

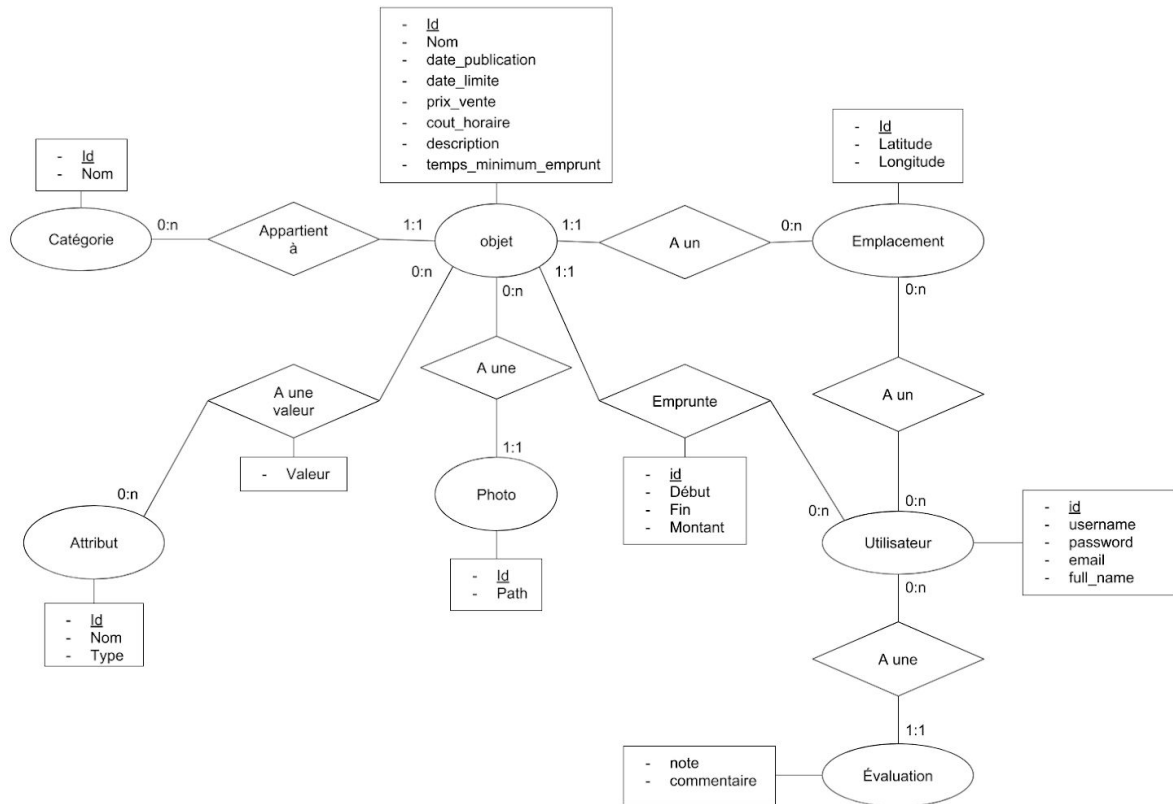
**Rapport Projet Final**  
IFT2945 - Bases de données

Présenté à:  
Michel Boyer  
Maude Sabourin

Fait par:  
Simon Bréard  
Marc-André Brochu  
Olga Fadeitcheva  
Mathieu Matos

Université de Montréal  
6 mai 2018

## Représentation E-A de la base relationnelle



## Schéma relationnel de la base

Objet(id, nom, date\_publication, date\_limite, prix\_vente, cout\_horaire, description, temps\_minimum\_emprunt, photo, #emplacement\_id, #categorie\_id, #proprietaire\_id)

Categorie(id, nom)

Utilisateur(id, username, password, email, full\_name, #emplacement)

Photo(id, path, #objet)

Attribut(id, #categorie\_id, nom, type)

ValeurAttribut(#attribut, #objet, valeur)

Emprunt(id, #objet, #emprunteur, #annonceur, debut, fin, montant)

Evaluation(#evaluator, #evaluate, note, commentaire)

Emplacement(id, longitude, latitude)

## Explication du code DDL

Pour implémenter la logique des objets à vendre/louer il a fallu créer 9 tables:

- **“objet”** sert à stocker tous les objets à vendre et louer présents dans le système. La table contient des attributs globaux propres à chaque objet peu importe la catégorie (par exemple la date de publication, le prix de vente, le coût horaire, etc.). L'id est la clé primaire de cette table car elle est suffisante pour définir l'objet en question. De plus, il y a trois attributs qui réfèrent aux clés primaires des autres tables (emplacement, catégorie, annonceur->utilisateur) car ils sont nécessaires dans la définition de l'objet.
- **“attribut”** sert à définir les attributs propres à chaque catégorie. Ainsi il y a seulement 4 colonnes, une pour le id, une pour le nom, une pour le type, et la dernière référant l'id de la table des catégories, pour indiquer à quelle catégorie d'objets cet attribut appartient. La clé primaire de cette table est simplement son id.
- **“catégorie”** est une table simple ayant un id et un nom. Sa clé primaire est son id et son nom ne peut pas être nulle car justement la table est basée uniquement sur les différents noms des catégories.
- **“valeurattribut”** sert à définir la valeur spécifique d'un objet en fonction de sa catégorie. Ce dont pour quoi une colonne va référencer à l'id de l'attribut et une autre à l'id de l'objet, et le tuple de ces deux valeurs va former la clé primaire. La 3e colonne est pour la valeur de l'attribut, et comme le type de chaque attribut peut être différent, on a décidé de mettre le type de cet attribut à *text* (un typage approprié à n'importe quelle donnée).
- **“emplacement”** est une petite table simplement indiquant la longitude et la latitude d'une position précise. Ces données vont être utiles pour repérer la position d'un utilisateur ou d'un objet. La clé primaire va être l'id, référant ainsi à la position précise.
- **“utilisateur”** sert à définir tous les utilisateurs du système. L'id va être la clé primaire. De plus, l'attribut *emplacement* va référer à l'id d'un élément de la table “emplacement”
- **“photo”** est une petite table avec un id, le chemin de la photo et la référence à l'objet représenté dans la photo à travers le *object\_id*. La clé primaire est le id. Nous avons décidé de faire une table pour les photos plutôt que de le mettre comme un attribut de la table “objet” pour permettre à l'utilisateur d'insérer plusieurs photos pour un objet.
- **“emprunt”** sert à stocker l'information des emprunts. L'id va être la clé primaire. Il y aura également une colonne pour la date du début de l'emprunt et pour la date de la fin de l'emprunt. La colonne d'objet réfère à l'id de l'objet, celle de l'emprunteur à l'id d'un utilisateur et celle de l'annonceur à l'id d'un autre utilisateur.
- **“evaluation”** est une table qui stocke les évaluations sur les utilisateurs qui ont empruntés des objets (un peu comme Uber). Ainsi, il y a l'attribut *évaluateur* ainsi que l'attribut *évalué* référant à l'id d'un utilisateur. De plus il y a l'attribut *score*, l'attribut *text*. La clé primaire de cette table est le tuple (*évaluateur\_id*, *évalué\_id*) car ce sont les deux informations qui permettent de définir l'évaluation.

## Vues

1) Cette vue permet un accès plus facile aux données portant sur les divers objets de la base. Elle réunit en une table virtuelle les informations utiles et nécessaires à l'affichage du listing d'un objet.

```
CREATE VIEW objet_view AS SELECT
    objet.id,
    objet.nom AS nom_objet,
    objet.proprietaire_id AS vendeur_id,
    nom_vendeur,
    email_vendeur,
    objet.description AS desc,
    objet.cout_horaire AS cout,
    objet.prix_vente AS prix,
    objet.date_publication AS pub_date,
    objet.date_limite AS lim_date,
    objet.temps_min_emprunt AS temps_min,
    chemin_photo,
    lat, lon
FROM (
    objet
    NATURAL JOIN (
        SELECT utilisateur.id AS proprietaire_id,
            utilisateur.username AS nom_vendeur,
            utilisateur.email AS email_vendeur
        FROM utilisateur
    ) AS vendeur
    NATURAL JOIN (
        SELECT emplacement.id AS emplacement_id,
            lat, lon
        FROM emplacement
    ) AS place
    LEFT JOIN (
        SELECT photo.objet_id,
            chemin AS chemin_photo
        FROM photo
    ) AS p ON objet.id = p.objet_id
);
```

2) Cette vue permet un accès plus direct aux attributs de chaque objets.

```
CREATE VIEW objet_attribut_view AS SELECT
    obj_id AS objet_id,
    attr_id AS attribut_id,
    nom_attribut,
    valeur AS val_attribut
FROM
    (SELECT objet.id AS obj_id,
        attribut.id AS attr_id,
        attribut.nom AS nom_attribut
    FROM objet
    RIGHT JOIN attribut ON objet.categorie_id = attribut.categorie_id)
    AS a
    LEFT JOIN valeurattribut ON
        obj_id = valeurattribut.objet_id AND
        valeurattribut.attribut_id = attr_id;
```

## Requêtes SQL

Vérifier la connexion utilisateur:

```
SELECT id FROM utilisateur WHERE  
    username = <username> AND password = <password>;
```

Emprunter un objet:

```
INSERT INTO emprunt (  
    debut, fin, montant, objet_id, emprunteur_id, annonceur_id  
) VALUES (  
    <date debut>,  
    <date fin>,  
    <montant en dollars>,  
    <objet>,  
    <emprunteur>,  
    <annonceur>  
);
```

Liste des emprunts de quelqu'un (avec toutes les informations des objets):

```
SELECT * FROM objet_view NATURAL JOIN (  
    SELECT objet_id AS id FROM emprunt WHERE emprunteur_id = <user_id>  
) AS e; > Obtenir les informations sauf la valeur des attributs  
  
SELECT * FROM objet_attribut_view NATURAL JOIN (  
    SELECT objet_id FROM emprunt WHERE emprunteur_id = <user_id>  
) AS e; > Obtenir les valeurs des attributs pour tout les objets empruntés
```

Liste des annonces de quelqu'un (avec toutes les informations des objets):

```
SELECT * FROM objet_view WHERE vendeur_id = <user_id>;
```

Évaluer quelqu'un:

```
INSERT INTO evaluation (evaluateur_id, evalue_id, score, commentaire)  
VALUES (  
    <évaluateur>,  
    <évalué>,  
    <score>,  
    <commentaire>  
);
```

Liste des objets disponibles en ce moment ou dans le futur (avec toutes leurs informations, dont leurs attributs et leurs valeurs):

```
SELECT * FROM (
    SELECT * FROM objet_view WHERE
        pub_date <= now() AND
        lim_date >= now()) AS l
NATURAL JOIN (
    SELECT objet_id AS id, nom_attribut, val_attribut
    FROM objet_attribut_view
) AS r;
```

Chercher les informations d'un objet:

```
SELECT * FROM (
    SELECT * FROM objet_view WHERE id = <objet_id>) AS l
NATURAL JOIN (
    SELECT objet_id AS id, nom_attribut, val_attribut
    FROM objet_attribut_view
) AS r; > Pour obtenir beaucoup d'information (toutes celles disponibles)

SELECT * FROM objet_attribut_view WHERE objet_id = <objet_id>;
> Pour obtenir seulement les valeurs des attributs d'un objet.
```

Créer un objet:

```
INSERT INTO objet (
    nom, date_publication,
    date_limite, prix_vente,
    cout_horaire, description,
    temps_min_emprunt, emplacement_id,
    categorie_id, proprietaire_id
) VALUES (
    <nom>,
    now(), -- la date de publication peut être changée ultérieurement
    <date limite>,
    <prix de vente si applicable>,
    <cout horaire si applicable>,
    <description de l'objet>,
    <temps minimal d'un emprunt>,
    <emplacement de l'objet>,
    <categorie de l'objet>,
    <propriétaire de l'objet>
);
```



Modifier un objet:

```
UPDATE objet SET <liste d'attributs> = <nouvelles valeurs>
WHERE id = <objet_id>;
```

Obtenir les informations d'un utilisateur:

```
SELECT username, email, full_name, telephone, lat, lon
FROM utilisateur JOIN emplacement
ON (utilisateur.emplacement_id = emplacement.id)
WHERE utilisateur.id = <user_id>;
```

Ajouter un utilisateur:

```
INSERT INTO utilisateur (
    username, password, email, full_name, telephone, emplacement_id
) VALUES (
    <username>,
    <password>,
    <email>,
    <nom complet>,
    <telephone>,
    <emplacement>
);
```

Modifier un utilisateur:

```
UPDATE utilisateur SET <liste d'attributs> = <nouvelles valeurs>
WHERE id = <user_id>;
```

Liste des catégories:

```
SELECT * FROM categorie;
```

## Mini-guide utilisateur de l'application

Pour accéder à l'application, le serveur du DIRO héberge la page à :

<http://www-ens.iro.umontreal.ca/~breardsi/ift2935-final>

Comme utilisateur d'essai, entrer ces informations suivantes :

le nom d'utilisateur : test

Le mot de passe : test

Dans la page de l'index de Jikiki, on demande d'entrer les informations d'authentification. On peut aussi se créer un nouveau compte et insérer les informations demandées.

Suite à l'authentification, le menu nous offre 5 choix. Le premier étant de faire une recherche parmi toutes les annonces disponibles et de filtrer parmi les attributs différents de ces annonces. Aussi, il y a possibilité de faire une recherche. Pour prendre contact avec le propriétaire de l'objet, appuyer sur 'Contacter'.

Il est également possible d'afficher les annonces de l'utilisateur courant. Comme la page de recherche, différentes annonces seront affichées avec leurs caractéristiques respectives le tout en étant les annonces de l'utilisateur présent.

Pour créer une annonce, choisir s'il s'agit d'une vente ou d'une location ainsi que la catégorie de l'objet. Ensuite, une redirection vers les spécifications de l'objet à remplir. Il y a aussi possibilité d'ajouter une image.

Pour voir les informations du profil ainsi que les modifier, l'option 'Mon Profil' affiche les données de l'utilisateur comme dans la création de celui-ci. C'est possible de les changer et sauvegarder ces changements.

Finalement, on peut se déconnecter pour terminer la session courante et retourner à l'index.