

---

# Rapport de Projet

Rédigé par Raphaël Trancoso & Cédric Laguerre

---

Université Paris-Diderot - 29 mai 2017

## Projet Bases de Données Avancées

Il n'est pas permis de copier, distribuer et/ou modifier ce document.

---

---

## TABLE DES MATIÈRES

Introduction	3
Contacts	3
Le Modèle Conceptuel de Données (MCD)	4
Le modèle relationnel	4
La liste des fonctions utilisées	5
Conclusion	7

---

# Introduction

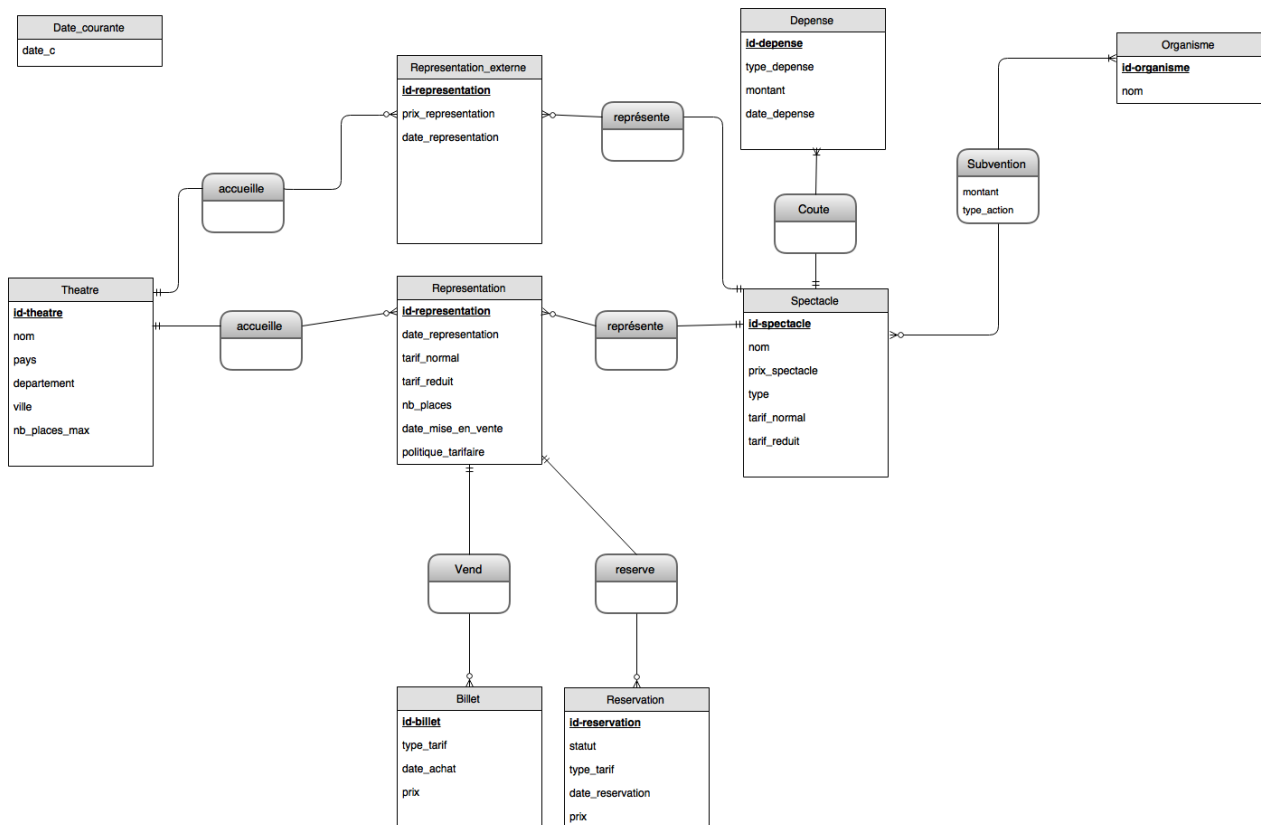
Dans le cadre du module de Bases de Données Avancées de la première année de Master à l'Université Paris-Diderot, nous avons informatisé la gestion d'une compagnie de théâtre..

## Contacts

Pour toutes informations relatives à ce projet, vous pouvez nous contacter aux adresses mails suivantes :

- Raphaël Trancoso : [raphael.trancoso@laposte.net](mailto:raphael.trancoso@laposte.net)
- Cédric Laguerre : [ced.laguerre@gmail.com](mailto:ced.laguerre@gmail.com)

# Le Modèle Conceptuel de Données (MCD)



## Le modèle relationnel

```
Theatre(id_theatre, nom, pays, département, ville, nb_places_max)
Organisme(id organisme, nom)
Spectacle(id spectacle, nom, type, prix_spectacle, tarif_normal,
            tarif_reduit)
Representation(id representation, nb_places, #id_theatre,
                #id_spectacle, date_representation, date_mise_en_vente,
                politique_tarifaire)
Representation_externe(id representation, prix_representation,
                        #id_theatre, #id_spectacle, date_representation)
Depense(id depense, type_depense, montant, date_depense,
          #id_spectacle)
Billet(id billet, prix, type_tarif, date_achat,
         #id_representation)
Reservation(id reservation, statut, type_tarif, date_reservation,
              prix, #id_representation)
Subvention(#id organisme, #id spectacle, montant, type_action)
Date(date_courante)
```

---

## La liste des fonctions utilisées

### **Trigger insert\_spectacle (BEFORE INSERT)**

- `verif_tarifSpectacle()` : vérifie si la valeur du tarif normal est supérieure à celle du tarif réduit.

### **Trigger insert\_depense (BEFORE INSERT)**

- `verif_insertDepense()` : vérifie si le spectacle entré en paramètre existe grâce à la fonction `verif_spectacleExist(id_spectacle int)`.

### **Trigger insert\_subvention (BEFORE INSERT)**

- `verif_insertSubvention()` : vérifie si l'organisme entré en paramètre existe à l'aide de la fonction `verif_organismeExist(id_organisme int)` et vérifie si le spectacle entré en paramètre existe également à l'aide de la fonction `verif_spectacleExist(id_spectacle int)`.

### **Trigger insert\_representation (BEFORE INSERT)**

- `verif_insertRepresentation()` : vérifie si les 5 fonctions suivantes retournent « vrai » pour pouvoir ajouter une représentation dans la table :
  - `verif_theatreExist(id_theatre int)` : vérifie si le théâtre entré en paramètre existe.
  - `verif_spectacleExist(id_spectacle int)` : vérifie si le spectacle entré en paramètre existe.
  - `verif_capacitePlace(nb_places_max_th int, nb_places_spec int)` : vérifie si le nombre de places alloué au spectacle ne dépasse pas la capacité maximale du théâtre.
  - `verif_dateRep(date_rep date, date_courante date)` : vérifie si la date de représentation n'est pas inférieure à la date courante.
  - `verif_venteBillet(date_rep date, date_vente date)` : vérifie si la date de la mise en vente des billets est inférieure à

---

la date de la représentation et supérieure à la date courante.

### **Trigger insert\_repEx (BEFORE INSERT)**

- `verif_representationExt()` : vérifie si le spectacle entré en paramètre appartient à notre compagnie à l'aide de la fonction `verif_notreSpec(id_spectacle int)`.

### **Trigger insert\_billet (BEFORE INSERT)**

- `verif_insertBillet()` : vérifie si l'on peut ajouter un billet grâce aux fonctions suivantes :
  - `verif_BilletDispo(nb_places_tot int, nb_places_rep int)` : vérifie s'il reste encore un billet disponible à l'achat.
  - `verif_dateAchatBillet(date_achat date, date_rep date)` : vérifie si la date d'achat du billet est inférieure à la date de représentation.

### **Trigger insert\_reservation (BEFORE INSERT)**

- `verif_insertReservation()` : vérifie, comme pour l'achat de billet (voir ci-dessus : Trigger `insert_billet`), si l'on peut ajouter une réservation grâce aux fonctions `verif_BilletDispo(nb_places_tot int, nb_places_rep int)` et `verif_dateAchatBillet(date_achat date, date_rep date)`.

### **Trigger resa\_expire (AFTER UPDATE)**

- `resa_expire()` : permet d'afficher une première alerte lorsque qu'il reste moins de 5 jours pour payer une réservation de billet, une deuxième alerte lorsque la personne a jusqu'à demain pour payer sa réservation, sinon une notification est envoyée pour signaler que la réservation du billet est annulée et supprime donc la réservation.

### **Autres fonctions**

- Fonctions d'ajout basique dans les tables

- 
- Fonctions permettant d'afficher l'historique des données de la compagnie
  - Fonctions permettant de calculer les recettes mais également les dépenses d'une compagnie, ou même d'un spectacle ou représentation.

## Conclusion

Ce projet nous aura permis d'approfondir nos connaissances en bases de données.