***ACADEMIA DE STUDII ECONOMICE, BUCUREŞTI***

***FACULTATEA DE CIBERNETICĂ, STATISTICĂ ŞI INFORMATICĂ ECONOMICĂ***

BAZE DE DATE

PROIECT

AGENŢIE DE TURISM

1/19/2017

Prunceac Tiberiu

Grupa 1045, Seria D

Specializare: Informatică Economică

Prof. coordonator: Oprea Simona

***Agenţie de turism***

***1.Descrierea bazei de date: obiectivul proiectului, descrierea tabelelor şi a atributelor, precizarea restricţiilor şi a tipurilor de legături.***

Am ales să creez o bază de date cu informaţii referitoare la o agenţie de turism pentru gestiunea activităţii de contractare a ofertelor.

Baza de date a agenţiei cuprinde detalii atât despre ofertele societăţii, cât şi despre angajaţii firmei, clienţii acesteia şi contractele încheiate.

Oferă o gamă de oferte din care clienţii îşi pot alege orice sejur doresc şi să beneficieze de toate informaţiile necesare încheierii contractului.

Pentru aceasta vom folosi 6 tabele:

* **Sediu\_agenţie\_turism:** conţine informaţii generale despre sediul agenţiei descrise

|  |  |  |
| --- | --- | --- |
| **Câmp** | **Tip** | **Restricţii** |
| cod\_agenţie | NUMBER(3) | Primary Key |
| nume\_agenţie | VARCHAR2(20) | Not Null |
| Adresa | VARCHAR2(50) |  |
| Telefon | VARCHAR2(20) | Not Null |
| Email | VARCHAR2(30) | Check |

* **Oferta**: conţine criterii despre ofertele agenţiei destinate clienţilor

|  |  |  |
| --- | --- | --- |
| **Câmp** | **Tip** | **Restricţii** |
| cod\_ofertă | NUMBER(3) | Primary Key |
| cod\_agenţie | NUMBER(3) | Foreign Key |
| Preţ | NUMBER(5,2) | Not Null |
| număr\_zile | NUMBER(2) | Not Null |
| Locaţie | VARCHAR2(35) |  |
| cod\_destinaţie | NUMBER(3) | Foreign Key |

* **Clienti\_ag**: conţine informaţii despre clienţii interesaţi de o anumită destinaţie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Câmp** | | **Tip** | **Restricţii** | |
| cod\_client | NUMBER(3) | | | Primary Key |
| nume\_client | VARCHAR2(20) | | | Not Null |
| prenume\_client | VARCHAR2(20) | | | Not Null |
| adresa\_client | VARCHAR2(50) | | |  |
| telefon\_client | VARCHAR2(20) | | | Not Null |
| email\_client | VARCHAR2(30) | | | Check |

* **Angajat**: conţine datele angajaţilor agenţiei de turism

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Câmp** | | **Tip** | **Restricţii** | |
| cod\_angajat | NUMBER(3) | | | Primary Key |
| nume\_angajat | VARCHAR2(20) | | | Not Null |
| prenume\_angajat | VARCHAR2(20) | | | Not Null |
| Salariu | NUMBER(6,2) | | |  |
| telefon\_angajat | VARCHAR2(20) | | | Not Null |
| cod\_agentie | NUMBER(3) | | | Foreign Key |

* **Destinatie**: conţine informaţii amănunţite despre fiecare destinaţie din baza de date

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Câmp** | | **Tip** | **Restricţii** | |
| cod\_destinatie | NUMBER(3) | | | Primary Key |
| Tara | VARCHAR2(20) | | |  |
| Distanta | NUMBER(5) | | | Not Null |
| Transport | VARCHAR2(20) | | |  |

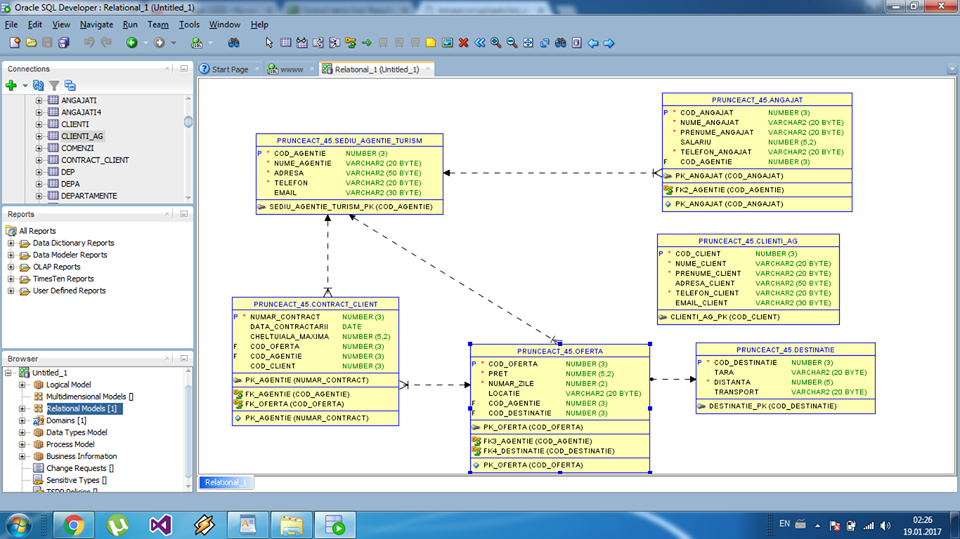
* **Contract\_client**: conţine informaţiile necesare clientului şi a sejurului ales de acesta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Câmp** | | **Tip** | **Restricţii** | |
| numar\_contract | NUMBER(3) | | | Primary Key |
| data\_contractarii | DATE | | |  |
| cheltuiala\_maxima | NUMBER(6,2) | | |  |
| cod\_oferta | NUMBER(3) | | | Foreign Key |
| cod\_agentie | NUMBER(3) | | | Foreign Key |
| cod\_client | NUMBER(3) | | | Foreign Key |

În această agenţie de turism avem urmatoarele **legături**:

* O agenţie poate avea mai multe oferte (între tabela Sediu\_agenţie\_turism şi Ofertă) : 1-M
* O agenţie poate avea unul sau mai mulţi angajaţi (între tabela Sediu\_agenţie\_turism şi Angajat): 1-M
* O destinaţie poate avea mai multe oferte (între tabela Destinaţie şi Ofertă):1-M
* O ofertă poate avea mai multe contracte ale clienţilor (între tabela Ofertă şi Contract\_client): 1-M
* Un angajat poate încheia mai multe contracte(între tabela Angajat şi Contract\_client): 1-M
* Un client poate avea mai multe contracte incheiate (între tabela Clienţi şi Contract\_client): 1-M

***2. Schema bazei de date***

****

***3. Crearea tabelelor***

**CERINŢĂ**: ***Să se creeze tabelele Sediu\_agentie\_turism, Oferta, Angajat, Destinatie, Clienti şi Contract\_client pe baza structurii şi tipurilor de date prezentate mai sus.***

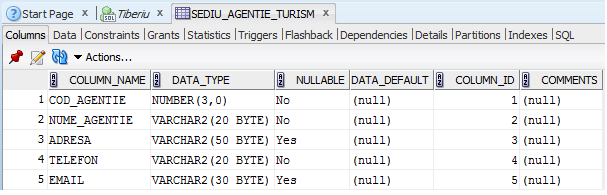
1. CREATE TABLE **Sediu\_agentie\_turism** ( cod\_agentie NUMBER(3) PRIMARY KEY,

nume\_agentie VARCHAR2(20) NOT NULL,

adresa VARCHAR2(50) ,

telefon VARCHAR2(20) NOT NULL,

email VARCHAR2(30), CONSTRAINT ck\_email CHECK (email LIKE ‘%@%.%’) );

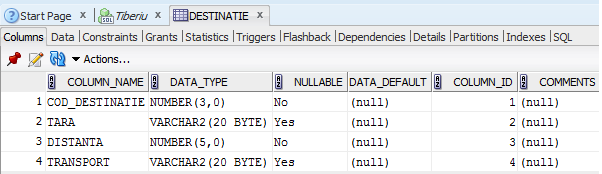
****

1. CREATE TABLE **Destinatie** ( cod\_destinatie NUMBER(3) PRIMARY KEY,

tara VARCHAR2(20),

distanta NUMBER(5) NOT NULL,

transport VARCHAR2(20) );

****

1. CREATE TABLE **Oferta** ( cod\_oferta NUMBER(3) CONSTRAINT pk\_oferta PRIMARY KEY,

pret Number(5,2) NOT NULL,

numar\_zile NUMBER(2) NOT NULL,

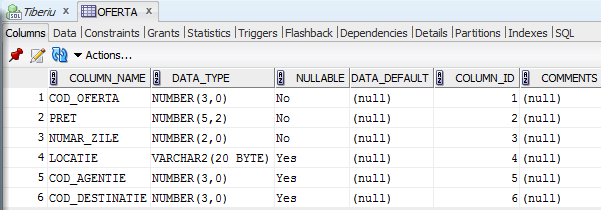
locatie VARCHAR2(20),

cod\_agentie NUMBER(3),

cod\_destinatie NUMBER(3),

CONSTRAINT fk3\_agentie FOREIGN KEY (cod\_agentie) REFERENCES Sediu\_agentie\_turism(cod\_agentie),

CONSTRAINT fk 4\_destinatie FOREIGN KEY (cod\_destinatie) REFERENCES Destinatie(cod\_destinatie));

****

1. CREATE TABLE **Clienti\_ag** ( cod\_client NUMBER(3) PRIMARY KEY,

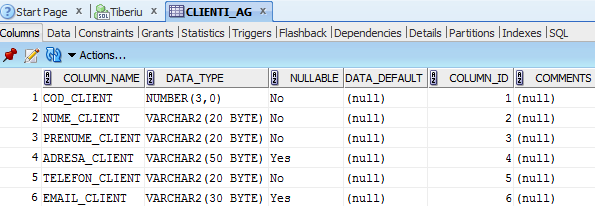
nume\_client VARCHAR2(20) NOT NULL,

prenume\_client VARCHAR2(20) NOT NULL,

adresa\_client VARCHAR2(50),

telefon\_client VARCHAR2(20) NOT NULL,

email\_client VARCHAR2(30), CONSTRAINT ck2 CHECK( email\_client LIKE '%@%.%') );



1. CREATE TABLE **Angajat** ( cod\_angajat NUMBER(3) CONSTRAINT pk\_angajat PRIMARY KEY,

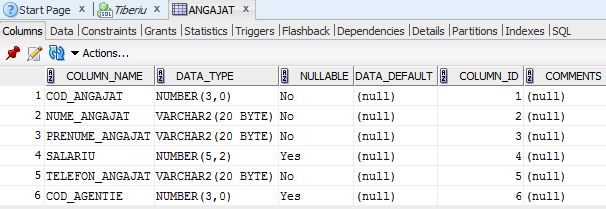
nume\_angajat VARCHAR2(20) NOT NULL,

prenume\_angajat VARCHAR2(20) NOT NULL,

salariu NUMBER(5,2), telefon\_angajat VARCHAR2(20) NOT NULL,

cod\_agentie NUMBER(3),

CONSTRAINT fk2\_agentie FOREIGN KEY (cod\_agentie) REFERENCES Sediu\_agentie\_turism(cod\_agentie) );



1. CREATE TABLE **Contract\_client** ( numar\_contract NUMBER(3) CONSTRAINT pk\_agentie

PRIMARY KEY,

data\_contractarii DATE, cheltuiala\_maxima NUMBER(5,2),

cod\_oferta NUMBER(3),

cod\_agentie NUMBER(3),

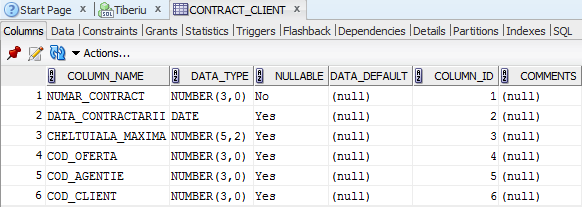
cod\_client NUMBER(3),

CONSTRAINT fk\_oferta FOREIGN KEY(cod\_oferta) REFERENCES Oferta(cod\_oferta),

CONSTRAINT fk\_agentie FOREIGN KEY(cod\_agentie) REFERENCES Sediu\_agentie\_turism(cod\_agentie),

CONSTRAINT fk\_client FOREIGN KEY(cod\_client) REFERENCES Clienti\_ag(cod\_client));

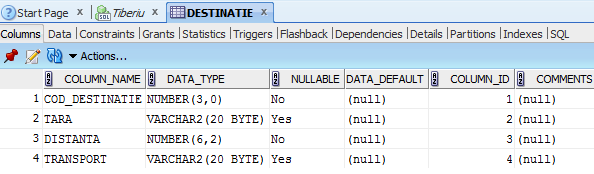
SELECT \* FROM Contract\_client;



***4.Actualizarea structurii tabelelor şi modificarea restricţiilor de integritate***

**CERINŢA 1: *Să se modifice lungimea atributului “ distanta” din tabela Destinatie astel încât aceasta sa fie de 4 caractere şi să conţină 2 zecimale.***

ALTER TABLE Destinatie



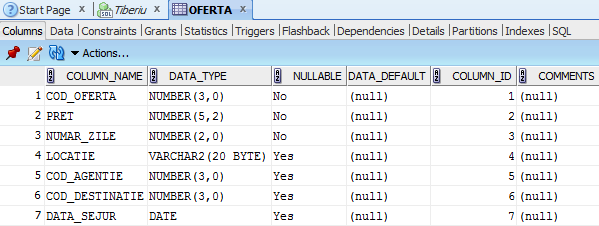
MODIFY( distanta NUMBER(6,2) );

**CERINTA 2: *Să se adauge in tabela Oferta un câmp nou numit data\_sejur de tip DATE.***

ALTER TABLE Oferta

ADD( data\_sejur DATE);

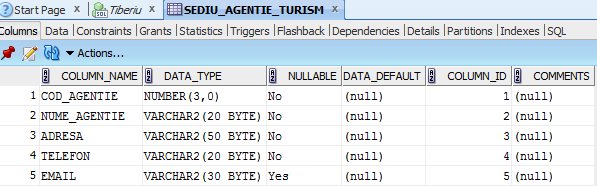
SELECT\* FROM Oferta;



**CERINŢA 3: *Să se modifice coloana adresa din tabela Sediu\_agentie\_turism astfel încât să nu poată lua valori NULL.***

ALTER TABLE Sediu\_agentie\_turism

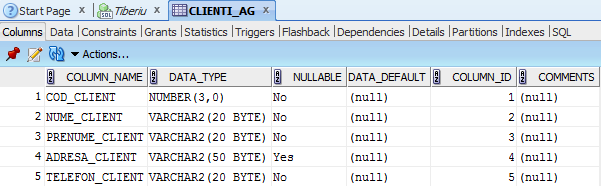
MODIFY adresa NOT NULL;



**CERINTA 4: *Să se şteargă coloana email\_client din tabela Clienti\_ag.***

ALTER TABLE Clienti\_ag

DROP COLUMN email\_client;



**CERINŢA 5:** ***Să se adauge o restricţie pe coloana numar\_zile din tabela Oferta astfel încât perioada sejurului sa fie cuprinsa intre 7-14 zile. Să se dezactiveze apoi această restricţie.***

ALTER TABLE Oferta

ADD CONSTRAINT check\_durata CHECK (numar\_zile>7 and numar\_zile<14);

SELECT \* FROM Oferta;

După executarea scriptului, dacă dorim să introducem o înregistrare nouă în tabela Oferta care nu are în câmpul numar\_zile o valoare cuprinsa intre 7-14 zile, se va genera o eroare.

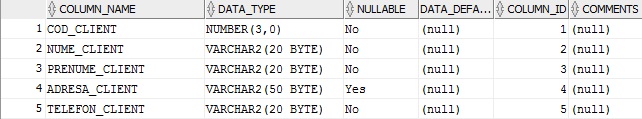
ALTER TABLE Oferta

DISABLE CONSTRAINT check\_durata;

**CERINŢA 6: *Să se adauge o restricţie de tip NOT NULL cu numele “restrictie” coloanei adresa\_client din tabela Clienti\_ag . Să se redenumească această restricţie în „restrictie\_adresa” şi apoi să se şteargă această restricţie.***

ALTER TABLE Clienti\_ag

ADD CONSTRAINT restrictie CHECK( adresa\_client IS NOT NULL);

SELECT \* FROM Clienti\_ag;

ALTER TABLE Clienti\_ag

RENAME CONSTRAINT restrictie TO restrictie\_adresa;

ALTER TABLE Clienti\_ag

DROP CONSTRAINT restrictie\_adresa;

***5.Adăugarea (10-15) de înregistrări în fiecare tabelă***

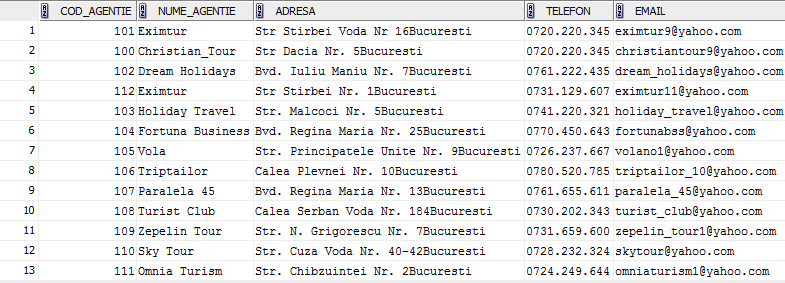
* **Tabela Sediu\_agentie\_turism**

INSERT INTO Sediu\_agentie\_turism VALUES ( 101, 'Eximtur', 'Str Stirbei Voda Nr 16

Bucuresti','0720.220.345', 'eximtur9@yahoo.com');

INSERT INTO Sediu\_agentie\_turism VALUES ( 102, 'DREAM HOLIDAYS', 'Bvd Iuliu Maniu Nr 7

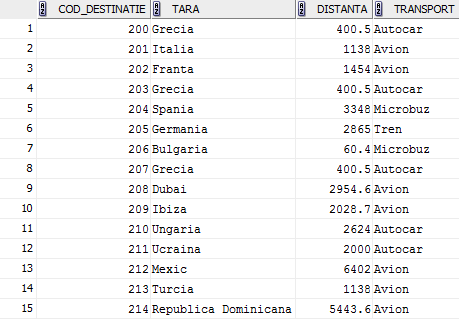
Bucuresti','0721.229.607', 'dreamholidays1@yahoo.com');



* **Tabela Destinatie**

INSERT INTO Destinatie VALUES (200, 'Grecia', 20.11 , 'Autocar');

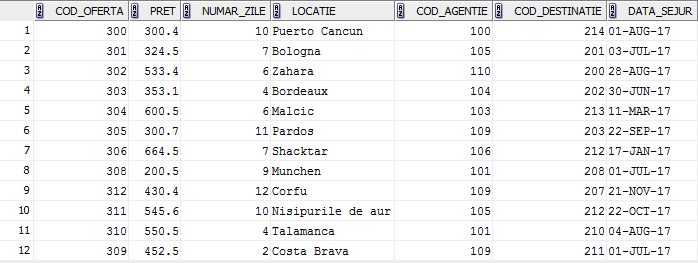
INSERT INTO Destinatie VALUES (201, 'Italia', 1138 , 'Avion');

****

* **Tabela Oferta**

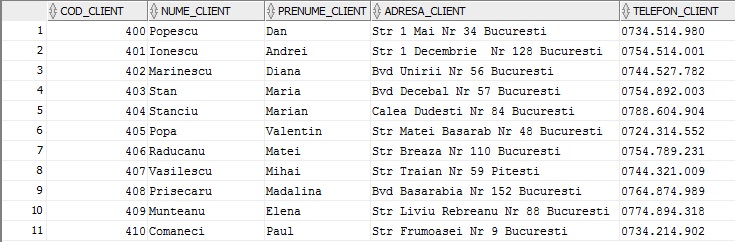
INSERT INTO Oferta VALUES (300, 225.50, 10, 'Puerto Cancun ', 100, 214, TO\_DATE('01-AUG-2017','DD-MM-YYYY') );

INSERT INTO Oferta VALUES (301, 300.7, 7, 'Roma', 105, 201, TO\_DATE('01-AUG-2017','DD-MON -YYYY') );



* **Tabela Clienti\_ag**

INSERT INTO Clienti\_ag VALUES (400, 'Popescu', 'Dan', 'Str 1 Mai Nr 34 Bucuresti', '0734.514.980');

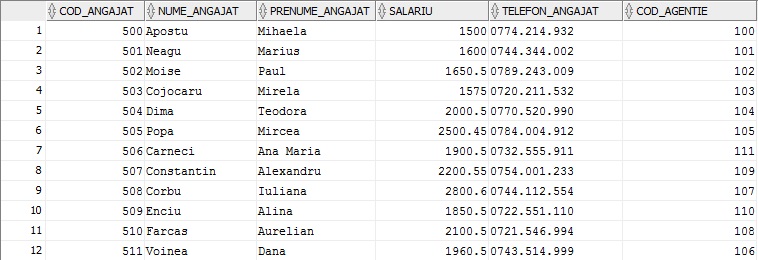
INSERT INTO Clienti\_ag VALUES (401, 'Ionescu', 'Andrei', 'Str 1 Decembrie Nr 128 Bucuresti', '0754.514.001');

* **Tabela Angajat**

INSERT INTO Angajat VALUES (500, 'Apostu', 'Mihaela', 500, '0774.214.932', 100);

INSERT INTO Angajat VALUES (501, 'Neagu', 'Marius', 1600, '0744.344.002', 101);

INSERT INTO Angajat VALUES (502, 'Moise', 'Paul', 1650.5, '0789.243.009', 102);

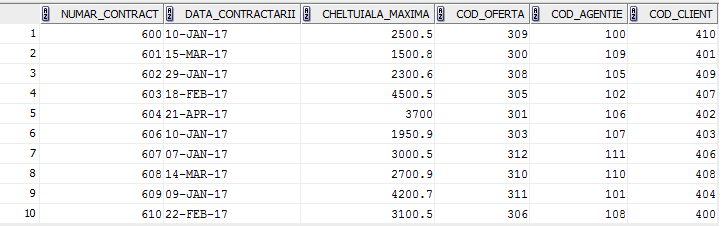


* **Tabela Contract\_client**

INSERT INTO Contract\_client VALUES (600, TO\_DATE('10-JAN-2017', 'DD-MON-YYYY'), 2500.5, 309, 100, 410);

INSERT INTO Contract\_client VALUES (601, TO\_DATE('15-MAR-2017', 'DD-MON-YYYY'), 1500.8, 300, 109, 401);

INSERT INTO Contract\_client VALUES (602, TO\_DATE('29-JAN-2017', 'DD-MON-YYYY'), 2300.6, 308, 105, 409);



***6.Actualizarea înregistrărilor***

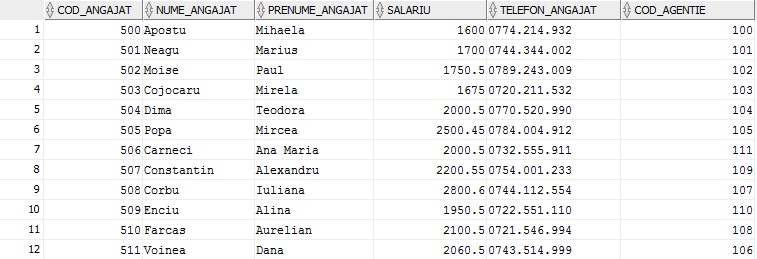
**Cerinţa 1**: ***Să se crească cu 100 salariul angajaţilor din tabela Angajat care au salariul mai mic decat 2000.***

UPDATE Angajat

SET salariu = salariu+100

WHERE salariu<2000;

SELECT \* FROM Angajat;



**Cerinţa 2: *Să se actualizeze salariul angajatului , din tabela Angajat, al cărui cod este egal cu 505, cu salariul angajatului care are cod\_angajat = 510.***

UPDATE Angajat

SET salariu = (SELECT salariu FROM Angajat WHERE cod\_angajat=510)

WHERE cod\_angajat=505;

SELECT\* FROM Angajat;



**Cerinţa 3: *Să se modifice email-ul agenţiei ,cu codul 110, cu valoarea: „skytagency10@yahoo.com”.***

UPDATE Sediu\_agentie\_turism

SET email = 'skytagency10@yahoo.com'

WHERE cod\_agentie = 110;

SELECT \* FROM Sediu\_agentie\_turism;



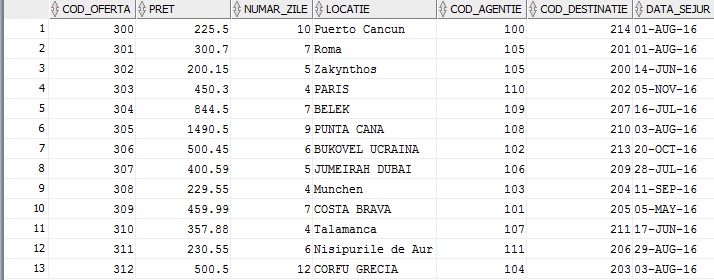
**Cerinţa 4: *Să se modifice numele locaţiilor din tabela Oferta astfel încât să apară cu litere mari pentru acelea care au preţul cel puţin egal cu 400.***

UPDATE Oferta

SET locatie = UPPER(locatie)

WHERE pret IN ( SELECT pret FROM Oferta WHERE pret >= 400);

SELECT \* FROM Oferta;



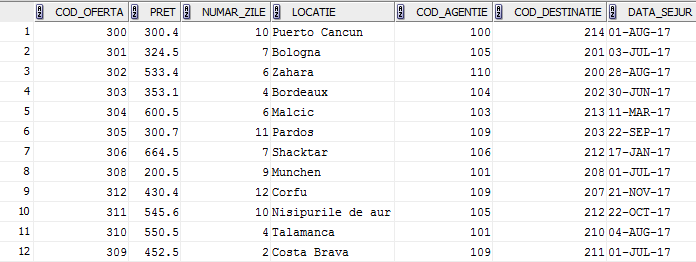
**Cerinţa 5: *Să se modifice cheltuiala maximă , ce are numărul contractului egal cu 603, cu cea mai mică cheltuială din tabela Contract\_client.***

UPDATE Contract\_client

SET cheltuiala\_maxima = ( SELECT MIN(cheltuiala\_maxima) FROM Contract\_client)

WHERE numar\_contract = 603;

SELECT \* FROM Contract\_client;

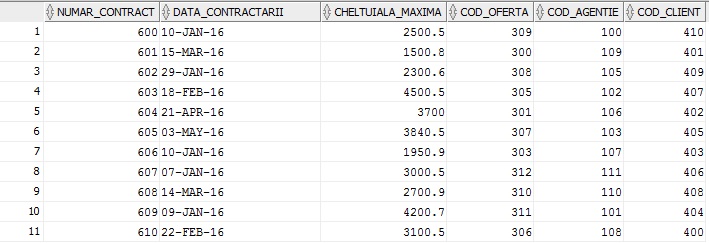


**Cerinţa 6:** ***Să se şteargă toate contractele înregistrate înainte de 10 ianuarie 2016 din tabela Contract\_client .Să se anuleze apoi comanda.***

DELETE FROM Contract\_client

WHERE data\_contractarii < TO\_DATE('10-JAN-2016', 'DD-MON-YYYY');

SELECT \* FROM Contract\_client;



ROLLBACK;

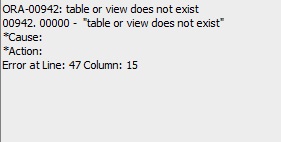
SELECT \* FROM Contract\_client;

***7.Ştergerea şi recuperarea unei tabele***

**CERINŢA 1: *Să se şteargă tabela Destinatie.***

DROP TABLE Destinatie CASCADE CONSTRAINTS;

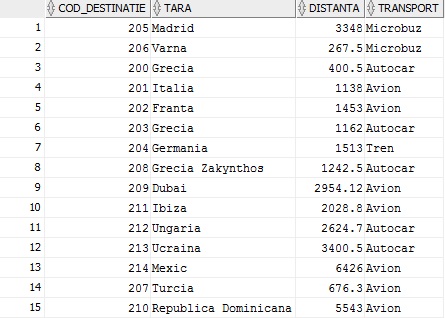
SELECT \* FROM Destinatie;



**CERINŢA 2:** ***Să se recupereze tabela Destinatie.***

FLASHBACK TABLE Destinatie TO BEFORE DROP;

SELECT \* FROM Destinatie;



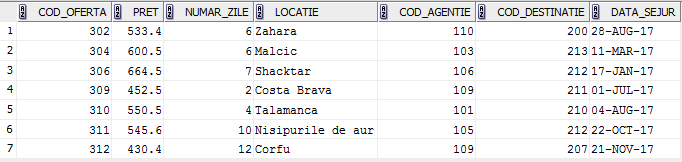
***8.Exemple de interogări variate***

**CERINŢA 1: *Să se selecteze toate ofertele , valabile după 1 august 2016, cu preţul mai mare decât 400.***

SELECT \* FROM Oferta

WHERE data\_sejur > TO\_DATE('01-AUG-2016', 'DD-MON-YYYY')

AND pret > 400;

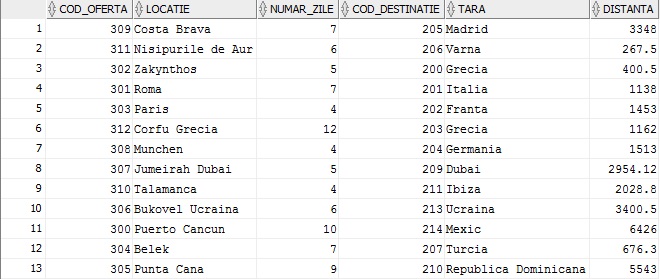


**CERINŢA 2: *Să se selecteze cod\_oferta, locatie, numar\_zile, cod\_destinatie, tara din tabela Oferta si tabela Destinatie şi să se realizeze joncţiunea dintre cele două tabele.***

SELECT o.cod\_oferta, o.locatie, o.numar\_zile, d.cod\_destinatie, d.tara, d.distanta

FROM Oferta o, Destinatie d

WHERE o.cod\_destinatie = d.cod\_destinatie;



**CERINŢA 3: *Să se afişeze cod\_angajat, nume\_angajat, prenume\_angajat, salariul, cod\_agentie pentru angajatii care nu lucrează la agenţia ce are codul 105 şi al căror salariu este mai mic decat oricare angajat care lucrează la agenţia ce are codul 105.***

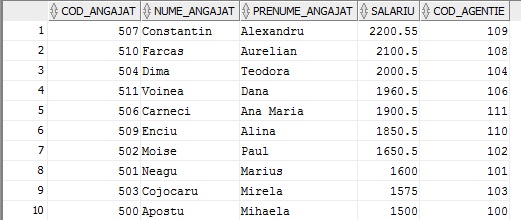
SELECT cod\_angajat, nume\_angajat, prenume\_angajat, salariu, cod\_agentie

FROM Angajat

WHERE salariu < ANY (SELECT salariu FROM Angajat WHERE cod\_agentie = 105)

AND cod\_agentie <> 105

ORDER BY salariu DESC;



**CERINŢA 4: *Să se selecteze contractele clienţilor încheiate numai în luna ianuarie.***

SELECT cc.\*, cl.\*

FROM Contract\_client cc, Clienti\_ag cl

WHERE cc.cod\_client = cl.cod\_client

AND cc.data\_contractarii LIKE '%JAN%';

**CERINŢA 5: *Să se selecteze numai ofertele cu preţul cuprins între 400 si 1000 euro şi denumirile lor. Să se ordoneze crescător după pret.***

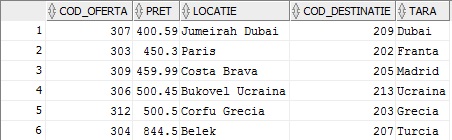
SELECT o.cod\_oferta, o.pret, o.locatie, d.cod\_destinatie, d.tara

FROM Oferta o, Destinatie d

WHERE o.cod\_destinatie = d.cod\_destinatie

AND pret BETWEEN 400 and 1000

ORDER BY pret ASC;

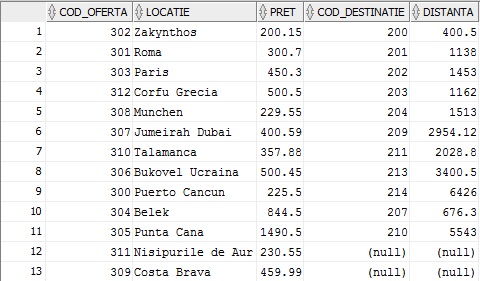


**CERINŢA 6: *Să se afişeze codul ofertei, locatia, pretul , codul destinatiei şi distanta chiar dacă acestea nu au fost comandate.***

SELECT o.cod\_oferta, o.locatie, o.pret, d.cod\_destinatie, d.distanta

FROM Oferta o, Destinatie d

WHERE o.cod\_destinatie = d.cod\_destinatie(+);

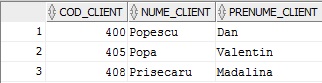


**CERINŢA 7:** ***Să se afişeze clienţii al căror nume începe cu litera „p” utilizând funcţia INITCAP.***

SELECT cod\_client, nume\_client, prenume\_client

FROM Clienti\_ag

WHERE nume\_client like INITCAP('p%');

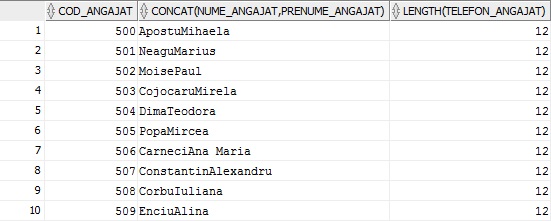


**CERINŢA 8: *Să se afişeze cod\_client numai pentru clienţii cu coduri care încep cu cifra 5, de lungime 2, de forma 50, numele clienţilor concatenate cu prenumele acestora şi lungimea numărului de telefon al fiecărui angajat.***

SELECT cod\_angajat, CONCAT(nume\_angajat, prenume\_angajat ), LENGTH(telefon\_angajat )

FROM Angajat

WHERE SUBSTR(cod\_angajat,1,2) = 50;

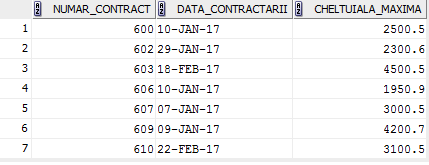


**CERINŢA 9: *Să se afişeze contractele încheiate în lunile ianuarie şi februarie.***

SELECT numar\_contract, data\_contractarii, cheltuiala\_maxima

FROM Contract\_client

WHERE EXTRACT(MONTH FROM data\_contractarii) IN(1,2);



**CERINŢA 10: *Să se calculeze veniturile anuale ale angajaţilor.***

SELECT cod\_angajat, nume\_angajat, LENGTH(nume\_angajat),

prenume\_angajat, LENGTH(prenume\_angajat),

salariu\*12 + salariu\*12\*NVL(cod\_angajat,500) venit\_anual

FROM Angajat;



**CERINŢA 11: *Să se calculeze valoarea maximă, valoarea medie, valoarea minimă şi valoarea totală a preţurilor ofertelor.***

SELECT AVG(pret), MAX(pret), MIN(pret), SUM(pret)

FROM Oferta;



**CERINŢA 12: *Să se afişeze destinaţiile şi distanţa medie pentru acele destinaţii ale căror distanţe medii sunt mai mari de 1000 km. Să se sorteze descrescător după distanţa medie.***

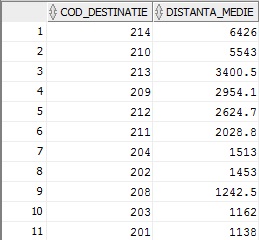
SELECT cod\_destinatie, ROUND(AVG(distanta),1) distanta\_medie

FROM Destinatie

GROUP BY cod\_destinatie

HAVING ROUND(AVG(distanta),1) >1000

ORDER BY distanta\_medie DESC;



**CERINŢA 13: *Să se afişeze tipul ofertei pentru clienţi în funcţie de preţul acesteia, astfel:***

* ***Dacă preţul este < 300 , să se afişeze : ofertă standard***
* ***Dacă preţul este între 300 şi 500, să se afişeze : ofertă cu transport,cazare şi mic-dejun inclus***
* ***Dacă preţul este > 500 ,să se afişeze : ofertă all inclusive.***

SELECT cod\_oferta, locatie, pret,

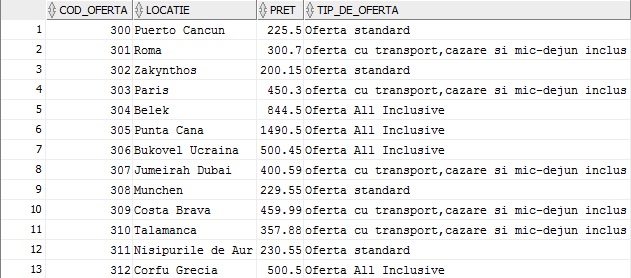
CASE WHEN pret < 300 THEN 'Oferta standard'

WHEN pret BETWEEN 300 and 500 THEN 'ofertă cu transport,cazare şi mic-dejun inclus'

else 'Oferta All Inclusive'

END tip\_de\_oferta

FROM Oferta;



**CERINŢA 14: *Să se afişeze capitala fiecărei ţări în funcţie de numele lor, astfel:***

* ***Dacă ţara este Grecia, atunci să se afişeze: Atena***
* ***Dacă ţara este Italia, atunci să se afişeze: Roma***
* ***Dacă ţara este Germania,atunci să se afişeze: Berlin***
* ***Dacă ţara esteTurcia, atunci să se afişeze: Ankara***
* ***Dacă ţara este Franţa, atunci să se afişeze: Paris***
* ***Dacă ţara este Ungaria, atunci să se afişeze: Budapesta***
* ***Altfel să se afişeze : altă capitală.***

SELECT cod\_destinatie, distanta, tara,

**DECODE**(tara, 'Grecia', 'Atena', 'Italia', 'Roma', 'Germania', 'Moscova',

'Turcia', 'Ankara', 'Franta', 'Paris', 'Ungaria', 'Budapesta','Alta capitala')

AS capitale\_tari

FROM Destinatie;



**CERINŢA 15: *Să se calculeze discountul (DC) pentru clienţi , astfel:***

* ***Dacă clientul are cheltuiala maxima mai < 2500 atunci DC=15%***
* ***Dacă clientul are cheltuiala maxima cuprinsă între 3500 şi 3000 atunci DC=20%***
* ***Dacă clientul are cheltuiala maxima mai > 3500 atunci DC=30%.***

SELECT cl.nume\_client, cc.cheltuiala\_maxima,

**CASE** WHEN cc.cheltuiala\_maxima <2500 THEN 0.15

WHEN cc.cheltuiala\_maxima BETWEEN 2500 AND 3500 THEN 0.2

WHEN cc.cheltuiala\_maxima >=3500 THEN 0.3

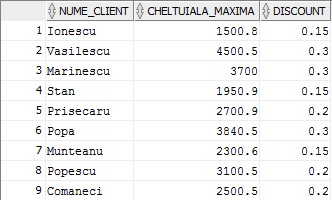
ELSE 0

END discount

FROM Clienti\_ag cl, Contract\_client cc

WHERE cl.cod\_client = cc.cod\_client

GROUP BY cl.nume\_client, cc.cheltuiala\_maxima;

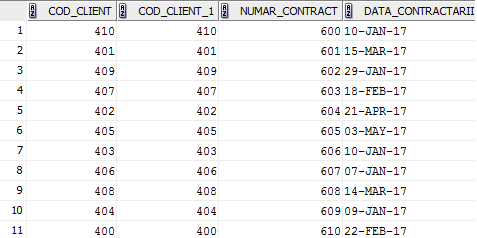


**CERINŢA 16: *Să se afişeze clienţii care au încheiat contracte cu angajaţii firmei, precum şi cei care nu au încheiat comenzi.***

select cl.cod\_client, cl.cod\_client, cc.numar\_contract, cc.data\_contractarii

FROM Clienti\_ag cl, Contract\_client cc

WHERE cl.cod\_client = cc.cod\_client (+);



**CERINŢA 17*: Să se afişeze clienţii in funcţie de reţeaua de telefonie mobilă din care fac parte, astfel:***

* ***Clientul nume poate fi in reţeaua Orange (dacă numărul de telefon începe cu 074 sau 075)***
* ***Clientul nume poate fi in reţeaua Cosmote ( dacă numărul de telefon începe cu 076)***
* ***Clientul nume poate fi in reţeaua Vodafone ( dacă numărul de telefon începe 072 sau 073 sau 077).***

SELECT 'Clientul ' ||nume\_client|| ' este in reteaua de telefonie mobila Orange.'

AS RETEA

FROM Clienti\_ag

WHERE telefon\_client LIKE ('075%') OR telefon\_client LIKE ('074%')

**UNION**

SELECT 'Clientul ' ||nume\_client||' este in reteaua de telefonie mobila Cosmote.'

AS RETEA

FROM Clienti\_ag

WHERE telefon\_client LIKE ('076%') OR telefon\_client LIKE ('078%')

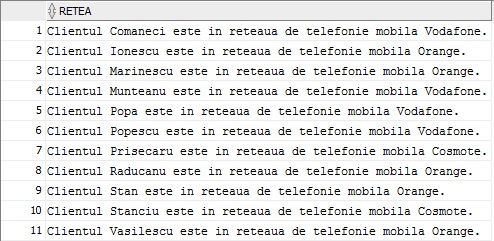
**UNION**

SELECT 'Clientul ' ||nume\_client||' este in reteaua de telefonie mobila Vodafone.'

AS RETEA

FROM Clienti\_ag

WHERE telefon\_client LIKE ('072%') OR telefon\_client LIKE ('073%') OR telefon\_client LIKE ('077%');



**CERINŢA 18: *Să se selecteze codul şi numele agenţiilor de turism şi angajaţii acestora care au salariul cel puţin egal cu 1600, numărul de telefon să fie în reţeaua de telefonie mobilă Orange, iar codul acestora să fie diferit de 504 şi 506.***

SELECT s.cod\_agentie, s.nume\_agentie,a.cod\_angajat, a.nume\_angajat, a.salariu, a.telefon\_angajat

FROM Sediu\_agentie\_turism s, Angajat a, Oferta o

WHERE s.cod\_agentie = a.cod\_agentie

AND a.salariu >=1600

**INTERSECT**

SELECT s.cod\_agentie, s.nume\_agentie,a.cod\_angajat, a.nume\_angajat, a.salariu, a.telefon\_angajat

FROM Sediu\_agentie\_turism s, Angajat a, Oferta o

WHERE s.cod\_agentie = a.cod\_agentie

AND telefon\_angajat LIKE ('074%')

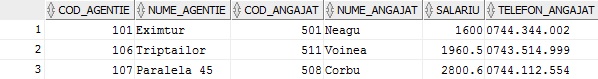
**INTERSECT**

SELECT s.cod\_agentie, s.nume\_agentie, a.cod\_angajat, a.nume\_angajat, a.salariu, a.telefon\_angajat

FROM Sediu\_agentie\_turism s, Angajat a, Oferta o

WHERE s.cod\_agentie = a.cod\_agentie

AND a.cod\_angajat NOT IN (504, 506);



**CERINŢA 19*: Să se afişeze angajaţii care au salariul cuprins între 1000 şi 3000 fără cei care au salariul 1500 şi 1600.***

SELECT \* FROM Angajat WHERE salariu BETWEEN 1000 AND 3000

**MINUS**

SELECT \* FROM Angajat WHERE salariu IN (1500, 1600);



**CERINŢA 20: *Să se afişeze data\_contractarii , codul\_contractului şi clientul căruia i s-a încheiat contractual în anul 2016, exceptându-le pe acelea care au fost realizate în luna ianuarie a anului 2016.***

SELECT cc.numar\_contract, cc.data\_contractarii, cl.cod\_client, cl.nume\_client, cl.prenume\_client

FROM Contract\_client cc, Clienti\_ag cl

WHERE cl.cod\_client = cc.cod\_client

AND EXTRACT (YEAR FROM data\_contractarii) = 2016

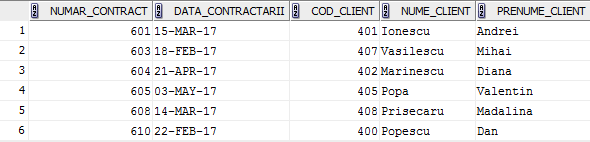
MINUS

SELECT cc.numar\_contract, cc.data\_contractarii, cl.cod\_client, cl.nume\_client, cl.prenume\_client

FROM Contract\_client cc, Clienti\_ag cl

WHERE cl.cod\_client = cc.cod\_client

AND EXTRACT( MONTH FROM data\_contractarii) = 01;



**CERINŢA 21: *Să se afişeze preţul minim al ofertelor, numărul maxim de zile ale unei oferte şi distanţa medie a destinaţiilor în funcţie de codul acestora. Să se ordoneze după codul destinaţiilor.***

SELECT d.cod\_destinatie,

MIN(o.pret) AS pret\_minim,

MAX(o.numar\_zile) AS maxim\_numar\_zile,

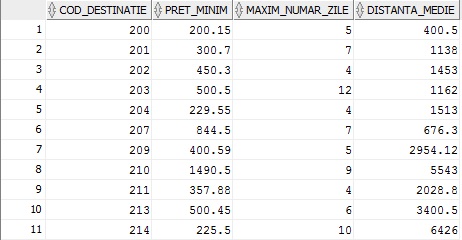
AVG(d.distanta) AS distanta\_medie

FROM Oferta o, Destinatie d

WHERE o.cod\_destinatie = d.cod\_destinatie

GROUP BY d.cod\_destinatie

ORDER BY cod\_destinatie ASC;



**CERINŢA 22*: Să se afişeze locaţia ofertelor, numărul de zile , numărul ofertelor şi distanţa către destinaţii. Să se ordoneze după locaţie.***

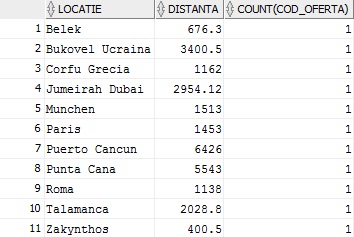
SELECT o.locatie, d.distanta, COUNT(cod\_oferta)

FROM Oferta o, Destinatie d

WHERE o.cod\_destinatie = d.cod\_destinatie

GROUP BY o.locatie, o.numar\_zile, d.distanta

ORDER BY o.locatie;



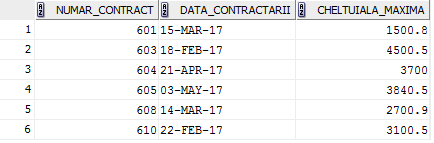
**CERINŢA 23: *Să se afişeze numărul contractelor, data şi cheltuiala maximă care au fost realizate după ce s-a realizat contractul cu numărul 602.***

SELECT numar\_contract, data\_contractarii, cheltuiala\_maxima

FROM Contract\_client

WHERE data\_contractarii> ( SELECT data\_contractarii FROM Contract\_client

Where numar\_contract = 602);



**CERINŢA 24: *Să se afişeze cod\_ofertă, cod\_agentie, nume\_agentie, adresa şi preţul minim al ofertelor din lunile iulie şi august. Să se ordoneze după preţul minim.***

SELECT s.cod\_agentie, s.nume\_agentie, MIN(o.pret) pret\_minim

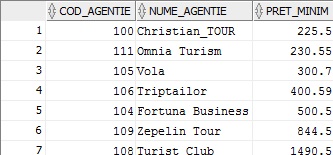
FROM Sediu\_agentie\_turism s, Oferta o

WHERE s.cod\_agentie = o.cod\_agentie

AND EXTRACT( MONTH FROM o.data\_sejur) IN (7,8)

GROUP BY s.cod\_agentie, s.nume\_agentie

ORDER BY MIN(o.pret);



**CERINŢA 25: *Să se afişeze contractele , data contractării lor, numărul de luni între data contractării şi data curentă, următoarea zi de luni după data contractării, precum şi data corespunzătoare după 1 lună de la data contractării.***

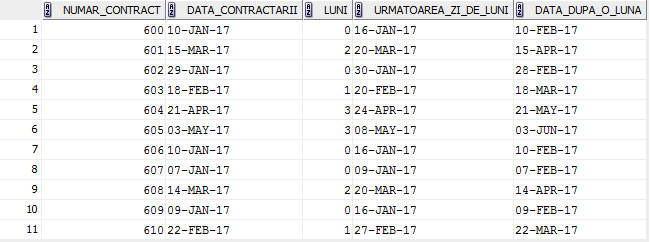
SELECT numar\_contract, data\_contractarii,

round( MONTHS\_BETWEEN( data\_contractarii, sysdate)) luni,

NEXT\_DAY(data\_contractarii, 'MONDAY') urmatoarea\_zi\_de\_luni,

ADD\_MONTHS(data\_contractarii,1) data\_dupa\_o\_luna

FROM Contract\_client;



***9.Gestiunea altor obiecte ale bazei de date***

* **Vederi**

**CERINŢA 1: *Să se creeze o tabelă virtuală care să conţină toate informaţiile despre ofertele care au numărul de zile = 6.***

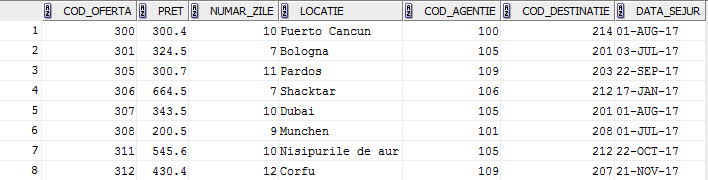
CREATE VIEW nume\_oferta

AS

SELECT \* FROM Oferta

WHERE numar\_zile > 6;

SELECT \* FROM nume\_oferta;



**CERINŢA 2: *Să se creeze o tabelă virtuală care să conţină codul agenţiilor şi numele angajaţilor acestora doar pentru angajaţii cu salariul mai mare sau egal cu 1600 lei şi cu atributul prenume\_angajat NOT NULL, iar permisiunea acestei tabele să fie doar în scopul vizualizării datelor.***

CREATE VIEW v\_angajat

AS

SELECT s.cod\_agentie, a.nume\_angajat, a.prenume\_angajat, a.salariu

FROM Sediu\_agentie\_turism s JOIN Angajat a ON (s.cod\_agentie = a.cod\_agentie)

WHERE a.prenume\_angajat IS NOT NULL

AND a.salariu >= 1600

WITH READ ONLY;

SELECT \* FROM v\_angajat;



**CERINŢA 3:** ***Să se creeze o tabelă virtuală care să conţină toate informaţiile despre contractele elaborate în lunile ianuarie şi martie şi care au cheltuiala maximă cuprinsă între 1500 şi 3000.***

CREATE VIEW v\_contract

AS

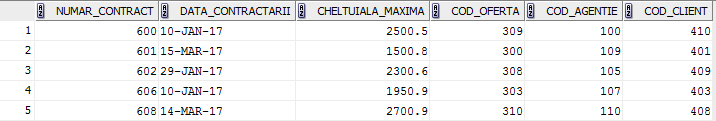
SELECT \* FROM Contract\_client

WHERE cheltuiala\_maxima BETWEEN 1500 AND 3000

AND EXTRACT ( MONTH FROM data\_contractarii) IN (1,3)

WITH READ ONLY;

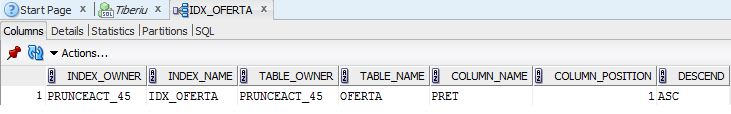
SELECT \* FROM v\_contract;



* **Indecşi**

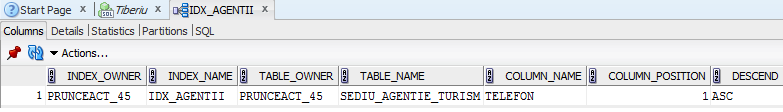
**CERINŢA 1: *Să se creeze un index pentru facilitarea accesului la preţurile ofertelor.***

CREATE INDEX idx\_oferta ON Oferta(pret);



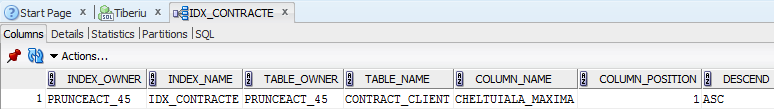
**CERINŢA 2: *Să se creeze un index pentru facilitarea accesului la numărul de telefon al agenţiilor de turism.***

CREATE INDEX idx\_agentii ON Sediu\_agentie\_turism(telefon);



**CERINŢA 3: *Să se creeze un index pentru facilitarea accesului la costul maxim al clienţilor.***

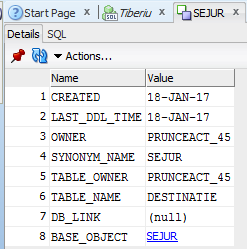
CREATE INDEX idx\_contracte ON Contract\_client(cheltuiala\_maxima);



* **Sinonime**

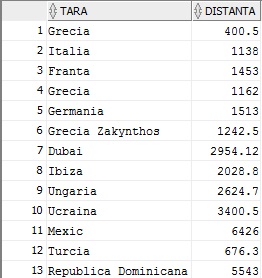
**CERINŢA 1: *Să se creeze un sinonim „Sejur” pentru tabela Destinatie, să se interogheze această tabelă utilizând sinonimul creat şi apoi să se şteargă sinonimul „Sejur”.***

CREATE SYNONYM Sejur FOR Destinatie;



SELECT tara, distanta

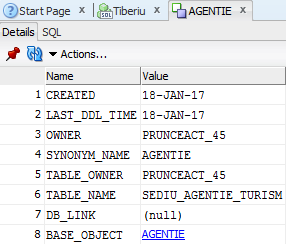
FROM Sejur;



DROP SYNONYM Sejur;

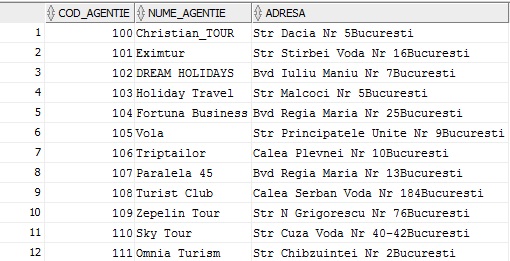
**CERINŢA 2: *Să se creeze un sinonim „Agentie” pentru tabela Sediu\_agentie\_turism, să se interogheze această tabelă utilizând sinonimul creat şi apoi să se şteargă sinonimul creat.***

CREATE SYNONYM Agentie FOR Sediu\_agentie\_turism;



SELECT cod\_agentie, nume\_agentie, adresa

FROM Agentie;



DROP SYNONYM Agentie;

* **Secvenţe**

**CERINŢA 1: *Să se creeze o secvenţă care să aibă valoarea de început 10, iar valoarea maximă 110. Să se incrementeze cu pasul 1 şi să nu cicleze.***

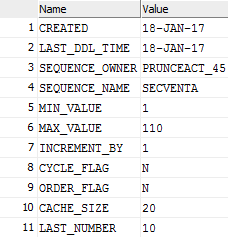
CREATE SEQUENCE secventa

START WITH 10

INCREMENT BY 1

MAXVALUE 110

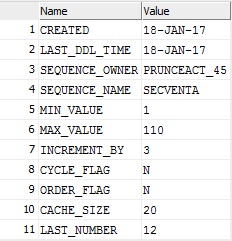
NOCYCLE;



**CERINŢA 2: *Să se modifice secvenţa astfel încât pasul de incrementare să fie 3.***

ALTER SEQUENCE secventa

INCREMENT BY 3;



**CERINŢA 3: *Să se interogheze tabela Oferta.***

SELECT \* FROM Oferta;

**CERINŢA 4:** ***Să se folosească secvenţa pentru adăugarea unei înregistrări în tabela Oferta.***

INSERT INTO Oferta (cod\_oferta, pret, numar\_zile,data\_sejur)

VALUES( 355, secventa.nextval, 9, sysdate);

SELECT \* FROM Oferta;

