

Exploitation d'une base de données

Département Informatique

IUT2 de Grenoble

BUT1 - Ressource 2.06

Introduction à l'exploration et l'administration des BD

1. Révisions : BD relationnelle et SQL simple
2. SQL pour explorer et analyser
3. SQL augmenté : *triggers*
4. Administration de 1^{er} niveau

3. SQL augmenté : *triggers*

3.1. Triggers et fonctions simples

- Introduction
- Fonction en PL/pgSQL
- Trigger sur table
- Pptés : déclenchement, var. prédéf., validation/annulation

3.2. Triggers sur vues

3.3. Fonctions avancées en PL/pgSQL

- Déclaration de variables
- Instructions

Vues modifiables(1/2)

Vue simple : peut être l'objet de **INSERT**, **UPDATE**, **DELETE**

- ✓ si répercussion de la modif. sur la table d'origine sans ambiguïté :
 - pas de jointure
 - pas de **DISTINCT**, **GROUP BY**, **HAVING**, **LIMIT**, ou **OFFSET**
 - pas de **UNION**, **INTERSECT**, ou **EXCEPT**
 - pas de fonction d'agrégation

Vues modifiables(2/2)

⚠ Attention aux incohérences :

- modif. d'un tuple qui n'est ou ne sera pas visible dans la vue.
- peut être interdit avec **WITH CHECK OPTION** ou un ***trigger***.

Pour les vues non simples

modifications possibles avec un ***trigger***.

Triggers sur vues

Syntaxe

```
CREATE TRIGGER nom_trigger  
INSTEAD OF  
{ INSERT | UPDATE | DELETE [ OR ... ] }  
ON nom_vue  
FOR EACH ROW  
EXECUTE FUNCTION nom_fonc();
```

Principe

- Une modification de la vue déclenche une fonction qui réalise des actions sur les tables d'origine.
- Le n-uplet renvoyé *informe* le contexte :
 - **null** = aucune modification n'a eu lieu ;
 - **new/old** sinon .