

# Introduction aux Bases de Données

Ulysse COUTAUD  
ulyse.coutaud@gmail.com

# Désactiver temporairement une contrainte

**Problème:** Les contraintes peuvent rendre très difficile voire impossible les manipulations de la BDD.

**Solution:** Désactiver temporairement les contraintes (cad le temps d'une transaction).

```
SET CONSTRAINTS ma_contrainte DEFERRED;
```

Remarque: PostgreSQL ne permet pas de différer les contraintes NOT NULL et CHECK.

# Désactiver temporairement toutes les contraintes

```
SET CONSTRAINTS ALL DEFERRED;
```

*Mettre à jour la BDD fabrique. Observer les contraintes. Faire une transaction pour intégrer à la base fabrique les employés Ozgur et Marie dans le nouveau service "IT".*

# Valeurs par default

## Autoriser/interdire le DEFERRED.

A la creation de la table / contrainte:

- DEFERRABLE
- NOT DEFERRABLE

## Contrôle de la contrainte par default:

A la creation de la table / contrainte:

- INITIALLY DEFERRED
- INITIALLY IMMEDIATE

```
ALTER TABLE ma_table  
ADD CONSTRAINT ma_nouvelle_contrainte  
FOREIGN KEY (ma_colonne) REFERENCES mon_autre_table(autre_colonne)  
DEFERRABLE INITIALLY DEFERRED;
```

# Extraire les résultats des requêtes

## La commande \o

```
\o mon_fichier_de_sortie  
... mes commandes sql ...  
\o -- Désactive la sortie vers le fichier, retour à la ligne de  
commande.
```

## La commande \copy

```
\COPY (SELECT * FROM employes) TO  
'./mon_fichier_de_sortie.csv' WITH DELIMITER AS ',' CSV  
HEADER;
```

# Exécuter un script directement depuis la console

*Cad sans être au préalable connecté dans l'interface psql.*

## L'option -f

```
psql -f mon_fichier_de_script.sql ma_base_de_donnees  
mon_utilisateur
```

- Se connecte à la base de donnée,
- exécute le script *mon\_fichier\_de\_script.sql*,
- puis quitte.

# Passer le mot de passe en ligne de commande

```
psql postgresql://utilisateur:mot_de_passe@localhost:5432/ma_bdd
```

Dans notre cas:

```
psql postgresql://postgres:postgres@localhost:5432/fabrique
```

- *Faire un script (batch), exécutable par une simple double clic, qui extrait la table employees de la BDD fabrique, au format csv.*
- *Ajouter un fichier de log doit être générer afin de pouvoir vérifier la bonne exécution du script.*

# Les variables psql: assignation et utilisation

## Déclarer et assigner une variable dans l'interface psql

```
\set nom_de_variable valeur
```

## Utiliser une variable

```
SELECT * FROM ma_table WHERE  
colonne1=:nom_de_variable;
```

Remarque: Pour une variable de type texte: `SELECT * FROM ma_table WHERE colonne1=: 'nom_de_variable';`

*Faire un script qui crée un nouveau service, y ajoute 1 employé et le déclare comme chef du service. Utiliser des variables déclarées et assignées en début de script.*



# Les variables psql: assignments "avancée"

Déclarer et assigner une variable dans l'interface psql

```
\set mon_nom_de_variable valeur
```

Déclarer et assigner une variable en argument de la connexion à psql

```
psql -v var1='Ulysse' -v variable2=130 -v v3="'Maintenance'"
```

Lire dynamiquement une variable dans un script psql

```
\prompt 'Svp, entrez la valeur pour variable1: ' variable1
```

*Reprendre le script précédent et le rendre dynamique. Cad l'utilisateur doit taper les valeurs à ajouter.*

Sur la base fabrique:

- *Faire un script qui demande à taper le nom d'un employé, et retourne les horaires de cet employés.*

Sur votre base atelier:

- *Faire un script qui demande à taper le nom d'un employé, et retourne la liste des machines sur lequel il est formé.*
- *Faire un script qui demande à taper le nom d'une machine, et retourne la liste des employes qui sont formés à l'utiliser.*