INSERT INTO angajat VALUES(1003,'Moldovan','Karina','Manager', 8000);

INSERT INTO angajat VALUES (1006, 'Constantinescu', 'Alexandru', 'Ospatar', 3500);

INSERT INTO angajat VALUES (1007, 'Dragomir', 'Ana-Maria', 'Camerista', 2500);

INSERT INTO angajat VALUES (1008, 'Popa', 'Gabriel', 'Ospatar', 6000);

INSERT INTO angajat VALUES (1009, 'Radu', 'Adriana', 'Ospatar', 5800);

INSERT INTO angajat VALUES (1010, 'Munteanu', 'Cristian', 'Ospatar', 5500);

INSERT INTO angajat VALUES (1011, 'Dumitru', 'Elena', 'Camerista', 5200);

INSERT INTO angajat VALUES (1012, 'Gheorghe', 'Andreea', 'Camerista', 5000);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (1, 'Serviciu excelent!', 5, 4001);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (2, 'Camere curate, dar mic dejun mediocru.', 3, 4002);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (3, 'Personal amabil', 4, 4003);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (4, 'Experienta excelenta!', 5, 4004);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (5, 'Serviciu minunat, nu necesită îmbunătățiri.', 5, 4005);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (6, 'Nu a fost o experienta placuta.', 1, 4008);

INSERT INTO feedback (ID\_FEEDBACK, COMENTARIU, NOTA, ID\_CLIENT)

VALUES (7, 'Ospitalitate minunata, dar camerele necesita curatare.', 4, 4007);

INSERT INTO client VALUES(4010,'Constantinescu','Alexandru','constalex@gmail.com','0765417295','angajat')

SELECT \*

From rezervare;

--Sa se afiseze valoarea totala a tuturor rezervarilor

SELECT SUM(camera.pret) as ValoareaTotala

FROM rezervare

JOIN camera ON rezervare.id\_camera = camera.id\_camera;

--Sa se afiseze cate camere double are hotelul

SELECT COUNT(id\_camera)

FROM camera

WHERE tip\_camera='Double';

SELECT \*

From angajat;

ALTER TABLE angajat

ADD salariu NUMBER(10);

UPDATE angajat

SET salariu = 5000

WHERE id\_angajat = 1001;

UPDATE angajat

SET salariu = 3500

WHERE id\_angajat = 1002;

--Afiseaza numele complet al fiecarui angajat care are salariul mai mare sau egal cu 5000 de lei

SELECT Nume || ' ' || Prenume AS NumeComplet, salariu

FROM angajat

WHERE salariu >= 5000;

SELECT \*

FROM rezervare;

-- Sa se afiseze durata fiecare rezervari facute;

SELECT ID\_REZERVARE,

ID\_CAMERA,

ID\_CLIENT,

ID\_ANGAJAT,

(DATA\_SFARSIT - DATA\_INCEPUT) AS DurataRezervare

FROM rezervare;

--Salariul mediu al angajatilor

SELECT ROUND(AVG(salariu),2) AS SalariuMediu

FROM angajat;

--Realizeaza o evaluare scurta a notelor pentru a fi verificata mai rapid

Select \*

FROM feedback;

SELECT ID\_FEEDBACK,

DECODE(NOTA,

1, 'Foarte Slab',

2, 'Slab',

3, 'Acceptabil',

4, 'Bun',

5, 'Excelent',

'N/A') AS EvaluareScurta

FROM feedback;

--Stabileste o ierarhie pe 3 nivele a angajatilor

SELECT NUME, PRENUME,

CASE FUNCTIE

WHEN 'Manager' THEN 'Nivel 1'

WHEN 'Receptioner' THEN 'Nivel 2'

ELSE 'Nivel 3'

END AS NivelFunctie

FROM angajat;

SELECT\*

FROM angajat;

SELECT\*

FROM client;

--afiseaza angajatii si clientii

SELECT 'Angajat' AS Tip, NUME, PRENUME FROM angajat

UNION

SELECT 'Client' AS Tip, NUME, PRENUME FROM client;

--Lista angajatilor care au fost si clienti ai hotelului

SELECT NUME, PRENUME FROM angajat

INTERSECT

SELECT NUME, PRENUME FROM client;

--Angajatii care nu au fost si clienti

SELECT NUME, PRENUME FROM angajat

MINUS

SELECT NUME, PRENUME FROM client;

--Stabilirea unor categorii pentru salariile angajatilor

SELECT ID\_ANGAJAT, NUME, PRENUME, FUNCTIE, SALARIU,

CASE

WHEN SALARIU < 3000 THEN 'Salariu Mic'

WHEN SALARIU >= 3000 AND SALARIU < 5000 THEN 'Salariu Mediu'

WHEN SALARIU >= 5000 AND SALARIU < 7000 THEN 'Salariu Mare'

WHEN SALARIU >= 7000 THEN 'Salariu Excelent'

ELSE 'Necunoscut'

END AS CategorieSalariu

FROM angajat;

--Creste cu 20% salariul angajatilor care au mai putin de 3000 de lei

UPDATE angajat

SET SALARIU = SALARIU \* 1.20

WHERE SALARIU < 3000;

--Cate rezervari a facut fiecare anagajat

SELECT NUME, PRENUME, FUNCTIE,

(SELECT COUNT(\*) FROM rezervare WHERE rezervare.ID\_ANGAJAT = angajat.ID\_ANGAJAT) AS NumarRezervari

FROM angajat;

--Sa se afise lista simplificata a angajatilor si a clientilor

CREATE VIEW angajati\_si\_clienti AS

SELECT

ID\_ANGAJAT,

NUME AS ANGAJAT\_NUME,

PRENUME AS ANGAJAT\_PRENUME,

FUNCTIE

FROM

angajat

UNION

SELECT

ID\_CLIENT,

NUME AS CLIENT\_NUME,

PRENUME AS CLIENT\_PRENUME,

'N/A' AS FUNCTIE

FROM

client;

SELECT \* FROM angajati\_si\_clienti;

CREATE SYNONYM evaluare FOR feedback;

SELECT \* FROM evaluare;