# Laborator 3

## Task 1 (2p)

Scrieti modelul corespunzator urmatorului cod SQL:

CREATE TYPE categ\_prajitura AS ENUM( 'comanda speciala', 'aniversara', 'editie limitata', 'pentru copii', 'dietetica','comuna');

CREATE TYPE tipuri\_produse AS ENUM('cofetarie', 'patiserie', 'gelaterie');

CREATE TABLE IF NOT EXISTS prajituri (

id serial PRIMARY KEY,

nume VARCHAR(50) UNIQUE NOT NULL,

descriere TEXT,

pret NUMERIC(8,2) NOT NULL,

gramaj INT NOT NULL,

tip\_produs tipuri\_produse DEFAULT 'cofetarie',

calorii INT NOT NULL,

categorie categ\_prajitura DEFAULT 'comuna',

pt\_diabetici BOOLEAN NOT NULL DEFAULT FALSE,

imagine VARCHAR(300),

data\_adaugare TIMESTAMP DEFAULT current\_timestamp

);

Creati un model pentru entitatea Ingredient cu proprietatile:

* nume - string unic de maxim 30 caractere
* calorii - intreg pozitiv
* unitate - string de maxim 10 caracatere

Realizati o legatura **many-to-many** intre Ingredient si Prajitura

Creati un model pentru entitatea Ambalaj cu proprietatile:

* nume - string unic de maxim 20 caractere
* material string cu optiunile: plastic, hartie, carton
* pret - numar float de 5 cifre din care doua zecimale

Realizati o legatura **one-to-many** intre Ambalaj si Prajitura (o prajitura are un singur tip de ambalaj, dar un ambalaj poate corespunde mai multor prajituri)

Inserati 3 prajituri.

Realizati o pagina numita prajituri, in care afisati o lista cu datele tuturor prajiturilor din baza de date.

## Task 2(2p) PROIECT!

Stabiliti intre 5 si 7 entitati pentru baza voastra de date **(pentru proiect) doar legate de produsele vandute (nu si de utilizatori)**. Trebuie sa aveti minim o relatie one-to-many si o relatie many-to-many. Trebuie minim 6 tipuri de campuri diferite folosite in cadrul tabelelor (nu neaparat toate in acelasi tabel). De exemplu CharField, TextField

## Task 3(fiecare model 0.5p) PROIECT!

Realizati modelele corespunzatoare entitatilor din diagrama, dupa ce vi s-a confirmat ca sunt corecte.

## Anexe

Exemplu views.py

from django.shortcuts import render

from .models import Locatie

def afis\_template(request):

return render(request,"aplicatie\_exemplu/articole.html",

{

"locatii":Locatie.objects.all()

}

)

Exemplu template:

<!DOCTYPE html>

<html lang="ro">

<head>

<title>Locatii</title>

</head>

<body>

{% for locatie in locatii %}

<p>{{ locatie.adresa }}</p>

{% endfor %}

</body>

</html>

class Prajitura(models.Model):

nume = models.CharField(max\_length=20, unique=True)

descriere = models.TextField(null=True)

pret = models.DecimalField(max\_digits=8, decimal\_places=2)

gramaj = models.PositiveIntegerField()

temperatura= models.IntegerField()

#completati voi ......