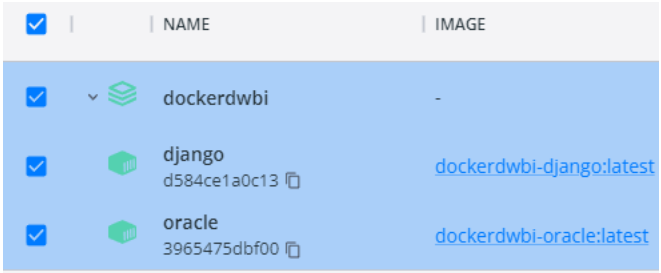
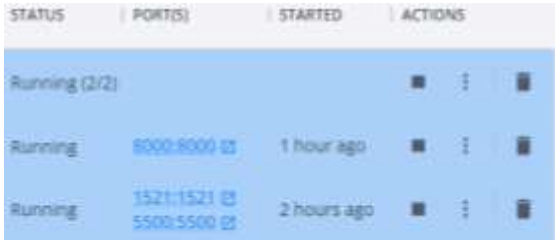
DATA WAREHOUSE & BUSINESS INTELLIGENCE

Sistem de gestiune a rezervarilor hoteliere

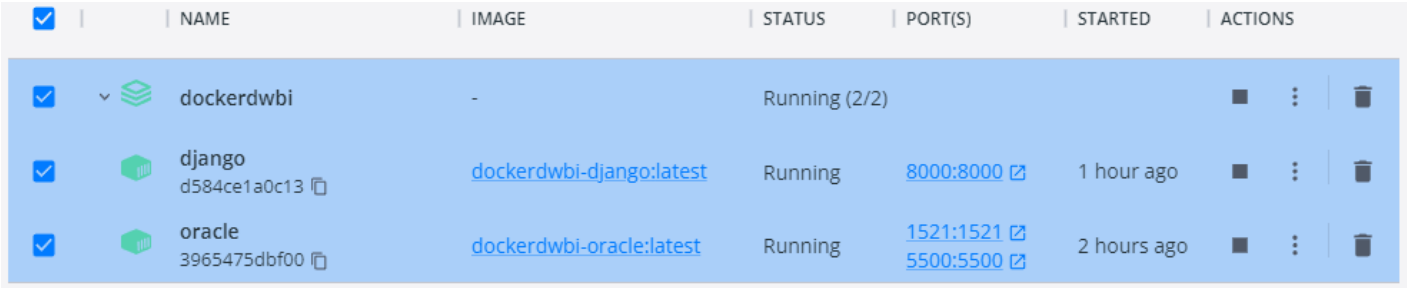
Pentru implementarea bazei de date am folosit ca sursa baza de date Oracle, versiunea 21c. Baza de date am creat-o direct intr-un container, folosind Docker. Pentru obtinerea imaginea bazei de date de tip oracle, am folosit site-ul oficial Oracle, https://container-registry.oracle.com/ . Am decis sa 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:



docker-compose.yml

```yaml
version: '3'

services:
  oracle:
    build:
      context: .
      dockerfile: docker/Dockerfile.oracle
    ports:
      - 1521:1521
      - 5500:5500
    volumes:
      - "./oracle:/ORCL"
    environment:
      -  DB_SID=ORCLCDB
      -  ORACLE_SID=ORCLCDB
      -  DB_PDB=orclpdb1
      -  SYS_PASSWORD=Oradoc_db1
    container_name: oracle
  django:
    build:
      context: .
      dockerfile: docker/Dockerfile.django
    ports:
      - "8000:8000"
    volumes:
      - .:/usr/src/app/
      - /tmp:/tmp
    hostname: django
    environment:
      - RUN_DOCKERIZED=1
      - PYTHONDONTWRITEBYTECODE=1
      - PYTHONUNBUFFERED=1
    container_name: django
    depends_on:
      - oracle
```

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.

```oracle
alter session set "_ORACLE_SCRIPT"=true;

show con_name;
alter session set container= orclpdb1;
show con_name;
-- ALTER PLUGGABLE DATABASE orclpdb1 open; ALREADY OPEN?
-- alter system disable restricted session;
alter system set local_listener = '(ADDRESS=(PROTOCOL=TCP)(HOST=0.0.0.0)(PORT=1521))' scope = both;
CREATE USER marius IDENTIFIED BY db_pass;
GRANT ALL PRIVILEGES TO marius;
CREATE USER dw_manager IDENTIFIED BY mng_pass;
GRANT CREATE SESSION TO dw_manager;
GRANT CREATE ANY TABLE TO dw_manager;
GRANT CREATE ANY INDEX TO dw_manager;
GRANT CREATE VIEW TO dw_manager;

GRANT CREATE TRIGGER TO dw_manager;
GRANT CREATE ANY SEQUENCE TO dw_manager;
GRANT SELECT ANY TABLE TO dw_manager;
GRANT INSERT ANY TABLE TO dw_manager;
GRANT DELETE ANY TABLE TO dw_manager;
GRANT UPDATE ANY TABLE TO dw_manager;
GRANT ALTER ANY TABLE TO dw_manager;
GRANT UNLIMITED TABLESPACE TO dw_manager;
```

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

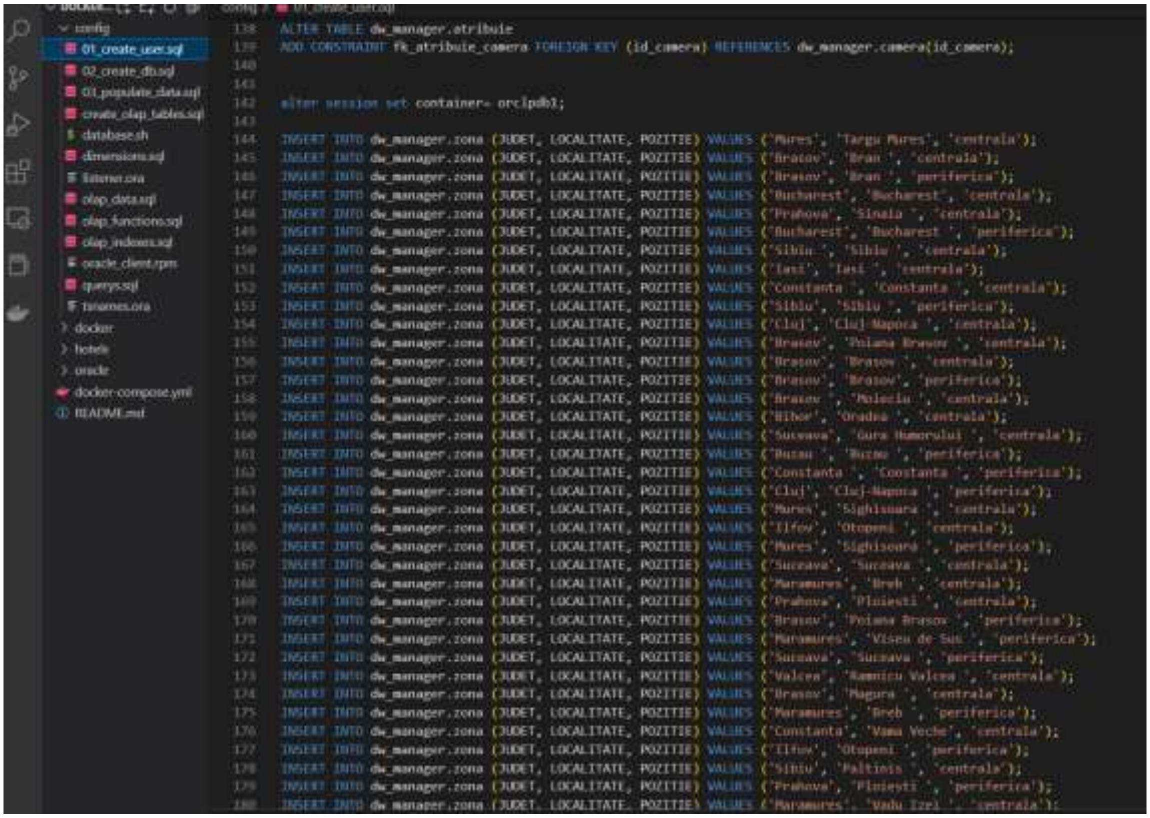
```oracle
SELECT *
FROM session_privs;


SET FEEDBACK 1
SET NUMWIDTH 10
SET LINESIZE 80
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100
SET ECHO OFF

CREATE TABLE dw_manager.utilizator
    ( id_utilizator NUMBER GENERATED ALWAYS as IDENTITY(START WITH 1 INCREMENT BY 1),
      nume_utilizator VARCHAR(30) CONSTRAINT nume_utilizator_nn NOT NULL,
      hash_parola VARCHAR(25) CONSTRAINT hash_parola_utilizator_nn NOT NULL,
      nume_complet VARCHAR(30) CONSTRAINT nume_complet_utilizator_nn NOT NULL,
```

```
> Dockerfile.oracle > ...
FROM container-registry.oracle.com/database/enterprise:latest

COPY /config/01_create_user.sql /opt/oracle/scripts/startup/
COPY /config/02_create_db.sql /opt/oracle/scripts/startup/
COPY /config/03_populate_data.sql /opt/oracle/scripts/startup/
COPY /config/create_olap_tables.sql /opt/oracle/scripts/startup/
COPY /config/dmensions.sql /opt/oracle/scripts/startup/
COPY /config/olap_functions.sql /opt/oracle/scripts/startup/
COPY /config/olap_data.sql /opt/oracle/scripts/startup/
COPY /config/olap_indexes.sql /opt/oracle/scripts/startup/


COPY /config/database.sh /opt/oracle/scripts/setup/

# CMD ["rm /opt/oracle/product/21c/dbhome_1/network/admin/samples/listener.ora"]

COPY /config/listener.ora /opt/oracle/product/21c/dbhome_1/network/admin/samples/
COPY /config/tsnames.ora /opt/oracle/product/21c/dbhome_1/network/admin/samples/


# CMD ["rm /opt/oracle/product/21c/dbhomeXE/network/admin/tnsnames.ora"]
```

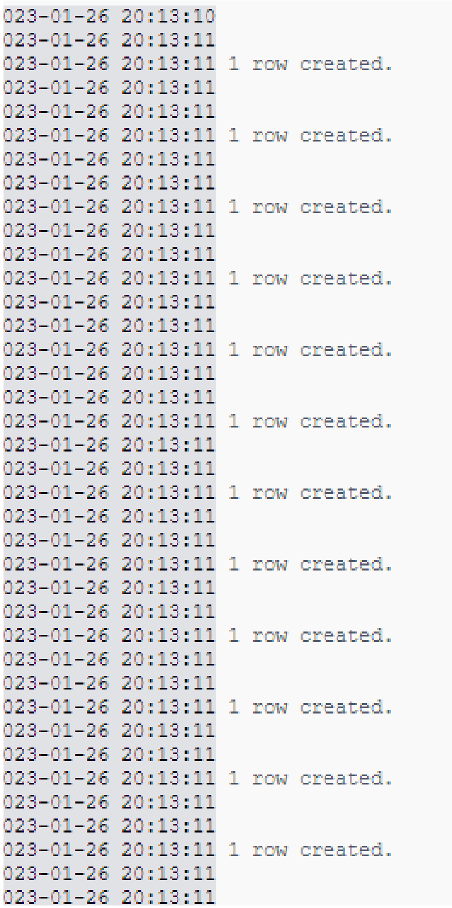
Se putea obtine acelasi rezultat folosind un volum, dar pentru vizibilitate, am copiat script-urile in folderele predefinite.

```
023-01-26 20:13:09 DONE: Executing user defined scripts
023-01-26 20:13:09
023-01-26 20:13:09 The Oracle base remains unchanged with value /opt/oracle
023-01-26 20:13:10 #########################
023-01-26 20:13:10 DATABASE IS READY TO USE!
023-01-26 20:13:10 #########################
023-01-26 20:13:10
023-01-26 20:13:10 Executing user defined scripts
023-01-26 20:13:10 /opt/oracle/runUserScripts.sh: running /opt/oracle/scripts/startup/01_create_user.sql
023-01-26 20:13:10
023-01-26 20:13:10 Session altered.
023-01-26 20:13:10
023-01-26 20:13:10
023-01-26 20:13:10 CON_NAME
023-01-26 20:13:10 ------------------------------
023-01-26 20:13:10 CDB$ROOT
023-01-26 20:13:10
023-01-26 20:13:10 Session altered.
023-01-26 20:13:10
023-01-26 20:13:10
023-01-26 20:13:10 CON_NAME
023-01-26 20:13:10 ------------------------------
023-01-26 20:13:10 ORCLPDB1
023-01-26 20:13:10
023-01-26 20:13:10 System altered.
023-01-26 20:13:10
023-01-26 20:13:10
023-01-26 20:13:10 User created.
023-01-26 20:13:10
023-01-26 20:13:10
023-01-26 20:13:10 Grant succeeded.
023-01-26 20:13:10
023-01-26 20:13:10
023-01-26 20:13:10 User created.
023-01-26 20:13:10
023-01-26 20:13:10
023-01-26 20:13:10 Grant succeeded.
```

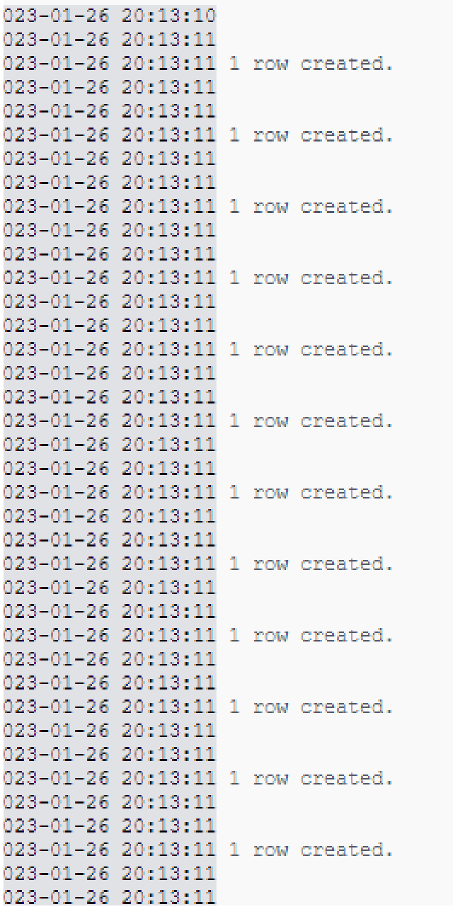
Dupa crearea tabelelor, si a userilor asociati, se insereaza datele, tot din cadrul fisierilor care ruleaza la startup:



```sql
ALTER TABLE dw_manager.atribuie
ADD CONSTRAINT fk_atribuie_camera FOREIGN KEY (id_camera) REFERENCES dw_manager.camera(id_camera);


alter session set container= orclpdb1;

INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Mures', 'Targu Mures', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Bran ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Bran ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Bucharest', 'Bucharest', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Prahova', 'Sinaia ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Bucharest', 'Bucharest ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Sibiu ', 'Sibiu ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Iasi', 'Iasi ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Constanta ', 'Constanta ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Sibiu', 'Sibiu ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Cluj', 'Cluj-Napoca ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Poiana Brasov ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Brasov ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Brasov', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov ', 'Moieciu ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Bihor', 'Oradea ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Suceava', 'Gura Humorului ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Buzau ', 'Buzau ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Constanta ', 'Constanta ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Cluj', 'Cluj-Napoca ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Mures', 'Sighisoara ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Ilfov', 'Otopeni ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Mures', 'Sighisoara ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Suceava', 'Suceava ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Maramures', 'Breb ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Prahova', 'Ploiesti ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Poiana Brasov ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Maramures', 'Viseu de Sus ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Suceava', 'Suceava ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Valcea', 'Ramnicu Valcea ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Brasov', 'Magura ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Maramures', 'Breb ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Constanta', 'Vama Veche', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Ilfov', 'Otopeni ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Sibiu', 'Paltinis ', 'centrala');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Prahova', 'Ploiesti ', 'periferica');
INSERT INTO dw_manager.zona (JUDET, LOCALITATE, POZITIE) VALUES ('Maramures', 'Vadu Izei ', 'centrala');
```



```
023-01-26 20:13:10
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
```



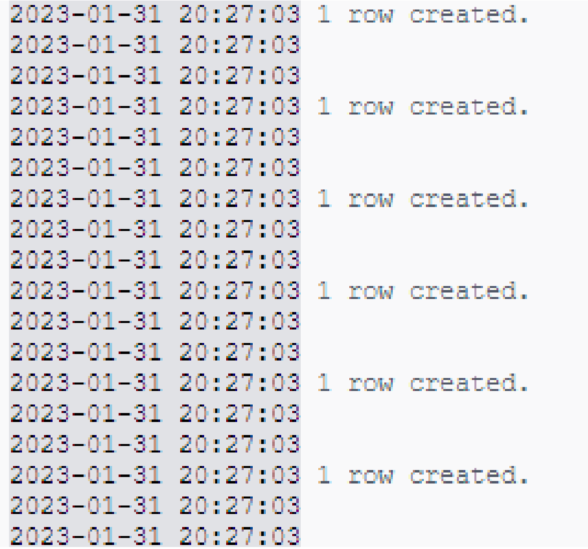
```
023-01-26 20:13:10
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
023-01-26 20:13:11 1 row created.
023-01-26 20:13:11
023-01-26 20:13:11
```

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

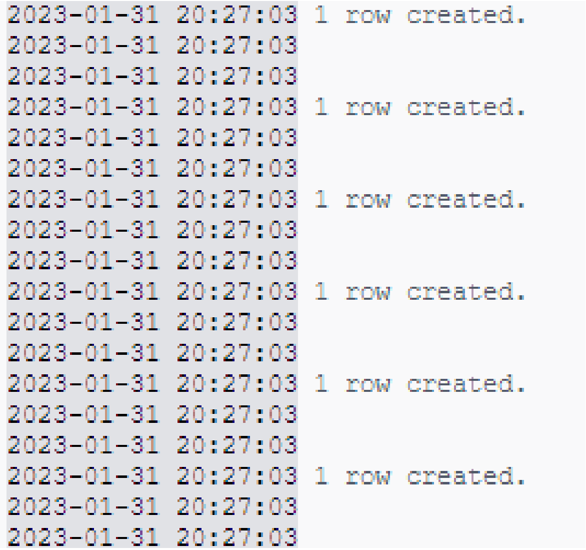
```
INSERT INTO perioada_rezervare_OLAP(zi_din_luna_inceput, luna_inceput, an_inceput,
zi_din_saptamana_inceput, zi_din_an_inceput, zi_din_luna_sfarsit, luna_sfarsit,
an_sfarsit, zi_din_saptamana_sfarsit, zi_din_an_sfarsit, durata_in_zile)
SELECT DISTINCT TO_NUMBER(TO_CHAR(data_inceput,'DD')), TO_CHAR(data_inceput,'MON'),
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_CHA
R(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;
```

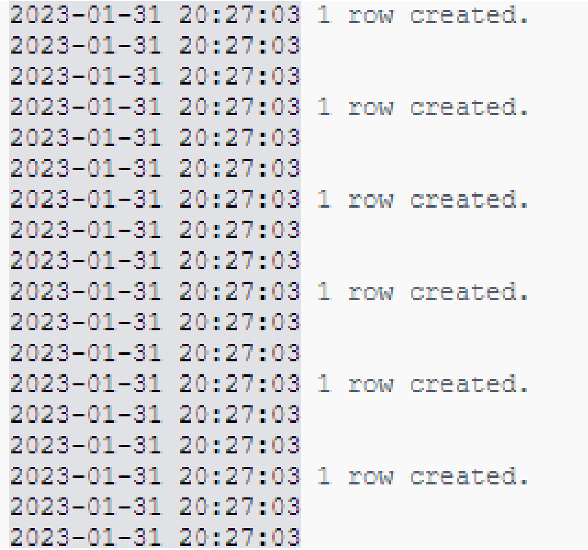
Dupa rularea script-ului, observam in docker aceleasi log-uri:



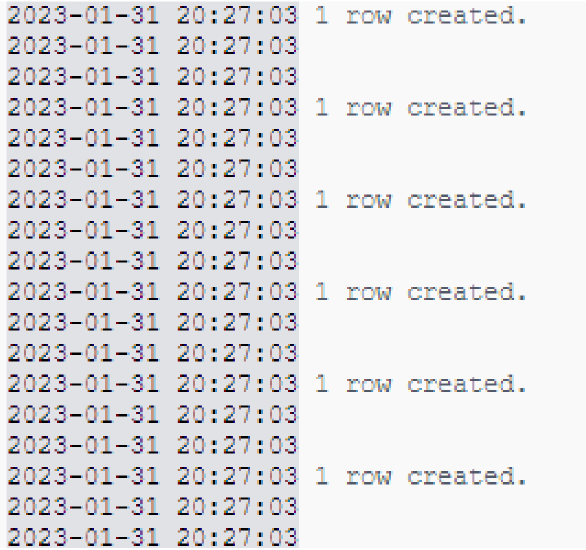
```
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
```



```
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
```



```
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
```



```
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
2023-01-31 20:27:03 1 row created.
2023-01-31 20:27:03
2023-01-31 20:27:03
```

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.

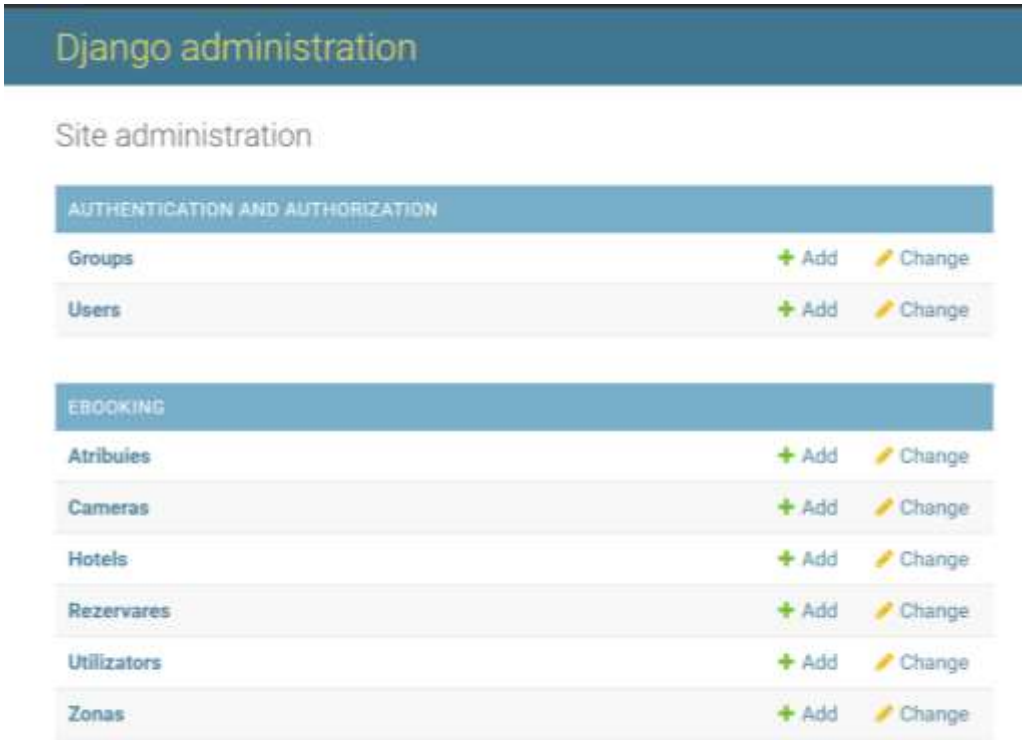
```
from django.db import models

# Create your models here.

class Atribuie(models.Model):
    id_rezervare = models.OneToOneField('Rezervare', models.DO_NOTHING, db_column='id_rezervare', primary_key=True)
    id_camera = models.OneToOneField('Camera', models.DO_NOTHING, db_column='id_camera')

    class Meta:
        # managed = False
        db_table = 'atribuie'
        unique_together = (('id_rezervare', 'id_camera'),)
```

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

```
class Camera(models.Model):
    id_camera = models.FloatField(primary_key=True, blank=True)
    id_hotel = models.FloatField()
    nr_camera = models.FloatField(blank=True, null=True)
    nr_etaj = models.FloatField(blank=True, null=True)
    nr_paturi_duble = models.FloatField()
    nr_paturi_simple = models.FloatField()
    are_terasa = models.BooleanField()
    are_televizor = models.BooleanField()
    pret_per_noapte = models.FloatField()

    def __str__(self) -> str:
        return "Camera " + str(self.id_camera)

    class Meta:
        # managed = False
        db_table = 'camera'
```

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 .

```python
    def _do_insert(self, manager, using, fields, update_pk, raw):
        fields = [
            f for f in fields if f.attname not in ['id_camera']
        ]
        return super()._do_insert(manager, using, fields, update_pk, raw)

    def _do_update(self, base_qs, using, pk_val, values, update_fields, forced_update):
        values = [
            value for value in values if value[0].attname not in ['id_camera']
        ]
```



WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Recent actions

My actions

- Camera 659.0
  Camera
- Camera 659.0
  Camera
- Camera 659.0
  Camera
- + Camera object (None)
  Camera
- + Zona object (1231.0)
  Zona

Django administration

Home › Ebooking › Cameras › Camera 1099

Change camera

Camera 1099

Id camera:	1099
Id hotel:	76
Nr camera:	527
Nr etaj:	5
Nr paturi duble:	0
Nr paturi simple:	2

- [x] Are terasa
- [ ] Are televizor

Pret per noapte:	180

Delete

Save and add another | Save and continue editing | SAVE

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

Crearea rapoartelor cu complexitate diferită

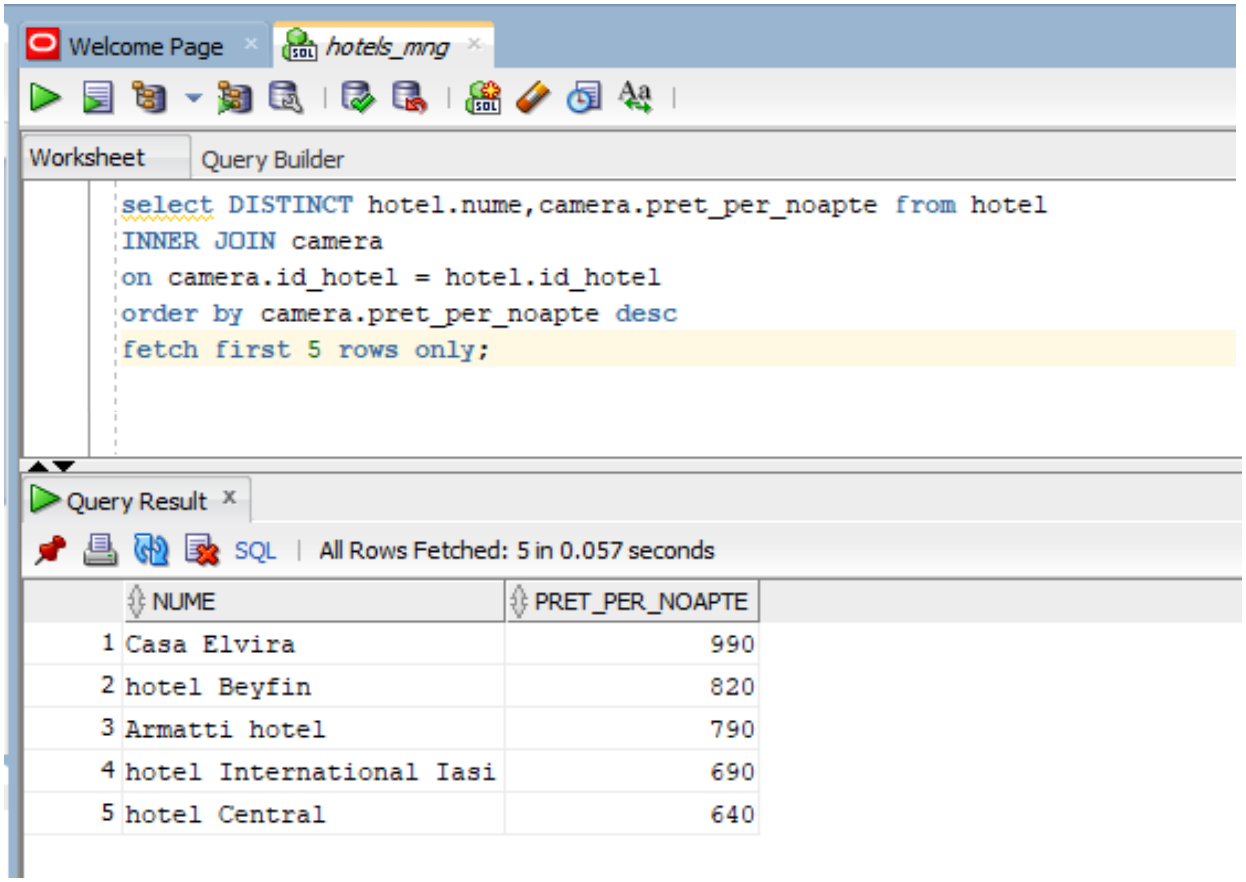
Pentru aceste rapoarte au fost create grafice de tip chart in front-end aplicatiei Web.

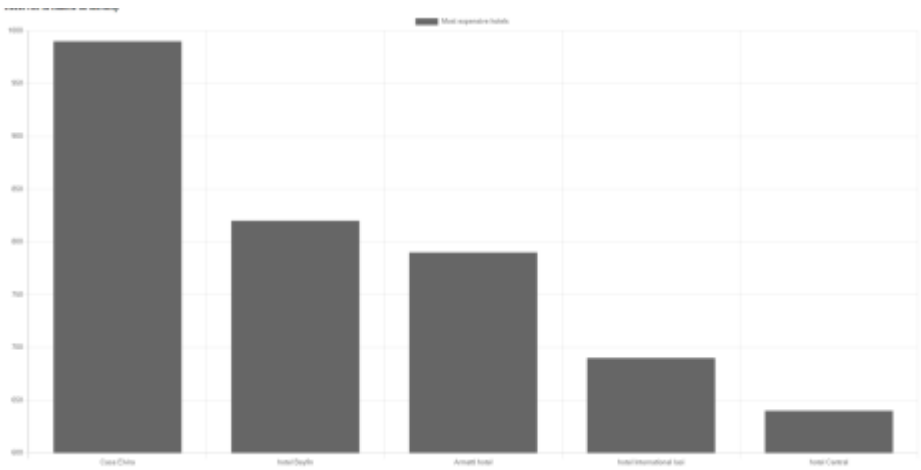
```
1*  - Cea mai scumpa camera per noapte  din romania

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
```

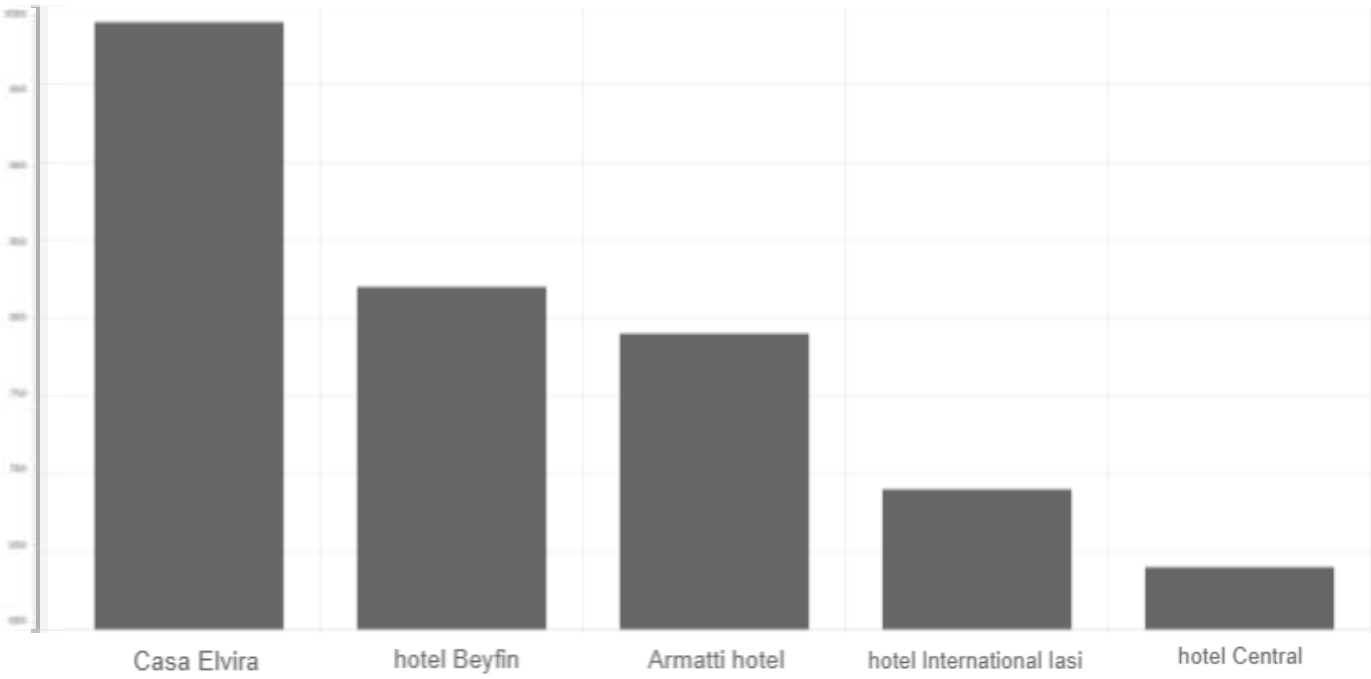
```
3* Hotelul cu cele mai multe rezervari efectuate

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;
```





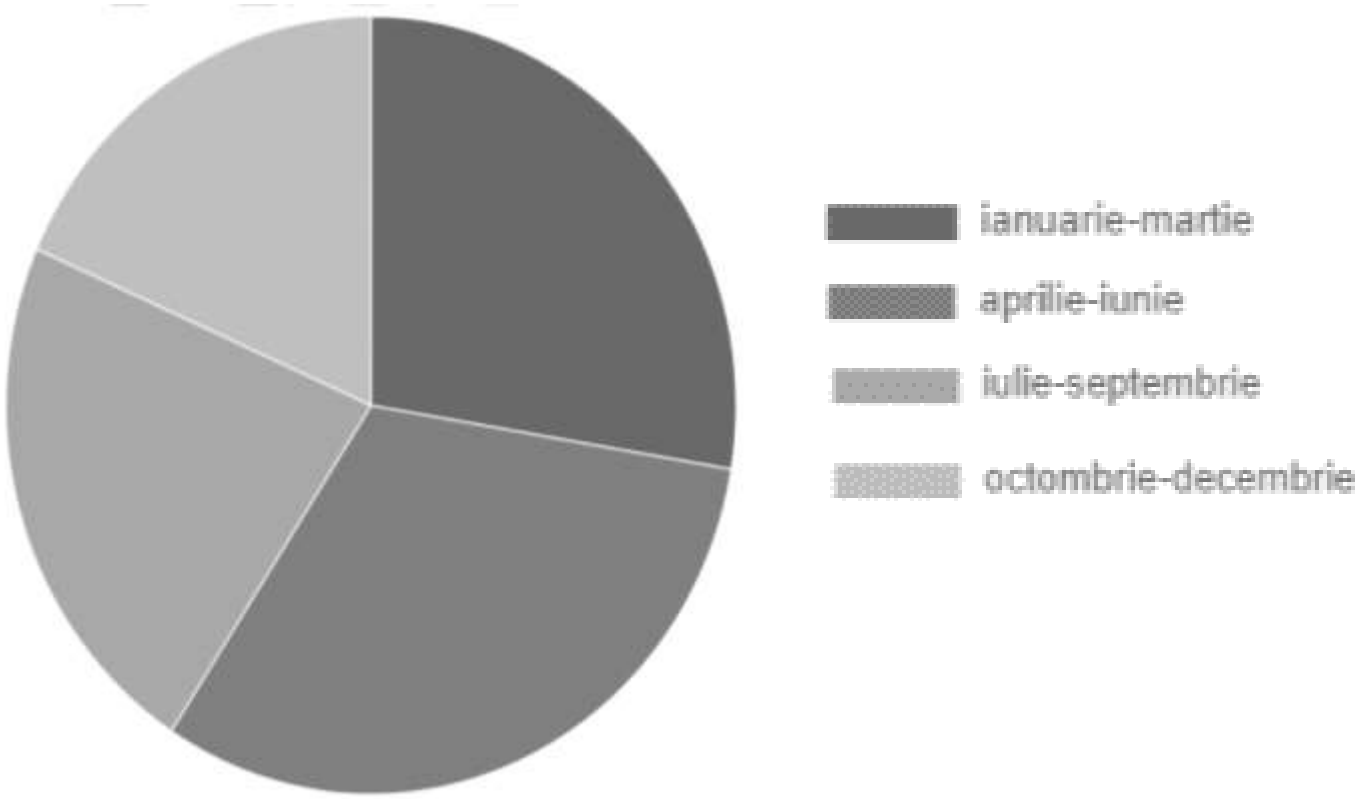
Cele mai scumpe hoteluri 5 din Romania




```
5* Rezervari 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)
```

```
bar_data = []

with connection.cursor() as cursor:
    cursor.execute("SELECT COUNT(*) FROM REZERVARE WHERE EXTRACT(MONTH FROM data_inceput) IN (1,4)")
    first_3_months = cursor.fetchone()
    cursor.execute("SELECT COUNT(*) FROM REZERVARE WHERE EXTRACT(MONTH FROM data_inceput) IN (4,7)")
    first_6_months = cursor.fetchone()
    cursor.execute("SELECT COUNT(*) FROM REZERVARE WHERE EXTRACT(MONTH FROM data_inceput) IN (7,10)")
    first_9_months = cursor.fetchone()
    cursor.execute("SELECT COUNT(*) FROM REZERVARE WHERE EXTRACT(MONTH FROM data_inceput) IN (10,1)")
    first_12_months = cursor.fetchone()
```

```
labels.extend(['ianuarie-martie','aprilie-iunie', 'iulie-septembrie','octombrie-decembrie'])
data.extend([first_3_months[0], first_6_months[0], first_9_months[0], first_12_months[0]])

print("LABELS:", labels)
print("DATa :", data)

# *** Most expensive hotels ***
```



Cereri in functie de perioada anului

```
with connection.cursor() as cursor:
    cursor.execute("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;")
most_expensive = cursor.fetchall()

  print("MOST EXPENSIVE: ", most_expensive)

  for hotel in most_expensive:
      bar_labels.append(hotel[0])
      bar_data.append(hotel[1])
```