

FOOD REVIEWS

Nume Prenume: Stoean Vlad

Grupa: 1310B

Profesor Îndrumator: Mironeanu Cătălin

Scopul Aplicatiei:

Proiectul are ca scop atribuirea de recenzii tipurilor de mancare. Cel mai util aspect al proiectului este ca, in urma recenziilor diverselor feluri de mancare, utilizatorii pot atat sa aleaga restaurantul de la care sa-si ia o anumita masa, cat si sa primeasca puncte pentru reviewurile lasate.

Problema pe care o rezolva proiectul este atat cea de **"Unde mancam azi?"** cat si cea de **"Ce mancam azi?"**.

"Ce mancam azi?": Avand o lista de categorii de mancare (Ex : Mancare Mexicana, Fast Food, Mancare Traditionala Romaneasca etc.), fiecare cu o scurta descriere a meselor incluse, este foarte usor sa alegi lucrul de care iti e cel mai pofta. Cu ajutorul tabelii DISH, care include toate specialitatile dintr-o anumita categorie si cu tabela RESTAURANT_DISHES, devine chiar mai usor sa te hotarasti ce mananci in functie de un anumit buget (cu un minim, un maxim si un average extras din tabela).

"Unde mancam azi?": Dupa ce ai ales ce vrei sa mananci, poti sa te hotarasti si unde cu ajutorul celor doua tabele RESTAURANT_DISHES SI REVIEW, RESTAURANT_DISHES ajutand din nou la selectarea in functie de buget, iar cu ajutorul recenziilor din REVIEW te pot orienta avand un average dintre toate notele acordate mancarii (de la 1 la 10) un tip de mancare si recenzia aferenta.

"Sunt alergic la ... / Nu vreau sa mananc in oras astazi": In acest proiect mai exista si optiunea de a vedea ingredientele generale ale felurilor de mancare din tabela DISH.

"Am intampinat o problema la restaurant": Exista in tabela MANAGERS doua metode pentru a contacta managerul magazinului si a discuta problemele intampinate, prin telefon si prin email.

"Unde se afla restaurantul la care am decis sa merg?": In tabela RESTAURANTS exista un camp care-ti zice adresa exacta a restaurantului (poate chiar si un ling de google maps).

"De ce as lasa un review?": Pentru a ajuta pe altii la a decide ce sa manance si pentru a primi puncte pe care le poti folosi mai apoi pentru reduceri la restaurantele tale preferate.

Cu ajutorul acestui proiect se poate implementa o aplicatie pentru monitorizarea unor tipuri de mancare si a usura deciziile oamenilor cand vine vorba de luat sau facut mancare.

Asa arata aplicatia:

ID_USER	NAME	PHONE_NUMBER	POINTS
11	Avram Iancu	0711926088	0.35
1	Luiza Patras	0788888888	0.75
6	Anagus Asparagus	0788986722	0.35
8	Nimeru Asachi	0768993298	0.35
9	Pegaru Ioan Da...	0788993324	0.35
13	Anghel Siberiu	0731958282	0.35
14	Zirleanu Ionut	0789227095	0.35
*			

Exista un tab pentru fiecare tabela in care se pot adauga sterge si modifica valorile din tabela respectiva. Additional exista tabela Selections de unde se poate selecta un tip de mancare in functie de pret, scorul reviewurilor, tipul de mancare, categoria de mancare si greutate mesei. + Reviewurile oamenilor (doar scorul momentan)

Food Review

Users | Reviews | Food Categories | Dishes | Managers | Restaurants | Restaurant Dishes | Selection

☐ Food Category
 ☐ Dish
 ☐ Price Under
 ☐ Weight Under
 ☐ Score Over

Load

	Dish Name	Food Category Name	Restaurant Name	Price	Weight	Score	Average Score
▶	Taco	Mexican Food	We serve Pizza	5	0.2	3	7
	Taco	Mexican Food	We serve Pizza	5	0.2	8	7
	Taco	Mexican Food	We serve Pizza	5	0.2	10	7
	Gingerbread	Sweets	Bombonica lui Mit...	7	0.2	8	7
	Gingerbread	Sweets	Bombonica lui Mit...	7	0.2	7	7
	Gingerbread	Sweets	Bombonica lui Mit...	7	0.2	6	7
	Jelly	Sweets	Bombonica lui Mit...	4	0.1	4	4
	Taco	Mexican Food	LaMasa	10	0.3	9	9
	Taco	Mexican Food	LaMasa	10	0.3	9	9
	Milka Chocolate	Sweets	Bombonica lui Mit...	3.5	0.1	7	6.5
	Milka Chocolate	Sweets	Bombonica lui Mit...	3.5	0.1	6	6.5
*							

Tranzactia:

Daca un utilizator incarca un review, acesta primeste 10% din valoarea produsului cumparat sub forma de puncte pe care mai apoi le poate folosi sub forma de reduceri.

Tehnologii Folosite:

Am folosit Windows Forms cu C#, cu windows forms am facut interfata grafica iar comunicarea cu baza de data sql s-a facut prin API-ul ODP.NET (Oracle Data Provider for .NET) cu ajutorul unei comenzi ce conectare:

```

static OracleConnection connection;

1 reference
static public void connectToDataBase()
{
    string connectionString = "User Id=██████; Password=██████; Data Source=bd-dd
    connection = new OracleConnection(connectionString);
    try
    {
        connection.Open();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Error: {ex.Message}");
    }
}

```

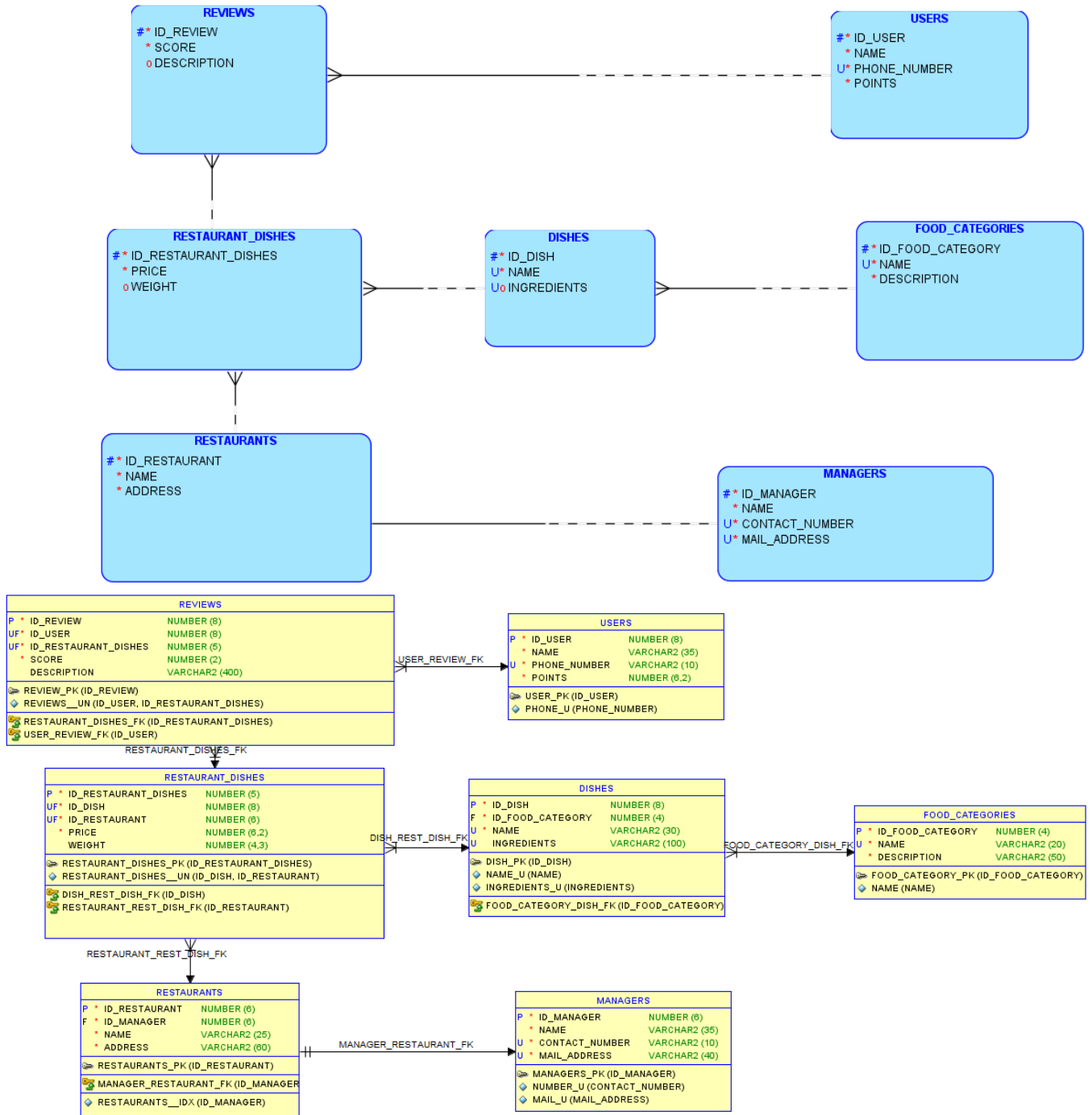
iar majoritatea comenzilor au fost rulate folosind 2 comenzi:

```

using (OracleDataReader reader = command.ExecuteReader())
{
    while (reader.Read())
    {
        comboBox.Items.Add(reader[0]);
    }
}
using (OracleCommand command = new OracleCommand(sqlQuery, connection))
{
    command.CommandTimeout = 5;
    command.ExecuteNonQuery();
}

```

Tabele:



- **USER:** Este tabela cu informatiile relevante despre persoana care si-a facut cont in aplicatie. In afara de numele complet si numarul de telefon, userul mai are si un numar de puncte pe care le primeste in momentul in care isi cumpara ceva de la un restaurant si lasa o recenzie. Punctele vor putea fi folosite in viitor in aplicatie pentru reduceri.
- **FOOD_CATEGORY:** Descrie categoria de mancare, fie ca e Fast Food sau mancare traditionala reprezentativa unei culturi sau a unei tari. Are si o mica descriere pentru a ajuta utilizatorul la alegerea categoriei.
- **MANAGERS:** Contine managerii fiecarui restaurant si informatiile lor de contact pentru a putea fi contactati in cazul in care exista probleme.
- **RESTAURANTS:** Are ca si campuri date referitoare la restaurant precum numele, adresa si managerul care-l detine.
- **DISH:** Aceasta tabela este strans lipita de FOOD_CATEGORY, fiecare tip de mancare apartinand unei categorii. Pe langa asta, fiecare tip de mancare are o lista de ingrediente pentru a putea fi preparat acasa si pentru a evita alergiile.
- **REVIEW:** In aceasta tabela fiecare sunt stocate reviewurile userilor cand vine vorba de un anumit tip de mancare prin o nota de la 1 la 10 si o mica descriere
- **RESTAURANT DISHES:** Contine preturile tipurilor de mancare de la restaurante impreuna cu pretul fiecareia pentru a fi usor de comparat cu mancaruri de la alte restaurante

Relatii:

- Relatii **ONE TO ONE:**
 1. Intre MANAGERS si RESTAURANTS: Un manager poate fi manageria un singur restaurant iar un restaurant poate fi detinut de un singur om.
- Relatii **ONE TO MANY:**
 1. Intre FOOD_CATEGORY si DISH: Un tip de mancare poate apartine unei singure cateogrii dar o categorie poate avea mai multe tipuri de mancare.
 2. Intre USERS si REVIEWS: Un User poate da review la mai multe tipuri de mancare dar un review nu poate fi dat de mai multi useri.
 3. Intre RESTAURANT_DISHES si REVIEWS: un review poate fi la un singur tip de mancare dintr-un restaurant.
 4. Intre DISHES si RESTAURANT_DISHES: pot fi mai multe mancaruri la restaurante din mai multe tipuri de mancare dar nu pot fi mai multe mancaruri de-odata.

5. Intre RESTAURANTS si RESTAURANT_DISHES: o mancare la restaurant poate fi servita doar la restaurantul la care e facuta.

Normalizare:

- **1NF:** In afara de campul ingrediente din tabela DISH, nu exista grupuri repetitive in alte tabele
- **2NF:** In tabelele REVIEW si RESTAURANT_DISHES exista cate un id unic pentru fiecare inserare din tabela pe langa perechile unice (USER + DISH respectiv DISH +RESTAURANT)
- **3NF:** Perechile din RESTAURANT_DISHES si REVIEW sunt unice

Constrangeri:

Exista multe constrangeri pentru a se asigura functionarea normala a unei aplicatii care ar putea fi implementate folosind aceasta baza de date.

PRIMARY KEY: Tabelele au cate un primary key dupa care se identifica o anumita inserare din tabela pentru ca aceasta sa fie unica si nenula.

NOT NULL: Aceasta constrangere se aplica in multe locuri precum:

- Numele unui utilizator
- Un numar de telefon
- O adresa de email pentru contact
- O adresa a unui restaurant
- O descriere a unei categorii de mancare
- Pretul unui anumit tip de mancare
- Scorul pentru o anumita mancare

UNIQUE: Folosit unde anumite coloane nu trebuie sa se repete precum:

- Numarul de telefon al unui utilizator
- Numele unei categorii de mancare
- Numele unui tip de mancare
- Ingredientele unui tip de mancare
- Metodele de contact pentru un manager
- Combinatia dintre mancarea din restaurant si user trebuie sa fie unica

FOREIGN KEY: Constrangere folosita pentru a asigura existenta unei valori deja existente dintr-o alta tabela:

- ID-urile userului si al tipului de mancare din REVIEW
- ID-ul categoriei de mancare din care face parte o masa in DISH
- ID-ul managerului care detine un restaurant in RESTAURANTS
- ID-urile unui restaurant si al unui tip de mancare in RESTAURANT_DISHES etc.

CONSTRANGERI DE MAXIM DE CARACTERE: Nu are sens ca si unele campuri sa aibe alocat prea mult spatiu asa incat se introduce o limita de caractere peste tot unde exista siruri.

CONSTRANGERI PENTRU NUME: Numele in marea majoritate a cazurilor (cu exceptia numelor de restaurante) trebuie sa contina doar litere si respectiv spatii.

CONSTRANGERI PENTRU NUMERE DE TELEFON: Numerele de telefon trebuie sa fie formate neaparat din 10 cifre pentru a fi valide

CONSTRANGERE PENTRU ADRESA DE EMAIL: Adresa de email trebuie sa fie de tipul caractere@ceva.ceva, sa aiba @ si cel putin un punct dupa @.

CONSTRANGERE DE PRET: Pretul unui produs nu poate sa fie negativ si nici sa treaca de o anumita limita (9999).

CONSTRANGERE DE SCOR: Unei mancari i se va atribui doar o nota cuprinsa intre 1 si 10.

Autoincrement:

Pentru toate tabelele unde exista un primary key exista si un autoincrement la acea coloana care incepe de la 1 si creste cu fiecare adaugare.