

【データモデルを通して業務を理解しよう<第54回IT勉強宴会>】

ITエンジニアのための データベース再入門解説

2017年3月24日
株式会社データアーキテクト
真野 正



Contents.

1. 自己紹介
2. 執筆の背景など
3. データモデリング観点でのトピックから
 - データモデルの多義性(P.56)
 - 整合性維持の仕組み(P.86)
 - NoSQLとRDB(P.92)
 - リレーショナル理論の適用(P.106)

1. 自己紹介

■経歴

大手SI会社勤務(株式会社シーエーシー他)を経て2005年独立。

産業系での生産管理、販売管理系を中心とした業務システム構築でのリーダー、PMを経験。DOAによるシステム開発、データモデリング、データベース設計の実務またはコンサルティングに従事する。

リポジトリ、モデリングツール(ERwin等)の普及に関わる。

システム基盤はメインフレーム、C/S、Web系を経験、RDB黎明期より携わり、データモデリングは、概念レベルにとどまることなく、実装を意識した設計を心掛けている。DBA業務、SQL性能改善にも携わり、データアーキテクチャ全般を対応。

大規模開発プロジェクトでのデータモデルレビュー、複数モデル間での共通・個別の調整、データ統制業務に複数携わる。

■外部活動

DOA+コンソーシアム(休眠)

DAMA日本支部

日本データマネジメント協会(JDMC)

日本ガイドシェア委員会・研究支援

最近の業務

業種	プロジェクト	役割
流通	全社システム再構築システム化計画	論理データモデル、データ管理標準策定
電力	資材. 予算実績管理システム	複数データモデリングチームの統轄、論理モデルレビュー
官公庁	決済システム再構築	データモデリング、データ管理
鉄道	次期販売システムのための論理DB構想	現行DB課題分析、新論理DB検討
鉄道	車両管理システム再構築	データ管理業務全般、マルチベンダー統轄
交通	管制支援システム構築・DB支援	DA、DBA業務
製造業	購買システムデータ移行支援	モデルベースの現新データマッピング
鉄道	設備管理システム性能改善	SQL解析、パーティション化
出版卸	基幹再構築データモデリングのためのデータ統制	ドメイン、ネーミングルール評価
金融(銀行)	統合顧客システム構築・概念DB設計	統合顧客データモデリング
金融(損保)	基幹システム再構築データモデリング支援	データモデリング標準化 論理データモデル作成
電力	スマートメータ管理システム	DB設計リファクタリング、DB性能改善
金融(生保)	保険. 統合顧客システム. 基本計画策定	概念・論理データモデル作成、レビュー
金融	保険. リスク管理システム. 要件定義	概念データモデル作成
金融	保険. リスク管理システム. 基本設計	DA、DBA業務

データモデリング経験業務から

■ 鉄道・車両管理システム再構築

鉄道会社で保有する車両の保守作業を支援するシステムの再構築において、概念データモデル作成、設計フェーズで複数の開発ベンダーが作成するデータモデルを統率していくDA(データ管理者)の役割を担った。エンティティの総数は、約800。

車両毎の製造年数や走行距離などを管理し、保守時期を通知する車両管理、保守・修理作業での入庫から出庫までの作業計画、工程管理、必要資材の管理など。

データモデルだけでなく、設計情報をリポジトリとして管理し、工程間でのトレーサビリティを確保した。

■ 金融・統合顧客システム概念データモデル作成

メガバンクグループ内でのバーゼル対応や貸付リスク等の信頼性を上げるために、顧客企業や企業グループとしての名寄せを行うための顧客データのモデルを作成した。

データモデルの定石モデルとなっているパーティモデルを適合させる形で進めた。作成したデータモデルは、その後要件定義を経て、システム実装された。

■ 損保・契約系システム再構築データモデリング

長年の間に硬直化したシステムとなり、新商品や新ビジネスへの対応ができなくなったため、基幹システム再構築が決断され、データモデル主導での構築が進められ、その立ち上がりを支援した。

超大規模システムでの段階的なデータモデリングの進め方を議論し、モデリング手順や規約としてまとめた。また、レビューのための観点や具体的なチェックシートを作成し、初期段階でのレビューにも参加した。

■ 金融・リスク管理システム構築でのデータモデリング

保有資産の運用リスク、負債としての保険商品の事故発生率や死亡リスクなどを統合的に捉えたリスク量を測定して、新商品開発や顧客開拓に寄与するためのデータウェアハウス構築のためのデータモデリングを行った。リスク量測定のリソースとなる業務データを分析し、経営指標となるKPI算出のための集約データのモデルを作成した。

2. 執筆の背景

- パフォーマンスチューニングの視点からデータモデリングの重要性を知ってもらいたい
- 実行計画を意識しよう
- モデリングは大事だけど動いてなんぼの世界
- 懸念
 - ◆ データ構造・特性を十分に把握しないで設計・製造している
 - ◆ RDBが単なるファイルシステムとして使われている
 - ◆ 要求される性能を満足できていない
 - ◆ 増え続けるデータへの備えができていない
 - ◆ データベースが当たり前のように使われているが正しい使われ方が...
- 目指すところ
 - ◆ Web上に氾濫している技術情報を判断できるように
 - ◆ 基本に立ちかえるためのトリガーとなれば

構成

◆第1章と2章は「課題編」です。

第1章では、DBMSをよく知らないために起こった事件等を例示し、問題を提起します。

続く第2章では、ITエンジニアが陥りやすい、誤ったDBの使い方を見ていきます。

◆第3章と4章は「理論編」です。

第3章では、関係モデルを学び直します。第4章はDBMSの内部構造と操作を学び直します。

◆第5章以降は「解決編」です。

第5章は「DBMSの苦手を知る」ことから初めます。誤った使い方を分析し、理論と照らし合わせて、問題の所在を解明します。

第6章ではDBMSを使いこなす術、第7章でSQLを使いこなすための代表的な手法を詳しく解説。

最後の第8章では「DB利用の心得」をまとめて、本書を締め括ります。

3. データモデリング観点でのトピックから

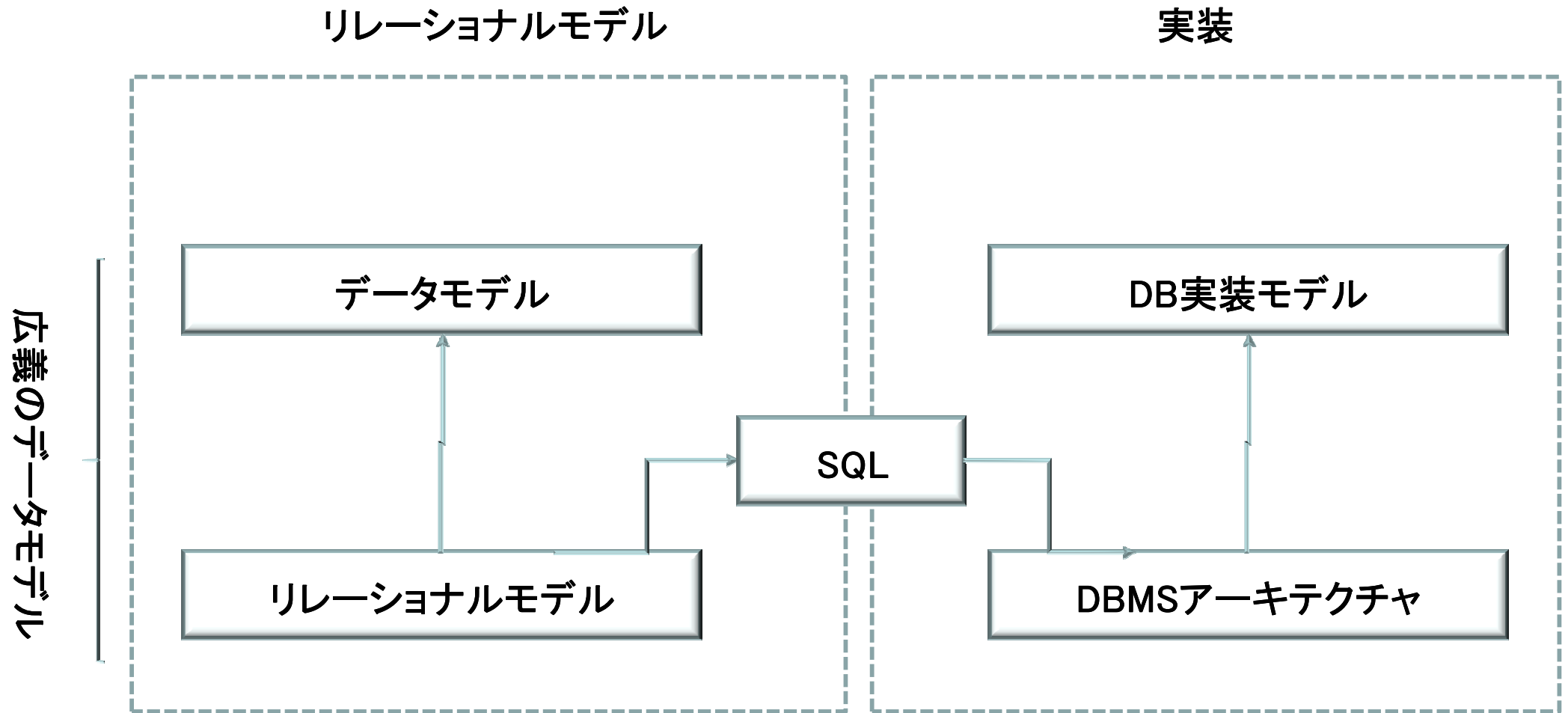
- ① データモデルの多義性 (P.56)
- ② 整合性維持の仕組み (P.86)
- ③ NoSQLとRDB(P.92)
- ④ リレーショナル理論の適用(P.106)

(1) データモデルの多義性

- 業務の実体を表わすためのモデル
- DB実装のためのモデル
- DBMSの内部構造
- 概念・論理・物理
- IE記法、IDEF1X記法、渡辺法、TH法、TM法、...
- トップダウンとボトムアップアプローチ

リレーショナルモデルと実装関係

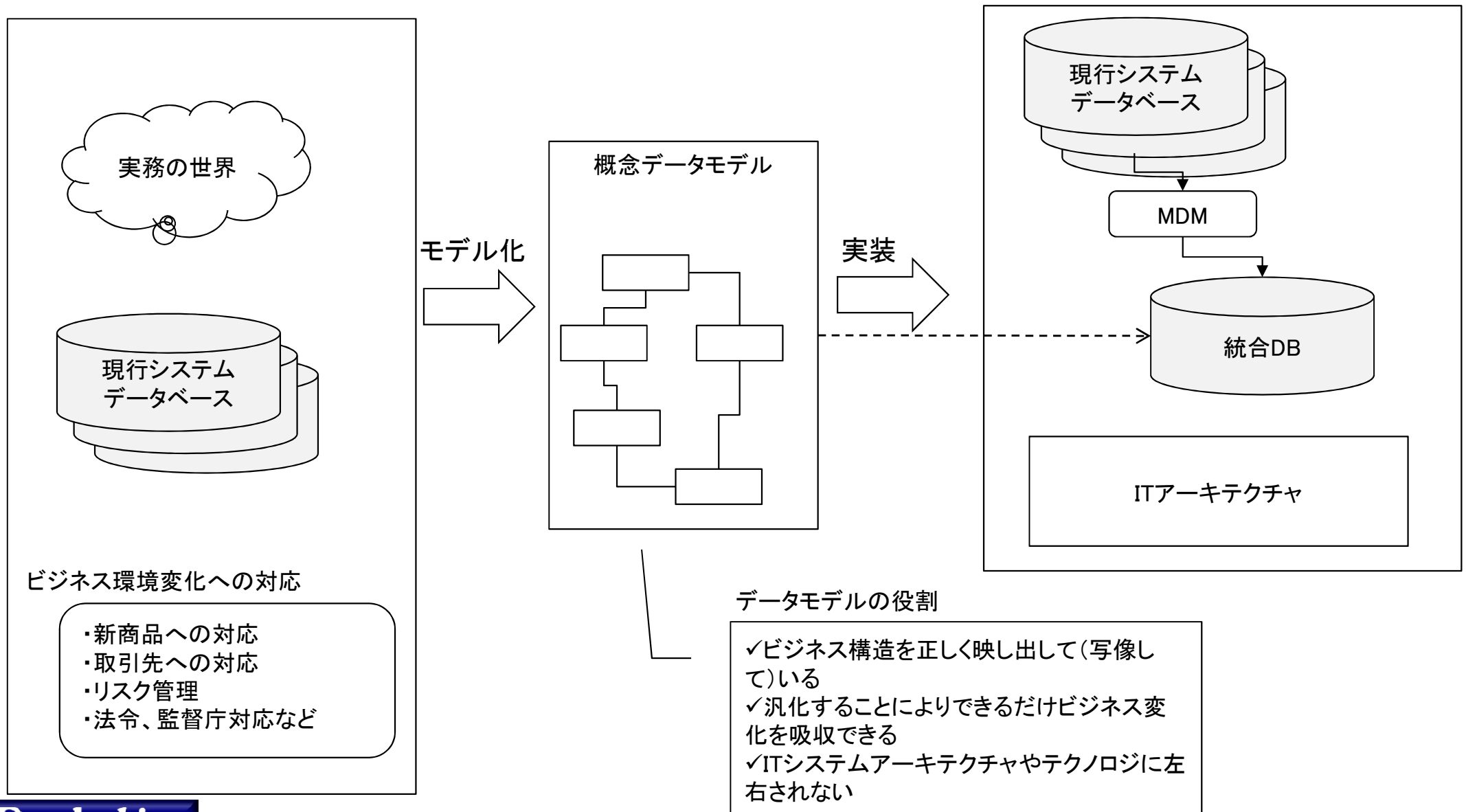
■本書でのデータモデルの定義



※SQLは、リレーショナルモデルの表現手段であると共に、DBMSの操作言語でもある。

データモデルの1つの定義

- データモデルは、業務の世界を正しく写像し、ビジネス変化に耐えうる、ITインフラテクノロジーに依存しないビジネスの世界を描いたものです



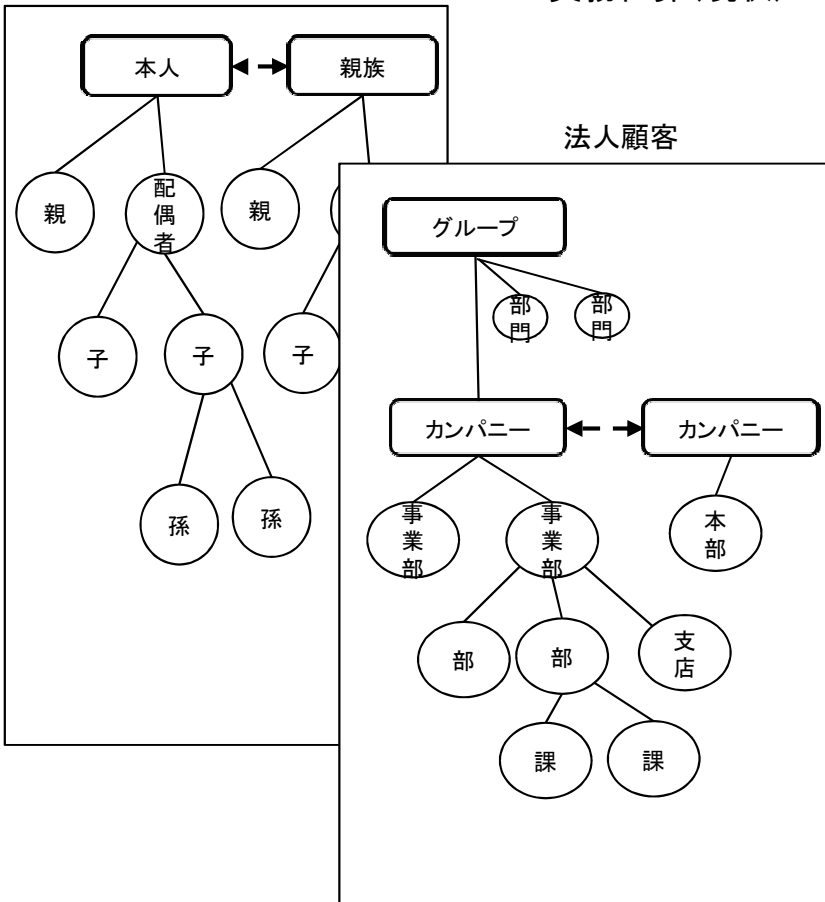
データモデリングアプローチ

顧客マスターでの例示

- 【顧客マスターの例】個人顧客、法人及びその組織構造をパーティという概念で統括して捉えることにより、パーティデータモデルを参照モデルとし、トップダウンによる構造の骨格決め(全体鳥瞰)と現行分析を中心としたボトムアップ分析による属性の付加等の精緻化を併用したモデリングを行う。

リテール顧客

<<実務世界(現状)>>



トップダウンモデリング

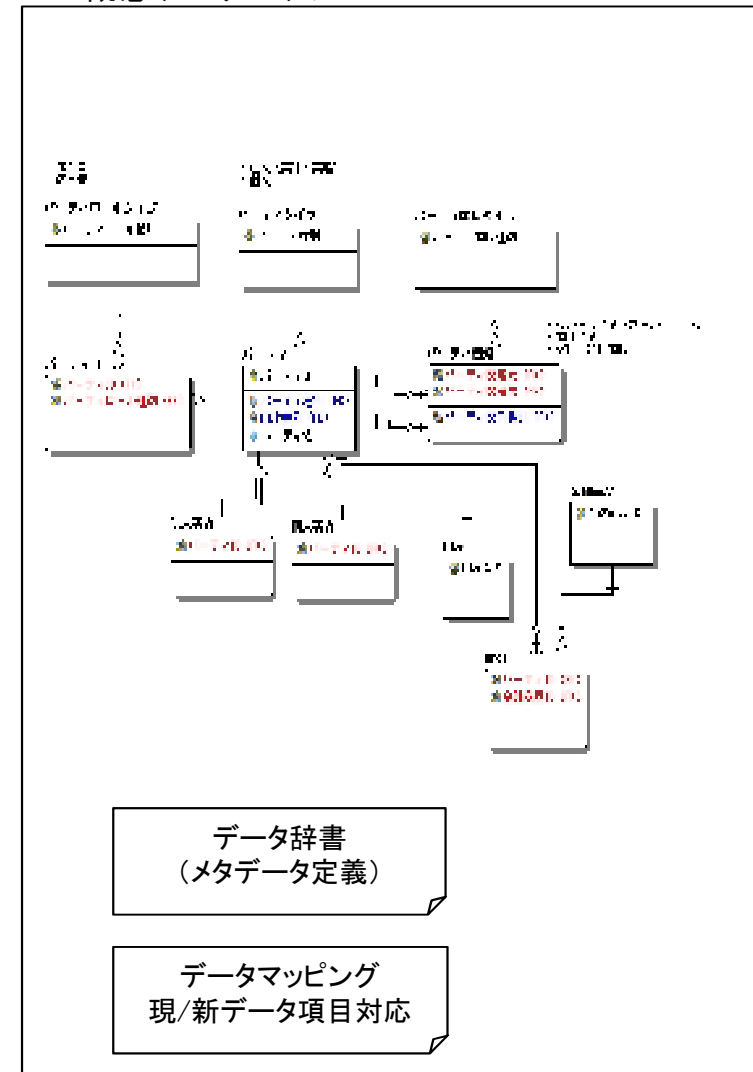
例えば、「パーティモデルパターン」を適用して以下の概念で捉える

- ◆個人/法人・組織の識別
→パーティタイプ
- ◆預金者、融資者等の役割
→パーティロール
- ◆家族、親族関係、企業グループ・組織構造
→パーティ関係

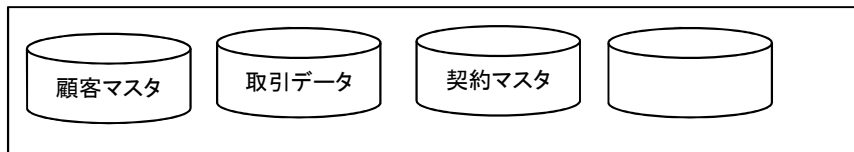
- ◆属性抽出
- ◆異音同義語、ネーミング
- ◆共通/個別属性の識別

ボトムアップモデリング

<<概念データモデル>>

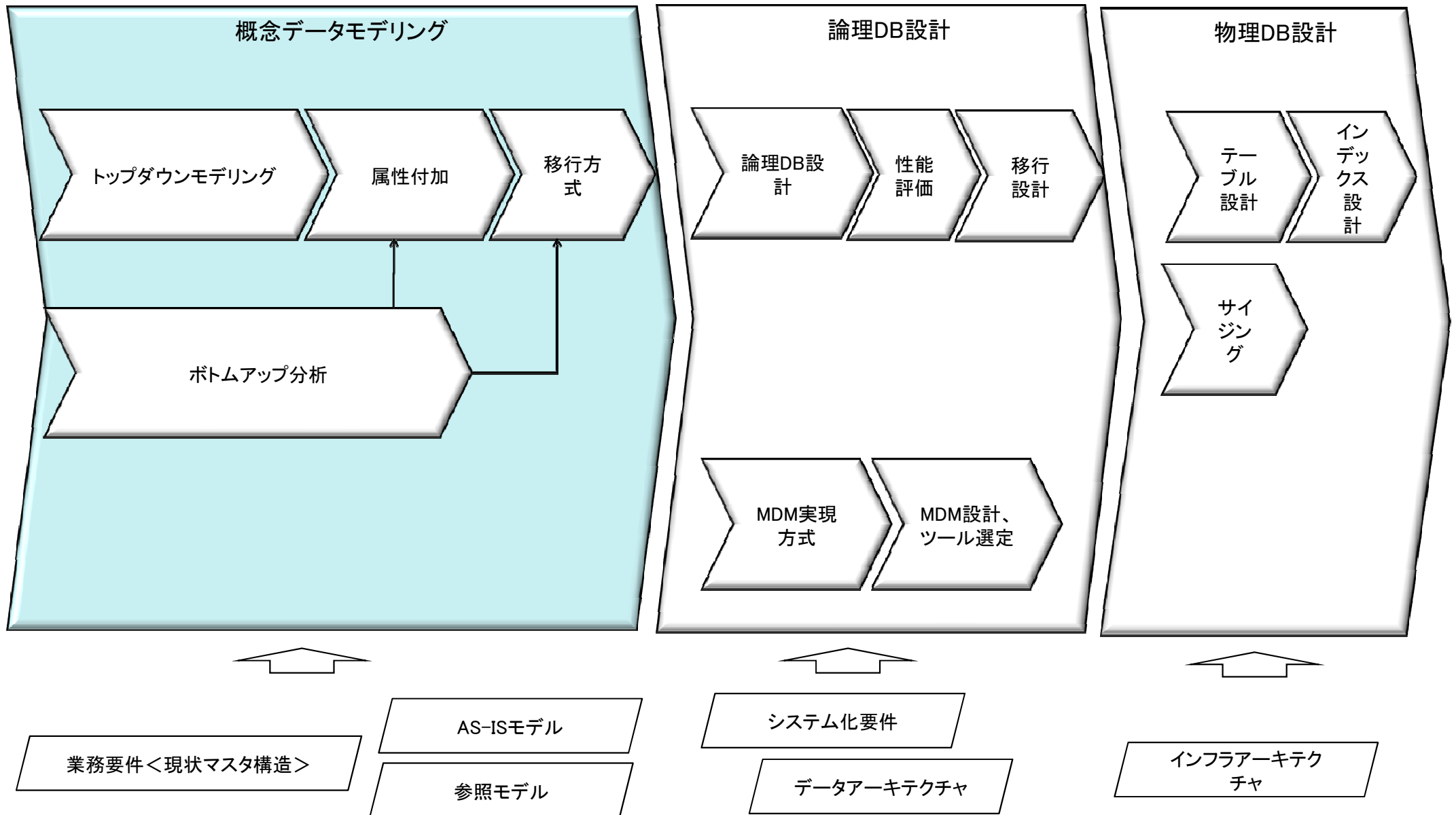


顧客情報を保有するDB群



フェーズ別データモデル

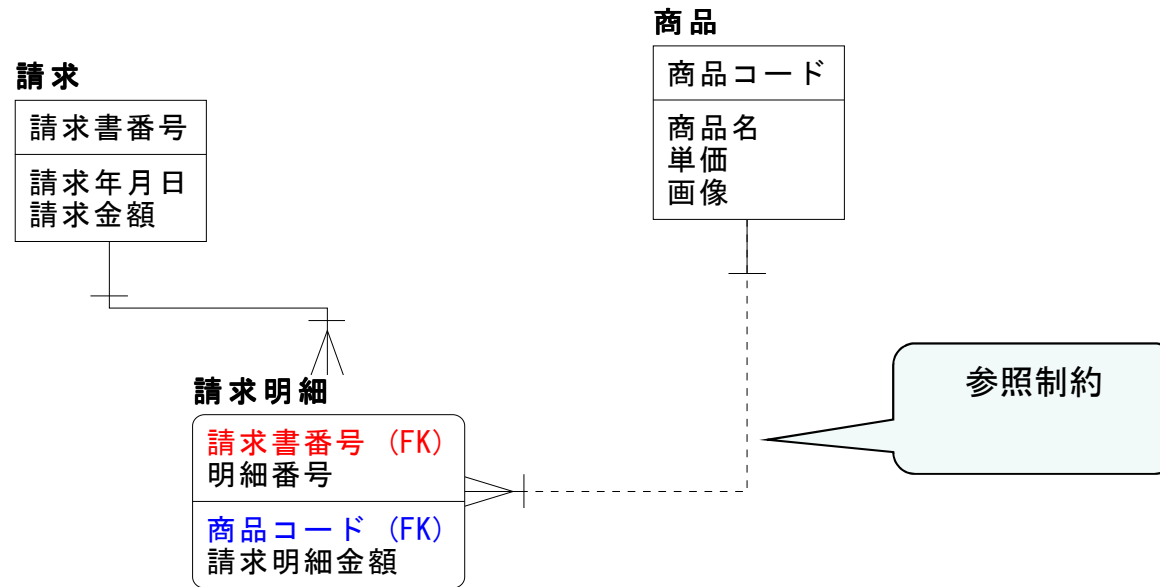
- 概念データモデルを受けて論理DB設計、物理DB設計の手順で行う



(2) 整合性維持の仕組み

- 複数のテーブルからなるデータ構造での関係維持
- データの整合性維持
 - ◆ 参照制約
- トランザクションの一貫性制御
 - ◆ コミット／ロールバック
- 排他制御
 - ◆ ロック機能

データの整合性維持



請求明細

請求番号	明細番号	商品コード	請求明細金額
S0001	001	CDO120	¥3,500
S0001	002	CDO113	¥3,200

商品

商品コード	商品名	単価	画像
CD0100			
CD0110	雨の横浜	¥4,000	
CD0120	たそがれ日本	¥3,500	

挿入

商品マスターに存在しない

(3) NoSQLとRDB

■ NoSQL登場の背景

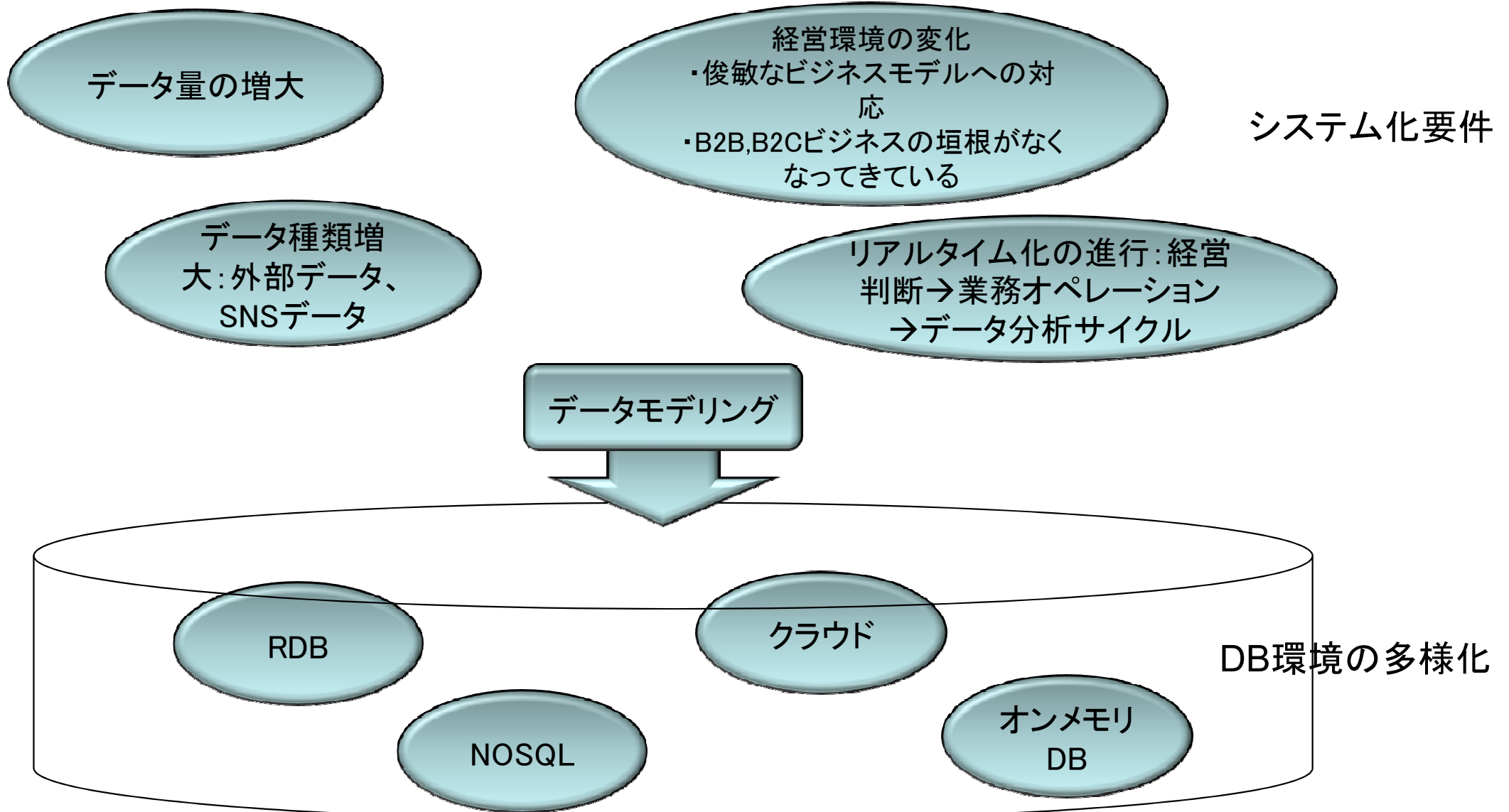
◆DB環境の多様化とシステム化要件

■ データアーキテクチャ

■ モデリングの変化・役割

DB環境の多様化とシステム化要件

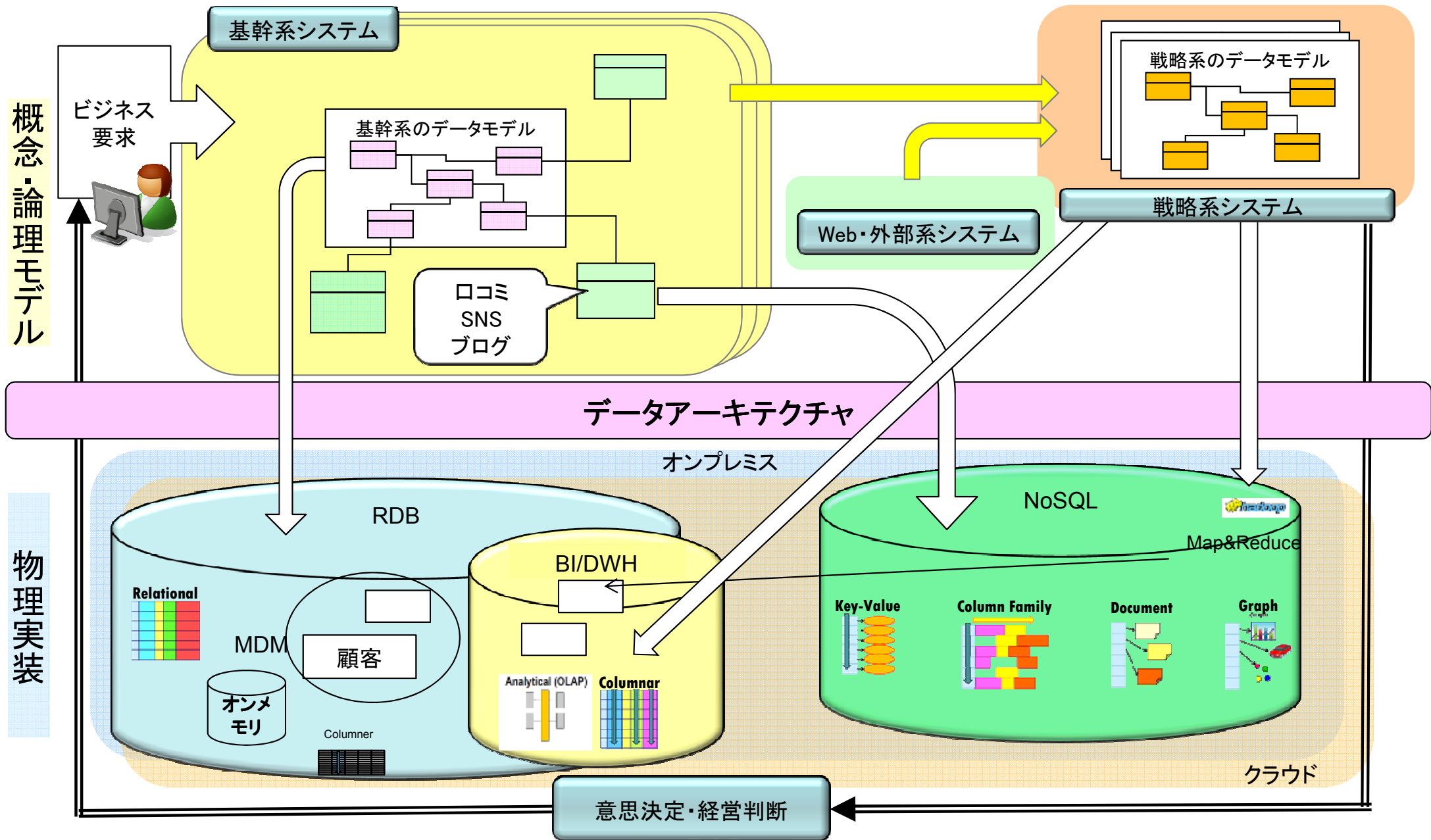
- データモデル化によりシステム化要件を適切なDB環境へのマッピングが可能となる



NoSQLの例

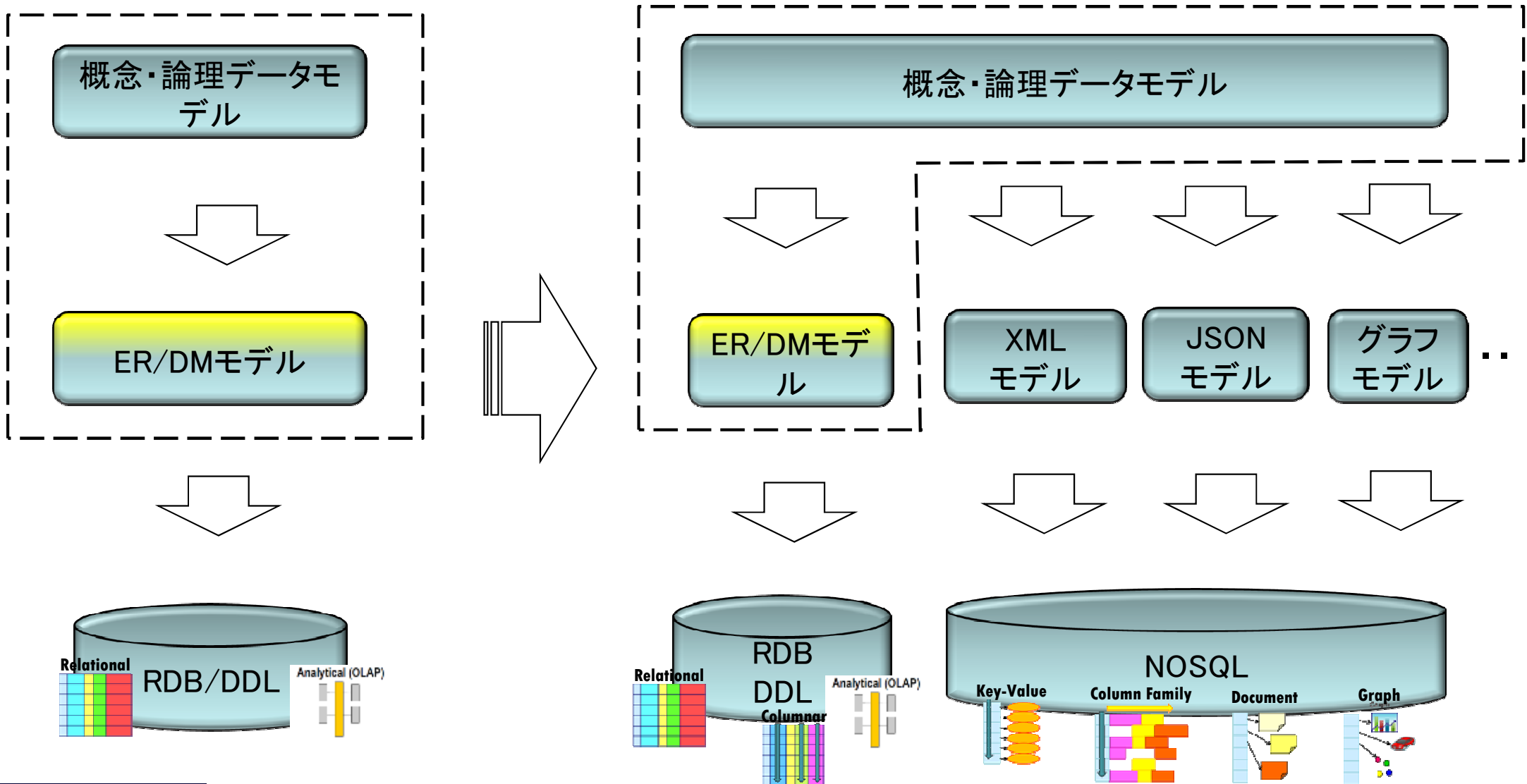
形式		特徴	主な利用法	プロダクツ例
 <p>Key-Value</p>	キー・バリュー型	キーと値だけでデータを保管する単純な形式	<ul style="list-style-type: none"> ・画像ライブラリ ・ファイルシステム ・オブジェクトキャッシュ ・拡張予定のシステム 	<ul style="list-style-type: none"> ・Memcache ・Redis ・Riak ・DynamoDB
 <p>Column Family</p>	カラム指向型	行と列をキーに使った、抜けのあるマトリクス型	<ul style="list-style-type: none"> ・Webクローラーの動作結果 ・強い整合性を必要としないビッグデータ処理 	<ul style="list-style-type: none"> ・Hbase ・Cassandra ・Hypertable
 <p>Document</p>	ドキュメント指向型	DB内に直接、階層構造をもったデータ構造を格納	<ul style="list-style-type: none"> ・可変長データ ・ドキュメントサーチ ・インテグレーションハブ ・Webコンテンツ管理 ・出版 	<ul style="list-style-type: none"> ・MongoDB (10Gen) ・CouchDB ・Couchbase ・MarkLogic ・eXist-db
 <p>Graph</p>	グラフ指向型	関連の強い課題に対応	<ul style="list-style-type: none"> ・SNS ・不正検出 ・強い関連をもつデータ 	<ul style="list-style-type: none"> ・Neo4J ・AllegroGraph ・Bigdata (RDF store) ・InfiniteGraph (Objectivity)

データアーキテクチャとモデリングの価値



モデリングの変化

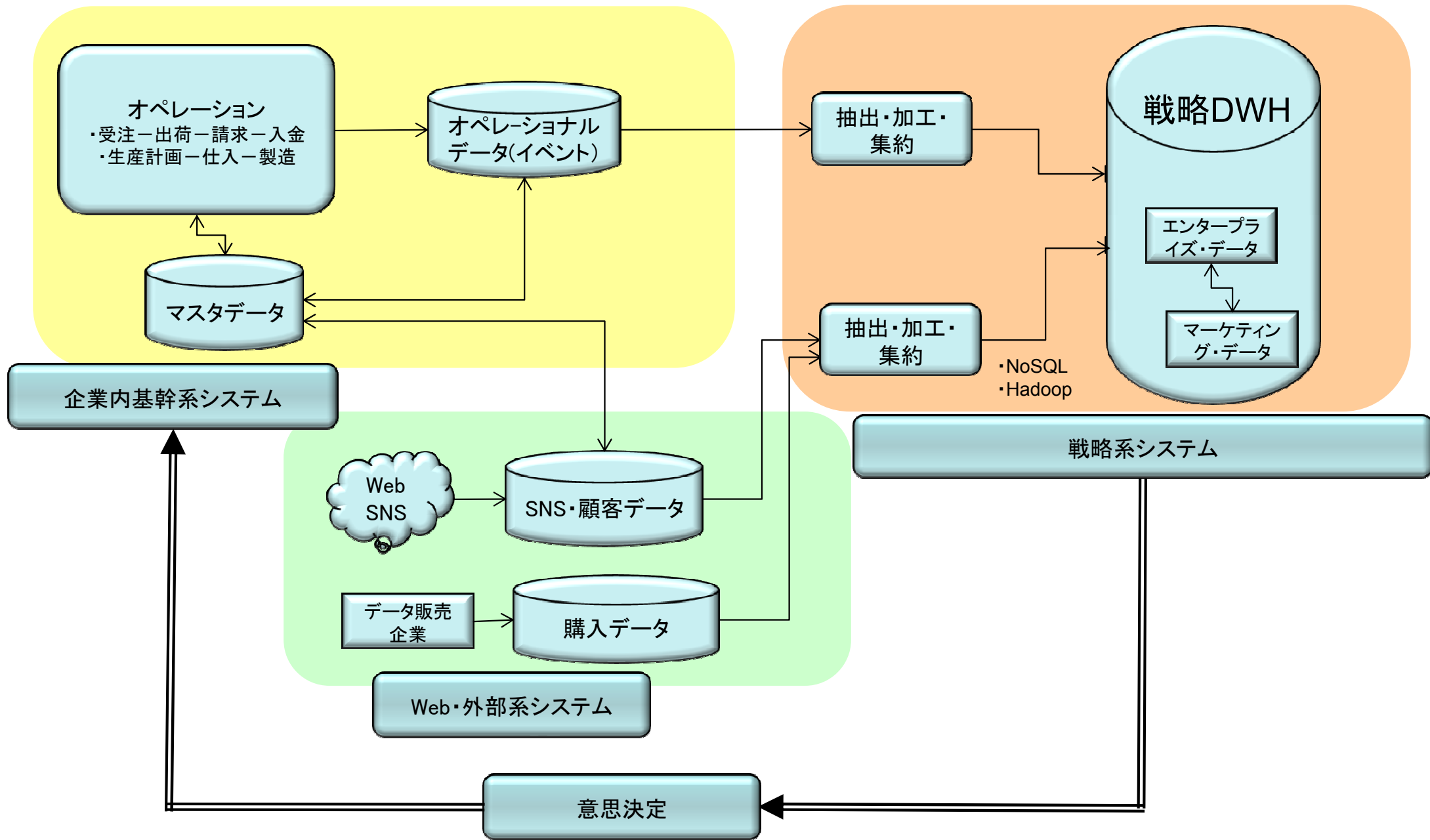
- 物理モデルとしてのERモデリングは、必須ではなくなるだろう
- ビジネスを写像した概念・論理データモデルの役割は不変である



概念・論理モデルの役割

- ビジネス要求の具現化への近道
- 時代にマッチした最適な実装環境の選択(適材適所)
- 経営支援のための戦略情報提供
- ビジネスモデル変化への俊敏な対応

データアーキテクチャ



データアーキテクチャ

- 基幹系、情報系の垣根がなくなっている
- 基幹系から取得したデータを集計・分析して、基幹系へフィードバックする
 - ◆EX)発注→集計・分析→需要予測→発注リコメンデーション
- リアルタイム経営
 - ◆オペレーショナルデータの分析結果を次の意思決定へ迅速に生かす

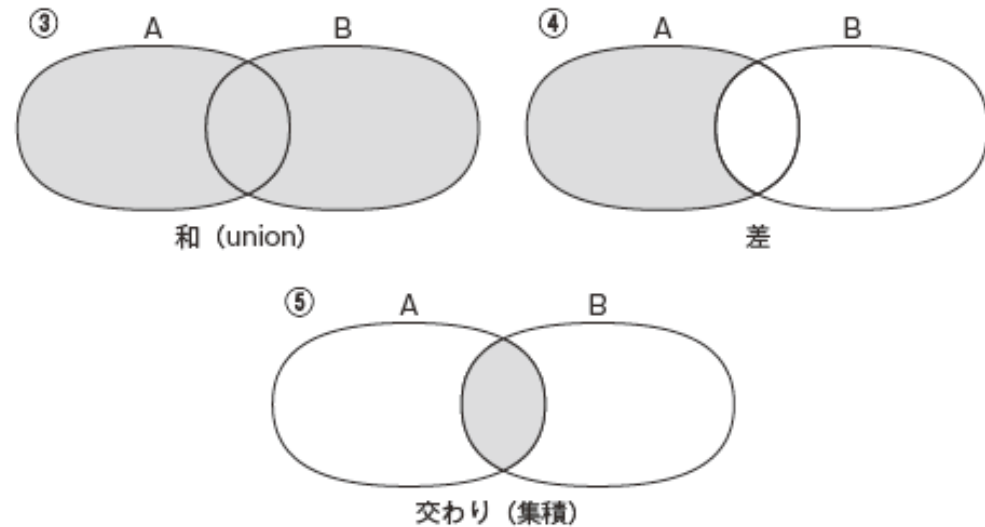
(4)リレーショナル理論の適用

- リレーショナルモデルと操作
- 実店舗、web店舗顧客の例
- データモデルによる裏付け

リレーショナルモデル操作

- Insert
- Update
- Delete
- Select
- Restrict
- Projection
- Union
- Diference except/minus
- Intersect
- Product
- Join
- Divide

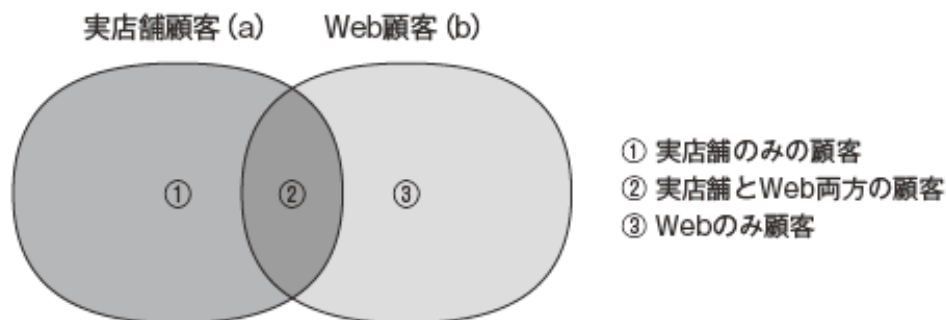
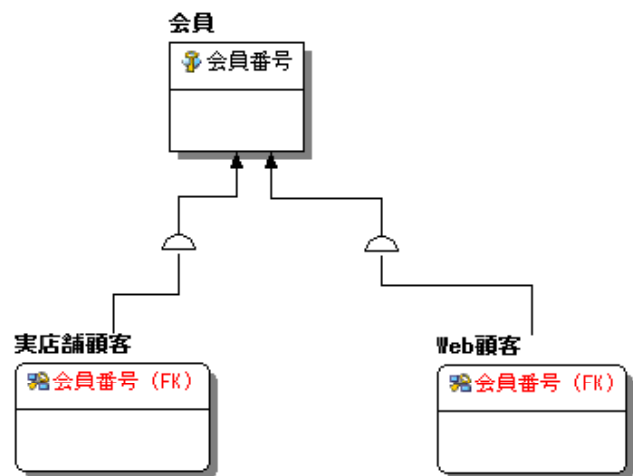
図3.4-2 リレーショナルモデルの操作



例示 実店舗、web店舗顧客の例

■ 和・差の関係など、データモデルだけでは表現が難しいものもある

図3.10 ベン図



- ・ 実店舗のみで購入している顧客 ①
集合aから実店舗とWebの両方を利用している顧客 (②) を除いたものとなります。
- ・ Webのみを利用している顧客 ③
- ・ 実店舗とWebの両方を利用している顧客 ②
aとbの結合として求めることができ、①よりも先に求めておきます。
- ・ 実店舗またはWebを利用している顧客 (重複を除く)..... ①+②+③
aとbの和として求めることができます (UNIONによりaとbの重複データは排除されます)。

①実店舗のみで購入している顧客

A.

```
SQL> SELECT a.会員番号,a.氏名 FROM 実店舗顧客A a
```

```
2 MINUS
```

```
3 (SELECT c.会員番号,c.氏名 FROM 実店舗顧客A c INNER JOIN WEB  
顧客A b ON c.会員番号 = b.会員番号);
```

B.

```
SQL> SELECT a.会員番号,a.氏名 FROM 実店舗顧客A a WHERE
```

```
2 NOT EXISTS (SELECT c.会員番号,c.氏名 FROM WEB顧客A c WHERE  
a.会員番号 =c.会員番号);
```

実行計画

```
SQL> SELECT a.会員番号,a.氏名 FROM 実店舗顧客A a
2 MINUS
3 (SELECT c.会員番号,c.氏名 FROM 実店舗顧客A c INNER JOIN WEB顧客A b ON
   c.会員番号 = b.会員番号);
```

実行計画

Plan hash value: 2901267341

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10007	537K	30 (57)	00:00:01
1	MINUS					
2	SORT UNIQUE		10007	234K	15 (14)	00:00:01
3	TABLE ACCESS FULL	実店舗顧客A	10007	234K	13 (0)	00:00:01
4	SORT UNIQUE		10005	302K	16 (19)	00:00:01
5	NESTED LOOPS		10005	302K	14 (8)	00:00:01
6	TABLE ACCESS FULL	実店舗顧客A	10007	234K	13 (0)	00:00:01
* 7	INDEX UNIQUE SCAN	SYS_C0015376	1	7	0 (0)	00:00:01

```
SQL> SELECT a.会員番号,a.氏名 FROM 実店舗顧客A a WHERE
2 NOT EXISTS (SELECT c.会員番号,c.氏名 FROM WEB顧客A c WHERE a.会員番号 =
   c.会員番号) ;
```

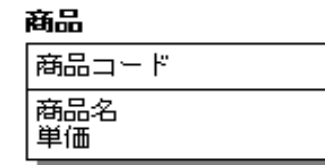
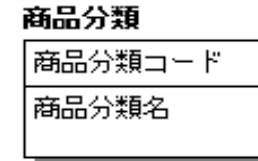
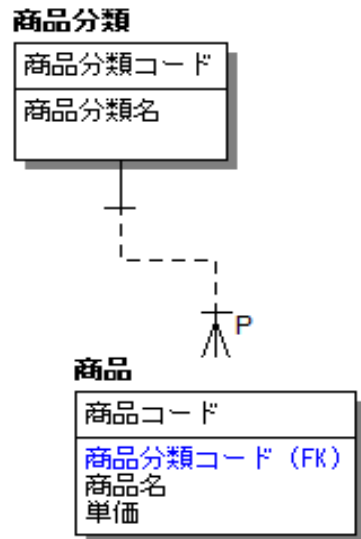
実行計画

Plan hash value: 3617630561

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		10007	302K	14 (8)	00:00:01
1	NESTED LOOPS ANTI		10007	302K	14 (8)	00:00:01
2	TABLE ACCESS FULL	実店舗顧客A	10007	234K	13 (0)	00:00:01
* 3	INDEX UNIQUE SCAN	SYS_C0015376	1	7	0 (0)	00:00:01

データモデルによる裏付け

■ER図として関連が描ける === リレーショナル理論の適用 === SQL記述可



カラムの一部が結合キーとなっており、型変換が発生。

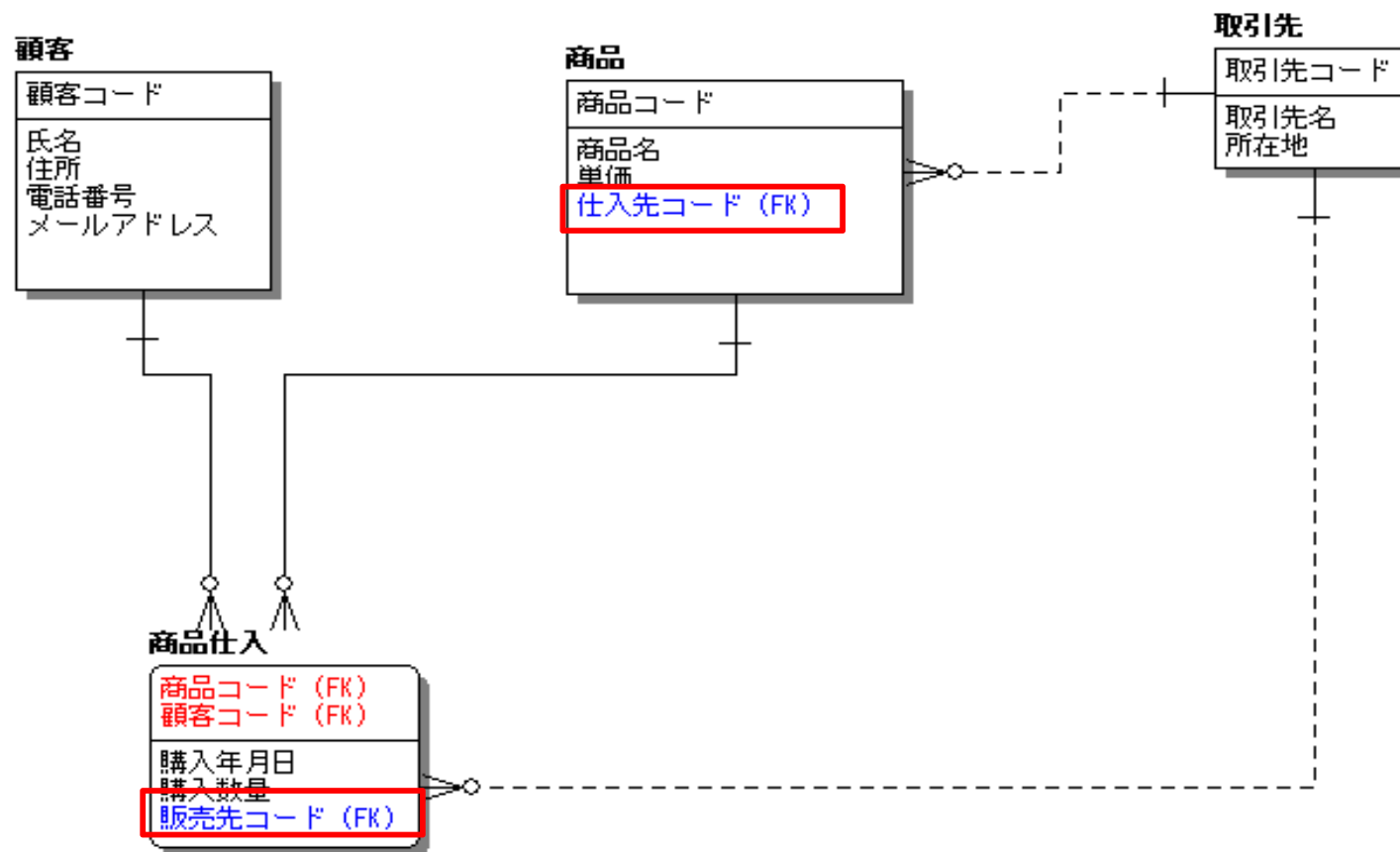
商品コード体系
□□○○○○○○○○ (8桁)
□□ : 商品分類コード
○○○○ : 商品識別番号

```
select t1.商品コード、t1.商品名、t2.商品分類名
from 商品 t1
left outer join 商品分類 t2
on t2.商品分類コード = t1.商品分類コード
;
```

```
select t1.商品コード、t1.商品名、t2.商品分類名
from 商品 t1
left outer join 商品分類 t2
on t2.商品分類コード = t1substr(商品コード,1,2)
;
```

データモデルによる裏付け

- リレーションシップにより、商品の「仕入先」と「販売先」の情報を誤り無く取得することができる。
- 仕入先、販売先は、取引先のロール名と呼ばれデータモデリングの基本作法です。



お問い合わせ先

■ データモデリングに関しては、以下の書籍で詳しく解説しています

◆ 実践的データモデリング入門



◆ 独習データベース設計



株式会社データアーキテクト
代表取締役
ITコンサルタント 真野 正

E-mail.mano@dataarch.co.jp

URL:<http://dataarch.co.jp>