

# Introduction aux Bases de Données

Ulysse COUTAUD  
ulyse.coutaud@gmail.com

# Modifier des données

# Insérer un enregistrement

BDD "atelier":

employes(prenom,salaire,adresse,service) horaires(service,debut,fin)

## Clause INSERT INTO

```
INSERT INTO table VALUES ('valeur 1', 'valeur 2', ...)
INSERT INTO table (nom_colonne_1, nom_colonne_2, ...)
VALUES ('valeur 1', 'valeur 2', ...)
```

*Ajouter un nouvel employé dans le service "Production"*

*Ajouter un service "Production Après-Midi" dont les horaires seront 13h-20h.*

*Ajouter les horaires 16h-5h pour le service "Astreinte"*

*Ajouter un nouvel employé "Abdoul" au service "Production Après-Midi".*

## Modifier un enregistrement

## BDD "atelier":

employees(prenom,salaire,adresse,service) horaires(service,debut,fin)

## Clause UPDATE

```
UPDATE table SET colonne_1 = 'valeur 1', colonne_2 = 'valeur 2', ... WHERE conditions;
```

*Augmenter le salaire des employes de "Astreinte" de 30%*

*L'employé Tom change de service et passe à l'après midi*

# Supprimer un enregistrement

BDD "atelier":

employees(prenom,salaire,adresse,service) horaires(service,debut,fin)

## Clause DELETE FROM

DELETE FROM table WHERE conditions;

*Supprimer le service de production après-midi.*

# Modifier le schéma



# Supprimer une table

**DROP TABLE**

```
DROP TABLE ma_table;
```

*Supprimer la table horaire.*



# Modifier une table

## ALTER TABLE

ALTER TABLE ma\_table mon\_action ;

Avec mon\_action:

- ADD ma\_colonne INTEGER/TEXT/...
- DROP ma\_colonne

*Ajouter la colonne quantite (le nombre de pièces à produire) à la table ordre\_fabrication.*

# Les contraintes

# Les contraintes

- Règles qui restreignent les données dans la BDD.
- Chaque contrainte est associée à une table.
- Bloque tout ajout/modification qui viole les contraintes.
- Garantis **certains** aspects de la cohérence de la BDD.





# Clé Secondaire

## UNIQUE

CONSTRAINT *ma\_clé\_secondaire* UNIQUE (*colonne1*)

Unique mais peut être nulle.

*Avec la commande psql "*

*d+ employes", trouver une clé secondaire de la table employes et expliquer sont sens.*

# Clé étrangère

## FOREIGN KEY

```
CONSTRAINT ma_clé_étrangère FOREIGN KEY (colonne1)  
REFERENCES table_étrangère(colonneX))
```

Assure l'intégrité référentielle.

Existe nécessairement dans la table de référence.

## Et si la clé de référence disparaît ?

Trois options:

- ON DELETE RESTRICT
- ON DELETE CASCADE
- ON DELETE SET NULL

# Check

## CHECK

CONSTRAINT *mon\_check* CHECK (condition)

Remarque: Même genre de conditions que dans un WHERE. *En*

*lisant le fichier fabriqueBDD.sql: quelle contrainte CHECK existe sur la table employes ?*

*Quelle contrainte devrait on ajouter sur la colonne salaire ?*



# Les transactions

# Le problème

```
UPDATE accounts SET balance = balance - 100.00
WHERE name = 'Alice';
```

```
UPDATE branches SET balance = balance - 100.00
WHERE name = (SELECT branch_name
               FROM accounts
               WHERE name = 'Alice');
```

```
UPDATE accounts SET balance = balance + 100.00
WHERE name = 'Bob';
```

```
UPDATE branches SET balance = balance + 100.00
WHERE name = (SELECT branch_name
               FROM accounts
               WHERE name = 'Bob');
```

# La solution

## Je valide une transaction avec COMMIT

```
BEGIN;  
UPDATE accounts SET balance = balance - 100.00  
    WHERE name = 'Alice';  
-- etc etc  
COMMIT;
```

## J'annule une transaction avec COMMIT

```
BEGIN;  
UPDATE accounts SET balance = balance - 100.00  
    WHERE name = 'Alice';  
-- etc etc  
ROLLBACK;
```

# La solution

Le SGBDR garantis des transactions ACID:

- **A**tomacité
- **C**ohérence
- **I**solation
- **D**urabilité