ENTREPÔTS DE DONNÉES

Guide pratique de modélisation dimensionnelle

2e édition

Ralph Kimball et Margy Ross

Traduction de Claude Raimond

Chapitre 2

Grande distribution

La meilleure façon de comprendre les principes de la modélisation dimensionnelle est d'étudier une série d'exemples concrets. En examinant des cas réels, nous pouvons mieux distinguer les difficultés de conception et leurs solutions que si elles nous sont présentées de manière abstraite. Dans ce livre, nous développerons des exemples provenant d'une série de secteurs d'activité différents, ce qui vous aidera à transcender les détails de votre propre organisation pour mieux atteindre vos objectifs de conception.

Pour apprendre la modélisation dimensionnelle, je vous invite à *lire tous les chapitres* du livre, même si par exemple, vous ne gérez pas une activité de distribution ou une société de télécommunication. Ces chapitres *ne visent pas* à fournir une solution complète pour un secteur d'activité ou un domaine fonctionnel particuliers. Chaque chapitre est une métaphore pour un ensemble caractéristique de problèmes de modélisation dimensionnelle qui se manifestent dans pratiquement n'importe quel secteur d'activité. Les universités, les compagnies d'assurance, les banques et les compagnies aériennes ont à coup sûr besoin des techniques développées dans ce chapitre sur la grande distribution. Par ailleurs, il est parfois rafraîchissant de réfléchir à l'activité de quelqu'un d'autre. Les influences historiques peuvent être mauvaise conseillère quand nous traitons des données de nos propres sociétés. En prenant du recul par rapport à nos propres organisations, pour y revenir avec un (ou deux) principes bien compris, nous maîtriserons mieux les principes de conception lorsque nous plongerons dans les détails complexes de notre activité.

Le chapitre 2 traite des concepts suivants :

- *les quatre étapes de la conception des modèles dimensionnels ;*
- *tables de faits au niveau transaction ;*
- *faits additifs et faits non additifs ;*

- *exemples d'attributs de tables de dimension ;*
- *dimensions causales telles que les promotions ;*
- *dimensions dégénérées, telles que le numéro de ticket de transaction ;*
- *extension d'un modèle dimensionnel existant ;*
- *attributs de dimension en flocons de neige ;*
- *comment éviter le piège « trop de dimensions » ;*
- *clés artificielles ;*
- *analyse du panier d'achat.*

2.1 Processus de modélisation dimensionnelle en quatre étapes

Dans cet ouvrage, nous procéderons à la conception d'une base de données dimensionnelle en passant systématiquement par quatre étapes, toujours dans le même ordre. La signification de ces étapes deviendra de plus en plus évidente à mesure que nous avancerons dans les conceptions successives, mais commençons par les définir.

1. Sélectionner le processus d'entreprise à modéliser. Un processus d'entreprise est une activité normale de votre organisation, généralement assistée par un système source collectant des informations. Écouter les utilisateurs est le moyen le plus efficace pour faire cette sélection. Les mesures de performance qu'ils veulent instamment pouvoir analyser avec l'entrepôt de données sont le résultat de mesures enregistrées au cours du processus d'entreprise. Des exemples sont les achats de matières premières, les commandes, les expéditions, la facturation, la gestion des stocks et la comptabilité générale.

   Rappelons que nous ne nous référons pas à un service ou une fonction d'une organisation quand nous parlons d'un processus d'entreprise. Par exemple, nous construisons un seul modèle dimensionnel pour les commandes client au lieu de construire des modèles distincts pour les services vente et marketing qui veulent l'un et l'autre accéder aux données de vente. En nous focalisant sur les processus au lieu des services, nous pouvons fournir plus économiquement des informations cohérentes sur toute l'organisation. En établissant des modèles liés aux services, nous dupliquerions inévitablement des données différant par les désignations et la terminologie. De multiples flux de données arrivant dans des modèles dimensionnels séparés nous exposent au risque d'incohérence dans les données. La meilleure façon d'assurer la cohérence est de publier les données une seule fois. Un unique traitement de publication réduit l'effort de développement lié à l'activité ETC (extraction/transformation/chargement) ainsi que la gestion courante des données et la consommation d'espace disque.

2. Déclarer le grain du processus. Déclarer le grain veut dire spécifier *exactement* ce que représente une ligne individuelle de table de faits. Le grain représente le niveau de détail des mesures de la table de faits. Il répond à la question « Comment décrivez-vous une ligne unique de la table de faits ? ».

   Voici des exemples de déclarations de grain :
   - une ligne individuelle sur le ticket d'achat d'un client, enregistrée par un dispositif de saisie électronique ;
   - une ligne de prestation sur la note d'honoraires d'un médecin ;
   - un instantané quotidien du niveau de stock de chaque produit dans un entrepôt ;
   - un instantané mensuel de la situation de chaque compte d'une banque.

   Les équipes d'entrepôt de données tentent souvent de sauter cette étape qui leur paraît inutile. N'en faites surtout pas autant ! Il est extrêmement important que tous les membres de l'équipe de conception soient d'accord sur la granularité des tables de faits. Il est pratiquement impossible d'aller au bout de l'étape 3 sans avoir déclaré le grain. Nous vous avertissons qu'une déclaration de grain inadéquate vous hantera tout au long de la réalisation de l'entrepôt de données. Il s'agit d'une étape critique à ne pas prendre à la légère. Cela dit, vous pouvez vous apercevoir aux étapes 3 ou 4 que la déclaration de grain ne convient pas. Ce n'est pas grave, mais vous devez alors retourner à l'étape 2 et déclarer le grain correctement avant de revenir aux étapes 3 et 4.

3. Choisir les dimensions qui s'appliquent à chaque ligne de la table de faits. Les dimensions résultent de la question « Comment les gestionnaires décrivent-ils des données qui résultent du processus concerné ? » Nous voulons enrichir nos tables de faits d'un ensemble fourni de dimensions représentant toutes les descriptions possibles susceptibles de prendre des valeurs particulières dans le contexte de chaque mesure. Si le grain est défini de façon claire, les dimensions peuvent généralement être définies très facilement. Ayant choisi chaque dimension, nous listons tous les attributs distincts, sous forme de texte, qui vont étoffer chaque table de dimension. Voici des exemples courants de dimensions : date, client, type de transaction et état.

4. Identifier les faits numériques qui vont renseigner chaque ligne de la table de faits. Les faits sont déterminés par la réponse à la question, « Que mesurons-nous ? ». Les utilisateurs sont particulièrement intéressés par l'analyse des mesures de performance de l'activité. Tous les faits envisagés doivent être au grain défini à l'étape 2. Les faits qui ont un grain différent doivent figurer dans une autre table de faits. Les faits typiques sont des données numériques, additives, telles que quantité commandée ou coût en euros.

Nous garderons à l'esprit ces quatre étapes tout au long du livre, en développant chacune des études de cas. Pour décider des dimensions et des faits devant figurer dans le modèle dimensionnel, nous serons guidés par la vision que les utilisateurs ont de leur activité. À chacune des quatre étapes de la conception de la base de données dimensionnelle, nous devons évidemment tenir compte à la fois des exigences de nos utilisateurs et de ce qu'il y a effectivement dans nos données sources, comme le montre la figure 2.1. Nous vous recommandons vivement de résister à la tentation de modéliser les données en examinant seulement les données sources. Nous comprenons qu'il est beaucoup moins intimidant de se plonger dans les structures de fichiers et la documentation des programmes que d'interviewer un utilisateur ; mais rien ne remplace l'apport des utilisateurs. Malheureusement, nombreuses sont les organisations qui ont tenté d'emprunter ce chemin de moindre résistance des données sources, avec des résultats mitigés.

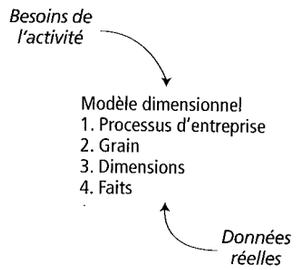

Figure 2.1 **Les deux éléments clés contribuant aux quatre étapes du processus de modélisation dimensionnelle**

2.2 Étude de cas de la distribution

Commençons par une brève description de l'activité d'une entreprise de distribution, pour en rendre plus compréhensibles les tables de dimension et les tables de faits. Nous commençons par cette activité, dont nous avons tous une idée. Supposons que nous travaillions au siège d'une chaîne importante de magasins d'alimentation. Notre entreprise compte cent magasins d'alimentation répartis sur un ensemble de cinq régions. Chacun des magasins est un supermarché moderne classique pourvu de tous les rayons habituels, notamment l'épicerie, les surgelés, les produits laitiers, la viande, les fruits et légumes, la boulangerie, les fleurs et la parapharmacie. Chaque magasin a sur ses étagères environ 60 000 produits individualisés. Les produits individualisés sont appelés des *unités de stock* ou US (en anglais *Stock Keeping Units* ou SKU). Environ 55 000 US viennent de fabricants extérieurs et leurs codes à barres sont imprimés sur l'emballage. Ces codes à barres sont appelés code universel de produits, CUP (en anglais *Universal Product Codes*, UPC). Les CUP sont au même grain que les US. Chaque variante d'emballage d'un produit correspond à un code CUP distinct et a donc une US distincte.

Les 5 000 US restants proviennent de rayons tels que la viande, les fruits et légumes, la boulangerie ou les fleurs et n'ont pas de code CUP. Il nous faut toutefois assigner des codes US à ces produits. Notre chaîne de magasins étant très automatisée, nous collons des étiquettes scannérisables sur la plupart des articles de ces autres rayons. Les codes en question ne sont pas des CUP, mais ce sont néanmoins des US.

La collecte des données peut s'effectuer à plusieurs endroits dans un magasin d'alimentation. L'endroit le plus adéquat et le plus utilisé est la caisse enregistreuse. L'immense majorité des magasins modernes scanne les codes à barres directement au niveau du système de TPV (terminaux point de vente, en anglais *Point of Sale* ou POS). Le PV, sur l'avant du magasin, est l'endroit où le client fait ses achats. L'arrière du magasin, où les fournisseurs livrent les marchandises, est un autre endroit permettant la collecte de données.

Les préoccupations du management d'un magasin d'alimentation sont la logistique des commandes, le stockage et la vente des produits ainsi que l'obtention d'un profit maximal. Les éléments contribuant au profit sont un prix le plus élevé possible pour chaque produit, l'abaissement du coût d'acquisition et des frais de gestion et aussi le fait d'attirer le plus grand nombre possible de clients dans le contexte d'une forte concurrence axée sur le prix. Les décisions les plus importantes portent sur les prix et les promotions. Aussi bien les dirigeants du magasin que ceux du siège passent un temps considérable à ajuster les prix et à réaliser des promotions. Les promotions d'un magasin d'alimentation comportent des réductions de prix temporaires, des annonces et des encarts dans les journaux, des présentations dans le magasin comprenant les présentations sur gondole et les présentations en tête de gondole, ainsi que les coupons. Le moyen le plus efficace pour provoquer une brusque augmentation du volume (la quantité vendue) consiste à réduire le prix de façon spectaculaire. Une réduction de 50 cents du prix des rouleaux de papier absorbant, surtout si elle est couplée avec une annonce et une présentation, peut faire multiplier les ventes de ces rouleaux de papier par un facteur dix. Malheureusement, une réduction de prix de cette importance ne peut continuer, les rouleaux étant probablement vendus à perte. Il résulte de ces diverses considérations que l'analyse des activités d'un magasin d'alimentation doit prendre en compte toutes les formes de promotion.

Connaissant les grandes lignes du cas étudié, nous pouvons commencer à concevoir le modèle dimensionnel

Étape 1. Sélection du processus d'entreprise à modéliser

La première étape de la conception consiste à décider quels processus d'entreprise doivent être modélisés, en confrontant notre compréhension des exigences de l'activité à la réalité des données disponibles.

Le premier modèle dimensionnel construit doit être celui dont l'incidence est la plus forte. Il doit répondre aux questions les plus pressantes des utilisateurs et ses données doivent être parmi les plus faciles à extraire.

Dans notre étude de cas sur la distribution, le management veut mieux comprendre les achats des clients saisis par le système de terminaux point de vente. C'est pourquoi nous allons modéliser en premier les ventes au niveau des TPV (terminaux point de vente). Ces données serviront à déterminer quels produits se vendent, dans quels magasins, quels jours, dans quelles conditions de promotion.

Étape 2. Déclaration du grain

Une fois le processus identifié, l'équipe de l'entrepôt de données doit prendre une décision importante concernant la granularité. Quel niveau de détail doit-il être rendu disponible dans le modèle dimensionnel ? Ceci nous conduit à formuler une recommandation importante.

Vous avez intérêt à développer des modèles dimensionnels basés sur les données les plus atomiques saisies lors d'un processus de l'activité. Les données atomiques sont les informations les plus détaillées susceptibles d'être collectées ; par définition, il n'est pas possible de les subdiviser davantage.

Retenir les données au niveau le plus granulaire est avantageux à bien des égards. Les données atomiques sont hautement dimensionnelles. Plus le niveau de détail des mesures est fin, plus nous savons de choses avec certitude. Toutes ces choses que nous savons se traduisent par des dimensions. Les données atomiques conviennent parfaitement à l'approche dimensionnelle.

Les données atomiques se prêtent à des analyses d'une souplesse maximale, car on peut les soumettre à des contraintes et les cumulées de toutes les manières possibles. Les données détaillées sont toujours prêtes à subir les assauts des demandes *ad hoc* des utilisateurs.

Bien sûr, vous pouvez toujours déclarer un grain plus élevé pour un processus qui représente un niveau d'agrégation des données atomiques. Cependant, dès que vous choisissez un grain plus élevé, vous vous limitez à des dimensions en nombre plus faible et/ou moins détaillées. Un modèle moins granulaire est immédiatement vulnérable à des demandes inattendues d'utilisateurs qui veulent faire un forage plus profond. S'ils n'ont pas accès aux données atomiques, les utilisateurs se heurteront tôt ou tard à un mur. Comme nous le verrons au chapitre 16, les données cumulées jouent un rôle important en tant qu'outils d'optimisation des performances, mais elles n'éliminent pas la nécessité de permettre aux utilisateurs d'accéder au plus bas niveau de détail. Malheureusement, certains décideurs de la profession ont semé la confusion dans ce domaine. Ils affirment que les modèles dimensionnels ne conviennent que pour les données récapitulatives et critiquent ensuite

l'approche de la modélisation dimensionnelle en lui reprochant d'obliger à prévoir à l'avance les questions des utilisateurs. Ce malentendu disparaît dès lors que des données atomiques, détaillées, sont disponibles dans un modèle dimensionnel.

Dans notre étude de cas, la donnée la plus granulaire est une ligne individuelle d'une transaction saisie sur un TPV. En vue d'une dimensionnalité et d'une souplesse maximales, c'est ce grain que nous retenons. Notez que cette déclaration de granularité représente un changement par rapport à la première édition du livre. Nous y avions prévu d'utiliser les données des TPV, mais au lieu de représenter le détail des lignes de transaction dans le modèle dimensionnel, nous avions choisi de fournir des données récapitulées par produit et par promotion dans un magasin et une journée. À cette époque, ces totaux représentaient l'état de l'art pour les bases de données des fournisseurs d'information de cette nature. Il était déraisonnable d'attendre du matériel et du logiciel de l'époque qu'il puisse traiter efficacement les volumes de données associés aux lignes individuelles des transactions TPV.

L'accès aux transactions des terminaux point de vente nous donne une vue extrêmement détaillée des ventes d'un magasin. Les utilisateurs ne sont probablement pas intéressés par l'analyse d'éléments individuels au niveau d'une transaction TPV spécifique, mais nous ne pouvons pas prédire toutes les façons dont ils voudront puiser dans les données. Par exemple, ils peuvent vouloir comprendre la différence entre les ventes du lundi par comparaison à celles du dimanche. Ou ils peuvent vouloir déterminer si cela vaut la peine de stocker autant de références de certaines marques, par exemple dans le cas de marques de céréales. Ou ils peuvent vouloir comprendre combien d'utilisateurs ont tiré parti de la promotion avec 50 % de rabais sur le shampoing. Ou ils peuvent vouloir déterminer la réduction des ventes consécutive à une forte promotion d'un produit concurrent d'une marque de soda. Aucune de ces requêtes ne fait appel à des données d'une transaction spécifique, mais elles posent des questions générales qui exigent des coupes dans les données selon des critères très précis. Aucune de ces questions n'aurait pu être traitée si nous avions choisi de ne fournir qu'un accès à des données cumulées.

Un entrepôt de données exige presque toujours des données exprimées au niveau du grain le plus fin possible pour chaque dimension, non parce que les requêtes veulent voir les lignes individuelles de bas niveau, mais parce que les requêtes ont besoin de faire des coupes dans les détails de diverses manières très précises.

Étape 3. Choix des dimensions

Une fois le grain de la table de faits choisi, les dimensions temps, produit et magasin s'imposent instantanément. Nous supposons que la date du jour est la valeur de date qui nous est fournie par le système TPV. Nous verrons plus tard ce qu'il faut faire si nous obtenons aussi une heure en même temps que la date. Dans le cadre de nos dimensions élémentaires, nous pouvons nous demander quelles

autres dimensions attribuer aux données, telles que la promotion sous laquelle le produit a été vendu, ou le fournisseur qui a livré le produit au magasin, ou le nom du responsable du magasin ce jour-là. Nous reprenons ces points sous la forme d'un autre principe de conception :

> **L'énoncé précis du grain détermine les dimensions principales de la table de faits. Il est en général possible d'ajouter ensuite des dimensions supplémentaires, à condition que ces dimensions supplémentaires ne puissent prendre qu'une seule valeur pour chaque combinaison des dimensions élémentaires. Si l'on veut une dimension supplémentaire qui détruit le grain choisi initialement en entraînant la génération de lignes de faits supplémentaires, il faut alors revoir la définition du grain pour prendre en compte la dimension supplémentaire.**

Dans notre étude de cas, nous avons retenu les quatre dimensions : date, produit, magasin et promotion. En outre, nous incluons le numéro du ticket de transaction TPV en tant que dimension spéciale. Nous en dirons davantage sur ce point plus loin dans ce chapitre.

Notre schéma préliminaire commence à prendre forme, tel qu'il est représenté à la figure 2.2. Avant d'aborder le remplissage des tables de dimension par leurs attributs descriptifs, qui pourrait vous faire perdre la vue d'ensemble, examinons la quatrième et dernière étape de la modélisation.

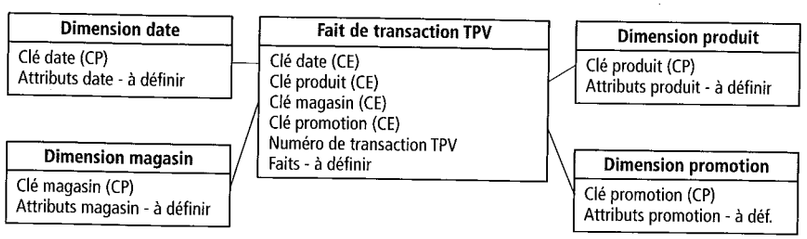

Figure 2.2 **Schéma préliminaire des ventes grande distribution**

Étape 4. Identification des faits

La quatrième et dernière étape de la conception est la détermination des faits qui apparaîtront dans la table de faits. Ici encore, notre choix est fonction de la déclaration du grain : en l'occurrence, la ligne individuelle d'article d'une transaction TPV. Et comme pour l'étape précédente, la détermination des faits peut conduire à revoir la déclaration du grain ou le choix des dimensions.

Les faits collectés par le système TPV incluent la quantité vendue (par exemple le nombre de boîtes de potage au poulet) pour chaque prix unitaire et le montant de la vente en euros. Ce montant est égal à la quantité multipliée par le prix unitaire. Les TPV les plus sophistiqués fournissent en outre un coût standard pour le produit tel qu'il est livré au magasin par le fournisseur. Supposant que ce coût soit

disponible sans requérir un effort complexe de détermination à partir de l'activité, nous l'incluons dans la table de faits. Notre table de faits prend la forme de la figure 2.3.

Trois de ces faits, la quantité vendue, le montant de la vente en euros et le coût en euros, sont parfaitement additifs sur toutes les dimensions. Nous pouvons à volonté faire des coupes en tranches et en dés dans la table de faits et toute somme de l'un de ces trois faits sera valide et exacte.

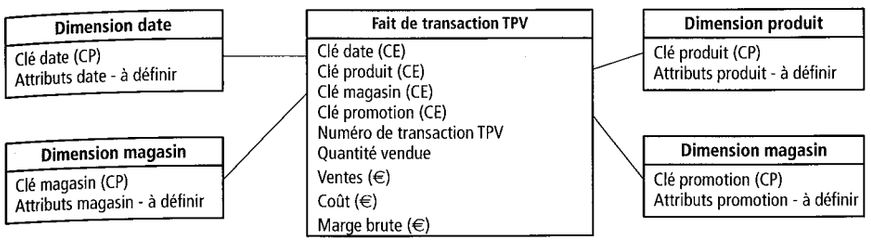

Figure 2.3 **Faits mesurés du schéma vente au détail**

Nous pouvons calculer le bénéfice brut en soustrayant le coût du montant de la vente. Bien qu'il soit calculé, ce bénéfice brut est lui aussi parfaitement additif sur toutes les dimensions — nous pouvons le calculer pour toute combinaison de produits vendus dans tout ensemble de magasins pour tout intervalle de jours. Les concepteurs de modèles dimensionnels se demandent parfois si un fait calculé doit être stocké dans la base de données. Nous recommandons généralement de le stocker physiquement. Dans notre étude de cas, le calcul du bénéfice brut est simple, mais son stockage élimine les possibilités d'erreur par les utilisateurs. Le coût d'une erreur de calcul sur le bénéfice brut est infiniment plus élevé que celui d'une augmentation minime de l'espace de stockage. Le stockage assure également que tous les utilisateurs et leurs applications de reporting se réfèrent de manière cohérente au bénéfice brut. Comme il peut être calculé à partir de données adjacentes au sein d'une table de faits, certains affirment que nous devons faire le calcul dans une vue qui se confond avec la table. Cette approche est raisonnable si *tous* les utilisateurs accèdent aux données par le biais de cette vue et si *aucun* utilisateur disposant d'outils de requête *ad hoc* ne peut parvenir jusqu'à la table sans passer par la vue. De même, certaines organisations veulent faire le calcul avec l'outil de requête. Ici encore, cette approche est valable si *tous* les utilisateurs accèdent aux données par un outil commun (ce qui, d'après notre expérience, est rarement le cas).

La marge brute peut être calculée en divisant le bénéfice brut par le chiffre d'affaires. La marge brute est un fait non additif parce qu'on ne peut l'additionner sur aucune des dimensions. Nous pouvons calculer la marge brute pour tout ensemble de produits, de magasins ou de jours en pensant à additionner les chif-

fres d'affaires et les coûts avant de faire la division. Ce qui peut être énoncé dans un principe de conception :

> **Les pourcentages et les ratios, tels que la marge brute, sont non additifs. Le numérateur et le dénominateur doivent être stockés dans la table de faits. Le ratio peut être calculé dans un outil d'accès aux données pour toute tranche de la table de faits en pensant à faire le ratio des sommes, non la somme des ratios.**

Le prix unitaire est lui aussi un fait non additif. L'addition de prix unitaires sur n'importe quelle dimension donne un nombre dépourvu de sens. Pour analyser le prix moyen d'un produit dans une série de magasins ou sur une période, nous devons additionner les ventes en euros et les quantités avant de diviser le total des ventes par le total des quantités vendues. Tout logiciel de reporting ou outil de requête du marché des entrepôts de données devrait exécuter cette fonction automatiquement, mais malheureusement certains ne le font pas de très bonne grâce.

À ce stade préliminaire de la conception, il est souvent utile d'estimer le nombre de lignes de la plus grande table, la table de faits. Dans notre étude de cas, il peut suffire de parler avec l'un des gourous du système source pour apprendre combien de lignes de transactions TPV sont générées périodiquement. Le trafic de la distribution fluctue de manière significative d'un jour à l'autre et nous avons besoin de connaître l'activité des transactions sur une période raisonnable. Une autre manière d'estimer le nombre de lignes de la table de faits pourrait consister à diviser le chiffre d'affaires brut annuel de la chaîne de distribution par le prix de vente unitaire moyen. En supposant que le chiffre d'affaires soit de 4 milliards d'euros par an et que le prix moyen d'un article sur le ticket d'un client soit de 2 euros, nous calculons qu'il y a environ 2 milliards de lignes de transaction par an. Ce genre de calcul permet parfois d'approcher la réalité de très près. En tant que concepteurs, nous devons faire de nombreux recoupements pour déterminer si nos calculs sont raisonnables.

2.3 Attributs de table de dimension

Ayant parcouru notre processus en quatre étapes, revenons aux tables de dimension et fixons notre attention sur leur remplissage en attributs bien fournis.

Dimension date

Nous commençons par la dimension date. Cette dimension est présente dans pratiquement tous les marchés d'infos, car les marchés d'infos sont presque toujours des séries temporelles. En fait, la date est habituellement la première dimension dans l'ordre de tri de la base de données, de sorte que les chargements successifs de données relatives à un intervalle de temps s'inscrivent en territoire vierge sur le disque.

Dans la première édition du livre (Wiley, 1996), cette dimension s'appelait *dimension temps*. Dans la présente édition, nous avons remplacé cette dénomination assez vague par *dimension date* pour nous référer à des tables de dimension au grain de la journée. Ceci nous permet de distinguer la date et les dimensions heure du jour, évoquées plus loin dans ce chapitre.

Contrairement à la plupart de nos autres dimensions, nous pouvons construire la table dimension date à l'avance. Nous pouvons prévoir cinq ou dix ans de lignes représentant chacun des jours dans la table, de manière à couvrir le passé enregistré, ainsi que plusieurs années dans l'avenir. Les jours de dix années ne représentent que 3560 lignes, soit une table de dimension relativement petite. Pour une table de dimension date du jour dans un environnement de distribution, nous recommandons la liste partielle de colonnes de la figure 2.4.

Le contenu de chaque colonne de la table dimension date dépend du jour particulier que la ligne représente. La colonne jour de la semaine contient le nom du jour, par exemple lundi. Cette colonne peut servir pour créer des états comparant l'activité du lundi avec celle d'un autre jour. Le quantième (numéro du jour dans le mois) va de 1 à 28, 29, 30 ou 31 selon le mois. Cette colonne est utile pour comparer des jours à la même distance du début du mois. De la même façon, nous pourrions avoir un numéro de mois dans l'année (1,..., 12). Le numéro de jour de l'époque est un numéro de jour dans un calendrier comme le calendrier julien (c'est-à-dire un numéro d'une série de jours commençant au début d'une époque donnée). Nous pourrions aussi inclure des colonnes de numéro absolu de semaine et de mois. Tous ces nombres entiers servent à des calculs arithmétiques simples entre des jours situés de part et d'autre de limites d'années ou de mois. Pour des états, il nous faut un nom de mois tel que janvier. En outre, une colonne mois de l'année (AAAA-MM) est utile comme intitulé de colonne d'état. Nous voudrons probablement aussi un numéro de trimestre (T1,...,T4), ainsi qu'un trimestre d'année, comme 2001-T4. Nous aurons aussi des colonnes similaires pour les périodes comptables si elles diffèrent des périodes calendaires.

L'indicateur de jour férié prend les valeurs *férié* ou *non férié*. N'oubliez pas que les attributs de la table de dimension servent comme intitulés dans les états. Renseigner simplement l'indicateur de jour férié d'un Oui ou d'un Non serait bien moins parlant. Imaginez un état dans lequel nous comparons les ventes d'un produit un jour férié à celles d'un jour non férié. Il vaut évidemment beaucoup mieux que les colonnes concernées aient des intitulés significatifs comme férié/non férié plutôt qu'un mystérieux O/N. Au lieu de décoder des drapeaux abscons dans l'application de reporting, nous préférons que la signification du drapeau soit dans la base de données, pour qu'une valeur cohérente apparaisse à tous les utilisateurs, indépendamment de leur environnement de reporting.

Un même argument s'applique à l'indicateur de jour de la semaine, qui a la valeur jour de semaine ou week-end. Les samedis et les dimanches se voient affecter évidemment la valeur week-end. Bien sûr, nous pouvons établir des contraintes en même temps sur plusieurs attributs de la table date et pouvons ainsi comparer des jours fériés tombant le week-end et des jours fériés tombant en semaine, par exemple.

La colonne saison de vente est le nom d'une période particulière, par exemple, la période de Noël, Pâques, le Jour de la Saint-Valentin, ou pas de saison de vente. La colonne événement majeur est similaire à la colonne saison de vente et peut servir à marquer des jours spéciaux comme la coupe du monde de football, une élection, etc. Les événements promotionnels ordinaires ne sont généralement pas placés dans la table dimension date mais sont décrits de manière plus complète au moyen de la dimension promotions, surtout parce que les événements promotionnels ne dépendent pas uniquement de la date mais habituellement d'une combinaison date, produit et magasin.

À ce stade, certains concepteurs mettent en cause l'utilité d'une table de dimension date explicite. Leur raisonnement est que si la clé date dans la table de faits est un champ de type date, toute requête SQL peut directement établir une contrainte sur la clé date de la table de faits et utiliser la sémantique naturelle du SQL relative aux dates pour faire un filtrage sur le mois ou l'année, ce qui évite une jointure supposée coûteuse. Ce raisonnement ne tient pas, pour plusieurs raisons. Si notre base de données ne peut pas traiter efficacement une jointure sur la table de dimension date, nous sommes vraiment très mal partis. La plupart des optimiseurs de bases de données sont tout à fait capables de traiter efficacement des requêtes dimensionnelles. Par ailleurs, toujours sur le front des performances, la plupart des bases de données n'indexent pas les calculs de date en SQL, de sorte que des contraintes sur un champ SQL calculé ne pourraient pas tirer parti d'un index.

Sous l'angle de la commodité, l'utilisateur type n'est pas très au courant de la sémantique du SQL concernant les dates et serait incapable d'exploiter directement les fonctions associées à un type de donnée date. Ces fonctions SQL ne supportent pas le filtrage par des attributs tels que des jours de semaine par rapport aux jours de week-end, les jours fériés, les périodes comptables, les saisons de vente, ou les événements majeurs. Dès lors que les analystes de l'activité ont besoin de faire des coupes selon ces attributs non standard, une table de dimension date explicite est indispensable. Conclusion, la logique du calendrier est du ressort d'une table de dimension, elle n'a rien à faire dans le code des applications. Et pour finir, nous allons suggérer que la clé date soit de toute façon un entier et non un champ de type date. Une clé date SQL prend généralement 8 octets, ce qui gaspillerait 4 octets pour chaque clé date de chaque ligne de la table de faits. Nous reviendrons sur ce point plus loin dans ce chapitre.

La figure 2.5 représente un fragment d'une table de dimension date.

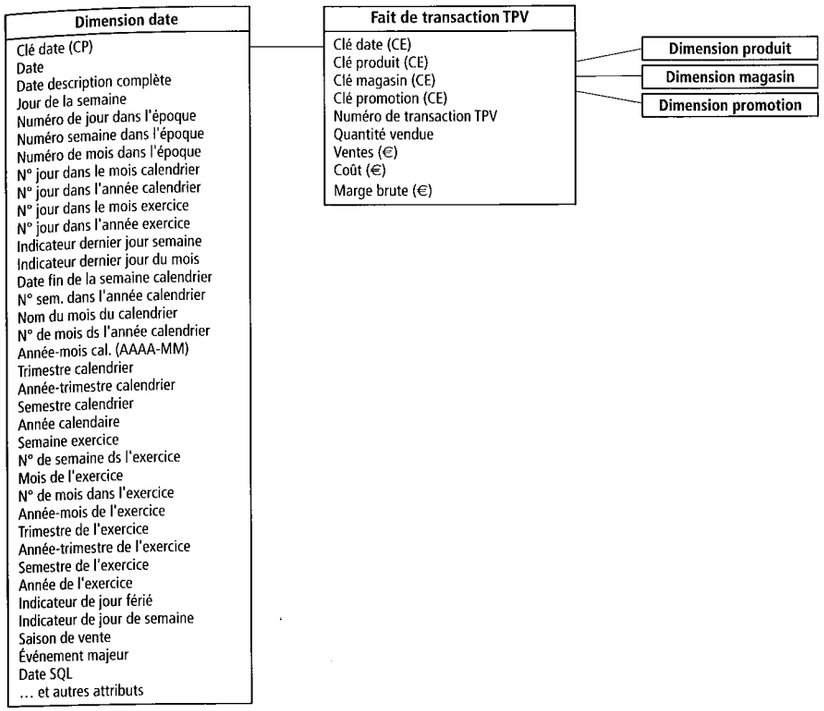

Figure 2.4 **Schéma d'une dimension date pour la distribution**

Clé date	Date	Date description complète	Jour de la semaine	Mois calendaire	Année calendaire	Année-mois de l'exercice	Indicateur de jour férié	Indicateur j. de sem.
1	01/01/2002	1er janvier 2002	mardi	janvier	2002	E2002-01	férié	j. de sem.
2	02/01/2002	2 janvier 2002	mercredi	janvier	2002	E2002-01	non férié	j. de sem.
3	03/01/2002	3 janvier 2002	jeudi	janvier	2002	E2002-01	non férié	j. de sem.
4	04/01/2002	4 janvier 2002	vendredi	janvier	2002	E2002-01	non férié	j. de sem.
5	05/01/2002	5 janvier 2002	samedi	janvier	2002	E2002-01	non férié	week-end
6	06/01/2002	6 janvier 2002	dimanche	janvier	2002	E2002-01	non férié	week-end
7	07/01/2002	7 janvier 2002	lundi	janvier	2002	E2002-01	non férié	j. de sem.
8	08/01/2002	8 janvier 2002	mardi	janvier	2002	E2002-01	non férié	j. de sem.

Figure 2.5 **Détails d'une table de dimension date**

Les entrepôts de données ont toujours une table de dimension date explicite. Il existe de nombreux attributs de date non supportés par la fonction date de SQL, notamment les périodes comptables, les saisons de vente, les jours fériés et les jours de week-end. Au lieu d'essayer de déterminer ces éléments de calendrier non standard dans une requête, nous devons les intégrer dans une table de dimension date.

Si nous voulions accéder à l'heure de la transaction en vue d'une analyse sur une partie de la journée (par exemple l'activité dans la soirée durant le rush consécutif à la sortie du travail ou à un changement d'équipe), nous le ferions par le biais d'une table de dimension séparée heure du jour jointe à la table de faits. La date et l'heure sont presque complètement indépendantes. Si nous combinions les deux dimensions, la dimension date augmenterait fortement de volume ; notre table de dimension date bien rangée avec 3 650 lignes pour une durée de dix ans aurait d'un coup 5 256 000 lignes si nous essayions de traiter le temps au niveau des minutes dans la même table (ou par l'intermédiaire d'une table de dimension déportée). Il est évidemment préférable de créer une table de dimension date de 3 650 lignes et une table de dimension heure du jour par minute de 1 440 lignes.

Nous verrons au chapitre 5 le traitement de dates multiples dans un seul schéma. Et nous examinerons les dates internationales et les questions de décalage horaire aux chapitres 11 et 14.

Dimension produit

La dimension produit décrit chacun des articles (ou US, unités de stock) offerts dans l'un des magasins d'alimentation. Alors qu'un magasin type de notre chaîne peut avoir 60 000 articles en stock, si nous tenons compte des différences de marchandising d'un magasin à un autre et des produits qui ne sont plus disponibles, notre dimension produit atteint au moins 150 000 lignes et peut-être jusqu'à un million de lignes. La dimension produit est presque toujours obtenue à partir du fichier maître produit de l'application opérationnelle. La plupart des distributeurs gèrent leur fichier maître au siège et téléchargent un sous-ensemble à des intervalles rapprochés vers le système TPV de chaque magasin. C'est le siège qui définit l'enregistrement du fichier maître (et l'US) correspondant à chaque nouveau produit créé par les fournisseurs de produits emballés. Le siège définit aussi les règles selon lesquelles les US sont attribuées à des articles tels que ceux de la boulangerie, la viande et les fruits et légumes. Nous extrayons le fichier maître produit pour le placer dans la table de dimension produit à chaque modification du fichier maître.

Une fonction importante du fichier maître produit est de contenir les nombreux attributs descriptifs de chaque US. En général, les US se regroupent en marques. Les marques en catégories et les catégories en rayons. Chacune de ces relations est du type un-à-plusieurs. La hiérarchie des marchandises et d'autres attributs sont présentés à la figure 2.6 pour un sous-ensemble de table de dimension produit.

Tous les niveaux de la hiérarchie des marchandises sont bien définis pour chacun des produits. Certains attributs, tels que la description de l'US, sont différents pour chaque US, de sorte qu'il y a au moins 150 000 valeurs différentes dans la colonne description du produit. À l'autre extrême, il n'y a peut-être que 50 valeurs différentes pour l'attribut rayon. Ainsi, en moyenne, une même valeur de l'attribut rayon se répète 3 000 fois. C'est sans importance ! Nous n'avons pas besoin de ranger dans une table séparée ces valeurs qui se répètent pour économiser de

Clé produit	Description du produit	Description marque	Description de la catégorie	Description du rayon	Contenu en mat. grasses
1	Pain frais bien cuit au levain	Bien cuit	Pain	Boulangerie	Réduit
2	Au froment complet en tranches	Soufflé	Pain	Boulangerie	Normal
3	Au froment complet léger en tranches	Soufflé	Pain	Boulangerie	Réduit
4	Petits pains à la canelle sans matières grasses	Léger	Pains sucrés	Boulangerie	Sans mat. gr.
5	5 litres glace vanille régime	Paquet froid	Desserts glacés	Surgelés	Sans mat. gr.
6	Demi-litre glace à la noix de pécan	Froidure	Desserts glacés	Surgelés	Réduit
7	1 litre glace des amis du chocolat	Superchoco	Desserts glacés	Surgelés	Normal
8	Demi-litre glace fraise	Le glacier	Desserts glacés	Surgelés	Normal
9	Sandwiches à la crème glacée	Le glacier	Desserts glacés	Surgelés	Normal

Figure 2.6 **Détails d'une table de dimension produit**

l'espace. Rappelez-vous que les exigences en espace des tables de dimension sont très faibles par comparaison à celles des tables de faits.

De nombreux attributs de la table de dimension produit ne font pas partie de la hiérarchie de marchandise. L'attribut *type d'emballage*, par exemple, pourrait avoir des valeurs telles que Bouteille, Sac, Boîte etc. N'importe quel article de n'importe quel rayon peut avoir l'une de ces valeurs. Il est parfaitement raisonnable d'établir une contrainte sur un tel attribut en même temps que sur un attribut de la hiérarchie de marchandise. Par exemple, nous pourrions regarder toutes les US de la catégorie céréales vendues en sac. Autrement dit, nous pouvons naviguer parmi les attributs de dimension, qu'ils appartiennent ou non à la hiérarchie de marchandise et nous pouvons faire des forages vers le haut ou le bas au moyen des divers attributs sans nous préoccuper de leur appartenance à la hiérarchie de marchandise. Nous pouvons même avoir plus d'une hiérarchie explicite dans notre table de dimension produit.

La figure 2.7 représente une partie d'une dimension produit que nous suggérons d'utiliser pour un marché d'infos de magasin d'alimentation.

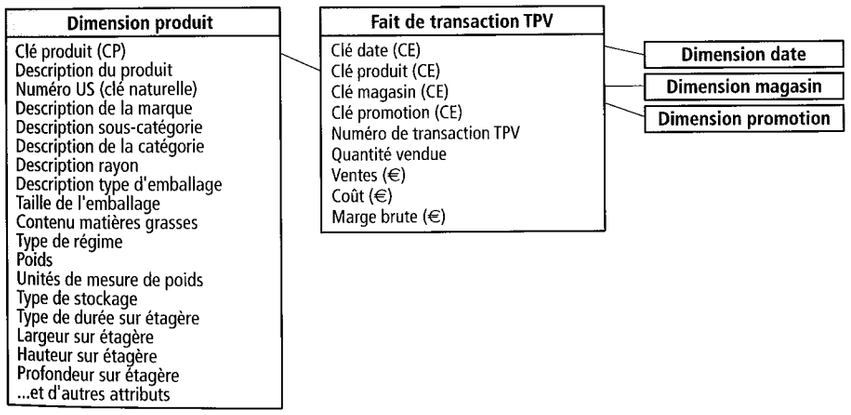

Figure 2.7 **Schéma d'une dimension produit pour la distribution**

Une table de dimension produit de taille raisonnable pourrait avoir 50 attributs ou plus. Chaque attribut est une source abondante de contraintes et d'intitulés pour les lignes d'état. Sous cet angle, nous voyons qu'un forage vers le bas consiste à demander à un intitulé de ligne de fournir plus d'informations. Supposons que nous ayons sous les yeux un simple état résumant les montants en euros et les quantités vendues par rayon.

Description du rayon	Montant ventes en euro	Quantité vendue
Boulangerie	12 331	5 088
Surgelés	31 776	15 565

Pour faire un forage vers le bas, nous pouvons faire glisser pratiquement n'importe quel attribut, comme par exemple la marque, depuis la dimension produit jusqu'à l'état, à côté du rayon et nous avons aussitôt un forage vers le bas à ce niveau de détail. Un forage typique vers le bas à l'intérieur de la hiérarchie de marchandise aurait l'aspect suivant :

Description du rayon	Description de la marque	Montant ventes en euro	Quantité vendue
Boulangerie	Bien cuit	3 009	1 138
Boulangerie	Vaporeux	3 024	1 476
Boulangerie	Léger	6 298	2 474
Surgelés	Dufroid	5 321	2 640
Surgelés	Froidure	9 476	5 234
Surgelés	Glacière	7 328	3 092
Surgelés	Banquise	2 184	1 437
Surgelés	Antartique	6 467	3 172

Ou nous pourrions faire un forage vers le bas pour le contenu en matières grasses, même si cet attribut n'est pas dans la hiérarchie de marchandise :

Description du rayon	Matières grasses	Montant ventes en euro	Quantité vendue
Boulangerie	Sans	6 298	2 474
Boulangerie	Réduit	5 027	2 086
Boulangerie	Normal	1 006	1 528
Surgelés	Sans	5 321	2 640

Description du rayon	Matières grasses	Montant ventes en euro	Quantité vendue
Surgelés	Réduit	9 476 euros	5 234
Surgelés	Normal	9 979 euros	7 691

Nous avons insisté sur ces exemples de forage vers le bas pour bien faire comprendre un point que nous formulons sous forme d'un principe de conception :

> **Le forage vers le bas dans un marché d'infos n'est rien d'autre que l'adjonction d'intitulés de lignes supplémentaires à partir des tables de dimension. Le forage vers le haut consiste à supprimer des intitulés de lignes. Nous pouvons forer vers le haut ou le bas sur des attributs appartenant à plusieurs hiérarchies explicites différentes, combinés éventuellement à des attributs n'appartenant à aucune hiérarchie.**

La dimension produit est l'une des deux ou trois dimensions primaires de pratiquement tout marché d'infos. Il est indispensable de remplir soigneusement cette dimension du plus grand nombre possible d'attributs. Un ensemble complet et riche d'attributs de dimension se traduit par des possibilités d'analyse complètes et riches pour les utilisateurs. Nous continuerons d'examiner la dimension produit au chapitre 4, où nous verrons le traitement des modifications dans les attributs de produits.

Dimension magasin

La dimension magasin décrit chaque magasin de notre chaîne de distribution de produits d'alimentation. Alors que le fichier maître produit est pratiquement disponible à coup sûr dans toute entreprise importante de distribution alimentaire, il se peut qu'il n'existe pas de fichier maître complet pour les magasins. Alors que le fichier maître produit doit être téléchargé vers chaque magasin à chaque fois qu'il y a un nouveau produit ou une modification sur un produit, les systèmes TPV n'ont pas besoin d'un fichier maître des magasins. Les équipes informatiques doivent souvent rassembler les divers éléments de la dimension magasin à partir de multiples sources opérationnelles du siège.

La dimension magasin est la principale dimension géographique de notre étude de cas. On peut en effet se représenter chaque magasin comme un emplacement géographique. Tout magasin peut être inséré dans un attribut géographique tel qu'un code postal, une province ou un état. En général, les magasins sont également regroupés selon des districts et des régions définis par l'entreprise. Ces deux hiérarchies différentes sont faciles à représenter dans la dimension géographique parce que l'une et l'autre sont bien définies pour toute ligne affectée à un magasin.

Il est courant de représenter de multiples hiérarchies dans une table de dimension ; les noms d'attribut et les valeurs correspondantes doivent cependant être les mêmes pour les différentes hiérarchies.

La figure 2.8 montre une table de dimension magasin recommandée pour le secteur alimentation.

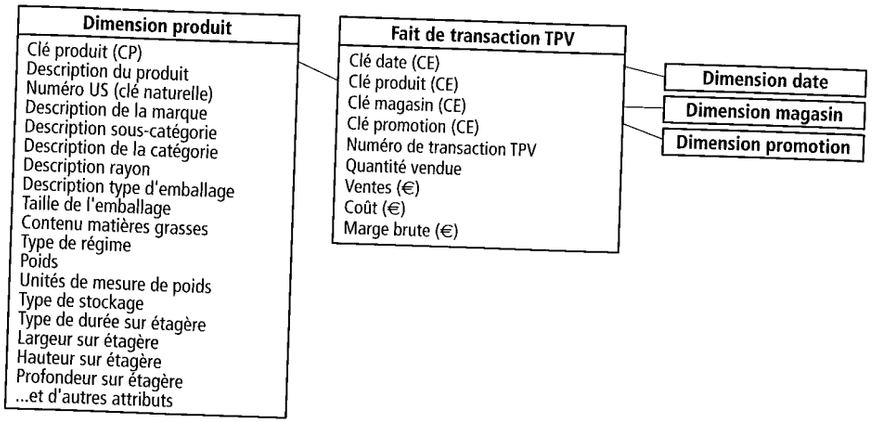

Figure 2.8 **Dimension magasin dans un schéma pour la distribution alimentaire**

Le type d'agencement de surface, le type de développement photo et le type de services commerciaux sont autant de descripteurs textuels propres au magasin particulier. Ces descripteurs ne doivent pas être des codes de un caractère mais des descripteurs standard lisibles et compréhensibles lorsqu'ils apparaissent sous forme d'une liste déroulante ou lorsqu'ils sont utilisés comme intitulé de ligne dans un état.

La colonne décrivant la surface consacrée à la vente est numérique et théoriquement additive sur les magasins. On peut être tenté de la placer dans la table de faits. Il s'agit toutefois d'un attribut constant d'un magasin, utilisé plus souvent comme contrainte ou comme intitulé de ligne dans un état que comme élément additif dans une récapitulation. C'est pourquoi nous le plaçons sans hésiter dans la table de dimension magasin.

La date de première ouverture et la date de dernière rénovation sont des champs sous forme de date auxquels on peut accéder directement à l'aide de SQL ou qui peuvent être utilisés comme code de jointure à des copies de la table de dimension date. Ces copies de la table de dimension date sont déclarées en SQL par le terme `VIEW` et sont différentes d'un point de vue sémantique de la dimension principale date. Une déclaration `VIEW` classique se présente comme suit:


```
CREATE VIEW PREMIERE_DATE_OUVERTURE (PREMIER_JOUR_OUVERT,
                                     PREMIER_MOIS_OUVERT...)
AS SELECT NUMERO_JOUR, MOIS,...
FROM DATE
```

Le système se comporte maintenant comme s'il y avait une autre copie physique de la table de dimension date appelée `PREMIER_JOUR_OUVERT`. Les contraintes sur cette nouvelle table n'ont rien à voir avec les contraintes sur la table primaire de dimension date. La vue premier jour ouvert est un élément déporté légitime de la dimension magasin. Remarquez que nous avons soigneusement renommé toutes les colonnes de la vue pour qu'elles ne puissent pas être confondues avec les colonnes de la dimension date primaire. Nous reviendrons sur les éléments déportés au chapitre 6.

Dimension promotion

La dimension promotion est peut-être la dimension la plus intéressante de tout le schéma. Elle décrit chacune des modalités de promotion suivant lesquelles un produit peut être vendu dans notre chaîne de magasins. Ces modalités de promotion peuvent être des réductions de prix temporaires, des présentations en tête de gondole, des annonces dans les journaux et des coupons. Cette dimension est parfois nommée dimension *causale* parce qu'elle décrit des facteurs supposés provoquer des modifications dans les ventes des produits.

Les responsables tant du siège que du magasin sont très intéressés de savoir si une promotion est ou non efficace. Les promotions sont jugées selon un ou plusieurs des critères ci-après:

- Y a-t-il eu une augmentation des ventes au cours de la période de promotion? C'est ce que l'on appelle le *gain*. Le gain ne peut être mesuré que si le magasin peut déterminer ce qu'auraient été les ventes normales (en anglais *baseline sales*) en l'absence de promotion. Les valeurs des ventes normales peuvent être estimées à partir de l'historique des ventes antérieures et dans certains cas, à l'aide de modèles mathématiques sophistiqués.
- Les produits en promotion ont-ils connu une baisse des ventes avant ou après la promotion, annulant ainsi les ventes gagnées au cours de la promotion? (décalage temporel). En d'autres termes, y a-t-il eu transfert de vente de produits à prix normaux vers des produits à prix temporairement réduits?
- Les produits en promotion ont-ils connu une augmentation de leurs ventes mais d'autres produits voisins sur l'étagère ont-ils subi une diminution correspondante de leurs ventes ? (cannibalisation).
- Tous les produits de la catégorie en promotion ont-ils connu une augmentation nette globale des ventes, en tenant compte de la période de promotion et des périodes qui l'ont précédée et suivie ? (croissance du marché).

- La promotion a-t-elle été profitable? Le profit d'une promotion est habituellement supposé être égal à l'augmentation du profit pour la catégorie concernée par rapport à celui des ventes normales en tenant compte du décalage temporel et de la cannibalisation, ainsi que des coûts entraînés par la promotion, tels que les réductions de prix temporaires, les annonces, les présentations et les coupons.

Les modalités causales d'une vente ne sont pas toujours directement saisies par le système transactionnel TPV. Ce système enregistre les baisses de prix et les rabais. Le traitement des coupons est normalement saisi avec la transaction, parce qu'on sait si le client présente ou non le coupon au moment de la vente. Les publicités dans la presse et les présentations spéciales dans le magasin doivent être puisées à d'autres sources.

Les différentes modalités causales possibles sont fortement corrélées. Une réduction de prix temporaire est couramment associée à une annonce et parfois à une présentation en tête de gondole. Les coupons sont souvent associés à des annonces. Pour ces raisons il est sensé de créer un enregistrement dans la table de dimension promotion pour chaque combinaison de types de promotion susceptible d'être appliquée. Au cours d'une année il peut y avoir 1 000 annonces, 5 000 réductions de prix temporaires et 1 000 présentations en tête de gondole, mais il peut n'y avoir que 10 000 combinaisons de ces trois types de promotion pour un produit quel qu'il soit. Par exemple, pour une certaine promotion, presque tous les magasins feraient appel simultanément aux trois mécanismes promotionnels, mais quelques-uns ne seraient pas en mesure de déployer les présentations en tête de gondole. On aurait besoin dans ce cas de deux enregistrements distincts de modalité de promotion, un pour la réduction de prix normale, plus l'annonce, plus la présentation, un autre pour une réduction de prix exceptionnelle plus l'annonce seulement. La table de dimension promotion dont nous recommandons l'utilisation est présentée à la figure 2.9.

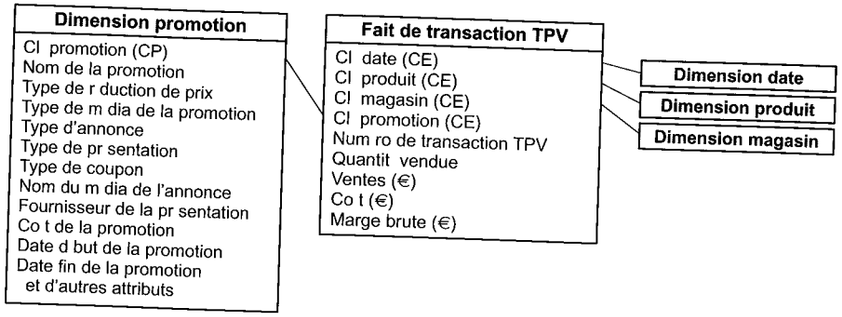

Figure 2.9 **Dimension promotion du schéma distribution alimentaire**

D'un point de vue strictement logique nous pourrions enregistrer à peu près la même information sur les promotions en séparant les quatre mécanismes de promotion (réductions de prix, annonces, présentations et coupons) en quatre dimensions distinctes au lieu de les combiner en une seule dimension. Ce choix est en définitive la prérogative du concepteur. Voici les avantages qui résultent de la combinaison des dimensions :

- Les quatre mécanismes de modalités causales sont fortement corrélés, de sorte que la dimension unique est seulement un peu plus grande que n'importe laquelle des dimensions séparées.
- La dimension unique combinée peut être parcourue efficacement pour voir les diverses manières d'utiliser conjointement les réductions de prix, les annonces, les présentations et les coupons. Cependant cette navigation montre seulement les combinaisons possibles. Elle ne révèle pas les magasins ou les produits affectés par une promotion. Cette information se trouve dans la table de faits.

La séparation des quatre modalités causales en tables de dimension distinctes présente les avantages suivants :

- Les dimensions séparées peuvent être plus compréhensibles pour les utilisateurs dans la mesure où ils tendent à les envisager séparément. Leur façon de voir peut être révélée au cours des interviews sur les exigences de l'activité.
- La gestion des dimensions séparées peut être plus simple que celle d'une dimension combinée.

Notez que les informations disponibles dans l'entrepôt de données sont les mêmes pour ces deux variantes.

En général, un grand nombre de transactions portent sur des ventes d'articles sans promotion. C'est pourquoi nous devons aussi inclure dans la dimension promotion une ligne comportant sa propre clé unique, pour préciser « pas de promotion en cours » et éviter d'avoir une clé promotion sans valeur (*null*) dans la table de faits. L'intégrité référentielle serait violée par la présence d'un null dans une colonne de la table de faits déclarée comme clé étrangère pour une table de dimension. En plus des dangers pour l'intégrité référentielle, les clés null sont une source de perplexité pour les utilisateurs, qui ne peuvent pas faire de jointures sur de telles clés.

Vous devez éviter les clés *null* dans les tables de faits. Une conception correcte consiste à placer une ligne dans la table de dimension correspondante pour préciser que la dimension ne s'applique pas au fait mesuré.

Table de faits sans fait relative aux promotions

Quelle que soit la manière dont nous traitons la dimension promotion, il reste une question importante négligée jusqu'ici par notre schéma pour la distribution : quels sont les produits qui étaient en promotion et qui n'ont pas été vendus ? La table de faits n'enregistre que les US effectivement vendues. Il n'y a pas de ligne avec des faits égaux à zéro pour les articles qui n'ont pas été vendus, parce que le volume de la table deviendrait alors démesuré. Dans l'univers relationnel, il est nécessaire de prévoir une seconde table de faits relative aux promotions ou aux événements pour répondre aux questions concernant ce qui ne s'est pas produit. Les clés de cette table de faits des promotions seraient dans notre étude de cas la date, le produit, le magasin et la promotion. Cela ressemble évidemment à la table de faits des ventes que nous venons de concevoir, mais ici le grain est nettement différent. Dans le cas de la table de faits pour la couverture des promotions, nous insérons une ligne pour chaque produit en promotion dans un magasin chaque jour (ou semaine, puisque de nombreuses promotions durent une semaines) que le produit soit vendu ou non. Cette table de faits de couverture nous permet de voir la relation entre les clés telles qu'elles sont définies pour une promotion, indépendamment d'autres événements tels que les ventes effectives de produits. Nous l'appelons *table de faits sans fait* parce qu'elle ne comporte pas de mesure ; elle ne fait que saisir la relation entre les clés concernées. La détermination des produits en promotion qui ne se sont pas vendus est un processus en deux temps. Nous commençons par une requête sur la table de couverture des promotions pour déterminer le périmètre des produits en promotion un jour donné. Nous déterminons ensuite ceux de ces produits qui ont été vendus à partir de la table de faits des ventes. La réponse à notre question de départ est la différence des deux ensembles. Nous reviendrons plus en détail sur les tables de faits sans fait au chapitre 12, qui comporte un exemple de table de couverture des promotions et le code SQL donnant la différence des ensembles. Si vous travaillez avec des données dans un environnement de cube OLAP (*Online Analytical Processing*), dit aussi multidimensionnel, il vous sera en général plus facile de répondre à la question sur ce qui ne s'est pas vendu parce que le cube contient généralement des cellules explicites pour l'absence de fait.

Dimension numéro de transaction dégénérée

La table de faits des ventes au détail mentionne un numéro de transaction TPV pour chaque ligne d'article vendu. Dans une base de données classique du type parent-enfant, le numéro de transaction TPV serait la clé de l'enregistrement entête pour la transaction, contenant toutes les informations valables pour l'ensemble de la transaction, comme la date et l'identification du magasin. Cependant, dans notre modèle dimensionnel, nous avons déjà extrait cette intéressante information d'en-tête pour la placer dans d'autres dimensions. Le numéro de transaction TPV garde néanmoins son intérêt en tant que clé permettant de regrouper tous les produits achetés au cours d'une même transaction.

Bien que le numéro de transaction situé dans la table de faits ait l'air d'une clé de dimension, il ne pointe vers aucun élément descriptif, car tous ceux qu'il comportait lui ont été enlevés. La dimension correspondante étant vide, nous disons que le numéro de transaction TPV est une *dimension dégénérée* (identifiée par la notation DD à la figure 2.10). Les dimensions dégénérées sont très courantes lorsque le grain d'une table de faits est au niveau d'une seule transaction ou d'un élément d'une transaction, car elles représentent alors l'unique identificateur du parent. Les numéros de commande, de facture et de lettre de transport apparaissent le plus souvent sous forme de dimension dégénérée dans un modèle dimensionnel.

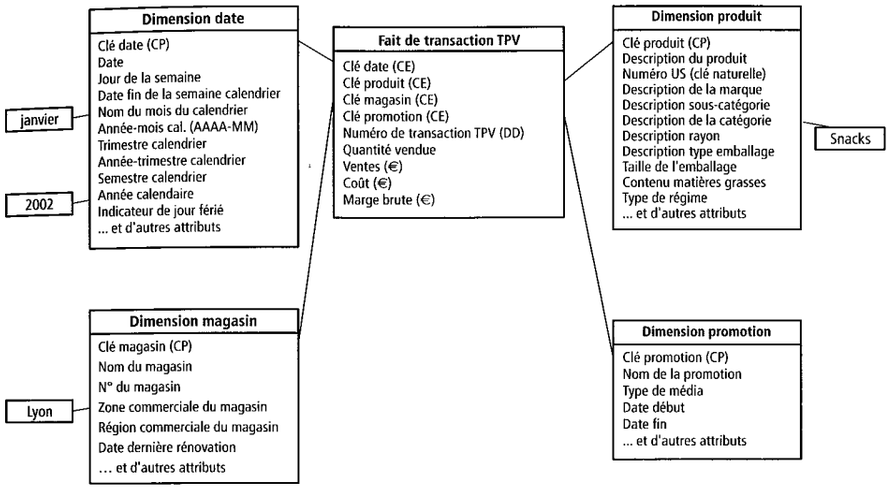

Figure 2.10 **Requête sur le schéma de vente au détail**

Les dimensions dégénérées font souvent partie de la clé primaire dans une table de faits. Dans notre étude de cas, la clé primaire de la table de faits se compose du numéro de transaction TPV dégénéré et de la clé produit (dans l'hypothèse où le système TPV cumule en une seule ligne d'article tous les exemplaires d'un même produit qui se trouvent dans le panier de l'acheteur). Souvent, la clé primaire d'une table de faits est un sous-ensemble des clés étrangères de la table. En effet, généralement, l'association de toutes les clés étrangères n'est pas indispensable pour garantir qu'on se réfère à une seule ligne d'une table de faits.

Les numéros d'identification opérationnels tels que numéros de commande, numéros de facture et numéros de lettre de transport donnent souvent naissance à des dimensions vides appelées dimensions dégénérées (c'est-à-dire des clés de dimension qui ne correspondent pas à des tables de dimension) dans des tables de faits dont le grain est le document lui-même ou une ligne du document.

Si pour une raison quelconque un ou plusieurs des attributs subsistent après la création de toutes les autres dimensions et se rapportent à cet élément de type en-tête, nous créons alors un enregistrement normal de dimension. Mais alors nous n'avons plus affaire à une dimension dégénérée.

2.4 Mise en œuvre du schéma de grande distribution

Ayant fait la conception de notre schéma pour la grande distribution, voyons maintenant son utilisation au niveau des requêtes. Un utilisateur pourrait chercher à mieux comprendre le montant des ventes hebdomadaires en euros dans la catégorie des snacks pour la zone de Lyon en janvier 2002. Comme on peut le voir à la figure 2.10, nous plaçons des contraintes sur le mois et l'année dans la dimension date, la zone dans la dimension magasin et la catégorie dans la dimension produit.

Si l'outil de requête fait le total des montants en euros groupés par date de fin de semaine et par promotion, nous obtenons un résultat semblable à celui indiqué plus loin. Il met en évidence la relation entre le modèle dimensionnel et la requête correspondante. La qualité des attributs dimensionnels est primordiale car ils permettent les contraintes de la requête et fournissent les intitulés des ensembles de résultats.

Date de fin semaine calendaire	Nom de la promotion	Montant ventes en euro
6 janvier 2002	Pas de promotion	22 647
13 janvier 2002	Pas de promotion	10 851
20 janvier 2002	Promotion Petite faim	11 248
27 janvier 2002	Promotion Petite faim	13 793

Avec un outil d'accès aux données plus sophistiqué, les résultats peuvent apparaître sous forme de tables croisées. Ce genre d'état est plus parlant pour les utilisateurs que les résultats en colonnes produits par des instructions SQL.

Date de fin semaine calendaire	Ventes promotion Montant en euro	Pas de promotion Montant ventes en euro
6 janvier 2002	0	22 647
13 janvier 2002	0	10 851

Date de fin semaine calendaire	Ventes promotion Montant en euro	Pas de promotion Montant ventes en euro
20 janvier 2002	10 248	0
27 janvier 2002	13 793	0

2.5 Extensibilité du schéma de grande distribution

Ayant mené à bien notre premier modèle dimensionnel, portons maintenant notre attention sur l'extension de la conception. Supposons que notre distributeur décide de mettre en place un programme de fidélisation des clients. Désormais, au lieu de savoir qu'un acheteur ou une acheteuse avait 26 articles dans son panier, nous allons être en mesure de savoir exactement ce qu'un acheteur particulier, par exemple Julie Kimball, achète par semaine. Représentez-vous l'intérêt potentiel des utilisateurs de l'entrepôt de données pour l'analyse des structures d'achat selon une multitude de caractéristiques différenciant les acheteurs, comme la géographie, la démographie, le comportement etc.

Le traitement de ces nouvelles informations sur les acheteurs réguliers est relativement aisé. Nous pourrions créer une table de dimension acheteur régulier et ajouter une clé étrangère de plus dans la table de faits. Comme nous ne pouvons pas demander aux acheteurs de rapporter tous leurs reçus de caisse enregistreuse pour étiqueter leurs transactions d'achat passées avec leur nouveau numéro de client régulier, nous introduisons une clé d'acheteur correspondant à « avant le programme client régulier » sur les lignes de la table de faits se rapportant au passé. De même, tous les acheteurs du magasin n'ont pas de carte de client régulier et donc nous introduisons une ligne « client régulier non identifié » dans notre table de dimension acheteur régulier. Comme nous l'avons vu précédemment à propos de la dimension promotion, nous devons éviter les clés null dans la table de faits.

De la même façon que nous avons enrichi notre conception d'origine d'une dimension client régulier, nous pourrions aussi ajouter des dimensions pour l'heure de la journée et l'employé associé à la transaction, comme à la figure 2.11. Tout attribut descriptif ayant une valeur unique pour une ligne de mesures dans la table de faits est un bon candidat pour être soit ajouté à une dimension existante, soit être sa propre dimension. La question de savoir si une dimension peut ou non être attachée à une table de faits dépend du grain qui a été déclaré. En cas de doute, reportez-vous à l'étape 2 du processus de conception.

Si notre schéma d'origine s'adapte facilement pour accueillir ces nouvelles dimensions, c'est avant tout parce que nous avons modelé nos données de transaction TPV avec la plus fine granularité. Les nouvelles dimensions à cette même granularité ne modifient pas les clés des dimensions ni les faits existants ; toutes les applications antérieures continuent de fonctionner comme avant, sans modification. Si nous avions précédemment choisi comme grain la vente journalière (tran-

sactions récapitulées par jour, magasin, produit et promotion) au lieu de la ligne de transaction, nous n'aurions pas pu incorporer facilement les dimensions client régulier, heure du jour ou employé. L'agrégation prématurée conduit à un grain plus grossier, ce qui réduit les possibilités d'ajout de dimensions supplémentaires, lesquelles ne s'appliquent généralement pas au grain supérieur.

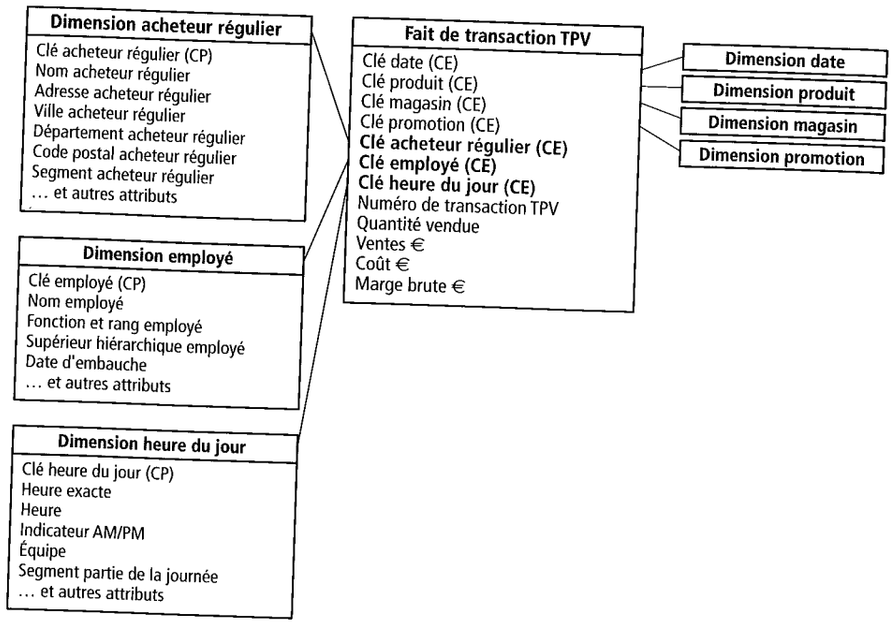

Figure 2.11 **Un schéma de vente au détail enrichi**

Il y a évidemment des changements que l'on ne peut jamais traiter en douceur. Si une source de données cesse d'être disponible et si l'on ne dispose d'aucune source de substitution compatible, les applications de l'entrepôt de données qui dépendent de cette source ne fonctionneront plus. Toutefois, la symétrie inhérente aux modèles dimensionnels leur permet d'absorber des changements assez importants des données sources ou des hypothèses de modélisation, sans invalider les applications existantes. Nous allons décrire plusieurs types de modifications imprévues, en commençant par les plus simples :

Nouveaux attributs de dimension. Si nous découvrons de nouvelles descriptions textuelles d'un produit, par exemple, nous ajoutons ces attributs à la dimension sous forme de nouvelles colonnes. Toutes les applications existantes ignoreront ces nouveaux attributs et continueront de fonctionner. Si les nouveaux attributs ne sont disponibles qu'à partir d'un certain point dans le temps, il convient de placer la mention « non disponible » ou une mention analogue dans les anciens enregistrements de dimension.

Nouvelles dimensions. Comme nous venons de le voir à la figure 2.11, nous pouvons ajouter une nouvelle dimension à une table de faits existante en ajoutant un nouveau champ de clé étrangère et en le renseignant correctement avec les valeurs de la clé primaire de la nouvelle dimension.

Nouveaux faits mesurés. Si de nouveaux faits mesurés apparaissent, nous pouvons les ajouter en douceur à la table de faits. Dans le cas le plus simple, les nouveaux faits sont rendus disponibles par le même événement de mesure et sont au même grain que les faits existants. On ajoute alors à la table de faits de nouvelles colonnes, que l'on renseigne des faits appropriés. Si l'on ne peut procéder par l'instruction ALTER TABLE, il faut alors définir une seconde table de faits, comportant les anciennes et les nouvelles colonnes et dans laquelle les valeurs sont copiées à partir de la table à remplacer. Si les nouveaux faits ne sont disponibles qu'à partir d'un certain point dans le temps, des valeurs null doivent être placées sur les lignes de faits antérieures. Une situation plus complexe est celle où les nouveaux faits mesurés se produisent naturellement à un grain différent. Si les nouveaux faits ne peuvent pas être convertis au grain d'origine de la table de faits, ces faits doivent presque certainement être placés dans une table de faits différente. C'est presque toujours une erreur de mélanger des grains dans la même table de faits.

Une dimension devient plus granulaire. Il est parfois souhaitable d'augmenter la granularité d'une dimension. Dans la plupart des cas, les attributs de la dimension d'origine peuvent être inclus dans la nouvelle dimension à un grain plus fin grâce à la relation de plusieurs-à-un entre la nouvelle et l'ancienne dimension. Une dimension à un grain plus fin implique souvent une table de faits également plus granulaire. Il peut ne pas y avoir d'autre solution que d'abandonner la table de faits et de la reconstruire. Toutefois, les applications existantes n'en seront pas affectées.

Addition d'une source de données entièrement nouvelle mettant en jeu les dimensions existantes ainsi que de nouvelles dimensions imprévues. Presque toujours, une nouvelle source de données a sa propre granularité et ses propres dimensions et c'est pourquoi nous créons une nouvelle table de faits. Évitez de faire rentrer de force de nouvelles mesures dans une table de faits contenant des mesures cohérentes. Les applications existantes continueront de fonctionner parce que les tables de faits et de dimension existantes sont inchangées.

2.6 Ne pas céder à la facilité

Ayant conçu un premier modèle dimensionnel, examinons maintenant plusieurs tendances des concepteurs venant d'un environnement normalisé. Nous contrevenons volontairement à diverses règles de modélisation parce que notre objectif est la commodité d'utilisation et la performance, non l'efficacité du traitement transactionnel.

Normalisation des dimensions (flocons de neige)

Les tables de dimension aplaties, dénormalisées avec des valeurs textuelles qui se répètent, créent une sensation de malaise chez un concepteur habitué à la normalisation. Revoyons la table de dimension denotre étude de cas. Les 150 000 produits se répartissent dans cinquante rayons distincts. Au lieu de stocker de façon redondante les 20 octets de description des rayons dans la table de dimension, un concepteur formé à la normalisation veut stocker un code rayon de deux octets, puis créer une nouvelle dimension rayon pour le décodage des rayons. En fait, il serait plus à l'aise si toutes les descriptions de notre modèle original étaient normalisées et placées dans des tables de dimension séparées. Son argument est que cette approche économise de la place parce que nous n'avons plus que des codes non significatifs dans notre table de dimension à 150 000 lignes au lieu de descriptions encombrantes.

En outre, certains concepteurs affirment que la conception normalisée pour les tables de dimension est plus facile à tenir à jour. Si la description d'un rayon change, il n'y a qu'une occurrence à mettre à jour au lieu des 3 000 occurrences du rayon dans notre dimension produit. La mise à jour est souvent effectuée en recourant à des règles de normalisation, mais n'oubliez pas que cette opération s'effectue dans la zone de préparation, bien avant que les données ne soient chargées dans le schéma dimensionnel de la zone de présentation.

La normalisation des tables de dimension est souvent décrite comme produisant des flocons. Les attributs redondants sont supprimés de la table de dimension plate dénormalisée et placés dans des tables de dimension secondaires normalisées. La figure 2.2 illustre la mise en flocons partielle de notre schéma. Poussée à l'extrême, elle conduirait à un diagramme entité-relation en troisième forme normale. Le contraste entre la figure 2.12 et la figure 2.10 de la conception d'origine est saisissante. Les deux tables de faits des deux figures sont identiques, mais on est frappé par la pléthore de tables de dimension (même dans notre présentation simplifiée) de la seconde figure.

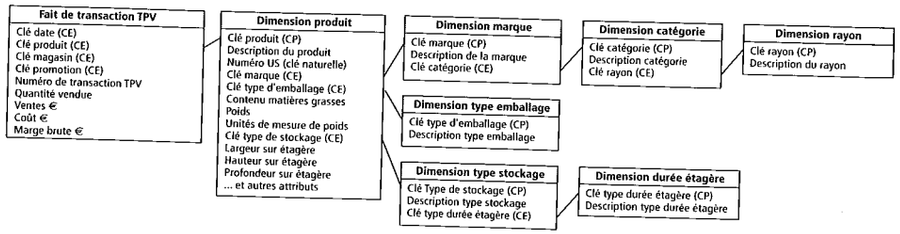

Figure 2.12 **Dimension produit partiellement en flocons de neige**

Bien que les flocons soient une extension acceptable du modèle dimensionnel, nous vous invitons néanmoins à y renoncer, pour mieux atteindre les objectifs majeurs de la conception, qui sont la commodité d'utilisation et la performance.

- La multitude des tables en flocons complexifie la présentation à l'envi, ce qui ne manque pas de perturber les utilisateurs. Souvenez-vous que la simplicité est l'un des principaux objectifs du modèle dimensionnel dénormalisé.
- Les fonctions d'optimisation des SGBD seront elles aussi perturbées par la complexité du schéma en flocons. L'augmentation du nombre de tables et de jointures se traduit habituellement par une moindre performance des requêtes. La complexité des jointures augmente le risque de voir l'optimiseur se fourvoyer et choisir une approche médiocre.
- L'économie d'espace disque associée à une dimension en flocons est insignifiante. Si nous remplaçons la description en 20 octets dans la table de dimension produit à 150 000 lignes par un code de 2 octets, nous économisons 2,7 Mo (150 000 × 18 octets), mais notre table de faits peut stocker 10 Go ! Les tables de dimension sont presque toujours de plusieurs ordres de grandeur plus petites que les tables de faits. Les efforts visant à normaliser la plupart des tables de dimension pour économiser l'espace disque sont une perte de temps.
- Les flocons de neige réduisent les possibilités de navigation à l'intérieur d'une dimension. La navigation implique souvent des contraintes sur un ou plusieurs attributs de dimension et l'examen des valeurs distinctes d'un autre attribut en présence de ces contraintes. La navigation permet aux utilisateurs de comprendre les relations entre les valeurs des attributs d'une dimension.

  De toute évidence, une table de dimension en flocons conviendra si nous voulons seulement une liste des descriptions d'une catégorie. Mais si nous voulions voir toutes les marques au sein d'une catégorie, nous devrions traverser les dimensions marque et catégorie. Si nous voulons ensuite lister les types d'emballage de chaque marque dans une catégorie, nous devrions traverser encore plus de tables. Le SQL nécessaire pour effectuer ces requêtes d'apparence simple est complexe, alors que nous n'avons même pas touché aux autres dimensions ni à la table de faits.
- Enfin, les flocons de neige empêchent l'utilisation des index sous forme de bitmaps. Ces index bitmap sont très utiles pour indexer des champs de faible cardinalité, tels que les colonnes catégorie et rayon de nos tables de dimension produit. Ils augmentent fortement la performance d'une requête ou d'une contrainte sur l'unique colonne en question. Les flocons vous empêchent de tirer parti de cette technique d'amélioration des performances.

Les tables de dimension doivent rester des tables physiques plates. Les tables de dimension normalisées en flocons de neige pénalisent la navigation au sein des attributs d'une même table et s'opposent à l'utilisation des index bitmap. Les économies d'espace disque résultant de la normalisation des tables de dimension sont inférieures à 1 % du total de l'espace disque requis pour l'ensemble du schéma. Nous renonçons en connaissance de cause à cette économie en vue d'une plus grande facilité d'utilisation et d'une meilleure performance.

Il y a des cas où les flocons sont admissibles, comme dans notre exemple précédent mettant en jeu une table déportée avec les dates dans la table de dimension magasin, lorsqu'un groupe d'attributs corrélés est utilisé de façon répétée dans différents rôles indépendants. Nous vous invitons seulement à utiliser cette approche avec la plus grande modération, la réservant aux situations où son emploi est réellement avantageux.

Trop de dimensions

La table de faits d'une schéma dimensionnel est naturellement très normalisée et compacte. Il n'y a aucun moyen de normaliser davantage les relations extrêmement complexes de plusieurs-à-plusieurs entre les clés de la table de faits parce que les dimensions ne sont pas corrélées. Chaque magasin est ouvert chaque jour. Tôt ou tard, presque chaque produit est vendu en promotion dans la plupart des magasins, voire dans tous les magasins.

Il est curieux de constater qu'en dépit du malaise que leur procure les tables de dimension dénormalisées, certains concepteurs sont tentés de dénormaliser la table de faits. Au lieu d'avoir une clé étrangère unique dans la table de faits, ils prévoient des clés étrangères pour des éléments de la hiérarchie produit fréquemment analysés, comme la marque, la sous-catégorie, la catégorie et le rayon. De même, la clé date se transforme brusquement en une série de clés permettant des jointures sur des dimensions semaine, mois, trimestre et année. Bientôt, notre table de faits compacte se métamorphose en un monstre qui peut se joindre littéralement à des douzaines de tables de dimension. Nous conférons affectueusement à ces structures l'appellation *mille-pattes*, compte tenu de leur ressemblance avec cet insecte familier, comme on peut le voir à la figure 2.13. Il est clair que la conception représentée est tombée dans le piège des dimensions trop nombreuses.

N'oubliez surtout pas que malgré son format très serré, la table de faits est en réalité un véritable colosse dans une conception dimensionnelle. La conception d'une table de faits avec trop de dimensions en augmente notablement le volume. Alors que nous n'hésitons pas à utiliser un plus grand espace disque pour les tables de dimension, nous surveillons la taille de la table de faits, qui est de plusieurs ordres plus grande que les autres tables. Il n'y a aucun moyen d'indexer efficacement l'énorme clé à parties multiples de notre exemple de mille-pattes. Les

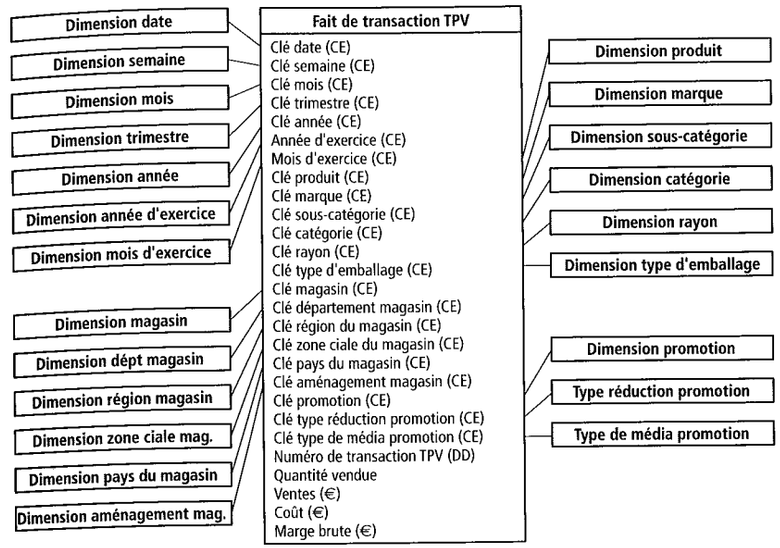

Figure 2.13 **Table de faits mille-pattes avec un trop grand nombre de dimensions**

nombreuses jointures qu'il impose réduisent la commodité d'utilisation et la performance.

La plupart des processus d'entreprise peuvent être représentés avec moins de 15 dimensions dans la table de faits. Si notre conception a 25 dimensions ou plus, nous devons rechercher des moyens de combiner des dimensions corrélées en une seule dimension. Des attributs parfaitement corrélés, tels que les différents niveaux d'une hiérarchie de même que des attributs avec une corrélation statistique raisonnable, doivent faire partie d'une même dimension. Vous avez obtenu un résultat satisfaisant quand l'unique dimension résultante est nettement plus petite que le produit cartésien des dimensions que vous avez combinées.

Un nombre très élevé de dimensions est en général le signe que certaines dimensions ne sont pas complètement indépendantes et doivent être combinées en une seule dimension. Représenter les éléments d'une hiérarchie sous forme de dimensions séparées dans la table de faits est contraire aux principes de la modélisation dimensionnelle.

2.7 Clés artificielles

Nous recommandons vivement l'usage de clés artificielles dans les modèles dimensionnels plutôt que de s'appuyer sur des codes venant du système opérationnel. Les clés artificielles ont de nombreuses autres appellations, *clés sans signification,*

clés entières, clés synthétiques etc. En bref, les clés artificielles sont des entiers affectés séquentiellement lors du garnissage d'une dimension. Par exemple, le premier enregistrement produit se voit assigner la clé artificielle de valeur 1, le suivant la clé artificielle de valeur 2 et ainsi de suite. Les clés artificielles servent simplement à joindre les tables de dimension à la table de faits.

Les concepteurs hésitent parfois à abandonner leurs clés naturelles parce qu'ils veulent parcourir la table de faits sur la base du code opérationnel en évitant une jointure sur la table de dimension. Souvenez-vous cependant que les tables de dimension sont notre point d'entrée dans la table de faits. Si les positions cinq à neuf du code opérationnel identifient le fabricant, alors le nom du fabricant doit être inclus en tant qu'attribut dans la table de dimension. D'une manière générale, nous devons éviter d'inclure des informations dans les clés de l'entrepôt de données, car toute hypothèse faite sur la signification d'une portion de clé risque d'être invalidée. De même, les requêtes et les applications d'accès aux données ne doivent pas dépendre des clés naturelles, car leur logique pourrait être invalidée par des modifications du sens de ces clés.

Toute jointure entre des tables de dimension et des tables de faits doit se baser sur des clés entières artificielles sans signification. Évitez d'utiliser les codes opérationnels naturels des systèmes de production. Aucune clé de l'entrepôt de données ne doit être intelligible, c'est-à-dire suffire à elle seule à donner une indication sur la ligne concernée.

Il peut être initialement plus rapide de créer un modèle dimensionnel en utilisant des codes opérationnels, mais les clés artificielles sont toujours payantes sur la durée. Nous les comparons parfois à un vaccin de l'entrepôt de données contre la grippe — comme pour un vaccin, il est un peu douloureux d'initialiser et de gérer les clés artificielles, mais les avantages qu'on en tire sur la durée sont importants.

L'un des principaux avantages des clés artificielles est qu'elles représentent un tampon protégeant l'environnement de l'entrepôt de données à l'égard des modifications opérationnelles. Elles permettent à l'équipe de l'entrepôt de données de garder le contrôle de l'environnement au lieu d'être assujetti à des règles opérationnelles pour la génération, la mise à jour, la suppression, le recyclage de codes de production. Dans maintes organisations, les codes opérationnels historiques (par exemple, les numéros de compte inactifs ou les codes produit obsolètes) sont recyclés après une période d'inactivité. Si des numéros de compte sont recyclés après douze mois d'inactivité, les systèmes opérationnels n'auront pas à en pâtir parce que leur mode de fonctionnement ne prévoit pas la conservation d'informations sur une telle durée. En revanche, l'entrepôt de données conserve des données pendant des années. Les clés artificielles dotent l'entrepôt de données d'un mécanisme de différenciation entre deux significations d'un même numéro de compte opérationnel. Si nous nous fions aux codes opérationnels, nous nous exposons à

des problèmes de recouvrement de clés en cas d'acquisition ou de consolidation de données. Les clés artificielles permettent à l'équipe de l'entrepôt de données d'intégrer des données provenant de multiples systèmes opérationnels sources, même si les clés ne sont pas cohérentes.

Les clés artificielles offrent également un avantage sous l'angle des performances. La clé artificielle est le plus petit entier possible, tout en garantissant qu'elle pourra accueillir sans difficulté une augmentation quelconque du nombre maximal de lignes de la dimension. Souvent, le code opérationnel est une chaîne encombrante de caractères alphanumériques. La taille plus modeste de la clé artificielle se traduit par des tables de faits plus petites, des index de table de faits plus petits et un plus grand nombre de lignes de table de faits à chaque opération d'entrée/sortie. Un entier de 4 octets (non quatre positions décimales) est généralement suffisant. Ses 32 bits peuvent traiter approximativement 2 milliards de valeurs positives (2^{32-1}) ou 4 milliards de valeurs négatives et positives (-2^{32-1} à $+2^{32-1}$). N'oubliez pas que dans le cas d'une grande table de faits d'un milliard de lignes, tout octet supplémentaire par ligne exige un Go d'espace disque supplémentaire.

Rappelons que les clés artificielles servent aussi à représenter des conditions de dimension qui n'ont pas de code opérationnel, comme la condition « pas de promotion en cours ».

Dans le même ordre d'idées, vous pouvez avoir affaire dans un modèle dimensionnel à des dates qui ne sont pas encore déterminées. Il n'y a pas de valeur de date en SQL telle que « date à déterminer » ou « date non applicable ». C'est une raison de plus pour utiliser des clés artificielles comme clés de date au lieu du format date du SQL.

La dimension date est la seule dimension pour laquelle les clés artificielles doivent obligatoirement être affectées dans un ordre séquentiel significatif. Ainsi, on affecte au 1er janvier de la première année la valeur de clé artificielle 1, au 2 janvier la valeur 2, au 2 février la valeur de clé artificielle 32 et ainsi de suite. Ne donnez pas à ces clés une forme suggérant d'autre signification que le rang (par exemple AAAA-MM-JJ). Cette pratique peut encourager certains utilisateurs à délaisser la table de dimension d'accès aux dates et bien sûr elle ne permet pas de représenter la valeur « ne s'est pas encore produit » ni d'autres cas de date assez courants. Nous voulons seulement que nos lignes de table de faits soient dans l'ordre séquentiel. Le traitement de la clé artificielle date, comme un numéro d'ordre des dates, permet d'avoir une table de faits dont les partitions sont basées sur la clé date. Dans le cas d'une grande table de faits, cette disposition permet d'éliminer en douceur les anciennes données et de charger et indexer les nouvelles données sans toucher au reste de la table de faits.

Enfin, les clés artificielles sont nécessaires au support de l'une des principales techniques de traitement des modifications des attributs des tables de dimension. C'est d'ailleurs l'une des justifications les plus importantes pour l'emploi des clés

artificielles. Nous consacrerons une section entière du chapitre 4 à l'utilisation des clés artificielles pour les dimensions à évolution lente.

Naturellement, l'affectation et la gestion de clés artificielles imposent un certain effort, mais il n'y a pas lieu de s'en effrayer. Nous devons simplement créer et tenir à jour une table de références croisées dans la zone de préparation, servant à insérer la clé étrangère appropriée dans chaque ligne de table de faits et de dimension. Le chapitre 16 présente un organigramme de la gestion et du traitement des clés artificielles des schémas dimensionnels.

Avant de quitter le sujet des clés, nous voudrions décourager l'usage de clés concaténées ou composées pour les tables de dimension. On ne peut pas créer une véritable clé étrangère en regroupant simplement des clés naturelles ou en combinant la clé naturelle avec un groupe date-heure. Nous voulons aussi éviter les multiples jointures parallèles entre les tables de dimension et les tables de faits, qui ont une incidence néfaste sur les performances.

Nous n'affectons généralement pas de clé artificielle à des dimensions dégénérées, mais il convient d'en décider en fonction de chaque cas particulier. Une clé artificielle s'impose si les numéros d'identification des transactions ne sont pas uniques pour l'ensemble des magasins ou s'ils sont réutilisés. Par exemple, le système TPV peut ne pas être en mesure d'affecter des numéros de transaction distincts pour les différents magasins, ou il peut revenir à zéro et réutiliser les numéros précédents après avoir atteint un maximum. Il se peut aussi que le numéro d'identification de la transaction soit une colonne encombrante de 24 octets. Dans tous ces cas, les clés artificielles offrent des avantages. Techniquement, des dimensions de numéro d'identification modélisées de cette façon ne sont plus des dimensions dégénérées.

Supposons pour le moment que la première version de notre schéma de vente au détail représente aussi bien la conception logique que la conception physique de notre base de données. Autrement dit, que la base de données relationnelle ne contienne que cinq tables réelles : table de faits des ventes et tables de dimension date, produit, magasin et promotion. Chacune des tables de dimension a une clé primaire et la table de faits comporte une clé composite faite de ces quatre clés étrangères, en plus du numéro de transaction dégénéré. L'aspect le plus frappant de la conception à ce stade est probablement la simplicité de la table de faits. Si les quatre clés étrangères sont des entiers consécutifs gérés parcimonieusement, nous pouvons nous contenter de 14 octets pour l'ensemble des 4 clés (4 octets chacun pour date, produit et promotion et 2 octets pour les magasins). Le numéro de transaction peut demander 8 octets supplémentaires. Si les quatre faits de la table de faits avaient 4 octets chacun, nous n'aurions besoin de réserver que 16 autres octets. Notre ligne de table de faits n'aurait qu'une largeur de 38 octets. Même avec un milliard de lignes, la table de faits n'occuperait que 38 Go d'espace. Une ligne de table de faits aussi ramassée est l'un des effets typiques de la conception dimensionnelle.

Notre schéma de vente au détail, présenté à la figure 2.11, s'enrichit de trois dimensions supplémentaires. Si nous allouons 4 octets chacun au client et à l'employé et 2 octets pour l'heure de la journée (à une minute près), notre table de faits n'augmentera que de 48 octets. Notre table de faits de un milliard de lignes n'occupe alors que 48 Go.

2.8 Analyse de panier de marché

Notre schéma des ventes au détail nous dit de façon très précise ce qui a été acheté dans chaque magasin et dans quelles conditions. Toutefois, ce schéma ne nous permet pas de déterminer facilement les produits qui ont été achetés ensemble. Cette analyse des combinaisons de produits cherche à déterminer des *groupements par affinité*, mais on utilise aussi le terme plus parlant d'*analyse de panier de marché*. Ce type d'analyse donne au commerçant des indications sur la façon de présenter diverses combinaisons d'articles. Si les plats de pâtes surgelés se vendent bien avec des produits à base de cola, peut-être convient-il de placer ces produits à proximité l'un de l'autre ou de les associer au niveau du prix. Le concept d'analyse de panier de marché peut être facilement étendu à d'autres domaines. Dans le contexte de la fabrication, il est utile de savoir quels produits sont commandés ensemble, pour éventuellement proposer des offres spéciales avec un prix attractif.

La table de faits des ventes aux détail ne se prête pas facilement aux analyses de panier de marché parce que SQL n'a jamais été prévu pour faire des groupements et des contraintes impliquant différentes lignes de faits. Les outils de forage de données et certains produits OLAP peuvent faciliter les analyses de panier de marché, mais à défaut, nous vous proposons ci-après une approche plus directe. Il s'agit d'une technique assez avancée, de sorte que si vous n'avez pas besoin actuellement d'une analyse de panier de marché, vous pouvez vous contenter d'un survol de cette section pour avoir une idée générale des techniques impliquées.

Nous représentons à la figure 2.14 une table de faits d'analyse de panier de marché issue de transactions de vente au détail. La table de faits panier de marché est un instantané périodique des paires de produits achetés ensemble au cours d'une période de temps spécifique. Les faits comportent le nombre total de paniers (tickets de client) qui contenaient les produits A et B, le cumul des valeurs en euros et des quantités pour le sous-ensemble des achats du produit A, le cumul des valeurs en euros et des quantités pour ce sous-ensemble des achats du produit B. Le nombre de paniers est un fait semi-additif. Par exemple, si un ticket de client comporte des lignes pour les pâtes alimentaires, les boissons non alcoolisées et le fromage, apparaissant dans la table de faits de panier de marché, cette combinaison est comptée une fois sur la ligne pâtes et boissons non alcoolisée, une fois sur la ligne pâtes et fromage et une fois sur la ligne boissons non alcoolisées et fromage. Il est clair qu'il faut éviter d'additionner des comptages portant sur plus d'un produit.

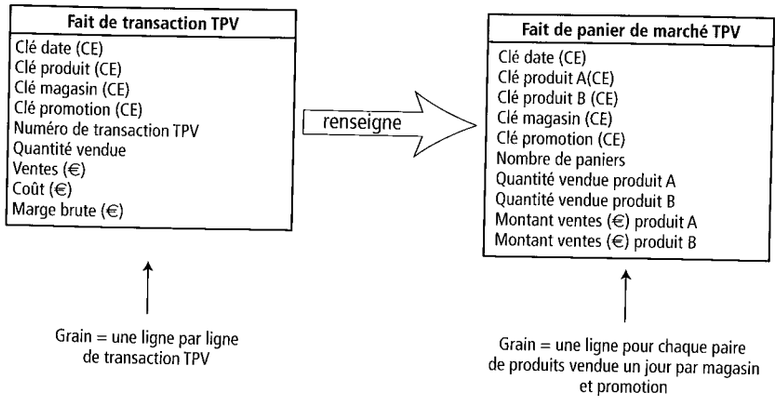

Figure 2.14 **Table de faits de panier de marché renseignée à partir des transactions d'achat**

Vous remarquerez la présence dans la table de faits panier de marché de deux clés produit généralisées (les clés produit A et B). Nous avons construit ici une seule table de dimension produit contenant des entrées à de multiples niveaux de la hiérarchie, tels que le niveau des produits individuels, ceux des marques et des catégories. Cette variante de notre table de dimension produit normale contient un petit nombre d'attributs de nature assez générique. Les clés artificielles pour les différents niveaux de la hiérarchie des produits ont été affectées en évitant tout recouvrement.

Si le concept d'enregistrement des corrélations entre les éléments du panier de marché est simple, le nombre des combinaisons de produits fait de l'analyse de ces corrélations un véritable défi. Si nous avons N produits en magasin et si nous essayons de construire une table de tous les couples de produits possibles, cette table aura $N \times (N - 1)$ éléments. Par exemple, pour 10 000 produits, il y aura près de 100 000 000 couples possibles. Le nombre de combinaisons atteint des niveaux absurdes dans le cas d'un grand nombre de produits. Si un magasin de détail compte 100 000 US, il y aura près de dix milliards de combinaisons.

Le secret de l'analyse du panier de marché tient dans le fait que le but de l'opération est d'analyser les seules combinaisons *significatives* de produits vendus ensemble. Dans le cas de notre table de faits panier de marché, nous nous intéressons d'abord aux lignes ayant un comptage élevé de paniers de marché. Ces combinaisons de produits fréquemment observées méritent d'être analysées de plus près. Ensuite, nous recherchons les cas où les montants ou les quantités de produits A et B sont raisonnablement équilibrés. Si les montants ou les quantités sont très différents, nous avons affaire à des produits qui se vendent bien associés à des produits secondaires insignifiants, ne justifiant aucun effort particulier de présentation ou de promotion.

Pour éviter l'explosion combinatoire de paires de produits dans la table de faits panier de marché, nous utilisons un algorithme d'élagage progressif. Nous commençons au sommet de la hiérarchie des produits, que nous supposons être la catégorie. Nous énumérons d'abord toutes les combinaisons de panier entre catégories. S'il y a vingt-cinq catégories, cette première étape génère 625 lignes de panier de marché. Préalablement à une analyse plus fine, nous élaguons cette liste en ne conservant que les lignes ayant un comptage de paniers suffisamment élevé et où les unités et les montants de produits A et B (qui à ce stade sont des catégories) sont relativement équilibrés. Nous déterminons par expérimentation les seuils de comptage des paniers de marché et les limites à retenir pour les associations d'éléments A et B.

Nous passons ensuite au niveau suivant de détail, que nous supposons être la marque. Partant de l'ensemble des combinaisons élaguées de la première étape, nous faisons un forage vers le bas en énumérant toutes les combinaisons de marque (produit A) et de catégorie (produit B). De façon analogue, nous faisons un forage d'un niveau vers le bas en examinant toutes les combinaisons de marque (produit A) et de marque (produit B). Une fois encore, nous élaguons la liste pour ne conserver que les couples présentant les comptages de paniers les plus élevés et les montants et volumes les plus équilibrés. Nous faisons ensuite un forage vers le bas au niveau suivant de la hiérarchie.

En descendant dans la hiérarchie, nous obtenons des lignes comportant des comptages de paniers de plus en plus petits. Finalement, nous ne trouvons plus de comptages supérieurs au seuil retenu. On peut envisager de stopper le processus dès lors que la curiosité de l'analyste a été satisfaite. L'un des avantages de cette approche du haut en bas (en anglais *top down*) est que les lignes trouvées à chaque étape sont celles qui ont le plus d'intérêt. L'élagage progressif de la liste permet de voir de plus près les résultats antérieurs les plus significatifs. On peut envisager d'automatiser le processus, parcourant la hiérarchie produit vers le bas, éliminant les comptages trop faibles et sélectionnant les couples les mieux équilibrés en montant et en quantité. Le processus pourrait s'arrêter quand le nombre de couples de produits obtenu atteint le seuil désiré ou quand l'activité globale en termes de comptage de paniers, de montants et de quantités atteint une limite inférieure.

Une variante de cette approche pourrait commencer par une catégorie, une marque ou même un produit. Cette variante combinerait cet élément spécifique d'abord avec toutes les catégories, puis descendrait dans la hiérarchie. Un autre procédé consisterait à examiner la combinaison de produits achetés par un client donné pendant une période de temps, que ces produits soient ou non dans le même panier. Dans tous les cas, une grande partie du travail nécessaire doit avoir été effectuée par les processus ETC (extraction/transformation/chargement) de la zone préparation, pour simplifier le plus possible la phase de requête et de présentation de l'analyse.

Résumé

Ce chapitre a permis la construction d'un premier modèle dimensionnel. Quelle que soit la branche d'activité, nous recommandons fortement la mise en œuvre d'un processus en quatre étapes pour la conception d'un modèle dimensionnel. Souvenez-vous de l'importance du grain associé au schéma dimensionnel. Le chargement de la table de faits avec des données atomiques procure la plus grande souplesse, car ces données peuvent alors être récapitulées de toutes les manières possibles. Quand une table de faits se limite à des informations agrégées, nous nous heurtons à un mur si les hypothèses sur les besoins d'analyse, faites lors du choix des agrégats, s'avèrent inexactes. Souvenez-vous aussi qu'il est vital de renseigner les tables de dimension d'attributs descriptifs clairs, parlants, sous forme de texte.

Le chapitre suivant porte lui aussi sur le secteur de la distribution et montre comment un deuxième processus d'entreprise peut être traité au sein d'une même organisation en tirant pleinement partie de ce qui a été réalisé antérieurement, évitant ainsi le piège des applications isolées.