UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

**PROIECT**

**BAZE DE DATE**

STUDENT:

OPREA CALIN

GRUPA: 134

BUCUREȘTI

2023

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

**AGENȚIE DE TURISM**

CUPRINS

Contents

[1 Prezentarea bazei de date. 3](#_Toc135165863)

[1.1 Prezentarea modelului real. 3](#_Toc135165864)

[1.2 Reguli generale 3](#_Toc135165865)

[2 Diagrama Entitate-Relație 4](#_Toc135165866)

[2.1 Prezentarea diagramei 4](#_Toc135165867)

[2.2 Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților 4](#_Toc135165868)

[3 Diagrama Conceptuală 8](#_Toc135165869)

[3.1 Prezentarea diagramei 8](#_Toc135165870)

[3.2 Descrierea constrângerilor de integritate 8](#_Toc135165871)

[3.3 Schemele relationale 11](#_Toc135165872)

[4 Implementarea bazei de date 11](#_Toc135165873)

[4.1 Crearea tabelelor (inclusiv a constrângerilor) 11](#_Toc135165874)

[4.2 Introducerea datelor 16](#_Toc135165875)

[5 Cereri complexe: 22](#_Toc135165876)

# Prezentarea bazei de date.

Tema pe care am ales-o pentru proiect este crearea unei baze de date pentru gestionarea unei agenții de turism.

## Prezentarea modelului real.

Agenția de turism este una dintre cele mai importante organizații din sectorul privat de turism, care joacă un rol important în întregul proces de dezvoltare si promovare a turismului în țara sau într-o altă destinație. Aceasta ambalează și procesează toate atracțiile, facilitățile și serviciile auxiliare ale unei țari și le prezintă turiștilor.

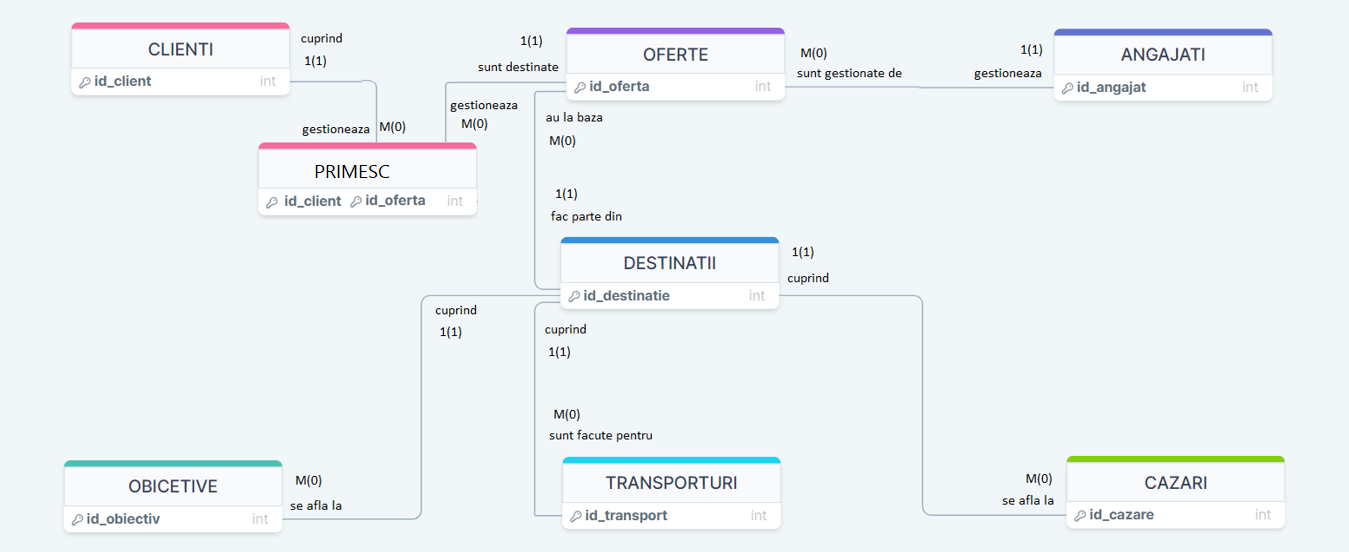
O societate care oferă servicii de turism către populație trebuie să se ridice la un anumit nivel. De calitatea serviciilor prestate depinde în final eficiența economică a societații. În acest context se impune sporirea calității ofertelor turistice și a serviciilor auxiliare. Unul din factorii ce ar putea imprima un nivel mai bun serviciilor prestate îl constituie informatizarea activităților tursitice, asigurând astfel promptitudine și o felxibilitate în relațiile cu clienții.

## Reguli generale

* Clienții primesc oferte.
* Ofertele sunt destinate clienților.
* Ofertele sunt gestionate de angajați.
* Angajații gestioneaza oferte.
* Ofertele au la baza destinații.
* Destinațiile fac parte din oferte.
* Destinațiile cuprind obiective turistice.
* Obiectivele turistice se află la destinații.
* Destinațiile cuprind transporturi.
* Transporturile sunt făcute pentru destinații.
* Destinațiile cuprind cazări.
* Cazările se află la destinații.
* Un obiectiv trebuie să aibă neapărat o destinație.
* O modalitate de transport trebuie să aibă neapărat o destinație.
* O variantă de cazare trebuie să aibă neapărat o destinație.
* O ofertă primită de un client trebuie să aibă neapărat un angajat care gestionează acea ofertă.
* In baza de date pot exista clienți care nu au primit nicio ofertă.
* Daca ne dorim să ștergem o destinație din baza de date, se vor șterge automat obiectivele, transporturile si cazările pentru respectiva destinație, deoarece nu are rost să păstram date pentru o destinație care nu mai face parte din baza de date.
* Daca vrem să ștergem un client, o ofertă sau un angajat, tabelele care derivă din ele nu vor fi afectate, deoarece ne dorim să păstrăm o evidență a acestora.

# Diagrama Entitate-Relație

## Prezentarea diagramei

  
Fig. 1 – Diagrama Entitate-Relatie

## Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

1. Tabelul **ANGAJATI** stochează datele angajaților agenției de turism.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_angajat | identificator unic al angajatului |
|  | nume | numele angajatului |
|  | prenume | prenumele angajatului |
|  | telefon | numărul de telefon al angajatului |
|  | tip\_contract\_munca | tipul de contract de muncă pe care angajatul îl are |
|  | data\_angajarii | data în care persoana a fost angajată |

Tabelul **angajati** se află in relație de **1(1) – M(0)** cu tabelul **oferte**, deoarece:

* Oferta este gestionată de **un** angajat.
* Oferta **trebuie** să fie gestionată de **un** angajat.
* Un angajat poate să gestioneze **mai multe** oferte.
* Un angajat **poate** să nu gestioneze **nicio** ofertă (în cazul în care este un angajat nou).

1. Tabelul **CLIENTI** contine datele esențiale pentru clienții agenției.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_client | identificator unic al clientului |
|  | nume | numele clientului |
|  | prenume | prenumele clientului |
|  | telefon | numărul de telefon al clientului |
|  | CNP | codul numeric personal al clientului |

Tabel **clienti** se află în relație de **M(0) – N(0)** cu tabelul **oferte**, deoarece:

* Un client poate să primescă **mai multe** oferte.
* Un client **poate** să nu primeasca **nicio** ofertă (am declarant în reguli că în baza de date pot exista clienți care nu au nicio ofertă primită).
* O ofertă poate fi destinată **mai multor** clienți.
* O ofertă **poate** să nu fie detinată **niciunui** client.

1. Tabelul **DESTINATII** conține date pentru locațiile pe care agenția de turism le oferă.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_destinatie | identificator unic al destinației |
|  | tara | țara destinației in cauză |
|  | oras | orașul destinației in cauză |

Tabel **destinatii** se află în relație de **1(1) – M(0)** cu tabelele **oferte, obiective, transporturi și cazari**.

1. Tabelul **OFERTE** are la bază date specifice pentru fiecare oferta din agentie.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_oferta | identificator unic al ofertei |
|  | pret/persoana | prețul pentru oferta în cauză |
|  | activ\_de\_la\_data | data de la care oferta este activă (începând de la acestă dată oferta poate să fie selectată) |
|  | activ\_pana la data | data până la care oferta este activă (după acestă dată oferta nu mai poate să fie selectată) |
| FK | id\_angajat | identificator unic al angajatului |
| FK | id\_destinatie | identificator unic al destinației |

Tabel **oferte** se află în relație de **1(1) – M(0)** cu tabelul **oferte**, deoarece:

* O ofertă are la bază **o** destinație.
* O ofertă **trebuie** să aibă la bază **o** destinație.
* O destinație poate să facă parte din **mai multe** oferte.
* O destinație **poate** să nu facă parte din **nicio** ofertă.

Cum relația dintre tabelele **clienti** și **oferte** este o relație **M(0) – N(0)** va rezulta apariția unui tabel asociativ “**primesc**”.

1. Tabelul **PRIMESC** memorează datele calendaristice când un anumit client a primit o ofertă anume.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
|  | data | data calendaristică când oferta a fost primită |
| FK,PK | id\_client | identificator unic al clientului |
| FK,PK | id\_oferta | identificator unic al ofertei |

1. Tabelul **OBIECTIVE** are în componență date pentru obictivele de la fiecare destinație care face parte din agenție.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_obiectiv | identificator unic al obiectivului |
|  | denumire | denumirea obiectivului |
|  | tip\_obiectiv | tipul de obiectiv (muzeu, monument, parc, etc.) |
|  | adresa | adresa la care se găsește obiectivul |
|  | telefon | numarul de telefonul (pentru a avea un mod de contact) |
| FK | id\_destinatie | identificator unic al destinației |

Tabel **obiective** se află în relație de **1(1) – M(0)** cu tabelul **destinatii**, deoarece:

* O destinație cuprinde **mai multe** obiective turistice.
* O destinație **poate** să nu cuprindă **niciun** obiectiv turistic.
* Un obiectiv turistic se află la **o** **singură** destinație.
* Un obiectiv turistic **trebuie** neapărat să aibă **o** destinație la care se află.

1. Tabelul **TRANSPORTURI** cuprinde date pentru variantele de transport atribuite pentru fiecare destinație în parte.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_transport | identificator unic al transportului |
|  | modalitate | modalitatea de deplasare (feroviar, aerian, rutier, etc.) |
|  | companie | compania care furnizează respectiva variantă |
|  | locul\_plecarii | locația de unde se pleacă cu respectiva variantă |
| FK | id\_destinatie | identificator unic al destinației |

Tabel **transporturi** se află în relație de **1(1) – M(0)** cu tabelul **destinatii**, deoarece:

* O destinație cuprinde **mai multe** tipuri de transport.
* O destinație **poate** să nu aibă **nicio** modalitate de transport.
* O modalitate de transport are **o** destinație aferentă.
* O modalitate de transport **trebuie** sa aiba **o** destinație.

1. Tabelul **CAZARI** memorează date pentru variantele de cazări pe care fiecare destinație din agenția de turism le ofertă.

|  |  |  |
| --- | --- | --- |
| Cheie | Denumire atribut | Descriere |
| PK | id\_cazare | identificator unic al cazării |
|  | nume | numele cazării în cauză |
|  | adresa | adresa unde se află cazarea |
|  | contact | o modalitate de contact (telefon, email) |
|  | calificativ | calificativul pe care cazarea îl are |
|  | facilitati | lista de facilități de care cazarea dispune |
| FK | id\_destinatie | identificator unic al destinației |

Tabel **cazari** se află în relație de **1(1) – M(0)** cu tabelul **destinatii**, deoarece:

* O destinație are **mai multe** variante de cazări.
* O destinație **poate** să nu aibă **nicio** variantă de cazare.
* O cazare se află la **o** destinație.
* O cazare **trebuie** să aibă **o** destinație la care să se afle.

# Diagrama Conceptuală

## Prezentarea diagramei

Fig. 2 – Diagrama Conceptuala

## Descrierea constrângerilor de integritate

**Tabelul DESTINATII:**

id\_destinatie – primary key, not null (cardinalitatea minima este 1)

tara – not null (fiecare destinație trebuie să aibă o țară)

oras – not null (fiecare destinație trebuie să aibă un oraș)

**Tabelul CLIENTI:**

id\_client – primary key, not null (cardinalitatea minima este 0, pot exista date de la clienți care nu au primit nicio ofertă încă)

nume – not null (nu există persoană care să nu aibă nume)

prenume – not null (nu există persoană care să nu aibă prenume)

telefon – inițializat cu null (pot exista persoane care nu au număr de telefon)

cnp – not null (nu există persoane care să nu aibă cod numeric personal)

**Tabelul ANGAJATI:**

id\_angajat – primary key, not null (cardinalitatea minima este 1, fiecare ofertă trebuie să fie gestionată de un angajat)

nume – not null (nu există persoană care să nu aibă nume)

prenume – not null (nu există persoană care să nu aibă prenume)

telefon – initializat cu null (pot exista persoane care nu au număr de telefon)

tip\_contract\_munca – not null (o persoană nu poate să fie angajată fară un contract de muncă)

data\_angajarii – not null (fiecare angajat trebuie să aibă o dată a angajarii)

**Tabelul OBIECTIVE:**

id\_obictiv – primary key, not null (cardinalitatea minima este 0, o destinație poate să nu aibă niciun obiectiv)

denumire – not null (nu există obiective fără nume)

tip\_obiectiv – initializat cu null (obiectivele a căror tip este evident din denumire nu trebuie neapărat să aibă completat acest atribut)

adresa – not null (fiecare obiectiv se află la o anumită adresă)

telefon – initializat cu null (pot exista obiective care nu au un număr de telefon pentru contact, ex: parcuri)

id\_destinatie – foreign key, not null, on delete cascade, on update cascade (dacă din baza de date se va șterge o destinație, nu va avea rost să păstrăm obiectivele pentru acea destinație)

**Tabelul TRANSPORTURI:**

id\_transport – primary key, not null (cardinalitatea minima este 0, o destinație poate să nu aibă nicio modalitate de transport)

modalitate – intializat cu null (pot exista variante de transport care nu au o modalitate explicită)

companie – initializat cu null (pot exista modalitați de transport care nu țin de o companie)

locul\_plecarii – initializat cu null (pot exista variante de transport care nu sunt dependente de un singur loc de plecare)

id\_destinatie – foreign key, not null, on delete cascade, on update cascade (dacă din baza de date se va șterge o destinație, nu va avea rost să păstrăm modalitățile de transport pentru acea destinație)

**Tabelul CAZARI:**

id\_cazare – primary key, not null (cardinalitatea minima este 0, o destinație poate să nu aibă nicio variantă de cazare)

nume – not null (fiecare cazare trebuie să aibă un nume)

adresa – not null (fiecare cazare trebuie să aibă o adresă)

contact – not null (fiecare cazare trebuie să aibă o variantă de contact)

calificativ – intializat cu null (pot exista cazări pentru care agenția să nu aibă suficiente informații, cum ar fi calificativul)

facilitati – initalizat cu null (pot exista cazări pentru care agenția să nu aibă suficiente informații, cum ar fi facilitațile)

id\_destinatie – foreign key, not null, on delete cascade, on update cascade (dacă din baza de date se va șterge o destinatie, nu va avea rost să păstrăm variantele de cazări pentru acea destinație)

**Tabelul OFERTE:**

id\_oferta – primary key, not null

pret/persoana – not null (fiecare ofertă trebuie să aibă un preț)

activ\_de\_de\_data – not null (data calendaristică de când este activă oferta este obligatorie)

activ\_pana la data – not null (data calendaristică până când este activă oferta este obligatorie)

id\_angajat – foreign key, not null, on delete no action, on update no action (în cazul în care ne dorim să ștergem un angajat, nu ar trebui să ștergem si ofertele pe care acesta le-a gestionat, acestea ar trebui să rămână în baza noastră de date, pentru a avea o evindența)

id\_destinatie – foreign key, not null, on delete no action, on update no action (în cazul în care ne dorim să ștergem o destinație, ofertele făcute pe baza destinației respective trebuie să rămână în baza de date a agenției)

**Tabelul PRIMESC:**

data – not null (un client poate să apeleze la agenția de turism de mai multe ori, de aceea vrem să păstrăm fiecare dată calendaristică la care acesta a primit o ofertă)

id\_clienti – foreign key, primary key, on delete no action, on update no action (în cazul în care dorim să ștergem un client din baza de date, nu vom șterge datele când acesta a apelat la agenție)

id\_oferta – foreign key, primary key, on delete no action, on update no action (în cazul în care ne dorim să ștergem o ofertă din baza nostră de date, nu vom șterge datele când această ofertă a fost utilizată)

## Schemele relationale

CLIENTI(id\_client#, nume, prenume, telefon, CNP);

PRIMESC(data, id\_client#, id\_oferta#);

OFERTE(id\_oferta#, pret/persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat#, id\_destinatie#);

ANGAJATI(id\_angajat#, nume, prenume, telefon, tip\_contract\_munca, data\_angajarii);

DESTINATII(id\_destinatie#, tara, oras);

OBIECTIVE(id\_obiectiv#, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie#);

TRANSPORTURI(id\_transport#, modalitate, companie, locul\_plecarii, id\_destinatie#);

CAZARI(id\_cazare#, nume, adresa, contact, calificativ, facilitati, id\_destinatie#);

# Implementarea bazei de date

## Crearea tabelelor (inclusiv a constrângerilor)

Modelul prezentat anterior a fost implementat într-un sistem de gestiune a bazei de date – ORACLE, instalat local.

Am început prin crearea tabelelor care nu au in componență chei străine (**destinatii, clienti, angajati**).

Apoi am creat tabelele **obiective, transporturi si cazari** cu constrângerea că, dacă voi șterge o destinație se vor șterge si datele din aceste 3 tabele care depind de destinația respectivă.

Ultimele tabele create au fost **oferte si primesc**. Mai întâi am creat **oferte**, deoarece **primesc** are în componență cheia străină “id\_oferta”.

**Crearea tabelului DESTINATII:**

CREATE TABLE destinatii (

id\_destinatie INT NOT NULL,

tara VARCHAR(30) NOT NULL,

oras VARCHAR(30) NOT NULL,

PRIMARY KEY (id\_destinatie));

**Crearea tabelului CLIENTI:**

CREATE TABLE clienti (

id\_clienti INT NOT NULL,

nume VARCHAR(45) NOT NULL,

prenume VARCHAR(45) NOT NULL,

telefon VARCHAR(16) NULL,

cnp VARCHAR(13) NOT NULL,

PRIMARY KEY (id\_clienti));

**Crearea tabelului ANGAJATI:**

CREATE TABLE angajati (

id\_angajati INT NOT NULL,

nume VARCHAR(45) NOT NULL,

prenume VARCHAR(45) NOT NULL,

telefon VARCHAR(16) NULL,

tip\_contract\_munca VARCHAR(15) NOT NULL,

data\_angajarii DATE NOT NULL,

PRIMARY KEY (id\_angajati));

**Crearea tabelului OBIECTIVE:**

CREATE TABLE obiective (

id\_obiectiv INT NOT NULL,

denumire VARCHAR(45) NOT NULL,

tip\_obiectiv VARCHAR(45) NULL,

adresa VARCHAR(60) NOT NULL,

telefon VARCHAR(16) NULL,

id\_destinatie INT NOT NULL,

PRIMARY KEY (id\_obiectiv),

CONSTRAINT id\_destinatie

FOREIGN KEY (id\_destinatie)

REFERENCES destinatii (id\_destinatie)

ON DELETE CASCADE);

**Crearea tabelului TRANSPORTURI:**

CREATE TABLE transporturi(

id\_transport INT NOT NULL,

modalitate VARCHAR(45) NULL,

companie VARCHAR(45) NULL,

locul\_plecarii VARCHAR(45) NULL,

id\_destinatie INT NOT NULL

PRIMARY KEY (id\_transport),

CONSTRAINT fk\_transport

FOREIGN KEY (id\_destinatie)

REFERENCES destinatii (id\_destinatie)

ON DELETE CASCADE);

**Crearea tabelului CAZARI:**

CREATE TABLE cazari(

id\_cazare INT NOT NULL,

nume VARCHAR(45) NOT NULL,

adresa VARCHAR(60) NOT NULL,

contact VARCHAR(45) NOT NULL,

calificativ FLOAT NULL,

facilitati VARCHAR(100) NULL,

id\_destinatie INT NULL,

PRIMARY KEY(id\_cazare),

CONSTRAINT fk\_cazari

FOREIGN KEY (id\_destinatie)

REFERENCES destinatii(id\_destinatie)

ON DELETE CASCADE

);

**Crearea tabelului OFERTE:**

CREATE TABLE oferte (

id\_oferta INT NOT NULL,

pret/persoana VARCHAR(15) NOT NULL,

activ\_de\_la\_data DATE NOT NULL,

activ\_pana\_la\_data DATE NOT NULL,

id\_angajat INT NOT NULL,

id\_destinatie INT NOT NULL,

PRIMARY KEY (id\_oferta),

CONSTRAINT fk.angajat

FOREIGN KEY (id\_angajat)

REFERENCES angajati(id\_angajat)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT fk.oferta

FOREIGN KEY (id\_destinatie)

REFERENCES destinatii (id\_destinatie)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

**Crearea tabelului PRIMESC:**

CREATE TABLE primesc(

data DATE NOT NULL,

id\_client INT NULL,

id\_oferta INT NULL,

CONSTRAINT fk\_client

FOREIGN KEY (id\_client)

REFERENCES clienti (id\_client)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT fk\_primesc

FOREIGN KEY (id\_oferta)

REFERENCES oferte (id\_oferta)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

După crearea tabelului **primesc** având cele două chei stăine (id\_client, id\_oferta), am **alterat** tabelul pentru crearea unei chei primare comune între id\_client și id\_oferta.

ALTER TABLE primesc

DROP FOREIGN KEY fk\_client,

DROP FOREIGN KEY fk\_primesc;

ALTER TABLE primesc

CHANGE COLUMN id\_client id\_client INT NOT NULL ,

CHANGE COLUMN id\_oferta id\_oferta INT NOT NULL ,

ADD PRIMARY KEY (id\_client, id\_oferta);

ALTER TABLE primesc

add(CONSTRAINT fk\_client

FOREIGN KEY (id\_client)

REFERENCES clienti(id\_clienti)),

CONSTRAINT fk\_primesc

FOREIGN KEY (id\_oferta)

REFERENCES oferte(id\_oferta));

## Introducerea datelor

Pentru partea de introducere a datelor am abordat aceeași modalitate ca la crearea tabelelor.

Am introdus datele mai întâi în tabelele **destinatii, clienti și angajati**, deoarece acestea nu au în componență chei străine.

Am continuat prin introducerea datelor pentru tabelele **obiective, transporturi și cazari**, acestea depinzând de tabelul **destinatii**.

Ultimele tabele pentru care am introdus date au fost **oferte si primesc**. Mai întâi pentru **oferte**, deoarece tabelul **primesc** are în componență cheia straina “id\_oferta”.

**Introducerea datelor in tabelul DESTINATII:**

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES (‘1’,’Spania’, ‘Barcelona’);

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('2', 'Franta', 'Paris');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('3', 'Italia', 'Roma');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('4', 'Austria', 'Viena');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('5', 'Romania', 'Bucuresti');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('6', 'Regatul Unit', 'Londra');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('7', 'Turcia', 'Ankara');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('8', 'Grecia', 'Atena');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('9', 'Suedia', 'Stockholm');

INSERT INTO destinatii (id\_destinatie, tara, oras) VALUES ('10', 'Germania', 'Berlin');

**Introducerea datelor in tabelul CLIENTI:**

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('1', 'Popescu', 'Maria', '0723679037', '6001216018978');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('2', 'Stan', 'Andrei', '0758676097', '1950719338313');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES('3', 'Matei', 'Ana', '0769432023', '2990513059974');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES('4', 'Ciobanu', 'Alexandru', '0738566622', '1880721247770');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('5', 'Popovici', 'George', '0744892757', '1930723305095');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES('6', 'Ungureanu', 'Irina', '0755122947', '2991016418331');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('7', 'Petre', 'Madalina', '0743752100', '2221231359322');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES('8', 'Nistor', 'Simona', '0770058461', '6020423335751');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('9', 'Simion', 'Florin', '0753110648', '5010720229063');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('10', 'Voicu', 'Bianca', '0748392742', '6221216098318');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('11', 'Radulescu', 'Elena', '0745271975', '2730509416424');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('12', 'Vlad', 'Codrin', '0769382647', '2800819417125');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('13', 'Andrei', 'Georgiana', '0739248448', '2861129316831');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('14', 'Rus', 'Andreea', '0799329219', '2861111255204');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES('15', 'Marcu', 'Eduard', '0747201859', '1890601287095');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('16', 'Rosu', 'Madalina', '0729447599', '2890208286428');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('17', 'Muntean', 'Darius', '0700930209', '1890601245131');

**Introducerea datelor in tabelul ANGAJATI:**

INSERT INTO angajati(id\_angajat, nume, prenume, telefon, tip\_contract\_munca, data\_angajarii) VALUES ('1', 'Toma', 'Dan', '0725894313', 'norma intreaga', to\_date('2009-12-01','yyyy-mm-dd'));

INSERT INTO angajati(id\_angajat, nume, prenume, telefon, tip\_contract\_munca, data\_angajarii) VALUES ('2', 'Negru', 'Iustina', '0748249200', 'norma intreaga', to\_date('2010-03-23','yyyy-mm-dd'));

INSERT INTO angajati(id\_angajat, nume, prenume, telefon, tip\_contract\_munca, data\_angajarii) VALUES ('3', 'Lupu', 'Diana', '0739428485', 'timp partial', to\_date('2016-06-14','yyyy-mm-dd'));

INSERT INTO angajati(id\_angajat, nume, prenume, telefon, tip\_contract\_munca, data\_angajarii) VALUES ('4', 'Lungu', 'Cosmin', '0755339289', 'norma intreaga', to\_date('2016-10-07','yyyy-mm-dd'));

INSERT INTO angajati(id\_angajat, nume, prenume, telefon, tip\_contract\_munca, data\_angajarii) VALUES ('5', 'Surdu', 'Alina', '0794728748', 'timp partial', to\_date('2018-08-01','yyyy-mm-dd'));

**Introducerea datelor in tabelul OBIECTIVE:**

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('1', 'Colosseum', 'Cladire istorica', 'Piazza del Colosseo, 1, 00184 Roma RM, Italy', '+390639967700', '3');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('2', 'Forumul Roman', 'Cladire istorica', 'Via della Salara Vecchia, 5/6, 00186 Roma RM, Italy', '+390639967700', '3');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('3', 'Turnul Eiffel', 'Turn de observare', 'Champ de Mars, 5 Av. Anatole France, 75007 Paris, France', ' ', '2');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('4', 'Catedrala Sfantul Stefan', 'Catedrala', 'Stephansplatz 3, 1010 Wien, Austria', '+431515523530', '4');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('5', 'La Sagrada Familia', 'Biserica', 'C/ de Mallorca, 401, 08013 Barcelona, Spain', '+34932080414', '1');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('6', 'Louvre', 'Muzeu', 'Rue de Rivoli, 75001 Paris, France', '+33140205050', '2');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('7', 'Park Güell', 'Parc', '08024 Barcelona, Spain', '+34934091831', '1');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('8', 'Casa Mozart', 'Monument istoric', 'Mozarthaus, Domgasse 5, 1010 Wien, Austria', '+4315121791', '4');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('9', 'Picasso Museum', 'Muzeu', 'Carrer de Montcada, 15-23, 08003 Barcelona, Spain', '+34932563000', '1');

INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('10', 'Opera de Stat din Viena', 'Opera', 'Opernring 2, 1010 Wien, Austria', '+431514442250', '4');

**Introducerea datelor in tabelul TRANSPORTURI:**

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('1', 'aeriana', 'Wizz-Air', 'Cluj-Napoca', '1');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('2', 'rutiera', 'Romfor', 'Bucuresti', '4');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('3', 'aeriama', 'Tarom', 'Bucuresti', '2');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('4', 'feroviara', 'SNCF', 'Timisoara', '2');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('5', 'aeriana', 'Wizz-Air', 'Iasi', '1');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('6', 'rutiera', 'Atlassib', 'Iasi', '3');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('7', 'feroviara', 'OBB', 'Timisoara', '4');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('8', 'rutiera', 'Euro-trans', 'Bucuresti', '3');

INSERT INTO transporturi (id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie) VALUES ('9', 'feroviara', 'RENFE', 'Bucuresti', '1');

INSERT INTO transporturi(id\_transport, modalitate, companie, locul\_plecarii, id\_destinatie)VALUES('10', 'aeriana', 'Delta', 'Bucuresti', '2');

**Introducerea datelor in tabelul CAZARI:**

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('1', 'Hôtel Saint Marcel', '43 Boulevard Saint-Marcel, 13th arr., 75013 Paris, France', '+58739573824','7', 'sala de sport, wi-fi, loc pentru nefumatori, parcare', '2');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('2', 'Hotel Eden', '110 rue Blomet, 15th arr., 75015 Paris, France', 'hotel.eden@gmail.com', '8.4', 'baie proprie, room service, piscina, wi-fi, parcare', '2');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('3', 'Hotel Pacific', 'Viale Delle Medaglie Italy', '+57358939479', '8', 'acces acoperis, wi-fi, primeste animale', '3');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('4', 'Hotel Paxton Barcelona', '160 Carrer de Llull, Sant Martí, 08005 Barcelona, Spain', 'HotelPaxtomB@gamil.com', '9.4', 'baie proprie, mic-dejun, piscina, bar, wi-fi, parcare', '1');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('5', 'Hostal Orleans', 'Avenida Marqués de la Argentera, 08003 Barcelona, Spain', 'ORLEANS@gamil.com', '8.3', 'room service, wi-fi, camere pentru familie', '1');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('6', 'Hotel Daniel Vienna', 'Landstraßer Gürtel 5, 1030 Vienna, Austria', '+73858395895', '8.3', 'wi-fi, parcare, piscina, baie comuna, camere pentru familie', '4');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('7', 'Hotel Caroline', 'Gudrunstraße 138, 10. Favoriten, 1100 Vienna, Austria', 'Caroline.Vienna@gmail.com', '9.8', 'wi-fi, parcare, baie proprie, camere de familie, piscina, spa', '4');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('8', 'Hotel Lutetia', '45 Boulevard Raspail, 6th arr., 75006 Paris, France', 'LutetiaH@gmail.com', '8.7', 'lift, wi-fi, baie comuna, acces bucatarie comuna', '2');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('9', 'Bonanova Suite', 'Bisbe Sivilla, 7, Barcelona, Spain', '+357391748294', '7.9', 'lift, baie proprie, parcare, wi-fi', '1');

INSERT INTO cazari (id\_cazare, nume, adresa, contact, calificativ, facilitati, id\_destinatie) VALUES ('10', 'Travelodge Barcelona Poblenou', 'Llull, 170, Sant Martí, 08005 Barcelona, Spain', '+694845939879', '8.8', 'primeste animale, wi-fi, parcare', '1');

**Introducerea datelor in tabelul OFERTE:**

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('1', '120', to\_date('2014-02-23','yyyy-mm-dd'), to\_date('2013-05-23','yyyy-mm-dd'), '2', '3');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('2', '120', to\_date('2019-05-01','yyyy-mm-dd'), to\_date('2015-03-23','yyyy-mm-dd'), '1', '4');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('3', '120', to\_date('2016-08-15','yyyy-mm-dd'), to\_date('2015-07-21','yyyy-mm-dd'), '1', '2');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('4', '120', to\_date('2019-07-19','yyyy-mm-dd'), to\_date('2011-02-23','yyyy-mm-dd'), '3', '3');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('5', '120', to\_date('2018-11-11','yyyy-mm-dd'), to\_date('2012-02-23','yyyy-mm-dd'), '3', '1');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('6', '120', to\_date('2020-01-01','yyyy-mm-dd'), to\_date('2013-02-23','yyyy-mm-dd'), '2', '1');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('7', '120', to\_date('2014-05-18','yyyy-mm-dd'), to\_date('2017-04-17','yyyy-mm-dd'), '1', '2');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('8', '120', to\_date('2014-05-14','yyyy-mm-dd'), to\_date('2016-10-25','yyyy-mm-dd'), '4', '2');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('9', '120', to\_date('2021-12-13','yyyy-mm-dd'), to\_date('2024-10-24','yyyy-mm-dd'), '4', '4');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('10', '120', to\_date('2022-06-11','yyyy-mm-dd'), to\_date('2025-09-22','yyyy-mm-dd'), '3', '4');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('11', '170', to\_date('2022-06-11','yyyy-mm-dd'), to\_date('2025-09-22','yyyy-mm-dd'), '3', '4');

INSERT INTO oferte (id\_oferta, pret\_persoana, activ\_de\_la\_data, activ\_pana\_la\_data, id\_angajat, id\_destinatie) VALUES ('12', '230', to\_date('2022-06-11','yyyy-mm-dd'), to\_date('2025-09-22','yyyy-mm-dd'), '3', '4');

**Introducerea datelor in tabelul asociativ PRIMESC:**

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-05-12','yyyy-mm-dd'),'2', '1');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-06-12','yyyy-mm-dd'),'2', '2');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-07-12','yyyy-mm-dd'), '3', '3');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-08-12','yyyy-mm-dd'), '4', '4');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-09-12','yyyy-mm-dd'), '5', '5');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-10-12','yyyy-mm-dd'), '6', '6');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-12-12','yyyy-mm-dd'), '7', '10');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-01-12','yyyy-mm-dd'), '8', '9');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2015-01-12','yyyy-mm-dd'), '9', '8');

INSERT INTO primesc (data, id\_client, id\_oferta) VALUES (to\_date('2019-11-21','yyyy-mm-dd'), '10', '7');

# Cereri complexe

In continuare, voi implementa 5 subcereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

1. subcereri sincronizate în care intervin cel puțin 3 tabele

In cererea SQL realizata sunt incluse 3 tabele: Oferte, Angajati, Destinatii. O subcerere sincronizata este cea din ultimul ‘AND’, adaugandu-se un filtru de care trec doar daca pret\_persoana este minim.

SELECT o.ID\_oferta, o.pret\_persoana, a.nume, d.tara, d.oras

FROM Oferte o, Angajati a, Destinatii d

WHERE o.id\_angajat = a.id\_angajat

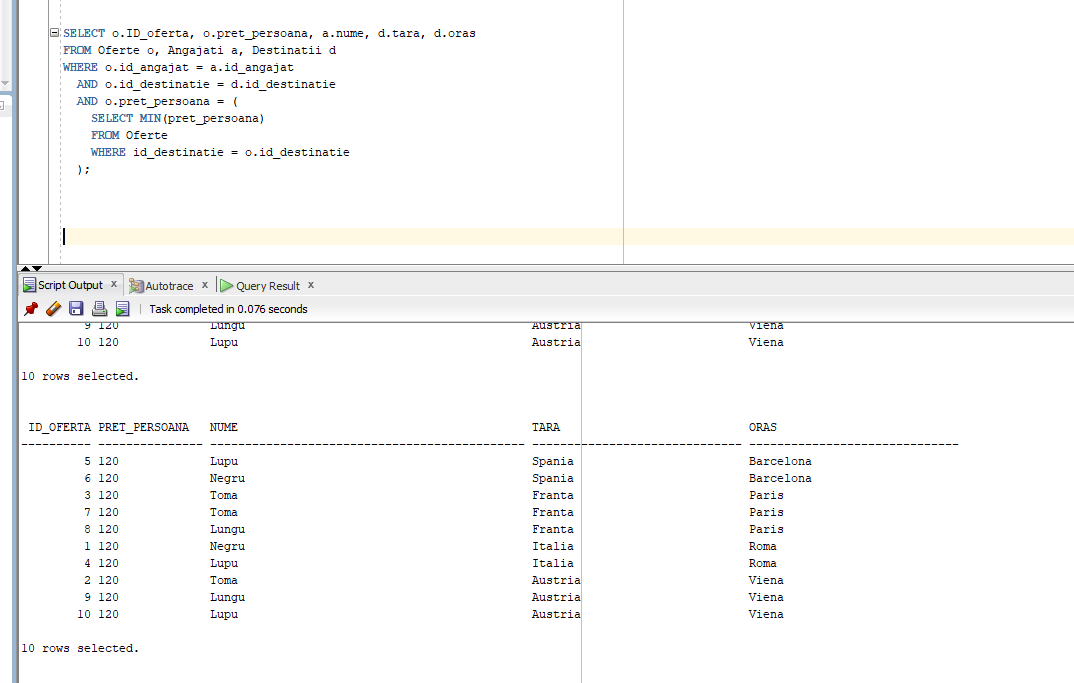
AND o.id\_destinatie = d.id\_destinatie

AND o.pret\_persoana = (

SELECT MIN(pret\_persoana)

FROM Oferte

WHERE id\_destinatie = o.id\_destinatie );

 );

1. subcereri nesincronizate în clauza FROM

In cererea SQL de mai jos sunt 3 subcereri nesincronizate(in clauza FROM). Prima selecteaza doar destinatiile din Frata, a doua filtreaza si returneaza doar cazarile cu calificativ minim 8, iar a treia returneaza ofertele disponibile de la data de 01.01.2022.

SELECT d.tara, c.nume, AVG(o.pret\_persoana) AS medie\_pret

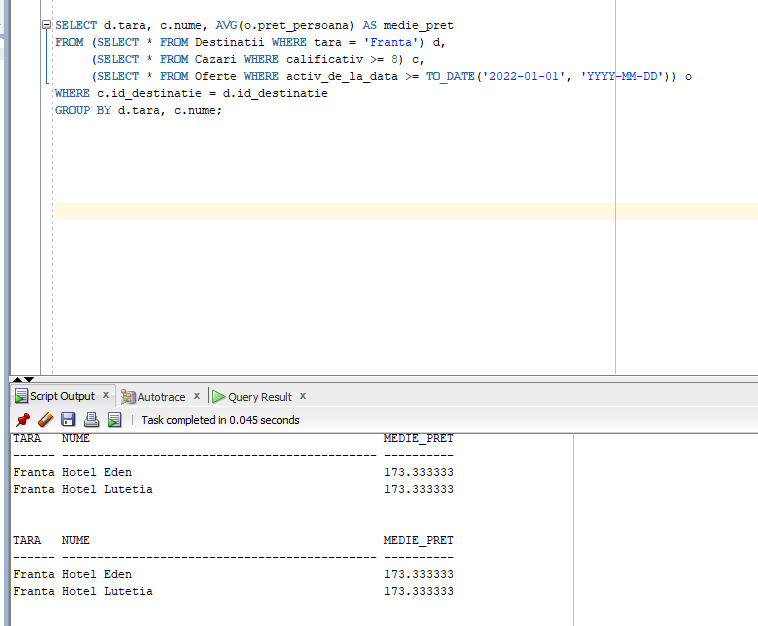
FROM (SELECT \* FROM Destinatii WHERE tara = 'Franta') d,

(SELECT \* FROM Cazari WHERE calificativ >= 8) c,

(SELECT \* FROM Oferte WHERE activ\_de\_la\_data >= TO\_DATE('2022-01-01', 'YYYY-MM-DD')) o

WHERE c.id\_destinatie = d.id\_destinatie

GROUP BY d.tara, c.nume;



1. grupări de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)

În această cerere SQL complexă, avem următoarele elemente:

* Subcererea nesincronizată:

SELECT

id\_cazare,

id\_destinatie,

MAX(calificativ) AS calificativ

FROM

Cazari

GROUP BY

id\_cazare,

id\_destinatie

* Functii de grup:
* COUNT(o.id\_oferta) este utilizată pentru a număra numărul de oferte pentru fiecare destinație.
* AVG(o.pret\_persoana) este utilizată pentru a calcula prețul mediu de persoana al ofertelor pentru fiecare destinație.
* MIN(c.calificativ) este utilizată pentru a obține calificativul minim al cazărilor pentru fiecare destinație.
* Filtrare la nivel de grupuri (filtrare după funcțiile de grup):
  + HAVING COUNT(o.id\_oferta) > 5 filtrează grupurile astfel încât să fie returnate doar destinațiile cu mai mult de 5 oferte.
* ORDER BY numar\_oferte DESC sortează rezultatele în ordine descrescătoare după numărul de oferte.

SELECT

d.tara,

COUNT(o.id\_oferta) AS numar\_oferte,

AVG(o.pret\_persoana) AS pret\_mediu,

MIN(c.calificativ) AS calificativ\_minim

FROM

Destinatii d

JOIN

Oferte o ON d.id\_destinatie = o.id\_destinatie

JOIN

(

SELECT

id\_cazare,

id\_destinatie,

MAX(calificativ) AS calificativ

FROM

Cazari

GROUP BY

id\_cazare,

id\_destinatie

) c ON d.id\_destinatie = c.id\_destinatie

GROUP BY

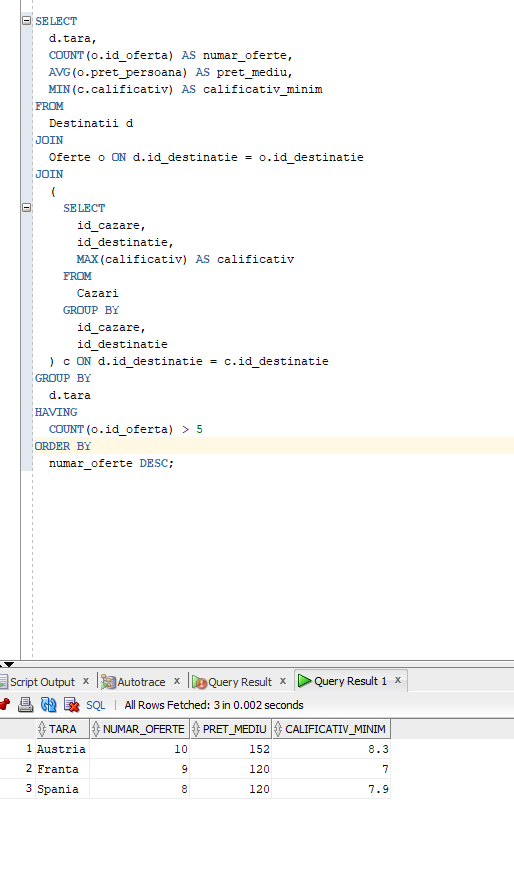
d.tara

HAVING

COUNT(o.id\_oferta) > 5

ORDER BY

numar\_oferte DESC;



1. ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)

În această cerere SQL complexă, avem următoarele elemente:

* Utilizarea funcției NVL:

NVL(o.pret\_persoana, 0) este utilizată pentru a înlocui valorile nule din coloana "pret\_persoana" a tabelului "Oferte" cu valoarea 0.

* Utilizarea funcției DECODE:

DECODE(c.calificativ, NULL, 'N/A', c.calificativ) este utilizată pentru a verifica dacă coloana "calificativ" a tabelului "Cazari" este nulă. Dacă este nulă, se afișează "N/A", în caz contrar se afișează valoarea capacitatii.

* Ordonarea rezultatelor:

ORDER BY d.tara ASC sortează rezultatele în ordine crescătoare după numele tarii de destinație.

Astfel, cererea va returna numele tarii de destinație, pretul ofertei și calificativul cazării pentru fiecare destinație.

SELECT

d.tara,

NVL(o.pret\_persoana, 0) AS pret\_oferta,

DECODE(c.calificativ, NULL, 'N/A', c.calificativ) AS calificativ\_cazare

FROM

Destinatii d

LEFT JOIN

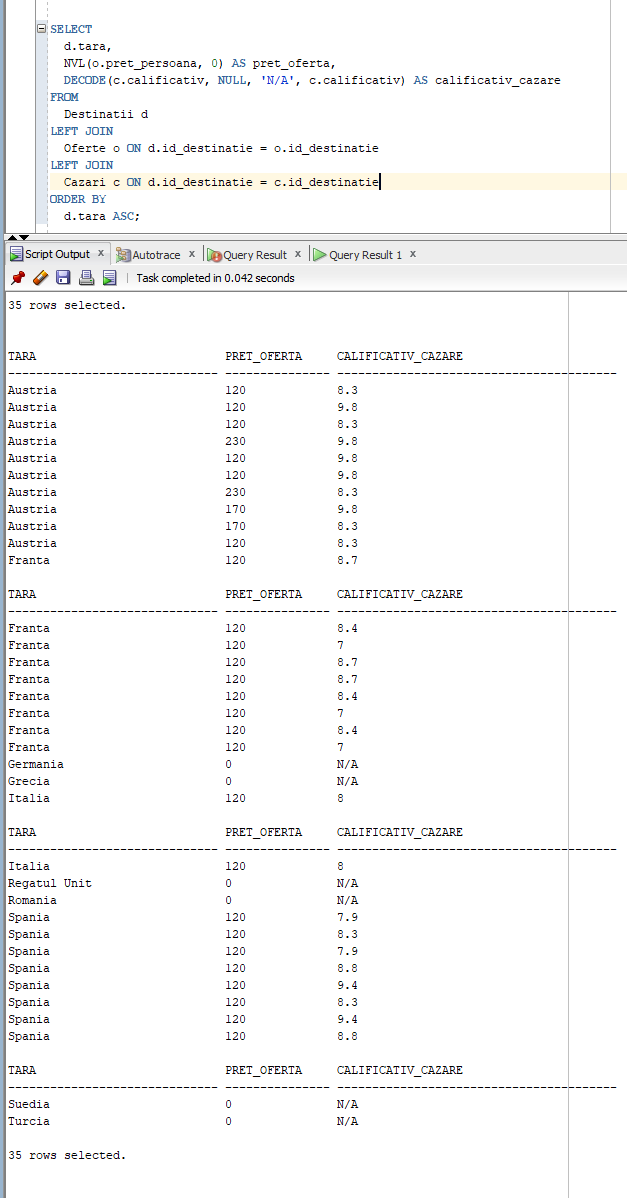
Oferte o ON d.id\_destinatie = o.id\_destinatie

LEFT JOIN

Cazari c ON d.id\_destinatie = c.id\_destinatie

ORDER BY

d.tara ASC;



1. utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
2. utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Punctele 5 si 6 le-am inclus intr-o singura cerere. În această cerere SQL complexă, avem următoarele elemente:

* Blocul de cerere (clauza WITH):

CTE reprezintă o "tabelă virtuală" care combină informațiile din tabelele Oferte, Destinatii și Cazari.

* Utilizarea funcțiilor pe șiruri de caractere:

UPPER este utilizată pentru a converti în majuscule primele trei caractere din numele destinației.

SUBSTR este utilizată pentru a obține o parte din șirul de caractere (primele trei caractere din numele destinației).

* Utilizarea funcțiilor pe date calendaristice:

SYSDATE este o funcție care returnează data și ora curentă.

TO\_CHAR(date, 'DD/MM/YYYY') este utilizată pentru a formata datele calendaristice (activ\_de\_la\_data și activ\_pana\_la\_data) în formatul DD/MM/YYYY.

* Utilizarea expresiei CASE:

CASE este utilizată pentru a atribui o valoare (stare\_oferta) bazată pe condiții. În funcție de data\_inceput și data\_sfarsit, ofertele sunt clasificate ca "Expirate", "Valabile" sau "În curs".

* Ordonarea rezultatelor:

ORDER BY id\_oferta sortează rezultatele în ordine crescătoare după id-ul ofertei.

WITH CTE AS (

SELECT

o.id\_oferta,

d.tara,

c.nume,

o.activ\_de\_la\_data,

o.activ\_pana\_la\_data

FROM

Oferte o

JOIN

Destinatii d ON o.id\_destinatie = d.id\_destinatie

JOIN

Cazari c ON d.id\_destinatie = c.id\_destinatie

)

SELECT

id\_oferta,

tara,

nume,

TO\_CHAR(activ\_de\_la\_data, 'DD/MM/YYYY') AS data\_inceput\_formatata,

TO\_CHAR(activ\_pana\_la\_data, 'DD/MM/YYYY') AS data\_sfarsit\_formatata,

UPPER(SUBSTR(tara, 1, 3)) AS prefix\_destinatie,

LENGTH(nume) AS lungime\_cazare,

CASE

WHEN activ\_de\_la\_data < SYSDATE THEN 'Expirată'

WHEN activ\_pana\_la\_data > SYSDATE THEN 'Valabilă'

ELSE 'În curs'

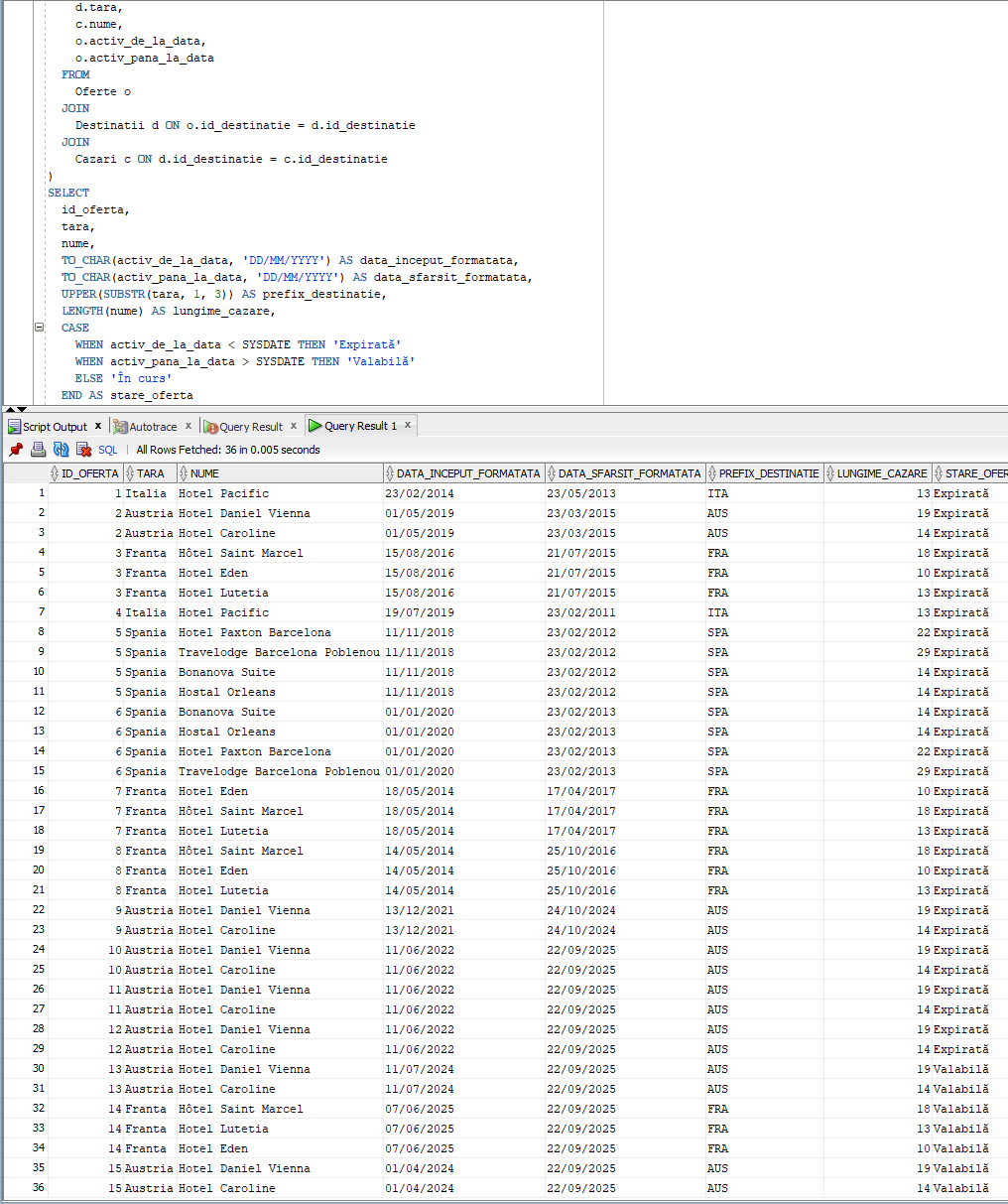
END AS stare\_oferta

FROM

CTE

ORDER BY

id\_oferta;



# Realizarea normalizării până la forma normală 3(FN1-FN2-FN3 )

* Forma Normală 1 (FN1):

În această etapă, identificăm și separăm valorile atomice unice în tabele individuale. De asemenea, adăugăm cheile primare pentru fiecare tabelă. Baza de date inițială nu conține informații redundante sau dependențe funcționale, deci deja se află în FN1.

* Forma Normală 2 (FN2):

În această etapă, eliminăm dependențele funcționale parțiale prin descompunerea tabelelor în entități separate. Pentru a atinge FN2, trebuie să identificăm dependențele funcționale parțiale din baza de date.

Observăm că tabela "Oferte" are o dependență funcțională parțială, deoarece "pret\_persoana" și "activ\_de\_la\_data" depind doar de "id\_oferta". Pentru a corecta această dependență, vom crea o nouă tabelă care va conține doar aceste atribute și cheia primară asociată:

Tabelul "DetaliiOferta":

id\_oferta (FK, PK)

pret\_persoana

activ\_de\_la\_data

După această modificare, tabela "Oferte" va rămâne cu atributele "id\_oferta", "activ\_pana\_la\_data", "id\_angajat" și "id\_destinatie".

* Forma Normală 3 (FN3):

În această etapă, eliminăm dependențele funcționale tranzitive prin descompunerea tabelelor în entități separate. Pentru a atinge FN3, trebuie să identificăm dependențele funcționale tranzitive din baza de date.

Observăm că tabela "Oferte" are o dependență funcțională tranzitivă prin intermediul atributelor "id\_destinatie" și "id\_angajat" către tabela "Destinatii" și, respectiv, tabela "Angajati". Pentru a corecta această dependență, vom crea două noi tabele:

Tabelul "DetaliiDestinatie":

id\_destinatie (PK)

tara

oras

Tabelul "DetaliiAngajat":

id\_angajat (PK)

nume

prenume

telefon

tip\_contract\_munca

data\_angajarii

După această modificare, tabela "Oferte" va avea atributele "id\_oferta", "pret/persoana", "activ\_de\_la\_data", "activ\_pana\_la\_data" și cheile externe "id\_angajat" și "id\_destinatie" care vor referenția noile tabele.

1. Exemplu de non-FN1: Tabelul "Angajati":
   * id\_angajat (PK)
   * nume
   * prenume
   * telefon
   * id\_departament (FK)
   * nume\_departament

În acest exemplu, avem o dependență funcțională parțială, deoarece atributul "nume\_departament" depinde doar de "id\_departament". Pentru a aduce baza de date în Forma Normală 1, trebuie să separăm atributul "nume\_departament" într-un tabel separat, astfel încât să obținem:

Tabelul "Departamente":

* + id\_departament (PK)
  + nume\_departament

1. Exemplu de non-FN2: Tabelul "Angajati":
   * id\_angajat (PK)
   * nume
   * prenume
   * telefon
   * id\_departament (FK)
   * nume\_departament
   * manager\_id (FK)
   * manager\_nume
   * manager\_prenume

În acest exemplu, avem o dependență funcțională tranzitivă, deoarece atributele "manager\_nume" și "manager\_prenume" depind de "manager\_id", care depinde la rândul său de "id\_departament". Pentru a aduce baza de date în Forma Normală 2, trebuie să descompunem tabelul astfel:

Tabelul "Departamente":

* + id\_departament (PK)
  + nume\_departament
  + manager\_id (FK)

Tabelul "Manageri":

* + manager\_id (PK)
  + manager\_nume
  + manager\_prenume

1. Exemplu de non-FN3: Tabelul "Angajati":
   * id\_angajat (PK)
   * nume
   * prenume
   * telefon
   * id\_departament (FK)
   * nume\_departament
   * id\_companie (FK)
   * nume\_companie
   * tara\_companie

În acest exemplu, avem o dependență funcțională tranzitivă prin intermediul atributelor "nume\_companie" și "tara\_companie" către atributul "id\_departament". Pentru a aduce baza de date în Forma Normală 3, trebuie să descompunem tabelul astfel:

Tabelul "Departamente":

* + id\_departament (PK)
  + nume\_departament
  + id\_companie (FK)

Tabelul "Companii":

* + id\_companie (PK)
  + nume\_companie
  + tara\_companie

# Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri

1. Actualizarea pretului unei oferte pentru o destinatie cu ajutorul unei subcereri:

UPDATE Oferte

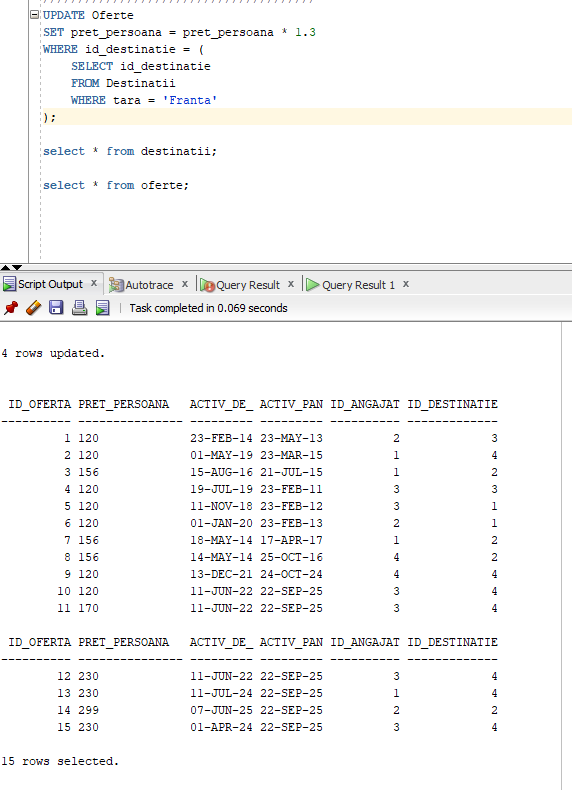
SET pret\_persoana = pret\_persoana \* 1.3

WHERE id\_destinatie = (

SELECT id\_destinatie

FROM Destinatii

WHERE tara = 'Franta'

);

1. Suprimarea unui obiectiv turistic care nu are cazari asociate cu ajutorul unei subcereri:

DELETE FROM Obiective

WHERE id\_obiectiv = (

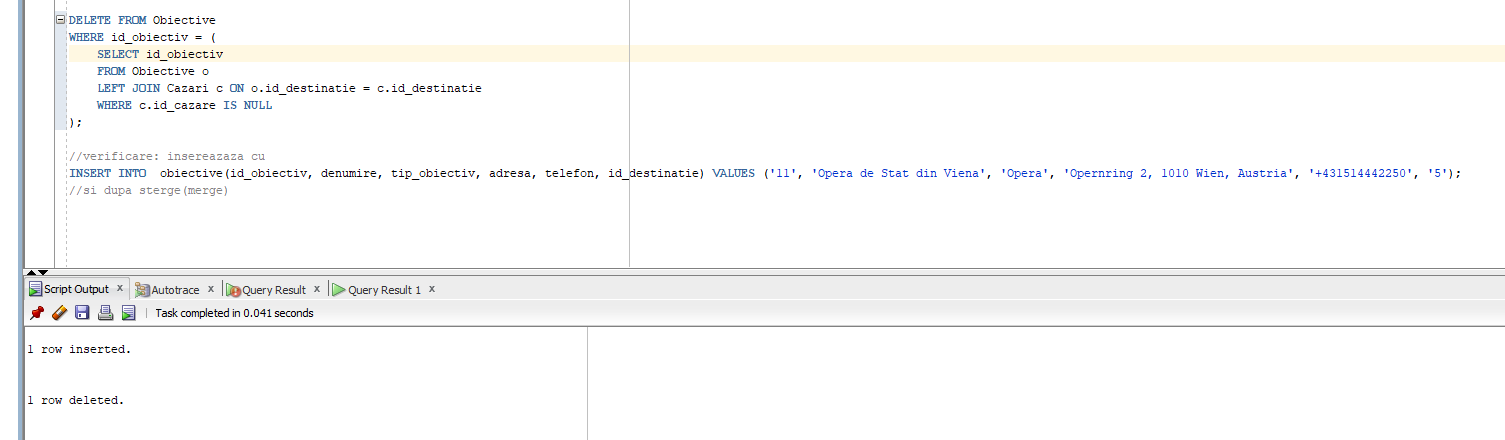
SELECT id\_obiectiv

FROM Obiective o

LEFT JOIN Cazari c ON o.id\_destinatie = c.id\_destinatie

WHERE c.id\_cazare IS NULL

);

Întâi se inserează un obiectiv care nu are cazări asociate: INSERT INTO obiective(id\_obiectiv, denumire, tip\_obiectiv, adresa, telefon, id\_destinatie) VALUES ('11', 'Opera de Stat din Viena', 'Opera', 'Opernring 2, 1010 Wien, Austria', '+431514442250', '5');  
Dupa se ruleaza DELETE-ul și îl elimină.

1. Actualizarea numarului de telefon al unui client, după nume, prenume:

UPDATE Clienti

SET telefon = '0720123456'

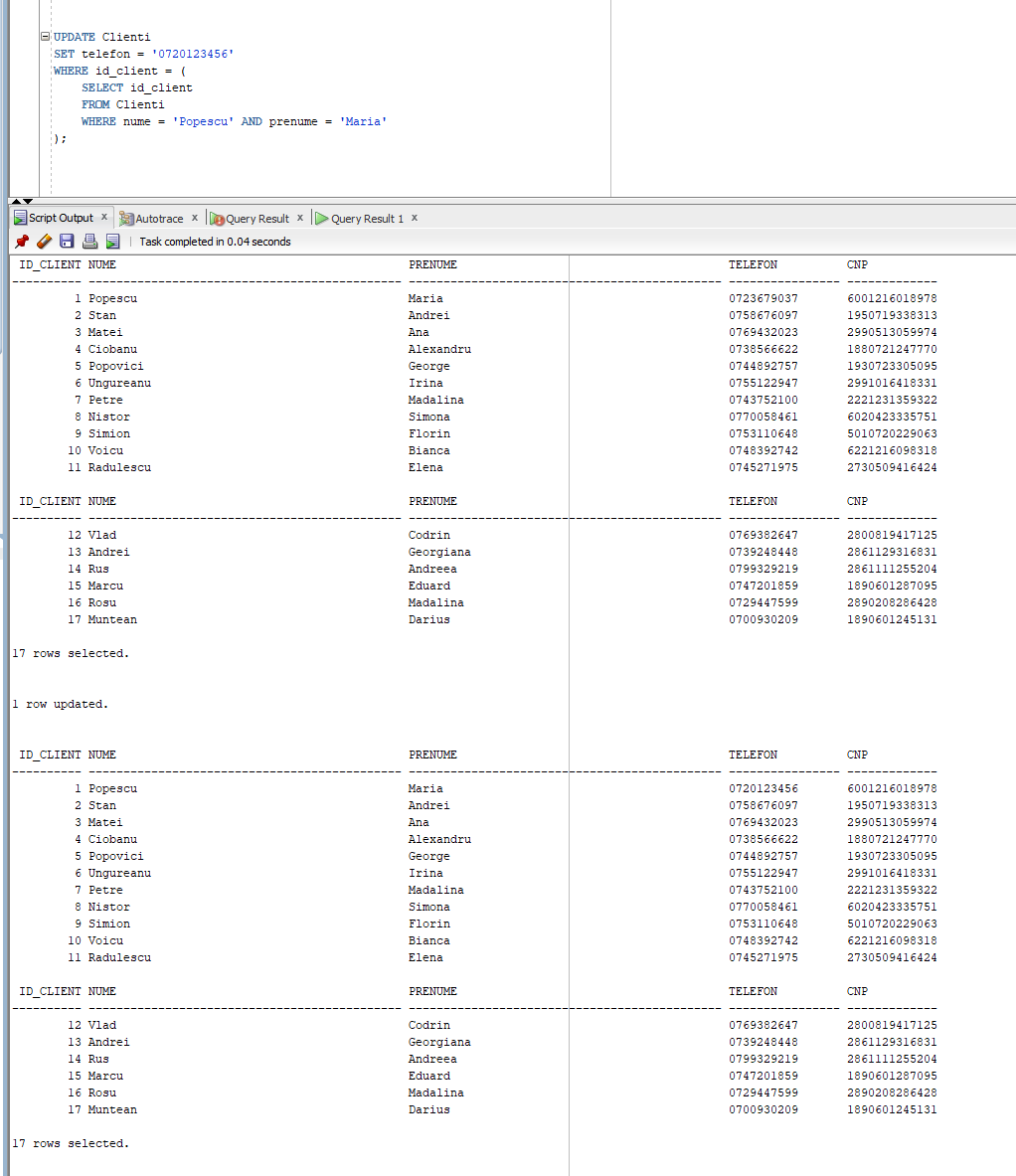
WHERE id\_client = (

SELECT id\_client

FROM Clienti

WHERE nume = 'Popescu' AND prenume = 'Maria'

);



# Crearea unei vizualizari complexe. Operatii LMD permise/nepermise

WITH VizualizareComplexa AS (

SELECT a.id\_angajat, a.nume, a.prenume, a.telefon, c.id\_client, c.nume AS nume\_client, c.prenume AS prenume\_client, c.telefon AS telefon\_client

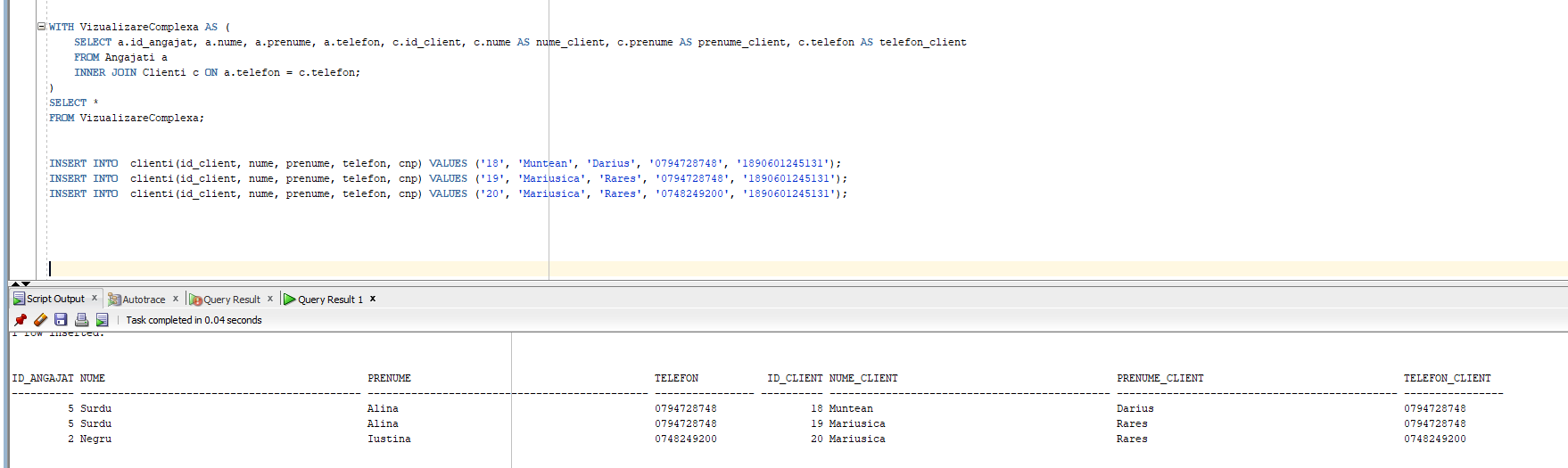
FROM Angajati a

INNER JOIN Clienti c ON a.telefon = c.telefon

)

SELECT \*

FROM VizualizareComplexa;



Inițial, trebuie inserați 3 clienți care au același număr de telefon cu angajații.

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('18', 'Muntean', 'Darius', '0794728748', '1890601245131');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('19', 'Mariusica', 'Rares', '0794728748', '1890601245131');

INSERT INTO clienti(id\_client, nume, prenume, telefon, cnp) VALUES ('20', 'Mariusica', 'Rares', '0748249200', '1890601245131');

* Exemplu de operație LMD permisă pe vizualizarea respectivă:

SELECT id\_angajat, nume, prenume

FROM VizualizareComplexa

WHERE telefon\_client = '0720123456';

Această operație SELECT permite extragerea informațiilor despre angajatul asociat unui anumit client pe baza numărului de telefon al clientului.

* Exemplu de operație LMD nepermisă pe vizualizarea respectivă:

UPDATE VizualizareComplexa

SET nume\_client = 'Smith'

WHERE id\_angajat = 1;

Această operație UPDATE nu este permisă pe vizualizarea complexă, deoarece vizualizarea este rezultatul unei interogări, iar modificările trebuie să fie efectuate direct pe tabelele de bază.

# Outer-Join pe minimum 4 tabele, division și analiza top-n.

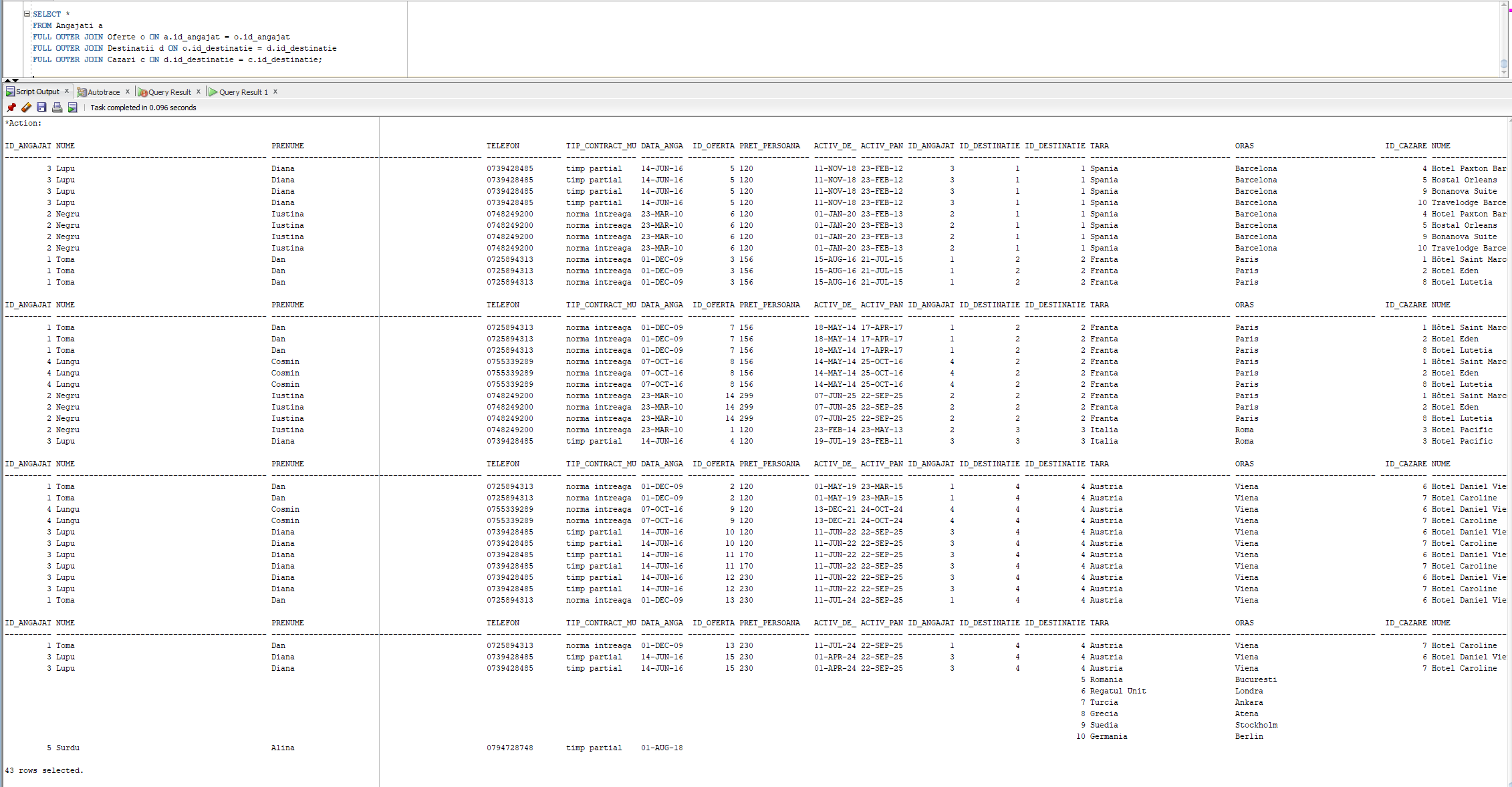
* Obțineți toate informațiile despre angajați și destinații, inclusiv ofertele și cazările disponibile, indiferent dacă există potriviri între acestea.

SELECT \*

FROM Angajati a

FULL OUTER JOIN Oferte o ON a.id\_angajat = o.id\_angajat

FULL OUTER JOIN Destinatii d ON o.id\_destinatie = d.id\_destinatie

FULL OUTER JOIN Cazari c ON d.id\_destinatie = c.id\_destinatie;

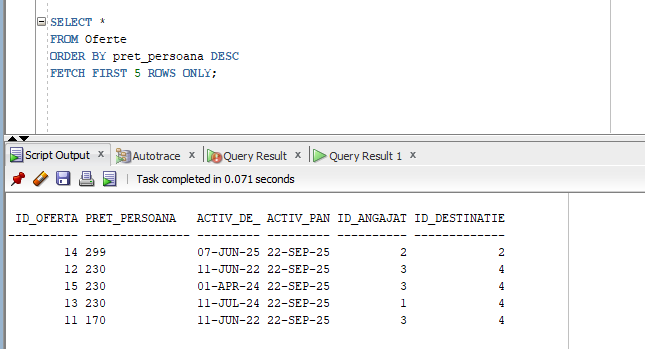
* Obțineți primele 5 cele mai scumpe oferte disponibile, ordonate descrescător după preț\_persoană

SELECT \*

FROM Oferte

ORDER BY pret\_persoana DESC

FETCH FIRST 5 ROWS ONLY;



* Obțineți numele și prenumele angajaților care au gestionat toate destinațiile disponibile(cu division).

SELECT a.nume, a.prenume

FROM Angajati a

WHERE NOT EXISTS (

SELECT \*

FROM Destinatii d

WHERE NOT EXISTS (

SELECT \*

FROM Oferte o

WHERE o.id\_angajat = a.id\_angajat

AND o.id\_destinatie = d.id\_destinatie

)

);

