データ ウェアハウス構築における データ モデリングの利点

2008 年 12 月

目次

概要 1

セクション 1 2

はじめに 2

セクション 2 2

データ ウェアハウスを設計する 2

ファクト テーブル 2

ディメンション 2

セクション 3 5

抽出、変換、および読み込み（ETL） 5

セクション 4 6

データ ウェアハウス構築のベスト プラクティス - データ モデリングの重要性 6

ビジネス要件を収集する 6

データベースのパフォーマンスを最適化する 7

ソース システムとターゲット システムの情報を提供する 7

セクション 5 7

まとめ 7

概要

課題

組織は今や膨大な量のデータを保有しています。蓄積されたデータにはビジネスに役立つ情報が含まれていますが、この情報を収集してレポートにまとめるのは非常に困難な作業です。次のような課題に対処する必要があります。

- データを探索、収集、および変換して、単一のソース レコードに保存する
- 保存されたデータが、ビジネス レポートの作成に役立つ正確な情報であることを保証する
- 大量のデータから高速に検索できる形式で履歴データを保存する

成果

組織が保有するデータから有益な情報を取り出し、ビジネス ユーザーに提供することで、次のような成果が期待できます。

- ビジネス ユーザーは必要な情報を入手し、その情報に基づいて意思決定を下すことができる
- 高速なクエリー処理によってデータへのアクセスが容易になり、履歴データから一定のパターンや傾向を見つけ出すことができる
- 個人顧客の購入から国際企業の総売り上げに至るまで、任意のレベルでデータにアクセスできる

利点

データ モデルを使用して適切なデータ ウェアハウスを設計すると、データ処理に関する多くの課題を解決できます。主な利点は次のとおりです。

- ビジネス レポート作成用のクエリーが高速に処理されるような構造を設計できる
- ビジネスの要件を確実に満たし、正確で有益なレポートを作成できる
- ソースおよびターゲットのシステムを適切に文書化して、開発作業を支援し、効果的なバージョン管理を実現し、システムの理解を深めることができる

セクション 1

はじめに

多くの組織は、膨大な量のデータを保有しています。取引を行い、従業員の審査を実施し、見込み客の情報を得るなど、さまざまな業務のもとでデータは絶え間なく蓄積されていきます。これらのデータは、組織が運用するシステムの中核に位置しており、各システムに構築されたデータベースは、組織のビジネス プロセスをできるだけ効率的に実行できるよう設計されています。このような環境で問題が発生するのは、たとえば、ビジネス ユーザーが今年の総取引数を知るために、これらのデータからレポートを作成するような場合です。

トランザクション システムを使用したビジネス レポートの作成には、次のような課題があります。

- レポート作成に必要なデータベース設計は、トランザクション システムのパフォーマンスを最適化するような設計とは大きく異なります。
- 基幹トランザクション システム上でレポート作成タスクを実行しても、処理は遅く、基幹システムにも悪影響を与えます。
- トランザクション データベース システムに保存されたデータは集中管理されていません。つまり、各種のレポート作成に利用できる単一の情報ソースがありません。

これらの課題を解決するには、ビジネス レポートの作成を目的としたデータ ウェアハウスを設計して、構築したデータ ウェアハウスに、レポート作成に必要なすべての情報を保存します。

データ ウェアハウスは、ビジネスに役立つ情報の重要な提供元になるため、その物理設計と論理設計の両方をモデル化することが不可欠です。物理設計は、データ ウェアハウスのパフォーマンスと機能を決定し、論理設計は、開発者とユーザーがビジネスの要件を把握するための視点を提供します。

セクション 2

データ ウェアハウスを設計する

データ ウェアハウスのすべてのデータは、トランザクション性能やストレージ容量ではなく、クエリーのパフォーマンスを優先した形式で保存されます。ユーザーは、ビジネスのさまざまな要素についての情報にアクセスできます。また、それらの要素と他の要素との関係や、パフォーマンス評価指標との関係を知ることもできます。ビジネスに関する理解が深まると、組織の業務を十分に把握でき、各種の計画立案にも役立つため、競争上の優位性を得ることができます。

完全なデータ ウェアハウスには、利用可能なすべてのデータが保存されます。ただし、実際には、一部の必要なデータのみがデータベースに保存されることも多く、厳密にはこれをデータ マートと呼びます。

データ ウェアハウスには通常、履歴データが保存されるため、過去の事象に対するクエリーを正確に実行できます。たとえば、オンライン トランザクション処理（OLTP）システムには、ある製品の現在の輸入業者についての情報が保存されています。クエリーを実行すると、この輸入業者の情報が返されますが、過去の取引時の輸入業者と同じかどうかという点は考慮されません。これに対し、データ ウェアハウスには通常、製品のすべての輸入業者についての情報が保存されており、各取引とその時点での輸入業者が正確に関連付けられています。

ビジネスに関する質問に対して有効な回答を返すには、データ ストレージの設計をモデル化することが不可欠です。このセクションでは、データ ウェアハウスの設計手法をいくつか説明します。これらの手法では、多次元データ モデルと呼ばれる特殊なデータ モデルを作成します。

ファクト テーブル

ファクト テーブルは、データ ウェアハウス設計の中心に位置するテーブルです。このテーブルの各 1 行は、1 つのファクト（事象）を表しています。各ファクトには、数値で定量化できるメジャー（たとえば価格など）が 1 つ以上含まれています。さらに、ファクト テーブルには複数のディメンション値も含まれています。ディメンション値はファクトを説明するもので、時間、従業員、顧客、および場所のような値です。

ディメンション

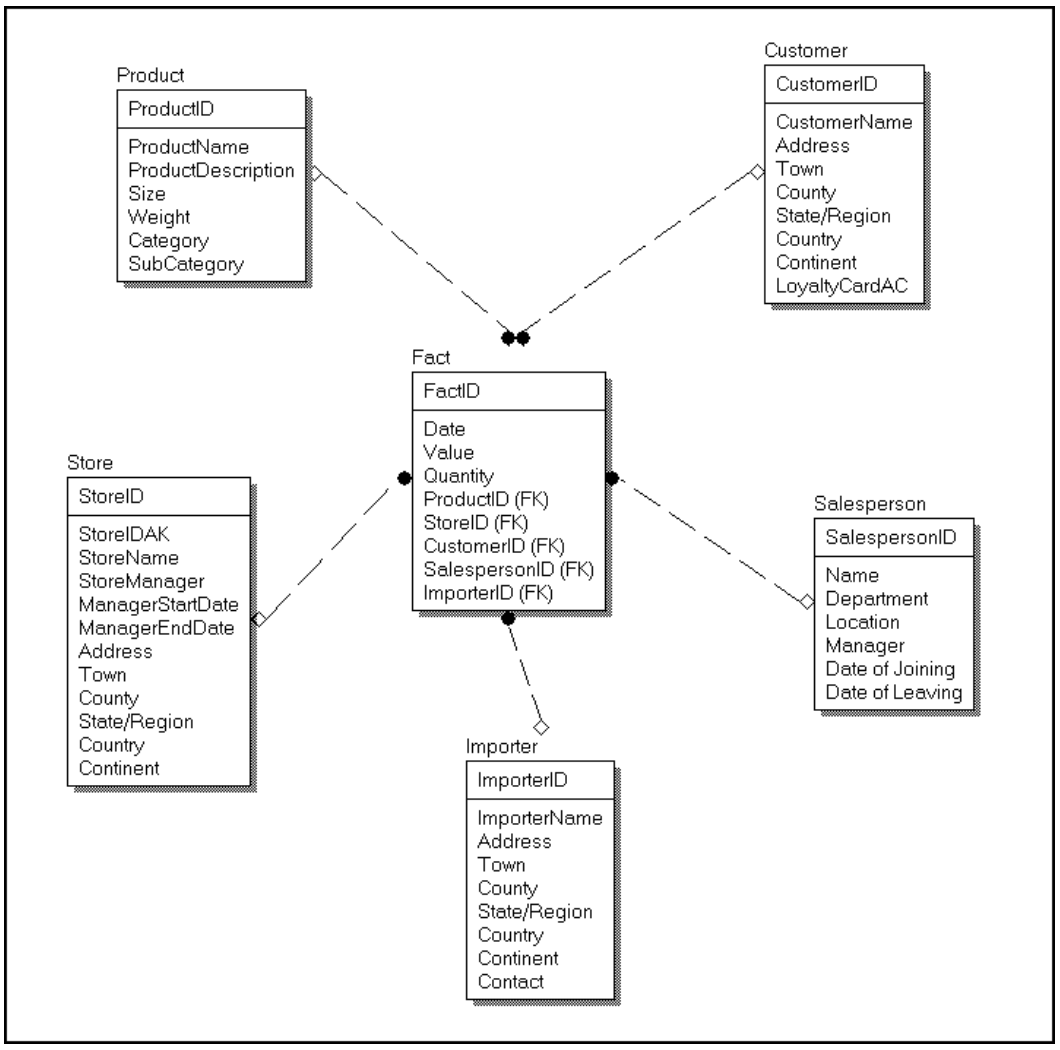
ディメンション値は通常、外部キーとしてファクト テーブルに保存されており、これらの外部キーは、ディメンション テーブルの主キーに関連付けられています。ディメンション テーブルには、各ディメンション メンバーが記述されます。たとえば、ファクト テーブルには販売員の社員番号が含まれており、従業員ディメンションには社員番号、名前、勤務地、およびマネージャー名が含まれています。ディメンション テーブルは、すべてのディメンションに必須という訳ではありません。たとえば、時間ディメンションに時間以外のプロパティが存

在しない場合、ファクト テーブルに含まれることもあります。祝日など、他のプロパティが存在する場合は、ディメンション テーブルが必要です。

スター スキーマ

ファクト テーブルが 1 レベルのディメンション テーブルを持つ場合、そのデータベース設計はスター スキーマになります（図 1 を参照）。

図 1: スター スキーマ



スター スキーマを表すデータ モデル

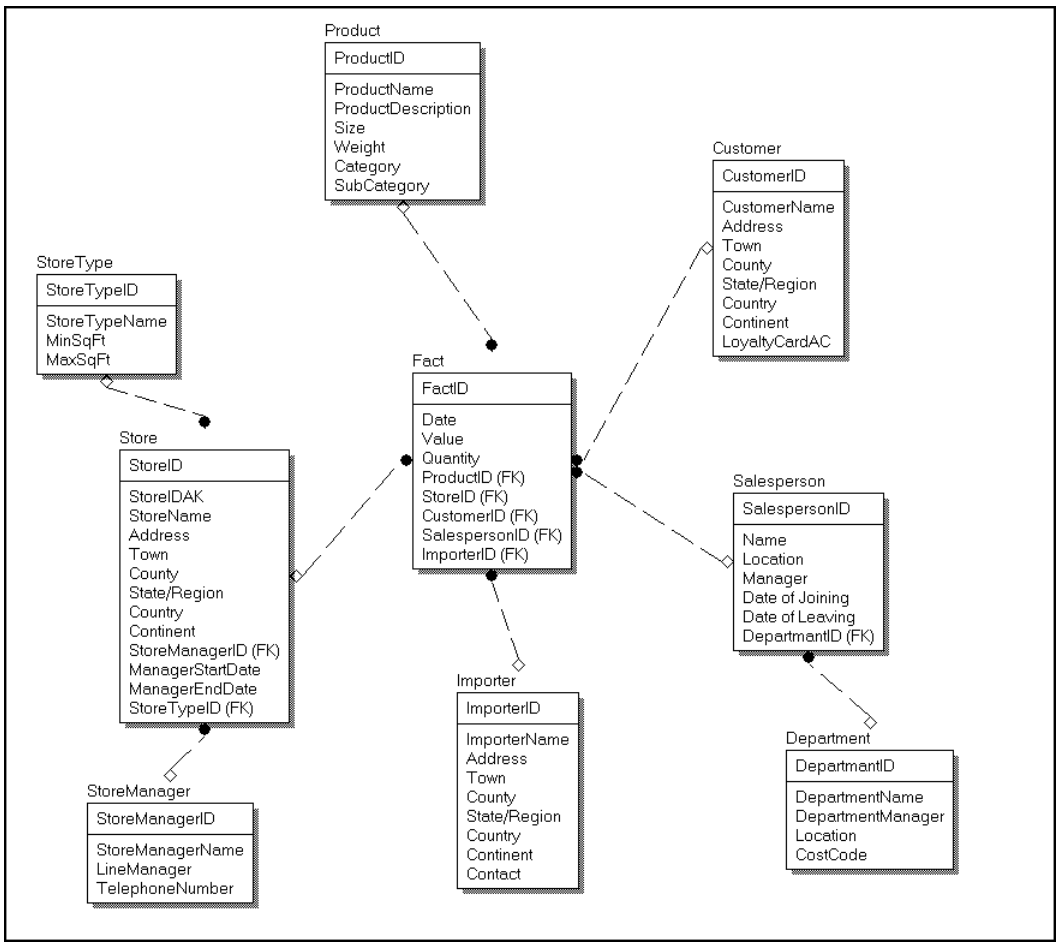
スター スキーマの利点は、ディメンションの任意のプロパティを取得する際に、ファクト テーブルから関連するディメンション テーブルへの 1 回の結合操作で済むことです。これにより、クエリーのパフォーマンスは向上しますが、データの容量は増大します。

スノーフレーク スキーマ

一部の詳細情報が、クエリーであまり使用されない場合は、スター スキーマとは異なるモデルを作成します。たとえば、従業員の所属部門についての情報をクエリーで取得するケースは少ないでしょう。すべての販売員は営業部門に所属しているため、クエリーを使用して営業部門に関するデータを分析する利点はほとんどありません。それでも所属部門のデータを保存しておきたい場合は、従業員のディメンションに関連した別のディメンション テーブルを作成できます。これがスノーフレーク スキーマです（図 2 を参照）。このスキーマの利点は、従業員の所属部門が同じ場合に生じる、情報の重複を取り除くことができることです。ただし、クエリーでスノーフレーク データを使用する頻度には注意してください。スノーフレーク データを使用すると、余分な結合操作が必要になり、クエリー速度が低下するからです。

パフォーマンス上の問題から、通常は、モデルでスノーフレーク スキーマを使用することはお勧めしません。

図 2: スノーフレーク スキーマ



スノーフレーク スキーマを含むデータ モデル

スター スキーマとスノーフレーク スキーマは、ディメンション レベルでモデル化され、データ ウェアハウス全体の設計には適用されません。図 2 では、店舗（Store）と販売員（Salesperson）のディメンションに、スノーフレーク スキーマが使用されていますが、製品（Product）、顧客（Customer）、および輸入業者（Importer）のディメンションには、スター スキーマが使用されています。

ファクト テーブルとディメンション テーブルの設計を見ると、データ ウェアハウスの設計は大幅に非正規化されていることが分かります。正規化は、重複した部分を取り除いて OLTP システムの効率を改善しようとする手法ですが、クエリー速度の最適化を目指して設計されたシステムでは、正規化を行うとパフォーマンスが低下します。

緩やかに変化するディメンション

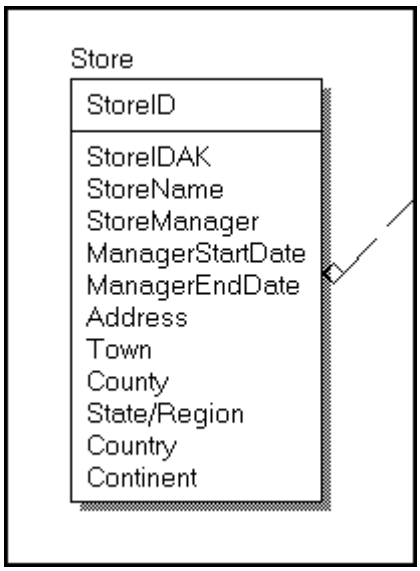
これまで、各ディメンションをある時点におけるスナップショットとして扱ってきましたが、実際には、ディメンション属性は変化します。たとえば、ある顧客がカナダに居住していると記録されていますが、最近フランスから移住していたとします。このような場合、過去にフランスで発生した事象（ファクト）をカナダに割り当てるべきではありません。同様に、従業員が別の部門に異動したり、店舗のマネージャーが交替したりすることもあります。

ディメンション メンバーの時間的変化をうまく保存するには、モデル内に、緩やかに変化するディメンション（SCD: Slowly Changing Dimension）を作成する必要があります。この作業では、正しい結果が得られるようにあらかじめ計画しておかないと、うまくいかない可能性が高いため、モデリングが不可欠です。次の 3 種類の SCD を使用すれば、ほとんどの時間的変化に対応することができます。

タイプ 1 の SCD は、変更が可能な事実上標準のディメンションです。ある属性を変更する場合は、単にその値を変更するだけで、その他の特別な操作は不要です。これによってレコードを最新の状態に保つことができますが、先に説明したような時間的変化による問題は解決されません。

タイプ 2 の SCD（図 3 を参照）では、変更される各ディメンション メンバーに対し複数のレコードを作成して、時間的変化の問題を解決します。たとえば、店舗のマネージャーが交替になると、別のレコードを作成して、新しいマネージャー名を入力します。ただし、この操作によってレコードのキーが重複するため、店舗の元のキーを代替キーとして保存し、新しく一意な主キーを作成します。また、タイプ 2 の SCD では、ディメンションのレコードとファクト テーブルの各ファクトとの対応関係を特定する必要があります。この問題を解決するには、各レコードの開始日と終了日を保存します。

図 3: タイプ 2 の SCD



代替キー、開始日、および終了日が含まれる、タイプ 2 の SCD

図 3 の SCD には、代替キー（StoreIDAK）、マネージャーの開始日（ManagerStartDate）、および終了日（ManagerEndDate）が含まれています。現在のマネージャー名は、ManagerEndDate フィールドが NULL のレコードを検索すれば簡単に見つかります。

タイプ 3 の SCD は、タイプ 1 とタイプ 2 の SCD の中間的な方法です。ディメンション メンバーの元の値と現在の値のみが保存されます。たとえば、初期価格と現在の価格を知りたい場合、その他の価格は不要なので、タイプ 2 の SCD のような複雑な構造にする必要はありません。単純に、元の値と現在の値をディメンションの別々の属性として保存するため、各ディメンション メンバーに必要なのは 1 行のみです。この方法ではストレージを単純化できますが、機能が制限されるため、ほとんどのモデルではタイプ 1 またはタイプ 2 の SCD が使用されます。

このように、データ ウェアハウスの設計は OLTP システムとは大幅に異なるため、ビジネス レポートの作成用にデータ ウェアハウスを最適化するには、特別な設計手法が必要です。

セクション 3

抽出、変換、および読み込み（ETL）

データ ウェアハウスの設計は OLTP システムとは大きく異なりますが、データ ウェアハウスの主なデータ ソースは OLTP システムです。OLTP システムからデータ ウェアハウスにデータを移動する方法は、慎重に計画する必要があります。このプロセスを ETL（Extract, Transform, and Load; 抽出、変換、および読み込み）と呼びます。以下に、ETL システムをモデル化する上で決定すべき事項を説明します。

最初に、利用可能なデータを見つける作業から始めます。小規模で比較的新しい IT 系の企業では、これは難しい作業ではありません。しかし、コンピュータがない時代から取引を行い、何度も合併を繰り返してきたような大規模な国際企業では、非常に複雑な作業になるでしょう。稼働中の各システムを調査して、既存の文書を確認するか、文書が存在しない場合はデータ ストアの情報を文書化する必要があります。これは非常に時間と手間のかかる作業なので、モデリングと文書化の利点は明らかです。データ モデリング ソフトウェアを使用して、多数のシステムをリバース エンジニアリングすると、データ ストアの情報を文書化する時間を節約できます。

データ ウェアハウスに読み込むデータを決定したら、抽出処理を計画します。現在のシステムでは、ODBC のような接続の標準仕様が使用されることが多いため、データの抽出は容易です。これに対し、レガシー システムでは、システムを適切にモデル化してデータ ウェアハウスに読み込む前に、あらかじめデータ プロファイリングのような分析が必要になることがあります。

最初からデータ ウェアハウスに適したデータ形式であることはほとんどないでしょう。先に説明したとおり、データ ウェアハウスと OLTP システムは根本的に異なった設計であるため、データ変換が不可欠です。

データ変換では、SQL クエリーのような簡単な手法で正しい形式のデータを生成できることもありますが、ほとんどのシステムでは、はるかに複雑な変換処理が必要になります。以下の例に示すように、ソース データの多くは一貫性を欠いています。

- ムンバイという都市は、以前はボンベイという名前であり、サンクト・ペテルブルグは、以前はレニングラードという名前でした。
- さまざまな種類の通貨が使用されることがあります。
- 一部のデータ ストアでは必須の属性が省略されていることがあります。
- 同じ値を表すのに異なるコードが使用されていることがあります。

得られたデータを正しく分析できるようにするには、エンティティの属性を標準化する必要があります。データ モデリング ツールを使用して、ソース システムとターゲット システム（特にリポジトリ ベースのシステム）を分析すると、これらの標準を作成して実装する上で役立ちます。

ソース データとターゲット データの文書化および設計が完了すると、ETL ツールを使用してほとんどの変換処理を実行できます。ただし、変換が非常に複雑な場合は、独自のプログラミングが必要になることもあります。

抽出および変換の処理と比べると、読み込み処理は比較的簡単です。抽出および変換後のデータは、データ ウェアハウスへの読み込みに適した形式になっている必要があります。読み込み処理でもっとも重要な検討事項は、実行のタイミングです。この処理はリソースの消費が大きいため、システムがほとんど使用されていないか、あるいはまったく使用されていない時間帯に実行すべきです。一般には、夜間や週末に実行するのが良いでしょう。

ETL 処理には、ソースと対象のデータベース システムについての広範な知識が求められます。データ モデリングを活用すれば、これらの設計を記述することができ、ETL プロセスを正しく設計する上で役立ちます。

セクション 4

データ ウェアハウス構築のベスト プラクティス - データ モデリングの重要性

ほとんどのデータ ウェアハウス設計者は、データ モデリング ツールを使用して、データ ウェアハウスの論理設計と物理設計を作成します。論理設計では、すべてのビジネス要件、定義、およびルールがサポートされます。物理設計では、インデックス、リレーションシップ、データ型、およびプロパティを検討する過程で、パフォーマンスが最適化されます。OLAP、データ マイニング、およびレポート作成システムの開発者にとって、データ モデルは最終的なデータ ウェアハウスの構造が記述されたドキュメントとして役立ちます。

ビジネス要件を収集する

データ ウェアハウス構築のために、物理モデルだけでなく論理モデルを作成することは特に重要です。通常は、ビジネス ユーザーやデータ アーキテクトと共にデータ ウェアハウスの設計を開始して、必要なエンティティや保存するファクトを決定します。この初期設計には、データ ウェアハウスの概要が記述されますが、すべての関係者が満足するまで、設計が何度も繰り返されることがあります。この段階では、データ ウェアハウスの設計で陥りがちな間違いをおかさないように、注意深く作業をすすめます。たとえば、既存システムのデータをデータ ウェアハウスに読み込むことが目的なので、必要のない既存システムの要素を最終設計に放置したり、時間を節約するために既存モデルの設計の大部分を流用したりしてしまうことがよくあります。データ モデリングを使用すると、非常に早い段階でこれらの問題に気づくことができます。

論理モデルは物理モデルの構成要素にすぎないと考えるべきではありません。論理モデルは、物理モデルを作成するために必要な段階であるものの、物理モデルとデータ ウェアハウスを作成した後にも、論理モデルにはさまざまな用途があります。論理モデルにはビジネスの要件が反映されており、モデルの名前付け規則は、組織で使用されるビジネス用語と厳密に一致しています。そのため、論理モデルは外部公開用の設計として役立ちます。他のシステムの開発者は、この設計を使用して、データ ウェアハウスへのインターフェイスを作成します。物理モデルの元になる構造を変えることなく、データ利用者のニーズに応じて、さま

ざまな論理モデルを作成できます。論理モデルの開発と保守を続けることで、物理および論理モデリングの両方を同時に修正しようとして物理モデルに生じるリスクを回避できます。

データベースのパフォーマンスを最適化する

クエリーのパフォーマンスは、データ ウェアハウスにおける重要な要素です。クエリーのパフォーマンスを最大限に引き上げるために、データの容量やトランザクションのパフォーマンスを犠牲にしても、データベース設計が適切でない場合は、クエリーの実行は最適化されません。

膨大な量のデータが処理されるため、明確な方針がないままデータ ウェアハウスの設計を行うのは非常に困難です。データ モデリング製品を活用すると、このプロセスの自動化に役立ち、データ ウェアハウスに関連するメタデータや、ビジネス インテリジェンス（BI）システムで使用されるデータの管理が容易になります。ディメンション エンティティとファクト テーブルの設計を終えると、テーブル間のリレーションシップを決定することができます。作成された設計は BI チームによってレビューおよび評価され、その報告に基づいて必要な変更を加えます。この段階では、非常に簡単に設計を変更できます。特に、データ ウェアハウスの構築後に変更作業を行う手間を考えると、その違いは明らかです。論理設計が承認されると、物理設計の作業を開始できます。

物理設計では、データ型やインデックスのような詳細情報を追加しますが、パフォーマンスを向上させるために、基本エンティティの設計を変更することもあります。物理設計は、パフォーマンスの向上において特に重要です。OLTP システムでは、保存するデータで必要とされるより若干大きいデータ型が使用されることがよくあります。たとえば、4 バイト整数で十分な場合でも、代わりに 8 バイト整数が使用されることがあります。数百万ものレコードが含まれるシステムでは、このようなわずかな非効率性が無視できないほどになります。データ ウェアハウスのインデックス作成は、さらに重要な要素です。データ ウェアハウスの設計では、クエリーのパフォーマンスを重視しますが、この点でもっとも重要な要素は、スターやスノーフレークのようなスキーマと、インデックス設計です。必要があれば、データ ウェアハウスのスキーマを、スター スキーマからスノーフレーク スキーマに変更することもあります。

ソース システムとターゲット システムの情報を提供する

ETL システムのモデリングでは、対象システムのモデルだけでなく、ソース システムの論理モデルと物理モデルを検証する作業が重要です。多数の工程を一度に実行することはできないため、しばしばステージング領域用の中間モデルを作成する必要があります。ソース システムと対象システムの要件によっては、抽出、変換、および読み込み（ETL）の処理を一度に実行できないこともよくあります。

セクション 5

まとめ

組織が保有する膨大な量のデータを活用することができれば、ビジネスで大きな利益を得ることができます。過去の結果を的確に分析し、その情報を BI システムに投入してデータの相関関係を見つけ出し、さらに、ビジネス ユーザーが利用しやすい方法で必要な情報を表示することができます。

このようなシステムを構築するには、データ ウェアハウスのモデルを注意深く作成しなければなりません。さまざまなソース システムがあり、その一部が正確さを欠く場合もあります。対象システムにはさまざまな要件があり、各ビジネス ユーザーからもさまざまな要求があります。そのため、データ ウェアハウス システムのモデリングには十分に時間をかける必要があります。データ ウェアハウスは費用と時間がかかる作業ですが、正しく設計できれば、ビジネスに大きな利益をもたらします。

ビジネスの目標を達成するには、その目標に向けて設計されたデータ ウェアハウス システムをモデル化することが重要です。さまざまなデータ利用者のニーズに応じた論理モデルを作成し、必要なデータベース パフォーマンスを実現できる物理モデルを作成します。そしてもっとも重要な点は、常にビジネスのニーズを満たすよう努めることです。


Transforming
IT Management.