

UE 8 : SIG

Cours

>> Voici une des parties du cours qui a été mise à jour

Normalisation des relations

On mesure la qualité d'une relation par son degré de normalisation :

Première Forme Normale (1FN)

Pour être 1FN, il faut que chaque attribut soit atomique = Aucune subdivision de la donnée initiale n'apporte une info supplémentaire ou complémentaire.

=> Donc aucun attribut ne doit pas être multivalué = plusieurs valeurs listées (ex : Téléphone 1, Téléphone 2..) ou composé = plusieurs informations groupées (ex : Nom + Prénom).

Exemple d'une table SALARIÉ qui ne respecte pas la 1FN :

NumIdentifiant	Nom	Email
101	Madame Julie BRUN	julie.brun@la-normalisation.fr, julie.b@forme-normale.fr
102	Monsieur Jean ROBERT	Jean-robert@la-normalisation.fr

Dans cette table :

- ⇒ Si un salarié a plusieurs adresses mails alors elles sont regroupées dans l'attribut Email (séparées par des virgules).
- ⇒ L'attribut « Nom » ne contient pas que le Nom mais contient plusieurs informations : Civilité, Prénom et Nom.

✓ Pour passer en 1FN il faut modifier la table de cette façon :

NumIdentifiant	Civilité	Prénom	Nom	Email 1	Email 2
101	Madame	Julie	BRUN	julie.brun@la-normalisation.fr	j.b@forme-normale.fr
102	Monsieur	Jean	ROBERT	Jean-robert@la-normalisation.fr	

Par contre si un salarié peut avoir par exemple des dizaines d'adresses mails on ne va pas créer des dizaines de colonnes juste au cas où => Dans ce cas-là il faudra modifier directement le modèle de données.

Dans l'exemple il faut donc transformer l'attribut multivalué « Email » en une autre table distincte qui sera alors liée à la table d'origine « SALARIÉ » par une relation de type « un à plusieurs ».

MPD = Création d'une nouvelle table Email qui liste toutes les adresses mail des salariés

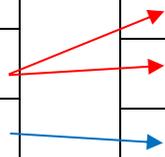
SALARIE	
<u>NumIdentifiant</u>	
Civilite	
Prenom	
Nom	



Contenu de la table EMAIL :

SALARIE		
Identifiant	Prénom	...
101	Julie	
102	Jean	

EMAIL	
SALARIE_NumIdentifiant	Email
101	julie.brun@la-normalisation.fr
101	j.b@forme-normale.fr
102	Jean-robert@la-normalisation.fr



Deuxième Forme Normale (2FN)

La deuxième forme normale (2FN) ne concerne que les tables avec une clé primaire composite = clé primaire composée de plusieurs champs.

Une relation est en 2FN si :

- ✓ Elle est en 1FN
- ✓ Chaque attribut qui ne fait pas partie de la clé primaire dépend de la clé primaire dans son intégralité = Si un attribut ne fait pas partie de la clé primaire il ne doit pas dépendre que d'une partie de la clé primaire

=> La 2FN permet donc d'éliminer les attributs qui ne décrivent pas l'entité représentée par la relation : gain d'espace de stockage et mise à jour de données améliorée.

Exemple : La table « RESULTAT » répertorie les notes obtenues par des étudiants sur différentes épreuves.

RESULTAT
<u>NumEtudiant</u> <u>NomEpreuve</u> Note

NumEtudiant [PK]	NomEpreuve [PK]	Note
001	Économie	15
001	Droit	17
001	Management	9
002	Économie	12
002	Comptabilité	16

[PK] = Primary Key

Si on rajoute le coefficient de chaque épreuve :

NumEtudiant [PK]	NomEpreuve [PK]	Coefficient	Note
001	Économie	1	15
001	Droit	1	17
001	Management	2	9
002	Économie	1	12
002	Comptabilité	2	16

Le coefficient d'une épreuve ne dépend que de l'épreuve et ne dépend pas de l'étudiant.

⇒ L'attribut « Coefficient » ne dépend donc que d'une partie de la clé primaire (NomEpreuve) et non de la clé primaire complète (NumEtudiant, NomEpreuve) = La 2FN n'est donc pas respectée.

✓ Pour corriger le problème il faut donc à isoler les attributs concernés (ici l'attribut Coefficient) dans des tables dédiées. On va donc créer une table EPREUVE et déplacer la colonne « Coefficient » dans cette nouvelle table :

(...)

**>> Pour télécharger l'intégralité des fiches
UE8 clique [ICI](#)**