexe) pas de changement

Syntaxe :

Pour mettre une valeur dans une variable mettre ‘:=’

nb\_enreg :=0 ;

Après un IF mettre toujours THEN puis END ;

IF () THEN

Faire quelque chose

END ;

Mettre un point-virgule pour finir une ligne de code et après un END

# Que font chaque Triggers de Geedoc ?

Triggers sur la table ANIMAL



### Trigger Tr\_maj\_fact\_animal



Si la date de fin d’activité (d\_fact) et/ou le code fin d’activité (cd\_fact) est modifié (Update) :

Table MISE\_TRAITE : MAJ d\_fin\_traite=d\_fact et cse\_fin=cd\_fact

### Trigger Tr\_maj\_famille



Si la famille est modifiée

On garde la trace du changement dans la table FAMILLE\_HIST dans les 2 champs famille\_avant et famille\_apres.

INSERT INTO geedoc.famille\_hist

 (animal\_id,troupeau\_id,d\_modif,intra\_jour,famille\_avant,famille\_apres,modif\_id) VALUES(old.animal\_id,v\_troupeau\_id,current\_date,v\_intra\_jour,v\_old\_famille,v\_new\_famille,v\_modif\_id) ;

### Trigger Tr\_maj\_lignee



On garde la trace du changement de lignée dans la table LIGNEE\_HIST

INSERT INTO geedoc.lignee\_hist

 (animal\_id,troupeau\_id,d\_modif,intra\_jour,lignee\_avant,lignee\_apres,modif\_id) VALUES(old.animal\_id,v\_troupeau\_id,current\_date,v\_intra\_jour,v\_old\_lignee,v\_new\_lignee,v\_modif\_id) ;

### Trigger Tr\_maj\_sexe



On garde la trace du changement de sexe dans la table SEXE\_HIST

 INSERT INTO geedoc.sexe\_hist

 (animal\_id,troupeau\_id,d\_modif,intra\_jour,sexe\_avant,sexe\_apres,modif\_id)

 VALUES(old.animal\_id,v\_troupeau\_id,current\_date,v\_intra\_jour,old.sexe,new.sexe,v\_modif\_id) ;

## Trigger sur la table REPRO

Vu dans le Comment créer un trigger





* MAJ dans la table REPRO : On garde la dernière campagne/saison de reproduction de l’animal dans le champ last\_camprepro\_id

 UPDATE geedoc.PRESENT

 SET last\_camprepro\_id=new.camprepro\_id WHERE present\_id=new.present\_id;

### Trigger sur la table SAILLIE





Table ANIMAL : MAJ du dernier évènement lié à l’animal

Si d\_saillie > d\_etat\_repro 🡪 MAJ du champ d\_etat\_repro=d\_saillie et cd\_etat\_repro=’S’ pour cet animal animal\_id

Table REPRO : MAJ du dernier évènement intra campagne/saison

Si d\_saillie > d\_etat\_repro 🡪 MAJ du champ d\_etat\_repro=d\_saillie et cd\_etat\_repro=’S’ pour cet animal en campagne/saison repro\_id

Si c’est une IA mod\_saillie=’2’

* MAJ dans la table CAMPAGNE : MAJ de la date min des IA intra campagne/saison (d\_ia\_min) et MAJ de la date max des IA intra campagne/saison (d\_ia\_max)

### Trigger sur la table LUTTE\_NAT





### Trigger sur la table EMB\_COLL





* MAJ dans la table SAILLIE : la saillie est marquée ‘C’ pour collecté

UPDATE geedoc.saillie SET utilise='C' WHERE saillie\_id=v\_saillie\_id;

Si d\_avorte est null 🡪 v\_etat=’A’ pour Avortée et d\_etat\_repro=d\_avorte

Sinon

Si nb\_cj\_tot=0 🡪 cd\_etat=’V’ pour Vide et d\_etat\_repro=d\_collecte

Si nb\_cj\_tor >0 🡪 cd\_etat =’C’ pour Collectée et d\_etat\_repro=d\_collecte

* MAJ dans la table ANIMAL : MAJ du dernier évènement de la femelle
* MAJ dans la table REPRO : MAJ du dernier évènement de la femelle intra campagne/saison

## Trigger sur la table GESTATION



Si le résultat est ‘N’ (Négatif) ou ‘P’ (Pseudo-gestante) -- > Code=’V’ qui veut dire femelle Vide

Si le résultat est ‘G’ ->Code=’G’ qui veut dire femelle Gestante

Si le résultat est ‘A’ 🡪 code=’A’ qui veut dire femelle Avortée

MAJ du dernier évènement dans la table ANIMAL

Si d\_diagnostic > d\_etat\_repro -- >MAJ du champ d\_etat\_repro=d\_diagnostic et cd\_etat\_repro=code pour cet animal animal\_id

MAJ du dernier évènement dans la table Table REPRO :

Si d\_diagnostic > d\_etat\_repro -- >MAJ du champ d\_etat\_repro=d\_diagnostic et cd\_etat\_repro=code pour cet animal en campagne/saison repro\_id

Table CAMPAGNE : maj de la date min du DG intra campagne/saison (d\_dg\_min) et maj de la date mas intra campagne/saison (d\_dg\_max)

### Trigger sur la table MISEBAS





Rappel des codes de Mise-Bas : (mod\_mb)

 1 = Normale, 2=Difficile, 3=Pas à terme 4=Vide, 5=Césarienne, 6=Aide facile, 7=Morte+portée morte, 8=Avorté, (Revoir code=9 n'existe plus)

Si le mode de mise bas est ‘4’ 🡪 v\_code=’V’ pour Femelle **V**ide

Si le mode de mise bas est ‘7’ ou ‘8 🡪 v\_code=’A’ pour Femelle **A**vortée

Pour les autres codes 🡪 v\_code=’M’ pour Femelle qui a **M**is-Bas

* MAJ dans la table ANIMAL :
	+ MAJ du dernier évènement de la femelle

UPDATE geedoc.ANIMAL SET d\_etat\_repro=new.d\_misebas, cd\_etat\_repro=v\_code

WHERE animal\_id = v\_animal\_id

 AND ((d\_etat\_repro IS NULL) OR (d\_etat\_repro IS NOT NULL AND d\_etat\_repro<=new.d\_misebas));

* + MAJ de la dernière date de mise-bas avec portée >0 pour la femelle

-- Portée >0

UPDATE geedoc.ANIMAL SET d\_dern\_mb=new.d\_misebas

 WHERE animal\_id=v\_animal\_id

AND ((d\_dern\_mb IS NULL) OR (d\_dern\_mb IS NOT NULL AND d\_dern\_mb<=new.d\_misebas));

* MAJ dans la table REPRO : MAJ du dernier évènement intra campagne/saison

UPDATE geedoc.REPRO SET d\_etat\_repro=new.d\_misebas, cd\_etat\_repro=v\_code

WHERE repro\_id=new.repro\_id

 AND ((d\_etat\_repro IS NULL) OR (d\_etat\_repro IS NOT NULL AND d\_etat\_repro<=new.d\_misebas));

* MAJ dans la table CAMAPGNE : MAJ de la date min de mise-bas (d\_mb\_min) intra campagne/saison et MAJ de la date max de mise-bas intra campagne/saison

### Trigger sur la table PESEE





Table ANIMAL :

* MAJ de la date et de l’heure de la dernière pesée (d\_dern\_pesee et h\_dern\_pesee)

**Remarque** : je n’ai pas mis dern\_pesee\_id ou last\_pesee\_id dans la table ANIMAL car sinon il faut faire un select sur la table Pesee pour avoir la date et l’heure et pour comparer les dates et je crois qu’l y aura un problème car on va travailler sur la même table.

UPDATE geedoc.animal

SET d\_dern\_pesee=new.d\_pesee, h\_dern\_pesee=new.h\_pesee

WHERE animal\_id=v\_animal\_id;

* En fonction du code de pesée et du sexe de l’animal (que pour les femelles sexe=’2’) MAJ du dernier état physio de l’animal

Cd\_pesee=’17’ (Sevrage de la Mère) 🡪 v\_cd\_etat\_physio :='L'; Femelle en cours de Lactation

Cd\_pesee=’23’ (Femelle Tarie) 🡪 v\_cd\_etat\_physio :='T'; Femelle Tarie

UPDATE geedoc.animal SET d\_etat\_physio=new.d\_pesee, cd\_etat\_physio=v\_cd\_etat\_physio

 WHERE animal\_id=v\_animal\_id;

- Si cd\_pesee= ‘04’ (Pesée Sevrage des jeunes) 🡪 Il faut chercher la Mère Génétique (mg\_animal\_id) et si elle existe, marquer cette femelle ‘T’ (Tarie) : Bizarre de mettre Tarie demander à Karine Vazeille

 UPDATE geedoc.animal SET d\_etat\_physio=new.d\_pesee, cd\_etat\_physio='T'

 WHERE id\_animal=v\_mg\_animal\_id

 AND ((d\_etat\_physio IS NULL) OR (d\_etat\_physio < new.d\_pesee));

### Trigger sur la table PRESENT



Si c’est un mouvement d’animaux (transfert=’F’ 🡪 v\_typ\_sortie=’M’

Sinon (Réforme) v\_typ\_sortie=’R’

MAJ dans la table LOT\_PHYSIQUE : l’animal est marqué sorti du dernier lot physique

 UPDATE geedoc.lot\_physique

 SET d\_sortie=new.d\_sortie, typ\_sortie=v\_typ\_sortie

 WHERE present\_id=old.present\_id

 AND (d\_sortie IS NULL OR (d\_sortie IS NOT NULL AND d\_sortie=old.d\_sortie));

MAJ dans la table ANIM\_EXPE : l’animal est marqué sorti du dernier lot expérimental pour toutes les expérimentation en cours.

UPDATE geedoc.anim\_expe

 SET d\_sortie\_expe=new.d\_sortie, typ\_sortie=v\_typ\_sortie

 WHERE present\_id=old.present\_id AND dern\_expe='D'

 AND (d\_sortie\_expe IS NULL OR (d\_sortie\_expe IS NOT NULL AND d\_sortie\_expe=old.d\_sortie));

MAJ dans la table ANIM\_MODCOND : l’animal est marqué sorti du dernier mode de conduite

UPDATE geedoc.anim\_modcond

 SET d\_sortie\_modcond=new.d\_sortie, typ\_sortie=v\_typ\_sortie

 WHERE present\_id=old.present\_id AND dern\_modcond='O'

 AND (d\_sortie\_modcond IS NULL OR (d\_sortie\_modcond IS NOT NULL AND d\_sortie\_modcond=old.d\_sortie));

MAJ dans la table ANIM\_LOC : l’animal est marqué sorti de la dernière localisation

 UPDATE geedoc.anim\_loc

 SET d\_sortie\_loc=new.d\_sortie, typ\_sortie=v\_typ\_sortie

 WHERE present\_id=old.present\_id AND dern\_loc='O'

 AND (d\_sortie\_loc IS NULL OR (d\_sortie\_loc IS NOT NULL AND d\_sortie\_loc=old.d\_sortie));

### Trigger sur la table ANIM\_NOURRICE

Cette table contient les changements de nourrice (Mère Allaitante)



Déclenchement : Events



MAJ dans la table ANIMAL : champ d\_chg\_nourrice pour l’animal

-- Si la date d\_chg\_nourrice est inférieure à celle à ajouter

 IF (v\_d\_nourrice<new.d\_nourrice) THEN

UPDATE geedoc.animal SET d\_chg\_nourrice=new.d\_nourrice WHERE animal\_id=new.animal\_id;

 END IF;

### Trigger sur la table MISE\_TRAITE





* Table CAMPAGNE\_TRAITE : MAJ de la dernière date de traite (d\_fin\_traite) intra campagne de traite.

Rappel des codes :

Cse\_fin=’023’ 🡪 Tarie

Cse\_fin=’115’ 🡪 Lactation Longue

IF ((new.cse\_fin='023') OR (new.cse\_fin='115')) THEN

-- MAJ de la date de fin de la campagne de traite dans ma table CAMPAGNE\_TRAITE

 UPDATE geedoc.campagne\_traite

 SET d\_fin\_camptraite=new.d\_fin\_traite

 WHERE camptraite\_id=old.camptraite\_id

 AND ((d\_fin\_camptraite IS NULL) OR (d\_fin\_camptraite< new.d\_fin\_traite)) ;

 END IF;