多様化するデータベース環境とモデリングの価値

2013年8月29日

株式会社データアーキテクト
真野　正

Copyright 2013 DataArchitect Co.,Ltd.

概要

- 従来、企業における一般的なデータベース環境はRDBMSがほぼ全てを占めていましたが、現在ではテクノロジーの進歩により、様々な選択肢が存在します。
- このような状況のなか、実際の設計やアーキテクチャーはどのように考えれば良いのか、という声も多く囁かれています。
- 一方、開発プロジェクトの現場では、データベース設計上直面している課題があります。
- 現在のデータベース関連技術のトレンドを念頭に置きながら、現場で直面している課題への対応、モデリングの有用性について考えてまいります。

Copyright 2013 DataArchitect Co.,Ltd.

目次

1. DB関連技術環境の変化
2. DB設計現場で遭遇する身近な課題
3. DB環境の多様化とシステム化要件
4. データアーキテクチャとモデリングの価値

Copyright 2013 DataArchitect Co.,Ltd.

1．DB関連技術環境の変化

- NOSQL登場によるDBMSの多様化
  - Key-Value
  - カラム指向
  - ドキュメント指向
  - グラフ
  - Hadoop、Map&Reduce
- RDBMSの革新
  - パーティション化
  - カラムナー
  - NOSQL対応
  - 改訂SQLへの対応
- メモリー技術の発展
- クラウド化の進展
  - AWS

Copyright 2013 DataArchitect Co.,Ltd.

NOSQL登場によるDBMSの多様化

データモデル（DB格納構造）からの分類

- ■Key-Value　DynamoDB
- ■カラム指向　Cassandra,Hypertable,Bigtable
- ■ドキュメント指向　MongoDB,CouchDB
- ■グラフ　InfiniteGraph,Neo4j

DBMSではないが

- ■Hadoop、Map&Reduce

Copyright 2013 DataArchitect Co.,Ltd.

NOSQLへの要請

- データ量増大へのスケールアウト対応
  - ビッグデータ対応　3V（Volume：量、Variety:種類、Verocity：速度）+1V
- 非構造化データへの対応
- 固定のスキーマ定義ができない
- オープンソース化の波
- 始まりはエンタープライズ以外から
  - ソーシャル
  - ゲーム

Copyright 2013 DataArchitect Co.,Ltd.

NOSQLNOW2013

- 開催：2013年8月20-22日、米国サンノゼ・コンベンションセンター
- Datavesity（WilshereConfrence)社主催。3回目。
- 出席者は、前回とほぼ同じ300名近く。
- 一部のWeb系企業での使用からエンタープライズでの利用に拡大しつつある。
- ドキュメント型DBのMongoは、RDBMSの機能を取込んできている。
- 大手DBMSベンダーのNOSQL領域への参入
  - Oracleは独自NOSQL(Berkeley　DB）
  - IBMは、MongoDBのセキュリティを強化
- JASONのシンタックスを強化するためにJASONスキーマ（XMLスキーマ相当）が、提案されているが、途上。
- エンタープライズで使用していくにはまだまだ未熟である。
- 10gen（MongoDB)社は、Oracle社を超えることを目指している。（10gen、Max,Schireson　CEO）
- 実装上スキーマレスとはいっても、概念・論理モデルの有用性が無くなったわけではない。Jsonコードの生成機能などが望まれる。
- NOSQLのためのSQLライクにアクセスできるJSONiqなど周辺ツールも登場してきてる。
- グラフDBが熱い。
  - NOSQLは、RDBからリレーションシップを排除したものともいえるが、グラフDBは関係性をビジュアルに表現したものである。SNSでの人間関係などが例示されることが多いが、エンタープライズでも利用が広がる可能性がある。
  - オブジェクト指向DBからの進化系か。
  - marvel entertainment社でのヒーローやストーリーの関係をグラフ表現した事例。
- Denormalizationを是とするNOSQLとの間で正規化論争が盛り上がるかもしれない。

http://nosql2013.dataversity.net/

Copyright 2013 DataArchitect Co.,Ltd.

キー・バリュー型データモデル

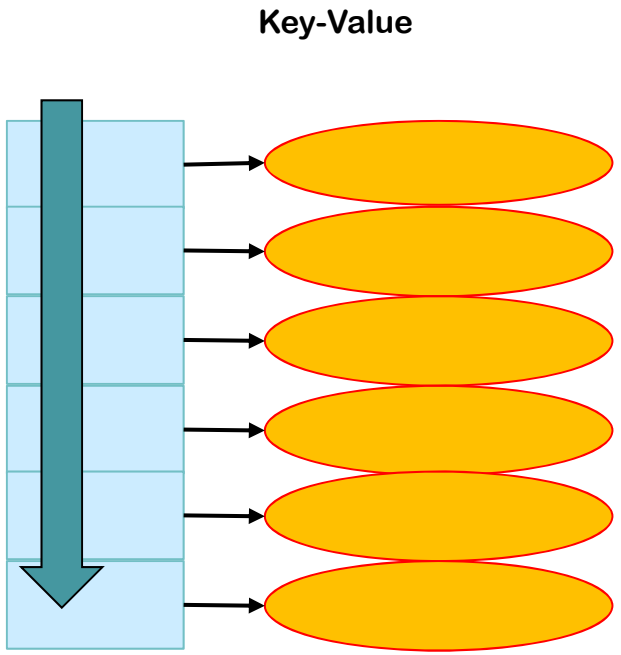
■全てのバリューに識別のためのキーを付与

■大規模ハッシュテーブル

■Key/Valueペアとして格納

■キーを使用したアクセスに最適

■単純なクエリーをサポート



Ex)
Dynamo　:Amazon
Voldemort　:Linkdin
Riak
Hibari
Redis
Scalaris
Tokyo Cabiner/Tyrant

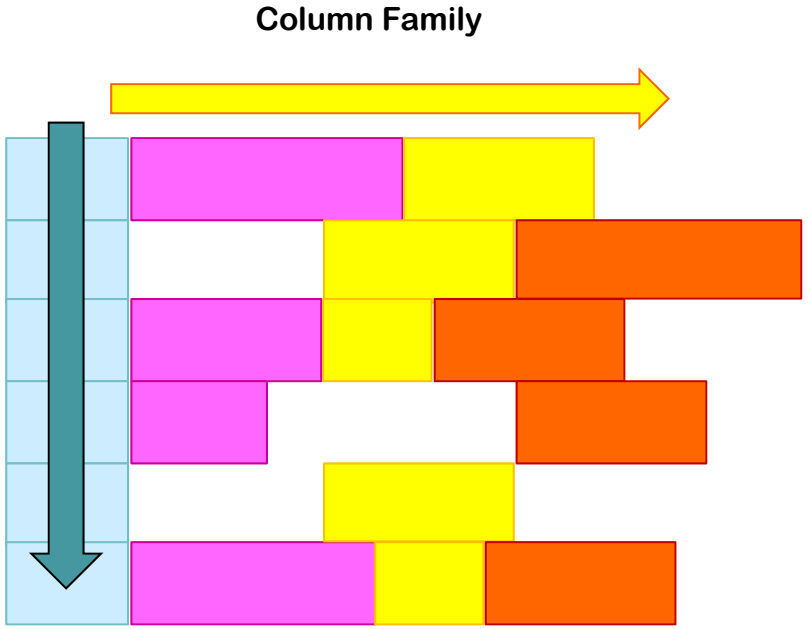
Copyright 2013 DataArchitect Co.,Ltd.

カラム指向型データモデル

■1つのキーに対して複数のバリューを持てる

■1つのキーに対して複数のカラムまたはカラムファミリーを持つ

■カラムは、行毎に可変



Ex)
Bigtable
Cassandra
Hbase
Hypertable

Copyright 2013 DataArchitect Co.,Ltd.

ドキュメント指向型データモデル

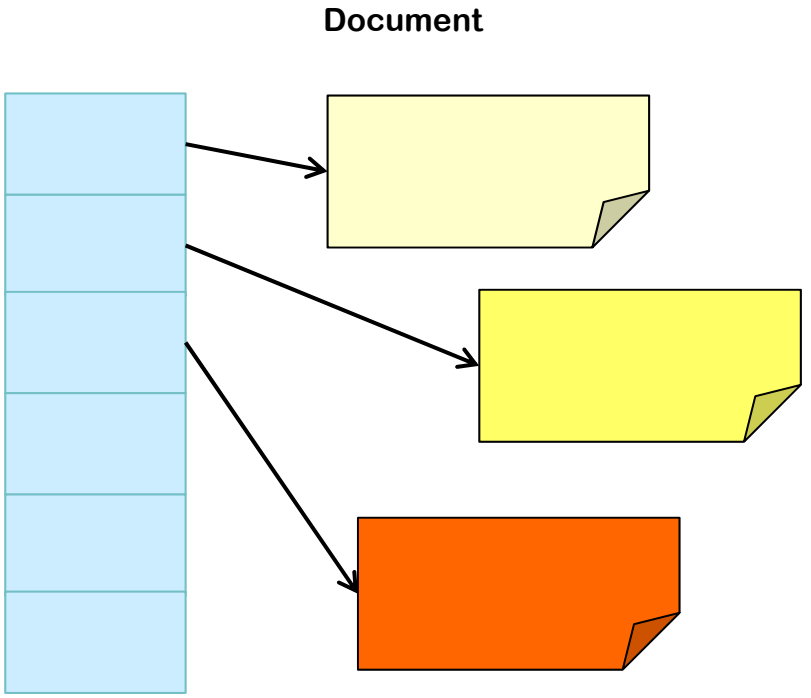
■ JSON、XMLでデータ書式を定義

■階層構造で格納

■スキーマレス
　DB上は器だけを定義、スキーマは、プログラム上でJSONで定義

■MongoDB:
- ✓Key-valueとRDBの良いところを
- ✓JSONが開発者に受けている

■進展
- ✓JSONスキーマによるチェック強化
- ✓SQLライク：ex)JSONiq



Ex)
CouchDB
MongoDB

Copyright 2013 DataArchitect Co.,Ltd.

グラフ型データモデル

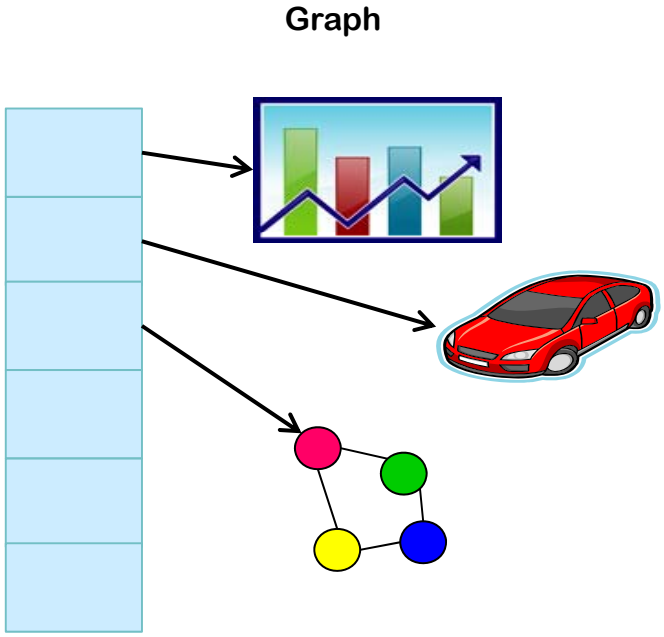
■データとデータの間のつながりを管理

■グラフ理論に基づく

■オブジェクト指向DB

■RDF:Resource Description Format(W3C)
　Cypher(Neo4j)

■SPARQL：Standardlized query language(W3C)
　✓RDFのアクセス言語

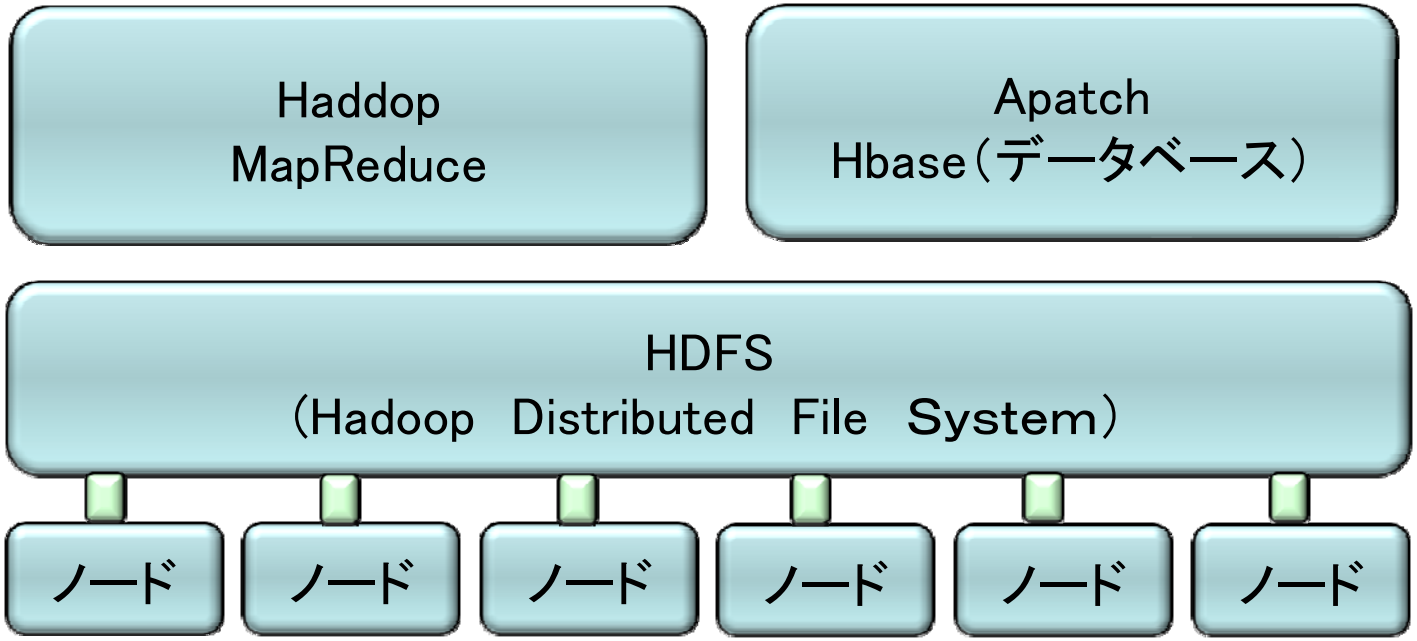


Ex)
InfiniteGraph
Neo4j

Copyright 2013 DataArchitect Co.,Ltd.

Hadoop処理

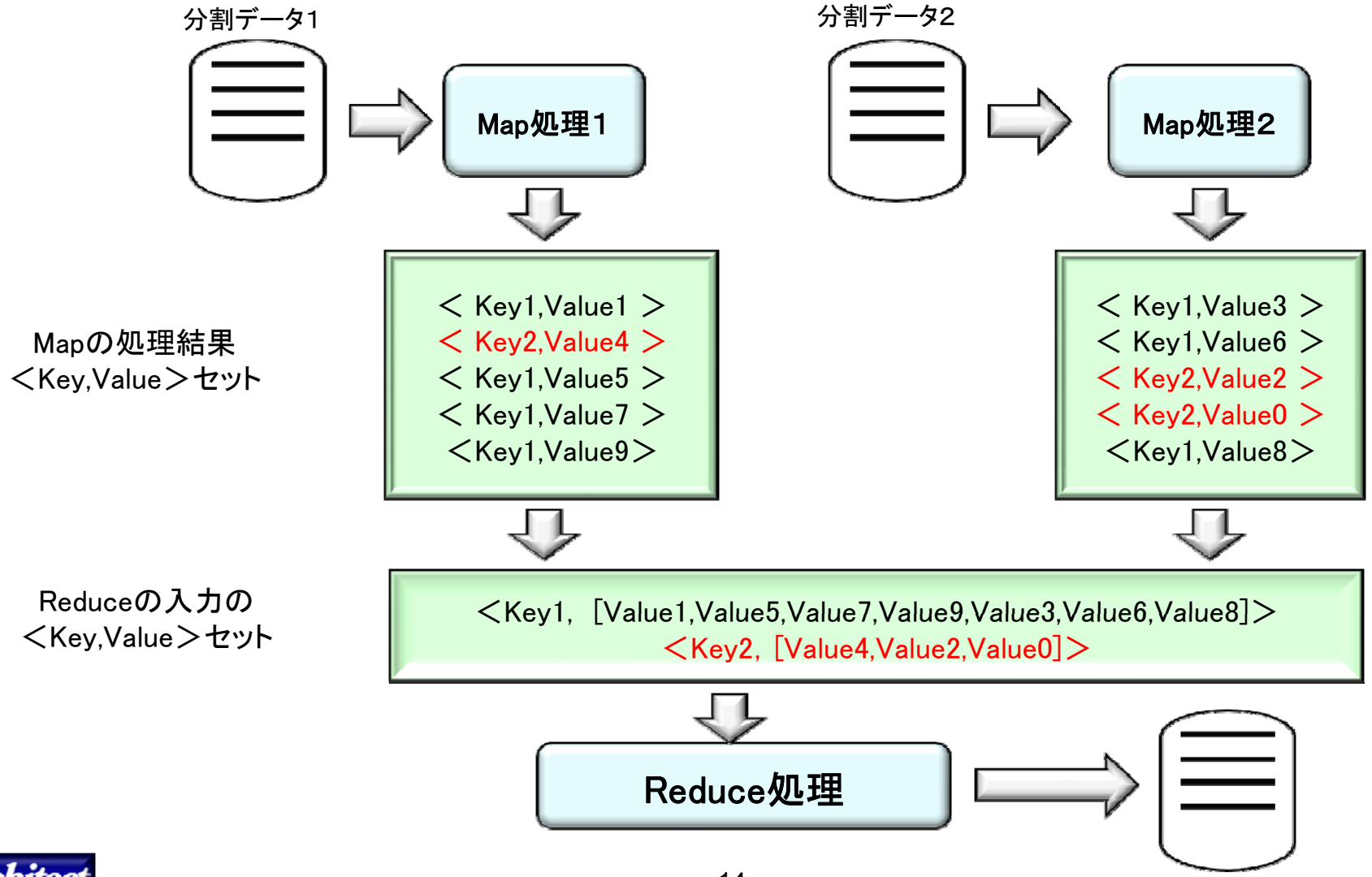
- オープンソース
- データの並列処理による時間短縮
- 並列化可能な大量データの処理に向いている
- 基幹業務での長時間バッチ処理にも適用可能
- 更新処理には向かない



Copyright 2013 DataArchitect Co.,Ltd.

MapReduceに向いているデータ

- Hadoopが扱うデータは、キーが重複しているデータ
- パラレル処理で同一キーデータを集約（Map）してReduceする



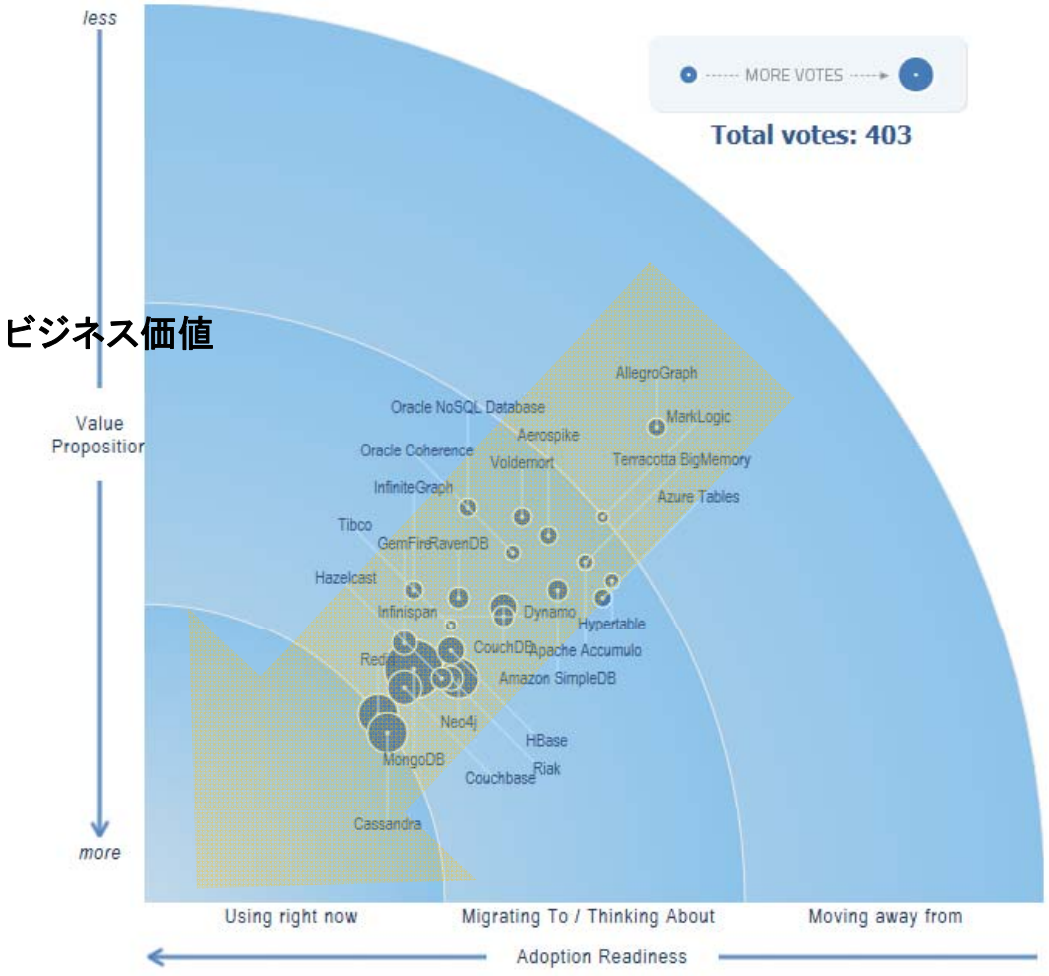
Copyright 2013 DataArchitect Co.,Ltd.

NoSQL分類

形式		特徴	主な利用法	プロダクツ
	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	・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	・Memcache ・Redius ・Riak ・**DynamoDB**
	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	・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	・Hbase ・**Cassandra** ・**Hypertable**
	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	・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	・**MongoDB** （10Gen） ・**CouchDB** ・Couchbase ・MarkLogic ・eXist-db
	グラフ指向型	関連の強い課題に対応	・SNS ・不正検出 ・強い関連をもつデータ	・**Neo4J** ・AllegroGraph ・Bigdata （RDF store） ・**InfiniteGraph** （Objectivity）

Copyright 2013 DataArchitect Co.,Ltd.

NOSQL　voting

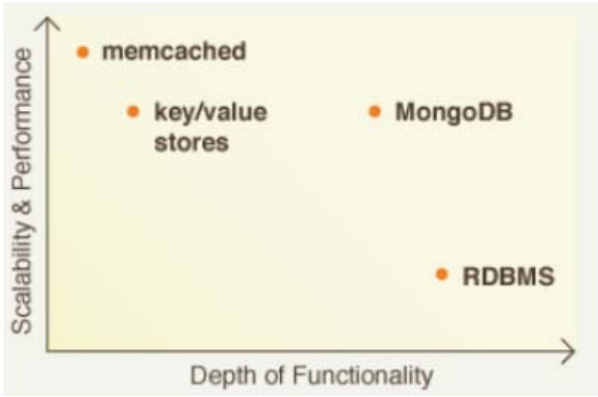


ビジネス価値

実現可能性

■ NOSQLの現状

http://www.infoq.com/jp/articles/State-of-NoSQL



出典：http://www.infoq.com/research/nosql-databases

Copyright 2013 DataArchitect Co.,Ltd.

NOSQL voting

Document Databases

MongoDB: MongoDB is an open-source document oriented database.

CouchDB: Apache CouchDB is a database that uses JSON for documents, JavaScript for MapReduce queries, and HTTP for an API.

Couchbase: NoSQL document database based on JSON model.

RavenDB: RavenDB is a document-oriented database based on .NET language.

MarkLogic: MarkLogic NoSQL database is used to store XML-based, document-centric information. It supports schema flexibility.

Other Document Database

Graph Databases

Neo4j: Neo4j is a property graph database; supports ACID transactions.

InfiniteGraph: Graph database used to persist and traverse relationships between objects, supports distribute data stores.

AllegroGraph: AllegroGraph is a graph database that uses memory utilization in combination with disk-based storage for scalability, supports SPARQL, RDFS++, and Prolog reasoning.

Other Graph Database

Key Value Data Stores

Riak: Riak is an open source, distributed key value database, supports data replication and fault-tolerance.

Redis: Redis is an open source key-value store. Supports master-slave replication, transactions, Pub/Sub, Lua scripting, Keys with a limited time-to-live.

Dynamo: Dynamo is a key-value distributed data store. It is directly implemented as Amazon DynamoDB; used in Amazon S3 product.

Oracle NoSQL Database: Key-value NoSQL database from Oracle. It supports ACID transactions and JSON.

Voldemort: Distributed key-value storage system with the data replication and partitioning.

Aerospike: Aerospike database is a key-value store; supports hybrid memory architecture and data integrity with strong or tunable consistency.

Other Key Value Data Store

Columnar Databases

Cassandra: Cassandra is column database that supports data replication across multiple data centers. Its data model offers column indexes, log-structured updates, support for denormalization, materialized views, and built-in caching.

HBase: Apache HBase is an open-source, distributed, versioned, column-oriented store modeled after Google's Bigtable. It provides Bigtable-like capabilities on top of Hadoop and HDFS.

Amazon SimpleDB: Amazon SimpleDB is a non-relational data store that offloads the work of database administration. Developers store and query data items using web services requests.

Apache Accumulo: Apache Accumulo sorted, distributed key/value data store created based on Google's BigTable design and is built on top of Apache Hadoop, Zookeeper, and Thrift technologies.

Hypertable: Hypertable is an open source, scalable database, also modeled after Bigtable; supports sharding.

Azure Tables: Windows Azure Table Storage Service offers NoSQL capabilities for applications that require storage of large amounts of unstructured data. Tables can auto-scale to store up to several terabytes of data. They are accessible via REST and managed APIs.

Other Columnar Database

In-Memory Data Grids

Hazelcast: Hazelcast CE is an open source data distribution platform. It allows the developers to share and partition the data across the database cluster.

Oracle Coherence: Oracle's in-memory data grid solution that provides fast access to frequently used data. Coherence supports event capabilities and dynamic partitioning of data.

Terracotta BigMemory: Distributed in-memory management solution from Terracotta. The product includes an Ehcache interface, Terracotta Management Console and BigMemory-Hadoop Connector (early access).

GemFire: VMware vFabric GemFire is a distributed data management platform and provides elastic in-memory data management, replication, partitioning, data-aware routing, and continuous querying.

Infinispan: Infinispan is a Java based open source key/value NoSQL datastore and distributed data grid platform. It supports transactions and peer-to-peer as well as client/server architecture.

GridGain: Distributed, object-based, in-memory, SQL+NoSQL key-value database. Supports ACID transactions.

GigaSpaces: GigaSpaces in-memory data grid (the Space) serves as the system of record for the applications and supports a variety of caching scenarios.

Tibco: ActiveSpaces product from Tibco provides an infrastructure to create virtual data caches from the aggregate memory of participating nodes in the cluster and to scale as nodes join and leave.

Other In-Memory Data Grid

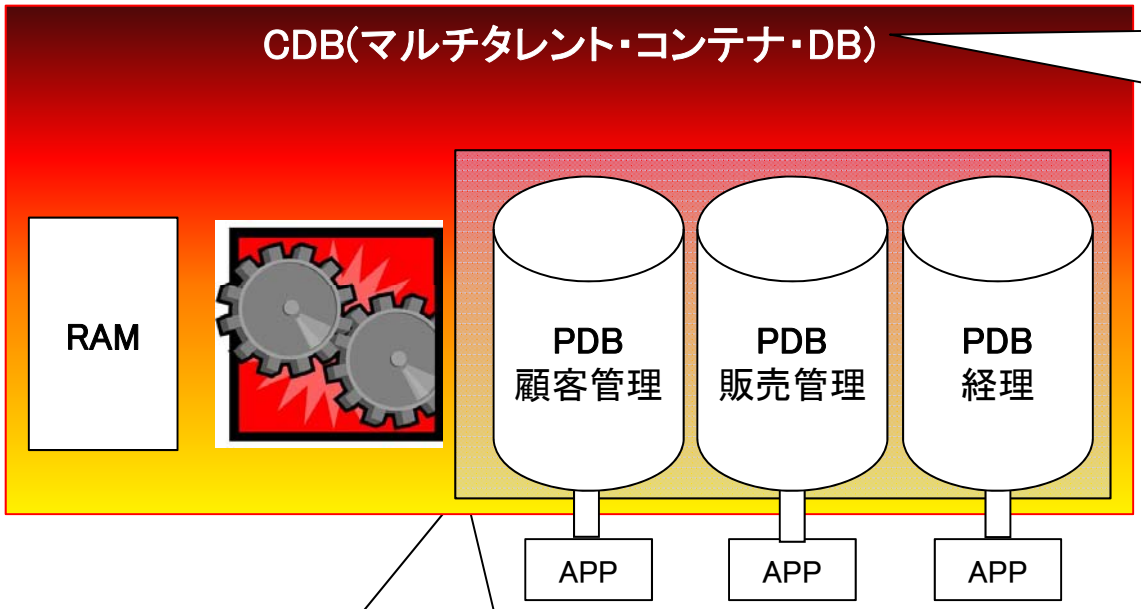
Copyright 2013 DataArchitect Co.,Ltd.

RDBMSの革新

- パーティション化
  - ◆大量データ、平行Run、パーティション単位でのバックアップや切換
- マルチテナント
- カラムナー
  - ◆DWHでの検索処理
- NOSQL対応
  - ◆Map&Reduce結果のRDB取込み
  - ◆Oracle NOSQL
  - ◆IBM MongoDB
- 改訂SQLへの対応
  - ◆ISO:SQL2011

Copyright 2013 DataArchitect Co.,Ltd.

Oracle12Cマルチテナント



マルチテナント・アーキテクチャを
実現するDB

マルチテナントを
データベースのレイヤーで実現

・複数の顧客を単一のDBに統合
・サーバーや運用管理にかかるコストを削減
→クラウド環境を利用したい企業
→アプリケーションごとにDBを分けて運用している企業

プラガブル・データベース

ORACLEデータベース内における
論理的なセット

ユーザーやアプリケーションから
は通常のDBと同じように扱える

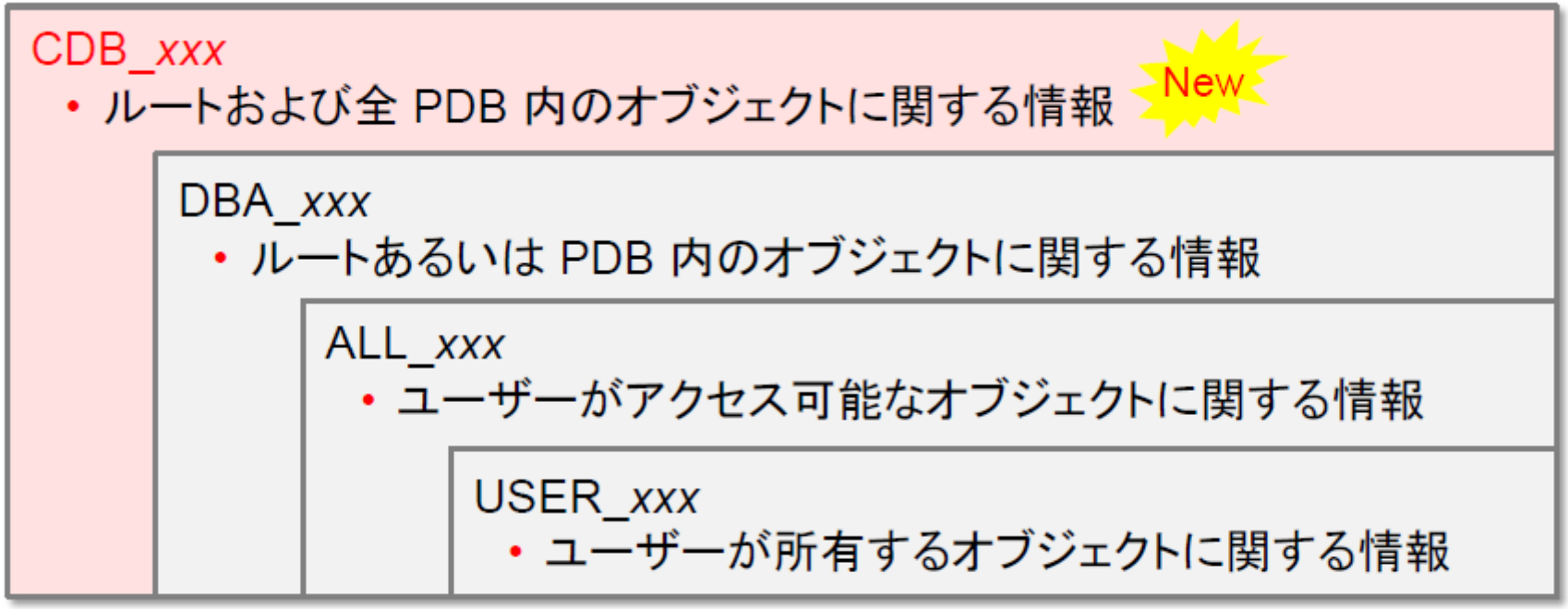
Copyright 2013 DataArchitect Co.,Ltd.

Oracle12Cでのマルチテナント DB

データ・ディクショナリ・ビュー

データ・ディクショナリ・ビューの種類

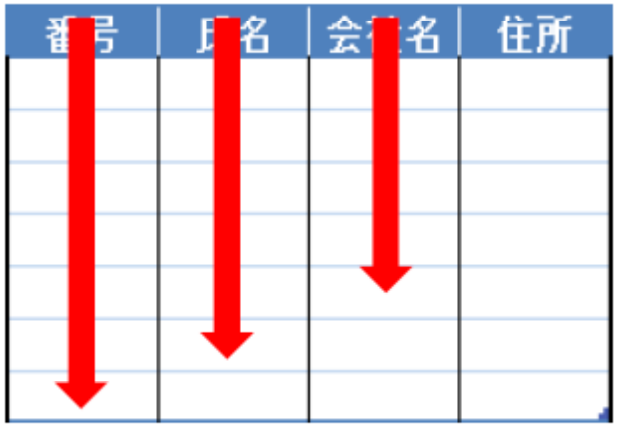
- データ・ディクショナリ・ビューの種類には、次の 4つがある

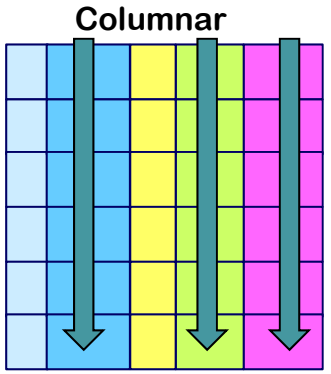


Copyright 2013 DataArchitect Co.,Ltd.

カラムナー

カラム型データベースが高速な理由





- 列指向ごとにデータが保存され、処理されるのがカラム型データベース
- 同一データ型、似たデータが並ぶため、圧縮率が高い
- インデックスを付けなくとも、データのソート、集計、グループ化などが高速
- 列ごとに分散処理が容易
- ただし行ごとの操作が必要となるデータの追加更新は苦手

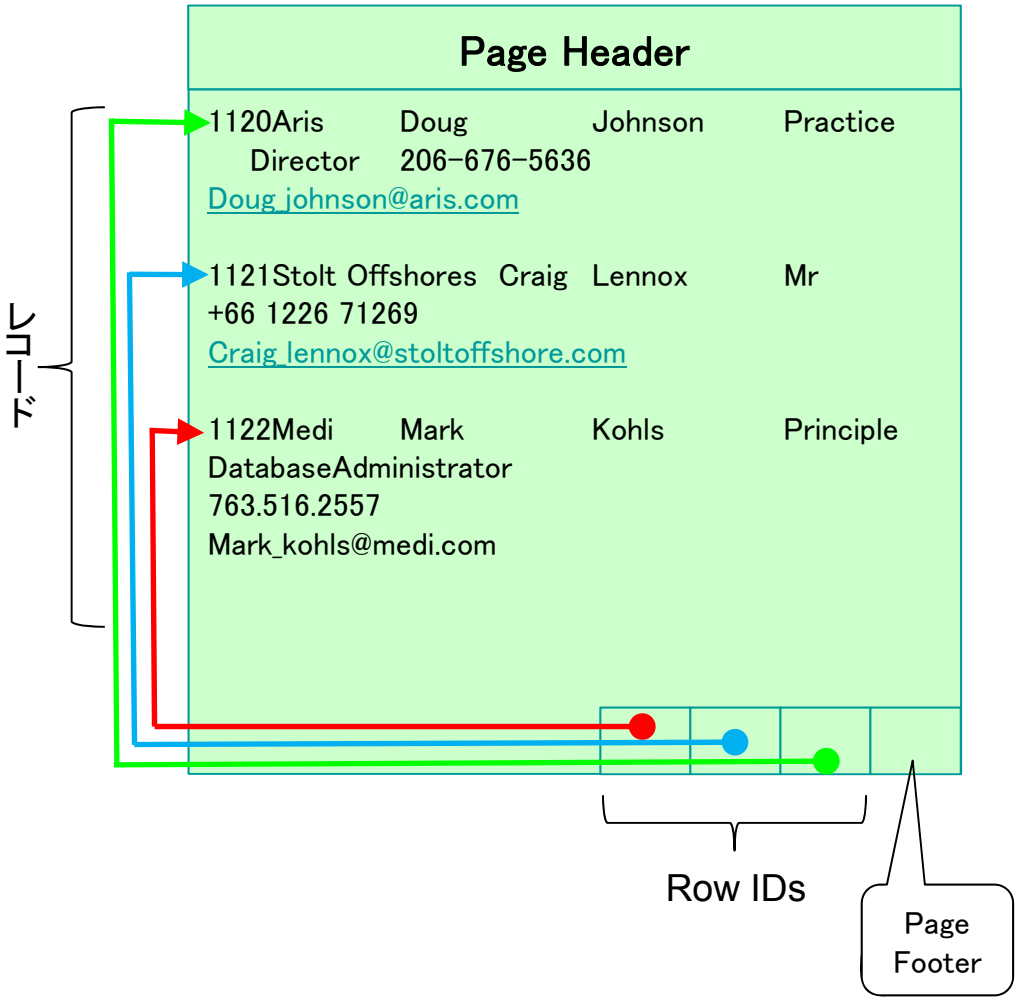
➡ ここ数年で急速にOLAP/BI用の仕組みとして普及へ

DB2 V11、SQL Server2012、SybaseIQ

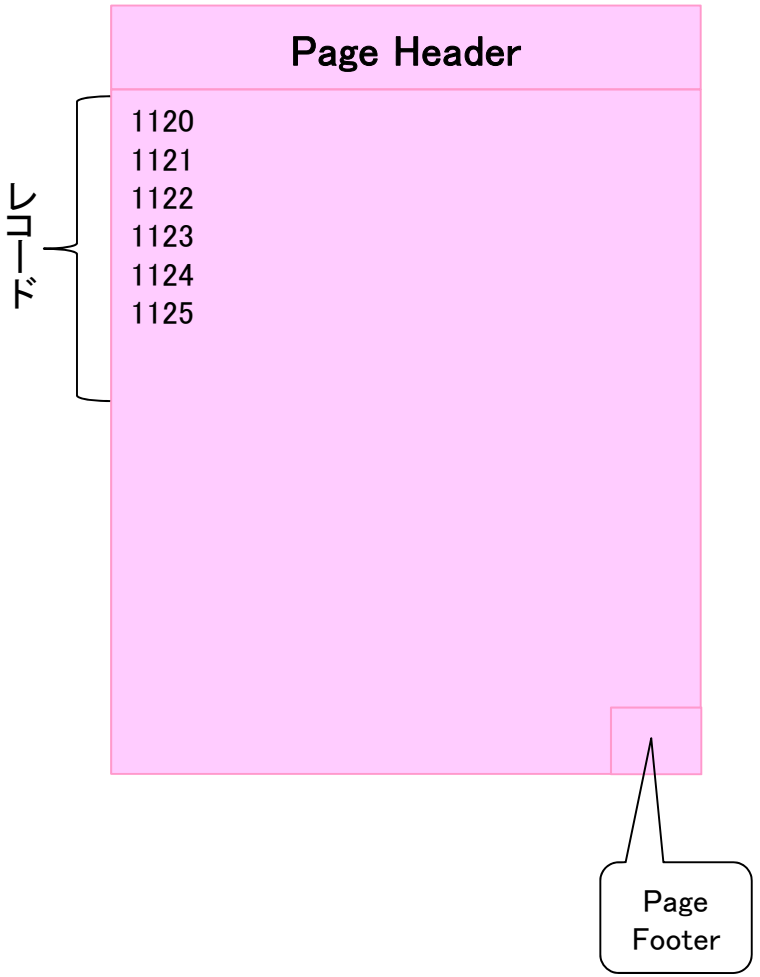
Copyright 2013 DataArchitect Co.,Ltd.

ページマップ対比

従来型RDBでのページマップ



カラムナーDBでのページマップ



出典：Columnar Database, William McKnight氏(Mcknight Consulting Group)

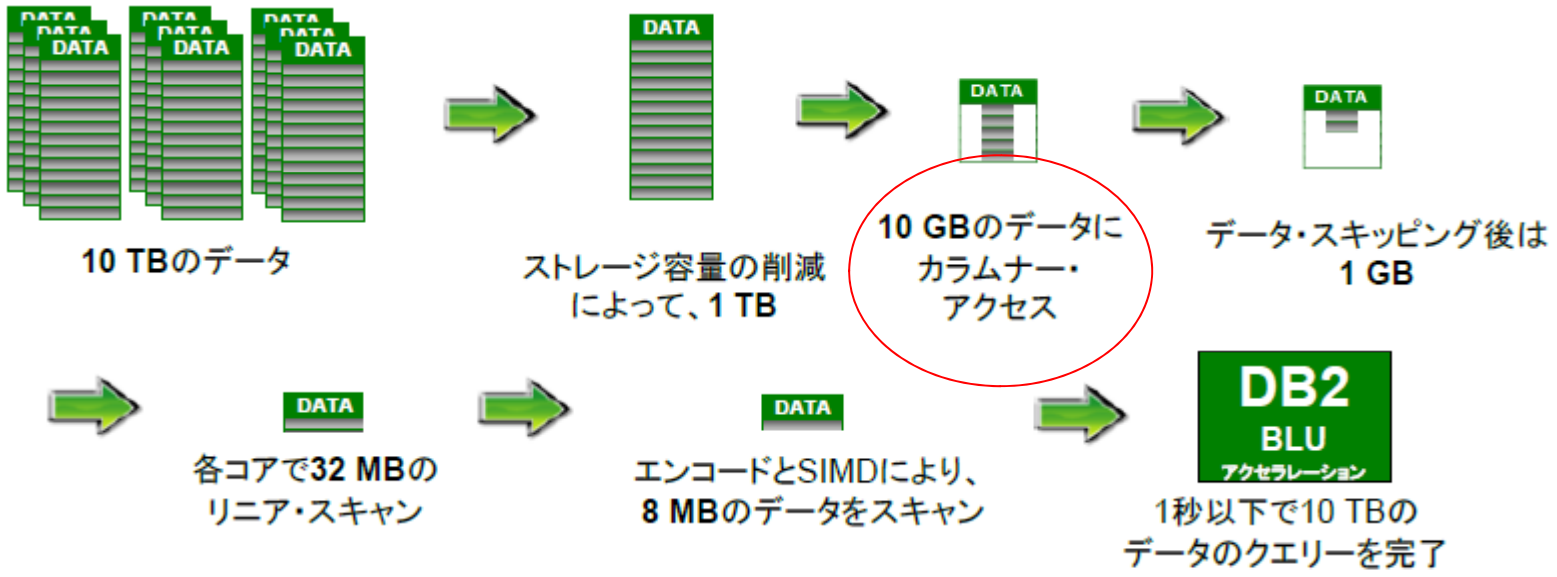
Copyright 2013 DataArchitect Co.,Ltd.

カラムナー処理による高速アクセス

DB2 BLUアクセラレーションのイメージ

10 TBのデータへのクエリーを1秒以内で完了

- システム: 32のコア、100の列を持つ10 TBのテーブル、10年分のデータ
- クエリー:「2010年には何件のトランザクションが存在したか？」
  - `SELECT COUNT(*) from MYTABLE where YEAR = '2010'`
- 実現可能な結果: 10 TBのデータへのクエリーを1秒以内で完了!
  各CPUコアは、たった8 MBのデータを検証すればよい

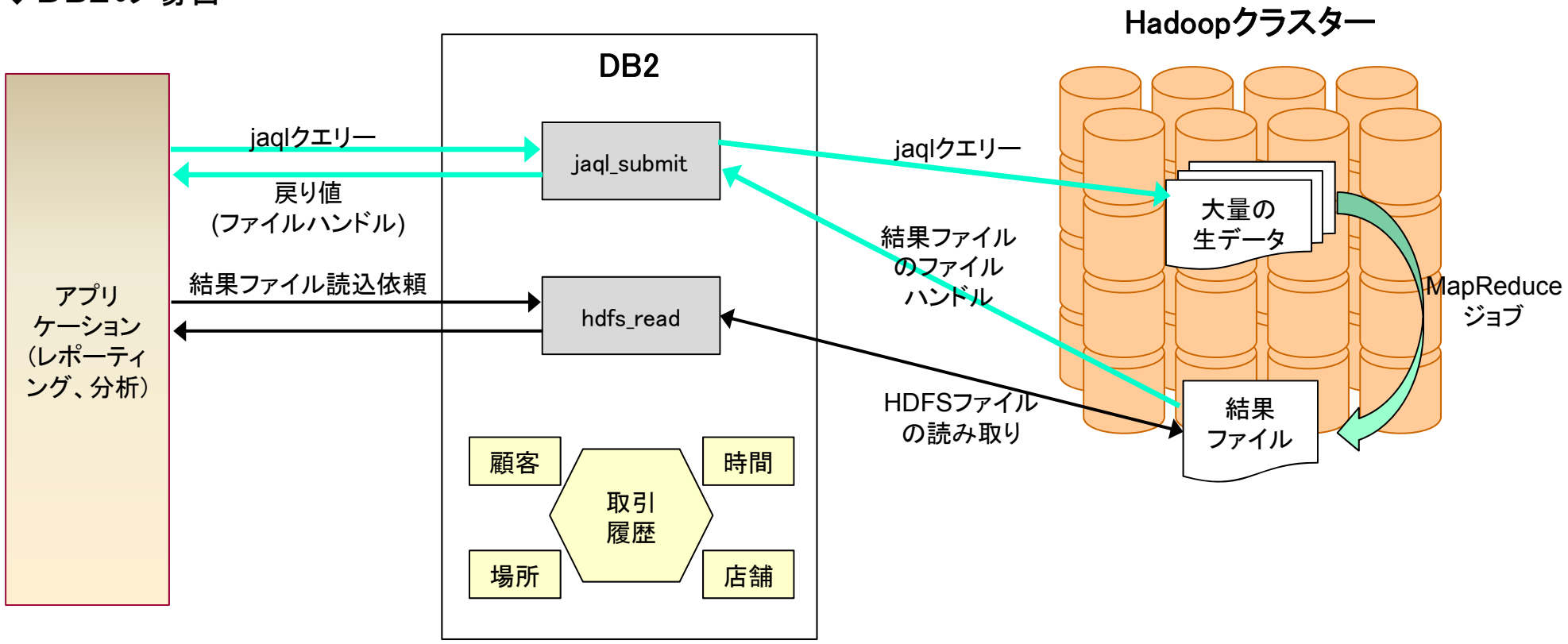


Copyright 2013 DataArchitect Co.,Ltd.

RDBMSとHadoopの連携

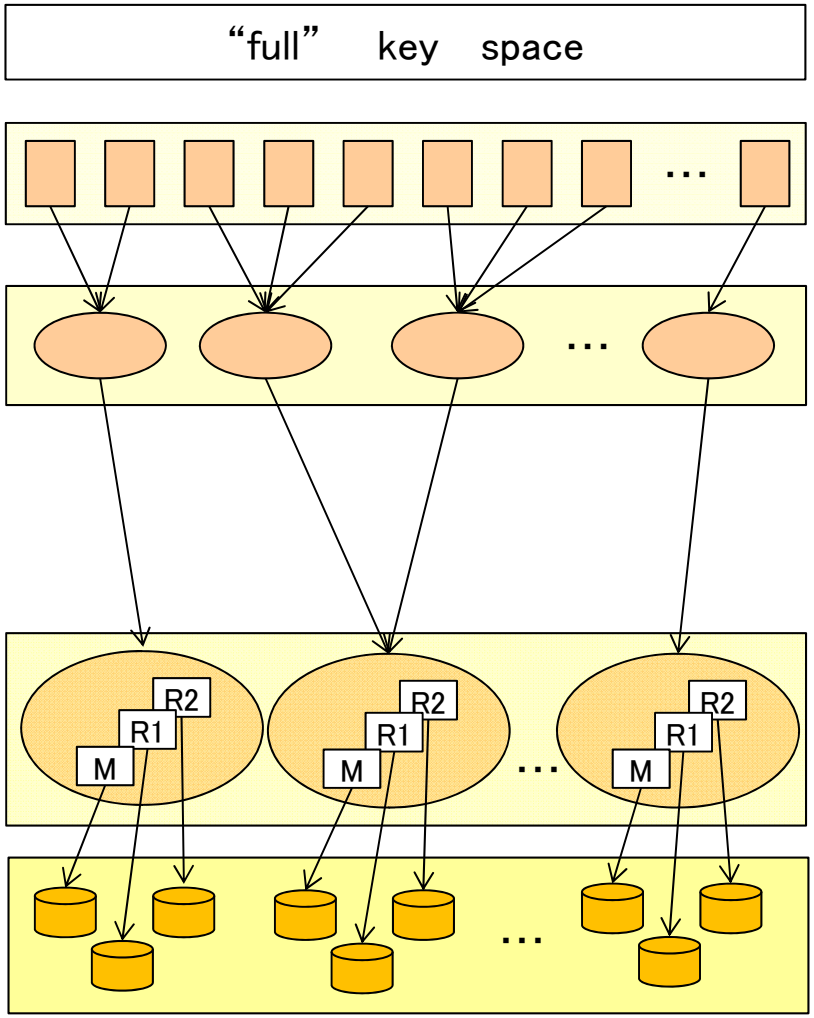
■ Hadoopで処理した結果を呼び出し、大量データの分析をリアルタイムに近づける

◆DB2の場合



Copyright 2013 DataArchitect Co.,Ltd.

Oracle NoSQL Database概要



Key-Value Data ⇒ Partitions:

- Key spaceを複数のハッシュバケットに分割(partitions)

Partitions ⇒ Replication Group:

- パーティションの集合をレプリケーショングループにマッピング(データサブセットのロジカルコンテナ)

Replication Groups ⇒ Storage Nodes:

- レプリケーショングループはHAとリードのスケーラビリティを実現
- ストレージノードはそれぞれのレプリケーションノードに対応

Copyright 2013 DataArchitect Co.,Ltd.

NOSQLの進化

- トランザクション管理
- JSONスキーマ
- SQLライクのアクセス
- セキュリティ強化
- DBMSベンダーの参入

Copyright 2013 DataArchitect Co.,Ltd.

クラウド化の進展

- エンタープライズシステムのクラウド化が進行している
- AWSでのDB関連サービス
  - ◆RDB
    - Oracle、MySQL、SQL Server
  - ◆DWH
    - Redshift
  - ◆NOSQL
    - Dynamo
    - Elastic MapReduce(EMR)　Hadoop

Copyright 2013 DataArchitect Co.,Ltd.

2. DB設計現場で遭遇する身近な課題

- 長期保存、オンデマンド検索
  - 大量データ分散配置：パーティション化
  - 並列処理：パーティション化、Hadoop
- 管理対象DBの多種・多量化（DB環境の数、DBMSのバリエーション）
  - マルチテナントで解決
  - クラウド、オンプレミス
- ビジネス変化への対応
  - グローバル対応
  - ビジネスの垣根がなくなりつつある中での顧客管理
  - マスターデータ管理
- オペレーショナル
  - 履歴管理（長期保存、有効日管理）
  - データ品質向上
  - データ移行
  - 性能改善

Copyright 2013 DataArchitect Co.,Ltd.

パーティション化

- オンラインデータ量の削減
- 月単位でのオフライン／オンライン化
- 大量バッチ処理の並列化
- 商用RDBでのサポート
  - ◆Oracle、SQL Server、DB2、HiRDB、…
- パーティションキー定義に注意

Copyright 2013 DataArchitect Co.,Ltd.

マルチテナント化

- Oracleマルチテナント
  - ◆マルチDB/インスタンス・・・EnterpriseでのDB管理の簡素化を目指して
- クラウドサービス
  - ◆クラウドへのシフト
  - ◆プライベートクラウド化

Copyright 2013 DataArchitect Co.,Ltd.

グローバル対応

- 主キーに「会社コード」を付与
  - ◆マルチテナント化の考え方：会社の枠を取り払っても一意性を保つ
  - ◆M&A
- 個人レベルではマイナンバー制
  - ●社会保険、税、住民サービス、．．．

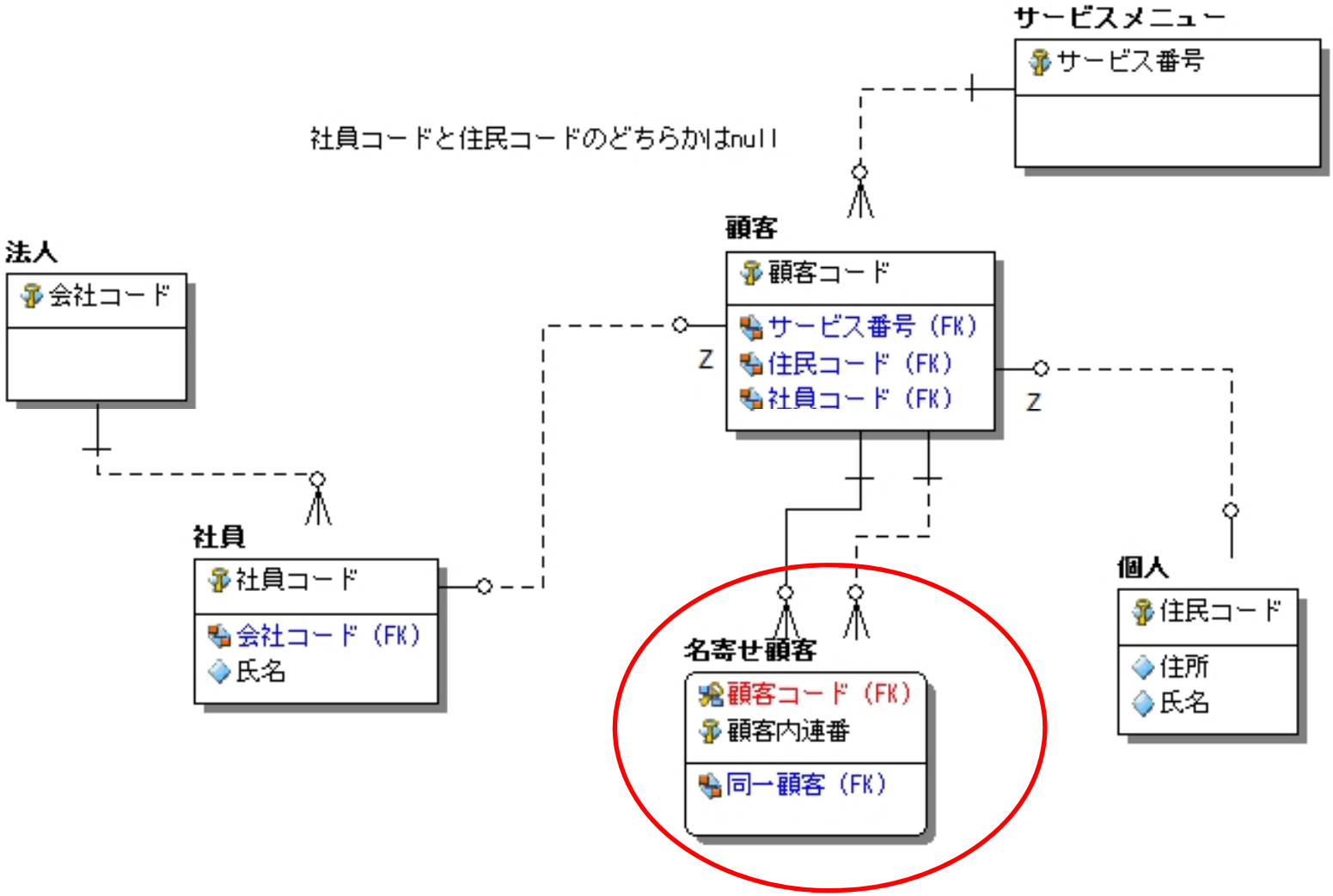
Copyright 2013 DataArchitect Co.,Ltd.

ビジネスの垣根がなくなりつつある中での顧客管理

- 真の顧客は誰か
  - ◆B2B企業がB2Cで消費者ニーズを探る
- 競合者が顧客、アライアンスとなる
  - ◆法人（取引先）の多様化
- 法人と個人の顧客
  - ◆個人の多様化
  - ◆法人に所属する個人
  - ◆プライベートでの個人
- サンプルモデル

Copyright 2013 DataArchitect Co.,Ltd.

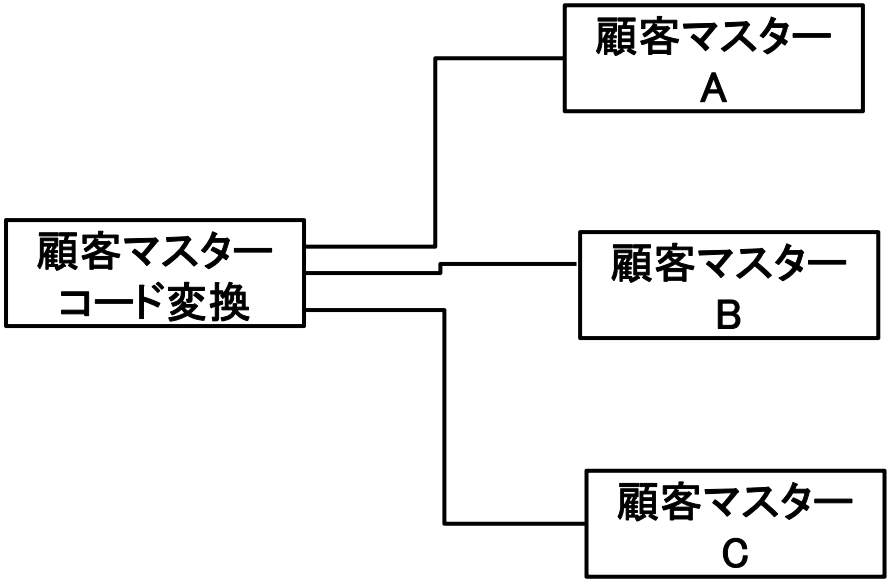
顧客管理



Copyright 2013 DataArchitect Co.,Ltd.

マスターデータ管理

■ 個別に存在しているマスターへの対処



Copyright 2013 DataArchitect Co.,Ltd.

データ品質維持の組み込み

- 量から質が問われるようになる
- 質が問われる時のために、今から準備しておく
  - ◆必要となった時では遅い
- DB設計（モデリング）での品質向上施策

データ品質低下	データモデルでの向上施策
誤ったデータが混入している	✓掛け持ちデータ項目を作らない ✓NULL項目を回避する ✓主キーだけで統合しない（発生タイミングの異なるデータの混在抑止）
データが重複している	✓One Fact in One Place原則 ✓移行データは期間限定で保持 ✓重複データの導出元の明示
必要なデータがない	✓誤った正規化 ✓マスターの断面管理

Copyright 2013 DataArchitect Co.,Ltd.

データ移行

- 文字コード問題
  - ◆UTFコード化による予期せぬ桁あふれ
    - 現行　char(1)　でドキュメント上有効値：0,1,2となっているが、半角カナのデータが存在した。
    - Char(n)　−→nchar(1)　またはセマンテックcharに変換
- コードバリデーションの統一化
  - ◆コード値の有効値がシステムによって異なっている（現行）
  - ◆データドメインの定義＝データモデリング
- データ洗浄

Copyright 2013 DataArchitect Co.,Ltd.

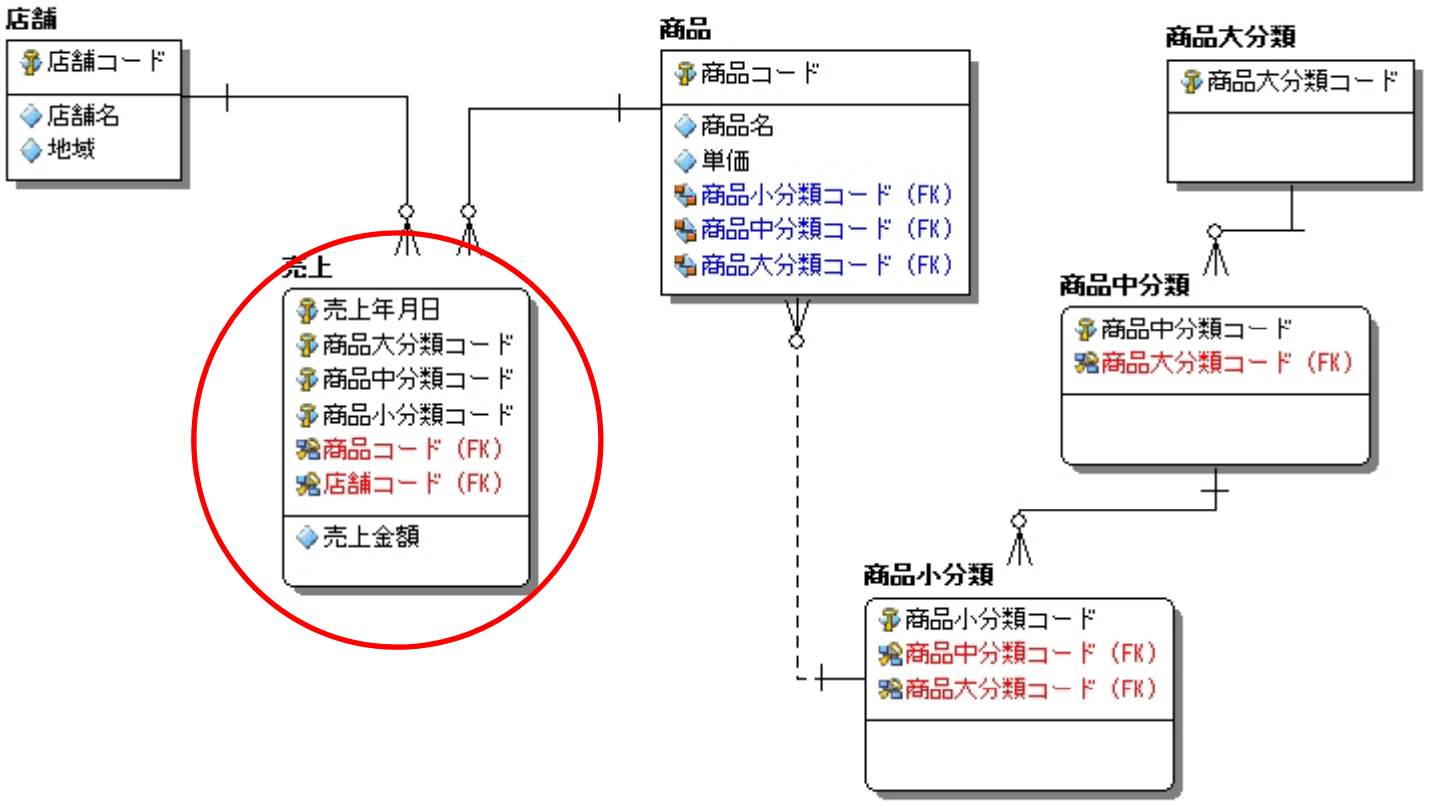
インデックス設計

- まだまだ現実では、性能維持のために重要
  - 実行計画によるチェック
  - 統計情報
  - 更新処理とのトレードオフ
  - インデックス・スペース
  - ヒント句
  - インデックスオンリー処理
    - インクルードインデックス(SQL Server)
  - カラムナー
    - SQL Server2008：列ストアインデックス
    - DB2　V10 BLUアクセラレーション
- 主キーとクラスター化キー
- ナチュラルキーと人工キー(ID)

Copyright 2013 DataArchitect Co.,Ltd.

インデックス設計

- 主キーとクラスター化キー
  - DB2,SQL Server:Primary Key定義がCluster Keyとなる（既定）ので注意が必要
  - クラスター化キーのみで他の非クラスター化キーを定義しないのが最良
    - そのために、本来の主キーにクラスタ化キー用の項目を付加する場合がある。→論理モデルとしては、望ましくない。



Copyright 2013 DataArchitect Co.,Ltd.

履歴管理

■ 従来

◆ID＋有効開始・終了日で作成

◆アプリケーションで履歴データをinsert

■ ISO,SQL2011　テンポラルデータ管理

◆シンプルなテーブル構造、リレーションシップ付けが明瞭

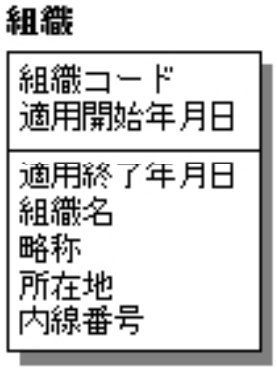
◆DB2 V10 タイムトラベル機能

- システム期間テンポラル表
- ビジネス期間テンポラル表
- ハイテンポラル表

Copyright 2013 DataArchitect Co.,Ltd.

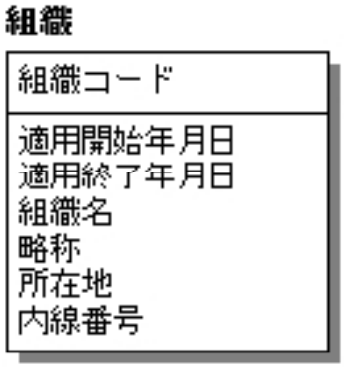
履歴管理

従来のテーブル定義




```
CREATE TABLE ‘組織’
  組織コード            CHAR(5)  NOT NULL ,
  適用開始年月日        DATE  NOT NULL,
  適用終了年月日        DATE  NOT NULL,
   組織名               VARCHR(40)  NULL,
   ・・・・
PRIMARY KEY(組織, 適用開始年月日);
```

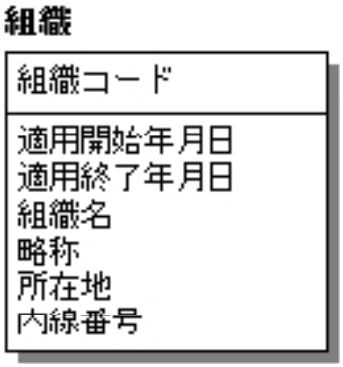
テンポラル管理機能を使用したテーブル定義




```
CREATE TABLE ‘組織’
  組織コード            CHAR(5)  NOT NULL ,
  適用開始年月日        DATE  NOT NULL,
  適用終了年月日        DATE  NOT NULL,
   組織名               VARCHR(40)  NULL,
   ・・・・
PERIOD BUSINESS_TIME (適用開始日,適用終了日),
PRIMARY KEY(組織, BUSINESS_TIME WITHOUT OVERLAPS);
```


Copyright 2013 DataArchitect Co.,Ltd.

履歴管理



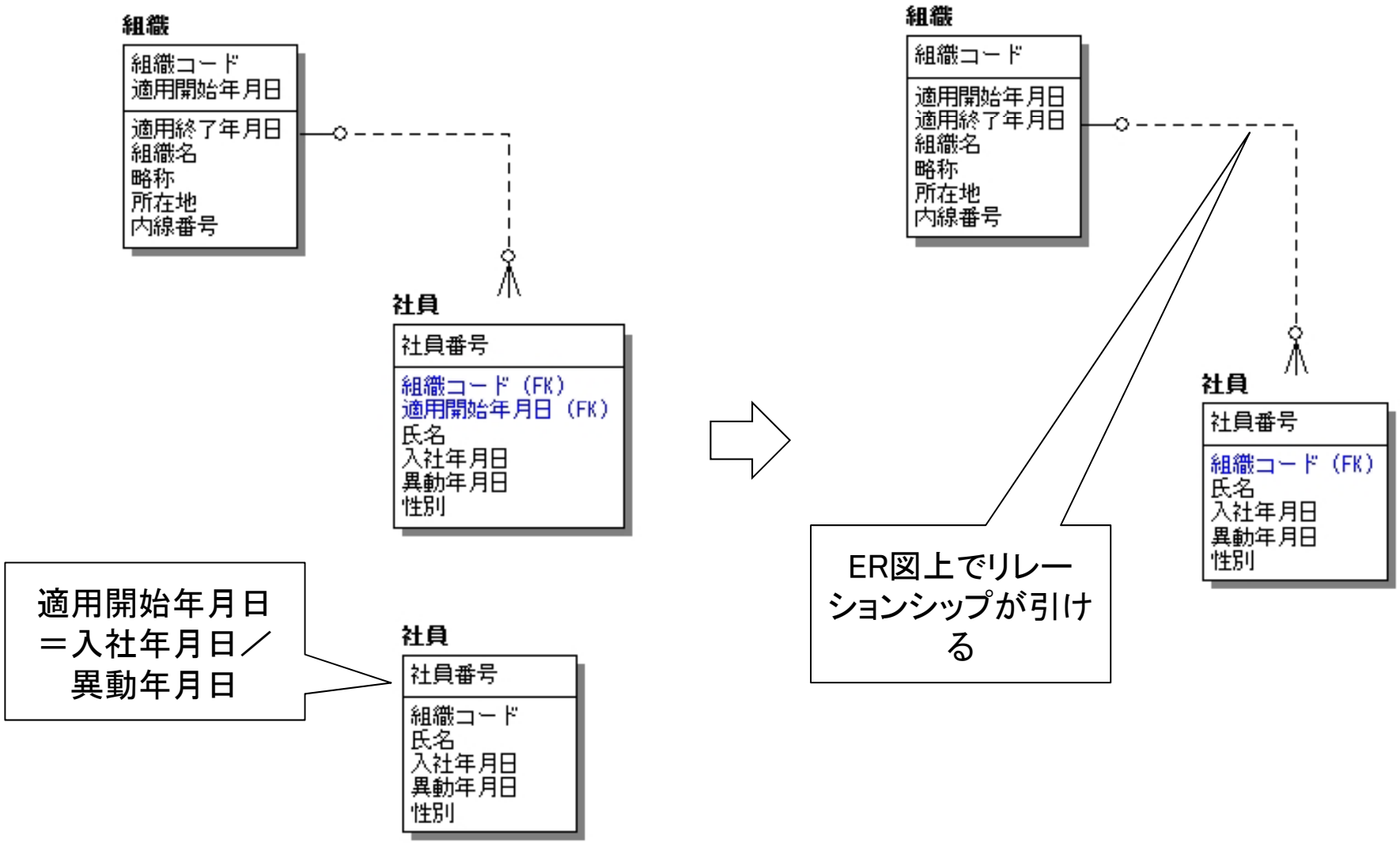
組織コード	適用開始年月日	適用終了年月日	組織名
500	19900401	19950331	電算部
500	19950401	19980331	システム部
500	19980401		IT部

```
UPADTE  組織
    FOR  PORTION  OF BUSINESS_TIME  FROM  '1996-04-01' TO '1997-04-01'
 SET  組織名   = '情報システム部'
WHERE  組織コード = '500'
```

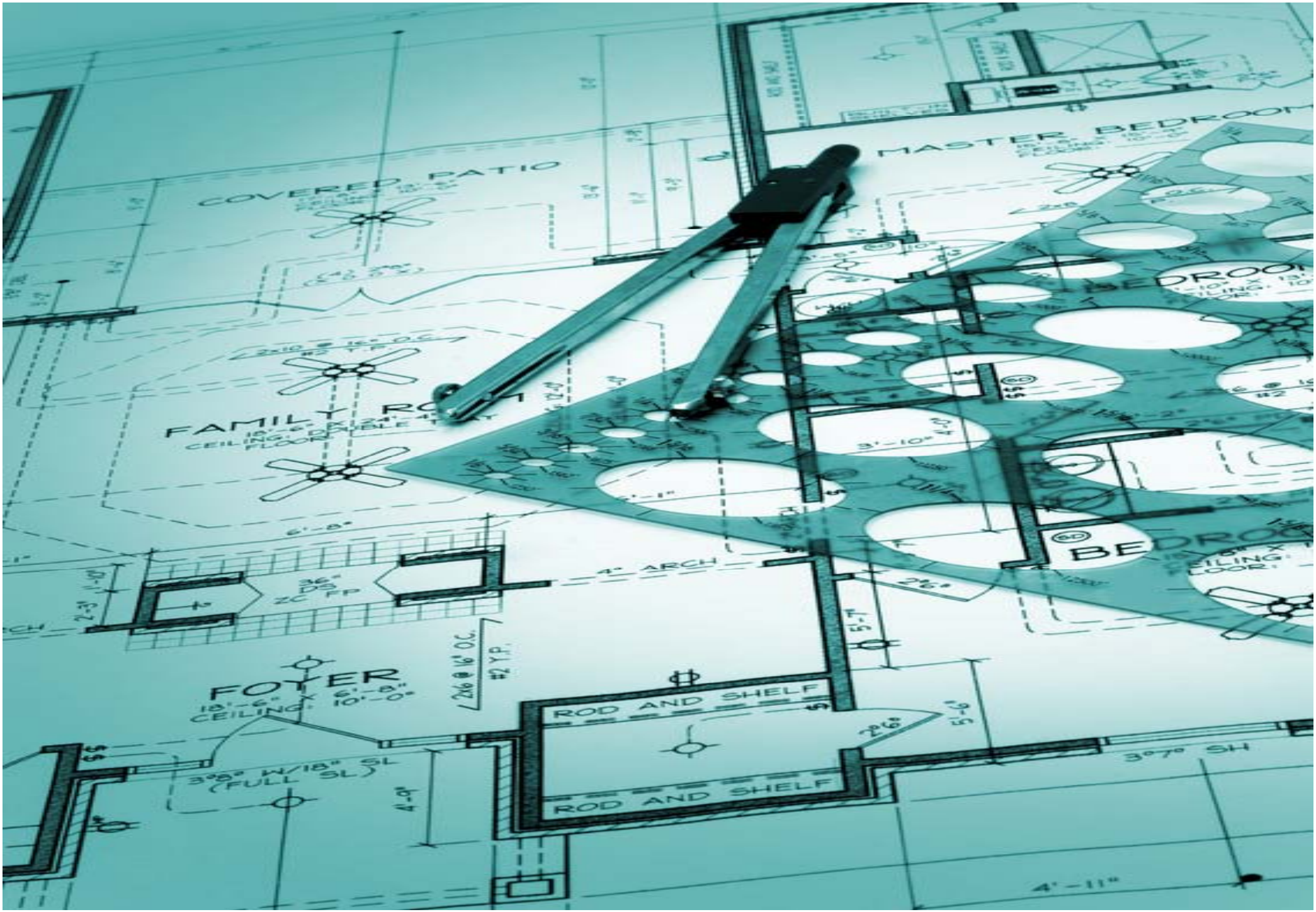
```
SELECT  組織コード,組織名   FROM 組織
    FOR   BUSINESS_TIME  AS OF   '1996-04-01'
WHERE  組織コード = '500'
```


Copyright 2013 DataArchitect Co.,Ltd.

履歴管理

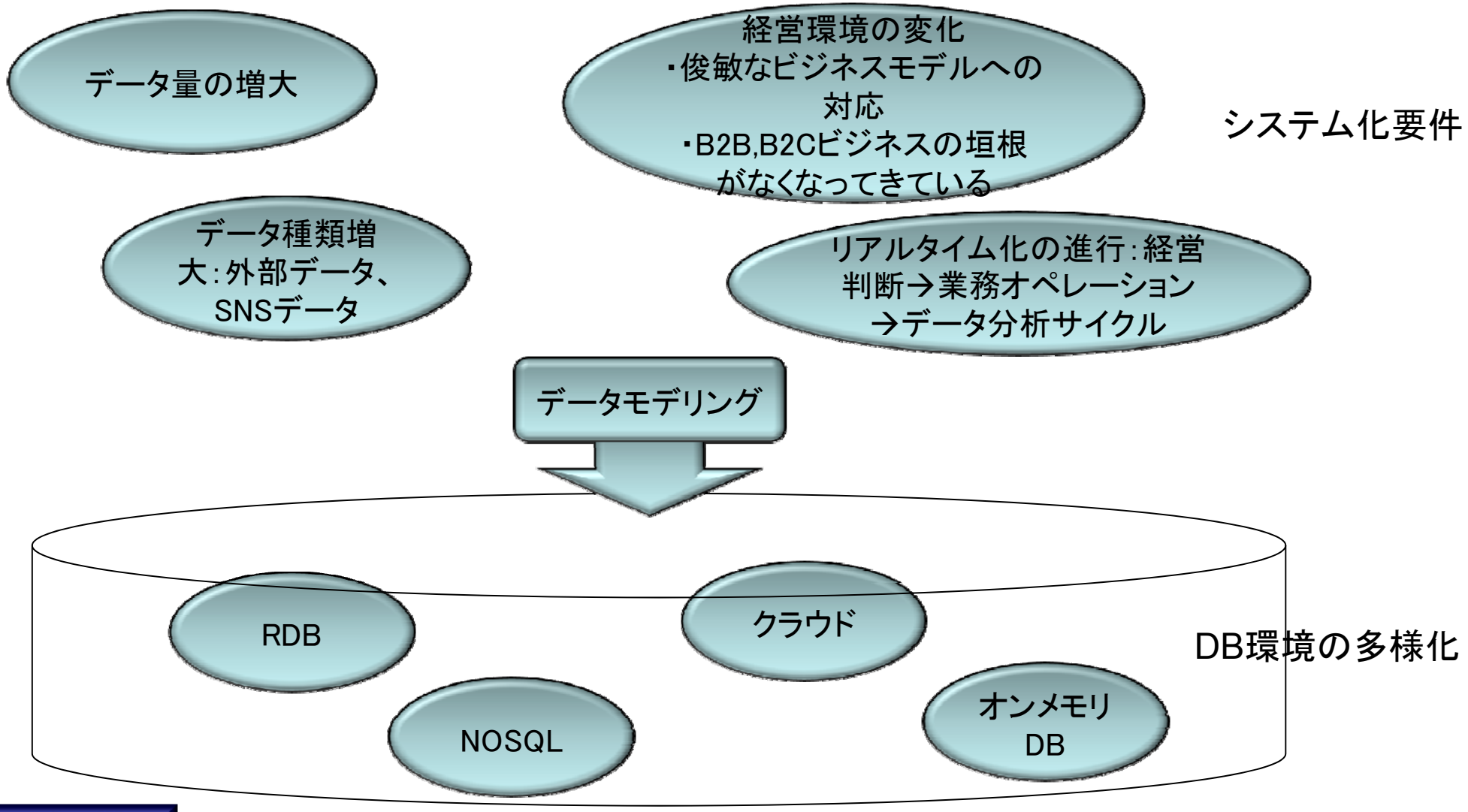


Copyright 2013 DataArchitect Co.,Ltd.


COVERED PATIO
MASTER BEDROOM
FAMILY ROOM
4" ARCH
FOYER
ROD AND SHELF
ROD AND SHELF

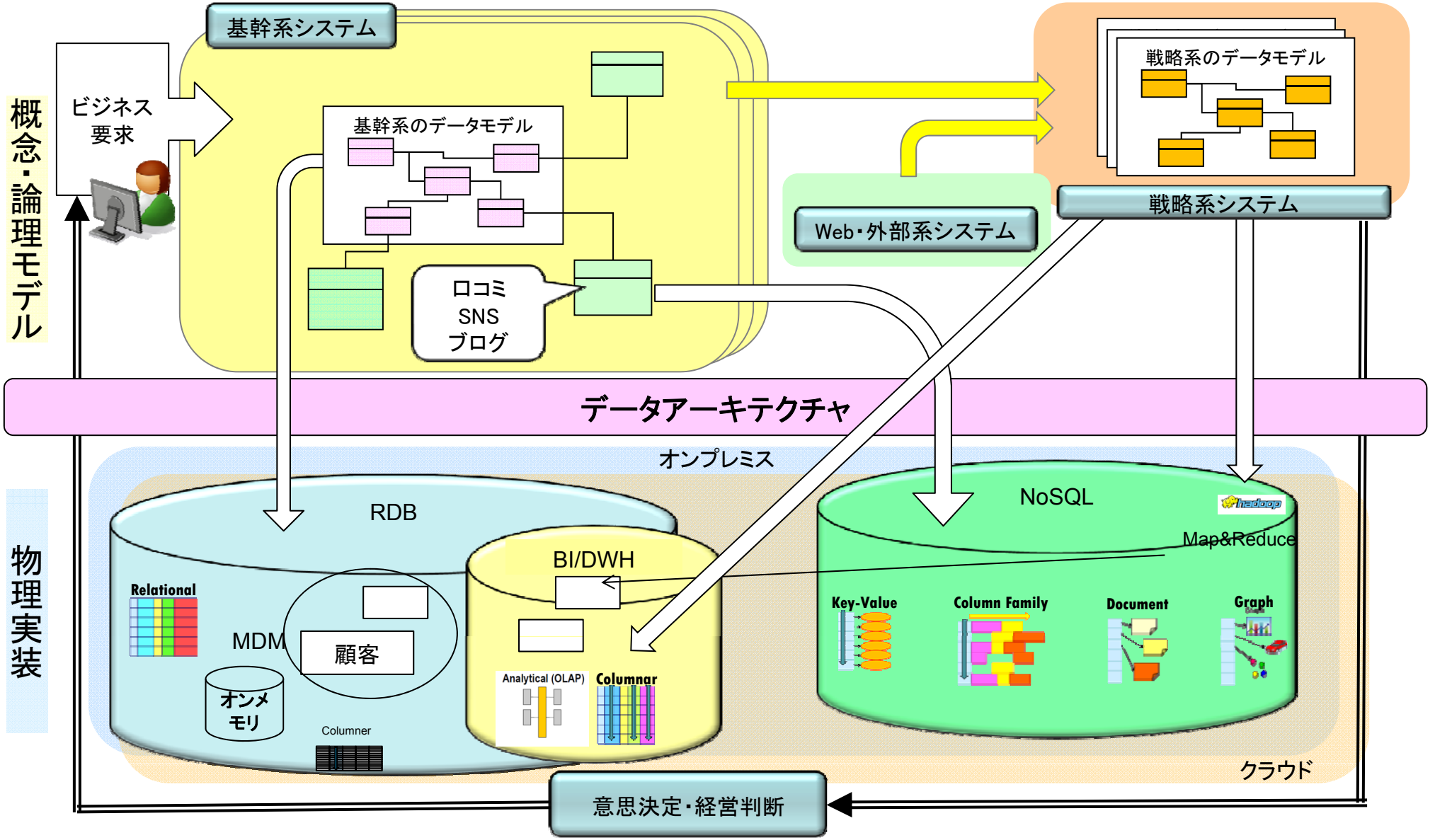
3．DB環境の多様化とシステム化要件

■ 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



Copyright 2013 DataArchitect Co.,Ltd.

4. データアーキテクチャとモデリングの価値



Copyright 2013 DataArchitect Co.,Ltd.

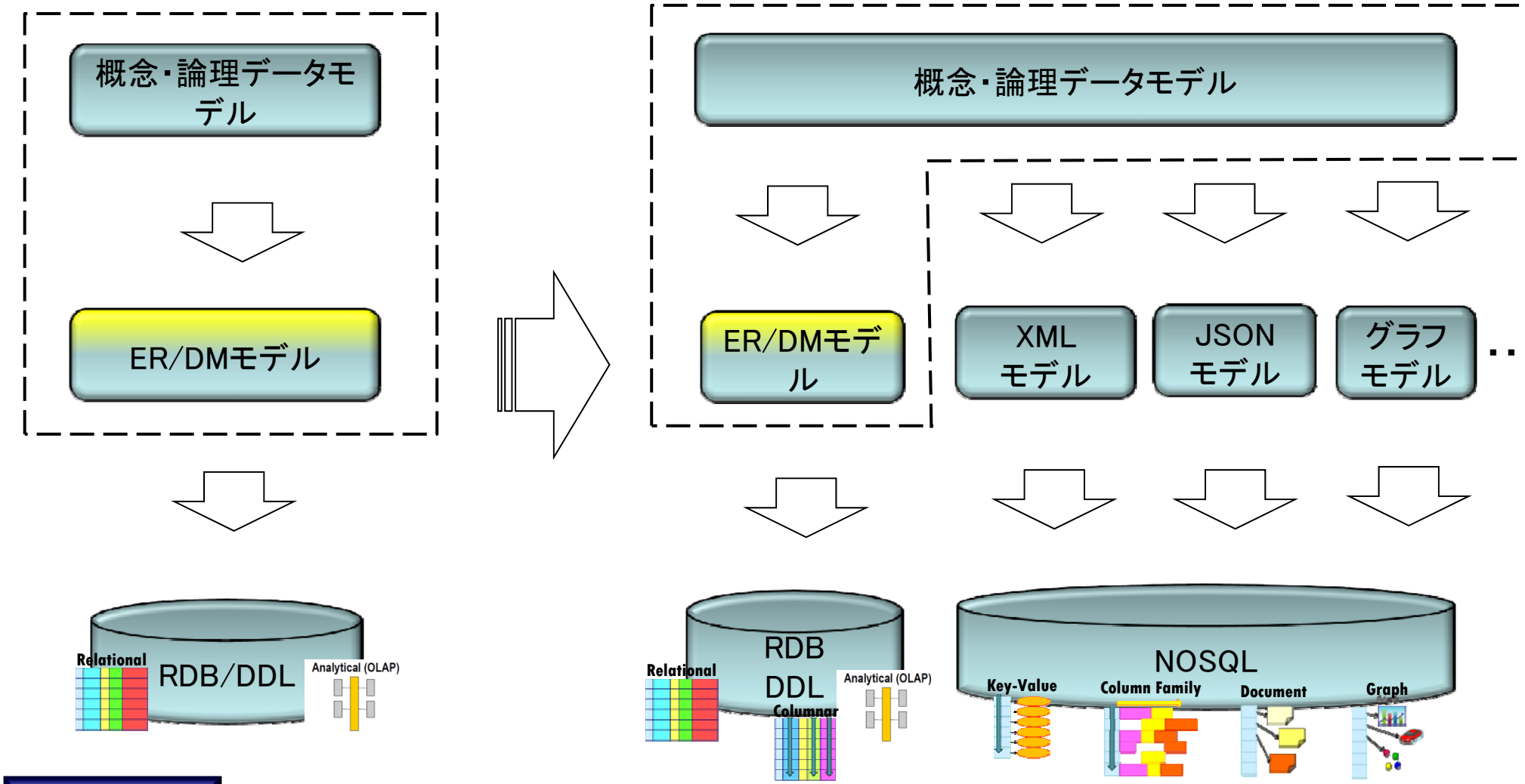
概念・論理モデルの役割

- 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

Copyright 2013 DataArchitect Co.,Ltd.

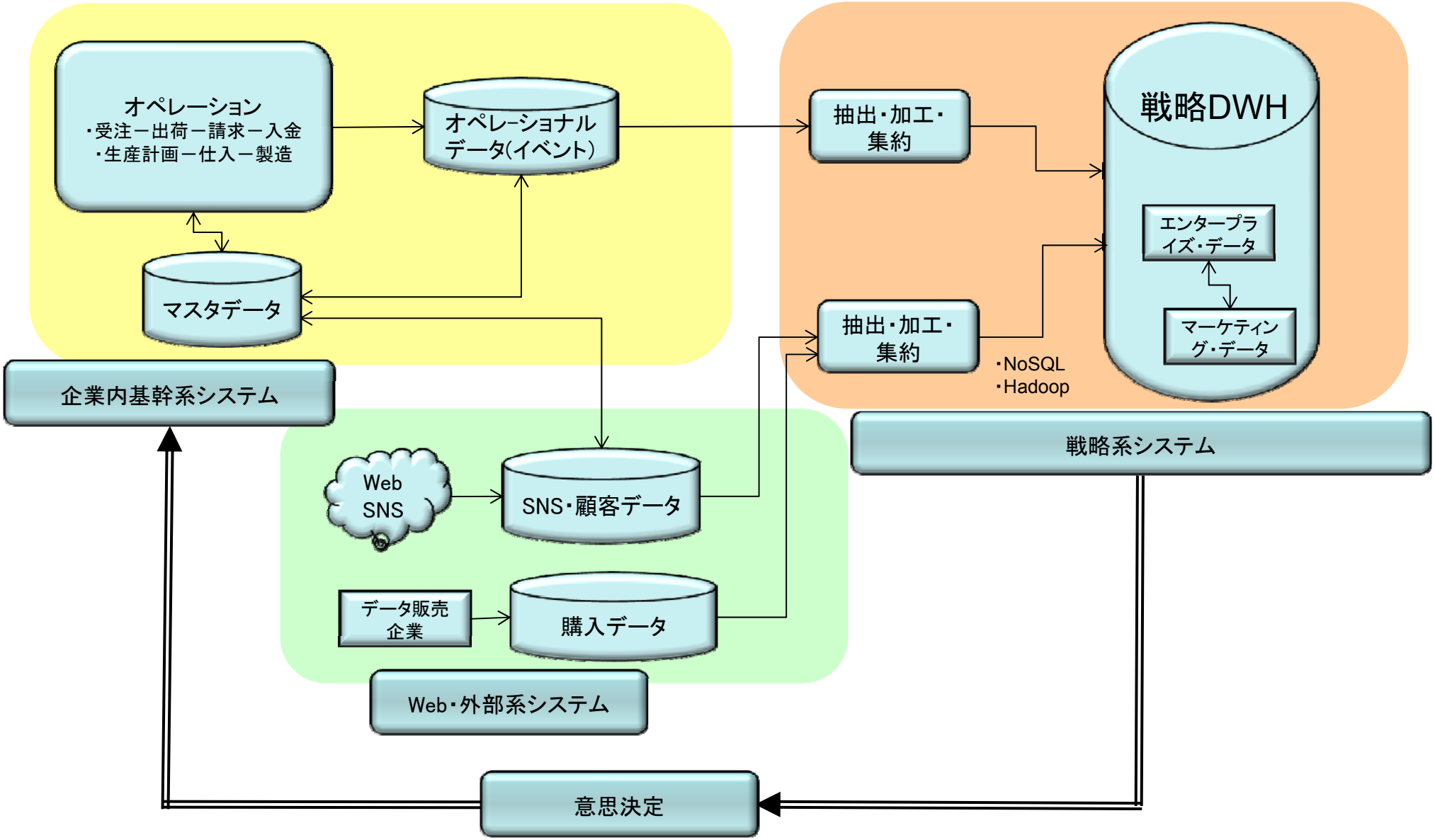
モデリングの変化

- 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



Copyright 2013 DataArchitect Co.,Ltd.

データアーキテクチャ



Copyright 2013 DataArchitect Co.,Ltd.

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きている
- 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
- リアルタイム経営
  - ◆オペレーショナルデータからの分析結果を次の意思決定へ迅速に

Copyright 2013 DataArchitect Co.,Ltd.

謝辞

ご静聴有難うございました。

講演内容に関してご質問がございましたら、下記までお問い合わせください。

http://dataarch.co.jp

E-mail : mano@dataarch.co.jp

Copyright 2013 DataArchitect Co.,Ltd.