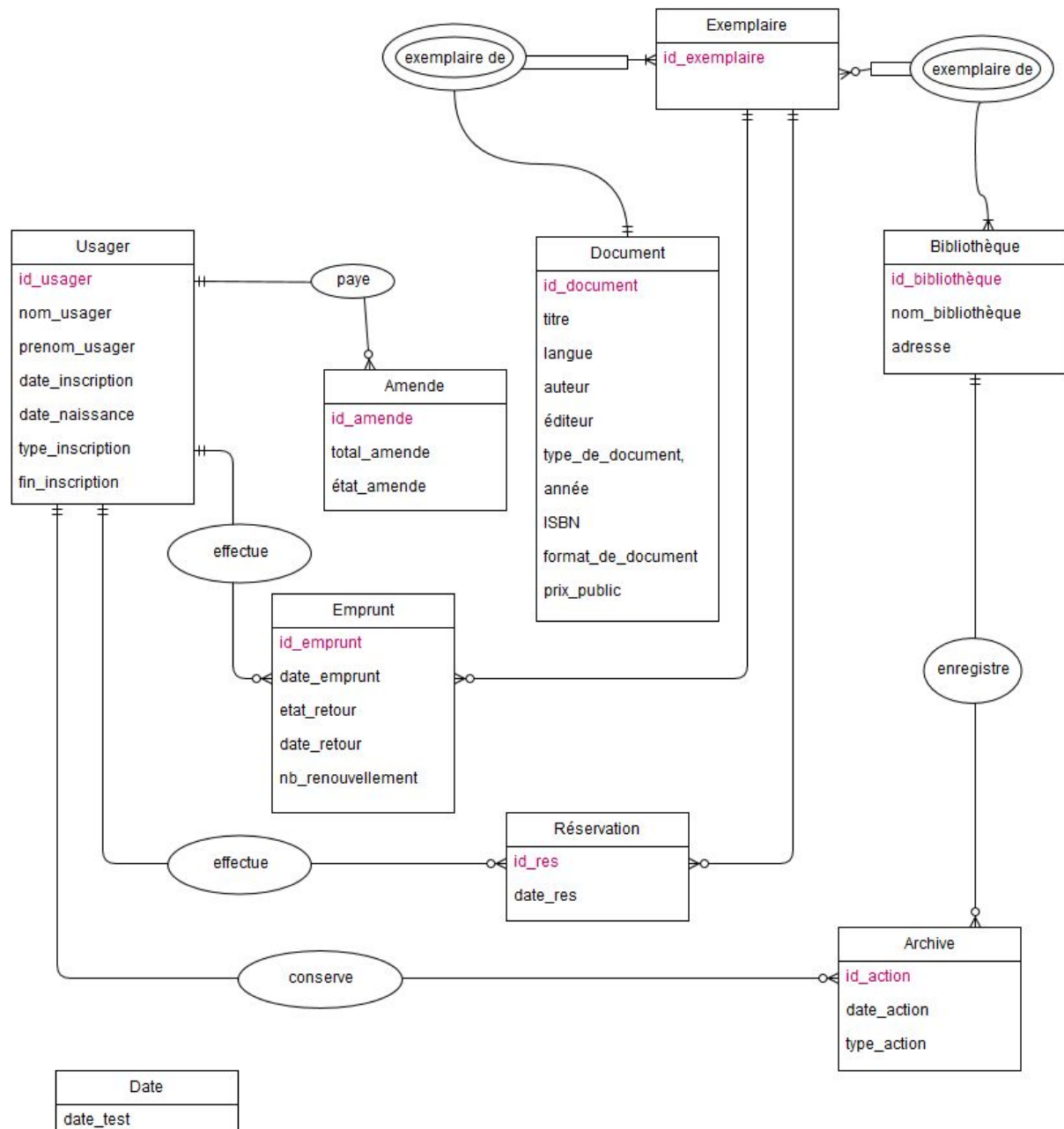


## Projet Base de Données Avancées

### Modèle entité-association:



### **Modèle relationnel :**

**Usager**(\*id\_usager, nom\_usager, prenom\_usager, date\_naissance, date\_inscription, type\_inscription, fin\_inscription)

**Amende**(\*id\_amende, id\_usager#, état\_amende, total\_amende )

**Bibliothèque**(\*id\_bibliothèque, nom\_bibliothèque, adresse)

**Document**(\*id\_document, titre, langue, auteur, éditeur, type\_de\_document, année, ISBN, format\_de\_document, prix\_public)

**Exemplaire**(\*id\_exemplaire, id\_document#, id\_bibliothèque#)

**Réservation**(\*id\_res, date\_res, (id\_exemplaire, id\_bibliothèque, id\_document)#, id\_usager#)

**Archive**(\*id\_action, date\_action, type\_action, id\_usager#, id\_bibliothèque#)

**Emprunt**(\*id\_emprunt, date\_emprunt, état\_retour, date\_retour, nb\_renov, (id\_exemplaire, id\_document, id\_bibliothèque)#, id\_usager#)

### **Contraintes externes et règles de gestion :**

#### **Par CHECK :**

- La date de retour doit être postérieure à la date d'emprunt
- Les dates ne peuvent pas être nulles.
- Le prix public du document doit être supérieur à 0.
- L'année du document ne peut pas être postérieure à l'année en cours.
- type\_inscription (usager) est ("GRATUITE", "CD", "CD/DVD")
- état\_amende (amende) est ("IMPAYÉE", "PAYÉE")
- type\_action (archive) est ("INSCRIPTION", "DESINSCRIPTION", "EMPRUNT", "RETOUR", "PERTE")
- type\_de\_document(document) est ("JEUNESSE", "ADULTE", "TOUT PUBLIC")
- format\_de\_document (document) est ("LIVRE", "DVD", "CD", "BANDE DESSINÉE", "METHODE MUSICALE", "PARTITION")

#### **Par TRIGGER :**

##### ***Usager***

- Une inscription expirée (date fin\_inscription > CURRENT\_DATE) empêche d'emprunter ou réserver un livre tant qu'elle n'a pas été renouvelée.

##### ***Emprunt***

- Un livre ne peut être emprunté que dans la mesure de son stock disponible selon la bibliothèque
- Un usager âgé de moins de 18 ans n'a accès qu'aux documents de type "Jeunesse".
- Un usager gratuit (niveau 1) peut tout emprunter sauf les CD/DVD.
- Un usager CD (niveau 2) peut tout emprunter sauf les DVD.
- Un usager CD/DVD (niveau 3) peut tout emprunter.
- Un usager peut emprunter 20 documents simultanément (exceptés DVD) par bibliothèque, 40 en tout.
- Un usager peut emprunter 5 DVD simultanément.
- Un prêt est de 3 semaines.
- Un prêt est renouvelable maximum 2 fois : date allongée de 3 semaines à partir de la date de renouvellement.

- Un prêt n'est pas renouvelable s'il a été réservé ou si l'utilisateur a un retard (toutes les amendes sont réglées).
- On ne peut pas emprunter un livre déjà emprunté.
- On ne peut pas rendre un livre qui n'a pas été emprunté.

#### *Réservation*

- Un usager ne peut avoir que 5 réservations simultanées.
- Une réservation n'est possible que si tous les exemplaires du document dans la bibliothèque sont prêtés.

#### *Amende*

- En cas de retard ( $\text{total\_amende} > 0$ ), l'usager ne peut plus emprunter ni réserver.
- L'amende augmente de 0.15 par document en retard, par jour.
- Si le document est perdu/dégradé, l'amende est fixée du prix public du document.

#### **ALERTE AUTOMATIQUE :**

- L'amende d'un usager atteint 15€ sans être payée.
- Un livre réservé a été rendu.
- Un livre doit être rendu dans deux jours.
- Une inscription a expirée.

#### **Liste des hypothèses simplificatrices de l'implémentation**

On ne prend pas en compte la différence été et reste de l'année pour les prêts. Les nouveautés ne sont pas gérées et sont considérées comme des documents classiques.

Les limites ont été divisées par deux (sauf celle des DVD), pour une simple raison pratique : c'était beaucoup plus facile de tester avec 10 et 20 qu'en gardant 20 et 40.

#### **Liste des fonctions et triggers**

```
f_getUsager(id_usager)
f_getDocument(id_document)
f_getCurrentDate()
```

Les trois fonctions ont été créées pour éviter la redondance dans certaines fonctions où on cherche un document ou un usager en particulier ; la troisième donne la date actuelle.

```
f_fin_inscription()
t_insert_fin_inscription
t_update_fin_inscription
```

Les triggers s'activent lors de l'insertion ou la modification d'une inscription (généralement un renouvellement) ; la fonction modifie la date de fin de validité de l'inscription en la plaçant à un an de la date actuelle. Une notice signale la nouvelle date de fin de validité.

```
f_alerte_retours(new_date)
```

```

f_alerte_inscription(new_date)
f_amende(old_date, new_date)
f_update_date()
t_update_date

```

Le trigger s'active lorsque la date est modifiée. La fonction `f_update_date()` vérifie successivement les trois autres fonctions ; des notices signalent si une amende est en cours et son montant, si une inscription va expirer ou si un document doit être rendu sous peu.

```

f_verification_age(id_usager, id_document)
f_verification_utilisateur(id_usager)
f_verification_expiration(id_usager)
f_verification_inscription(id_usage, format_document)
f_verification_amende(id_usager)
f_verification_emprunts(id_usager, id_bibliothèque, format_document)
f_verification_existence(id_exemplaire, id_document, id_bibliothèque)
f_verification_disponibilite_emprunt(id_exemplaire, id_document, id_bibliothèque)
f_verification_disponibilite_reservation(id_exemplaire, id_document,
id_bibliothèque)
f_emprunt()
t_emprunt

```

Le trigger s'enclenche lorsqu'un utilisateur veut emprunter un livre ; les fonctions permettent de vérifier toutes les informations relatives à l'utilisateur (s'il existe, son âge, s'il a le droit d'emprunter le livre, si son inscription est toujours valide, s'il n'a pas une amende supérieure à 15€, s'il n'a pas atteint le maximum d'emprunt) et au document qu'il veut emprunter (s'il existe, si le document est disponible à l'emprunt, s'il est disponible à la réservation). Des notices permettent de signaler l'étape à laquelle le test échoue.

```

f_emprunt_retard(id_emprunt)
f_emprunt_renouvellement()
t_emprunt_renouvellement

```

Le trigger s'enclenche quand on essaye de renouveler l'emprunt d'un document ; les fonctions vérifient que l'utilisateur n'a pas de retards en cours et s'il n'a pas atteint le maximum de renouvellement pour emprunter le document.

```

f_verification_etat_retour()
t_verification_etat_retour

```

Le trigger s'enclenche quand l'utilisateur rend son livre ; la fonction vérifie l'état donné par l'utilisateur et augmente l'amende en conséquence.

```
f_verification_nb_reservation(id_usager)
f_verification_dispo_exemplaire(id_exemplaire, id_document, id_bibliothèque)
f_reservation()
t_reservation
```

Le trigger s'enclenche quand l'utilisateur veut faire une réservation ; les fonctions vérifient si le livre est disponible (s'il l'est, une notice le signale et la réservation est stoppée), si l'utilisateur n'a pas atteint le maximum de réservation possible.

```
f_archive_inscription()
t_archive_inscription

f_archive_emprunt()
t_archive_emprunt
```

Les archives nous permettent de garder en mémoire dès qu'il se passe quelque chose sur une table : si un document est emprunté, s'il est rendu, si un utilisateur s'inscrit. Elles nous permettent de garder une trace de ce qu'il s'est passé, même dans le cas où on efface quelque chose dans une table ; par exemple, on a la possibilité de nettoyer la table des emprunts (uniquement ceux datant de plus de deux mois et ayant été rendus). La table archive n'est pas nettoyée et garde donc en mémoire même les emprunts les plus vieux.

```
f_alerte_resa()
t_alerte_resa
```

Le trigger s'enclenche quand un document a été rendu : une notice signale à un usager que le document qu'il a été réservé a été rendu.