Proiect Baza de date

-documentație-

Tema: INFORMAȚII CONTINENTE, ȚĂRI, ORAȘE-TURISM

Student: Brânzea Ana-Maria

Grupa 40317

Cuprins

1. Nevoia de aplicație ………………………………………………….
2. Diagrama Entitate-Legatură……………………………………..
3. Schema bazei de date și codul SQL ………………………..
4. Operații asupra bazei de date……………………………….
5. Bibliografie……………………………………………………..
6. **Nevoia de aplicație**

În acest capitol, cititorul va fi familiarizat cu aplicarea mecanismelor conceptuale pentru dezvoltarea aplicației de baza de date care modulează un segment din lumea reală, acela fiind gestionarea informațiilor cu privire la turismul de pe planetă, în functie de continente, țari și orașe.

În mod traditional, oamenii doritori de a vizita și de a cunoaste mai de aproape ce îi înconjoară, se informau cu privire la obiectivele turistice din materiale precum reviste,cărți, ziare. De obicei, informațiile erau în numar mic, iar pentru a plănui o călătorie în scopul vizitării, persoanele îsi alocau un timp destul de mare pentru a se informa( căutarea unei cazari, căutarea adreselor obiectivelor turistice, informarea cu privire la prețurile pentru vizitare etc.). Din această cauză, s-a decis să se simplifice căutarea prin construirea unei aplicații care să ofere toate informațiile necesare,la un singur click distantă.

În urma celor spuse mai devreme, au rezultat următoarele cerințe principale pentru aplicația noastră:

* Să ofere posibilitatea păstrării de informații detaliate despre obiectivele turistice de interes aflate pe un anumit continent, într-o anumita țară, într-un anumit oraș;
* Să păstreze și să ofere informații despre programurile de vizită și adresele fiecărui obiectiv turistic, cu scopul de a ajuta utilizatorul să se orienteze mult mai repede;
* Să ofere posibilitatea de a căuta dupa diverse criterii:

1. Dupa continentul/țara/orașul în care se dorește să se viziteze;
2. Dupa tipul obiectivelor turistice (saline, muzee, peșteri, castele, grădini zoo, grădini botanice etc);

* Să fie disponibilă online.

1. **Diagrama Entitate-Legatură si normalizarea**

**bazei de date**

Pentru a realiza descrierea datelor am folosit modelarea Entitate-Legatură care ilustrează grafic descrierea bazei de date.

În aceasta diagrama Entitate-Legatură, regăsim următoarele mulțimi entitate: CONTINENTE, TARI, LIMBI\_OFICIALE,ORASE, CAZARI, CATEGORII\_CAZARE, OBIECTIVE,CATEGORII\_PRETURI, MUZEE, BISERICI\_ȘI\_MANASTIRI, PARCURI, MONUMENTE\_ALE\_NATURII, SALINE, GRADINI\_ZOOLOGICE, GRADINI\_BOTANICE, CASTELE\_ȘI\_CETATI, PESTERI, ADRESE, PROGRAME\_DE\_VIZITA.

Mulțimea entitate CONTINENTE conține toate numele continentelor de pe planeta și are ca atribute: ID\_CONTINENT(cheie primară)- prin care se identifică unic fiecare continent , NUME.

Mulțimea entitate TARI stochează informații utile pentru utilizatori precum atributele NUME- denumirea țării,COD\_ISO\_MONEDA- codul monedei ce se folosește în acea țară, \*LIMBA- limbile oficiale vorbite, cheia primară ID\_TARA -desemneaza toate tarile din care vrem să aflam informații legate de turism, \*LIMBA- limbile oficiale vorbite în țari.

După cum ne putem da seama, fiecare țară poate avea mai multe limbi oficiale, astfel tabela nu se afla în FN3, ci în FN1. Prin urmare, pentru a elimina atributul "limba" din tabela "TARI", a fost creată o altă tabelă cu denumirea "LIMBI\_OFICIALE".Astfel, în loc să includem un atribut "limba" în tabela "TARI" și să permitem înregistrări multiple pentru limbi în aceeași înregistrare a unei țări, tabela " LIMBI\_OFICIALE " este creată separat. Aceasta va conține o cheie primara( ID\_LIMBA) și atributul LIMBA\_OFICIALA- limba prevăzută pentru oficiile și instituțiile statului, atât ele între ele, cât și în comunicația cu cetățenii.

ORASE este o mulțime entitate care conține atributele: ID\_ORAȘ(cheie primară), NUME- numele tarii, NR\_ATRACTII\_TURISTICE- totalul de atracții din acea tara.

Entitatea CAZARI cuprinde informații despre diferite hoteluri,apartamente, pensiuni, în alte cuvinte, ajută la alegerea unei cazări. Atributele care fac parte din entitate sunt: ID\_CAZARE(cheie primară), DENUMIRE, \*TIP( dacă este hotel/ pensiune/ motel/ complex de apartamente/vilă),\*STRADA,\*NUMAR- realizează localizarea mai ușor.

Entitatea CAZARI se găsește în FN2, iar pentru a muta tabela în FN3, am eliminat dependențele funcționale nedorite și am creat tabele separate pentru atributele marcate cu \*. Astfel, am creat entitatea CATEGORII\_CAZARE, care are următoarele atribute: id\_categorie (cheie primară), tip.De asemenea, am conectat entitatea CAZARI cu entitatea ADRESE.

În ceea ce privește mulțimea entitate OBIECTIV, acesta deține o descriere (DESCRIERE) despre un obiectiv(NUME) care are anumite coordonate (COORDONATE) și se identifică prin cheia primară ID\_OBIECTIV.

La început, anumite obiective impuneau un cost de intrare. Pentru a standardiza această situație, am introdus entitatea "CATEGORII\_PRETURI" pentru a administra prețurile biletelor de intrare la diverse obiective. Această entitate are două caracteristici: "cod\_categorie" și "denumire", care descriu categoriile de prețuri disponibile, precum "elev", "student", "pensionar" și "adult".

Mulțimea entitate MUZEE conține atributele: ID\_MUZEU(cheie primară), TIP- daca este muzeu istoric, de artă, istorie locală, geologic, etnomuzeu etc..

Mulțimea BISERICI\_ȘI\_MANASTIRI conține: ID(cheie primară), STIL- stilul arhitectural în care este construit, ANUL\_CONSTRUIRII- numele fiind sugestiv.

Mulțimea entitate PARCURI are atribute precum: ID\_PARC(cheie primară), TIP- tipul parcului, de exemplu, parc de joacă, parc natural, parc urban etc., SUPRAFATA, ANUL\_INFINTARII- cel în care a fost deschis/ a apărut.

MONUMENTE\_ALE\_NATURII se referă la obiectivele unde nu a fost implicată mâna omului. Atributele specifice entității sunt: ID\_MONUMENT(cheie primară), ALTITUDINE- cea la care se găsește obiectivul.

Entitații SALINE îi corespund atributele: ID\_SALINA( cheie primară), ADANCIME- cât de adâncă este salina.

GRADINI\_ZOOLOGICE conține atributele: ID\_ZOO(cheie primară), SUPRAFATA, ATRACTIE- cel mai apreciat animal.

De asemenea, entitatea GRADINI\_BOTANICE are o cheie primară ID\_GRADINA, urmată de atribute precum: SUPRAFATA, NR\_SPECII- numărul de specii de plante întâlnite în acea grădină.

Mulțimea entitate CASTELE\_ȘI\_CETATI dispune de o cheie primară ID, și doua atribute, SUPRAFATA și AN/SECOL-în care a apărut obiectivul.

În mulțimea entitate PESTERI exista 4 atribute: ID\_PESTERA(cheie primară), FORMARE- modul în care s-a format, ALTITUDINE, LUNGIME\_MAX- pe cați metrii se întinde peștera respectivă.

Mulțimea ADRESE desemnează unde se pot găsii obiectivele și are figurate următoarele atribute descriptive: STRADA, NUMAR și cheia primară ID\_ ADRESĂ.

În final, mulțimea rămasă PROGRAME\_DE\_VIZITA ajută utilizatorul să fie informat cu privire la orele în care poate să viziteze obiectivul. Atributele ce descriu această entitate sunt: ID\_PROGRAM(cheie primară), ZILE- care este programul în functie de zile, ORA\_DESCHIDERE și ORA\_INCHIDERE.

Între aceste mulțimi există diverse legături. Prima legatură *Apartin* se realizează între CONTINENTE și TARI. Aceasta arata ca un continent poate avea mai multe tari, însa o țară se poate află pe mai multe continente( cum este Turcia), deci este o legatură multi-multi. TARI-le *Au* unul sau mai multe ORASE,iar orașele au o singură tară căruia îi apartin, deci este o relație unu-multi. Asemanator, ORASE-le *Ofera* una sau mai multe CAZARI, iar o anumita cazare este oferită de un singur oraș, nu de mai multe(relație unu-multi) *.* În ORASE *Exista* mai multe OBIECTIVE turistice, dar un obiectiv nu poate exista în mai multe orașe, astfel avem tot o relație unu-multi.

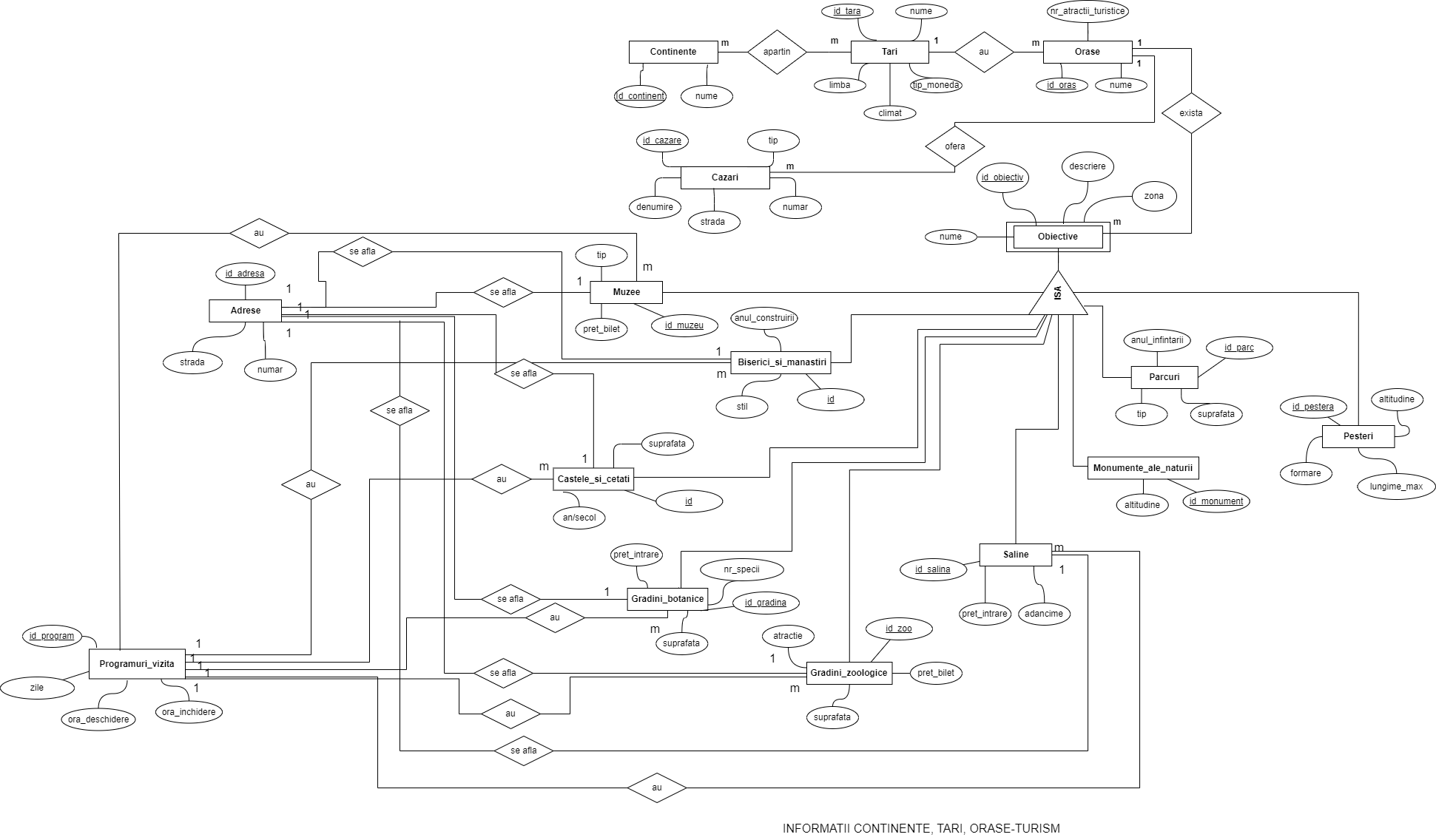
În cadrul sistemului, țările pot avea mai multe limbi oficiale, iar aceste limbi pot fi vorbite în mai multe țări, ceea ce reprezintă o relație multi-multi între ele. De asemenea, fiecare cazare are o adresă specifică la care se află, ceea ce implică o relație de tip unu-la-unu între cazări și adrese.

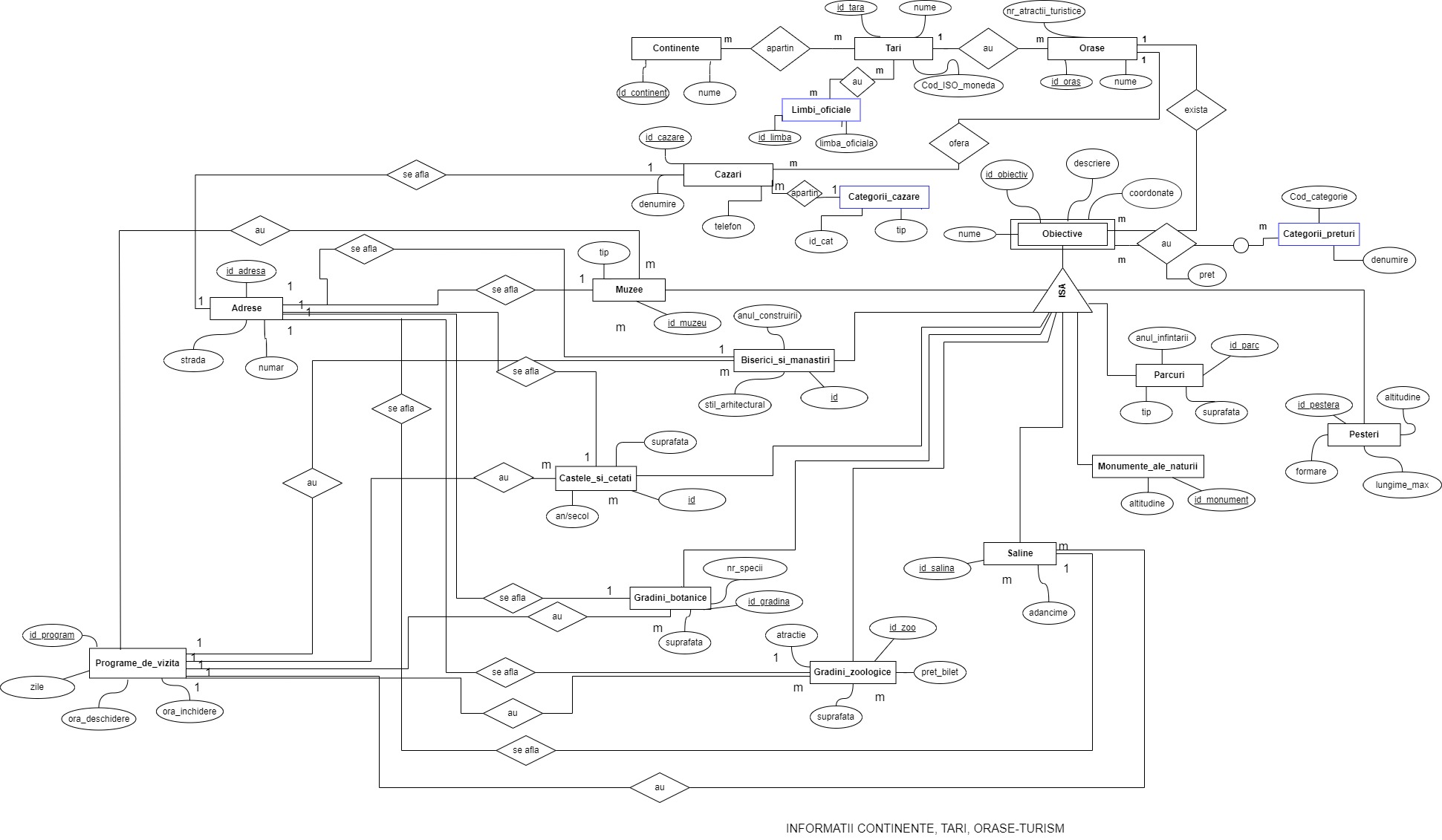
Pentru obiective, există posibilitatea ca acestea să aibă un preț pentru biletul de intrare, însă aceasta este o relație opțională multi-multi. Acest lucru înseamnă că un obiectiv poate avea mai multe prețuri pentru biletul de intrare și, de asemenea, există obiective care nu au niciun preț asociat.

Mai departe, OBIECTIVE-le sunt legate printr-o legatură de tip ISA de toate entitățile ce reprezintă obiective : MUZEE, BISERICI\_ȘI\_MANASTIRI, PARCURI, MONUMENTE\_ALE\_NATURII, SALINE, GRADINI\_ZOOLOGICE, GRADINI\_BOTANICE, CASTELE\_ȘI\_CETATI, PESTERI, ele moștenind astfel atributele OBIECTIVE-lor. Pe de alta parte, această mulțime entitate este incompletă deoarece într-un oraș există posibilitatea de a nu fi obiective.

În ceea ce privește ADRESE-le, există 6 legături de tip unu-unu cu urmatoarele entități: CAZARI, MUZEE, CASTELE\_ȘI\_CETATI, GRADINI\_ZOOLOGICE, GRADINI\_BOTANICE,SALINE, deoarece la fiecare mulțime entitate *Se află* o adresă unica și o singură adresă se află la o singură mulțime entitate. La PROGRAME\_DE\_VIZITA se află tot 5 legaturi cu aceleași entităti (MUZEE, CASTELE\_ȘI\_CETATI, GRADINI\_ZOOLOGICE, GRADINI\_BOTANICE,SALINE) , însa de tip unu-multi deoarece obiectivele *Au* un program, însa programurile pot să aiba asocierea catre mai multe obiective( pot avea același program).

Diagrama inițială:



Diagrama finală după normalizare

1. **Schema bazei de date și codul SQL**

Pentru fiecare mulțime entitate se creează o relație, care va conține în schema sa toate atributele mulțimii entitate respective, iar cheia mulțimii entitate va deveni cheia relației.

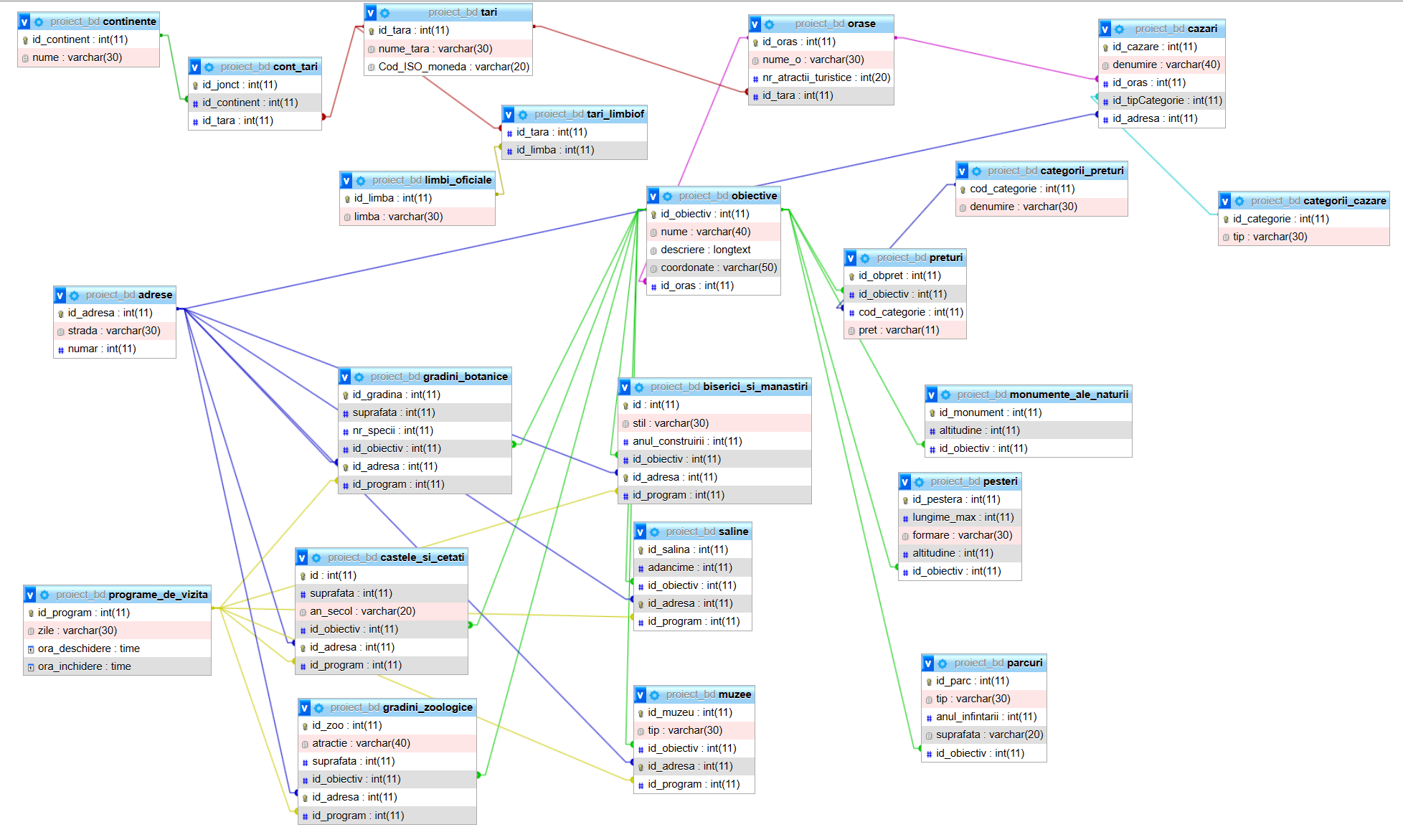
Schema relațională a bazei de date:

* CONTINENTE( id\_continent, nume)
* TARI (id\_tara, nume, cod\_ISO\_moneda)
* LIMBI\_OFICIALE(id\_limba, limba\_oficiala)
* ORASE (id\_oraș, nume, nr\_atractii\_turistice, *id\_tara*)
* CAZARI (id\_cazare, denumire, *id\_oraș, id\_adresa,id\_tipCategorie*)
* OBIECTIVE(id\_obiectiv, nume, descriere,coordonate, *id\_oraș*)
* CATEGORII\_PRETURI(cod\_categorie, denumire)
* MUZEE(id\_muzeu,tip,*id\_obiectiv,id\_adresă,id\_program*)
* BISERICI\_ȘI\_MANASTIRI(id, stil, anul\_construirii,*id\_obiectiv,id\_adresă,id\_prog*ram)
* PARCURI (id\_parc, tip, anul\_infintarii, suprafata, *id\_obiectiv*)
* MONUMENTE\_ALE\_NATURII(id\_monument, altitudine,*id\_obiect*)
* SALINE ( id\_salina,tip,*id\_obiectiv,id\_adresă,id\_program*)
* GRADINI\_ZOOLOGICE (id\_zoo, atractie, suprafata,*id\_obiectiv,id\_adresă,id\_program*)
* GRADINI\_BOTANICE (id\_gradina, suprafata, nr\_specii,*id\_obiectiv,id\_adresă,id\_program*)
* CASTELE\_ȘI\_CETATI (id, suprafata, an/secol,*id\_obiectiv,id\_adresă,id\_program*)
* PESTERI (id\_pestera, lungime\_max, formare,altitudine,*id\_obiectiv*)
* ADRESE(id\_adresă, strada, numar)
* PROGRAMURI\_VIZITA(id\_program, zile, ora\_deschidere, ora\_inchiere)

În schemele de mai sus, apar în italice atributele care au fost incluse pe post de chei străine(FK) în relații, pentru a reprezenta legăturile în care acestea sunt implicate.

Pentru legatură multi-multi dintre CONTINENTE și TARI, s-a creat o altă entitate numită CONT\_TARI cu atributele: id\_jonct(cheie primară), id\_continent (FK) și id\_tara(fk). Astfel, dacă dorim afișarea țărilor dintr-un anumit continent, se va folosi tabela intermediară CONT\_TARI.

Pentru legatura multi-multi dintre OBIECTIVE si CATEGORII\_PRETURI am creat tabela de jonctiune OBIECTIVE\_PRETURI cu atributele:id\_obpret(cheie primara), *id\_obiectiv(FK)*, *cod\_categorie(FK)*, pret.

 Mai jos se pot observa toate tabelele cu toate legăturile din baza de date:

Baze de date tranzactionale:

Baza de date utilizată este InnoDB, iar o tranzacție aplicată pe această bază de date este următoarea:

Start transaction;

Lock tables continente read;

Insert into continente( nume) values( ,’Africa’);

Select \* from continente;

Unlock tables;

CODUL MYSQL PENTRU CREAREA BAZEI DE DATE

|  |  |
| --- | --- |
| CREATE TABLE `continente` (  `id\_continent` INT NOT NULL AUTO\_INCREMENT ,  `nume` VARCHAR(30) NOT NULL ,  PRIMARY KEY (`id\_continent`)); | CREATE TABLE `monumente\_ale\_naturii` (  `id\_monument` INT NOT NULL AUTO\_INCREMENT ,   `altitudine` INT NOT NULL ,   `id\_obiectiv` INT NOT NULL ,  PRIMARY KEY (`id\_monument`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv)); |
| CREATE TABLE `tari` (  `id\_tara` INT NOT NULL AUTO\_INCREMENT ,  `nume` VARCHAR(30) NOT NULL , `clima` VARCHAR(20) NOT NULL,  `limba` VARCHAR(20) NOT NULL ,  `tip\_moneda` VARCHAR(20) NOT NULL ,  PRIMARY KEY (`id\_tara`)); | CREATE TABLE `gradini\_zoologice` (  `id\_zoo` INT NOT NULL AUTO\_INCREMENT ,  `pret\_bilet` INT NOT NULL ,  `atractie` VARCHAR(40) NOT NULL ,  `suprafata` INT NOT NULL ,   `id\_obiectiv` INT NOT NULL , `id\_adresa` INT NOT NULL UNIQUE,  `id\_program` INT NOT NULL ,   PRIMARY KEY (`id\_zoo`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv),  FOREIGN KEY (id\_adresa) REFERENCES adrese(id\_adresa),  FOREIGN KEY (id\_program) REFERENCES programuri\_vizita(id\_program)); |
| CREATE TABLE `cont\_tari` (  `id\_jonct` INT NOT NULL AUTO\_INCREMENT ,   `id\_continent` INT NOT NULL ,   `id\_tara` INT NOT NULL ,   PRIMARY KEY (`id\_jonct`),  FOREIGN KEY (id\_continent) REFERENCES continente(id\_continent),  FOREIGN KEY (id\_tara) REFERENCES tari(id\_tari)); | CREATE TABLE `gradini\_botanice` (  `id\_gradina` INT NOT NULL AUTO\_INCREMENT ,   `pret\_intrare` INT NOT NULL ,  `suprafata` INT NOT NULL ,  `nr\_specii` INT NOT NULL ,  `id\_obiectiv` INT NOT NULL, `id\_adresa` INT NOT NULL UNIQUE,  `id\_program` INT NOT NULL ,  PRIMARY KEY (`id\_gradina`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv),  FOREIGN KEY (id\_adresa) REFERENCES adrese(id\_adresa),  FOREIGN KEY (id\_program) REFERENCES programuri\_vizita(id\_program)); |
| CREATE TABLE`orase` (  `id\_oraș` INT NOT NULL AUTO\_INCREMENT ,   `nume` VARCHAR(30) NOT NULL ,  `nr\_atractii\_turistice` INT(20) NOT NULL ,  `id\_tara` INT NOT NULL ,   PRIMARY KEY (`id\_oraș`),  FOREIGN KEY (id\_tara) REFERENCES tari(id\_tara)); | CREATE TABLE `biserici\_și\_manastiri` (  `id` INT NOT NULL AUTO\_INCREMENT ,   `stil` VARCHAR(30) NOT NULL ,  `anul\_construirii` INT NOT NULL ,  `id\_obiectiv` INT NOT NULL ,  `id\_adresa` INT NOT NULL UNIQUE,  `id\_program` INT NOT NULL ,   PRIMARY KEY (`id`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv),  FOREIGN KEY (id\_adresa) REFERENCES adrese(id\_adresa),  FOREIGN KEY (id\_program) REFERENCES programuri\_vizita(id\_program)); |
| CREATE TABLE`cazari` (   `id\_cazare` INT NOT NULL AUTO\_INCREMENT ,   `denumire` VARCHAR(40) NOT NULL ,   `tip` VARCHAR(20)  NOT NULL , `strada` VARCHAR(40)  NOT NULL, `numar` INT NOT NULL ,  `id\_oras` INT NOT NULL ,  PRIMARY KEY (`id\_oraș`),  FOREIGN KEY (id\_oras) REFERENCES orase(id\_oras)); | CREATE TABLE `castele\_și\_cetati` (  `id` INT NOT NULL AUTO\_INCREMENT ,   `suprafata` INT NOT NULL ,  `an/secol` VARCHAR(20) NOT NULL ,  `id\_obiectiv` INT NOT NULL ,  `id\_adresa` INT NOT NULL UNIQUE,  `id\_program` INT NOT NULL ,  PRIMARY KEY (`id`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv),  FOREIGN KEY (id\_adresa) REFERENCES adrese(id\_adresa),  FOREIGN KEY (id\_program) REFERENCES programuri\_vizita(id\_program)); |
| CREATE TABLE`obiective` (  `id\_obiectiv` INT NOT NULL AUTO\_INCREMENT ,   `nume` VARCHAR(40) NOT NULL ,   `descriere` LONGTEXT  NOT NULL , `zona` VARCHAR(30)  NOT NULL, `id\_oras` INT NOT NULL ,   PRIMARY KEY (`id\_oras`),  FOREIGN KEY (id\_oras) REFERENCES orase(id\_oras)); | CREATE TABLE `muzee` (  `id\_muzeu` INT NOT NULL AUTO\_INCREMENT ,  `pret\_bilet` INT NOT NULL ,  `tip` VARCHAR(30) NOT NULL ,   `id\_obiectiv` INT NOT NUL,  `id\_adresa` INT NOT NULL UNIQUE,  `id\_program` INT NOT NULL,  PRIMARY KEY (`id\_muzeu`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv),  FOREIGN KEY (id\_adresa) REFERENCES adrese(id\_adresa),  FOREIGN KEY (id\_program) REFERENCES programuri\_vizita(id\_program)); |
| CREATE TABLE `parcuri` (  `id\_parc` INT NOT NULL AUTO\_INCREMENT ,  `tip` VARCHAR(30) NOT NULL ,   `anul\_infintarii` INT NOT NULL ,  `suprafata` INT NOT NULL ,  `id\_obiectiv` INT NOT NULL ,   PRIMARY KEY (`id\_parc`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv)); | CREATE TABLE `adrese` (  `id\_adresa` INT NOT NULL AUTO\_INCREMENT ,   `strada` INT NOT NULL ,   `numar` INT NOT NULL  ,  PRIMARY KEY (`id\_adresa`)); |
| CREATE TABLE `pesteri` (  `id\_pestera` INT NOT NULL AUTO\_INCREMENT ,  `formare` VARCHAR(40) NOT NULL ,  `lungime` INT NOT NULL ,  `altitudine` INT NOT NULL ,  `id\_obiectiv` INT NOT NULL ,  PRIMARY KEY (`id\_pestera`),  FOREIGN KEY (id\_obiectiv) REFERENCES obiective(id\_obiectiv)); | CREATE TABLE `program\_vizita` (  `id\_program` INT NOT NULL AUTO\_INCREMENT ,   `zile` VARCHAR(30) NOT NULL ,   `ora\_deschidere` TIME NOT NULL ,   `ora\_inchidere` TIME NOT NULL ,  PRIMARY KEY (`id\_program`)); |

**INSERAREA DE INFORMAȚII ÎN BAZĂ**

Mai jos se vor prezenta câteva comenzi prin care am adăugat informații în baza noastră.

|  |
| --- |
| INSERT INTO `continente` (`id\_continent`, `nume`) VALUES (NULL, 'Europa'), (NULL, 'Asia'), (NULL, 'America de Nord'), (NULL, 'America de Sud'); |
| INSERT INTO `tari` (`id\_tara`, `nume`, `limba`, `tip\_moneda`) VALUES (NULL, 'Romania', 'romana', 'RON'), (NULL, 'Turcia', 'turca', 'Lira turceasca'), (NULL, 'Statele Unite ale Americii', 'engleza', 'dolar'), (NULL, 'Thailanda', 'thailandeza', 'baht'); |
| INSERT INTO `orase` (`id\_oraș`, `nume`, `nr\_atractii\_turistice`, `id\_tara`) VALUES (NULL, 'Sinaia', '50', '1'), (NULL, 'Istambul', '71', '2'), (NULL, 'New York', '150', '3'); |
| INSERT INTO `cont\_tari` (`id\_jonct`, `id\_continent`, `id\_tara`) VALUES (NULL, '1', '1'), (NULL, '1', '2'), (NULL, '3', '3'), (NULL, '2', '2'), (NULL, '2', '4'); |
| INSERT INTO `cazari` (`id\_cazare`, `denumire`, `tip`, `strada`, `numar`, `id\_oraș`) VALUES (NULL, 'Hotel Sinaia', 'hotel', 'Bulevardul Carol I', '8', '1'), (NULL, 'Hotel Apollo Central Sibiu', 'hotel', 'Strada Nicolae Teclu', '14', '4'), (NULL, 'Hotel Continental Forum Sibiu', 'hotel', 'Piata Unirii', '10', '4'), (NULL, 'Central Park West Hostel', 'hostel', '111th St', '137', '3'), (NULL, 'Pod 51 Hotel', 'hotel', '51st S', '230', '3'), (NULL, 'La Vie Suits', 'hotel', 'Cihangir, Inkilap Sk', '2', '2'), (NULL, 'Royal Inci Hotel', 'hotel', 'Kartaltepe, Sadik Sk.', '8', '2'); |
| INSERT INTO `obiective` (`id\_obiectiv`, `descriere`, `zona`, `id\_oraș`) VALUES (NULL, 'Castelul Peles a fost construit la initiativa primului Rege al Romaniei, Carol I, în afara perimetrului comunei Podul Neagului, localitate cu o suprafata de 24 de km2 în anul 1874, an în care, din initiativa suveranului, comuna primeste numele de Sinaia. Un an mai tarziu, în centrul localitatii sunt construite primele case boieresti, iar în 1876 incepe construirea caii ferate Ploiesti – Predeal, care strabate și Sinaia. Concomitent, între anii 1873 și 1875 a fost edificata fundatia castelului Peles. Ceremonia de punere a pietrei de temelie a resedintei a avut loc într-un cadru festiv la 10/22 august 1875.', 'Sinaia', '1'), (NULL, 'De peste trei secole Manastirea Sinaia strajuieste pitoreasca vale a Prahovei, fiind situata în cadrul Masivului Bucegi, la poalele Muntelui Furnica, la altitudinea de 860 de metri, aproape de confluenta raului Prahova cu paraul Peles, la 122 km nord de Bucuresti și la 49 km sud de Brasov. Face parte din orasul Sinaia și are în imediata vecinatate vestitul Castel Peles, resedinta de vara a familiei regale romane între anii 1883-1947. ', 'Sinaia', '1')); |
| INSERT INTO `adrese` (`id\_adresă`, `strada`, `numar`) VALUES (NULL, 'Aleea Pelesului', '2'), (NULL, 'Str. Manastirii', '2'), (NULL, 'Strada Padurea Dumbrava', '16'), (NULL, 'Calea Dumbravii', '142'), (NULL, '5th Ave', '1000'), (NULL, 'Aleea Durgaului', '7'), (NULL, 'Strada Salinei', '1'), (NULL, 'Southern Blvd', '2300'), (NULL, 'Piri Reis, Tuzla', '297'), (NULL, 'Porumbacu de Sus', '630'), (NULL, 'Piata Mare', '4'), (NULL, 'Washington Ave', '990'), (NULL, 'Ayvansaray, Kuytu Sk.', '34086'); |
| INSERT INTO `program\_vizita` (`id\_program`, `zile`, `ora\_deschidere`, `ora\_inchidere`) VALUES (NULL, 'l-d', '08:00', '18:00'), (NULL, 'l-d', '09:30', '17:00'), (NULL, 'l-d', '09:30', '16:00'); |
| INSERT INTO `biserici\_și\_manastiri` (`id`, `stil`, `anul\_construirii`, `id\_obiectiv`, `id\_adresă`, `id\_program`) VALUES (NULL, 'bizantin', '1690', '2', '2', '2'), (NULL, 'bizantin', '1867', '20', '13', '3'); |
| INSERT INTO `castele\_și\_cetati` (`id`, `suprafata`, `an/secol`, `id\_obiectiv`, `id\_adresă`, `id\_program`) VALUES (NULL, '3200', '1914', '1', '1', '1'), (NULL, '7600', '2014', '12', '10', '8'); |
| INSERT INTO `monumente\_ale\_naturii` (`id\_monument`, `altitudine`, `id\_obiectiv`) VALUES (NULL, '452', '14'), (NULL, '450', '15'); |
| INSERT INTO `muzee` (`id\_muzeu`, `pret\_bilet`, `tip`, `id\_obiectiv`, `id\_adresă`, `id\_program`) VALUES (NULL, '35', 'etnomuzeu', '3', '3', '3'), (NULL, '37', 'muzeu de arta', '5', '5', '4'), (NULL, '35', 'muzeu de arta', '13', '11', '5'); |
| INSERT INTO `parcuri` (`id\_parc`, `tip`, `anul\_infintarii`, `suprafata`, `id\_obiectiv`) VALUES (NULL, 'rezervatie naturala', '2000', '933', '4'), (NULL, 'urban', '1880', '1', '19'), (NULL, 'urban', '1859', '315', '18'); |
| INSERT INTO `pesteri` (`id\_pestera`, `lungime\_max`, `formare`, `altitudine`, `id\_obiectiv`) VALUES (NULL, '46', 'calcar', '1020', '7'), (NULL, '167', 'calcar', '1021', '17'); |

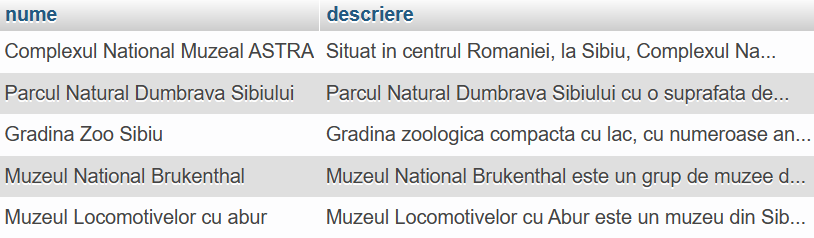
1. **Operații asupra bazei de date**

În aceasta secțiune vom prezenta un set de interogari și răspunsurile pentru ele, alături de formula în algerbă relatională pentru fiecare operație în parte.

**Interogarea 1.** Să se afișeze numele și descrierea obiectivelor aflate în zona Sibiu.

SELECT nume,descriere FROM `obiective` WHERE zona='Sibiu';

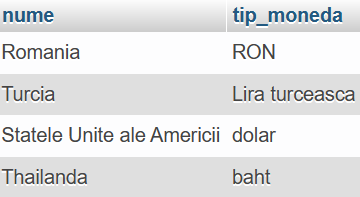
**∏**nume,descriere (σ zona='Sibiu' (obiective))



**Interogarea 2**. Să se afișeze numele și tipul de monedă cu care se plătește în fiecare tară.

SELECT DISTINCT nume,tip\_moneda FROM `tari`;

**∏**(nume,tip\_moneda)(tari)



 **Interogare 3.** Să se afișeze pesterile si gradinile zoo din Brașov.

select obiective.nume as pesteri\_si\_gradini from obiective join pesteri where obiective.id\_obiectiv=pesteri.id\_obiectiv and obiective.id\_oras=7 UNION select obiective.nume from obiective join gradini\_zoologice where obiective.id\_obiectiv=gradini\_zoologice.id\_obiectiv and obiective.id\_oras=7;

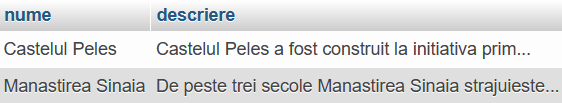
**∏**obiective.nume (σ obiective.id\_oras=7 (obiective ⨝ pesteri) ) **∪ ∏**obiective.nume (σ obiective.id\_oras=7 (obiective ⨝ gradini\_zoologice))

**Interogarea 4**. Să se afișeze numele tuturor obiectivelor din Sinaia ,fără peșteri.

select o.nume,o.descriere from obiective as o join orase as oa on o.id\_oras=oa.id\_oras

where not EXISTS(select \*from pesteri where o.id\_obiectiv= pesteri.id\_obiectiv) and oa.nume\_o='Sinaia';

**∏** (¬∃(pesteri) (o.id\_obiectiv = pesteri.id\_obiectiv) ∧ oa.nume\_o = 'Sinaia') (∏ (o.nume, o.descriere) (obiective ⨝ o.id\_oras = oa.id\_oras orase))



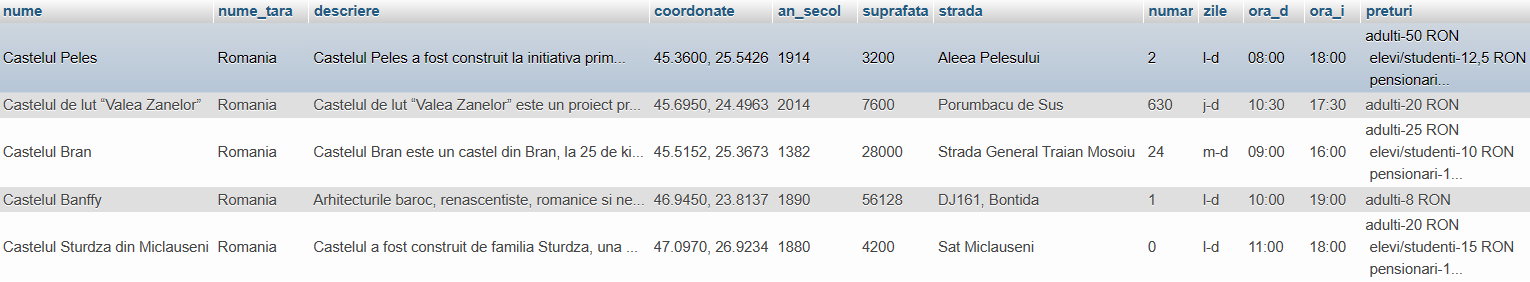
**Interogarea 5.** Să se afișeze denumirea cazărilor din orașe.

select orase.nume as oraș, cazari.denumire as denumire\_cazare from orase join cazari where orase.id\_oraș=cazari.id\_oraș;

**∏**oraș=oraș.nume, denumire\_cazare=cazari.denumire (orase**⨝** cazari.id\_oraș=orase.id\_oraș)

**Interogare 6.** Să se afișeze castelele și cetățile din România, împreună cu prețul de intrare și programul de vizită.

SELECT o.nume,t.nume\_tara, o.descriere, o.coordonate,c.an\_secol, c.suprafata ,a.strada, a.numar,p.zile, TIME\_FORMAT(p.ora\_deschidere, '%H:%i') as ora\_d, TIME\_FORMAT(p.ora\_inchidere, '%H:%i') as ora\_i, IFNULL(GROUP\_CONCAT(cp.denumire, '-', pr.pret ORDER BY cp.denumire SEPARATOR '\n '), 'GRATUIT') as preturi FROM castele\_si\_cetati as c Join obiective as o on o.id\_obiectiv=c.id\_obiectiv join orase as os on o.id\_oras=os.id\_oras join tari as t on t.id\_tara=os.id\_tara and t.nume\_tara='Romania' join adrese as a on a.id\_adresa=c.id\_adresa join programe\_de\_vizita as p on p.id\_program=c.id\_program LEFT JOIN preturi as pr on pr.id\_obiectiv = o.id\_obiectiv LEFT JOIN categorii\_preturi as cp on cp.cod\_categorie = pr.cod\_categorie GROUP BY o.id\_obiectiv;

**∏**o.nume, t.nume\_tara, o.descriere, o.coordonate, c.an\_secol, c.suprafata, a.strada, a.numar, p.zile, TIME\_FORMAT(p.ora\_deschidere, '%H:%i') as ora\_d, TIME\_FORMAT(p.ora\_inchidere, '%H:%i') as ora\_i, IFNULL(σ t.nume\_tara='Romania' (castele\_si\_cetati ⨝ obiective ⨝ orase ⨝ tari ⨝ adrese ⨝ programe\_de\_vizita ⨝ (preturi ⨝ categorii\_preturi)), 'GRATUIT')

1. Bibliografie

* <https://dev.mysql.com/doc/refman/5.7/en/data-types.html>
* <https://dev.mysql.com/doc/refman/8.0/en/select.html>
* <https://dev.mysql.com/doc/refman/5.7/en/insert.html>
* <https://dev.mysql.com/doc/refman/5.7/en/>