

あなたにもわかるデータベース設計

えムナウ (児玉宏之)



<http://mnow.jp/>

<http://mnow.wankuma.com/>

<http://blogs.wankuma.com/mnow/>

<http://www.ailight.jp/blog/mnow/>

.NET UX Lab

.Net ユーザーエクスペリエンス研究所

えムナウのC#

プログラミングのページ



わんくま同盟 東京勉強会 #31 – DATABASE DAY

自己紹介

- 福井県越前市(武生市)出身、東京都杉並区在住。
- ソフトウェア業務歴30年・フリー技術者歴20年以上、でも「**永遠の28歳**」。
- 4Bitワンチップからワークステーションまで開発言語も多種多様、最近はSQLサーバー、Web、Windows Forms や WPFの Visual C# ソフト開発が多い。
- Microsoft MVP for Development Tools Visual C# を5年連続で受賞。
- INETAJ 理事、INETAJ スピーカビュロー登録。
- わんくま同盟他でスピーカ経験多数。



アジェンダ

- はじめに
- 業務モデリング
- ドメインモデリング
- エンティティモデルの設計方法
- まとめ



はじめに

- 顧客と開発者のギャップを埋める

- 顧客のやりたい事と開発者の作るものには大きな壁があります。
- この壁を越えるためにやりたいことをモデル化して開発者にわかりやすくする必要があります。
- 家を建てる場合でも平面図やパースを使って顧客にわかりやすく確認してもらいながら設計を進めていきます。
- 顧客もこんなものができるというイメージが湧いて安心できます。



はじめに

- 業務モデリングとドメインモデル

- 顧客のやりたいことをモデル化していくのが業務モデリング（アプリケーションモデル）

- 業務の目標や問題点を明確化し業務の全体像と作成するシステムとの境界と利用範囲を明らかにします。

- 顧客の現実世界をモデル化していくのがドメインモデリング（ドメインモデル）

- 業務における事象をソフトウェアが扱えるように抽象化します。



業務モデリング

• 業務モデル

– 業務ビジョンモデル

- 顧客の目的や導入利益を明確化し開発のゴールを記述しビジョンを共有します。

– 業務プロセスモデル

- どんな人が何を使ってどんなタイミングでどんなことをやるというのを記述します。

– 業務ユースケースモデル

- 顧客のやりたいことを達成するための手順を描いた物語です。

業務モデリング

• 業務ビジョンモデル

– 業務ポジション

- 対象者、対象者のやってほしいこと、使うメリット、代わりの方法、優位となる相違点を記述します。

– コンテキスト図

- 従来と導入後の対比を行います。

– 業務ゴール

- 財務、顧客、業務プロセス、学習と成長の分野での問題と目標を整理します。

業務モデリング

• えムナウ商会-業務ビジョンモデル

– 業務ポジション

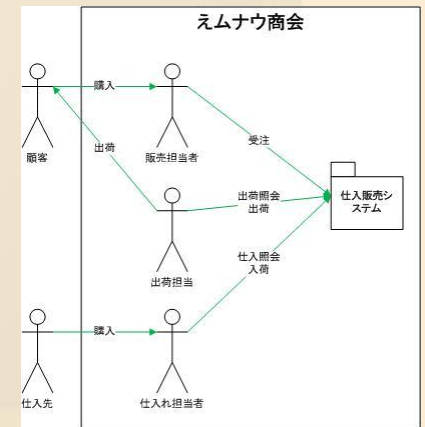
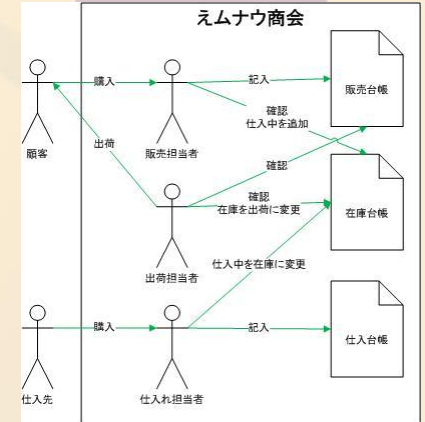
- お客さんが待たされないで購入できる、会社内の遅滞の軽減で仕入れ販売をスムーズに、今までの伝票による社内依頼を代用し、購入から出荷までを迅速に処理する。

– コンテキスト図

- 右図

– 業務ゴール

- 財務:社内工数の削減、顧客:迅速な購入、業務プロセス:伝票確認漏れ、転記ミスの軽減、学習と成長:購入品目の蓄積調査、先行仕入の情報収集



業務モデリング

- 業務プロセスモデル

- 業務アクター一覧

- 業務にかかわる人を一覧にして役割を確認します。

- 業務プロセス図

- アクター、リソース、イベント、ドキュメント、サービスを記述して関係を整理します。

- 業務フロー図

- 業務プロセスを達成するための業務の流れを記述します。

業務モデリング

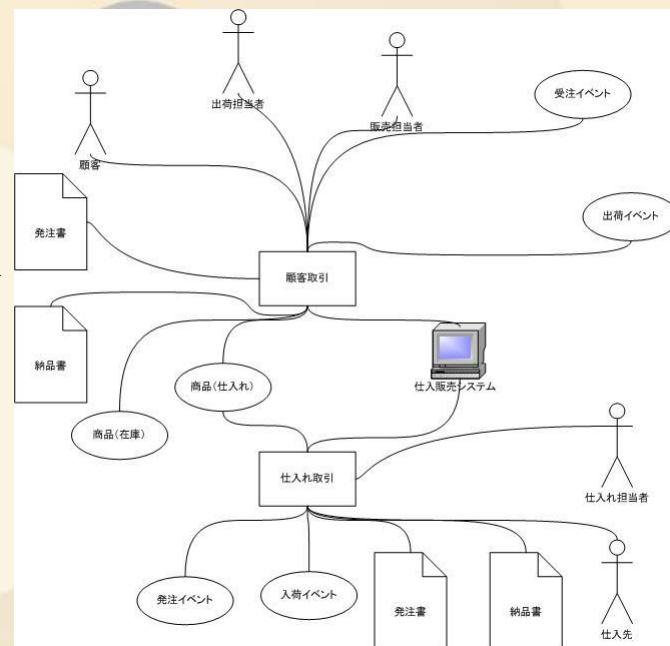
• えムナウ商会-業務プロセスモデル

– 業務アクター一覧

- 顧客(顧客、仕入先)、作業者(販売担当者、出荷担当者、仕入担当者)、サービス(仕入れ販売システム)、管理者、保守者

– 業務プロセス図

- 業務プロセスは顧客取引と仕入れ取引に分けて、各々の関係するアクター、リソースイベント、ドキュメント、サービスを、記述します。
- たとえば右図。
- いろんな書き方がありません。

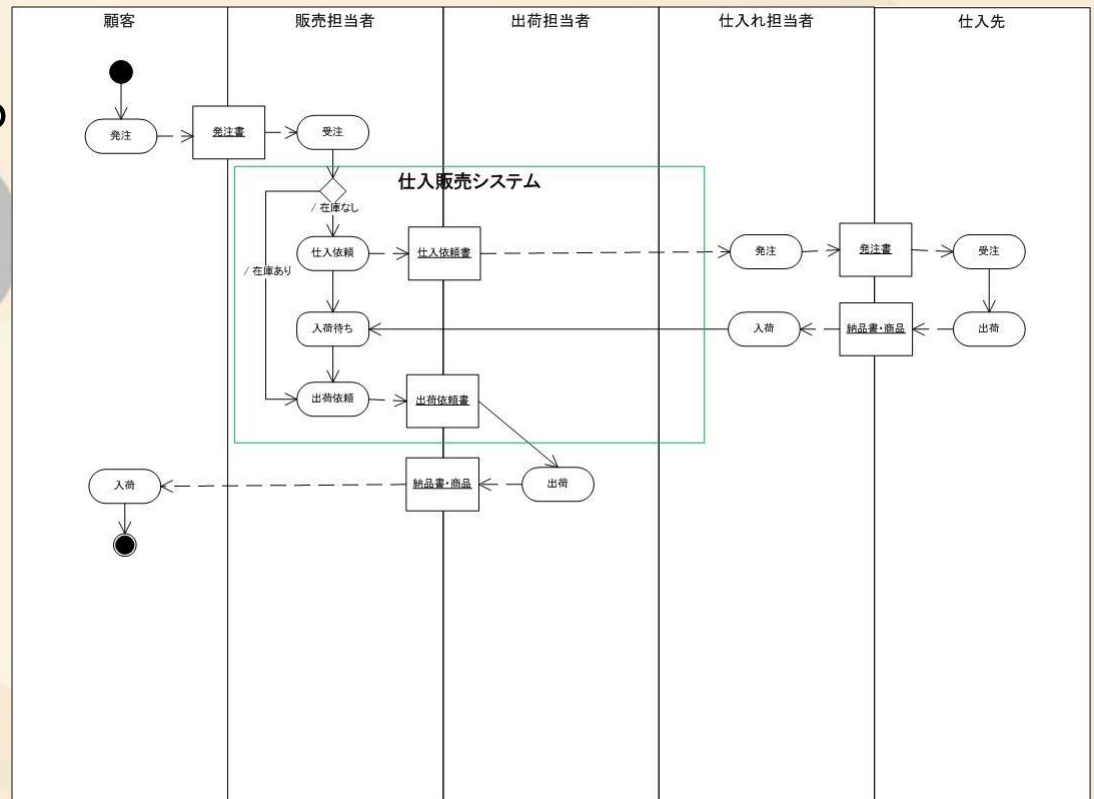
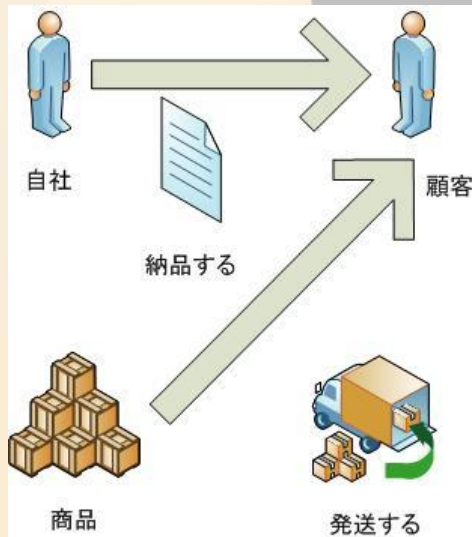


業務モデリング

• えムナウ商会-業務プロセスモデル

– 業務フロー図

- 業務の流れを、
わかるようにする



業務モデリング

• 業務ユースケースモデル

- 業務ユースケースは開発者が作成するシステムユースケース（開発者はこちらを一般にユースケースと言います）の基礎になり、顧客と開発者の理解の橋渡しになる重要なものです。
- ユースケース一覧表
 - ユースケースの一覧です。
- ユースケース図
 - ユースケースを図にしたものです。
- ユースケース
 - ユースケース名、アクター（かかわる人やサービス）、トリガー（起点）、ガード（事前条件）、目標（事後条件）、脚本、代わりの脚本、失敗の脚本について文章で共通認識を図ります。

業務モデリング

- えムナウ商会-業務ユースケースモデル
– ユースケース(受注のみ)

ユースケース名	受注
アクター(かかわる人やサービス)	販売担当者、仕入販売システム
トリガー(起点)	顧客よりの受注
ガード(事前条件)	なし
目標(事後条件)	在庫不足があれば仕入指示して在庫不足を解消し出荷指示をする
脚本	1.在庫照会をする 2.在庫があれば出荷指示する
代わりの脚本	2a.在庫が不足があれば仕入指示する 3.出荷指示する
失敗の脚本	



ドメインモデリング

- ドメインモデル

- サービスモデル

- 現実の人の代わりに目標を達成してくれるもので、システムであったり通信であったりします。

- ルールモデル

- 現実の規則を抽象化します、制約・演算・検索・探索・推論・監視に分類することが多いです。

- エンティティモデル

- 現実の人・物・事を抽象化します、アクター、リソースイベントが一般的に候補となります。

ドメインモデリング

- サービスモデル
 - WebサービスやREST、他システムとの連携など、本システム以外で利用する手段です。
 - 郵便や電話、宅配などの物理的なサービスも含める場合があります。
- ルールモデル
 - 消費税の計算や仕入れ販売間の利益率、売れ筋商品の最低在庫確保など、決まり事をモデル化したものです。

ドメインモデリング

- エンティティモデル

- ドメインモデルの主軸になるのがエンティティモデル、本セッションの主役の登場です。
- 現実の人・物・事を抽象化します、アクター、リソース、イベントが一般的に候補となります。
- 顧客が業務について語った言葉や文章を単語に分割するとエンティティモデルやその構成要素になることがよくあります。



ドメインモデリング

- えムナウ商会-サービスモデル
 - 今回は特にありません、強いてあげるとすればデータベースとクライアントマシンが通信するのでそれを盛り込むかどうかでしょう。
- えムナウ商会-ルールモデル
 - 現段階ではルールは見えてきません、商品の販売価格は決まっています、仕入とは連動しませんし、請求はシステムに入っていないので消費税も関係ありません、もっと細かく設計すれば出てくるかもしれません。

ドメインモデリング

- えムナウ商会-エンティティモデル
 - アクター、リソース、イベント
 - アクター: 顧客(顧客、仕入先)、作業者(販売担当者、出荷担当者、仕入担当者)
 - リソース: 商品
 - イベント: 受注、出荷、発注、入荷
 - 顧客が業務について語った言葉や文章
 - 台帳類: 販売、在庫、仕入
 - 在庫、仕入指示、出荷指示など

エンティティモデルの設計方法

- えムナウ商會を例にエンティティモデル(概念モデル)を設計する
 - 現状やっていることをヒアリング
 - 顧客から注文書が郵送してくる
 - 販売担当が注文書と在庫台帳を見比べて足りてれば出荷指示を出荷担当に行い、販売台帳に書く
 - 足りなきゃ仕入担当に不足分仕入指示をして出荷指示を出荷担当に行い、販売台帳に書く
 - 仕入担当は仕入指示を元に仕入先に発注して仕入台帳に追加し入荷したら在庫台帳の在庫を増加させる
 - 出荷担当は販売台帳と在庫台帳を見比べて足りるなら顧客に納品書と一緒に出荷して在庫台帳の在庫を減少させる



エンティティモデルの設計方法

• 単語の羅列

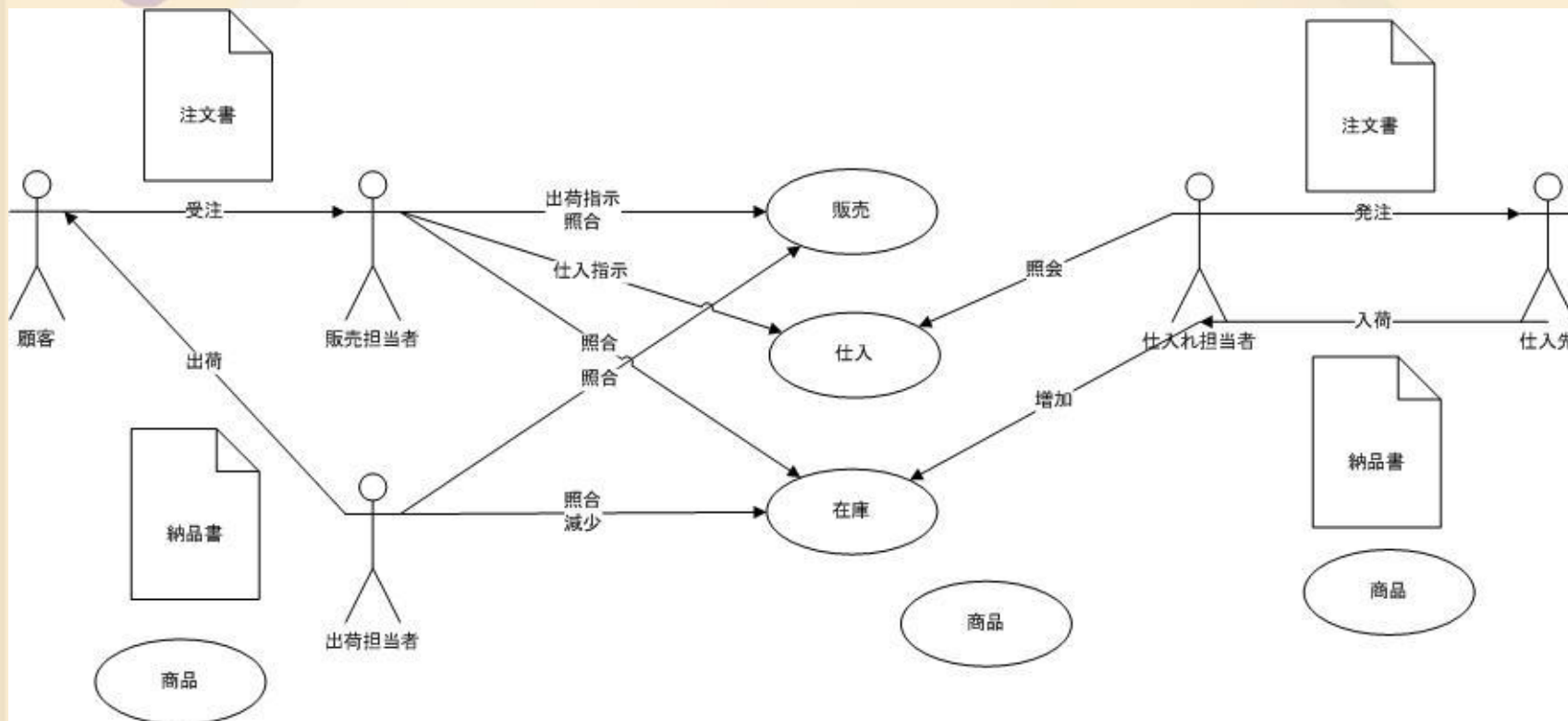
顧客	注文書	郵送	販売担当	在庫台帳
出荷指示	出荷担当	販売台帳	仕入担当	仕入指示
仕入先	発注	入荷	仕入台帳	納品書

• 単語の分類と不足の確認

アクター	顧客	販売担当	出荷担当	仕入担当	仕入先	
リソース	在庫 (台帳)	販売 (台帳)	仕入 (台帳)	商品		
ドキュメント	注文書	納品書	注文書 (仕入)	納品書 (仕入)		
イベント	発注	入荷	出荷指示	仕入指示	受注	出荷

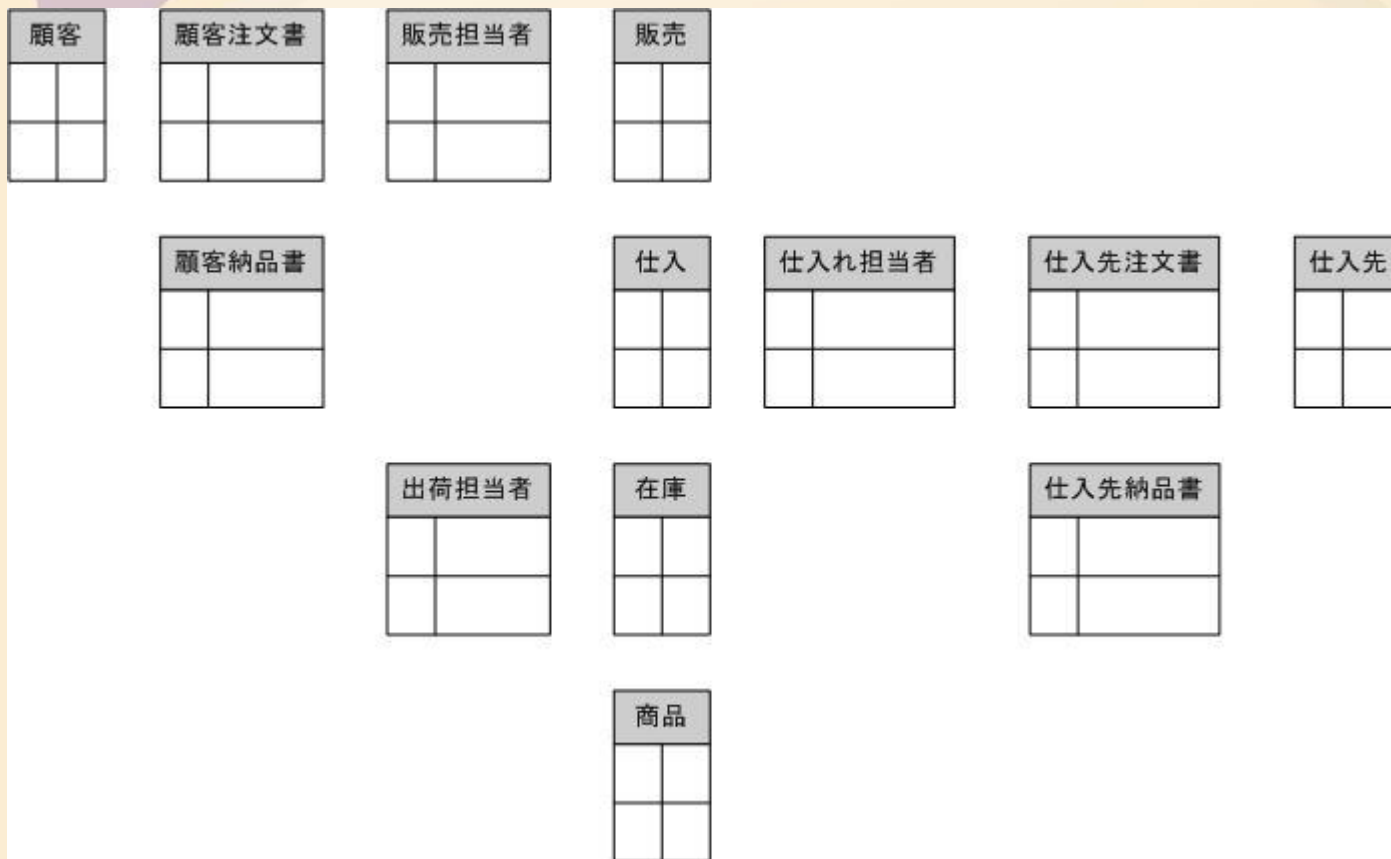
エンティティモデルの設計方法

- 図にしてみる



エンティティモデルの設計方法

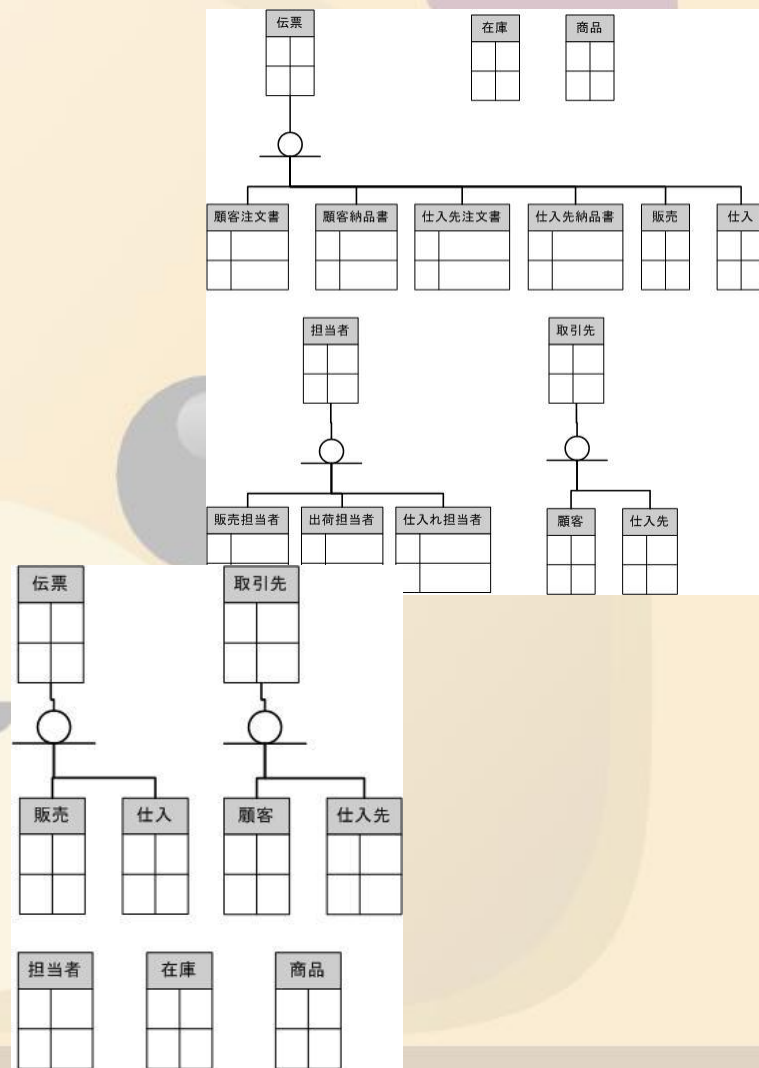
- エンティティ関係図を書いてみる



エンティティモデルの設計方法

• カテゴリライズ

- 注文書と納品書は、同じ形式、販売仕入は、転記しただけ
- 顧客注文書納品書は、販売で代用できる
- 仕入先注文書納品書は、仕入で代用できる
- 担当者はまとめて考えられる
- 顧客と仕入先も同一要素が多い



エンティティモデルの設計方法

- 項目を作成する
 - 各種伝票や台帳を入手してヒアリングする

担当者	伝票	商品	在庫	取引先
社員番号	伝票番号	型名	型名	名称
氏名	発行日	単位	数量	担当
	取引先	売価		郵便番号
	担当者			住所1
	型名1			住所2
	数量1			電話番号
	単価1			FAX
	:			

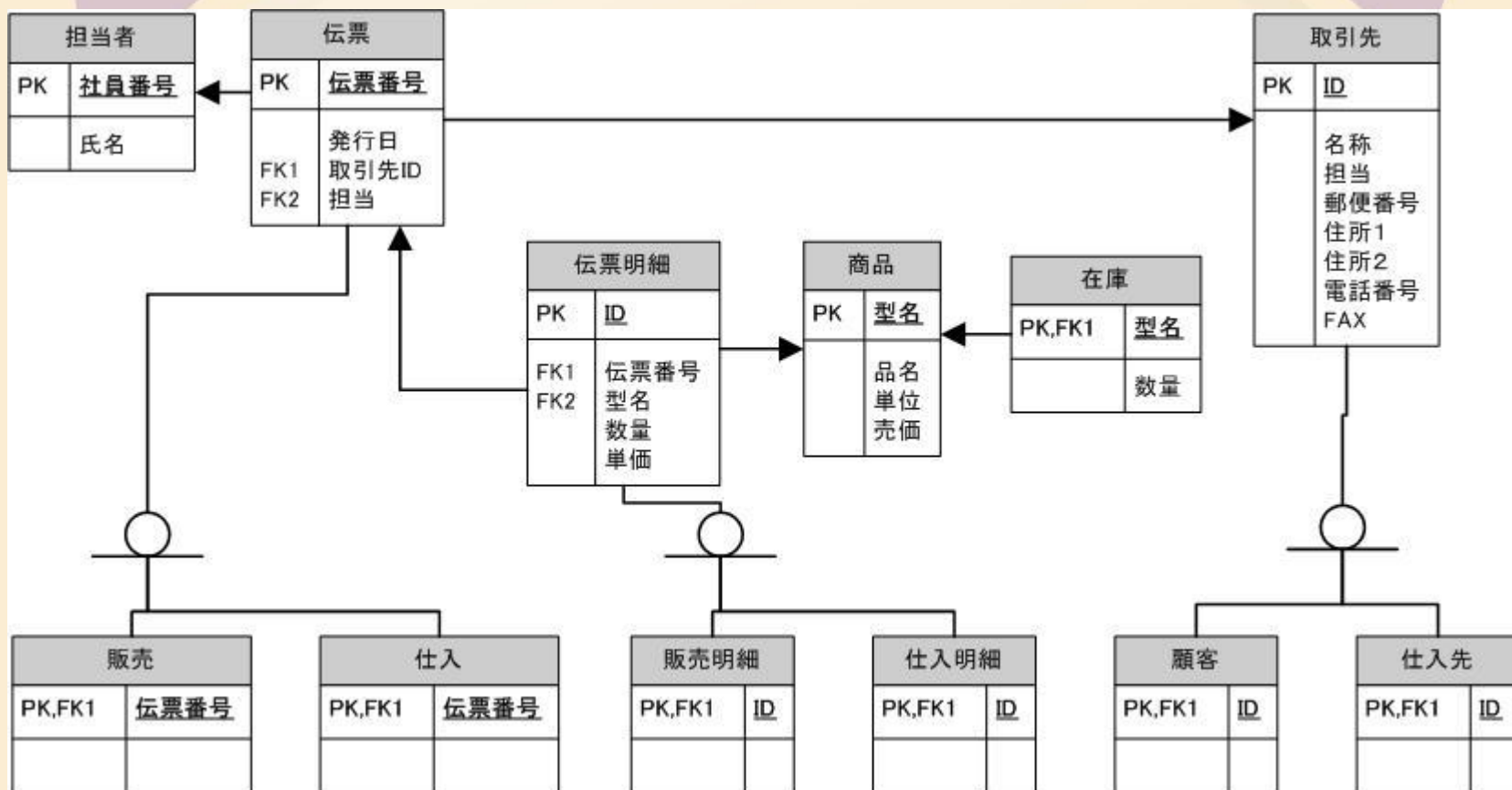
エンティティモデルの設計方法

- キーを考える
 - キーになりそうなものがない場合はID
- 正規形を作成する
 - 第1から第3まで
 - 今日の初音さんの話や私の東京27の「Linq to Entities をつかってみよう」を参考に



エンティティモデルの設計方法

• エンティティモデル(概念モデル)



エンティティモデルの設計方法

● 論理モデルを作成

- 論理モデルはユーザーインターフェースを固めながら作成していきます。
- ボトムアップな作成方法をとります。
- 例えばこんな納品書に販売納品書が決まります。

納品書 2008年11月01日 No. 1001

〒 111-1234 東京都新宿区東新宿 1-2-3
東新宿ビル5F 株式会社児玉文具店 永福町支店
光栄産業株式会社 様 電話番号：03-1234-5678 FAX番号：03-1234-5678
下記の通り納品いたしました 振込先：東京三菱UFJ銀行永福町支店 普通1234567

品名	数量	単価	金額（税抜・税込）	備考
123456789 A4 コピー用紙500枚 5ケース入り	5 箱	6000	¥30,000	No. 123456789 宅配便
111111111 ボールペン黒10本入り	10 ケース	500	¥5,000	No. 123456789 宅配便
合計			¥35,000	

20日締め分 税率 5 消費税額等 ¥1,750 税込合計金額 ¥36,750



エンティティモデルの設計方法

- 販売納品書

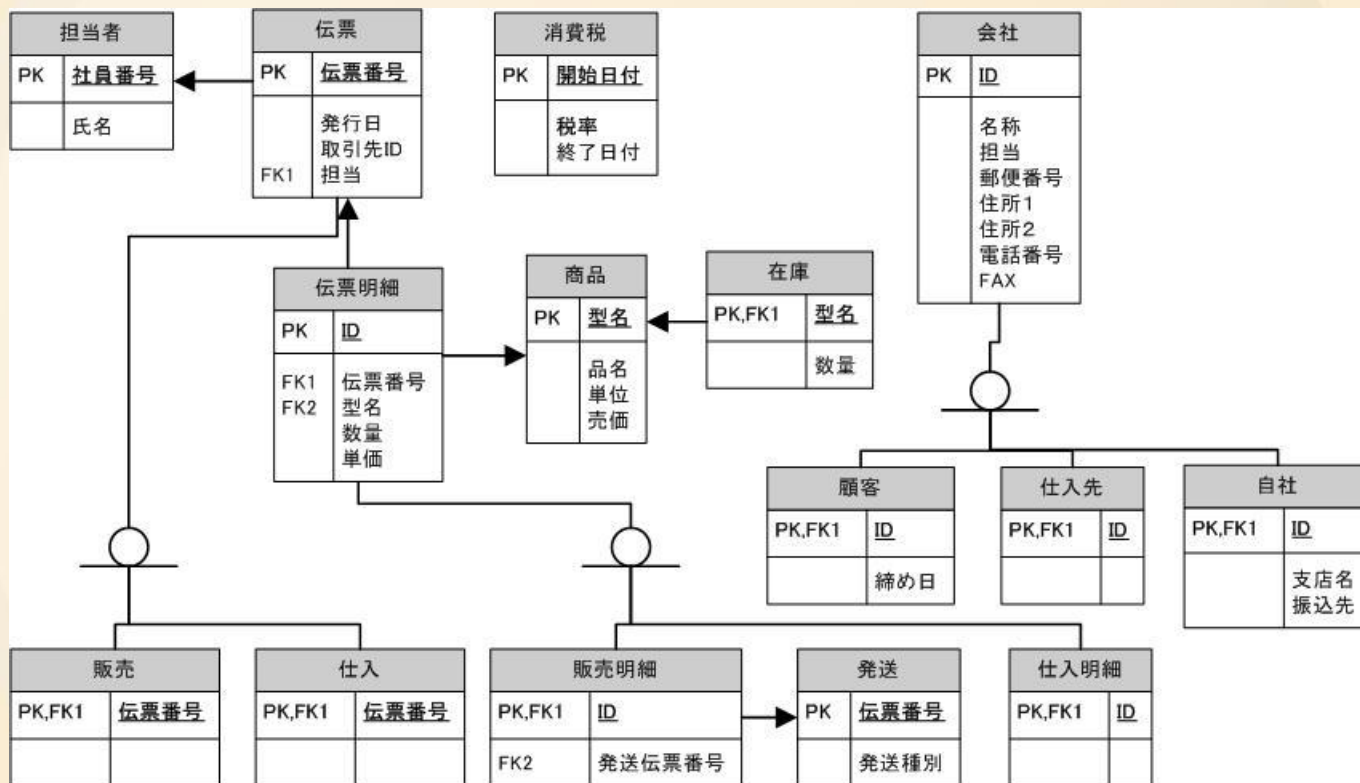
- 消費税計算が必要になります。
- 自社の情報が必要です。
- 発送用の情報が必要です。
- 顧客の締め日が必要です。



エンティティモデルの設計方法

● 論理モデル

－ 概念モデルを修正した論理モデルです。



エンティティモデルの設計方法

- 物理モデルを作成
 - SQLサーバー用にカスタマイズする。
 - 不変のキーと言われているものを信じないでID列をつける。
 - 排他処理は先更新を有効、後更新をエラーにするなら、timestampが便利。
 - データベースの配置やインデックスなど決めていく。

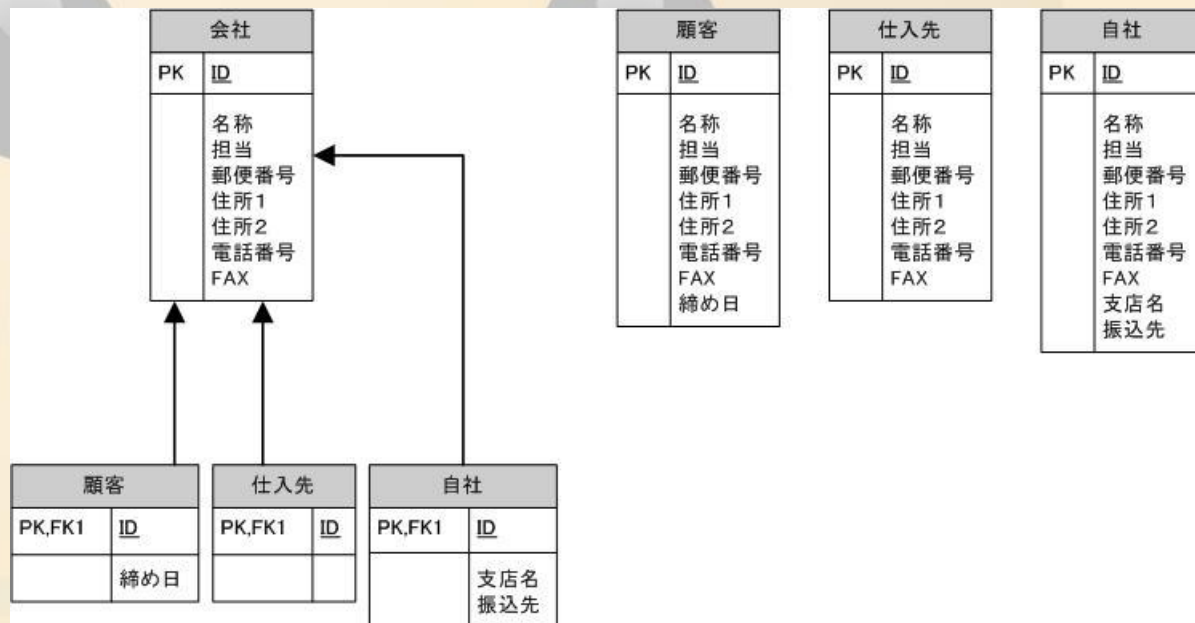


エンティティモデルの設計方法

- マスターとトランザクションに分類する。
 - マスターは物や事柄の名前で名詞が基本。
 - 顧客、仕入先、自社、商品、担当者、消費税率、在庫
 - 削除は行わない、有効期限や削除済みフラグを持つ。
 - 作成者と最終更新者や更新履歴をもつか考慮する。
 - 排他制御を考慮する。
 - トランザクションは「する」がついて動詞化できる。
 - 販売、販売明細、仕入、仕入明細、発送
 - 発生時刻を明確化、必要なら完了時刻も付加する。
 - 修正や削除を行わない、赤伝票を発行する。

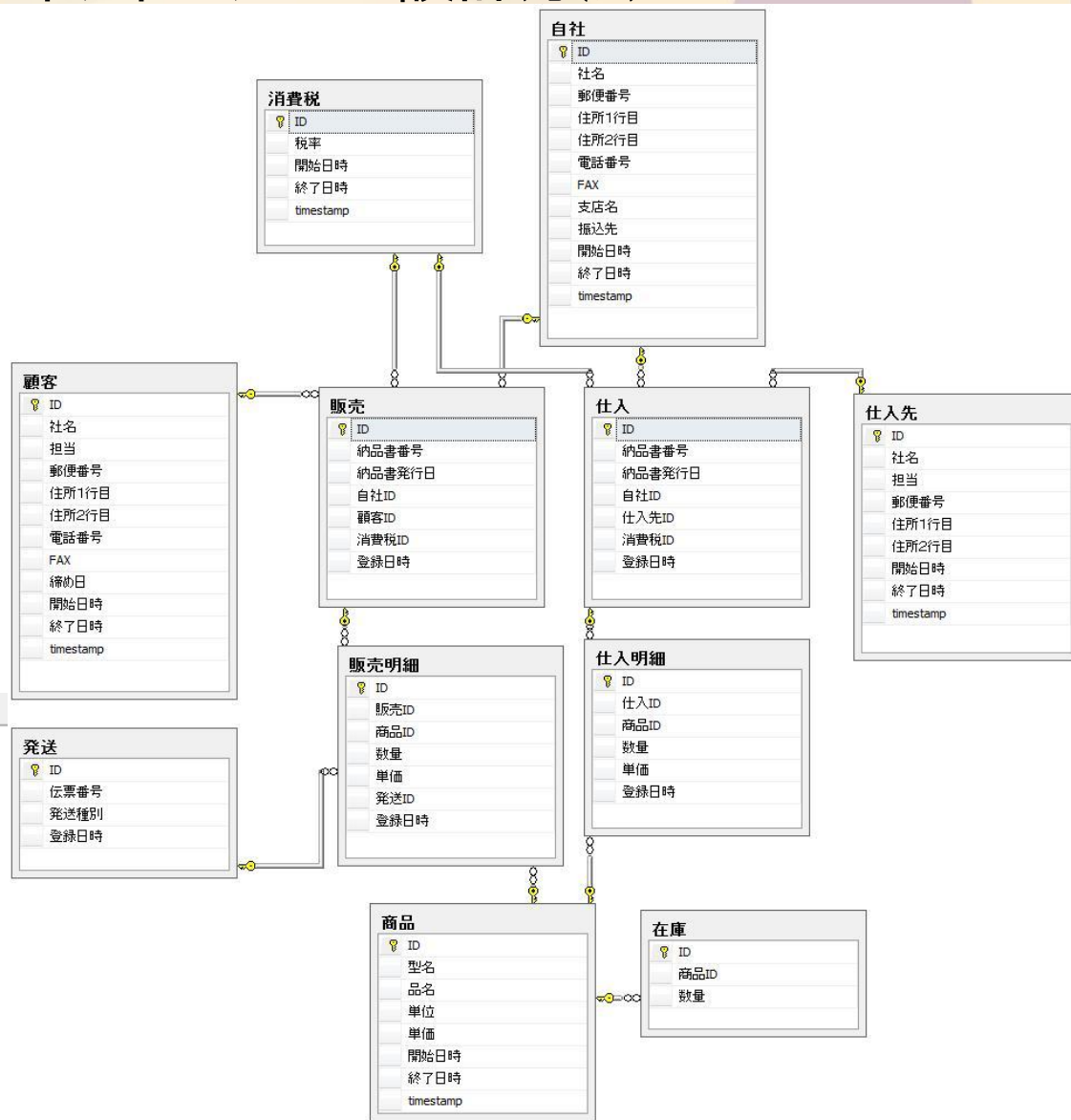
エンティティモデルの設計方法

- カテゴリの親を残すか子に吸収するか決定
 - 親に残すと会社のレコードが増加する
 - 子に吸収すると、顧客でも仕入先でもある場合2重に必要



エンティティモデルの設計方法

• 物理モデル



DELLQUAD.販売仕入...テム - dbo.販売明細

列名	データ型	Nullを許容
ID	int	<input type="checkbox"/>
販売ID	int	<input type="checkbox"/>
商品ID	int	<input type="checkbox"/>
数量	float	<input type="checkbox"/>
単価	money	<input type="checkbox"/>
発送ID	int	<input checked="" type="checkbox"/>
登録日時	datetime	<input type="checkbox"/>



まとめ

- 業務モデリング・ドメインモデリングという上流設計をながめながら概念データモデルの位置づけを確認しました。
- ユーザーインターフェースにあった論理モデルの作成を確認しました。
- 実際のデータベースの特性に合わせて物理データベースの設計を行いました。



まとめ

- 皆さんはこのようなステップを踏んでデータベースを作ってきていますか？
- 今日は実際にグループに分かれてデータベースを作ってもらおうと思います。