XEBIA NATIS

CARDINALITES

Comprendre les relations et les cardinalités

XWIKI SAS

XWIKI NATIS

Exercices #3

XWIKI NATIS

Dans ces exercices, il vous est demandé de stocker les relations de tables d‘un système dans leur globalité

XWIKI NATIS

Vous gérez le système de gestion de Carrefour.
On vous demande de faire l'évolution suivante : il faut stocker les fournisseurs, les produits qu'ils apportent et les rayons dans lesquels il faut mettre les produits.

Comment stocker toutes ces informations ? (20 min)

XWNATIS

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

- Fournisseur
- Produit
- Rayon

XWNATIS

Regardons les relations possibles en chaque paire de tables :

1. Fournisseur – Produit ?
2. Fournisseur – Rayon ?
3. Produit – Rayon ?

XMNATIS

Regardons les relations possibles en chaque paire de tables :

1. Fournisseur – Produit ? (fournir)
2. Fournisseur – Rayon ? (aucun)
3. Produit – Rayon ? (composer)

XWIKI NATIS

Regardons les relations possibles en chaque pair de tables :

1. Fournisseur – Produit ? (fournir)
2. ~~Fournisseur – Rayon ? (aucun)~~
3. Produit – Rayon ? (composer)

XWNATIS

Nous avons donc 2 relations que l'on souhaite stocker :

1. Fournisseur – Produit ? (fournir)
2. Produit – Rayon ? (composer)

XWIKI NATIS

On va traiter la 1er relation :

- Fournisseur – Produit ? (fournir)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Un « Fournisseur » peut « fournir » combien de « Produit » ? Plusieurs produits
- Un «Produit » peut être « fourni » de combien de « Fournisseur » ? Plusieurs fournisseurs

XWIKI NATIS

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

- Plusieurs fournisseus
- Plusieurs produits

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

XWIKI NATIS

On stocke la relation « fournir »

id	…
ref	…

fournisseur

fournisseur_id	…
produit_id	…

Fournir_founisseur_produit

id	…
titre	…
prix	…

produit

XWIKI NATIS

On va traiter la 2eme relation :

- Produit – Rayon ? (composer)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Un « Produit » peut « composer/être dans » combien de « Rayon » ? 1 seul rayon
- Un « Rayon » peut être « composé » de combien de « Produit » ? Plusieurs produits

XWIKI NATIS

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

- Plusieurs produits
- Un seul rayon

XWIKI NATIS

On stocke la relation « composer »

id	…
ref	…

fournisseur

fournisseur_id	…
produit_id	…

Fournir_founisseur_produit

id	…
nom	…

rayon

id	…
titre	…
prix	…
rayon_id	…

produit

XWIKI NATIS

Vous faites un site de location de voitures. Le client vous demande de gérer les voitures, les personnes qui s‘inscrivent sur le site et leur réservation.

Comment stocker toutes
ces informations ? (20 min)

XWIKI NATIS

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

- Voiture
- Utilisateur
- Réservation

XWIKI NATIS

Regardons les relations possibles en chaque paire de tables :

1. Voiture – Utilisateur ?
2. Voiture – Réservation ?
3. Utilisateur – Réservation ?

XWINATIS

Regardons les relations possibles en chaque paire de tables :

1. Voiture – Utilisateur ? (aucun lien de manière générale, sans le cas de la réservation)
2. Voiture – Réservation ? (réserver)
3. Utilisateur – Réservation ? (effectuer)

XWIKI NATIS

Regardons les relations possibles en chaque pair de tables :

1. ~~Voiture – Utilisateur ? (aucun lien de manière générale, sans le cas de la réservation)~~
2. Voiture – Réservation ? (réserver)
3. Utilisateur – Réservation ? (effectuer)

XWINATIS

Nous avons donc 2 relations que l’on souhaite stocker :

1. Voiture – Réservation ? (réserver)
2. Utilisateur – Réservation ? (effectuer)

XWNATIS

On va traiter la 1er relation :

- Voiture – Réservation ? (réserver)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Une « Réservation » peut « concerner » combien de « Voiture » ? 1 seule voiture
- Une « Voiture » peut être « réservée » via combien de « Réservation » ? Plusieurs réservations (sur des jours différents par exemple)

XWIKI NATIS

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

- Plusieurs réservations
- Une seule voiture

XWNATIS

On stocke la relation « réserver »

id	…
modele	…
couleur	..
puissance	..

voiture

id	…
date	…
prix	…
voiture_id	…

reservation

XMNATIS

On va traiter la 2eme relation :

- Utilisateur – Réservation ? (effectuer)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Un « Utilisateur » peut « effectuer » combien de « Réservation » ? Plusieurs réservations
- Une « Réservation » peut être « effectuée » par combien d’ « Utilisateur » ? 1 seul utilisateur

XEBIA NATIS

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

- Plusieurs réservations
- Un seul utilisateur

XWNATIS

On stocke la relation « effectuer »

id	…
modele	…
couleur	..
puissance	..

voiture

id	…
nom	…
email	..

utilisateur

id	…
date	…
prix	…
voiture_id	…
utilisateur_id	

reservation

XWIKI NATIS

Un client vous demande de réaliser un site pour gérer des recettes de cuisine. On vous demande de gérer les recettes et leurs étapes ainsi que les ingrédients.

Comment stocker toutes
ces informations ? (20 min)

XWIKI NATIS

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

- Recette
- Etape
- Ingredient

XWNATIS

Regardons les relations possibles en chaque paire de tables :

1. Recette – Etape ?
2. Recette – Ingredient ?
3. Etape – Ingredient ?

XEBIA NATIS

Regardons les relations possibles en chaque paire de tables :

1. Recette – Etape ? (composer)
2. Recette – Ingredient ? (utiliser)
3. Etape – Ingredient ? (utiliser)

XMNATIS

On va traiter la 1er relation :
Recette – Etape ? (composer)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Une « Etape » peut « composer » combien de « Recette » ? Plusieurs recettes
- Une « Recette » peut être «composée » de combien de « Etape » ? Plusieurs étapes

XEBIANATIS

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

- Plusieurs recettes
- Plusieurs étapes

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

XWIKI NATIS

On stocke la relation « composer »

id	…
titre	…
duree	…

recette

recette_id	…
etape_id	…

composition_recette_etape

id	…
contenu	…

etape

XWIKI NATIS

On va traiter la 2eme relation :
Recette – Ingredient ? (utiliser)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Une « Recette » peut « utiliser » combien d’ « Ingredient » ? Plusieurs ingrédients
- Un « Ingredient » peut être « utilisé » dans combien de « Recette » ? Plusieurs recettes

XEBIA NATIS

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

- Plusieurs recettes
- Plusieurs ingrédients

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

XDNATIS

On stocke la relation « utiliser »

id	…
titre	…
duree	…

recette

recette_id	…
etape_id	…

composition_recette_etape

recette_id	…
ingredient_id	…

utilisation_recette_ingredient

id	…
contenu	…

etape

id	…
nom	…

ingredient

XWIKI NATIS

On va traiter la 3eme relation :
Etape – Ingredient ? (utiliser)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Une « Etape » peut « utiliser » combien de « Ingredient » ? Plusieurs ingrédients
- Un « Ingredient » peut être « utilisé » dans combien d’ « Etape » ? Plusieurs étapes

XEBIA NATIS

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

- Plusieurs étapes
- Plusieurs ingrédients

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

XWNATIS

On stocke la relation « utiliser »

id	…
contenu	…

etape

id	…
titre	…
duree	…

recette

recette_id	…
etape_id	…

composition_recette_etape

ingredient_id	…
etape_id	…

utilisation_etape_ingredient

recette_id	…
ingredient_id	…

utilisation_recette_ingredient

id	…
nom	…

ingredient

XWIKI NATIS

On vous demande d‘intervenir sur un site existant de formation en ligne. On vous demande de gérer les videos de chaque cours, ainsi que le langage auquel elles se rapportent.

Comment stocker toutes
ces informations ? (20 min)

XWIKI NATIS

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

- Video
- Cours
- Langage

XWINATIS

Regardons les relations possibles en chaque pair de tables :

1. Video – Cours ?
2. Video – Langage ?
3. Cours – Langage ?

XWNATIS

Regardons les relations possibles en chaque pair de tables :

1. Video – Cours ? (composer)
2. Video – Langage ? (aucun, une video n’as pas de lien direct avec le langage, c’est plutôt le cours qui a un lien avec le langage)
3. Cours – Langage ? (introduire)

XWIKI NATIS

Regardons les relations possibles en chaque pair de tables :

1. Video – Cours ? (composer)
2. ~~Video – Langage ? (aucun, une video n'as pas de lien direct avec le langage, c'est plutôt le cours qui a un lien avec le langage)~~
3. Cours – Langage ? (introduire)

XWIKI NATIS

Nous avons ici 2 relations :

1. Video – Cours ? (composer)
2. Cours – Langage ? (introduire)

XWIKI NATIS

On va traiter la 1er relation :
Video – Cours ? (composer)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Une « Video » peut « composer » combien de « Cours » ? 1 seul cours
- Un « Cours » peut être « composé » de combien de « Video » ? Plusieurs vidéos

XDNATIS

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

- Plusieurs videos
- Un seul cours

XDNATIS

On stocke la relation « composer »

id	…
intitulé	…
couleur	..
puissance	..

cours

id	…
date	…
prix	…
cours_id	…

video

XEBIANATIS

On va traiter la 2eme relation :
Cours – Langage ? (introduire)

XWNATIS

Par exemple, une bonne réponse peut être :

- Une « Cours » peut « introduire » combien de «Langage » ? 1 seul langage
- Un « Langage » peut être «introduit » de combien de « Cours » ? Plusieurs cours

XWIKI NATIS

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

- Plusieurs cours
- Un seul langage

XNATIS

On stocke la relation « composer »

id	…
intitulé	…
couleur	…
puissance	…
langage_id	…

cours

id	…
nom	…

langage

id	…
date	…
prix	…
cours_id	…

video

XWIKI NATIS

Vous sohaitez faire un reseau social comme facebook. Le client vous demande de gérer les personnes qui s‘inscrivent sur le site et les demandes d‘amis.

Comment stocker toutes
ces informations ? (20 min)

XWIKI NATIS

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

- Utilisateur
- Demande (demande d’ami)

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

XWIKI NATIS

Regardons les relations possibles en chaque pair de tables :

1. Utilisateur – Demande ? (envoyer)

XWINATIS

Regardons les relations possibles en chaque pair de tables :

1. Utilisateur – Demande ? (envoyer)
2. Utilisateur – Demande ? (recevoir)

Il s’agit bien de 2 relations différentes, mais entre 2 mêmes entités !

XWIKI NATIS

On va traiter la 1er relation :
Utilisateur – Demande ? (envoyer)

XNATIS

Par exemple, une bonne réponse peut être :

- Un « Utilisateur » peut «envoyer » combien de « Demande » ? Plusieurs demande
- Un « Demande » peut être «envoyé » par combien de « Utilisateur » ? 1 seul utilisateur

XWIKI NATIS

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

- Plusieurs demandes
- Un seul utilisateur

XWNATIS

On stocke la relation « envoyer »

id	…
nom	…
email	..

utilisateur

id	…
date	…
envoyeur_id	..

demande

XWIKI NATIS

On va traiter la 2eme relation :
Utilisateur – Demande ? (recevoir)

XWIKI NATIS

Par exemple, une bonne réponse peut être :

- Un « Utilisateur » peut « recevoir » combien de « Demande » ? Plusieurs demande
- Un « Demande » peut être « recu » par combien de « Utilisateur » ? 1 seul utilisateur

XWIKI NATIS

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

- Plusieurs demandes
- Un seul utilisateur

XWNATIS

On stocke la relation « envoyer »

id	…
nom	…
email	…

utilisateur

id	…
date	…
envoyeur_id	…
receveur_id	…

demande

XONATIS