Trigger de la table ETAPE : **trg_update_historique**

Auteur : Thibaut MESLIN
Objectif du trigger :

Garder une trace des précédentes informations concernant les normes dans un table dédiée (HISTORIQUE).

Cas de déclenchement :

Se déclenche quand la colonne ETP_NORME et/ou ETP_DATE_NORME de la table ETAPE sont modifiés.

Paramètre(s) d'entrée :

- @dateMAJ → date : *date de la mise à jour (date système)*
- @util → int : *id de l'utilisateur ayant fait la modification*
- @numEtape → int : *numéro de l'étape qui a été modifié*
- @normeAvant → varchar de longueur 20 : *norme précédente (avant insertion)*
- @normeApres → varchar de longueur 20 : *norme actuelle (après insertion)*
- @dateAvant → date : *date de la norme précédente (avant insertion)*
- @dateApres → date : *date de la norme actuelle (après insertion)*

Requête(s) utile(s) au(x) variable(s) :

- SELECT @dateApres = ETP_DATE_NORME, @normeApres = ETP_NORME, @numEtape = ETP_NUM from inserted
  - *Récupérer les nouvelles informations (nouvelle norme et date) et le numéro de l'étape modifiée*
- SELECT @normeAvant = ETP_NORME, @dateAvant = ETP_DATE_Norme from deleted
  - *Récupérer les informations supprimées (ancienne norme et date)*
- SELECT @util = UTL_NUM from UTILISATEUR
  - *Récupérer le numéro de l'utilisateur*
- SELECT @dateMAJ = convert (date, SYSDATETIME())
  - *Récupérer la date du système*

Requête(s) utile(s) à l'objectif :

- INSERT INTO HISTORIQUE(HST_DATE_MODIF, HST_ETP_DATE_BEFORE, HST_DATE_AFTER, HST_UTL, HST_ETP_NUM, HST_ETP_NORME_BEFORE, HST_ETP_NORME_AFTER) VALUES (@dateMAJ, @dateAvant, @dateApres, @util, @numEtape, @normeAvant, @normeApres)
  - *Insertion d'une nouvelle ligne dans la table HISTORIQUE contenant toutes les informations souhaitées*

Trigger de la table ETAPE : **trg_insert_subir**

Auteur : Benjamin GERBE
Objectif du trigger :

Met à jour la dernière étape d'un médicament après l'insertion d'un nouveau workflow dans la table subir.

Cas de déclenchement :

Se déclenche après l'insertion dans la table subir.

Paramètre(s) d'entrée :

- @depotlegal → nvarchar de longueur 10: *dépôt légal du médicament*
- @numeroEtp → int : *numéro de l'étape ajouté*

Requête(s) utile(s) au(x) variable(s) :

- SELECT @depotlegal = S.MED_DEPOTLEGAL_SUB, @numeroEtp = S.ETP_NUM_SUB from inserted s
  - *Initialise le dépôt légal et la dernière étape.*

Requête(s) utile(s) à l'objectif :

- UPDATE dbo.MEDICAMENT
  set ETP_NUM_MED = @numeroEtp
  where MED_DEPOTLEGAL = @depotlegal
  - *Met à jour la table médicament*

Trigger de la table ETAPE : **trg_medicament_insert**

Auteur : Benjamin GERBE
Objectif du trigger :

Met à jour le nombre de médicament sur le marché dans la table famille

Cas de déclenchement :

Après la mise à jour de la classe médicament

Paramètre(s) d'entrée :

- @codeFamille → nvarchar de longueur 255: *code famille du médicament*
- @etape → int : *numéro de l'étape mis à jour*

Requête(s) utile(s) au(x) variable(s) :

- SELECT
  @codefamille = i.FAM_CODE_MED,
  @etape = i.ETP_NUM_MED
  from inserted i
  - *Initialise le code famille et l'étape.*

Requête(s) utile(s) à l'objectif :

- Update FAMILLE
  set FAM_NB_MEDI_AMM = (select COUNT(*) from MEDICAMENT M where M.FAM_CODE_MED = @codefamille and M.ETP_NUM_MED >= 8)
  where FAM_CODE = @codefamille
  - *Met à jour la table famille*