

# 420-C42

Langages d'exploitation des bases de données

# Partie 19

PL/pgSQL III

Gestion des exceptions et imbrication de blocs

# PL/pgSQL III

## Gestion d'exceptions

- Il est possible de *produire* des messages de priorité variable avec la clause RAISE.
- Il existe 6 types de priorité:
  - DEBUG – information de débogage
  - LOG – information de journalisation
  - INFO – information générique
  - NOTICE – un avis
  - WARNING – avertissement
  - EXCEPTION (la valeur par défaut) – une erreur fatale
- Dans tous les cas, les messages produits sont reportés au programme appelant après l'exécution de la requête.
- Sauf pour la priorité EXCEPTION, les opérations en cours se poursuivent.
- Toutefois, pour la priorité EXCEPTION, les opérations en cours sont interrompues et un ROLLBACK est effectué. (*lève une exception*)

# PL/pgSQL III

## Gestion d'exceptions

- Le synopsis simplifié est :

```
RAISE [ level ] 'format' [, expression [, ... ]]  
      [ USING option = expression [, ... ] ];
```

- *level* est l'un des 6 niveaux de priorité mentionné.
- *format* est le texte à produire. On utilise le caractère % pour effectuer la substitution de valeur avec les *expressions* qui sont passés en paramètres par la suite.
- *option* permet de préciser des paramètres spécifiques du message :
  - MESSAGE – le message
  - DETAIL – des détails sur le message
  - HINT – des indices liés au message
  - ERRCODE – retourne un code d'erreur spécifique (voir [les messages existants](#))

- Il est important de savoir que PostgreSQL utilise directement ce mécanisme pour toutes ses opérations. Il est ainsi possible d'écrire une requête SQL ou des portions de programmes PL/pgSQL et de recevoir des informations.
- Lorsqu'une requête est refusée :
  - une *exception* est lancée
  - la transaction est annulée (« *rollback* »)
  - le retour nous indique que la requête a été refusée

# PL/pgSQL III

## Gestion d'exceptions

- En PL/pgSQL, le bloc EXCEPTION permet de réagir aux exceptions sans nécessairement annuler la requête en cours.

```
... $$  
DECLARE
```

```
BEGIN
```

```
EXCEPTION
```

```
    WHEN condition THEN
```

```
        ... opération à faire ...
```

```
    WHEN condition THEN
```

```
        RAISE NOTICE ...;
```

```
    WHEN OTHERS THEN
```

```
        ... opérations à faire ...
```

```
        RAISE;
```

```
END$$;
```

-- un exemple où une exception devient un avis

-- attrape toutes les autres exceptions

-- exemple ou on relance la même exception

# PL/pgSQL III

## Imbrication de blocs PL/pgSQL

- Le PL/pgSQL permet d'imbriquer des blocs dans des blocs.
- Cette possibilité permet :
  - de clarifier du code plus complexe en le compartimentant
  - de déclarer un espace de variable spécifique
  - les sous-blocs peuvent accéder aux variables des blocs de niveau supérieur mais l'inverse n'est pas possible
  - de gérer des exceptions spécifiques à certaines opérations

# PL/pgSQL III

## Imbrication de blocs PL/pgSQL

```
...  
DECLARE  
BEGIN  
    -- opérations niveau 1  
    DECLARE  
    BEGIN  
        -- opération niveau 2  
        DECLARE  
        BEGIN  
            -- opérations niveau 3  
            EXCEPTION  
            END;  
        -- opérations niveau 2  
        EXCEPTION  
        END;  
    -- opérations niveau 1  
    EXCEPTION  
END;
```