

## JPA – Transaction

Guillaume Dufrêne – Lionel Seinturier

Université de Lille – Sciences et Technologies

## Définition

Groupe de requêtes devant être exécutées de façon indivisible

## Objectifs

- Concurrence des traitements et Consistance des données

## Opérations

- début de transaction (*begin*)
- engagement de la transaction (*commit*)
- annulation de la transaction (*rollback*)

## Exemple (1/2)

Transfert d'argent entre deux comptes bancaires

- contrainte : le solde d'un compte ne peut être négatif

```
@Entity
public class Compte {
    private String nom;
    private double solde;

    // constructeur
    // méthodes getNom, setNom, getSolde

    // ContrainteSoldePositifOuNull
    public void setSolde( double montant ) { this.solde = montant; }
}
```

## Exemple (2/2)

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("myapp");
EntityManager em = emf.createEntityManager();
EntityTransaction tx = em.getTransaction();

Compte bob = em.find( Compte.class, "Bob" );
Compte anne = em.find( Compte.class, "Anne" );

tx.begin();
try {
    bob.setSolde( bob.getSolde() - 100 );
    anne.setSolde( anne.getSolde() + 100);
    tx.commit();
}
catch( Exception e ) {
    tx.rollback();
}
```