ENTREPÔTS DE DONNÉES

Guide pratique de modélisation dimensionnelle

2e édition

Ralph Kimball et Margy Ross

Traduction de Claude Raimond

Chapitre 4

Achats

Ce chapitre est consacré au processus des achats. Ce sujet concerne évidemment d'innombrables activités puisqu'il intervient dès lors que des produits ou services sont achetés pour être utilisés ou revendus. Nous développerons plusieurs modèles d'achat dans ce chapitre et nous présenterons des techniques pour traiter les changements d'attribut de nos tables de dimension. Bien que les attributs descriptifs des tables de dimension soient relativement stables, ils sont sujets à des variations sur la durée. Des lignes de produits sont restructurées, ce qui entraîne des modifications de la hiérarchie des produits. Des clients déménagent et leurs informations géographiques se modifient. Les commerciaux sont mutés ou promus, ce qui modifie les attributions de territoires. Nous évoquerons différentes approches pour gérer les inévitables changements affectant nos tables de dimension.

Le chapitre 4 traite des concepts suivants :

- *importance de la chaîne de valeur ;*
- *schémas de transaction mélangés par opposition aux schémas distincts ;*
- *techniques élémentaires et techniques avancées de traitement des dimensions à évolution lente.*

consiste à enregistrer le fait dans les deux unités de mesure pour qu'un état puisse glisser le long de la chaîne de valeur en sélectionnant des faits compatibles. Nous reviendrons sur les unités de mesure multiples au chapitre 5.

Résumé

Le stock est un processus qu'il est important de mesurer et de suivre dans de nombreuses industries. Nous avons développé dans ce chapitre des modèles dimensionnels pour les trois vues complémentaires du stock. Les modèles d'instantané périodique ou d'instantané récapitulatif fournissent une bonne description indépendante du stock. L'instantané périodique sera choisi pour des scénarios de stocks permanents, constamment renouvelés. L'instantané récapitulatif convient pour des produits stockés une seule fois et dont le séjour en stock a un début et une fin bien définis. Les applications de stock plus élaborées doivent compléter l'un de ces modèles ou les deux par le modèle des transactions.

Nous avons présenté des concepts clé relatifs à l'architecture et à la matrice de bus de l'entrepôt de données. Tout processus d'entreprise de la chaîne de valeur supporté par un système principal de données sources se traduit par un marché d'infos, ainsi que par une ligne dans la matrice de bus. Les marchés d'infos partagent un nombre important de dimensions standardisées et conformes. Le développement et le respect d'une architecture de bus est la condition *sine qua non* du succès d'un entrepôt de données comportant un ensemble intégré de marchés d'infos.

Chapitre 4

Achats

Ce chapitre est consacré au processus des achats. Ce sujet concerne évidemment d'innombrables activités puisqu'il intervient dès lors que des produits ou services sont achetés pour être utilisés ou revendus. Nous développerons plusieurs modèles d'achat dans ce chapitre et nous présenterons des techniques pour traiter les changements d'attribut de nos tables de dimension. Bien que les attributs descriptifs des tables de dimension soient relativement stables, ils sont sujets à des variations sur la durée. Des lignes de produits sont restructurées, ce qui entraîne des modifications de la hiérarchie des produits. Des clients déménagent et leurs informations géographiques se modifient. Les commerciaux sont mutés ou promus, ce qui modifie les attributions de territoires. Nous évoquerons différentes approches pour gérer les inévitables changements affectant nos tables de dimension.

Le chapitre 4 traite des concepts suivants :

- *importance de la chaîne de valeur ;*
- *schémas de transaction mélangés par opposition aux schémas distincts ;*
- *techniques élémentaires et techniques avancées de traitement des dimensions à évolution lente.*

4.1 Étude de cas des achats

Nous avons étudié jusqu'ici des processus de vente au détail et de stockage, situés dans la partie aval de la chaîne de valeur. Nous avons vu l'importance du développement de l'architecture de bus de l'entrepôt de données, avec des dimensions conformes communes à des tables de faits centrées sur les processus d'entreprise. Dans ce chapitre nous étendrons ces concepts en remontant vers l'amont de la chaîne de valeur où se trouve le processus d'achat.

Pour beaucoup d'entreprises, les achats sont une activité cruciale. L'achat efficace d'un produit au bon prix, en vue de sa revente, est évidemment important pour des distributeurs tels que notre chaîne de distribution alimentaire. Les achats ont aussi un effet majeur sur les résultats de toute grande organisation qui achète des produits en tant que matières premières servant à la fabrication. Des possibilités d'économie importantes sont associées à la réduction du nombre de fournisseurs et à la négociation d'accords avec les fournisseurs attitrés.

La prévision de la demande est à la base d'une gestion efficace des matières premières. Une fois la demande prévisionnelle établie, le but de la fonction achats est de déterminer les sources les plus économiques de matières ou de produits. Les achats mettent en jeu un nombre considérable de tâches allant de l'émission de demandes d'achat et de commande jusqu'au suivi des livraisons et à l'autorisation des paiements. La liste suivante donne une idée des besoins d'analyse courants d'un service achats :

- Quels matières ou produits sont achetés le plus souvent ? Combien de fournisseurs vendent ces produits ? À quels prix ? Dans quelles unités de mesure (par exemple en vrac ou en rouleaux) ?
- En tenant compte de la demande pour l'ensemble de l'entreprise (et non pour un seul établissement), y a-t-il une possibilité de négocier des conditions de prix favorables en réduisant le nombre de fournisseurs, en utilisant une source unique ou en garantissant des quantités achetées ?
- Nos collaborateurs achètent-ils aux fournisseurs attitrés de l'entreprise ou contournent-ils les accords négociés avec les fournisseurs (dépenses non conformes aux règles) ?
- Nos fournisseurs tiennent-ils les prix qui ont été négociés (variation par rapport aux prix contractuels) ?
- Quelles sont les performances de nos fournisseurs ? Quel est le taux de couverture des besoins par le fournisseur ? Son respect des délais de livraison ? Les livraisons tardives en attente ? Le pourcentage des commandes faisant l'objet d'un complément de commande ? Le taux de rejet sur la base des contrôles à la réception ?

4.2 Transactions d'achat

Suivant notre processus de conception en quatre étapes, nous commençons par choisir les achats comme processus d'entreprise à modéliser. Nous étudions ce processus d'entreprise en détail et nous remarquons un grand nombre de transactions d'achat, telles que les demandes d'achat, les commandes, les bons de livraison, les réceptions et les règlements. Comme pour les transactions de stock du chapitre 3, nous choisissons d'abord de construire une table de faits au grain d'une ligne par transaction. Nous identifions la date de transaction, le produit, le fournisseur, les conditions contractuelles et le type de transaction d'achat comme étant nos dimensions principales. Le nombre d'unités achetées et le montant de la transaction sont les faits. La conception obtenue est représentée à la figure 4.1.

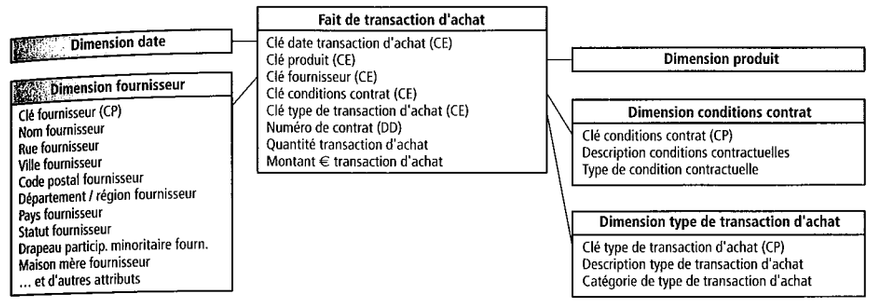

Figure 4.1 **Table de faits d'achat avec de multiples types de transaction**

Si nous travaillons toujours pour le même distributeur de produits alimentaires, les dimensions date de transaction et produit sont les mêmes dimensions conformes développées en premier lieu au chapitre 2. Si nous avons affaire à des achats dans un contexte de fabrication, les matières premières seront probablement situés dans une table de dimension matière première au lieu de figurer dans la table de dimension des produits destinés à être vendus. Les dimensions fournisseur et conditions contractuelles contiennent une ligne par jeu de conditions générales négociées avec un fournisseur, à la manière de la dimension promotion du chapitre 2. La dimension type de transaction d'achat nous permet de faire des groupements ou des filtrages sur des types de transaction, comme les commandes fournisseurs. Le numéro de contrat est une dimension dégénérée. Elle pourrait servir à déterminer le volume d'affaires traité dans le cadre de chaque contrat négocié.

Tables de faits mélangeant ou non les types de transaction

En discutant la conception initiale du schéma des achats avec les utilisateurs, nous prenons conscience de plusieurs détails nouveaux. Nous apprenons d'abord que les utilisateurs décrivent différemment les diverses transactions d'achat. Les com-

mandes, les bons de livraison, les réceptions et les règlements des fournisseurs sont perçus comme des processus distincts.

Il s'avère que plusieurs de ces transactions d'achat viennent en effet de différentes applications sources. Il y a une application pour les demandes d'achat et les commandes fournisseur, une autre pour le traitement des bons de livraison et les réceptions et une application de comptabilité fournisseur qui traite les règlements des fournisseurs.

Nous découvrons aussi que plusieurs types de transaction ont des dimensions différentes. Par exemple, les déductions s'appliquent aux paiements des fournisseurs mais non aux autres types de transaction. De même, le nom du réceptionnaire n'a de sens que pour les réceptions de produits dans l'entrepôt.

Nous découvrons aussi toute une série de numéros d'identification intéressants, comme des numéros de commande et des numéros de chèque de règlement, qui sont créés à différents stades du processus d'achat. Ces numéros d'identification sont de parfaits candidats au statut de dimension dégénérée. Pour certains types de transaction, il peut y avoir plusieurs numéros d'identification.

Tout en classant ces nouveaux détails, nous devons prendre une décision affectant la conception du modèle. Devons-nous construire une table de faits de transaction avec une dimension type de transaction pour voir nos transactions ensemble ou devons-nous construire une table de faits séparée pour chaque type de transaction ? C'est là un dilemme fréquent, que l'on rencontre à propos d'autres transactions que les transactions d'achat.

Notre décision doit être basée sur une bonne compréhension des besoins de l'entreprise, confrontée aux avantages de chacune des options de conception. En l'occurrence, il n'y a pas de formule simple pour déterminer si l'on doit utiliser une table de faits unique ou plusieurs. Pour faire un choix, nous devons prendre en compte différentes considérations :

- En premier lieu, quels sont les besoins d'analyse des utilisateurs ? Notre objectif, en tant que concepteurs, est de réduire la complexité et de présenter les données sous la forme qui convient le mieux aux utilisateurs. Quelles seront les analyses les plus courantes faites par les utilisateurs ? Si les analyses font souvent appel à différents types de transaction, nous tendrons à envisager une table unique et si elle portent le plus souvent sur un seul type de transaction, nous tendrons à prévoir des tables séparées pour chaque type de transaction.
- Y a-t-il vraiment de multiples processus d'entreprise distincts ? Dans notre exemple des achats, il semble que l'achat des produits (les commandes) soit vraiment différent de la réception. L'existence de numéros d'ordre distincts pour chaque étape du processus nous suggère que nous avons affaire à des processus distincts. Nous envisagerons alors des tables de faits séparées.

Dans l'exemple de stock du chapitre 3, toutes les transactions mises en jeu se rapportaient clairement à un processus de stock unique, ce qui nous a orienté vers une conception ne comportant qu'une seule table de faits.

- Les données proviennent-elles de plusieurs applications sources différentes ? Dans notre exemple, nous avons affaire à trois applications sources distinctes, les achats, l'entrepôt et la comptabilité fournisseurs. Cette situation nous porte à choisir des tables de faits séparées. La préparation d'une table de faits unique à partir de trois systèmes sources différents sera sans doute une tâche ardue.
- Quelles sont les dimensions des faits ? Dans notre exemple d'achat nous avons découvert plusieurs dimensions qui s'appliquent à certains types de transactions mais non à d'autres, ce qui nous pousse également à envisager des tables de faits séparées.

Dans notre étude de cas, nous décidons de créer de multiples tables de faits, comme le montre la figure 4.2. Nous avons des tables de faits distinctes pour les demandes d'achats, les commandes, les bons de livraisons, les réceptions et les règlements fournisseurs. Nous somme arrivés à cette décision parce que les utilisateurs perçoivent ces activités comme des processus d'entreprise distincts, parce que les données proviennent de différentes applications sources et parce qu'il y a des dimensions spécifiques pour divers types de transactions. Des tables de faits multiples permettent des dimensions et des attributs plus riches et plus descriptifs. En cheminant depuis les demandes d'achat jusqu'aux règlements des fournisseurs, nous héritons des dimensions date et des dimensions dégénérées des étapes précédentes. L'approche de la table de faits unique aurait obligé à généraliser l'appellation de certaines dimensions. Par exemple, la date de commande et la date de réception auraient été appelées date de transaction. De même, l'acheteur et le réceptionnaire seraient devenus des employés. Dans une autre organisation, différente par les besoins de sa gestion, ses applications sources et la nature des dimensions, la table de faits unique mélangeant les types de transactions aurait pu être préférable.

Nous nous rendons compte que des tables de faits multiples peuvent demander plus de temps pour être créées et gérées, parce qu'il faut charger, indexer et agréger un plus grand nombre de tables. Certains prétendront que cette approche augmente la complexité des travaux de préparation. En fait, les travaux de préparation peuvent se trouver simplifiés. Les données sources se trouvant dans différentes applications opérationnelles, nous avons besoin de plusieurs travaux de préparation dans chacun des deux scénarios de table de faits. Le chargement des données dans des tables de faits séparées sera probablement moins compliqué que l'intégration en une seule table de faits de données de provenances diverses.

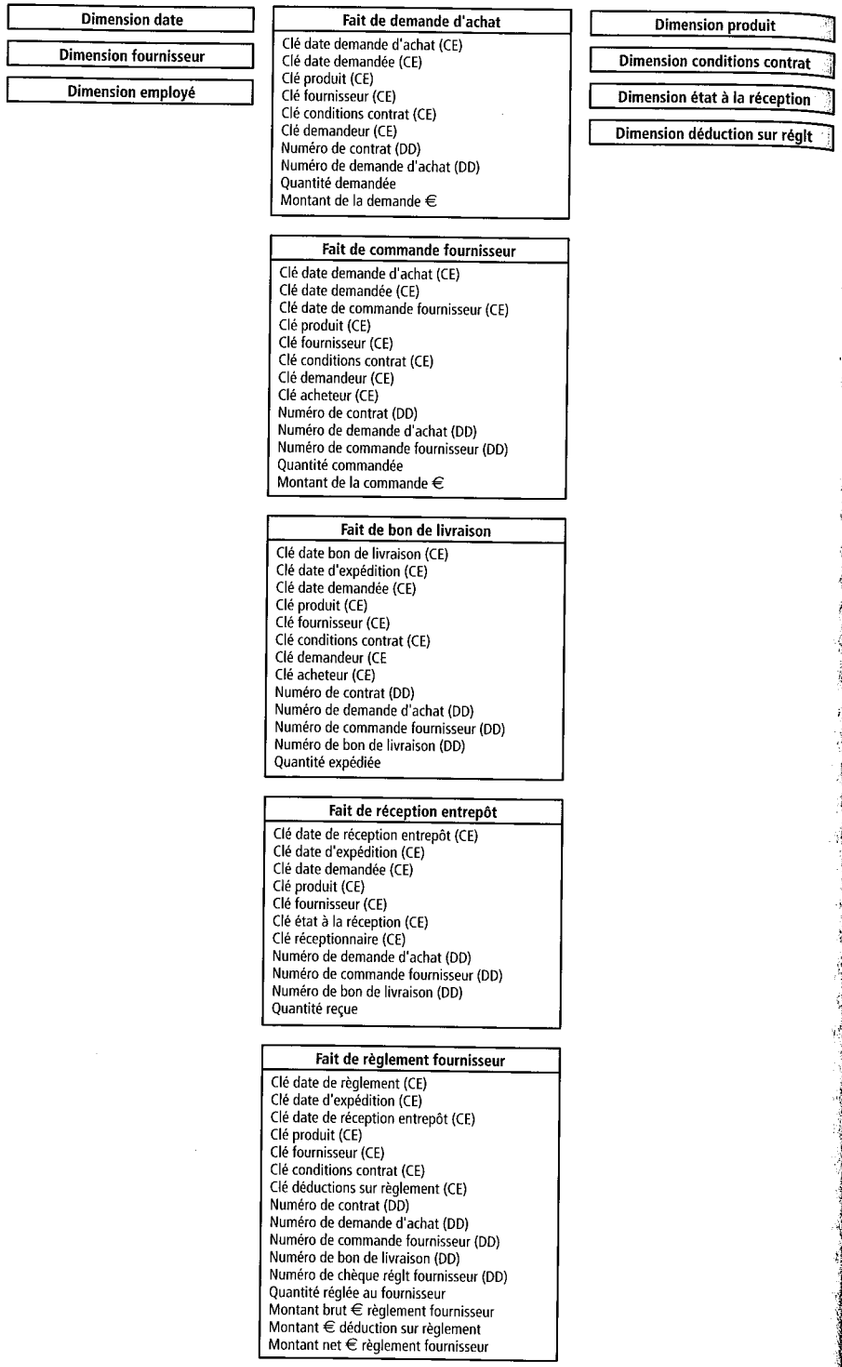

Figure 4.2 **Multiples tables de faits pour les processus d'achat**

Instantané complémentaire d'achat

Indépendamment de la décision concernant les tables de faits d'achat, nous pouvons avoir besoin de prévoir une sorte de table d'instantanés pour répondre pleinement aux besoins de la gestion. Comme nous l'avons vu au chapitre 3, un instantané récapitulatif recouvrant les divers processus peut être extrêmement utile si les gestionnaires souhaitent surveiller les mouvements des produits à travers la filière des achats (et mesurer la durée ou les décalages de temps à chaque stade). Nous consacrerons davantage de temps à ce thème au chapitre 5.

4.3 Dimensions à évolution lente

Nous avons fait jusqu'ici comme si chaque dimension était indépendante de toutes les autres. En particulier, nous avons supposé que les dimensions étaient indépendantes du temps. Ce n'est malheureusement pas le cas dans le monde réel. Les attributs des tables de dimension sont relativement stables, mais ils ne sont pas immuables. Ils se modifient, lentement, avec le temps. Un dialogue actif avec les gestionnaires peut seul permettre aux concepteurs de déterminer la meilleure façon de traiter les dimensions à évolution lente. Nous n'avons pas le droit de conclure prématurément que les gestionnaires se moquent des changements de dimension simplement parce qu'ils n'en ont pas parlé en décrivant leurs besoins. Nous pourrions penser que le suivi des changements est superflu, alors que les utilisateurs supposent que l'entrepôt de données leur permettra de mesurer avec précision l'incidence de chacun des changements. Il peut être tentant d'éviter par notre silence les travaux de développement supplémentaires liés au suivi des changements, mais il est évidemment préférable de recevoir le message d'emblée.

Lorsque le suivi des changements s'avère nécessaire, nous ne devons pas tout mettre dans la table de faits ni rendre chaque dimension dépendante du temps. Cette approche nous replongerait dans une structure normalisée complète, difficile à comprendre et entraînant une moindre performance des requêtes. Il vaut mieux tirer parti du fait que la plupart des dimensions sont presque constantes dans le temps. Nous pouvons préserver la structure dimensionnelle indépendante au moyen de quelques ajustements relativement mineurs pour prendre en compte les modifications. Nous appelons ces dimensions presque constantes des *dimensions à évolution lente*. Depuis la présentation de ce concept des *slowly changing dimensions* par Ralph Kimball en 1994, certains informaticiens américains les appellent des SCD.

Pour chacun des attributs de nos tables de dimension, nous devons spécifier une approche de gestion du changement, c'est-à-dire une réaction à toute modification d'une valeur d'attribut dans le monde opérationnel. Dans la section suivante, nous décrivons trois techniques de base pour le traitement des changements d'attributs, ainsi que deux approches hybrides. Vous pouvez avoir besoin d'utiliser une combinaison de ces techniques au sein d'une même table de dimension.

Type 1 : écrasement de la valeur précédente

Dans cette réaction de type 1, nous remplaçons simplement l'ancienne valeur de l'attribut par la nouvelle dans la ligne de dimension concernée. De ce fait, l'attribut a toujours la valeur la plus récente.

Supposons que nous travaillons pour un distributeur de produits électroniques. Les acheteurs sont classés de la même manière que les rayons des magasins de vente et les produits achetés peuvent être cumulés selon les rayons. L'un des produits achetés est le logiciel IntelliKidz. Voici la ligne de la table de dimension produit pour IntelliKidz :

Clé produit	Description du produit	Rayon	Numéro US (clé naturelle)
12345	IntelliKidz 1.0	Éducation	ABC922-Z

Il y a bien sûr beaucoup d'autres attributs descriptifs dans la dimension produit, mais nous avons abrégé la liste des colonnes pour qu'elles tiennent dans la page de cet ouvrage. Nous avons adopté une clé artificielle produit comme clé primaire de la table au lieu de prendre le code US (unité de stock). Le code US a été rabaissé au rang d'attribut ordinaire du produit, mais il a toujours une signification spéciale, étant une clé naturelle. Contrairement à tous les autres attributs du produit, la clé naturelle doit rester inviolée. Dans toute notre discussion sur les trois types de dimension à évolution lente, nous supposerons que la clé naturelle d'une dimension demeure constante.

Supposons qu'une personne nouvelle s'occupant de la vente des produits décide qu'IntelliKids doit être déplacé le 15 janvier 2002 du rayon Logiciel éducatif vers le rayon Stratégie, dans le but de développer les ventes. Avec l'approche du type 1, nous mettons simplement à jour la ligne existante de la table de dimension en remplaçant l'indication du rayon. La ligne en question est désormais :

Clé produit	Description du produit	Rayon	Numéro US (clé naturelle)
12345	IntelliKidz 1.0	Stratégie	ABC922-Z

Dans ce cas, aucune clé de la table de dimension ou de la table de faits n'a été modifiée à la suite du changement de rayon d'IntelliKidz. Les lignes de la table de faits ont toujours 12345 comme clé produit, indépendamment du rayon où se trouve IntelliKidz. Lorsque les ventes s'accélèrent à la suite du déplacement dans le rayon Stratégie, nous n'avons aucune information expliquant l'amélioration des performances, parce les données historiques comme les données récemment chargées se présentent comme si elles avaient toujours fait partie du rayon Stratégie.

La solution de type 1 est l'approche la plus simple pour le traitement des changements d'attributs de dimension. Ses avantages sont la rapidité et la simplicité. Nous écrasons simplement la valeur existante dans la table de dimension par la nouvelle valeur. La table de faits ne change pas. Le problème de cette solution est que nous perdons tout l'historique des changements d'attribut. Les valeurs antérieures des attributs sont effacées, nous ne connaissons plus que les valeurs actuelles. Une solution de type 1 est évidemment la meilleure si la modification d'attribut est une correction. Elle peut aussi convenir s'il n'y a aucun intérêt à conserver l'ancienne description. Nous avons besoin de l'opinion des utilisateurs pour savoir s'il y a intérêt à conserver l'ancienne valeur de l'attribut modifié ; choisir unilatéralement cette solution pour sa simplicité technique risque de priver les utilisateurs du suivi rigoureux des évolutions dont ils peuvent avoir besoin.

La solution de type 1 est facile à mettre en œuvre, mais elle ne conserve aucune trace des valeurs antérieures des attributs.

Avant de quitter la solution de type 1, signalons un piège qui pourrait passer inaperçu. Si nous choisissons cette solution pour le reclassement d'IntelliKidz, toutes les agrégations antérieures basées sur la valeur du rayon doivent être reconstruites. Les données agrégées doivent continuer de correspondre aux données atomiques dont elles se composent et donc les agrégats doivent être mis à jour pour faire comme si IntelliKidz avait toujours fait partie du rayon Stratégie.

Type 2 : ajout d'une ligne de dimension

Plus haut dans cet ouvrage, nous avons affirmé que l'un des buts essentiels de l'entrepôt de données était de représenter correctement les données antérieures. La solution de type 2 est la principale technique pour répondre à cette exigence dans le cas des dimensions à évolution lente.

Avec l'approche de type 2, lorsque le rayon d'IntelliKidz est modifié, nous inscrivons une nouvelle ligne de dimension produit indiquant la nouvelle valeur de l'attribut rayon. Nous avons alors deux lignes de dimension produit pour IntelliKidz, comme ceci :

Clé produit	Description du produit	Rayon	Numéro US (clé naturelle)
12345	IntelliKidz 1.0	Éducation	ABC922-Z
25984	IntelliKidz 1.0	Stratégie	ABC922-Z

Nous voyons maintenant pourquoi la clé de dimension produit ne peut pas être la clé naturelle Numéro US. Nous avons besoin de deux numéros de clé artificielle pour la même US (unité de stock) ou le même code barre physique. Chacune de ces

deux clés identifie un profil unique d'attributs du produit valable sur une certaine durée. Avec les modifications de type 2, la table de faits reste toujours inchangée. Nous n'avons pas à revenir aux lignes antérieures de la table de faits pour modifier la clé produit. Dans la table de faits, les lignes concernant IntelliKidz avant le 15 janvier 2002 ont comme clé produit 12345 et peuvent se cumuler au niveau du rayon Éducation. Après le 15 janvier, les lignes concernant IntelliKids ont comme clé produit 25984 et refléteront le placement du produit dans le rayon Stratégie jusqu'à ce que nous ayons fait un autre changement du type 2. C'est pourquoi nous disons que la solution de type 2 segmente parfaitement l'historique pour tenir compte de la modification.

Une contrainte sur le seul attribut rayon fera très exactement la différence entre les deux profils de produit. Si nous appliquons seulement une contrainte sur la description du produit, c'est-à-dire sur IntelliKidz 1.0, la requête ira automatiquement chercher les deux lignes de la dimension produit pour IntelliKidz 1.0 et fera automatiquement les jointures à la table de faits pour obtenir l'historique complet du produit. Pour compter correctement le nombre de produits, nous utiliserons seulement l'attribut clé naturelle US (unité de stock) comme base du comptage et non la clé artificielle. Le champ clé naturelle devient une sorte de colle servant à raccorder en toute sécurité les deux enregistrements de type 2 relatifs à un même produit. Un attribut indicateur de ligne plus récente peut être ajouté pour permettre aux utilisateurs d'appliquer plus vite des contraintes sur les seuls profils actuels.

La solution de type 2 est la principale technique permettant de suivre correctement les attributs des dimensions à évolution lente. Elle est très puissante, la ligne de dimension supplémentaire permettant de segmenter la table de faits en fonction de l'historique.

Il semble tout à fait naturel d'inclure une date d'effet ou une date d'expiration sur une ligne de changement de type 2. Ces attributs sont nécessaires dans la zone de préparation des données parce que nous avons besoin de savoir laquelle des clés artificielles utiliser quand nous chargeons des enregistrements de faits antérieurs. Dans la table de dimension, ces dates sont des suppléments intéressants mais non nécessaires à la segmentation basique de l'historique. Si vous les utilisez, souvenez-vous qu'il n'y a pas besoin d'établir de contrainte sur la date d'effet située dans la table de dimension pour obtenir la bonne réponse. Ceci est souvent une source de confusion lors de la conception et de l'utilisation des dimensions à évolution lente.

Les concepteurs de bases de données peuvent trouver logique de prévoir des attributs date d'effet et date d'expiration, mais nous devons être conscients que la date d'effet d'une ligne de la table de dimension n'a pas grand chose à voir avec les dates de la table de faits. Une tentative de contrainte sur la ligne date d'effet de la table de dimension peut en fait produire un résultat incorrect. Il se peut que la ver-

sion 2.0 du logiciel IntelliKidz soit diffusée le 1er mai 2002. Un nouveau code opérationnel US (et une nouvelle clé artificielle) seront créés pour ce nouveau produit. Il ne s'agit pas d'une modification de type 2 parce que le produit est une entité physique entièrement différente. Mais si nous regardons dans une table de faits du distributeur, nous n'apercevons pas une segmentation nette de l'historique. L'ancienne version 1.0 du logiciel continuera inévitablement d'être vendue dans les magasins au-delà du 1er mai 2002, jusqu'à épuisement des stocks. La nouvelle version 2.0 apparaîtra sur les étagères le 1er mai et remplacera progressivement l'ancienne version. Il y aura une période de transition où les deux versions du logiciel passeront devant les caisses enregistreuses des différents magasins. Naturellement, la période de recouvrement entre les deux produits variera d'un magasin à l'autre. Les caisses enregistreuses reconnaîtront les deux codes opérationnels US et n'auront pas de difficulté à traiter les ventes de l'une ou l'autre version. Si nous avions une date d'effet sur la ligne de dimension produit, nous n'oserions pas établir de contrainte sur cette date pour segmenter les ventes parce que la date ne cadre pas avec les réalités. Pis encore, une contrainte sur cette date peut même donner un résultat faux.

Toutefois, les dates d'effet/expiration de la ligne dans la table de dimension peuvent servir dans le cas d'analyses avancées. Les dates permettent un découpage très précis de la dimension elle-même en fonction du temps. La ligne comportant la date d'effet indique le premier jour où le profil descriptif est valide. La date d'expiration est soit le jour qui précède la date d'effet du classement suivant, soit la date de suppression du produit du catalogue. Nous pourrions déterminer ce à quoi ressemblait le catalogue au 31 décembre 2001 au moyen d'une requête avec une contrainte sur la table produit donnant toutes les lignes telles que la date d'effet de la ligne soit inférieure ou égale au 31 décembre 2001 et telles que la date d'expiration de la ligne soit supérieure ou égale au 31 décembre 2001. Nous verrons d'autres façons d'exploiter les dates d'effet et d'expiration à propos du schéma des ressources humaines au chapitre 8.

La solution de type 2 est la technique de prédilection pour les analyses basées sur des attributs reflétant correctement l'historique. La raison pour laquelle elle permet de segmenter correctement le passé dans la table de faits est que les lignes antérieures à une modification utilisent une clé artificielle antérieure à cette modification. Un autre avantage du type 2 est qu'il nous permet de suivre facilement plusieurs modifications successives pour un même produit. Contrairement à la solution 1, elle n'impose pas de revoir les tables d'agrégats antérieures aux modifications.

La solution de type 2 pour les dimensions à évolution lente impose évidemment l'usage de clés artificielles, recommandé en tout état de cause. On ne peut se contenter d'utiliser la clé opérationnelle sous-jacente même avec deux ou trois positions de version, qui entraîne tous les inconvénients mentionnés au chapitre 2. Il

n'est pas non plus souhaitable d'adjoindre une date d'effet à la clé primaire de la table de dimension pour identifier chaque version de manière unique. La solution de type 2 crée une nouvelle ligne de dimension avec une nouvelle clé primaire identifiant de manière unique le nouveau profil du produit. Cette clé primaire établit le lien entre les tables de faits et les tables de dimension pour un ensemble donné de caractéristiques du produit. Il n'y a pas besoin d'envisager une jointure secondaire basée sur des dates d'effet ou d'expiration, comme nous l'avons déjà souligné.

Certains d'entre vous peuvent s'inquiéter de la gestion des clés artificielles supportant les modifications dans la solution de type 2. Nous verrons au chapitre 16 un organigramme du traitement des clés artificielles intégrant une gestion détaillée des changements dans la solution de type 2. En attendant, nous voulons vous rassurer quant au fardeau administratif correspondant. Au stade de la préparation des tables de dimension, nous récupérons le plus souvent une copie intégrale de la source de données complète à jour. Ce serait bien pratique si nous recevions simplement les modifications, c'est-à-dire le delta par rapport au dernier extrait, mais en général l'application de préparation des données doit trouver les dimensions modifiées. Une comparaison champ par champ de chaque ligne de dimension avec la ligne de dimension de la veille serait extrêmement laborieuse, notamment si nous avons cent attributs dans une table de dimension de plusieurs millions de lignes. Au lieu de vérifier chaque champ pour voir s'il a changé, nous calculons d'un coup un total de contrôle pour toute la ligne. Un algorithme de type CRC (*Cyclic Redundancy Checksum*, contrôle par redondance cyclique) nous permet de voir immédiatement qu'une ligne a été modifiée, sans recourir à une comparaison des nombreux champs qu'elle contient. Dans notre zone de préparation, nous calculons le CRC pour chaque ligne de la table de dimension et nous le plaçons dans une colonne d'administration. Au prochain chargement de données, nous calculons les CRC des nouveaux enregistrements et les comparons aux CRC antérieurs. S'ils sont égaux, tous les attributs des deux lignes sont identiques ; il est inutile de vérifier chaque champ. Évidemment, toute nouvelle ligne de la source de données déclenche la création d'une nouvelle ligne de dimension. Enfin, quand nous rencontrons un CRC modifié, nous traitons la modification en accord avec la solution choisie. Dans le cas de la solution de type 2, nous créons une nouvelle ligne. Si nous utilisons une combinaison de techniques, nous devons examiner les champs de façon plus précise pour déterminer l'action appropriée.

Comme la solution de type 2 donne naissance à de nouvelles lignes de dimension, elle présente l'inconvénient de faire croître le volume de la table. C'est la raison pour laquelle cette technique peut ne pas convenir pour des tables ayant déjà plus d'un million de lignes. Nous verrons une autre possibilité de traitement des changements dans de grandes tables de dimension comportant des millions de lignes en examinant la dimension client au chapitre 6.

Type 3 : ajout d'une colonne de dimension

Si la solution de type 2 permet de segmenter l'historique, elle ne permet pas d'associer l'ancienne valeur de l'attribut aux faits antérieurs et vice versa. Avec la solution de type 2, quand nous posons une contrainte sur Rayon = Stratégie, nous ne voyons pas les faits IntelliKidz antérieurs au 15 janvier 2002. Dans la plupart des cas, c'est précisément ce que nous voulons.

Cependant, il arrive que nous voulions voir des données de faits comme si le changement ne s'était pas produit. Cette situation se produit le plus souvent en cas de réorganisation des forces de vente. Les limites des zones ont été redessinées, mais certains utilisateurs veulent voir les ventes d'aujourd'hui dans le cadre des zones d'hier pour voir ce qu'elles auraient donné dans l'ancienne configuration. Pendant les quelques mois de la transition, il peut exister une volonté de suivre l'historique en utilisant les nouveaux noms de zone ou de suivre les nouvelles données en utilisant les anciens noms de zone. Une solution de type 2 ne répond pas à une telle demande, mais la solution de type 3 vient à notre secours.

Dans notre exemple de logiciel, nous supposerons qu'il existe un besoin légitime de suivre à la fois les nouvelles et les anciennes valeurs de l'attribut rayon vers l'amont et vers l'aval pendant une période de temps voisine de la date du changement. Avec la solution de type 3, nous ne créons pas de ligne supplémentaire, mais nous ajoutons une nouvelle colonne pour refléter le changement d'attribut. Dans le cas d'IntelliKidz, nous modifions la table de dimension produit et lui adjoignons un attribut rayon antérieur. Nous renseignons cette nouvelle colonne avec la valeur de rayon existante (Éducation). Nous traitons ensuite l'attribut rayon comme dans le cas de la solution 1, où nous avions écrit le nouveau rayon à la place de l'ancien (Stratégie). Tous les états et toutes les requêtes existantes commutent immédiatement sur la nouvelle description du rayon, mais nous pouvons toujours faire des requêtes sur l'ancienne valeur du rayon en utilisant l'attribut rayon antérieur.

Clé produit	Description du produit	Rayon	Rayon antérieur	Numéro US (clé naturelle)
25984	IntelliKidz 1.0	Stratégie	Éducation	ABC922-Z

La solution 3 convient lorsqu'il y a un besoin impérieux de permettre deux visions du monde simultanément. Certains concepteurs parlent alors de *réalité alternée*. C'est souvent le cas lorsque le changement ou la redéfinition se fait en douceur ou bien s'il s'agit d'un changement d'appellation plutôt que du changement d'une caractéristique physique. Bien que le changement se soit produit, il est toujours logiquement possible de faire comme s'il n'avait pas eu lieu. Cette solution de type 3 se distingue de la solution de type 2 en ce que la description actuelle et la description antérieure peuvent être considérées comme vraies en même temps.

Dans le cas d'une réorganisation des ventes, le management peut vouloir être en mesure de prévoir un recouvrement et d'analyser les résultats pendant un certain temps au moyen des deux cartes de territoire. On rencontre couramment une autre variante dans laquelle les utilisateurs veulent voir la valeur actuelle et en plus conserver la valeur d'origine, non la valeur précédente.

La solution de type 3 est assez peu utilisée. N'allez pas penser que son numéro plus élevé souligne une préférence. Ces techniques n'ont pas été présentées dans un ordre reflétant un classement selon un critère de qualité. Chacune d'elle est la meilleure réponse à certaines situations.

La solution de type 3 au problème des dimensions à évolution lente nous permet de voir les données antérieures à la fois selon les nouvelles et les anciennes valeurs d'attribut.

Une solution de type 3 est inadaptée à une situation où vous devez suivre l'incidence de nombreuses valeurs d'attribut intermédiaires. Elle entraîne de sérieuses complications pour la réalisation comme pour l'utilisation si elle doit prendre en compte des attributs reflétant l'état du monde actuel, l'état antérieur, l'état antérieur –1, l'état antérieur –2, l'état antérieur –3 et il faut alors abandonner la possibilité d'analyser ces valeurs intermédiaires. S'il existe un besoin de suivre une myriade de modifications imprévisibles, une solution de type 2 sera préférable dans la plupart des cas.

4.4 Techniques hybrides de traitement des dimensions à évolution lente

Dans cette section, nous allons voir des approches hybrides combinant les techniques de base du traitement des dimensions à évolution lente. De nombreux informaticiens ne jurent que par ces techniques parce qu'elles donnent l'impression de gagner sur tous les tableaux. Toutefois, nous payons cette plus grande souplesse est d'une plus grande complexité. La souplesse et l'élégance d'une solution peut séduire des informaticiens, mais sa complexité peut rebuter les utilisateurs auxquels on la destine. N'envisagez ces options que si les utilisateurs confirment qu'elles répondent à leurs besoins.

Changements prévisibles et application aux données de multiples versions des attributs modifiés

Cette technique est très fréquemment utilisée pour traiter des modifications de la structure commerciale, aussi laisserons-nous de côté notre exemple IntelliKidz pour présenter le concept à travers un scénario plus réaliste. Prenez la situation d'une structure commerciale qui revoit la carte de ses zones de vente sur une base annuelle. Sur une période de cinq ans, les zones seront réorganisées cinq fois. À

première vue, ce scénario semble un bon candidat pour la solution de type 2, mais nous découvrons au cours des interviews des utilisateurs qu'ils ont un ensemble d'exigences plus complexes, y compris les demandes suivantes :

- Préparer des états sur les ventes de chaque année en utilisant la carte des zones de l'année en question.
- Préparer des états sur les ventes de chaque année sur la base des zones d'une quelconque autre année.
- Préparer des états sur un nombre d'années arbitraire en utilisant une seule carte des zones choisie parmi celles des différentes années. La version la plus courante de cette demande concerne des états dans lesquels les données de toutes les années considérées utilisent la carte des zones actuelle.

Nous ne pouvons pas répondre à toutes ces demandes avec une solution standard de type 2 parce qu'elle segmente le passé : une année de données de faits ne peut entrer dans un état qu'en utilisant la carte affectée à l'année en question. Les demandes ne peuvent pas non plus être satisfaites par la solution de type 3 standard parce que nous voulons avoir plus de deux cartes en même temps.

Dans le cas d'espèce, nous tirons parti du caractère régulier et prévisible de ces changements en généralisant l'approche de type 3 pour avoir cinq versions de l'attribut zone pour chaque commercial. La dimension commercial comporte les attributs indiqués à la figure 4.3.

Dimension commercial
Clé commercial
Nom commercial
Adresse commercial...
Zone actuelle
Zone 2001
Zone 2000
Zone 1999
Zone 1998
... et la suite

Figure 4.3 **Exemple de table de dimension avec de multiples versions successives**

Chaque ligne de la dimension commercial contient toutes les zones auxquelles l'intéressé a été affecté antérieurement. L'utilisateur peut choisir de cumuler les faits de vente au moyen de n'importe laquelle des cinq cartes de zones. Si un commercial a été embauché en 2000, les attributs de dimension pour 1998 et 1999 contiendront des valeurs telles que « non applicable ».

Nous appelons l'affectation la plus récente « Zone actuelle ». Cet attribut sera très fréquemment utilisé ; nous ne voulons pas avoir à modifier nos requêtes et états existants pour les adapter au changement d'année. À la prochaine modification des zones, nous modifierons la table pour lui ajouter un attribut zone 2002. Nous placerons dans cette colonne les valeurs de zone actuelle et nous remplacerons l'attribut Zone actuelle par les affectations applicables en 2003.

Changements imprévisibles avec application aux données antérieures de la version actuelle de l'attribut modifié

Cette dernière technique convient s'il faut préserver la vision exacte du passé dans le contexte de modifications d'attribut imprévisibles tout en ayant le moyen de présenter les données antérieures selon les valeurs actuelles de l'attribut modifié. Aucune des trois techniques standard ne satisfait à elle seule cette double exigence.

Dans le cas de la dimension produit du distributeur de produits électroniques, notre nouvelle solution hybride prévoit deux attributs de rayon sur chaque ligne. La colonne Rayon actuel représente l'affectation actuelle, la colonne Rayon ancien représente la valeur précédente de l'attribut rayon.

Lorsque IntelliKidz est mis en vente pour la première fois, la dimension produit se présente comme suit :

Clé produit	Description du produit	Rayon actuel	Rayon antérieur	Numéro US (clé naturelle)
12345	IntelliKidz 1.0	Éducation	Éducation	ABC922-Z

Lorsque les rayons sont réorganisés et que IntelliKidz est placé dans le rayon Stratégie, nous utilisons la technique de la solution 2 pour représenter ce changement d'attribut en ajoutant une nouvelle ligne. Dans cette nouvelle ligne de dimension, le rayon actuel est identique au rayon antérieur. L'attribut Rayon actuel est mis à jour sur toutes les autres lignes de dimension pour IntelliKidz. Les deux lignes que nous avons à ce stade identifient le rayon Stratégie comme étant le rayon actuel.

Clé produit	Description du produit	Rayon actuel	Rayon antérieur	Numéro US (clé naturelle)
12345	IntelliKidz 1.0	Stratégie	Éducation	ABC922-Z
25984	IntelliKidz 1.0	Stratégie	Stratégie	ABC922-Z

Nous pouvons ainsi utiliser la valeur antérieure pour segmenter le passé et voir les données antérieures selon les rayons où se trouvaient alors les produits. Mais nous pouvons aussi cumuler les faits des clés produit 12345 et 25984 en utilisant l'affectation au rayon actuel. Au cas où IntelliKidz serait ensuite placé dans le rayon du logiciel pour le développement du sens critique, notre table de dimension produit deviendrait :

Clé produit	Description du produit	Rayon actuel	Rayon antérieur	Numéro US (clé naturelle)
12345	IntelliKidz 1.0	Logiciel éducatif	Éducation	ABC922-Z
25984	IntelliKidz 1.0	Logiciel éducatif	Stratégie	ABC922-Z
31726	IntelliKidz 1.0	Logiciel éducatif	Logiciel éducatif	ABC922-Z

Notre approche hybride crée une nouvelle ligne pour représenter le changement (type 2), ajoute une nouvelle colonne pour suivre l'affectation actuelle (type 3) et traite les changements ultérieurs selon la technique du type 1. On a suggéré de dire que cette approche combinée est du type 6 (2 + 3 + 1). Elle nous permet de suivre correctement les modifications antérieures et de voir le passé sur la base des affectations actuelles. Nous pourrions continuer d'enrichir (et de compliquer) cette approche en intégrant des structures supplémentaires d'organisation des rayons comme des attributs distincts, en plus du rayon actuel.

Cette technique puissante peut faire la joie de certains lecteurs, mais rappelons que nous devons garder à l'esprit le point de vue de l'utilisateur et nous efforcer de trouver le meilleur compromis entre la souplesse et la complexité.

4.5 Dimensions à évolution plus rapide

Ce chapitre a porté sur le traitement de modifications relativement peu fréquentes affectant les tables de dimension. Mais que se passe-t-il quand l'évolution s'accélère ? Si un attribut de dimension change tous les mois, nous n'avons plus affaire à une évolution susceptible d'être traitée raisonnablement par les techniques qui viennent d'être décrites. Une solution efficace du problème posé par ces changements fréquents consiste à isoler ces attributs qui changent rapidement et à les placer dans une ou plusieurs dimensions distinctes. Nous aurons alors deux clés étrangères dans la table de faits — une pour la table de dimension primaire et une autre pour l'attribut ou les attributs à évolution rapide. Ces dimensions sont associées les unes aux autres à chaque fois que nous plaçons une ligne dans une table de faits. Nous reviendrons sur ce sujet à propos des dimensions client, au chapitre 6.

Résumé

Nous avons vu dans ce chapitre différentes approches pour le traitement des données de stock. Une gestion performante des stocks peut avoir une incidence très positive sur les résultats d'une organisation.

Nous avons aussi présenté diverses techniques pour traiter les modifications des attributs des tables de dimension. Les solutions en cas de changements relativement peu fréquents des attributs sont leur remplacement pur et simple (type 1), l'adjonction d'une ligne supplémentaire à la table de dimension (type 2); une solution moins fréquente consistant à ajouter une colonne à la table de dimension (type 3). Nous avons aussi évoqué plusieurs solutions hybrides puissantes, bien que plus complexes, combinant les techniques de base.