Normalisation

La **normalisation** est un concept clé en conception de bases de données relationnelles. Elle sert à organiser les données pour éviter la **redondance** (duplication inutile des données) et assurer la **cohérence** (éviter les erreurs dues à des données incohérentes). Elle repose sur des **formes normales**, qui sont des règles ou étapes progressives pour structurer les données.

---

1. Pourquoi normaliser ?

Problèmes sans normalisation :

1. **Redondance :** Si une information est répétée plusieurs fois dans différentes lignes, toute modification ou suppression peut causer des erreurs.
    - Exemple : Si un client change d'adresse et que l'adresse est stockée dans plusieurs endroits, il faut la mettre à jour partout.
2. **Incohérence :** Si certaines données sont modifiées mais pas d'autres, des erreurs peuvent survenir.
    - Exemple : Deux lignes pour un même client avec deux adresses différentes.
3. **Espace gaspillé :** La répétition des mêmes informations consomme inutilement de la mémoire.

Solution : Normalisation

- Diviser les données en plusieurs tables liées par des **relations** (via des clés primaires et étrangères).
- Organiser les données en suivant les règles des **formes normales**.

---

2. Les principales formes normales

Première forme normale (1NF)

Règle : Toutes les colonnes d'une table doivent contenir des **valeurs atomiques** (non décomposables). Chaque colonne doit représenter une seule valeur, pas une liste ou un ensemble.

Exemple d'une table non normalisée :

ID_Client	Nom	Téléphones
1	Alice	06 12 34 56 78, 06 98 76 54 32
2	Bob	06 22 33 44 55

- **Problème :** La colonne "Téléphones" contient plusieurs valeurs dans une seule cellule.

Solution : Découper en deux tables :

1. Table `Clients` :
ID_Client	Nom
1	Alice
2	Bob
2. Table `Téléphones` :
ID_Téléphone	ID_Client	Téléphone
1	1	06 12 34 56 78
2	1	06 98 76 54 32
3	2	06 22 33 44 55

- Chaque téléphone est stocké dans une ligne distincte.

Deuxième forme normale (2NF)

Règle : Toute colonne non clé doit dépendre entièrement de la clé primaire. Cela évite les dépendances partielles.

Exemple d'une table non normalisée :

ID_Commande	ID_Client	Nom_Client	Date_Commande
1	1	Alice	2024-01-01
2	2	Bob	2024-01-02

- **Problème :** La colonne `Nom_Client` dépend de `ID_Client`, mais pas de la clé primaire complète `ID_Commande`.

Solution : Découper en deux tables :

1. Table `Clients` :
ID_Client	Nom_Client
1	Alice
2	Bob
2. Table `Commandes` :
ID_Commande	ID_Client	Date_Commande
1	1	2024-01-01
2	2	2024-01-02

- Les informations sur les clients sont séparées des commandes.

Troisième forme normale (3NF)

Règle : Une table ne doit pas contenir de **dépendances transitives**. Autrement dit, aucune colonne ne doit dépendre indirectement de la clé primaire.

Exemple d'une table non normalisée :

ID_Client	Nom_Client	Ville	Code_Postal
1	Alice	Paris	75000
2	Bob	Lyon	69000

- **Problème :** La colonne `Code_Postal` dépend de `Ville`, et non directement de `ID_Client`.

Solution : Découper en trois tables :

1. Table `Clients` :
ID_Client	Nom_Client	Ville
1	Alice	Paris
2	Bob	Lyon
2. Table `Villes` :
Ville	Code_Postal

| Paris | 75000 |
| Lyon | 69000 |

- Cela évite la duplication des informations sur les villes et les codes postaux.