Graphical user interface, logo

Description automatically generated

**Projeto de Base de Dados**

Parte 2

2º Semestre – 2021/ 2022

**Grupo**: 49

**Turno**: BDL06

**Docente**: Pedro Miguel Leão Veloso Dias

|  |  |
| --- | --- |
| **Aluno** | **Esforço** |
| 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 name 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)

installled-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. πean, descr σ name = "Barras Energéticas" AND units > 10 AND instant > "2021-12-31"(product ⋈ has ⋈category ⋈ replenishment\_event)
2. πivm.serial\_number σproduct.ean = 9002490100070(ivm ⋈ shelve ⋈category ⋈ has ⋈ product)
3. πCOUNT (\*) σsuper\_name = "Sopas Take-Away"(has\_other)
4. Event <- πean, descr (replenishment\_event ⋈ Product)

Grouped\_Event <- *ean, descr* G*SUM(units)->total*(Event)

Result <- *ean, descr* G*MAX(total)*( Grouped\_Events) ⋈ 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’;

1. SELECT ivm.serial\_number

FROM ivm

JOIN shelve ON ivm.serial\_number = shelve.seriel\_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;

1. SELECT COUNT(\*)

FROM has\_other

WHERE super\_name = ‘Sopas Take-Away’;

1. 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);