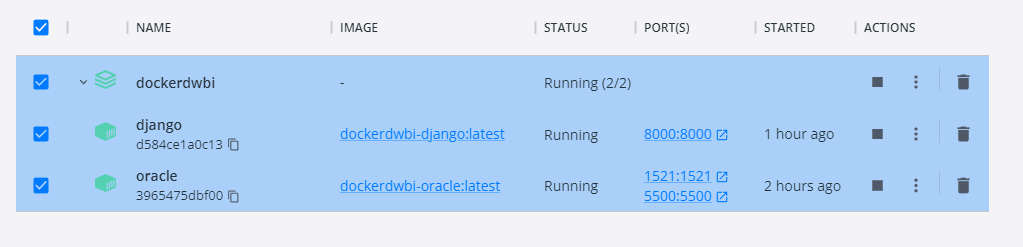
DATA WAREHOUSE & BUSINESS INTELLIGENCE

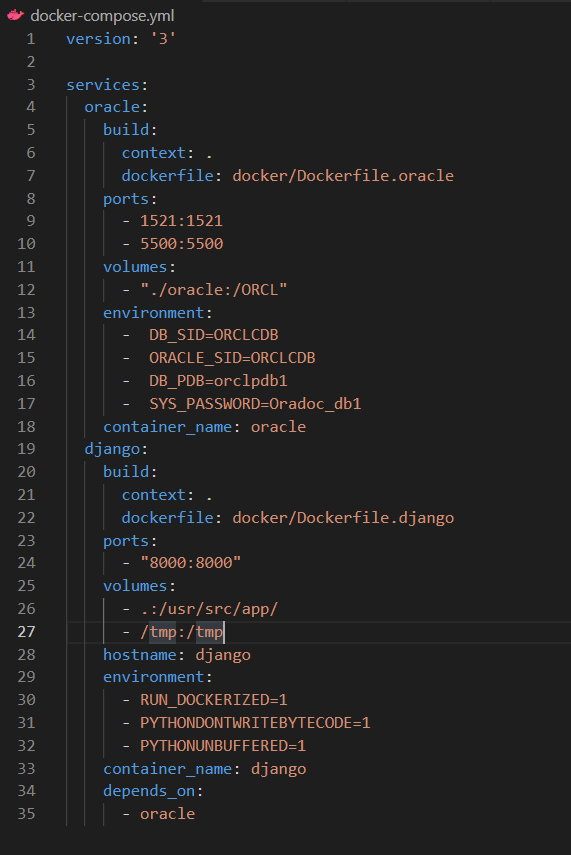
**Sistem de gestiune a rezervarilor hoteliere**

Pentru implementarea bazei de date am folosit ca sursa baza de date Oracle, versiunea 21c. Baza de date am creat-o direct intr-un container, folosind Docker. Pentru obtinerea imaginea bazei de date de tip oracle, am folosit site-ul oficial Oracle, <https://container-registry.oracle.com/> . Am decis sa folosim docker pentru a eficientiza procesul de instalare si configurare a bazei de date, care este utilizata de pe mai multe dispozitive.

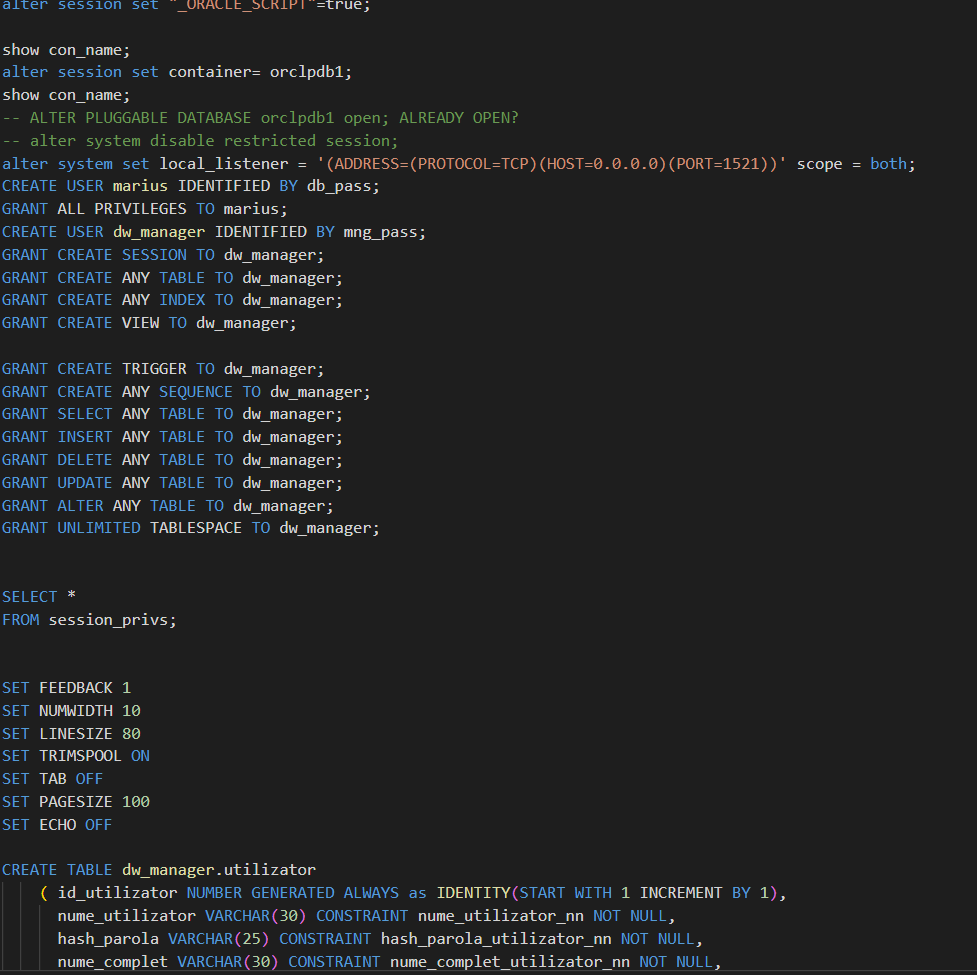
Pentru backend & frontend , am folosit Python, framework-ul Django, prin care sub forma unei aplicatii web putem altera si observa datele. Pentru managererea containerul, am folosit Docker Desktop, unde putem observa containerele ruland. La momentul crearii containerelor, sunt rulate scripturi si fisiere de configurare care creaza si populeaza bazele de date,respectiv tabele asociate OTLP is OLAP.



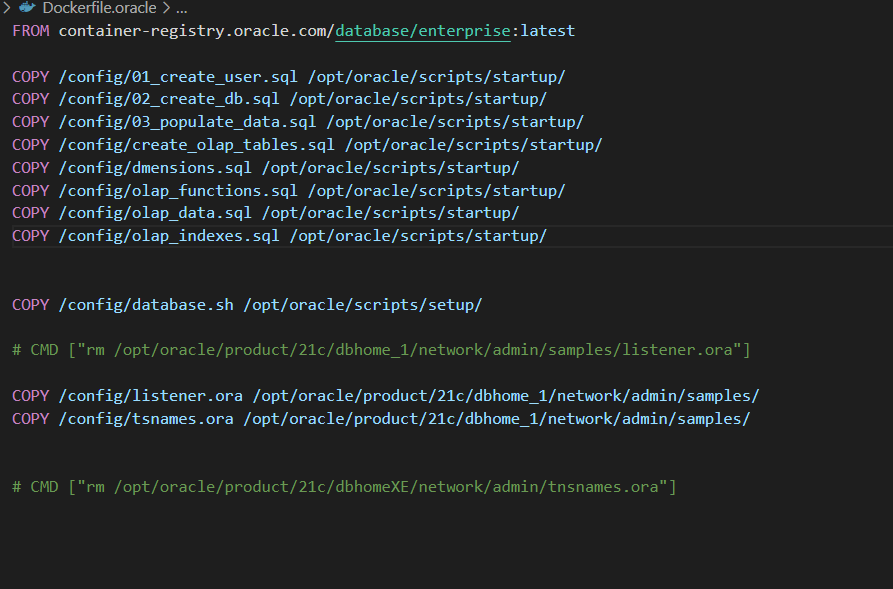
Pentru crearea containerelor, am folosit docker compose, unde avem cele 2 servicii definite:



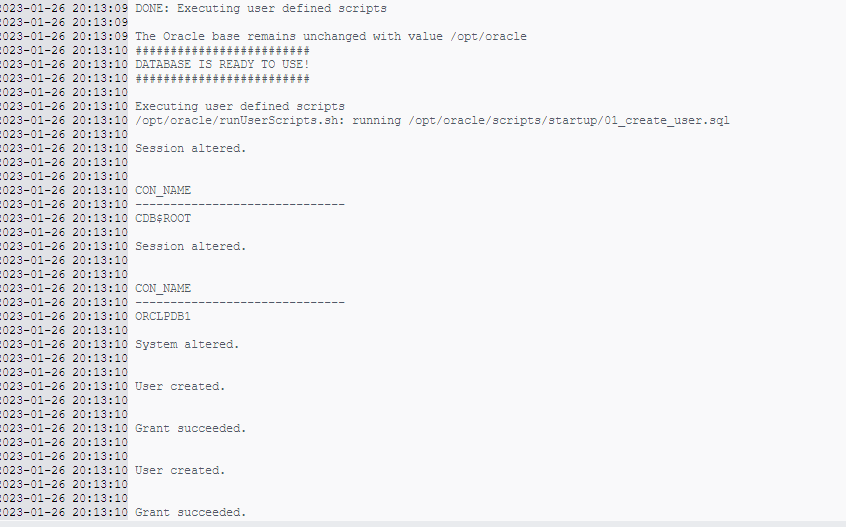
In script-urile de intializare, pe langa anumite conditii de conectivitate,vizibilitate a bazei de date intr-un sistem containerizat, incepem prin a crea userii, si tabelele OLTP.



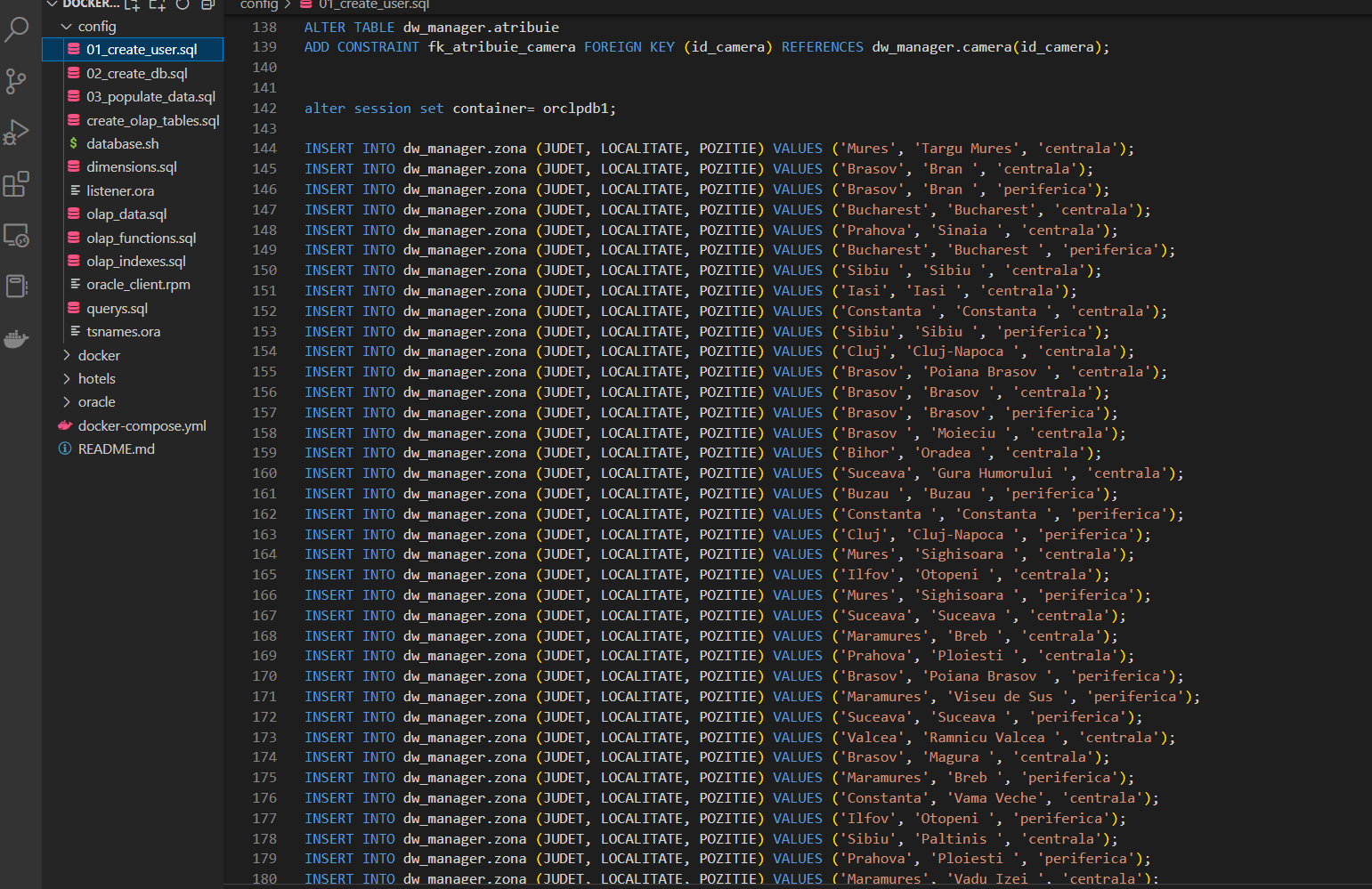
Script-urile de intializare, configurare sunt copiate in foldere predefinite de Oracle in scopul rularii acestora la startup.

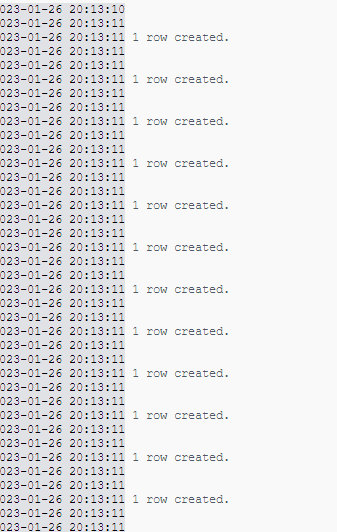


Se putea obtine acelasi rezultat folosind un volum, dar pentru vizibilitate, am copiat script-urile in folderele predefinite.



Dupa crearea tabelelor, si a userilor asociati, se insereaza datele, tot din cadrul fisierilor care ruleaza la startup:





Tabele pentru baza de date OLAP se populeaza din cadrul urmatorului script:

INSERT INTO perioada\_rezervare\_OLAP(zi\_din\_luna\_inceput, luna\_inceput, an\_inceput, zi\_din\_saptamana\_inceput, zi\_din\_an\_inceput, zi\_din\_luna\_sfarsit, luna\_sfarsit, an\_sfarsit, zi\_din\_saptamana\_sfarsit, zi\_din\_an\_sfarsit, durata\_in\_zile)

SELECT DISTINCT TO\_NUMBER(TO\_CHAR(data\_inceput,'DD')), TO\_CHAR(data\_inceput,'MON'), TO\_NUMBER(TO\_CHAR(data\_inceput,'YYYY')),

TO\_CHAR(data\_inceput,'DY'),TO\_NUMBER(TO\_CHAR(data\_inceput,'DDD')),

TO\_NUMBER(TO\_CHAR(data\_sfarsit,'DD')), TO\_CHAR(data\_sfarsit,'MON'), TO\_NUMBER(TO\_CHAR(data\_sfarsit,'YYYY')),

TO\_CHAR(data\_sfarsit,'DY'),TO\_NUMBER(TO\_CHAR(data\_sfarsit,'DDD')),

data\_sfarsit - data\_inceput

FROM rezervare; COMMIT;

INSERT INTO moment\_efectuare\_rezervare\_OLAP(zi\_din\_luna, luna, an, zi\_din\_saptamana, zi\_din\_an)

SELECT DISTINCT TO\_NUMBER(TO\_CHAR(data\_efectuarii,'DD')), TO\_CHAR(data\_efectuarii,'MON'),

TO\_NUMBER(TO\_CHAR(data\_efectuarii,'YYYY')),TO\_CHAR(data\_efectuarii,'DY'),TO\_NUMBER(TO\_CHAR(data\_efectuarii,'DDD'))--,

--TO\_NUMBER(TO\_CHAR(data\_efectuarii,'HH')) + CASE WHEN TO\_NUMBER(TO\_CHAR(data\_efectuarii,'MI')) > 29 THEN 1 ELSE 0 END AS ora\_aprox

FROM rezervare; COMMIT;

INSERT INTO hotel\_OLAP(id\_hotel, nume, regiune, judet, localitate, pozitie, nr\_stele, are\_mic\_dejun\_inclus)

SELECT id\_hotel, nume, regiune, judet, localitate, pozitie, nr\_stele, are\_mic\_dejun\_inclus

FROM hotel JOIN zona

USING(id\_zona); COMMIT;

INSERT INTO tip\_client\_OLAP(varsta,gen,stare\_civila)

SELECT DISTINCT FLOOR(MONTHS\_BETWEEN(SYSDATE,data\_nasterii)/12) AS varsta, gen, stare\_civila

FROM utilizator; COMMIT;

INSERT INTO tip\_camera\_OLAP(nr\_paturi\_duble, nr\_paturi\_simple,are\_terasa,are\_televizor)

SELECT DISTINCT nr\_paturi\_duble, nr\_paturi\_simple,are\_terasa,are\_televizor

FROM camera; COMMIT;

INSERT INTO rezervare\_camera\_OLAP(id\_rezervare,id\_hotel,id\_perioada,id\_moment\_efectuare,id\_tip\_camera,id\_tip\_client,pret)

SELECT DISTINCT id\_rezervare,id\_hotel,

                gaseste\_id\_perioada\_OLAP(TO\_NUMBER(TO\_CHAR(data\_inceput,'DDD')),TO\_NUMBER(TO\_CHAR(data\_inceput,'YYYY')),

                                        TO\_NUMBER(TO\_CHAR(data\_sfarsit,'DDD')),TO\_NUMBER(TO\_CHAR(data\_sfarsit,'YYYY'))) AS id\_perioada,

                gaseste\_id\_moment\_efectuare\_OLAP(TO\_NUMBER(TO\_CHAR(data\_efectuarii,'DDD')),TO\_NUMBER(TO\_CHAR(data\_efectuarii,'YYYY'))) AS id\_moment\_efectuare,

                gaseste\_id\_tip\_camera\_OLAP(camera.nr\_paturi\_duble,camera.nr\_paturi\_simple,camera.are\_terasa,camera.are\_televizor) AS id\_tip\_camera,

                gaseste\_id\_tip\_client\_OLAP(calculeaza\_varsta(utilizator.data\_nasterii),utilizator.gen,utilizator.stare\_civila) AS id\_tip\_client,

                camera.pret\_per\_noapte \* (rezervare.data\_sfarsit - rezervare.data\_inceput) AS pret

FROM utilizator JOIN rezervare

ON utilizator.id\_utilizator=rezervare.id\_client

JOIN rezervare\_camera

USING(id\_rezervare)

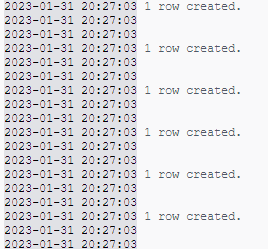
JOIN camera

USING(id\_camera)

JOIN hotel

USING(id\_hotel); COMMIT;

Dupa rularea script-ului, observam in docker aceleasi log-uri:



**Crearea rapoartelor cu complexitate diferită**

1\*  - Cea mai scumpa camera per noapte  din romania

select max(pret\_per\_noapte) from

(select hotel.nume,camera.pret\_per\_noapte from hotel

INNER JOIN camera on camera.id\_hotel = hotel.id\_hotel

order by hotel.nume)

2\* Cea mai ieftina camera dintr-o zona centrala

select \* from (select hotel.nume, MIN(pret\_per\_noapte) from camera

INNER JOIN hotel

on hotel.id\_hotel=camera.id\_hotel

INNER JOIN zona on zona.id\_zona = hotel.id\_zona

WHERE zona.pozitie = 'centrala'

GROUP BY hotel.nume)

where rownum = 1

3\* Hotelul cu cele mai multe rezervari efectuate

select \* from(

SELECT MAX(rezervari) as mres, numes  FROM (SELECT  hotel.nume as numes,COUNT(atribuie.id\_rezervare) as rezervari from atribuie

inner join camera on camera.id\_camera=atribuie.id\_camera

inner join hotel on hotel.id\_hotel = camera.id\_hotel

GROUP BY hotel.nume)

group by numes

ORDER BY mres DESC)

where rownum=1;

4\* Cele mai scumpe 5 hoteluri

select DISTINCT hotel.nume,camera.pret\_per\_noapte from hotel

INNER JOIN camera

on camera.id\_hotel = hotel.id\_hotel

order by camera.pret\_per\_noapte desc

fetch first 5 rows only;

5\* Rezezrvari in functie de luna anului

SELECT COUNT(\*) FROM REZERVARE

WHERE EXTRACT(MONTH FROM data\_inceput) IN (1,4)

SELECT COUNT(\*) FROM REZERVARE

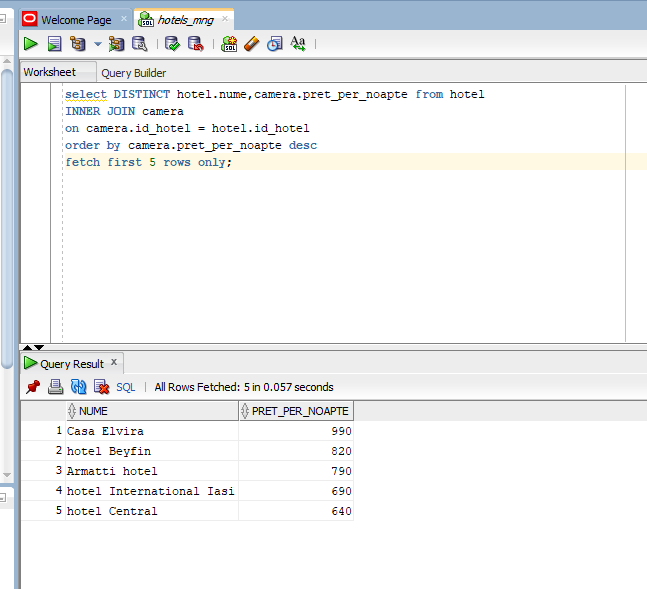
WHERE EXTRACT(MONTH FROM data\_inceput) IN (4, 7)

SELECT COUNT(\*) FROM REZERVARE

WHERE EXTRACT(MONTH FROM data\_inceput) IN (7, 10)

SELECT COUNT(\*) FROM REZERVARE

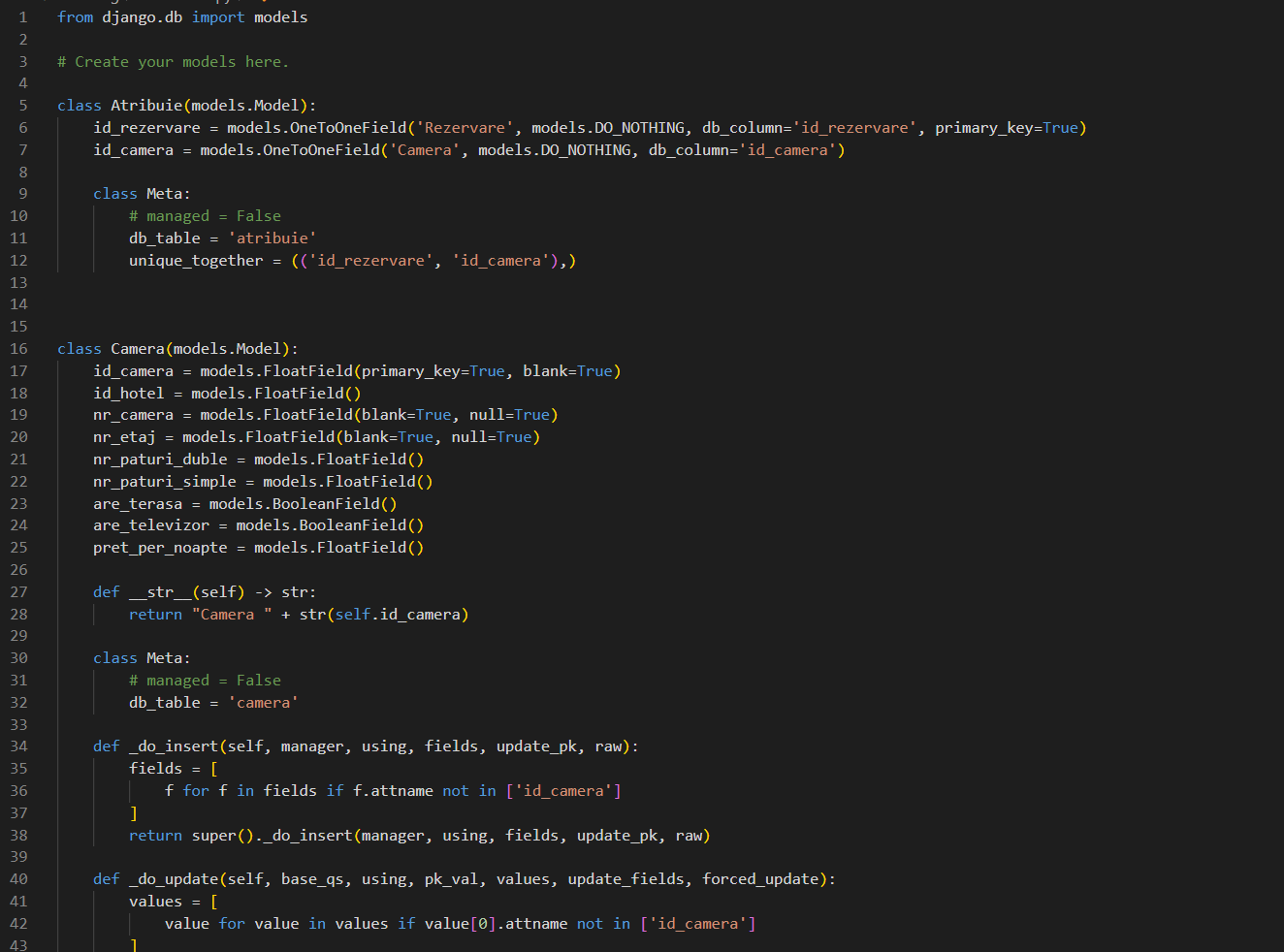
WHERE EXTRACT(MONTH FROM data\_inceput) IN (10, 1)



Pentru aceste rapoarte au fost create grafice de tip chart in front-end aplicatiei Web.

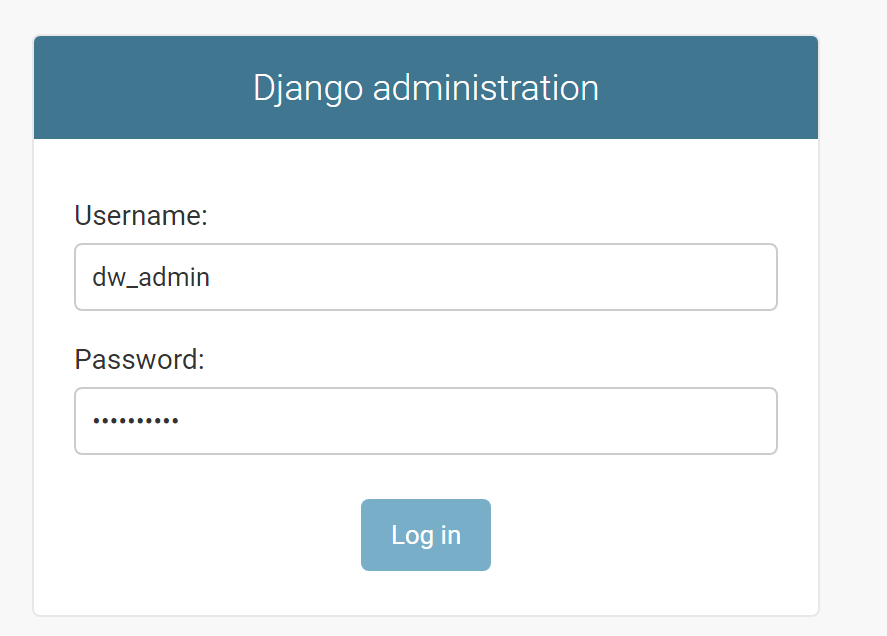
# Aplicatie – Frontend,

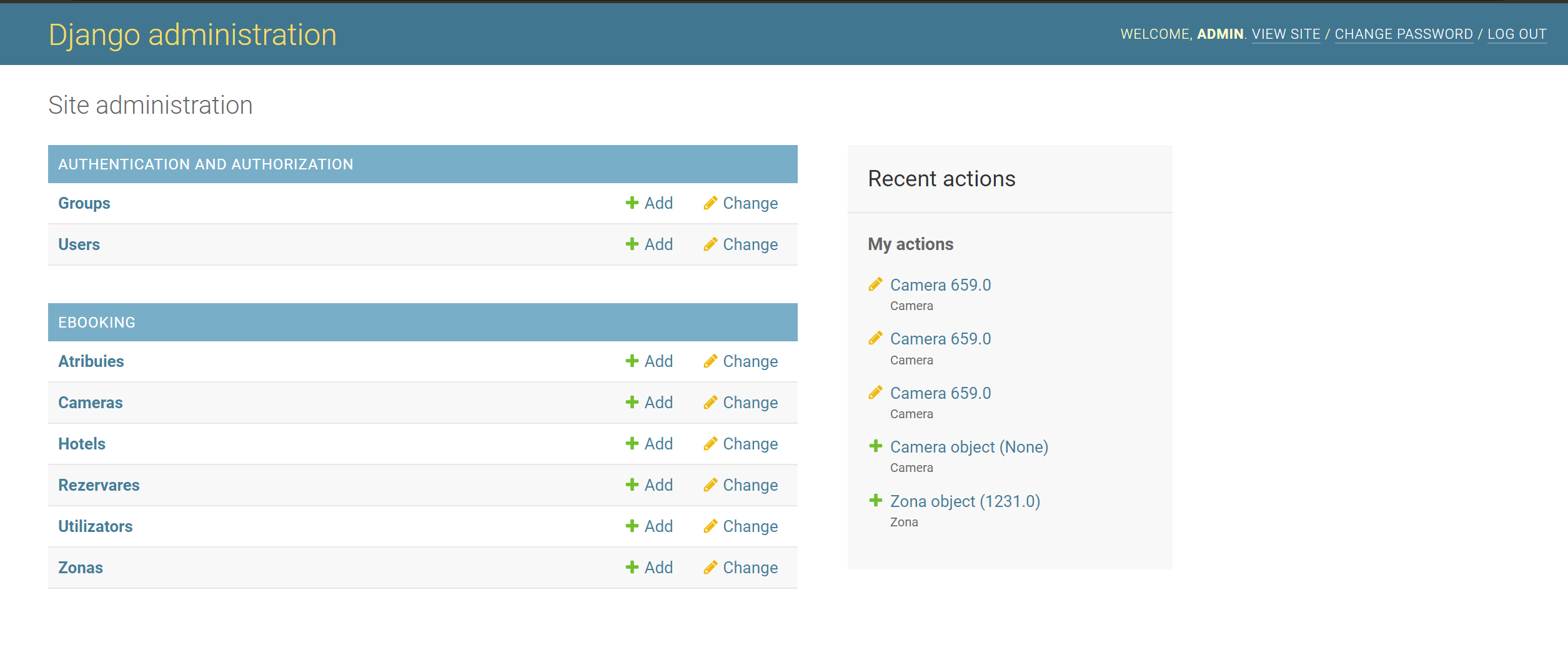
(Back end) Datele de tip OLTP au fost proiectate in back end, sub forma unor modele, folosind ORM-ul django. Comanda *python manage.py inspectdb* genereaza aceste tabele, care trebuie putin customizate pentru a functiona in diferite scenarii.

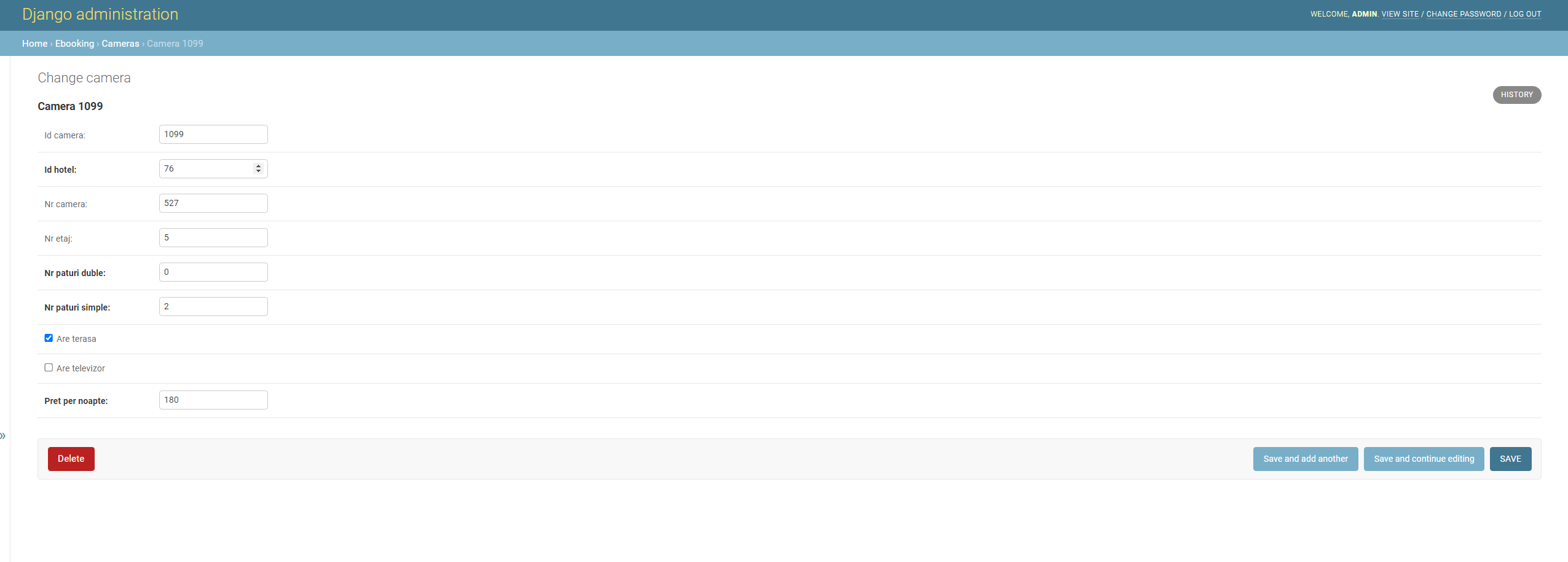


Aceste modele sunt vizibile in interfata web a aplicatiei django, printr-o pagina de management alocata administratorului bazei de date.

Dupa introducerea datelor de logare definite in fisierele de config ale oracle, admin-ul va putea vedea toate datele de tip OLTP din cadrul bazei de date .

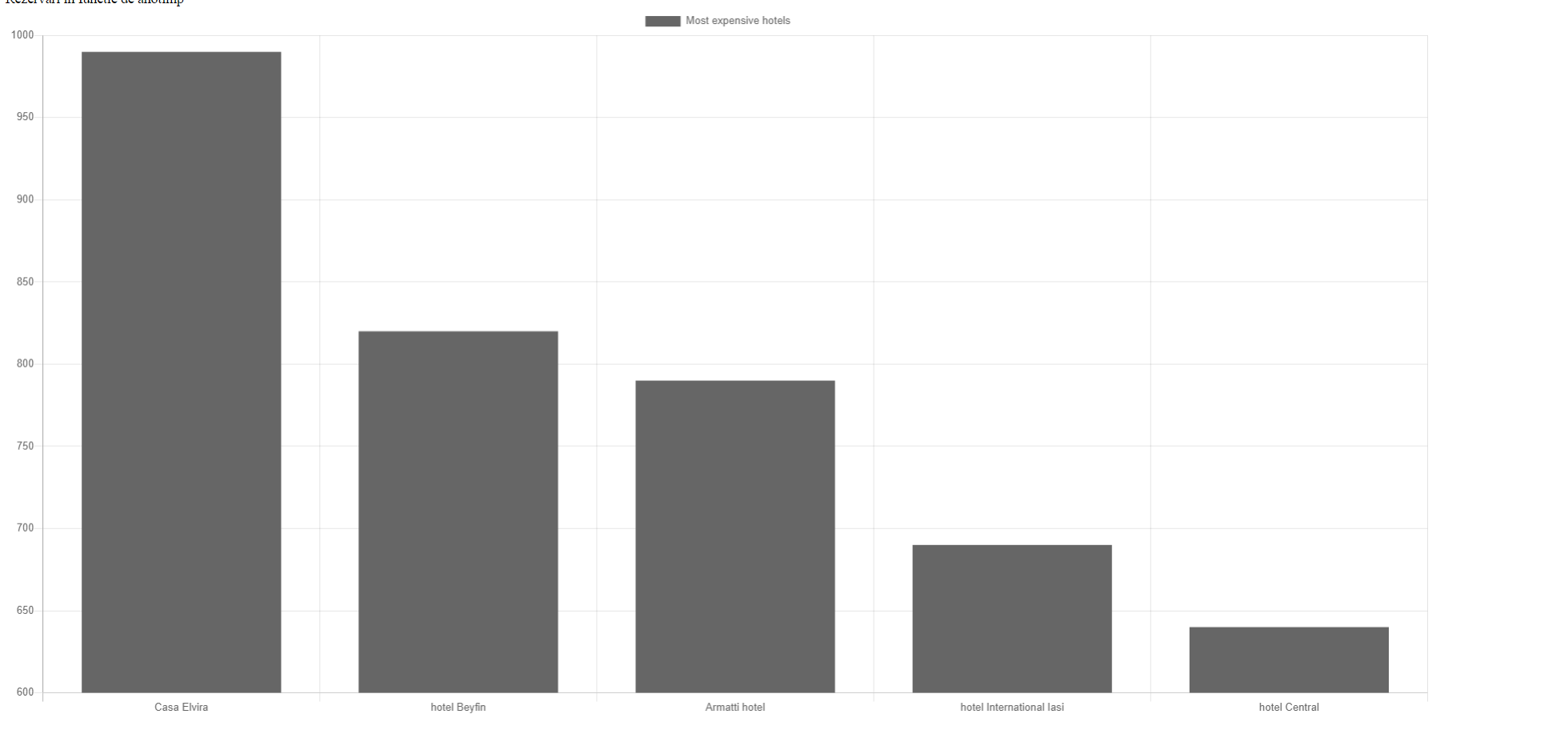






Pe pagina principala, <http://localhost:8000/>, ne sunt prezentate graficele corespunzatoare script-urilor sql create anterior.

Cele mai scumpe hoteluri 5 din Romania



Cereri in functie de perioada anului

