Academia de Studii Economice

Facultatea de Cibernetică, Statistică și Informatică Economică

Proiect SGBD

Gestionarea unei baze de date in domeniul hotelier (mediul extern)

Profesor coordonator: Realizator:

Vlad Diaconita Adrian Hapseas

Seria A, grupa 1039

-Bucuresti-

2016

Introducere-Prezentarea proiectului

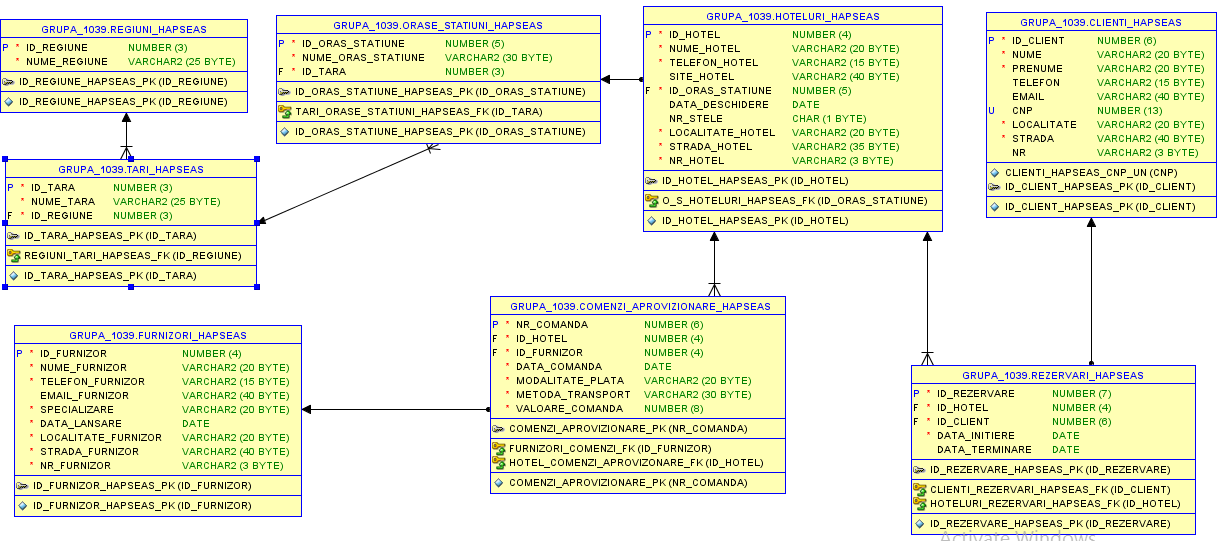
Baza de date ce am folosit-o in acest proiect am utilizat-o si in semestrul trecut pentru disciplina „Baze de date”.Aceasta avea ca scop gestionarea activitatilor hoteliere, mai precis partea ce tine de mediul extern al companiilor:clienti, furnizori, plasament, etc

Baza de date contine urmatoarele tabele :

Regiuni\_hapseas,Tari\_hapseas,orase\_statiuni\_hapseas,hoteluri\_hapseas,clienti\_hapseas,rezervari\_hapseas,furnizori\_hapseas,comenzi\_aprovizionare\_hapseas

Toate relatiile ce exista intre tabele sunt relatii de tip one to many (Initial existau 2 relatii de tip many to many intre hoteluri\_hapseas si clienti\_hapseas, si intre hoteluri\_hapseas si furnizori\_hapseas.Aceasta problema a fost rezolvata prin introducerea tabelelor rezervari\_hapseas si comenzi\_aprovizionare\_hapseas)

Schema bazei de date:



I.Interacțiunea cu serverul Oracle prin intermediul comenzilor SQL (LDD şi LMD):

1.Sa se creeze o tabela identica cu tabela orase\_statiuni\_hapseas:

SET SERVEROUTPUT ON

Begin execute immediate ’CREATE table oras\_hapseas1 AS SELECT \* FROM orase\_statiuni\_hapseas’;

end;

/

2.Sa se afiseze Numele si ID\_Tara pentru orasul cu ID\_ul egal cu 3 din tabela creeata anterior:

SET SERVEROUTPUT ON

declare

h\_ido oras\_hapseas1.id\_oras\_statiune%type;

h\_nume oras\_hapseas1.nume\_oras\_statiune%type;

h\_idt oras\_hapseas1.id\_tara%type;

begin

Select id\_oras\_statiune,nume\_oras\_statiune,id\_tara into h\_ido,h\_nume,h\_idt from oras\_hapseas1 where id\_oras\_statiune=3;

DBMS\_OUTPUT.PUT\_LINE ('Nume Oras:'||h\_nume||' ID Tara:'||h\_idt);

end;

/

3.Sa se introduca in tabela oras\_hapseas1 coloana „nr\_locuitori”

SET SERVEROUTPUT ON

Declare

H\_NRLOC number(10);

begin

H\_NRLOC:='ALTER TABLE oras\_hapseas1 ADD (Nr\_locuitori NUMBER(7))';  
DBMS\_OUTPUT.PUT\_LINE (H\_NRLOC);  
EXECUTE IMMEDIATE H\_NRLOC;

End;

/

4.Sa se introduca in tabela oras\_hapseas1 o valoarea pe coloana „nr\_locuitori” unde id\_oras\_statiune este egal cu 7;

Sa se introduca in tabela clienti o valoare pe coloana age unde id\_client=2;

Set SERVEROUTPUT ON

Accept key\_nrloc prompt 'Introduceti numarul de locuitori:';

declare

h\_nrloc number(10):=&key\_nrloc;

begin

update oras\_hapseas1

set nr\_locuitori=h\_nrloc

where id\_oras\_statiune=7;

end;

/

select \*from oras\_hapseas1;

Numarul de locuitori introdus a fost:2000183

5.Sa se stearga tabela creata anterior

SET SERVEROUTPUT ON

begin

execute immediate 'DROP table oras\_hapseas1';

end;

/

II. Structuri alternative și repetitive

6. Afisati numele si prenumele clientului ce sta in localitatea “Pipera” si verificati daca au lungimi egale. Daca au, sa se afiseze numele, daca nu, sa se afiseze prenumele.

set serveroutput on

declare

nume\_h clienti\_hapseas.nume%type;

prenume\_h clienti\_hapseas.prenume%type;

begin

select nume,prenume into nume\_h,prenume\_h from clienti\_hapseas where localitate='Pipera';

if length(nume\_h)!=length(prenume\_h) then

dbms\_output.put\_line ('Prenume:'||prenume\_h);

else

dbms\_output.put\_line ('Nume:'||nume\_h);

end if;

end;

/

7. Sa se modifice valoare comenzii, din tabela comenzi\_aprovizionare\_hapseas al carei numar va fi introdus de la tastatura. Fiecare modificare va depinde de valoare introdusa.

set serveroutput on

accept nr prompt 'Introduceti nr\_comanda';

declare

h\_nr number(5):=&nr;

h\_valc comenzi\_aprovizionare\_hapseas.valoare\_comanda%type;

nr comenzi\_aprovizionare\_hapseas.nr\_comanda%type;

begin

select valoare\_comanda into h\_valc from comenzi\_aprovizionare\_hapseas where nr\_comanda=h\_nr;

dbms\_output.put\_line ('Valoarea initiala a comenzii este:'||h\_valc);

h\_valc:= case

when nr<4 then 1.2\*h\_valc

when nr between 4 and 7 then 1.5\*h\_valc

else 1.75\*h\_valc

end;

dbms\_output.put\_line ('Valoarea finala a comenzii este:'||h\_valc);

end;

/

8.Sa se afiseze valoarea comenzii,modalitatea de plata si modul de transport pentru comenzile ce au nr\_comanda cu valori de la 1 la 8.

set serveroutput on

declare

h\_n number(2):=1;

h\_val comenzi\_aprovizionare\_hapseas.valoare\_comanda%type;

h\_mod comenzi\_aprovizionare\_hapseas.modalitate\_plata%type;

h\_trans comenzi\_aprovizionare\_hapseas.metoda\_transport%type;

begin

while h\_n<=8 loop

select valoare\_comanda,modalitate\_plata,metoda\_transport into h\_val,h\_mod,h\_trans

from comenzi\_aprovizionare\_hapseas where nr\_comanda=h\_n;

dbms\_output.put\_line ('Valoare:'||h\_val||' Mod Plata:'||h\_mod||' Metoda transport:'||h\_trans);

h\_n:=h\_n+1;

end loop;

end;

/

III. Tratarea excepțiilor

9. Sa se afiseze valoarea comenzii si metoda de transport pentru comanda cu numarul 27. Daca comanda cu numarul respectiv nu exista, sa se trateze exceptia.

set serveroutput on

declare

h\_val comenzi\_aprovizionare\_hapseas.valoare\_comanda%type;

h\_trans comenzi\_aprovizionare\_hapseas.metoda\_transport%type;

begin

select valoare\_comanda,metoda\_transport into h\_val,h\_trans

from comenzi\_aprovizionare\_hapseas where nr\_comanda=27;

dbms\_output.put\_line ('Valoare:'||h\_val||' Metoda transport:'||h\_trans);

exception

when no\_data\_found then

dbms\_output.put\_line ('Nu exista comanda respectiva');

end;

/

9. Sa se afiseze Numele hotelului,nr de stele si localitatea hotelului pentru hotelul ce are id-ul ce urmeaza sa fie introdus de la tastatura. Daca hotelul nu exista, sa se trateze exceptia.

set serveroutput on

accept nr prompt 'Introducet ID hotel:';

declare

h\_nr number(4):=&nr;

h\_nume hoteluri\_hapseas.nume\_hotel%type;

h\_stele hoteluri\_hapseas.nr\_stele%type;

h\_loc hoteluri\_hapseas.localitate\_hotel%type;

begin

select nume\_hotel,nr\_stele,localitate\_hotel into h\_nume,h\_stele,h\_loc

from hoteluri\_hapseas where id\_hotel=h\_nr;

dbms\_output.put\_line ('Nume Hotel:'||h\_nume||' Nr.Stele:'||h\_stele||' Localitate:'||h\_loc);

exception

when no\_data\_found then

dbms\_output.put\_line ('Nu exista hotelul respectiv');

end;

/

10. Sa se afiseze numele hotelului, nr de stele si localitatea hotelului pentru hotelul ce are id-ul statiunii ce urmeaza sa fie introdus de la tastatura. Daca blocul returneaza mai multe randuri, sa se trateze exceptia.

set serveroutput on

accept nr prompt ' Introduceti ID Statiune:';

declare

h\_nr number(4):=&nr;

h\_nume hoteluri\_hapseas.nume\_hotel%type;

h\_stele hoteluri\_hapseas.nr\_stele%type;

h\_loc hoteluri\_hapseas.localitate\_hotel%type;

begin

select nume\_hotel,nr\_stele,localitate\_hotel into h\_nume,h\_stele,h\_loc

from hoteluri\_hapseas where id\_oras\_statiune=h\_nr;

dbms\_output.put\_line ('Nume Hotel: ' || h\_nume || ' Nr.Stele: ' || h\_stele || ' Localitate:' || h\_loc);

exception

when too\_many\_rows then

dbms\_output.put\_line ('Returneaza prea multe randuri');

when no\_data\_found then

dbms\_output.put\_line ('Nu exista statiunea respectiva');

end;

/

11.Sa se introduca o noua linie in Tabela regiuni ce primeste numele NULL si ID\_ul 12.Daca numele regiuni nu poate primi valoarea NULL, sa se trateze exceptia.

set serveroutput on

declare

nul\_val exception;

pragma exception\_init(nul\_val,-01400);

begin

insert into regiuni\_hapseas(id\_regiune,nume\_regiune)

Values(12,NULL);

exception

when nul\_val then

dbms\_output.put\_line('Nu pot fi inserate valori nule');

end;

/

12.Sa se modifice numele tarii al carei ID este introdus de la tastatura.Daca ID-ul nu exista sa se trateze exceptia.

set serveroutput on

accept nr prompt 'Introduceti ID-ul regiunii';

declare

not\_found exception;

h\_nr number(4):=&nr;

h\_id regiuni\_hapseas.id\_regiune%type;

h\_nume regiuni\_hapseas.nume\_regiune%type;

begin

update regiuni\_hapseas

set nume\_regiune='Regiune'

where id\_regiune=h\_nr;

if sql%notfound then

raise not\_found;

end if;

exception

when not\_found then

dbms\_output.put\_line('Nu exista regiunea cu acel ID');

end;

/

IV Gestionarea cursorilor

13.Sa se afiseze modifice prenumele tuturor clientilor din Bucuresti in Mihai si apoi sa se afiseze cate linii au fost modificate.

set serveroutput on

declare

nr number(3);

h\_id clienti\_hapseas.id\_client%type;

begin

update clienti\_hapseas

set prenume='Mihai'

where localitate='Bucuresti';

nr:=sql%rowcount;

dbms\_output.put\_line('Numarul de clienti din Bucuresti este egal cu '||nr);

end;

/

14.Aflati ce comenzi de aprovizionare au o valoare mai mare decat valoarea primita ca parametru.Sa se afiseze pentru fiecare ID-ul, metoda de plata si transport si valoarea comenzii.

set serveroutput on

accept val prompt 'Introduceti valoarea';

declare

c\_val number(10):=&val;

cursor h(com\_val number) is select nr\_comanda,modalitate\_plata,metoda\_transport,valoare\_comanda

from comenzi\_aprovizionare\_hapseas where valoare\_comanda>com\_val;

rec h%rowtype;

begin

dbms\_output.put\_line('Comenzile ce au o valoarea mai mare decat'||c\_val);

if not h%isopen then open h(c\_val);

end if;

loop

fetch h into rec;

exit when h%notfound;

dbms\_output.put\_line('Nr Comanada:'||rec.nr\_comanda||' Mod Plata:'||rec.modalitate\_plata||' Metoda Transport:'||rec.metoda\_transport||' Valoare:'||rec.valoare\_comanda);

end loop;

close h;

end;

/

15.Sa se afiseze ID-ul,numele si nr de stele pentru fiecare hotel ce se afla in Dubai

set serveroutput on

declare

cursor h is select id\_hotel,nume\_hotel,nr\_stele

from hoteluri\_hapseas where localitate\_hotel='Dubai';

rec h%rowtype;

begin

dbms\_output.put\_line('Hotelurile inregistrate ce se afla in Dubai sunt:');

if not h%isopen then open h;

end if;

loop

fetch h into rec;

exit when h%notfound;

dbms\_output.put\_line('ID Hotel:'||rec.id\_hotel||' Nume Hotel:'||rec.id\_hotel||' Nr Stele:'||rec.nr\_stele);

end loop;

close h;

end;

/

V.Funcții, proceduri, includerea acestora în pachete.

16.Verificati daca valoarea unei comenzi din tabela comenzi\_aprovizionare este mai mare decat o valoare introdusa de la tastatura. Sa se foloseasca o functie ce primeste ca parametru id-ul comenzii si valoarea ce trebuie comparata.

create or replace function val\_com

(nr\_c in comenzi\_aprovizionare\_hapseas.nr\_comanda%type,val in number)

return boolean

is

valoare comenzi\_aprovizionare\_hapseas.valoare\_comanda%type;

Begin

select valoare\_comanda into valoare from comenzi\_aprovizionare\_hapseas

where nr\_comanda=nr\_c;

if valoare>val then

return true;

else return false;

end if;

exception

when no\_data\_found then

return null;

end;

/

set serveroutput on

accept v prompt 'Introduceti valoare ce trebuie comparata:';

accept nr prompt 'Introduceti nr-ul comenzii:';

declare

val number(10):=&v;

nr\_c number(5):=&nr;

begin

if (val\_com(nr\_c,val)is NULL) then

dbms\_output.put\_line('Nu exista comanda respectiva');

elsif (val\_com(nr\_c,val))then

dbms\_output.put\_line('Comanda are o valoare mai mare decat cea introdusa');

else

dbms\_output.put\_line('Comanda are o valoare mai mica decat cea introdusa');

end if;

end;

/

17.Sa se verifice care este valoarea medie a comenzilor din tabela comenzi aprovizionare folosind o functie.

create or replace function val\_med (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

return number is

begin

select AVG(valoare\_comanda) into valoare from comenzi\_aprovizionare\_hapseas;

return valoare;

end;

/

Set serveroutput on

declare valoare number;

begin

dbms\_output.put\_line(val\_med(valoare));

end;

/

18.Rezolvati cerinta de mai sus utilizand o procedura.

Create or replace procedure valoare\_med (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

is

begin

select AVG(valoare\_comanda) into valoare from comenzi\_aprovizionare\_hapseas;

end;

/

variable nr number;

execute valoare\_med(:nr);

Print nr

19.Sa se creeze o functie care primeste ca parametru id-ul unui client si returneaza prenumele sau.

create or replace function prenume\_c

(id\_c number)

return clienti\_hapseas.prenume%type

is

pr CLIENTI\_HAPSEAS.PRENUME%type;

begin

select prenume into pr from clienti\_hapseas where id\_client=id\_c;

return pr;

end;

/

accept id\_c prompt 'Introduceti id-ul clientului:';

declare

pre\_c clienti\_hapseas.prenume%type;

h\_id\_c number(4):=&id\_c;

begin

pre\_c:=prenume\_c(h\_id\_c);

dbms\_output.put\_line(pre\_c);

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista clientul cu acel ID');

end;

/

20.Sa se creeze o procedura ce calculeaza valoarea totala a comenzilor din tabela comenzi\_aprovizionare

select\*from comenzi\_aprovizionare\_hapseas;

create or replace procedure v\_totala (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

is begin

Select sum(valoare\_comanda) into valoare from comenzi\_aprovizionare\_hapseas;

end;

/

variable nr number

execute v\_totala(:nr);

Print nr

21.Sa se creeze un pachet ce contine functiile si procedurile de mai sus

Create or replace package pachet\_h is

function val\_com

(nr\_c in comenzi\_aprovizionare\_hapseas.nr\_comanda%type,val in number)

return boolean;

function val\_med (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

return number;

procedure valoare\_med

(valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type);

function prenume\_c

(id\_c number)

return clienti\_hapseas.prenume%type;

procedure v\_totala

(valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type);

end;

/

Create or replace package body pachet\_h is

function val\_com

(nr\_c in comenzi\_aprovizionare\_hapseas.nr\_comanda%type,val in number)

return boolean

is

valoare comenzi\_aprovizionare\_hapseas.valoare\_comanda%type;

Begin

select valoare\_comanda into valoare from comenzi\_aprovizionare\_hapseas

where nr\_comanda=nr\_c;

if valoare>val then

return true;

else return false;

end if;

exception

when no\_data\_found then

return null;

end;

function val\_med (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

return number is

begin

select AVG(valoare\_comanda) into valoare from comenzi\_aprovizionare\_hapseas;

return valoare;

end;

procedure valoare\_med (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

is

begin

select AVG(valoare\_comanda) into valoare from comenzi\_aprovizionare\_hapseas;

end;

function prenume\_c

(id\_c number)

return clienti\_hapseas.prenume%type

is

pr CLIENTI\_HAPSEAS.PRENUME%type;

begin

select prenume into pr from clienti\_hapseas where id\_client=id\_c;

return pr;

end;

procedure v\_totala (valoare out comenzi\_aprovizionare\_hapseas.valoare\_comanda%type)

is begin

Select sum(valoare\_comanda) into valoare from comenzi\_aprovizionare\_hapseas;

end;

end;

/

VI Declanșatori la nivel de instrucțiune și de rând

22.Sa se creeze un trigger care sa nu permita introducerea valorilor dupa ora 15.

create or replace trigger Ora

before insert on comenzi\_aprovizionare\_hapseas

begin

if inserting and to\_char(sysdate,'hh24:mi')>'15:00' then

raise\_application\_error(-20000,'Nu sunt permise adaugari dupa ora 15');

end if;

end;

/

23.Sa se creeze un trigger care sa nu permita modificare Localitatii unui client.

create or replace trigger loc

before update of localitate on clienti\_hapseas

begin

raise\_application\_error(-20000,'Nu este posibila modificare localitatii!');

end;

/

24.Sa se creeze un trigger care se declanseaza atunci cand se incearca introducerea sau modificarea unei valori ce depaseste 12000000 pentru comenzile din tabela comenzi\_aprovizionare.

create or replace trigger valoare

before insert or update of valoare\_comanda on comenzi\_aprovizionare\_hapseas

for each row

declare

val number(10):=12000000;

begin

if :new.valoare\_comanda>val then

RAISE\_APPLICATION\_ERROR (-20202, 'Nu se poate depasi valoarea de 12000000');

end if;

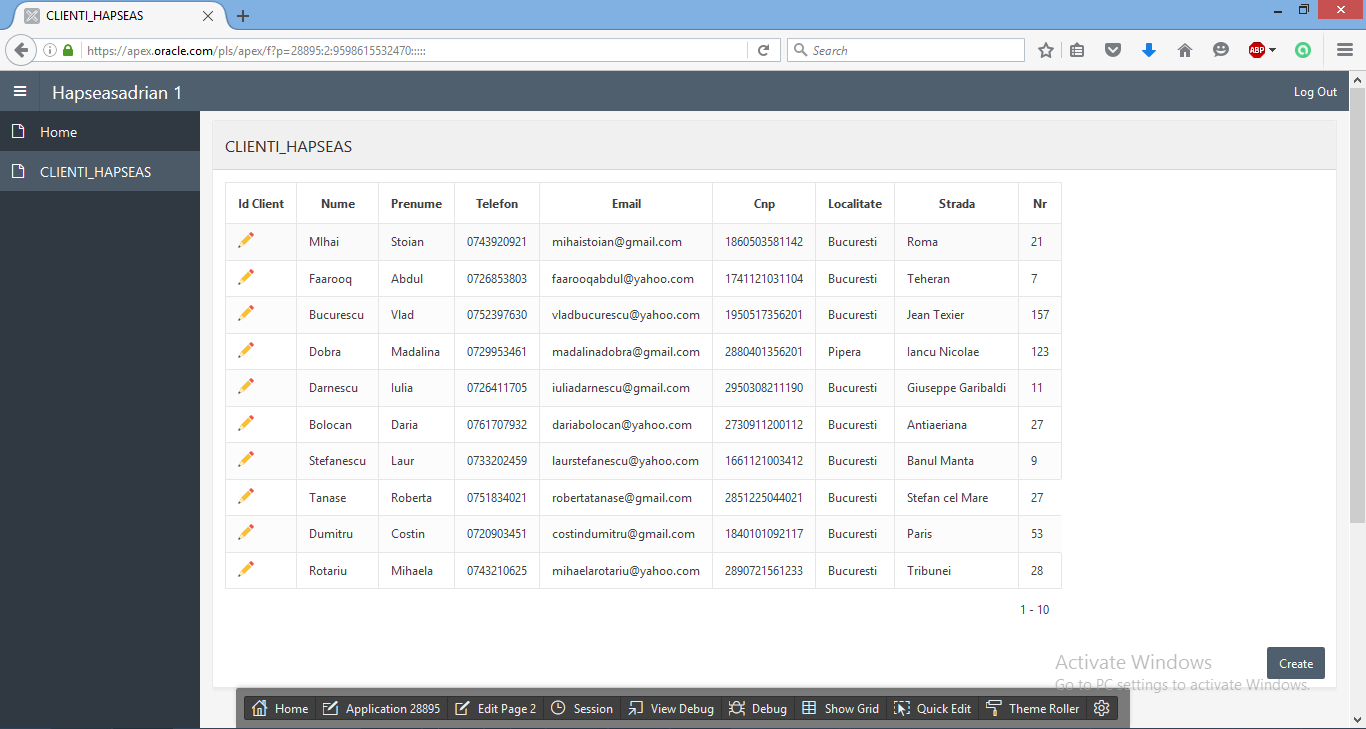
end;

/

VII.Machete de intrare (videoformate, Forms) si machete de iesire (rapoarte)

25.Construiti un videoformat (Hapseasadrian1), avand ca Master Table clienti\_hapseas, iar ca Detail Table rezervari\_hapseas.

<https://apex.oracle.com/pls/apex/f?p=28895:2:9598615532470::NO:::>

26.Sa se construiasca un raport ce afiseaza clienti ce au fost cazati la hoteluri si data de cand au initalizat rezervarea.

select ID\_REZERVARE,

DATA\_INITIERE,

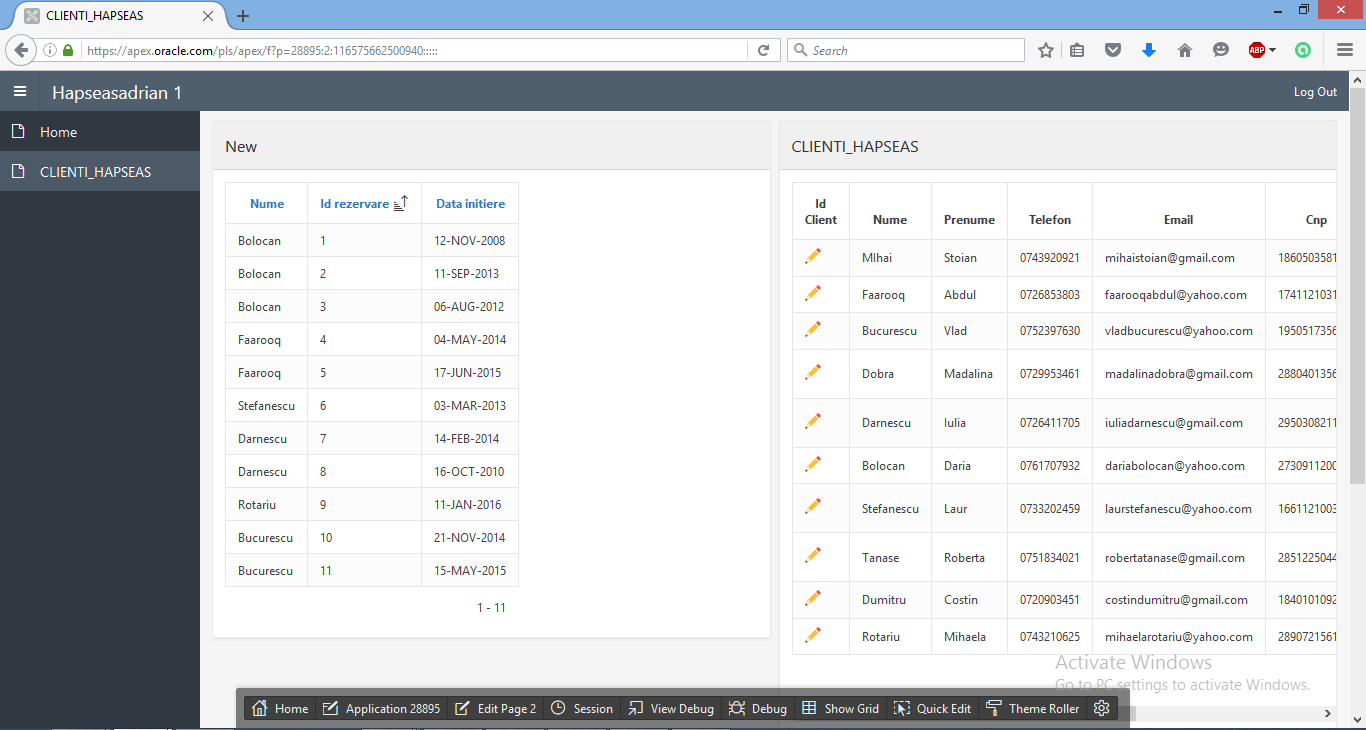
(select nume

from clienti\_hapseas c where

c.id\_client=r.id\_client) Nume

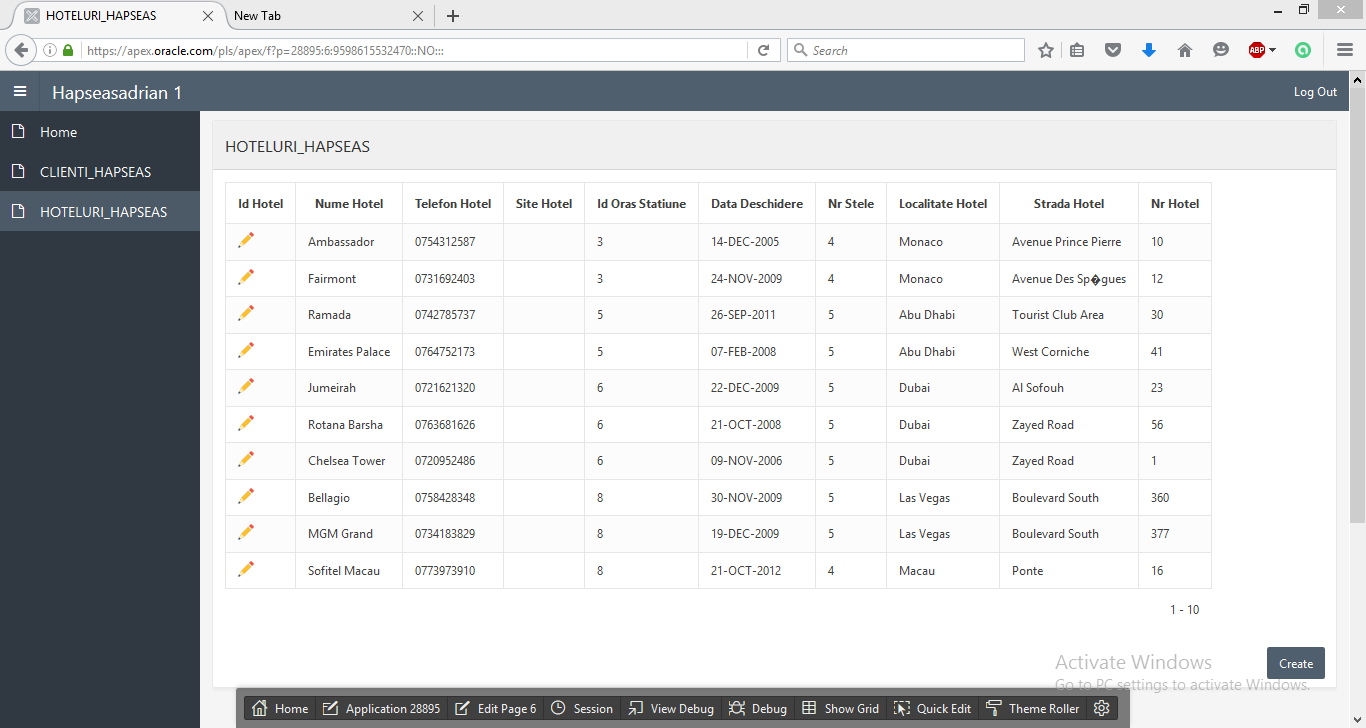
from rezervari\_hapseas r

<https://apex.oracle.com/pls/apex/f?p=28895:2:9598615532470::NO:::>



27. Construiti un videoformat(Hapseasadrian1), avand ca Master Table hoteluri\_hapseas, iar ca Detail Table comenzi\_aprovizionare\_hapseas.

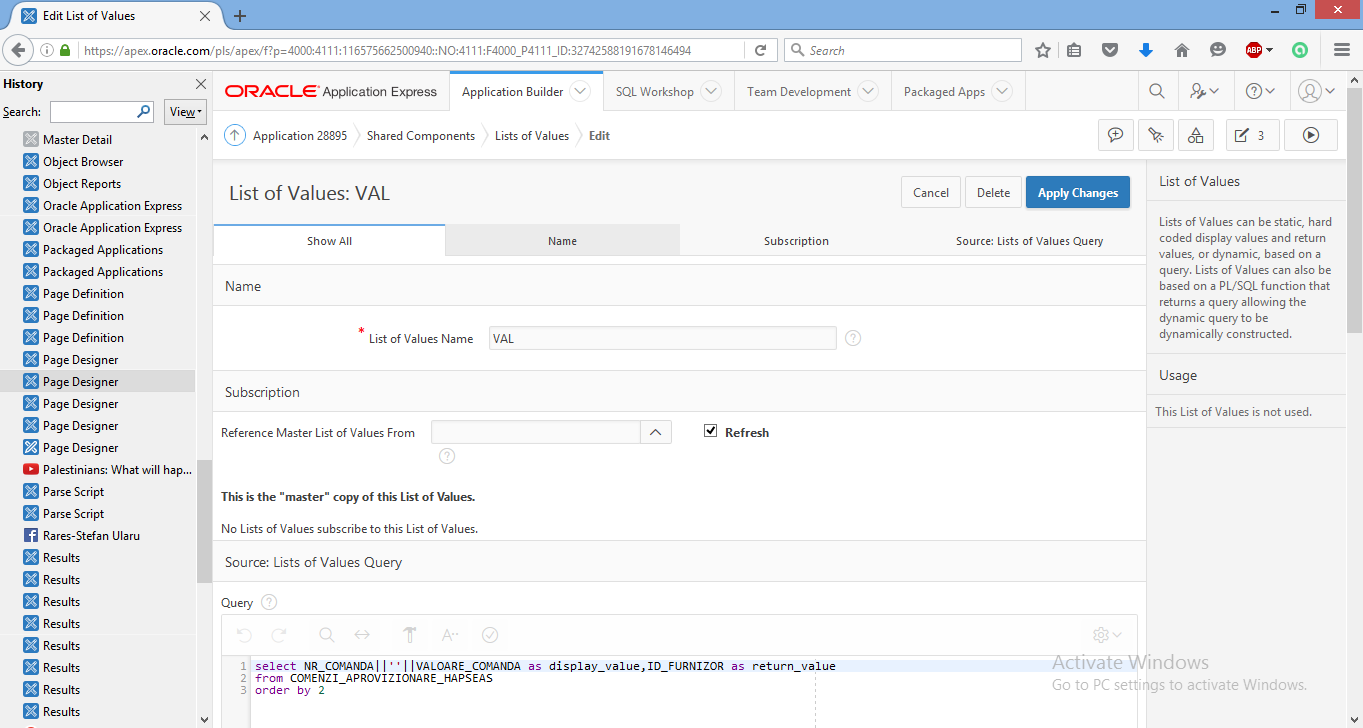
<https://apex.oracle.com/pls/apex/f?p=28895:6:9598615532470::NO:::>



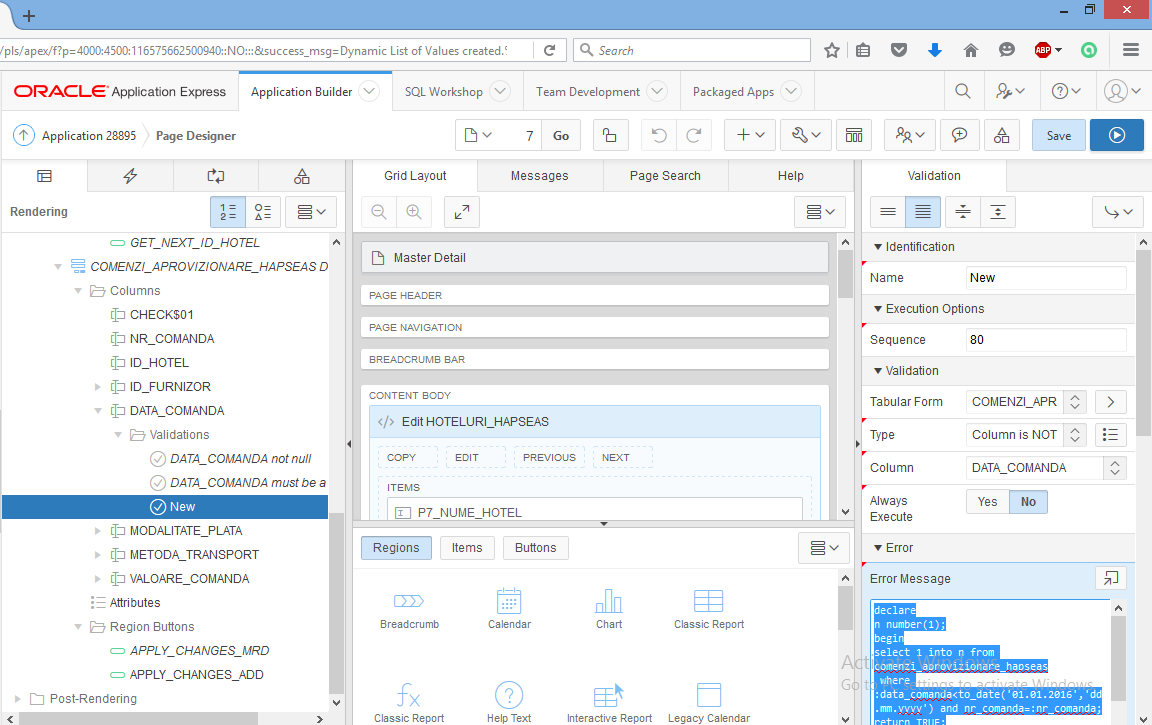
28.Adaugati o lista de valori cu denumirea Val

Select Nr\_comanda||' '||Valoare as display\_value, ID\_furnizor as return\_value

from comenzi\_aprovizionare\_hapseas order by 2



29.Creati o validare asupra coloanei data\_comanda.Data nu trebuie sa depaseasca ziua de 01.01.2016



<https://apex.oracle.com/pls/apex/f?p=4000:4500:116575662500940::NO:::>