

Les familles de Données

Les familles de données

3 grandes familles

Un Système d'Information (SI) est composé d'une multitude croissante de données. Toutes les données n'ont pas la même typologie, le même impact et ne sont pas soumises aux mêmes obligations légales.

Nous pouvons les regrouper en trois grandes familles

Données référentielles

Données **indispensables** aux opérations de gestions du SI et de l'entreprise

Données d'exploitation

Données **issues des opérations** de gestions du SI et de l'entreprise

Données techniques

Données **transparentes pour l'utilisateur** mais centrales au **bon fonctionnement des outils** numériques

Les familles de données

Données Référentielles (ou Données de Référence)

Les données référentielles sont des données qui servent à décrire les éléments fondamentaux de l'entreprise, souvent stables sur le long terme. Elles permettent de donner un cadre aux autres types de données en définissant des éléments communs, standardisés, et centralisés.

La gestion de ces données est si critique qu'elle a conduit à la création d'une discipline spécifique : le Master Data Management (**MDM**), qui garantit leur pérennité, unicité, intégrité, et fiabilité.

Exemples :

- **Produits** : *Liste des produits que l'entreprise commercialise, avec des informations comme le nom, la référence, le prix de base.*
- **Clients** : *Données de base sur les clients comme leur nom, adresse, numéro de téléphone, identifiant unique.*

Imaginez un client enregistré dans le SI depuis des années, avec des factures associées. Si la donnée du client est modifiée ou supprimée sans précaution, cela pourrait gravement perturber le système. C'est pourquoi il est crucial de maintenir l'intégrité de ces données.

Les familles de données

Données d'Exploitation (ou Données Transactionnelles)

Les données d'exploitation sont des données générées par les opérations quotidiennes de l'entreprise. Elles reflètent les transactions, les événements et les activités qui se produisent au sein de l'organisation. Ces données sont souvent volumineuses et dynamiques, car elles évoluent rapidement avec l'activité de l'entreprise.

Exemples :

- **Commandes** : *Informations sur les commandes passées par les clients, comme la date, les produits commandés, la quantité, et le montant total.*
- **Factures** : *Détails sur les factures émises, comprenant les montants dus, les dates d'échéance, et les paiements reçus.*
- **Transactions de vente** : *Enregistrements de chaque vente effectuée, y compris le produit vendu, le prix, et le client.*
- **Support client** : *Tickets ou incidents signalés par les clients, avec des informations sur la résolution du problème.*

Les familles de données

Données Techniques

Les **données techniques** concernent les informations liées à l'infrastructure technologique de l'entreprise. Elles sont souvent utilisées pour la gestion des systèmes informatiques, la sécurité, et l'optimisation des performances.

Exemples :

- **Configurations des serveurs** : Informations sur les paramètres des serveurs, comme les adresses IP, les versions des logiciels installés, et les configurations réseau.
- **Logs systèmes** : Journaux d'événements qui enregistrent les actions effectuées sur les systèmes informatiques, comme les connexions d'utilisateurs, les erreurs, ou les activités système.
- **Schémas de base de données** : La structure des bases de données, comprenant les tables, les relations, les index, etc.
- **Scripts et automatisations** : Scripts utilisés pour automatiser des tâches, comme des backups réguliers, ou des tâches de maintenance.
- **Résumé** :
- **Données Référentielles** : Décrivent les entités stables de l'entreprise, comme les produits ou les clients.
- **Données d'Exploitation** : Représentent les opérations quotidiennes de l'entreprise, comme les ventes ou les commandes.
- **Données Techniques** : Concernent les aspects technologiques et informatiques, comme les configurations système ou les logs.

Chacune de ces familles de données joue un rôle crucial dans la gestion efficace de l'information au sein de l'entreprise.

Les familles de données

Métadonnées :

Les métadonnées sont des données qui décrivent d'autres données. Elles fournissent des informations contextuelles, comme le type de donnée, le format, la date de création, l'auteur, ou les droits d'accès. Les métadonnées permettent de comprendre, classer, trouver, ou gérer d'autres données.

Relation entre Données Techniques et Métadonnées :

Les métadonnées sont souvent considérées comme un sous-ensemble des données techniques, mais elles ne se limitent pas uniquement à cela.

Exemple :

Un fichier log (donnée technique) pourrait contenir des métadonnées comme la date de création du log ou l'utilisateur ayant généré le log.

Un document stocké dans un système de gestion de contenu aurait des métadonnées comme le titre, l'auteur, la date de modification, et le type de document.

Résumé :

Données Techniques : Concernent les aspects liés à l'infrastructure informatique et au fonctionnement des systèmes.

Métadonnées : Sont des informations descriptives qui aident à identifier, comprendre et gérer d'autres données.

En somme, les données référentielles et les données d'exploitation diffèrent par leur nature (stable vs. dynamique) et leur rôle (référence vs. transaction). Les données techniques incluent divers types de données, dont les métadonnées font partie.

Les bases de Données

Les bases de données relationnelles

Qu'est-ce qu'une base de données relationnelle ?

Une base de données relationnelle est un outil qui permet de collecter et d'organiser des informations dans des tableaux à deux dimensions appelés des relations ou tables.

Selon ce modèle *relationnel*, une base de données consiste en une ou plusieurs relations. Les lignes de ces relations sont appelées des *nuplets* ou enregistrements. Les colonnes sont appelées des *attributs*.

Certains principes guident le processus de conception de base de données. Le premier principe est que les informations en double (également appelées données redondantes) sont incorrectes, car elles gaspillent de l'espace et augmentent la probabilité d'erreurs et d'incohérences.

Une base de données relationnelle est le type de base de données le plus courant. Il utilise un schéma, un modèle, pour dicter la structure de données stockées dans la base de données.

Par exemple, une entreprise qui vend des produits à ses clients doit disposer d'une certaine forme de connaissance stockée de l'endroit où ces produits vont, à qui et en quelle quantité.

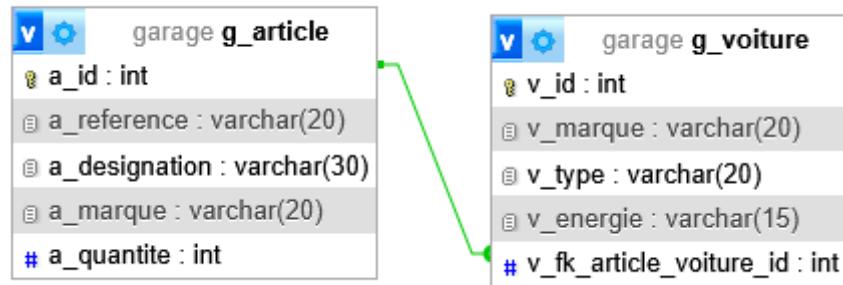
Il peut y avoir différents types de bases de données relationnelles utilisées pour chaque approche. Par exemple, la première table peut afficher des informations essentielles sur le client, la seconde le nombre de produits vendus et la troisième énumère qui a acheté ce produit et où.

Il existe des clés associées à des tables dans une base de données relationnelle. Ils fournissent un résumé rapide de la base de données ou un accès à la ligne ou à la colonne que vous souhaitez vérifier.

Les bases de données relationnelles

Qu'est-ce qu'un schéma de base de données?

Dans le contexte d'une base de données et de son organisation, le terme « schéma de base de données » décrit la structure des données et les relations entre les éléments qui la composent. Ces informations sont souvent présentées visuellement à l'aide de diagrammes illustrant les connexions entre les différentes tables et éléments, offrant ainsi une vue d'ensemble logique de la base de données.



Voici un exemple de schéma de base de données très simple, comportant deux tables : **article** et **voiture**. Les colonnes de clé primaire sont en gras, les relations de clé étrangère sont représentées avec un suffixe **fk**, et le type de données de chaque colonne est indiqué. Ces exemples de schémas illustrent la manière dont les différents éléments de la base de données sont connectés et interagissent.

Notez que **le schéma n'est pas le diagramme lui-même, le schéma est l'ensemble des règles et des relations qui régissent les données de cette base de données** et qui sont représentées dans le diagramme.

En résumé, un schéma de base de données :

- Définit la structure des données dans une base de données.
- Définit la manière dont les éléments d'une base de données sont liés les uns aux autres.
- Réalise ce qui précède grâce à la mise en œuvre de règles codées

Les bases de données relationnelles

Le processus de conception

Le processus de conception comprend les étapes suivantes :

Déterminez la raison d'être de votre base de données

Cela vous permet de vous préparer pour les étapes restantes.

Recherchez et organisez les informations requises

Rassemblez tous les types d'informations que vous souhaiterez peut-être enregistrer dans la base de données, comme le nom du produit et le numéro de commande.

Diviser les informations en tables

Divisez vos éléments d'information en entités ou sujets principaux, tels que Produits ou Commandes. Chaque sujet devient alors une table.

Transformer des éléments d'informations en colonnes

Déterminez les informations que vous souhaitez stocker dans chaque table. Chaque élément devient un champ et s'affiche sous la forme d'une colonne dans la table. Par exemple, une table Articles peut inclure des champs tels que **Gencod, SKU**.

Les bases de données relationnelles

Le processus de conception (suite)

Spécifier des clés primaires

Choisissez la clé primaire de chaque table. La clé primaire est une colonne utilisée pour identifier chaque ligne de manière unique. Par exemple, l'ID de produit ou l'ID de commande.

Configurer les relations de table

Examinez chaque table et déterminez comment les données d'une table sont liées aux données des autres tables. Ajoutez des champs à des tables ou créez des tables pour clarifier les relations, si nécessaire.

Affiner votre conception

Analysez votre conception à la recherche d'erreurs. Créez les tables et ajoutez quelques enregistrements d'exemples de données. Vérifiez si vous pouvez obtenir les résultats souhaités à partir de vos tables. Apportez des ajustements à la conception, selon vos besoins.

Appliquer les règles de normalisation

Appliquez les règles de normalisation des données pour voir si vos tables sont correctement structurées. Ce processus comprend la création de tables et l'établissement de relations entre celles-ci conformément à des règles conçues à la fois pour protéger les données et pour rendre la base de données plus flexible grâce à l'élimination de la redondance et des dépendances incohérentes.

Le moteur de base de données

Les **transactions** sont des paquets de données de plusieurs opérations indissociables.

Lorsqu'une opération ne peut pas être exécutée, l'ensemble de la transaction échoue. Pour y remédier, il faut non seulement annuler l'opération erronée, mais aussi toutes les opérations précédentes. Ce processus demande plus de temps que de simples opérations, mais c'est le seul moyen de garantir la cohérence de la base de données.

Le moteur de base de données est le composant central d'un système de gestion de base de données. C'est un composant essentiel des systèmes d'informations, ainsi que de nombreux logiciels qui manipulent des grandes quantités de données.

Un moteur de base de données enregistre et puise les données dans un ou plusieurs fichiers en suivant un ensemble de règles prédéfinies enregistrées dans la base de données. Ces règles déterminent le schéma conceptuel de la base de données, les règles de cohérence et de confidentialité. Le moteur de base de données refusera toute opération qui n'est pas conforme aux règles.

Les SGBDR, qui utilisent SQL, doivent présenter quatre propriétés, connues sous l'acronyme **ACID**. Celles-ci garantissent le bon déroulement des transactions et la haute fiabilité de la base de données SQL :

- **Atomicité** : Toutes les transactions doivent réussir ou échouer complètement et ne peuvent pas rester partiellement terminées, même en cas de panne du système.
- **Cohérence** : La base de données doit respecter des règles qui valident et empêchent la corruption à chaque étape.
- **Isolation** : Les transactions simultanées ne peuvent pas s'affecter mutuellement.
- **Durabilité** : Les transactions sont définitives et même une panne du système ne peut pas annuler une

Les bases de données

Le moteur de base de données

Contrôle des transactions : lors d'une transaction plusieurs modifications sur la base de données correspondent à une seule opération. Le moteur assure la cohérence du contenu de la base de données, y compris en cas d'échec ou de panne. Le moteur vérifie que les modifications concurrentes des mêmes informations n'aboutissent pas à un résultat incohérent.

sécurité : le moteur vérifie qu'aucun utilisateur n'accède à des informations non autorisées, et qu'aucun utilisateur n'effectue des modifications qui seraient contraires aux règles de cohérence.

accès aux fichiers : le moteur manipule l'espace réservé au stockage. Les informations sont groupées par nature, et chaque fichier stocke une collection d'informations de même nature. Le programme d'accès au fichier structure les différents fichiers conformément au schéma d'organisation de la base de données.

MyISAM :

Il s'agit d'un moteur non transactionnel assez rapide en écriture et très rapide en lecture. Ceci vient en grande partie du fait qu'il ne gère pas les relations ni les transactions et évite donc des contrôles gourmands en ressources mais perd en sûreté ce qu'il gagne en vitesse.

InnoDB :

Permet d'exécuter des transactions (**type ACID**) de données isolées les unes des autres : c'est uniquement lorsqu'une **transaction** est terminée qu'elle est inscrite sur le support de données correspondant. Cette procédure évite tout accès incomplet dans une base de données existante.

Comprendre l'intégrité référentielle

Lorsque vous créez une base de données, vous répartissez vos informations de base de données entre plusieurs tables en fonction de leurs sujets pour réduire la redondance des données.

Pour que MySQL puisse rassembler les données des tables associées, il faut placer des champs communs entre ces tables.

Pour représenter une relation un-à-plusieurs (1-N), vous prenez la clé primaire de la table « une », et vous l'ajoutez comme champ supplémentaire (clé étrangère) à la table « plusieurs ».

Pour rassembler les données, MySQL prend la valeur de la table « plusieurs » et recherche la valeur correspondante dans la table « un ». Ainsi, les valeurs de la table « plusieurs » font référence aux valeurs correspondantes dans la table « une ».

Exemple :

Imaginons que vous ayez une relation un-à-plusieurs entre les tables Expéditeurs et Commandes et que vous souhaitiez supprimer un expéditeur. Si l'expéditeur à supprimer présente des commandes dans la table Commandes, celles-ci deviennent « orphelines » dès lors que vous supprimez l'enregistrement de l'expéditeur. Cela dit, la référence de l'expéditeur figurera toujours sur les commandes, mais elle ne sera plus valide, car l'enregistrement à laquelle elle fait référence n'existe plus.

L'intégrité référentielle consiste à empêcher les orphelins et à assurer la synchronisation des références pour éviter que cette situation hypothétique ne se présente.