Baze de date 2022

Proiect final

**GESTIUNEA UNEI BAZE HOTELIERE**

Lect. Dr. Marin Letiția Ana

Student: Georgescu Elena- Alexandra

Seria 13, Grupa 131

Anul I, Semestrul II

1. Descrierea modelului

In cadrul acestui proiect vom analiza gestiunea unui lant hotelier. Modelul prezentat cuprinde mai multe hoteluri cu acelasi nume, situate in orase diferite ale Romaniei. Cu ajutorul bazei de date proiectata, putem tine o evidenta clara a angajatilor din toate hotelurile, a rezervarilor facute, a clientilor care calca pragul hotelurilor, a platilor si a facturilor efectuate pentru sejururi.

Fiecare angajat din cadrul hotelului are un job, un program de munca si un salariu bine definit.

Pretul unei camere/unui apartament variaza in functie de tipul acesteia/ acestuia.

Rezervarea cuprinde o serie de informatii esentiale: obiectul rezervat, persoana care a facut rezervarea, data rezervarii, perioada care se doreste rezervata, tipul de sejur pentru care se opteaza si factura emisa.

Factura furnizeaza informatii cu privire la plata efectuata si data efectuarii acesteia.

Un client isi poate alege tipul de serviciu dorit atat in functie de preferinte, cat si in functie de programul sau (clientul va alege dintre serviciile disponibile la ora dorita). De asemenea, serviciile pot lipsi din pachetul unui client, intrucat sunt optionale. In cazul alegerii unui serviciu, pretul acestuia va fi adaugat la pretul pe care il are de platit, iar serviciul va fi disponibil pe toata durata sederii clientului.

Mai jos voi anunta regulile de functionare.

* Clientii pot rezerva camere de tip single, double sau triple sau apartamente cu doua, trei sau patru camere;
* Camerele dintr-un apartament sunt exclusiv de tip double;
* Clientii pot alege mai multe tipuri de servicii dintre piscina, sala de sport si excursii organizate.
* Piscina si sala de sport sunt la dispozitia clientilor in fiecare zi intre orele de functionare corespunzatoare;
* Excursiile organizate au asigurate transportul;
* Clientii pot opta pentru unul dintre serviciile de masa: room-only, mic dejun, demipensiune (ce cuprinde mic dejun si cina), pensiune completa sau all-inclusive;
* La rezervare, clientii trebuie sa specifice data venirii si data parasirii hotelului;
* Valoarea de plata presupune suma preturilor dintre tipul camerei sau al apartamentului selectat multiplicat cu numarul noptilor petrecute la hotel, suma serviciilor selectate si pretul tipului de sejur dorit;
* In cazul platilor online, data de emitere a facturii va coincide cu data rezervarii. Altfel , va coincide cu data check-in-ului clientului in hotel;
* Rezervarile se fac, bineinteles, pe baza disponibilitatii camerelor si apartamentelor detinute de hotel;
* Salariul unui angajat este determinat strict de locul de munca pe care acesta il ocupa.

1. Constrangerile modelului

* Un angajat poate lucra intr-un singur hotel, unde are un singur job;
* Hotelurile sunt regasite in mai multe orase;
* Un apartament/ o camera este disponibil/a doar daca intre data ultimului check-out si data noului check-in exista cel putin o zi, ce este alocata pentru curatarea camerei, igienizarea si pregatirea ei pentru noii oaspeti;
* Un client poate opta pentru o excursie doar daca data excursiei se afla exclusiv intre data check-in-ului si data check-out-ului clientului din hotel;
* Pentru excursii pot fi folosite mai multe mijloace de transport(ce pot fi si diferite);
* Un client poate rezerva o singura camera, un singur apartament, sau o camera si un apartament printr-o rezervare;
* Clientul trebuie sa isi aleaga un singur tip de serviciu de masa pe durata intreaga a sejurului;
* Alegerea serviciilor extra este optionala: pot fi mai multe servicii extra alese, dar si niciunul.

1. Descrierea entitatilor

Entitatile modelului sunt urmatoarele:

HOTEL- entitatea de baza, tinand cont ca modelul prezentat reprezinta gestiunea unui lant hotelier. Are informatii privind locatia in care se afla respectivul hotel. Cheie primara: id\_hotel;

ANGAJAT- persoana fizica, ce lucreaza in cadrul hotelului. Aici gasim informatii referitoare la persoana angajata, cat si programul ei de lucru. Cheie primara: id\_angajat;

JOB- functia ce poate fi ocupata de un angajat. Cuprinde salariul pe care il obtine un angajat in aceasta pozitie. Cheie primara: id\_job;

CAMERA- loc unde se realizeaza cazarea clientilor pe parcursul sejururilor acestora. Contine informatii referitoare la tipul de camera, numarul camerei, pretul acesteia pe noapte si disponibilitatea acesteia. Cheie primara: id\_camera;

APARTAMENT- loc unde se realizeaza cazarea clientilor pe parcursul sejururilor acestora. Cuprinde informatii referitoare la numarul de camere din apartament, numarul apartamentului, pretul acestuia pe noapte si disponibilitatea acestuia. Cheie primara: id\_apartament;

TIP\_SEJUR- reprezinta serviciul de masa dorit de client pe perioada sejurului sau.

Cheie primara: id\_sejur;

CLIENT- persoana fizica, ce face rezervare pentru un sejur petrecut in cadrul hotelului. Cheie primara: id\_client;

SERVICII\_EXTRA- beneficii optionale de care se poate bucura un client pe parcursul sejurului sau, contra cost. Cheie primara: id\_serv;

MIJLOC\_TRANSPORT- entitate ce reprezinta modalitatea de transport a clientilor ce au optat pentru o excursie ca extra-serviciu. Cheie primara: id\_mijl;

FACTURA- obtinuta in urma unei rezervari, furnizand informatii despre data achitarii sejurului de catre client. Cheie primara: id\_factura;

PLATA- furnizeaza informatii despre pretul total al sejurului clientului. Cheie primara: id\_plata;

REZERVARE- facuta de client in cadrul hotelului; salveaza informatii despre client, ii asigura un loc in tipul de camera preferat, servind tipul de masa dorit si ofera extra servicii pentru o experienta cat mai placuta in cadrul sejurului sau. Cheie primara: id\_rezervare.

1. Descrierea relatiilor

* Relația SERVICII\_EXTRA\_folosește\_MIJLOC\_TRANSPORT:

Serviciile extra folosesc mai multe mijloace de transport (relație dintre entitățile SERVICII EXTRA și MIJLOC\_TRANSPORT)

Cardinalitatea maximă: many-many (M:M)

- Câte servicii extra pot folosi mijloace de transport? Multe!

- Câte mijloace de transport se pot folosi pentru servicii extra? Multe!

Cardinalitate minima: unu-unu (1:1)

- Câte servicii extra trebuie sa foloseasca mijloace de transport? Unu!

- Câte mijloace de transport trebuie sa se foloseasca pentru servicii extra? Unu!

* Relația HOTEL\_dispune\_SERVICII\_EXTRA:

Într-un hotel sunt dispuse mai multe servicii extra. (relație dintre entitățile HOTEL și SERVICII\_EXTRA)

Cardinalitatea maximă: many-many (M:M)

- Câte hoteluri pot dispune de servicii extra? Multe!

- Câte servicii extra poate pune la dispoziție hotelul? Multe!

Cardinalitate minima: zero-unu (0:1)

- Câte hoteluri trebuie să dispună de servicii extra? Unu!

- Câte servicii extra trebuie să puna la dispoziție un hotel? Zero!

* Relația HOTEL\_lucrează\_ANGAJAT:

Într-un hotel lucrează mai mulți angajați. (relație dintre entitățile HOTEL și ANGAJAT)

Cardinalitatea maximă: one-many (1:M)

- Câți angajați pot lucra într-un hotel? Mulți!

- În câte hoteluri poate să lucreze un angajat ? Unu!

Cardinalitate minima: unu-unu (1:1)

- Câți angajați trebuie să lucreze la un hotel? Unu!

- În câte hoteluri trebuie să lucreze un angajat? Unu!

* Relația ANGAJAT\_are\_JOB:

Un angajat are mai multe joburi. (relație dintre entitățile ANGAJAT și JOB)

Cardinalitatea maximă: one-many (1:M)

- Câți angajați pot avea un job? Mulți!

- Câte joburi poate să aibă un angajat? Unu!

Cardinalitate minima: unu-unu (1:1)

- Câți angajati trebuie să aibă un job? Unu!

- Câte joburi trebuie să aibă un angajat? Unu!

* Relația HOTEL\_deține\_APARTAMENT:

Un hotel deține mai multe apartamente. (relație dintre entitățile HOTEL și APARTAMENT)

Cardinalitatea maximă: one-many (1:M)

- Câte hoteluri pot avea un apartament? Unu!

- Câte apartamente pot să fie deținute de un hotel? Multe!

Cardinalitate minima: unu-unu (1:1)

- Câte hoteluri trebuie să dețină un apartamet? Unu!

- Câte apartamente trebuie să fie deținute de un hotel? Unu!

* Relația HOTEL\_deține\_CAMERA:

Un hotel deține mai multe camere. (relație dintre entitățile HOTEL și CAMERA)

Cardinalitatea maximă: one-many (1:M)

- Câte hoteluri pot avea o cameră? Unu!

- Câte camere pot să fie deținute de un hotel? Multe!

Cardinalitate minima: unu-unu (1:1)

- Câte hoteluri trebuie să dețină o camera? Unu!

- Câte camere trebuie să fie deținute de un hotel? Unu!

* Relația CAMERA\_ocupată\_prin\_REZERVARE:

Mai multe camere pot fi ocupate prin rezervare. (relație dintre entitățile CAMERA și REZERVARE)

Cardinalitatea maximă: one-many (1:M)

- Câte camere pot fi ocupate prin rezervare? Multe!

- Prin cate rezervări se poate ocupa o cameră? Una!

Cardinalitate minima: zero-unu (0:1)

- Câte camere trebuie să fie ocupate prin rezervare? Una!

- Câte rezervări trebuie să ocupe o camera? Zero!

* Relația APARTAMENT\_ocupat\_prin\_REZERVARE:

Mai multe apartamente pot fi ocupate prin rezervare. (relație dintre entitățile APARTAMENT și REZERVARE)

Cardinalitatea maximă: one-many (1:M)

- Câte apartamente pot fi ocupate prin rezervare? Multe!

- Prin cate rezervări se poate ocupa un apartament? Una!

Cardinalitate minima: zero-unu (0:1)

- Câte apartamente trebuie să fie ocupate prin rezervare? Unu!

- Câte rezervări trebuie să ocupe un apartament? Zero!

* Relația REZERVARE\_facută\_de\_CLIENT:

Mai multe rezervări pot fi făcute de clienti. (relație dintre entitățile REZERVARE și CLIENT)

Cardinalitatea maximă: one-many (1:M)

- Câte rezervări pot fi făcute de un client? Una!

- Câți clienți pot face rezervări? Mulți!

Cardinalitate minima: unu-unu (1:1)

- Câte rezervări trebuie făcute de un client? Una!

- Câți clienți trebuie să fi făcut o rezervare? Unu!

* Relația REZERVARE\_beneficiază\_de\_TIP\_SEJUR:

Într-o rezervare beneficiezi de un singur sejur. (relație dintre entitățile REZERVARE și TIP\_SEJUR)

Cardinalitatea maximă: one-many (1:M)

- Câte rezervări pot beneficia de un sejur? Multe!

- De câte sejururi se poate beneficia în urma unei rezervări? Unu!

Cardinalitate minima: unu-unu (1:1)

- Câte rezervări trebuie să beneficieze de un sejur? Una!

- De câte sejururi trebuie să se beneficieze in urma unei rezervări? Unu!

* Relația REZERVARE\_emisa\_in\_urma\_FACTURA:

În urma unei rezervări se emite o factură. (relație dintre entitățile REZERVARE și FACTURA)

Cardinalitatea maximă: one-one (1:1)

- Câte rezervări pot emite o factură? Una!

- Câte facturi se pot emite în urma unei rezervări? Una!

Cardinalitate minima: unu-unu (1:1)

- Câte rezervări trebuie să emită o factură? Una!

- Câte facturi trebuie să se emită în urma unei rezervări? Una!

* Relația FACTURA\_conține\_PLATA:

O factură conține o plată. (relație dintre entitățile FACTURA și PLATA)

Cardinalitatea maximă: one-one (1:1)

- Câte facturi pot conține o plată? Una!

- Câte plăți se pot conține într-o factură? Una!

Cardinalitate minima: unu-unu (1:1)

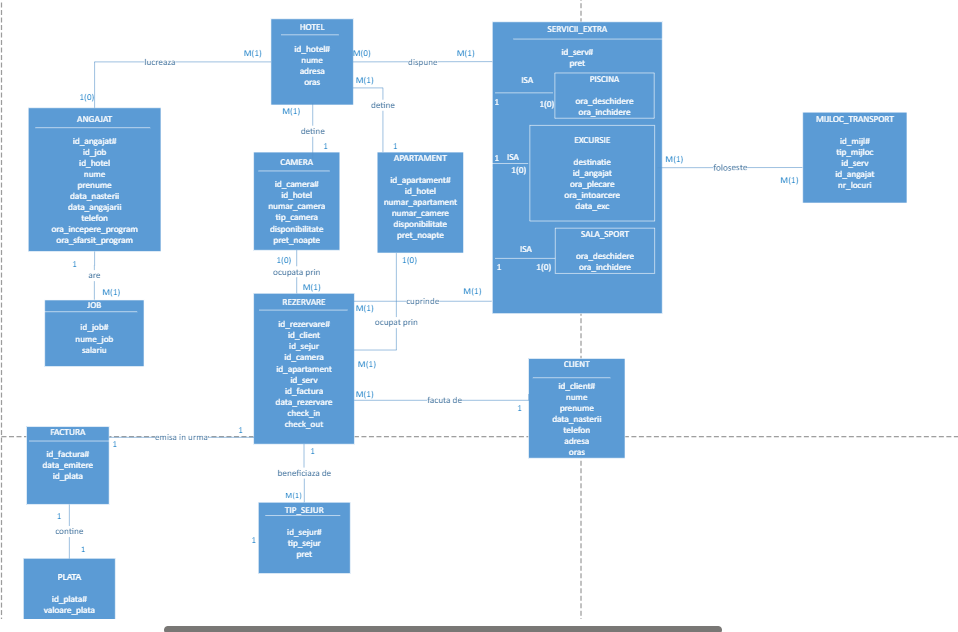
- Câte facturi trebuie să conțina o plată? Una!

- Câte plați trebuie să se conțina într-o factură? Una!

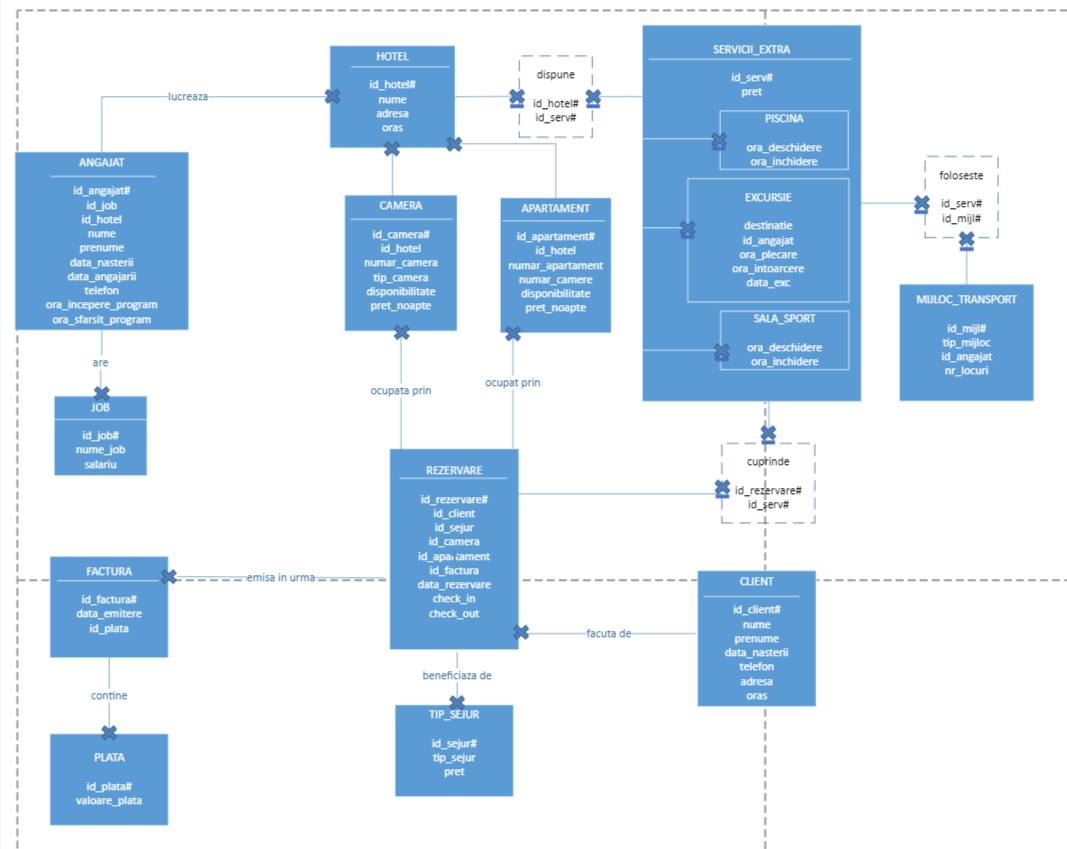
1. Descrierea atributelor

|  |  |
| --- | --- |
| **ENTITĂȚI** | **ATRIBUTE** |
| MIJLOC\_TRANSPORT | * id\_mijl: variabila de tip intreg, constrangere de cheie primara, este automat not null si unique, de lungime maxima 9, reprezinta codul unui mijloc de transport; * tip\_mijloc: variabila de tip sir de caractere, de lungime maxima 15, retine tipul fiecarui mijloc de transport; * id\_angajat: variabila de tip intreg, retine codul fiecarui angajat care se ocupa de conducerea mijloacelor de transport, de lungime maxima 9, trebuie sa corespunda unei valori a cheii primare din tabelul ANGAJAT (FK); * nr\_locuri: variabila de tip intreg, retine numarul de locuri existente in mijlocul de transport, de lungime maxima 3. |
| SERVICII\_EXTRA | * id\_serv: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui serviciu extra, constrangere de cheie primara, este automat not null si unique; * pret: variabila de tip intreg, de lungime maxima 9, retine pretul unui serviciu extra; * ora\_deschidere: variabila de tip sir de caractere, de lungime maxima 5, retine ora deschiderii piscinei; * ora\_inchidere: variabila de tip sir de caractere, de lungime maxima 5, retine ora inchiderii piscinei; * destinatie: variabila de tip sir de caractere, de lungime maxima 20, retine destinatia excursiei; * id\_angajat: variabila de tip intreg, retine codul fiecarui angajat care este responsabil de excursii, de lungime maxima 9, trebuie sa corespunda unei valori a cheii primare din tabelul ANGAJAT (FK); * ora\_plecare: variabila de tip sir de caractere, de lungime maxima 5, retine ora plecarii fiecarei excursii; * ora\_intoarcere: variabila de tip sir de caractere, de lungime maxima 5, retine ora intoarcerii fiecarei excursii; * data\_exc: variabila de tip data calendaristica, retine data in care va avea loc fiecare excursie; * ora\_deschidere: variabila de tip sir de caractere, de lungime maxima 5, retine ora deschiderii salii de sport; * ora\_inchidere: variabila de tip sir de caractere, de lungime maxima 5, retine ora inchiderii salii de sport. |
| HOTEL | * id\_hotel: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui hotel, constrangere de cheie primara, este automat not null si unique; * nume: variabila de tip sir de caractere, de lungime maxima 20, retine numele fiecarui hotel; * adresa: variabila de tip sir de caractere, de lungime maxima 50, retine adresa fiecarui hotel; * oras: variabila de tip sir de caractere, de lungime maxima 20, retine orasul in care se afla fiecare hotel. |
| ANGAJAT | * id\_angajati: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui angajat, constrangere de cheie primara, este automat not null si unique; * id\_job: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui job, trebuie sa corespunda unei valori a cheii primare din tabelul JOB(FK); * id\_hotel: variabila de tip intreg, de lungime maxima 9, retine codul fiecarui hotel, trebuie sa corespunda unei valori a cheii primare din tabelul HOTEL(FK); * nume: variabila de tip sir de caractere, de lungime maxima 15, retine numele fiecarui angajat; * prenume: variabila de tip sir de caractere, de lungime maxima 15, retine prenumele fiecarui angajat; * data\_nasterii: variabila de tip data calendaristica, retine data nasterii fiecarui angajat; * data\_angajarii: variabila de tip data calendaristica, retine data angajarii fiecarui angajat; * telefon: variabila de tip sir de caractere, de lungime maxima 10, retine numarul de telefon al fiecarui angajat, atributul este unique; * ora\_incepere\_program: variabila de tip sir de caractere, de lungime maxima 5, retine ora la care incepe lucrul fiecare angajat; * ora\_sfarsit\_program: variabila de tip sir de caractere, de lungime maxima 5, retine ora la care termina lucrul fiecare angajat. |
| JOB | * id\_job: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui job, constrangere de cheie primara, este automat not null si unique; * nume\_job: variabila de tip sir de caractere, de lungime maxima 20, retine denumirea jobului unui angajat; * salariu: variabila de tip intreg, de lungime maxima 9, retine salariul lunar al fiecarui angajat. |
| CAMERA | * id\_camera: variabila de tip intreg, de lungime maxima 9, reprezinta codul unei camere, constrangere de cheie primara, este automat not null si unique; * id\_hotel: variabila de tip intreg, de lungime maxima 9, retine codul unui hotel, trebuie sa corespunda unei valori a cheii primare din tabelul HOTEL(FK); * numar\_camera: variabila de tip intreg, de lungime maxima 9, retine numarul camerei; * tip\_camera: variabila de tip sir de caractere, de lungime maxima 10, retine ce fel de tip este fiecare camera; * disponibilitate: variabila de tip sir de caractere, de lungime maxima 2, retine daca o camera este ocupata sau nu, se completeaza doar cu “DA” sau “NU”; * pret\_noapte: variabila de tip intreg, de lungime maxima 9, retine pretul pe noapte al unei camere. |
| APARTAMENT | * id\_apartament: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui apartament, constrangere de cheie primara, este automat not null si unique; * id\_hotel: variabila de tip intreg, de lungime maxima 9, retine codul unui hotel, trebuie sa corespunda unei valori a cheii primare din tabelul HOTEL(FK); * numar\_apartament: variabila de tip intreg, de lungime maxima 9, retine numarul apartamentului; * numar\_camere: variabila de tip intreg, retine numarul de camere din apartament, trebuie sa fie intre 1 si 5; * disponibilitate: variabila de tip sir de caractere, de lungime maxima 2, retine daca un apartament este ocupat sau nu; * pret\_noapte: variabila de tip intreg, de lungime maxima 9, retine pretul pe noapte al unui apartament. |
| REZERVARE | * id\_rezervare: variabila de tip intreg, de lungime maxima 9, reprezinta codul unei rezervari, constrangere de cheie primara, este automat not null si unique; * id\_client: variabila de tip intreg, de lungime maxima 9, reprezinta codul clientului care a facut rezervarea, trebuie sa corespunda unei valori a cheii primare din tabelul CLIENT(FK); * id\_sejur: variabila de tip intreg, de lungime maxima 9, retine codul sejurului dorit de client, trebuie sa corespunda unei valori a cheii primare din tabelul TIP\_SEJUR(FK); * id\_camera: variabila de tip intreg, de lungime maxima 9, retine codul camerei rezervate, nullable, trebuie sa corespunda unei valori a cheii primare din tabelul CAMERA(FK); * id\_apartament: variabila de tip intreg, de lungime maxima 9, retine codul apartamentului rezervat, nullable, trebuie sa corespunda unei valori a cheii primare din tabelul APARTAMENT(FK); * id\_factura: variabila de tip intreg, de lungime maxima 9, retine codul facturii rezervarii, trebuie sa corespunda unei valori a cheii primare din tabelul FACTURA(FK); * data\_rezervare: variabila de tip data calendaristica, retine data in care rezervarea a fost facuta; * check\_in: variabila de tip data calendaristica, retine data check-in-ului rezervarii; * check\_out: variabila de tip data calendaristica, retine data check-out-ului rezervarii. |
| FACTURA | * id\_factura: variabila de tip intreg, de lungime maxima 9, reprezinta codul unei facturi, constrangere de cheie primara, este automat not null si unique; * data\_emitere: variabila de tip data calendaristica, retine data emiterii facturii; * id\_plata: variabila de tip intreg, de lungime maxima 9, retine codul platii, trebuie sa corespunda unei valori a cheii primare din tabelul PLATA(FK). |
| PLATA | * id\_plata: variabila de tip intreg, de lungime maxima 9, reprezinta codul unei plati, constrangere de cheie primara, este automat not null si unique; * valoare plata: variabila de tip intreg, de lungime maxima 9, retine valoarea de plata. |
| CLIENT | * id\_client: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui client, constrangere de cheie primara, este automat not null si unique; * nume: variabila de tip sir de caractere, de lungime maxima 15, retine numele clientului; * prenume: variabila de tip sir de caractere, de lungime maxima 15, retine prenumele clientului; * data\_nasterii: variabila de tip data calendaristica, retine data nasterii clientului; * telefon: variabila de tip sir de caractere, de lungime maxima 10, retine numarul de telefon al clientului, unique; * adresa: variabila de tip sir de caractere, de lungime maxima 50, retine adresa clientului; * oras: variabila de tip sir de caractere, de lungime maxima 20, retine orasul de provenienta al clientului. |
| TIP\_SEJUR | * id\_sejur: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui tip de sejur, constrangere de tip cheie primara, este automat not null si unique; * tip\_sejur: variabila de tip sir de caractere, de lungime maxima 20, retine denumirea tipului de sejur; * pret: variabila de tip intreg, de lungime maxima 9, retine pretul tipului de sejur pe toata durata sederii clientului. |
| DISPUNE | * id\_hotel: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui hotel, constrangere de cheie primara, este automat not null si unique; * id\_serv: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui serviciu extra, constrangere de cheie primara, este automat not null si unique. |
| FOLOSESTE | * id\_serv: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui serviciu extra, constrangere de cheie primara, este automat not null si unique; * id\_mijl: variabila de tip intreg, constrangere de cheie primara, este automat not null si unique, de lungime maxima 9, reprezinta codul unui mijloc de transport. |
| CUPRINDE | * id\_rezervare: variabila de tip intreg, de lungime maxima 9, reprezinta codul unei rezervari, constrangere de cheie primara, este automat not null si unique; * id\_serv: variabila de tip intreg, de lungime maxima 9, reprezinta codul unui serviciu extra, constrangere de cheie primara, este automat not null si unique. |

1. Diagrama E/R



1. Diagrama conceptuala



1. Schemele relationale

HOTEL(id\_hotel#, nume, adresa, oras)

ANGAJAT(id\_angajat#, id\_hotel, id\_job, nume, prenume, data\_nasterii, data\_angajarii, telefon, ora\_incepere\_program, ora\_sfarsit\_program)

JOB(id\_job#, nume\_job, salariu)

CAMERA(id\_camera#, tip\_camera, disponibilitate, pret\_noapte)

APARTAMENT(id\_apartament#, numar\_camere, disponibilitate, pret\_noapte)

SERVICII\_EXTRA(id\_serv#, pret)

DISPUNE(id\_hotel#, id\_serv#)

PISCINA(id\_serv#, ora\_deschidere, ora\_inchidere)

SALA\_SPORT(id\_serv#, ora\_deschidere, ora\_inchidere)

EXCURSIE(id\_serv#, destinatie, id\_angajat, ora\_plecare, ora\_intoarcere, data\_exc)

MIJLOC\_TRANSPORT(id\_mijl#, id\_angajat, tip\_mijloc, nr\_locuri)

FOLOSESTE(id\_serv#, id\_mijl#)

CLIENT(id\_client#, nume, prenume, data\_nasterii, telefon, adresa)

PLATA(id\_plata#, valoare\_plata)

FACTURA(id\_factura#, data\_emitere, id\_plata)

TIP\_SEJUR(id\_sejur#, tip\_sejur, pret)

REZERVARE(id\_rezervare#, id\_client, id\_camera, id\_sejur, id\_factura, check\_in, check\_ou, data\_rezervare)

CUPRINDE(id\_rezervare#, id\_serv#)

1. Formele normale

O relatie de tip FN1 presupune ca tuturor atributelor din cadrul ei sa le corespunda o valoare indivizibila. Pentru a exemplifica o normalizare de tipul 1, voi ilustra un caz ipotetic non-FN1 in cadrul diagramei construite pe care il voi aduce la forma normala 1.

In cadrul tabelei REZERVARE, regasim cheia primara id\_rezervare si cheia externa id\_serv, corespunzatoare tabelei SERVICII\_EXTRA. Astfel, o rezervare poate contine mai multe servicii, identificate prin codul unic id\_serv.

Relatia REZERVARE

|  |  |
| --- | --- |
| Id\_rezervare# | Id\_serv |
| 1 | 1, 31, 2 |
| 2 | 2, 33 |

In tabelul de mai sus, se poate observa faptul ca nu este ilustrata o relatie de tipul FN1, intrucat intrarilor din coloana id\_rezervare le corespund mai multe valori din coloana id\_serv. Normalizarea se realizeaza astfel:

Relatia CUPRINDE

|  |  |
| --- | --- |
| Id\_rezervare# | Id\_serv# |
| 1 | 1 |
| 1 | 31 |
| 1 | 2 |
| 2 | 2 |
| 2 | 33 |

In acest moment, fiecare atribut din cadrul relatiei are o valoare indivizibila.

Forma normala 2

O relatie este incadrata in tipul FN2 daca si numai daca aceasta este o relatie de tip FN1 si fiecare atribut ce nu participa la compunerea cheii primare este dependent de intreaga cheie primara.

Pentru a exemplifica mai bine normalizarea de tip 2, ilustrez un nou caz ipotetic pe baza diagramei concepute. Construiesc relatia ANGAJAT\_LA\_JOB, compusa din entitatile ANGAJAT si JOB, deja existente.

Relatia ANGAJAT\_LA\_JOB include cheile primare id\_angajat si id\_job, cu mai multe atribute componente, dintre care amintesc nume\_angajat, nume\_job si salariu.

Relatia ANGAJAT\_LA\_JOB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id\_angajat# | Id\_job# | Nume\_angajat | Nume\_job | Salariu |
| 1 | 4 | Popescu | Receptionist | 23000 |
| 2 | 4 | Ionescu | Receptionist | 23000 |
| 3 | 6 | Mihailescu | Bucatar | 27000 |

Se observa ca atributele nu depind in totalitate de cheia primara (numele angajatului nu depinde de id\_ul job-ului; numele job-ului nu depinde de id-ul angajatului; salariul nu depinde de id-ul angajatului). Dependentele arata astfel:

* Id\_angajat# => nume\_angajat
* Id\_job# => nume\_job, salariu

Pentru aducerea relatiei la forma normala 2, despartim relatia ANGAJAT\_LA\_JOB in ANGAJAT si JOB.

Relatia ANGAJAT

|  |  |  |
| --- | --- | --- |
| Id\_angajat# | Id\_job | nume |
| 1 | 4 | Popescu |
| 2 | 4 | Ionescu |
| 3 | 6 | Mihailescu |

Relatia JOB

|  |  |  |
| --- | --- | --- |
| Id\_job# | Nume\_job | Salariu |
| 4 | Receptionist | 23000 |
| 4 | Receptionist | 23000 |
| 6 | Bucatar | 27000 |

Utilizand cele doua relatii, se respecta atat forma normala de tip 1, cat si faptul ca fiecare atribut depinde de cheia primara a relatiei din care face parte. Deci, in ambele relatii, am obtinut o relatie de tip FN2.

Forma normala de tip 3

O relatie se considera in FN3 daca si numai daca este in FN2, iar totalitatea atributelor care nu compun cheia primara depind in mod direct de aceasta.

Pentru exemplificare, ilustram un caz ipotetic. Orice camera poate fi rezervata la pretul de 125 lei/noapte. Valoarea de plata reprezinta pretul anterior mentionat multiplicat cu numarul de nopti petrecute la hotel.

Relatia REZERVARE

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_  rezervare# | Id\_  camera | Nr\_nopti | Valoare\_plata |
| 1 | 12 | 4 | 975 |
| 2 | 13 | 5 | 600 |
| 3 | 14 | 7 | 1200 |

Relatia de mai sus nu se incadreaza in FN3 deoarece atributul “valoare\_plata” depinde atat de cheia primara, cat si de atributul “nr\_nopti”, fiind o dependenta tranzitiva fata de cheia primara, fapt ce scoate relatia din FN3. Dependentele sunt urmatoarele:

* Id\_rezervare# => {id\_camera, nr\_nopti}
* Id rezervare# => nr\_nopti => valoare\_plata

In vederea normalizarii, vom construi o relatie FACTURA, care sa detina informatii privind valoarea de plata.

Relatia REZERVARE

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_  rezervare# | Id\_  camera | Nr\_nopti | Id\_factura |
| 1 | 12 | 4 | 4 |
| 2 | 23 | 5 | 5 |
| 3 | 5 | 7 | 6 |

Relatia FACTURA

|  |  |
| --- | --- |
| Id\_factura# | Valoare\_plata |
| 4 | 975 |
| 5 | 600 |
| 6 | 1200 |

Astfel, cele doua relatii se incadreaza in FN3, intrucat apartin FN2 (fiecare atribut are o valoare indivizibila si toate atributele depind de intreaga cheie primara) si atributele depind in mod direct de cheia primara.

1. Secventa inserari

create sequence seq\_hotel

increment by 1

start with 6

maxvalue 999999999

nocycle;

insert into hotel values

(seq\_hotel.nextval, 'Glimpse', 'Bulevardul Muncii 1-15', 'Cluj-Napoca');

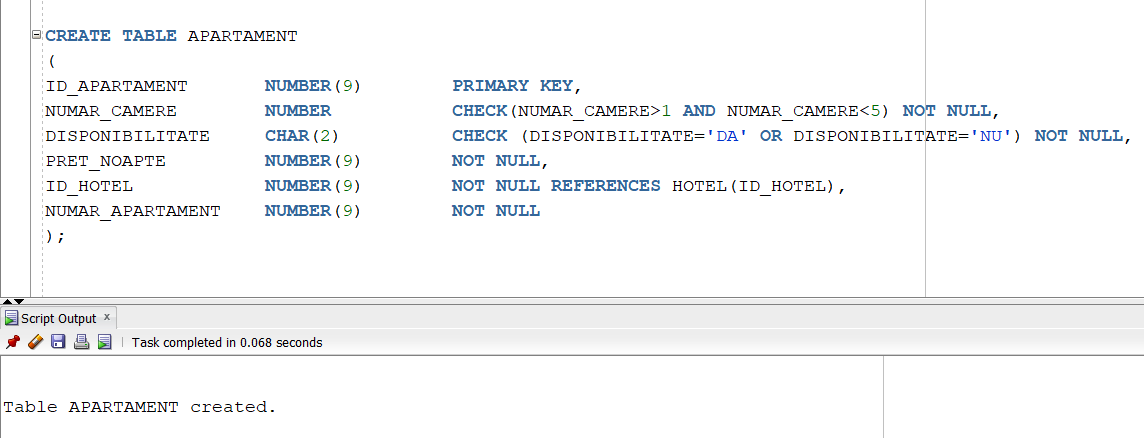
insert into hotel values

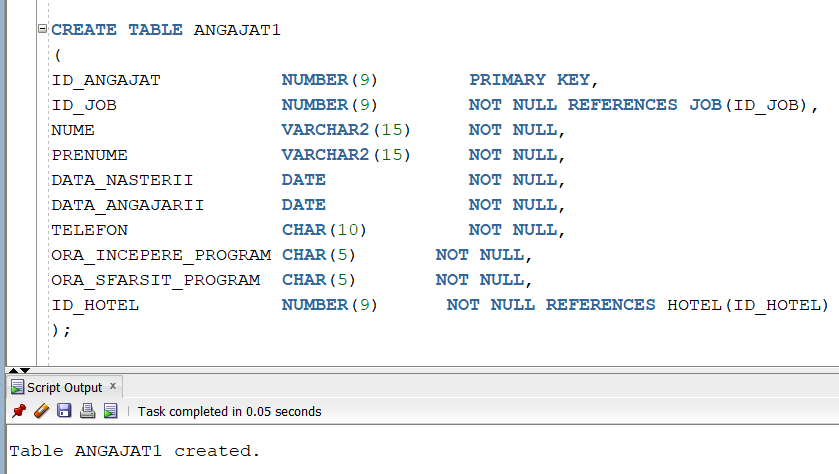
(seq\_hotel.nextval, 'Glimpse', 'Strada Antim Ivireanu 48', 'Ramnicu Valcea');

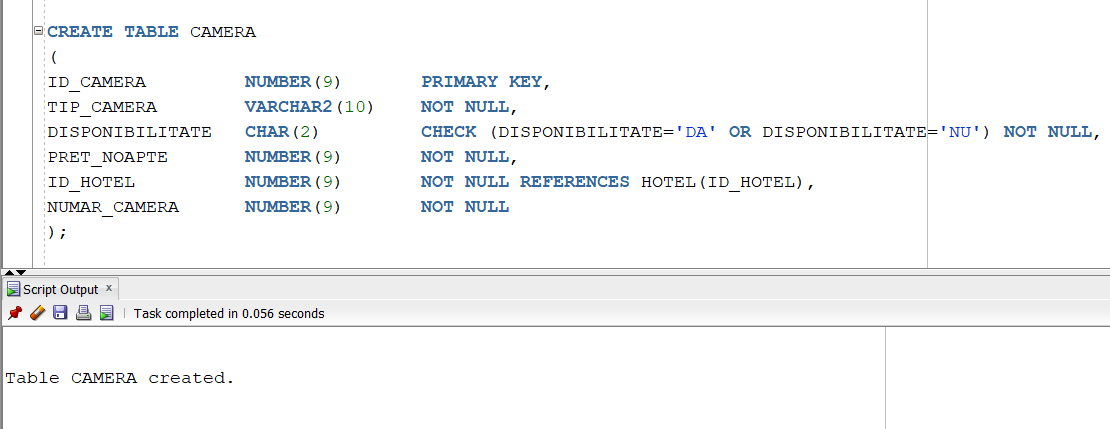
insert into hotel values

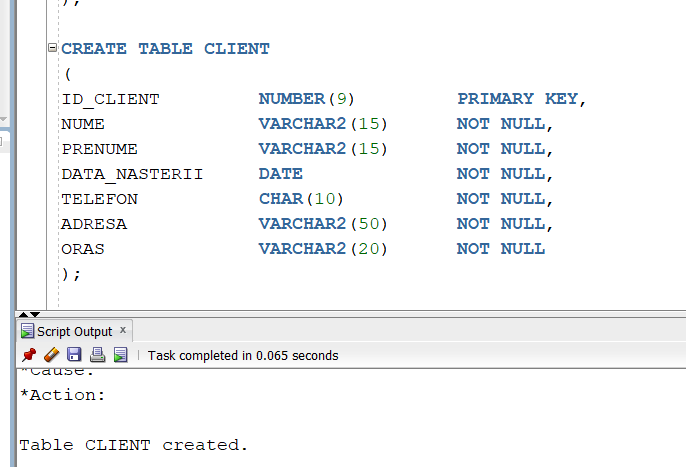
(seq\_hotel.nextval, 'Glimpse', 'Strada Anastasie Panu 54', 'Iasi');

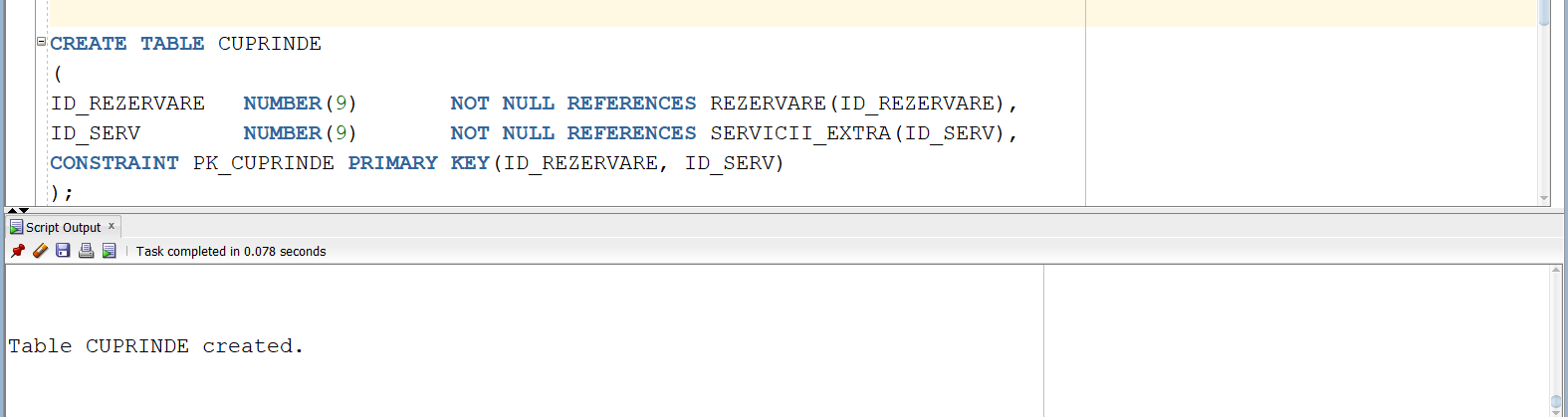
1. Crearea tabelelor in SQL si inserarea de date

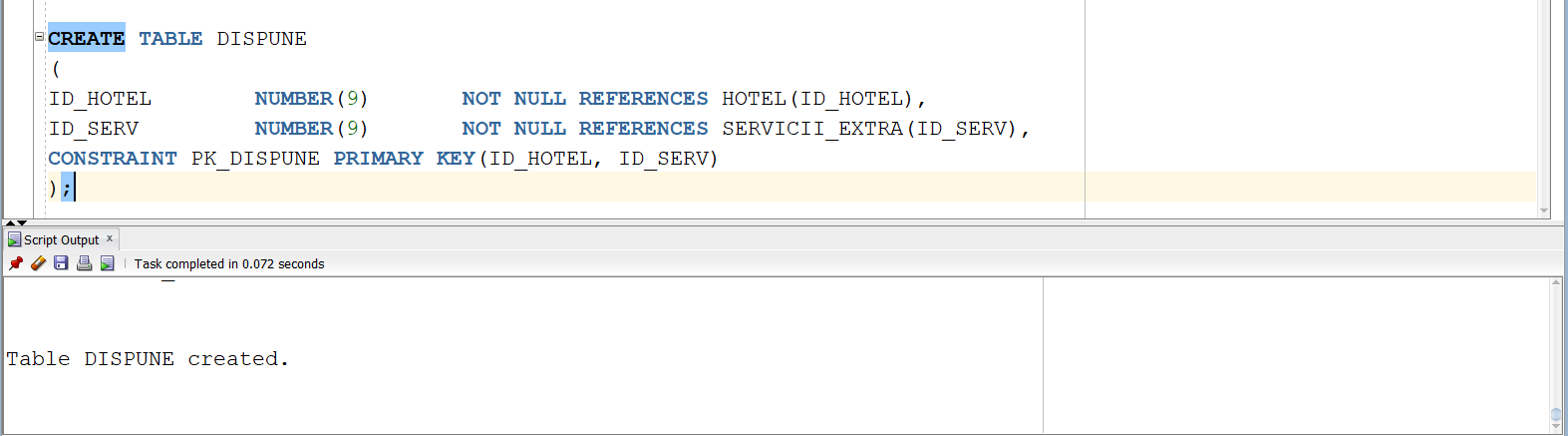


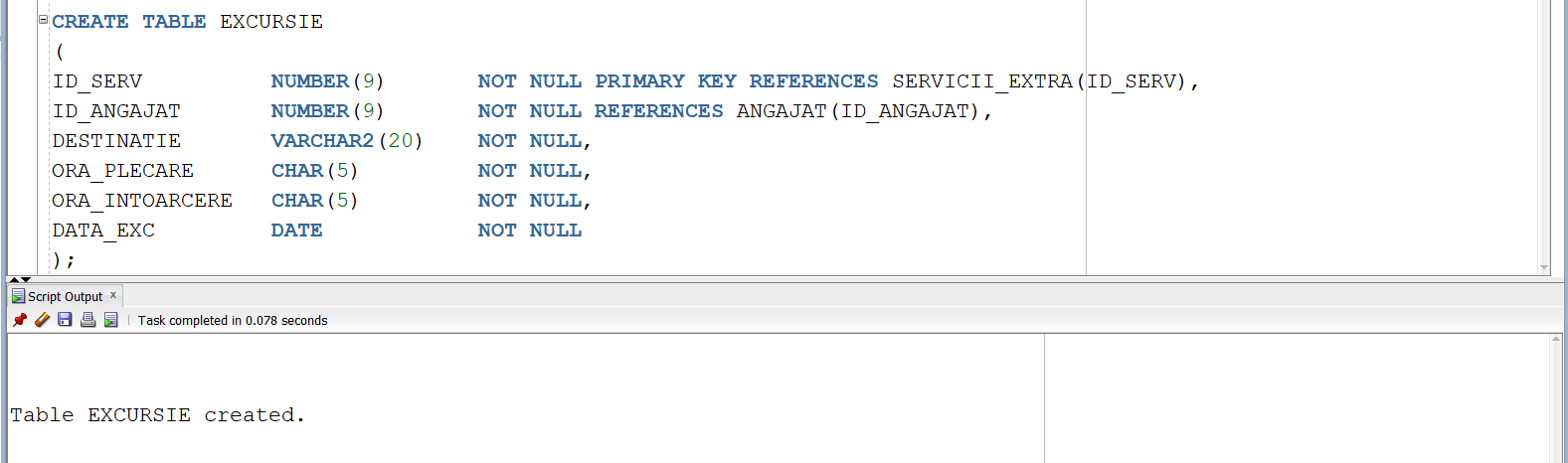


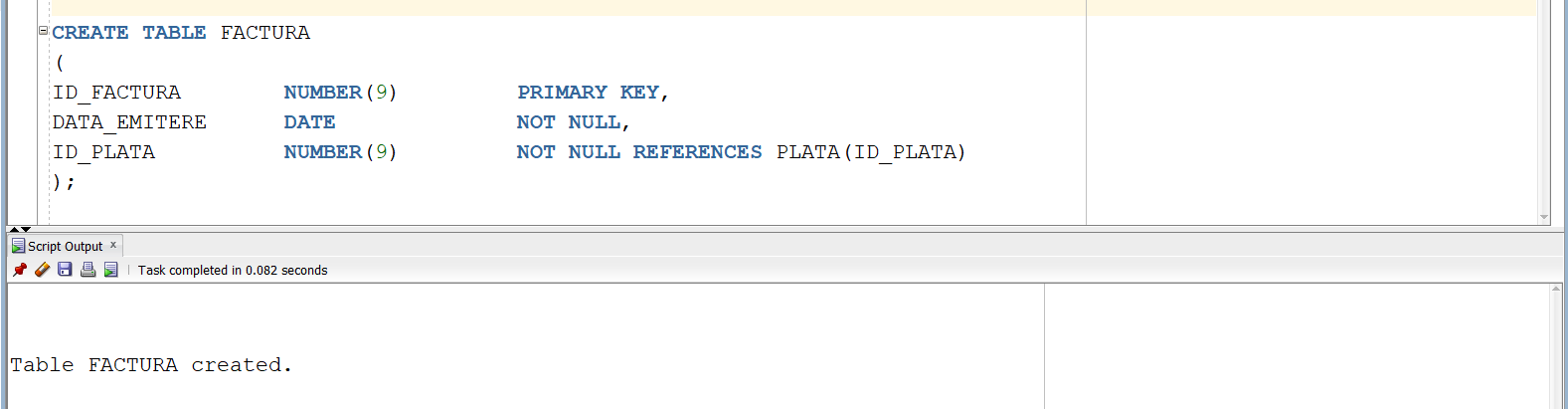


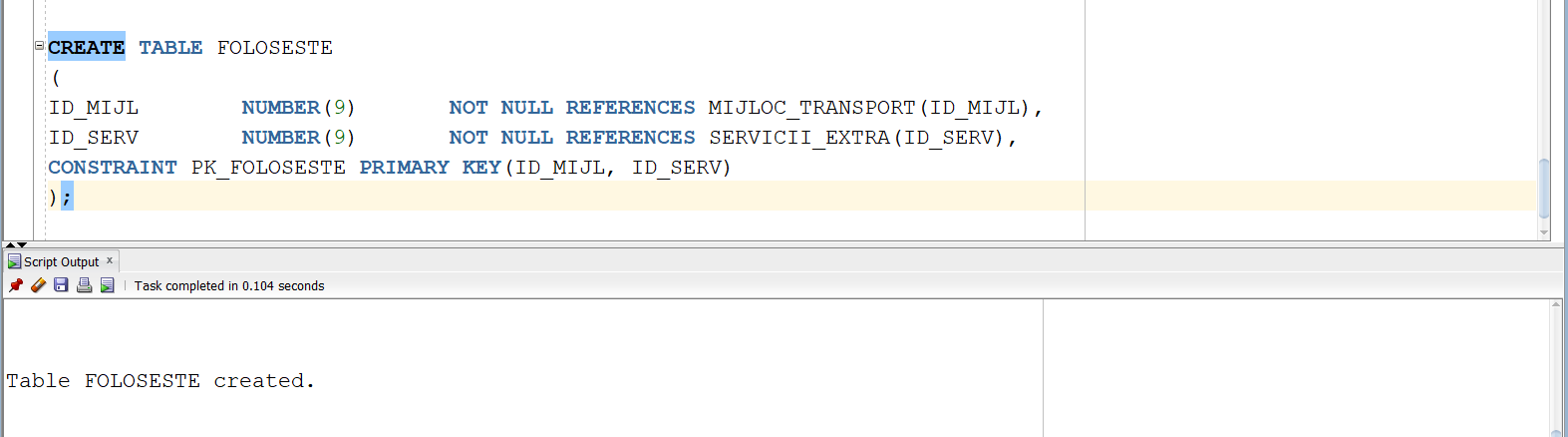


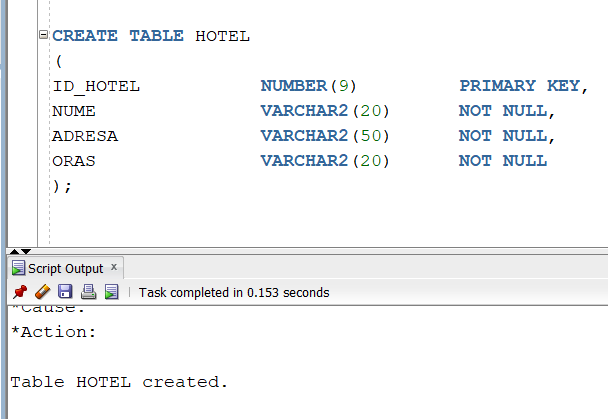


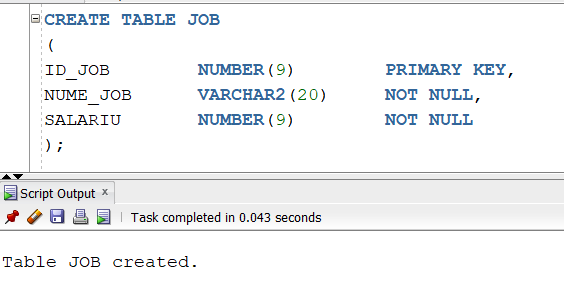


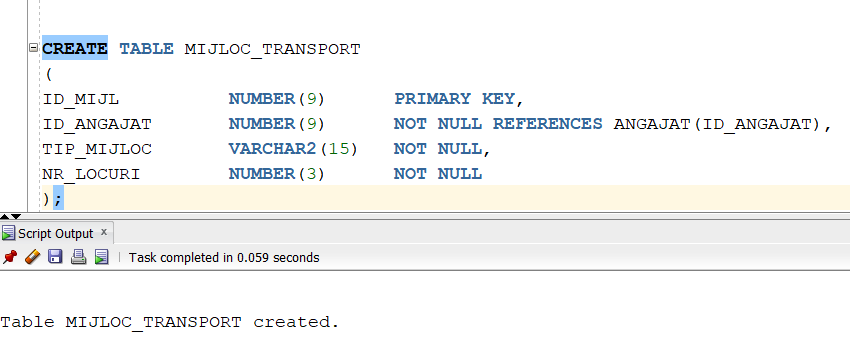


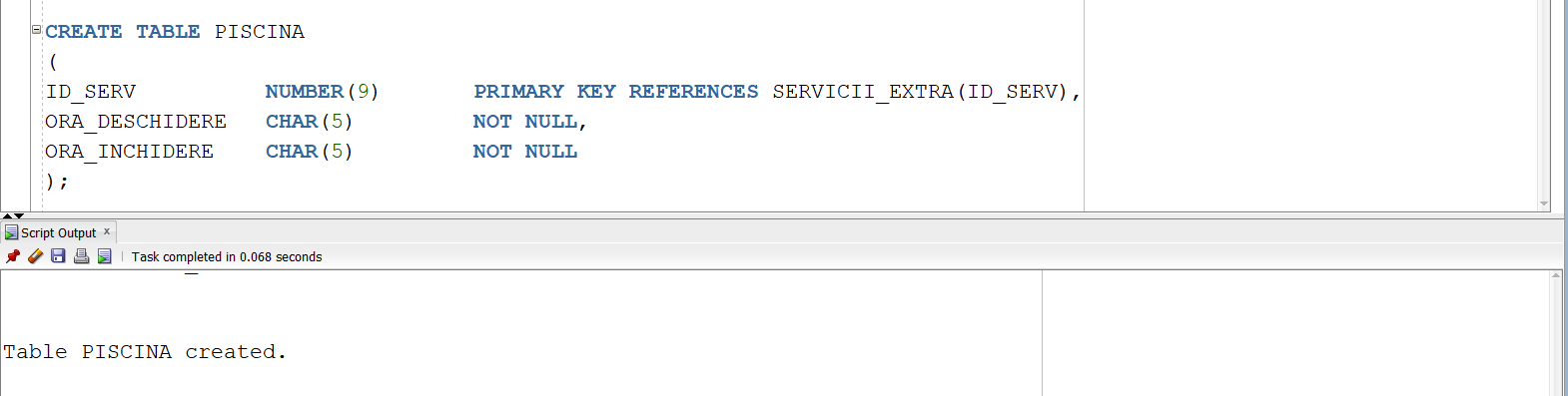


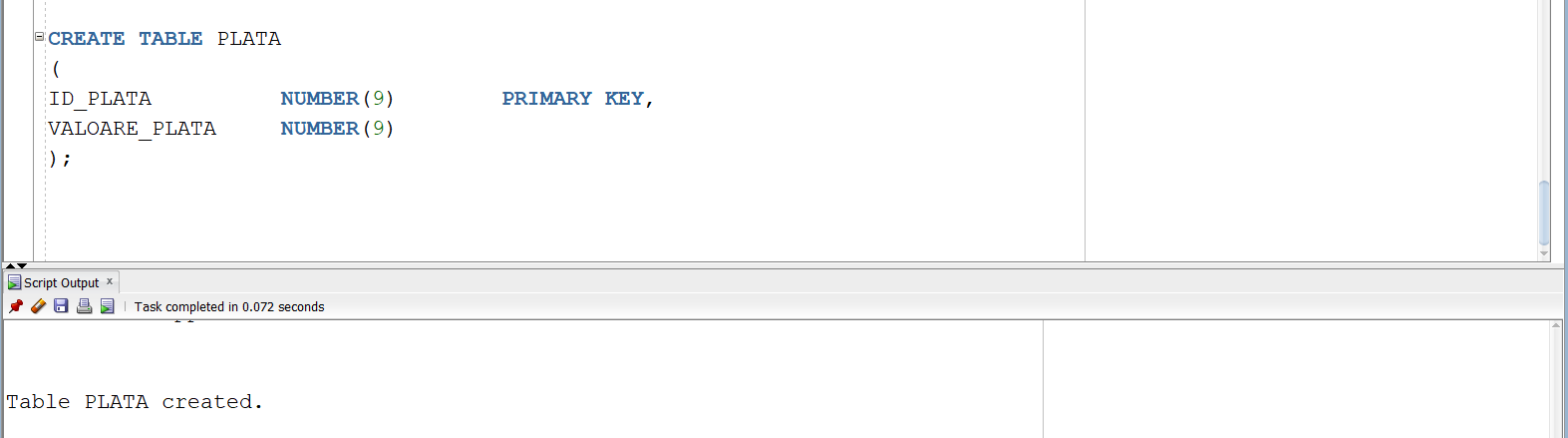


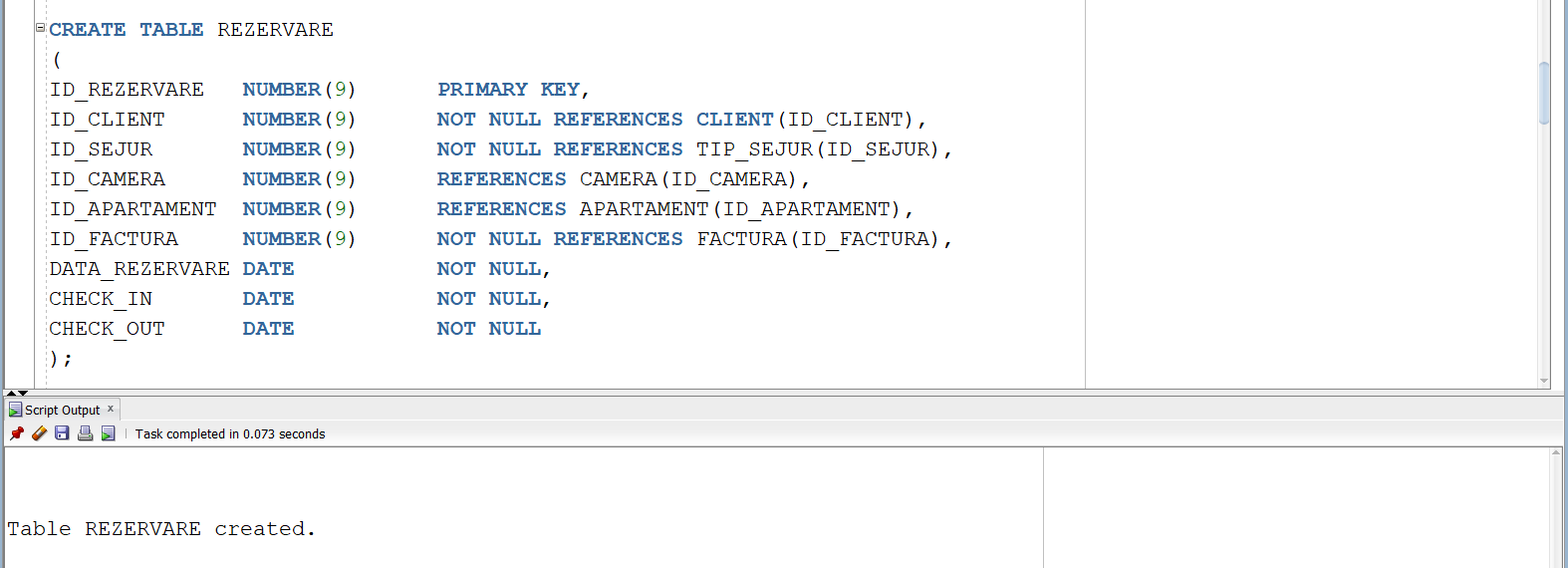


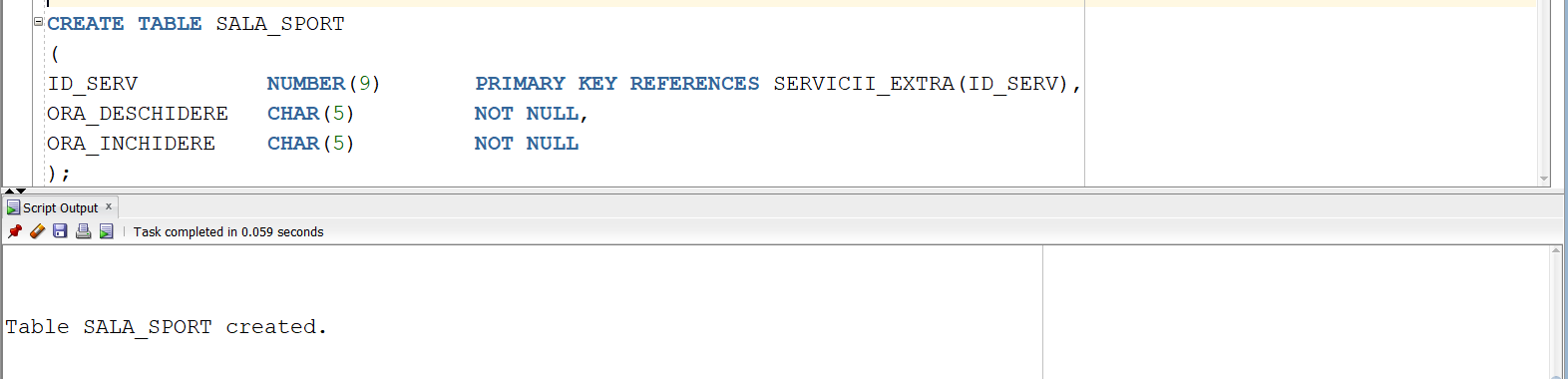


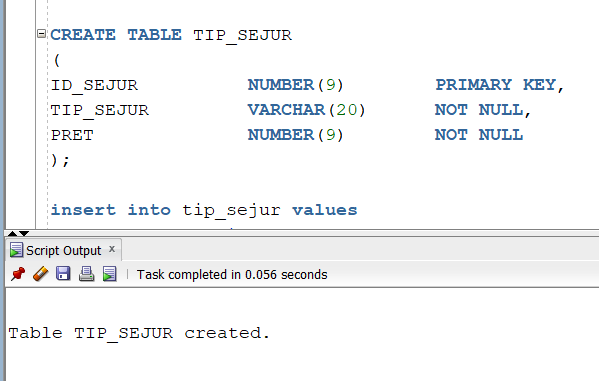




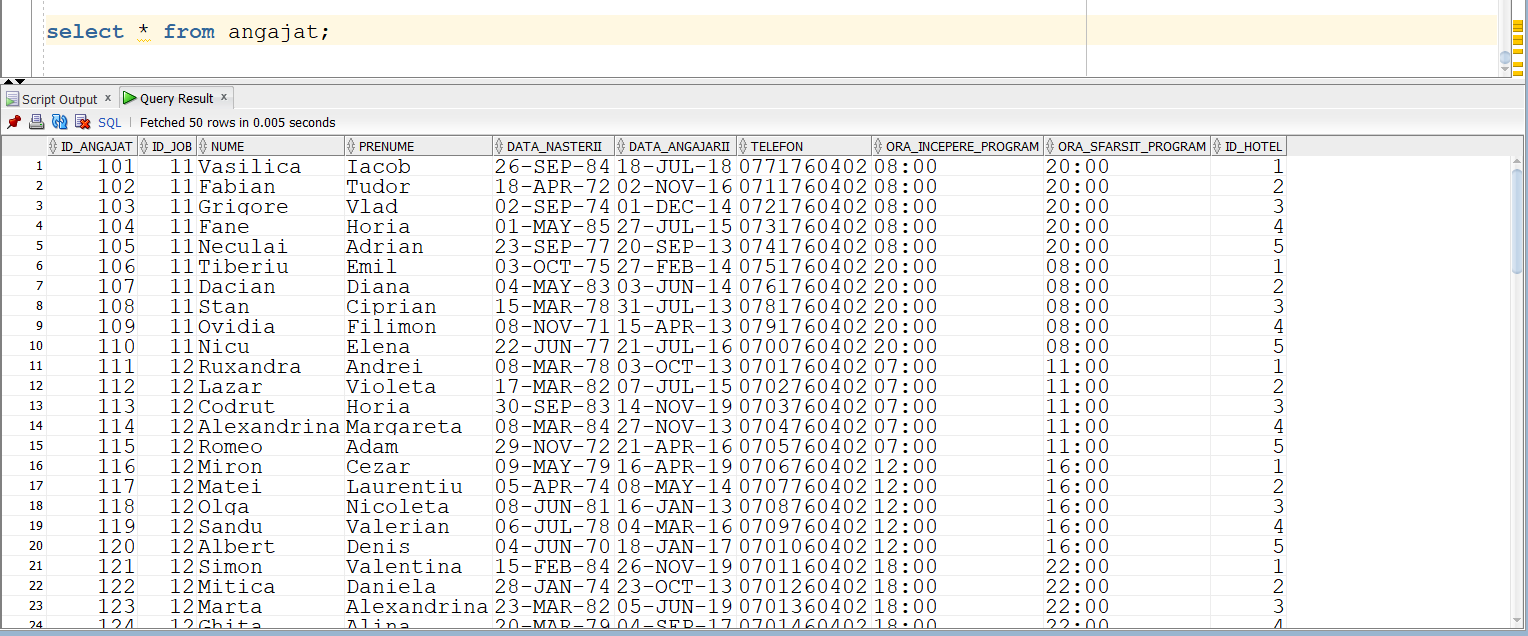


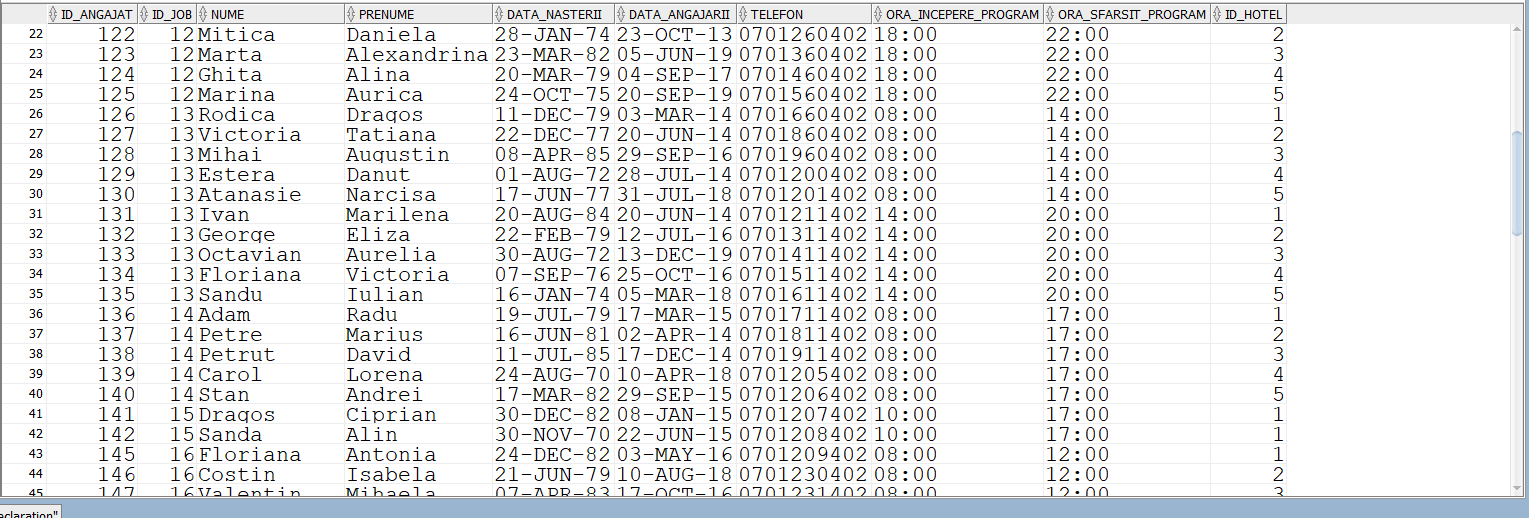


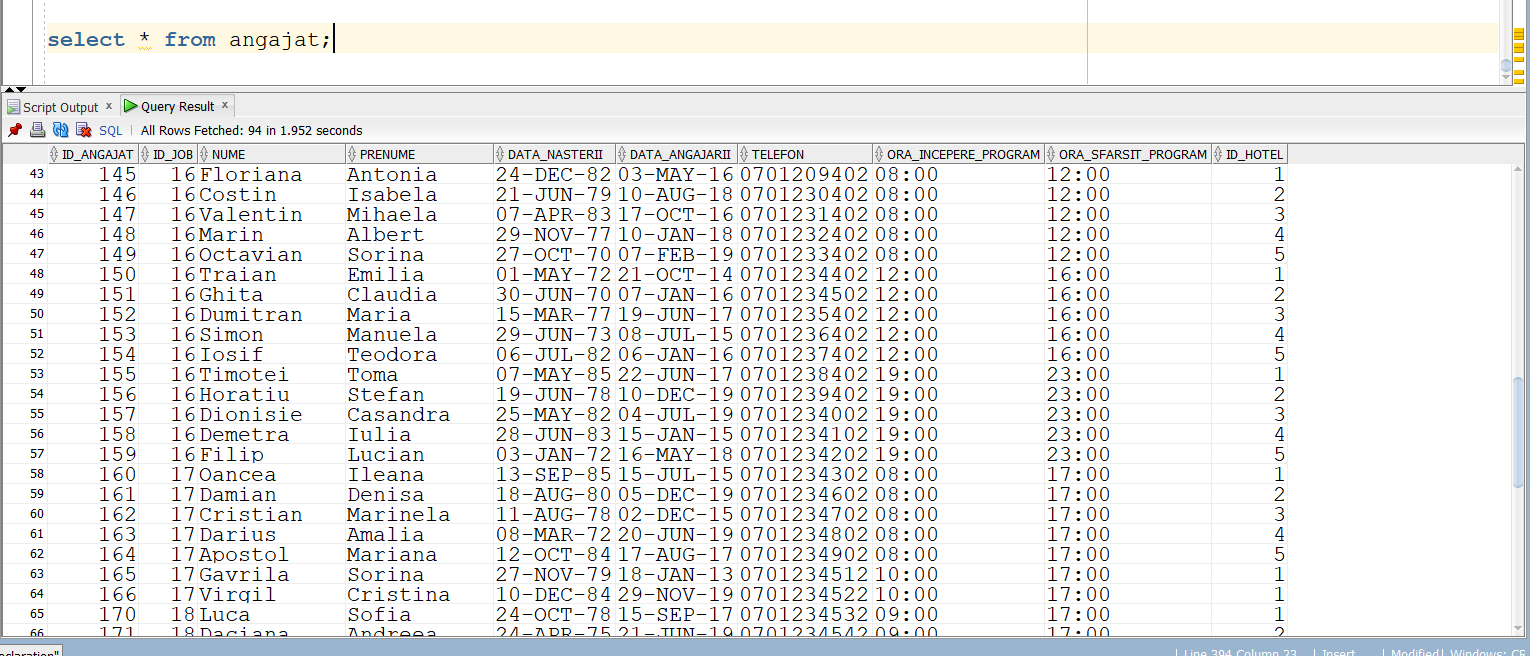


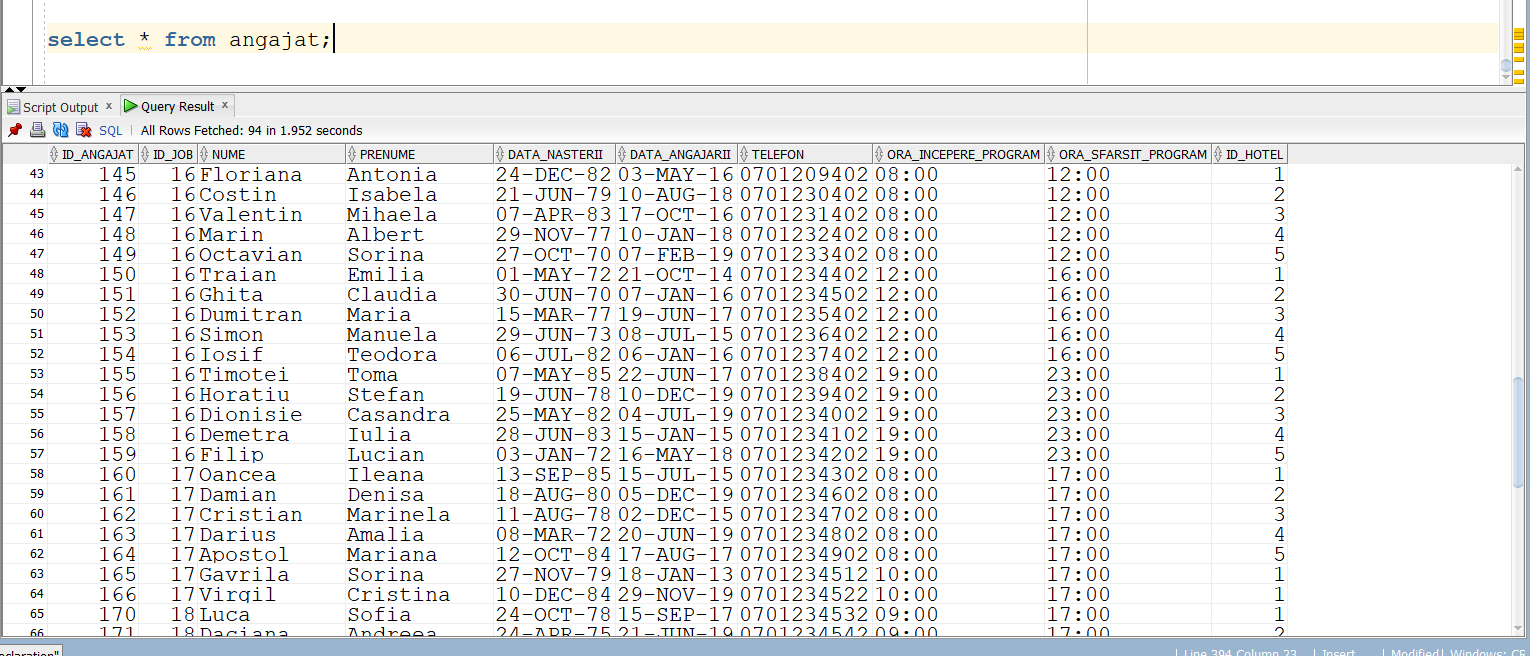


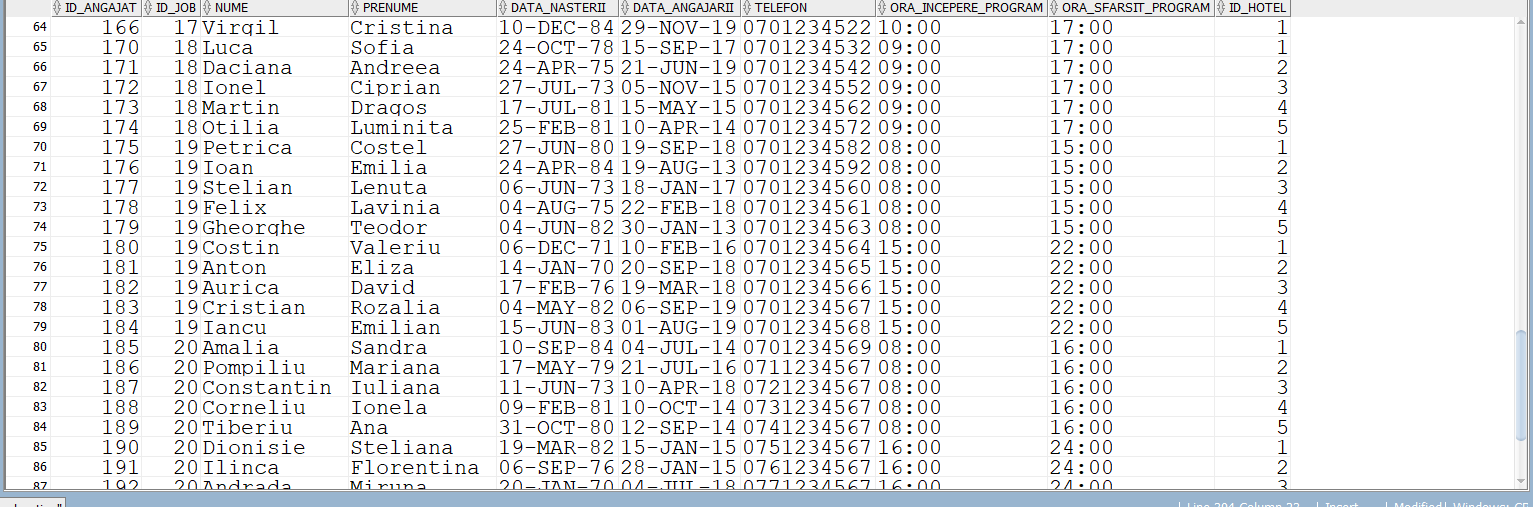


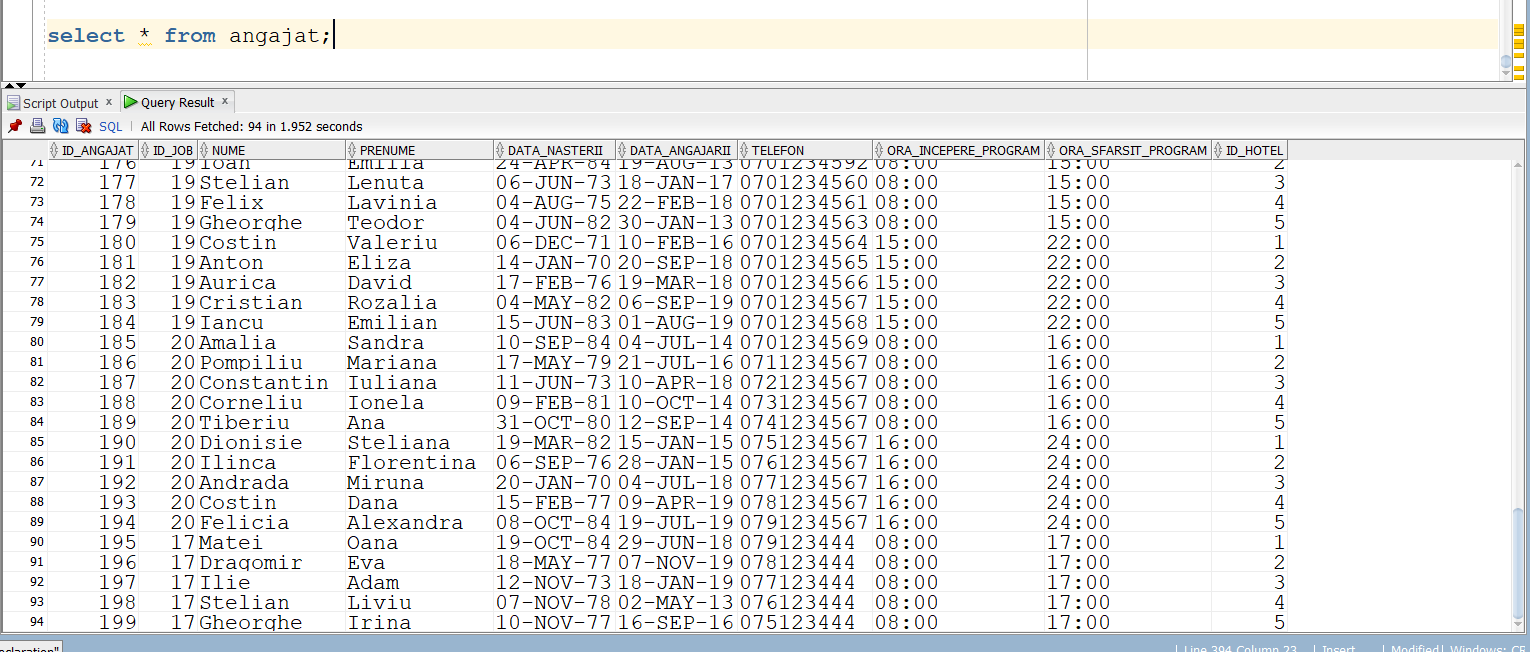


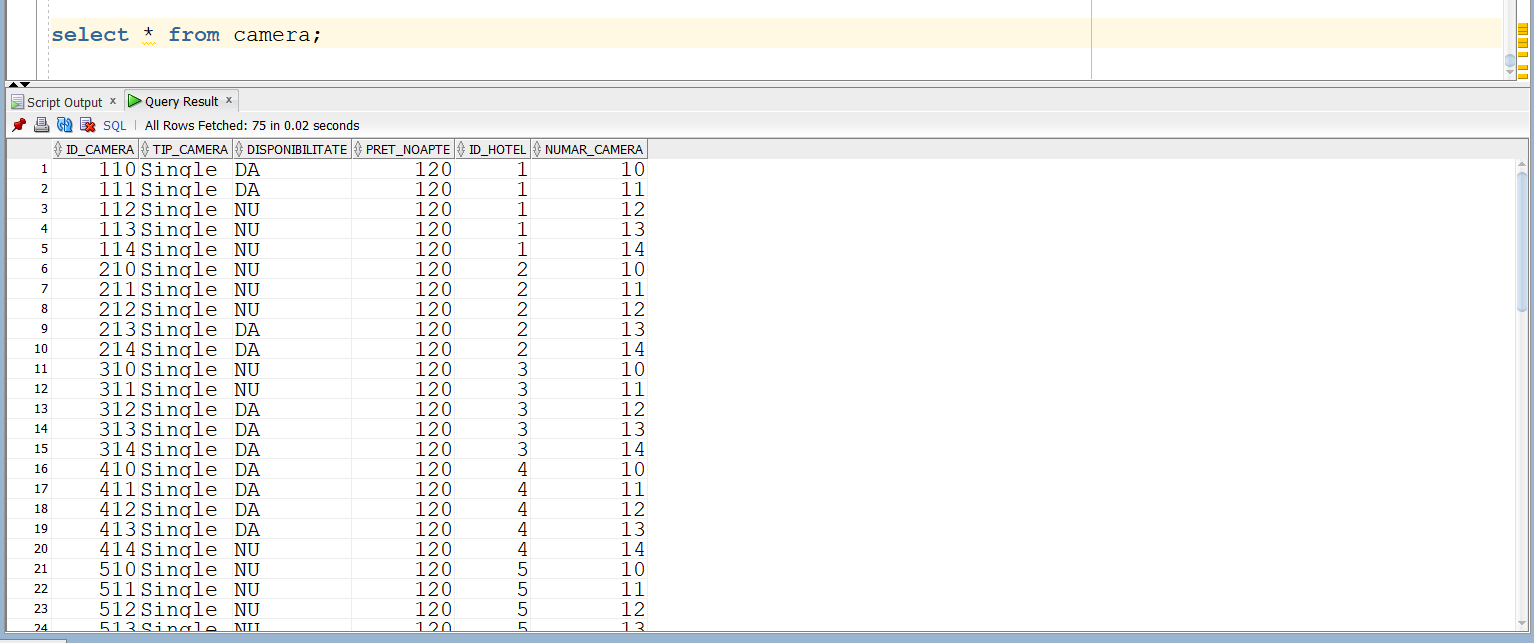


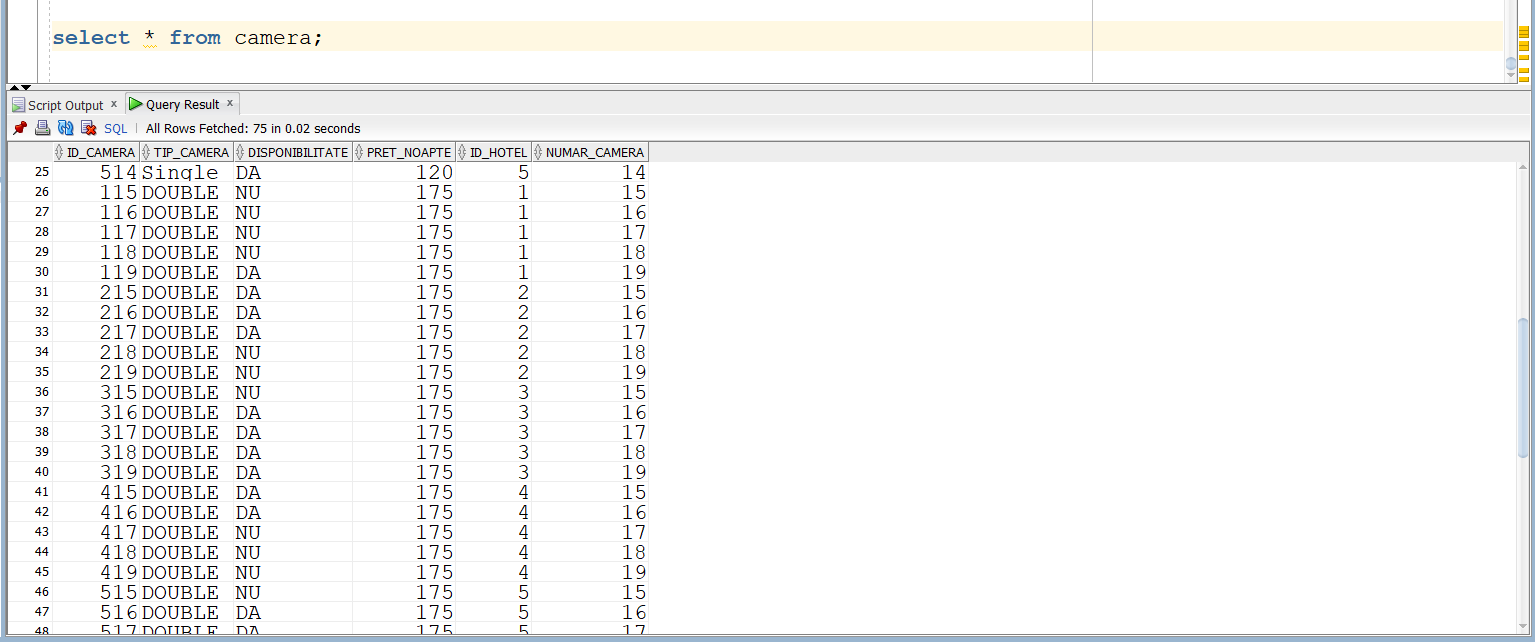


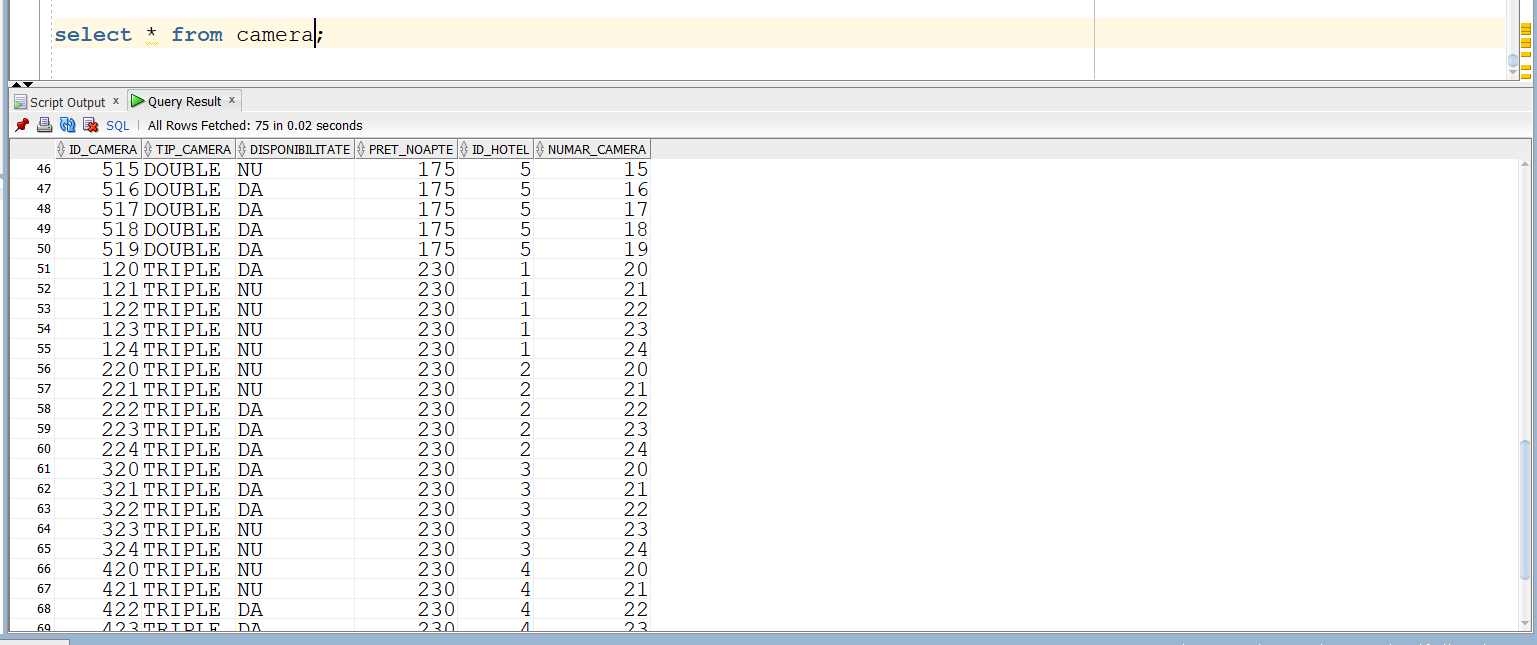


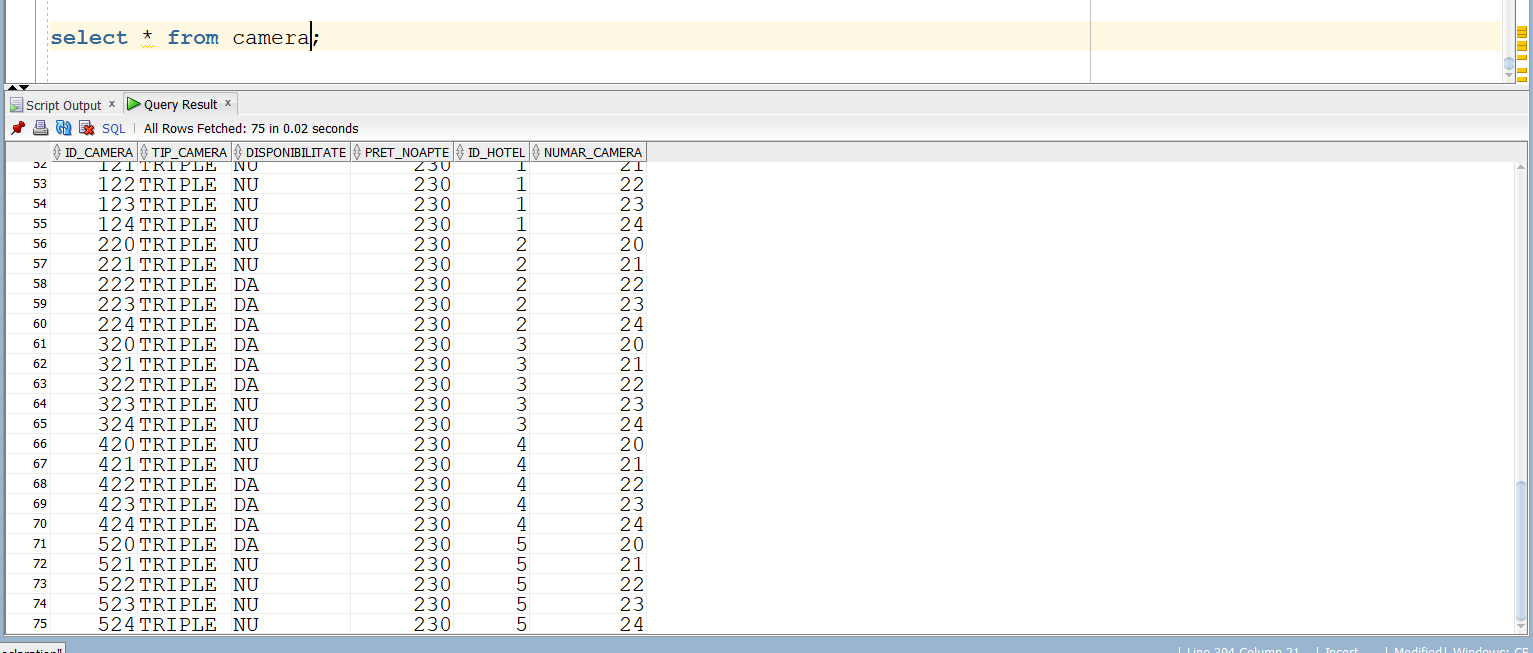


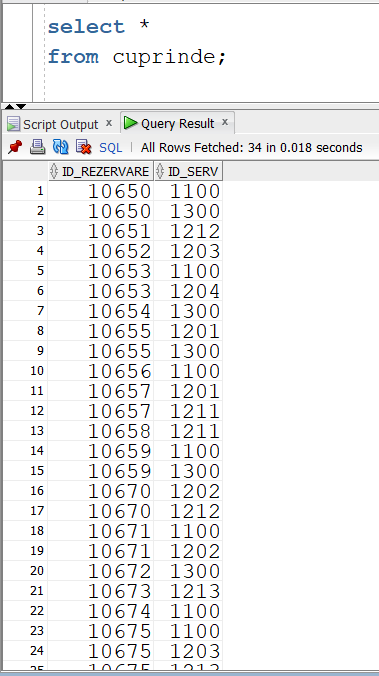


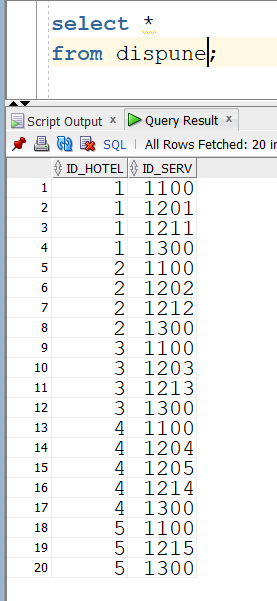


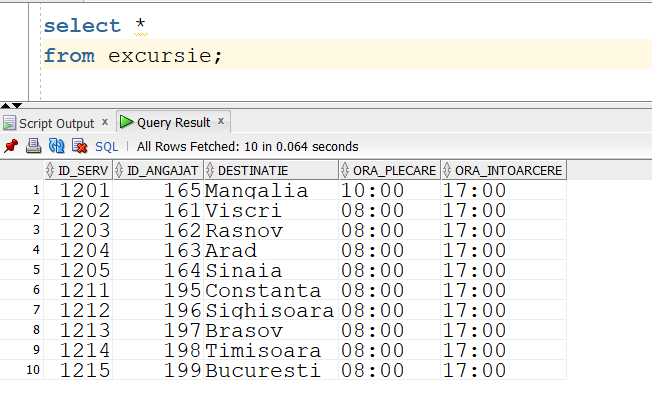


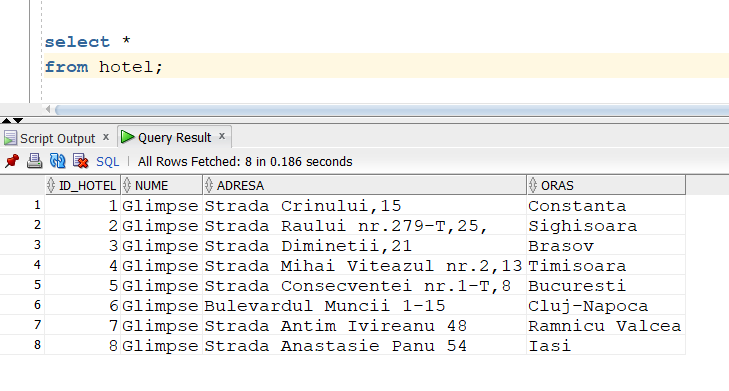


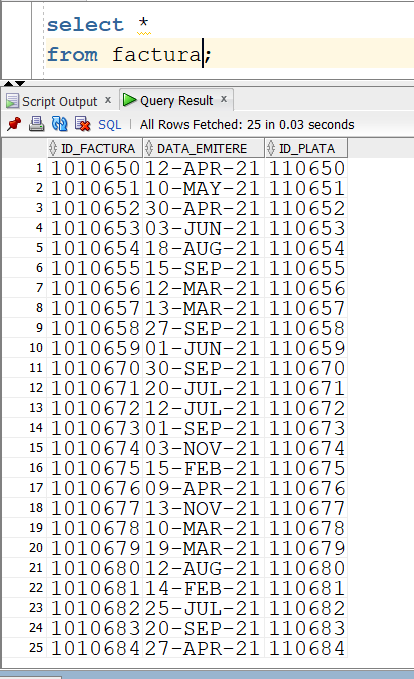


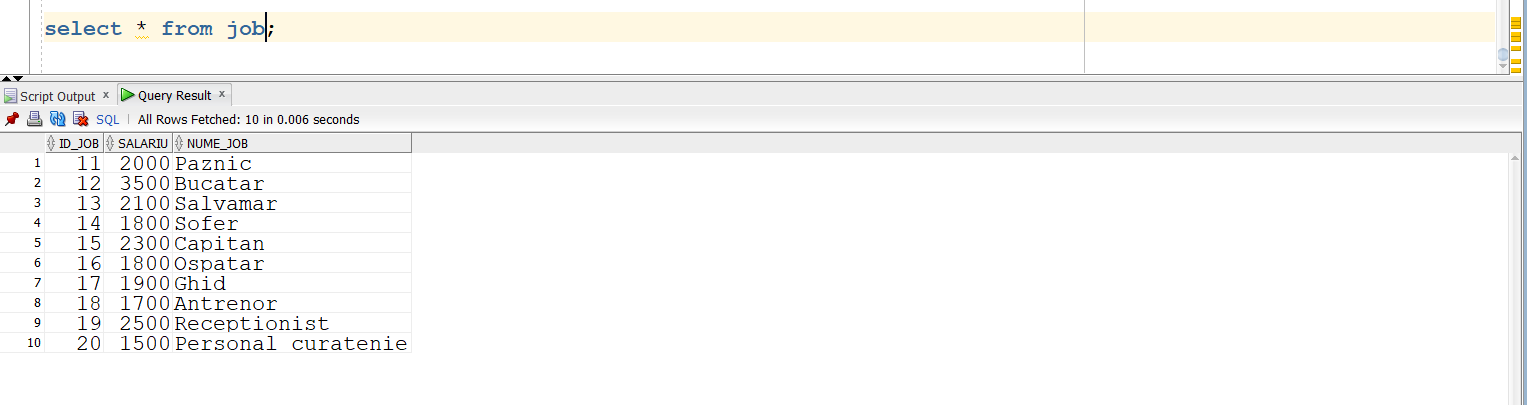


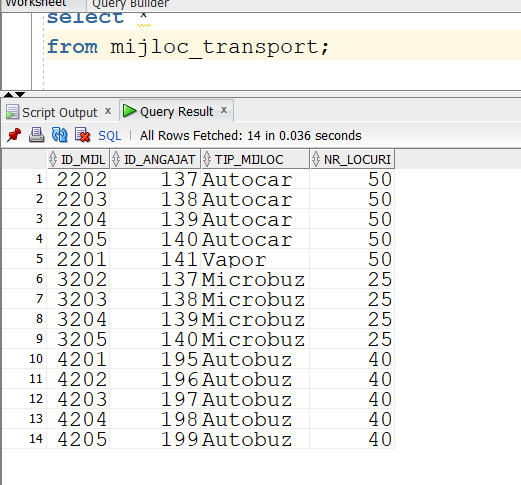


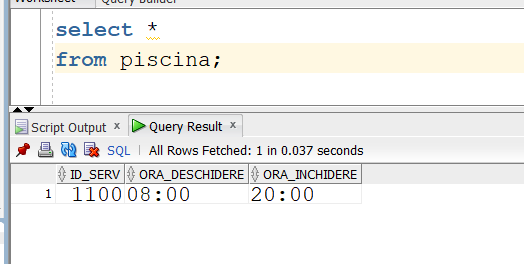


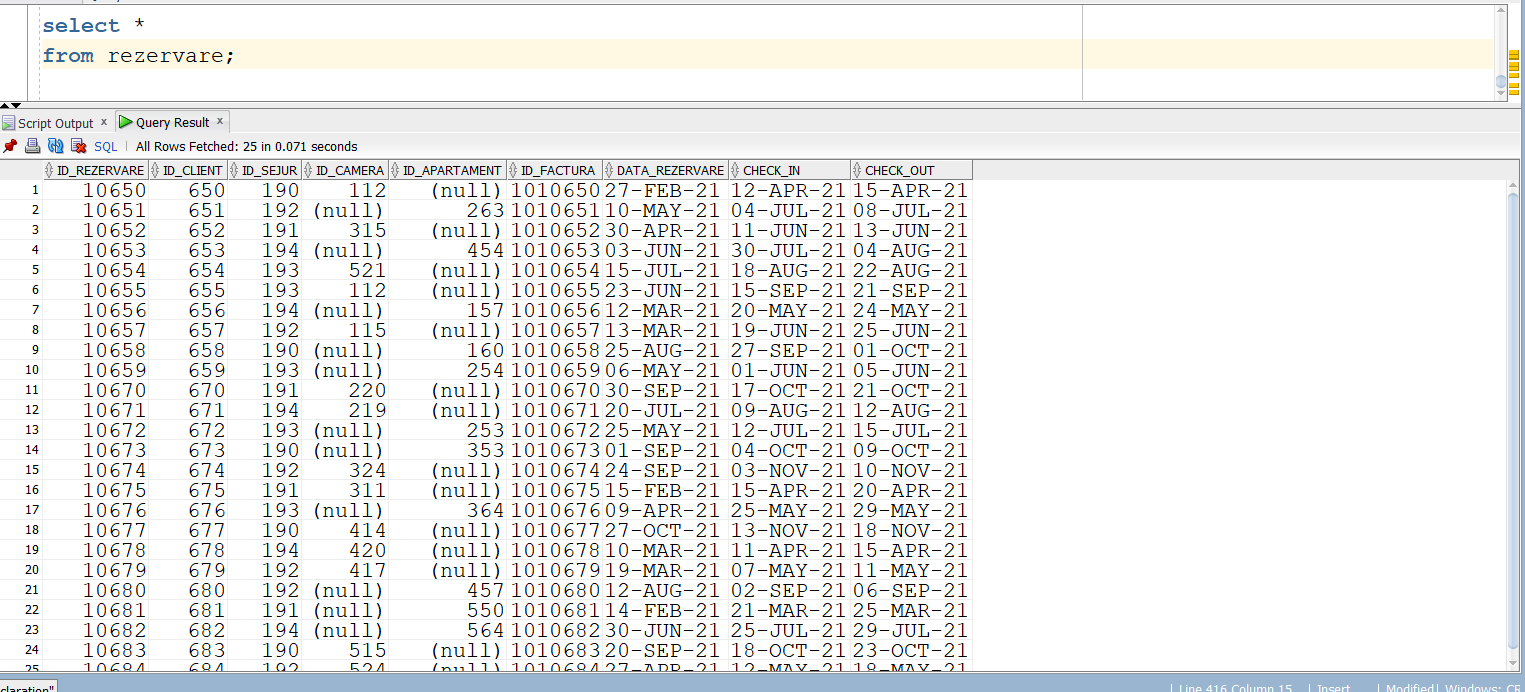


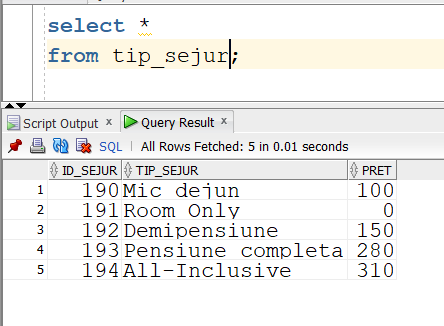


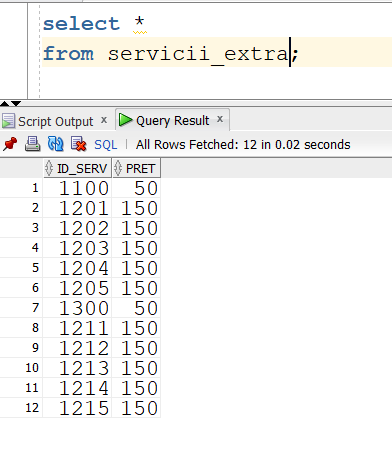


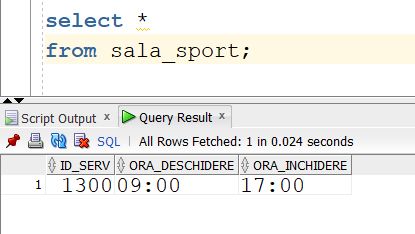






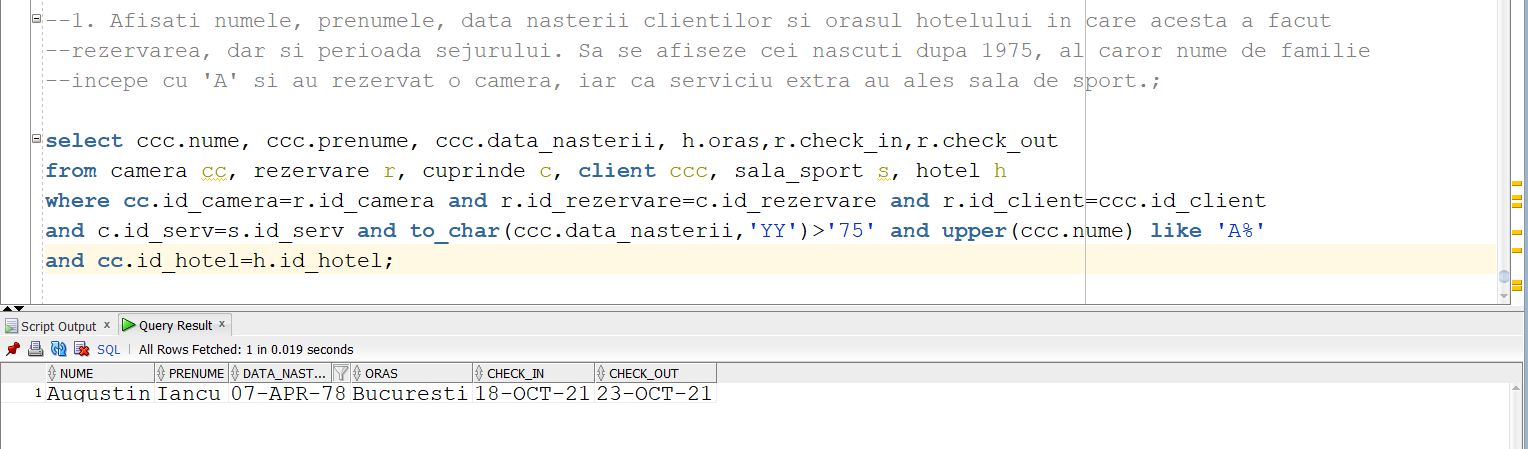




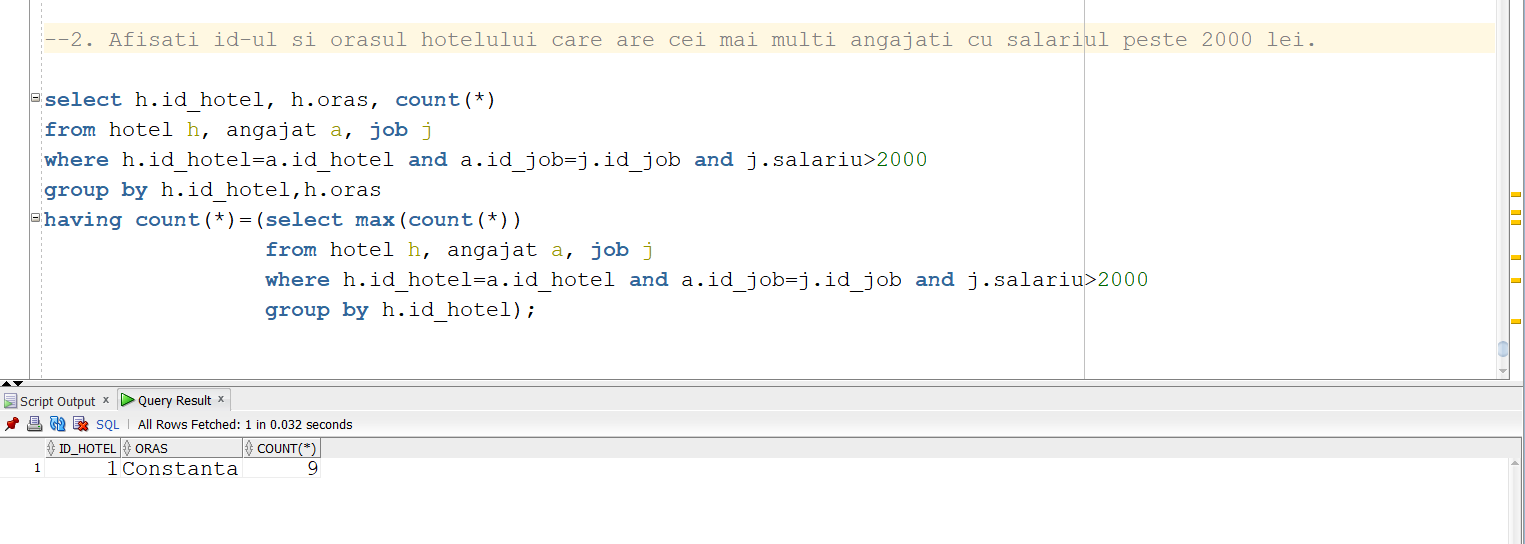


1. Implementare cerinte

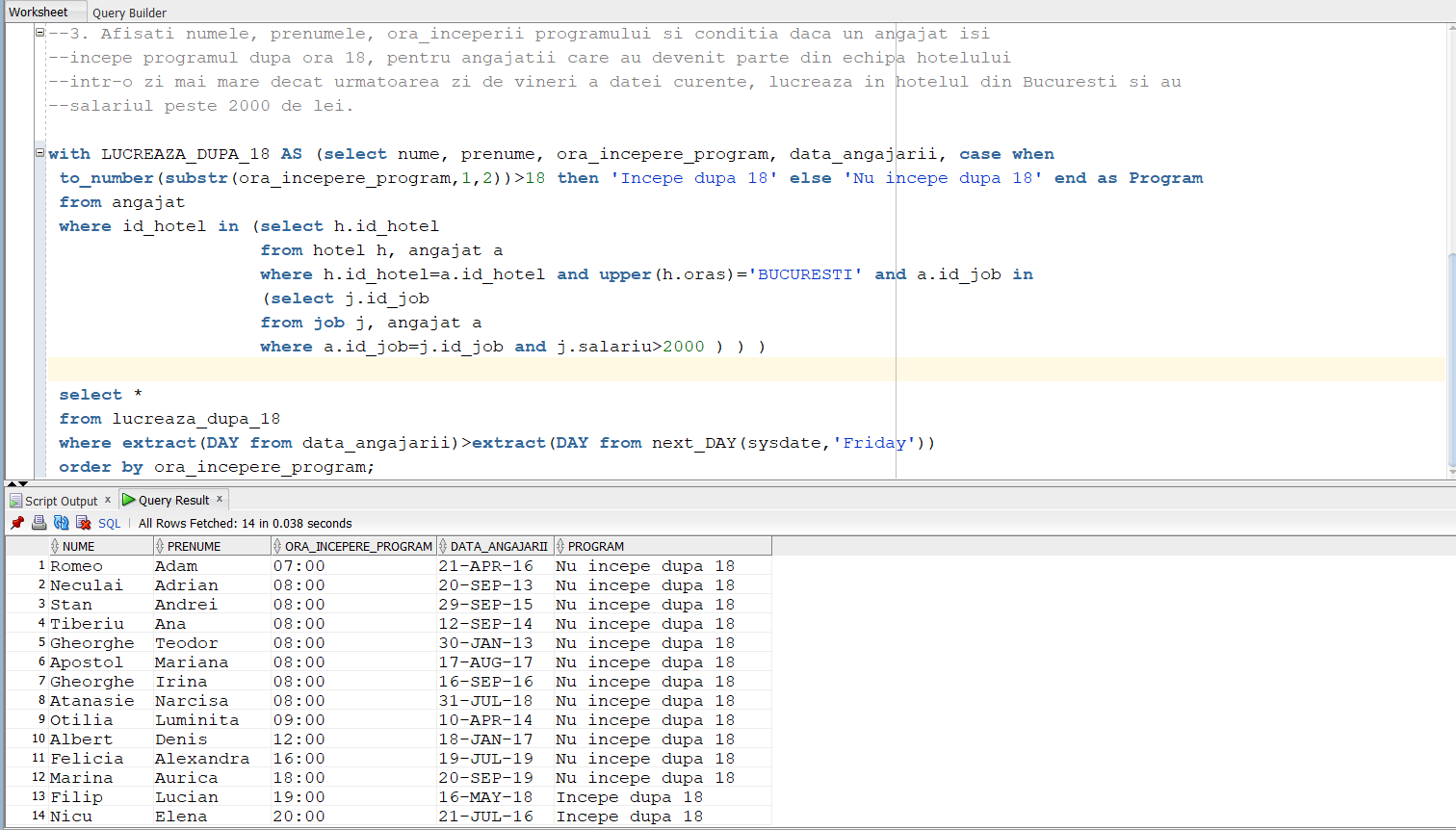
Cerinta 1



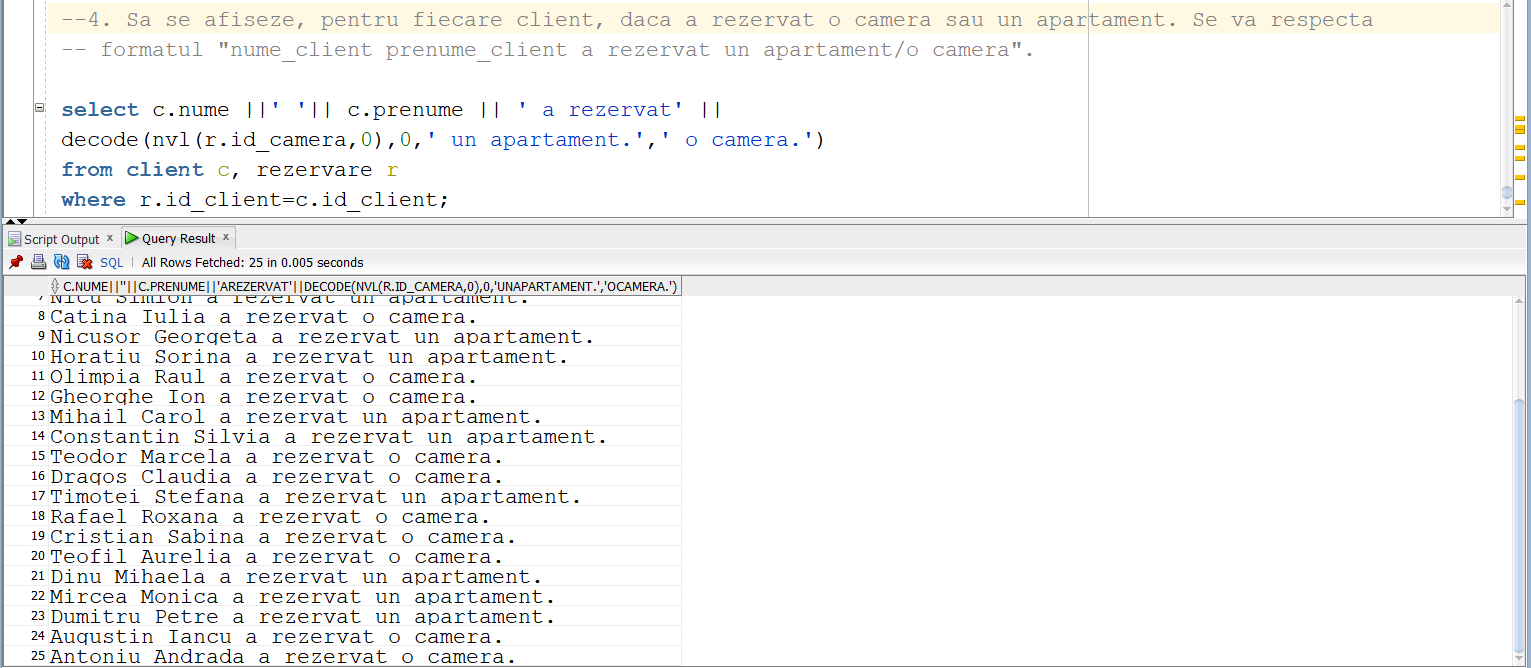
Cerinta 2



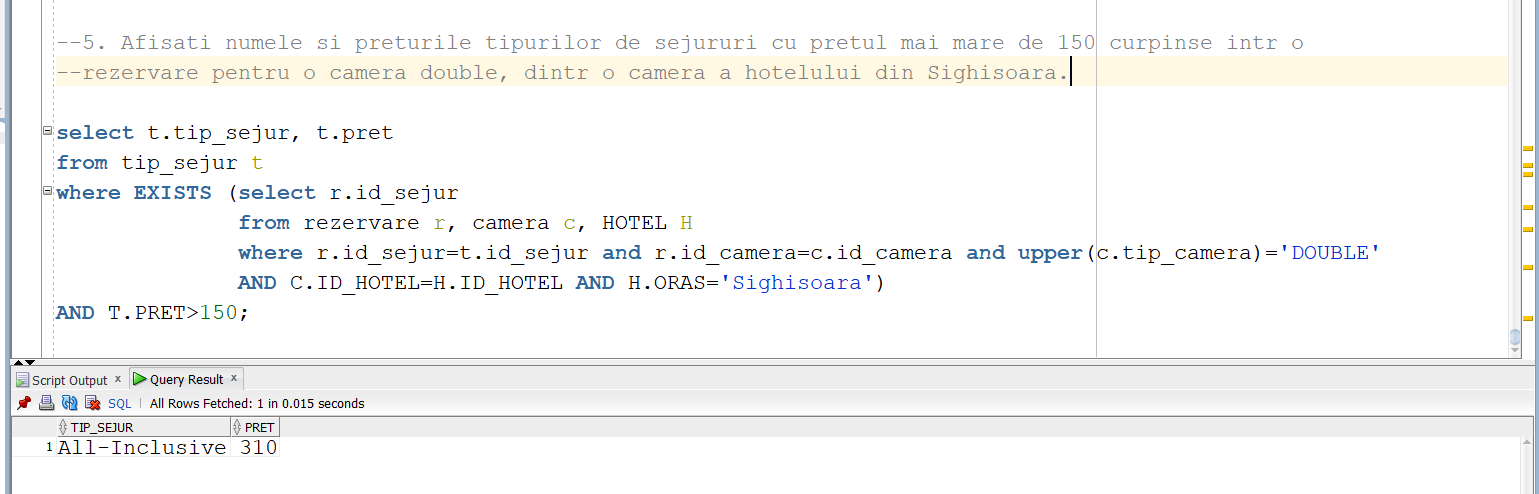
Cerinta 3



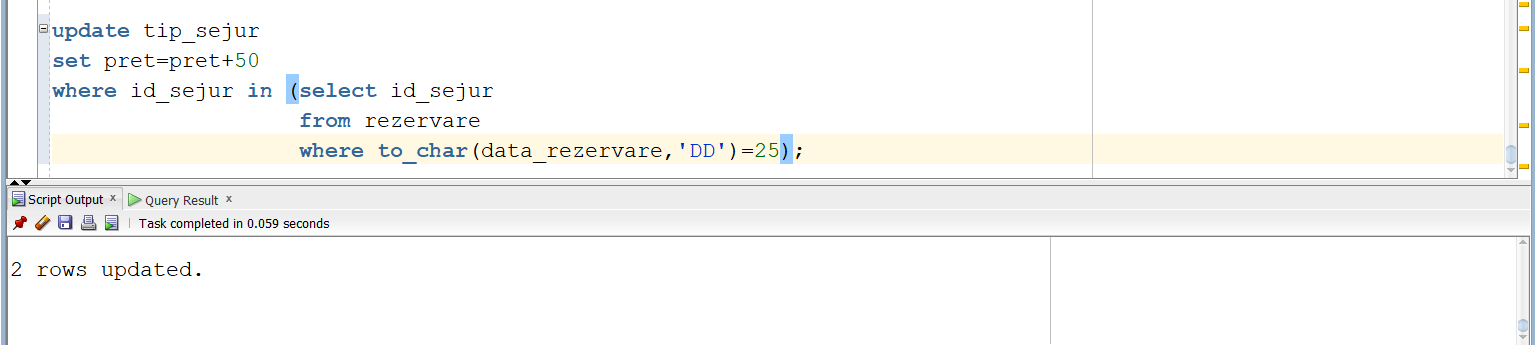
Cerinta 4

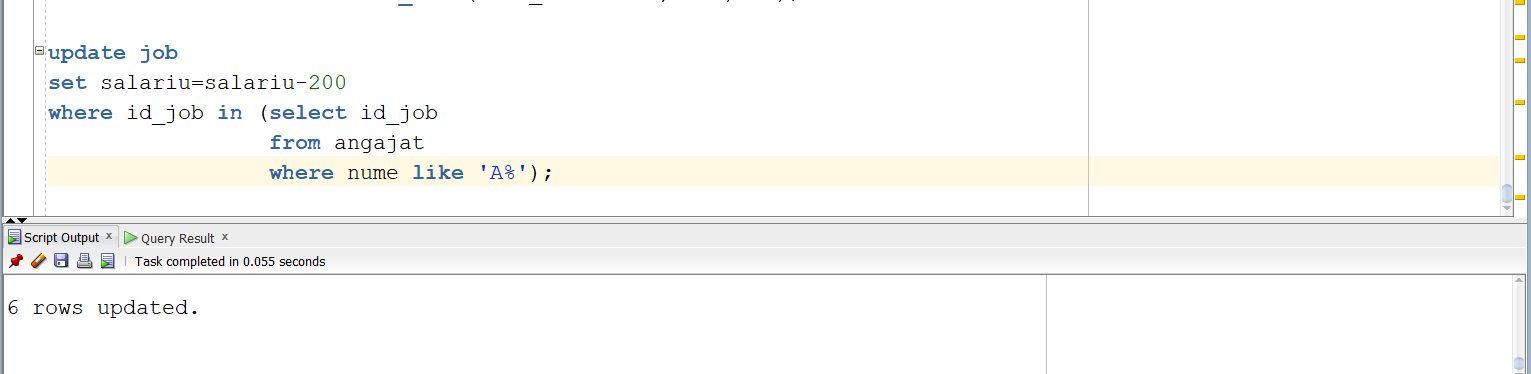


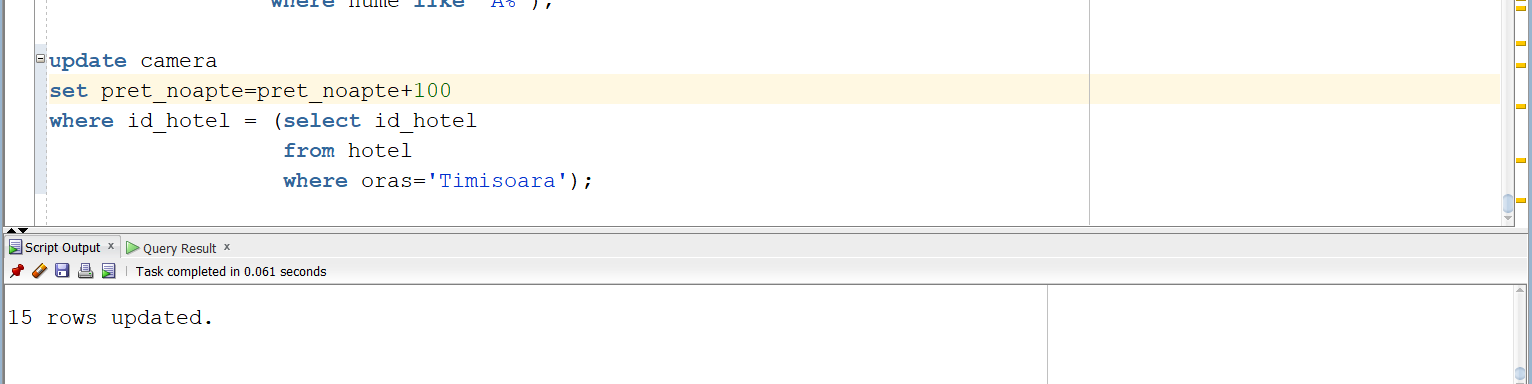
Cerinta 5



1. Implementare actualizari

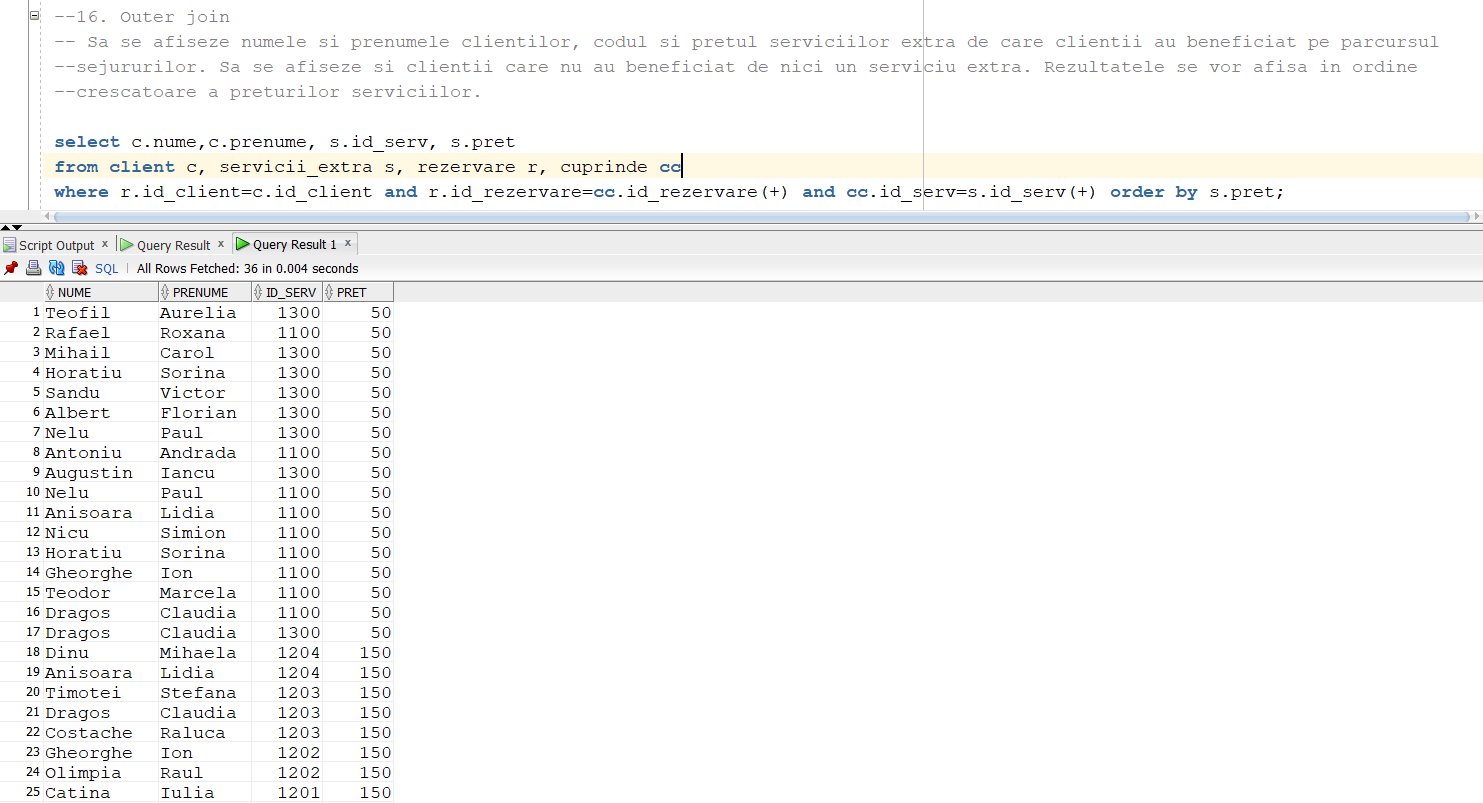


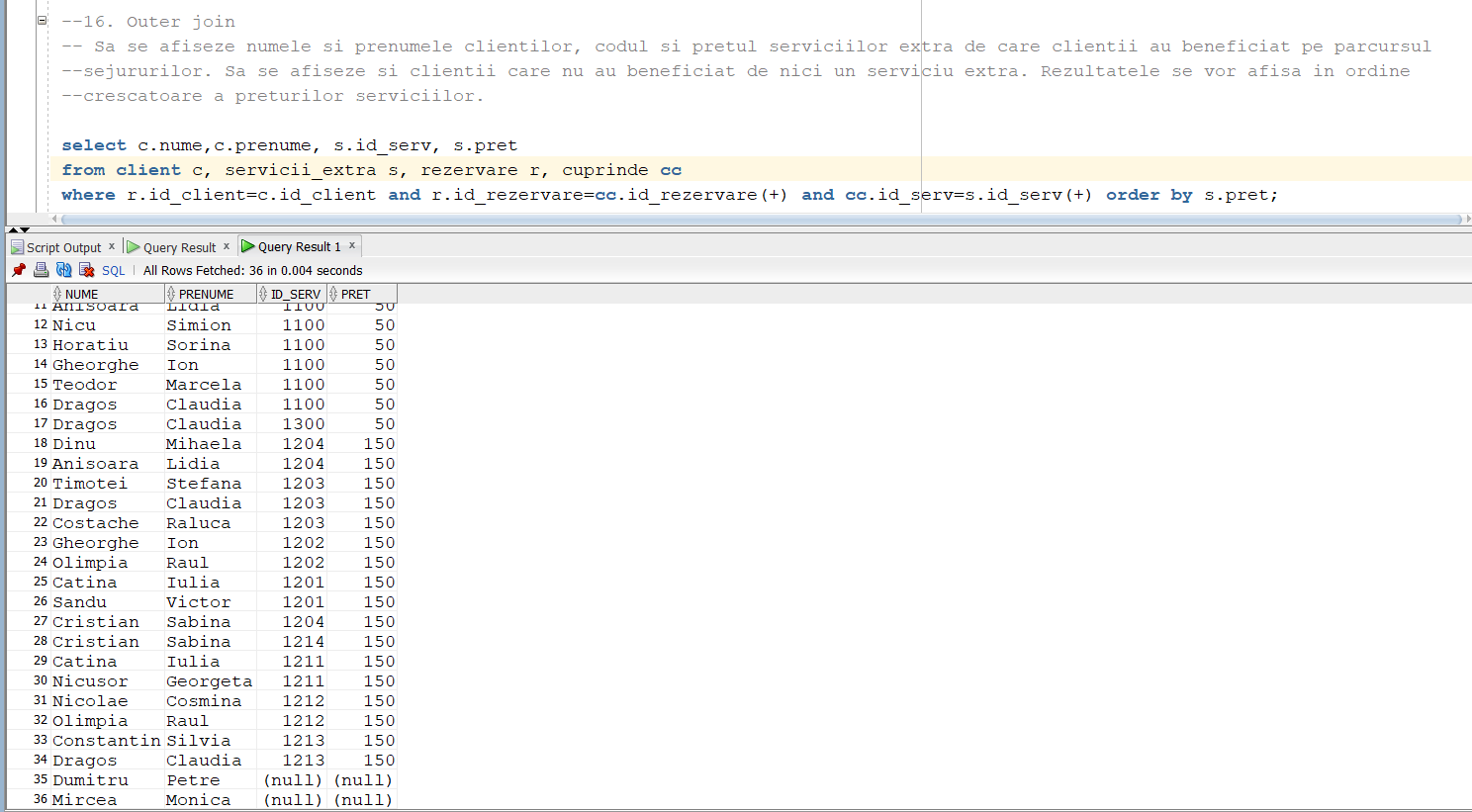




**16**. Cereri pentru outer join + division

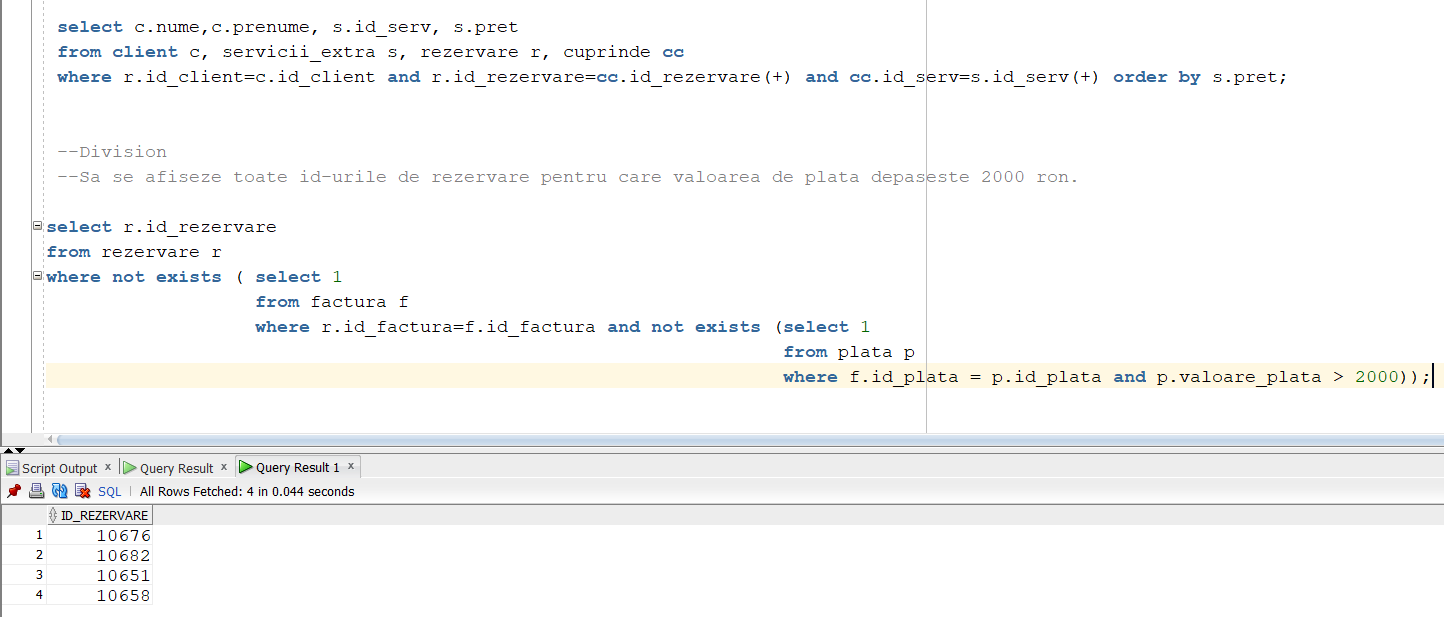
Outer join





Division

--1



--2

