

Projeto DB – Parte 2
Professor Daniel Faria
Grupo 27 – Turno L19
Esforço 27h

Érik Bianchi – 103580 (7h – 33%)
João Melo – 102843 (7h – 33%)
André Santos – 103597 (7h – 33%)

Department(name)

Employee(ssn, TIN, bdate, name)

UNIQUE(TIN)

IC-1: Every Employee(ssn) must participate in the Works association

Workplace(address, lat, lon)

UNIQUE(lat, lon)

Works(ssn, name, address)

ssn: FK(Employee)

name: FK(Department)

address: FK(Workplace)

Office(address)

address: FK(Workplace)

Warehouse(address)

address: FK(Workplace)

Customer(cust_no, name, email, phone, address)

UNIQUE(email)

Order(order_no, cust_no, date)

cust_no: FK(Customer)

IC-2: Every Order(order_no) must participate in Contains association

Process(ssn, order_no)

ssn: FK(Employee)

order_no: FK(Order)

Sale(order_no)

order_no: Fk(Order)

Pay(order_no, cust_no)

order_no: FK(Sale)

cust_no: FK(Customer) NOT NULL

Product(SKU, description, price)

I(C-3): Every Product(SKU) must exist in entity Supplier

Supplier(TIN, SKU, name, address, date)

SKU: FK(Product)

EANPRODUCT(SKU, ean)

SKU: FK(Product)

Contains(order_no, SKU, qtd)

order_no: FK(Order)

SKU: FK(Product)

Delivery(address, TIN)

address: FK(Warehouse)

TIN: FK(Supplier)

Other Integrity constraints:

IC-4: Customers can only pay for the Sale of an Order they have placed themselves

1.

```
Πname(  
σ price >= 50 ∧ date >= "2023-01-01" ∧ date < "2024-01-01" (  
Orders ⋈ Contains ⋈ Product ⋈ Customer)  
)
```

2.

```
R1 ← ΠEmployee.name→name (  
σ Employee.ssn = works.ssn ∧ works.address=office.address (  
Employee x Office x Works)  
)  
R2 ← ΠEmployee.name→name (  
σ Employee.ssn = works.ssn ∧ works.address=warehouse.address (  
Employee x Warehouse x Works)  
)  
R3 ← Πname (  
σ date >= "2023-01-01" ∧ date < "2023-02-01" (  
Employee ⋈ Process ⋈ Orders)  
)  
(R2 – R1) ∩ R3
```

3.

```
R ← (SkuGsum(qnt)→s (Sale ⋈ Contains ⋈ Product))  
Πname ( Product ⋈ Gmax(s)(R))
```

4.

```
Gsum(qtd*price) (  
Sale ⋈ Contains ⋈ Product  
order_no  
)
```