Le design de datawarehouse

Objectifs

Présentation

- Les sources des données
- Fact vs Dimension
- Les types de dimensions
- Les Fact Tables
- Les clés dans une datawarehouse
- Quelques considèrations de design

1.

Le design de datawarehouse

Commençons à découvrir les divers aspects de ce design!!!

“

Lab: Créer une première Datawarehouse

Les sources de données

La problématique

- Quel type de données sources peuvent alimenter des cubes et des tableaux
- En principe la principale source de données qui alimente une DW est une base de données OLTP “Online Transactional Processing”
- Les types des bases OLTP:
  - Des fichiers
  - Une base de donées relationelle
  - Un ERP
- Un système OLTP est souvent de forme complexe en se contente de le réduire en base de données
- La forme de la base change à travers le temps, I faut prendre en considèration les changements, par fois on est conforenté à des non conformités
- Souvent on est confrenté à des systèmes qui comportent des données erronées, considèrons un système ERP de plusieurs années sans une vraie gestion normalisée
- La question est ou le nétoyage doit être fait?
  - Le système OLTP
  - Le processus ETL

Les sources de données

La problématique

- Souvent les systèmes OLTP sont sous documentés ou non documenté
- Outre que la documentation, il faut prendre en considèration la validation et les normes de données
- Quel type de produit nous devons générer
  - Des cubes
  - Des tableaux
- Quel model devons nous adopter
  - Model Kimbal
  - Model Inmon
- Deux livres à absoluent consulter
  - Inmon: Building the Datawarehouse
  - Kimbal: The datawarehous toolkit Kimball
- Dans tout les cas on fini par consommer des datamarts à la fin

Facts vs Dimensions

Les types d'entités

❑ Il existe deux types d'entités
- Les dimensions
- Les Facts

❑ Une dimension est généralement une collection d'attributs qui décrivent un phénomène à étudier quantitativement

Exemple: couleur, temps, poid, taille

❑ Une dimension est représentées par une table , avec une clé outre que la clé naturelle, elle s'appelle clé **Surrogate**

❑ La clé **Surrogate** sera utilisée pour lier la dimension à la table Fact, elle est en général un entier

❑ Le Fact est souvent expliqué par
- Une chose qui c'est passé
- Une chose meusurable

❑ Exemple: Vente d'un p roduit, la moyenne des poids , le chiffre d'affaire du meilleur produit durant les trois derniers mois

❑ Les Facts rèsident dans les tables Fact en principe

Facts vs Dimensions

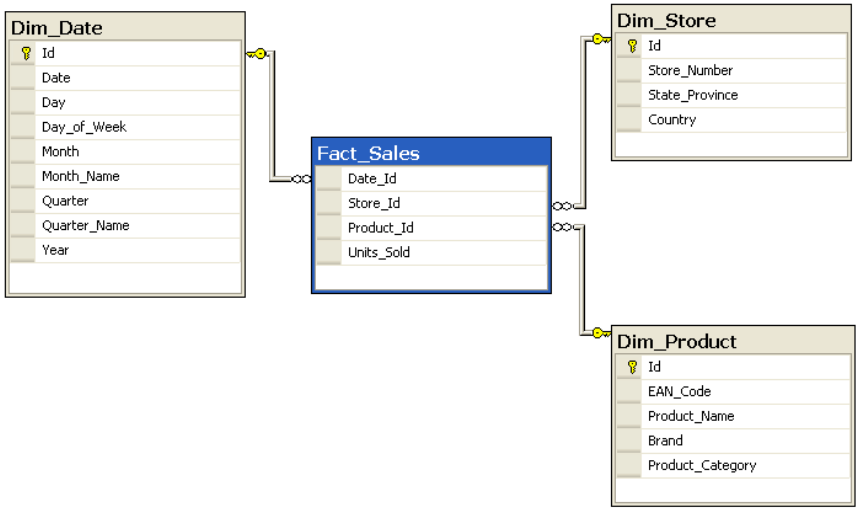


Figure qui montre un exemple de schéma en étoile

Facts vs Dimensions

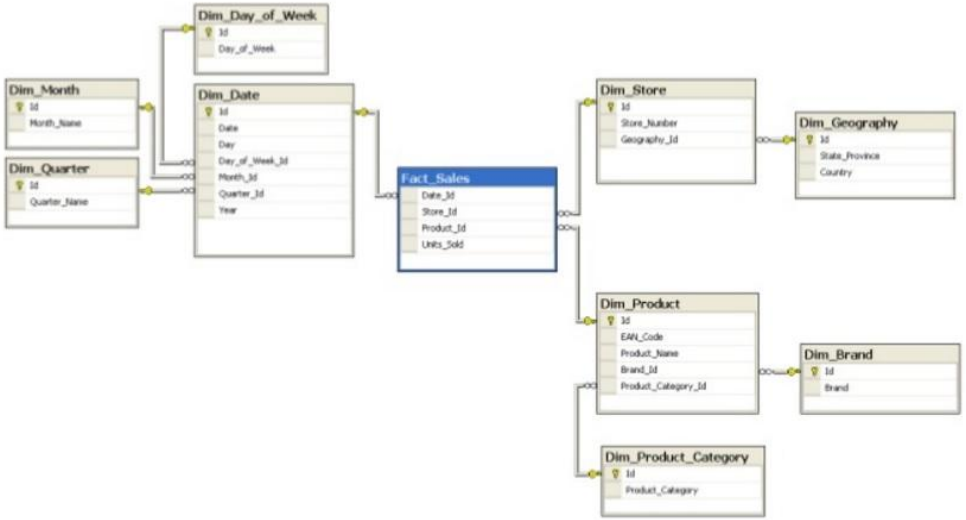
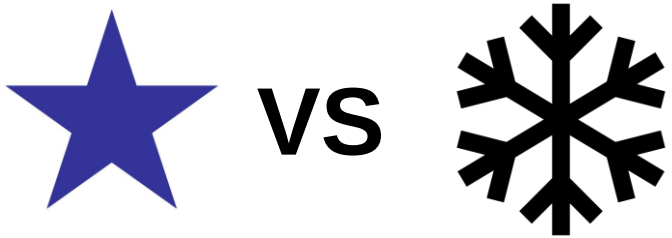


Figure qui montre un exemple de schéma en étoile

Facts vs Dimensions

Les différences entre les schémas

- Les schémas en étoiles sont caractérisés par la simplicité de lecture pour l'être humain et les performances pour le moteur d'analyse
- Les schéma flocon de neige sont plus compliqués et peuvent mener vers des erreurs de processsing lors des générations des cubes
- Le point en commun c'est le modèle d'étoile en base
- Le point de différence c'est que pour certaines dimensions le lien n'est pas direct avec le Fact



Facts vs Dimensions

Les questions à poser

- Quand est ce que nous sommes obligés d'utiliser le schéma Flocon de neige?
- Comment utiliser les dimensions du second rand?
- Est-ce qu'il y a des possibilités d'adaptations de schéma flocon de neige vers des schémas en étoile?
- Quelle recommandations à suivre pour modéliser un schéma DW?

Facts vs Dimensions

Les réponses

- ❑ Les schémas flocons de neige sont principalement utilisés pour:
  - ✓ Des raisons de normalisation « exemple: Store-Geography »
  - ✓ Des raisons de performance
  - ✓ Des raisons d'organisation « Hiérarchie »

- ❑ Les dimensions de second rand sont utilisées seulement pour ajouter une organisation hiérarchique à celles du premier rand

- ❑ Il est possible de générer des vues pour présenter un schéma flocon de neige en model en étoile

- ❑ Il est possible de générer des vues de sources de données pour présenter un schéma flocon de neige en model en étoile

Facts vs Dimensions

Les réponses

- Il y on a qui optent à créer de nouvelles dimensions alimentées par les anciennes mais cette solution pour mener à des dégradations de performances et des complexités de modélisations
- La règle est que toujours il faut garder la solution la plus simple que possible

Les types Dimensions

Les types de dimensions

- ❑ Junk dimensions
- ❑ Degenerate dimensions
- ❑ Slowly Changing dimensions
- ❑ Bridge tables Factless-Fact tables
- ❑ Time dimensions

Les Junk dimensions

- ❑ Après avoir finit le processus de modélisation, nous finirons souvent de trouver quelques attributs qui ne font partie à aucune dimension

- ❑ Normalement ces attributs ne sont pas de grand nombre, ni d'une quantité de données considérable

- ❑ La bonne pratique de ne pas les ignorer

- ❑ Comment les intégrer dans le model?
- ✓ Créer une dimension pour chaque attribut
- ✓ Les regrouper dans une dimension « Junk Dimension »

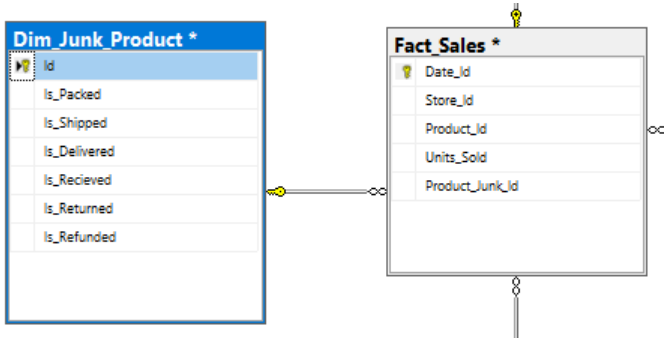
Les types Dimensions

Pourquoi les Junk dimensions

- ❑ Réduire le nombre de dimensions
- ❑ Impact de réduction sur le nombre de lignes ajoutées au niveau de la Fact
- ❑ Simplifier la vie de l'utilisateur final

Remarque: Cette solution n'est pas l'idéal, lorsqu'on change d'avis en retirant l'un des attributs dans une dimension à part, dans ce cas il faut penser

- ✓ L'impact sur le design et les performances
- ✓ Les dépendances des autres produits comme les rapports



Les types Dimensions

Les Degenerate dimensions

- Parfois nous aurons recourt à des colonnes au niveau des tables Fact qui n'ont pas de dimensions relatives
- Ce sont des normalement des clés comme les numéros d'ordre, les numéros de transactions effectuées
- Parfois en les nécessite, exemple le calcul du total de ventes du produit X par transaction
- Il est nécessaire aussi pour retourner vers OLTP pour collecter des informations
- Par contre il ne faut pas en abuser pour ne pas aboutir à des problèmes de performance
- Ce genre d'attribut est conseillé d'être utilisé pour des raisons de Reporting que pour des requêtes, vue la cardinalité presque la même de Fact
- Pour empêcher l'utilisation des dimensions dégénérées, il est possible de mettre la propriété **AttributeHierarchyEnabled** à **False**

Les types Dimensions

Les dimensions SCD

- ❑ Parfois un client change d'adresse et ce genre de changement n'est pas effectué souvent en général
- ❑ Dans ce cas il y a deux façons de gérer

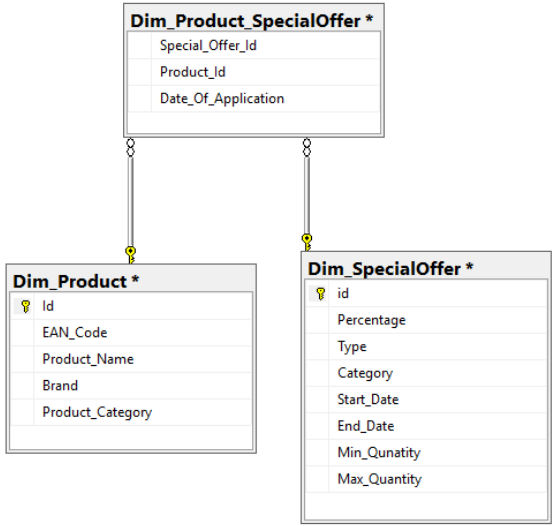
- ✓ Ecraser les anciennes données
- ✓ Créer une nouvelle ligne qui représente les modifications
- ✓ Garder trace de la dernière valeur avant le changement

- ❑ Concernant les fréquences d'utilisation, le type 1 est plus adopté que le type 2 et 3
- ❑ Pour certain cas, nous aurons à utiliser la même dimension avec plusieurs cubes chacun à son propre besoin en terme de SCD
- ❑ Généralement pour le type 2 on ajoute un champ à notre table de dimension pour indiquer la fin de validité d'ancien enregistrement

Les types Dimensions

Les factless tables

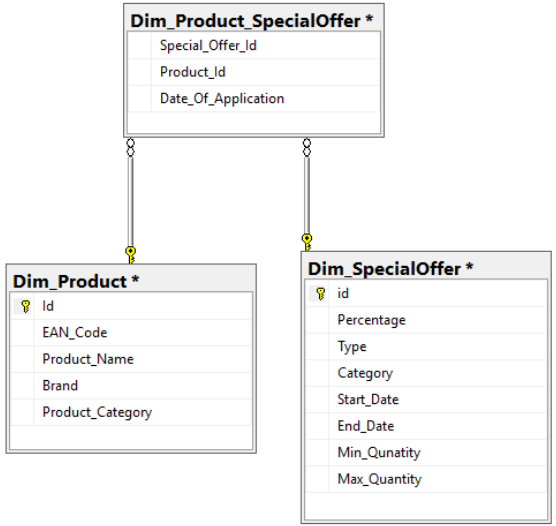
- A ne pas confondre avec les dimensions classiques du deuxième rang du schéma flocon de neige
- A ne pas confondre aussi avec les Fact car ce genre de tables porte le nom de Factless-Fact
- Ce genre de table représente une relation many-to-many entre deux dimensions
- Normalement les tables Fact assurent la relation many-to-many entre la totalité des dimensions



Les types Dimensions

Les factless tables

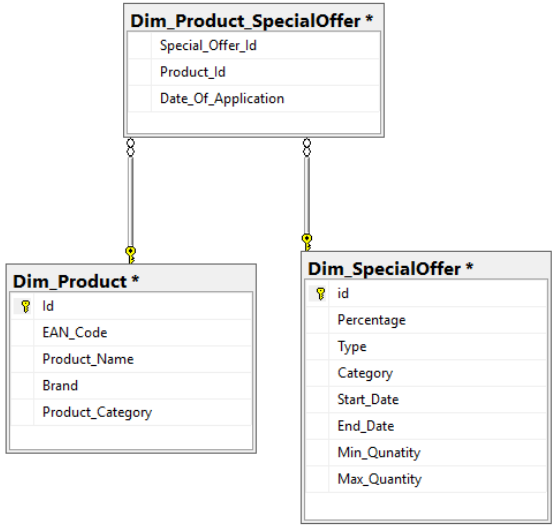
- Le nom Factless veut dire que les bridge tables jouent presque le même rôle que les Fact à l'exception qui ne comportent pas de mesures
- La question est quel est l'utilité des bridge tables?
- Il est possible de le faite et il est correct
- On suppose stocker trois informations à propos de produit, sales et offre spéciales



Les types Dimensions

Les factless tables

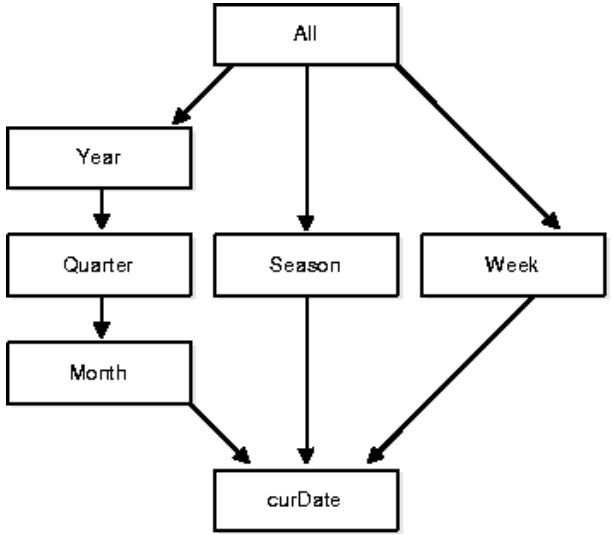
- Si on suppose que certaines offres spéciales non pas aboutis à des sales dans ce cas on perd des informations décisionnelles très précieuses à propos des offres spéciales qui ne marchaient pas
- De même toujours dans la même situation, on garde pas trace d'une relation entre produit et offre spéciale car la seule info qu'on trouve au niveau de Fact c'est l'acte de vente et sa date
- L'information d'absence d'un Fact est parfois plus importante l'information de sa présence



Les types Dimensions

Les dimensions temps

- Les dimensions sont de deux types
  - Calendrier
  - Fiscal



Les types de Fact

Les Transaction Fact

- Une transaction Fact est une table qui enregistre les événements, achat de produit, vente de produit y compris les mesures
- L'enregistrement est effectué pour chaque opération faite

Les Snapshot

- Les Snapshot représentent les divers états d'un même phénomène étudié suivant un repère donné
- Le repère peut être spécial, comme le chiffre d'affaires des ventes d'un produit X par région
- Le repère peut être temporel, comme le chiffre d'affaires des ventes d'un produit X par trimestre
- Les Snapshot sont utilisées aussi pour effectuer d'impact indirect entre deux phénomènes étudiés

Les types de Fact

Les Snapshot

- Exemple, l'impact des augmentation des coûts d'approvisionnements de la matière Y sur les ventes du produit X
- Les Snapshot présentent les informations sous forme d'agrégations

Avantages et inconvénients

- Parmi les avantages, c'est la réduction de nombre d'enregistrements
- Parmi les avantages, l'amélioration des performances
- Parmi les inconvénients, la perte des traces de données élémentaires, exemple la transaction X relative à la vente d'un produit Y à un instant T

Les mises à jour des tables

Le monde de la théorie

- ❑ L'idéal que nous utilisons uniquement des insertions
- ❑ Le monde réel est un peut différent, exemple les SCD
- ❑ Il existe deux types de mise à jour
  - ✓ Les mises à jour de données
  - ✓ Les mises à jour structurelles

Le monde réel

- ❑ Parfois nous nécessitons l'ajout de nouvelles mesures, ou des nouveaux attributs de dimensions
- ❑ Parfois nous nécessitons la correction des données entrées

Les mises à jour des tables

Le monde réel

- Il faut distinguer les tailles des mises à jour effectuées relativement aux tables dimension ou Fact
- Pour les tables dimension, nous parlons de 100 000 milliers
- Pour les tables Fact nous parlons des 1000 000 millions d'enregistrements
- Concernant la nullité, il est possible de préserver une valeur par défaut
- Le soucis qui persiste concerne la taille des Fact, surtout lorsqu'il s'agit de création d'un nouveau champ même avec une valeur par défaut
- Dans ce cas, la meilleure altérative est de recharger la table Fact entière plutôt que la mettre à jour , sa s'appelle « One Shot »
- A condition que nous détenons les permissions

Les clés

Les types de clés

- ❑ Il existe deux types de clé
  - ✓ Clés naturelles
  - ✓ Clés Surrogate
- ❑ La clé naturelle est la clé primaire représentées dans la table dans le contexte relationnel
- ❑ La Surrogate Key est une clé qu'on ajoute au niveau de la DW
- ❑ Elle est représente l'incrémentation sous forme entier selon Kimball
- ❑ Exemples de formes de Surrogate Key
  - ✓ Pour objet: Incrément
  - ✓ Pour date: YYYYMMJJ
  - ✓ Pour rangés: ID-Valeur de rangé
  - ✓ Pour Junk: ID-Table1IDTable2IDTableN

Les clés

La nullité et les erreurs

- La question posé est ce qu'il est nécessaire d'implémenter les clés étrangères comme contraintes
- Il y a deux opinons dans ce cas, certains disent oui s'il on compte sur un bon design ETL pour gagner en terme d'intégrité et de cohérence
- Les autres disent non car nous avons besoin des clés étrangères non nulles en cas de mise à jour et d'insertion et pour éviter les conséquences des erreurs ETL
- En cas ou la base est en lecture seule, il est possible d'accepter la nullité des clés étrangères
- Certain autres disent, il faut ajouter une phase de validation des valeurs des clés au niveau ETL
- En cas d'utilisation des cubes, il faut bien définir les contraintes de clé étrangères pour éviter les erreurs de procès de cubes ou les procès partiel des cubes
- Les erreurs ne sont pas tolérés, dans ce cas si nous activons les vérifications au niveau des clés étrangères nous n'aurons pas d'erreurs

“

Lab: Créer un premier projet SSAS

“

Lab: Configuration initiale