

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-Legătură.....	
3. Schema bazei de date și codul SQL	
4. Operații asupra bazei de date.....	
5. Bibliografie.....	

1. 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 funcție de continente, țări și orașe.

În mod traditional, oamenii doritori de a vizita și de a cunoaște mai de aproape ce îi înconjoară, se informau cu privire la obiectivele turistice din materiale precum reviste, cărți, ziare. De obicei, informațiile erau în număr mic, iar pentru a planui o călătorie în scopul vizitării, persoanele își 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 anumită 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 după diverse criterii:
 - a) După continentul/țara/orașul în care se dorește să se viziteze;
 - b) După tipul obiectivelor turistice (saline, muzee, peșteri, castele, grădini zoo, grădini botanice etc);
- Să fie disponibilă online.

2. 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 attribute: ID_CONTINENT(cheie primară)- prin care se identifică unic fiecare continent , NUME.

Mulțimea entitate TARI stochează informații utile pentru utilizatori precum attributele NUME- denumirea țării, COD_ISO_MONEDA- codul monedei ce se folosește în acea țară, *LIMBA- limbile oficiale vorbite, cheia primară ID_TARA - desemnează toate țările din care vrem să aflăm informații legate de turism, *LIMBA- limbile oficiale vorbite în țări.

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 primară (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 attributele: ID_ORAȘ(cheie primară), NUME- numele tarii, NR_ATTRACTII_TURISTICE- totalul de atracții din acea țară.

Entitatea CAZARI cuprinde informații despre diferite hoteluri, apartamente, pensiuni, în alte cuvinte, ajută la alegerea unei cazări. Attributele 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 attributele marcate cu *. Astfel, am creat entitatea CATEGORII_CAZARE, care are următoarele attribute: 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 билетelor 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 attributele: ID_MUZEU(cheie primară), TIP- dacă 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 attribute 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. Attributele specifice entității sunt: ID_MONUMENT(cheie primară), ALTITUDINE- cea la care se găsește obiectivul.

Entității SALINE îi corespund attributele: ID_SALINA(cheie primară), ADANCIME- cât de adâncă este salina.

GRADINI_ZOOLOGICE conține attributele: ID_ZOO(cheie primară), SUPRAFATA, ATRACTIE- cel mai apreciat animal.

De asemenea, entitatea GRADINI_BOTANICE are o cheie primară ID_GRADINA, urmată de attribute 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 attribute, SUPRAFATA și AN/SECOL-în care a apărut obiectivul.

În mulțimea entitate PESTERI exista 4 attribute: ID_PESTERA(cheie primară), FORMARE- modul în care s-a format, ALTITUDINE, LUNGIME_MAX- pe câți metrii se întinde peștera respectivă.

Mulțimea ADRESE desemnează unde se pot găsi obiectivele și are figurate următoarele attribute 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. Attributele ce descriu această entitate sunt: ID_PROGRAM(cheie primară), ZILE- care este programul în funcție de zile, ORA_DESCHIDERE și ORA_INCHIDERE.

Între aceste mulțimi există diverse legături. Prima legătură *Apartin* se realizează între CONTINENTE și TARI. Aceasta arată că un continent poate avea mai multe țări, însă o țară se poate afla pe mai multe continente (cum este Turcia), deci este o legătură multi-multi. TARI-le *Au* unul sau mai multe ORASE, iar orașele au o singură țară căruia îi aparțin, deci este o relație unu-multi. Asemănător, ORASE-le *Oferă* una sau mai multe CAZARI, iar o anumită cazare este oferită de un singur oraș, nu de mai multe (relație unu-multi). În ORASE *Există* 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 legătură 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 altă 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 următoarele entități: CAZARI, MUZEE, CASTELE_ȘI_CETATI, GRADINI_ZOOLOGICE, GRADINI_BOTANICE, SALINE, deoarece la fiecare mulțime entitate *Se află* o adresă unică și o singură adresă se află la o singură mulțime entitate. La PROGRAME_DE_VIZITA se află tot 5 legături cu aceleași entități (MUZEE, CASTELE_ȘI_CETATI, GRADINI_ZOOLOGICE, GRADINI_BOTANICE, SALINE), însă de tip unu-multi deoarece obiectivele *Au* un program, însă programurile pot să aibă asocierea către mai multe obiective (pot avea același program).

Diagrama inițială:

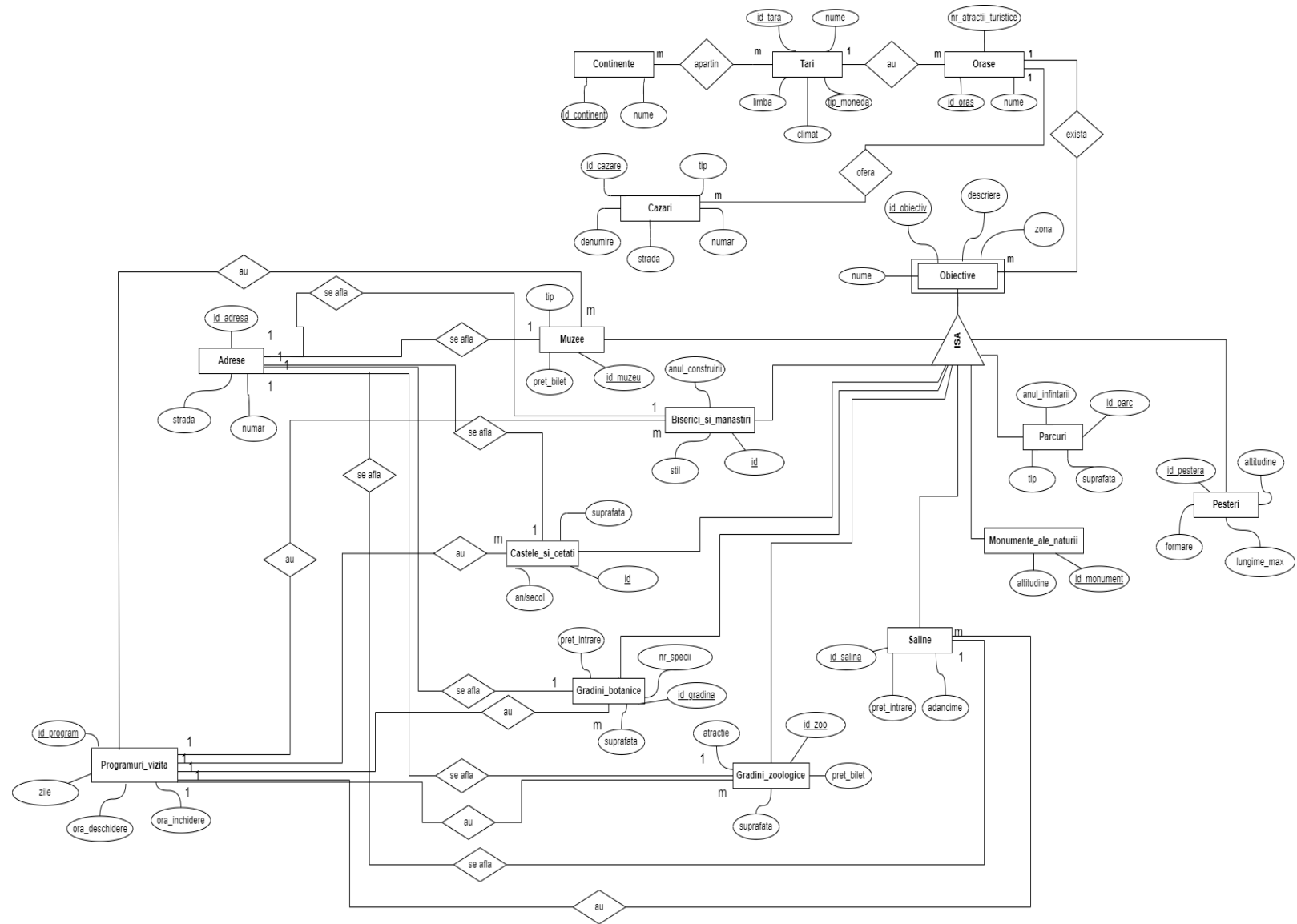
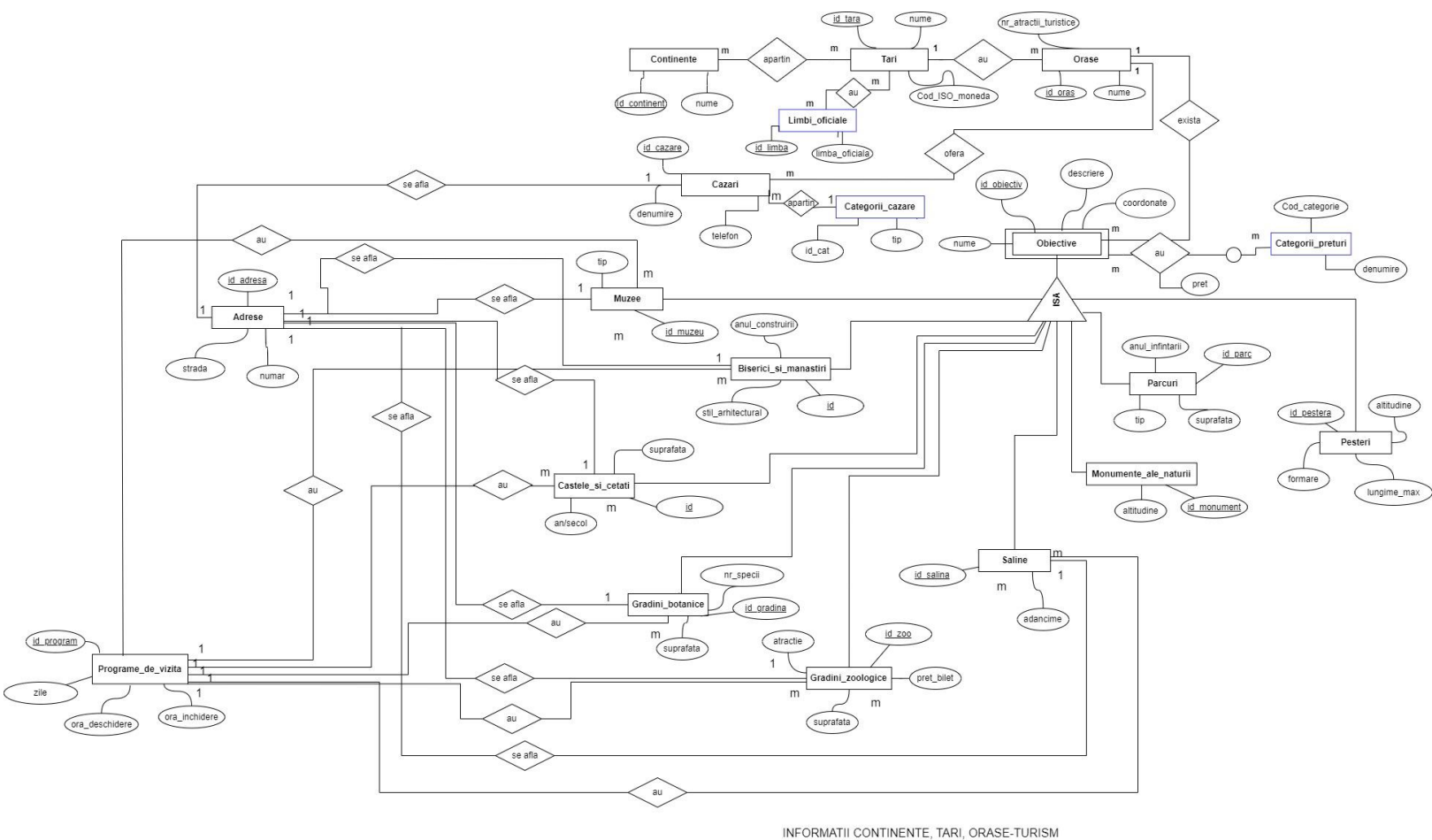


Diagrama finală după normalizare



3. 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_oras, nume, nr_atractii_turistice, id_tara)

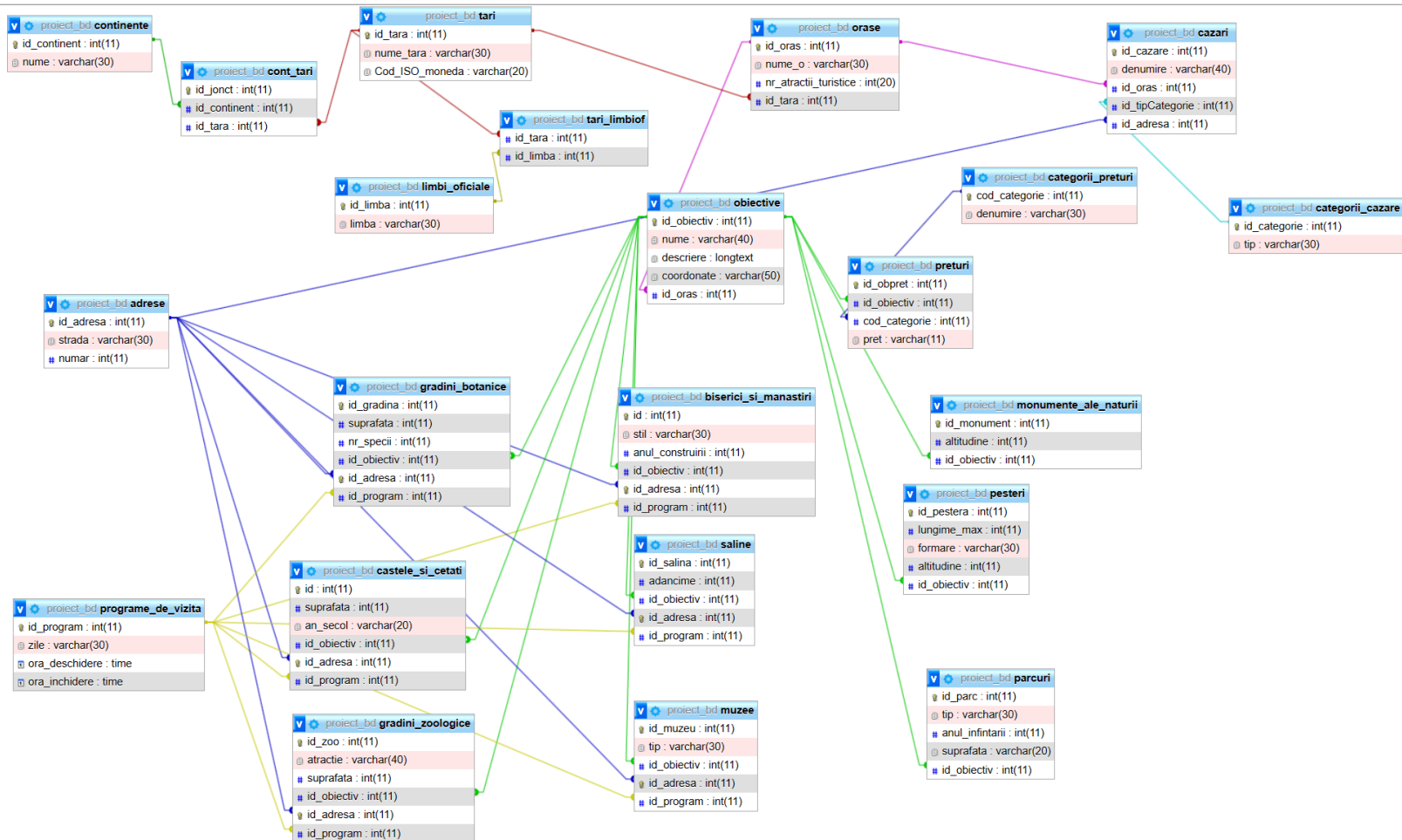
- CAZARI (id_cazare, denumire, *id_oras*, *id_adresa*, *id_tipCategorie*)
- OBIECTIVE(id_obiectiv, nume, descriere, coordonate, *id_oras*)
- CATEGORII_PRETURI(cod_categorie, denumire)
- MUZEE(id_muzeu, tip, *id_obiectiv*, *id_adresa*, *id_program*)
- BISERICI_SI_MANASTIRI(id, stil, anul_construirii, *id_obiectiv*, *id_adresa*, *id_program*)
- PARCURI (id_parc, tip, anul_infintarii, suprafata, *id_obiectiv*)
- MONUMENTE_ALE_NATURII(id_monument, altitudine, *id_obiect*)
- SALINE (id_salina, tip, *id_obiectiv*, *id_adresa*, *id_program*)
- GRADINI_ZOOLOGICE (id_zoo, atractie, suprafata, *id_obiectiv*, *id_adresa*, *id_program*)
- GRADINI_BOTANICE (id_gradina, suprafata, nr_specii, *id_obiectiv*, *id_adresa*, *id_program*)
- CASTELE_SI_CETATI (id, suprafata, an/secol, *id_obiectiv*, *id_adresa*, *id_program*)
- PESTERI (id_pestera, lungime_max, formare, altitudine, *id_obiectiv*)
- ADRESE(id_adresa, strada, numar)
- PROGRAMURI_VIZITA(id_program, zile, ora_deschidere, ora_inchidere)

În schemele de mai sus, apar în *italice* attributele 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 attributele: 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 attributele: 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

<pre>CREATE TABLE `continente` (`id_continent` INT NOT NULL AUTO_INCREMENT , `nume` VARCHAR(30) NOT NULL , PRIMARY KEY (`id_continent`));</pre>	<pre>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));</pre>
<pre>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`));</pre>	<pre>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));</pre>
<pre>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));</pre>	<pre>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));</pre>
<pre>CREATE TABLE `orase` (`id_oras` 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_oras`), FOREIGN KEY (id_tara) REFERENCES tari(id_tara));</pre>	<pre>CREATE TABLE `biserici_si_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),</pre>

	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_oras`), FOREIGN KEY (id_oras) REFERENCES orase(id_oras));	CREATE TABLE `castele_si_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 NULL , `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 ,	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`));

PRIMARY KEY (`id_pestera`), FOREIGN KEY (id_obiectiv) REFERENCES objective(id_obiectiv));	
---	--

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_oras`, `nume`, `nr_atractii_turistice`, `id_tara`) VALUES (NULL, 'Sinaia', '50', '1'), (NULL, 'Istanbul', '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_oras`) 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_oras`) 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_adresa`, `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, '1-d', '08:00', '18:00'), (NULL, '1-d', '09:30', '17:00'), (NULL, '1-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');
```

4. 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';
```

Π nume,descriere (σ zona='Sibiu' (obiective))

nume	descriere
Complexul National Muzeal ASTRA	Situat in centrul Romaniei, la Sibiu, Complexul Na...
Parcul Natural Dumbrava Sibiului	Parcul Natural Dumbrava Sibiului cu o suprafata de...
Gradina Zoo Sibiu	Gradina zoologica compacta cu lac, cu numeroase an...
Muzeul National Brukenthal	Muzeul National Brukenthal este un grup de muzee d...
Muzeul Locomotivelor cu abur	Muzeul Locomotivelor cu Abur este un muzeu din Sib...

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

```
SELECT DISTINCT nume,tip_moneda FROM `tari`;
```

$\Pi_{(nume,tip_moneda)}(tari)$

nume	tip_moneda
Romania	RON
Turcia	Lira turceasca
Statele Unite ale Americii	dolar
Thailanda	baht

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;
```

pesteri_si_gradini

Pestera laptelui

Pestera Rasnoavei

Pestera Liliiecilor

Gradina zoo Brasov

$\Pi_{\text{obiective.ume}} (\sigma_{\text{obiective.id_oras}=7} (\text{obiective} \bowtie \text{pesteri})) \cup \Pi_{\text{obiective.ume}} (\sigma_{\text{obiective.id_oras}=7} (\text{obiective} \bowtie \text{gradini_zoologice}))$

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

`select o.ume,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.ume_o='Sinaia';`

$\Pi_{(\neg \exists (\text{pesteri}) (o.\text{id_obiectiv} = \text{pesteri}.\text{id_obiectiv}) \wedge oa.\text{ume_o} = 'Sinaia'))} (\Pi_{(o.\text{ume}, o.\text{descriere})} (\text{obiective} \bowtie o.\text{id_oras} = oa.\text{id_oras} \text{ orase}))$

ume	descriere
Castelul Peles	Castelul Peles a fost construit la initiativa prim...
Manastirea Sinaia	De peste trei secole Manastirea Sinaia strajuieste...

Interogarea 5. Să se afișeze denumirea cazărilor din orașe.

`select orase.ume as oraș, cazari.denumire as denumire_cazare from orase join cazari where orase.id_oras=cazari.id_oras;`

$\Pi_{\text{oraș=oraș.ume, denumire_cazare=cazari.denumire}} (\text{orase} \bowtie \text{cazari}.\text{id_oras} = \text{orase}.\text{id_oras})$

oras	denumire_cazare
Sinaia	Hotel Sinaia
Sibiu	Hotel Apollo Central Sibiu
Sibiu	Hotel Continental Forum Sibiu
New York	Central Park West Hostel
New York	Pod 51 Hotel
Istanbul	La Vie Suits
Istanbul	Royal Inci Hotel

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;
```

Πo.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')

nume	nume_tara	descriere	coordonate	an_secol	suprafata	strada	numar	zile	ora_d	ora_i	preturi
Castelul Peles	Romania	Castelul Peles a fost construit la initiativa prim...	45.3600, 25.5426	1914	3200	Aleea Pelesului	2	l-d	08:00	18:00	adulti-50 RON elevi/studenti-12,5 RON pensionari...
Castelul de lut "Valea Zanelor"	Romania	Castelul de lut "Valea Zanelor" este un proiect pr...	45.6950, 24.4963	2014	7600	Porumbacu de Sus	630	j-d	10:30	17:30	adulti-20 RON
Castelul Bran	Romania	Castelul Bran este un castel din Bran, la 25 de ki...	45.5152, 25.3673	1382	28000	Strada General Traian Mosoiu	24	m-d	09:00	16:00	adulti-25 RON elevi/studenti-10 RON pensionari-1...
Castelul Banffy	Romania	Arhitecturile baroc, renascentiste, romanice si ne...	46.9450, 23.8137	1890	56128	DJ161, Bontida	1	l-d	10:00	19:00	adulti-8 RON
Castelul Sturdza din Miclauseni	Romania	Castelul a fost construit de familia Sturdza, una ...	47.0970, 26.9234	1880	4200	Sat Miclauseni	0	l-d	11:00	18:00	adulti-20 RON elevi/studenti-15 RON pensionari-1...

5. 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/>