

# T字形ER手法の概要と WebObjectsへの展開に向けて

---

WR

[WR@Csus4.net](mailto:WR@Csus4.net)

<http://www.csus4.net/WR/>

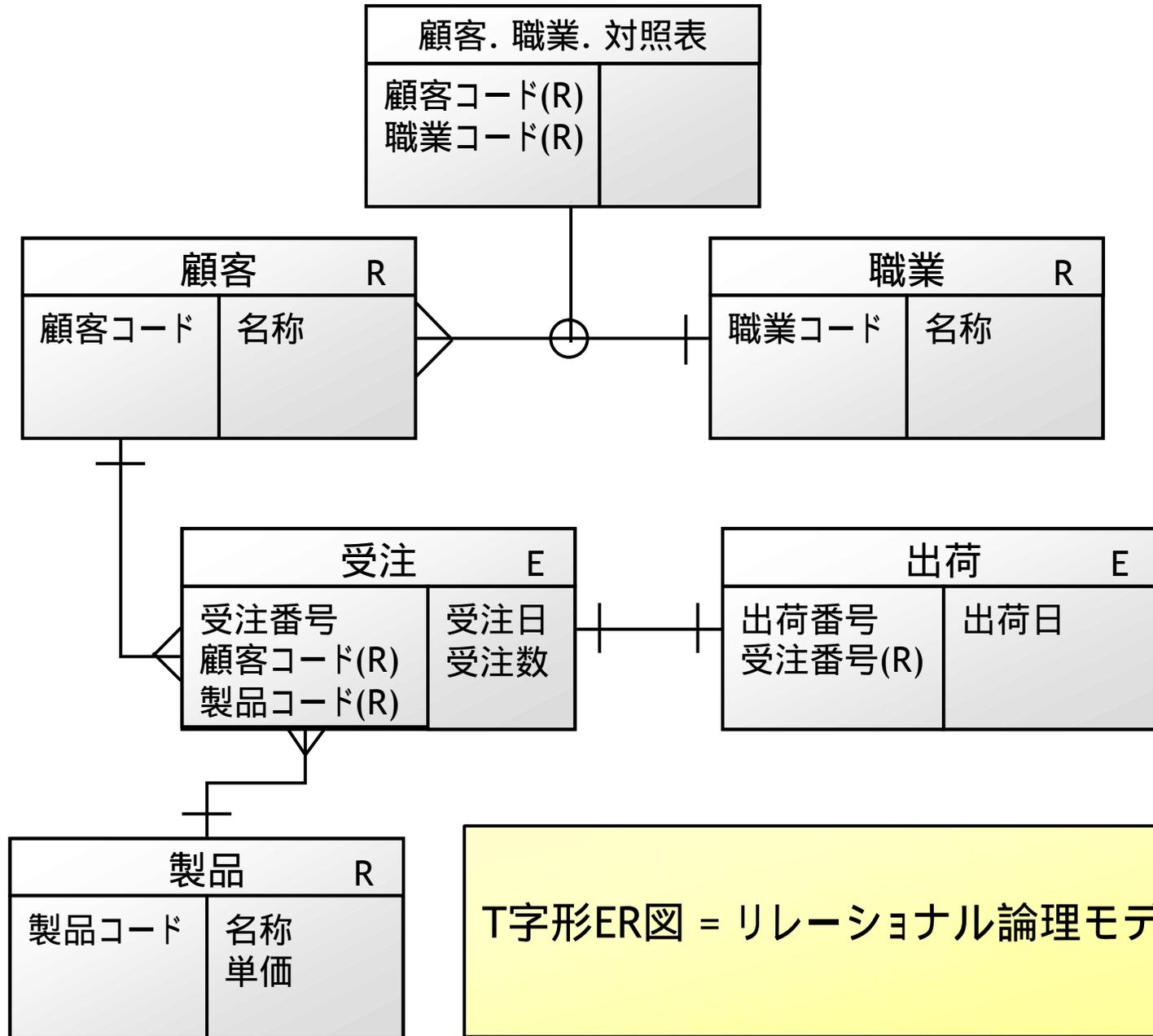
# 目次

- T字形ER手法とは何か?
- T字形ER論理モデルの作成
- EOFモデルへの展開
- 関連情報

# T字形ER手法とは何か?

- T字形ER図の例
- T字形ER手法の狙い
- そもそもER手法とは?

# T字形ER図の例



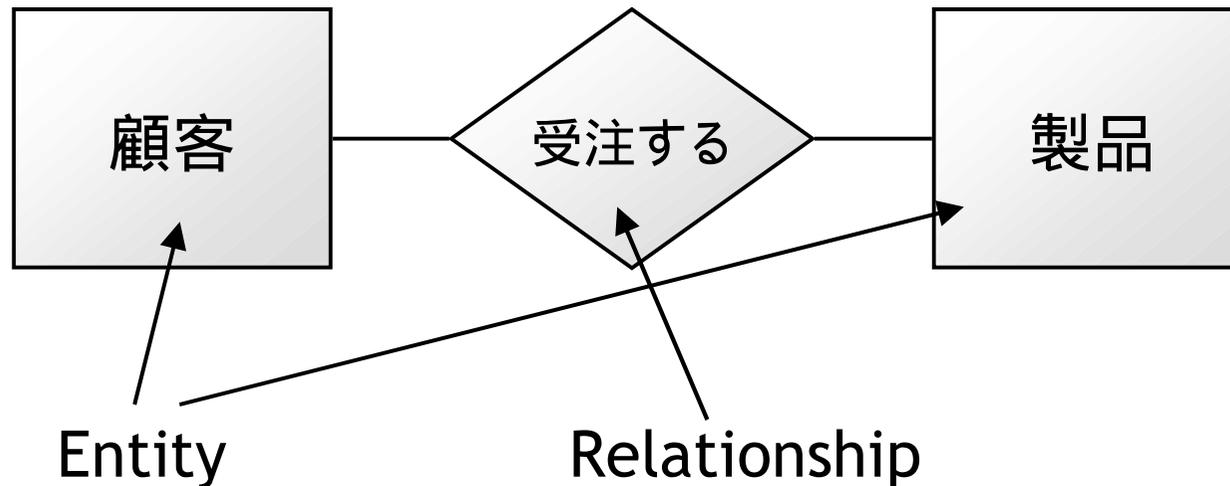
T字形ER図 = リレーショナル論理モデル

# T字形ER手法の狙い

- ビジネス解析
  - 既存のコード体系に着目した“As is” 解析
- 高次のデータ正規化
  - 定められた手順(後述)により自動的に実現
- アルゴリズムのI/O化
  - データI/O処理を、手続き型言語を使用せずに極力SQLで実施する

# ER手法

- P. P. Chen氏が提唱
  - The Entity-Relationship Model - Toward a Unified View of Data (1976)
- 事物および事象を存在(entity)と作用 (relationship)で記述する方法論

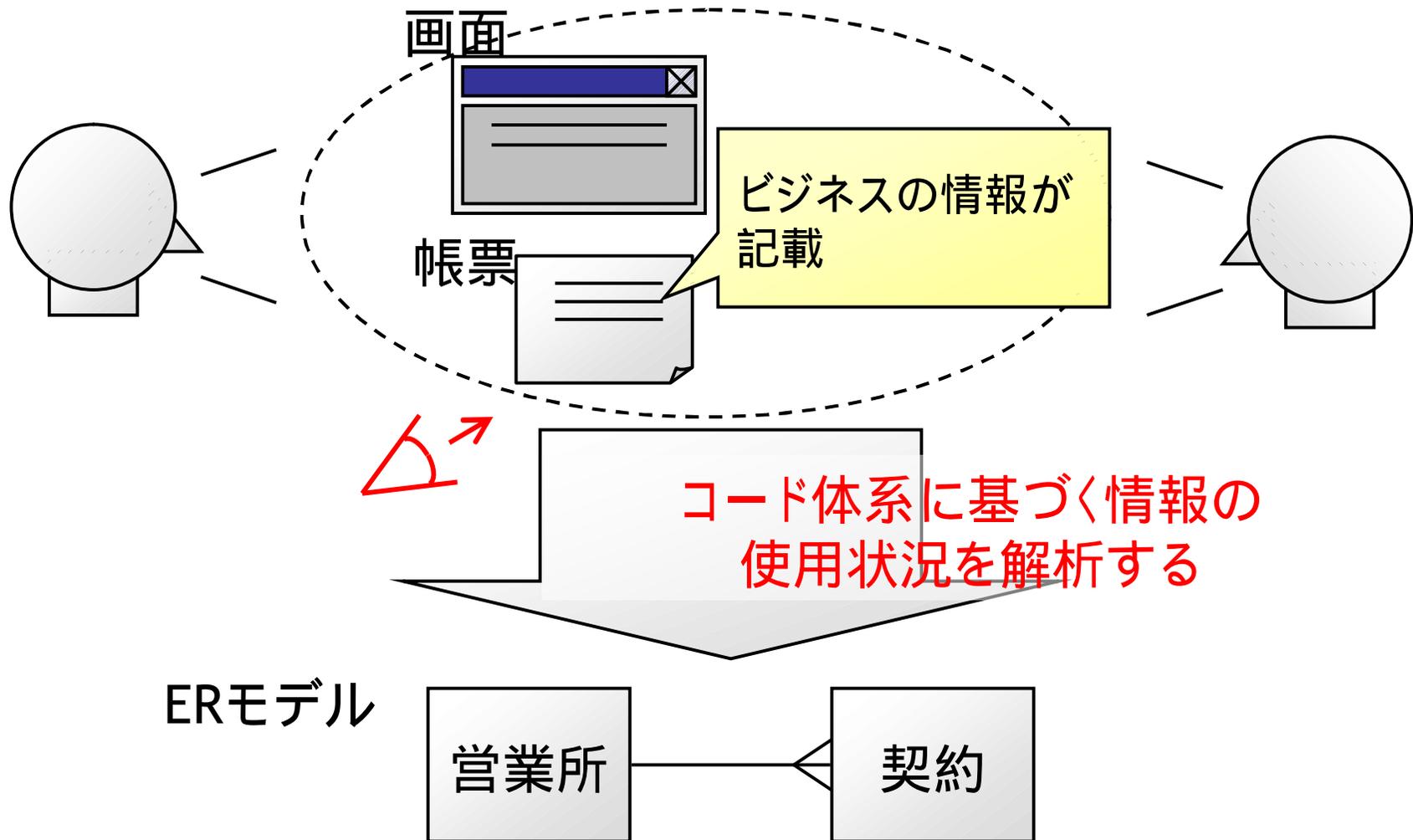


T字形ER手法は Chen ER手法を強化・発展させたもの

# T字形ER図の作成手順

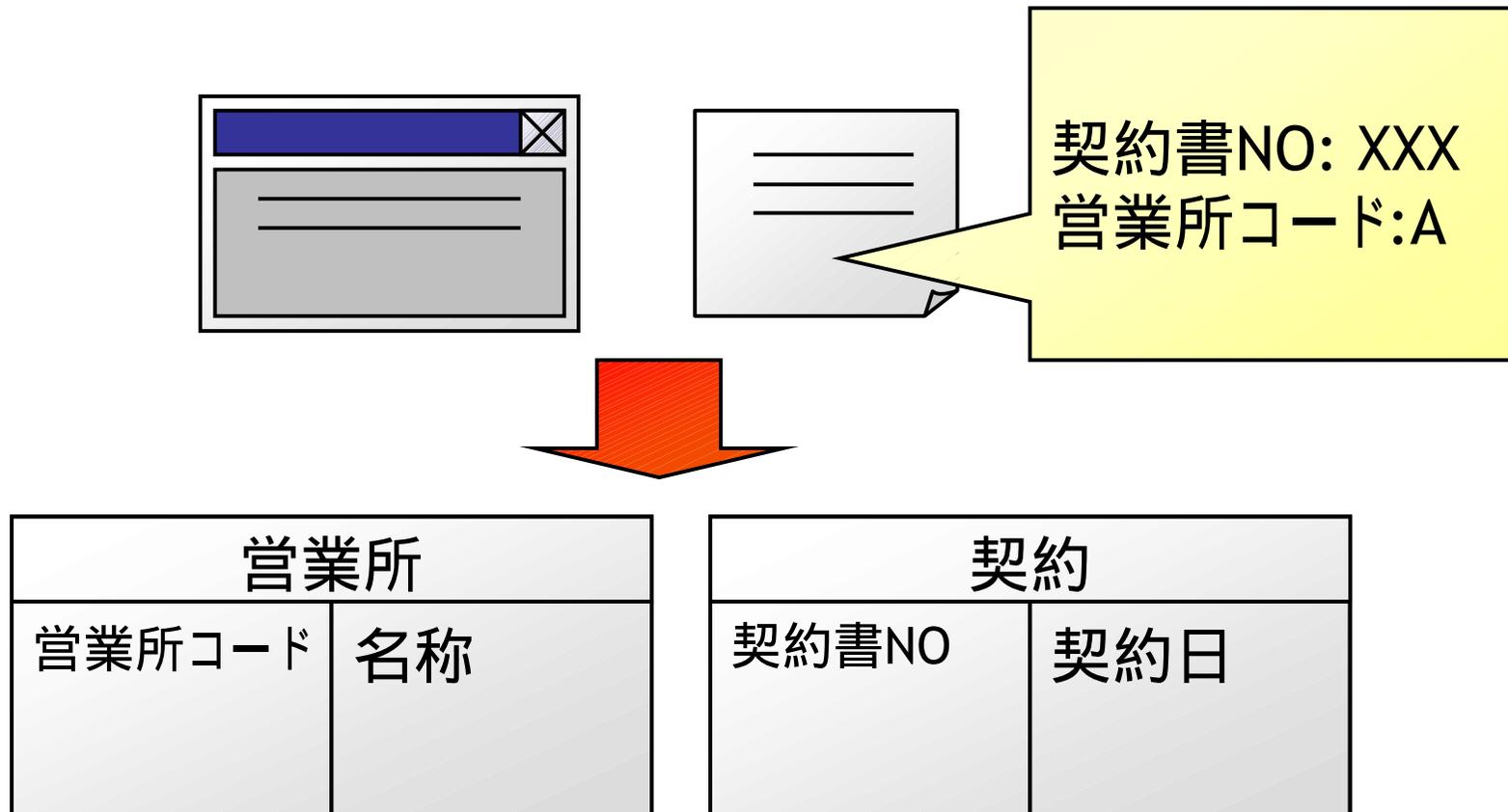
- 以下の手順に従い**ビジネス解析**を進め、**リレーショナル論理モデル**を得る。
  1. データの認知
  2. データの類別
  3. データの関係
  4. データの周延
  5. データの多義
  6. みなし

# T字形ER手法におけるビジネス解析の基本的考え方

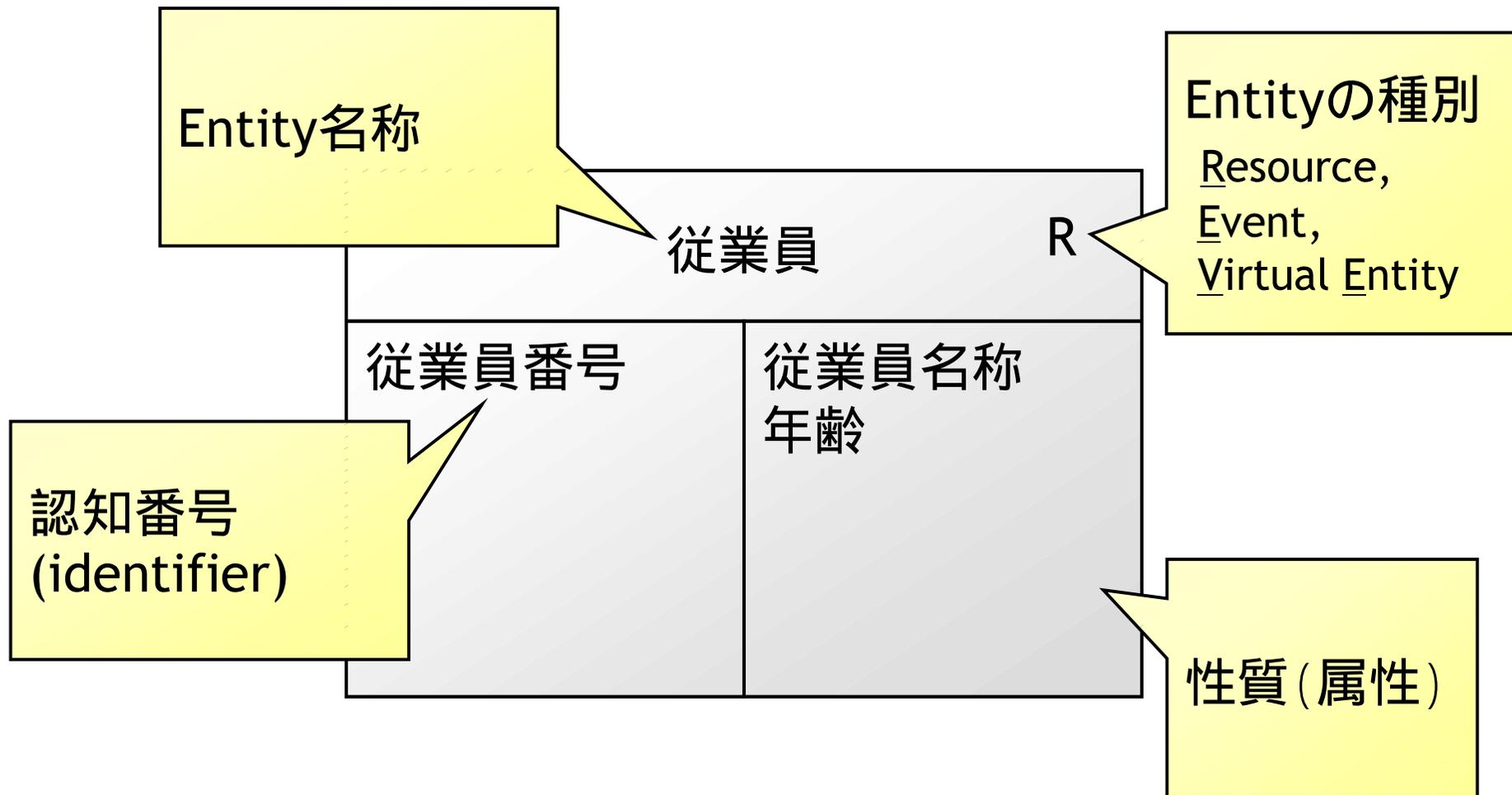


# 1. データの認知

- 画面、帳票などから**認知番号(コード、identifier)**を捕捉し、Entityとする。



# T字形ER手法の表記 - Entity



# なぜ認知番号(コード, identifier)を捕捉するだけで Entityが認知できるのか?

- 企業体における業務の遂行とコード
  - 企業はさまざまな人間のコラボレーションにより業務を遂行する
  - 企業は認知したいデータには必ずコードを振る
  - さまざまな人間がコードを頼りに情報をやり取りする
  - **コードは合意された公的な言語である**
    - 個々人の勝手な考えは排除される
    - 認識にブレがない

## 2. データの類別

- EntityをEventとResourceに分類する。
  - Entityの性質に“**DATE**”が帰属するものをEventとする。
  - それ以外をResourceとする。

Resource : いわゆるモノ

営業所		R
営業所コード	名称	

Event : いわゆるコト

契約		E
契約書NO	<b>契約日</b>	

- 「～する」が成立するかどうかで判断しても良い。

# なぜ「日(Date)」にこだわるのか?

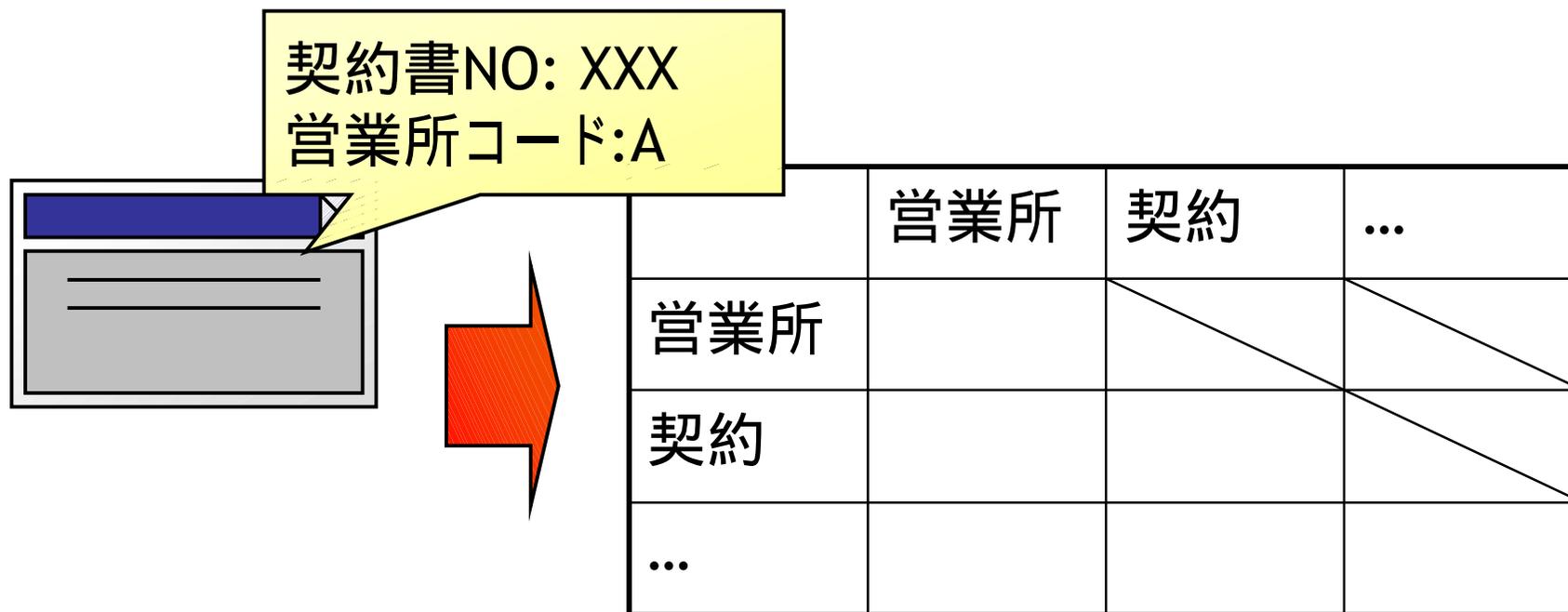
- データには時系列が問題となるものとならないものが存在する
  - Resource : 時系列関係なし!
    - 従業員と部署   あとさき関係なし
  - Event : 時系列関係あり!
    - 請求 出荷と出荷   請求 は別のビジネスモデルを表現する!
- 「日(Date)」を属性に持つかどうかで、**時系列が問題となるかどうか**が区別できる
- 関係のモデル化ルールにも関係する(後述)

## 3. データの関係

- 関係の捕捉
  - どのEntityとどのEntityの間に関係が成立するか？
- 関係のモデル化
  - 捕捉された関係を、どのようにリレーショナル論理モデルとしてモデル化するか？

# データの関係 - 関係の捕捉

- 画面、帳票といった**情報の使用状況**からEntityの包含関係、依存関係を捕捉する。
- 網羅性を確保するために、下記の様なテーブルを利用するのが一般的

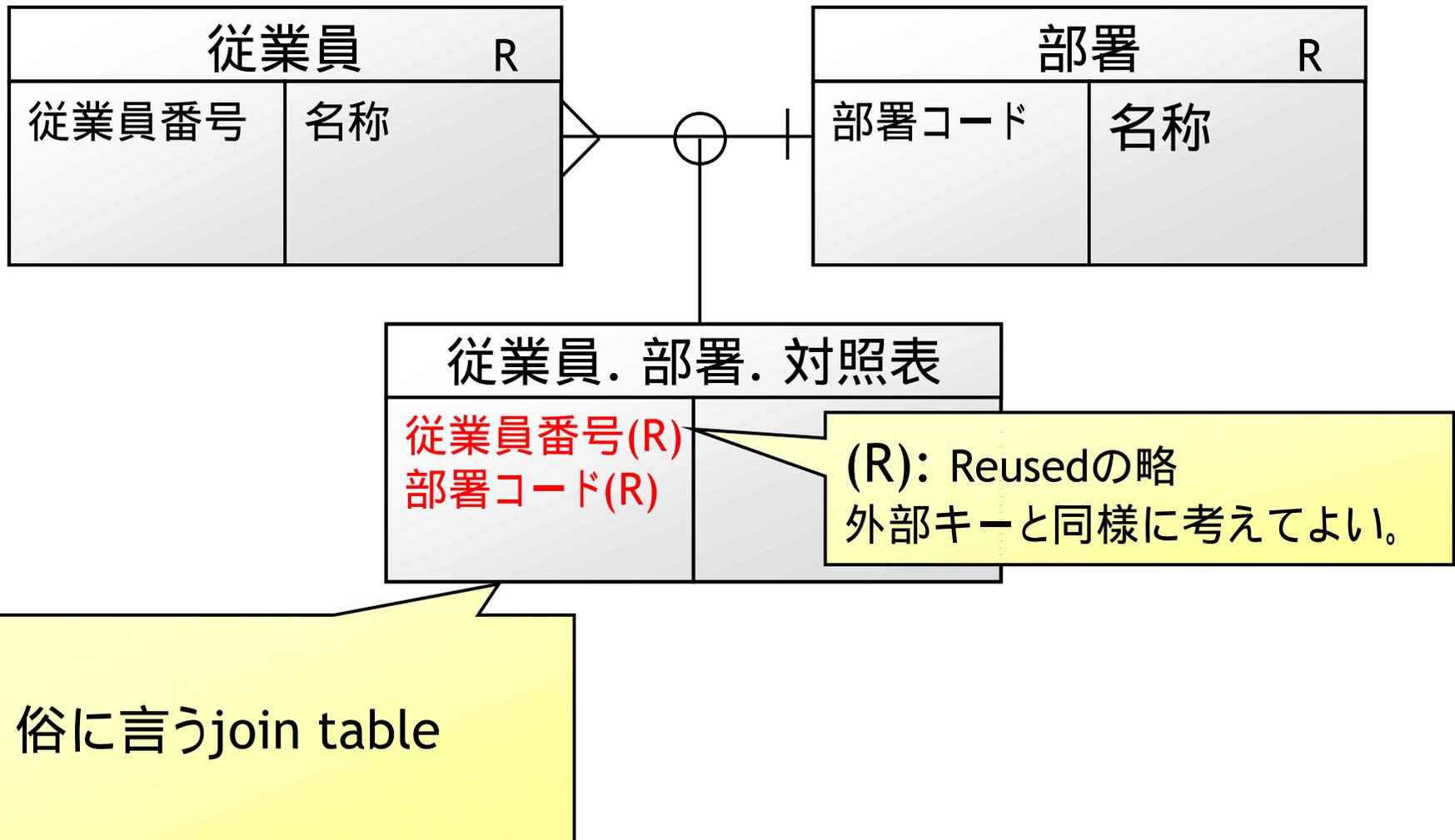


# データの関係 - 関係のモデル化ルール

- Resource : Resource → 対照表を生成
- Resource : Event → 認知番号をEventへ
- Event : Event
  - 1:1, 1:n → 先行Eventの認知番号を後続のEventへ
  - n:1, n:n → 対応表を生成
- (再帰の場合は今回は省略)
  
- では、具体的に...

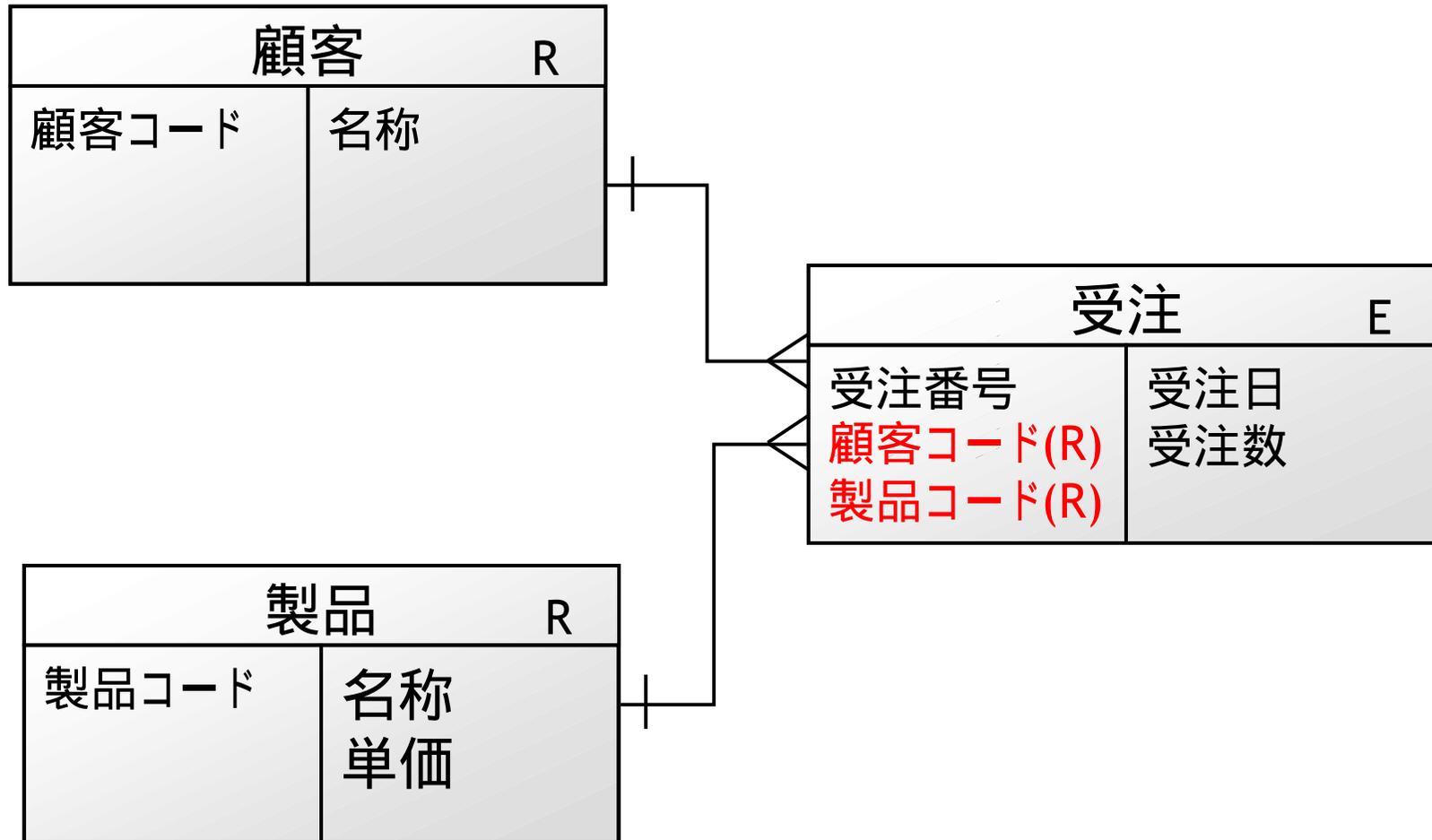
# データの関係 - Resource : Resource

- 対照表を生成



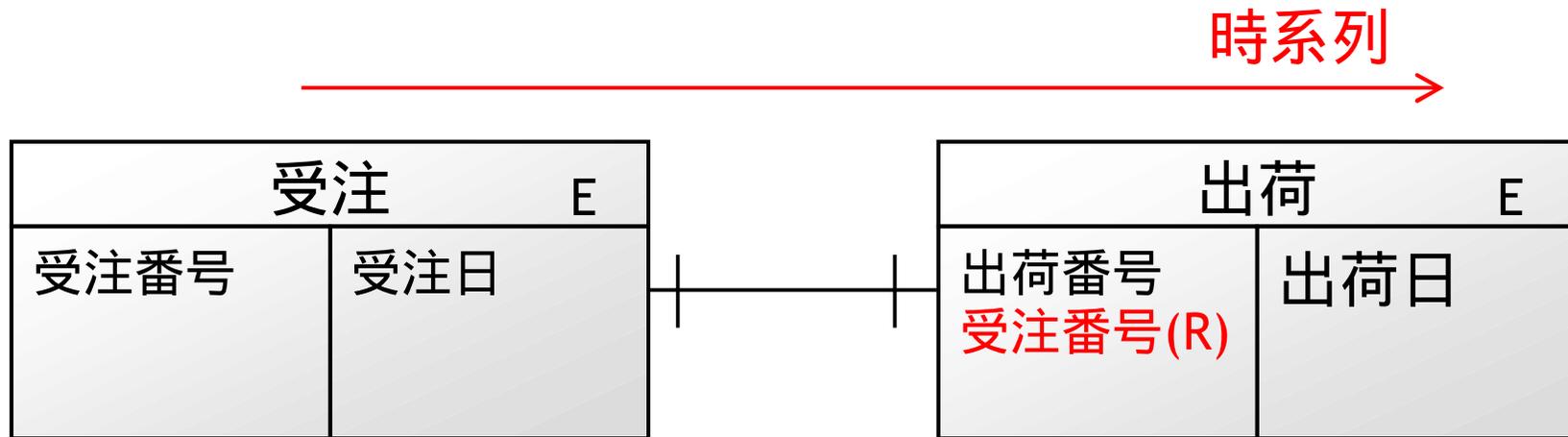
# データの関係 - Resource : Event

- Resourceの認知番号をEventへ



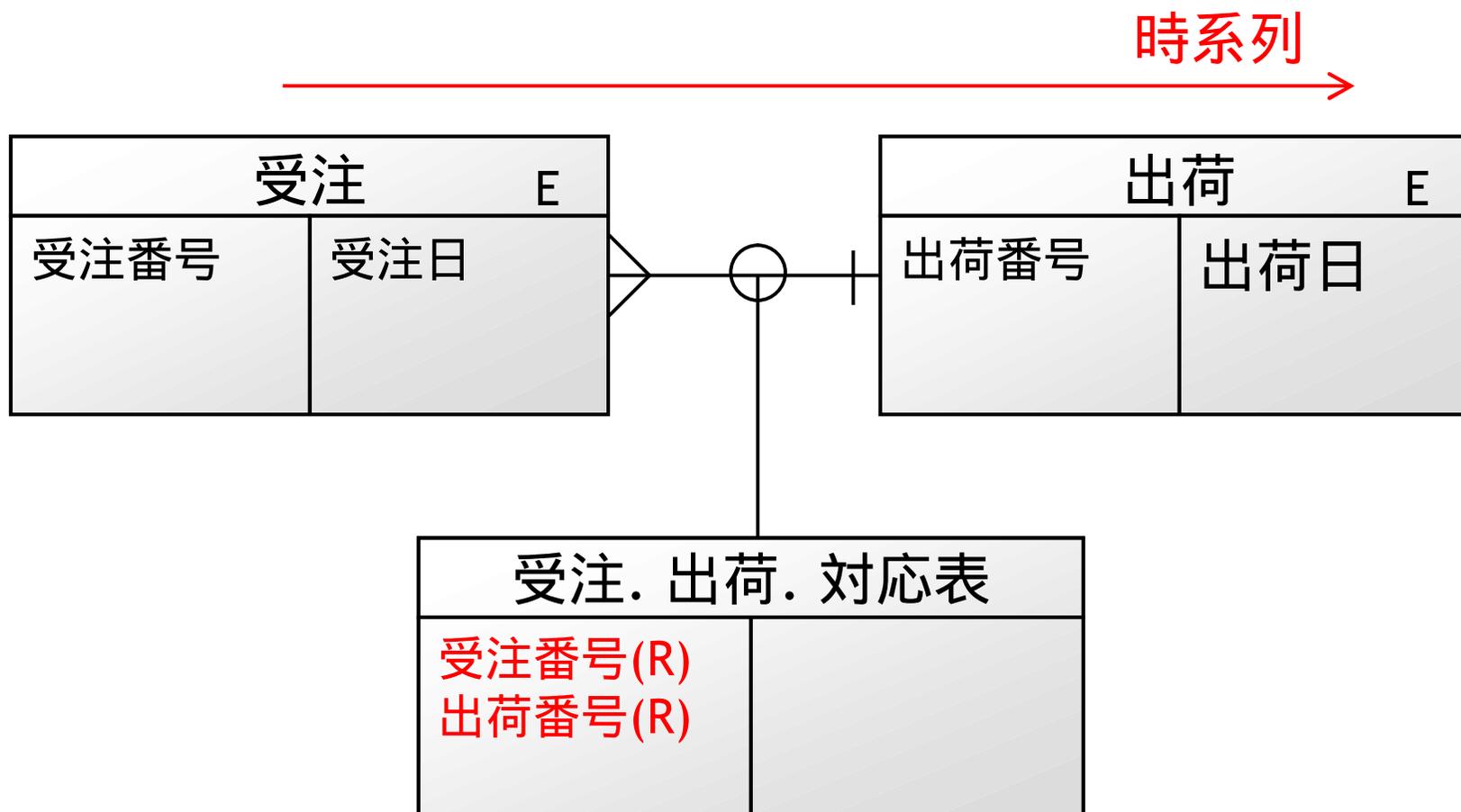
# データの関係 - Event : Event

- 1:1, 1:n → 先行Eventの認知番号を後続のEventに置く

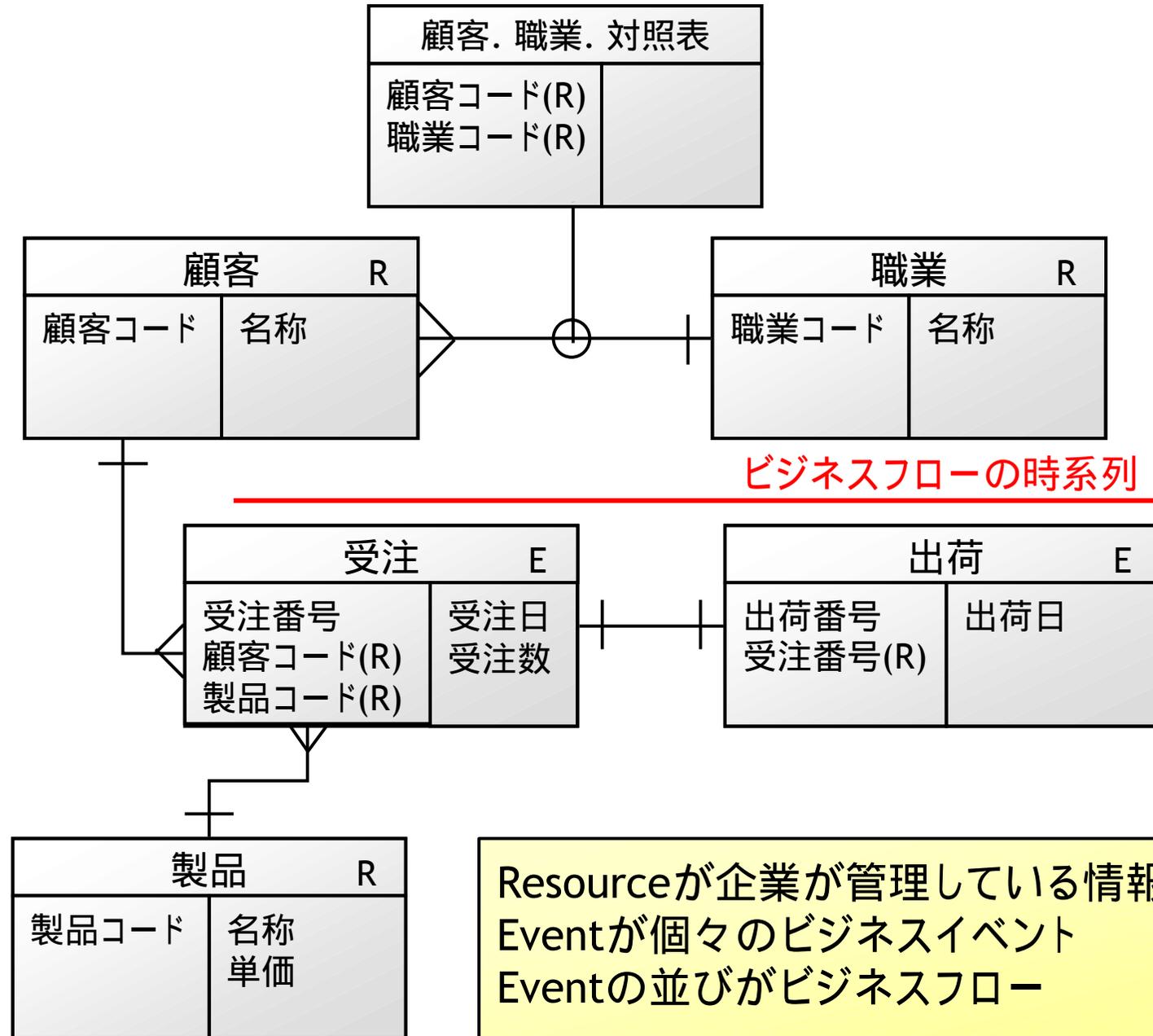


# データの関係 - Event : Event

- $n : 1, n : n \rightarrow$  対応表を置く



# ERモデル例

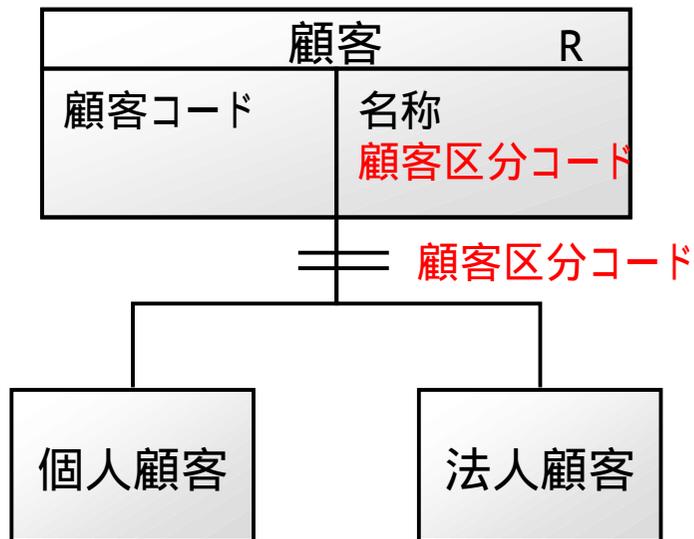


Resourceが企業が管理している情報資源  
Eventが個々のビジネスイベント  
Eventの並びがビジネスフロー

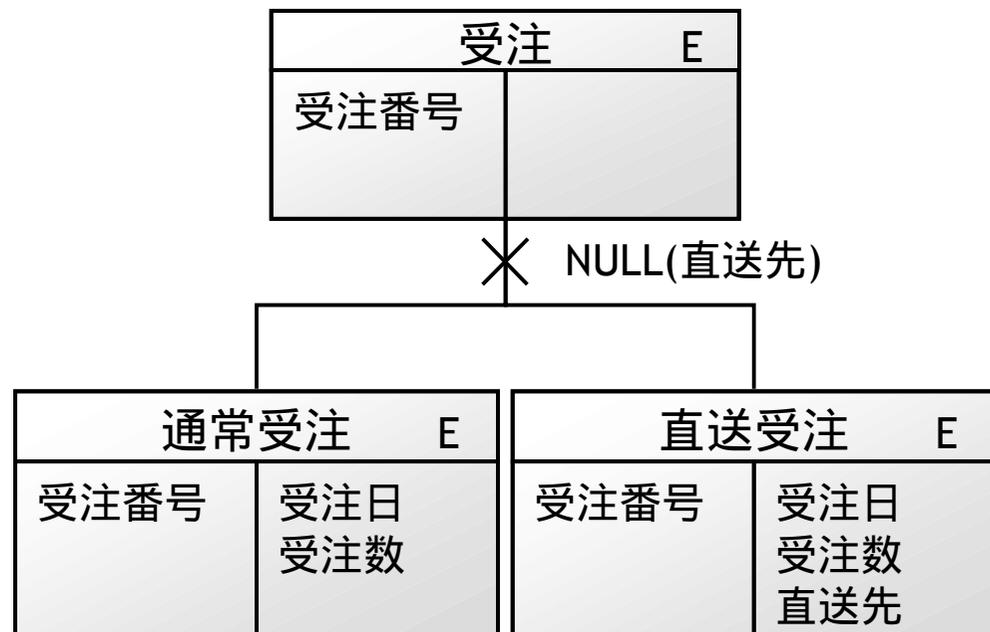
## 4. データの周延

- サブセット: 区分関係を明らかにする

同一のサブセット:  
アトリビュートの構成が同一

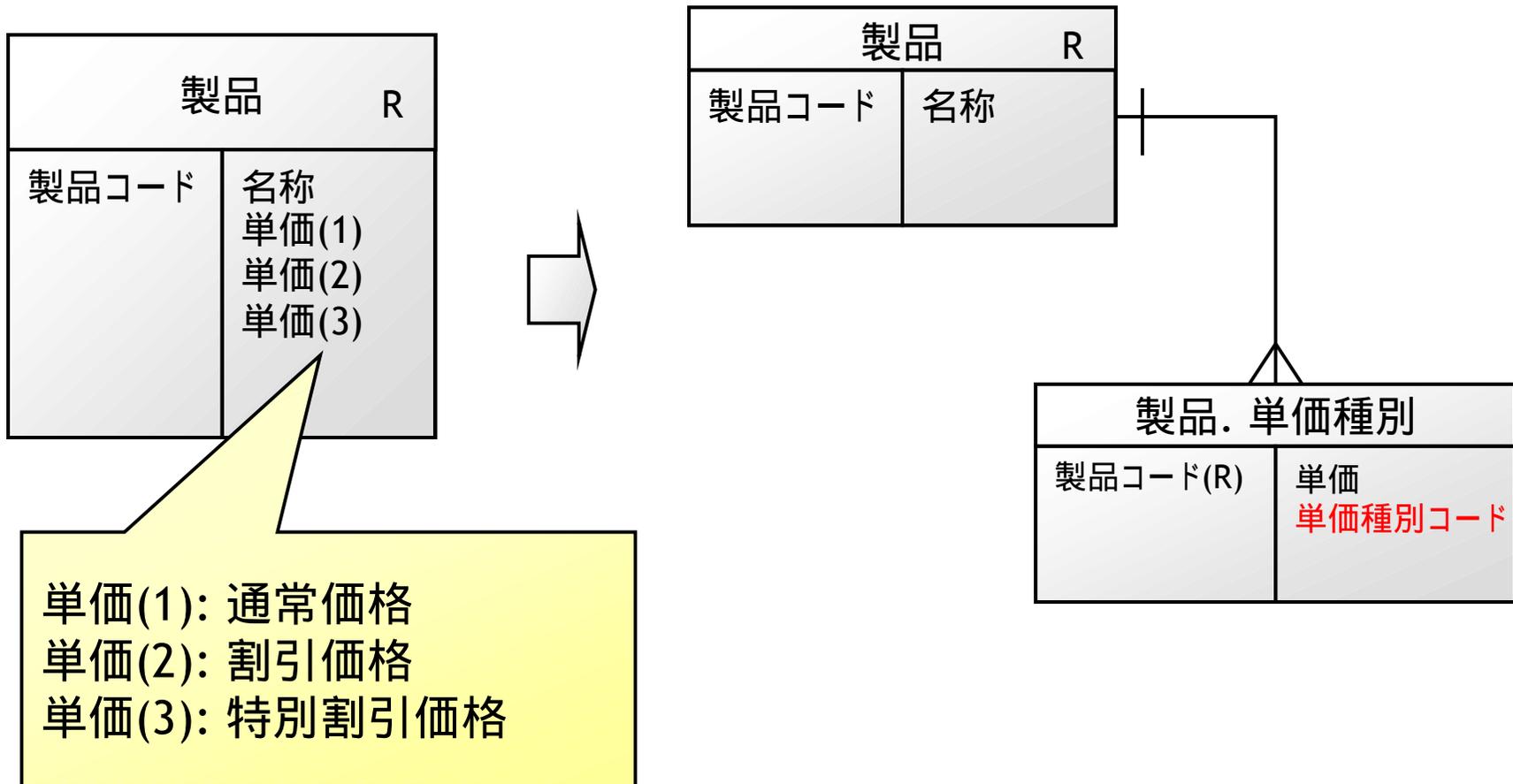


相違のサブセット:  
アトリビュートの構成が異なる



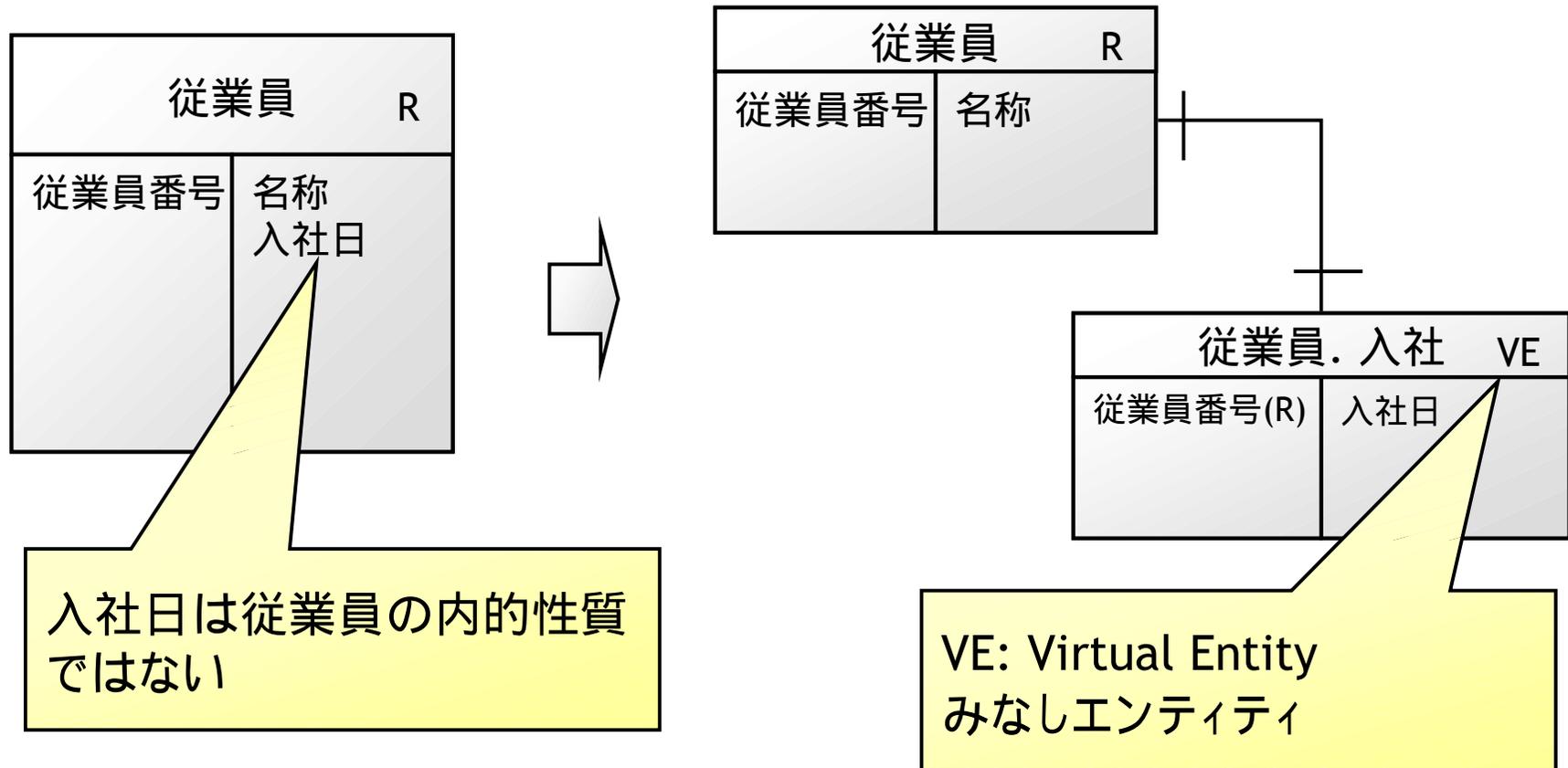
## 5. データの多義

- 繰り返し項目を排除する



## 6. みなし

- Entityの内的性質(本質的な性質)でない性質をEntityから排除する



## 【復習】T字形ER図の作成手順

1. データの認知: 認知番号(コード、identifier)が付与されているモノがEntityである。
2. データの類別: 性質としてDATEが帰属するEntityがEventである。それ以外のEntityはResourceである。
3. データの関係: 関係を捕捉し、ルールに従い関係をモデル化する。
4. データの周延: Entityをサブセットに区分する。
5. データの多義: 繰り返し項目を排除する。
6. みなし: 内的性質以外をEntityから追い出す。

# EOFへの展開

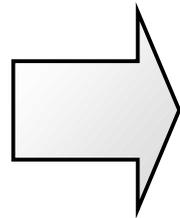
- リレーショナル論理モデルから実装モデルへの変換に相当する
- ポイントは以下の3つ(?)
  1. 主キーの扱い
  2. サブセットと継承
  3. 対照表と多重度

# 1. 主キーの扱い

- T字形ER手法では既存のコードを主キーに使用すると仮定している。
- 無意味な連番を主キーとしたほうがEOFと相性が良いはず。

T字形ERモデル Entity

顧客 R	
顧客コード	顧客名称



EOModel Entity

	属性	備考
PK	oid	連番
	顧客コード	ユニーク制約
	顧客名称	

ユニーク制約違反を拾う必要あり

## 2. サブセットと継承

- T字形ER手法 : サブセット
  - 同一のサブセット : アトリビュート構成が同一
  - 相違のサブセット : アトリビュート構成が異なる
- EOF : 継承
  - 垂直継承
  - 水平継承
  - 単一テーブル継承
- 同一のサブセット      単一テーブル継承
- 相違のサブセット      垂直/水平継承

### 3. 対照表と多重度

- T字形ER手法
  - Resource: Resourceの関係は**多重度によらず**常に対照表( = JOINテーブル)を用いてモデル化する。
- EOF
  - **多対多関連の場合のみ**JOINテーブルを使用する。
- 上手いやり方募集中・・・。

# まとめ

- T字形ER手法は、既存業務が存在する場合、ほぼ機械的にERモデリングを実行できる方法論である
  - ビジネス解析としても使える(とのこと)
- RDB設計方法論としても示唆に富む
  - 「関係」のモデル化方法
  - 区分コード、種別コード、(分類コード)の使用
- EOFへの展開については今後の検討課題

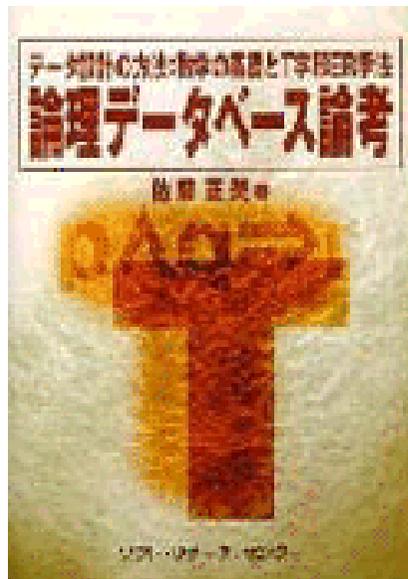
## 関連URL

- 株式会社SDI
  - <http://www.sdi-net.co.jp/>
  - 代表取締役 佐藤正美氏 (T字形ER手法の提唱者)
- FactoryWR (笑)
  - <http://www.csus4.net/WR/d/>
  - T字形ER手法の学習記録、T字形ER手法セミナーの講義メモ、etc...

# 関連書籍



- T字形ER図の作成手順が中心
- まあまあオススメ



- 記号論理学、集合論などが半分
- T字形ER手法が半分
- 洗練(?)されすぎていてかなり読みにくい
- 正直 オススメできない

## Q&A

- お手柔らかに . . .