Tema laborator SGBD 2

Nume: Hodivoianu Anamaria

Grupa: 232

1. Identificați în diagrama Entitate-Relație utilizată în proiectul prezentat la materia Baze de Date din anul I, o relație din tip many-to-many. Dacă nu aveți o astfel de relație în acest proiect, atunci definiți una folosind tema aleasă în acel proiect:

a. descrieți entitățile și relația dintre acestea;

Zbor: zbor cu un avion de la un aeroport la altul; cheie primară: id\_zbor

Atribute:

Id\_zbor: cheie primară; number(4)

Dată\_plecare: date

Dată\_sosire: date

Însoțitor de zbor: angajat al companiei care participă la zbor; cheie primară: id\_insoțitor

Atribute:

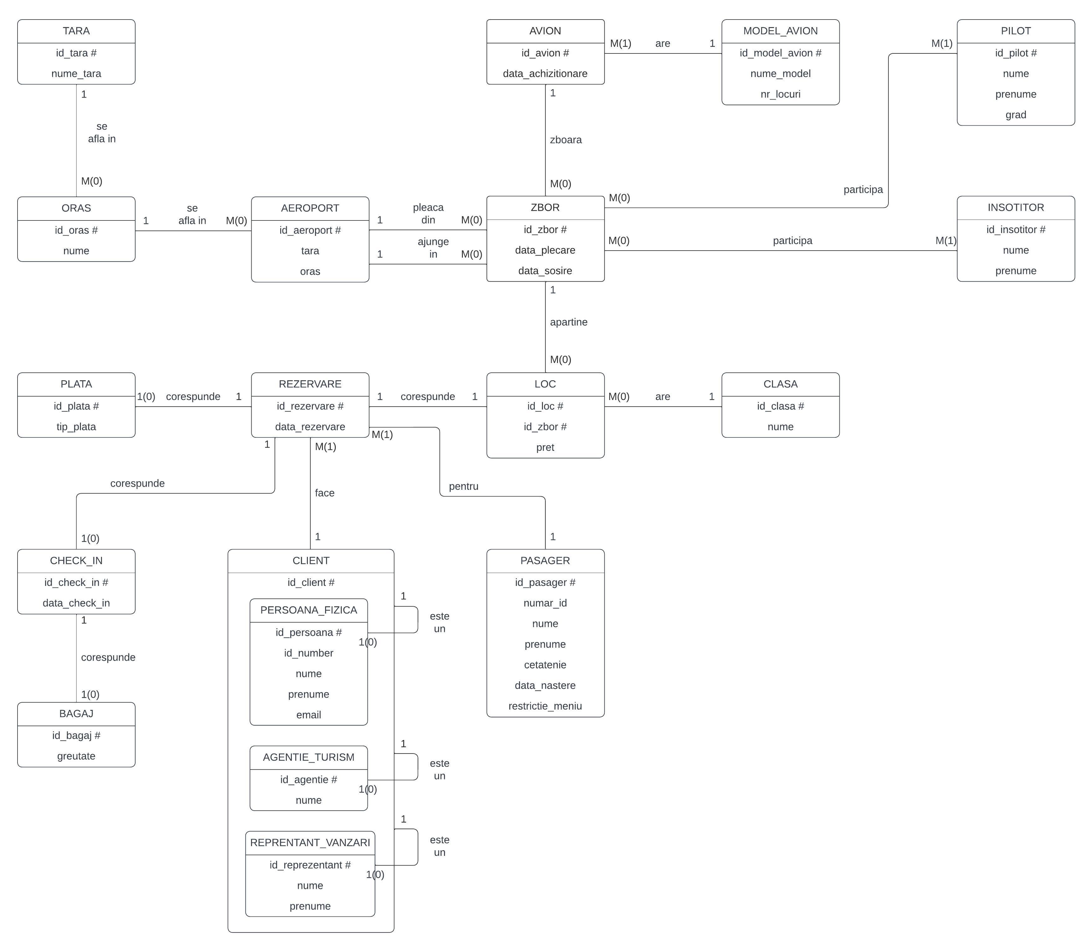
Id\_însoțitor: cheie primară; number(4)

Nume: varchar2(30); not null

Prenume: varchar2(30)

Relație: Un însoțitor de zbor participă la zero sau mai multe zboruri. La un zbor participă unul sau mai mulți însoțitori. M(1) - M(0)

b. realizați diagrama Entitate-Relație doar pentru această parte din proiect;



c. descrieți modul de transformare al acestei relații în diagrama conceptuală, precizând toate cheile primare, cheile externe și alte atribute esențiale;

Relatia se transforma intr-un tabel asociativ numit echipaj\_insotitori. Acest tabel are drept cheie primara compusa id\_zbor si id\_insotitor, ambele fiind si chei straine.

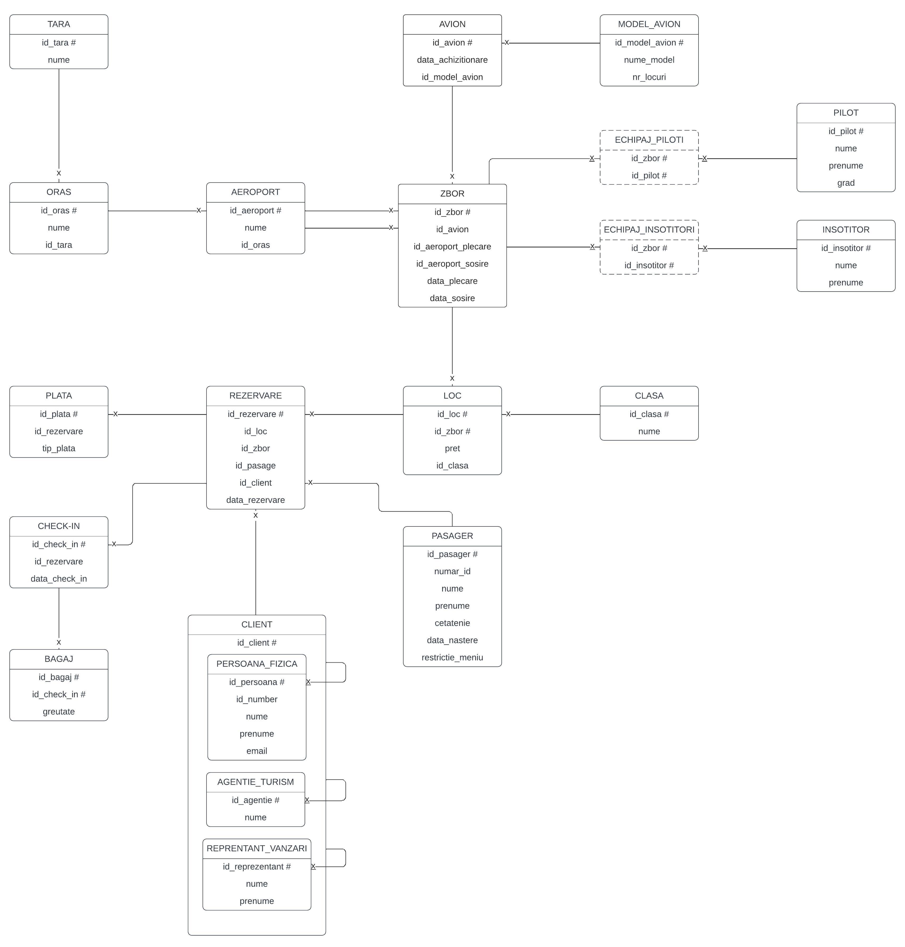
Echipaj însoțitori:

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

d. realizați diagrama conceptuală doar pentru această parte din proiect;



e. pe baza diagramei conceptuale de la punctul d, definiți în SQL tabelele și toate constrângerile necesare;

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));

Insotitori 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 insotitori:

create table echipaj\_insotitori (

id\_zbor number(4),

id\_insotitor number(4),

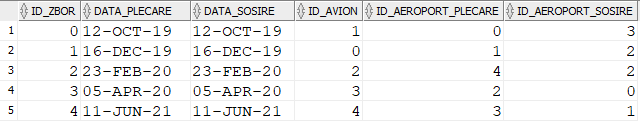
constraint pk\_echipaj\_insotitori primary key(id\_zbor, id\_insotitor),

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

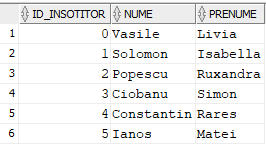
constraint fk\_echipaj\_insotitori\_insotitor foreign key(id\_insotitor) references insotitori\_zbor(id\_insotitor));

f. adaptați una dintre cerințele exercițiilor 4-12 pentru diagrama obținută la punctul d (formulați cerința în limbaj natural, inserați 5-10 înregistrări în fiecare tabelă utilizată, apoi rezolvați cererea propusă în SQL).

Zboruri:



Insotitori zbor:



Echipaj insotitori:

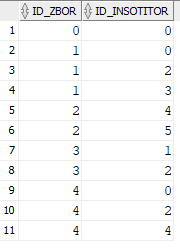
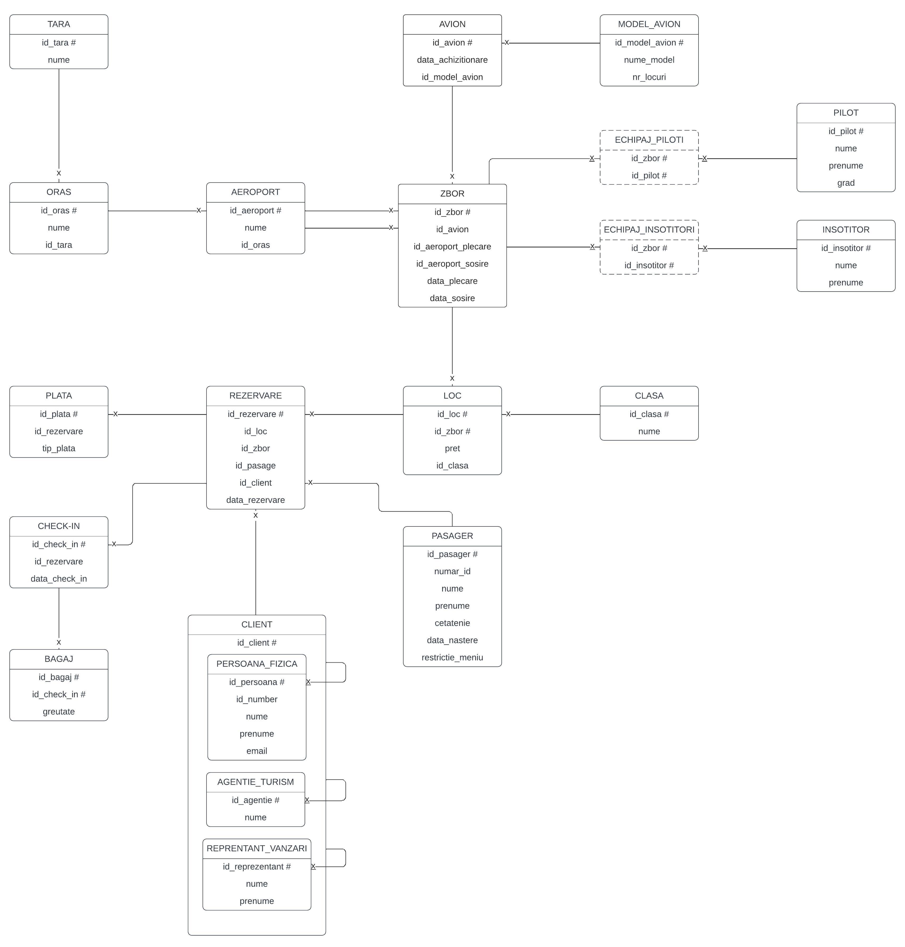
  
  
Adaptare cerinta 10: De cate ori a zburat un insotitor de zbor (id, nume, prenume) cu fiecare avion (id, id model)?

Diagrama pentru celelalte tabele:



Cod text:

select i.id\_insotitor, i.nume, i.prenume, a.id\_avion, a.id\_model\_avion, count(\*)

from insotitori\_zbor i, echipaj\_insotitori e, zboruri z, avioane a

where i.id\_insotitor = e.id\_insotitor

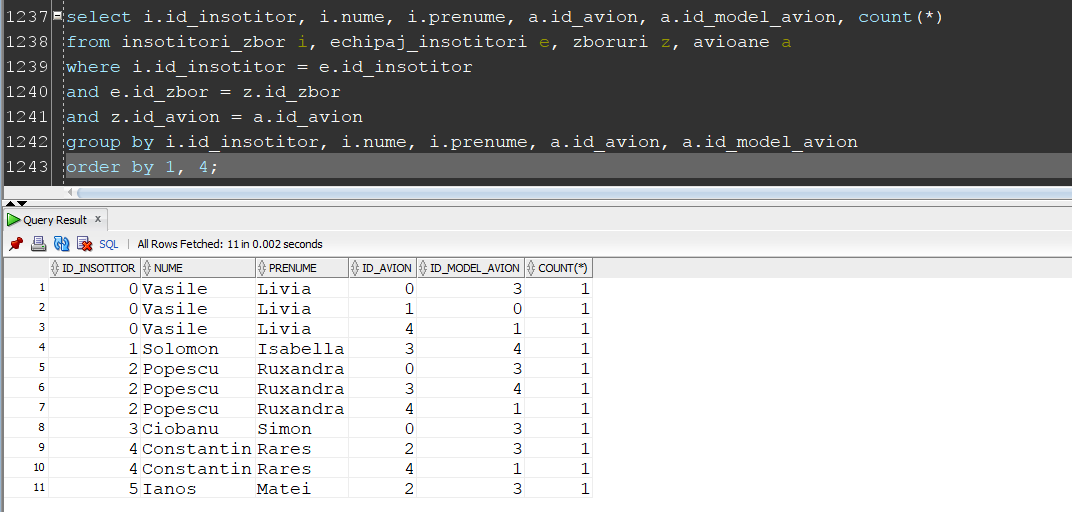
and e.id\_zbor = z.id\_zbor

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

group by i.id\_insotitor, i.nume, i.prenume, a.id\_avion, a.id\_model\_avion

order by 1, 4;

Cod screenshot:



2.(11) Obțineți statusul celui mai des împrumutat exemplar al fiecărui film (titlu).

Cod text:

select t.title\_id, tc.copy\_id, tc.status, count(tc.copy\_id)

from title t, title\_copy tc, rental r

where t.title\_id = tc.title\_id

and tc.title\_id = r.title\_id

and tc.copy\_id = r.copy\_id

group by t.title\_id, tc.copy\_id, tc.status

having count(tc.copy\_id) = (select max(count(tc2.copy\_id))

from title t2, title\_copy tc2, rental r2

where t2.title\_id = t.title\_id

and t2.title\_id = tc2.title\_id

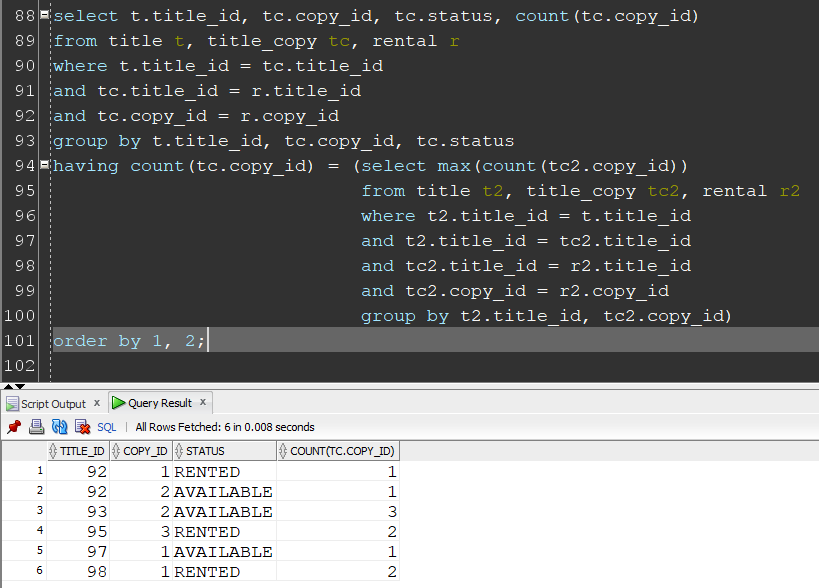
and tc2.title\_id = r2.title\_id

and tc2.copy\_id = r2.copy\_id

group by t2.title\_id, tc2.copy\_id)

order by 1, 2;

Cod screenshot:



3.(12) Pentru anumite zile specificate din luna curentă, obțineți numărul de împrumuturi efectuate.

a. Se iau în considerare doar primele 2 zile din lună.

Cod text:

declare

zi int;

nr int;

begin

dbms\_output.enable;

for zi in 1..2 loop

begin

select count(\*) into nr

from rental

where zi = extract(day from book\_date)

and extract(month from book\_date) = extract(month from sysdate)

group by zi;

dbms\_output.put\_line('Ziua ' || zi || ' imprumuturi ' || nr);

exception when no\_data\_found

then dbms\_output.put\_line('Ziua ' || zi || ' imprumuturi ' || 0);

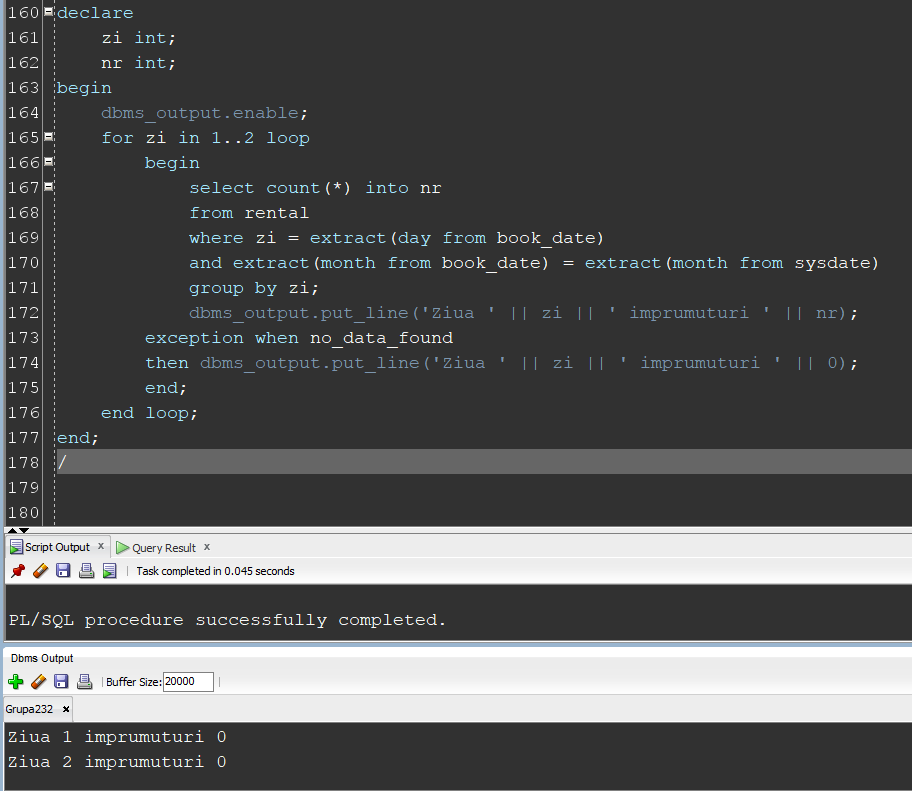
end;

end loop;

end;

/

Cod screenshot:



b. Se iau în considerare doar zilele din lună în care au fost efectuate împrumuturi.

Cod text:

select extract(day from book\_date) as zi, count(\*) as nr\_imprumuturi

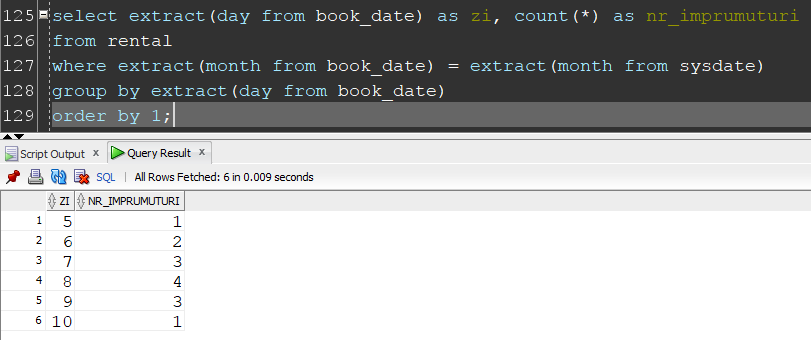
from rental

where extract(month from book\_date) = extract(month from sysdate)

group by extract(day from book\_date)

order by 1;

Cod screenshot:



c. Se iau în considerare toate zilele din lună, incluzând în rezultat și zilele în care nu au fost efectuate împrumuturi.

Cod text:

declare

zi int;

nr int;

begin

dbms\_output.enable;

for zi in 1..31 loop

begin

select count(\*) into nr

from rental

where zi = extract(day from book\_date)

and extract(month from book\_date) = extract(month from sysdate)

group by zi;

dbms\_output.put\_line('Ziua ' || zi || ' imprumuturi ' || nr);

exception when no\_data\_found

then dbms\_output.put\_line('Ziua ' || zi || ' imprumuturi ' || 0);

end;

end loop;

end;

/

Cod screenshot:

