

Data Warehousing et modélisation dimensionnelle

Plan du cours

1. Introduction aux concepts

- ☐ Objectifs de la Business Intelligence
- ☐ Objectifs d'une solution de Business Intelligence/Data Warehouse
- ☐ Objectifs de la modélisation dimensionnelle

2. Modélisation dimensionnelle

- ☐ Star schema
- ☐ Fait
- ☐ Table de faits
- ☐ Table de dimensions
- ☐ Bonnes pratiques
- ☐ Mesures numériques vs dimensions numériques
- ☐ Hiérarchies

Plan du cours

3. Architecture d'une solution BI/DW

- ☐ Back room
- ☐ Front room
- ☐ Architecture Kimball classique
- ☐ Architecture Data mart
- ☐ Architecture Hub-and-spoke
- ☐ Architecture Hybride Kimball/Hub-and-spoke

Plan du cours

Case study 1 : Vente au détail

- ☐ Etapes de modélisation
- ☐ Faits dérivés
- ☐ Faits non-additifs
- ☐ Table de faits transactionnelle
- ☐ Dimensions
 - ☐ Dimension date
 - ☐ Dimension produit
 - ☐ Dimension magasin
 - ☐ Dimension promotion
- ☐ Valeurs nulles
- ☐ Tables de faits sans fait
- ☐ Clés artificielles et naturelles
- ☐ Snowflake Schema
- ☐ Exercice : Modélisation d'une table de faits et des dimensions associées

Plan du cours

Case study 2 : Inventorisation

- ☐ Chaîne de valorisation
- ☐ Importance de l'inventorisation
- ☐ Snapshot périodique
- ☐ Inventaire transactionnel
- ☐ Snapshot cumulé
- ☐ Bus Matrix et dimensions communes

Plan du cours

Case study 3 : Approvisionnement

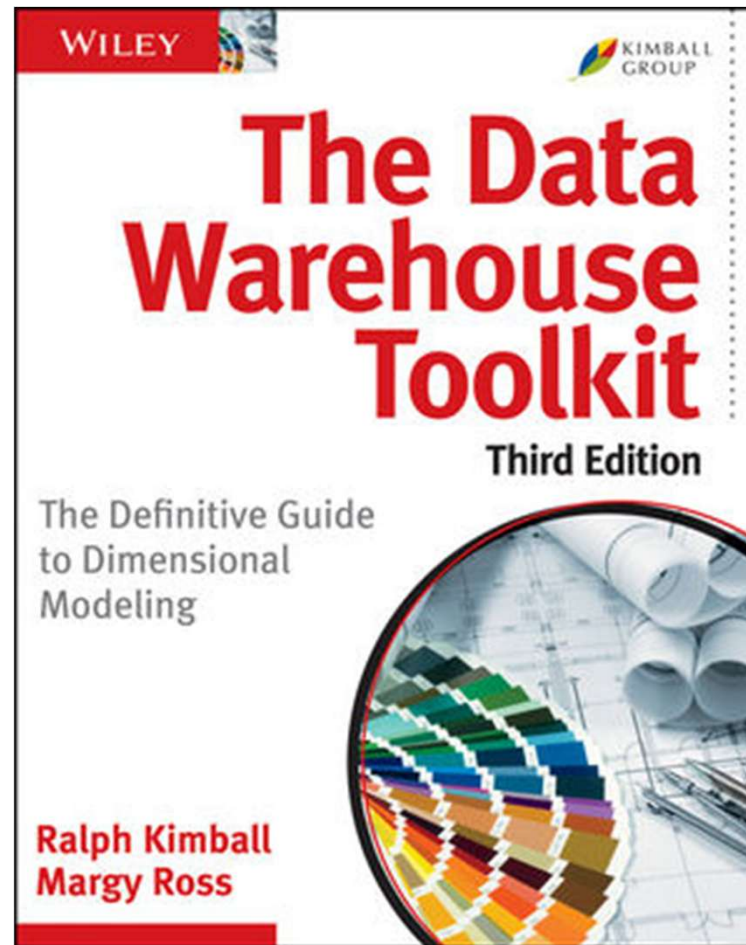
- ❑ Exemple de table de faits transactionnelle
- ❑ Dimensions à variation lente
 - ❑ Cas 1 : Réécriture
 - ❑ Cas 2 : Ajout d'une entrée
 - ❑ Cas 3 : Ajout d'une colonne
 - ❑ Cas 4 : Création d'une mini-dimension

Plan du cours

Case study 4 : Gestion des commandes

- ❑ Chaîne de valorisation
- ❑ Commandes
 - ❑ Etapes de modélisation
 - ❑ Dimensions à rôles multiples
 - ❑ Dimension produit
 - ❑ Dimension client
 - ❑ Dimension contrat
 - ❑ Pattern à éviter
- ❑ Facturation
 - ❑ Etapes de modélisation

Références



Data Warehousing et modélisation dimensionnelle

Introduction aux concepts



Introduction aux concepts

Objectifs de la Business Intelligence

- ❑ Les processus des entreprises génèrent des quantités de plus en plus importantes de données
- ❑ Ces données peuvent provenir aussi de l'opérationnel que de l'extérieur
- ❑ La business intelligence peut se définir comme étant l'ensemble des processus mis en place pour **transformer des données brutes en information directement exploitable** par les preneurs de décisions

Introduction aux concepts

Mon data provient de plusieurs sources => Je n'arrive pas à le consolider

- ❑ raisons possibles :
 - ❑ Manque de connaissance en matière de croisement de data (principe des jointures)
 - ❑ Mise en correspondance difficile entre les différents systèmes
 - ❑ La structure de mon data ne me permet pas de le croiser facilement
 - ❑ Mes data sont sur des support différents
 - ❑ Mon data est trop sale pour être facilement croiser

Introduction aux concepts

Problématiques

- ❑ L'extraction de l'information à partir des données brutes de l'entreprises va s'accompagner de problématiques récurrentes

Introduction aux concepts

Mon data provient de plusieurs sources => Il est difficile d'établir une et une seule vérité

- ❑ Vos informations concernant vos employés ou vos clients se trouvent peut-être dans plusieurs bases de données. Laquelle est la plus fiable, laquelle est la plus à jour? Laquelle peut servir de référence aux autres.
- ❑ Beaucoup de sociétés travaillent avec des extractions en excel. Qui a la bonne version des données ?

Introduction aux concepts

Mon data provient de plusieurs sources

- ❑ Si plusieurs personnes essaient de répondre à la même question business, il se peut que les résultats soient différents. La multiplicité des sources rend difficile l'obtention d'un résultat cohérent.
- ❑ L'absence d'un modèle unifié empêche également d'intégrer des logiques business uniformes à certains KPIs. Un modèle unifié tend à essayer d'inclure ces logiques business dans le modèle.

Introduction aux concepts

Mon data provient de l'opérationnel, je n'arrive pas à le comprendre :

- ❑ La modélisation des systèmes opérationnels est très souvent une modélisation relationnelle. Par définition, c'est une modélisation qui va découpler les différentes entités.
- ❑ La multiplicité des entités (tables) rend extrêmement compliqué la compréhension des bases de données.
 - ❑ source d'erreurs de requêtage
 - ❑ Multiplicité = jointures peu performantes

Introduction aux concepts

Mon data provient de l'opérationnel, le requêtage des bases de données est très gourmand :

- ❑ Lié à la problématique précédente : la multiplicité des entités oblige à effectuer de nombreuses jointures coûteuses en ressources

Introduction aux concepts

Mon data provient de l'opérationnel, l'interroger impacte celui-ci

- ❑ L'opérationnel est ce dont nous avons besoin pour travailler au quotidien
- ❑ L'interroger pour du reporting revient à lui assigner une charge de travail supplémentaire

Introduction aux concepts

Mon data provient de l'opérationnel, cette structure est adaptée au besoin opérationnel, il ne l'est pas forcément pour le reporting

- ❑ Ma base de données opérationnelle n'archive pas certaines informations nécessaires à mon reporting. (Principe d'historisation)
- ❑ Que faire ? Me sera t-il impossible d'obtenir des réponses ?
- ❑ Dans le même ordre d'idée, le data nécessaire à la résolution de mes questions n'est peut-être pas existant dans mon opérationnel ou en tout cas pas directement (doit parfois être précalculé)

Introduction aux concepts

Business et IT : se comprendre

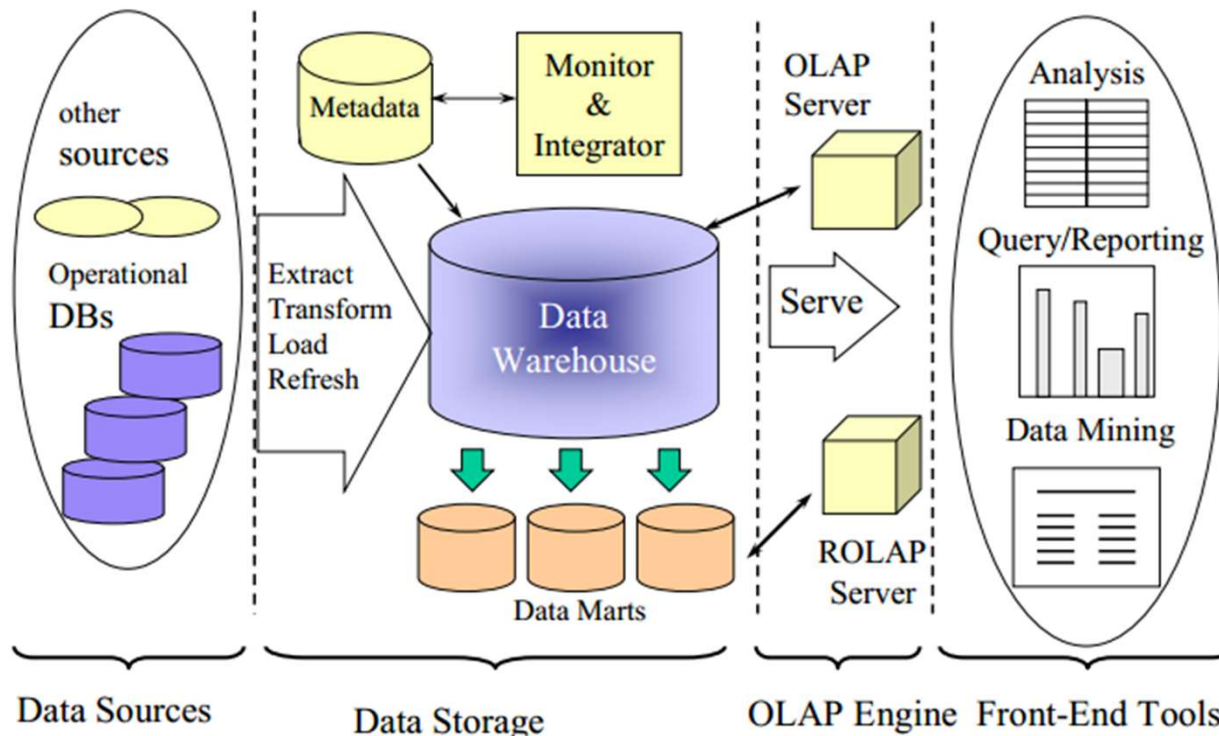
- ❑ Dans beaucoup d'entreprises, le business travaille avec des extraits de données fourni par l'IT.
- ❑ Nécessité pour l'IT de bien comprendre le besoin business afin de répondre avec des extraits corrects
- ❑ Nécessité pour le Business de bien expliciter son besoin à l'IT

Introduction aux concepts

Business et IT : le temps

- ❑ Pour les business travaillant avec des extraits de données fournis par l'IT, dépendant de la structure de l'entreprise, il n'est pas toujours possible de fournir des extraits de données rapidement.
- ❑ De plus travailler avec des extraits ne facilite pas l'intégration des données de manière continue dans le processus de reporting
- ❑ Un modèle unifié rafraîchi au besoin permettrait un gain de temps conséquent

Introduction aux concepts



Introduction aux concepts

Objectifs d'une solution Business Intelligence/Data Warehousing (BI/DW)

- ☐ Fournir de l'information intelligible aux utilisateurs métiers
- ☐ Centraliser et uniformiser les données provenant des différentes sources
- ☐ Fournir l'information dans des délais appropriés aux besoins métiers
- ☐ Protéger, contrôler l'accès à l'information
- ☐ Mettre à disposition des utilisateurs de l'information de qualité afin que leurs décisions puissent être prises en toute confiance
- ☐ Être adaptée aux utilisateurs métiers en terme d'utilisation

Introduction aux concepts

Le designer d'une solution doit:

- ☐ Comprendre les utilisateurs métiers
 - ☐ Comprendre leurs responsabilités et objectifs
 - ☐ Comprendre les questions des utilisateurs auxquelles doit répondre une solution BI/DW
- ☐ Adapter l'information aux utilisateurs métiers
 - ☐ Les applications doivent être pensées **pour les utilisateurs** en terme d'interface et de compréhension
- ☐ Penser la solution sur le long terme
 - ☐ La solution doit survivre aux changements métiers
 - ☐ Être maintenable
 - ☐ Conserver la confiance des utilisateurs

Introduction aux concepts

Objectifs de la modélisation dimensionnelle (vs modélisation relationnelle)

- ❑ Les bases dimensionnelles diffèrent des bases relationnelles par leur degré de normalisation
- ❑ 2 objectifs :
 - ❑ Simplifier les structures relationnelles à des fins de compréhension
 - ❑ *“Make everything as simple as possible, but not simpler” A. Einstein*
 - ❑ Améliorer les performances lors des requêtes
 - ❑ Moins de jointures = plus de performances

Introduction aux concepts

Objectifs de la modélisation dimensionnelle

- ❑ Ci-dessous, une requête visant à récupérer le montant des ventes pour le produit “All-purpose Bike Stand” au Canada et le regrouper par client dans une **base de donnée relationnelle**

```
select sum(LineTotal) as Total, C.CustomerID
from Sales.SalesOrderDetail as OD
    left join Sales.SalesOrderHeader as OH on OH.SalesOrderID = OD.SalesOrderID
    left join Sales.SpecialOfferProduct as OP on OP.ProductID = OD.ProductID
    left join Production.Product as P on P.ProductID = OP.ProductID
    left join Sales.Customer as C on C.CustomerID = OH.CustomerID
    left join Sales.SalesTerritory ST on ST.TerritoryID = C.TerritoryID
    left join Person.CountryRegion as CR on CR.CountryRegionCode = ST.CountryRegionCode
where OH.OnlineOrderFlag = 1 and P.Name = 'All-Purpose Bike Stand' and CR.Name = 'Canada'
Group by C.CustomerID
```

Introduction aux concepts

Objectifs de la modélisation dimensionnelle

- ❑ La même requête dans une **base de données dimensionnelle**

```
select sum(S.SalesAmount) as Total, C.CustomerKey
from FactInternetSales as S
  left join DimProduct as P on P.ProductKey = S.ProductKey
  left join DimCustomer as C on C.CustomerKey = S.CustomerKey
  left join DimGeography as G on C.GeographyKey = G.GeographyKey
where P.EnglishProductName = 'All-Purpose Bike Stand' and g.FrenchCountryRegionName = 'Canada'
group by C.CustomerKey
```

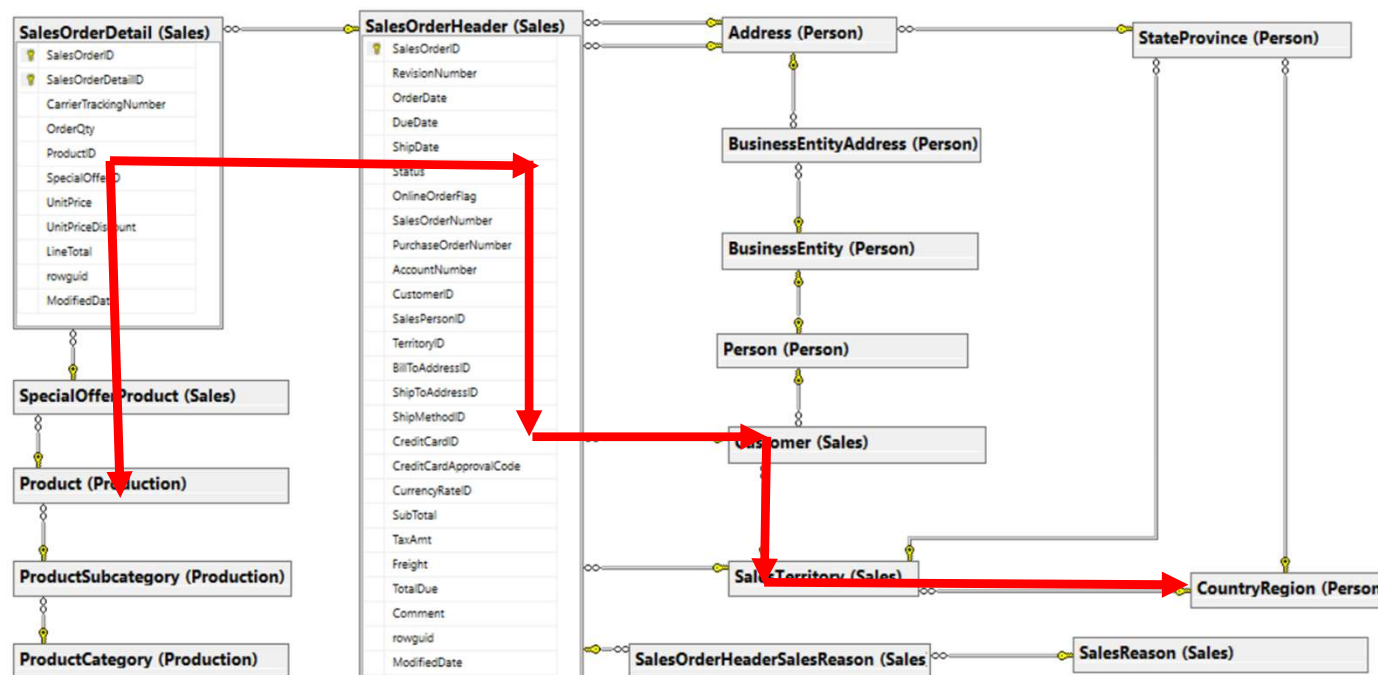
Introduction aux concepts

Objectifs de la modélisation dimensionnelle

- ❑ Bilan :
 - ❑ 3 jointures vs 6 jointures
 - ❑ clause “where” double vs clause where “triple”
 - ❑ => Moins de performance et risque d’erreur accru
 - ❑ Mais surtout...

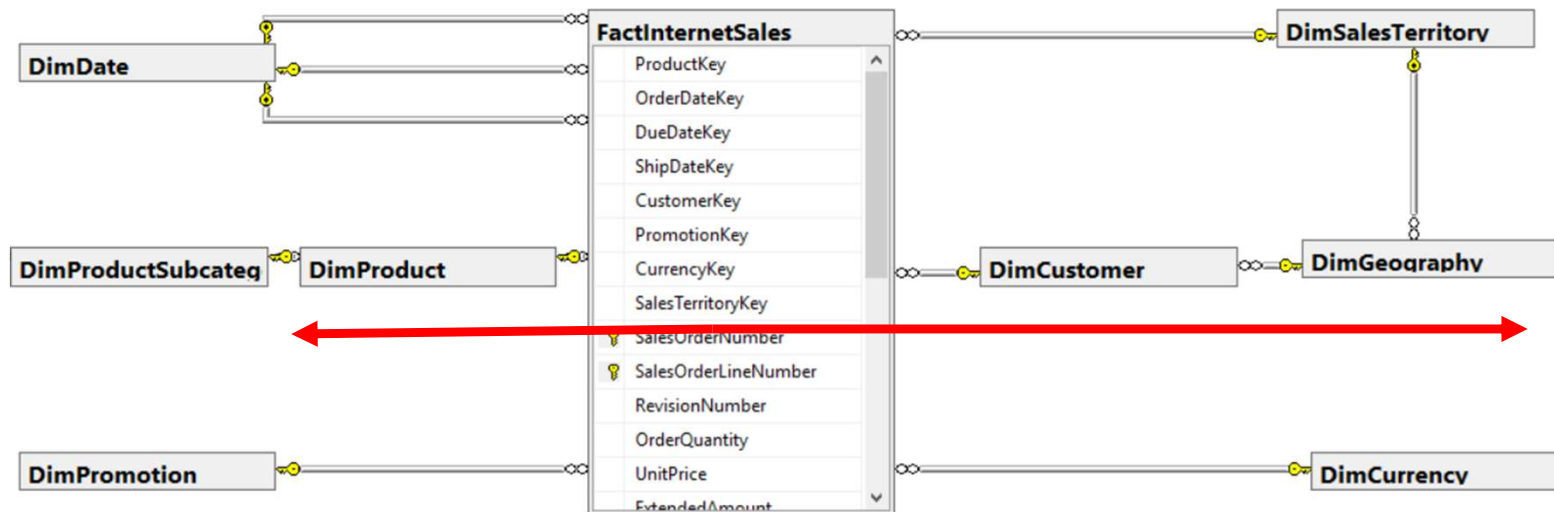
Introduction aux concepts

Objectifs de la modélisation dimensionnelle



Introduction aux concepts

Objectifs de la modélisation dimensionnelle



Modélisation dimensionnelle

2

Modélisation dimensionnelle

Principes de base : La dénormalisation

- ❑ La dénormalisation consiste à regrouper plusieurs tables entre elles, quitte à créer de la redondance
 - ❑ Requête simplifiée => Plus besoin d'aller "chercher" les infos du clients sur 15 tables
 - ❑ Lecture simplifiée => dans un modèle relationnel, une table pays peut servir à plusieurs autres entités (ex : le client et le fournisseur). Néanmoins, cela complique la compréhension.

Modélisation dimensionnelle

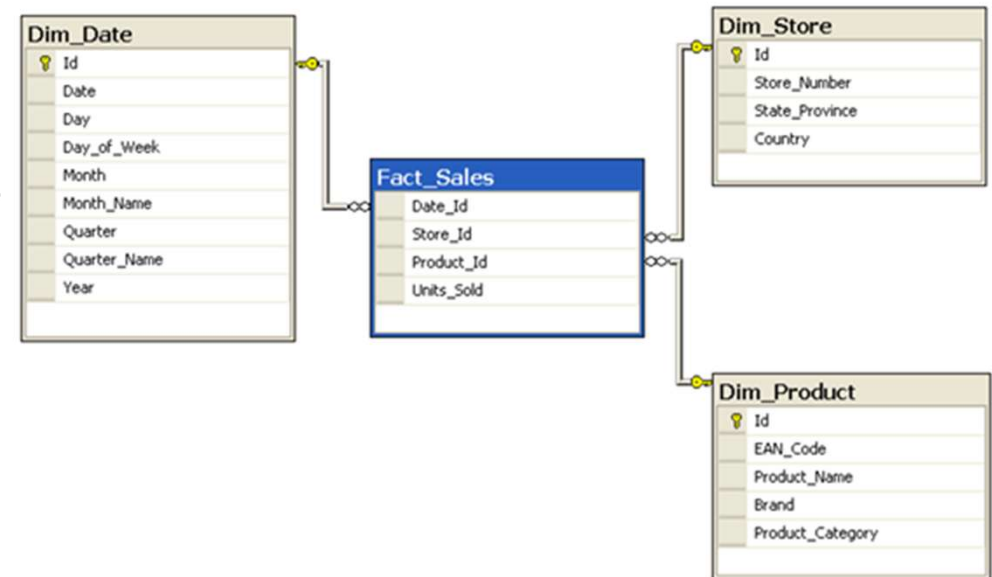
Principes de base : La dénormalisation

- ❑ Donc il faut créer une grosse table unique ?

Modélisation dimensionnelle

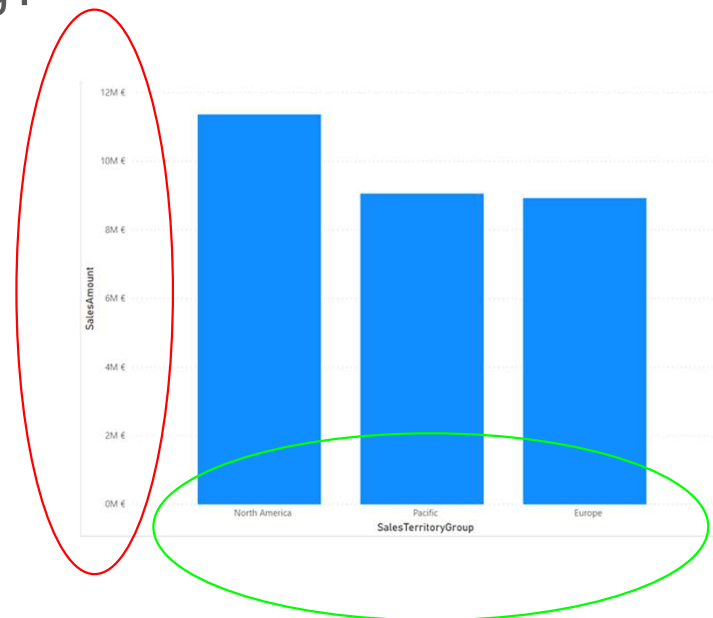
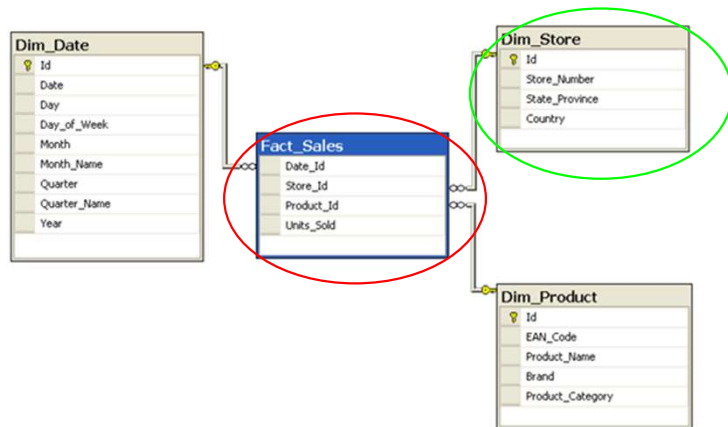
Star schema :

- ❑ Modèle dimensionnel implémenté dans un système de gestion de base de données relationnelles
- ❑ Appelé “Star schema” de par sa structure, en forme d’étoile



Modélisation dimensionnelle

Pourquoi ce modèle est-il adapté au reporting?



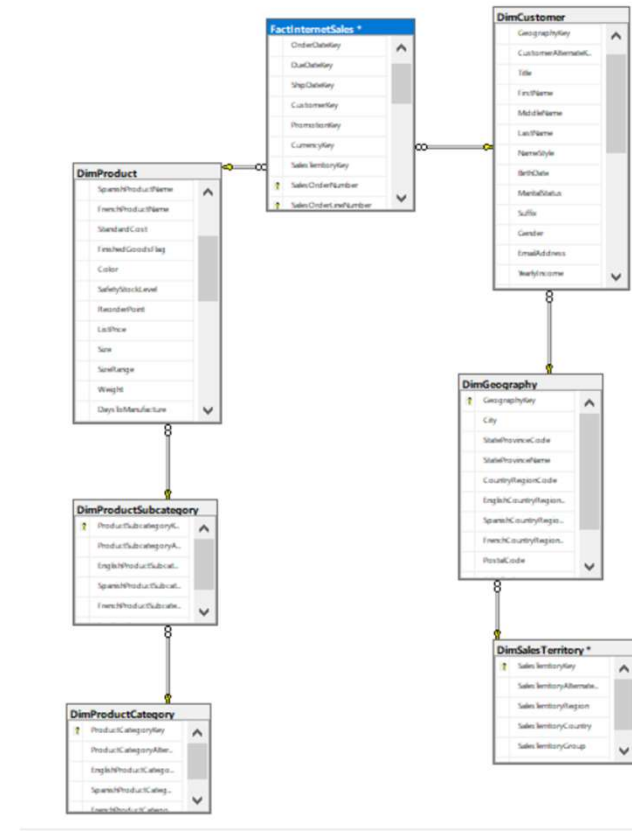
Modélisation dimensionnelle

Modèle en flocon

Il s'agit d'une variation du modèle en étoile dans lequel les hiérarchies sont renormalisées.

Il est conseillé d'utiliser un modèle en étoile et de ne "floconner" que si nécessaire.

(cf : faits sur différentes granularités)



Modélisation dimensionnelle

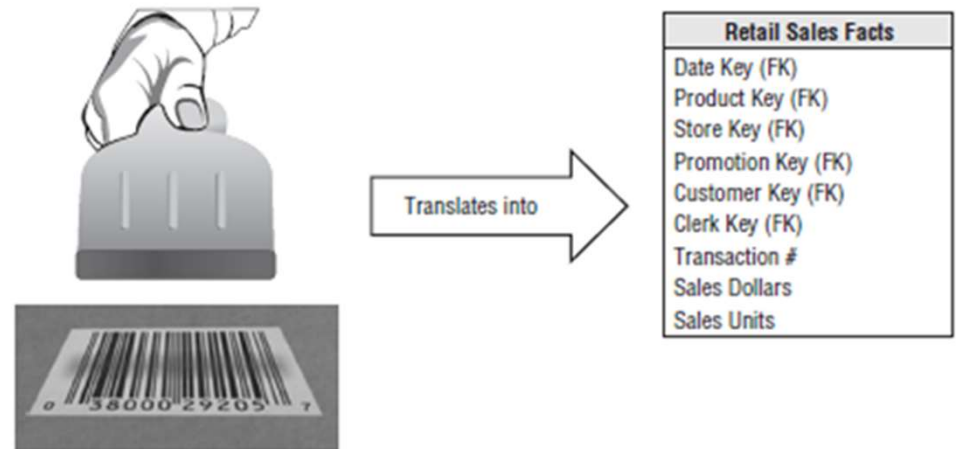
Fait :

- ☐ Fait = une mesure métier. Exemple : une transaction
- ☐ Chaque entrée dans une table de faits correspond à un événement réel
- ☐ Le niveau de détail de ce fait est appelé le grain (granularité)
- ☐ Toutes les entrées doivent posséder la même granularité
- ☐ Les faits les plus courants sont, en principe, numériques et additifs
- ☐ Ce sont principalement des valeurs de type continu

Modélisation dimensionnelle

Table de faits :

- ❑ Table enregistrant les mesures résultant des processus métiers d'une organisation



Modélisation dimensionnelle

- ❑ Une table de faits est généralement grande en terme d'entrées mais plutôt étroite (faible en terme de colonnes)
- ❑ Les tables de faits peuvent être de 3 types :
 - ❑ Transactionnelle (cf : case study 1)
 - ❑ Périodique (cf : case study 2)
 - ❑ Accumulative(cf : case study 2)
- ❑ Elles possèdent toujours des clés étrangères afin d'être associées aux clés primaires des tables de dimensions
- ❑ Elles possèdent une clé primaire qui est généralement une clé composite de différentes clés étrangères

Modélisation dimensionnelle

Table de dimensions :

- ☐ Table enregistrant la description du contexte associé à un fait
- ☐ Elles répondent aux questions :
 - ☐ Qui ?
 - ☐ Quoi ?
 - ☐ Où ?
 - ☐ Quand ?
 - ☐ Comment ?
 - ☐ Pourquoi ?

Product Dimension
Product Key (PK)
SKU Number (Natural Key)
Product Description
Brand Name
Category Name
Department Name
Package Type
Package Size
Abrasive Indicator
Weight
Weight Unit of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
...

Modélisation dimensionnelle

- ❑ Une table de dimensions est généralement faible en terme d'entrées mais plutôt large en terme de colonnes
- ❑ Elles possèdent toujours une clé primaire afin de respecter les contraintes d'intégrité référentielle d'une table de faits à laquelle elle sont liées
- ❑ Les attributs des tables de dimensions sont la première source de conditions de tri ou de regroupement lors des requêtes
- ❑ Ses attributs sont généralement de type textuel. Par exemple : Un utilisateur peut être intéressé de connaître *Les ventes par produit*, où la vente est la **mesure** et où le produit est la **dimension**

Modélisation dimensionnelle

Bonnes pratiques

- ☐ Les attributs des dimensions devraient toujours être libellés de la manière la plus claire et explicite qui soit
- ☐ Ne pas hésiter à créer des attributs supplémentaires pour rendre textuel un attribut de type code par exemple
- ☐ Si un attribut de type code possède de la signification embarquée, ne pas hésiter à le décomposer en plusieurs attributs

Cela facilitera grandement l'utilisation de votre Data Warehouse et par conséquent, améliorera sa qualité

Modélisation dimensionnelle

Mesure numérique vs dimension numérique

- ❑ Une valeur numérique n'est pas forcément une mesure
- ❑ Par exemple : le coût de fabrication d'un produit
 - ❑ S'il varie peu ou pas, il s'agit d'une dimension
 - ❑ S'il a tendance à varier souvent, il est alors envisageable de le considérer comme une mesure de sorte à pouvoir l'analyser

Modélisation dimensionnelle

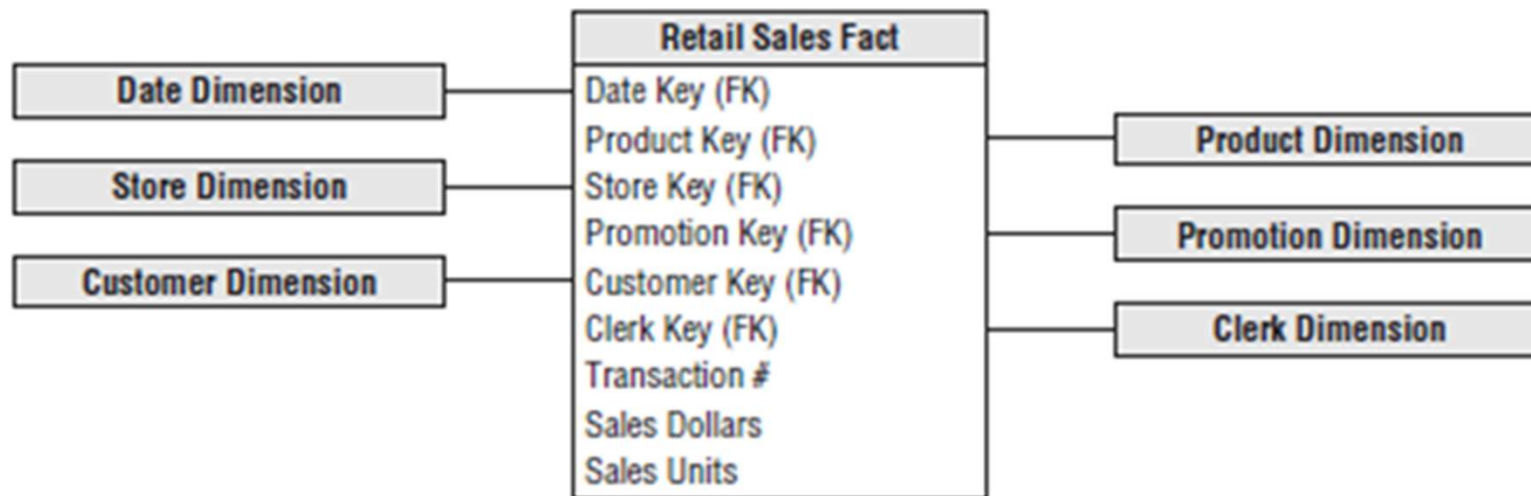
Hiérarchies

- ❑ Certaines tables de dimensions présentent des attributs qui peuvent être subordonnés dans une hiérarchie
- ❑ Exemple : Catégorie => Marque => Description du produit
- ❑ Dans un modèle de type star schema, on le représente comme ci-contre :

Product Key	Product Description	Brand Name	Category Name
1	PowerAll 20 oz	PowerClean	All Purpose Cleaner
2	PowerAll 32 oz	PowerClean	All Purpose Cleaner
3	PowerAll 48 oz	PowerClean	All Purpose Cleaner
4	PowerAll 64 oz	PowerClean	All Purpose Cleaner
5	ZipAll 20 oz	Zippy	All Purpose Cleaner
6	ZipAll 32 oz	Zippy	All Purpose Cleaner
7	ZipAll 48 oz	Zippy	All Purpose Cleaner
8	Shiny 20 oz	Clean Fast	Glass Cleaner
9	Shiny 32 oz	Clean Fast	Glass Cleaner
10	ZipGlass 20 oz	Zippy	Glass Cleaner
11	ZipGlass 32 oz	Zippy	Glass Cleaner

Modélisation dimensionnelle

- ❑ Chaque processus métier représenté dans un modèle dimensionnel en étoile est constitué d'une table de faits autour de laquelle “gravitent” des tables de dimensions



Architecture d'une solution BI/DW



Architecture d'une solution BI/DW

Back room :

- ☐ Désigne tous les processus se déroulant en amont de la présentation des données dans le modèle dimensionnel
- ☐ Processus ETL => Extract, Transform, Load
- ☐ Extract : Capture des données des différentes sources
- ☐ Transform : Transformation et uniformisation des données
- ☐ Load : Intégration des données dans le système dimensionnel

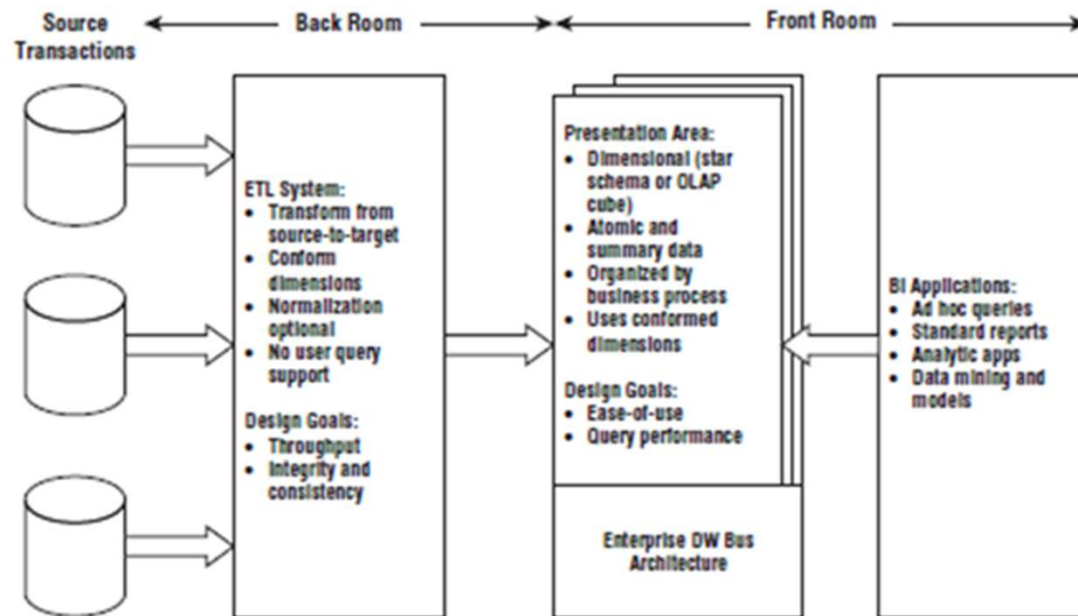
Architecture d'une solution BI/DW

Front Room :

- ❑ Presentation area :
 - ❑ Environnement sur lequel les requêtes sont effectuées, là où sont présentées les données aux utilisateurs
- ❑ Bi Application :
 - ❑ Au sens large : Toute application de business intelligence développée pour les utilisateurs à des fins de reporting

Architecture d'une solution BI/DW

Architecture "Kimball" classique



Data Warehousing et modélisation dimensionnelle

Architecture d'une solution BI/DW

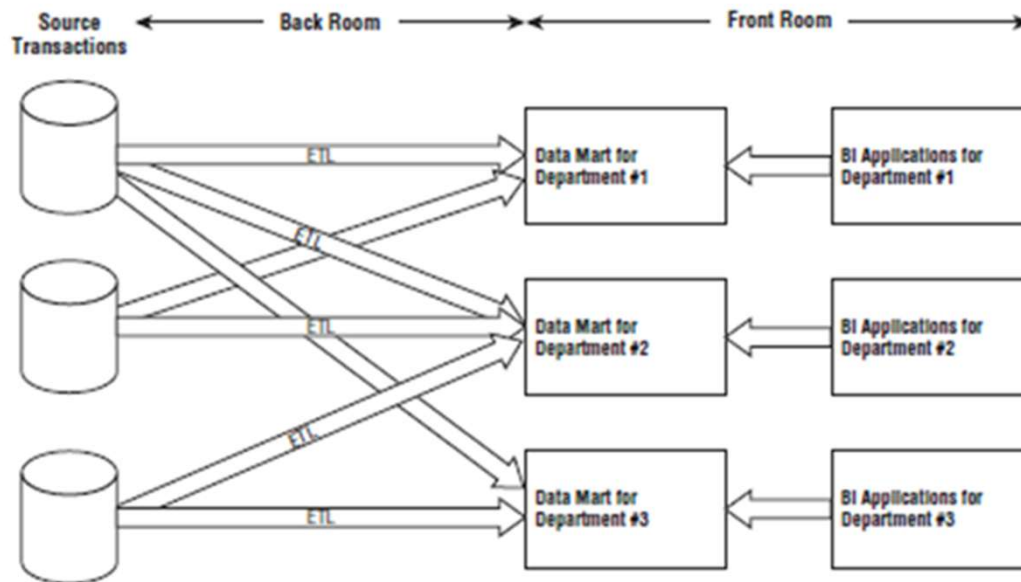
Architecture "Kimball" classique :

- ❑ Extraction du data brut à partir des différentes sources de données de l'entreprise
- ❑ Centralisation, uniformisation, transformation et intégration du data dans un modèle dimensionnel
- ❑ Connexion des applications de business intelligence à ce modèle dimensionnel afin de générer du reporting

Remarque : S'il est acceptable de concevoir une structure normalisée pour supporter un processus ETL, ce type d'approche ne peut être recommandé dans une approche "Kimball" dans la mesure où il va à l'encontre de nos objectifs en matière d'intelligibilité et de recherche de performance

Architecture d'une solution BI/DW

Architecture alternative : "Data mart"



Data Warehousing et modélisation dimensionnelle

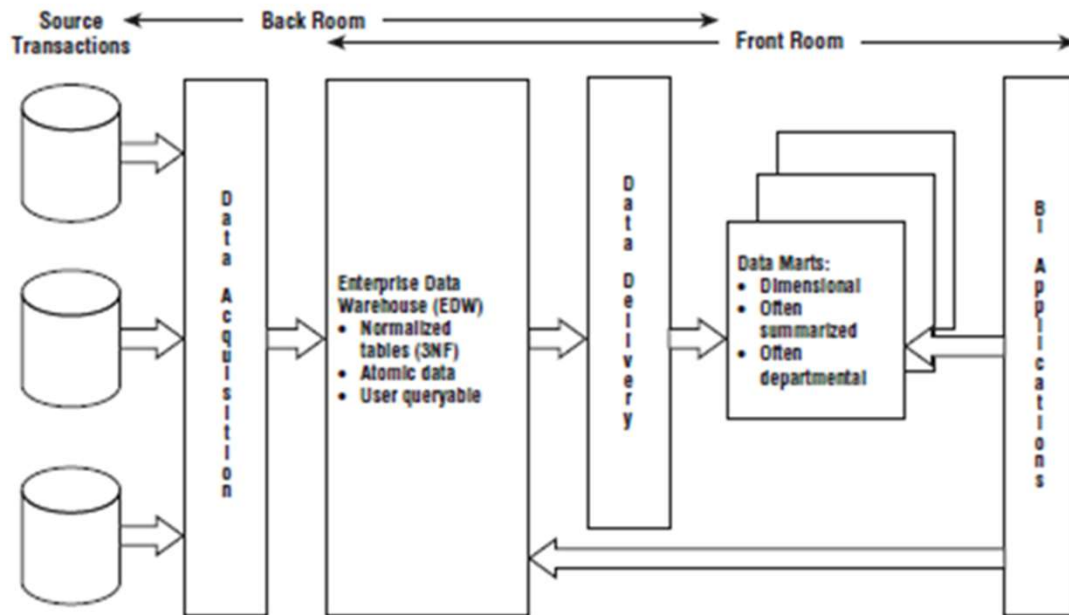
Architecture d'une solution BI/DW

Architecture alternative : "Data mart"

- ❑ Approche similaire dans la construction mais la centralisation des données a lieu à un niveau départemental dans l'entreprise
- ❑ Surtout utilisé dans les très grandes entreprises
- ❑ + : Développement plus rapide et moins onéreux (relatif)
- ❑ - : Manque de concordance des extraits de données entre les différents départements, manque de vision globale

Architecture d'une solution BI/DW

Architecture alternative : "Hub-and-spoke"



Data Warehousing et modélisation dimensionnelle

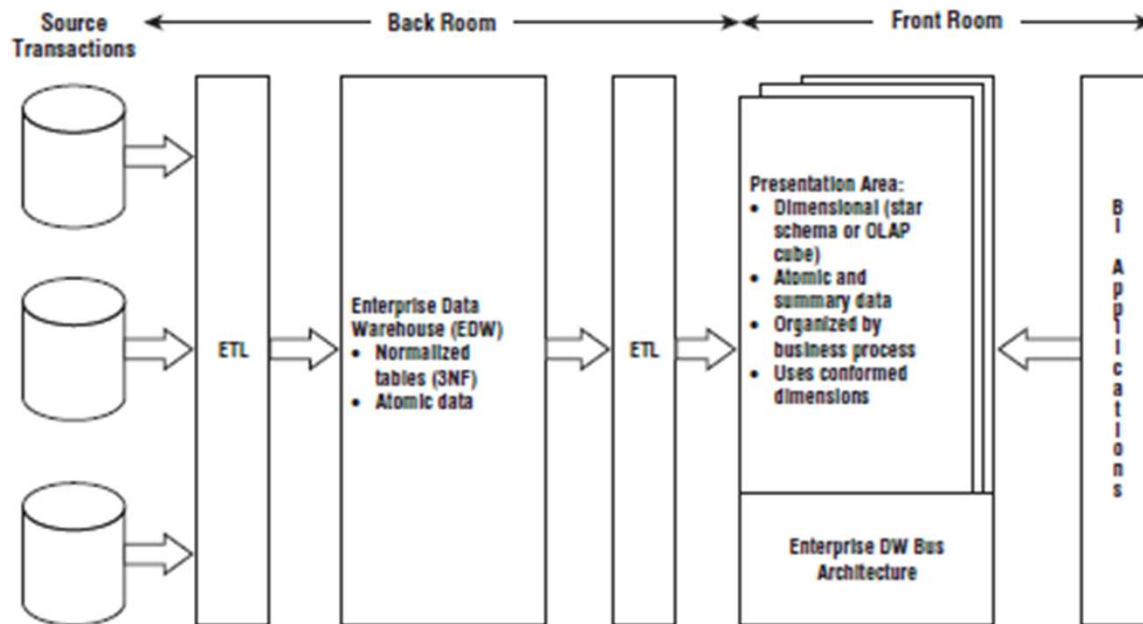
Architecture d'une solution BI/DW

Architecture alternative : "Hub-and-spoke"

- ❑ Centralisation des données dans une structure relationnelle, **Enterprise Data Warehouse** (Obligatoire, normalisée, granularité la plus fine)
- ❑ "Livraison" du data vers des data marts (généralement agrégé)
- ❑ Applications BI qui peuvent se connecter à la fois aux data marts et à l'EDW
- ❑ + : Cohérence des données, data marts spécifiques aux différents départements
- ❑ - : Granularité la plus fine difficilement accessible

Architecture d'une solution BI/DW

Architecture alternative : “Kimball/Hub-and-spoke”



Data Warehousing et modélisation dimensionnelle

Architecture d'une solution BI/DW

Architecture alternative : "Kimball/Hub-and-spoke"

- ❑ ETL 1 : Centralisation des données dans une structure relationnelle, **Enterprise Data Warehouse** (normalisé, granularité atomique)
- ❑ ETL 2 : Intégration des données centralisées dans un modèle dimensionnel (dénormalisé, granularité atomique et agrégation des données)
- ❑ Applications BI qui se connectent au modèle dimensionnel
- ❑ + : Cohérence des données, granularité fine et accessible
- ❑ - : Développement plus long et plus onéreux

Case study : Ventes au détail

4

Case study : Ventes au détail

- ☐ Etapes de modélisation
- ☐ Faits dérivés
- ☐ Faits non-additifs
- ☐ Table de faits transactionnelle
- ☐ Dimensions
 - ☐ Dimension date
 - ☐ Dimension produit
 - ☐ Dimension magasin
 - ☐ Dimension promotion
- ☐ Valeurs nulles
- ☐ Tables de faits sans fait
- ☐ Dimension dégénérée
- ☐ Clés artificielles et naturelles
- ☐ Snowflake Schema
- ☐ Exercice : Modélisation d'une table de faits et des dimensions associées

Case study : Ventes au détail

Un modèle dimensionnel est un modèle fait pour le business. Il doit donc se baser sur les questions que le business cherche à résoudre. Dans le cas du retail, les questions peuvent être, du type :

- ☐ Comment se répartissent mes ventes sur mes différents pays, régions, villes ?
- ☐ Quels articles génèrent le plus de profit ?
- ☐ Comment se répartissent les ventes en fonction du genre de mes clients ? De leur âge ?
- ☐ Comment évoluent les ventes ? par année ? par mois ?
- ☐ Quelle est ma progression par rapport au même mois de l'année précédente ?
- ☐ Etc, etc,...

Case study : Ventes au détail

Etapes de modélisation :

☐ Identifier un processus métier :

- ☐ Les processus métiers sont généralement exprimables par des “verbes” car ils représentent des actions qui sont effectuées par l’entreprise. Ex : La vente
- ☐ Ils sont souvent liés à un système opérationnel. Ex : Système de facturation
- ☐ Ils génèrent des métriques. Ex : Le montant d’une vente

☐ Déclarer le grain :

- ☐ Signifie spécifier à quoi correspond exactement une entrée dans la table de fait
- ☐ Cette déclaration s’exprime avec des termes métiers
- ☐ Ex : Une entrée pour un article scanné lors d’une transaction
- ☐ Ex : Une entrée pour carte d’embarquement scannée dans un aéroport

Case study : Ventes au détail

Etapes de modélisation :

☐ Identifier les dimensions

- ☐ Si la granularité est bien définie, il est plus facile d'identifier les dimensions
- ☐ Par quels “regroupements d'attributs”, les utilisateurs métiers peuvent-ils décrire une entrée ?
- ☐ Ex : Une ligne vente, notre grain, peut être décrit par une dimension temporelle, une dimension géographique, elle concerne un produit en particulier, etc

☐ Identifier les faits :

- ☐ Quelles métriques vont être mesurées par notre fait?
- ☐ Ces métriques doivent posséder le même niveau de granularité que le grain déclaré
- ☐ Ex : Le montant, le profit, la quantité, etc

Case study : Ventes au détail

Identifier le processus métier

- Exprimable avec un verbe
- Souvent liés à un système opérationnel
- Génère des métriques

Déclarer le grain

- Spécifie exactement à quoi correspond une entrée de la table de fait
- Exprimé en terme métier

Identifier les dimensions

- Suivant quels axes le fait peut-il être identifié ?
- Répond aux questions:
 - Qui
 - Quoi
 - Où
 - Comment
 - Pour qui
 - ...

Identifier les faits

- Qu'est ce qui peut-être (numériquement) mesuré concernant le fait ?
- Répond aux questions "combien"

Case study : Ventes au détail

Faits dérivés :

- ❑ Il est souvent possible de dériver des métriques au départ d'autres métriques
- ❑ Exemple : En soustrayant le coût total d'un produit de son prix de vente, vous pouvez obtenir le bénéfice brut pour celui-ci
- ❑ Il est recommandé de rajouter ces mesures dérivées dans votre table de faits
- ❑ En effet, le faible coût en stockage de ces mesures dérivées compense largement les erreurs de calcul potentielles provenant des utilisateurs

Case study : Ventes au détail

Faits non-additifs

- ❑ Les tables de faits contiennent parfois des métriques additives par nature, comme le prix unitaire, mais dont l'agrégation par une somme n'a pas de sens.
- ❑ Un autre exemple est la marge brut (en pourcentage) obtenue en divisant le bénéfice brut par le prix de vente.
- ❑ Additionner ces pourcentages a également peu de sens.
- ❑ La légitimité de leur présence ou non dans une table de faits peut être sujet à débat
- ❑ Considérés comme non-additifs, il convient simplement d'être prudent et de faire preuve de bons sens lors de leur utilisation dans un outil BI.

Case study : Ventes au détail

Table de faits transactionnelle :

❑ Caractéristiques :

- ❑ Type de table de faits les plus courantes
- ❑ La granularité de ce type de table est souvent exprimable très succinctement.

Exemple : Une entrée pour une ligne de transaction

- ❑ Comme elles relatent chaque évènement transactionnel, elles peuvent être gigantesques en termes de nombre d'entrées
- ❑ Elles tendent à posséder beaucoup de dimensions
- ❑ Leurs métriques sont typiquement additives

Case study : Ventes au détail

Table de dimensions :

- ❑ Dimension date :
 - ❑ Elle est presque systématiquement présente dans tous les modèles dimensionnels
 - ❑ Contrairement aux autres dimensions, elle peut être peuplée “à l’avance”
 - ❑ Exemple : Il est possible de couvrir une période de 20 ou 30 ans avec granularité journalière avec une “petite” table d’environ 10000 entrées

Case study : Ventes au détail

Table de dimensions :

❑ Dimension date :

- ❑ Même s'il est possible de récupérer toutes ces informations d'un champ de type "date" dans un SGBD grâce à des fonctions SQL, il est recommandé d'être relativement exhaustif en terme d'attributs pour cette dimension
- ❑ En effet, les utilisateurs ne sont pas forcément familiers avec le langage SQL
- ❑ De plus, certains attributs temporels peuvent être propre à une organisation - comme les trimestres fiscaux - et ont besoin d'être "calculés" à l'avance
- ❑ La clé utilisée est une *smart date key* représenté par un entier de type YYYYMMDD

Case study : Ventes au détail

Table de dimensions :

❑ Dimension date :

- ❑ Il est envisageable d'inclure des indicateurs de type booléen dans une dimension date
- ❑ Exemple : Inclure un booléen signalant si une date est un jour de semaine ou non, permettrait à l'utilisateur de déterminer différencier les ventes qui ont lieu en semaine ou en week-end
- ❑ Il est préférable d'utiliser des indicateurs avec des valeurs explicites plutôt que des valeurs binaires standards
- ❑ Exemple : Un booléen "Jour de semaine/Jour de week-end" sera plus pratique à l'usage qu'un "True/False" ou un "0/1"

Case study : Ventes au détail

Table de dimensions :

- ❑ Dimension time-of-day :
 - ❑ Un champ de type timestamp est la combinaison d'une date et d'une heure
 - ❑ S'il est possible définir une granularité à la seconde pour une dimension date, cela est toutefois peu recommandé
 - ❑ En effet, pour une période de 20 ans : 7.300 entrées vs 630.000.000 entrées
 - ❑ Il est donc préférable pour des raisons de performance de créer une dimension time-of-day avec la granularité désirée

Case study : Ventes au détail

❑ Dimension date

Date Key	Date	Full Date Description	Day of Week	Calendar Month	Calendar Quarter	Calendar Year	Fiscal Year-Month	Holiday Indicator	Weekday Indicator
20130101	01/01/2013	January 1, 2013	Tuesday	January	Q1	2013	F2013-01	Holiday	Weekday
20130102	01/02/2013	January 2, 2013	Wednesday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130103	01/03/2013	January 3, 2013	Thursday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130104	01/04/2013	January 4, 2013	Friday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130105	01/05/2013	January 5, 2013	Saturday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130106	01/06/2013	January 6, 2013	Sunday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130107	01/07/2013	January 7, 2013	Monday	January	Q1	2013	F2013-01	Non-Holiday	Weekday
20130108	01/08/2013	January 8, 2013	Tuesday	January	Q1	2013	F2013-01	Non-Holiday	Weekday

Date Dimension
Date Key (PK)
Date
Full Date Description
Day of Week
Day Number in Calendar Month
Day Number in Calendar Year
Day Number in Fiscal Month
Day Number in Fiscal Year
Last Day in Month Indicator
Calendar Week Ending Date
Calendar Week Number in Year
Calendar Month Name
Calendar Month Number in Year
Calendar Year-Month (YY(KY-MM)
Calendar Quarter
Calendar Year-Quarter
Calendar Year
Fiscal Week
Fiscal Week Number in Year
Fiscal Month
Fiscal Month Number in Year
Fiscal Year-Month
Fiscal Quarter
Fiscal Year-Quarter
Fiscal Half Year
Fiscal Year
Holiday Indicator
Weekday Indicator
SQL Date Stamp
...

Case study : Ventes au détail

Table de dimensions :

❑ Dimension Produit :

- ❑ Contient des attributs typiques tels que le nom, la marque, le prix unitaire, le fabricant etc
- ❑ Ce type de table contient souvent une **hiérarchie** de type :
Catégorie → Sous-catégorie → Nom du produit
- ❑ Ces hiérarchies sont des relations du type One-To-Many
- ❑ Dans un modèle dimensionnel avec un schéma en étoile, cela implique de la redondance. Toutefois cela est totalement acceptable dans un contexte non-relationnel
- ❑ En effet, le besoin de qualité de performance en requête supplante largement le besoin de préserver de l'espace disque

Product Dimension
Product Key (PK)
SKU Number (NK)
Product Description
Brand Description
Subcategory Description
Category Description
Department Number
Department Description
Package Type Description
Package Size
Fat Content
Diet Type
Weight
Weight Unit of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
...

Case study : Ventes au détail

Table de dimensions :

❑ Dimension Produit :

- ❑ Les clés business de ce type de dimension possèdent souvent du “sens embarqué”, ne pas hésiter à créer des colonnes supplémentaires pour “capturer ce sens”
- ❑ Rappel : si un prix unitaire du produit varie peu, il est totalement acceptable que cette métrique se retrouve dans la dimension produit et pas dans une table de faits
- ❑ Note : Rien n’interdit aux métriques qui peuvent être utilisés à des fins de calcul (faits) ou à des fins de regroupement (dimensions) de se retrouver dans les 2 types de table

Case study : Ventres au détail

Table de dimensions :

❑ Dimension Produit :

- ❑ Drill-down signifie ajouter un axe de répartition supplémentaire à un extrait de données
- ❑ Roll-up signifie retirer un axe de répartition à un extrait
- ❑ L'exemple ci-contre nous montre un premier drill par marque et un second par teneur en graisse

Department Name	Sales Dollar Amount
Bakery	12,331
Frozen Foods	31,776

Drill down by brand name:

Department Name	Brand Name	Sales Dollar Amount
Bakery	Baked Well	3,009
Bakery	Fluffy	3,024
Bakery	Light	6,298
Frozen Foods	Coldpack	5,321
Frozen Foods	Freshlike	10,476
Frozen Foods	Frigid	7,328
Frozen Foods	Icy	2,184
Frozen Foods	QuickFreeze	6,467

Or drill down by fat content:

Department Name	Fat Content	Sales Dollar Amount
Bakery	Nonfat	6,298
Bakery	Reduced fat	5,027
Bakery	Regular fat	1,006
Frozen Foods	Nonfat	5,321
Frozen Foods	Reduced fat	10,476
Frozen Foods	Regular fat	15,979

Case study : Ventas au détail

Table de dimensions :

❑ Dimension Magasin :

- ❑ Ce type de dimension contient souvent une hiérarchie du type
Pays → Etat/Province → Ville
- ❑ Note : attention au fait que certains pays peuvent avoir des Etats/Provinces qui ont des villes aux noms identiques !
- ❑ Il est alors préférable pour constituer une hiérarchie correcte de générer un champ supplémentaire “Etat-Ville”
Pays → Etat/Province → “Etat-Ville”

Store Dimension
Store Key (PK)
Store Number (NK)
Store Name
Store Street Address
Store City
Store County
Store City-State
Store State
Store Zip Code
Store Manager
Store District
Store Region
Floor Plan Type
Photo Processing Type
Financial Service Type
Selling Square Footage
Total Square Footage
First Open Date
Last Remodel Date
...

Case study : Ventres au détail

Table de dimensions :

- ❑ Dimension Promotion :
 - ❑ Dimension qui indique sous quelles conditions les ventes peuvent avoir lieu
 - ❑ Attributs typiques :
 - ❑ nom de promotion,
 - ❑ type de publicité,
 - ❑ type de coupon,
 - ❑ type de réduction,
 - ❑ **date de début de promotion,**
 - ❑ **date de fin de promotion**
 - ❑ C'est une dimension de type causal

Promotion Dimension
Promotion Key (PK)
Promotion Code
Promotion Name
Price Reduction Type
Promotion Media Type
Ad Type
Display Type
Coupon Type
Ad Media Name
Display Provider
Promotion Cost
Promotion Begin Date
Promotion End Date
...

Case study : Ventes au détail

Table de dimensions :

☐ Dimension Promotion :

- ☐ Elle peut indiquer l'impact d'une promotion sur les ventes d'un produit (lift) en comparant les ventes à celles des périodes précédentes ou encore grâce à des modèles statistiques plus élaborés
- ☐ Elle permet de vérifier si une promotion a impacté négativement les ventes habituelles (time shifting)
- ☐ Elle permet de vérifier une promotion a impacté les ventes des autres produits de même catégorie (cannibalization)
- ☐ Elle permet de mesurer l'impact d'une promotion sur les ventes habituelles (market growth)
- ☐ Contrôler simplement si une promotion est profitable financièrement
- ☐ Comparer l'efficacité d'un type de promotion sur un autre

Case study : Ventes au détail

Table de dimensions :

❑ Dimension Promotion :

- ❑ Une transaction n'a pas forcément lieu sous une promotion
- ❑ Il faut alors créer une entrée unique dans la dimension promotion identifiant la condition de "non-promotion" afin de ne pas créer de null dans votre table de faits
- ❑ La valeur null se retrouve parfois **dans les attributs des tables de dimensions**, notamment lorsque les tables de dimensions ne sont pas entièrement peuplées ou quand les attributs de faits ne sont pas applicables à toutes les entrées des dimensions
- ❑ Il est toutefois recommandé pour éviter les confusions, notamment lors des jointures, de **remplacer cette valeur null par un champ texte du type "non-applicable" ou "inconnu"**
- ❑ **Dans les mesures tables de faits**, il est préférable de **conserver cette valeur null** afin de ne pas biaiser les conditions d'agrégation

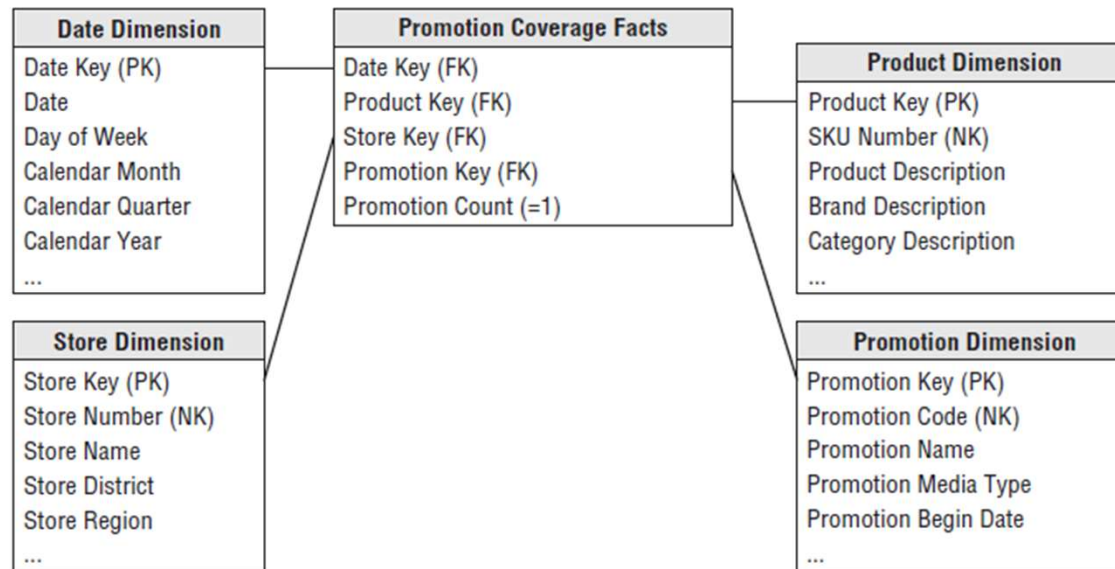
Case study : Ventes au détail

Table de faits sans faits:

- ❑ Malgré la dimension promotion que nous venons d'établir, il reste toutefois une question à laquelle nous ne pouvons pas répondre
 - ❑ Quels produits étaient en promotion mais ne se sont pas vendus ?
- ❑ En effet, pour qu'une entrée soit ajoutée à la table de faits et donc mesurée d'un point de vue promotionnel, il est nécessaire qu'il y ait une vente
- ❑ La solution est de créer une table de faits supplémentaire qui identifie "l'action d'assignation" d'un produit à une promotion
- ❑ La granularité est une entrée par produit en promotion par jour

Case study : Ventres au détail

Table de faits sans faits:



Case study : Ventes au détail

Table de faits sans faits:

- ❑ Ce type de table est appelé “table sans fait” car elle ne comporte pas de mesures
- ❑ Elle capture simplement les relations qui existent entre les produits et les promotions
- ❑ Pour répondre à notre question, il faudra donc :
 - ❑ Identifier les produits qui étaient en promotion à un certain moment
 - ❑ Vérifier si des ventes ont été enregistrées dans la table de faits pour ces produits

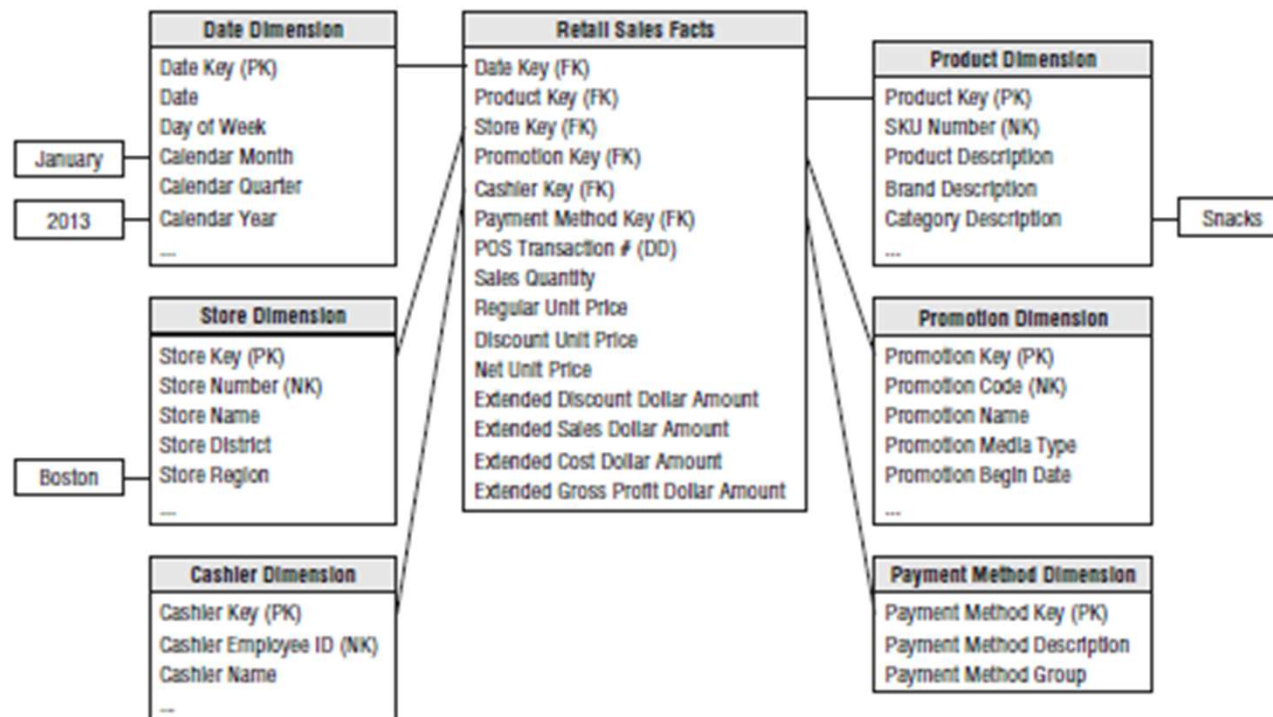
Case study : Ventes au détail

Table de dimensions :

☐ Dimension dégénérée :

- ☐ Cas de la dimension transaction :
- ☐ Une vente est associée à une transaction par un numéro de transaction
- ☐ La transaction pourrait être une dimension qui regroupe d'autres informations intéressantes telles que l'identifiant du magasin, l'identifiant de l'employé, etc
- ☐ Toutefois ces attributs ont certainement déjà été associés à d'autres dimensions comme une dimension magasin ou employé
- ☐ La seule information intéressante restante est finalement le numéro de transaction (qui va permettre de grouper les lignes de commandes par transaction unique)
- ☐ Pour ne pas créer de dimension supplémentaire inutilement, cette dimension est intégrée directement en tant qu'attribut de la table de faits

Case study : Ventres au détail



Case study : Ventes au détail

Clés artificielles:

- ❑ L'identifiant unique d'une entrée dans une table de dimensions devrait toujours être une clé artificielle et non la clé naturelle du système duquel il provient, car :
- ❑ La clé artificielle est plus simple d'utilisation qu'une clé composite
- ❑ Même si une clé naturelle peut sembler stable, rien n'indique qu'une clé naturelle inactive ne puisse être recyclée à un moment ou l'autre
- ❑ Elle augmente les performances, en effet, elle est toujours le plus petit entier possible servant à référencer une entrée, ce qui n'est pas forcément le cas d'une clé naturelle => Facilité de tri, de regroupement

Case study : Ventes au détail

Clés artificielles:

- ❑ Permet de gérer la valeur null dans une dimension (cf : null dans la table promotion)
- ❑ Et plus important encore, elle permet de tracer les changements dans une table de dimensions. (Historisation : cf case study 3)
- ❑ Pour la dimension date, la clé artificielle est en général, un entier de type *YYYYMMDD*, plus malin à utiliser qu'un entier *DDMMYYYY* pour délimiter ou trier un intervalle de temps avec une clé primaire

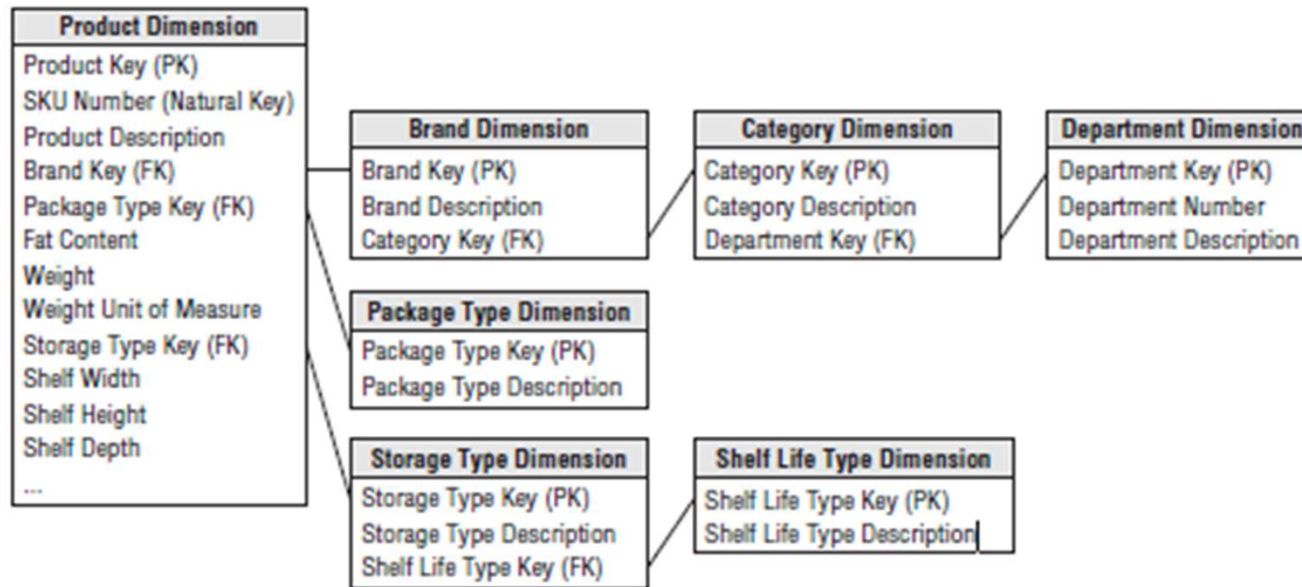
Case study : Ventes au détail

Schéma en étoile vs schéma en flocon

- ❑ Pour de nombreux designers de modèles relationnels la redondance que l'on peut retrouver au niveau de certaines tables de dimensions peut sembler une hérésie
- ❑ Si on observe l'exemple de la table produit présentée précédemment, il est totalement possible de normaliser partiellement cette dimension
- ❑ La normalisation d'une table dénormalisée s'appelle le *snowflaking*

Case study : Ventres au détail

Schéma en étoile vs schéma en flocon



Case study : Ventes au détail

Schéma en étoile vs schéma en flocon

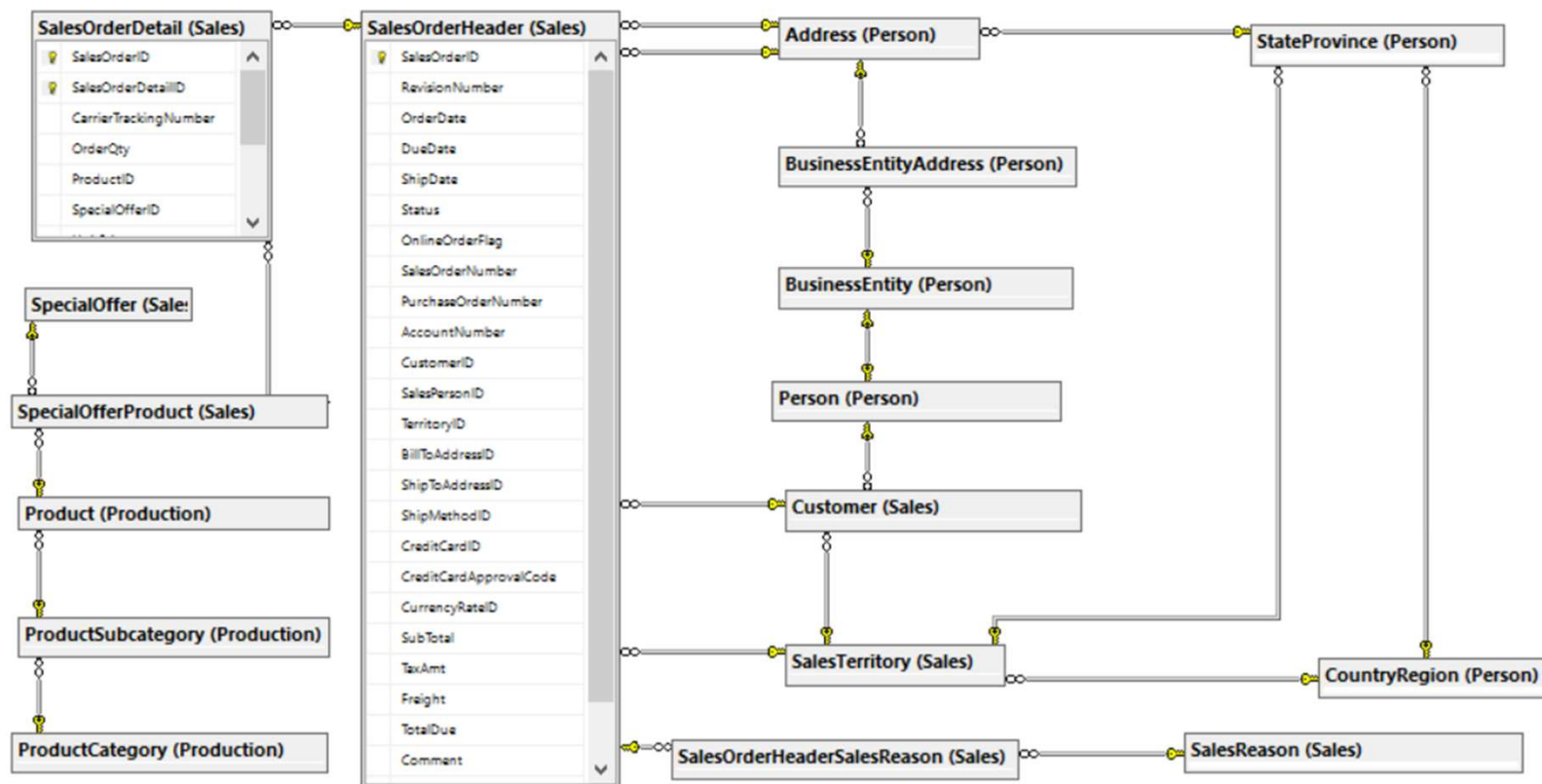
- ❑ Si ce type de schéma est absolument justifié et acceptable, il est recommandé de résister à l'envie de vouloir normaliser "à-tout-va".
- ❑ Cette démarche ne s'inscrit pas réellement dans la **volonté de simplification et de gain de performance** recherchée par un modèle dimensionnel
- ❑ La multiplicité des tables ne facilite pas l'usage pour l'utilisateur
- ❑ L'augmentation du nombre de jointures ne fait que diminuer la performance
- ❑ Le gain d'espace est faible comparativement à ces objectifs (moins d'1%)

Case study : Ventes au détail

Exercice :

- ❑ Construire un modèle dimensionnel des ventes internet de la base de données Adventure Works en partant de la structure relationnelle suivante :

Case study : Ventas au détail



Data Warehousing et modélisation dimensionnelle

Case study Inventaire



Case study : Inventaire

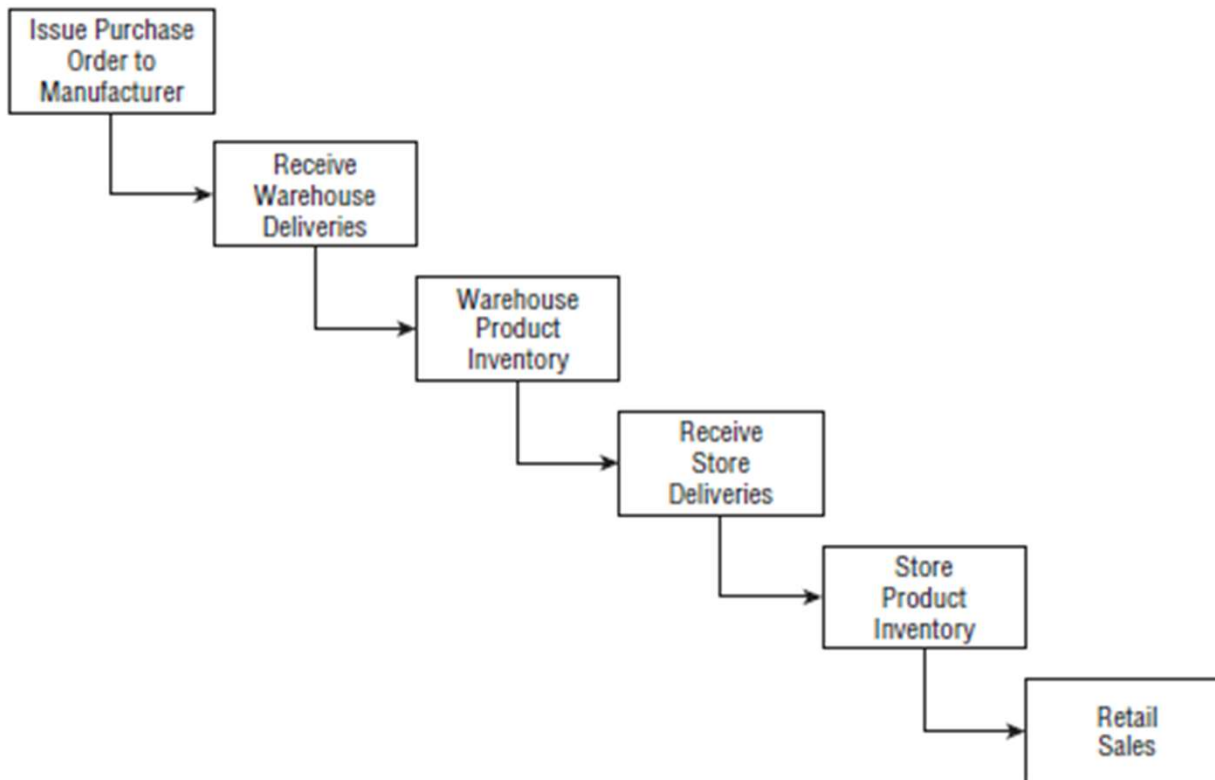
- ☐ Chaîne de valorisation
- ☐ Importance de l'inventorisation
- ☐ Snapshot périodique
- ☐ Inventaire transactionnel
- ☐ Snapshot cumulé
- ☐ Exercice

Case study : Inventaire

Chaîne de valorisation

- ❑ Les processus métier des organisations sont contenus dans un flux global d'activités
- ❑ Ce flux est appelé "chaîne de valorisation"
- ❑ Exemple :
 - ❑ Une entreprise de retail achète à différents fabricants
 - ❑ Les produits sont acheminés et stockés dans un entrepôt, où ils sont inventoriés
 - ❑ Ils sont ensuite livrés dans chaque point de vente où ils seront à nouveau stockés et inventoriés
 - ❑ Le produit est vendu au consommateur final

Case study : Inventaire

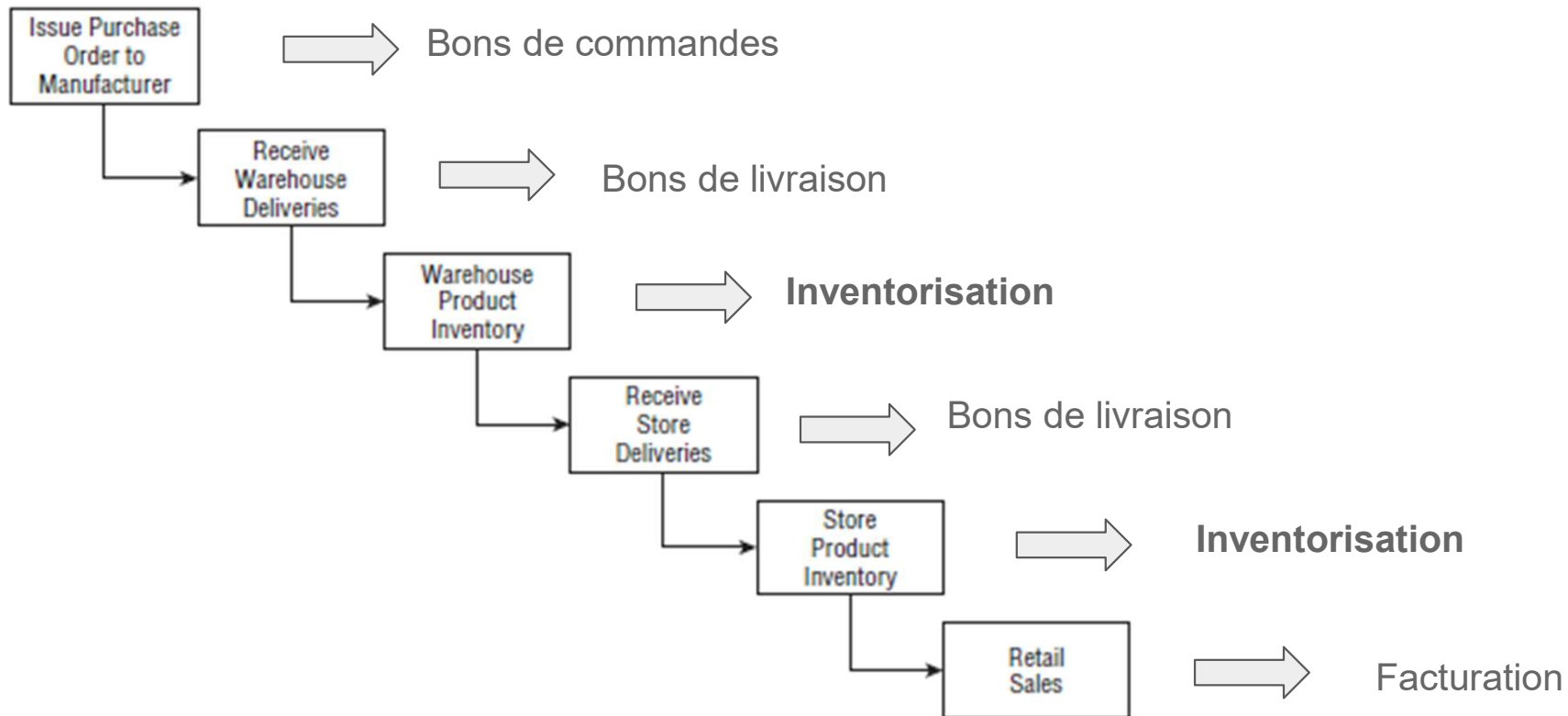


Case study : Inventaire

Chaîne de valorisation

- ❑ Les différents systèmes opérationnels génèrent typiquement des transactions ou des “photographies” (snapshots) à chaque étape de cette chaîne.
- ❑ Puisque chaque processus génère ses propres métriques avec sa propre granularité, chaque processus est susceptible de générer une table de faits différente

Case study : Inventaire



Case study : Inventaire

Importance de l'inventorisation

- ☐ Optimiser les niveaux de stock des produits peut avoir des effets très bénéfiques sur la rentabilité globale
- ☐ Minimisation des risques des ruptures de stock
- ☐ Minimisation des pertes pour les produits périssables
- ☐ Réduction des coûts globaux en matière de transport

Case study : Inventaire

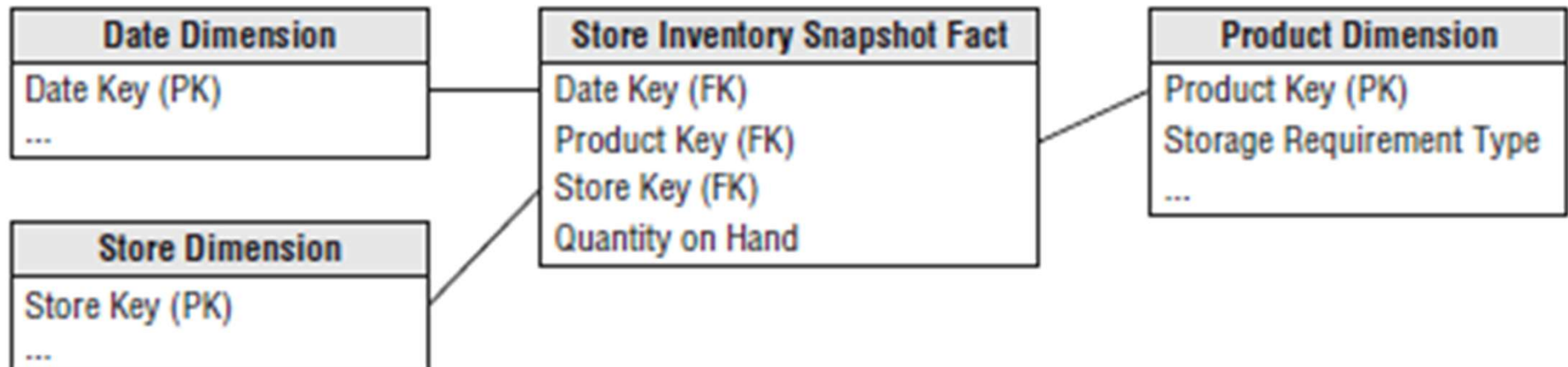
Une première façon de modéliser un processus d'inventorisation est de mesurer le niveau des quantités disponibles => **Snapshot périodique**

Etapas de modélisation :

- ☐ **Identifier le processus métier** : l'état de l'inventaire à différents intervalles de temps
- ☐ **Déterminer la granularité** : le niveau des stock journalier pour chaque produit dans chaque magasin
- ☐ **Identifier les dimensions** : Date, magasin, produit
- ☐ **Identifier les faits** : La quantité disponible

Case study : Inventaire

Modèle d'inventorisation : Snapshot périodique



Case study : Inventaire

Modèle d'inventorisation : Snapshot périodique

- ❑ La dimension date est identique à la dimension date développée dans le modèle précédent
- ❑ La dimension produit est presque identique. Elle peut être agrémentée d'attributs tels que la quantité minimale à commander ou l'espace de stockage requis par exemple à supposer que leurs valeurs soient constantes et discrètes
- ❑ La dimension magasin pourrait inclure des attributs qui indiquent la taille de l'espace de stockage réfrigéré, standard, etc

Case study : Inventaire

Modèle d'inventorisation : Snapshot périodique

- ❑ La table de faits d'un snapshot périodique doit cependant faire face à un énorme challenge : sa densité potentielle
- ❑ Un inventaire journalier de tous les produits dans tous les magasins implique que cette table va croître de manière très significative à chaque insertion (chaque nuit)
- ❑ Une solution possible serait de conserver un historique sur un intervalle moins large (ex : le mois), lequel serait récupéré périodiquement dans une seconde table de fait à la granularité temporelle plus large (ex : la semaine)

Case study : Inventaire

Faits semi-additifs

- ❑ La quantité disponible dans un snapshot périodique peut être additionnée selon les produits ou les magasins
- ❑ Elle ne peut cependant pas être sommée à travers le temps
- ❑ Exemple : On peut faire **la somme des ventes** d'un produit sur une période mais faire **la somme des niveaux d'inventaire** sur un mois par exemple n'a aucun sens
- ❑ Ce type de mesure est appelée semi-additive. Cette nature semi-additive est typique des mesures reflétant un niveau
- ❑ Exemple : le montant disponible sur votre compte en banque

Case study : Inventaire

Coefficient de rotation de stock

- ❑ Si la quantité disponible est une mesure très utile, il est tout de même possible de dégager d'autres mesures très intéressantes de celle-ci
- ❑ En ajoutant la quantité vendue à un snapshot périodique, il est possible de déterminer le **coefficient de rotation des stocks**
- ❑ Journalier => Quantité vendue/Quantité disponible
- ❑ Sur une période de temps (plus intéressant) => Somme des quantités vendues/Moyenne du stock disponible sur cette période
- ❑ Si l'indice est bas => probabilité de surstock
- ❑ Si l'indice est haut => risque de rupture de stock

Case study : Inventaire

Délai moyen avant rupture de stock

- ❑ Dans le même ordre d'idée, on peut obtenir le nombre de jours endéans lequel il faudra recommander un produit
- ❑ Dernière quantité disponible / Moyenne des quantités vendues sur une période = approximation du nombre de jours
- ❑ Si ce nombre est élevé => Pas de commande dans l'immédiat
- ❑ Si ce nombre est faible => Commande urgente

Case study : Inventaire

Vitesse de rotation du stock sur une période

- ❑ Pour une année, si on divise le nombre de jours sur une année par le coefficient de stock annuel on obtient le nombre de jours théoriques nécessaires à la liquidation de ce stock
- ❑ Il est possible de calculer cette vitesse sur d'autres périodes de temps comme le mois
- ❑ Si ce nombre est très élevé => il serait sans doute judicieux de rabaisser le seuil de réserve pour ce produit
- ❑ S'il est très bas => le seuil de réserve doit sans doute être augmenté

Case study : Inventaire

Une deuxième façon de modéliser un processus d'inventorisation serait d'enregistrer les transactions qui l'affectent => **Inventaire transactionnel**

- ☐ **Identifier le processus métier** : Enregistrement des transactions sur le stock d'un entrepôt
- ☐ **Déterminer la granularité** : Une entrée pour chaque événement affectant le stock d'un entrepôt
- ☐ **Identifier les dimensions** : Date, produit, entrepôt, **type de transaction**
- ☐ **Identifier les faits** : L'impact monétaire ou quantitatif d'une transaction

Case study : Inventaire

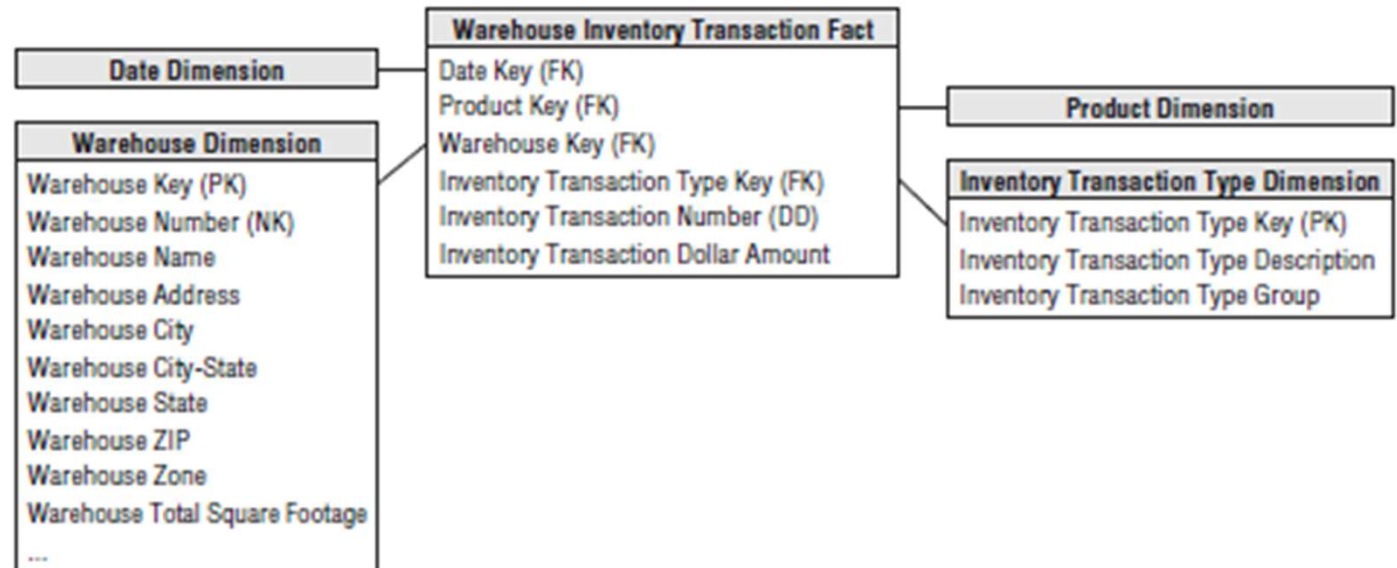
Modèle d'inventorisation : inventaire transactionnel

- ☐ La dimension type de transaction pourrait inclure des transactions du type :
 - ☐ Réception de produit
 - ☐ Envoi d'un produit
 - ☐ Retour d'un produit par un client
 - ☐ Renvoi d'un retour client
 - ☐ Retrait d'un produit d'un inventaire
 - ☐ Etc,...

Case study : Inventaire

Modèle d'inventorisation : inventaire transactionnel

Ce type d'inventaire permet de mesurer les fréquences et les timings de certaines transactions spécifiques



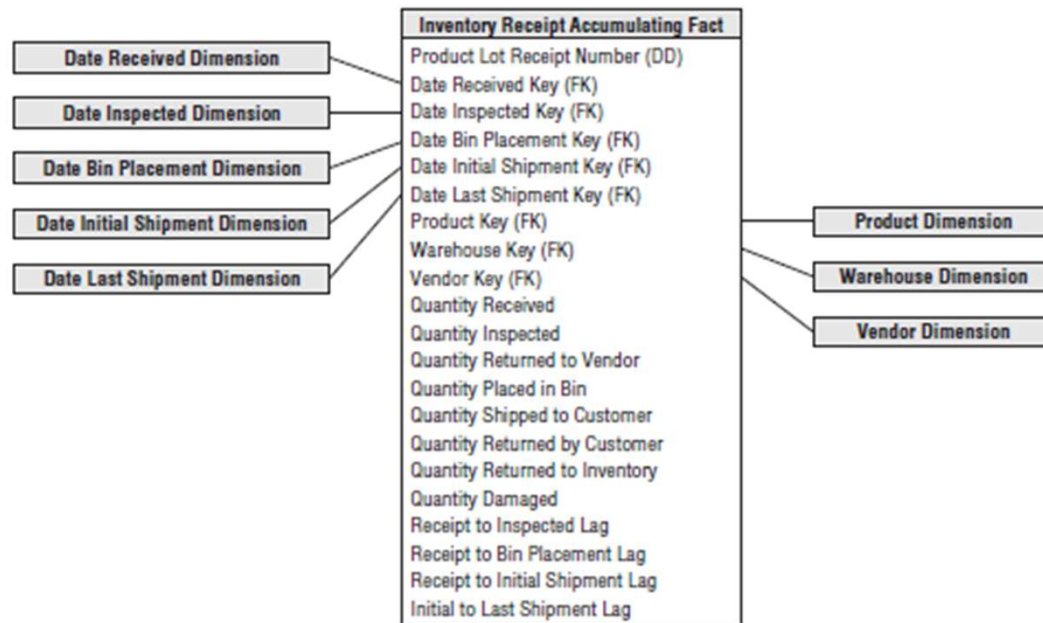
Case study : Inventaire

Une dernière façon de modéliser un inventaire serait d'historiser le cycle de vie d'un produit dans l'entrepôt. De sa réception jusqu'au moment où il "n'existe plus" dans ce contexte => **snapshot cumulé**

- ❑ **Identifier le processus métier** : Historiser le cycle de vie d'un produit au travers de moments clés
- ❑ **Déterminer la granularité** : Une entrée pour un évènement affectant le cycle de vie d'un lot de produits
- ❑ **Identifier les dimensions** : Date, produit, entrepôt, vendeur
- ❑ **Identifier les faits** : Des faits de quantités ou d'intervalles de temps

Case study : Inventaire

Modèle d'inventorisation : Snapshot cumulé



Case study : Inventaire

Modèle d'inventorisation : Snapshot cumulé

- ☐ Une entrée est ajoutée à chaque évènement dans le cycle de vie du produit :
 - ☐ Entrée dans l'entrepôt
 - ☐ Contrôle
 - ☐ Livraison
 - ☐ Retour
 - ☐ Renvoi
 - ☐ Rejet
- ☐ Cela implique plusieurs dimensions temporelles et des mesures au moment de ces évènements

Case study : Inventaire

Bus Matrix et dimensions communes

- ❑ Si les tables de faits sont indépendantes sémantiquement, elles communiquent tout de même “au travers des dimensions”
- ❑ En effet, certains processus métiers font intervenir les mêmes dimensions

BUSINESS PROCESSES	COMMON DIMENSIONS						
	Date	Product	Warehouse	Store	Promotion	Customer	Employee
Issue Purchase Orders	X	X	X				
Receive Warehouse Deliveries	X	X	X				X
Warehouse Inventory	X	X	X				
Receive Store Deliveries	X	X	X	X			X
Store Inventory	X	X		X			
Retail Sales	X	X		X	X	X	X
Retail Sales Forecast	X	X		X			
Retail Promotion Tracking	X	X		X	X		
Customer Returns	X	X		X	X	X	X
Returns to Vendor	X	X		X			X
Frequent Shopper Sign-Ups	X			X		X	X

Case study : Inventaire

Bus Matrix et dimensions communes

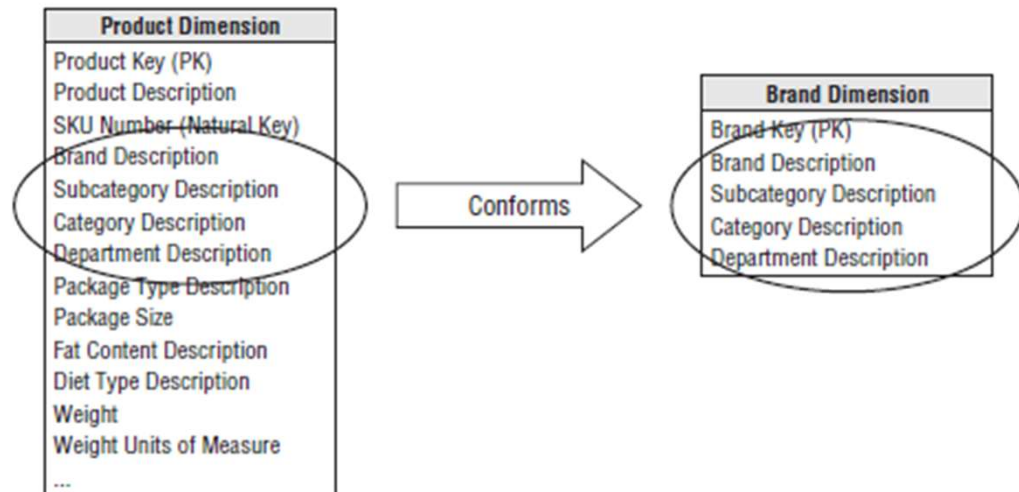
- ❑ La Data Warehouse bus matrix donne une vision globale de l'architecture BI
- ❑ L'uniformité des dimensions va permettre, au niveau du reporting, de “driller” au travers des différentes tables de faits qui possèdent des dimensions communes (drill-across)

Product Description	Open Orders Qty	Inventory Qty	Sales Qty
Baked Well Sourdough	1,201	935	1,042
Fluffy Light Sliced White	1,472	801	922
Fluffy Sliced Whole Wheat	846	513	368

Case study : Inventaire

Bus Matrix et dimensions communes

- ❑ Cette uniformité donne également la possibilité de driller au travers de tables de faits qui communiquent grâce à un sous-ensemble d'attributs de dimensions communs



Case study : Inventaire

Bus Matrix et dimensions communes

- ❑ Cela implique néanmoins au niveau de l'architecture du DataWarehouse une **atomicité homogène** au travers des différentes tables de faits
- ❑ Une labellisation homogène et précise des mesures et attributs
- ❑ Toutes les mesures qui représentent le même calcul doivent d'être nommées de la même manière

Case study Approvisionnement

6

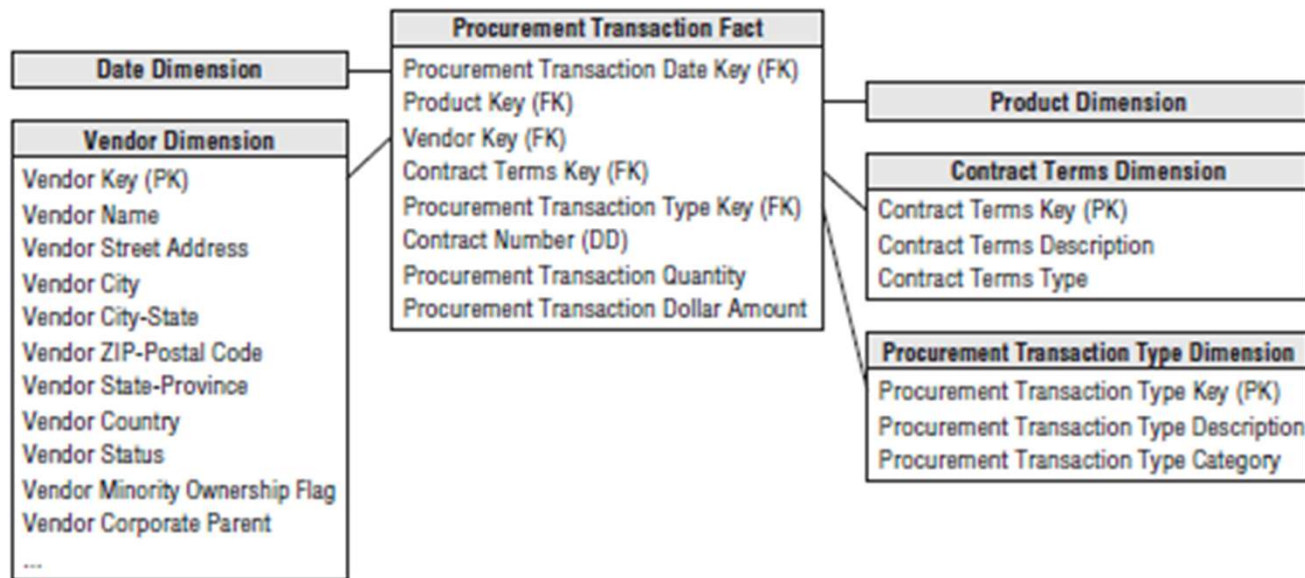
Case study : Approvisionnement

Raisons :

- ❑ Le développement d'une table de faits basé sur l'approvisionnement afin de permettre de répondre aux questions du type :
 - ❑ Qu'achetons-nous le plus? Combien de fournisseurs nous approvisionnent pour tel produit? A quels prix? Existe-t-il des opportunités de négocier des prix plus favorables, en regroupant certaines commandes ou en garantissant des minimas de commandes?
 - ❑ Nos revendeurs respectent-ils bien nos accords commerciaux?
 - ❑ Les prix négociés sont-ils respectés?
 - ❑ Certains fournisseurs sont-ils plus fiables que d'autres dans les délais de livraison? Y'a-t-il souvent des retours de marchandises?

Case study : Approvisionnement

Exemple de modélisation d'une table de faits d'approvisionnement



Case study : Approvisionnement

Exemple de modèle d'approvisionnement transactionnel

- ❑ Les dimensions date et produit sont identiques à celle déjà créées (sauf si les produits livrés sont des produits qui doivent être transformés pour la vente, auquel cas une nouvelle dimension produit peut être judicieuse)
- ❑ Une dimension fournisseur est forcément apparue
- ❑ Une dimension termes de contrat peut être définie

Case study : Approvisionnement

Exemple de modèle d'approvisionnement transactionnel

- ❑ Une transaction listant les différents types de transactions est également envisageable
- ❑ En effet, typiquement, l'approvisionnement va générer différentes métriques dans différents processus métier de l'organisation
 - ❑ Ordres de commandes, accusés de réception de marchandises, notifications de livraisons vers un point de vente, etc
- ❑ Il est possible de rassembler ces faits au sein d'une même table ou de les considérer comme des faits indépendants

Case study : Approvisionnement

Dimension à variation lente

- ❑ Les attributs de dimensions ne sont pas forcément immuables
- ❑ De manière générale, on considérera qu'un attribut de dimensions peut varier de manière ponctuelle
- ❑ Exemple : le prix d'achat auprès d'un fournisseur
- ❑ Il convient au designer d'une solution BI/DW d'évaluer la manière dont un attribut peut varier dans le monde réel afin de traiter, d'anticiper son évolution dans un modèle dimensionnel

Case study : Approvisionnement

Dimension à variation lente

❑ Cas 1 : Réécriture

- ❑ Consiste simplement à remplacer l'ancienne valeur par la nouvelle
- ❑ Il est impossible dans ce cas de conserver la moindre trace d'historisation
- ❑ Attention au fait que cela impacte les valeurs de mesures de la table de faits même si elles ont été intégrées avant la réécriture
- ❑ A n'utiliser que pour les attributs dont on sait à l'avance, que l'ancienne valeur n'a pas d'importance

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name
12345	ABC922-Z	IntelliKidz	Education

Updated row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name
12345	ABC922-Z	IntelliKidz	Strategy

Case study : Approvisionnement

Dimension à variation lente

- ❑ Cas 2 : Ajout d'une nouvelle entrée
 - ❑ Consiste à ajouter des valeurs de début de validité/fin de validité et/ou un booléen de validité

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	...	2012-01-01	9999-12-31	Current

Rows in Product dimension following department reassignment:

Product Key	SKU (NK)	Product Description	Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	...	2012-01-01	2013-01-31	Expired
25984	ABC922-Z	IntelliKidz	Strategy	...	2013-02-01	9999-12-31	Current

Case study : Approvisionnement

Dimension à variation lente

- ❑ Cas 3 : Ajout d'une nouvelle colonne
 - ❑ Consiste à ajouter une colonne pour marquer traquer un changement
 - ❑ Très peu (voire jamais) utilisé

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name
12345	ABC922-Z	IntelliKidz	Education

Updated row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name	Prior Department Name
12345	ABC922-Z	IntelliKidz	Strategy	Education

Case study : Approvisionnement

Dimension à variation lente

❑ Cas 4 : Création d'une mini-dimension

- ❑ Supposons que plusieurs des attributs d'une dimension client puissent changer relativement fréquemment
- ❑ Nous voudrions historiser ces changements sans toutefois à chaque fois ajouter une nouvelle entrée
- ❑ Cela peut arriver lorsque qu'on pratique des analyses de type démographiques
- ❑ Un client a un nom, un prénom
- ❑ mais il également un âge (qui varie tous les ans), une tranche de revenus (qui peut varier au cours du temps), il fait partie d'une certaine catégorie de client, etc

Case study : Approvisionnement

Dimension à variation lente

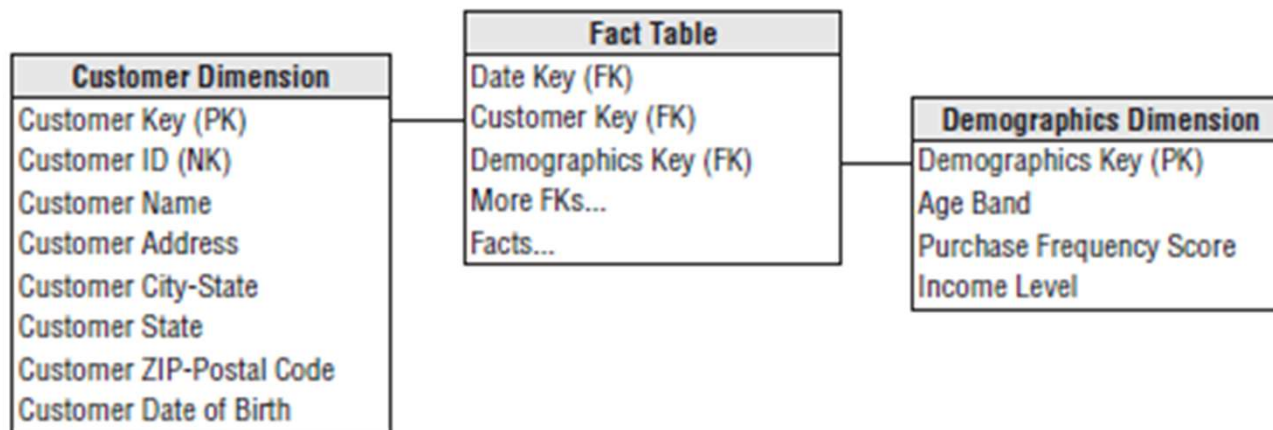
- ❑ Cas 4 : Création d'une mini-dimension
 - ❑ La solution est de créer un sous-ensemble de la dimension client qui reprend ces attributs susceptibles de varier sous la forme d'une liste reprenant les diverses combinaisons possibles et d'y assigner une clé primaire
 - ❑ Cette table ne peut cependant être trop large auquel cas elle possèderait bien trop de combinaisons

Demographics Key	Age Band	Purchase Frequency Score	Income Level
1	21-25	Low	<\$30,000
2	21-25	Medium	<\$30,000
3	21-25	High	<\$30,000
4	21-25	Low	\$30,000-39,999
5	21-25	Medium	\$30,000-39,999
6	21-25	High	\$30,000-39,999
...
142	26-30	Low	<\$30,000
143	26-30	Medium	<\$30,000
144	26-30	High	<\$30,000
...

Case study : Approvisionnement

Dimension à variation lente

- ❑ Cas 4 : Création d'une mini-dimension



Case Study

Gestion des commandes



Case study : Gestion des commandes

La chaîne de valorisation d'une commande pourrait être présentée comme suit :

1. Remise d'un devis
2. **Réception d'un commande => génère un bon de commande**
3. Envoi de la commande
4. **Facturation => génère une facture**
5. Réception des paiements
6. Retour marchandises

Les processus métier sur lequel nous allons travailler sont **les commandes d'un client** et **la facturation d'une commande**

Case study : Gestion des commandes

Etapes de modélisation

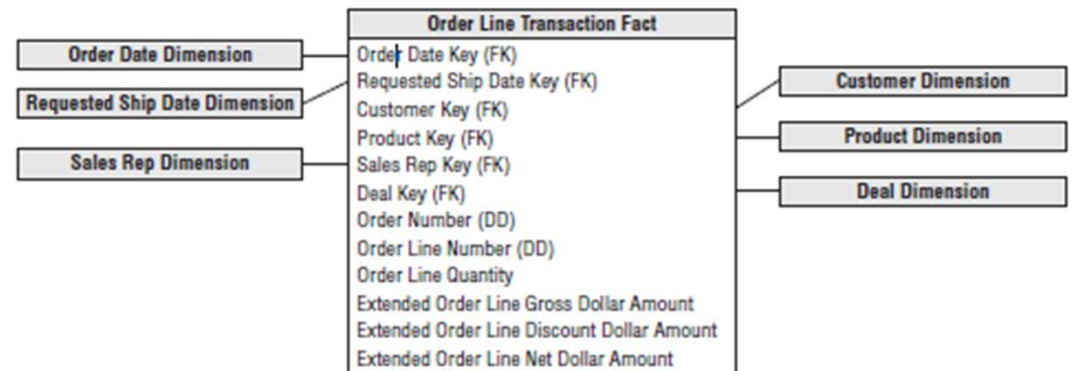
- ☐ Le premier processus métier identifié est **la commande d'un produit**
- ☐ La granularité de la table est une entrée pour une ligne sur une commande
- ☐ Les dimensions identifiables sont :
 - ☐ La date de commande
 - ☐ La date d'envoi
 - ☐ La date prévue de livraison
 - ☐ Le produit
 - ☐ Le client
 - ☐ Le délégué commercial
 - ☐ Les contrats conclus entre le client et le délégué

Case study : Gestion des commandes

Etapas de modélisation

❑ Les faits, de manière générale, seront :

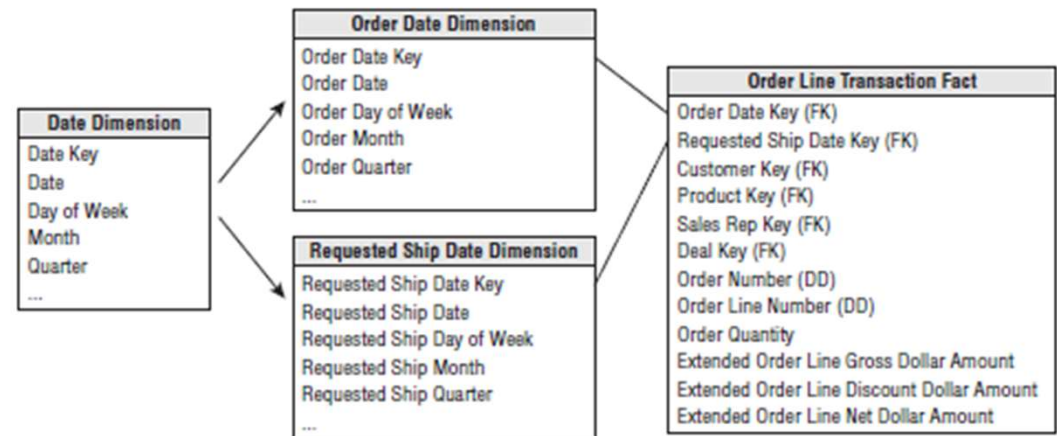
- ❑ La quantité commandée
- ❑ Le total brut
- ❑ Les remises
- ❑ Le total net (total brut - remises)



Case study : Gestion des commandes

Dimension à rôle multiples

- ❑ Dans ce cas, la dimension date joue 3 rôles :
 - ❑ La date de commande
 - ❑ La date d'envoi
 - ❑ La date prévue de livraison
- ❑ Il est nécessaire de dupliquer la dimension date (via des vues) parce qu'elle est différente pour chacun des 3 champs de la table de faits



Case study : Gestion des commandes

Dimension produit

- ❑ La dimension produit est une des dimensions les plus importantes dans ce genre de modèle
- ❑ Cette dimension peut être très variable selon le type d'entreprise
- ❑ Un fabricant peut avoir des milliers de variations pour ses produits tandis qu'un autre peut avoir des millions d'entrées unique pour chaque produit (une entreprise qui vend des produits de sur-mesure par exemple)

Case study : Gestion des commandes

Dimension produit

❑ Rappel des bonnes pratiques

- ❑ Résister à l'envie de normaliser, inclure les hiérarchies dans la table dénormalisée
- ❑ Libeller les colonnes de la manière la plus explicite qui soit
- ❑ Ne pas oublier d'inclure des clés artificielles
- ❑ Décrire et/ou décomposer les codes opérationnels (clés naturelles)
- ❑ Eviter les valeurs null dans les attributs (source de mauvaises interprétations)
- ❑ Homogénéité des valeurs des colonnes, en effet ces valeurs sont les ressources premières en matières de tris et de regroupement dans les requêtes. Des valeurs non-homogènes risquent de biaiser les résultats

Case study : Gestion des commandes

Dimension client

- ❑ Cette dimension va typiquement inclure les coordonnées de localisation d'un client. Cela va forcément générer de la redondance. Rappelons que cela est totalement acceptable dans un modèle dimensionnel
- ❑ Ces coordonnées de localisation sont généralement incluses dans une hiérarchie
- ❑ Il est parfois judicieux d'inclure également une hiérarchie organisationnelle du client (si celui-ci fait partie d'un groupe par exemple)

Case study : Gestion des commandes

Dimension client

- ❑ Il est également fréquent que les adresses de livraisons soient différentes des adresses de facturation. Il faut donc en tenir compte dans le modèle

Customer Dimension
Customer Key (PK)
Customer ID (Natural Key)
Customer Name
Customer Ship To Address
Customer Ship To City
Customer Ship To County
Customer Ship To City-State
Customer Ship To State
Customer Ship To ZIP
Customer Ship To ZIP Region
Customer Ship To ZIP Sectional Center
Customer Bill To Name
Customer Bill To Address
Customer Organization Name
Customer Corporate Parent Name
Customer Credit Rating

Case study : Gestion des commandes

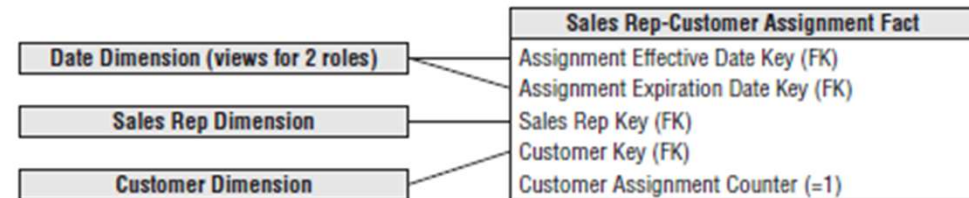
Dimension client

- ❑ Il est également possible d'inclure le délégué commercial et ses caractéristiques en tant qu'attribut du client
- ❑ Cela n'est toutefois possible que s'il existe une relation one-to-one ou one to many clairement définie entre le client et le délégué, et que cette relation varie peu
- ❑ Dans le cas d'une relation many-to-many, on privilégiera la mise en place d'une dimension "délégué" séparée

Case study : Gestion des commandes

Dimension client

- ❑ Si les utilisateurs veulent mesurer l'efficacité de leur délégués, il faut tenir compte des cas où un client leur est assigné mais où aucune vente n'est conclue
- ❑ On peut représenter cela par une **table de faits sans fait** qui identifie un processus d'assignation d'un délégué à un client



- ❑ Avec la requête appropriée, il est ainsi possible d'identifier les clients avec lesquels encore aucun contrat n'est conclu

Case study : Gestion des commandes

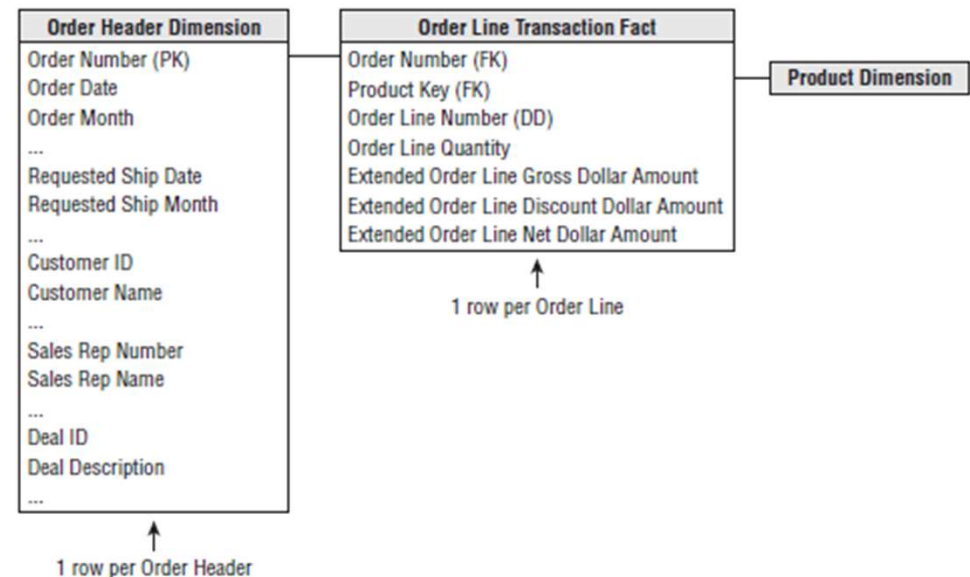
Dimension contrat

- ❑ Semblable à une dimension promotion, cette dimension décrit les incitants, les bonus, les accords conclus entre un client et un délégué afin de favoriser les ventes
- ❑ Comme pour une dimension promotion, ne pas oublier le cas où une vente est conclue sans accord particulier

Case study : Gestion des commandes

Pattern à éviter

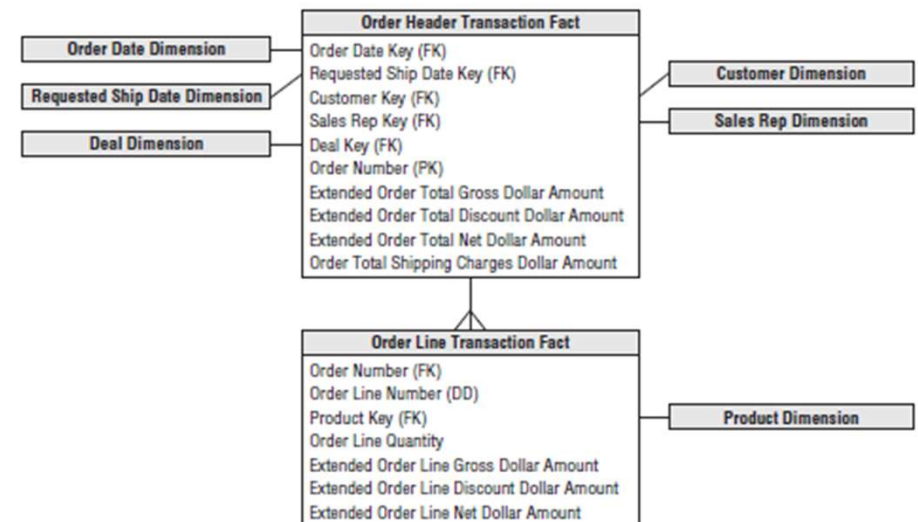
- ❑ La dimension header - qui-plus-est particulièrement large - va obligatoirement croître avec chaque nouvelle transaction
- ❑ A supposer qu'une commande soit en moyenne composée de 5 lignes, cette table header sera équivalente à 20% de la table de faits
- ❑ La moindre requête suppose de traverser cette énorme dimension



Case study : Gestion des commandes

Pattern à éviter

- ❑ Même si ce design est assez représentatif de la relation entre un header et une ligne, le simple fait de vouloir analyser une commande sur une dimension autre que le produit nécessite de joindre les 2 tables de faits



Case study : Gestion des commandes

Etapes de modélisation

- ❑ Le second processus métier identifié est la **facturation d'une commande**

La facturation rassemble énormément d'information très pertinentes

- ❑ Mesures : Prix de revient, coût de distribution, ristourne, etc
- ❑ Dimensions : Client, produits, fournisseurs, entrepôt

Pour les entreprises qui livrent/envoient leurs produits à leurs clients, la facturation est souvent le meilleur endroit pour démarrer un projet de business intelligence

Case study : Gestion des commandes

Etapes de modélisation

- ☐ La granularité de la table est une entrée pour une ligne sur une facture
- ☐ Les dimensions identifiables sont :
 - ☐ La date (qui aura plusieurs rôles comme dans la table précédente)
 - ☐ Le produit
 - ☐ Le client
 - ☐ Le délégué commercial
 - ☐ Les contrats conclus entre le client et le délégué
 - ☐ L'expéditeur
 - ☐ L'entrepôt
 - ☐ Le niveau de service (qui capture la qualité des livraisons en mesurant les délais par exemple)

Case study : Gestion des commandes

Etapas de modélisation

- ☐ Les faits identifiables sont nombreux :
 - ☐ Quantité livrée
 - ☐ Montant total brut (prix unitaire * quantité)
 - ☐ Montant total des réduction (sur et/ou hors facture)
 - ☐ Montant total facturé au client
 - ☐ Montant total des coût de fabrication
 - ☐ Montant total des coût de distribution
 - ☐ Montant total des coût de stockage
 - ☐ Etc,...

Case Study: Comptabilité



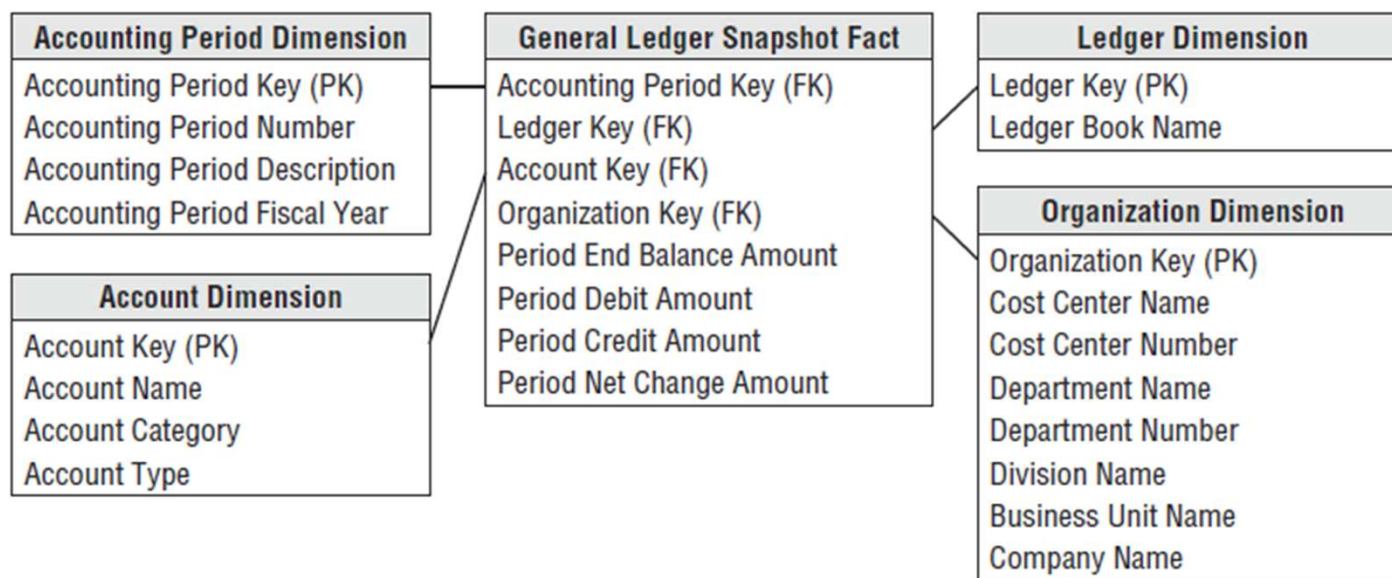
	<i>Date</i>	<i>Ledger</i>	<i>Account</i>	<i>Organization</i>	<i>Budget Line</i>	<i>Commitment Profile</i>	<i>Payment Profile</i>
General Ledger Transactions	X	X	X	X			
General Ledger Snapshot	X	X	X	X			
Budget	X	X	X	X	X		
Commitment	X	X	X	X	X	X	
Payments	X	X	X	X	X	X	X
Actual-Budget Variance	X	X	X	X			

Grand Livre

- Snapshot périodique du grand livre pour montrer une 'vue' de fin de période. Le niveau de détail (grain) est une ligne par période par compte
- Plan comptable: clé de voûte de la comptabilité. Numéros de comptes 'intelligents', hiérarchiques.
- Organisation: dimension clé d'une société. Dimensions non-alignées avec le plan comptable (frustration management local)
- Clôture de période: dans la plupart des sociétés les données sont mis dans le DW pour vérifier l'exactitude (aiguille ds botte de foin), quitte à corriger après.
- Faits YTD: à éviter: donnent une fausse image. Devrait toujours être calculé dans le reporting ou dans les cubes.
- Devises multiples: chaque événement dans une autre devise devra contenir son équivalent dans la devise principale pour faciliter le reporting. Prévoir bien sûr une clé vers la dimension devises.
- Table de faits des transactions du journal des comptes: contient uniquement un débit ou un crédit, rien d'autre.
- Calendrier fiscal (multiple): 2 approches possibles. (1) Inclure le calendrier fiscal dans la dimension date (pour chaque business unit, avec dates de début et de fin par période) ou (2), créer une dimension séparée pour chaque succursale avec une clé étrangère (FK) dans la table des faits
- Drilling down (forer, percer) / drill up à travers une hiérarchie multiples niveaux (département, division, entreprise, géographie,...). La ligne pourrait par exemple contenir la clé (surrogate key) vers le niveau inférieur/supérieur pour faciliter l'analyse
- États financier. Le DW ne peut se substituer au système comptable, mais peut néanmoins fournir une vue agrégée plus facilement visible et partageable.

Comptabilité

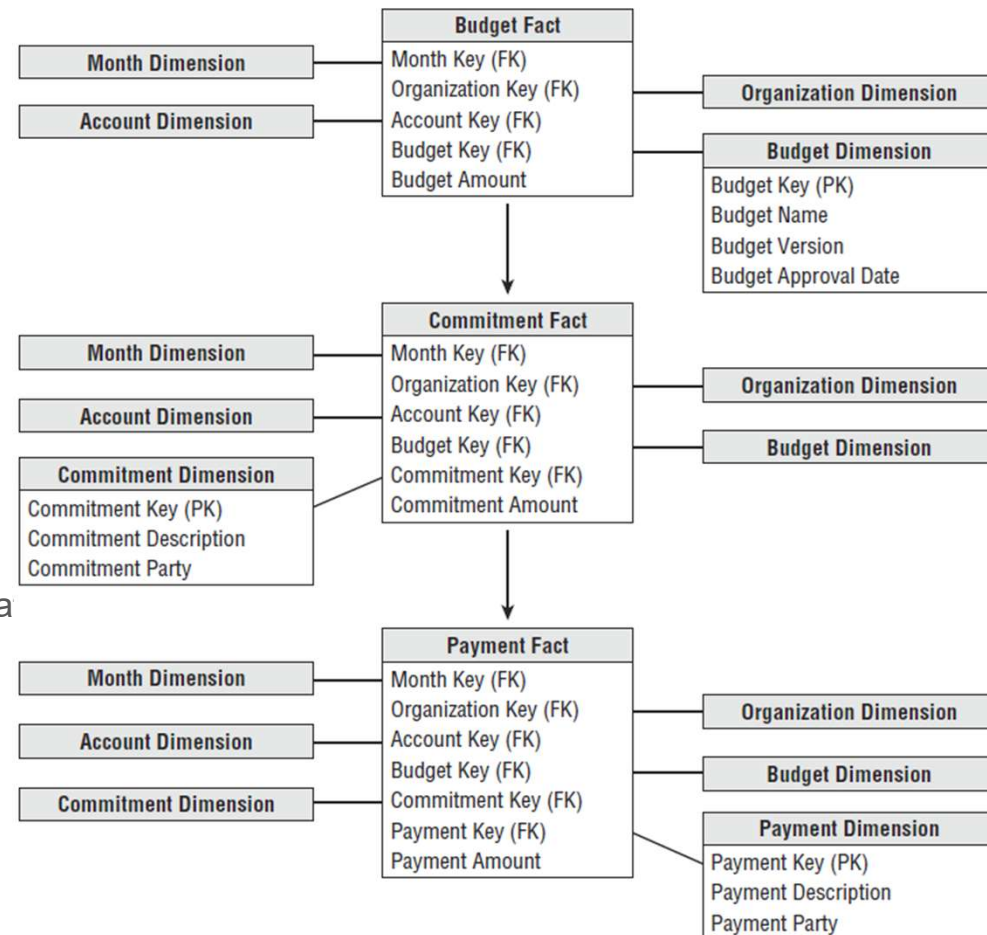
- Momentané périodique grand livre



Comptabilité

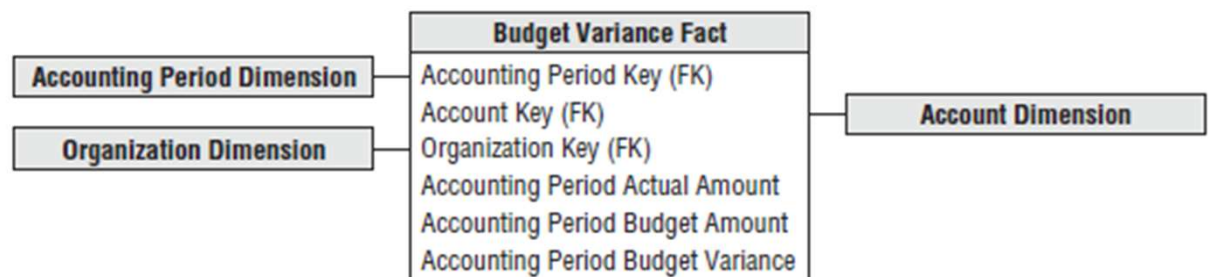
Processus budgétaire.

- Souvent une séries d'événements:
 - création du budget ligne par ligne
 - approbation du budget
 - engagement du budget
 - paiement
 - révision du budget
- Montant additif; première ligne montant initial, puis montant en + ou - du budget initial
- Souvent lié à 1 ou + comptes G/L (réconcilia
- Table de faits 'engagement' avec les mêmes dimensions pour faire du monitoring
- Argent plus dispo quand 'commit', mais dept. finance veut analyser lien entre commit et paiement



Comptabilité

- Consolidated Fact Tables
 - Si le business fait beaucoup de 'drill-across' (ex. comparaison budget vs promesses), il peut être intéressant de créer une table de résultats consolidés, au lieu de laisser la tâche d'agrégation à l'outil de reporting ou à l'utilisateur.
Ces tables de résultats consolidés devront tenir compte éventuellement des différentes devises utilisées
 - Si cette consolidation porte sur plusieurs processus business, les faits doivent être du même niveau. (exemple?)



Notions avancées sur les hiérarchies



Techniques de modélisation dimensionnelle Kimball

- Dimensions hiérarchiques
 - Hiérarchie avec nombre de positions fixe, avec des noms de niveau significatifs.

Exemple:

- jour → période fiscale → année.
- jour → mois → année.

Ce sont 2 hiérarchies différentes sans lien, car dépendant du calendrier fiscal de l'entreprise. Les deux (ou +) hiérarchies peuvent se trouver dans la même table de dimension date, car la granularité est la même (jour).

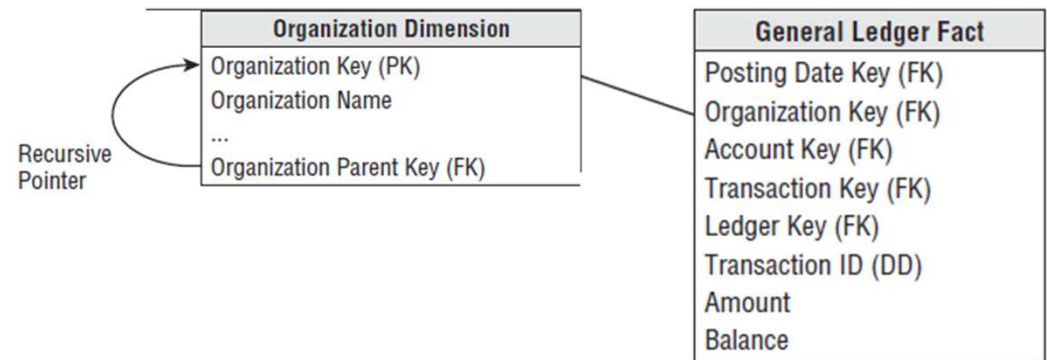
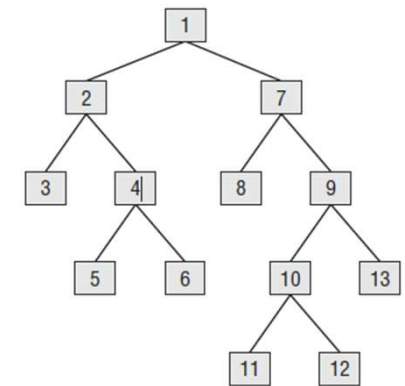
A éviter: le noms comme 'Niveau 1' ou 'Niveau 2' → non-significatifs!

Techniques de modélisation dimensionnelle Kimball

- Dimensions hiérarchiques
 - Hiérarchies variables
 - On associe un fait à une dimension et on peut dans la dimension mettre une référence récursive

Les (+) et les (-) de cette approche récursive:

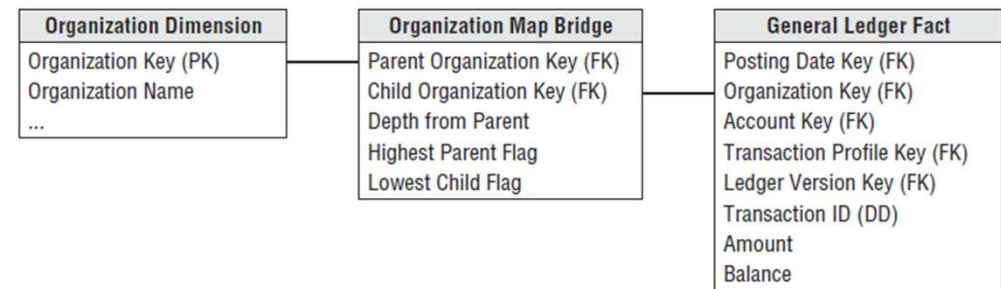
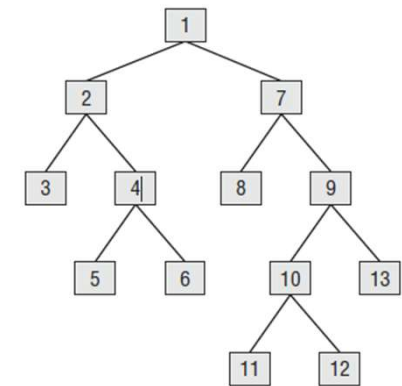
- (+) les relations hiérarchique sont dans la table de dimension
- (+) maintenance simplifiée pour les changements de structure
- (-) besoin de requêtes SQL compliquées, et donc performances (-)
- (-) pas de parents partagés (par ex. pas de compte bancaire commun)
- (-) très difficile pour gérer des hiérarchies changeantes dans le temps (SCD Type 2)



Techniques de modélisation dimensionnelle Kimball

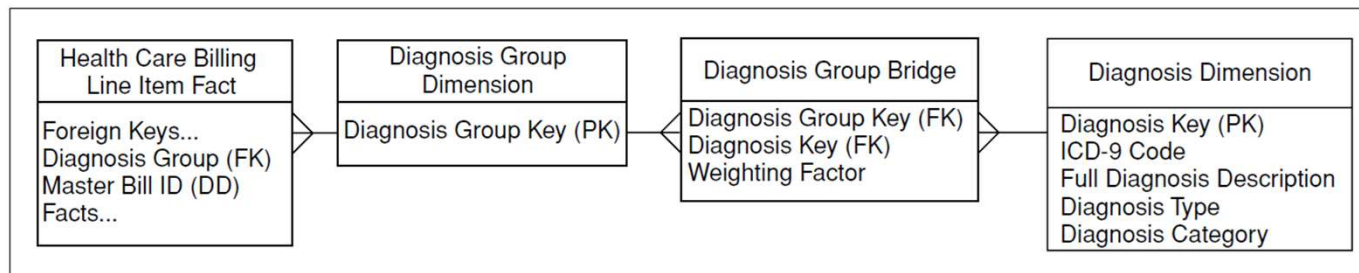
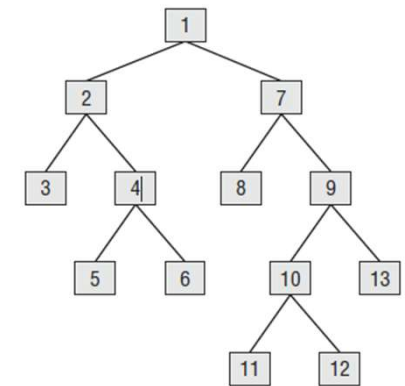
- Dimensions hiérarchiques
 - Hiérarchies variables
 - ou insérer une table intermédiaire ('bridge'), aussi avec ses (+) et ses (-)

(+) queries SQL standard pour évaluer la hiérarchie
 (+) peut être utilisé pour des hiérarchies partagées (voir après)
 (+) maintenance 'facile' de la dimension (SCD type 2), car contenu dans la table dimension, pas dans la table bridge
 (-) besoin de créer un record pour chaque relation enfant-parent dans la table bridge, ainsi que pour les multi niveaux
 (-) plus complexe pour ajouter ou déplacer une partie de la structure
 (-) mise à jour nécessaire si il y a une SCD Type 2 dans la dimension principale



Techniques de modélisation dimensionnelle Kimball

- Dimensions hiérarchiques
 - Hiérarchies variables partagées. Avec une ou deux tables intermédiaires (pour grouper).
Difficulté supplémentaire dans le processus ETL pour maintenir les partage (weighting, ownership,...)



Techniques de modélisation dimensionnelle Kimball

- Dimensions hiérarchiques
 - Hiérarchies variable dans le temps: ajout de date/heure de début et de fin
 - Attention: les requêtes DOIVENT impérativement être contraintes à une seule clé dans la dimension date/heure pour ne retourner qu'un seul record
 - Processus ETL!!
 - A envisager: même hiérarchie avec partage. !! Processus ETL pour maintenance !!

