Lupu Ionel- gr.406

Ceachi Bogdan- gr.406

Proiect Baze de date Orientate pe Obiecte

Agentie de Turism

**1. Descrierea modelului real**

In acest proiect prezentam modelul bazei de date al unei agentii de turism. Organizarea activitatilor turistice se realizeaza de catre diversi agenti de turism(persoane fizice care presteaza servicii specifice), iar agentia de turism este intermediarul privilegiat intre acesti prestatori si turist. Activitatile firmei, consta in producerea si promovarea de aranjamente turistice diverse.

Agentia prezinta doua mari avantaje fata de alti distribuitori de produse turistice: protectia aproape totala a consumatorului de turism si garantiile financiare acordate atat turistilor cat si prestatorilor.

Agentia ofera urmatoarele servicii: rezervari la hoteluri, cluburi, vile si pensiuni in toate orasele si statiunile lumii; programe turistice externe si interne pentru grupuri organizate; programe speciale cu excursii de weekend in capitalele sau marile orase ale lumii.

Clientii aleg oferta agentiei pentru ca le ofera valoare si satisfactie, iar personalul este calificat si sunt in permantenta legatura cu clientul dandu-i posibilitatea de a-s exprima opiniile favorabile sau nefavorabile.

Modelul de stocare a datelor realizat in cadrul acestui proiect este unul relational, asftel incat datele sunt organizate sub forma de tabele intre in care exista diverse legaturi.

**2. Constrangerile(restrictiile,regulile) modelului.**

1. **O unitate de cazare trebuie sa aiba un tip de unitate de cazare, cel putin o camera, o adresa si un contact.**
2. **Un pachet trebuie sa aiba un tip de pachet si sa apartina unui proiect global.**
3. **Un pachet poate avea o lista de oferte ale unitatii de cazare.**
4. **Un client poate rezerva camerele expuse de unitatile de cazare.**
5. **Un client trebuie sa aiba informatii de contact.**
6. **Un client poate cauta puncte de interes si poate lasa comentarii aferente acestora.**
7. **Un angajat al agentiei trebuie sa apartina unui departament si trebuie sa fie legat de un job.**
8. **O adresa trebuie sa apartina unui oras.**
9. **Un oras trebuie sa apartina unui judet.**
10. **Un judet trebuie sa apartina unei regiuni.**
11. **O regiune trebuie sa apartina unei tari.**
12. **Un punct de interes trebuie sa aiba un tip.**
13. **Angajatii administreaza informatiile inregistrate de punctele de interes.**

**3. Maparea relatiilor**

**-entitati,atribute (descriere, cheie primara, tip constrangeri)**

Tabela: **Accomodations**

***Descriere tabela:*** Detine informatii legate de unitatile de cazare( ex: hoteluri, vile, case de vacanta), pe care agentia le are disponibile.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK, NN | \* |
| name | Nume unitate de cazare | VARCHAR(45) | NN | \* |
| stars | Numarul de stele | INT | NN | \* |
| description\_short |  | TEXT |  | O |
| description\_full |  | TEXT |  | O |
| address\_id |  | INT | FK, NN | \* |
| accomodation\_type\_id |  | INT | FK, NN | \* |
| contact\_id |  | INT | FK,NN | \* |

Tabela: **Accomodation\_types**

**Descriere:** Lista pentru tipurile de unitati de cazare (ex:hotel, pensiune etc.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(25) | NN | \* |

Tabela: **Images**

Descriere: Pozele de promovare ale unitatii de cazare.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| path |  | VARCHAR(45) | NN | \* |
| is\_main |  | INT |  | O |
| accomodation\_id |  | INT | FK,NN | \* |

Tabela: **Facilities**

Descriere: Lista cu facilitatile disponibile ce le poate avea o unitate de cazare(ex: piscina, scari rulante, puzzle, loc de joaca)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |

Tabela: **Accomodation\_facilities**

**Descriere: tabel de legatura Facility->Accomodation (M:M)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| facility\_id |  | INT | PK,FK,NN | \* |
| accomodation\_id |  | INT | PK,FK,NN | \* |

Tabela: **Rooms**

Descriere: Lista de camere disponibile ale unei unitati de cazare.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| room\_type\_id |  | INT | FK,NN | \* |
| accomodation\_id |  | INT | FK,NN | \* |

Tabela: **Room\_types**

Descriere: Lista tipurilor de camere disponibile ale unei unitati de cazare.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | ID | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |

Tabela: **Contact**

Descriere: Lista cu informatii de contact ale unei unitati de cazare sau client (ex: email, telefon, website etc.);

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| email |  | VARCHAR(255) | NN | \* |
| phone |  | VARCHAR(45) | NN | \* |
| phone2 |  | VARCHAR(45) |  | O |
| fax |  | VARCHAR(45) |  | O |
| website |  | VARCHAR(255) |  | O |
| facebook |  | VARCHAR(45) |  | O |
| linkedin |  | VARCHAR(45) |  | O |

Tabela: **Adress**

Descriere: Lista locatiilor pentru fiecare entitate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| postal\_code |  | VARCHAR(45) | NN | \* |
| latitude |  | DOUBLE |  | O |
| longitude |  | DOUBLE |  | O |
| city\_id |  | INT | FK,NN | \* |

Tabela: **Cities**

**Descriere:** Lista cu orase ale agentiei de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(150) | NN | \* |
| ISO3 |  | VARCHAR(45) | NN | \* |
| latitude |  | DOUBLE |  | O |
| longitude |  | DOUBLE |  | O |
| county\_id |  | INT | FK,NN | \* |

Tabela: **Counties**

Descriere: Lista cu judete ale unei agentii de turism

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(255) | NN | \* |
| region\_id |  | INT | FK,NN | \* |

Tabela: **Regions**

Descriere: Lista cu regiunile agentiei de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| country\_id |  | INT | FK,NN | \* |

Tabela: **Countries**

Descriere: Lista cu tarile disponibile agentiei de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(99) | NN | \* |

Tabela: **Departments**

Descriere: Lista cu departamentele angentiei de turism(ex: relatii cu clienti, support etc.);

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| adress\_id |  | INT | FK,NN | \* |

Tabela: **Employees**

Descriere: Lista cu angajatii agentiei de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| last\_name |  | VARCHAR(45) | NN | \* |
| birth\_date |  | DATETIME | NN | \* |
| contact\_id |  | INT | FK, NN | \* |
| hire\_date |  | DATETIME | NN | \* |
| salary |  | INT | NN | \* |
| department\_id |  | INT | FK,NN | \* |
| job\_id |  | INT | FK,NN | \* |

Tabela: **Jobs**

Descriere: Lista joburilor disponibile din agentia de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | FK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| min\_salary |  | INT | NN | \* |
| max\_salary |  | INT | NN | \* |

Tabela: **Job\_Histories**

Descriere: Lista joburilor angajatilor agentiei de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| start\_date |  | DATETIME | NN | \* |
| end\_date |  | DATETIME | NN | \* |
| job\_id |  | INT | FK,NN | \* |
| employee\_id |  | INT | FK,NN | \* |

Tabela: **Clients**

Descriere: Lista de clienti ale agentiei de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| first\_name |  | VARCHAR(45) | NN | \* |
| last\_name |  | VARCHAR(45) | NN | \* |
| address\_id |  | INT | FK,NN | \* |
| contact\_id |  | INT | FK,NN | \* |

Tabela: **Things\_of\_interests**

Descriere: Lista cu punctele de interes inregistrate in agentia de turism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| description |  | TEXT |  | O |
| type\_id |  | INT | FK,NN | \* |
| address\_id |  | INT | FK,NN | \* |
| toy\_type\_id |  | INT | FK,NN | \* |

Tabela: **Things\_of\_interests\_types**

Descriere: Lista cu tipurile puctelor de interes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |

Tabela: **Client\_comments**

Descriere: Lista cu comentariile clientilor, aduse unor puncte de interes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| recommended |  | TINYINT |  | O |
| created\_date |  | DATETIME | NN | \* |
| message |  | TEXT | NN | \* |
| client\_id |  | INT | FK,NN | \* |
| things\_of\_interest\_id |  | INT | FK,NN | \* |

Tabela: **Projects**

Descriere: Lista cu proiectele disponibile ale agentiei de turism, folosite pentru a categoriza global pachetele.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| date\_start |  | DATE | NN | \* |
| date\_end |  | DATE | NN | \* |

Tabela: **Packages**

Descriere: Lista cu pachetele oferite de o unitate de cazare.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| description |  | TEXT |  | O |
| image |  | VARCHAR(255) |  | O |
| project\_id |  | INT | FK,NN | \* |
| package\_type\_id |  | INT | FK,NN | \* |

Tabela: **Package\_types**

Descriere: Lista cu tipurile de pachete oferite.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |

Tabela: **Offers**

Descriere: Lista cu ofertele disponibile dintr-un pachet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| duration |  | INT | NN | \* |
| duration\_unit |  | VARCHAR(45) |  | O |
| date\_start |  | DATE | NN | \* |
| date\_end |  | DATE | NN | \* |
| package\_id |  | INT | FK,NN | \* |

Tabela: **Configurations**

Descriere: Lista cu configuratii ale unei oferte cu o camera, ce contine pretul si disponibilitatea.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| availability |  | VARCHAR(45) | NN | \* |
| price |  | DOUBLE | NN | \* |
| room\_id |  | INT | FK,NN | \* |
| offer\_id |  | INT | FK,NN | \* |
| currency\_id |  | INT | FK,NN | \* |

Tabela: **Guests**

Descriere: Lista persoanelor ce se pot caza intr-o configuratie.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| code |  | VARCHAR(45) | NN | \* |
| count |  | INT | NN | \* |
| min\_age |  | INT | NN | \* |
| max\_age |  | INT | NN | \* |
| configuration\_id |  | INT | FK,NN | \* |

Tabela: **Currencies**

Descriere: Lista tipurilor de monezi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| name |  | VARCHAR(45) | NN | \* |
| code |  | VARCHAR(45) | NN | \* |

Tabela: **Rezervations**

Descriere: Lista rezervarilor ale clientilor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numele coloanei** | **Descriere** | **Tip** | **Constrangeri** | **Optionalitatea** |
| id |  | INT | PK,NN | \* |
| created\_date |  | TIMESTAMP | NN | \* |
| client\_id |  | INT | FK,NN | \* |
| configuration\_id |  | INT | FK,NN | \* |

**4. Relatii: descriere, Cardinalitate**

**Relatia Accomodation -> Image : 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** imagini **poate** avea o unitate de cazare? Multe

**Câte** imagini **trebuie** să aibă o unitate de cazare? Minim 0

**La câte** unitati de cazare **poate** să corespundă o imagine? Maxim 1

**La câte** unitati de cazare **trebuie** să corespundă o imagine? Minim 1

**Relatia Accomodation -> Facility: M:M**

**Optionalitatea si Cardinalitatea:**

**Cate** facilitati **poate** sa aiba o unitate de cazare? Multe

**Cate** facilitati **trebuie** sa aiba o unitate de cazare? Minim 0

**In cate** unitati de cazare **poate** sa fie o facilitate? Multe

**In cate** unitati de cazare **trebuie** sa fie o facilitate? Minim 0

**Relatia Accomodation\_Type -> Accomodation: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** unitati de cazare **poate** avea un tip de unitate de cazare? Multe

**Câte** unitati de cazare **trebuie** să aibă un tip de unitate de cazare? Minim 0

**La câte** tipuri de unitati de cazare **poate** să corespundă o unitate de cazare? Maxim 1

**La câte** tipuri de unitati de cazare **trebuie** să corespundă o unitate de cazare? Minim 1

**Relatia Accomodation -> Contact: 1:1**

**Optionalitatea si Cardinalitatea:**

**Câte** contacte **poate** avea o unitate de cazare? Maxim 1

**Câte** contacte **trebuie** să aibă o unitate de cazare? Minim 1

**La câte** unitati de cazare **poate** să corespundă un contact? Maxim 1

**La câte** unitati de cazare **trebuie** să corespundă un contact? Minim 1

**Relatia Accomodation -> Address: 1:1**

**Optionalitatea si Cardinalitatea:**

**Câte** adrese **poate** avea o unitate de cazare? Maxim 1

**Câte** adrese **trebuie** să aibă o unitate de cazare? Minim 1

**La câte** unitati de cazare **poate** să corespundă o adresa? Maxim 1

**La câte** unitati de cazare **trebuie** să corespundă o adresa? Minim 1

**Relatia Accomodation -> Room: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** camere **poate** avea o unitate de cazare? Multe

**Câte** camere **trebuie** să aibă o unitate de cazare? Minim 0

**La câte** unitati de cazare **poate** să corespundă camera? Maxim 1

**La câte** unitati de cazare **trebuie** să corespundă o imagine? Minim 1

**Relatia Room\_Type -> Room: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** camere **poate** avea un tip de camera? Multe

**Câte** camere **trebuie** să aibă un tip de camera? Minim 0

**La câte** tipuri de camere **poate** să corespundă o camera? Maxim 1

**La câte** tipuri de camere **trebuie** să corespundă o camera? Minim 1

**Relatia Country -> Region: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** regiuni **poate** avea o tara? Multe

**Câte** regiuni **trebuie** să aibă o tara? Minim 0

**La câte** tari **poate** să corespundă o regiune? Maxim 1

**La câte** tari **trebuie** să corespundă o regiune? Minim 1

**Relatia Region -> County: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** judete **poate** avea o regiune? Multe

**Câte** judete **trebuie** să aibă o regiune? Minim 0

**La câte** regiuni **poate** să corespundă unui judet? Maxim 1

**La câte** regiuni **trebuie** să corespundă unui judet? Minim 1

**Relatia County -> City: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** orase **poate** avea un judet? Multe

**Câte** orase **trebuie** să aibă un judet ? Minim 0

**La câte** judete **poate** să corespundă unui oras? Maxim 1

**La câte** judete **trebuie** să corespundă unui oras? Minim 1

**Relatia City -> Address: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** adrese **poate** avea un oras? Multe

**Câte** adrese **trebuie** să aibă un oras? Minim 0

**La câte** orase **poate** să corespundă unei adrese? Maxim 1

**La câte** orase **trebuie** să corespundă unei adrese? Minim 1

**Relatia Address-> Department: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** departamente **poate** avea o adresa? Multe

**Câte** departamente **trebuie** să aibă o adresa? Minim 0

**La câte** adrese **poate** să corespundă unui departament? Maxim 1

**La câte** adrese **trebuie** să corespundă unui departament? Minim 1

**Relatia Department-> Employee: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câti** angajati **poate** avea un departament? Multi

**Câti** angajati **trebuie** să aibă un departament? Minim 0

**La câte** departamente **poate** să corespundă unui angajat? Maxim 1

**La câte** departamente **trebuie** să corespundă unui angajat? Minim 1

**Relatia Job -> Employee: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câti** angajati **poate** avea un job? Multi

**Câti** angajati **trebuie** să aibă un job? Minim 0

**La câte** job-uri **poate** să corespundă unui angajat? Maxim 1

**La câte** job-uri **trebuie** să corespundă unui angajat? Minim 1

**Relatia Job -> Job\_History: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** istorice de job **poate** avea un job? Multe

**Câte** istorice de job **trebuie** să aibă un job?Minim 0

**La câte** job-uri **poate** să corespundă un istoric de job? Maxim 1

**La câte** job-uri **trebuie** să corespundă un istoric de job ? Minim 1

**Relatia Employee-> Job\_History: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** istorice de job **poate** avea un angajat? Multe

**Câte** istorice de job **trebuie** să aibă un angajat?Minim 1

**La câti** angajati **poate** să corespundă un istoric de job? Maxim 1

**La câti** angajati **trebuie** să corespundă un istoric de job? Minim 1

**Relatia Employee-> Contact: 1:1**

**Optionalitatea si Cardinalitatea:**

**Câte** contacte **poate** avea un angajat? Maxim 1

**Câte** contacte **trebuie** să aibă un angajat? Minim 1

**La câti** angajati **poate** să corespundă un contact? Maxim 1

**La câti** angajati **trebuie** să corespundă un contact? Minim 1

**Relatia Client-> Contact: 1:1**

**Optionalitatea si Cardinalitatea:**

**Câte** contacte **poate** avea un client? Maxim 1

**Câte** contacte **trebuie** să aibă un client? Minim 1

**La câti** clienti **poate** să corespundă un contact? Maxim 1

**La câti** clienti **trebuie** să corespundă un contact? Minim 1

**Relatia Client-> Address: 1:1**

**Optionalitatea si Cardinalitatea:**

**Câte** adrese **poate** avea un client? Maxim 1

**Câte** adrese **trebuie** să aibă un client? Minim 1

**La câti** clienti **poate** să corespundă o adresa? Maxim 1

**La câti** clienti **trebuie** să corespundă o adresa? Minim 1

**Relatia Things\_of\_interest\_types -> Things of interest: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** puncte de interes **poate** avea un tip de punct de interes? Multe

**Câte** puncte de interes **trebuie** să aibă un tip punct de interes? Minim 0

**La câte** tipuri de puncte de interes **poate** să corespundă unui punct de interes? Maxim 1

**La câte** tipuri de puncte de interes **trebuie** să corespundă unui punct de interes? Minim 1

**Relatia Things\_of\_interest-> Address: 1:1**

**Optionalitatea si Cardinalitatea:**

**Câte** adrese **poate** avea un punct de interes? Maxim 1

**Câte** adrese **trebuie** să aibă un punct de interes? Minim 1

**La câte** puncte de interes **poate** să corespundă o adresa? Maxim 1

**La câte** puncte de interes **trebuie** să corespundă o adresa? Minim 1

**Relatia Things\_of\_interest -> Client\_comments: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** comentarii **poate** avea un punct de interes? Multe

**Câte** comentarii **trebuie** să aibă un punct de interes? Minim 0

**La câte** puncte de interes **poate** să corespundă un comentariu? Maxim 1

**La câte** puncte de interes **trebuie** să corespundă un comentariu? Minim 1

**Relatia Client -> Client\_comments: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** comentarii **poate** avea un client? Multe

**Câte** comentarii **trebuie** să aibă un client? Minim 0

**La câti** clienti **poate** să corespundă un comentariu? Maxim 1

**La câti** clienti **trebuie** să corespundă un comentariu? Minim 1

**Relatia Project -> Package: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** pachete **poate** avea un proiect? Multe

**Câte** pachete **trebuie** să aibă un proiect? Minim 0

**La câte** proiecte **poate** să corespundă un pachet? Maxim 1

**La câte** proiecte **trebuie** să corespundă un pachet? Minim 1

**Relatia Package\_Type -> Package: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** pachete **poate** avea un tip de pachete? Multe

**Câte** pachete **trebuie** să aibă un tip de pachete? Minim 0

**La câte** tipuri de pachete **poate** să corespundă un pachet? Maxim 1

**La câte** tipuri de pachete **trebuie** să corespundă un pachet? Minim 1

**Relatia Package -> Offer: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** oferte **poate** avea un pachet? Multe

**Câte** oferte **trebuie** să aibă un pachet? Minim 0

**La câte** pachete **poate** să corespundă o oferta? Maxim 1

**La câte** pachete **trebuie** să corespundă o oferta? Minim 1

**Relatia Room-> Offer: M:M**

**Optionalitatea si Cardinalitatea:**

**Cate** oferte **poate** sa aiba o camera? Multe

**Cate** oferte **trebuie** sa aiba o camera? Minim 0

**Cate** camere **poate** sa aiba o oferta? Multe

**Cate** camere **trebuie** sa aiba o oferta? Minim 0

**Relatia Room-> Guest: M:M**

**Optionalitatea si Cardinalitatea:**

**Cati** vizitatori **poate** sa aiba o camera? Multi

**Cati** vizitatori **trebuie** sa aiba o camera? Minim 0

**Cate** camere **poate avea** un vizitator? Multe

**Cate** camere **trebuie** sa aibe un vizitator? Minim 0

**Relatia Room-> Reservation: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** rezervari se **pot** face asupra unei camere? Multe

**Câte** rezervari **trebuie** să se poata face asupra unei cameret? Minim 0

**La câte** camere **pot** să corespundă o rezervare? Maxim 1

**La câte** camere **trebuie** să corespundă o rezervare? Minim 1

**Relatia Client -> Reservation: 1:M**

**Optionalitatea si Cardinalitatea:**

**Câte** rezervari **poate** avea un client? Multe

**Câte** rezervari **trebuie** să aibă un client? Minim 0

**La câti** clienti **poate** să corespundă o rezervare? Maxim 1

**La câti** clienti **trebuie** să corespundă o rezervare? Minim 1

**Workbench :** [**https://photos.app.goo.gl/cclZh1C7rU6cr8GJ3**](https://photos.app.goo.gl/cclZh1C7rU6cr8GJ3)

**6. Diagrama E|R**

[**https://docs.google.com/drawings/d/1pjI523rh-uMP8zZaEX2eanBynVqYd8yEiX-qrhBrAIg/edit?usp=sharing**](https://docs.google.com/drawings/d/1pjI523rh-uMP8zZaEX2eanBynVqYd8yEiX-qrhBrAIg/edit?usp=sharing)

**2) Diagrama Contra-exemplu**

[**https://docs.google.com/drawings/d/1iafqprXBzpUAUuGAmeRkffoYZFWuoVGrmlNs7z\_9ucU/edit?usp=sharing**](https://docs.google.com/drawings/d/1iafqprXBzpUAUuGAmeRkffoYZFWuoVGrmlNs7z_9ucU/edit?usp=sharing)

**3) Diagrama E|R finala**

[**https://docs.google.com/drawings/d/11CYJBO0AeyRBPPxftEV9KMMf6nc8LAmqmkphflfQSJo/edit?usp=sharing**](https://docs.google.com/drawings/d/11CYJBO0AeyRBPPxftEV9KMMf6nc8LAmqmkphflfQSJo/edit?usp=sharing)

**7. Diagrama conceptuala**

[**https://docs.google.com/drawings/d/193bTYqwrQUJqG-Ey5WA9AMU7jv9hg075wQf\_\_wTpmhA/edit?usp=sharing**](https://docs.google.com/drawings/d/193bTYqwrQUJqG-Ey5WA9AMU7jv9hg075wQf__wTpmhA/edit?usp=sharing)

**8. Scheme relationale**

**ACCOMODATION**{#id, stars, name, image, description\_short, description\_full, address\_id, accomodation\_type\_id, contact\_id}

**FACILITY**{#id, name}

**ACCOMODATION\_FACILITY**{#facility\_id, accomodation\_id}

**ROOM**{#id, name, room\_type\_id, accomodation\_id}

**CONFIGURATION** {#id, availability, price, room\_id, offer\_id, currency\_id}

**GUEST** { #id, code, count, min\_age, max\_age, configuration\_id }

**OFFER** { #id, duration, duration\_unit, date\_end, currency, price, package\_id }

**PACKAGE** { #id\_package, #id\_project, name, date\_start, date\_end, package\_type, description, image }

**CURRENCY** { #id, name, code}

**REZERVATION** { #id, created\_date, client\_id, room\_id, configuration\_id}

**CLIENT** { #id, first\_name, last\_name, address\_id, contact\_id }

**CLIENT\_COMMENT** { #id, recommended, created\_date, message, client\_id, things\_of\_interest\_id}

**THINGS\_OF\_INTEREST** { #id, name, things\_of\_interest\_type, description, type\_id, adress\_id,

Toy\_type\_id}

**ADDRESS** { #id, name, postal\_code, city, county, region, country, latitude, longitude}

**DEPARTMENT**{#id\_departament, #id\_employee, name, last\_name, birth\_date, hire\_date, salary, job, job\_history, address\_id}

**9. Normalizare ( sau denormalizare?) FN3**

[**https://docs.google.com/drawings/d/1MJ-7ZXzjqDUCtts\_G3ve7wpXv52PjIfluu2sPpRl6SA/edit?usp=sharing**](https://docs.google.com/drawings/d/1MJ-7ZXzjqDUCtts_G3ve7wpXv52PjIfluu2sPpRl6SA/edit?usp=sharing)

Exemplu:

Din diagrama conceptuala, in entitatea Departament, atributele: **data\_nasterii, hire\_date, salary** etc., depind doar de id-ul angajatului nu de intregul **UID** care este combinatia dintre atributele **id\_departmen**t, **id\_employee**. Aceasta situatie se rezolva prin creearea unei noi entitati **EMPLOYEE** pe care o legam de entitatea **Department**. Echivalent, pentru proiectarea eficienta a bazei de date, evitarea anomaliilor si inconsistentelor, acelasi principiu se aplica si asupra celorlalte tabele ( de exemplu: PACKAGE, ADDRESS, THINGS\_OF\_INTEREST).

**10. Crearea tabelelor in SQLs**

**-- -----------------------------------------------------**

**-- Schema mydb**

**-- -----------------------------------------------------**

**DROP USER mydb CASCADE ;**

**1**

**-- -----------------------------------------------------**

**-- Schema mydb**

**-- -----------------------------------------------------**

**CREATE USER mydb IDENTIFIED BY mydb /\* DEFAULT CHARACTER SET utf8 \*/ ;**

**ALTER SESSION SET CURRENT\_SCHEMA = mydb ;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`contact`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.contact';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.contact (**

**id NUMBER(10) CHECK (id > 0) NOT NULL,**

**email VARCHAR2(255) NOT NULL,**

**phone VARCHAR2(45) NOT NULL,**

**phone2 VARCHAR2(45) NULL,**

**fax VARCHAR2(45) NULL,**

**website VARCHAR2(255) NULL,**

**facebook VARCHAR2(45) NULL,**

**linkedin VARCHAR2(45) NULL,**

**PRIMARY KEY (id))**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.contact\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.contact\_seq\_tr**

**BEFORE INSERT ON mydb.contact FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.contact\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**-- -----------------------------------------------------**

**-- Table `mydb`.`country`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.country';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.country (**

**id NUMBER(10) CHECK (id > 0) NOT NULL,**

**name VARCHAR2(99) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.country\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.country\_seq\_tr**

**BEFORE INSERT ON mydb.country FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.country\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**-- -----------------------------------------------------**

**-- Table `mydb`.`region`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.region';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.region (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**country\_id NUMBER(10) CHECK (country\_id > 0) NOT NULL,**

**PRIMARY KEY (id, country\_id)**

**,**

**CONSTRAINT fk\_region\_country1**

**FOREIGN KEY (country\_id)**

**REFERENCES mydb.country (id)**

**)**

**;**

**CREATE INDEX fk\_region\_country1\_idx ON mydb.region (country\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`county`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.county';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.county (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(255) NOT NULL,**

**region\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, region\_id)**

**,**

**CONSTRAINT fk\_county\_region1**

**FOREIGN KEY (region\_id)**

**REFERENCES mydb.region (id)**

**)**

**;**

**CREATE INDEX fk\_county\_region1\_idx ON mydb.county (region\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`city`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.city';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.city (**

**id NUMBER(10) CHECK (id > 0) NOT NULL,**

**name VARCHAR2(150) NOT NULL,**

**ISO3 VARCHAR2(45) NOT NULL,**

**latitude BINARY\_DOUBLE NULL,**

**longitude BINARY\_DOUBLE NULL,**

**county\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, county\_id)**

**,**

**CONSTRAINT fk\_agatur\_cities\_agatur\_counties1**

**FOREIGN KEY (county\_id)**

**REFERENCES mydb.county (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.city\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.city\_seq\_tr**

**BEFORE INSERT ON mydb.city FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.city\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_agatur\_cities\_agatur\_counties1\_idx ON mydb.city (county\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`address`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.address';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.address (**

**id NUMBER(10) CHECK (id > 0) NOT NULL,**

**name VARCHAR2(255) NOT NULL,**

**postal\_code VARCHAR2(45) NOT NULL,**

**latitude BINARY\_DOUBLE NULL,**

**longitude BINARY\_DOUBLE NULL,**

**city\_id NUMBER(10) CHECK (city\_id > 0) NOT NULL,**

**PRIMARY KEY (id, city\_id)**

**,**

**CONSTRAINT fk\_address\_city1**

**FOREIGN KEY (city\_id)**

**REFERENCES mydb.city (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.address\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.address\_seq\_tr**

**BEFORE INSERT ON mydb.address FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.address\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_address\_city1\_idx ON mydb.address (city\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`accomodation\_type`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.accomodation\_type';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.accomodation\_type (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`facility`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.facility';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.facility (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`room\_type`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.room\_type';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.room\_type (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`agatur\_accommodation\_secondary\_service\_types`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.agatur\_accommodation\_secondary\_service\_types';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.agatur\_accommodation\_secondary\_service\_types (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NULL,**

**code VARCHAR2(45) NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`accomodation`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.accomodation';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.accomodation (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**stars NUMBER(10) NOT NULL,**

**description\_full CLOB NULL,**

**description\_short CLOB NULL,**

**address\_id NUMBER(10) CHECK (address\_id > 0) NOT NULL,**

**contact\_id NUMBER(10) CHECK (contact\_id > 0) NOT NULL,**

**accomodation\_type\_id NUMBER(10) NULL,**

**PRIMARY KEY (id, address\_id, contact\_id, accomodation\_type\_id)**

**,**

**CONSTRAINT fk\_unitate\_cazare\_adresa1**

**FOREIGN KEY (address\_id)**

**REFERENCES mydb.address (id)**

**,**

**CONSTRAINT fk\_accomodation\_contact1**

**FOREIGN KEY (contact\_id)**

**REFERENCES mydb.contact (id)**

**,**

**CONSTRAINT fk\_accomodation\_accomodation\_type1**

**FOREIGN KEY (accomodation\_type\_id)**

**REFERENCES mydb.accomodation\_type (id)**

**)**

**;**

**CREATE INDEX fk\_unitate\_cazare\_adresa1\_idx ON mydb.accomodation (address\_id ASC);**

**CREATE INDEX fk\_accomodation\_contact1\_idx ON mydb.accomodation (contact\_id ASC);**

**CREATE INDEX fk\_accomodation\_accomodation\_type1\_idx ON mydb.accomodation (accomodation\_type\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`accomodation\_facility`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.accomodation\_facility';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.accomodation\_facility (**

**facility\_id NUMBER(10) NOT NULL,**

**accomodation\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (facility\_id, accomodation\_id)**

**,**

**CONSTRAINT fk\_agatur\_accommodations\_has\_agatur\_accommodation\_facilities\_2**

**FOREIGN KEY (facility\_id)**

**REFERENCES mydb.facility (id)**

**,**

**CONSTRAINT fk\_accomodation\_facility\_accomodation1**

**FOREIGN KEY (accomodation\_id)**

**REFERENCES mydb.accomodation (id)**

**)**

**;**

**CREATE INDEX fk\_agatur\_accommodations\_has\_agatur\_accommodation\_facilitie\_idx ON mydb.accomodation\_facility (facility\_id ASC);**

**CREATE INDEX fk\_accomodation\_facility\_accomodation1\_idx ON mydb.accomodation\_facility (accomodation\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`image`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.image';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.image (**

**id NUMBER(10) NOT NULL,**

**path VARCHAR2(45) NOT NULL,**

**is\_main NUMBER(3) NULL,**

**accomodation\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id)**

**,**

**CONSTRAINT fk\_image\_accomodation1**

**FOREIGN KEY (accomodation\_id)**

**REFERENCES mydb.accomodation (id)**

**)**

**;**

**CREATE INDEX fk\_image\_accomodation1\_idx ON mydb.image (accomodation\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`room`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.room';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.room (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**accomodation\_id NUMBER(10) NOT NULL,**

**room\_type\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, room\_type\_id)**

**,**

**CONSTRAINT fk\_room\_accomodation1**

**FOREIGN KEY (accomodation\_id)**

**REFERENCES mydb.accomodation (id)**

**,**

**CONSTRAINT fk\_room\_room\_type1**

**FOREIGN KEY (room\_type\_id)**

**REFERENCES mydb.room\_type (id)**

**)**

**;**

**CREATE INDEX fk\_room\_accomodation1\_idx ON mydb.room (accomodation\_id ASC);**

**CREATE INDEX fk\_room\_room\_type1\_idx ON mydb.room (room\_type\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`agatur\_package\_categories`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.agatur\_package\_categories';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.agatur\_package\_categories (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NULL,**

**code VARCHAR2(45) NULL,**

**parent\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, parent\_id)**

**,**

**CONSTRAINT fk\_agatur\_package\_categories\_agatur\_package\_categories1**

**FOREIGN KEY (parent\_id)**

**REFERENCES mydb.agatur\_package\_categories (id)**

**)**

**;**

**CREATE INDEX fk\_agatur\_package\_categories\_agatur\_package\_categories1\_idx ON mydb.agatur\_package\_categories (parent\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`project`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.project';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.project (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**date\_start DATE NOT NULL,**

**date\_end DATE NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`package\_type`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.package\_type';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.package\_type (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`package`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.package';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.package (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**description CLOB NULL,**

**image VARCHAR2(255) NULL,**

**project\_id NUMBER(10) NOT NULL,**

**package\_type\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, project\_id, package\_type\_id)**

**,**

**CONSTRAINT fk\_package\_project1**

**FOREIGN KEY (project\_id)**

**REFERENCES mydb.project (id)**

**,**

**CONSTRAINT fk\_package\_package\_type1**

**FOREIGN KEY (package\_type\_id)**

**REFERENCES mydb.package\_type (id)**

**)**

**;**

**CREATE INDEX fk\_package\_project1\_idx ON mydb.package (project\_id ASC);**

**CREATE INDEX fk\_package\_package\_type1\_idx ON mydb.package (package\_type\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`agatur\_package\_category`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.agatur\_package\_category';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.agatur\_package\_category (**

**package\_id NUMBER(10) NOT NULL,**

**category\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (package\_id, category\_id)**

**,**

**CONSTRAINT fk\_agatur\_package\_has\_agatur\_package\_categories\_agatur\_package1**

**FOREIGN KEY (package\_id)**

**REFERENCES mydb.package (id)**

**,**

**CONSTRAINT fk\_agatur\_package\_has\_agatur\_package\_categories\_agatur\_packag1**

**FOREIGN KEY (category\_id)**

**REFERENCES mydb.agatur\_package\_categories (id)**

**)**

**;**

**CREATE INDEX fk\_agatur\_package\_has\_agatur\_package\_categories\_agatur\_pack\_idx ON mydb.agatur\_package\_category (category\_id ASC);**

**CREATE INDEX fk\_agatur\_package\_has\_agatur\_package\_categories\_agatur\_pack\_idx1 ON mydb.agatur\_package\_category (package\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`offer`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.offer';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.offer (**

**id NUMBER(10) NOT NULL,**

**duration NUMBER(10) NOT NULL,**

**duration\_unit VARCHAR2(45) NULL,**

**date\_start DATE NOT NULL,**

**date\_end DATE NOT NULL,**

**package\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, package\_id)**

**,**

**CONSTRAINT fk\_oferta\_package1**

**FOREIGN KEY (package\_id)**

**REFERENCES mydb.package (id)**

**)**

**;**

**CREATE INDEX fk\_oferta\_package1\_idx ON mydb.offer (package\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`currency`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.currency';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.currency (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**code VARCHAR2(45) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- -----------------------------------------------------**

**-- Table `mydb`.`configuration`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.configuration';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.configuration (**

**id NUMBER(10) NOT NULL,**

**availability VARCHAR2(45) NOT NULL,**

**price BINARY\_DOUBLE NOT NULL,**

**room\_id NUMBER(10) NOT NULL,**

**currency\_id NUMBER(10) NOT NULL,**

**offer\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, room\_id, currency\_id, offer\_id)**

**,**

**CONSTRAINT fk\_configuration\_room1**

**FOREIGN KEY (room\_id)**

**REFERENCES mydb.room (id)**

**,**

**CONSTRAINT fk\_configuration\_currency1**

**FOREIGN KEY (currency\_id)**

**REFERENCES mydb.currency (id)**

**,**

**CONSTRAINT fk\_configuration\_offer1**

**FOREIGN KEY (offer\_id)**

**REFERENCES mydb.offer (id)**

**)**

**;**

**CREATE INDEX fk\_configuration\_room1\_idx ON mydb.configuration (room\_id ASC);**

**CREATE INDEX fk\_configuration\_currency1\_idx ON mydb.configuration (currency\_id ASC);**

**CREATE INDEX fk\_configuration\_offer1\_idx ON mydb.configuration (offer\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`guest`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.guest';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.guest (**

**id NUMBER(10) NOT NULL,**

**code VARCHAR2(45) NOT NULL,**

**count NUMBER(10) NOT NULL,**

**min\_age NUMBER(10) NOT NULL,**

**max\_age NUMBER(10) NOT NULL,**

**configuration\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, configuration\_id)**

**,**

**CONSTRAINT fk\_guest\_configuration1**

**FOREIGN KEY (configuration\_id)**

**REFERENCES mydb.configuration (id)**

**)**

**;**

**CREATE INDEX fk\_guest\_configuration1\_idx ON mydb.guest (configuration\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`departament`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.departament';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.departament (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**address\_id NUMBER(10) CHECK (address\_id > 0) NOT NULL,**

**PRIMARY KEY (id, address\_id)**

**,**

**CONSTRAINT fk\_departament\_adresa1**

**FOREIGN KEY (address\_id)**

**REFERENCES mydb.address (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.departament\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.departament\_seq\_tr**

**BEFORE INSERT ON mydb.departament FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.departament\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_departament\_adresa1\_idx ON mydb.departament (address\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`job`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.job';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.job (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**min\_salary NUMBER(10) NOT NULL,**

**max\_salary NUMBER(10) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.job\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.job\_seq\_tr**

**BEFORE INSERT ON mydb.job FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.job\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**-- -----------------------------------------------------**

**-- Table `mydb`.`employee`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.employee';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.employee (**

**id NUMBER(10) NOT NULL,**

**first\_name VARCHAR2(45) NOT NULL,**

**last\_name VARCHAR2(45) NOT NULL,**

**data\_nastere TIMESTAMP(0) NOT NULL,**

**hire\_date TIMESTAMP(0) NOT NULL,**

**salary NUMBER(10) NOT NULL,**

**departament\_id NUMBER(10) NOT NULL,**

**job\_id NUMBER(10) NOT NULL,**

**contact\_id NUMBER(10) CHECK (contact\_id > 0) NOT NULL,**

**PRIMARY KEY (id, departament\_id, job\_id, contact\_id)**

**,**

**CONSTRAINT fk\_employee\_departament1**

**FOREIGN KEY (departament\_id)**

**REFERENCES mydb.departament (id)**

**,**

**CONSTRAINT fk\_employee\_job1**

**FOREIGN KEY (job\_id)**

**REFERENCES mydb.job (id)**

**,**

**CONSTRAINT fk\_employee\_contact1**

**FOREIGN KEY (contact\_id)**

**REFERENCES mydb.contact (id)**

**)**

**;**

**COMMENT ON TABLE mydb.employee IS ' '**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.employee\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.employee\_seq\_tr**

**BEFORE INSERT ON mydb.employee FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.employee\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_employee\_departament1\_idx ON mydb.employee (departament\_id ASC);**

**CREATE INDEX fk\_employee\_job1\_idx ON mydb.employee (job\_id ASC);**

**CREATE INDEX fk\_employee\_contact1\_idx ON mydb.employee (contact\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`things\_of\_interest\_types`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.things\_of\_interest\_types';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.things\_of\_interest\_types (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**PRIMARY KEY (id))**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.things\_of\_interest\_types\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.things\_of\_interest\_types\_seq\_tr**

**BEFORE INSERT ON mydb.things\_of\_interest\_types FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.things\_of\_interest\_types\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**-- -----------------------------------------------------**

**-- Table `mydb`.`things\_of\_interest`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.things\_of\_interest';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.things\_of\_interest (**

**id NUMBER(10) NOT NULL,**

**name VARCHAR2(45) NOT NULL,**

**description CLOB NULL,**

**address\_id NUMBER(10) CHECK (address\_id > 0) NOT NULL,**

**toi\_type\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, address\_id, toi\_type\_id)**

**,**

**CONSTRAINT fk\_things\_of\_interest\_adresa1**

**FOREIGN KEY (address\_id)**

**REFERENCES mydb.address (id)**

**,**

**CONSTRAINT fk\_things\_of\_interest\_toi\_type1**

**FOREIGN KEY (toi\_type\_id)**

**REFERENCES mydb.things\_of\_interest\_types (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.things\_of\_interest\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.things\_of\_interest\_seq\_tr**

**BEFORE INSERT ON mydb.things\_of\_interest FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.things\_of\_interest\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_things\_of\_interest\_adresa1\_idx ON mydb.things\_of\_interest (address\_id ASC);**

**CREATE INDEX fk\_things\_of\_interest\_toi\_type1\_idx ON mydb.things\_of\_interest (toi\_type\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`job\_history`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.job\_history';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.job\_history (**

**id NUMBER(10) NOT NULL,**

**start\_date TIMESTAMP(0) NOT NULL,**

**end\_date TIMESTAMP(0) NOT NULL,**

**job\_id NUMBER(10) NOT NULL,**

**employee\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, job\_id, employee\_id)**

**,**

**CONSTRAINT fk\_job\_history\_job1**

**FOREIGN KEY (job\_id)**

**REFERENCES mydb.job (id)**

**,**

**CONSTRAINT fk\_job\_history\_employee1**

**FOREIGN KEY (employee\_id)**

**REFERENCES mydb.employee (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.job\_history\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.job\_history\_seq\_tr**

**BEFORE INSERT ON mydb.job\_history FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.job\_history\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_job\_history\_job1\_idx ON mydb.job\_history (job\_id ASC);**

**CREATE INDEX fk\_job\_history\_employee1\_idx ON mydb.job\_history (employee\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`client`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.client';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.client (**

**id NUMBER(10) NOT NULL,**

**first\_name VARCHAR2(45) NOT NULL,**

**last\_name VARCHAR2(45) NOT NULL,**

**address\_id NUMBER(10) CHECK (address\_id > 0) NOT NULL,**

**contact\_id NUMBER(10) CHECK (contact\_id > 0) NOT NULL,**

**PRIMARY KEY (id, address\_id, contact\_id)**

**,**

**CONSTRAINT fk\_client\_adresa1**

**FOREIGN KEY (address\_id)**

**REFERENCES mydb.address (id)**

**,**

**CONSTRAINT fk\_client\_contact1**

**FOREIGN KEY (contact\_id)**

**REFERENCES mydb.contact (id)**

**)**

**;**

**CREATE INDEX fk\_client\_adresa1\_idx ON mydb.client (address\_id ASC);**

**CREATE INDEX fk\_client\_contact1\_idx ON mydb.client (contact\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`client\_comment`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.client\_comment';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.client\_comment (**

**id NUMBER(10) NOT NULL,**

**recommended NUMBER(3) NULL,**

**created\_date TIMESTAMP(0) NOT NULL,**

**message CLOB NOT NULL,**

**client\_id NUMBER(10) NOT NULL,**

**things\_of\_interest\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id)**

**,**

**CONSTRAINT fk\_client\_comment\_client1**

**FOREIGN KEY (client\_id)**

**REFERENCES mydb.client (id)**

**,**

**CONSTRAINT fk\_client\_comment\_things\_of\_interest1**

**FOREIGN KEY (things\_of\_interest\_id)**

**REFERENCES mydb.things\_of\_interest (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.client\_comment\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.client\_comment\_seq\_tr**

**BEFORE INSERT ON mydb.client\_comment FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.client\_comment\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_client\_comment\_client1\_idx ON mydb.client\_comment (client\_id ASC);**

**CREATE INDEX fk\_client\_comment\_things\_of\_interest1\_idx ON mydb.client\_comment (things\_of\_interest\_id ASC);**

**-- -----------------------------------------------------**

**-- Table `mydb`.`rezervation`**

**-- -----------------------------------------------------**

**BEGIN**

**EXECUTE IMMEDIATE 'DROP TABLE mydb.rezervation';**

**EXCEPTION**

**WHEN OTHERS THEN NULL;**

**END;**

**/**

**CREATE TABLE mydb.rezervation (**

**id NUMBER(10) NOT NULL,**

**created\_date TIMESTAMP(0) NOT NULL,**

**client\_id NUMBER(10) NOT NULL,**

**configuration\_id NUMBER(10) NOT NULL,**

**PRIMARY KEY (id, client\_id, configuration\_id)**

**,**

**CONSTRAINT fk\_rezervation\_client1**

**FOREIGN KEY (client\_id)**

**REFERENCES mydb.client (id)**

**,**

**CONSTRAINT fk\_rezervation\_configurare1**

**FOREIGN KEY (configuration\_id)**

**REFERENCES mydb.configuration (id)**

**)**

**;**

**-- Generate ID using sequence and trigger**

**CREATE SEQUENCE mydb.rezervation\_seq START WITH 1 INCREMENT BY 1;**

**CREATE OR REPLACE TRIGGER mydb.rezervation\_seq\_tr**

**BEFORE INSERT ON mydb.rezervation FOR EACH ROW**

**WHEN (NEW.id IS NULL)**

**BEGIN**

**SELECT mydb.rezervation\_seq.NEXTVAL INTO :NEW.id FROM DUAL;**

**END;**

**/**

**CREATE INDEX fk\_rezervation\_client1\_idx ON mydb.rezervation (client\_id ASC);**

**CREATE INDEX fk\_rezervation\_configurare1\_idx ON mydb.rezervation (configuration\_id ASC);**