Descriere

În cadrul acestui proiect, ne concentrăm pe crearea unei baze de date pentru o aplicație destinată gestionării unei agenții turistice care oferă tururi ghidate. Această agenție organizează tururi către diferite destinații, oferind servicii de rezervare, ghidare și gestionarea plăților. Scopul bazei de date este să stocheze informațiile referitoare la turiști, ghizi, tururi, plecări, destinații, rezervări și plăți, asigurând integritatea și validitatea datelor.s

**Etapa de proiectare**

A screenshot of a computer

Description automatically generated

Tabelul **Turisti**, care conține detalii personale despre turiști, cum ar fi numele, prenumele, emailul, telefonul și CNP-ul. Tabelul include constrângeri precum NOT NULL pentru coloane precum nume, prenume și email, asigurând că aceste informații sunt obligatorii pentru fiecare turist. Colona **cnp** are constrângerea UNIQUE, deoarece fiecare turist trebuie să aibă un CNP unic. În plus, sexul turistului este limitat de o constrângere CHECK la valorile acceptate, 'M' sau 'F', pentru a se asigura coerența datelor.

A screenshot of a computer

Description automatically generated

Tabelul **Ghid** stochează informații despre ghizii care conduc tururile. Coloanele **nume\_ghid**, **prenume\_ghid**, și **email\_ghid** sunt esențiale, marcate ca NOT NULL, iar emailul este, de asemenea, unic, pentru a evita înregistrările duplicate. Telefonul ghidului este permis să fie NULL, deoarece nu toți ghizii pot avea un telefon asociat. Astfel, acest tabel asigură integritatea datelor legate de ghizi și disponibilitatea informațiilor de contact.

A screenshot of a computer

Description automatically generated

**Destinatie** este un tabel care gestionează detalii despre destinațiile turistice, incluzând numele, tipul și ora sosirii. În timp ce numele și ora sosirii sunt obligatorii, tipul destinației (cum ar fi destinație culturală, naturală etc.) poate fi lăsat NULL, dacă nu este specificat. Acest lucru permite o flexibilitate în descrierea destinațiilor.

A screenshot of a computer

Description automatically generated

Tabelul **Plecari** definește informațiile legate de locul și ora plecării fiecărui tur. **loc\_plecare** și **ora\_plecare** sunt NOT NULL, deoarece sunt esențiale pentru planificarea logistică a fiecărui tur. Totuși, coloana **observatii\_plecare** poate fi NULL, deoarece unele plecări ar putea să nu necesite comentarii suplimentare. Astfel, baza de date asigură stocarea informațiilor necesare pentru gestionarea eficientă a itinerariilor de plecare.

A screenshot of a computer

Description automatically generated

Unul dintre cele mai importante tabele este **Tur**, care conține detalii despre fiecare tur organizat de agenție. Acest tabel include numele turului, durata și numărul de locuri disponibile. Coloana **locuri\_disponibile** are o constrângere CHECK pentru a garanta că numărul locurilor disponibile este întotdeauna egal sau mai mare decât zero, prevenind erori în calcularea locurilor rămase. De asemenea, acest tabel se leagă de **ghid**, **plecare** și **destinatie** prin chei străine, asigurându-se astfel că fiecare tur are un ghid, un loc și o oră de plecare, precum și o destinație bine definite.

A screenshot of a computer

Description automatically generated

Tabelul **Rezervare** înregistrează detaliile fiecărei rezervări făcute de turiști. Acest tabel conține referințe la **turisti**, **tururi** și **plăți**, prin chei străine. Coloanele **data\_rezervare** și **ora\_rezervare** sunt esențiale pentru a ști exact când a fost făcută fiecare rezervare, oferind o trasabilitate completă a activităților.

A screenshot of a computer

Description automatically generated

**Plata** este tabelul responsabil pentru gestionarea tranzacțiilor financiare, stocând informații precum data plății, suma și metoda de plată. Coloana **suma** este obligatorie și are o constrângere CHECK care se asigură că valoarea este întotdeauna mai mare decât zero. **metoda\_plata** poate fi lăsată NULL, în cazul în care informațiile despre metoda de plată nu sunt imediat disponibile.

A screenshot of a computer

Description automatically generated

**Constrângeri de integritate:**

**Chei primare**:

Fiecare tabel are o cheie primară pentru a identifica în mod unic fiecare înregistrare. Cheile primare sunt definite în tabelele Turisti, Ghid, Destinatie, Plecare, Tur, Rezervare și Plata.

**Chei străine**:

Cheile străine asigură că fiecare înregistrare dintr-un tabel dependent este legată de o înregistrare validă dintr-un alt tabel:

* + - id\_ghid în Tur referă id\_ghid din Ghid.
    - id\_plecare în Tur referă id\_plecare din Plecare.
    - id\_destinatie în Tur referă id\_destinatie din Destinatie.
    - id\_plata în Rezervare referă id\_plata din Plata.

Aceste chei străine asigură integritatea referențială: o înregistrare într-un tabel dependent nu poate exista fără o înregistrare validă în tabelul de referință.

**Constrângeri NOT NULL**:

Constrângerile NOT NULL sunt utilizate pentru a impune existența unor valori în coloanele critice:Coloanele care reprezintă chei primare sau străine trebuie să fie întotdeauna populate cu valori valide.De asemenea, alte coloane esențiale, cum ar fi nume, email, durata, ora\_plecare, sunt NOT NULL pentru a preveni lipsa datelor critice.

**Constrângeri UNIQUE**:

Pentru a preveni duplicatele, coloana **cnp** în tabelul Turisti și **email** în Ghid au constrângerea UNIQUE. Aceste constrângeri asigură că fiecare CNP sau email este unic în baza de date.

**Constrângeri CHECK**:

Constrângerea CHECK este utilizată pentru a valida valorile inserate în anumite coloane:În tabelul Turisti, coloana **sex** are o constrângere CHECK pentru a permite doar valorile 'M' sau 'F'.În tabelul **Tur**, coloana **locuri\_disponibile** are o constrângere CHECK (locuri\_disponibile >= 0) pentru a asigura că numărul locurilor disponibile nu poate fi negativ. În tabelul **Plata**, coloana **suma** are o constrângere CHECK (suma > 0) pentru a garanta că plățile sunt întotdeauna pozitive.

Tipuri de relații:

**(One-to-Many)**:

Ghid → Tur: Un ghid poate conduce mai multe tururi.

Plecare → Tur: Un loc de plecare poate fi utilizat de mai multe tururi.

Destinație → Tur: O destinație poate fi vizitată de mai multe tururi.

Plata → Rezervare: O plată poate fi asociat

ă cu mai multe rezervări

**Turisti** ↔ **Rezervare** ↔ **Tur**

( **Many-to-many**):

Fiecare turist poate face mai multe rezervări pentru mai multe tururi.

Fiecare tur poate avea mai multe rezervări pentru turiști diferiți.

**Cheile străine** id\_turist și id\_tur din tabelul **Rezervare** leagă turiștii și tururile.

**Chei străine în tabelul Rezervare**:

**id\_turist** este o cheie străină care face referire la **id\_turist** din tabelul Turisti.

**id\_tur** este o cheie străină care face referire la **id\_tur** din tabelul Tur.

Aceste chei străine asigură integritatea referențială și leagă corect turiștii de tururi.

**Cheie compusă în tabelul Rezervare**:

O **cheie primară compusă** (sau o constrângere UNIQUE) formată din **id\_turist** și **id\_tur** poate fi utilizată pentru a preveni duplicatele, adică pentru a ne asigura că un turist poate rezerva un tur o singură dată. Astfel,**cheia compusă** va fi (id\_turist, id\_tur)

**Functionarea aplicatiei**

In cadrul aplicatiei se pot efectua operatii precum afisarea, adaugarea, stergere si modificarea datelor din baza noastra de date.

Operatiile de inserare , stergere si modificare se efectueaza in tabelul in care sunt inregistrati turistii si plecarile din locatiile in care incep tururile.

Aici o sa prezint operatiile pentru tabelul de turisti.

Operatia de inserare: INSERT INTO Turisti (nume, prenume, email,sex, telefon,cnp) VALUES (:nume, :prenume, :email,:sex, :telefon,:cnp)

Operatia de modificare: UPDATE Turisti

SET nume = :nume, prenume = :prenume, email = :email, telefon = :telefon,

sex = :sex, cnp = :cnp

WHERE id\_turist = :id\_turist

Operatia de stergere: DELETE FROM Turisti WHERE id\_turist = :id\_turist

De asemenea se pot efectua o serie de interogari simple si complexe,in care ne putem alege tipul de interogare care dorim sa actioneze pentru a afisa rezultatul.

Toate aceste operatii sunt posibile prin conectarea bazei noastre de date la un server care stocheaza datele si le actualizeaza conform dorintelor noastre.Conectarea la server se realizeaza astfel:

# Configurația pentru conexiune

conf = {

'host': "mysql-cca70b9-cnaic-44df.f.aivencloud.com",

'port': '23157',

'user': "avnadmin",

'password': "AVNS\_cUJm7IhN4qNPcWKWWKP",

'database': "defaultdb"

}

Dupa ce putem avea stocata baza de date intr-un mediu SQL apoi interactionata prin intermediul serverului, trebuie realizata o interfata care sa afiseze functionalitatile aplicatiei.

**Interogarile simple JOIN**

Această interogare arata numele tururilor și informațiile despre ghizii care le însoțesc

SELECT Tr.nume\_tur AS "Tour Name", G.nume\_ghid AS "Guide Name", G.prenume\_ghid AS "Guide Surname"

FROM Tur Tr

JOIN Ghid G ON Tr.id\_ghid = G.id\_ghid

Această interogare afiseaza detalii despre plățile turiștilor, specificând numele turistului și metoda de plată utilizată.

SELECT DISTINCT

t.nume,

t.prenume,

p.metoda\_platii

FROM Turisti t

JOIN Rezervare r ON t.id\_turist = r.id\_turist

JOIN Plate p ON r.id\_plata = p.id\_plata

ORDER BY t.nume, t.prenume;

Această interogare cu paramtru variabil returnează tururile in functie de perioada care dorim sa se desfasoare:

SELECT D.nume\_destinatie AS "Destination Name",

Tr.nume\_tur AS "Tour Name",

Tr.durata AS "Duration"

FROM Destinatie D

JOIN Tur Tr ON D.id\_destinatie = Tr.id\_destinatie

WHERE Tr.durata >= :min\_duration

ORDER BY Tr.durata

Această interogare afișează programul plecărilor pentru tururi, incluzând ora și locația plecării.

SELECT Tr.nume\_tur AS "Tour Name", P.ora\_plecare AS "Departure Time", P.loc\_plecare AS "Departure Location"

FROM Tur Tr

JOIN Plecare P ON Tr.id\_plecare = P.id\_plecare

Această interogare afișează totalul plăților realizate pentru fiecare tur.

SELECT Tr.nume\_tur AS "Tour Name", SUM(P.suma) AS "Total Payments"

FROM Tur Tr

JOIN Rezervare R ON Tr.id\_tur = R.id\_tur

JOIN Plate P ON R.id\_plata = P.id\_plata

GROUP BY Tr.nume\_tur

Această interogare afișează destinațiile populare pe baza numărului de rezervări făcute.

SELECT d.nume\_destinatie, COUNT(r.id\_plata) AS total\_rezervari

FROM Destinatie d

JOIN Tur t ON d.id\_destinatie = t.id\_destinatie

JOIN Rezervare r ON t.id\_tur = r.id\_tur

GROUP BY d.nume\_destinatie

ORDER BY total\_rezervari DESC;

**Interogari complexe**

Această interogare afiseaza informații despre tururile eficiente, adică tururile care generează un venit mai mare pe zi decât media tuturor tururilor. Se face o subinterogare pentru a calcula venitul mediu pe zi pentru toate tururile, iar tururile care au venituri pe zi mai mari decât această medie sunt afișate.

SELECT

t.nume\_tur,

t.durata,

COUNT(r.id\_rezervare) as numar\_rezervari,

SUM(p.suma) as venit\_total,

SUM(p.suma) / t.durata as venit\_per\_zi,

AVG(r.nr\_locuri\_rezervate) as medie\_locuri\_per\_rezervare

FROM Tur t

JOIN Rezervare r ON t.id\_tur = r.id\_tur

JOIN Plate p ON r.id\_plata = p.id\_plata

GROUP BY t.id\_tur, t.nume\_tur, t.durata

HAVING (SUM(p.suma) / t.durata) > (

SELECT AVG(venit\_per\_zi)

FROM (

SELECT SUM(p2.suma) / t2.durata as venit\_per\_zi

FROM Tur t2

JOIN Rezervare r2 ON t2.id\_tur = r2.id\_tur

JOIN Plate p2 ON r2.id\_plata = p2.id\_plata

GROUP BY t2.id\_tur, t2.durata

) sub

)

ORDER BY venit\_per\_zi DESC;"""

Această interogare selectează ghizii care au generat cele mai mari venituri din tururi. Interogarea include o subinterogare pentru a calcula venitul mediu per ghid, iar ghizii care au generat un venit mai mare decât media sunt incluși în rezultat.

SELECT g.nume\_ghid, g.prenume\_ghid,

SUM(p.suma) as venit\_total

FROM Ghid g

JOIN Tur t ON g.id\_ghid = t.id\_ghid

JOIN Rezervare r ON t.id\_tur = r.id\_tur

JOIN Plate p ON r.id\_plata = p.id\_plata

GROUP BY g.id\_ghid, g.nume\_ghid, g.prenume\_ghid

HAVING SUM(p.suma) > (

SELECT AVG(venit\_per\_ghid)

FROM (

SELECT SUM(p2.suma) as venit\_per\_ghid

FROM Ghid g2

JOIN Tur t2 ON g2.id\_ghid = t2.id\_ghid

JOIN Rezervare r2 ON t2.id\_tur = r2.id\_tur

JOIN Plate p2 ON r2.id\_plata = p2.id\_plata

GROUP BY g2.id\_ghid

) as avg\_venituri

)

ORDER BY venit\_total DESC

Această interogarecu **parametru variabil** afișează turiștii care au plătit prin card,numerar sau transfer pentru rezervările lor. Se selectează ID-ul turului (id\_tur), numele turului (nume\_tur), ID-ul turistului (id\_turist), numele și prenumele turistului (nume\_turist, prenume\_turist) și metoda de plată (metoda\_platii).

SELECT DISTINCT

t.id\_tur AS id\_tur,

t.nume\_tur AS nume\_tur,

tr.id\_turist AS id\_turist,

tr.nume AS nume\_turist,

tr.prenume AS prenume\_turist,

p.metoda\_platii AS metoda\_platii

FROM Tur t

JOIN Rezervare r ON t.id\_tur = r.id\_tur

JOIN Turisti tr ON r.id\_turist = tr.id\_turist

JOIN Plate p ON r.id\_plata = p.id\_plata

WHERE p.metoda\_platii = :payment\_method

ORDER BY t.id\_tur, tr.nume, tr.prenume;

Această interogare are ca scop identificarea tururilor care au generat un venit total mai mare decât venitul mediu al tuturor tururilor

SELECT t.nume\_tur, SUM(p.suma) AS venit\_total

FROM Tur t

JOIN Rezervare r ON t.id\_tur = r.id\_tur

JOIN Plate p ON r.id\_plata = p.id\_plata

GROUP BY t.id\_tur, t.nume\_tur

HAVING venit\_total > (

SELECT AVG(venit\_total\_tur)

FROM (

SELECT SUM(p2.suma) AS venit\_total\_tur

FROM Tur t2

JOIN Rezervare r2 ON t2.id\_tur = r2.id\_tur

JOIN Plate p2 ON r2.id\_plata = p2.id\_plata

GROUP BY t2.id\_tur

) AS venituri

)

ORDER BY venit\_total DESC;