Universitatea din București

Facultatea de Matematică și Informatică

Departamentul Tehnologia Informației

**PROIECT LA BAZE DE DATE**

Coordonator stintific:

Vasile Silviu Laurentiu

Student:

Răducu Constantin Daniel

București, 2020

Universitatea din București

Facultatea de Matematică și Informatică

Departamentul Tehnologia Informației

**PIZZADANIEL**

Coordonator stiintific:

Vasile Silviu Laurentiu

Student:

Răducu Constantin Daniel

București, 2020

* **MODELUL SI REGULILE ACESTUIA**

*Model:*

O pizzerie, dar nu ca oricare alta. PizzaDaniel, e cea mai nouă pizzerie din oraș, aduce cu ea tipuri de pizza din toată lumea, încercând în același timp să lase preturile la nivel de student. Din păcate sau din fericire, momentan, e doar la nivel de pizza cu livrare. Cum funcționează? Intrii în aplicație, îți faci un cont și te loghezi cât ai zice “Pizza”. Vei fi trimis într-o pagină de început de unde poți să te delectezi uitandu-te la nenumăratele pizza cu fel și fel și fel de ingrediente. Niciuna nu-i la fel, toate sunt unice în felul lor, exact ca tine . Îți poți alege câte pizza vrei, când ești pregătit, poți să finalizezi comanda, iar unul din cei 5 șoferi argentinieni vor fi trimis imediat la locația ta.

*Exemplu din viața reală:*

Să spunem că e prima dată când folosiți aceasta aplicație. Primul pas este să vă faceți un cont, apoi vă logati cu acel cont înregistrat. Intrați în meniu și vă selectați câte pizza vreți să comandați. Când sunteți gata vă duceți la coș, și acolo puteți finaliza comanda dacă totul este în regulă. Apoi veți putea să urmăriți comanda în contul dumneavoastră la comenzile mele. Până când aceasta ajunge la dumneavoastră.

Având în vedere cele menționate mai sus, baza de date cuprinde 8 tabele:

* Locații: unde stocăm toate adresele utilizatorilor
* Utilizatori: utilizatorii care pot să facă comenzi, și să-și aleage la ce locație să vină comanda
* Comenzi: toate comenziile sunt făcute de utilizatori și se înregistrează aici
* Livratori: informații despre livratorii care duc comenzile
* IstoricComenzi: informații despre comenzi, ce s-a comandat, cantitatea și numărul comenzii
* Pizza: toate tipurile de pizza care sunt disponibile pentru a putea fi comandate
* IngredientPizza: aici se înregistrează toate ingredientele pe fiecare pizza în parte
* Ingrediente: ingredientele posibile pe anumite pizza

**Reguli**:

Un utilizator poate să aibă mai multe locații și o locație poate să aibă doar un utilizator. Tot un utilizator poate să facă mai multe comenzi, dar o comandă poate să fie făcută doar de un utilizator. O comandă poate să aibă mai multe tipuri de pizza și poate să fie livrată doar de un livrator, iar un tip de pizza poate să fie găsit pe mai multe comenzi. Un livrator poate să-și aleage ce comenzi să ducă. Un fel de pizza poate să aibă mai multe tipuri de ingrediente, dar și un ingredient poate să fie găsit pe mai multe tipuri de pizza. La crearea unui utilizator, utilizatorul e obligat să-și facă și o locație, dar ulterior poate să o modifice sau să adauge mai multe locații

* **a. Diagrama entitate-relație**



**b. Descrierea entitaților, atributetelor, cheilor, relațiilor si a cardinalităților.**

* Utilizatori: acest tabel se referă la clienți. Un client dacă dorește să dea comandă la această pizzerie trebuie mai întăi să-și facă un utilizator.

Atributele acestei entități sunt:

* #Utilizator – PK; cheie primară pentru tabelul Utilizatori, sub forma unui nume, acesta la rândul lui acționează ca și cheia externă pentru Locații și Comenzi; tip varchar(20);
* Parola; un câmp care se folosește pentru logarea la contul utilizatorului; tip varchar(20);
* Nume; pentru a putea lua legătură într-un mod mai formal; tip varchar(20);
* Prenume: este un câmp opțional doar oferă mai multe informații despre client;tip varchar(20);
* Email: este folosit pentru a putea trimite mai ușor utilizatorului informații, oferte și update-uri;tip varchar(20);
* Telefon: pentru a putea lua legătură cu clientul în momentul livrării comenzii;tip varchar(10);
* Locații: este folosit pentru a putea livra comanda la utilizator. Fară acest tabel nu am știi unde să livrăm comanda.

Atributele acestei entități sunt:

* #Locație – PK; cheie primară pentru tabelul Locații, este folosit pentru a putea indentifica mai ușor locațiile clienților; tip de date int;
* Cartier; în caz de sunt mai multe străzi cu același nume în același oraș, linia “Cartier” ne ajută în a le putea diferenția; tip de date varchar(20);
* Strada; strada pe care se află clientul, îl ajută în mod special pe Livrator în aș-i gestiona comenzile; tip de date varchar(20);
* Adresa: oferă informații în plus pentru a-l găsii pe client mai ușor
* Utilizator – FK; cheie externă pentru Utilizator, ne ajută să indentificam locațiile unui anumit client; tip de date varchar(20);
* Comenzi: acest tabel are rolul de a da permisiune utilizatorului de a face comenzi, cât și livratorului în a știi ce comenzi să livreze;

Atributele acestei entități sunt:

* Comanda\_ID – PK; acționează că și cheie primară pentru tabelul comenzi, ajută în a gestiona mai ușor comenzile; tip de date int;
* Utilizator – FK; oferă informația: cine a comandat această comanda; tip de date varchar(20);
* Cost; prețul total al comenzii; tip de date int; Livrator\_ID – FK; oferă informația: cine a livrat această comandă; tip de date int;
* Data; când s-a înregistrat această comandă; tip de date datetime;
* Status; oferă informația: statusul comenzii: nelivrat, în curs de livrare, livrat; tip de date varchar(20);
* Livrator: oferă informații despre toți livratorii din pizzerie;

Atributele acestei entități sunt:

* Livrator\_ID – PK; cheie prima pentru tabelul Livratori, gestionează mai bine livratorii; tip de date int;
* Nume; trebuie să aibă un nume, doar nu o să spunem la comandă: livratorul1 tocmai ți-a adus comanda; tip de date varchar(20);
* Data\_angajarii; când a fost angajat acest livrator; tip datetime;
* Salariu; ce salariu primește fiecare livrator în LEI; tip de date int;
* IstoricComenzi: acest tabel este asociativ și arată ce pizza au fost comandate și în ce cantitate;

Atributele acestei entități sunt:

* Comanda\_ID – PK, FK1; cheie primară și cheie externă pentru stabilirea legăturii cu tabelul Comenzi; tip de date int;
* Pizza\_ID – PK,FK2; cheie primară și cheie externă pentru stabilirea legăturii cu tabelul pizza; tip de date int;
* Pizza\_cantitate; în ce cantitate a fost comandată un tip de pizza; tip de date int;
* Pizza: acest tabel se folosește pentru a putea da informațiile generale pe fiecare pizza, mai puțin fiecare ingredient. Ingredientele se găsesc în tabelul IngredientPizza;

Atributele acestei entități sunt:

* Pizza\_ID – PK; cheie primară pentru tabelul Pizza, oferă o gestionare mai ușoară a acestui tabel; tip de date int;
* Nume; orice pizza are un nume, aceeași poveste că și la livrator; tip de date varchar(20);
* Preț; doar nu e și gratis, o pizza trebuie să aibe și un preț atunci când e comandată; tip de date int;
* Origine; informație bonus; poate cineva vrea să-și aleagă o pizza specială dintr-o anumită tară; tip de date varchar(20);
* IngredientPizza: tabel asociativ, combină tabelele Pizza și Ingredient pentru a putea spune mai precis ce ingredient are fiecare pizza;

Atributele acestei entități sunt:

* Pizza\_ID – PK,FK1; cheie primară și cheie externă pentru stabilirea legăturii cu tabelul Pizza; tip de date int;
* Ingredient\_ID – PK,FK2; cheie primară și cheie externă pentru stabilirea legăturii cu tabelul Ingredient; tip de date int;
* Ingrediente: tabel care prezintă toate ingredientele posibile în această pizzerie;

Atributele acestei entități sunt:

* Ingredient – PK; cheie primară pentru tabelul Ingrediente; tip de date int;
* Nume; Pentru a putea spune mai precis ce este acest ingredient; tip de date varchar(20);

***Descrierea cardinalităților:***

* **Utilizatori – Locații**
* Un utilizator poate sa aiba 1 sau mai multe locații;
* O locație aparține doar unui singur utilizator;
* **Utilizatori – Comenzi**
* Un utilizator poate să facă 0 sau mai multe comenzi;
* O comandă poate să fie facută de doar de 1 utilizator;
* **Comenzi – Livrator**
* O comandă poate fi dusă doar de 1 livrator;
* Un livrator poate sa ducă 1 sau mai multe comenzi ( alegerea lui );
* **Comenzi – Pizza**
* O comandă poate sa aibe 1 sau mai multe tipuri de pizza;
* Un fel de pizza poate să fie pe 1 sau mai multe comenzi;
* **Pizza – Ingredient**
* O pizza poate să conțină 1 sau mai multe ingrediente;
* Un ingredient poate să fie găsit pe 1 sau mai multe tipuri de pizza;
* **DIAGRAMA CONCEPTUALA**
* **Diagrama conceptuală**



* **Descrierea constrângerilor de integritate**
* Cheie primară:
* Utilizatori: Utilizator
* Locatii: Locatie\_ID
* Comenzi: Comanda\_ID
* Livrator: Livrator\_ID
* IstoricComenzi: (Comanda\_ID, Pizza\_ID)
* Pizza: Pizza\_ID
* IngredientPizza: (Pizza\_ID, Ingredient\_ID)
* Ingrediente: Ingredient\_ID
* Unique:
* Utilizatori: Telefon, (Nume,Prenume)
* Locatii: -
* Comenzi: -
* Livrator: -
* Pizza: Nume
* Not null:
* Locatii: Cartier, Strada, Utilizator
* Utilizatori: Parola, Nume, Telefon
* Livratori: Nume
* Comenzi: Utilizator, Livrator\_ID, Cost, Status
* Pizza: Nume, Pret
* IstoricComenzi: Pizza\_cantitate
* Ingrediente: nume
* Cheie externa:
* Locatii: Utilizator
* Comenzi: Utilizator, Livrator\_ID
* IstoricComenzi: Comanda\_ID, Pizza\_ID
* IngredientPizza: Pizza\_ID, Ingredient\_ID
* Check constraints:
* Locatii: locatie\_check ( Locatie\_ID > 0)
* Utilizator: telefon\_check (([telefon] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
* Default:
* Comenzi: Data\_comenzii
* Livratori: Data\_angajarii
* **Schemele Relaționale**

Am adaugat constrângerile de tip *on delete cascade* la cheile externe:

* Utilizator:

- la tabelul Locatii, pentru că atunci cand se sterge un utilizator sa se șteargă automat si toate Locatiile inregistrate pe numele utilizatorului respectiv;

-la tabelul Comenzi, pentru că atunci cand se șterge un utilizator sa se șteargă automat si comenzile făcute de el

* Pizza\_ID:

-la tabelul Ingredient Pizza, pentru că atunci când se șterge o anumita pizza sa se șteargă si toate informațiile in legătură cu ce ingrediente avea pizza respectiva

* Ingredient\_ID:

-la tabelul IngredientPizza, pentru că atunci când se sterge un ingredient, sa se șteargă si toate inregistrariile din acest tabel pentru că să nu se nască o pizza care are un ingredient care nici măcar nu există in baza de date a pizzeriei

La tabelul comenzi, nu am pus *on delete cascade* pentru că nu ar fi normal ca atunci cand șterg un livrator sa se șteargă si comandă. Pentru că astfel utilizatorul nu mai are acces la o comanda pe care a dus-o acel livrator.

**Crearea tabelelor inclusiv a constrangerilor**

* Utilizatori:

create table Utilizatori

(Utilizator varchar(20) constraint pk\_utilizator PRIMARY KEY,

Parola varchar(20) not null,

Nume varchar(20) not null,

Prenume varchar(20),

Email varchar(20),

Telefon varchar(10) not null constraint telefon\_unique UNIQUE,

contraint telefon\_check check (([telefon] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

constraint nume\_prenume\_unique unique(nume, prenume));

* Locatii:

create table Locatii

(Locatie\_ID int constraint pk\_locatie PRIMARY KEY,

Cartier varchar(20) not null,

Strada varchar(20) not null,

Adresa varchar(20),

Utilizator varchar(20) not null constraint fk\_utilizator references Utilizatori(Utilizator) on delete cascade,

constraint locatie\_check check(Locatie\_ID > 0 ));

* Livratori:

create table Livratori

(Livrator\_ID int constraint pk\_livrator PRIMARY KEY,

Nume varchar(30) not null,

Data\_angajarii datetime default getdate());

* Comenzi:

create table Comenzi

( Comanda\_ID int constraint pk\_comanda PRIMARY KEY,

Utilizator varchar(20) not null constraint fk\_utilizator FOREIGN KEY references Utilizatori(Utilizator) on delete cascade,

Cost int not null,

Livrator\_ID int constraint not null fk\_livrator FOREIGN KEY references Livratori(Livrator\_ID),

Data\_comenzii datetime default getdate()

Status varchar(20) not null);

* Pizza:

create table Pizza

( Pizza\_ID int constraint pk\_pizza PRIMARY KEY,

Nume varchar(20) not null constraint pizza\_unique UNIQUE,

Pret int not null,

Origine varchar(20));

* IstoricComenzi:

create table IstoricComenzi

( Comanda\_ID int,

Pizza\_ID int,

Pizza\_Cantitate int not null,

constraint fk1\_comanda foreign key (Comanda\_ID) references Comenzi(Comanda\_ID),

constraint fk2\_pizza foreign key (Pizza\_ID) references Pizza(Pizza\_ID),

constraint pk\_comanda\_pizza primary key(Comanda\_ID, Pizza\_ID));

* Ingrediente:

create table Ingrediente

(Ingredient\_ID int constraint pk\_ingredient primary key,

Nume varchar(20) not null constraint nume\_unique UNIQUE);

* IngredientPizza:

create table IngredientPizza

( Pizza\_ID int,

Ingredient\_ID int,

constraint fk1\_pizza foreign key(Pizza\_ID) references Pizza(Pizza\_ID) on delete cascade,

constraint fk2\_ingredient foreign key(Ingredient\_ID) references Ingrediente(Ingredient\_ID) on delete cascade,

constraint pk\_pizza\_ingredient primary key(Pizza\_ID, Ingredient\_ID));

**Introducere date:**

* Utilizatori

INSERT INTO dbo.Utilizatori VALUES('Ghita','ghita','Alex','Ghita',null,'0743325626');

INSERT INTO dbo.Utilizatori VALUES('Cornel','cornel','Viezure',null,'cornel@email.com','0743435626');

* Locatii:

INSERT INTO dbo.Locatii VALUES(1,'lujerului','dreptatii 22',null,'alex');

INSERT INTO dbo.Locatii VALUES(2,'veteranilor','eroilor 22',null,'Cornel');

* Livratori:

INSERT INTO dbo.Livratori VALUES(1,'Pedro',null,3000);

INSERT INTO dbo.Livratori VALUES(2,'Alejandro',null,3500);

* Comenzi:

INSERT INTO dbo.Comenzi VALUES(1,'alex',60,2,'nelivrat');

INSERT INTO dbo.Comenzi VALUES(2,'Cornel',50,2,'nelivrat');

* Pizza:

INSERT INTO dbo.Pizza VALUES(1,'Carnivora',30,'America');

INSERT INTO dbo.Pizza VALUES(2,'Margherita',20,'Napoli');

* IstoricComenzi:

INSERT INTO dbo.IstoricComenzi VALUES(1,1,2);

INSERT INTO dbo.IstoricComenzi VALUES(2,1,1);

INSERT INTO dbo.IstoricComenzi VALUES(2,2,1);

* Ingrediente:

INSERT INTO dbo.Ingrediente VALUES(1,'Sos de rosii');

INSERT INTO dbo.Ingrediente VALUES(2,'Sos de bbq');

INSERT INTO dbo.Ingrediente VALUES(3,'Mozzarella');

INSERT INTO dbo.Ingrediente VALUES(4,'Pui');

INSERT INTO dbo.Ingrediente VALUES(4,'Vita');

* IngredientPizza:

INSERT INTO dbo.Ingredient VALUES(1,2);

INSERT INTO dbo.Ingredient VALUES(1,3);

INSERT INTO dbo.Ingredient VALUES(1,4);

INSERT INTO dbo.Ingredient VALUES(1,3);

INSERT INTO dbo.Ingredient VALUES(2,1);

INSERT INTO dbo.Ingredient VALUES(2,3);