

# Projeto BD - Parte 2

Grupo 002 — Turno L13 — LEIC-A

Prof. Flávio Martins



**Gonçalo Bárias** (103124) - 33.33% - 12h

**Raquel Braunschweig** (102624) - 33.33% - 12h

**Vasco Paisana** (102533) - 33.33% - 12h

## Modelo Relacional

Renomeámos a relação `order` para `package` na conversão para o modelo relacional, pois em SQL a palavra `order` é uma keyword protegida. Mantivemos a numeração das *Integrity Constraints* consoante as fornecidas no enunciado, iniciando a contagem a 6 para as *ICs* adicionadas. A única *IC* que não foi passível de converter para o modelo relacional foi a **(IC-1)**, sendo todas as outras convertidas através da propriedade **UNIQUE**.

`customer(cust_no, name, email, phone, address)`

- **UNIQUE(email)**

`package(package_no, date, cust_no)`

- `cust_no`: FK(customer) NOT NULL
- **(IC-6)**: any `package_no` in `package` must exist in `contains`

`sale(package_no)`

- `package_no`: FK(package)

`pay(package_no, cust_no)`

- `package_no`: FK(sale)
- `cust_no`: FK(customer) NOT NULL
- **(IC-1)**: `cust_no` must exist in the package identified by `package_no`

`product(sku, name, description, price)`

- **(IC-7)**: any `sku` in `product` must exist in `supplier`
- **(IC-8)**: when a product is removed from the database it must also be removed from `ean_product` if present

`ean_product(sku, ean)`

- `sku`: FK(product)

`contains(package_no, sku, qty)`

- `package_no`: FK(package)
- `sku`: FK(product)

`supplier(tin, name, address, sku, supply_contract_date)`

- `sku`: FK(product)

`department(name)`

`workplace(address, lat, long)`

- **UNIQUE(lat, long)**
- **(IC-9)**: when a workplace is removed from the database it must also be removed from `warehouse` and/or `office` if present

`warehouse(address)`

- `address`: FK(workplace)

`delivery(address, tin)`

- `address`: FK(warehouse)
- `tin`: FK(supplier)

`office(address)`

- `address`: FK(workplace)

`employee(ssn, tin, b_date, name)`

- **UNIQUE(tin)**
- **(IC-10)**: any `ssn` in `employee` must exist in `works`

`works(ssn, name, address)`

- `ssn`: FK(employee)
- `name`: FK(department)
- `address`: FK(workplace)

`process(ssn, package_no)`

- `ssn`: FK(employee)
- `package_no`: FK(package)

# Álgebra Relacional

1. **Liste o nome de todos os clientes que fizeram encomendas contendo produtos de preço superior a 50€ no ano de 2023.**

$$C \leftarrow \sigma_{\text{date} \geq '2023/01/01' \wedge \text{date} \leq '2023/12/31'} (\text{costumer} \bowtie \text{package}) \bowtie \text{contains} \\ \Pi_{\text{costumer.name}} (\sigma_{\text{price} > 50} (C \bowtie_{\text{contains.sku} = \text{product.sku}} \text{product}))$$

2. **Liste o nome de todos os empregados que trabalham em armazéns e não em escritórios e processaram encomendas em Janeiro de 2023.**

$$e \leftarrow \sigma_{\text{date} \geq '2023/01/01' \wedge \text{date} \leq '2023/01/31'} (\text{employee} \bowtie \text{process} \bowtie \text{package}) \bowtie_{\text{employee.ssn} = \text{works.ssn}} \text{works} \\ \Pi_{\text{employee.name}} ((e \bowtie \text{warehouse}) - (e \bowtie \text{office}))$$

3. **Indique o nome do produto mais vendido.**

Em caso de empate do produto mais vendido decidimos apresentar todos esses produtos.

$$p \leftarrow \text{sku } G_{\text{sum}(\text{qty}) \mapsto \text{p\_qty}} (\text{product} \bowtie \text{contains} \bowtie \text{sale}) \\ \Pi_{\text{name}} (G_{\text{max}(\text{p\_qty}) \mapsto \text{p\_qty}} (p) \bowtie p \bowtie \text{product})$$

4. **Indique o valor total de cada venda realizada.**

Apresentámos com o valor total o correspondente package\_no da venda, pois para casos em que há duas vendas com o mesmo valor total não seria possível distinguir esses valores.

$$\text{package\_no } G_{\text{sum}(\text{price} * \text{qty}) \mapsto \text{total\_val}} (\Pi_{\text{package\_no}, \text{sku}, \text{price} * \text{qty}} (\text{sale} \bowtie \text{contains} \bowtie \text{product}))$$