White Paper

embarcadero®

マーケットが XML、NoSQL、ビッグデータ、クラウドにシフトする中、データ モデリングはまだ重要か？

O'Kelly Associates ウェビナーの関連文書

Joe Maguire、Peter O'Kelly 共著

エンバカデロ・テクノロジーズ

〒102 – 0072 東京都千代田区飯田橋 4-7-1 ロックビレイビル 8F

TEL 03-4577- 4530 FAX 03-6843-0961

目次

はじめに

ここ数年、データベース マーケットにおけるいくつかの動的要素の影響で、多くの人々がデータ モデリング[1] の有用性に疑問を持つようになりました。特に、XML 情報管理の出現、従来のリレーショナル データベース管理システム（RDBMS）の機能およびベンダとの関係への高まる不満、クラウド プラットフォームの影響の拡大により、多くの組織がデータ モデリングの実践への取り組みを見直すようになりました。

その他にも、NoSQL システムやビッグ データ システムといったデータベース マーケットの動的要素によって、多くの人々がデータ モデリングの価値を見直す結果となりました。このドキュメントでは、NoSQL やビッグ データの有望な役割が今日広く誤解されていると筆者らが考えている理由をこれから説明しますが、定義のあやふやなこれらの領域にマーケットが熱心になっていることで、従来のデータ モデリングありきの考え方に疑問が投げかけられたことは間違いありません。

全体的に見て、データ モデリングは従来にも増して重要になっており、データベース マーケットの動的要素を十分に活用しようとする組織ではデータ モデリングへの重点的な取り組みを強化する必要があると筆者らは考えています。このドキュメントの残りの紙面では、筆者らの見方がどのような論理で成り立っているかについて説明します。

情報管理の枠組み

ここ数年、インターネットを中心にしたコンピューティングへの転換やクラウド プラットフォームの出現など、情報管理において多くの技術革新があり、さまざまな組織が、開発成果によって可能となった新たなチャンスを活かそうとしています。ここで、マーケットのさまざまな動的要素が互いにどう関係しているかを理解するのに使用可能な枠組みについて紹介したあと、次のセクションでは、マーケットの最も重要な動的要素を概観します。この枠組みは、現在の

[1] このドキュメントで言及しているデータ モデリングは、従来の（一般にリレーショナル）モデリングに限りません。エンタープライズ コンテンツ管理や Web コンテンツ管理などの領域にも当てはまります。このようなより広いドメインは、"情報モデリング" と呼んだ方が正確でしょう。これには、データ モデリングとさまざまな種類のコンテンツ モデリングの両方が包含されます。

マーケットにおける動的要素とそれらが全体としてデータ モデリングの役割にどう影響するかを理解するうえで、きわめて重要な役割を担っています。

リレーションとリソース：二分論か連続論か

枠組みの第 1 次元では、リソースおよびリレーションという 2 種類の情報項目の区別を扱います。リソースは、（たとえば、話を共有するために）話の流れを伝えるのに最適化されたデジタル成果物であり、通常は、談話、階層、系列の形式で構成されます。よく使用されるリソース タイプの例としては、書籍、雑誌、ドキュメント（たとえば、PDF ファイルや Word ファイル）のほか、Web ページや XBRL（eXtensible Business Reporting Language：拡張可能ビジネス レポート言語）レポートなどのハイパーテキスト文書があります。話を非同期的に共有する手段として、リソースは、人間が理解できるように最適化されています。

これに対して、リレーションは、現実世界の事物とそれらの事物同士の関係をアプリケーション独立な形で記述したものです。その例としては、顧客モデル、販売モデル、人的資源モデルなどのよく知られたデータベース ドメイン（対象領域）があります。リレーションは、アプリケーションやツール（たとえば、クエリ/レポート ツールなど）で使用するためのもので、そのようなツールがない場合は、話を伝えたり、特定のドメインに対する人間の理解を高めるのに役立つことはまれです。

図 1 は、リソース/リレーションの連続スペクトルを表しています。

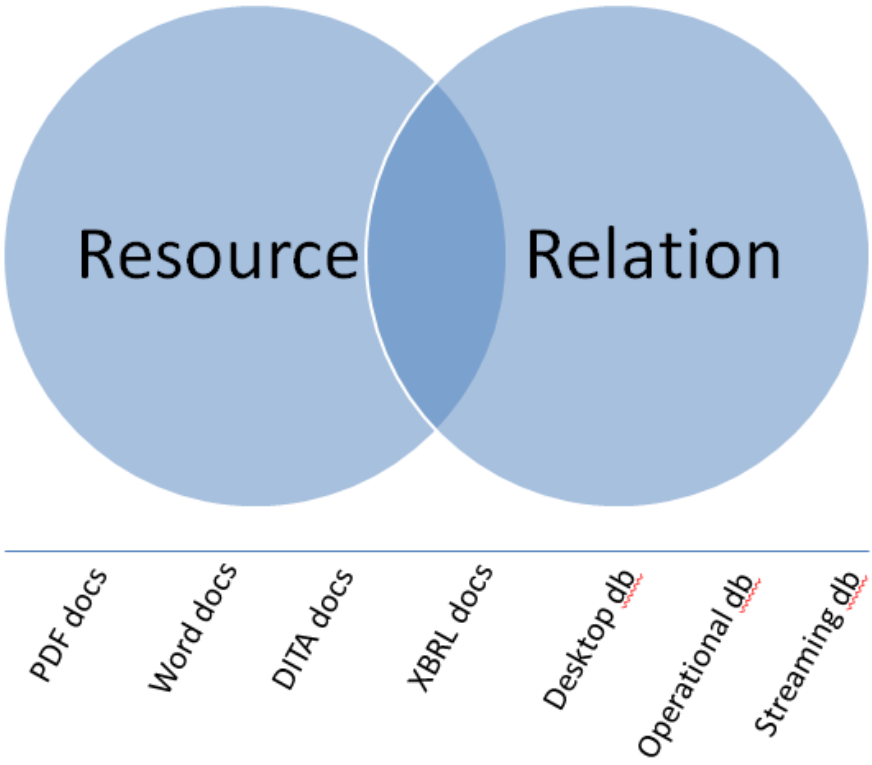


図 1：リソース/リレーションの連続スペクトル

図 1 に示したモデルは正確さや網羅性を意図したものではありませんが、リソースおよびリレーションのさまざまなカテゴリ間の相違点を説明するうえでは役に立ちます。

リソースとリレーションは相補的です。たとえば、インターネットベースの注文処理システムには、リレーション中心のデータベースと、発注詳細を記述した XML ドキュメントなどのリソースが両方とも含まれている可能性が高く、それに加えて、認証目的に使用されるデジタル署名などの、ドキュメントを中心としたその他の関心事も含まれていると思われます。

関心事の分離：概念的、論理的、物理的

枠組みの第 2 次元では、次のような、モデリング抽象化の 3 つの相補的なレベルを扱います。

- 概念レベル：技術に依存せず、主として、モデリング ドメインの利害関係者間で文脈的合意を確立するのに役立てるために使用されます。情報労働者は原則的に抽象化の概念モデル レベルで作業をします。
- 論理レベル：概念モデルを技術的に表現します。今日最も広く使用されている論理モデルには、ハイパーテキスト（たとえば、Web ページがそうですが、XBRL レポートのよ

うにもっと複雑な場合もよくあります）とリレーショナルの 2 種類があります。アプリケーション開発者は原則的に抽象化の論理レベルで作業をします。

- 物理レベル：インデックス作成やフェデレーションなどの実装レベルの詳細で構成されます。モデリングの物理的関心事に関与するのは、原則的にシステム アーキテクトとシステム管理者に限られ、情報労働者やアプリケーション開発者は物理モデルの詳細に頭を悩ますべきではありません。

大半の組織では現在、論理モデルと物理レベルを組み合わせたものを扱っています。概念モデルは、現在では、広く使用または理解されているわけではありません。図 2 には、2 つの概念モデルの例が一部だけ示されています。一方は Evernote のもので、もう一方は Microsoft OneNote のものです。どちらもよく知られている個人情報管理ツールです。

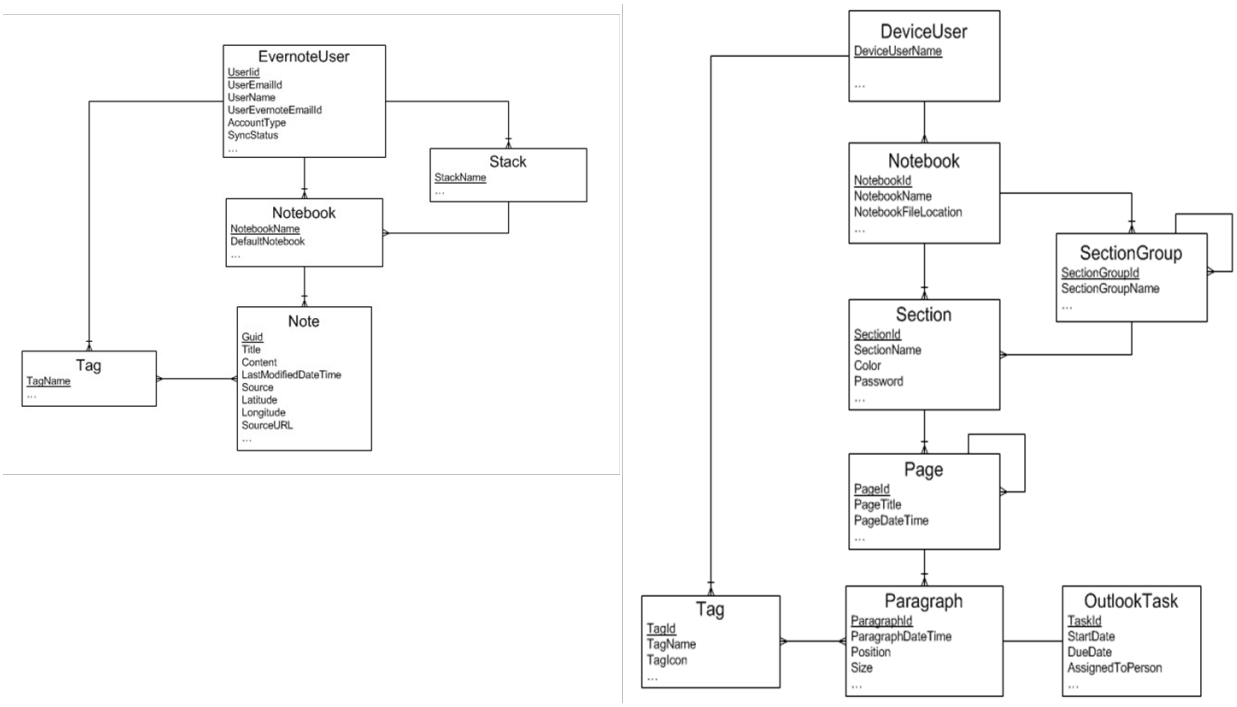


図 2：概念モデル図の例

たとえモデリング手法のトレーニングを受けていなくても、図 2 に示した例を見れば、OneNote の包含モデルの方が Evernote より複雑であることと、OneNote ではサブページ/パラグラフ レベルでのタグ付けをサポートしているのに対して、Evernote のタグ付けはノート/ページ レベルであることが容易に見て取れます。

概念モデルの規約は論理モデリング ツールの経験がある人にはおなじみのものに見えるかもしれませんが、外部キー リレーションシップ、データ型、ヌル制約などの詳細事項はその中に含まれていないことに注意してください。概念モデルは、モデリング ドメインの利害関係者間で文脈的合意を確立するのに役立つよう最適化されます。そのため、エンティティ、属性、リレーションシップ、識別子に的を絞っており、特定の論理モデルを前提とすることもなければ、ビジネス ドメインを専門に扱っている人々を散漫にしたり混乱させるおそれのある低レベルの詳細事項が含まれることもありません。

図 3 は、統一された枠組みにおける次元を表しています。

	Resources	Relations
Conceptual	Resources and links	Entities, attributes, relationships, and identifiers
Logical	Model: hypertext Language: XQuery	Model: extended relational Language: SQL
Physical	Indexing (e.g., scalar data types, XML, full-text), locking and isolation levels, federation, replication, in-memory databases, columnar storage, table spaces, caching, and more	

図 3：統一された枠組み

この枠組みについては、この後のセクションで NoSQL やビッグ データといったデータベース マーケットの動的要素を概観するときに再び取り上げます。

アプリケーション、データ、アプリケーション/データ独立性

枠組みの最後の次元である第 3 次元はアプリケーションとデータの区別で、これを考慮することが重要です。データ管理はもちろん情報だけの話ではありません。情報があっても、それを役立てるアプリケーションや分析ツールがなければ、アーカイブにすぎません。ただし、情報管理の対象範囲はアプリケーション要件に左右されるとはいえ、データベースが必ず複数のア

プリケーション ドメインに役立つようにするには、アプリケーション/データ独立性が重要な目標になります。

一部のデータベース シナリオでは、情報モデリングに対する過度の影響と筆者らが見なしているものがアプリケーション開発者の好みに含まれている場合があります。たとえば、プログラマの生産性の最適化ばかり考えている組織では、プログラミング言語フレームワーク（通常はオブジェクト指向パラダイムを具現化したもの）と情報格納に関する事項（通常は XML や SQL）とのいわゆる "インピーダンス ミスマッチ" を最小限に抑えようとする可能性があります。極端な場合、このようなアプローチでは、情報管理がきわめて複雑になるおそれのある "ファイルがプログラムに属する" という考え方につながり、事実上、DBMS が登場する前の時代に戻ることになります。データベースの専門家には矛盾しているように思われるでしょうが、このアプローチは、データ モデリングに関する問題を洗い直す際に検討しなければならないよくある情報管理パターンであり、これについては、データベース マーケットの動的要素を概観する際に再び取り上げます。

まとめ：情報管理の枠組みの価値

リソースとリレーションの相補的な性質に注目した枠組みを扱うと共に、概念モデル、論理モデル、物理モデルを区別することにより、現在のデータベース マーケットの動的要素が互いに対してどう適合するかを探ることが可能です。このような枠組みを扱わないことにする組織には、以下のリスクがあります。

- データベース システムおよびモデリング ツールに関して何をいつ使用するかがはっきりしないため、アプリケーション開発者が、ドメインに最も合うと思われるものではなく、自分に最もなじみのあるまたは使い心地の良い手法やツールを使用する結果となることが多い。
- 文脈的合意がないため、情報モデルを中心とした実際の問題点の伝達ミスや誤解に基づいた対立が発生するおそれがある。
- 以下の事項への注目が不十分になる。
    - アプリケーション/データ独立性：これが重視されないと、DBMS が登場する前の時代の "ファイルがプログラムに属する" という考え方のアプローチにうっかり戻ってしまうおそれがあります。

    - 概念モデル、論理モデル、物理モデルの独立性：これが重視されないと、情報労働者やアプリケーション開発者が実装に関する低レベルの検討事項に頭を悩ます結果となり、生産性が低下するおそれがあります。
- リレーショナル DBMS と XML DBMS を持続的かつ相補的な形で適合させることは、現在のデータベース マーケットにおける動的要素の中核を成すにもかかわらず、それの重要性を正しく認識できないおそれがある。

データベース マーケットの動的要素

このセクションでは、現在のデータベース マーケットの動的要素を概観します。

RDBMS 再考

この 10 年間は、大手の商用 RDBMS ベンダにとって力量が試される年月でした。これらのベンダは、RDBMS の旧態依然とした規範に対する IT 側の不満が広まったことで、いくつかの点で絶えず非難にさらされているように見えます。それは 1 つには、RDBMS ベンダの時に非生産的なポリシー（たとえば、評判の良くない価格設定やライセンス ポリシー）や顧客サービスに原因があります。DBA は過度に保守的か管理志向であると多くのアプリケーション開発者が信じているため、データベース管理者（DBA）にとっても努力を必要とする日々が続いています。

以下のように、RDBMS の機能について長年抱かれてきた仮定への課題もいくつか出てきています。

- "Web スケール" コンピューティング（たとえば、Facebook、Google、Twitter といった世界規模のサービスで必要とされるもの）への RDBMS の適合
- 急速に変化するアプリケーション ドメインに対応するのに不可欠な柔軟性または "アジリティ"（俊敏性）
- プログラミング フレームワークと SQL の境界およびプログラミング フレームワークと XML の境界（このケースがますます増えている）に対する長年の不満についてのアプリケーション/データ "インピーダンス ミスマッチ"

これらのダイナミクスにより、オープン ソース DBMS や専用 DBMS（たとえば、個別の

DBMS 型が "Web スケールの" 解析や情報ストリーム処理用にプロモートされるものなど）の出現とも相まって、一部には、"万能型データベースの終焉" つまり従来の RDBMS アプローチの終焉であると断言する人々も出てきました。

情報管理の技術革新にはまだ十分な余地がありますが、主要な RDBMS 商用製品やオープン ソース プロジェクトは非常に強力かつ柔軟であり、たとえば主流による大容量メモリ サーバーや SSD（固体ディスク）ストレージの採用などにより急速に発展し続けると筆者らは確信しています。

ただし、それでも RDBMS の現ユーザーや支持者は現在、これまでにない（技術と政治の両面での）課題に直面しており、それらの課題は、自らの否定的な経験（RDBMS の効果的な対処能力ではなく RDBMS の使用方法に関係していることが多い）のために、時には、不満を抱いているアーキテクトや開発者の共感を呼ぶことがあります。

図 4 は、先ほど紹介した枠組みのどこに RDBMS が当てはまるのかについての筆者らの考え方をざっと示しています。

	Resources	Relations
Conceptual	Resources and links	Entities, attributes, relationships, and identifiers
Logical	Model: hypertext Language: XQuery	Model: extended relational Language: SQL
Physical	Indexing (e.g., scalar data types, XML, full-text), locking and isolation levels, federation, replication, in-memory databases, columnar storage, table spaces, caching, and more	

図 4：統一された枠組みにおける RDBMS の位置づけ

XML 情報管理のための XDBMS

XML 情報管理への DBMS の利用は、データベース マーケットのもう 1 つの重要な動的要素となります。XML 情報管理により、XML Schema、XPath、XSLT、XQuery、SQL/XML などの関連する業界標準に基づいて、いくつかの重要な機会が提供されます。

XML を中心とする情報管理システムは、XML ハイパーテキスト文書（リソース）と XML データ（リレーション）の両方を扱うのに役に立ちます。リソースの場合、XML システムは、独自に開発した以前のコンテンツ/ドキュメント管理システムを置き換えるのに主として役に立ちます。XML データの場合、システムは、XML がアプリケーション間での構造化データ交換のシリアル化に使用されるシナリオに主として役に立ちます。

XML リソースと XML リレーションの関心事の区別を十分理解していないために、相当な混乱が生じるおそれがあります。すべての XML 情報管理ドメインを同じように扱うことは、1980 年代後半に起こった "オブジェクト データベース" 製品の不運な盛り上がりに関連する "何もかもがオブジェクト" 的なアプローチと同じくらい役に立ちません。その種の過剰な一般化が、情報管理への影響について十分な検討も行われずにプログラマの好みによって突き動かされている場合には、特にそうです。

今日、XDBMS（XML DBMS）のアーキテクチャに対する一般的なアプローチは以下の 2 とおりあります。

- 主要な商用 RDBMS に XML モデル管理が付加されたハイブリッド型マルチモデル DBMS。これは、主要な商用 RDBMS ベンダである IBM、Microsoft、Oracle によって推進されたアプローチです（実際には、これら 3 ベンダはいずれも現在ではマルチモデル管理システムを提供しているので、これらベンダの製品の最新リリースを RDBMS に分類するのは誤った名称ですが、RDBMS カテゴリの関連はそう簡単には変わりません）。これらのベンダは、シームレス SQL/XML や XQuery サポートなどの XML 機能にかなりの研究開発資源を注ぎ込みました。
- eXist オープン ソース プロジェクトや商用の MarkLogic Server 製品を例とする専用 XML DBMS。現代の XML DBMS は、市販のサーバー ハードウェア上で大規模なスケーラビリティを実現できるように設計されています。

図 5 はハイブリッド アプローチの例（IBM DB2 9.x）です。

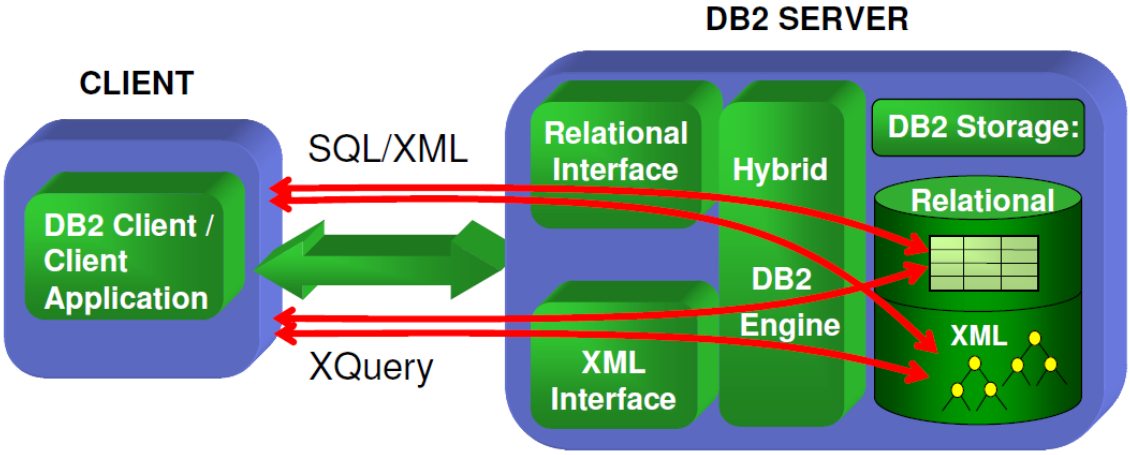


図 5：ハイブリッド DBMS（出典：IBM）

この図で示唆されるように、ハイブリッド アプローチによって、リソース指向の開発者はリレーショナル情報と XML 情報の両方を扱いながら XQuery を使用できるようになるのに対して、リレーション指向の開発者は XML 拡張機能を備えた SQL を使用してそうした情報を扱えます。

図 6 は、統一された枠組みのどこに XDBMS が当てはまるかを示しています（ハイブリッド X/RDBMS 製品は、図 4 と図 6 に示した領域の複合領域を扱います）。

	Resources	Relations
Conceptual	Resources and links	Entities, attributes, relationships, and identifiers
Logical	Model: hypertext Language: XQuery	Model: extended relational Language: SQL
Physical	Indexing (e.g., scalar data types, XML, full-text), locking and isolation levels, federation, replication, in-memory databases, columnar storage, table spaces, caching, and more	

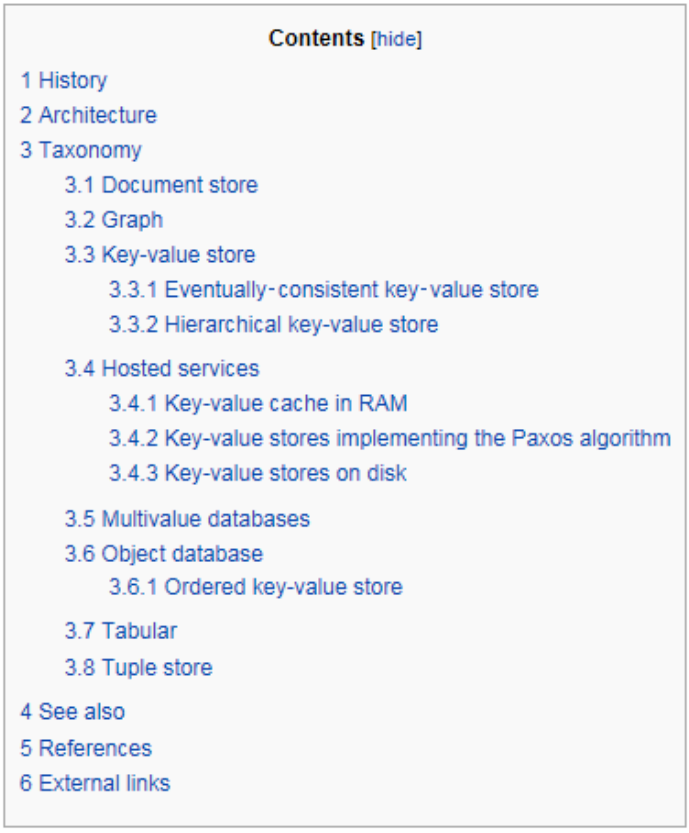
図 6：統一された枠組みにおける XDBMS の位置づけ

NoSQL

NoSQL は非常に興味深いデータベース マーケット事象ですが、それは 1 つには、"NoSQL" の意味について明確な合意が得られていないためです。業界主導のプロジェクトとして、NoSQL では、まず、それが何であるかではなく何でないのかを定義します（SQL ベースでない）。それでも、NoSQL の支持者たちは、RDBMS での旧態依然とした経験への不満から、意見を聞いてくれる相手を見つけては自分の主張を繰り広げます。

NoSQL も急速に発展していることに注意することが大切です。これは最初 "SQL への拒否" を勧める活動として受け取られましたが、ここ数年間で、RDBMS に対して競合する立場ではなく相補的なものであることをもっと強調して、"Not Only SQL"（SQL だけではない）であると過去にさかのぼってそっと再定義されました。

図 7 は、NoSQL の Wikipedia 記事から抜粋した、NoSQL システム タイプの分類です。


Contents [hide]

1 History
2 Architecture
3 Taxonomy
- 3.1 Document store
- 3.2 Graph
- 3.3 Key-value store
  - 3.3.1 Eventually-consistent key-value store
  - 3.3.2 Hierarchical key-value store
- 3.4 Hosted services
  - 3.4.1 Key-value cache in RAM
  - 3.4.2 Key-value stores implementing the Paxos algorithm
  - 3.4.3 Key-value stores on disk
- 3.5 Multivalue databases
- 3.6 Object database
  - 3.6.1 Ordered key-value store
- 3.7 Tabular
- 3.8 Tuple store

4 See also
5 References
6 External links

図 7：Wikipedia 記事に示されている NoSQL の分類

記事で明らかにされている 8 種類のシステムを概観することは、このドキュメントの範囲を越えていますが、さまざまなタイプがこのドキュメントでこれまで論じてきたトピックにどう対応するかについての筆者らの考え方を図 8 に示しておきます。

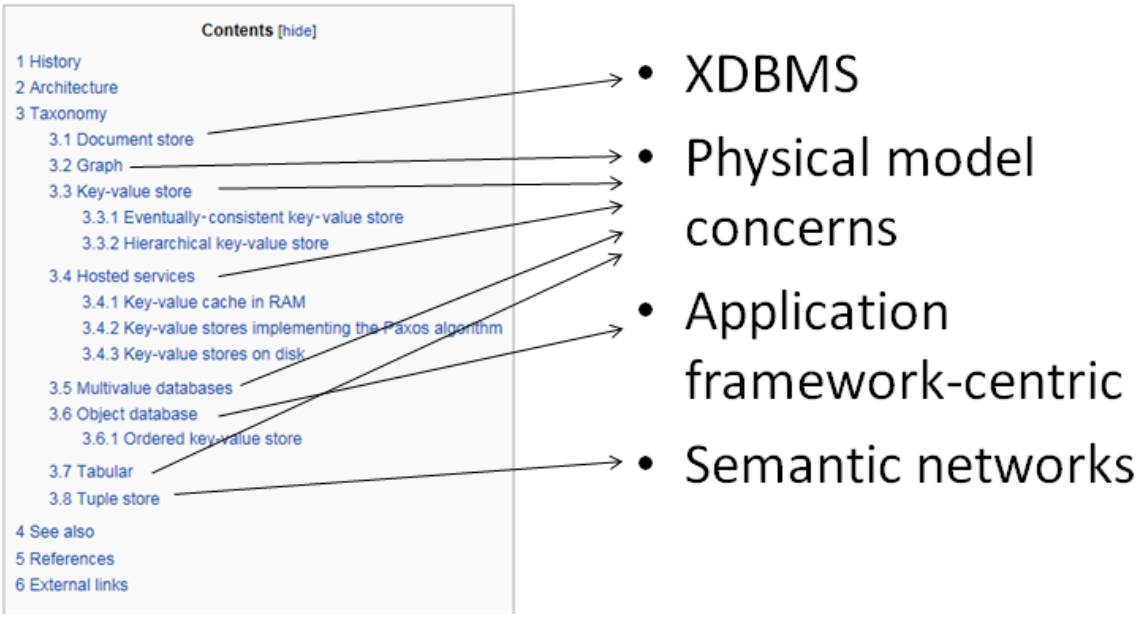


図 8：NoSQL コンセプトの対応関係

図 8 で示唆されるように、NoSQL ミームは以下のものが雑然と融合したものです。

- XDBMS アプローチで最もよく満たされるドキュメント ストア要件（さらにややこしいことに、XDBMS は NoSQL システムのサブクラスと見なされます）
- DBA やシステム アーキテクトが専念すべき物理モデル関心事（RDBMS や XDBMS に競合するというよりもむしろそれらと相補的）
- オブジェクト データベース：勢いを増すべく数十年間にわたって奮闘してきましたが、それでも、"インピーダンス ミスマッチ" が最小限に抑えられるという認識と内在する情報管理上の危険性の軽視から、これに夢中になるアプリケーション開発者も一部にいます。
- 意味モデル：このドキュメントの範囲を越えていますが、一般に、これらも、RDBMS や XDBMS に競合するというよりもむしろそれらと相補的です。

ビッグ データ

"ビッグ データ" は、明確に定義されていないもう 1 つのマーケット動的要素で、多くの人々に、

NoSQL のサブクラスであると見なされています。ビッグ データは一般に、以下のアプリケーション ドメインと関係があります。

- 情報ストリーム処理シナリオ（たとえば、携帯電話使用量、センサ ネットワーク、Web サイト ログの解析など）
- 大規模データ解析（たとえば、Google が自社のビジュアル検索技術と音声認識技術を改良するために適用しているものなど）

Hadoop もビッグ データとよく関係しています。Google の MapReduce や Google ファイル システムなどのプロジェクトに端を発する Hadoop は分散処理用の Apache アーキテクチャで、通常、市販ハードウェアのネットワークを使用する場合に関係してきます。

データ ウェアハウスやビジネス インテリジェンスの分野を専門に扱っている主要ベンダの多くは、顧客に提供する価値を増やす手段として Hadoop を利用しています。ただし、全体としては、Forrester Research 社が『Stay Alert to Database Technology Innovation』（データベースの技術革新に常に注意を!）と題する 2010 年 11 月のレポートで述べているように、「BI ユースケースの 90% については、サイズが 50 テラバイト未満であることが多く、リレーショナル データベースでまだ十分である」ということです。

図 9 は、このドキュメントで既に述べた枠組みにおけるビッグ データの位置づけについての筆者らの考え方を示しています。

	Resources	Relations
Conceptual	Resources and links	Entities, attributes, relationships, and identifiers
Logical	Model: hypertext Language: XQuery	Model: extended relational Language: SQL
Physical	Indexing (e.g. scalar data types, XML, full-text), locking and isolation levels, federation, replication, in-memory databases, columnar storage, table spaces, caching, and more	

図 9：統一された枠組みにおけるビッグ データの位置づけ

言い換えれば、ビッグ データは（残りの NoSQL タイプ分類の大半と同様に）、主として物理モデルの関心事に的を絞っており、RDBMS や XDBMS のアプローチに競合するというよりもむしろそれらと相補的であると考えています。

クラウド

このドキュメントで最後に取り上げる、データベース マーケットの動的要素は、クラウド データベース プラットフォームの出現です。主要なクラウド データベースは、さまざまな RDBMS、XDBMS、ファイル システム指向の代替手段で構成されており、たとえば以下のものがあります。

- Amazon の SimpleDB および RDS
- Google の BigTable および MegaStore
- Microsoft の SQL Azure

"サービスとしてのプラットフォーム" 製品には一般に、複数のデータベース サービスが選択肢として含まれています。

クラウド データベース サービスの創出の結果、データベース アーキテクチャの重要な技術革新がもたらされ、その革新の大半は従来型 RDBMS の今後のリリースにも適用されつつあります。ただし、全体としては、クラウド データベースは、データベース モデルの別個の代替案というよりもむしろ配置オプションを表していると筆者らは考えています。クラウドはまた、多くの場合、 財務会計主導の決定事項になります。クラウド コンピューティングは、資本設備集約的な能力を営業経費に転換するのに使用できるからです。

データベース マーケットの動的要素：まとめ

データ モデリングについての関連検討事項の概観に移る前に、筆者らの考えを以下にまとめておきます。

- 以下の技術の間には、実質的、持続的、相乗的な関係があります。
  - RDBMS
  - XDBMS
  - Hadoop
  - 情報管理プラットフォームとしてのクラウド
- NoSQL とビッグ データの両マーケット要素は、相対的に言って、明確に定義されておらず、一時的で、宣伝過剰です。

情報モデリングの役割の再検討

現在のデータベース マーケットの動的要素は、対応するだけの価値があります。つまり、新たな現実は新しい振る舞いを引き起こすはずです。ただし、実証済みの従来の振る舞いを行き当たりばったり捨て去るべきではありません。情報モデリングの長年にわたる長所の多くは、依然として有効なままです。

古くから変わらない現実

情報管理マーケットにおける最近および近い将来の変化は重大かつ心躍るものですが、情報モデリングについての基本的現実の大半は不変です。場合によっては、これらの基本的現実を尊重することがかつてないほど重要になることがあります。

関心事の分離

関心事の分離は、設計の労力とその結果得られる成果物を区分しようとする取り組みです。関心事の分離では、概念モデリング、論理モデリング、物理モデリングを区別します。これら 3 つの形態のモデリングは、目標も参加する要員も必要なスキルも異なるため、設計者と要件分析者はこれらを区別しなければなりません。

同様に、関心事の分離では、データとプロセスを区別します。

データとプロセスの区別

情報モデル作成者と要件分析者がデータ要件とプロセス要件を区別することは依然としてきわめて重要です。それができない場合は、広範囲にわたる悪影響が生じます。つまり、アプリケ

ーション/データの独立性が損なわれたり、データ モデルとそれらのベースとなるデータベースが早々に陳腐化したり、不正確なデータ モデルに起因して情報の品質が低下したりするのです。

情報管理における新たな現実によって新たな誤解が既に生じているため、このベスト プラクティスは、改めて留意するだけの価値があります。特に、データ モデリングをメッセージ モデルの設計に置き換えてしまった設計者も一部にはいます（たとえば SOA システムの場合など）。まず第一に、メッセージ モデルは処理成果物であり、データフロー図の一部を表現したものです（つまりはプロセス モデルなのです）。メッセージ モデルの設計は重要ですが、データ モデリングに取って代わるものではなく、データ モデリングと同じ恩恵をもたらすわけではありません。

文脈的合意

モデル作成が概念データ モデリングの主な価値を認識していることは、依然としてきわめて重要です。概念データ モデリングは、情報の意味、つまり、どのようなカテゴリが存在するか、それらのカテゴリにどのよう特性が当てはまるか、それらのカテゴリにどのような相互関係があり得るか、任意のカテゴリのメンバが互いにどう区別されるか、それらのカテゴリ、特性、関係の命名にどのようなビジネス用語が使用されるかといったことについて文脈的合意を確立するプロセスです。

情報管理における新たな現実を誤解すると、概念データ モデリングが損なわれる（あるいは、うっかり完全に削除される）おそれがあるため、このベスト プラクティスは改めて留意するだけの価値があります。NoSQL 要素の下にひとまとめにされた手法の多くは、物理データ モデリングの性質を根本的に変えます。ただし、物理データ モデリングの性質が変わっても、必ずしもそれが、概念データ モデリングの同等の変化に変換されるわけではありません。概念的な合意は "あれば便利だが、なくても支障がない" というようなものではなく、以下のように、不可欠なものです。

- アプリケーション/データ独立性の要である：データを共有する異なるアプリケーションの開発者およびユーザーは、そのデータの意味について合意する必要があります。

- ビジネス ルールの表現の基礎となる：概念データ モデルで定義された名詞は、処理やポリシー要件を形式的に記述するビジネス ルールのオペランドとなります。

ユーザーが認識するメタモデル

モデル作成者、IT ストラテジスト、ソフトウェア アーキテクトは、ユーザーがさまざまな概念メタモデルに精通していることを認識している必要があります。ユーザーは空腹時には一方のメタモデルからもう一方に切り替えることさえできます。田舎の典型的な簡易食堂のメニューには、以下の 4 つのメタモデルが含まれています。

- 物語データ（たとえば、3 世代前に作られた簡易食堂の風変わりで面白い歴史など）
- 構造化データ（たとえば、名前、数量、説明、価格、提供された時刻が記述された食品項目など）
- 画像データ（たとえば、数年前の簡易食堂の絵のように美しいエッチングなど）
- 空間情報データ（たとえば、簡易食堂に至る地方道路を示す地図など）

メタモデルがユーザーの期待にそぐわなければ、システム設計者には、使い勝手の良いメタモデルを選ぶ権限はありません。

情報管理における新たな現実によって、これまでデータとコンテンツを隔ててきた理論上の壁が崩れるので、このベスト プラクティスは改めて留意するだけの価値があります。そうした壁が崩れると、データとコンテンツは、主としてエンタープライズ組織の利益に合うように混合されます。ただし、このような混合は、ユーザーが認識する概念メタモデルと開発者が選んだ実装メタモデルとの親和性について慎重に考えている開発者とアーキテクトに有利に働きます。同様に、ユーザー ニーズではなく実装上の利便性に基づいてメタモデルを選ぶ開発者は、データの品質が低下したりユーザーが混乱する原因を生むことになります。

ユーザーが好むメタモデルは要件分析フェーズで明らかになることに注意してください。言い換えれば、モデリング表記法が特定のメタモデルを仮定している場合には、概念データ モデリングはうまく機能しません。

注目に値する新たな現実

データ モデリングに関する基本の大半は不変であるとはいえ、データベース マーケットの新たな現実の影響は多少あります。

クラウド コンピューティング向けのデータ モデリング

クラウド アプリケーションでは、物理データ モデリングの必要性は低くなる可能性があります。エンタープライズ組織がデータベースの設計と運用の負荷を外部のクラウドに移す場合、その組織は、テーブル空間やインデックスといった物理設計上の決定をもう行わなくてもよくなる可能性があります。

ただし、だからと言って、エンタープライズ組織が概念モデリングの責務から解放されるわけではないことに注意してください。"クラウド内データベース" 技術を使用しても、文脈的合意の必要性はなくなりません。事実、概念モデリングがぜいたくなことであり、既存のデータベースを稼働させ続ける必要性の方が一見差し迫っているように見えるため概念モデリングの優先順位は低いと見なした組織でも、今となっては、概念モデリングを適切に実行する時間があることに気づくでしょう。

概念モデリングと論理モデルの区別

情報管理における新たな現実によって、概念データ モデリングと論理データ モデリングの違いが明確に緩和されます。

良かれ悪しかれ（通常は悪い）、一部のエンタープライズ組織では、論理モデル手法を用いてユーザー データ要件を収集し明確に記述してきました。データ要件が論理メタモデルに合っていれば、辛うじてかつ不自然ではありますが、この方法でも何とかなります。最もよく使用されてきた論理メタモデルは、拡張リレーショナル モデルに近いもの、つまり、エンティティ-リレーションシップ モデリング、Barker 記法、IDEF1X などです。当然のことながら、概念データ モデリングは通常は、（リソースではなく）リレーションに適用されます。

ただし、情報管理における新たな現実の結果、アーキテクトやシステム設計者は、リレーションではなくリソースに最適化された論理記法（たとえば、HTML、XML Schema、あるいはそ

れらに類似したもの）を選択できます。非階層的な現象の場合、そのような表記法では、データ モデルが不正確になったり、ユーザー要件が不完全になります。

リレーションとリソースの両方についての要件を表現できる概念モデリング手法および表記法に合うように、要件分析者が自らの責務を更新することが適切な手順です。

パラドックス：データ モデリングへの過少投資

マーケットの動的要素の変化によって、IT ストラテジストは、データ モデリングの新旧の現実にとどまらず、それ以上のこともじっくり検討しようという気になるはずです。古くからの（および新たな）誤った考えについて考えてみるのも有益です。歴史的に見ると、そのような思い違いは、データ モデリングの過小評価や過小投資を招いてきました。

文明の歴史では、悪い考えは良い考えより圧倒的に数が多いものです。データ モデリングもまったく同じです。したがって、ここでは、データ モデリングに関する誤った考えをすべて列挙することはしません。ただし、以下に挙げるいくつかの思い違いは特に注目に値します。

- データ モデリングの価値は過小評価されることが多い。

  データ モデリングの質が悪かったり不十分だった場合の影響については、広く認識されていません。そうした影響がすぐには現れない可能性があることが、しばしばその理由となっています。たとえば、後でデータ品質上の問題によって、監督官庁に誤って通知され法外な罰金を支払わされる結果となったときに、こうした影響が明らかになるのです。

- データ モデリングによって、データ成果物よりもコード/プロセス成果物を重視する IT 組織の報酬制度が混乱する。

  IT 組織の報酬制度がデータ モデリングを行うことを奨励するようになれば、IT ストラテジストおよびアーキテクトはデータ モデリングに十分注意を払うようになります。これには、データ品質の尺度をコード品質の尺度と同じくらい重要なものにすることも含まれます。

長年にわたるこれらの誤った考えに加えて、以下のものもデータ モデリングを広く使用するう

えで妨げとなります。

- 誤ってデータ モデリングと呼ばれる活動が一部あり、IT ストラテジストが、実際には違うのにデータ モデリングを行っていると信じてしまうことがある。

  たとえば、メッセージ モデルに対応する XML スキーマを設計することは、プロセス モデリングであって、データ モデリングではない。

- データ モデリングが XML ベースの情報管理には重要でないと誤って信じている IT ストラテジストが一部にいる。

  データがリレーショナル データベースか XML ファイルのどちらで表現されようと、あるいはハンマーとのみで石の板に刻まれようと、文脈的合意は常に重要です。

まとめと推奨事項

情報管理に携わることは心躍る経験です。マーケットの新たな現実によって、長年にわたって IT 組織を悩ませてきたある種の問題に対する解決策がもたらされます。それでもやはり、IT 専門家は、有効性が実証済みのベスト プラクティスの多くを引き続き活用することができます。

ソフトウェアの能力、ハードウェアの能力、そしてそれに伴う価格/性能比の驚くべき向上によって、従来の RDBMS で解決できる問題の規模と対象範囲は広がります。こうした向上には、SSD（固体ディスク）技術や市販の大容量メモリ サーバーが含まれます。

配置オプションとしては、クラウド内データベースによって、作動部分を一部削除したり、内部の IT 組織から運用負荷を取り除くことで、情報アーキテクチャを簡単なものにすることができます。これで、物理モデル作成者の負荷は軽減され、より概念的なモデリングを実行できるようになります（ただし、スキルを伸ばす必要があります）。

簡略化へのもう 1 つの機会：リレーションとリソースの管理の調整に対する新しいアプローチにより、多くの作動部分（Web コンテンツ管理、エンタープライズ コンテンツ管理、ファイル管理）を XML ベースのデータ管理（XDBMS またはハイブリッド型のマルチモデル DBMS）

に置き換えることができます。さらに、SQL と XQuery の組み合わせにより、アプリケーション スタックを簡略化し、宣言的な開発を最大限に生かし、低レベルのスクリプト作成やプログラミングを最小限に抑えることができます。

Google の Map-Reduce パラダイム（特に Hadoop）の実装と近似により、市販ハードウェア ネットワーク上での分散処理用の実行可能かつ有用なフレームワークが提供されます。

マーケット要素 "NoSQL" はおそらく姿を消し、a) NoSQL に組み込まれているとされる個々の技術や手法がより統制された形で計画および配置され、b) 従来の RDBMS 技術が現代の情報管理の基盤であり続けることが、十分な情報に基づいて冷静に認められるようになります。

推奨事項：戦略的合意の確立

マーケットの新たな現実に呼応して、エンタープライズ組織では、戦略的な構想と長期的な計画を確立しなければなりません。この構想は、IT 組織だけでなく、エンタープライズ組織全体をカバーするものでなければなりません。それが、データ ガバナンス、情報品質、情報ライフサイクル管理、セキュリティおよびリスク管理、その他のビジネス関心事に影響を及ぼすからです。

計画は、技術を効果的に選択し利用するための目標を設定するものでなければなりません。これらの目標には、XDBMS やハイブリッド型マルチモデル DBMS を慎重に使用することで、Web コンテンツ管理、エンタープライズ コンテンツ管理、ファイル管理ツールを一気に置き換えることができるように、一部の技術を効果的に入れ替える作業が含まれていなければなりません。

計画では、ベスト プラクティスを練り直し、マーケットの新たな現実にもう当てはまらないものを再検討しなければなりません。さらに、IT 組織の報酬制度を明確化/改善して、ベスト プラクティスがプログラマ、モデル作成者、情報アーキテクトなどの IT 実務者に必ず尊重されるようにしなければなりません。

推奨事項：データ モデリングへの再注目

ベスト プラクティスには、データ モデリングが含まれていなければならず、モデリング固有

のプラクティスは、言語、文化、分類、技術の第 1 原則から記述し直す必要があります。モデリング ベスト プラクティスでは、関心事の分離に注意を払う必要があります。IT 組織の報酬制度では、モデリングを奨励し必須とする必要があり、IT の評価尺度は、コードの品質だけでなく情報の品質にも報いるように拡張されなければなりません。

推奨事項：リレーションとリソースとの相互作用の正しい認識

マーケットの新しい動的要素により、データ管理とコンテンツ管理を隔てる理論上の壁は取り払われます。その結果生じる、リレーションとリソースの混合により、新たな効率性がもたらされますが、それと同時に、間違いが発生する機会も新たに出てきます。こうした間違いを避けるには、IT ストラテジストおよび実務者の手で、リソースとリレーションとの相互作用に関する専門知識を練り上げる必要があります。こうした専門知識はまず、基本の域を越えた以下のようなメタモデルを綿密に調べることから始まります。

- 拡張リレーショナル モデル（お気に入りの DBMS ベンダで実装されているモデルに限らない）
- ハイパーテキスト情報モデル（HTML や XML に限らない）

推奨事項：XQuery の習得と活用

多くの場合、リレーションを管理するためのソフトウェア工学ベスト プラクティスは、リソースの管理にもうまく当てはまります。例を 1 つ示しましょう。宣言的プログラミングは、手続き型コードに適用するのが望ましいです。XQuery は、XML データに対する宣言的なクエリを記述するのに選択できる言語です。情報管理の新たな現実を十分に活かしたいと考えている IT 実務者は、XQuery を習得する必要があります。

詳細情報

データベース マーケットの主要な動的要素と、データ モデリング、情報管理、コラボレーションに対するそれらの影響の詳細については、エンタープライズ組織が新たな機会を十分に活

用する上で役に立つワークショップやコンサルティング サービスの情報も含めて、www.okellyassociates.com にアクセスしてください。

言語、文化、分類、技術の第 1 原則をはじめとして、データ モデリングにおける基本的なベスト プラクティスの詳細については、『Mastering Data Modeling：A User-Driven Approach（データ モデリングの習得：ユーザー主導アプローチ）』（John Carlis および Joseph Maguire 共著、Addison-Wesley、2000 年）を参照してください。

エンバカデロ・テクノロジーズについて

エンバカデロ・テクノロジーズは、1993 年にデータベースツールベンダーとして設立され、2008 年にボーランドの開発ツール部門「CodeGear」との合併によって、アプリケーション開発者とデータベース技術者が多様な環境でソフトウェアアプリケーションを設計、構築、実行するためのツールを提供する最大規模の独立系ツールベンダーとなりました。米国企業の総収入ランキング「フォーチュン 100」のうち 90 以上の企業と、世界で 300 万以上のコミュニティが、エンバカデロの Delphi®、C++Builder®といった CodeGear™製品や ER/Studio®、DBArtisan®、RapidSQL®をはじめとする DatabaseGear™製品を採用し、生産性の向上と革新的なソフトウェア開発を実現しています。エンバカデロ・テクノロジーズは、サンフランシスコに本社を置き、世界各国に支社を展開しています。詳細は、www.embarcadero.com/jp をご覧ください。