**Dobre Adriana Lia, grupa 141**

***Agenție de turism***

**Exercițiul 1:**

Exemplele din acest document se referă la proiectarea unui model de date ce furnizează informații despre angajați, oferte, cazări și tot ceea ce reprezintă evenimente importante în lumea agențiilor de turism.

Vom prezenta modelul de date, restricțiile pe care trebuie să le respecte și diagrama E/R corespunzătoare acestuia.

Modelul de date va gestiona informații legate de organizarea și funcționarea activităților din cadrul unei agenții de turism. Buna funcționare a acestora este asigurată de angajații agenției care pot fi de mai multe tipuri: contabil, ghid turistic sau agent de vânzare. Fiecare angajat are un contract de muncă .

Clienții care aleg să îsi organizeze vacanța cu ajutorul agenției de turism apelează la agenții de vânzări ai companiei. La rândul lui, agentul analizează cerințele clientului și le oferă cea mai bună ofertă de vacanță.

În momentul în care clienții acceptă oferta, aceștia rezervă o cameră care poate avea mai multe facilități. Oferta prezentată de agent are o destinație specifică și poate include transport care este asigurat de firme specializate în acest domeniu.

**Exercițiul 2:**

Modelul de date respectă anumite restricții de funcționare:

* Un angajat poate avea un singur contract, iar un contract poate aparține unui singur angajat.
* Un agent de vânzări poate avea mai mulți clienți sau niciunul, iar clienții apelează la agenți de vânzări pentru organizarea vacanței.
* Clienții pot cumpăra mai multe oferte sau niciuna, iar ofertele pot fi cumpărate de clienți sau de niciunul.
* Agentul de vânzări poate obține o ofertă sau niciuna, iar o ofertă poate fi obținută de un agent de vânzare sau nu.
* Un client poate rezerva o cameră care poate avea cel puțin o facilitate sau niciuna. O facilitate trebuie să aparțină cel puțin unei camere.
* O ofertă include o cameră și putem avea camere nerezervate.
* O cameră aparține unui hotel, iar hotelul poate avea cel puțin o cameră.
* Pentru localizarea unei oferte, s-a considerat o singură adresă. Destinația se poate repeta ( într-o destinație se poate desfășura cel puțin o ofertă).
* O ofertă include maxim un transport, iar transportul poate fi inclus în mai multe oferte.
* O firmă poate asigura cel puțin un transport, iar un transport poate fi asigurat de o singură firmă.

**Exercițiul 3:**

**Entități**

Pentru modelul de date, referitor la agenția de turism, structurile CONTRACT\_ANGAJAT, ANGAJAT, CLIENT, OFERTA, CAMERA, HOTEL,FACILITATI, DESTINATIE, TRANSPORT, FIRMA reprezintă entități.

Vom prezenta entitățile modelului de date, dând o descriere completă a fiecăreia. În plus, pentru fiecare entitate se va preciza cheia primară.

CONTRACT\_ANGAJAT = contract încheiat de un angajat cu agenția de turism, prin care cel dintâi este obligat să presteze în favoarea celei din urmă o anumită muncă în schimbul unui salariu. Atributul *id\_contract* reprezintă cheia primară.

ANGAJAT = persoană fizică, aparținând agenției de turism, care este responsabilă de buna desfășurare a activităților din cadrul acesteia. Ea poate fi contactată fie pentru a beneficia de serviciile oferite de agenție, fie pentru probleme întâmpinate pe parcusul acestora. Atributul *id\_angajat* reprezintă cheia primară.

CONTABIL = subentitate a entității ANGAJAT, persoană calificată care ține contabilitatea agenției de turism. Cheia primară a entității este *id\_angajat.*

GHID\_TURISTIC = subentitate a entității ANGAJAT, persoana calificată care conduce și îndrumă un grup de turiști, oferind explicațiile necesare referitoare la locurile vizitate și (dacă este cazul) asigurând totodată desfășurarea în cele mai bune condiții a programului turistic. Cheia primară a entității este *id\_angajat*.

AGENT\_VANZARI = subentitate a entității ANGAJAT, persoana calificată care se ocupă de vinderea serviciilor. Cheia primară a entității este *id\_angajat*.

CLIENT = persoană fizică care beneficiază de serviciile oferite de agenția de turism. Atributul *id\_client* reprezintă cheia primară.

CAMERA = entitate care conține detalii despre spațiul în care va fi cazat clientul care se bucură de serviciile agenției. Cheia primară a entității este *id\_camera*.

HOTEL = enititate care conține detalii despre clădirea în care va fi cazat clientul pe durata serviciilor. Cheia primară a entității este *id\_hotel*.

FACILITATI = entitate care identifică caracteristicile pe care le prezintă o cameră. Atributul *id\_facilitate* reprezintă cheia primară.

OFERTA = entitate care include detaliile legate de vacanța aleasă de client. Cheia primară a entității este *id\_oferta*.

DESTINATIE = entitate care identifică locaţia în care se desfăşoară o anumită ofertă. Cheia primară a entităţii este *id\_locatie*.

TRANSPORT = entitate care oferă informații legate de deplasarea clienților către destinația aleasă. Atributul *id\_transport* este cheia primară.

FIRMA = persoană juridică ce asigură transportul clienților către destinațiile alese. Cheia primară a entității este *id\_firma*.

**Exercițiul 4:**

**Relații**

Vom prezenta relațiile modelului de date, dând o descriere completă a fiecăruia. Denumirile acestor legături sunt sugestive, reflectând conținutul acestora și entitățile pe care le leagă. De asemenea, vom preciza și cardinalitatea minimă și maximă.

ANGAJAT\_semneaza\_CONTRACT\_ANGAJAT = relație care leagă entitățile ANGAJAT și CONTRACT\_ANGAJAT, reflectând legătura dintre acestea ( ce contract semnează angajatul). Ea are cardinalitate minimă 1:1 (un angajat trebuie să semneze un contract, iar un contract trebuie să fie semnat de un angajat) și cardinalitate maximă 1:1 (un angajat poate semna un singur contract, iar un contract poate fi semnat de un singur contract).

AGENT\_VANZARI\_obtine\_pentru\_CLIENT\_OFERTA\_care\_se\_cazeaza\_în\_CAMERA= relație de tip 3 ce leagă entitățile AGENT\_VANZARI, CLIENT, OFERTA și CAMERĂ reflectând de către cine și pentru cine a fost obținută oferta, ce ofertă și ce cameră a fost rezervată. Denumirea acestei relații va fi *OBȚINE*.

CAMERA\_are\_FACILITATI = relatie de tip many\_to\_many dintre entitățile CAMERA și FACILITATI, reflectând caracteristicile pe care le prezintă camera. Relația are cardinalitatea minimă 1:0 (o facilitate trebuie să aparțină unei camere) și cardinalitate maximă M:M (o camera poate avea mai multe facilitate, iar o facilitate poate aparține mai multor camere).

CAMERA\_exista\_în\_HOTEL = relație ce leagă entitățile CAMERA și HOTEL (în ce hotel se află camera). Relația are cardinalitatea minimă 1:1 (o cameră trebuie să aparțină unui hotel, iar un hotel trebuie să aibă cel puțin o cameră) și cardinalitatea maxima M:1 (un hotel poate avea mai multe camere, iar o cameră poate aparține unui hotel).

OFERTA\_se\_afla\_la\_DESTINATIE = relație ce leagă entitățile OFERTA și DESTINATIE ( în ce destinație este plasată oferta). Relația are cardinalitatea minimă 1:1 (o oferta trebuie să se desfășoare într-o destinație, iar o destinație trebuie să aparțină unei oferte) și cardinalitate maxima M:1 (o ofertă se poate desfășura într-o singură destinație, iar o destinație poate să aparțină mai multor oferte).

OFERTA\_include\_TRANSPORT = relație ce leagă entitățile OFERTA și TRANSPORT ( ce transport include oferta). Relația are cardinalitatea minimă 1:0 (o oferta poate să nu includă transport, iar un transport trebuie să aparțină unei oferte) și cardinalitate maximă M:1 (o ofertă poate include un transport, iar un transport poate să aparțină mai multor oferte).

TRANSPORT\_apartine\_FIRMA = relație ce leagă entitățile TRANSPORT și FIRMA ( carei firme aparține transportul). Relația are cardinalitatea minima 0:1 (un transport trebuie să aparțină unei firme) și cardinalitatea maximă M:1 ( un transport poate aparține unei firme, iar o firmă poate avea mai multe transporturi).

**Exercițiul 5:**

**Atribute**

Entitatea CONTRACT\_ANGAJAT are ca atribute: id\_contract, id\_angajat, data\_angajarii, experienta și salariu.

id\_contract = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui contract de muncă.

id\_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui angajat. Acesta trebuie să fie unic.

data\_angajării = variabilă de tip dată calendaristică, care reprezintă data angajării angajatului respectiv. Are ca valoare implicită data actuală (din ziua respectivă).

experientă = variabilă de tip întreg, de lungime maximă 3, care reprezintă numărul de ani al unui angajat petrecut la locul de muncă.

salariu= variabilă de tip întreg, de lungime maximă 10, care reprezintă salariul unui angajat. Acesta nu poate fi null.

Entitatea ANGAJAT are ca atribute: id\_angajat, nume , prenume, adresa, telefon, tip\_angajat.

id\_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui angajat.

nume = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele angajatului.

prenume = variabilă de tip caracter, de lungime maximă 20, care reprezintă prenumele angajatului.

Numele împreună cu prenumele trebuie să fie unic și nu pot fi nule.

adresa= variabilă de tip caracter, de lungime maximă 20, care reprezintă adresa angajatului.Nu poate fi nulă.

telefon= variabilă de tip caracter, de lungime maximă 20, care reprezintă numărul de telefon al angajatului. Nu poate fi nul și este unic.

tip\_angajat= variabilă de tip caracter, de lungime maximă 20, care reprezintă tipul/numele locului de muncă al angajatului. Nu poate fi nul.

Subentitatea CONTABIL are ca atribute : id\_angajat, functie.

id\_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui contabil.

funcție= variabilă de tip caracter, de lungime maximă 40, care reprezintă rolul contabilului în cadrul agenției.

Subentitatea GHID\_TURISTIC are ca atribute: id\_angajat, limba\_vorbita.

id\_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui ghid turistic.

limba\_vorbita= variabilă de tip caracter, de lungime maximă 20, care reprezintă limba străină pe care o vorbește ghidul turistic.

Subentitatea AGENT\_VANZARI are ca atribute: id\_angajat, comision.

id\_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui agent de vânzări.

comision= variabilă de tip întreg, de lungime maximă 10, care reprezintă suma de bani pe care trebuie să o plătească agentul de vânzare.

Entitatea CLIENT are ca atribute: id\_client, nume, prenume, CNP, adresa\_email.

id\_client = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui client.

nume = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele clientului.

prenume = variabilă de tip caracter, de lungime maximă 20, care reprezintă prenumele clientului.

Numele împreună cu prenumele trebuie să fie unic și nu pot fi nule.

CNP= variabilă de tip caracter, de lungime maximă 20, care reprezintă CNP-ul clientului.Nu poate fi nul și trebuie să fie unic.

adresa\_email= variabilă de tip caracter, de lungime maximă 20, care reprezintă adresa de email a clientului. Nu poate fi nulă și este unică.

telefon= variabilă de tip caracter, de lungime maximă 20, care reprezintă numărul de telefon al clientului. Nu poate fi nul și este unic.

Entitatea CAMERA are ca atribute: id\_camera, id\_hotel, nr\_camera,etaj,capacitate.

id\_camera= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei camere.

id\_hotel= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui hotel.Nu poate fi nul.

nr\_camera= variabilă de tip întreg, de lungime maximă 5, care reprezintă numărul camere în care este cazat clientul. Nu poate fi nul.

etaj= variabilă de tip întreg, de lungime maximă 5, care reprezintă etajul la care se află camera.

capacitate= variabilă de tip caracter, de lungime maximă 20, care reprezintă tipul camerei.

Entitatea HOTEL are ca atribute:id\_hotel, nume\_hotel, nr\_de\_contact, nr\_stele.

id\_hotel= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui hotel.

nume\_hotel = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele hotelului și nu poate fi nul.

nr\_de\_contact= variabilă de tip caracter, de lungime maximă 20, care reprezintă numărul de telefon al hotelului. Nu poate fi nul și este unic.

nr\_stele= variabilă de tip întreg, de lungime maximă 10, care reprezintă numărul de stele pe care le deține hotelul.

Entitatea FACILITATI are ca atribute: id\_facilitate, denumire.

id\_facilitate= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei facilități.

denumire= variabilă de tip caracter, de lungime maximă 40, care reprezintă facilitatea de care dispune camera. Nu poate fi nulă.

Entitatea OFERTA are ca atribute: id\_oferta, id\_destinatie, id\_transport, tip\_oferta, pret.

id\_oferta= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei oferte.

id\_destinatie= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei destinații. Nu poate fi nul.

id\_transport= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui transport.

tip\_oferta= variabilă de tip caracter, de lungime maximă 30, care reprezintă tipul ofertei. Nu poate fi nul.

pret= variabilă de tip întreg, de lungime maximă 10, care reprezintă prețul ofertei. Nu poate fi nul.

Entitatea DESTINATIE are ca atribute: id\_destinatie, nume\_destinatie, tara, continent.

id\_destinatie= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei destinații.

oras= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele orașului în care se desfășoară oferta. Nu poate fi nul.

tara= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele țării în care se desfășoară oferta. Nu poate fi nul.

continent= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele continentului în care se desfășoară oferta. Nu poate fi nul.

Entitatea TRANSPORT are ca atribute: id\_transport, id\_firma, mijloc\_de\_transport, durata\_max, durata\_min.

id\_transport= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui transport.

id\_firma= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei firme.

mijloc\_de\_transport= variabilă de tip caracter, de lungime maximă 20, care reprezintă tipul de mijloc de transport cu care se poate realiza transportul către destinație. Nu poate fi nul.

durata\_max=variabilă de tip întreg, de lungime maximă 5, care reprezintă numărul maxim de ore pe care o poate avea durata transportului.

durata\_min=variabilă de tip întreg, de lungime maximă 5, care reprezintă numărul minim de ore pe care o poate avea durata transportului.

durata\_max trebuie să fie mai mare sau egală decât durata\_min.

Entitatea FIRMA are ca atribute id\_firma, denumire\_firma, sediu, telefon.

id\_firma= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei firme.

denumire= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele firmei care se ocupă de transport. Nu poate fi nul.

sediu= variabilă de tip caracter, de lungime maximă 20, care reprezintă numele sediului firmei de transport.

telefon= variabilă de tip caracter, de lungime maximă 20, care reprezintă numărul de telefon al firmei. Nu poate fi nul și este unic.

Relația AGENT\_VANZARI\_obtine\_pentru\_CLIENT\_OFERTA\_care\_se\_cazeaza\_în\_ CAMERA are ca atribute: id\_client, id\_camera, id\_angajat, id\_oferta, data\_start, data\_end.

id\_client = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui client.

id\_camera= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei camere.

id\_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unui angajat.

id\_oferta= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei oferte.

data\_start= variabilă de tip dată calendaristică, care reprezintă data în care se cazează clientul. Are ca valoare implicită data actuală (din ziua respectivă).

data\_end= variabilă de tip dată calendaristică, care reprezintă data în care ia sfârșit oferta clientului. Are ca valoare implicită data actuală (din ziua respectivă).

Data\_end trebuie să fie mai mare sau egală cu data\_start.

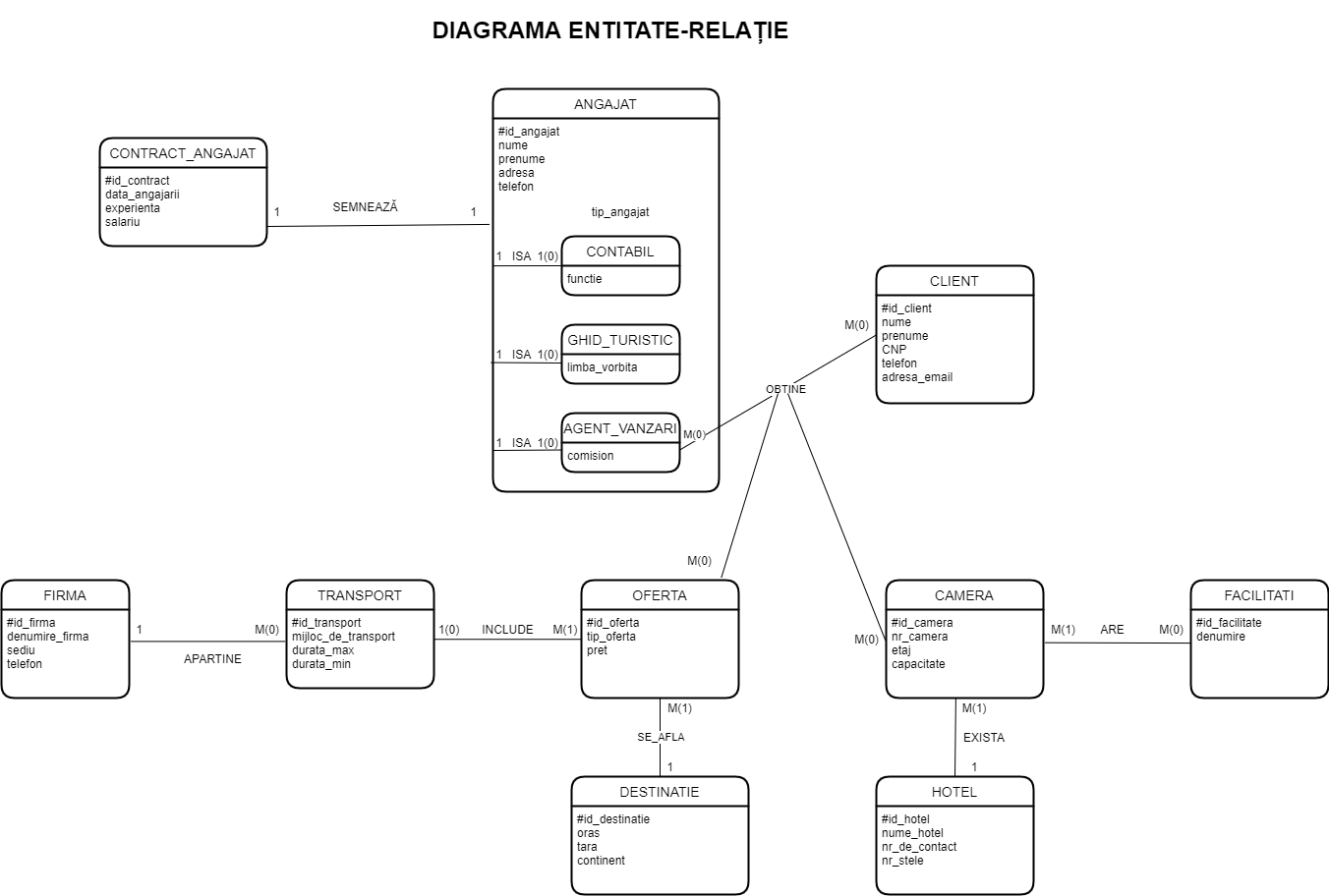
Relația CAMERA\_are\_FACILITATI are ca atribute: id\_camera, id\_facilitate, pret\_suplimentar.

id\_camera= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei camere.

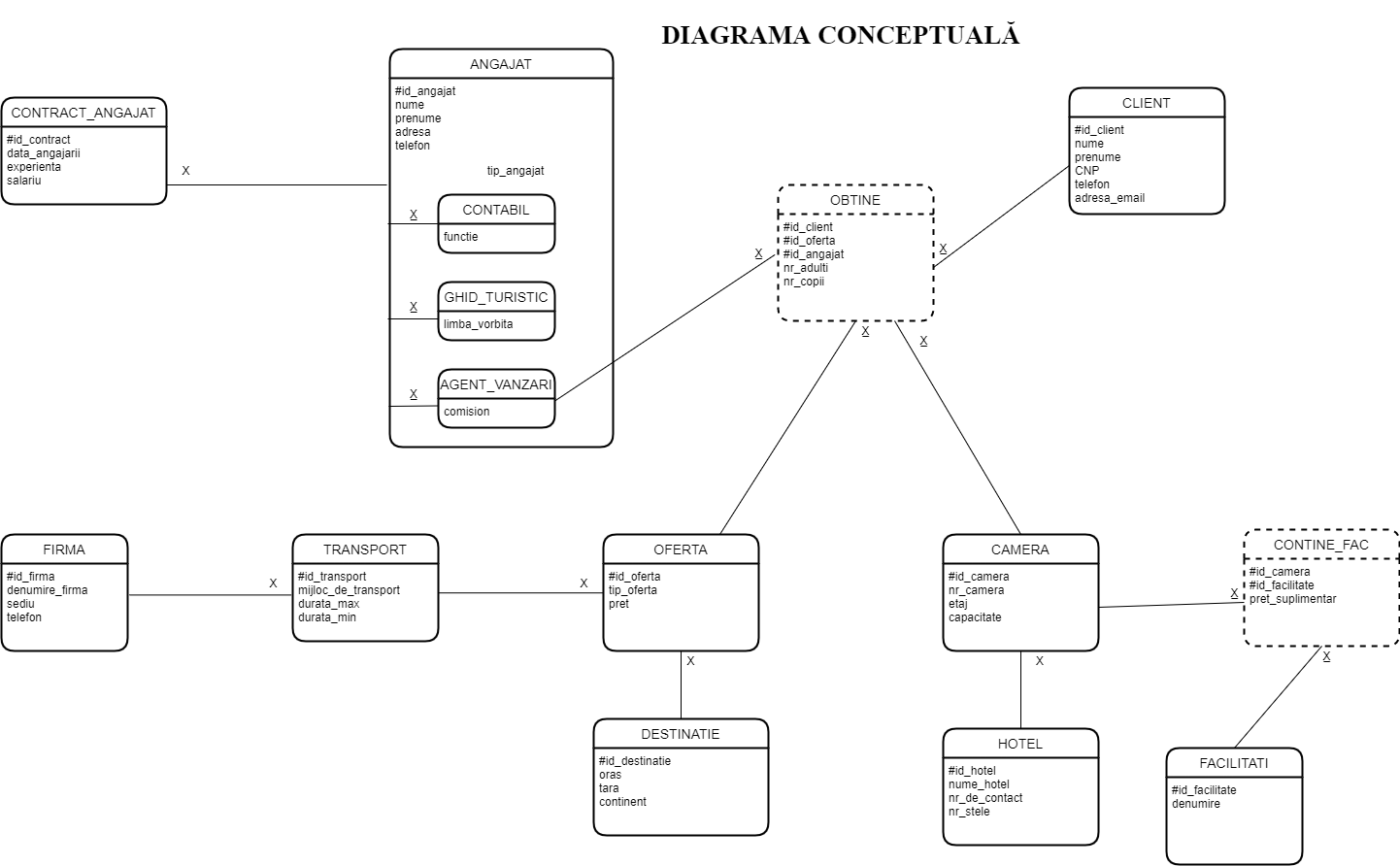
id\_facilitate= variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul unei facilități.

pret\_suplimentar= variabilă de tip întreg, de lungime maximă 5, care reprezintă suma de bani pe care trebuie să o plătească clientul pentru o facilitate suplimentară.

**Exercițiul 6:**



**Exercițiul 7:**



**Exercițiul 8:**

Schemele relaționale corespunzătoare diagramei conceptuale de mai sus sunt următoarele:

CONTRACT\_ANGAJAT(#id\_contract, id\_angajat, data\_angajarii, experienta,salariu)

ANGAJAT(#id\_angajat, nume , prenume, adresa, telefon, tip\_angajat)

CONTABIL(#id\_angajat,functie)

GHID\_TURISTIC(#id\_angajat,limba\_vorbita)

AGENT\_VANZARI(#id\_client,comision)

CLIENT(#id\_client,nume,prenume,CNP,adresa\_email)

CAMERA(#id\_camera,id\_hotel,nr\_camera,etaj,capacitate)

HOTEL(#id\_hotel,nume\_hotel,nr\_de\_contact,nr\_stele)

FACILITATI(#id\_facilitate,denumire)

DESTINATIE(#id\_destinatie,oras,tara,continent)

TRANSPORT(#id\_transport,id\_firma,mijloc\_de\_transport,durata\_max,durata\_min)

FIRMA(#id\_firma,denumire\_firma,sediu,telefon)

OFERTA(#id\_oferta,tip\_oferta,pret,id\_destinatie,id\_transport)

CONTINE\_FAC(#id\_camera, #id\_facilitate,pret\_suplimentar)

OBTINE(#id\_client,#id\_oferta,#id\_angajat,#id\_camera,data\_start,data\_end)

**Exercițiul 9:**

**FN1:**

CONTINE\_FAC (Non FN1)

|  |  |
| --- | --- |
| #id\_camera | id\_facilitate |
| 201 | 1,2 |
| 202 | 3,2 |
| 203 | 4 |
| 207 | 1,5 |

O cameră poate avea mai multe facilități.

Relația nu se află în FNI1, deoarece există atribute cărora nu le corespund valori indivizibile. #id\_camera este cheie primară și trebuie să aibă valoare unică. Pentru a avea cel puțin o facilitate la aceeasi cameră avem nevoie să repetăm valoarea camerei, ceea ce este în contradicție cu unicitatea cheii primare. Soluția ar fi implementarea unei chei primare compuse formată din #id\_cameră si #id\_facilitate(tabelul de mai jos).

CONTINE\_FAC (FN1)

|  |  |
| --- | --- |
| #id\_camera | #id\_facilitate |
| 201 | 1 |
| 201 | 2 |
| 202 | 3 |
| 202 | 2 |
| 203 | 4 |
| 207 | 1 |
| 207 | 5 |

Relația este în FN1, deoarece avem identificator unic pentru toate intrările din tabel.

**FN2:**

OBTINE (Non FN2)

|  |  |  |  |
| --- | --- | --- | --- |
| #id\_camera | data\_start | #id\_client | nume |
| 203 | 10-06-2020 | 101 | Radu |
| 202 | 20-05-2021 | 101 | Radu |
| 206 | 02-11-2020 | 107 | Bianca |
| 203 | 10-07-202 | 107 | Bianca |

O cameră poate fi rezervată de mai mulți clienți., chiar de același client dar în perioade diferite.

O relație R este în a doua formă normală dacă și numai dacă:

* Relația R se află în FNI1; (În cazul nostru, relația R se află în FN1 deoarece avem identificator unic pentru toate intrările din tabel.)
* Fiecare atribut care nu participă la cheia primară este dependent de întreaga cheie primară. În cazul nostru, se observă că atributele data\_start și nume nu sunt chei primare și trebuie să depindă direct de întreaga cheia primară #id\_client și #id\_camera. Însă aceste atribute nu depind direct de întreaga cheie primară, deoarece se pot observa dependența directă dintre id\_client și nume, însemnând că nume depinde direct doar de o parte a cheii primare, și anume doar de id\_client. Deci relația R nu se află în FN2.

Deoarece relația R nu este in FN2, se aplică regula Casey Delobel:

OBTINE(#id\_client, #id\_camera, data\_start)

CLIENT(#id\_client, nume)

Prin urmare, se va transforma în:

|  |  |  |
| --- | --- | --- |
| #id\_camera | data\_start | #id\_client |
| 203 | 10-06-2020 | 101 |
| 202 | 20-05-2021 | 101 |
| 206 | 02-11-2020 | 107 |
| 203 | 10-07-202 | 107 |

|  |  |
| --- | --- |
| #id\_client | nume |
| 101 | Radu |
| 107 | Bianca |

**FN3:**

CAMERA(Non FN3)

|  |  |  |  |
| --- | --- | --- | --- |
| #id\_camera | etaj | nume\_hotel | nr\_stele |
| 201 | 1 | Aqua | 5 |
| 209 | 2 | Aqua | 5 |
| 207 | 3 | Palace | 4 |

O relație R este în FN3 dacă și numai dacă:

* Relația R este în FN2. În cazul nostru, relația R se află în FN2,deoarece deoarece este în FN1 și îndeplinește condiția ca fiecare atribut care nu este cheie (nu participă la cheia primară) să fie dependent de întreaga cheie primară.
* O relaţie este în FN3 dacă şi numai dacă fiecare atribut care nu este cheie, depinde de cheie, de întreaga cheie şi numai de cheie. In exemplul de mai sus se observă că atributul nr\_stele depinde tranzitiv de cheia primară id\_camera prin intermediul atributului nume\_hotel.

Pentru a aduce relaţia în FN3 se aplică regula Casey-Delobel. Relaţia se descompune, prin eliminarea dependenţelor funcţionale tranzitive, în proiecţiile:

CAMERA(#id\_camera,etaj)

HOTEL(nume\_hotel, nr\_stele)

|  |  |
| --- | --- |
| #id\_camera | etaj |
| 201 | 1 |
| 209 | 2 |
| 207 | 3 |

|  |  |
| --- | --- |
| nume\_hotel | nr\_stele |
| Aqua | 5 |
| Palace | 4 |

**Exercițiul 10:**

create table ANGAJAT

( id\_angajat number(5) constraint pk\_ang primary key,

nume varchar2(20) not null,

prenume varchar2(20) not null,

adresa varchar2(20) not null,

telefon varchar2(20) not null,

tip\_angajat varchar2(20) not null,

functie varchar2(40),

limba\_vorbita varchar2(20),

comision number(10)

);

alter table ANGAJAT

add constraint tel\_unic unique(telefon);

alter table ANGAJAT

add constraint nume\_prenume\_unic unique(nume, prenume);

CREATE SEQUENCE SEQ\_ANG

INCREMENT by 10

START WITH 10

MAXVALUE 10000

NOCYCLE;

insert into ANGAJAT

values (10 , 'Popescu', 'Ion', 'Strada Bucuriei', '0745678192', 'Contabil', 'inregistreaza tranzactii', NULL, NULL);

insert into ANGAJAT

values (SEQ\_ANG.NEXTVAL, 'Costache', 'Ioana', 'Strada Aviatiei', '0731278142', 'Ghid turistic', NULL , 'engleza' , NULL);

insert into ANGAJAT

values (SEQ\_ANG.NEXTVAL, 'Bucur', 'Alexandru', 'Strada Soarelui', '0749078122', 'Agent de vanzare', NULL, NULL, 10);

insert into ANGAJAT

values (SEQ\_ANG.NEXTVAL, 'Marinescu', 'Catalina', 'Strada Victoriei', '0721908192', 'Contabil', 'verifica tranzactii', NULL, NULL);

insert into ANGAJAT

values (SEQ\_ANG.NEXTVAL, 'Tudorache', 'Marian', 'Strada Romana', '0731182812', 'Ghid turistic', NULL , 'franceza' , NULL);

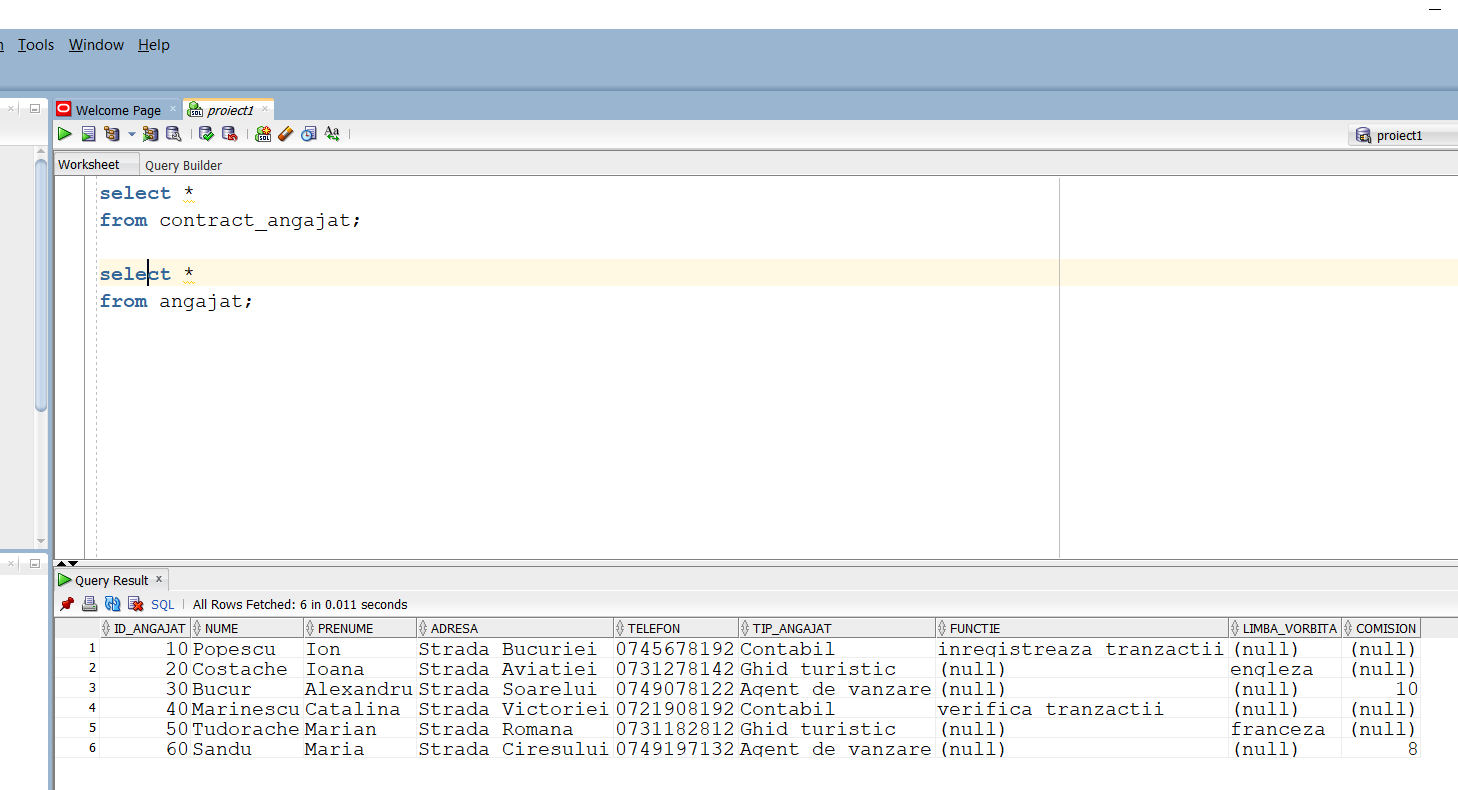
insert into ANGAJAT

values (SEQ\_ANG.NEXTVAL, 'Sandu', 'Maria', 'Strada Ciresului', '0749197132', 'Agent de vanzare', NULL, NULL, 8);

select \*

from angajat;

commit;



create table CONTRACT\_ANGAJAT

( id\_contract number(5) constraint pk\_contract primary key,

id\_angajat number(5) not null constraint fkey\_ang references ANGAJAT(id\_angajat),

data\_angajarii date default sysdate ,

experienta number(3),

salariu number(10) not null,

constraint ang unique(id\_angajat)

);

insert into CONTRACT\_ANGAJAT

values (1, 10, to\_date('20-06-2015','dd-mm-yyyy'), 10, 4500);

insert into CONTRACT\_ANGAJAT

values (2, 20, to\_date('18-06-2018','dd-mm-yyyy'), 5, 3800);

insert into CONTRACT\_ANGAJAT

values (3, 30, to\_date('08-11-2000','dd-mm-yyyy'), 20, 6500);

insert into CONTRACT\_ANGAJAT

values (4, 40, to\_date('12-03-2005','dd-mm-yyyy'), 15, 5000);

insert into CONTRACT\_ANGAJAT

values (5, 50, to\_date('18-05-2021','dd-mm-yyyy'), NULL, 3000);

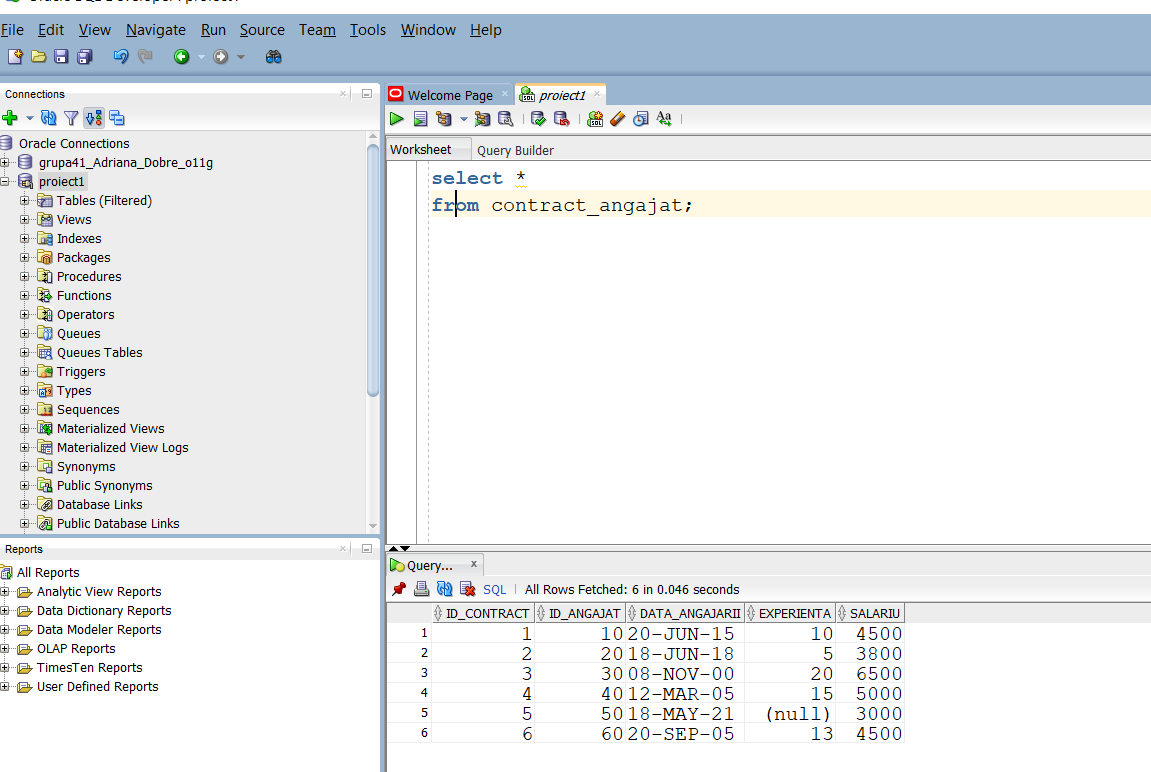
insert into CONTRACT\_ANGAJAT

values (6, 60, to\_date('20-09-2005','dd-mm-yyyy'), 13, 4500);

select \*

from contract\_angajat;

commit;



create table CLIENT

( id\_client number(5) constraint pk\_client primary key,

nume varchar2(20) constraint nume\_client not null,

prenume varchar2(20) constraint prenume\_client not null,

CNP varchar2(20) not null,

telefon varchar2(20) not null,

adresa\_email varchar(40) not null,

constraint cnp\_unic unique(CNP),

constraint nr\_tel unique(telefon),

constraint email unique(adresa\_email),

constraint nume\_pren\_unic unique(nume, prenume)

);

CREATE SEQUENCE SEQ\_CL

INCREMENT by 1

START WITH 100

MAXVALUE 10000

NOCYCLE;

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Badea', 'Radu', '1851021345131', '0739069122', 'badea.radu22@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Rebegea', 'Tiberiu', '1920617149053', '0746069782', 'rebegea.tiberiu22@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Oprea', 'Sofia', '1870505168646', '0732156278', 'oprea.sofia@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Iancu', 'Lorena', '1870619152998', '0749068132', 'lorena.iancu@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Ene', 'Robert', '1930729213031', '0728114281', 'ene.robert01@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Stoica', 'Ruxandra', '1900806182888', '0728567102', 'ruxandra.stoica@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Preda', 'Bianca', '1860114313017', '074128934', 'bianca.preda@yahoo.com');

insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Parvu', 'Vlad', '1921113049026', '0748901283', 'vlad.parvu@yahoo.com');

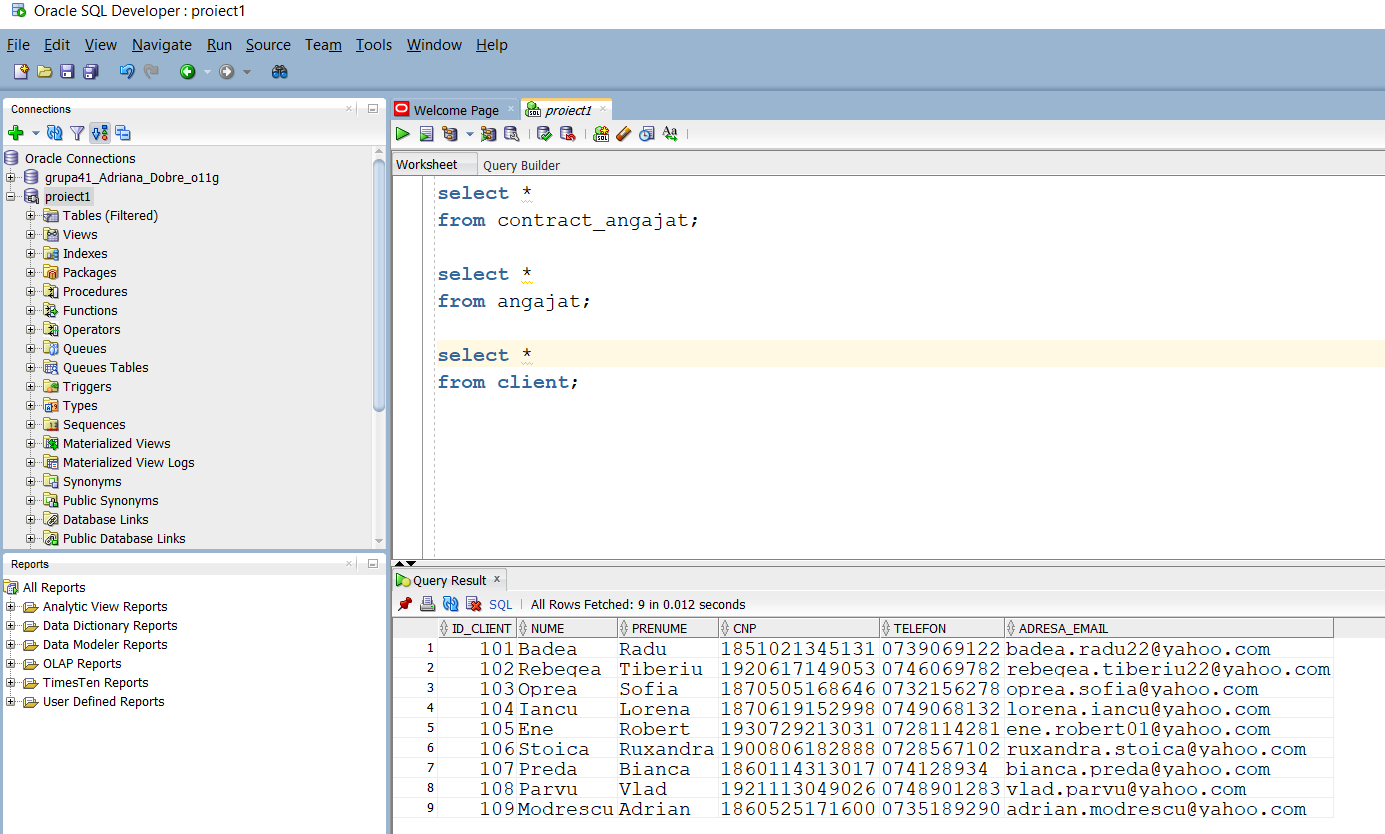
insert into CLIENT

values (SEQ\_CL.NEXTVAL, 'Modrescu', 'Adrian', '1860525171600', '0735189290', 'adrian.modrescu@yahoo.com');

select \*

from client;

commit;



create table HOTEL

( id\_hotel number(5) constraint pk\_hotel primary key,

nume\_hotel varchar2(20) not null,

nr\_de\_contact varchar2(20) not null,

nr\_stele number(10),

constraint nr\_contact unique(nr\_de\_contact)

);

insert into HOTEL

values (1, 'Aqua', '0244173384', 4);

insert into HOTEL

values (2, 'Astoria', '0244378414', 3);

insert into HOTEL

values (3, 'Golden', '0244970084', 5);

insert into HOTEL

values (4, 'Intercontinental', '0244571184', 4);

insert into HOTEL

values (5, 'Palace', '0244026352', 5);

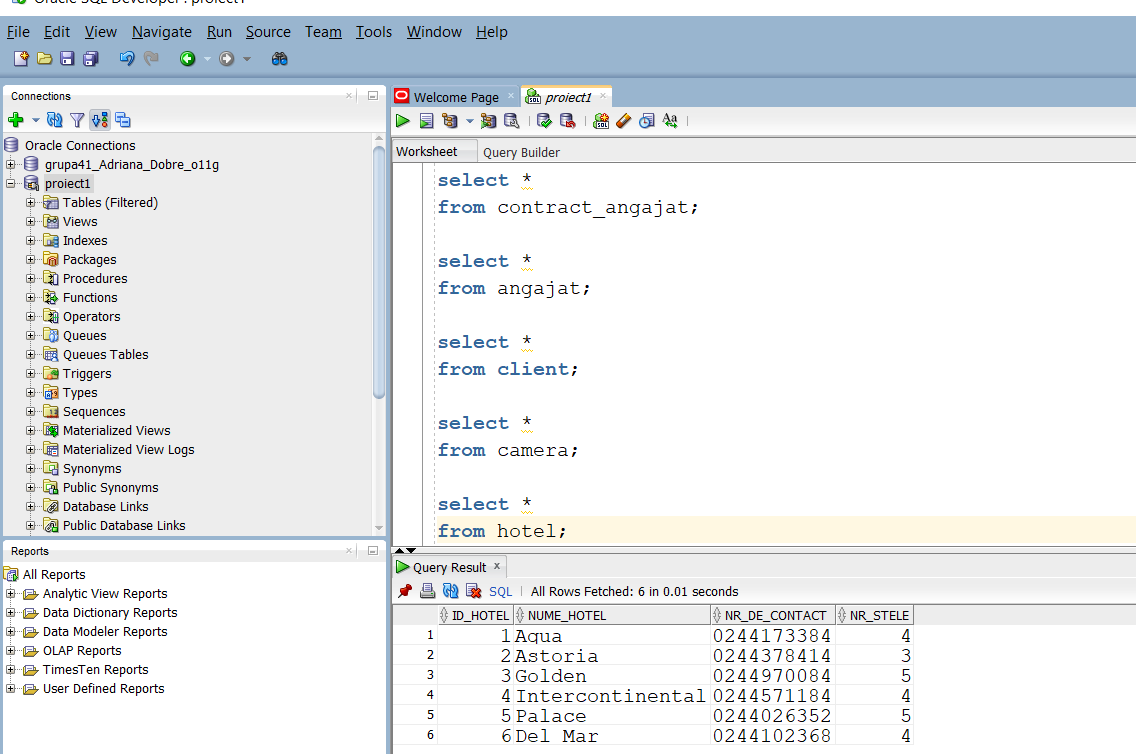
insert into HOTEL

values (6, 'Del Mar', '0244102368', 4);

select \*

from hotel;

commit;



create table CAMERA

( id\_camera number(5) constraint pk\_camera primary key,

id\_hotel number(5) not null constraint fkey\_hotel references HOTEL(id\_hotel),

nr\_camera number(5) not null,

etaj number(5),

capacitate varchar2(20)

);

CREATE SEQUENCE SEQ\_CAM

INCREMENT by 1

START WITH 200

MAXVALUE 10000

NOCYCLE;

insert into CAMERA

values (SEQ\_CAM.NEXTVAL,1 , 301, 1, 'dubla');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 1, 302, 1, 'standard');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 1, 303, 1, 'dubla');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 2, 301, 2, 'tripla');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 3, 310, 4, 'dubla');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 4, 340, 5, 'tripla');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 5, 313, 3, 'tripla');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 6, 321, 8, 'matrimoniala');

insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 3, 317, 7, 'dubla');

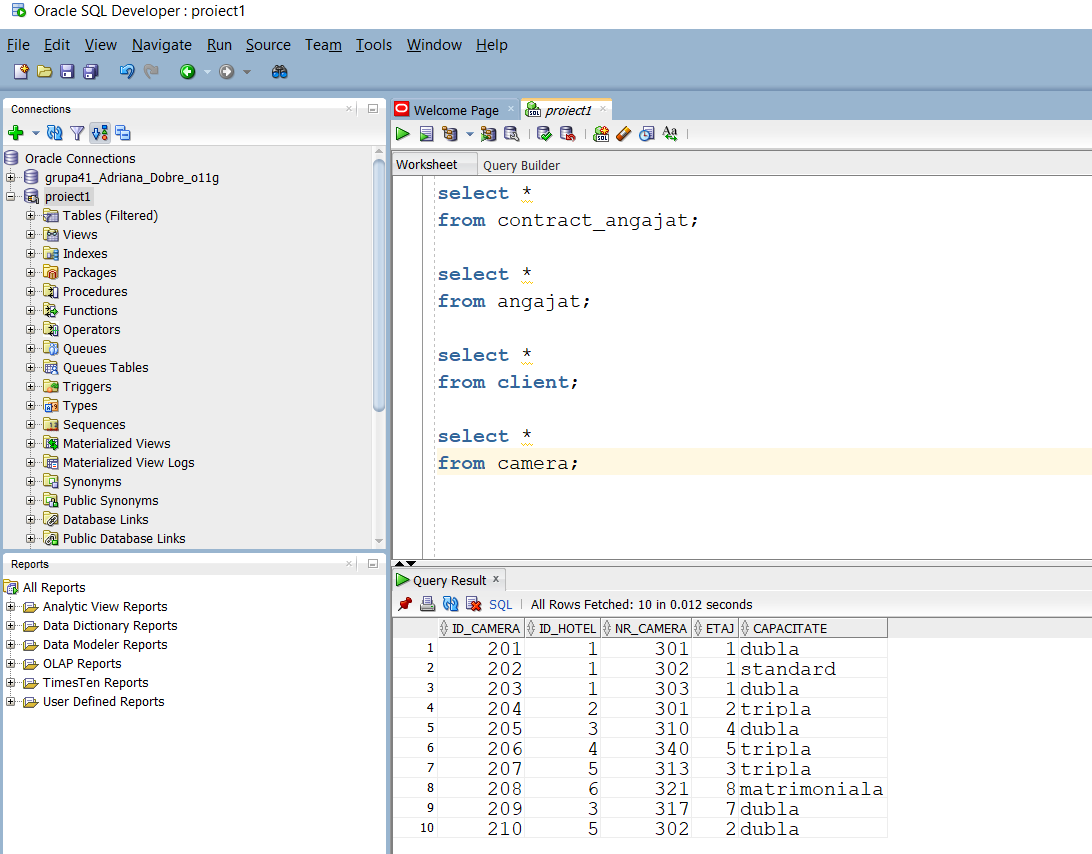
insert into CAMERA

values (SEQ\_CAM.NEXTVAL, 5, 302, 2, 'dubla');

select \*

from camera;

commit;



create table FACILITATI

( id\_facilitate number(5) constraint pk\_fac primary key,

denumire varchar2(40) not null

);

insert into FACILITATI

values (1, 'balcon');

insert into FACILITATI

values (2, 'aer conditionat');

insert into FACILITATI

values (3, 'frigider');

insert into FACILITATI

values (4, 'pat suplimentar');

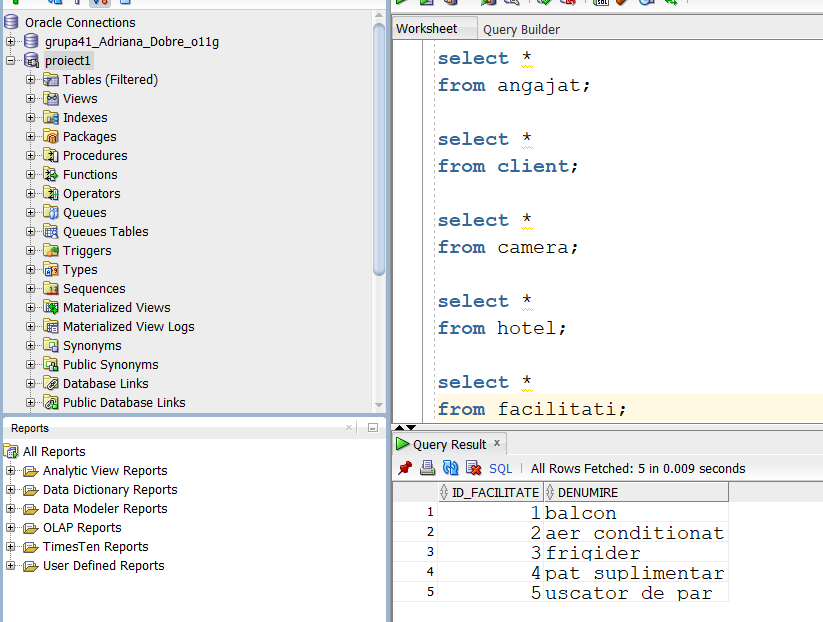
insert into FACILITATI

values (5, 'uscator de par');

select \*

from facilitati;

commit;



create table CONTINE\_FAC

(

id\_camera number(5) constraint fk\_cam references CAMERA(id\_camera),

id\_facilitate number(5) constraint fk\_fac references FACILITATI(id\_facilitate),

pret\_suplimentar number(5),

constraint pk\_compus2 primary key(id\_camera, id\_facilitate)

);

insert into CONTINE\_FAC

values (201, 1, null);

insert into CONTINE\_FAC

values (201, 2, null);

insert into CONTINE\_FAC

values (203, 3, null);

insert into CONTINE\_FAC

values (202, 4, 200);

insert into CONTINE\_FAC

values (206, 5, null);

insert into CONTINE\_FAC

values (201, 3, null);

insert into CONTINE\_FAC

values (203, 2, null);

insert into CONTINE\_FAC

values (208, 2, null);

insert into CONTINE\_FAC

values (210, 1, null);

insert into CONTINE\_FAC

values (207, 5, null);

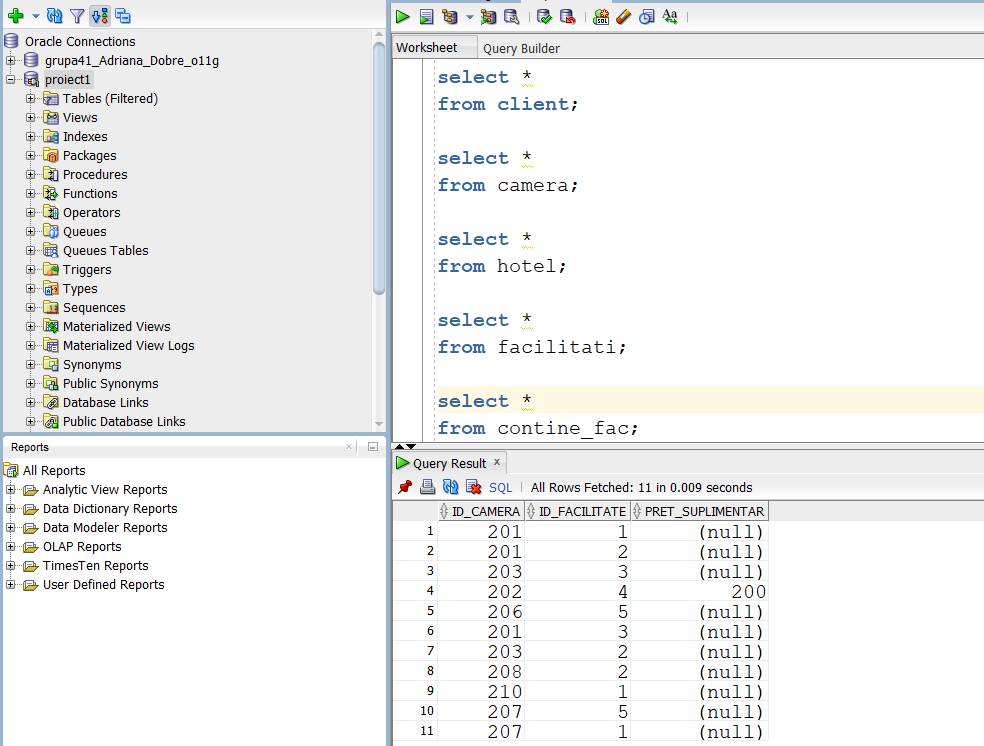
insert into CONTINE\_FAC

values (207, 1, null);

select \*

from contine\_fac;

commit;



create table DESTINATIE

(id\_destinatie number(5) constraint pk\_destinatie primary key,

oras varchar2(20) not null,

tara varchar(20) not null,

continent varchar2(20) not null

);

insert into DESTINATIE

values (1, 'Constanta', 'Romania','Europa');

insert into DESTINATIE

values (2, 'Paris', 'Franta','Europa');

insert into DESTINATIE

values (3, 'Barcelona', 'Spania','Europa');

insert into DESTINATIE

values (4, 'Venetia', 'Italia','Europa');

insert into DESTINATIE

values (5, 'Brasov', 'Romania','Europa');

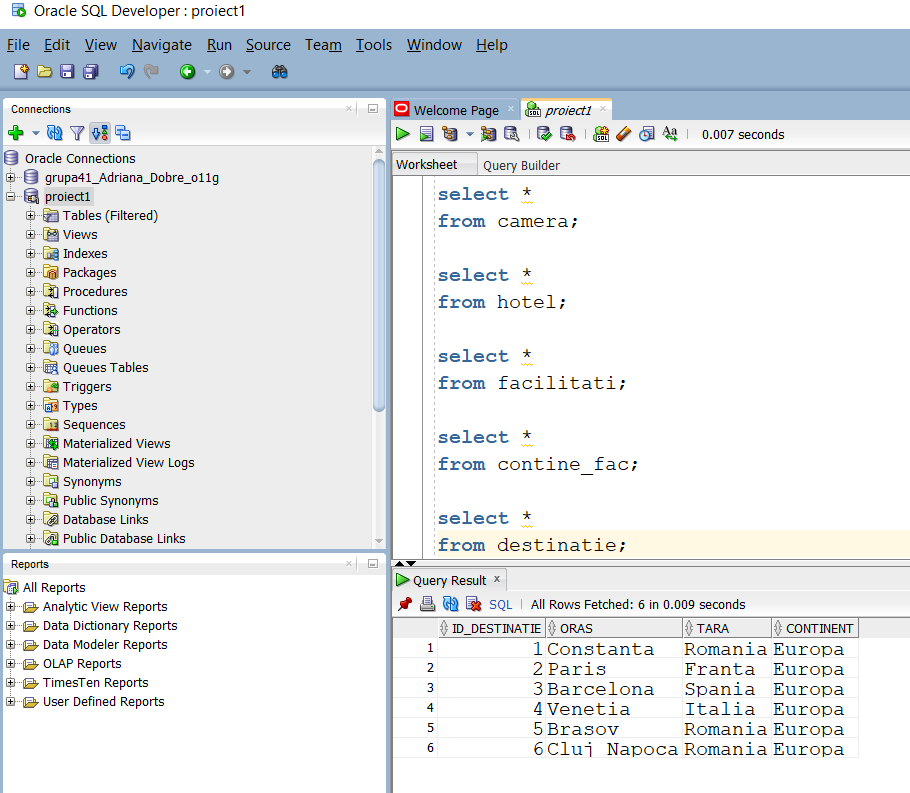
insert into DESTINATIE

values (6, 'Cluj Napoca','Romania', 'Europa');

select \*

from destinatie;

commit;



create table FIRMA

(id\_firma number(5) constraint pk\_firma primary key,

denumire\_firma varchar2(20) not null,

sediu varchar2(20),

telefon varchar(20) not null,

constraint t\_unic unique (telefon)

);

insert into FIRMA

values (1, 'Matrix Transport', 'Bucuresti','0742102828');

insert into FIRMA

values (2, 'Mateo Tour', 'Bucuresti','0743792828');

insert into FIRMA

values (3, 'Air Bucharest', 'Bucuresti','0742100815');

insert into FIRMA

values (4, 'AVX Auto', 'Bucuresti','0722101228');

insert into FIRMA

values (5, 'Blue Air', 'Bucuresti','0736725278');

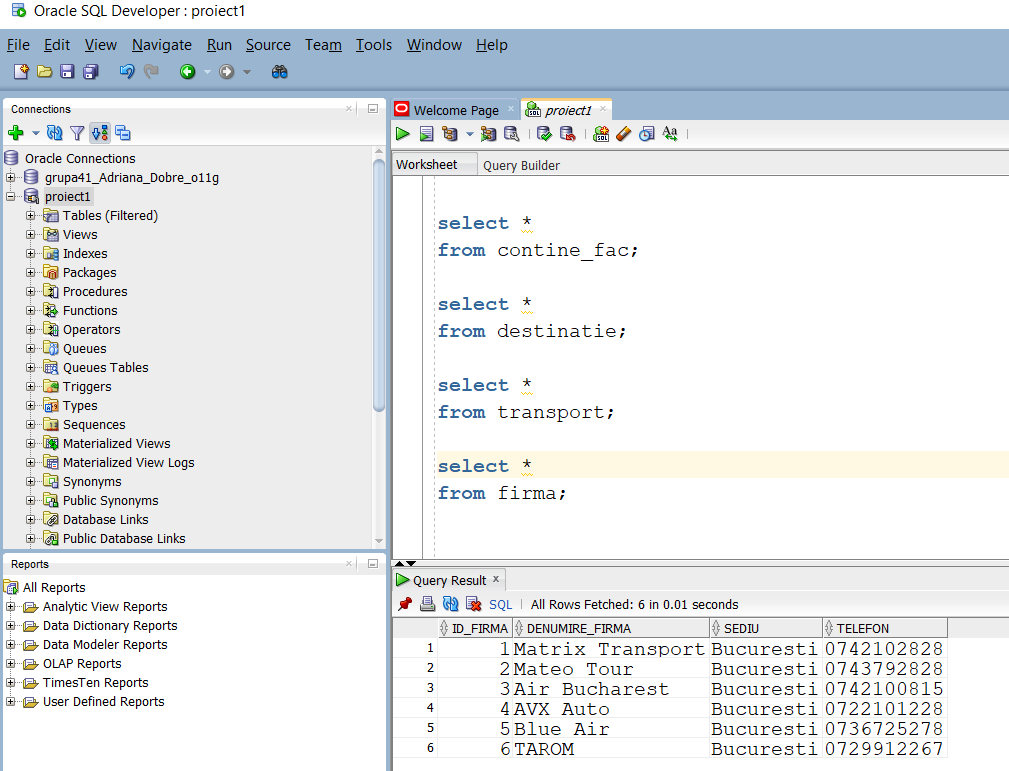
insert into FIRMA

values (6, 'TAROM', 'Bucuresti','0729912267');

select \*

from firma;

commit;



create table TRANSPORT

(id\_transport number(5) constraint pk\_transport primary key,

id\_firma number(5) not null constraint fk\_firma references FIRMA(id\_firma),

mijloc\_de\_transport varchar2(20) not null,

durata\_max number(10),

durata\_min number(10),

constraint max\_min check(durata\_max>=durata\_min)

);

CREATE SEQUENCE SEQ\_TR

INCREMENT by 10

START WITH 10

MAXVALUE 10000

NOCYCLE;

insert into TRANSPORT

values (SEQ\_TR.NEXTVAL, 2 , 'autocar' , 6, 4);

insert into TRANSPORT

values (SEQ\_TR.NEXTVAL, 5 , 'avion' , 4, 3);

insert into TRANSPORT

values (SEQ\_TR.NEXTVAL, 1 , 'autocar' , 17, 15);

insert into TRANSPORT

values (SEQ\_TR.NEXTVAL, 4 , 'autoturism' , 3, 2);

insert into TRANSPORT

values (SEQ\_TR.NEXTVAL, 3 , 'avion' , 3, 2);

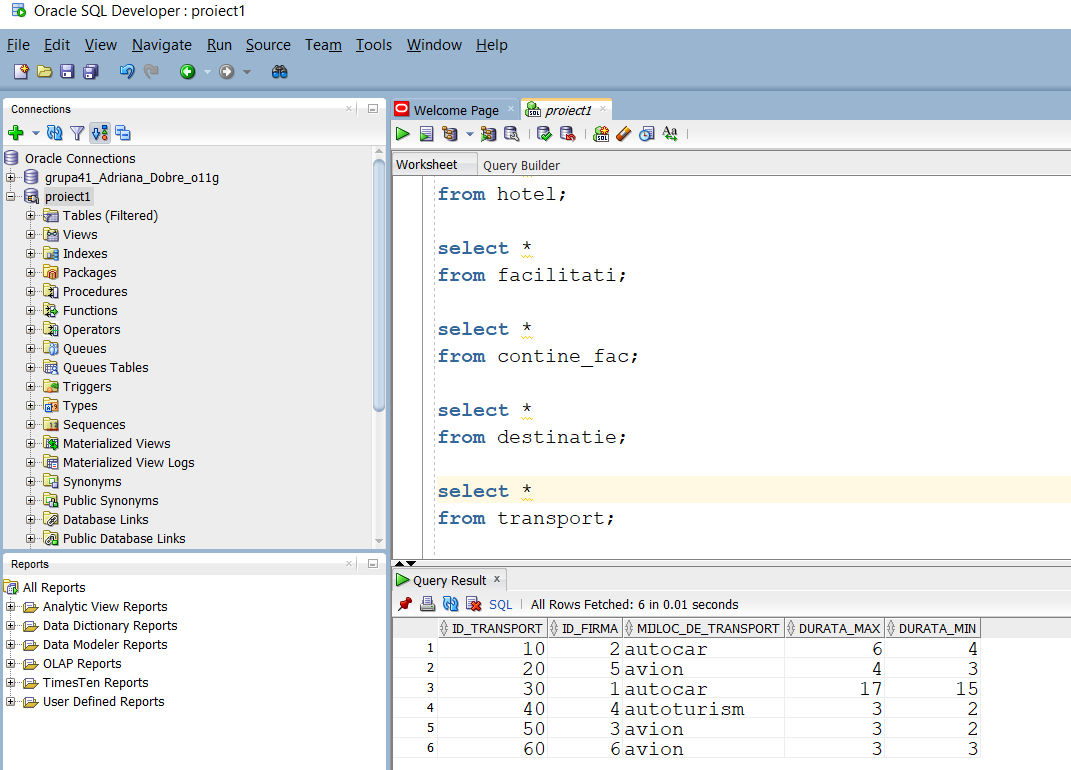
insert into TRANSPORT

values (SEQ\_TR.NEXTVAL, 6 , 'avion' , 3, 3);

select \*

from transport;

commit;



create table OFERTA

( id\_oferta number(5) constraint pk\_oferta primary key,

tip\_oferta varchar2(30) not null,

pret number(10) not null,

id\_destinatie number(5) not null constraint fk\_dest references DESTINATIE(id\_destinatie),

id\_transport number(5) constraint fk\_tr references TRANSPORT(id\_transport)

);

insert into OFERTA

values (100, 'sejur', 5000,1,10);

insert into OFERTA

values (200, 'sejur', 3800,1,null);

insert into OFERTA

values (300, 'sejur', 7000,2,20);

insert into OFERTA

values (400, 'sejur', 9500,4,50);

insert into OFERTA

values (500, 'sejur', 3200,1,null);

insert into OFERTA

values (600, 'sejur', 4500,1,null);

insert into OFERTA

values (700, 'sejur', 9500,3,60);

insert into OFERTA

values (800, 'sejur', 5500,1,60);

insert into OFERTA

values (900, 'city break', 3000,1,null);

insert into OFERTA

values (1000, 'sejur', 5000,6,30);

insert into OFERTA

values (1100, 'sejur', 8600,5,40);

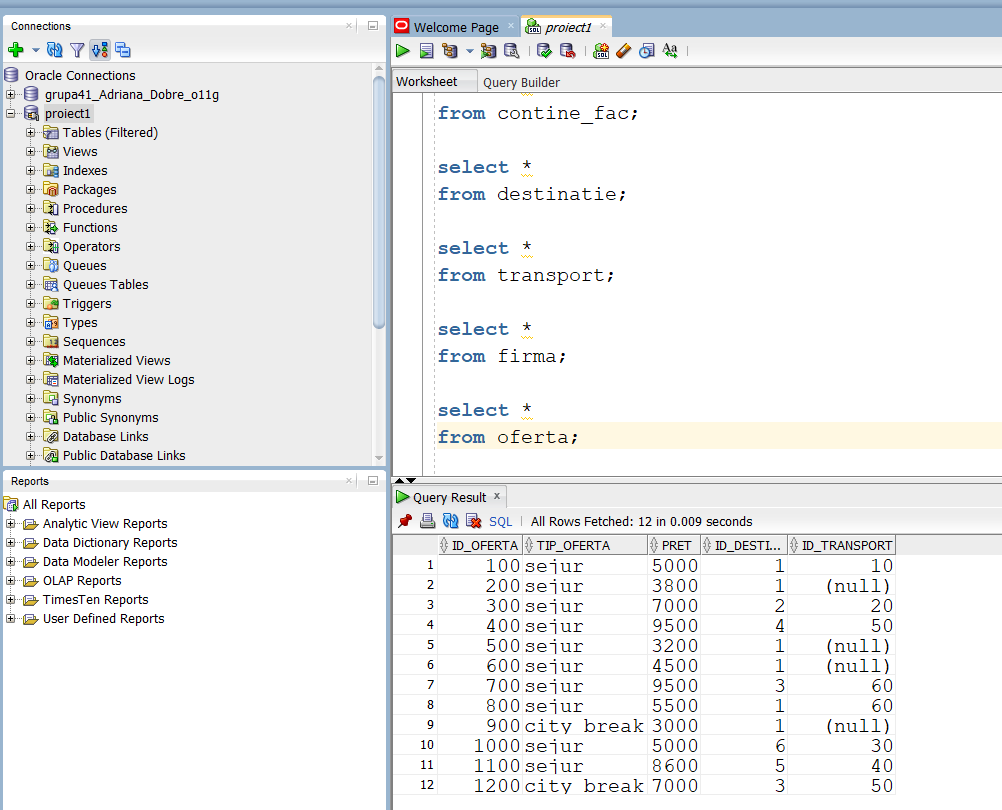
insert into OFERTA

values (1200, 'city break', 7000,3,50);

select \*

from oferta;

commit;



create table OBTINE

(id\_client number(5) constraint fk\_ct references CLIENT(id\_client),

id\_angajat number(5) constraint fk\_a references ANGAJAT(id\_angajat),

id\_oferta number(5) constraint fk\_of references OFERTA(id\_oferta),

id\_camera number(5) constraint fk\_cam1 references CAMERA(id\_camera),

data\_start date default sysdate,

data\_end date default sysdate,

constraint pk\_compus3 primary key(id\_client, id\_angajat, id\_oferta, id\_camera),

constraint start\_end check(data\_end>=data\_start)

);

insert into OBTINE

values (101, 30, 100,203, to\_date('10-06-2020','dd-mm-yyyy'), to\_date('16-06-2020','dd-mm-yyyy'));

insert into OBTINE

values (101, 60, 500, 202,to\_date('20-05-2021','dd-mm-yyyy'), to\_date('24-05-2021','dd-mm-yyyy'));

insert into OBTINE

values (103, 60, 600, 203, to\_date('20-07-2020','dd-mm-yyyy'), to\_date('26-07-2020','dd-mm-yyyy'));

insert into OBTINE

values (102, 60, 200, 201, to\_date('12-07-2020','dd-mm-yyyy'), to\_date('18-07-2020','dd-mm-yyyy'));

insert into OBTINE

values (107, 60, 800, 203, to\_date('10-07-2020','dd-mm-yyyy'), to\_date('16-07-2020','dd-mm-yyyy'));

insert into OBTINE

values (104, 30, 700, 205, to\_date('10-08-2020','dd-mm-yyyy'), to\_date('20-08-2020','dd-mm-yyyy'));

insert into OBTINE

values (105, 60, 300, 204, to\_date('28-07-2020','dd-mm-yyyy'), to\_date('02-08-2020','dd-mm-yyyy'));

insert into OBTINE

values (106, 30, 900, 208, to\_date('10-05-2021','dd-mm-yyyy'), to\_date('13-05-2021','dd-mm-yyyy'));

insert into OBTINE

values (107, 60, 1200, 206, to\_date('02-11-2020','dd-mm-yyyy'), to\_date('05-11-2020','dd-mm-yyyy'));

insert into OBTINE

values (109, 30, 1000, 207, to\_date('17-03-2021','dd-mm-yyyy'), to\_date('24-03-2021','dd-mm-yyyy'));

insert into OBTINE

values (108, 30, 1100, 206, to\_date('25-12-2020','dd-mm-yyyy'), to\_date('03-01-2021','dd-mm-yyyy'));

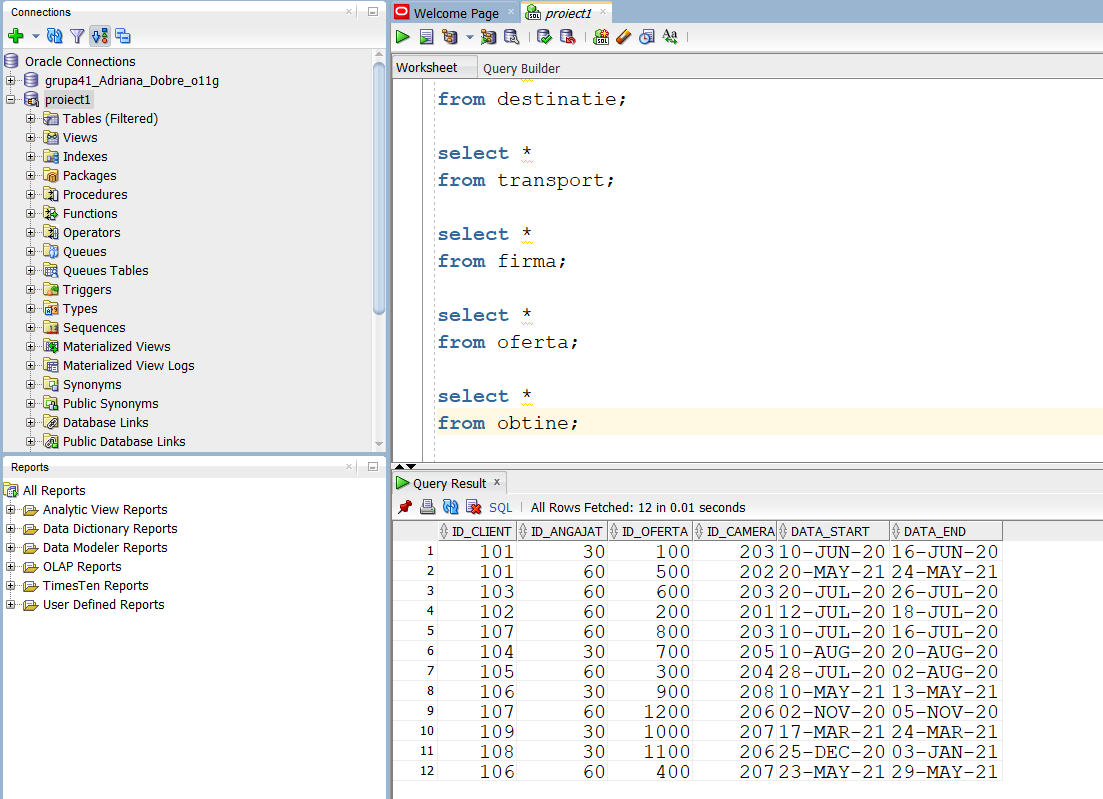
insert into OBTINE

values (106, 60, 400, 207, to\_date('23-05-2021','dd-mm-yyyy'), to\_date('29-05-2021','dd-mm-yyyy'));

select \*

from obtine;

commit;



**Exercițiul 11:**

---operație join pe cel puțin 4 tabele

---Sa se afiseze in ordine crescatoare toate detaliile (camera, destinatie,hotel) despre ofertele care incep din luna iulie obtinute de angajatul cu id-ul 60.

select o.id\_angajat, o.id\_oferta, tip\_oferta, pret, nume\_hotel,cam.id\_camera, nr\_camera,etaj, capacitate, data\_start, data\_end,oras, tara, continent

from obtine o join oferta ofe on (o.id\_oferta = ofe.id\_oferta)

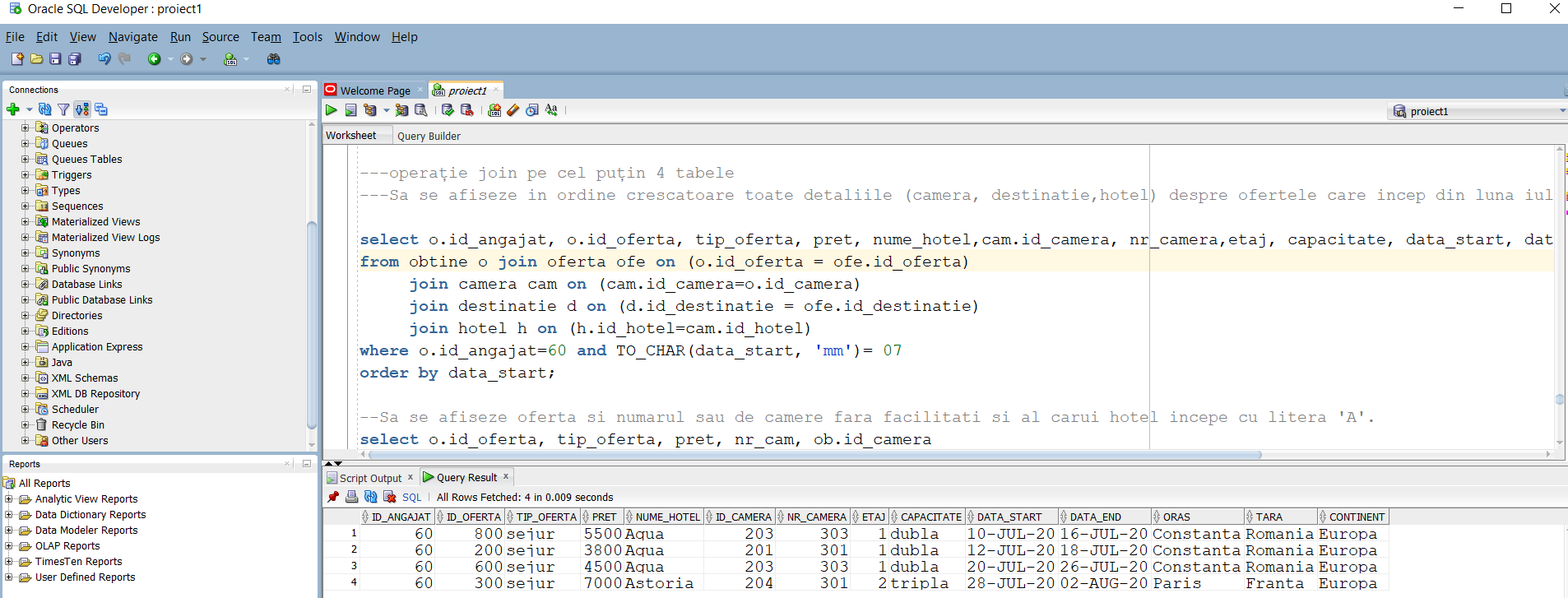
join camera cam on (cam.id\_camera=o.id\_camera)

join destinatie d on (d.id\_destinatie = ofe.id\_destinatie)

join hotel h on (h.id\_hotel=cam.id\_hotel)

where o.id\_angajat=60 and TO\_CHAR(data\_start, 'mm')= 07

order by data\_start;



--Sa se afiseze oferta si numarul sau de camere fara facilitati si al carui hotel incepe cu litera 'A'.

select o.id\_oferta, tip\_oferta, pret, nr\_cam, ob.id\_camera

from obtine ob, oferta o, camera c, hotel h,( select id\_camera,count(id\_camera) nr\_cam

from obtine

where id\_camera not in (select id\_camera

from contine\_fac

)

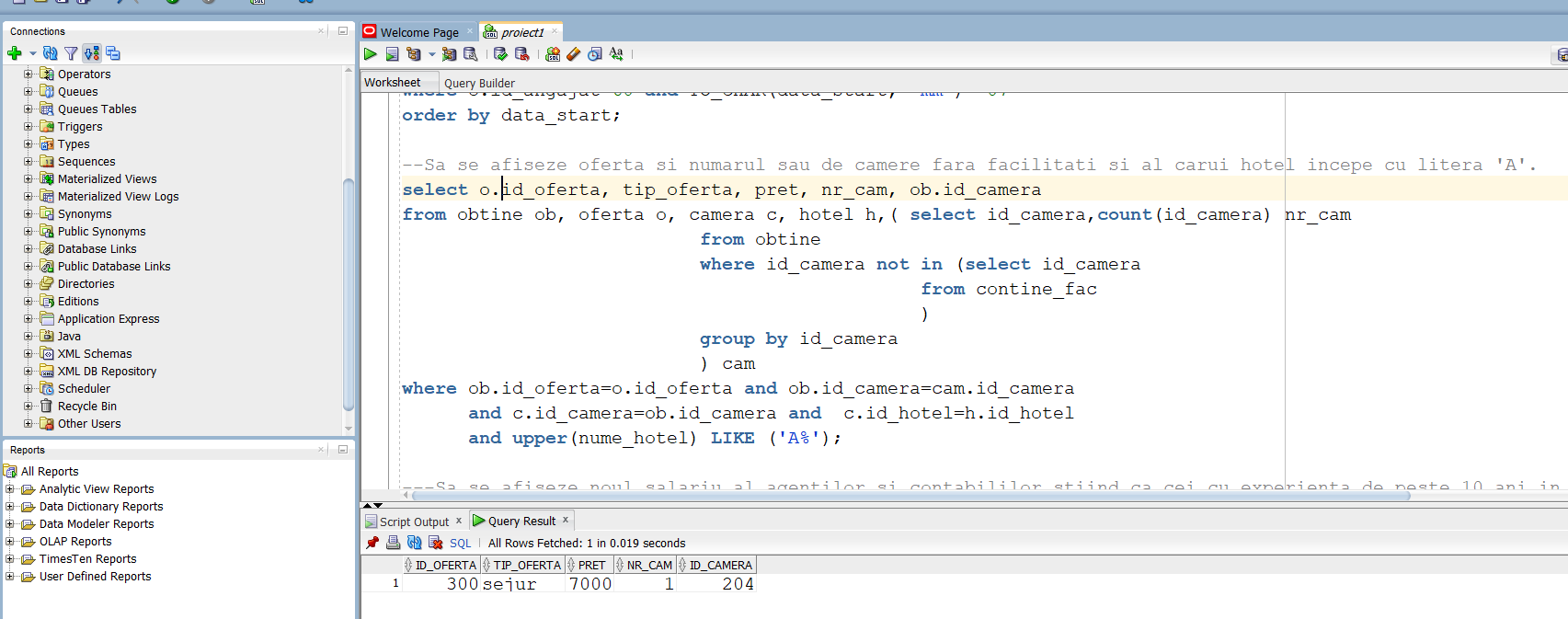
group by id\_camera

) cam

where ob.id\_oferta=o.id\_oferta and ob.id\_camera=cam.id\_camera

and c.id\_camera=ob.id\_camera and c.id\_hotel=h.id\_hotel

and upper(nume\_hotel) LIKE ('A%');



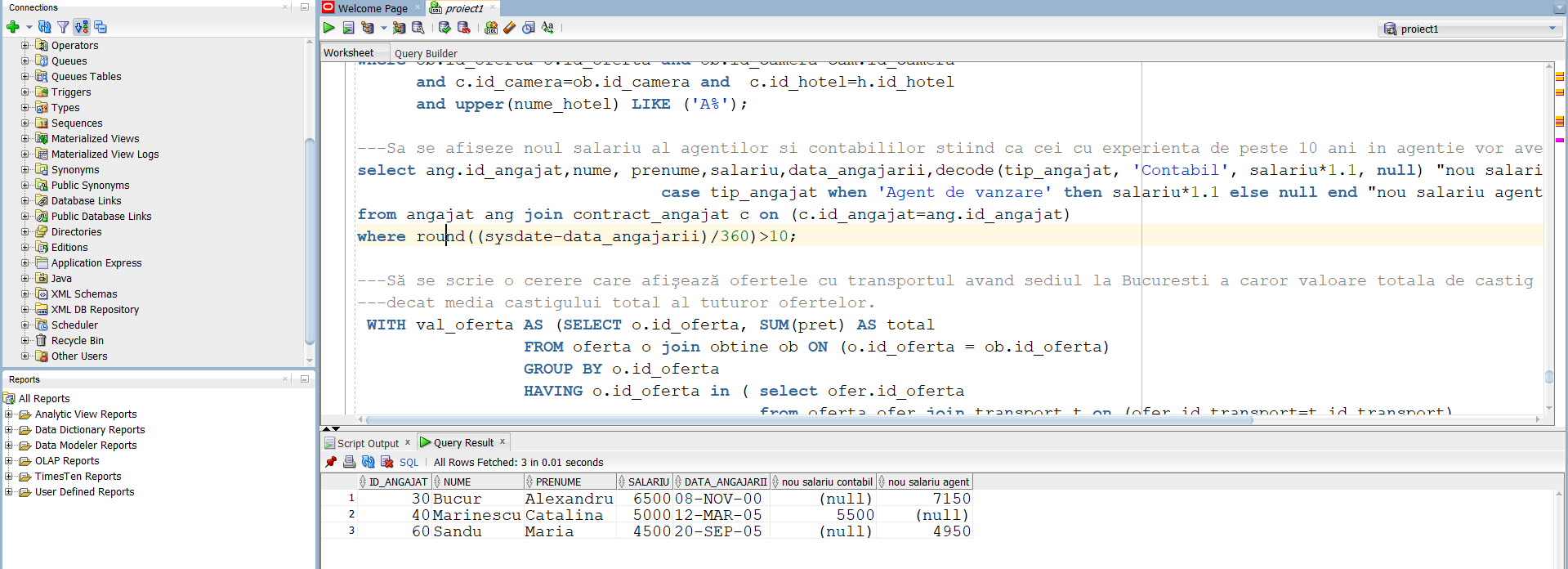
---Sa se afiseze noul salariu al agentilor si contabililor stiind ca cei cu experienta de peste 10 ani in agentie vor avea o marire de salariu de 10%.

select ang.id\_angajat,nume, prenume,salariu,data\_angajarii,decode(tip\_angajat, 'Contabil', salariu\*1.1, null) "nou salariu contabil",

case tip\_angajat when 'Agent de vanzare' then salariu\*1.1 else null end "nou salariu agent"

from angajat ang join contract\_angajat c on (c.id\_angajat=ang.id\_angajat)

where round((sysdate-data\_angajarii)/360)>10;



---Să se scrie o cerere care afişează ofertele cu transportul avand sediul la Bucuresti a caror valoare totala de castig este mai mare

---decat media castigului total al tuturor ofertelor.

WITH val\_oferta AS (SELECT o.id\_oferta, SUM(pret) AS total

FROM oferta o join obtine ob ON (o.id\_oferta = ob.id\_oferta)

GROUP BY o.id\_oferta

HAVING o.id\_oferta in ( select ofer.id\_oferta

from oferta ofer join transport t on (ofer.id\_transport=t.id\_transport)

where t.id\_firma = ( select id\_firma

from firma

where t.id\_firma=id\_firma and sediu='Bucuresti'))

),

val\_medie AS (SELECT SUM(total)/COUNT(total) AS medie

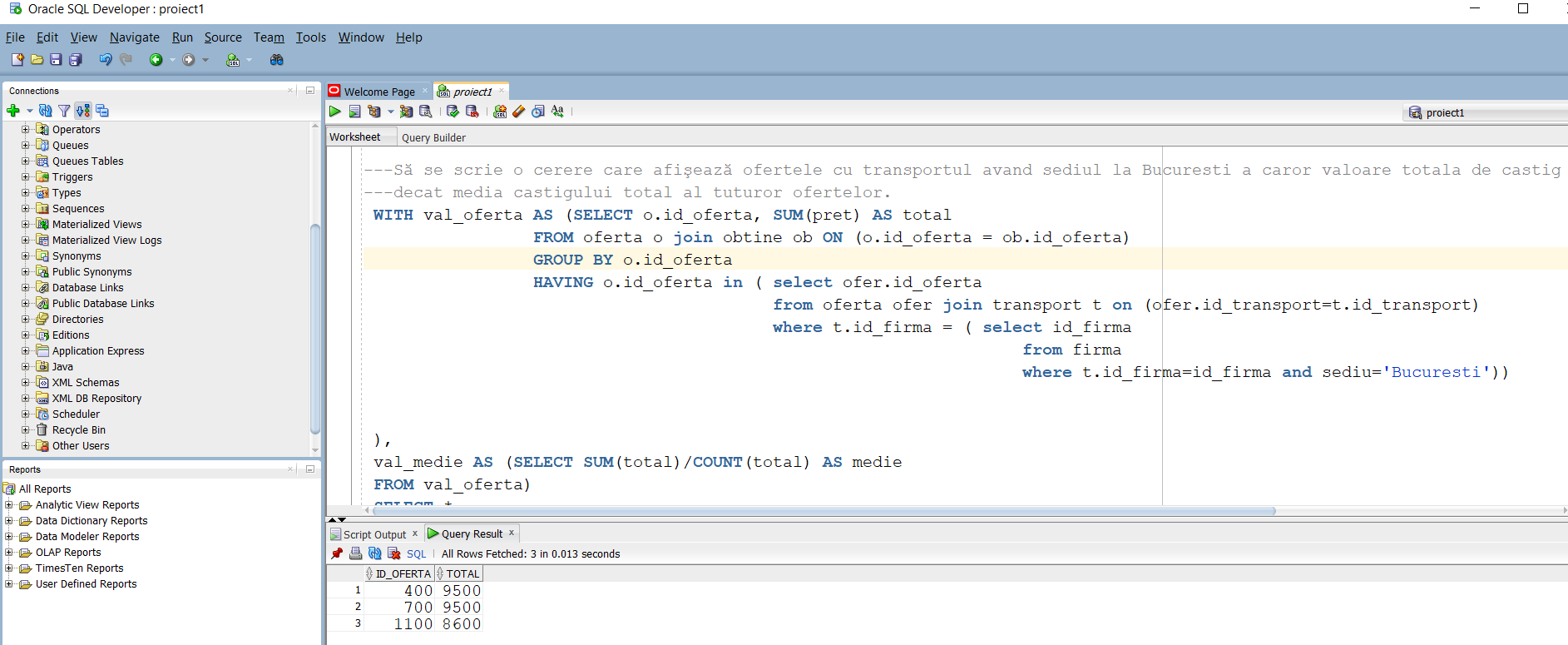
FROM val\_oferta)

SELECT \*

FROM val\_oferta

WHERE total > (SELECT medie

FROM val\_medie);



--Sa se afiseze oferta, pretul si numarul de clienti care au cumparat oferta respectiva in anul 2020, iar orasul in care se

---desfasoara oferta sa aiba a doua litera 'a'.

select o.id\_oferta, tip\_oferta, pret, NVL(nr\_cl,0)

from obtine ob, oferta o,destinatie d,( select id\_oferta, count(id\_client) nr\_cl

from obtine

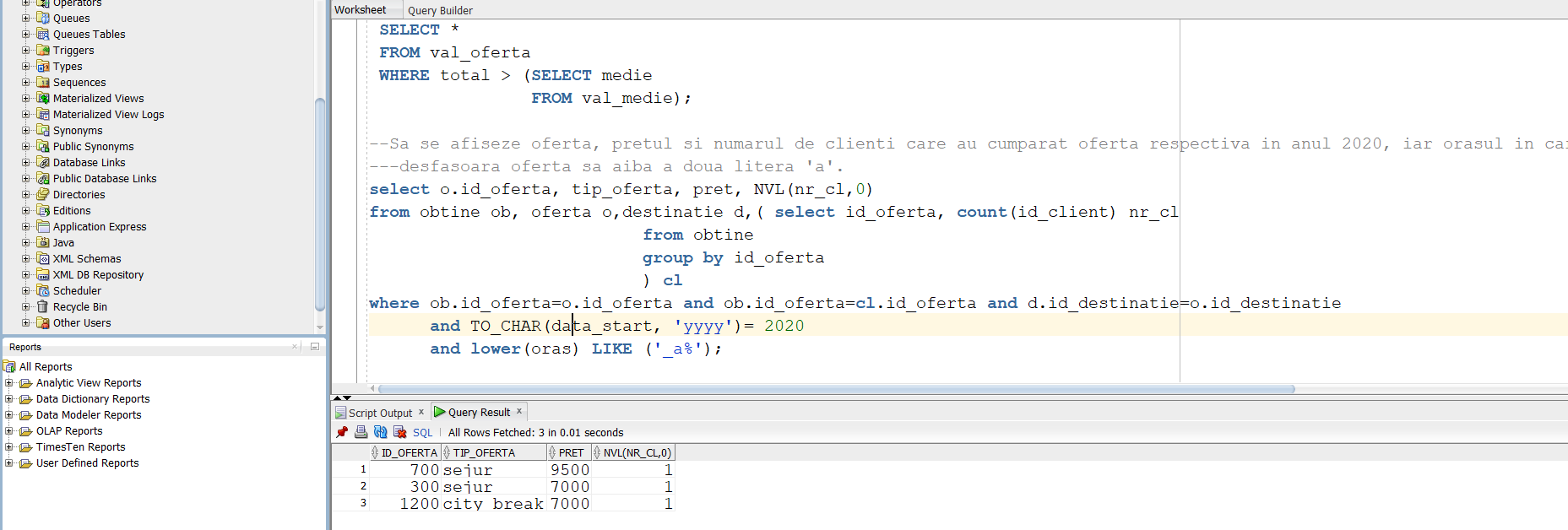
group by id\_oferta

) cl

where ob.id\_oferta=o.id\_oferta and ob.id\_oferta=cl.id\_oferta and d.id\_destinatie=o.id\_destinatie

and TO\_CHAR(data\_start, 'yyyy')= 2020

and lower(oras) LIKE ('\_a%');



**Exercițiul 12:**

--Sa se mareasca cu o ora durata maxima de intarziere a ofertelor cu destinatia Constanta.

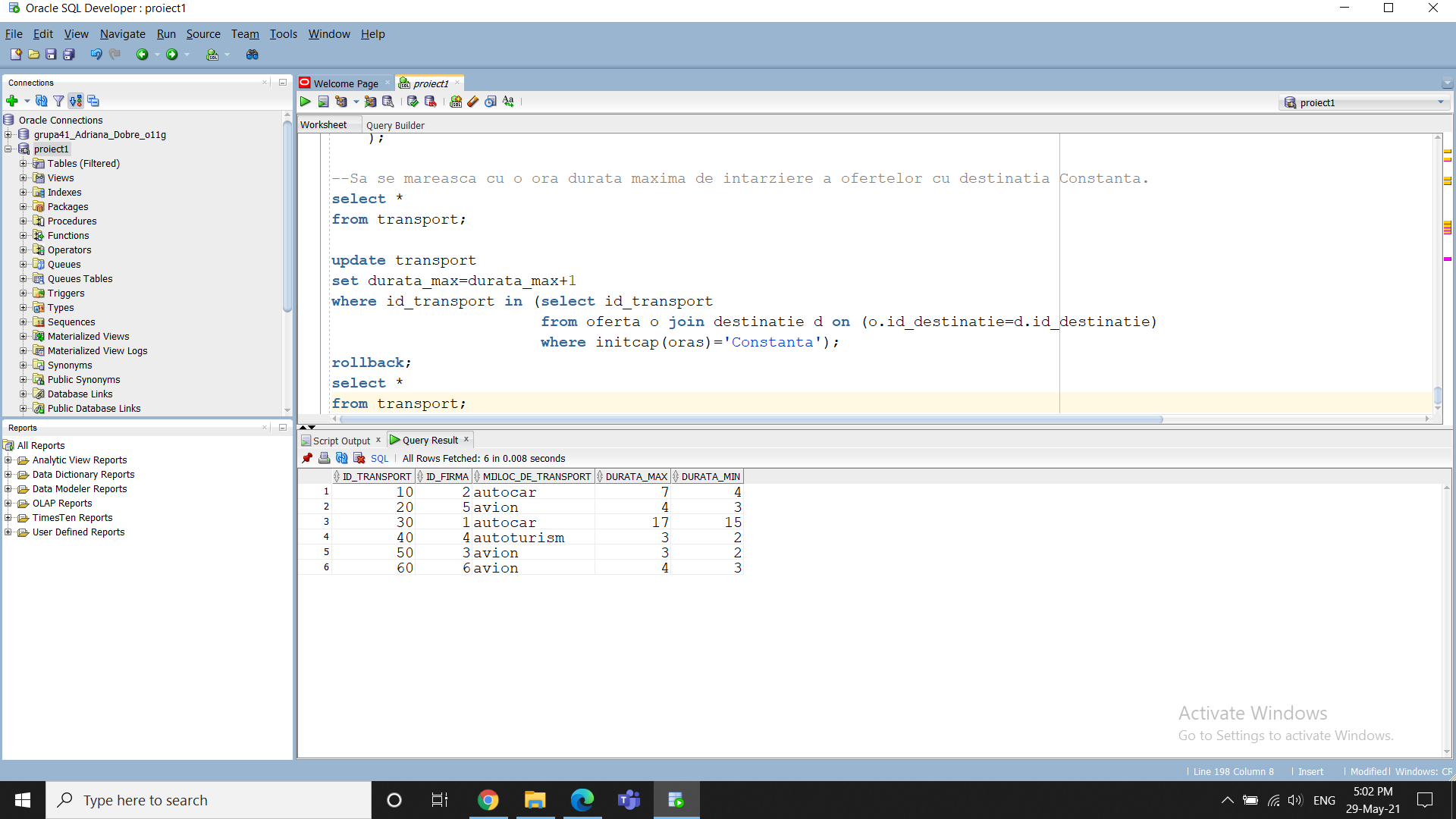
update transport

set durata\_max=durata\_max+1

where id\_transport in (select id\_transport

from oferta o join destinatie d on (o.id\_destinatie=d.id\_destinatie)

where initcap(oras)='Constanta');



--Sa se actualizeze pretul suplimentar la valoare de 20 pentru facilitatile pentru camerele de la etajul 1.

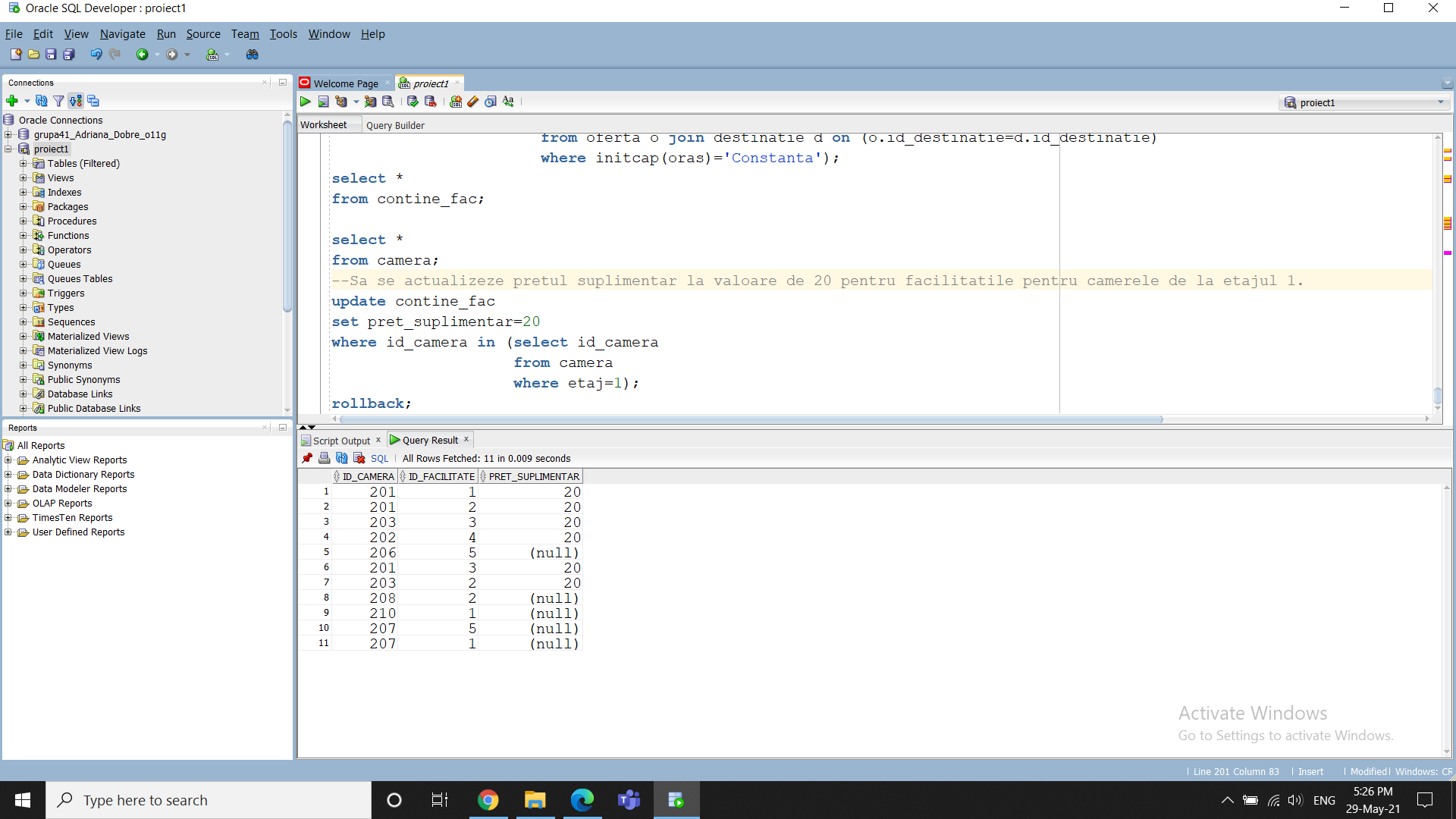
update contine\_fac

set pret\_suplimentar=20

where id\_camera in (select id\_camera

from camera

where etaj=1);



--Sa se stearga angajatii fara experienta.

--Pentru a putea face acest lucru trebuie sa stergem intai contractele angajatilor fara experienta.

delete

from contract\_angajat

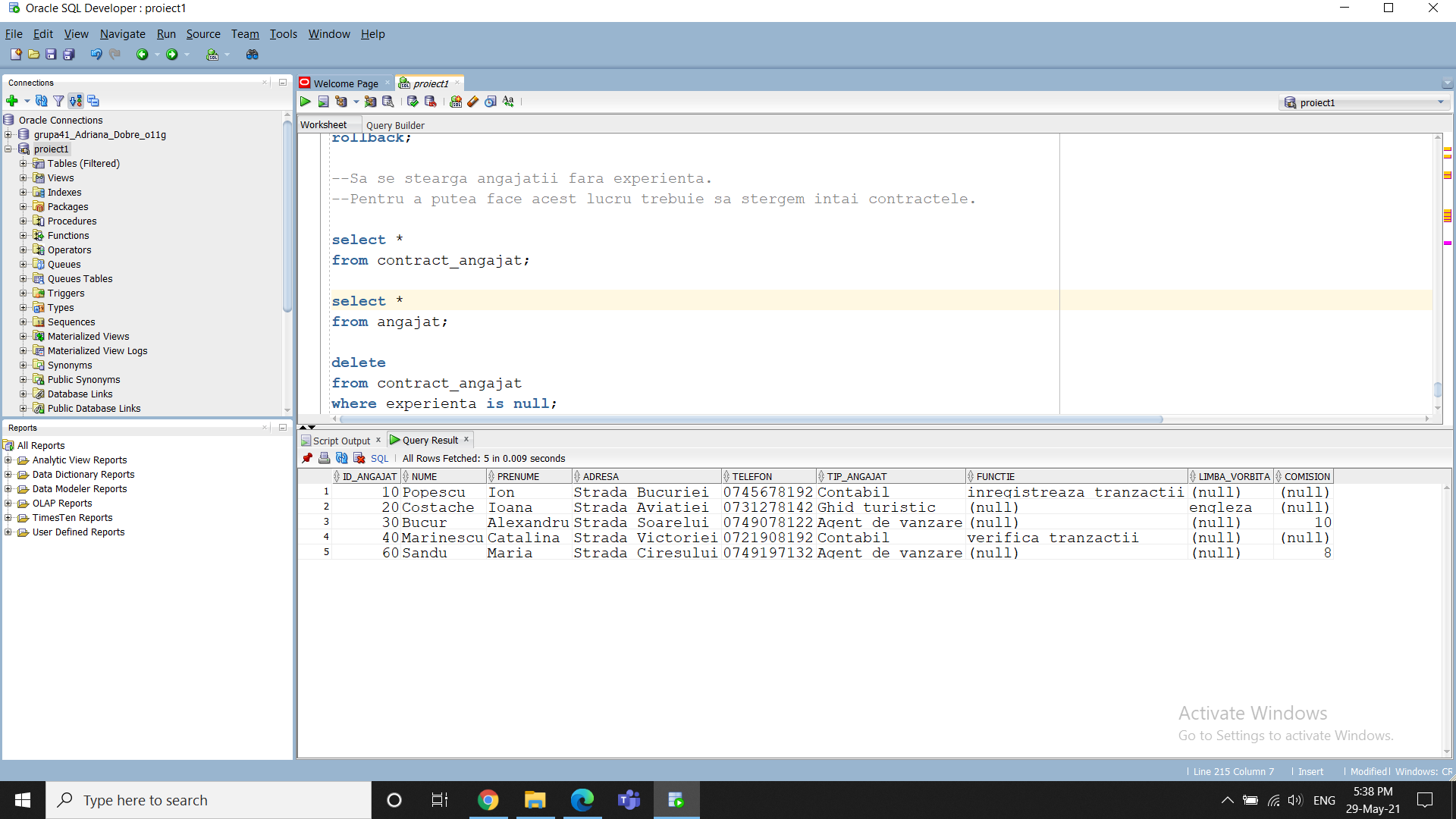
where experienta is null;

delete

from angajat

where id\_angajat not in (select id\_angajat

from contract\_angajat);



**Exercițiul 16:**

--Sa se afiseze toate ofertele obtinute de angajatul cu id-ul 60 in anul 2020. Se vor include si cele fara transport.

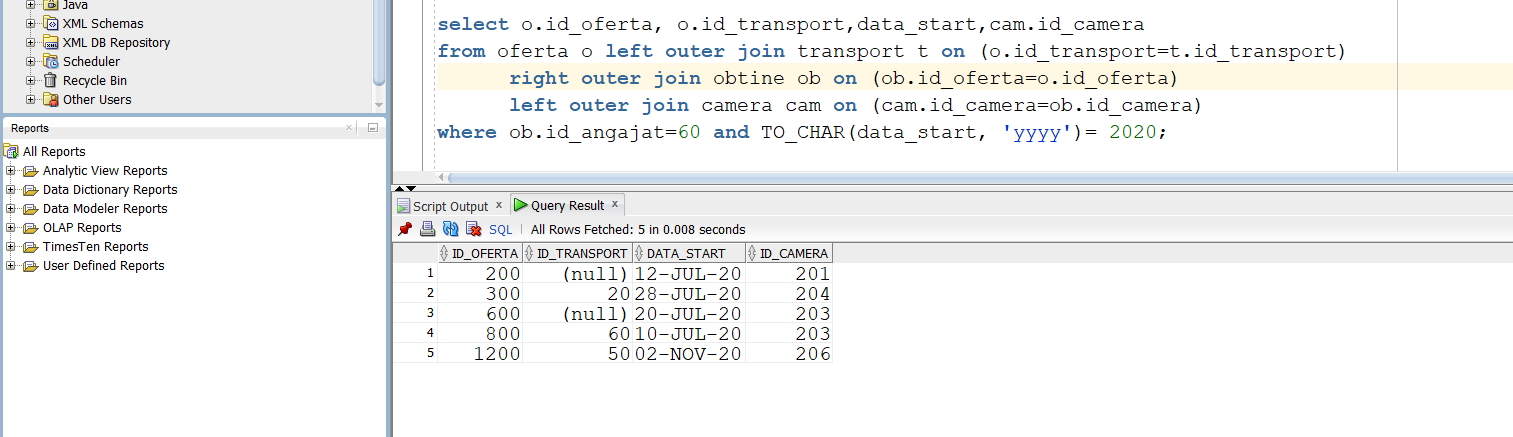
select o.id\_oferta, o.id\_transport,data\_start,cam.id\_camera

from oferta o left outer join transport t on (o.id\_transport=t.id\_transport)

right outer join obtine ob on (ob.id\_oferta=o.id\_oferta)

left outer join camera cam on (cam.id\_camera=ob.id\_camera)

where ob.id\_angajat=60 and TO\_CHAR(data\_start, 'yyyy')= 2020;



--Sa se afiseze clientul care a stat numai in camera cu numarul 203.

select id\_client

from obtine

where id\_camera in (select id\_camera

from camera

where id\_camera=203

)

MINUS

select id\_client

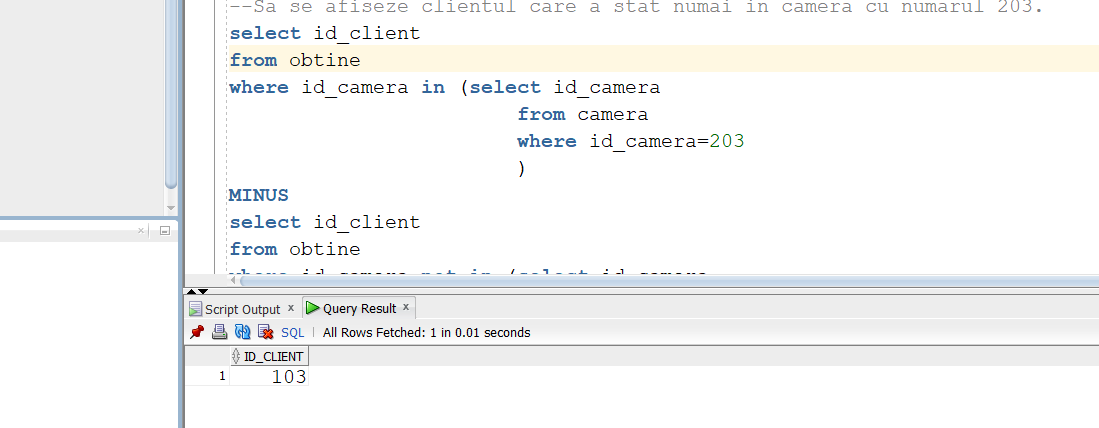
from obtine

where id\_camera not in (select id\_camera

from camera

where id\_camera=203

);



--Sa se afiseze camerele care au numai facilitatea 5.

select id\_camera

from contine\_fac

where id\_facilitate in (select id\_facilitate

from facilitati

where id\_facilitate=5

)

MINUS

select id\_camera

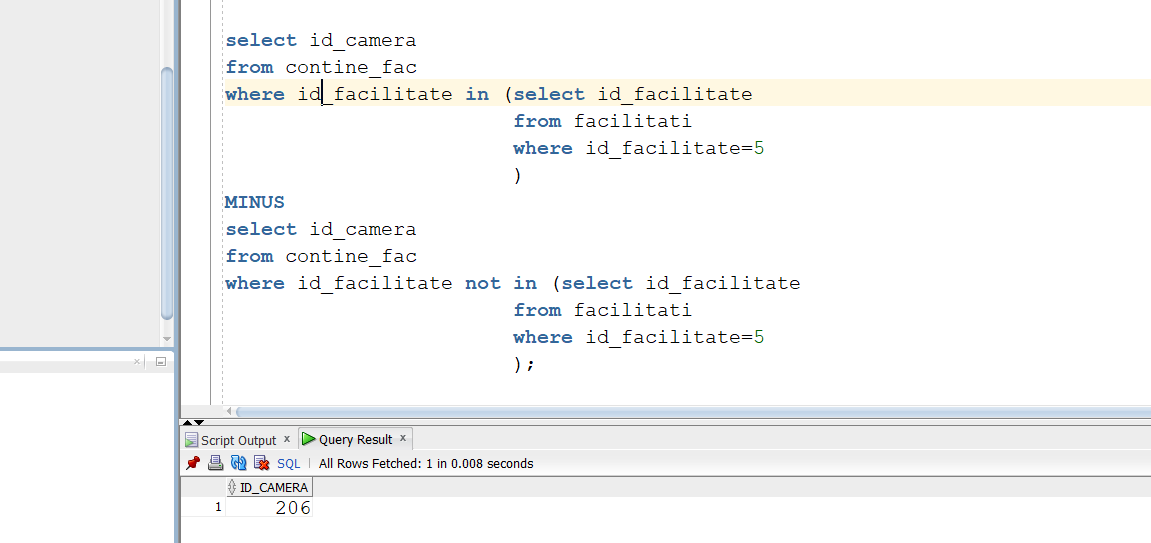
from contine\_fac

where id\_facilitate not in (select id\_facilitate

from facilitati

where id\_facilitate=5

);



**Exercițiul 17:**

Să se afișeze id-ul ofertelor și valoarea acestora care au ca mijloc de transport avionul și care încep de la data de 1 martie 2020 , iar camera inclusă în oferte se află la un etaj mai mare decat 2.

Cerere SQL:

select o.id\_oferta, pret

from oferta o join transport t on (o.id\_transport=t.id\_transport)

join obtine ob on (ob.id\_oferta=o.id\_oferta)

join camera cam on (cam.id\_camera=ob.id\_camera)

where mijloc\_de\_transport='avion' and to\_char(data\_start,'dd-mm-yyyy')>'01-03-2020'

and etaj>2;

Expresie algebrică:

R1=PROJECT(OFERTA,id\_oferta,pret)

R2=SELECT(TRANSPORT,mijloc\_de\_transport=’avion’)

R3=PROJECT(R2,id\_transport)

R4=SEMIJOIN(R1,R3)

R5=SELECT(OBTINE,data\_start>’01-03-2020’)

R6=PROJECT(R5,id\_oferta,id\_camera)

R7=JOIN(R4,R7)

R8=SELECT(CAMERA,etaj>2)

R9=PROJECT(R8,id\_camera)

R10=JOIN(R7,R9)

Rezultat=R11=PROJECT(R10,id\_oferta,id\_pret)

Arborele algebric:

