



DB設計とテーブル間連携

今日の講義内容

- ①データベースの基本設計
- ②データベースの正規化
- ③PHPとSQLで実践

データの理解

開発するアプリケーションには一体どのようなデータが存在するのかを洗い出す。

登録・表示・更新・削除について理解すること。
(Create/Read/Update/Delete)

良い設計のデータベースとは？

参照（検索・取得）と更新が行いやすい

画面を表示する際にSelect文が書きやすいか

検索のパフォーマンスは良いのか

データ更新において、矛盾なく追加/更新/削除ができるのか

将来的な更新に強い

機能追加の際に簡単に追加できるか

DB設計 Step1 データの洗い出し

対象の物事を概念化し具体的な名前をつけて表現してみましょう

エンティティ(テーブル)

商品
商品コード
商品名
単価
在庫数

← エンティティ名

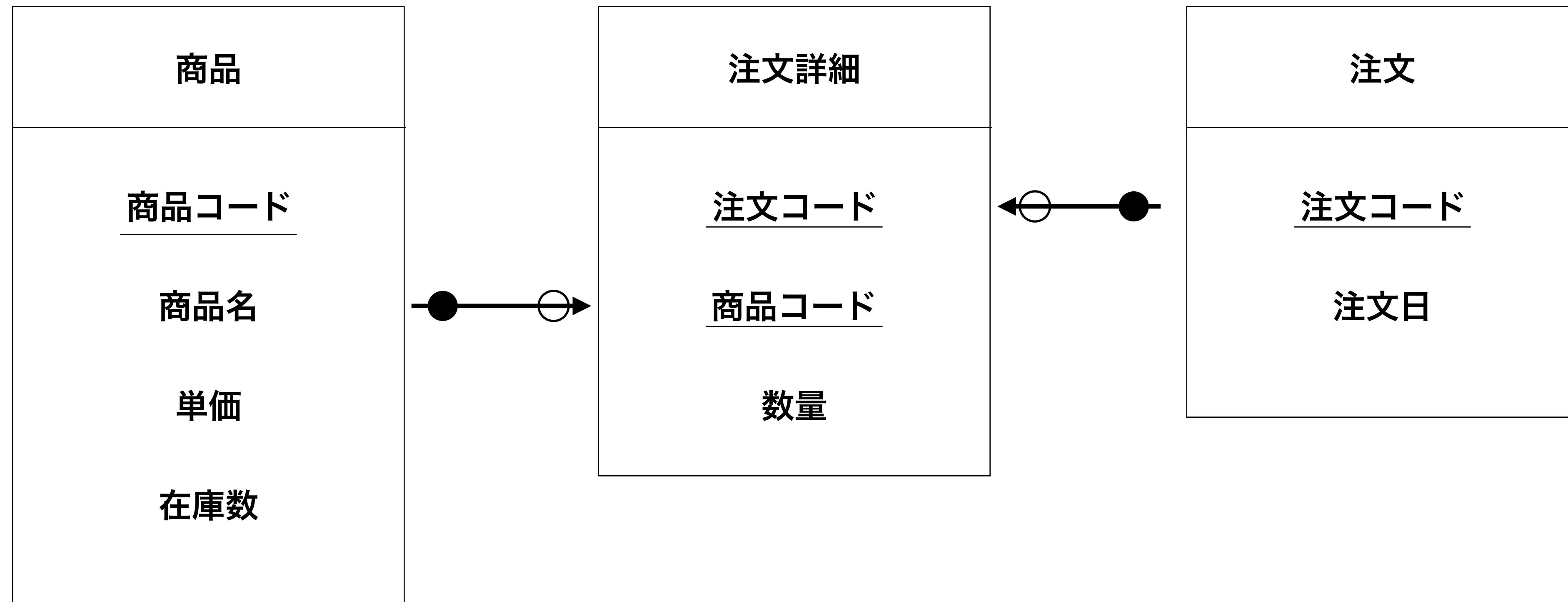
← 属性名

インスタンス (レコード)

1	りんご	¥200	30
2	みかん	¥100	10

DB設計 Step2 データ同士の関連性を書き出す

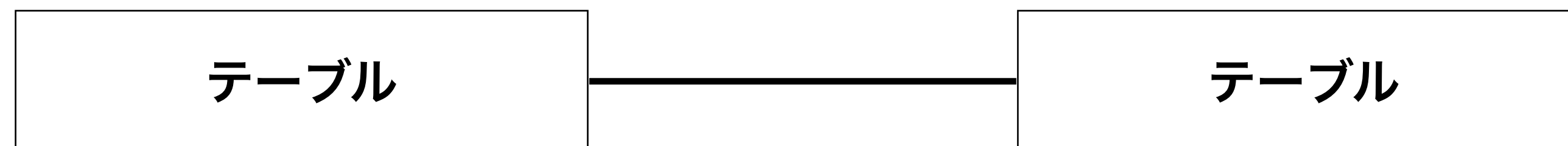
エンティティ（テーブル）間でのデータ関係性をリレーションシップと言います



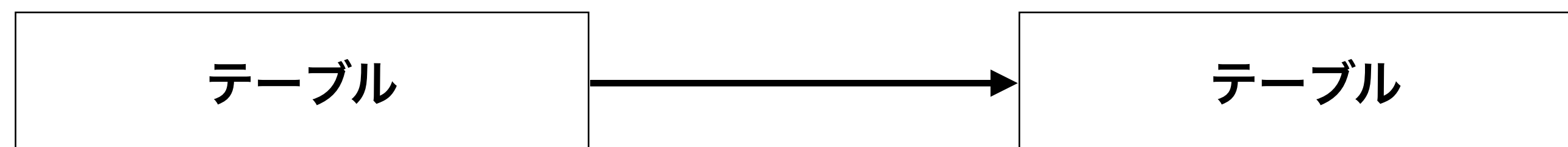
DB設計 リレーションの種類

リレーションは3種類あります。

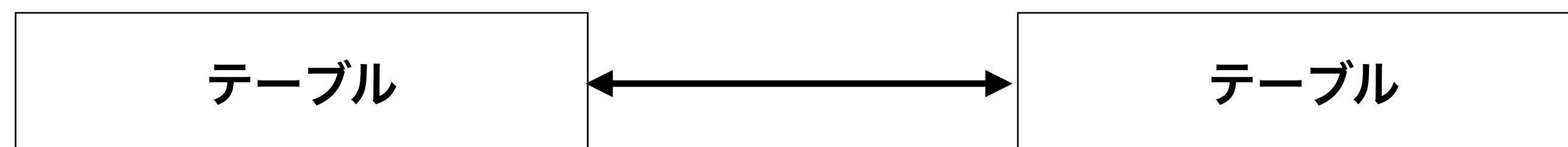
1対1



1対多

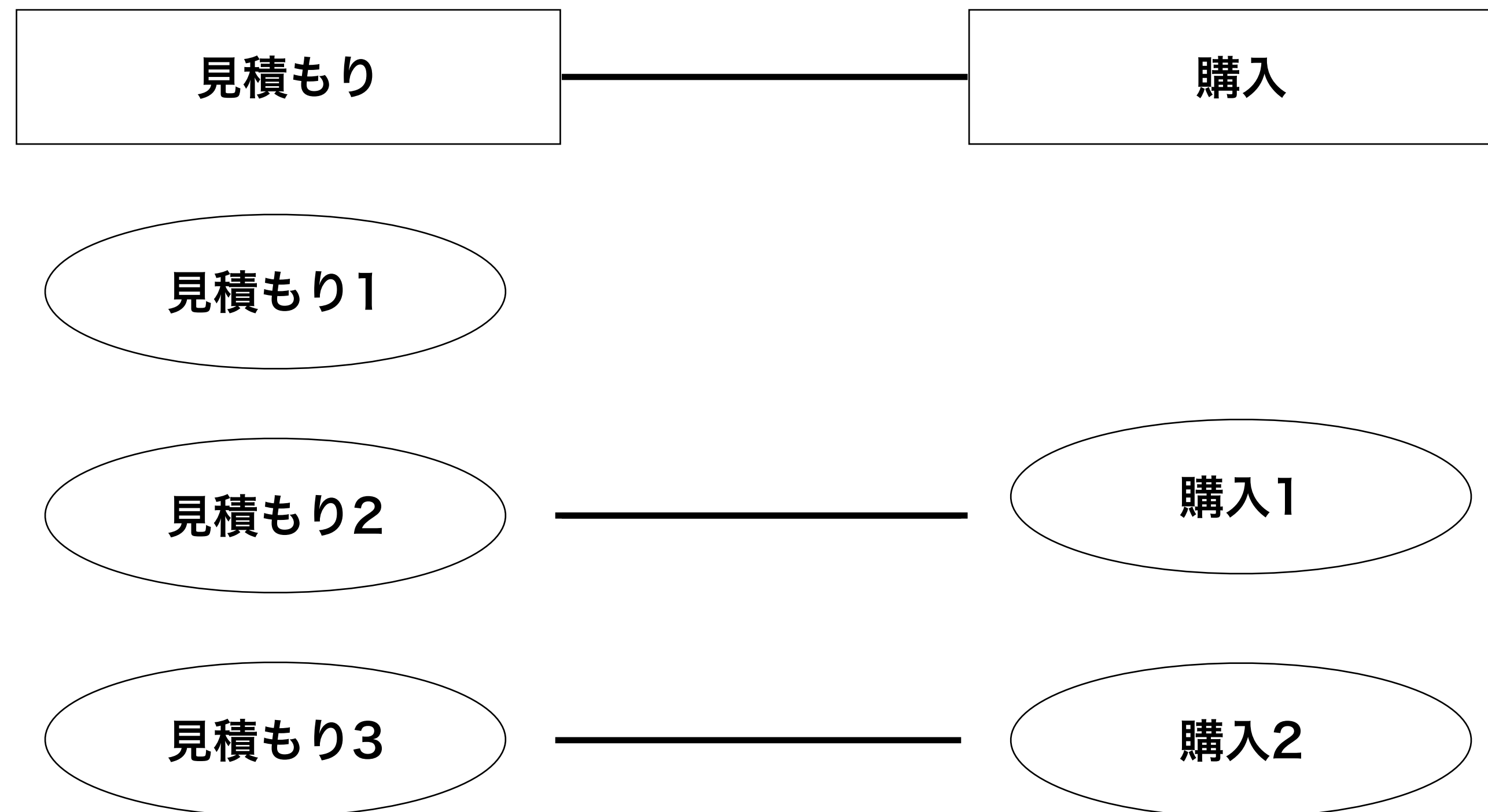


多対多



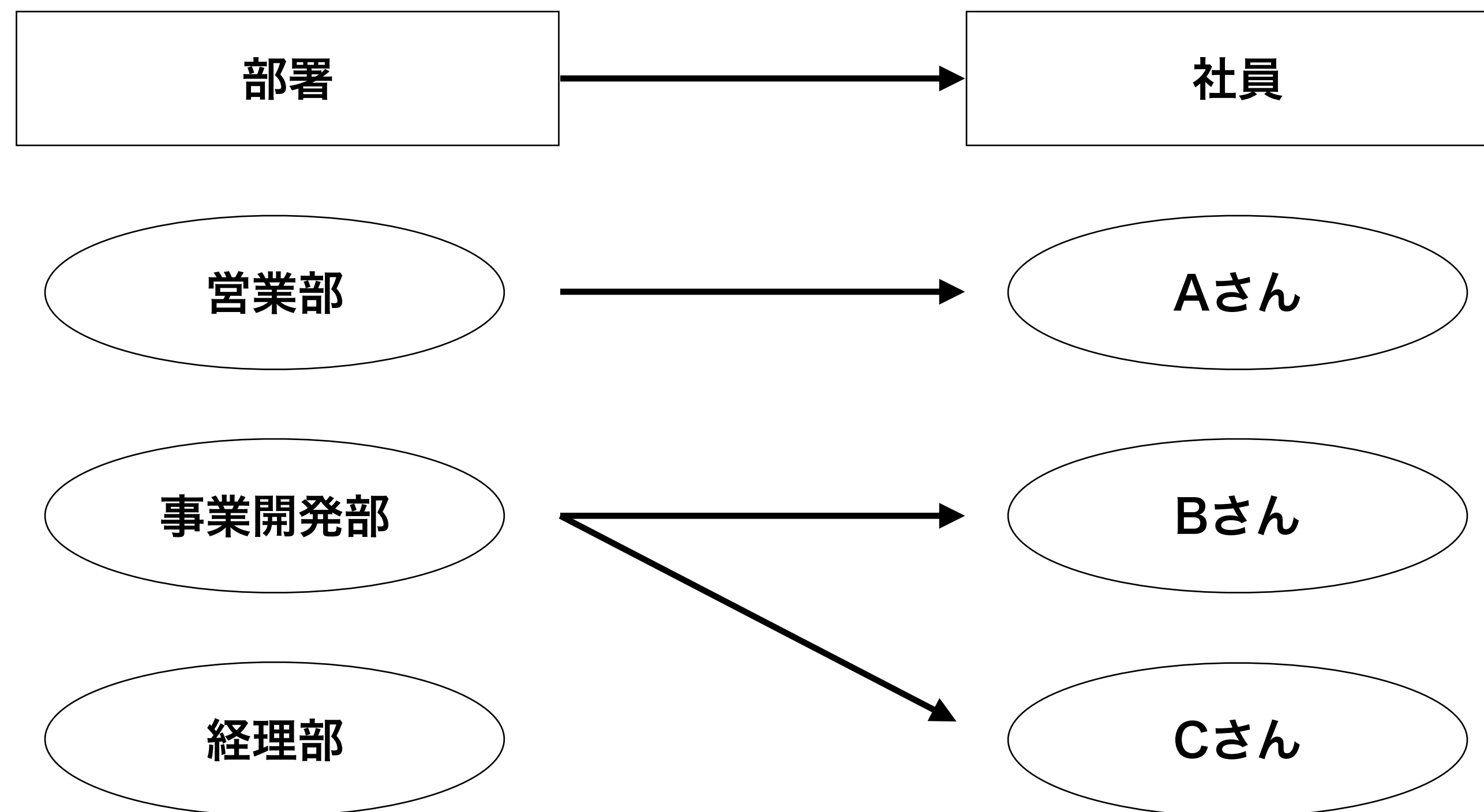
DB設計 リレーション：1対1

1対1の場合は一番簡単な関係性。



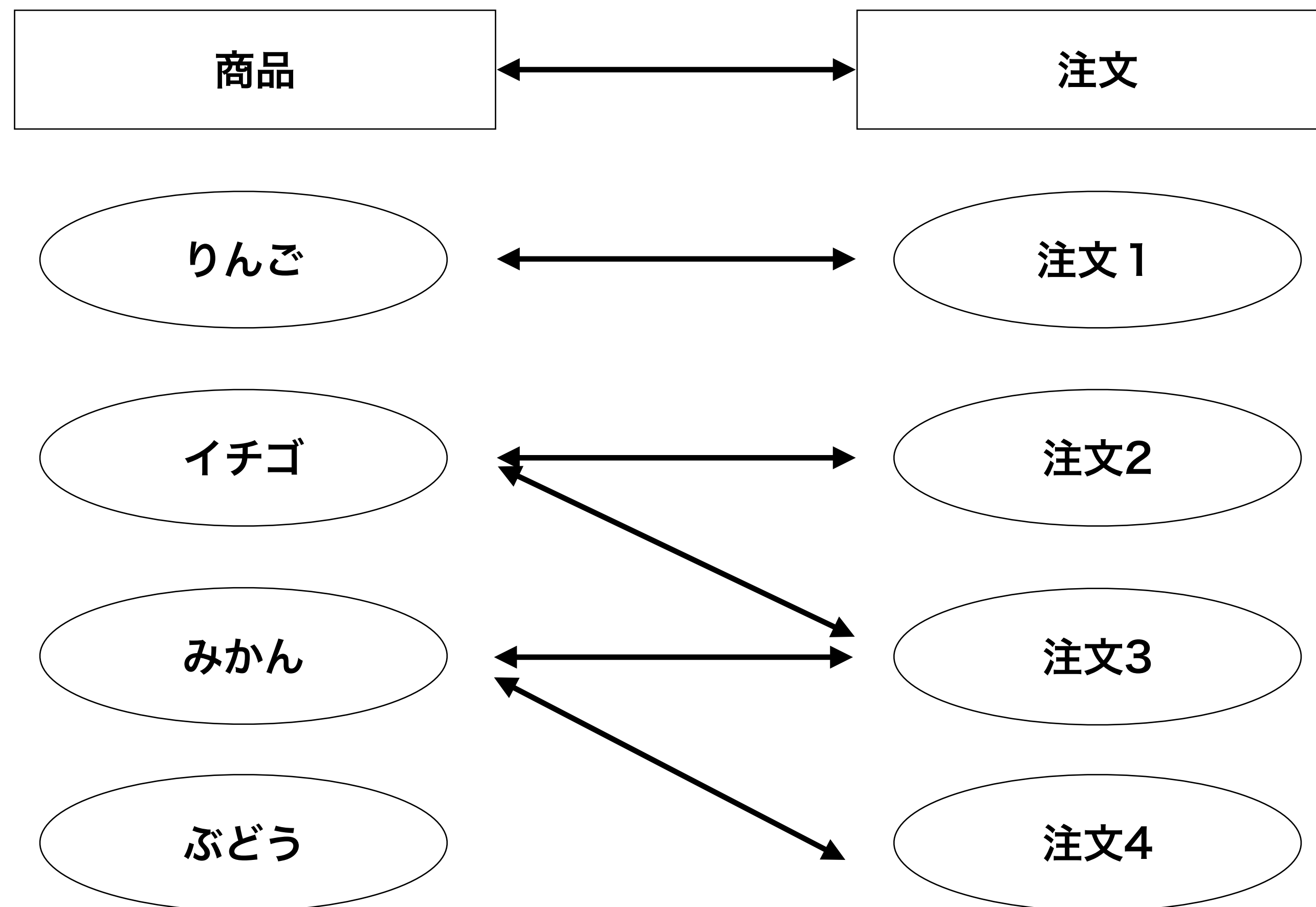
DB設計 リレーション：1対多

部署から見ると複数の社員がいる。社員から見ると一つの部署に所属している



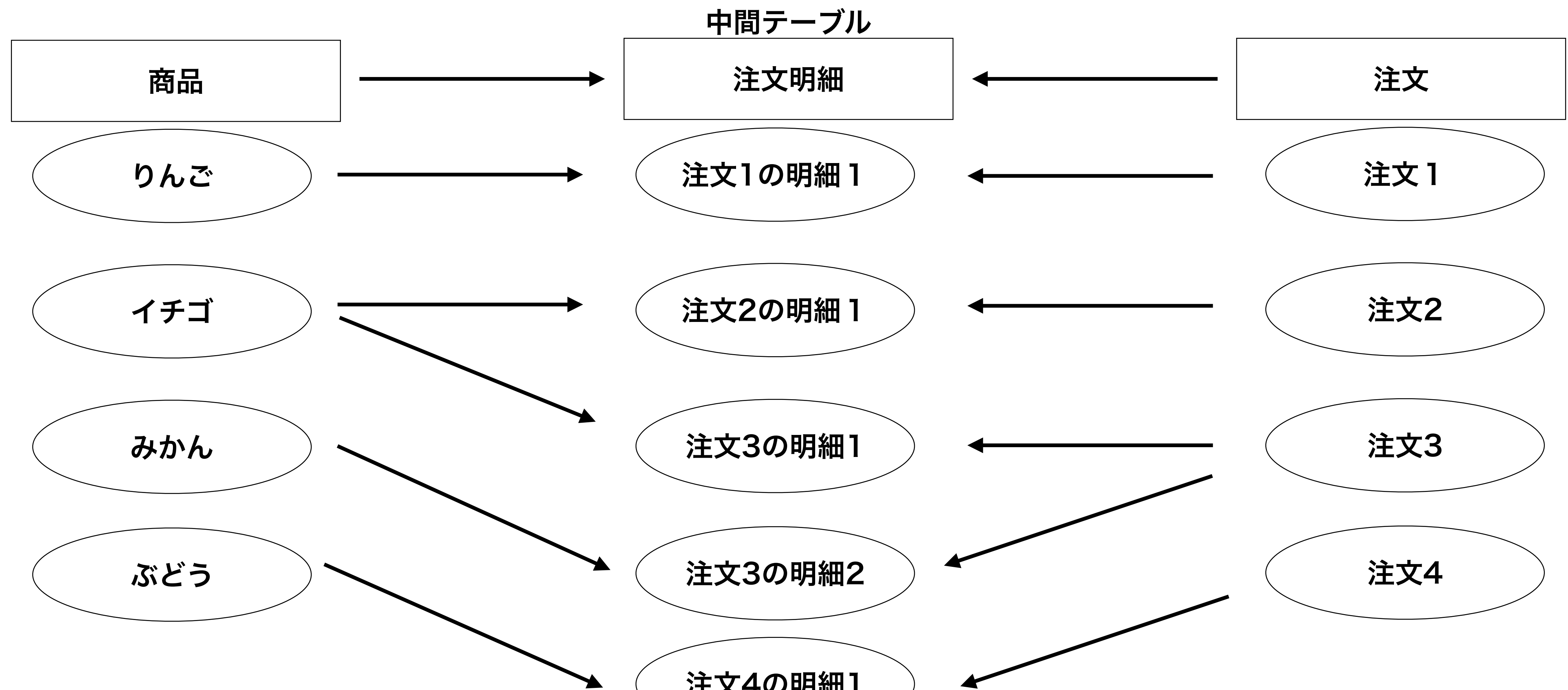
DB設計 リレーション：多対多

商品からは複数の注文で、注文から見ると複数の商品がある関係。



DB設計 リレーション：多対多の表現方法

MySQLなどのRDBでは表現できないので、1対多で表現する



DB設計 Step3 データベースの正規化

伝票番号	店舗ID	店舗名	店舗住所	売上日	商品明細					合計	消費税	請求額	担当者ID	担当者名
					商品コード	商品名	単価	数量	小計					
1	1	原宿店	東京都〇〇	2021/07/07	A001	りんご	100	2	200	2200	220	2429	2001	鈴木
					B004	みかん	200	5	1000					
					B012	イチゴ	1000	1	1000					
2	2	表参道店	東京都〇〇	2021/07/08	A001	りんご	100	2	200	5200	520	5720	3301	田中
					B012	イチゴ	1000	5	5000					



伝票番号	店舗ID	売上日	合計	消費税	請求額	担当者ID
1	1	2021/07/07	2200	220	2420	2001
2	2	2021/07/08	5200	520	5720	3301

店舗ID	店舗名	店舗住所
1	原宿店	東京都〇〇
2	表参道店	東京都〇〇

伝票番号	商品コード	数量	小計
1	A001	2	200
1	B004	5	1000
1	B012	1	1000
2	りんご	2	200
2	イチゴ	5	5000

担当者ID	担当者名
2001	鈴木
3301	田中

商品コード	商品名	単価
A001	りんご	100
B004	みかん	200
B012	イチゴ	1000

非正規化状態

正規化前の状態です。繰り返し項目などが存在します。

果物屋さんの注文データ

伝票番号	店舗ID	店舗名	店舗住所	売上日	商品明細					合計	消費税	請求額	担当者ID	担当者名
					商品コード	商品名	単価	数量	小計					
1	1	原宿店	東京都〇〇	2021/07/07	A001	りんご	100	2	200	2200	220	2429	2001	鈴木
					B004	みかん	200	5	1000					
					B012	イチゴ	1000	1	1000					
2	2	表参道店	東京都〇〇	2021/07/08	A001	りんご	100	2	200	5200	520	5720	3301	田中
					B012	イチゴ	1000	5	5000					

この状態で運用すると問題点や面倒な部分があります。

第1正規化

繰り返し項目をなくするのが第一正規化です。

果物屋さんの注文データ

伝票番号	店舗ID	店舗名	店舗住所	売上日	商品明細					合計	消費税	請求額	担当者ID	担当者名
					商品コード	商品名	単価	数量	小計					
1	1	原宿店	東京都〇〇	2021/07/07	A001	りんご	100	2	200	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B004	みかん	200	5	1000	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B012	イチゴ	1000	1	1000	2200	220	2429	2001	鈴木
2	2	表参道店	東京都〇〇	2021/07/08	A001	りんご	100	2	200	5200	520	5720	3301	田中
2	2	表参道店	東京都〇〇	2021/07/08	B012	イチゴ	1000	5	5000	5200	520	5720	3301	田中

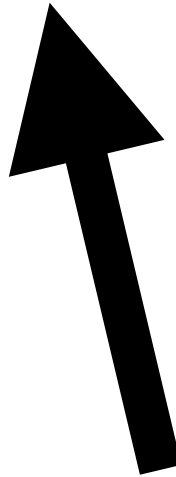
基本キーを定義

基本キー（主キー）を基準に第2正規化を行います。

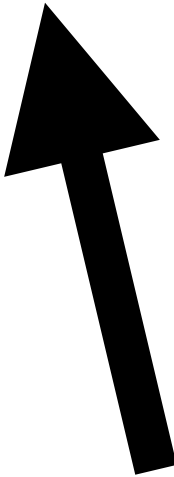
基本キー（主キー）は特定の1レコードを抽出するのに必要な一意なキー

果物屋さんの注文データ

伝票番号	店舗ID	店舗名	店舗住所	売上日	商品明細					合計	消費税	請求額	担当者ID	担当者名
					商品コード	商品名	単価	数量	小計					
1	1	原宿店	東京都〇〇	2021/07/07	A001	りんご	100	2	200	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B004	みかん	200	5	1000	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B012	イチゴ	1000	1	1000	2200	220	2429	2001	鈴木
2	2	表参道店	東京都〇〇	2021/07/08	A001	りんご	100	2	200	5200	520	5720	3301	田中
2	2	表参道店	東京都〇〇	2021/07/08	B012	イチゴ	1000	5	5000	5200	520	5720	3301	田中



基本キー 1



基本キー 2

第2正規化

第2正規化では、部分関数従属を排除します。

果物屋さんの注文データ

伝票番号	店舗ID	店舗名	店舗住所	売上日	商品明細					合計	消費税	請求額	担当者ID	担当者名
					商品コード	商品名	単価	数量	小計					
1	1	原宿店	東京都〇〇	2021/07/07	A001	りんご	100	2	200	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B004	みかん	200	5	1000	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B012	イチゴ	1000	1	1000	2200	220	2429	2001	鈴木
2	2	表参道店	東京都〇〇	2021/07/08	A001	りんご	100	2	200	5200	520	5720	3301	田中
2	2	表参道店	東京都〇〇	2021/07/08	B012	イチゴ	1000	5	5000	5200	520	5720	3301	田中

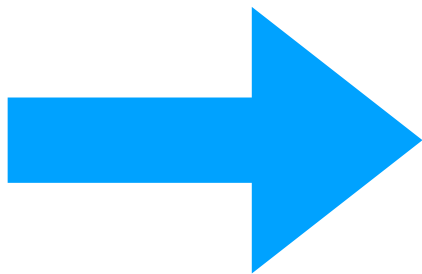
部分関数従属とは

あるデータが基本キーの一部に従属している状態です。

伝票番号	商品コード	商品名
------	-------	-----



部分関数従属



伝票番号	商品コード
商品コード	商品名

第2正規化

伝票番号のみに従属するデータと、商品コードにのみ従属するデータが存在します。

伝票番号	店舗ID	店舗名	店舗住所	売上日	商品明細					合計	消費税	請求額	担当者ID	担当者名
					商品コード	商品名	単価	数量	小計					
1	1	原宿店	東京都〇〇	2021/07/07	A001	りんご	100	2	200	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B004	みかん	200	5	1000	2200	220	2429	2001	鈴木
1	1	原宿店	東京都〇〇	2021/07/07	B012	イチゴ	1000	1	1000	2200	220	2429	2001	鈴木
2	2	表参道店	東京都〇〇	2021/07/08	A001	りんご	100	2	200	5200	520	5720	3301	田中
2	2	表参道店	東京都〇〇	2021/07/08	B012	イチゴ	1000	5	5000	5200	520	5720	3301	田中

第2正規化

部分関数従属のデータは、エンティティ（テーブル）を別に作成します。

注文明細

伝票番号	商品コード	数量	小計
1	A001	2	200
1	B004	5	1000
1	B012	1	1000
2	りんご	2	200
2	イチゴ	5	5000

商品

商品コード	商品名	単価
A001	りんご	100
B004	みかん	200
B012	イチゴ	1000

注文

伝票番号	店舗ID	店舗名	店舗住所	売上日	合計	消費税	請求額	担当者ID	担当者名
1	1	原宿店	東京都〇〇	2021/07/07	2200	220	2420	2001	鈴木
2	2	表参道店	東京都〇〇	2021/07/08	5200	520	5720	3301	田中

第3正規化

第3正規化では、推移関数従属を排除します。

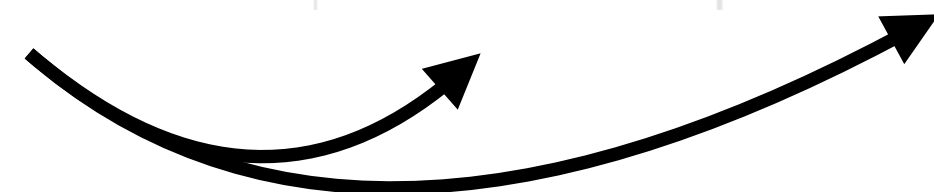
注文

伝票番号	店舗ID	店舗名	店舗住所	売上日	合計	消費税	請求額	担当者ID	担当者名
1	1	原宿店	東京都〇〇	2021/07/07	2200	220	2420	2001	鈴木
2	2	表参道店	東京都〇〇	2021/07/08	5200	520	5720	3301	田中

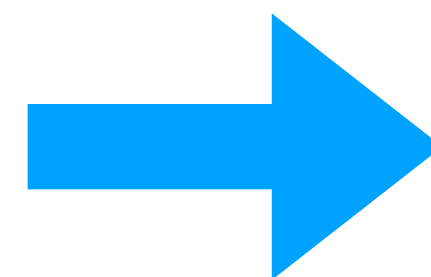
推移関数従属とは

基本キー以外のデータ同士で従属関係がある状態です。

伝票番号	店舗ID	店舗名	店舗住所
------	------	-----	------



推移関数従属



テーブルを分割

伝票番号	店舗ID
------	------

店舗ID	店舗名	店舗住所
------	-----	------

第3正規化

注文

伝票番号	店舗ID	売上日	合計	消費税	請求額	担当者ID
1	1	2021/07/07	2200	220	2420	2001
2	2	2021/07/08	5200	520	5720	3301

店舗

店舗ID	店舗名	店舗住所
1	原宿店	東京都〇〇
2	表参道店	東京都〇〇

注文詳細

伝票番号	商品コード	数量	小計
1	A001	2	200
1	B004	5	1000
1	B012	1	1000
2	りんご	2	200
2	イチゴ	5	5000

担当者

担当者ID	担当者名
2001	鈴木
3301	田中

商品

商品コード	商品名	単価
A001	りんご	100
B004	みかん	200
B012	イチゴ	1000

ちなみに

算出できる項目はエンティティ（テーブル）に入れないのが基本です。

注文

伝票番号	店舗ID	売上日	合計	消費税	請求額	担当者ID
1	1	2021/07/07	2200	220	2420	2001
2	2	2021/07/08	5200	520	5720	3301

店舗

店舗ID	店舗名	店舗住所
1	原宿店	東京都〇〇
2	表参道店	東京都〇〇

注文詳細

伝票番号	商品コード	数量	小計
1	A001	2	200
1	B004	5	1000
1	B012	1	1000
2	りんご	2	200
2	イチゴ	5	5000

担当者

担当者ID	担当者名
2001	鈴木
3301	田中

商品

商品コード	商品名	単価
A001	りんご	100
B004	みかん	200
B012	イチゴ	1000

DB設計 Step4 列名の定義

エンティティと属性の論理名と物理名を決めていきます。

商品

商品コード	商品名	単価
A001	りんご	100
B004	みかん	200
B012	イチゴ	1000

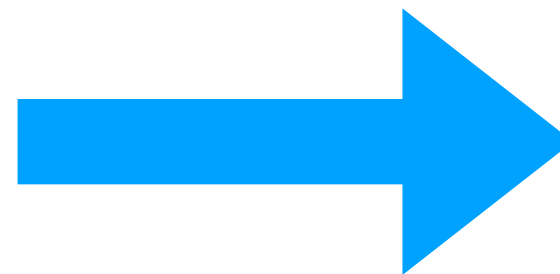


Products

id
name
price

担当者

担当者ID	担当者名
2001	鈴木
3301	田中



Staffs

id
name

物理名の決める時のTips

テーブル名は単数形か複数形か？

Laravelでは複数形が推奨です。

列の名前はとにかくGoogle翻訳しながら作成

自社サービスとかで特殊な固有名の時はローマ字もある
でも基本は翻訳すれば必ずわかる英単語にするのが吉

外部テーブルを参照するキーの名前の決め方は？

参照先テーブル名_id

user_idとか

演習：卒業制作のDB設計にチャレンジ

20分ほどお時間をお渡ししますので設計を書き出してみましよう。

＊書き出したものを4人1グループで見せ合ってフィードバックや見落としが無いかのチェックをしましょう

Step1：データの洗い出し→Step2：関連性の書き出し
→Step3：正規化→Step4：列名の定義

https://qiita.com/koki_73/items/bddafd00ea0f235f25d4