

TP 3 – Triggers

Fichier de réponses

Pour conserver trace de votre travail et pouvoir discuter de vos solutions avec votre enseignant, vous écrirez les énoncés des questions (en commentaires entre /* et */) suivis de vos solutions dans un fichier `~/M2106/tp3.sql`. Nous vous recommandons d'éditer ce fichier puis d'effectuer des copier-coller dans le terminal où s'exécute votre connexion postgresql.

A. Triggers sur tables

Question 1 : (*Contrainte d'intégrité*) Un adhérent qui a l'agrément skipper ne peut pas le perdre. Ecrire un **trigger** et sa **procédure** qui assurent cette contrainte d'intégrité.

Expériences : Faire une mise à jour d'adhérent pour tenter d'enlever l'agrément skipper à un adhérent, puis pour donner l'agrément à un adhérent qui ne le possédait pas.

Question 2 : (*Contrainte d'intégrité Référentielle*) Lorsque l'on supprime une activité liée à des inscriptions, que ce soit une activité future que l'on annule, ou une activité passée lors d'une purge de la base, une contrainte de clé étrangère lève une erreur. Ecrire un **trigger** et sa **procédure** qui, lors de la suppression d'une activité, suppriment préalablement tous les enregistrements dépendant de l'activité à supprimer, afin qu'aucune exception ne soit levée (attention également à l'ordre entre ces suppressions).

Expériences : Supprimer le rallye créé lors du TP1 (question 5) et constater que la suppression a réussi, entraînant avec elle la suppression des 4 régates.

Question 3 : On souhaite conserver l'historique du nombre total d'activités au cours du temps dans une table `historique_activites(heure,nbact)` qui enregistre la date et l'heure (obtenues grâce à la fonction `now()` renvoyant un résultat de type `timestamp`) de chacune des instructions faisant évoluer le nombre d'activités, ainsi que ce nombre d'activités à leur issue.

Expériences :

- Créer cette table `historique_activites(heure,nbact)` ;
- Créer un trigger qui se déclenche `FOR EACH ROW` et le tester en créant deux sorties pour 2018, puis en les supprimant par un seul `DELETE`. En quoi cette solution n'est pas satisfaisante ? Supprimer ce trigger et les lignes insérées dans `historique_activites`.
- Créer un deuxième trigger qui se déclenche `FOR EACH STATEMENT`, et le tester de la même manière que précédemment. Qu'observez-vous ?

B. Triggers sur vues

Question 4 : Créer une **vue** `VAdherentsDebiteurs` contenant la liste des numéros des adhérents qui n'ont pas terminé (attribut `paye = 'non'`) de payer leur cotisation en 2019.

Expériences : Visualiser le résultat de cette vue, puis écrire une requête permettant d'obtenir le nom de ces adhérents.

Question 5 : Suite au courrier de relance qu'il reçoit, l'un de ces adhérents (par exemple le numéro 4) paye le restant de sa cotisation pour 2019. Modifier la table `Cotisation` pour prendre en compte cette modification. **Expériences :** Vérifier que cet adhérent ne figure plus dans la vue `VAdherentsDebiteurs`.

Question 6 : Créer une **vue** `VAgencesMarseille` contenant la liste des agences de Marseille. Insérer dans cette vue l'Agence 'La Tour Eiffel' qui vient de s'installer à Paris. L'insertion s'est-elle traduite dans la table `Agence` ? Pourquoi ? Cette agence est-elle toujours visible par cette vue, après insertion ? Pourquoi ?

Question 7 : Créer un **trigger** et sa **procédure** permettant d'insérer une nouvelle agence dans la liste des agences de Marseille (l'adresse de ces agences est donc forcément 'Marseille...').

Expériences : Insérer dans la vue **VAgencesMarseille** la nouvelle agence qui vient de s'installer à Marseille près du Vieux Port : 'Le Vieux Port', dont le numéro de téléphone est le 0400000000 et le fax est le 0499999999. Vérifier que cette agence a été insérée dans la table **Agence**.

C. BONUS

Question 8 : (*Contrainte d'intégrité*) La force du vent d'une régates ne peut être connue que si l'activité a commencé. Dans le cas contraire :

- s'il s'agit d'une insertion, on impose la valeur NULL à la force du vent et on en prévient l'utilisateur ;
- s'il s'agit d'une mise à jour, on lève une exception.

Ecrire un **trigger** et sa **procédure** qui assurent cette contrainte d'intégrité.

Expériences : Créer un nouveau rallye commençant dans 10 jours, puis tenter d'ajouter à ce rallye une régates avec une valeur (non NULL) pour la force du vent. Observer ce qui a été effectivement inséré. Tenter de mettre à jour cette régates afin de donner une valeur (non NULL) à la force du vent.

Question 9 : Créer une **vue VRésultats** qui donne les résultats des bateaux aux rallyes, ainsi que les informations sur ces rallyes (notez qu'une activité pour laquelle des résultats ont été enregistrés est nécessairement un rallye). Par défaut, pouvez-vous modifier le nombre de points obtenu par un bateau à un rallye via cette vue ? Pourquoi ? Pour corriger cela, créer un **trigger** et sa **procédure** qui, en cas de modification du classement ou du nombre de points d'un bateau à un rallye, répercute ces modifications dans la table **Resultat**.

Expériences : Modifier le résultat d'un bateau (par exemple le numéro 4) à une régates (1) d'un rallye (5) grâce à cette vue et vérifier que cette modification a été répercute dans la table **Resultat**.

Question 10 : Créer une **vue VAdherentsSkippers** contenant la liste des adhérents skippers (numadh, nom et prénom). Créer ensuite un **trigger** et sa **procédure** pour pouvoir insérer un nouvel adhérent via cette vue.