

Universitatea „Petrol – Gaze” din Ploiești

# **Bază de date - Informații de interes public zonal**

Realizat de: Coșereanu Emanuel, INFO an 2, grupa 40322

Coordonator: Lector Dr. Monica Vlădoiu

# Cuprins

Introducere .....	3
Descrierea segmentului din lumea reala.....	3
Cerințe utilizator.....	3
Diagrama Entitate-Legatură.....	4
Mulțimile entitate și attributele acestora (chei candidat și primare) .....	5
Mulțimile entitate și legăturile acestora .....	6
Mulțimea entitate incompletă și ierarhiile ISA .....	7
Schema BD și codul său MySQL .....	7
Interogări.....	15
Reuniune.....	15
Selecție .....	16
Proiecție.....	16
Joncțiune.....	17
Diferența.....	17

# Introducere

## Descrierea segmentului din lumea reala

Aplicația va fi concepută pentru a veni în sprijinul utilizatorilor dintr-o anumită regiune, oferindu-le o modalitate simplă, rapidă și interactivă de a accesa informații relevante despre localurile din zonă. Aceasta va include o gamă variată de locuri de interes, precum restaurante, cafenele, cofetării și gelaterii. Scopul principal al aplicației va fi să faciliteze accesul utilizatorilor la date de interes public zonal, contribuind astfel la o mai bună conectare cu serviciile și oportunitățile disponibile în proximitatea lor.

Prin intermediul aplicației, utilizatorii vor putea accesa detalii esențiale despre fiecare local, cum ar fi adresa exactă unde se află, serviciile oferite, programul de funcționare și informațiile de contact, toate acestea fiind prezentate într-un format ușor de înțeles și utilizat. Acest lucru va permite utilizatorilor să ia decizii informate atunci când vor dori să viziteze un local, să organizeze o ieșire cu prietenii sau să planifice o comandă la domiciliu.

Un alt aspect important al aplicației va fi integrarea informațiilor despre firmele de livrare. Astfel, utilizatorii vor putea vizualiza cu ce companii vor colabora fiecare local pentru efectuarea livrărilor, ceea ce le va oferi o imagine clară asupra opțiunilor disponibile pentru comenzi la domiciliu. Acest lucru va fi util mai ales pentru cei care vor dori să beneficieze de servicii de livrare rapide și de încredere.

În plus, aplicația va oferi funcții avansate care vor permite utilizatorilor să își gestioneze un istoric al tranzacțiilor efectuate la diverse localuri. Această caracteristică le va oferi posibilitatea de a ține evidența comenzilor anterioare, de a analiza obiceiurile de consum și de a reveni rapid la localurile preferate. Prin aceste funcționalități, aplicația va deveni nu doar un instrument de informare, ci și un partener util în organizarea activităților zilnice ale utilizatorilor.

Prin urmare, baza de date utilizată de aplicație va fi proiectată pentru a stoca și organiza informații de interes public zonal, asigurând o actualizare constantă și o accesibilitate crescută a datelor. Această structură va permite atât utilizatorilor, cât și operatorilor de localuri să beneficieze de o platformă modernă și eficientă, adaptată nevoilor comunității locale.

## Cerințe utilizator

Să ofere posibilitatea păstrării de informații detaliate despre localurile din zonă, precum adresa, detalii de contact și firmele de livrare cu care acestea lucrează.

Să faciliteze stocarea de informații „extra” despre localuri, în funcție de specificul acestora. De exemplu, la restaurante să se memoreze specificul bucătăriei, la cafenele să se specifice dacă

acestea oferă spațiu de lucru, iar la cofetării să se menționeze dacă oferă prăjituri fără zahăr, alături de alte informații similare.

Să mențină un registru cu toate chitanțele pe care utilizatorul le are, astfel încât să fie ușor să se observe care sunt tendințele acestuia și ce localuri preferă.

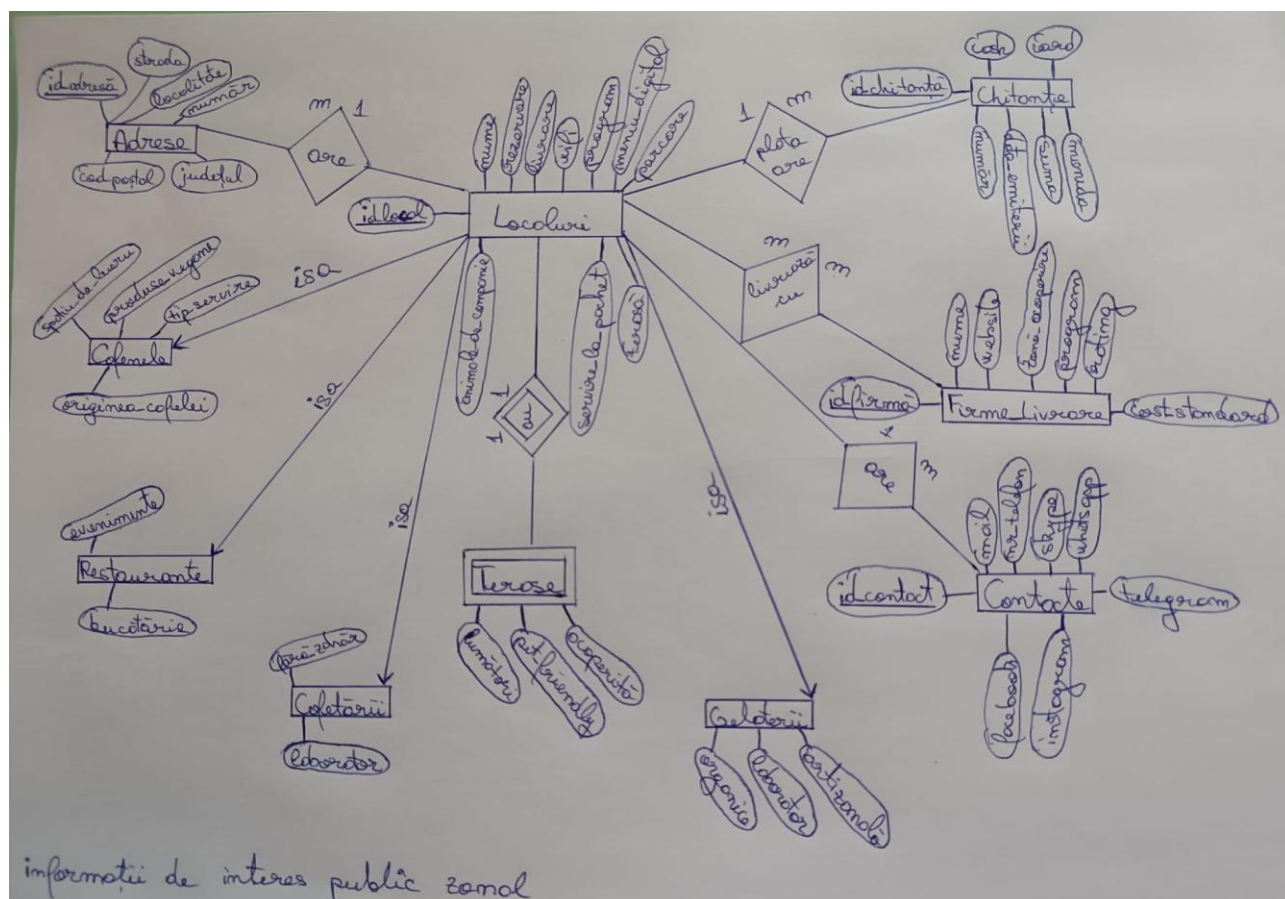
Să ofere posibilitatea de căutări rapide după diverse criterii:

- cuvinte cheie (nume local);
- adresă;
- firme de livrare cu care cooperează localurile.

Să fie disponibilă online.

## Diagrama Entitate-Legatură

Mai jos este o poza a diagramei enitate-legatură:



## **Mulțimile entitate și atributele acestora (chei candidat și primare)**

Mai jos sunt prezentate informații despre fiecare mulțime din baza de date, specificându-se cheia primară și cheile candidat:

### 1) Localuri

- Atribute: id\_local (cheie primară), nume, rezervare, livrare, wifi, program, meniu\_digital, parcare, animale\_de\_companie, servire\_la\_pachet, terasă;
- Chei candidate: nume (unic, identifică fiecare local);
- Cheia primară aleasă: id\_local (generat autoamt, unic și reducere riscul de ambiguitate).

### 2) Restaurante

- Atribute: evenimente, bucătărie;
- Chei candidate: -;
- Cheia primară: -.

### 3) Terasa

- Atribute: acoperită, fumători, pet\_friendly;
- Chei candidate: -;
- Cheia primară: -.

### 4) Cafenele

- Atribute: originea\_cafelei, tip\_servire, produse\_vegane, spatiu\_de\_lucru;
- Chei candidate: -;
- Cheia primară: -.

### 5) Cofetării

- Atribute: laborator, fără\_zahăr;
- Chei candidate: -;
- Cheia primară: -.

### 6) Gelaterii

- Atribute: artizanală, laborator, organice
- Chei candidate: -;
- Cheia primară: -.

## 7) Adrese

- Atribute: id\_adresă (cheie primară), strada, localitate, număr, cod\_poștal, județul;
- Chei candidate: strada, localitate, număr, cod\_poștal, județul (pot obține o adresă unică dacă știu aceste atribute);
- Cheia primară: id\_adresă (generat autoamt, unic și reducere riscul de ambiguitate).

## 8) Chitanțe

- Atribute: id\_chitanță (cheie primară), număr\_serie, data\_emiterii, suma, moneda, cash, card;
- Chei candidate: număr\_serie, data\_emiterii (cu aceste atribute putem identifica o chitanță din tabel);
- Cheia primară: id\_chitanță (generat autoamt, unic și reducere riscul de ambiguitate).

## 9) Firme\_Livrare

- Atribute: id\_firmă (cheie primară), website, zonă\_acoperire, program, rating, nume, cost\_standard;
- Chei candidate: nume (unic, identifică fiecare firmă);
- Cheia primară: id\_firmă (generat autoamt, unic și reducere riscul de ambiguitate).

## 10) Contacte

- Atribute: id\_contact (cheie primară), mail, nr\_telefon, skype, whatsapp, telegram, facebook, instagram;
- Chei candidate: mail, nr\_telefon (putem identifica un rând din tabelul Contacte, dacă știm aceste atribute);
- Cheia primară: id\_contact (generat autoamt, unic și reducere riscul de ambiguitate).

## 11) Localuri\_Firme\_Livrare

- Atribute: -;
- Chei candidate: -;
- Chei primare: id\_local, id\_firmă (unic și reducere riscul de ambiguitate).

# Mulțimile entitate și legăturile acestora

În rândurile ce urmează va fi explicată fiecare legătură din diagrama entitate-legătură:

- Localuri – Adrese (are). Tipul de apartenență: 1 – m. Gradul legăturii: 2. Conectivitate: -.

- Localuri – Firme\_Livrare (livrează cu). Tipul de apartenență: m – m. Gradul legăturii: 2. Conectivitate: -.
- Localuri – Contacte (are). Tipul de apartenență: 1 – m. Gradul legăturii: 2. Conectivitate: -.
- Localuri – Chitante (oferă). Tipul de apartenență: 1 – m. Gradul legăturii: 2. Conectivitate: -.
- Localuri – Terasse (au). Tipul de apartenență: 1 – 1. Gradul legăturii: 2. Conectivitate: -.
- Firme\_Livrare – Contacte (are). Tipul de apartenență: 1 – m. Gradul legăturii: 2. Conectivitate: -.

## Mulțimea entitate incompletă și ierarhiile ISA

Mai jos sunt explicate ierahiile ISA, logica prin care acestea au fost create și mulțimea entitate incompletă „Terasse” și legătura dintre ea și „Localuri”.

- Localuri – Terasse (au). Aceasta este o mulțime entitate incompletă, deoarece terasele nu pot exista fără localuri, iar un local poate avea 0 sau 1 terasă.
- Ierahie ISA: Localuri (entitate generală). Subtipuri:
  - Restaurante (susțin evenimente, au un anumit tip de bucătărie).
  - Cafenele (au cafea de o anumită origine, un anumit tip de servire, oferă sau nu spațiu de lucru, are sau nu produse vegane).
  - Cofetării (au sau nu laborator propriu / produse fără zahăr).
  - Gelaterii (este sau nu artizanală, are sau nu laborator / produse organice).

## Schema BD și codul său MySQL

Mai jos sunt prezentate tabele din baza de date și atributele acestora:

- ➔ Localuri (**id\_local**, nume, rezervare, livrare, wifi, program, meniu\_digital, parcare, animale\_de\_companie, servire\_la\_pachet, terasa);
- ➔ Restaurante (evenimente, bucătărie, *id\_local*);
- ➔ Terasse (fumatori, pet\_friendly, acoperita, *id\_local*);
- ➔ Cafenele (saptiu\_de\_lucru, produse\_vegane, tip\_servire, originea\_cafelei, *id\_local*);

- ➔ Cofetarii (fara\_zahar, laborator, *id\_local*);
- ➔ Gelaterii (organice, laborator, artizanala, *id\_local*);
- ➔ Adrese (**id\_adresa**, strada, localitate, numar, cod\_postal, judetul, *id\_local*);
- ➔ Chitante (**id\_chitanta**, numar, data\_emiterii, suma, moneda, cash, card, *id\_local*);
- ➔ Contacte (**id\_contact**, mail, nr\_telefon, skype, whatsapp, telegram, facebook, instagram, *id\_local*, *id\_firma*);
- ➔ Firme\_Livrare (**id\_firma**, nume, website, zona\_acoperita, program, rating, cost\_standard);
- ➔ Localuri\_Firme\_Livrare (*id\_local*, *id\_firma*);

#### 1. Relația „are” între Localuri și Adrese

- ✓ Cum este realizată? Am adăugat în tabela Adrese o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

#### 2. Relația „are” între Localuri și Contacte

- ✓ Cum este realizată? Am adăugat în tabela Contacte o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

#### 3. Relația „livrează cu” între Localuri și Firme\_Livrare

- ✓ Cum este realizată? Am adăugat în tabela Firme\_Livrare o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

#### 4. Relația „oferă” între Localuri și Chitanțe

- ✓ Cum este realizată? Am adăugat în tabela Chitante o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

#### 5. Relația „au” între Localuri și Terase (mulțime entitate incompletă)

- ✓ Cum este realizată? Am adăugat în tabela Terase o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`



6. Relația „are” între Firme\_Livrare și Contacte

- ✓ Cum este realizată? Am adăugat în tabela Contacte o cheie străină denumită **id\_firma** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_firma INT, FOREIGN KEY (id_firma) REFERENCES Firme_Livrare(id_firma) ON DELETE CASCADE.`

7. Ierarhia ISA între Localuri și Restaurante

- ✓ Cum este realizată? Am adăugat în tabela Restaurante o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

8. Ierarhia ISA între Localuri și Cafenele

- ✓ Cum este realizată? Am adăugat în tabela Cafenele o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

9. Ierarhia ISA între Localuri și Cofetării

- ✓ Cum este realizată? Am adăugat în tabela Cofetării o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

10. Ierarhia ISA între Localuri și Gelaterii

- ✓ Cum este realizată? Am adăugat în tabela Gelaterii o cheie străină denumită **id\_local** pentru a face legătura între cele doua tabele.
- ✓ Cod SQL: `id_local INT, FOREIGN KEY (id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE.`

### (1) Tabela **Localuri**

```
CREATE TABLE Localuri (id_local INT AUTO_INCREMENT PRIMARY KEY,
nume VARCHAR(255), rezervare BOOLEAN, livrare BOOLEAN, wifi BOOLEAN, program
VARCHAR(255), meniu_digital VARCHAR(255), parcare BOOLEAN, animale_de_companie
BOOLEAN, servire_la_pachet BOOLEAN, terasa BOOLEAN);
```

```
mysql> show columns from localuri;
```

Field	Type	Null	Key	Default	Extra
id_local	int(11)	NO	PRI	NULL	auto_increment
nume	varchar(255)	NO		NULL	
rezervare	tinyint(1)	YES		NULL	
livrare	tinyint(1)	YES		NULL	
wifi	tinyint(1)	YES		NULL	
program	varchar(255)	YES		NULL	
meniu_digital	varchar(255)	YES		NULL	
parcare	tinyint(1)	YES		NULL	
animale_de_companie	tinyint(1)	YES		NULL	
servire_la_pachet	tinyint(1)	YES		NULL	
terasa	tinyint(1)	NO		0	

```
11 rows in set (0,01 sec)
```

### (2) Tabela **Gelaterii**

```
CREATE TABLE Gelaterii(organice BOOLEAN, laborator BOOLEAN, artizanala
BOOLEAN, id_local INT, FOREIGN KEY(id_local) REFERENCES Localuri(id_local) ON
DELETE CASCADE);
```

```
mysql> show columns from gelaterii;
```

Field	Type	Null	Key	Default	Extra
organice	tinyint(1)	YES		NULL	
laborator	tinyint(1)	YES		NULL	
artizanala	tinyint(1)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
4 rows in set (0,00 sec)
```

### (3) Tabela **Cofetării**

```
CREATE TABLE Cofetarii(fara_zahar BOOLEAN, laborator BOOLEAN,  
id_local INT, FOREIGN KEY(id_local) REFERENCES Localuri(id_local) ON DELETE  
CASCADE);
```

```
mysql> show columns from cofetarii;
```

Field	Type	Null	Key	Default	Extra
fara_zahar	tinyint(1)	YES		NULL	
laborator	tinyint(1)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
3 rows in set (0,00 sec)
```

### (4) Tabela **Cafenele**

```
CREATE TABLE Cafenele (spatiu_de_lucru BOOLEAN, produse_vegane  
BOOLEAN, tip_servire VARCHAR(255), originea_cafelei VARCHAR(255), id_local INT,  
FOREIGN KEY(id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE);
```

```
mysql> show columns from cafenele;
```

Field	Type	Null	Key	Default	Extra
spatiu_de_lucru	tinyint(1)	YES		NULL	
produse_vegane	tinyint(1)	YES		NULL	
tip_servire	varchar(255)	YES		NULL	
originea_cafelei	varchar(255)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
5 rows in set (0,00 sec)
```

### (5) Tabela **Restaurante**

```
CREATE TABLE Restaurante (evenimente BOOLEAN, bucatarie  
VARCHAR(255), id_local INT, FOREIGN KEY(id_local) REFERENCES  
Localuri(id_local) ON DELETE CASCADE);
```

```
mysql> show columns from restaurante;
```

Field	Type	Null	Key	Default	Extra
evenimente	tinyint(1)	YES		NULL	
bucatarie	varchar(255)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
3 rows in set (0,00 sec)
```

(6) Tabela **Terase**

```
CREATE TABLE Terasa (fumatori BOOLEAN, pet_friendly
VARCHAR(255), acoperita BOOLEAN, id_local INT, FOREIGN KEY(id_local)
REFERENCES Localuri(id_local) ON DELETE CASCADE);
```

```
mysql> show columns from terase;
```

Field	Type	Null	Key	Default	Extra
fumatori	tinyint(1)	YES		NULL	
pet_friendly	varchar(255)	YES		NULL	
acoperita	tinyint(1)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
4 rows in set (0,00 sec)
```

(7) Tabela **Adrese**

```
CREATE TABLE Adrese (id_adresa INT AUTO_INCREMENT PRIMARY
KEY, strada VARCHAR(255), localitate VARCHAR(255), numar VARCHAR(255),
cod_postal VARCHAR(255), judetul VARCHAR(255), id_local INT, FOREIGN KEY
(id_local) REFERENCES Localuri(id_local) ON DELETE CASCADE);
```

```
mysql> show columns from adrese;
```

Field	Type	Null	Key	Default	Extra
id_adresa	int(11)	NO	PRI	NULL	auto_increment
strada	varchar(255)	YES		NULL	
localitate	varchar(255)	YES		NULL	
numar	varchar(255)	YES		NULL	
cod_postal	varchar(255)	YES		NULL	
judetul	varchar(255)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
7 rows in set (0,00 sec)
```

(8) Tabela **Chitante**

```
CREATE TABLE Chitante (id_chitanta INT AUTO_INCREMENT
PRIMARY KEY, numar INT, data_emiterii DATE, suma FLOAT, moneda VARCHAR(3),
cash BOOLEAN, card BOOLEAN, id_local INT, FOREIGN KEY (id_local)
REFERENCES Localuri(id_local) ON DELETE CASCADE);
```

```
mysql> show columns from chitante;
```

Field	Type	Null	Key	Default	Extra
id_chitanta	int(11)	NO	PRI	NULL	auto_increment
numar	int(11)	YES		NULL	
data_emiterii	date	YES		NULL	
suma	float	YES		NULL	
moneda	varchar(3)	YES		NULL	
cash	tinyint(1)	YES		NULL	
card	tinyint(1)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	

```
8 rows in set (0,01 sec)
```

(9) Tabela **Firme\_Livrare**

```
CREATE TABLE Firme_Livrare(id_firma INT AUTO_INCREMENT
PRIMARY KEY, nume VARCHAR(255), website VARCHAR(255), zona_acoperire
VARCHAR(255), program VARCHAR(255), rating INT, cost_standard FLOAT);
```

```
mysql> show columns from firme_livrare;
```

Field	Type	Null	Key	Default	Extra
id_firma	int(11)	NO	PRI	NULL	auto_increment
nume	varchar(255)	YES		NULL	
website	varchar(255)	YES		NULL	
zona_acoperire	varchar(255)	YES		NULL	
program	varchar(255)	YES		NULL	
rating	int(11)	YES		NULL	
cost_standard	float	YES		NULL	

```
7 rows in set (0,00 sec)
```

(10) Tabela **Localuri\_Firme\_Livrare**

```
CREATE TABLE Localuri_Firme_Livrare(id_local INT, id_firma INT,
PRIMARY KEY (id_local, id_firma), FOREIGN KEY (id_local) REFERENCES
Localuri(id_local) ON DELETE CASCADE, FOREIGN KEY (id_firma) REFERENCES
Firme_Livrare(id_firma) ON DELETE CASCADE);
```

```
mysql> mysql> show columns from localuri_firme_livrare;
```

Field	Type	Null	Key	Default	Extra
id_local	int(11)	NO	PRI	NULL	
id_firma	int(11)	NO	PRI	NULL	

```
2 rows in set (0,03 sec)
```

(11) Tabela **Contacte**

```
CREATE TABLE Contacte(id_contact INT AUTO_INCREMENT
PRIMARY KEY, mail VARCHAR(255), nr_telefon VARCHAR(25), skype VARCHAR(60),
whatsapp VARCHAR(25), telegram VARCHAR(25), facebook VARCHAR(60), instagram
VARCHAR(60), id_local INT, id_firma INT, FOREIGN KEY(id_local) REFERENCES
Localuri(id_local) ON DELETE CASCADE, FOREIGN KEY(id_firma) REFERENCES
Firme_Livrare(id_firma) ON DELETE CASCADE);
```

```
mysql> show columns from contacte;
```

Field	Type	Null	Key	Default	Extra
id_contact	int(11)	NO	PRI	NULL	auto_increment
mail	varchar(255)	YES		NULL	
nr_telefon	varchar(25)	YES		NULL	
skype	varchar(60)	YES		NULL	
whatsapp	varchar(25)	YES		NULL	
telegram	varchar(25)	YES		NULL	
facebook	varchar(60)	YES		NULL	
instagram	varchar(60)	YES		NULL	
id_local	int(11)	YES	MUL	NULL	
id_firma	int(11)	YES	MUL	NULL	

```
10 rows in set (0,00 sec)
```

# Interogări

## Reuniune

Interogare: Să se afișeze numele tuturor localurilor și firmelor de livrare din zona.

SQL-query:

SELECT        nume  
FROM        Localuri  $R = \pi_{nume}(Localuri) \cup \pi_{nume}(Firme\_Livrare)$   
UNION SELECT nume FROM Firme\_Livrare;

Forma algebrică:

```
mysql> SELECT nume FROM Localuri UNION SELECT nume FROM Firme_Livrare;
```

nume
McDonald's
KFC
MADO
Gigi Burger
Mesopotamia
Nabucco
DaVinci
Pizza Hut
Cafeneaua Nației
5 to go
Cafe Milano
Street Coffee Roasters
Okfea Ploiești
Kaldio Ploiești Poșta Mare
Cofetăria Ana State
Cofetăriile Delice
Cofetăria Narcisa
Cofetăria Oli
Cofetăria DoelPan
Eva Cake House
Simply Delicious by ANNA
Dulcelle - Laborator Cofetărie   Ciocolaterie
Hanul Dacilor
Taverna Grecească Dionysos
Akua Garden Restaurant
Steak House No.16
Mămăliguța Restaurant
The Fable
Jinx
Restaurant Antique
Burger King
Popeyes
Bolt Food
Glovo
Tazz

35 rows in set (0,00 sec)

## Selecție

Interogare: Să se afișeze numele, programul și meniul digital al localurilor care nu au wifi.

SQL-query:

SELECT nume,  $R = \pi_{nume, program, meniu\_digital}(\sigma_{wifi=0}(\text{Localuri}))$   
program,  
meniu\_digital FROM Localuri WHERE wifi = 0;

Forma  
algebrică

```
[mysql> SELECT nume, program, meniu_digital FROM Localuri WHERE wifi = 0;
```

nume	program	meniu_digital
MADO	Luni-Duminica: 10:00 - 22:00	https://www.mado.ro/meniu-mado
Gigi Burger	Luni-Sambata: 10:00 - 22:00	nu are meniu digital
Simply Delicious by ANNA	Luni-Sambata: 09:00 - 19:00, Duminica: 10:00 - 18:00	nu are meniu digital
Mămăliguța Restaurant	Luni-Duminică: 09:00 - 22:00	nu are meniu digital

4 rows in set (0,00 sec)

## Proiecție

Interogare: Să se afișeze numele, programarul localurilor și daca acestea acceptă rezervări și dacă au wifi.

SQL-query: SELECT

DISTINCT nume,  $R = \pi_{nume, rezervare, wifi, program}(\text{Localuri})$   
rezervare, wifi,  
program from Localuri;

Forma algebrică:

```
[mysql> SELECT DISTINCT nume, rezervare, wifi, program from Localuri;
```

nume	rezervare	wifi	program
McDonald's	0	1	Luni-Vineri: 8:00 - 00:00, Sambata-Duminica: 10:00 - 22:00
KFC	0	1	Luni-Duminica: 8:00 - 22:00
MADO	0	0	Luni-Duminica: 10:00 - 22:00
Gigi Burger	1	0	Luni-Sambata: 10:00 - 22:00
Mesopotamia	1	1	Luni-Duminica: 10:00 - 23:00
Nabucco	1	1	Luni-Duminica: 12:00 - 23:00
DaVinci	1	1	Luni-Duminica: 8:00 - 23:00
Pizza Hut	1	1	Luni-Duminică: 10:00 - 22:00
Cafeneaua Nației	1	1	Luni-Vineri: 08:00 - 20:00, Sămbătă-Duminică: 10:00 - 18:00
5 to go	0	1	Luni-Vineri: 07:00 - 19:00, Sămbătă-Duminică: 08:00 - 18:00
Cafe Milano	1	1	Luni-Duminică: 09:00 - 23:00
Street Coffee Roasters	1	1	Luni-Vineri: 08:00 - 20:00, Sămbătă-Duminică: 09:00 - 18:00
Okfea Ploiești	1	1	Luni-Duminică: 08:00 - 22:00
Kaldio Ploiești Poșta Mare	1	1	Luni-Vineri: 09:00 - 19:00, Sămbătă-Duminică: 10:00 - 20:00
Cofetăria Ana State	1	1	Luni-Duminică: 09:00 - 19:00
Cofetăriile Delice	0	1	Luni-Vineri: 08:00 - 20:00, Sămbătă-Duminică: 09:00 - 18:00
Cofetăria Narcisa	1	1	Luni-Duminică: 10:00 - 22:00
Cofetăria Oli	0	1	Luni-Sămbătă: 09:00 - 19:00, Duminica: 10:00 - 18:00
Cofetăria DoelPan	1	1	Luni-Duminică: 09:00 - 20:00
Eva Cake House	1	1	Luni-Vineri: 08:00 - 20:00, Sămbătă-Duminică: 09:00 - 18:00
Simply Delicious by ANNA	1	0	Luni-Sămbătă: 09:00 - 19:00, Duminica: 10:00 - 18:00
Dulcelle - Laborator Cofetărie	0	1	Luni-Duminică: 08:00 - 21:00
Hanul Dacilor	1	1	Luni-Duminică: 12:00 - 23:00
Taverna Grecească Dionysos	1	1	Luni-Duminică: 11:00 - 23:00
Akua Garden Restaurant	1	1	Luni-Duminică: 10:00 - 22:00
Steak House No.16	1	1	Luni-Duminică: 10:00 - 23:00
Mămăliguța Restaurant	1	0	Luni-Duminică: 09:00 - 22:00
The Fable	1	1	Luni-Duminică: 10:00 - 22:00
Jinx	0	1	Luni-Duminică: 12:00 - 23:00
Restaurant Antique	1	1	Luni-Duminică: 11:00 - 23:00
Burger King	0	1	Luni-Duminică: 10:00 - 23:00
Popeyes	0	1	Luni-Duminică: 10:00 - 22:00

32 rows in set (0,02 sec)



## Joncțiune

Interogare: Să se afișeze numele tuturor localurilor pentru care restaurantele oferă bucătărie românească și dacă acestea au wifi și terasă.

SQL-query: `SELECT l.num, l.wifi, r.bucatarie, r.terasa FROM Localuri l JOIN Restaurante r ON l.id_local = r.id_local WHERE r.bucatarie = 'romaneasca';`

Forma algebrică:  $R = \sigma_{bucatarie='romaneasca'}(Localuri \bowtie_{Localuri.id\_local=Restaurante.id\_local} Restaurante)$

```
mysql> SELECT l.num, l.wifi, r.bucatarie, l.terasa FROM Localuri l JOIN Restaurante r ON l.id_local = r.id_local WHERE r.bucatarie = 'romaneasca';
```

nume	wifi	bucatarie	terasa
McDonald's	1	romaneasca	1
KFC	1	romaneasca	0
Gigi Burger	0	romaneasca	1
Hanul Dacilor	1	romaneasca	1
Taverna Grecească Dionysos	1	romaneasca	1
Akua Garden Restaurant	1	romaneasca	0
Mămăliguța Restaurant	0	romaneasca	0
Jinx	1	romaneasca	1

8 rows in set (0,00 sec)

## Diferența

Interogare: Să se afișeze numele tuturor localurilor care au firma de livrare proprie.

SQL-query: `SELECT l.num FROM localuri l WHERE EXISTS ( SELECT NULL FROM localuri_firme_livrare lf JOIN firme_livrare f ON lf.id_firma = f.id_firma WHERE lf.id_local = l.id_local AND f.num = l.num);`

$R = \pi_{num}(Localuri) - \pi_{num}(\sigma_{f.num \neq l.num}(Localuri \bowtie Localuri\_Firme\_Livrare \bowtie Firme\_Livrare))$

Forma algebrică:

```

-->      FROM lo' at line 2
mysql> SELECT l.num FROM localuri l WHERE EXISTS ( SELECT NULL FROM localuri_firme_livrare lf JOIN firme_livrare f ON lf.id_firma = f.id_firma WHERE lf.id_local = l.id_local AND f.num = l.num);
+-----+
| num |
+-----+
| DaVinci |
| Pizza Hut |
| Cofetăria Ana State |
| Cofetăria Oli |
| Taverna Grecească Dionysos |
+-----+
5 rows in set (0,00 sec)

mysql> █

```