データ モデルを活用した データ ウェアハウスの設計

Joshua Jones, Eric Johnson

目次

はじめに

データの保存と操作がもっとも重要となる状況の一つは、そのデータを使用して重大な意思決定が下されるときでしょう。企業は数十年も前から、社内に保存されたデータを検索してきましたが、高度なデータ マイニングやデータ ウェアハウス構築の技術が大企業で注目されるようになったのは、ここ数年のことに過ぎません。データ ウェアハウス構築が特に有益なのは、売上高、注文、生産高などについて大量の履歴データを蓄積している大企業です。企業のビジネスを成功へ導くために、拡張性の高い正確なデータ ウェアハウス ソリューションを構築することが、ますます重要になっています。

データベース モデリングは、従来のリレーショナル データベースに対するオンライン トランザクション処理（OLTP）モデルだけでなく、データ ウェアハウス構築の領域でも利用されるようになっています。データ ウェアハウスには、スキーマの設計という要件に加えて、いくつかの重要な要素があります。データ ウェアハウスの作業では、メタデータについても考慮しなければならず、さらに、データを 1 つ以上のソースからデータ ウェアハウスへ配置する方法も検討する必要があります。このホワイト ペーパーでは、データ ウェアハウスの設計、メタデータの作業、および ETL プロセスの構築について概説し、データ ウェアハウス ソリューションを設計するための一般的な指針を示します。

データ ウェアハウスの設計

一言で言えば、データ ウェアハウスはデータの保存場所です。通常は、企業によるビジネスの履歴データが保存されます。これはデータベースに似ているように思えますし、実際、もっとも基本的な定義ではデータベースそのものです。しかし、一般的にデータベースという用語は、リレーショナル データベースを指します。さらに具体的に言えば、オンライン トランザクション処理（OLTP）に使用されるリレーショナル データベースのことです。OLTP データベースは、トランザクション負荷のもとですぐれた性能を発揮するように設計されており、多くの場合、購買システムのインターフェイスのような特定のアプリケーションをサポートしています。一方、データ ウェアハウスはオンライン分析処理（OLAP）のために構築され、データを分析し、その結果に基づいて意思決定を下すユーザーのニーズに応えるものです。

データ ウェアハウスの構築については、厳密に定義され統一された標準の方法はありませんが、データ ウェアハウスの設計については、少なくとも多次元モデルと正規化モデルという、よく知られた 2 つの方法論があります。これらの方法はどちらも有効ですが、それぞれ利点と欠点があります。このホワイト ペーパーでは多次元モデルのアプローチを使用しますが、以下に 2 つの方法についての基本的な知識を説明します。

正規化モデルは、E. F. Codd によって定義された正規形に基づいています。このモデルは、データを第 3 正規形（3NF）で保存して、トピック（顧客、注文、製品など）ごとにデータをグループ化するというものです。正規化モデルを使用すると、既存のデータに影響を与えずに新しいトピックを追加できるため、データ ウェアハウスに新しい種類のデータを追加する更新作業を簡単に行うことができます。ただし、非技術系のユーザーがアドホック クエリーを実行する際に、データがどのように関連付けられているかを理解しなければならないという難点があります。さらに、各クエリーに関わるテーブルの数によっては、レポート作成用のクエリーが高速に実行されないこともあるでしょう。

多次元モデルでは、トランザクション データを「ファクト」と「ディメンション」の集まりにまとめて、データ ウェアハウスを一から構築して記述します。通常、ファクトは数値データ（金額や在庫数など）であり、ディメンションは、それらの数値データ（ファクト）をさまざまなコンテキスト（販売員の名前や地域など）に関連付けるための情報です。多次元モデルでは、非技術系のユーザーでも効率的なクエリーを実行できます。データが論理的な手法でグループ化されており、同種のデータがまとめて保存されることで高いパフォーマンスを実現できるからです。ただし、多次元データ ウェアハウスの管理にはいくらか手間がかかります。新しい種類のデータを追加するには、データ ウェアハウス全体の更新作業が必要になります。

最後に、どのようなデータがデータ ウェアハウスに格納されるのかを把握しておくことが重要です。明らかにデータの大部分（またはすべて）は、企業が運用している業務用データベースから収集されます。大企業では、メインフレームや分散システム上のデータベースからデータが収集されるでしょう。ファイル共有や電子メール サーバーなど、非リレーショナル ソースにデータが保存されている場合もあります。そのため、データ ウェアハウスを設計する際には、データをデータ ウェアハウスに移行するプロセスも設計する必要があります。このようなプロセスを、しばしば ETL（Extract, Transform, and Load）と呼びます。ETL は主に、抽出、変換、読み込みという 3 つのステップから構成され、未加工のデータを処理してデータ ウェアハウスに格納します。

データ ウェアハウスのモデリング

リレーショナル データ モデリングと同じく、データ ウェアハウスのデータ モデルを構築するには、プロジェクトを開始する前に多くの作業を行う必要があります。データ ウェアハウスは、高度に構造化された固有の成果物であるためです。モデル作成者は、最初に保存されるデータや、将来追加されるデータをあらかじめ把握しておかなければなりません。データ ウェアハウスを設計するためのアプローチは、アプリケーションの設計プロジェクトと非常によく似ています。実際、データ ウェアハウスは多

くの場合、フロント エンドのレポート作成システム（Web ベースのアプリケーションやデスクトップ クライアント）と統合されます。そのため、作業をすすめる際には、要件を収集し、ドキュメントとダイアグラムを作成して、多数のメタデータ（データに関するデータ）を生成するといった標準の設計手順に従います。これはデータ ウェアハウスの構造と範囲を定義するのに役立ちます。詳細については説明を省きますが、データ ウェアハウスを構築するには、これから説明する技術的なプロセスを開始する前に、上記の手順を完了している必要があります。

データ ウェアハウスの要素

多次元データ ウェアハウス モデルの基本的な構成要素は、スキーマに配置されたテーブルとリレーションシップです。具体的には、ファクト テーブルとディメンション テーブルがあり、それらはテーブル間のリレーションシップを把握しやすい形で互いに関連付けられます。このようなレイアウトをスキーマと呼びます。多次元データ ウェアハウスでは、主に「スター スキーマ」と「スノーフレーク スキーマ」という 2 種類のスキーマが使用されます。

スター スキーマ

スター スキーマは、もっとも単純なスキーマの一つです。通常、1 つのファクト テーブルの周囲に複数のディメンション テーブルが単一の階層で関連付けられています。星型の構造になるため、この名前が付いています。各ディメンション テーブルには 1 つの主キーがあり、1 種類のデータに関与しています。ファクト テーブルには複合主キーがあり、これは各ディメンション テーブルの主キーから構成されます。図 1 は、売上データの概要を表す単純なスター スキーマです。

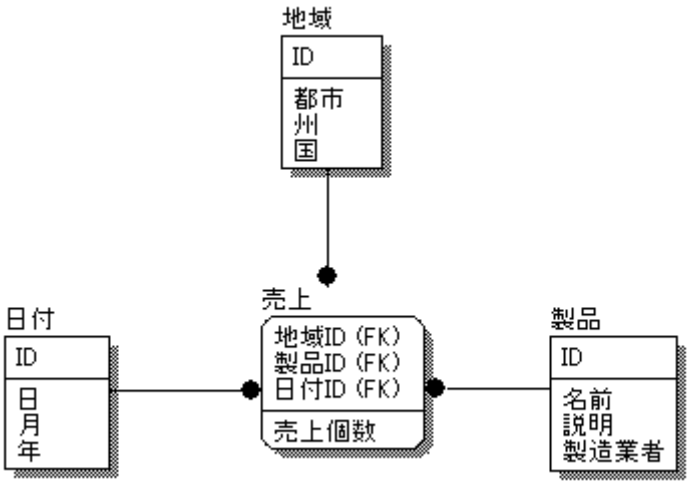

図 1: 単純なスター スキーマ

スター スキーマは単純な構造なので、ファクト テーブルに対して 1 階層のディメンション テーブルを結合すれば、データを取得できます。スター スキーマの欠点は、

複雑なシステムで大量のファクト データを取得するときに、データの読み取りとクエリーが非常に複雑になる場合があることです。

スノーフレーク スキーマ

スノーフレーク スキーマは、スター スキーマとよく似ていますが、ディメンションとファクトとの間のリレーションシップに違いがあります。ファクト テーブルが中心に配置され、周囲のディメンション テーブルとの間にリレーションシップがある点は同じですが、多数の属性を含むディメンションが、サブディメンション テーブルに分割されています。そのようなスキーマは、ディメンション テーブルが複数の階層にわたっているため、見た目がスノーフレーク（雪片）に似ています。図 2 は、データ ウェアハウス内の売上データをやや詳しく表したスノーフレーク スキーマです。

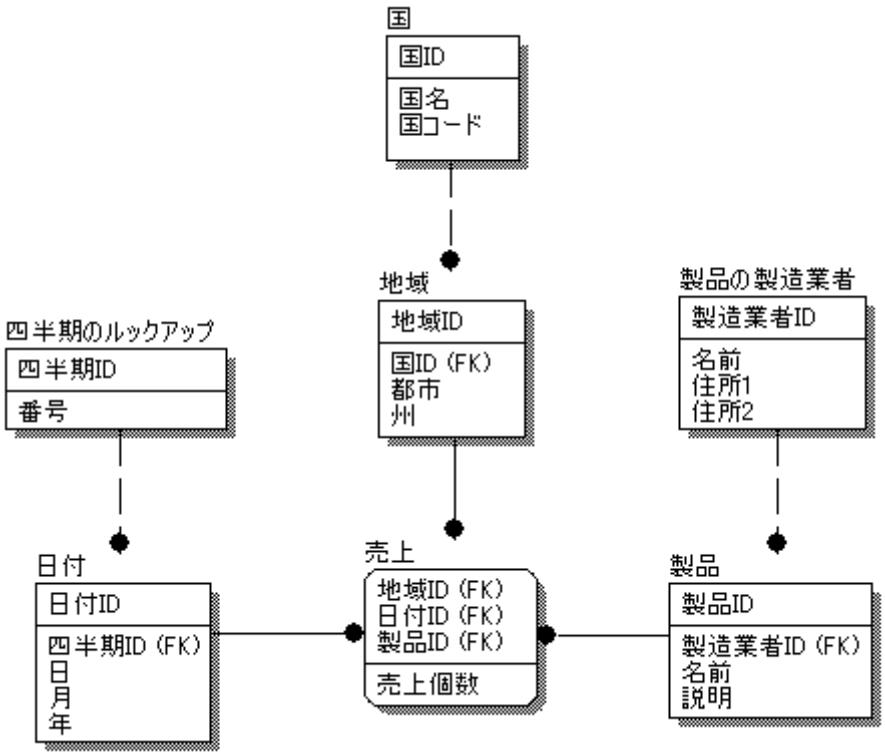

図 2: スノーフレーク スキーマ

図 2 はデータ モデル全体を表したものではありませんが、基本的なスノーフレーク スキーマになっています。中心にファクト テーブル、その周囲に 2 階層のディメンション テーブルがあり、データは効果的に正規化されています。スノーフレーク スキーマの利点は、データの論理的な関連性を把握しやすいことです。たとえば、図 2 で売上から地域までドリルダウンして、国コードを参照することができます。大規模なシステムでは、スノーフレーク スキーマを使用すると、さまざまなディメンションにわたって複雑な比較操作が必要なデータ マイニング クエリーを容易に実行することもできます。たとえば、ある国における 2007 年の第 3 四半期の総売り上げを製品の

製造業者ごとに確認するには、その言葉どおり 7 つのテーブルを結合します。データがスター スキーマに保存されていれば、4 つのテーブルを結合することになります。ただし、パフォーマンスの観点から見ると、スノーフレーク スキーマは、少数のディメンションからデータを取得するだけのクエリーが多数ある場合に速度が低下することがあります。スター スキーマとスノーフレーク スキーマは、パフォーマンスとユーザビリティに違いがあるため、目的に合ったスキーマを構築することが重要です。

モデルの構築

あるプロジェクトが始まり、要件は収集済みであるという状況を考えてみましょう。DVD ストアのデータ ウェアハウスを構築して、受注製品とその購買層を知ろうという、かなり単純なプロジェクトです。このようなニーズはよくあります。ほとんどのビジネスは、製品またはサービスのいずれかを販売するものであり、詳細が違っていても基本は同じだからです。必要な作業は、すべてのファクトとディメンションを含むスキーマのモデルを作成して、データ ウェアハウスの管理方法と、ETL プロセスをどのように実行するのかを検討することです。

これから行う作業はモデリングの段階なので、標準的なモデリング用語（エンティティや属性など）を使用します。データ ウェアハウス用のデータ モデル構築は、通常のリレーショナル データベースの場合よりもはるかに容易ですが（エンティティやテーブル間のリレーションシップがほとんど常に 1 対 1 となるため）、物理データベースの構築を実際に開始するまでは、適切なモデリング用語を使用すべきです。

主なデータ ソースは、受注システムで使用される業務用データベースであることが分かっています。業務用データベースのデータ モデルは作成済みなので、少なくとも設計段階では、そのデータ モデルをデータ ウェアハウス構築の出発点として利用できます。このように恵まれた状況でない場合は、最初の作業として既存データベースのデータ モデルを作成する必要があるかもしれません。また、このデータ ウェアハウスの構築では、リレーショナル データベース エンジンと、データ ウェアハウス用のデータ ストレージ領域の両方に Microsoft SQL Server を使用します。はじめにデータモデルを作成してから、そのモデルをサポートするデータベース上に必要なテーブルを配置していきます。

図 3 に、ソース データベースの単純なデータ モデルを示します。

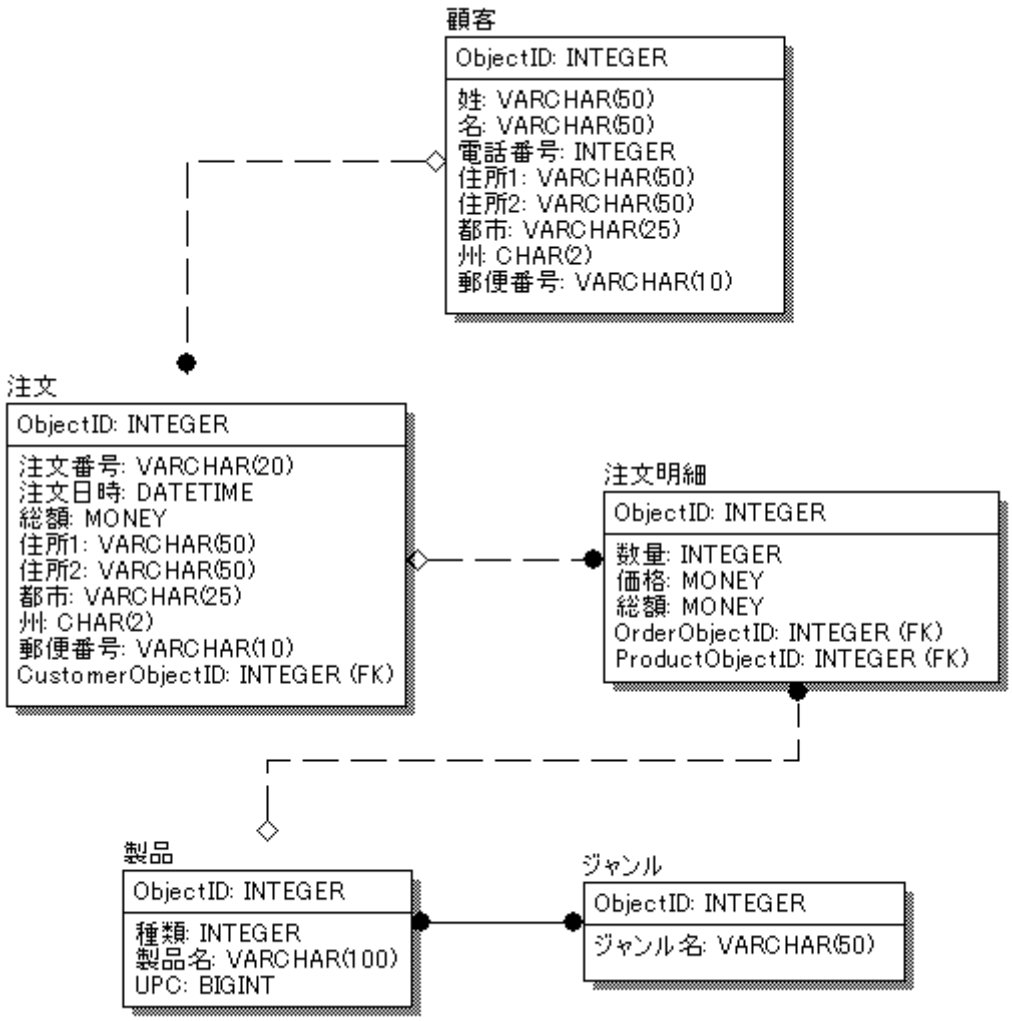

図 3: リレーショナル データ モデル ソース

図 3 は標準的な正規化データ モデルであり、注文、顧客、および製品のデータが含まれています。このモデルをスター スキーマに変換して、必要なデータを基にファクト テーブルを配置します。そのためには、保存されるデータを論理的な観点から理解する必要があります。

モデルを確認して最初に分かる事実は、DVD（製品）に対して注文が発生するということです。そこで、「注文ファクト」というファクト エンティティを作成して、注文の日付、品目、および送付先などを記述するディメンションを追加します。リレーショナル モデルには、注文に注文明細があり、そこから製品へのリレーションシップがあります。これは 1 回の注文に複数の製品が含まれる場合があるためです。また、注文は顧客が行いますが、顧客データは比較的単純な 1 つのエンティティに保存されています。これらのデータを分析する上で注目すべき点は、通常、データを説明するときの観点は、特定の製品（X という作品は先月いくつ売れたか?）、または顧客（国内と国外ではどんな作品が人気なのか?）のいずれかになるということです。したがっ

て、［注文］エンティティと［製品］エンティティ間のリレーションシップが離れている事に注意を払う必要はありません。製品と顧客は、いずれも「注文が発生する」という 1 つの事実（ファクト）に関連付けられます。その他すべては、注文の詳細を説明しているにすぎません。

それでは、データ モデルにオブジェクトを追加していきましょう。図 4 に、ファクト エンティティである［注文ファクト］を示します。スター スキーマでは、ファクト エンティティの主キーはディメンション エンティティの外部キーから構成されます。そこで、注文を条件付けるディメンションではなく、注文に固有の属性を先に追加します。これらの属性は、どの注文品目に対してもフラットな構造であるため、リレーショナル データベースの［注文］エンティティと［注文明細］エンティティのデータを 1 つにまとめます。

図 4: ディメンションが関連付けられていない［注文ファクト］エンティティ

このエンティティは、各注文を表すファクトの基本構造です。"注文番号 ID"および"注文明細 ID"属性は、それぞれ［注文］および［注文明細］エンティティの主キーです。これによって、すべての品目を対応する注文に関連付けることができます。"注文数量"、"単価"、および"総額"属性は、必要な詳細情報です。次に、これらの数値に関連するコンテキストを配置します。注文があったのはいつか? これらの品目を注文したのは誰か? 各品目の詳細情報は? といった質問に答えることで、必要なディメンションを決定することができます。

はじめに、製品のエンティティをモデルに追加します。製品の情報は注文内容の一部なので、注文のディメンションになります。図 5 では、［製品ディメンション］エンティティが追加され、［注文ファクト］エンティティとの間にリレーションシップが作成されています。

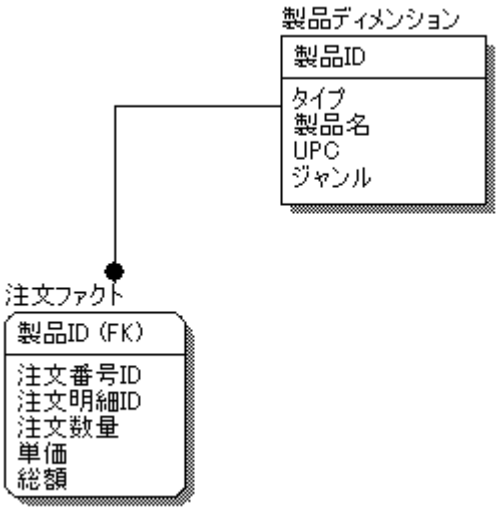

図 5: ［注文ファクト］および［製品ディメンション］エンティティ

次に、顧客のエンティティを追加します。ここで取り上げている例では、顧客情報のリレーショナル ストレージが 1 つのエンティティに保存されているため、顧客エンティティは比較的シンプルです。図 6 に、［顧客ディメンション］エンティティが追加された単純なデータ ウェアハウスのデータ モデルを示します。

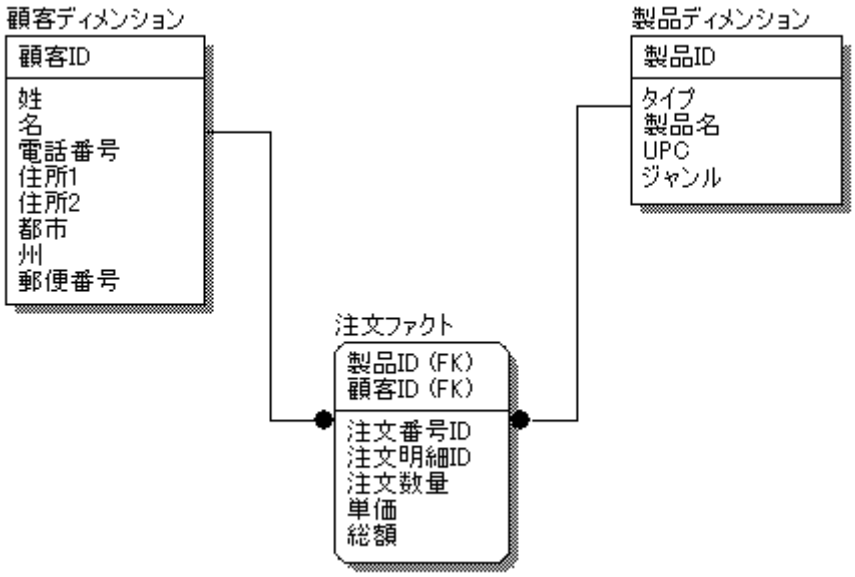

図 6: 注文情報を表す単純なスター スキーマ

これを見ると、データ ウェアハウスでクエリーを実行するために、データがどのようにフラット化（非正規化）されているかが分かります。図 6 のモデルは、トランザクション システムではうまく動作しないでしょうが、多数のレコードを結合して特定のデータ要素を集約するようなクエリーを実行する場合に役立ちます。ここでは、非常に簡単な例を挙げて、論理モデルや物理データベースに基づいてデータ ウェアハウス

のスキーマを構築する方法を説明していますが、ソース データベースに多数のデータが保存されているプロジェクトでは、さらに複雑な結果が得られます。下位レベルの構造は、複数のスター スキーマの集まりになるか、または 1 つのスノーフレーク スキーマ（スター スキーマの集まりをいくらか正規化したもの）のいずれかになるでしょう。図 7 は、これまでの単純なデータ ウェアハウスをいくらか拡張したものです。一部のデータが、ルックアップ用に別のディメンションに移動されました。

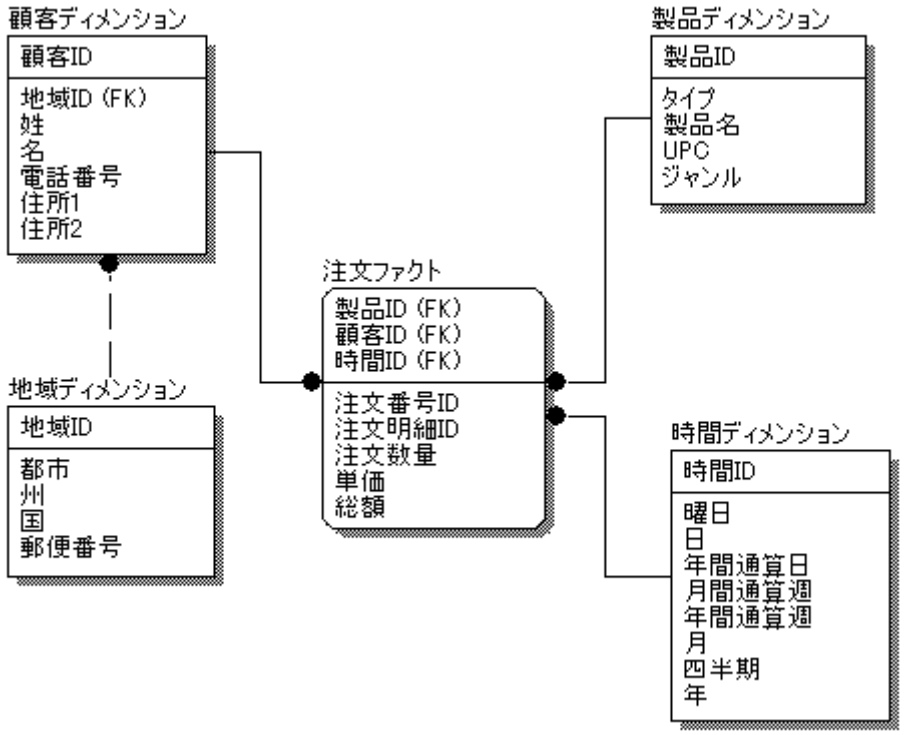

図 7: より詳細なデータ ウェアハウスのスター スキーマ

このスキーマから、さらに詳しい情報が分かります。［時間ディメンション］エンティティが追加され、注文を受けた日付をより詳しく分析できます。［地域ディメンション］エンティティも追加され、顧客の居住地域についての詳細情報が別のエンティティに保存されました。ファクト エンティティから複数の階層にわたってディメンションが存在するので、図 7 はスノーフレーク スキーマの形式です。必要があれば、別のファクト エンティティから特定のディメンション（［時間ディメンション］や［地域ディメンション］）を再利用することもできます。たとえば、このスキーマが、ベンダー、発送情報、従業員などのデータを含む大規模なシステムの一部である場合、再利用可能なディメンションを使用すると、データを標準化して一貫性を保つのに役立ちます。ただし、パフォーマンスが低下したり、データを視覚的に分析するのが困難になることがあります。データ ウェアハウスを構築する際は、非技術系のユーザーが、特定のデータを参照できるアドホック ツールを使用する可能性を考慮して、できるだけ構造を最適化する一方で、ある程度の分かりやすさを保つようにしてください。

論理モデルを構築する際に CA ERwin Data Modeler のようなツールを使用すると、論理モデルに基づいた物理モデルを簡単に生成できます。その後、Microsoft SQL Server などの物理ストレージ媒体に必要なスキーマを作成して、データの読み込みを開始できます。

抽出、変換、および読み込み（ETL）

これまでに何度か説明したように、データ ウェアハウスのデータは他のシステムから収集されます。他のシステムのデータをどのように抽出して、データ ウェアハウスの仕様に沿って変換し、最終的にデータ ウェアハウスに読み込むのでしょうか? このプロセスは、抽出、変換、および読み込みという 3 つのステップから構成され、ETL（Extract, Transform, and Load）と呼ばれます。データ ウェアハウスには、さまざまな種類のソースからすべてのデータを取得して読み込むために、適切な ETL ソリューションが必要です。また、すでに述べたように、データ ウェアハウスのスキーマは OLTP システムのスキーマとは大きく異なっているため、取り込むデータを変換しなければなりません。独自のアプリケーションまたはサードパーティ製の ETL ツールのどちらを使用しても構いませんが、この処理は必須です。

ETL 機能を実行するツールは、数十種類の製品が市販されています。Microsoft の SQL Server Integration Services のようなデータベース ベンダー固有の統合ツールから、Informatica がリリースしているようなスタンドアロンの ETL ツールまで、さまざまな製品があります。ETL ツールによって処理しなければならない重要な領域はいくつかあります。特に、データの変換、データ ソースとの接続性、およびメタデータの相互運用性に関する機能は、すぐれた ETL ツールではいずれも不可欠です。ツールを必ず購入しなければならない訳ではありませんし、独自の ETL プロセスを開発する必要が生じることもあります。いずれにせよ、データ ウェアハウスには、クレンジング済みの正確なデータをタイミングよく読み込む必要があります。それでは、ETL プロセスの各フェーズを詳しく見ていきましょう。

抽出

すでに説明したように、データ ウェアハウスでは、さまざまな種類のデータ ソースが使用されます。多くの場合、データ ソースは、企業情報の大部分が保存されたリレーショナル データベースですが、XML ファイルなどのフラット ファイルや、VSAM データベースなどのレガシー システムに情報が保存されていることもあります。ソースの種類によらず、抽出フェーズにおいて重要な点は、データを解析して共通の形式にすることです。これによって、次の変換フェーズで、データの検証およびクレンジング処理を高速に繰り返すことができます。

変換

変換フェーズでは、抽出されたデータに対してルールの適用と計算を実行して、データ ウェアハウスに取り込む形式に変換します。リレーショナル データベースからのデータ変換は、例外はあるものの、通常はかなり容易な作業です。一方で、その他のデータ ソースのデータをクリーンアップする作業は、ストレージの違いのため（したがってデータ型が異なる）かなり困難な場合があります。また、異なるソースに保存されているデータを、読み込みフェーズで結合しなければならないこともあります。たとえば、請求システムの基本データは、ソースの業務データベースに保存されており、実際にクライアントに送られた請求明細書は、特定のクライアントのシステムに配布するために XML ファイルで保存されているとします。この 2 つの情報をデータ ウェアハウスに保存するには、変換フェーズで関連するデータを結合して、読み込みが容易な形式にまとめる必要があります。このために、日付および顧客番号を比較して、請求明細書と業務データベースの請求情報を正しく関連付ける処理も行います。変換フェーズで実行される一般的なタスクの一部を次に示します。

- 重複したデータを除去する
- 形式やビジネス ルールの観点からデータを検証する
- ディメンション テーブルに代理キーの値を生成する
- 計算値を導出する
- データをクレンジングして、コード化されている値を人が理解できる形式にデコードする（1=男性、2=女性など）

読み込み

データをクレンジングして共通の形式と構造にまとめると、データ ウェアハウスに読み込む準備が完了します。このフェーズではタイミングが重要なので、データを読み込む頻度や読み込みに最適な時間を検討します。読み込みフェーズは負荷の高いプロセスであることが多いため、ソースおよび対象のシステムの使用率が低いときにプロセスを実行するようにスケジュールし、ユーザーや顧客への影響を最小限に抑える必要があります。また、高度な分析用ソフトウェアを使用してデータ ウェアハウスのデータを処理している場合、データ ウェアハウスのデータの再読み込み後に、分析処理を再実行する必要があります。いずれにせよ、読み込みフェーズは、パフォーマンスとユーザーへの影響を考慮して注意深く設計しなければなりません。

最後に、すべての ETL プロセスで重要となるのは拡張性です。ETL プロセスは、ビジネス ルールやニーズの変化に伴って、新規ソースの追加や変換フェーズの修正に対応できる必要があります。データ ウェアハウス用の ETL プロセスを設計する際には、容量の増加と機能の拡張を考慮に入れるようにしてください。

メタデータ

メタデータとは、データに関するデータです。データ ウェアハウスにおけるメタデータは、データがどこで、どのように取得され、どのような意味を持つのかということを表します。

図書館を例に考えてみましょう。図書館に所蔵されている本には膨大な量のデータが含まれていますが、目的の本を見つけるにはどうすればいいでしょうか? このような場合にメタデータが役立ちます。本のメタデータは、書名、著者、出版日付、棚の場所、内容の簡単な説明など、本についてのさまざまな情報から構成されます。このメタデータがなければ、特定の本を見つけることはもちろん、何千冊もの本の中から必要な 1 冊を探し出すことなどできないでしょう。データ ウェアハウス システムも同様です。データ ウェアハウスに保存されるデータの全要素についてメタデータを生成して、データの収集元、使用目的、集約方法、およびデータ ウェアハウス内での保存場所を知る必要があります。

データ ウェアハウスのメタデータには、ETLプロセスおよび保存済みデータの両方についての情報がすべて含まれていることが理想的です。つまり、データ ウェアハウス内のデータに関するデータだけでなく、データ ウェアハウスにデータを取り込む方法についてのデータも保存されます。また、本の索引と同じように、メタデータは階層構造によってその論理順序が表されるため、必要な情報をすばやく読み取って理解することができます。ストレージの形式によらず、データ ウェアハウスのメタデータには次のような 3 つの基本タイプがあります [1]。

1. **RDBMS のメタデータ**: データ ウェアハウスを格納するリレーショナル データベースのメタデータです。テーブル構造やテーブル内のデータに関するデータを表します。
2. **ソース システムのメタデータ**: すべてのコンテンツ データの収集元に関するデータです。スキーマ、更新の頻度、およびプロセス情報はすべて、このメタデータに保存されます。
3. **ステージング メタデータ**: ETL データです。変換の定義、変換先オブジェクトの定義、およびログなどが保存されます。

データ ウェアハウスでは、メタデータの使用法や定義は頻繁に変更されるものの、これらの項目の一部は、データ ウェアハウスの読み込みプロセスや、データ ウェアハウスに保存されているデータを知るための参考資料として利用できます。特に後者の情報は、ビジネス ユーザーがデータ ウェアハウス内のデータを参照する際に役立つため、頻繁に更新するようにしてください。

[1] Ralph Kimball, The Data Warehouse Lifecycle Toolkit, Wiley, 1998, ISBN 0-471-25547-5

まとめ

データ ウェアハウスは、企業が将来の戦略を正しく決定する上で欠かせないツールです。データ ウェアハウスの構築はデータベースの構築と似ていますが、独自の手法と準備が必要です。既存のデータを徹底的に分析して、適切なファクトとディメンションを決定し、使用するスキーマを早い段階で決定してください。モデルを構築したら、データ ウェアハウスの ETL プロセスを設計するために、サードパーティのツールを検討することになるでしょう。通常、ETL プロセスをゼロから構築する作業は、非常に複雑で困難であるためです。必要な情報をすべて文書化して、自身や他のユーザーのためにデータ ウェアハウスのメタデータを構築してください。CA ERwin Data Modeler を使用すると、ユーザーの環境に基づく詳細なメタデータの作成や、データ ウェアハウスのモデリング プロセス全体の管理に役立てることができます。さらに、リレーショナル データベースとデータ ウェアハウスの両方のモデルを ERwin で作成すると、モデル管理のプロセスを統一することができ、将来のデータ モデル拡張が容易になります。