Proiect Baze de Date

Bază de date pentru o companie aeriană

Nume student: Hodivoianu Anamaria

Grupă: 132

Cuprins

[1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare 2](#_Toc1373046860)

[2. Prezentarea constrângerilor (restricții, reguli) 2](#_Toc1660461344)

[3. Descrierea entităților 2](#_Toc1619581543)

[4. Descrierea relațiilor 3](#_Toc1091589406)

[5. Descrierea atributelor 4](#_Toc393898755)

[6. Diagrama entitate-relație 6](#_Toc2053965930)

[7. Diagrama conceptuală 7](#_Toc161217670)

[8. Schemele relaționale 8](#_Toc403228317)

[9. Normalizările FN1, FN2, FN3 9](#_Toc1359053434)

[a. FN1 9](#_Toc240223043)

[b. FN2 9](#_Toc1119804129)

[c. FN3 10](#_Toc872219493)

[10. Secvențe 11](#_Toc384293599)

[11. Crearea tabelelor în SQL și inserarea datelor 15](#_Toc1214284869)

[12. Cereri SQL complexe 31](#_Toc2018191)

[13. Operații de actualizare și suprimare a datelor 39](#_Toc708726631)

[14. Cereri SQL 44](#_Toc1223460965)

[a. Outer-join pe minim 4 tabele 44](#_Toc1518616939)

[b. Division 45](#_Toc2141108589)

[c. Analiza top-n 46](#_Toc1812227874)

[15. Optimizarea unei cereri 47](#_Toc1237004637)

[16. Normalizările BCNF, FN4, FN5 51](#_Toc1934286839)

[a. BCNF 51](#_Toc1255537835)

[b. FN4 52](#_Toc7870859)

[c. FN5 53](#_Toc1319203742)

# Descrierea modelului real, a utilității acestuia și a regulilor de funcționare

Acest model descrie o bază de date pentru o companie aeriană. Compania aeriană deține avioane și vinde locuri în zboruri. Zborurile pleacă din și ajung în aeroporturi. La zboruri participă piloți și însoțitori de zbor. Clienții rezervă locuri într-un zbor pentru anumiți pasageri. O rezervare poate fi făcută doar pentru un singur pasager. Pasagerii care au o rezervare și se prezintă la zbor trec prin check-in.

# Prezentarea constrângerilor (restricții, reguli)

* Un avion are neapărat un model de avion.
* Un oraș se află neapărat într-o țară.
* Un aeroport se află neapărat într-un oraș.
* Un zbor are neapărat un avion, un aeroport de plecare și unul de sosire.
* Un loc are neapărat un preț și o clasă.
* O rezervare are neapărat un loc, un pasager (pentru cine este făcută rezervarea), un client (cine a făcut rezervarea) și o dată de rezervare.
* O rezervare poate fi făcută pentru un singur loc, un singur pasager, de către un singur client.

# Descrierea entităților

* Model avion: tip de avion; cheie primară: id\_model\_avion
* Avion: avion deținut de companie; cheie primară: id\_avion
* Țară; cheie primară: id\_țară
* Oraș; cheie primară: oraș
* Aeroport: loc de unde pleacă și unde ajung avioane; cheie primară: id\_aeroport
* Zbor: zbor cu un avion de la un aeroport la altul; cheie primară: id\_zbor
* Pilot: angajat al companiei care pilotează avioane; cheie primară: id\_pilot
* Însoțitor de zbor: angajat al companiei care participă la zbor; cheie primară: id\_insoțitor
* Clasă: categorie de locuri în funcție de confort; cheie primară: id\_clasă
* Loc: loc într-un zbor; cheie primară: id\_loc, id\_zbor
* Pasager: persoană care călătorește cu avionul; cheie primară: id\_pasager
* Client: persoană / entitate care rezervă locuri; cheie primară: id\_client
* Persoană fizică: subentitate client; cheie primară: id\_persoană
* Agenție turism: subentitate client; cheie primară: id\_agenție
* Reprezentant vânzări: subentitate client, angajat care vinde bilete; cheie primară: id\_reprezentant
* Rezervare: rezervare pentru un loc dintr-un zbor; cheie primară: id\_rezervare
* Plată: plata pentru o rezervare; cheie primară: id\_plată
* Check-in: check-in-ul pentru o rezervare; cheie primară: id\_check\_in
* Bagaj: bagajul de cală, declarat la check-in; cheie primară: id\_bagaj

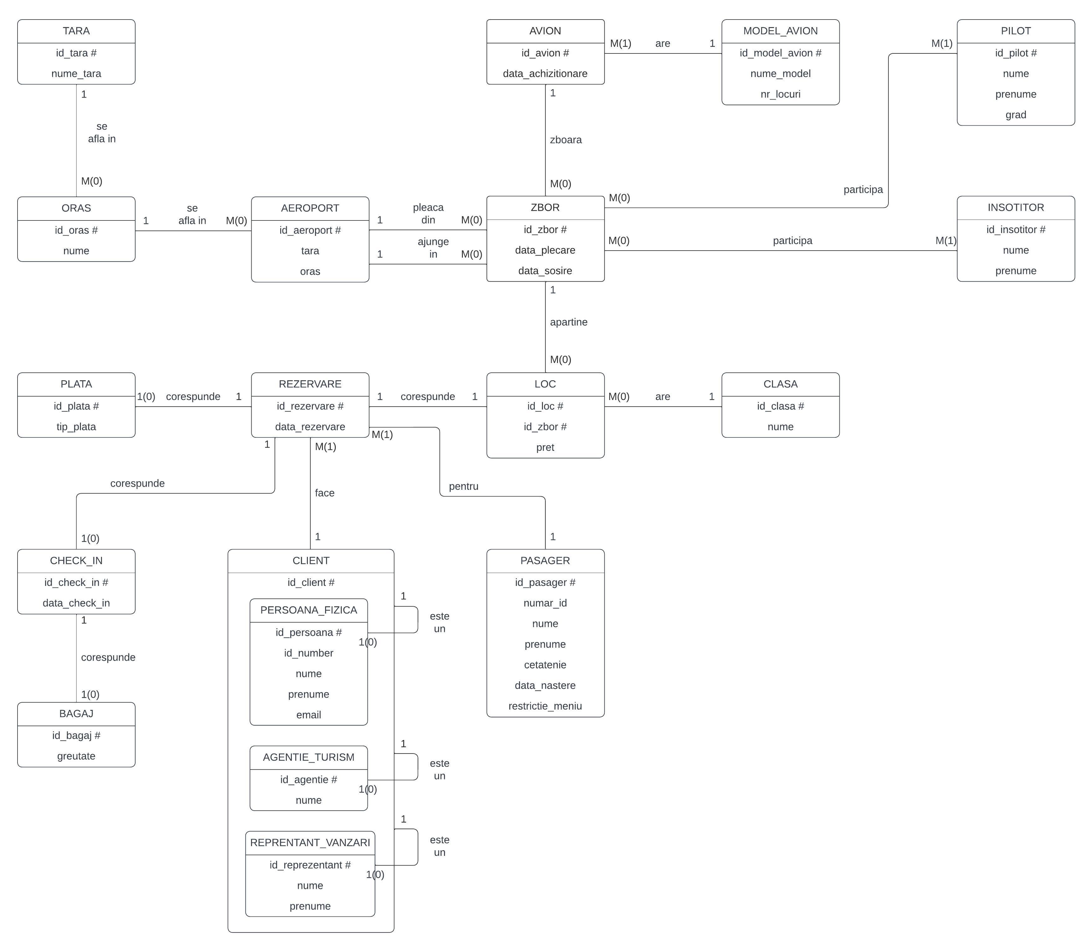
# Descrierea relațiilor

* Unui model de avion îi corespund zero sau mai multe avioane. Un avion are un model de avion și numai unul. 1 – M(0)
* Într-o țară există zero sau mai multe orașe. Un oraș se află într-o țară și numai una. 1 – M(0)
* Într-un oraș există zero sau mai multe aeroporturi. Un aeroport se află într-un oraș și numai unul. 1 – M(0)
* Dintr-un aeroport pleacă zero sau mai multe zboruri. Un zbor pleacă dintr-un aeroport și numai unul.1 - M(0)
* Într-un aeroport ajung zero sau mai multe zboruri. Un zbor ajunge într-un aeroport și numai unul. 1 – M(0)
* Un pilot pilotează zero sau mai multe zboruri. Un zbor este pilotat de unul sau mai mulți piloți. M(1) - M(0)
* Un însoțitor de zbor participă la zero sau mai multe zboruri. La un zbor participă unul sau mai mulți însoțitori. M(1) - M(0)
* Unei clase îi corespund zero sau mai multe locuri. Un loc are o clasă și numai una. 1 – M(0)
* Unui pasager îi corespund zero sau mai multe rezervări. O rezervare este pentru un pasager și numai unul. 1 – M(0)
* Un client face zero sau mai multe rezervări. O rezervare este făcută de un client și numai unul. 1 – M(0)
* Unei rezervări îi corespund zero sau o plată. O plată este pentru o rezervare și numai una. 1 – 1(0)
* Unei rezervări îi corespund zero sau un check-in. Un check-in se realizează pe baza unei rezervări și numai una. 1 – 1(0)
* Un bagaj îi corespunde unui check-in și numai unul. Unui check-in îi corespund zero sau un bagaj. 1(0) - 1

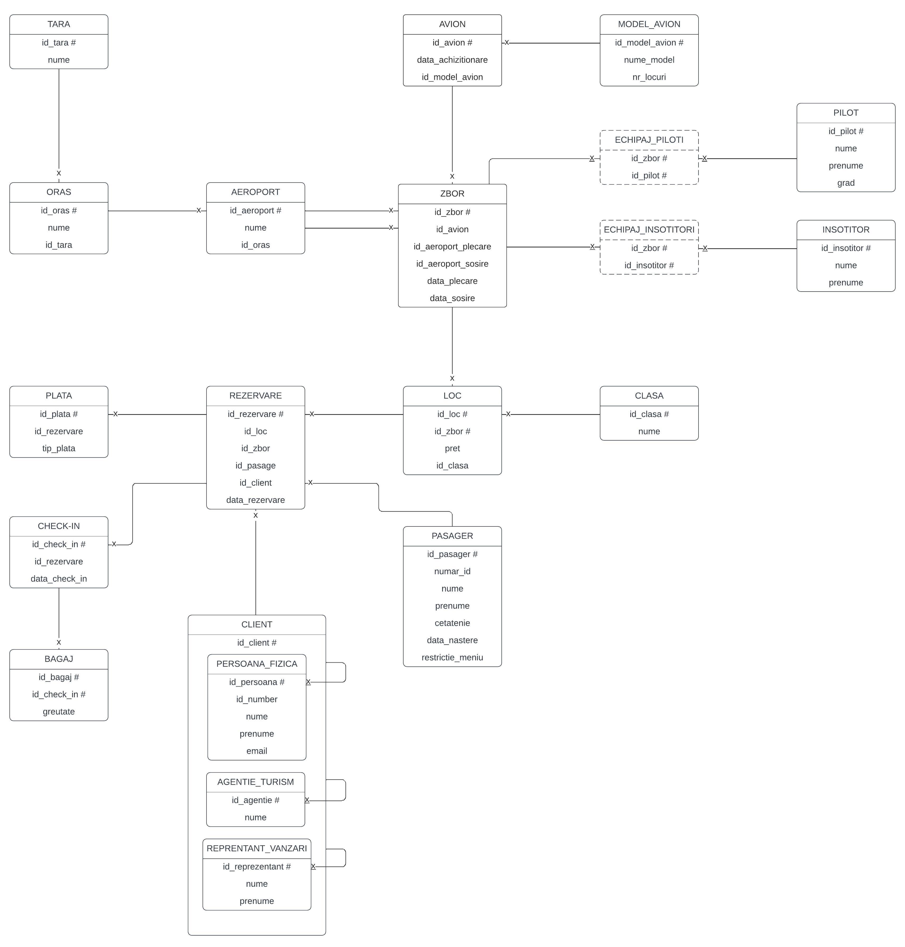
# Descrierea atributelor

* Model avion
  + Id\_model\_avion: cheie primară; number(4)
  + Nume\_model: denumirea modelului; varchar2(30)
  + Nr\_locuri: numărul de locuri din acel model; number(4); not null
* Avion
  + Id\_avion: cheie primară; number(4)
  + Dată\_achiziționare: data la care a fost cumpărat; date; default sysdate
  + Id\_model\_avion: number(4); not null; cheie externă - model\_avion(id\_model\_avion)
* Țară
  + Id\_țară: cheie primară; number(4)
  + Nume: numele țării; varchar2(30)
* Oraș
  + Id\_oraș: cheie primară; number(4)
  + Nume: numele orașului; varchar2(30)
  + Id\_țară: number(4); not null; cheie externă - tara(id\_țară)
* Aeroport
  + Id\_aeroport: cheie primară; number(4)
  + Nume: numele aeroportului; varchar2(30)
  + Id\_oraș: number(4); not null; cheie externă - oraș(id\_oraș)
* Zbor
  + Id\_zbor: cheie primară; number(4)
  + Dată\_plecare: date
  + Dată\_sosire: date
  + Id\_avion: number(4); not null; cheie externă - avion(id\_avion)
  + Id\_aeroport\_plecare: number(4); not null; cheie externă - aeroport(id\_aeroport)
  + Id\_aeroport\_sosire: number(4); not null; cheie externă - aeroport(id\_aeroport)
* Pilot
  + Id\_pilot: cheie primară; number(4)
  + Nume: varchar2(30); not null
  + Prenume: varchar2(30)
  + Grad: pilot sau copilot; varchar2(30)
* Însoțitor de zbor
  + Id\_însoțitor: cheie primară; number(4)
  + Nume: varchar2(30); not null
  + Prenume: varchar2(30)
* Echipaj piloți
  + Id\_zbor: number(4); cheie externă - zbor(id\_zbor)
  + Id\_pilot: number(4); cheie externă - pilot(id\_pilot)
  + Cheie primară compusă: id\_zbor + id\_pilot
* Echipaj însoțitori
  + Id\_zbor: number(4); cheie externă - zbor(id\_zbor)
  + Id\_însoțitor: number(4); cheie externă - insoțitori\_zbor(id\_insoțitor)
  + Cheie primară compusă: id\_zbor + id\_însoțitor
* Clasă
  + Id\_clasa: cheie primară; number(4)
  + Nume: varchar2(30); economy, business, întâi
* Loc
  + Id\_loc: number(4); cheie externă - loc(id\_loc)
  + Id\_zbor: number(4); cheie externă - zbor(id\_zbor)
  + Cheie primară compusă: id\_loc + id\_zbor
  + Id\_clasă: number(4); cheie externă - clasă(id\_clasă)
* Pasager
  + Id\_pasager: cheie primară; number(4)
  + Număr\_id: number(4); not null; cnp, security number
  + Nume: varchar2(30); not null
  + Prenume: varchar2(30)
  + Cetățenie: varchar2(30) not null
  + Dată\_naștere: date
  + Restricție\_meniu: varchar2(30); diabet, vegetarian
* Client
  + Id\_client: number(4); cheie primară
* Persoană fizică
  + Id\_persoană: cheie primară; number(4); cheie externă - client(id\_client)
  + Id\_number: varchar2(30); not null; cnp, security number
  + Nume: varchar2(30) not null
  + Prenume: varchar2(30)
  + Email: varchar2(30)
* Agenție turism
  + Id\_agenție: cheie primară; number(4); cheie externă - client(id\_client)
  + Nume: varchar2(30)
* Reprezentant vânzări
  + Id\_reprezentant: cheie primară; number(4); cheie externă - client(id\_client)
  + Nume: varchar2(30) not null
  + Prenume: varchar2(30)
* Rezervare
  + Id\_rezervare: cheie primară; number(4)
  + Id\_loc: number(4); not null
  + Id\_zbor: number(4); not null
  + Cheie externă: id\_loc + id\_zbor – loc(id\_loc, id\_zbor)
  + Id\_pasager: number(4); not null; cheie externă - pasager(id\_pasager)
  + Id\_client: number(4); not null; cheie externă - client(id\_client)
  + Dată\_rezervare: date; not null; default sysdate
* Plată
  + Id\_plată: cheie primară; number(4)
  + Id\_rezervare: number(4); not null; unique; cheie externă - rezervare(id\_rezervare)
  + Tip\_plată: varchar2(30); online, transfer bancar
* Check-in
  + Id\_check\_in: cheie primară; number(4)
  + Id\_rezervare: number(4); not null; unique; cheie externă - rezervare(id\_rezervare)
  + Dată\_check\_in: date; not null
* Bagaj
  + Id\_bagaj: cheie primară; number(4)
  + Id\_check\_in: number(4); not null; unique; cheie externă - check\_in(id\_check\_in)

# Diagrama entitate-relație



# Diagrama conceptuală



# Schemele relaționale

* Modele\_avioane (id\_model\_avion#, nume\_model, nr\_locuri)
* Avioane (id\_avion#, dată\_achiziționare, id\_model\_avion)
* Țări (id\_țară#, nume)
* Orașe (id\_oraș#, nume, id\_tara)
* Aeroporturi (id\_aeroport#, nume, id\_oraș)
* Zboruri (id\_zbor#, dată\_plecare, dată\_sosire, id\_avion, id\_aeroport\_plecare, id\_aeroport\_sosire)
* Piloți (id\_pilot#, nume, prenume grad)
* Insotitori\_zbor (id\_insoțitor#, nume, prenume)
* Echipaj\_piloți (id\_zbor#, id\_pilot#)
* Echipaj\_însoțitori (id\_zbor#, id\_insoțitor#)
* Clase (id\_clasa#, nume)
* Locuri (id\_loc#, id\_zbor#, preț, id\_clasă)
* Pasageri (id\_pasager#, număr\_id, nume, prenume, cetățenie, dată\_naștere, restricție\_meniu)
* Clienți (id\_client#)
* Persoane\_fizice (id\_persoană#, id\_number, nume, prenume, email)
* Agenții\_turism (id\_agenție#, nume)
* Reprezentanți\_vânzări (id\_reprezentant#, nume, prenume)
* Rezervări (id\_rezervare#, id\_loc, id\_zbor, id\_pasager, id\_client, dată\_rezervare)
* Plați (id\_plată#, id\_rezervare, tip\_plată)
* Check\_in (id\_check\_in#, id\_rezervare, dată\_check\_in)
* Bagaje (id\_bagaj#, id\_check\_in, greutate)

# Normalizările FN1, FN2, FN3

## FN1

* Diagrama se află în FN1 deoarece pentru fiecare relație din diagramă, fiecărui atribut care o compune îi corespunde o singură valoare indivizibilă.
* Un exemplu de non-FN1 ar fi (în tabelul zboruri):

|  |  |
| --- | --- |
| id\_zbor # | id\_aeroporturi |
| 0 | 0, 1 |
| 1 | 2, 4 |
| 2 | 0, 2 |

* În celulele coloanei id\_aeroporturi sunt mai multe valori.
* Rezolvare:

|  |  |  |
| --- | --- | --- |
| id\_zbor # | id\_aeroport\_plecare | id\_aeroport\_sosire |
| 0 | 0 | 1 |
| 1 | 2 | 4 |
| 2 | 0 | 2 |

* Am înlocuit coloana id\_aeroporturi cu două coloane, id\_aeroport\_plecare și id\_aeroport\_sosire.

## FN2

* Diagrama se află în FN2 deoarece se află în FN1 și fiecare atribut care nu este cheie primară depinde de întreaga cheie primară.
* Un exemplu de non-FN2 ar fi (în tabelul avioane):

|  |  |  |
| --- | --- | --- |
| id\_avion # | nume\_model # | nr\_locuri |
| 0 | Airbus A320 | 10 |
| 1 | Airbus A340 | 15 |
| 2 | Boeing 757 | 15 |

* Numărul de locuri depinde doar de nume\_model, nu și de id\_avion.
* Rezolvare:

|  |  |
| --- | --- |
| id\_avion # | id\_model\_avion |
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |

|  |  |
| --- | --- |
| id\_avion # | id\_model\_avion |
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |

* Acum toate atributele care nu sunt cheie primară depind de întreaga cheie primară.

## FN3

* Diagrama se află în FN3 deoarece se află în FN1 și în FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară, adică depinde de întreaga cheie primară și numai de ea.
* Un exemplu de non-FN3 ar fi:

|  |  |  |
| --- | --- | --- |
| id\_locatie # | oras | tara |
| 0 | Bucuresti | Romania |
| 1 | Cluj-Napoca | Romania |
| 2 | Munich | Germania |

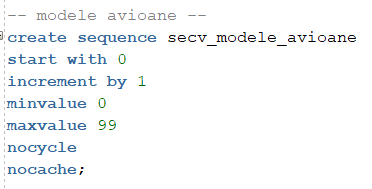
* Atributul tara depinde tranzitiv de id\_locatie prin atributul oras.
* Rezolvare:

|  |  |  |
| --- | --- | --- |
| id\_oras # | nume | id\_tara |
| 0 | Bucuresti | 0 |
| 1 | Cluj-Napoca | 0 |
| 2 | Munich | 1 |

|  |  |
| --- | --- |
| id\_tara # | nume |
| 0 | Romania |
| 1 | Germania |

* Acum fiecare atribut care nu e cheie depinde direct de cheia primară.

# Secvențe



create sequence secv\_modele\_avioane

start with 0

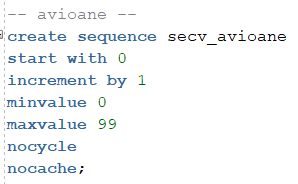
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_avioane

start with 0

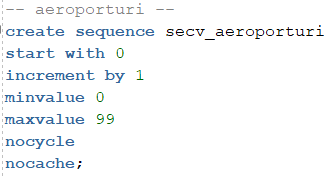
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_aeroporturi

start with 0

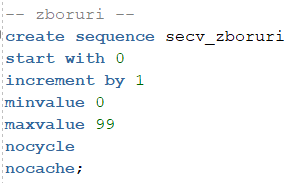
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_zboruri

start with 0

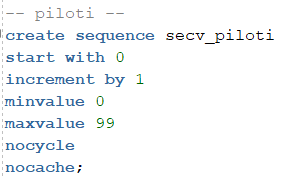
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_piloti

start with 0

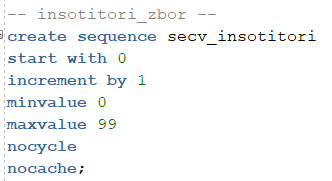
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_insotitori

start with 0

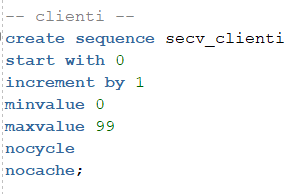
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_clienti

start with 0

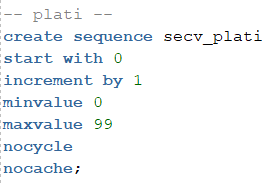
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_plati

start with 0

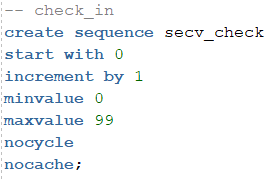
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_check

start with 0

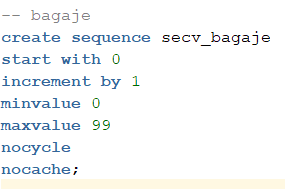
increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;



create sequence secv\_bagaje

start with 0

increment by 1

minvalue 0

maxvalue 99

nocycle

nocache;

# Crearea tabelelor în SQL și inserarea datelor

* Modele avioane

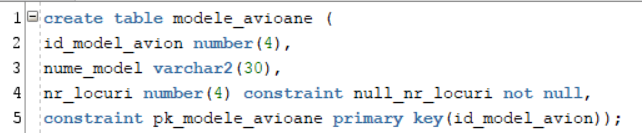
create table modele\_avioane (

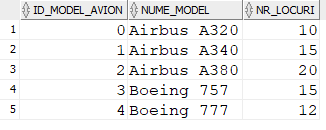
id\_model\_avion number(4),

nume\_model varchar2(30),

nr\_locuri number(4) constraint null\_nr\_locuri not null,

constraint pk\_modele\_avioane primary key(id\_model\_avion));





* Avioane

create table avioane (

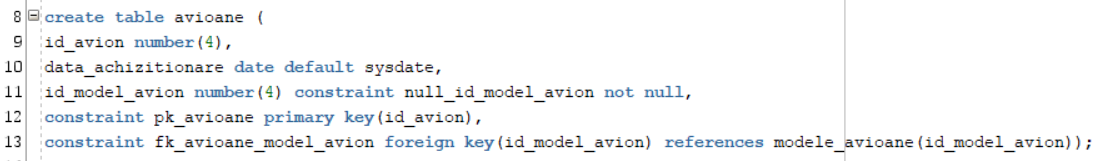
id\_avion number(4),

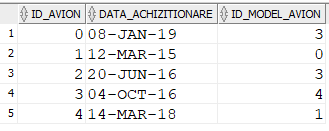
data\_achizitionare date default sysdate,

id\_model\_avion number(4) constraint null\_id\_model\_avion not null,

constraint pk\_avioane primary key(id\_avion),

constraint fk\_avioane\_model\_avion foreign key(id\_model\_avion) references modele\_avioane(id\_model\_avion));





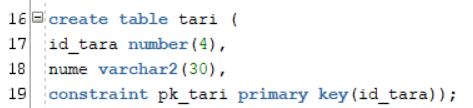
* Țări

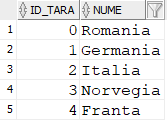
create table tari (

id\_tara number(4),

nume varchar2(30),

constraint pk\_tari primary key(id\_tara));





* Orașe

create table orase (

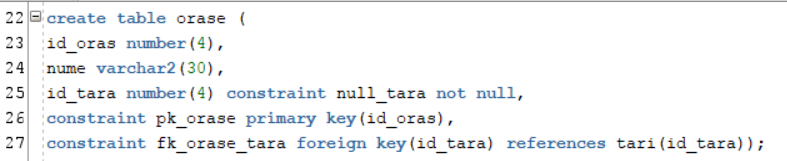
id\_oras number(4),

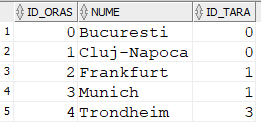
nume varchar2(30),

id\_tara number(4) constraint null\_tara not null,

constraint pk\_orase primary key(id\_oras),

constraint fk\_orase\_tara foreign key(id\_tara) references tari(id\_tara));





* Aeroporturi

create table aeroporturi (

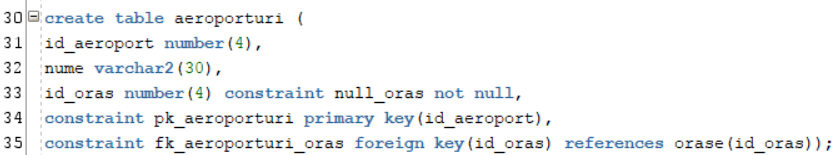
id\_aeroport number(4),

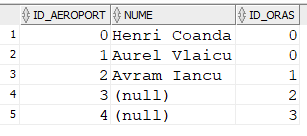
nume varchar2(30),

id\_oras number(4) constraint null\_oras not null,

constraint pk\_aeroporturi primary key(id\_aeroport),

constraint fk\_aeroporturi\_oras foreign key(id\_oras) references orase(id\_oras));





* Zboruri

create table zboruri (

id\_zbor number(4),

data\_plecare date,

data\_sosire date,

id\_avion number(4) constraint null\_avion not null,

id\_aeroport\_plecare number(4) constraint null\_plecare not null,

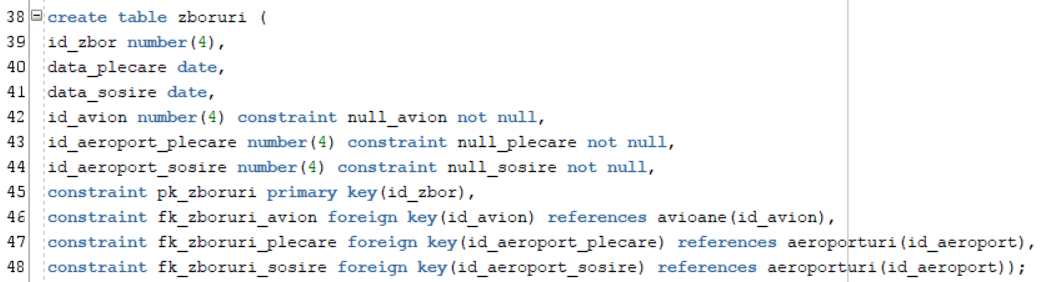
id\_aeroport\_sosire number(4) constraint null\_sosire not null,

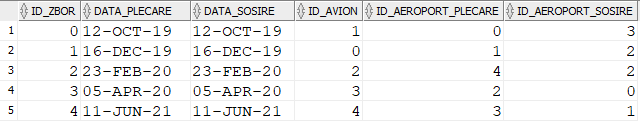
constraint pk\_zboruri primary key(id\_zbor),

constraint fk\_zboruri\_avion foreign key(id\_avion) references avioane(id\_avion),

constraint fk\_zboruri\_plecare foreign key(id\_aeroport\_plecare) references aeroporturi(id\_aeroport),

constraint fk\_zboruri\_sosire foreign key(id\_aeroport\_sosire) references aeroporturi(id\_aeroport));





* Piloți

create table piloti (

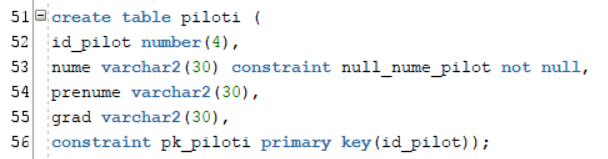
id\_pilot number(4),

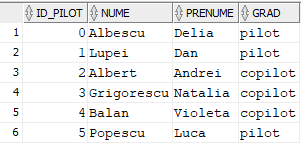
nume varchar2(30) constraint null\_nume\_pilot not null,

prenume varchar2(30),

grad varchar2(30),

constraint pk\_piloti primary key(id\_pilot));





* Insoțitori zbor

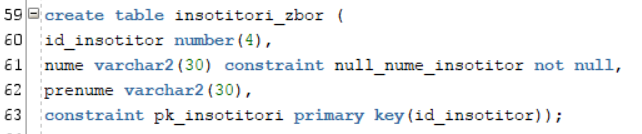
create table insotitori\_zbor (

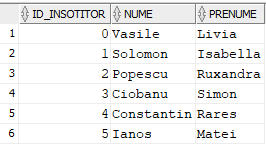
id\_insotitor number(4),

nume varchar2(30) constraint null\_nume\_insotitor not null,

prenume varchar2(30),

constraint pk\_insotitori primary key(id\_insotitor));





* Echipaj piloți

create table echipaj\_piloti (

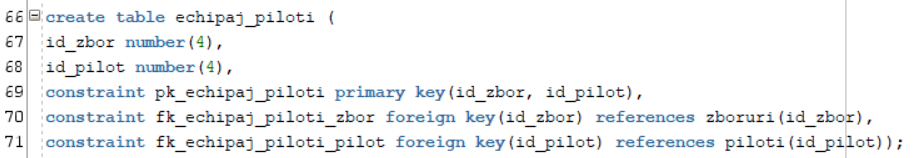
id\_zbor number(4),

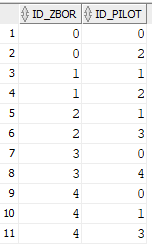
id\_pilot number(4),

constraint pk\_echipaj\_piloti primary key(id\_zbor, id\_pilot),

constraint fk\_echipaj\_piloti\_zbor foreign key(id\_zbor) references zboruri(id\_zbor),

constraint fk\_echipaj\_piloti\_pilot foreign key(id\_pilot) references piloti(id\_pilot));





* Echipaj însoțitori

create table echipaj\_insotitori (

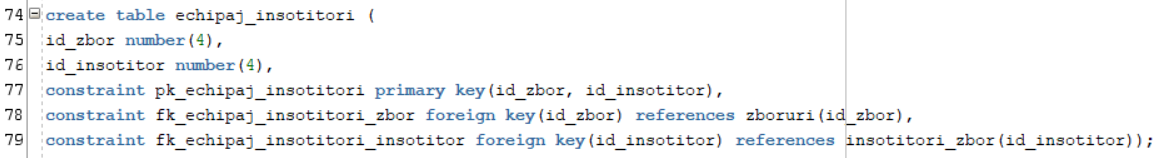
id\_zbor number(4),

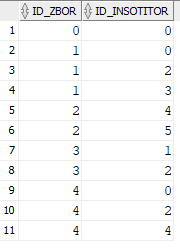
id\_insotitor number(4),

constraint pk\_echipaj\_insotitori primary key(id\_zbor, id\_insotitor),

constraint fk\_echipaj\_insotitori\_zbor foreign key(id\_zbor) references zboruri(id\_zbor),

constraint fk\_echipaj\_insotitori\_insotitor foreign key(id\_insotitor) references insotitori\_zbor(id\_insotitor));





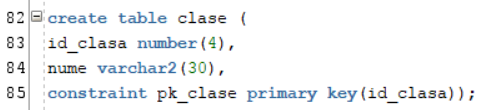
* Clase

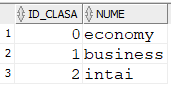
create table clase (

id\_clasa number(4),

nume varchar2(30),

constraint pk\_clase primary key(id\_clasa));





* Locuri

create table locuri (

id\_loc number(4),

id\_zbor number(4),

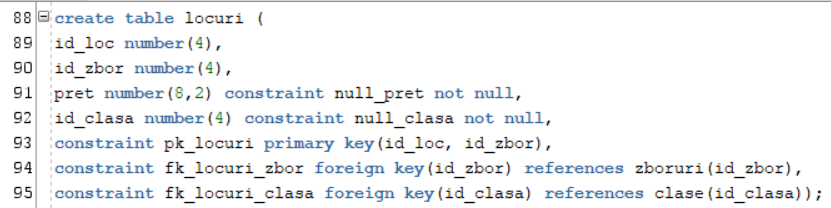
pret number(8,2) constraint null\_pret not null,

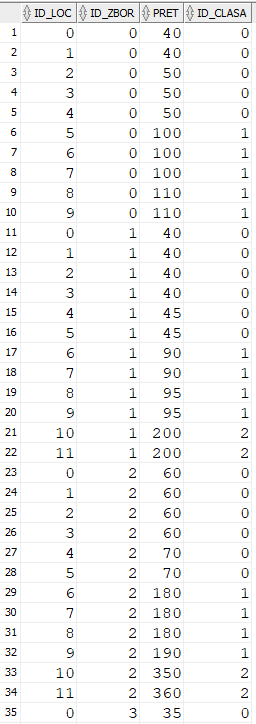
id\_clasa number(4) constraint null\_clasa not null,

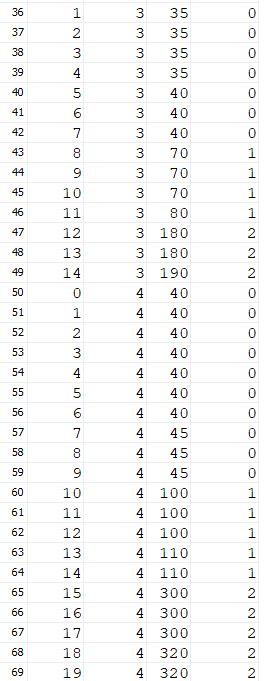
constraint pk\_locuri primary key(id\_loc, id\_zbor),

constraint fk\_locuri\_zbor foreign key(id\_zbor) references zboruri(id\_zbor),

constraint fk\_locuri\_clasa foreign key(id\_clasa) references clase(id\_clasa));







* Pasageri

create table pasageri (

id\_pasager number(4),

numar\_id varchar2(20) constraint null\_cnp\_pasageri not null,

nume varchar2(30) constraint null\_nume\_pasageri not null,

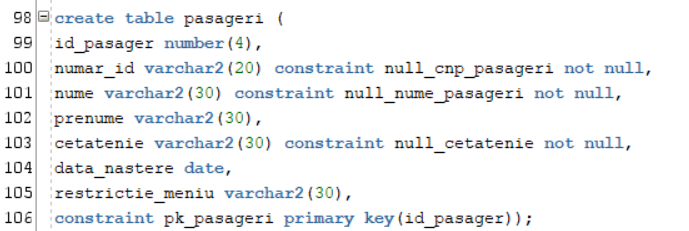
prenume varchar2(30),

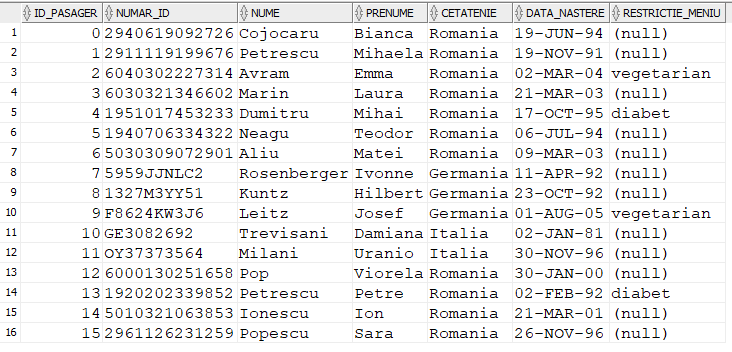
cetatenie varchar2(30) constraint null\_cetatenie not null,

data\_nastere date,

restrictie\_meniu varchar2(30),

constraint pk\_pasageri primary key(id\_pasager));



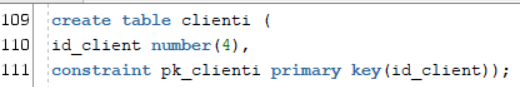


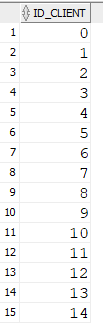
* Clienți

create table clienti (

id\_client number(4),

constraint pk\_clienti primary key(id\_client));





* Persoane fizice

create table persoane\_fizice (

id\_persoana number(4),

id\_number number(20) constraint null\_cnp\_pers not null,

nume varchar2(30) constraint null\_nume\_pers not null,

prenume varchar2(30),

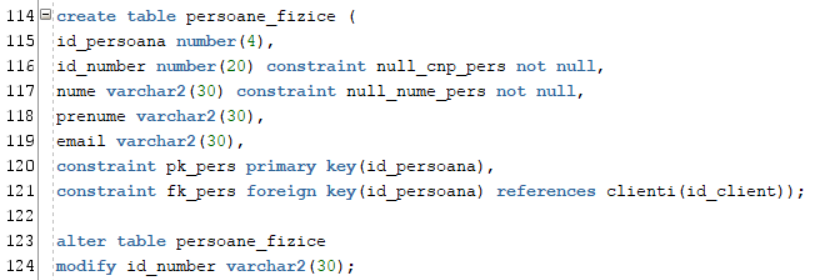
email varchar2(30),

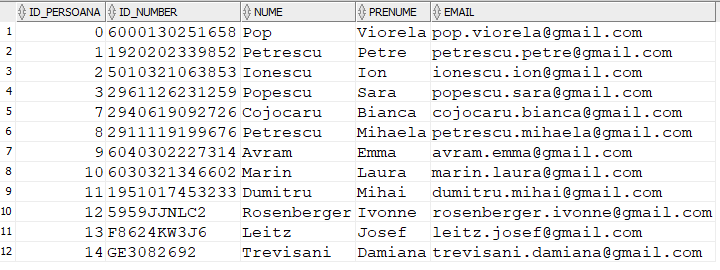
constraint pk\_pers primary key(id\_persoana),

constraint fk\_pers foreign key(id\_persoana) references clienti(id\_client));

alter table persoane\_fizice

modify id\_number varchar2(30);





* Agenții turism

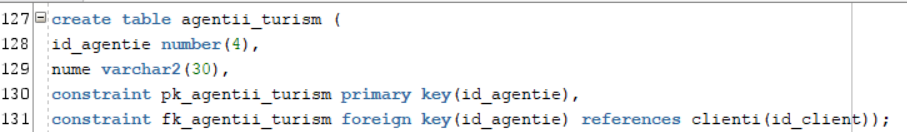
create table agentii\_turism (

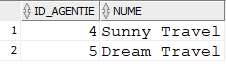
id\_agentie number(4),

nume varchar2(30),

constraint pk\_agentii\_turism primary key(id\_agentie),

constraint fk\_agentii\_turism foreign key(id\_agentie) references clienti(id\_client));





* Reprezentanți vânzări

create table reprezentanti\_vanzari (

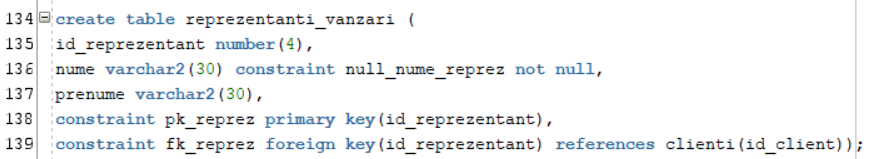
id\_reprezentant number(4),

nume varchar2(30) constraint null\_nume\_reprez not null,

prenume varchar2(30),

constraint pk\_reprez primary key(id\_reprezentant),

constraint fk\_reprez foreign key(id\_reprezentant) references clienti(id\_client));





* Rezervări

create table rezervari (

id\_rezervare number(4),

id\_loc number(4) constraint null\_loc not null,

id\_zbor number(4) constraint null\_zbor not null,

id\_pasager number(4) constraint null\_pasager not null,

id\_client number(4) constraint null\_client not null,

data\_rezervare date default sysdate constraint null\_data not null,

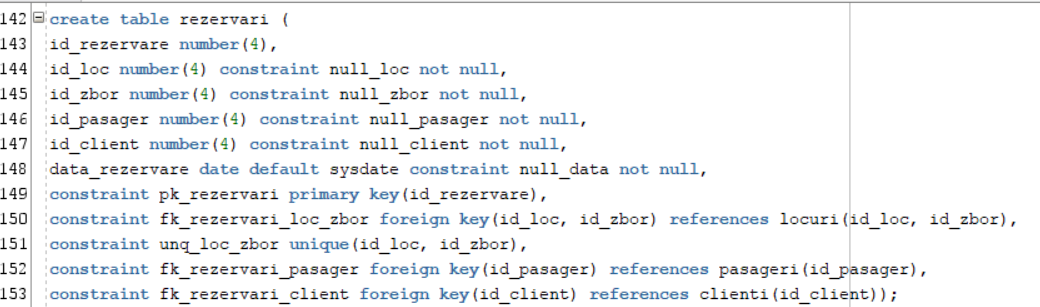
constraint pk\_rezervari primary key(id\_rezervare),

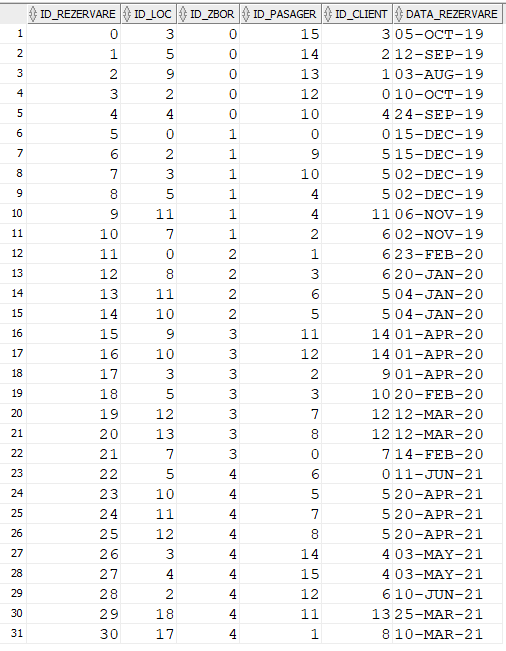
constraint fk\_rezervari\_loc\_zbor foreign key(id\_loc, id\_zbor) references locuri(id\_loc, id\_zbor),

constraint unq\_loc\_zbor unique(id\_loc, id\_zbor),

constraint fk\_rezervari\_pasager foreign key(id\_pasager) references pasageri(id\_pasager),

constraint fk\_rezervari\_client foreign key(id\_client) references clienti(id\_client));





* Plăți

create table plati (

id\_plata number(4),

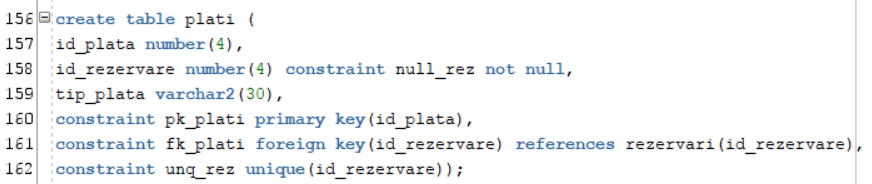
id\_rezervare number(4) constraint null\_rez not null,

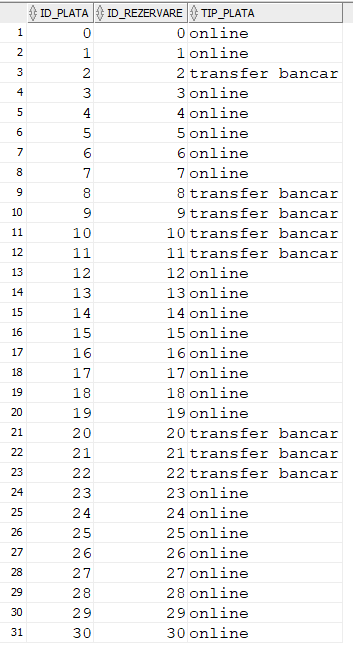
tip\_plata varchar2(30),

constraint pk\_plati primary key(id\_plata),

constraint fk\_plati foreign key(id\_rezervare) references rezervari(id\_rezervare),

constraint unq\_rez unique(id\_rezervare));





* Check-in

create table check\_in (

id\_check\_in number(4),

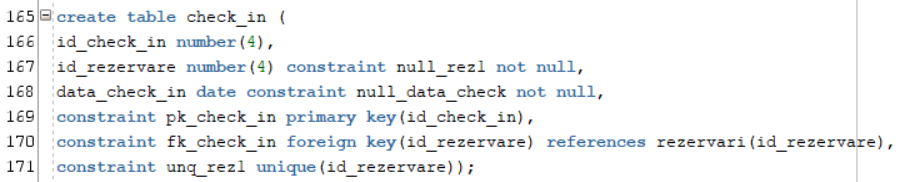
id\_rezervare number(4) constraint null\_rez1 not null,

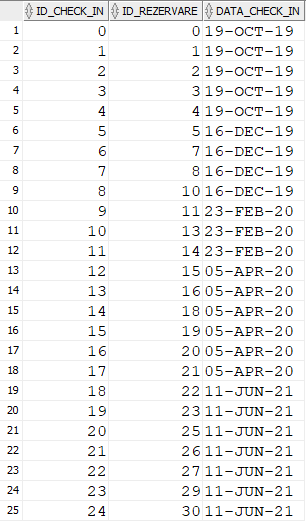
data\_check\_in date constraint null\_data\_check not null,

constraint pk\_check\_in primary key(id\_check\_in),

constraint fk\_check\_in foreign key(id\_rezervare) references rezervari(id\_rezervare),

constraint unq\_rez1 unique(id\_rezervare));





* Bagaje

create table bagaje (

id\_bagaj number(4),

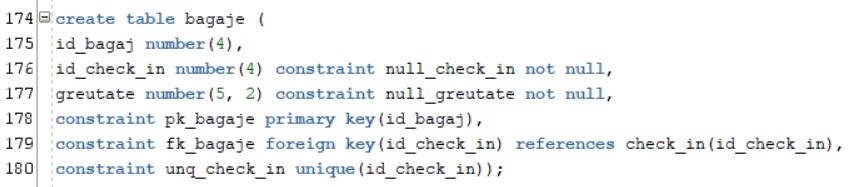
id\_check\_in number(4) constraint null\_check\_in not null,

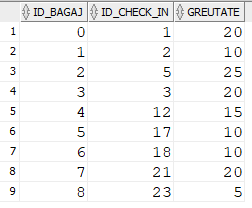
greutate number(5, 2) constraint null\_greutate not null,

constraint pk\_bagaje primary key(id\_bagaj),

constraint fk\_bagaje foreign key(id\_check\_in) references check\_in(id\_check\_in),

constraint unq\_check\_in unique(id\_check\_in));





# Cereri SQL complexe

* 1) Să se obțină însoțitorii de zbor al căror nume conține litera „a” care au participat la zborurile din anii 2020 și 2021 care au plecat din orașul Frankfurt și care au avut gradul de ocupare (câți pasageri au trecut de check-in față de câte locuri sunt în avionul respectiv) mai mic de 40%.
* Cod text:

select distinct z.id\_zbor, z.data\_plecare, i.id\_insotitor, i.nume, a.id\_aeroport, o.nume oras

from insotitori\_zbor i, echipaj\_insotitori ei, zboruri z, aeroporturi a, orase o

where ei.id\_insotitor = i.id\_insotitor

and ei.id\_zbor = z.id\_zbor

and z.id\_aeroport\_plecare = a.id\_aeroport

and a.id\_oras = o.id\_oras

and extract(year from z.data\_plecare) in (2020, 2021)

and i.nume like '%a%'

and lower(o.nume) = 'frankfurt'

and z.id\_zbor in (with locuri\_zbor as (select id\_zbor, count(id\_loc) nr\_loc

from locuri

group by id\_zbor

order by id\_zbor),

check\_zbor as (select r.id\_zbor zbor, count(c.id\_check\_in) nr\_check

from rezervari r, check\_in c

where r.id\_rezervare = c.id\_rezervare

group by r.id\_zbor

order by r.id\_zbor)

select a.id\_zbor

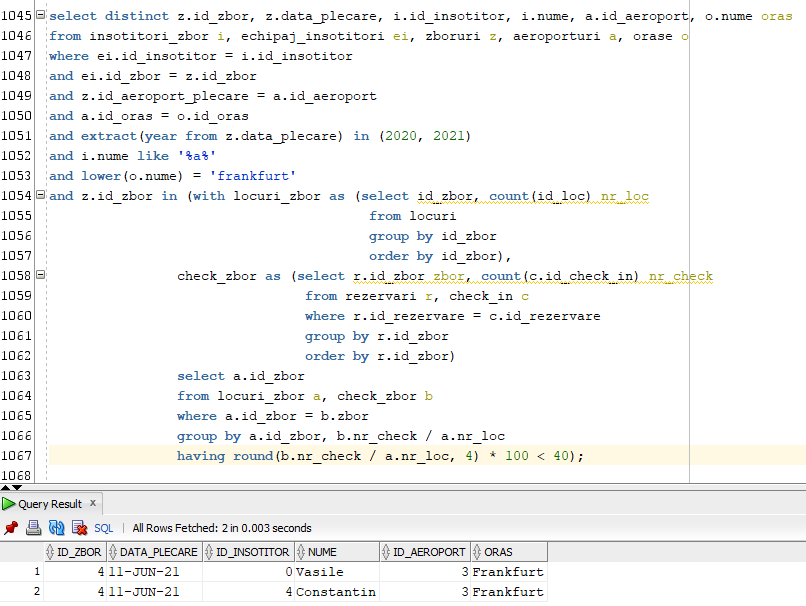
from locuri\_zbor a, check\_zbor b

where a.id\_zbor = b.zbor

group by a.id\_zbor, b.nr\_check / a.nr\_loc

having round(b.nr\_check / a.nr\_loc, 4) \* 100 < 40);

* Print-screen + rezultat:



* Elemente: blocuri de cerere(clauza with), funcție pe șiruri de caractere, funcție pe date calendaristice
* 2) Să se obțină suma greutății bagajelor din fiecare zbor și să se determine dacă este egală cu 25 sau 35. Zborurile să fie ordonate crescător după suma greutății bagajelor.
* Cod text:

select z.id\_zbor, z.suma, decode(z.suma, 25, 'suma egala cu 25', 35, 'suma egala cu 35', 'suma diferita de 25 si 35') rezultat

from (select nvl(sum(b.greutate), 0) suma, r.id\_zbor

from bagaje b, check\_in c, rezervari r

where b.id\_check\_in(+) = c.id\_check\_in

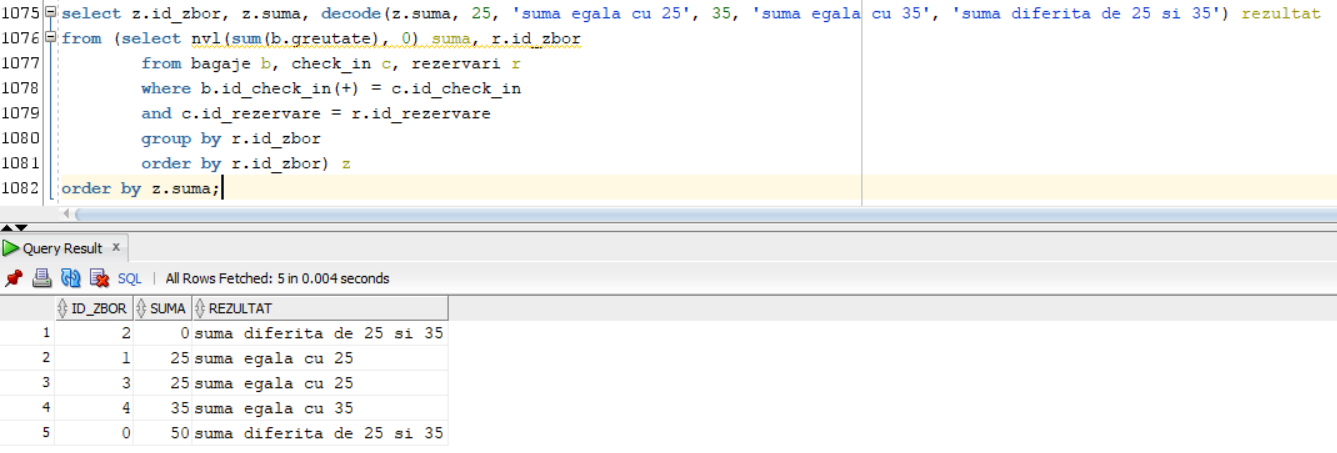
and c.id\_rezervare = r.id\_rezervare

group by r.id\_zbor

order by r.id\_zbor) z

order by z.suma;

* Print-screen + rezultat:



* Elemente: subcerere nesincronizată în clauza from, ordonări și utilizarea funcțiilor nvl și decode
* 3) Să se obțină avioanele și modelul acestora (id + nume) care au participat la primele 2 zboruri care au încasat cei mai puțini bani din vânzarea cu plată de tip online a locurilor la clasa business, dar care au încasat mai mult de 100 din vânzarea acestor bilete (dacă sunt mai multe zboruri cu aceeași sumă de bani minimă, se afișează toate).
* Cod text:

select distinct a.id\_avion, z.id\_zbor, m.id\_model\_avion, m.nume\_model

from avioane a, zboruri z, modele\_avioane m

where a.id\_avion = z.id\_avion

and m.id\_model\_avion = a.id\_model\_avion

and (select sum(l.pret) bani

from locuri l, rezervari r, plati p, clase c

where z.id\_zbor = l.id\_zbor

and l.id\_loc = r.id\_loc

and l.id\_zbor = r.id\_zbor

and r.id\_rezervare = p.id\_rezervare

and c.id\_clasa = l.id\_clasa

and lower(p.tip\_plata) = 'online'

and initcap(c.nume) = 'Business') in (select \*

from (select sum(l.pret) bani

from locuri l, rezervari r, plati p, clase c

where l.id\_zbor = r.id\_zbor

and l.id\_loc = r.id\_loc

and r.id\_rezervare = p.id\_rezervare

and c.id\_clasa = l.id\_clasa

and p.tip\_plata = 'online'

and initcap(c.nume) = 'Business'

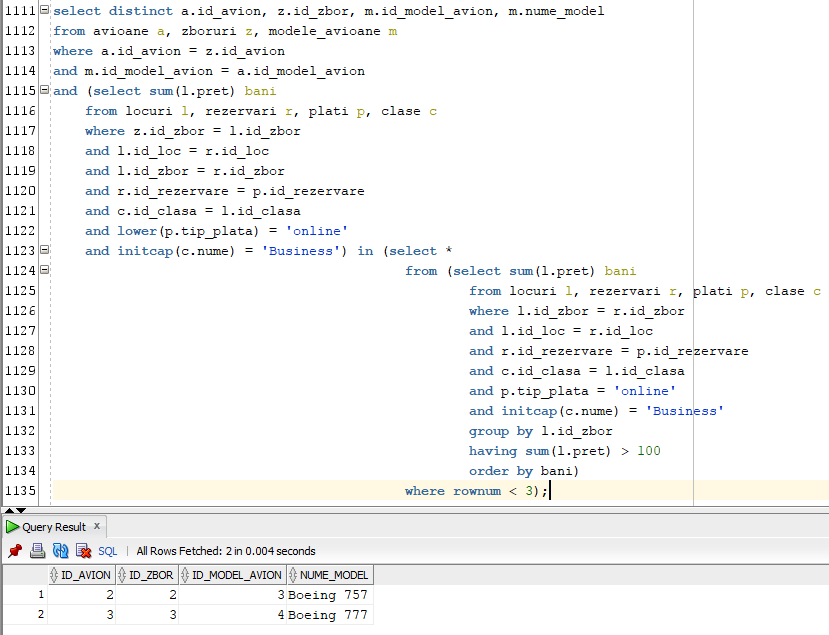
group by l.id\_zbor

having sum(l.pret) > 100

order by bani)

where rownum < 3);

* Print-screen + rezultate:



* Elemente: subcerere sincronizată în care intervin cel puțin 3 tabele, grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri, funcții pe șiruri de caractere
* 4) Se cere ca pentru fiecare rezervare care a fost făcută cu cel puțin o lună și jumătate înainte de data de plecare a zborului să se afle dacă rezervarea a fost făcută de către pasager (acest lucru se poate afla comparând numar\_id, adică atributul de cnp/security number), de către o altă persoană cu aceeași ultimă literă a numelui ca și a pasagerului, de către o altă persoană cu o altă ultimă literă a numelui, de către o agenție de turism sau de către un reprezentant de vânzări.
* Cod text:

select r.id\_rezervare, r.id\_loc, r.id\_zbor, r.id\_pasager, r.id\_client, round(months\_between(z.data\_plecare, r.data\_rezervare), 2) nr\_luni,

case when (exists(select pf.id\_persoana

from persoane\_fizice pf, pasageri p

where pf.id\_persoana = r.id\_client

and r.id\_pasager = p.id\_pasager

and pf.id\_number = p.numar\_id)) then 'rezervare facuta de pasager'

when (exists(select pf.id\_persoana

from persoane\_fizice pf, pasageri p

where pf.id\_persoana = r.id\_client

and r.id\_pasager = p.id\_pasager

and pf.id\_number != p.numar\_id

and substr(p.nume, length(p.nume), 1) = substr(pf.nume, length(pf.nume), 1))) then 'rezervare facuta de alta persoana cu aceeasi ultima litera a numelui'

when (exists(select pf.id\_persoana

from persoane\_fizice pf, pasageri p

where pf.id\_persoana = r.id\_client

and r.id\_pasager = p.id\_pasager

and pf.id\_number != p.numar\_id

and substr(p.nume, -1, 1) != substr(pf.nume, -1, 1))) then 'rezervare facuta de alta persoana'

when (exists(select a.id\_agentie

from agentii\_turism a, pasageri p

where a.id\_agentie = r.id\_client

and r.id\_pasager = p.id\_pasager)) then 'rezervare facuta de o agentie de turism'

when (exists(select rv.id\_reprezentant

from reprezentanti\_vanzari rv, pasageri p

where rv.id\_reprezentant = r.id\_client

and r.id\_pasager = p.id\_pasager)) then 'rezervare facuta de un reprezentant de vanzari'

else 'nu'

end as rezervare

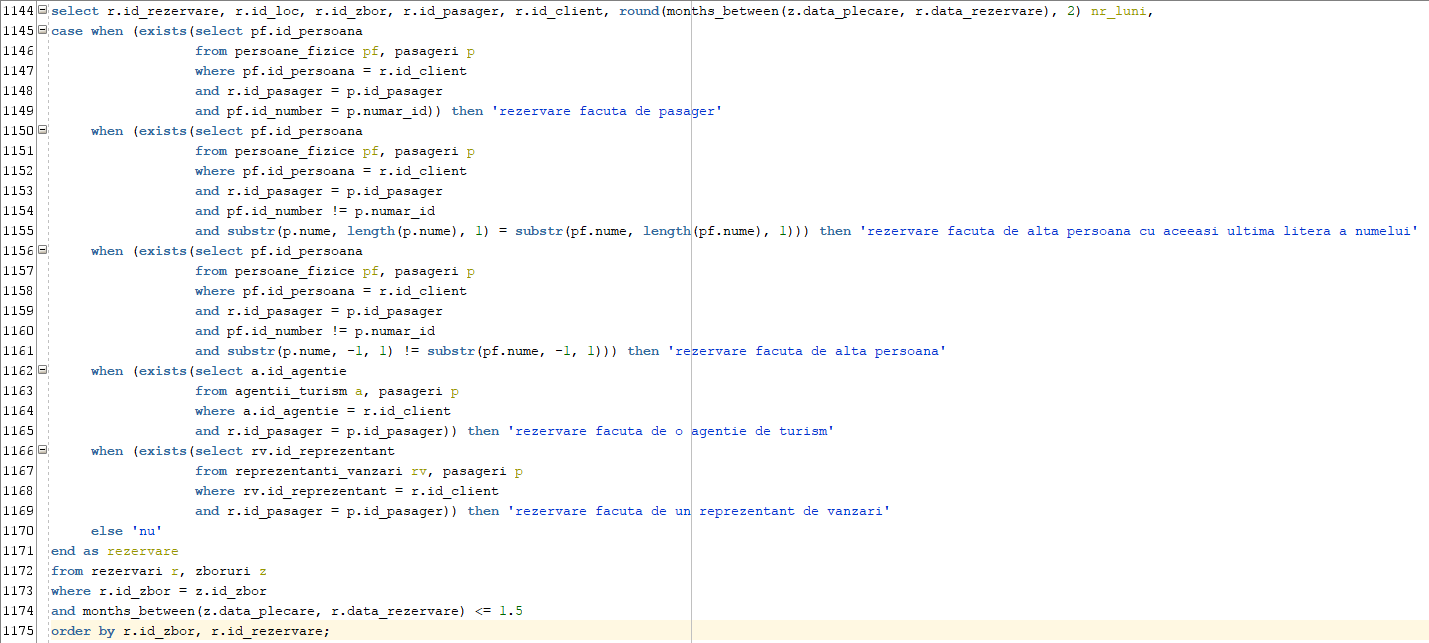
from rezervari r, zboruri z

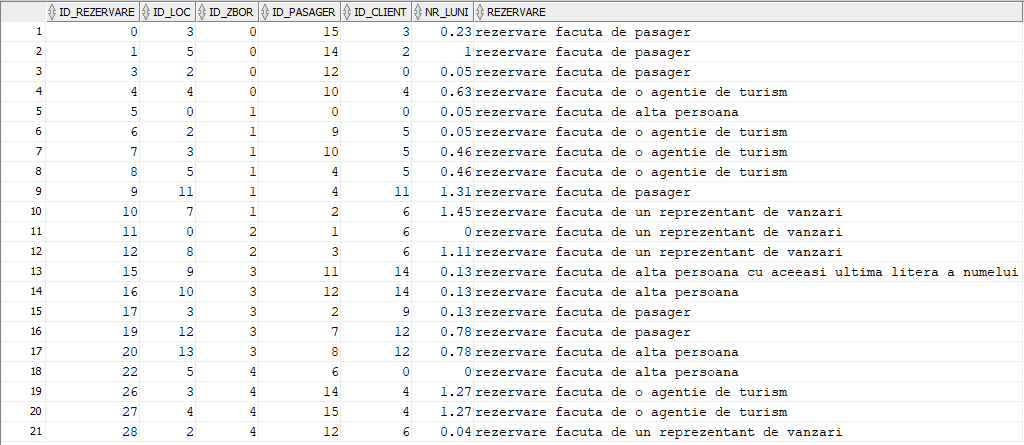
where r.id\_zbor = z.id\_zbor

and months\_between(z.data\_plecare, r.data\_rezervare) <= 1.5

order by r.id\_zbor, r.id\_rezervare;

* Print-screen + rezultate:





* Elemente: expresie case, funcții pe șiruri de caractere, funcție pe date calendaristice
* 5) Să se obțină zborurile care au numărul de rezervări pentru fiecare clasă (cele care au fost disponibile în acel zbor) mai mare decât media numărului de rezervări pentru clasa respectivă.
* Cod text:

with b as (select z.id\_zbor z1, count(a.zbor) nr\_aparitii

from zboruri z, (select r.id\_zbor zbor, l.id\_clasa, c.nume, count(r.id\_rezervare) nr

from rezervari r, locuri l, clase c

where r.id\_zbor = l.id\_zbor

and r.id\_loc = l.id\_loc

and l.id\_clasa = c.id\_clasa(+)

group by r.id\_zbor, l.id\_clasa, c.nume

having count(r.id\_rezervare) >= (select count(r1.id\_rezervare) / (select count(\*) from zboruri) medie

from rezervari r1, locuri l1

where r1.id\_loc = l1.id\_loc

and r1.id\_zbor = l1.id\_zbor

and l.id\_clasa = l1.id\_clasa

group by l1.id\_clasa)

order by r.id\_zbor) a

where z.id\_zbor = a.zbor

group by z.id\_zbor

order by z.id\_zbor),

c as (select l.id\_zbor z2, count(distinct(l.id\_clasa)) nr\_clase

from locuri l

group by l.id\_zbor

order by l.id\_zbor)

select z.id\_zbor

from zboruri z, b, c

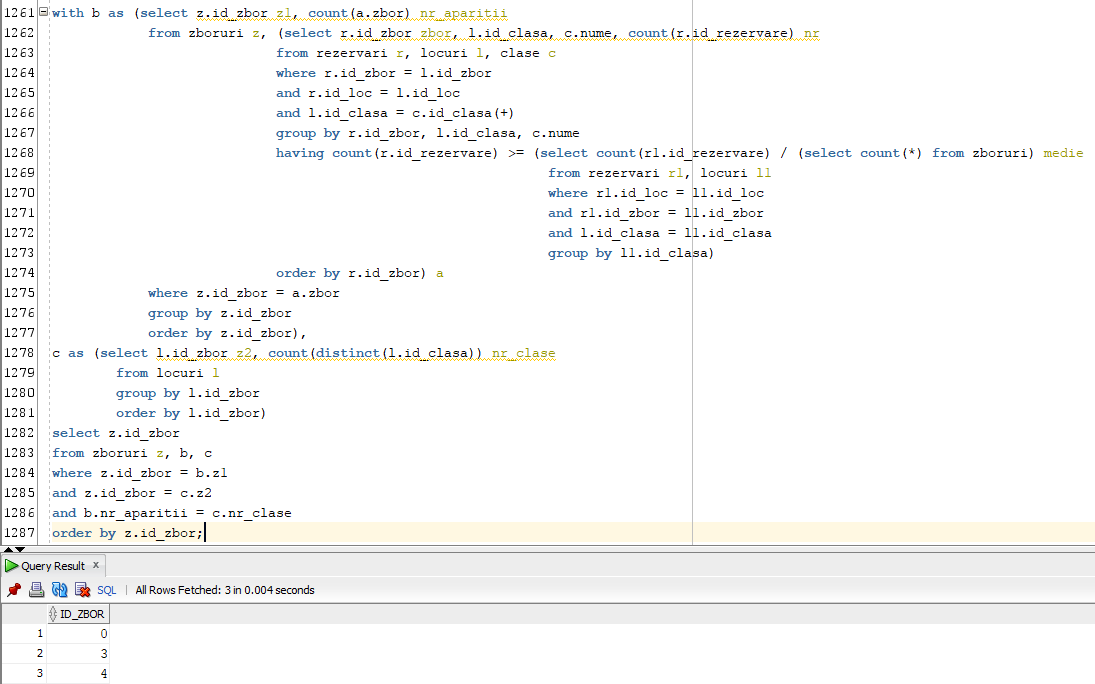
where z.id\_zbor = b.z1

and z.id\_zbor = c.z2

and b.nr\_aparitii = c.nr\_clase

order by z.id\_zbor;

* Print-screen + rezultate:



* Elemente: subcerere nesincronizată în clauza from, grupări de date cu subcereri nesincronizate in care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri, blocuri de cerere (clauza with)

# Operații de actualizare și suprimare a datelor

* 1) Actualizare
  + Cerință: Să se micșoreze cu 10% prețurile locurilor la clasa economy care nu au fost rezervate din zborurile la care au fost rezervate mai puțin de jumătate din locurile puse la vânzare la clasa economy.
  + Cod text:

update locuri

set pret = pret - 10/100 \* pret

where (id\_zbor, id\_loc) in (select l.id\_zbor, l.id\_loc

from locuri l, clase c

where l.id\_clasa = c.id\_clasa

and lower(c.nume) = 'economy'

and not exists (select r.id\_loc

from rezervari r

where r.id\_loc = l.id\_loc

and r.id\_zbor = l.id\_zbor)

and l.id\_zbor in (select l.id\_zbor

from locuri l, (select l1.id\_zbor z1, count(r1.id\_rezervare) nrr

from locuri l1, rezervari r1, clase c1

where c1.id\_clasa = l1.id\_clasa

and l1.id\_loc = r1.id\_loc

and l1.id\_zbor = r1.id\_zbor

and lower(c1.nume) = 'economy'

group by l1.id\_zbor) a,

(select l2.id\_zbor z2, count(l2.id\_loc) nrl

from locuri l2, clase c2

where c2.id\_clasa = l2.id\_clasa

and lower(c2.nume) = 'economy'

group by l2.id\_zbor) b

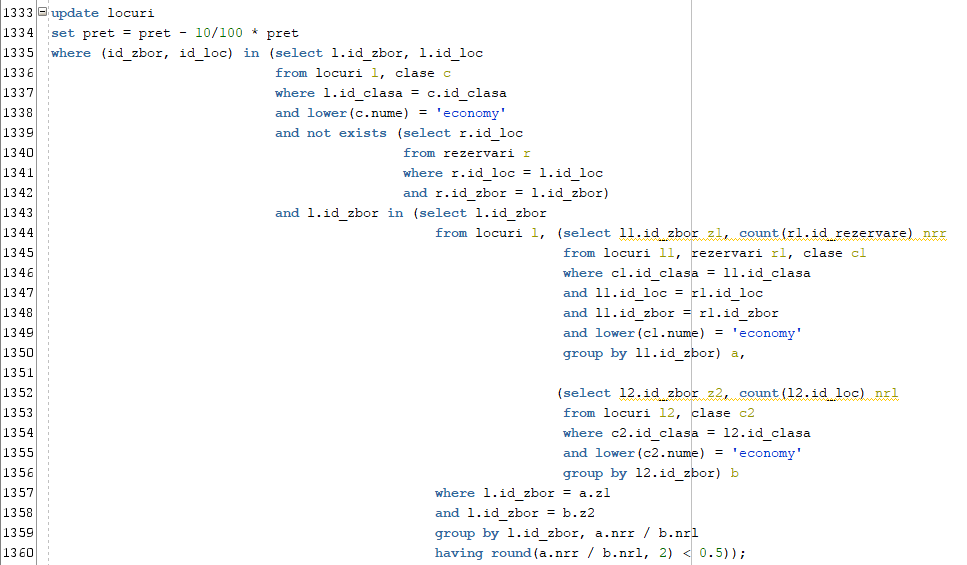
where l.id\_zbor = a.z1

and l.id\_zbor = b.z2

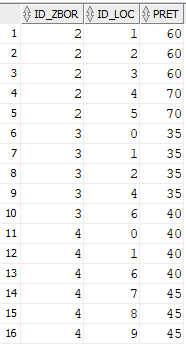
group by l.id\_zbor, a.nrr / b.nrl

having round(a.nrr / b.nrl, 2) < 0.5));

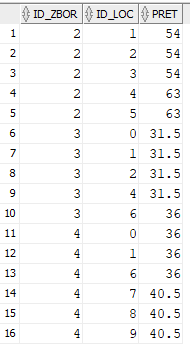
* Print-screen:



* Rândurile care vor fi actualizate:



* După acutalizare:



* 2) Actualizare:
  + Cerință: Să se schimbe tipul de plată din online în transfer bancar pentru toate plățile făcute pentru rezervări din zborurile cu cel mai mare număr de plăți online.
  + Cod text:

update plati

set tip\_plata = 'transfer bancar'

where id\_plata in (select p.id\_plata

from plati p, rezervari r

where p.id\_rezervare = r.id\_rezervare

and lower(p.tip\_plata) = 'online'

and r.id\_zbor in (select r.id\_zbor

from plati p, rezervari r

where p.id\_rezervare = r.id\_rezervare

and lower(p.tip\_plata) = 'online'

group by r.id\_zbor

having count(p.id\_plata) = (select max(count(p1.id\_plata))

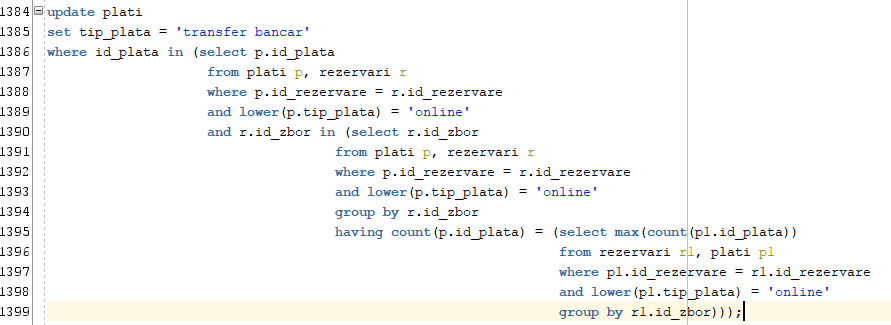
from rezervari r1, plati p1

where p1.id\_rezervare = r1.id\_rezervare

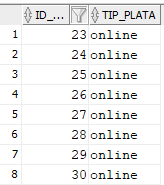
and lower(p1.tip\_plata) = 'online'

group by r1.id\_zbor)));

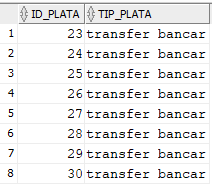
* Print-screen:



* Rândurile care vor fi actualizate:



* După actualizare:



* 3) Suprimare
  + Cerință: Să se șteargă toate orașele în care nu ajunge niciun zbor.
  + Cod text:

delete

from orase o

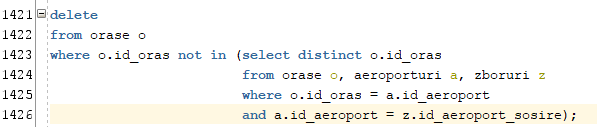
where o.id\_oras not in (select distinct o.id\_oras

from orase o, aeroporturi a, zboruri z

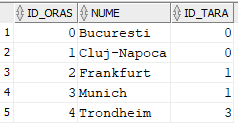
where o.id\_oras = a.id\_aeroport

and a.id\_aeroport = z.id\_aeroport\_sosire);

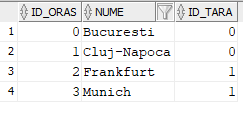
* Print-screen:



* Înainte de ștergere:



* După ștergere:



# Cereri SQL

## Outer-join pe minim 4 tabele

* Cerință: Pentru fiecare țară, să se afișeze toate orașele din acea țară, toate zborurile care au plecat din ea, cu ce avion și ce model de avion a fost folosit, inclusiv dacă nu există.
* Cod text:

select t.id\_tara, t.nume tara, o.id\_oras, o.nume oras, a.id\_aeroport, z.id\_zbor, a1.id\_avion, m.id\_model\_avion

from tari t

left outer join orase o on t.id\_tara = o.id\_tara

left outer join aeroporturi a on o.id\_oras = a.id\_oras

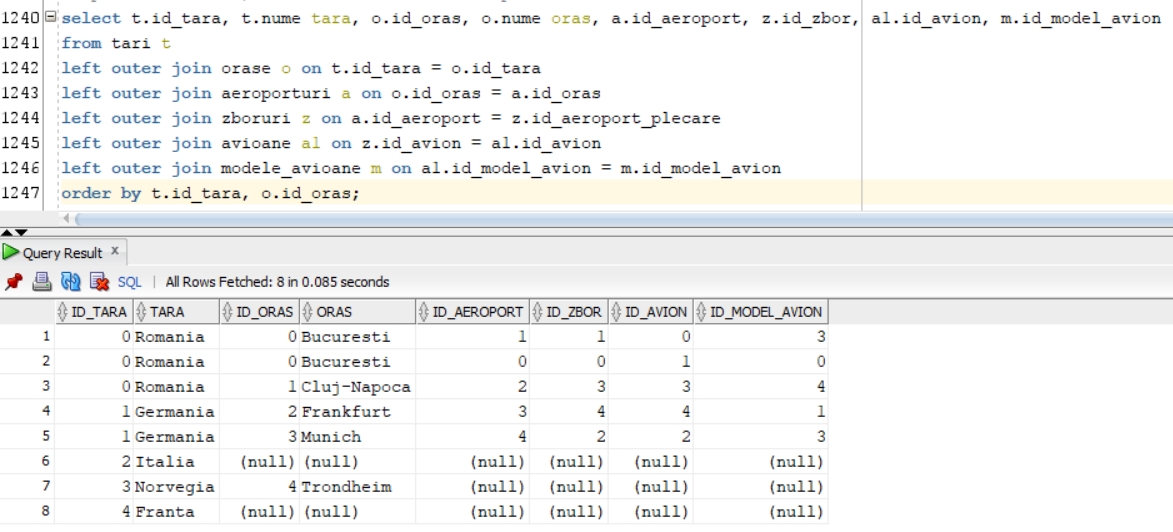
left outer join zboruri z on a.id\_aeroport = z.id\_aeroport\_plecare

left outer join avioane a1 on z.id\_avion = a1.id\_avion

left outer join modele\_avioane m on a1.id\_model\_avion = m.id\_model\_avion

order by t.id\_tara, o.id\_oras;

* Print-screen + rezultat:



## Division

* Cerință: Să se afișeze id-ul și numele pasagerilor care au avut rezervări în zboruri care au ajuns în toate aeroporturile în care au ajuns zborurile în care pasagerul cu id-ul 11 a avut rezervare.
* Cod text:

select distinct p1.id\_pasager, p1.nume

from pasageri p1, rezervari r1

where p1.id\_pasager = r1.id\_pasager

and not exists ((select a.id\_aeroport

from rezervari r, zboruri z, aeroporturi a

where r.id\_pasager = r1.id\_pasager

and r.id\_zbor = z.id\_zbor

and z.id\_aeroport\_sosire = a.id\_aeroport)

minus

(select a.id\_aeroport

from rezervari r, zboruri z, aeroporturi a

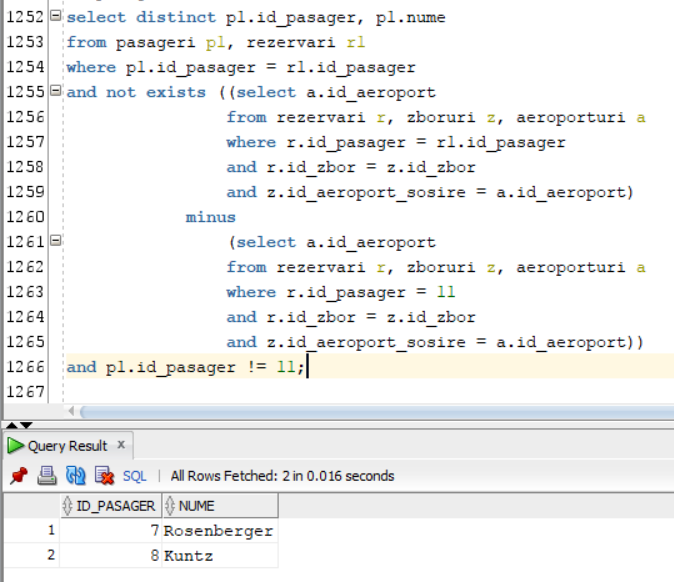
where r.id\_pasager = 11

and r.id\_zbor = z.id\_zbor

and z.id\_aeroport\_sosire = a.id\_aeroport))

and p1.id\_pasager != 11;

* Print-screen + rezultate:



## Analiza top-n

* Cerință: Să se afișeze id-ul și numele și numărul de ore de zbor ale primelor 3 cele mai folosite modele de avioane din punctul de vedere al orelor de zbor.
* Cod text:

select \*

from (select m.id\_model\_avion, m.nume\_model, round(sum(z.data\_sosire - z.data\_plecare) \* 24, 2) nr\_ore

from modele\_avioane m, avioane a, zboruri z

where m.id\_model\_avion = a.id\_model\_avion

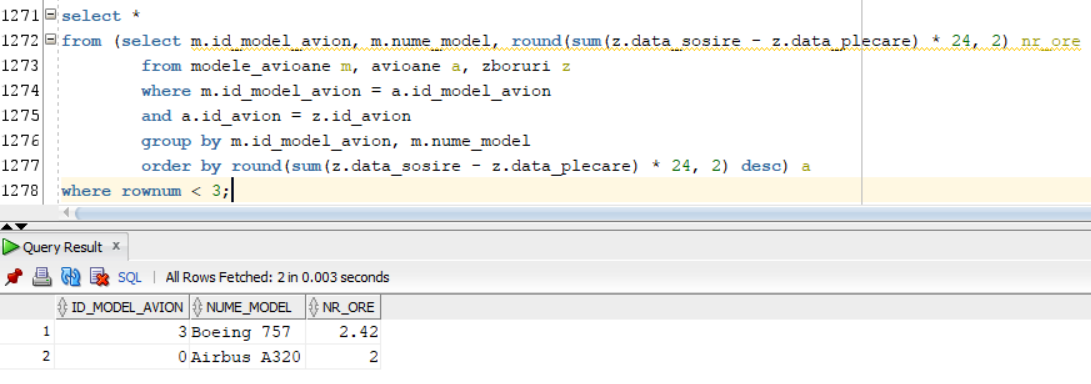
and a.id\_avion = z.id\_avion

group by m.id\_model\_avion, m.nume\_model

order by round(sum(z.data\_sosire - z.data\_plecare) \* 24, 2) desc) a

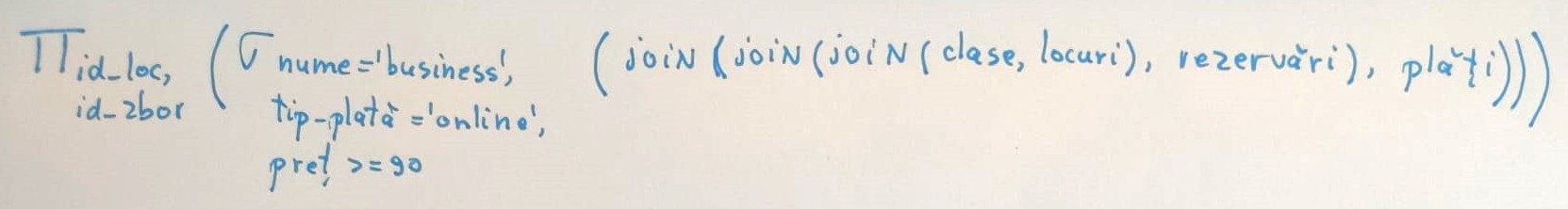
where rownum < 3;

* Print-screen + rezultate:

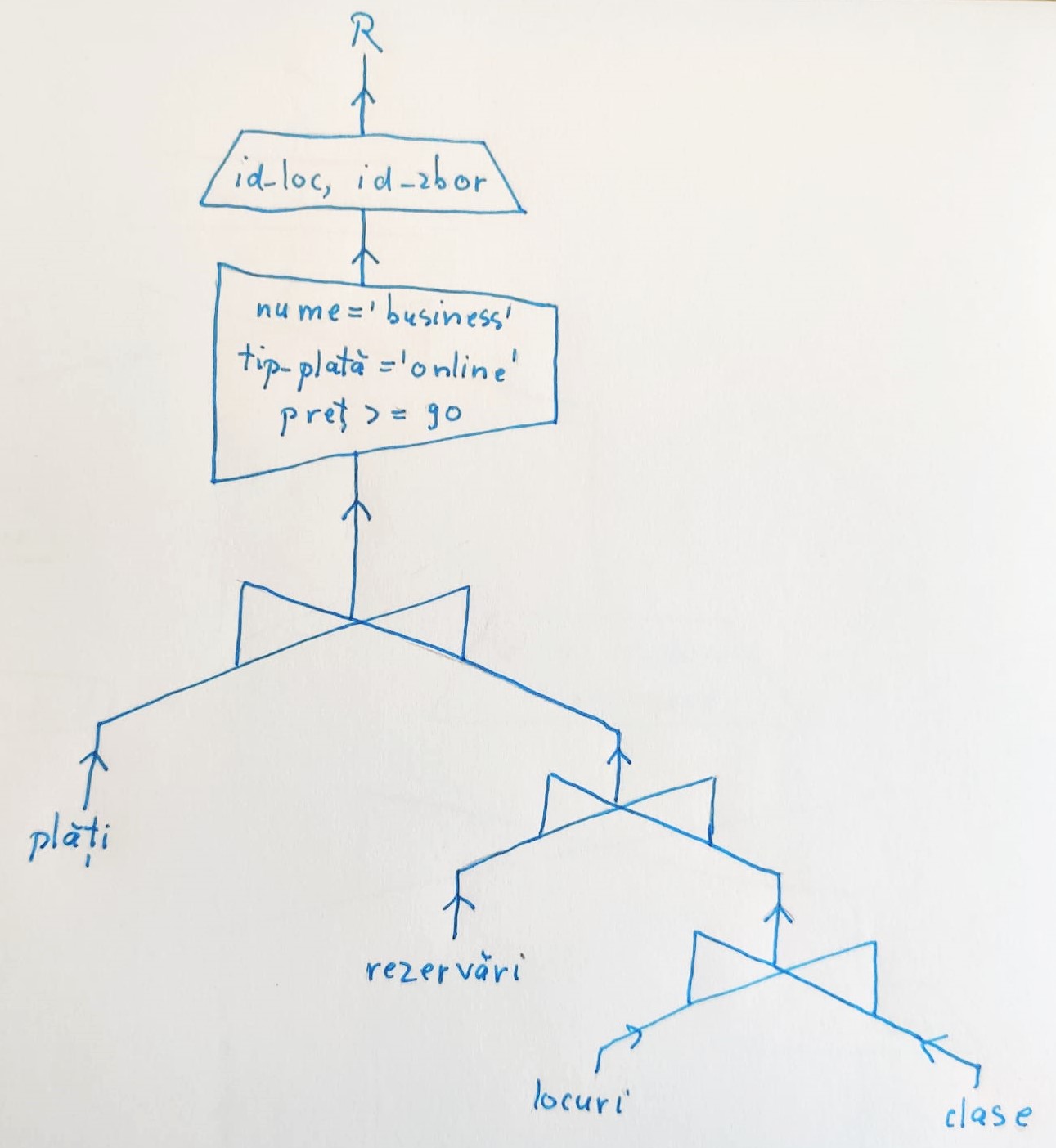


# Optimizarea unei cereri

* Cerință: Să se obțină locurile (id\_loc, id\_zbor) din clasa business care au fost rezervate, au fost plătite online și au avut prețul mai mare sau egal cu 90.
* Cererea înaintea optimizării:
  + Expresie algebrică:



* Arbore algebric:



* SQL text:

select l.id\_loc, l.id\_zbor

from locuri l, rezervari r, clase c, plati p

where c.id\_clasa = l.id\_clasa

and l.id\_loc = r.id\_loc

and l.id\_zbor = r.id\_zbor

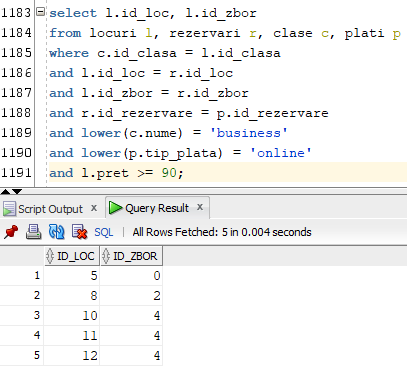
and r.id\_rezervare = p.id\_rezervare

and lower(c.nume) = 'business'

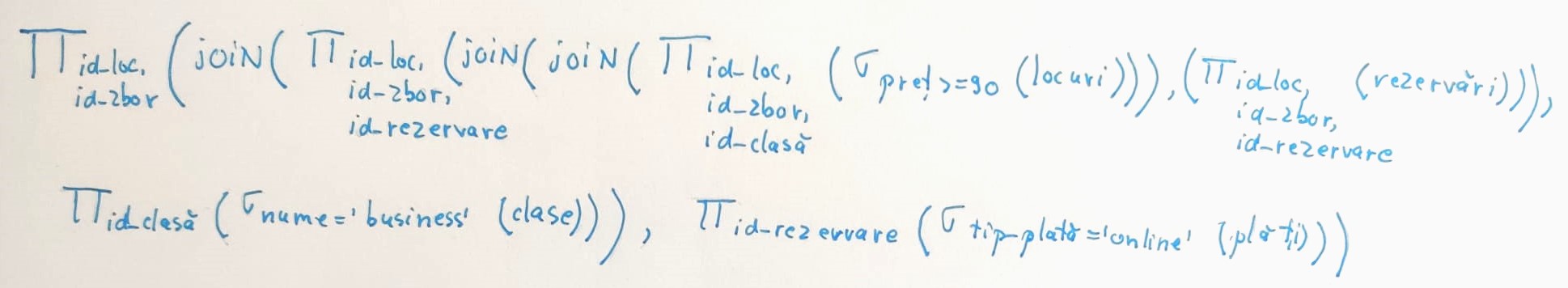
and lower(p.tip\_plata) = 'online'

and l.pret >= 90;

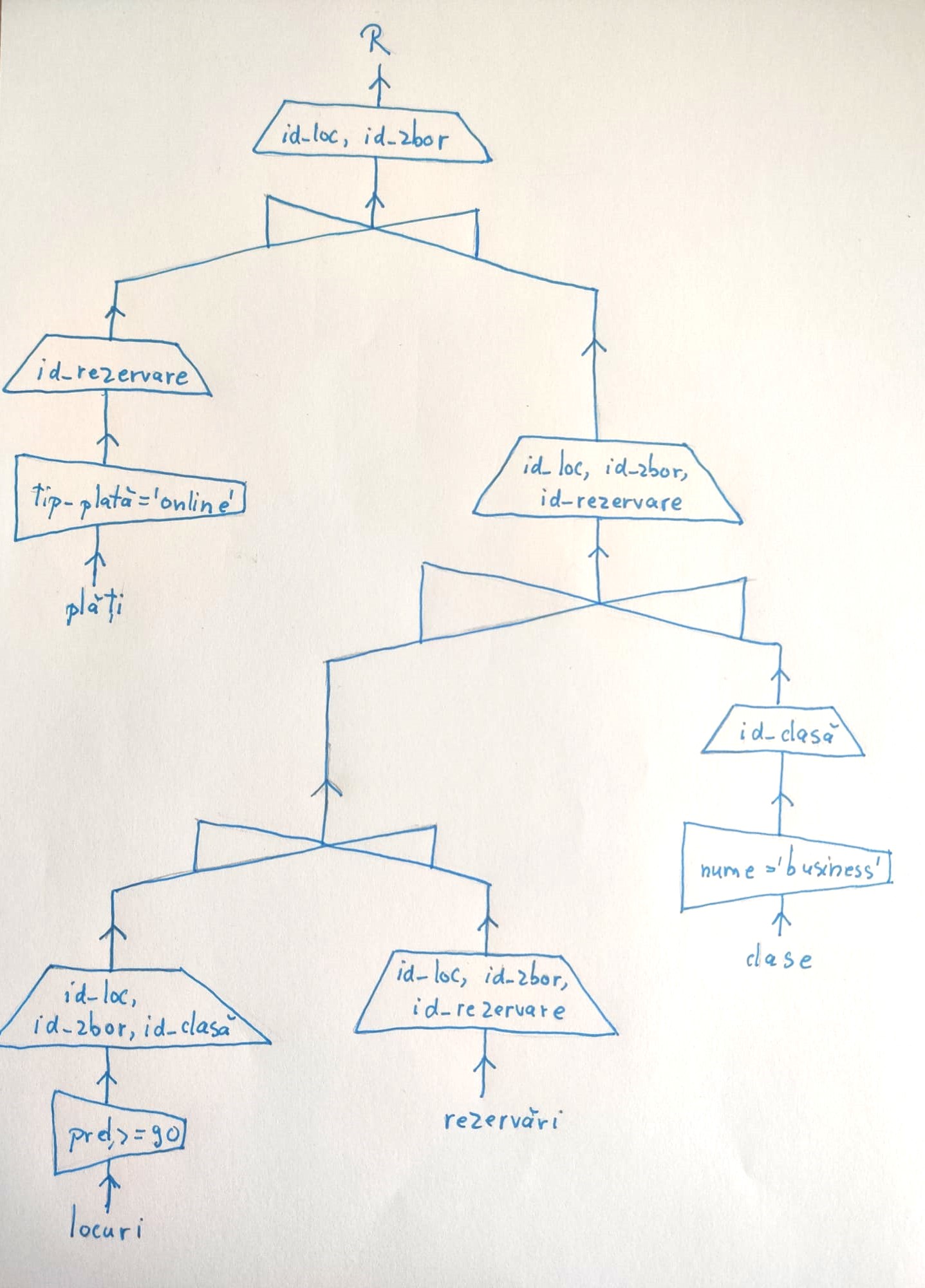
* SQL print-screen:



* Cererea după optimizare:
  + Expresie algebrică:



* + Arbore algebric:



* + SQL text:

select lrc.id\_loc, lrc.id\_zbor

from (select lr.id\_loc, lr.id\_zbor, lr.id\_rezervare

from (select l.id\_loc, l.id\_zbor, l.id\_clasa, r.id\_rezervare

from (select id\_loc, id\_zbor, id\_clasa

from locuri

where pret >= 90) l,

(select id\_loc, id\_zbor, id\_rezervare

from rezervari) r

where l.id\_loc = r.id\_loc

and l.id\_zbor = r.id\_zbor) lr,

(select id\_clasa

from clase

where lower(nume) = 'business') c

where c.id\_clasa = lr.id\_clasa) lrc,

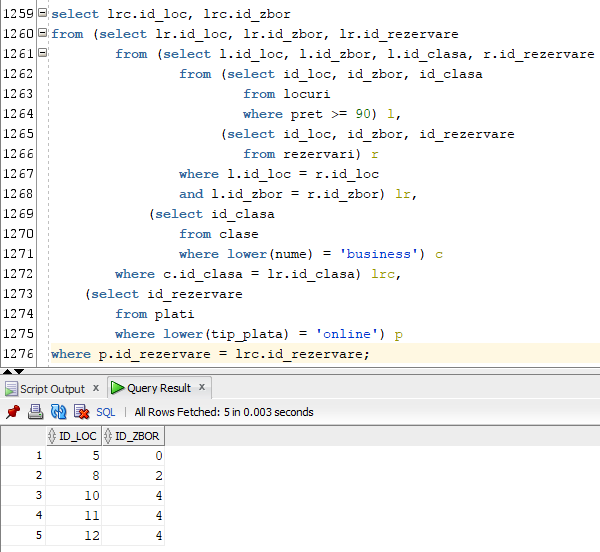
(select id\_rezervare

from plati

where lower(tip\_plata) = 'online') p

where p.id\_rezervare = lrc.id\_rezervare;

* + SQL print-screen:



# Normalizările BCNF, FN4, FN5

## BCNF

* Diagrama se află în BCNF deoarece
* Un exemplu de non-BCNF ar fi: (în tabelul rezervări)
* Impunem ca în fiecare zi să fie doar un singur zbor.

|  |  |  |
| --- | --- | --- |
| id\_zbor # | id\_pilot # | data |
| 0 | 3 | 03-Jan-22 |
| 0 | 1 | 03-Jan-22 |
| 1 | 2 | 16-Feb-22 |
| 1 | 4 | 16-Feb-22 |

* Atunci atributul data determină atributul id\_zbor, care face parte din cheia primară, dar nu îl determină și pe id\_pilot, care face parte din cheia primară.
* Rezolvare:

|  |  |  |
| --- | --- | --- |
| id\_zbor # | data | id\_pilot |
| 0 | 03-Jan-22 | 3 |
| 0 | 03-Jan-22 | 1 |
| 1 | 16-Feb-22 | 2 |
| 1 | 16-Feb-22 | 4 |

|  |
| --- |
| id\_pilot # |
| 1 |
| 2 |
| 3 |
| 4 |

## FN4

* Diagrama se află în FN4 deoarece se află în BCNF și nu conține relații m:n independente.
* Un exemplu de non-FN4 ar fi: (în tabelul zboruri)

|  |  |  |
| --- | --- | --- |
| id\_zbor # | id\_avion | id\_pilot |
| 0 | 1 | 0 |
| 0 | 1 | 3 |
| 1 | 2 | 1 |
| 1 | 2 | 2 |

* Cu acest tabel sunt probleme la inserare (deoarece nu putem insera un pilot care nu a participat la niciun zbor) și la ștergere (deoarece dacă vrem să ștergem un pilot trebuie să ștergem toate liniile în care a apărut)
* Rezolvare:

|  |  |
| --- | --- |
| id\_zbor # | id\_avion |
| 0 | 1 |
| 1 | 2 |

|  |  |
| --- | --- |
| id\_zbor # | id\_pilot # |
| 0 | 0 |
| 0 | 3 |
| 1 | 1 |
| 1 | 2 |

|  |
| --- |
| id\_pilot # |
| 0 |
| 1 |
| 2 |
| 3 |

## FN5

* Diagrama se află în FN5 deoarece se află în FN4 și deoarece nu conține dependențe ciclice.
* Un exemplu de non-FN5 ar fi dacă am avea o relație de tip 3 între tabelele rezervări, pasageri și check-in. Așa ar fi tabelul asociativ:

|  |  |  |
| --- | --- | --- |
| id\_rezervare # | id\_pasager # | id\_check\_in # |
| 0 | 1 | 3 |
| 1 | 2 | 4 |
| 2 | 5 | 1 |
| 3 | 2 | 6 |
| 4 | 1 | 2 |

* Pentru o rezervare nu poate să existe decât un check-in, deci după ce despărțim relația în 3 relații de tip 2, apoi facem toate join-urile, vom obține același rezultat cu cel din relația de tip 3.
* Rezolvare:

|  |
| --- |
| id\_pasager # |
| 1 |
| 2 |
| 5 |

|  |  |
| --- | --- |
| id\_rezervare # | id\_pasager |
| 0 | 1 |
| 1 | 2 |
| 2 | 5 |
| 3 | 2 |
| 4 | 1 |

|  |  |
| --- | --- |
| id\_check\_in # | id\_rezervare |
| 3 | 0 |
| 4 | 1 |
| 1 | 2 |
| 6 | 3 |
| 2 | 4 |