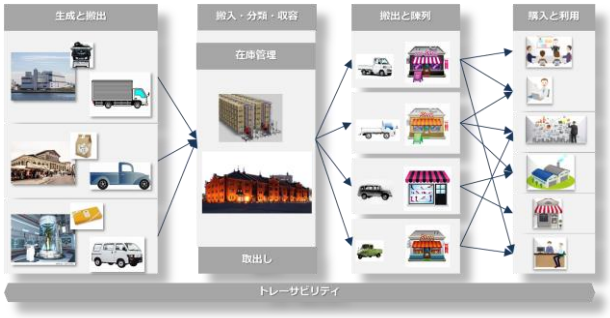
D-019

データウェアハウスの設計と実装

中村　正治
Masaharu NAKAMURA

1．データウェハウスの定義

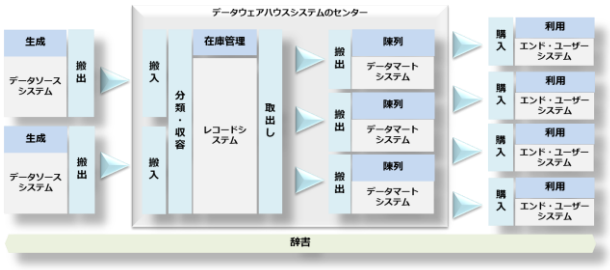
データウェアハウスは文字通り、データの「倉庫」である。その根本機能は、データの使途に依存しない、データそのものの「倉庫」での管理である。そして、管理である以上、データをその倉庫に受け入れるメカニズム、分類整理して棚に収める仕掛け、多様な用途時間的要求に対応できるような蔵出しと出荷の機能、および使用者に対するその在庫の提示の要件がシステムアーキテクチャーに内包されていなければならない。



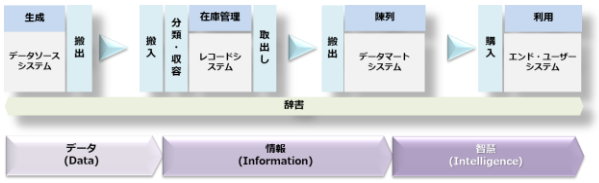
この構造は、そのままデータウェアハウスの構成要素となっている。

このうち、データソースシステムは、業務そのものを実行する基幹部分であり、エンドユーザーシステムは、開発管理と予算配置を実際に使用するユーザー自体が行うものである。

従って、管理対象としてのデータウェアハウス・システムは、レコードシステムとデータマートシステムを指す言葉として使用されることが多い。



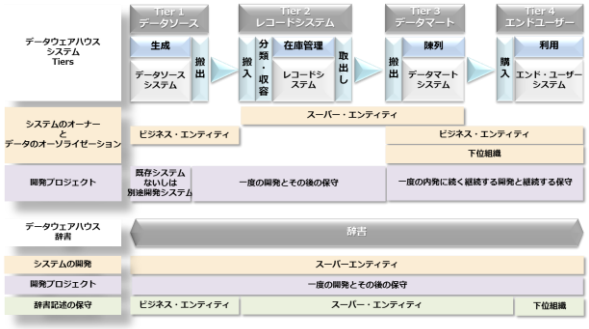
データウェアハウスは、企業体など、ビジネスの目的と実施を共有する「スーパー・エンティティ」が、刻一刻生成する様々なデータを統一的・一元的な視点から整合性を持つ情報として蓄え、それを智慧として終端的利用者に届けるメカニズムであるといえる。



2．データウェハウスの構造

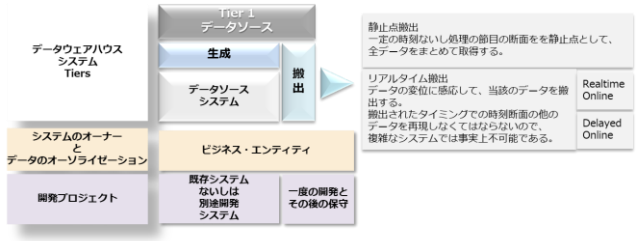
データウェアハウスは、4 つの「Tier」が直列したシステムである。この一連の流れは逆流がないという大きな特徴を持つ。

夫々の Tier は、顕著な特徴を持つので、企画・開発・管理の面での十分な考慮が必要である。



2.1 データソースシステム群

データソースシステム群はデータウェアハウス・システムのセンターの外にある。その多くは、いわゆる Mission Critical なシステムであり、個々のビジネス・エンティティーの心臓部となっている。



データウェアハウス・システムの構築に際しては、そのデータを搬出する機能の構築が求められる。

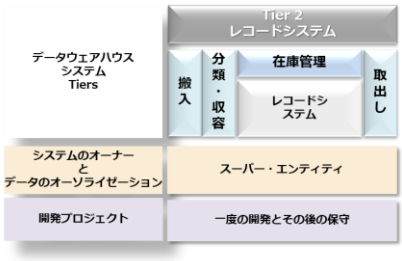
データの搬出は、静止点を設定する方法と、随時搬出する方法を選択することになる。

静止点を設ける場合、静止点と静止点の間のデータの動きは、レコードシステムでは失われることになる。随時搬出する場合、リアルタイム搬出は、高い性能を要求されるソースシステムにとっては大きな負担になるので、遅延(delayed)オンラインとすることが合理的である。

2.2 レコードシステム

System of Record とも呼ばれる、文字通り、データウェアハウス・システムのセンターである。

このシステムの一番の特徴は、中心部にある大きな正規化データベースである。このシステムは、ビジネスプロセスの実装を全く持たないことも大きな特徴である。

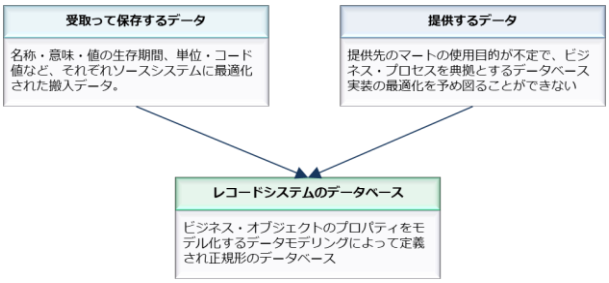


データソースシステム群から搬入されるデータは、名称・意味・値の生存期間、単位・コード値など、すべて、それぞれのシステムに最適化されたものである。

しかしながら、レコードシステムでは、各ソースシステムや、そのソースシステムを所有するビジネス・エンティティに依存しない形で、データを保持しなくてはならない。

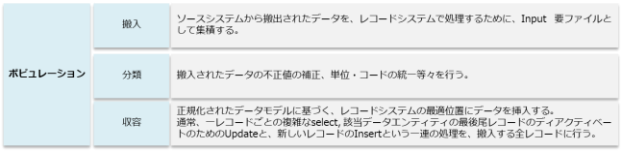
さらに、レコードシステムのデータは、その提供先のマートの使用目的が不定である。したがって、ビジネス・プロセスを典拠とするデータベース実装の最適化を図ることができない。

このため、レコードシステムのデータベースは、現実のビジネス・オブジェクトのプロパティをモデル化するデータモデリングによって定義しなければならないし、正規形として定義されたこのモデルを非正規化する根拠が存在しない、正規化されたままのデータベースとして実装されるのである。



2.3 ポピュレーション

データソースシステム群からレコードシステムへのデータの導入は、論理的には 3 段階に分かれる。これを総称してポピュレーション(データ移植)と呼んでいる。



一般に、ポピュレーションは大変時間のかかる処理となる。

また、このことは、処理中の障害によるバックアウトも大変な時間を要することを意味する。

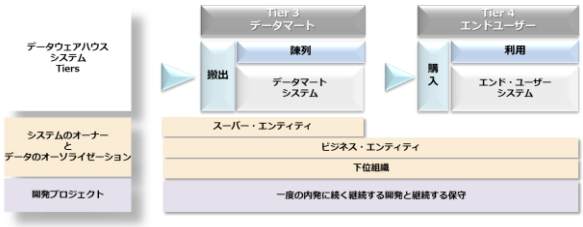
さらに、データベースの障害にともなうフォワードリカバリーを行う場合、蓄積されたポピュレーションは、処理時間を大きく要する。

データベース回復のためのシステムは、このポピュレーションを十分に考慮に入れて、設計しなくてはならない。

2.4 データマートとエンドユーザー

データマートとエンドユーザーは、合わせて単一のものとする場合とそうでない場合など、マートデータベースの特性とエンドユーザーの使用する処理系に応じて、形が変わるものである。

また、データマートから直接帳票が作られる場合などは、実装上の Tire4 エンドユーザーシステムは存在しない、など、多彩な実装が行われる。



ただし、ここで特徴的なのは、Tire3, Tire4 ともに、次々とダイナミックに発生する案件のために、継続的な開発が求められる、という点である。

継続開発のリソース配分が行われない場合、データウェアハウスそのものが滅亡する原因ともなる。

データマートのデータベースは、後続の処理に最適化されたデータベースである。

通常のデータベース設計によるものもあるが、処理系が要求する固有の方法論や固有のデータ編成を要求されるものもある。

これらは、データの応用とそれを行う特定の処理系を前提にしたものなので、むやみに一般化して使用するのは避けるべきである。

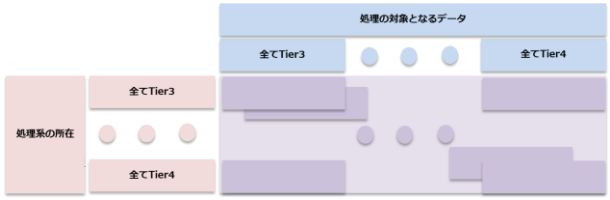
データベースは、システム資源として厳密な管理を受けるものであるが、同時に、柔軟な新設・変更が行えるような、手続きと資源配置を行っておく必要がある。

マートに使用される。 データモデルの例	プラットフォームの DBMS
特別のモデルを持たない	RDB
スター・スキーマ	RDB
多次元キューブ	RDBないしは 固有のDBMS

註 多次元キューブとスタースキーマは数学的には等値なモデルである。

Tier3, Tier4 において、どのような処理系をどう配置するかは、一般的に決められるものではない。

使用するツールとシステム環境、ビジネス環境、管理と予算のスキームによってさまざまな形態をとりうる。



2.4 辞書

データウェアハウスにおいて、その永続的な活用と成長のためには、辞書は極めて重要である。

辞書は、単一のシステムとして、全ての関係者がそれぞれの目的のために参照できるものでなくてはならない。

くく

辞書には、

データウェアハウスに関与するすべてのシステムのデータ・エンティティとアトリビュートの名称と意味記述、

同じく、そのデータ・エンティティとアトリビュートの由来と適用されているアルゴリズム、

同じく、それをオリジンとする、後続 Tier における、データ・エンティティとアトリビュート、

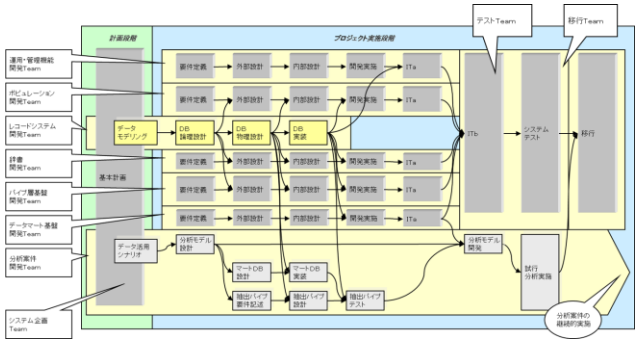
の記述を含んでいる必要がある。

また、データウェアハウスの中心となっているレコードシステムは、そのデータモデルとデータ・エンティティ、アトリビュートの意味記述は、全ての関係者に共有されるべきものである。

データ辞書にはこのデータモデルをグラフィカルに表示して、上記内容を検索できることが、求められる。

3．開発プロジェクト

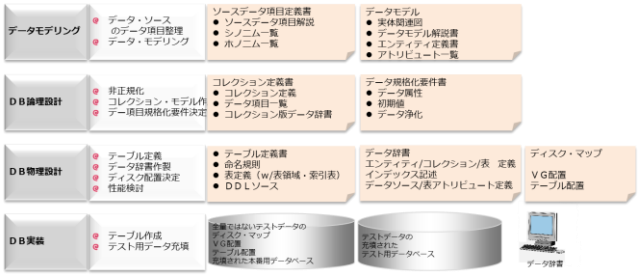
プロジェクトの設計は、その前提、環境で、大きく変わりが、典型としては、計画段階、プロジェクト実施段階の中で、開発局面を設定することができる。開発対象分野は、データウェアハウスの各 Tier とデータウェアハウスの運用・管理機能である。開発対象分野毎にそれぞれの開発 Team の設定が必要である。データマート、エンドユーザーの Tier は、サービス・イン後も、データマートの開発、新しい分析モデルやキャンペーンの開発とそられの実使用という、開発・実施が案件単位で続く。



データウェアハウス・システムの中で、ユニークでアプローチが難しいものがレコードシステムのデータベースである。

データモデリングの知識なしでこの定義を行うのは無理である。

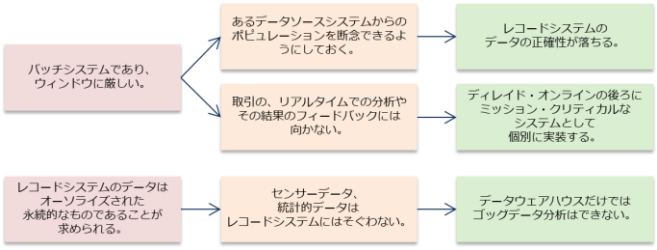
また、前の図のように、プロジェクトと実施より前にモデリングが開始される点も、特異な点といえる。



4．データウェアハウス・システムの限界と補完的メカニズム

データウェアハウスは、データ整備のための時間的ウィンドウに関して、常に厳しい状態に置かれる。また、データソースシステム群は厳格なシステムであり、レコードシステムに当座のデータを含むことは考えられていない。

このため、一定の誤差の許容や、補完的なシステムの装備などが必要になることがある。



以上