



WEBFORCE
BE THE CHANGE



RÉSUMÉ THÉORIQUE – FILIÈRE INTELLIGENCE ARTIFICIELLE

M105 - Appréhender la modélisation des données



60 heures



SOMMAIRE

01 – INTRODUIRE LE CONTEXTE DE LA MODÉLISATION DES DONNÉES

1. Introduire la modélisation des données
2. Introduire le domaine du business intelligence

02 – II. APPREHENDER LE MODELE DIMENSIONNEL

1. Maitriser les faits
2. Maitriser les dimensions
3. Appréhender les dimensions à évolution lente



MODALITÉS PÉDAGOGIQUES



WEBFORCE
BE THE CHANGE



1

LE GUIDE DE SOUTIEN

Il contient le résumé théorique et le manuel des travaux pratiques



2

LA VERSION PDF

Une version PDF est mise en ligne sur l'espace apprenant et formateur de la plateforme WebForce Life



3

DES CONTENUS TÉLÉCHARGEABLES

Les fiches de résumés ou des exercices sont téléchargeables sur WebForce Life



4

DU CONTENU INTERACTIF

Vous disposez de contenus interactifs sous forme d'exercices et de cours à utiliser sur WebForce Life



5

DES RESSOURCES EN LIGNES

Les ressources sont consultables en synchrone et en asynchrone pour s'adapter au rythme de l'apprentissage



PARTIE 1

INTRODUIRE LE CONTEXTE DE LA MODÉLISATION DES DONNÉES

Dans ce module, vous allez :

- Introduire la modélisation des données
- Définir le modèle dimensionnel



15 heures

CHAPITRE 1

INTRODUIRE LA MODÉLISATION DES DONNÉES

Ce que vous allez apprendre dans ce chapitre :

- Définir la modélisation des données
- Connaître les type de modélisation des données



05 heures



CHAPITRE 1

INTRODUIRE LA MODÉLISATION DES DONNÉES

1. Définitions

2. Types

- Modèle relationnel
- Modèle entité-relation (ERM)
- Modèle dimensionnel
- Modèle en réseau



1 - INTRODUIRE LA MODÉLISATION DES DONNÉES

Définitions



Modélisation des données

La modélisation des données est le processus de création d'un modèle qui représente la structure, les règles et les relations des données d'un système ou d'une organisation. Il s'agit d'une étape importante dans la conception d'une base de données et dans le développement d'applications logicielles qui utilisent des données.

Le processus de modélisation des données implique la définition des entités, des attributs et des relations entre les entités, ainsi que des règles et des contraintes qui s'appliquent aux données. Le résultat de la modélisation des données est généralement un diagramme qui représente graphiquement la structure des données.

Le modèle de données permet de comprendre comment les données sont stockées, organisées et utilisées dans un système ou une organisation. Il peut être utilisé pour faciliter la communication entre les développeurs, les utilisateurs et les administrateurs de la base de données, et pour garantir la cohérence et la qualité des données.

La modélisation des données est une étape importante dans la conception de systèmes d'information et de bases de données, ainsi que dans la gestion et l'analyse des données. Elle est utilisée dans de nombreux domaines, y compris l'informatique, la gestion de projet, la science des données et l'ingénierie logicielle.

CHAPITRE 1

INTRODUIRE LA MODÉLISATION DES DONNÉES

1. Définitions

2. Types

- Modèle relationnel
- Modèle entité-relation (ERM)
- Modèle dimensionnel
- Modèle en réseau



Vue générale

- Il existe plusieurs types de modèles des données, dont voici les principaux :

Modèle relationnel

- Ce modèle utilise des tables pour stocker les données et des relations entre les tables pour représenter les relations entre les données.

Modèle entité-relation (ERM)

- C'est le type le plus courant de modélisation des données. Il utilise des entités (objets ou concepts) et des relations pour représenter la structure et les relations des données.

Modèle dimensionnel

- c'est un autre modèle couramment utilisé pour la modélisation des Data Warehouse (entrepôts de données). Il utilise des dimensions (caractéristiques) et des mesures (valeurs numériques) pour représenter les données.

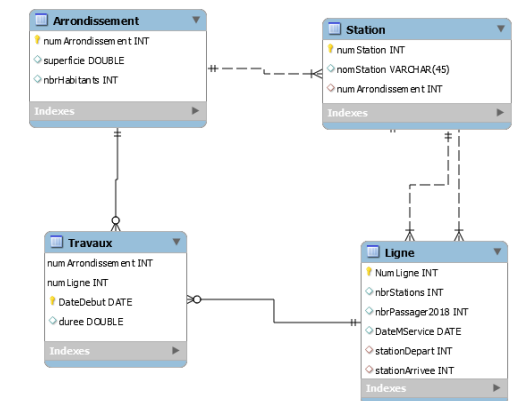
Modèle en réseau

- C'est un modèle qui organise les données sous forme de graphes connectés, où les enregistrements sont liés les uns aux autres par des relations complexes. Dans ce modèle, les données sont représentées par des nœuds et les relations entre les données sont représentées par des arêtes.

- Ces différents types de modélisation des données peuvent être utilisés pour différents types de systèmes et de bases de données, en fonction des besoins spécifiques de chaque projet.

Modèle relationnel

- Le modèle relationnel est une approche de modélisation de données qui est basée sur la théorie des ensembles mathématiques et qui repose sur l'utilisation de tables et des relations pour organiser et structurer les données.
- Dans un modèle relationnel, les données sont représentées sous forme de tables. Chaque table ayant une série de colonnes qui représentent les attributs ou les caractéristiques des données et des lignes qui représentent les enregistrements ou les instances des données. Les tables sont reliées les unes aux autres à travers des clés primaires et étrangères, qui permettent d'établir des relations entre les différentes tables et donc de lier les données entre elles.
- Le modèle relationnel est devenu très populaire dans les années 1970 grâce à la publication des travaux de Edgar F. Codd. Il est aujourd'hui largement utilisé dans la conception de systèmes de gestion de bases de données relationnelles (SGBDR).
- L'avantage du modèle relationnel est qu'il permet de structurer les données de manière logique et cohérente, ce qui facilite leur manipulation et leur accès par les utilisateurs et les applications. De plus, la normalisation des données permet d'assurer l'intégrité et la qualité des données stockées dans la base de données.



Modèle relationnel

- Le modèle relationnel peut parfois être rigide et ne pas convenir à toutes les situations. Par exemple, lorsque les données sont très complexes et nécessitent une modélisation plus flexible, d'autres approches de modélisation peuvent être préférables, comme le modèle objet-relationnel ou le NoSQL.
- Voici quelques exemples de bases de données relationnelles :



Oracle
Database



MySQL



Microsoft SQL
Server



PostgreSQL



IBM DB2

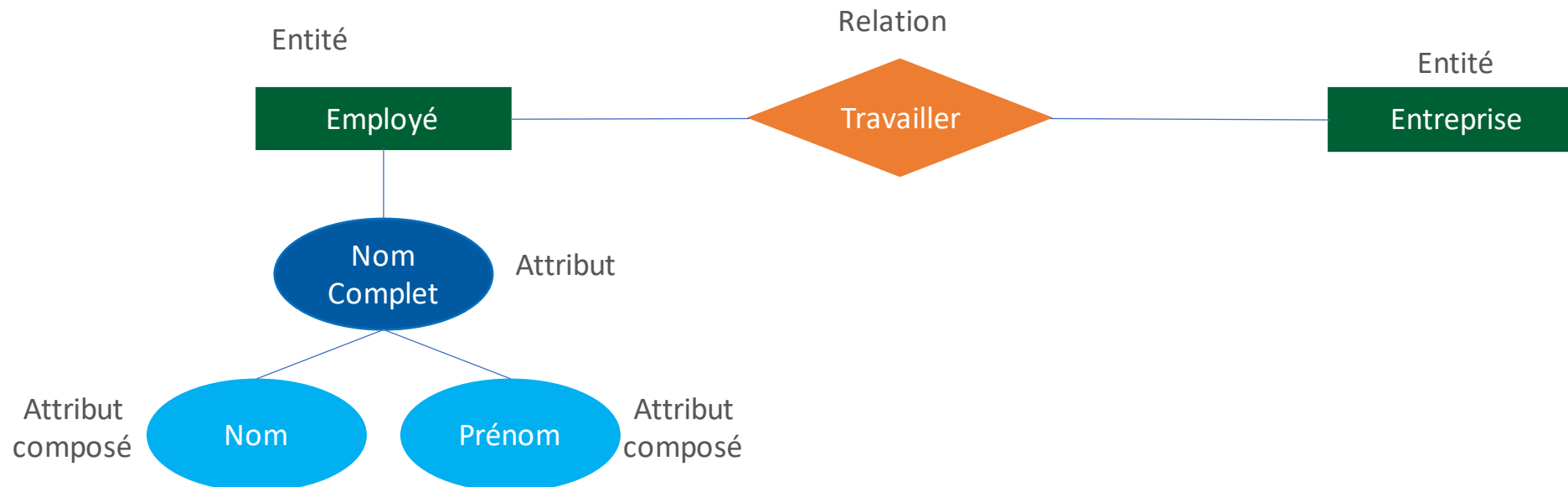


MariaDB

- Ces bases de données sont utilisées dans de nombreux domaines d'application, tels que la gestion de la relation client, la gestion des stocks, la finance, la santé, l'éducation, etc.

Modèle entité-relation (ERM)

- Le modèle entité-relation (ERM) est un modèle de données conceptuel utilisé pour représenter les entités (objets, concepts, personnes, etc.) d'un système ainsi que les relations entre ces entités.
- Il se base sur la notion d'entités, qui sont des objets distincts et identifiables, et les relations qui existent entre ces entités.
- Le modèle ERM utilise des diagrammes entité-relation pour illustrer les entités, leurs attributs (caractéristiques) et les connexions entre elles, permettant de décrire la structure et les règles de gestion des données d'un système de manière graphique et intuitive.



Modèle entité-relation (ERM)

- Voici les principales différences entre le modèle relationnel et l'ERM

Structure des données

- Le modèle entité-relation utilise des entités pour représenter les objets ou les concepts, tandis que le modèle relationnel utilise des tables pour stocker les données.

Relations

- Le modèle entité-relation utilise des relations pour connecter les entités et représenter les associations entre les objets ou les concepts, tandis que le modèle relationnel utilise des clés étrangères pour relier les tables et représenter les relations entre les données.

Abstraction

- Le modèle entité-relation est plus abstrait et peut être utilisé pour représenter des concepts ou des idées plus générales, tandis que le modèle relationnel est plus concret et est utilisé pour stocker des données spécifiques.

Diagrammes

- Le modèle entité-relation est généralement représenté par un diagramme entité-relation, qui montre les entités et les relations, tandis que le modèle relationnel peut être représenté par un schéma relationnel, qui montre les tables et les relations entre les tables.

- En général, le modèle entité-relation est utilisé pour concevoir le schéma conceptuel d'une base de données, tandis que le modèle relationnel est utilisé pour implémenter le schéma logique de la base de données en utilisant des tables et des clés étrangères. Le modèle entité-relation est donc plus abstrait et conceptuel, tandis que le modèle relationnel est plus concret et technique.

Modèle entité-relation (ERM)

- Voici quelques exemples de bases de données ERM populaires:



Oracle
Database



MySQL



Microsoft SQL
Server



PostgreSQL



IBM DB2



Teradata

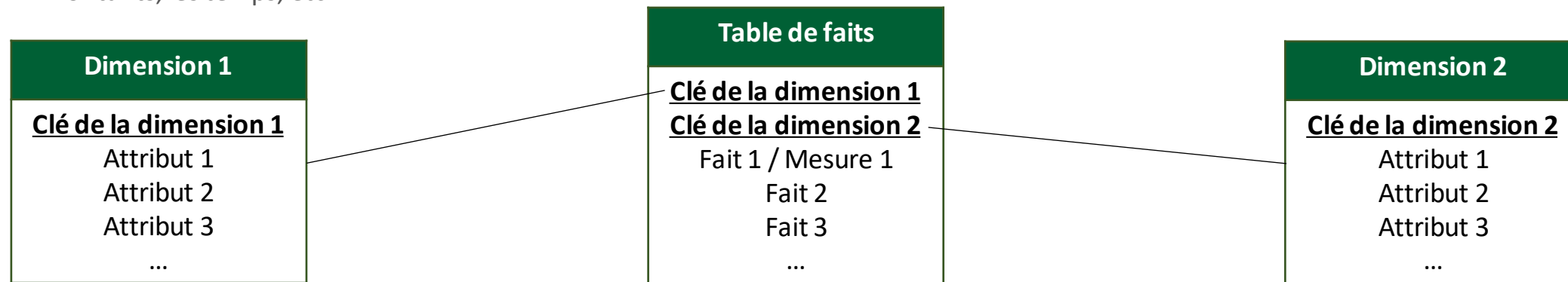


MariaDB

- Ces bases de données sont utilisées dans de nombreux domaines, notamment dans les entreprises, les organisations gouvernementales et les sites web pour stocker des données structurées.

Modèle dimensionnel

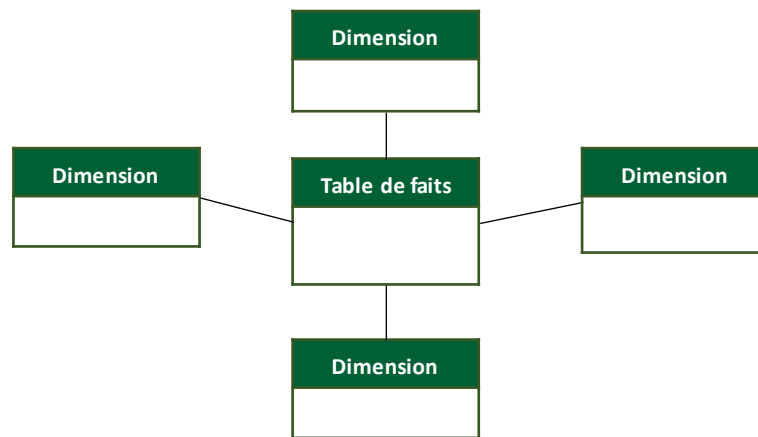
- Un modèle dimensionnel est un modèle de données utilisé en informatique décisionnelle pour organiser les données en fonction de leur signification dans un contexte métier. Il est spécifiquement conçu pour prendre en compte les aspects de temps et de mesure dans les données.
- Le modèle dimensionnel est basé sur deux types de tables : les tables de dimensions et les tables de faits. Les tables de dimensions contiennent les données de référence qui décrivent les caractéristiques des objets métier, tels que des clients, des produits, des emplacements, des temps, etc. Les tables de faits contiennent les mesures numériques qui représentent les performances, les quantités, les montants, les temps, etc.



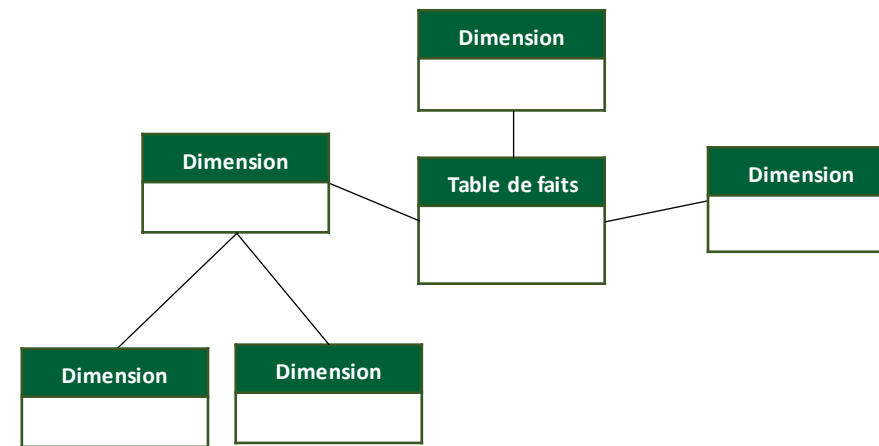
Remarque : On va détailler ce modèle dans ce cours

Modèle dimensionnel

- Le modèle dimensionnel utilise également des schémas en étoile ou en flocon, qui sont des structures de tables organisées autour d'une table centrale de faits, entourée de tables de dimensions. Cette organisation permet d'optimiser les requêtes et les analyses sur les données en minimisant le nombre de jointures nécessaires.
- Le modèle dimensionnel est largement utilisé dans les entrepôts de données et les systèmes décisionnels, car il permet une analyse rapide et facile des données métier. Il est considéré comme une alternative efficace au modèle relationnel traditionnel pour l'analyse de données complexes et volumineuses.



schémas en étoile



schémas en flocon

Remarque : On va détailler ce modèle dans ce cours

Modèle dimensionnel

- Plusieurs exemples de base de données peuvent utiliser le modèle dimensionnel, à savoir :

Les ventes

- Cette base de données comprend des informations sur les ventes de produits ou de services, ainsi que sur les clients qui les ont achetés. Les dimensions pourraient inclure les clients, les produits, les emplacements, les dates et les promotions. Les faits/mesures pourraient inclure les ventes brutes, les remises, les coûts et les marges bénéficiaires.

Les ressources humaines

- Cette base de données comprend des informations sur les employés et leur performance. Les dimensions pourraient inclure les employés, les postes, les départements, les emplacements et les périodes. Les mesures pourraient inclure le salaire, les heures travaillées, les absences et les performances.

Les finances

- Cette base de données comprend des informations sur les transactions financières, les comptes et les portefeuilles. Les dimensions pourraient inclure les clients, les comptes, les dates, les devises et les types de transactions. Les mesures pourraient inclure les soldes, les revenus, les dépenses, les bénéfices et les pertes.

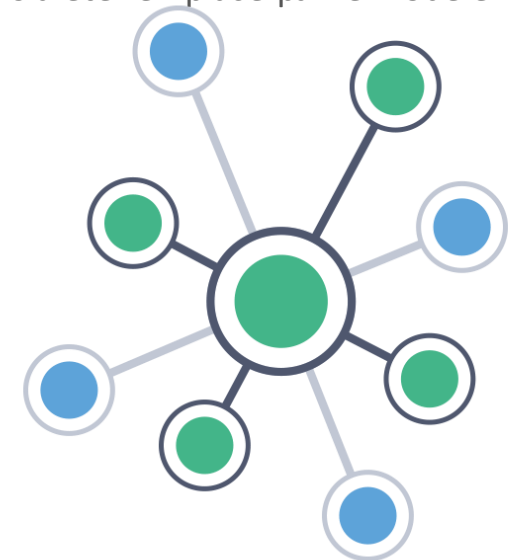
Les voyages

- Cette base de données comprend des informations sur les voyages, les destinations et les réservations. Les dimensions pourraient inclure les clients, les destinations, les hôtels, les compagnies aériennes et les dates. Les mesures pourraient inclure les coûts, les réservations, les annulations et les revenus.

- Ces exemples sont tous des bases de données dimensionnelles car elles sont organisées autour de dimensions clés et de mesures numériques pour permettre une analyse facile et efficace.

Modèle en réseau

- Le modèle en réseau est un modèle de données hiérarchique qui a été développé dans les années 1960. Dans ce modèle, les données sont représentées comme des enregistrements reliés entre eux par des relations hiérarchiques. Les enregistrements sont organisés en ensembles appelés "nœuds", et chaque nœud est connecté à d'autres nœuds par des "liens". Les liens peuvent être de plusieurs types, tels que les "liens propriétaires" qui relient un nœud à ses enregistrements associés, ou les "liens non-propriétaires" qui relient un nœud à d'autres nœuds.
- Contrairement au modèle relationnel, qui utilise des tables pour stocker les données, le modèle en réseau utilise des enregistrements interconnectés pour représenter les données. Le modèle en réseau était populaire dans les années 1970, mais a été remplacé par le modèle relationnel dans les années 1980.
- Le modèle en réseau est toujours utilisé dans certains systèmes de gestion de bases de données, tels que les bases de données orientées graphes. Cependant, ces systèmes ont évolué pour prendre en charge des fonctionnalités supplémentaires qui n'étaient pas possibles dans le modèle en réseau traditionnel.



Modèle en réseau

- Il peut encore être utilisé dans certaines situations spécifiques, telles que :

Stockage et analyse de données hiérarchiques

- Il est bien adapté à la modélisation de données hiérarchiques, où les éléments sont organisés en une structure d'arbre.

Recherche de chemin

- Il est également utile pour les applications qui impliquent la recherche de chemins entre les enregistrements de données. Les bases de données basées sur le modèle en réseau permettent de naviguer facilement entre les enregistrements en suivant les relations.

Stockage d'informations complexes

- Il peut être utilisé pour stocker des informations complexes telles que des schémas de données et des catalogues.

Modélisation des réseaux sociaux

- Il est également utilisé dans les bases de données de réseaux sociaux pour stocker des informations sur les relations entre les individus et les groupes.

- Cependant, le modèle relationnel est généralement plus populaire et plus adapté aux besoins de la plupart des applications de base de données modernes en raison de sa simplicité, de sa flexibilité et de sa capacité à gérer efficacement les données relationnelles.

CHAPITRE 2

INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Ce que vous allez apprendre dans ce chapitre :

- Introduire l'informatique décisionnelle
- Présenter un Data Warehouse
- Connaître l'architecture d'un Data Warehouse
- Savoir les différents types des bases de données
- Avoir une idée générale sur l'ODS (Operational Data Storage)
- Introduire le modèle dimensionnel



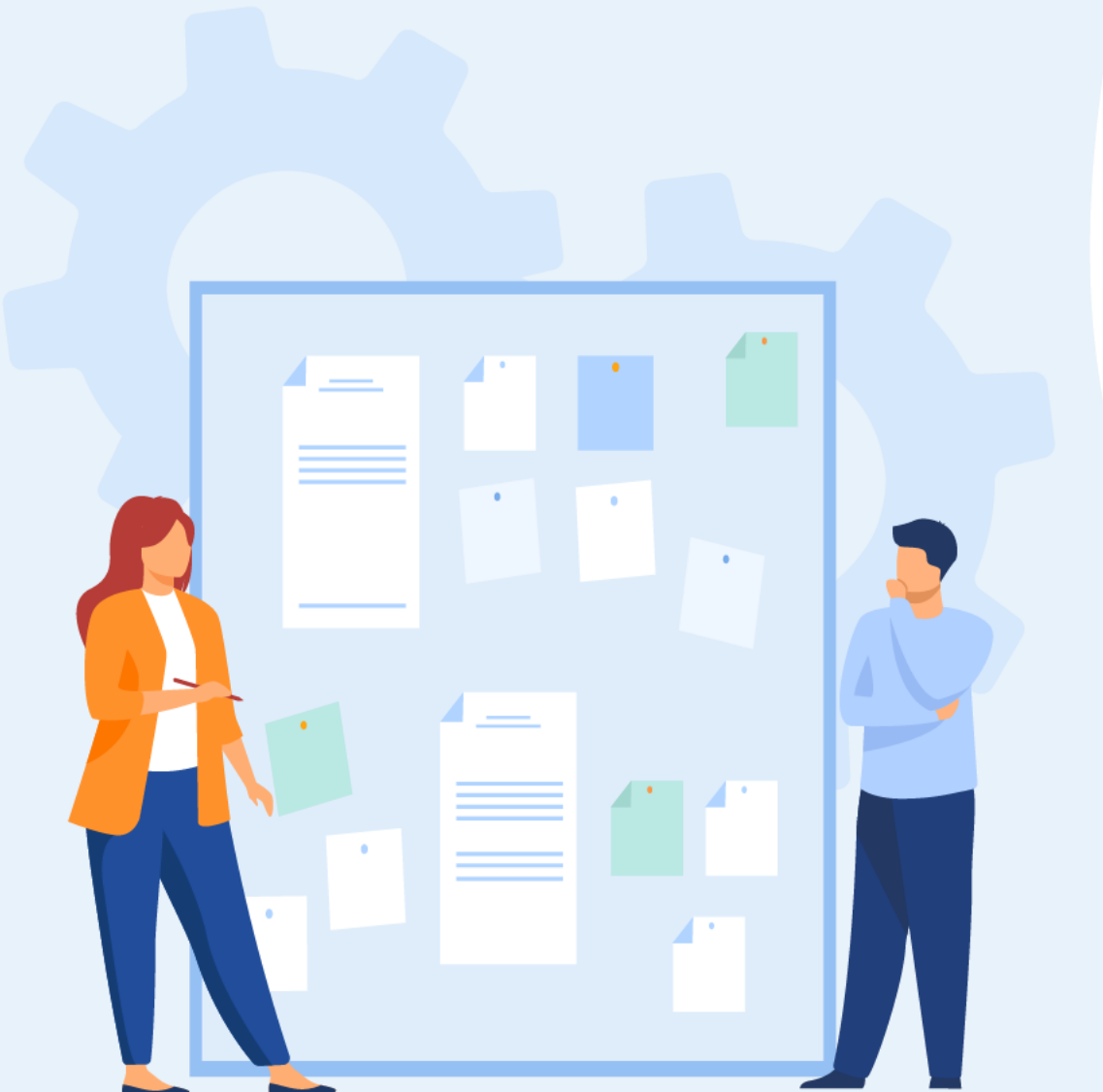
10 heures



CHAPITRE 2

INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

1. **Introduction à l'informatique décisionnelle**
2. Présentation générale d'un Data Warehouse
3. Architecture d'un Data Warehouse
4. Types des bases de données
5. Data Warehouse vs ODS (Operational Data Storage)
6. Introduction au Modèle dimensionnel



2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Introduction à l'informatique décisionnelle



Notions de base

- L'existence de différentes bases de données du même type et de différents types, ainsi que la présence de plusieurs progiciels intégrés a multiplié la quantité de données à traiter au sein des entités.
- L'analyse de cette grande quantité de données est une phase incontournable dans le processus de la gestion des projets décisionnels.
- La gestion et l'analyse des données massives (big data), afin d'extraire des informations utiles dans la prise de décision, n'est pas une tâche évidente. C'est pourquoi, plusieurs entreprises en quête de croissance s'appuient de plus en plus sur les outils de l'informatique décisionnelle.
- L'instauration d'un **Système d'Information Décisionnel (SID)** semble être comme un défi majeur à relever pour les entreprises.
- **Qu'est-ce qu'un système d'information décisionnel ?**



Un Système d'Information Décisionnel (SID)

- **Un Système d'Information Décisionnel (SID)** est un système qui repose sur l'informatique décisionnel (ou Business Intelligence) et est adressé aux responsables des entreprises.
- C'est un ensemble des moyens, des outils et des méthodes qui permettent de collecter, consolider, stocker, modéliser, agréger et restituer **les données**, matérielles ou immatérielles, qui peuvent provenir de différentes sources hétérogènes, en vue d'offrir une **aide à la décision**.
- Les sources de données peuvent être des bases de données relationnelles, des fichiers, des services web, etc.. Ces données peuvent être stockées dans **un entrepôt de données (ou Data Warehouse)**.



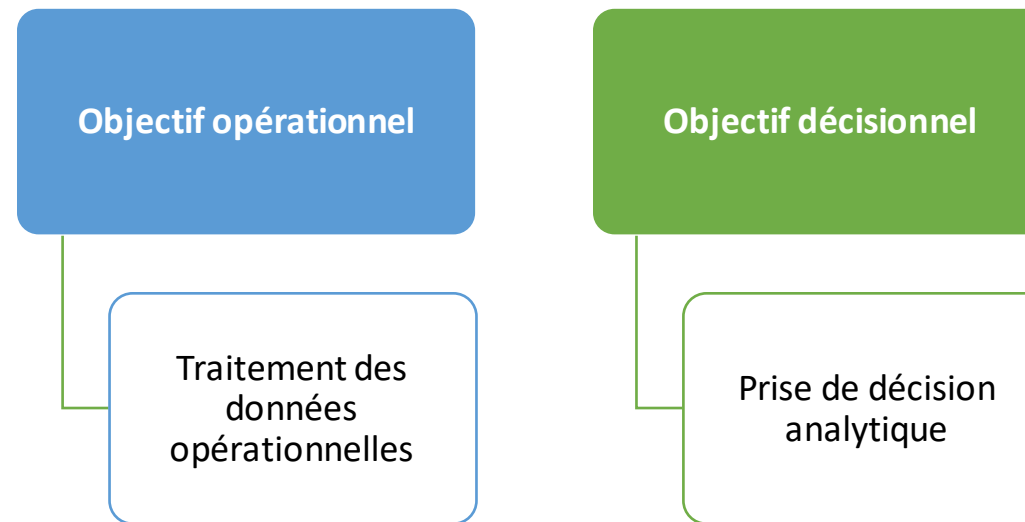
2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Introduction à l'informatique décisionnelle



Objectifs et exigences

L'informatique décisionnelle (ou Business Intelligence) à généralement 2 principaux objectifs :



Objectifs et exigences : Objectif opérationnel

- **L'objectif opérationnel** (Traitement des données opérationnelles) consiste à utiliser les données dans l'objectif de garantir le bon fonctionnement de l'entreprise, par :
 - La réception des requêtes
 - La réaction envers des demandes
 - L'alimentation du stock
- Le sauvegarde/la conservation opérationnel se réfère à l'OLTP (Online Transactional Processing) qui correspond au traitement transactionnel en ligne.
- Dans la partie **traitement des données opérationnelles**, on est amené à :
 - Traiter, généralement, un seul enregistrement à la fois
 - Enregistrer ou modifier des données
 - Se concentrer seulement sur les données actuelles, sans tenir compte de l'historique

Objectifs et exigences : Objectif décisionnel

- **L'objectif décisionnel** (Prise de décision analytique) consiste à utiliser les données dans l'objectif de prendre les bonnes décisions pour le futur et comprendre le fonctionnement de l'entreprise.
- Il consiste à évaluer la performance pour une bonne prise de décision. Réaliser cet objectif revient à répondre à des questions comme :

Quelle est le meilleur produit vendue par l'entreprise ?

Combien de ventes a réalisé l'entreprise cette année par rapport à l'année dernière ?

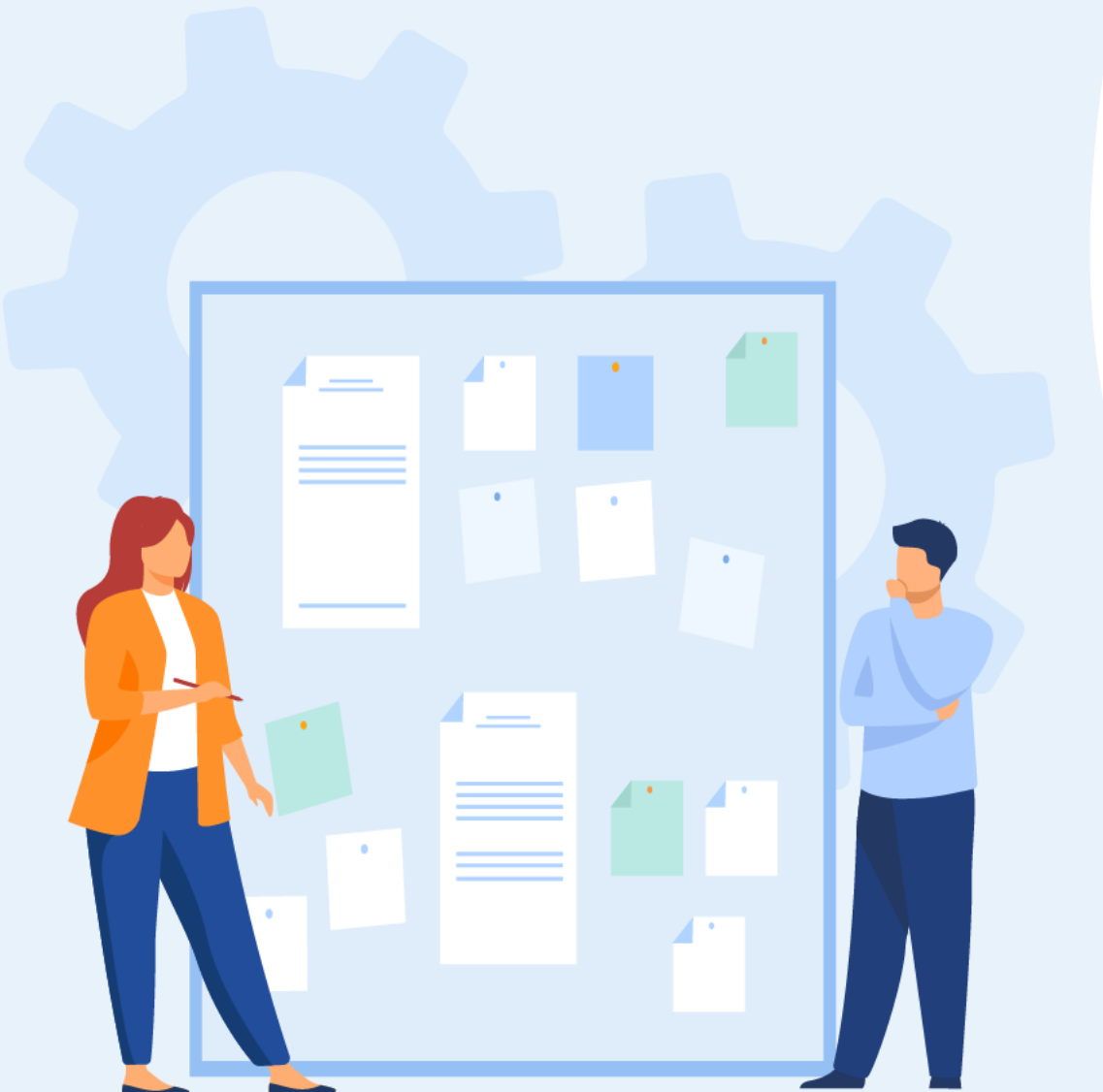
Comment l'entreprise peut-elle évoluer ?

- Le traitement analytique se réfère à l'OLAP (Online Analytical Processing) qui correspond à l'analyse analytique en ligne.
- Dans la partie **prise de décision analytique**, on est censé à :
 - Analyser et récupérer des millions d'enregistrements en même temps
 - Manipuler des requêtes rapides
 - Donner un sens/un contexte aux données, en les analysant les données au cours du temps (historique) en différents contextes
- Le Data Warehouse vient pour s'adresser aux besoins en données analytiques. Le Data Warehouse est donc un entrepôt de données utilisé pour le reporting (génération des rapports) et l'analyse des données.

CHAPITRE 2

INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

1. Introduction à l'informatique décisionnelle
- 2. Présentation générale d'un Data Warehouse**
3. Architecture d'un Data Warehouse
4. Types des bases de données
5. Data Warehouse vs ODS (Operational Data Storage)
6. Introduction au Modèle dimensionnel

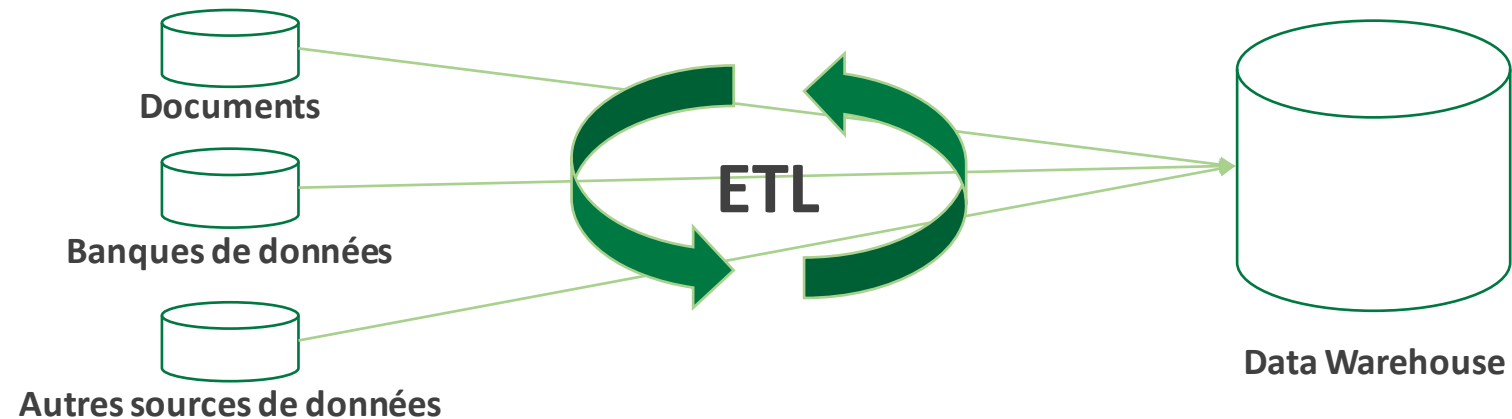


2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Présentation générale d'un Data Warehouse

Data Warehouse

- Un Data Warehouse est une base de données optimisée et utilisée pour les objectifs analytiques. Il est caractérisé par un ensemble de points :
 - **Facile à utiliser** : elle doit être très simple à comprendre et à manipuler par les utilisateurs afin de pouvoir analyser les données.
 - **Performante dans les requêtes rapides** : Récupérer et traiter une grande quantité de données très rapidement.
 - **Analyse de données optimale et facile.**
- Un Data Warehouse est composé de trois composants :



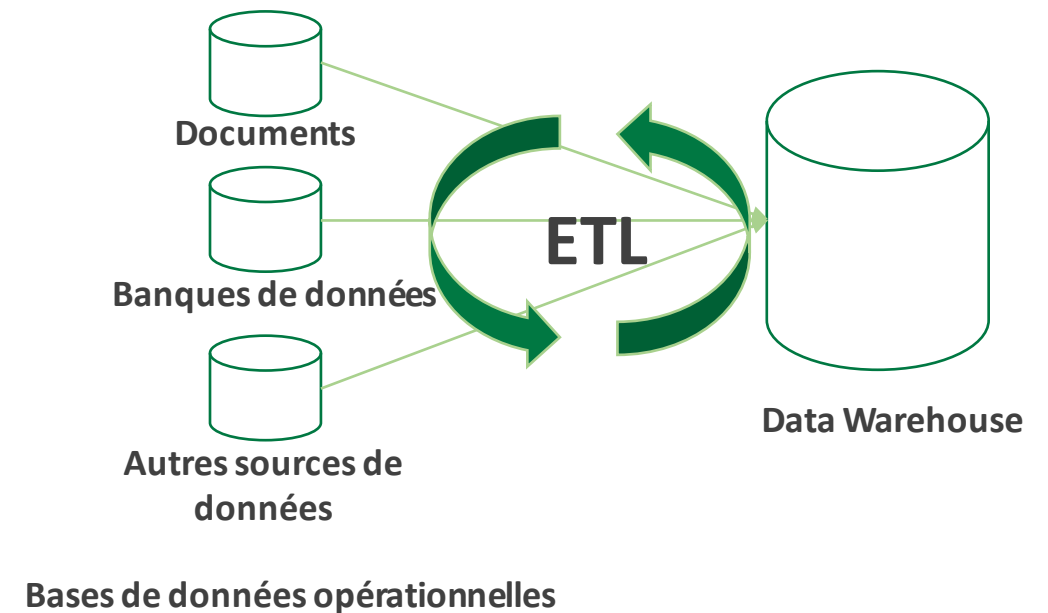
Bases de données opérationnelles

2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Présentation générale d'un Data Warehouse

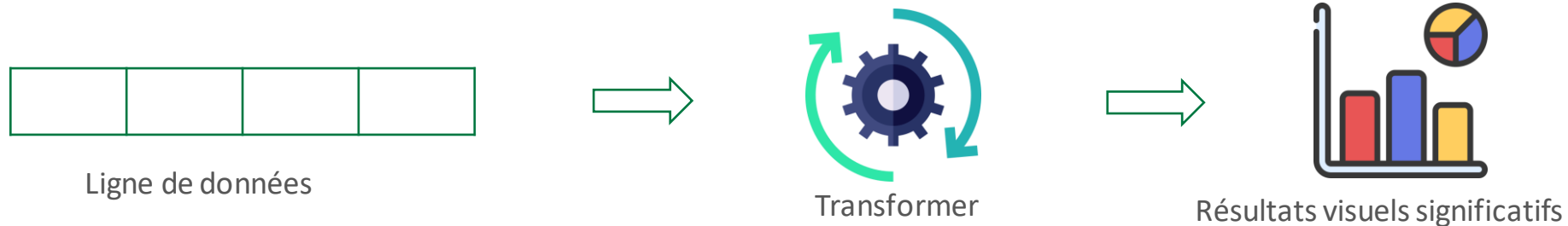
Data Warehouse

- Les bases de données opérationnelles représentent les différentes sources de données ayant différents formats et structures. Toutes ces sources sont regroupées dans un local centralisé qui est le Data Warehouse.
- Le processus de regroupement et d'enregistrement des différentes sources de données est appelé ETL (Extract Transform, Load) ou Extraire, transformer et charger :
 - L'extraction consiste à extraire les données des différentes sources existants.
 - La transformation consiste à transformer les données de telle sorte elles auront la même structure, pour pouvoir les traiter.
 - Le chargement consiste à charger ces données résultantes dans le Data Warehouse.



Business Intelligence

- Un Business Intelligence (BI) est l'ensemble des stratégies, procédures, technologies et infrastructures qui permettent de donner un sens aux données analysées.
- Pour analyser les données il faut les collecter, les gérer et les sauvegarder afin de créer des rapports (reporting), visualiser les données (Data visualization), explorer les données (data mining) ou fournir des analyses prédictives (predictive analytics)
- En général, le BI permet de trouver une ligne de données, transformer cette dernière pour avoir des résultats visuels significatifs, afin de bien comprendre le contexte et prendre les bonnes décisions dans le futur.



- Le Data Warehouse est un composant très important dans le processus du BI, puisqu'il est l'entrepôt de toutes les données nécessaires qui sont déjà transformées et structurées.

Data Lake

- Comme le Data Warehouse, un data lake est aussi une base de données centralisée, mais il ne peut pas remplacer le Data Warehouse, car ils sont différents dans les points suivants :

Data lake	Data Warehouse
On stocke une ligne de données comme elle est sans traitement	On stocke les données traitées via le processus ETL
Les données sont différentes (fichiers CSV et JSON, images, vidéos, etc.) et leur volume est très grand (Big Data). On utilise différentes technologies	Les données ont la même structure
Les données ne sont pas structurées	Les données sont structurées
Les cas d'utilisation ne sont pas prédéfinis au préalable	Le but est déjà défini et le Data Warehouse est prêt à être utilisé
Il est utilisé par les data scientists	Il est utilisé par les BI et IT
La qualité des données n'est pas assurée	La qualité des données est assurée

- Un Data Warehouse et un data lake peuvent exister tous les deux dans un même système.

CHAPITRE 2

INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

1. Introduction à l'informatique décisionnelle
2. Présentation générale d'un Data Warehouse
- 3. Architecture d'un Data Warehouse**
4. Types des bases de données
5. Data Warehouse vs ODS (Operational Data Storage)
6. Introduction au Modèle dimensionnel

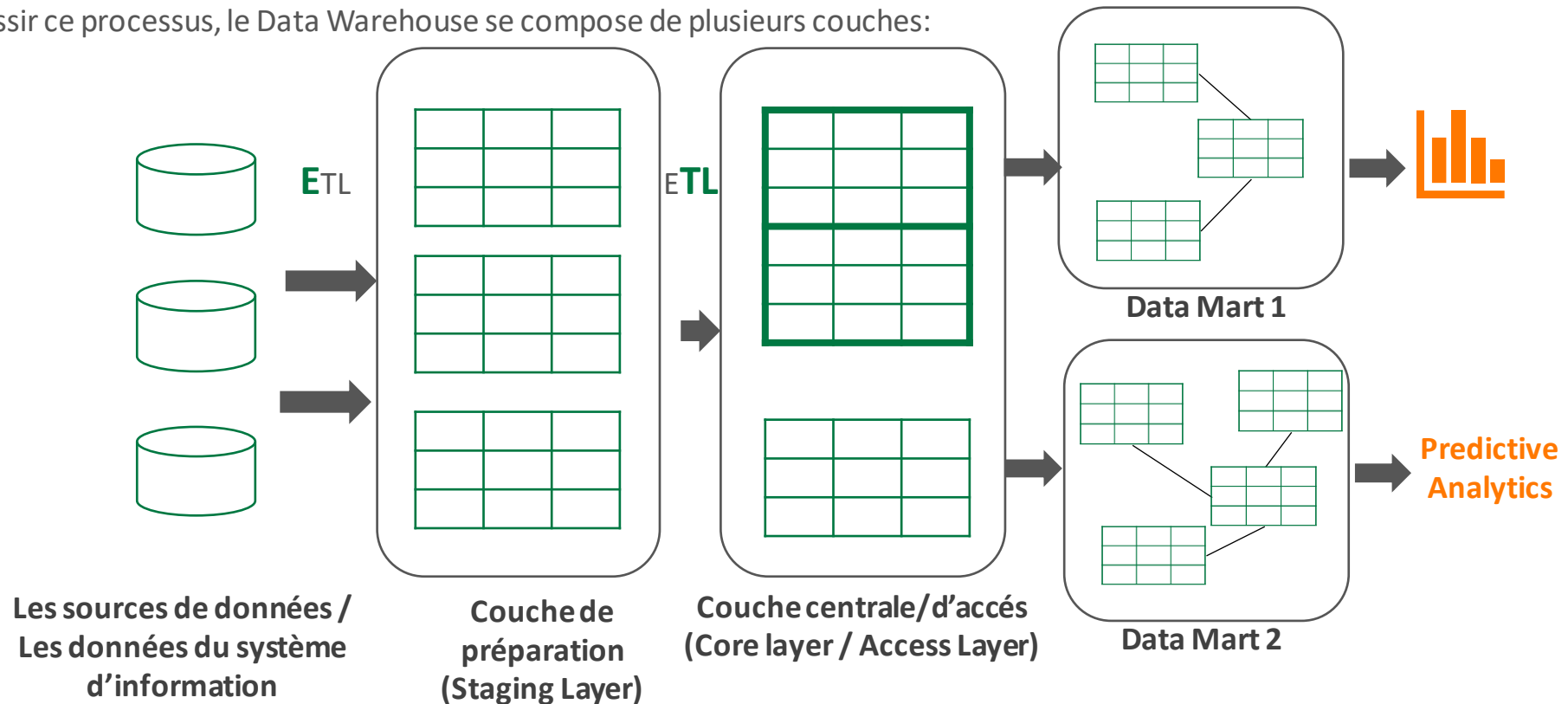


2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Architecture d'un Data Warehouse

Les couches d'un Data Warehouse

- Comme on a vu dans sa définition, on dispose d'un ensemble de sources de données qui sont traitées par le processus ETL et enregistrées dans le Data Warehouse.
- Afin de réussir ce processus, le Data Warehouse se compose de plusieurs couches:

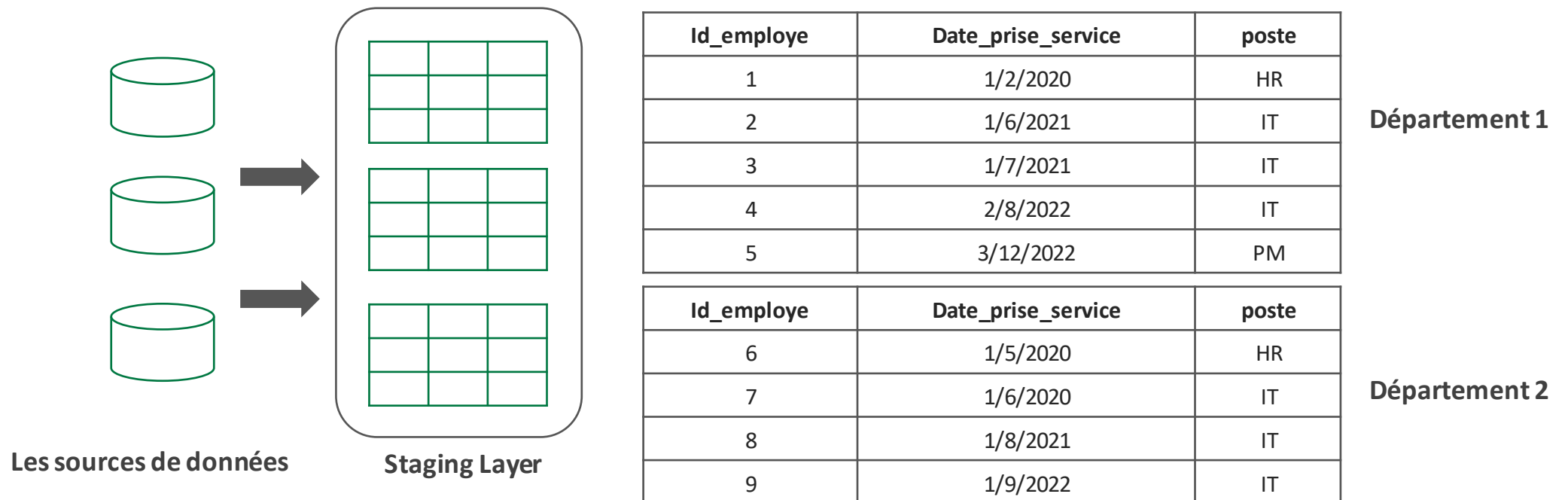


Les couches d'un Data Warehouse : Staging Layer (couche de préparation)

- On utilise le processus ETL pour extraire les données des différentes sources dans la première couche de préparation (Staging Layer).
- Cette couche permet juste d'extraire les tables comme elles sont, sans faire aucune transformation majeure.

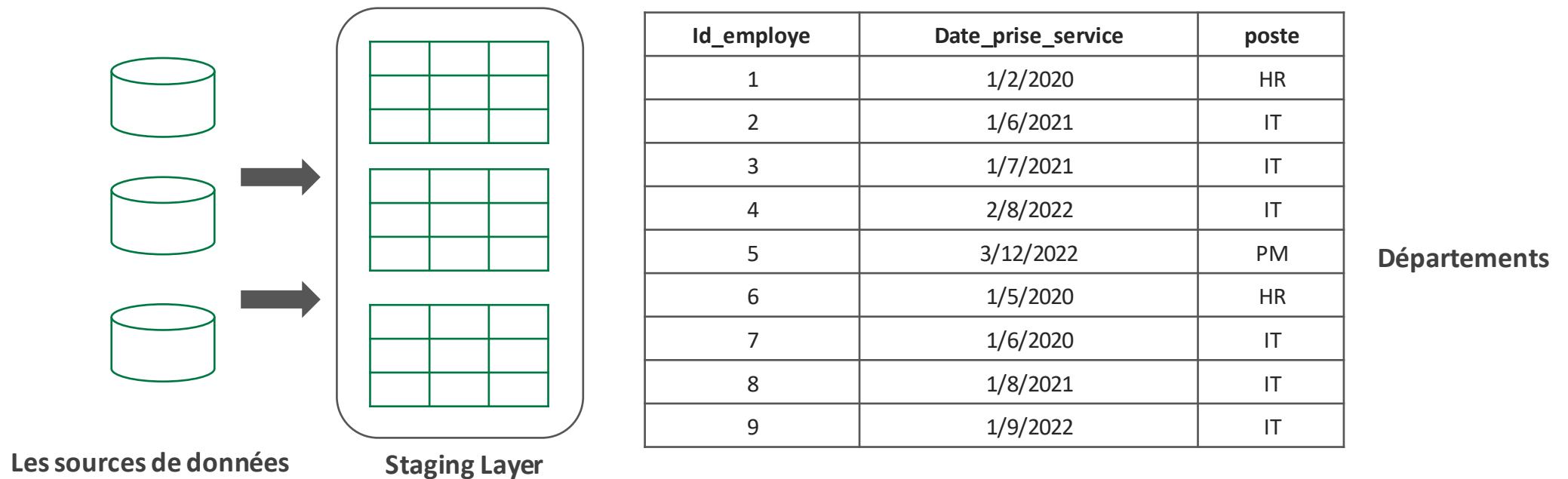
Exemple :

- On extrait les tableaux des employés qui existent dans deux différents départements.



Les couches d'un Data Warehouse : Staging Layer

- Si les tableaux extraits se ressemblent, on peut faire quelques petites transformations, comme la combinaison, tout en adaptant les intitulés des colonnes et des enregistrements si nécessaire.
- Dans certains cas on peut ajouter une autre couche juste après la couche de préparation afin de nettoyer les données, toujours via l'ETL.
- Dans l'exemple précédent, on peut combiner les deux tables en un seul.



Les couches d'un Data Warehouse : Staging Layer

- La question qui se pose par rapport à cette couche : **Pourquoi on utilise la couche de préparation au lieu de traiter les données existants directement à partir de leurs sources?**
- L'application directe des requêtes sur les données des systèmes opérationnels, qui tournent d'une manière permanente, peut causer un ralentissement ou un arrêt du système. Cette couche permet d'interroger les systèmes le moins de temps possible, pour un accès rapide en lecture, une extraction des données et leur enregistrement dans les tables appropriées.
- Les systèmes opérationnels contiennent différents formats de données (bases de données, fichiers JSON, fichiers CSV et d'autres types de données). La couche de préparation permet de déplacer tous ces données dans une base de données relationnelle, de telle sorte à ne voir que des tables.

Les couches d'un Data Warehouse : Staging Layer

- Afin de comprendre le fonctionnement de la couche de préparation, on va prendre un exemple pratique.
- On suppose avoir une table de ventes comme données.
 1. On lit et extrait rapidement les données à partir des systèmes opérationnels
 2. On applique les étapes de transformation et chargement dans le Data Warehouse
 3. On tronque (nettoyage) la couche de préparation après chaque cycle. C'est une couche intermédiaire

id	date	produit
1	1/6/2022	Produit 1
2	1/6/2022	Produit 2
3	1/6/2022	Produit 3
4	1/6/2022	Produit 4
5	1/6/2022	Produit 5

Sources des données



id	date	produit
1	1/6/2022	Produit 1
2	1/6/2022	Produit 2
3	1/6/2022	Produit 3
4	1/6/2022	Produit 4
5	1/6/2022	Produit 5

Lecture est extraction des données
dans la couche de préparation
temporaire

Les couches d'un Data Warehouse : Staging Layer

- On suppose qu'après quelques jours il y aura des données additionnelles dans la table des ventes (enregistrements 6 et 7). On a besoin de savoir quelles sont les nouvelles données ajoutées. C'est pourquoi on doit avoir une **colonne delta** qui permet vérifier si les données sont nouvelles. Elle peut être la colonne **id** si c'est un nombre auto incrémental ou ascendant.
- Dans notre exemple, on sait que le dernier id était 5, donc tous les enregistrements ayant un id supérieur à soit sont nouveaux.
- Dans le cas où les valeurs de l'id ne sont pas ascendantes, on peut utiliser la colonne date comme une colonne delta.
- Une fois l'extraction est faite, on peut appliquer les autres étapes et ajouter les nouvelles valeurs au Data Warehouse.

id	date	produit
1	1/6/2022	Produit 1
2	1/6/2022	Produit 2
3	1/6/2022	Produit 3
4	1/6/2022	Produit 4
5	1/6/2022	Produit 5
6	1/7/2022	Produit 6
7	1/7/2022	Produit 7

Sources des données



id	date	produit
6	1/7/2022	Produit 6
7	1/7/2022	Produit 7

Lecture est extraction des données

Les couches d'un Data Warehouse : Staging Layer

- Les transformations faites sur les données extraites peuvent être problématiques et causer quelques erreurs d'où la nécessité de revenir en arrière et commencer dès le départ puisque la couche de préparation est temporaire.
- On peut donc avoir une autre alternative qui est une couche de préparation persistante, qui n'est jamais tronquée. Au lieu de recommencer à partir des systèmes opérationnels qu'on ne veut pas interroger régulièrement, on peut revenir à cette couche facilement.
- L'utilisation d'une couche de préparation persistante n'est pas toujours conseillée.

Sources des données

id	date	produit
1	1/6/2022	Produit 1
2	1/6/2022	Produit 2
3	1/6/2022	Produit 3
4	1/6/2022	Produit 4
5	1/6/2022	Produit 5
6	1/7/2022	Produit 6
7	1/7/2022	Produit 7



id	date	produit
1	1/6/2022	Produit 1
2	1/6/2022	Produit 2
3	1/6/2022	Produit 3
4	1/6/2022	Produit 4
5	1/6/2022	Produit 5
6	1/7/2022	Produit 6
7	1/7/2022	Produit 7

Lecture est extraction des données
dans la couche intermédiaire
persistante

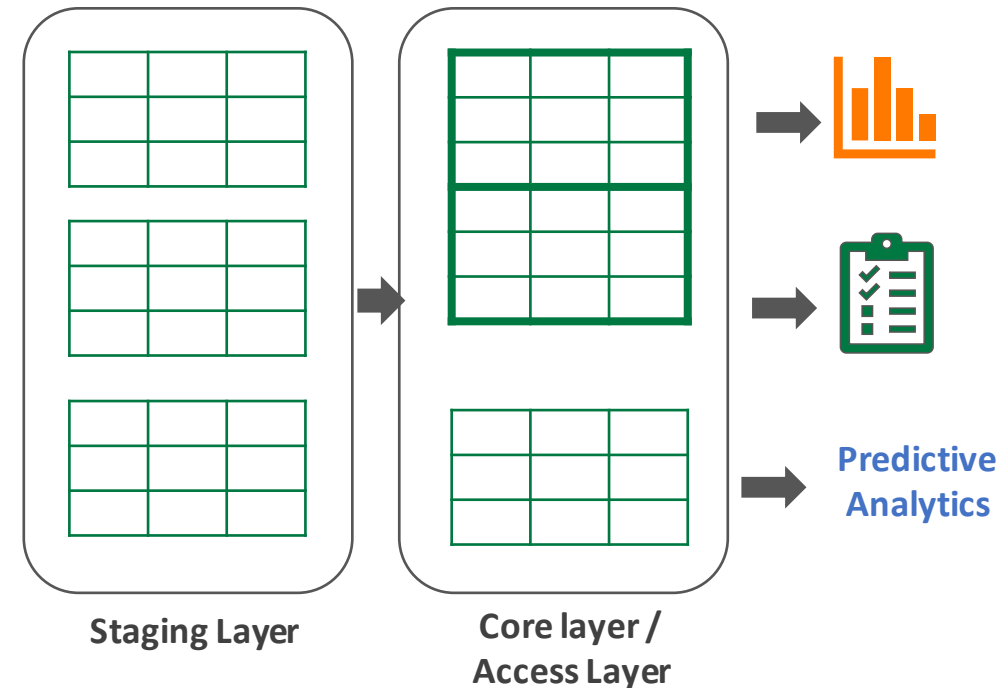
2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Architecture d'un Data Warehouse

Les couches d'un Data Warehouse : Core Layer

- On utilise toujours l'ETL pour copier les données, de la couche de préparation à la couche centrale, qui est considérée aussi comme le Data Warehouse lui-même, pour intégrer les transformations nécessaires.
- Cette couche est généralement utilisée par les utilisateurs ou les applications afin de générer des rapports via le data mining* ou l'analyse prédictive.

* Le data mining est le processus d'exploration et d'analyse de grandes quantités de données pour découvrir des motifs, des relations et des tendances cachées, permettant ainsi de prendre des décisions éclairées et de générer des connaissances précieuses.

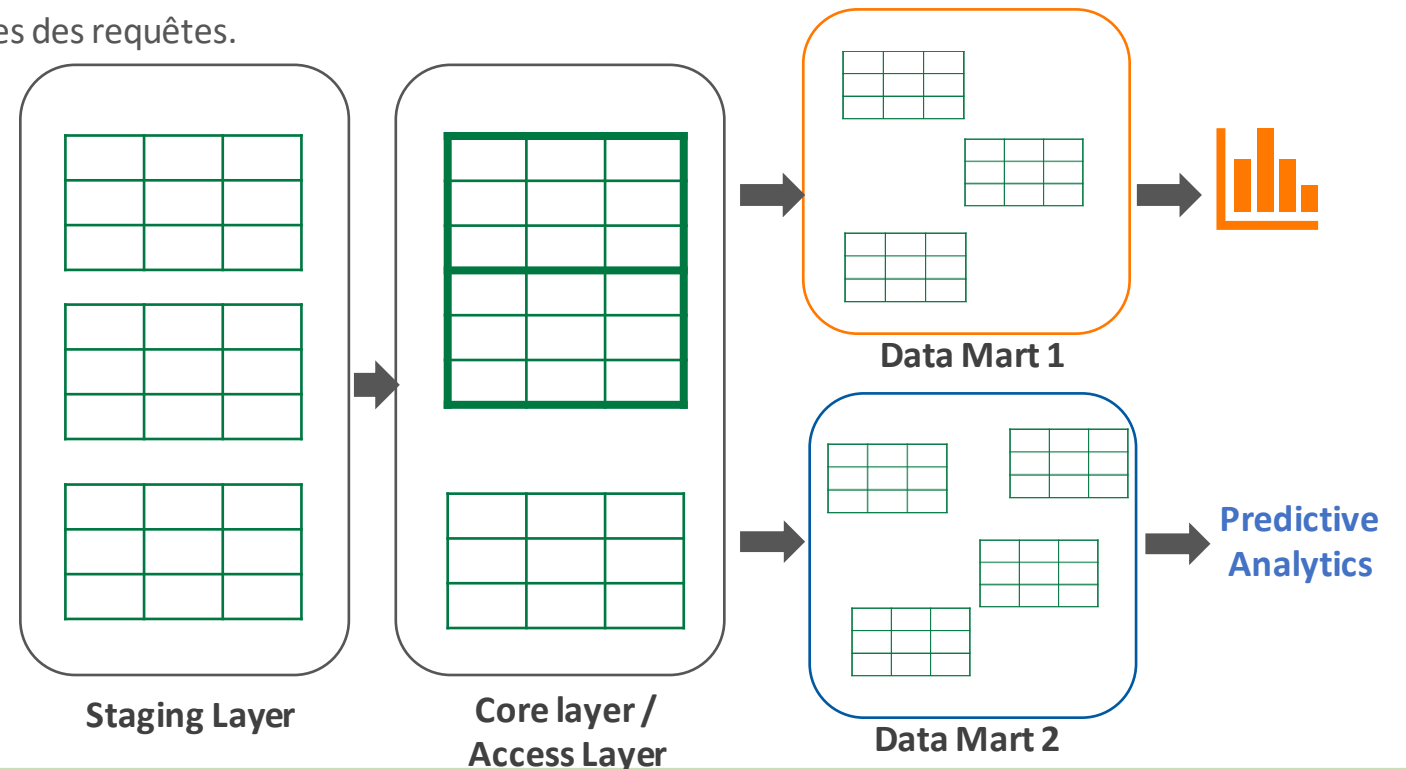


2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Architecture d'un Data Warehouse

Les couches d'un Data Warehouse : Data Mart

- Lorsqu'on dispose d'un grand Data Warehouse, qui est constitué d'un très grand nombre de tables et différents cas d'utilisation, on inclut les **data marts** comme couche entre la couche centrale et les résultats.
- Un data mart prend simplement un ensemble de tables du Data Warehouse (du noyau) qui sont pertinentes pour un cas d'utilisation très spécifique. Cela permet à augmenter les performances des requêtes.



Remarque : les data marts ne sont pas toujours nécessaires

Les couches d'un Data Warehouse : Data Mart

- Un data mart est un sous-ensemble d'un data warehouse (la couche centrale).
- Les données dans un data mart sont modélisées avec un **modèle dimensionnel**. Même dans un Data Warehouse (le noyau) les données peuvent être modélisées avec un modèle dimensionnel.
- Un data mart peut être, à son tour aussi, agrégé, comme un Data Warehouse.
- Les data marts augmentent la convivialité des données puisque on se focalise sur les données pertinentes pour un cas d'utilisation donné.
- Ils peuvent être utilisés pour augmenter la performance puisqu'ils utilisent les modèles dimensionnels, donc on peut utiliser une technologie spécifique à ce modèle :
 - Power BI : In-memory databases
 - Dimensional cubes, etc.

Remarque : L'utilisation des data marts est optionnelle. Cela dépend des cas d'utilisation

2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Architecture d'un Data Warehouse



Les couches d'un Data Warehouse : Data Mart

- Différents outils :
 - Visualisation des données (par exemple Power BI)
 - L'analyse prédictive avec d'autres outils
 - Etc.
- Différents départements, avec différents cas d'utilisation :
 - Département des ventes
 - Département de la finance
 - Département de commerce
 - Département de Management
 - Etc.
- Différentes régions

CHAPITRE 2

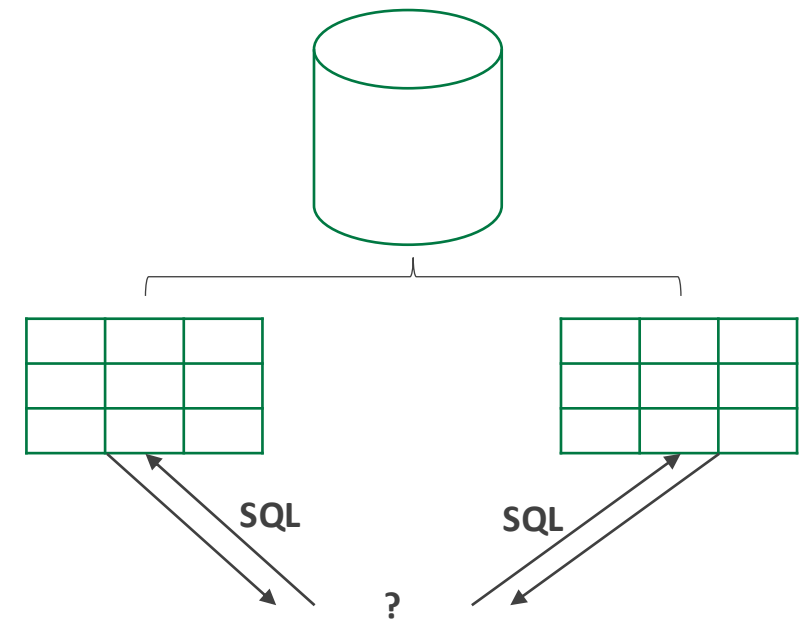
INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

1. Introduction à l'informatique décisionnelle
2. Présentation générale d'un Data Warehouse
3. Architecture d'un Data Warehouse
- 4. Types des bases de données**
5. Data Warehouse vs ODS (Operational Data Storage)
6. Introduction au Modèle dimensionnel



Bases de données relationnelles : Rappel

- Une base de données relationnelles est une simple base de données où les données sont stockées dans des tables. Les données sont structurées sous forme de colonnes et de lignes.
- On utilise le langage SQL pour interroger une base de données relationnelle
 - SELECT pour l'affichage
 - UPDATE pour la modification
 - DELETE pour la suppression
 - INSERT pour l'ajout
- Dans les bases de données relationnelles, on identifie les données dans une table par des clés primaires uniques.
- Les tables sont mises en relations via des clés étrangères.
- Afin de combiner plusieurs tables, on utilise les jointures JOIN.



2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Types des bases de données



Bases de données relationnelles : Rappel

- Les bases de données sont généralement implémentées dans un système d'exploitation (en local). De ce fait on n'interroge qu'une seule au même temps.
- Puisque les tables sont liées par des relations, on peut interroger plusieurs tables, en utilisant l'OLAP* (Online Analytical Processing).
- En plus des bases de données relationnelles, il y a d'autres types comme les bases de données en mémoire (In-Memory Databases)

* ça sera expliqué par la suite

Bases de données en mémoire

- Les bases de données en mémoire (In-Memory Databases) sont hautement optimisées pour la performance des requêtes.
- Elles sont utilisées pour des fins analytiques ou n'importe quel cas d'utilisation se servant d'un grand nombre de requêtes, et demandant des réponses rapides.
- Elles sont généralement utilisées dans les data marts.
- Cette technologie est indépendante de la structure des données (relationnelle ou non-relationnelle). Elle a des solutions pour les deux cas.
- Les données sont généralement stockées dans un disque dur. Lorsque les données sont interrogées par une requête, ces dernières sont chargées en mémoire avec un temps de réponse. Durant cette étape, on a un temps d'attente pour que le résultat de la requête soit retourné, ce qui n'est pas optimal, si on demande une performance élevée.
- Les bases de données en mémoire n'utilisent pas le disque. Toutes les données sont stockées dans la mémoire et donc le temps de réponse du disque est éliminé.



Bases de données en mémoire

- Comme pour les bases de données relationnelles, les bases de données en mémoire disposent d'un ensemble de technologies, algorithmes et méthodes utilisés :
 - Stockage par colonnes (columnar storage)
 - Parallélisation des requêtes (parallel query plans)
 - Etc.
- Les bases de données en mémoire font face à plusieurs challenges :

Durabilité : On risque de perdre toutes les données si la base de données perd l'alimentation ou se redémarre. Afin de garder la durabilité, on doit ajouter d'autres technologies comme les clichés (snapshots) ou les images qui représentent un état spécifique de la base de données, afin d'y retourner en cas de perte des données.

Coût : Le stockage d'une grande quantité de données dans la mémoire est très coûteux.

Même les bases de données traditionnelles essaient d'optimiser l'utilisation du disque dur.

Si on veut adapter ce type de bases de données pour les data marts, on ne doit charger que les données pertinentes pour un cas d'utilisation très spécifique.

2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Types des bases de données

Bases de données en mémoire : exemples



SAP HANA



MS SQL Server In-Memory Tables



Oracle In-Memory



Amazon MemoryDB
(Services Cloud)

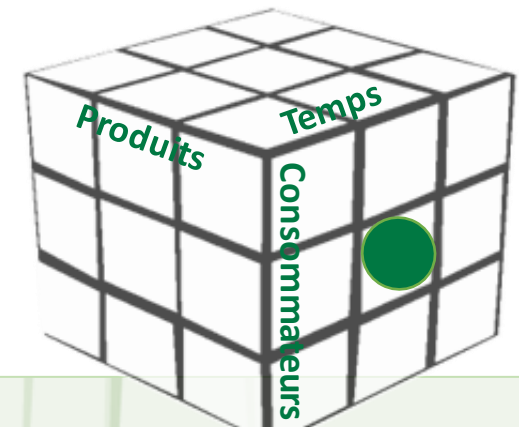
- Tous ces produits peuvent être utilisés dans les data marts si on cherche à avoir des requêtes de bonnes performances.
- En plus des bases de données en mémoire, les **cubes** peuvent être aussi utilisés dans les data marts pour une bonne performance.

Cubes OLAP

- Les cubes OLAP (OnLine Analytical Processing) sont aussi une méthode alternative pour augmenter la performance des data marts.
- Les Data Warehouses traditionnels se basent sur les bases de données relationnelles (ROLAP : Relational OLAP).
- Dans un cube, les données ne sont pas organisées dans des tables liées avec de relations, mais avec les dimensions. On ne parle pas d'une base de données relationnelle. On parle des bases de données multidimensionnelles (MOLAP : Multidimensional OLAP).
- Dans les bases de données multidimensionnelles, les données sont organisées sous forme de tableaux multidimensionnels, c'est-à-dire des tableaux de tableaux (des cubes). Un cube organise les données.
- Les cubes s'utilisent pour des fins analytiques avec une bonne performance. Ils peuvent être utilisés dans différentes solutions BI.

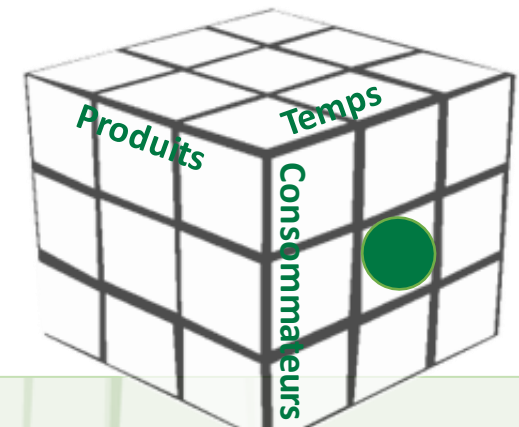
Cubes OLAP : Structure des données

- Prenons l'exemple d'analyse multidimensionnelles des données des ventes. L'exemple présenté se compose de 3 dimensions, juste pour voir visuellement la représentation des données, mais on peut avoir plus de 3 dimensions.
- Dans cet exemple, on a 3 dimensions : produits, temps et consommateurs et on veut analyser les ventes. On peut utiliser ces cellules pour diviser les données. On peut utiliser par exemple l'intersection pour un consommateur dans un certain temps, afin de récupérer un point spécifique de donnée calculée qui représente la quantité de ventes pour ce cas de figure.
- Un des avantages des cubes est qu'ils offrent des valeurs pré-calculées (agrégées).
- Contrairement aux bases de données relationnelles, le langage SQL ne peut pas être utilisé pour les cubes, c'est le langage MDX (Multi Dimensional eXpression) qui s'utilise dans ce cas. C'est un langage de requête développé par Microsoft qui est le plus utilisé pour les cubes.



Cubes OLAP : Recommandations

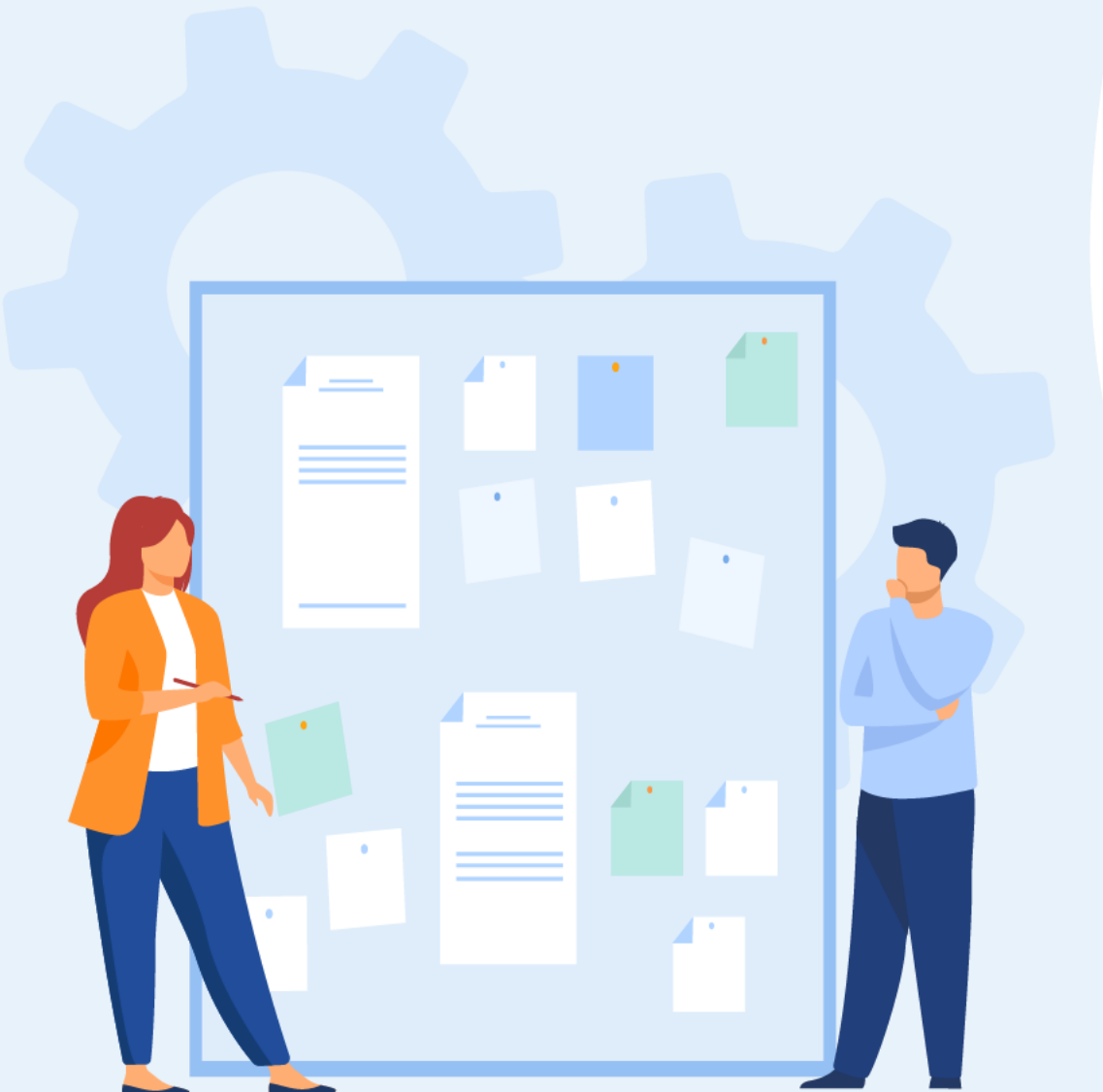
- Ces cubes multi dimensionnels doivent être construits pour un cas d'utilisation spécifique, c'est pourquoi on les utilise dans les data marts. Les cubes sont bénéfiques lorsqu'idéalement on n'a pas plusieurs dimensions.
- Les cubes sont très pratiques pour les requêtes interactives avec des hiérarchies.
- L'utilisation des cubes dans les data marts est optionnelle. On peut tout simplement utiliser les bases de données relationnelles, mais tout en les organisant afin d'avoir une bonne performance.
- Parmi les outils utilisés pour l'organisation des bases de données relationnelles, on cite le « schéma en étoile ».
- Toujours dans le but d'optimiser les performances, elles existent des méthodes alternatives du stockage des données :
 - Modélisation tabulaire (SSAS)
 - ROLAP
 - Stockage en colonnes
 - Traitement parallèle
- Toutes ces technologies peuvent être utilisées dans les data marts afin d'augmenter la performance.



CHAPITRE 2

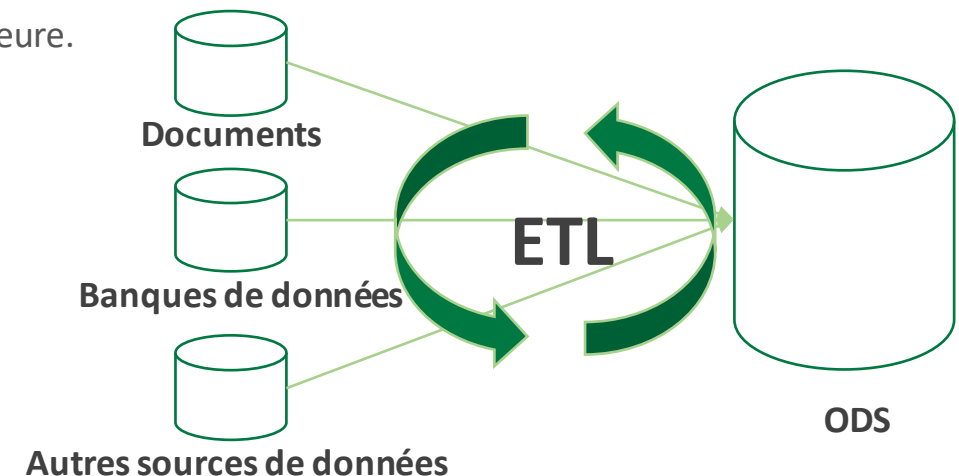
INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

1. Introduction à l'informatique décisionnelle
2. Présentation générale d'un Data Warehouse
3. Architecture d'un Data Warehouse
4. Types des bases de données
- 5. Data Warehouse vs ODS (Operational Data Storage)**
6. Introduction au Modèle dimensionnel



ODS (Operational Data Storage)

- La différence entre un ODS et un Data Warehouse n'est pas claire, car l'architecture de l'ODS ressemble à celle du Data Warehouse.
- Comme pour un Data Warehouse, dans un ODS on dispose d'un ensemble de données de différents types qu'on veut intégrer dans une seule base de données (ODS) en utilisant toujours l'ETL.
- L'ODS est utilisé pour des prises de décision opérationnelles, c'est ce qui rend le processus différent par rapport à un Data Warehouse, puisqu'il n'est pas utilisé pour des prises de décisions analytique ou stratégique.
- Puisque le seul cas d'utilisation d'un ODS est l'opérationnel, il n'a pas besoin d'une longue historique. L'état actuel des données peut être suffisant. Idéalement, cet état actuel doit être retourné immédiatement en temps réel à partir des systèmes sources, contrairement à un Data Warehouse où les données peuvent être mises à jour une fois par jour ou par heure.

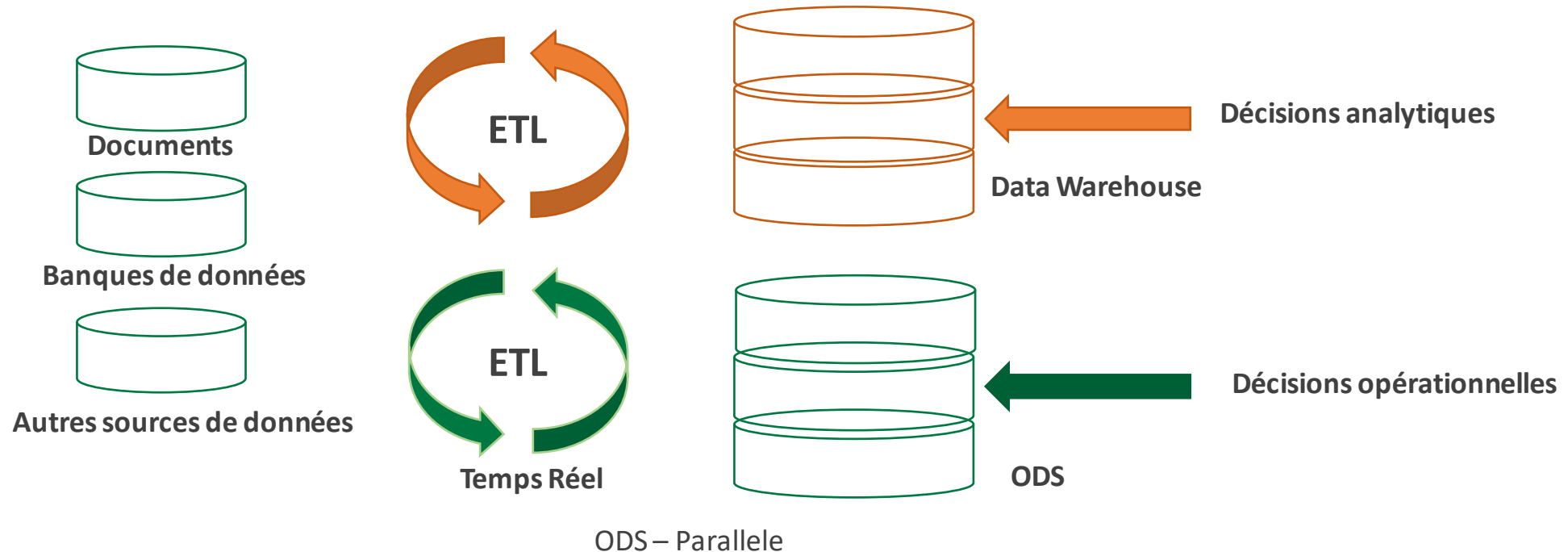


2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Data Warehouse vs ODS (Operational Data Storage)

ODS (Operational Data Storage)

- Si on veut décider si on peut donner à un consommateur un crédit ou non. Dans ce cas on a besoin de consulter ses données comme elles sont dans ces systèmes opérationnels dans un temps réel afin de prendre des décisions plus précises. De plus on n'a pas besoin d'une longue historique. L'utilisation d'un ODS peut être adéquate.
- On peut avoir en plus d'un Data Warehouse, un ODS **en parallèle** dans une même structure, avec un autre ETL (temps réel).

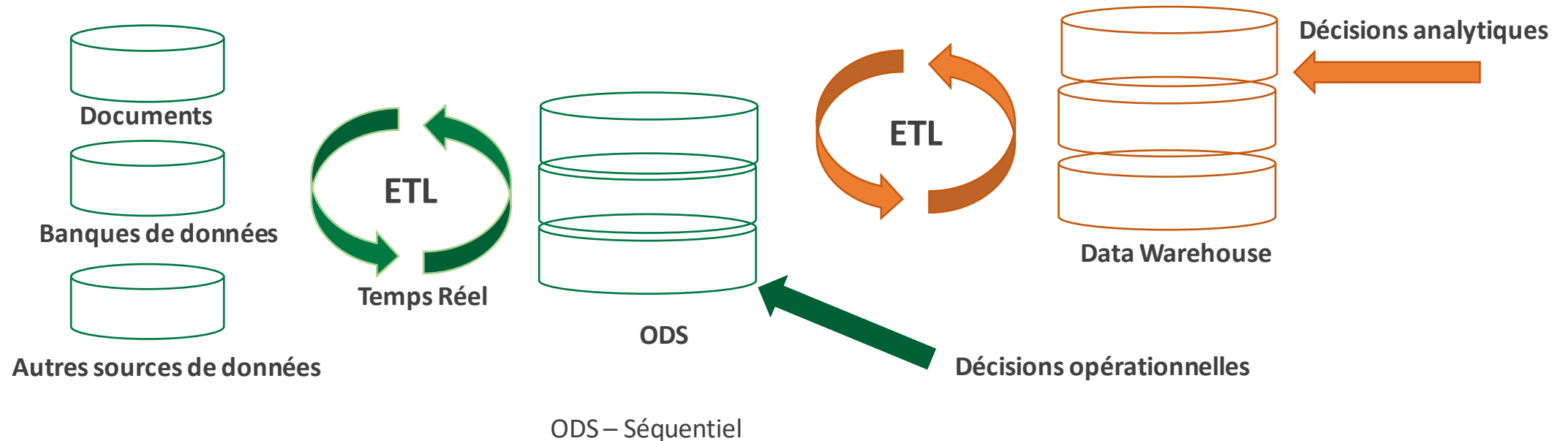


2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Data Warehouse vs ODS (Operational Data Storage)

ODS (Operational Data Storage)

- Afin de mettre l'ODS et le Data Warehouse dans un même organisme, une autre solution est possible. On peut les mettre d'une manière séquentielle au lieu de les mettre en parallèle.
- Dans ce cas, on garde toujours l'ODS, mais on ajoute une couche de l'ETL pour le Data Warehouse au dessus de l'ODS, puisque le premier ETL a déjà fait le grand travail.
- Cet ODS peut être vu comme la source de données, ou la couche de préparation du Data Warehouse.



2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Data Warehouse vs ODS (Operational Data Storage)



ODS (Operational Data Storage)

- L'ODS a devenu moins pertinent, surtout à cause des nouvelles performances du matériel (ETL et bases de données plus rapides). Les données peuvent être chargées très rapidement sans avoir besoin d'un ODS.
- L'apparition des technologies Big Data a aussi baissé la pertinence des ODS, puisqu'elles ont rendu le traitement d'une très grande quantité de données plus rapide.

CHAPITRE 2

INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

1. Introduction à l'informatique décisionnelle
2. Présentation générale d'un Data Warehouse
3. Architecture d'un Data Warehouse
4. Types des bases de données
5. Data Warehouse vs ODS (Operational Data Storage)
- 6. Introduction au Modèle dimensionnel**



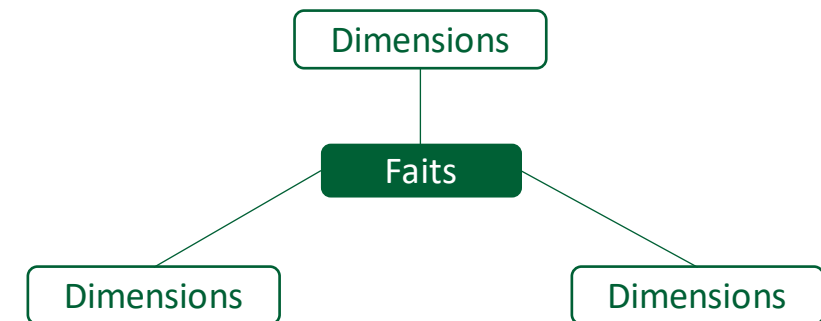
2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Introduction au Modèle dimensionnel



Définition

- Un modèle dimensionnel fait référence à des méthodes permettant l'organisation des données d'une manière spécifique. Ceci est utilisé généralement dans un Data Warehouse, puisque ce dernier a des exigences spécifiques sur les données (facilité d'utilisation, performance, ...).
- Dans un modèle dimensionnel, toutes les données sont organisées sous forme de **faits** et de **dimensions**.
- Un **fait** représente tout ce qui est généralement mesuré (par exemple un gain qui peut être cumulé).
- Les **dimensions** donnent un contexte supplémentaire à ces mesures (par exemple : un mois, une période, une catégorie du produit, etc.). Avec ces dimensions, on peut retourner les faits (les mesures) dans un contexte avec des résultats significatifs (par exemple : analyser les gains par année ou par catégorie).
- Un fait est généralement modélisé par une table au milieu entouré par un ensemble de dimensions regroupées autour de ce fait. Ces différentes dimensions peuvent être utilisées pour analyser les données et les mesures dans la table de fait.



2 - INTRODUIRE LE DOMAINE DU BUSINESS INTELLIGENCE

Introduction au Modèle dimensionnel



Utilité

- Récupération rapide des données, cela est lié à la fois aux performances et à la convivialité
- La modélisation dimensionnelle est utilisée dans les Data Warehouse, les cubes OLAP, etc.

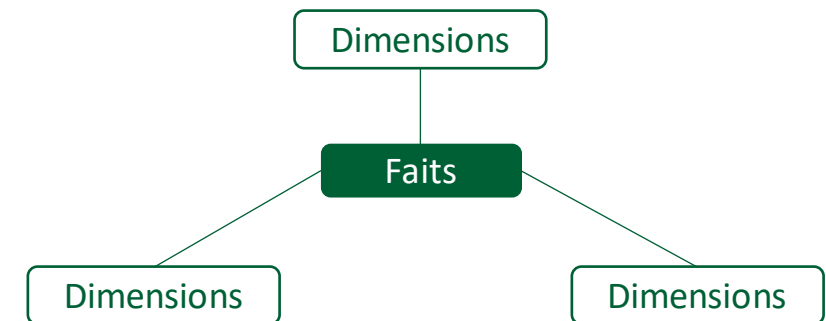


Table de faits

- Comme nous avons vu dans un schéma en étoile, on dispose d'une table de faits au milieu un ensemble de tables de dimensions qui entourent ce dernier.
- La table de fait est la base d'un Data Warehouse (par exemple), parce qu'elle contient les mesures clés d'une entité.
- Ces faits sont les résultats qu'on veut généralement agréger et analyser par les dimensions dont on dispose.

Exemple :

- La table des achats représente la table de fait, tandis que les tables de dimensions, dans cet exemple, sont les produits, les clients et la date.

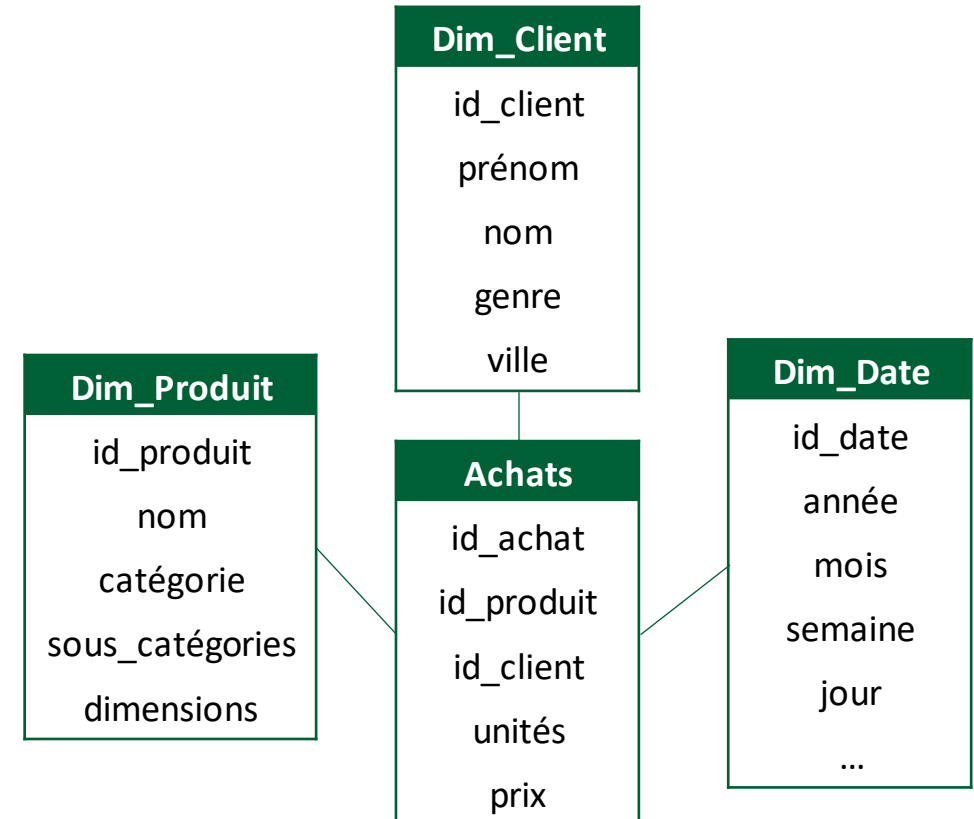


Table de faits : Caractéristiques

- Les faits sont généralement additives, c'est-à-dire qu'on peut les additionner. On peut ajouter des nombres tout en gardant un sens aux résultats (le nombre d'unités par exemple est calculable).
- Contrairement aux dimensions, les faits ne sont pas descriptifs, ils sont mesurables.
- Un fait peut être événementiel ou transactionnel (exemple : Un achat peut être traité comme une transaction ou un autre événement qui se produit dans un temps spécifique).
- On trouve des fois le temps ou la date comme une colonne incluse dans la table de faits. Ces données ne sont pas des faits eux même, mais elles sont incluses dans la table de faits.

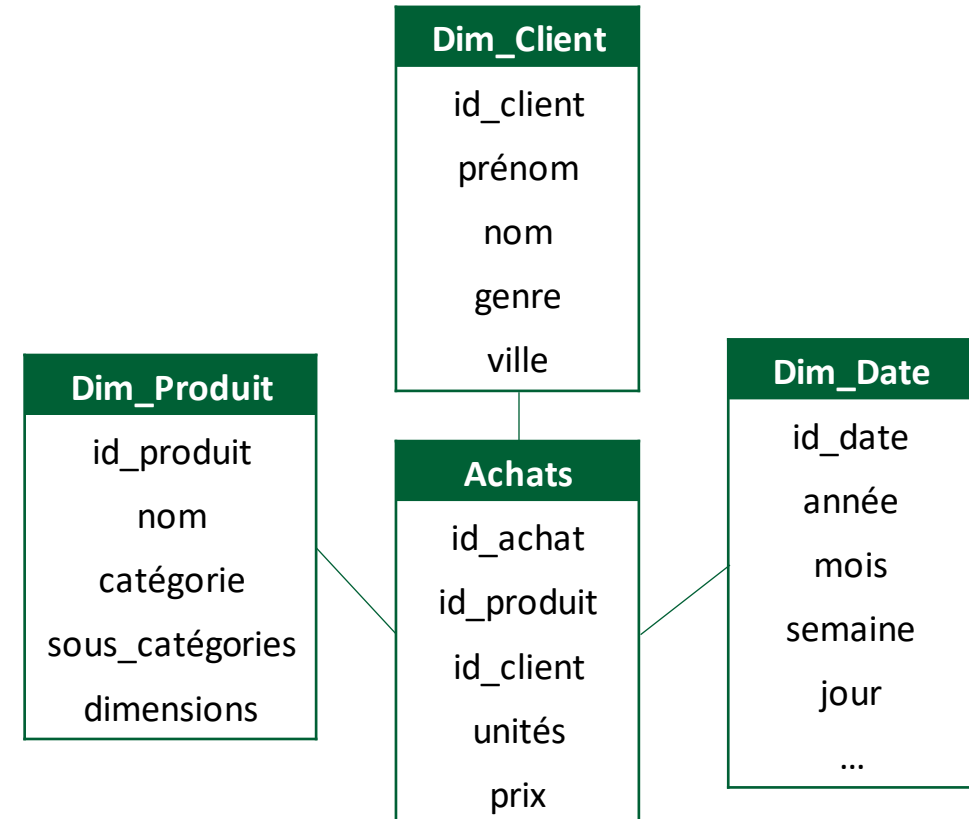


Table de faits

- Une table de fait se compose d'une clé primaire (comme dans les bases de données relationnelles), multiples clés étrangères qui font références aux dimensions et les faits eux-mêmes qui sont les mesures clés.
- Une table de faits est définie par ce qu'on appelle le **grain**. Le grain est le niveau le plus atomique d'un fait.

Exemple

- Prenons la table suivante, on remarque qu'on a un profit pour chaque région et date.
- On a un profit dans une ligne pour une région spécifique dans un temps spécifique. Ceci donc est le niveau le plus atomique. C'est le grain de cette table.

id	id_date	id_region	profit (DH)
1	20092022	1	200
2	20092022	2	150
3	20092022	10	900
4	20092022	4	300
5	20092022	3	250

Table de dimensions

- Les dimensions servent à catégoriser les faits afin d'obtenir un contexte significatif à nos mesures. Si par exemple, on n'a qu'un nombre total des unités vendues, on n'a pas vraiment des résultats significatifs.
- Le caractère de ces dimensions est plus **descriptif**. On ne peut pas le mesurer mais on peut l'utiliser pour décrire un fait (un produit, une catégorie, ...). Ceci aide à supporter les faits et **analyser**, **filtrer** les groupes et **labéliser** les données.
- Afin de distinguer entre les faits et les dimensions, voici les caractéristiques communes des dimensions :

Les tables de faits sont agrégées (calculables : +, - . Etc.) et donc numériques, tandis que les dimensions peuvent être numériques sauf qu'ils ne peuvent plus être agrégées.

Leur caractère est descriptif, contrairement à celui des faits qui est mesurable.

Dans la table de faits, on a toujours des mesures qui changent (des valeurs), par contre aux dimensions qui sont plus statiques.

Table de dimensions

- Comme une table de faits, la table de dimension se compose aussi d'une clé primaire et les dimensions. Elle peut même comporter des clés étrangères. Cela devient important lorsqu'on parle **d'un schéma en flocons**.
- Les dimensions peuvent représenter des personnes (employés, consommateurs, managers), des produits, des lieux (régions, villes, adresses), des temps, etc.

Exemple

- Prenons la table des consommateurs suivante comme une table de dimension. On a id_consommateur comme une clé primaire et les différents dimensions (prénom_consommateur, nom_consommateur et email_consommateur).
- Contrairement aux faits, les dimensions peuvent être changées lentement (de temps à autre).

id_consommateur	prénom_consommateur	nom_consommateur	email_consommateur
1	Malak	Grini	mgrini@gmail.com
2	Sofia	Falahi	falahiS@gmail.com
3	Karim	Fassi	KarimFassi@gmail.com
4	Khalid	Chouaib	Chouaib111@gmail.com
5	Fatima Zahra	Tlemsani	FZTlemsani@gmail.com

Table de dimensions

- Comme nous avons vu dans la première représentation de la modélisation dimensionnelle, les faits et les dimensions sont présentées sous forme d'une étoile. On parle du schéma en étoile.
- En plus du schéma en étoile le modèle dimensionnel peut être représenté par un autre schéma appelée « **schéma en flocon** »

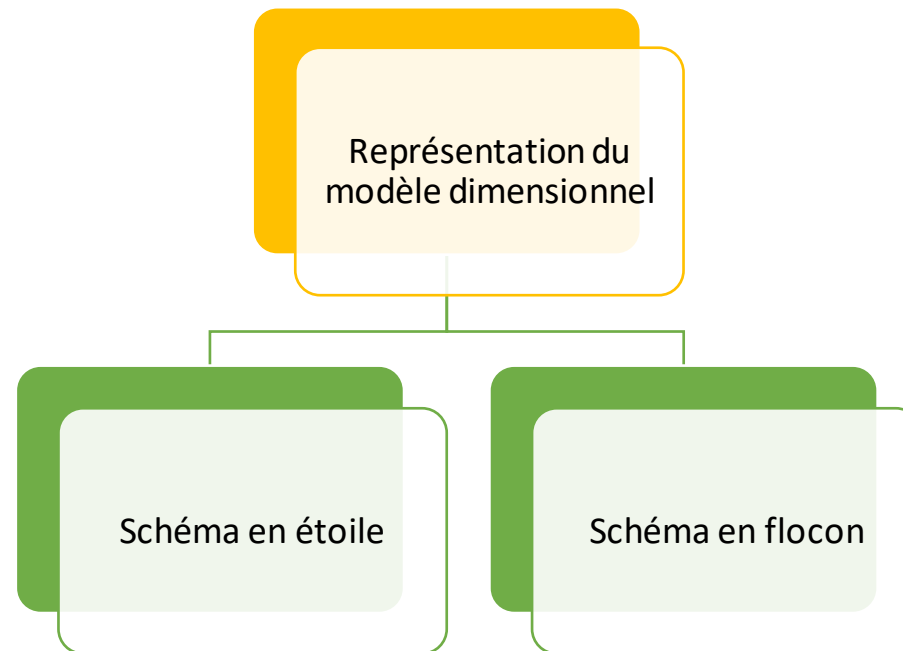


Schéma en étoile

- Le schéma en étoile est le schéma le plus important dans un Data Warehouse, et plus précisément dans les data marts.
- Comme on a déjà vu dans un modèle dimensionnel, on arrange et structure les données sous formes de faits et de dimensions. Si on reprend le même exemple des achats, on a la table des achats qui contient tous les faits importants, et on crée des relations avec les dimensions (afin de joindre cette table avec les tables de dimensions), en utilisant les clés étrangères et primaires.
- Comme dans le modèle relationnel, on parle de différents types d'associations (relations) (1:n, 1:1, etc.). Dans cet exemple, et généralement entre une table de fait et une autre de dimension, on a une relation 1:n, 1 coté dimensions et n coté faits. Chaque produit (dimension) peut faire l'objet de plusieurs achats, tandis qu'un achat ne se fait que sur un seul produit.

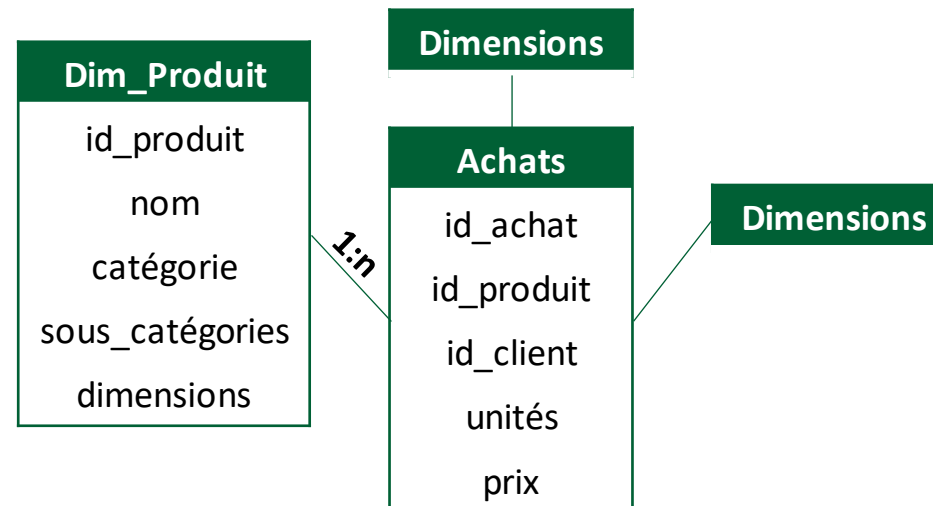


Schéma en étoile

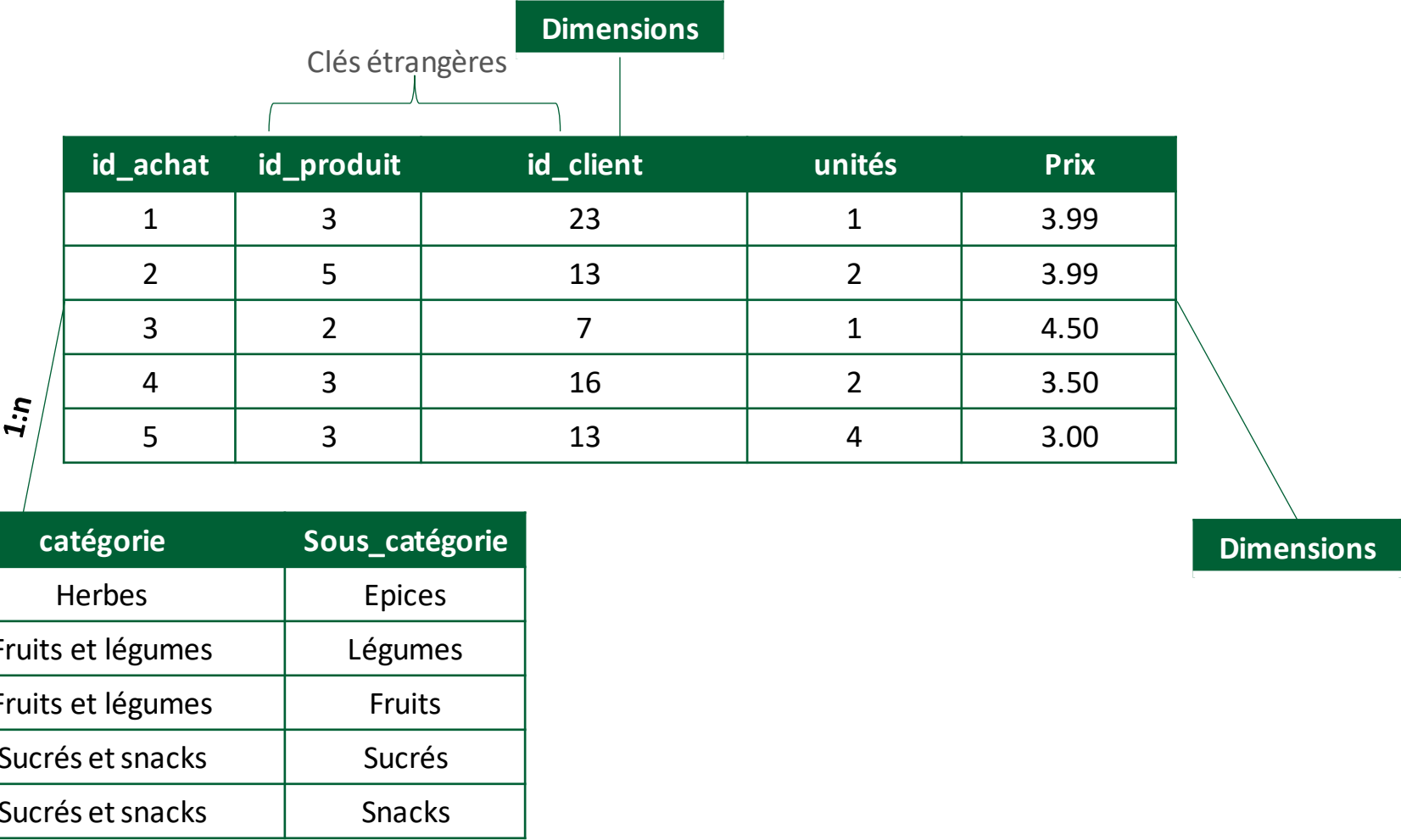


Schéma en étoile

- Comme on n'a, dans un schéma en étoile, qu'un seul niveau hiérarchique, on a donc une seule connexion entre une table de dimension et une autre de faits (on ne trouve pas une autre connexion partant de la même table de dimensions) . Cela peut engendrer une redondance de données, parce que la colonne catégorie par exemple doit appartenir un autre niveau d'hiérarchie.
- On verra par la suite, le schéma en flocon qui est une alternative qui peut réduire cette redondance. Cette réduction de la redondance des données est appelée « normalisation ». C'est une technique mathématique. Dans le cas d'un schéma en étoile, la base de données est plus au moins **dénormalisée**.
- La normalisation est importante dans certains cas, mais pas vraiment idéale pour récupérer les données et avoir une bonne lecture de nos opérations, surtout lorsqu'on dispose de plusieurs tables et requêtes. Dans ce cas, on peut accepter cette redondance des données si elle répond à ce qu'on veut faire avec ces données.

id_produit	nom	catégorie	Sous_catégorie
1	Chili	Herbes	Epices
2	Ail	Fruits et légumes	Légumes
3	Banane	Fruits et légumes	Fruits
4	Chocolat	Sucrés et snacks	Sucrés
5	Chips	Sucrés et snacks	Snacks

Schéma en étoile : Remarques

- On peut trouver, dans certains cas, plusieurs tables de faits. Mais la situation la plus commune et idéale est d'avoir une seule table de faits.
- Dans le cas de plusieurs tables de faits, la même table de dimensions peut être pertinente pour plus qu'une seule table de faits (plusieurs connexions).
- Le schéma en étoile est applicable pour des besoins très spécifiques (un ensemble de requêtes bien précises).

Schéma en flocon

- Théoriquement, le schéma en étoile est un cas spécifique d'un schéma en flocon. Cette dernière est le concept le plus général, parce qu'il permet plusieurs niveaux d'hierarchies. Un schéma en étoile st un schéma en flocon avec un seul niveau d'hierarchie.
- Prenons le même exemple afin de comprendre à quoi ressemble un schéma en flocon. On a remarqué une redondance de données au niveau de la catégorie du produit qu'on a accepté dans le schéma en étoile pour la raison de la visibilité et la bonne lecture.

id_produit	nom	catégorie	Sous_catégorie
1	Chili	Herbes	Epices
2	Ail	Fruits et légumes	Légumes
3	Banane	Fruits et légumes	Fruits
4	Chocolat	Sucrés et snacks	Sucrés
5	Chips	Sucrés et snacks	Snacks

id_achat	id_produit	id_client	unités	Prix
1	3	23	1	3.99
2	5	13	2	3.99
...

Schéma en flocon

- Mais si on veut utiliser un schéma en flocon, cette redondance peut être réduite, en ajoutant une table de catégorie dans un niveau inférieur (2 ème niveau de hiérarchie) et en gardant seulement l'identifiant de la catégorie (clé étrangère). Ceci permet de prendre moins d'espaces dans le disque.
- Contrairement au schéma en étoile, le schéma en flocon est plus normalisé.

id_achat	id_produit	id_client	unités	Prix
1	3	23	1	3.99
2	5	13	2	3.99
...

Faits

id_produit	nom	catégorie	Sous_catégorie
1	Chili	1	Epices
2	Ail	2	Légumes
3	Banane	2	Fruits
4	Chocolat	3	Sucrés
5	Chips	3	Snacks

Dimensions

Dimensions

id_produit	catégorie
1	Herbes
2	Fruits et légumes
3	Sucrés et snacks

Schéma en étoile vs Schéma en flocon

- Le tableau suivant résume les avantages et les inconvénients d'un schéma en flocon par rapport au schéma en étoile.

Avantages	Inconvénients
Moins d'espace de stockage	Beaucoup plus compliqué par rapport au schéma en étoile (plus de tables qui peuvent être plus compliquées à comprendre)
Moins de redondance → facile à maintenir et à modifier et donc moins de données endommagées	Besoins de beaucoup de jointures afin de récupérer une information (des requêtes SQL plus complexes)
Résoudre quelques ralentissements (lors des mises à jour des données)	Data Marts / Data Warehouse moins performants à cause des jointures

- Un schéma en flocon n'est pas utilisé dans les data marts, puisqu'on a besoin des requêtes rapides. On utilise généralement un schéma en étoile dans la mesure du possible.
- Le schéma en flocon est utilisé lorsqu'on rencontre des difficultés dans la maintenance ou la mise à jour des données, ou lorsque le coût du stockage est un vrai challenge, chose qui est très rare.



PARTIE 2

APPRÉHENDER LE MODÈLE DIMENSIONNEL

Dans ce module, vous allez :

- Maîtriser les faits
- Maîtriser les dimensions
- Appréhender les dimensions à évolution lente



45 heures

CHAPITRE 1

MAITRISER LES FAITS

Ce que vous allez apprendre dans ce chapitre :

- Appliquer l'additivité dans les faits
- Traiter les cas des Nulls
- Connaitre le Year-to-date
- Différencier entre les différents types des tables de faits
- Traiter le cas des tables de faits sans faits
- Déterminer les étapes de création d'une table de fait
- Maitriser la notion du clé de substitution



20 heures



CHAPITRE 1

MAITRISER LES FAITS

1. **Additivité**
2. Nulls
3. Year-to-date
4. Types des tables de faits
5. Tables de faits sans faits
6. Étapes de création d'une table de fait
7. Clé de substitution



Additivité

- Les faits sont généralement des valeurs numériques. Ça veut dire que plusieurs opérations arithmétiques peuvent être appliquées sur ces derniers, y compris l'addition afin de préparer des analyse ou des rapports et déduire des totaux.
- Ces opérations ne sont pas toujours possibles (le total n'a pas toujours un sens) selon le fait en question. Il faut donc distinguer entre trois types d'additivité dans les faits :

Le fait entièrement additif

- C'est le fait le plus commun.
- Il peut être ajouté dans toutes les dimensions.
- Le résultat reste toujours significatif.
- C'est le fait le plus flexible et utilisable.
- C'est la valeur la plus analytique.

Le fait semi-additif

- Il peut être ajouté dans quelques dimensions.
- Il est utilisé soigneusement.
- Il est moins flexible.
- Le calcul de la moyenne peut être une alternative.
- Exemple : Soldes des comptes.

Le fait non-additif

- Il ne peut pas être ajouté dans les dimensions.
- Ce sont des valeurs analytiques limitées.
- Le stockage des valeurs sous-jacentes peut être une alternative.
- Exemple: pour le ratio, on peut par exemple stocker le numérateur et dénominateur et calculer le ratio en cas de besoin.

Exemples : Fait additif

- On remarque qu'on a, dans la table des achats, cinq différents achats, avec un nombre d'unités différents. Ces unités peuvent être additionnées afin de récupérer le nombre total d'unités (10). Ce chiffre reste toujours significatif, même en sommant tous les valeurs des unités.
- On peut regrouper les données par produit, client ou par catégorie afin d'obtenir des résultats plus significatifs.
- Les noms des produits, des clients et des catégories peuvent être récupérés à partir des tables de dimensions associées à cette table de faits.
- Les unités est un fait **entièrement additif**.

id_achat	id_produit	id_client	unités	Prix
1	3	23	1	3.99
2	5	13	2	3.99
3	2	7	1	4.50
4	3	16	2	3.50
5	3	13	4	3.00
			10	

id_produit	unités
3	7
5	2
2	1

id_client	unités
23	1
13	6
7	1
16	2

Exemples : Fait semi-additif

- On suppose qu'on dispose des soldes des comptes, c'est l'exemple le plus commun dans le cas des faits semi additifs.
- On a dans la première ligne qui correspond au premier jour, un solde de compte de 50 \$, dans le deuxième jour (2^{ème} ligne) un solde de 100\$ qui est le résultat de l'ajout d'une somme au dernier solde qui était 50\$. De même pour le troisième jour, aucune transaction n'a été faite, c'est pourquoi le solde n'a pas changé.

id_portfeuille	type
1	Espèces en USD
2	Actions

id_solde	id_portfeuille	id_date	solde
1	1	20230203	50\$
2	1	20230204	100\$
3	1	20230205	100\$
4	2	20230203	120\$
5	2	20230204	170\$

id_date	type
20230203	170\$
20230204	270\$
20230205	160\$

Exemples : Fait semi-additif

- Dans ce cas, l'addition de ces valeurs n'a aucun sens, car 100\$ correspond déjà au solde total du portefeuille 1. Mais, on peut en effet additionner les valeurs entre les types des portefeuilles. Par exemple, pour le jour numéro 1 (20230203), on peut sommer les deux soldes (50 + 120 = 170), puisqu'ils correspondent à deux opérations différentes. On n'additionne pas par rapport aux dates, mais seulement les types de portefeuilles. Le total des soldes par rapport aux dates n'est plus significatif. Par contre la moyenne peut avoir un sens.
- Le solde ici est un fait **semi-additif**.
- La date est un exemple typique d'une dimension où les valeurs semi-additives ne peuvent pas s'additionner.

id_portfeuille	type
1	Espèces en USD
2	Actions

id_solde	id_portfeuille	id_date	solde
1	1	20230203	50\$
2	1	20230204	100\$
3	1	20230205	100\$
4	2	20230203	120\$
5	2	20230204	170\$

id_date	type
20230203	170\$
20230204	270\$
20230205	160\$

Exemples : Fait non-additif

- On a déjà vu un fait non-additif dans l'exemple des achats, qui est le prix unitaire. Si on veut récupérer le revenu total, on doit sommer le produit du prix par le nombre d'unités. Le revenu retourné est maintenant additif et non pas le prix. Car on ne peut pas les sommer comme ils sont. Même par rapport à la catégorie ou le produit, la somme ne sera pas significative.
- Le prix unitaire est un fait complètement non-additif. Même le calcul de la moyenne est difficilement interpretable, car il faut toujours considérer le nombre d'unités.
- Ils existent autres exemples de faits typiques comme les pourcentages dans certains cas, et les ratios.

id_achat	id_produit	id_client	unités	Prix
1	3	23	1	3.99
2	5	13	2	3.99
3	2	7	1	4.50
4	3	16	2	3.50
5	3	13	4	3.00

CHAPITRE 1

MAITRISER LES FAITS

1. Additivité
- 2. Nulls**
3. Year-to-date
4. Types des tables de faits
5. Tables de faits sans faits
6. Étapes de création d'une table de fait
7. Clé de substitution



Manipulation des nulls dans les faits

- Comme pour l cas de l'additivité des faits, on va étudier le cas des nulls. On va voir ce qui va se produire si on a des nulls dans les faits. Pour ce fait, on va prendre un exemple.
- Généralement, les valeurs nulls ne causent aucun problème, car toutes les agrégations qu'on fait en utilisant SQL ou un outil BI (Power BI, Tableau, etc) peuvent les traiter facilement (Exemple suivant).

id_solde	id_portfeuille	solde	Incoming	Outcoming
1	1	50\$	null	null
2	1	100\$	50\$	null
3	1	100\$	null	null
4	2	120\$	null	null
5	2	170\$	50\$	null
6	2	60\$	null	110\$

Manipulation des nulls dans les faits

- Mais parfois, il faut être un peu prudent. Le calcul de la moyenne dans cet exemple a donné 50\$ comme valeur, car SQL a considéré qu'on ne dispose que 2 valeurs $((50\$ + 50\$)/2)$, ce qui n'est pas correct parce qu'on dispose de 6 valeurs (soldes) et donc il faut avoir 16,67\$.
- Dans certains cas, on peut remplacer les nulls par des zéros si le sens du null est vraiment zero.

id_solde	id_portfeuille	solde	Incoming	Outcoming
1	1	50\$	null	null
2	1	100\$	50\$	null
3	1	100\$	null	null
4	2	120\$	null	null
5	2	170\$	50\$	null
6	2	60\$	null	110\$

```
SELECT
    AVG(Incoming),
    MIN(Incoming),
    SUM(Incoming)
FROM table_soldes
```

AVG	MIN	SUM
50\$	50\$	100\$
16.67\$	0\$	

Manipulation des nulls dans les faits

- Si on a des clés étrangères nulles (aucun portefeuille n'est associé), on peut avoir des conflits (manque de données) lorsqu'on veut connecter la table de faits avec d'autres tables de dimensions. Dans ce cas, il faut créer une valeur fictive (999 par exemple) indiquant que c'est une valeur spéciale/différente.
- Cette valeur ajoutée peut être intégrée dans la table de dimensions associée. Dans cet exemple, cette valeur peut signifier un compte obsolète par exemple.

id_portfeuille	type
1	Espèces en USD
2	Actions
999	Autres types

id_solde	id_portfeuille	solde	Incoming	Outcoming
1	1	50\$	0\$	0\$
2	999	100\$	50\$	0\$
3	1	100\$	0\$	0\$
4	2	120\$	0\$	0\$
5	999	170\$	50\$	0\$
6	2	60\$	0\$	0\$

CHAPITRE 1

MAITRISER LES FAITS

1. Additivité
2. Nulls
- 3. Year-to-date**
4. Types des tables de faits
5. Tables de faits sans faits
6. Étapes de création d'une table de fait
7. Clé de substitution

Manipulation des faits Year-to-date

- Les year-to-date sont des calculs spécifiques sur des faits qui sont en fait problématiques. Par conséquent, on va voir comment gérer ces calculs.
- Ces calculs sont souvent demandés par les utilisateurs professionnels afin de récupérer des résultats / données annuels ou mensuels dans leurs rapports. Et par conséquent, il faut calculer les cumuls annuels ou mensuels et les stocker en colonnes dans un Data Warehouse par exemple. Cela se fait même pour les cumuls trimestriels par exemple.
- Ces variations sont problématiques, car ces calculs ne sont pas dans les **grains** définis dans la table de faits. La table de faits a toujours des objectifs (**grains**) définis (journalier, mensuel, ...). si par exemple on veut stocker les revenus journaliers, chaque ligne correspondra au revenu de chaque jour. Mais lorsqu'on a des valeurs qui n'ont pas le même grain, cela posera des problèmes et mènera à des calculs et requêtes erronés, lorsque les utilisateurs veulent récupérer ces valeurs par rapport à différentes dimensions de dates.
- De ce fait, il ne faut pas stocker ces résultats physiquement. La meilleure alternative est de ne stocker que les valeurs sous-jacentes (les plus générales), par exemple, les revenus journaliers et calculer les autres variations to-date dans le cas de besoin.
- Les outils BI comme Power BI, Tableau, etc. peuvent calculer ces variations très efficacement.

CHAPITRE 1

MAITRISER LES FAITS

- 
1. Additivité
 2. Nulls
 3. Year-to-date
 - 4. Types des tables de faits**
 5. Tables de faits sans faits
 6. Étapes de création d'une table de fait
 7. Clé de substitution

1 - MAITRISER LES FAITS

Types des tables de faits



Manipulation des nulls dans les faits

Il existe différents types de tables de fait, on cite :

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

1 - MAITRISER LES FAITS

Types des tables de faits



Tables de faits transactionnelles

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

- La table de faits la plus fondamentale et flexible est la transactionnelle.
- Dans ce type de table, chaque ligne est définie par un seul événement. Les faits sont donc des mesures d'un seul événement ou une seule transaction.
- Cette seule transaction a généralement lieu à un moment et lieu bien précis, comme une opération de vente par exemple.
- Cette transaction est la définition basique de notre grain. Le grain signifie qu'on a une seule transaction qui est représentée dans une seule.

id_achat	id_produit	id_date	unités
1	3	20230201	1
2	5	20230202	1
3	2	20230202	2
4	3	20230203	1

1 - MAITRISER LES FAITS

Types des tables de faits



Tables de faits transactionnelles

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

- On remarque dans l'exemple des achats, que chaque ligne représente exactement une opération/transaction de vente et pour chaque transaction, on peut avoir différentes clés étrangères (id_produit, id_date) et mesures (le nombre d'unités dans ce cas) et donc on dispose de toute l'information sur la transaction (le produit, la date et la quantité).
- Généralement, une table de faits transactionnelle est flexible, elle permet une analyse de différents manières et avec différentes dimensions. Elle a des mesures qui sont généralement additives et a tendance d'être associée à plusieurs dimensions (via les clés étrangères).
- Ces tables peuvent avoir une taille plus grande et se grandit très rapidement en fonction du nombre des transactions.

id_achat	id_produit	id_date	unités
1	3	20230201	1
2	5	20230202	1
3	2	20230202	2
4	3	20230203	1

1 - MAITRISER LES FAITS

Types des tables de faits



Tables de faits périodiques

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

- Dans la table de faits périodique, une ligne est définie par la synthèse ou l'agrégation d'une mesure (calcul) par rapport à plusieurs événements. Elle se déroule dans une période précise (1 heure, 1 jour, 1 semaine, etc.). Donc on agrège tous les événements et les transactions et on calcule la mesure relative à cette période.
- La période la plus courte définit notre grain.
- Prenons l'exemple des ventes, généralement il a une table de transaction derrière et qui accompagne cette table périodique, car on n'aura que les enregistrements d'une seule période (1 semaine dans cet exemple). Cette information se déduit à partir de la table transactionnelle à la fin de chaque semaine.

id_semaine	Revenu	Ventes	Coût
1	326	125	12
2	252	147	31
3	414	265	12
4	105	120	51

1 - MAITRISER LES FAITS

Types des tables de faits



Tables de faits périodiques

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

- On remarque que la table périodique peut contenir plusieurs mesures (revenu, ventes, etc.) mais peu de dimensions.
- La table de faits périodique se caractérise par une taille moins grande par rapport à la table de faits transactionnelle, parce que le grain n'est pas assez détaillé.
- La croissance de la taille de la table est très contrôlée. On a toujours une seule ligne qui s'ajoute dans un jour, une semaine ou une période précise.
- Les faits dans la table périodique sont généralement additifs.
- Les périodes qui ne présentent aucun événement sont présentées par null ou 0.

id_semaine	Revenu	Ventes	Coût
1	326	125	12
2	252	147	31
3	414	265	12
4	105	120	51

1 - MAITRISER LES FAITS

Types des tables de faits



Tables de faits cumulatives

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

- La table de faits cumulative ressemble un peu à la table périodique, sauf qu'on a une seule ligne dans cette table qui définit le résumé/récapitulatif/cumul d'une mesure de plusieurs transactions. Dans ce type, la période n'est pas normalisée, mais elle est définie par la durée de vie d'un processus qui a un début et une fin spécifiques (i.e. une exécution d'une commande). C'est le type de tables de faits le moins utilisé.
- Ce type de tables est bénéfique dans l'analyse du flux du travail ou d'un processus par exemple.

1 - MAITRISER LES FAITS

Types des tables de faits



Tables de faits cumulatives

Types des tables de faits

Tables de faits transactionnelles

Tables de faits périodiques

Tables de faits cumulatives

- Prenons l'exemple des commandes. On a une commande qui arrive chez un fabricant, avec une date (clé étrangère), un nombre de produits, le type du produit lui-même (clé étrangère), les dates de début et de fin de production, la date d'inspection, la date de livraison (elles sont toutes des clés étrangères) et enfin le nombre de produits endommagés qui correspond au fait associé à cette table en plus du nombre de produits. Toutes les dates sont des périodes ou étapes. Ceci est un très bon exemple des tables de faits cumulatives.
- Dans ce type de tables, on trouve des mesures/faits et beaucoup de dimensions (plusieurs dates/clés étrangères) pour chaque processus.
- Même si la table contient plusieurs clés étrangères des dates, mais on a plusieurs connexions avec une seule dimension date, cela s'appelle une dimension de jeu de rôle.

Id_ commande	Id_date_ commande	Nb_ produits	Id_ produit	Début_ production	Fin_ production	Date_ inspection	Date_ livraison	Produits_ endommagés
1	20230203	100	35	20230204	20230210	20230212	20230213	3
2	20230204	100	35	20230204	20230212	20230213	20230213	4
3	20230204	100	35	20230204	20230212	20230213	20230214	1
4	20230205	100	35	20230206	20230213	20230215	20230216	0

1 - MAITRISER LES FAITS

Types des tables de faits



Comparaison

La différence entre les 3 types de tables de faits.

Type	Table de faits de transaction	Table de faits périodique	Table de faits cumulative
Grain	1 ligne = 1 transaction	1 ligne = une période définie (plus autres dimensions)	1 ligne = durée de vie d'un processus/événement avec un début et fin précis
Dimensions Date	1 date de la transaction	Date de la période (clé étrangère qui représente la fin de la période)	Dates de plusieurs périodes
Nombre de dimensions	Grand	Peu	Très grand
Faits	Mesures des transactions	Mesures cumulatives des transactions dans la période	Mesures du processus dans sa durée de vie
Taille	Grande (grains plus détaillés)	Moyen (grain moins détaillé)	Petite
Performance	Moins performant (peut être amélioré par l'agrégation)	Mieux (moins de détails)	Meilleure performance

CHAPITRE 1

MAITRISER LES FAITS

1. Additivité
2. Nulls
3. Year-to-date
4. Types des tables de faits
- 5. Tables de faits sans faits**
6. Étapes de création d'une table de fait
7. Clé de substitution



1 - MAITRISER LES FAITS

Tables de faits sans faits



Tables de faits sans faits

- La table de faits sans faits reste toujours une table de faits, car on sait qu'une table de faits est différente d'un fait. Dans une table de faits, on peut trouver un ensemble de faits. Le fait n'est qu'une mesure numérique utilisée pour suivre les performances d'un certain processus et la table de faits est la table entière qui garde le suivi de tous ces faits et des clés étrangères.
- Il est possible de trouver une table de faits qui ne contient aucun fait (que les clés étrangères par exemple), dans le cas où seulement les aspects dimensionnels d'un événement ou d'une transaction sont stockés.

1 - MAITRISER LES FAITS

Tables de faits sans faits



Tables de faits sans faits

Exemple

- On travaille dans une entreprise, et on enregistre chaque nouvel employé. On a donc cette table de fait, ou on stocke chaque enregistrement et toutes les dimensions associées; la date d'entrée de l'employé, son département, sa région et son manager. Dans cet exemple, on ne dispose d'aucune mesure, mais on enregistre tous les aspects dimensionnels.
- Même dans l'absence des mesures (faits), on peut répondre aux questions : Combien d'employés ont été enregistrés le mois dernier? Combien d'employés ont été enregistrés dans une certaine région ou département? en utilisant des simples requêtes SQLs.
- Avec ce type de table, on ne stocke que l'événement (en utilisant les clés étrangères) afin de garder une trace de ce dernier, sans avoir besoin d'aucune métrique.

Id_employe	Id_date_entree	Id_département	Id_région	Id_manager	Id_Pos
1	20230203	1	2	3	10
2	20230204	3	5	4	112
3	20230204	4	6	3	202
4	20230205	4	8	6	110

CHAPITRE 1

MAITRISER LES FAITS

- 
1. Additivité
 2. Nulls
 3. Year-to-date
 4. Types des tables de faits
 5. Tables de faits sans faits
 - 6. Étapes de création d'une table de fait**
 7. Clé de substitution

1 - MAITRISER LES FAITS

Étapes de création d'une table de faits



Étapes de création d'une table de faits

Il y a des décisions clés qu'on doit prendre lorsqu'on veut concevoir les tables. Essentiellement, il y a 4 décisions principales. On les prend lorsqu'on répond aux questions sur nos besoins commerciaux.

1 - Identifier le processus métier réel qu'on veut analyser

- Exemples :
 - les ventes
 - Le traitement des commandes

2 - Définir le grain (le niveau de détails)

- Le niveau de détails qu'on prévoit dans la table
- Il s'agit simplement de définir à quoi un rôle/une transaction fait référence
- C'est une décision cruciale puisque le grain est très important dans les analyses
- Exemples :
 - Transaction, commande, ligne de commande
 - Périodes : journalier ou bien une combinaison entre la période et l'emplacement
- Il est recommandé d'opter pour un grain plus fin (un niveau de détails plus élevé) afin d'éviter les données pré-agrégées, de ne pas limiter le nombre d'analyses possibles et de maximiser le nombre de cas d'utilisations

1 - MAITRISER LES FAITS

Étapes de création d'une table de faits



Étapes de création d'une table de faits

Il y a décisions clés qu'on doit prendre lorsqu'on veut concevoir les tables. Essentiellement, il y a 4 décisions principales. On prend ces décisions lorsqu'on répond aux questions sur nos besoins commerciaux.

3 - Identifier les dimensions pertinentes


- En répondant aux questions quoi, quand, ou, comment et pourquoi ?
- Exemples :
 - Temps, emplacements, produits, clients, ...
- Ces dimensions offrent la possibilité de filtrer et grouper nos données
- Ce sont les points d'entrée pour notre analyse. On dit qu'elles sont l'âme de l'analyse des données

4 - Identifier les faits pour nos mesures

- Ils sont définis par le grain et non pas par un cas d'utilisation spécifique

CHAPITRE 1

MAITRISER LES FAITS

- 
1. Additivité
 2. Nulls
 3. Year-to-date
 4. Types des tables de faits
 5. Tables de faits sans faits
 6. Étapes de création d'une table de fait
 7. **Clé de substitution**

Clé primaire vs clé de substitution

- Afin de comprendre la définition de la clé de substitution, on va voir un exemple de la table de dimension des produits.
- On remarque dans cet exemple, que les identifiants des produits sont alphanumériques. Ils peuvent être donc assez volumineux et pas vraiment simples à manipuler. Ce n'est pas la manière la plus idéale dont on gère les processus dans un Data Warehouse par exemple.
- Ces clés sont nos clés naturelles qui proviennent du système source. Mais on peut aussi générer ce qu'on appelle des clés de substitution (ou des clés artificielles).
- Ce sont des clés artificielles qui sont différents des clés naturelles. Ils sont tout simplement des nombres entiers.
- Afin de les identifier, on ajoute généralement les suffixes PK (Primary Key) et FK (Foreign Key). Mais on peut toujours les identifier rapidement sans suffixe.
- Ces clés sont créées dans la base de données ou par un outil ETL.

Id_produit	Nom	Catégorie
PX12	Chili	Herbes
PL54	Ail	Fruits et légumes
AZ26	Banane	Fruits et légumes
TP32	Chocolat	Sucrés et snacks

PK_produit	Id_produit	Nom	Catégorie
1	PX12	Chili	Herbes
2	PL54	Ail	Fruits et légumes
3	AZ26	Banane	Fruits et légumes
4	TP32	Chocolat	Sucrés et snacks

Avantages

- La génération de ces clés ne demande pas beaucoup d'effort additionnel, mais elle a un ensemble d'avantages, à savoir:
 - Améliorer la performance, en ayant moins de stockage (un nombre entier au lieu d'une chaîne alphanumérique) et des jointures plus optimisées entre les faits et les dimensions.
 - Gérer les valeurs fictives : les valeurs manquantes peuvent être remplacées par exemple par un nombre négatif ou un nombre très grand afin de signaler qu'on a une indisponibilité de valeurs.
 - Intégrer différents systèmes sources : Dans le cas des données dupliquées dans différents sources qui utilisent les mêmes nombres, l'utilisation des clés de substitution facilite l'intégration de plusieurs éléments de sources.
 - Administrer et mettre à jours les données facilement.
- Parfois on ne trouve pas les clés naturelles, donc il est nécessaire d'auto générer ces clés en utilisant SQL ou bien un outil ETL.

1 - MAITRISER LES FAITS

Clé de substitution



Consignes pratiques

- Il est recommandé de toujours utiliser les clés de substitution au lieu des clés naturelles pour les clés primaires ou étrangères.
- L'utilisation de ces clés de substitution est recommandée dans les tables de faits et aussi les tables de dimensions sauf pour la table de dimension date. Dans cette dernière, on n'a pas besoin de générer ces nombres entiers car elle est très prévisible et on peut juste s'en tenir à notre date.
- Il est optionnel de garder les clés naturels. On peut les garder juste dans le cas du besoin.

CHAPITRE 2

MAITRISER LES DIMENSIONS

Ce que vous allez apprendre dans ce chapitre :

- Avoir une idée claire sur les dimensions
- Traiter le cas de la dimension date
- Traiter les nulls dans les dimensions
- Connaitre les hiérarchies dans les dimensions
- Comprendre les dimensions conformes
- Maitriser les dimensions dégénérées
- Maitriser les notions de Junk dimension et Role-playing dimension



20 heures



CHAPITRE 2

MAITRISER LES DIMENSIONS

1. Présentation

2. Dimension date
3. Nulls dans les dimensions
4. Hiérarchies dans les dimensions
5. Dimension conforme
6. Dimension dégénérée
7. Junk dimension
8. Role-playing dimension

Les tables de dimensions

- Comme nous avons déjà vu, une table de dimensions à toujours besoin d'une clé primaire. On remarque dans cet exemple qu'on a une clé naturelle qui vient du système source, mais ce n'est pas la bonne pratique de définir une clé primaire. On doit donc les modifier par des clés de substitution (un nombre entier auto incrémental).

Id_produit	Nom	Catégorie
P001	Lunettes SU-6	Accessoires
P002	Tablette chocolat 70% cacao	Sucreries
P003	Biscuits d'avoine	Sucreries



Produit_PK	Nom	Catégorie
1	Lunettes SU-6	Accessoires
2	Tablette chocolat 70% cacao	Sucreries
3	Biscuits d'avoine	Sucreries

Les tables de dimensions

- On peut garder les clés naturelles comme on peut les retirer parce que, généralement, ils ne sont pas nécessaires. Par contre on peut ajouter une table de correspondance qui sert juste d'associer une référence (clé naturelle) à chaque clé primaire de substitution.

Produit_PK	Id_produit
1	P001
2	P002
3	P003

- Mais la question qui se pose est : Comment créer la bonne référence à partir de notre table de faits lorsqu'on utilise des clés de substitution?

Les tables de dimensions

Ventes_PK	Client_FK	Produit_id
1001	312	P034
1002	312	P156
1003	312	P643

- On peut garder les deux clés naturelles et de substitution ou bien ne garder que la clé de substitution et se servir de la table de correspondance afin de récupérer la bonne référence à l'aide d'une jointure (JOIN/LEFT JOIN).

Produit_PK	Nom	Catégorie
1	Lunettes SU-6	Accessoires
2	Tablette chocolat 70% cacao	Sucreries
3	Biscuits d'avoine	Sucreries

Produit_PK	Id_produit
1	P001
2	P002
3	P003
...	...

Les tables de dimensions

- Par exemple : « **SELECT V.*, P.Produit_PK FROM ventes V LEFT JOIN Produits as P ON P.Produit_id = V.produit_id** » va afficher la table de ventes avec les références associées.

Ventes_PK	Client_FK	Produit_id	Produit_FK
1001	312	P034	34
1002	312	P156	156
1003	312	P643	643

- Les tables de dimensions servent aussi comme des tables de correspondance, ils n'ont pas généralement plusieurs lignes et ne se mettent pas à jour quotidiennement comme les tables de faits. Ils ont un ensemble de colonnes qui correspondent aux différents attributs descriptifs.
- La table de dimension est utilisée pour filtrer et grouper les données (par un attribut comme le nom du produit dans cet exemple). C'est le point d'entrée pour l'analyse des données. C'est pourquoi ces dimensions sont très importantes.

CHAPITRE 2

MAITRISER LES DIMENSIONS

- 
1. Présentation
 - 2. Dimension date**
 3. Nulls dans les dimensions
 4. Hiérarchies dans les dimensions
 5. Dimension conforme
 6. Dimension dégénérée
 7. Junk dimension
 8. Role-playing dimension

2 - MAITRISER LES DIMENSIONS

Dimension date



La dimension date

- La date est la dimension la plus utilisée. Elle est quasiment disponible dans tous les processus. Elle est parmi les aspects les plus importants dans les analyses dimensionnelles.
- Elle contient toutes les caractéristiques liées à la date qu'on veut analyser (par exemple : l'année, le mois (nom et chiffre), le jour (nom et chiffre), le trimestre, la semaine, etc.
- La dimension date a une caractéristique particulière par rapport au clé de substitution. Ce n'est pas une simple nombre auto incrémental, mais généralement c'est un nombre significatif. Il se compose de l'année, le mois et le jour. Par exemple la date 03/05/2023 est représentée par 03052023 ou 20230503.

2 - MAITRISER LES DIMENSIONS

Dimension date



La dimension date

- Généralement, on a une ligne supplémentaire dans la dimension date. Elle représente simplement une valeur fictive si jamais on aura pas une valeur date dans la table de faits associée, afin d'éviter les valeurs null dans les clés étrangères. Ça peut être par exemple une date comme 01/01/1990.
- Si l'heure est aussi un aspect important et la granularité du temps est à prendre en compte, on peut l'ajouter comme une autre dimension, sinon les attributs correspondants (heure, minutes, etc.) peuvent être ajoutés à la table date tout en ajoutant ces caractéristiques dans la clé de substitution (AnnéeMoisJourHeureMinutes).
- La date est l'une des rares dimensions calculables et donc très prévisibles. On peut la remplir à l'avance par des futures dates qui n'existent pas encore dans la table de faits.

2 - MAITRISER LES DIMENSIONS

Dimension date



La dimension date

- On doit toujours inclure les chiffres et les noms (textes). Par exemple le nom du mois Janvier correspond au mois numéro 1 de l'année. Il est aussi recommandé d'ajouter les noms dans ses formes longs et abrégés (Jan, Janvier), dépendamment du cas d'utilisation.
- On peut avoir des combinaisons des attributs (par exemple le trimestre + l'année : t2-2023)
- La dimension date peut être alimentée aussi par une date fiscale, qui ne commence pas forcément le 1^{er} janvier, comme la date de la rentrée scolaire.
- On peut aussi avoir quelques flags (Un flag est un objet indiquant si une valeur est vraie ou fausse). Par exemple un flag pour définir si une telle date est un weekend ou non.

2 - MAITRISER LES DIMENSIONS

Dimension date



La dimension date

Exemple

- Afin de bien comprendre la dimension Date, nous allons prendre un exemple de cette dernière.
- On a une clé primaire **Date_PK**, la date originale **Date** avec le format (yyyy-mm-jj par exemple), le mois dans ses deux formes (long **Mois** et short **Mois_short**), une combinaison de l'année et du trimestre **Année_trimestre**, le jour de la semaine et enfin un flag **Est_weekend** indiquant si la date courante est un weekend ou non (1 = oui et 0 = non).

Date_PK	Date	Mois	Mois_short	Année_trimestre	Année	Jour semaine	Est_weekend
20230420	20-04-2023	Avril	Avr	2023-S2	2023	Jeudi	0
20230421	21-04-2023	Avril	Avr	2023-S2	2023	Vendredi	0
20230422	21-04-2023	Avril	Avr	2023-S2	2023	Samedi	1

CHAPITRE 2

MAITRISER LES DIMENSIONS

- 
1. Présentation
 2. Dimension date
 - 3. Nulls dans les dimensions**
 4. Hiérarchies dans les dimensions
 5. Dimension conforme
 6. Dimension dégénérée
 7. Junk dimension
 8. Role-playing dimension

2 - MAITRISER LES DIMENSIONS

Nulls dans les dimensions



Nulls dans les dimensions

- Comme pour les faits, on va traiter le cas des nulls dans les dimensions. Nous avons vu que :
 - Les nulls dans les clés étrangères rompent l'intégrité référentielle. Donc toutes lignes qui présentent des nulls dans leurs clés étrangères vont être éliminées et ne peuvent pas apparaître dans les résultats des jointures.
 - Les nulls doivent être évités dans les clés étrangères. Ils peuvent être modifiés par des valeurs fictives (-1 par exemple).

Id_client	Nom_client	Id_commande	Nom_ligne_commande	Id_produit	Quantité	PU	Prix_réduit	Promo_FK	Montant_ventes	Cout_produit	date
14	Mohamed Amrani	521	Lunettes SU-6	12	2	220.99	220.99	Null-1	441.98	140.84	23/04/2023 13:34
14	Mohamed Amrani	521	Serviette de plage rouge	145	3	80.99	80.99	Null-1	242.97	40.87	23/04/2023 13:35
14	Mohamed Amrani	521	Maillot de bain bleu	234	1	160.99	140.99	3	140.97	120.53	23/04/2023 13:36

2 - MAITRISER LES DIMENSIONS

Nulls dans les dimensions



Nulls dans les dimensions

- Dans ce cas il faut garder une ligne dans la table de dimensions associée qui correspond à la valeur fictive ajoutée et ayant des valeurs des attributs compatibles avec leurs types (chaîne de caractères dans cet exemple).

Id_client	Nom_client	Id_commande	Nom_ligne_commande	Id_produit	Quantité	PU	Prix_réduit	Promo_FK	Montant_ventes	Cout_produit	date
14	Mohamed Amrani	521	Lunettes SU-6	12	2	220.99	220.99	Null -1	441.98	140.84	23/04/2023 13:34
14	Mohamed Amrani	521	Serviette de plage rouge	145	3	80.99	80.99	Null -1	242.97	40.87	23/04/2023 13:35
14	Mohamed Amrani	521	Maillot de bain bleu	234	1	160.99	140.99	3	140.97	120.53	23/04/2023 13:36

Promo_PK	Nom_Promo
1	Promo 1
2	Promo 2
3	Promo 3
-1	Pas de promo

2 - MAITRISER LES DIMENSIONS

Nulls dans les dimensions



Nulls dans les dimensions

- La présence des nulls dans une table de **faits** est possible, elle peut avoir un sens. Si par exemple on n'a pas de ventes pendant le weekend parce que les magasins sont fermés, une ligne pour chaque jour du weekend avec des nulls (au lieu des zéros) seront très significatives, car les zéros vont affecter le calcul de la somme ou de la moyenne des achats par exemple.
- Contrairement à la table de faits, la table de dimensions ne doit contenir aucun null. Tous les nulls doivent être remplacés par des valeurs descriptives (un texte significatif, une date, etc.), afin d'avoir une table lisible et significative permettant aux utilisateurs de décider eux même s'ils veulent faire apparaître ses valeurs ou non dans leurs analyses, graphes ou rapports.
- Les nulls n'apparaissent pas dans les résultats (graphes par exemple).

CHAPITRE 2

MAITRISER LES DIMENSIONS

1. Présentation
2. Dimension date
3. Nulls dans les dimensions
- 4. Hiérarchies dans les dimensions**
5. Dimension conforme
6. Dimension dégénérée
7. Junk dimension
8. Role-playing dimension



Hiérarchies dans les dimensions

- On a souvent des hiérarchies dans les dimensions. Quelle est donc la meilleure façon de les aborder, surtout qu'il y a quelques pièges dans lesquels on peut rencontrer?
- Dans les sources de données, ces dernières sont utilisées pour un traitement transactionnel, c'est pourquoi elles sont souvent normalisées.

Exemple : Produits et catégories sont deux tables de dimensions séparées et associées par une clé étrangères. Cette normalisation aide à gagner en terme d'espace de stockage et de performance.

Produit_nom	Catégorie_id
Lait	1
Imprimante	2
Serviette rouge	3
Serviette verte	3
Serviette bleue	3

Catégorie_id	Catégorie
1	Epiceries
2	Électroniques
3	Ménage

Hiérarchies dans les dimensions

- Mais si pour chaque relation (association) on ajoute une clé étrangère et une autre table de dimensions, on obtient un schéma en flocon. Alors qu'on a vu précédemment qu'il faut éviter ce type de schéma surtout dans les Data Warehouse, car on risque de perdre la bonne visibilité des données.
- C'est pourquoi la dénormalisation des données peut être une solution dans certains cas surtout lorsqu'on a une seule dimension dans la table de dimensions (exemple rassembler les tables produits et catégories dans une seule table de dimension aplatie sans avoir besoin d'ajouter des clés étrangères -> le résultat de jointure des deux tables).

Produit_nom	Catégorie
Lait	Epiceries
Imprimante	Électroniques
Serviette rouge	Ménage
Serviette verte	Ménage
Serviette bleue	Ménage

2 - MAITRISER LES DIMENSIONS

Hiérarchies dans les dimensions



Hiérarchies dans les dimensions

- On peut aussi combiner les attributs des différents niveaux d'hiérarchies dans une seule colonne (un seul attribut). Cette combinaison peut être utilisée facilement, et surtout si l'utilisateur en a besoin. Ceci est bénéfique aussi lorsqu'on a des valeurs dupliquées (le pays "**maroc**" par exemple est une valeur dupliquée).

Année-Trimestre
2022-Q3
2023-Q1
2023-Q2

Ville-Pays
Rabat-Maroc
Casablanca-Maroc
Paris-France

- En général, les hiérarchies doivent être, en général, combinées et rassemblées dans une seule table lorsque cela est possible, afin d'avoir moins de dimensions et plus de tables aplaties.

CHAPITRE 2

MAITRISER LES DIMENSIONS

- 
1. Présentation
 2. Dimension date
 3. Nulls dans les dimensions
 4. Hiérarchies dans les dimensions
 - 5. Dimension conforme**
 6. Dimension dégénérée
 7. Junk dimension
 8. Role-playing dimension

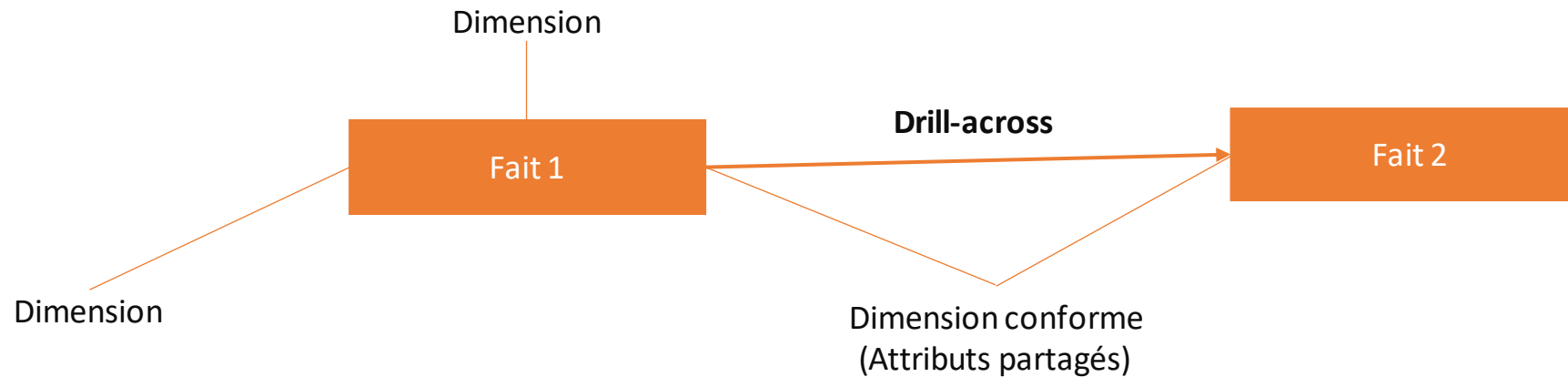
2 - MAITRISER LES DIMENSIONS

Dimension conforme



Dimensions conformes

- Une dimension conforme est une dimension utilisée dans plusieurs tables de faits / étoiles (pour partager les mêmes attributs).
- La date et l'heure sont les exemples des dimensions conformes les plus utilisés.
- L'idée derrière cette connexion est de pouvoir comparer plusieurs faits dans un seul rapport ou une seule analyse. Cela est appelé **Drill-across** ou **jointure**



2 - MAITRISER LES DIMENSIONS

Dimension conforme



Dimension conforme

Exemples :

- On veut utiliser une dimension date conforme. C'est-à-dire qu'on va utiliser un attribut partagé qui est disponible dans le coût et les ventes aussi, afin de comparer les coûts et les ventes (drill-across).
- Une autre table de dimension contenant le mois peut être une dimension conforme.

	Mois		Coût
	Janvier		50000
	Février		55000
	Mars		65000

	Mois		Ventes
	Janvier		68000
	Février		72000
	Mars		79000

2 - MAITRISER LES DIMENSIONS

Dimension conforme



Dimension conforme

- On parle d'une dimension conforme lorsqu'on a les mêmes attributs ou au moins un sous ensemble d'attributs utilisés par des faits différents. Cela peut être vu concrètement par l'utilisation de la même clé étrangère dans les différentes tables de faits (Date_FK par exemple).
- Ce n'est pas nécessaire d'avoir la même granularité. Par exemple dans la table de fait Coûts, chaque ligne représente un jour différent, par contre dans la table Ventes, le même jour peut figurer dans plusieurs lignes. Ceci rend une dimension conforme plus puissante.

Ventes_PK	Ventes	Date_FK
1	9500	20230101
2	7400	20230101
3	5300	20230102

Cout_PK	Coût	Date_FK
1	8100	20230101
2	5900	20230102
3	4200	20230103

- Même lorsque les tables de faits qui partagent une même dimension, utilisent différents formats de cette dernière (mois-année vs jour-mois-année), on parle toujours de dimension conforme et les deux tables peuvent être comparées.
- Résumé : les dimensions conformes sont utiles lorsqu'on a plus qu'une table de faits.

CHAPITRE 2

MAITRISER LES DIMENSIONS

1. Présentation
2. Dimension date
3. Nulls dans les dimensions
4. Hiérarchies dans les dimensions
5. Dimension conforme
- 6. Dimension dégénérée**
7. Junk dimension
8. Role-playing dimension



2 - MAITRISER LES DIMENSIONS

Dimension dégénérée



Dimension dégénérée

- Parfois on identifie une dimension qui n'est pas vraiment une dimension. Cette dimension n'a pas une table de dimension séparée, mais qui fonctionne comme une dimension. C'est une dimension dite **dégénérée**.
- Prenons l'exemple de la table de faits transactionnelle des ventes. On dispose de différentes transactions, avec différents montants qui peuvent être regroupés ensemble dans un seul paiement. Cette table est donc associée à une autre table de dimension « Paiements » avec la clé étrangère.

Transaction_PK	Montant	Paiement_FK_DD
1	550	234-032
2	560	234-032
3	650	234-033

Paiement_DD est
une dimension
dégénérée

Paiement_FK	Entête
234-032	Type A
234-033	Type A
234-034	Type B

Dimension dégénérée

- Mais parfois, tous les attributs de la table de dimensions (Entête) ont été déjà extraites par d'autres dimensions, et dans certains cas les valeurs de ces attributs ne sont pas vraiment importantes. Dans ce cas il ne reste que les clés primaires de cette table de dimension et donc la présence de cette dernière n'a aucune valeur ajoutée. Mais on peut toujours garder les clés étrangères dans la table, car ils peuvent être bénéfiques dans les analyses dans l'objectif par exemple de grouper les résultats par type de paiement.
- Dans ce cas on ne parle pas d'une clé étrangère, mais il faut indiquer explicitement qu'on a pas une dimension associée. Dans ce cas, on peut ajouter un suffixe **_DD** (Dimension Dégénérée) au lieu de **_FK**
- On peut trouver ce type de dimensions dans les faits transactionnels ou dans les attributs comme les numéros de commandes, de factures, ou tous les IDs qui avaient des informations déjà extraites.

Transaction_PK	Montant	Paiement_FK_DD
1	550	234-032
2	560	234-032
3	650	234-033

Paiement_DD est
une dimension
dégénérée

Paiement_PK	Entête
234-032	Type A
234-033	Type A
234-034	Type B

CHAPITRE 2

MAITRISER LES DIMENSIONS

1. Présentation
2. Dimension date
3. Nulls dans les dimensions
4. Hiérarchies dans les dimensions
5. Dimension conforme
6. Dimension dégénérée
- 7. Junk dimension**
8. Role-playing dimension



2 - MAITRISER LES DIMENSIONS

Junk Dimension



Junk dimension

- Comme on a déjà vu, on peut avoir des flags / indicateurs qui sont en fait des dimensions mais qui ne sont associés à aucune dimension donnée. Dans ce cas on peut créer ce qu'on appelle **junk dimension**(ou "dimension résiduelle" ou "dimension de détail").
- La dimension de genre « Junk dimension » est une dimension qui contient toutes sorte de flags, statuts, codes qui ne font partie à aucune dimension régulière.
- La table suivante est une table de faits transactionnelle qui contient quelques flags (Quel est le type du paiement? C'est quoi le type de la transaction : entrante ou sortante? Est-ce qu'elle associée à un bonus ou non?).

Transaction_PK	Montant	Paiement_type	Entrant/sortant	Est_bonus
1	550	Virement	Entrant	Oui
2	560	Carte crédit	Sortant	Non
3	650	Espèces	Entrant	Non

Junk dimension

Dans ce cas on a quelques options pour traiter ces attributs/flags dimensionnels:

- Les éliminer s'ils ne sont pas pertinents.
- Les garder comme ils sont dans la table de faits. Ils vont rester comme des valeurs dimensionnelles juste dans la table de faits, si on ne veut pas créer une table dimensions additionnelle. Mais dans le cas des valeurs textuelles très longues la table devient très volumineuse.
- Dans ce cas, on peut créer une dimension distincte pour chaque flag. Mais si on a déjà une table de faits très large, cela peut aussi augmenter la taille de cette table, ce qui n'est pas vraiment une solution idéale, par rapport à la performance et à l'ergonomie.
- Une autre alternative est la création de ce qu'on appelle les « junk dimension ». Une «junk dimension » est une dimension avec divers flags ayant une cardinalité inférieure (i.e. les choix ne sont pas si nombreux). On ne veut pas créer une table de dimensions séparée pour chaque flag, et donc on les met tous dans un seul box, et c'est vraiment le rôle d'un junk dimension.

2 - MAITRISER LES DIMENSIONS

Junk Dimension



Junk dimension

Exemple

Dans la table de faits précédente, on a vu qu'on a 3 flags : Paiement_type, entrant/sortant et Est_bonus. On peut donc remplacer ces flags par une clé étrangère faisant référence à la table de dimensions « junk dimension » créée. Cette dernière contient toutes les combinaisons possibles des flags. On a donc $3 * 2 * 2 = 12$ combinaison possibles.

Transaction_PK	Montant	Flag_transactionnel_FK
1	550	1
2	560	7
3	650	12

Flag_PK	Paiement_type	Entrant/sortant	Est_bonus
1	Virement	Entrant	Oui
2	Virement	Entrant	Non
3	Virement	Sortant	Oui
4	Virement	Sortant	Non
...

Junk dimension

Dans ce type de dimension, il faut faire attention au nombre de combinaison qui évolue exponentiellement avec le nombre de flags et de choix par flags. Avec 9 indicateurs ayant 4 choix chacun, on obtient $4^9 = 262144$ combinaisons, ce qui donne une table très grande. Dans ce cas, on peut essayer les alternatives suivantes :

- N'extraire que les combinaisons qui existent déjà dans la table de faits, sans prendre en compte les autres combinaisons possibles qui peuvent se produire. Cela peut avoir des risques, car il peut y avoir d'autres nouvelles combinaisons qui n'étaient pas produites avant.
- Créer 2 ou plus de tables junk dimension, en divisant ces flags en plusieurs sous ensembles de flags et créer une junk dimension pour chaque sous ensemble (exemple : au lieu d'avoir une seule table avec les 4^9 combinaisons, on peut créer trois tables avec 4^3 combinaisons chacun, c'est à dire 3 colonnes par tables).

CHAPITRE 2

MAITRISER LES DIMENSIONS

- 
1. Présentation
 2. Dimension date
 3. Nulls dans les dimensions
 4. Hiérarchies dans les dimensions
 5. Dimension conforme
 6. Dimension dégénérée
 7. Junk dimension
 - 8. Role-playing dimension**

Role-playing Dimension

- Role-playing dimension (ou "dimension à rôles multiples" ou "dimension avec plusieurs rôles") est une dimension qui est référencée dans une table de faits plusieurs fois. La date est l'exemple le plus populaire d'une dimension à rôles multiples.

Id_commande	Date_commande_FK	Nb_produits	produit_FK	Début_production_FK
1	20230203	100	35	20230204
2	20230204	100	35	20230204
3	20230204	100	35	20230204
4	20230205	100	35	20230206
5	20230205	100	35	20230208

- L'exemple des commandes illustre bien le concept du role-playing dimension. On a une table de faits avec deux clés étrangères dates ici (date_commande et début_production) avec une seule table de dimension date.

Date_PK	Date	Mois	Mois_short	Année_trimestre	Année	Jour semaine	Est_weekend
20230420	20-04-2023	Avril	Avr	2023-S2	2023	Jeudi	0
20230421	21-04-2023	Avril	Avr	2023-S2	2023	Vendredi	0
20230422	21-04-2023	Avril	Avr	2023-S2	2023	Samedi	1

2 - MAITRISER LES DIMENSIONS

Role-playing Dimension



Role-playing Dimension

- Ce n'est pas nécessaire de dupliquer la dimension date physiquement dans la base de données pour chaque clé étrangère ou référence. On peut utiliser la même table avec différents rôles.
- On crée des associations / relations avec la même table avec des manières différentes. On utilise pour chaque association la clé étrangère appropriée. Ces associations identifient les différents rôles de la dimension (Rôle 1 : date de commande, Rôle 2 : date de production).

Id_commande	Date_commande_FK	Nb_produits	produit_FK	Début_production_FK
1	20230203	100	35	20230204
2	20230204	100	35	20230204
3	20230204	100	35	20230204
4	20230205	100	35	20230206
5	20230205	100	35	20230208

Rôle 1

Rôle 2

Date_PK	Date	Mois	Mois_short	Année_trimestre	Année	Jour semaine	Est_weekend
20230420	20-04-2023	Avril	Avr	2023-S2	2023	Jeudi	0
20230421	21-04-2023	Avril	Avr	2023-S2	2023	Vendredi	0

Role-playing Dimension

- Après l'identification des différents rôles, on peut par exemple analyser les commandes reçues (avec le premier rôle) et les productions lancées (avec le deuxième rôle) pour des périodes données (par mois par exemple).

Mois	Produits (Commandes reçues)
Janvier	2500
Février	2700
Mars	2800
...	...

Mois	Produits (productions lancées)
Janvier	2650
Février	2600
Mars	2740
...	...

- Toutes les analyses peuvent se faire en utilisant les outils d'analyse comme Tableau, Power BI, etc. ou bien avec des requêtes SQLs.
- En utilisant SQL, on peut créer une vue pour chaque rôle, afin de permettre une analyse des données sans avoir besoin de dupliquer les données. Les vues créées peuvent être manipulées exactement comme des tables, cela permet de créer autant de jointures avec ces vues.

CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Ce que vous allez apprendre dans ce chapitre :

- Comprendre la notion de Slowly Changing Dimension
- Connaitre les types (Type 0, 1, 2 et 3)



05 heures



CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

1. Introduction

2. Type 0 : Original
3. Type 1 : Ecrasement
4. Type 2 : Nouvelle ligne
5. Type 1 & Type 2
6. Type 3 : Attributs supplémentaires



Slowly Changing Dimension

- Jusqu'à présent, on a juste supposé, d'une certaine manière qu'il n'y aurait pas de changements ou de modifications dans les dimensions. En effet, les dimensions sont plutôt statiques par rapport aux faits. Mais dans la réalité, les dimensions peuvent subir à quelques changements (quelques attributs peuvent changés). Il y a une stratégie mise en place afin de permettre la gestion de ces changements. Mais avant d'entamer ces stratégies, il faut cerner ces changements :

Être proactif

- Poser des questions sur les changements potentiels.

Développer une stratégie pour chaque changement d'attribut.

- En fonction de la situation et des besoins, on peut distinguer entre différents types de dimensions à évolution lente ou Slowly Changing Dimension (SCD).
- Une dimension à évolution lente est un terme très populaire en informatique décisionnelle. Les **types** de ces SCD ont été introduites par Kimbal en 1995.

CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

1. Introduction
2. **Type 0 : Original**
3. Type 1 : Ecrasement
4. Type 2 : Nouvelle ligne
5. Type 1 & Type 2
6. Type 3 : Attributs supplémentaires



Type 0 : Original

- Dans ce type 0, on conserve les données d'origine comme elles sont. Cela est applicable lorsqu'il n'y a pas de changements dans les dimensions concernées.
- Il faut donc être très sûr qu'il n'y a pas de changements dans ces dimensions. Et dans ce cas on n'a pas besoin d'appliquer aucune stratégie.
- Cela est généralement applicable sur la dimension date, sauf que dans des cas, la table contient des attributs, comme les jours fériés de l'entreprise qui peuvent subir à des modifications de temps à autre.
- Dans ce type de SCD, il peut y avoir de nombreux attributs qui sont simplement appelés ou étiquetés par 'Original'. Comme par exemple « le nom original du produit » ou n'importe quel attribut qu'on sait qu'il va rester comme il est sans aucun changement.
- C'est l'option la plus simple à maintenir, puisqu'il n'y aura aucun changement à faire sur la dimension après.

CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

1. Introduction
2. Type 0 : Original
- 3. Type 1 : Ecrasement**
4. Type 2 : Nouvelle ligne
5. Type 1 & Type 2
6. Type 3 : Attributs supplémentaires



Type 1 : Ecrasement

- Dans la réalité, de nombreux attributs des dimensions sont sujets à des changements. On veut que ces changements soient visibles dans ces dimensions et que l'utilisateur voit clairement ces mises à jour.
- Cela nous amène à au premier type du SCD, qui est l'écrasement des dimensions à évolution lente.
- Dans ce cas, les anciennes valeurs sont simplement écrasées et modifiées par les nouvelles valeurs. On n'a donc dans la dimension que son état actuel.

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 1

Type 1 : Ecrasement

Exemple

- On a une table de produits et quelques changements dans le nom et la catégorie d'un produit. Dans ce cas, on ne fait qu'écraser l'ancienne valeur et la remplacer par la nouvelle et ne garder que la table mise à jour.

Produit_PK	Nom	Catégorie
1	Lunettes SU-6	Accessoires
2	Tablette chocolat 70% cacao	Sucreries
3	Biscuits d'avoine	Sucreries



Produit_PK	Nom	Catégorie
1	Lunettes SU-6	Accessoires
2	Tablette chocolat 70% cacao	Sucreries
3	Biscuits d'avoine délicieux	Biscuits

- Ce type de SCD est très simple à implémenter puisqu'on ne fait que la mise à jour des valeurs.
- La table de fait n'est pas affectée par ces modifications.

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 1



Type 1 : Ecrasement

- Même si le Type 1 du SCD est simple mais on a quelques problèmes avec ce dernier :
 - Pas d'historique des mises à jour
 - Si les mises à jour ne sont pas assez importantes (le changement du nom du produit), les analyses ne seront pas affectées et l'historique peut être ignoré
- Dans certains cas, les mises à jour sont significatives (la modification de la catégorie par exemple), il faut être un peu prudent car ce type de changement peut affecter les analyses et les rapports existants.

Vente_PK	Nom	Montant
1	Lunettes SU-6	300
2	Tablette chocolat 70% cacao	30
3	Biscuits d'avoine	40
4	Tablette chocolat 70% cacao	30
5	Biscuits d'avoine	40

Avant mise à jour de
la catégorie

Après mise à jour de
la catégorie

Catégorie	Montant
Accessoires	300
Sucreries	140

Catégorie	Montant
Accessoires	300
Sucreries	60
Biscuits	80

CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

1. Introduction
2. Type 0 : Original
3. Type 1 : Ecrasement
- 4. Type 2 : Nouvelle ligne**
5. Type 1 & Type 2
6. Type 3 : Attributs supplémentaires



3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2



Type 2 : Nouvelle ligne

- Le type 2 du SCD est le plus puissant.
- On a vu que l'historique n'est pas prise en compte dans le Type 1, ce qui peut être problématique surtout dans les analyses et les rapports. On a vu dans l'exemple précédent (tableau ci-dessous), que les deux lignes 3 et 5 sont considérées dans l'analyse faite par catégorie, sachant que c'est seulement la ligne 5 qui est nouvelle par rapport à la dernière mise à jour de la catégorie du produit 'Biscuits d'avoine'. Ceci ne respecte pas l'historique.

Vente_PK	Nom	Montant
1	Lunettes SU-6	300
2	Tablette chocolat 70% cacao	30
3	Biscuits d'avoine	40
4	Tablette chocolat 70% cacao	30
5	Biscuits d'avoine	40

Après mise à jour de
la catégorie



Catégorie	Montant
Accessoires	300
Sucreries	60
Biscuits	80

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2



Type 2 : Nouvelle ligne

- Si on tenait compte de l'historique, l'achat 3 ne doit pas apparaître dans la catégorie 'biscuits mais plutôt dans les sucreries. Le Type 2 du SCD résout ce problème.
- Avec ce type, on peut partitionner et segmenter notre historique. Les anciennes valeurs peuvent restées associées à l'ancienne catégorie et les nouvelles avec la nouvelle catégorie.

Catégorie	Montant
Accessoires	300
Sucreries	100
Biscuits	40

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2



Type 2 : Nouvelle ligne

- Le Type 2 est très utilisé (le plus utilisé) dans les Data Warehouses, car l'historique est très importante des ces derniers.
- Contrairement au type 1, on ne modifie pas les anciennes valeurs mais on ajoute **une nouvelle ligne** (avec sa nouvelle clé primaire) à chaque fois qu'on a des mises à jour.

Produit_PK	Nom	Catégorie
1	Lunettes SU-6	Accessoires
2	Tablette chocolat 70% cacao	Sucreries
3	Biscuits d'avoine	Sucreries



Produit_PK	Nom	Catégorie
1	Lunettes SU-6	Accessoires
2	Tablette chocolat 70% cacao	Sucreries
3	Biscuits d'avoine	Sucreries
4	Biscuits d'avoine	Biscuits

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2

Type 2 : Nouvelle ligne

- Cela affecte la table de faits, en utilisant la nouvelle clé étrangère (4 par exemple), à partir du moment du changement au niveau de la dimension associée. Avec cette solution, on a pas besoin de mettre à jour la table de faits et on peut avoir le bon résultat des analyses.

Vente_PK	Nom	Produit_FK	Montant
1	Lunettes SU-6	1	300
2	Tablette chocolat 70% cacao	2	30
3	Biscuits d'avoine	3	40
4	Tablette chocolat 70% cacao	2	30
5	Biscuits d'avoine	4	40



Catégorie	Montant
Accessoires	300
Sucreries	100
Biscuits	40

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2



Type 2 : Nouvelle ligne

- La question qui se pose, dans ce type de SCD, est “Comment calculer le nombre de produits?”
- Les clés naturelles dans ce cas sont très importantes, il faut donc les garder, afin de compter les valeurs distinctes de ces dernières (Id_Produit).

Produit_PK	Id_Produit	Nom	Catégorie
1	P001	Lunettes SU-6	Accessoires
2	P002	Tablette chocolat 70% cacao	Sucreries
3	P003	Biscuits d’avoine	Sucreries
4	P003	Biscuits d’avoine	Biscuits

- La solution des clés naturelles est bonne pour le calcul du nombre d’article dans une dimension, mais on peut toujours avoir quelques objections. Par exemple “comment connaître la liste de tous les produits courants?”. Il faut donc implémenter quelques stratégies afin d’administrer ce type de SCD.

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2



Administration du Type 2

- Afin de pouvoir administrer le Type 2 du SCD et l'utiliser efficacement, il faut ajouter des attributs supplémentaires.
- Afin de pouvoir identifier la valeur courante (par exemple le nom du produit courant), on doit ajouter la date effective et la date d'expiration afin de définir la période de validité des valeurs. Pour la date d'expiration, il faut toujours la mettre afin d'éviter les nulls. Il est préférable d'utiliser une date dans le future (01-01-2100) au lieu de mettre des valeurs nulls, afin d'éviter les conflits liés aux valeurs nulls qu'on a déjà vu.
- Ces deux colonnes sont nécessaires afin d'utiliser les bonnes clés étrangères dans la table de faits.

Produit_PK	Id_Produit	Nom	Catégorie	Date_eff	Date_exp
1	P001	Lunettes SU-6	Accessoires	01-01-2023	01-01-2100
2	P002	Tablette chocolat 70% cacao	Sucreries	01-01-2023	01-01-2100
3	P003	Biscuits d'avoine	Sucreries	01-01-2023	08-05-2023
4	P003	Biscuits d'avoine Délicieux	Biscuits	09-05-2023	01-01-2100

- Il faut bien noter qu'il faut utiliser dans ce Type 2, les clés de substitution (numérique), car les clés naturelles peuvent ne pas être uniques. Mais comment peut-on s'assurer d'utiliser la bonne clé étrangère dans la table de faits? On utilise la clé naturelle afin de trouver les lignes correspondantes et les dates effective et d'expiration afin de choisir la bonne ligne est donc la bonne clé étrangère.

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 2



Administration du Type 2

- On peut aussi utiliser une autre manière afin d'administrer ce SCD, en utilisant un flag **est_courant**. C'est une colonne qui indique si la valeur est courante ou non.

Produit_PK	Id_Produit	Nom	Catégorie	Date_eff	Date_exp	Est_courant
1	P001	Lunettes SU-6	Accessoires	01-01-2023	01-01-2100	Oui
2	P002	Tablette chocolat 70% cacao	Sucreries	01-01-2023	01-01-2100	Oui
3	P003	Biscuits d'avoine	Sucreries	01-01-2023	08-05-2023	Non
4	P003	Biscuits d'avoine Délicieux	Biscuits	09-05-2023	01-01-2100	Oui

- Pour quelques attributs, comme le nom du produit par exemple, on n'a pas vraiment besoin d'utiliser ces méthodes et même ce Type 2 du SCD. Cela soulève la question : est ce qu'on peut combiner les types 1 et 2 dans la même table de dimension et qu'est ce que ça va donner come résultats?

CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

1. Introduction
2. Type 0 : Original
3. Type 1 : Ecrasement
4. Type 2 : Nouvelle ligne
- 5. Type 1 & Type 2**
6. Type 3 : Attributs supplémentaires



3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 1 & Type 2



Combinaison de Type 1 et Type 2

- Pour une table de dimensions donnée, on n'a pas à décider quel type à utiliser Type 1 ou Type 2, mais ça dépend des attributs à mettre à jour.
- Par exemple, pour le nom d'un produit, ce n'est pas nécessaire d'avoir toute l'historique et donc le Type 1 peut être suffisant. Par contre la mise à jour de la catégorie a un effet sur les analyses, il faut donc garder son historique en utilisant le Type 2.
- On peut donc utiliser les 2 types dans la même table de dimension. L'utilisation du type dépend de l'importance de l'historique par rapport à l'attribut à changer. Ce n'est pas une décision technique à prendre auparavant. La décision se prend au moment de la mise à jour.

	Produit_PK	Id_Produit	Nom	Catégorie	Date_eff	Date_exp
	1	P001	Lunettes SU-6	Accessoires	01-01-2023	01-01-2100
	2	P002	Tablette chocolat 70% cacao	Sucreries	01-01-2023	01-01-2100
Type 1 →	3	P003	Biscuits d'avoine Délicieux	Sucreries	01-01-2023	08-05-2023
Type 2 →	4	P003	Biscuits d'avoine Délicieux	Biscuits	09-05-2023	01-01-2100

- Dans certaines situations, Type 1 et Type 2 ne sont pas les bon choix, on a donc une autre alternative qui est le Type 3 du SCD.

CHAPITRE 3

APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

1. Introduction
2. Type 0 : Original
3. Type 1 : Ecrasement
4. Type 2 : Nouvelle ligne
5. Type 1 & Type 2
6. **Type 3 : Attributs supplémentaires**



3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 3



Type 3 : Attributs supplémentaires

- Le Type 3 est un peu entre les deux types Type 1 et Type 2. Elle permet de basculer entre les versions.
- Au lieu d'ajouter une nouvelle ligne, on ajoute une nouvelle colonne (un nouvel attribut). Cette nouvelle colonne contient l'ancienne valeur de l'attribut mis à jour. La colonne modifiée et celle ajoutée reflètent les deux différents états de l'attribut (après et après mises à jour).

Produit_PK	Id_Produit	Nom	Catégorie	Catégorie_précédante
1	P001	Lunettes SU-6	Accessoires	Accessoires
2	P002	Tablette chocolat 70% cacao	Sucreries	Sucreries
3	P003	Biscuits d'avoine	Biscuit	Sucreries


3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 3

Type 3 : Attributs supplémentaires

- Le Type 3 permet de non seulement utiliser la nouvelle valeur dans les analyses mais on peut aussi basculer à l'ancienne valeur et agréger les données avec cette dernière.

Catégorie	Montant
Accessoires	300
Sucreries	60
Biscuits	80



Catégorie	Montant
Accessoires	300
Sucreries	140

- Ce type est généralement utilisé lorsqu'on a un nombre important de modifications planifiées à s'exécuter en un seul moment (exemple : les restructurations dans les entreprises, les changements des statuts, etc.).

3 - APPRÉHENDER LES DIMENSIONS À ÉVOLUTION LENTE

Type 3



Type 3 : Attributs supplémentaires

- Les nouvelles enregistrements (nouveau produit par exemple) sont tout simplement ajoutés via des nouvelles lignes comme d'habitude, sauf que dans la nouvelle colonne ajoutée, qui reflète l'historique, on n'associe aucune valeur à la nouvelle ligne ajoutée. On peut mettre une valeur juste pour éviter les nulls mais qui signifie que c'est une nouvelle valeur.

Produit_PK	Id_Produit	Nom	Catégorie	Catégorie_precedante
1	P001	Lunettes SU-6	Accessoires	Accessoires
2	P002	Tablette chocolat 70% cacao	Sucreries	Sucreries
3	P003	Biscuits d'avoine	Biscuit	Sucreries
4	P004	Assiette blanche	Cuisine	Non applicable

- On n'a pas accès à seulement deux versions, mais il est possible d'ajouter autant de colonnes que de versions si cela est nécessaire.
- Le Type 3 n'est pas vraiment la bonne solution dans le cas des changements fréquents et imprévisibles. Il est applicable pour une situation bien spécifique (une restructuration dans un moment donné par exemple). Dans ce cas c'est le Type 2 du SCD le plus convenable. Sinon, si les changements sont mineurs, il faut adopter le Type 1.