TRIGGERS

Procédures associées à un événement de mise à jour sur une relation. Ces procédures sont automatiquement exécutées par le SGBD lorsque l'événement associé se produit. Ils assurent un contrôle d'intégrité dynamique et répondent aussi au besoin de base de données active, c'est-à-dire qui réagit à un événement. Il se produit ceci, j'insère cela dans telle relation…

Bien pratiques : Des triggers pour :

Appliquer des contraintes d'intégrités dynamiques

Les contraintes d'intégrité dynamiques ou comportementales désignent des règles à suivre lors de mises à jour. Les triggers apportent une solution souple à l'implémentation de ces CI.

Exprimer la règle de gestion suivante : un salaire ne peut pas diminuer.

Mettre à jour des attributs calculés

L'attribut calculé contient une information immédiatement disponible, information bâtie à l'aide d'une expression impliquant des attributs de plusieurs relations. Il évite des requêtes longues ou complexes lors de la recherche de cette information. Un inconvénient majeur des attributs calculés concerne le risque d'incohérence entre la valeur stockée et le calcul de cette valeur. Le trigger a ici pour but d'assurer une mise à jour systématique de l'attribut calculé et d'éviter ainsi les incohérences.

Définir l'attribut calculé MONTANT de la relation COMMANDE tout en assurant sa mise à jour.

Réaliser un suivi de l'activité

Il est parfois nécessaire de répertorier toutes les modifications effectuées sur une relation importante. Qui a fait quoi ? Quand ? La traçabilité impose de pouvoir savoir qui a effectué une manipulation dans la base et à quel moment cela a été effectué.

Enregistrer toute évolution de prix dans la relation PRODUIT.
Qui a effectué la manipulation, Quelle modification ? Quand ?

Supprimer en cascade

Mais coûteux :

L'inconvénient des triggers est qu'ils sont coûteux en ressource.

On pensera à utiliser des procédures stockées. Dans ces procédures, est généré l'evt et les vérifications ou mise à jour associées.

Syntaxe :

```
CREATE TRIGGER  nomTrigger
      BEFORE / AFTER                les actions du trigger sont exécutées av/ap l'evt bd
      DELETE / INSERT / UPDATE [OF nomAtt1 [, nomAtti ] ]
      [ OR …]
      ON [schema.] relation
      [FOR EACH ROW]                                  /  FOR EACH STATEMENT
      WHEN condition                                  limiter à une portion de la relation
            …
            Bloc PL/SQL
            …
END  nomTrigger ;
```

:NEW.nomAtt / :OLD.nomAtt
Utilisation // Ne peuvent s'employer qu'avec FOR EACH ROW