

# CARDINALITES

Comprendre les relations et les  
cardinalités



# Exercices

Comment stocker la relation  
« avoir » entre « Hotel » et  
« Chambre » ? (15 min)



Par exemple, une bonne réponse peut être :

- Un « Hotel » peut « avoir » dans combien de « Chambre » ? Plusieurs chambres
- Une « Chambre » peut être « eu/possédé » pour combien de « Hotel » ? Un seul hotel

C'est donc une relation OneToMany (ou 1-n) :

- Plusieurs chambres
- Un seul hotel

Pour stocker la relation 1 hotel – n chambres, on va créer une nouvelle colonne à la patte n, donc chambre :

id	...
nom	...
adresse	...

hotel

id	...
numero	...
capacite	...
avoir_hotel_id	...

chambre

Comment stocker la relation  
« contracter » entre « Compte  
bancaire » et « Client » ? (15  
min)





Par exemple, une bonne réponse peut être :

- Un « Client » peut « contracter » combien de « Compte bancaire » ? Plusieurs comptes bancaires
- Un « Compte bancaire » peut être « contracté » par combien de « Client » ? Plusieurs clients

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

- Plusieurs comptes bancaires
- Plusieurs clients

Pour une relation ManyToMany, nous allons créer une nouvelle structure.

# On stocke la relation « contrater »

id	...
rib	...
solde	...

compte

compte_id	...
client_id	...

contrater\_compte\_client

id	...
nom	...
prenom	...

client

Comment stocker la relation  
« visiter » entre « Touriste » et  
« Monument » ? (15 min)



Par exemple, une bonne réponse peut être :

- Un « Touriste » peut « visiter » combien de « Monument » ? Plusieurs monuments
- Un « Monument » peut être « visité » par combien de « Touriste » ? Plusieurs touristes

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

- Plusieurs monuments
- Plusieurs touristes

Pour une relation ManyToMany, nous allons créer une nouvelle structure.

# On stocke la relation « visiter »

id	...
adresse	...

monument

monument_id	...
touriste_id	...

visiter\_monument\_touriste

id	...
nom	...
prenom	...

touriste

Comment stocker la relation  
« composer » entre « Artiste » et  
« Musique » ? (15 min)





Par exemple, une bonne réponse peut être :

- Un « Artiste » peut « composer » combien de « Musique » ? Plusieurs musiques
- Une « Musique » peut être « composée » par combien de « Artiste » ? Plusieurs artistes

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

- Plusieurs musiques
- Plusieurs artistes

Pour une relation ManyToMany, nous allons créer une nouvelle structure.

# On stocke la relation « composer »

id	...
genre	...

musique

musique_id	...
artiste_id	...

composer\_musique\_artiste

id	...
nom	...
prenom	...

artiste

Comment stocker la relation  
« produire » entre  
« Producteur » et « Film » ? (15  
min)



Par exemple, une bonne réponse peut être :

- Un « Producteur » peut « produire » combien de « Film » ? Plusieurs films
- Un « Film » peut être « produit » par combien de « Producteur » ? Un seul producteur

C'est donc une relation OneToMany (ou 1-n) :

- Plusieurs films
- Un seul producteur

Pour stocker la relation 1 producteur – n films,  
on va créer une nouvelle colonne à la patte n,  
donc film :

id	...
nom	...
adresse	...

producteur

id	...
titre	...
duree	...
produit_producteur_id	...

film

Comment stocker la relation  
« envoyer » entre « Message »  
et « Expéditeur » ? (15 min)





Par exemple, une bonne réponse peut être :

- Un « Expéditeur » peut «envoyer » combien de « Message » ? Plusieurs messages
- Un « Message » peut être «envoyé » par combien de « Expéditeur » ? Un seul expéditeur

C'est donc une relation OneToMany (ou 1-n) :

- Plusieurs messages
- Un seul expéditeur

Pour stocker la relation 1 expéditeur – n messages, on va créer une nouvelle colonne à la patte n, donc message :

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

expediteur

id	...
objet	...
corps de email	...
envoi_expediteur_id	...

message

Un utilisateur commande des produits sur un site e-commerce. On veut stocker quel produit est dans quelle commande. Comment faire ? (15 min)



Comment stocker la relation  
« composer » entre « Produit »  
et « Commande » ?

Par exemple, une bonne réponse peut être :

- Un « Produit » peut « composer » combien de « Commande » ? Plusieurs commandes
- Une « Commande » peut être « composée » de combien de « Produit » ? Plusieurs produits

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

- Plusieurs commandes
- Plusieurs produits

Pour une relation ManyToMany, nous allons créer une nouvelle structure.

# On stocke la relation « composer »

id	...
ref	...

commande

commande_id	...
produit_id	...

composer\_commande\_produit

id	...
titre	...
prix	...

produit



On réalise un site comme leboncoin.  
On veut stocker quel particulier a  
posté quelle annonce. Comment  
faire ? (15 min)



Comment stocker la relation  
« poster » entre « Particulier » et  
« Annonce » ?

Par exemple, une bonne réponse peut être :

- Un « Particulier » peut « poster » combien de « Annonce » ? Plusieurs annonces
- Une « Annonce » peut être « posté » par combien de « Particulier » ? Un seul particulier

C'est donc une relation OneToMany (ou 1-n) :

- Plusieurs annonces
- Un seul particulier

Pour stocker la relation 1 particulier – n annonces, on va créer une nouvelle colonne à la patte n, donc message :

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

particulier

id	...
objet	...
contenu	...
post_particulier_id	...

annonce

On réalise un site pour le  
gouvernement.

On veut stocker quelle voiture a été  
immatriculée à quelle préfecture.

Comment faire ? (15 min)



Comment stocker la relation  
« immatriculer » entre  
« Voiture » et « Préfecture » ?

Par exemple, une bonne réponse peut être :

- Une « Préfecture » peut « immatriculer » combien de « Voiture » ? Plusieurs voitures
- Une « Voiture » peut être «immatriculée » chez combien de « Préfecture » ? Plusieurs préfectures (en cas de changement de propriétaire)



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

- Plusieurs voitures
- Plusieurs préfectures

Pour une relation ManyToMany, nous allons créer une nouvelle structure.

# On stocke la relation « immatriculer »

id	...
ville	...

prefecture

prefecture_id	...
voiture_id	...

immatriculation\_voiture\_prefecture

id	...
modele	...
puissance	...

voiture

On réalise un site comme  
booking.com.

On veut stocker quel particulier a  
réservé quelle chambre. Comment  
faire ? (15 min)



Comment stocker la relation  
« reserver » entre « Particulier »  
et « Chambre » ?

Par exemple, une bonne réponse peut être :

- Un « Particulier » peut « réserver » combien de « Chambre » ? Plusieurs chambres
- Une « Chambre » peut être « réservée » par combien de « Particulier » ? Plusieurs particuliers

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

- Plusieurs chambres
- Plusieurs particuliers

Pour une relation ManyToMany, nous allons créer une nouvelle structure.

# On stocke la relation « reserver »

id	...
numero	...

chambre

chambre_id	...
particulier_id	...

reserver\_particulier\_chambre

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

particulier

