

# P4b

## Base de données non relationnelles

## Table des matières

<b>CHOIX D'IMPLEMENTATION .....</b>	<b>3</b>
<b>ADAPTATION DU SCHEMA ACTUEL VERS LE RELATIONNEL-OBJET .....</b>	<b>3</b>
ADAPTATION .....	3
<i>Badge et BadgeObtenus :</i> .....	3
<i>SignaleCommentaire, LikeCommentaire:</i> .....	3
<i>SignaleAnnonce :</i> .....	3
<i>VoteAnnonce :</i> .....	3
SCHEMA SIMPLIFIE DES IMBRICATIONS : .....	4
<b>NOUVEAU MLD (EN RELATIONNEL-OBJET) .....</b>	<b>5</b>
<b>LDD - CREATION DES TABLES ET TYPES .....</b>	<b>6</b>
CREATION DE BADGE : .....	6
CREATION DE UTILISATEUR : .....	6
CREATION DE COMMENTAIRE : .....	7
CREATION DE ANNONCE : .....	7
<b>EXEMPLE DE REQUETES .....</b>	<b>8</b>
INSERTION DES BADGES : .....	8
<i>Première insertion :</i> .....	8
<i>Deuxième insertion :</i> .....	8
<i>Troisième insertion :</i> .....	8
<i>Quatrième insertion :</i> .....	8
INSERTION DES UTILISATEURS : .....	9
<i>Première insertion :</i> .....	9
<i>Deuxième insertion :</i> .....	9
<i>Troisième insertion :</i> .....	10
<i>Quatrième insertion :</i> .....	10
2) INSERTION DES ANNONCES .....	11
<i>Première annonce :</i> .....	11
<i>Deuxième annonce :</i> .....	11
3) INSERTION DES COMMENTAIRES : .....	11
<i>Premier commentaire :</i> .....	11
<i>Deuxième commentaire :</i> .....	12
<i>Troisième commentaire :</i> .....	12
4) INSERTION D'UN COMMENTAIRE DANS UNE ANNONCE : .....	12
<i>Premier commentaire inséré dans la première annonce:</i> .....	12
<i>Deuxième commentaire inséré dans la deuxième annonce:</i> .....	12
<i>Troisième commentaire inséré dans une annonce :</i> .....	13
5) EXEMPLE DE SELECTION .....	13
<i>Sélection d'un commentaire spécifique d'une annonce</i> .....	13
<i>Sélection des badges d'un utilisateur</i> .....	13
<i>Sélection de tous les commentaires de l'annonce 1</i> .....	13
<i>Sélection du nom et prénom des utilisateurs ayant liker un commentaire de l'annonce 1</i> .....	13
<i>Sélection des noms prénoms et intitulés des badges des utilisateurs ayant liker une annonce, sans doublon et trié par nom</i> .....	13
<i>Sélection des nom et prénoms des utilisateurs qui ont le badge "likeur" :</i> .....	13

## Choix d'implémentation

Pour l'implémentation de notre BDD, nous choisissons d'utiliser Oracle, en retravaillant notre modèle logique pour l'adapter à un modèle relationnel-objet.

On va briser la 1NF qui permet l'identification des tuples par les données. En relationnel-objet nous utiliserons l'identification par adresse mémoire (pointeur) ou disque, par OID. On va donc passer de la 1NF à la Non First Normal Form (NF<sup>2</sup>).

Nous faisons ce choix car certaines de nos données sont complexes (utilisateur, annonce par exemple), et nous voulons un modèle logique plus proche du modèle conceptuel. Ce choix devrait aussi réduire les pertes de performances liées à la normalisation et aux jointures.

## Adaptation du schéma actuel vers le relationnel-objet

### Adaptation

Les données représentées sous forme d'objets sont : Utilisateurs - Annonces - Commentaires

En effet ce sont des données complexes (car de nombreux attributs). Nous voulons doter chacun de ses enregistrements d'un identifiant d'objet (OID) et l'utiliser comme système de référencement à la place des clés étrangères, pour gagner en efficacité.

Nous allons créer une table d'objets Utilisateurs et une table d'objet Annonces, ainsi qu'une table Commentaires.

Et Annonces contiendra un attribut ListeCommentaires (imbrication des commentaires dans Annonces).

### *Badge et BadgeObtenus :*

Les badges seront représentés comme une table imbriquée dans Utilisateurs. Cette table contiendra le contenu du Badge, mais aussi sa date d'obtention, ce qui nous permet de fusionner les tables Badge et BadgeObtenus du MLD précédent en un type BadgeT puis une liste imbriquée dans Utilisateurs, de type ListeBadgeDate.

### *SignaleCommentaire, LikeCommentaire:*

Pour chacune de ces deux relations on a un lien d'un commentaire vers un utilisateur. On va les imbriquer dans Commentaires. Du coup il ne reste qu'à référencer les Utilisateurs, on va donc simplement créer les types (ListeSignaleCommentaire, ListeLikeCommentaire) qui sont des collections de Utilisateurs. CommentaireT contiendra un attribut de type ListeSignaleCommentaire et un ListeLikeCommentaire, afin de créer une imbrication.

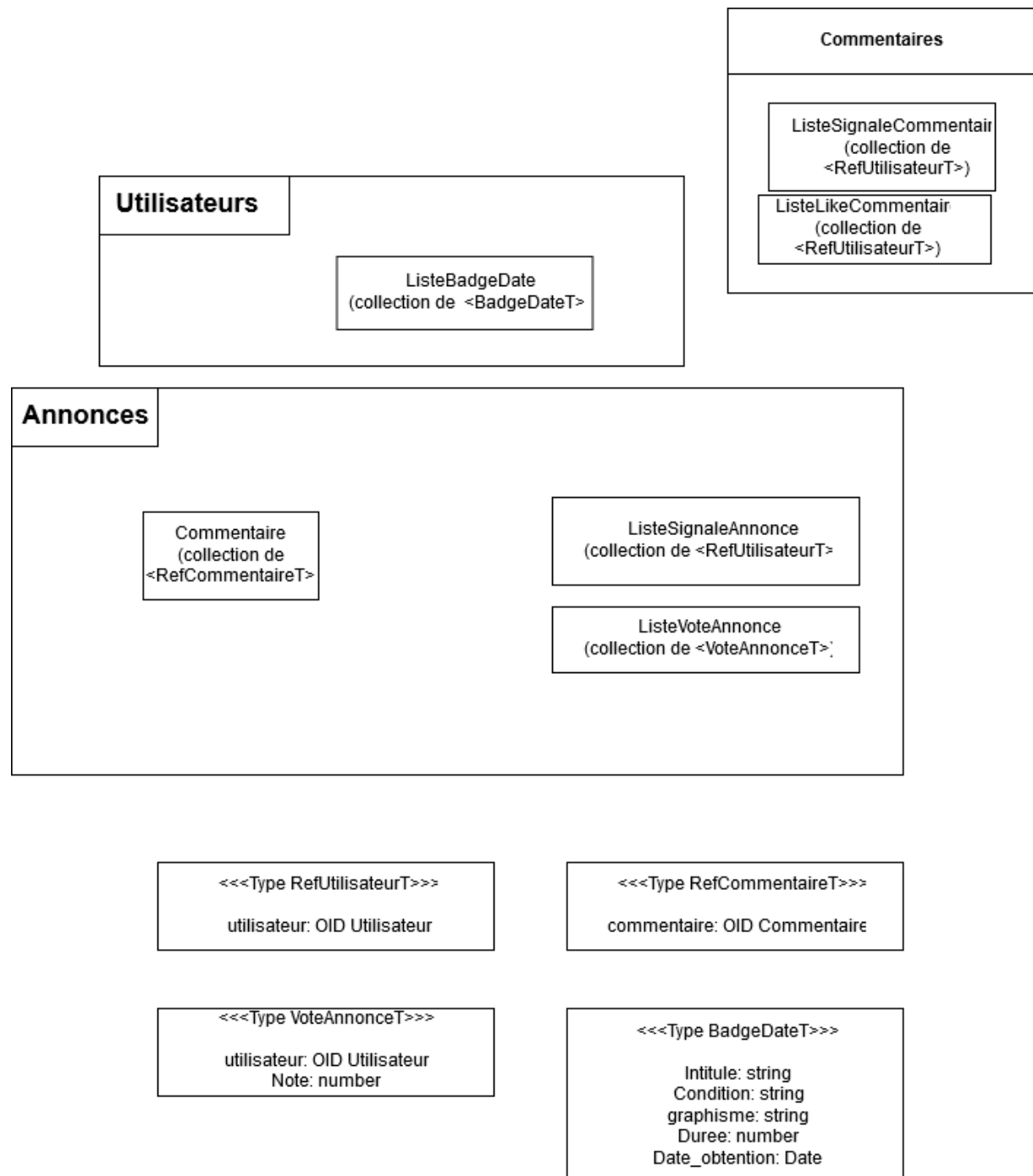
### *SignaleAnnonce :*

On va créer un type objet SignaleAnnonceT, puis un type liste de ces objets ListeSignaleAnnonceT, et enfin Annonces contiendra un attribut de type ListeSignaleAnnonceT pour créer une imbrication.

### *VoteAnnonce :*

On va créer un type objet VoteAnnonceT, puis un type liste de ces objets ListeVoteAnnonceT, et enfin Annonces contiendra un attribut de type ListeVoteAnnonceT pour créer une imbrication.

## Schéma simplifié des imbrications



## Nouveau MLD (en relationnel-objet)

Type BadgeT : <intitulé: string, condition: {'avoir posté 5 annonces', 'avoir aimé 10 commentaires', 'signaler 50 annonces', 'avoir voté pour 10 annonces', 'avoir publié une annonce'}, graphisme: string, durée: integer>

Badges de BadgeT(#intitulé) avec condition not null

Type BadgeObtenuT: <utilisateur=>o Utilisateur, date\_obtention DATE>

Type ListeBadgeObtenu: collection de <BadgeObtenuT>

Type UtilisateurT : <pseudo: string, statut: {'user','admin'}, nom: string, prenom: string, photo: string, badges: ListeBadgeObtenu>

Utilisateurs de UtilisateurT(#pseudo) avec pseudo NOT NULL, nom NOT NULL, prenom NOT NULL

Type RefUtilisateurT: <utilisateur =>o Utilisateurs>

Type ListeUtilisateur: collection de <RefUtilisateurT>

Type CommentaireT : <ID\_commentaire: integer, pseudo => o Utilisateurs, SignaleCommentaire: ListeUtilisateur, LikeCommentaire:ListeUtilisateur, texte: text>

Commentaires de CommentaireT(#ID\_commentaire) avec pseudo NOT NULL, texte NOT NULL

Type RefCommentaireT : <commentaire =>o Commentaire>

Type ListeCommentaire: collection de <RefCommentaireT>

Type VoteAnnonceT: <pseudo =>o Utilisateurs, valeur: integer>

Type ListeVoteAnnonce: collection de <VoteAnnonceT>

Type AnnonceT : <ID : integer, description: text, categorie : {'Informatique', 'Voiture', 'Loisir', 'Service'}, enseigne: varchar, image: varchar, etat: integer, date\_debut: date, date\_fin: date, lien: varchar, code: varchar, pseudo =>o Utilisateurs, pseudo\_moderateur =>o Utilisateurs, commentaires: listeCommentaire, signal\_annonce: ListeUtilisateur, vote\_annonce: listeVoteAnnonce>

Annonces de AnnonceT(#ID) avec categorie NOT NULL, enseigne NOT NULL, compteur\_vote NOT NULL DEFAULT(0), etat NOT NULL DEFAULT(0), pseudo NOT NULL, pseudo\_moderateur NOT NULL

## LDD - Création des tables et types

L'implémentation de la BDD est faite sur Oracle SQL Developer.

### Création de badge :

```
create or replace type badgeT as object(  
  intitule varchar2(100),  
  condition varchar2(200),  
  graphisme varchar2(200),  
  duree number);
```

```
create table Badges of badgeT(  
  check( condition='avoir posté 5 annonces' or condition= 'avoir aimé 10 commentaires' or condition=  
  'signaler 50 annonces' or condition= 'avoir voté pour 10 annonces' or condition= 'avoir publié une  
  annonce'),  
  primary key (intitule));
```

```
create or replace type BadgeObtenuT as object(  
  badge ref badgeT,  
  date_obtention DATE);
```

```
create or replace type listeBadgeObtenu as table of badgeObtenuT;
```

### Création de Utilisateur :

```
create or replace type utilisateurT as object(  
  pseudo varchar2(50),  
  statut varchar2(10),  
  nom varchar2(50),  
  prenom varchar2(50),  
  photo varchar2(200),  
  badge ListeBadgeObtenu);
```

```
create table Utilisateurs of utilisateurT(  
  pseudo NOT NULL,  
  nom NOT NULL,  
  prenom NOT NULL,  
  primary key (pseudo))  
  nested table badge store as LBadge_Table;
```

```
create or replace type refutilisateurT as object(  
  utilisateur ref utilisateurT);
```

```
create or replace type ListeUtilisateur as table of refUtilisateurT;
```

## Création de Commentaire :

```
create or replace type CommentaireT as object(  
  ID_commentaire number,  
  pseudo ref utilisateurT,  
  signalCommentaire listeUtilisateur,  
  likecommentaire listeUtilisateur,  
  texte clob);
```

```
create table commentaires of commentaireT(  
  pseudo not null,  
  texte not null,  
  scope for (pseudo) is Utilisateurs,  
  primary key (ID_commentaire))  
nested table signalCommentaire store as signalCommentaire_table,  
nested table likeCommentaire store as likeComentaire_table;
```

```
create or replace type RefCommentaire as object(  
  commentaire ref CommentaireT);
```

```
create or replace type listeCommentaire as table of refCommentaire;
```

## Création de Annonce :

```
create or replace type voteAnnonceT as object(  
  utilisateur ref utilisateurT,  
  valeur number);
```

```
create or replace type listeVoteAnnonce as table of voteAnnonceT;
```

```
create or replace type annonceT as object(  
  ID number,  
  enseigne varchar2(100),  
  categorie varchar2(100),  
  image varchar2(200),  
  etat number,  
  date_debut date,  
  date_fin date,  
  lien varchar2(200),  
  code varchar2(50),  
  pseudo ref utilisateurT,  
  pseudo_moderateur ref utilisateurT,  
  commentaires listeCommentaire,  
  signal_annonce listeUtilisateur,  
  vote_annonce listeVoteAnnonce);
```

```
create table Annonces of annonceT(  
  check (categorie='Informatique' or categorie='Voiture' or categorie='Loisir' or categorie='Service'),  
  enseigne not null,  
  etat default 0 not null,  
  pseudo not null,  
  scope for (pseudo) is Utilisateurs,  
  scope for(pseudo_moderateur) is Utilisateurs,  
  
  primary key (ID))  
nested table commentaires store as commentaire_Table,  
nested table signal_annonce store as signalAnnonce_Table,  
nested table vote_annonce store as voteAnnonce_Table;
```

## Exemple de requêtes

### Insertion des badges :

#### *Première insertion :*

```
insert into badges (intitule, condition, graphisme, duree) values ('likeur', 'avoir aimé 10  
commentaires', 'cheminGraphismeBadge1.png', NULL);
```

#### *Deuxième insertion :*

```
insert into badges (intitule, condition, graphisme, duree) values ('Signaleur fou', 'signaler 50  
annonces', 'cheminGraphismeBadge2.png', 5);
```

#### *Troisième insertion :*

```
insert into badges (intitule, condition, graphisme, duree) values ('premier pas', 'avoir publié une  
annonce', 'cheminGraphismeBadge3.png', NULL);
```

#### *Quatrième insertion :*

```
insert into badges (intitule, condition, graphisme, duree) values ('voteur expérimenté', 'avoir voté  
pour 10 annonces', 'cheminGraphismeBadge4.png', 5);
```



## Insertion des utilisateurs :

### *Première insertion :*

```
declare
    ref1 ref badgeT;
    ref2 ref badgeT;
begin
    select ref(i) into ref1
    from badges i
    where intitule='likeur';

    select ref(i) into ref2
    from badges i
    where intitule='Signaleur fou';

    insert into utilisateurs (nom, prenom, badge, photo, pseudo, statut)
values ('Courdavault', 'Antoine', listebadgeObtenu(
    badgeObtenuT(ref1, SYSDATE),
    badgeObtenuT(ref2, SYSDATE)),
'cheminPhoto1.png','acourdav', 'admin');
end;
```

### *Deuxième insertion :*

```
declare
    ref1 ref badgeT;
    ref2 ref badgeT;
begin
    select ref(i) into ref1
    from badges i
    where intitule='premier pas';

    select ref(i) into ref2
    from badges i
    where intitule='Signaleur fou';

    insert into utilisateurs (nom, prenom, badge, photo, pseudo, statut)
values ('Bastoul', 'Théo', listebadgeObtenu(
    badgeObtenuT(ref1, SYSDATE),
    badgeObtenuT(ref2, SYSDATE)),
'cheminPhotoTheo.png','tbastoul', 'admin');

end;
```

### *Troisième insertion :*

```
declare
    ref1 ref badgeT;
    ref2 ref badgeT;
begin
    select ref(i) into ref1
    from badges i
    where intitule='likeur';

    select ref(i) into ref2
    from badges i
    where intitule='Signaleur fou';

    insert into utilisateurs (nom, prenom, badge, photo, pseudo, statut)
    values ('Romon', 'Cailloux', listebadgeObtenu(
        badgeObtenuT(ref1, SYSDATE),
        badgeObtenuT(ref2, SYSDATE)),
        'cheminPhotoPierre.png','promon', 'admin');
end;
```

### *Quatrième insertion :*

```
declare
    ref1 ref badgeT;
    ref2 ref badgeT;
begin
    select ref(i) into ref1
    from badges i
    where intitule='likeur';

    select ref(i) into ref2
    from badges i
    where intitule='voteur expérimenté';

    insert into utilisateurs (nom, prenom, badge, photo, pseudo, statut)
    values ('Duret', 'Hugo', listebadgeObtenu(
        badgeobtenuT(ref1,SYSDATE),
        badgeObtenuT(ref2, SYSDATE)),
        'cheminPhotoHughHugh.png','durethug', 'admin');
end;
```

## Insertion des Annonces

### *Première annonce :*

```
declare
    ref1 ref utilisateurT;
begin
    select ref(i) into ref1
    from utilisateurs i
    where i.pseudo='acourdav';

    insert into annonces (ID, enseigne, categorie, image, etat, Date_debut, Date_fin, lien,
        code, pseudo, pseudo_moderateur, commentaires, signal_annonce, vote_annonce)
    values(1, 'carrefour', 'Informatique', 'ImageAnnonce1.png', 0, SYSDATE,
        TO_DATE('30-06-2018','DD-MM-YYYY'), 'lienVersSite', NULL, ref1, ref1, listeCommentaire(),
        ListeUtilisateur(), listeVoteAnnonce());
end;
```

### *Deuxième annonce :*

```
declare
    ref2 ref utilisateurT;
begin
    select ref(i) into ref2
    from utilisateurs i
    where i.pseudo='durethug';

    insert into annonces (ID, enseigne, categorie, image, etat, Date_debut, Date_fin, lien, code,
        pseudo, pseudo_moderateur, commentaires, signal_annonce, vote_annonce)
    values(2, 'Playboy', 'Loisir', 'ImageAnnonce2.png', 0, SYSDATE,
        TO_DATE('26-06-2018','DD-MM-YYYY'), 'lienVersSite2', NULL, ref2, ref2, listeCommentaire(),
        ListeUtilisateur(), listeVoteAnnonce());
end;
```

## Insertion des commentaires :

### *Premier commentaire :*

```
declare
    ref1 ref utilisateurT;
begin

    select ref(i) into ref1
    from utilisateurs i
    where pseudo='acourdav';

    insert into commentaires values(1, ref1,
listeUtilisateur(),listeUtilisateur(refutilisateurT(ref1)),
    'Une bien belle annonce');
end;
```

### *Deuxième commentaire :*

```
declare
    ref3 ref utilisateurT;
begin

    select ref(i) into ref3
    from utilisateurs i
    where pseudo='tbastoul';

    insert into commentaires values(2, ref3, listeUtilisateur(),
    listeUtilisateur(refutilisateurT(ref3)), 'Une annonce de qualité');
end;
```

### *Troisième commentaire*

```
declare
    ref ref utilisateurT;
begin
    select ref(i) into ref
    from utilisateurs i
    where pseudo='promon';

    insert into commentaires values(3, ref, listeUtilisateur(), listeUtilisateur(), 'Merci beaucoup');
end;
```

### *Insertion d'un commentaire dans une annonce :*

#### *Premier commentaire inséré dans la première annonce:*

```
declare
    ref1 ref CommentaireT;
begin
    select ref(i) into ref1
    from commentaires i
    where ID_commentaire=1;

    insert into the (select i.commentaires from Annonces i where i.ID=1)
    values (refcommentaire( ref1 ));
end;
```

#### *Deuxième commentaire inséré dans la deuxième annonce:*

```
declare
    ref2 ref CommentaireT;
begin
    select ref(i) into ref2
    from commentaires i
    where ID_commentaire=2;

    insert into the (select i.commentaires from Annonces i where i.ID=2)
    values (refcommentaire( ref2 ));
end;
```

### *Troisième commentaire inséré dans une annonce :*

```
declare
    ref1 ref CommentaireT;
begin
    select ref(i) into ref1
    from commentaires i
    where ID_commentaire=3;

    insert into the (select i.commentaires from Annonces i where i.ID=1)
    values (refcommentaire( ref1 ));
end;
```

### Exemple de sélection

#### *Sélection d'un commentaire spécifique d'une annonce*

```
select i2.commentaire.texte from
annonces i1, table(i1.commentaires) i2
where i2.commentaire.ID_commentaire=1
```

#### *Sélection des badges d'un utilisateur*

```
select tb.badge.intitule
from utilisateurs u, table(u.badge) tb
where u.pseudo='acourdav';
```

#### *Sélection de tous les commentaires de l'annonce 1*

```
select c.commentaire.texte
from annonces a, table(a.commentaires) c
where a.ID=1;
```

#### *Sélection du nom et prénom des utilisateurs ayant liker un commentaire de l'annonce 1*

```
select l.utilisateur.nom, l.utilisateur.prenom
from annonces a, table(a.commentaires) c, table(c.commentaire.likecommentaire) l
where a.ID=1;
```

#### *Sélection des noms prénoms et intitulés des badges des utilisateurs ayant liker une annonce, sans doublon et trié par nom.*

```
select l.utilisateur.nom, l.utilisateur.prenom, b.badge.intitule
from annonces a, table(a.commentaires) c, table(c.commentaire.likecommentaire) l,
table(l.utilisateur.badge) b
group by (l.utilisateur.nom, l.utilisateur.prenom, b.badge.intitule)
order by (l.utilisateur.nom);
```

#### *Sélection des nom et prénoms des utilisateurs qui ont le badge "likeur" :*

```
select u.nom, u.prenom
from utilisateurs u, table(u.badge) b
where b.badge.intitule='likeur';
```