

# **Projeto de Base de Dados**

## **Parte 2**

**2º Semestre – 2021/ 2022**

**Grupo: 49**

**Turno: BDL06**

**Docente: Pedro Miguel Leão Veloso Dias**

<b>Aluno</b>	<b>Esforço</b>
André Torres - 99053	7h 33% de contribuição
Gonçalo Nunes - 99074	7h 33% de contribuição
Stefan Knutsen - 99123	7h 33% de contribuição

## Tradução para o Modelo Relacional

Point\_of\_Retail(address, name)

IVM(serial\_number, manuf)

Retailer(TIN, name)

- UNIQUE(name)

Category(name)

- RI-1: O nome deve existir em Simple\_Category e/ou Super\_Category.
- RI-2: Nenhuma categoria pode existir ao mesmo tempo em Simple\_Category e Super\_Category.

Simple\_Category(name)

- name: **FK**(Category.name)

Super\_Category(name)

- name: **FK**(Category.name)

Product(ean, descr)

Shelve(serial\_number, manuf, nr, height, category\_name)

- serial\_number, manuf: **FK**(IVM.serial\_number, IVM.manuf)
- category\_name: **FK**(Category.name)
- RI-1: O nr deve existir em Ambient\_Temp\_Shelf e/ou Warm\_Shelf e/ou Cold\_Shelf.
- RI-2: Nenhuma categoria pode existir ao mesmo tempo em Ambient\_Temp\_Shelf e/ou Warm\_Shelf e/ou Cold\_Shelf.
- RI-3: Um Product só pode ser reposto (Replenished) numa Shelve onde sua Category seja apresentada.

Ambient\_Temp\_Shelf(ean)

- ean: **FK**(Shelve)

Warm\_Shelf(ean)

- ean: **FK**(Shelve)

Cold\_Shelf(ean)

- ean: **FK**(Shelve)

Replenishment\_Event(ean, nr, instant, units, TIN)

- ean, nr: **FK**(Replenishment.ean, Replenishment.nr)
- TIN: **FK**(Retailer)

installed-at(serial\_number, manuf, address, nr)

- address: **FK**(Point\_of\_Retail)
- serial\_number: **FK**(IVM)

responsible-for(TIN, serial\_number, manuf, category\_name)

- TIN: **FK**(Retailer)
- serial\_number: **FK**(IVM)
- manuf: **FK**(IVM)
- category\_name: **FK**(Category.name)

has-other(super\_name, sub\_name)

- super\_name: **FK**(Super\_Category.name)
- sub\_name: **FK**(Category.name)
- RI-1: Uma categoria não pode estar contida em si própria.
- RI-2: Não podem existir ciclos nas hierarquias de Categorias.
- RI-3: Todas as Super Categorias têm de participar na associação “has-other”.

has(name, ean)

- name: **FK**(Category)
- ean: **FK**(Product)
- RI-1: Todos os Produtos têm de participar na associação “has”.

planogram(ean, nr, faces, units, loc)

- ean: **FK**(Product)
- nr: **FK**(Shelve)
- RI-1: O número de unidades repostas num Replenishment\_Event não pode exceder o número de unidades ‘nr’ especificado no ‘Planogram’.

replenishment(ean, nr, instant)

- ean, nr: **FK**(Planogram.ean, Planogram.nr)
- instant: **FK**(Replenishment\_Event)
- RI-1: Um Product só pode ser reposto (Replenished) numa Shelve onde sua Category seja apresentada.
- RI-2: Um Product só pode ser reposto (Replenished) pelo Point\_of\_Retail responsável pela Category do Product.

## Algebra Relacional

1.  $\pi_{ean, descr} \sigma_{name = "Barras Energéticas" \wedge units > 10 \wedge instant > "2021-12-31"} (product \bowtie has \bowtie category \bowtie replenishment\_event)$
2.  $\pi_{ivm.serial\_number} \sigma_{product.ean = 9002490100070} (ivm \bowtie shelve \bowtie category \bowtie has \bowtie product)$
3.  $\pi_{COUNT(*)} \sigma_{super\_name = "Sopas Take-Away"} (has\_other)$

4.  $\text{Event} \leftarrow \pi_{ean, descr}(\text{replenishment\_event} \bowtie \text{Product})$   
 $\text{Grouped\_Event} \leftarrow \pi_{ean, descr} \text{G}_{SUM(units) \rightarrow total}(\text{Event})$   
 $\text{Result} \leftarrow \pi_{ean, descr} \text{G}_{MAX(total)}(\text{Grouped\_Events}) \bowtie \text{Grouped\_Events}$

## SQL

1. 

```
SELECT product.ean, product.descr
FROM product
      JOIN has ON product.ean = category.ean
      JOIN category ON has.name = category.name
      JOIN replenishment ON product.ean = replenishment.ean
WHERE category.name = 'Barras Energéticas' AND replenishment.nr > 10 AND
      replenishment.instant > '2021-12-31';
```
2. 

```
SELECT ivm.serial_number
FROM ivm
      JOIN shelve ON ivm.serial_number = shelve.serial_number AND ivm.manuf =
      shelve.manuf
      JOIN category ON shelve.category_name = category.name
      JOIN has ON category.name = has.name
      JOIN product ON has.ean = product.ean
WHERE product.ean = 9002490100070;
```
3. 

```
SELECT COUNT(*)
FROM has_other
WHERE super_name = 'Sopas Take-Away';
```
4. 

```
SELECT product.ean, product.descr
FROM product
      JOIN replenishment_event ON product.ean = replenishment_event.ean
HAVING SUM(replenishment_event.units) >= ALL(
      SELECT SUM(units)
      FROM replenishment_event
      GROUP BY replenishment_event.ean);
```