ESIN

José Ornelas

zeornelas@fe.up.pt

LinkedIn: https://www.linkedin.com/in/josé-pedro-ornelas/

GitHub: https://github.com/zeornelas

Veterinary Clinic
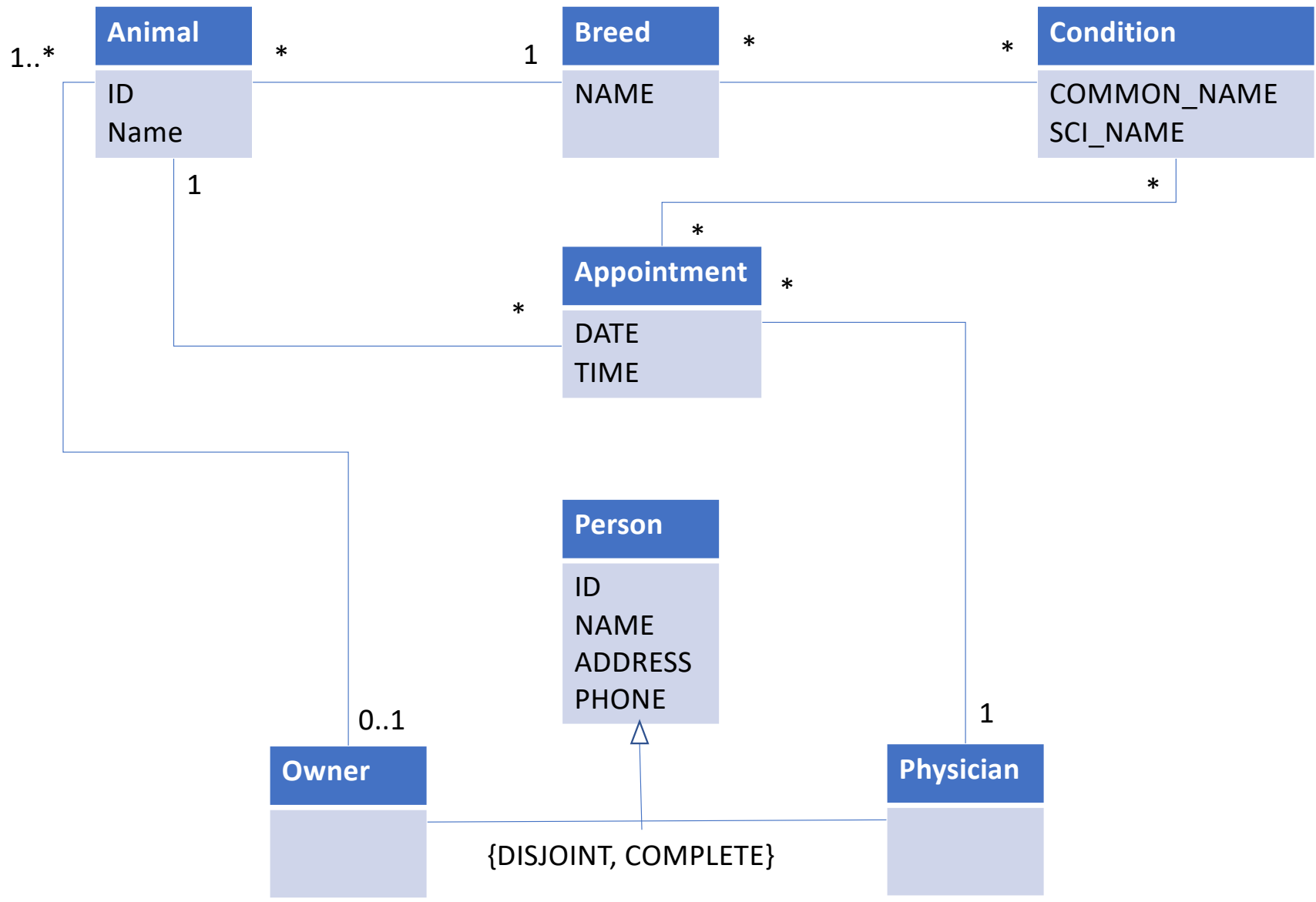
Animal(<u>id</u>, name, breed →Breed, owner→Owner)
NOT NULL(breed, name)
Breed(<u>name</u>)
Condition(<u>c_name</u>, s_name)
NOT NULL(s_name)
UNIQUE(s_name)
Appointment(<u>id</u>, date, time, animal →Animal, physician→Physician)
NOT NULL(animal, physician, date, time)
Owner(<u>id</u>, name, address, phone_no)
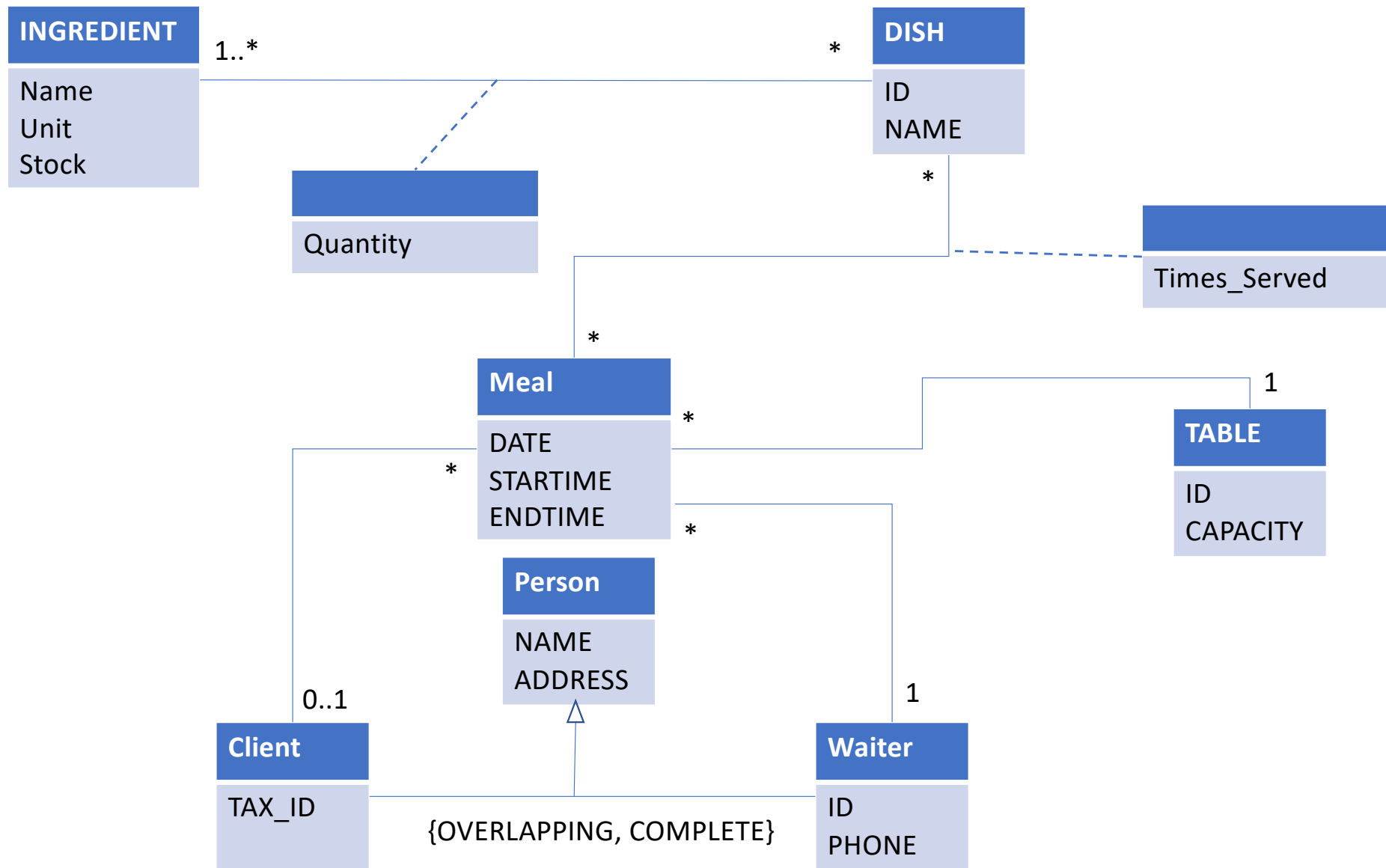NOT NULL(name)
Physician(<u>id</u>, name, address, phone_no)
NOT NULL(name)
ConditionsIdentified(<u>condition</u> → Condition, <u>appointment</u> → Appointment)
FrequentCondition(<u>breed</u>→Breed, <u>condition</u> → Condition)

Restaurant

INGREDIENT
Name
Unit
Stock

1..*

DISH
ID
NAME

*

Quantity

*

Times_Served

*

Meal
DATE
STARTIME
ENDTIME

1

TABLE
ID
CAPACITY

*

*

*

Person
NAME
ADDRESS

*

0..1

Client
TAX_ID

1

Waiter
ID
PHONE

{OVERLAPPING, COMPLETE}

Person(<u>id</u>, name, address)
NOT NULL(name)
Waiter(<u>id</u>→Person, identifier, phone_number)
NOT NULL(identifier, phone_number)
UNIQUE(identifier)
Client(<u>id</u>→Person, tax_id)
NOT NULL(tax_id)
UNIQUE(tax_id)
Meal(<u>id</u>, date, start_time, end_time, served_to →Client, served_by → Waiter, table → Table)
CHECK(start_time < end_time OR end_time IS NULL)
NOT NULL(served_to, served_by, date, start_time)
Dish(<u>identifier</u>, name)
NOT NULL(name)
UNIQUE(name)
CHECK(
 (SELECT COUNT(*)
 FROM QuantitiesNeededForDishes
 JOIN Dishes ON Dishes.identifier = QuantitiesNeededForDishes.dish
 WHERE Dishes.identifier = NEW.identifier
) >= 1
)
Ingredient(<u>name</u>, meas_unit, qtt_stock)
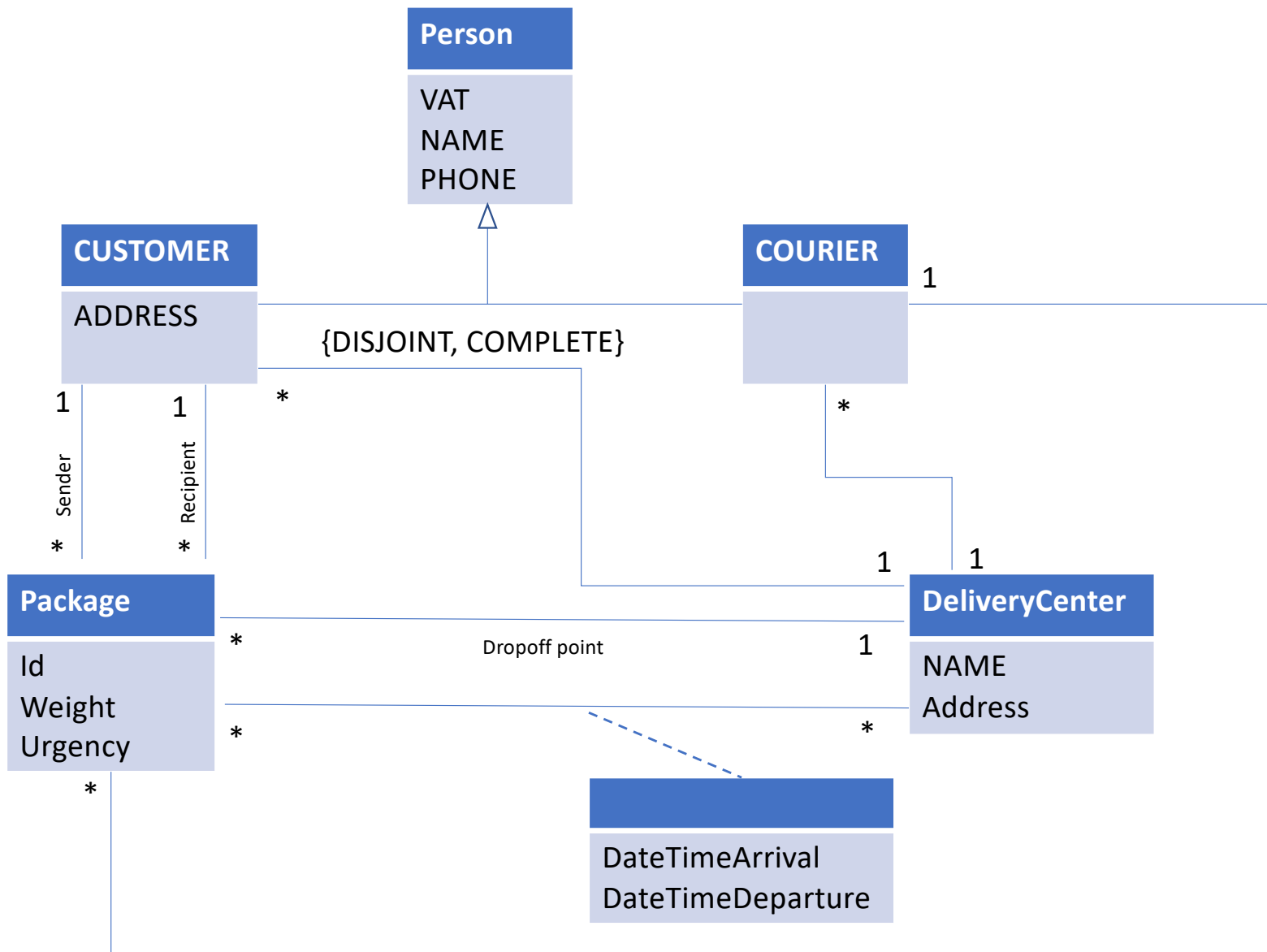NOT NULL(meas_unit, qtt_stock)
CHECK(qtt_stock >= 0)
Table(<u>identifier</u>, capacity)
NOT NULL(capacity)
CHECK(capacity > 0)
QuantitiesNeededForDishes(<u>dish</u>→Dish, <u>ingredient</u> →Ingredient, qtt_needed)
NOT NULL(qtt_needed)
CHECK(qtt_needed > 0)
TimesServed(<u>dish</u>→Dish, <u>meal</u> →Meal, times_served)
NOT NULL(times_served)
CHECK(times_served > 0)

Deliveries

Courier(<u>id</u>, name, vat_no, phone_num, delivery_center->DeliveryCenter)
NOT NULL(name, vat_no, phone_num, delivery_center)
UNIQUE(vat_no)
Customer(<u>id</u>, vat_no,name, phone_num, address, closest_delivery_center → DeliveryCenter)
NOT NULL(name, vat_no, phone_num, address, closest_delivery_center)
UNIQUE(vat_no)
Package(<u>identifier</u>, weight, urgency, dropoff_point →DeliveryCenter, sender →Customer,
recipient →Customer, courier →Courier)
NOT NULL(weight, urgency, dropoff_point, sender, recipient, courier)
CHECK(urgency = "normal" OR urgency ="urgent")
CHECK(weight > 0)
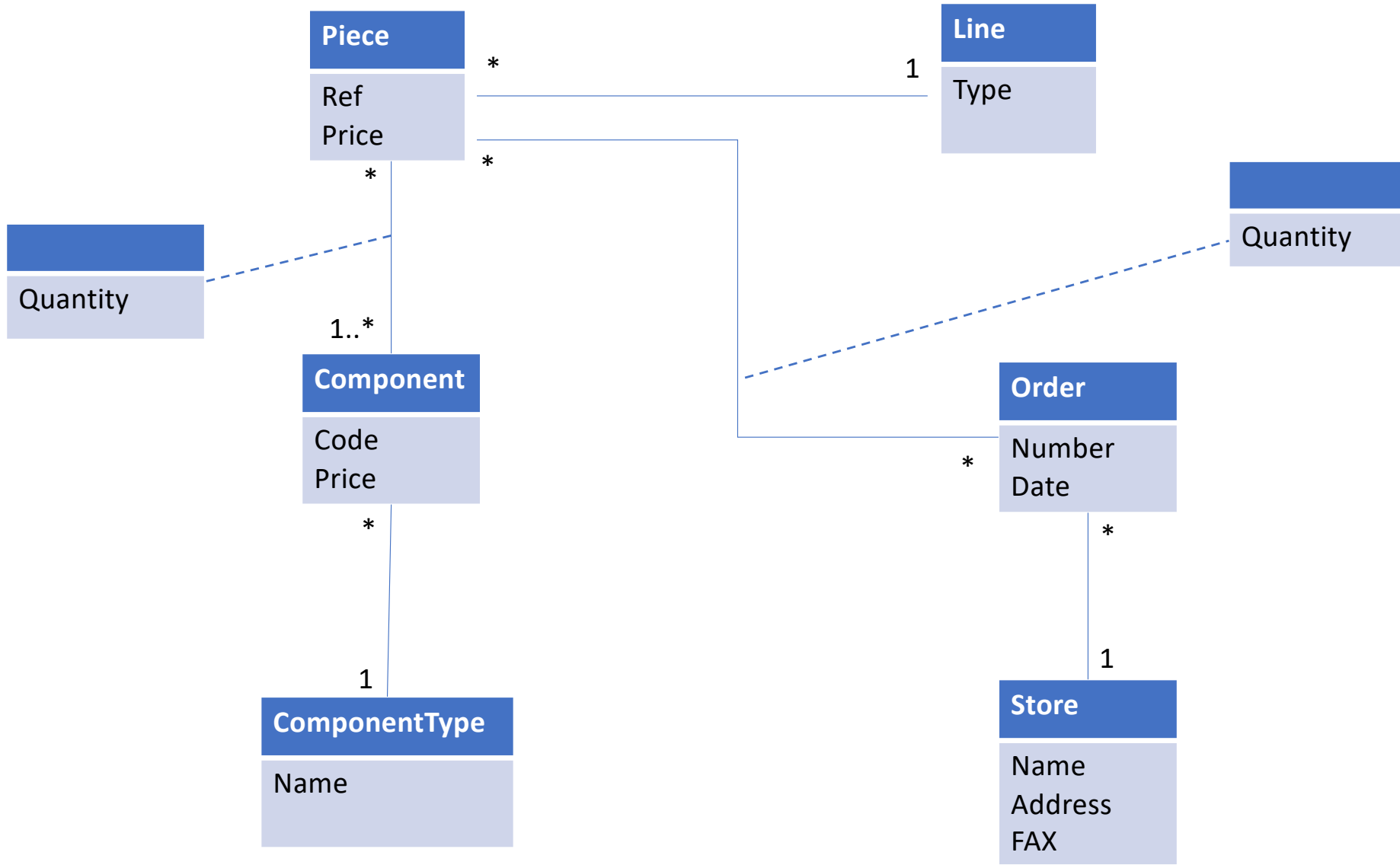**DeliveryCenter**(<u>id</u>, name, address)
NOT NULL (name, address)
UNIQUE(name, address)
InternalTracking(<u>package</u>→Package, <u>delivery_center</u>→DeliveryCenter, date_time_of_arrival,
date_time_of_departure)
CHECK(date_time_of_arrival >= date_time_of_departure)

Furniture Factory

Piece

Ref
Price

Line

Type

Quantity

Component

Code
Price

Order

Number
Date

Quantity

ComponentType

Name

Store

Name
Address
FAX

* 1

* *

* 1..*

* *

* 1

Piece(<u>reference</u>, price, line → Line)
NOT NULL(line, price)
CHECK(price > 0)
Line(<u>id</u>, type)
NOT NULL(type)
UNIQUE(type)
Component(<u>code</u>, price, component_type → ComponentTypes)
NOT NULL(component_type, price)
CHECK(price > 0)
Order(<u>number</u>, date)
NOT NULL(date)
Store(<u>id</u>, name, address, fax_no)
NOT NULL(name, address)
PieceComponent(<u>piece</u> → Piece, <u>component</u> → Component, quantity)
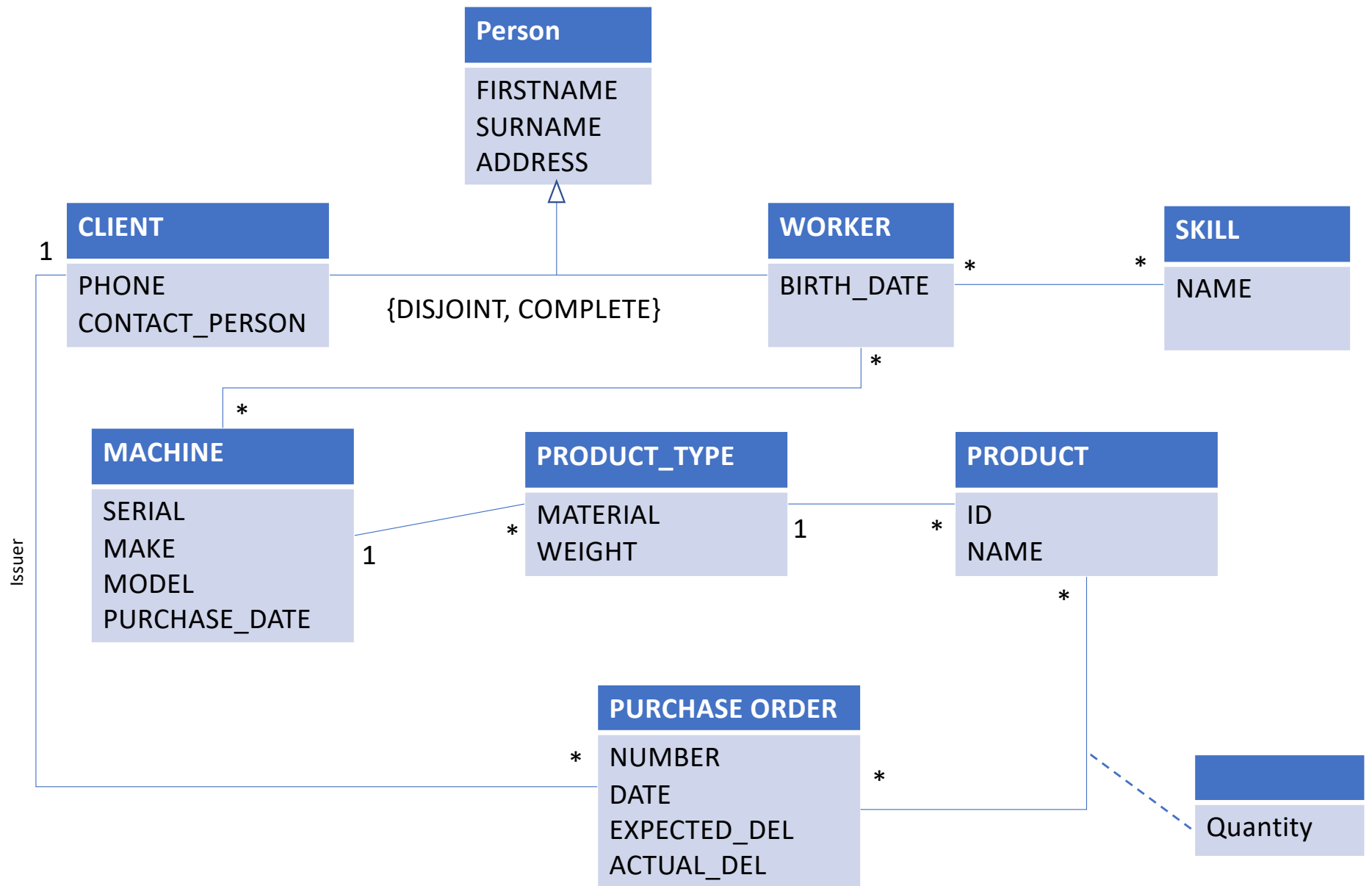NOT NULL(quantity)
CHECK(quantity > 0)
ComponentType(<u>name</u>)
OrderPiece(<u>order</u> → Order, piece → <u>Piece</u>, quantity)
CHECK(quantity > 0)

Factory

Person

FIRSTNAME
SURNAME
ADDRESS

CLIENT

PHONE
CONTACT_PERSON

WORKER

BIRTH_DATE

SKILL

NAME

{DISJOINT, COMPLETE}

1

*   *

*

Issuer

MACHINE

SERIAL
MAKE
MODEL
PURCHASE_DATE

PRODUCT_TYPE

MATERIAL
WEIGHT

PRODUCT

ID
NAME

*

1

1   *

*

PURCHASE ORDER

NUMBER
DATE
EXPECTED_DEL
ACTUAL_DEL

*

*

Quantity

Client(<u>id</u>, first name, surname, address, phone, name_contact)
Worker(<u>id</u>, first name, surname, address, birth_date)
Skill(<u>name</u>)
Machine(<u>serial_no</u>, make, model, purchase_date)
NOT NULL(purchase_date)
Product(<u>identifier</u>, name, product_type → ProductTypes)
NOT NULL(name)
ProductType(<u>id</u>, material, weight, produced_by → Machine)
NOT NULL(material)
CHECK(weight > 0)
PurchaseOrder(<u>number</u>, date_made, expected_delivery, actual_delivery, issuer → Client)
NOT NULL(date_made, expected_delivery, actual_delivery, issuer)
CHECK(date_made <= expected_delivery AND date_made <= actual_delivery)
WorkerSkill(<u>worker</u> → Worker, <u>skill</u> → Skill)
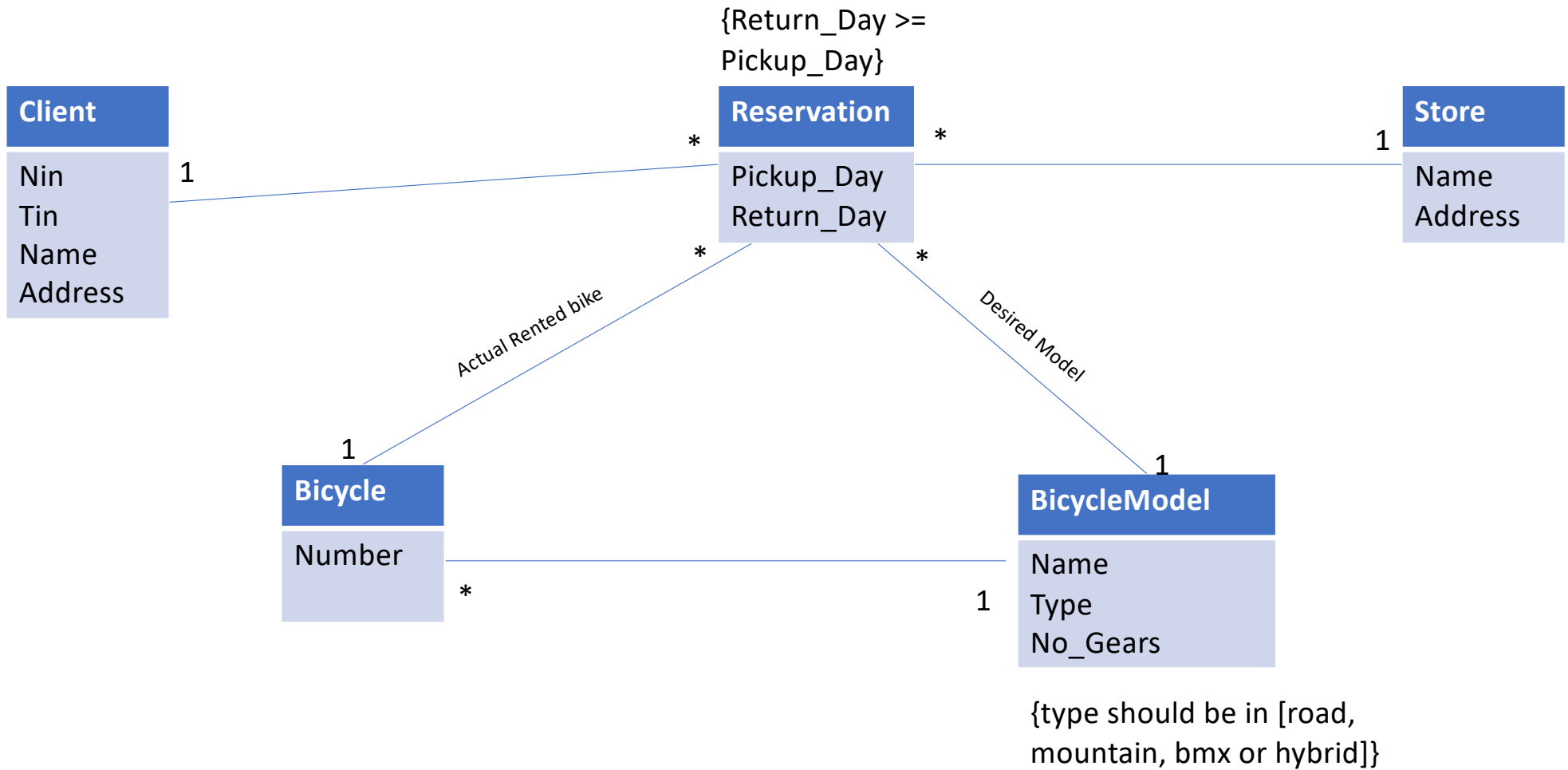**MachineWorker**(<u>worker</u> → Worker, <u>machine</u> → Machine)
OrdersProduct(<u>product</u> → Products, <u>purchase_order</u> → PurchaseOrders, quantity)
CHECK(quantity > 0)

Bicycle Renting

Client

Nin
Tin
Name
Address

1

{Return_Day >=
Pickup_Day}

Reservation

Pickup_Day
Return_Day

*

*

Store

Name
Address

1

*

Actual Rented bike

*

Desired Model

1

Bicycle

Number

1

*

BicycleModel

Name
Type
No_Gears

1

{type should be in [road,
mountain, bmx or hybrid]}

Client(id, NIN, TIN, name, address)
UNIQUE(NIN, TIN)
NOT NULL(name, NIN, TIN)
Reservation(id, pickup_day, return_day, actual_rented_bike → Bicycle, desired_model →
BicycleModel, pickup_place → Store, client → Client)
NOT NULL(pickup_day, actual_rented_bike, desired_model, pickup_place, client)
CHECK(return_day >= pickup_day OR return_day is NULL)
Store(name, address)
NOT NULL(address)
Bicycle(number, model → BicycleModels)
NOT NULL(model)
BicycleModel(name, type, no_gears)
NOT NULL(type, no_gears)
CHECK(no_gears >= 1)
CHECK(type = 'bmx' or type = 'mountain' or type = 'road' or type = 'hybrid')