

```
-- illustration de l'utilisation des exceptions

-- on ajoute dans la base le fait qu'un joueur est à un poste
-- Code vu en cours :
CREATE OR REPLACE FUNCTION inserer(nomJoueur joueur.nom%type, nomPoste poste.libelle%type)
RETURNS boolean AS $$
DECLARE
numPoste integer ; numJoueur integer ;
tester record ;

BEGIN
-- recherche du poste
SELECT num INTO numPoste from poste where libelle=nomPoste ;
IF NOT FOUND THEN -- creation poste
    insert into poste(libelle) values (nomPoste) RETURNING num INTO numPoste;
END IF ;
-- recherche du joueur
SELECT num INTO numJoueur FROM joueur WHERE nom=nomJoueur ;
IF NOT FOUND THEN -- creation joueur
insert into joueur (nom) values (nomJoueur) RETURNING num INTO numJoueur ;
END IF ;

-- mise a jour Joueur_poste
SELECT * INTO tester FROM Joueur_poste WHERE numj=numJoueur AND nump=numPoste ;
IF FOUND THEN RAISE EXCEPTION 'déjà inséré' ; --> on déclenche une exception
END IF ;
insert into Joueur_poste values (numJoueur, numPoste);
return true ;
END ;$$
LANGUAGE 'plpgsql' ;

-- V2 on laisse le serveur traiter l'erreur de doublon
CREATE OR REPLACE FUNCTION inserer(nomJoueur joueur.nom%type, nomPoste poste.libelle%type)
RETURNS boolean AS $$
DECLARE
numPoste integer ; numJoueur integer ;
tester record ;

BEGIN
-- recherche du poste
SELECT num INTO numPoste from poste where libelle=nomPoste ;
IF NOT FOUND THEN -- creation poste
    insert into poste(libelle) values (nomPoste) RETURNING num INTO numPoste;
END IF ;
-- recherche du joueur
SELECT num INTO numJoueur FROM joueur WHERE nom=nomJoueur ;
IF NOT FOUND THEN -- creation joueur
```

```
insert into joueur (nom) values (nomJoueur) RETURNING num INTO numJoueur ;
END IF ;

-- mise a jour Joueur_poste, ici je laisse le serveur déclencher l'erreur
insert into Joueur_poste values (numJoueur, numPoste);
return true ;
END ;$$
LANGUAGE 'plpgsql' ;

-- V3 on récupère l'exception du serveur'
CREATE OR REPLACE FUNCTION inserer(nomJoueur joueur.nom%type, nomPoste poste.libelle%type)
RETURNS boolean AS $$
DECLARE
numPoste integer ; numJoueur integer ;
tester record ;

BEGIN
-- recherche du poste
SELECT num INTO numPoste from poste where libelle=nomPoste ;
IF NOT FOUND THEN -- creation poste
    insert into poste(libelle) values (nomPoste) RETURNING num INTO numPoste;
END IF ;
-- recherche du joueur
SELECT num INTO numJoueur FROM joueur WHERE nom=nomJoueur ;
IF NOT FOUND THEN -- creation joueur
insert into joueur (nom) values (nomJoueur) RETURNING num INTO numJoueur ;
END IF ;

-- mise a jour Joueur_poste, ici je traite l'erreur déclenchée par le serveur
begin
    insert into Joueur_poste values (numJoueur, numPoste);
    return true ;
exception
    when unique_violation then RAISE EXCEPTION 'déjà inséré' ;
end ;
END ;$$
LANGUAGE 'plpgsql' ;

delete from joueur_poste where numj=15;

select inserer('Diacre','entraîneur.e'); -- Corinne Diacre, entraîneure/sélectionneure
-- renvoie TRUE
-- si on réexécute, déclenche une erreur

/* avec la V2, l'erreur est celle du serveur
Erreur SQL :
```

ERREUR: la valeur d'une clé dupliquée rompt la contrainte unique « joueur_poste_pkey »

DETAIL: La clé « (numj, nump)=(15, 6) » existe déjà.

CONTEXT: instruction SQL « insert into Joueur_poste values (numJoueur, numPoste) »

fonction PL/pgsql inserer(character varying,character varying), ligne 19 à instruction SQL

Dans l'instruction :

```
select inserer('Diacre','entraîneur.e');
```

```
*/
```

```
/*
```

avec la V3, on a :

ERREUR: déjà inséré

CONTEXT: fonction PL/pgsql inserer(character varying,character varying), ligne 23 à RAISE

```
*/
```