

# Location Vidéo

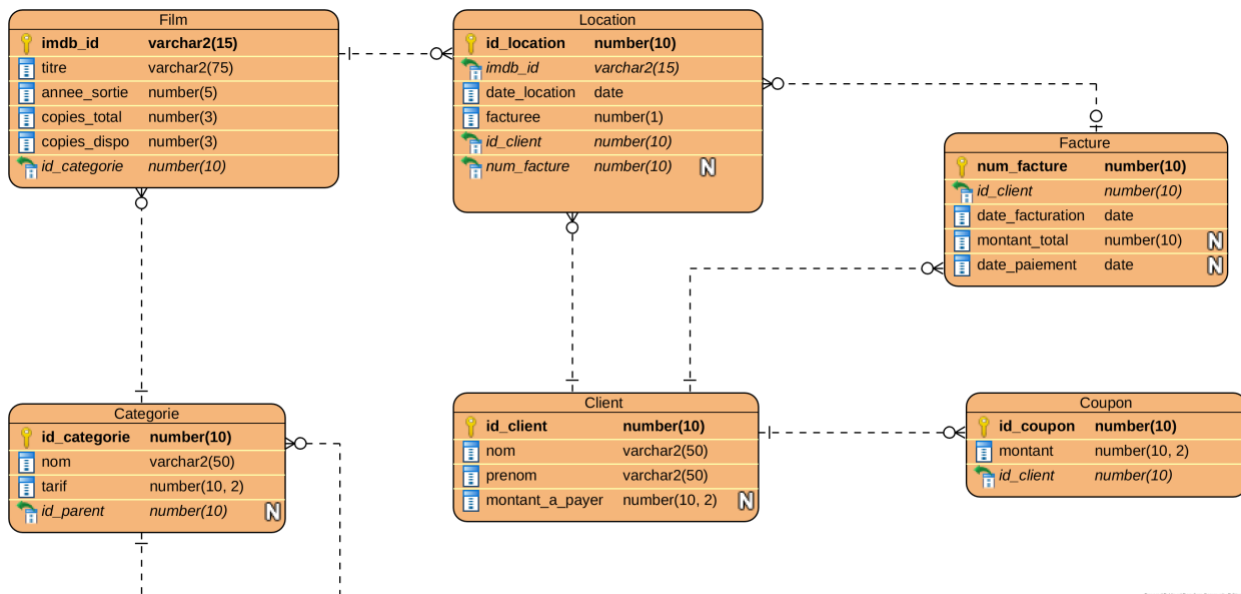
## 1. Présentation générale du laboratoire

Ce quatrième laboratoire, d'une durée de deux semaines, vise à mettre en pratique vos apprentissages des interfaces entre SQL et Java.

Vous utiliserez la base de données de locations vidéo du laboratoire 3 et dont le LDD vous a été fourni. Vous devez étendre cette base de données en programmant des procédures et des fonctions qui automatisent des traitements. La section 2 rappelle le modèle relationnel de la base de données. La section 3 énumère l'ensemble des fonctions que vous devez implémenter à l'aide du langage de programmation Java.

Ce laboratoire doit être réalisé en groupes, en préservant les mêmes groupes que le premier laboratoire (à moins qu'un changement ait été autorisé par votre enseignant-e).

## 2. Modèle relationnel :



### 3. Fonctions à programmer en Java:

Un gabarit de projet vous est fourni sur Moodle. Deux versions vous sont fournies : une version pour NetBeans 8.2 (disponible dans les laboratoires) et une version pour Eclipse. Vous pouvez importer ces gabarits dans NetBeans ou Eclipse. Vous devez, ensuite, compléter les méthodes directement dans le projet Java. En plus de cet énoncé, les commentaires des fonctions vous donnent des indications sur le travail à réaliser.

→ La méthode *charger()*, de la classe *Client*, vous est fournie (pour vous inspirer).

**Question 0 :** Commencez par configurer vos codes d'accès à la BD dans la classe *LocationVideo*.

**Question 1 :**

Dans la classe *LocationVideo*, implémentez la méthode *obtenirConnection* qui crée une connexion à la base de données oracle (retourne un objet de type *java.sql.Connection*). Les informations de connexion (nom d'utilisateur, etc.) sont celles qui sont définies comme constantes de la classe *LocationVideo*.

**Question 2 :**

Implémenter la méthode statique *listeClients* qui retourne la liste des clients se trouvant en base de données. La liste de clients retournée est une *ArrayList<Client>*.

**Question 3 :**

Implémenter la méthode statique *ajouterClient* qui crée un nouveau client à partir des informations reçues en paramètre. Votre fonction doit valider la transaction si tout se produit tel que prévu, ou l'annuler sinon.

**Question 4 :**

Écrire la méthode *sauvegarder*, de la classe *Client*, qui met à jour un client en base de données.

Vous devez vous assurer que l'attribut *idClient* contient une valeur non-nulle. Votre fonction doit valider la transaction si tout se produit tel que prévu, ou l'annuler sinon.

**Question 5 :**

Implémenter la méthode statique *supprimerClient* qui crée supprime le client dont l'identifiant est fourni en paramètre. Votre fonction doit valider la transaction si tout se produit tel que prévu, ou l'annuler sinon.

**Question 6 :**

Implémenter la méthode statique *rechercheParNom* qui retourne une liste de Clients dont le nom est semblable au nom passé en paramètre.

### 4. Livrable et remise finale :

Vous devez remettre un fichier à l'extension **.zip** contenant les classes **Client.java**, **LocationVideo.java** et **InterfaceUsager.java**.

La remise doit être effectuée sur Moodle à remettre durant la semaine du 1<sup>er</sup> avril 2019 (voir date précise selon votre groupe sur Moodle). En respect des règles du Service des Enseignements Généraux, toute remise tardive se verra attribuer la note de 0 (à moins d'entente justifiée et préalable avec l'enseignant-e).

**Bon travail 😊**