

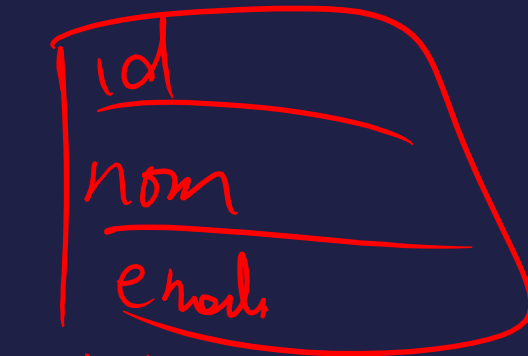
# CONCEPTION

Concevoir une base de données

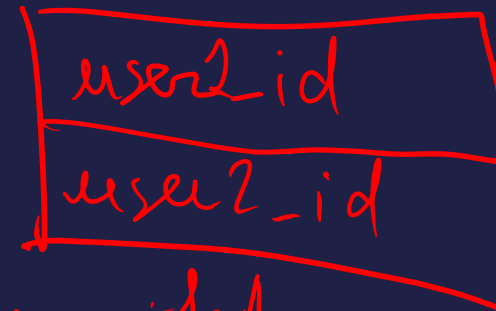


# METHODOLOGIE

1. Identifier les entités (noms communs)
2. Identifier les relations (verbes)
3. Identifier la cardinalité (se poser les bonnes questions)
4. En déduire où placer les FK (sur la patte n si 1-n, dans une nouvelle structure si n-n)



utilisateur



n : follow - utilisateur - utilisateur



- 1 utilisateur peut faire combien d'utilisateurs?  
N utilisateurs.
- 1 utilisateur peut être followed par combien d'utilisateur ? N utilisateurs

- constraint techniques (null, type, unique)

- relation

FK

~~1-1~~

nouvelle table

1	• follower_id	→ 1 eng user
2	• follower_id	→ 1 eng user

relation

1 eng user - 1 eng user

Vous souhaitez faire un reseau social comme twitter.  
Le client vous demande de gérer les personnes qui s'inscrivent sur le site et les follows (1 personne peut follow un autre utilisateur).

Veuillez concevoir la  
base de données.



# SOLUTION

1. Identifier les entités (noms communs)
2. Identifier les relations (verbes)
3. Identifier la cardinalité (se poser les bonnes questions)
4. En déduire où placer les FK (sur la patte n si 1-n, dans une nouvelle structure si n-n)



Il s'agit tout d'abord d'identifier les entités/tables de la demande :

- Utilisateur

Les amis ne sont pas une entité à part, car les amis sont en fait des utilisateurs !

1. Identifier les entités (noms communs)
2. Identifier les relations (verbes)
3. Identifier la cardinalité (se poser les bonnes questions)
4. En déduire où placer les FK (sur la patte n si 1-n, dans une nouvelle structure si n-n)

Regardons les relations possibles en chaque pair de tables :

1. Utilisateur – Utilisateur ? (follow)

1. Identifier les entités (noms communs)
2. Identifier les relations (verbes)
3. Identifier la cardinalité (se poser les bonnes questions)
4. En déduire où placer les FK (sur la patte n si 1-n, dans une nouvelle structure si n-n)

On va traiter la 1<sup>er</sup> relation :  
Utilisateur – Utilisateur ? (follow)

Par exemple, une bonne réponse peut être :

- Un « Utilisateur » peut «follow » combien de «Utilisateur » ? Plusieurs utilisateur
- Un «Utilisateur » peut être « followed » par combien de « Utilisateur » ? Plusieurs utilisateur

C'est donc une relation ManyToMany (ou n-n) :

- Plusieurs utilisateurs
- Plusieurs utilisateurs

1. Identifier les entités (noms communs)
2. Identifier les relations (verbes)
3. Identifier la cardinalité (se poser les bonnes questions)
4. En déduire où placer les FK (sur la patte n si 1-n, dans une nouvelle structure si n-n)



id	...
nom	...
email	...

utilisateur

follower_id	...
followee_id	...

follow\_utilisateur\_utilisateur

