

Chocolates \Rightarrow 10/day

100 pieces \Rightarrow 10 days

Vendor 2 days to deliver

8 \Rightarrow (20 chocolates)
 \downarrow
order again

(10) \Rightarrow 0 - 2 (20 chocolates)

PRD ⇒

Classes

→ attributes ✓
methods ✓

Product

id

name

category

sales price

cost

quantity

is Active

getProfit()

Vendor

name

contactNo.

emailId

street

city

state

~~getContactDetails~~

~~getAddressDetails~~

~~update contact~~

Customer

name

contactNo.

emailId

street

city

state

~~getContactDetails~~

~~getAddressDetails~~

~~update contact~~

~~update Address~~

Order

activate()
deactivate()
isBelowThreshold()
update Address
getCredit()
getDiscount()

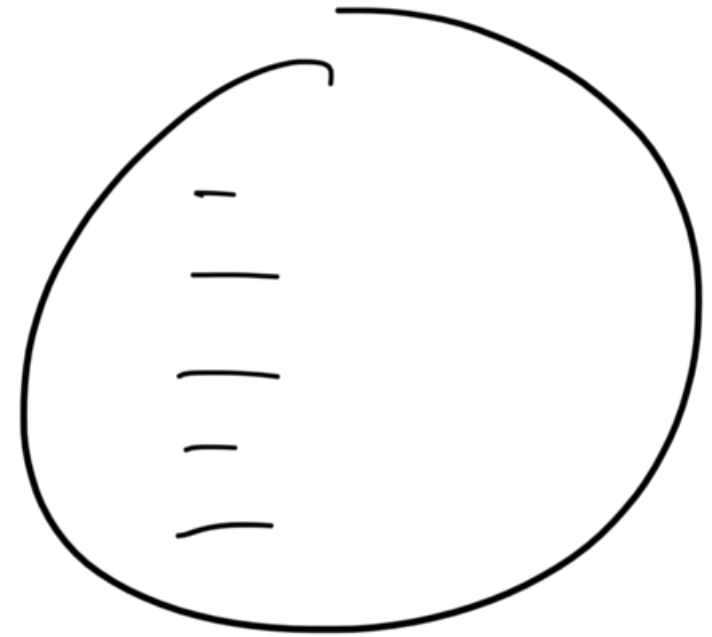
⇒ * Can order only 1 product—
per order

Order → Vendor
→ Product

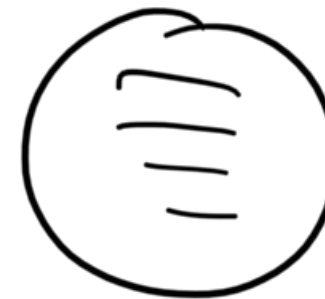
Order object is vendor name ?

↓
Order ?

Vendor



product



Association

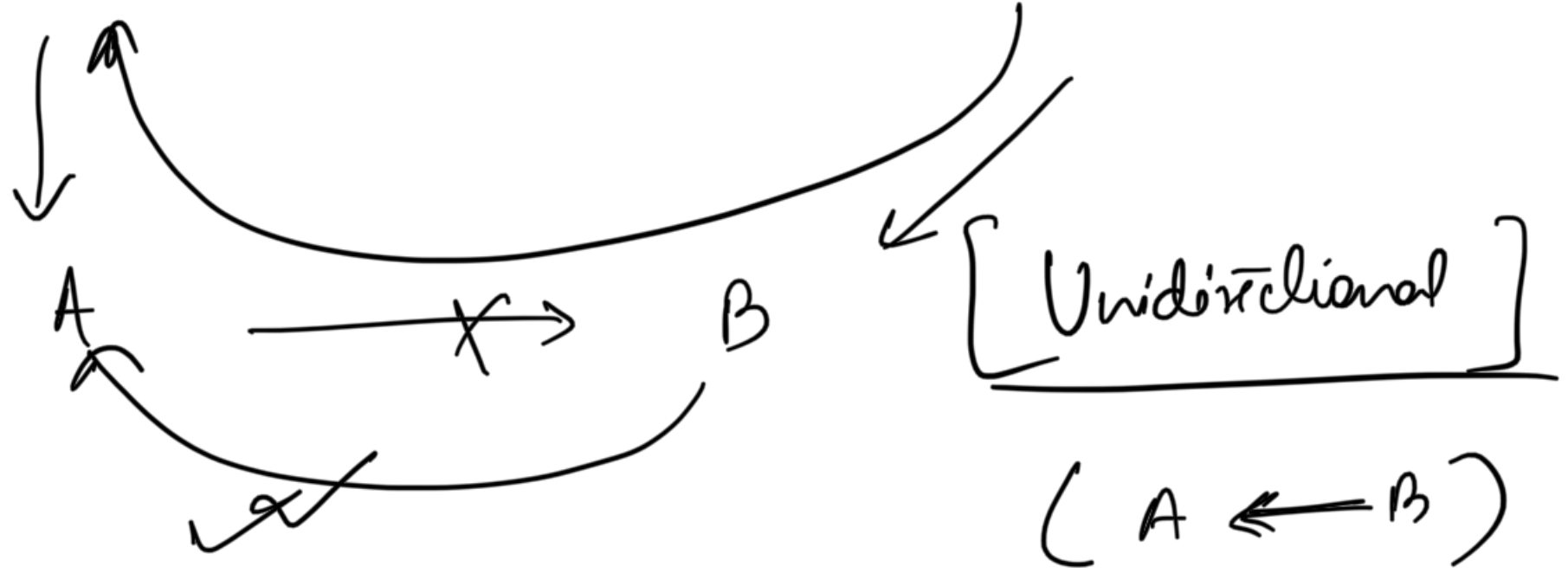
1) Directional

2) Numbers

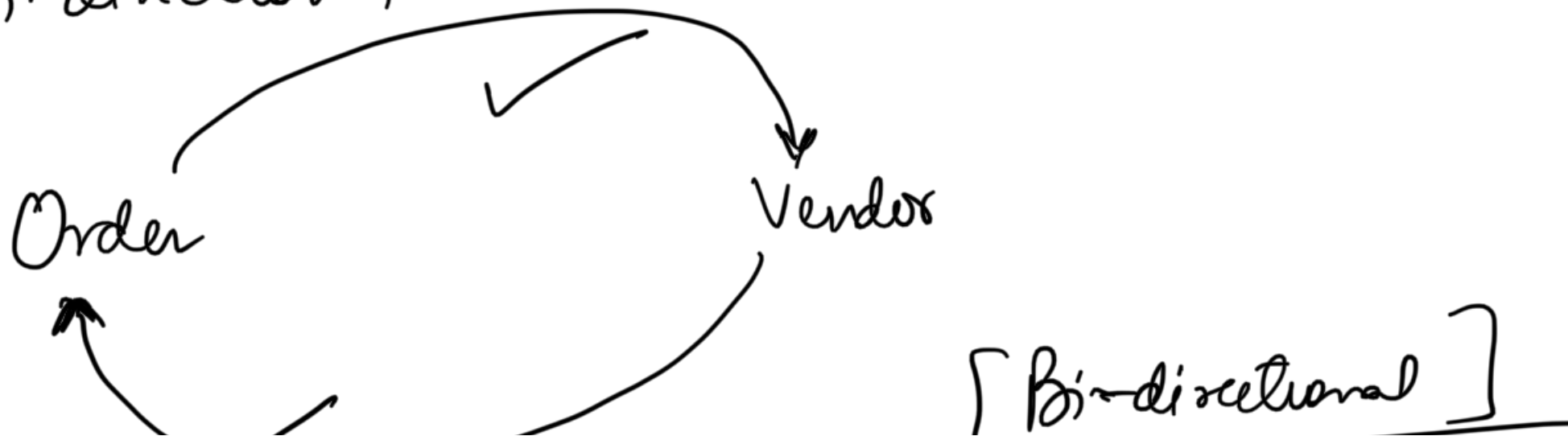
⇒

Directional 1- Unidirection 2 Bi-Direction

Product ~~→~~ Order



Bi-directional



(A - B)

numeric

↓ ∅ ↓

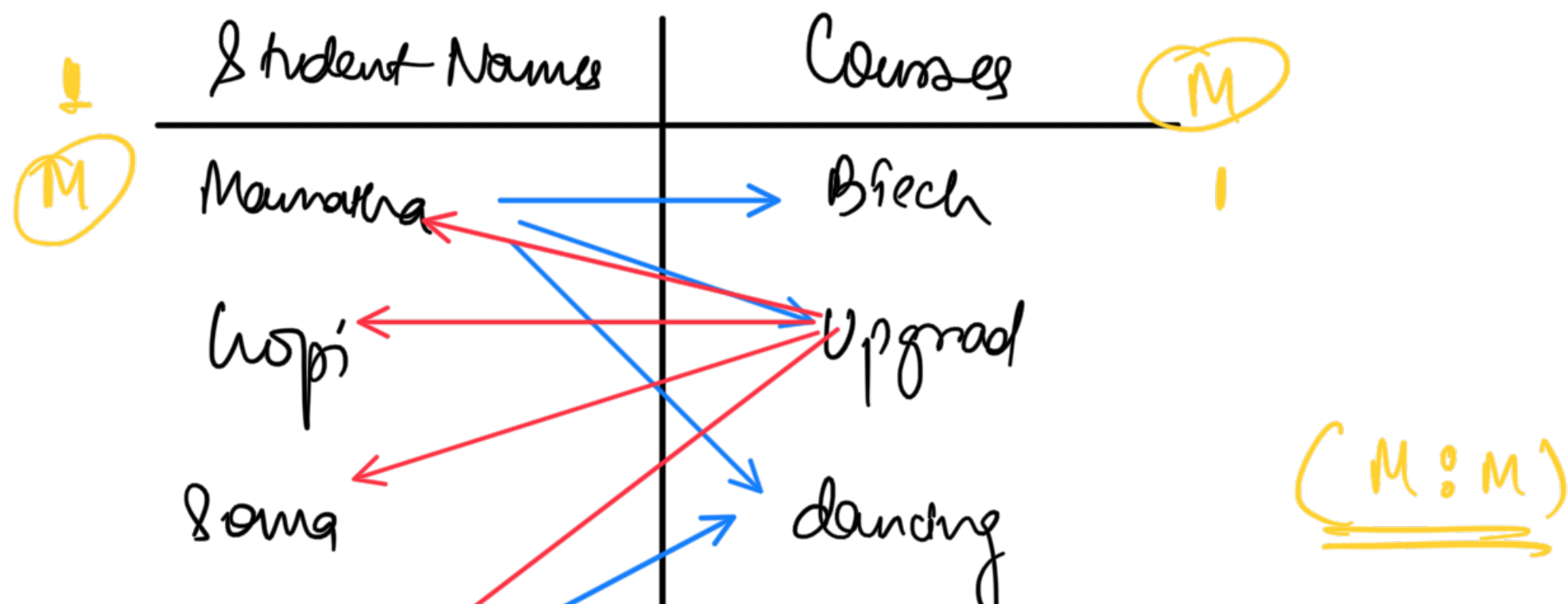
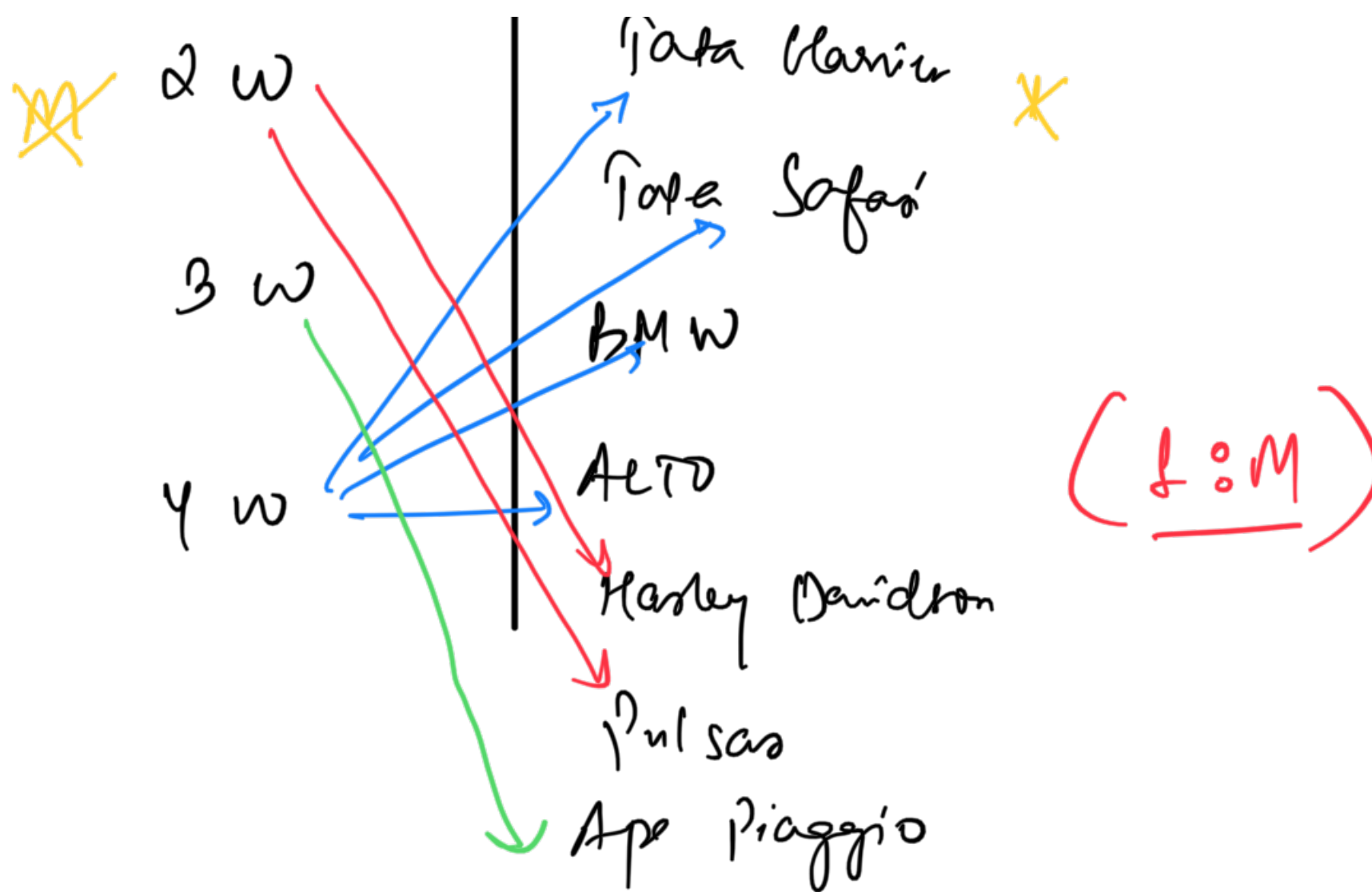
↓ ∅ M

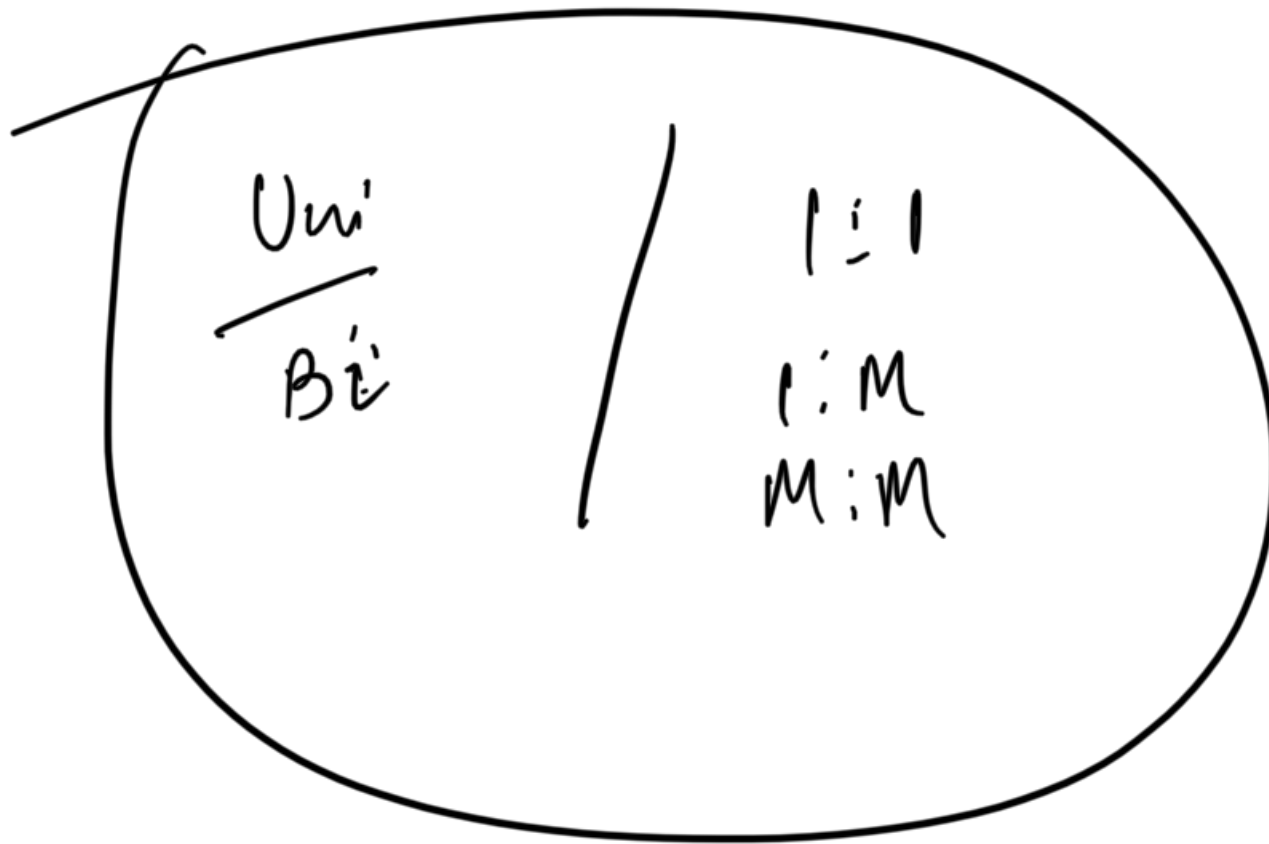
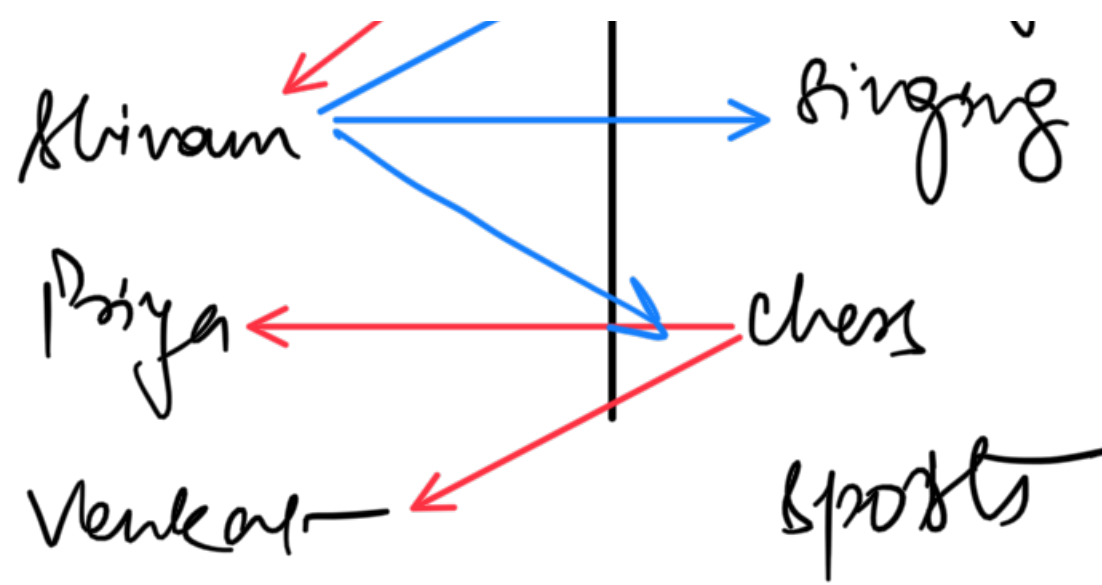
M ∅ M

People		Aadhar Card
Rajat	→	1234
Abdul	→	5678
Shivam	→	91011
Pritya	→	121516

(↓ ∅ ↓)

<u>Vehicle types</u>	Vehicle Name	M
----------------------	--------------	---





Vender
name

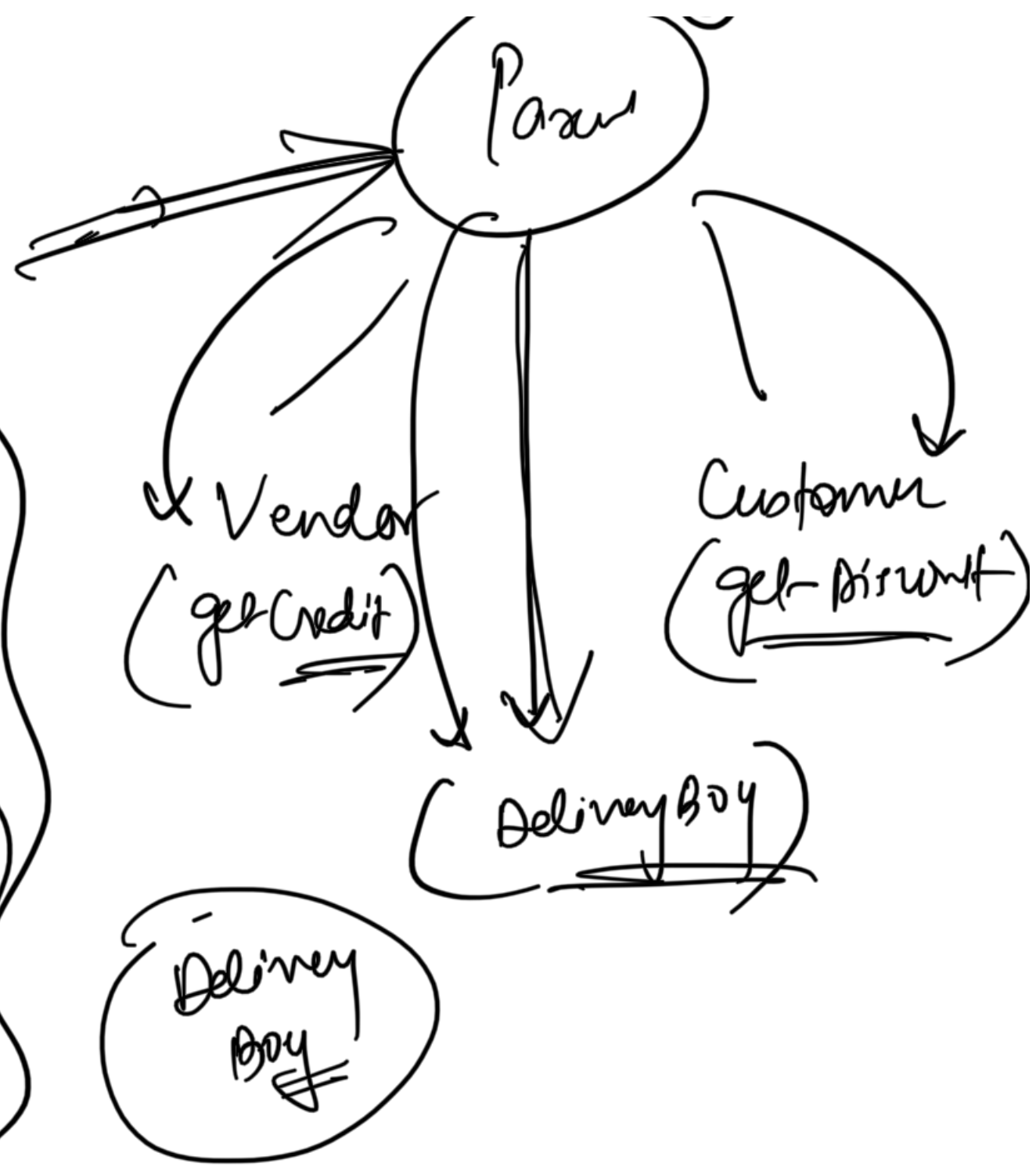
①

- name
 - Contact No.
 - email Id
 - Street
 - city
 - State

- getContactDetails
 - get Address Details
 - update contact
 - update Address
 - getCredit()

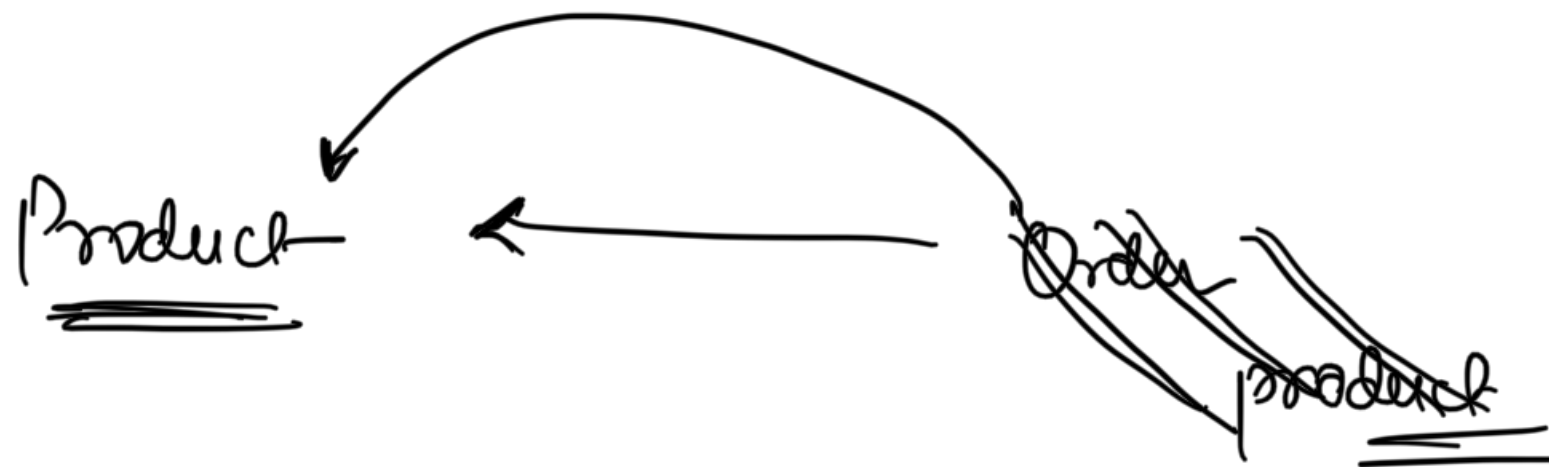
get
 get

aggregation



A uses B

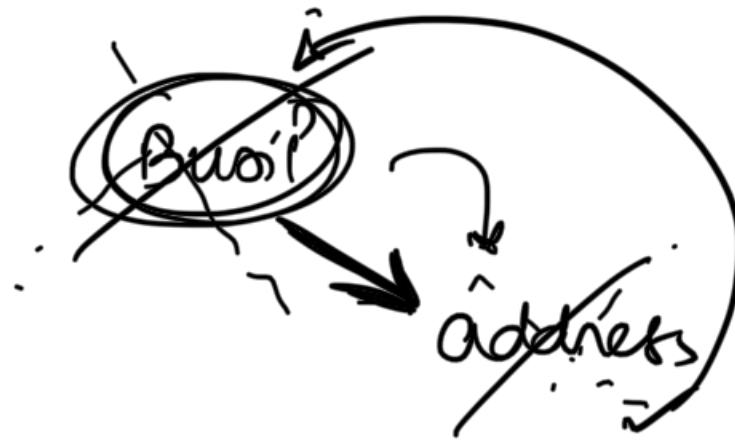
Composition A owns B



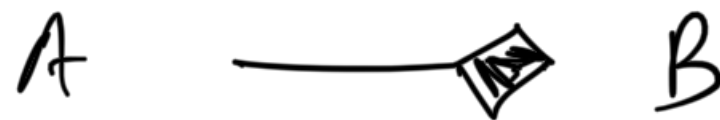
Order uses product

A uses B → Aggregation





A Owns B \Rightarrow Composition



- 1) Read PRD
- 2) Tried to find classes \rightarrow attributes & method

- 3) drew class diagrams
- 4) updated visibility & multiplicity
- 5) Associativity \Rightarrow Un/B: & 1:1 / 1:M / M:M
- 6) Generalization
- 7) Associating \Rightarrow 1:1 \Rightarrow Aggregation & Composition

nd

Draw the complete design diagram

DMS