

【A3】DatabaseGearテクニカルセッション



DEVELOPER CAMP

データモデリングを見直そう！
モデリングの基本とモデル駆動 DB 設計のポイント
日揮情報ソフトウェア（株）・モデリング スペシャリスト
山本 よしの

アジェンダ



DEVELOPER CAMP

- データモデリングの必要性
- データモデリングの基本
- データモデル駆動型開発

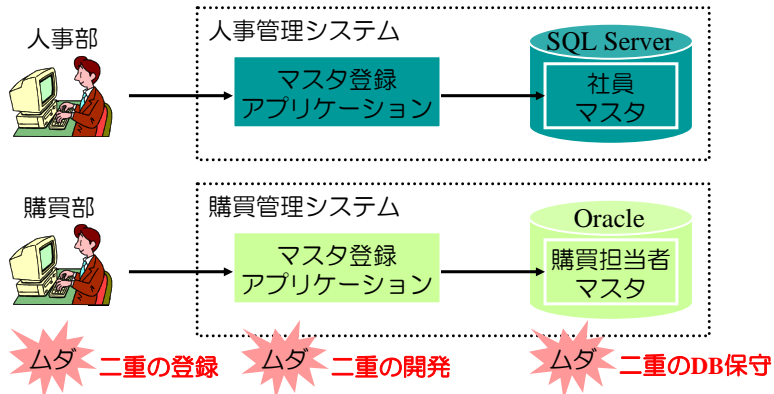
データモデリングの必要性

情報システムの役割

- 情報システムとは？
 - データから情報へ
 - 情報への変換(コーディング)
 - データのインプット、情報のアウトプット

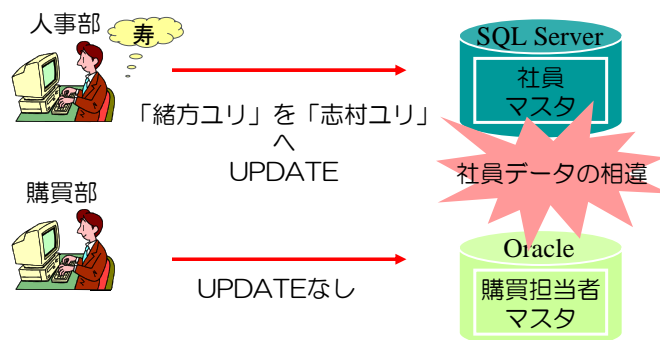
個別最適とその問題点(1)

- 従来は、**業務毎の個別最適**を追求
『極端な例ですが、何か心当たりはないですか？』

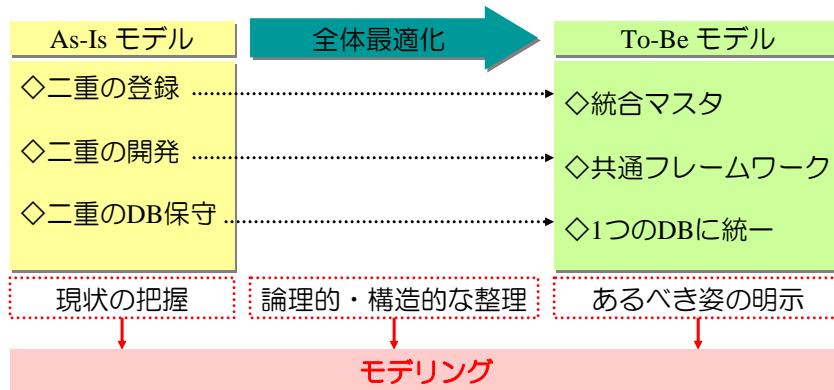


個別最適とその問題点(2)

- 個別最適化による開発では
アウトプットされる情報の精度が低い



- 多かれ、少なかれ、ユーザ企業は個別最適の問題を抱えている
だからこそ、EA(業務・システム最適化計画)が注目されている



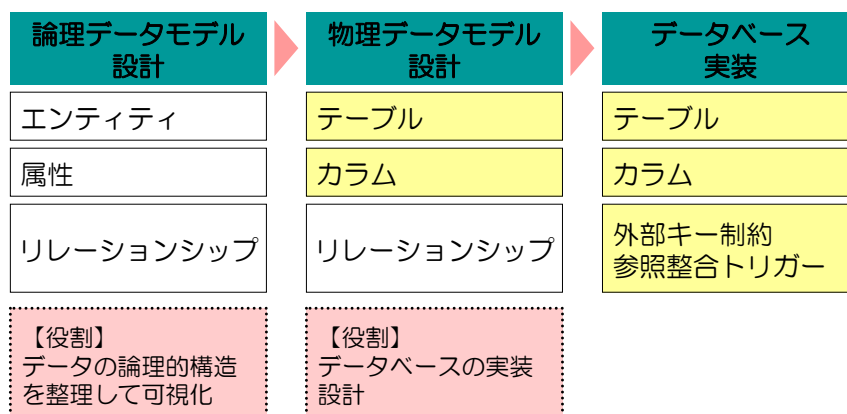
- ユーザ企業の生の声
 - 東京三菱銀行CIOが語る「CIOの真の仕事」
(<http://www.atmarkit.co.jp/news/200305/29/btm.html>)

『「長期経営戦略に則したエンタープライズ・アーキテクチャ(EA)の構築」というプロセスは、組織全体のITシステムを「共通言語」「統一手法」でモデル化することを指す。各部門がバラバラの言語で、別々の手法で、しかも千差万別のアーキテクチャを採用するという事態は決して珍しいものではない。しかし、長期的な視野でITシステムを経営戦略の中核に据えようと考えた場合、すべてのシステムが共通の基盤の上に構築されるのは、拡張性の面でも至極当然のことだろう。』



- モデルとは
 - 分析対象の仕組みや構造を理解するために、分析対象を正確、単純かつ簡潔に表現したもの
 - モデルを読み取ることによって、ビジュアルに適切にビジネスを理解することができる
- EA(Enterprise Architecture)の父John Zachman氏の言葉
 - モデルを作り
 - モデルを蓄え
 - モデルを管理し、拡張し
 - モデルを変えるのだ
 - なぜなら、企業は変化する
 - 変化する企業をその変化に合わせて管理することができるのはモデル

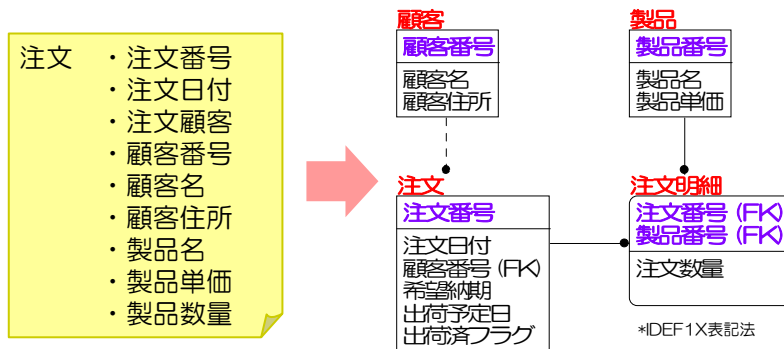
- システム開発でのモデルの使用

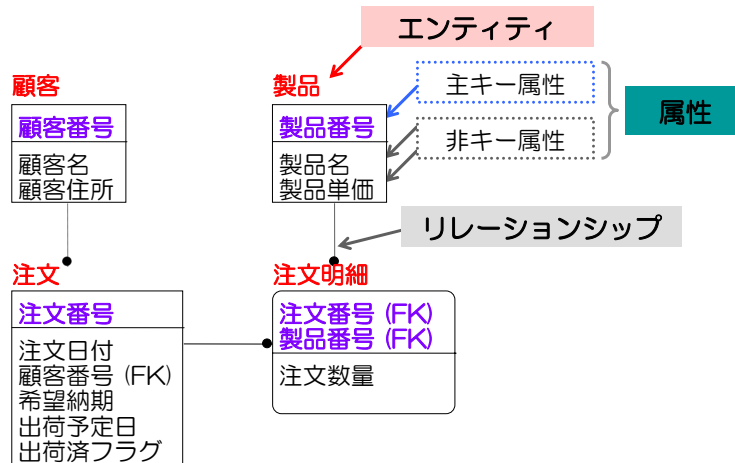


データモデリングの基本

データモデルから対象システムのルールを読み解く

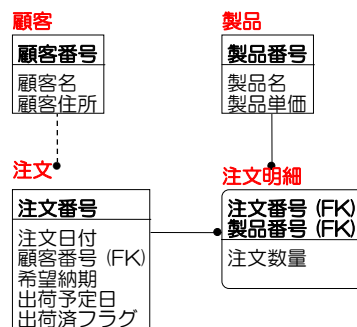
- 「注文」「注文明細」「顧客」「製品」が管理対象であることを鳥瞰図的に把握できる
- 『1回の「注文」では複数の「製品」が注文でき、その製品毎に「注文明細」をつくる』など、ルールが明確になる



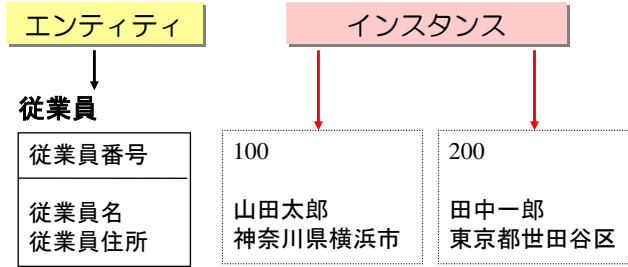


データ(情報)として管理すべき対象

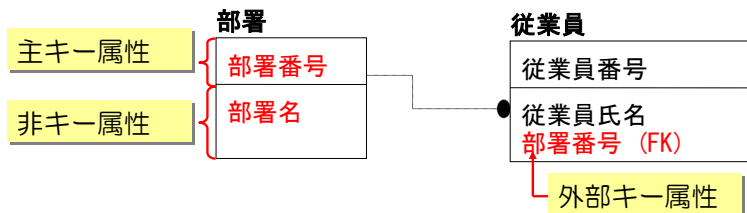
- 人(例えば、「顧客」)
- 物(例えば、「製品」)
- 取引(例えば、「注文」「注文明細」)
- 組織
- 場所
- 分類概念
- など



- エンティティを「器」とすると、インスタンスはその「中身」1つ1つのこと



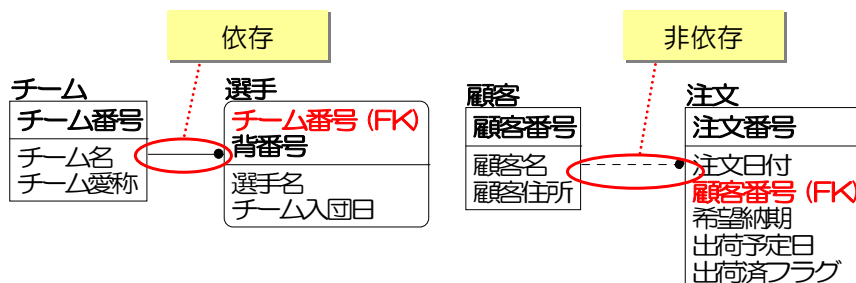
- エンティティの構成要素(データ項目)
- 主キー属性**: 1つ以上の属性の組み合わせであり、エンティティ内のインスタンスを一意に識別する属性
- 非キー属性**: 主キー以外の属性
- 外部キー属性**: 親エンティティの主キーであり、リレーションシップを介して親エンティティから子エンティティに移行した属性



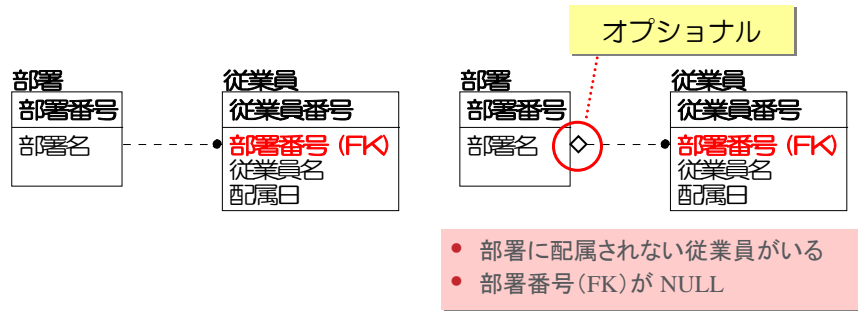
- エンティティ間の関係(ビジネスルール)を表す
- リレーションシップの種類

依存 リレーションシップ	
非依存 (必須) リレーションシップ	
非依存 (オプション) リレーションシップ	
多対多 リレーションシップ	

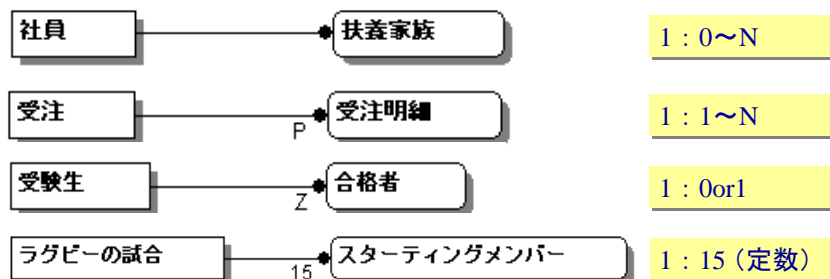
- 子エンティティが親エンティティに依存するのか？
- 子エンティティの外部キー(FK)が「主キー」なのか、それとも「非キー」なのか？

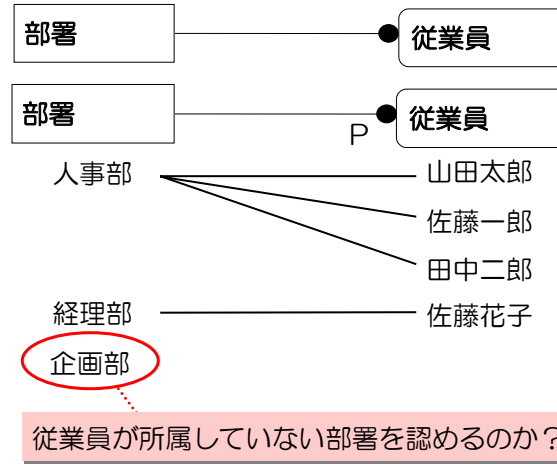


- 親インスタンスと関連付かない子インスタンスを認めるのか？
- 外部キー(FK)に NULL を認めるのか？

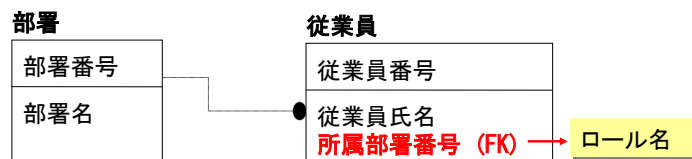


- エンティティ間の関係をより明確にする

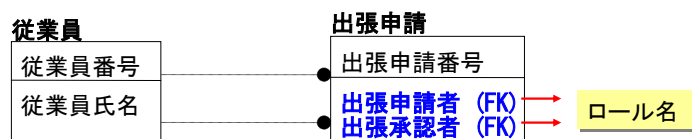




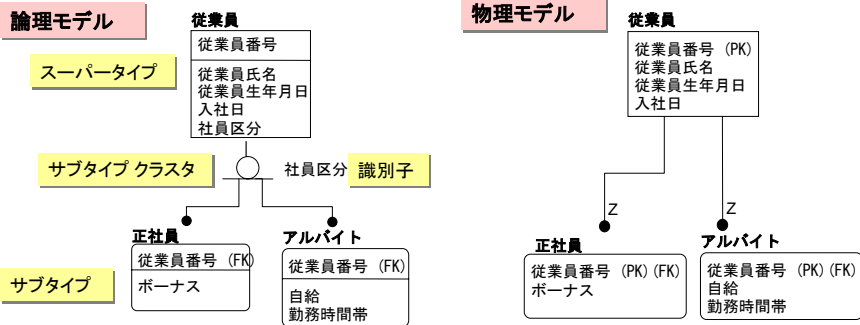
- 子エンティティでの役割を表す外部キーの別名



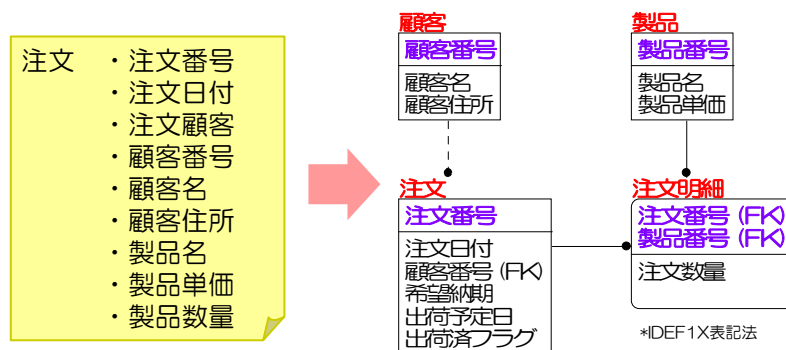
- エンティティ間に複数のリレーションシップ
 - 外部キー名称の重複をなくすために



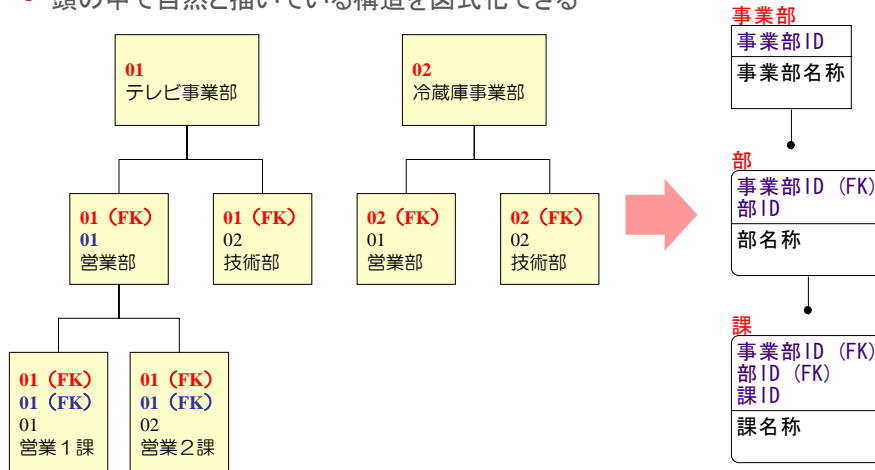
- エンティティをタイプ別に分類する
- 共通属性と固有属性を明確にする
- 確定型と未確定型がある
- 物理モデルではDB実装に向けて解決する



- 管理対象を鳥瞰図的に把握できる (管理対象は、「注文」「注文明細」「顧客」「製品」である)
- ルールが明確になる (『1回の「注文」では複数の「製品」が注文でき、その製品毎に「注文明細」をつくる』など)



- 管理対象の階層構造を整理できる
(課は部に属し部は事業部に属する)
- 頭の中で自然と描いている構造を図式化できる



本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

25

正規化とは

- 正規化とは、One Fact in One Place(一つの事実は一つの場所に)を実現するための手法
- 属性が適切なエンティティに属しているか?
- データに冗長性がないか?
- 第1正規化: 繰り返しの排除
- 第2正規化: 主キー(複合キー)の一部に依存する属性の排除
- 第3正規化: 非キーに依存する属性の排除

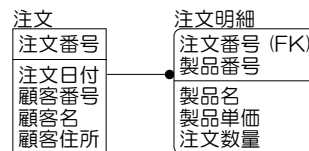
本文書の一部または全部の転載を禁止します。本文書の著作権は、著作者に帰属します。

26

第1正規化

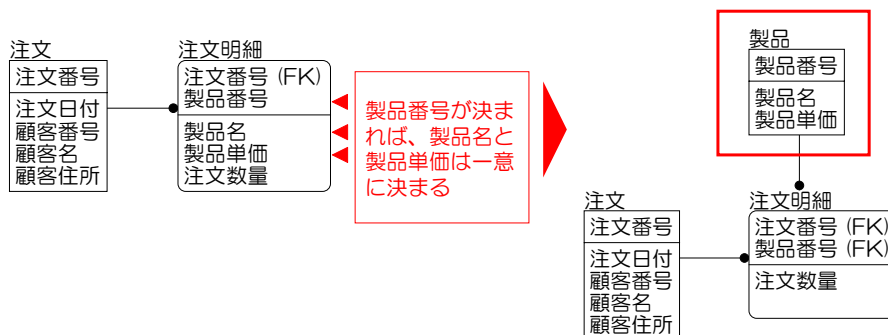
- 繰り返し属性の排除

注文番号	注文日付	顧客番号	顧客名	顧客住所	製品番号	製品名	製品単価	注文数量
1001	2004/3/1	001	鈴木	東京都	100	ER/Studio	500,000	5
					200	Rapid SQL	135,000	5
					300	DBAirtisan	240,000	1
1002	2004/3/1	002	佐藤	神奈川県	100	ER/Studio	500,000	10
1003	2004/3/2	003	高橋	千葉県	200	Rapid SQL	135,000	3
					300	DBAirtisan	240,000	3
1004	2004/3/3	001	鈴木	東京都	300	DBAirtisan	240,000	4

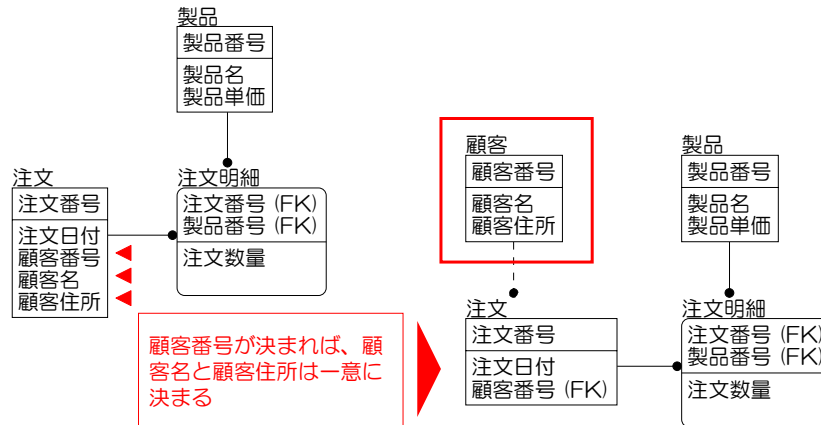


第2正規化

- 複数の属性で構成される主キーの一部に依存する属性の排除



- 非キー属性に依存する属性の排除



- 主キーに依存していない属性を排除する
(主キーに依存する属性のみ存在する状態を作る)
- 業務に関係するエンティティを切り出していく
(特定の対象を管理するためのエンティティが増えていく)
- 管理単位を決める
(主キーを確定する)
- 主キーを確定することによりエンティティ間の関連を把握
(リレーションシップを作成できる)

非正規化の問題点1

- 重複更新が必要になる(更新不整合)
- 適切なタイミングで適切なデータ管理ができなくなる(挿入・削除不整合)

注文番号	注文日付	顧客番号	製品番号	製品名	製品単価	注文数量
1111	2005/10/19	222	100	ER/Studio	500,000	5
1111	2005/10/19	222	300	Repository	200,000	5
1112	2005/10/19	435	100	ER/Studio	500,000	10

更新不整合	「製品番号100はER/Studio、500,000である」との関係が重複して登録され、製品名、製品単価を変更する際に重複更新が必要になる。
挿入不整合	注文を登録する前に「製品番号100はER/Studio、500,000である」との関係に登録しておくことができない。
削除不整合	注文を削除してしまうと「製品番号100はER/Studio、500,000である」との関係も削除してしまう。

非正規化の問題点2

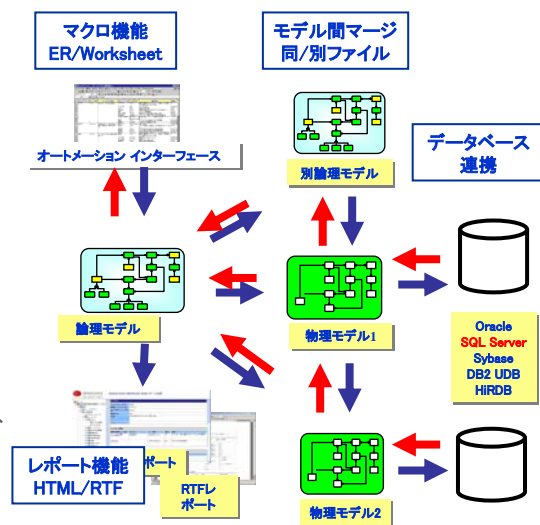
- マスタからデータを参照できないので複雑な画面を作る必要がある
→ たくさんのコードを書かなくてはいけない。潜在バグの可能性。
- 繰り返し登録を排除するためのコーディングが必要がある
→ DISTINCTやGROUP BYの多用によりDB性能が低下。
- さまざまな可能性を秘めたDBとなる
→ 信頼性の低いデータのインプット、アウトプット

データモデル駆動型開発

RDB設計支援ツール

● ER/Studio

- ERダイアグラムを簡単に
 - 論理設計/物理設計/多次元
- スキーマを自動生成
- リバースエンジニアリング
- マージ機能
- 多次元モデリング
- 非正規化マッピング機能
- 自動レポート生成
 - HTML、Word、Excelに
- マクロ機能
 - オートメーションインターフェース
 - ER/Worksheet



- 属性の設定情報(データ型、定義、デフォルト ...)を一元的に管理
- 統一性とメンテナンス容易性を確保する

郵便番号ドメイン

従業員

従業員 id
従業員 氏名
従業員 住所 郵便番号
従業員 住所 都道府県コード

得意先

得意先 id
得意先 名称
得意先 住所 郵便番号
得意先 住所 都道府県コード

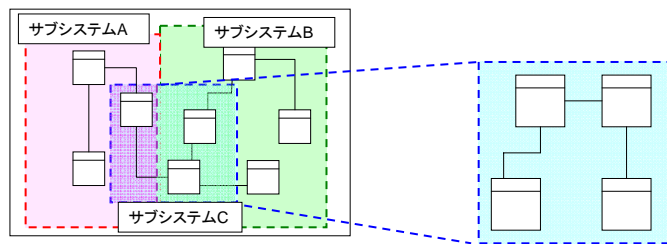
データの詳細を定義する

- データが持つ意味や特性の理解を深め、データの利用方法を把握するため
- データの品質を確保するため

定義項目	ER/Studio
データが持つ意味をテキストで明記する	定義
補足のテキスト説明をつける	ノート
明示的な値の指定が無い時に(初期)設定する値を定義する	デフォルト
許容される値(評価式)を定義する	ルール
許容される値(範囲、リスト)を定義する	参照値
モデラーが自由に定義内容を追加する(例 分析元資料、サンプルデータ)	アタッチメント

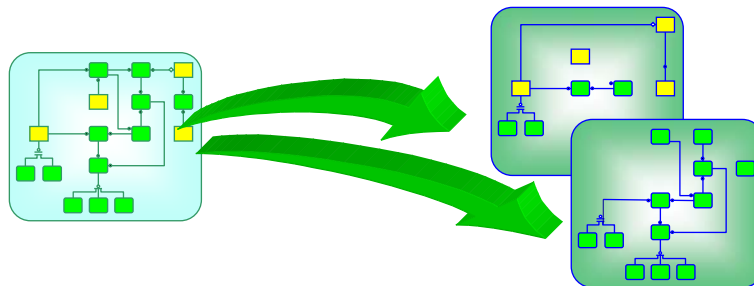
サブシステムの範囲を定義する

- システム化対象全体から現在のシステム化対象システムを切り出すことができる
- 既存のシステムとの関わりを確認できる
- 他システムへの影響範囲を把握できる
- 自分の必要な範囲のデータ構造のみを抽出できる

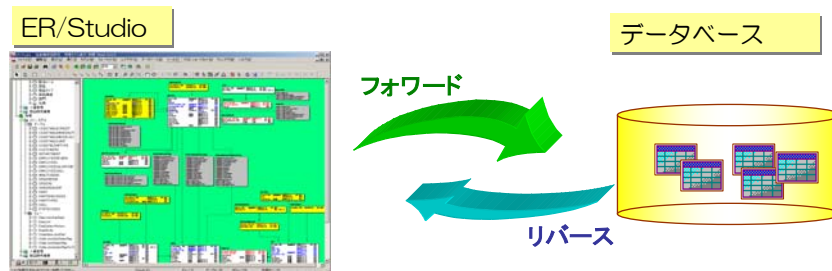


物理モデル生成

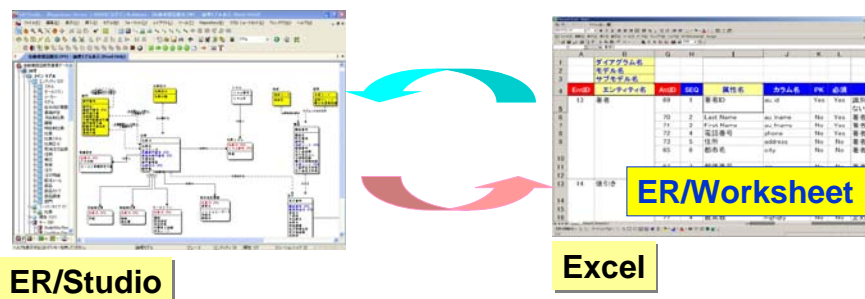
- 多対多リレーションシップ、汎化階層の解決
- データベースへの実装を考慮
 - データ型、容量、パフォーマンスなど
- DBで使用するオブジェクト名の定義
- データベースの分散を考慮したモデル分割
- 必要な場合には非正規化も



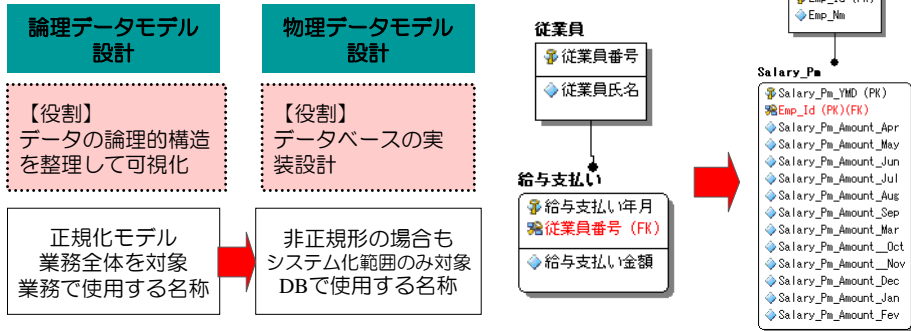
- モデルからDDLを自動生成
- データベースからモデルを自動生成
- モデルとデータベースの差分DDLを自動生成



- モデリングツールとExcelとの連携を実現するツール
- ER/Studioのマクロ機能を活用したツール
マクロ機能を活用すると、定型作業の自動化・効率化が図れる



- 論理モデルの正規化されたモデルによって影響範囲やデータの意味の把握、間違ったデータをinsertしない仕組みができる
- 物理モデルによってDBの構造を把握できる
- 論理モデルと物理モデルによって業務とデータベース構造のマッピング、影響範囲の分析ができる



ご清聴ありがとうございました！

J-SYS SOFTWARE high productivity around Data
日揮情報ソフトウェア株式会社
<http://www.jsys-products.com/>
 営業本部
 〒222-0033
 横浜市港北区新横浜3-6-12 日総第12ビル
 TEL : 045-474-7852
 FAX : 045-474-7882
 E-Mail : request@jsys-products.com

モデリング技術に関する
 情報発信中！『岩田研究所』
<http://www.jsys-products.com/iwaken/>
 『岩田研究所ブログ』
<http://tech.jsys-soft.jp/iwaken.php>

製品のトライアルは
 ダウンロードセンターへ！
<http://download.jsys-soft.jp/>