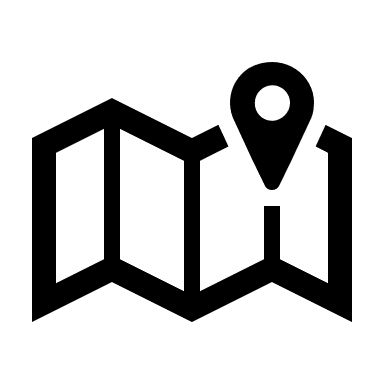
**Proiect Baze de Date**

**Gestiunea unei agenții de turism**

****

# **Cerințele 1+2**



În zilele noastre tendința populație este spre a-și petrece concediul în destinații menite relaxării și deconectării de viața cotidiană. Mai mult de atât, în rândul tinerilor, din ce în ce mai mulți doresc să aibă parte de vacanțe în zone exotice sau cu număr de atracții turistice ridicat.

Agenția de turism asupra căreia am creat baza de date oferă pachete de vacanță și are mai mulți angajați. Un pachet de vacanță oferă transport și hotel într-o anumită locație, și orice tip de pachet poate fi achiziționat de către un cumpărător prin semnarea unui contract de către acesta și un angajat de tip agent. În contract se află id\_ul cumpărătorului, al agentului și al tipului de pachet pe care dorește să îl cumpere, plus alte detalii cu legătură la plată și data plecării și a întoarcerii din vacanță.

Restricții și reguli:

1. un cumparator poate achizitiona doar cate un pachet pe rand deoarece e vorba de servicii diferite

2. un contract este semnat intre un agent si un cumparator

3. pachetul contine cate o locatie si un hotel

4. un pachet include un singur hotel

5. toti agentii au un salariu de baza, iar peste salariul de baza obtin cate un comision din contracte. Acest comision nu e identic pentru toti agentii(bazat pe cel negociat la inceperea contractului de munca)

6. tipurile de transport sunt doar cele folosite

7. hotelurile inregistrate sunt doar cele care se folosesc in pachete

8. locatiile inregistrate sunt doar cele care se folosesc in pachete

9. id-ul unui contract se schimba la fiecare contract nou, insemnand ca un alt contract poate avea acelasi cumparator sau acelasi tip de pachet inclus

Constrângeri:

Modelul relațional va impune constrângeri de tip:

- cheie primară (PK)

- cheie străină (FK)

# **Cerința 3**

CONTRACT(#id\_contract) - entitate ce va conține informații despre părțile care semnează contractul, cât și detalii despre pachetul oferit

CUMPARATOR(#id\_cumparator) - entitate ce va conține informații despre cumpărătorul unui pachet

ANGAJATI(#id\_angajat) - entitate ce va conține informații despre angajații agenției și rolul lor

AGENT - subentitatea AGENT, subentitatea supraentității ANGAJAȚI preia toate valoriile specificie unui angajat, plus comisionul care este specific doar angajaților de tip AGENT. Apare separat de entitatea ANGAJAȚI în diagrame deoarece un contract poate fi semnat doar de către un angajat de tip agent.

PACHET(#id\_pachet) - entitate ce va conține informații despre pachetul pe care un cumpărător ar vrea să îl achiziționeze

TRANSPORT(#id\_transport) - entitate ce va conține informații despre tipurile de transport

HOTEL(#id\_hotel) - entitate ce va conține informații despre hotel și locația unde se află

LOCATIE(#id\_locatie) - entitate ce va conține informații despre țara și orașul unde se află o locație

# **Cerința 4**

Relații:

1. CUMPARATOR-CONTRACT

- un cumpărător poate semna mai multe contracte, dar un contract este semnat de către un singur cumpărător

- tip relație: one-many

- cardinalitate: 1-M(1)

2. AGENT-CONTRACT

- un agent poate semna mai multe contracte, dar un contract este semnat de către un singur agent

- tip relație: one-many

- cardinalitate: 1-M(0)

3. ANGAJATI-AGENT

- un angajat poate fi agent, dar un agent sigur face parte dintre angajați

-tip relație: one-one

cardinalitate: 1-1(0)

4. PACHET-CONTRACT

- un pachet poate fi achiziționat și să apară pe mai multe contracte, dar un contract conține un singur pachet

- tip relație: one-one

- cardinalitate: 1-M(0)

5. PACHET\_TRANSPORT

- un pachet poate avea mai multe tipuri de transport diferite, iar tipurile de transport pot corespunde mai multor pachete

- relație: many-many

- cardinalitate: M(1)-M(1)

6. PACHET-HOTEL

- un pachet include un singur hotel, iar un hotel poate fi prezent intr-un singur tip de pachet de vacanta

- relație: one-one

- cardinalitate: 1-1

7. LOCATIE-HOTEL

- o locație oferă mai multe hotele, dar un hotel se află într-o singură locație

- relație: one-many

- cardinalitate: 1-M(1)

# **Cerința 5**

CUMPARATOR

id\_cumparator int PK

nume\_cumparator varchar2(25)    not null

prenume\_cumparator    varchar2(50) not null

numar\_telefon    varchar2(11)

email    varchar2(70)

CONTRACT

id\_contract int PK

id\_cumparator int FK -cumparatorul care semneaza contractul

id\_angajat int FK -angajatul(agentul) care semneaza contractul

id\_pachet int FK - pachetul achizitionat

pret\_contract decimal(10,2) not null

data\_plata datet not null

data \_plecare datet not null

data\_ intoarcere date not null

AGENT

comision decimal(10,2)

ANGAJATI

id\_angajat int PK

nume varchar2(25) not null

prenume varchar2(25) not null

data\_angajare date not null

salariu decimal(10,2)

departament varchar2(30) not null

PACHET

id\_pachet int PK

denumire varchar2(200) not null

numar\_zile int not null

TRANSPORT

id\_transport int PK

modalitate\_transport varchar2(20) not null

HOTEL

id\_hotel int PK

id\_locatie int FK -locatia unde se afla hotelul

id\_pachet int FK

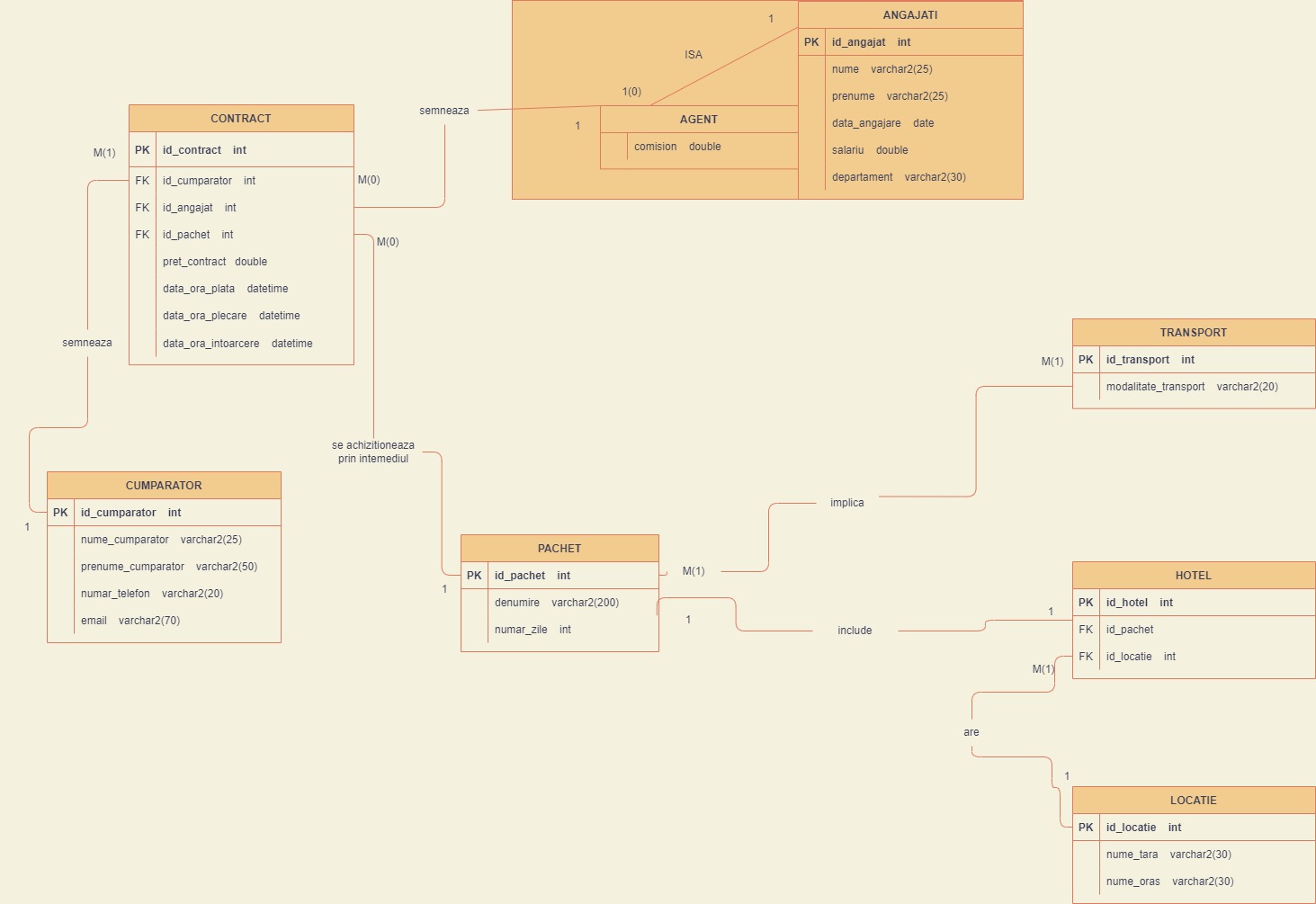
LOCATIE

id\_locatie int PK

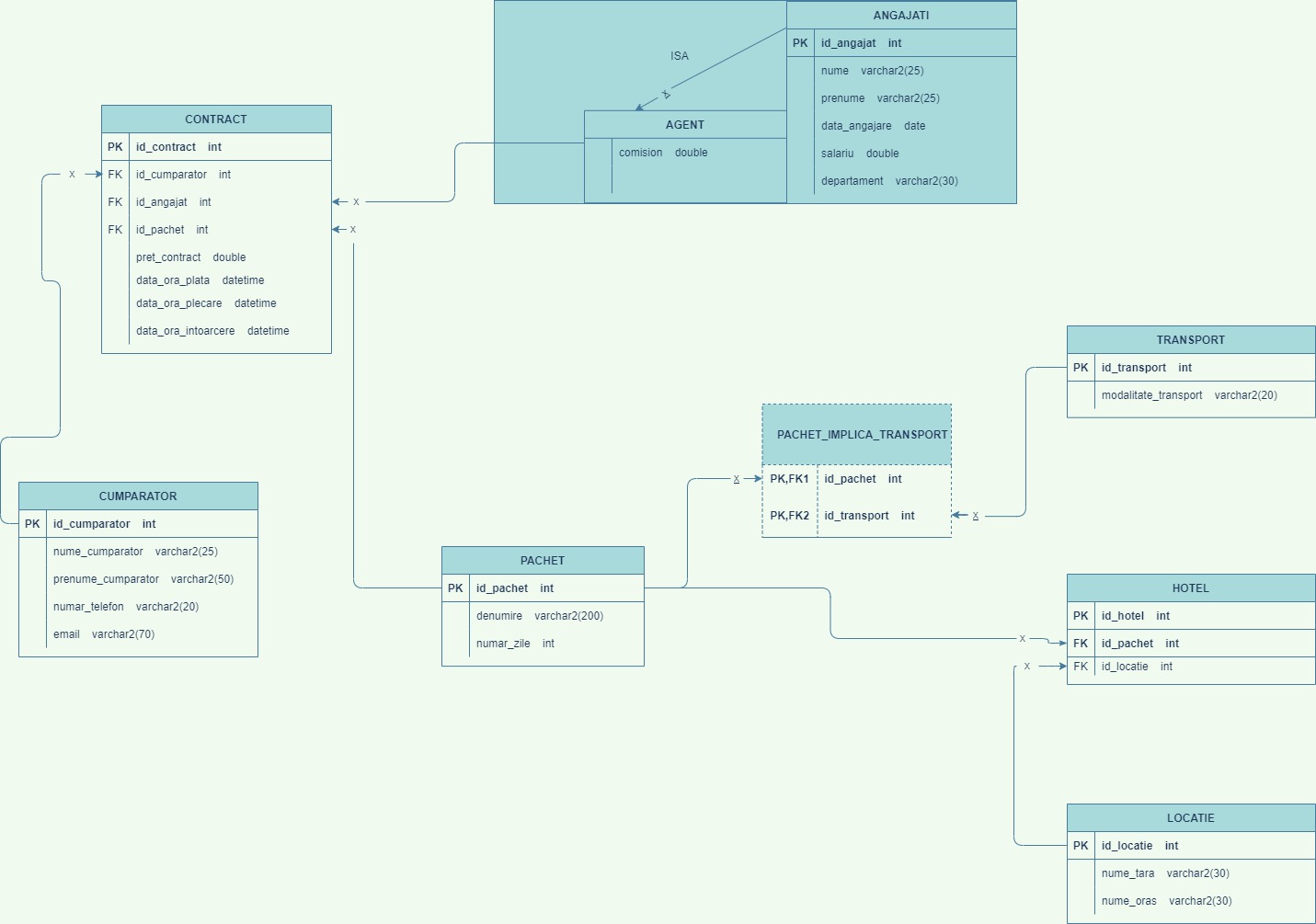
nume\_tara varchar2(30) not null

nume\_oras varchar2(30) not null

# **Cerința 6**



# **Cerința 7**



# **Cerința 8**

CUMPARATOR( PK id\_cumparator, nume\_cumparator, prenume\_cumparator, numar\_telefon, email )

CONTRACT( PK id\_contract int, FK id\_cumparator, FK id\_angajat, FK id\_pachet, pret\_contract, data\_ora\_plata, data\_ora\_plecare, data\_ora\_intoarcere )

AGENT( comision )

ANGAJATI( PK id\_angajat, nume, prenume, data\_angajare, salariu, departament )

PACHET(PK id\_pachet, denumire, numar\_zile )

TRANSPORT( PK id\_transport, modalitate\_transport )

HOTEL( PK id\_hotel, FK id\_locatie, FK id\_pachet )

LOCATIE( PK id\_locatie, nume\_tara, nume\_oras )

PACHET\_IMPLICA\_TRANSPORT( PK, FK1 id\_pachet, PK, FK2 id\_transport )

# **Cerința 9**

EXEMPLE TRANSFORMARI PE BAZA DIAGRAMEI

**(FN1)**O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizabilă. Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.

NON-FN1

Ex: Un cumparator care are mai multe numere de telefon

|  |  |
| --- | --- |
| NUME\_CUMPARATOR | NUMAR\_TELEFON |
| Chiriac | 0758349269 |
| Oltean | 0748392587  0769365926 |
| Mocanu | 0759275926 |

FN1

|  |  |
| --- | --- |
| NUME\_CUMPARATOR | NUMAR\_TELEFON |
| Chiriac | 0758349269 |
| Oltean | 0748392587 |
| Oltean | 0769365926 |
| Mocanu | 0759275926 |

|  |  |  |
| --- | --- | --- |
| ID\_CUMPARATOR | NUME\_CUMPARATOR | NUMAR\_TELEFON |
| 100 | Chiriac | 0758349269 |
| 101 | Oltean | 0748392587 |
| 102 | Oltean | 0769365926 |
| 103 | Mocanu | 0759275926 |

**(FN2)** O relație se află în a doua formă normal dacă și numai dacă această relație este deja in FN1 si fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară. FN2 interzice existența dependențelor funcționale parțiale în cadrul relației. Daca unul sau mai multe elemente sunt dependente functional numai de o parte a cheii primare, atunci ele trebuie sa fie separate in tabele diferite. Daca tabela are o cheie primara formata din numai un atribut, atunci ea este automat in FN2.

NON-FN2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID\_CONTRACT | ID\_CUMPARATOR | ID\_PACHET | PRET\_CONTRACT | DATA\_PLATA |
| 200 | 100 | 10 | 1500 | 22/11/2021 |
| 201 | 101 | 11 | 1750 | 01/10/2021 |
| 202 | 102 | 10 | 1500 | 23/04/2022 |

FN2

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_CONTRACT | ID\_CUMPARATOR | ID\_PACHET | DATA\_PLATA |
| 200 | 100 | 10 | 22/11/2021 |
| 201 | 101 | 11 | 01/10/2021 |
| 202 | 102 | 10 | 23/04/2022 |

|  |  |
| --- | --- |
| ID\_PACHET | PRET\_CONTRACT |
| 10 | 1500 |
| 11 | 1750 |

**(FN3)** O relație este în a treia formă normală dacă și numai dacă este in FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

NON-FN3

|  |  |  |
| --- | --- | --- |
| ID\_PACHET | TARA | ORAS |
| 105 | FRANTA | PARIS |
| 106 | ANGLIA | LONDRA |
| 107 | FRANTA | PARIS |

FN3

|  |  |
| --- | --- |
| ID\_PACHET | TARA |
| 105 | FRANTA |
| 106 | ANGLIA |
| 107 | FRANTA |

|  |  |
| --- | --- |
| TARA | ORAS |
| FRANTA | PARIS |
| ANGLIA | LONDRA |

# **Cerința 10**

create sequence sec\_contract

increment by 1

start with 400

minvalue 1

maxvalue 10000

nocycle;

# **Cerința 11**

create table pachet(

id\_pachet int not null primary key,

denumire varchar2(200) not null,

numar\_zile int not null);

create table locatie(

id\_locatie int not null primary key,

nume\_tara varchar2(30) not null,

nume\_oras varchar2(30) not null);

create table hotel(

id\_hotel int not null primary key,

id\_locatie int not null,

id\_pachet int not null,

foreign key (id\_locatie) references locatie(id\_locatie),

foreign key (id\_pachet) references pachet(id\_pachet));

create table cumparator(

id\_cumparator int not null primary key,

nume\_cumparator varchar2(25) not null,

prenume\_cumparator varchar2(50) not null,

numar\_telefon varchar2(11),

email varchar2(70));

create table angajati(

id\_angajat int not null primary key,

nume varchar2(25) not null,

prenume varchar2(25) not null,

data\_angajare date not null,

salariu decimal(10,2),

departament varchar2(30) not null,

comision decimal(10,2));

create table contract(

id\_contract int not null primary key,

id\_cumparator int not null,

id\_angajat int not null,

id\_pachet int not null,

pret\_contract decimal(10,2) not null,

data\_plata date not null,

data\_plecare date not null,

data\_intoarcere date not null,

foreign key (id\_cumparator) references cumparator(id\_cumparator),

foreign key (id\_angajat) references angajati(id\_angajat),

foreign key (id\_pachet) references pachet(id\_pachet));

create table transport(

id\_transport int not null primary key,

modalitate\_transport varchar2(20) not null);

create table pachet\_implica\_transport(

id\_pachet int not null,

id\_transport int not null,

constraint pk\_pachet\_transport primary key(id\_pachet, id\_transport));

alter table pachet\_implica\_transport add foreign key(id\_pachet) references pachet(id\_pachet);

alter table pachet\_implica\_transport add foreign key(id\_transport) references transport(id\_transport);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(100, 'Ionescu', 'Raluca', to\_date('05/13/2021', 'MM/DD/RRRR'), 2000, 'AGENT', 15);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(101, 'Popescu', 'Mihai', to\_date('05/08/2021', 'MM/DD/RRRR'), 2000, 'AGENT', 17);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(102, 'Ionescu', 'Stefania', to\_date('06/18/2021', 'MM/DD/RRRR'), 2000, 'AGENT', 20);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(103, 'Barbu', 'Maria',to\_date('05/18/2011', 'MM/DD/RRRR'), 2000, 'AGENT', 20.5);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(104, 'Preda', 'Roxana', to\_date('05/18/2012', 'MM/DD/RRRR'), 2000, 'AGENT', 10);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(105, 'Stefanescu', 'Simona', to\_date('05/18/2020', 'MM/DD/RRRR'), 2500, 'IT', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(106, 'Andrei', 'Alexandru', to\_date('05/10/2021', 'MM/DD/RRRR'), 2500, 'IT', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(107, 'Ion', 'Raul', to\_date('09/18/2021', 'MM/DD/RRRR'), 1700, 'MARKETING', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(108, 'Ciobanu', 'Raluca', to\_date('05/22/2021', 'MM/DD/RRRR'), 3000, 'MANAGER', null);

insert into angajati(id\_angajat, nume, prenume, data\_angajare, salariu, departament,comision)

values

(109, 'Ciocan', 'Teodora', to\_date('05/08/2020', 'MM/DD/RRRR'), 3000, 'MANAGER', null);

select \* from angajati;

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(200, 'Ion', 'Ionescu');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(201, 'Irina', 'Alina');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(202, 'Paun', 'Ionela');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(203, 'Iancu', 'Emanuel');

insert into cumparator(id\_cumparator, nume\_cumparator, prenume\_cumparator)

values

(204, 'Preda', 'Irina');

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(300, 'Paris', 4);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(301, 'Londra', 5);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(302, 'Brasov', 7);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(303, 'Roma', 4);

insert into pachet( id\_pachet, denumire, numar\_zile)

values

(304, 'Madrid', 3);

insert into transport(id\_transport, modalitate\_transport)

values

(500, 'avion');

insert into transport(id\_transport, modalitate\_transport)

values

(501, 'tren');

insert into transport(id\_transport, modalitate\_transport)

values

(502, 'autobuz');

insert into transport(id\_transport, modalitate\_transport)

values

(503, 'vapor');

insert into transport(id\_transport, modalitate\_transport)

values

(504, 'masina');

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(300, 500);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(301, 501);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(302, 502);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(303, 503);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(304, 504);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(300, 504);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(301, 503);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(302, 504);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(303, 502);

insert into pachet\_implica\_transport(id\_pachet, id\_transport)

values

(304, 501);

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( sec\_contract.nextval, 200, 100, 300, 1900,to\_date('05/08/2020', 'MM/DD/RRRR'),to\_date('05/10/2020', 'MM/DD/RRRR'),to\_date('05/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( sec\_contract.nextval, 201, 101, 301, 1500,to\_date('06/08/2020', 'MM/DD/RRRR'),to\_date('06/10/2020', 'MM/DD/RRRR'),to\_date('06/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( sec\_contract.nextval, 202, 102, 302, 1005,to\_date('07/08/2020', 'MM/DD/RRRR'),to\_date('07/10/2020', 'MM/DD/RRRR'),to\_date('07/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( sec\_contract.nextval, 203, 103, 303, 1000,to\_date('08/08/2020', 'MM/DD/RRRR'),to\_date('08/10/2020', 'MM/DD/RRRR'),to\_date('08/15/2020', 'MM/DD/RRRR'));

insert into contract(id\_contract, id\_cumparator, id\_angajat, id\_pachet, pret\_contract, data\_plata, data\_plecare, data\_intoarcere)

values ( sec\_contract.nextval, 204, 104, 304, 1070,to\_date('09/08/2020', 'MM/DD/RRRR'),to\_date('09/10/2020', 'MM/DD/RRRR'),to\_date('09/15/2020', 'MM/DD/RRRR'));

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(500, 'Franta', 'Paris');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(503, 'Italia', 'Roma');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(504, 'Spania', 'Madrid');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(501, 'UK', 'Londra');

insert into locatie (id\_locatie, nume\_tara, nume\_oras)

values(502, 'Romania', 'Brasov');

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(600, 500, 300);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(601, 501, 301);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

values

(602, 502, 302);

insert into hotel (id\_hotel, id\_locatie, id\_pachet)

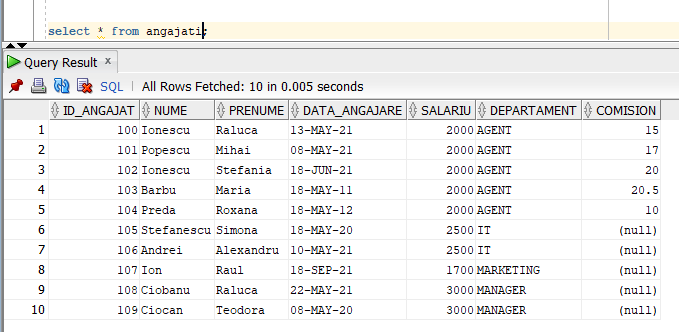
values

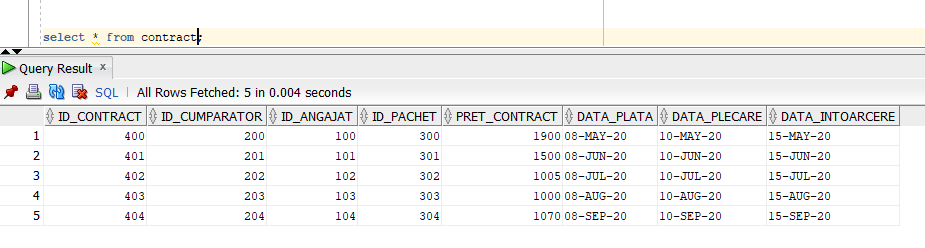
(603, 503, 303);

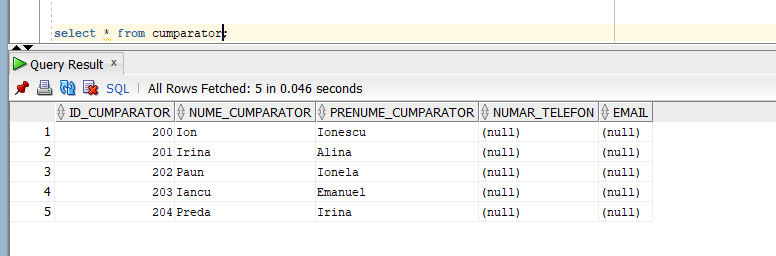
insert into hotel (id\_hotel, id\_locatie, id\_pachet)

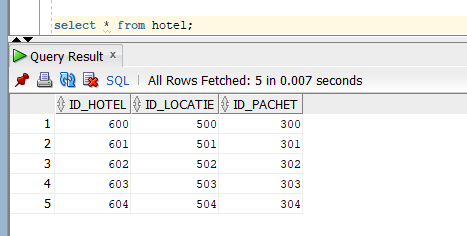
values

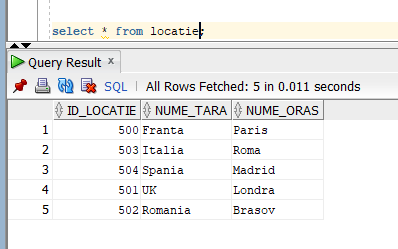
(604, 504, 304);

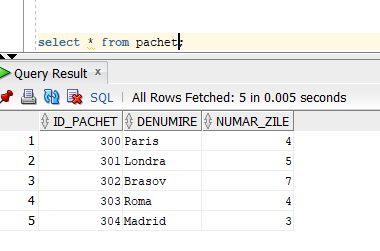


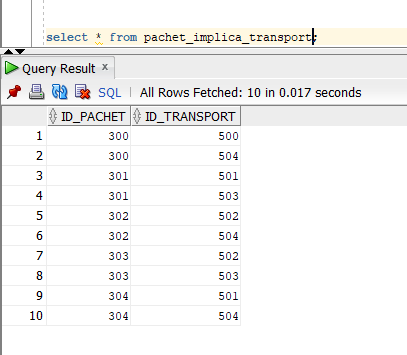


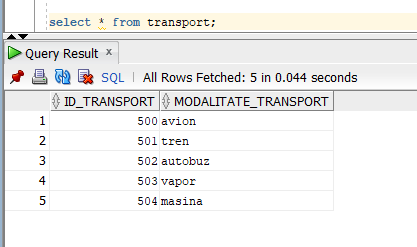












# **Cerința 12**

/\*selecteaza numele angajatului, pretul contractului semnat, denumirea

pachetului din contract, id-ul hotelului din pachet si lunile de la plata contractului unde angajatul

are IO in nume si data platii contractului este dupa 2001\*/

select a.nume, c.data\_plata, p.denumire, h.id\_hotel,round( MONTHS\_BETWEEN(SYSDATE,c.data\_plata)) as Luni\_de\_la\_plata

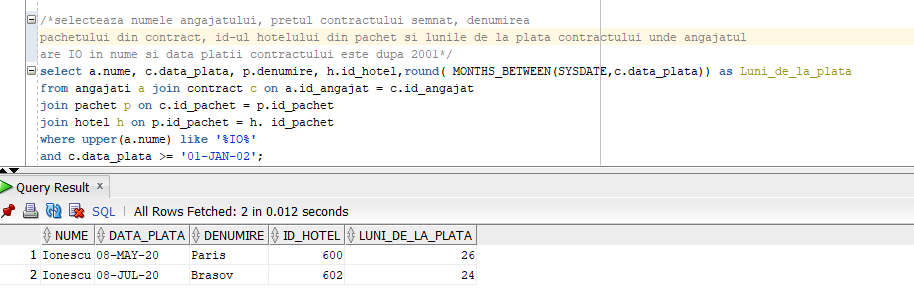
from angajati a join contract c on a.id\_angajat = c.id\_angajat

join pachet p on c.id\_pachet = p.id\_pachet

join hotel h on p.id\_pachet = h. id\_pachet

where upper(a.nume) like '%IO%'

and c.data\_plata >= '01-JAN-02';



/\*sa se selecteze numele angajatului, pretul contractului semnat de acesta si denumirea pachetului din contract,

acolo unde pretul contractului se afla in media salariilor angajatilor din fiecare departament care au un salariu mai mare

de 1700 de lei si din care se scade 1000 din media calculata pe fiecare departament si denumirea pachetului este Londra

sau Roma\*/

select a.nume, c.pret\_contract, p.denumire

from angajati a join contract c on a.id\_angajat = c.id\_angajat

join pachet p on c.id\_pachet = p.id\_pachet

where c.pret\_contract in(

select avg(salariu)-1000

from angajati

where salariu > 1700

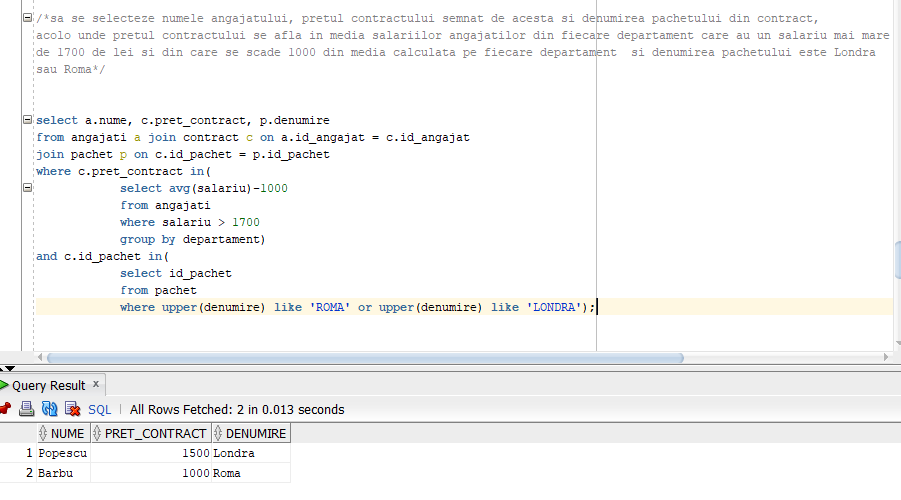
group by departament)

and c.id\_pachet in(

select id\_pachet

from pachet

where upper(denumire) like 'ROMA' or upper(denumire) like 'LONDRA');



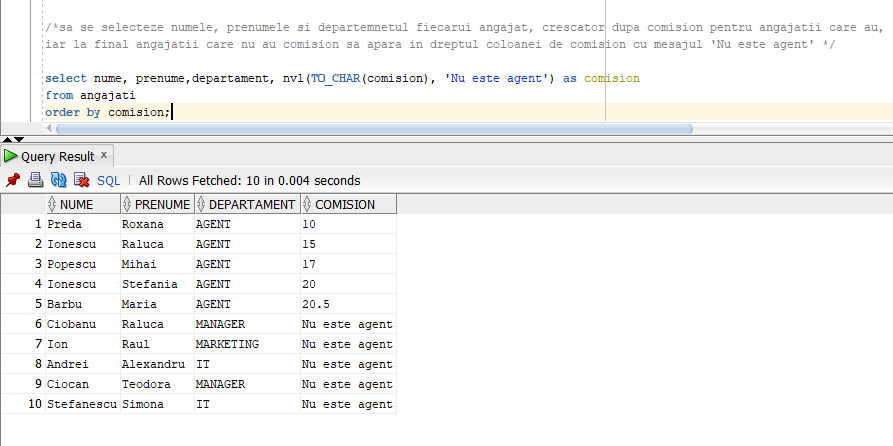
/\*sa se selecteze numele, prenumele si departemnetul fiecarui angajat, crescator dupa comision pentru angajatii care au,

iar la final angajatii care nu au comision sa apara in dreptul coloanei de comision cu mesajul 'Nu este agent' \*/

select nume, prenume,departament, nvl(TO\_CHAR(comision), 'Nu este agent') as comision

from angajati

order by comision;



/\*sa se selecteze numele, prenumele, departamentul, data angajarii si clasificarea salariului(daca este mai mare sau nu

decat 2500 lei) pentru angajatii care au semnat un contract cu un cumparator al carui prenume incepe cu I \*/

select concat( concat(nume,' '), prenume)as Nume\_Prenume,departament, data\_angajare,

case when salariu < 2500 then 'Salariu sub 2500 lei'

else 'Salariu mai mare sau egal cu 2500 lei'

end as clasificare\_salariu

from angajati a

where a.id\_angajat in(

select c.id\_angajat

from contract c

where a.id\_angajat = c.id\_angajat

and c.id\_cumparator in(

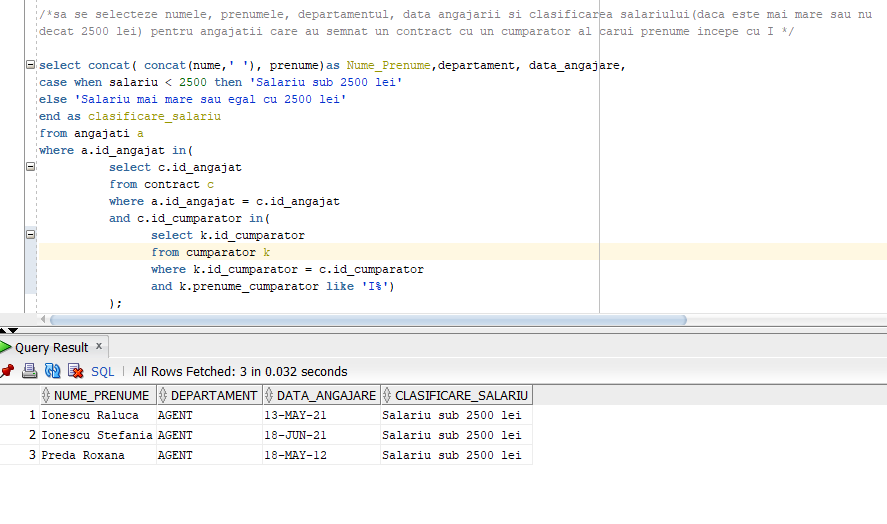
select k.id\_cumparator

from cumparator k

where k.id\_cumparator = c.id\_cumparator

and k.prenume\_cumparator like 'I%')

);



/\*sa se selecteze id\_ul contractului, pretul si un mesaj corespunzator pentru preturile mai mari de 2000 si mai mici,

in cazul in care toate preturile din tabelul de contract ar fi fost mai mari cu 500 de lei \*/

with contract2(id\_contract, pret\_contract) as (

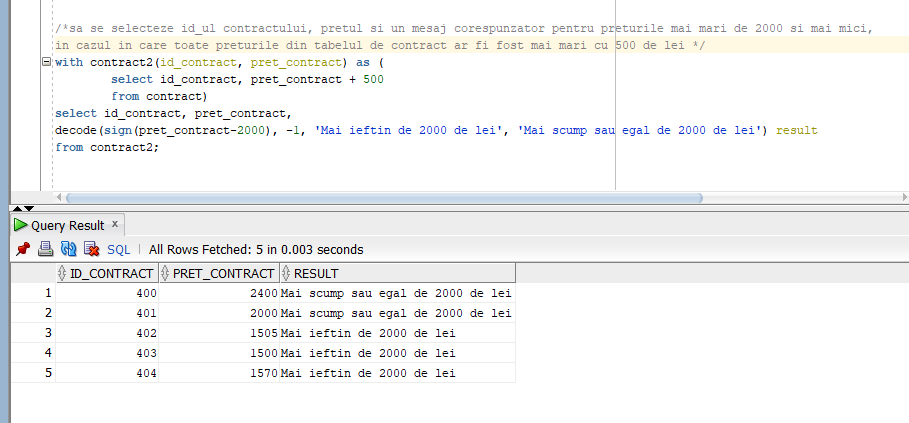
select id\_contract, pret\_contract + 500

from contract)

select id\_contract, pret\_contract,

decode(sign(pret\_contract-2000), -1, 'Mai ieftin de 2000 de lei', 'Mai scump sau egal de 2000 de lei') result

from contract2;



# **Cerința 13**

/Sa se stearga hotelurile care se afla intr-o tara care contine litera a/

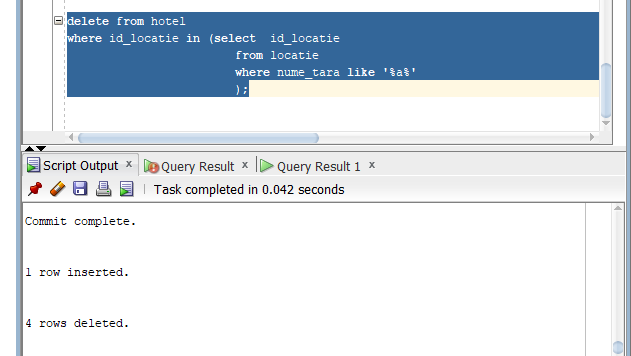
delete from hotel

where id\_locatie in (select id\_locatie

from locatie

where nume\_tara like '%a%'

);



4 linii sterse

/\*Sa se actualizeze modalitatea de transport cu 'masina' pentru pachetele

al caror id este 303 sau 304\*/

update transport

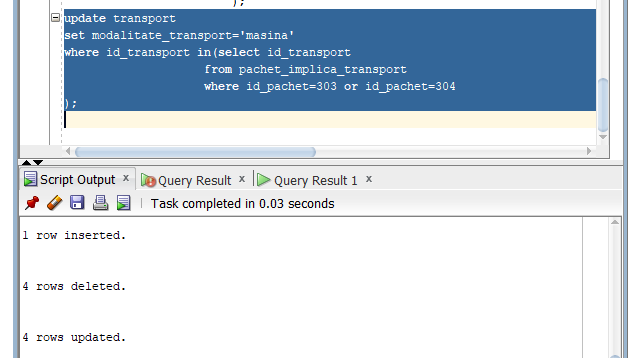
set modalitate\_transport='masina'

where id\_transport in(select id\_transport

from pachet\_implica\_transport

where id\_pachet=303 or id\_pachet=304

);



4 linii actualizate

/Sa se schimbe numarul de zile in 7 al pachetele a caror hoteluri se afla in Madrid/

update pachet

set numar\_zile = 7

where id\_pachet in (select id\_pachet

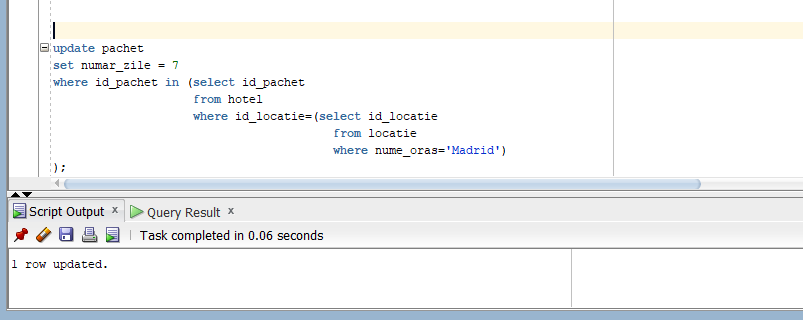
from hotel

where id\_locatie=(select id\_locatie

from locatie

where nume\_oras='Madrid')

);



1 linie actualizata

# **Cerința 16**

Outer join pe 4 tabele

/\* sa se afiseaze pentru fiecare contract denumirea pachetului inclus, id-ul modalitatilor de transport incluse si denumirea acestora\*/

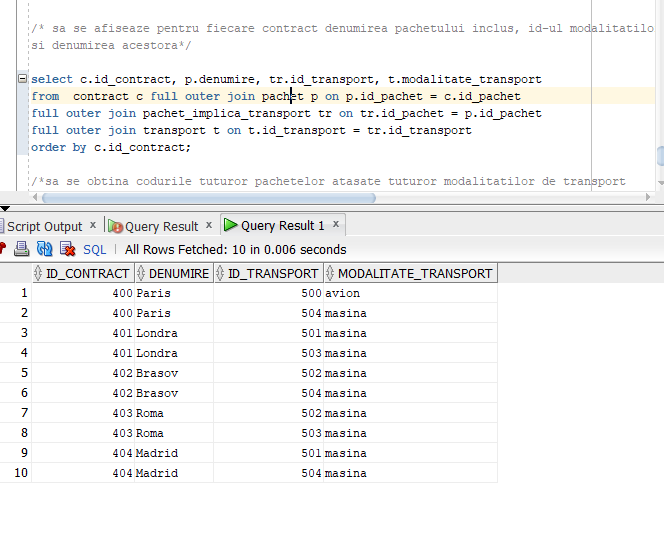
select c.id\_contract, p.denumire, tr.id\_transport, t.modalitate\_transport

from contract c full outer join pachet p on p.id\_pachet = c.id\_pachet

full outer join pachet\_implica\_transport tr on tr.id\_pachet = p.id\_pachet

full outer join transport t on t.id\_transport = tr.id\_transport

order by c.id\_contract



Cereri cu operatia division

/\*sa se obtina codurile tuturor pachetelor atasate tuturor modalitatilor de transport

care sunt 'avion' \*/

select distinct id\_pachet

from pachet\_implica\_transport a

where not exists

(select 1

from transport p

where upper(modalitate\_transport) like 'AVION'

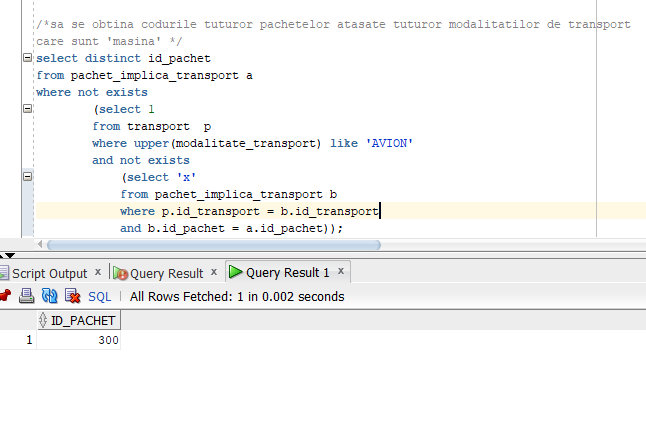
and not exists

(select 'x'

from pachet\_implica\_transport b

where p.id\_transport = b.id\_transport

and b.id\_pachet = a.id\_pachet));



/\* sa se selecteze toate id-urile pachetelor atasate unor modalitati de transport care

au id-ul transportului egal cu 500 \*/

select distinct a.id\_pachet

from pachet\_implica\_transport a

where a.id\_transport in

(select id\_transport

from transport

where id\_transport = 500)

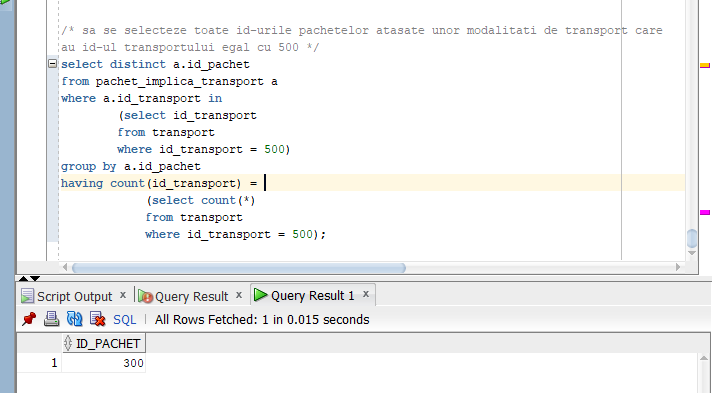
group by a.id\_pachet

having count(id\_transport) =

(select count(\*)

from transport

where id\_transport = 500);



# **Cerința 17**

/\* sa se selecteze id-ul tuturor hotelurilor care se localizeaza in UK \*/

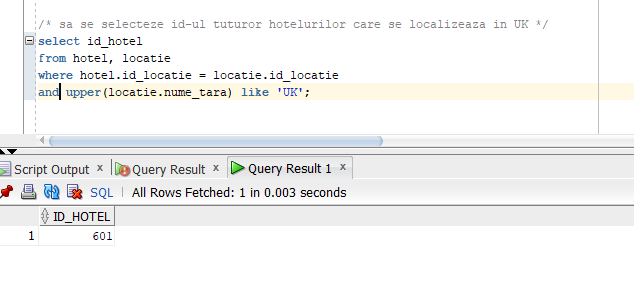
Neoptimizat:

select id\_hotel

from hotel, locatie

where hotel.id\_locatie = locatie.id\_locatie

and upper(locatie.nume\_tara) like 'UK';



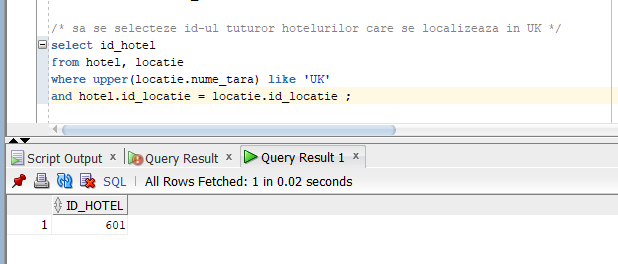
Optimizat:

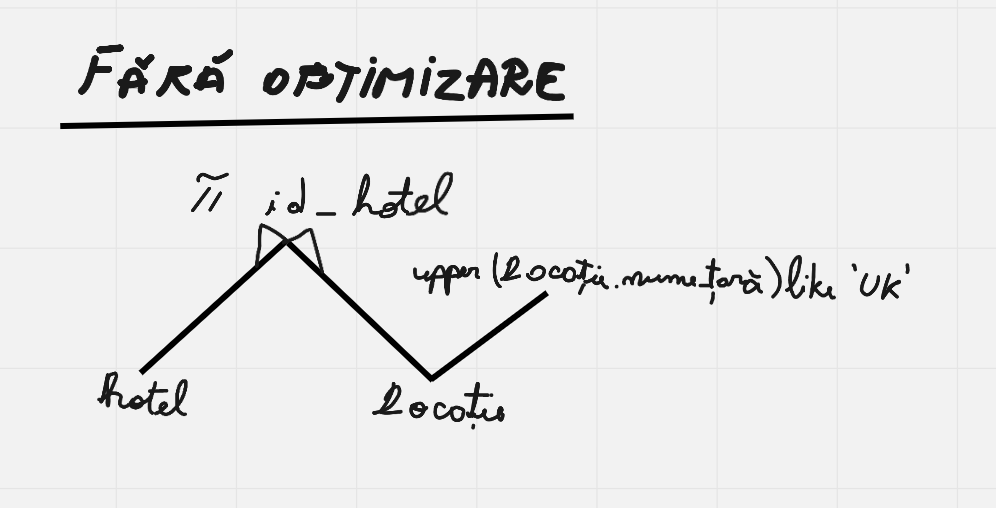
select id\_hotel

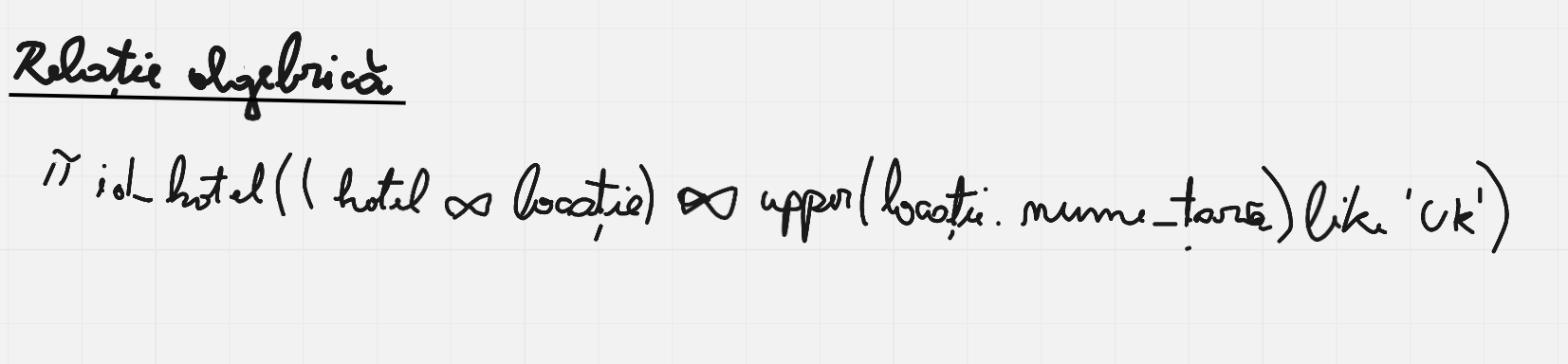
from hotel, locatie

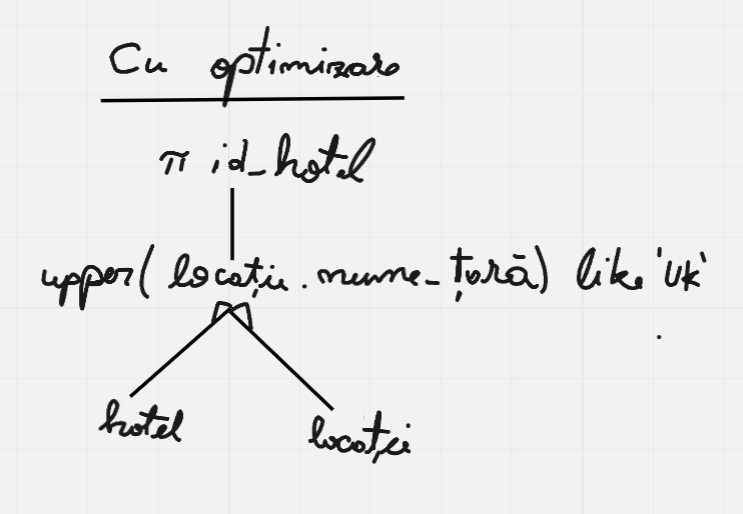
where upper(locatie.nume\_tara) like 'UK'

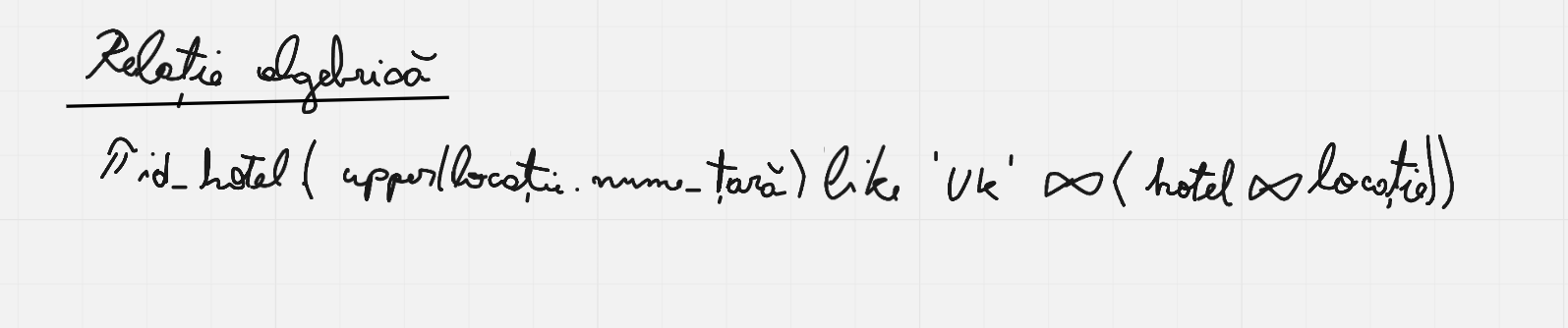
and hotel.id\_locatie = locatie.id\_locatie ;











# **Cerința 18**

**BNCF**

Forma normală Boyce-Codd se bazează pe dependenţele funcţionale care iau în consideraţie toate cheile candidat dintr-o relaţie. Pentru relatiile cu o singura cheie candidat, formele FN3 si BCNF sunt echivalente

Exemplu non-BNCF:

Presupunem tabelul Pachet\_implica\_transport(#id\_pachet, #id\_transport, pret\_transport)

Pentru a fi în BNCF vom împărți tabelul în 2 astfel:

Pachet\_implica\_transport1(#id\_transport, pret\_transport)

Pachet\_implica\_transport2(pret\_transport, id\_pachet)

**FN4**

FN4 elimină redundanțele datorate relațiilor m:n, adică datorate dependenței multiple. O relație este în a patra formă nomală dacă și numai dacă este în BCNF și nu conține relații m:n independente.

Ca exemplu de non-FN4 presupun relația Angajat(#id\_angajat, departament, salariu) mergând pe premisa că un om poate avea două job-uri ( cum un angajat part-time în IT poate să fie și agent) și implicit să aibă două salarii pentru fiecare job al său

Pentru a aduce relația în FN4 o vom descompune în două relații:

Angajat1(#id\_angajat, departament)

Angajat2(#id\_angajat, salariu)

Angajat = join(Angajat1, Angajat2)

**FN5**

O relaţie R este în FN5 (numită şi forma normală proiecţie-uniune) dacă şi numai dacă orice dependenţă de uniune a lui R este o consecinţă a unei chei candidat a lui R. Orice relaţie care este în FN5 este şi în FN4, deoarece fiecare dependenţă multivaloare poate fi privită ca un caz particular de dependenţă de uniune. Orice relaţie poate fi descompusă fără pierderi la uniune într-o mulţime de relaţii care sunt în FN5.

Pentru a preciza dacă o relaţie este în FN5, este suficient să cunoaştem cheile candidate şi toate dependenţele de uniune din R. Aducerea în FN5 presupune eliminarea join dependențelor.

Presupunem relația Locatie(id\_locatie, nume\_tara, nume\_oras) si exista mai multe dependente:

(id\_locatie, nume\_tara), (id\_locatie, nume\_oras), (nume\_tara, nume\_oras), iar soluția ar fi să îl împărțim în 3 tabele

Locatie1(id\_locatie, nume\_tara)

Locatie2(id\_locatie, nume\_oras)

Locatie3(nume\_tara, nume\_oras)

Locatie = join(Locatie1, Locatie2, Locatie3)

**Denormalizare**

Denormalizare Denormalizarea este procesul invers al procesului de normalizare. Denormalizarea funcționează adăugând date redundante sau grupând date pentru a optimiza performanța. Denormalizare are rolul de a realiza executarea mai rapidă a interpgărilor prin introducerea redundanței, punând accentul pe rapiditatea analizei și scăzând numărul de tabele. Motivul efectuării denormalizării este costurile unui produs în procesor de interogare printr-o structură supra-normalizată.

În cadrul bazei de date proiectate de mine ar fi mai costisitor din punct de vedere al timpului să execut mai multe operații de join( cum ar fi exemplul de la FN5 cu Locatie1(id\_locatie, nume\_tara), Locatie2(id\_locatie, nume\_oras), Locatie3(nume\_tara, nume\_oras) ), iar relațiile dintre tabele sunt mai ușor de vizualizat