



# Application Gange

Zineb Et-tarraf - Nada Boukhriss-Safia Echerif-Youssef Benhachem-Reda Achouhad

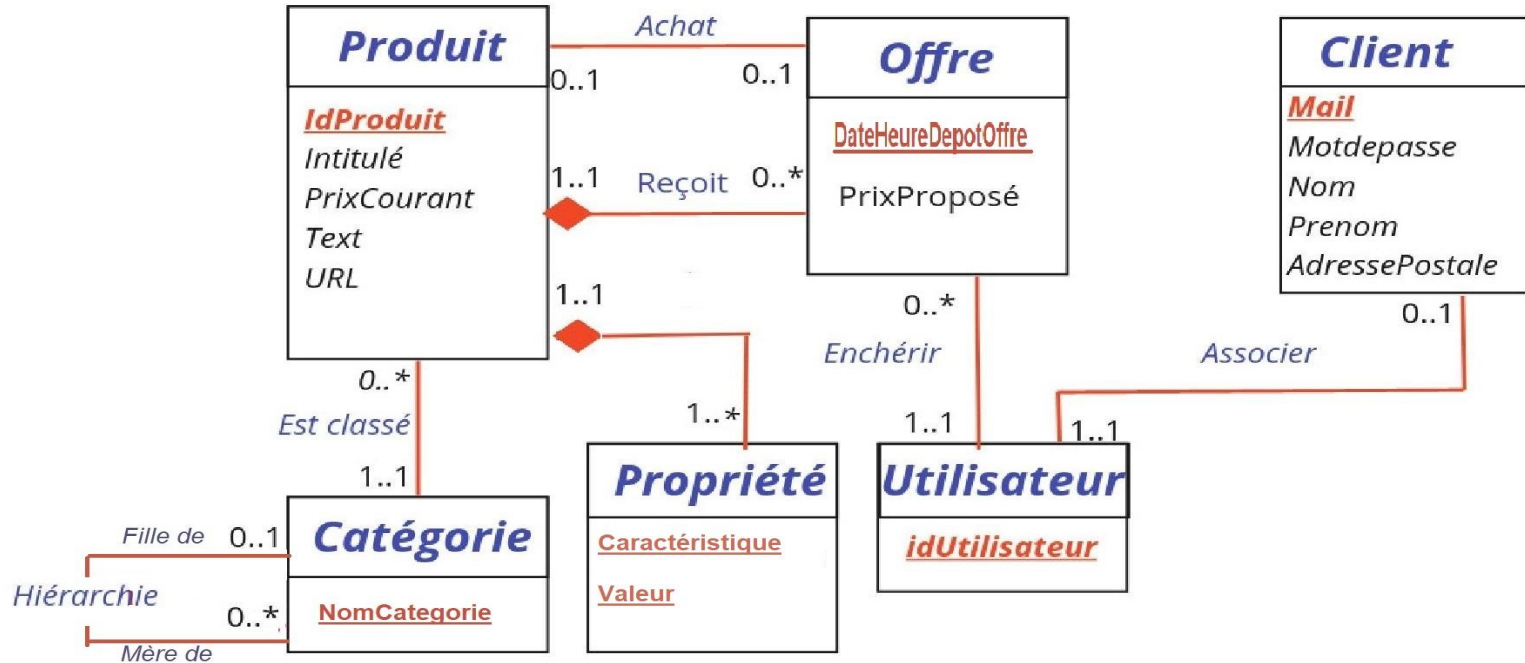


# Introduction

# DF et contraintes

Dépendance Fonctionnelle	Contraintes de Valeur	Contraintes de multiplicité	Contraintes Contextuelles
<ul style="list-style-type: none"><li>- Idproduit → Intitulé, PrixCourant, description, url</li><li>- Idproduit → NomCategorie</li><li>- mail → MotDePasse, NomClient, PrenomClient, AdressePostale</li><li>- mail → idUtilisateur</li><li>- idproduit, dateheureDepotOffre → prix proposé, idUtilisateur <i>(Sur le même produit deux offres ne peuvent être faites au même instant.)</i></li></ul>	<ul style="list-style-type: none"><li>- PrixCourant &gt; 0</li><li>- PrixPropose &gt; PrixCourant (Principe d'enchères)</li></ul>	<ul style="list-style-type: none"><li>- Idproduit --&gt;&gt; caractéristique, valeur ( ' Un produit admet au moins un couple caractéristique/valeur' )</li><li>- idUtilisateur - -&gt; mail (Un utilisateur peut être un client ou non)</li><li>- idUtilisateur -- -&gt;&gt; dateheureDepotOffre, IdProduit (Un utilisateur peut faire des offres de prix à préciser sur un produit)</li><li>- Catégorie -- -&gt;&gt; CatégorieFille (Chaque Catégorie peut avoir des sous-catégories)</li></ul>	<ul style="list-style-type: none"><li>- Le prix proposé doit être toujours supérieur au prix courant</li><li>- La cinquième offre emporte l'enchère</li><li>- Les recommandations concerneront les catégories pour lesquelles l'utilisateur a fait le plus d'offres sans remporter aucune <i>(Recommandations Personnelles)</i>, ensuite les catégories qui ont le plus d'offres en moyenne <i>(Tendances)</i>.</li><li>- Le Client a le droit à l'oubli, il peut supprimer son compte.</li></ul>

# Schéma entité association



## Les entités simples



- Produit (**IdProduit**, Intitulé, PrixCourant, Description, URL)
- Client (**Mail**, MotDePasse, Nom, Prénom, AdressePostale)
- Utilisateur (**IdUtilisateur**)
- Catégorie (**NomCatégorie** )

## Les entités faibles



- Offre (**DateHeureDepotOffre**, **IdProduit**, PrixProposé)
- Propriété(**Caractéristique**, **Valeur**, **IdProduit**(étrangère))

## Associations de cardinalité (1..1)



- **L'association Catégorie / Produit**

Produit (IdProduit, Intitulé, PrixCourant, description, URL, NomCatégorie (étrangère))

- **L'association Utilisateur / Client**

Client (Mail, MotDePasse, Nom, Prénom, AdressePostale, IdUtilisateur (étrangère))

- **L'association Utilisateur / Offre**

Offre (DateHeureDepotOffre, IdProduit (étrangère), PrixProposé, IdUtilisateur (étrangère))

## Associations de cardinalité (0..1)



- **L'association Produit / Offre**

Achat(IdProduit, DateHeureDepotOffre)

- **L'association Catégorie / Catégorie**

Hiérarchie(NomMère, NomFille)



# Schéma Relationnel final



- Produit (IdProduit, Intitulé, PrixCourant, Description, URL, NomCatégorie(étrangère)).
- Propriété(Caractéristique, Valeur, IdProduit(étrangère)).
- Offre (DateHeureDepotOffre, IdProduit (étrangère), PrixProposé, IdUtilisateur (étrangère)).
- Achat (IdProduit, DateHeureDepotOffre).
- Utilisateur (IdUtilisateur) .
- Client (Mail, MotDePasse, Nom, Prénom, AdressePostale, IdUtilisateur('etrang`ere)).
- Catégorie (NomCatégorie ).
- Hiérarchie(NomMère, NomFille).

# Vérification des formes normales



- Propriété, Achat, Utilisateur, Catégorie et Hiérarchie n'ont que des attributs clefs, ils sont donc tous des 3FNBCK
- Vérification de la première forme normale
- Vérification de la deuxième forme normale
- Vérification de la troisième forme normale
- Vérification de la forme normale de Boyce-Codd-Kent

## Login :

Le client ayant déjà un compte dans l'application fait le login en entrant son mail et son mot de passe ensuite on vérifie si ces identifiants existe dans la base de donnée si oui le client se connecte et on conserve son mail dans une variable global le long de sa connexion sinon un message d'alert s'affiche indiquant que les identifiants sont incorrects .





The image shows a 'Create Account' form with the following fields and a button:

- Nom
- Prenom
- Mail
- Adresse Postale
- Password
- Create Account

## Sign Up :

Le client souhaitant créer un compte dans l'application entre ses informations personnelles

- On lui affiche une alerte s'il a oublié de remplir une case .
- Puis on vérifie s'il y a dans la base un client ayant le même mail si oui on lui affiche une alerte .
- Finalement on insère ses informations dans les table Client,Utilisateur(ID bien choisi).

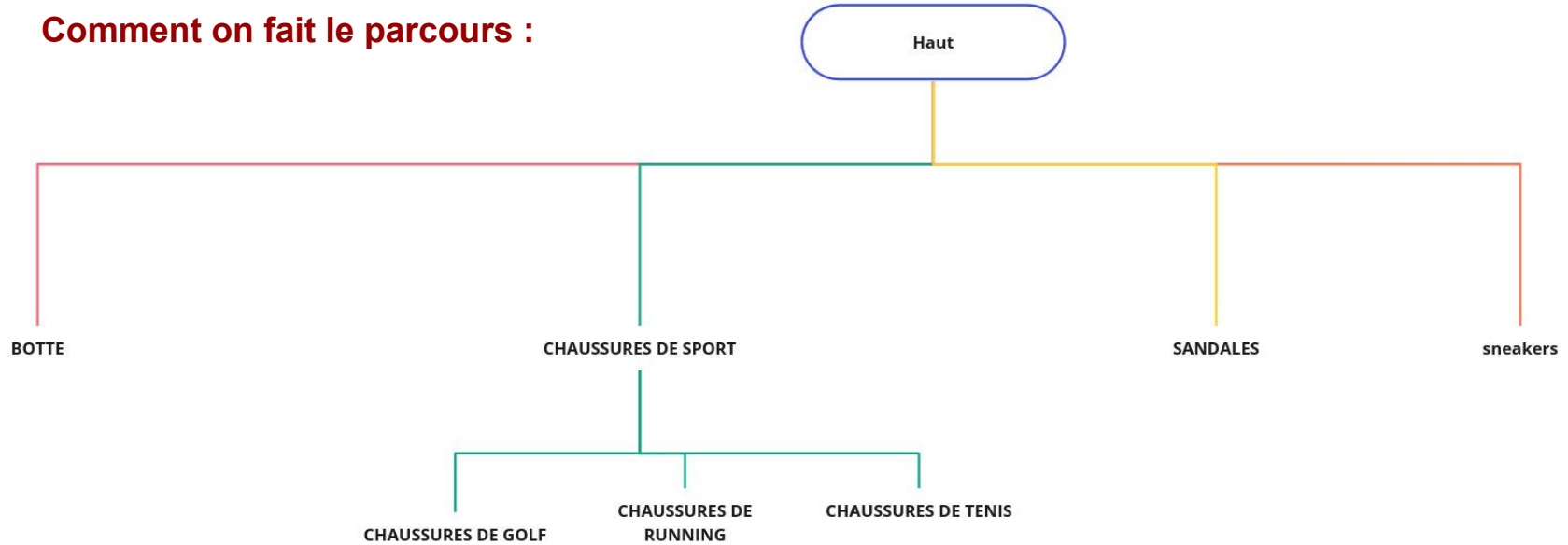
## Parcours des catégories :

Browse Categories		
Browse Categories My Info		
Category Name	Browse Category	Check Products
ACCESSOIRES	Browse Category	Browse Products
BAS	Browse Category	Browse Products
CHAUSSURES	Browse Category	Browse Products
HAUT	Browse Category	Browse Products

Si l'utilisateur choisit de parcourir toutes les catégories via le bouton all categories .On récupère de la base de donnée les catégories root en sélectionnant de la table Hiérarchie toutes les catégories mères .

Ici l'utilisateur a le choix de parcourir les sous catégories ou de parcourir directement les produit ce choix est valide pour toutes les catégories et les sous catégories.

## Comment on fait le parcours :



miro

On a implémenté une fonction récursive ayant pour cas de base : une catégorie qui est à la dernière génération et qui n'a pas de produits et une catégorie qui n'a pas de sous catégories (ex une catégorie root qu'on vient d'ajouter à la base de donnée ) pour ces cas de base la liste de produits qu'on retourne est vide puis on fait un parcours récursif des filles de chaque catégorie.

IDPRODUIT	INTITULE
51	JACKET
80	BIKER CLASSIQUE VEAU NOIR
53	JACKET CUIR HOMME
70	SHERPA TRUCKER JACKET

recupérer les produits pour une catégorie de la dernière génération

0.015 s [Edit](#), [Explain](#), [Export](#)

```

/*for category that have not subcategories return String[] of product's ID if category in Catégorie ; String[] is empty if category has any products*/
SELECT a.* FROM (SELECT PRODUIT.IDPRODUIT, INTITULE FROM PRODUIT JOIN OFFRE on PRODUIT.IDPRODUIT = OFFRE.IDPRODUIT
WHERE NOMCATEGORIE = 'JACKET' GROUP BY PRODUIT.IDPRODUIT,INTITULE ORDER BY COUNT(OFFRE.IDPRODUIT) DESC, INTITULE ) a
UNION ALL SELECT b.* FROM (SELECT PRODUIT.IDPRODUIT,INTITULE FROM PRODUIT WHERE NOMCATEGORIE = 'JACKET' AND PRODUIT.IDPRODUIT NOT IN (SELECT DISTINCT OFFRE.IDPRODUIT FROM OFFRE) ORDER BY INTITULE) b;

```

On récupère ainsi la liste des produits de chaque catégorie que se soit une catégorie parcourable ou pas .Puis on les ordonne par ordre décroissant du nombre d'offres qui ont été faites et par ordre alphabétique.

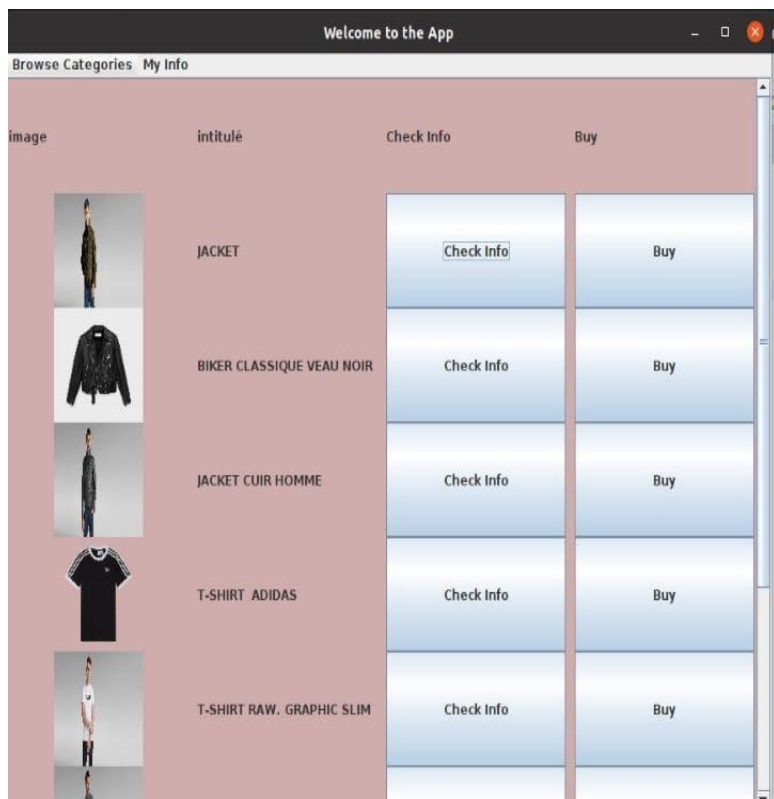
IDPRODUIT	INTITULE
51	JACKET
80	BIKER CLASSIQUE VEAU NOIR
53	JACKET CUIR HOMME
70	SHERPA TRUCKER JACKET

0.012 s [Edit](#), [Explain](#), [Export](#)

```

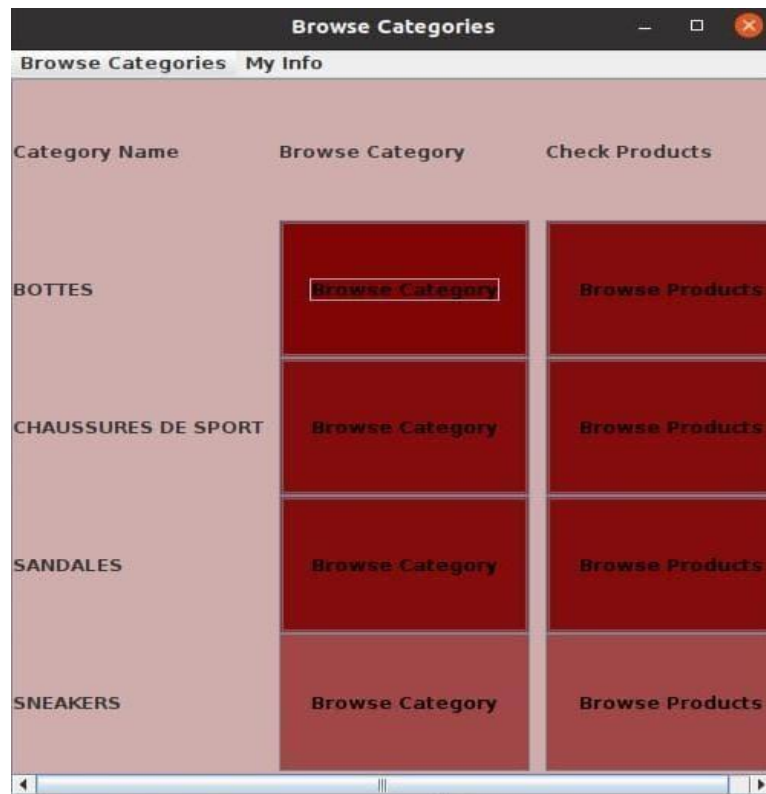
SELECT a.* FROM (SELECT PRODUIT.IDPRODUIT, INTITULE FROM PRODUIT JOIN OFFRE on PRODUIT.IDPRODUIT = OFFRE.IDPRODUIT WHERE PRODUIT.IDPRODUIT IN (53,80,51,70)
GROUP BY PRODUIT.IDPRODUIT,INTITULE ORDER BY COUNT(OFFRE.IDPRODUIT) DESC, INTITULE ) a
UNION ALL SELECT b.* FROM (SELECT PRODUIT.IDPRODUIT,INTITULE FROM PRODUIT WHERE PRODUIT.IDPRODUIT IN (53,80,51,70) AND PRODUIT.IDPRODUIT
NOT IN (SELECT DISTINCT OFFRE.IDPRODUIT FROM OFFRE) ORDER BY INTITULE) b

```



Exemple de parcours direct des produits de la catégorie Haut

Exemple de parcours par sous catégories pour la catégorie Chaussure







# Les Recommandations:

Recommandations  
Personnelles

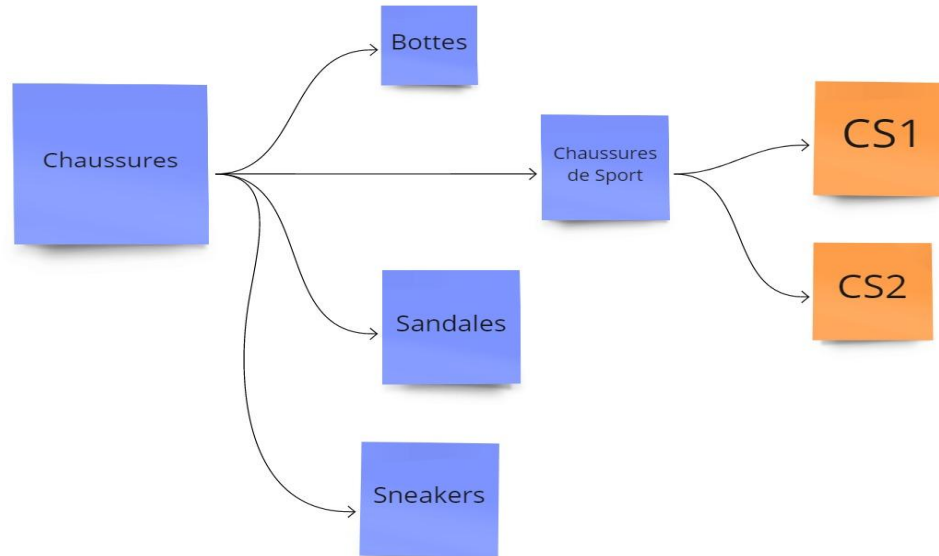


Recommandations  
des Tendances



# Les Recommandations Personnelles:

## Une Première Idée:



# Les Recommandations Personnelles:

## Son Implémentation SQL:

```
CREATE VIEW MostBidOn AS
Select categorie.nom
FROM (Categorie join Produit on Produit.NomCategorie = Categorie.nom) join offre on offre.idProduit = Produit.idProduit
Where Offre.IdUtilisateur = 4 and  offre.idProduit not in (Select idProduit from achat)
group by (categorie.nom)
Order by count(Offre.DATEHEUREDEPOTOFFRE) DESC;
```

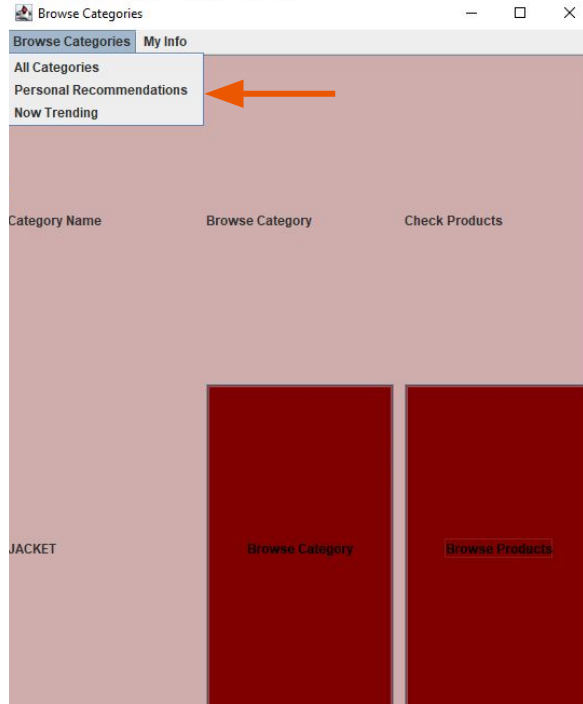
```
CREATE VIEW Mothers AS
Select distinct hierarchie.Nommere as nom
From (Select categorie.nom as nom
FROM (Categorie join Produit on Produit.NomCategorie = Categorie.nom) join offre on offre.idProduit = Produit.idProduit
Where Offre.IdUtilisateur = 4 and  offre.idProduit not in (Select idProduit from achat)
group by (categorie.nom)
Order by count(Offre.DATEHEUREDEPOTOFFRE) DESC ) join hierarchie on hierarchie.nomfille = nom;
```

```
Select hierarchie.nomfille
From ((hierarchie join mothers on mothers.nom = hierarchie.nommere) join Produit on Produit.NomCategorie = hierarchie.nomfille) join offre on offre.idProduit = Produit.idProduit
group by (hierarchie.nomfille)
Order by count(Offre.DATEHEUREDEPOTOFFRE) DESC;
```

```
Drop View MostBidON;
Drop View Mothers;
```

# Les Recommandations Personnelles:

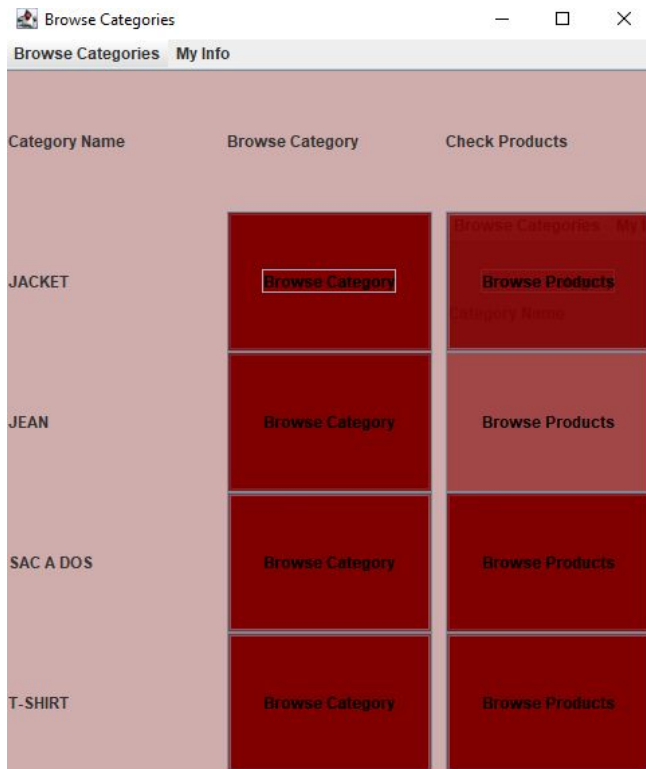
## Notre Version Finale:



## La requête SQL:

```
Select categorie.nom  
FROM (Categorie join Produit on Produit.NomCategorie = Categorie.nom) join offre on offre.idProduit = Produit.idProduit  
Where Offre.IdUtilisateur = 4 and offre.idProduit not in (Select idProduit from achat)  
group by (categorie.nom)  
Order by count(Offre.DATEHEUREDEPOTOFFRE) DESC;
```

## Les Recommandations Tendances:



### La requête SQL:

```
Create view newtable as
Select offre.idproduit as prod, count(Offre.DATEHEUREDEPOTOFFRE) as nboffres
from offre
group by offre.idproduit;
```

```
select produit.nomcategorie, avg(newtable.nboffres)
from produit join newtable on produit.idproduit = newtable.prod
group by produit.nomcategorie
order by avg(newtable.nboffres) desc;
```

```
drop view newtable;
```

# Les Enchères et Les Achats:

Si une offre acceptable est faite :

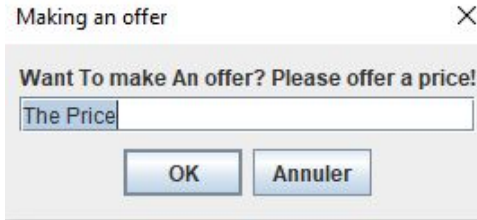
```
Update Produit set prixcourant=? where idproduit = ?;  
INSERT INTO OFFRE VALUES(CURRENT_TIMESTAMP,?,?,?);
```

(Mise à jour du prix courant et ajout de l'offre)

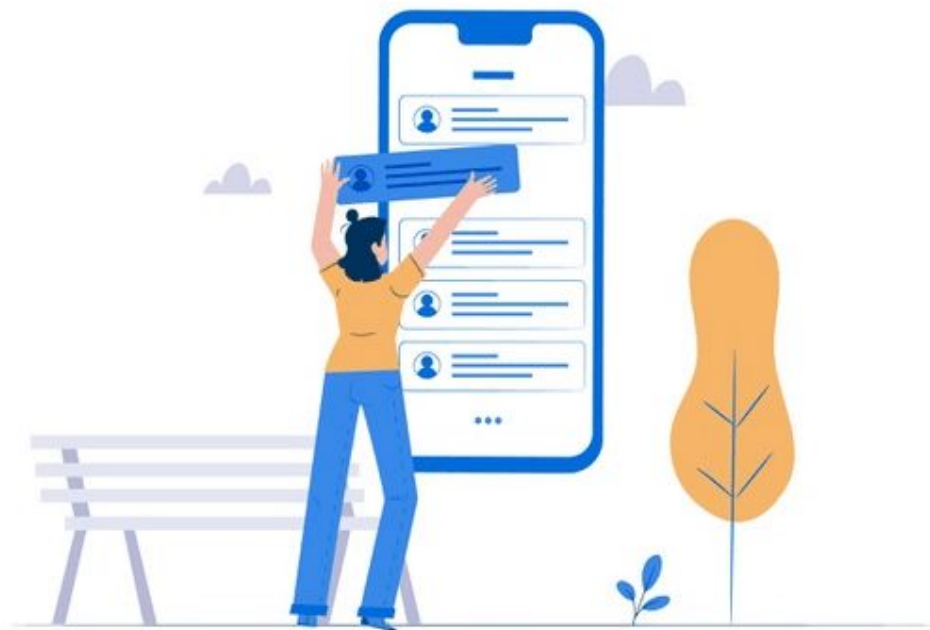
Si De plus c'est une offre gagnante:

```
INSERT INTO ACHAT VALUES(?, (select DATEHEUREDEPOTOFFRE from offre where idProduit = ? and PRIXPPOSE = (select max(PRIXPPOSE) from offre where idProduit = ? )))
```

(Ajout de l'achat en plus)



## Droit à l'oubli (RGPD):



## Droit à l'oubli (RGPD):



### Problème

Client : comment puis-je renoncer à la création de mon compte sur cette application?



### Solution

C'est simple : il faut juste consulter la fiche personnelle, et cliquer sur le bouton delete et c'est bon.

Your Personal Info	
Browse Categories My Info	
Name	Comoth
Last Name	Antoin
Adresse Mail	antoin@gmail.com
Mot de Passe	123456789
Adresse Postale	38 Paris
Delete Account	<button>Delete Account</button>





## Droit à l'oubli (RGPD):

### Implantation de cette fonctionnalité :

```
oldIdDeCetUtilisateur = SELECT IDUTILISATEUR FROM CLIENT WHERE MAIL = mailDeCetUtilisateur;  
newIdDeCetUtilisateur = SELECT Max(IDUTILISATEUR) FROM UTILISATEUR + 1 ;  
INSERT INTO UTILISATEUR VALUES(newIdDeCetUtilisateur);  
UPDATE OFFRE SET IDUTILISATEUR=newIdDeCetUtilisateur WHERE IDUTILISATEUR = oldIdDeCetUtilisateur;  
Delete from client where mail = mailDeCetUtilisateur;  
Delete from UTILISATEUR where IDUTILISATEUR = oldIdDeCetUtilisateur;
```



## Conclusion :

- Problèmes rencontrés :
  - C'était difficile de implémenter les fonctionnalités en utilisant sql seule, d'où l'importance du Java.
  - Coordination entre les différents membres du groupe.
  - Ce qui en résulte une redondance dans notre travail.



**Merci pour  
votre attention.**