

Bases de données

Triggers

Silviu Maniu

2022/2023

Polytech App3

Trigger (déclencheur) – permet exécution automatique d'une *fonction* une fois q'une certain type d'opération est exécuté

- attaché a une table ou une vue
- **avant** ou **après** commandes INSERT, UPDATE ou DELETE
- **par ligne** ou **par instruction**

Triggers – syntaxe de base

```
CREATE TRIGGER nom
  BEFORE | AFTER | INSTEAD OF  evenement
  ON nom_table
  [FOR EACH ROW ]
  EXECUTE nom_fonction ( arguments );
```

Triggers – ingredients

BEFORE – le trigger se déclenche avant la modification de la table

AFTER – le trigger se déclenche après la modification de la table

FOR EACH ROW – le trigger se déclenche pour toute ligne résultat
(p.ex., `SELECT WHERE`, `UPDATE WHERE`) *niveau ligne*

evenement – spécification de l'instruction de déclenchement `INSERT`,
`UPDATE`, `DELETE`

Triggers – fonctionnement

Visibilité :

- *niveau instruction* : toutes les modifications sont visible aux trigger **AFTER**, aucune aux triggers **AVANT**
- les instructions modification ne sont pas visible dans un déclencheur **BEFORE** en mode ligne
- quand un trigger **AFTER** est exécuté, toutes les modifications sont visibles

Accès à deux **record** :

- **OLD**, le contenu de la ligne avant modifications
- **NEW**, le contenu de la ligne apres modifications

Fonction trigger

Les déclencheur nécessitent la spécification d'une fonction par exemple PL/pgSQL, avec des restrictions :

- doit toujours avoir RETURNS TRIGGER
- les paramètres doivent être vides ou des varchar
- doit retourner OLD/NEW ou NULL (annule les modifications)

```
CREATE OR REPLACE FUNCTION function_trigger()  
RETURNS TRIGGER  
AS $$  
DECLARE  
    ...  
BEGIN  
    --order SQL. boucles, etc. en utilisant OLD et NEW  
    RETURN NEW;  
END;  
$$ language plpgsql;
```

Trigger – exemple (1/2)

Table *Etudiant*(*nom*,*moyenne*,*mention*) – ajout automatique de la mention

```
CREATE OR REPLACE FUNCTION mention()  
RETURNS TRIGGER  
AS $$  
BEGIN  
    IF NEW.MOYENNE>16 THEN NEW.MENTION='Tres bien';  
    ELSIF NEW.MOYENNE>14 THEN NEW.MENTION='Bien';  
    ELSE NEW.MENTION='<aucun>';  
    END IF;  
    RETURN NEW;  
END;  
$$ language plpgsql;  
CREATE TRIGGER trg_mention BEFORE INSERT ON etudiant  
FOR EACH ROW  
EXECUTE FUNCTION mention();
```

Trigger – exemple (2/2)

Table *Etudiant*(*nom,moyenne,mention*) – maximum 10 étudiants

```
CREATE OR REPLACE FUNCTION max10()  
RETURNS TRIGGER  
AS $$  
DECLARE  
    compte INT;  
BEGIN  
    SELECT COUNT(*) INTO compte FROM etudiant;  
    IF compte >= 10 THEN  
        RETURN NULL;  
    END IF;  
    RETURN NEW;  
END;  
$$ language plpgsql;  
CREATE TRIGGER trg_compte AFTER INSERT ON etudiant  
EXECUTE FUNCTION max10();
```